

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: **«Розробка веб-додатку для збору та обміну
туристичними даними»**

Виконала: студентка 4 курсу групи Іт-41

Спеціальності 126 «Інформаційні системи та
технології»

(шифр і назва)

Прач Анастасія Вікторівна

(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.

(Прізвище та ініціали)

Рецензент: _____

(Прізвище та ініціали)

ДУБЛЯНИ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Рівень вищої освіти перший (бакалаврський)
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А. М. Тригуба

«____» _____ 2025 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Прач Анастасії Вікторівні
(прізвище, ім'я, по-батькові)

- Тема роботи «Розробка веб-додатку для збору та обміну туристичними враженнями»
Керівник роботи _____
(ім'я та прізвище, науковий ступінь, вчене звання)
Затверджена наказом Львівського НУВМБ ім. С.З. Гжицького _____
- Строк подання здобувачем роботи _____
- Вихідні дані до роботи: аналітичні дані роботи та характеристика об'єкту дослідження, опис бібліотек мов програмування, науково-технічна і довідкова література.
- Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити):
Вступ
 - Постановка задачі дослідження
 - Проектування
 - Програмна реалізація проекту
 - Охорона праці та безпека в надзвичайних ситуаціях.Висновки
Бібліографічний список.

5. Перелік графічного матеріалу :

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, Дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3	Тригуба А.М., д.т.н. проф., завідувач кафедри			
4	Городецький І.М., к.т.н., доцент кафедри інженерної механіки			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Отримання завдання. Вивчення рекомендованої літератури по темі роботи. Написання аналітичного огляду предметної області.	01.11.2024 – 15.12.2024	
2.	Проектування та опис технічного завдання, функціональних вимог та технічної сторони реалізації проекту (написання проектної частини).	16.12.2024 – 05.02.2025	
3.	Програмна реалізація проекту (вибір мови програмування, розробка сайту, тестування його роботи, розрахунок ефективності проекту)	06.02.2025 – 31.03.2025	
4.	Розгляд питань з охорони праці та безпеки у надзвичайних ситуаціях	01.04.2025 – 30.04.2025	
5.	Завершення оформлення основної частини, написання висновків та підготовка презентаційного матеріалу	01.05.2025 – 26.05.2025	
6.	Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи	27.05.2025 – 10.06.2025	

Студент

_____ (підпис)

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

_____ Тригуба А.М.

_____ (прізвище та ініціали)

УДК 004.738./:338.486

Кваліфікаційна робота: 61 сторінок текстової частини, 2 таблиці, 19 рисунків, 30 джерел літератури, 3 додатки.

Розробка веб-додатку для збору та обміну туристичними враженнями.

Прач А.В. - Кафедра інформаційних технологій. Дубляни, Львівський національний університет ветеринарної медицини та біотехнологій імені С.З. Гжицького, 2025 р.

Подано господарсько–виробничу характеристику галузі туристичних послуг. Проаналізовано предметну область, існуючі аналоги та визначено функціональні вимоги до сайту. Здійснено проектування веб-додатку відповідно до сучасних методик розробки, включаючи архітектуру клієнт-серверної взаємодії.

Запропоновано інтерфейс веб-застосунку для збирання, публікації та перегляду туристичних вражень, а також модуль рекомендацій на основі інтересів користувачів.

Здійснено аналіз травматичних ситуацій при виконанні різних робіт у сфері використанням комп'ютерної техніки, викладено питання охорони праці.

Ключові слова: веб-додаток, туризм, обмін даними, інтерфейс, враження, локальне розгортання, атракції.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	9
1.1. Поняття та класифікація туристичних веб-додатків.....	9
1.2. Огляд існуючих аналогів.....	10
1.2.1 Сайт «Tripadvisor»	11
1.2.2. Сайт «Lonely Planet».....	12
1.2.3. Сайт «TourRadar».....	13
1.3. Технічне завдання проекту	15
РОЗДІЛ 2. ВИБІР ЗАСОБІВ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ.....	17
2.1. Проектування структури веб-додатку	17
2.2. Дизайн інтерфейсу користувача.....	19
2.3. Інструменти для розробки веб-додатку	21
2.4. HTML – мова розмітки гіпертексту	22
2.5. CSS – каскадні таблиці стилів	23
2.6. JavaScript – реалізація клієнтської логіки	24
2.7. Організація зберігання даних у LocalStorage.....	25
2.8. Середовище розробки Visual Studio Code	26
2.9. Моделювання використання веб-додатку	28
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ	30
3.1. Реалізація функціональних модулів.....	30
3.2. Дизайн інтерфейсу користувача.....	32
3.3. Тестування функціоналу	33
3.4. Локальне розгортання веб-додатку та запуск через локальний сервер	35

3.5.Інструкція користувача сайту	36
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	38
4.1 Умови праці під час створення програмного забезпечення	38
4.2. Електробезпека при роботі з ПК	39
4.3. Дії у разі надзвичайних ситуаціях	40
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
ДОДАТКИ.....	47
ДОДАТОК А Фрагменти коду головної сторінки	48
ДОДАТОК Б Фрагменти коду JavaScript сторінки реєстрації та входу	56
ДОДАТОК В Фрагменти коду особистого кабінету	58

ВСТУП

Сучасний світ неможливо уявити без туризму, який перетворився на одну з найвпливовіших галузей світової економіки. Бурхливий розвиток цифрових технологій та зростаюча мобільність населення кардинально змінюють підходи до планування подорожей та способів обміну враженнями від мандрівок.

Якщо раніше туристи покладалися на друковані путівники та рекомендації знайомих, то сьогодні мандрівники прагнуть отримувати більш актуальну та персоналізовану інформацію від реальних людей. Вони хочуть не просто відвідувати туристичні атракції, а й ділитися власними відкриттями, створювати спільноту однодумців та зберігати спогади про свої пригоди.

Незважаючи на існування численних туристичних платформ та соціальних мереж, більшість з них орієнтована на комерційні цілі або має обмежені можливості для систематичного збору особистих туристичних вражень. Популярні сервіси на кшталт TripAdvisor чи Booking.com хоча і надають можливості для обміну відгуками, проте не дозволяють користувачам створювати повноцінні особисті колекції своїх подорожей з детальними спогадами та практичними порадами.

Саме ця прогалина зумовлює актуальність розробки спеціалізованого веб-додатку, який би поєднав зручний інструментарій для ведення особистого туристичного щоденника та ефективну платформу для обміну досвідом між мандрівниками.

Тематика туристичних інформаційних систем активно досліджується українськими науковцями (В. Кифяк, Т. Ткаченко, І. Школа, О. Бейдик, В. Федорченко та ін.). Водночас розробка персоналізованих платформ для обміну туристичними враженнями залишається недостатньо вивченою.

Мета роботи - розробка веб-додатку, який дозволить туристам зручно збирати, систематизувати та ділитися своїми враженнями від подорожей.

Об'єкт роботи – процес збору та обміну туристичною інформацією в цифровому середовищі.

Предмет роботи: технології та методи створення такого додатку.

Практичне значення роботи полягає у створенні функціонального веб-додатку, який стане корисним інструментом як для звичайних туристів, так і для професіоналів туристичної індустрії. Додаток може стати основою для комерційного сервісу або бути адаптованим для потреб туристичних агенцій та готелів. Результати дослідження сприятимуть розвитку методології створення туристичних інформаційних систем.

Структура роботи включає чотири розділи: перший присвячено аналізу проблеми та існуючих рішень, другий – проектуванню системи та вибору технологій, третій - практичній реалізації веб-додатку та результатам тестування. Останній відповідальний за охорону праці та безпеки, оскільки ми працюємо з технікою.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Поняття та класифікація туристичних веб-додатків

Туристичні веб-додатки представляють собою спеціалізовані програмні рішення, призначені для надання інформаційних послуг мандрівникам, планування подорожей та обміну туристичним досвідом у цифровому середовищі. Порівнюючи із звичайними веб-сайтами, вони характеризуються високим рівнем інтерактивності, персоналізації та орієнтацією на практичні потреби, власне подорожуючих.

Сучасний ринок туристичних веб-рішень можна класифікувати за кількома основними критеріями. За функціональним призначенням виділяють інформаційні платформи, які надають довідкову інформацію про туристичні дестинації, визначні пам'ятки, готелі та ресторани. Прикладами таких ресурсів є TripAdvisor, Lonely Planet, WikiTravel, які зосереджені на огляді туристичних об'єктів та збиранні відгуків від відвідувачів. Деякі з них розглянемо пізніше.

Комерційні туристичні платформи орієнтовані на продаж туристичних послуг та включають системи бронювання готелів, авіаквитків, туристичних пакетів. Серед найпопулярніших представників цієї категорії - Booking.com, Expedia, Skyscanner, які інтегрують функції пошуку, порівняння цін та онлайн-бронювання.

Планувальники подорожей становлять окрему категорію, що допомагає туристам організовувати маршрути, складати розклади та координувати логістичні аспекти поїздок. Такі додатки, як TripIt, Google Trips, Sygic Travel, зазвичай пропонують інструменти для створення персональних планів подорожей та синхронізації з календарями.

Соціальні туристичні мережі призначені для обміну досвідом, фотографіями та рекомендаціями між мандрівниками. Платформи на кшталт Foursquare, Polarsteps,

Minube дозволяють користувачам ділитися локаціями, створювати фотозвіти про подорожі та знаходити однодумців.

Туристичні веб-додатки поділяються на масові платформи для широкої аудиторії та спеціалізовані рішення для окремих груп. За технологією це веб-, мобільні та гібридні додатки. За рівнем персоналізації - статичні ресурси та інтелектуальні системи, що використовують машинне навчання для персональних рекомендацій.

Інтерактивні додатки дозволяють користувачам створювати, оцінювати та коментувати контент, формуючи спільноти. Геолокаційні сервіси надають інформацію про об'єкти поблизу, інтегруючись з навігаційними системами.

Попри різноманіття, більшість платформ мають спільні функції: пошук, фільтрація, профілі, відгуки, інтеграція з соцмережами. Спостерігається тенденція до об'єднання сервісів у комплексні платформи для планування, бронювання й обміну досвідом.

Класифікація туристичних веб-додатків є динамічною та постійно еволюціонує разом з розвитком технологій та зміною потреб користувачів. Межі між різними категоріями стають дедалі більш розмитими, а нові види додатків з'являються у відповідь на специфічні запити туристичної аудиторії.

1.2 Огляд існуючих аналогів

Метою роботи над цим розділом є аналіз функціоналу декількох популярних веб-сервісів у сфері туризму, що спеціалізуються на зборі та обміні туристичними враженнями. Цей розділ описує загальні принципи роботи обраних платформ. Для аналізу було обрано три сайти зі схожою тематикою і призначенням.

1.2.1 Сайт «Tripadvisor»

Сайт є провідною світовою платформою для туристичних відгуків та рекомендацій. Tripadvisor [15] має яскравий і впізнаваний дизайн з використанням фірмових кольорів - зеленого, білого та оранжевого. Дизайн створює дружню та довірливу атмосферу, підкреслюючи надійність платформи.

Даний веб-додаток має інтуїтивну навігаційну структуру, що дозволяє легко знайти потрібну інформацію. Головне меню містить розділи «Готелі», «Розваги», «Ресторани», «Авіаквитки», «Відпочинок» та «Оренда житла». Пошукова система дозволяє швидко знайти інформацію про конкретні місця або послуги.

На платформі користувачі можуть залишати детальні відгуки про місце відвідування. Кожен відгук містить рейтинг, фотографії, текстовий опис та корисні поради. Система рейтингів допомагає іншим користувачам швидко оцінити якість послуг.

Tripadvisor пропонує додаткові функції, такі як можливість бронювання готелів та квитків, порівняння цін, створення персональних списків бажань та інтерактивні карти. Платформа також має мобільний додаток та соціальні функції для спілкування між мандрівниками.

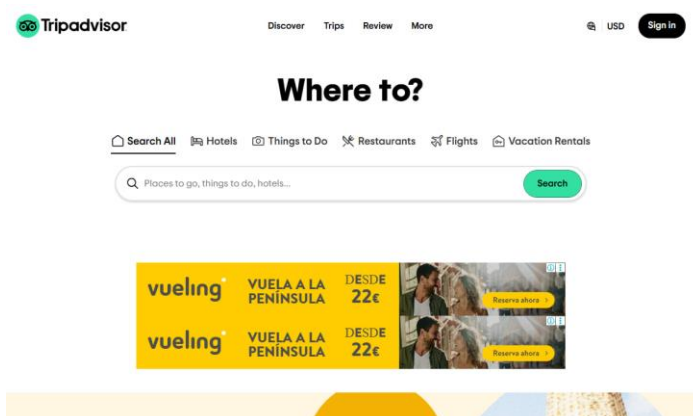


Рисунок 1.1 – Головна сторінка Tripadvisor

Загалом, Tripadvisor надає користувачам комплексний інструмент для планування власних подорожей на основі реальних відгуків та рекомендацій від інших мандрівників.

1.2.2 Сайт «Lonely Planet»

Веб-додаток «Lonely Planet» [16] є одним з найвідоміших туристичних ресурсів, який спеціалізується на наданні деталізованої інформації про напрямки подорожей, поради мандрівникам та натхнення для самостійних подорожей. Платформа має стильний, чистий дизайн з акцентом на якісний візуальний контент, а саме фотографії пейзажів, культурних пам'яток та людей.

Навігація сайту зручна і поділена за регіонами, країнами та типами досвіду: «Найкращі напрямки», «Путівники», «Подорожі з обмеженим бюджетом», «Сімейні подорожі», «Їжа та напої», «Пригоди» тощо. Є також інтерактивна карта для швидкого доступу до інформації про конкретну країну або місто.

На платформі представлені огляди міст, поради щодо транспорту, проживання, кухні, безпеки, місцевих звичаїв, а також списки вартих уваги локацій. Всі матеріали написані досвідченими travel-журналістами та неодноразовими мандрівниками. Користувачі можуть зберігати улюблені статті, створювати маршрути та читати блоги інших мандрівників.

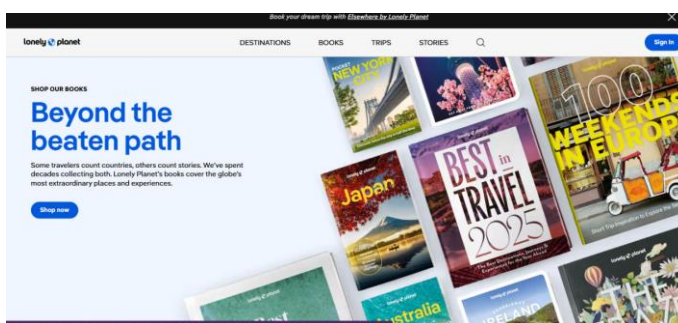


Рисунок 1.2 – Головна сторінка Lonely Planet

Lonely Planet пропонує цифрові та друковані путівники, інструменти планування маршрутів, а також розділ «Travel News», де публікуються актуальні новини для туристів. Особливу увагу приділено сталому туризму та відповідальному мандрівництву.

1.2.3 Сайт «TourRadar»

Сайт «TourRadar» [17] спеціалізується в основному на організованих турах та пригодницьких подорожах. Платформа має сучасний та динамічний дизайн з яскравими кольорами та великими зображеннями, який відразу захоплює та передає дух пригод.

Навігація сайту структурована за типами подорожей, напрямками та тривалістю. Головне меню містить розділи «Пригодницькі тури», «Культурні подорожі», «Сафарі», «Круїзи» та інші. Розширені фільтри дозволяють знайти тур за бюджетом, віковою групою та рівнем активності.

На платформі представлені детальні описи турів з фотографіями, маршрутами, включеними послугами та цінами. Користувачі можуть залишати відгуки про пройдені тури, оцінювати якість обслуговування та ділитися враженнями.

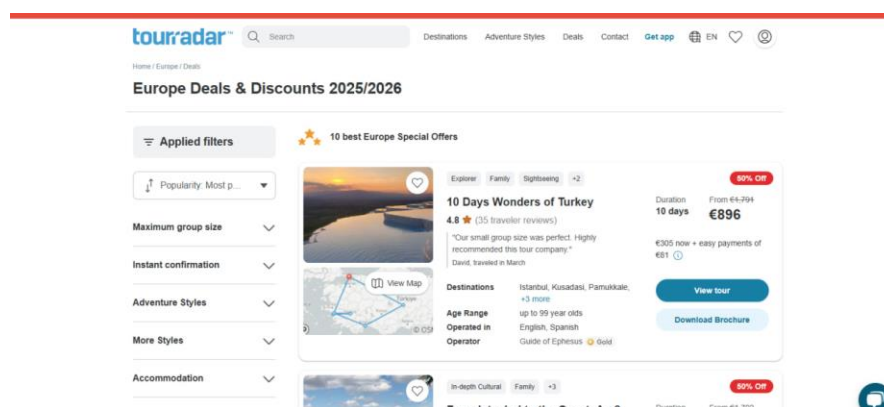


Рисунок 1.3 – Підбір туру на сайті TourRadar

TourRadar пропонує функції порівняння турів, онлайн-бронювання, систему знижок та консультації з експертами. Платформа також має розділ з корисними статтями про подорожі та поради для мандрівників.

Щоб сформувати наш майбутній веб-додаток, варто порівняти аналоги, що ми й зробили в таблиці 1.1.

Таблиця 1.1 - Порівняння аналогів сайтів

Характеристика	Lonely Planet	TourRadar	Tripadvisor
Основний функціонал	Путівники, маршрути, спільнота мандрівників	Каталог турів, бронювання, порівняння пропозицій	Відгуки, рейтинги, бронювання, рекомендації
Тип контенту	Професійні путівники + користувацький контент	Організовані тури та пригоди	Відгуки про готелі, ресторани, атракції
Цільова аудиторія	Самостійні мандрівки	Любителі активного відпочинку	Широка аудиторія туристів
Дизайн	Сучасний, пригодницький	Яскравий, динамічний	Дружній, корпоративний
Система оцінювання	Рейтинги та коментарі	Відгуки клієнтів	Рейтинги та відгуки
Соціальні функції	Форуми, Q&A, спільні поїздки	Групові тури	Форуми та Q&A
Монетизація	Продаж путівників, партнерські програми	Комісія з бронювань	Комісія з бронювань, реклама
Унікальні особливості	Офлайн путівники, експертний контент	Пригодницькі тури, групові подорожі	Найбільша база відгуків

Отже, проаналізувавши сайти зі схожою тематикою і призначенням, було вирішено розробити веб-платформу, яка об'єднає переваги всіх розглянутих вище аналогів.

1.3 Технічне завдання проекту

Метою цього проекту є розробка веб-додатку під назвою "TravelShare" з використанням сучасних веб-технологій для створення платформи для збору та обміну туристичними враженнями. До завдань проекту входить розробка повнофункціонального веб-додатку, який забезпечить ефективну взаємодію між мандрівниками та створить спільноту для обміну туристичним досвідом.

Веб-додаток "TravelShare" надасть користувачам можливість створювати персональні акаунти з детальною інформацією про профіль, включаючи особисті дані та преференції подорожей. Система реєстрації та автентифікації забезпечить безпечний доступ до персоналізованого контенту та можливість управління власними туристичними матеріалами.

Основний функціонал додатку TravelShare буде зосереджений на створенні, публікації та перегляді туристичних вражень. Користувачі зможуть ділитися детальними описами своїх подорожей, завантажувати фотографії з мандрівок, залишати відгуки про відвідані місця та створювати рекомендації для інших мандрівників. Система категоризації дозволить структурувати контент за типами подорожей, географічними регіонами та тематичними напрямками.

Додаток включатиме інтерактивну систему пошуку та фільтрації туристичних вражень, що дозволить користувачам легко знаходити релевантну інформацію про цікаві їм дестинації.

Крім того, веб-додаток буде оснащений системою збереження улюблених матеріалів, що дозволить користувачам створювати персональні колекції туристичних ідей та планувати майбутні подорожі.

Технічна реалізація передбачає використання клієнт-серверної архітектури з застосуванням HTML, CSS та JavaScript для frontend-частини, а також організацію локального зберігання даних користувачів. Ці функціональні можливості веб-додатку TravelShare дозволять створити ефективну платформу для туристичної спільноти, сприятимуть обміну досвідом між мандрівниками та забезпечать зручний інструмент для планування подорожей і пошуку туристичних рекомендацій.

РОЗДІЛ 2

ВИБІР ЗАСОБІВ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ

2.1 Проектування структури веб-додатку

На цьому етапі визначається логічна структура веб-сайту TravelShare. Створюється схема навігації, визначаються основні розділи та підрозділи, їх взаємозв'язок та ієрархія.

Важливо ретельно розробити структуру веб-сайту, оскільки це визначає організацію інформації та навігаційну логіку, що впливає на користувацький досвід та ефективність платформи для обміну враженнями про подорожі.

Структура веб-сайту "TravelShare" включатиме наступні розділи:

- Головна сторінка

Центральна сторінка містить рекомендовані враження від зареєстрованих користувачів та перехід до сторінки з відгуками. Збоку розташовано sidebar, де новий користувач може розглянути популярні напрямки, типи подорожей, а також, зацікавившись подібними публікаціями, зареєструватись, щоб мати змогу самому їх створювати або просто увійти в свій персональний акаунт.

- Відгуки подорожників

Основний розділ включає сторінку з усіма відгуками та розширеною фільтрацією за локацією, рейтингом, сортуванням та типами подорожей. Кожен відгук має окрему сторінку з повним описом, фотографіями та корисними порадами.

- Особистий кабінет

Структура кабінету поділена на вкладки:

1. Пов'язані з враженнями → вкладки надають можливість додавати нові враження з детальним описом та фотографіями, управляти власними

публікаціями, зберігати цікаві відгуки інших користувачів та отримувати персоналізовані рекомендації.

2.Налаштування профілю → користувачі можуть редагувати особисту інформацію, вказуючи власне ім'я, кредо, аватарку та електронну пошту.

➤ Категорії подорожей

Система включає п'ять основних категорій: пляжний відпочинок з морськими курортами та островами, культурний туризм з музеями та історичними пам'ятками, пригодницький туризм з екстремальними видами спорту, гастрономічний туризм з кулінарними подорожами та екотуризм з національними парками.

➤ Фільтрація

Система пошуку включає розширені фільтри за різними критеріями, зокрема, локації, сортування та інші, а також швидкий доступ до популярних напрямків.

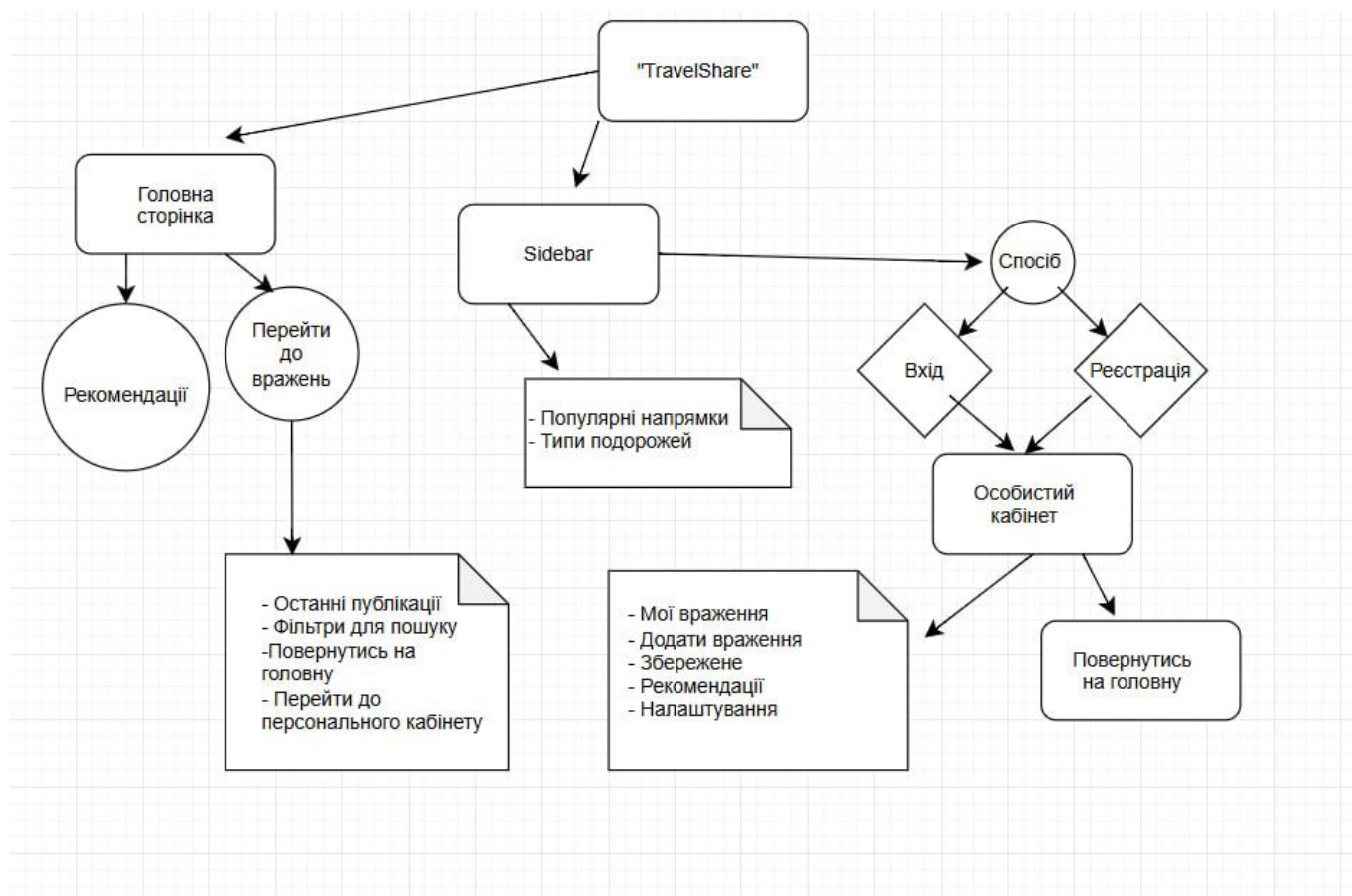


Рисунок 2.1 - Графічна структура веб-додатку

Саме така структура сайту допоможе користувачам легко орієнтуватись у контенті, знаходити цікаві відгуки про подорожі, ділитися власними враженнями та планувати майбутні мандрівки на основі досвіду інших мандрівників.

2.2 Дизайн інтерфейсу користувача

Веб-застосунок TravelShare реалізовано відповідно до сучасних вимог до веб-дизайну, з акцентом на зручність використання, візуальну привабливість та інтуїтивну навігацію. Для оформлення інтерфейсу було обрано поєднання ніжних фонів, пастельних відтінків та акцентних кольорів, що використовуються для виділення важливих елементів інтерфейсу (наприклад, кнопок, активних вкладок, заголовків).

Головна сторінка слугує точкою входу до вебзастосунку та виконує як інформаційну, так і навігаційну функції. У дизайні навігації використано чітку типографіку, зручні відступи та контрастне підсвічування активної сторінки.

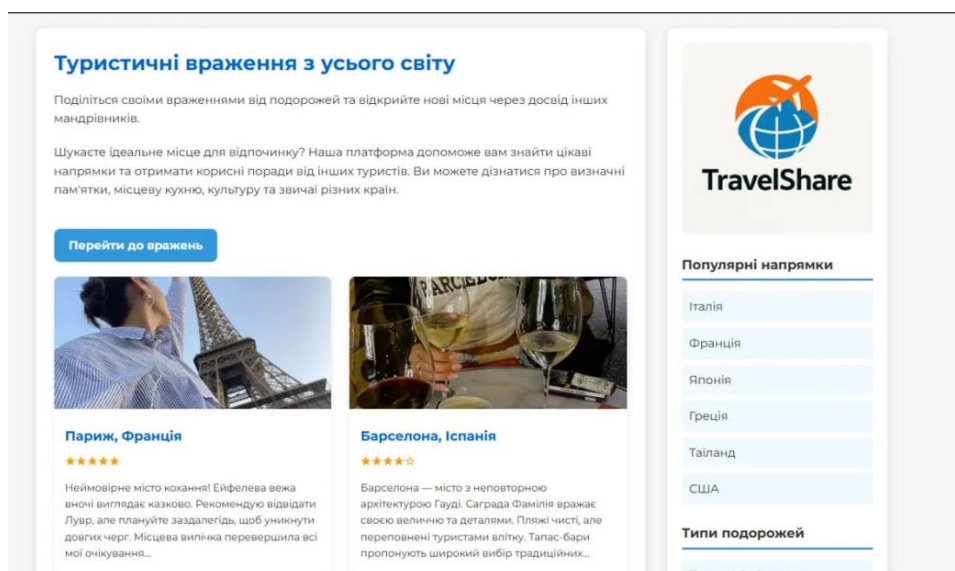


Рисунок 2.2 – Основна сторінка «TravelShare»

Центральне місце головної сторінки займає блок із популярними напрямками подорожей, представлений у вигляді горизонтальної галереї. Кожен напрямок оформлений як окрема картка з фотографією, назвою місця та коротким описом. Візуальне оформлення сприяє швидкому ознайомленню з вмістом і заохочує до подальшого перегляду. Галерея адаптивна й підтримує коректне відображення на мобільних пристроях.

Нижче розташовано секцію з типами подорожей, реалізовану у вигляді іконок із підписами. Вона дозволяє користувачеві швидко зорієнтуватися у категоріях доступного контенту.

Інтерфейс форми для додавання нових вражень розроблений з урахуванням зручності введення тексту та можливості додавання мультимедійних матеріалів. Дизайн передбачає використання великих полів вводу, кнопки завантаження зображень, а також візуальний індикатор успішного додавання або помилок валідації.

Кнопки оформлено з використанням м'яких тіней, округлених кутів і легких анімацій при наведенні, що підвищує візуальну привабливість і чітко вказує на інтерактивність елементів. Особистий кабінет витримано в єдиному стилі, що відповідає тематиці TravelShare, а візуальні елементи гармонійно інтегровані в дизайн.

Для шрифтів використано сучасні веб-безпечні гарнітури, що добре читаються на різних пристроях. Інтерфейс підтримує контрастність та дотримується принципів доступності — достатній розмір кнопок, помітні фокуси елементів, логічна послідовність переходів при навігації з клавіатури.

2.3 Інструменти для розробки веб-додатку

У цьому пункті ми розглянемо основні інструменти, які використовуються для створення сучасного туристичного веб-додатку. Кожен інструмент відіграє важливу роль у побудові структури, оформленні, програмній логіці та забезпеченні ефективного процесу розробки.

Один з основних інструментів, що використовуються у веб-розробці, - це текстовий редактор або інтегроване середовище розробки (IDE). У рамках створення туристичного веб-додатку застосовувався Visual Studio Code - сучасний багатофункціональний редактор коду, який підтримує автодоповнення, підсвічування синтаксису, перевірку помилок у реальному часі, роботу з терміналом, а також інтеграцію з системами контролю версій. Цей інструмент забезпечив зручне середовище для написання коду HTML, CSS та JavaScript.

Для побудови структури сторінки було використано мову HTML, яка дозволяє визначати блоки інтерфейсу - заголовки, форми, кнопки, розділи тощо. Стилзація сайту виконувалась за допомогою CSS, що дало змогу надати веб-додатку привабливий вигляд, реалізувати адаптивну верстку, налаштувати шрифти, кольори та інші візуальні властивості. При цьому використовувались сучасні можливості CSS, зокрема Flexbox та Grid Layout, що забезпечили гнучке компонування елементів на сторінці.

Для реалізації інтерактивних можливостей сайту була застосована мова JavaScript. Вона дозволила реалізувати динамічну зміну вмісту сторінок, обробку подій (наприклад, натискання кнопок чи перемикавання вкладок), перевірку введених користувачем даних у формах тощо. Таким чином, JavaScript виконує роль рушія логіки веб-додатку.

Окрім основних мов, важливим інструментом під час розробки є Git - система керування версіями. Вона дозволяє зберігати зміни коду, відслідковувати історію змін, створювати резервні копії та працювати над проектом у команді.

Для оформлення елементів інтерфейсу використовувались ресурси SVG-графіки та популярні бібліотеки іконок, наприклад, Font Awesome. Це дало змогу зробити інтерфейс зрозумілим, зручним і візуально привабливим для користувача.

Під час розробки активно використовувались інструменти розробника браузера (наприклад, Chrome DevTools), які дозволяють перевіряти HTML-розмітку, стилі CSS, продуктивність JavaScript, а також виявляти та виправляти помилки. Це значно полегшило процес налагодження та адаптації сайту до різних пристроїв.

На сучасному етапі веб-розробка охоплює безліч завдань - від створення простих інформаційних сторінок до складних інтерактивних додатків. Для реалізації функціонального та безпечного веб-сайту необхідно правильно підібрати мови програмування, інструменти розробки та візуальні ресурси. Вибір технологій багато в чому залежить від специфіки проекту, технічного завдання, а також від рівня підготовки самого розробника.

2.4 HTML – мова розмітки гіпертексту

HTML (Hypertext Markup Language) є головною мовою розмітки для створення веб-сторінок, що визначає структуру та семантику контенту. У веб-додатку для збору туристичних вражень HTML забезпечує організацію інформації та взаємодію користувачів з самою системою.

Семантичні теги, як-от `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, дозволяють логічно структурувати компоненти додатку - від навігації до відгуків, створюючи зрозумілу архітектуру подання даних.

Ключовим аспектом є використання HTML форм для збору користувацьких даних, що в даному разі є вкрай необхідним. Елементи `input` різних типів, `textarea` та `select` забезпечують введення описів вражень, завантаження фотографій, вибір категорій та рейтингів. Правильно структуровані форми покращують користувацький досвід при додаванні контенту до системи.

Семантична розмітка HTML покращує доступність та SEO-оптимізацію, що критично важливо для туристичних ресурсів. Використання спеціалізованих елементів для дат, зображень та локацій створює структуровані дані, які легко інтерпретуються браузером та пошуковими системами.

HTML надає сучасні можливості для мультимедійного контенту через елементи `picture`, `video` та `audio` з підтримкою адаптивного завантаження. Це особливо важливо для туристичних додатків, де візуальний контент є ключовим компонентом.

HTML структура служить основою для інтеграції з CSS та JavaScript, забезпечуючи стилізацію та інтерактивність туристичного контенту. Мета-теги та структуровані дані Schema.org покращують інтеграцію з соціальними мережами та пошуковими системами, що важливо для просування туристичного контенту.

2.5 CSS – каскадні таблиці стилів

CSS (Cascading Style Sheets) є мовою стилів, що визначає візуальне представлення веб-сторінок, включаючи позиціонування, шрифти та інші властивості HTML-елементів. У веб-додатку для збору туристичних вражень CSS забезпечує привабливий та функціональний інтерфейс для представлення туристичного контенту.

Каскадна природа CSS дозволяє застосовувати стилі на різних рівнях з використанням правил пріоритету для визначення переважних стилів. Це особливо важливо для туристичного додатку, де різні компоненти потребують специфічного стилювання: галереї фотографій, форми відгуків та користувацькі профілі.

CSS забезпечує контроль над візуальними аспектами контенту за допомогою властивостей і селекторів. Це дає змогу створювати стильні галереї, зручні форми для вражень і привабливе оформлення відгуків з рейтингами, формуючи узгоджений стиль, що відповідає тематиці й покращує користувацький досвід.

CSS також підвищує доступність туристичного додатку через правильне використання класів та ідентифікаторів, що покращує навігацію для користувачів з обмеженими можливостями. Це особливо важливо для туристичних ресурсів, які повинні бути доступними для широкого кола користувачів різного віку та технічних можливостей.

2.6 JavaScript - реалізація клієнтської логіки

JavaScript є інтерпретованою мовою програмування, що базується на концепції прототипів та широко використовується для створення інтерактивних веб-додатків. У контексті розробки веб-додатку для збору туристичних вражень, JavaScript забезпечує динамічну функціональність та покращує користувацький досвід через інтерактивні елементи інтерфейсу.

JavaScript на стороні клієнта забезпечує ключову функціональність туристичного додатку через маніпуляції з DOM для динамічного оновлення контенту без перезавантаження сторінки. Це дозволяє користувачам додавати нові враження, та переглядати фотографії в режимі реального часу.

JavaScript забезпечує валідацію форм, асинхронне оновлення контенту та зберігання користувацьких даних, що підвищує зручність і швидкість роботи TravelShare.

Потужна екосистема JavaScript надає доступ до спеціалізованих бібліотек для обробки зображень та створення інтерактивних компонентів. Простота застосування дозволяє швидко реалізувати функції автодоповнення для пошуку локацій, динамічної фільтрації контенту та реагування на користувацькі дії.

Основні обмеження JavaScript у туристичному додатку - відкритість клієнтського коду та залежність від налаштувань браузера. Різна інтерпретація скриптів вимагає тестування на різних платформах. Вимкнення JavaScript може обмежити функціональність, тож важливі опції слід дублювати альтернативно.

JavaScript є незамінним інструментом для створення сучасного туристичного веб-додатку, забезпечуючи інтерактивність, динамічність та зручність використання для збору та обміну туристичними враженнями.

2.7 Організація зберігання даних у LocalStorage

LocalStorage використовується у веб-додатку TravelShare як основний механізм клієнтського зберігання даних, що забезпечує швидкий доступ до інформації та можливість роботи в офлайн-режимі. Цей підхід дозволяє зменшити кількість запитів до сервера та покращити відгук користувацького інтерфейсу.

Архітектура зберігання даних базується на структурованих JSON-об'єктах, що містять інформацію про профіль користувача, враження та налаштування. Зберігаються особисті дані, переваги щодо подорожей і мов. Система підтримує чернетки незавершених постів, що запобігає втраті інформації. Кешування

популярних місць і нещодавно переглянутих вражень забезпечує швидке завантаження й зручність користування.

Реалізація включає механізми серіалізації та десеріалізації даних з використанням `JSON.stringify()` та `JSON.parse()`, а також систему валідації для забезпечення цілісності збереженої інформації. Кожний запис доповнюється міткою часу для контролю актуальності даних та можливості їх періодичного оновлення.

Управління розміром сховища здійснюється через стратегії стиснення даних та автоматичного очищення застарілих записів. Система контролює максимальний розмір зображень та інших медіафайлів, а також реалізує механізми резервного копіювання критично важливих даних користувача.

Безпека даних забезпечується через шифрування чутливої інформації та дотримання політики автоматичного видалення даних після тривалого періоду неактивності. Користувач має можливість повністю контролювати локально збережені дані, включаючи їх експорт та видалення.

Синхронізація з серверною частиною додатку відбувається в фоновому режимі з автоматичним вирішенням конфліктів між локальними та віддаленими версіями даних. Це забезпечує консистентність інформації across різних пристроїв користувача та гарантує збереження важливих даних навіть при локальних збоях.

Використання `LocalStorage` у контексті веб-додатку для туристичних вражень дозволяє створити надійну та ефективну систему управління даними, яка балансує між продуктивністю, безпекою та зручністю використання.

2.8 Середовище розробки Visual Studio Code

Visual Studio Code представляє собою сучасне інтегроване середовище розробки, яке було обране як основний інструмент для створення веб-додатку

TravelShare завдяки своїй універсальності та потужним можливостям для веб-розробки. Цей редактор забезпечує комплексне рішення для розробки клієнт-серверних додатків, поєднуючи легкість використання з професійними інструментами розробки.

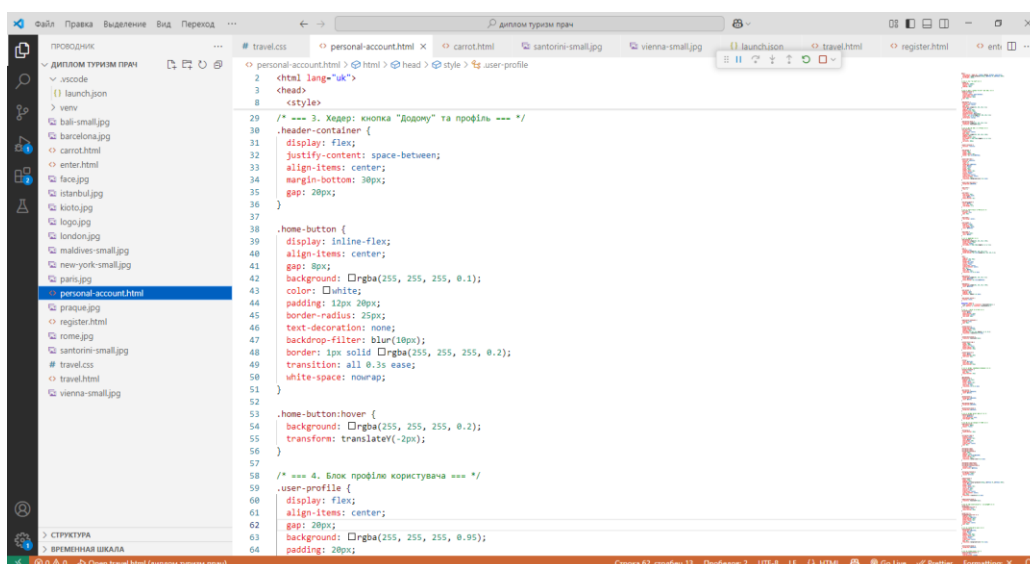


Рисунок 2.3 – Інтерфейс роботи у VS Code

Ключовою перевагою VS Code для розробки туристичного веб-додатку є його вбудована підтримка веб-технологій. Інтелектуальне підсвічування синтаксису та автодоповнення коду значно прискорюють процес написання інтерфейсних компонентів, форм реєстрації користувачів та інтерактивних елементів для відображення туристичних вражень.

Система розширень VS Code дає змогу налаштувати середовище розробки під потреби проєкту. Для TravelShare встановлено розширення для фреймворків, оптимізації зображень і автоматизації розгортання.

Інтеграція з системою контролю версій Git забезпечує ефективне управління версіями коду TravelShare, що критично важливо для відстеження змін у функціональності додатку, управління різними гілками розробки та координації роботи над різними модулями системи збору туристичних вражень.

Вбудований термінал VS Code дозволяє запускати локальний сервер, встановлювати залежності та тестувати веб-додаток, що зручно для роботи з JavaScript-фреймворками у проєкті TravelShare.

Інструменти налагодження спрощують пошук помилок, зокрема під час обробки асинхронних запитів і форм. Розширення Live Server забезпечує автоматичне оновлення сторінок, що пришвидшує тестування інтерфейсу та перегляд змін у реальному часі.

VS Code також забезпечує ефективну роботу з JSON-файлами, що використовуються для зберігання конфігурацій додатку та структурування даних туристичних вражень, а вбудовані інструменти для форматування та валідації коду гарантують високу якість та читабельність вихідного коду проєкту TravelShare.

2.9 Моделювання використання веб-додатку

Use Case Diagram - це тип діаграми, що ілюструє взаємодію користувачів із системою через типові сценарії використання. Для забезпечення ефективного функціонування веб-додатку TravelShare та його відповідності потребам користувачів було проведено детальне моделювання таких процесів, включаючи аналіз поведінки, взаємодії з інтерфейсом і ключових сценаріїв роботи з додатком.

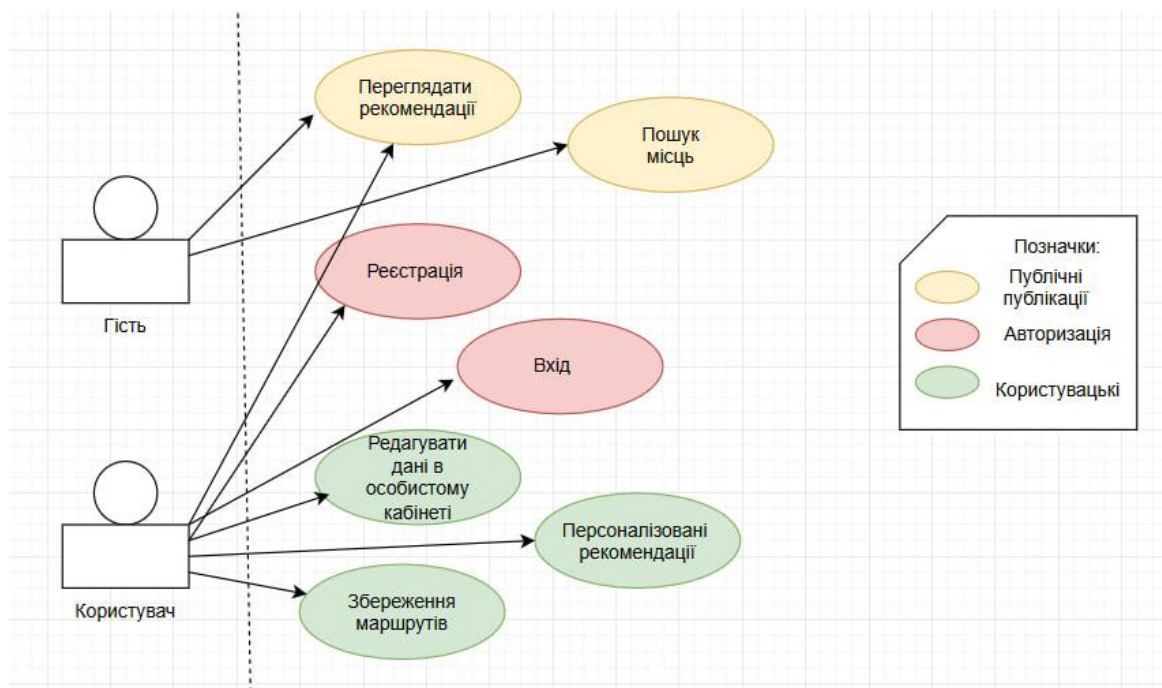


Рисунок 2.4 – Діаграма варіантів використання «TravelShare»

Моделювання використання веб-додатку дозволило структурувати функціональність, визначити ролі користувачів та оптимізувати інтерфейс. Завдяки діаграмам і чіткому розподілу дій забезпечено зручну навігацію, безпечну взаємодію з даними та стабільну роботу системи.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Реалізація функціональних модулів

У процесі розробки TravelShare було реалізовано низку функціональних модулів, які забезпечують інтерактивну взаємодію користувача із системою, підтримують збереження особистої інформації, а також сприяють соціальній взаємодії між мандрівниками. Основна увага зосереджена на таких модулях:

- реєстрація та авторизація користувачів (дозволяє створювати облікові записи, входити в систему та керувати сесіями користувача) :

```
document.getElementById('registerForm').addEventListener('submit', function(event) {
    event.preventDefault(); // блокуємо стандартну відправку форми

    let isValid = true;

    // Перевірка імені
    const firstName = document.getElementById('firstName');
    if (firstName.value.trim() === '') {
        document.getElementById('firstNameError').style.display = 'block';
        isValid = false;
    } else {
        document.getElementById('firstNameError').style.display = 'none';
    }
})
```

Рисунок 3.1 – Фрагмент коду дії перевірки

- публікація вражень (дозволяє додавати записи про подорожі з текстом, зображеннями та рейтингом):

```

const experienceForm = document.getElementById('experience-form');
const experiencesList = document.getElementById('experiences-list');

experienceForm.addEventListener('submit', e => {
  e.preventDefault();

  const location = document.getElementById('location').value.trim();
  const date = document.getElementById('date-visited').value;
  const ratingValue = parseInt(document.getElementById('rating').value);
  const description = document.getElementById('description').value.trim();
  const photoInput = document.getElementById('photo');
  const photoFile = photoInput.files[0];

```

Рисунок 3.2 – Фрагмент коду обробки форми додавання вражень

- керування особистим профілем (дозволяє редагувати особисті дані):

```

const settingsForm = document.getElementById('settings-form');
settingsForm.addEventListener('submit', (e) => {
  e.preventDefault();

  const newName = document.getElementById('settings-username').value.trim();
  const newDescription = document.getElementById('settings-description').value.trim();
  const newEmail = document.getElementById('settings-email').value.trim();
  const avatarSrc = avatarPreview.src;

  // Валідація
  if (!newName || !newDescription || !newEmail) {
    alert('Будь ласка, заповніть усі поля.');
```

```

    return;
  }

  // Оновлення профілю в шапці
  document.getElementById('user-name').textContent = newName;
  document.getElementById('user-description').textContent = newDescription;
  document.getElementById('user-avatar').src = avatarSrc;

  alert('Профіль успішно оновлено!');
});

```

Рисунок 3.3 – Фрагмент коду обробки форми налаштування профілю

Ці модулі дозволяють користувачам не лише персоналізувати свій досвід, а й активно взаємодіяти з контентом і спільнотою.

3.2 Верстка сторінок та їх стилізація

На даному етапі було виконано верстку сторінок веб-додатку та їх стилізацію. Для реалізації розмітки було використано мову HTML, а для візуального оформлення – CSS.

Сторінки побудовано з використанням семантичних тегів HTML. У структурі застосовано контейнерний підхід: основні блоки обгорнуті у div з відповідними класами для гнучкого позиціонування. Для уніфікації елементів інтерфейсу використано класи CSS з ієрархічною структурою.

Продемонструємо візуалізацію кнопки «Повернутись на головну».

```
<a href="travel.html" class="home-button">
  <i class="fas fa-home"></i> Повернутися на головну
</a>
```

Рисунок 3.4 – Приклад коду мови розмітки – HTML

Це посилання у вигляді кнопки з іконкою будинку, що веде на сторінку travel.html. Клас home-button робить її з білим текстом, прозорим фоном і плавним ефектом при наведенні.

```
.home-button:hover {
  background: rgba(255, 255, 255, 0.2);
  transform: translateY(-2px);
}
```

Рисунок 3.5 – Приклад коду CSS

Цей CSS-код задає стиль для кнопки при наведенні курсору (:hover): фон стає напівпрозорим білим, а сама кнопка трохи піднімається вгору на 2 пікселі для створення ефекту "підняття".

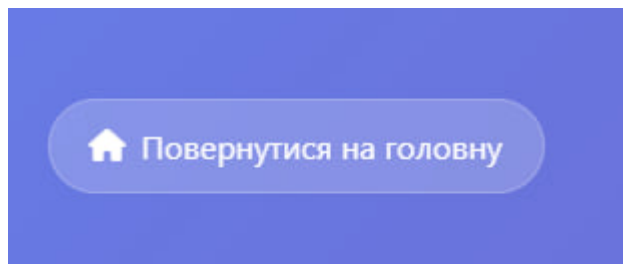


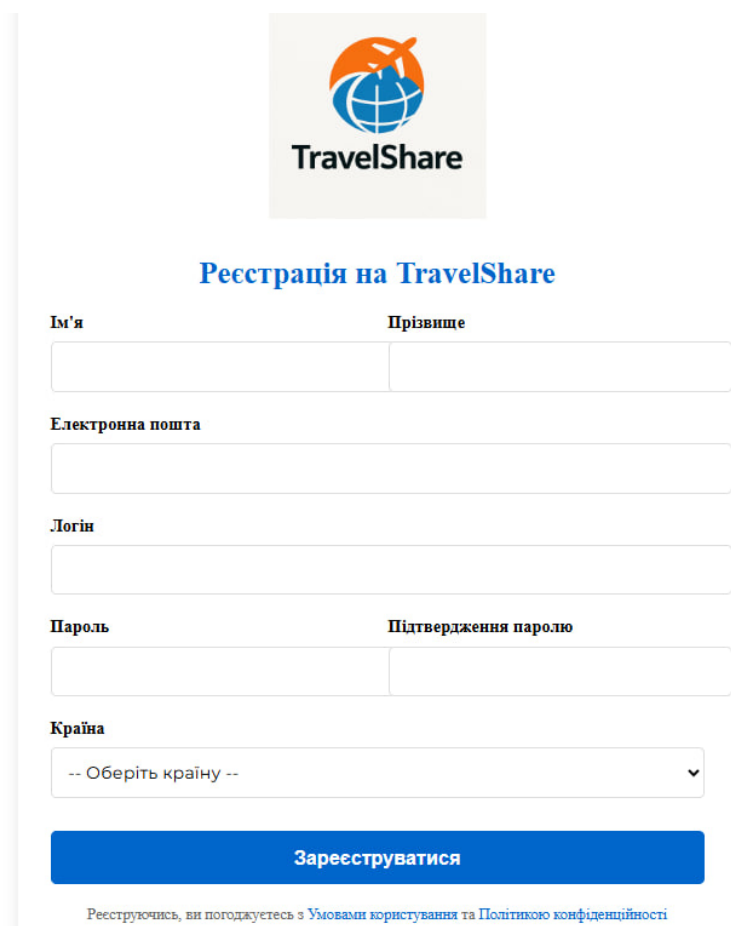
Рисунок 3.6 – Загальний вигляд кнопки

Таким чином, за допомогою HTML та CSS показано повноцінну верстку та стилізацію одного з елементів особистого кабінету користувача. На основі таких елементів і будується структура веб-додатку з можливістю подальшого розширення функціоналу та інтеграції JavaScript для динамічної взаємодії.

3.3 Тестування функціоналу

Тестування функціоналу туристичного веб-додатку TravelShare проводилось комплексно з перевіркою всіх оновлених компонентів системи та їх взаємодії між собою. Для забезпечення якості додатку було застосовано поетапний підхід до тестування, що включав перевірку користувацького інтерфейсу, функціональні системи, взаємодії між компонентами та загальної продуктивності платформи.

Одним з ключових аспектів тестування була перевірка системи реєстрації нових користувачів. Форма реєстрації включає всі необхідні поля: ім'я, прізвище, електронну пошту, логін, пароль та підтвердження пароля, а також вибір країни з випадаючого списку. Під час тестування було перевірено валідацію всіх полів, правильність обробки помилок при введенні некоректних даних та функціонування посилань на умови користування і політику конфіденційності.




TravelShare

Реєстрація на TravelShare

Ім'я **Прізвище**

Електронна пошта

Логін

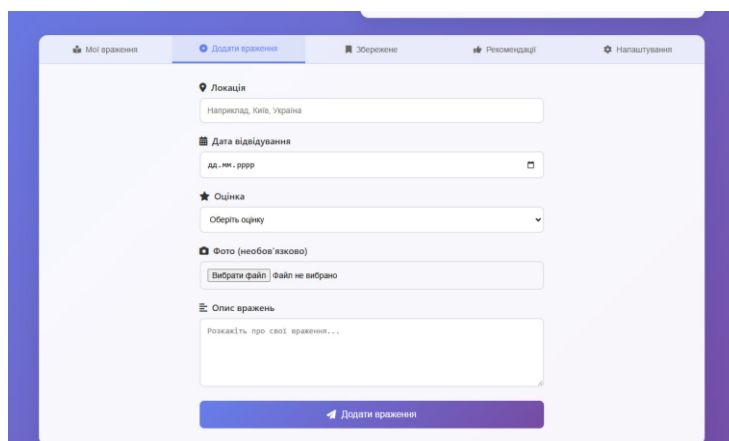
Пароль **Підтвердження паролю**

Країна

Реєструючись, ви погоджуєтесь з [Умовами користування](#) та [Політикою конфіденційності](#)

Рисунок 3.7 – Форма реєстрації «TravelShare»

Форма додавання нових вражень містить всі необхідні поля для створення повноцінного контенту. Під час тестування було перевірено функціонування завантаження зображень, валідацію дат, роботу системи оцінювання та правильність збереження введення даних.



Мої враження | **Додати враження** | Збережене | Рекомендації | Налаштування

Локація

Дата відвідування

Оцінка

Фото (необов'язково)
 файл не вибрано

Опис вражень

Рисунок 3.8 – Форма додавання вражень «TravelShare»

Окрема сторінка з враженнями демонструє ефективне систему відгуків за різними критеріями, включаючи фільтри за локацією, рейтингом, сортуванням та типами подорожей. Кожен фідбек має прев'ю-зображення, назву локації, рейтинг у вигляді зірок, короткий опис та кнопки для детального перегляду.

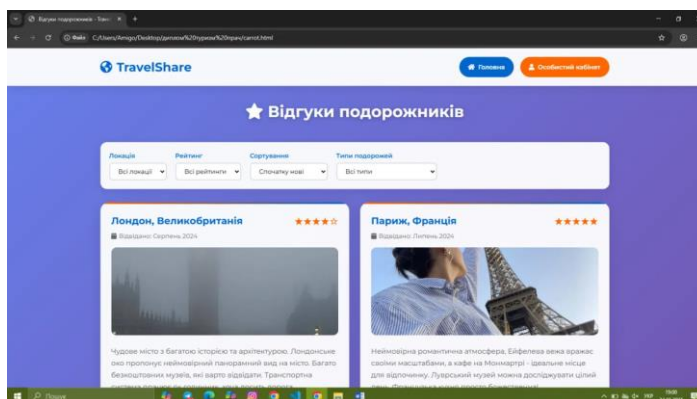


Рисунок 3.9 – Сторінка з враженнями «TravelShare»

Веб-додаток TravelShare протестовано у різних браузерах: інтерфейс стабільно працює, має фірмовий дизайн із градієнтом, логотипом та зручною навігацією. Усі функції працюють коректно, забезпечуючи швидкий відгук, інтуїтивне управління та надійну валідацію форм.

3.4 Локальне розгортання веб-додатка та запуск через локальний сервер

Даний пункт є важливим етапом розробки, що дозволяє протестувати функціональність системи перед розміщенням на продуктивному сервері.

Для локального розгортання було налаштовано середовище розробки з необхідними компонентами: веб-сервер, інтерпретатор мови програмування, систему управління базами даних та сучасний веб-браузер для тестування. Відповідно попрацювали з шляхами маршрутизації, обробкою статистичних файлів та правилами безпеки.

Локальна база даних створена та наповнена текстовими даними, включаючи таблиці для користувачів, туристичних вражень, коментарів та метаданих зображень. Запуск додатку здійснюється через localhost з перевіркою з'єднання з базою даних та ініціалізацією основних модулів.

Після запуску проведено комплексне тестування всіх функцій додатку в локальному середовищі. Виявлені помилки були виправлені, оптимізовано запити до бази даних та налаштовано кешування для покращення продуктивності.

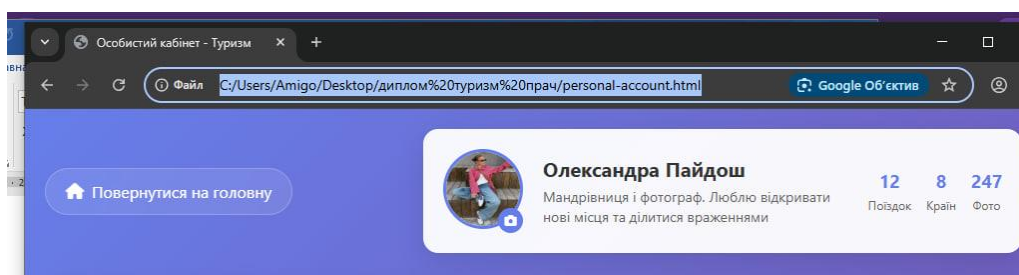


Рисунок 3.10 – Локальне розгортання веб-додатку

Локальне середовище дозволило провести аналіз продуктивності, включаючи вимірювання часу завантаження сторінок та використання ресурсів. Успішне розгортання підтвердило готовність додатку до переносу на продуктивний сервер.

3.5 Інструкція користувача сайту

Розглянемо по кроках, як користуватись сайтом та які блоки взаємодії в ньому є. Після переходу на сайт відображається головна сторінка, описана детальніше в попередніх пунктах. Безпосередньо, наш веб-додаток відкриває більше можливостей для зареєстрованих користувачів. Тому нижче можна побачити форму для входу або реєстрування для нових користувачів.

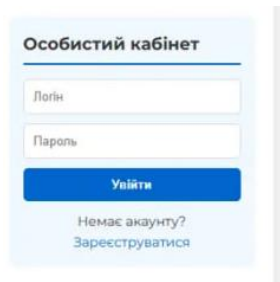


Рисунок 3.11 – Вхід до кабінету веб-додатку

Особливості особистого кабінету полягають в персоналізованих рекомендаціях, можливості зберігати вподобаний контент, додавати власний досвід з подорожей та висвітлювати їх, а також редагуванню інформації про себе. Для зручності передбачена функція «Повернутись на головну».

Окрему увагу слід приділити вкладці з загальними враженнями, де можна здійснювати пошук бажанок за вибраними категоріями, серед яких локація, рейтинг, тип подорожі та сортування. Додатково реалізовано можливість збереження результатів пошуку для подальшого перегляду.

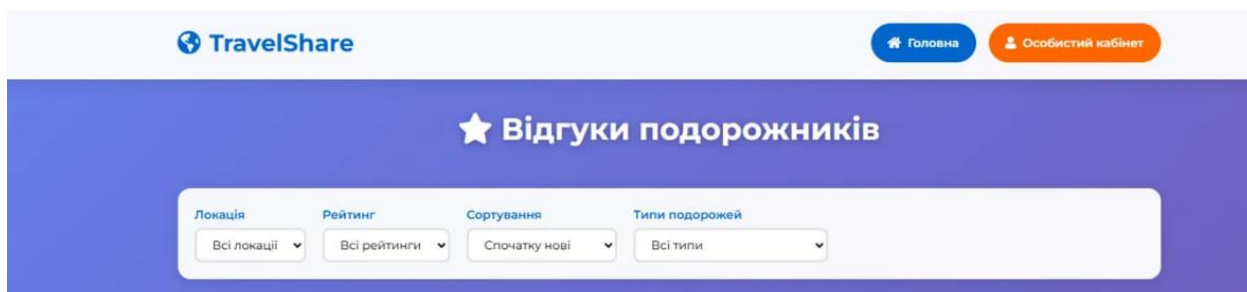


Рисунок 3.12 – Header вкладки з всіма відгуками

Дизайн виконано в сучасному стилі з урахуванням принципів UX/UI, що забезпечує комфортне використання як на десктопних пристроях, так і на мобільних. Адаптивна верстка гарантує коректне відображення контенту на різних розмірах екранів, а швидкість завантаження сторінок оптимізована для покращення користувацького досвіду.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Умови праці під час створення програмного забезпечення

Робота програміста відноситься до категорії інтелектуальної праці, що характеризується тривалим перебуванням у статичному положенні, підвищеним навантаженням на зоровий аналізатор та нервову систему. Особливу увагу необхідно приділити організації робочого місця та створенню оптимальних умов для продуктивної діяльності.

Робоче місце програміста повинно відповідати ергономічним вимогам та забезпечувати комфортне виконання трудових функцій. Правильна організація робочого простору включає:

- Оптимальне розміщення монітора на відстані 50-70 см від очей;
- Розташування клавіатури та миші на одному рівні з ліктями;
- Використання стільця з регульованою висотою та підтримкою спини.

Мікроклімат приміщення має підтримуватися в межах комфортних значень через використання систем вентиляції та кондиціонування. Оптимальна температура для розумової діяльності становить 20-22°C при відносній вологості 40-60%. Перевищення цих показників призводить до зниження працездатності та погіршення самопочуття.

Режим праці та відпочинку програміста потребує особливої організації через специфіку діяльності. Рекомендується проводити 10-15-хвилинні перерви кожні 45-60 хвилин роботи з комп'ютером, під час яких виконувати вправи для очей, шиї та спини. Загальна тривалість роботи з відеодисплейними терміналами не повинна перевищувати 6 годин на добу.

4.2 Електробезпека при роботі з ПК

Персональні комп'ютери та супутнє обладнання є джерелами електричної небезпеки, оскільки працюють від мережі змінного струму напругою 220 В. Забезпечення електробезпеки є критично важливим аспектом організації робочого місця програміста.

Основні принципи електробезпеки базуються на створенні умов, що виключають можливість ураження електричним струмом. Це досягається через правильне заземлення обладнання, використання справних електричних приладів, дотримання правил експлуатації та регулярне проведення профілактичних оглядів.

Заземлення є основним засобом захисту від ураження електричним струмом при роботі з комп'ютерною технікою. Опір заземлюючого пристрою не повинен перевищувати 4 Ом для мереж напругою 380/220 В. Всі металеві корпуси обладнання підлягають обов'язковому заземленню через спеціальний провідник.

Захисне відключення (УЗО) забезпечує автоматичне вимкнення електроустановки при виникненні струму витoku на землю. Для офісних приміщень рекомендується використання УЗО з номінальним струмом спрацьовування 30 мА, що гарантує безпеку персоналу при роботі з електрообладнанням.

Профілактичні заходи включають регулярну перевірку справності електропроводки, розеток, вилок та з'єднувальних кабелів. Забороняється експлуатація обладнання з пошкодженою ізоляцією, самостійний ремонт електричних приладів неспеціалістами, використання несправних подовжувачів та перевантаження електричної мережі.

4.3 Дії у разі надзвичайних ситуацій

Надзвичайні ситуації в офісних приміщеннях можуть виникати внаслідок пожеж, аварій інженерних систем, стихійних лих або техногенних катастроф. Правильні дії персоналу в перші хвилини після виникнення надзвичайної ситуації часто визначають масштаби наслідків та кількість постраждалих.

Пожежна безпека в приміщеннях з комп'ютерною технікою потребує особливої уваги через наявність легкозаймистих матеріалів та можливість виникнення коротких замикань. Основними причинами пожеж є порушення правил експлуатації електрообладнання, перевантаження електричної мережі, несправність систем вентиляції та недотримання правил зберігання паперових документів.

В таблиці 4.1 описано класифікацію надзвичайних ситуацій та дій при їх виникненні.

Таблиця 4.1 – Класифікація НС

Тип НС	Ознаки виникнення	Перші дії	Засоби захисту	Час евакуації
Пожежа	Дим, запах гару, полум'я	Сигнал тривоги, відключення електроенергії	Вогнегасники, пожежні крани	3-5 хвилин
Землетрус	Коливання будівлі, тріск	Укриття під столом, вихід	Міцні конструкції	1-2 хвилини
Повінь	Підйом води, затоплення	Підйом на верхні поверхи	Рятувальні засоби	10-15 хвилин
Хімічне забруднення	Незвичайний запах, подразнення	Закриття вікон, евакуація	Засоби індивідуального захисту	5-10 хвилин

План евакуації повинен бути розроблений для кожного приміщення з урахуванням його планування та можливих шляхів виходу. Евакуація проводиться організовано, без паніки, під керівництвом відповідальної особи. Пріоритет надається допомозі людям з обмеженими можливостями, жінкам та дітям.

Система оповіщення про надзвичайні ситуації включає звукові та світлові сигнали, що дозволяють своєчасно інформувати персонал про небезпеку. Кожен працівник повинен знати розташування найближчих виходів, засобів пожежогасіння та аптечок першої допомоги.

Першу допомогу при ураженні електричним струмом необхідно надавати негайно після припинення дії струму на постраждалого. Основні заходи включають штучну вентиляцію легень, непрямий масаж серця, обробку опіків та викликання швидкої медичної допомоги. До прибуття медиків постраждалому забезпечується спокій та тепло.

Регулярне проведення навчань з евакуації та надання першої допомоги підвищує готовність персоналу до дій в надзвичайних ситуаціях. Такі заходи повинні проводитися не рідше двох разів на рік з обов'язковим залученням всіх працівників та аналізом виявлених недоліків.

ВИСНОВКИ

У результаті виконання дипломної роботи було успішно розроблено веб-додаток для обміну та збору туристичними враженнями, який забезпечує комплексне вирішення задач взаємодії між мандрівниками та створення спільноти для обміну досвідом подорожей.

В ході дослідження було проведено детальний аналіз існуючих туристичних платформ, включаючи TripAdvisor, Lonely Planet та TourRadar, що дозволило виявити їх переваги та недоліки. На основі цього аналізу було сформульовано технічне завдання проекту, яке враховує сучасні потреби користувачів у сфері туризму та забезпечує конкурентоспроможність розробленого рішення.

Архітектурне проектування веб-додатку базувалося на принципах користувацько-орієнтованого дизайну та сучасних веб-стандартах. Розроблена структура додатку забезпечує інтуїтивну навігацію та ефективну організацію контенту. Моделювання використання веб-додатку підтвердило доцільність обраних архітектурних рішень.

Для реалізації проекту було обрано оптимальний технологічний стек, що включає HTML для семантичної розмітки, CSS для стилізації дизайну, JavaScript для реалізації інтерактивної функціональності та LocalStorage для локального збереження даних користувачів. Використання Visual Studio Code як середовища розробки забезпечило високу продуктивність процесу програмування.

Програмна реалізація проекту включала розробку функціональних модулів для додавання та перегляду туристичних вражень, системи рейтингування, пошуку за категоріями та географічними локаціями. Дизайн інтерфейсу користувача реалізовано з урахуванням принципів юзабіліті та сучасних тенденцій веб-дизайну, що забезпечує привабливий зовнішній вигляд та зручність використання.

Комплексне тестування функціоналу підтвердило коректність роботи всіх модулів системи та відповідність реалізованих функцій технічним вимогам. Локальне розгортання веб-додатку було успішно виконано з налаштуванням локального сервера для тестування та демонстрації результатів роботи.

Розроблена інструкція користувача забезпечує простоту освоєння функціоналу додатку новими користувачами та містить детальні пояснення всіх можливостей системи. Це сприяє підвищенню користувацького досвіду та швидкому залученню аудиторії до використання платформи.

Окремий розділ роботи присвячено питанням охорони праці та безпеки в надзвичайних ситуаціях при розробці програмного забезпечення. Розглянуто оптимальні умови праці програміста, заходи електробезпеки при роботі з комп'ютерною технікою та алгоритми дій у випадку виникнення надзвичайних ситуацій, що забезпечує безпечні умови трудової діяльності.

Створений веб-додаток успішно вирішує поставлені задачі щодо забезпечення платформи для обміну туристичними враженнями, має потенціал для подальшого розвитку та масштабування. Функціональні можливості додатку відповідають сучасним вимогам туристичної індустрії та потребам цільової аудиторії.

Перспективи подальшого розвитку проекту включають інтеграцію з зовнішніми API туристичних сервісів, розширення функціоналу соціальної взаємодії користувачів та реалізацію мобільної версії додатку.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи візуалізації туристичних маршрутів у веб-застосунках / М. О. Поліщук, В. А. Ткаченко, Л. І. Бондаренко, Р. С. Федоров. Інформаційні системи та мережі. 2024. № 914. С. 23-31.
2. Haverbeke M. Eloquent JavaScript: A Modern Introduction to Programming. 3rd ed. San Francisco: No Starch Press, 2018. 472 p.
3. Розробка чат-ботів для туристичних веб-платформ засобами Node.js / К. В. Іванов, Н. О. Миронова, А. С. Петренко, Д. М. Волков. Системи підтримки прийняття рішень. Теорія і практика. 2023. № 6. С. 92-99.
4. Simpson K. You Don't Know JS: Scope & Closures. 2nd ed. Sebastopol: O'Reilly Media, 2020. 168 p. Visual Studio Code Documentation. Microsoft. [Електронний ресурс]. URL: <https://code.visualstudio.com/docs> .
5. Курінний С. Розробка веб-сайтів для початківців: HTML – CSS – JavaScript. Київ: Видавничий дім "Києво-Могилянська академія", 2023. 328 с.
6. Іваненко О.В. Сучасні тенденції розробки туристичних веб-платформ. // Інформаційні технології в туризмі. 2024. № 3. С. 45-52.
7. Дакетт Дж. HTML і CSS. Розробка та дизайн веб-сайтів. Переклад з англ. Київ: Наш формат, 2022. 512 с.
8. Петренко М.І., Коваленко А.С. Використання JavaScript для створення інтерактивних туристичних додатків. // Вісник Київського університету. Серія: Інформаційні технології. 2023. № 12. С. 78-85.
9. Основи веб-розробки (HTML, CSS, JavaScript). EdEra. [Електронний ресурс]. URL: <https://study.ed-era.com/courses/course/5159>.
10. Сидоренко В.П. Організація локального збереження даних у веб-додатках засобами LocalStorage. // Комп'ютерні науки та інформаційні технології. 2024. № 5. С. 123-130.

11. LocalStorage, sessionStorage. JavaScript.info. [Електронний ресурс]. URL: <https://uk.javascript.info/localstorage> .
12. Бойко Р.М. Адаптивний веб-дизайн: методи та засоби реалізації. // Інформатика та обчислювальна техніка. 2024. № 2. С. 34-41.
13. HTML Підручник. W3Schools українською. [Електронний ресурс]. URL: <https://w3schoolsua.github.io/html/index.html>.
14. CSS Specifications. W3C. [Електронний ресурс]. URL: <https://www.w3.org/Style/CSS/specs.en.html>.
15. TripAdvisor. Офіційний сайт. [Електронний ресурс]. URL: <https://www.tripadvisor.com/> (дата звернення: 01.05.2025).
16. Lonely Planet. Офіційний сайт. [Електронний ресурс]. URL: <https://www.lonelyplanet.com/>.
17. TourRadar. Офіційний сайт. [Електронний ресурс]. URL: <https://www.tourradar.com/>.
18. Visual Studio Code Documentation. Microsoft. [Електронний ресурс]. URL: <https://code.visualstudio.com/docs>.
19. Архітектура веб-додатків для обміну геоінформаційними даними в туризмі / Т. А. Мельник, В. І. Гриценко, О. П. Сердюк, А. М. Kovalenko. Системи обробки інформації. 2024. № 1. С. 67-74.
20. Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol: O'Reilly Media, 2020. 1096 p.
21. Карпенко О. С., Василенко Н. М. Методи захисту персональних даних у туристичних веб-платформах. Кібербезпека: освіта, наука, техніка. 2023. № 4. С. 112-119.
22. Brown E., Simpson K. Modern Web Development: Building Scalable Applications. New York: Addison-Wesley Professional, 2023. 448 p.

- 23.Інтеграція API-сервісів у туристичні веб-додатки: практичні аспекти / Д. П. Романенко, Л. М. Шевченко, М. В. Петров, К. О. Іваненко. Вісник НТУ "ХПІ". Серія: Системний аналіз, управління та інформаційні технології. 2024. № 1. С. 89-96.
- 24.Гавриш О. А., Стеценко І. В. Проектування інтерфейсів користувача для туристичних мобільних додатків. Вісник Харківського національного університету імені В. Н. Каразіна. Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління. 2023. № 60. С. 78-86.
- 25.Freeman E. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. 2nd ed. Sebastopol: O'Reilly Media, 2021. 672 p.
- 26.Оптимізація продуктивності веб-додатків для туристичної індустрії / В. М. Коваль, Т. С. Левченко, О. Р. Мазур, Н. В. Савченко. Вісник Київського політехнічного інституту. Серія: Інформатика, управління та обчислювальна техніка. 2024. № 80. С. 45-53.
- 27.Resig J., Bibeault B., Maras J. Secrets of the JavaScript Ninja. 2nd ed. Greenwich: Manning Publications, 2016. 464 p.
- 28.Система управління контентом для туристичних порталів / А. В. Семенов, О. М. Кравець, І. П. Дмитренко, С. О. Гончар. Радіoeлектронні і комп'ютерні системи. 2023. № 4. С. 156-163.
- 29.Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Upper Saddle River: Prentice Hall, 2008. 464 p.
- 30.Nixon R. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites. 6th ed. Sebastopol: O'Reilly Media, 2021. 832 p.

ДОДАТКИ

Додаток А. Фрагмент коду головної сторінки

Travel.html

```
<!DOCTYPE html>
<html lang="uk">
  <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">
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="travel.css" />
    <title>TravelShare - Поділіться враженнями</title>
  </head>
  <body>
    <div class="wrapper">
      <div class="main-content">
        <h2 class="my-profession">Туристичні враження з усього світу</h2>
        <p>Поділіться своїми враженнями від подорожей та відкрийте нові місця через досвід інших мандрівників.</p>
        <p>Шукаєте ідеальне місце для відпочинку? Наша платформа допоможе вам знайти цікаві напрямки та отримати корисні поради від інших туристів. Ви можете дізнатися про визначні пам'ятки, місцеву кухню, культуру та звичаї різних країн.</p>
        <a href="carrot.html" class="go-to-experiences-button">Перейти до вражень</a>
        <div class="experiences-container">
          <div class="content-block">
            
            <h3>Париж, Франція</h3>
            <div class="rating">★★★★★</div>
            <p>Неймовірне місто кохання! Ейфелева вежа вночі виглядає казково. Рекомендую відвідати Лувр, але плануйте заздалегідь, щоб уникнути довгих черг. Місцева випічка перевершила всі мої очікування...</p>
            <a href="experience-paris.html" class="read-more-button">Читати далі</a>
          </div>
          <div class="content-block">
            
            <h3>Барселона, Іспанія</h3>
            <div class="rating">★★★★☆</div>
            <p>Барселона – місто з неповторною архітектурою Гауді. Саграда Фамілія вражає своєю величчю та деталями. Пляжі чисті, але переповнені туристами влітку. Тапас-бари пропонують широкий вибір традиційних...</p>
            <a href="experience-barcelona.html" class="read-more-button">Читати далі</a>
          </div>
          <div class="content-block">
            
            <h3>Киото, Японія</h3>
            <div class="rating">★★★★★</div>
```


<p>Кіото – культурна столиця Японії, де традиції зустрічаються з сучасністю. Храми та сади створюють атмосферу спокою та гармонії. Особливо вразив Золотий павільйон та бамбуковий ліс Арашіяма...</p>

Читати далі
</div>

<div class="content-block">

<h3>Рим, Італія</h3>

<div class="rating">★★★★☆</div>

<p>Вічне місто з багатою історією та культурою. Колізей вражає своїми масштабами, а римські руїни переносять вас у минуле. Ватикан обов'язковий для відвідування. Справжня італійська піца та паста - це...</p>

Читати далі

</div>

</div>

</div>

<aside class="sidebar">

<div class="tech-skills-container">

<h3 class="sidebar-title">Популярні напрямки</h3>

<ul class="country-list">

<li class="country-item">Італія

<li class="country-item">Франція

<li class="country-item">Японія

<li class="country-item">Греція

<li class="country-item">Таїланд

<li class="country-item">США

</div>

<div class="tech-skills-container">

<h3 class="sidebar-title">Типи подорожей</h3>

<ul class="country-list">

<li class="country-item">Пляжний відпочинок

<li class="country-item">Культурний туризм

<li class="country-item">Пригодницький туризм

<li class="country-item">Гастрономічний туризм

<li class="country-item">Екотуризм

```

    </ul>
  </div>
  <div class="login-container">
    <h3 class="sidebar-title">Особистий кабінет</h3>
    <!-- ==== ФОРМА ЛОГІНУ ==== -->
    <form id="loginForm" action="/login" method="post" class="login-form">
      <input type="text" name="username" placeholder="Логін" required>
      <input type="password" name="password" placeholder="Пароль" required>
      <button type="submit" class="login-button">Увійти</button>
    </form>
    <script>
      document.getElementById('loginForm').addEventListener('submit', async
function (e) {
      e.preventDefault(); // блокуємо стандартне відправлення
      // Збираємо дані форми
      const data = new FormData(e.target);
      // TODO: реальний запит на сервер
      // const resp = await fetch('/login', { method:'POST', body:data });
      // if (resp.ok) { ... }
      /* Демонстраційно вважаємо, що логін успішний */
      const ok = true;
      if (ok) {
        // ← сторінка особистого кабінету
        window.location.href = 'personal-account.html';
      } else {
        alert('Невірний логін або пароль');
      }
    });
    </script>
    <p class="register-link">Немає акаунту? <a
href="register.html">Зареєструватися</a></p>
  </div>
</aside>
</div>
<script> // Показати/сховати форму додавання досвіду
document.getElementById('showFormBtn').addEventListener('click', function() {
document.getElementById('experienceForm').style.display = 'block';
});
document.getElementById('cancelBtn').addEventListener('click', function() {
document.getElementById('experienceForm').style.display = 'none';
});
</script>
</body>
</html>

```

Travel.css

```

/* Загальні стилі */
* {
  margin: 0; padding: 0; box-sizing: border-box;
}
body {
  font-family: 'Montserrat', sans-serif; background-color: #f5f5f5; color: #333;
  line-height: 1.6;
}
.wrapper {
  max-width: 1200px; margin: 0 auto; display: flex; padding: 20px; gap: 30px;
}
/* Головний контент */
.main-content {
  flex: 1; background-color: #fff; padding: 25px; border-radius: 10px; box-
  shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
}
.my-profession {
  color: #0066cc; font-size: 28px; margin-bottom: 15px;
}
.main-content p {
  margin-bottom: 20px; font-size: 16px;
}
/* Поле пошуку */
.top-search {
  display: flex; margin-bottom: 25px;
}
.search-input {
  flex: 1; padding: 12px 15px; border: 1px solid #ddd; border-radius: 5px 0 0 5px;
  font-size: 16px; outline: none;
}
.search-button {
  padding: 12px 20px; background-color: #0066cc; color: white; border: none;
  border-radius: 0 5px 5px 0; cursor: pointer; font-weight: 600; transition:
  background-color 0.3s;
}
.search-button:hover {
  background-color: #0055aa;
}
.go-to-experiences-button {
  display: inline-block; margin: 20px 0; padding: 10px 20px; background-color:
  #3498db; color: white; text-decoration: none; border-radius: 8px; font-weight: bold;
}
.go-to-experiences-button:hover {
  background-color: #2980b9;
}
/* Кнопка додати враження */

```

```

.add-experience {
  margin-bottom: 30px;
}

.add-button {
  padding: 12px 24px; background-color: #ff6600; color: white; border: none;
border-radius: 5px; cursor: pointer; font-weight: 600; transition: all 0.3s; font-
size: 16px;
}

.add-button:hover {
  background-color: #e65c00; transform: translateY(-2px); box-shadow: 0 4px 8px
rgba(0, 0, 0, 0.1);
}
/* Форма додавання вражень */
.experience-form {
  background-color: #f9f9f9; padding: 20px; border-radius: 8px; margin-bottom:
30px; border: 1px solid #eee; display: none;
}

.experience-form h3 {
  color: #333; margin-bottom: 15px;
}

.form-group {
  margin-bottom: 15px;
}

.form-group label {
  display: block; margin-bottom: 5px; font-weight: 600;
}

.form-group input, .form-group textarea, .form-group select {
  width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 5px;
font-family: 'Montserrat', sans-serif;
}

.form-group textarea {
  resize: vertical;
}

.submit-button, .cancel-button {
  padding: 10px 18px; border: none; border-radius: 5px; cursor: pointer;
font-weight: 600; margin-right: 10px;
}

.submit-button {
  background-color: #4CAF50; color: white;
}

.submit-button:hover {
  background-color: #45a049;
}

.cancel-button {
  background-color: #f44336; color: white;
}

.cancel-button:hover {
  background-color: #d32f2f;
}

```

```

    }

/* Контейнер вражень */
.experiences-container {
    display: grid; grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
    gap: 25px;
}
.content-block {
    background-color: #fff; border-radius: 8px; overflow: hidden; box-shadow: 0 4px
8px rgba(0, 0, 0, 0.1); transition: transform 0.3s, box-shadow 0.3s;
}
.content-block:hover {
    transform: translateY(-5px); box-shadow: 0 8px 16px rgba(0, 0, 0, 0.15);
}
.content-block img {
    width: 100%; height: 180px; object-fit: cover;
}
.content-block h3 {
    padding: 15px 15px 5px; color: #0066cc; font-size: 18px;
}
.rating {
    color: #ff9900; padding: 0 15px 10px; font-size: 18px;
}
.content-block p {
    padding: 0 15px 15px; font-size: 14px;
}
.read-more-button {
    display: block; background-color: #0066cc; color: white; text-align: center;
padding: 10px; text-decoration: none; font-weight: 600; transition: background-color
0.3s;
}
.read-more-button:hover {
    background-color: #0055aa;
}
/* Бічна панель */
.sidebar {
    width: 300px; padding: 20px; background-color: #fff; border-radius: 10px;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
}
.my-photo {
    width: 100%; border-radius: 8px; margin-bottom: 20px;
}
.sidebar-title {
    font-size: 18px; color: #333; margin-bottom: 15px; padding-bottom: 5px; border-
bottom: 2px solid #0066cc;
}
.tech-skills-container {
    margin-bottom: 25px;
}

```

```

    }
.country-list {
    list-style: none;
}
.country-item {
    margin-bottom: 8px;
}
.country-item a {
    display: block; padding: 8px 10px; background-color: #f0f8ff; border-radius:
5px; color: #333; text-decoration: none; transition: all 0.2s;
}
.country-item a:hover {
    background-color: #0066cc; color: white; transform: translateX(5px);
}
/* Форма входу */
.login-container {
    background-color: #f0f8ff; padding: 15px; border-radius: 8px;
}
.login-form input {
    width: 100%; padding: 10px; margin-bottom: 10px; border: 1px solid #ddd;
border-radius: 5px;
}
.login-button {
    width: 100%; padding: 10px; background-color: #0066cc; color: white; border:
none; border-radius: 5px; cursor: pointer; font-weight: 600; margin-bottom: 10px;
transition: background-color 0.3s;
}
.login-button:hover {
    background-color: #0055aa;
}
.register-link {
    text-align: center; font-size: 14px;
}
.register-link a {
    color: #0066cc; text-decoration: none;
}
.register-link a:hover {
    text-decoration: underline;
}
/* Адаптивність */
@media (max-width: 900px) {
.wrapper {
    flex-direction: column;
}
.sidebar {
    width: 100%; order: -1;
}
.experiences-container {

```

```
        grid-template-columns: repeat(auto-fill, minmax(240px, 1fr));
    }
}
@media (max-width: 600px) {
    .experiences-container {
        grid-template-columns: 1fr;
    }
    .top-search {
        flex-direction: column;
    }
    .search-input {
        border-radius: 5px; margin-bottom: 10px;
    }
    .search-button {
        border-radius: 5px;
    }
    .wrapper {
        padding: 10px;
    }
    .main-content, .sidebar {
        padding: 15px;
    }
}
```

Додаток Б. Фрагменти коду JavaScript сторінки реєстрації та входу

Register. html

```
<script>
document.getElementById('registerForm').addEventListener('submit', function(event) {
    event.preventDefault(); // блокуємо стандартну відправку форми
    let isValid = true; // Перевірка імені
    const firstName = document.getElementById('firstName');
    if (firstName.value.trim() === '') {
        document.getElementById('firstNameError').style.display = 'block';
        isValid = false;
    } else {
        document.getElementById('firstNameError').style.display = 'none';
        } // Перевірка прізвища
    const lastName = document.getElementById('lastName');
    if (lastName.value.trim() === '') {
        document.getElementById('lastNameError').style.display = 'block';
        isValid = false;
    } else {
        document.getElementById('lastNameError').style.display = 'none';
        } // Перевірка email
    const email = document.getElementById('email');
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailRegex.test(email.value)) {
        document.getElementById('emailError').style.display = 'block';
        isValid = false;
    } else {
        document.getElementById('emailError').style.display = 'none';
        } // Перевірка логіну
    const username = document.getElementById('username');
    if (username.value.length < 4 || username.value.length > 16) {
        document.getElementById('usernameError').style.display = 'block';
        isValid = false;
    } else {
        document.getElementById('usernameError').style.display = 'none';
        } // Перевірка паролю
    const password = document.getElementById('password');
    if (password.value.length < 8) {
        document.getElementById('passwordError').style.display = 'block';
        isValid = false;
    } else {
        document.getElementById('passwordError').style.display = 'none';
        } // Перевірка підтвердження паролю
    const confirmPassword = document.getElementById('confirmPassword');
    if (password.value !== confirmPassword.value) {
        document.getElementById('confirmPasswordError').style.display = 'block';
        isValid = false;
    } else {
```



```

document.getElementById('confirmPasswordError').style.display = 'none';
    } // Перевірка країни
const country = document.getElementById('country');
if (country.value === '') {
document.getElementById('countryError').style.display = 'block';
isValid = false;
} else {
document.getElementById('countryError').style.display = 'none';
    }
    if (isValid) {
alert('Реєстрація успішна! Ласкаво просимо до TravelShare!');
window.location.href = 'personal-account.html'; // редирект після успішної
реєстрації
    }
});
</script>

```

Enter.html

```

<script>
    document.getElementById('loginForm').addEventListener('submit',
function(event) {
    event.preventDefault(); // блокуємо стандартне надсилання форми
    let isValid = true;
    const username = document.getElementById('username');
    const password = document.getElementById('password');
    if (username.value.trim() === '') {
document.getElementById('usernameError').style.display = 'block';
        isValid = false;
    } else {
document.getElementById('usernameError').style.display = 'none';
    }

    if (password.value.trim() === '') {
document.getElementById('passwordError').style.display = 'block';
        isValid = false;
    } else {
document.getElementById('passwordError').style.display = 'none';
    }

    if (isValid) {
        alert('Вхід успішний! Ласкаво просимо!');
        // Переадресація на сторінку особистого кабінету
        window.location.href = 'personal-account.html';
    }
});
</script>

```

Додаток В. Фрагменти коду особистого кабінету

Html

```
<!DOCTYPE html>

<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Особистий кабінет - Туризм</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
</head>

<body>

  <div class="wrapper">
    <!-- Заголовок з кнопкою повернення і профілем користувача -->
    <div class="header-container">
      <!-- Кнопка повернення на головну сторінку -->
      <a href="travel.html" class="home-button">
        <i class="fas fa-home"></i> Повернутися на головну
      </a>
      <!-- Профіль користувача -->
      <div class="user-profile">
        <div class="avatar-container">
          
          <button class="avatar-edit-btn" id="edit-avatar-btn">
            <i class="fas fa-camera"></i>
          </button>
        </div>
        <div class="user-info">
          <div class="user-name" id="user-name">Олександра Пайдош</div>
          <div class="user-description" id="user-description">Мандрівниця і
фотограф. Люблю відкривати нові місця та ділитися враженнями</div>
        </div>
      <!-- Система табуляції -->
    <div class="tab-container">
      <div class="tabs">
        <button class="tab active" data-tab="my-experiences">
          <i class="fas fa-book-reader"></i> Мої враження
        </button>
        <button class="tab" data-tab="add-experience">
          <i class="fas fa-plus-circle"></i> Додати враження
        </button>
        <button class="tab" data-tab="saved">
          <i class="fas fa-bookmark"></i> Збережене
        </button>
        <button class="tab" data-tab="recommendations">
          <i class="fas fa-thumbs-up"></i> Рекомендації
```

```

        </button>
        <button class="tab" data-tab="settings">
            <i class="fas fa-cog"></i> Налаштування
        </button>
    </div>

```

Css

```
/* === Базові стилі та градієнтний фон === */
```

```

    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; background: linear-
gradient(135deg, #667eea 0%, #764ba2 100%); min-height: 100vh;
    }

```

```
/* === Профіль користувача з ефектом скла === */
```

```

    .user-profile {
        display: flex; align-items: center; gap: 20px; background: rgba(255, 255, 255,
0.95); padding: 20px; border-radius: 15px; box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
        flex: 1; max-width: 600px;
    }

```

```
/* === Анімації та переходи === */
```

```

    .content-block {
        background: white; border-radius: 12px; padding: 20px; box-shadow: 0 4px 15px
rgba(0, 0, 0, 0.1); transition: transform 0.3s ease;
    }
    .content-block:hover {
        transform: translateY(-2px);
    }

```

JavaScript

```
// Логіка перемикання вкладок
```

```

const tabs = document.querySelectorAll('.tab');
const tabContents = document.querySelectorAll('.tab-content');

```

```

tabs.forEach(tab => {
    tab.addEventListener('click', () => {
        const target = tab.getAttribute('data-tab');

```

```
// Зняти активний стан з усіх вкладок
```

```

        tabs.forEach(t => t.classList.remove('active'));
        tabContents.forEach(tc => tc.classList.remove('active'));

```

```
// Активувати обрану вкладку
```

```

        tab.classList.add('active');
        document.getElementById(target).classList.add('active');
    });

```

```
});
```

```
// Обробка форми додавання вражень
```

```

        const experienceForm = document.getElementById('experience-
form');
        const experiencesList = document.getElementById('experiences-
list');

        experienceForm.addEventListener('submit', e => {
            e.preventDefault();

            const location =
document.getElementById('location').value.trim();
            const date = document.getElementById('date-visited').value;
            const ratingValue =
parseInt(document.getElementById('rating').value);
            const description =
document.getElementById('description').value.trim();
            const photoInput = document.getElementById('photo');
            const photoFile = photoInput.files[0];
// Валідація обов'язкових полів
            if (!location || !date || isNaN(ratingValue) || !description)
            {
                alert('Будь ласка, заповніть усі поля форми.');
```

return;

```

            }

            const ratingStars = '★'.repeat(ratingValue) + '☆'.repeat(5 -
ratingValue);
            // Створення нового блоку контенту
            const newBlock = document.createElement('div');
            newBlock.classList.add('content-block');

            function insertExperience(imgSrc = '') {
                const imgTag = imgSrc ? `` : '';
                newBlock.innerHTML = `
                    ${imgTag}
                    <h3>${location}</h3>
                    <div class="rating">${ratingStars}</div>
                    <div class="actions-container">
                        <button class="edit-button">Редагувати</button>
                        <button class="delete-button">Видалити</button>
                    </div>
                    <p>${description}</p>
                    <a href="#" class="read-more-button">Читати далі</a>
                `;
            }
            experiencesList.appendChild(newBlock);
            alert('Враження успішно додано!');
            experienceForm.reset();
// Обробка зміни аватара в налаштуваннях

```

```

const avatarInput = document.getElementById('settings-avatar');
const avatarPreview = document.getElementById('avatar-preview');

avatarInput.addEventListener('change', (e) => {
  const file = e.target.files[0];
  if (file) {
    const reader = new FileReader();
    reader.onload = function(e) {
      avatarPreview.src = e.target.result;
    };
    reader.readAsDataURL(file);
  }
});

// Обробка форми налаштувань профілю
const settingsForm = document.getElementById('settings-form');
settingsForm.addEventListener('submit', (e) => {
  e.preventDefault();

  const newName = document.getElementById('settings-username').value.trim();
  const newDescription = document.getElementById('settings-description').value.trim();
  const newEmail = document.getElementById('settings-email').value.trim();
  const avatarSrc = avatarPreview.src;

  // Валідація
  if (!newName || !newDescription || !newEmail) {
    alert('Будь ласка, заповніть усі поля.');
```

return;

```

  }
});

// Оновлення профілю в шапці
document.getElementById('user-name').textContent = newName;
document.getElementById('user-description').textContent = newDescription;
document.getElementById('user-avatar').src = avatarSrc;
alert('Профіль успішно оновлено!');
```

});