

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально–науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: «**РОЗРОБКА WEB-ДОДАТКУ ДЛЯ БАЗИ ВІДПОЧИНКУ МОВОЮ**

JAVA SCRIPT»

Виконала: студентка 4 курсу, групи ПД– 42
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Музика Мілена Андріївна

(прізвище та ініціали)

Керівник Бондарчук А.П.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерія програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

О. В. Негоденко

« » 2023 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Музиці Мілені Андріївні

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка web-додатку для бази відпочинку мовою Java Script»

Керівник роботи: Бондарчук А. П. професор

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

затверджені наказом вищого навчального закладу від «24» лютого 2023 року №
26

2. Строк подання студентом роботи «3» червня 2023 року

3. Вихідні дані до роботи:

3.1 Основні положення побудови веб-додатку;

3.2 Методи побудови веб-додатку;

3.3 Існуючі веб-додатки для бази відпочинку;

3.4 Науково-технічна література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- 4.1 Аналіз предметної області
- 4.2 Огляд існуючих аналогів веб-додатків
- 4.3 Програмна реалізація додатку
- 4.4 Приклади використання та тестування системи
- 5. Перелік графічного матеріалу
 - 5.1. Аналоги
 - 5.2. Засоби реалізації
 - 5.3. Опис програми
 - 5.4. Архітектура веб-додатку
 - 5.5. карта веб-сайту
 - 5.6. Екрана форма
- 2. Графічне представлення веб-додатку
- 6. Дата видачі завдання: 19.03.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи роботи	Примітка
1	Підбір науково-технічної літератури	01.04-07.04	Виконано
2	Ознайомлення зі сферою відпочинку	02.04-04.04	Виконано
3	Написання першого розділу	11.04-14.04	Виконано
4	Написання другого розділу	14.04-18.04	Виконано
5	Розробка веб-додатку	14.04-18.04	Виконано
6	Написання третього розділу	18.04-19.04	Виконано
7	Висновки, оформлення роботи	20.04-21.04	Виконано
8	Розробка демонстраційних матеріалів	21.04-30.04	Виконано
9	Попередній захист роботи	01.06.2023	
10	Здача роботи	06.06.2023	

Студентка

Музика М. А.

(підпис)

(прізвище та ініціали)

Керівник роботи

Бондарчук А. П.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 40с., 40 рис., 20 джерел.

Об'єкт дослідження – процес покращення взаємодії з клієнтом за допомогою веб додатку.

Предмет дослідження – веб технології для розробки додатку.

Мета роботи – спрощення взаємної з клієнтами бази відпочинку за допомогою веб додатку.

Методи дослідження – аналіз предметної області, аналіз існуючих рішень, методи проектування та розробки web-додатків, методи тестування web-додатків.

В науковій роботі було проведено аналіз наукових та літературних джерел з заданої тематики дослідження, проаналізовано сучасні середовища та методи розробки веб-додатків. Описано архітектуру, основні принципи та етапи розробки додатку, а також визначено його основні функціональні можливості.

В результаті роботи спроектовано та розроблено web-додаток для бронювання котеджів мовою програмування JavaScript з використанням CSS, HTML.

Даний додаток може бути використаний будь-якою базою відпочинку. Дане рішення значно зменшить витрату часу та коштів компанії при обробці бронюванні місць. Люди зможуть використовувати веб-додаток з οποї точки світу на будь-якій операційній системі пристрою з мінімальними вимогами, легко орієнтуючись на веб-сторінці. Користувач зможе забронювати котедж без реєстрації облікового запису, вказавши основні дані - прізвище, ім'я, дату заселення та виселення, номер телефону, не витрачаючи на це купу часу.

Галузь використання – сфера туризму та відпочинку.

ВЕБ ДОДАТОК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, БАЗА ВІДПОЧИНКУ, БРОНЮВАННЯ КОТЕДЖУ, JAVASCRIPT, HTML, CSS, VISUAL STUDIO CODE.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Основні поняття	11
1.2 Основні принципи веб-розробки	12
1.3 Огляд існуючих рішень	17
1.3 Постановка задачі.....	25
2 ДОСЛІДЖЕННЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ	27
2.1 HTML	27
2.2 CSS	29
2.3 JavaScript	31
2.4 Visual Studio Code	33
2.5 Express.JS.....	35
2.6 NODEMAILER	36
3 РОЗРОБКА ВЕБ-ДОДАТКУ.....	38
3.1 Html частина	38
3.2 JavaScript	41
3.3 CSS	44
3.3. Графічна частина.....	48
ВИСНОВКИ	50
ПЕРЕЛІК ПОСИЛАНЬ.....	51
ДОДАТОК А.....	53
ДОДАТОК Б.....	60

ВСТУП

На сьогоднішній день відпочинок – це незамінна складова нашого фізичного, емоційного та психічного благополуччя. У світі, де ритм життя стрімко набирає обертів, а вимоги та виклики постійно зростають, відпочинок стає більш життєво необхідною потребою, ніж коли-небудь раніше. Адже нестача відпочинку може призвести до негативних наслідків для нашого фізичного, емоційного та психічного стану. А саме - зниження працездатності, імунітету, збільшення вразливості до захворювань, стресу, зниження настрою та появи емоційних розладів, таких як депресія або тривожність. Тому слід приділяти час та увагу вашому відпочинку.

Одним з популярних варіантів відпочинку є проведення часу на базах відпочинку - спеціалізованих туристичних комплексах, де люди можуть насолоджуватися різноманітними видами відпочинку та розвагами. Проте не всі люди можуть знати про нові бази та бронювати місця через відсутність інформації про них.

Слід зазначити, на сьогоднішній день веб-сайти стали невід'ємною частиною багатьох бізнес-процесів, включаючи галузь відпочинку та туризму. Відповідно розробка веб-додатку допоможе забезпечити ефективну роботу бази відпочинку.

Існуючі веб-додатки мають такі недоліки як: відсутність адаптивності для різних розмірів екранів та пристроїв, немає сумісності з різними пристроями та браузерами, низька швидкість завантаження сторінок, недостатня оптимізація, веб-сайти мають лише одну мову, інтерфейс не завжди простий та інтуїтивно зрозумілий тощо. Тому розробка веб-додатку буде досить актуальною, бо від цього теж залежить успіх бази відпочинку.

Актуальність роботи:

- Зростання популярності веб-додатків та онлайн-бронювання серед туристів.
- Потреба у покращенні ефективності та організації баз відпочинку.
- Важливість забезпечення доступу до системи онлайн-бронювання та відгуків клієнтів.
- Розвиток технологій та зміни у способах відпочинку та планування

подорожей.

– Потенційна можливість покращення конкурентоспроможності та збільшення прибутку підприємства через розробку веб-додатку для баз відпочинку.

Об'єктом дослідження є процес покращення взаємодії з клієнтом за допомогою веб додатку.

Предмет дослідження – веб технології для розробки додатку.

Мета даної роботи – спрощення взаємної з клієнтами бази відпочинку за допомогою веб додатку.

Для досягнення зазначеної мети було поставлено такі завдання:

1. Проведення аналізу наукових та літературних джерел з тематики дослідження;
2. Розгляд сучасних веб-додатків, що використовуються в галузі відпочинку;
3. Проектування етапів розробки додатку;
4. Проведення аналізу сучасних інструментів та програмних засобів реалізації веб-додатку;
5. Написання програмного коду для розв'язування поставленого завдання;
6. Описання архітектури та функціоналу додатку;
7. Проведення функціонального тестування розробленої системи;

Методи дослідження – аналіз предметної області, аналіз існуючих рішень, методи проектування та розробки web-додатків, методи тестування web-додатків.

Наукова новизна полягає у тому, що розроблений веб-додаток, який може бути використаний будь-якою базою відпочинку. Дане рішення значно зменшить витрату часу та коштів компанії при обробці бронюванні місць. Люди зможуть використовувати веб-додаток з будь-якої точки світу на будь-якій операційній системі пристрою з мінімальними вимогами, легко орієнтуючись на веб-сторінці. Користувач зможе забронювати котедж без реєстрації облікового запису, вказавши основні дані - прізвище, ім'я, дату заселення та виселення, номер телефону, не витрачаючи на це купу часу.

Практична значущість результатів дослідження може полягати в можливості рішення практичної задачі, проведення подальших наукових досліджень або для використання в процесі підготовки фахівців.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття

Веб-додаток - це програмне забезпечення, яке доступне для використання через веб-браузер. Він зазвичай розташовується на веб-сервері та надає користувачам можливість взаємодії з ним за допомогою веб-інтерфейсу, що відображається на веб-сторінках.

Веб-додатки можуть виконувати різні функції, такі як обробка даних, зберігання інформації, робота з базами даних, розрахунки, візуалізація даних, взаємодія з іншими користувачами тощо. Вони можуть бути розроблені на різних мовах програмування, таких як JavaScript, PHP, Python, Ruby, Java, C# та інших.

Одні з прикладів веб-додатків включають соціальні мережі, електронну пошту, онлайн-банківські системи, електронні магазини, системи управління вмістом, онлайн-ігри, веб-сервіси, інструменти для співпраці в реальному часі та багато інших. Веб-додатки можуть бути використані на різних платформах, таких як комп'ютери, смартфони, планшети тощо, і вони можуть бути доступні в режимі онлайн або офлайн залежно від їхньої функціональності та налаштувань [1].

Веб-система - це комплексне програмне забезпечення, яке використовується для побудови та функціонування веб-додатків або веб-сайтів. Вона складається з ряду взаємодіючих компонентів, таких як веб-сервери, бази даних, веб-клієнти, додатки та інші елементи, які співпрацюють між собою для надання веб-послуг.

Веб-системи забезпечують можливість взаємодії користувачів з веб-додатками через веб-браузери або інші веб-клієнти. Вони можуть бути різноманітні за своєю природою, розміром та функціональністю, від простих веб-сайтів до складних веб-додатків, таких як електронна комерція, соціальні мережі, банківські системи, системи управління контентом та інші [2].

Вебсайт - це сукупність веб-сторінок, пов'язаних між собою гіперпосиланнями, які розміщені на веб-сервері та доступні для перегляду в Інтернеті. Вебсайт може містити різноманітний контент, такий як текст, зображення, відео, аудіо, графіку, форми вводу, функції взаємодії та інші елементи.

Вебсайти можуть мати різні цілі та призначення, такі як особисті, корпоративні, комерційні, навчальні, розважальні та багато інших. Вони можуть бути розроблені для різних аудиторій та використовуватися для різних цілей, таких як представлення продуктів або послуг, надання інформації, комунікації з користувачами, розповсюдження контенту, проведення електронної комерції, спілкування, навчання, розваги тощо. Вебсайти можуть бути розроблені за допомогою різних технологій та платформ, таких як HTML, CSS, JavaScript, PHP, Python, WordPress, Joomla, Drupal, Shopify, Magento та багато інших [3].

Вебсторінка - це електронний документ, який доступний для перегляду в Інтернеті. Вона є основною одиницею веб-контенту і може містити текст, зображення, відео, аудіо, графіку, гіперпосилання та інші елементи. Вебсторінки розміщуються на веб-серверах і відображаються веб-браузерами на різних пристроях, таких як комп'ютери, смартфони, планшети тощо.

Вебсторінки використовуються для відображення різноманітного контенту в Інтернеті, такого як веб-сайти, блоги, новини, електронні магазини, соціальні мережі, форуми, веб-додатки та інші різновиди веб-застосунків [4].

Слід відмітити, що різниця між веб-додатком та веб-системою може бути неоднозначна, оскільки вони можуть мати різні рівні складності та функціональність в залежності від конкретного контексту використання. Деякі веб-додатки можуть мати велику кількість функцій та бути більш складними за деякі веб-системи, в той час як інші веб-системи можуть бути дуже простими та містити лише обмежений набір функцій. Веб-система - комплексний набір взаємопов'язаних веб-сторінок та функціональності, а веб-додаток - це програмне забезпечення, що виконується веб-браузером і забезпечує користувачам можливість взаємодії з веб-сервером і виконання складніших функцій.

1.2 Основні принципи веб-розробки

Веб-розробка - це процес створення та вдосконалення веб-додатків або веб-сайтів. Вона складається в основному з двох частин, включає в себе розробку

фронтенду (клієнтської сторони) та бекенду (серверної сторони) веб-додатка, нижче на рис.1.1 представлена схема:

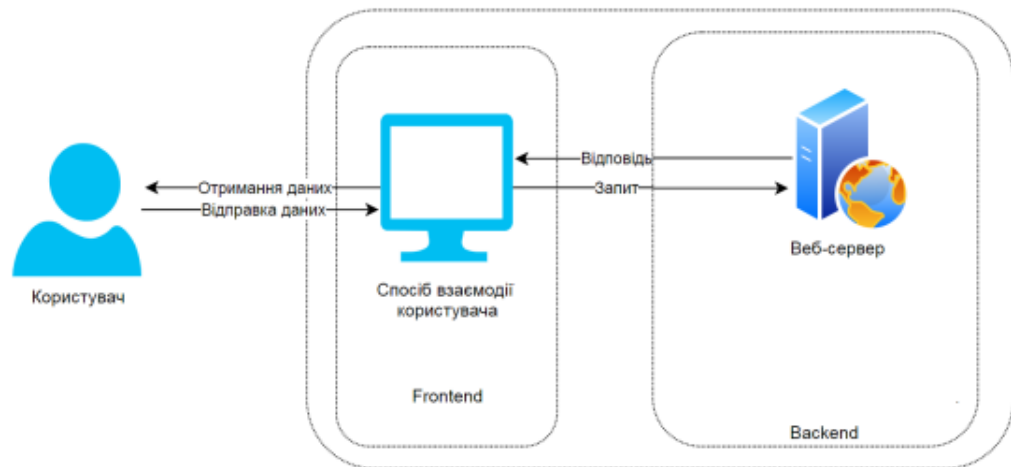


Рисунок 1.1 – Архітектура веб-додатку

Принцип роботи схеми архітектури «клієнт-сервер» такий: користувач (клієнт) відправляє дані через фронтенд (це візуальне представлення даних, наприклад інтерфейс веб-сайту), потім йде запит від фронтенду до бекенду, після цього бекенд відправляє відповідь на запит через фронтенд до клієнту.

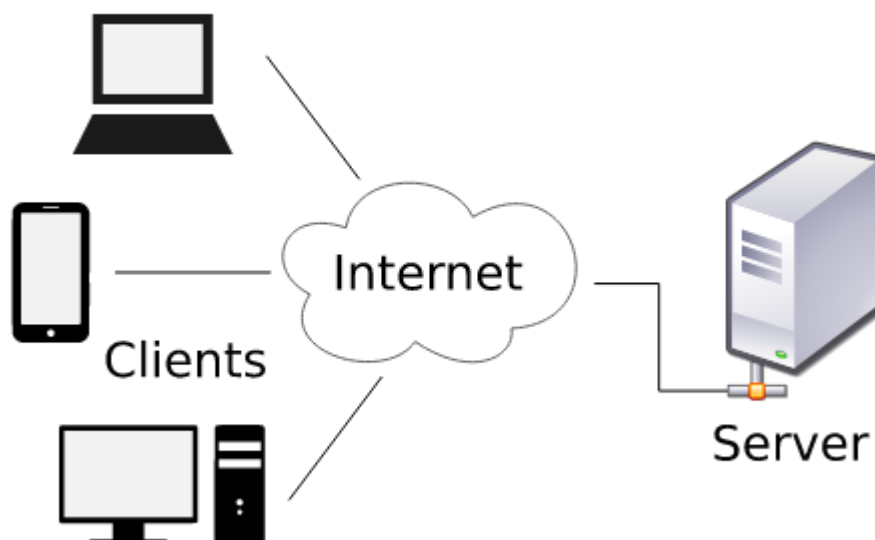


Рисунок 1.2 – Схема архітектури «клієнт-сервер»

Основні етапи веб-розробки включають аналіз вимог, проектування, розробку,

тестування та розгортання.

Архітектура веб-додатку - це план або схема, яка описує організацію компонентів та взаємодії між ними в рамках веб-додатку. Основні компоненти архітектури веб-додатку можуть включати:

1) Клієнтська сторона (Front-end). Це інтерфейс, з яким користувач взаємодіє, зазвичай в браузері. Включає веб-сторінки, веб-форми, стилі, скрипти та інші елементи, що відображаються на екрані користувача.

2) Серверна сторона (Back-end). Це сервер, який обробляє запити від клієнтської сторони. Включає серверні програми, бази даних, сервіси та інші компоненти, що виконують бізнес-логіку та забезпечують взаємодію з базами даних, зовнішніми сервісами та іншими ресурсами.

3) Комунікація: це механізми, які дозволяють взаємодіяти між різними компонентами системи, такими як HTTP-протокол для передачі запитів та відповідей між клієнтською та серверною сторонами, REST або SOAP для взаємодії з веб-сервісами, та інші механізми.

4) Забезпечення безпеки: це механізми, які забезпечують захист веб-додатку від зловживання, злому та витоку даних. Може включати автентифікацію, авторизацію, захист від крос-сайтового скриптування (XSS), крос-сайтового поділу ресурсів.

Веб-розробка дозволяє створювати різноманітні функціональні веб-додатки, такі як електронні магазини, соціальні мережі, бронювання, фінансові системи та багато інших. Тому вона є важливою галуззю інформаційних технологій, яка швидко розвивається та має великий вплив на сучасний світ Інтернету [5].

Основні принципи веб-розробки включають в себе наступне:

1) Клієнтоорієнтованість: фокус на потребах та вимогах користувачів, забезпечення зручного та ефективного взаємодії з веб-додатком.

2) Кросплатформеність: розробка веб-додатків, які можуть працювати на різних веб-платформах, браузерах та пристроях.

3) Відповідність веб-стандартам: дотримання рекомендацій та стандартів,

таких як HTML, CSS, JavaScript, для забезпечення сумісності, доступності та розширюваності веб-додатків.

4) Безпека: захист веб-додатків від потенційних загроз, таких як кросс-сайтові скрипти, SQL-ін'єкції, злами, забезпечення конфіденційності та цілісності даних.

5) Ефективність: оптимізація продуктивності веб-додатків, забезпечення швидкої відповіді на запити користувачів та ефективного використання ресурсів сервера.

6) Масштабованість: забезпечення можливості розширення веб-додатків, враховуючи зростання обсягу даних та навантаження на сервер.

7) Тестування: використання різних методів тестування, таких як модульні, інтеграційні та функціональні тести, для перевірки якості та надійності веб-додатків.

8) Резервне копіювання та відновлення: забезпечення регулярного резервного копіювання та можливості відновлення веб-додатків в разі виникнення помилок чи втрати даних.

9) Навігація – це система орієнтування, яка використовується на веб-сайтах, де відвідувачі взаємодіють і знаходять те, що шукають. Навігація веб-сайтом є ключем до утримання відвідувачів.

Ці принципи допомагають забезпечити високу якість та ефективність веб-додатків, забезпечують функціональність веб-додатків на різних платформах та браузерах, забезпечення безпеки та захисту даних, ефективне використання ресурсів сервера, можливість розширення та масштабування веб-додатків з ростом обсягу даних, виконання регулярних тестів для впевненості в якості та надійності додатків, а також резервне копіювання та відновлення для забезпечення надійності роботи веб-додатків [6].

Врахування цих принципів допомагає розробникам створювати веб-додатки, які відповідають вимогам сучасного Інтернету та задовольняють потреби користувачів.

Основні принципи розробки інтерфейсу веб-додатку включають:

- Візуальна привабливість: Веб-додаток повинен бути привабливим та залучати користувача візуально. Візуальний дизайн повинен бути зручним, зрозумілим та привабливим для користувачів.

- Інтуїтивність та простота використання: Інтерфейс веб-додатку має бути логічним, зрозумілим та легким для використання. Користувачі повинні швидко зорієнтуватись в інтерфейсі та з легкістю виконувати необхідні дії.

- Консистентність: інтерфейс веб-додатку повинен бути узгодженим в усіх його елементах, включаючи розташування елементів керування, стиль та оформлення. Це допомагає користувачам легко розуміти та використовувати різні функції додатку.

- Адаптивність: інтерфейс веб-додатку повинен бути адаптивним та резиновим, тобто він повинен коректно відображатись на різних розмірах екранів, включаючи мобільні пристрої, планшети та настільні комп'ютери.

- Ефективність та продуктивність: інтерфейс веб-додатку повинен бути ефективним та продуктивним, допомагати користувачам виконувати їхні завдання швидко та ефективно, з мінімальною кількістю кроків.

- Стан помилок та відновлення: інтерфейс веб-додатку повинен передбачати помилкові ситуації та надавати користувачам зрозумілі повідомлення про помилки та шляхи відновлення.

- Доступність: інтерфейс веб-додатку повинен бути доступним для різних категорій користувачів, включаючи осіб з обмеженими можливостями. Доступність включає в собі такі аспекти, як зручність для користувачів з вадами зору, слуху або руховими обмеженнями.

- Спрощення взаємодії: інтерфейс веб-додатку повинен спрощувати взаємодію користувачів з додатком, зменшувати кількість необхідних дій та спрощувати процеси роботи з додатком.

- Системні повідомлення: веб-додаток повинен надавати зрозумілі та зрозумілі системні повідомлення, такі як підтвердження виконання дій, попередження про можливі наслідки видалення або зміни даних, а також інші повідомлення, що стосуються стану системи.

- Тестування та відладка: при розробці інтерфейсу веб-додатку важливо виконувати тестування та відлагодження, щоб переконатися в його працездатності, відсутності помилок та налагодити взаємодію з користувачами [7].

Ці принципи веб-розробки інтерфейсу веб-додатку допомагають забезпечити зручну, привабливу, ефективну та доступну взаємодію користувачів з додатком, забезпечуючи позитивний досвід використання та задоволення від взаємодії з веб-додатком.

1.3 Огляд існуючих рішень

Перед початком розробки веб-додатку важливо провести аналіз існуючих аналогів, оцінити їх переваги та недоліки, щоб уникнути повторення помилок та використати переваги вже наявних систем.

Було обрано 3 веб-ресурси для дослідження.

Спочатку розглянемо веб-сайт бази відпочинку «П'ятниця» (режим доступу <https://www.bazaryatnitsa.kiev.ua/>) [8].

На рисунок 1.3 представлено вигляд головної сторінки бази відпочинку «П'ятниця».

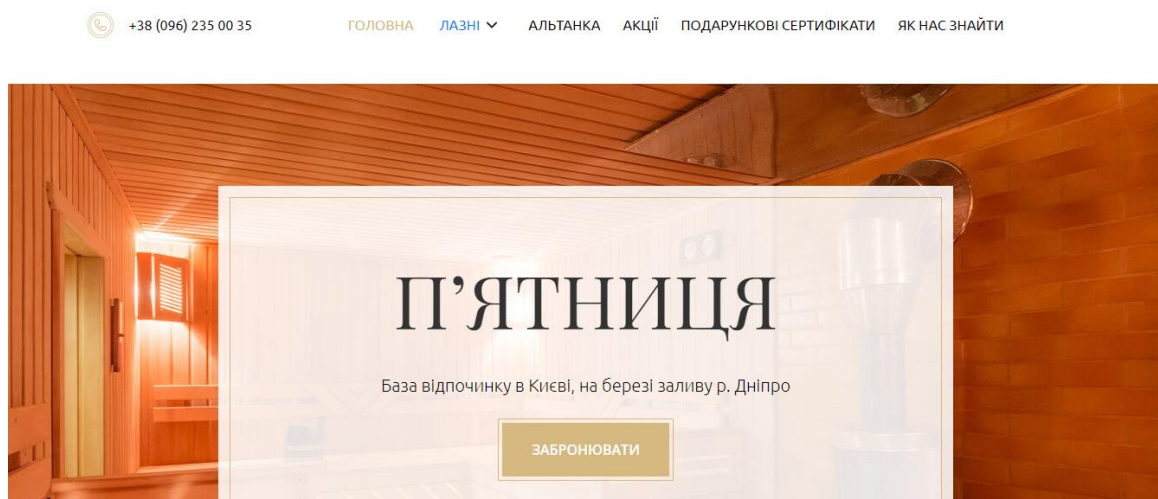


Рисунок 1.3 – Головна сторінка бази відпочинку «П'ятниця»

Дана сторінка має досить зручний інтерфейс, але трохи не звичний дизайн. Сайт має адаптивність під різні розміри екрану, швидкість завантаження сторінок

гарна.

На даному ресурсі відображається одразу кнопка бронювання, навігація по системі представлена у вигляді горизонтального меню. При натисканні на вкладки відкривається докладна інформація, що прикріплена до них. Щоб забронювати номер потрібно натиснути кнопку бронювання, і заповнити викликану форму.

Один з недоліків веб-системи – наявність тільки однієї мови, і те, що при переході на інші вкладки з меню – не відображається ніде назва бази (рис. 1.4). Також відсутня вкладка «Контакти», це зменшить ефективність роботи сайту – взаємодію з клієнтами. Відсутні коментарі та можливість їх залишити на сайті – це знизить довіру клієнта до даного комплексу.

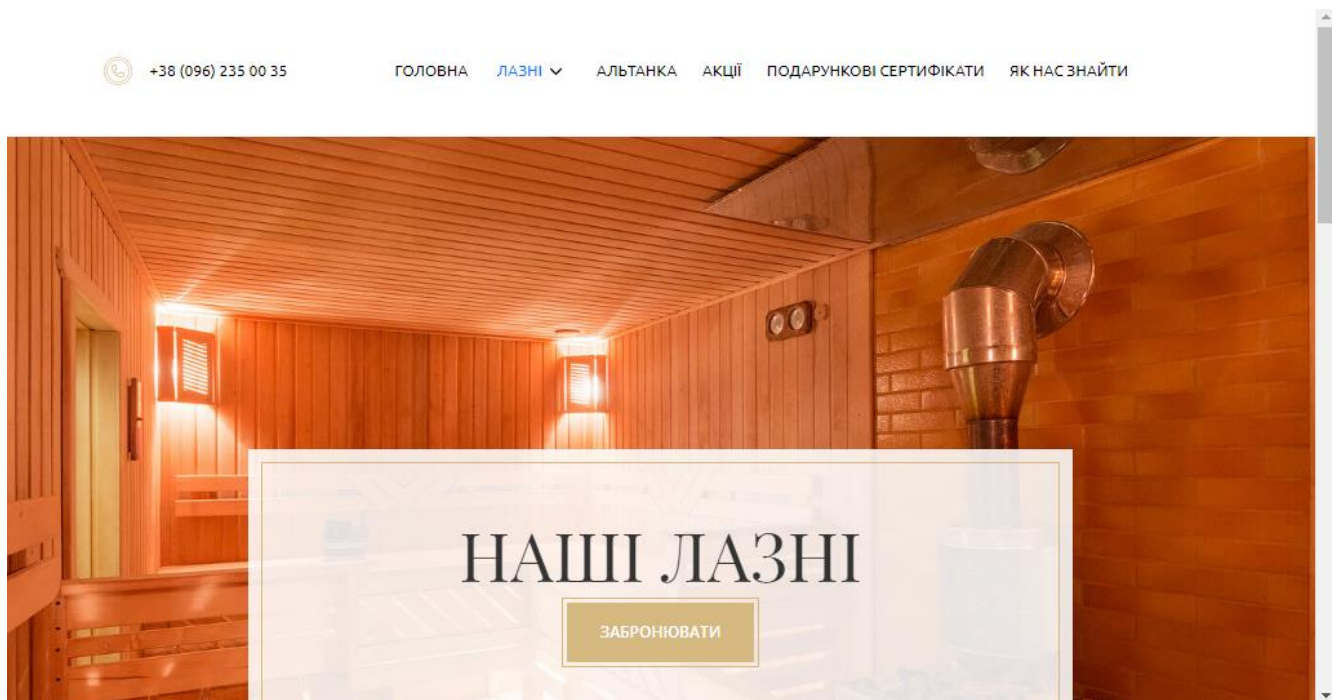


Рисунок 1.4 – Відсутність назви бази відпочинку на інших вкладках

Форма заповнення бронювання зручна, без зайвих дій та реєстрації – це зменшить витрачання часу клієнта та придасть зручність, а з іншого боку – немає захисту від атак ботів, наприклад, САРТСНА. Ще, як недолік, у формі, там де поле часу, немає вибору варіантів часу зі списку, яке б можна було обрати. Це призведе до дезорієнтації клієнта, та витрачання часу операторів, менеджерів чи адміністратора. А також при не заповненні полів та натискання кнопки

«Надіслати»- система ніяк не реагує і не показує помилки, що не всі поля були заповнені користувачем. Тобто людина може не зрозуміти, що вона не виконала і не зможе замовити послугу, а база відпочинку втратить клієнта і гроші.

Форма бронювання представлена на рисунку 1.5:

The screenshot shows a web form for booking a service. At the top, there is a logo and a close button (X). The main heading reads "ЗАЛИШІТЬ ЗАЯВКУ НА БРОНЮВАННЯ І МИ ВАМ ЗАТЕЛЕФОНУЄМО". Below this, a prompt says "Оберіть лазню чи/та алятанку." followed by a dropdown menu with "Оберіть" and a downward arrow. Another prompt says "Вкажіть дату та час бажаного бронювання." followed by two input fields: "18-04-2023" and "11:00". Below these is a prompt: "Заповніть Ваші контактні дані, щоб ми могли обробити заявку і зв'язатися з Вами." followed by three input fields: "Ваше ім'я", "E-mail", and "Мобільний Номер". At the bottom center is a prominent "НАДІСЛАТИ" button.

Рисунок 1.5 – Форма бронювання послуги на сайті бази відпочинку «П'ятниця»

Розглянемо другий аналог веб-додатку «Княжий Двір» [9].

Як і в попередньому аналозі, навігація представлена у вигляді горизонтального меню. Він буде більш популярний серед клієнтів – привабливий, простий та інтуїтивно зрозумілий інтерфейс, має 3 мови сайту, зручне та узгоджене меню, на кожній його вкладці відображається назва комплексу. Але є і мінус сайту – відсутня адаптивність під різні екрани пристроїв.

Головна сторінка зображена на рисунку 1.6:

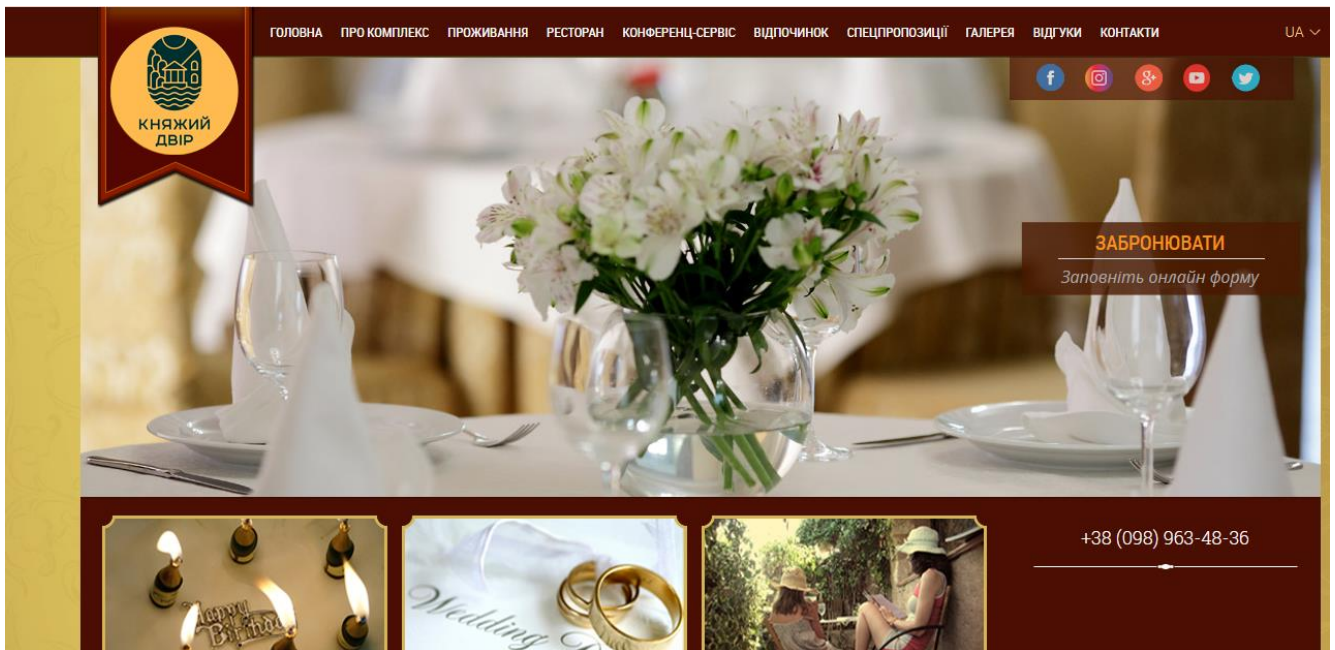


Рисунок 1.6 – Головна сторінка веб-ресурсу комплексу «Княжий двір»

Наявність вкладки «Коментарі» підніме довіру клієнтів до даної бази відпочинку. Тут є форма для написання свого відгуку з використанням технології CAPTCHA, що дозволить запобігти атакам ботів (рис. 1.7)

Відгуки

Людмила 15.05.2022
Щиро вдячні Вам за гостину, за чудові смачні страви та затишні номери. В нас було одразу дві події - ювілей мами нареченого та наше одруження. Обрали ваш заклад та залишилися задоволені. Дякуємо !!! Завітаємо обов'язково ще не один раз!

Аліна 25.01.2021
Мали нагоду відвідати готельно-ресторанний комплекс «Княжий Двір», були приємно враженні сервісом і відношенням персоналу до відпочиваючих. Все дуже сподобалось, обов'язково повернемося !!

Нина 11.01.2017
Корпоратив дело серьёзное и готовятся к нему всегда заранее. При этом важно не только выбор места, а и таких неотъемлемых составляющих праздника как кухня, гостеприимство хозяев. Ведь даже самая мелочь может испортить праздник и настроение гостям. В этом году мы сделали бесспорно правильный выбор. Кооператив прошел на ура в Комплексе Княжий двор. Стоит отметить, лояльность персонала комплекса к клиентам. Ведь все наши пожелания были удовлетворены, и мы остались довольны праздником.

Залишити відгук

Як вас звати

E-mail

Текст повідомлення

Введіть код з зображення:


Залишити відгук

Рисунок 1.7 – Вкладка «Відгуки» веб-сайту комплексу «Княжий двір»

Також тут є сповіщення про помилки, коли не заповненні всі поля (рис. 1.8).

Подтвердите действие на странице www.kdvir.com

Заповніть всі поля!

OK

Залишити відгук

Як вас звати

E-mail

Текст повідомлення

Введіть код з зображення:

8048

Залишити відгук

Рисунок 1.8 – Повідомлення при не заповненні всіх полів веб-сайту комплексу «Княжий двір»

Форма бронювання представлена на рисунку 1.9. Тут є все необхідне, нічого зайвого, повністю задовольняє потреби клієнта, і теж є вікно сповіщення при не заповненні всіх полів.

Ім'я *

Телефон *

Email *

Дата заїзду *

Дата виїзду *

Кількість дорослих *

Кількість дітей

Отель

Тип номера

Коментарий

ЗАБРОНЮВАТИ

Рисунок 1.9 – Форма бронювання на веб-сайту комплексу «Княжий двір»

Третім аналогом буде розглянуто веб-сайт готельно-відпочинкового

комплексу «Чарівна Долина» [10]. Головна сторінка представлена нижче на рисунку 1.10.

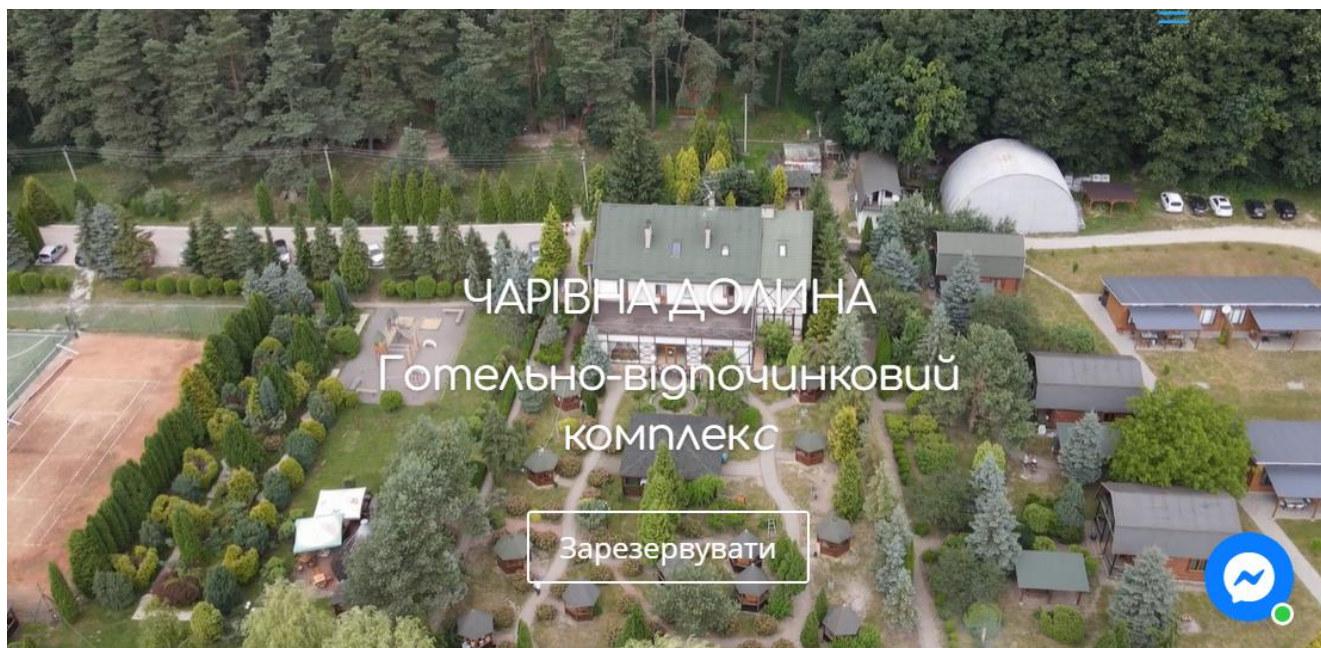


Рисунок 1.10 – Головна сторінка веб-сайту комплексу «Чарівна Долина»

Перше, що можна сказати за головну сторінку, тут є анімація, це значно впливає на швидкість завантаження сторінки. Дизайн сайту не зручний, текст зливається з зображенням, кнопку зарезервувати погано видно. Навігаційне меню відсутнє, вся інформація з'являється при прокрутці вниз сторінки. Інтерфейс не інтуїтивно зрозумілий та простий. Все це буде безпосередньо впливати на популярність та відвідування сайту, що призведе до зниження прибутку.

Даний веб-сайт має тільки одну мову, інших немає.

Розділ для відгуків відсутній, довіра нових клієнтів буде вже знижена.

Адаптація під різні екрани пристроїв та масштабованість є. Якщо відкрити головну сторінку з планшета, вже буде видно номер телефону та пошту, але меню не працює. Тобто сайт не повністю функціональний.

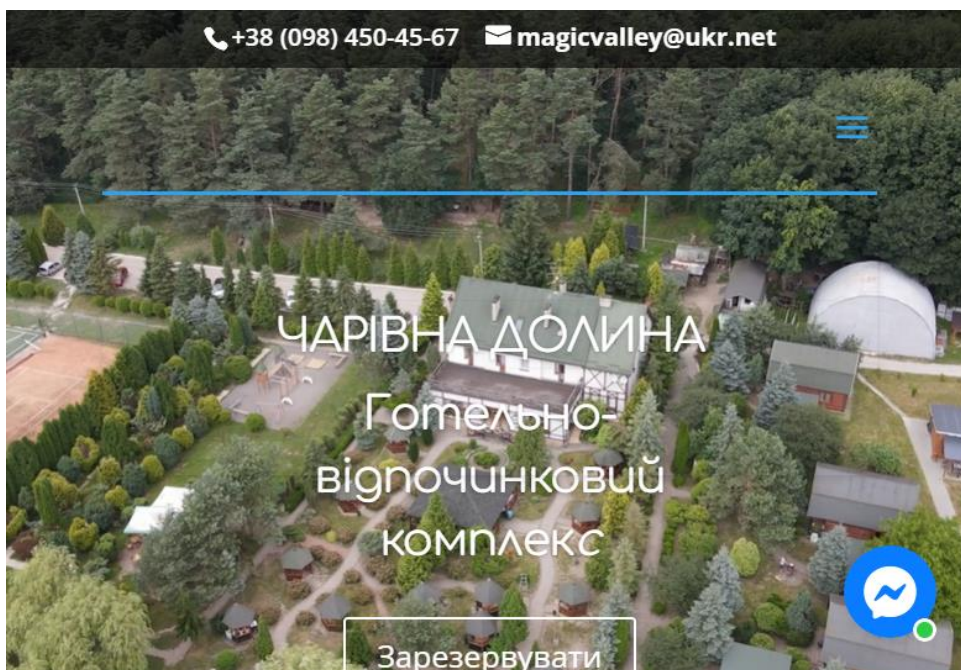


Рисунок 1.11 – Головна сторінка з планшету веб-сайту комплексу «Чарівна Долина»

При натисканні кнопки «Зарезервувати» з’явиться форма бронювання, яка представлена на рисунку 1.12. З мінусів це те, що тут нема поля для зазначення проміжку часу та дати і теж є вікно сповіщення при не заповненні всіх полів.

Доба (24h): 4300 грн

Повідомте нас про свій приїзд на Чарівну Долину

Рисунок 1.12 – Форма бронювання на веб-сайту комплексу «Чарівна Долина»

Якщо не були заповнені всі поля форми, то спрацює сповіщення про це як на рисунку 1.13.

Повідомте нас про свій приїзд на
Чарівну Долину

Please, fill in the following fields:

- Ваше ім'я
- Номер телефону
- Повідомлення

Ваше ім'я

Номер телефону







Рисунок 1.13 – Повідомлення при не заповненні всіх полів веб-сайту комплексу «Чарівна Долина»

Нижче представлена таблиця результату аналіз аналогів веб-сайтів з чотирма стовпчиками: емблема та назва веб-сайту, їх переваги, недоліки.

Таблиця 1.1 Порівняння аналогів веб-сайтів

Веб-додаток	Переваги	Недоліки
 «П'ятниця»	Зручний інтерфейс, висока швидкість завантаження веб-сайту, адаптивність під різні розміри екрану, зручна форма бронювання.	Наявність тільки однієї мови, відсутність назви бази відпочинку на деяких вкладках, немає вкладки «Контакти», а також відгуків, немає технології CAPTCHA. При не заповненні полів та натискання кнопки «Надіслати»- система ніяк не реагує і не показує помилки, що не всі поля були заповнені користувачем..
 «Княжий Двір»	Привабливий, простий та інтуїтивно зрозумілий інтерфейс, зручне вікно бронювання, має 3 мови сайту, зручне та узгоджене меню, висока швидкість завантаження	Відсутня адаптивність під різні екрани пристроїв

		веб-сайту, є форма для написання свого відгуку з використанням технології CAPTCHA, є сповіщення про помилки, коли не заповненні всі поля	
	«Чарівна Долина»	Адаптація під різні екрани пристроїв та масштабованість.	Низька швидкість завантаження веб-сайту, не зручний дизайн сайту, лише одна мова, відсутній розділ відгуків. Якщо відкрити головну сторінку з планшету, вже буде видно номер телефону та пошту, але меню не працює. Тут нема поля для зазначення проміжку часу та дати.

1.3 Постановка задачі

Головна задача роботи полягає у розробці веб-додатку для бази відпочинку мовою JavaScript.

Розроблений веб-додаток повинен виконувати такі завдання:

- Надання загальної інформації про базу відпочинку;
- Надання інформації про послуги;
- Надання контактів;
- Можливість бронювати котедж на вказану дату та час;
- Відправка броні на електронну пошту;

База відпочинку матиме назву "Затишний куточок". Це добре облаштована територія, площею більше 10 га, з двома озерами та різноманітними тваринами. Тут можна насолодитися гармонією з природою, відпочиваючи в двох затишних котеджах, відвідаючи кафе або відпочиваючи в альтанках під час перекусу. Крім того, тут є будиночок на вуликах, оздоровчий чан та багато інших розваг, таких як: батут, надувна гірка, дитячий майданчик і т. д., які допоможуть людям та вашим дітям гарно провести час у комфортній атмосфері.

В біопарку проживають до 30-ти видів ссавців та до 100-та видів птахів, які забезпечені камфорними умовами, щоб вони почували себе як вдома в природі.

Кожний котедж обладнаний побутовою технікою (холодильник, електрочайник, мікрохвильова піч, телевізор), посудом, постільною білизною та рушниками, Wi-Fi.

Надається послуга «Сон на вуликах» - унікальна процедура оздоровлення всього організму. Ця процедура має ряд позитивних властивостей, а саме:

- благотворний вплив на судини і серце: нормалізує артеріальний тиск, усуває порушення циркуляції крові, позбавляючи від набряків;
- лікувальну дію на дихальні шляхи: знімає спазми і полегшує вивід мокротиння, очищує бронхи;
- прискорює метаболізм, нормалізує обмін речовин, очищає організм від шлаків, допомагає позбавитися від зайвої ваги і схильності до переїдання;
- позитивно позначається на роботі репродуктивної і сечостатевої систем;
- уповільнює процеси старіння, служить мірою профілактики від вікових хвороб опорно-рухової та серцево-судинної систем;
- наповнює життєвою силою, усуває відчуття втоми;
- знижує ризик виникнення онкологічних захворювань.

2 ДОСЛІДЖЕННЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 HTML

HTML - це така мова розмітки, що використовується для повідомлення вашому браузеру, як відобразити веб-сторінки, які ви відвідуєте. Вона може бути складною або простою, залежно від того, як веб-дизайнер хоче. HTML складається з ряду елементів, які ви використовуєте, щоб вкладати або обертати різні частини контенту, щоб змусити контент відобразитися або діяти певним чином.

Опанування HTML є важливим навичкою для веб-розробників, оскільки воно є основою для подальшого вивчення веб-технологій та розробки веб-додатків. Документація та ряд онлайн-ресурсів доступні для вивчення HTML, і вона може бути відмінним відправним пунктом для новачків у веб-розробці.

У 1989 році Бернерс-Лі написав меморандум, в якому пропонував систему гіпертексту в Інтернеті. Бернерс-Лі вказав HTML і написав програмне забезпечення для браузера та сервера наприкінці 1990 року.

Першим загальнодоступним описом HTML був документ під назвою «HTML Tags», який вперше згадав в Інтернеті Тім Бернерс-Лі наприкінці 1991 року.

Після того, як термін дії чернеток HTML і HTML+ закінчився на початку 1994 року, IETF створив робочу групу HTML, яка в 1995 році завершила «HTML 2.0», першу специфікацію HTML, яка мала розглядатися як стандарт, на якому повинні базуватися майбутні реалізації.

Головними частинами є:

1. Відкриваючий тег (Opening tag) : Складається з імені елемента (в даному випадку, "p"), укладеного в кутові дужки, що відкривають і закривають. Відкриваючий тег вказує, де елемент починається чи починає діяти, у разі — де починається абзац.

2. Закриваючий тег (Closing tag): Це те саме, що і тег, що відкриває, за винятком того, що він включає в себе косу межу перед ім'ям елемента. Закриваючий елемент показує, де елемент закінчується, у разі — де

закінчується абзац. Відсутність тега, що закриває, є однією з найпоширеніших помилок початківців і може призводити до дивних результатів.

3. Контент (Content) : Це контент елемента, який у разі є просто текстом.

Елемент(Element) : Відкриваючий тег, що закриває тег і контент разом складають елемент.

4. Атрибути містять додаткову інформацію про елемент, який ви не бажаєте показувати у фактичному контенті. У даному випадку class це ім'я атрибута, а editor-notepad значення атрибута. Клас дозволяє дати елементу ідентифікаційне ім'я, яке може пізніше використовуватися, щоб звертатися до елемента з інформацією про стиль та інші речі.

Атрибут завжди повинен мати:

1. Пробіл між ним та ім'ям елемента (або попереднім атрибутом, якщо елемент вже має один або кілька атрибутів).
2. Ім'я атрибута, за яким слідує знак рівності.
3. Значення атрибута, укладене з двох сторін у лапки [15].

Основні елементи HTML включають:

<html>: кореневий елемент, який визначає початок і кінець HTML-документа.

<head>: використовується для збереження метаданих HTML-документа, таких як заголовки сторінки, підключення зовнішніх файлів CSS або JavaScript, тощо.

<title>: визначає заголовок сторінки, який відображається в рядку заголовка веб-переглядача.

<body>: містить вміст сторінки, такий як текст, зображення, посилання, тощо.

<h1>, <h2>, <h3> і т.д.: використовуються для визначення заголовків різних рівнів.

<p>: використовується для відображення абзаців з текстом.

: використовується для вставки зображень на сторінку з вказанням шляху до зображення та альтернативного тексту.

<a>: використовується для створення посилань на інші веб-сторінки або ресурси.

2.2 CSS

Знання CSS є важливим для створення сучасних веб-інтерфейсів, розробки веб-сайтів та веб-додатків. Використання CSS дозволяє забезпечити стильний та привабливий вигляд веб-сторінок, зручну навігацію та взаємодію з користувачем.

CSS (Cascading Style Sheets) - це мова програмування, яка використовується для опису зовнішнього вигляду веб-сторінок, створених з використанням мови розмітки HTML. CSS дозволяє веб-розробникам задавати стилі, такі як кольори, розміри, шрифти, відступи, рамки та інші властивості елементів на веб-сторінці, що дозволяє керувати виглядом і розташуванням елементів на сторінці.

CSS був вперше запропонований Хоконом Віумом Лі 10 жовтня 1994 року. У той час Лі працював з Тімом Бернерсом-Лі в CERN. Приблизно в той же час було запропоновано кілька інших мов таблиць стилів для Інтернету, а дискусії в публічних списках розсилки та в Консорціумі World Wide Web призвели до першої рекомендації W3C CSS (CSS1), яка була випущена в 1996 році. Зокрема, пропозиція Берта Боса мала вплив; він став співавтором CSS1 і вважається співавтором CSS.

CSS розроблено для того, щоб уможливити розділення вмісту та презентації, включаючи макет, кольори та шрифти. Це розділення може покращити доступність вмісту; забезпечують більшу гнучкість і контроль у специфікації характеристик презентації; дозволити кільком веб-сторінкам використовувати форматування, вказавши відповідний CSS в окремому файлі .css, що зменшує складність і повторення структурного вмісту; і ввімкніть хешування файлу. CSS, щоб покращити швидкість завантаження сторінок між сторінками, які спільно використовують файл, і його форматування.

Основні поняття та концепції CSS включають:

- Селектори: Використовуються для вибору HTML-елементів, до яких будуть застосовані стилі. Селектор може бути класом, ідентифікатором, типом елемента або іншими атрибутами.
- Властивості: Визначають стилі, такі як кольори, шрифти, розміри,

відступи, рамки та інші властивості елементів.

- Значення: Встановлюють конкретні значення для властивостей, такі як розміри в пікселях, кольори в HEX-або RGB-форматі, типи шрифтів тощо.
- Каскадність: CSS застосовує каскадність, що означає, що стилі можуть бути успадковані від батьківських елементів, а також перезаписуватися більш конкретними стилями.
- Блоки оголошень: CSS може бути включений в HTML-документи за допомогою внутрішніх стилів, зовнішніх файлів CSS, вбудованих стилів або використанням сторонніх бібліотек CSS.
- Резиновий дизайн: CSS також дозволяє створювати адаптивний дизайн, що забезпечує коректне відображення веб-сторінок на різних пристроях з різними розмірами екранів.

Специфікації CSS підтримує Консорціум Всесвітньої павутини (W3C). Тип Інтернет-медіа (тип MIME) `text/css` зареєстровано для використання з CSS відповідно до RFC 2318 (березень 1998 р.). W3C надає безкоштовну службу перевірки CSS для документів CSS.

CSS має простий синтаксис і використовує кілька англійських ключових слів для визначення назв різних властивостей стилю. Таблиця стилів складається зі списку правил . Кожне правило або набір правил складається з одного або кількох селекторів і блоку оголошень.

У CSS селектори оголошують, до якої частини розмітки застосовується стиль, зіставляючи теги й атрибути в самій розмітці.

Селектори можуть застосовуватися до наступного:

- усі елементи певного типу, наприклад, заголовки другого рівня `h2`;
- елементи, визначені атрибутом , зокрема:
- `id` : ідентифікатор, унікальний у документі, позначений на мові селектора префіксом хешу, наприклад `#id`
- `клас` : ідентифікатор, який може анотувати кілька елементів у документі, позначається префіксом `.classname`;

- елементів залежно від того, як вони розміщені відносно інших у дереві документа [16].

2.3 JavaScript

JavaScript (JS) є мовою програмування високого рівня програмування, яка є однією з основних технологій Всесвітньої павутини, поряд з HTML і CSS. Вона дозволяє додавати динаміку, взаємодію та функціональність до веб-сайтів, реалізовувати взаємодію з користувачем, виконувати різні операції на сторінці, обробляти події, взаємодіяти з веб-сервером та багато іншого.

Станом на 2022 рік 98% веб-сайтів використовують JavaScript на стороні клієнта для поведінки веб-сторінок, часто використовуючи бібліотеки сторонніх розробників

Усі основні веб-браузери мають спеціальний механізм JavaScript для виконання коду на пристроях користувачів.

Перший популярний веб-браузер із графічним інтерфейсом користувача, Mosaic, був випущений у 1993 році. Доступний для людей, які не мають технічних знань, він відіграв помітну роль у швидкому розвитку Всесвітньої павутини, що зароджувалася. Потім провідні розробники Mosaic заснували корпорацію Netscape, яка випустила більш досконалий браузер Netscape Navigator у 1994 році. Він швидко став найбільш використовуваним.

Найперша реалізація JavaScript була створена Бренданом Ейхом у компанії Netscape, і з того часу оновлюється, щоб відповідати ECMA-262 Edition 5 і пізнішим версіям. Цей двигун називається SpiderMonkey і реалізований мовою C/C++. Двигун Rhino створений Норрісом Бойдом (англ. Norris Boyd) і створено мовою Java. Як і SpiderMonkey, Rhino відповідає ECMA-262 Edition 5.

У 1996 році компанія Microsoft випустила аналог мови JavaScript, названий JScript. Анонсована ця мова була 18 липня 1996 року. Першим браузером, який підтримує цю реалізацію, був Internet Explorer 3.0.

Основні можливості JavaScript включають:

1) Маніпуляція DOM: JavaScript може змінювати структуру та вигляд веб-сторінок за допомогою Document Object Model (DOM), що дозволяє змінювати вміст, стилі, атрибути та інші властивості елементів на сторінці.

2) Реакція на події: JavaScript може обробляти події, такі як кліки мишею, введення даних, клавіші на клавіатурі та багато інших, для взаємодії з користувачем та виконання відповідних дій.

3) Взаємодія з веб-сервером: JavaScript може взаємодіяти з веб-сервером, виконувати запити та отримувати дані з сервера без перезавантаження сторінки, що дозволяє реалізовувати асинхронний обмін даними.

4) Робота з даними: JavaScript дозволяє виконувати операції з даними, такі як робота з рядками, числами, масивами, об'єктами та іншими типами даних, що дозволяє виконувати складні операції та логіку на веб-сторінках.

5) Створення функцій та класів: JavaScript підтримує об'єктно-орієнтований підхід до програмування, дозволяючи створювати власні функції та класи, що робить його потужним засобом для створення розширених функціональних компонентів та бізнес-логіки на веб-сторінках.

6) Обробка помилок: JavaScript має можливості для обробки помилок та виключень, що дозволяє забезпечити більш надійну та стабільну роботу веб-додатків, виявляти та вирішувати проблеми.

7) Використання сторонніх бібліотек та фреймворків: JavaScript має велику кількість сторонніх бібліотек та фреймворків, таких як jQuery, React, Angular, Vue та багато інших, які розширюють можливості мови та спрощують розробку веб-додатків.

8) Взаємодія з іншими технологіями: JavaScript може взаємодіяти з іншими технологіями, такими як AJAX, WebSockets, LocalStorage, IndexedDB та інші, що дозволяє розробникам створювати багатофункціональні веб-додатки.

JavaScript використовується як на стороні клієнта (клієнтський JavaScript) для взаємодії з веб-сторінками у браузері, так і на стороні сервера (серверний JavaScript) за допомогою платформ та середовищ виконання, таких як Node.js. Він має велику спільноту розробників та багатий екосистему інструментів, що робить

його потужним і високо популярним інструментом для розробки веб-додатків [17].

2.4 Visual Studio Code

Visual Studio Code (зазвичай відомий як VS Code) - це відкрите, безкоштовне та поросте середовище розробки (IDE) від Microsoft, призначене для розробки різноманітних типів додатків, включаючи веб-додатки, мобільні додатки, хмарні додатки та інші.

Редактор допоможе в роботі веб-розробникам та верстальникам усіх рівнів. З одного боку, він підходить новачкам, тому що його інтерфейс інтуїтивно простий і зрозумілий. З іншого боку, в VS Code вбудовано багато можливостей, які цікаві досвідченим розробникам.

Він поставляється з вбудованою підтримкою JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов і середовищ виконання (таких як Java, Python, C++, C#, C, PHP, Go, .NET).

VS Code із коробки підтримує роботу з системою контролю версій Git. Можна виконати базові операції та переглянути, як змінилися файли з моменту останнього коміту.

VS Code відрізняє від конкурентів наявність вбудованого відладчика. Після невеликого налаштування можна шукати баги в коді прямо з редактора, наприклад, поставити точку зупинки та спостерігати за виконанням конкретної ділянки коду. Крім цього редактор має вбудовану консоль, в яку може виводитися результат роботи або повідомлення про помилку, якщо щось пішло не так. Відладчик можна налаштувати під різні мови та різні завдання

Основні особливості Visual Studio Code включають:

1) Кросплатформеність: VS Code доступний на різних операційних системах, таких як Windows, macOS та Linux, що дозволяє розробникам використовувати його на різних платформах.

2) Розширюваність: VS Code має велику кількість розширень, які розширюють його можливості та дозволяють використовувати різноманітні мови

програмування, фреймворки, бібліотеки, інструменти та інші розширені функції.

3) Редактор коду з підсвіткою синтаксису: VS Code має потужний редактор коду з підсвіткою синтаксису для багатьох мов програмування, що робить розробку більш зручною та продуктивною.

4) Вбудована підтримка системи керування версіями: VS Code має вбудовану підтримку популярних систем керування версіями, таких як Git, що дозволяє розробникам здійснювати контроль версій свого коду безпосередньо в IDE.

5) Відладка: VS Code надає можливість відладки коду для різних мов програмування, дозволяючи розробникам ефективно відстежувати та виправляти помилки у своєму коді.

6) Інтеграція з веб-технологіями: VS Code має ряд вбудованих функцій, таких як підтримка HTML, CSS, JavaScript, TypeScript, Node.js, та інших веб-технологій, що дозволяє зручно розробляти веб-додатки.

7) Співпраця: VS Code має вбудовану підтримку розширень для різних фреймворків та бібліотек, таких як React, Angular, Vue, Django, Flask, Laravel, та інші, що дозволяє зручно розробляти додатки на популярних платформах та фреймворках.

8) Інтелектуальні функції: VS Code надає ряд інтелектуальних функцій, таких як автодоповнення, підказки, висвітлення коду, переходи між файлами, рефакторинг та інші, що покращують продуктивність розробника.

9) Інтеграція з іншими інструментами: VS Code може легко інтегруватись з іншими популярними інструментами розробки, такими як термінал, Docker, GitHub, Jira, та багатьма іншими, що сприяє зручній розробці та співпраці в команді.

10) Активна спільнота: VS Code має велику активну спільноту розробників, яка постійно розвиває нові розширення, надає підтримку та допомогу, що робить це середовище розробки популярним та актуальним [13].

Ось тому було обрано даний інструмент для розробки програмного забезпечення.

2.5 Express.JS

Express.JS — швидкий, непереборний, мінімальний і гнучкий веб-фреймворк для Node.js, який забезпечує надійний набір функцій для веб- і мобільних додатків.

Завдяки безлічі методів утиліт HTTP та проміжного програмного забезпечення у вашому розпорядженні створити надійний API можна швидко й легко.

Express надає тонкий рівень основних функцій веб-додатків, не приховуючи функцій Node.js .

Основні особливості Express.js:

1) **Мінімалістичність:** Express.js є легким і мінімалістичним фреймворком, що надає лише базовий набір функцій, не нав'язуючи жодних обмежень щодо архітектури або структури додатку. Це дозволяє розробникам вільно організовувати свій код і використовувати різні бібліотеки та модулі за їхнім бажанням.

2) **Роутинг:** Express.js має потужну систему роутингу, що дозволяє визначати різні маршрути та обробляти різні HTTP-запити, такі як GET, POST, PUT, DELETE та інші. Це дозволяє легко налаштовувати роутинг логіки веб-додатка.

3) **Middleware:** Middleware є центральним елементом Express.js, що дозволяє розробникам виконувати різні функції обробки запитів, такі як автентифікація, авторизація, логування, перехоплення помилок, обробка CORS, та багато інших. Middleware може бути доданий до ланцюжка обробки запитів в будь-якому порядку, що дозволяє гнучко налаштовувати обробку HTTP-запитів.

4) **Розширюваність:** Express.js дозволяє розробникам додавати власні роутери, middleware, шаблонізатори та інші функції за допомогою власних розширень (middleware) або встановлювати сторонні пакети.

Велика спільнота: Express.js має велику спільноту розробників, що підтримує його активним розвитком та надає велику кількість ресурсів, документації, прикладів коду та плагінів. Це робить його одним з найпопулярніших виборів для розробки веб-додатків на платформі Node.js [14].

2.6 NODEMAILER

Nodemailer - це бібліотека на платформі Node.js, яка дозволяє відправляти електронні листи з використанням простого та зрозумілого API. Вона дозволяє розробникам легко відправляти електронні листи з Node.js-додатків, незалежно від поштового провайдера, такого як Gmail, Yahoo, Microsoft Exchange, інші SMTP-сервери або навіть власний SMTP-сервер.

Це популярний відкритий пакет, який дозволяє розробникам легко використовувати Node.js для відправки електронних листів з їх додатків.

Nodemailer надає ряд функцій, таких як відправка текстових та HTML-листів, вкладення файлів, встановлення заголовків листа, використання SSL / TLS для захищеного з'єднання, аутентифікація на SMTP-сервері, обробка помилок, шаблони листів та багато іншого. Вона також підтримує використання різних провайдерів електронної пошти, таких як SMTP, Sendmail, Amazon SES, Mailgun, Gmail, інші.

Nodemailer має кілька переваг, таких як:

1) Простота використання: Nodemailer має простий і легкий в освоєнні API, що дозволяє легко відправляти електронні листи з вашого додатку Node.js без необхідності глибокого розуміння протоколу SMTP.

2) Гнучкість: Nodemailer надає широкі можливості налаштування електронних листів, таких як налаштування кодування, форматування тексту, вкладення файлів, безпеки та багато іншого. Це дозволяє вам створювати різноманітні види листів відповідно до ваших потреб.

3) Підтримка різних транспортів: Nodemailer дозволяє використовувати різні транспорти для відправлення листів, такі як SMTP, Sendmail, Amazon SES, Microsoft Exchange та інші. Це дає вам можливість вибрати найбільш зручний транспорт для вашого додатку.

4) Обробка помилок: Nodemailer надає можливість обробляти помилки, пов'язані з відправкою листів, такі як відмови серверу, помилки автентифікації, неправильні адреси електронної пошти і т. д. Це дозволяє вам взаємодіяти з

помилками, які можуть виникнути під час відправки листів.

5) Розширені можливості: Nodemailer має ряд розширених можливостей, таких як відправлення HTML-листів, використання шаблонів, вкладення файлів, робота з декількома адресатами, розробка динамічного контенту та інші. Це дає вам можливість створювати різноманітний контент в електронних листах [18].

3 РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Html частина

Розробку сайту розпочнемо з написання HTML коду. Усього у нас буде сім сторінок. Головна сторінка, з інформацією, чотири сторінки з послугами, сторінка з формою для відправки даних на пошту бази відпочинку для бронювання котеджу та сторінка з контактами.

Кожна сторінка має метадані (рис. 3.1), шапку з навігаційним меню (рис. 3.2), тіло сторінки та підвал (рис. 3.3). Також додаємо кнопку переміщення сторінки вгору для зручності користування сайтом.

```

<!DOCTYPE html>
<html lang="ua">
<!-- Метаданні сторінки -->

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

Рисунок 3.1 – метадані html сторінки

```

<header>
  <nav class="two">
    <ul class="menu">
      <span id="nameCompany">База відпочинку "Затишний куточок"</span>
      <li><a href="index.html" style="border-bottom: 4px solid □ black;">Головне меню</a></li>
      <li><a href="services/services.html">Наші послуги<i class="fa fa-angle-down"></i></a>
        <ul class="submenu">
          <li><a style="font-size: 19px" href="services/BIOpark.html">БІОпарк</a></li>
          <li><a style="font-size: 19px" href="services/Cottages.html">Котеджі</a></li>
          <li><a style="font-size: 19px" href="services/Sleep.html">Сон на вуликах</a></li>
        </ul>
      </li>
      <li><a href="Booking/form.html">Бронювання</a></li>
      <li><a href="contacts/contacts.html">Контакти</a></li>
    </ul>
  </nav>
</header>

```

Рисунок 3.2 – шапка з навігаційним меню

```

<footer>
  База відпочинку "Затишний куточок". Працюємо без вихідних.<br>
  м.Одеса, Україна <br>Телефон +380-000-000000; E-mail: milena2002@ukr.net
</footer>

```

Рисунок 3.3 – підвал сторінки

На головній сторінці додаємо слайдер, де будемо відображати послуги, з можливістю переходу на них (рис. 3.4).

```

<div class="textSlide"><b>Наші послуги:</b></div>
<section class="main-menu">
  <!-- Фото для слайдера -->
  <div class="slideshow-container">
    <div class="mySlides fade">
      <div class="numbertext">1 / 3</div>
      <a href="services\BIOpark.html"></a>
      <div class="text">БІОпарк</div>
    </div>
    <div class="mySlides fade">
      <div class="numbertext">2 / 3</div>
      <a href="services\Cottages.html"></a>
      <div class="text">Котеджі та риболовля</div>
    </div>
    <div class="mySlides fade">
      <div class="numbertext">3 / 3</div>
      <a href="services\Sleep.html"><br></a>
      <div class="text">Сон на вуликах</div>
    </div>
  </div>
  <!-- Кнопки для слайдера -->
  <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
  <a class="next" onclick="plusSlides(1)">&#10095;</a>
</div>
</section>

```

Рисунок 3.4 – слайдер на головній сторінці

На сторінках з послугами бази відпочинку, також додаємо слайдери з фотографіями.

Сторінка з формою для відправки даних на пошту бази відпочинку для бронювання котеджу. Має власне саму форму з інтерактивними елементами керування (рис. 3.5) та модального вікна, для показу сповіщення про відправку даних (рис. 3.6).

```

<div class="container">
  <form id="myForm">

    <label for="fname">Ім'я</label>
    <input type="text" id="fname" name="firstname" placeholder="Your name..">

    <label for="lname">Прізвище</label>
    <input type="text" id="lname" name="lastname" placeholder="Your last name..">

    <label for="dateS">Дата заселення</label>
    <input type="date" name="dateS" id="dateS">

    <label for="dateF">Дата виселення</label>
    <input type="date" name="dateF" id="dateF">

    <label for="phone">Ваш номер телефону</label>
    <input type="text" id="phone" name="phone" placeholder="Your phone...">

    <label for="wish">Побажання</label>
    <textarea id="wish" name="wish" placeholder="Write something.." style="height:200px"></textarea>

    <p><button type="send" id="send">Забронювати</button></p>

  </form>
</div>

```

Рисунок 3.5 – форма бронювання

```

<div class="modal">
  <div class="modal_dialog">
    <div class="modal_content">
      Дякуємо. Ми зв'яжемося з Вами найближчим часом
      <form action="#">
        <div data-close class="modal_close">&times;</div>
        <div id="modalText" class="modal_title"></div>
      </form>
    </div>
  </div>
</div>

```

Рисунок 3.6 – модальне вікно

Сторінка з контактними даними має google карту з відміткою місця знаходження «Затишний куточок» (рис 3.7).

```

<iframe
  src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d175848.00715441958!2d30
  width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"
  referrerpolicy="no-referrer-when-downgrade"></iframe>

```

Рисунок 3.7 – google карта

3.2 JavaScript

Далі ми робимо frontend та backend частину сайту. Frontend частину робимо за допомогою JavaScript. Для початку зробимо щоб наші слайдери на сторінках працювали (рис 3.8).

```
//Пошук елементів слайдеру на сторінці html
let slides = document.getElementsByClassName("mySlides");

let slideIndex = 1;
showSlides(slideIndex);
//Функція для кнопок переключення слайду
function plusSlides(n) {
  showSlides(slideIndex += n);
}
//Функція для слайдера. Відповідає за роботу слайдера.
function showSlides(n) {
  let i;
  if (n > slides.length) { slideIndex = 1 }
  if (n < 1) { slideIndex = slides.length }
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slides[slideIndex - 1].style.display = "block";
}
```

Рисунок 3.8 – слайдер

Потім зробимо так, що при прокрутці сторінки у самий низ, з'являлася кнопка, що повертає сторінку вгору (рис. 3.9).

```
//Пошук кнопки вгору на сторінці html
const btnUp = {
  el: document.querySelector('.btn-up'),
  show() {
    // видаляємо у кнопки клас btn-up_hide
    this.el.classList.remove('btn-up_hide');
  },
  hide() {
    // додаємо до кнопки клас btn-up_hide
    this.el.classList.add('btn-up_hide');
  },
  addEventListener() {
    // під час прокручування вмісту сторінки
    window.addEventListener('scroll', () => {
      // визначаємо величину прокручування
      const scrollY = window.scrollY || document.documentElement.scrollTop;
      // якщо сторінка прокручена більше ніж на 400px,
      // то робимо кнопку видимою, інакше приховуємо
      scrollY > 400 ? this.show() : this.hide();
    });
    // при натисканні на кнопку .btn-up
    document.querySelector('.btn-up').onclick = () => {
      // перемістимо на початок сторінки
      window.scrollTo({
        top: 0,
        left: 0,
        behavior: 'smooth'
      });
    };
  }
}
btnUp.addEventListener();
```

Рисунок 3.9 – Кнопка вгору

На сторінці з формою, зробимо так, щоб при натисканні кнопки «Забронювати» з’являлось модальне вікно, воно буде повідомляти користувача, що данні відправлені. При натисканні на кнопку «Закрити вікно», «Escape» або порожню частину екрану, зникало модальне вікно (рис. 3.10).

```
//Обробник події при натисканні кнопки "Відправити анкету."  
function closeModal() {  
    modal.classList.add('hide');  
    modal.classList.remove('show');  
    document.body.style.overflow = '';  
    myForm.reset();  
};  
  
//Обробник події при натисканні на "закрити модальне вікно".  
modalCloseBtn.addEventListener('click', closeModal);  
  
//Обробник події при натисканні на екран, при появі модального вікна "  
modal.addEventListener('click', (e) => {  
    if (e.target === modal) {  
        closeModal()  
    }  
});  
  
//Обробник події при натисканні кнопки  
document.addEventListener('keydown', (e) => {  
    if (e.code === 'Escape' && modal.classList.contains('show')) {  
        closeModal()  
    }  
})
```

Рисунок 3.10 – Модальне вікно

Для реалізації backend частини, а саме відправки даних користувача на пошту, нам необхідно зробити локальний сервер. Використовувати для цього будемо Express - фреймворк веб-застосунків Node.js. Після його встановлення в наш проект, потрібно його налаштувати. Вказавши порт для сервера, бібліотеку json десеріалізації тіла запиту та вказуємо ресурс через який отримуємо статичні публічні дані (js та css), (рис. 3.11).

```
const PORT = 3001;  
app.use(express.json());  
app.use('/public', express.static('public'));
```

Рисунок 3.11 – Налаштування express.js

Також нам потрібно встановити модуль nodemailer. Після чого створюємо функцію для відправки листа з вказанням конфігурацій (рис. 3.12).

```
const transporter=nodemailer.createTransport({
  host: 'smtp.ukr.net',
  port: 465,
  secure: true,
  auth: {
    user: 'milena2002@ukr.net',
    pass: 'EBhvqKrKTIXbqdmf'
  },
})
```

Рисунок 3.12 – Параметри nodemailer

Готуємо тіло письма для відправки (рис. 3.13).

```
function sendBookingMail(fname, lname, dateF, dateS, phone, wish) { //функція готує тіло листа та викликає відправлення повідомлення
  let text= `Бронювання котеджу. Клієнт: ${fname} ${lname}, час бронювання з ${dateS} до ${dateF}. Номер телефону: ${phone}. Побаження клієнта: ${wish}`;
  let html = `Бронювання котеджу. Клієнт: ${fname} ${lname}, час бронювання з ${dateS} до ${dateF}. Номер телефону: ${phone}. Побаження клієнта: ${wish}`;
  sendMail(text, html, 'milena2002@ukr.net','milena2002@ukr.net')
}
```

Рисунок 3.13 – Повідомлення для відправки

Створюємо ресурс для відправки письма на пошту (рис. 3.14).

```
app.post('/notification', (req, res) => { // ресурс який надсилає повідомлення
  let fname = req.body.fname;
  let lname = req.body.lname;
  let dateF = req.body.dateF;
  let dateS = req.body.dateS;
  let phone = req.body.phone;
  let wish = req.body.wish;
  console.log(fname);
  console.log(lname);
  console.log(dateF);
  console.log(dateS);
  console.log(phone);
  console.log(wish);
  mailer.sendBookingMail(fname, lname, dateF, dateS, phone, wish);
  res.status(202).send();// відправка відповіді клієнту
})
```

Рисунок 3.14 – Додавання можливості серверу відправляти письмо

І прив'яжемо цю функцію до кнопки «Забронювати» (рис. 3.15).

```
//Обробник події при натисканні кнопки "Відправити анкету."
send.addEventListener('click', function(event){

  let data = {
    fname: document.getElementById('fname').value,
    lname: document.getElementById('lname').value,
    dateF: document.getElementById('dateF').value,
    dateS: document.getElementById('dateS').value,
    phone: document.getElementById('phone').value,
    wish: document.getElementById('wish').value,
  };

  fetch("/notification", { // отправка сообщения на сервер
    method: "POST",
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify(data)
  }).then(res => {
    console.log("Request complete! response:", res);
  });

  modal.classList.add('show');
  modal.classList.remove('hide');
  document.body.style.overflow = 'hidden';
  event.preventDefault();
},false);
```

Рисунок 3.15 – Прив'язка відправки письма до кнопки

3.3 CSS

В кінці додаємо стилі до наших елементів сайту. Додаємо нові шрифти взяті з google fonts (рис 3.16).

```
@import url('https://fonts.googleapis.com/css2?family=Marck+Script&display=swap');
@import url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');
```

Рисунок 2.16 –Шрифти з google fonts

Змінюємо розмір тексту та його місце розташування. Зробимо навігаційне меню з випадаючим списком, а при наведенні, щоб активні елементи

підсвічувались (рис. 3.17).

```
.menu a:hover {
  color: □rgb(109, 109, 109);
}

.menu a:active {
  color: □#2c2e53;
}

ul {
  list-style: none;
  margin: 0;
  padding: 0;
  text-align: center;
}

.submenu {
  position: absolute;
  top: 100%;
  left: 0;
  width: 100%;
  z-index: 10;
  -webkit-transition: 0.25s ease-in-out;
  -moz-transition: 0.25s ease-in-out;
  -o-transition: 0.25s ease-in-out;
  transition: 0.25s ease-in-out;
}

.two .submenu {
  visibility: hidden;
  background: □#cccccc;
}

.two ul li:hover .submenu {
  visibility: visible;
}

.submenu li a {
  border-bottom: 1px solid □rgba(255,255,255,.3);
  color: □black;
}
```

Рисунок 3.17 – Навігаційне меню

Додаємо стилі до кнопок (рис. 3.18).

```
button {
  color: white;
  background-color: #7a7a7a;
  font-size: 20px;
  margin: 10px;
  border-radius: 5px;
  border: 2px solid black;
}

button:hover {
  background-color: black;
}

button:active {
  background-color: #7a7a7a;
}

#send {
  color: white;
  background-color: #7a7a7a;
  font-size: 20px;
  border-radius: 5px;
  border: 2px solid black;
  margin: 0;
  height: 50px;
}

#send:hover {
  background-color: black;
}

#send:active {
  background-color: #7a7a7a;
}
```

Рисунок 3.18 – Кнопки

Міняємо колір заднього фону, шапки та підвалу сайту. Також потрібно додати стилі для модального вікна (рис. 3.19) та слайдерів (рис 3.20).

```

.modal {
  position: fixed;
  top: 0;
  left: 0;
  z-index: 1050;
  display: none;
  width: 100%;
  height: 100%;
  overflow: hidden;
  background-color: rgba(0, 0, 0, .5)
}

.show {
  display: block
}

.hide {
  display: none;
}

.modal_content {
  text-align: center;
  font-size: 19px;
  position: relative;
  width: 100%;
  padding: 40px;
  background-color: #fff;
  border: 1px solid rgba(0, 0, 0, .2);
  border-radius: 4px;
  max-height: 80vh;
  overflow-y: auto
}

.modal_dialog {
  max-width: 500px;
  margin: 40px auto
}

.modal_title {
  text-align: center;
  font-size: 22px;
}

```

Рисунок 3.19 – Модальне вікно

```

/* Стили для маленького слайдера на головній сторінці*/
.main-menu .slideshow-container {
  max-width: 600px;
  max-height: 300px;
  position: relative;
  margin: auto;
}

/* Стили для великого слайдера*/
.slideshow-container {
  max-width: 800px;
  max-height: 400px;
  position: relative;
  margin: auto;
}

```

Рисунок 3.20 – Слайдер

3.3. Графічна частина

Мапа веб-сайту (sitemap) є файлом або сторінкою, що містить список всіх доступних сторінок веб-сайту. Цей список зазвичай організований у структуровану ієрархію, що відображає посилання між різними сторінками.

Мапа веб-сайту використовується для допомоги пошуковим системам, таким як Google, Bing, Yahoo та іншим, зрозуміти структуру вашого веб-сайту та ефективно індексувати його сторінки. Коли пошукова система обходить ваш веб-сайт, вона переглядає мапу сайту, щоб знайти всі посилання на сторінки, які можна проіндексувати. Це дозволяє пошуковій системі оновлювати свою базу даних посилань на ваш веб-сайт і показувати його у результатах пошуку.

Мапа веб-сайту також може бути корисною для веб-розробників та користувачів, оскільки вона надає загальний огляд структури веб-сайту і допомагає знайти конкретні сторінки швидше.

Існують різні формати мапи сайту, такі як XML-формат, який прийнятий багатьма пошуковими системами, і HTML-формат, придатний для перегляду людиною. Крім того, існують інструменти, які допомагають автоматично генерувати мапу сайту на основі структури веб-сайту. [21]

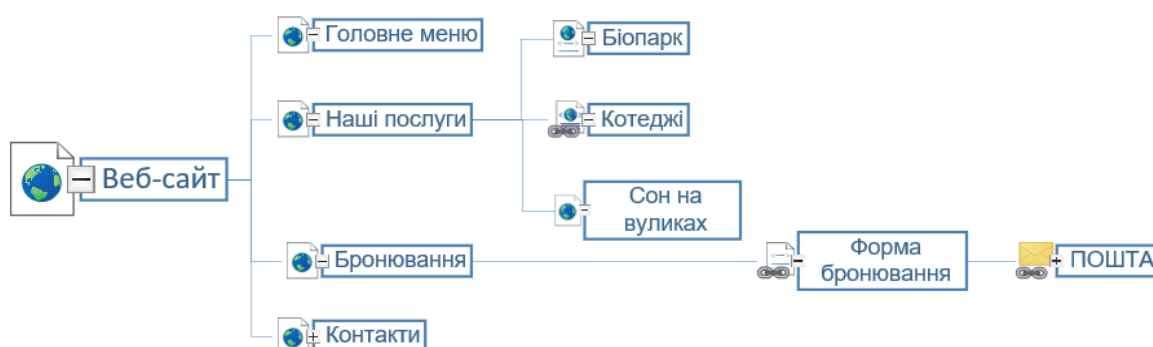


Рисунок 3.21 – Мапа розробленого веб-сайту

Карта має таку структуру: є головна сторінка, яка має вкладки (Головне меню, Наші послуги, Бронювання та Контакти). Якщо зайти на вкладку «Головне меню»-

будуть посилання на ще вкладки (Біопарк, Котеджі, Сон на вуликах).

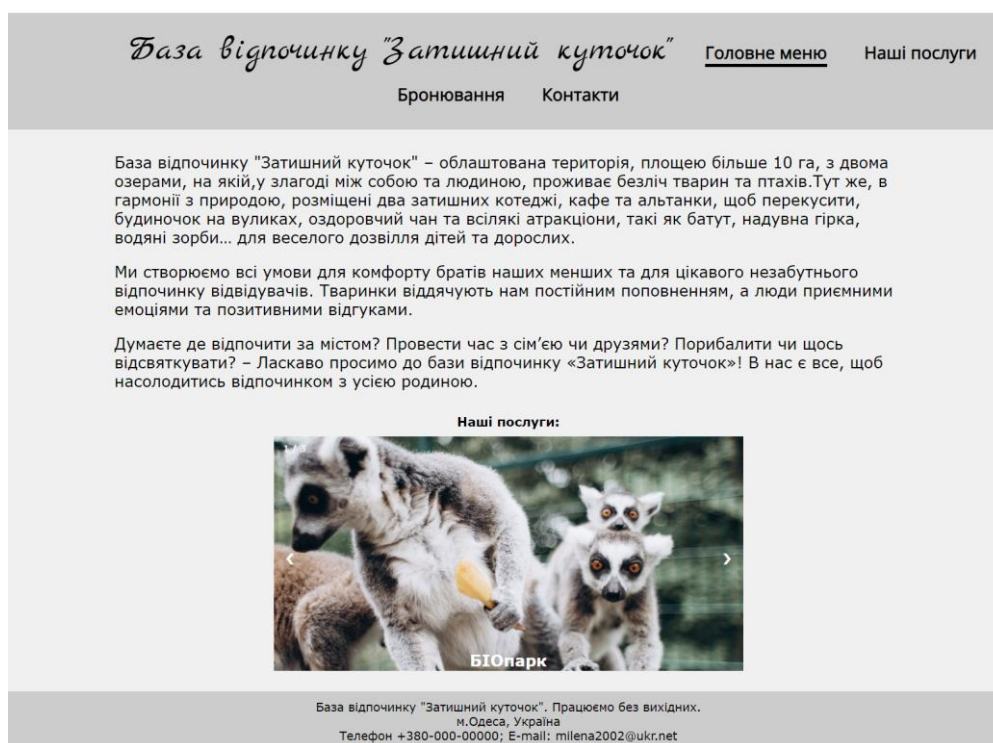


Рисунок 3.22 – Головна сторінка веб-сайту

Вкладка бронювання приведе до форми бронювання, де користувач заповнює такі поля як: Прізвище, Ім'я, дата заселення та виселення, номер телефону та побажання. Результат заповненої форми бронювання відправиться на пошту адміністратора, який передзвонить клієнту для підтвердження бронювання.

Бронювання котеджу

Ім'я

Прізвище

Дата заселення

Дата виселення

Ваш номер телефону

Побажання

Рисунок 3.23 –Екранна форма

ВИСНОВКИ

Отже, в результаті роботи було спроектовано та розроблено web-додаток для бронювання котеджів мовою програмування JavaScript з використанням CSS, HTML.

Розроблений додаток може бути використаний будь-якою базою відпочинку. Таке рішення значно зменшить витрату часу та коштів компанії при обробці бронюванні місць. Люди зможуть використовувати веб-додаток з будь-якої точки світу на будь-якій операційній системі пристрою з мінімальними вимогами, легко орієнтуючись на веб-сторінці завдяки інтуїтивно зрозумілому інтерфейсу. Користувач зможе забронювати котедж без реєстрації облікового запису, вказавши основні дані - прізвище, ім'я, дату заселення та виселення, номер телефону, не витрачаючи на це купу часу. Потім бронювання прийде на корпоративну пошту і після цього оператор бази відпочинку набере замовника для подальших дій.

Для досягнення заданої мети були виконанні наступні завдання:

1. проведення аналізу наукових та літературних джерел з тематики дослідження;
2. розгляд сучасних веб-додатків, що використовуються в галузі відпочинку;
3. проектування етапів розробки додатку;
4. проведення аналізу сучасних інструментів та програмних засобів реалізації веб-додатку;
5. написання програмного коду для розв'язування поставленого завдання;
6. описання архітектуру та функціоналу додатку;
7. проведення функціонального тестування розробленої системи;

ПЕРЕЛІК ПОСИЛАНЬ

1. Особливості веб-додатків [Електронний ресурс] – Режим доступу до ресурсу: <http://sites.znu.edu.ua>
2. Різновиди веб-систем [Електронний ресурс]. –Режим доступу: <http://websecurity.com.ua/security/chapter1/>
3. Вебсайт [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1%D1%81%D0%B0%D0%B9%D1%82>
4. Вебсторінка [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0>
5. What Is a Web Application? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.indeed.com/career-advice/career-development/what-is-web-application>
6. PRINCIPLES OF GOOD WEBSITE DESIGN [Електронний ресурс] – Режим доступу до ресурсу: <https://www.feelingpeaky.com/9-principles-of-good-web-design/>
7. 10 Fundamental UI Design Principles You Need to Know [Електронний ресурс] – Режим доступу до ресурсу: <https://dribbble.com/resources/ui-design-principles>
8. База відпочинку "П'ятниця"[Електронний ресурс] – Режим доступу до ресурсу: <https://www.bazapyatnitsa.kiev.ua/>
9. Комплекс "Княжий Двір" [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kdvir.com/>
10. Готельно-відпочинковий комплекс «Чарівна Долина» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dolyna.com.ua/>
11. Сучасний підручник JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://cutt.ly/E2Mwzxs>
12. Resources for Developers by Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org>

13. Український веб-довідник [Електронний ресурс] – Режим доступу до ресурсу: <https://css.in.ua/>
14. Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/>
15. Express.JS [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/>
16. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/HTML>
17. CSS [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/CSS#cite_note-2
18. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/JavaScript>
19. NODEMAILER [Електронний ресурс] – Режим доступу до ресурсу: <https://nodemailer.com/about/>
20. Офіційна сторінка платформи «Binary Studio Academy» [Електронний ресурс] – Режим доступу до ресурсу: <https://academy.binarystudio.com/ua/>
21. Мапа веб-сайту[Електронний ресурс] – Режим доступу до ресурсу: <https://inweb.ua/blog/ua/what-is-sitemap/>

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



«РОЗРОБКА WEB-ДОДАТКУ ДЛЯ БАЗИ ВІДПОЧИНКУ МОВОЮ JAVA SCRIPT»

Виконала студентка 4 курсу,
групи ПД– 42
Музика Мілена Андріївна
Керівник роботи
Професор, завідувач кафедри ІПЗ Бондарчук А. П.

Київ-2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: спрощення взаємної з клієнтами бази відпочинку за допомогою веб додатку.




Об'єкт дослідження: процес покращення взаємодії з клієнтом за допомогою веб додатку.

Предмет дослідження: веб технології для розробки додатку.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проведення аналізу наукових та літературних джерел з тематики дослідження.
2. Розгляд сучасних веб-додатків, що використовуються в галузі відпочинку.
3. Проектування етапів розробки додатку.
4. Проведення дослідів сучасних інструментів та програмних засобів реалізації веб-додатку.
5. Написання програмного коду для розв'язування поставленого завдання.
6. Описання архітектури та функціоналу додатку.
7. Проведення функціонального тестування розробленої системи.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Веб-додаток	Переваги	Недоліки
 «П'ятниця»	Зручний інтерфейс, висока швидкість завантаження веб-сайту, адаптивність під різні розміри екрану, зручна форма бронювання.	наявність тільки однієї мови, відсутність назви бази відпочинку на деяких вкладках, немає вкладки «Контакти», а також відгуків, немає технології САРТСНА. При не заповненні полів та натискання кнопки «Надіслати»- система ніяк не реагує і не показує помилки, що не всі поля були заповнені користувачем..
 «Княжий Двір»	привабливий, простий та інтуїтивно зрозумілий інтерфейс, зручне вікно бронювання, має 3 мови сайту, зручне та узгоджене меню, висока швидкість завантаження веб-сайту, є форма для написання свого відгуку з використанням технології САРТСНА, є сповіщення про помилки, коли не заповненні всі поля	відсутня адаптивність під різні екрани пристроїв
 «Чарівна Долина»	Адаптація під різні екрани пристроїв та масштабованість.	Низька швидкість завантаження веб-сайту, не зручний дизайн сайту, лише одна мова, відсутній розділ відгуків. Якщо відкрити головну сторінку з планшету, вже буде видно номер телефону та пошту, але меню не працює. Тут нема поля для зазначення проміжку часу та дати.

ЗАСОБИ РЕАЛІЗАЦІЇ

Під час розробки веб-додатку були, середовище розробки Visual Studio Code , технологія Express використані такі технології: мова програмування JavaScript, CSS, HTMLs.JS, бібліотека Nodemailer .



Express



ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціонал:

1. Надання загальної інформації про базу відпочинку;
2. Надання інформації про послуги;
3. Надання контактів;
4. Можливість бронювати котедж на вказану дату та час;
5. Відправка броні на електронну пошту;

АРХІТЕКТУРА ВЕБ-ДОДАТКУ

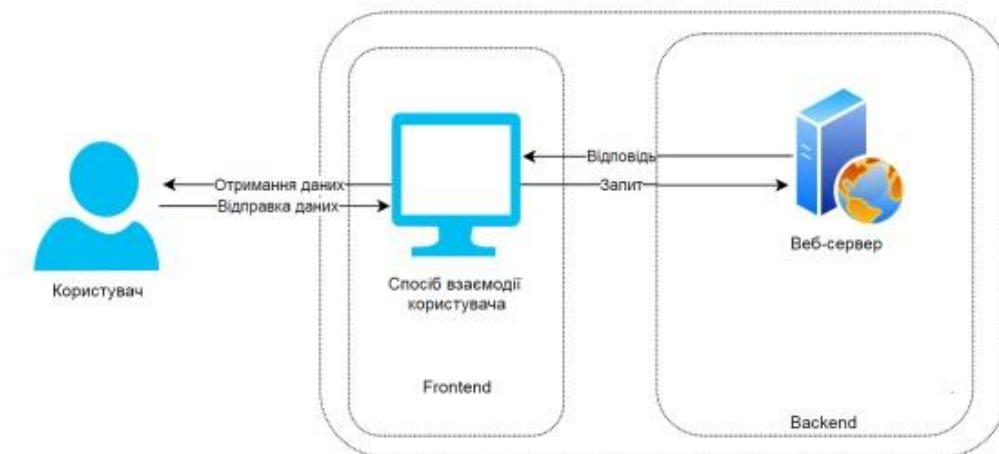
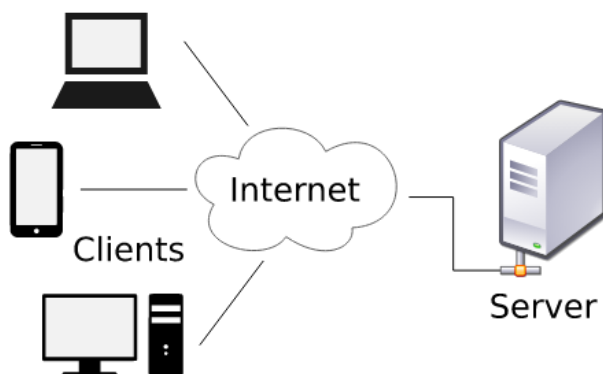


СХЕМА АРХІТЕКТУРИ ВЕБ -ДОДАТКУ



МАПА ВЕБ-САЙТУ



ГОЛОВНА СТОРІНКА ВЕБ -САЙТУ


База відпочинку "Затишний куточок" [Головне меню](#) [Наші послуги](#)
[Бронювання](#) [Контакти](#)

База відпочинку "Затишний куточок" – облаштована територія, площею більше 10 га, з двома озерами, на якій у злагоді між собою та людиною, проживає безліч тварин та птахів. Тут же, в гармонії з природою, розміщені два затишних котеджі, кафе та альтанки, щоб перекусити, будиночок на вулицях, оздоровчий чан та всілякі атракціони, такі як батут, надувна гірка, водяні зорби... для веселого дозвілля дітей та дорослих.

Ми створюємо всі умови для комфорту братів наших менших та для цікавого незабутнього відпочинку відвідувачів. Тваринки віддячують нам постійним поповненням, а люди приємними емоціями та позитивними відгукками.

Думаєте де відпочити за містом? Провести час з сім'єю чи друзями? Порибалити чи щось відсвяткувати? – Ласкаво просимо до бази відпочинку «Затишний куточок»! В нас є все, щоб насолодитись відпочинком з усією родиною.

Наші послуги:



Біопарк

База відпочинку "Затишний куточок". Працюємо без вихідних.
 м.Одеса, Україна
 Телефон +380-000-000000; E-mail: milena2002@ukr.net

ЕКРАНА ФОРМА

Бронювання котеджу

Ім'я
Your name...

Прізвище
Your last name...

Дата заселення
27.04.2023

Дата виселення
дд.мм.рррр

Ваш номер телефону
Your phone...

Побажання
Write something...

Забронювати

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Музика М. А. Розробка web-додатку для бази відпочинку мовою JavaScript/ Музика М. А., Бондарчук А. П/ ||| науково-технічна конференція, м. Київ, 26 квітня 2023 р.

ВИСНОВКИ

В результаті виконання роботи було:

1. Проведено аналіз наукових та літературних джерел з тематики дослідження.
2. Проаналізовано сучасні веб-додатки, що використовуються в галузі відпочинку.
3. Проведено дослідження сучасних інструментів та програмних засобів реалізації веб-додатку.
4. Розроблено веб-додаток бази відпочинку мовою Java Script з можливістю бронювання.
5. Описано архітектуру та функціонал додатку.
6. Проведено тестування розробленої системи.

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК Б

//Скрипт, що відповідає за форму:

```
let counter;

//Пошук елементів модального вікна на сторінці html
let myForm = document.getElementById('myForm');
const modal = document.querySelector('.modal');
const modalCloseBtn = document.querySelector('[data-close]');
let modalText = document.getElementById('modalText');
let send = document.getElementById('send');
let input = document.querySelectorAll("input");

//Обробник події при натисканні кнопки "Відправити анкету."
send.addEventListener('click', function (event) {
  counter =0;
  for (let elem of input) {
    if (elem.value.length > 0) {
      console.log();
      counter++
    }
  }

  if(counter == 5) {
    let data = {
      fname: document.getElementById('fname').value,
      lname: document.getElementById('lname').value,
      dateF: document.getElementById('dateF').value,
      dateS: document.getElementById('dateS').value,
      phone: document.getElementById('phone').value,
      wish: document.getElementById('wish').value,
```

```

    });

    fetch("/notification", { // отправка сообщения на сервер
      method: "POST",
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(data)
    }).then(res => {
      console.log("Request complete! response:", res);
    });

    modal.classList.add('show');
    modal.classList.remove('hide');
    document.body.style.overflow = 'hidden';
    event.preventDefault();
  }
}
event.preventDefault();
console.log(counter);
}, false);

//Обробник події при натисканні кнопки "Відправити анкету."
function closeModal() {
  modal.classList.add('hide');
  modal.classList.remove('show');
  document.body.style.overflow = "";
  myForm.reset();
};

//Обробник події при натисканні на "закрити модальне вікно".

```

```

modalCloseBtn.addEventListener('click', closeModal);

//Обробник події при натисканні на екран, при появі модального вікна "
modal.addEventListener('click', (e) => {
  if (e.target === modal) {
    closeModal()
  }
});
//Обробник події при натисканні кнопки
document.addEventListener('keydown', (e) => {
  if (e.code === 'Escape' && modal.classList.contains('show')) {
    closeModal()
  }
})
//Скрипт, що відповідає за кнопку "перейти на вгору сторінки"

//Пошук кнопки вгору на сторінці html
const btnUp = {
  el: document.querySelector('.btn-up'),
  show() {
    // видаляємо у кнопки клас btn-up_hide
    this.el.classList.remove('btn-up_hide');
  },
  hide() {
    // додаємо до кнопки клас btn-up_hide
    this.el.classList.add('btn-up_hide');
  },
  addEventListener() {
    // під час прокручування вмісту сторінки

```

```

window.addEventListener('scroll', () => {
  // визначаємо величину прокручування
  const scrollY = window.scrollY || document.documentElement.scrollTop;
  // якщо сторінка прокручена більше ніж на 400px,
  //то робимо кнопку видимою, інакше приховуємо
  scrollY > 400 ? this.show() : this.hide();
});
// при натисканні на кнопку .btn-up
document.querySelector('.btn-up').onclick = () => {
  // перемістимо на початок сторінки
  window.scrollTo({
    top: 0,
    left: 0,
    behavior: 'smooth'
  });
}
}
}
btnUp.addEventListener();

```

//Скрипт, що відповідає за слайдер:

```

//Пошук елементів слайдеру на сторінці html
let slides = document.getElementsByClassName("mySlides");

let slideIndex = 1;
showSlides(slideIndex);
//Функція для кнопок переключення слайду
function plusSlides(n) {
  showSlides(slideIndex += n);
}

```

```

}
//Функція для слайдера. Відповідає за роботу слайдера.
function showSlides(n) {
  let i;
  if (n > slides.length) { slideIndex = 1 }
  if (n < 1) { slideIndex = slides.length }
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slides[slideIndex - 1].style.display = "block";
}

\\css:
/* Стили сайту, крім слайдеру*/
/* Підключення шрифтів з google fonts*/
@import
url('https://fonts.googleapis.com/css2?family=Marck+Script&display=swap');
@import
url('https://fonts.googleapis.com/css2?family=Open+Sans&display=swap');

/* Стили тіла сайту*/
body {
  margin: 0;
  font-family: 'Open Sans', sans-serif;
  display: flex;
  flex-direction: column;
}

main {

```



```
flex: 1 1 auto;
display: flex;
flex-flow: column nowrap;
min-height: 100vh;
background-color: #f0f0f0;
}

.main-container {
flex: 1 0 auto;
width: 80%;
margin: auto;
}

/* Стили шапки сайту*/

header {
background-color: #cccccc;
color: black;
font-size: 40px;
padding: 10px;
font-family: 'Marck Script', cursive;
}

#nameCompany {
font-size: 44px;
padding-left: 20px;
padding-right: 20px;
margin-left: 9%;
}
```

```
.menu {  
  list-style-type: none;  
  margin: 0;  
  padding: 10px;  
  font-size: 16px;  
}  
  
.menu li {  
  display: inline-block;  
  position: relative;  
}  
  
.menu a {  
  text-decoration: none;  
  font-family: 'Open Sans', sans-serif;  
  color: black;  
  display: inline-block;  
  margin: 10px 20px 10px 20px;  
  font-size: 22px;  
  font-weight: 800;  
  
}  
  
.menu a:hover {  
  color: rgb(109, 109, 109);  
}  
  
.menu a:active {
```

```
    color: #2c2e53;
}

ul {
    list-style: none;
    margin: 0;
    padding: 0;
    text-align: center;
}

.submenu {
    position: absolute;
    top: 100%;
    left: 0;
    width: 100%;
    z-index: 10;
    -webkit-transition: 0.25s ease-in-out;
    -moz-transition: 0.25s ease-in-out;
    -o-transition: 0.25s ease-in-out;
    transition: 0.25s ease-in-out;
}

.two .submenu {
    visibility: hidden;
    background: #cccccc;
}

.two ul li:hover .submenu {
    visibility: visible;
}

.submenu li a {
    border-bottom: 1px solid rgba(255,255,255,.3);
```

```
    color: black;
}

/* Стили підвалу сайту*/

footer {
    background-color: #cccccc;
    text-align: center;
    color: black;
    font-size: 15px;
    margin: 0px;
    padding: 10px;
    flex: 0 0 auto;
}

/* Стили секцій сайту сайту*/
section {
    color: black;
    font-size: 20px;
    margin: 10px;
    overflow: hidden;
}

.textSlide {
    text-align: center;
}

h1 {
    color: black;
```

```
text-align: center;
}
```

```
.contacts {
text-align: center;
}
```

```
link {
color: #000;
}
```

```
a:visited {
color: #000;
}
```

```
/* Стилi кнопки вгору сторiнки*/
```

```
.btn-up {
position: fixed;
background-color: black;
right: 50px;
bottom: 80px;
border-radius: 30px;
cursor: pointer;
width: 60px;
height: 60px;
}
```

```
.btn-up::before {
```

```

content: "";
text-align: center;
position: absolute;
width: 20px;
height: 20px;
left: 20px;
top: 20px;
background-image: url("data:image/svg+xml,%3Csvg
xmlns='http://www.w3.org/2000/svg' width='20' height='20' viewBox='0 0 20
20'%3E%3Cg fill='none' stroke='%23fff' stroke-width='2' stroke-linecap='round' stroke-
linejoin='round'%3E%3Cpath d='M10 17V4M3 10 7 7 7
7'%3E%3C/g%3E%3C/svg%3E");
}

```

```

.btn-up_hide {
display: none;
}

```

```

@media (hover: hover) and (pointer: fine) {
.btn-up:hover {
background-color: #cccccc;
/* цвет заднего фона при наведении */
}
}

```

```

table td{
width: 30%;
margin: 40px;
}

```

```
table{  
    margin-top: 50px;  
    text-align: center;  
}
```

```
td:hover{  
    background-color:rgb(226, 226, 226);  
}
```

```
/* СТИЛІ КНОПОК*/
```

```
button {  
    color: white;  
    background-color: #7a7a7a;  
    font-size: 20px;  
    margin: 10px;  
    border-radius: 5px;  
    border: 2px solid black;  
}
```

```
button:hover {  
    background-color: black;  
}
```

```
button:active {  
    background-color: #7a7a7a;  
}
```

```
#send {  
    color: white;  
    background-color: #7a7a7a;  
    font-size: 20px;  
    border-radius: 5px;  
    border: 2px solid black;  
    margin: 0;  
    height: 50px;  
}
```

```
#send:hover {  
    background-color: black;  
}
```

```
#send:active {  
    background-color: #7a7a7a;  
}
```

```
/* Стили форми*/  
input[type=text],  
textarea {  
    width: 100%;  
    padding: 12px;  
    border: 1px solid black;  
    border-radius: 4px;  
    box-sizing: border-box;  
    margin-top: 6px;  
    margin-bottom: 16px;
```



```
    resize: vertical
}

input[type=date],
textarea {
    width: 100%;
    padding: 12px;
    border: 1px solid black;
    border-radius: 4px;
    box-sizing: border-box;
    margin-top: 6px;
    margin-bottom: 16px;
    resize: vertical
}

input:placeholder-shown {
    border: 1px solid tomato;
}

.container {
    border-radius: 5px;
    padding: 20px;
    width: 50%;
    margin: auto;
}

/* Стили модального вікна*/
```

```
.modal {  
  position: fixed;  
  top: 0;  
  left: 0;  
  z-index: 1050;  
  display: none;  
  width: 100%;  
  height: 100%;  
  overflow: hidden;  
  background-color: rgba(0, 0, 0, .5)  
}
```

```
.show {  
  display: block  
}
```

```
.hide {  
  display: none;  
}
```

```
.modal__content {  
  text-align: center;  
  font-size: 19px;  
  position: relative;  
  width: 100%;  
  padding: 40px;  
  background-color: #fff;  
  border: 1px solid rgba(0, 0, 0, .2);  
  border-radius: 4px;
```

```
    max-height: 80vh;
    overflow-y: auto
}

.modal__dialog {
    max-width: 500px;
    margin: 40px auto
}

.modal__title {
    text-align: center;
    font-size: 22px;
}

.modal__close {
    position: absolute;
    top: 8px;
    right: 14px;
    font-size: 30px;
    color: #000;
    opacity: .5;
    font-weight: 700;
    border: none;
    background-color: transparent;
    cursor: pointer
}

\\css slider:
/* Стили для слайдерів*/
```

```
* {  
    box-sizing: border-box  
}  
  
body {  
    font-family: Verdana, sans-serif;  
    margin: 0  
}  
  
.mySlides {  
    display: none  
}  
  
img {  
    vertical-align: middle;  
}  
  
/* Стили для маленького слайдера на головній сторінці*/  
.main-menu .slideshow-container {  
    max-width: 600px;  
    max-height: 300px;  
    position: relative;  
    margin: auto;  
}  
  
/* Стили для великого слайдера*/  
.slideshow-container {  
    max-width: 800px;
```

```
    max-height: 400px;
    position: relative;
    margin: auto;
}

/* Кнопки next & previous*/
.prev,
.next {
    cursor: pointer;
    position: absolute;
    top: 50%;
    width: auto;
    padding: 16px;
    margin-top: -22px;
    color: white;
    font-weight: bold;
    font-size: 18px;
    transition: 0.6s ease;
    border-radius: 0 3px 3px 0;
    user-select: none;
}

/* Измените положение "next button" */
.next {
    right: 0;
    border-radius: 3px 0 0 3px;
}

/* Добавьте черный фоновый цвет с небольшой прозрачностью*/
```

```
.prev:hover,  
.next:hover {  
    background-color: rgba(0, 0, 0, 0.8);  
}
```

```
/* Number text (1/3 etc) */
```

```
.numbertext {  
    color: #ffffff;  
    font-size: 12px;  
    padding: 8px 12px;  
    position: absolute;  
    top: 0;  
}
```

```
/* Выцветание анимации */
```

```
.fade {  
    -webkit-animation-name: fade;  
    -webkit-animation-duration: 1.5s;  
    animation-name: fade;  
    animation-duration: 1.5s;  
}
```

```
@-webkit-keyframes fade {
```

```
    from {  
        opacity: .4  
    }
```

```
    to {  
        opacity: 1  
    }
```

```
    }  
  }  
  
@keyframes fade {  
  from {  
    opacity: .4  
  }  
  
  to {  
    opacity: 1  
  }  
}  
  
/* Для маленьких экранов увеличьте размер текста */  
@media only screen and (max-width: 300px) {  
  
  .prev,  
  .next,  
  .text {  
    font-size: 11px  
  }  
}  
  
/* Подпись под изображением */  
.text {  
  color: white;  
  font-weight: 900;  
  font-size: 20px;  
  padding: 1px 12px;
```

```

    position: absolute;
    bottom: 1px;
    width: 100%;
    text-align: center;
}
\\index
<!DOCTYPE html>
<html lang="ua">
<!-- Метаданні сторінки -->

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Заголовок сторінки -->
  <title>База відпочинку "Затишний куточок"</title>
  <!-- Підключення css файлів з стилями -->
  <link rel="stylesheet" href="public/style.css">
  <link rel="stylesheet" href="public/slider.css">
</head>

<body>
  <main>
    <!-- Шапка сторінки (Назва компанії та Навігаційне меню) -->
    <header>
      <nav class="two">
        <ul class="menu">
          <span id="nameCompany">База відпочинку "Затишний
куточок"</span>

```



```

    <li><a href="index.html" style="border-bottom: 4px solid
black;">Головне меню</a></li>
    <li><a href="services\services.html">Наші послуги<i class="fa fa-
angle-down"></i></a>
        <ul class="submenu">
            <li><a style="font-size: 19px"
href="services/БІОпарк.html">БІОпарк</a></li>
            <li><a style="font-size: 19px"
href="services/Cottages.html">Котеджі</a></li>
            <li><a style="font-size: 19px" href="services/Sleep.html">Сон на
вуликах</a></li>
        </ul>
    </li>
    <li><a href="Booking/form.html">Бронювання</a></li>
    <li><a href="contacts\contacts.html">Контакти</a></li>
</ul>
</nav>
</header>
<div class="main-container">
    <!-- Тіло сторінки з інформацією -->
    <section class="info">
        <p>База відпочинку "Затишний куточок" – облаштована територія,
площею більше 10 га, з двома озерами, на
        який,у злагоді між собою та людиною, проживає безліч тварин та
птахів.Тут же, в гармонії з природою,
        розміщені
        два затишних котеджі, кафе та альтанки, щоб перекусити, будиночок
на вуликах, оздоровчий чан та всілякі
        атракціони, такі як батут, надувна гірка, водяні зорби... для веселого

```

дозвілля дітей та дорослих.

<p>Ми створюємо всі умови для комфорту братів наших менших та для цікавого незабутнього відпочинку

відвідувачів. Тваринки віддячують нам постійним поповненням, а люди приємними емоціями та позитивними

відгуками.</p>

<p>Думаєте де відпочити за містом? Провести час з сім'єю чи друзями? Порибалити чи щось відсвяткувати? –

Ласкаво просимо до бази відпочинку «Затишний куточок»! В нас є все, щоб насолодитись відпочинком з усією

родиною.</p>

</section>

<div class="textSlide">Наші послуги:</div>

<section class="main-menu">

<!-- Фото для слайдера -->

<div class="slideshow-container">

<div class="mySlides fade">

<div class="numbertext">1 / 3</div>

<div class="text">БІОпарк</div>

</div>

<div class="mySlides fade">

<div class="numbertext">2 / 3</div>


```

        style="width:100%"></a>
    <div class="text">Котеджі та риболовля</div>
</div>
<div class="mySlides fade">
    <div class="numbertext">3 / 3</div>
    <a href="services\Sleep.html"><br></a>
    <div class="text">Сон на вуликах</div>
</div>
<!-- Кнопки для слайдера -->
<a class="prev" onclick="plusSlides(-1)">&#10094;</a>
<a class="next" onclick="plusSlides(1)">&#10095;</a>
</div>
</section>
<!--Кнопка повернення сторінки нагору -->
<div class="btn-up btn-up_hide"></div>
</div>
<!--Підвал сайту з інформацією -->
<footer>
    База відпочинку "Затишний куточок". Працюємо без вихідних.<br>
    м.Одеса, Україна <br>Телефон +380-000-00000; E-mail:
milena2002@ukr.net
</footer>
</main>
<!-- Підключення скриптів для сайт -->
<script src="public\slider.js"></script>
<script src="..\public\script.js"></script>
</body>

```

```
</html>
```

//Contacts:

```
<!DOCTYPE html>
```

```
<html lang="ua">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>База відпочинку "Затишний куточок"</title>
```

```
  <link rel="stylesheet" href="../public/style.css">
```

```
</head>
```

```
<body>
```

```
  <main>
```

```
    <header>
```

```
      <nav class="two">
```

```
        <ul class="menu">
```

```
          <span id="nameCompany">База відпочинку "Затишний  
куточок"</span>
```

```
          <li><a href="../index.html">Головне меню</a></li>
```

```
          <li><a href="../services/services.html">Наші послуги<i class="fa fa-  
angle-down"></i></a>
```

```
            <ul class="submenu">
```

```
              <li><a style="font-size: 19px"  
href="../services/БІОпарк.html">БІОпарк</a></li>
```

```
              <li><a style="font-size: 19px"
```

```

href=" ../services/Cottages.html">Котеджі</a></li>
      <li><a style="font-size: 19px" href=" ../services/Sleep.html">Сон на
вуликах</a></li>
    </ul>
  </li>
  <li><a href=" ../Booking/form.html">Бронювання</a></li>
  <li><a href="contacts.html" style="border-bottom: 4px solid
black;">Контакти</a></li>
</ul>
</nav>
</header>
<section class="info contacts main-container">
  <h1>Наші контакти</h1>
  м.Одеса, Україна<br>
  Телефон +380-000-00000<br>
  Е-mail: milena2002@ukr.net<br>
  <!--Посилання на google карту-->
  <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d175848.0071544
1958!2d30.865061210605155!3d46.47593319710949!2m3!1f0!2f0!3f0!3m2!1i1024!2
i768!4f13.1!3m3!1m2!1s0x40c6318a0b864c43%3A0x129f8fe28cf2176c!2z0J7QtNC1
0YHRgdCwLCDQntC00LXRgdGB0LrQsNGPINC-
0LHQu9Cw0YHRgtGMLCA2NTAwMA!5e0!3m2!1sru!2sua!4v1674300540674!5m2
!1sru!2sua"
width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"
referrerpolicy="no-referrer-when-downgrade"></iframe>
</section>

```

```

<footer>
    База відпочинку "Затишний куточок". Працюємо без вихідних.<br>
    м.Одеса,    Україна    <br>Телефон    +380-000-00000;    E-mail:
milena2002@ukr.net
</footer>
</main>
</body>
</html>

```

//Booking form:

```

<!DOCTYPE html>
<html lang="ua">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>База відпочинку "Затишний куточок"</title>
    <link rel="stylesheet" href="../public/style.css">
</head>

<body>
    <main>
        <header>
            <nav class="two">
                <ul class="menu">
                    <span id="nameCompany">База відпочинку "Затишний
куточок"</span>
                    <li><a href="../index.html">Головне меню</a></li>

```

```

<li><a href="../services/services.html">Наші послуги<i class="fa fa-
angle-down"></i></a>
  <ul class="submenu">
    <li><a style="font-size: 19px"
href="../services/БІОпарк.html">БІОпарк</a></li>
    <li><a style="font-size: 19px"
href="../services/Cottages.html">Котеджі</a></li>
    <li><a style="font-size: 19px" href="../services/Sleep.html">Сон на
вулицах</a></li>
  </ul>
</li>
<li><a href="form.html" style="border-bottom: 4px solid
black;">Бронювання</a></li>
<li><a href="../contacts/contacts.html">Контакти</a></li>
</ul>
</nav>
</header>
<!--Форма-Анкета -->
<div class="main-container">
  <section class="info ">
    <h1>Бронювання котеджу</h1>
    <div class="container">
      <form id="myForm">

        <label for="fname">Ім'я</label>
        <input type="text" id="fname" name="firstname" placeholder="Your
name..">

        <label for="lname">Прізвище</label>

```

```
<input type="text" id="lname" name="lastname" placeholder="Your
last name..">
```

```
<label for="dateS">Дата заселення</label>
```

```
<input type="date" name="dateS" id="dateS">
```

```
<label for="dateF">Дата виселення</label>
```

```
<input type="date" name="dateF" id="dateF">
```

```
<label for="phone">Ваш номер телефона</label>
```

```
<input type="text" id="phone" name="phone" placeholder="Your
phone...">
```

```
<label for="wish">Побажання</label>
```

```
<textarea id="wish" name="wish" placeholder="Write something.."
style="height:200px"></textarea>
```

```
<p><button type="send" id="send">Забронювати</button></p>
```

```
</form>
```

```
</div>
```

```
</section>
```

```
<div class="btn-up btn-up_hide"></div>
```

```
<!--Модальне вікно-->
```

```
<div class="modal">
```

```
<div class="modal_dialog">
```

```
<div class="modal_content">
```

```
Дякуємо. Ми зв'яжемося з Вами найближчим часом
```

```
<form action="#">
```



```
<div data-close class="modal__close">&times;</div>
<div id="modalText" class="modal__title"></div>
</form>
</div>
</div>
</div>
<div class="btn-up btn-up_hide"></div>
</div>
<footer>
База відпочинку "Затишний куточок". Працюємо без вихідних.<br>
м.Одеса, Україна <br>Телефон +380-000-00000; E-mail:
milena2002@ukr.net
</footer>
</main>
<script src=" ../public/script.js"></script>
<script src=" ../public/form.js"></script>
</body>

</html>
```