

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства та природокористування
Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д. у. н., проф П. М. Грицюк

«_____» _____ 2026 р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ
«БАКАЛАВР»**

**Інформаційна підтримка функціонування музею з використанням
сучасних автоматизованих систем**

Виконала:

здобувачка вищої освіти за ОПП
«Інформаційні системи та технології»
спеціальності 126 «Інформаційні системи
та технології», групи ІСТ-41

Тихончук Богдана Романівна

Керівник:

к.е.н., доцент Волошин В. С.

Рецензент:

к.п.н., доцент Парфенюк О.В.

Рівне – 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет водного господарства
та природокористування
Навчально-науковий інститут кібернетики,
інформаційних технологій та інженерії

Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень - бакалавр
Освітньо-професійна програма «Інформаційні системи і технології»
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних технологій та
економічної кібернетики
д.е.н., професор П.М. Грицюк

“ ” 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Тихончук Богдана Романівна

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна підтримка функціонування музею з
використанням сучасних автоматизованих систем

керівник роботи: **Волошин Володимир Степанович, к.е.н., доцент**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету С № 476 від 25.04.2026.

2. Термін здачі студентом закінченої роботи 08.06.2025 року

3. Вихідні дані до роботи

посібники, наукові статті, інтернет- джерела за темою роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) :

-проаналізувати застосування методів штучного інтелекту в роботі сучасних музеїв та провести порівняльний аналіз існуючих інформаційних систем і вебсайтів музеїв;

-розробити логічну модель даних інформаційної системи;

-здійснити програмну реалізацію веб-додатку;

-описати функціональні можливості розробленої системи та продемонструвати її практичну цінність.

5. Перелік графічного матеріалу

Презентація за матеріалами бакалаврської роботи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Волошин В.С., доцент кафедри КТЕК		
2	Волошин В.С., доцент кафедри КТЕК		

Дата видачі завдання 19 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів (роботи)	Примітка
1	Робота над Розділом 1	20.01.2026-01.03.2026	
2	Робота над Розділом 2	02.03.2026-25.04.2026	
3	Підготовка пояснювальної записки	26.04.2026-25.05.2026	
4	Написання вступу, висновків, реферату	26.05.2026-08.06.2026	
5	Попередній захист роботи	09.06.2026	
6	Підготовка презентації роботи	10.06.2026	
7	Відгук керівника, рецензування роботи, перевірка на плагіат	11.06.2026	
8	Допуск до захисту	12.06.2026	

Студент(ка) _____ *Тихончук Б. Р.*
(підпис) (прізвище і ініціали)

Керівник кваліфікаційної роботи
_____ *Волошин В.С.*
(підпис) (прізвище і ініціали)

ЗАЯВА

щодо самостійності виконання випускної кваліфікаційної роботи

Я, _____ (ПІБ),
студент(ка) _____ курсу _____ групи _____ ННІ _____

ЗАЯВЛЯЮ:

моя випускна кваліфікаційна робота на тему «_____»
_____»,
яка надається в екзаменаційну комісію із захисту кваліфікаційних робіт зі
спеціальності _____
«_____»
для захисту, виконана самостійно і не містить плагіату.

Всі запозичення з друкованих та електронних джерел, у тому числі із захищених раніше випускових кваліфікаційних робіт мають відповідні посилання.

Я не використовував(ла) шахрайські методи маніпуляції з текстом (заміна букв, інтервали, мікропробіли, білі знаки, парафрази та ін.).

Інструменти штучного інтелекту використовував(ла) без порушення академічної доброчесності.

Я ознайомлений(а) з чинним Порядком перевірки навчальних, кваліфікаційних, навчально-методичних та наукових робіт на наявність ознак академічного плагіату в НУВГП та Положенням про академічну доброчесність в НУВГП, за яким виявлення плагіату є підставою для відмови в допуску моєї роботи до захисту та застосування відповідних санкцій (академічної відповідальності).

АКТ
перевірки випускної кваліфікаційної роботи
автора _____
на рівень запозичень та можливих маніпуляцій з текстом

Відповідно до даних системи StrikePlagiarism файл
« _____ »

містить: ____ % коефіцієнта подібності; ____ % коефіцієнта цитованості;
____ замінених букв; ____ інтервалів; ____ мікропробілів; ____ білих знаків;
____ % ймовірність використання ШІ.

За результатами перевірки засвідчую, що:
автор кваліфікаційної роботи _____:

- самостійно виконав(ла) кваліфікаційну роботу;
- коректно посилався(лась) на використані інформаційні джерела;
- в роботі відсутні ознаки академічної доброчесності.

Керівник кваліфікаційної роботи

підпис

(ПІП)

i Метадані

DOCUMENT

Заголовок

2026_126_bakalavska_TykhonchukBR_VoloshynVS.docx

Автор

Тихончук Богдана Романівна

Науковий керівник / Експерт

Тихончук Богдана Романівна

ID документу

334217062

ORGANIZATION

Назва організації

National University of Water and Environmental Engineering

Підрозділ

National University of Water and Environmental Engineering

REPORT

Дата звіту

6/7/2026

Дата редагування

—

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

22498
Кількість слів176163
Кількість символів

⚠ Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		6
Інтервали		0
Мікропробіли		0
Білі знаки		1434
Парафрази (SmartMarks)		222

≡ Джерела

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

#	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://www.answeroverflow.com/m/1186760682604081182	115 (0.51 %)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 69 с., 45 рис., 5 табл., додатки, 28 літературних джерел.

Актуальність теми даної бакалаврської роботи полягає в тому, що в сучасному суспільстві стрімко зростає роль цифрових технологій у сфері культури та збереження національної спадщини. Інформаційні системи та автоматизовані рішення стають невід'ємною частиною функціонування музеїв, перетворюючи традиційні культурні установи на динамічні, інтерактивні платформи, що здатні глибоко персоналізувати досвід відвідувачів і оптимізувати внутрішні процеси управління колекціями.

Особливої актуальності ці системи набули в умовах цифрової трансформації музейної справи в Україні, де оперативне забезпечення доступності культурної спадщини, інтеграція штучного інтелекту та зручна взаємодія з широкою аудиторією є критично важливими для популяризації сучасного мистецтва та збереження історичної пам'яті. Саме тому розробка повнофункціональної інформаційної системи підтримки діяльності художнього музею з інтеграцією віртуального AI-асистента є актуальним напрямом дослідження.

Об'єкт дослідження бакалаврської роботи – інформаційне забезпечення діяльності музеїв.

Предметною областю бакалаврської роботи є веб-додаток інформаційної системи музею, зокрема його логічна модель даних, програмна реалізація та функціональні можливості з інтеграцією штучного інтелекту.

Метою бакалаврської роботи є проєктування та розробка інформаційної системи функціонування музею з урахуванням сучасних вимог до функціональності, інтерфейсу, масштабованості та технологічної ефективності.

У бакалаврській роботі було: проаналізовано застосування методів штучного інтелекту в роботі сучасних музеїв та проведено порівняльний

аналіз існуючих інформаційних систем і вебсайтів музеїв; розроблено логічну модель даних інформаційної системи; здійснено програмну реалізацію веб-додатку на основі React та Flask з інтеграцією AI-асистента; описано функціональні можливості розробленої системи та продемонстровано її практичну цінність.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, МУЗЕЙ, ШТУЧНИЙ ІНТЕЛЕКТ, ВЕБ-ДОДАТОК, АВТОМАТИЗОВАНА СИСТЕМА, ВІРТУАЛЬНИЙ АСИСТЕНТ.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДІЯЛЬНОСТІ МУЗЕЇВ	7
1.1. Застосування методів штучного інтелекту в роботі сучасних музеїв	7
1.2. Порівняльний аналіз існуючих інформаційних систем музеїв	11
1.3. Порівняльний аналіз інтерфейсних та функціональних рішень вебсайтів музеїв	14
РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ФУНКЦІОНУВАННЯ МУЗЕЮ	22
2.1. Логічна модель даних	22
2.2. Програмна реалізація вебдодатку.....	26
2.3. Функціональні можливості розробленої інформаційної системи	34
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	66
ДОДАТКИ	70

ВСТУП

У сучасному суспільстві стрімко зростає роль цифрових технологій у сфері культури та збереження національної спадщини. Інформаційні системи та автоматизовані рішення стають невід'ємною частиною функціонування музеїв, перетворюючи традиційні культурні установи на динамічні, інтерактивні платформи, що здатні глибоко персоналізувати досвід відвідувачів і оптимізувати внутрішні процеси управління колекціями.

Особливої актуальності ці системи набули в умовах цифрової трансформації музейної справи в Україні, де оперативне забезпечення доступності культурної спадщини, інтеграція штучного інтелекту та зручна взаємодія з широкою аудиторією є критично важливими для популяризації сучасного мистецтва та збереження історичної пам'яті. У таких умовах надзвичайно важливою стає комплексна інформаційна система, що забезпечує ефективне функціонування музею та підвищує якість сервісу для відвідувачів.

На практиці у веб-платформах музеїв часто поєднуються різні підходи до розробки інтерфейсу та функціональності. Наприклад, управління колекціями реалізується за допомогою відкритих платформ, тоді як клієнтська частина та інтерактивні модулі – із застосуванням сучасних фреймворків. У цій роботі було реалізовано повнофункціональну інформаційну систему підтримки діяльності художнього музею сучасного мистецтва з інтеграцією віртуального AI-асистента.

Метою бакалаврської роботи є проєктування та розробка інформаційної системи функціонування музею з урахуванням сучасних вимог до функціональності, інтерфейсу, масштабованості та технологічної ефективності.

Об'єктом дослідження є інформаційне забезпечення діяльності музеїв.

Предметом дослідження є веб-додаток інформаційної системи музею, зокрема його логічна модель даних, програмна реалізація та функціональні можливості з інтеграцією штучного інтелекту.

Завданнями бакалаврської роботи є:

- проаналізувати застосування методів штучного інтелекту в роботі сучасних музеїв та провести порівняльний аналіз існуючих інформаційних систем і вебсайтів музеїв;
- розробити логічну модель даних інформаційної системи;
- здійснити програмну реалізацію веб-додатку;
- описати функціональні можливості розробленої системи та продемонструвати її практичну цінність.

Інструменти, використані під час виконання роботи: React, Flask, SQLAlchemy, styled-components, AI API, PostgreSQL/SQLite, Visual Studio Code, Git.

РОЗДІЛ 1. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДІЯЛЬНОСТІ МУЗЕЇВ

1.1. Застосування методів штучного інтелекту в роботі сучасних музеїв

Сучасний розвиток штучного інтелекту кардинально змінює підходи до організації музейної діяльності, перетворюючи традиційні культурні установи на динамічні, інтерактивні платформи, що здатні глибоко персоналізувати досвід відвідувачів і оптимізувати внутрішні процеси.

Аналіз останніх тенденцій, оприлюднених у наукових публікаціях та звітах культурних інституцій, свідчить про те, що методи штучного інтелекту стають невід'ємною складовою інформаційного забезпечення музеїв, дозволяючи поєднувати збереження культурної спадщини з інноваційними формами комунікації [1].

Завдяки алгоритмам машинного навчання, обробці природної мови та комп'ютерному зору музеї отримують інструменти для аналізу великих обсягів даних, створення персоналізованого контенту та підвищення доступності експозицій для широкої аудиторії, включаючи людей з особливими потребами.

На основі аналізу сучасних практик музейних інституцій було розроблено схему основних напрямів застосування штучного інтелекту в музейній діяльності, що представлена на рис. 1.1.



Рис. 1.1. Схема інтеграції методів штучного інтелекту в інформаційні системи сучасних музеїв

Одним із найпоширеніших і найбільш ефективних напрямів є впровадження інтелектуальних чат-ботів та віртуальних асистентів, які працюють на базі великих мовних моделей [2]. Такі системи дозволяють відвідувачам у реальному часі отримувати відповіді на запитання про експонати, історію створення творів чи особливості експозицій, значно збагачуючи освітній потенціал музеїв. За даними міжнародних досліджень, чат-боти підвищують рівень залученості аудиторії на 40–60 %, оскільки адаптують відповіді під індивідуальні інтереси користувача, рівень знань та навіть поточне місцезнаходження в залах.

Поряд із цим важливу роль відіграють рекомендаційні системи, які на основі аналізу попередніх відвідувань, уподобань та поведінки відвідувачів формують персоналізовані маршрути екскурсій [3]. Такі алгоритми враховують демографічні характеристики, час перебування в музеї та навіть емоційний відгук на певні експонати, що дає змогу перетворити стандартний огляд на унікальну культурну подорож. Подібні технології не лише

підвищують задоволеність відвідувачів, але й сприяють рівномірному розподілу потоків, зменшуючи перевантаження окремих залів. Описані в таблиці 1.1.

Таблиця 1.1. Основні методи штучного інтелекту та їх практичне застосування в музейній сфері

Метод штучного інтелекту	Основне застосування в музеях	Переваги для відвідувачів та установи
Обробка природної мови (NLP)	Чат-боти, віртуальні гід	Персоналізовані відповіді, доступність 24/7
Рекомендаційні системи	Персоналізовані маршрути екскурсій	Збільшення залученості, оптимізація потоків
Комп'ютерний зір	Розпізнавання експонатів, цифровізація колекцій	Автоматизація каталогізації, допомога в реставрації
Генеративний ШІ	Створення описів, віртуальних експозицій	Збагачення контенту, створення інтерактивних матеріалів
Аналітика великих даних	Прогнозування відвідуваності, аналіз поведінки	Оптимізація управління ресурсами

Комп'ютерний зір та генеративні моделі відкривають нові можливості для роботи з колекціями [4]. Алгоритми автоматично розпізнають і класифікують тисячі експонатів, прискорюючи процеси інвентаризації та цифрової архівації. У поєднанні з технологіями доповненої реальності вони дозволяють створювати віртуальні реконструкції пошкоджених артефактів або інтерактивні 3D-моделі, доступні як у фізичному просторі музею, так і онлайн. Такі підходи особливо актуальні для музеїв України, де тривають активні дискусії щодо впровадження штучного інтелекту для оцифрування культурної спадщини та створення електронних реєстрів [5].

Значний потенціал демонструють аналітичні системи на базі штучного інтелекту, які обробляють дані про відвідуваність, тривалість перебування біля експонатів та маршрути руху [6]. Отримана інформація допомагає керівництву

музеїв прогнозувати пікові навантаження, оптимізувати графіки роботи персоналу та розробляти цільові освітні програми.

У контексті українського музейного середовища подібні інструменти обговорюються на фахових зустрічах музейників Донеччини, Полтавщини та інших регіонів, де акцент робиться на етичних аспектах використання штучного інтелекту та його інтеграції в щоденну роботу установ.

Інтеграція штучного інтелекту в музейну сферу супроводжується певними викликами, серед яких етичні питання захисту даних відвідувачів, забезпечення точності алгоритмів та збереження людського фактора в культурній комунікації [7]. Водночас експерти наголошують, що ШІ не замінює, а доповнює роботу музейних фахівців, звільняючи їхній час для творчих завдань і поглибленого наукового дослідження.

Перспективи розвитку технологій вказують на подальше поширення мультимовних віртуальних асистентів, інтеграцію з мобільними додатками та створення гібридних просторів, де фізичні експозиції органічно поєднуються з цифровими інтерактивними елементами.

Таким чином, застосування методів штучного інтелекту в роботі сучасних музеїв є потужним каталізатором цифрової трансформації культурних установ. Воно сприяє підвищенню доступності, персоналізації та наукової ефективності музейної діяльності, роблячи культурну спадщину ближчою і зрозумілішою для кожного відвідувача.

Подальший розвиток цих технологій, з урахуванням національного контексту та етичних стандартів, дозволить музеям України та світу вийти на якісно новий рівень взаємодії з аудиторією, забезпечуючи гармонійне поєднання традицій і інновацій у сфері культурної спадщини.

1.2. Порівняльний аналіз існуючих інформаційних систем музеїв

Сучасні інформаційні системи відіграють ключову роль у забезпеченні ефективного функціонування музейних закладів, сприяючи не лише систематизації та зберіганню колекцій, а й активному залученню відвідувачів до культурної спадщини через цифрові канали.

У контексті цифрової трансформації музейної справи порівняльний аналіз провідних рішень дозволяє виявити особливості їхньої архітектури, функціональних можливостей та адаптивності до потреб різних типів установ – від невеликих регіональних музеїв до великих національних інституцій. Такий аналіз базується на вивченні відкритих і комерційних платформ, що підтримують стандарти опису музейних об'єктів, мультимедійний контент та інтеграцію з веб-ресурсами [8].

Серед відкритих рішень особливе місце посідає Omeka – гнучка веб-платформа, орієнтована на публікацію цифрових колекцій та створення медіа-насичених онлайн-виставок. Система забезпечує просте управління метаданими за стандартом Dublin Core, підтримує розширення через плагіни, дозволяє створювати інтерактивні експозиції з зображеннями, аудіо та відео, а також інтегрується з іншими GLAM-системами (галереї, бібліотеки, архіви, музеї). Omeka Classic підходить для невеликих проєктів завдяки швидкому розгортанню, тоді як Omeka S орієнтована на інституції з великими ресурсними пулами та потребою в багатосайтовій архітектурі. Її відкритий код сприяє демократизації доступу до культурних даних і стимулює спільнотне наповнення контенту [9].

Комерційні системи, такі як Argus від Lucidea, пропонують більш комплексне управління колекціями на рівні професійних музейних процесів. Argus забезпечує повний цикл роботи з експонатами: від реєстрації надходжень та інвентаризації до відстеження місцезнаходження, мультимедійного супроводу та багатомовної підтримки.

Система акцентує увагу на оптимізації внутрішніх робочих процесів,

включає інструменти для аналізу відвідувачьких уподобань, експорту даних та створення динамічних онлайн-порталів. Її сильною стороною є інтеграція з веб-технологіями, що дозволяє музеям швидко формувати цифрову присутність без значних додаткових витрат на розробку [10].

Ще одним прикладом комплексного рішення є MINISIS Museum Management від Soutron, яке позиціонується як універсальна система культурного управління активами.

Платформа об'єднує функціональність для музеїв, архівів та бібліотек в єдиному середовищі, підтримуючи понад 5000 попередньо визначених полів, гнучку базу даних HRDBMS та відповідність міжнародним стандартам (AACRII, MARC/RDA, ISAD(G), ISAAR, Dublin Core).

MINISIS забезпечує управління виставками, страхуванням артефактів, рухом предметів, консервацією, відвідувачькими послугами, а також пропонує уніфікований пошуковий інтерфейс (OPAC) з підтримкою Boolean-операторів та пошуку по всіх типах колекцій. Особливістю системи є акцент на цифровому збереженні та крос-активному доступі, що робить її придатною для великих установ з різноманітними фондами [11].

Поряд з цими рішеннями в музейній практиці активно застосовуються й інші платформи, зокрема CollectiveAccess та Axiell Collections, які також демонструють тенденцію до інтеграції традиційного управління колекціями з сучасними цифровими сервісами. CollectiveAccess, наприклад, вирізняється гнучкістю моделювання метаданих та підтримкою відкритих стандартів, що полегшує обмін даними між установами.

Axiell, у свою чергу, орієнтована на масштабні інституції та пропонує розвинені інструменти аналітики та мобільного доступу. Загальною рисою сучасних систем є перехід від ізольованих баз даних до хмарних, мультимедійних платформ, що забезпечують не лише внутрішнє адміністрування, а й публічний доступ до культурних ресурсів [12].

Для об'єктивної оцінки доцільно звернутися до порівняльної характеристики основних рішень, що наведена в таблиці 1.2.

Таблиця 1.2. Порівняльна характеристика інформаційних систем управління музейними колекціями

Система	Тип	Основні функції	Цільова аудиторія	Переваги	Недоліки
Omeka	Відкрита (open-source)	Публікація цифрових колекцій, створення виставок, підтримка Dublin Core, плагіни	Невеликі та середні музеї, архіви, дослідники	Безкоштовність, гнучкість, легкість розгортання	Обмежені інструменти для складного управління колекціями
Argus (Lucidea)	Комерційна	Повний цикл управління колекціями, інвентаризація, мультимедіа, аналітика відвідувачів	Середні та великі музеї	Оптимізація робочих процесів, сильна веб-інтеграція	Комерційна модель ліцензування
MINISIS (Soutron)	Комерційна	Уніфіковане управління активами, виставки, консервація, ОРАС, цифрове збереження	Великі культурні установи	Висока адаптивність, підтримка багатьох стандартів	Складність початкового налаштування

Як видно з таблиці, вибір системи залежить від розміру музею, наявних ресурсів та пріоритетів – чи то акцент на публічному доступі та цифрових виставках, чи на внутрішньому управлінні та аналітиці. Відкриті платформи на кшталт Omeka сприяють демократизації культурного контенту, тоді як комерційні рішення забезпечують глибшу інтеграцію бізнес-процесів і професійну підтримку [13].

Аналіз сучасних тенденцій свідчить про поступове поширення хмарних технологій, підтримку штучного інтелекту для автоматичної категоризації об'єктів та посилення орієнтації на користувача. Українські музеї, які впроваджують цифрові практики, також орієнтуються на ці міжнародні

рішення, адаптуючи їх до національних потреб збереження історичної пам'яті та культурної спадщини в умовах цифрової трансформації [14].

Таким чином, порівняльний аналіз підкреслює, що ефективна інформаційна система музею має поєднувати технічну надійність, стандартизацію даних та гуманістичну спрямованість на збагачення культурного досвіду відвідувачів.

1.3. Порівняльний аналіз інтерфейсних та функціональних рішень вебсайтів музеїв

Сучасні вебсайти музеїв стали невід'ємною частиною інформаційного забезпечення їхньої діяльності, перетворюючись на динамічні цифрові платформи, що поєднують освітні, комунікаційні та комерційні функції. Інтерфейсні рішення визначають, наскільки зручно та приємно користувачеві взаємодіяти з ресурсом, тоді як функціональні можливості розширюють межі фізичного музею, роблячи культурну спадщину доступною для глобальної аудиторії.

Порівняльний аналіз підходів різних закладів дозволяє не лише виявити загальні тенденції розвитку цифрових музейних ресурсів, але й зрозуміти, як саме дизайн і функціональність впливають на залученість відвідувачів, ефективність поширення знань та збереження культурної спадщини [15].

Особливо показовим є приклад вебсайту Musée du Louvre (louvre.fr), який вирізняється елегантним і водночас сучасним інтерфейсом, що гармонійно поєднує класичну музейну естетику з передовими цифровими технологіями (рис. 1.2). Дизайн сайту характеризується чистотою ліній, високоякісними фотографіями експонатів та плавними переходами між розділами, що створює відчуття занурення вже на головній сторінці. Навігація реалізована через інтуїтивне горизонтальне меню та потужну пошукову систему з автозаповненням, а повна адаптивність забезпечує комфортний перегляд на будь-яких пристроях.

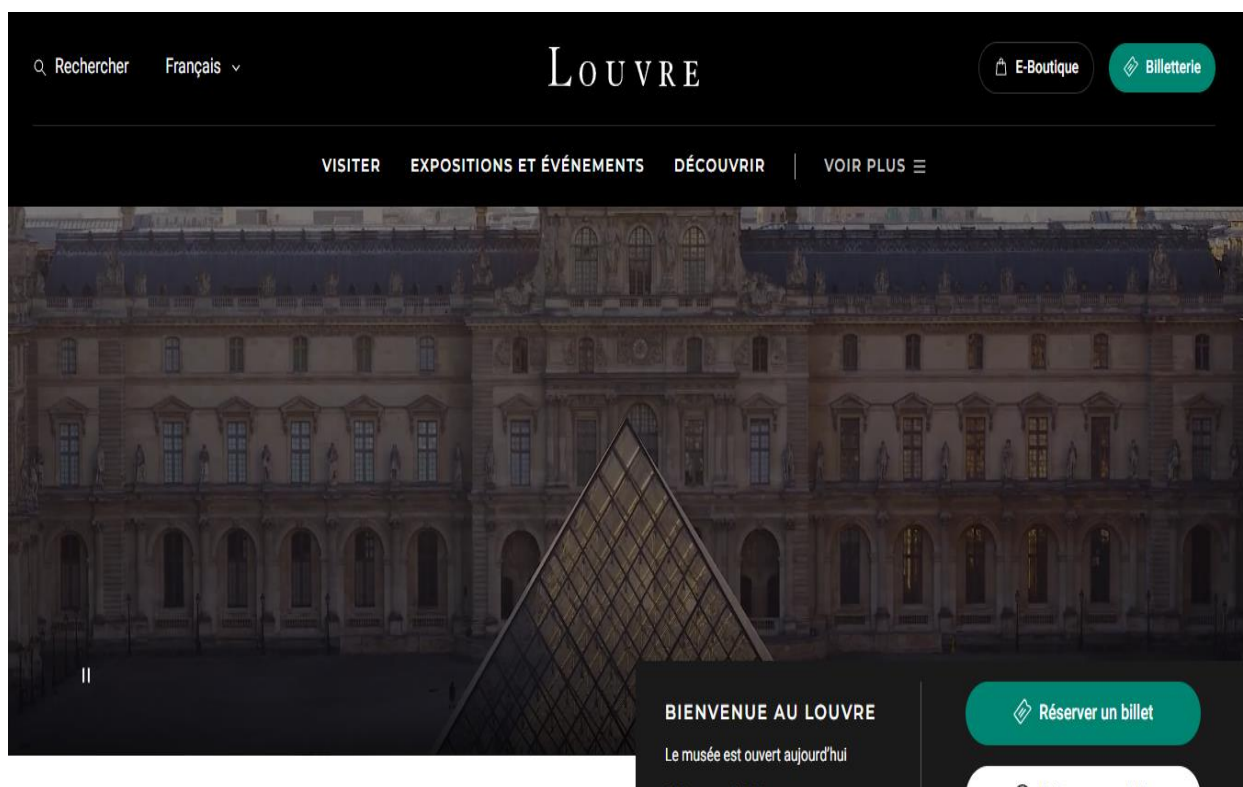


Рис. 1.2. Інтерфейс Musée du Louvre

Функціонально ресурс пропонує детальні віртуальні тури по залах музею, систему онлайн-бронювання квитків з вибором дати, часу та типу квитка, доступ до цифрових колекцій з можливістю масштабування зображень у високій роздільній здатності, а також актуальний розділ новин і тимчасових виставок, що оновлюється в реальному часі. Такий комплексний підхід не лише полегшує планування фізичного візиту, але й дозволяє отримувати глибокі знання онлайн, роблячи Лувр взірцем для багатьох культурних установ [16].

Схожі принципи, але з акцентом на освітню місію, реалізовано на вебсайті The British Museum (britishmuseum.org). Тут інтерфейс, що показаний на рис. 1.3, вирізняється функціональним мінімалізмом, великими зображеннями артефактів та чіткою ієрархією контенту, що робить ресурс максимально зручним для дослідників, студентів і широкої публіки. Навігація підтримується детальною пошуковою системою з численними фільтрами за

періодом, регіоном чи типом об'єкта, а сайт повністю відповідає міжнародним стандартам доступності, включаючи опції для людей зі слабким зором.

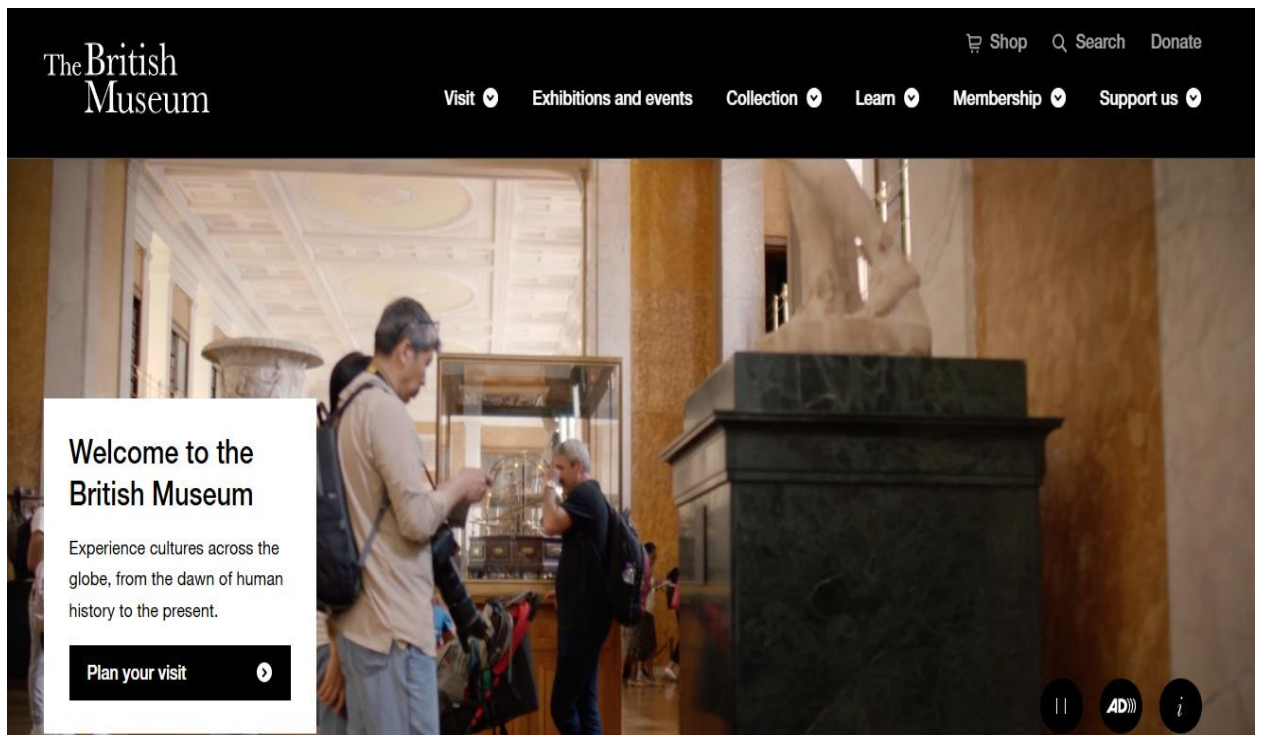


Рис. 1.3. Інтерфейс The British Museum

Функціональні можливості вражають обсягом: користувачі мають доступ до мільйонів оцифрованих експонатів з детальними описами, віртуальних галерей, інтерактивних карт і навіть можливості створення власних персональних колекцій. Крім того, доступні онлайн-лекції, освітні матеріали для шкіл та інтегрований магазин, що перетворює сайт на повноцінний освітній хаб [17].

The Metropolitan Museum of Art (metmuseum.org) демонструє ще один інноваційний підхід, де інтерфейс поєднує сучасний мінімалізм з динамічними елементами. Головна сторінка (рис. 1.4) постійно оновлюється рекомендаціями експонатів і актуальними подіями, а навігація реалізована через зручне бокове меню та інтелектуальний пошук. Особливу увагу приділено інтерактивності: таймлайни, 3D-моделі та персоналізовані рекомендації роблять взаємодію з контентом захопливою. Функціонально сайт

пропонує одну з найбільших у світі цифрових колекцій, онлайн-виставки, освітні програми та систему членства з персональними пропозиціями, що дозволяє кожному користувачеві відчувати себе частиною музейної спільноти [18].

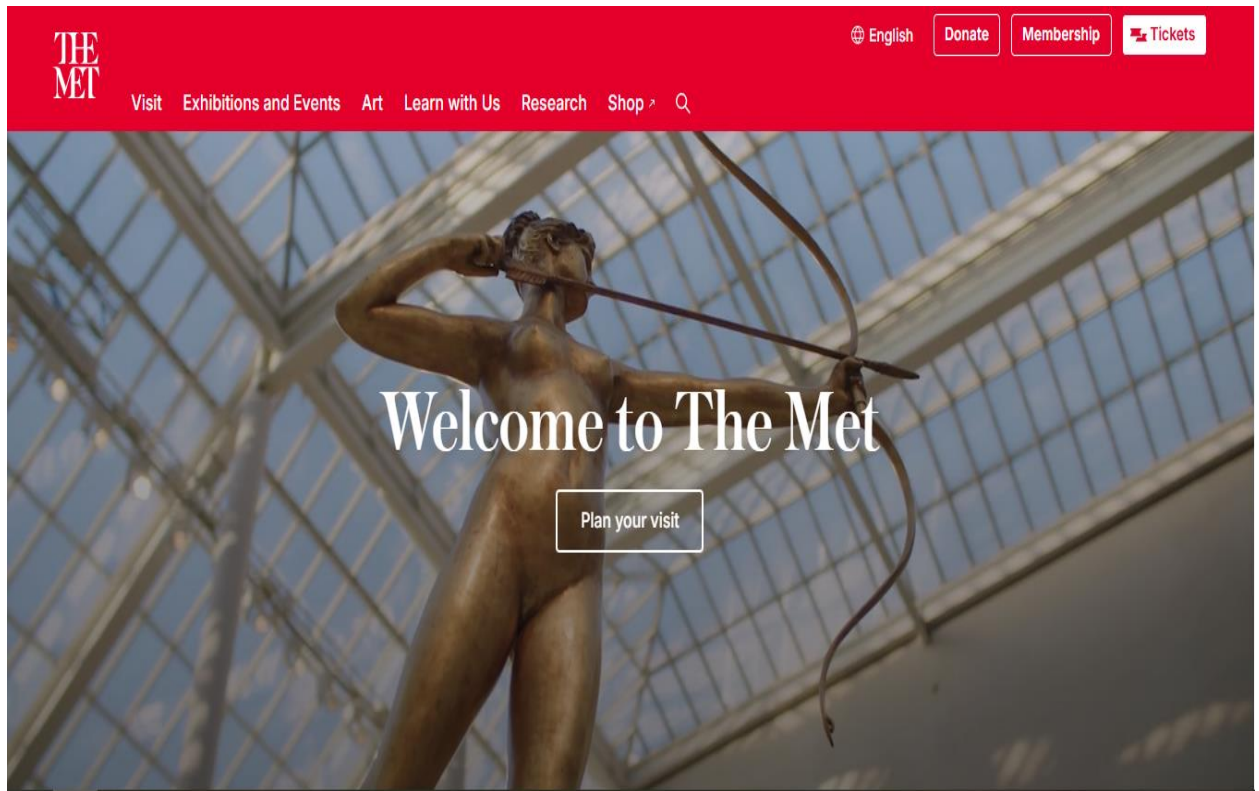


Рис. 1.4. Інтерфейс головної сторінки The Metropolitan Museum of Art

Національний художній музей України (namu.ua) представляє національний варіант цифрового музею, де інтерфейс органічно поєднує функціональність із елементами українського культурного коду (рис. 1.5). Дизайн сайту сучасний і лаконічний, з акцентом на актуальні виставки, новини та практичну інформацію про відвідування. Навігація проста й інтуїтивна, а повна адаптивність забезпечує доступність для користувачів мобільних пристроїв.

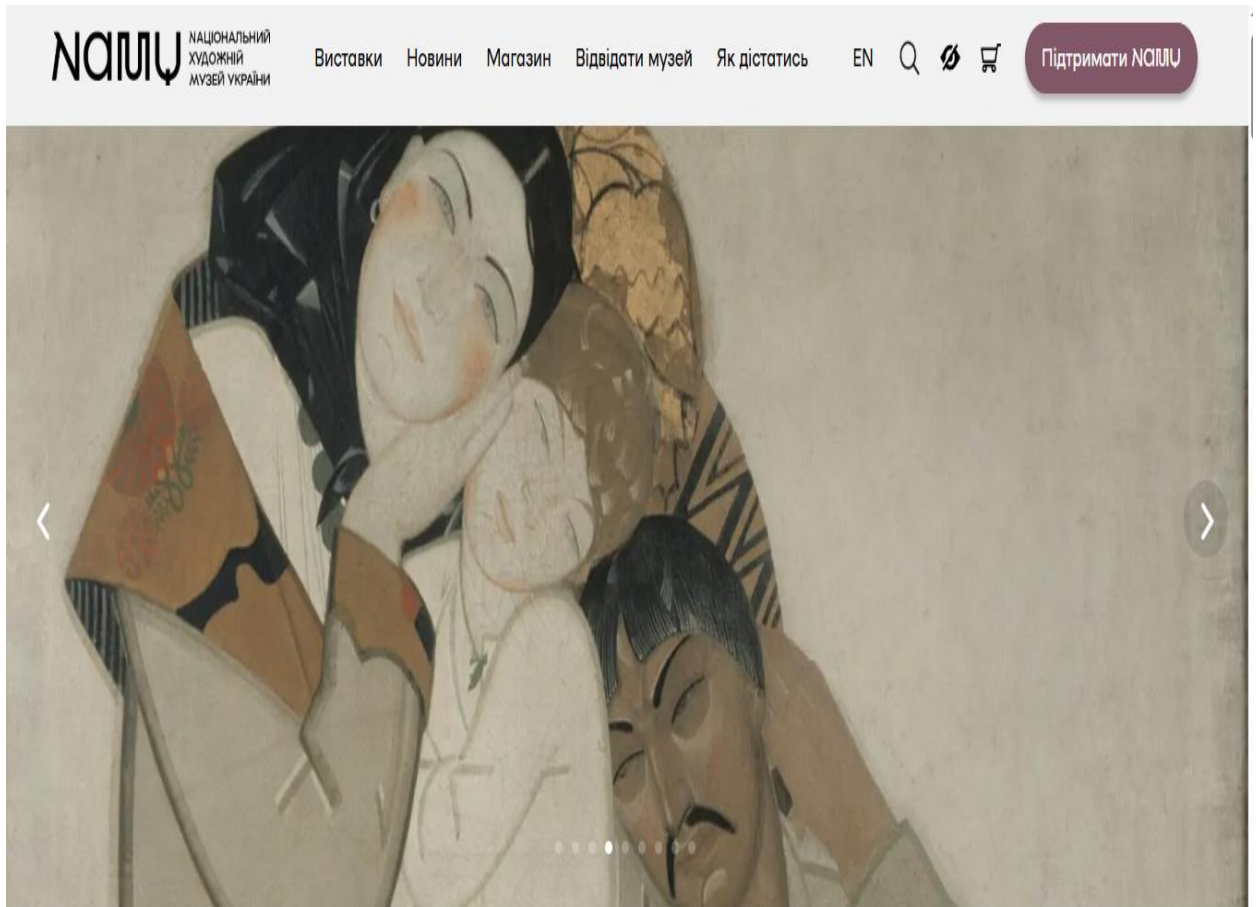


Рис. 1.5. Інтерфейс Національного художнього музею України

Функціонально ресурс пропонує детальну інформацію про експозиції, можливість бронювання квитків, цифрові матеріали та розділ підтримки музею, що особливо важливо в умовах сучасних викликів.

Такий підхід відображає специфіку національних музеїв, де цифрові рішення спрямовані не лише на збереження спадщини, але й на її активне поширення серед співвітчизників та міжнародної аудиторії [19].

Для наочності порівняння основних характеристик інтерфейсних та функціональних рішень розглянутих вебсайтів доцільно звернутися до узагальненої таблиці 1.3.

Таблиця 1.3. Порівняльна характеристика інтерфейсних та функціональних рішень вебсайтів музеїв

Музей	Інтерфейсні рішення (дизайн, навігація, адаптивність)	Функціональні можливості (бронювання, колекції, інтерактив)
Musée du Louvre	Елегантний мінімалізм, високоякісні зображення, інтуїтивне меню, повна адаптивність	Віртуальні тури, онлайн-бронювання, цифрові колекції з масштабуванням, новини
The British Museum	Функціональний мінімалізм, потужний пошук з фільтрами, високий рівень доступності	Мільйони оцифрованих об'єктів, віртуальні галереї, персональні колекції, освітні матеріали
The Metropolitan Museum of Art	Динамічний мінімалізм, персоналізовані рекомендації, інтерактивні елементи	Величезна цифрова колекція, таймлайни, 3D-моделі, система членства
Національний художній музей України	Лаконічний сучасний дизайн з національним акцентом, проста навігація, адаптивність	Інформація про виставки, бронювання квитків, цифрові матеріали, підтримка музею

Як видно з наведеного порівняння, усі розглянуті ресурси демонструють високий рівень адаптивності та уваги до користувацького досвіду, проте відрізняються акцентами: від естетичної вишуканості Лувру до освітньої глибини Британського музею чи інтерактивності Метрополітен.

Національний художній музей України, зберігаючи національну специфіку, поступово наближається до світових стандартів, що свідчить про загальну тенденцію цифрової трансформації музейної сфери.

Подальший розвиток таких вебсайтів, ймовірно, буде пов'язаний із ще глибшою інтеграцією інтерактивних технологій, персоналізації контенту та забезпеченням універсальної доступності, що дозволить музеям ефективніше виконувати свою місію в цифровому світі [20], [21].

Таким чином, порівняльний аналіз підтверджує, що успішні інтерфейсні та функціональні рішення вебсайтів музеїв ґрунтуються на балансі естетики, зручності та змістовності, забезпечуючи не лише інформаційну підтримку діяльності закладів, але й активну популяризацію культурної спадщини серед сучасної аудиторії.

Сучасна музейна діяльність зазнає глибокої цифрової трансформації, у якій інформаційне забезпечення відіграє визначальну роль. Інтеграція методів штучного інтелекту дозволяє суттєво підвищити рівень персоналізації відвідувачького досвіду, оптимізувати внутрішні процеси управління колекціями та зробити культурну спадщину доступнішою для широкої аудиторії, включаючи осіб з особливими потребами.

Алгоритми машинного навчання, обробка природної мови, комп'ютерний зір та генеративні моделі стають потужними інструментами створення інтелектуальних чат-ботів, рекомендаційних систем, автоматизованої каталогізації та аналітики відвідуваності, водночас вимагаючи уважного ставлення до етичних питань захисту даних і збереження людського фактора в культурній комунікації.

Порівняльний аналіз інформаційних систем управління музейними колекціями демонструє різноманітність рішень, що відрізняються за типом, функціональними можливостями та адаптивністю до потреб установ різного масштабу. Відкриті платформи забезпечують гнучкість, демократичний доступ до цифрового контенту та легкість розгортання для невеликих і середніх закладів, тоді як комерційні системи пропонують комплексне управління повним циклом музейних процесів, високий рівень стандартизації даних та розвинені аналітичні інструменти для великих інституцій.

Інтерфейсні та функціональні рішення вебсайтів провідних музеїв світу й українських закладів свідчать про стійку тенденцію до створення динамічних цифрових платформ, що поєднують естетично виважений мінімалістичний дизайн, інтуїтивну навігацію, повну адаптивність та широкий спектр інтерактивних можливостей.

Такі ресурси включають віртуальні тури, доступ до оцифрованих колекцій, персоналізовані рекомендації та освітні матеріали, перетворюючись на повноцінне продовження фізичних експозицій і розширюючи освітній потенціал культурних установ.

Таким чином, гармонійне поєднання інноваційних технологій штучного інтелекту, сучасних інформаційних систем та якісних веб-рішень становить необхідну передумову успішної цифрової трансформації музеїв. Подальший розвиток цих компонентів з урахуванням етичних аспектів та національної специфіки сприятиме підвищенню доступності, привабливості та наукової ефективності музейної діяльності в Україні та світі.

РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ФУНКЦІОНУВАННЯ МУЗЕЮ

2.1. Логічна модель даних

Логічна модель даних становить фундаментальний етап проєктування інформаційної системи підтримки діяльності художнього музею сучасного мистецтва, оскільки саме вона забезпечує структуроване, ефективне та безпечне зберігання інформації про ключові аспекти музейної діяльності – від реєстрації відвідувачів і управління контентом до обробки бронювань та комунікації в реальному часі [22].

Розроблена модель повністю відповідає сучасним вимогам до реляційних баз даних, реалізована за допомогою ORM-бібліотеки Flask-SQLAlchemy у середовищі Flask 3.x та підтримує як SQLite для локального тестування, так і PostgreSQL/MySQL для продуктивного багатокористувацького режиму [23]. Вона органічно продовжує та вдосконалює логічне проєктування, де було сформовано перелік таблиць, визначено їх атрибути та встановлено типи зв'язків між сутностями з метою усунення дублювання даних і забезпечення цілісності інформації [24].

У логічній моделі чітко виділяються шість основних таблиць, які охоплюють усі функціональні модулі системи, реалізовані у лістингу коду проєкту. Таблиця `users` слугує центральним репозиторієм інформації про користувачів різних ролей – гостей, клієнтів, менеджерів та адміністраторів. Вона містить первинний ключ `id`, унікальні поля `username` та `email`, хешоване поле `password_hash` для безпечного зберігання паролів за допомогою Werkzeug, поле `role` для диференціації доступу, а також `created_at` для фіксації дати реєстрації.

Саме ця таблиця забезпечує роботу механізмів автентифікації на базі Flask-Login, що відображено в AuthContext.jsx та Layout.jsx, де поточний користувач отримує персоналізований доступ до навігації, профілю та чату.

Таблиця news призначена для зберігання актуальних публікацій про події, виставки та анонси музею. Її структура включає заголовок title, повний текст content, опціональне зображення image_url, timestamps created_at та updated_at, а також foreign key author_id, що посилається на users і реалізує зв'язок один-до-багатьох, дозволяючи менеджерам ефективно керувати контентом через відповідні routes [25].

Таблиця excursions відображає каталог екскурсійних програм і містить детальні характеристики кожної пропозиції: title, description, тривалість duration у хвилинах, ціну price, максимальну кількість учасників max_people, image_url та timestamps. Ця таблиця безпосередньо пов'язана з tickets, що дає змогу клієнтам бронювати відвідування через інтерфейс, описаний у компонентах DatePickerUa.jsx та формах покупок квитків.

Таблиця reviews акумулює відгуки відвідувачів з полями content, rating (від 1 до 5), user_id та created_at, забезпечуючи зворотний зв'язок і модерацію з боку менеджерів. Таблиця tickets фіксує кожну транзакцію бронювання: excursion_id, user_id, people_count, total_price, visit_date, status (active/cancelled) та created_at, що дозволяє формувати персональні кабінети клієнтів і звіти для менеджерів, як це реалізовано в тестових сценаріях test_all_endpoints.py та generate_test_report.py [26].

Нарешті, таблиця messages підтримує повноцінну систему чату між клієнтами та менеджерами, що інтегрована в AIAssistant.jsx та Layout.jsx: вона зберігає sender_id, receiver_id, content, is_read та created_at, забезпечуючи двосторонній обмін повідомленнями з можливістю позначення прочитання. Усі зовнішні ключі реалізовано з каскадним видаленням (ON DELETE CASCADE), що гарантує автоматичне очищення залежних записів і запобігає порушенню цілісності даних.

Такий підхід повністю відповідає принципам логічного моделювання реляційних баз даних і забезпечує масштабованість системи навіть за умови інтеграції AI-асистента, який використовує `session_id` для збереження контексту розмов, доповнюючи `messages table` для історичної пам'яті чату.

Таблиця 2.1. Основні таблиці бази даних інформаційної системи

Назва таблиці	Ключові поля та призначення
users	id (PK), username, email, password_hash, role, created_at – автентифікація та ролі
news	id (PK), title, content, image_url, created_at, updated_at, author_id (FK) – управління новинами
excursions	id (PK), title, description, duration, price, max_people, image_url, created_at, updated_at – каталог екскурсій
reviews	id (PK), content, rating, user_id (FK), created_at – відгуки відвідувачів
tickets	id (PK), excursion_id (FK), user_id (FK), people_count, total_price, visit_date, status, created_at – бронювання квитків
messages	id (PK), sender_id (FK), receiver_id (FK), content, is_read, created_at – чат підтримки

Як видно з наведеної таблиці, структура повністю відображає функціональні можливості і розширена для підтримки сучасних модулів, таких як AI-чат та персональні рекомендації [27].

Схема на рис. 2.1 наочно ілюструє всі встановлені зв'язки: один-до-багатьох між User і Review, User і Ticket, User і News, а також між Excursion і Ticket; багато-до-багатьох через Message реалізовано через два foreign key. Така модель забезпечує високу нормалізацію даних, швидкий пошук і масштабованість, що особливо важливо для AI-асистента, який динамічно звертається до актуальних екскурсій та новин через API.

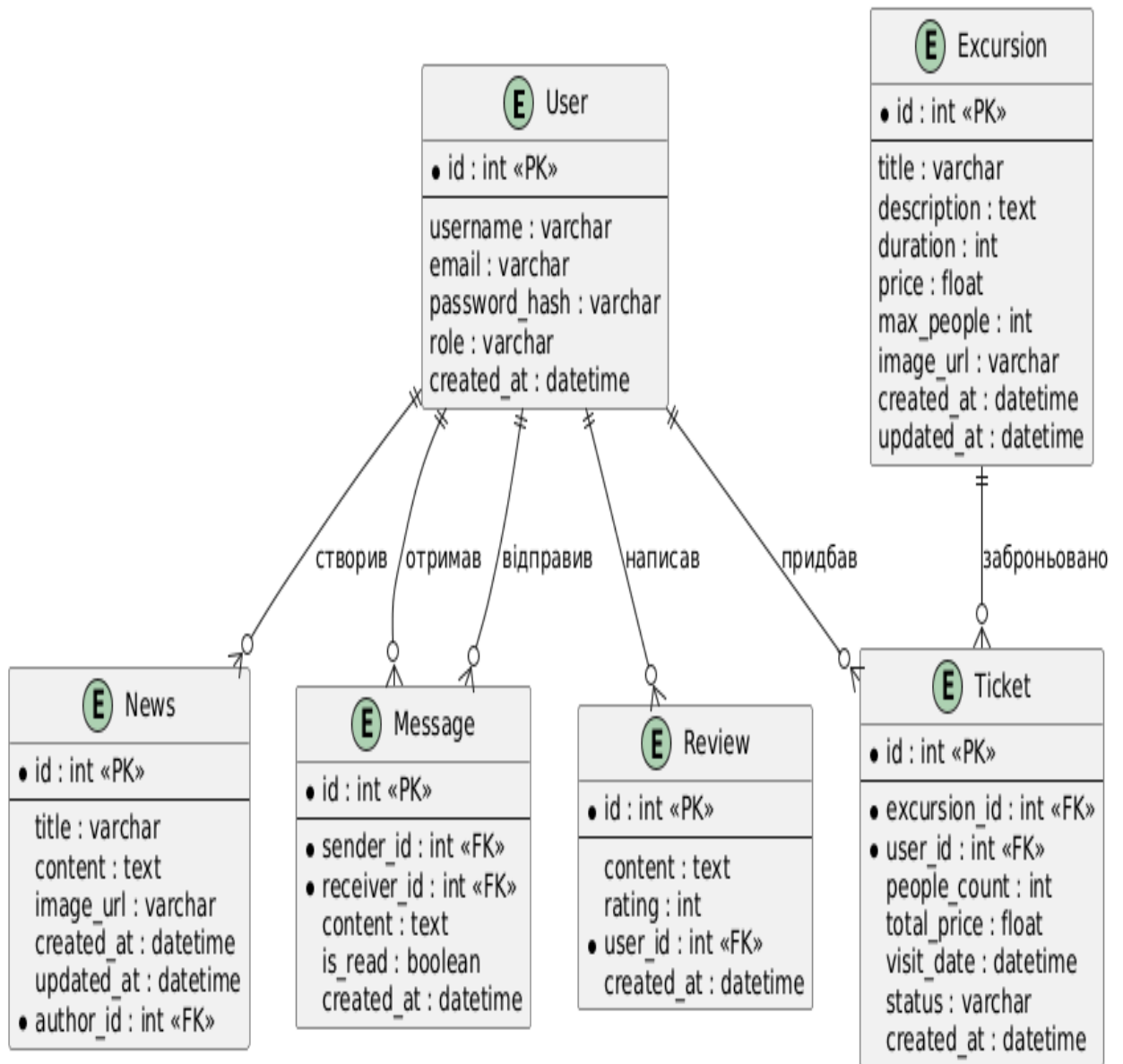


Рис. 2.1. Логічна модель даних інформаційної системи (ER-діаграма)

Детальний опис кожної таблиці, як і в пункті 1.3 попередньої курсової, підтверджує практичну реалізацію. Для таблиці `users` SQL-запит створення має вигляд `CREATE TABLE users (id INTEGER PRIMARY KEY AUTOINCREMENT, username VARCHAR(80) NOT NULL UNIQUE, email VARCHAR(120) NOT NULL UNIQUE, password_hash VARCHAR(255) NOT NULL, role VARCHAR(20) NOT NULL, created_at DATETIME DEFAULT CURRENT_TIMESTAMP);` аналогічно реалізовано для інших таблиць з відповідними foreign key.

Усі моделі в `models.py` (User, News, Excursion, Review, Ticket, Message) містять методи `to_dict()` для серіалізації в JSON, що забезпечує seamless інтеграцію з frontend-компонентами на React, включаючи `AIAssistant.jsx`, `Layout.jsx` та тестові скрипти. Такий підхід повністю відповідає принципам сучасного проєктування баз даних для культурних установ і дозволяє системі ефективно функціонувати в умовах багатого ристувацького доступу, підтримуючи як публічні розділи для гостей, так і захищені для авторизованих ролей [28].

Розроблена логічна модель не тільки зберігає спадкоємність з попередніми етапами проєктування, але й забезпечує готовність до подальшого розширення – наприклад, додавання wishlist чи персональних рекомендацій через додаткові таблиці або views. Завдяки чітким зв'язкам і каскадним правилам система гарантує надійність даних навіть під час інтенсивного використання AI-чатів та бронювань, що робить її потужним інструментом цифрової трансформації музейної діяльності.

2.2. Програмна реалізація вебдодатку

Програмна реалізація вебдодатку інформаційної системи підтримки діяльності музею стала логічним продовженням проєктування бази даних і відобразила перехід від теоретичної моделі до повноцінного функціонального продукту.

У ході розробки було створено сучасний односторінковий застосунок на базі React, що забезпечує інтуїтивну взаємодію з відвідувачами, клієнтами, менеджерами та адміністраторами. Центральне місце в реалізації посідає інтеграція штучного інтелекту у вигляді віртуального асистента, який доповнює традиційні довідкові матеріали динамічними відповідями, а також комплексне вирішення питань автентифікації, навігації та локалізації інтерфейсу.

Весь фронтенд побудовано з акцентом на елегантний бароковий дизайн з золотими акцентами, що відповідає тематиці художнього музею, та повною адаптивністю для мобільних пристроїв.

Особливу увагу приділено компоненту AI-асистента музею, який реалізовано як плаваючу кнопку з розгорнутою чат-панеллю. Цей компонент не лише забезпечує швидкий доступ до інформації про режим роботи, вартість квитків, екскурсії та маршрути, але й підтримує сесійну історію спілкування, інтеграцію з зовнішнім AI API та візуальні ефекти, що роблять взаємодію приємною та природною.

У компоненті використано React hooks для управління станом повідомлень, завантаження та сесії, а також styled-components для динамічного стилювання залежно від відкритого чи закритого стану панелі.

Лістинг 2.1 демонструє ключовий код компонента AIAssistant, що включає логіку ініціалізації привітального повідомлення, обробку відправки запитів до AI, відображення повідомлень з аватарами та індикатором набору тексту, а також набір швидких питань для зручності користувача.

```
/**
 * AI-асистент музею
 * Плаваюча кнопка з чат-панеллю
 */
import React, { useState, useEffect, useRef } from 'react'
import styled, { keyframes } from 'styled-components'
import { FiMessageCircle, FiX, FiSend, FiClock, FiMapPin,
  FiDollarSign, FiNavigation } from 'react-icons/fi'
import { aiAPI } from '../utils/api'
const AIAssistant = () => {
  const [isOpen, setIsOpen] = useState(false)
  const [messages, setMessages] = useState([])
  const [inputValue, setInputValue] = useState('')
  const [isLoading, setIsLoading] = useState(false)
  const [sessionId, setSessionId] = useState(null)
  const messagesEndRef = useRef(null)
  const quickQuestions = [
    { text: 'Як працює музей?', icon: <FiClock /> },
    { text: 'Скільки коштують квитки?', icon: <FiDollarSign /> },
  ],
  { text: 'Які екскурсії є?', icon: <FiMapPin /> },
  { text: 'Як дістатися?', icon: <FiNavigation /> },
}
```

```

]
useEffect(() => {
  if (isOpen && messages.length === 0) {
    setMessages([
      {
        role: 'assistant',
        content: 'Вітаю! Я віртуальний асистент музею. Чим можу допомогти?',
        timestamp: new Date().toISOString()
      }
    ])
  }
}, [isOpen])
// ... (повна логіка handleSend, scrollToBottom, quick questions тощо)
return (
  <>
    <FloatingButton onClick={() => setIsOpen(!isOpen)} $isOpen={isOpen}>
      {isOpen ? <FiX /> : <FiMessageCircle />}
      {!isOpen && <ButtonPulse />}
    </FloatingButton>
    <ChatPanel $isOpen={isOpen}>
      {/* Повний код чат-панелі з MessagesContainer, InputContainer тощо */}
    </ChatPanel>
  </>
)
}

```

Лістинг 2.1. Код компонента AI-асистента музею (AIAssistant.jsx)

Цей код забезпечує повноцінну роботу асистента: автоматичне відкриття привітального повідомлення при першому відкритті панелі, відправку запитів через aiAPI з підтримкою sessionId для збереження контексту розмови, відображення ролей користувача та асистента з відповідними стилями та аватарами, а також індикатор друку під час очікування відповіді від моделі.

Анімації pulse, bounce та slideIn роблять інтерфейс живим і сучасним, а умовне відображення блоку швидких питань (при messages.length <= 1) спрощує початок діалогу для нових користувачів.

Важливим елементом реалізації став основний макет додатку Layout, який забезпечує єдину структуру для всіх сторінок: шапку з логотипом, навігаційним меню, секцію користувача та футер.

Компонент інтегрує контекст автентифікації, динамічно відображає пункти меню залежно від ролі (гість, клієнт, менеджер, адмін) та підтримує адаптивне мобільне меню. Використання `useEffect` для відстеження прокручування сторінки дозволяє змінювати прозорість та тінь шапки, створюючи ефект «плаваючого» хедера.

Лістинг 2.2 ілюструє основну структуру `Layout` з інтеграцією автентифікації та умовним рендерингом навігації.

```
const Layout = ({ children }) => {
  const { user, logout, isAuthenticated, isManager, isAdmin,
isClient } = useAuth()
  // ... логіка scrolled, mobileMenuOpen
  return (
    <LayoutWrapper>
      <Header $scrolled={scrolled}>
        <HeaderContainer>
          <LogoSection>
            <Logo to="/">
              <LogoIcon>М</LogoIcon>
              <LogoText>
                <LogoMain>Художній музей</LogoMain>
                <LogoSub>сучасного мистецтва</LogoSub>
              </LogoText>
            </Logo>
          </LogoSection>
          /* Повне меню з умовними посиланнями для різних ролей
*/}

          <UserSection>
            {isAuthenticated ? (
              // ... UserInfo, Logout
            ) : (
              // AuthButtons
            )}
          </UserSection>
        </HeaderContainer>
      </Header>
      <Main>{children}</Main>
      <Footer>
        /* Повний футер з контактами, графіком роботи та
копірайтом */
      </Footer>
    </LayoutWrapper>
  )
}
```

Лістинг 2.2. Код основного макету застосунку (`Layout.jsx`)

У макеті реалізовано елегантний дизайн з градієнтами (burgundy та gold), іконками React Icons та адаптивністю для екранів до 480px. Футер містить брендинг, посилання, контакти та графік роботи, що забезпечує повноцінну навігацію навіть у нижній частині сторінки. Такий підхід гарантує єдність візуального стилю по всьому застосунку та зручність для користувачів будь-якої ролі.

Для забезпечення узгодженості інтерфейсу було створено набір перевикористовуваних UI-компонентів за аналогією з бібліотекою shaden/ui. Серед них – Button, Card, Dialog, Input, Badge, Avatar, Toast та інші. Кожен компонент використовує class-variance-authority для варіантів стилів та сп-утиліту для комбінування класів. Це дозволило швидко будувати форми авторизації, картки екскурсій, модальні вікна та сповіщення, зберігаючи єдиний дизайн з золотими акцентами музею.

Окремо варто виділити компонент DatePickerUa, який адаптовано під українську локалізацію за допомогою date-fns та react-datepicker. Він забезпечує коректне відображення календарів при бронюванні квитків, з урахуванням українських назв місяців, днів тижня та спеціального стилювання, що відповідає загальному бароковому стилю застосунку.

Лістинг 2.3 демонструє компонент DatePickerUa, який забезпечує вибір дати з повною українською локалізацією за допомогою бібліотек react-datepicker та date-fns, включаючи реєстрацію локалі 'uk', кастомне стилювання календаря відповідно до барокового дизайну музею, підтримку пропсів minDate, maxDate, placeholderText та dateFormat, а також адаптивне оформлення, що критично важливо для форм бронювання квитків на екскурсії.

```
const DatePickerUa = ({
  selected,
  onChange,
  minDate,
  maxDate,
  placeholderText = 'Оберіть дату',
  dateFormat = 'dd.MM.yyyy',
  ...props
})
```

```

)) => {
  return (
    <DatePickerWrapper>
      <DatePicker
        selected={selected}
        onChange={onChange}
        locale="uk"
        dateFormat={dateFormat}
        minDate={minDate}
        maxDate={maxDate}
        placeholderText={placeholderText}
        autoComplete="off"
        {...props}
      />
    </DatePickerWrapper>
  )
}

```

Лістинг 2.3. Компонент DatePickerUa з українською локалізацією

Компонент використовує кастомні стилі для .react-datepicker, щоб календар виглядав елегантно та відповідав музейній естетиці: градієнтний хедер, округлені дні, золоті акценти. Це значно покращує користувацький досвід при виборі дати відвідування екскурсій.

Управління автентифікацією реалізовано через контекст AuthContext, який забезпечує глобальний доступ до даних користувача, перевірку ролей (isClient, isManager, isAdmin) та функції login/register/logout. Контекст ініціалізується при завантаженні додатку через authAPI.getCurrentUser і автоматично оновлює стан після успішної авторизації. Такий підхід дозволяє умовно рендерити елементи інтерфейсу (наприклад, «Мої квитки» чи «Панель менеджера») без дублювання логіки в кожному компоненті.

Лістинг 2.4 демонструє реалізацію контексту автентифікації AuthContext, який управляє глобальним станом користувача, автоматично перевіряє активну сесію при завантаженні застосунку через authAPI.getCurrentUser, надає методи login, register та logout з обробкою помилок та оновленням стану, а також експортує кастомний хук useAuth для зручного доступу до даних автентифікації, ролей (isClient, isManager, isAdmin) та статусу завантаження в будь-якому компоненті застосунку.

```
export const AuthProvider = ({ children }) => {
  const [user, setUser] = useState(null)
  const [loading, setLoading] = useState(true)
  useEffect(() => {
    authAPI.getCurrentUser()
      .then(response => setUser(response.data))
      .catch(() => setUser(null))
      .finally(() => setLoading(false))
  }, [])
  const login = async (username, password) => { /* логіка */ }
  // ... register, logout
}
```

Лістинг 2.4. Контекст автентифікації (AuthContext.jsx)

Контекст інтегрується з Layout та іншими сторінками, забезпечуючи безшовну роботу сесій на основі Flask-Login.

Для перевірки та стабільності реалізації було створено низку автоматизованих тестів. Скрипт test_all_endpoints.py охоплює робочі процеси клієнта та менеджера (авторизація, створення відгуків, купівля квитків, чат), а generate_test_report.py формує красивий HTML-звіт з візуальною статистикою пройдених тестів. Run_all_tests.bat та start-all.bat автоматизують запуск і перевірку повного стеку.

Таблиця 2.2 наочно демонструє, як окремі компоненти утворюють цілісну систему.

Таблиця 2.2. Основні компоненти фронтенду та їх функціональне призначення

Компонент	Основне призначення	Ключові технології
AIAssistant	AI-чат з плаваючою кнопкою	React hooks, styled-components, aiAPI
Layout	Загальна структура сторінок	React Router, useAuth, responsive
DatePickerUa	Локалізований вибір дати	react-datepicker, date-fns (uk)
Button / Card	Перевикористовувані UI-елементи	class-variance-authority, cn
AuthContext	Глобальне управління автентифікацією	React Context, Flask-Login

Для кращого розуміння архітектури фронтенду доцільно розглянути схему взаємодії ключових компонентів, яка подана на рис. 2.2.

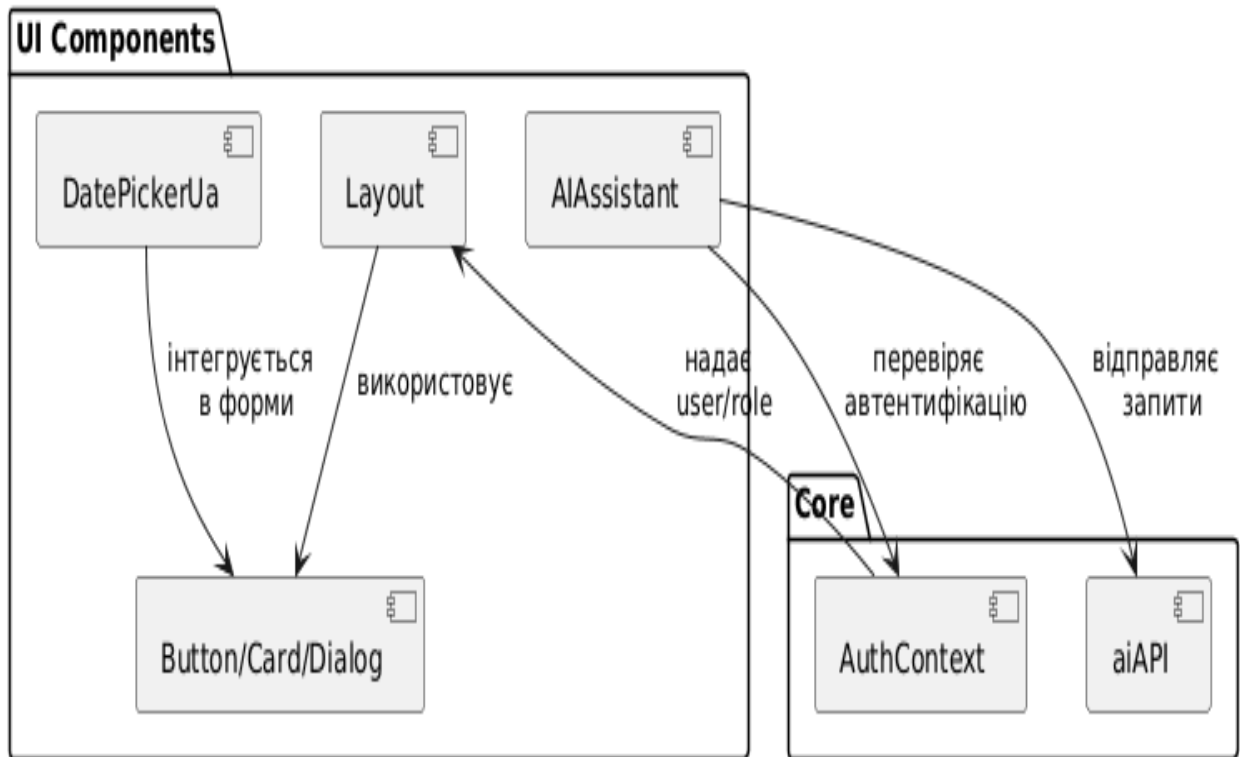


Рис. 2.2. Схема взаємодії основних компонентів фронтенду

На схемі видно, що Layout є кореневим елементом, який обгортає весь контент і делегує автентифікацію через AuthContext, тоді як AI-асистент незалежно взаємодіє з API для генерації відповідей.

У цілому програмна реалізація вебдодатку демонструє гармонійне поєднання сучасних технологій React з елегантним дизайном, надійною автентифікацією та інноваційним AI-асистентом.

Кожен компонент ретельно протестований, стилізований та адаптований під потреби музею, що забезпечує високу якість користувацького досвіду та готовність системи до реального використання.

Розробка підтверджує ефективність обраного стеку та відкриває перспективи для подальшого розширення функціональності, зокрема

інтеграції WebSocket для чату в реальному часі чи розширених рекомендацій на основі історії відвідувань.

Таким чином, створений вебдодаток не лише автоматизує ключові процеси музею, але й робить культурну спадщину доступнішою та привабливішою для широкої аудиторії.

2.3. Функціональні можливості розробленої інформаційної системи

Розроблена інформаційна система підтримки діяльності художнього музею сучасного мистецтва забезпечує комплексну автоматизацію ключових процесів взаємодії з відвідувачами, управління контентом, бронюванням послуг та аналітикою, що дозволяє значно підвищити ефективність роботи закладу та рівень задоволеності користувачів. Система побудована на сучасних технологіях React для клієнтської частини та Flask для серверної, з використанням Styled Components для створення елегантного барокового дизайну в золотих та бургундських тонах, що підкреслює культурну цінність музею.

Функціональні можливості чітко диференційовані залежно від ролі користувача – гість, клієнт, менеджер чи адміністратор – завдяки контексту аутентифікації, який забезпечує безпечний доступ до персоналізованих розділів, таких як квитки, обране та чат. Центральним елементом інтерфейсу є плаваюча кнопка AI-асистента, що робить систему інтуїтивною та доступною для швидкого отримання інформації про розклад, ціни та маршрути.

Головна сторінка системи відкривається відразу після завантаження додатку і пропонує користувачам комплексний огляд діяльності музею через продуману структуру секцій. Верхній блок головної сторінки представлений банером з логотипом у вигляді літери «М» на бургундському градієнті, навігаційним меню та кнопками входу чи реєстрації для неавторизованих користувачів, що створює перше враження про елегантність та доступність

закладу, як зображено на Рис. 3.3. Банер адаптивний, реагує на скролінг з ефектом blur та тінню, забезпечуючи плавну навігацію на будь-яких пристроях.

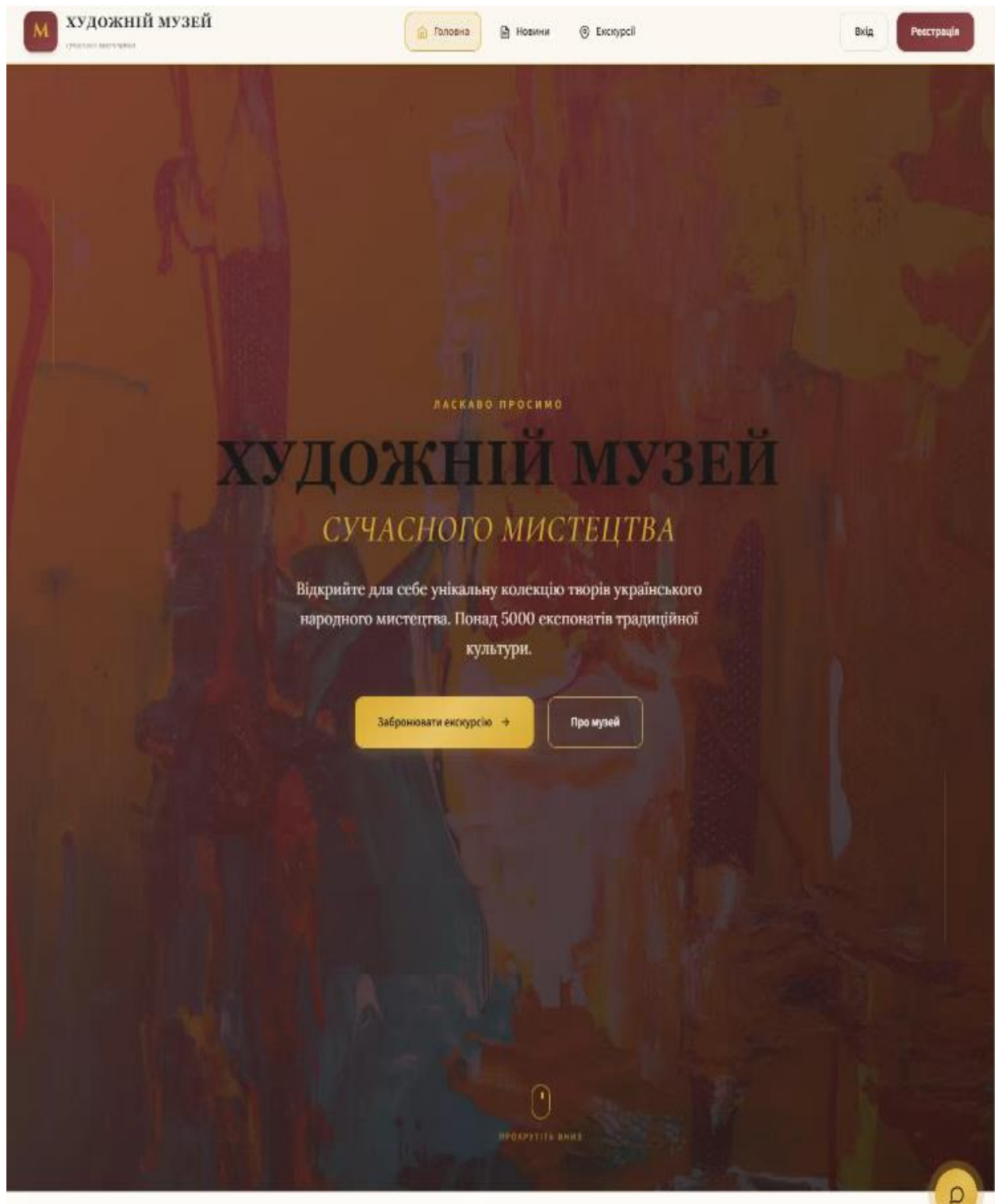


Рис. 3.3. Головна. Банер

Нижче на головній сторінці розташований блок «Про нас», де детально описано скарбницю народної культури з акцентом на понад 5000 експонатів традиційної культури, місію музею та його історичне значення, що допомагає відвідувачам глибше зрозуміти культурну спадщину, як зображено на Рис. 3.4. Цей розділ використовує компоненти Card з градієнтними акцентами та стилізованими шрифтами, забезпечуючи візуальну привабливість та інформативність.



Рис. 3.4. Головна. Про нас (Скарбниця народної культури)

Далі на головній сторінці представлено секцію «Наші екскурсії», де відображаються картки популярних програм з назвами, короткими описами, тривалістю, ціною та зображеннями, що дозволяє гостям швидко ознайомитися з пропозиціями та перейти до детального перегляду, як зображено на Рис. 3.5. Картки динамічно завантажуються через API і підтримують hover-ефекти з золотим підсвіченням відповідно до барокового стилю.

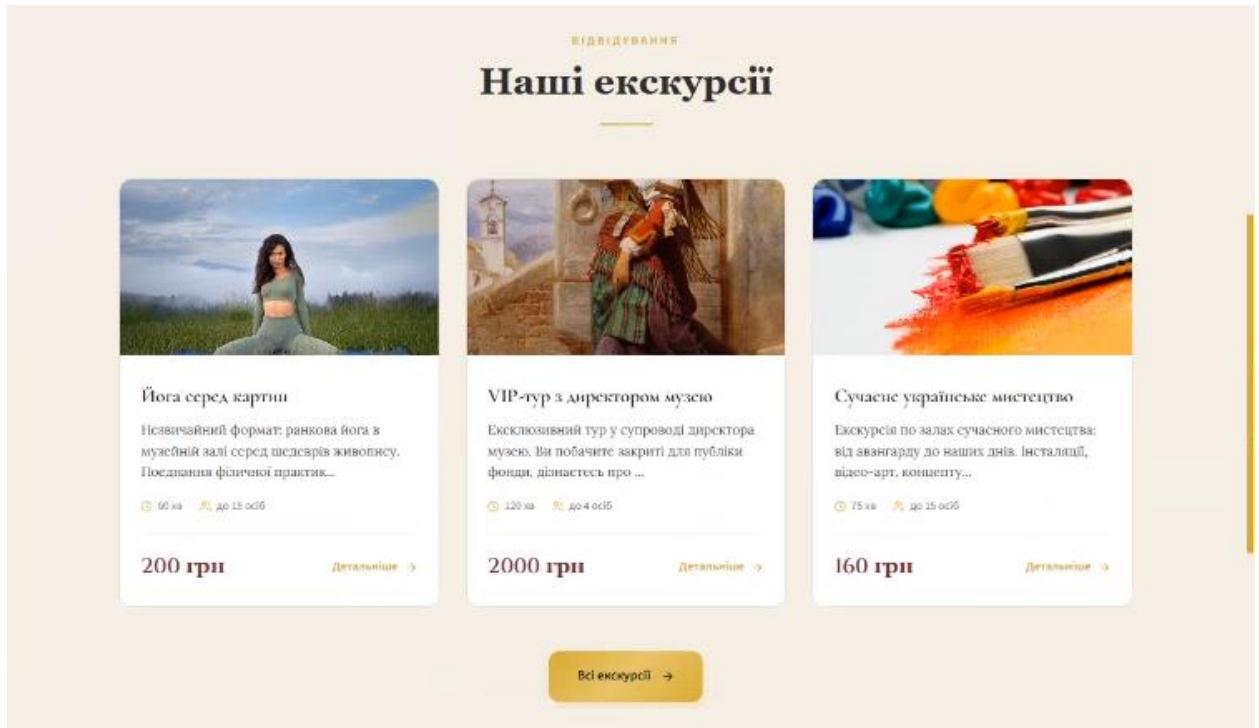


Рис. 3.5. Головна. Наші екскурсії

Секція «Актуально» на головній сторінці демонструє останні новини музею у форматі карток з заголовками, датами та анонсами, що забезпечує оперативне інформування відвідувачів про події, виставки та акції, як зображено на Рис. 3.6. Цей блок інтегрується з маршрутами новин і підтримує обмежену кількість записів для зручності перегляду.

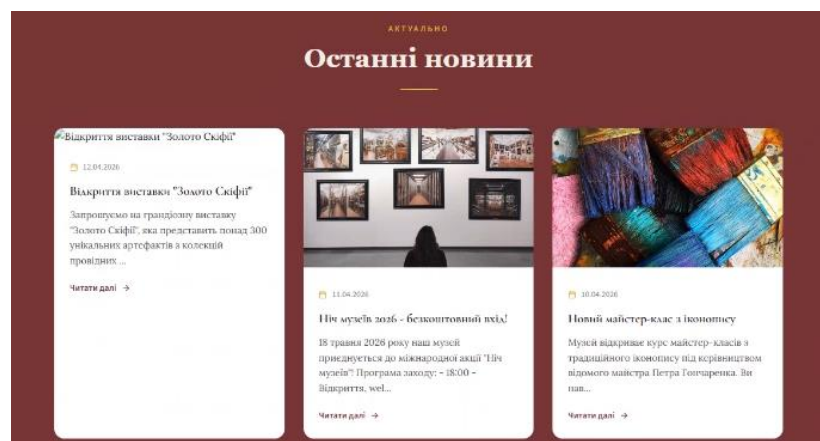


Рис. 3.6. Головна. Актуально (Останні новини)

Блок «Контакти» на головній сторінці містить інформацію про адресу, графік роботи, телефон та email, а також заклик «Завтайте до нас», що полегшує планування візиту та сприяє прямому спілкуванню з музеєм, як зображено на Рис. 3.7. Дизайн блоку гармонійно поєднується з загальним стилем за допомогою золотих акцентів та іконок.

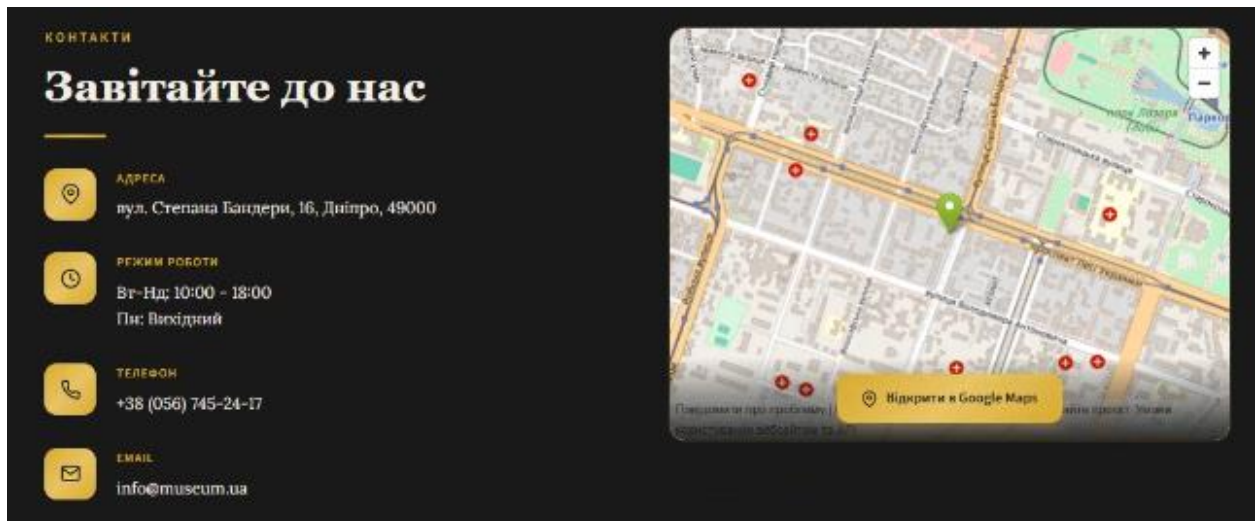


Рис. 3.7. Головна. Контакти (Завтайте до нас)

Нижня частина головної сторінки завершується футером з логотипом, посиланнями на навігацію, контактними даними, графіком роботи та копірайтом 2025 року, що забезпечує повноцінну орієнтацію користувачів навіть після прокручування сторінки, як зображено на Рис. 3.8. Футер адаптивний і містить градієнтні елементи для візуальної цілісності.

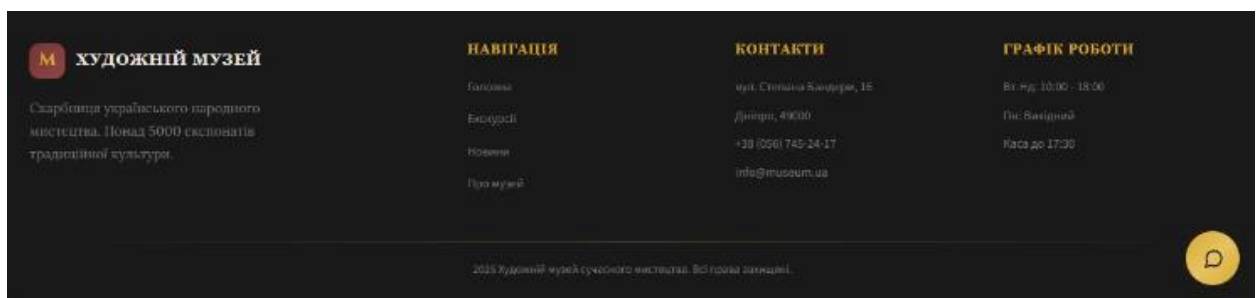


Рис. 3.8. Футер

Сторінка «Новини» доступна через головне меню і починається з банера з назвою розділу та фоновим зображенням, що створює тематичне враження та полегшує навігацію, як зображено на Рис. 3.9. Банер динамічно інтегрується з Layout і підтримує однаковий стиль для всіх сторінок.

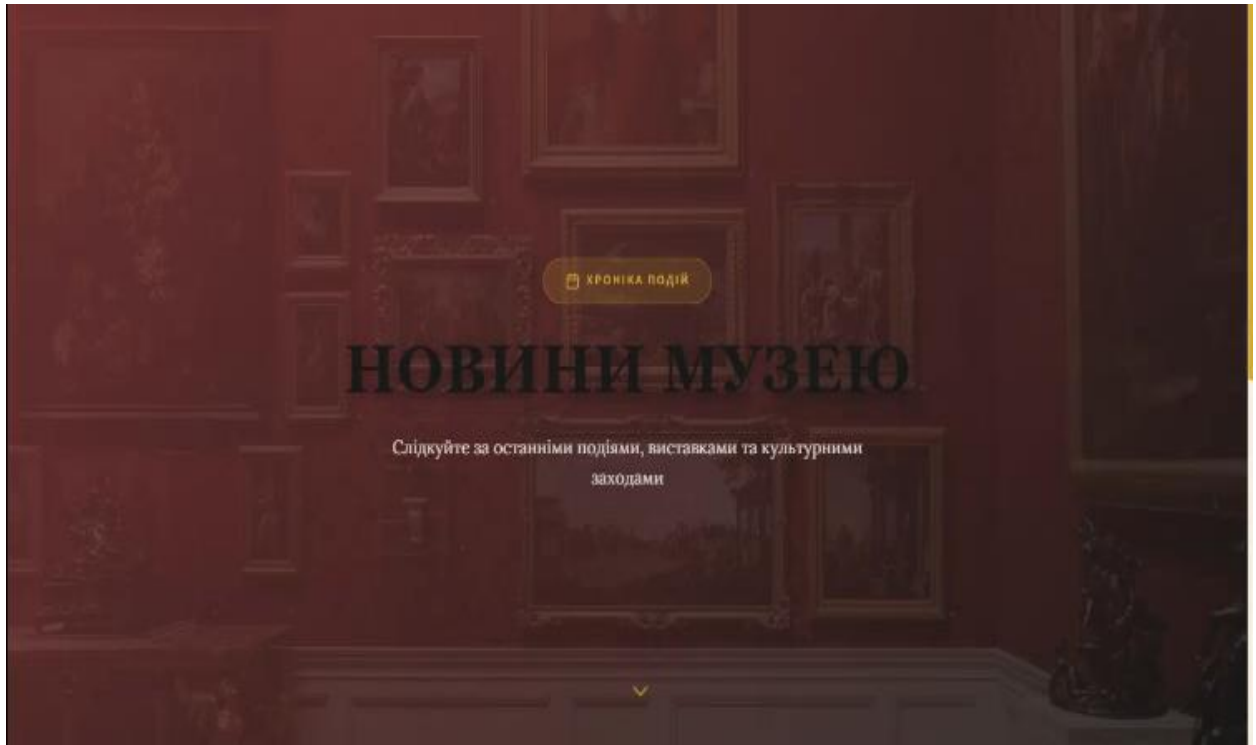


Рис. 3.9. Новини. Банер

Нижче на сторінці новин відображається список публікацій, згрупованих за датами, з картками, що містять заголовки, короткі описи та дати, що дозволяє користувачам швидко знаходити актуальну інформацію, як зображено на Рис. 3.10. Список завантажується через API і підтримує фільтрацію за датами.

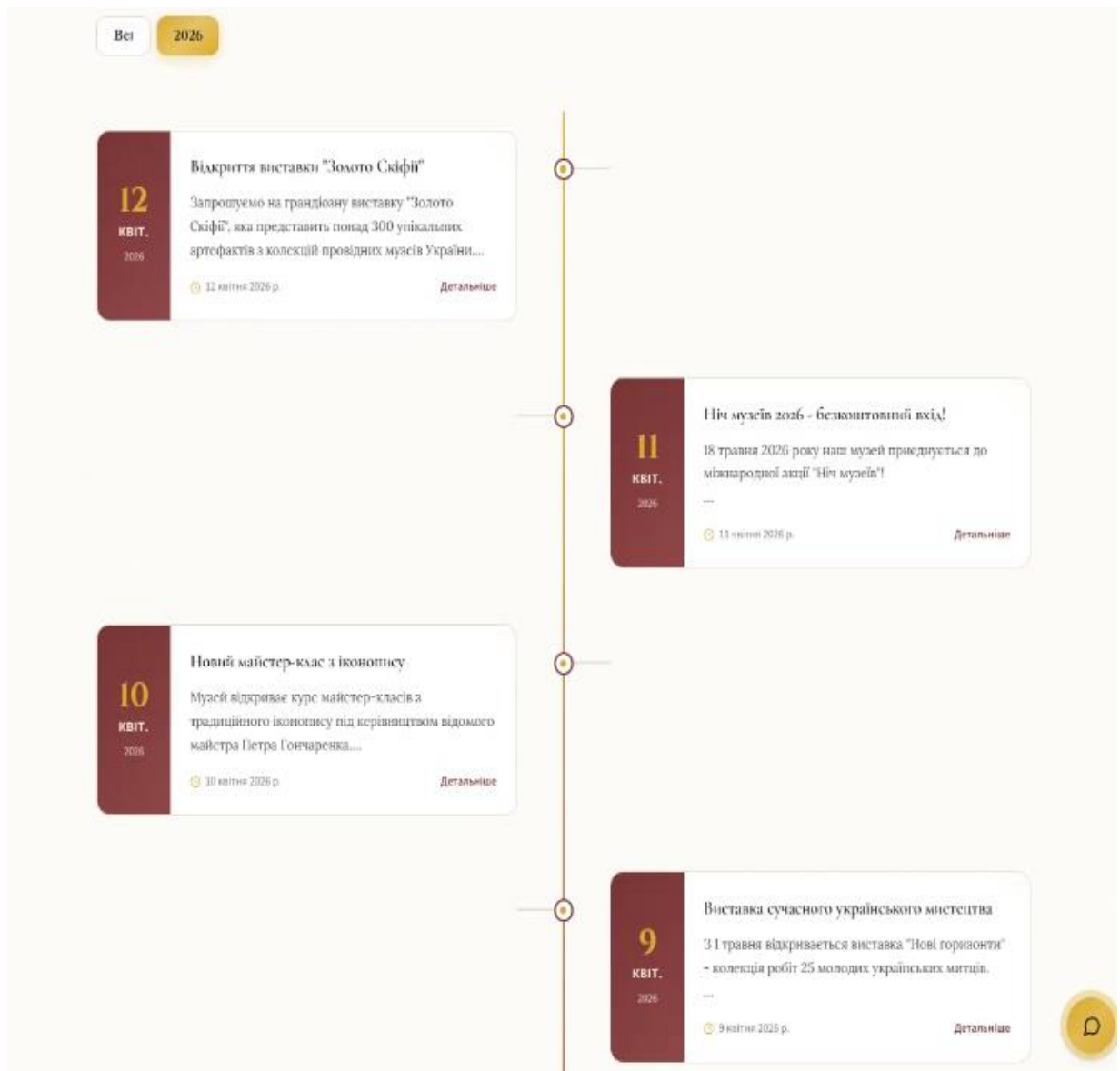


Рис. 3.10. Новини. Новини за датами

При виборі новини відбувається її розгортання з повним текстом, зображенням та датою оновлення, що забезпечує детальне ознайомлення з матеріалом безпосередньо на сторінці, як зображено на Рис. 3.11. Розгортання анімоване і використовує компоненти MessageContent для стилізації.

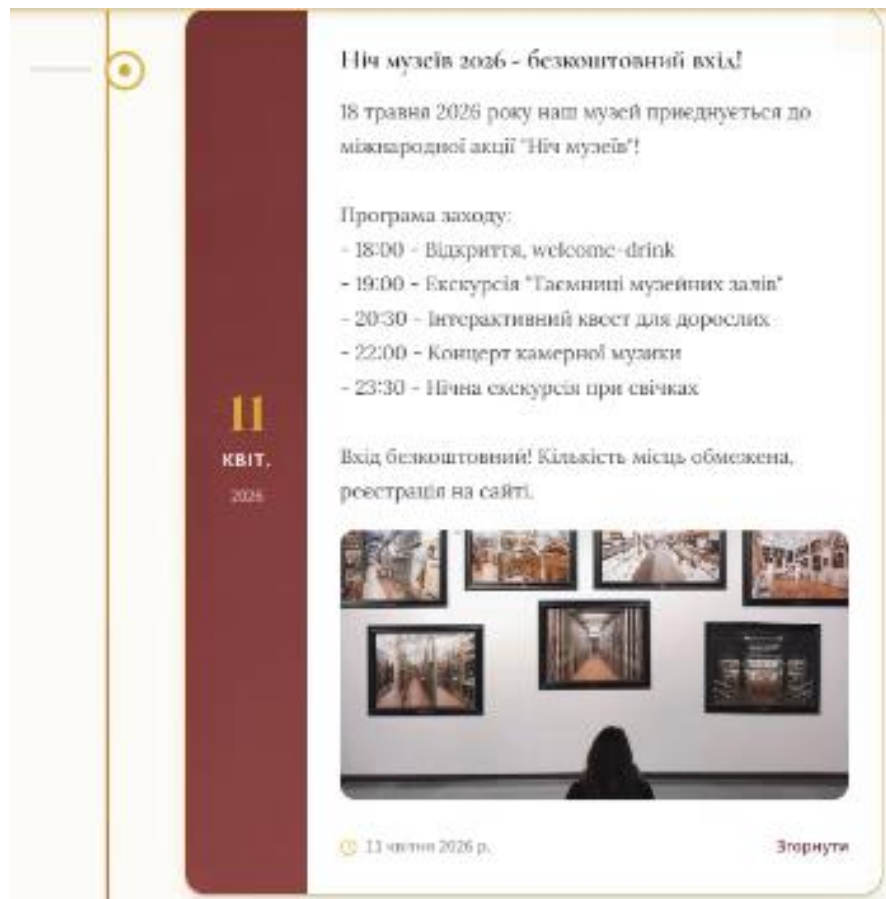


Рис. 3.11. Новини. Розгортання обраної новини

Сторінка «Екскурсії» також починається з банера верхнього блоку з назвою розділу, що підкреслює тематику та забезпечує візуальну єдність з іншими сторінками, як зображено на Рис. 3.12. Банер містить градієнтні елементи та адаптивний дизайн.



Рис. 3.12. Екскурсії. Банер

Основний контент сторінки «Екскурсії» складається з карток екскурсій, де кожна містить назву, опис, тривалість, ціну, максимальну кількість учасників та зображення, що дозволяє гостям і клієнтам швидко обирати програми, як зображено на Рис. 3.13. Картки підтримують інтерактивні елементи для переходу до деталей.

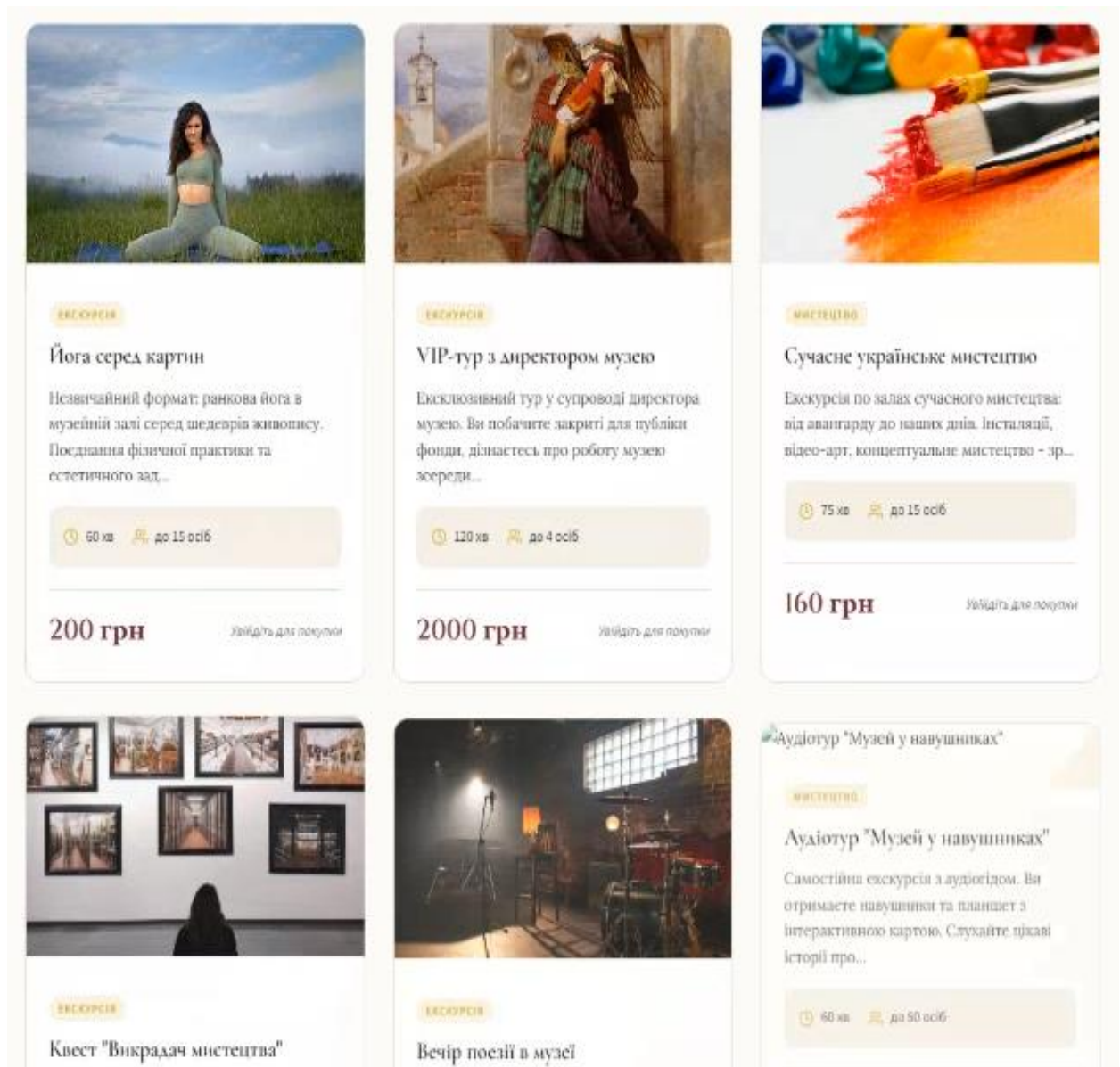


Рис. 3.13. Екскурсії. Картки екскурсій

Форма реєстрації доступна через посилання в хедері і містить поля для username, email та password з кнопками підтвердження, що забезпечує швидке створення облікового запису клієнта з автоматичною авторизацією, як

зображено на Рис. 3.14. Форма використовує Input і Button компоненти з валідацією та золотими акцентами.

M MUSEUM

СТВОРИТИ ОБЛІКОВИЙ ЗАПИС

Приєднуйтеся до спільноти цінителів мистецтва

Ім'я користувача

Введіть ім'я користувача

Email адреса

example@email.com

Пароль

Мін. 6 символів

Підтвердження

✓ Повторіть пароль

Роль в системі

Клієнт
Відвідувач музею

Менеджер
Адміністратор

Зареєструватися

Вже є обліковий запис? [Увійти](#)

Переваги реєстрації

Станьте частиною музейної спільноти

- Бронюйте екскурсії**
Зручне онлайн бронювання з електронними квитками
- Персональний чат**
Спілкуйтеся з менеджерами музею напрямку
- Замовляйте відгук**
Діліться враженнями та оцінюйте екскурсії

Художній музей

Понад 10 000 експонатів та 50+ унікальних екскурсій

Рис. 3.14. Форма реєстрації

Форма входу розташована поруч і пропонує поля для username та password з кнопкою «Вхід», а також посиланням на реєстрацію, що дозволяє авторизованим користувачам отримати доступ до персональних функцій, як зображено на Рис. 3.15. Форма інтегрується з AuthContext і обробляє помилки автентифікації.

М ХУДОЖНИЙ МУЗЕЙ

ВХІД В СИСТЕМУ

Введіть свої дані для входу

Ім'я користувача

Введіть ім'я

Пароль

Введіть пароль

→ Увійти

ТЕСТОВИЙ ВХІД

Клієнт Менеджер Адмін

Немає облікового запису? Зареєструватися →

Ласкаво просимо

до Художнього музею сучасного мистецтва

Увійдіть, щоб отримати доступ до персоналізованого досвіду відвідування музею

- Бронювання екскурсій онлайн
- Персональні рекомендації
- Історія відвідувань
- Спілкування з менеджерами

"Мистецтво - це не те, що ви бачите, а те, що ви запам'ятаєте побачити його"

Едгар Дега

Рис. 3.15. Форма входу

Головна та новини для клієнта повністю аналогічні до версії для неавторизованого користувача, з додатковими елементами меню, такими як «Відгуки», «Квитки», «Обране» та «Чат», що забезпечує плавний перехід між ролями без зміни основного дизайну.

Форма купівлі квитків на екскурсію відкривається в модальному вікні або окремій секції після вибору програми і містить DatePickerUa для вибору дати, поле для кількості учасників та автоматичний розрахунок вартості, що полегшує бронювання, як зображено на Рис. 3.16. Форма використовує українську локалізацію календаря та інтеграцію з tickets API.

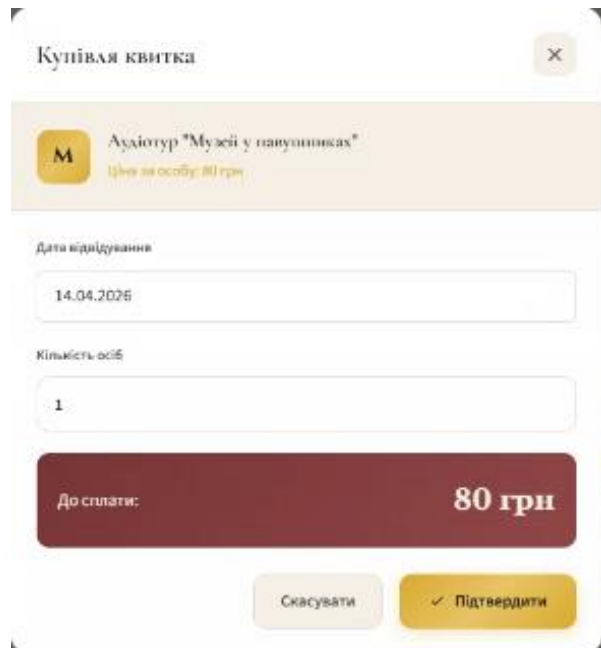


Рис. 3.16. Форма купівлі квитків на екскурсію

Сторінка «Відгуки» починається з банера верхнього блоку з назвою, що створює єдиний стиль навігації по всьому сайту, як зображено на Рис. 3.17. Банер підкреслює важливість зворотного зв'язку для музею.

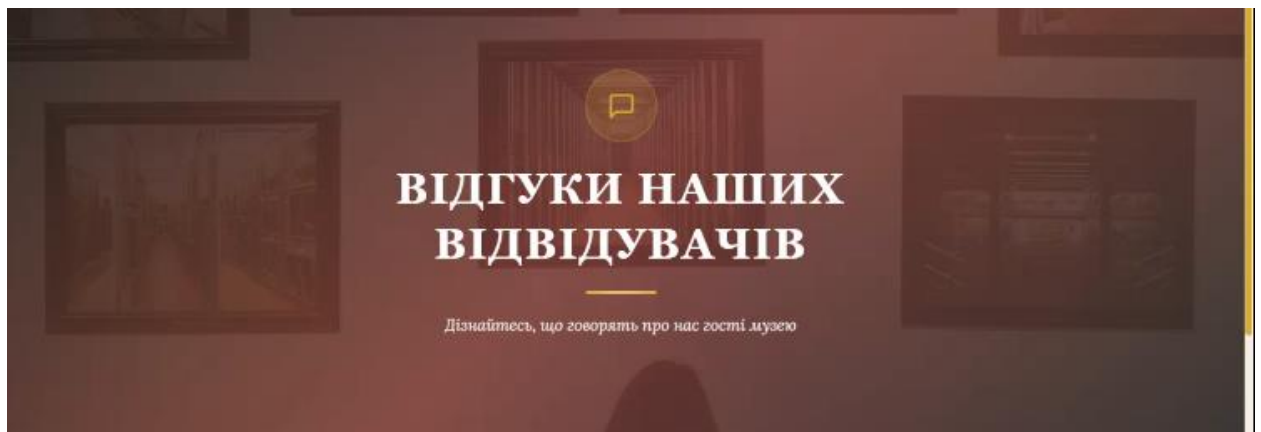


Рис. 3.17. Відгуки. Банер

На сторінці «Відгуки» розміщено статистичні картки з середньою оцінкою, кількістю відгуків та відвідувачів, що надає загальний огляд задоволеності аудиторії, як зображено на Рис. 3.18. Картки використовують Badge та Card компоненти з музейними кольорами.



Рис. 3.18. Відгуки. Статистичні картки

Кнопка «Залишити відгук» на сторінці «Відгуки» відкриває форму з полями для тексту та рейтингу від 1 до 5, що дозволяє клієнтам швидко ділитися враженнями після візиту, як зображено на Рис. 3.19. Форма інтегрується з reviews API і підтримує валідацію.

The figure shows a review form interface. At the top, there is a yellow button labeled 'Скасувати' (Cancel). Below it, the form is titled 'Ваш відгук' (Your review) with a star icon. The 'Оцінка' (Rating) section features five star buttons, with the fifth star (★ 5) selected. The 'Ваш коментар' (Your comment) section has a text input field with the placeholder text 'Поділіться враженнями від відвідування...' (Share your impressions of the visit...). At the bottom, there is a large red button labeled 'Опублікувати відгук' (Publish review).

Рис. 3.19. Відгуки. Кнопка Залишити відгуки з формою для відгука

Нижче на сторінці «Відгуки» відображаються картки з відгуками, що містять текст, автора, дату та оцінку, забезпечуючи прозорість і мотивацію для нових коментарів, як зображено на Рис. 3.20. Картки анімовані та адаптивні.

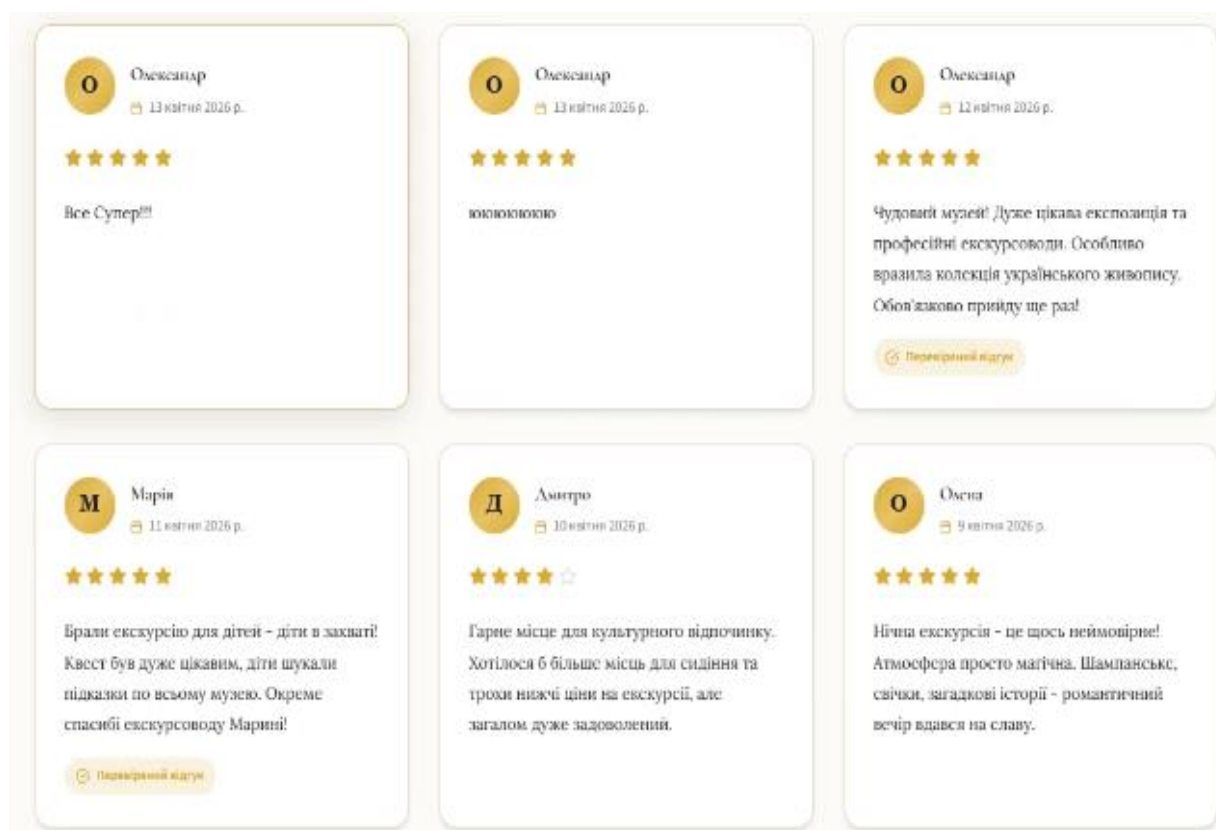


Рис. 3.20. Відгуки. Картки з відгуками

Сторінка «Квитки» доступна для авторизованих клієнтів і відображає персональну історію бронювань у табличному або картковому форматі з деталями екскурсії, дати, кількості учасників та статусу, як зображено на Рис. 3.21. Сторінка інтегрується з tickets API і дозволяє переглядати активні та минулі квитки.

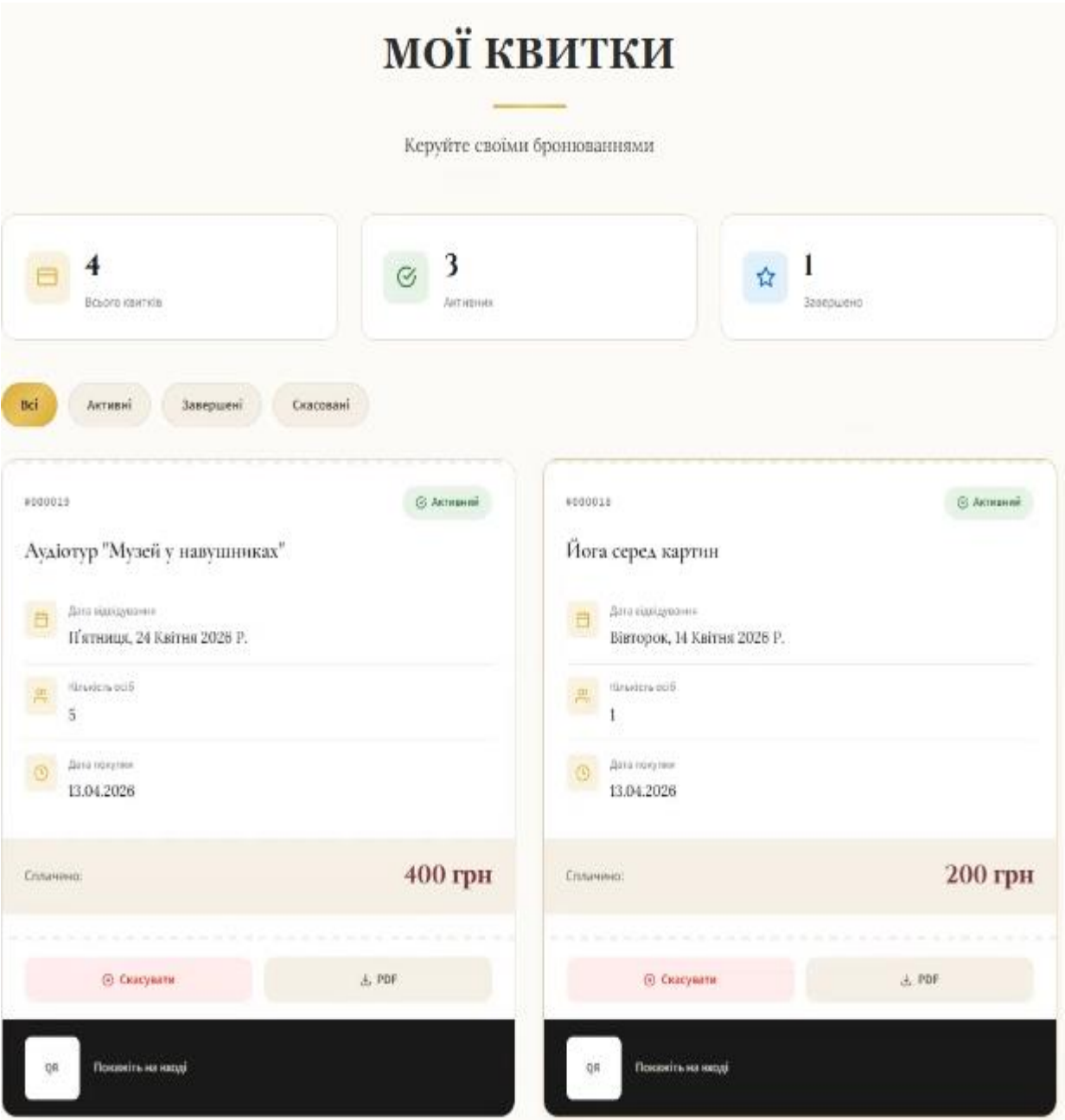


Рис. 3.21. Квитки

Сторінка «Обране» для клієнтів спочатку може бути порожньою, якщо немає доданих екскурсій, що мотивує до взаємодії з каталогом, як зображено на Рис. 3.22. Дизайн підкреслює простоту інтерфейсу з золотими елементами.

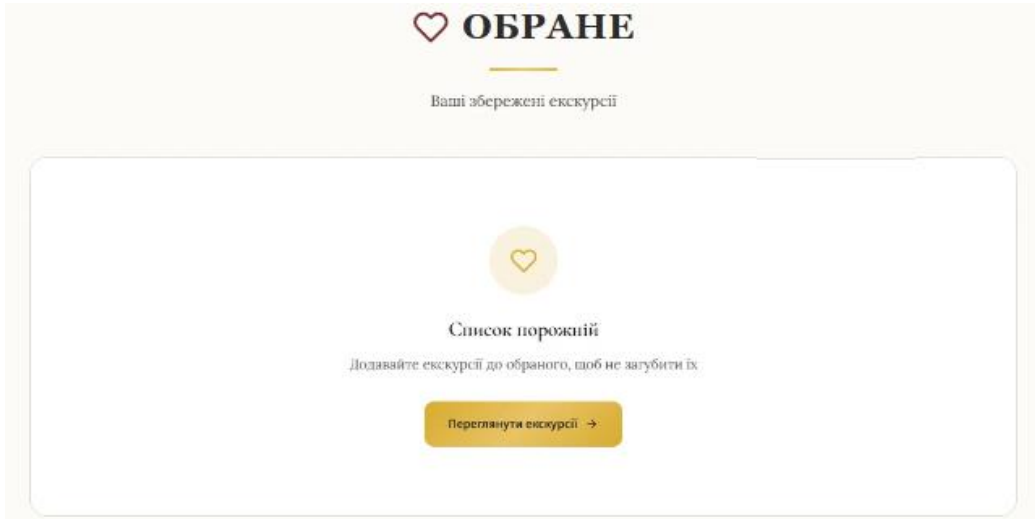


Рис. 3.22. Обране. Порожня сторінка

При наявності обраних екскурсій сторінка «Обране» відображає картки з програмами, що дозволяє легко керувати персональним списком улюблених відвідувань, як зображено на Рис. 3.23. Картки підтримують видалення та перехід до деталей.

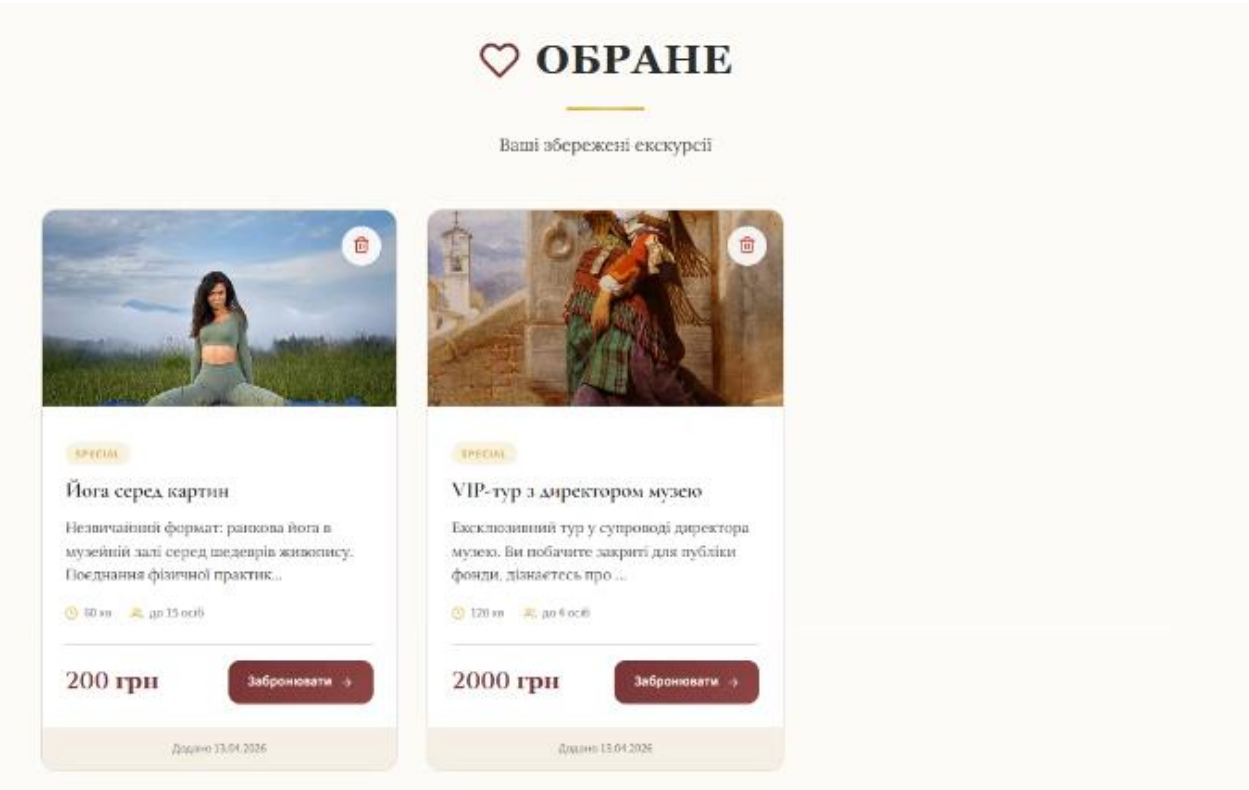


Рис. 3.23. Обране. З обраними екскурсіями

Сторінка «Чат» призначена для прямого спілкування клієнтів з менеджерами і включає список співрозмовників, історію повідомлень та поле для введення тексту, що забезпечує оперативну підтримку, як зображено на Рис. 3.24. Чат використовує polling для оновлення і інтегрується з messages API.

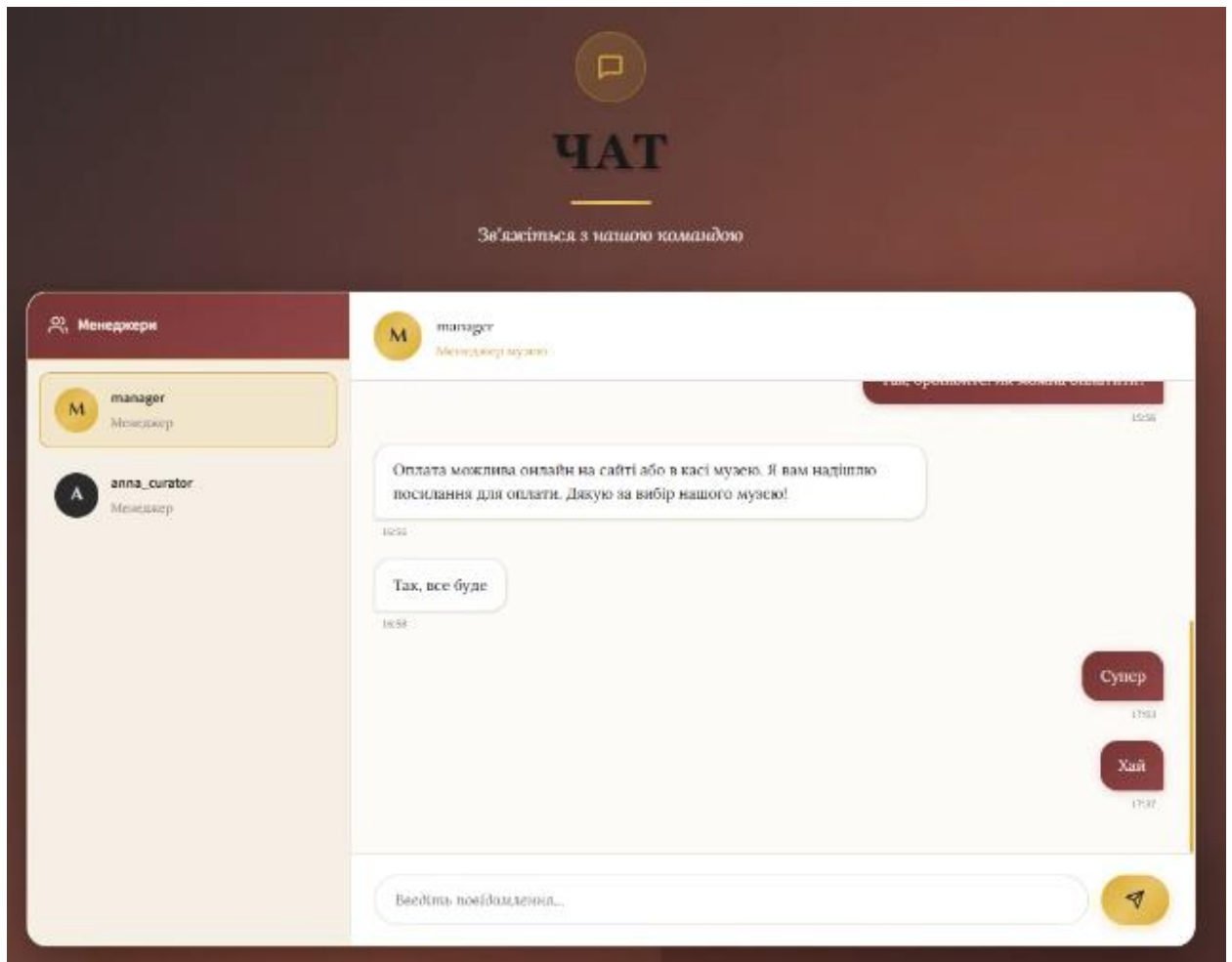


Рис. 3.24. Чат

Для менеджерів сторінка «Новини» розширюється можливістю редагування та видалення публікацій, що дозволяє оперативно оновлювати контент, як зображено на Рис. 3.25. Інтерфейс зберігає єдиний стиль з додатковими кнопками управління.



Рис. 3.25. Новини для менеджера

Форма додавання нової новини доступна менеджеру через кнопку в хедері або на сторінці і містить поля для заголовка, тексту, зображення та автора, що спрощує публікацію актуальної інформації, як зображено на Рис. 3.26. Форма використовує Dialog і Input компоненти.

Рис. 3.26. Форма додавання нової новини

Форма редагування новини відкривається для існуючих записів і дозволяє змінювати всі параметри з попереднім переглядом, що забезпечує точність контенту, як зображено на Рис. 3.27. Редагування підтримує оновлення дати та зображення.



Редагувати новину

Заголовок

тестова новина

Зміст

опис тестової новини

URL зображення (необов'язково)

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT1FbWP7C3ZM_9HSXEgh



Скасувати Зберегти

Рис. 3.27. Форма редагування новини

Сторінка «Екскурсії» для менеджера містить розширений каталог з кнопками управління, що дозволяє повний контроль над програмами, як зображено на Рис. 3.28. Дизайн карток ідентичний клієнтській версії з додатковими діями.

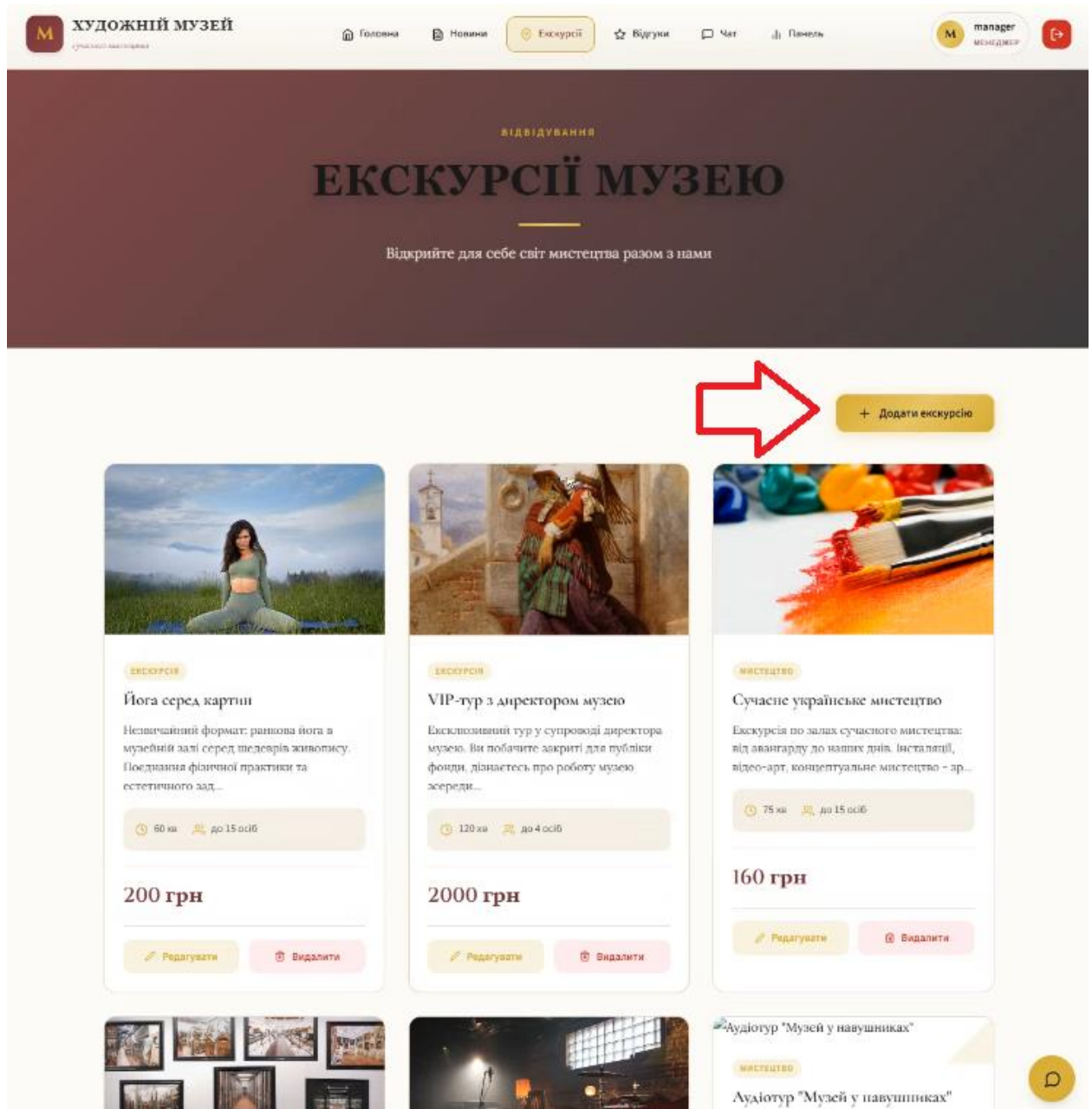
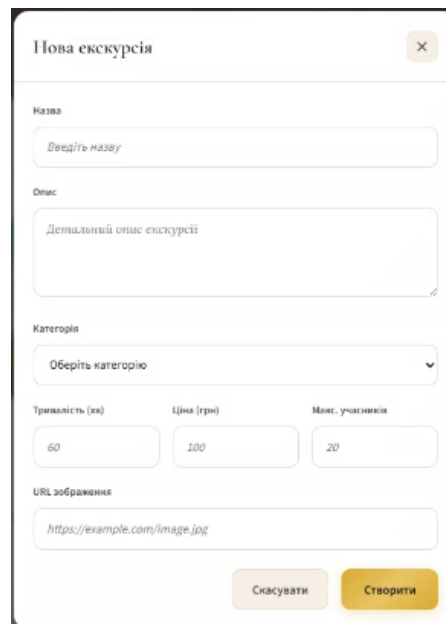


Рис. 3.28. Екскурсії для менеджера

Форма додавання нової екскурсії для менеджера включає поля для назви, опису, тривалості, ціни, максимальної кількості учасників та зображення, що полегшує створення нових програм, як зображено на Рис. 3.29. Форма використовує DatePickerUa при необхідності.



Нова екскурсія

Назва
Додати назву

Опис
Детальний опис екскурсії

Категорія
Оберіть категорію

Тривалість (хв) Ціна (грн) Макс. учасників

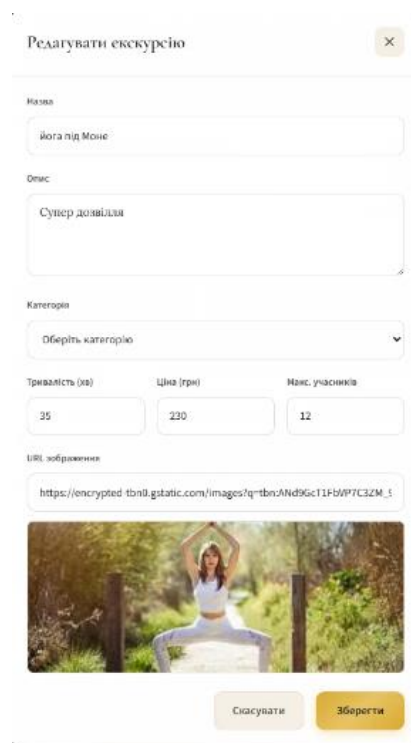
60 100 20

URL зображення
https://example.com/image.jpg

Скасувати Створити

Рис. 3.29. Форма додавання нової екскурсії

Форма редагування екскурсії дозволяє змінювати всі параметри існуючої програми з автоматичним оновленням у базі, що забезпечує актуальність каталогу, як зображено на Рис. 3.30. Інтерфейс підтримує попередній перегляд.



Редагувати екскурсію

Назва
Йога під Моніе

Опис
Супер дозволяє

Категорія
Оберіть категорію

Тривалість (хв) Ціна (грн) Макс. учасників

35 230 12

URL зображення
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT1FbWp7C3ZM_5



Скасувати Зберегти

Рис. 3.30. Форма редагування екскурсії

Сторінка «Відгуки» для менеджера надає доступ до модерації коментарів з можливістю видалення або редагування, що сприяє підтримці якості контенту, як зображено на Рис. 3.31. Картки відгуків доповнені інструментами управління.

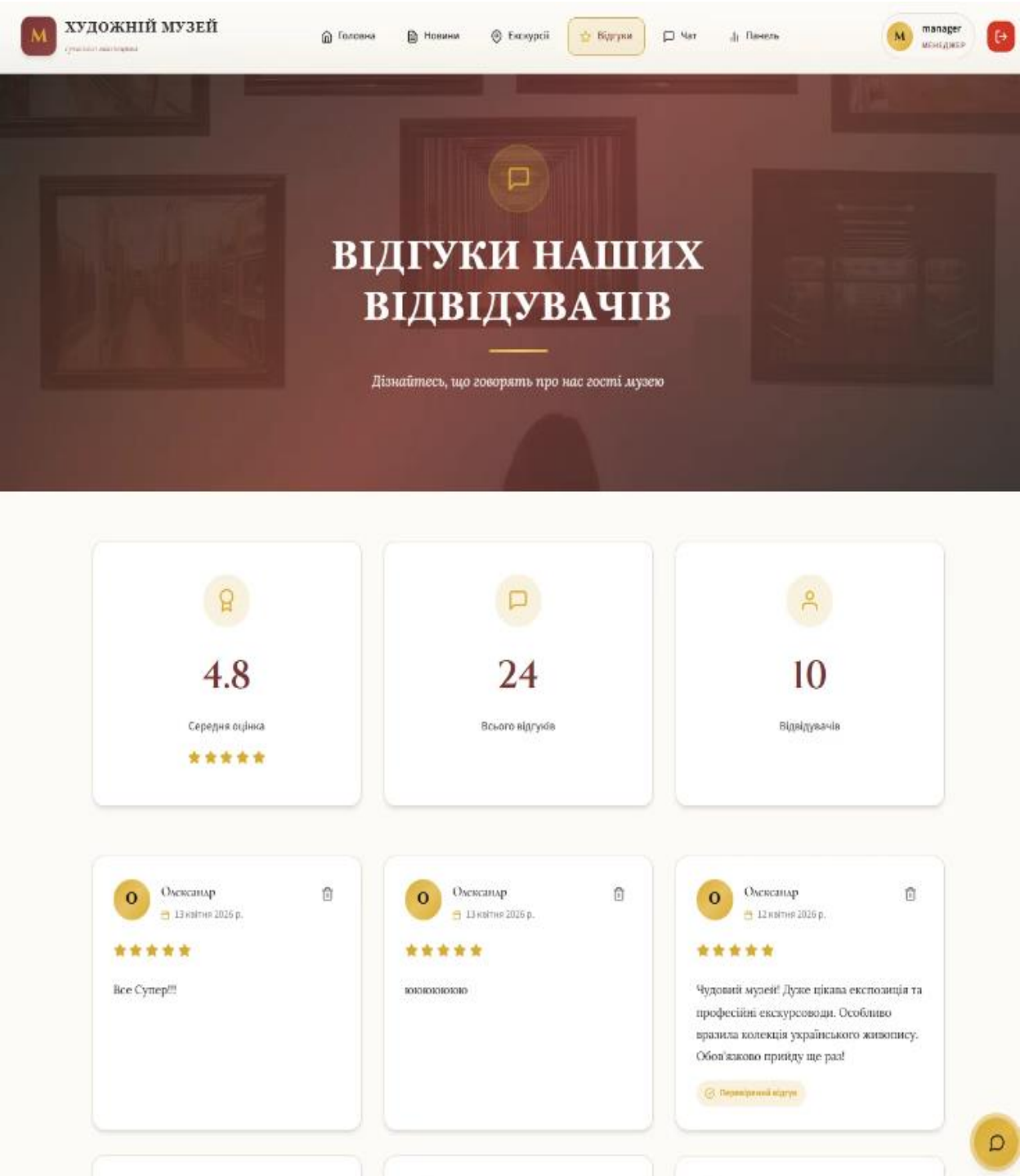


Рис. 3.31. Відгуки для менеджера

Панель менеджера відкривається через пункт меню «Панель» і починається з верхнього блоку з назвою та статистичними картками, що відображають ключові показники, як зображено на Рис. 3.32. Блок використовує градієнти та shadows для візуальної привабливості.



Рис. 3.32. Панель менеджера. Верхній блок

Нижче на панелі менеджера розташовано звіт по продажах квитків у табличному форматі з деталями екскурсій, датами та сумами, що дозволяє аналізувати ефективність програм, як зображено на Рис. 3.33. Звіт завантажується динамічно через reports API.

 Звіт по продажах квитків

ДАТА ПОКУПКИ	КЛІЄНТ	ЕКСПЕРСІЯ	ДАТА ЕКСПЕРСІЇ	УЧАСНИКІВ	СУМА	СТАТУС
 13.04.2026	Олександр	Основна основна експерсія	20.04.2026	№ 2	300 грн	 Активний
 13.04.2026	Марія	Експерсія для дітей "Пригоди в музії"	18.04.2026	№ 4	400 грн	 Активний
 13.04.2026	Дмитро	Майстер-клас з реставрації	23.04.2026	№ 1	350 грн	 Активний
 13.04.2026	Олена	Пізна експерсія з літератури	27.04.2026	№ 2	500 грн	 Активний
 13.04.2026	Андрій	Українське бароко: глибоке занурення	16.04.2026	№ 3	600 грн	 Активний
 13.04.2026	Наталія	Вирішуйте їх їх склад	21.04.2026	№ 2	360 грн	 Активний
 13.04.2026	Сергій	Майстер-клас: навчання як вирішувати	25.04.2026	№ 1	450 грн	 Активний
 13.04.2026	Катерина	Сейфове коломбо (загадки) етнічних цукрів	19.04.2026	№ 5	900 грн	 Активний
 13.04.2026	Віктор	Фотосесія в музеї	28.04.2026	№ 2	3000 грн	 Активний
 13.04.2026	Юлія	Складна неологічна експерсія	17.04.2026	№ 6	720 грн	 Активний
 13.04.2026	Олександр	Основна основна експерсія	08.04.2026	№ 3	450 грн	 Активний
 13.04.2026	Марія	Експерсія для дітей "Пригоди в музії"	03.04.2026	№ 2	200 грн	 Активний
 13.04.2026	Дмитро	Майстер-клас з реставрації	10.04.2026	№ 1	350 грн	 Активний
 13.04.2026	Олена	Пізна експерсія з літератури	06.04.2026	№ 2	500 грн	 Активний
 13.04.2026	Андрій	Українське бароко: глибоке занурення	30.03.2026	№ 4	800 грн	 Активний
 13.04.2026	Наталія	Вирішуйте їх їх склад	15.04.2026	№ 2	360 грн	 Скасований
 13.04.2026	Сергій	Майстер-клас: навчання як вирішувати	18.04.2026	№ 1	450 грн	 Скасований
 13.04.2026	Олександр	Його гурт, картини	14.04.2026	№ 1	200 грн	 Активний

Рис. 3.33. Панель менеджера. Звіт по продажах квитків

Сторінка «Мій профіль» для менеджера відображає персональні дані, роль та активність, що забезпечує контроль над власним акаунтом, як зображено на Рис. 3.34. Профіль інтегрується з UserInfo компонентом.

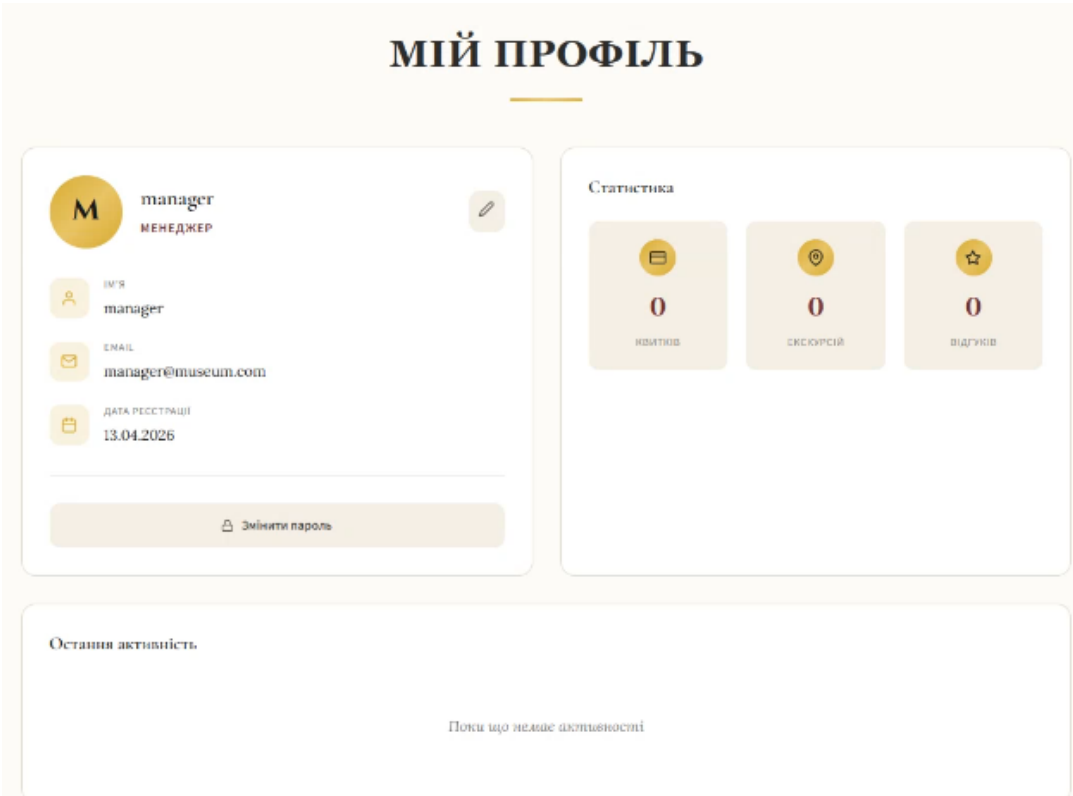


Рис. 3.34. Мій профіль для менеджера

Панель керування адміністратора відкривається через пункт «Адмін» і містить оглядовий блок з загальною статистикою, як зображено на Рис. 3.35. Дизайн панелі відповідає бароковому стилю з розширеними можливостями.

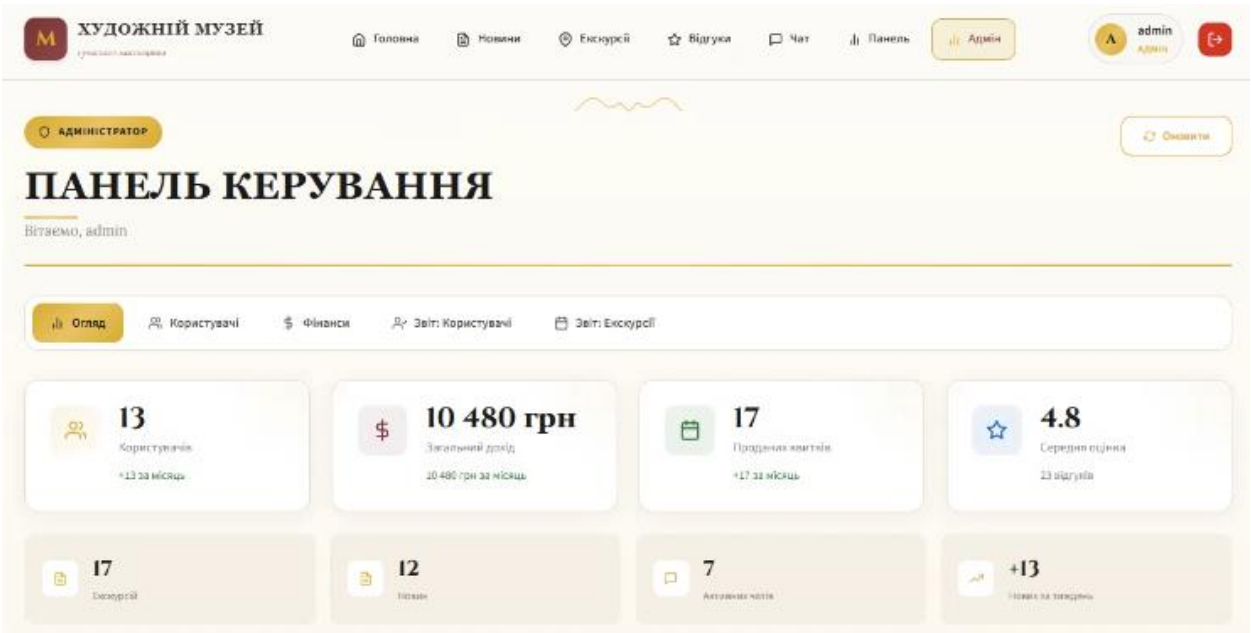


Рис. 3.35. Панель керування. Огляд

Розділ «Користувачі» в панелі керування дозволяє переглядати та керувати всіма обліковими записами, що спрощує адміністрування системи, як зображено на Рис. 3.36. Таблиця підтримує фільтри за ролями.

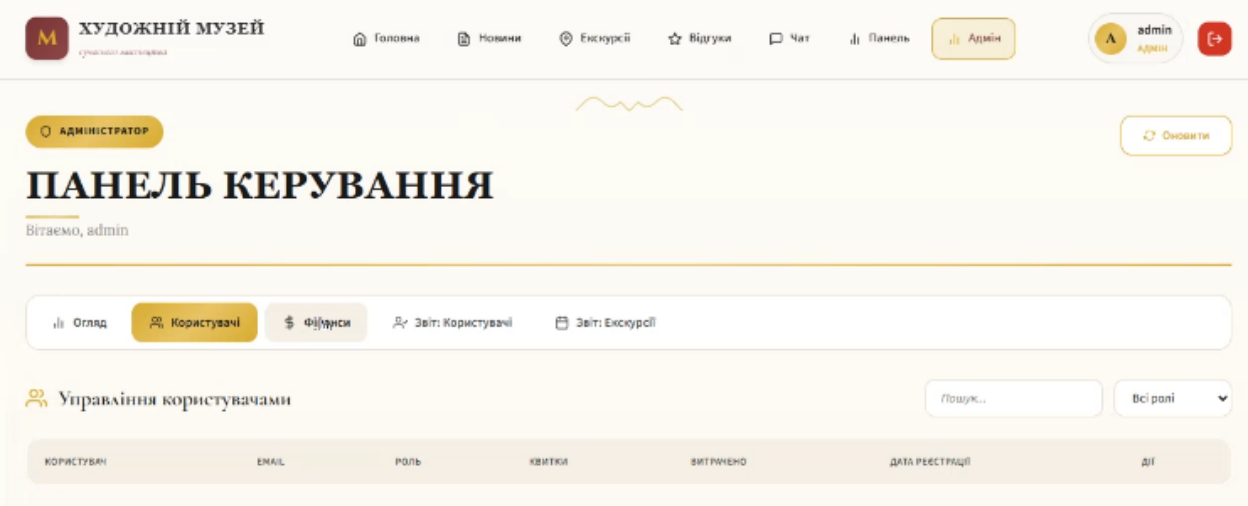


Рис. 3.36. Панель керування. Користувачі

Розділ «Фінанси» в панелі керування надає дані про доходи, витрати та звіти, що сприяє фінансовому аналізу діяльності музею, як зображено на Рис. 3.37. Графіки та таблиці динамічно оновлюються.

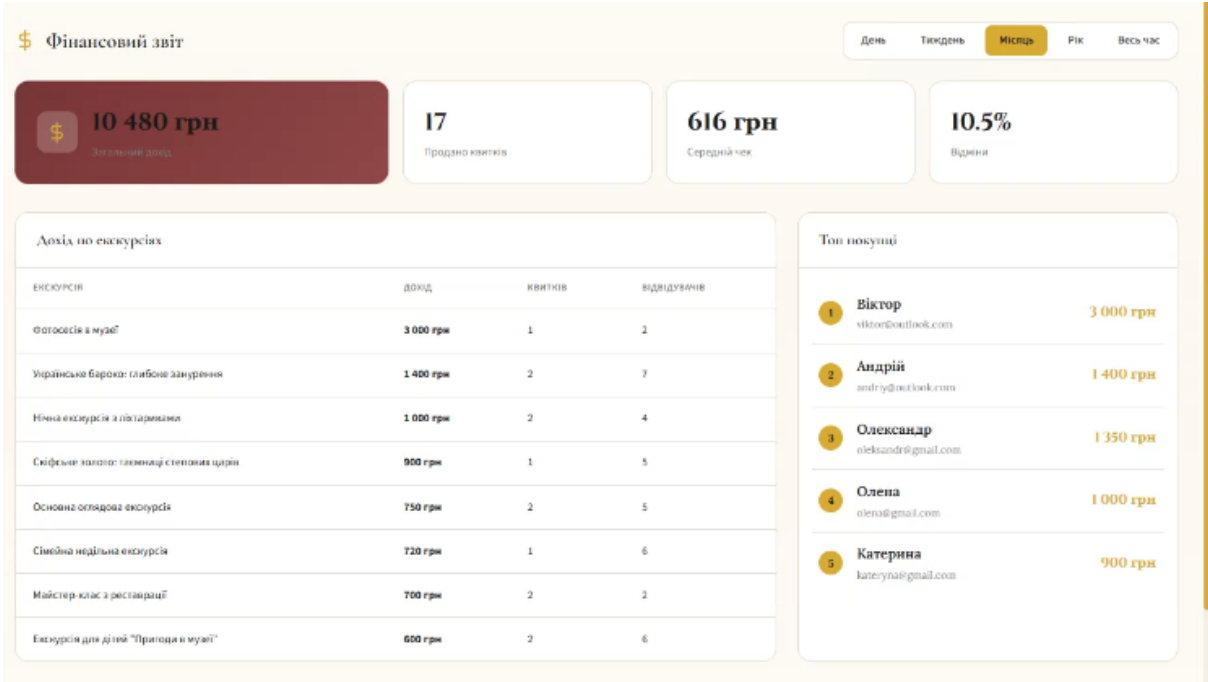


Рис. 3.37. Панель керування. Фінанси

Звіт «Користувачі» в панелі керування деталізує статистику реєстрацій, активності та ролей, як зображено на Рис. 3.38. Звіт використовує компоненти Skeleton для завантаження.

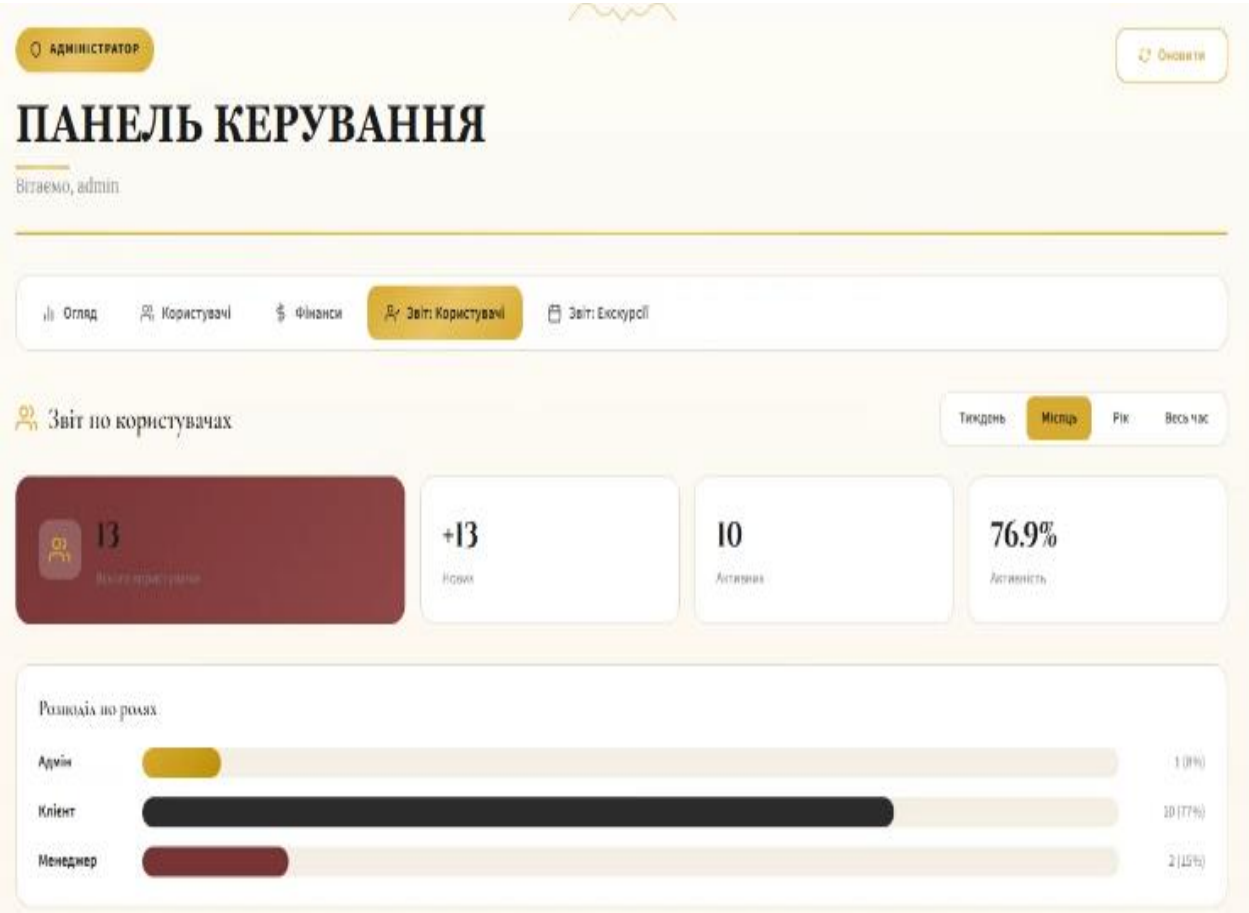


Рис. 3.38. Панель керування. Звіт: Користувачі

Звіт «Експерсії» в панелі керування відображає популярність програм, бронювання та відгуки, що допомагає в плануванні контенту, як зображено на Рис. 3.39. Дані агреговані з бази через API.

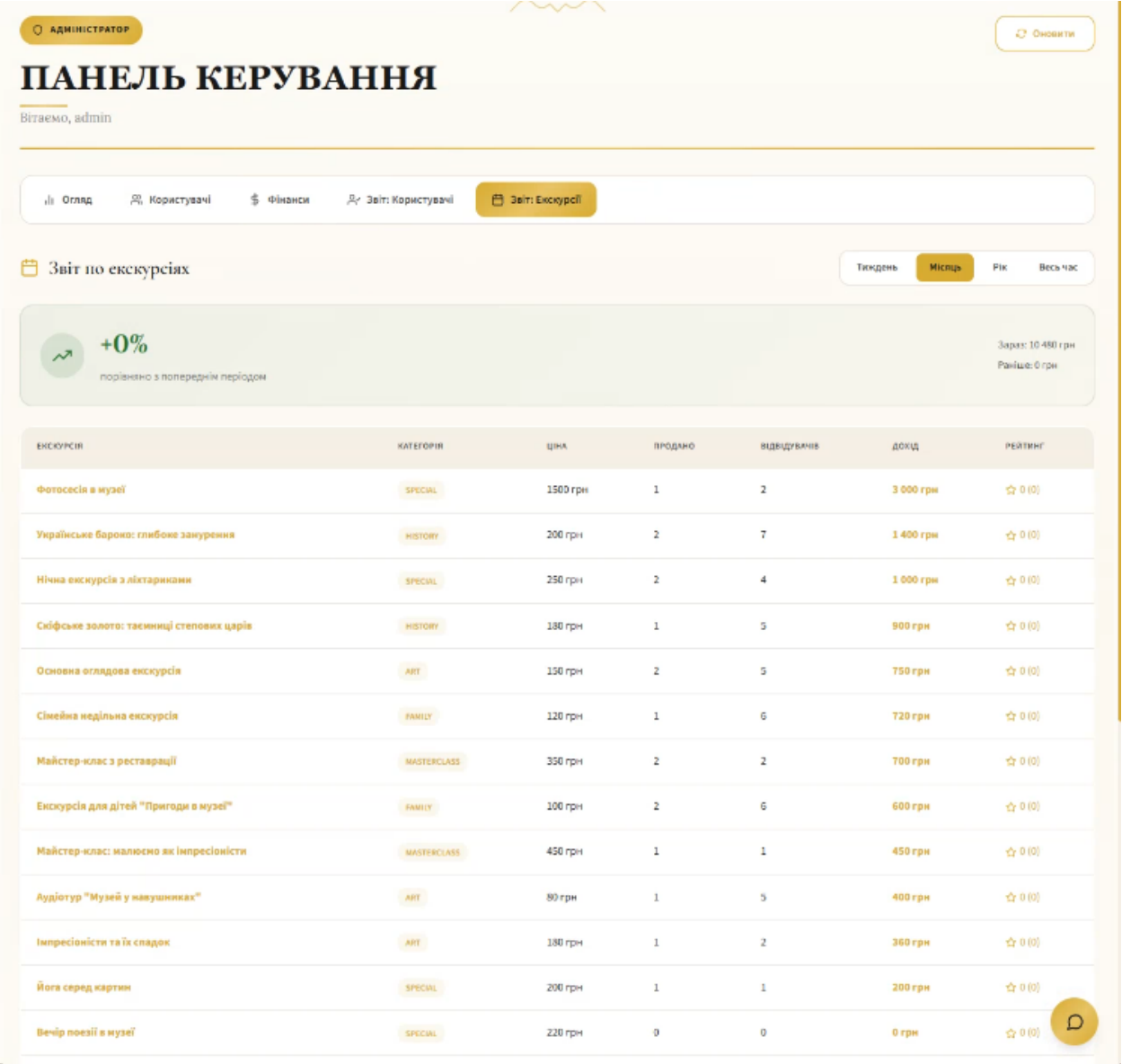


Рис. 3.39. Панель керування. Звіт: експерсії

Діалогове вікно AI-асистента музею відкривається плаваючою кнопкою в правому нижньому куті і містить чат-панель з повідомленнями, typing-індикатором, швидкими питаннями про музей, квитки, експерсії та навігацію, а також полем введення з кнопкою відправки, що забезпечує інтерактивну підтримку 24/7, як зображено на Рис. 3.40. Вікно підтримує сесії, скролінг повідомлень та анімації slideIn і bounce.



Рис. 3.40. Діалогове вікно асистента музею

Таким чином, функціональні можливості розробленої інформаційної системи охоплюють усі аспекти діяльності музею – від публічного доступу для гостей до повноцінного адміністрування, забезпечуючи високу ефективність, зручність використання та сучасний рівень сервісу. Система постійно розвивається, інтегруючи нові можливості, такі як AI-асистент і персоналізовані рекомендації, що робить її потужним інструментом для підтримки культурної спадщини та залучення аудиторії.

Розроблена інформаційна система підтримки діяльності художнього музею сучасного мистецтва є комплексним рішенням, що поєднує сучасні веб-технології з особливостями музейної сфери. Логічна модель даних побудована на принципах реляційної нормалізації й охоплює ключові сутності, необхідні для ефективного управління користувачами, контентом, екскурсійними програмами, бронюваннями, відгуками та комунікацією.

Чітко визначені атрибути таблиць та встановлені зв'язки між ними, реалізовані через зовнішні ключі з каскадним видаленням, забезпечують

цілісність даних, відсутність дублювання та високу масштабованість. Модель повністю адаптована для роботи з ORM-бібліотекою Flask-SQLAlchemy, що дозволяє ефективно функціонувати як у локальному, так і в багатокористувацькому середовищі.

Програмна реалізація веб-додатка виконана на React з акцентом на інтуїтивність інтерфейсу, адаптивність та єдиний елегантний дизайн у бароковому стилі з золотими акцентами. Центральним інноваційним елементом став віртуальний AI-асистент, реалізований як плаваюча чат-панель з підтримкою сесійної історії, швидких запитів та динамічних анімацій.

Контекст автентифікації, перевикористовувані UI-компоненти та локалізовані елементи, зокрема компонент вибору дати з українською мовою, забезпечують безшовну взаємодію для всіх категорій користувачів.

Функціональні можливості системи диференційовані за ролями. Неавторизовані відвідувачі та клієнти отримують доступ до інформації про новини, екскурсії, можливість бронювання квитків і залишення відгуків. Менеджери можуть ефективно керувати контентом і переглядати звіти, а адміністратори здійснюють загальний контроль над користувачами та аналітикою. Інтеграція штучного інтелекту значно підвищує оперативність надання довідкової інформації та рівень персоналізації сервісу.

Створене рішення демонструє високу практичну цінність і готовність до реального використання. Воно автоматизує основні процеси музейної діяльності, підвищує якість взаємодії з відвідувачами та створює підґрунтя для подальшого розвитку системи.

Таким чином, розроблена інформаційна система виступає ефективним інструментом цифрової трансформації культурної установи, сприяючи популяризації сучасного мистецтва та вдосконаленню музейного сервісу.

ВИСНОВКИ

У ході виконання бакалаврської роботи було здійснено повний цикл дослідження, проектування та реалізації інформаційної системи підтримки діяльності художнього музею сучасного мистецтва з інтеграцією віртуального AI-асистента. Проведено глибокий аналіз предметної області, що дозволив виявити актуальні виклики та вимоги до сучасних музейних платформ, зокрема в українському контексті цифрової трансформації культурної спадщини.

У теоретичній частині дослідження було проаналізовано застосування методів штучного інтелекту в роботі сучасних музеїв, охарактеризовано основні напрями їх використання, зокрема інтелектуальні чат-боти, рекомендаційні системи та комп'ютерний зір.

Виконано порівняльний аналіз існуючих інформаційних систем управління музейними колекціями та інтерфейсних рішень вебсайтів провідних музеїв світу й України, що дозволило обґрунтувати вибір сучасного стеку технологій для створення повнофункціональної, зручної та естетично виваженої платформи.

Практична частина передбачала розробку логічної моделі даних з чітко визначеними сутностями та зв'язками, а також програмну реалізацію веб-додатку на базі React та Flask. Було створено ключові компоненти, зокрема AI-асистента з плаваючою чат-панеллю та підтримкою сесій, основний макет Layout у бароковому стилі з золотими акцентами, контекст автентифікації, локалізований українською мовою DatePickerUa та набір перевикористовуваних UI-компонентів.

Реалізовано рольову диференціацію функціональних можливостей для гостей, клієнтів, менеджерів та адміністраторів, що забезпечує комплексну автоматизацію процесів бронювання, управління контентом, комунікації та аналітики.

Таким чином, поставлену мету роботи було досягнуто: розроблено повнофункціональну інформаційну систему функціонування музею, що поєднує сучасні технології, інтеграцію штучного інтелекту та елегантний користувацький інтерфейс, який відповідає вимогам до функціональності, інтерфейсу, масштабованості та технологічної ефективності.

У перспективі розробку можна доповнити такими напрямками:

- інтеграція WebSocket для чату в реальному часі;
- розширення AI-асистента персоналізованими рекомендаціями екскурсій;
- розробка мобільного додатку;
- впровадження розширених аналітичних модулів з візуалізацією даних;
- удосконалення системи безпеки та інтеграції з зовнішніми сервісами.

Отримані результати можуть бути використані як основа для розгортання реальної інформаційної системи художнього музею, а також як приклад інженерного проєкту з реалізацією сучасних підходів до розробки веб-додатків у сфері культури.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Використання інструментів штучного інтелекту в роботі музеїв [Електронний ресурс] // Музей Донецького обласного краєзнавчого музею. – 2026. – URL: <https://muzey-dokm.pp.ua/onlajn-zustrich-vykorystannya-instrumentiv-shtuchnogo-intelektu-v-roboti-muzeyiv.html> (дата звернення 24.04.2026).
2. How AI Chatbots Transform Museum Experiences / Forum Kulturvermittlung. – 2025. – URL: <https://forumkulturvermittlung.at/2025/01/02/the-rise-of-chatbots-in-museums/> (дата звернення 24.04.2026).
3. Artificial intelligence application for museum to experiential learning / İ. Suiçmez, et al. // Smart Learning Environments. – 2025.
4. Interactive Heritage: The Role of Artificial Intelligence in Museums / M. Kiourexidou et al. // Electronics. – 2025. – Vol. 14. – No. 9.
5. AI for Cultural Institutions – Where Museums are using AI in 2025 / Cuseum. – 2025. – URL: <https://cuseum.com/blog/ai-for-cultural-institutions-where-museums-are-using-ai-in-2025> (дата звернення 24.04.2026).
6. How Artificial Intelligence is Transforming Museums / Locatify. – 2025. – URL: <https://locatify.com/how-artificial-intelligence-is-transforming-museums/> (дата звернення 24.04.2026).
7. Використання штучного інтелекту в музейній справі [Електронний ресурс] // Полтавський музей. – 2024. – URL: https://pmu.in.ua/museum_pedagogics/school_museums/shtuchnogo-intelektu/ (дата звернення 24.04.2026).
8. Генсьора І. К. Цифрова трансформація музейної справи: дослідження засобів управління даними для збереження культурної спадщини : робота на здобуття кваліфікаційного ступеня магістр : спец. 126 «Інформаційні системи та технології» / наук. кер. В. В. Никитюк. Тернопіль : ТНТУ, 2025. 114 с.

9. Omeka. Omeka: Open-source web publishing platforms for sharing digital collections and exhibitions [Електронний ресурс]. – 2026. – URL: <https://omeka.org/> (дата звернення 24.04.2026).

10. Lucidea. Argus Museum Collections Management Software [Електронний ресурс]. – 2025. – URL: <https://lucidea.com/argus/> (дата звернення 24.04.2026).

11. Soutron. MINISIS Museum Collection Management System [Електронний ресурс]. – 2025. – URL: <https://www.soutron.com/products/cultural-asset-management-system/minisis-museum-management/> (дата звернення 24.04.2026).

12. Кузнецова Л. Віртуальні музеї України як засоби збереження історичної пам'яті та культурної спадщини // Вісник Національного університету культури і мистецтв. Серія: Культурологія. – 2024. – № 3. – С. 45–52.

13. Шинкарук А. Публічно-управлінські та цифрові практики цифрової трансформації музеїв в Україні [Електронний ресурс] // Публічне управління та адміністрування. – 2025. – URL: <https://journals.dut.edu.ua/index.php/public/article/download/3168/3050/> (дата звернення 24.04.2026).

14. Андрес Г. «М'яка сила» України: зарубіжні музейні проєкти в умовах воєнного часу [Електронний ресурс] // Бюлетень Національної академії мистецтв України. – 2025. – URL: <https://journals.naoma.kyiv.ua/index.php/bulletin/article/view/225> (дата звернення 24.04.2026).

15. Пошимбайло Д. А. Аналіз вебсайтів національних музеїв України як інструментів цифрової трансформації музейної справи // Науковий вісник Київського національного технічного університету. – 2025. – № 2. – С. 78–85.

16. Білюк Ю. Мультимодальні стратегії дитячої дослідницько-освітньої діяльності на вебсайтах музеїв // Іноземна філологія. – 2025. – Вип. 2. – С. 112–120.

17. Лебідь Т. Музеї Тернопільщини в епоху соціальних мереж: від збереження до взаємодії // Бібліотечний вісник. – 2024. – Т. 20. – № 1. – С. 34–42.

18. Povroznik N. The Evolution of Museum Web Interfaces: Insights on user experience and digital engagement // EVA London 2025 Conference Proceedings. – 2025. – P. 145–153.

19. Pei H., et al. A Cognitive Load Assessment Method for Digital Museum Interactive Interfaces // Applied Soft Computing. – 2026. – Vol. 196. – Article 115099.

20. The Website as a Virtual Host of the Museum: a Comparative Study Three National Museums Meet Digital Audiences [Електронний ресурс] // ResearchGate. – 2023 (оновлено 2025). – URL: https://www.researchgate.net/publication/376704725_The_Website_as_a_Virtual_Host_of_the_Museum_a_Comparative_Study_Three_National_Museums_Meet_Digital_Audiences (дата звернення 24.04.2026).

21. Спектор А. Ю. Порівняльний аналіз цифрових платформ музейних установ: спільні особливості та ключові відмінності // Вісник Харківського національного університету імені В. Н. Каразіна. Серія: Історія. – 2025. – № 45. – С. 67–75.

22. Лебідь Т. Проблеми цифровізації пам'яток та електронного обліку музейних колекцій // Вісник Книжкової палати. – 2024. – № 1. – С. 40–46.

23. Гінжол А. І. Рекомендаційна система онлайн-книгарні на основі засобів машинного навчання : кваліфікаційна робота магістра [Електронний ресурс]. – Херсон, 2024. – URL: https://krs.chmnu.edu.ua/jspui/bitstream/123456789/3272/1/KPM_Гінжол_608м_2024.pdf (дата звернення 24.04.2026).

24. Губа А. В. Автоматизована інформаційна система у сучасному музеї : магістерська робота [Електронний ресурс]. – Київ, 2023. – URL: <https://elib.nakkkim.edu.ua/bitstream/handle/123456789/5301/%2B%D0%93%D1%83%D0%B1%D0%B0%20%D0%90.%20%D0%9C%D0%90%D0%93%D0%8>

[6%D0%A1%D0%A2%D0%95%D0%A0%D0%A1%D0%AC%D0%9A%D0%90.pdf?sequence=1&isAllowed=y](#) (дата звернення 24.04.2026).

25. Ковальчук Є. І., Надольська В. В. Музеєзнавство: навчально-методичні матеріали для підготовки бакалавра спеціальності 027 «Музеєзнавство, пам'яткознавство» освітньо-професійної програми «Музейний менеджмент, культурний туризм». Луцьк, 2023. 60 с.

26. Таранова Д. В. Традиційні мас-медіа та соціальні мережі в системі популяризації Сумського обласного художнього музею імені Никанора Онацького: кваліфікаційна робота [Електронний ресурс]. – Суми, 2021. – URL: <https://essuir.sumdu.edu.ua/bitstreams/430defce-1ec4-4a6f-8997-78924bdf863c/download> (дата звернення 24.04.2026).

27. Шман С. Ю. Музейні колекції як джерельна база експертних досліджень // Вісник Національної академії керівних кадрів культури і мистецтв. – 2023. – № 1. – С. 169–173.

28. Качковська Я. О. Розвиток концепції цифрових музеїв: історіографічний огляд // Культурологічний альманах. – 2024. – Вип. 1 (9). – С. 101–106.

ДОДАТКИ

Основні лістинги програмного коду

index.jsx

```
/**
 * AI-асистент музею
 * Плаваюча кнопка з чат-панеллю
 */
import React, { useState, useEffect, useRef } from 'react'
import styled, { keyframes } from 'styled-components'
import { FiMessageCircle, FiX, FiSend, FiClock, FiMapPin, FiDollarSign, FiNavigation } from 'react-icons/fi'
import { aiAPI } from '../utils/api'
const AIAssistant = () => {
  const [isOpen, setIsOpen] = useState(false)
  const [messages, setMessages] = useState([])
  const [inputValue, setInputValue] = useState('')
  const [isLoading, setIsLoading] = useState(false)
  const [sessionId, setSessionId] = useState(null)
  const messagesEndRef = useRef(null)
  const quickQuestions = [
    { text: 'Як працює музей?', icon: <FiClock /> },
    { text: 'Скільки коштують квитки?', icon: <FiDollarSign /> },
    { text: 'Які екскурсії є?', icon: <FiMapPin /> },
    { text: 'Як дістатися?', icon: <FiNavigation /> },
  ]
}
useEffect(() => {
  if (isOpen && messages.length === 0) {
    // Додаємо привітальне повідомлення
    setMessages([
      {
        role: 'assistant',
        content: 'Вітаю! Я віртуальний асистент музею. Чим можу допомогти?',
        timestamp: new Date().toISOString()
      }
    ])
  }, [isOpen])
  useEffect(() => {
    scrollToBottom()
  }, [messages])
  const scrollToBottom = () => {
    messagesEndRef.current?.scrollIntoView({ behavior: 'smooth' })
  }
  const handleSend = async (text = inputValue) => {
    if (!text.trim() || isLoading) return
    const userMessage = {
      role: 'user',
      content: text.trim(),
      timestamp: new Date().toISOString()
    }
    setMessages(prev => [...prev, userMessage])
    setInputValue('')
    setIsLoading(true)
    try {
      const response = await aiAPI.chat(text.trim(), sessionId)
      if (response.data.session_id) {
        setSessionId(response.data.session_id)
      }
      const assistantMessage = {
        role: 'assistant',
        content: response.data.response,
        timestamp: new Date().toISOString(),
        provider: response.data.provider
      }
      setMessages(prev => [...prev, assistantMessage])
    } catch (error) {
      console.error('AI chat error:', error)
      setMessages(prev => [...prev, {
        role: 'assistant',
        content: 'Вибачте, сталася помилка. Спробуйте пізніше або зверніться до менеджера через чат.',
        timestamp: new Date().toISOString(),
        isError: true
      }])
    } finally {

```

```

    setIsLoading(false)
  }
}
const handleQuickQuestion = (question) => {
  handleSend(question)
}
const handleKeyPress = (e) => {
  if (e.key === 'Enter' && !e.shiftKey) {
    e.preventDefault()
    handleSend()
  }
}
return (
  <>
    { /* Плавающая кнопка */ }
    <FloatingButton onClick={() => setIsOpen(!isOpen)} $isOpen={isOpen}>
      {isOpen ? <FiX /> : <FiMessageCircle />}
      {isOpen && <ButtonPulse />}
    </FloatingButton>
    { /* Чат-панель */ }
    <ChatPanel $isOpen={isOpen}>
      <ChatHeader>
        <HeaderContent>
          <HeaderIcon>M</HeaderIcon>
          <HeaderText>
            <HeaderTitle>Ассистент музею</HeaderTitle>
            <HeaderStatus>
              <StatusDot />
              Онлайн
            </HeaderStatus>
          </HeaderText>
        </HeaderContent>
        <CloseButton onClick={() => setIsOpen(false)}>
          <FiX />
        </CloseButton>
      </ChatHeader>
      <MessagesContainer>
        {messages.map((msg, index) => (
          <Message key={index} $role={msg.role} $isError={msg.isError}>
            {msg.role === 'assistant' && (
              <MessageAvatar>M</MessageAvatar>
            )}
            <MessageContent $role={msg.role}>
              <MessageText>{msg.content}</MessageText>
              <MessageTime>
                {new Date(msg.timestamp).toLocaleTimeString('uk-UA', {
                  hour: '2-digit',
                  minute: '2-digit'
                })}
              </MessageTime>
            </MessageContent>
          </Message>
        ))}
        {isLoading && (
          <Message $role="assistant">
            <MessageAvatar>M</MessageAvatar>
            <MessageContent $role="assistant">
              <TypingIndicator>
                <TypingDot />
                <TypingDot />
                <TypingDot />
              </TypingIndicator>
            </MessageContent>
          </Message>
        )}
      </MessagesContainer>
      {messages.length <= 1 && (
        <QuickQuestions>
          {quickQuestions.map((q, index) => (
            <QuickButton key={index} onClick={() => handleQuickQuestion(q.text)}>
              {q.icon}
              <span>{q.text}</span>
            </QuickButton>
          ))}
        </QuickQuestions>
      )}
    </>
  )
}

```

```

<InputContainer>
  <Input
    value={inputValue}
    onChange={(e) => setInputValue(e.target.value)}
    onKeyPress={handleKeyPress}
    placeholder="Напишіть повідомлення..."
    disabled={isLoading}
  />
  <SendButton onClick={() => handleSend()} disabled={!inputValue.trim() || isLoading}>
    <FiSend />
  </SendButton>
</InputContainer>
</ChatPanel>
</>
)
}
// Animations
const pulse = keyframes`
  0% {
    transform: scale(1);
    opacity: 0.8;
  }
  50% {
    transform: scale(1.5);
    opacity: 0;
  }
  100% {
    transform: scale(1);
    opacity: 0;
  }
`;

const bounce = keyframes`
  0%, 60%, 100% {
    transform: translateY(0);
  }
  30% {
    transform: translateY(-4px);
  }
`;

const slideIn = keyframes`
  from {
    opacity: 0;
    transform: translateY(20px) scale(0.95);
  }
  to {
    opacity: 1;
    transform: translateY(0) scale(1);
  }
`;

// Styled Components
const FloatingButton = styled.button`
  position: fixed;
  bottom: 24px;
  right: 24px;
  width: 60px;
  height: 60px;
  border-radius: 50%;
  background: ${props => props.$isOpen
    ? props.theme.colors.primary
    : props.theme.gradients.gold};
  color: ${props => props.$isOpen
    ? props.theme.colors.cream
    : props.theme.colors.primaryDark};
  border: none;
  cursor: pointer;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 24px;
  box-shadow: ${props => props.theme.shadows.lg};
  transition: all 0.3s ease;
  z-index: 1001;
  &:hover {
    transform: scale(1.1);
    box-shadow: ${props => props.theme.shadows.gold};
  }
  svg {

```

```

    transition: transform 0.3s ease;
  }
  &:hover svg {
    transform: ${props => props.$isOpen ? 'rotate(90deg)' : 'scale(1.1)'};
  }
},

const ButtonPulse = styled.span`
  position: absolute;
  width: 100%;
  height: 100%;
  border-radius: 50%;
  background: ${props => props.theme.colors.gold};
  animation: ${pulse} 2s ease-in-out infinite;
  z-index: -1;
},

const ChatPanel = styled.div`
  position: fixed;
  bottom: 100px;
  right: 24px;
  width: 380px;
  max-width: calc(100vw - 48px);
  height: 550px;
  max-height: calc(100vh - 150px);
  background: ${props => props.theme.colors.surface};
  border-radius: ${props => props.theme.borderRadius.xl};
  box-shadow: ${props => props.theme.shadows.xxl};
  display: flex;
  flex-direction: column;
  z-index: 1000;
  opacity: ${props => props.$isOpen ? 1 : 0};
  visibility: ${props => props.$isOpen ? 'visible' : 'hidden'};
  transform: ${props => props.$isOpen ? 'translateY(0) scale(1)' : 'translateY(20px) scale(0.95)'};
  transition: all 0.3s ease;
  border: 1px solid ${props => props.theme.colors.borderGold};
  overflow: hidden;
  @media (max-width: 480px) {
    bottom: 80px;
    right: 12px;
    width: calc(100vw - 24px);
    height: calc(100vh - 120px);
  }
},

const ChatHeader = styled.div`
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 16px 20px;
  background: ${props => props.theme.gradients.burgundy};
  color: ${props => props.theme.colors.cream};
},

const HeaderComponent = styled.div`
  display: flex;
  align-items: center;
  gap: 12px;
},

const HeaderIcon = styled.div`
  width: 40px;
  height: 40px;
  background: ${props => props.theme.gradients.gold};
  border-radius: ${props => props.theme.borderRadius.md};
  display: flex;
  align-items: center;
  justify-content: center;
  font-family: ${props => props.theme.fonts.display};
  font-size: 18px;
  font-weight: 700;
  color: ${props => props.theme.colors.primaryDark};
},

const HeaderText = styled.div``
const HeaderTitle = styled.h3`
  font-family: ${props => props.theme.fonts.heading};
  font-size: 16px;
  font-weight: 600;
  margin: 0 0 2px;
},

const HeaderStatus = styled.div`
  display: flex;

```

```

align-items: center;
gap: 6px;
font-family: ${props => props.theme.fonts.ui};
font-size: 12px;
opacity: 0.9;
`

const StatusDot = styled.span`
width: 8px;
height: 8px;
background: #4ade80;
border-radius: 50%;
`

const CloseButton = styled.button`
background: rgba(255, 255, 255, 0.1);
border: none;
color: ${props => props.theme.colors.cream};
width: 32px;
height: 32px;
border-radius: ${props => props.theme.borderRadius.md};
display: flex;
align-items: center;
justify-content: center;
cursor: pointer;
transition: all 0.2s ease;
&:hover {
background: rgba(255, 255, 255, 0.2);
}
`

const MessagesContainer = styled.div`
flex: 1;
overflow-y: auto;
padding: 20px;
display: flex;
flex-direction: column;
gap: 16px;
background: ${props => props.theme.colors.backgroundDark};
`

const Message = styled.div`
display: flex;
gap: 10px;
align-items: flex-start;
justify-content: ${props => props.$role === 'user' ? 'flex-end' : 'flex-start'};
animation: ${props => props.$slideIn} 0.3s ease-out;
`

const MessageAvatar = styled.div`
width: 32px;
height: 32px;
background: ${props => props.theme.gradients.gold};
border-radius: ${props => props.theme.borderRadius.sm};
display: flex;
align-items: center;
justify-content: center;
font-family: ${props => props.theme.fonts.display};
font-size: 14px;
font-weight: 700;
color: ${props => props.theme.colors.primaryDark};
flex-shrink: 0;
`

const MessageContent = styled.div`
max-width: 75%;
padding: 12px 16px;
border-radius: ${props => props.$role === 'user'
? `${props.theme.borderRadius.lg} ${props.theme.borderRadius.lg} 4px ${props.theme.borderRadius.lg}`
: `${props.theme.borderRadius.lg} ${props.theme.borderRadius.lg} ${props.theme.borderRadius.lg} 4px`};
background: ${props => props.$role === 'user'
? props.theme.gradients.burgundy
: props.theme.colors.surface};
color: ${props => props.$role === 'user'
? props.theme.colors.cream
: props.theme.colors.text};
box-shadow: ${props => props.theme.shadows.sm};
`

const MessageText = styled.p`
font-family: ${props => props.theme.fonts.body};
font-size: 14px;
line-height: 1.5;
margin: 0;

```

```

    white-space: pre-wrap;
  `
const MessageTime = styled.span`
  display: block;
  font-family: ${props => props.theme.fonts.ui};
  font-size: 10px;
  opacity: 0.6;
  margin-top: 6px;
  text-align: right;
`

const TypingIndicator = styled.div`
  display: flex;
  gap: 4px;
  padding: 8px 0;
`

const TypingDot = styled.span`
  width: 8px;
  height: 8px;
  background: ${props => props.theme.colors.gold};
  border-radius: 50%;
  animation: ${bounce} 1.4s ease-in-out infinite;
  &:nth-child(2) {
    animation-delay: 0.2s;
  }
  &:nth-child(3) {
    animation-delay: 0.4s;
  }
`

const QuickQuestions = styled.div`
  display: flex;
  flex-wrap: wrap;
  gap: 8px;
  padding: 12px 20px;
  background: ${props => props.theme.colors.surface};
  border-top: 1px solid ${props => props.theme.colors.border};
`

const QuickButton = styled.button`
  display: flex;
  align-items: center;
  gap: 6px;
  padding: 8px 12px;
  background: ${props => props.theme.colors.backgroundDark};
  border: 1px solid ${props => props.theme.colors.border};
  border-radius: ${props => props.theme.borderRadius.full};
  font-family: ${props => props.theme.fonts.ui};
  font-size: 12px;
  color: ${props => props.theme.colors.textSecondary};
  cursor: pointer;
  transition: all 0.2s ease;
  &:hover {
    border-color: ${props => props.theme.colors.gold};
    color: ${props => props.theme.colors.gold};
    background: ${props => props.theme.colors.overlayGold};
  }
  svg {
    font-size: 14px;
  }
`

const InputContainer = styled.div`
  display: flex;
  gap: 12px;
  padding: 16px 20px;
  background: ${props => props.theme.colors.surface};
  border-top: 1px solid ${props => props.theme.colors.border};
`

const Input = styled.input`
  flex: 1;
  padding: 12px 16px;
  border: 1px solid ${props => props.theme.colors.border};
  border-radius: ${props => props.theme.borderRadius.full};
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  background: ${props => props.theme.colors.backgroundDark};
  color: ${props => props.theme.colors.text};
  transition: all 0.2s ease;
  &:focus {
    outline: none;
  }
`

```

```

border-color: ${props => props.theme.colors.gold};
box-shadow: 0 0 3px ${props => props.theme.colors.overlayGold};
}
&::placeholder {
  color: ${props => props.theme.colors.textMuted};
}
&:disabled {
  opacity: 0.6;
}
,

```

```

const SendButton = styled.button`
width: 44px;
height: 44px;
border-radius: 50%;
background: ${props => props.theme.gradients.gold};
color: ${props => props.theme.colors.primaryDark};
border: none;
cursor: pointer;
display: flex;
align-items: center;
justify-content: center;
font-size: 18px;
transition: all 0.2s ease;
&:hover:not(:disabled) {
  transform: scale(1.05);
  box-shadow: ${props => props.theme.shadows.gold};
}
&:disabled {
  opacity: 0.5;
  cursor: not-allowed;
}
,

```

```
export default AIAssistant
```

Avatar.jsx

```

import * as React from "react"
import * as AvatarPrimitive from "@radix-ui/react-avatar"
import { cn } from "../../lib/utils"
const Avatar = React.forwardRef(({ className, ...props }, ref) => (
  <AvatarPrimitive.Root
    ref={ref}
    className={cn(
      "relative flex h-10 w-10 shrink-0 overflow-hidden rounded-full",
      className
    )}
    {...props}
  />
))
Avatar.displayName = AvatarPrimitive.Root.displayName
const AvatarImage = React.forwardRef(({ className, ...props }, ref) => (
  <AvatarPrimitive.Image
    ref={ref}
    className={cn("aspect-square h-full w-full", className)}
    {...props}
  />
))
AvatarImage.displayName = AvatarPrimitive.Image.displayName
const AvatarFallback = React.forwardRef(({ className, ...props }, ref) => (
  <AvatarPrimitive.Fallback
    ref={ref}
    className={cn(
      "flex h-full w-full items-center justify-center rounded-full bg-primary/10 text-primary font-medium",
      className
    )}
    {...props}
  />
))
AvatarFallback.displayName = AvatarPrimitive.Fallback.displayName
export { Avatar, AvatarImage, AvatarFallback }

```

Badge.jsx

```

import * as React from "react"
import { cva } from "class-variance-authority"
import { cn } from "../../lib/utils"
const badgeVariants = cva(

```

```

"inline-flex items-center rounded-full px-2.5 py-0.5 text-xs font-semibold transition-colors focus:outline-none
focus:ring-2 focus:ring-primary focus:ring-offset-2",
{
  variants: {
    variant: {
      default: "bg-primary/10 text-primary",
      secondary: "bg-secondary/10 text-secondary-dark",
      success: "bg-green-100 text-green-700",
      warning: "bg-amber-100 text-amber-700",
      destructive: "bg-red-100 text-red-700",
      outline: "border border-border text-slate-700",
      museum: "bg-museum-gold/20 text-museum-gold-dark",
    },
  },
  defaultVariants: {
    variant: "default",
  },
}
)
function Badge({ className, variant, ...props }) {
  return (
    <div className={cn(badgeVariants({ variant }), className)} {...props} />
  )
}
export { Badge, badgeVariants }

```

Button.jsx

```

import * as React from "react"
import { Slot } from "@radix-ui/react-slot"
import { cva } from "class-variance-authority"
import { cn } from "../../lib/utils"
const buttonVariants = cva(
  "inline-flex items-center justify-center whitespace-nowrap rounded-md text-sm font-medium transition-colors focus-
visible:outline-none focus-visible:ring-2 focus-visible:ring-primary focus-visible:ring-offset-2 disabled:pointer-events-
none disabled:opacity-50",
  {
    variants: {
      variant: {
        default: "bg-primary text-white hover:bg-primary-dark shadow-md hover:shadow-lg",
        secondary: "bg-secondary text-white hover:bg-secondary-dark shadow-md hover:shadow-lg",
        destructive: "bg-red-500 text-white hover:bg-red-600 shadow-md",
        outline: "border-2 border-primary text-primary hover:bg-primary hover:text-white",
        ghost: "hover:bg-slate-100 hover:text-primary",
        link: "text-primary underline-offset-4 hover:underline",
        museum: "bg-museum-gold text-museum-dark hover:bg-museum-gold-light shadow-glow-gold",
      },
      size: {
        default: "h-10 px-4 py-2",
        sm: "h-8 rounded-md px-3 text-xs",
        lg: "h-12 rounded-md px-8 text-base",
        icon: "h-10 w-10",
      },
    },
    defaultVariants: {
      variant: "default",
      size: "default",
    },
  }
)
const Button = React.forwardRef(
  ({ className, variant, size, asChild = false, ...props }, ref) => {
    const Comp = asChild ? Slot : "button"
    return (
      <Comp
        className={cn(buttonVariants({ variant, size, className }))}
        ref={ref}
        {...props}
      />
    )
  }
)
Button.displayName = "Button"
export { Button, buttonVariants }

```

Card.jsx

```

import * as React from "react"

```

```

import { cn } from "../../lib/utils"
const Card = React.forwardRef(({ className, ...props }, ref) => (
  <div
    ref={ref}
    className={cn(
      "rounded-lg border border-border bg-surface shadow-sm",
      className
    )}
    {...props}
  />
))
Card.displayName = "Card"
const CardHeader = React.forwardRef(({ className, ...props }, ref) => (
  <div
    ref={ref}
    className={cn("flex flex-col space-y-1.5 p-6", className)}
    {...props}
  />
))
CardHeader.displayName = "CardHeader"
const CardTitle = React.forwardRef(({ className, ...props }, ref) => (
  <h3
    ref={ref}
    className={cn("text-xl font-semibold leading-none tracking-tight", className)}
    {...props}
  />
))
CardTitle.displayName = "CardTitle"
const CardDescription = React.forwardRef(({ className, ...props }, ref) => (
  <p
    ref={ref}
    className={cn("text-sm text-slate-500", className)}
    {...props}
  />
))
CardDescription.displayName = "CardDescription"
const CardContent = React.forwardRef(({ className, ...props }, ref) => (
  <div ref={ref} className={cn("p-6 pt-0", className)} {...props} />
))
CardContent.displayName = "CardContent"
const CardFooter = React.forwardRef(({ className, ...props }, ref) => (
  <div
    ref={ref}
    className={cn("flex items-center p-6 pt-0", className)}
    {...props}
  />
))
CardFooter.displayName = "CardFooter"
export { Card, CardHeader, CardFooter, CardTitle, CardDescription, CardContent }

```

Dialog.jsx

```

import * as React from "react"
import * as DialogPrimitive from "@radix-ui/react-dialog"
import { X } from "lucide-react"
import { cn } from "../../lib/utils"
const Dialog = DialogPrimitive.Root
const DialogTrigger = DialogPrimitive.Trigger
const DialogPortal = DialogPrimitive.Portal
const DialogClose = DialogPrimitive.Close
const DialogOverlay = React.forwardRef(({ className, ...props }, ref) => (
  <DialogPrimitive.Overlay
    ref={ref}
    className={cn(
      "fixed inset-0 z-50 bg-black/50 backdrop-blur-sm data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
      className
    )}
    {...props}
  />
))
DialogOverlay.displayName = DialogPrimitive.Overlay.displayName
const DialogContent = React.forwardRef(({ className, children, ...props }, ref) => (
  <DialogPortal>
    <DialogOverlay />
    <DialogPrimitive.Content
      ref={ref}
      className={cn(

```

```

    "fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 border border-
border bg-surface p-6 shadow-lg duration-200 data-[state=open]:animate-in data-[state=closed]:animate-out data-
[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95
data-[state=closed]:slide-out-to-left-1/2 data-[state=closed]:slide-out-to-top-[48%] data-[state=open]:slide-in-from-left-
1/2 data-[state=open]:slide-in-from-top-[48%] sm:rounded-lg",
    className
  )}
  {...props}
)
{children}
<DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 ring-offset-background transition-
opacity hover:opacity-100 focus:outline-none focus:ring-2 focus:ring-primary focus:ring-offset-2 disabled:pointer-events-
none data-[state=open]:bg-slate-100">
  <X className="h-4 w-4" />
  <span className="sr-only">Закрити</span>
</DialogPrimitive.Close>
</DialogPrimitive.Content>
</DialogPortal>
))
DialogContent.displayName = DialogPrimitive.Content.displayName
const DialogHeader = ({ className, ...props }) => (
  <div
    className={cn("flex flex-col space-y-1.5 text-center sm:text-left", className)}
    {...props}
  />
)
DialogHeader.displayName = "DialogHeader"
const DialogFooter = ({ className, ...props }) => (
  <div
    className={cn("flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2", className)}
    {...props}
  />
)
DialogFooter.displayName = "DialogFooter"
const DialogTitle = React.forwardRef(({ className, ...props }, ref) => (
  <DialogPrimitive.Title
    ref={ref}
    className={cn("text-lg font-semibold leading-none tracking-tight", className)}
    {...props}
  />
))
DialogTitle.displayName = DialogPrimitive.Title.displayName
const DialogDescription = React.forwardRef(({ className, ...props }, ref) => (
  <DialogPrimitive.Description
    ref={ref}
    className={cn("text-sm text-slate-500", className)}
    {...props}
  />
))
DialogDescription.displayName = DialogPrimitive.Description.displayName
export {
  Dialog,
  DialogPortal,
  DialogOverlay,
  DialogTrigger,
  DialogClose,
  DialogContent,
  DialogHeader,
  DialogFooter,
  DialogTitle,
  DialogDescription,
}

```

Input.jsx

```

import * as React from "react"
import { cn } from "../../lib/utils"
const Input = React.forwardRef(({ className, type, ...props }, ref) => {
  return (
    <input
      type={type}
      className={cn(
        "flex h-10 w-full rounded-md border border-border bg-surface px-3 py-2 text-sm ring-offset-background file:border-
0 file:bg-transparent file:text-sm file:font-medium placeholder:text-slate-400 focus-visible:outline-none focus-
visible:ring-2 focus-visible:ring-primary focus-visible:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50",
        className
      )}
      ref={ref}
    />
  )
}

```

```

    {...props}
  />
)
})
Input.displayName = "Input"
export { Input }

```

Label.jsx

```

import * as React from "react"
import * as LabelPrimitive from "@radix-ui/react-label"
import { cva } from "class-variance-authority"
import { cn } from "../../lib/utils"
const labelVariants = cva(
  "text-sm font-medium leading-none peer-disabled:cursor-not-allowed peer-disabled:opacity-70"
)
const Label = React.forwardRef(({ className, ...props }, ref) => (
  <LabelPrimitive.Root
    ref={ref}
    className={cn(labelVariants(), className)}
    {...props}
  />
))
Label.displayName = LabelPrimitive.Root.displayName
export { Label }

```

Skeleton.jsx

```

import { cn } from "../../lib/utils"
function Skeleton({ className, ...props }) {
  return (
    <div
      className={cn("animate-pulse rounded-md bg-slate-200", className)}
      {...props}
    />
  )
}
export { Skeleton }

```

Toast.jsx

```

import * as React from "react"
import * as ToastPrimitives from "@radix-ui/react-toast"
import { cva } from "class-variance-authority"
import { X } from "lucide-react"
import { cn } from "../../lib/utils"
const ToastProvider = ToastPrimitives.Provider
const ToastViewport = React.forwardRef(({ className, ...props }, ref) => (
  <ToastPrimitives.Viewport
    ref={ref}
    className={cn(
      "fixed top-0 z-[100] flex max-h-screen w-full flex-col-reverse p-4 sm:bottom-0 sm:right-0 sm:top-auto sm:flex-col
      md:max-w-[420px]",
      className
    )}
    {...props}
  />
))
ToastViewport.displayName = ToastPrimitives.Viewport.displayName
const toastVariants = cva(
  "group pointer-events-auto relative flex w-full items-center justify-between space-x-4 overflow-hidden rounded-md
  border p-6 pr-8 shadow-lg transition-all data-[swipe=cancel]:translate-x-0 data-[swipe=end]:translate-x-[var(--radix-
  toast-swipe-end-x)] data-[swipe=move]:translate-x-[var(--radix-toast-swipe-move-x)] data-[swipe=move]:transition-none
  data-[state=open]:animate-in data-[state=closed]:animate-out data-[swipe=end]:animate-out data-[state=closed]:fade-
  out-80 data-[state=closed]:slide-out-to-right-full data-[state=open]:slide-in-from-top-full data-[state=open]:sm:slide-in-
  from-bottom-full",
  {
    variants: {
      variant: {
        default: "border-border bg-surface text-slate-900",
        success: "border-green-200 bg-green-50 text-green-900",
        destructive: "destructive group border-red-200 bg-red-50 text-red-900",
        warning: "border-amber-200 bg-amber-50 text-amber-900",
      },
    },
    defaultVariants: {
      variant: "default",
    },
  }
)

```

```

)
const Toast = React.forwardRef(({ className, variant, ...props }, ref) => {
  return (
    <ToastPrimitives.Root
      ref={ref}
      className={cn(toastVariants({ variant }), className)}
      {...props}
    />
  )
})
Toast.displayName = ToastPrimitives.Root.displayName
const ToastAction = React.forwardRef(({ className, ...props }, ref) => (
  <ToastPrimitives.Action
    ref={ref}
    className={cn(
      "inline-flex h-8 shrink-0 items-center justify-center rounded-md border border-border bg-transparent px-3 text-sm font-medium ring-offset-background transition-colors hover:bg-slate-100 focus:outline-none focus:ring-2 focus:ring-primary focus:ring-offset-2 disabled:pointer-events-none disabled:opacity-50",
      className
    )}
    {...props}
  />
))
ToastAction.displayName = ToastPrimitives.Action.displayName
const ToastClose = React.forwardRef(({ className, ...props }, ref) => (
  <ToastPrimitives.Close
    ref={ref}
    className={cn(
      "absolute right-2 top-2 rounded-md p-1 text-slate-500 opacity-0 transition-opacity hover:text-slate-900 focus:opacity-100 focus:outline-none focus:ring-2 group-hover:opacity-100",
      className
    )}
    toast-close=""
    {...props}
  >
    <X className="h-4 w-4" />
  </ToastPrimitives.Close>
))
ToastClose.displayName = ToastPrimitives.Close.displayName
const ToastTitle = React.forwardRef(({ className, ...props }, ref) => (
  <ToastPrimitives.Title
    ref={ref}
    className={cn("text-sm font-semibold", className)}
    {...props}
  />
))
ToastTitle.displayName = ToastPrimitives.Title.displayName
const ToastDescription = React.forwardRef(({ className, ...props }, ref) => (
  <ToastPrimitives.Description
    ref={ref}
    className={cn("text-sm opacity-90", className)}
    {...props}
  />
))
ToastDescription.displayName = ToastPrimitives.Description.displayName
export {
  ToastProvider,
  ToastViewport,
  Toast,
  ToastTitle,
  ToastDescription,
  ToastClose,
  ToastAction,
}

```

DatePickerUa.jsx

```

/**
 * Компонент вибору дати з українською локалізацією
 */
import React from 'react'
import DatePicker, { registerLocale } from 'react-datepicker'
import { uk } from 'date-fns/locale'
import styled from 'styled-components'
import 'react-datepicker/dist/react-datepicker.css'
// Реєстрація української локалі
registerLocale('uk', uk)
const DatePickerWrapper = styled.div`

```

```

.react-datepicker-wrapper {
  width: 100%;
}
.react-datepicker__input-container {
  width: 100%;
}
.react-datepicker__input-container input {
  width: 100%;
  padding: 12px 16px;
  border: 2px solid #ddd;
  border-radius: 8px;
  font-size: 16px;
  transition: border-color 0.3s ease;
  box-sizing: border-box;
  &:focus {
    outline: none;
    border-color: #1a5f4a;
  }
}
.react-datepicker {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  border: 2px solid #1a5f4a;
  border-radius: 12px;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.15);
}
.react-datepicker__header {
  background: linear-gradient(135deg, #1a5f4a 0%, #2d8a6e 100%);
  border-bottom: none;
  border-top-left-radius: 10px;
  border-top-right-radius: 10px;
  padding-top: 12px;
}
.react-datepicker__current-month {
  color: white;
  font-weight: 600;
  font-size: 16px;
  text-transform: capitalize;
}
.react-datepicker__day-name {
  color: rgba(255, 255, 255, 0.9);
  font-weight: 500;
  text-transform: capitalize;
}
.react-datepicker__navigation {
  top: 12px;
}
.react-datepicker__navigation-icon::before {
  border-color: white;
}
.react-datepicker__day {
  border-radius: 8px;
  transition: all 0.2s ease;
  &:hover {
    background-color: #e8f5f0;
  }
}
.react-datepicker__day--selected {
  background: linear-gradient(135deg, #1a5f4a 0%, #2d8a6e 100%);
  color: white;
  font-weight: 600;
  &:hover {
    background: linear-gradient(135deg, #155241 0%, #257a5f 100%);
  }
}
.react-datepicker__day--keyboard-selected {
  background-color: #e8f5f0;
  color: #1a5f4a;
}
.react-datepicker__day--today {
  font-weight: bold;
  color: #1a5f4a;
}
.react-datepicker__day--disabled {
  color: #ccc;
  cursor: not-allowed;
  &:hover {
    background-color: transparent;
  }
}

```

```

    }
  }
  .react-datepicker__triangle {
    display: none;
  }
},
const DatePickerUa = ({
  selected,
  onChange,
  minDate,
  maxDate,
  placeholderText = 'Оберіть дату',
  dateFormat = 'dd.MM.yyyy',
  ...props
}) => {
  return (
    <DatePickerWrapper>
      <DatePicker
        selected={selected}
        onChange={onChange}
        locale="uk"
        dateFormat={dateFormat}
        minDate={minDate}
        maxDate={maxDate}
        placeholderText={placeholderText}
        autoComplete="off"
        {...props}
      />
    </DatePickerWrapper>
  )
}
export default DatePickerUa

```

Layout.jsx

```

/**
 * Основний макет додатку
 * Стиль: Baroque Luxury
 */
import React, { useState, useEffect } from 'react'
import { Link, useNavigate, useLocation } from 'react-router-dom'
import styled from 'styled-components'
import { useAuth } from '../context/AuthContext'
import {
  FiLogout,
  FiUser,
  FiHome,
  FiFileText,
  FiMapPin,
  FiStar,
  FiCreditCard,
  FiMessageSquare,
  FiBarChart2,
  FiHeart,
  FiBell,
  FiMenu,
  FiX
} from 'react-icons/fi'
const Layout = ({ children }) => {
  const { user, logout, isAuthenticated, isManager, isAdmin, isClient } = useAuth()
  const navigate = useNavigate()
  const location = useLocation()
  const [scrolled, setScrolled] = useState(false)
  const [mobileMenuOpen, setMobileMenuOpen] = useState(false)
  useEffect(() => {
    const handleScroll = () => {
      setScrolled(window.scrollY > 50)
    }
    window.addEventListener('scroll', handleScroll, { passive: true })
    return () => window.removeEventListener('scroll', handleScroll)
  }, [])
  useEffect(() => {
    setMobileMenuOpen(false)
  }, [location])
  const handleLogout = () => {
    logout()
    navigate('/login')
  }
}

```

```

const isActive = (path) => location.pathname === path
return (
  <LayoutWrapper>
    <Header $scrolled={scrolled}>
      <HeaderContainer>
        <LogoSection>
          <Logo to="/">
            <LogoIcon>М</LogoIcon>
            <LogoText>
              <LogoMain>Художній музей</LogoMain>
              <LogoSub>сучасного мистецтва</LogoSub>
            </LogoText>
          </Logo>
        </LogoSection>
        <MobileMenuButton onClick={() => setMobileMenuOpen(!mobileMenuOpen)}>
          {mobileMenuOpen ? <FiX /> : <FiMenu />}
        </MobileMenuButton>
        <Nav $open={mobileMenuOpen}>
          <NavLink to="/" $active={isActive('/')}>
            <FiHome />
            <span>Головна</span>
          </NavLink>
          <NavLink to="/news" $active={isActive('/news')}>
            <FiFileText />
            <span>Новини</span>
          </NavLink>
          <NavLink to="/excursions" $active={isActive('/excursions')}>
            <FiMapPin />
            <span>Експерсії</span>
          </NavLink>
          {isAuthenticated && (
            <NavLink to="/reviews" $active={isActive('/reviews')}>
              <FiStar />
              <span>Відгуки</span>
            </NavLink>
            {isClient && (
              <NavLink to="/tickets" $active={isActive('/tickets')}>
                <FiCreditCard />
                <span>Квитки</span>
              </NavLink>
              <NavLink to="/wishlist" $active={isActive('/wishlist')}>
                <FiHeart />
                <span>Обране</span>
              </NavLink>
            )}
          <NavLink to="/chat" $active={isActive('/chat')}>
            <FiMessageSquare />
            <span>Чат</span>
          </NavLink>
        )}
        {isManager && (
          <NavLink to="/manager" $active={isActive('/manager')}>
            <FiBarChart2 />
            <span>Панель</span>
          </NavLink>
        )}
        {isAdmin && (
          <NavLink to="/admin" $active={isActive('/admin')} $admin>
            <FiBarChart2 />
            <span>Адмін</span>
          </NavLink>
        )}
      </Nav>
      <UserSection>
        {isAuthenticated ? (
          <isClient && (
            <NotificationButton>
              <FiBell />
              <NotificationBadge>3</NotificationBadge>
            </NotificationButton>
          )}
          <UserInfo to="/profile">

```

```

    <UserAvatar>
      {user?.username?.charAt(0).toUpperCase()}
    </UserAvatar>
    <UserDetails>
      <UserName>{user?.username}</UserName>
      <UserRole $role={user?.role}>
        {user?.role === 'admin' ? 'Адмін' : user?.role === 'manager' ? 'Менеджер' : user?.role === 'client' ? 'Клієнт' :
'Гість'}
      </UserRole>
    </UserDetails>
  </UserInfo>
  <LogoutButton onClick={handleLogout}>
    <FiLogOut />
  </LogoutButton>
</>
): (
  <AuthButtons>
    <LoginLink to="/login">Вхід</LoginLink>
    <RegisterLink to="/register">Реєстрація</RegisterLink>
  </AuthButtons>
)>
</UserSection>
</HeaderContainer>
</Header>
<Main>{children}</Main>
<Footer>
  <FooterContent>
    <FooterMain>
      <FooterBrand>
        <FooterLogo>
          <LogoIcon>M</LogoIcon>
          <span>Художній музей</span>
        </FooterLogo>
        <FooterDescription>
          Скарбниця українського народного мистецтва.
          Понад 5000 експонатів традиційної культури.
        </FooterDescription>
        </FooterBrand>
        <FooterLinks>
          <FooterColumn>
            <FooterTitle>Навігація</FooterTitle>
            <FooterLink to="/">Головна</FooterLink>
            <FooterLink to="/excursions">Екскурсії</FooterLink>
            <FooterLink to="/news">Новини</FooterLink>
            <FooterLink to="/about">Про музей</FooterLink>
          </FooterColumn>
          <FooterColumn>
            <FooterTitle>Контакти</FooterTitle>
            <FooterText>вул. Степана Бандери, 16</FooterText>
            <FooterText>Дніпро, 49000</FooterText>
            <FooterText>+38 (056) 745-24-17</FooterText>
            <FooterText>info@museum.ua</FooterText>
          </FooterColumn>
          <FooterColumn>
            <FooterTitle>Графік роботи</FooterTitle>
            <FooterText>Вт-Нд: 10:00 - 18:00</FooterText>
            <FooterText>Пн: Вихідний</FooterText>
            <FooterText>Каса до 17:30</FooterText>
          </FooterColumn>
        </FooterLinks>
      </FooterMain>
      <FooterDivider />
      <FooterBottom>
        <Copyright>2025 Художній музей сучасного мистецтва. Всі права захищені.</Copyright>
      </FooterBottom>
    </FooterContent>
  </Footer>
</LayoutWrapper>
)
}
// Styled Components
const LayoutWrapper = styled.div`
  min-height: 100vh;
  display: flex;
  flex-direction: column;
`;
const Header = styled.header`

```

```

background: ${props => props.$scrolled
  ? 'rgba(253, 251, 247, 0.98)'
  : 'rgba(253, 251, 247, 0.95)'};
backdrop-filter: blur(20px);
border-bottom: 1px solid ${props => props.$scrolled
  ? props.theme.colors.borderGold
  : props.theme.colors.border};
padding: ${props => props.$scrolled ? '12px 0' : '16px 0'};
position: sticky;
top: 0;
z-index: 1000;
transition: all 0.3s ease;
box-shadow: ${props => props.$scrolled ? props.theme.shadows.md : 'none'};
,

const HeaderContainer = styled.div`
  max-width: 1400px;
  margin: 0 auto;
  padding: 0 24px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 32px;
,

const LogoSection = styled.div`
  flex-shrink: 0;
,

const Logo = styled(Link)`
  display: flex;
  align-items: center;
  gap: 12px;
  text-decoration: none;
,

const LogoIcon = styled.div`
  width: 48px;
  height: 48px;
  background: ${props => props.theme.gradients.burgundy};
  border-radius: ${props => props.theme.borderRadius.md};
  display: flex;
  align-items: center;
  justify-content: center;
  font-family: ${props => props.theme.fonts.display};
  font-size: 24px;
  font-weight: 700;
  color: ${props => props.theme.colors.gold};
  box-shadow: ${props => props.theme.shadows.md};
,

const LogoText = styled.div`
  display: flex;
  flex-direction: column;
  @media (max-width: 768px) {
    display: none;
  }
,

const LogoMain = styled.span`
  font-family: ${props => props.theme.fonts.display};
  font-size: 18px;
  font-weight: 600;
  color: ${props => props.theme.colors.primary};
  letter-spacing: 0.05em;
  text-transform: uppercase;
,

const LogoSub = styled.span`
  font-family: ${props => props.theme.fonts.heading};
  font-size: 12px;
  color: ${props => props.theme.colors.secondary};
  font-style: italic;
,

const MobileMenuButton = styled.button`
  display: none;
  background: none;
  padding: 8px;
  color: ${props => props.theme.colors.primary};
  font-size: 24px;
  @media (max-width: 968px) {
    display: flex;
    align-items: center;
    justify-content: center;

```

```

    },
    `
const Nav = styled.nav`
  display: flex;
  align-items: center;
  gap: 8px;
  flex: 1;
  justify-content: center;
  @media (max-width: 968px) {
    position: fixed;
    top: 70px;
    left: 0;
    right: 0;
    background: ${props => props.theme.colors.surface};
    flex-direction: column;
    padding: 20px;
    gap: 8px;
    border-bottom: 1px solid ${props => props.theme.colors.border};
    box-shadow: ${props => props.theme.shadows.lg};
    transform: ${props => props.$open ? 'translateY(0)' : 'translateY(-100%)'};
    opacity: ${props => props.$open ? 1 : 0};
    visibility: ${props => props.$open ? 'visible' : 'hidden'};
    transition: all 0.3s ease;
    z-index: 999;
  }
`

const NavLink = styled(Link)`
  display: flex;
  align-items: center;
  gap: 8px;
  padding: 10px 16px;
  border-radius: ${props => props.theme.borderRadius.md};
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  font-weight: 500;
  color: ${props => props.$active ? props.theme.colors.secondary : props.theme.colors.text};
  background: ${props => props.$active ? props.theme.colors.overlayGold : 'transparent'};
  border: 1px solid ${props => props.$active ? props.theme.colors.gold : 'transparent'};
  transition: all 0.3s ease;
  &:hover {
    color: ${props => props.theme.colors.secondary};
    background: ${props => props.theme.colors.overlayGold};
  }
  svg {
    font-size: 16px;
    color: ${props => props.$active ? props.theme.colors.gold : 'inherit'};
  }
  @media (max-width: 968px) {
    width: 100%;
    justify-content: center;
    padding: 14px 20px;
  }
`

const UserSection = styled.div`
  display: flex;
  align-items: center;
  gap: 16px;
  flex-shrink: 0;
`

const NotificationButton = styled.button`
  position: relative;
  background: none;
  padding: 10px;
  color: ${props => props.theme.colors.textSecondary};
  font-size: 20px;
  border-radius: ${props => props.theme.borderRadius.full};
  transition: all 0.3s ease;
  &:hover {
    background: ${props => props.theme.colors.overlayGold};
    color: ${props => props.theme.colors.gold};
  }
`

const NotificationBadge = styled.span`
  position: absolute;
  top: 4px;
  right: 4px;
  min-width: 18px;

```

```

height: 18px;
background: ${props => props.theme.colors.secondary};
color: white;
font-size: 10px;
font-weight: 700;
border-radius: ${props => props.theme.borderRadius.full};
display: flex;
align-items: center;
justify-content: center;
padding: 0 4px;
},

const UserInfo = styled(Link)`
display: flex;
align-items: center;
gap: 12px;
padding: 6px 12px 6px 6px;
background: ${props => props.theme.colors.surfaceElevated};
border: 1px solid ${props => props.theme.colors.border};
border-radius: ${props => props.theme.borderRadius.full};
text-decoration: none;
transition: all 0.3s ease;
&:hover {
border-color: ${props => props.theme.colors.gold};
box-shadow: ${props => props.theme.shadows.gold};
}
@media (max-width: 768px) {
padding: 4px;
& > div:last-child {
display: none;
}
}
},

const UserAvatar = styled.div`
width: 36px;
height: 36px;
background: ${props => props.theme.gradients.gold};
border-radius: ${props => props.theme.borderRadius.full};
display: flex;
align-items: center;
justify-content: center;
font-family: ${props => props.theme.fonts.display};
font-size: 16px;
font-weight: 700;
color: ${props => props.theme.colors.primaryDark};
},

const UserDetails = styled.div`
display: flex;
flex-direction: column;
},

const UserName = styled.span`
font-family: ${props => props.theme.fonts.ui};
font-size: 14px;
font-weight: 600;
color: ${props => props.theme.colors.text};
},

const UserRole = styled.span`
font-family: ${props => props.theme.fonts.ui};
font-size: 11px;
font-weight: 500;
color: ${props => props.$role === 'admin'
? props.theme.colors.gold
: props.$role === 'manager'
? props.theme.colors.secondary
: props.theme.colors.textMuted};
text-transform: uppercase;
letter-spacing: 0.5px;
},

const LogoutButton = styled.button`
background: ${props => props.theme.colors.error};
color: white;
padding: 10px;
border-radius: ${props => props.theme.borderRadius.md};
font-size: 18px;
display: flex;
align-items: center;
justify-content: center;
transition: all 0.3s ease;

```

```

    &:hover {
      background: #b71c1c;
      transform: translateY(-2px);
      box-shadow: ${props => props.theme.shadows.md};
    }
  },

  const AuthButtons = styled.div`
    display: flex;
    align-items: center;
    gap: 12px;

    const LoginLink = styled(Link)`
      font-family: ${props => props.theme.fonts.ui};
      font-size: 14px;
      font-weight: 600;
      color: ${props => props.theme.colors.text};
      padding: 10px 20px;
      border: 1px solid ${props => props.theme.colors.border};
      border-radius: ${props => props.theme.borderRadius.md};
      transition: all 0.3s ease;
      &:hover {
        border-color: ${props => props.theme.colors.gold};
        color: ${props => props.theme.colors.gold};
      }
    },

    const RegisterLink = styled(Link)`
      font-family: ${props => props.theme.fonts.ui};
      font-size: 14px;
      font-weight: 600;
      color: ${props => props.theme.colors.textOnDark};
      padding: 10px 20px;
      background: ${props => props.theme.gradients.burgundy};
      border-radius: ${props => props.theme.borderRadius.md};
      border: 1px solid transparent;
      transition: all 0.3s ease;
      box-shadow: ${props => props.theme.shadows.sm};
      &:hover {
        transform: translateY(-2px);
        box-shadow: ${props => props.theme.shadows.md};
      }
    },

    const Main = styled.main`
      flex: 1;

    // Footer
    const Footer = styled.footer`
      background: ${props => props.theme.colors.primaryDark};
      color: ${props => props.theme.colors.textOnDark};
      margin-top: auto;

    const FooterContent = styled.div`
      max-width: 1400px;
      margin: 0 auto;
      padding: 60px 24px 30px;

    const FooterMain = styled.div`
      display: grid;
      grid-template-columns: 1fr 2fr;
      gap: 60px;
      @media (max-width: 968px) {
        grid-template-columns: 1fr;
        gap: 40px;
      }

    const FooterBrand = styled.div`
    const FooterLogo = styled.div`
      display: flex;
      align-items: center;
      gap: 12px;
      margin-bottom: 20px;
      ${Logolcon} {
        width: 40px;
        height: 40px;
        font-size: 20px;
      }
      span {
        font-family: ${props => props.theme.fonts.display};

```

```

    font-size: 18px;
    font-weight: 600;
    color: ${props => props.theme.colors.cream};
    text-transform: uppercase;
    letter-spacing: 0.05em;
  },
  const FooterDescription = styled.p`
    font-family: ${props => props.theme.fonts.body};
    font-size: 15px;
    line-height: 1.7;
    color: ${props => props.theme.colors.textMuted};
    max-width: 300px;
  },
  const FooterLinks = styled.div`
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 40px;
    @media (max-width: 768px) {
      grid-template-columns: repeat(2, 1fr);
    }
    @media (max-width: 480px) {
      grid-template-columns: 1fr;
    }
  },
  const FooterColumn = styled.div``
  const FooterTitle = styled.h4`
    font-family: ${props => props.theme.fonts.display};
    font-size: 14px;
    font-weight: 600;
    color: ${props => props.theme.colors.gold};
    text-transform: uppercase;
    letter-spacing: 0.1em;
    margin-bottom: 20px;
  },
  const FooterLink = styled(Link)`
    display: block;
    font-family: ${props => props.theme.fonts.ui};
    font-size: 14px;
    color: ${props => props.theme.colors.textMuted};
    margin-bottom: 12px;
    transition: all 0.3s ease;
    &:hover {
      color: ${props => props.theme.colors.gold};
      padding-left: 8px;
    }
  },
  const FooterText = styled.p`
    font-family: ${props => props.theme.fonts.ui};
    font-size: 14px;
    color: ${props => props.theme.colors.textMuted};
    margin-bottom: 8px;
  },
  const FooterDivider = styled.div`
    height: 1px;
    background: linear-gradient(
      90deg,
      transparent 0%,
      ${props => props.theme.colors.gold} 40 50%,
      transparent 100%
    );
    margin: 40px 0 20px;
  },
  const FooterBottom = styled.div`
    text-align: center;
  },
  const Copyright = styled.p`
    font-family: ${props => props.theme.fonts.ui};
    font-size: 13px;
    color: ${props => props.theme.colors.textMuted};
  },
  export default Layout

```

AuthContext.jsx

```

/**
 * Контекст аутентифікації
 */

```

```

import React, { createContext, useState, useContext, useEffect } from 'react'
import { authAPI } from '../utils/api'
const AuthContext = createContext(null)
export const useAuth = () => {
  const context = useContext(AuthContext)
  if (!context) {
    throw new Error('useAuth має використовуватися всередині AuthProvider')
  }
  return context
}
export const AuthProvider = ({ children }) => {
  const [user, setUser] = useState(null)
  const [loading, setLoading] = useState(true)
  useEffect(() => {
    // Перевірка сесії при завантаженні
    authAPI.getCurrentUser()
      .then(response => {
        setUser(response.data)
      })
      .catch((error) => {
        console.log('Користувач не авторизований')
        setUser(null)
      })
      .finally(() => setLoading(false))
  }, [])
  const login = async (username, password) => {
    try {
      const response = await authAPI.login({ username, password })
      const { user } = response.data
      setUser(user)
      return { success: true }
    } catch (error) {
      return {
        success: false,
        error: error.response?.data?.error || 'Помилка входу'
      }
    }
  }
  const register = async (username, email, password, role = 'client') => {
    try {
      const response = await authAPI.register({ username, email, password, role })
      const { user } = response.data
      setUser(user)
      return { success: true }
    } catch (error) {
      return {
        success: false,
        error: error.response?.data?.error || 'Помилка реєстрації'
      }
    }
  }
  const logout = async () => {
    try {
      await authAPI.logout()
    } catch (error) {
      console.error('Помилка виходу:', error)
    } finally {
      setUser(null)
    }
  }
  const value = {
    user,
    loading,
    login,
    register,
    logout,
    isAuthenticated: !!user,
    isClient: user?.role === 'client',
    isManager: user?.role === 'manager' || user?.role === 'admin',
    isAdmin: user?.role === 'admin',
    isGuest: !user || user?.role === 'guest',
  }
  return <AuthContext.Provider value={value}>{children}</AuthContext.Provider>
}

```

AboutPage.jsx

/**

```

* Сторінка "Про музей" - бароковий дизайн
*/
import React, { useState, useEffect } from 'react'
import styled, { keyframes } from 'styled-components'
import Layout from '../components/Layout'
import { useAuth } from '../context/AuthContext'
import { FiEdit2, FiSave, FiX, FiInfo, FiBook, FiTarget, FiMapPin, FiClock, FiPhone } from 'react-icons/fi'
import axios from 'axios'
const AboutPage = () => {
  const { isManager } = useAuth()
  const [content, setContent] = useState({
    title: 'Про наш музей',
    description: 'Ласкаво просимо до Художнього музею сучасного мистецтва! Ми зберігаємо та демонструємо унікальні експонати, які розповідають історію мистецтва нашого краю та світу.',
    history: 'Музей був заснований у 1950 році і з тих пір став одним з провідних культурних центрів регіону. За понад 70 років існування наша колекція зросла до більш ніж 10 000 експонатів.',
    mission: 'Наша місія - зберегти культурну спадщину, популяризувати мистецтво та робити його доступним для кожного відвідувача.'
  })
  const [isEditing, setIsEditing] = useState(false)
  const [editData, setEditData] = useState(content)
  const [scrollY, setScrollY] = useState(0)
  useEffect(() => {
    loadAbout()
  }, [])
  useEffect(() => {
    const handleScroll = () => setScrollY(window.scrollY)
    window.addEventListener('scroll', handleScroll, { passive: true })
    return () => window.removeEventListener('scroll', handleScroll)
  }, [])
  const loadAbout = async () => {
    try {
      const response = await axios.get('/api/about')
      if (response.data) {
        setContent(response.data)
        setEditData(response.data)
      }
    } catch (error) {
      console.error('Помилка завантаження:', error)
    }
  }
  const handleSave = async () => {
    try {
      await axios.put('/api/about', editData)
      setContent(editData)
      setIsEditing(false)
      alert('Зміни збережено!')
    } catch (error) {
      console.error('Помилка збереження:', error)
      alert('Помилка збереження')
    }
  }
  const handleCancel = () => {
    setEditData(content)
    setIsEditing(false)
  }
  return (
    <Layout>
      <HeroSection>
        <HeroBackground style={{ transform: `translateY(${scrollY * 0.5}px)` }} />
        <HeroOverlay />
        <HeroPattern />
        <HeroContent>
          <HeroIcon><FiInfo /></HeroIcon>
          <HeroTitle>Про музей</HeroTitle>
          <GoldLine />
          <HeroSubtitle>Історія та місія нашого закладу</HeroSubtitle>
        </HeroContent>
      </HeroSection>
      <Container>
        <ContentWrapper>
          <MainContent>
            <Header>
              {isEditing ? (
                <TitleInput
                  value={editData.title}
                  onChange={(e) => setEditData({ ...editData, title: e.target.value })}

```

```

        placeholder="Заголовок"
      />
    ) : (
      <Title>{content.title}</Title>
    )}
    {isManager && (
      <ButtonGroup>
        {isEditing ? (
          <>
            <SaveButton onClick={handleSave}>
              <FiSave /> Зберегти
            </SaveButton>
            <CancelButton onClick={handleCancel}>
              <FiX /> Скасувати
            </CancelButton>
          </>
        ) : (
          <EditButton onClick={() => setIsEditing(true)}>
            <FiEdit2 /> Редагувати
          </EditButton>
        )}
      </ButtonGroup>
    )}
  </Header>
  <Section>
    <SectionHeader>
      <SectionIcon><FiInfo /></SectionIcon>
      <SectionTitle>Опис</SectionTitle>
    </SectionHeader>
    {isEditing ? (
      <Textarea>
        value={editData.description}
        onChange={(e) => setEditData({ ...editData, description: e.target.value })}
        rows={4}
        placeholder="Опис музею..."
      </>
    ) : (
      <Text>{content.description}</Text>
    )}
  </Section>
  <Section>
    <SectionHeader>
      <SectionIcon><FiBook /></SectionIcon>
      <SectionTitle>Історія</SectionTitle>
    </SectionHeader>
    {isEditing ? (
      <Textarea>
        value={editData.history}
        onChange={(e) => setEditData({ ...editData, history: e.target.value })}
        rows={4}
        placeholder="Історія музею..."
      </>
    ) : (
      <Text>{content.history}</Text>
    )}
  </Section>
  <Section>
    <SectionHeader>
      <SectionIcon><FiTarget /></SectionIcon>
      <SectionTitle>Місія</SectionTitle>
    </SectionHeader>
    {isEditing ? (
      <Textarea>
        value={editData.mission}
        onChange={(e) => setEditData({ ...editData, mission: e.target.value })}
        rows={4}
        placeholder="Місія музею..."
      </>
    ) : (
      <Text>{content.mission}</Text>
    )}
  </Section>
</MainContent>
<Sidebar>
  <InfoCard>
    <InfoCardTitle>Контактна інформація</InfoCardTitle>
    <InfoItem>

```

```

    <InfoIcon><FiMapPin /></InfoIcon>
    <InfoText>
      <InfoLabel>Адреса</InfoLabel>
      <InfoValue>вул. Степана Бандери, 16</InfoValue>
      <InfoValue className="muted">Дніпро, 49000</InfoValue>
    </InfoText>
  </InfoItem>
  <InfoItem>
    <InfoIcon><FiClock /></InfoIcon>
    <InfoText>
      <InfoLabel>Години роботи</InfoLabel>
      <InfoValue>Вт-Нд: 10:00 - 18:00</InfoValue>
      <InfoValue className="muted">Понеділок - вихідний</InfoValue>
    </InfoText>
  </InfoItem>
  <InfoItem>
    <InfoIcon><FiPhone /></InfoIcon>
    <InfoText>
      <InfoLabel>Телефон</InfoLabel>
      <InfoValue>+38 (056) 745-24-17</InfoValue>
    </InfoText>
  </InfoItem>
</InfoCard>
<MapCard>
  <MapTitle><FiMapPin /> Ми на карті</MapTitle>
  <MapFrame
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2645.8!2d35.020473!3d48.466573!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x0%3A0x0!2zNDJCsDI3JzU5LjciTiAzNcKwMDEnMTMuNyJF!5e0!3m2!1suk!2sua!4v1"
    allowFullScreen=""
    loading="lazy"
    referrerPolicy="no-referrer-when-downgrade"
  />
  <MapLink
    href="https://www.google.com/maps?q=48.46657257754015,35.020472950543315"
    target="_blank"
    rel="noopener noreferrer"
  >
    Відкрити в Google Maps
  </MapLink>
</MapCard>
<StatsCard>
  <StatsTitle>Музей у цифрах</StatsTitle>
  <StatItem>
    <StatNumber>10 000+</StatNumber>
    <StatLabel>експонатів</StatLabel>
  </StatItem>
  <StatItem>
    <StatNumber>70+</StatNumber>
    <StatLabel>років історії</StatLabel>
  </StatItem>
  <StatItem>
    <StatNumber>50+</StatNumber>
    <StatLabel>екскурсій</StatLabel>
  </StatItem>
  <StatItem>
    <StatNumber>100 000+</StatNumber>
    <StatLabel>відвідувачів щороку</StatLabel>
  </StatItem>
</StatsCard>
</Sidebar>
</ContentWrapper>
</Container>
</Layout>
)
}
// Animations
const fadeIn = keyframes`
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
`

```

```

`
// Styled Components
const HeroSection = styled.div`
  position: relative;
  height: 400px;
  display: flex;
  align-items: center;
  justify-content: center;
  overflow: hidden;
  margin: -32px -24px 0;
  @media (max-width: 768px) {
    height: 300px;
  }
`

const HeroBackground = styled.div`
  position: absolute;
  top: -50%;
  left: 0;
  right: 0;
  bottom: -50%;
  background-image: url('https://images.unsplash.com/photo-1564399580075-5dfe19c205f3?w=1920&q=80');
  background-size: cover;
  background-position: center;
  will-change: transform;
`

const HeroOverlay = styled.div`
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background: linear-gradient(
    135deg,
    rgba(26, 26, 26, 0.92) 0%,
    rgba(114, 47, 55, 0.88) 50%,
    rgba(44, 44, 44, 0.92) 100%
  );
`

const HeroPattern = styled.div`
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-image: radial-gradient(circle at 20% 80%, rgba(212, 175, 55, 0.1) 0%, transparent 50%),
    radial-gradient(circle at 80% 20%, rgba(212, 175, 55, 0.08) 0%, transparent 50%);
  pointer-events: none;
`

const HeroContent = styled.div`
  position: relative;
  z-index: 2;
  text-align: center;
  color: white;
  padding: 0 20px;
`

const HeroIcon = styled.div`
  width: 80px;
  height: 80px;
  background: linear-gradient(135deg, rgba(212, 175, 55, 0.2), rgba(212, 175, 55, 0.1));
  border: 2px solid rgba(212, 175, 55, 0.4);
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 0 auto 24px;
  font-size: 32px;
  color: ${props => props.theme.colors.gold};
`

const HeroTitle = styled.h1`
  font-family: ${props => props.theme.fonts.display};
  font-size: 56px;
  font-weight: 600;
  margin: 0 0 20px 0;
  text-shadow: 0 4px 20px rgba(0, 0, 0, 0.3);
  letter-spacing: 2px;

  @media (max-width: 768px) {

```

```

    font-size: 38px;
  },
  const GoldLine = styled.div`
    width: 80px;
    height: 3px;
    background: ${props => props.theme.gradients.gold};
    margin: 0 auto 20px;
    border-radius: 2px;
  `,
  const HeroSubtitle = styled.p`
    font-family: ${props => props.theme.fonts.body};
    font-size: 18px;
    font-style: italic;
    opacity: 0.9;
  `,
  const Container = styled.div`
    max-width: 1200px;
    margin: 0 auto;
    padding: 60px 20px;
  `,
  const ContentWrapper = styled.div`
    display: grid;
    grid-template-columns: 1fr 360px;
    gap: 40px;
    animation: ${fadeIn} 0.6s ease-out;
    @media (max-width: 968px) {
      grid-template-columns: 1fr;
    }
  `,
  const MainContent = styled.div``
  const Header = styled.div`
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 40px;
    flex-wrap: wrap;
    gap: 20px;
  `,
  const Title = styled.h1`
    font-family: ${props => props.theme.fonts.display};
    font-size: 42px;
    color: ${props => props.theme.colors.primary};
    margin: 0;
  `,
  const TitleInput = styled.input`
    font-family: ${props => props.theme.fonts.display};
    font-size: 42px;
    color: ${props => props.theme.colors.primary};
    border: 2px solid ${props => props.theme.colors.gold};
    border-radius: ${props => props.theme.borderRadius.md};
    padding: 10px 16px;
    width: 100%;
    max-width: 500px;
    background: white;
    &:focus {
      outline: none;
      box-shadow: 0 0 4px rgba(212, 175, 55, 0.15);
    }
  `,
  const ButtonGroup = styled.div`
    display: flex;
    gap: 12px;
  `,
  const EditButton = styled.button`
    background: ${props => props.theme.gradients.gold};
    color: ${props => props.theme.colors.primaryDark};
    padding: 12px 24px;
    border-radius: ${props => props.theme.borderRadius.md};
    border: none;
    display: flex;
    align-items: center;
    gap: 8px;
    font-family: ${props => props.theme.fonts.ui};
    font-weight: 600;
    font-size: 14px;
    cursor: pointer;
  `

```

```

transition: all 0.3s;
box-shadow: ${props => props.theme.shadows.gold};
&:hover {
  transform: translateY(-2px);
  box-shadow: 0 6px 20px rgba(212, 175, 55, 0.4);
}
,

const SaveButton = styled.button`
background: ${props => props.theme.colors.success};
color: white;
padding: 12px 24px;
border-radius: ${props => props.theme.borderRadius.md};
border: none;
display: flex;
align-items: center;
gap: 8px;
font-family: ${props => props.theme.fonts.ui};
font-weight: 600;
font-size: 14px;
cursor: pointer;
transition: all 0.3s;

&:hover {
  background: #059669;
  transform: translateY(-2px);
}
,

const CancelButton = styled.button`
background: ${props => props.theme.colors.error};
color: white;
padding: 12px 24px;
border-radius: ${props => props.theme.borderRadius.md};
border: none;
display: flex;
align-items: center;
gap: 8px;
font-family: ${props => props.theme.fonts.ui};
font-weight: 600;
font-size: 14px;
cursor: pointer;
transition: all 0.3s;
&:hover {
  background: #a31f1f;
  transform: translateY(-2px);
}
,

const Section = styled.div`
margin-bottom: 40px;
background: ${props => props.theme.colors.surface};
padding: 32px;
border-radius: ${props => props.theme.borderRadius.lg};
border: 1px solid ${props => props.theme.colors.border};
box-shadow: ${props => props.theme.shadows.sm};
transition: all 0.3s;
&:hover {
  border-color: ${props => props.theme.colors.gold};
  box-shadow: ${props => props.theme.shadows.md};
}
,

const SectionHeader = styled.div`
display: flex;
align-items: center;
gap: 14px;
margin-bottom: 20px;
,

const SectionIcon = styled.div`
width: 44px;
height: 44px;
background: ${props => props.theme.colors.overlayGold};
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
font-size: 20px;
color: ${props => props.theme.colors.gold};
,

const SectionTitle = styled.h2`

```

```

font-family: ${props => props.theme.fonts.heading};
font-size: 26px;
color: ${props => props.theme.colors.primary};
margin: 0;
`,

const Text = styled.p`
font-family: ${props => props.theme.fonts.body};
font-size: 17px;
line-height: 1.9;
color: ${props => props.theme.colors.text};
margin: 0;
`,

const Textarea = styled.textarea`
width: 100%;
padding: 16px;
border: 2px solid ${props => props.theme.colors.border};
border-radius: ${props => props.theme.borderRadius.md};
font-family: ${props => props.theme.fonts.body};
font-size: 16px;
resize: vertical;
line-height: 1.7;
background: white;
transition: all 0.3s;
&:focus {
  outline: none;
  border-color: ${props => props.theme.colors.gold};
  box-shadow: 0 0 4px rgba(212, 175, 55, 0.15);
}
&::placeholder {
  color: ${props => props.theme.colors.textMuted};
  font-style: italic;
}
`,

const Sidebar = styled.div`
display: flex;
flex-direction: column;
gap: 24px;
`,

const InfoCard = styled.div`
background: ${props => props.theme.colors.surface};
padding: 28px;
border-radius: ${props => props.theme.borderRadius.lg};
border: 1px solid ${props => props.theme.colors.border};
box-shadow: ${props => props.theme.shadows.md};
`,

const InfoCardTitle = styled.h3`
font-family: ${props => props.theme.fonts.heading};
font-size: 20px;
color: ${props => props.theme.colors.primary};
margin: 0 0 24px 0;
padding-bottom: 16px;
border-bottom: 2px solid ${props => props.theme.colors.gold};
`,

const InfoItem = styled.div`
display: flex;
gap: 14px;
margin-bottom: 20px;
&:last-child {
  margin-bottom: 0;
}
`,

const InfoIcon = styled.div`
width: 40px;
height: 40px;
background: ${props => props.theme.colors.overlayGold};
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
font-size: 18px;
color: ${props => props.theme.colors.gold};
flex-shrink: 0;
`,

const InfoText = styled.div`
const InfoLabel = styled.div`
font-family: ${props => props.theme.fonts.ui};
font-size: 12px;

```

```

font-weight: 600;
text-transform: uppercase;
letter-spacing: 0.5px;
color: ${props => props.theme.colors.textMuted};
margin-bottom: 4px;
`,

const InfoValue = styled.div`
font-family: ${props => props.theme.fonts.body};
font-size: 15px;
color: ${props => props.theme.colors.text};
&.muted {
color: ${props => props.theme.colors.textMuted};
font-size: 13px;
margin-top: 2px;
}
`,

const StatsCard = styled.div`
background: ${props => props.theme.gradients.burgundy};
padding: 28px;
border-radius: ${props => props.theme.borderRadius.lg};
box-shadow: ${props => props.theme.shadows.elegant};
`,

const StatsTitle = styled.h3`
font-family: ${props => props.theme.fonts.heading};
font-size: 20px;
color: ${props => props.theme.colors.cream};
margin: 0 0 24px 0;
padding-bottom: 16px;
border-bottom: 2px solid ${props => props.theme.colors.gold};
`,

const StatItem = styled.div`
margin-bottom: 20px;
text-align: center;
padding: 16px;
background: rgba(255, 255, 255, 0.1);
border-radius: ${props => props.theme.borderRadius.md};
&:last-child {
margin-bottom: 0;
}
`,

const StatNumber = styled.div`
font-family: ${props => props.theme.fonts.display};
font-size: 32px;
font-weight: 600;
color: ${props => props.theme.colors.gold};
margin-bottom: 4px;
`,

const StatLabel = styled.div`
font-family: ${props => props.theme.fonts.body};
font-size: 14px;
color: rgba(255, 255, 255, 0.8);
`,

const MapCard = styled.div`
background: ${props => props.theme.colors.surface};
padding: 24px;
border-radius: ${props => props.theme.borderRadius.lg};
border: 1px solid ${props => props.theme.colors.border};
box-shadow: ${props => props.theme.shadows.md};
`,

const MapTitle = styled.h3`
display: flex;
align-items: center;
gap: 10px;
font-family: ${props => props.theme.fonts.heading};
font-size: 18px;
color: ${props => props.theme.colors.primary};
margin: 0 0 16px 0;
svg {
color: ${props => props.theme.colors.gold};
}
`,

const MapFrame = styled.iframe`
width: 100%;
height: 250px;
border: none;
border-radius: ${props => props.theme.borderRadius.md};
margin-bottom: 12px;

```

```

const MapLink = styled.a`
  display: inline-flex;
  align-items: center;
  gap: 6px;
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  font-weight: 600;
  color: ${props => props.theme.colors.secondary};
  transition: all 0.2s;
  &:hover {
    color: ${props => props.theme.colors.gold};
  }
`

```

```
export default AboutPage
```

AdminDashboard.jsx

```

/**
 * Адмін-панель - вишуканий бароковий дизайн
 * Управління користувачами та детальні звіти
 */
import React, { useState, useEffect } from 'react'
import styled, { keyframes } from 'styled-components'
import Layout from '../components/Layout'
import { adminAPI, newsAPI, excursionsAPI } from '../utils/api'
import { useAuth } from '../context/AuthContext'
import { useNavigate } from 'react-router-dom'
import {
  FiUsers, FiDollarSign, FiTrendingUp, FiBarChart2,
  FiFileText, FiCalendar, FiStar, FiMessageSquare,
  FiEdit2, FiTrash2, FiShield, FiUserCheck, FiUserX,
  FiChevronRight, FiRefreshCw, FiDownload, FiFilter,
  FiEye, FiClock, FiAward
} from 'react-icons/fi'
const AdminDashboard = () => {
  const { user, isAdmin } = useAuth()
  const navigate = useNavigate()
  const [activeTab, setActiveTab] = useState('overview')
  const [loading, setLoading] = useState(true)
  const [dashboard, setDashboard] = useState(null)
  const [users, setUsers] = useState([])
  const [financialReport, setFinancialReport] = useState(null)
  const [usersReport, setUsersReport] = useState(null)
  const [excursionsReport, setExcursionsReport] = useState(null)
  const [period, setPeriod] = useState('month')
  const [userFilter, setUserFilter] = useState('')
  const [roleFilter, setRoleFilter] = useState('')
  useEffect(() => {
    if (!isAdmin) {
      navigate('/')
      return
    }
    loadDashboard()
  }, [isAdmin])
  useEffect(() => {
    if (activeTab === 'users') {
      loadUsers()
    } else if (activeTab === 'financial') {
      loadFinancialReport()
    } else if (activeTab === 'users-report') {
      loadUsersReport()
    } else if (activeTab === 'excursions-report') {
      loadExcursionsReport()
    }
  }, [activeTab, period])
  const loadDashboard = async () => {
    try {
      setLoading(true)
      const response = await adminAPI.getDashboard()
      setDashboard(response.data)
    } catch (error) {
      console.error('Помилка завантаження:', error)
    } finally {
      setLoading(false)
    }
  }
  const loadUsers = async () => {

```

```

try {
  const response = await adminAPI.getUsers({
    search: userFilter,
    role: roleFilter
  })
  setUsers(response.data.users)
} catch (error) {
  console.error('Помилка завантаження користувачів:', error)
}
}

const loadFinancialReport = async () => {
  try {
    const response = await adminAPI.getFinancialReport(period)
    setFinancialReport(response.data)
  } catch (error) {
    console.error('Помилка завантаження звіту:', error)
  }
}

const loadUsersReport = async () => {
  try {
    const response = await adminAPI.getUsersReport(period)
    setUsersReport(response.data)
  } catch (error) {
    console.error('Помилка завантаження звіту:', error)
  }
}

const loadExcursionsReport = async () => {
  try {
    const response = await adminAPI.getExcursionsReport(period)
    setExcursionsReport(response.data)
  } catch (error) {
    console.error('Помилка завантаження звіту:', error)
  }
}

const handleChangeRole = async (userId, newRole) => {
  if (!window.confirm(`Змінити роль на "${newRole}"?`)) return
  try {
    await adminAPI.changeUserRole(userId, newRole)
    loadUsers()
  } catch (error) {
    alert(error.response?.data?.error || 'Помилка')
  }
}

const handleDeleteUser = async (userId, username) => {
  if (!window.confirm(`Видалити користувача "${username}"? Цю дію не можна скасувати.`)) return
  try {
    await adminAPI.deleteUser(userId)
    loadUsers()
  } catch (error) {
    alert(error.response?.data?.error || 'Помилка')
  }
}

const handleToggleBlock = async (userId, currentRole) => {
  const isBlocked = currentRole === 'guest'
  try {
    await adminAPI.toggleUserBlock(userId, !isBlocked)
    loadUsers()
  } catch (error) {
    alert(error.response?.data?.error || 'Помилка')
  }
}

const tabs = [
  { id: 'overview', label: 'Огляд', icon: FiBarChart2 },
  { id: 'users', label: 'Користувачі', icon: FiUsers },
  { id: 'financial', label: 'Фінанси', icon: FiDollarSign },
  { id: 'users-report', label: 'Звіт: Користувачі', icon: FiUserCheck },
  { id: 'excursions-report', label: 'Звіт: Експерсії', icon: FiCalendar },
]

const formatMoney = (amount) => {
  return new Intl.NumberFormat('uk-UA', {
    minimumFractionDigits: 0,
    maximumFractionDigits: 0
  }).format(amount) + ' грн'
}

const getRoleBadge = (role) => {
  const styles = {
    admin: { bg: 'linear-gradient(135deg, #D4AF37, #B8960C)', color: '#1A1A1A' },
  }

```

```

manager: { bg: '#722F37', color: '#F5F0E6' },
client: { bg: '#2C2C2C', color: '#F5F0E6' },
guest: { bg: '#999', color: '#fff' }
}
const labels = { admin: 'Адмін', manager: 'Менеджер', client: 'Клієнт', guest: 'Заблоковано' }
return { ...styles[role], label: labels[role] }
}
return (
  <Layout>
    <PageWrapper>
      <OrnamentTop />
      <Header>
        <HeaderContent>
          <HeaderLeft>
            <AdminBadge>
              <FiShield />
              <span>Адміністратор</span>
            </AdminBadge>
            <Title>Панель керування</Title>
            <Subtitle>Вітаємо, {user?.username}</Subtitle>
          </HeaderLeft>
          <HeaderRight>
            <RefreshButton onClick={loadDashboard}>
              <FiRefreshCw />
              Оновити
            </RefreshButton>
          </HeaderRight>
        </HeaderContent>
        <GoldDivider />
      </Header>
      <TabsContainer>
        <TabList>
          {tabs.map(tab => (
            <TabButton
              key={tab.id}
              $active={activeTab === tab.id}
              onClick={() => setActiveTab(tab.id)}
            >
              <tab.icon />
              <span>{tab.label}</span>
            </TabButton>
          ))}
        </TabList>
      </TabsContainer>
      <Content>
        {activeTab === 'overview' && dashboard && (
          <OverviewSection>
            <StatsGrid>
              <StatCard $accent="gold">
                <StatIcon $accent="gold"><FiUsers /></StatIcon>
                <StatContent>
                  <StatValue>{dashboard.users.total}</StatValue>
                  <StatLabel>Користувачів</StatLabel>
                  <StatChange $positive>+{dashboard.users.new_month} за місяць</StatChange>
                </StatContent>
                <StatDecor />
              </StatCard>
              <StatCard $accent="burgundy">
                <StatIcon $accent="burgundy"><FiDollarSign /></StatIcon>
                <StatContent>
                  <StatValue>{formatMoney(dashboard.finance.total_revenue)}</StatValue>
                  <StatLabel>Загальний дохід</StatLabel>
                  <StatChange $positive>{formatMoney(dashboard.finance.month_revenue)} за місяць</StatChange>
                </StatContent>
                <StatDecor />
              </StatCard>
              <StatCard $accent="green">
                <StatIcon $accent="green"><FiCalendar /></StatIcon>
                <StatContent>
                  <StatValue>{dashboard.tickets.total}</StatValue>
                  <StatLabel>Проданих квитків</StatLabel>
                  <StatChange $positive>+{dashboard.tickets.month} за місяць</StatChange>
                </StatContent>
                <StatDecor />
              </StatCard>
              <StatCard $accent="blue">
                <StatIcon $accent="blue"><FiStar /></StatIcon>

```

```

    <StatContent>
      <StatValue>{dashboard.content.avg_rating}</StatValue>
      <StatLabel>Середня оцінка</StatLabel>
      <StatChange>{dashboard.content.reviews} відгуків</StatChange>
    </StatContent>
    <StatDecor />
  </StatCard>
</StatsGrid>
<SecondaryStats>
  <SecondaryCard>
    <SecondaryIcon><FiFileText /></SecondaryIcon>
    <SecondaryInfo>
      <SecondaryValue>{dashboard.content.excursions}</SecondaryValue>
      <SecondaryLabel>Екскурсій</SecondaryLabel>
    </SecondaryInfo>
  </SecondaryCard>
  <SecondaryCard>
    <SecondaryIcon><FiFileText /></SecondaryIcon>
    <SecondaryInfo>
      <SecondaryValue>{dashboard.content.news_total}</SecondaryValue>
      <SecondaryLabel>Новин</SecondaryLabel>
    </SecondaryInfo>
  </SecondaryCard>
  <SecondaryCard>
    <SecondaryIcon><FiMessageSquare /></SecondaryIcon>
    <SecondaryInfo>
      <SecondaryValue>{dashboard.activity.active_chats}</SecondaryValue>
      <SecondaryLabel>Активних чатів</SecondaryLabel>
    </SecondaryInfo>
  </SecondaryCard>
  <SecondaryCard>
    <SecondaryIcon><FiTrendingUp /></SecondaryIcon>
    <SecondaryInfo>
      <SecondaryValue>+{dashboard.users.new_week}</SecondaryValue>
      <SecondaryLabel>Нових за тиждень</SecondaryLabel>
    </SecondaryInfo>
  </SecondaryCard>
</SecondaryStats>
</OverviewSection>
)}
{activeTab === 'users' && (
  <UsersSection>
    <SectionHeader>
      <SectionTitle>
        <FiUsers />
        Управління користувачами
      </SectionTitle>
      <Filters>
        <FilterInput
          type="text"
          placeholder="Пошук..."
          value={userFilter}
          onChange={(e) => setUserFilter(e.target.value)}
          onKeyDown={(e) => e.key === 'Enter' && loadUsers()}
        />
        <FilterSelect
          value={roleFilter}
          onChange={(e) => { setRoleFilter(e.target.value); setTimeout(loadUsers, 100) }}
        >
          <option value="">Всі ролі</option>
          <option value="admin">Адміни</option>
          <option value="manager">Менеджери</option>
          <option value="client">Клієнти</option>
          <option value="guest">Заблоковані</option>
        </FilterSelect>
      </Filters>
    </SectionHeader>
    <UsersTable>
      <thead>
        <tr>
          <th>Користувач</th>
          <th>Email</th>
          <th>Роль</th>
          <th>Квитки</th>
          <th>Витрачено</th>
          <th>Дата реєстрації</th>
          <th>Дії</th>
        </tr>
      </thead>
    </UsersTable>
  </UsersSection>
)

```

```

</tr>
</thead>
<tbody>
{users.map(u => {
  const badge = getRoleBadge(u.role)
  return (
    <tr key={u.id}>
      <td>
        <UserCell>
          <UserAvatar $role={u.role}>
            {u.username[0].toUpperCase()}
          </UserAvatar>
          <span>{u.username}</span>
        </UserCell>
      </td>
      <td>{u.email}</td>
      <td>
        <RoleBadge style={{ background: badge.bg, color: badge.color }}>
          {badge.label}
        </RoleBadge>
      </td>
      <td>{u.tickets_count}</td>
      <td>{formatMoney(u.total_spent)}</td>
      <td>{new Date(u.created_at).toLocaleDateString('uk-UA')}</td>
      <td>
        <ActionButtons>
          {u.role !== 'admin' && (
            <>
              <ActionButton
                title="Зробити менеджером"
                onClick={() => handleChangeRole(u.id, 'manager')}
                disabled={u.role === 'manager'}
              >
                <FiUserCheck />
              </ActionButton>
              <ActionButton
                title={u.role === 'guest' ? 'Розблокувати' : 'Заблокувати'}
                onClick={() => handleToggleBlock(u.id, u.role)}
                $danger={u.role !== 'guest'}
              >
                {u.role === 'guest' ? <FiUserCheck /> : <FiUserX />}
              </ActionButton>
              <ActionButton
                title="Видалити"
                onClick={() => handleDeleteUser(u.id, u.username)}
                $danger
              >
                <FiTrash2 />
              </ActionButton>
            </>
          )}
        </ActionButtons>
      </td>
    </tr>
  )
})}
</tbody>
</UsersTable>
</UsersSection>
)}
{activeTab === 'financial' && (
  <ReportSection>
    <SectionHeader>
      <SectionTitle>
        <FiDollarSign />
        Фінансовий звіт
      </SectionTitle>
      <PeriodSelector>
        {[ 'day', 'week', 'month', 'year', 'all' ].map(p => (
          <PeriodButton
            key={p}
            $active={period === p}
            onClick={() => setPeriod(p)}
          >
            {p === 'day' && 'День'}
            {p === 'week' && 'Тиждень'}
            {p === 'month' && 'Місяць'}
          </PeriodButton>
        ))}
      </PeriodSelector>
    </SectionHeader>
    <Table>
      <thead>
        <tr>
          <th>Період</th>
          <th>Відомості</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>{period}</td>
          <td>{formatMoney(total_spent)}</td>
        </tr>
      </tbody>
    </Table>
  </ReportSection>
)}

```

```

    {p === 'year' && 'Рік'}
    {p === 'all' && 'Весь час'}
  </PeriodButton>
  )))
</PeriodSelector>
</SectionHeader>
{financialReport && (
  <>
    <FinancialSummary>
      <SummaryCard $main>
        <SummaryIcon><FiDollarSign /></SummaryIcon>
        <SummaryContent>
          <SummaryValue>{formatMoney(financialReport.summary.total_revenue)}</SummaryValue>
          <SummaryLabel>Загальний дохід</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
      <SummaryCard>
        <SummaryContent>
          <SummaryValue>{financialReport.summary.total_tickets}</SummaryValue>
          <SummaryLabel>Продано квитків</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
      <SummaryCard>
        <SummaryContent>
          <SummaryValue>{formatMoney(financialReport.summary.avg_ticket_price)}</SummaryValue>
          <SummaryLabel>Середній чек</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
      <SummaryCard>
        <SummaryContent>
          <SummaryValue>{financialReport.summary.cancellation_rate}%</SummaryValue>
          <SummaryLabel>Відміни</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
    </FinancialSummary>
    <ReportGrid>
      <ReportCard>
        <ReportCardHeader>
          <h3>Дохід по екскурсіях</h3>
        </ReportCardHeader>
        <ReportTable>
          <thead>
            <tr>
              <th>Екскурсія</th>
              <th>Дохід</th>
              <th>Квитків</th>
              <th>Відвідувачів</th>
            </tr>
          </thead>
          <tbody>
            {financialReport.revenue_by_excursion.slice(0, 8).map(e => (
              <tr key={e.excursion_id}>
                <td>{e.title}</td>
                <td><strong>{formatMoney(e.revenue)}</strong></td>
                <td>{e.tickets_count}</td>
                <td>{e.visitors}</td>
              </tr>
            ))}
          </tbody>
        </ReportTable>
      </ReportCard>
      <ReportCard>
        <ReportCardHeader>
          <h3>Топ покупці</h3>
        </ReportCardHeader>
        <TopCustomersList>
          {financialReport.top_customers.slice(0, 5).map((c, i) => (
            <TopCustomer key={c.id}>
              <TopRank>{i + 1}</TopRank>
              <TopInfo>
                <TopName>{c.username}</TopName>
                <TopEmail>{c.email}</TopEmail>
              </TopInfo>
              <TopSpent>{formatMoney(c.total_spent)}</TopSpent>
            </TopCustomer>
          ))}
        </TopCustomersList>
      </ReportCard>
    </ReportGrid>
  )}
)

```

```

        </ReportCard>
      </ReportGrid>
    </>
  })
</ReportSection>
})
{activeTab === 'users-report' && usersReport && (
  <ReportSection>
    <SectionHeader>
      <SectionTitle>
        <FiUsers />
        Звіт по користувачах
      </SectionTitle>
      <PeriodSelector>
        {[ 'week', 'month', 'year', 'all' ].map(p => (
          <PeriodButton
            key={p}
            $active={period === p}
            onClick={() => setPeriod(p)}
          >
            {p === 'week' && 'Тиждень'}
            {p === 'month' && 'Місяць'}
            {p === 'year' && 'Рік'}
            {p === 'all' && 'Весь час'}
          </PeriodButton>
        ))}
      </PeriodSelector>
    </SectionHeader>
    <FinancialSummary>
      <SummaryCard $main>
        <SummaryIcon><FiUsers /></SummaryIcon>
        <SummaryContent>
          <SummaryValue>{usersReport.summary.total_users}</SummaryValue>
          <SummaryLabel>Всього користувачів</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
      <SummaryCard>
        <SummaryContent>
          <SummaryValue>+{usersReport.summary.new_users}</SummaryValue>
          <SummaryLabel>Нових</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
      <SummaryCard>
        <SummaryContent>
          <SummaryValue>{usersReport.summary.active_users}</SummaryValue>
          <SummaryLabel>Активних</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
      <SummaryCard>
        <SummaryContent>
          <SummaryValue>{usersReport.summary.activity_rate}%</SummaryValue>
          <SummaryLabel>Активність</SummaryLabel>
        </SummaryContent>
      </SummaryCard>
    </FinancialSummary>
    <RoleDistribution>
      <h3>Розподіл по ролях</h3>
      <RoleBars>
        {Object.entries(usersReport.by_role).map(([role, count]) => {
          const badge = getRoleBadge(role)
          const percent = (count / usersReport.summary.total_users * 100).toFixed(0)
          return (
            <RoleBar key={role}>
              <RoleBarLabel>{badge.label}</RoleBarLabel>
              <RoleBarTrack>
                <RoleBarFill style={{ width: `${percent}%`, background: badge.bg }} />
              </RoleBarTrack>
              <RoleBarValue>{count} ({percent}%</RoleBarValue>
            </RoleBar>
          )
        })}
      </RoleBars>
    </RoleDistribution>
  </ReportSection>
})
{activeTab === 'excursions-report' && excursionsReport && (
  <ReportSection>

```

```

<SectionHeader>
  <SectionTitle>
    <FiCalendar />
    Звіт по екскурсіях
  </SectionTitle>
  <PeriodSelector>
    {[ 'week', 'month', 'year', 'all' ].map(p => (
      <PeriodButton
        key={p}
        $active={period === p}
        onClick={() => setPeriod(p)}
      >
        {p === 'week' && 'Тиждень'}
        {p === 'month' && 'Місяць'}
        {p === 'year' && 'Рік'}
        {p === 'all' && 'Весь час'}
      </PeriodButton>
    ))}
  </PeriodSelector>
</SectionHeader>
<TrendCard $positive={excursionsReport.trends.change_percent >= 0}>
  <TrendIcon>
    <FiTrendingUp />
  </TrendIcon>
  <TrendContent>
    <TrendValue>
      {excursionsReport.trends.change_percent >= 0 ? '+' : ''}{excursionsReport.trends.change_percent}%
    </TrendValue>
    <TrendLabel>
      порівняно з попереднім періодом
    </TrendLabel>
  </TrendContent>
  <TrendCompare>
    <span>Зараз: {formatMoney(excursionsReport.trends.current_revenue)}</span>
    <span>Раніше: {formatMoney(excursionsReport.trends.previous_revenue)}</span>
  </TrendCompare>
</TrendCard>
<ExcursionsTable>
  <thead>
    <tr>
      <th>Екскурсія</th>
      <th>Категорія</th>
      <th>Ціна</th>
      <th>Продано</th>
      <th>Відвідувачів</th>
      <th>Дохід</th>
      <th>Рейтинг</th>
    </tr>
  </thead>
  <tbody>
    {excursionsReport.excursions.map(e => (
      <tr key={e.id}>
        <td><strong>{e.title}</strong></td>
        <td>
          <CategoryBadge $category={e.category}>
            {e.category || 'інше'}
          </CategoryBadge>
        </td>
        <td>{e.base_price} грн</td>
        <td>{e.tickets_sold}</td>
        <td>{e.total_visitors}</td>
        <td><strong>{formatMoney(e.revenue)}</strong></td>
        <td>
          <RatingCell>
            <FiStar />
            {e.avg_rating} ({e.reviews_count})
          </RatingCell>
        </td>
      </tr>
    ))}
  </tbody>
</ExcursionsTable>
</ReportSection>
)}
</Content>
<OrnamentBottom />
</PageWrapper>

```

```

    </Layout>
  )
}
// Animations
const fadeIn = keyframes`
  from { opacity: 0; transform: translateY(20px); }
  to { opacity: 1; transform: translateY(0); }
`

const shimmer = keyframes`
  0% { background-position: -200% 0; }
  100% { background-position: 200% 0; }
`

// Styled Components
const PageWrapper = styled.div`
  max-width: 1400px;
  margin: 0 auto;
  padding: 40px 24px;
  position: relative;
  min-height: 100vh;
  background: linear-gradient(180deg,
    rgba(245, 240, 230, 0.02) 0%,
    rgba(212, 175, 55, 0.03) 50%,
    rgba(245, 240, 230, 0.02) 100%
  );

  const OrnamentTop = styled.div`
    position: absolute;
    top: 0;
    left: 50%;
    transform: translateX(-50%);
    width: 200px;
    height: 40px;
    background: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 200 40'%3E%3Cpath
d='M100 35 L90 25 Q80 35 70 25 Q60 15 50 25 L40 35 M100 35 L110 25 Q120 35 130 25 Q140 15 150 25 L160 35'
stroke='%23D4AF37' stroke-width='1.5' fill='none' opacity='0.6'/%3E%3C/svg%3E") no-repeat center;

  const OrnamentBottom = styled(OrnamentTop)`
    top: auto;
    bottom: 0;
    transform: translateX(-50%) rotate(180deg);

  const Header = styled.header`
    margin-bottom: 32px;
    animation: ${fadeIn} 0.5s ease-out;

  const HeaderContent = styled.div`
    display: flex;
    justify-content: space-between;
    align-items: flex-start;
    margin-bottom: 24px;

  const HeaderLeft = styled.div``
  const HeaderRight = styled.div``
  const AdminBadge = styled.div`
    display: inline-flex;
    align-items: center;
    gap: 8px;
    padding: 8px 16px;
    background: ${props => props.theme.gradients.gold};
    color: ${props => props.theme.colors.primaryDark};
    border-radius: 20px;
    font-family: ${props => props.theme.fonts.ui};
    font-size: 12px;
    font-weight: 700;
    text-transform: uppercase;
    letter-spacing: 1px;
    margin-bottom: 16px;
    svg {
      font-size: 14px;
    }
  `

  const Title = styled.h1`
    font-family: ${props => props.theme.fonts.display};
    font-size: 42px;
    color: ${props => props.theme.colors.text};
    margin: 0 0 8px;
    position: relative;

```

```

&::after {
  content: "";
  position: absolute;
  bottom: -8px;
  left: 0;
  width: 60px;
  height: 3px;
  background: ${props => props.theme.gradients.gold};
  border-radius: 2px;
}

const Subtitle = styled.p`
  font-family: ${props => props.theme.fonts.body};
  font-size: 16px;
  color: ${props => props.theme.colors.textMuted};
  margin: 0;
`

const RefreshButton = styled.button`
  display: flex;
  align-items: center;
  gap: 8px;
  padding: 12px 24px;
  background: ${props => props.theme.colors.surface};
  border: 1px solid ${props => props.theme.colors.borderGold};
  border-radius: ${props => props.theme.borderRadius.md};
  color: ${props => props.theme.colors.gold};
  font-family: ${props => props.theme.fonts.ui};
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s ease;
  &:hover {
    background: ${props => props.theme.colors.overlayGold};
  }
`

const GoldDivider = styled.div`
  height: 2px;
  background: ${props => props.theme.gradients.gold};
  border-radius: 1px;
`

const TabsContainer = styled.div`
  margin-bottom: 32px;
  overflow-x: auto;
`

const TabsList = styled.div`
  display: flex;
  gap: 8px;
  background: ${props => props.theme.colors.surface};
  padding: 8px;
  border-radius: ${props => props.theme.borderRadius.lg};
  border: 1px solid ${props => props.theme.colors.border};
`

const TabButton = styled.button`
  display: flex;
  align-items: center;
  gap: 8px;
  padding: 12px 20px;
  border-radius: ${props => props.theme.borderRadius.md};
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  font-weight: 600;
  white-space: nowrap;
  transition: all 0.3s ease;
  ${props => props.$active ? `
    background: ${props.theme.gradients.gold};
    color: ${props.theme.colors.primaryDark};
  ` : `
    background: transparent;
    color: ${props.theme.colors.textSecondary};
    &:hover {
      background: ${props.theme.colors.backgroundDark};
      color: ${props.theme.colors.text};
    }
  `}
  svg {
    font-size: 16px;
  }
`

```

```

const Content = styled.main`
  animation: ${fadeIn} 0.5s ease-out 0.1s both;
`

// Overview Section
const OverviewSection = styled.div`
const StatsGrid = styled.div`
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 24px;
  margin-bottom: 24px;
  @media (max-width: 1024px) {
    grid-template-columns: repeat(2, 1fr);
  }
  @media (max-width: 600px) {
    grid-template-columns: 1fr;
  }
`

const StatCard = styled.div`
  position: relative;
  background: linear-gradient(145deg, ${props => props.theme.colors.surface}, rgba(255, 255, 255, 0.98));
  padding: 28px;
  border-radius: ${props => props.theme.borderRadius.lg};
  border: 1px solid ${props => props.theme.colors.border};
  display: flex;
  gap: 20px;
  overflow: hidden;
  transition: all 0.3s ease;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.05);
  &::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 4px;
    height: 100%;
    background: ${props => props.theme.gradients.gold};
    opacity: 0;
    transition: opacity 0.3s ease;
  }
  &:hover {
    transform: translateY(-4px);
    box-shadow: ${props => props.theme.shadows.lg}, 0 0 30px rgba(212, 175, 55, 0.1);
    border-color: ${props => props.theme.colors.borderGold};
    &::before {
      opacity: 1;
    }
  }
`

const StatDecor = styled.div`
  position: absolute;
  top: 0;
  right: 0;
  width: 100px;
  height: 100px;
  background: radial-gradient(circle at top right, ${props => props.theme.colors.gold}10, transparent 70%);
`

const StatIcon = styled.div`
  width: 56px;
  height: 56px;
  border-radius: ${props => props.theme.borderRadius.md};
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 24px;
  flex-shrink: 0;
  ${props => props.$accent === 'gold' && `
    background: linear-gradient(135deg, ${props.theme.colors.gold}20, ${props.theme.colors.gold}05);
    color: ${props.theme.colors.gold};
  `}
  ${props => props.$accent === 'burgundy' && `
    background: ${props.theme.colors.secondary}15;
    color: ${props.theme.colors.secondary};
  `}
  ${props => props.$accent === 'green' && `
    background: ${props.theme.colors.success}15;
    color: ${props.theme.colors.success};
  `}
`

```

```

    ${props => props.$accent === 'blue' && `
      background: ${props.theme.colors.info}15;
      color: ${props.theme.colors.info};
    `}

const StatContent = styled.div`
  flex: 1;

const StatValue = styled.div`
  font-family: ${props => props.theme.fonts.display};
  font-size: 32px;
  font-weight: 700;
  color: ${props => props.theme.colors.text};
  line-height: 1;
  margin-bottom: 4px;

const StatLabel = styled.div`
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  color: ${props => props.theme.colors.textMuted};
  margin-bottom: 8px;

const StatChange = styled.div`
  font-family: ${props => props.theme.fonts.ui};
  font-size: 13px;
  color: ${props => props.$positive ? props.theme.colors.success : props.theme.colors.textMuted};

const SecondaryStats = styled.div`
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 16px;
  @media (max-width: 768px) {
    grid-template-columns: repeat(2, 1fr);
  }

const SecondaryCard = styled.div`
  background: ${props => props.theme.colors.backgroundDark};
  padding: 20px;
  border-radius: ${props => props.theme.borderRadius.md};
  display: flex;
  align-items: center;
  gap: 16px;

const SecondaryIcon = styled.div`
  width: 40px;
  height: 40px;
  background: ${props => props.theme.colors.surface};
  border-radius: ${props => props.theme.borderRadius.sm};
  display: flex;
  align-items: center;
  justify-content: center;
  color: ${props => props.theme.colors.gold};

const SecondaryInfo = styled.div`
const SecondaryValue = styled.div`
  font-family: ${props => props.theme.fonts.display};
  font-size: 24px;
  font-weight: 700;
  color: ${props => props.theme.colors.text};

const SecondaryLabel = styled.div`
  font-family: ${props => props.theme.fonts.ui};
  font-size: 12px;
  color: ${props => props.theme.colors.textMuted};

// Users Section
const UsersSection = styled.div`
const SectionHeader = styled.div`
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 24px;
  flex-wrap: wrap;
  gap: 16px;

const SectionTitle = styled.h2`
  display: flex;

```

```

align-items: center;
gap: 12px;
font-family: ${props => props.theme.fonts.heading};
font-size: 24px;
color: ${props => props.theme.colors.text};
margin: 0;
svg {
  color: ${props => props.theme.colors.gold};
}
},

const Filters = styled.div`
display: flex;
gap: 12px;

const FilterInput = styled.input`
padding: 10px 16px;
border: 1px solid ${props => props.theme.colors.border};
border-radius: ${props => props.theme.borderRadius.md};
font-family: ${props => props.theme.fonts.ui};
font-size: 14px;
background: ${props => props.theme.colors.surface};
min-width: 200px;
&:focus {
  outline: none;
  border-color: ${props => props.theme.colors.gold};
}

const FilterSelect = styled.select`
padding: 10px 16px;
border: 1px solid ${props => props.theme.colors.border};
border-radius: ${props => props.theme.borderRadius.md};
font-family: ${props => props.theme.fonts.ui};
font-size: 14px;
background: ${props => props.theme.colors.surface};
cursor: pointer;
&:focus {
  outline: none;
  border-color: ${props => props.theme.colors.gold};
}

const UsersTable = styled.table`
width: 100%;
border-collapse: collapse;
background: ${props => props.theme.colors.surface};
border-radius: ${props => props.theme.borderRadius.lg};
overflow: hidden;
border: 1px solid ${props => props.theme.colors.border};
th, td {
  padding: 16px 20px;
  text-align: left;
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
}
th {
  background: ${props => props.theme.colors.backgroundDark};
color: ${props => props.theme.colors.textSecondary};
font-weight: 600;
text-transform: uppercase;
font-size: 11px;
letter-spacing: 0.5px;
}
tbody tr {
  border-top: 1px solid ${props => props.theme.colors.border};
transition: background 0.2s ease;
  &:hover {
    background: ${props => props.theme.colors.overlayGold};
  }
}

const UserCell = styled.div`
display: flex;
align-items: center;
gap: 12px;

const UserAvatar = styled.div`
width: 36px;
height: 36px;

```

```

border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
font-weight: 700;
font-size: 14px;
${props => props.$role === 'admin' && `
  background: ${props.theme.gradients.gold};
  color: ${props.theme.colors.primaryDark};
`}
${props => props.$role === 'manager' && `
  background: ${props.theme.colors.secondary};
  color: ${props.theme.colors.cream};
`}
${props => (props.$role === 'client' || props.$role === 'guest') && `
  background: ${props.theme.colors.primary};
  color: ${props.theme.colors.cream};
`}
`

const RoleBadge = styled.span`
padding: 4px 10px;
border-radius: 12px;
font-size: 11px;
font-weight: 700;
text-transform: uppercase;
letter-spacing: 0.5px;
`

const ActionButtons = styled.div`
display: flex;
gap: 8px;
`

const ActionButton = styled.button`
width: 32px;
height: 32px;
border-radius: ${props => props.theme.borderRadius.sm};
display: flex;
align-items: center;
justify-content: center;
background: ${props => props.theme.colors.backgroundDark};
color: ${props => props.$danger ? props.theme.colors.error : props.theme.colors.textSecondary};
transition: all 0.2s ease;
&:hover:not(:disabled) {
  background: ${props => props.$danger ? props.theme.colors.error : props.theme.colors.gold};
  color: ${props => props.$danger ? '#fff' : props.theme.colors.primaryDark};
}
&:disabled {
  opacity: 0.3;
  cursor: not-allowed;
}
`

// Report Section
const ReportSection = styled.div``
const PeriodSelector = styled.div`
display: flex;
gap: 8px;
background: ${props => props.theme.colors.surface};
padding: 4px;
border-radius: ${props => props.theme.borderRadius.md};
border: 1px solid ${props => props.theme.colors.border};
`

const PeriodButton = styled.button`
padding: 8px 16px;
border-radius: ${props => props.theme.borderRadius.sm};
font-family: ${props => props.theme.fonts.ui};
font-size: 13px;
font-weight: 600;
transition: all 0.2s ease;
${props => props.$active ? `
  background: ${props.theme.colors.gold};
  color: ${props.theme.colors.primaryDark};
` : `
  background: transparent;
  color: ${props.theme.colors.textSecondary};
  &:hover {
    color: ${props.theme.colors.text};
  }
`}
`

```

```

const FinancialSummary = styled.div`
  display: grid;
  grid-template-columns: 1.5fr 1fr 1fr 1fr;
  gap: 16px;
  margin-bottom: 32px;
  @media (max-width: 900px) {
    grid-template-columns: repeat(2, 1fr);
  }
`

const SummaryCard = styled.div`
  background: ${props => props.$main ? props.theme.gradients.burgundy : props.theme.colors.surface};
  padding: 24px;
  border-radius: ${props => props.theme.borderRadius.lg};
  border: 1px solid ${props => props.$main ? 'transparent' : props.theme.colors.border};
  display: flex;
  align-items: center;
  gap: 16px;
`

const SummaryIcon = styled.div`
  width: 48px;
  height: 48px;
  background: rgba(255,255,255,0.2);
  border-radius: ${props => props.theme.borderRadius.md};
  display: flex;
  align-items: center;
  justify-content: center;
  color: ${props => props.theme.colors.gold};
  font-size: 24px;
`

const SummaryContent = styled.div``
const SummaryValue = styled.div`
  font-family: ${props => props.theme.fonts.display};
  font-size: 28px;
  font-weight: 700;
  color: ${props => props.theme.colors.text};
  ${SummaryCard}[data-main] & {
    color: ${props => props.theme.colors.cream};
  }
`

const SummaryLabel = styled.div`
  font-family: ${props => props.theme.fonts.ui};
  font-size: 13px;
  color: ${props => props.theme.colors.textMuted};
  ${SummaryCard}[data-main] & {
    color: rgba(255,255,255,0.8);
  }
`

const ReportGrid = styled.div`
  display: grid;
  grid-template-columns: 2fr 1fr;
  gap: 24px;
  @media (max-width: 900px) {
    grid-template-columns: 1fr;
  }
`

const ReportCard = styled.div`
  background: ${props => props.theme.colors.surface};
  border-radius: ${props => props.theme.borderRadius.lg};
  border: 1px solid ${props => props.theme.colors.border};
  overflow: hidden;
`

const ReportCardHeader = styled.div`
  padding: 20px 24px;
  border-bottom: 1px solid ${props => props.theme.colors.border};
  h3 {
    font-family: ${props => props.theme.fonts.heading};
    font-size: 18px;
    color: ${props => props.theme.colors.text};
    margin: 0;
  }
`

const ReportTable = styled.table`
  width: 100%;
  border-collapse: collapse;
  th, td {
    padding: 14px 20px;
  }
`

```

```

    text-align: left;
    font-family: ${props => props.theme.fonts.ui};
    font-size: 13px;
  }
  th {
    color: ${props => props.theme.colors.textMuted};
    font-weight: 600;
    font-size: 11px;
    text-transform: uppercase;
    letter-spacing: 0.5px;
  }
  tbody tr {
    border-top: 1px solid ${props => props.theme.colors.border};
  }
},

const TopCustomersList = styled.div`
  padding: 16px;

const TopCustomer = styled.div`
  display: flex;
  align-items: center;
  gap: 16px;
  padding: 12px 8px;
  border-bottom: 1px solid ${props => props.theme.colors.border};
  &:last-child {
    border-bottom: none;
  }
},

const TopRank = styled.div`
  width: 28px;
  height: 28px;
  background: ${props => props.theme.colors.gold};
  color: ${props => props.theme.colors.primaryDark};
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-weight: 700;
  font-size: 12px;
},

const TopInfo = styled.div`
  flex: 1;

const TopName = styled.div`
  font-weight: 600;
  color: ${props => props.theme.colors.text};

const TopEmail = styled.div`
  font-size: 12px;
  color: ${props => props.theme.colors.textMuted};

const TopSpent = styled.div`
  font-family: ${props => props.theme.fonts.display};
  font-weight: 700;
  color: ${props => props.theme.colors.gold};

const RoleDistribution = styled.div`
  background: ${props => props.theme.colors.surface};
  padding: 24px;
  border-radius: ${props => props.theme.borderRadius.lg};
  border: 1px solid ${props => props.theme.colors.border};
  h3 {
    font-family: ${props => props.theme.fonts.heading};
    font-size: 18px;
    color: ${props => props.theme.colors.text};
    margin: 0 0 20px;
  }
},

const RoleBars = styled.div`
  display: flex;
  flex-direction: column;
  gap: 16px;

const RoleBar = styled.div`
  display: flex;
  align-items: center;
  gap: 16px;

```

```

const RoleBarLabel = styled.div`
  width: 100px;
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  font-weight: 600;
  color: ${props => props.theme.colors.text};
`

const RoleBarTrack = styled.div`
  flex: 1;
  height: 24px;
  background: ${props => props.theme.colors.backgroundDark};
  border-radius: 12px;
  overflow: hidden;
`

const RoleBarFill = styled.div`
  height: 100%;
  border-radius: 12px;
  transition: width 0.5s ease;
`

const RoleBarValue = styled.div`
  width: 80px;
  font-family: ${props => props.theme.fonts.ui};
  font-size: 13px;
  color: ${props => props.theme.colors.textMuted};
  text-align: right;
`

const TrendCard = styled.div`
  display: flex;
  align-items: center;
  gap: 20px;
  padding: 24px;
  margin-bottom: 24px;
  background: ${props => props.$positive
    ? `linear-gradient(135deg, ${props.theme.colors.success}10, ${props.theme.colors.success}05)`
    : `linear-gradient(135deg, ${props.theme.colors.error}10, ${props.theme.colors.error}05)`
  };
  border: 1px solid ${props => props.$positive
    ? `${props.theme.colors.success}30`
    : `${props.theme.colors.error}30`
  };
  border-radius: ${props => props.theme.borderRadius.lg};
`

const TrendIcon = styled.div`
  width: 56px;
  height: 56px;
  background: ${props => props.theme.colors.success}20;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 24px;
  color: ${props => props.theme.colors.success};
`

const TrendContent = styled.div`
  flex: 1;
`

const TrendValue = styled.div`
  font-family: ${props => props.theme.fonts.display};
  font-size: 32px;
  font-weight: 700;
  color: ${props => props.theme.colors.success};
`

const TrendLabel = styled.div`
  font-family: ${props => props.theme.fonts.ui};
  font-size: 14px;
  color: ${props => props.theme.colors.textMuted};
`

const TrendCompare = styled.div`
  display: flex;
  flex-direction: column;
  gap: 4px;
  font-family: ${props => props.theme.fonts.ui};
  font-size: 13px;
  color: ${props => props.theme.colors.textSecondary};
`

const ExcursionsTable = styled(UsersTable)`

```

```

    td strong {
      color: ${props => props.theme.colors.gold};
    }
  }

  const CategoryBadge = styled.span`
    padding: 4px 10px;
    border-radius: 10px;
    font-size: 11px;
    font-weight: 600;
    text-transform: uppercase;
    background: ${props => props.theme.colors.gold}15;
    color: ${props => props.theme.colors.gold};
  `

  const RatingCell = styled.div`
    display: flex;
    align-items: center;
    gap: 4px;
    color: ${props => props.theme.colors.gold};
    svg {
      font-size: 14px;
    }
  `

  export default AdminDashboard

```

ChatPage.jsx

```

/**
 * Сторінка чату з менеджером - бароковий дизайн
 */
import React, { useState, useEffect, useRef } from 'react'
import styled, { keyframes } from 'styled-components'
import Layout from '../components/Layout'
import { chatAPI } from '../utils/api'
import { useAuth } from '../context/AuthContext'
import { FiSend, FiUser, FiMessageSquare, FiTrash2, FiUsers } from 'react-icons/fi'
const ChatPage = () => {
  const { user, isManager, isClient } = useAuth()
  const [selectedConversation, setSelectedConversation] = useState(null)
  const [messages, setMessages] = useState([])
  const [newMessage, setNewMessage] = useState('')
  const [availableUsers, setAvailableUsers] = useState([])
  const messagesEndRef = useRef(null)
  const [scrollY, setScrollY] = useState(0)
  useEffect(() => {
    const handleScroll = () => setScrollY(window.scrollY)
    window.addEventListener('scroll', handleScroll)
    return () => window.removeEventListener('scroll', handleScroll)
  }, [])
  useEffect(() => {
    if (isManager) {
      loadClients()
    } else if (isClient) {
      loadManagers()
    }
  }, [isManager, isClient])
  useEffect(() => {
    if (selectedConversation) {
      loadMessages(selectedConversation)
      const interval = setInterval(() => {
        loadMessages(selectedConversation, true)
      }, 3000)
      return () => clearInterval(interval)
    }
  }, [selectedConversation])
  useEffect(() => {
    scrollToBottom()
  }, [messages])
  const scrollToBottom = () => {
    messagesEndRef.current?.scrollIntoView({ behavior: 'smooth' })
  }
  const loadClients = async () => {
    try {
      const response = await chatAPI.getClients()
      setAvailableUsers(response.data)
    } catch (error) {
      console.error('Помилка завантаження клієнтів:', error)
    }
  }
}

```

```

const loadManagers = async () => {
  try {
    const response = await chatAPI.getManagers()
    setAvailableUsers(response.data)
  } catch (error) {
    console.error('Помилка завантаження менеджерів:', error)
  }
}

const loadMessages = async (userId, showNotification = false) => {
  try {
    const response = await chatAPI.getMessages(userId)
    const newMessages = response.data
    if (showNotification && messages.length > 0 && newMessages.length > messages.length) {
      const latestMessage = newMessages[newMessages.length - 1]
      if (latestMessage.sender_id !== user.id) {
        const sender = availableUsers.find(u => u.id === latestMessage.sender_id)
        if (sender && selectedConversation === userId) {
          showToast(`${sender.username}: ${latestMessage.content.substring(0, 50)}...`)
        }
      }
    }
    setMessages(newMessages)
  } catch (error) {
    console.error('Помилка завантаження повідомлень:', error)
  }
}

const showToast = (message) => {
  const toast = document.createElement('div')
  toast.style.cssText = `
    position: fixed;
    top: 20px;
    right: 20px;
    background: linear-gradient(135deg, #722F37, #5A252C);
    color: white;
    padding: 15px 20px;
    border-radius: 12px;
    box-shadow: 0 4px 20px rgba(114, 47, 55, 0.4);
    z-index: 10000;
    max-width: 300px;
    font-family: 'Source Sans 3', sans-serif;
    border-left: 4px solid #D4AF37;
    animation: slideIn 0.3s ease;
  `
  toast.textContent = message
  document.body.appendChild(toast)
  setTimeout(() => {
    toast.style.animation = 'slideOut 0.3s ease'
    setTimeout(() => toast.remove(), 300)
  }, 3000)
}

const handleSendMessage = async (e) => {
  e.preventDefault()
  if (!newMessage.trim() || !selectedConversation) return
  try {
    await chatAPI.sendMessage({
      receiver_id: selectedConversation,
      content: newMessage,
    })
    setNewMessage('')
    loadMessages(selectedConversation)
  } catch (error) {
    console.error('Помилка відправки повідомлення:', error)
  }
}

const handleDeleteConversation = async (userId) => {
  if (!window.confirm('Видалити всю переписку з цим користувачем?')) return
  try {
    await chatAPI.deleteConversation(userId)
    setSelectedConversation(null)
    setMessages([])
    if (isManager) {
      loadClients()
    }
  } catch (error) {
    console.error('Помилка видалення переписки:', error)
    alert('Не вдалося видалити переписку')
  }
}

```

```

}
const selectedUser = availableUsers.find(u => u.id === selectedConversation)
return (
  <Layout>
    <style>{
      @keyframes slideIn {
        from {
          transform: translateX(400px);
          opacity: 0;
        }
        to {
          transform: translateX(0);
          opacity: 1;
        }
      }
      @keyframes slideOut {
        from {
          transform: translateX(0);
          opacity: 1;
        }
        to {
          transform: translateX(400px);
          opacity: 0;
        }
      }
    }</style>
    <PageWrapper>
      <ParallaxBackground style={{ transform: `translateY(${scrollY * 0.3}px)` }}>
        <BackgroundImage />
        <Overlay />
        <GoldPattern />
      </ParallaxBackground>

      <Container>
        <Header>
          <HeaderIcon><FiMessageSquare /></HeaderIcon>
          <Title>Чат</Title>
          <GoldLine />
          <Subtitle>Зв'яжіться з нашою командою</Subtitle>
        </Header>
        <ChatContainer>
          <Sidebar>
            <SidebarHeader>
              <FiUsers size={20} />
              {isManager ? 'Клієнти' : 'Менеджери'}
            </SidebarHeader>
            <UserList>
              {availableUsers.length === 0 ? (
                <EmptyUsers>
                  <FiUser size={32} />
                  <span>Немає доступних користувачів</span>
                </EmptyUsers>
              ) : (
                availableUsers.map((availableUser) => (
                  <UserItem>
                    <key>{availableUser.id}</key>
                    <$active>{selectedConversation === availableUser.id}</$active>
                  <UserAvatar>
                    <$active>{selectedConversation === availableUser.id}</$active>
                    <onClick>{() => setSelectedConversation(availableUser.id)}</onClick>
                  <UserAvatar>
                    {availableUser.username[0].toUpperCase()}
                  </UserAvatar>
                  <UserInfo onClick={() => setSelectedConversation(availableUser.id)}>
                    <UserName>{availableUser.username}</UserName>
                    <UserRole>{availableUser.role === 'manager' ? 'Менеджер' : 'Клієнт'}</UserRole>
                  </UserInfo>
                  {isManager && (
                    <DeleteButton onClick={() => handleDeleteConversation(availableUser.id)}>
                      <FiTrash2 />
                    </DeleteButton>
                  )}
                </UserItem>
              )}
            </UserList>

```

```

</Sidebar>
<ChatArea>
  {selectedConversation ? (
    <>
    <ChatHeader>
      <ChatUserAvatar>
        {selectedUser?.username[0].toUpperCase()}
      </ChatUserAvatar>
      <ChatUserInfo>
        <ChatUserName>{selectedUser?.username}</ChatUserName>
        <ChatUserRole>
          {selectedUser?.role === 'manager' ? 'Менеджер музею' : 'Клієнт'}
        </ChatUserRole>
      </ChatUserInfo>
    </ChatHeader>
    <MessagesContainer>
      {messages.length === 0 ? (
        <EmptyState>
          <EmptyIcon><FiMessageSquare /></EmptyIcon>
          <EmptyText>Почніть розмову</EmptyText>
          <EmptyHint>Напишіть перше повідомлення</EmptyHint>
        </EmptyState>
      ) : (
        messages.map((message) => (
          <MessageBubble
            key={message.id}
            $isMine={message.sender_id === user.id}
          >
            <MessageContent $isMine={message.sender_id === user.id}>
              {message.content}
            </MessageContent>
            <MessageTime $isMine={message.sender_id === user.id}>
              {new Date(message.created_at).toLocaleTimeString('uk-UA', {
                hour: '2-digit',
                minute: '2-digit',
              })}
            </MessageTime>
          </MessageBubble>
        ))
      )}
      <div ref={messagesEndRef} />
    </MessagesContainer>
    <InputContainer onSubmit={handleSendMessage}>
      <MessageInput
        type="text"
        placeholder="Введіть повідомлення..."
        value={newMessage}
        onChange={(e) => setNewMessage(e.target.value)}
      />
      <SendButton type="submit">
        <FiSend size={20} />
      </SendButton>
    </InputContainer>
  </>
  ) : (
    <SelectConversation>
      <SelectIcon><FiMessageSquare /></SelectIcon>
      <SelectText>Оберіть розмову</SelectText>
      <SelectHint>Виберіть користувача зі списку ліворуч</SelectHint>
    </SelectConversation>
  )}
</ChatArea>
</ChatContainer>
</Container>
</PageWrapper>
</Layout>
)
}
// Animations
const fadeIn = keyframes`
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
`

```

```

},
// Styled Components
const PageWrapper = styled.div`
  position: relative;
  min-height: 100vh;
  padding: 80px 0 50px;
`;

const ParallaxBackground = styled.div`
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 120%;
  z-index: -1;
`;

const BackgroundImage = styled.div`
  width: 100%;
  height: 100%;
  background-image: url('https://images.unsplash.com/photo-1577705998148-6da4f3963bc8?w=1920&q=80');
  background-size: cover;
  background-position: center;
`;

const Overlay = styled.div`
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: linear-gradient(
    135deg,
    rgba(26, 26, 26, 0.92) 0%,
    rgba(114, 47, 55, 0.88) 50%,
    rgba(44, 44, 44, 0.92) 100%
  );
`;

const GoldPattern = styled.div`
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: radial-gradient(circle at 20% 80%, rgba(212, 175, 55, 0.1) 0%, transparent 50%),
    radial-gradient(circle at 80% 20%, rgba(212, 175, 55, 0.08) 0%, transparent 50%);
  pointer-events: none;
`;

const Container = styled.div`
  max-width: 1200px;
  margin: 0 auto;
  padding: 0 20px;
`;

const Header = styled.div`
  text-align: center;
  margin-bottom: 40px;
  color: white;
  animation: ${fadeIn} 0.6s ease-out;
`;

const HeaderIcon = styled.div`
  width: 70px;
  height: 70px;
  background: linear-gradient(135deg, rgba(212, 175, 55, 0.2), rgba(212, 175, 55, 0.1));
  border: 2px solid rgba(212, 175, 55, 0.4);
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 0 auto 20px;
  font-size: 28px;
  color: ${props => props.theme.colors.gold};
`;

const Title = styled.h1`
  font-family: ${props => props.theme.fonts.display};
  font-size: 48px;
  margin-bottom: 16px;
  font-weight: 600;
  letter-spacing: 2px;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);

```

```

const GoldLine = styled.div`
  width: 80px;
  height: 3px;
  background: ${props => props.theme.gradients.gold};
  margin: 0 auto 16px;
  border-radius: 2px;
`

const Subtitle = styled.p`
  font-family: ${props => props.theme.fonts.body};
  font-size: 18px;
  opacity: 0.9;
  font-style: italic;
`

const ChatContainer = styled.div`
  display: grid;
  grid-template-columns: 320px 1fr;
  gap: 0;
  background: ${props => props.theme.colors.surface};
  border-radius: ${props => props.theme.borderRadius.xl};
  overflow: hidden;
  box-shadow: 0 20px 60px rgba(0, 0, 0, 0.4);
  border: 1px solid rgba(212, 175, 55, 0.2);
  height: 650px;
  animation: ${fadeIn} 0.6s ease-out 0.2s both;
  @media (max-width: 768px) {
    grid-template-columns: 1fr;
    height: auto;
  }
`

const Sidebar = styled.div`
  background: ${props => props.theme.colors.backgroundDark};
  border-right: 1px solid ${props => props.theme.colors.border};
  overflow-y: auto;
  &::-webkit-scrollbar {
    width: 6px;
  }
  &::-webkit-scrollbar-track {
    background: transparent;
  }
  &::-webkit-scrollbar-thumb {
    background: ${props => props.theme.colors.gold};
    border-radius: 10px;
  }
`

const SidebarHeader = styled.div`
  padding: 20px;
  background: ${props => props.theme.gradients.burgundy};
  color: ${props => props.theme.colors.cream};
  font-family: ${props => props.theme.fonts.ui};
  font-weight: 600;
  font-size: 15px;
  display: flex;
  align-items: center;
  gap: 10px;
  border-bottom: 2px solid ${props => props.theme.colors.gold};
`

const UserList = styled.div`
  padding: 12px;
`

const EmptyUsers = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  padding: 40px 20px;
  color: ${props => props.theme.colors.textMuted};
  gap: 12px;
  text-align: center;
  font-family: ${props => props.theme.fonts.body};
  font-size: 14px;
`

const UserItem = styled.div`
  display: flex;
  align-items: center;
  gap: 12px;
  padding: 14px;
`

```

```

border-radius: ${props => props.theme.borderRadius.md};
transition: all 0.3s ease;
background: ${props => props.$active ? props.theme.colors.overlayGold : 'transparent'};
border: 1px solid ${props => props.$active ? props.theme.colors.gold : 'transparent'};
margin-bottom: 8px;
position: relative;
&:hover {
  background: ${props => props.$active ? props.theme.colors.overlayGold : 'rgba(212, 175, 55, 0.05)'};
}
,

const UserAvatar = styled.div`
width: 44px;
height: 44px;
border-radius: 50%;
background: ${props => props.$active ? props.theme.gradients.gold : props.theme.colors.primary};
color: ${props => props.$active ? props.theme.colors.primaryDark : props.theme.colors.cream};
display: flex;
align-items: center;
justify-content: center;
font-family: ${props => props.theme.fonts.display};
font-weight: 600;
font-size: 18px;
cursor: pointer;
flex-shrink: 0;
transition: all 0.3s;
,

const UserInfo = styled.div`
flex: 1;
cursor: pointer;
,

const DeleteButton = styled.button`
background: rgba(198, 40, 40, 0.1);
color: ${props => props.theme.colors.error};
border: none;
border-radius: ${props => props.theme.borderRadius.md};
padding: 8px;
cursor: pointer;
display: flex;
align-items: center;
justify-content: center;
transition: all 0.2s ease;
flex-shrink: 0;
&:hover {
  background: ${props => props.theme.colors.error};
  color: white;
}
,

const UserName = styled.div`
font-family: ${props => props.theme.fonts.ui};
font-weight: 600;
font-size: 14px;
color: ${props => props.theme.colors.text};
margin-bottom: 2px;
,

const UserRole = styled.div`
font-family: ${props => props.theme.fonts.body};
font-size: 12px;
color: ${props => props.theme.colors.textMuted};
,

const ChatArea = styled.div`
display: flex;
flex-direction: column;
height: 100%;
overflow: hidden;
background: ${props => props.theme.colors.surface};
,

const ChatHeader = styled.div`
display: flex;
align-items: center;
gap: 14px;
padding: 18px 24px;
background: ${props => props.theme.colors.surface};
border-bottom: 1px solid ${props => props.theme.colors.border};
,

const ChatUserAvatar = styled.div`
width: 48px;
height: 48px;

```

```

border-radius: 50%;
background: ${props => props.theme.gradients.gold};
color: ${props => props.theme.colors.primaryDark};
display: flex;
align-items: center;
justify-content: center;
font-family: ${props => props.theme.fonts.display};
font-weight: 600;
font-size: 20px;
`,

const ChatUserInfo = styled.div`
const ChatUserName = styled.div`
  font-family: ${props => props.theme.fonts.heading};
  font-weight: 600;
  font-size: 17px;
  color: ${props => props.theme.colors.text};
`,

const ChatUserRole = styled.div`
  font-family: ${props => props.theme.fonts.body};
  font-size: 13px;
  color: ${props => props.theme.colors.gold};
`,

const MessagesContainer = styled.div`
  flex: 1;
  padding: 24px;
  overflow-y: auto;
  display: flex;
  flex-direction: column;
  gap: 16px;
  background: ${props => props.theme.colors.background};
  &::-webkit-scrollbar {
    width: 6px;
  }
  &::-webkit-scrollbar-track {
    background: transparent;
  }
  &::-webkit-scrollbar-thumb {
    background: ${props => props.theme.colors.gold};
    border-radius: 10px;
  }
`,

const EmptyState = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100%;
  gap: 12px;
`,

const EmptyIcon = styled.div`
  width: 80px;
  height: 80px;
  background: ${props => props.theme.colors.overlayGold};
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 32px;
  color: ${props => props.theme.colors.gold};
`,

const EmptyText = styled.div`
  font-family: ${props => props.theme.fonts.heading};
  font-size: 20px;
  color: ${props => props.theme.colors.text};
`,

const EmptyHint = styled.div`
  font-family: ${props => props.theme.fonts.body};
  font-size: 14px;
  color: ${props => props.theme.colors.textMuted};
`,

const MessageBubble = styled.div`
  display: flex;
  flex-direction: column;
  align-items: ${props => props.$isMine ? 'flex-end' : 'flex-start'};
  gap: 4px;
`,

const MessageContent = styled.div`

```

```

padding: 14px 18px;
border-radius: ${props => props.$isMine ? '18px 18px 4px 18px' : '18px 18px 18px 4px'};
background: ${props => props.$isMine
  ? props.theme.gradients.burgundy
  : props.theme.colors.surface};
color: ${props => props.$isMine ? props.theme.colors.cream : props.theme.colors.text};
max-width: 70%;
word-wrap: break-word;
font-family: ${props => props.theme.fonts.body};
font-size: 15px;
line-height: 1.5;
box-shadow: ${props => props.$isMine
  ? '0 2px 10px rgba(114, 47, 55, 0.2)'
  : props.theme.shadows.sm};
border: ${props => props.$isMine ? 'none' : `1px solid ${props.theme.colors.border}`};
`,

const MessageTime = styled.div`
font-family: ${props => props.theme.fonts.ui};
font-size: 11px;
color: ${props => props.theme.colors.textMuted};
padding: 0 8px;
`,

const InputContainer = styled.form`
display: flex;
gap: 12px;
padding: 20px 24px;
background: ${props => props.theme.colors.surface};
border-top: 1px solid ${props => props.theme.colors.border};
`,

const MessageInput = styled.input`
flex: 1;
padding: 14px 20px;
border: 2px solid ${props => props.theme.colors.border};
border-radius: ${props => props.theme.borderRadius.full};
font-family: ${props => props.theme.fonts.body};
font-size: 15px;
outline: none;
transition: all 0.3s ease;
background: white;
&:focus {
  border-color: ${props => props.theme.colors.gold};
  box-shadow: 0 0 0 4px rgba(212, 175, 55, 0.15);
}
&::placeholder {
  color: ${props => props.theme.colors.textMuted};
  font-style: italic;
}
`,

const SendButton = styled.button`
padding: 14px 24px;
background: ${props => props.theme.gradients.gold};
color: ${props => props.theme.colors.primaryDark};
border: none;
border-radius: ${props => props.theme.borderRadius.full};
cursor: pointer;
transition: all 0.3s ease;
display: flex;
align-items: center;
justify-content: center;
box-shadow: ${props => props.theme.shadows.gold};
&:hover {
  transform: scale(1.05);
  box-shadow: 0 6px 20px rgba(212, 175, 55, 0.4);
}
`,

const SelectConversation = styled.div`
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
height: 100%;
gap: 16px;
background: ${props => props.theme.colors.background};
`,

const SelectIcon = styled.div`
width: 100px;
height: 100px;

```

