Міністерство освіти і науки України

Відокремлений структурний підрозділ «Тернопільський фаховий коледж Тернопільського національного технічного університету імені Івана Пулюя»
(повне найменування вищого навчального закладу)

Відділення інформаційних технологій, менеджменту, туризму та підготовки іноземних громадян

(назва відділення)

Циклова комісія комп'ютерної інженерії
(повна назва циклової комісії)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра
(освітньо-професійного ступеня)

на тему: **Розробка вебсайту туристичної агенції "Сонячні мандри"**

Виконав: студент IV курсу, групи КІ-405п

Спеціальності 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Василь МУЛЯРЧУК
(ім'я та прізвище)

Керівник Леся ШТОКАЛО
(ім'я та прізвище)

Рецензент
(ім'я та прізвище)

Тернопіль – 2026

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
імені ІВАНА ПУЛЮЯ»

Відділення **інформаційних технологій, менеджменту, туризму та підготовки іноземних громадян**
Циклова комісія **комп'ютерної інженерії**
Освітньо-професійний ступінь **фаховий молодший бакалавр**
Освітньо-професійна програма: **Обслуговування комп'ютерних систем і мереж**
Спеціальність: **123 Комп'ютерна інженерія**
Галузь знань: **12 Інформаційні технології**

ЗАТВЕРДЖУЮ
Голова циклової комісії
комп'ютерної інженерії
_____ Андрій ЮЗЬКІВ
"30" березня 2026 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Мулярчуку Василю Володимировичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи**:** **Розробка вебсайту туристичної агенції "Сонячні мандри"**

керівник роботи Штокало Леся Ярославівна
(прізвище, ім'я, по батькові)

затверджені наказом Відокремленого структурного підрозділу «Тернопільський фаховий коледж Тернопільського національного технічного університету імені Івана Пулюя» від 27.03.2026р № 4/9-167.

2. Строк подання студентом роботи: 15 червня 2026 року.

3. Вихідні дані до роботи: методичні вказівки для виконання кваліфікаційної роботи, технічне завдання на розробку програмного забезпечення, мови програмування: HTML5, JavaScript, CSS3, PHP

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): Загальний розділ. Розробка технічного та робочого проєкту. Спеціальний розділ. Економічний розділ. Охорона праці та безпека життєдіяльності.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

— структурна схема сайту;
— діаграма класів;
— блок-схема вибору та взаємодії з пропозиціями;
— таблиця техніко-економічних показників .

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Богдана МАРТИНЮК викладач		
Охорона праці та безпека життєдіяльності	Володимир ШТОКАЛО викладач		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	01.04	
2	Збір і узагальнення інформації	08.05	
3	Написання першого розділу	15.05	
4	Розробка технічного та робочого проекту	22.05	
5	Написання спеціального розділу	28.05	
6	Розрахунок економічної частини	1.06	
7	Написання розділу охорони праці	3.06	
8	Виконання графічної частини	8.06	
9	Оформлення проєкту	10.06	
10	Погодження нормоконтролю	11.06	
11	Попередній захист роботи	12.06	
12	Захист кваліфікаційної роботи		

7. Дата видачі завдання: 31 березня 2026 року

Студент _____ Василь МУЛЯРЧУК
(підпис) (ім'я та прізвище)

Керівник роботи _____ Леся ШТОКАЛО
(підпис) ім'я та прізвище

АНОТАЦІЯ

Мулярчук В.В. Розробка вебсайту туристичної агенції «Сонячні мандри»: кваліфікаційна робота на здобуття освітньо-професійного ступеня фахового молодшого бакалавра за спеціальністю 123 Комп'ютерна інженерія. Тернопіль: ВСП «ТФК ТНТУ», 2026р. – 88 с.

Метою даної роботи є проєктування та програмна реалізація сучасного вебсайту для туристичної агенції. Актуальність теми зумовлена необхідністю автоматизації процесу взаємодії з клієнтами та розширення маркетингових можливостей підприємства в умовах цифрової трансформації галузі туризму.

В процесі виконання роботи було проведено аналіз предметної області, досліджено потреби цільової аудиторії та проаналізовано конкурентне середовище. На основі отриманих даних розроблено архітектуру сайту, що включає інформаційні сторінки. Створено готовий до експлуатації програмний продукт. Розроблено інструкції по встановленню та налаштуванню веб ресурсу. Приділено увагу економічному обгрунтуванню розробки, а також дотримання вимог охорони праці та техніки безпеки.

Кваліфікаційна робота містить графічну частину, яка виконана на 4 аркушах формату А3, пояснювальну записку яка складається з 88 аркушів, містить таблиць 9 та 13 рисунків, 7 сторінок додатків.

Ключові слова: вебсайт, туристична агенція, HTML, CSS, JavaScript, адаптивний дизайн, інтерфейс користувача, інформаційні технології в туризмі.

ANNOTATION

Mulyarchuk V.V. Development of the website of the travel agency "Solar Journeys": qualification work for obtaining the educational and professional degree of a professional junior bachelor in the specialty 123 Computer Engineering.Ternopil:VSP "TFK TNTU", 2026. – 88 p.

The purpose of this work is to design and programmatically implement a modern website for a travel agency. The relevance of the topic is due to the need to automate the process of interaction with customers and expand the marketing capabilities of the enterprise in the context of digital transformation of the tourism industry.

In the process of performing the work, an analysis of the subject area was conducted, the needs of the target audience were studied, and the competitive environment was analyzed. Based on the data obtained, the site architecture was developed, which includes information pages. A ready-to-use software product was created. Instructions for installing and configuring the web resource were developed. Attention was paid to the economic justification of the development, as well as compliance with labor protection and safety requirements.

The qualification work contains a graphic part A3, which is made on 4 sheets of format, and an explanatory note consisting of 88 sheets, containing 9 tables and 13 figures, 7 pages of applications.

.Keywords: website, travel agency, HTML, CSS, JavaScript, adaptive design, user interface, information technologies in tourism.

ЗМІСТ

Вступ……7
1 Загальний розділ……8
1.1 Аналітичний огляд існуючих рішень……...8
1.2 Технічне завдання…….....11
1.2.1 Найменування та область застосування……....11
1.2.2 Призначення розробки…….....11
1.2.3 Вимоги до програмного забезпечення……...12
1.2.4 Вимоги до програмної документації…….....13
1.2.5 Техніко-економічні показники…….....14
1.2.6 Стадії та етапи розробки…….....16
1.2.7 Порядок контролю та прийому…….....18
2 Розробка технічного та робочого проекту…….....20
2.1 Постановка задачі……....20
2.2 Опис та обгрунтування вибору структури та методу організації вхідних та вихідних даних…….....21
2.3 Розробка алгоритму……23
2.4 Визначення інформаційних зв’язків……..26
2.5 Написання текстів програми……...28
2.6 Тестування та налагодження веб сайту……39
3 Спеціальний розділ……..43
3.1 Інструкція по встановленню та налаштуванню веб ресурсу……...43
3.2 Інструкція з використання тестових наборів……46
3.3 Інструкція з експлуатації веб сайту……50

4 Економічний розділ…….55
4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР……...55
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи…….56
4.3 Розрахунок матеріальних витрат……58
4.4 Розрахунок витрат на електроенергію……...59
4.5 Розрахунок суми амортизаційних відрахувань……61
4.6 Обчислення накладних витрат……61
4.7 Складання кошторису витрат та визначення собівартості НДР……...62
4.8 Розрахунок ціни НДР……...62
4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень ……...63
5 Охорона праці, техніка безпеки та екологічні вимоги……64
5.1 Вимоги безпеки до відеодисплейних терміналів та профілактика порушень зору……..…...64
5.2 Безпечність застосування інформаційного обладнання……..66
5.3 Забезпечення нормованого рівня освітленості робочого місця програміста
Висновки……....74
Перелік посилань……..76
Додатки…….....78
Додаток А. Лістінг Головної сторінки сайту ……78
Додаток В. Лістінг сторінки Напрямки……..80
Додаток С. Лістінг сторінки Замовити тур ……83
Додаток D. Лістінг сторінки панель Адміністратора ……86

ВСТУП

Сучасний етап розвитку світової економіки характеризується стрімкою цифровізацією всіх сфер бізнесу, і туристична галузь не є винятком. Для успішного функціонування туристичної агенції наявність якісного вебсайту перестала бути просто конкурентною перевагою, перетворившись на виробничу необхідність. Вебсайт — це не лише візитна картка компанії, а й потужний інструмент маркетингу, продажу та взаємодії з клієнтами в режимі 24/7. Розробка власного вебресурсу для агенції «Сонячні мандри» дозволяє розширити клієнтську базу, підвищити рівень довіри до бренду та оперативно інформувати користувачів про нові напрямки та гарячі пропозиції.

Мета роботи полягає у проектуванні та створенні функціонального, адаптивного вебсайту для туристичної агенції «Сонячні мандри», який забезпечує зручний доступ до інформації про тури та автоматизує процес збору заявок від потенційних клієнтів.

Об’єкт дослідження — процес інформаційної взаємодії між туристичною агенцією та споживачами послуг у цифровому середовищі.

Предмет дослідження — методи, мови програмування та програмні засоби розробки клієнтської частини вебсайту (Front-end), спрямовані на створення зручного інтерфейсу та високої продуктивності ресурсу.

Практичне значення роботи визначається створенням реального програмного продукту, який може бути впроваджений у господарську діяльність ТА «Сонячні мандри» для підвищення ефективності її роботи. Використання сучасних стандартів HTML5, CSS3 та JavaScript дозволяє отримати швидкий та надійний інструмент, що відповідає актуальним вимогам веб-індустрії.

1. ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

В сучасному туристичному бізнесі роль вебтехнологій є надзвичайно великою. Вебсайт є ключовим інструментом продажу для туристичних агенцій. Сучасний користувач очікує від ресурсу не лише інформативності, а й високої швидкості завантаження, адаптивності та інтерактивності. Основними функціями, які має виконувати сайт ТА «Сонячні мандри», є візуалізація турів, надання актуальної інформації про ціни та умови, а також забезпечення швидкого зв'язку з менеджером.

Для визначення вимог до майбутнього проєкту було проаналізовано три типи існуючих рішень:

1. Великі агрегатори (Платформа Booking).

Booking це провідна світова платформа (агрегатор) для онлайн-бронювання житла та туристичних послуг. Вважається одним із найбільших онлайн-сервісів для мандрівників у світі. Ця система виконує такі основні функції та надає такі послуги:

— бронювання житла. Основний напрямок — готелі, хостели, апартаменти, гостьові будинки, вілли та інші варіанти розміщення.

— платформа пропонує понад 28 мільйонів варіантів розміщення у понад 220 країнах та регіонах.

— окрім житла, сервіс дозволяє бронювати авіаквитки, орендувати автомобілі, замовляти трансфер з аеропорту та шукати цікаві місця/розваги.

Ключовими особливостями для користувачів в цій системі є великий вибір послуг. Тут наявні як відомі готельні мережі, так і приватні апартаменти. Наявність вебсайту та мобільного додатка, що підтримують понад 40 мов дає можливість користувачу зручно переглядати тури та замовляти послуги. Користувачі мають можливість підбирати житло за ціною, розташуванням,

рейтингом. Більшість пропозицій передбачають можливість скасувати бронювання без штрафу до певного часу.

Спеціальна система для партнерів, яка дозволяє керувати цінами, наявністю місць та контентом є важливою для використання цієї платформи бізнесом. Booking.com інвестує величезні кошти в інтернет-рекламу (SEM), забезпечуючи партнерам високий трафік. Booking.com працює як платформа-посередник, яка отримує комісію з кожного успішного бронювання. Платформа займає домінуюче становище на ринку онлайн-бронювання, через що часто стикається з антимонопольною увагою у різних країнах.

Booking.com орієнтований на максимально швидке та зручне планування подорожей, пропонуючи користувачам інструменти для порівняння цін та пошуку найкращого варіанту проживання. Ця система має потужний функціонал, величезну базу даних, складні фільтри.

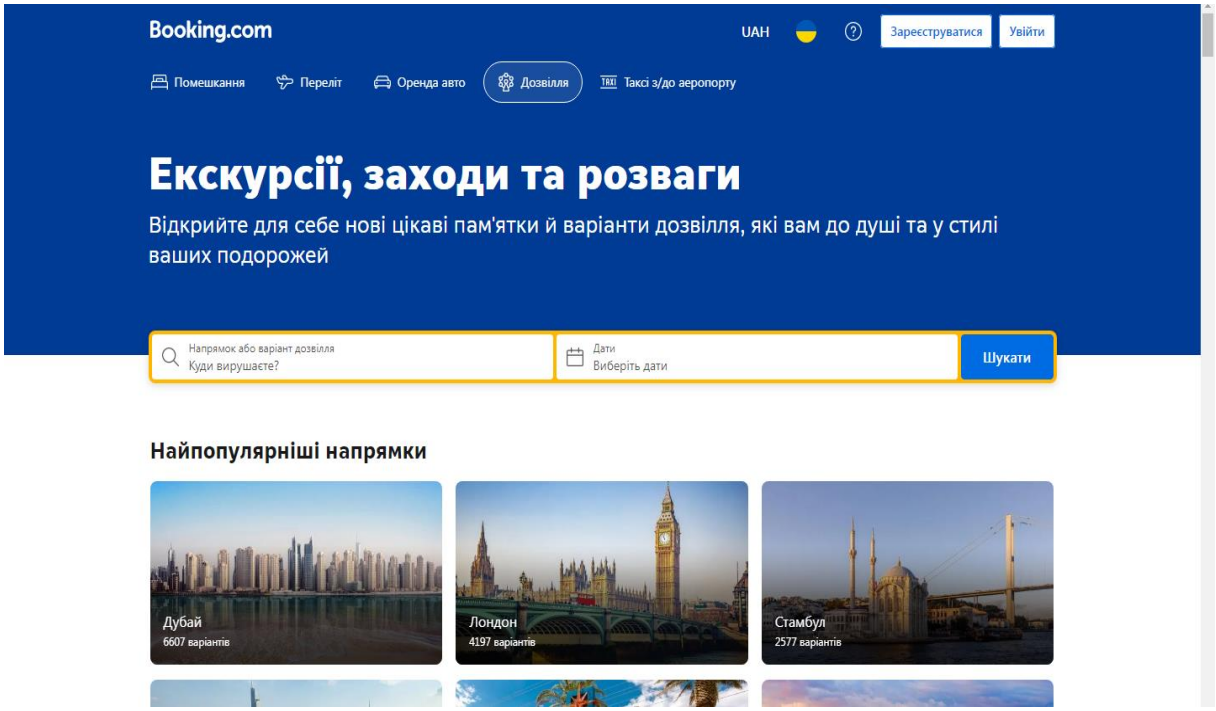

Рисунок 1.1 – Головна сторінка платформи Booking

Недоліком цієї системи є складність інтерфейсу для швидкого підбору туру, відсутність персоналізованого підходу, характерного для невеликих агенцій .

2. Вебсайти великих мережевих операторів, а саме туроператор Coral Travel. Coral Travel — це один із найбільших міжнародних туроператорів, що працює понад 30 років та входить до складу великої міжнародної групи. За допомогою сайту можна замовити організацію групових та індивідуальних турів, бронювання готелів, чартерні та регулярні авіаперельоти, страхування. Сайт цієї компанії має високу технологічність, наявність особистих кабінетів, але стандартизований дизайн не підкреслює індивідуальність локальної агенції.

3. Сайти локальних туристичних агенцій. Прикладом такого сайту є сайт туристичної агенції «Поїхали з нами».

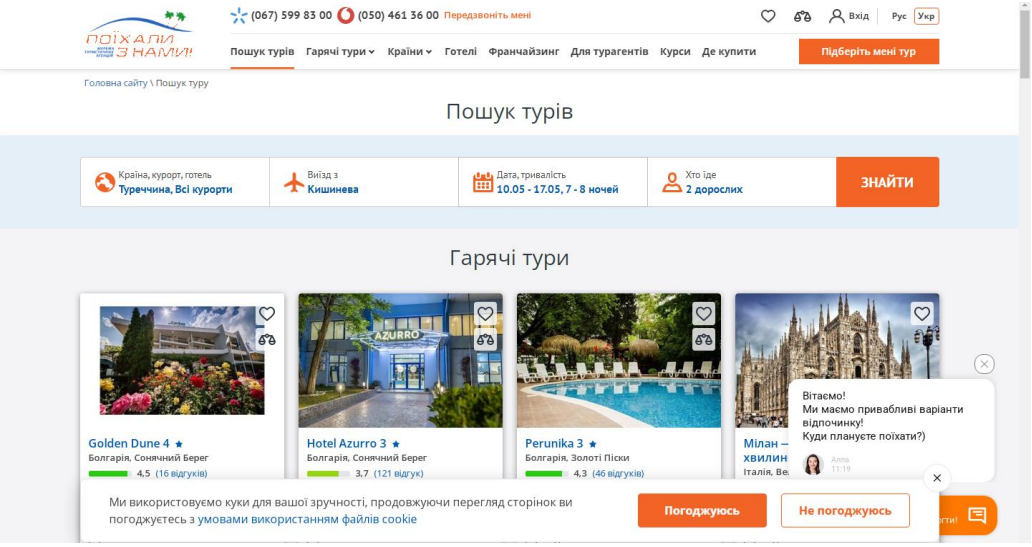

Рисунок 1.2 - Головна сторінка сайту туристичної агенції «Поїхали з нами»

Сайт цієї компанії орієнтований на конкретний регіон, має простоту та зручну навігацію. Але тут застарілий дизайн, відсутність адаптивності під мобільні пристрої, перевантаженість зайвими елементами.

Аналітичний огляд підтвердив, що для успішної конкуренції на ринку ТА «Сонячні мандри» потребує легкого, адаптивного та візуально привабливого вебсайту, який поєднує простоту навігації з сучасними технологічними рішеннями.

1.2 Технічне завдання

Метою кваліфікаційної роботи є розробка вебсайту туристичної агенції «Сонячні мандри». Розробка передбачає створення сучасного, інтуїтивно зрозумілого та адаптивного рішення для автоматизації взаємодії з клієнтами.

Вихідними даними для проектування є інформаційна структура майбутнього ресурсу, яка повинна забезпечити повний цикл взаємодії користувача з туристичним продуктом: від ознайомлення з пропозиціями до оформлення заявки на тур.

1.2.1 Найменування та область застосування

Найменування - «Вебсайт туристичної агенції «Сонячні мандри».

Область застосування: інтернет-ресурс, який є офіційним онлайн-офісом агенції в мережі Інтернет, що дозволяє компанії працювати поза межами фізичного офісу у цілодобовому режимі. Область застосування включає просування туристичних послуг, публікацію «гарячих турів» та проведення рекламних акцій для залучення нових клієнтів. Вебсайт виступає інструментом для первинного збору заявок, що дозволяє менеджерам отримувати готові запити на бронювання через форми зворотного зв'язку.

1.2.2 Призначення розробки

Користувачі можуть використовувати інформаційний ресурс як довідник для пошуку актуальної інформації про напрямки відпочинку, ціни та умови проживання в готелях. Сайт надає можливість клієнту самостійно порівняти різні туристичні продукти та обрати оптимальний варіант без попереднього візиту до офісу. Розроблений програмний продукт може бути використаний як базова платформа для подальшого розширення функціоналу. Завдяки адаптивній

верстці, область застосування не обмежується стаціонарними ПК — сайт розрахований на використання на мобільних пристроях, планшетах та ноутбуках, що розширює коло потенційних споживачів послуг.

1.2.3 Вимоги до програмного забезпечення

На основі аналізу вихідних даних визначено інструменти розробки HTML5, CSS3, JavaScript, PHP. Для створення вебресурсу туристичної агенції «Сонячні мандри» використати сучасний стек технологій Front-end розробки мови розмітки HTML5 та стилізації CSS3 для створення адаптивного інтерфейсу, семантичні теги: <header>, <nav>, <section>, <footer>, що покращить SEO-оптимізацію та доступність сайту.

Логотип агенції реалізувати за допомогою формату Scalable Vector Graphics (SVG), це забезпечить ідеальну чіткість при будь-якому масштабуванні екрана без збільшення об'єму файлів.

Забезпечити сумісність з сучасними браузерами (Google Chrome, Firefox, Safari) та оптимізацію швидкості завантаження контенту (особливо великих зображень турів).

Для побудови дизайну сайту мають бути використанні сучасні стандарти каскадних таблиць стилів CSS Custom Properties (Змінні). Використати конструкції :root для визначення кольорової палітри (--primary-color, --bg-warm тощо). Це дозволяє швидко змінювати візуальну тему всього сайту через один параметр.

Модуль display: flex використати для створення сітки гарячих пропозицій (offers-grid) та вирівнювання елементів у «шапці» сайту.

Застосувати властивість transition для плавного підйому карток турів (transform: translateY(-5px)) та плавну прокрутку сторінки (scroll-behavior: smooth).

Для Header використати linear-gradient, що створює сучасний вигляд без використання важких фонових зображень.

Використати безпечні системні шрифти (Segoe UI, Arial), що гарантує швидке відображення тексту без очікування завантаження зовнішніх шрифтових файлів. Для візуалізації турів інтегрувати високоякісні зображення через CDN (Unsplash API), що дозволяє демонструвати привабливий контент без перевантаження власного хостингу.Для розробки використати середовище Visual Studio Code — для написання та структурування вихідного коду. Для налагодження стилів у режимі реального часу та перевірки адаптивності під різні роздільні здатності екранів використати систему Google Chrome DevTools.

1.2.4 Вимоги до програмної документації

Програмна документація до вебсайту туристичної агенції «Сонячні мандри» повинна відповідати вимогам єдиної системи програмної документації (ЄСПД) та забезпечувати повне розуміння структури, функціоналу та порядку експлуатації продукту.

До комплекту документації розробленого вебресурсу мають входити:

— технічне завдання (ТЗ) основний документ, що визначає вимоги до функціональності, дизайну та технічних характеристик сайту;

— пояснювальна записка, яка містить опис обраних технологій (HTML5, CSS3, JS), обґрунтування структури коду та опис алгоритмів взаємодії користувача з формами замовлення.

— інструкція користувача (адміністратора) повинна містити детальний опис порядку розміщення сайту на хостингу, редагування текстового контенту та моніторингу заявок, що надходять через форму зворотного зв’язку.

Програмна документація повинна відповідати таким критеріям:

— документація має бути достатньою для того, щоб сторонній розробник міг зрозуміти структуру проєкту та внести необхідні корективи.

— текст повинен бути розділений на логічні блоки (розділи, підрозділи) з використанням уніфікованої термінології.

— обов’язкове використання графічних матеріалів для візуалізації архітектури сайту.

1.2.5 Техніко-економічні показники

У даному розділі наведено опис матеріально-технічної бази та програмного забезпечення, що використовувалися для розробки, тестування та розгортання вебресурсу. Обґрунтування вибору технічних і програмних засобів базується на критеріях доступності, продуктивності та економічної доцільності

Розробка сайту здійснювалася на базі персонального комп'ютера стандартної конфігурації. Потужності обраного обладнання повністю задовольняють вимогам сучасних засобів веб розробки.

Технічні характеристики ПК на якому розроблявся застосунок:

— Процесор (CPU) Intel Pentium з тактовою частотою від 2.8 GHz;

— Оперативна пам'ять (RAM) 4 GB;

— Накопичувач SSD 256 GB / 512 GB ;

— Відеокарта інтегрована (Intel UHD Graphics / AMD Radeon Graphics);

— Монітор Philips 190VW з роздільною здатністю Full HD (1440*900), що є оптимальним для верстки та тестування адаптивності інтерфейсу.

Програмне забезпечення. Для мінімізації витрат на розробку було обрано ліцензійно чисте, безкоштовне програмне забезпечення.

Системне ПЗ та середовище розробки:

— Операційна система Microsoft Windows XP. Забезпечує стабільну роботу всіх необхідних утиліт та інструментів розробника.

— Інтегроване середовище розробки (IDE) Visual Studio Code (версія 1.85). Безкоштовний, легковаговий та високопродуктивний редактор коду.

Завдяки розширенням (Live Server, Prettier, Auto Rename Tag) дозволяє значно прискорити процес написання та відладки коду.

Інструменти веброзробки (пехнологічний стек):

— Базова мова розмітки HTML5 та мова стилізації для створення семантичної та адаптивної структури сайту CSS3.

— JavaScript (ES6) для створення динамічних елементів інтерфейсу .

— Серверна мова сценаріїв PHP (версія 8.1) використовувалася для обробки та відправки форм, взаємодії з базою даних, налаштування базової логіки сайту.

Локальний сервер та допоміжні сервіси:

— Локальний сервер Open Server Panel використовувався як локальне середовище для тестування PHP-скриптів та адміністрування бази даних через інструмент phpMyAdmin.

— Для тестування відображення сайту використовувалися вебраузери Google Chrome, Mozilla Firefox . Вбудовані інструменти розробника (Chrome DevTools) застосовувалися для відладки CSS-стилів, перевірки адаптивності під мобільні пристрої та аналізу помилок у консолі JavaScript.

— Хостинг та контроль версій: Git (версія 2.43) система контролю версій для фіксації етапів розробки.

— Для зберігання репозиторію проєкту та безкоштовного хостингу статичної частини сайту використана хмарна платформа GitHub / GitHub Pages .

Для обробки та пересилання повідомлень із контактних форм сайту на електронну пошту без необхідності складного налаштування поштового сервера використано Formspree — сторонній сервіс.

Вибір зазначеного комплексу технічних та програмних засобів дозволив досягти нульових витрат на ліцензування ПЗ. Усі використані технології (HTML, CSS, JS, PHP) та інструменти (VS Code, Git, GitHub) є повністю безкоштовними. Розробка вебресурсу не потребувала закупівлі дорогого спеціалізованого обладнання чи потужних графічних станцій.

1.2.6 Стадії та етапи розробки

Процес розробки вебсайту поділяється на кілька послідовних етапів. Кожен етап має свої цілі, завдання та результати.

1. Передпроектне обстеження та аналіз вимог. Це фундамент розробки, на якому визначається призначення сайту та його цільова аудиторія. Виконується аналіз предметної області, проводиться дослідження конкурентів та сучасних трендів у обраному сегменті. Створюється технічне завдання.

2. Проектування структури та логіки (UX-дизайн). На цьому етапі розробляється інформаційна архітектура. Визначається ієрархія сторінок та навігаційного меню. Створюються прототипи (Wireframes), будуються схематичні макети сторінок для візуалізації розміщення блоків (шапка, контентна частина, футер, форми зворотного зв'язку). Моделюються процеси через побудова діаграм, що описують взаємодію користувача з системою.3.

3. Візуальне оформлення (UI-дизайн). Етап створення графічного інтерфейсу, який забезпечує естетичну привабливість та зручність користування. Вибирається стилістика. Підбір кольорової гами, типографіки (шрифтів) та графічних елементів. Оформлення всіх сторінок сайту з урахуванням принципів адаптивності (Responsive Design) для коректного відображення на десктопах, планшетах та смартфонах.

4. Програмна реалізація (Розробка). Процес перетворення графічних макетів у робочий програмний код. Поділяється на:

— Верстка (Frontend). Написання коду HTML для структури, CSS для стилізації та JavaScript для інтерактивності (анімації, валідація форм).

— Програмування (Backend). Реалізація серверної логіки, налаштування бази даних (якщо передбачено) та інтеграція з зовнішніми сервісами.

5. Тестування та налагодження. Перевірка працездатності сайту та виправлення помилок перед запуском. Використовуються такі види тестування:

— Кросбраузерне тестування. Перевірка відображення в Google Chrome, Firefox, Safari, Edge.

— Адаптивне тестування. Перевірка роботи на різних роздільних здатностях екрана.

— Валідація коду. Перевірка відповідності коду стандартам W3C.

— Функціональне тестування. Перевірка роботи всіх посилань, кнопок та форм.

6. Розміщення та підтримка. Вибір провайдера, реєстрація доменного імені та завантаження файлів проекту на сервер. Налаштування мета-тегів для кращої індексації пошуковими системами. Подальше оновлення контенту та технічна підтримка ресурсу.

1.2.7 Порядок контролю та прийому

У цьому розділі визначено види, обсяги та методи випробувань, які необхідно провести для перевірки відповідності розробленого вебсайту вимогам технічного завдання, а також порядок документального оформлення результатів та передачі готового продукту в експлуатацію.

Контроль якості та працездатності вебсайту здійснюється поетапно і включає такі види випробувань:

— Внутрішнє (авторське) тестування виконується розробником у процесі створення окремих модулів та системи в цілому. Включає перевірку коректності коду, базової логіки та відсутності критичних помилок (багів).

— Приймально-здавальні випробування (ПЗВ) проводяться спільно розробником та замовником (або керівником кваліфікаційної роботи) після завершення всіх етапів розробки та розгортання сайту на тестовому або робочому сервері (хостингу).

Приймання вебсайту здійснюється на основі перевірки його відповідності критеріям, що зафіксовані в технічному завданні. Вебсайт вважається готовим до здачі за умов виконання таких вимог:

1. Реалізовано всі модулі та функції, передбачені завданням.
2. Сайт коректно відображається та функціонує у сучасних браузерах (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) та на різних типах пристроїв (ПК, планшети, смартфони).
3. Валідність коду та оптимізація: Код відповідає сучасним стандартам розробки, а швидкість завантаження сторінок задовольняє базові вимоги

Під час приймально-здавальних випробувань обов'язковій перевірці підлягають такі компоненти системи, які відображені в таблиці 1.1.

Таблиця 1.1 - Методи та об’єкти контролю

Об'єкт контролю	Метод перевірки	Очікуваний результат
Користувацький інтерфейс (UI/UX)	Візуальний огляд, мануальне тестування на різних екранах.	Відповідність дизайн-макету, відсутність зміщення елементів.
Функціональність сайту	Виконання тестових сценаріїв (тест-кейсів) для кожної ролі користувача.	Всі інтерактивні елементи, форми та кнопки працюють коректно.
Валідація даних	Введення некоректних даних у форми (пусті поля, невірний формат email).	Система виводить зрозумілі повідомлення про помилки, захищаючи базу даних.
Безпека	Тестування стійкості до базових вразливостей.	Наявність захисту, шифрування паролів.

2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Постановка задачі

Метою даного розділу кваліфікайної роботи є визначення основних цілей розробки, переліку функціональних можливостей майбутнього вебсайту туристичної агенції «Сонячні мандри» та обґрунтування вибору технологічних засобів реалізації.

Об’єктом розробки є процес інформаційної взаємодії між туристичною агенцією та потенційними клієнтами в мережі Інтернет. Предметом розробки є вебсайт туристичної агенції «Сонячні мандри», що забезпечує презентацію послуг компанії та збір заявок від користувачів.

Основна мета проекту — створення сучасного, адаптивного та зручного веб-ресурсу, який дозволить автоматизувати процес ознайомлення клієнтів із турами, підвищити впізнаваність бренду та забезпечити ефективний канал комунікації через форми зворотного зв’язку.

Відповідно до специфіки діяльності турагенції, розроблювана система повинна виконувати наступні функції:

1. Інформаційна функція: відображення детальної інформації про компанію, актуальні напрямки відпочинку (гарячі тури, екскурсійні програми, пляжний відпочинок).
2. Навігаційна функція: зручне меню для швидкого переходу між розділами сайту (Головна, Про нас, Послуги, Галерея, Контакти).
3. Комунікаційна функція: інтеграція форми зворотного зв'язку для отримання заявок на підбір туру та консультацій.
4. Інтерактивна функція: використання інтерактивних елементів (слайдери, карти проїзду, кнопки соціальних мереж).
5. Адаптивність: коректне відображення контенту на пристроях з різною діагоналлю екрана (мобільні телефони, планшети, ПК).

Для забезпечення високої швидкості завантаження та кросбраузерності, розробка має базуватися на таких технологіях:

— Мови розмітки та стилізації HTML5 для створення семантичної структури та CSS3 для візуального оформлення і створення адаптивної сітки (Flexbox/Grid).

— Мови сценаріїв JavaScript для реалізації динамічних елементів інтерфейсу та валідації даних користувача у формах.

— Інструменти розробки - текстовий редактор коду (Visual Studio Code), графічні редактори для підготовки медіаконтенту, а також засоби тестування у різних браузерах.

Інтерфейс вебсайту «Сонячні мандри» повинен мати інтуїтивно зрозуміле розташування елементів керування. В стилістиці переважне використання теплих, «сонячних» кольорів, що асоціюються з відпочинком, та високоякісних тематичних зображень. Повинен забезпечуватися швидкий доступ до контактної інформації та кнопок швидкого зв'язку (Viber, Telegram).

Логічна структура взаємодії користувача з ресурсом деталізована у діаграмі варіантів використання (Use Case), наведеній у наступних підрозділах.

2.2 Опис та обгрунтування вибору структури та методу організації вхідних та вихідних даних

Ефективна організація даних є фундаментом стабільної роботи вебсайту. Вона визначає швидкість доступу до інформації, легкість масштабування системи та безпеку збереження контенту.

Вхідні дані — це інформація, яка надходить до системи від користувача або зовнішніх джерел для подальшої обробки. Вхідними даними є :

— реєстраційні та авторизаційні дані, це логіни, паролі, адреси електронної пошти.

— контент користувача, це текстові повідомлення, файли (зображення, PDF-документи), параметри пошукових запитів.

— технічні метадані: HTTP-заголовки, IP-адреси, дані файлів cookie.

Обґрунтування методу організації. Для передачі вхідних даних обрано формат JSON (JavaScript Object Notation). Він є легким, читабельним для людини та має вбудовану підтримку в більшості сучасних мов програмування та фреймворків. У порівнянні з XML, JSON має меншу надмірність, що критично для швидкості завантаження вебсторінок.

Вихідні дані — це результат обробки вхідної інформації, що повертається користувачу у зручному для сприйняття вигляді. До них відноситься:

— візуалізований контент, HTML-сторінки, сформовані на основі даних;

— системні сповіщення, статуси виконання операцій;

— дані у форматах PDF або Excel для адміністративної панелі, як звітність.

Вихідні дані структуруються за допомогою RESTful API архітектури. Це дозволяє розділити логіку сервера та інтерфейсу користувача (Frontend/Backend separation), забезпечуючи можливість майбутньої розробки мобільного додатка без зміни архітектури бази даних. Для організації довгострокового зберігання даних обрано реляційну модель NoSQL.

Типи даних та методи їх організації представлені у таблиці 2.1.

Таблиця 2.1 - Типи вхідних та вихідних даних та методи їх організації

Тип даних	Метод організації	Обґрунтування
Користувачі	Реляційна таблиця	Чітка структура, необхідність зв'язків з іншими сутностями
Медіафайли	Файлова система + URL у БД	Економія ресурсів БД, швидка віддача через CDN
Кеш даних	NoSQL	Для миттєвого доступу до частих запитів

2.3 Розробка алгоритму

На початку роботи була розроблена структурна схема веб-ресурсу, яка відображена на рисунку 2.1 та в графічній частині 2026.КВР.123.405.14.00.00 СС.

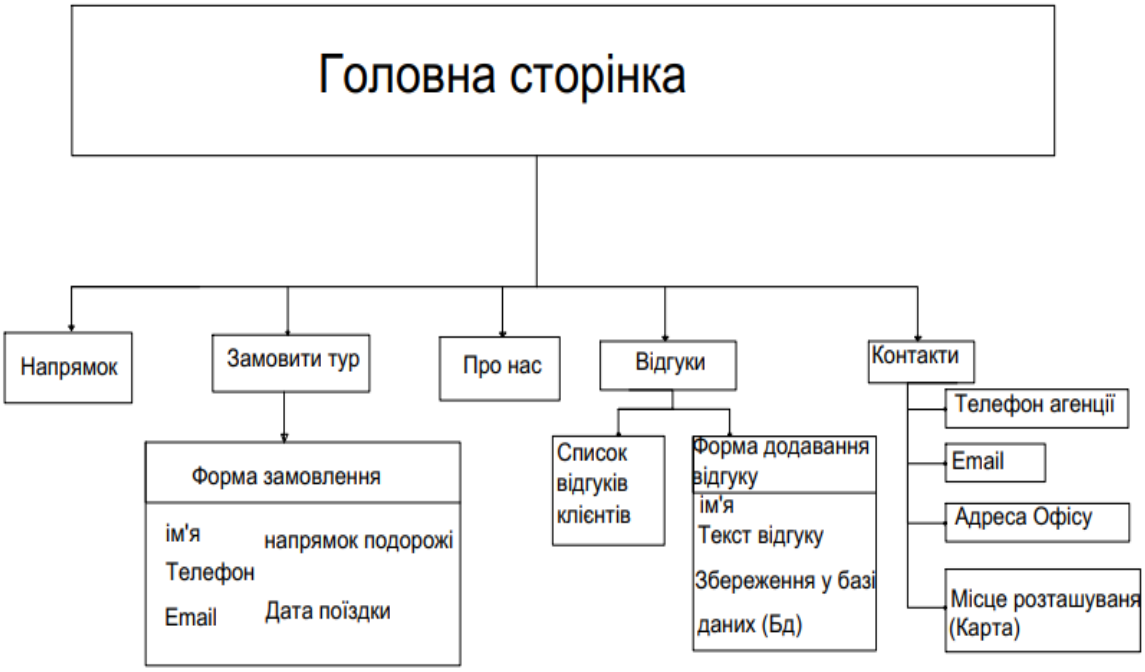

Рисунок 2.1 - Структурна схема сайту

Структурна схема відображає ієрархію сторінок та взаємозв'язки між основними логічними блоками. Вона демонструє шлях користувача від головної сторінки до цільових дій. Структура веб-сторінки — це її внутрішній каркас, який визначає, як саме розташовані елементи на екрані та як їх сприймають пошукові роботи (SEO) і користувачі.

У сучасній веброзробці для створення структури використовують семантичні теги HTML5. При розробці було дотримано стандартної структури сторінки.

Стандартна сторінка сайту ділиться на кілька ключових смислових блоків:

1. Шапка сайту (<header>). Це верхня частина сторінки, яка зазвичай повторюється на всіх сторінках сайту. Вона містить логотип компанії, назву сайту, головне меню навігації (<nav>), рядок пошуку, контакти або кнопку перемикання мови/теми.

2. Навігація (<nav>). Блок, який містить посилання на інші сторінки або розділи сайту. Містить вертикальне або горизонтальне меню. Часто вкладається всередину <header>.

3. Головний контент (<main>). Найважливіша частина сторінки, яка містить унікальний контент саме для цієї URL-адреси. На сторінці може бути лише один тег <main>. Всередині нього контент ділиться на дрібніші блоки:

— <section> — тематичні розділи.
— <article> — самостійні, завершені блоки інформації, які можна вирвати з контексту й перенести на інший сайт.
— <aside> — бічна панель (сайдбар). Містить додатковий контент: банери, рекламу, список популярних статей, теги.

4. Підвал сайту (<footer>). Нижня частина сторінки, яка зазвичай дублюється на всьому сайті. Містить аввторські права, посилання на політику конфиденційності, іконки соцмереж, карту сайту та повторне (або розширене) меню.

Перший рівень сайту туристичної агенції «Сонячні мандри» - Головна сторінка (index.html) це центральний вузол, що об'єднує всі секції через внутрішню навігацію.

Другий рівень це контентні блоки:

— Інформаційний: «Про нас» та «Напрямки».
— Взаємодія: «Гарячі пропозиції» (картки турів).
— Форма зворотного зв'язку: «Замовити тур».

На третьому рівні зовнішні сторінки/файли:

1. destinations.html – модуль для перегляду та замовлення туру;
2. order.html - модуль для замовлення туру;

3. reviewc.html - окремий модуль для перегляду та додавання відгуків;
4. contacts.html - детальна інформація про розташування та засоби зв'язку.

В цьому розділі описано логіку взаємодії користувача з сайтом та послідовність процесів, які відбуваються на сторінці. Алгоритм функціонування сайту туристичної агенції «Сонячні мандри» побудований на принципах лінійної навігації та подійної моделі взаємодії з інтерфейсом.

Алгоритм завантаження та відображення сторінки. При переході користувача за адресою сайту виконується наступна послідовність дій:

1. Ініціалізація змінних оточення. Браузер зчитує CSS-змінні (кольорову палітру) для формування візуального стилю.
2. Побудова DOM-дерева. Послідовне завантаження структури (шапка, меню, сітка турів, футер).
3. Рендеринг медіа-контенту. Асинхронне завантаження зображень з фотостоку Unsplash для оптимізації швидкості відображення тексту. Unsplash — це один із найпопулярніших у світі вебсайтів із безкоштовними фотографіями та зображеннями високої якості. Усі фотографії можна завантажувати й використовувати як для особистих, так і для комерційних цілей (наприклад, для дизайну сайтів, презентацій, соцмереж чи реклами).

Алгоритм навігації користувача. Для забезпечення зручного UX (User Experience) реалізовано алгоритм плавної навігації по секціях (Anchor Navigation). Вхідні дані це клік користувача на пункт меню (наприклад, "Напрямки"). Після цього відбувається процес:

1. Перехоплення ідентифікатора секції destinations.html.
2. Виклик функції scroll-behavior: smooth.
3. Переміщення в'юпорту до відповідного блоку.

Результатом процесу є фокусування уваги користувача на потрібному контенті без перезавантаження сторінки.

Логіка вибору та взаємодії з пропозиціями представлена у вигляді блок-схеми, яка розміщена на рисунку 2.2:

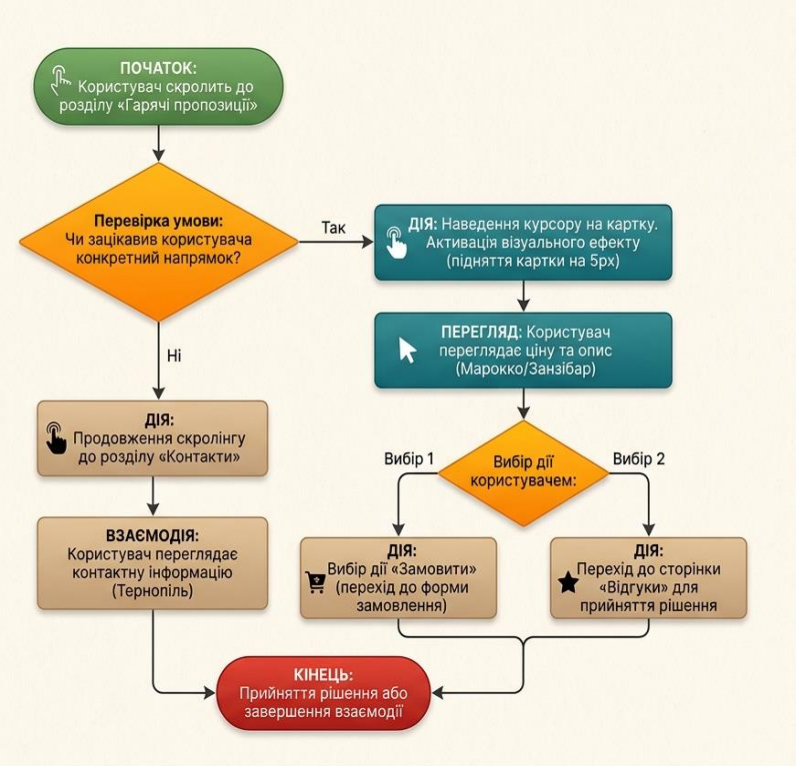

Рисунок 2.2 – Блок-схема вибору та взаємодії з пропозиціями

2.4 Визначення інформаційних зв’язків

Інформаційні зв’язки в межах розроблюваного веб-ресурсу туристичної агенції «Сонячні мандри» визначають структуру взаємодії між користувачем, інтерфейсом сайту та зовнішніми сервісами обробки даних.

Впровадження панелі адміністратора в архітектуру чистого клієнтського додатка (Clean Client-Side App) зумовлює специфіку руху інформаційних потоків. Усі інформаційні зв'язки між користувацьким інтерфейсом (Client Side), панеллю адміністратора (Admin Side) та зовнішніми сервісами реалізовані на рівні фронтенд-частини за допомогою JavaScript (API-запитів та локального сховища).

Виділяються три ключові напрямки інформаційних зв'язків, які представлені на рисунку 2.3.

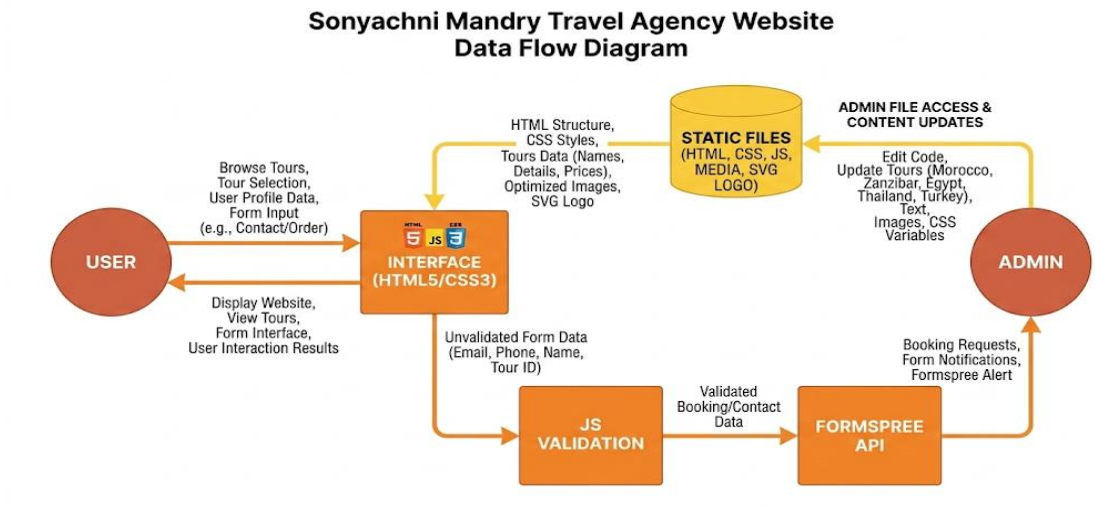

Рисунок 2.3 - Діаграма потоків даних

1. Зв'язок Панель адміністратора - Зовнішні сервіси (Керування контентом). Адміністративна панель виступає точкою ініціації потоків даних для налаштування додатка.

— При надходженні даних від користувачів (наприклад, через форму зворотного зв'язку Formspree), інформаційні потоки спрямовуються на інтегровані зовнішні API-сервіси. Панель адміністратора може взаємодіяти з API цих сервісів для моніторингу, агрегації або відображення статусів обробки.

— Зміна контенту сайту адміном відбувається шляхом надсилання API-запитів до сторонніх хмарних баз даних (наприклад, Firebase) або через оновлення статичних конфігураційних файлів.

2. Зв'язок Панель адміністратора - Браузер (Локальне керування станом). Для забезпечення безпеки, збереження сесії адміна та швидкої зміни конфігурації без перевантаження сторінок використовуються внутрішні зв'язки браузера:

— Перевірка прав доступу адміністратора (токен авторизації) передається та зберігається у LocalStorage або SessionStorage.

— Внесені адміном зміни спочатку валідуються та формуються у внутрішньому стані (State) додатка, мінімізуючи кількість запитів до зовнішніх API.

3. Зв'язок Панель адміністратора - Користувацька частина (Client Interface). Зворотний зв'язок, який визначає, як дії адміністратора впливають на те, що бачить звичайний відвідувач:

Інформаційні потоки, сформовані в адмін-панелі й збережені у зовнішній базі даних (або локальному сховищі), миттєво зчитуються клієнтською частиною сайту при завантаженні сторінки або через реактивні зв'язки (Reactive State Management). На основі цих даних інтерфейс користувача динамічно перебудовується.

2.5 Написання текстів програми

При розробці сайтів використовуються три основні категорії засобів: основні технології (код), графічні засоби та інструменти розробки.

Мови програмування та розмітки це фундамент сторінки, який забезпечує її структуру, зовнішній вигляд та базову інтерактивність.

Для реалізації даного сайту використана мова розмітки HTML5 (HyperText Markup Language).

HTML5 — це п’ята і поточна версія мови розмітки гіпертексту (HTML), призначена для структурування та представлення контенту у Всесвітній павутині.

Основне призначення HTML5 полягає в модернізації веб-стандартів для створення сучасних, кросбраузерних вебсайтів та додатків без потреби у додаткових плагінах. HTML5 — це фундамент сучасного вебу, що об'єднує розмітку, стилі (CSS) та інтерактивність (JavaScript) для створення зручних вебдодатків.

Ключові призначення та переваги HTML5:

— Забезпечення чіткої структури сайту (текст, зображення, відео) та покращення семантики за допомогою нових тегів (<header>, <footer>, <article>, <section>, <nav>).

— Вбудована підтримка відтворення аудіо та відео через теги <audio> та <video>, що усуває потребу в сторонніх програмах.

— Використання елемента <canvas> для малювання 2D та 3D графіки, анімації та створення ігор прямо в браузері (інтерактивна графіка).

— Завдяки кращій інтеграції з JavaScript, HTML5 дозволяє створювати складні, швидкі та інтерактивні вебпрограми.

— Допомагає пошуковим системам краще розуміти структуру сторінки, що покращує SEO (оптимізацію для пошукових систем).

— Можливість зберігати дані на пристрої користувача (Local Storage), що дозволяє додатку функціонувати навіть без інтернету.

— Забезпечення коректного відображення сайтів на різних пристроях, включаючи мобільні телефони та планшети.

CSS3 (Cascading Style Sheets) це сучасна мова стилів, що використовується для оформлення зовнішнього вигляду вебсторінок, написаних на HTML або XML. Вона визначає кольори, шрифти, розташування елементів (макет), анімацію та адаптивність під різні екрани, розділяючи структуру вмісту та його візуальне представлення.

Ключові особливості CSS3:

— HTML відповідає за структуру (текст, заголовки), а CSS — за те, як це виглядає.

— Дозволяє сайтам автоматично підлаштовуватися під розміри екранів телефонів, планшетів та ПК.

— CSS3 має підтримку нових візуальних ефектів округлених кутів (border-radius), тіней (box-shadow), градієнтів та прозорості без використання зображень.

— Можливість створювати складні переходи (transition) та анімації (animation) прямо в CSS.

— Підтримка сучасних модулів, таких як Flexbox та Grid Layout, для гнучкого позиціонування елементів.

Стилі зазвичай зберігаються в окремому файлі з розширенням .css, який підключається до HTML-документа. Це дозволяє змінювати дизайн усього сайту, редагуючи лише один файл. Відповідає за повне візуальне оформлення, дизайн, анімації та адаптивність сторінки.

Основні техніки та модулі мови стилів CSS3:

— CSS-змінні (Custom Properties) використані у блоці :root для централізованого управління колірною палітрою (--primary-color, --accent-color, --bg-warm). Це забезпечує візуальну цілісність усього сайту та легкість внесення змін.

— Modern Layouts (Сучасні макети). CSS Grid (Module Level 1): Основний інструмент для створення сітки напрямків (.destinations-grid). Використання repeat(auto-fit, minmax(300px, 1fr)) дозволило створити повністю адаптивну сітку, яка самостійно змінює кількість колонок залежно від ширини екрана, без використання медіа-запитів (@media).

— Flexbox (Flexible Box Layout Module) використаний для вирівнювання елементів всередині карток (наприклад, щоб кнопка "Замовити" завжди була внизу, незалежно від довжини тексту) та в шапці/навігації.

— Box Model & Styling. Властивості для закруглення кутів (border-radius), додавання тіней (box-shadow), налаштування відступів (margin, padding) та типографіки.

— Transitions & Transforms (Анімації). Властивості transition у поєднанні з transform: translateY() та scale() використані для створення плавних, візуально приємних ефектів при наведенні курсору на картку (картка піднімається, а фотографія збільшується).

JavaScript (JS) — це високорівнева мова програмування, яка є одним із основних засобів веб-розробки поряд із HTML та CSS. Якщо HTML відповідає

за структуру сторінки, а CSS — за її зовнішній вигляд, то JavaScript робить його інтерактивним.

Як вже було сказано, розробка веб-ресурсу туристичної агенції «Сонячні мандри» виконувалася за допомогою сучасних стандартів верстки та принципів семантичного програмування. Код написаний з використанням мови розмітки HTML5 та мови стилізації CSS3. Структура сайту створювалася шляхом написання семантичних тегів, які визначають призначення кожного елемента. Базова структура документа створювалася за допомогою тегів <html>, <head>, <body>. Створення шапки сайту з назвою «Сонячні мандри» використовувався тег <header>. Створення навігаційного меню тег <nav>. Для наповнення карток контентом (зображення, заголовки, опис, ціна) були використані теги <img>, <h3>, <p>. Тег <footer> використаний для створення підвалу сайту з контактами. Клієнтська частина — це все, що завантажується в браузер користувача (клієнта), відображається на екрані та з чим користувач безпосередньо взаємодіє. Вона відповідає за структуру, дизайн та базову логіку інтерфейсу. Клієнтська частина цього сайту була створена за допомогою трьох фундаментальних технологій веб-розробки, які поєднані в одному файлі index.html.

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Туристична агенція | Твій теплий відпочинок</title>
  <style>
    :root {
      --primary-color: #D35400;
      --secondary-color: #E67E22;
      --accent-color: #F1C40F;
      --bg-warm: #FFF5E1;
      --text-dark: #4A2700;    }
```

Програмний код, який є початком коду вебсторінки, написаний мовами HTML (структура сайту) та CSS (його стилі/зовнішній вигляд). У ньому створюється основа сайту для туристичної агенції та задається тепла колірна гама для дизайну. Використані теги:

— <!DOCTYPE html> — вказує браузеру, що це сучасний документ стандарту HTML5.

— <html lang="uk"> — відкриває головний тег сторінки та вказує, що її мова — українська.

— <head> — технічний розділ сторінки, інформація з якого не відображається безпосередньо на сайті, але потрібна браузеру та пошуковим системам.

— <meta name="viewport" content="width=device-width, initial-scale=1.0"> — забезпечує адаптивність (щоб сайт гарно відображався і на смартфонах, і на комп'ютерах).

Головна сторінка сайту туристичної агенції «Сонячні мандри» показана на рисунку 2.4:

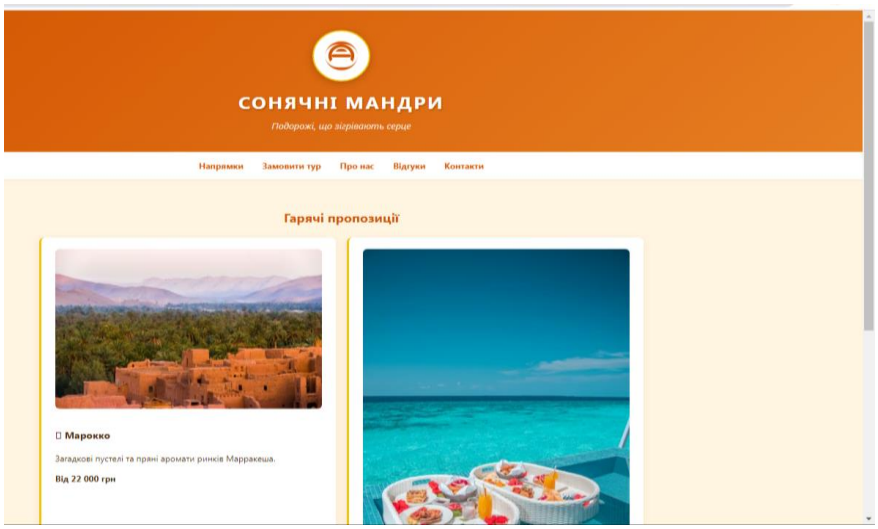

Рисунок 2.4 - Головна сторінка сайту

Структура HTML-документа. Код побудований за модульним принципом, що забезпечує чіткий поділ контенту на логічні блоки:

— Header містить інтерактивний SVG-логотип та головний заголовок. Використано лінійний градієнт для створення «теплого» візуального ефекту.

— Nav блок навігації з використанням властивості position: sticky, що дозволяє меню залишатися видимим під час скролінгу.

— Section (Tours) основний контентний блок, реалізований за допомогою гнучких блоків (Flexbox), що забезпечує адаптивність карток турів.

— Footer містить контактну інформацію та метадані сайту.

Для забезпечення цілісності дизайну використано CSS-змінні (Custom Properties). Це дозволяє легко змінювати кольорову гаму сайту в одному місці:

CSS

```
:root {
   --primary-color: #D35400;  /* Теплий теракотовий */
   --secondary-color: #E67E22;
   --accent-color: #F1C40F;   /* Сонячний жовтий */
   --bg-warm: #FFF5E1;        /* Кремовий фон */
}
```

Основні технічні особливості реалізації:

1. Використано властивість CSS flex-wrap для коректного відображення сайту на мобільних пристроях.

2. Додано властивість transition для плавного підняття карток турів (transform: translateY) при наведенні курсору, що покращує UX (досвід користувача).

3. Логотип реалізовано за допомогою вбудованого SVG-коду, що мінімізує кількість HTTP-запитів та забезпечує ідеальну чіткість на будь-яких екранах.

Опис роботи з медіа-ресурсами. Зображення для напрямків (Марокко, Занзібар) інтегровані через зовнішні посилання з параметрами оптимізації (auto=format&fit=crop), що дозволяє завантажувати полегшені версії фотографій залежно від розміру контейнера.

Поточна архітектура дозволяє легко додати:

— Динамічну фільтрацію турів за допомогою JavaScript.
— Обробку форм бронювання через інтеграцію з PHP або Node.js API.
— Підключення шрифтів Google Fonts для покращення типографіки.

У контексті проєкту «Сонячні мандри», JavaScript використаний як інструмент для перетворення інформаційного ресурсу на інтерактивний сервіс.

JavaScript керує потоками інформації в напрямку сторонніх сервісів:

— Після успішної валідації JavaScript перехоплює стандартне надсилання форми й передає валідовані дані (Validated Booking/Contact Data) на Formspree API.

— JS приймає відповідь від API про успішну доставку листа й виводить користувачу повідомлення (наприклад: «Дякуємо! Ваша заявка на тур успішно відправлена»), не перезавантажуючи при цьому саму сторінку сайту.

JavaScript динамічно керує контентом та інтерфейсом, реагує на дії відвідувачів. Зчитуються масиви даних про доступні тури (назви, описи, ціни, зображення для Марокко, Занзібару, Єгипту, Таїланду, Туреччини) та їх динамічний рендеринг на сторінці у вигляді карток.

Для адміністрування ресурсу засобами JavaScript створено окрему логіку. JS відстежує зміни конфігурації сайту, які робить адміністратор, валідує їх у реальному часі та формує внутрішній стан додатка.

Перевіряє права доступу адміністратора, генерація та збереження токена авторизації всередині браузера через LocalStorage або SessionStorage. Це дозволяє адміну залишатися в системі без необхідності постійного введення пароля на кожній сторінці.

Здійснюється оновлення статичних файлів або робота з конфігураційними даними (наприклад, JSON) для миттєвої перебудови інтерфейсу користувача

Реалізація форми замовлення та інтеграція з Formspree. Особлива увага при розробці сторінки «Замовити тур» була приділена створенню функціонального інтерфейсу для взаємодії з клієнтом. Оскільки проект на даному етапі реалізований як статичний клієнтський додаток (Clean Client-Side App), для обробки користувацьких даних було обрано раціональне технічне рішення — інтеграцію з сервісом Formspree.

Для передачі даних використовується Форма зворотного зв'язку. Ця форма налаштована на відправку POST-запитів безпосередньо на API-ендпоінт Formspree. Це дозволяє отримувати дані з форми (ім’я, контактні дані, обраний тур) на електронну пошту адміністратора агенції без потреби у власному серверному сценарії (PHP чи Node.js).

Перед відправкою даних JavaScript виконує базову перевірку коректності заповнення полів (валідація email-адреси та номеру телефону). Це мінімізує ризик отримання некоректних заявок та покращує досвід користувача.

Розробка сторінки «Напрямки» (destinations.html), яка відображена на рисунку 2.5 створювалася за допомогою сучасних, легких та адаптивних веб-інтерфейсів без використання важких JavaScript-фреймворків.

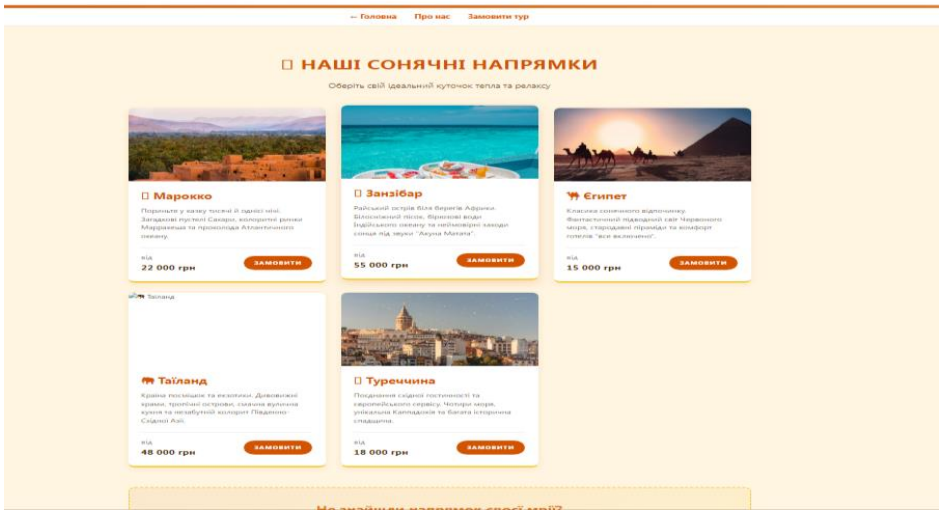

Рисунок 2.5 - Сторінка «Напрямки»

Основний акцент було зроблено на семантичній розмітці та потужних можливостях сучасного CSS. Сторінка «Напрямки» була розроблена як чистий статичний клієнтський додаток (Clean Client-Side App). Вибір на користь семантичного HTML5 та сучасного CSS3 (зокрема Grid та Variables) дозволив створити візуально привабливий, швидкий та повністю адаптивний інтерфейс, використовуючи мінімальний набір інструментів, що робить проект легким у підтримці та масштабуванні.

Сторінка «Замовити тур» призначена для перетворення інформаційного ресурсу на інтерактивний сервіс. На даному етапі реалізована форма зворотного зв'язку, яка працює через зовнішній сервіс Formspree. Це дозволяє обробляти запити користувачів без складного серверного коду, зберігаючи високу продуктивність сайту.

Форма реалізована за допомогою тегу <form>, а поля введення (ім'я, телефон, email, вибір туру) структуровані за допомогою тегів <label> та <input>. Використовуються вбудовані атрибути HTML5 (required, type="email", pattern), що забезпечує перевірку коректності даних безпосередньо в браузері перед відправкою. Для вирівнювання полів форми та кнопок використано модуль Flexbox, що гарантує зручне заповнення анкети як з настільних ПК, так і з мобільних пристроїв. Особлива увага приділена властивості flex-wrap, яка дозволяє елементам форми коректно перешиковуватися на вузьких екранах. У межах подальшого масштабування проєкту «Сонячні мандри», сторінка «Замовити тур» отримає наступні вдосконалення через JavaScript:

1. Впровадження скриптів для глибшої перевірки масок номерів телефонів та формату електронної пошти.

2. Використання JS для створення модальних вікон, що підтверджують успішне відправлення замовлення, або слайдерів для вибору додаткових послуг у турі.

3. У майбутньому планується заміна Formspree на власне серверне рішення (PHP або Node.js) для обробки та зберігання бронювань у базі даних у режимі реального часу.

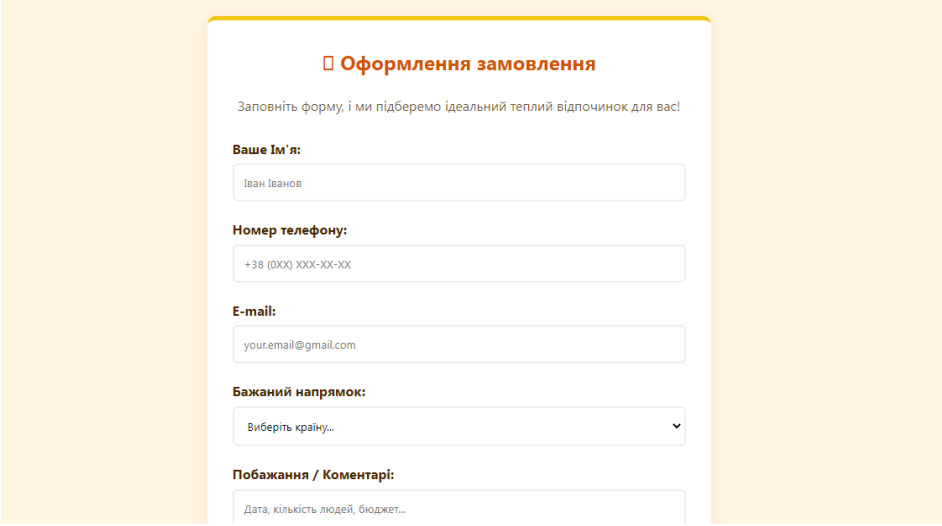

Рисунок 2.6 - Сторінка «Замовити тур»

Панель адміністратора призначена для керування туристичними напрямками на сайті турагенції "Сонячні мандри". Вона написана на HTML, CSS (стилі) та JavaScript, а для збереження інформації використовується хмарна база даних Google Firebase.

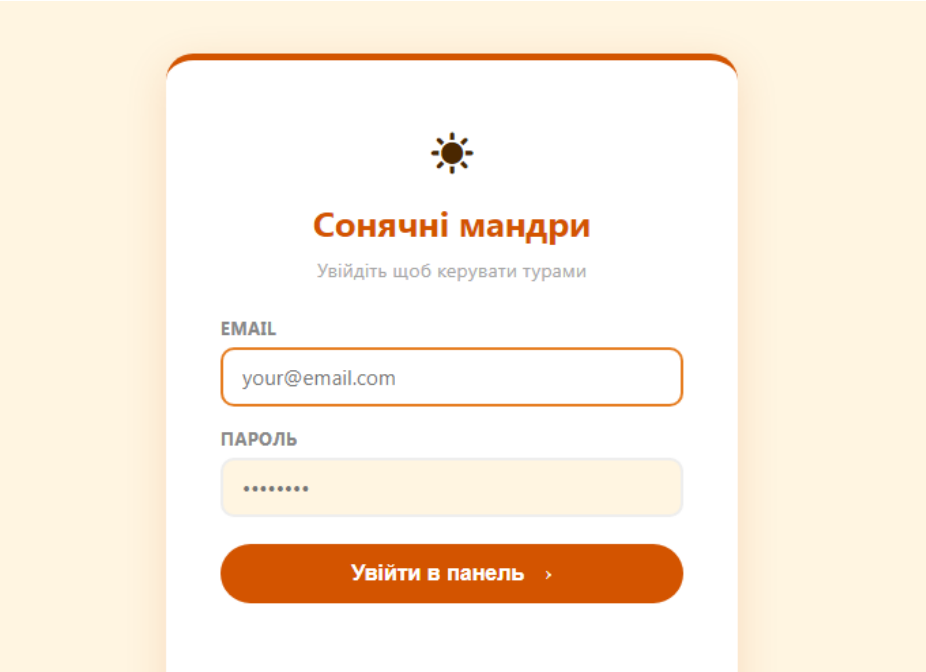

Рисунок 2.7 - Вхід в панель Адміністратора

Головні функції сторінки адміністратора. Для будь-якої системи потрібне керування вмістом (CMS) через авторизацію (Вхід). Сторонній користувач не побачить список турів, доки не введе правильний логін (Email) та пароль.

CRUD-операції (Керування даними):

— Create — додавання нового туру (назва, ціна, опис, посилання на фото).

— Read — відображення всіх наявних турів у зручній таблиці з попереднім переглядом картинок.

— Update — редагування вже створеного туру.

— Delete — видалення туру з бази даних.

Спливаючі віконця внизу екрана, які хвалять за успішне збереження або сварять за помилки.

Код об'єднаний в один файл, але чітко ділиться на три великі частини: За допомогою CSS (Стилі в тегу <style>) прописаний сучасний, адаптивний дизайн у теплих "сонячних" тонах (помаранчевий, жовтий, кремовий). Використовуються технології Flexbox та CSS Grid. Це означає, що адмінка буде акуратно виглядати як на великому моніторі комп'ютера, так і на екрані смартфона. Поля введення підсвічуються при натисканні, таблиця підсвічує рядки при наведенні мишкою, а кнопки плавно змінюють колір.

За допомогою HTML (Структура в тегу <body>) розроблено два блоки, які динамічно змінюють один одного:

 — картка форми входу.

 — сама адміністративна частина.

Сторінка Панель Адміністратора має шапку (header), меню навігації (nav), форму для додавання/редагування картки туру, таблицю для перегляду списку та "підвал" (footer). З використанням JavaScript & Firebase (Логіка в тегу <script>) розроблено аналітичну структуру сторінки. Він використовує офіційні модулі Google Firebase (версії 11.0.0):

— onAuthStateChanged: Спеціальний "наглядач" від Firebase Auth. Якщо адміністратор уже залогінений — скрипт автоматично ховає вікно входу, показує адмінку і завантажує тури. Якщо ні — тримає форму входу.

— loadTours та renderTable: Звертаються до колекції 'tours' у базі даних Firestore, сортують їх за внутрішнім порядком (orderBy('order')), забирають дані та "малюють" рядки таблиці прямо на льоту за допомогою інструменту map().

— saveTour: Універсальна функція. Якщо ми просто заповнили поля — вона викликає addDoc (створити новий запис). Якщо ми натиснули кнопку "Редагувати" біля якогось туру — форма запам'ятовує його ID, перемикається в режим редагування, а кнопка збереження викликає updateDoc (оновити існуючий запис). У коді прописано конфігурацію:

JavaScript const firebaseConfig = { projectId: "soniachni-mandry",
authDomain: "soniachni-mandry.firebaseapp.com",};

Це означає, що проект підключено до реального хмарного бекенду Firebase користувача. Всі додані тури миттєво відлітають у хмару Google і звідти ж можуть транслюватися на головний сайт для звичайних клієнтів (index.html).

2.6 Тестування та налагодження веб сайту

Тестування та налагодження є критично важливими етапами життєвого циклу розробки вебсайту, які забезпечують відповідність кінцевого продукту технічному завданню, вимогам безпеки та стандартам якості. Для перевірки працездатності розробленого ресурсу застосовується комплексний підхід, що включає наступні рівні тестування:

— Функціональне тестування. Перевірка коректності роботи всіх елементів інтерфейсу (форм зворотного зв'язку, навігаційного меню, кнопок, фільтрів). Кожна функція має бути протестована на відповідність очікуваному результату за методом «чорної скриньки».

— Кросбраузерне тестування. Перевірка ідентичності відображення та функціонування сайту в сучасних браузерах (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge). Це дозволить переконатися, що специфічні CSS-властивості та скрипти JS працюють стабільно у різних середовищах. Кросбраузерне тестування, яке дозволяє переконатися у стабільній та ідентичній роботі вебресурсу в різних сучасних браузерах і на мобільних пристроях є надзвичайно важливим. Ретельний контроль усіх цих процесів гарантує високу якість кінцевого продукту, забезпечуючи швидку, безпечну та комфортну взаємодію користувача з інтерфейсом сайту.

— Адаптивне тестування (Responsive Testing). Контроль коректності відображення контенту на пристроях з різною роздільною здатністю екрана (смартфони, планшети, десктопи).

— Тестування продуктивності. Аналіз швидкості завантаження сторінок та оптимізації медіафайлів.

Процес налагодження (Debugging) полягає у виявленні, локалізації та усуненні помилок, знайдених під час тестування. В ролі основних інструментів для виконання цих завдань можна використати Chrome DevTools для інспектування DOM-структури, налагодження CSS-стилів у реальному часі та моніторингу помилок у консолі JavaScript. Network Panel для аналізу швидкості завантаження ресурсів та перевірки статусів HTTP-запитів до сервера. Lighthouse для автоматизованого аудиту продуктивності, доступності (accessibility) та SEO-оптимізації. На етапі налагодження особливу увагу потрібно приділяти обробці виняткових ситуацій (Error Handling), таких як некоректне введення даних, з метою запобігання критичним збоям системи.

На рисунках 2.8 та 2.9 наведено відображення сайту в різних браузерах Google Chrome та Mozilla Firefox відповідно.

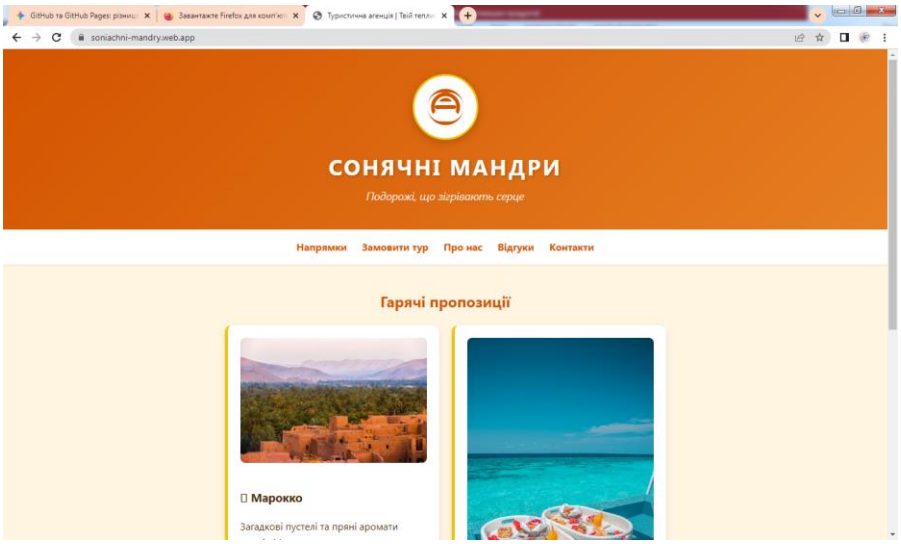

Рисунок 2.8 - Відображення сайту в браузері Google Chrome

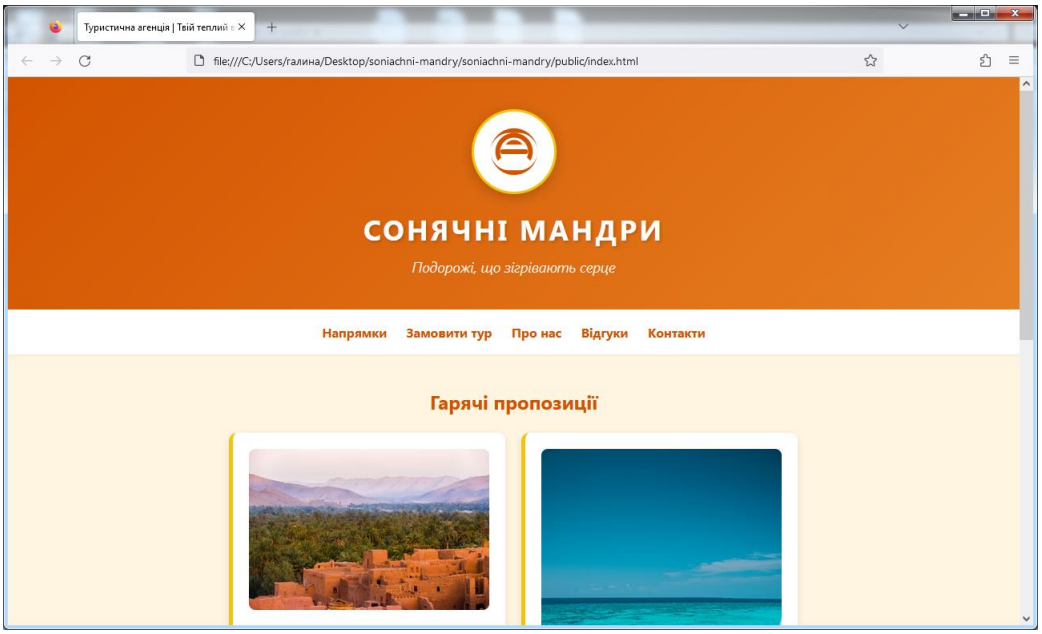

Рисунок 2.9 - Відображення сайту в браузері Mozilla Firefox

Аналізуючи два скриншоти відображення сайту «Сонячні мандри» у різних браузерах (Google Chrome та Mozilla Firefox) та на моніторах із різною роздільною здатністю, можна зробити детальний порівняльний висновок.

Основна структура сайту (логотип, головна назва, слоган, навігаційне меню та блок «Гарячі пропозиції») залишається абсолютно стабільною. Елементи не «з'їжджають» і тримають свої позиції в обох випадках.

Тексти відображаються однаково чітко, з правильним згладжуванням як у Chrome, так і у Firefox.

Помаранчевий колір шапки (header), кремовий колір фону та відтінки на фотографіях картриджів турів збігаються. Немає видимих спотворень через роботу рендерингу різних браузерів.

Головна відмінність між двома зображеннями масштаб та роздільна здатність:

— Перший скриншот (Chrome, web.app). Сайт відкрито на моніторі з ширшим екраном (або меншим масштабуванням). Завдяки цьому ми бачимо більше контенту по вертикалі — зокрема, підпис до туру «Марокко» («Загадкові пустелі та пряні аромати...»).

— Другий скриншот (Firefox, локальний файл). Монітор має меншу висоту екрана. Через це нижня частина карток турів зрізається краєм екрана, і користувачеві доведеться гортати (скролити) сторінку вниз, щоб побачити текст під фотографіями.

Після перегляду сайту через різні браузери можна зробити висновок. Сайт має якісну адаптивну основу. Головні елементи дизайну, колірна гама та читабельність тексту повністю збережені в обох браузерах. Різниця у видимій кількості контенту є природною, оскільки вона залежить від фізичних розмірів монітора користувача та роздільної здатності екрана. Сайт адекватно підлаштовується під простір.

3. СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція по встановленню та налаштуванню веб ресурсу

Розміщення розробленого вебсайту в глобальній мережі Інтернет є фінальним заключним етапом. Це робить вебсайт доступним для кінцевих користувачів. Розміщення сайту в Інтернеті складається з трьох основних етапів: підготовка файлів, вибір платформи (хостингу) та налаштування адреси (домену). Цей процес включає вибір хостинг-провайдера, налаштування серверного середовища, розгортання кодової бази та фінальну перевірку працездатності за новою доменною адресою. Для розміщення даного сайту в інтернеті було використано систему Firebase.

Google Firebase — це потужна хмарна платформа від компанії Google, створена для швидкої розробки мобільних застосунків (iOS, Android) та веб-сайтів. Вона дозволяє розробнику повністю сфокусуватися на створенні гарного інтерфейсу (фронтенду), дизайну та логіки на стороні клієнта, не витрачаючи час на написання серверного коду, розгортання баз даних, налаштування сертифікатів безпеки. Увесь бекенд Google бере на себе.

Основні сервіси та можливості. Firebase — це не просто база даних, а ціла екосистема, яка ділиться на кілька категорій:

1. Розробка та створення (Build). Використовує гнучкі NoSQL бази даних. Їхня головна перевага — синхронізація в реальному часі. Як тільки адмін змінює ціну туру в базі, вона без перезавантаження сторінки змінюється на екранах усіх користувачів, які зараз відкрили сайт.

Готова система реєстрації та входу користувачів. Дозволяє налаштувати вхід через Email/пароль , за номером телефону, або через Google, Apple, Facebook всього за кілька рядків коду.

Хмарне сховище для важких файлів (фотографій, відео, звітів, аудіо), які завантажують користувачі сайту.

Безкоштовний, дуже швидкий і захищений (SSL/HTTPS) хостинг для розміщення твоїх готових HTML/CSS/JS сайтів в інтернеті.

2. Аналітика та покращення (Google Analytics). Firebase безкоштовно збирає детальну аналітику: скільки людей зайшло на сайт, з яких вони країн, які кнопки натискають частіше та на якому етапі закривають сторінку.

В даному розділі описано процедуру розміщення вебсайту «Сонячні мандри» на сучасній хмарній платформі.

1) Першим кроком є підготовка проєкту на Firebase Console. Потрібно перейти на Firebase Console, увійти під своїм Google-акаунтом і натиснути Create a project (Створити проєкт). Вказати назву проєкту та завершити створення (Google Analytics можна увімкнути за бажанням).

2) Встановлення інструментів Firebase (CLI). Відкрити термінал на комп'ютері та встановити інструменти командного рядка Firebase глобально за допомогою npm (йде в комплекті з Node.js): npm install -g firebase-tools.

3) Авторизуватися в системі. У терміналі вводимо команду для входу у свій Google-акаунт: firebase login. У браузері відкриється вікно, де вибираємо той акаунт, на якому створювали проєкт і дозволити доступ.

4) Ініціалізація проєкту. Через термінал перейти у папку зі своїм сайтом (cd / soniachni-mandry) і запустити майстер налаштувань: firebase init hosting.

5) Публікація сайту (Деплой). Потрібно переконайтися, що всі фінальні файли сайту (обов'язково з файлом index.html) знаходяться у вказаній у пункті 4 папці (наприклад, public). Після цього виконати фінальну команду: firebase deploy.

Після успішного деплою в терміналі з'явиться рядок Hosting URL. Це є безкоштовна адреса створеного сайту в інтернеті https://soniachni-mandry.web.app. При внесені змін в коді сайту, не потрібно буде проходити весь процес знову — достатньо просто оновити файли в робочій папці та знову прописати команду firebase deploy.

Open Server Panel (OSP) — це потужне та зручне середовище для локальної розробки вебсайтів на Windows. Воно містить усе необхідне: вебсервери (Apache, Nginx), бази даних (MySQL, PostgreSQL, MariaDB) та різні версії PHP.

Open Server Panel (OSP) — це потужне та зручне середовище для локальної розробки вебсайтів на ОС Windows. Воно містить усе необхідне: вебсервери (Apache, Nginx), бази даних (MySQL, PostgreSQL, MariaDB) та різні версії PHP.

Інструкція з інсталяції та першого запуску актуальної версії (6.x+).

Завантаження та системні вимоги:

1. Перейдіть на офіційний сайт ospanel.io.

2. Перейдіть у розділ «Завантаження».

3. Завантажте інсталятор актуальної версії 6.x.

Інсталяція:

1. Запустіть завантажений файл інсталятора (.exe файл).

2. Оберіть мову встановлення та ознайомтеся з ліцензійною угодою.

3. Вибір шляху інсталяції. Встановлюйте програму в корінь диска, наприклад: C:\OSPanel. Шлях до програми не повинен містити пробілів та кириличних символів (кирилиця в шляху або імені користувача Windows часто стає причиною помилок роботи модулів).

4. Дочекайтеся завершення копіювання файлів.

5. Встановлення системних компонентів (MSVC). Для роботи PHP та баз даних у Windows обов'язково потрібні бібліотеки Microsoft Visual C++. Після завершення інсталяції Open Server запропонує вам встановити або оновити ці компоненти. Обов'язково погоджуйтеся та встановлюйте їх, інакше модулі сервера не зможуть запуститися. Після встановлення компонентів перезавантажте комп'ютер.

Перший запуск та керування.

У новій версії керування є повністю портативним та консольним. Щоб запустити панель керування:

1. Перейдіть у папку з програмою (C:\OSPanel).

2. Запустіть файл ospanel.exe. Відкриється фірмовий командний рядок Open Server Panel.

3. Головні команди, які знадобляться в консолі:

— ospan status — перевірити стан сервера та активні модулі;

— ospan start — запустити всі увімкнені модулі;

— ospan stop — зупинити роботу сервера;

— ospan menu — відкрити графічне веб-меню керування в браузері.

Також керувати сервером можна через браузер за адресою http://localhost/ або увівши команду `ospan menu` — там ви зможете вмикати потрібні версії PHP (наприклад, PHP 8.1 чи 8.3) та бази даних одним кліком.

Створення першого сайту. У новій версії Open Server Panel структура доменів стала набагато гнучкішою.

1. У папці OSPanel знайдіть директорію home.

2. Створіть там папку для вашого проєкту, наприклад: soniachni-mandry.

3. Всередині папки soniachni-mandry створити підпапку з назвою public (це сучасний стандарт для безпеки вебсерверів).

4. Саме в папку public покладіть ваші робочі файли сайту (index.html).

5. Перезапустіть сервер, виконавши почергово команди ospan stop та ospan start (або через веб-інтерфейс керування).

6. Тепер сайт буде доступний у браузері за локальною адресою: http://soniachni-mandry/.

3.2 Інструкція з використання тестових наборів

Для розробленого проєкту туристичної агенції "Сонячні мандри" тестування є надзвичайно важливим етапом. Оскільки сайт побудований на базі HTML5, CSS3, JavaScript та інтегрований із сервісом Formspree для обробки заявок клієнтів, тестові набори (тест-кейси) мають покривати як візуальну частину, так і логіку взаємодії користувача з інтерфейсом.

Перевірка синтаксису та форматування. Перед запуском будь-яких тестів код має бути «чистим».

1. Запустіть Linter, виконайте команду у терміналі (наприклад, npm run lint).
2. Виправте помилки, лінтер вкаже на пропущені крапки з комою, невірні відступи або дублювання правил у CSS.
3. Форматування, використовуйте Prettier, щоб привести всі файли до єдиного вигляду.

Статичні сайти мають проходити перевірку на відповідність стандартам, валідація HTML та CSS.

Скористайтеся онлайн-сервісом W3C Markup Validation або CLI-інструментом html-validator. Він знайде незакриті теги або атрибути, які більше не підтримуються.

Використовуйте інструменти типу broken-link-checker, щоб переконатися, що всі внутрішні та зовнішні посилання ведуть на існуючі сторінки.

Тестування продуктивності та доступності (Lighthouse). Це критичний етап для SEO та зручності користувачів.

1. Відкрийте сайт у Chrome.
2. Перейдіть у DevTools (F12) -> вкладка Lighthouse.
3. Оберіть категорію "Accessibility" та "Best Practices".
4. Аналіз результату, якщо показник нижче 90, вивчіть рекомендації (наприклад, відсутність alt у зображень або низький контраст тексту).

Тестування адаптивності (Visual Regression). Якщо проект великий, можна використовувати інструменти візуального порівняння (наприклад, BackstopJS). Система робить скриншот «еталонної» версії сайту. Після змін робиться новий скриншот. Програма підсвічує кожен змінений піксель, щоб ви випадково не посунули футер на іншій сторінці.

Найкращий спосіб використовувати набори — автоматизувати їх за допомогою GitHub Actions або GitLab CI.

Типовий робочий процес (Workflow):

1. Потрібно зробити git push.
2. Сервер автоматично запускає npm test.
3. Якщо хоча б один тест (лінтер або валідатор посилань) не пройшов — збірка сайту блокується до виправлення помилок.

Таблиця 3.1 - Чек-лист перед релізом

Що перевірити	Інструмент
Валідність HTML	W3C Validator
Швидкість завантаження	PageSpeed Insights / Lighthouse
Коректність JS-логіки	Jest / Vitest
Адаптивність	Chrome Device Mode / BrowserStack
Мета-теги та SEO	SEO Quake / OpenGraph Check

Налаштування автоматичних скріншотів (Visual Regression Testing) дозволяє автоматично порівнювати зовнішній вигляд сайту «до» та «після» внесення змін. Найпопулярнішим інструментом для цього є BackstopJS.

BackstopJS працює на базі Node.js. Встановіть його локально у ваш проект:

Bash

```
npm install --save-dev backstopjs
```

Після встановлення ініціалізуйте структуру конфігурації:

Bash

```
npx backstop init
```

Ця команда створить файл backstop.json та папки для зберігання скріншотів.

Відкрийте створений файл backstop.json. Основні параметри, які варто налаштувати:

— viewports: масив пристроїв (наприклад, Mobile та Desktop), на яких будуть робитись знімки.

— scenarios: список сторінок (URL), які треба перевірити.

— misMatchThreshold: допустимий відсоток різниці (наприклад, 0.1%), при якому тест вважатиметься успішним.

Процес використання складається з трьох основних команд:

1. Створення еталона (Reference). Коли ви впевнені, що сайт виглядає ідеально, створіть «золоті знімки»:

Bash

npx backstop reference

Ці зображення збережуться в папку backstop_data/bitmaps_reference.

2. Запуск тесту. Після того, як ви змінили CSS або структуру HTML, запустіть порівняння:

Bash

npx backstop test

Програма відкриє браузер, зробить нові скріншоти та порівняє їх з еталоном.

Поради для стабільного тестування

— Якщо на сайті є анімації при завантаженні, потрібно додати delay: 2000 (2 секунди), щоб скріншот робився після завершення всіх ефектів.

— Оскільки різні ОС (Windows/macOS/Linux) рендерять шрифти по-різному, для командної роботи краще запускати BackstopJS через Docker. Це гарантує, що у всіх розробників результати будуть ідентичними.

3. Якщо зміни у верстці були запланованими, навмисно змінили колір кнопок, то потрібно оновити еталони:

Bash

npx backstop approve

Це зробить останні тестові знімки новими еталонами для майбутніх перевірок.

3.3 Інструкція з експлуатації вебсайту

Сайт є цифровою візитівкою туристичної агенції. Його головна мета — залучення клієнтів, демонстрація гарячих турів та забезпечення швидкого переходу до бронювання відпочинку.

Головна сторінка (Landing Page / Index). Підтримує мобільні пристрої та ПК завдяки динамічній розмітці flex-wrap та метатегу viewport. Локалізація українська (lang="uk").

Сайт розділений на чотири логічні блоки:

1. Шапка сайту (Header)

— Логотип: Стилізоване SVG-зображення у золотистому колі з тінню (втілює сонячну тематику).

— Назва: «Сонячні мандри».

— Слоган: *«Подорожі, що зігрівають серце»*.

2. Меню навігації містить 5 активних посилань для переходу на інші сторінки проєкту. У таблиці 3.2 записано HTML-файли, які знаходяться в кореневій папці soniachni-mandry .

Таблиця 3.2 – Файли кореневої папки soniachni-mandry

Пункт меню	Цільовий файл (по	Призначення сторінки
Напрямки	destinations.html	Повний каталог країн та курортів
Замовити тур	order.html	Форма заявки або бронювання туру
Про нас	about.html	Інформація про компанію, сертифікати, команду
Відгуки	reviewc.html	Враження клієнтів та форма зворотного зв'язку
Контакти	contacts.html	Карта, соцмережі, додаткові телефони
Панель адміністратора	admin.html	Редагування турів

3. Блок «Гарячі пропозиції» (Tours Section). Центральна частина сторінки, яка демонструє актуальні тури у вигляді карток:

1) Марокко — вартість від 22 000 грн.
2) Занзібар — вартість від 55 000 грн.

4. Підвал сайту (Footer). Містить юридичні та контактні дані:

— Фізична адреса: м. Тернопіль, вул. Сонячна, 10
— Телефон для зв'язку: +38 (067) 368-45-12
— Знак авторського права (Copyright) станом на 2026 рік.

Адмін-панель захищена системою Firebase Authentication. Доступ до керування мають лише співробітники з підтвердженим логіном та паролем.

Алгоритм роботи в адмін-панелі:

1. Перейдіть за адресою адмін-панелі (введіть у браузері посилання на адмін-панель).

2. У вікні, що з'явилося, заповніть поле електронної адреси та секретний пароль. Натисніть кнопку «Увійти в панель →» або натисніть клавішу *Enter*.

Якщо з'явилася помилка червоний текст *«Невірний email або пароль»* означає, що ви не правильно записали ці дані або цей акаунт не зареєстрований у консолі Firebase. Спробуйте ще раз.

Після успішного входу перед вами відкриється форма додавання, яка автоматично завантажує дані на сайт.

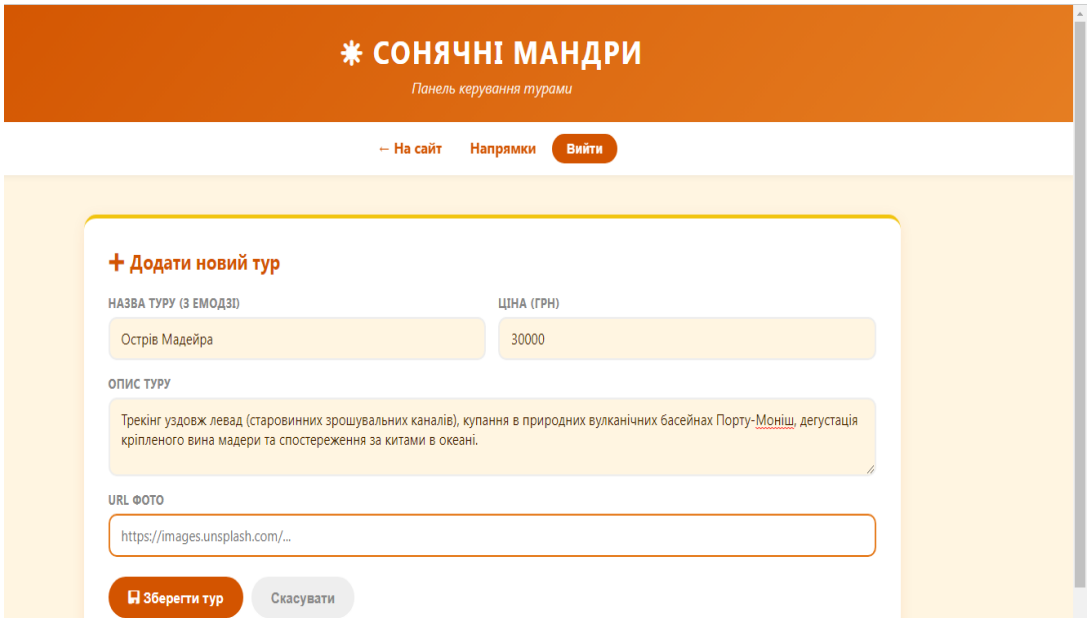

Рисунок 3.1 - Вікно додавання туру

1. Назва туру: введіть країну чи місто.

2. Ціна (грн): вкажіть вартість туру (лише цифри, без пробілів, точок та слова «грн»). Система сама відформатує її гарно (наприклад, перетворить 22000 на від 22 000 грн).

3. Опис туру: напишіть короткий текст (до 2-3 речень), що входить у вартість (готель, харчування, екскурсії). На сайті довгий текст автоматично скоротиться трикрапкою, щоб картки були рівними.

4. URL фото**:** скопіюйте посилання на якісне зображення з інтернету (краще використовувати безкоштовні стоки, як-от Unsplash).

Як тільки ви вставите коректне посилання, знизу форми з'явиться Попередній перегляд фото. Якщо картинки немає — посилання недійсне.

5. Натисніть «Зберегти тур». У правому нижньому кутку екрана з'явиться зелене сповіщення *«Тур додано!»*, і він миттєво з'явиться в таблиці та на головній сторінці сайту.

Редагування існуючого туру. Якщо змінилася ціна, опис або готель, тур можна легко оновити:

1. Прокрутіть сторінку вниз до таблиці «Всі тури на сайті».

2. Знайдіть потрібний тур і в колонці «Дії» натисніть жовту кнопку « Редагувати».

3. Сторінка плавно підніметься вгору, а форма автоматично заповниться даними цього туру. Заголовок форми зміниться на *«Редагувати тур»*.

4. Внесіть необхідні зміни (наприклад, оновіть ціну).

5. Натисніть « Зберегти тур». Дані в базі оновляться, а порядок туру на сайті не порушиться.

6. Якщо ви передумали редагувати, натисніть сіру кнопку «Скасувати» — форма очиститься і повернеться в режим створення.

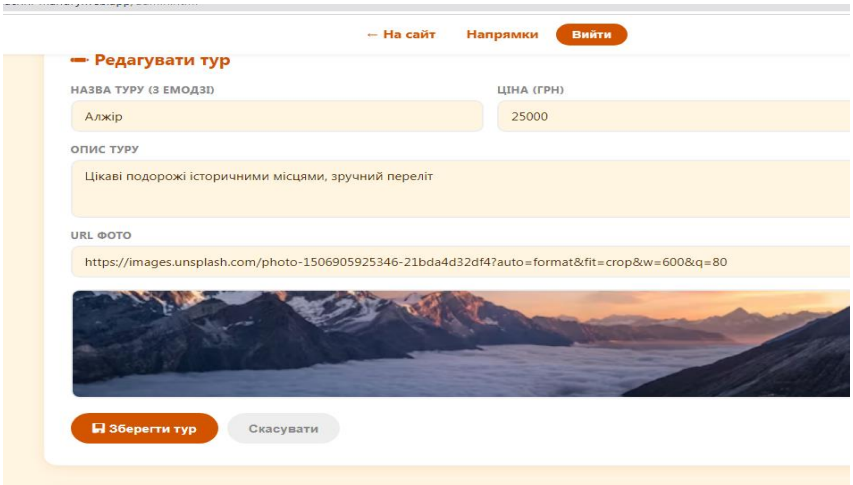

Рисунок 3.2 - Вікно редагування туру

Якщо тур більше неактуальний або місця закінчилися, його можна видалити назовсім.

1. У таблиці турів знайдіть пропозицію, яку треба прибрати.
2. Натисніть червону кнопку з кошиком «Видалити» (або просто значок кошика).
3. Браузер покаже спливаюче вікно для підтвердження: *«Видалити тур "Назва туру"?»*.
4. Натисніть ОК для підтвердження. Тур зникне з бази даних та із сайту назавжди.

Після завершення роботи з турами обов'язково вийдіть із системи, щоб сторонні особи не змогли змінити дані вашого сайту.

— У верхньому білому меню (навігації) натисніть помаранчеву кнопку «Вийти».
— Система закриє сесію та поверне вас на безпечний екран авторизації з сонечком.

Кнопка «← На сайт». У верхньому меню є швидке посилання. Ви можете відкрити сайт в іншій вкладці, щоб одразу перевіряти, як виглядають додані вами тури. Намагайтеся не брати занадто «важкі» фотографії за посиланнями, щоб головна сторінка сайту «Сонячні мандри» завантажувалася у клієнтів максимально швидко.

4. ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини кваліфікаційної роботи на тему: «Розробка вебсайту туристичної агенції «Сонячні мандри» є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності розробки, і прийняття рішення про її подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки або робіт в даному напрямку.

Об'єктом розробки є вебсайт.

Розрахунок вартості розробки виконується в декілька етапів:

– описати технологічний процес розробки із зазначенням трудомісткості кожної операції;

– визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;

– визначити суму матеріальних затрат;

– обчислити витрати на електроенергію;

– розрахувати транспортні витрати;

– нарахувати суму амортизаційних відрахувань;

– визначити суму накладних витрат;

– скласти кошторис та визначити собівартість робіт;

– розрахувати ціну робіт;

– визначити економічну ефективність та термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Розробка технічного завдання	кер. проекту	16
2.	Розробка дизайну	інженер	24
3.	Верстка та програмування	інженер	80
4.	Наповнення контентом	інженер	16
5.	Тестування	лаборант	12
8.	Запуск та підтримка	інженер	4
		кер. проекту	4
Разом	–		156

Сумарний час виконання операцій технологічного процесу становить 156 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України “Про оплату праці” заробітна плата – це “винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу”.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$З_{осн.} = Т_{с} \cdot К_{г} \qquad (4.1)$$

де: T_c – тарифна ставка, грн.(приймаємо для керівника – 70 грн./год, інженера – 65 грн./год., лаборанта – 53 грн./год.);

$K_г$ – кількість відпрацьованих годин.

Отже основна заробітна плата для:

Керівника проекту $З_{осн2}$= 20*70,00= 1400,00 грн.

інженера $З_{осн3}$= 124*65,00= 8060,00 грн.

лаборанта $З_{осн4}$= 12*53,00= 636,00 грн.

Сумарна основна заробітна плата становить

$$З_{осн}=1400,00+8060,00+636,00=10096 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$З_{дод.} = З_{осн.} \cdot К_{допл.}, \qquad (4.2)$$

де $K_{допл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15.

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проекту $З_{дод2}$=1400,00 *0,12 = 168 грн.

інженера $З_{дод3}$= 8060,00*0,12= 967,20 грн.

лаборанта $З_{дод4}$= 636,00*0,12 = 76,40 грн.

Загальна додаткова заробітна плата становить:

$$З_{дод}=168+967,20+76,32= 1211,52 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{о.п.}$) визначаються за формулою:

$$В_{о.п.} = З_{осн.} + З_{дод.} \qquad (4.3)$$

$$B_{о.п}=10096,00+1211,52 = 11307,52 \text{грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$В_{єсв} = Воп \cdot 0{,}22, \qquad (4.4)$$

$$B_{єсв} = 11307{,}52*0{,}22=2487{,}65 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 - Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. *6=3+4+5*
		Тариф-на ставка, грн.	К-сть від-працьов. год.	Фактичн нарах. зарплати, грн.			
1	Кер.проекту	70	20	1400,00	168,00	-	-
2	Інженер	65	124	8060,00	*967*,20	-	-
3	Лаборант	35	35	636,00	76,32	-	-
Разом				10096,00	1211,52	2487,65	13795,17

Отже, загальні витрати на оплату праці становлять 13795,17 грн.

4.3 Розрахунок матеріальних витрат

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни:

$$M_{B\,i} = q_i \cdot p_i, \qquad (4.5)$$

де: q_i – кількість витраченого матеріалу *i*-го виду;

p_i – ціна матеріалу *i*-го виду.

Звідси, загальні матеріальні витрати можна визначити:

$$З_{м.в.} = \sum М_{B\ \ i}, \qquad (4.6)$$

Проведені розрахунки занесемо у таблицю 4.3.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна 1-ці, грн.	Загальна сума витрат, грн.
1	Стокові зображення	шт.	10	35,00	350
2	Друк	листів	112	0,50	56
3	Друк ватманів	листів	1	18,00	18
4	USB Флешка	шт.	1	2004.4	200
Р а з о м					464,00

Отже, загальна сума матеріальних витрат на розробку програми становить 464,00 грн.

4.4 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$З_е = W \cdot T \cdot S\ , \qquad (4.7)$$

де: W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп’ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в

залежності від виконуваних робіт (згідно табл.4.1) і споживані потужності наступні: комп’ютер – 0,82 кВт/год.

$$З_{ек}=0,82*136*15,94= 1777,63 \text{ грн.}$$

Витрати на електроенергію становлять 1777,63 грн.

4.5 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки.

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$А = \frac{Б_B \cdot Н_A}{100}, \qquad (4.8)$$

де: А – амортизаційні відрахування за звітний період, грн.;

$Б_B$ – балансова вартість групи основних фондів на початок звітного періоду, грн.;

$Н_A$ – норма амортизації.

Оскільки для розробки сайту та його тестування використовується один ПК, вартістю 27800 грн., то сума амортизаційних відрахувань становитиме :

$$А = \frac{27800,00 \cdot 4}{100} = 1008,21 \text{ грн.}$$

4.6 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$Н_в = В_{о.п.} * 0,2..0,6 \qquad (4.9)$$

де $Н_В$ – накладні витрати.

$$Н_В = 11307,52*0,4 = 5518,07 \text{ грн.}$$

4.7 Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	11307,52	50,11
Єдиний соціальний внесок	2487,65	11,02
Матеріальні витрати	464,00	2
Витрати на електроенергію	1777,63	7,8
Амортизаційні відрахування	1008,21	4,4
Накладні витрати	5518,07	24,4
Собівартість	22563,08	100

Собівартість ($С_В$) НДР розрахуємо за формулою:

$$C_B = B_{о.п.} + B_{с.з.} + З_{м.в.} + З_е + T_в + A + H_в \qquad (4.10)$$

Отже, собівартість дорівнює $С_В$= 22563,08 грн.

4.8 Розрахунок ціни НДР

Ціну робіт можна визначити за формулою:

$$Ц = \frac{C_B \cdot (1 + P_{рен}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \qquad (4.11)$$

де: $P_{рен.}$ – рівень рентабельності;

K – кількість замовлень, од.;

$B_{і.н.}$ – вартість носія інформації, грн.;

$ПДВ$ – ставка податку на додану вартість.

$Ц$= 22563,08*(1+0,30)*(1+0,2)= 35198,40 грн.

4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності ($T_{ОК}$).

$$ЧТВ = -К_B + \sum_{i=1}^{t} \frac{Г_П}{(1+i)^t}, \qquad (4.12)$$

де:

$К_В$ – затрати на проект;

$Г_п$ – грошовий потік за t – ий рік;

t – відповідний рік проекту;

i - величина дисконтної ставки (10…15%).

$$ЧТВ = -22563{,}08 + \frac{12635{,}32}{(1+0{,}1)^1} + \frac{12635{,}32}{(1+0{,}1)^2} + \frac{12635{,}32}{(1+0{,}10)^3} =$$

$$= 8859{,}12 \text{ грн.}$$

Якщо ЧТВ ≥ 0, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ОК} = T_{ПВ} + \frac{H_В}{Г_{ПР}} \qquad (4.13)$$

де: $T_{ПВ}$ – період до повного відшкодування витрат, років;

$H_В$ – невідшкодовані витрати на початок року, грн.;

$Г_{ПР}$ – грошовий потік на початок року, грн.

$$T_{ок} = 2 - \frac{633{,}98}{12635{,}32} = 2{,}1 \text{ р}$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 -Техніко-економічні показники розробки веб-сервісу

№ п/п	Показник	Значення
1.	Собівартість, грн.	22563,08
2.	Плановий прибуток, грн.	12635,4
3.	Ціна, грн.	35198,4
4.	Чиста теперішня вартість, грн.	8859,12
5.	Термін окупності, рік	2,1

Загальна вартість розробленого веб-сервісу становить 35198,4 грн. Чистої теперішньої вартості ми досягаємо на третій рік, що є позитивним показником. Тому проводити проектні роботи варто і вкладені кошти окупляться за 2,1 року

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ

5.1 Вимоги безпеки до відеодисплейних терміналів та профілактика порушень зору

Сучасні вимоги безпеки до організації робочих місць, обладнаних моніторами, регулюються Наказом Мінсоцполітики України № 207. Основна увага приділяється параметрам робочої зони, які мінімізують навантаження на організм та органи зору.

Організація робочого місця та ергономіка.

— Екран монітора має розташовуватися на відстані 60–70 см від очей користувача (але не ближче ніж 50 см).

— Оптимальний напрямок погляду — зверху вниз. Лінія зору має бути перпендикулярною до центра екрана, а сам центр монітора повинен знаходитися на 10–20° нижче рівня очей.

— Робочий стіл і стілець мають регулюватися за висотою. Кут між стегном і тулубом, а також у ліктьовому та колінному суглобах має становити приблизно 90–100°, а стопи повинні повністю спиратися на підлогу або підставку.

Вимоги до освітлення та параметрів середовища

— Освітленість на поверхні столу в зоні розміщення клавіатури та екрана має становити 300–500 люкс (згідно з ДБН В.2.5-28:2018).

— Монітор не можна розміщувати екраном до вікна або безпосередньо під лампами верхнього освітлення, щоб уникнути відблисків (бліків), які викликають сильне перевтомлення очей. Вікна обов'язково мають бути обладнані жалюзі або шторами.

Розташування пристрою введення - виведення інформації має забезпечувати добру видимість екрана ВДТ, зручність ручного керування в зоні досяжності моторного поля і за висотою - 900...1300 мм, за шириною 400...500 мм.

Робоче місце з ВДТ слід обладнати пюпітром для документів, що легко переміщуються.

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідно застосування приекранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат.

Комп'ютерний зоровий синдром та профілактика порушень зору. Тривала робота за ВДТ призводить до розвитку Комп'ютерного зорового синдрому (КЗС). Його основними симптомами є астенопія (швидка втома очей, розмитість зображення, біль в ділянці очних ямок) та синдром «сухого ока» (відчуття піску в очах, печіння, почервоніння через рідке моргання — біля екрана людина моргає у 3–4 рази рідше).

Комплекс профілактичних заходів:

1. Під час 8-годинного робочого дня необхідно робити перерви по 5–10 хвилин через кожну годину роботи або по 15 хвилин через кожні дві години. Робота без перерви понад 2 години заборонена.
2. Кожні 20 хвилин роботи за монітором потрібно на 20 секунд перевести погляд на об'єкт, що знаходиться на відстані 20 футів (близько 6 метрів). Це розслаблює циліарний м'яз ока, який відповідає за фокусування.
3. Під час перерв слід виконувати прості вправи: рухи очима вгору-вниз, вліво-вправо, кругові обертання, а також заплющування очей на 5–10 секунд для відновлення слізної плівки.
4. Важливо виставити оптимальний баланс яскравості (екран не повинен «світитися» сильніше, ніж аркуш паперу, покладений поруч на столі).

5.2 Безпечність застосування інформаційного обладнання

Під інформаційним обладнанням (або обладнанням інформаційних технологій — OIT) розуміють комп'ютери, сервери, периферійні пристрої (принтери, сканери), мережеве та телекомунікаційне устаткування. Забезпечення безпеки під час його експлуатації поділяється на кілька ключових аспектів:

1. Електробезпека та захист від механічних пошкоджень

Електричний струм є головним фактором небезпеки при роботі з IT-інфраструктурою. Безпечність досягається через:

— Якісне заземлення. Усі металеві неструмопровідні частини корпусів системних блоків, серверних шаф та принтерів повинні бути надійно заземлені.

— Захист від перевантажень. Застосування пристроїв захисного відключення (ПЗВ) та джерел безперебійного живлення (ДБН/UPS), які запобігають короткому замиканню та перегріву кабелів.

— Конструкційну безпеку. Корпуси пристроїв мають унеможливлювати випадковий доступ користувача до внутрішніх елементів під напругою або гарячих деталей (наприклад, термовузлів у лазерних принтерах).

2. Пожежна безпека. Інформаційне обладнання виділяє значну кількість теплової енергії. Для запобігання пожежам встановлюються вимоги :

— Забороняється перекривати вентиляційні отвори системних блоків та моніторів. Серверні приміщення обов'язково обладнуються системами кондиціонування.

— Корпуси сучасного обладнання виготовляються з пластику, що не підтримує горіння (самозатухаючі матеріали).

— 3. Електромагнітна сумісність та випромінювання. Інформаційне обладнання не повинно генерувати електромагнітні завади, які можуть порушити роботу інших пристроїв (наприклад, медичного чи навігаційного обладнання), а рівень його власного електромагнітного та іонізуючого випромінювання має бути безпечним для людини.

4. Ергономічні вимоги та екологічність. Безпечність застосування також включає мінімізацію акустичного шуму (від кулерів охолодження) та утилізацію відпрацьованого обладнання (обмеження використання небезпечних речовин, як-от свинець чи ртуть, відповідно до європейських директив).

Для роботи з офісними моніторами та ПК слід керуватися ДСанПіН 3.3.2.007-98. НПАОП 0.00-1.29-97 «Правила захисту від статичної електрики».

Цей нормативно-правовий акт з охорони праці встановлює детальні технічні вимоги щодо запобігання шкідливому впливу статичної електрики та контролю за її накопиченням на робочих місцях промислових підприємств.

У сфері санітарних норм та гігієни праці слід дотримуватися ДСанПіН 3.3.6.096-2002 «Державні санітарні норми та правила при роботі з джерелами електромагнітних полів».

Саме цей документ визначає сучасні граничнодопустимі рівні (ГДР) постійних магнітних полів, електричних полів промислової частоти та інших випромінювань.

Оскільки найпоширенішим джерелом електростатичного поля в офісах та навчальних закладах є комп'ютери й монітори, для регулювання необхідних параметрів потрібно використовувати наказ Мінсоцполітики України № 207 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Цей документ повністю регулює мікроклімат, параметри випромінювання та безпеку на таких робочих місцях.

Вимоги безпеки до робочих місць працівників з екранними пристроями.

1. Робочі місця працівників з екранними пристроями мають бути спроектовані так і мати такі розміри, щоб працівники мали простір для зміни робочого положення та рухів.

2. Для забезпечення безпеки та захисту здоров'я працівників усе випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня (вплив на людину факторів довкілля - шуму, вібрації, забруднювачів, температури тощо, який не спричиняє соматичних або психічних

розладів, а також змін стану здоров'я, працездатності, поведінки, що виходять за межі пристосувальних реакцій) з погляду безпеки та охорони здоров'я працівників.

3. Організація робочого місця працівника з екранними пристроями має забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.

4. Освітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном і навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98.

5. Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями має підтримуватись на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 42 (далі - ДСН 3.3.6.042-99).

6. Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.

7. Робоче крісло має бути стійким і дозволяти працівнику з екранними пристроями легко рухатися та займати зручне положення.

Сидіння має регулюватися по висоті, спинка сидіння - як по висоті, так і по нахилу.

Мінімальні вимоги безпеки під час роботи з екранними пристроями

1. Щодня перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень.

2. Після закінчення роботи екранні пристрої слід відключати від електричної мережі.

3. У разі виникнення аварійної ситуації необхідно негайно відключити екранний пристрій від електричної мережі.

4. Не допускається:

виконувати технічне обслуговування, ремонт і налагодження екранних пристроїв безпосередньо на робочому місці працівника під час роботи з екранними пристроями;

відключати захисні пристрої, самочинно проводити зміни у конструкції та складі екранних пристроїв або їх технічне налагодження;

працювати з екранними пристроями, у яких під час роботи виникають нехарактерні сигнали, нестабільне зображення на екрані та інші несправності.

5. Під час виконання робіт операторського типу, пов'язаних з нервово-емоційним напруженням, у приміщеннях під час роботи з екранними пристроями, на пультах і постах керування технологічними процесами та в інших приміщеннях мають дотримуватися оптимальні умови мікроклімату відповідно до вимог ДСН 3.3.6.042-99.

Мінімальні вимоги безпеки до екранних пристроїв.

1. Екранні пристрої не мають бути джерелом ризику для працівників.

2. Усе випромінювання, за винятком видимої частини електромагнітного спектра, має бути зведене до незначного рівня з погляду безпеки і охорони здоров'я працівників.

3. Символи на екранних пристроях мають бути чіткими, відповідного розміру. Між символами і рядками символів має бути належна відстань.

4. Зображення на екрані має бути стабільним, без миготінь або інших видів нестабільності.

5. Яскравість та/або контрастність символів має легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов.

6. Вибираючи екрани, слід надавати перевагу таким екранам, які легко та вільно повертаються і нахиляються відповідно до потреби працівника.

7. За необхідності може використовуватись окрема підставка або регульований стіл для розміщення екрана.

8. Екран не має відблискувати або відбивати світло, щоб не викликати дискомфорту у працівника під час роботи з екранними пристроями.

9. Вибираючи клавіатуру, слід надавати перевагу такій клавіатурі, яка відкидається і є автономною (відокремленою від екрана), щоб працівник міг вибрати зручну робочу позу й уникнути втоми рук (кисті і верхньої частини руки).

10. Поверхня клавіатури має бути матовою, щоб уникнути віддзеркалювання. Розташування клавіш і самі клавіші мають полегшувати роботу із клавіатурою. Позначення клавіш повинно бути достатньо контрастним і розбірливим.

11. Устаткування, яке входить до робочої станції, не має виділяти надлишкового тепла, що може спричинити незручності працівникам під час роботи з екранними пристроями.

12. Під час розробки, вибору, замовлення та модифікації програмного забезпечення, а також під час розробки завдань, що передбачають використання устаткування з екранними пристроями, роботодавець має керуватися таким програмним забезпеченням, яке відповідає розв’язуваним завданням і є простим у використанні, а де необхідно - адаптованим до рівня знань і досвіду працівника.

Окрім технічних та програмних характеристик, безпека праці забезпечується регулярним моніторингом умов на робочих місцях. Роботодавець зобов'язаний організовувати періодичні інструктажі з охорони праці, контролювати дотримання працівниками режимів праці та відпочинку (регламентованих перерв для зниження зорової та нервово-емоційної втоми), а також забезпечувати проведення планових медичних оглядів. Впровадження комплексного підходу до ергономіки та гігієни праці дозволяє мінімізувати ризики виникнення професійних захворювань, пов'язаних із тривалим використанням екранних пристроїв, і підтримувати стабільно високий рівень працездатності персоналу.

5.3 Забезпечення нормованого рівня освітленості робочого місця програміста

Штучне освітлення в приміщеннях з робочими місцями, обладнаними ВДТ ЕОМ та ПЕОМ, має здійснюватись системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, у разі переважної роботи з документами, допускається застосування системи комбінованого освітлення (крім системи загального освітлення додатково встановлюються світильники місцевого освітлення).

Зазначення освітлення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк. Якщо ці значення освітленості неможливо забезпечити системою загального освітлення, допускається використовувати місцева освітлення. При цьому світильники місцевого освітлення слід встановлювати таким чином, щоб не створювати бліків на поверхні екрана, а освітленість екрана має не перевищувати 300 лк.

Як джерела світла в разі штучного освітлення мають застосовуватись переважно люмінісцентні лампи типу ЛБ. У разі влаштування відбитого освітлення у виробничих та адміністративно-громадських приміщеннях допускається застосування металогалогенних ламп потужністю 250 Вт. Допускається застосування ламп розжарювання у світильниках місцевого освітлення.

Система загального освітлення має становити суцільні або преривчасті лінії світильників, розташовані збоку від робочих місць (переважно ліворуч), паралельно лінії зору працюючих.

Допускається використання світильників таких класів світорозподілу

— прямого світла - П;

— переважно прямого світла - Н;

— переважно відбитого світла – В

Для загального освітлення слід застосовувати світильники серії ЛПО3б із дзеркальними гратами, укомплектовані високочастотними пускорегулювальними апаратами (ВЧ ПРА). Застосування світильників без розсіювачів та екрануючих грат заборонено.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 град. з вертикаллю в повздовжній та поперечній площинах має становити не більше ніж 200 кд/м^2, захисний кутсвітильників - не менше ніж 40 град.

Світильники місцевого освітлення повинні мати просвічуючий відбивач із захисним кутом, не меншим ніж 40 град.

Слід передбачити обмеження прямих відблисків від джерел природного та штучного освітлення. При цьому яскравість світлих поверхонь (вікна, джерела штучного освітлення), що розташовані в полі зору повинна бути не більше ніж 200 кд/м^2.

Необхідно обмежувати відблиски на робочих поверхнях відносно джерел природного і штучного освітлення. При цьому яскравість бліків на екрані ВДТ має не перевищувати 40 кд/м2, а яскравість стелі в разі застосування системи відбитого освітлення 200 кд/м^2.

Показник осліплености у разі використання джерел загального штучного освітлення у виробничих приміщеннях має не перевищувати 20, а показник дискомфорту в адміністративно-громадських приміщеннях має бути не більше за 40.

Необхідно обмежувати нерівномірність розподілу яскравості в полі зору працюючих з ВДТ. При цьому співвідношення яскравостей робочих поверхонь має бути не більшим ніж 3:1, а співвідношення яскравостей робочих поверхонь та поверхонь стін, бладнання тощо - 5:1.

Коефіцієнт пульсації має не перевищувати 5%, що забезпечується застосуванням газорозрядних ламп у світильниках загального та місцевого освітлення з ВЧ ПРА для світильників будь-яких типів. Якщо не має

світильників з ВЧ ПРА, то лампи багатолампових світильників або світильники загального освітлення, розташовані поруч, слід вмикати на різні фази трьохфазної мережі.

Для забезпечення нормованих значень освітленості у приміщеннях з ВДТ ЕОМ та ПЕОМ слід чистити шибки і світильники принаймні двічі на рік і вчасно замінювати лампи, що перегоріли.

З огляду на стрімкий розвиток світлодіодних технологій, у сучасній практиці проектування та модернізації систем штучного освітлення робочих місць з ВДТ ЕОМ, поряд із традиційними люмінесцентними лампами, дозволяється застосування енергоефективних LED-джерел світла. При цьому до світлодіодних світильників висуваються аналогічні жорсткі вимоги щодо якості світлового середовища: колірна температура повинна перебувати в межах комфортного нейтрального або теплого білого спектра ($3000–4500\text{ К}$), а індекс кольоропередачі (R_a) має становити не менше 80. Застосування сучасних LED-панелей із мікропризматичними або опаловими розсіювачами дозволяє досягти необхідної рівномірності світлового потоку та унеможливлює виникнення точкових сліпучих бліків на екранах моніторів.

Для підтримання безпечних і комфортних умов праці адміністрація підприємства або установи зобов'язана забезпечити проведення регулярного інструментального контролю параметрів освітленості. Вимірювання рівня освітленості робочих поверхонь, коефіцієнта пульсації та перевірка показників яскравості мають здійснюватися за допомогою сертифікованих люксметрів і пульсметрів під час атестації робочих місць, а також після кожної реконструкції системи освітлення чи масової заміни ламп. Усі результати замірів фіксуються у відповідних протоколах, а виявлені відхилення від нормативних значень (наприклад, зниження світлового потоку через старіння світлодіодів або деградацію люмінофору) підлягають негайному усуненню шляхом проведення позапланового технічного обслуговування.

ВИСНОВКИ

У кваліфікаційній роботі здійснено проектування, розробку та тестування сучасного вебсайту для туристичної агенції «Сонячні мандри». Створений програмний продукт спрямований на автоматизацію взаємодії між агенцією та клієнтами, покращення представлення туристичних послуг у мережі Інтернет та оптимізацію процесу підбору турів.

В процесі роботи проведено детальний аналіз предметної області й сформовано технічне завдання на розробку. Розроблено структурну схему сайту, діаграму класів, а також блок-схему вибору та взаємодії користувача з туристичними пропозиціями, що дозволило чітко визначити архітектуру системи та логіку її роботи.

На етапі технічного та робочого проектування реалізовано клієнтську та серверну частини вебсайту. Для створення адаптивного, кросбраузерного та динамічного інтерфейсу використано технології HTML5, CSS3 та JavaScript. Функціонал обробки запитів, взаємодії з клієнтами та серверна логіка були реалізовані за допомогою мови програмування PHP, що забезпечило надійність та швидкодію ресурсу.

У спеціальному розділі було проведено перевірку працездатності вебсайту. Особливу увагу приділено захисту персональних даних користувачів та стійкості ресурсу до типових кібератак (XSS, SQL-ін’єкції тощо). Тестування підтвердило повну відповідність розробленого ПЗ вихідним вимогам.

В економічному розділі проведено розрахунок витрат на розробку та впровадження вебсайту. Сформовано таблицю техніко-економічних показників, яка продемонструвала фінансову доцільність проекту, високий потенціал окупності за рахунок залучення нових клієнтів через онлайн-канал та зниження адміністративних витрат турфірми.

У розділі Охорона праці проаналізовано умови праці на робочому місці розробника комп'ютерних систем. Розроблено заходи щодо забезпечення

оптимальних параметрів мікроклімату, освітлення, електробезпеки та ергономіки під час тривалої роботи за комп’ютером згідно з чинними нормами безпеки життєдіяльності.

Створений вебсайт є завершеним, працездатним та економічно обґрунтованим продуктом, який готовий до реального впровадження в операційну діяльність туристичної агенції «Сонячні мандри» для підвищення її конкурентоспроможності на ринку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Date C.J. An Introduction to Database Systems. 8th ed. Boston : Addison-Wesley, 2004. 1013 с.
2. Design Patterns: Elements of Reusable Object-Oriented Software / Gamma E., Helm R., Johnson R. and Vlissides J. Boston : Addison-Wesley Professional, 1994. 395 с.
3. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). RFC 7519 : вебсайт. URL: https://tools.ietf.org/html/rfc7519 (дата звернення: 03.06.2025).
4. Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями. Затверджено Наказом Мінсоцполітики України від 14.02.2018 № 207.
5. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98
6. ДСанПіН 3.3.2.007-98. НПАОП 0.00-1.29-97 «Правила захисту від статичної електрики».
7. ДСанПіН 3.3.6.096-2002 «Державні санітарні норми та правила при роботі з джерелами електромагнітних полів»
8. ДСТУ EN 60950-1:2019 Обладнання інформаційних технологій. Безпека. Частина 1. Загальні вимоги. — К.: ДП «УкрНДНЦ», 2019. (Фундаментальний стандарт для перевірки безпеки комп'ютерної техніки та периферії).
9. ДСТУ EN IEC 62368-1:2022 Обладнання аудіо/відео, інформаційних та комунікаційних технологій. Частина 1. Вимоги щодо безпеки. — К.: ДП «УкрНДНЦ», 2023.
10. Кантелон М., Хартер Т. Node.js в дії. 2-е вид. Київ : Manning Publications, 2018. 480 с.
11. Методичні вказівки для виконання дипломного проекту Напрямок: “ Веб-програмування” URL:https://eguru1.tk.te.ua/pluginfile.php/20316/mod_resource/content/1/Metod_dypl_obsl_2019_1557946788.pdf

12. Офіційна документація Jest. JavaScript Testing Framework : вебсайт. URL: https://jestjs.io/docs/uk/getting-started (дата звернення: 03.06.2025).
13. Офіційна документація React.js: вебсайт. URL: https://react.dev/ (дата звернення: 01.06.2025).
14. Офіційна документація Sequelize ORM : вебсайт. URL: https://sequelize.org/docs/v6/ (дата звернення: 02.06.2025).
15. Річардсон Л., Рубі М. RESTful веб-служби. Проєктування та розробка веб-сервісів в стилі REST. Київ : O'Reilly, 2013. 450 с.
16. Соммервілл І. Інженерія програмного забезпечення. 9-е вид. Київ : Видавнича група BHV, 2015. 960 с.
17. Фленаган Д. JavaScript. Докладний посібник. 7-е вид. Київ : O'Reilly Media, 2021. 725 с.

ДОДАТКИ

Додаток А. Лістінг головної сторінки

```
<!DOCTYPE html>
<html lang="uk">
<head>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Туристична агенція | Твій теплий відпочинок</title
   <!-- Підключення зовнішнього стилю -->
<link rel="stylesheet" href="css/style.css">
   <!-- Додатковий захист: якщо CSS не завантажиться, ці стилі врятують вигляд -->
   <style>
      .logo-container {
         width: 100px !important;
         height: 100px !important;
         margin: 0 auto 15px auto !important;
      }
      .logo-svg {
         width: 70px !important;
         height: 70px !important;
      }
   </style>
</head>
<body>

<header style="display: flex; flex-direction: column; align-items: center; text-align: center;">
   <div class="logo-container" style="background: white; width: 100px; height: 100px; border-radius: 50%; display: flex; align-items: center; justify-content: center; border: 3px solid #F1C40F; box-shadow: 0 4px 15px rgba(0,0,0,0.2);">
      <!-- Додано width та height прямо в SVG -->
      <svg class="logo-svg" width="70" height="70" viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg" style="fill: #D35400;">
         <path d="M12,7L12,7C10.5,7,9.17,7.75,8.41,9H15.59C14.83,7.75,13.5,7,12,7z M2,12c0,4.41,3.59,8,8,8h4c4.41,0,8-3.59,8-8s-3.59-8-8-8h-4C5.59,4,2,7.59,2,12z M12,2c5.52,0,10,4.48,10,10s-4.48,10-10,10S2,17.52,2,12S6.48,2,12,2z M6,12c0,0.55,0.45,1,1,1h10c0.55,0,1-0.45,1-1s-0.45-1-1-1H7C6.45,11,6,11.45,6,12z"/>
         <path d="M12,4c-4.42,0-8,3.58-8,8c0,1.44,0.39,2.79,1.05,3.95l1.45-1.45C6.18,13.77,6,12.91,6,12c0-3.31,2.69-6,6-6s6,2.69,6,6c0,0.91-0.18,1.77-0.5,2.5l1.45,1.45C19.61,14.79,20,13.44,20,12C20,7.58,16.42,4,12,4z"/>
      </svg>
   </div>
   <h1>Сонячні мандри</h1>
   <p>Подорожі, що зігрівають серце</p>
</header>

<nav style="text-align: center; padding: 15px;">
   <a href="destinations.html" style="margin: 0 10px; text-decoration: none; font-weight: bold;">Напрямки</a>
```

```
    <a href="order.html" style="margin: 0 10px; text-decoration: none; font-weight: bold;">Замовити тур</a>
    <a href="about.html" style="margin: 0 10px; text-decoration: none; font-weight: bold;">Про нас</a>
    <a href="reviewc.html" style="margin: 0 10px; text-decoration: none; font-weight: bold;">Відгуки</a>
    <a href="contacts.html" style="margin: 0 10px; text-decoration: none; font-weight: bold;">Контакти</a>
</nav>

<section class="container" id="tours" style="max-width: 1100px; margin: 0 auto; padding: 20px;">
    <h2 style="text-align: center;">Гарячі пропозиції</h2>
    <div class="offers-grid" style="display: flex; flex-wrap: wrap; justify-content: center; gap: 20px;">
        <div class="card" style="max-width: 300px; padding: 20px; border-radius: 10px; box-shadow: 0 4px 10px rgba(0,0,0,0.1); background: white;">
            <img
src="https://images.unsplash.com/photo-1489749798305-
4fea3ae63d43?auto=format&fit=crop&w=400&q=80"
                alt="Марокко" style="width: 100%;
 border-radius: 8px; margin-bottom: 15px;">
            <h3>□ Марокко</h3>
            <p>Загадкові пустелі та пряні аромати ринків Марракеша.</p>
            <p><strong>Від 22 000 грн</strong></p>
        </div>
        <div class="card" style="max-width: 300px;
 padding: 20px;
 border-radius: 10px; box-shadow: 0 4px 10px rgba(0,0,0,0.1);
background: white;">
            <img                                src="https://images.unsplash.com/photo-1586861635167-
e5223aadc9fe?auto=format&fit=crop&w=400&q=80"
                alt="Занзібар" style="width: 100%; border-radius: 8px;
margin-bottom: 15px;">
            <h3>□ Занзібар</h3>
            <p>Білосніжний пісок та неймовірні заходи сонця над океаном.</p>
            <p><strong>Від 55 000 грн</strong></p>
        </div>
    </div>
</section>

<footer style="text-align: center; padding: 30px; margin-top: 20px;">
    <p>📍 м. Тернопіль, вул. Сонячна, 10</p>
    <p>📞 +38 (067) 368-45-12</p>
    <p>&copy; 2026 Туристична агенція "Сонячні мандри"</p>
</footer>

</body>
</html>
```

Додаток В. Лістінг сторінки Напрямки

```
<!DOCTYPE html>
<html lang="uk">
<head>
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Напрямки | Сонячні мандри - Ваші сонячні подорожі</title>
   <style>
      :root {
         --primary-color: #D35400;
         --secondary-color: #E67E22;
         --accent-color: #F1C40F;
         --bg-warm: #FFF5E1;
         --text-dark: #4A2700;
         --card-bg: #FFFFFF;
      }
body { font-family: 'Segoe UI', Arial, sans-serif; margin: 0; line-height: 1.6; color: var(--text-dark);
background-color: var(--bg-warm); display: flex; flex-direction: column;
 min-height: 100vh; }
      header { background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
color: white; padding: 25px 20px; text-align: center; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
      header h1 { margin: 0; font-size: 2.2em; text-transform: uppercase;
 letter-spacing: 2px; }
      nav { background: var(--card-bg); padding: 12px; text-align: center; box-shadow: 0 2px 5px
rgba(0,0,0,0.05); position: sticky; top: 0; z-index: 1000; }
      nav a { text-decoration: none; color: var(--primary-color); font-weight: bold; transition: 0.3s;
margin: 0 15px; font-size: 1.1em; }
      nav a:hover { color: var(--secondary-color); }
      .container { width: 90%; max-width: 1200px; margin: auto;
padding: 50px 0; flex: 1; }
      h2 { color: var(--primary-color); text-align: center; font-size: 2.5em
; margin-bottom: 10px; text-transform: uppercase; letter-spacing: 1px; }
      .section-subtitle { text-align: center; font-size: 1.2em; opacity: 0.8; margin-bottom: 50px;
color: var(--text-dark); }
      .destinations-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
gap: 30px; margin-bottom: 50px; }
card-image-wrapper { position: relative; width: 100%; height: 220px; overflow: hidden; }
      .destination-card img { width: 100%; height: 100%; object-fit: cover;
 transition: transform 0.5s ease; }
      .destination-card:hover img { transform: scale(1.05); }
      .
      .price { font-size: 1.3em; font-weight: bold; color: var(--text-dark); }
      .price-label { font-size: 0.8em; font-weight: normal; display: block;
 color: #777; }
      .btn-card { display: inline-block; background-color: var(--primary-color); color: white
!important; text-decoration: none; padding: 10px 20px; border-radius: 25px;
font-weight: bold; font-size: 0.9em; transition: background-color 0.3s;
text-transform: uppercase; }
      .btn-card:hover { background-color: var(--secondary-color); }
```

```
        footer p { margin: 8px 0; }
        .loading { text-align: center; padding: 50px;
color: var(--primary-color); font-size: 1.2em; }
    </style>
</head>
<body>
<header>
    <h1>Сонячні мандри</h1>
</header>

<nav>
    <a href="index.html">&larr; Головна</a>
    <a href="about.html">Про нас</a>
    <a href="order.html">Замовити тур</a>
</nav>
<section class="container">
    <h2>□ Наші сонячні напрямки</h2>
    <p class="section-subtitle">Оберіть свій ідеальний куточок тепла та релаксу</p>

    <div class="destinations-grid" id="tours-grid">
        <div class="loading">Завантажуємо тури... ✈□</div>
    </div>
    <div class="cta-destinations">
        <h3>Не знайшли напрямок своєї мрії?</h3>
        <p>Наші експерти з радістю розроблять для вас індивідуальний маршрут!</p>
        <a href="order.html" class="btn-large">Замовити консультацію</a>
    </div>
</section>

<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/11.0.0/firebase-app.js";
import { getFirestore, collection, getDocs, orderBy, query } from
"https://www.gstatic.com/firebasejs/11.0.0/firebase-firestore.js";


const DEFAULT_TOURS = [
    { name: '□ Марокко', price: 22000, desc: 'Пориньте у казку тисячі й однієї нічі. Загадкові
пустелі Сахари, колоритні ринки Марракеша та прохолода Атлантичного океану.', img:
'https://images.unsplash.com/photo-1489749798305-
4fea3ae63d43?auto=format&fit=crop&w=600&q=80' },
    { name: '□ Занзібар', price: 55000, desc: 'Райський острів біля берегів Африки. Білосніжний
пісок, бірюзові води Індійського океану та неймовірні заходи сонця.', img:
'https://images.unsplash.com/photo-1586861635167-
e5223aadc9fe?auto=format&fit=crop&w=600&q=80' },
    { name: '🐫 Єгипет', price: 15000, desc: 'Класика сонячного відпочинку. Фантастичний
підводний світ Червоного моря, стародавні піраміди та комфорт готелів "все включено".',
img: 'https://images.unsplash.com/photo-1539768942893-
daf53e448371?auto=format&fit=crop&w=600&q=80' },
    { name: '🐘 Таїланд', price: 48000, desc: 'Країна посмішок та екзотики. Дивовижні храми,
тропічні острови, смачна вулична кухня та незабутній колорит Азії.', img: },
```

```
    { name: '□ Туреччина', price: 18000, desc: 'Поєднання східної гостинності та європейського
сервісу. Чотири моря, унікальна Каппадокія та багата історична спадщина.', img:
'https://images.unsplash.com/photo-1524231757912-
21f4fe3a7200?auto=format&fit=crop&w=600&q=80' }
function renderTours(tours) {
    const grid = document.getElementById('tours-grid');
    if (tours.length === 0) {
        grid.innerHTML = '<div class="loading">Тури незабаром з\'являться!</div>';
        return;
    }
    grid.innerHTML = tours.map(t => `
        <div class="destination-card">
            <div class="card-image-wrapper">
                <img src="${t.img}" alt="${t.name}"
onerror="this.src='https://images.unsplash.com/photo-1506905925346-
21bda4d32df4?auto=format&fit=crop&w=600&q=80'">
            </div>
            <div class="card-content">
                <h3>${t.name}</h3>
                <p>${t.desc}</p>
                <div class="card-footer">
                    <div class="price">
                        <span class="price-label">від</span>
                        ${Number(t.price).toLocaleString('uk-UA')} грн
                    </div>
                    <a href="order.html" class="btn-card">Замовити</a>
                </div>
            </div>
        </div>
    `).join('');
}

async function loadTours() {
    try {
        const q = query(collection(db, 'tours'), orderBy('order'));
        const snapshot = await getDocs(q);
        if (snapshot.empty) {
            renderTours(DEFAULT_TOURS);
        } else {
            const tours = snapshot.docs.map(doc => doc.data());
            renderTours(tours);
        }
    } catch (e) {
        renderTours(DEFAULT_TOURS);
    }

loadTours();
</script>
</body>
</html>
```

Додаток С. Лістінг сторінки Замовити тур

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Замовити тур | Сонячні мандри</title>
    <style>
        :root {
            --primary-color: #D35400;
            --secondary-color: #E67E22;
            --accent-color: #F1C40F;
            --bg-warm: #FFF5E1;
            --text-dark: #4A2700;
        }

        body {
            font-family: 'Segoe UI', Arial, sans-serif;
            margin: 0;
            line-height: 1.6;
            color: var(--text-dark);
            background-color: var(--bg-warm);
            display: flex;
            flex-direction: column;
            min-height: 100vh; /* Щоб футер завжди був внизу */
        }

        /* ГАРМОНІЙНИЙ HEADER (такий же, як на головній) */
        header {
            background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
            color: white;
            padding: 20px;
            text-align: center;
            box-shadow: 0 2px 10px rgba(0,0,0,0.1);
        }

        header h1 {
            margin: 0;
            font-size: 1.8em;
            text-transform: uppercase;
            letter-spacing: 2px;
        }

        /* НАВІГАЦІЯ (спрощена для сторінки замовлення) */
        nav {
            background: white;
            padding: 10px;
            text-align: center;
            box-shadow: 0 2px 5px rgba(0,0,0,0.05);
        }
```

```
nav a {
    text-decoration: none;
    color: var(--primary-color);
    font-weight: bold;
    transition: 0.3s;
}

nav a:hover { color: var(--secondary-color); }

/* ОСНОВНИЙ КОНТЕЙНЕР */
.container {
    width: 85%;
    max-width: 600px; /* Вужчий контейнер для форми */
    margin: auto;
    padding: 40px 0;
    flex: 1; /* Займає доступний простір */
}

.order-box {
    background: white;
    border-top: 5px solid var(--accent-color);
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 4px 15px rgba(211, 84, 0, 0.1);
}

h2 {
    color: var(--primary-color);
    text-align: center;
    margin-top: 0;
}

p.subtitle {
    text-align: center;
    opacity: 0.8;
    margin-bottom: 25px;
}

/* СТИЛІ ФОРМИ */
.form-group {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
    color: var(--text-dark);
}
```

```
    input[type="text"],
    input[type="email"],
    input[type="tel"],
    }

    input:focus, select:focus, textarea:focus {
        outline: none;
        border-color: var(--secondary-color);
    }

    textarea {
        resize: vertical; /* Дозволяє міняти висоту, але не ширину */
        height: 100px;
    }

    /* КНОПКА ЗАМОВИТИ */
    .btn-submit {
        background-color: var(--primary-color);
        color: white;
        border: none;
        padding: 15px 30px;
        font-size: 1.1em;
        font-weight: bold;
        border-radius: 30px;
        cursor: pointer;
        width: 100%;
        transition: background-color 0.3s, transform 0.2s;
        text-transform: uppercase;
        letter-spacing: 1px;
    }

    .btn-submit:hover {
        background-color: var(--secondary-color);
    }

    .btn-submit:active {
        transform: translateY(1px);
    }

    /* ФУТЕР (такий же, як на головній) */
    footer {
        background: var(--text-dark);
        color: var(--bg-warm);
        text-align: center;
        padding: 20px;
        margin-top: 40px;
        font-size: 0.9em;
    }

    footer p { margin: 5px 0; }
</style>
```

```
</head>
<body>

<header>
   <h1>Сонячні мандри</h1>
</header>

<nav>
   <a href="index.html">&larr; Вернутися на головну</a>
</nav>

<section class="container">
   <div class="order-box">
      <h2>□ Оформлення замовлення</h2>
      <p class="subtitle">Заповніть форму, і ми підберемо ідеальний теплий відпочинок для
вас!</p>

      <form action="#" method="post"> <!-- Поки що без обробника -->

         <div class="form-group">
            <label for="name">Ваше Ім'я:</label>
            <input type="text" id="name" name="name" required placeholder="Іван Іванов">
         </div>


         <div class="form-group">
            <label for="destination">Бажаний напрямок:</label>
            <select id="destination" name="destination">
               <option value="" disabled selected>Виберіть країну...</option>
               <option value="morocco">Марокко □</option>
               <option value="zanzibar">Занзібар □</option>
               <option value="other">Інший напрямок (впишіть у коментарях)</option>
            </select>
         </div>

         <div class="form-group">
            <label for="message">Побажання / Коментарі:</label>
            <textarea id="message" name="message" placeholder="Дата, кількість людей,
бюджет..."></textarea>
         </div>
         <button type="submit" class="btn-submit">Надіслати запит</button>
      </form>
   </div>
</section>

<footer>
   <p>&copy; 2026 Туристична агенція "Сонячні мандри"</p>
</footer>

</body>
</html>
```

Додаток Д. Лістінг сторінки панель Адміністратора

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

   <title>Адмін панель | Сонячні мандри</title>

   <style>
        :root { --primary-color: #D35400;
 --secondary-color: #E67E22;
 --accent-color: #F1C40F;
 --bg-warm: #FFF5E1;
 --text-dark: #4A2700;
 --green: #27AE60;
 --red: #E74C3C; }
        * { box-sizing: border-box; margin: 0; padding: 0; }

     body { font-family: 'Segoe UI', Arial, sans-serif;
 background: var(--bg-warm);
color: var(--text-dark);
min-height: 100vh; display: flex;
 flex-direction: column; }

       header { background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
color: white; padding: 25px 20px; text-align: center; }
        header h1 { font-size: 2em; text-transform: uppercase;
letter-spacing: 2px; margin-bottom: 5px; }
        header p { opacity: 0.9; font-style: italic;
 }
        nav { background: white; padding: 12px; text-align: center;
box-shadow: 0 2px 5px rgba(0,0,0,0.05);
position: sticky; top: 0; z-index: 1000; }
        nav a { text-decoration: none;
color: var(--primary-color);
 font-weight: bold; margin: 0 15px;
transition: 0.3s; }

}
        .container { width: 90%; max-width: 1000px;
 margin: 40px auto; flex: 1; }
        .form-card { background: white; border-radius: 15px; padding: 30px;
margin-bottom: 30px; box-shadow: 0 4px 15px rgba(211,84,0,0.1);
 border-top: 4px solid var(--accent-color); }
        .form-card h2 { color: var(--primary-color); font-size: 1.3rem;
```

```
 header { background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
color: white; padding: 25px 20px; text-align: center; }
        header h1 { font-size: 2em; text-transform: uppercase; letter-spacing: 2px;
 margin-bottom: 5px; }
        header p { opacity: 0.9; font-style: italic;
 }
        nav { background: white; padding: 12px; text-align: center; box-shadow: 0 2px 5px
rgba(0,0,0,0.05); position: sticky; top: 0; z-index: 1000; }
        nav a { text-decoration: none;
color: var(--primary-color);
 font-weight: bold; margin: 0 15px; transition: 0.3s; }

  </style>
</head>
<body>
<div id="login-wrapper" class="login-wrapper">
    <div class="login-card">
        <div class="login-logo">☀□</div>
        <h2>Сонячні мандри</h2>
        <p>Увійдіть щоб керувати турами</p>
        <div class="form-group"><label>Email</label><input type="email" id="login-email"
placeholder="your@email.com"></div>
        <div class="form-group"><label>Пароль</label><input type="password" id="login-pass"
placeholder="••••••••" onkeydown="if(event.key==='Enter') doLogin()"></div>
        <button class="login-btn" onclick="doLogin()">Увійти в панель →</button>
        <p class="login-error" id="login-error"></p>
    </div>
</div>
<div id="admin-panel" style="display:none;
 flex-direction:column; min-height:100vh;">
    <header><h1>☀□ Сонячні мандри</h1><p>Панель керування турами</p></header>
    <nav>
        <a href="index.html">← На сайт</a>
        <a href="destinations.html">Напрямки</a>
        <button class="logout-btn" onclick="doLogout()">Вийти</button>
   </nav>
    <div class="container">
        <div class="form-card">

       <h2 id="form-title">✚ Додати новий тур</h2>

           <input type="hidden" id="edit-id" value="">
function showToast(msg, color) {
    const el = document.getElementById('toast');
    el.textContent = msg; el.style.background = color || '#27AE60';
    el.classList.add('show');
    setTimeout(() => el.classList.remove('show'), 2500);
</script>
</body>
</html>
```