

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: «Web-орієнтована система підбору походу в Карпати»

Здобувачки групи ІТ.м-23 Захарченко Анна Олександрівна
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Анна ЗАХАРЧЕНКО
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник _____ к.т.н., доц. Вікторія АНТИПЕНКО _____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ
_____ С. М. Ващенко

«__» _____ 2022 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Захарченко Анні Олександрівні

(прізвище, ім'я, по батькові)

1 Тема проєкту Web-орієнтована система підбору походу в Карпати

затверджена наказом по університету від «08» листопада 2023 р. № № 1249-VI

2 Термін здачі студентом закінченого проєкту «__» _____ грудня _____ 2022 р.

3 Вхідні дані до проєкту доступні туристичні маршрути, перелік визначних місць, перелік найвищих точок над рівнем моря.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі, методи дослідження, проектування web-орієнтованої системи підбору походу в Карпати, розробка web-орієнтованої системи, тестування web-орієнтованої системи.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність проблеми, мета дипломної роботи, задачі проєкту, функціональні вимоги до системи, структурно-функціональне моделювання, практична реалізація, демонстрація роботи, тестування, оприлюднення результатів роботи, висновки.

6 Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проєкту	Термін виконання етапів проєкту	Примітка
1	Планування робіт	до 02.11.2022	
2	Проведення аналізу предметної області	до 10.11.2022	
3	Проведення структурно-функціонального моделювання	до 15.11.2022	
4	Реалізація структури web-орієнтованої системи	до 20.11.2022	
5	Розробка функціональної частини web-орієнтованої системи	до 25.11.2022	
6	Тестування роботи web-орієнтованої системи	до 05.12.2022	
7	Оформлення та здача пояснювальної записки та файлів розробленого проєкту	до 10.12.2022	

Магістрант _____

Анна ЗАХАРЧЕНКО

Керівник роботи _____

к.т.н., доц. Вікторія АНТИПЕНКО

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Web-орієнтована система підбору походу в Карпати».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 42 найменувань, 2 додатків. Загальний обсяг роботи – 104 сторінки, у тому числі 49 сторінок основного тексту, 5 сторінок списку використаних джерел, 50 сторінок додатків.

Актуальність даної роботи полягає в тому, що туризм стає все більш важливим елементом нашого життя. Сучасні умови вплинули на його розвиток в Україні. Для досягнення більшого потенціалу туризму в західній Україні, актуальним є застосування сучасних ІТ-рішень. Тому нагальною є розробка web-орієнтованої системи для підбору походу в Карпати. Її використання сприятиме покращенню доступності та підвищенню якості надання туристичних послуг в цьому регіоні.

Мета роботи – це розробка web-орієнтованої системи підбору походу в Карпати. Для її досягнення було проведено дослідження предметної області та продуктів аналогів. Визначено задачі, функціональні вимоги до web-орієнтованої системи підбору походу в Карпати та засоби для її створення. Виконано структурно-функціональне моделювання. Здійснено програмну реалізацію запропонованої web-орієнтованої системи. Проведено тестування даного програмного продукту

Результатом виконаної роботи є реалізована web-орієнтована системи підбору походу в Карпати. Її використання зменшить час та зусилля при плануванні подорожі в цей регіон.

Ключові слова: WEB-ОРІЄНТОВАНА СИСТЕМА, ПОХІД, ТУРИСТИЧНИЙ МАРШРУТ, КАРПАТИ, ОПИТУВАННЯ, ДЕРЕВО РІШЕНЬ, REACT BOOTSTRAP, NODE.JS.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Огляд останніх досліджень і публікацій.....	10
1.2 Аналіз програмних продуктів-аналогів	11
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	20
2.2 Методи дослідження	22
2.3 Вибір технологій	23
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ	25
3.1 Діаграми нотації IDEF0.....	25
3.3 Дерево рішень.....	27
3.3 Діаграма Use Case.....	30
4 РОЗРОБКА WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДБОРУ ПОХОДУ В КАРПАТИ.....	32
4.1 Архітектура web-орієнтованої системи	32
4.2 Розробка бази даних.....	33
4.3 Програмна реалізація web-орієнтованої системи	34
4.4 Демонстрація роботи web-орієнтованої системи підбору походу в Карпати	38
4.5 Тестування web-орієнтованої системи підбору походу в Карпати	45
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А	56
ДОДАТОК Б	70

ВСТУП

Актуальність. У сучасному світі туризм все більше стає невід’ємною частиною нашого життя. Особливо після пандемії більшість країн почало його активізувати для відновлення економіки. І Україна не стала винятком. Завдяки своїй різноманітній природі та культурній спадщині наша країна має великий потенціал стати центром туризму, який приваблює туристів з усього світу. Однак сучасні умови війни поки що не дозволяють це робити масштабно з залученням іноземців. Проте внутрішні процеси в Україні своєрідно позитивно вплинули на розвиток національного туризму. Наприклад, обмеження виїзду за кордон певної частини населення під час воєнного стану. Це спонукало українців все більше досліджувати різноманітні куточки нашої мальовничої країни й відкривати для себе нові цікаві міста та місця. І, звісно, Карпати зайняли топ місце серед туристичних вподобань. Це один із найкрасивіших регіонів України, який пропонує чудові можливості для піших прогулянок, цікаві походи в гори та пригоди на природі.

Однак для досягнення більшого потенціалу туризму в Карпатах, необхідним є застосування сучасних інформаційних технологій (ІТ) і рішень. Це дозволить популяризувати його серед нашого населення, а в перспективі – і у всьому світі. Тому у цьому контексті актуальним є розробка web-орієнтованої системи для підбору походів в Карпати з можливістю їхнього придбання. Її використання сприятиме покращенню доступності та підвищенню якості надання туристичних послуг в даному регіоні.

Метою даної роботи є розробка web-орієнтованої системи для підбору походу в Карпати з можливістю його придбання онлайн. Робота спрямована на створення інноваційного інструмента, який забезпечить користувачів всебічною інформацією про доступні маршрути, а також надасть можливість забронювати

та/або придбати похід, використовуючи мережу Інтернет. Застосування даної розробки дозволить мінімізувати зусилля потенційних клієнтів при плануванні та організації подорожей в гори, що тільки покращить їх туристичний досвід в Карпатах.

Для досягнення поставленої мети необхідно виконати такі задачі:

- проаналізувати предметну область;
- виконати огляд останніх публікацій;
- провести дослідження програмних продуктів-аналогів;
- визначити технології для розробки web-орієнтованої системи;
- виконати проектування структури web-орієнтованої системи;
- реалізувати структуру та функціональні можливості web-орієнтованої системи;
- здійснити тестування одержаного програмного продукту (ПП).

Об’єктом дослідження є туристичний ринок в Карпатах, зокрема процес надання послуг при організації походу в гори.

Предметом дослідження є web-орієнтована система підбору походу в Карпатах.

Практичне значення. Скорочення часу та мінімізація зусиль користувача при організації походу в Карпати. Використання представленої web-орієнтованої системи сприятиме збільшенню туристичного потоку в даному регіоні, покращенню якості обслуговування мандрівників та розвитку місцевої travel індустрії. Вона також має перспективи стати важливим інструментом для популяризації Карпатських гір як серед населення України, так й за кордоном. У свою чергу, це допоможе зростанню місцевої економіки та підвищенню якості життя в регіоні.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Карпати є досить популярним туристичним напрямком, зокрема для любителів активного відпочинку. Щороку вони приймають тисячі відвідувачів з України та інших країн. Туристам пропонується великий асортимент послуг, серед яких почесне місце займають походи в гори. Однак для їх підбору й організації необхідно враховувати безліч факторів, наприклад, таких:

- складність маршруту;
- тривалість походу;
- пора року;
- фізична підготовка туристів;
- бюджет.

Це може бути досить складним завданням, особливо для туристів-початківців.

У [1] автори досліджують геологічні та геоморфологічні об'єкти Бескидського регіону Українських Карпат для подальшого створення геотуристичних маршрутів. Аналізується їх привабливість для мандрівників, використовуючи розроблену методику оцінки. На основі цього були запропоновані нові геотуристичні пішохідні, автобусні та мотоциклетні маршрути, включаючи описані геологічні та геоморфологічні визначні пам'ятки зазначених геокультурних регіонів Бескидів.

Робота [2] спрямована на дослідження впливу рекреаційної діяльності на природне середовище Українських Карпат. Особлива увага приділяється природним комплексам національних парків даного регіону. Автори надають

оцінку рекреаційної дигресії та активації ерозійних процесів на туристичних маршрутах. Вони також пропонують комплекс організаційних, управлінських та інженерних заходів, які дозволять усім досліджуваним маршрутам забезпечити сталий розвиток та зменшити негативний вплив на природне середовище.

У [3-12] описується алгоритм розробки пішохідного маршруту, надано поради та рекомендації які речі потрібно з собою брати у похід і як до нього готуватися, представлено правила поведінки у разі виникнення надзвичайних ситуацій.

Аналіз останніх публікацій показав, що тема організації походів у Карпати є нагальною та активно вивчається сьогодні. У цих дослідженнях розглядаються різні аспекти організації походів. Наприклад, аналіз популярності різних маршрутів, розробка методів оцінки їх складності та рекомендацій щодо вибору спорядження, аналіз правил безпеки подорожей в гори тощо.

Проаналізовані результати досліджень є корисними та важливими при розробці web-орієнтованої системи з підбору походу в Карпати. Використання такого програмного продукту допоможе туристам підібрати відповідний похід, враховуючи їхні індивідуальні потреби та можливості.

1.2 Аналіз програмних продуктів-аналогів

Для розробки web-орієнтованої системи з підбору походу в Карпати необхідно провести аналіз програмних продуктів-аналогів. Головним завданням при цьому є дослідження їхніх сильних і слабких сторін. У свою чергу це допоможе у визначенні вимог до створення як структури даної web-орієнтованої системи, так і її функціональних можливостей.

Для порівняльного аналізу було відібрано три web-орієнтовані системи для підбору походів в Карпати. А саме: KULUAR.UA [13], Mount Day [14] та Karpaty.ua [15].

Онлайн сервіс KULUAR.UA пропонує різноманітні походи в Карпатах від 3 до 12 днів, із рюкзаком і без, для новачків і бувалих, кожні вихідні.

До основних функцій даного ресурсу відносяться такі:

- пошук походів (рис. 1.1): користувачі мають можливість дослідити доступні варіанти подорожей за різними параметрами, наприклад, такими, як місце, тривалість, рівень складності тощо;
- ознайомлення з даними про походи (рис. 1.2-1.3): кожна подорож має сторінку з детальною інформацією про маршрут, тривалість, вартість, рівень складності та інші деталі;
- здійснення прокату спорядження: даний онлайн сервіс пропонує послуги прокату приладдя для походів;
- перегляд розкладу програм: ця web-орієнтована система має графік заходів, де користувачі можуть переглянути інформацію щодо майбутніх походів, які плануються.

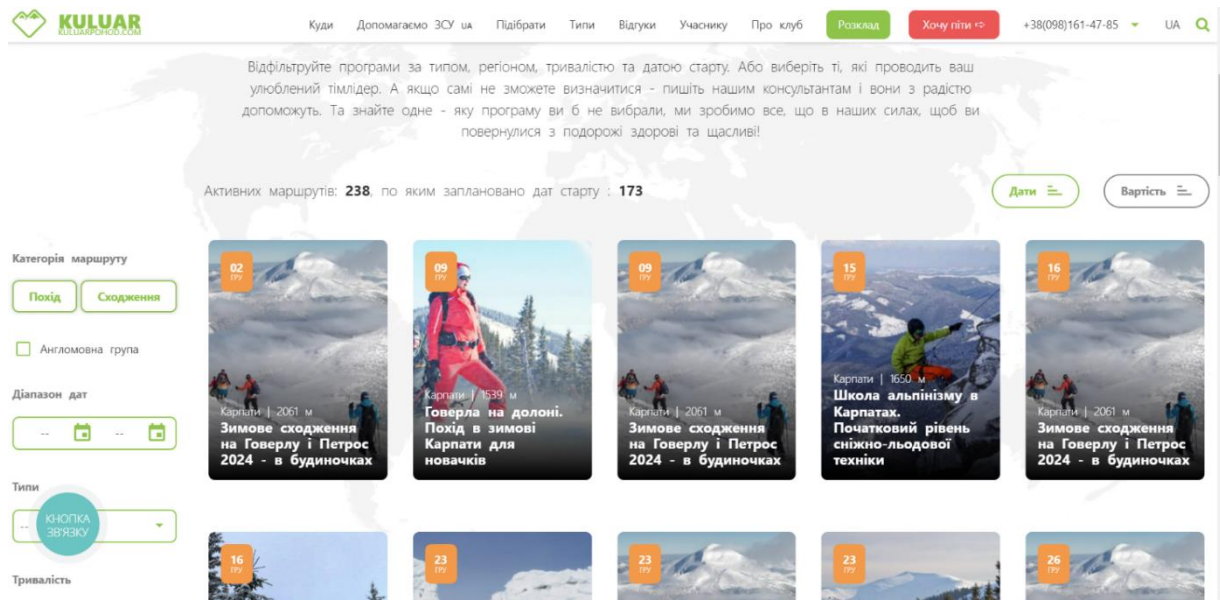
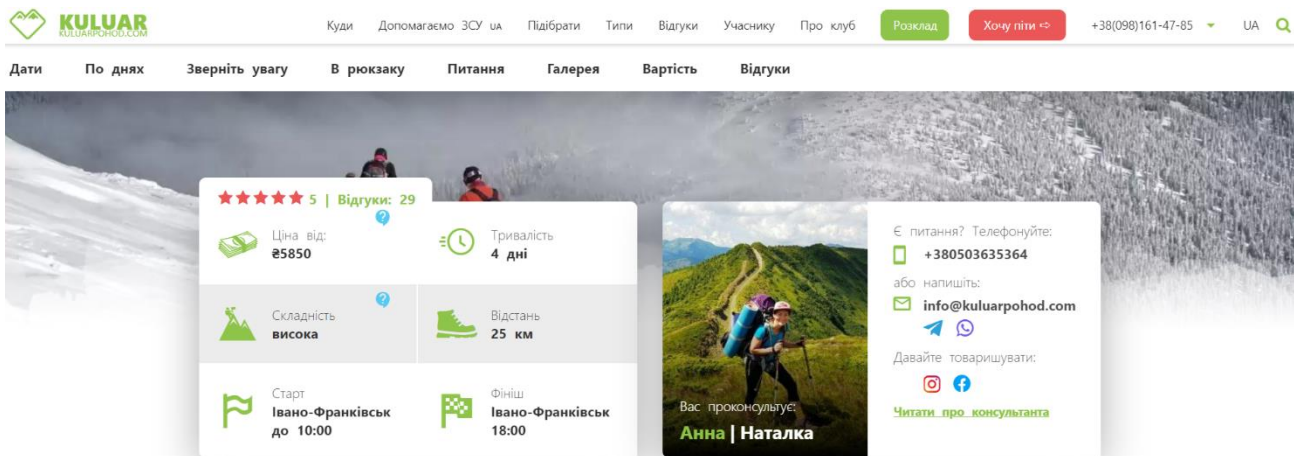


Рисунок 1.1 – Сторінка про походи та сходження в Карпати на сервісі KULUAR

Джерело:[13]



КОРОТКО ПРО ЗИМОВЕ СХОДЖЕННЯ НА ГОВЕРЛУ І ПЕТРОС

Напевно, більшість з вас вже побували на найвищій горі українських Карпат – на всім відомій і

Рисунок 1.2 – Інформація про тур на сервісі KULUAR

Джерело:[13]



Рисунок 1.3 – Особливості, які надає сервіс KULUAR

Джерело: [13]

Після аналізу даної web-орієнтованої системи можна дійти висновків, що хоча, на перший погляд вона має багатий функціонал, цей онлайн ресурс все ж не містить такі важливі елементи, як інформацію про погоду, інтерактивну мапу, а також не надає можливості самостійно організувати похід, а лише придбати готовий тур.

Сайт Mount Day пропонує туристичні походи в гори з досвідченим провідником. Цікаві маршрути по Карпатах від 1 до 9 днів. Похід на Говерлу, Черногору, Горгани, Синевир тощо, а також тури вихідного дня в Карпати.

Основні функції даного онлайн ресурсу є такими:

- вибір походів (рис. 1.4): серед великої кількості маршрутів користувачі мають можливість обрати той, який більше подобається;
- ознайомлення з даними про походи (рис. 1.5): кожен тур містить сторінку з детальною інформацією про маршрут, тривалість та вартість;
- здійснення прокату спорядження: даний сайт пропонує послуги прокату приладдя для походів;

– перегляд розкладу програм: де цей онлайн сервіс надає графік заходів, де користувачам доступна для перегляду інформація щодо наступних походів.

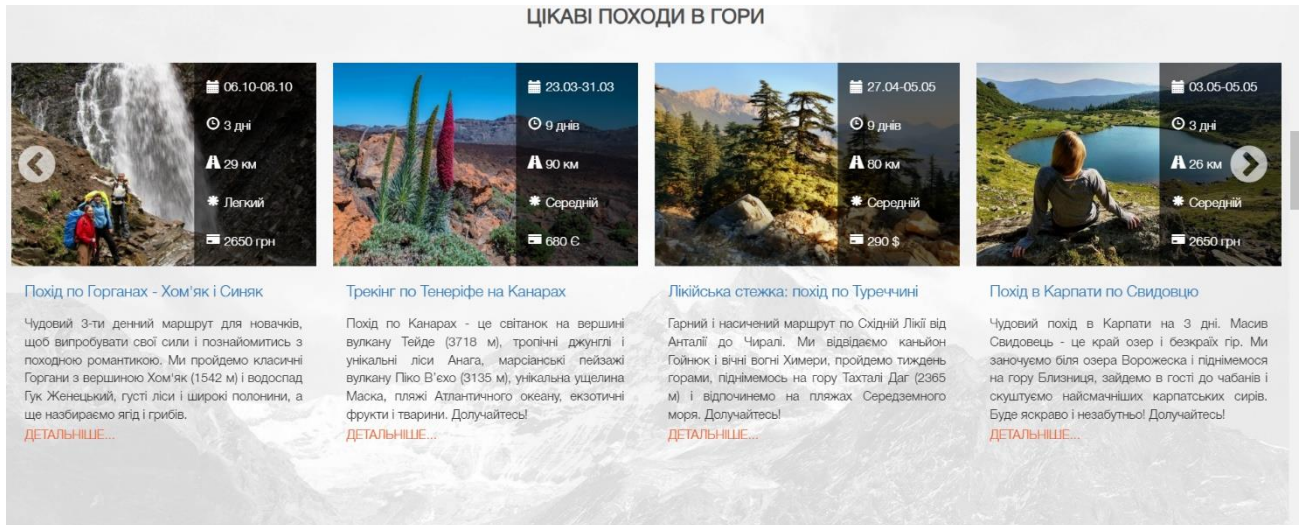


Рисунок 1.4 – Інформація про походи на сайті Mount Day

Джерело: [14]

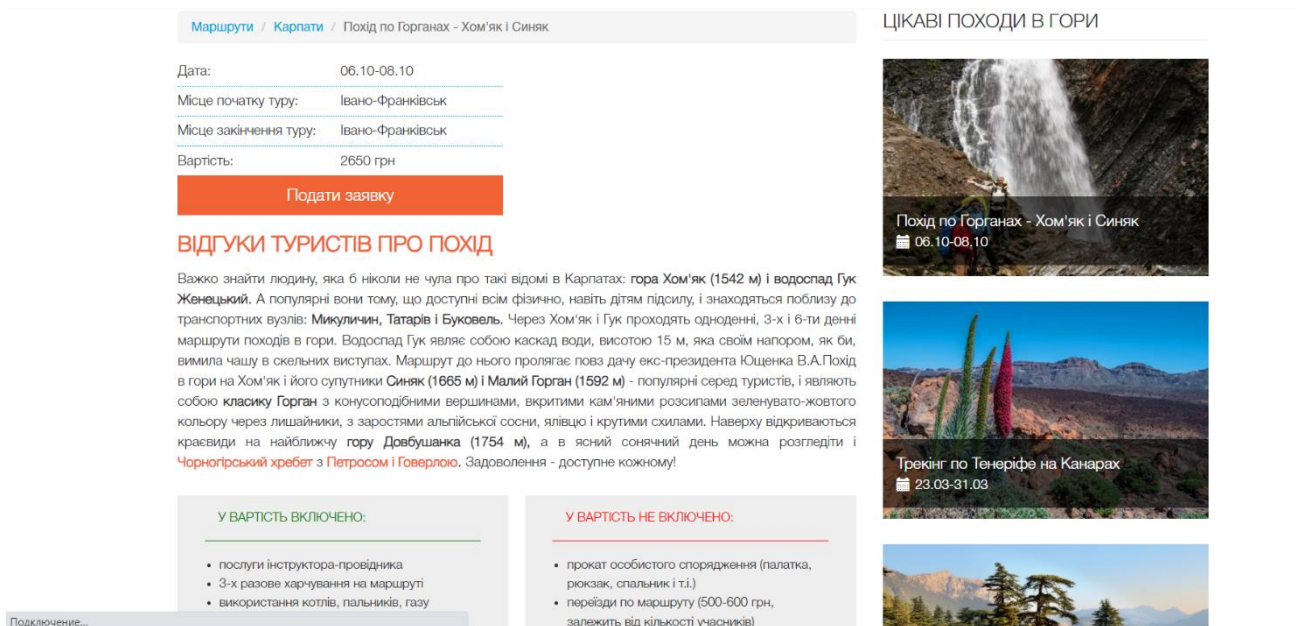


Рисунок 1.5 – Сторінка з детальною інформацією про похід на сайті Mount Day

Джерело: [14]

Проаналізувавши дану web-орієнтовану систему можна зробити висновок, що вона надає менший функціонал, ніж попередній продукт-аналог. Наприклад, відсутній фільтр походів. Це ускладнює пошук бажаного туру. Оскільки даний онлайн ресурс є комерційним проєктом, то неможливо спланувати похід самостійно.

GUIDE.KARPATY.UA є детальним путівником по Карпатах. Цей сайт пропонує різні категорії відпочинку, включаючи туристичні маршрути для пішохідного та велотуризму.

Його основні функції є наступними:

- доступ до даних про походи (рис. 1.6): кожен тур має окрему сторінку з детальною інформацією про подорож;
- ознайомлення з туристичними маршрутами (рис. 1.7): сайт надає відвідувачам інформацію про різні тури для пішохідного та велотуризму в Карпатах;
- перегляд даних про водні ресурси: даний сервіс містить інформацію про річки, озера, водоспади Карпат;
- ознайомлення з можливостями для екстремального відпочинку: цей сайт надає інформацію про заходи для не готельного відпочинку в Карпатах;
- доступ до даних про напрямки: представлений онлайн ресурс надає відомості про різні варіанти для відпочинку в Карпатах, включаючи інформацію про кращі місця для відвідування, визначні пам'ятки та оренду більш комфортного житла.

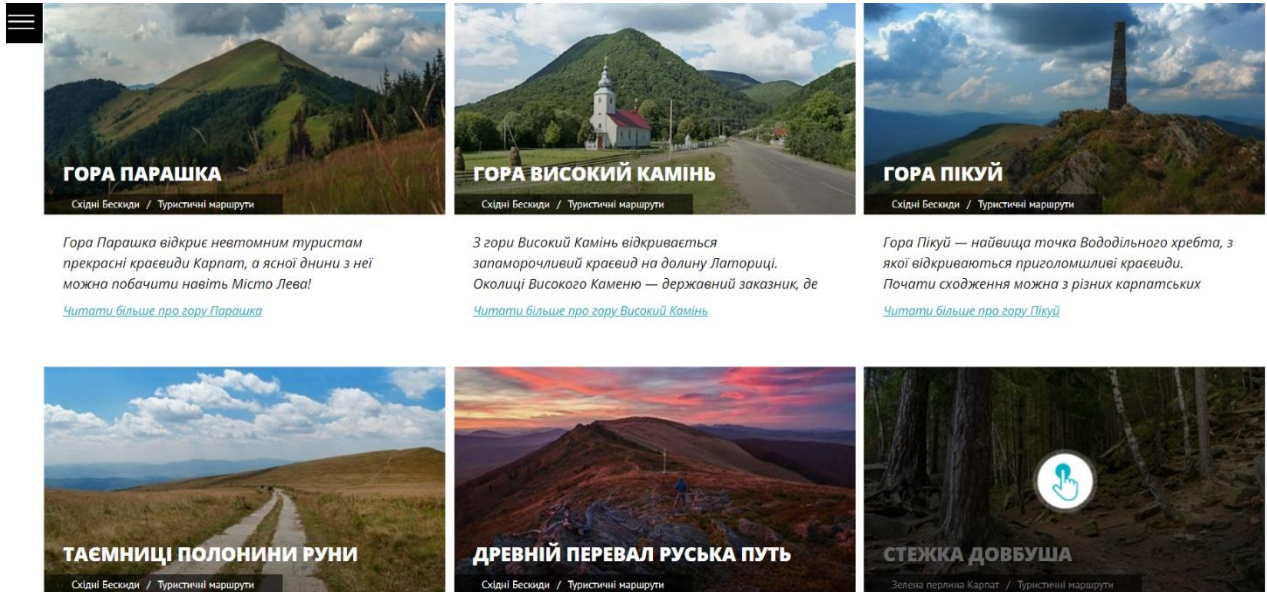



Рисунок 1.6 – Інформація про походи на сайті GUIDE.KARPATY.UA

Джерело: [15]



Гора Парашка



Гора Парашка відкриє невтомним туристам прекрасні краєвиди Карпат, а ясної днини з неї можна побачити навіть Місто Лева!

🔄 18 квіт. 2018 👁 70531

Одна з гір, що височіють над [містом Сколе](#) Львівської області, має назву Парашка. За переказами, під час бою, в якому загинув князь Святослав Володимирович, на цьому місці вбили і його дружину (за іншими джерелами — доньку) Параску.

Прямих доказів тому немає, але легенда далі собі живе. За тим і назва.

Між іншим, Парашка — найвища гора однойменного хребта у Сколівських Бескидах. Ясної погоди її прекрасно видно з Міста Лева, не зважаючи на відстань не менше 90 км.

Рисунок 1.7 – Інформація про певний маршрут

на сайті GUIDE.KARPATY.UA

Джерело: [15]

Аналіз даної web-орієнтованої системи показав, що цей програмний продукт має досить скупий функціонал. Окрім інформації про маршрут, а саме загального його опису, більше даних немає.

Результати проведеного порівняльного аналізу вищезазначених продуктів-аналогів представлені у таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця продуктів-аналогів

Характеристика/Сайт	KULUAR	Mount Day	GUIDE.KARPATY.UA
Ознайомлення зі складністю маршруту	+	+	–
Ознайомлення з часом	+	+	–
Ознайомлення з відстанню	+	+	–
Перегляд маршруту	+	+	+
Перегляд інтерактивної мапа	–	–	–
Перегляд погоди	–	–	–
Ознайомлення з необхідним спорядженням	+	+	–
Огляд цікавих місць на маршруті	–	–	+
Перегляд правил безпеки			–
Перегляд меню	+	–	–

Продовження табл. 1.1

Характеристика/Сайт	KULUAR	Mount Day	GUIDE.KARPATY.UA
Знайомство з підготовкою до тура	–	–	–
Перегляд довідника	–	–	–
Онлайн підбір тура	+	–	–
Можливість придбання	+	+	–

Джерело: побудовано автором

Проаналізувавши наведені у таблиці 1.1 результати, можна виокремити їхні переваги – актуальні та корисні функції, які можна використати у власному проєкті. А також недоліки, які варто подолати. Отже, запропонована web-орієнтована система має містити всі ті функції для зручного вибору походу в Карпати, а також можливості самостійного планування свого туру, та, за потреби, його придбання.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Мета даної роботи заключається в розробці web-орієнтованої системи для підбору походу в Карпати з можливістю його придбання онлайн. Споживачам буде представлено інноваційний інструмент, застосування якого забезпечить їх інформацією про доступні маршрути, а також дозволить здійснити бронювання та/або придбати відповідний тур через мережу Інтернет. Використання даної web-орієнтованої системи зменшить зусилля потенційних клієнтів при плануванні та організації гірських походів. Це, у свою чергу, покращить їхній туристичний досвід в Карпатах.

Розроблювана web-орієнтована система повинна містити наступні функції:

- надання інформації про маршрут: складність, час його проходження та відстань;
- представлення переліку цікавих місць або пам'яток на маршруті;
- надання інформації про проноз погоди на обрані дати походу;
- доступ до інтерактивної мапи маршруту;
- можливість придбання походу;
- надання інформації про необхідне спорядження;
- перегляд запропонованого меню під час походів;
- надання інформації про підготовку до походу;
- перегляд довідника з корисними номерами різних служб;
- надання інформації про прихисток у разі негоди;
- використання форми зворотного зв'язку;
- підбір походу за допомогою опитування.

Для досягнення мети даного проєкту необхідно виконати наступні задачі:

- проаналізувати предметну область: вивчення основних аспектів гірського туризму в Карпатах, включаючи типи маршрутів, особливості планування походів та основні виклики, з якими стикаються мандрівники;
- виконати огляд останніх публікацій: проведення оцінки актуальних досліджень, які пов'язані з гірським туризмом в Карпатах для розуміння поточного стану даного напрямку та його сучасні тенденції;
- провести дослідження програмних продуктів-аналогів: здійснити аналіз існуючих web-додатків і сервісів, які пропонують подібні послуги, для розуміння їхніх сильних і слабких сторін; на їх основі розробити відповідні вимоги до функціональних можливостей власного програмного продукту;
- визначити технології для розробки web-орієнтованої системи: обрати необхідні та відповідні для процесу створення даного ПП технології та інструменти;
- виконати проектування структури web-орієнтованої системи: розробити детальний проєкт web-орієнтованої системи, включаючи архітектуру, дизайн інтерфейсу та базу даних (БД);
- реалізувати структуру та функціональні можливості web-орієнтованої системи;
- здійснити тестування одержаного програмного продукту (ПП).

Дана робота спрямована на розробку інноваційного онлайн інструмента для скорочення часу та мінімізації зусиль потенційних користувачів при організації гірських походів в Карпати. Із врахуванням результатів аналізу предметної області та останніх досліджень, прийнято рішення розробити web-орієнтовану систему зі зручним, інтуїтивно-зрозумілим та інформативним інтерфейсом, забезпечуючи позитивний досвід туристів протягом використання даного ПП при плануванні, виборі та придбанню походів у Карпати.

Під час формулювання мети проєкту було проведено планування його робіт, яке детально описано у Додатку А.

2.2 Методи дослідження

Методи дослідження – це стратегії, техніки та процедури, які використовуються для збору та аналізу даних. Вони важливі для розуміння того, як отримуються нові знання або перевіряються існуючі.

Методи наукових досліджень є наступними:

- загальнонаукові (методи емпіричного дослідження, методи теоретичного дослідження та загальні методи);

- спеціальні методи (застосовуються в конкретній галузі науки) [16].

У даній кваліфікаційній роботі було використано такі методи:

- аналіз;
- індукція;
- дедукція;
- моделювання;
- дерево рішень [17].

Аналіз – метод, який дозволив дослідити продукти-аналоги шляхом вивчення їхніх характеристик, властивостей та взаємозв'язків.

Індукція – метод, який був використаний на етапі збору даних. А також для їх аналізу. Була досліджена інформація про різні гірські походи в Карпатах. На основі окремих фактів створено безпечні маршрути, визначено їхню складність і тривалість.

Метод дедукції було застосовано на етапі проектування та розробки web-орієнтованої системи. Використано загальні принципи дизайну інтерфейсу користувача (UI) [18, 19] та взаємодії користувача (UX) [18, 19] для створення конкретних елементів даного ПП. Також застосовано певні закономірності та принципи програмування для реалізації конкретних функцій даного ПП.

Метод моделювання також було використано на етапі проєктування та розробки web-орієнтованої системи, оскільки дозволяє розробити спрощену репрезентацію системи, яку можна використовувати для вивчення її поведінки та властивостей. Це включає створення моделей даних, процесів або інтерфейсу користувача.

Дерево рішень було використано на етапі проєктування роботи даної web-орієнтованої системи для розробки алгоритму опитування для підбору походу в Карпати.

Узагальнюючи, можна зробити висновок, що використання вищезазначених методів надало можливість глибше вивчити та проаналізувати предмет дослідження, а також забезпечило науковий підхід до розв'язання поставлених задач для досягнення мети даного проєкту.

2.3 Вибір технологій

Для реалізації даної web-орієнтованої системи було обрано мову програмування JavaScript [20]. Вона є досить популярною сьогодні, потужною та багатофункціональною, яка всесторонньо підходить для web-розробки.

Бібліотека React [21-23] була обрана для побудови користувацьких інтерфейсів. Вона також є сучасною, високопродуктивною та легкодоступною. React дозволяє створювати динамічні та інтерактивні користувацькі інтерфейси з застосуванням різних компонентів.

Бібліотека React-Bootstrap [24, 25] була обрана для швидкого створення відповідного дизайну. Даний інструмент є набором певних компонентів, які мають сучасне та інтуїтивно зрозуміле оформлення. Вони дозволяють створювати web-сторінки, які їм відповідають.

Для створення дизайну [26] інтерфейсу були обрані Figma [27] та Canva [28]. Перше є програмним забезпеченням, яке дозволяє розробляти прототипи web-додатків. Canva є онлайн-платформою для створення візуального контенту. Вона дозволяє створювати оформлення web-додатків без необхідності знань у галузі web-дизайну.

Для фільтрації даних було обрано Node.js [29]. Дана платформа для створення серверних додатків, яка дозволяє розробляти їх високопродуктивними та масштабованими. Для цього Node.js також дозволяє використовувати JavaScript.

Для реалізації БД було обрано систему управління базою даних (СУБД) MySQL [30]. Вона є надійною та широко застосовується сьогодні. MySQL можна використовувати безкоштовно завдяки відкритому коду.

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ

Наступним кроком є виконання етапу моделювання та проєктування представленої web-орієнтованої системи. Були розроблені діаграми нотації IDEF0 [31] з описаними послідовностями процесу підбору походу, та діаграма Use-Case [32] з демонстрацією наявних функцій, які користувач може використовувати, застосовуючи даний програмний продукт.

3.1 Діаграми нотації IDEF0

IDEF0 – це методологія графічного опису систем і процесів діяльності організації як безлічі взаємозалежних функцій.

Основні елементи діаграми в нотації IDEF0 є такими:

- блоки, у вигляді яких представлені процеси, функції, операції, дії (в залежності від ступеня деталізації);
- стрілки, у вигляді яких на діаграмі відображають інформаційні та матеріальні ресурси, пов'язані з функціями.

На рисунку 3.1 представлена контекстна діаграма процесу підбору походу в Карпати.

Процес підбору походу в Карпати містить такі дані:

- вхідні дані – запит користувача на підбір туру за допомогою опитування, запит користувача на придбання туру, запит користувача на пошук туру.

- вихідні дані – варіанти походів за результатами опитування, зв'язок менеджера з користувачем для оформлення походу, перелік варіантів походів за бажаними параметрами;
- управління – правила поведінки під час гірського походу, перелік походів, алгоритм придбання походів;
- механізми – web-орієнтована система, користувач, база даних та апаратне забезпечення.



Рисунок 3.1 – Контекстна діаграма нотації IDEF0

Джерело: побудовано автором (знімок з екрану)

Діаграма декомпозиції першого рівня вищезазначеного процесу підбору походу в Карпати представлено на рисунку 3.2.

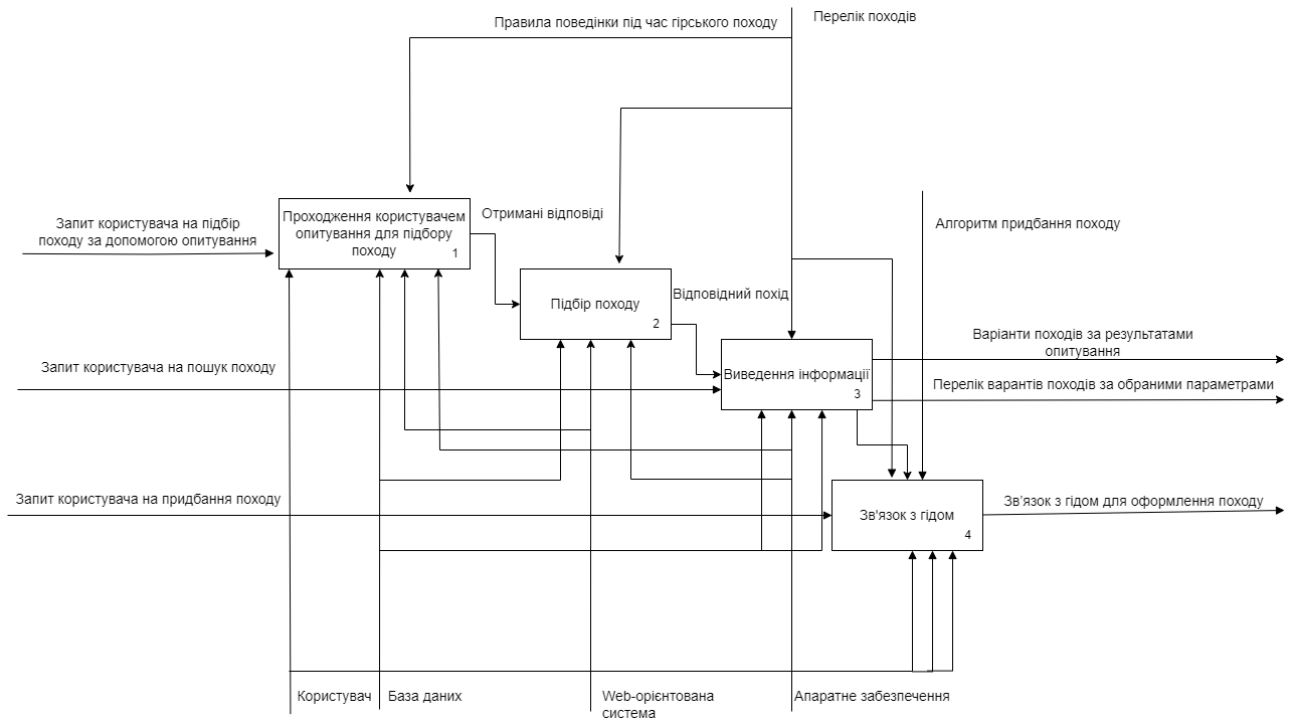


Рисунок 3.2 – Діаграма декомпозиції першого рівня

Джерело: побудовано автором (знімок з екрану)

3.3 Дерево рішень

«Дерево рішень» включає в себе набір вузлів та гілок, які виходять з них. Гілки дерева відображають суб'єктивні та об'єктивні оцінки можливих подій. Проходячи вздовж них, оцінюється кожен шлях, зазвичай на основі ймовірностей, і з усіх можливих оцінок вибирається варіант дій з найменшою ймовірністю. При цьому кожен варіант отримує кількісну оцінку.

Під час побудови дерева використовуються два типи гілок. Це гілка рішень і гілка результатів. Перша передбачає прийняття певного рішення. Вона позначається у вигляді вузла з гілками, які відходять від нього (рішеннями). Гілка подій малюється тоді, коли зовнішні фактори визначають, яка з можливих випадкових подій відбудеться [33].

Було розроблено алгоритм (рис. 3.3), використовуючи дерево рішень, в якому користувач, пройшовши опитування, отримає рекомендований похід у Карпати за результатами наданих відповідей.

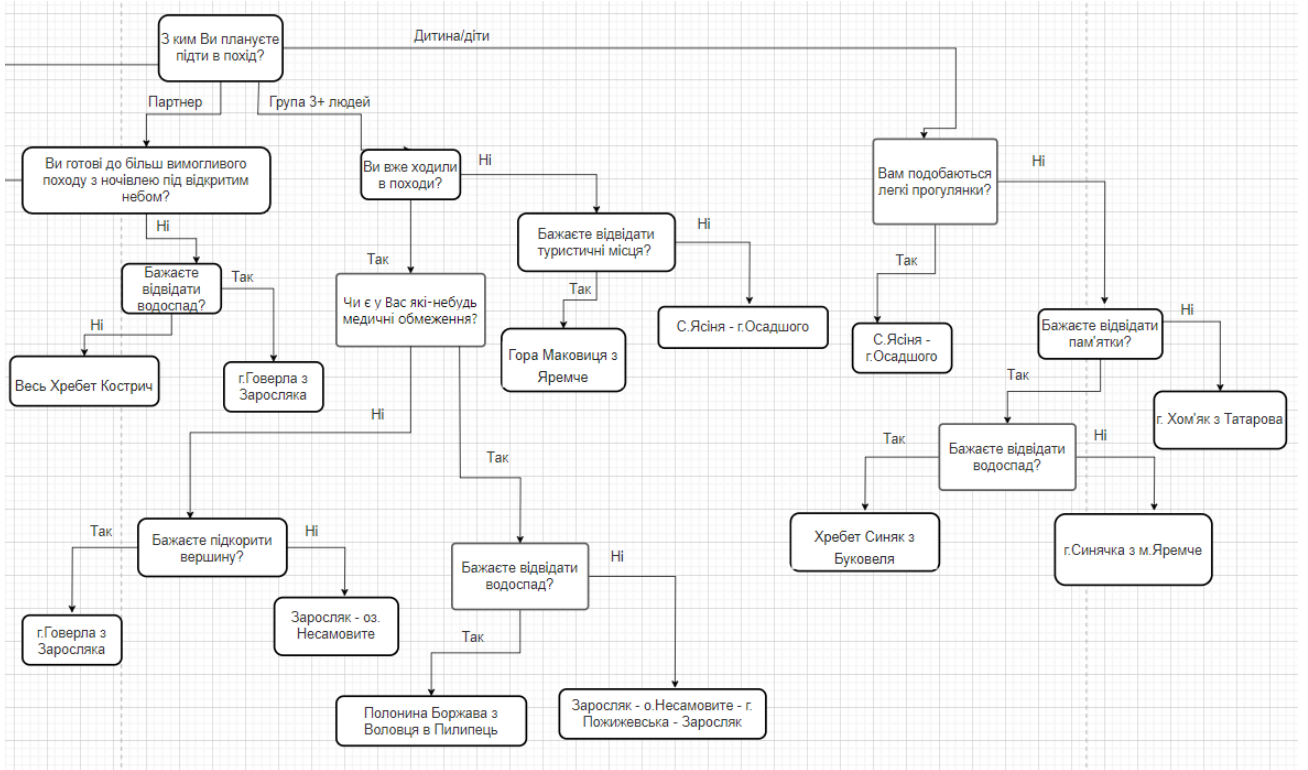


Рисунок 3.3 – Алгоритм підбору походу в Карпати. Частина 1

Джерело: побудовано автором (знімок з екрану)

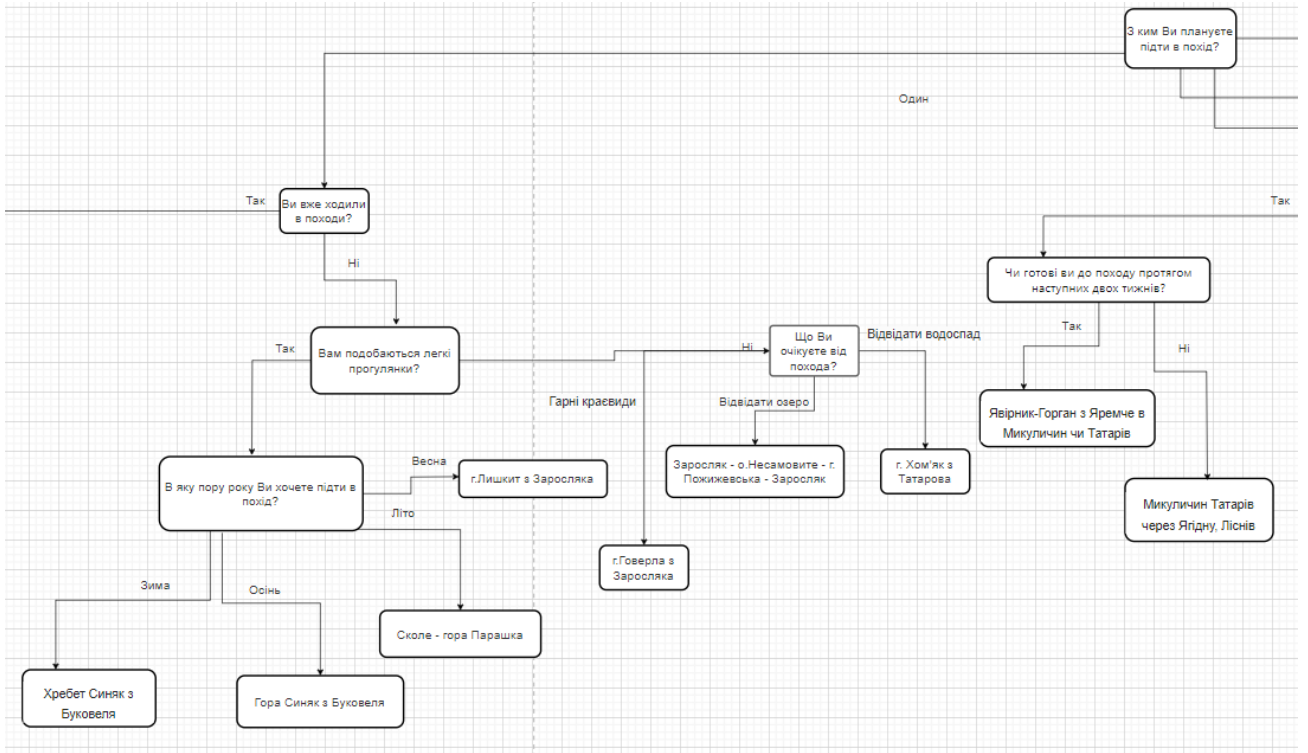


Рисунок 3.4 – Алгоритм підбору походу в Карпати. Частина 2

Джерело: побудовано автором (знімок з екрану)

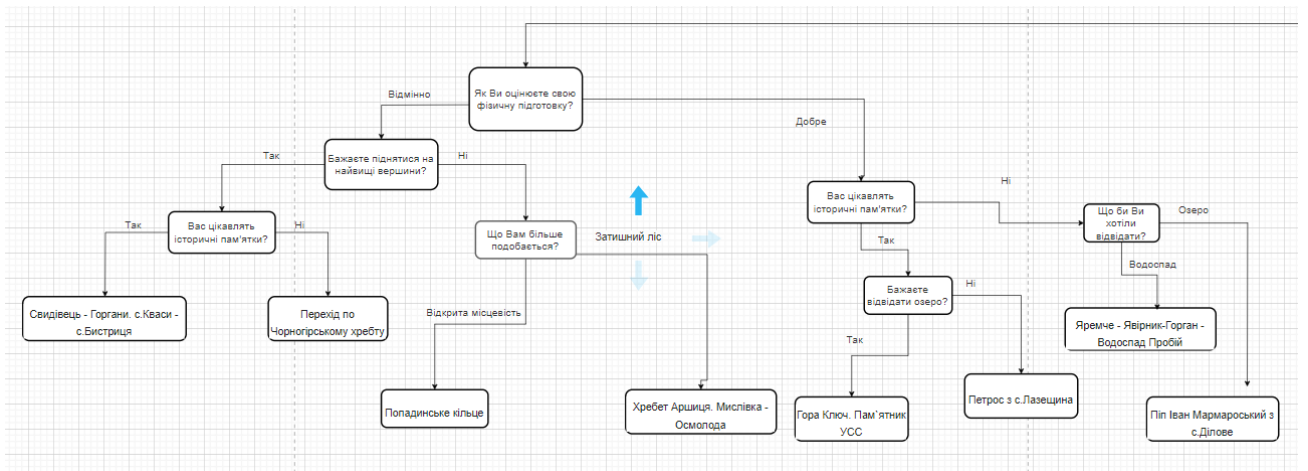


Рисунок 3.5 – Алгоритм підбору походу в Карпати. Частина 3

Джерело: побудовано автором (знімок з екрану)

3.3 Діаграма Use Case

Use case – це методологія, яка використовується в системному аналізі для визначення, уточнення та систематизації вимог до системи. Варіант використання складається з набору можливих послідовностей взаємодій між системами й користувачами в конкретному середовищі і пов'язаних з конкретною метою [32]. Було визначено два актори – користувач та база даних. Нижче відображено варіанти сценаріїв використання представленої системи:

- підібрати тур;
- відправити повідомлення;
- переглянути тур;
- переглянути інтерактивну мапу;
- переглянути мапу прихистку;
- переглянути погоду;
- переглянути запропоноване меню;
- переглянути довідник.

Також було визначено 6 акторів:

- користувач;
- 3 інформаційних ресурси;
- API;
- база даних.

Розроблена діаграма Use Case на основі даних сценаріїв представлена на рисунку 3.6.

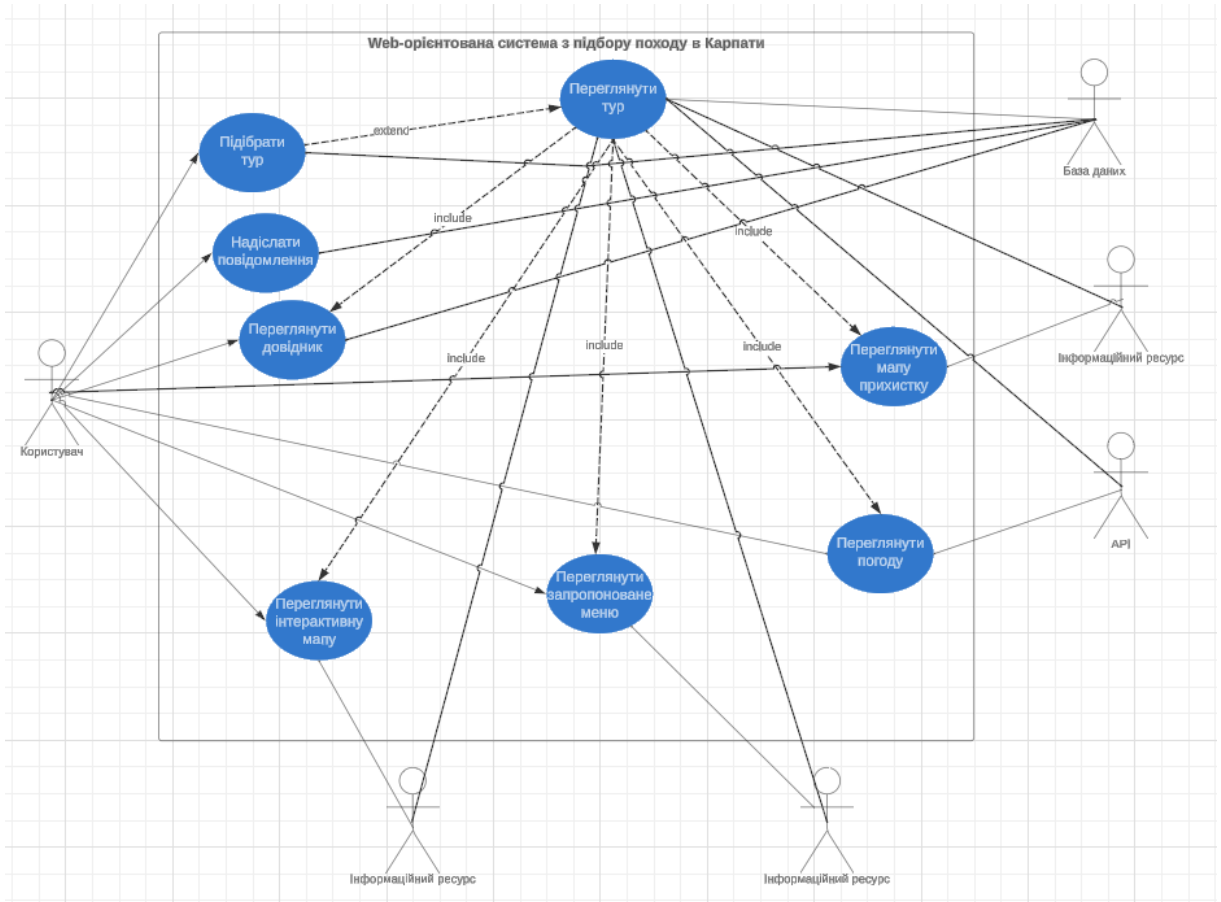


Рисунок 3.6 – Діаграма Use Case

Джерело: побудовано автором (знімок з екрану)

4 РОЗРОБКА WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДБОРУ ПОХОДУ В КАРПАТИ

4.1 Архітектура web-орієнтованої системи

Архітектура програмного забезпечення – це основа, на якій він будується, і яка регулює взаємодію його модулів та компонентів. Це стосується і web-продуктів. Архітектура контролює, як дані передаються через HTTP, і забезпечує процес, щоб сервер клієнта та внутрішній сервер могли їх інтерпретувати. Вона гарантує надійність даних у всіх користувацьких запитах, створення та управління записами, а також доступ та аутентифікацію на основі прав доступу [34].

Архітектура web-орієнтованої системи підбору походів в Карпати містить наступне:

- клієнтський компонент (React Bootstrap, що базується на двох основних елементах React та Bootstrap. React – це бібліотека JavaScript для побудови користувацьких інтерфейсів. Bootstrap – це найпопулярніший CSS-фреймворк для розробки адаптивних і мобільних web-сайтів [35]);

- серверний компонент (node.js – відкрите середовище виконання, яке працює на різних платформах і має велику кількість модулів JavaScript [36]);

- інтерфейс прикладного програмування (API) (не є технологією, а концепція, що є складовою частиною сервера, яка отримує запити та надсилає відповіді за протоколом HTTP [37]);

- база даних (для web-орієнтованої системи підбору походів в Карпати було обрано MySQL. MySQL – це реляційна система управління базами даних із відкритим вихідним кодом, яка дозволяє користувачам ефективно зберігати,

керувати та отримувати доступ до структурованих даних [38]).

Дана архітектура представлена на рисунку 4.1.

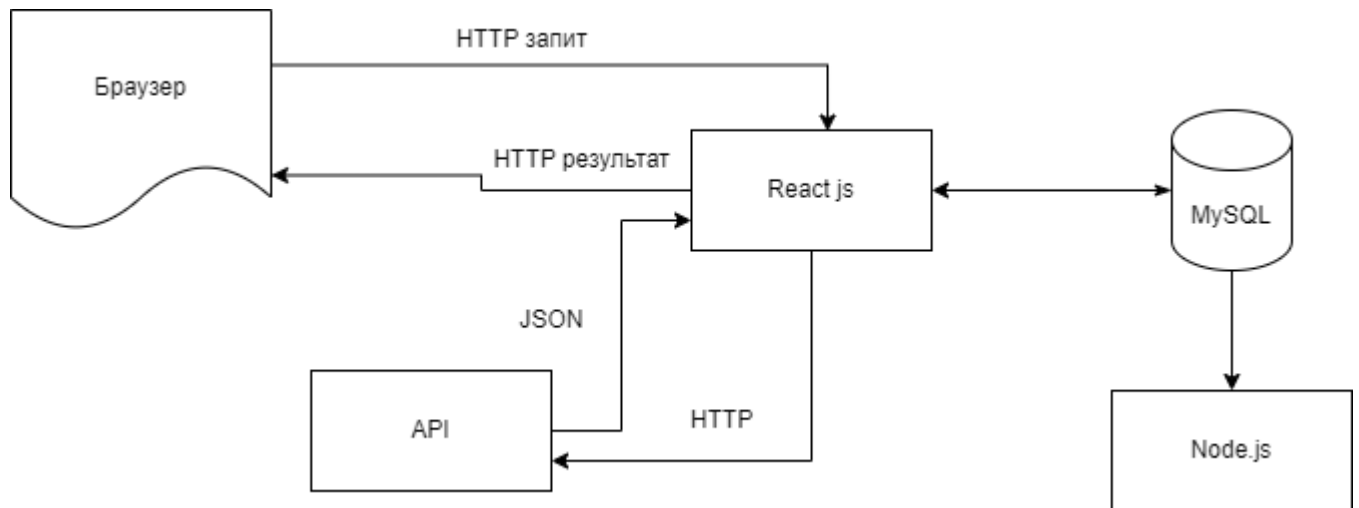


Рисунок 4.1 – Архітектура web-орієнтованої системи

Джерело: побудовано автором (знімок з екрану)

4.2 Розробка бази даних

Далі було реалізовано базу даних, яка містить всю необхідну інформацію для правильної роботи системи. БД є важливим елементом будь-якої web-орієнтованої системи, оскільки вона дозволяє структурувати та зберігати інформацію в одному місці. Це спрощує обробку даних та маніпуляції з ними.

Сформовано кількість таблиць та їх зміст. Кожна з них має свою унікальну структуру, яка включає стовпці, що визначають розмір, тип та характеристики даних, які в ній зберігаються. У результаті засобами phpmyadmin було створено структуру БД, яка зображена на рисунку 4.2.

mydb cards	mydb frankovsk	mydb chernivtsi
id : int(11)	id : int(11)	id : int(11)
imgSrc : varchar(255)	division_name : varchar(255)	division_name : varchar(255)
title : varchar(255)	location : varchar(255)	location : varchar(255)
duration : varchar(255)	contact_person : varchar(255)	contact_person : varchar(255)
distance : varchar(255)	contact_number : varchar(255)	contact_number : varchar(255)
rating : int(11)		
page : varchar(255)		

mydb contacts	mydb lviv	mydb zakarpattya
id : int(11)	id : int(11)	id : int(11)
email : varchar(255)	division_name : varchar(255)	division_name : varchar(255)
message : text	location : varchar(255)	location : varchar(255)
	contact_person : varchar(255)	contact_person : varchar(255)
	contact_number : varchar(255)	contact_number : varchar(255)

Рисунок 4.2 – Структура реалізованої бази даних засобами phpMyAdmin
Джерело: побудовано автором (знімок з екрану)

4.3 Програмна реалізація web-орієнтованої системи

Першим етапом у розробці даної web-орієнтованої системи було створення макету за допомогою Figma (рис. 4.3).

Також за допомогою Canva було розроблено логотип для запропонованого програмного продукту, який представлений на рисунку 4.4.

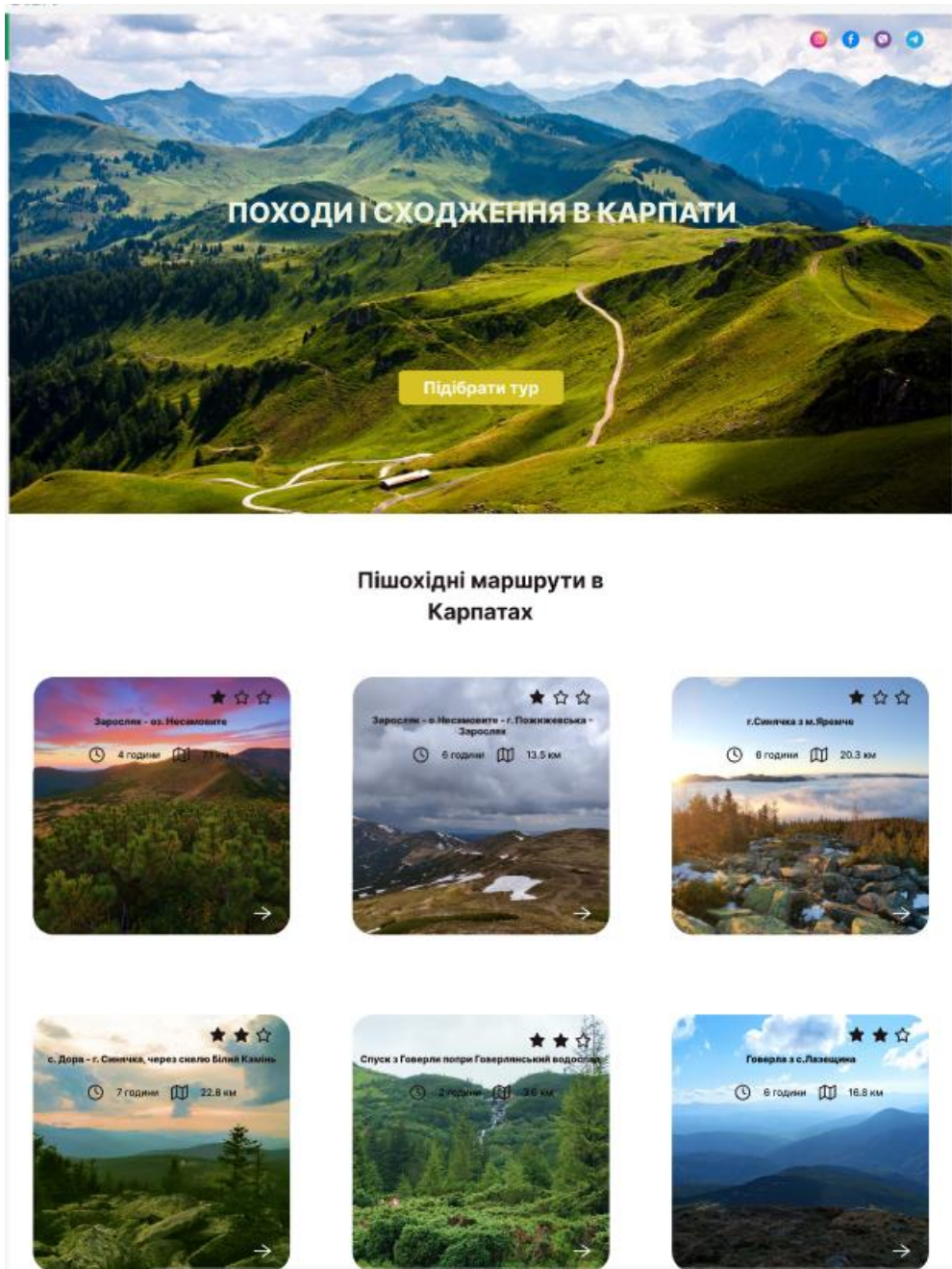


Рисунок 4.3 – Макет web-орієнтованої системи
 Джерело: побудовано автором (знімок з екрану)



Рисунок 4.4 – логотип додатку

Джерело: побудовано автором (знімок з екрану)

Далі було створено клієнтську частину відображення карток за допомогою React Bootstrap та налаштовано відображення зірочок для рівня складності кожного походу. Нижче наведено код налаштування зірочок.

```
renderStars(rating) {
  const stars = [];
  for (let i = 0; i < rating; i++) {
    stars.push(<MdOutlineStar key={i} style={{ fontSize: '24px', color: '#1D1515' }} />);
  }
  for (let i = rating; i < 3; i++) {
    stars.push(<MdOutlineStarBorder key={i} style={{ fontSize: '24px', color: '#1D1515' }} />);
  }
  return (
    <div className="stars" style={{ position: 'absolute', top: '18px', right: '17px' }}>
      {stars}
    </div>
  );
}
```

Наступним кроком було розроблено JavaScript-скрипти для взаємодії з сервером за допомогою fetch-методу [39]. Він надає можливість відправляти дані на сервер і отримувати відповідь без перезавантаження сторінки. Це значно полегшує використання web-орієнтованої системи. Цей метод був використаний для відображення карток походів на головній сторінці, для відображення карток походів після опитування та вивантаження даних контактів рятувальних служб. Нижче наведено код реалізації

```

fetchCardData = async () => {
  try {
    const response = await fetch('/home')
    if (!response.ok) {
      throw new Error('Failed to fetch card data');
    }
  }

```

Для відображення погоди було використано OpenWeatherMap API [41]. Застосовано бібліотеки Axios [42] для виконання GET-запиту до API OpenWeatherMap. Для цього також було використано useEffect для отримання даних з API та зберігання їх у змінній forecast. Нижче наведений код:

```

useEffect(() => {
  const fetchData = async () => {
    try {
      const response = await axios.get(
        `https://api.openweathermap.org/data/2.5/forecast?q=${city}&appid=${apiKey}&units=metric&lang=${lang}`
      );
      setForecast(response.data);
    } catch (error) {
      console.error('Помилка отримання погодного прогнозу:', error);
    }
  };

  fetchData();
}, [apiKey, city, lang]);

```

Лістинг коду основних модулів надано в додатку Б.

4.4 Демонстрація роботи web-орієнтованої системи підбору походу в Карпати

У верхній частині web-орієнтованої системи знаходиться меню: Головна, Про нас, Контакти, Довідник. Також розміщено зображення з кнопкою, яка посилається на сторінку опитування (рис.4.5).

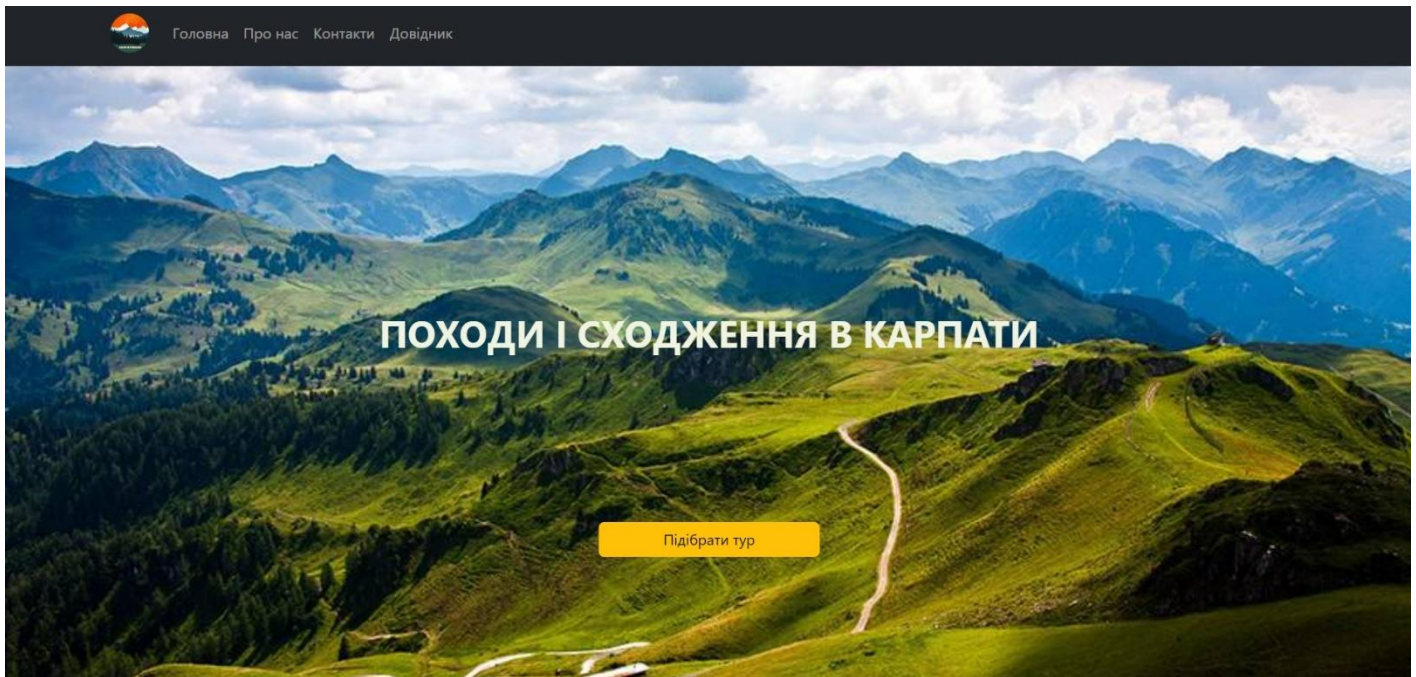


Рисунок 4.5 – Головна сторінка web-системи

Джерело: побудовано автором (знімок з екрану)

Якщо натиснути на відповідну картку походу відобразиться детальна інформація про нього (рис.4.6).

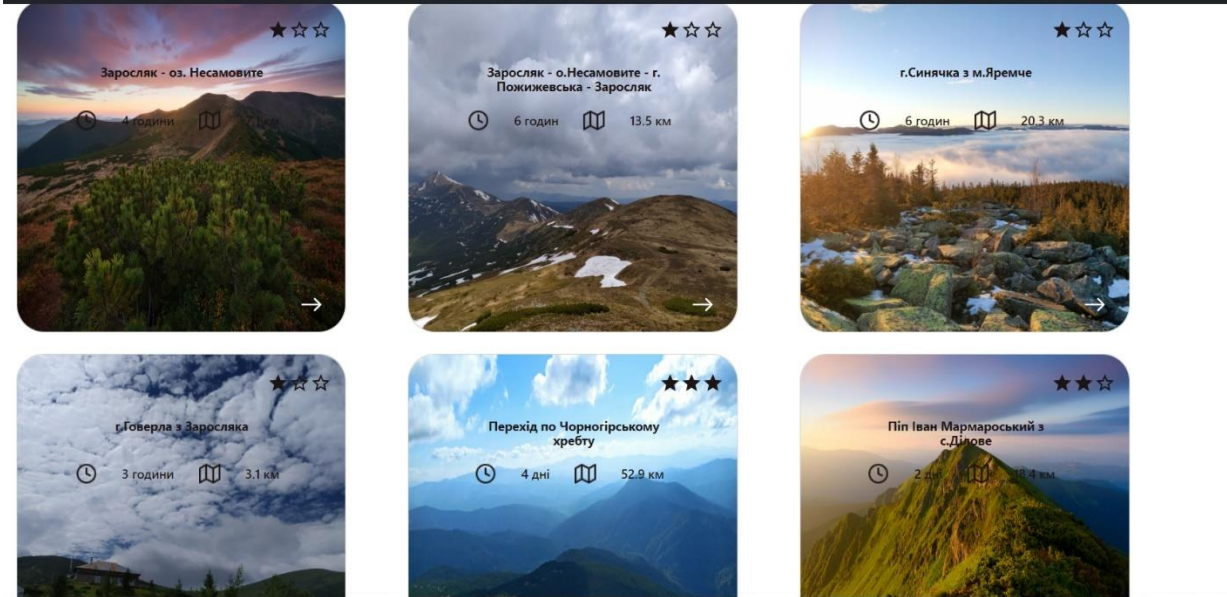


Рисунок 4.6 – Картки походів

Джерело: побудовано автором (знімок з екрану)

Перейшовши на сторінку детальної інформації походу, можна побачити таби з описом маршруту, необхідним спорядженням, підготовкою до туру, правилами безпеки, меню. Таб «Опис маршруту» містить інформацію про маршрут, ціну, гіда та зв'язок з ним, погоду, інтерактивну мапу та мапу прихистку (рис.4.7).

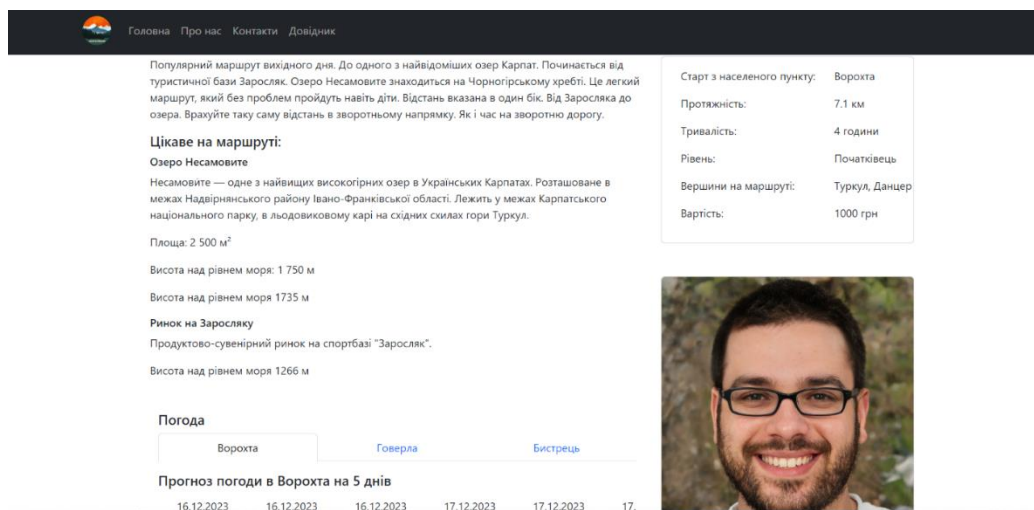
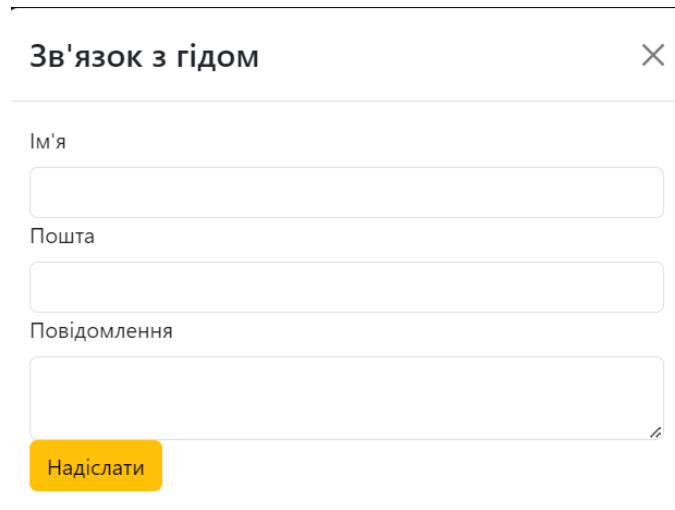


Рисунок 4.7 – Відображення детальної інформації походу

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.8 зображено модульне вікно для зв'язку з гідом.



Зв'язок з гідом

Ім'я

Пошта

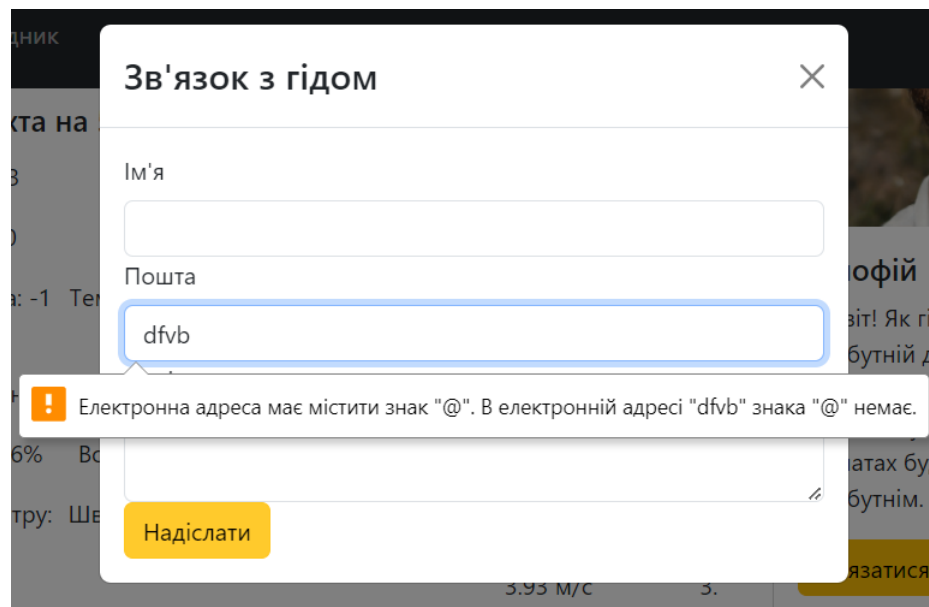
Повідомлення

Надіслати

Рисунок 4.8 – Модальне вікно форми зворотного зв'язку

Джерело: побудовано автором (знімок з екрану)

Якщо поля не заповнені або заповнені неправильно відображається відповідне повідомлення (рис.4.9-4.10).



Зв'язок з гідом

Ім'я

Пошта

dfvb

Електронна адреса має містити знак "@". В електронній адресі "dfvb" знака "@" немає.

Надіслати

Рисунок 4.9 – Повідомлення при некоректному введенні даних

Джерело: побудовано автором (знімок з екрану)

Зв'язок з гідом ×

Ім'я

Пошта

Повідомлення

Повідомлення має бути щонаймеше з 10 символів.

!
✓

Надіслати

Рисунок 4.10 – Повідомлення при відсутності введених даних
Джерело: побудовано автором (знімок з екрану)

На рисунку 4.11 зображено таб «Необхідне спорядження» і містить інформацію про важливу амуніцію для походу.

Головна [Про нас](#) [Контакти](#) [Довідник](#)

Опис маршруту
Необхідне спорядження
Підготовка до туру
Правила безпеки
Меню

Спорядження та одяг для походу

Комплектуючи інвентар та одяг для активного відпочинку, необхідно керуватись основними вимогами: міцність, легкість, надійність, відповідність до умов та місця проведення походу. Головний принцип добору і комплектування туристського спорядження – брати з собою тільки те, без чого не можна обійтись, і нічого зайвого.

Рекомендований список спорядження та одягу для багатоденних мандрівок:

Рюкзак обов'язково

При виборі туристичного рюкзака зверніть увагу на певні особливості. В першу чергу - це об'єм. Якщо похід розрахований на 2-3 дні, то вам вистачить рюкзака об'ємом до 50л, проте для багатоденної мандрівки та зимових походів вам знадобиться рюкзак більших розмірів - від 70л для чоловіків та від 50л для жінок. Наступна особливість - підвісна система (корпус). Рюкзаки з анатомічною спинкою є більш комфортними і безпечними. Це рюкзаки типу «труба» з внутрішнім корпусом у вигляді двох пластин, чи «станок» із зовнішнім корпусом у вигляді рами. Не менш важливою є міцність, легкість та водонепроникність рюкзака. Додаткові елементи: поясний ремінь, бокові стяжки, бокові та фронтальні кишені тощо. Середня вага

Рисунок 4.11 – Таб «необхідне спорядження»

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.12 зображено таб «Підготовка до туру» і містить інформацію про те, як підготуватися до походу.

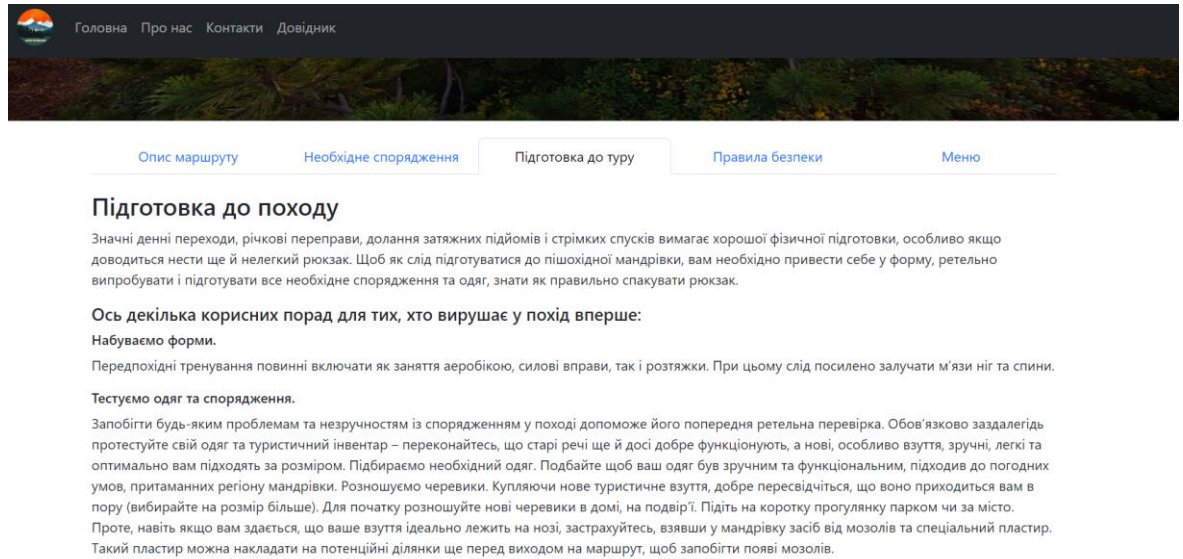


Рисунок 4.12 – Таб «підготовка до походу»

Джерело: побудовано автором (знімок з екрану)

Таб «Правила безпеки» зображено на рисунку 4.13.

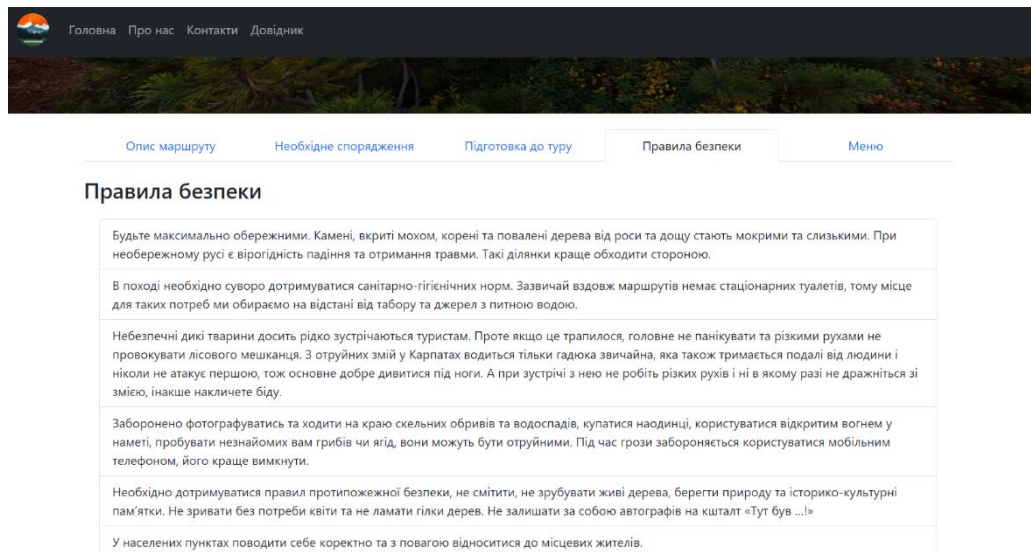


Рисунок 4.13 – Таб «правила безпеки»

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.14 зображено таб «Меню», який відображає рекомендований раціон у поході.

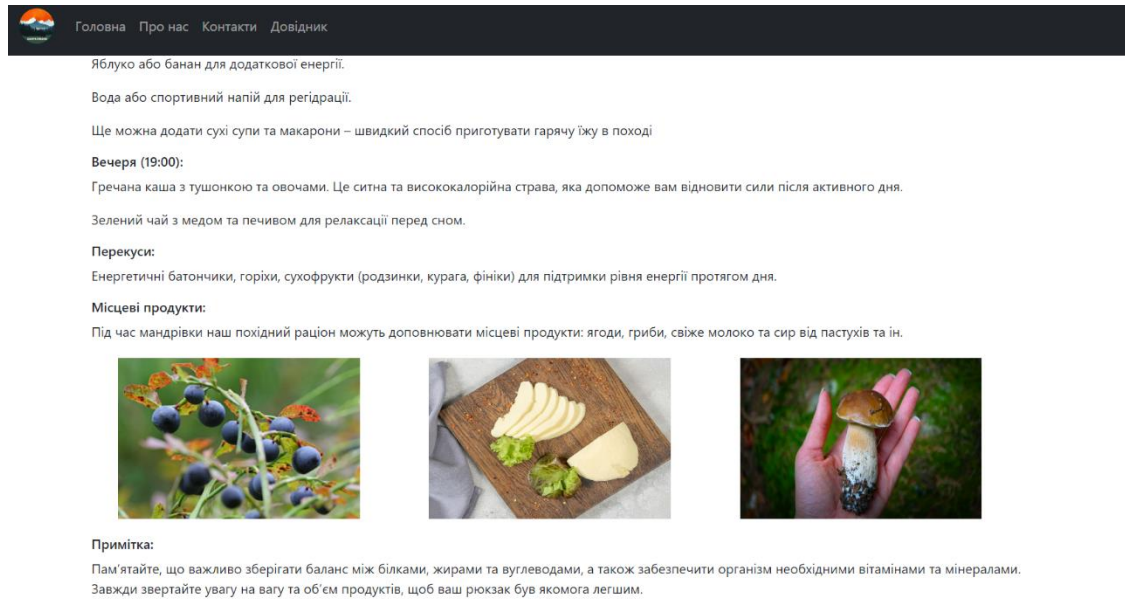


Рисунок 4.14 – Таб «меню»

Джерело: побудовано автором (знімок з екрану)

На головній сторінці є кнопка, яка перенаправляє користувача на сторінку опитування. Процес його проходження та відображення результатів зображено на рисунках 4.15-4.18.

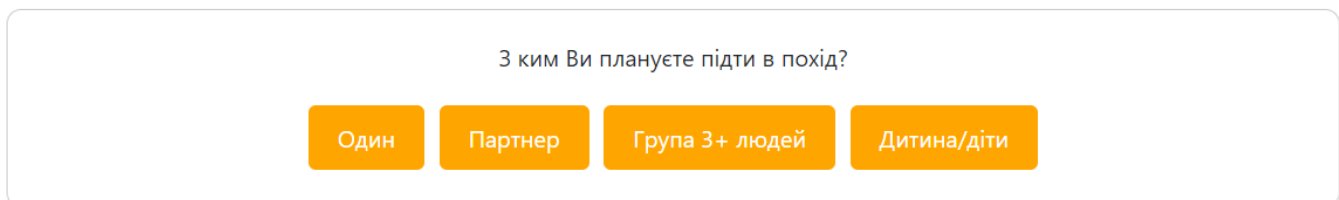


Рисунок 4.15 – Процес проходження опитування

Джерело: побудовано автором (знімок з екрану)

Ви вже ходили в походи?

Так Ні

Рисунок 4.16 – Процес проходження опитування
Джерело: побудовано автором (знімок з екрану)

Що Ви очікуєте від похода?

Гарні краєвиди Відвідати озеро Відвідати водоспад

Рисунок 4.17 – Процес проходження опитування
Джерело: побудовано автором (знімок з екрану)

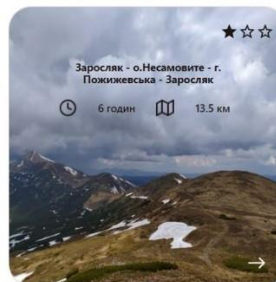
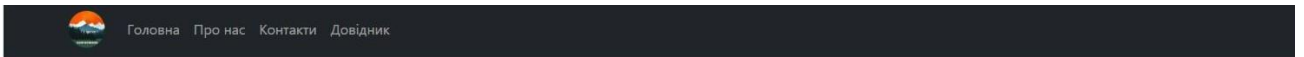


Рисунок 4.18 – Відображення результату
Джерело: зроблено автором (знімок з екрану)

4.5 Тестування web-орієнтованої системи підбору походу в Карпати

Після того, як програмний продукт реалізовано, необхідно перевірити, чи він працює коректно та відповідає вимогам. Для цього було проведено функціональне тестування. Під час нього треба перевірили такі аспекти, як коректність роботи посилань, внутрішнього функціоналу та API, валідація форми зворотного зв'язку, правильність відображення опитування та ефективний підбір походу відповідно до його результатів.

Підсумки проведеного тестування показано в таблиці 4.1.

Таблиця 4.1 – Результати тестування роботи web-орієнтованої системи

№	Опис тесту	Очікуваний результат	Фактичний результат	0/1
1	Вибір необхідного походу на головній сторінці	Виведення на сторінку home/{похід} детальну інформацію про похід	Виведення на сторінку home/{похід} детальну інформацію про похід	1
2	Перегляд погоди за містом	Відображення погоди	Рисунок 4.19	1
3	Перегляд інтерактивної мапи на сторінці походу	Відображення інтерактивної мапи	Відображення інтерактивної мапи	1
4	Перегляд мапи прихистку на сторінці походу	Відображення мапи прихистку	Рисунок 4.20	1
5	Зв'язок з гідом через модальне вікно зв'язку	Відображення модального вікна	Відображення модального вікна	1
6	Перегляд необхідного спорядження на сторінці походу	Відображення необхідного спорядження	Відображення необхідного спорядження	1
7	Перегляд інформації для підготовки до походу	Відображається інформація для підготовки до походу	Рисунок 4.21	1

Продовження таблиці 4.1

№	Опис тесту	Очікуваний результат	Фактичний результат	0/1
9	Перегляд меню	Відображається інформація про меню	Відображається інформація про меню	1
10	Підбір походу за допомогою опитування	Після проходження опитування з'являється похід згідно з обраними варіантами	Відображається похід згідно з обраними варіантами	1
11	Перегляд довідника	Відображення інформації в довіднику	Рисунок 4.22	1

Джерело побудоване автором

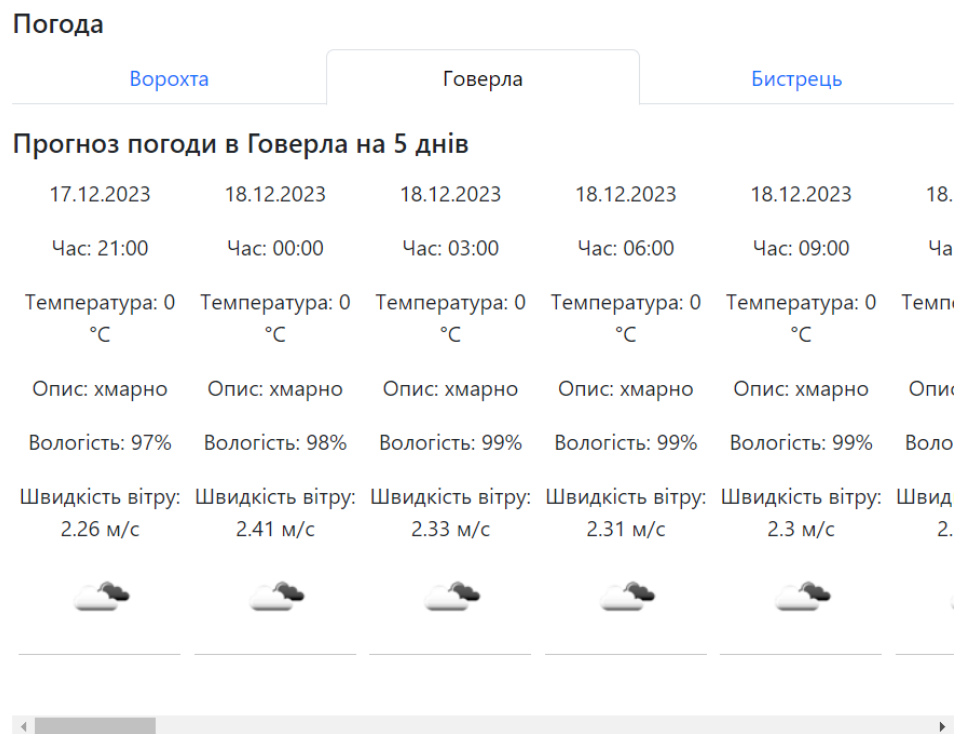


Рисунок 4.19 – Перегляд погоди за містом

Джерело: побудовано автором (знімок з екрану)

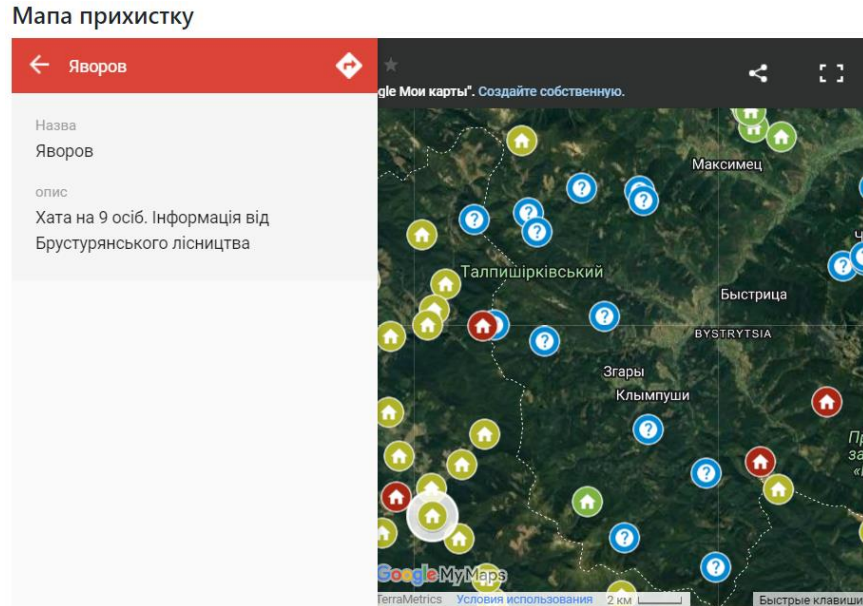


Рисунок 4.20 – Перегляд мапи прихистку

Джерело: побудовано автором (знімок з екрану)

[Опис маршруту](#)

[Необхідне спорядження](#)

[Підготовка до туру](#)

[Правила безпеки](#)

[Меню](#)

Підготовка до походу

Значні денні переходи, річкові переправи, долання зтяжних підйомів і стрімких спусків вимагає хорошої фізичної підготовки, особливо якщо доводиться нести ще й нелегкий рюкзак. Щоб як слід підготуватися до пішохідної мандрівки, вам необхідно привести себе у форму, ретельно випробувати і підготувати все необхідне спорядження та одяг, знати як правильно спакувати рюкзак.

Ось декілька корисних порад для тих, хто вирушає у похід вперше:

Набуваємо форми.

Передпохідні тренування повинні включати як заняття аеробікою, силові вправи, так і розтяжки. При цьому слід посилено залучати м'язи ніг та спини.

Тестуємо одяг та спорядження.

Запобігти будь-яким проблемам та незручностям із спорядженням у поході допоможе його попередня ретельна перевірка. Обов'язково заздалегідь протестуйте свій одяг та туристичний інвентар – переконайтесь, що старі речі ще й досі добре функціонують, а нові, особливо взуття, зручні, легкі та оптимально вам підходять за розміром. Підбираємо необхідний одяг. Подбайте щоб ваш одяг був зручним та функціональним, підходив до погодних умов, притаманних регіону мандрівки. Розношуємо черевики. Купляючи нове туристичне взуття, добре пересвідчіться, що воно приходить вам в пору (вибирайте на розмір більше). Для початку розношуйте нові черевики в домі, на подвір'ї. Підіть на коротку прогулянку парком чи за місто.

Проте, навіть якщо вам здається, що ваше взуття ідеально лежить на нозі, застрахуйтеся, взявши у мандрівку засіб від мозолів та спеціальний пластир. Такий пластир можна накладати на потенційні ділянки ще перед виходом на маршрут, щоб запобігти появі мозолів.

Рисунок 4.21 – Перегляд сторінки «Підготовка до туру»

Джерело: побудовано автором (знімок з екрану)

Контакти рятувальних служб

Контакти прикордонної служби

Транспортні послуги та трансфер

Військовий стан і походи

Контакти прикордонних відділень в Карпатах

Увага. Під час військового стану ця інформація не є актуальною. Більшість прикордонних районів є забороненими для походів.

Комунікувати з прикордонними службами туристам приходиться у випадках походів в прикордонних районах. Наприклад, похід на Піп Іван Мармароський. В цьому випадку потрібно отримати дозвіл від прикордонників на знаходження на цій ділянці кордону.

Дві ділянки кордону в Карпатах

Частина українського кордону на території карпатських гір обслуговується двома прикордонними відділами:

- Мукачівський прикордонний загін – обслуговує частину кордону на території Закарпатської області (Мармароси)
- Чернівецький прикордонний загін – обслуговує частину кордону на території Івано-Франківської та Чернівецької області (Чивчини)

Межа між цими двома прикордонними зонами - гора Стіг (умовний східний кінець "Мармаросів", далі східніше вже Чивчини та Гринявські гори).

Рисунок 4.22 – Перегляд довідника

Джерело: побудовано автором (знімок з екрану)

У результаті тестування було перевірено реалізовані функції, які коректно працюють.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи магістра було проведено аналіз предметної області та огляд останніх публікацій, визначено актуальність створення запропонованого програмного продукту, досліджено продукти-аналоги, виділено їх переваги та недоліки. На основі всього вищезазначеного було сформовано мету та задачі проекту, а також перелік наступних вимог до web-орієнтованої системи підбору походу в Карпати:

- надання інформації про маршрут: складність, час його проходження та відстань;
- представлення переліку цікавих місць і пам'яток на маршруті;
- надання інформації про проноз погоди на обрані дати походу;
- доступ до інтерактивної мапи маршруту;
- можливість придбання походу;
- надання інформації про необхідне спорядження;
- перегляд запропонованого меню під час походів;
- надання інформації про підготовку до походу;
- перегляд довідника з корисними номерами різних служб;
- надання інформації про прихисток у разі негоди;
- використання форми зворотного зв'язку;
- підбір походу за допомогою опитування.

Для реалізації даного проекту було обрано мову програмування JavaScript та бібліотеку React-Bootstrap. Для створення дизайну інтерфейсу взято Figma та Canva. Для реалізації серверної частини використано Node.js та MySQL.

У процесі виконання роботи були розроблені діаграми IDEF0 та їх декомпозиція, створена діаграма сценаріїв використання, побудована схема бази

даних. Крім того, було проведено оцінювання ризиків і здійснено планування роботи.

Розроблена серверна частина, яка відповідає за обробку запитів до бази даних. Протестовано роботу даного програмного продукту. Суттєві дефектів виявлено не було.

Результатом кваліфікаційної роботи магістра є готова web-орієнтована система підбору походу в Карпати. Її використання дозволить популяризувати даний напрям подорожей. У свою чергу, це допоможе зростанню місцевої економіки та підвищенню якості життя в регіоні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bayrak G. R., Teodorovych L. V. Geological and geomorphological objects of the Ukrainian Carpathians' Beskid Mountains and their tourist attractiveness. *Journal of Geology, Geography and Geoecology*. 2020. Vol. 29, no. 1. P. 16–29. URL: <https://doi.org/10.15421/112002> (date of access: 11.11.2023).
2. Brusak V., Gnatiak I., Shtuhlynets V. STATE AND MONITORING OF CARPATIAN NATIONAL PARK TOURIST ROUTES' MICRORELIEF. PROBLEMS OF GEOMORPHOLOGY AND PALEOGEOGRAPHY OF THE UKRANIAN CARPATHIANS AND ADJACENT AREAS. 2023. No. 15. P. 30–47. URL: <https://doi.org/10.30970/gpc.2023.1.3946> (date of access: 11.11.2023).
3. Basic Preparation of a Tourist for Sports and Fitness Hike / Mozolev O. et al. // *International Journal of Applied Exercise Physiology*. 2020., No. 12 Vol. 9. P. 99–106.
4. Белегай А. А., Бардин Г. В., Бардин О. І. Теоретичні аспекти формування спортивних пішохідних турів // *Матеріали ІХ Всеукраїнської науково-практичної конференції курсантів, студентів, аспірантів та ад'юнктів / Львівський державний університет безпеки життєдіяльності*. Львів, 2019. С. 145–147.
5. Клем-Мусатова, К. О. Теоретичні аспекти формування спортивних пішохідних турів [Текст] / К. О. Клем-Мусатова, Я. В. Євтушенко // *Стан і перспективи сучасного туризму : матеріали VI Всеукраїнської науково-практичної конференції (м. Суми, 26–27 листопада 2020)*. – Суми : ФОП Цьома С. П., 2020. – С. 137–142.
6. Кривенцова І. В., Цимбалюк Ж. О., Шип Н. Є. *Методика організації та проведення туристичного походу: посібник*. Харків, 2023. 96 с.

7. Шутяк І. ТУРИСТИЧНА ДІЯЛЬНІСТЬ ТА МЕТОДИКА ОРГАНІЗАЦІ ТУРИСТИЧНОГО ПОХОДУ. *Modern engineering and innovative technologies*. 2020. № 25-03. С. 51–58. URL: <https://doi.org/10.30890/2567-5273.2023-25-03-027> (дата звернення: 11.11.2023).
8. Romaniv A. , Myronets N. Adventure tourism in ukraine: current trends and directions of the european experience implementation // *European Science*. 2021. № 1. P. 163-168.
9. The Preference Analysis for Hikers' Choice of Hiking Trail / M. Molokáč et al. *Sustainability*. 2022. Vol. 14, no. 11. P. 6795. URL: <https://doi.org/10.3390/su14116795> (date of access: 11.11.2023).
10. Web-Based Recommendation System for Smart Tourism: Multiagent Technology / R. Hassannia et al. *Sustainability*. 2019. Vol. 11, no. 2. P. 323. URL: <https://doi.org/10.3390/su11020323> (date of access: 11.11.2023).
11. Мозолев О. ОРГАНІЗАЦІЯ МОЛОДІЖНОГО СПОРТИВНО-ОЗДОРОВЧОГО ТУРИЗМУ : Навч. посіб. Хмельницький, 2020. 159 с.
12. ПРАКТИКА ЗІ СПОРТИВНОГО ТУРИЗМУ ТА ОРІЄНТУВАННЯ. : Матеріали для самостійної роботи для здобувачів вищої освіти ступеня бакалавра за спеціальністю 014.11 Середня освіта (Фізична культура) та 017 Фізична культура і спорт. Київ, 2023. 96 с.
13. KULUAR UA ПОХОДИ В ГОРИ. СХОДЖЕННЯ. ТРЕКІНГ. URL: <https://kuliarpohod.com/ua/> (дата звернення: 24.11.2023).
14. Mount Day. URL: <https://mount-day.com/> (date of access: 24.11.2023).
15. Karpaty.ua. URL: <https://guide.karpaty.ua/uk/categories/marshruty> (date of access: 24.11.2023).
16. Брикова Т. М., Терешкін О. Г. Основи наукових досліджень: навчальний посібник. Харків: ХДУХТ, 2020. 103 с.

17. What is Decision Tree? (With Case Study). Visual Paradigm - Online Productivity Suite. URL: <https://online.visual-paradigm.com/ru/knowledge/decision-tree/what-is-decision-tree/> (date of access: 13.12.2023).
18. Design and Development a Website using HTML, CSS, PHP and MySQL Design and Development a Website using HTML, CSS, PHP and MySQL / A. A. Zubaer et al. Journal of Information Technology and Sciences. 2020. Vol. 6, no. 3.
19. Rebah H. B., Boukthir H., Chedebois A. Website Design and Development with HTML5 and CSS3. Wiley & Sons, Incorporated, John, 2021.
20. Sotnik S. , Shakurova T. , Lyashenko V. Development Features Web-Applications // International Journal of Academic and Applied Research (IJAAR). 2023., No. 1 Vol. 7. P. 79-85.
21. React. URL: <https://react.dev/> (date of access: 24.11.2023).
22. Choi D. Full-Stack React, TypeScript, and Node: Build Cloud-Ready Web Applications Using React 17 with Hooks and GraphQL. Packt Publishing, Limited, 2020.
23. Boersma S., Lungu M. React-bratus: Visualising React Component Hierarchies. 2021 Working Conference on Software Visualization (VISSOFT), Luxembourg, 27–28 September 2021. 2021. URL: <https://doi.org/10.1109/vissoft52517.2021.00025> (date of access: 25.11.2023).
24. React Bootstrap. URL: <https://react-bootstrap.netlify.app/> (date of access: 24.11.2023).
25. Subramanian V. React-Bootstrap. Pro MERN Stack. Berkeley, CA, 2019. P. 315–376. URL: https://doi.org/10.1007/978-1-4842-4391-6_11 (date of access: 25.11.2023).
26. Rebah H. B., Boukthir H., Chedebois A. Website Design and Development with HTML5 and CSS3. Wiley & Sons, Incorporated, John, 2021.
27. Figma. URL: <https://www.figma.com/> (date of access: 24.11.2023).
28. Canva. URL: <https://www.canva.com/> (date of access: 24.11.2023).

29. Node.js. URL: <https://nodejs.org/en/about> (date of access: 24.11.2023).
30. Šušter I., Ranisavljević T. Optimization of MySQL database. Journal of Process Management and New Technologies. 2023. Vol. 11, no. 1-2. P. 141–151. URL: <https://doi.org/10.5937/jouproman2301141q> (date of access: 25.11.2023).
31. Моделювання бізнес-процесів : [підручник] / [О.І. Долганова, Є.В. Виноградова, А.М. Лобанова]. –М. : Юрайт, 2016. – 289 с.
32. TechTarget. URL: <https://www.techtarget.com/searchsoftwarequality/definition/use-case> (date of access: 24.11.2023).
33. Борисевич А. Л. Прийняття управлінських рішень. Прийняття рішень в умовах ризику за допомогою «дерева рішень». ЕЛЕКТРОННИЙ МЕТОДИЧНИЙ КОМПЛЕКС навчальної практики з дисципліни «Інформаційна діяльність підприємства» для студентів III курсу спеціальності 076 «Підприємництво, торгівля та біржова діяльність». URL: https://ck.vk.mnau.edu.ua/ck/portfolio_Borisevich/prakt_idp/RN.HTML (дата звернення: 13.12.2023).
34. Журнал Вікторія. URL: <https://www.victoria.lviv.ua/library/students/wpr/lecture/6.docx> (дата звернення: 15.12.2023).
35. Bootstrap. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/> (date of access: 15.12.2023).
36. Lim G. Beginning Node.js, Express & MongoDB Development. Greg Lim, 2020. 152 p.
37. МJBnaf. Що таке API? Просте пояснення для початківців. dev.ua. URL: <https://dev.ua/news/valid-article-1662018442-1662018442> (дата звернення: 15.12.2023).
38. Робота з базою даних MySQL у cPanel • Hostpro Wiki. Hostpro Wiki. URL: <https://hostpro.ua/wiki/ua/panels/cpanel/how-to-create-and-maintain-mysql-databases-in-cpanel> (дата звернення: 15.12.2023).

39. Fetch [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.javascript.info/fetch>.

40. Що таке JSON | Навіщо потрібен цей формат [Електронний ресурс] – Режим доступу до ресурсу: <https://apix-drive.com/ua/blog/useful/scho-take-json>.

41. Current weather and forecast - OpenWeatherMap. URL: <https://openweathermap.org/> (дата звернення: 15.12.2023).

42. Починаючи | Axios Docs. Axios. URL: <https://axios-http.com/uk/docs/intro> (дата звернення: 15.12.2023).

ДОДАТОК А

Планування робіт

У сучасному світі туризм стає все більш важливим елементом нашого життя. Після пандемії багато країн активізували туристичну діяльність для відновлення своєї економіки. І Україна не є винятком. Завдяки своїй різноманітній природі та культурному спадку, наша Батьківщина має великий потенціал для становлення центром туризму, який приваблює відвідувачів із усього світу. Однак, через сучасні військові конфлікти, масштабне їх залучення ще не можливе. Проте внутрішні процеси в Україні позитивно вплинули на розвиток внутрішнього туризму. Наприклад, обмеження виїзду за кордон під час воєнного стану спонукало українців досліджувати свою власну країну та відкривати для себе нові цікаві місця. Карпати, один із найкрасивіших регіонів нашої Батьківщини, стали особливо популярними серед туристів. Дана робота має на меті розробити web-орієнтовану систему, яка допоможе користувачам підібрати та придбати похід в Карпатах онлайн. Це спрямовано на створення інноваційного інструменту, який надасть споживачам всебічну інформацію про доступні маршрути, а також дозволить здійснити бронювання та/або купити тур через мережу Інтернет. Застосування запропонованої системи допоможе зменшити зусилля користувачів при плануванні та організації походів. У свою чергу це покращить їхній туристичний досвід в Карпатах.

Деталізація мети методом SMART. Цей метод є ефективним способом встановлення мети, який збільшує шанси її досягнення та допомагає зосередитися на конкретних завданнях. Вона має ширше формулювання. А саме: «Розробка web-орієнтованої системи підбору походу в Карпати, застосування якої дозволить користувачам самостійно підібрати та організувати власну

подорож та сприятиме як збільшенню туристичного потоку в даному регіоні, так і покращенню якості обслуговування мандрівників до 15 грудня 2023 року». Результати деталізації мети проєкту методом SMART представлено у таблиці А.1

Таблиця А.1 – Деталізація мети проєкту методом SMART

Specific (Конкретна)	Розробити web-орієнтовану систему підбору походу в Карпати, яка дозволить користувачам самостійно підібрати та організувати власну подорож.
Measurable (Вимірювана)	Результатом роботи проєкту є розроблена web-орієнтована система підбору походу в Карпати.
Achievable (Досяжна)	Для виконання проєкту наявні необхідні знання Canva, Figma, React-Bootstrap, мови програмування JavaScript, PHP, фреймворку React, баз даних MySQL та навичок написання документації. Враховуючи доступні ресурсні можливості та обмеження мета є такою, яку можливо досягти.
Relevant (Реалістична)	Розроблена web-орієнтована система сприятиме збільшенню туристичного потоку в даному регіоні, покращенню якості обслуговування мандрівників та розвитку місцевої travel індустрії, що вплине на популяризацію Карпатських гір як серед населення України, так й за кордоном, і допоможе розвитку місцевої економіки та підвищенню якості життя в регіоні.
Time-framed (Обмежена у часі)	Ціль має часове обмеження. Проєкт повинен бути виконаний згідно з календарним планом – до 15 грудня 2023 року.

Джерело: побудовано автором

Планування змісту структури робіт. Work Breakdown Structure або WBS є ключовим інструментом для планування та структуризації проєктів. Він допомагає розділити великі завдання на менші, більш керовані компоненти. Це полегшує управління проєктом та забезпечує краще розуміння обсягу робіт. На рисунку А.1 представлена WBS-структура для розробки web-орієнтованої системи підбору походів в Карпати.

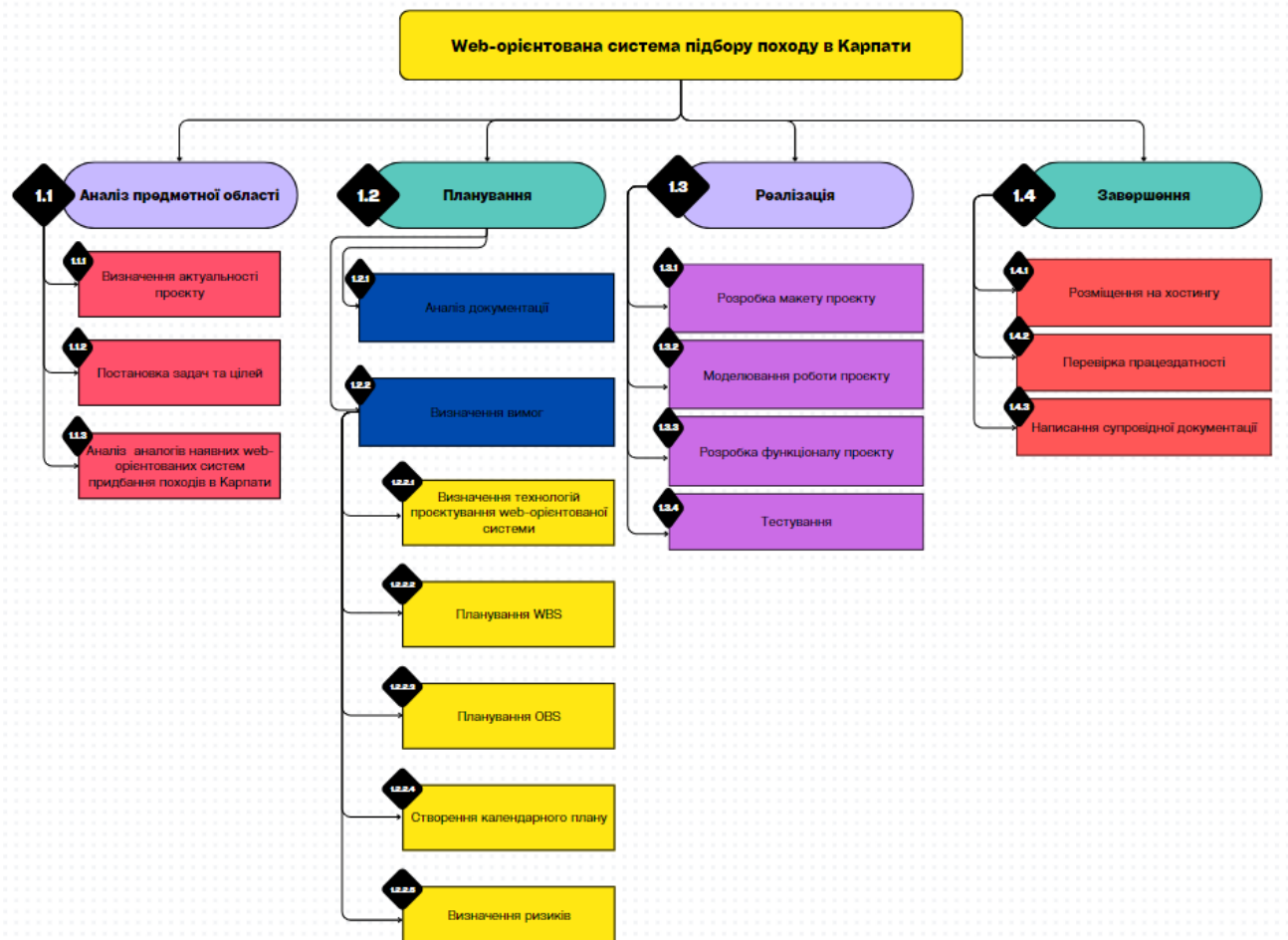


Рисунок А.1 – WBS-структура проєкту

Джерело: побудовано автором

Планування структури організації, для впровадження готового проєкту (OBS). Organizational Breakdown Structure або OBS є важливим інструментом для організації та визначення відповідальності учасників проєкту.

Він є досить корисним. OBS допомагає зрозуміти, як різні організації взаємодіють і співпрацюють для завершення задач. Список виконавців, які задіяні в проєкті, представлений в таблиці А.2. Організаційна структура проєкту зображена на рисунку А.2.

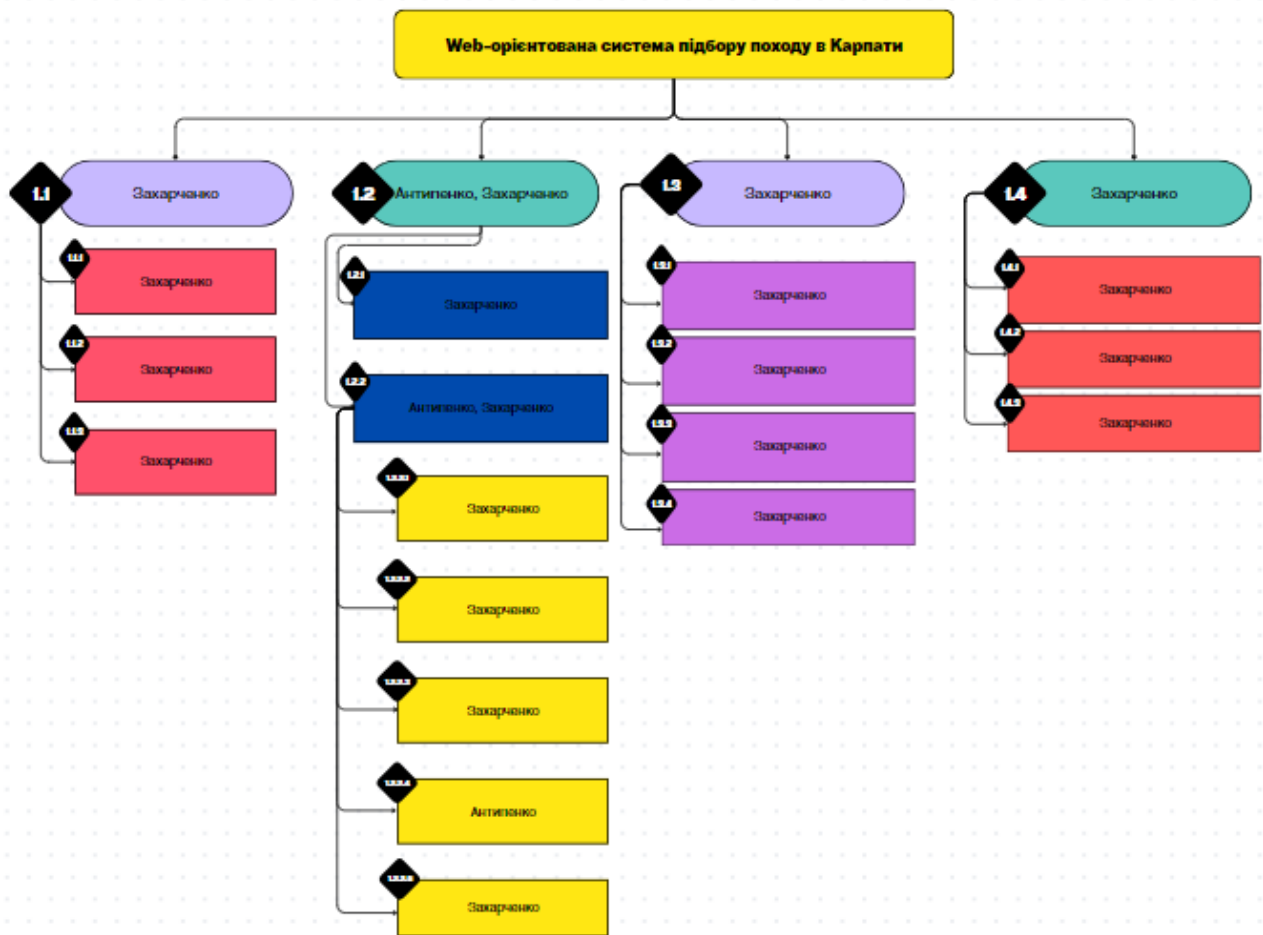


Рисунок А.2 – OBS-структура проєкту

Джерело: побудовано автором

Таблиця А.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Захарченко А.О.	Виконує frond-end та back-end розробку
Проектувальник	Захарченко А.О.	Виконує проектування бази даних та розробляє структуру web-орієнтованої системи
Тестувальник	Захарченко А.О.	Проводить тестування web-орієнтованої системи
Керівник проекту	Антипенко В.П.	Формує завдання на розробку проекту

Джерело: побудовано автором

Діаграма Ганта. Цей інструмент дозволяє ясно відобразити часові рамки виконання завдань, а також взаємозв'язки між ними. Діаграма Ганта може бути корисною для управління проектами та планування робіт, а також для візуалізації графіка реалізації задач.

Календарний графік проекту представлено на рисунку А.3.

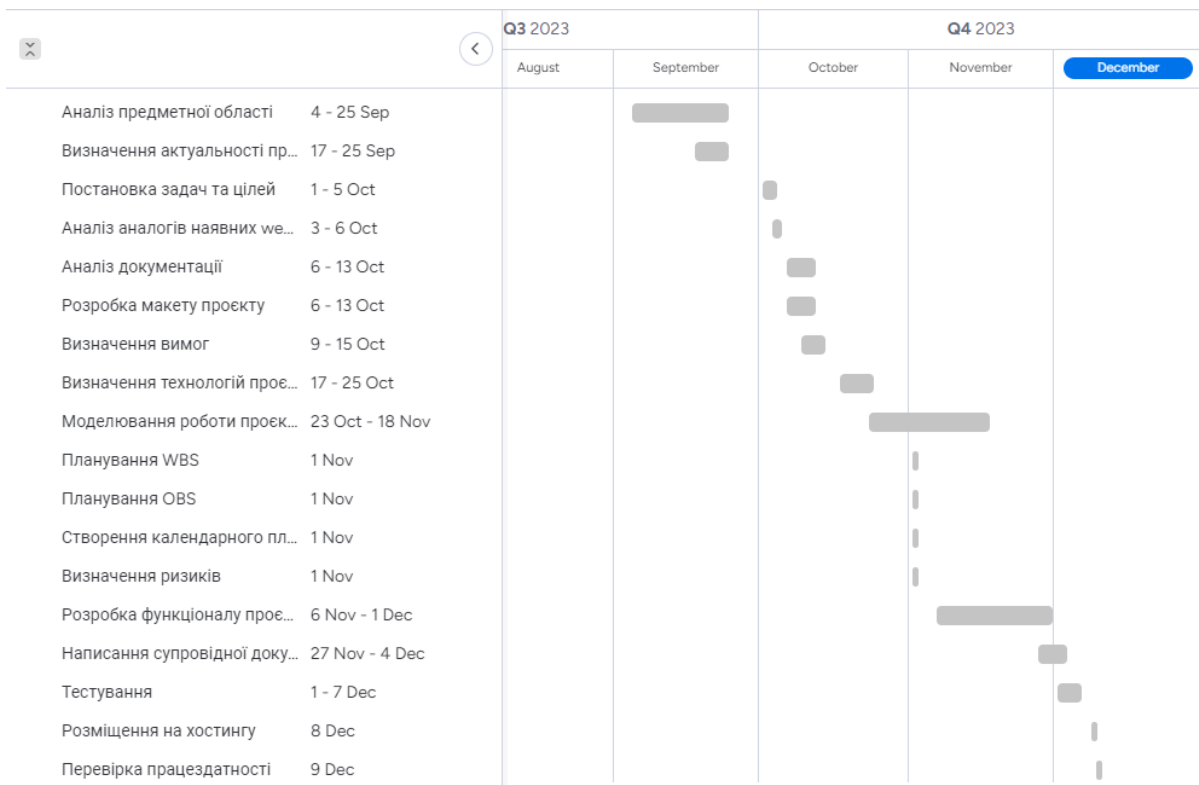


Рисунок А.3 – Діаграма Ганта проєкту

Джерело: побудовано автором (знімок з екрану)

Аналіз предметної області	📅	4 - 25 Sep
Визначення актуальності проє...	📅	17 - 25 Sep
Постановка задач та цілей	📅	1 - 5 Oct
Аналіз аналогів наявних web-o...	📅	3 - 6 Oct
Аналіз документації	📅	6 - 13 Oct
Визначення вимог	📅	9 - 15 Oct
Визначення технологій проєкт...	📅	17 - 25 Oct
Планування WBS	📅	1 Nov
Планування OBS	📅	1 Nov
Створення календарного плану	📅	1 Nov
Визначення ризиків	📅	1 Nov
Розробка макету проєкту	📅	6 - 13 Oct
Моделювання роботи проєкту	📅	23 Oct - 18 Nov
Розробка функціоналу проєкту	📅	6 Nov - 1 Dec
Тестування	📅	1 - 7 Dec
Розміщення на хостингу	📅	8 Dec
Перевірка працездатності	📅	9 Dec
Написання супровідної докуме...	📅	27 Nov - 4 Dec

Рисунок А.4 – Список робіт для побудови діаграми Ганта

Джерело: побудовано автором (знімок з екрану)

Управління ризиками проєкту. Ризики – це потенційні події чи умови, які можуть вплинути на досягнення цілей проєкту, якщо вони стаються. Ефективне управління ними дозволяє зменшити ймовірність негативних наслідків А також максимізувати можливості для досягнення успіху проєкту. У таблиці А.3 надано перелік ризиків. Результати їх оцінки представлено в таблиці А.4. Таблиця А.5 демонструє шкалу для класифікації ризиків за величиною впливу на проєкт та ймовірністю їх виникнення.

Таблиця А.3 – Ризики проєкту

№ ризику	Назва (опис) ризику
1	Недотримання календарного плану
2	Нечітка постановка задач
3	Вибір неефективної технології розробки
4	Збої в роботі програмного продукту
5	Неправильний розподіл часу
6	Хвороба розробника
7	Некоректне тестування
8	Відключення світла
9	Відсутність доступу до мережі Інтернет
10	Позапланова операція на очах

Джерело: побудовано автором

Таблиця А.4 – Результати визначення ймовірності, впливу та рангу ризиків проекту

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Недотримання календарного плану	0,3	0,4	0,12
2	Нечітка постановка задач	0,1	0,05	0,005
3	Вибір не ефективної технології розробки	0,5	0,6	0,3
4	Збої в роботі програмного продукту	0,3	0,7	0,21
5	Неправильний розподіл часу	0,5	0,4	0,2
6	Хвороба розробника	0,2	0,5	0,1
7	Некоректне тестування	0,4	0,2	0,08
8	Відключення світла	0,2	0,7	0,14
9	Відсутність доступу до мережі Інтернет	0,3	0,4	0,12
10	Позапланова операція на очах	0,1	0,8	0,08

Джерело: побудовано автором

Таблиця А.5 – Шкала оцінювання ризиків за ймовірністю виникнення на величину впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Джерело: побудовано автором

Для зниження негативного впливу ризиків на проєкт треба виконати планування реагування на них. До нього входить оцінка наслідків впливу на прогрес реалізації задач. І відповідно розробка певних заходів. Аналіз виконується за показниками, які описані в таблиці А.4. У результаті планування заходів реагування на ризики даного проєкту було отримано матрицю ймовірності виникнення та впливу ризиків (рис. А.5). Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Ймовірність виникнення ризику	Вплив ризику				
	0,05	0,1	0,2	0,4	0,8
0,9	0,045	0,09	0,18	0,36	0,72
0,7	0,035	0,07 R8	0,14 R3, R4, R5	0,28	0,56
0,5	0,025	0,05 R1, R2	0,1	0,2	0,4
0,3	0,015	0,03	0,06	0,12 R1, R6, R9	0,24
0,1	0,005	0,01	0,02	0,04	0,08 R7, R10

Рисунок А.5. – Матриця ймовірності та впливу

Джерело: побудовано автором

Класифікація ризиків проєкту за рівнем, відповідно до отриманого значення індексу, представлена у таблиці А.6. У таблиці А.7 описано ризики та стратегії реагування на кожен із них.

Таблиця А.6 – Шкала оцінювання ризику за рівнем

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$0,005 \geq R \geq 0,05$	1, 2
2	Виправдані	$0,05 < R \geq 0,14$	3, 4, 5, 6, 7, 8, 9, 10
3	Недопустимі	$0,14 < R \geq 0,72$	

Джерело: побудовано автором

Таблиця А.7 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_1	Відкритий	Недотримання календарного плану	Низька	Середній	0,05	Зменшення	<p>1. Сформувати детальний план проекту, з урахуванням всіх етапів та завдань.</p> <p>2. Впровадити систему контролю виконання, яка дозволить своєчасно виявляти відхилення від плану та приймати необхідні заходи.</p>	Розробити новий календарний план, який враховує можливі затримки.

Продовження табл. А.7

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_2	Відкритий	Нечітка постановка задач	Низька	Низький	0,005	Зменшення	Чітко окреслити задачі.	Сформувати список невідповідних або нечітко сформульованих вимог.
RS_3	Відкритий	Вибір не ефективної технології розробки	Середня	Високий	0,3	Зменшення	Проаналізувати різні технології розробки, які можуть бути використані для реалізації проєкту.	Проаналізувати терміни переходу на іншу технологію.
RS_4	Відкритий	Збої в роботі програмного продукту	Низька	Високий	0,21	Зменшення	Розробити план тестування, який буде включати в себе тестування всіх можливих сценаріїв використання продукту.	Визначити всі виявлені проблеми.

Продовження табл. А.7

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_5	Відкритий	Відключення світла	Низька	Високий	0,14	Зменшення	На випадок відключення світла, можна скористатися коворкінгом або кав'ярнею з генератором.	Купити генератор .
RS_6	Відкритий	Відсутність доступу до мережі Інтернет	Низька	Середній	0,12	Зменшення	На випадок відсутності інтернету, можна роздати мобільний інтернет.	Скористатися коворкінгом або кав'ярнею.
RS_7	Закритий	Позапланова операція на очах	Низька	Високий	0,08	Прийняття	Відтермінувати операцію за можливості	Скористатися спеціальними можливостями для погано зрячих, як от надиктовування, відтворювання аудіо.

Продовження табл. А.7

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування	План А	План Б
RS_8	Відкритий	Відключення світла	Низька	Високий	0,14	Зменшення	На випадок відключення світла, можна скористатися коворкінгом або кав'ярнею з генератором.	Купити генератор .
RS_9	Відкритий	Відсутність доступу до мережі Інтернет	Низька	Середній	0,12	Зменшення	На випадок відсутності інтернету, можна роздати мобільний інтернет.	Скористатися коворкінгом або кав'ярнею.
RS_10	Закритий	Позапланова операція на очах	Низька	Високий	0,08	Прийняття	Відтермінувати операцію за можливості	Скористатися спеціальними можливостями для погано зрячих, як от надиктовування, відтворення аудіо.

Джерело: побудовано автором

ДОДАТОК Б

Лістинг коду модулів web-орієнтованої системи

Код сторінки Home.js

```
export default class Home extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      cardData: [],  
    };  
  }  
  
  componentDidMount() {  
    this.fetchCardData();  
  }  
  
  fetchCardData = async () => {  
    try {  
      const response = await fetch('/home')  
      if (!response.ok) {
```

```
        throw new Error('Failed to fetch card data');
    }

    const cardData = await response.json();

    this.setState({ cardData });

} catch (error) {
    console.error('Error fetching card data:', error.message);
}
};

renderStars(rating) {
    const stars = [];

    for (let i = 0; i < rating; i++) {
        stars.push(<MdOutlineStar key={i} style={{ fontSize: '24px', color: '#1D1515' }} />);
    }

    for (let i = rating; i < 3; i++) {
        stars.push(<MdOutlineStarBorder key={i} style={{ fontSize: '24px', color: '#1D1515' }} />);
    }

    return (
        <div className="stars" style={{ position: 'absolute', top: '18px', right: '17px' }}>
```

```
    {stars}
  </div>
);
}

render() {
  const { cardData } = this.state;

  return (
    <>
      <div>
        <div className='main-image'>
          <img src={mainImg} alt="Mountains" />
          <div className='title'>
            <h1>ПОХОДИ І СХОДЖЕННЯ В КАРПАТИ</h1>
            <Link to="/Survey">
              <Button variant="warning" className="button-tour">
                Підібрати тур
              </Button>
            </Link>
          </div>
        </div>
      </div>
    </div>
  );
}
```

```

<Container>

  <h2 className='text-center mt-5'>Пішохідні маршрути в Карпатах</h2>

  <CardGroup className='card-group'>

    <Row xs={1} md={3} className="g-4">

      {cardData.map((card, idx) => (

        <Col key={idx}>

          <Card className='card' style={{ borderRadius: '30px' }}>

            <Card.Img className='card-style' style={{ borderRadius: '30px' }} src={card.imgSrc} />

            <Card.Body className="card-img-overlay" style={{ display: 'flex', flexDirection: 'column',
alignItems: 'center' }}>

              {this.renderStars(card.rating)}

              <h5 style={{ color: '#1D1515', textAlign: 'center', fontSize: '14px', fontStyle: 'normal',
fontWeight: '700', lineHeight: '16px', margin: '56px' }}>{card.title}</h5>

              <div style={{ marginTop: '100px', display: 'flex', textAlign: 'center', alignItems:
'center', gap: '1.125rem', position: 'absolute' }}>

                <img src={clockImg} alt='Clock' style={{ display: 'inline-block' }}/>

                <h5 style={{ color: '#1D1515', textAlign: 'center', fontSize: '14px', fontStyle:
'normal', fontWeight: '500', lineHeight: '16px', position: 'relative', whiteSpace: 'nowrap', width: 'fit-
content', margin: '10px' }}>{card.duration}</h5>

                <img src={mapImg} alt='Map' style={{ display: 'inline-block' }}/>

                <h5 style={{ color: '#1D1515', textAlign: 'center', fontSize: '14px', fontStyle:
'normal', fontWeight: '500', lineHeight: '16px', position: 'relative', whiteSpace: 'nowrap', width: 'fit-
content', margin: '10px' }}>{card.distance}</h5>

              </div>

              <Link to={card.page}>

```



```

        <IoIosArrowRoundForward style={{ width: '60px', height: '40px', color: '#fff', position:
'absolute', bottom: '10px', right: '10px' }}/>

        </Link>

    </Card.Body>

</Card>

</Col>

    )})

</Row>

</CardGroup>

</Container>

</div>

</>

);

}

}

```

Код сторінки Directory.js

```

export default class Directory extends Component {

  constructor(props) {

    super(props);

    this.state = {

      zakarpattyaData: [],

      frankovskData: [],

```

```
    lvivData: [],
    chernivtsiData: [],
  };
}

componentDidMount() {
  this.fetchZakarpattyaData();
  this.fetchFrankovskData();
  this.fetchLvivData();
  this.fetchChernivtsiData();
}

fetchZakarpattyaData = async () => {
  try {
    const response = await fetch('/api/directory/zakarpattya');
    if (!response.ok) {
      throw new Error('Failed to fetch zakarpattya data');
    }
    const zakarpattyaData = await response.json();
    this.setState({ zakarpattyaData });
  } catch (error) {
    console.error('Error fetching zakarpattya data:', error.message);
  }
}
```

```
    }  
};  
  
fetchFrankovskData = async () => {  
  try {  
    const response = await fetch('/api/directory/frankovsk');  
    if (!response.ok) {  
      throw new Error('Failed to fetch frankovsk data');  
    }  
    const frankovskData = await response.json();  
    this.setState({ frankovskData });  
  } catch (error) {  
    console.error('Error fetching frankovsk data:', error.message);  
  }  
};  
  
fetchLvivData = async () => {  
  try {  
    const response = await fetch('/api/directory/lviv');  
    if (!response.ok) {  
      throw new Error('Failed to fetch lviv data');  
    }  
  }  
}
```

```
    const lvivData = await response.json();
    this.setState({ lvivData });
  } catch (error) {
    console.error('Error fetching lviv data:', error.message);
  }
};

fetchChernivtsiData = async () => {
  try {
    const response = await fetch('/api/directory/chernivtsi');
    if (!response.ok) {
      throw new Error('Failed to fetch chernivtsi data');
    }
    const chernivtsiData = await response.json();
    this.setState({ chernivtsiData });
  } catch (error) {
    console.error('Error fetching chernivtsi data:', error.message);
  }
};

render() {
  const { zakarpattyaData } = this.state;
```

```
const { frankovskData } = this.state;  
const { lvivData } = this.state;  
const {chernivtsiData} = this.state;  
return (
```

```
<Container className='mt-3'>
```

```
<Tab.Container id='left-tabs-example' defaultActiveKey='first'>
```

```
<Row>
```

```
<Col sm={3}>
```

```
<Nav variant='pills' className='flex-column mt-3'>
```

```
<Nav.Item>
```

```
<Nav.Link eventKey='first'>Контакти рятувальних служб</Nav.Link>
```

```
</Nav.Item>
```

```
<Nav.Item>
```

```
<Nav.Link eventKey='second'>Контакти прикордоної служби</Nav.Link>
```

```
</Nav.Item>
```

```
<Nav.Item>
```

```
<Nav.Link eventKey='third'>Транспортні послуги та трансфер</Nav.Link>
```

```
</Nav.Item>
```

```
<Nav.Item>
```

```
<Nav.Link eventKey='fourth'>Військовий стан і походи</Nav.Link>
```

```
</Nav.Item>
```

```

    </Nav>
  </Col>
  <Col sm={9}>
    <Tab.Content>
      <Tab.Pane eventKey='first'>
        <h1 className='text-center'>Контакти гірських пошуково-рятувальних підрозділів в
Карпатах</h1>
        <Container>
          <h3 className='text-center mt-5'>Закарпатська область</h3>
          <Table className='mt-5'>
            <tbody>
              {zakarpattyaData.map((zakarpattya, idx) => (
                <React.Fragment key={idx}>
                  <tr>
                    <th>{zakarpattya.division_name}</th>
                    <td>{zakarpattya.location}</td>
                  </tr>
                  <tr>
                    <td>{zakarpattya.contact_person}</td>
                    <td>{zakarpattya.contact_number}</td>
                  </tr>
                </React.Fragment>
              )

```

```
    ))}
  </tbody>
</Table>

<h3 className='text-center mt-5'>Івано-Франківська область</h3>
<Table  className='mt-5'>
  <tbody>
    {frankovskData.map((frankovsk, idx) => (
      <React.Fragment key={idx}>
        <tr>
          <th>{frankovsk.division_name}</th>
          <td>{frankovsk.location}</td>
        </tr>
        <tr>
          <td>{frankovsk.contact_person}</td>
          <td>{frankovsk.contact_number}</td>
        </tr>
      </React.Fragment>
    ))}
  </tbody>
</Table>

<h3 className='text-center mt-5'>Львівська область</h3>
```

```
<Table className='mt-5'>
  <tbody>
    {lvivData.map((lviv, idx) => (
      <React.Fragment key={idx}>
        <tr>
          <th>{lviv.division_name}</th>
          <td>{lviv.location}</td>
        </tr>
        <tr>
          <td>{lviv.contact_person}</td>
          <td>{lviv.contact_number}</td>
        </tr>
      </React.Fragment>
    ))}
  </tbody>
</Table>

<h3 className='text-center mt-5'>Чернівецька область</h3>
<Table className='mt-5'>
  <tbody>
    {chernivtsiData.map((chernivtsi, idx) => (
      <React.Fragment key={idx}>
```



```
        <tr>
          <th>{chernivtsi.division_name}</th>
          <td>{chernivtsi.location}</td>
        </tr>
        <tr>
          <td>{chernivtsi.contact_person}</td>
          <td>{chernivtsi.contact_number}</td>
        </tr>
      </React.Fragment>
    )))
  </tbody>
</Table>
</Container>
</Tab.Pane>
```

Код сторінки Survey.js

```
const Survey = ({ cardData, setCardData }) => {
  const [question, setQuestion] = useState(1);
  const [showCard, setShowCard] = useState(false);
  const [selectedCardId, setSelectedCardId] = useState(null);
```

```
useEffect(() => {
  fetchCardData(selectedCardId);
}, [selectedCardId]);

const fetchCardData = async (id) => {
  try {
    const response = await fetch(`/api/home/survey/${id}`);
    const data = await response.json();
    console.log('Data from server:', data);
    setCardData([data]);
  } catch (error) {
    console.error('Error fetching card data:', error);
  }
};

const handleAnswer = (answer) => {
  switch (question) {
    case 1:
      if (answer === 'alone') setQuestion(2);
      else if (answer === 'partner') setQuestion(13);
```

```
    else if (answer === 'group') setQuestion(16);
    else if (answer === 'child') setQuestion(21);
    break;
case 2:
    if (answer === 'yes') setQuestion(3);
    else if (answer === 'no') setQuestion(10);
    break;
case 3:
    if (answer === 'excellent') setQuestion(4);
    else if (answer === 'great') setQuestion(6);
    break;
case 4:
    if (answer === 'yes') setQuestion(5);
    else if (answer === 'no') setQuestion(9);
    break;
case 5:
    if (answer === 'yes') {
        setSelectedCardId(76);
        setShowCard(true);
    } else if (answer === 'no') {
        setSelectedCardId(5);
        setShowCard(true);
    }
}
```

```
    }  
    break;  
case 6:  
    if (answer === 'yes') {  
        setQuestion(7);  
    } else if (answer === 'no') {  
        setQuestion(8);  
    }  
    break;  
case 7:  
    if (answer === 'yes') {  
        setSelectedCardId(16);  
        setShowCard(true);  
    } else if (answer === 'no') {  
        setSelectedCardId(8);  
        setShowCard(true);  
    }  
    break;  
case 8:  
    if (answer === 'lake') {  
        setSelectedCardId(6);  
        setShowCard(true);  
    }
```

```
} else if (answer === 'waterfall') {  
    setSelectedCardId(11);  
    setShowCard(true);  
}  
break;  
  
    case 9:  
if (answer === 'area') {  
    setSelectedCardId(12);  
    setShowCard(true);  
} else if (answer === 'forest') {  
    setSelectedCardId(15);  
    setShowCard(true);  
}  
break;  
case 10:  
if (answer === 'yes') {  
    setQuestion(11);  
} else if (answer === 'no') {  
    setQuestion(12);  
}  
break;
```

```
case 11:
    if (answer === 'winter') {
        setSelectedCardId(78);
        setShowCard(true);
    } else if (answer === 'autumn') {
        setSelectedCardId(80);
        setShowCard(true);
    } else if (answer === 'summer') {
        setSelectedCardId(14);
        setShowCard(true);
    } else if (answer === 'spring') {
        setSelectedCardId(82);
        setShowCard(true);
    }
    break;
case 12:
    if (answer === 'view') {
        setSelectedCardId(4);
        setShowCard(true);
    } else if (answer === 'lake') {
        setSelectedCardId(2);
        setShowCard(true);
```

```
    } else if (answer === 'waterfall') {  
        setSelectedCardId(84);  
        setShowCard(true);  
    }  
    break;  
case 13:  
    if (answer === 'yes') setQuestion(14);  
    else if (answer === 'no') setQuestion(15);  
    break;  
case 14:  
    if (answer === 'yes') {  
        setSelectedCardId(10);  
        setShowCard(true);  
    } else if (answer === 'no') {  
        setSelectedCardId(86);  
        setShowCard(true);  
    }  
    break;  
case 15:  
    if (answer === 'yes') {  
        setSelectedCardId(4);  
        setShowCard(true);
```

```
    } else if (answer === 'no') {  
        setSelectedCardId(7);  
        setShowCard(true);  
    }  
    break;  
case 16:  
    if (answer === 'yes') {  
        setQuestion(17);  
    } else if (answer === 'no') {  
        setQuestion(19);  
    }  
    break;  
case 17:  
    if (answer === 'yes') {  
        setQuestion(18);  
    } else if (answer === 'no') {  
        setQuestion(20);  
    }  
    break;  
case 18:  
    if (answer === 'yes') {  
        setSelectedCardId(13);
```



```
        setShowCard(true);
    } else if (answer === 'no') {
        setSelectedCardId(2);
        setShowCard(true);
    }
    break;
case 19:
    if (answer === 'yes') {
        setSelectedCardId(88);
        setShowCard(true);
    } else if (answer === 'no') {
        setSelectedCardId(90);
        setShowCard(true);
    }
    break;
case 20:
    if (answer === 'yes') {
        setSelectedCardId(4);
        setShowCard(true);
    } else if (answer === 'no') {
        setSelectedCardId(2);
        setShowCard(true);
    }
```

```
    }  
    break;  
case 21:  
    if (answer === 'yes') {  
        setSelectedCardId(90);  
        setShowCard(true);  
    } else if (answer === 'no') {  
        setQuestion(22);  
    }  
    break;  
case 22:  
    if (answer === 'no') {  
        setSelectedCardId(84);  
        setShowCard(true);  
    } else if (answer === 'yes') {  
        setQuestion(23);  
    }  
    break;  
case 23:  
    if (answer === 'yes') {  
        setSelectedCardId(78);  
        setShowCard(true);  
    }
```

```
    } else if (answer === 'no') {  
        setSelectedCardId(3);  
        setShowCard(true);  
    }  
    break;  
    default:  
        break;  
    }  
};
```

...

```
const renderQuestion = () => {  
    const buttonStyle = {  
        margin: '5px',  
        color: '#fff',  
        background: 'orange',  
        border: 'none',  
        padding: '10px 20px',  
        borderRadius: '5px',  
        cursor: 'pointer',  
    };  
    switch (question) {  
        case 1:
```

```
return (  
  <div style={centeredStyle}>  
    <div>  
      <p>З ким Ви плануєте підти в похід?</p>  
    </div>  
    <div>  
      <Button style={buttonStyle} onClick={() => handleAnswer('alone')}>Один</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('partner')}>Партнер</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('group')}>Група 3+ людей</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('child')}>Дитина/діти</Button>  
    </div>  
  </div>  
);  
case 2:  
return (  
  <div style={centeredStyle}>  
    <div>  
      <p>Ви вже ходили в походи?</p>  
    </div>  
    <div>  
      <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Ні</Button>  
    </div>  
  </div>  
);
```

```
        </div>
    );
case 3:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Як Ви оцінюєте свою фізичну підготовку?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('excellent')}>Відмінно</Button>
                <Button style={buttonStyle} onClick={() => handleAnswer('great')}>Добре</Button>
            </div>
        </div>
    );
case 4:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Бажаєте піднятися на найвищі вершини?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
                <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Ні</Button>
            </div>
        </div>
    );
```

```
        </div>
    </div>
);
case 5:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Вас цікавлять історичні пам'ятки?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
                <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
            </div>
        </div>
    );
case 6:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Вас цікавлять історичні пам'ятки?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
```

```
    <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
  </div>
</div>
);
case 7:
return (
  <div style={centeredStyle}>
    <div>
      <p>Бажаєте відвідати озеро?</p>
    </div>
    <div>
      <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
      <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
    </div>
  </div>
);
case 8:
return (
  <div style={centeredStyle}>
    <div>
      <p>Що би Ви хотіли відвідати?</p>
    </div>
    <div>
```

```
<Button style={buttonStyle} onClick={() => handleAnswer('lake')}>Озеро</Button>
<Button style={buttonStyle} onClick={() => handleAnswer('waterfall')}>Водоспад</Button>
</div>
</div>
);
case 9:
return (
  <div style={centeredStyle}>
    <div>
      <p>Що Вам більше подобається?</p>
    </div>
    <div>
      <Button style={buttonStyle} onClick={() => handleAnswer('forest')}>Затишний ліс</Button>
      <Button style={buttonStyle} onClick={() => handleAnswer('area')}>Відкрита місцевість</Button>
    </div>
  </div>
);
case 10:
return (
  <div style={centeredStyle}>
    <div>
      <p>Вам подобаються легкі прогулянки?</p>
    </div>
```



```
<div>
  <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
  <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
</div>
</div>
);
case 11:
  return (
    <div style={centeredStyle}>
      <div>
        <p>В яку пору року Ви хочете підти в похід?</p>
      </div>
      <div>
        <Button style={buttonStyle} onClick={() => handleAnswer('winter')}>Зима</Button>
        <Button style={buttonStyle} onClick={() => handleAnswer('autumn')}>Осінь</Button>
        <Button style={buttonStyle} onClick={() => handleAnswer('summer')}>Літо</Button>
        <Button style={buttonStyle} onClick={() => handleAnswer('spring')}>Весна</Button>
      </div>
    </div>
  );
case 12:
  return (
    <div style={centeredStyle}>
```

```

<div>
  <p>Що Ви очікуєте від похода?</p>
</div>

<div>
<Button style={buttonStyle} onClick={() => handleAnswer('view')}>Гарні краєвиди</Button>
<Button style={buttonStyle} onClick={() => handleAnswer('lake')}>Відвідати озеро</Button>
<Button style={buttonStyle} onClick={() => handleAnswer('waterfall')}>Відвідати водоспад</Button>
</div>
</div>

);
case 13:
return (
  <div style={centeredStyle}>
    <div>
      <p>Ви готові до більш вимогливого походу з ночівлею під відкритим небом?</p>
    </div>

    <div>
      <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
      <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
    </div>
  </div>

);
case 14:

```

```
return (  
  <div style={centeredStyle}>  
    <div>  
      <p>Чи готові ви до походу протягом наступних двох тижнів?</p>  
    </div>  
  
    <div>  
      <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>  
    </div>  
  </div>  
);  
  
case 15:  
  return (  
    <div style={centeredStyle}>  
      <div>  
        <p>Бажаєте відвідати водоспад?</p>  
      </div>  
  
      <div>  
        <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
        <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>  
      </div>  
    </div>  
  );
```

case 16:

```
return (  
  <div style={centeredStyle}>  
    <div>  
      <p>Ви вже ходили в походи?</p>  
    </div>  
    <div>  
      <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>  
    </div>  
  </div>  
);
```

case 17:

```
return (  
  <div style={centeredStyle}>  
    <div>  
      <p>Чи є у Вас які-небудь медичні обмеження?</p>  
    </div>  
    <div>  
      <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
      <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>  
    </div>  
  </div>  
);
```

```
);  
case 18:  
  return (  
    <div style={centeredStyle}>  
      <div>  
        <p>Бажаєте відвідати водоспад?</p>  
      </div>  
      <div>  
        <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
        <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>  
      </div>  
    </div>  
  );  
case 19:  
  return (  
    <div style={centeredStyle}>  
      <div>  
        <p>Бажаєте відвідати туристичні місця? </p>  
      </div>  
      <div>  
        <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>  
        <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>  
      </div>  
    </div>  
  );
```

```
        </div>
    );
case 20:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Бажаєте підкорити вершину?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
                <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
            </div>
        </div>
    );
case 21:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Вам подобаються легкі прогулянки?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
                <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
            </div>
        </div>
    );
```

```
        </div>
    </div>
);
case 22:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Бажаєте відвідати пам'ятки?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
                <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
            </div>
        </div>
    );
case 23:
    return (
        <div style={centeredStyle}>
            <div>
                <p>Бажаєте відвідати водоспад?</p>
            </div>
            <div>
                <Button style={buttonStyle} onClick={() => handleAnswer('yes')}>Так</Button>
```

```
    <Button style={buttonStyle} onClick={() => handleAnswer('no')}>Hi</Button>
  </div>
</div>

);
default:
  return null;
}
};
```