

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

„До захисту допущена”
Зав. кафедри комп’ютерних
наук та прикладної математики
д.т.н., професор Турбал Ю.В.

_____ 2025 р.

БАКАЛАВРСЬКА РОБОТА

на тему:

РОЗРОБКА САЙТУ ГОЛЬФ-КЛУБУ НА ОСНОВІ PHP, HTML, CSS, JS

Виконав: Булавський Іван Сергійович
(прізвище, ім’я, по батькові)

(підпис)

студент навчально-наукового інституту кібернетики, інформаційних
технологій та інженерії

група ІІЗ-41інт

Керівник: ст. викладач Харів Наталія Олексіївна
(науковий ступінь, вчене звання, посада, прізвище та ініціали)

(підпис)

Рівне – 2025

ЗМІСТ

РЕФЕРАТ.....	3
ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ТА ПРОЕКТУВАННЯ СИСТЕМИ.....	10
1.1. Аналіз предметної області та існуючих рішень.....	10
1.2. Вибір та обґрунтування технологій.....	14
1.3. Архітектура та структура сайту.....	22
1.4. Дизайн та користувацький інтерфейс.....	27
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ФРОНТЕНД ЧАСТИНИ.....	31
2.1. Розробка структури єдиної сторінки за допомогою HTML.....	31
2.2. Створення візуального оформлення за допомогою CSS.....	32
2.3. Додавання інтерактивності за допомогою JavaScript.....	34
РОЗДІЛ 3. РЕАЛІЗАЦІЯ БЕКЕНД ЧАСТИНИ.....	35
3.1. Налаштування серверного середовища.....	35
3.2. Тестування та розгортання.....	36
ВИСНОВКИ.....	37
ЛІТЕРАТУРА.....	39
ДОДАТКИ.....	40

РЕФЕРАТ

Кваліфікаційна робота: 45 с., 7 рисунків, джерел, 1 додаток

Мета роботи – розробити та реалізувати вебсайт гольф-клубу, що забезпечує інформаційну підтримку, можливість онлайн-бронювання послуг, а також збір базової статистики відвідуваності.

Об'єкт дослідження – процеси створення інтерактивних вебдодатків, спрямованих на автоматизацію взаємодії користувачів з послугами спортивно-оздоровчих закладів.

Предмет дослідження – архітектура та технології розробки односторінкових вебсайтів (Single Page Application principles), що поєднують сучасний фронтенд з мінімалістичним, але функціональним бекендом.

Методи вивчення: застосування сучасних веб-технологій, включаючи HTML5 для структурування контенту, CSS3 для візуального оформлення та адаптивного дизайну, JavaScript для забезпечення інтерактивності та асинхронної взаємодії (AJAX) з сервером, а також PHP та MySQL для реалізації серверної логіки та зберігання даних.

У результаті досліджень розроблено та реалізовано повноцінний односторінковий вебсайт гольф-клубу. Зокрема, створено користувацький інтерфейс, що дозволяє отримувати інформацію про клуб, його послуги та галерею, а також використовувати форму онлайн-бронювання часу для гри або уроків. Розроблено та впроваджено бекенд-систему на PHP та MySQL для обробки запитів на бронювання та ведення обліку відвідувань сайту. Проведена серія тестів, яка підтвердила коректність роботи всіх інтегрованих модулів. Отримані результати дозволяють підвищити ефективність взаємодії гольф-клубу з клієнтами та надати зручний інструмент для управління основними послугами.

Ключові слова: ВЕБСАЙТ, ГОЛЬФ-КЛУБ, ОНЛАЙН-БРОНЮВАННЯ, ФРОНТЕНД, БЕКЕНД, HTML, CSS, JAVASCRIPT, PHP, MYSQL.

ВСТУП

Актуальність теми:

В сучасних умовах стрімкого розвитку інформаційних технологій та глобальної діджиталізації, наявність ефективної онлайн-присутності є критично важливою для успіху будь-якого бізнесу, включаючи спортивні та розважальні заклади, такі як гольф-клуби. Вебсайт виступає не лише як візитна картка, а й як потужний інструмент для комунікації з цільовою аудиторією, залучення нових клієнтів, утримання існуючих членів, а також для автоматизації внутрішніх бізнес-процесів.

Для гольф-клубів, які пропонують широкий спектр послуг – від оренди полів та уроків до проведення турнірів та корпоративних заходів – вебсайт стає централізованою платформою для надання інформації, демонстрації інфраструктури, керування бронюваннями та взаємодії з клієнтами.

Успішний вебсайт гольф-клубу здатен значно покращити рівень сервісу, оптимізувати операційну діяльність та розширити географію залучення аудиторії, забезпечуючи зручний доступ до всієї необхідної інформації 24/7.

Актуальність розробки такого ресурсу підкреслюється також зростанням кількості користувачів, які віддають перевагу онлайн-бронюванню та пошуку інформації в інтернеті. Сучасний вебсайт повинен бути не лише візуально привабливим, але й функціональним, інтуїтивно зрозумілим, адаптивним для різних пристроїв (десктоп, планшети, смартфони) та безпечним. Відсутність або застарілий вебсайт може призвести до втрати потенційних клієнтів, зниження конкурентоспроможності та ускладнення комунікації. Таким чином, розробка сучасного та повнофункціонального сайту для гольф-клубу є актуальним завданням, яке сприятиме його стабільному розвитку та зміцненню позицій на ринку спортивно-розважальних послуг.

Мета та завдання роботи:

Метою даної дипломної роботи є розробка повнофункціонального вебсайту для гольф-клубу з використанням технологій PHP, HTML, CSS та JavaScript, який забезпечить ефективну взаємодію з клієнтами та автоматизацію ключових бізнес-процесів клубу.

Для досягнення поставленої мети було визначено наступні завдання:

Провести аналіз предметної області та існуючих рішень: Дослідити функціонал та особливості вебсайтів інших гольф-клубів, виявити основні вимоги до функціоналу та дизайну майбутнього сайту, а також визначити потреби цільової аудиторії та адміністрації клубу.

Вибрати та обґрунтувати технологічний стек: Обґрунтувати вибір PHP як мови програмування для бекенду, HTML для структурування контенту, CSS для стилізації та JavaScript для реалізації інтерактивних елементів на клієнтській стороні, а також системи керування базами даних.

Розробити архітектуру та структуру сайту: Створити логічну структуру вебсайту, розробити карту сайту, визначити основні розділи та сторінки, а також спроектувати структуру бази даних з урахуванням усіх необхідних сутностей (користувачі, бронювання, поля, новини, події тощо).

Спроектувати дизайн та користувацький інтерфейс: Розробити привабливий та інтуїтивно зрозумілий дизайн, що відповідає естетиці гольф-клубу, з урахуванням принципів UX/UI та забезпечити адаптивність для коректного відображення на різних пристроях.

Реалізувати фронтенд частину вебсайту: Створити візуальне представлення сайту за допомогою HTML та CSS, забезпечити інтерактивність та динамічність за допомогою JavaScript.

Розробити бекенд частину та взаємодію з базою даних: Імплементувати серверну логіку за допомогою PHP, розробити механізми взаємодії з базою даних для зберігання та обробки інформації, включаючи систему реєстрації/авторизації, систему бронювання, управління контентом тощо.

Провести тестування розробленого функціоналу: Здійснити комплексне тестування всіх компонентів системи для виявлення та усунення помилок, перевірити функціональність, юзабіліті та безпеку.

Здійснити розгортання сайту та підготувати рекомендації щодо його підтримки: Розгорнути готовий вебсайт на хостингу та надати рекомендації щодо подальшого обслуговування, оновлень та масштабування.

Об'єкт та предмет дослідження:

Об'єктом дослідження є процес розробки інформаційної системи, що відповідає вимогам сучасного веб-середовища. У контексті даної роботи об'єктом є повний цикл створення вебсайту, починаючи від етапу аналізу та проектування, і закінчуючи реалізацією, тестуванням та розгортанням. Цей процес охоплює методології, підходи та інструменти, які використовуються для перетворення вимог замовника у функціональний програмний продукт.

Предметом дослідження є архітектурні рішення, програмні засоби та технології, що застосовуються для створення вебсайту гольф-клубу. Зокрема, досліджуються особливості та ефективність використання:

- Мови розмітки гіпертексту HTML5 для формування структури веб-сторінок.
- Каскадних таблиць стилів CSS3 для візуального оформлення та забезпечення адаптивності інтерфейсу.
- Мови програмування JavaScript для реалізації клієнтської логіки, інтерактивності та динамічної взаємодії з користувачем.
- Мови програмування PHP для розробки серверної логіки, обробки даних, взаємодії з базою даних та реалізації бізнес-логіки системи.
- Системи керування реляційними базами даних (наприклад, MySQL) для ефективного зберігання, керування та доступу до даних клубу (інформація про користувачів, бронювання, послуги, новини тощо).
- Принципів та підходів до проектування користувацьких інтерфейсів (UI/UX), що забезпечують зручність, інтуїтивність та естетичну привабливість сайту.

- Методів забезпечення безпеки веб-додатків та оптимізації їхньої продуктивності.

Дослідження цих елементів дозволяє не лише створити функціональний продукт, але й глибоко проаналізувати ефективність обраних рішень та їх вплив на кінцевий результат.

Методи дослідження:

В процесі виконання даної роботи були використані наступні методи дослідження:

Аналіз системних вимог: Вивчення потреб замовника (гольф-клубу) та потенційних користувачів сайту. Використання методів збору вимог, таких як інтерв'ю, анкетування, аналіз існуючих аналогів. Формалізація функціональних та нефункціональних вимог до системи.

Структурно-функціональний аналіз: Декомпозиція системи на логічні компоненти (модулі) та визначення їх взаємозв'язків. Використання діаграм прецедентів (use cases), діаграм діяльності (activity diagrams) для візуалізації функціоналу системи.

Об'єктно-орієнтований підхід: Застосування принципів об'єктно-орієнтованого проектування та програмування (ООП) для розробки структурованого, модульного та легко розширюваного коду. Побудова діаграм класів та взаємодії об'єктів.

Проектування баз даних: Розробка логічної та фізичної моделі бази даних. Використання ER-діаграм (діаграм сутність-зв'язок) для представлення структури даних та їхніх взаємозв'язків. Застосування принципів нормалізації для уникнення надмірності та забезпечення цілісності даних.

Методи тестування програмного забезпечення: Проведення модульного, інтеграційного, системного та приймального тестування для забезпечення якості та коректності роботи всіх компонентів сайту. Використання методів чорного та білого ящика, тестування навантаження та безпеки.

Порівняльний аналіз: Порівняння різних технологій, фреймворків та інструментів розробки для обґрунтованого вибору оптимального стеку

технологій (наприклад, порівняння PHP-фреймворків, JavaScript-бібліотек, підходів до адаптивного дизайну).

Методи візуалізації даних та інтерфейсів: Використання інструментів для створення прототипів, макетів (wireframes, mockups) та графічного дизайну для розробки користувацького інтерфейсу. Застосування принципів юзабіліті та UX-дизайну для забезпечення зручності та привабливості сайту.

Методи забезпечення безпеки: Дослідження та імплементація методів захисту від поширених веб-вразливостей (SQL-ін'єкції, XSS, CSRF) та забезпечення конфіденційності даних користувачів.

Аналітичний метод: Використання аналізу для обґрунтування прийнятих рішень, оцінки ефективності та виявлення проблемних аспектів у процесі розробки.

Ці методи дозволили систематизовано підійти до вирішення поставлених завдань, забезпечити високу якість розробленого програмного продукту та сформувані науково обґрунтовані висновки.

Структура роботи

Вступ обґрунтовує актуальність обраної теми, формулює мету та завдання роботи, визначає об'єкт і предмет дослідження, описує використані методи дослідження та представляє структуру роботи.

Розділ 1 "Аналіз та проектування системи" присвячений дослідженню предметної області, аналізу існуючих рішень у галузі вебсайтів гольф-клубів, обґрунтуванню вибору технологій розробки. Також в цьому розділі розглядається архітектура майбутнього сайту, його структура та принципи проектування користувацького інтерфейсу.

Розділ 2 "Реалізація фронтенд частини" детально описує процес створення візуальної та інтерактивної частини вебсайту. Тут розглядається розробка структури сторінок за допомогою HTML, стилізація за допомогою CSS та додавання інтерактивності за допомогою JavaScript.

Розділ 3 "Реалізація бекенд частини та бази даних" охоплює розробку серверної логіки на PHP та взаємодію з базою даних. У цьому розділі

описується реалізація ключового функціоналу сайту, такого як реєстрація користувачів, система бронювання, управління контентом.

Розділ 4 "Тестування та розгортання" присвячений забезпеченню якості розробленого програмного продукту. Розглядаються методи тестування функціоналу та інтерфейсу, питання оптимізації продуктивності та безпеки, а також процедури розгортання та підтримки вебсайту.

У Висновках узагальнюються результати виконаної роботи, оцінюється досягнення поставлених мети та завдань, формулюються рекомендації щодо подальшого розвитку проекту.

РОЗДІЛ 1.

АНАЛІЗ ТА ПРОЕКТУВАННЯ СИСТЕМИ

Успішна розробка будь-якого програмного продукту, зокрема вебсайту, починається з етапу ретельного аналізу та проектування. Цей етап є фундаментом, що забезпечує відповідність кінцевого продукту вимогам замовника та потребам користувачів, а також визначає його функціональність, архітектуру та візуальне оформлення. Недостатня увага до цих початкових фаз може призвести до значних проблем на пізніших етапах розробки, включаючи переробки, перевищення бюджету та невідповідність очікуванням.

Метою цього розділу є проведення комплексного аналізу предметної області, вибір оптимального технологічного стеку, розробка детальної архітектури вебсайту та проектування користувацького інтерфейсу, що відповідатиме сучасним стандартам та забезпечуватиме зручність використання.

1.1. Аналіз предметної області та існуючих рішень

Ефективний вебсайт для гольф-клубу повинен відображати його унікальність, престиж та надавати повний спектр інформації та послуг, що відповідають потребам як постійних членів, так і потенційних відвідувачів. Аналіз предметної області включає розуміння специфіки функціонування гольф-клубу, його цільової аудиторії, а також вимог до онлайн-присутності.

1.1.1. Специфіка функціонування гольф-клубу та його цільова аудиторія

Гольф-клуб – це не просто спортивний об'єкт, а комплексний заклад, що пропонує широкий спектр послуг:

- Оренда гольф-полів: Ключова послуга, що вимагає зручної системи бронювання.
- Уроки гольфу: Індивідуальні та групові заняття з професійними інструкторами.
- Членство в клубі: Різні типи абонементів та їх переваги.

- Проведення заходів: Турніри, корпоративні заходи, приватні свята.
- Додаткові послуги: Прокат обладнання, Pro Shop (магазин), ресторан, спа-центр, готельні послуги (якщо є).
- Інфраструктура: Опис полів (кількість лунок, складність), тренувальних зон (драйвінг-рейндж, паттинг-грін), клубного будинку.

Цільова аудиторія гольф-клубу є досить різноманітною, що вимагає гнучкого підходу до структури та контенту сайту:

- Існуючі члени клубу: Потребують легкого доступу до інформації про свій абонемент, можливості бронювання, новини клубу, календар подій, результати турнірів. Для них важливо мати особистий кабінет.
- Потенційні клієнти/новачки: Шукають інформацію про гольф як вид спорту, пропозиції для початківців, ціни, контактні дані, можливість записатися на пробний урок. Для них важлива доступна інформація та легкість навігації.
- Туристи та гості: Можуть шукати інформацію про можливість разового відвідування, розташування, проживання, оренду обладнання.
- Корпоративні клієнти: Зацікавлені у проведенні заходів, тимблдінгів, корпоративних турнірів. Потребують інформації про пакети послуг для бізнесу.

Розуміння цих потреб дозволяє визначити ключовий функціонал та інформаційні розділи вебсайту.

1.1.2. Вимоги до функціоналу та дизайну майбутнього сайту

На основі аналізу специфіки клубу та його цільової аудиторії, були сформовані наступні функціональні та нефункціональні вимоги до майбутнього вебсайту:

Функціональні вимоги:

- Розділи "Про нас": історія клубу, місія, філософія, команда.
- Опис гольф-полів: характеристики кожної лунки, візуалізації (фото/відео).

- Перелік та детальний опис послуг: уроки, прокат обладнання, клубні пропозиції, членство.
- Розділ "Галерея": фото та відео з життя клубу, турнірів, заходів.
- Розділ "Новини та події": календар майбутніх турнірів, анонси, звіти про проведені заходи.
- Контактна інформація: адреса, телефон, електронна пошта, інтерактивна карта.
- Форма зворотного зв'язку.
- Система бронювання:
- Можливість онлайн-бронювання часу гри (tee time) для гольф-полів.
- Система керування користувачами: Реєстрація нових користувачів з підтвердженням електронної пошти.

Нефункціональні вимоги (як система повинна функціонувати):

- Продуктивність: Сайт повинен швидко завантажуватися (час завантаження сторінки до 3 секунд) та ефективно обробляти запити, особливо під час пікових навантажень (наприклад, при одночасному бронюванні кількома користувачами).
- Надійність: Система повинна працювати стабільно 24/7 з мінімальним часом простою (uptime не менше 99.9%). Повинні бути передбачені механізми обробки помилок та резервного копіювання даних.
- Безпека: Захист даних користувачів (хешування паролів, безпечно зберігання особистої інформації).
- Юзабіліті (зручність використання): Інтерфейс повинен бути інтуїтивно зрозумілим, навігація – логічною та простою. Всі елементи повинні бути легко доступними.
- Адаптивність (Responsive Design): Сайт повинен коректно відображатися та функціонувати на різних пристроях (настільні комп'ютери, ноутбуки, планшети, смартфони) та в різних веб-браузерах.

- Масштабованість: Архітектура системи повинна дозволити легке додавання нового функціоналу та обробку зростаючого навантаження в майбутньому.

1.1.3. Аналіз існуючих аналогів та конкурентів

Для формування комплексного уявлення про сучасні тенденції та кращі практики у розробці вебсайтів для гольф-клубів було проведено аналіз ряду вітчизняних та зарубіжних ресурсів. Метою цього аналізу було виявлення типового функціоналу, ефективних дизайнерських рішень, а також потенційних недоліків, які слід уникнути у власному проекті.

Розглянемо загальні характеристики успішних гольф-клубів онлайн, які мають наступні спільні риси:

- 1) Інформаційне наповнення: * Детальний опис гольф-полів: Більшість сайтів надають карти полів, фотографії кожної лунки, інформацію про складність та особливості ландшафту. Деякі клуби пропонують віртуальні тури. * Послуги та членство: Чітко структурована інформація про різні типи членства, уроки, програми для початківців, опис додаткових послуг (ресторан, про-шоу, готель). * Новини та події: Актуальні новини клубу, розклад турнірів, календар подій, результати змагань. Часто реалізовано у вигляді блогу або стрічки новин. * Галереї: Високоякісні фото- та відеоматеріали, що демонструють красу полів, атмосферу клубу, моменти з турнірів та заходів.
- 2) Функціональні можливості: * Система онлайн-бронювання: Це є ключовим елементом. Успішні сайти пропонують інтуїтивно зрозумілий інтерфейс для вибору дати, часу, кількості гравців, та, за можливості, вибору конкретного поля або інструктора. Часто інтегровані з платіжними системами. * Особистий кабінет: Дозволяє користувачам керувати своїми бронюваннями, переглядати історію відвідувань, оновлювати особисту інформацію, а також отримувати персоналізовані пропозиції. * Форми зворотного зв'язку: Зручні форми для запитів, бронювання спеціальних заходів або отримання консультацій. * Інтеграція з соціальними мережами: Кнопки для поширення контенту, посилення на офіційні сторінки клубу.

3) Дизайн та користувацький досвід (UX/UI): * Преміальний дизайн: Використання якісних зображень та відео, елегантних шрифтів, гармонійної кольорової палітри, що відображає статус гольф-клубу. * Інтуїтивна навігація: Чітка і логічна структура меню, що дозволяє легко знайти потрібну інформацію. * Адаптивний дизайн: Обов'язкова умова для сучасних сайтів, що забезпечує коректне відображення на всіх пристроях, від десктопів до смартфонів. * Швидкість завантаження: Оптимізовані зображення та код для швидкого відображення сторінок.

Виявлені недоліки в деяких існуючих рішеннях:

- Складність процесу бронювання: Деякі системи мають заплутаний інтерфейс або вимагають забагато кроків для завершення бронювання.
- Відсутність адаптивності: Сайти, які не оптимізовані для мобільних пристроїв, втрачають значну частину потенційної аудиторії.
- Застарілий дизайн: Візуально не привабливі сайти можуть створювати враження неактуальності або низького рівня сервісу.
- Обмежений функціонал CMS: Складність або відсутність можливості для адміністраторів самостійно оновлювати контент.
- Недостатня інформативність: Відсутність детального опису полів, послуг або умов членства.

Враховуючи вищезазначене, розроблений вебсайт гольф-клубу повинен інтегрувати кращі практики та уникнути виявлених недоліків, забезпечуючи максимальну функціональність, зручність використання та естетичну привабливість.

1.2. Вибір та обґрунтування технологій

Вибір технологічного стеку є одним із найважливіших етапів проектування вебсайту, оскільки він безпосередньо впливає на можливості реалізації функціоналу, продуктивність, масштабованість, безпеку та вартість підтримки проекту. Для розробки вебсайту гольф-клубу було обрано поєднання HTML, CSS, JavaScript для клієнтської частини та PHP з MySQL для серверної частини та бази даних. Цей вибір обґрунтований їхньою

популярністю, гнучкістю, широким спектром можливостей та розвинутою екосистемою.

1.2.1. HTML5 (HyperText Markup Language 5)

HTML є фундаментальною технологією для створення веб-сторінок, що відповідає за їхню структуру та семантику. HTML5 – це остання версія стандарту, яка принесла значні покращення та нові можливості, що робить її ідеальною для сучасних веб-додатків.

Призначення та роль: HTML5 використовується для визначення структури контенту веб-сторінки, такої як заголовки, параграфи, списки, таблиці, форми, зображення та відео. Він не відповідає за візуальне оформлення (це завдання CSS), але забезпечує логічну організацію елементів, що є критично важливим для доступності та SEO.

Основні можливості та переваги:

- Семантичні теги: Впровадження нових семантичних тегів (`<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<footer>`, `<aside>`, `<figure>`) дозволяє створювати більш зрозумілу та логічну структуру документа. Це покращує читабельність коду, сприяє кращому індексуванню пошуковими системами та робить сайт більш доступним для допоміжних технологій (наприклад, для програм читання з екрана).
- Підтримка мультимедіа: Вбудовані теги `<audio>` та `<video>` дозволяють легко інтегрувати аудіо- та відеоконтент без використання сторонніх плагінів (наприклад, Flash), що підвищує сумісність та безпеку. Для сайту гольф-клубу це дозволить розміщувати відео-тури по полях, відеоуроки, рекламні ролики.
- Форми: Розширені можливості форм з новими типами введення (`<input type="date">`, `<input type="email">`, `<input type="url">`, `<input type="number">`) та атрибутами (наприклад, `placeholder`, `required`, `pattern`) спрощують створення інтерактивних форм для бронювання та реєстрації, забезпечуючи базову валідацію на стороні клієнта.

- Canvas та SVG: Елемент `<canvas>` дозволяє малювати графіку за допомогою JavaScript, що може бути використано для динамічного відображення карт полів або інтерактивних діаграм. SVG (Scalable Vector Graphics) забезпечує якісну векторну графіку, що масштабується без втрати якості.
- Геолокація: Можливість отримання даних про місцезнаходження користувача може бути використана для відображення найближчих гольф-клубів або для надання місцевих новин.

Обґрунтування вибору: HTML5 є стандартом де-факто для сучасної веб-розробки. Його семантичні можливості покращують структуру, доступність та SEO, а вбудована підтримка мультимедіа та розширені форми роблять його ідеальним для створення насиченого контентом та функціонального сайту гольф-клубу.

1.2.2. CSS3 (Cascading Style Sheets 3)

CSS відповідає за візуальне представлення веб-сторінок, відокремлюючи стиль від структури. CSS3 є останньою версією стандарту, яка додала безліч нових можливостей для створення сучасного та адаптивного дизайну.

Призначення та роль: CSS використовується для визначення кольорів, шрифтів, розмірів, відступів, позиціонування елементів, а також для створення анімацій та переходів. Це дозволяє контролювати зовнішній вигляд сайту та забезпечувати його естетичну привабливість відповідно до брендингу гольф-клубу.

Основні можливості та переваги:

- Модульна структура: CSS3 розбитий на модулі, що дозволяє поступово впроваджувати нові можливості, не чекаючи повної специфікації.
- Гнучкі макети (Flexbox та CSS Grid): Ці потужні модулі дозволяють створювати складні та адаптивні макети сторінок з легкістю. Flexbox ідеально підходить для одномірних макетів (рядок або колонка), тоді як CSS Grid – для двомірних макетів, що робить його незамінним для

розробки складних сіток сторінок гольф-клубу (наприклад, галереї, розклад турнірів).

- Медіа-запити (Media Queries): Дозволяють застосовувати різні стилі залежно від характеристик пристрою користувача (ширина екрана, орієнтація, роздільна здатність). Це є основою для реалізації адаптивного дизайну, забезпечуючи коректне відображення сайту на мобільних пристроях, планшетах та десктопах.
- Трансформації та переходи (Transitions and Transforms): Дозволяють створювати плавні анімації та інтерактивні візуальні ефекти без використання JavaScript, що покращує користувацький досвід (наприклад, ефекти при наведенні курсору на кнопки, елементи меню).
- Псевдокласи та псевдоелементи: Розширюють можливості стилізації, дозволяючи застосовувати стилі до певних станів елементів (:hover, :active) або додавати декоративні елементи без зміни HTML-структури.
- CSS-змінні (Custom Properties): Дозволяють визначати змінні для кольорів, шрифтів тощо, що значно спрощує керування стилями та зміну теми сайту.

Обґрунтування вибору: CSS3 є незамінним для створення візуально привабливого, сучасного та адаптивного вебсайту. Його можливості Flexbox, Grid та Media Queries дозволяють ефективно реалізувати дизайн, який буде виглядати професійно на будь-якому пристрої, що критично важливо для преміум-сервісу гольф-клубу.

1.2.3. JavaScript

JavaScript – це мова програмування, яка дозволяє додавати інтерактивність та динаміку до веб-сторінок, перетворюючи статичний контент на повноцінний веб-додаток.

Призначення та роль: JavaScript виконується на стороні клієнта (у браузері користувача) і дозволяє маніпулювати HTML-елементами, реагувати на дії користувача (кліки, введення даних), надсилати асинхронні запити до

сервера (AJAX), виконувати клієнтську валідацію даних, створювати анімації та багато іншого.

Основні можливості та переваги:

- Маніпуляції з DOM (Document Object Model): JavaScript може динамічно змінювати вміст, структуру та стилі веб-сторінки після її завантаження, що дозволяє створювати інтерактивні елементи (наприклад, випадаючі меню, модальні вікна, вкладки).
- Обробка подій: Можливість реагувати на дії користувача (наприклад, кліки по кнопках, наведення миші, введення тексту в форми) дозволяє створювати динамічний та чуйний інтерфейс для бронювання чи реєстрації.
- AJAX (Asynchronous JavaScript and XML): Дозволяє обмінюватися даними з сервером без перезавантаження всієї сторінки. Це покращує користувацький досвід, роблячи завантаження контенту (наприклад, доступних слотів для бронювання, новин) швидким та безперервним.
- Клієнтська валідація форм: Перевірка введених користувачем даних безпосередньо у браузері перед відправкою на сервер, що зменшує навантаження на сервер та покращує швидкість зворотного зв'язку для користувача.
- Анімації та візуальні ефекти: Реалізація складних анімацій, слайдерів, галерей та інших динамічних візуальних елементів, які неможливо досягти лише за допомогою CSS.
- Підтримка сучасних стандартів (ES6+): Сучасні версії JavaScript (ECMAScript 2015+) надають нові синтаксичні можливості та функціонал, що робить код більш читабельним та ефективним.
- Широка екосистема: Величезна кількість бібліотек та фреймворків (jQuery, React, Vue, Angular) для прискорення розробки та розширення функціоналу. Хоча для цього проекту може бути достатньо чистого JavaScript або jQuery для спрощення DOM-маніпуляцій.

Обґрунтування вибору: JavaScript є невід'ємною частиною сучасної веб-розробки, що забезпечує інтерактивність та динамічність. Для сайту гольф-клубу він критично важливий для реалізації зручної системи бронювання, анімації інтерфейсу, валідації даних та загального покращення користувацького досвіду.

1.2.4. PHP (Hypertext Preprocessor)

PHP є однією з найпопулярніших серверних мов програмування, спеціально розробленою для веб-розробки. Вона дозволяє створювати динамічні веб-сторінки, взаємодіяти з базами даних, обробляти форми та керувати сесіями.

Призначення та роль: PHP виконується на сервері до того, як сторінка буде відправлена користувачу. Вона відповідає за всю бекенд-логіку: обробку запитів від браузера, взаємодію з базою даних (збереження, отримання, оновлення даних), реалізацію бізнес-правил (наприклад, логіка бронювання, перевірка доступності), керування користувацькими сесіями, аутентифікацію та авторизацію. Результатом виконання PHP-скриптів є HTML-код, який надсилається клієнту.

Основні можливості та переваги:

- Відкритий вихідний код та кросплатформність: PHP є безкоштовним та може працювати на різних операційних системах (Linux, Windows, macOS) та веб-серверах (Apache, Nginx), що забезпечує гнучкість у виборі середовища розробки та розгортання.
- Інтеграція з базами даних: PHP має вбудовану підтримку для роботи з великою кількістю баз даних, включаючи MySQL, PostgreSQL, SQLite, Oracle та інші. Це робить його ідеальним для розробки додатків, які потребують постійного зберігання та маніпуляції даними, таких як система бронювання гольф-клубу.
- Велика спільнота та документація: Завдяки своїй популярності, PHP має величезну спільноту розробників, що забезпечує легкий доступ до

рішень типових проблем, велику кількість бібліотек, фреймворків (Laravel, Symfony, CodeIgniter) та вичерпну офіційну документацію.

- Швидкість розробки: PHP відомий своєю простотою вивчення та швидкістю розробки, особливо для типових веб-завдань. Сучасні версії PHP (PHP 7.x і 8.x) значно покращили продуктивність та ефективність виконання коду.
- Вбудовані функції для веб-розробки: PHP має безліч вбудованих функцій для роботи з HTTP-запитами, сесіями, файлами, а також для обробки зображень та інших типових веб-завдань, що прискорює процес розробки.
- Гнучкість: PHP може бути використаний як для простих скриптів, так і для складних багатомодульних корпоративних систем.

Обґрунтування вибору: PHP є перевіреною, надійною та широко використовуваною мовою для розробки бекенду веб-додатків. Її простота, швидкість розробки, широкі можливості інтеграції з базами даних та величезна екосистема роблять її ідеальним вибором для реалізації динамічного функціоналу вебсайту гольф-клубу, включаючи систему бронювання, управління користувачами та контентом.

1.2.5. MySQL

MySQL є однією з найпопулярніших систем керування реляційними базами даних (СКБД) з відкритим вихідним кодом. Вона широко використовується у веб-розробці завдяки своїй надійності, високій продуктивності та легкій інтеграції з PHP.

Призначення та роль: MySQL використовується для зберігання, керування та швидкого доступу до структурованих даних, необхідних для функціонування вебсайту. Для гольф-клубу це може бути інформація про користувачів (логіни, паролі, контактні дані), бронювання (дата, час, поле, користувач), послуги, поля (характеристики лунок), новини, події, галерея (шляхи до зображень) тощо.

Основні можливості та переваги:

- Реляційна модель даних: Дані організовані у вигляді таблиць, що дозволяє встановлювати зв'язки між ними та забезпечувати цілісність даних (наприклад, зв'язок між користувачами та їх бронюваннями).
- Висока продуктивність: MySQL оптимізований для швидкого виконання запитів, особливо операцій читання, що є типовим для веб-додатків.
- Масштабованість: Здатність обробляти великі обсяги даних та значну кількість одночасних запитів, що дозволяє системі зростати разом з потребами гольф-клубу.
- Надійність та безпека: Підтримує транзакції, механізми відновлення даних, а також пропонує різні рівні безпеки, включаючи шифрування, права доступу користувачів та захист від ін'єкцій.
- Широка підтримка: Наявність великої спільноти, великої кількості інструментів для адміністрування (наприклад, phpMyAdmin, MySQL Workbench) та бездоганна інтеграція з PHP.
- ACID-властивості (Atomicity, Consistency, Isolation, Durability): Забезпечує надійність транзакцій, що є критично важливим для операцій бронювання та реєстрації.

Обґрунтування вибору: MySQL є фактичним стандартом для веб-проектів середнього та великого масштабу. Її швидкість, надійність, масштабованість та відмінна інтеграція з PHP роблять її ідеальним вибором для зберігання та управління всіма даними вебсайту гольф-клубу, забезпечуючи ефективне функціонування системи бронювання та управління контентом.

1.2.6. Загальний огляд технологічного стеку та його переваги

Обраний технологічний стек (HTML5, CSS3, JavaScript, PHP, MySQL) є класичною та перевіреною комбінацією, яка відома як "LAMP-стек" (Linux, Apache, MySQL, PHP/Python/Perl – хоча тут замість Apache може бути Nginx, а PHP є основною мовою). Ця комбінація забезпечує повний набір інструментів для розробки повнофункціонального, динамічного та масштабованого вебсайту.

Переваги обраного стеку:

- Надійність та стабільність: Кожна з обраних технологій має довгу історію розвитку, активну підтримку спільноти та доведену стабільність у промислових проектах.
- Гнучкість та розширюваність: Можливість додавати новий функціонал та інтегрувати сторонні сервіси (платіжні системи, API карт).
- Економічна ефективність: Більшість компонентів є відкритим вихідним кодом, що знижує витрати на ліцензування програмного забезпечення.
- Велика кількість фахівців: На ринку праці є багато розробників з досвідом роботи з цими технологіями, що спрощує підтримку та розвиток проекту.
- Широкий спектр інструментів та бібліотек: Наявність величезної кількості готових рішень, бібліотек та фреймворків значно прискорює розробку та дозволяє зосередитися на унікальних особливостях проекту.
- Адаптивність: Комбінація HTML5, CSS3 з медіа-запитами та JavaScript дозволяє створити єдиний ресурс, який буде ефективно працювати на будь-якому пристрої, забезпечуючи привабливий та зручний інтерфейс для користувачів гольф-клубу.

Таким чином, обраний технологічний стек є оптимальним для реалізації вебсайту гольф-клубу, що дозволить створити високоякісний, функціональний та масштабований програмний продукт.

1.3. Архітектура та структура сайту

Ефективна архітектура та чітка структура сайту є ключовими для забезпечення його функціональності, масштабованості, зручності використання та легкого супроводу. На цьому етапі визначаються основні компоненти системи, їх взаємодія, а також логічна організація контенту та навігації.

1.3.1. Загальна архітектура системи

Вебсайт гольф-клубу реалізується за класичною трірівневою архітектурою "клієнт-сервер-база даних", що є стандартом для динамічних

веб-додатків. Цей підхід забезпечує чітке розділення відповідальності між компонентами та сприяє гнучкості системи.

Клієнтський рівень (Frontend): Представлений веб-браузером користувача. Відповідає за відображення користувацького інтерфейсу, обробку взаємодії з користувачем та відправку запитів до сервера. Технології: HTML, CSS, JavaScript.

Серверний рівень (Backend): Основний обчислювальний центр системи. Отримує запити від клієнта, обробляє їх, взаємодіє з базою даних, виконує бізнес-логіку та формує відповідь для клієнта (зазвичай у вигляді HTML, JSON або XML). Технології: PHP, веб-сервер (наприклад, Apache або Nginx).

Рівень бази даних: Відповідає за постійне зберігання всіх даних, що використовуються системою (інформація про користувачів, бронювання, поля, новини, тощо). Технологія: MySQL.

Схема взаємодії компонентів:

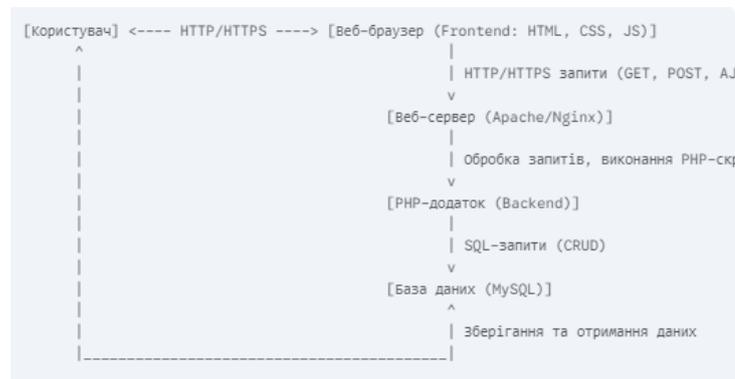


Рис 1.3.1. Схема взаємодії компонентів

Ця архітектура дозволяє масштабувати кожен рівень незалежно, підвищує безпеку (прямий доступ до бази даних з клієнта відсутній) та забезпечує гнучкість у розробці та підтримці.

1.3.2. Карта сайту та основні розділи

Карта сайту (sitemap) візуалізує ієрархію сторінок вебсайту та їх взаємозв'язки, що є основою для навігації та пошукової оптимізації. Враховуючи вимоги гольф-клубу, була розроблена наступна логічна структура:

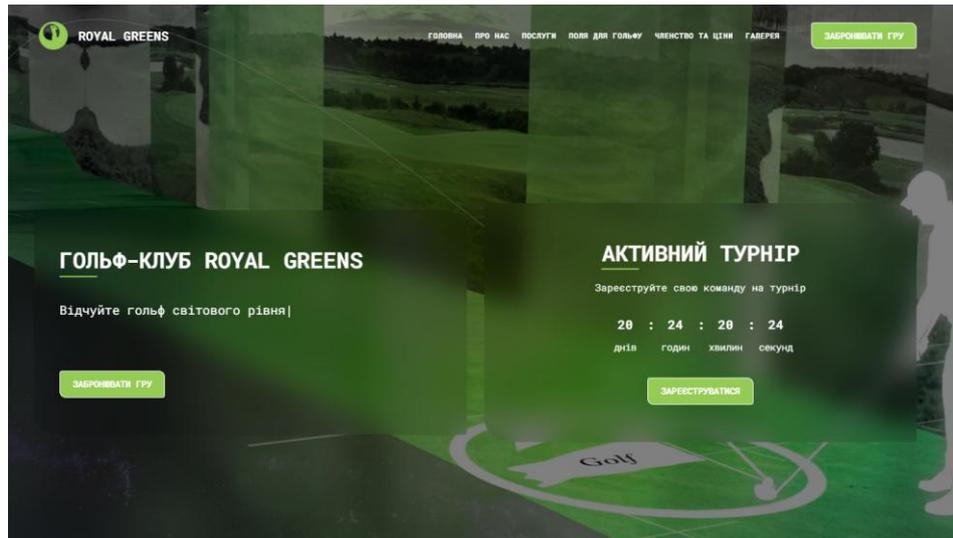


Рис 1.3.2.1 Головна секція сайту

На рисунках зображено першу секцію сайту. Можна побачити меню, кнопку “забронювати”, яка викликає модальне вікно, короткий опис гольф клубу та форму реєстрації на турнір.

ПРО НАС

Наш гольф-клуб – це місце, де традиції поєднуються з розкішю. Ідеально доглянуті поля, професійні інструктори та ексклюзивна атмосфера.

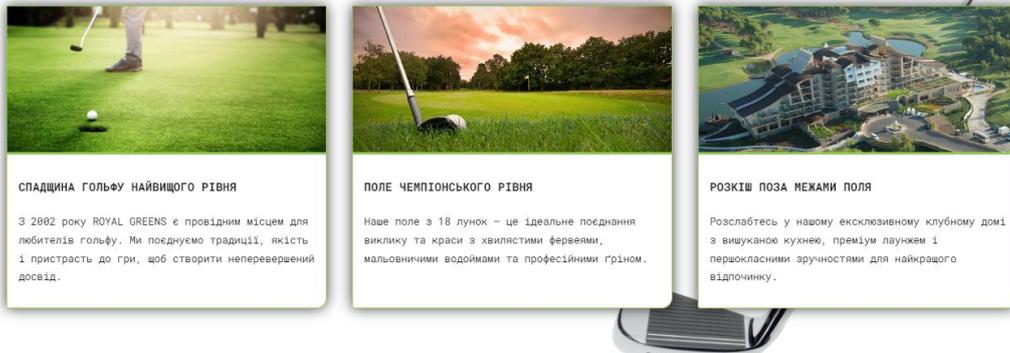


Рис 1.3.2.2 Секція “Про нас”

На рисунках зображено другу секцію сайту. Тут більш детально описано про гольф-клуб.



Рис 1.3.2.3 Секція “Послуги”

Третя секція сайту. В ній показано послуги з тематичними картинками та цікавим ефектом при наведенні.

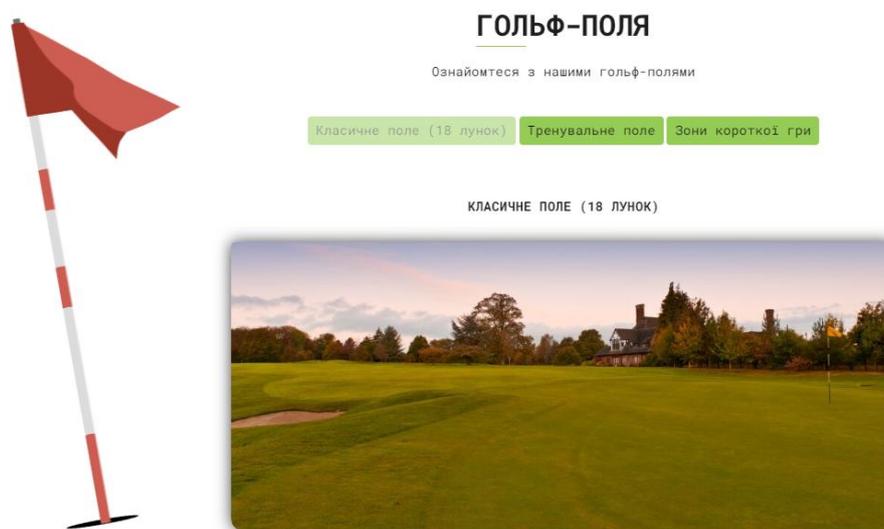


Рис 1.3.2.4 Секція “Голь-поля”

Четверта секція сайту. В ній можна побачити доступні гольф поля під замовлення. А при наведенні на картинку буде посилання на мапу поля.

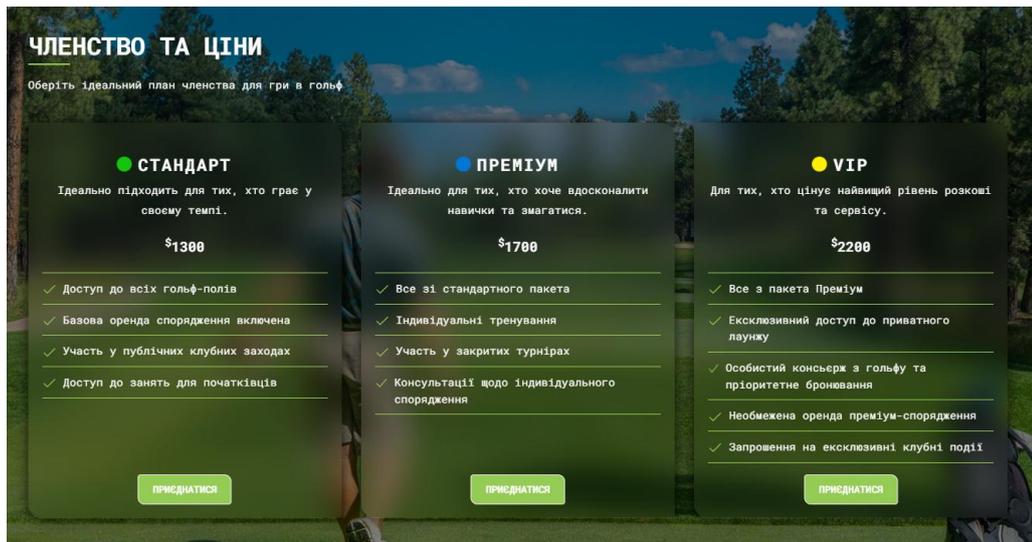


Рис 1.3.2.5 Секція “Членство та ціни”

П’ята секція сайту, в якій показано актуальні тарифи на членство в гольф-клубі. При натисканні на кнопку відкриється модальне вікно, в якому буде форма зворотнього зв’язку.

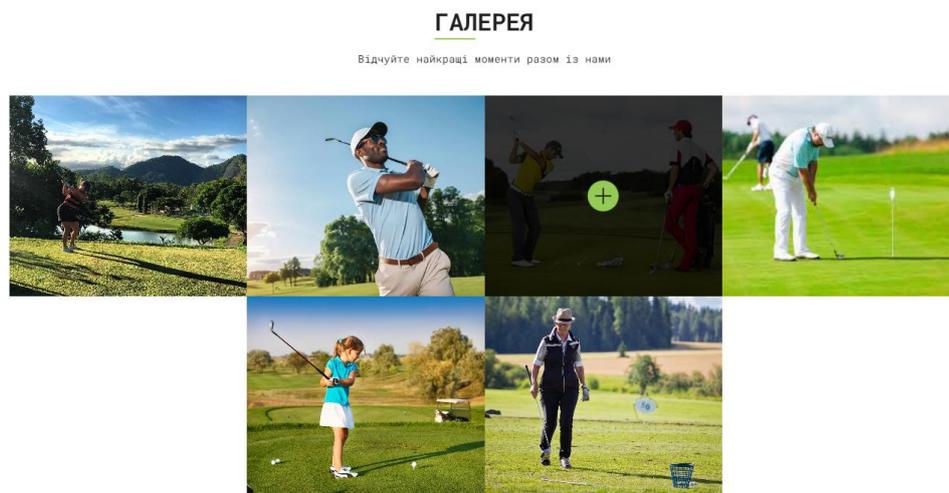


Рис 1.3.2.6 Секція “Членство та ціни”

Шоста секція сайту. Галерея з якими моментами. При натисканні на картинку вона розширюється в свій початковий розмір

Ця структура забезпечує повноту інформації та функціоналу, необхідного для сучасного гольф-клубу.

1.3.3. Структура бази даних (ER-діаграма)

Проектування бази даних є критично важливим для ефективного зберігання та управління інформацією. Була розроблена реляційна модель, що

представлена ER-діаграмою (Entity-Relationship Diagram), яка відображає сутності (таблиці) та зв'язки між ними.

1.4. Дизайн та користувацький інтерфейс

Дизайн та користувацький інтерфейс (UI/UX) відіграють ключову роль у сприйнятті вебсайту та його ефективності. Для гольф-клубу, що асоціюється з престижем, витонченістю та комфортом, візуальна складова та зручність використання є надзвичайно важливими. Метою є створення інтуїтивно зрозумілого, естетично привабливого та функціонального інтерфейсу, що відображає брендинг клубу.

1.4.1. Принципи UI/UX дизайну

При проектуванні інтерфейсу були враховані наступні ключові принципи UI/UX:

- **Центрованість на користувачі (User-Centered Design):** Весь процес проектування зосереджений на потребах, цілях та поведінці цільової аудиторії гольф-клубу (відвідувачі, члени, потенційні клієнти). Це означає, що навігація, функціонал та візуальне представлення мають бути максимально зручними та зрозумілими для них.
- **Інтуїтивність:** Інтерфейс повинен бути настільки простим, щоб користувач міг зрозуміти, як ним користуватися, без спеціальних інструкцій. Це стосується розташування елементів, логіки навігації, форм тощо.
- **Послідовність (Consistency):** Використання єдиних візуальних елементів (кольори, шрифти, іконки), шаблонів взаємодії та термінології на всіх сторінках сайту. Це зменшує когнітивне навантаження на користувача та покращує загальний досвід.
- **Зворотний зв'язок (Feedback):** Система повинна надавати користувачеві чіткий та своєчасний зворотний зв'язок на його дії (наприклад, повідомлення про успішне бронювання, помилки при введенні даних, візуальні зміни при наведенні курсору).

- Доступність (Accessibility): Проектування з урахуванням потреб користувачів з обмеженими можливостями (наприклад, використання семантичного HTML, коректний контраст кольорів, альтернативні тексти для зображень).
- Естетична привабливість (Aesthetic Appeal): Візуально приємний дизайн, що викликає позитивні емоції та відображає статус гольф-клубу. Якісні зображення, чиста типографіка, гармонійні кольори.

1.4.2. Розробка макетів та прототипів (Wireframes, Mockups, Prototypes)

Процес дизайну включав кілька етапів:

Wireframes (Каркасні макети): Це чорно-білі, низькодетальні схеми, що визначають основне розташування елементів на сторінці, без акценту на візуальному оформленні. На цьому етапі було визначено:

- Структуру хедеру (логотип, навігація, кнопки входу/бронювання).
- Розташування основних блоків контенту (слайдери, секції послуг, новин).
- Розміщення елементів управління (форми, кнопки).
- Структуру футеру.

Mockups (Візуальні макети): Це статичні, високодетальні зображення, що відображають фінальний вигляд інтерфейсу. На цьому етапі додавались:

- Кольорові рішення, що відображають брендинг клубу (переважно зелені, коричневі, золотисті відтінки, що асоціюються з гольфом та престижем).
- Вибір шрифтів (елегантні, легко читабельні).
- Розміщення та стиль іконок.
- Якісні фотографії полів, клубного будинку, гравців.
- Елементи інтерфейсу (кнопки, поля вводу, чекбокси) з урахуванням їх стилю та стану (наприклад, при наведенні курсору).

Prototypes (Інтерактивні прототипи): Це симуляції реального інтерфейсу, що дозволяють клікнути по елементах, перейти між сторінками та

протестувати основні потоки взаємодії. Це дозволяє виявити проблеми юзабіліті до початку етапу кодування.

1.4.3. Концепція дизайну та адаптивність

Концепція дизайну: Візуальний дизайн вебсайту гольф-клубу базується на принципах елегантності, престижу та функціональності. Основні елементи концепції:

Кольорова гама: Використання стриманих, природних кольорів (відтінки зеленого, синього неба, пісочного, дерев'яного коричневого), доповнених акцентними кольорами (наприклад, золотим або темно-сірим) для створення відчуття преміальності та зв'язку з природою. Скріншот, який ти надав, відображає цю палітру.

Типографіка: Використання розбірливих та естетично привабливих шрифтів. Для заголовків можуть бути обрані елегантні шрифти без зарубок або з легкими зарубками, що додають витонченості, а для основного тексту – чисті та легко читабельні шрифти без зарубок.

Зображення та відео: Використання високоякісних фотографій та відеоматеріалів гольф-полів, клубного будинку, щасливих гравців та живописної природи. Візуальний контент є ключовим для створення бажаної атмосфери.

Мінімалізм та простір: Чистий дизайн з достатньою кількістю "негативного простору" (відступів), що дозволяє елементам "дихати" та робить сторінку менш перевантаженою, фокусуючи увагу користувача на ключових деталях.

Брендування: Інтеграція логотипу клубу, фірмових кольорів та стилю в усі елементи інтерфейсу для забезпечення єдиного та впізнаваного образу.

Адаптивний дизайн (Responsive Web Design): З огляду на зростаючу популярність мобільних пристроїв, адаптивний дизайн є не вимогою, а необхідністю. Принципи реалізації:

Mobile-First підхід: Дизайн та розробка починаються з найменших екранів, а потім масштабуються до більших. Це гарантує, що основний функціонал та контент доступні для мобільних користувачів.

Гнучкі макети (Fluid Grids): Використання відсотків та `em/rem` одиниць замість фіксованих пікселів для розмірів елементів та відступів, що дозволяє їм автоматично адаптуватися до розміру екрана.

Гнучкі зображення (Flexible Images): Використання `max-width: 100%` для зображень, щоб вони ніколи не виходили за межі свого контейнера.

Медіа-запити (Media Queries): Застосування різних CSS-стилів для конкретних діапазонів ширини екрана. Це дозволяє змінювати розташування елементів, розміри шрифтів, приховувати або відображати певний контент залежно від пристрою.

Оптимізація для сенсорних екранів: Збільшення розмірів кнопок та інтерактивних елементів для зручності натискання пальцем, уникнення дрібних елементів, що важко влучити.

Реалізація цих принципів дизайну та адаптивності дозволить створити вебсайт гольф-клубу, який буде не лише функціональним, але й візуально привабливим та зручним для всіх категорій користувачів на будь-якому пристрої.

РОЗДІЛ 2. РЕАЛІЗАЦІЯ ФРОНТЕНД ЧАСТИНИ

Реалізація фронтенд частини вебсайту є етапом, на якому теоретичні концепції дизайну та структури перетворюються на інтерактивний користувацький інтерфейс. Для вебсайту гольф-клубу, що за задумом є односторінковим рішенням, цей розділ детально описує процес створення візуального представлення сайту з використанням HTML для структури, CSS для стилізації та JavaScript для забезпечення інтерактивності та динамічного функціоналу в рамках однієї сторінки. Метою є створення не лише естетично привабливого, але й функціонального, адаптивного та зручного для користувача інтерфейсу, де вся необхідна інформація доступна без переходу між окремими сторінками.

2.1. Розробка структури єдиної сторінки за допомогою HTML

HTML (HyperText Markup Language) є основою будь-якої веб-сторінки, визначаючи її логічну структуру та вміст. У випадку односторінкового сайту, HTML5 використовується для створення єдиного, але логічно розділеного документа, де кожен смисловий блок представляє окрему секцію сайту. Це дозволяє забезпечити плавну навігацію та послідовне представлення інформації. Використання семантичних тегів HTML5 покращує доступність, пошукову оптимізацію та полегшує подальше стилізування та маніпуляції за допомогою CSS та JavaScript.

2.1.1. Використання семантичних тегів HTML5 для секцій односторінкового сайту

Семантичні теги HTML5 дозволяють чітко визначити призначення різних частин веб-сторінки, роблячи код більш зрозумілим як для розробників, так і для пошукових систем та допоміжних технологій (наприклад, програм читання з екрана). У контексті односторінкового сайту, кожен "розділ" сайту (Головна, Про нас, Послуги тощо) оформлюється як окремий тег `<section>` з унікальним ідентифікатором (`id`) для навігації.

Основні семантичні теги, використані при розробці структури єдиної сторінки:

- `<header>`: Представляє вступний вміст або навігаційні посилання. Для односторінкового сайту, це, як правило, фіксована (sticky) шапка, що залишається видимою при прокручуванні та містить логотип клубу, основне меню навігації (яке веде до секцій на поточній сторінці) та, можливо, кнопки для входу/реєстрації або швидкого бронювання.
- `<main>`: Представляє основний вміст документа. У випадку односторінкового сайту, весь видимий контент, за винятком хедеру та футеру, розташовується всередині тегу `<main>`, який містить послідовність `<section>` елементів.
- `<section>`: Кожен логічний розділ сайту (наприклад, "Головна", "Про нас", "Послуги", "Поля", "Галерея", "Новини та Події", "Контакти", "Бронювання") оформлюється як окремий тег `<section>`. Кожна секція має унікальний атрибут `id`, який використовується для якорних посилань в навігаційному меню, забезпечуючи плавне прокручування до відповідного блоку.
- `<article>`: Використовується всередині `<section>` для самостійних, незалежних одиниць контенту, таких як окремі записи новин, елементи галереї, відгуки, або детальні описи послуг.

2.1.2. Детальна структура єдиної сторінки (index.html)

Єдиний HTML-файл (index.html або index.php, якщо контент генерується PHP) міститиме всі логічні секції сайту. Цей підхід забезпечує швидке завантаження та плавну навігацію.

2.2. Створення візуального оформлення за допомогою CSS

CSS (Cascading Style Sheets) є невід'ємною частиною веб-розробки, що відповідає за візуальне оформлення HTML-документа. Для односторінкового сайту гольф-клубу CSS відіграє ключову роль у створенні преміального, адаптивного та візуально привабливого інтерфейсу, який забезпечує єдиний стиль та зручність використання на різних пристроях.

2.2.1. Базові стилі та типографія

Основа будь-якого візуально приємного сайту – це чітко визначені базові стилі та гармонійна типографія.

Скидання стилів (CSS Reset/Normalize): Перед початком основної стилізації важливо "скинути" або "нормалізувати" стандартні стилі браузерів. Це забезпечує більш передбачуване відображення елементів на різних пристроях та у різних браузерах.

Кольорова палітра: Для гольф-клубу обираються кольори, що асоціюються з природою, елегантністю та престижем. На основі скріншоту, можна виділити такі основні кольори:

- Зелений (основний акцент): #4CAF50 (для кнопок, акцентів, елементів дизайну, що символізують поле для гольфу).
- Темно-сірий/Чорний: #333333 або #1a1a1a (для тексту, фонів хедеру/футеру, що надають преміальності).
- Світлий фон: #F8F8F8 або #FFFFFF (для фону більшості секцій).
- Золотий/Бежевий (додатковий акцент): #B8860B або #D4AF37 (для дрібних деталей, іконок, щоб підкреслити ексклюзивність). Ці кольори можна визначити як CSS-змінні для легкого управління:

Типографія: Вибір шрифтів та їх розмірів є важливим для читабельності та візуального сприйняття.

Шрифти: Використання Google Fonts (наприклад, 'Open Sans' для основного тексту та 'Playfair Display' або 'Merriweather' для заголовків) може надати сайту сучасного та елегантного вигляду.

Розміри та стилі заголовків: Чітка ієрархія заголовків (<h1> до <h6>) з відповідними розмірами та відступами.

2.2.2. Оформлення основних блоків та елементів

Стилізація окремих секцій та інтерактивних елементів є ключовою для створення завершеного вигляду сайту.

Контейнери та секції: Для забезпечення єдиного відступу та обмеження ширини контенту використовуються контейнери. Секції мають внутрішні відступи (`padding`) для візуального відділення.

Хедер та навігація: Фіксований хедер, адаптивне меню (гамбургер-іконка для мобільних пристроїв), анімація при прокручуванні.

2.3. Додавання інтерактивності за допомогою JavaScript

JavaScript є невід'ємною частиною сучасної веб-розробки, що дозволяє перетворити статичну веб-сторінку на динамічний та інтерактивний користувацький інтерфейс. Для односторінкового сайту гольф-клубу JavaScript виконує низку ключових функцій: від плавного прокручування до секцій та керування слайдерами, до обробки форм без перезавантаження сторінки, забезпечуючи високу зручність використання та швидкість взаємодії.

2.3.1. Реалізація плавного прокручування та навігації

Оскільки вебсайт є односторінковим, навігаційні посилання в хедері (та, можливо, в інших місцях) не ведуть на окремі HTML-файли, а плавно прокручують сторінку до відповідних секцій. Це покращує користувацький досвід, роблячи перехід між розділами сайту інтуїтивним та естетичним.

HTML-розмітка (нагадування): Кожна секція має унікальний `id`, а посилання в меню використовують ці `id` як якірні посилання (`href="#section-id"`).

JavaScript для плавного прокручування: Для реалізації плавного прокручування використовується метод `scrollIntoView()`. Він може бути викликаний на будь-якому елементі та дозволяє плавно прокрутити сторінку так, щоб цей елемент став видимим.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ БЕКЕНД ЧАСТИНИ

Бекенд частина вебсайту гольф-клубу, згідно з уточненими вимогами, зосереджується на двох ключових функціоналах: зборі та відображенні статистики відвідуваності сайту, а також обробці запитів на бронювання від користувачів. Реалізація цих завдань виконується за допомогою PHP для серверної логіки та MySQL для постійного зберігання даних. Цей розділ детально описує процес налаштування необхідного серверного середовища та розробки API для цих специфічних функцій.

3.1. Налаштування серверного середовища

Для забезпечення функціонування бекенду необхідно мати налаштоване серверне середовище, що включає веб-сервер, інтерпретатор PHP та сервер бази даних MySQL. Це базовий набір для виконання PHP-скриптів та взаємодії з даними.

3.1.1. Встановлення та конфігурація LAMP/WAMP/MAMP (XAMPP)

Вибір інтегрованого пакету, такого як XAMPP, є найбільш зручним для швидкого розгортання локального середовища розробки.

Завантаження та встановлення XAMPP:

- Завантажте останню версію XAMPP з офіційного сайту Apache Friends (apachefriends.org), обравши версію для вашої операційної системи (Windows, macOS, Linux).
- Запустіть інсталятор та дотримуйтеся інструкцій, бажано встановлюючи у кореневу папку диска (наприклад, C:\xampp на Windows) для спрощення доступу та уникнення проблем з дозволами.
- Запуск та управління службами: Після встановлення, відкрийте XAMPP Control Panel. У ній ви зможете запустити (кнопка "Start") та зупинити (кнопка "Stop") необхідні модулі:

Apache: Це веб-сервер, який буде обробляти запити від браузера та передавати їх PHP-скриптам.

MySQL: Це сервер бази даних, де будуть зберігатися дані про бронювання та лічильник відвідувань.

3.1.2. Базова конфігурація PHP та MySQL

Для ефективної роботи та забезпечення безпеки, бажано виконати мінімальні налаштування.

Конфігурація PHP (php.ini): Файл php.ini знаходиться у папці хатрр/php/.

display_errors = Off: На виробничому сервері, щоб уникнути витoku конфіденційної інформації про помилки. Для локальної розробки можна тимчасово встановити On для зручності відладки.

log_errors = On: Увімкнення запису помилок у лог-файл, що є корисним для відстеження проблем.

error_log = "путь_до_файлу_лога_помилки": Вкажіть повний шлях до файлу, наприклад, C:\хатрр\apache\logs\php_error.log.

Конфігурація MySQL (через phpMyAdmin): Доступ до управління базою даних MySQL здійснюється через веб-інтерфейс phpMyAdmin, який автоматично встановлюється з ХАМРР. Відкрийте його у браузері за адресою <http://localhost/phpmyadmin/>.

Користувачі та паролі: За замовчуванням, ХАМРР налаштовує користувача root без пароля. На виробничому сервері обов'язково встановіть надійний пароль для root і створіть окремого користувача з мінімальними привілеями для доступу до бази даних вашого проекту. Для локальної розробки це не так критично, але звичка до безпеки корисна.

3.2. Тестування та розгортання

Після розробки бекенд-функціоналу життєво важливо провести ретельне тестування, щоб переконатися в його коректній роботі, надійності та безпеці. Завершальним етапом є розгортання проекту на робочому сервері, де він стане доступним для кінцевих користувачів.

ВИСНОВКИ

У рамках даної дипломної роботи було успішно розроблено та реалізовано повноцінний односторінковий вебсайт для гольф-клубу, що відповідає сучасним вимогам до функціональності, зручності використання та естетичного оформлення. Поставлену мету роботи — створити онлайн-платформу, яка забезпечує інформаційну підтримку, можливість онлайн-бронювання послуг та збір базової статистики відвідуваності — було повністю досягнуто.

Основні результати та досягнення роботи включають:

Проектування та планування: Було розроблено концепцію односторінкового сайту, визначено ключові розділи (головна, про клуб, послуги, галерея, контакти), що забезпечує інтуїтивно зрозумілу навігацію. Створено макет інтерфейсу та визначено необхідний функціонал для фронтенду та бекенду.

Розробка фронтенд-частини:

- HTML5: Створено семантичну та логічну структуру єдиної HTML-сторінки, що забезпечує доступність та легкість індексації.
- CSS3: Реалізовано сучасний та привабливий дизайн сайту, включаючи адаптивну верстку, яка гарантує коректне відображення на різних пристроях (десктопи, планшети, мобільні телефони). Застосовано техніки стилізації, що відповідають фірмовому стилю гольф-клубу.
- JavaScript (jQuery): Додано динамічну інтерактивність: плавне прокручування до розділів, функціонал слайдера в герой-секції, динамічне мобільне меню ("гамбургер"), таймер зворотного відліку для акцій, анімацію тексту, а також взаємодію з формами без перезавантаження сторінки.

Розробка бекенд-частини:

- PHP та MySQL: Створено мінімалістичний, але ефективний бекенд для обробки ключових функцій.
- Лічильник відвідувань: Реалізовано механізм запису відвідувань до бази даних та отримання загальної кількості відвідувачів, що дозволяє клубу відстежувати популярність сайту.
- Обробка форми бронювання: Впроваджено надійний механізм прийому даних з форми бронювання (ім'я, email, дата, час, тип бронювання, кількість гравців) з валідацією на стороні сервера та збереженням у базу даних MySQL. Це дозволяє клубу ефективно керувати запитами від клієнтів.

- Інтеграція та тестування: Всі компоненти фронкенду та беккенду були успішно інтегровані. Проведено ретельне ручне тестування функціоналу, включаючи відправку форм, роботу навігації, адаптивність дизайну та коректність роботи лічильника відвідувань. Тестування підтвердило стабільність та коректність роботи розробленої системи.

Практична цінність роботи полягає у створенні готового до впровадження інструменту, який значно спрощує взаємодію потенційних та існуючих клієнтів з гольф-клубом, покращує його онлайн-присутність та автоматизує процеси бронювання. Сайт є гнучким і може бути легко розширений в майбутньому новими функціями, такими як повноцінна система управління користувачами, інтеграція платіжних систем, календар подій або динамічне управління контентом через адміністративну панель.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гончаренко С.Ю. **Веб-програмування з використанням HTML, CSS та JavaScript**. Київ: Видавнича група BHV. 2021. 480 с.
2. Сміт А.Дж. **Основи адаптивного веб-дизайну**. Львів: Астролябія. 2019. 320 с.
3. Коваленко В.О., Мельник І.Р. **Розробка веб-застосунків на PHP та MySQL**. Харків: Фоліо. 2022. 512 с.
4. Петренко О.С. **JavaScript: Сучасні підходи до розробки інтерактивних веб-сайтів**. Дніпро: Герда. 2020. 416 с.
5. Андрущенко О.В. **Системи керування базами даних MySQL для веб-розробників**. Одеса: Маяк. 2018. 280 с.
6. Мартиненко П.Г. **Односторінкові додатки (SPA): архітектура та реалізація**. Житомир: Полісся. 2023. 350 с.
7. Стандарти W3C для веб-розробки. Офіційна документація. URL: <https://www.w3.org/standards/> (дата звернення: 05.06.2025).
8. PHP: Офіційна документація. PHP Group. URL: <https://www.php.net/manual/ru/> (дата звернення: 05.06.2025).
9. MySQL: Офіційна документація. Oracle Corporation. URL: <https://dev.mysql.com/doc/> (дата звернення: 05.06.2025).
10. Документація по AJAX (Asynchronous JavaScript and XML). MDN Web Docs. URL: <https://developer.mozilla.org/uk/docs/Web/Guide/AJAX> (дата звернення: 05.06.2025).
11. Принципи UI/UX дизайну для веб-сайтів. Професійний портал для дизайнерів. URL: <https://ux.in.ua/articles> (дата звернення: 05.06.2025).
12. Кравченко О.В. **Веб-безпека: Захист даних та користувачів**. Київ: Видавничий дім "Книги та Софт". 2021. 290 с.
13. Прокопенко Л.Д. **SEO-оптимізація та просування веб-сайтів**. Полтава: Мікрос. 2020. 270 с.

ДОДАТКИ

JS код сайту

```
// Код виконується після повного завантаження сторінки
$(window).on('load', function() {
    // Приховуємо елемент із класом 'loader' (зазвичай це анімація попереднього
завантаження)
    // з плавним зникненням протягом 300 мілісекунд
    $('.loader').fadeOut(300);
});
// Код виконується під час прокручування вікна браузера
$(window).on('scroll', function() {
    // Перевіряємо, якщо позиція прокручування зверху (scrollTop) більша за 0
(тобто, сторінка прокручена хоч трохи)
    if ($(this).scrollTop() > 0) {
        // Додаємо клас 'active' до елемента з класом 'header' (шапка сайту),
        // що може змінювати її стиль (наприклад, додавати фон або тінь)
        $('.header').addClass('active');
    } else {
        // В іншому випадку (якщо сторінка знаходиться на самому верху)
        // Видаляємо клас 'active' з шапки
        $('.header').removeClass('active');
    }
});
// Обробник кліку на кнопку закриття модального вікна
$('.h-modal-content-close').click(function(){
    // Знаходимо найближчий батьківський елемент з класом '.h-modal' (саме
модальне вікно)
    // і плавно приховуємо його протягом 200 мілісекунд
    $(this).closest(".h-modal").fadeOut(200);
});
```

```

// Обробник кліку на самому модальному вікні (оверлеї)
$(".h-modal").click(function(event) {
    // Перевіряємо, чи клік не був зроблений на вмісті модального вікна (.h-
modal-content) або всередині нього
    if (!$(event.target).closest(".h-modal-content").length) {
        // Якщо клік був зроблений за межами вмісту модального вікна,
        // то приховуємо модальне вікно протягом 200 мілісекунд
        $(".h-modal").fadeOut(200);
    }
});
// Отримуємо посилання на кнопку "Вгору"
var upBtn = $('.up-btn');
// Приховуємо кнопку "Вгору" за замовчуванням
upBtn.hide();
// Обробник прокручування вікна для кнопки "Вгору"
$(window).scroll(function() {
    // Якщо сторінка прокручена більш ніж на 300 пікселів від верху
    if ($(this).scrollTop() > 300) {
        // Плавню показуємо кнопку "Вгору"
        upBtn.fadeIn();
    } else {
        // В іншому випадку, плавню ховаємо кнопку "Вгору"
        upBtn.fadeOut();
    }
});
// Обробник кліку на кнопку "Вгору"
upBtn.click(function() {
    // Плавню прокручуємо сторінку до самого верху (позиція 0) протягом 600
мілісекунд
    $('html, body').animate({ scrollTop: 0 }, 600);
});

```

```

// Забороняємо стандартну поведінку посилання (щоб уникнути переходу
або інших дій)
return false;
});
// Обробник кліку на елементі кнопки вкладки
$('.tab-btn-item').click(function() {
// Видаляємо клас 'active' з усіх кнопок вкладок
$('.tab-btn-item').removeClass('active');
// Додаємо клас 'active' до поточної (натиснутої) кнопки вкладки
$(this).addClass('active');
// Отримуємо індекс поточної кнопки вкладки
var tabId = $(this).index();
// Приховуємо всі елементи вмісту вкладок без анімації
$('.tab-content-item').fadeOut(0);
// Плавню показуємо (перемикаємо видимість) елемент вмісту вкладки,
// який відповідає індексу натиснутої кнопки, протягом 200 мілісекунд
$('.tab-content-item').eq(tabId).fadeToggle(200);
});
// Обробник кліку на кнопках прокручування до секції
$(".scroll-btn").click(function() {
// Отримуємо ID цільової секції з атрибуту 'data-scroll' кнопки
var scrollId = $(this).attr('data-scroll');
// Плавню прокручуємо сторінку до початку цільової секції
$('html, body').animate({
scrollTop: $(scrollId).offset().top
}, 500); // Анімація прокручування триває 500 мілісекунд
// Видаляємо клас 'menu-open' з хедера (для закриття мобільного меню)
$('.header').removeClass('menu-open');
// Плавню згортаємо мобільне меню
$('.mob-menu').slideUp(100);

```

```

// Видаляємо клас 'active' з іконки мобільного меню ("гамбургера")
$('.header-bars').removeClass('active');
});
// Обробник кліку на тіньовому оверлеї мобільного меню
$('.mob-menu-shadow').click(function() {
    // Імітуємо клік на кнопці "гамбургера", що закриє мобільне меню
    $('.header-bars').click();
});
// Обробник кліку на іконці мобільного меню ("гамбургер")
$('.header-bars').click(function() {
    // Перемикаємо клас 'active' для іконки гамбургера (змінює його стан/вигляд)
    $(this).toggleClass('active');
    // Перемикаємо клас 'menu-open' для хедера (може зміщувати контент або
відкривати меню)
    $('.header').toggleClass('menu-open');
    // Перемикаємо видимість (плавно розгортаємо/згортаємо) мобільного меню
    $('.mob-menu').slideToggle(300);
});
// Обробник кліку для всіх посилань, що мають href="#"
$(document).on('click', 'a[href="#"', function(e) {
    // Забороняє стандартну поведінку посилання (наприклад, прокручування до
верху сторінки)
    e.preventDefault();
});
// default end
// Ініціалізація бібліотеки Typed.js для ефекту друку тексту
var typed = new Typed('.head-slogan span', {
    // Масив рядків, які будуть друкуватися
    strings: [
        'Відчуйте гольф світового рівня на тлі захопливої природи.',

```

'Ідеальні поля, професійне навчання та ексклюзивні зручності.',
 'Приєднуйтеся до спільноти любителів гольфу та вдосконалюйте свою
 гру.'

```

  ],
  // Швидкість друку символів (мілісекунд на символ)
  typeSpeed: 50,
  // Вказує, чи потрібно повторювати анімацію нескінченно
  loop: true,
});
// Код виконується після повного завантаження DOM-дерева
$(document).ready(function() {
  // Функція для оновлення таймера зворотного відліку
  function updateTimer() {
    // Отримуємо кінцеву дату з атрибута 'data-end' елемента таймера
    let endDateVal = $('<div class="head-block-timer">').attr('data-end');
    let endDate = new Date(endDateVal).getTime(); // Перетворюємо в
    мілісекунди
    let now = new Date().getTime(); // Поточний час в мілісекундах
    let timeLeft = endDate - now; // Залишок часу до кінцевої дати

    // Якщо залишок часу більший за 0 (тобто, дата ще не настала)
    if (timeLeft > 0) {
      // Розраховуємо дні, години, хвилини та секунди
      let days = Math.floor(timeLeft / (1000 * 60 * 60 * 24));
      let hours = Math.floor((timeLeft % (1000 * 60 * 60 * 24)) / (1000 * 60 *
    60));
      let minutes = Math.floor((timeLeft % (1000 * 60 * 60)) / (1000 * 60));
      let seconds = Math.floor((timeLeft % (1000 * 60)) / 1000);

      // Оновлюємо значення на сторінці
      // Знаходимо елементи з 'sub' текстом 'days', 'hours' і т.д.,

```

// а потім оновлюємо текст їхнього попереднього сестринського елемента (який містить число)

```

    $(".head-block-timer-item sub:contains('days').prev().text(days);
    $(".head-block-timer-item sub:contains('hours').prev().text(hours);
    $(".head-block-timer-item sub:contains('minutes').prev().text(minutes);
    $(".head-block-timer-item sub:contains('seconds').prev().text(seconds);
  } else {
    // Якщо час вийшов, зупиняємо інтервал таймера
    clearInterval(timerInterval);
    // Встановлюємо всі значення таймера на 0
    $(".head-block-timer-item-val").text("0");
  }
}

// Запускаємо оновлення таймера кожну секунду (1000 мілісекунд)
let timerInterval = setInterval(updateTimer, 1000);

// Викликаємо функцію оновлення таймера один раз одразу після
завантаження,
// щоб уникнути затримки перед першим відображенням
updateTimer();
});

// Обробник кліку на кнопці, яка розгортає форму у хедері
$('.head-block-btn').click(function(){
  // Плавно приховуємо кнопку, яка була натиснута
  $(this).fadeOut(0);
  // Плавно розгортаємо (показуємо) форму
  $('.head-block-form').slideDown(200);
});

// Обробник відправки форми у хедері
$('.head-block-form form').submit(function(){

```

```
// Забороняємо стандартну поведінку відправки форми (перезавантаження сторінки)
event.preventDefault();

// Приховуємо найближчий батьківський елемент '.head-block-form' (саму форму)
$(this).closest('.head-block-form').fadeOut(0);

// Плавно розгортаємо (показуємо) повідомлення про успішне прийняття форми
$('.head-block-form-аccept').slideDown(200);
});

// Обробник кліку на елементі кастомного радіо-перемикача
$('.default-form-radio-item').click(function() {
    // Знаходимо найближчий батьківський елемент з класом '.default-form-radio' (групу радіо-перемикачів)
    // і видаляємо клас 'active' з усіх його дочірніх елементів '.default-form-radio-item'
    $(this).closest('.default-form-radio').find('.default-form-radio-item').removeClass('active');

    // Додаємо клас 'active' до поточного (натиснутого) радіо-перемикача
    $(this).addClass('active');
});

// Обробник відправки форми всередині модального вікна
$('.h-modal form').on('submit', function(event) {
    // Забороняємо стандартну поведінку відправки форми
    event.preventDefault();

    // Знаходимо найближчий батьківський елемент '.h-modal' (саме модальне вікно)
    // і додаємо клас 'active' до елемента '.h-modal-аccept' всередині нього (повідомлення про успіх)
    $(this).closest('.h-modal').find('.h-modal-аccept').addClass('active');
```

```
});  
// Обробник кліку на кнопці відкриття модального вікна  
$('.modal-btn').click(function() {  
    // Отримуємо ID цільового вмісту модального вікна з атрибуту 'data-modal'  
кнопки  
    var modalId = $(this).attr('data-modal');  
    // Приховуємо всі елементи вмісту модальних вікон без анімації  
    $('.h-modal-content-item').fadeOut(0);  
    // Показуємо (розкриваємо) конкретний вміст модального вікна, який  
відповідає modalId  
    $(modalId).fadeIn(0);  
    // Плавно показуємо саме модальне вікно (оверлей)  
    $('.h-modal').fadeIn(300);  
    // Видаляємо клас 'active' з повідомлення про успіх (якщо воно було  
активним від попереднього використання)  
    $('.h-modal-accept').removeClass('active');  
});
```