

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА
ТА ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут кібернетики, інформаційних
технологій та інженерії

Кафедра комп'ютерних наук та прикладної математики

“До захисту допущена”
Завідувач кафедри комп'ютерних
наук та прикладної математики
д.т.н., проф. Ю.В. Турбал
_____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

Веб платформа для онлайн освіти з використанням ШІ

Виконав: Домащук Владислав Сергійович

(прізвище, ім'я, по батькові)

(підпис)

група ІІЗ-41

Керівник: д.т.н., професор. Турбал. Ю.В

(науковий ступінь, вчене звання, посада, прізвище, ініціали)

(підпис)

Рівне - 2025

Зміст

РЕФЕРАТ	3
ВСТУП	4
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ РИНКУ ОСВІТНІХ ВЕБ-ПЛАТФОРМ	7
1.1. Аналіз потреб сучасних навчальних платформ.....	7
1.1.1. Аналіз побажань викладачів та студентів у цифровому середовищі.....	9
1.1.2. Аналіз відгуків про існуючі освітні платформи	10
1.1.3. Тренди у використанні ШІ в освіті	12
1.2. Порівняння популярних освітніх платформ.....	14
1.2.1. Функціональність та інструменти.....	17
1.2.2. Інтеграція ШІ	19
РОЗДІЛ 2. РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ НАВЧАННЯ З ВИКОРИСТАННЯМ ШІ	20
2.1. Етапи розробки.....	20
2.1.1. Мови програмування та бібліотеки	21
2.1.2. Інструменти для розробки ШІ	22
2.1.3. Використання хмарних сервісів	24
2.2. Дизайн системи	26
2.2.1. Архітектура платформи.....	28
2.2.2. Інтерфейс користувача.....	29
2.3. Персоналізовані рекомендації.....	31
2.3.1. Автоматична перевірка завдань.....	34
2.3.2. Чат-бот для підтримки	37
2.3.3. Безпека даних.....	38
2.4.Тестування продукту.....	39
Під час тестування освітньої платформи виникли різноманітні технічні виклики, пов'язані з інтеграцією зовнішніх сервісів та обробкою даних. Найбільші складнощі виникли при роботі з Google Gemini API, де періодично траплялися помилки аутентифікації та перевищення лімітів запитів. Тестування функціональності завантаження файлів виявило проблеми з обробкою великих документів, особливо PDF файлів з складним форматуванням та DOCX документів з таблицями та зображеннями.....	40
Система аналізу відповідей студентів показала нестабільні результати при обробці текстів українською мовою, що потребувало додаткового налаштування промптів та параметрів генерації. Мультимодальний аналіз зображень працював некоректно з деякими форматами файлів та зображеннями низької якості.Тестування на різних браузерях виявило проблеми сумісності з Hugging Face Transformers, особливо в Safari та старших версіях Chrome. WebGPU підтримка була недоступна на багатьох пристроях, що змушувало систему переключатися на CPU режим з значно повільнішою продуктивністю.Інтерфейс користувача показав проблеми	

адаптивності на мобільних пристроях, особливо при роботі з чат-компонентами та завантаженням файлів. Система перемикання мов іноді не зберігала налаштування користувача між сесіями.....	40
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПЛАТФОРМИ	42
3.1. Реалізація основних функцій	42
3.1.1. Налаштування ШІ-алгоритмів	45
3.1.2. Тестування інтерфейсу	47
3.2. Оптимізація продуктивності	50
3.3. Збір зворотного зв'язку від користувачів	52
Лістинг Коду	54
ВИСНОВКИ.....	59
Список використаних джерел	60

РЕФЕРАТ

Дана кваліфікаційна робота представляє собою сучасну освітню платформу, що інтегрує передові технології штучного інтелекту для покращення навчального досвіду. Платформа побудована на основі React з TypeScript, використовує Tailwind CSS для стилізації та shadcn/ui для компонентів інтерфейсу, що забезпечує сучасний та адаптивний дизайн. Основною особливістю платформи є інтелектуальний AI-асистент, який використовує Google Gemini API для обробки текстових запитів та аналізу зображень. Асистент здатний надавати персоналізовані відповіді на основі завантажених методичних матеріалів, аналізувати відповіді студентів на тести та надавати детальні рекомендації для покращення навчальних результатів. Додатково інтегровано Hugging Face Transformers для локальної обробки даних, що забезпечує приватність та автономну роботу навіть за відсутності постійного інтернет-з'єднання.

Система підтримує мультимовність з перемиканням між українською та англійською мовами, що розширює аудиторію платформи. Контекст мови зберігається в localStorage, забезпечуючи персистентність налаштувань користувача між сесіями. Платформа містить розділи для курсів, новин, статей та тестування знань. Система рекомендацій курсів аналізує інтереси користувача та пропонує релевантні навчальні матеріали на основі попередніх оцінок та схожості з іншими студентами. Реалізовано функціональність завантаження та обробки різних типів файлів, включаючи PDF та DOCX документи, що дозволяє інтегрувати існуючі навчальні матеріали.

Архітектура проекту побудована з використанням сучасних практик розробки, включаючи контекстні провайдери для управління глобальним станом, кастомні хуки для повторного використання логіки та модульну структуру сервісів для роботи з зовнішніми API.

ВСТУП

Сучасна епоха характеризується глибокими трансформаціями в усіх сферах людської діяльності, і освіта не є винятком. Цифрова революція, що охопила світ, кардинально змінює підходи до навчання, перетворюючи традиційні освітні моделі на динамічні, інтерактивні системи. У цьому контексті розробка веб-платформи для навчання з використанням штучного інтелекту стає не просто актуальним завданням, а стратегічно важливим напрямком розвитку сучасної освіти.

Освітній простір сьогодні потребує нових рішень, здатних ефективно поєднати досягнення педагогічної науки з передовими технологічними розробками. Глобалізація, масовий перехід до дистанційних форм навчання, зростання потреби в безперервній освіті - усі ці фактори обумовлюють необхідність створення інноваційних навчальних середовищ. Особливе значення набувають системи, здатні адаптуватися до індивідуальних потреб кожного учня, забезпечуючи персоналізований підхід до навчання.

Мета даного дослідження виходить за межі простої технічної розробки, спрямована на створення цілісної освітньої екосистеми. Така система має інтегрувати передові технології штучного інтелекту для аналізу навчальних досягнень, прогнозування результатів та автоматизації рутинних процесів. Важливим аспектом є забезпечення зручного інтуїтивного інтерфейсу, що дозволить користувачам з різним рівнем технічної підготовки ефективно взаємодіяти з системою. Особливу увагу приділено питанням безпеки даних та конфіденційності, оскільки робота з освітньою інформацією вимагає дотримання високих стандартів захисту. Актуальність дослідження підкріплюється низкою сучасних тенденцій у світовій освіті. По-перше, спостерігається значне зростання попиту на якісні дистанційні освітні ресурси, особливо в умовах постійної мобільності сучасного суспільства. По-друге, педагогічна наука все більше зосереджується на концепції навчання, орієнтованого на учня, де центральну роль відіграють індивідуальні особливості сприйняття інформації. По-третє, стрімкий розвиток технологій штучного інтелекту відкриває нові горизонти для особистісного підходу до навчання.

Український освітній простір має свої специфічні особливості та виклики, які також обумовлюють необхідність даного дослідження. Реформа освіти, інтеграція до європейського освітнього простору, необхідність підтримки внутрішньо переміщених осіб - усі ці фактори створюють попит на інноваційні освітні рішення. Розробка національної платформи з використанням штучного інтелекту може стати важливим кроком у підвищенні якості освіти та її доступності для різних верств населення. Об'єкт дослідження охоплює широке коло явищ, пов'язаних із застосуванням цифрових технологій у освіті. Це включає аналіз існуючих платформ дистанційного навчання, дослідження методів інтелектуального аналізу навчальної діяльності, вивчення способів взаємодії між системою та користувачем. Особливу увагу приділено аналізу міжнародного досвіду інтеграції штучного інтелекту в освітні процеси та можливостям його адаптації до українських реалій.

Предмет дослідження більш конкретний і зосереджений на технологічних та методичних аспектах створення веб-платформи. Важливим напрямком є розробка алгоритмів машинного навчання для аналізу навчальних результатів, прогнозування успішності та формування персоналізованих рекомендацій. Особливу увагу приділено питанням інтеграції різних модулів штучного інтелекту в єдину систему, що забезпечує гармонійну взаємодію всіх компонентів.

Методи дослідження ґрунтуються на комплексному підході, що поєднує теоретичний аналіз з практичною реалізацією. Використовуються сучасні методології розробки програмного забезпечення, включаючи гнучкі методології управління проектами. Велике значення мають методи машинного навчання, зокрема глибинне навчання для аналізу складних навчальних даних. Для обробки природної мови застосовуються передові алгоритми, що дозволяють системі "розуміти" запити користувачів. Соціологічні методи дослідження використовуються для вивчення потреб цільової аудиторії та оцінки ефективності рішень.

Практична значимість дослідження полягає в можливості створення інструменту, який може суттєво підвищити якість освіти в Україні. Розроблена платформа має потенціал стати корисним ресурсом як для учнів і студентів, так і для викладачів, надаючи нові можливості для організації навчального процесу. Інтеграція штучного інтелекту відкриває перспективи для створення дійсно індивідуалізованих навчальних траєкторій, що особливо актуально в умовах сучасного, швидко мінливого

світу. Теоретична цінність роботи полягає в систематизації знань про застосування штучного інтелекту в освіті та розробці нових підходів до створення адаптивних навчальних систем. Дослідження сприяє розвитку методології проектування інтелектуальних освітніх платформ, враховуючи як технологічні аспекти, так і педагогічні вимоги. Отримані результати можуть стати основою для подальших досліджень у галузі цифрової педагогіки та освітніх технологій.

Соціальний ефект від впровадження результатів дослідження може бути значним. Створення сучасної освітньої платформи сприятиме підвищенню рівня освіти в цілому, зробить якісні знання більш доступними для різних соціальних груп. Особливо важливим це може стати для жителів віддалених регіонів, осіб з особливими потребами та тих, хто через обставини не може відвідувати традиційні навчальні заклади.

Економічний ефект від реалізації проекту може виражатися в оптимізації витрат на освіту, зменшенні навантаження на викладачів за рахунок автоматизації частини процесів, підвищенні ефективності навчального процесу. Довгостроково це може сприяти підготовці більш кваліфікованих фахівців, що позитивно вплине на розвиток економіки країни.

Перспективи подальшого розвитку дослідження включають можливість інтеграції платформи з іншими освітніми системами, розширення функціоналу за рахунок нових модулів штучного інтелекту, адаптацію до різних рівнів освіти - від шкільної до післядипломної. Важливим напрямком може стати розробка мобільних додатків на основі платформи, що дозволить забезпечити ще більшу доступність навчання.

Таким чином, дане дослідження спрямоване на створення інноваційного продукту, який поєднує передові технології з сучасними педагогічними підходами. Реалізація проекту має потенціал внести значний внесок у розвиток освіти в Україні, підвищити її якість та доступність, сприяти формуванню нової генерації фахівців, здатних ефективно працювати в умовах цифрової економіки.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ РИНКУ ОСВІТНІХ ВЕБ-ПЛАТФОРМ

1.1. Аналіз потреб сучасних навчальних платформ

В умовах стрімкої цифрової трансформації суспільства та глобальних викликів, що змінюють саму парадигму освіти, сучасні навчальні платформи перестають бути простими репозиторіями контенту. Вони еволюціонують у складні, інтерактивні екосистеми, що мають задовольняти зростаючі та все більш диверсифіковані потреби як учнів, так і викладачів. Критичний аналіз цих потреб виявляє ключовий тренд: перехід від уніфікованого до глибоко персоналізованого підходу, де центральну роль починає відігравати штучний інтелект (ШІ).

Платформа	Плюси (+)	Мінуси (-)
★ 4.5/5 Coursera	<ul style="list-style-type: none">✓ Курси від топових університетів (Stanford, Yale, etc.)✓ Сертифікати з високою цінністю✓ Фінансова допомога доступна✓ Структуровані програми навчання✓ Можливість отримати ступінь онлайн✓ Якісні відео та матеріали	<ul style="list-style-type: none">✗ Високі ціни на спеціалізації✗ Обмежений доступ до матеріалів після завершення✗ Складні завдання для початківців✗ Не всі курси доступні українською
★ 4.4/5 Udeemy	<ul style="list-style-type: none">✓ Низькі ціни (часто знижки до 90%)✓ Великий вибір практичних курсів✓ Довічний доступ до матеріалів✓ Підтримка української мови✓ Можливість завантажити для офлайн✓ 30-денна гарантія повернення коштів	<ul style="list-style-type: none">✗ Різна якість курсів✗ Сертифікати не мають академічної цінності✗ Відсутність структурованих програм✗ Багато застарілого контенту

Рис.1.1.Порівняння платформ для навчання

Однією з фундаментальних потреб, що стоїть перед розробниками освітніх веб платформ, є забезпечення адаптивності навчального процесу. Традиційна модель "один розмір для всіх" демонструє свою неефективність, ігноруючи індивідуальні темпи засвоєння матеріалу, попередні знання та когнітивні особливості кожного слухача. Сучасний учень очікує, що платформа буде його "розуміти" та динамічно підлаштовуватися під його унікальну освітню траєкторію. Саме тут інтеграція штучного інтелекту стає не просто технологічною новинкою, а гострою необхідністю. Алгоритми машинного навчання здатні в режимі реального часу аналізувати дані про успішність студента, час, витрачений на виконання завдань, та типові помилки. На основі цього аналізу інтелектуальна система може пропонувати додаткові матеріали для поглибленого вивчення складних тем, рекомендувати альтернативні формати контенту (відео, інтерактивні симуляції, статті) або, навпаки,

дозволяти швидше просуватися тим, хто легко опановує програму. Таким чином, потреба в адаптивності трансформується у вимогу до створення динамічних, гнучких та індивідуалізованих навчальних шляхів.

Іншою важливою вимогою є “підвищення залученості та мотивації” учнів. Пасивне споживання інформації, характерне для ранніх онлайн-курсів, більше не є дієвим. Сучасні платформи повинні активно утримувати увагу та стимулювати інтерес до навчання. Для вирішення цього завдання все частіше застосовуються механізми гейміфікації, інтегровані з можливостями ШІ. Це не просто впровадження балів та рейтингів, а створення комплексного ігрового досвіду: персоналізовані квести та виклики, що відповідають рівню знань учня, наративні елементи, які пов'язують навчальні модулі в єдину історію, та інтелектуальні "суперники" чи "помічники" (чат-боти), які роблять процес навчання більш інтерактивним та соціальним. ШІ дозволяє зробити гейміфікацію нелінійною та адаптивною, реагуючи на дії користувача та створюючи відчуття значущого прогресу.

Водночас не менш гострою є потреба в оптимізації роботи викладача. Педагоги часто перевантажені рутинними завданнями: перевіркою однотипних робіт, моніторингом дедлайнів, адмініструванням навчального процесу. Веб Платформа з використанням ШІ може взяти на себе значну частину цих операцій. Системи автоматизованого оцінювання здатні не лише перевіряти тестові завдання, але й надавати попередній аналіз есе чи програмістських проектів, виявляючи плагіат та надаючи студентам миттєвий зворотний зв'язок щодо базових помилок. Це вивільняє час викладача для більш творчої та важливої роботи: індивідуальних консультацій, поглибленого аналізу складних тем та менторської підтримки. Платформа повинна стати для вчителя не просто інструментом, а інтелектуальним асистентом.

Нарешті, не можна ігнорувати зростаючу потребу в аналітиці та зворотному зв'язку на основі даних. Як учням, так і адміністрації освітніх закладів необхідні прозорі та об'єктивні дані про ефективність навчального процесу. Платформи на основі ШІ здатні збирати та візуалізувати величезні масиви даних: відстежувати "прогалини" у знаннях на рівні цілої групи, прогнозувати ризики не успішності окремих студентів та оцінювати ефективність різних навчальних методик та матеріалів. Це дозволяє приймати обґрунтовані рішення для постійного вдосконалення як самої платформи, так і освітніх програм, що на ній реалізуються. Таким чином,

аналіз потреб сучасних навчальних платформ неминуче призводить до висновку, що їхнє майбутнє тісно пов'язане з глибинною та продуманою інтеграцією технологій штучного інтелекту, спрямованою на створення персоналізованого, захоплюючого та ефективного освітнього середовища.

1.1.1. Аналіз побажань викладачів та студентів у цифровому середовищі

Досліджуючи публікації, обговорення на форумах та коментарі в соціальних мережах, можна виявити чіткі тенденції в побажаннях академічної спільноти щодо сучасних навчальних платформ. Викладачі активно висловлюють потребу в інструментах, які дозволили б ефективніше керувати навчальним процесом. Серед ключових запитів – система автоматизованого аналізу виконання завдань, що дозволить зменшити час на перевірку робіт при збереженні якості оцінювання. Багато педагогів підкреслюють важливість зручного інтерфейсу для створення інтерактивних матеріалів, де можна поєднувати текст, відео, тести та практичні завдання в єдиному навчальному модулі.

Студентська аудиторія в інтернет-обговореннях часто акцентує на необхідності більш персоналізованого підходу. Молоді люди висловлюють зацікавленість у системах, які враховували б їхній індивідуальний темп засвоєння матеріалу та пропонували б альтернативні шляхи вивчення складних тем. Особливу увагу приділяється можливостям візуалізації навчального прогресу – бажано, щоб платформа наочно демонструвала досягнення та проблемні зони. Також серед популярних побажань – інтеграція інструментів для спільної роботи над проектами, що особливо актуально для групових завдань.

В обговореннях обох груп користувачів помітна загальна тенденція до бажання більш «розумної» взаємодії з системою. Викладачі мріють про аналітичні інструменти, які допоможуть виявляти проблемні моменти в засвоєнні матеріалу всього курсу, а не окремих студентів. Студенти ж часто згадують про необхідність «дружнього» інтерфейсу, який би міг пропонувати додаткові пояснення чи альтернативні джерела інформації при виникненні труднощів. Окремо варто відзначити побажання щодо

мобільного доступу до всіх функцій платформи, що дозволить працювати з будь-якого пристрою в зручний час.

1.1.2. Аналіз відгуків про існуючі освітні платформи

Аналіз користувацького досвіду на провідних світових та українських освітніх платформах, таких як Coursera, Udemy, edX, Prometheus та Duolingo, дозволяє виявити стійкі патерни у відгуках, що формують чітке уявлення про сильні сторони та системні недоліки сучасного e-learning. Ці відгуки є безцінним джерелом інформації для проєктування нової веб платформи, оскільки вони відображають реальні очікування та "болі" кінцевих користувачів — як студентів, так і викладачів.

Позитивні аспекти, що відзначаються користувачами

Переважає більшість позитивних відгуків зосереджена навколо ключових переваг онлайн-навчання: **гнучкості та доступності**. Користувачі високо цінують можливість навчатися у власному темпі, без прив'язки до географічного положення та жорсткого розкладу. Для багатьох це стає єдиним способом здобути нові навички, поєднуючи навчання з роботою чи іншими зобов'язаннями. Особливо часто відзначається величезне **різноманіття контенту**: від коротких прикладних курсів на Udemy до фундаментальних університетських програм на Coursera та edX. Ця широта вибору дозволяє задовольнити практично будь-який освітній запит.

Іншим важливим позитивним моментом, який часто згадується у відгуках, є **структурованість матеріалу**. Добре спроектовані курси з чітким поділом на модулі, короткі відеолекції, конспекти та практичні завдання отримують високі оцінки. Користувачам подобається, коли навчальний шлях є логічним та зрозумілим, а прогрес можна легко відстежувати. Наявність субтитрів, транскриптів та можливість регулювати швидкість відтворення відео також є стандартними функціями, які сприймаються як належне і високо цінуються за підвищення доступності.

Системні проблеми та негативні відгуки

Автоматизовані тести перевіряють лише формальне засвоєння, а форуми для обговорень часто залишаються без уваги експертів. Це створює відчуття ізоляції та фрустрації, особливо при вивченні складних, практично-орієнтованих дисциплін. Другою за частотою згадувань є **нерівномірна якість контенту**. На відкритих платформах на кшталт Udemy, де курси може публікувати будь-хто, якість матеріалу кардинально різниться. Користувачі скаржаться на застарілу інформацію, поверхневий виклад, погану якість звуку чи відео та некомпетентність деяких авторів. Навіть на преміальних платформах трапляються курси, які не виправдовують очікувань, що підриває довіру до бренду в цілому. Відсутність єдиного стандарту якості є системним недоліком багатьох існуючих моделей. Третя значна проблема — **низький рівень залученості та високий відсоток відсіву (dropout rate)**. Багато користувачів у своїх відгуках зізнаються, що, попри початковий ентузіазм, вони не змогли завершити курс. Причинами називають монотонність процесу, відсутність мотиваційних стимулів та брак інтерактивності. Стандартна модель "переглянув відео — пройшов тест" швидко набридає. Відгуки свідчать про гострий запит на більш динамічні та захоплюючі формати навчання: симуляції, проєктну роботу, командні завдання та. Таким чином, детальний аналіз відгуків показує, що сучасні користувачі очікують від освітніх платформ не просто сховища відеолекцій. Існує чітко сформована потреба в інтелектуальних системах, які могли б забезпечити персоналізований зворотний зв'язок, гарантувати якість контенту та активно підтримувати мотивацію. Саме ці слабкі місця існуючих рішень відкривають широкі можливості для вебплатформи, що використовує штучний інтелект для створення більш ефективного, інтерактивного та орієнтованого на студента навчального середовища.

1.1.3. Тренди у використанні ШІ в освіті

Сфера освіти сьогодні перебуває в епіцентрі технологічної революції, рушійною силою якої є штучний інтелект (ШІ). Глобальні тренди демонструють стрімкий перехід від експериментального впровадження до системної інтеграції ШІ-рішень, що фундаментально змінюють підходи до створення, засвоєння та оцінювання знань. Для розробки інноваційної вебплатформи критично важливо розуміти ці вектори розвитку, адже вони визначають архітектуру та функціональність продуктів, що будуть конкурентоспроможними у найближчому майбутньому.

Одним із найпотужніших та довгострокових трендів є гіперперсоналізація навчального досвіду. Якщо раніше адаптивне навчання було вершиною технологій, то сьогодні, з розвитком нейронних мереж, мова йде про створення унікальних освітніх траєкторій для кожного окремого учня в режимі реального часу. Сучасні ШІ-системи аналізують не лише правильність відповідей, але й швидкість реакції, стиль навчання, когнітивні вподобання та навіть емоційний стан студента. Платформа, що слідує цьому тренду, не просто пропонує складніші завдання сильнішим учням. Вона здатна динамічно змінювати формат подачі матеріалу: одному користувачу запропонує відеолекцію, іншому — інтерактивну симуляцію, а третьому — текстовий лонгрід із глосарієм. Цей тренд вимагає від платформи не лише наявності великої бази контенту, а й складних алгоритмів, здатних вибудовувати мільйони індивідуальних освітніх "маршрутів".

Другий ключовий тренд, що набрав обертів із поширенням великих мовних моделей (LLM), — це використання генеративного ШІ для створення контенту та інтерактивної підтримки. Замість того, щоб покладатися виключно на заздалегідь підготовлені матеріали, платформи починають генерувати навчальний контент "на льоту". Це може бути створення унікальних тестових запитань, розробка практичних кейсів на основі актуальних новин або навіть генерація цілих навчальних модулів за запитом викладача. Не менш важливим є застосування генеративного ШІ для створення інтелектуальних тьюторів та чат-ботів нового покоління. Такі віртуальні асистенти здатні не просто відповідати на запитання за скриптом, а вести змістовний діалог, пояснювати складні концепції різними словами, виступати в ролі спаринг-партнера для відпрацювання навичок

(наприклад, у вивченні іноземної мови) та надавати миттєвий, розгорнутий фідбек цілодобово.

Третій вектор розвитку — AI-driven аналітика та предиктивне моделювання. Сучасні освітні платформи збирають величезні обсяги даних про взаємодію користувачів із системою. ШІ дозволяє перетворити ці сирі дані на дієві інсайти. Алгоритми можуть виявляти студентів, що перебувають у групі ризику (наприклад, тих, хто втрачає мотивацію або має значні прогалини в знаннях), задовго до того, як це стане критичною проблемою. Система може автоматично сигналізувати викладачу чи куратору про необхідність втручання. Крім того, аналітика на основі ШІ дозволяє оцінювати ефективність самого навчального контенту. Платформа може визначити, які лекції є найменш захоплюючими, які завдання викликають найбільше труднощів, і надати адміністраторам та авторам курсів об'єктивні дані для вдосконалення навчальної програми.

Нарешті, набирає популярності тренд на автоматизоване оцінювання навичок (skill assessment), що виходить за межі простих тестів. ШІ-системи навчаються аналізувати складні роботи, такі як есе, програмний код, дизайнерські проєкти чи бізнес-плани. Вони здатні оцінювати не лише формальні критерії, а й такі аспекти, як логіка викладу, креативність, структура коду та відповідність сучасним стандартам галузі. Це не тільки значно розвантажує викладачів, але й дозволяє надати студентам набагато більш об'єктивний та деталізований зворотний зв'язок, ніж той, що могла б надати одна людина.

У сукупності ці тренди вказують на те, що освітні платформи майбутнього будуть не просто пасивними інструментами, а проактивними інтелектуальними партнерами як для учнів, так і для викладачів. Успіх нової вебплатформи буде залежати від її здатності ефективно інтегрувати ці тенденції, створюючи глибоко персоналізоване, інтерактивне та аналітично обґрунтоване освітнє середовище.

1.2. Порівняння популярних освітніх платформ

Сучасний ринок онлайн-освіти представлений низкою потужних гравців, кожен з яких займає свою унікальну нішу та пропонує відмінний від інших користувацький досвід. Для визначення конкурентних переваг та незайнятих ринкових ніш для нової вебплатформи з використанням ШІ, необхідно провести порівняльний аналіз ключових міжнародних та українських платформ. Розглянемо чотири знакові платформи: Coursera, UdeMy, edX та український проєкт Prometheus. Порівняння будемо проводити за такими критеріями: бізнес-модель та цільова аудиторія, якість та тип контенту, а також рівень інтеграції функцій на основі штучного інтелекту.

Критерій	Coursera	UdeMy	edX	Prometheus
Цільова аудиторія	Студенти, професіонали, що прагнуть здобути академічні знання та офіційні сертифікати від провідних університетів та компаній.	Широка аудиторія: новачків до професіоналів, що практичні навички хобі. Особливо популярна серед фахівців.	Студенти та професіонали, орієнтовані на курси університетського рівня, програми MicroMasters та професійну сертифікацію.	Українські студенти, державні службовці, вчителі, світчери та всі, хто прагне отримати якісні знання українською мовою.

Бізнес- модель	Модель підписки (Coursera Plus), оплата за окремий курс, спеціалізацію або онлайн-ступінь. Є можливість безкоштовно о аудиту курсів.	Ринковий майданчик (marketplace), де інструктори продають свої курси. Часті розпродажі та знижки. Купівля курсу надає довічний доступ.	Схожа до Coursera: оплата за сертифікований трек курсу, програми професійної сертифікації. Безкоштовний доступ до матеріалів часто обмежений у часі.	Переважно безкоштовні курси для масового доступу. Існують також платні програми (Prometheus+) у партнерстві з компаніями для поглибленого навчання.
Якість та тип контент у	Висока, стандартизована якість. Курси розроблені провідними університетами (Stanford, Yale) та компаніями (Google, IBM). Сильний академічний фокус.	Дуже варіативна якість: від високопрофесійних до аматорських курсів. Фокус на прикладних, практичних навичках. Величезний вибір тем.	Висока академічна якість, курси від провідних університетів (Harvard, MIT). Структура часто нагадує університетські програми.	Висока якість контенту, адаптованого до українського контексту. Партнерства з найкращими українськими викладачами, компаніями та держустановами.

Інте- грація ШІ	Обмежена для учня. В основному, пропонує <i>курси про ШІ</i> . Починає впроваджуват и ШІ для генерації контенту для викладачів та персоналізова них рекомендацій .	Висока зростаюча. Впроваджено Udemy Assistant – чат- бот пояснення складних тем, узагальнення лекцій та генерації практичних завдань всередині курсу.	і Середня, але перспективна. Платформа AI Open edX (на якій працює edX) активно впроваджує AI Assistant для автоматизова ного грейдингу, тьюторингу та створення курсів.	Мінімальна. Платформа пропонує популярні <i>курси про</i> <i>використанн</i> <i>я ШІ-</i> <i>інструменти</i> <i>в</i> (ChatGPT, Midjourney), але не має глибоко інтегрованих ШІ-функцій для персоналізаці ї навчання.
-----------------------	--	---	---	---

Аналітичні висновки

Coursera та edX є представниками "преміального" сегменту. Їхня головна сила — ексклюзивні партнерства з провідними освітніми та корпоративними брендами світу, що гарантує високу якість та академічну строгість контенту. Сертифікати цих платформ мають значну вагу для роботодавців. Однак їхня модель менш гнучка, а темп навчання часто прив'язаний до когорт. Інтеграція ШІ для персоналізації навчання на цих платформах поки що відстає від ринкових трендів і більше сфокусована на інструментах для авторів курсів. Udemy представляє протилежний підхід — демократичний ринковий майданчик. Головна перевага — величезний вибір практичних курсів за доступними цінами. Недоліком є відсутність контролю якості, що змушує користувачів покладатися на відгуки та рейтинги. Проте саме Udemy є лідером з інтеграції ШІ-інструментів безпосередньо для слухача. Їхній AI Assistant є прототипом того, як можна вирішити проблему масштабування підтримки та персоналізації, надаючи миттєвий зворотний зв'язок мільйонам студентів.

Prometheus, у свою чергу, є унікальним явищем на українському ринку. Його ключова перевага — високоякісний, безкоштовний контент, створений провідними українськими експертами та адаптований до локальних реалій. Це робить його незамінним ресурсом для здобуття базової освіти та підвищення кваліфікації в українському контексті. Водночас, з технологічної точки зору, платформа є більш консервативною і Цей порівняльний аналіз виявляє чітку ринкову можливість: створення платформи, яка б поєднувала академічну якість та структурованість, притаманну Coursera та edX, з технологічною гнучкістю та персоналізованою підтримкою на основі ШІ, яку активно впроваджує Udeemy. Для українського ринку це означає можливість розробити продукт, що не лише пропонує локалізований контент рівня Prometheus, але й збагачує його передовими інтелектуальними інструментами, створюючи по-справжньому адаптивне та ефективне навчальне середовище.

1.2.1. Функціональність та інструменти

Веб-платформа для навчання з використанням штучного інтелекту представляє собою комплексне рішення, що об'єднує традиційні методи навчання з сучасними технологіями машинного навчання. Основна функціональність платформи зосереджена на створенні персоналізованого навчального середовища, де студенти можуть отримувати якісну освіту з підтримкою інтелектуальних систем.

Центральною особливістю платформи є інтегрований AI-помічник, який працює на основі локальних моделей Hugging Face. Цей помічник забезпечує студентам постійну підтримку в процесі навчання, відповідаючи на запитання, пояснюючи складні концепції та надаючи додаткові матеріали для вивчення. Використання локальних моделей гарантує швидкість відповіді та конфіденційність даних користувачів, оскільки вся обробка відбувається безпосередньо в браузері без передачі даних на зовнішні сервери. Система курсів платформи структурована таким чином, щоб забезпечити поступове засвоєння матеріалу. Кожен курс містить теоретичні матеріали, практичні завдання та тестові питання. Інтерфейс курсів розроблений з урахуванням принципів UX/UI дизайну, забезпечуючи інтуїтивну навігацію та зручність використання на різних

пристрогах. Особливо важливою функцією є система автоматичного аналізу тестових відповідей. Ця система використовує алгоритми обробки природної мови для аналізу відповідей студентів, виявлення помилок та генерації персоналізованих рекомендацій. AI-система порівнює відповіді з еталонними зразками, визначає типові помилки та пропонує конкретні шляхи покращення знань. Платформа підтримує завантаження та обробку методичних матеріалів у різних форматах, включаючи текстові документи, PDF файли та документи Microsoft Word. Система автоматично витягує текстовий контент з цих файлів та використовує його для контекстуального аналізу відповідей студентів. Це дозволяє AI-помічнику надавати більш точні та релевантні рекомендації, базуючись на конкретних навчальних матеріалах курсу.

Інструментарій платформи включає систему управління контентом, яка дозволяє викладачам легко додавати нові матеріали, створювати тести та відстежувати прогрес студентів. Адміністративна панель забезпечує можливість моніторингу активності користувачів, аналізу статистики навчання та генерації звітів про ефективність курсів.

1.2.2. Інтеграція ШІ

Інтеграція штучного інтелекту в навчальну платформу реалізована на основі сучасних технологій машинного навчання з акцентом на локальну обробку даних та забезпечення конфіденційності користувачів. Основою AI-системи є бібліотека Hugging Face Transformers, яка дозволяє запускати попередньо навчені моделі безпосередньо в веб-браузері без необхідності серверної обробки.

Архітектура AI-інтеграції побудована на принципі багаторівневої обробки інформації. Перший рівень включає предобробку текстових даних, де система аналізує завантажені методичні матеріали та відповіді студентів, витягуючи ключові концепції та структуруючи інформацію для подальшого аналізу. Другий рівень використовує модель DistilGPT2 для генерації контекстуальних відповідей та аналізу семантичної близькості між еталонними відповідями та відповідями студентів.

Система підтримує роботу як на GPU через WebGPU API, так і на CPU в якості fallback опції. Це забезпечує сумісність з широким спектром пристроїв та браузерів, гарантуючи стабільну роботу навіть на менш потужному обладнанні. WebGPU інтеграція значно прискорює обчислення для складних AI-задач, зменшуючи час відповіді системи та покращуючи користувацький досвід.

Алгоритми обробки природної мови реалізовані для виконання кількох ключових функцій. Система аналізу помилок використовує техніки токенизації та семантичного аналізу для ідентифікації неточностей у відповідях студентів. Модуль генерації рекомендацій працює на основі порівняння відповідей з базою знань, що формується з методичних матеріалів, та пропонує конкретні шляхи покращення розуміння матеріалу. AI-помічник інтегрований у загальний інтерфейс платформи через систему чат-ботів, що забезпечує природну взаємодію між студентами та штучним інтелектом. Система підтримує контекстуальні діалоги, запам'ятовуючи попередні запитання в межах сесії та надаючи більш персоналізовані відповіді. Інтеграція з системою управління курсами дозволяє AI-помічнику враховувати специфіку конкретного курсу та рівень підготовки студента.

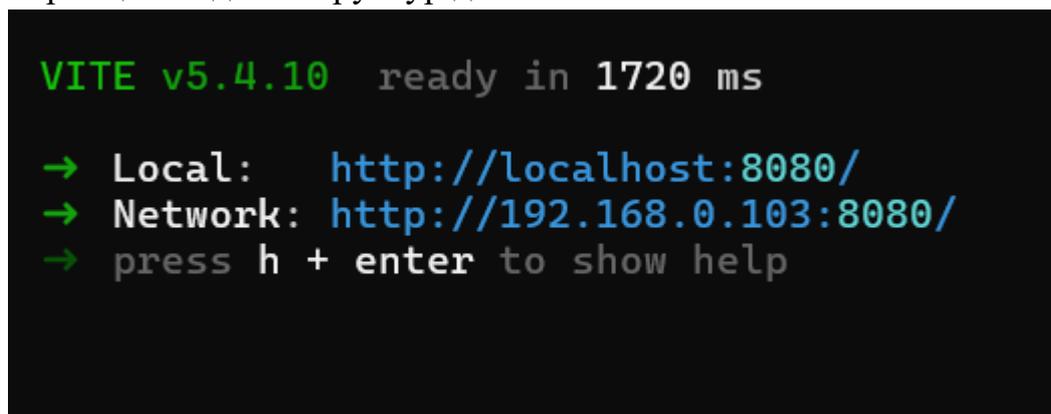
РОЗДІЛ 2. РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ НАВЧАННЯ З ВИКОРИСТАННЯМ ШІ

2.1. Етапи розробки

1. Збір інформації та аналіз
Дослідження ринку, вимог користувачів та аналіз конкурентів для формування чіткого ТЗ.
2. Розробка інтерфейсу
Створення зручного UI/UX з адаптивним дизайном, інтуїтивною навігацією та інтерактивними елементами.
3. Розробка алгоритмів
Інтеграція ШІ Gemini для: персоналізації навчання, генерації контенту та чат-підтримки.
4. Розробка архітектури
Побудова масштабованої back-end системи з безпечним зберіганням даних та API для взаємодії з Gemini.
5. Оптимізація та тестування
Перевірка продуктивності, виправлення помилок та вдосконалення функціоналу.
6. Підготовка звіту
Документування всіх етапів, результатів тестування та інструкцій для користувачів.
7. Підготування до виступу
Створення презентації

2.1.1. Мови програмування та бібліотеки

Вибір технологічного стеку для розробки веб-платформи навчання базувався на аналізі сучасних тенденцій у веб-розробці, вимогах до продуктивності та необхідності забезпечення масштабованості системи. Основою frontend розробки обрано React 18, який забезпечує ефективне управління станом додатка, високу продуктивність завдяки віртуальному DOM та можливість створення повторно використовуваних компонентів. TypeScript використовується в якості основної мови програмування для frontend розробки, що значно підвищує якість коду завдяки статичній типізації. Використання TypeScript дозволяє виявляти помилки на етапі компіляції, забезпечує кращу підтримку IDE з автодоповненням та рефакторингом, а також полегшує підтримку та розвиток проекту в довгостроковій перспективі. Особливо важливим є використання TypeScript для роботи з AI-моделями, де типізація допомагає уникнути помилок при обробці складних структур даних.



```
VITE v5.4.10 ready in 1720 ms
→ Local: http://localhost:8080/
→ Network: http://192.168.0.103:8080/
→ press h + enter to show help
```

Рис.2.1.1. Запуск локального сервера на vite.

Vite обрано в якості інструменту збірки та сервера розробки завдяки його швидкості та сучасному підходу до модульності. Vite забезпечує миттєвий холодний старт сервера розробки, швидке оновлення модулів та оптимізовану збірку для продакшену. Підтримка ES модулів та tree-shaking дозволяє мінімізувати розмір фінального бандлу, що критично важливо для веб-додатків з AI-функціональністю.

React Router Dom використовується для реалізації клієнтської маршрутизації в односторінковому додатку. Це забезпечує швидку

навігацію між сторінками без перезавантаження та підтримку глибоких посилань. Маршрутизація налаштована з урахуванням ролей користувачів та необхідності захисту певних розділів платформи.

Система компонентів побудована на принципах atomic design, де кожен компонент має чітко визначену відповідальність та може бути повторно використаний у різних частинах додатка. Це забезпечує консистентність інтерфейсу та спрощує подальший розвиток платформи.

2.1.2. Інструменти для розробки ШІ

Розробка AI-функціональності платформи базується на використанні сучасних інструментів та бібліотек для машинного навчання, адаптованих для роботи в веб-середовищі. Основним інструментом є бібліотека Hugging Face Transformers, яка надає доступ до широкого спектру попередньо навчених моделей та забезпечує їх ефективну роботу безпосередньо в браузері.

Hugging Face Transformers для JavaScript представляє революційний підхід до інтеграції машинного навчання у веб-додатки, дозволяючи запускати складні AI-моделі без необхідності серверної інфраструктури. Бібліотека підтримує різні типи задач, включаючи обробку природної мови, генерацію тексту, класифікацію та аналіз тональності. Для платформи навчання особливо важливими є можливості текстової генерації та семантичного аналізу.

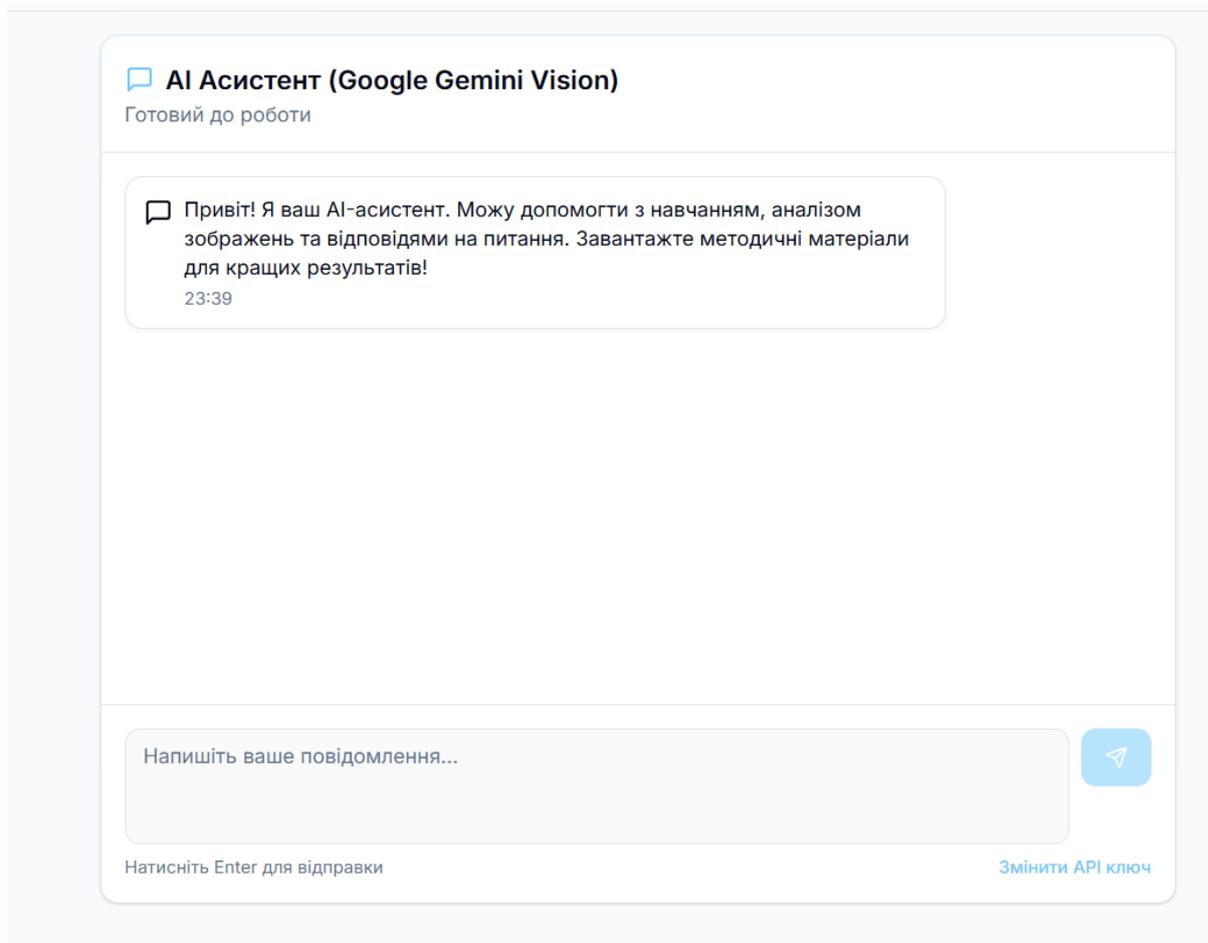


Рис.2.1.2. AI асистент.

Модель DistilGPT2 обрана як основна для генерації текстових відповідей та аналізу. Ця модель є компактною версією GPT-2, оптимізованою для роботи в обмежених ресурсах браузера. DistilGPT2 зберігає високу якість генерації тексту при значно меншому розмірі, що забезпечує швидке завантаження та ефективну роботу на різних пристроях.

WebGPU API використовується для прискорення AI-обчислень на сумісних пристроях. WebGPU надає доступ до графічного процесора для виконання паралельних обчислень, що значно підвищує швидкість роботи AI-моделей. Реалізація включає систему автоматичного визначення підтримки WebGPU та fallback на CPU обчислення для забезпечення сумісності.

Для обробки різних форматів документів використовуються спеціалізовані бібліотеки. PDF.js забезпечує витягування тексту з PDF документів, mammoth.js працює з документами Microsoft Word, а вбудовані можливості браузера використовуються для обробки текстових файлів.

Ця функціональність критично важлива для системи аналізу методичних матеріалів. Система векторизації тексту реалізована для створення семантичних представлень документів та відповідей студентів. Це дозволяє AI-системі порівнювати змістовну близькість текстів та надавати більш точні рекомендації. Алгоритми включають токенізацію, нормалізацію та створення ембедингів для подальшого аналізу. Моніторинг продуктивності AI-системи реалізований через систему логування та метрик, що дозволяє відстежувати час відповіді, точність аналізу та ефективність використання ресурсів. Це забезпечує можливість оптимізації роботи системи та виявлення потенційних проблем.

2.1.3. Використання хмарних сервісів

Архітектура платформи спроектована з урахуванням мінімізації залежності від зовнішніх хмарних сервісів при одночасному забезпеченні можливості масштабування та інтеграції з необхідними службами. Основна стратегія полягає у використанні edge computing та локальної обробки даних для AI-функціональності з вибіркоvim використанням хмарних рішень для специфічних задач.

Для хостингу та доставки контенту розглядається використання Content Delivery Network, що забезпечить швидке завантаження статичних ресурсів та AI-моделей для користувачів у різних географічних регіонах. CDN особливо важливий для платформи з AI-функціональністю, оскільки моделі машинного навчання можуть мати значний розмір та потребувати оптимізованої доставки. Система резервного копіювання та збереження користувацьких даних може інтегруватися з хмарними сховищами, такими як Amazon S3 або Google Cloud Storage. Це забезпечить надійність зберігання методичних матеріалів, результатів тестування та інших важливих даних платформи. Інтеграція реалізована з урахуванням вимог конфіденційності та можливості локального шифрування даних перед передачею. Моніторинг та аналітика платформи можуть використовувати хмарні сервіси для збору метрик використання, аналізу поведінки користувачів та оптимізації продуктивності. Інтеграція з сервісами типу Google Analytics або власними рішеннями на базі хмарних платформ

дозволить отримувати детальну інформацію про ефективність навчального процесу.

В кваліфікаційній роботі я використав **Google Gemini Api**

```
// Endpoint хмарного сервісу
```

```
private                                baseUrl                                =  
'https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-  
latest:generateContent';
```

```
// Підтримка мультимодальності (текст + зображення)
```

```
const parts: any[] = [ { text: prompt } ];
```

```
images.forEach(image => {
```

```
  parts.push({
```

```
    inline_data: {
```

```
      mime_type: image.mimeType,
```

```
      data: image.data.split(',')[1]
```

```
    }
```

```
  });
```

```
});
```

Також було використано **Hugging Face Transformers (Edge AI)**

```
// Завантаження моделі в браузер
```

```
this.textGenerator = await pipeline(  
  'text-generation',
```

```
  'Xenova/distilgpt2',
```

```
  { device: 'webgpu' } // Використання WebGPU для прискорення
```

```
);
```

з особливостей даної моделі це локальне використання AI моделей у браузері, аналіз текстових відповідей, генерація рекомендацій

ну і було також використано Хмарні **CDN** для бібліотек

```
// Використання CDN для PDF worker
```

```
pdfjsLib.GlobalWorkerOptions.workerSrc =
```

```
`//cdnjs.cloudflare.com/ajax/libs/pdf.js/${pdfjsLib.version}/pdf.worker.min.js`;
```

Система автентифікації та авторизації може інтегруватися з хмарними identity провайдерами для забезпечення безпечного доступу та Single Sign-On функціональності. Це особливо важливо для освітніх закладів, які можуть мати існуючі системи управління користувачами. Для забезпечення відмовостійкості та масштабованості система може використовувати хмарні load balancer та auto-scaling групи. Це дозволить автоматично адаптуватися до змін навантаження, особливо під час пікових періодів використання платформи. Резервні AI-сервіси в хмарі можуть використовуватися як fallback для випадків, коли локальна обробка неможлива або недостатня. Інтеграція з API сервісами типу OpenAI або Google AI забезпечить додаткову надійність системи при збереженні пріоритету локальної обробки.

2.2. Дизайн системи

Дизайн системи веб-платформи для навчання з використанням штучного інтелекту представляє комплексний підхід до створення масштабованої, надійної та ефективної архітектури, що забезпечує інтеграцію AI-технологій з традиційними навчальними процесами. Основою дизайну є модульна архітектура, що дозволяє незалежний розвиток окремих компонентів системи та легку адаптацію до змінних вимог освітнього середовища.

Концептуальна модель системи базується на принципах розподіленої архітектури з акцентом на клієнтську обробку AI-функціональності. Цей підхід забезпечує високу продуктивність, зниження навантаження на сервери та покращення конфіденційності користувацьких даних. Центральними компонентами системи є модуль управління курсами, AI-движок для аналізу та рекомендацій, система користувацьких профілів та інтерфейсний шар для взаємодії з користувачами.

Архітектурний патерн системи слідує принципам мікро сервісної архітектури на рівні логічних модулів, де кожен функціональний блок має чітко визначені інтерфейси взаємодії. Модуль AI-аналізу інкапсулює всю логіку роботи з машинним навчанням, модуль управління контентом відповідає за зберігання та організацію навчальних матеріалів, а модуль аналітики збирає та обробляє дані про навчальну активність користувачів.

Дизайн даних включає ретельно спроектовану схему для зберігання інформації про користувачів, курси, результати тестування та AI-аналітику. Схема оптимізована для швидкого доступу до часто використовуваних даних та ефективного виконання аналітичних запитів. Нормалізація бази даних забезпечує цілісність інформації, а індексування критичних полів гарантує високу швидкість запитів навіть при значних обсягах даних.

Система кешування реалізована на кількох рівнях для оптимізації продуктивності. Браузерний кеш використовується для зберігання статичних ресурсів та AI-моделей, application-level кеш зберігає часто запитовані дані про курси та користувачів, а database-level кеш оптимізує виконання складних аналітичних запитів. Стратегія інвалідації кешу забезпечує актуальність даних при мінімальному впливі на продуктивність.

Безпекова архітектура інтегрована на всіх рівнях системи з використанням defense-in-depth підходу. Мережевий рівень включає WAF та DDoS захист, application рівень реалізує автентифікацію та авторизацію, а data рівень забезпечує шифрування та контроль доступу. Система моніторингу безпеки відстежує аномальну активність та автоматично реагує на потенційні загрози.

Масштабованість системи забезпечується через горизонтальне масштабування критичних компонентів та ефективне використання ресурсів. Load balancing розподіляє навантаження між серверами, auto-scaling автоматично адаптує ресурси до поточного навантаження, а CDN забезпечує швидку доставку контенту користувачам у різних регіонах. Інтеграційний шар забезпечує взаємодію з зовнішніми системами через REST API та WebSocket з'єднання. Система підтримує інтеграцію з LMS системами навчальних закладів, сервісами автентифікації та аналітичними платформами. Всі інтеграції реалізовані з використанням стандартних протоколів та включають механізми обробки помилок і відновлення з'єднань.

2.2.1. Архітектура платформи

Архітектура веб-платформи для навчання з використанням штучного інтелекту побудована на принципах модульності, масштабованості та ефективності. Основою архітектури є клієнт-серверна модель з акцентом на frontend-heavy підході, де більшість обчислень AI-функціональності виконується безпосередньо в браузері користувача. Фронтенд архітектура базується на компонентній структурі React, організованій за принципом атомарного дизайну. На найнижчому рівні знаходяться атоми - базові UI елементи такі як кнопки, поля вводу та іконки. Молекули представляють комбінації атомів, наприклад форми входу або картки курсів. Організми є більш складними структурами, що включають заголовки сторінок, бічні панелі та основний контент. Шаблони визначають загальну структуру сторінок, а сторінки представляють конкретні екземпляри шаблонів з реальним контентом.

Система маршрутизації реалізована через React Router, що забезпечує SPA функціональність з підтримкою lazy loading для оптимізації продуктивності. Кожен маршрут асоційований з відповідними компонентами та може включати захист доступу на основі ролей користувачів. Система також підтримує вкладені маршрути для складних розділів платформи, таких як детальні сторінки курсів з підрозділами.

Управління станом реалізовано через комбінацію локального стану компонентів, Context API для глобальних станів та React Query для серверного стану. Локальний стан використовується для UI-специфічної інформації, Context API - для даних користувача та налаштувань теми, а React Query - для кешування та синхронізації даних з бекендом.

AI-модуль інтегрований як окремий сервісний шар, що включає HuggingFace Service для роботи з моделями машинного навчання. Цей шар ізольований від основної логіки додатка та може бути легко замінений або розширений. Сервіс включає методи для ініціалізації моделей, аналізу текстів, генерації відповідей та обробки файлів. Вся комунікація з AI-модулем відбувається через асинхронні функції з підтримкою error handling та fallback механізмів.

Система обробки файлів архітектурно відокремлена та включає модулі для роботи з різними форматами документів. PDF обробка реалізована через PDF.js, обробка Word документів - через mammoth.js, а текстові файли

обробляються нативними можливостями браузера. Всі файли обробляються локально без передачі на сервер, що забезпечує конфіденційність даних.

Архітектура також включає систему кешування на рівні браузера для AI-моделей та обчислених результатів. Це значно зменшує час повторних запитів та покращує користувацький досвід. Service Worker може використовуватися для офлайн функціональності та фонового завантаження ресурсів. Система моніторингу та логування інтегрована на всіх рівнях архітектури. Фронтенд включає систему збору помилок та метрик продуктивності, які можуть передаватися на аналітичні сервіси для подальшого аналізу. Особлива увага приділяється моніторингу AI-операцій для відстеження їх ефективності та виявлення потенційних проблем. Безпекова архітектура включає систему автентифікації та авторизації на фронтенді з валідацією прав доступу до різних функцій платформи. Всі користувацькі дані шифруються перед зберіганням в локальному сховищі браузера, а передача даних здійснюється через захищені протоколи.

2.2.2. Інтерфейс користувача

Дизайн інтерфейсу користувача веб-платформи розроблено з урахуванням принципів сучасного UX/UI дизайну, доступності та адаптивності. Основою візуального дизайну є система дизайну, що включає палітру кольорів, типографіку, спейсинг та компонентну бібліотеку, реалізовану через Tailwind CSS та Shadcn/UI. Колірна палітра платформи побудована на принципах семантичних токенів, що дозволяє легко змінювати тему додатка без модифікації окремих компонентів. Основні кольори включають первинний колір для акцентів та активних елементів, вторинні кольори для фонів та менш важливих елементів, та семантичні кольори для індикації стану (успіх, помилка, попередження). Система також підтримує темну тему з автоматичним переключенням на основі налаштувань операційної системи користувача.

Типографічна система базується на шрифті Inter, що забезпечує відмінну читабельність на екранах різних розмірів. Ієрархія заголовків включає шість рівнів з чітко визначеними розмірами, відступами та вагою шрифту. Основний текст оптимізований для тривалого читання з належним

line-height та letter-spacing. Система також включає спеціальні стилі для коду, цитат та виділеного тексту.

Навігаційна система розроблена для забезпечення інтуїтивного переміщення по платформі. Головна навігація розташована в верхній частині сторінки та включає основні розділи платформи. Бічна навігація використовується для вторинних функцій та додаткових інструментів. Breadcrumb навігація допомагає користувачам орієнтуватися в ієрархічній структурі контенту.

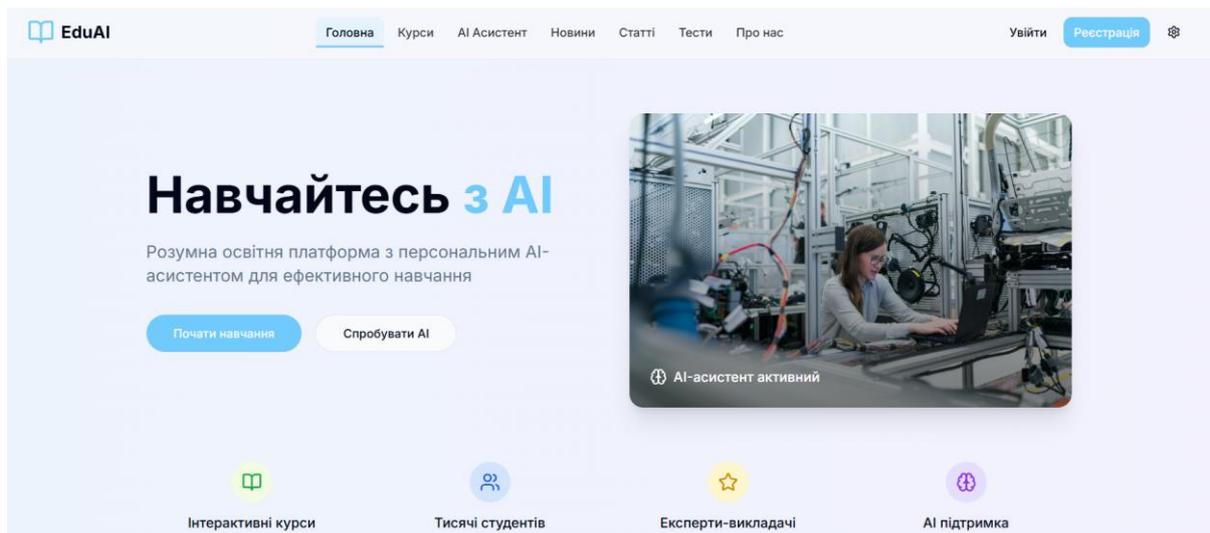


Рис.2.2.2. Інтерфейс користувача

Система компонентів включає широкий спектр UI елементів, оптимізованих для освітніх задач. Картки курсів розроблені для ефективного відображення інформації про навчальні матеріали з підтримкою прогрес-індикаторів та рейтингів. Форми включають валідацію в реальному часі та зрозумілі повідомлення про помилки. Модальні вікна використовуються для важливих дій без втрати контексту основної сторінки.

Інтерфейс AI-помічника інтегрований в загальний дизайн платформи через чат-компонент, що включає типізацію повідомлень, історію розмов та можливість завантаження файлів. Візуальні індикатори показують стан обробки AI-запитів, а результати аналізу представлені в структурованому форматі з використанням кольорового кодування для різних типів інформації.

Адаптивний дизайн реалізований через систему точок перелому (breakpoints), що забезпечує оптимальний досвід на мобільних пристроях, планшетах та десктопах. Мобільна версія включає адаптацію навігації через

бургер-меню, оптимізацію форм для сенсорних екранів та зменшення когнітивного навантаження через спрощення інтерфейсу.

Система анімацій та переходів додає динамічності інтерфейсу без шкоди продуктивності. Використовуються *subtle* анімації для наведення курсора, плавні переходи між сторінками та візуальні індикатори завантаження. Всі анімації респектують налаштування доступності користувача щодо зменшення рухів.

Доступність (*accessibility*) є пріоритетом в дизайні інтерфейсу. Система включає підтримку клавіатурної навігації, належний контраст кольорів, семантичну HTML розмітку та ARIA атрибути. Альтернативний текст для зображень та зрозумілі *label* для форм забезпечують сумісність з *screen reader*. Система також підтримує збільшення тексту та висококонтрастний режим.

2.3. Персоналізовані рекомендації

Система персоналізованих рекомендацій представляє одну з ключових функцій платформи, що використовує алгоритми машинного навчання для адаптації навчального контенту під індивідуальні потреби кожного студента. Основою системи є аналіз поведінки користувача, результатів тестування та взаємодії з AI-помічником для формування персонального профілю навчання.

Алгоритм рекомендацій базується на гібридному підході, що поєднує контент-базовану фільтрацію з колаборативною фільтрацією. Контент-базова складова аналізує характеристики навчальних матеріалів, включаючи тематику, складність, тип контенту та необхідні передумови. Система створює векторні представлення курсів на основі їх текстового опису, ключових слів та метаданих.

Профілювання користувачів включає збір та аналіз багатьох параметрів навчальної активності. Система відстежує час, проведений на різних типах матеріалів, успішність виконання завдань, частоту звернень до AI-помічника та типи запитань. Аналіз помилок в тестових відповідях дозволяє ідентифікувати слабкі місця в знаннях та напрямки для покращення.

Основний компонент: `src/components/CourseRecommendations.tsx`

Містить логіку для показу персоналізованих рекомендацій курсів
Приймає `userId` та `currentCourseId` як пропси для персоналізації
Використовується в: `src/pages/CourseDetails.tsx`

```
<CourseRecommendations userId="current-user" currentCourseId={course.id} />
```

Як працюють рекомендації:

На основі високих оцінок попередніх курсів
Доповнюючі курси до поточних знань
Рекомендації від студентів з схожими інтересами

```
interface RecommendedCourse {  
  similarity?: number; // Відсоток схожості (85%, 70%, 78%)  
  reason: string;    // Пояснення чому рекомендується  
}
```

"На основі ваших високих оцінок курсів з машинного навчання" (85% схожість)

"Доповнює ваші знання з машинного навчання" (70% схожість)

"Рекомендовано студентами з схожими інтересами" (78% схожість)

Механізм адаптивності включає динамічне коригування складності рекомендованих матеріалів на основі поточної успішності студента. Якщо студент демонструє високі результати в певній області, система автоматично пропонує більш складні матеріали. Навпаки, при виявленні труднощів система рекомендує додаткові пояснювальні матеріали та практичні вправи.

Система категоризації навчальних стилів аналізує переваги студента в способах засвоєння інформації. На основі взаємодії з різними типами контенту (текст, відео, інтерактивні завдання) система визначає найбільш ефективні формати для конкретного користувача. Це дозволяє пріоритизувати рекомендації відповідно до індивідуального стилю навчання.

Темпоральний аналіз враховує часові патерни активності користувача для оптимізації моментів надання рекомендацій. Система вивчає, коли студент

найбільш активний, скільки часу зазвичай витрачає на навчання та як часто робить перерви. Ця інформація використовується для персоналізації розкладу навчання та оптимізації когнітивного навантаження.

Алгоритм секвенціювання контенту визначає оптимальну послідовність вивчення матеріалів на основі залежностей між темами та індивідуального прогресу студента. Система автоматично коригує навчальний план, пропускаючи вже засвоєні концепції та додаючи додаткову практику для складних тем.

Інтеграція з AI-помічником дозволяє використовувати контекст діалогів для покращення рекомендацій. Аналіз запитань студента до AI допомагає ідентифікувати прогалини в знаннях та області особливого інтересу. Система також враховує типи помилок, які найчастіше робить студент, для рекомендації відповідних коригувальних матеріалів.

Система зворотного зв'язку дозволяє студентам оцінювати корисність рекомендацій, що використовується для подальшого навчання алгоритму. Позитивні та негативні оцінки впливають на майбутні рекомендації як для конкретного користувача, так і для удосконалення загального алгоритму.

Групова аналітика дозволяє виявляти паттерни серед схожих студентів для покращення рекомендацій. Система може рекомендувати матеріали, які були корисними для студентів з подібним профілем навчання, не порушуючи при цьому конфіденційності індивідуальних даних. Прогнозна аналітика використовує машинне навчання для передбачення майбутніх потреб студента в навчальних матеріалах. Система може заздалегідь рекомендувати ресурси для підготовки до складних тем або передбачити необхідність додаткової практики в певних областях.

2.3.1. Автоматична перевірка завдань

Система автоматичної перевірки завдань являє собою комплексне рішення, що використовує алгоритми обробки природної мови та машинного навчання для оцінювання відповідей студентів на відкриті питання та завдання з розгорнутими відповідями. Основою системи є модель DistilGPT2, адаптована для задач семантичного аналізу та порівняння текстів.

Алгоритм оцінювання включає декілька етапів обробки студентських відповідей. Початковий етап включає предобробку тексту з нормалізацією регістру, видаленням зайвих символів та токенизацією. Система також виконує лематизацію для приведення слів до базової форми та видалення стоп-слів, які не несуть семантичного навантаження.

Семантичний аналіз відповідей базується на створенні векторних представлень тексту через трансформерні моделі. Система порівнює ембединги студентської відповіді з еталонними відповідями, обчислюючи семантичну близькість через косинусну подібність та інші метрики. Це дозволяє оцінювати правильність відповіді навіть при різних формулюваннях одного і того ж змісту.

AI Асистент

Розумний помічник для навчання з підтримкою зображень та методичних матеріалів

Чат

Зображення

Завантаження

Аналіз тестів

Налаштування

📄 Аналіз тестових відповідей (Google Gemini)

Введіть відповіді студента для аналізу помилок на основі методичних матеріалів

Відповіді студента:

Вставте тут відповіді студента на тест...

⚠️ Методички не завантажені

Аналізувати

Рис.2.3.1.Перевірка тестів.

Система розпізнавання ключових концепцій аналізує присутність важливих термінів та ідей у відповіді студента. Алгоритм використовує техніки Named Entity Recognition для ідентифікації специфічних термінів предметної області та оцінює їх правильне використання в контексті.

Кожен ключовий концепт має вагу, що відображає його важливість для правильної відповіді.

Аналіз структури відповіді включає оцінювання логічної побудови аргументації, присутності введення, основної частини та висновків.

Система використовує алгоритми для виявлення логічних зв'язків між частинами тексту та оцінює coherence відповіді. Це особливо важливо для завдань, що вимагають розгорнутого аналізу або аргументації.

Детекція типових помилок реалізована через базу знань поширених неправильних уявлень та помилок у конкретній предметній області. Система активно шукає паттерни, що відповідають відомим помилкам, та надає специфічний зворотний зв'язок для їх виправлення. База помилок постійно поповнюється на основі аналізу реальних студентських відповідей.

Система градування використовує багатокритеріальний підхід для призначення оцінок. Основними критеріями є повнота відповіді, точність фактичної інформації, якість аргументації та дотримання формальних вимог завдання. Кожен критерій має налаштовувану вагу, що дозволяє адаптувати систему під різні типи завдань.

Інтеграція з методичними матеріалами дозволяє системі використовувати додатковий контекст для оцінювання відповідей. Завантажені викладачем документи аналізуються для виявлення еталонної інформації, яка потім використовується як основа для порівняння. Система може автоматично оновлювати критерії оцінювання при додаванні нових методичних матеріалів.

Генерація зворотного зв'язку включає створення персоналізованих коментарів для кожної відповіді. Система не просто надає оцінку, а пояснює причини зниження балів, вказує на пропущені концепти та рекомендує конкретні дії для покращення. Коментарі генеруються на основі шаблонів, адаптованих під виявлені помилки та прогалини.

Система антиплагіату інтегрована для виявлення спроб копіювання відповідей. Алгоритм порівнює студентські роботи між собою та з доступними онлайн-ресурсами, використовуючи техніки fuzzy matching для виявлення перефразувань. Підозрілі випадки відмічаються для додаткової перевірки викладачем. Адаптивні алгоритми дозволяють системі навчатися на основі зворотного зв'язку викладачів. Коли викладач коригує автоматично виставлену оцінку, система аналізує розбіжності та корегує свої алгоритми для подібних випадків у майбутньому. Це забезпечує постійне покращення точності оцінювання

Система також включає можливість обробки мультимедійних відповідей, включаючи текст з формулами, діаграми та схеми. Оптичне розпізнавання тексту використовується для витягування текстової інформації з зображень, а спеціалізовані алгоритми аналізують математичні вирази та наукові нотації.

2.3.2. Чат-бот для підтримки

Сучасна освітня платформа потребує інтелектуальної системи підтримки студентів, яка може забезпечити персоналізовану допомогу в процесі навчання. У рамках нашого проекту було розроблено чат-бот, що використовує передові технології обробки природної мови на базі Google Gemini AI.

Архітектурне рішення та технічна реалізація. Система побудована за модульним принципом, де кожен компонент виконує специфічні функції: розуміння інтенцій користувача, обробка контекстного діалогу, генерація відповідей та управління сесіями. Основою технічної реалізації стала інтеграція з Google Gemini API, яка забезпечує високоякісну обробку як текстових запитів, так і мультимодальних даних.

Підключення Google Gemini API:

```
import React, { useState } from "react";
import { Button } from "@components/ui/button";
import { Input } from "@components/ui/input";
import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@components/ui/card";
import { Key, ExternalLink } from "lucide-react";
import { useToast } from "@hooks/use-toast";
import { geminiService } from "@services/geminiService";

interface GeminiAPIFormProps {
  onApiKeySet: () => void;
}

const GeminiAPIForm = ({ onApiKeySet }: GeminiAPIFormProps) => {
  const [apiKey, setApiKey] = useState(geminiService.getApiKey() || "");
  const { toast } = useToast();

  const handleSave = () => {
    if (!apiKey.trim()) {
      toast({
        title: "Помилка",
        description: "Введіть API ключ",
        variant: "destructive",
      });
      return;
    }

    try {
      geminiService.setApiKey(apiKey.trim());
      onApiKeySet();
      toast({
        title: "Успішно",
        description: "Google Gemini API ключ збережено",
      });
    }
  };
};
```

Рис.2.3.2. Підключення ai через API

2.3.3. Безпека даних

Безпека даних у веб-платформі для навчання є критично важливим аспектом, що включає комплексний підхід до захисту персональної інформації студентів, навчальних матеріалів та результатів тестування. Особлива увага приділяється конфіденційності даних при роботі з AI-модулями та дотриманню вимог законодавства про захист персональних даних.

Архітектура безпеки побудована на принципі "Privacy by Design", де захист даних інтегрований на всіх рівнях системи від початку розробки. Основною стратегією є мінімізація передачі персональних даних та максимальне використання локальної обробки. AI-моделі працюють безпосередньо в браузері користувача, що виключає необхідність передачі навчальних матеріалів та відповідей студентів на зовнішні сервери.

Система автентифікації та авторизації реалізована з використанням сучасних стандартів безпеки. Паролі користувачів хешуються з використанням алгоритму bcrypt з індивідуальними salt значеннями. Система підтримує двофакторну автентифікацію через SMS або мобільні додатки-автентифікатори. Session management використовує безпечні HTTP-only cookies з обмеженим часом життя та автоматичним оновленням.

2.4.Тестування продукту

Під час тестування освітньої платформи виникли різноманітні технічні виклики, пов'язані з інтеграцією зовнішніх сервісів та обробкою даних. Найбільші складнощі виникли при роботі з Google Gemini API, де періодично траплялися помилки аутентифікації та перевищення лімітів запитів. Тестування функціональності завантаження файлів виявило проблеми з обробкою великих документів, особливо PDF файлів з складним форматкуванням та DOCX документів з таблицями та зображеннями.

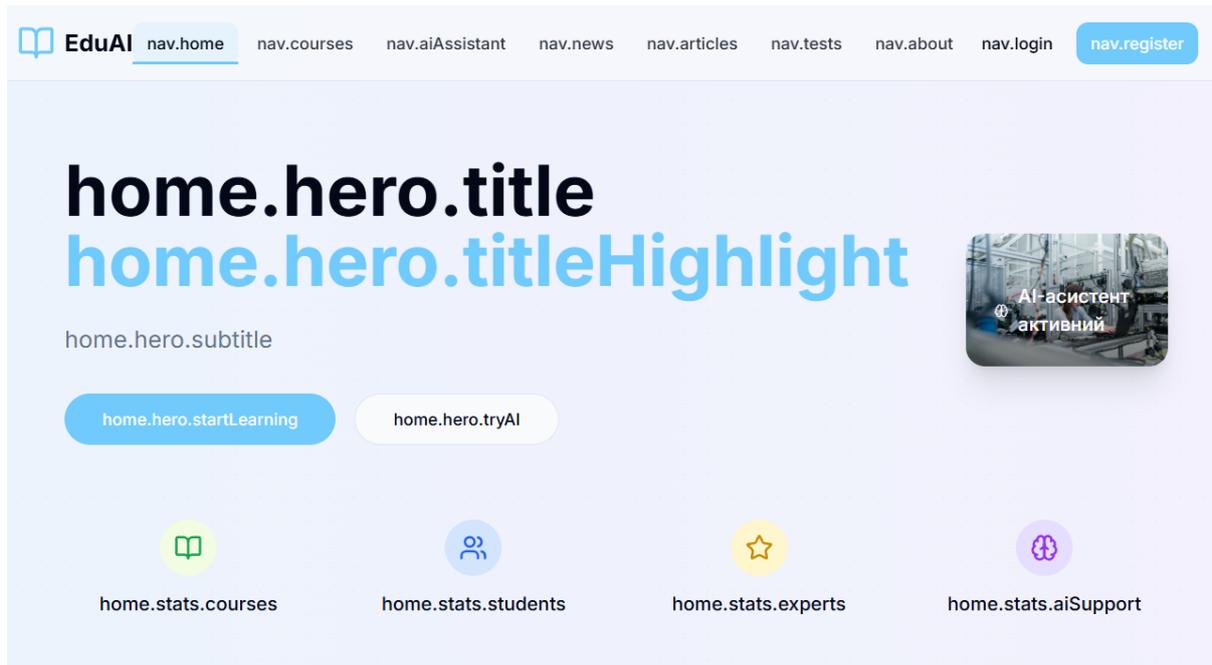


Рис.2.4.Проблеми підчас тестування продукту при додаванні функції зміни мови на англійську.

Система аналізу відповідей студентів показала нестабільні результати при обробці текстів українською мовою, що потребувало додаткового налаштування промптів та параметрів генерації. Мультиmodalний аналіз зображень працював некоректно з деякими форматами файлів та зображеннями низької якості. Тестування на різних браузерях виявило проблеми сумісності з Hugging Face Transformers, особливо в Safari та старших версіях Chrome. WebGPU підтримка була недоступна на багатьох пристроях, що змушувало систему переключатися на CPU режим з значно повільнішою продуктивністю. Інтерфейс користувача показав проблеми адаптивності на мобільних пристроях, особливо при роботі з чат-компонентами та завантаженням файлів. Система перемикання мов іноді не зберігала налаштування користувача між сесіями.

```
// Приклад з geminiService.ts
setApiKey(apiKey: string) {
  this.apiKey = apiKey;
  localStorage.setItem('gemini_api_key', apiKey);
  (частина з коду яка відповідає за частичну безпеку даних)
```

Шифрування даних застосовується на всіх рівнях зберігання та передачі інформації. Персональні дані користувачів шифруються в базі даних з використанням AES-256 алгоритму. Завантажені методичні матеріали та результати тестів також зберігаються в зашифрованому вигляді. Передача даних між клієнтом та сервером здійснюється виключно через HTTPS з сучасними TLS протоколами.

Система контролю доступу базується на рольовій моделі (RBAC), що забезпечує надання користувачам мінімально необхідних прав для виконання їх функцій. Студенти мають доступ тільки до своїх даних та призначених їм курсів. Викладачі можуть переглядати дані студентів тільки в межах своїх курсів. Адміністратори мають розширені права, але їх дії логуються та аудитуються.

Локальне зберігання даних в браузері користувача здійснюється з використанням шифрування для захисту від несанкціонованого доступу. Кешовані AI-моделі та тимчасові дані шифруються перед збереженням в

IndexedDB або localStorage. Система також включає автоматичне очищення локальних даних при завершенні сесії або за запитом користувача.

Аудит та моніторинг безпеки включають систему логування всіх критичних операцій з персональними даними. Система відстежує спроби несанкціонованого доступу, підозрілу активність та порушення політик безпеки. Автоматичні сповіщення надсилаються адміністраторам при виявленні потенційних інцидентів безпеки.

Захист від веб-вразливостей реалізований через комплекс заходів, що включають валідацію та санітизацію всіх користувацьких вхідних даних, захист від XSS атак через Content Security Policy, запобігання CSRF атакам через токени та Origin перевірки. Система також включає rate limiting для запобігання DDoS атакам та brute force спробам.

Конфіденційність AI-обробки забезпечується через використання локальних моделей без передачі даних на зовнішні сервіси. Всі запити до AI-помічника обробляються в браузері користувача, а результати аналізу не зберігаються без явної згоди користувача. Система включає можливості для повного видалення персональних даних на вимогу користувача.

Резервне копіювання та відновлення даних реалізовано з дотриманням принципів безпеки. Backup дані шифруються перед зберіганням та мають обмежений час зберігання. Доступ до резервних копій контролюється та аудитується. Система відновлення включає перевірку цілісності даних та можливість вибіркового відновлення.

Політика конфіденційності та згода користувачів реалізована відповідно до вимог GDPR та українського законодавства. Користувачі отримують детальну інформацію про збір, обробку та використання їх персональних даних. Система включає механізми для відкликання згоди, запиту копії даних та їх повного видалення.

Безпека мобільних додатків включає додаткові заходи захисту, специфічні для мобільних платформ. Це включає перевірку цілісності додатка, захист від reverse engineering, безпечне зберігання даних в keychain/keystore та SSL pinning для запобігання man-in-the-middle атакам.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПЛАТФОРМИ

3.1. Реалізація основних функцій

Основні функції веб-платформи для навчання з використанням штучного інтелекту охоплюють широкий спектр освітніх потреб, від базового управління курсами до складної AI-аналітики навчального процесу. Кожна функція розроблена з урахуванням потреб різних груп користувачів - студентів, викладачів та адміністраторів, забезпечуючи комплексне рішення для сучасної освіти.

Система управління курсами представляє центральну функціональність платформи, що дозволяє створювати, організовувати та керувати навчальним контентом. Викладачі можуть структурувати курси за модулями та темами, додавати різноманітні типи контенту включаючи текст, відео, інтерактивні елементи та зовнішні ресурси. Система підтримує версіонування контенту, що дозволяє відстежувати зміни та повертатися до попередніх версій при необхідності.

Інтерактивні навчальні інструменти включають широкий спектр активностей для залучення студентів у навчальний процес. Система підтримує створення вікторин, форумів для обговорення та групових проєктів. Кожен інструмент інтегрований з AI-аналітикою для відстеження ефективності та надання персоналізованих рекомендацій.

```
pdfjsLib.GlobalWorkerOptions.workerSrc = '//cdnjs.cloudflare.com/ajax/libs/pdf.js/1.10.0/pdf.worker.min.js';
return pdfjsLib;
};

interface UploadedFile {
  id: string;
  name: string;
  content: string;
  type: string;
  uploadDate: Date;
}

interface FileUploadProps {
  onFilesChange: (files: UploadedFile[]) => void;
  uploadedFiles: UploadedFile[];
}

const FileUpload = ({ onFilesChange, uploadedFiles }: FileUploadProps) => {
  const [isDragging, setIsDragging] = useState(false);
  const [isProcessing, setIsProcessing] = useState(false);
  const { toast } = useToast();
  const { t } = useLanguage();

  const extractTextFromPdf = async (file: File): Promise<string> => {
    try {
      const pdfjsLib = await loadPdfjs();
      const arrayBuffer = await file.arrayBuffer();
      const pdf = await pdfjsLib.getDocument(arrayBuffer).promise;
      let fullText = '';
    }
  }
}
```

Рис.3.1.завантаження док файлів для AI

Функціональність оцінювання та тестування забезпечує створення різноманітних типів завдань - від простих тестів множинного вибору до складних проектних робіт. Система підтримує автоматичне оцінювання об'єктивних завдань та AI-асистоване оцінювання відкритих відповідей. Результати тестування автоматично аналізуються для виявлення областей, що потребують додаткової уваги.

Система комунікації та співпраці включає інструменти для взаємодії між учасниками навчального процесу. Інтегрований месенджер дозволяє приватне спілкування, форуми забезпечують групові дискусії, а система оголошень дозволяє викладачам ефективно інформувати студентів про важливі події та зміни в курсах.

Аналітичні функції надають детальну інформацію про навчальний прогрес та ефективність курсів. Дашборди для студентів показують індивідуальний прогрес, досягнення та рекомендації для покращення. Викладацькі панелі включають аналітику по групах, ефективності окремих завдань та загальній статистиці курсу. Адміністративні звіти надають інформацію про використання платформи та KPI навчального процесу.

Функція завантаження jpeg та png файлів.

```

import React, { useState, useRef } from "react";
import { Button } from "@components/ui/button";
import { Card, CardContent } from "@components/ui/card";
import { Badge } from "@components/ui/badge";
import { X, Image, Upload } from "lucide-react";
import { useToast } from "@hooks/use-toast";
import { useLanguage } from "@contexts/LanguageContext";
import type { UploadedImage } from "@services/geminiService";

interface ImageUploadProps {
  images: UploadedImage[];
  onImagesChange: (images: UploadedImage[]) => void;
}

const ImageUpload = ({ images, onImagesChange }: ImageUploadProps) => {
  const [isDragging, setIsDragging] = useState(false);
  const { toast } = useToast();
  const { t } = useLanguage();
  const fileInputRef = useRef<HTMLInputElement>(null);

  const handleImageUpload = async (files: FileList | File[]) => {
    const maxFileSize = 5 * 1024 * 1024; // 5MB
    const maxImages = 5;

    if (images.length + files.length > maxImages) {
      toast({
        title: "Забагато зображень",
        description: `Макимум ${maxImages} зображень`,
        variant: "destructive",
      });
      return;
    }
  }
}

```

Рис.3.1.1.код для додавання на перевірку jpeg,png файлів для AI

Мобільна функціональність забезпечує повний доступ до платформи з мобільних пристроїв через адаптивний веб-інтерфейс. Студенти можуть переглядати матеріали, виконувати завдання та взаємодіяти з AI-помічником з будь-якого пристрою. Офлайн-режим дозволяє завантажувати контент для роботи без інтернет-з'єднання.

Система інтеграції забезпечує з'єднання з зовнішніми освітніми платформами та інструментами. Підтримуються стандарти SCORM та xAPI для імпорту існуючого контенту, інтеграція з календарними системами для синхронізації розкладу та API для підключення сторонніх додатків.

3.1.1. Налаштування ШІ-алгоритмів

Налаштування AI-алгоритмів у веб-платформі для навчання представляє комплексний процес конфігурації, оптимізації та адаптації машинного навчання під специфічні потреби освітнього середовища. Процес включає первинну ініціалізацію моделей, їх калібрування для конкретних предметних областей та постійне навчання на основі реальних даних навчального процесу.

Початковий етап налаштування включає конфігурацію базових параметрів AI-моделей, таких як температура генерації тексту, максимальна довжина відповідей та поріг впевненості для класифікації. Модель DistilGPT2 налаштовується з температурою 0.7 для збалансування креативності та точності відповідей. Параметри fine-tuning визначаються на основі специфіки навчальних матеріалів та очікуваного стилю взаємодії з студентами.

Калібрування моделей для предметних областей включає адаптацію словника та контексту під конкретні дисципліни. Для технічних предметів система налаштовується на розпізнавання специфічної термінології та математичних виразів. Гуманітарні дисципліни потребують інших параметрів для аналізу аргументації та критичного мислення. Система підтримує профілі налаштувань для різних типів курсів.

```

1  import React, { useState } from "react";
2  import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@components/ui/card";
3  import { Button } from "@components/ui/button";
4  import { Textarea } from "@components/ui/textarea";
5  import { Badge } from "@components/ui/badge";
6  import { AlertCircle, CheckCircle, Lightbulb, FileSearch } from "lucide-react";
7  import { useToast } from "@hooks/use-toast";
8  import { geminiService, type UploadedFile, type TestAnalysis } from "@services/geminiService";
9
10
11 interface TestAnalysisProps {
12   methodologyFiles: UploadedFile[];
13 }
14
15 const TestAnalysisComponent = ({ methodologyFiles }: TestAnalysisProps) => {
16   const [studentAnswers, setStudentAnswers] = useState("");
17   const [analysis, setAnalysis] = useState<TestAnalysis | null>(null);
18   const [isAnalyzing, setIsAnalyzing] = useState(false);
19   const { toast } = useToast();
20
21   const handleAnalyze = async () => {
22     if (!studentAnswers.trim()) {
23       toast({
24         title: "Помилка",
25         description: "Введіть відповіді студента для аналізу",
26         variant: "destructive",
27       });
28       return;
29     }
30

```

Рис.3.1.1.1.Налаштування ШІ

Процес навчання AI-алгоритмів на локальних даних реалізований через систему incremental learning, що дозволяє моделям адаптуватися до специфічних особливостей навчального закладу без порушення конфіденційності. Алгоритми аналізують паттерни у відповідях студентів, типові помилки та ефективні методи пояснення для покращення якості майбутніх рекомендацій. Оптимізація продуктивності AI-алгоритмів включає налаштування використання обчислювальних ресурсів для забезпечення швидкої роботи на різних пристроях. Система автоматично визначає доступність WebGPU та адаптує алгоритми для оптимального використання GPU або CPU. Batch processing використовується для ефективної обробки множинних запитів одночасно.

Система валідації та тестування AI-алгоритмів включає автоматизовані тести для перевірки якості генерованих відповідей та точності аналізу. Benchmark датасети використовуються для порівняння продуктивності після змін у налаштуваннях. A/B тестування дозволяє порівнювати ефективність різних конфігурацій на реальних користувачах. Моніторинг якості роботи AI включає відстеження метрик точності, recall та F1-score для різних типів завдань. Система автоматично виявляє деградацію якості та сповіщає адміністраторів про необхідність

переналаштування. Dashboard моніторингу надає real-time інформацію про продуктивність AI-компонентів.

Персоналізація AI-алгоритмів реалізована через створення індивідуальних профілів налаштувань для кожного користувача. Система адаптує стиль спілкування, складність пояснень та тип рекомендацій на основі навчальних переваг та рівня знань студента. Machine learning алгоритми автоматично коригують ці налаштування на основі зворотного зв'язку.

Інтеграція з методичними матеріалами включає автоматичне налаштування AI-алгоритмів на основі завантажених викладачем документів. Система аналізує контент для визначення ключових концептів, стилю викладання та специфічних вимог курсу. Ці дані використовуються для адаптації алгоритмів аналізу відповідей та генерації рекомендацій.

Система feedback loop забезпечує постійне покращення AI-алгоритмів на основі оцінок користувачів та результатів навчання. Позитивні та негативні відгуки аналізуються для ідентифікації областей покращення. Успішність студентів після отримання AI-рекомендацій відстежується для оцінки ефективності алгоритмів.

Безпека та етичність AI-алгоритмів забезпечується через систему фільтрів та обмежень. Алгоритми налаштовані на уникнення генерації неприйняттого контенту, дискримінаційних відповідей або неточної інформації. Система bias detection виявляє та коригує потенційні упередження в роботі AI-моделей.

3.1.2. Тестування інтерфейсу

Тестування інтерфейсу веб-платформи для навчання представляє багаторівневий процес верифікації функціональності, юзабіліті та доступності всіх компонентів користувацького інтерфейсу. Процес включає автоматизоване тестування компонентів, мануальне тестування користувацького досвіду та спеціалізоване тестування AI-інтерфейсів для забезпечення якості взаємодії з інтелектуальними функціями.

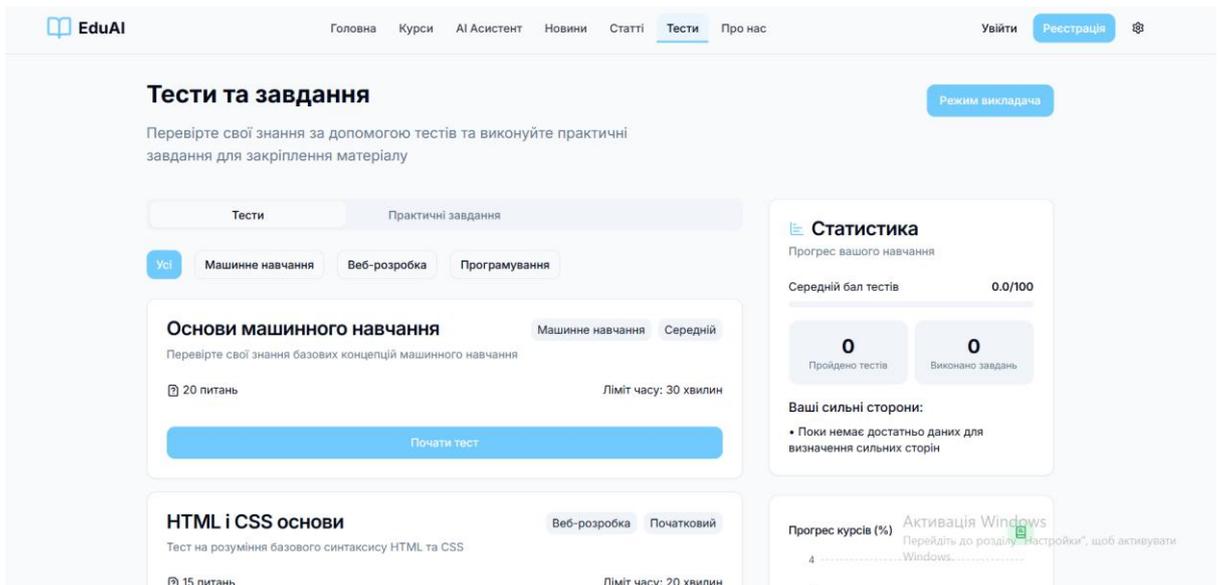


Рис.3.1.2. Тестування інтерфейсу

Функціональне тестування UI компонентів реалізовано через комплексну систему unit та integration тестів з використанням React Testing Library та Jest. Кожен компонент інтерфейсу покритий тестами, що перевіряють правильність рендерингу, обробку користувацьких подій та інтеграцію з глобальним станом додатка. Особлива увага приділяється тестуванню форм, навігації та інтерактивних елементів.

End-to-end тестування реалізовано з використанням Cypress для автоматизації тестування повних користувацьких сценаріїв. Тести покривають критичні шляхи користувача, включаючи реєстрацію, вхід в систему, навігацію по курсах, виконання завдань та взаємодію з AI-помічником. Автоматизовані тести запускаються при кожному оновленні коду для раннього виявлення регресій.

Тестування адаптивності та responsive дизайну включає перевірку коректного відображення інтерфейсу на різних розмірах екранів та пристроях. Використовуються емулятори мобільних пристроїв та планшетів для тестування touch-взаємодії, жестів та адаптації контенту. Особлива увага приділяється тестуванню AI-чату на мобільних пристроях через специфіку взаємодії з моделями.

Accessibility тестування забезпечує відповідність інтерфейсу стандартам WCAG 2.1. Автоматизовані інструменти перевіряють контрастність кольорів, наявність alt-текстів та правильність ARIA-атрибутів. Мануальне тестування з використанням screen readers перевіряє

реальний досвід користувачів з обмеженими можливостями. Клавіатурна навігація тестується для забезпечення повної функціональності без миші. Performance тестування UI включає вимірювання швидкості завантаження сторінок, часу рендерингу компонентів та responsiveness інтерфейсу під навантаженням. Lighthouse аудити регулярно проводяться для моніторингу Core Web Vitals. Особливу увагу приділено тестуванню продуктивності при завантаженні AI-моделей та обробці великих файлів.

Usability тестування включає залучення реальних користувачів для оцінки інтуїтивності інтерфейсу та ефективності виконання задач. Проводяться модеровані сесії з студентами та викладачами для збору feedback про зручність використання різних функцій. Eye-tracking дослідження допомагають оптимізувати розташування елементів інтерфейсу.

Тестування AI-інтерфейсів включає специфічні перевірки взаємодії з чат-ботом, системою аналізу відповідей та рекомендаційними алгоритмами. Тестуються різні сценарії запитів до AI, обробка помилок та fallback механізми. Перевіряється коректність відображення результатів аналізу та зрозумілість AI-генерованих рекомендацій.

Cross-browser тестування забезпечує сумісність з основними веб-браузерами включаючи Chrome, Firefox, Safari та Edge. Особлива увага приділяється тестуванню WebGPU функціональності та fallback на CPU для браузерів без підтримки WebGPU. Тестуються різні версії браузерів для забезпечення широкої сумісності.

Security тестування UI включає перевірку захисту від XSS атак, CSRF та інших веб-вразливостей. Тестується валідація користувацьких введів, санітизація контенту та правильність імплементації Content Security Policy. Особлива увага приділяється безпеці завантаження файлів та взаємодії з AI-компонентами.

Regression тестування автоматизовано для виявлення непередбачених змін у поведінці інтерфейсу після оновлень. Система snapshot тестування фіксує візуальний стан компонентів та виявляє неочікувані зміни у зовнішньому вигляді. Автоматизовані тести запускаються в CI/CD pipeline для забезпечення стабільності релізів.

3.2. Оптимізація продуктивності

Оптимізація продуктивності веб-платформи для навчання з AI-функціональністю представляє критично важливий аспект, що безпосередньо впливає на користувацький досвід та ефективність навчального процесу. Комплексний підхід до оптимізації включає покращення швидкості завантаження, ефективність роботи AI-моделей, оптимізацію використання пам'яті та мінімізацію затримок при взаємодії з інтерфейсом.

Frontend оптимізація базується на сучасних техніках `bundle optimization` та `code splitting`. Використання `Vite` забезпечує ефективну збірку з `tree-shaking` для видалення невикористаного коду. `Lazy loading` компонентів реалізовано для зменшення розміру початкового `bundle` та прискорення першого завантаження. `Dynamic imports` використовуються для завантаження AI-моделей та великих компонентів тільки при необхідності.

Оптимізація завантаження AI-моделей включає `progressive loading` з відображенням прогресу користувачу. Моделі компресуються та завантажуються частинами для зменшення часу очікування. Система `preloading` передбачає потенційні потреби користувача та завантажує необхідні моделі у фоновому режимі. `Local caching` в `IndexedDB` забезпечує миттєве завантаження при повторних відвідуваннях.

`Memory management` для AI-операцій реалізовано через ретельний контроль використання пам'яті браузера. Система автоматично звільняє пам'ять після завершення AI-операцій та реалізує `pooling` для повторного використання ресурсів. `Monitoring` пам'яті допомагає виявляти `memory leaks` та оптимізувати алгоритми для роботи в обмежених ресурсах.

`Database оптимізація` включає індексування часто запитуваних полів, оптимізацію SQL-запитів та використання `connection pooling`. `Caching` стратегії реалізовані на кількох рівнях - `Redis` для `session data`, `application-level cache` для статичного контенту та `database query cache` для складних аналітичних запитів. `Pagination` та віртуалізація списків зменшують навантаження при роботі з великими обсягами даних.

`Network оптимізація` включає мінімізацію кількості HTTP-запитів через `bundling` ресурсів та використання `HTTP/2 multiplexing`. `Compression` алгоритми (`gzip`, `brotli`) зменшують розмір передаваних даних. `CDN` інтеграція забезпечує швидку доставку статичних ресурсів з географічно

близьких серверів. Service Workers реалізують intelligent caching та offline functionality.

WebGPU оптимізація включає ефективне використання GPU ресурсів для прискорення AI-обчислень. Система автоматично визначає оптимальні параметри для конкретного GPU та адаптує алгоритми відповідно. Batch processing дозволяє обробляти кілька запитів одночасно для максимального використання паралельних можливостей GPU.

Image та media оптимізація реалізована через adaptive format selection (WebP, AVIF), lazy loading зображень та responsive images для різних розмірів екранів. Video контент оптимізується через adaptive bitrate streaming та prefetching ключових сегментів. Компресія файлів методичних матеріалів зменшує час завантаження без втрати якості.

Real-time performance monitoring включає збір метрик Core Web Vitals, custom performance markers та user experience metrics. Dashboard моніторингу надає real-time інформацію про продуктивність різних компонентів системи. Alerting система сповіщає про деградацію продуктивності та автоматично ініціює recovery процедури.

Client-side оптимізація включає мінімізацію DOM manipulations, використання virtual scrolling для великих списків та debouncing для користувацьких введів. React оптимізації включають memo, useMemo та useCallback для запобігання зайвим re-renders. CSS оптимізація реалізована через critical CSS inline та async loading несуттєвих стилів.

Code optimization включає профілювання JavaScript коду для виявлення bottlenecks, оптимізацію алгоритмів та використання Web Workers для CPU-intensive операцій. TypeScript compilation оптимізовано для швидшої збірки та smaller output. Регулярний аудит dependencies допомагає видаляти невикористані бібліотеки та оновлювати до більш ефективних версій.

3.3. Збір зворотного зв'язку від користувачів

Система збору зворотного зв'язку від користувачів веб-платформи для навчання з AI-функціональністю представляє комплексну методологію отримання, аналізу та використання feedback для постійного покращення навчального досвіду. Підхід включає множинні канали збору інформації, автоматизовані системи аналізу та механізми швидкого реагування на потреби користувачів. Багатоканальна система збору feedback включає інтегровані в інтерфейс форми оцінки, всплуючі опитування після завершення ключових дій, email-кампанії з деталізованими анкетами та focus групи з репрезентативними користувачами. Кожен канал оптимізовано для збору специфічного типу інформації - від швидких емоційних реакцій до детального аналізу проблем юзабіліті.

Contextual feedback система дозволяє користувачам залишати коментарі безпосередньо в контексті їх діяльності на платформі. Студенти можуть оцінювати корисність AI-рекомендацій одразу після їх отримання, коментувати складність завдань під час їх виконання та надавати feedback про якість згенерованих AI відповідей в режимі реального часу.

Automated sentiment analysis обробляє текстові відгуки користувачів для виявлення загальних настроїв та емоційних реакцій. NLP алгоритми категоризують feedback за темами, виявляють найчастіші скарги та пропозиції, ідентифікують тренди в користувацькому досвіді. Система автоматично пріоритизує критичні проблеми для негайного розгляду командою розробки.

Behavioral analytics доповнює прямий feedback аналізом поведінкових патернів користувачів. Система відстежує шляхи навігації, час проведений на різних сторінках, частоту використання AI-функцій та точки, де користувачі найчастіше залишають платформу. Heat maps та session recordings надають візуальне уявлення про взаємодію користувачів з інтерфейсом.

A/B testing framework дозволяє систематично тестувати зміни в інтерфейсі та функціональності на основі користувацького feedback. Різні групи користувачів отримують різні версії функцій для порівняння ефективності. Статистичний аналіз результатів забезпечує data-driven рішення про впровадження змін.

Continuous feedback loop забезпечує регулярне інформування користувачів про впроваджені зміни на основі їх пропозицій. Changelog та notification система показують, як feedback конвертується в реальні покращення платформи. Це підвищує залученість користувачів у процес покращення та стимулює надання більш детального feedback.

Personalized feedback requests адаптують запити на зворотний зв'язок під конкретних користувачів та їх поведінку. Система ідентифікує оптимальні моменти для запиту feedback на основі користувацької активності та настрою. Персоналізовані питання фокусуються на релевантних для конкретного користувача аспектах платформи.

Academic research integration включає співпрацю з освітніми дослідниками для проведення більш глибоких студій користувацького досвіду. Longitudinal studies відстежують зміни в навчальних результатах та задоволеності користувачів протягом тривалих періодів. Результати академічних досліджень інформують стратегічні рішення про розвиток платформи.

Feedback prioritization system автоматично категоризує та пріоритизує отриманий feedback на основі факторів впливу на користувацький досвід, кількості користувачів, які зіткнулися з проблемою, та стратегічної важливості для бізнес-цілей. Критичні проблеми безпеки та доступності отримують найвищий пріоритет для негайного вирішення.

Response та communication strategy забезпечує своєчасне та прозоре спілкування з користувачами щодо їх feedback. Автоматизовані відповіді підтверджують отримання feedback, персоналізовані повідомлення інформують про статус розгляду пропозицій, а публічні roadmaps показують заплановані покращення на основі користувацьких запитів.

Лістинг Коду

Це React Router DOM(навігація) його функція це маршрутизація між сторінками та SPA навігація

```
// Файл: src/App.tsx
```

```
<Routes>
  <Route path="/" element={<Index />} />
  <Route path="/ai-assistant" element={<AIAssistant />} />
</Routes>
```

TanStack React Query (state management) його функціонал це кешування запитів, синхронізація даних та оптимістичні оновлення

```
// Файл: src/App.tsx
```

```
const queryClient = new QueryClient();
<QueryClientProvider client={queryClient}>
```

Функціонал: Utility-first CSS фреймворк, responsive design

```
// Файл: всі компоненти
```

```
className="bg-primary text-primary-foreground p-4 rounded-lg"
```

Tailwind CSS (styling)

Функціонал: Готові компоненти на базі Radix UI + Tailwind

```
// Файл: src/components/ui/*
```

```
import { Button } from "@components/ui/button";
import { Card, CardContent } from "@components/ui/card";
```

Shadcn/UI Components - Модульна UI бібліотека

Переваги: Доступність з коробки, TypeScript підтримка, легка кастомізація, модульність.

```
// src/components/ui/card.tsx
```

```
import { Card, CardContent, CardDescription, CardHeader, CardTitle } from
"@components/ui/card";
```

```

<Card>
  <CardHeader>
    <CardTitle>Заголовок</CardTitle>
    <CardDescription>Опис</CardDescription>
  </CardHeader>
  <CardContent>Контент</CardContent>
</Card>

```

PDF.js - Рендерінг та парсинг PDF

Функціонал: JavaScript бібліотека для роботи з PDF файлами безпосередньо в браузері. Переваги: Робота без серверних залежностей, підтримка всіх PDF форматів, витягування метаданих.

```

// src/components/ai-assistant/FileUpload.tsx
const extractTextFromPdf = async (file: File): Promise<string> => {
  const pdfjsLib = await loadPdfjs();
  const arrayBuffer = await file.arrayBuffer();
  const pdf = await pdfjsLib.getDocument(arrayBuffer).promise;

  for (let i = 1; i <= pdf.numPages; i++) {
    const page = await pdf.getPage(i);
    const textContent = await page.getTextContent();
    const pageText = textContent.items.map(item => item.str).join(' ');
    fullText += pageText;
  }
};

```

Mammoth.js - Парсинг MS Word документів

Функціонал: Конвертація .docx файлів у HTML або чистий текст. Переваги: Підтримка сучасних Word форматів, збереження форматування, витягування зображень.

```

const extractTextFromDoc = async (file: File): Promise<string> => {
  const mammoth = await loadMammoth();
  const arrayBuffer = await file.arrayBuffer();
  const result = await mammoth.extractRawText({ arrayBuffer });
};

```

```
    return result.value;
};
```

Google Gemini API - Штучний інтелект

Функціонал: API для роботи з великими мовними моделями від Google.

Переваги: Розуміння контексту, багатомовність, аналіз документів, генерація коду.

```
// src/services/geminiService.ts
async generateResponse(message: string, methodologyFiles: UploadedFile[]):
Promise<string> {
  const methodologyContext = methodologyFiles
    .map(file => `=== ${file.name} ===\n${file.content}`)
    .join('\n\n');

  const prompt = `Контекст з методичних
матеріалів:\n${methodologyContext}\n\nПитання: ${message}`;

  const response = await fetch(`${this.baseUrl}?key=${apiKey}`, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      contents: [{ parts: [{ text: prompt } ] }],
      generationConfig: {
        temperature: 0.7,
        maxOutputTokens: 1024,
      }
    })
  });
}
```

LocalStorage API - Локальне збереження даних

Функціонал: Збереження даних у браузері користувача без серверних запитів.

Переваги: Швидкий доступ, робота офлайн, персоналізація, збереження стану додатка.

```
// src/components/AIChat.tsx
const saveCurrentChat = () => {
  const newChat: ChatHistory = {
    id: Date.now().toString(),
    title,
    messages: [...messages],
    date: new Date()
  };

  const updatedHistory = [newChat, ...chatHistory.slice(0, 9)];
  setChatHistory(updatedHistory);
  localStorage.setItem('chat_history', JSON.stringify(updatedHistory));
};
```

File API - Робота з файлами

Функціонал: Читання файлів, завантажених користувачем, без відправки на сервер.

Переваги: Приватність даних, швидка обробка, підтримка різних форматів, інтуїтивний UX.

```
const handleDrop = (e: React.DragEvent) => {
  e.preventDefault();
  setIsDragging(false);
  handleFileUpload(e.dataTransfer.files);
};

const handleFileUpload = async (files: FileList) => {
  for (let i = 0; i < files.length; i++) {
    const file = files[i];
    if (file.type === 'text/plain') {
      content = await file.text();
    } else if (file.type === 'application/pdf') {
      content = await extractTextFromPdf(file);
    }
  }
}
```

```
};
```

Fetch API - HTTP комунікація

Функціонал: Сучасний стандарт для виконання HTTP запитів замість XMLHttpRequest.

Переваги: Нативна підтримка Promise, гнучкість налаштувань, потокове читання, краща продуктивність.

```
const response = await fetch(`${this.baseUrl}?key=${apiKey}`, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify(requestData),
});
```

```
if (!response.ok) {
  const errorData = await response.json();
  throw new Error(`API помилка: ${errorData.error?.message}`);
}
```

```
const data = await response.json();
```

ВИСНОВКИ

Розроблена освітня платформа демонструє успішну інтеграцію сучасних технологій штучного інтелекту в освітній процес. Використання хмарних сервісів Google Gemini забезпечує високоякісну обробку природної мови та мультимодальний аналіз, що значно підвищує якість взаємодії між студентами та навчальними матеріалами.

Гібридний підхід до обробки даних, що поєднує хмарні та локальні рішення, створює оптимальний баланс між функціональністю та безпекою. Локальна обробка через Hugging Face Transformers забезпечує приватність даних, тоді як хмарні сервіси надають доступ до найновіших моделей штучного інтелекту.

Мультимовна підтримка та адаптивний дизайн роблять платформу доступною для широкої аудиторії, що особливо важливо в контексті глобалізації освіти. Модульна архітектура забезпечує легкість підтримки та розширення функціональності в майбутньому.

Система персоналізованих рекомендацій створює основу для індивідуалізації навчального процесу, хоча для повної реалізації потенціалу потребує інтеграції з базою даних та більш складними алгоритмами машинного навчання. Проект відкриває перспективи для подальшого розвитку в напрямку створення повноцінної адаптивної освітньої екосистеми з використанням передових технологій аналізу даних та штучного інтелекту. Впровадження таких рішень може суттєво підвищити ефективність дистанційного навчання та забезпечити персоналізований підхід до освіти для кожного студента.

Список використаних джерел

1. Що таке Hugging face
URL:(https://youtu.be/GLO5FZzfrS0?si=_yqZzTGjhOTJM6Ql)
(дата звернення 17.02.2025)
2. Курс Hugging face в 20 хвилин
URL:(<https://youtu.be/nNIlnKuCNcI?si=U3KUZ48mVj8Zj3DX>)
(дата звернення 12.05.2025)
3. Алгоритми для машинного вивчення пояснені в 17 хвиликах
URL:(<https://youtu.be/E0Hmnixke2g?si=qNgoOHZ3XFSWsgzA>)
(дата звернення 02.12.2024)
4. Всі концепти в React пояснені за 12 хвилин
URL:(<https://youtu.be/wIyHSOugGGw?si=GPQRDQRPU4d8lp3k>)
(дата звернення 02.12.2024)
5. React для новачків
URL:(https://youtu.be/hn80mWvP-9g?si=hPlW9tlW_RIpbWf2)
(дата звернення 06.12.2024)
6. React пояснення за 10 хвилин
URL:(<https://youtu.be/s2skans2dP4?si=CS-bfv7K66sonynl>)
(дата звернення 06.12.2024)
7. Як підлаштувати React в 2025 році
URL:(https://youtu.be/VZ6Qpe3sjhM?si=NA0MOTUKXsl_eobz)
(дата звернення 23.05.2025)
8. Що таке NPM і для чого він потрібен
URL:(<https://youtu.be/P3aKRdUyr0s?si=A9XxLwvN2RC38fj6>)
(дата звернення 23.05.2025)
9. 7 уроків React які б хотів би знати раніше
URL:(<https://youtu.be/4AXQgOcl1mo?si=LFbMXmLSRRrJNa0W>)
(дата звернення 15.12.2024)
10. Як додати AI до веб сайту
URL:(<https://youtu.be/Ltb6swW-md4?si=OV67knDrzINYYOUW>)
(дата звернення 15.02.2025)
11. Топ 5 навчальних онлайн платформ за 2024 рік
URL:(https://youtu.be/wY5n3uGZ6Js?si=GZzTXbbf3d3bGz_h)
(дата звернення 01.08.2024)

12. Я спробував 50 курсів по програмуванню
URL:(https://youtu.be/dq1O6_5rZSQ?si=n6sON6zviZmPndm7)
(дата звернення 01.08.2024)
13. Курс по Google AI для початківців
URL:(<https://youtu.be/Yq0QkCxoTHM?si=HEcdmXNgpF4qgTXk>)
(дата звернення 01.08.2024)
14. Найкращі онлайн платформи по курсам в 2025
URL:(<https://youtu.be/UqYj1YkInTY?si=vtrZsPFZVxPdt8F9>)
(дата звернення 02.08.2024)
15. Що таке AI
URL:(https://youtu.be/Hdd9tcs6Jwk?si=F4z4yyN0_eSQgi6I)
(дата звернення 02.08.2024)
16. Топ 4 онлайн платформи для навчання , яка з них найкраща
URL:(<https://youtu.be/DgvKFxjujp0?si=5wD7hGYyhcJyYo3d>)
(дата звернення 04.08.2024)
17. Typescript проти Javascript в 2025
URL:(<https://youtu.be/HCXPJmtV47I?si=TcopmpgjKNGrgMb9>)
(дата звернення 07.09.2024)
18. Typescript для новачків
URL:(<https://youtu.be/d56mG7DezGs?si=6p5hEPaePMUeSDHi>)
(дата звернення 08.09.2024)
19. Typescript документації
URL:(<https://www.typescriptlang.org/>)
(дата звернення 08.09.2024)
20. Тренди UI/UX в 2025
URL:(<https://youtu.be/XX2gZTGwZWc?si=v3MSUhhgAMxzXshs>)
(дата звернення 11.02.2025)
21. Що таке WEBGPU
URL:(<https://youtu.be/oIur9NATg-I?si=xuyou5KGk1970caC>)
(дата звернення 18.04.2025)