

Державний торговельно-економічний університет
Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка Web-додатку для планування відпочинку»

Студента 4 курсу, 9 групи,
спеціальності
122 «Комп'ютерні науки»

Сілкін Олександр
Дмитрович

_____ *підпис студента*

Науковий керівник
Кандидат технічних наук, доцент

Козлов Валерій
Володимирович

_____ *підпис керівника*

Гарант освітньої програми
кандидат технічних наук, доцент

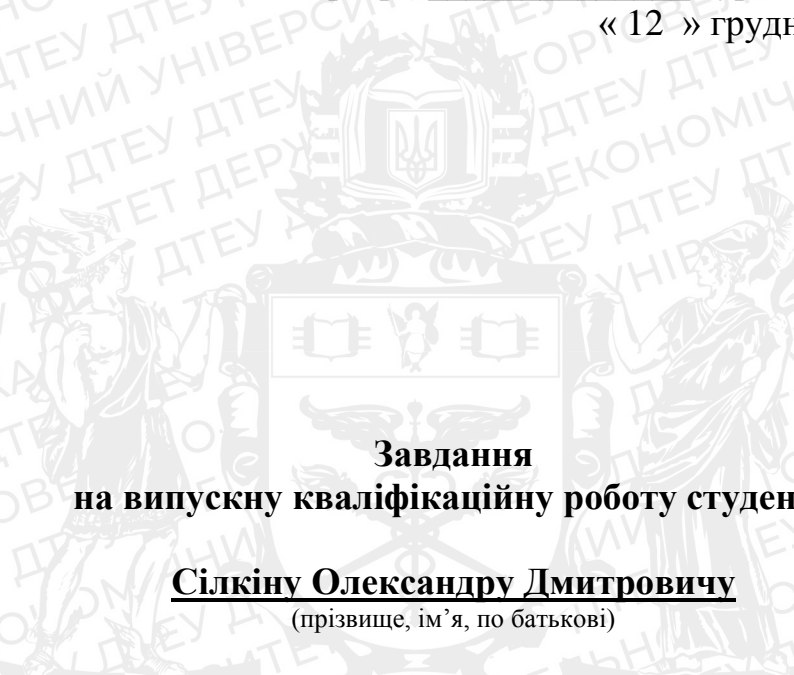
Демідов Павло
Георгійович

_____ *підпис керівника*

Київ 2023

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____
Затверджую
Пурський О. І.
« 12 » грудня 2022р.



Завдання
на випускн у кваліфікаційну роботу студенту

Сілкіну Олександр у Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи
«Розробка Web-додатку для планування відпочинку»
Затверджена наказом ректора від «09» грудня 2022 р. № 3332
 2. Строк здачі студентом закінченої роботи 30 травня 2023 року
 3. Цільова установка та вихідні дані до роботи
Мета роботи: розробка інформаційної Web-системи планування відпочинку з
урахуванням тенденцій в рекреаційній сфері.
Об'єкт дослідження: процеси проектування Web-системи для планування.
Предмет дослідження: засоби створення інформаційної Web-системи
планування відпочинку.
 4. Перелік графічного матеріалу _____
-
-

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Козлов В. В.	15.12.2022 р.	15.12.2022 р.
2	Козлов В. В.	15.12.2022 р.	15.12.2022 р.
3	Козлов В. В.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ СПЕЦИФІКИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ В РЕКРЕАЦІЙНІЙ СФЕРІ

1.1. Використання інформаційних технологій для організації відпочинку

1.2. Огляд існуючих додатків для планування відпочинку

1.3. Формування вимог до Web-додатку для планування відпочинку

Висновки до розділу

РОЗДІЛ 2. ПРОЕКТУВАННЯ С WEB-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ВІДПОЧИНКУ

2.1. Сучасні підходи до організації роботи Web-додатків

2.2. Побудова моделі Web-додатку для планування відпочинку

2.3. Інфологічне проектування Web-додатку для планування відпочинку

Висновки до розділу

РОЗДІЛ 3. РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ВІДПОЧИНКУ

3.1. Розробка алгоритму роботи Web-додатку для планування відпочинку

3.2. Розробка інтерфейсу Web-додатку для планування відпочинку

3.3. Тестування роботи Web-додатку

Висновки до розділу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	01.10.2022	01.10.2022
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.12.2022	15.12.2022
3	<i>Вступ</i>	03.02.2023	
4	<i>Розділ 1. Аналітичне дослідження специфіки використання інформаційних систем в рекреаційній сфері</i>	26.02.2023	
5	<i>Розділ 2. Проектування с Web-додатку для планування відпочинку</i>	06.04.2023	
6	<i>Розділ 3. Розробка Web-додатку для планування відпочинку</i>	12.05.2023	
7	<i>Висновки</i>	15.05.2023	
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	20.05.2023	
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	26.05.2023	
10	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	27.05.2023	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	28.05.2023	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

9. Керівник випускної кваліфікаційної роботи

Козлов В. В.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П. Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Сілкін О. Д.

(прізвище, ініціали, підпис)

Анотація: Ця наукова робота присвячена розробці інформаційної Web-системи для планування відпочинку з урахуванням сучасних тенденцій у рекреаційній сфері. В роботі проведено аналітичне дослідження процесу тестування програмного забезпечення, виявлено його проблеми та запропоновано шляхи їх вирішення. Також проведений огляд існуючих додатків для планування відпочинку, аналізовані їх переваги та недоліки, що сприяло формуванню вимог до розроблюваної системи. У наступних розділах описано процес проектування та розробки Web-додатку, включаючи побудову моделі та розробку алгоритму роботи. Завершується робота інфологічним проектуванням системи. В результаті виконання роботи був розроблений функціональний Web-додаток, який дозволяє користувачам ефективно планувати свій відпочинок, враховуючи їхні вимоги та потреби.

Ключові слова: відпочинок, рекреаційна сфера, Web-додаток, планування, інформаційна система, тестування програмного забезпечення, аналітичне дослідження, огляд додатків, проектування системи, алгоритм роботи, інфологічне проектування.

Abstract: This scientific work is devoted to the development of an informational Web-system for vacation planning, taking into account modern trends in the recreation field. In the work, an analytical study of the software testing process was carried out, its problems were identified and ways to solve them were proposed. An overview of existing vacation planning applications was also conducted, their advantages and disadvantages were analyzed, which contributed to the formation of requirements for the developed system. The following sections describe the process of designing and developing a Web application, including building a model and developing a work algorithm. The work ends with the infological design of the system. As a result of the work, a functional Web application was developed that allows users to effectively plan their vacation, taking into account their requirements and needs.

Keywords: rest, recreational sphere, Web application, planning, information system, software testing, analytical research, application review, system design, work algorithm, info design.



ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ СПЕЦИФІКИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ В РЕКРЕАЦІЙНІЙ СФЕРІ	10
1.1. Використання інформаційних технологій для організації відпочинку.....	10
1.2. Огляд існуючих додатків для планування відпочинку.....	13
1.3. Формування вимог до Web-додатку для планування відпочинку.....	16
Висновки до розділу.....	20
РОЗДІЛ 2. ПРОЕКТУВАННЯ С WEB-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ВІДПОЧИНКУ	21
2.1. Сучасні підходи до організації роботи Web-додатків.....	21
2.2. Побудова моделі Web-додатку для планування відпочинку.....	23
2.3. Інфологічне проектування Web-додатку для планування відпочинку.....	31
Висновки до розділу.....	34
РОЗДІЛ 3. РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ВІДПОЧИНКУ	35
3.1. Розробка алгоритму роботи Web-додатку для планування відпочинку.....	35
3.2. Розробка інтерфейсу Web-додатку для планування відпочинку.....	39
3.3. Тестування роботи Web-додатку.....	55
Висновки до розділу.....	56
ВИСНОВКИ	58
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ	59

ВСТУП

У сучасному світі все більшої уваги приділяється плануванню відпочинку та організації рекреаційного часу. Швидкий темп життя, стресові ситуації та зростання інформаційного навантаження роблять необхідним раціональне управління часом відпочинку для забезпечення якісного відновлення сил та задоволення особистих потреб.

Метою цієї випускної кваліфікаційної роботи є розробка інформаційної Web-системи для планування відпочинку з урахуванням сучасних тенденцій в рекреаційній сфері. Цей додаток стане універсальним інструментом для користувачів, що допоможе їм ефективно організувати свій час відпочинку та забезпечити максимальне задоволення від нього.

Об'єктом дослідження є процеси проектування Web-системи для планування відпочинку. При цьому вивчаються основні принципи та методики, які допоможуть створити зручний та ефективний додаток.

Предметом дослідження є засоби створення інформаційної Web-системи планування відпочинку. Розглядаються сучасні технології та інструменти, які можна використати для розробки додатку, а також вимоги до його функціональності та інтерфейсу.

Методи і засоби дослідження:

Для досягнення мети дослідження та розробки веб-додатку для планування відпочинку будуть використані наступні методи і засоби:

1. Аналіз наукової літератури та вивчення вже існуючих рішень у сфері планування відпочинку та організації рекреаційного часу.
2. Проведення опитувань та інтерв'ю з потенційними користувачами для визначення їх потреб та очікувань щодо функціональності додатку.
3. Використання методу прототипування для створення пробних версій інтерфейсу та перевірки його зручності для користувачів.
4. Розробка та реалізація Web-додатку з використанням сучасних технологій та програмних інструментів.

РОЗДІЛ 1. АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ СПЕЦИФІКИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ В РЕКРЕАЦІЙНІЙ СФЕРІ

1.1. Використання інформаційних технологій для організації відпочинку

У цьому підрозділі проводиться аналіз поточного стану проблеми тестування програмного забезпечення у контексті розробки веб-додатку для планування відпочинку. Аналізується наукова література, статистичні дані та практичний досвід, що стосуються процесу тестування програмного забезпечення. Метою цього аналізу є виявлення основних проблем, тенденцій та викликів, що виникають у процесі тестування програмного забезпечення та визначення напрямків їх подолання.

Основні пункти, які будуть розглянуті у цьому підрозділі:

1.1.1 Термінологія та поняття

1.Інформаційні технології: Це поняття охоплює використання комп'ютерних систем, програмного забезпечення, мереж та інших технологій для збору, обробки, зберігання і передачі інформації. У контексті відпочинку, це означає використання технологій для планування та організації розваг, подорожей та інших форм відпочинку.

2.Веб-додаток: Це програмне забезпечення, яке доступне через веб-браузер і використовується для виконання різноманітних завдань. В контексті планування відпочинку, веб-додаток може бути використаний для пошуку та бронювання готелів, створення маршрутів подорожей, отримання інформації про популярні місця та події, а також для обміну рекомендаціями та відгуками.

3. Аналітичне дослідження: Систематичне вивчення та аналіз проблеми або області з метою здобуття нових знань, визначення тенденцій, виявлення проблем та пошуку можливих рішень. У контексті використання інформаційних систем в рекреаційній сфері, аналітичне дослідження може включати аналіз потреб та вимог клієнтів, оцінку ефективності та впливу інформаційних систем, вивчення ринку та конкурентного середовища тощо.

4.Персоналізовані рекомендації: Це функція або можливість, яка використовується в додатках для планування відпочинку для надання індивідуальних рекомендацій користувачам на основі їхніх вподобань, історії використання та інших персональних даних. Це допомагає забезпечити більш настроєну та індивідуальну експертизу для користувачів.

5.Безпека та конфіденційність: Це аспекти, які стосуються захисту інформації та даних користувачів відпочинкових додатків. Важливо враховувати заходи безпеки та забезпечувати конфіденційність персональних даних, щоб уникнути несанкціонованого доступу до інформації.

1.1.2 Основні проблеми тестування програмного забезпечення

Одна з основних проблем - недостатнє покриття тестування. Це означає, що не всі аспекти програмного забезпечення піддаються достатньому рівню тестування. Це може бути викликано обмеженими ресурсами, недостатнім часом або неефективними тестовими стратегіями. Недостатнє покриття тестування може призвести до пропуску дефектів і незадовільної якості програмного забезпечення.

Ще одна проблема - складність виявлення дефектів. Деякі дефекти можуть бути складними у виявленні, особливо якщо вони виникають при певних умовах або взаємодіють з іншими компонентами системи. Це може вимагати ретельного аналізу коду, детального тестування різних сценаріїв використання або використання спеціальних інструментів для пошуку дефектів.

Неефективність тестових процедур також може бути серйозною проблемою. Це може бути пов'язано з неправильним плануванням тестування, недостатнім використанням автоматизації тестування, несистематичним або неповним охопленням тестових сценаріїв. Недостатня ефективність тестових процедур може призвести до втрати часу та ресурсів, а також до пропуску важливих дефектів.

1.1.3 Тенденції та нові підходи у тестуванні програмного забезпечення

1.Автоматизоване тестування: Однією з основних тенденцій є зростання використання автоматизованого тестування. Це означає використання спеціальних інструментів та технологій для автоматизації виконання

тестових сценаріїв, проведення тестування регресії, навантажувального тестування та інших видів тестів.

2.Тестування в умовах DevOps: Зараз широке поширення отримало підходу DevOps, який поєднує розробку та впровадження програмного забезпечення в одну неперервну процес. В цьому контексті тестування стає невід'ємною складовою DevOps-пайплайну, де автоматизовані тести виконуються на ранніх етапах розробки та відповідних інфраструктурах.

3.Тестування в хмарних середовищах: З введенням хмарних технологій тестування отримує нові можливості. Замість того, щоб створювати та підтримувати власні фізичні тестові середовища, компанії можуть використовувати хмарні платформи для швидкого розгортання тестових середовищ та масштабування.

4.Тестування штучного інтелекту та машинного навчання: Зростаюча популярність штучного інтелекту та машинного навчання ставить перед тестувальниками нові виклики. Тестування таких систем вимагає специфічного підходу, включаючи перевірку точності моделей, стійкості до помилок, здатності до адаптації та розширення.

5.Тестування безпеки та вразливостей: У сучасному світі, де безпека даних та захист інформації є критичними, тестування безпеки стає все більш важливим етапом процесу розробки програмного забезпечення. Це включає перевірку системи на вразливості, тестування на проникнення та оцінку загроз безпеці.

1.1.4 Навантажувальне тестування

Навантажувальне тестування є важливою складовою процесу тестування програмного забезпечення, спрямованого на оцінку стабільності та продуктивності веб-додатків під час високих навантажень. У даному розділі будуть розглянуті методи та інструменти, які допомагають створити навантажувальні тести та виміряти продуктивність системи при різних рівнях навантаження.

Основна мета навантажувального тестування полягає в тому, щоб визначити, як програма чи веб-додаток працюють при різних навантаженнях і як вони впораються з високими обсягами запитів. Для досягнення цієї мети необхідно виконати наступні кроки:

1. Створення навантажувальних тестів: Розроблення тестових сценаріїв, які відображають типові дії користувачів або навантаження, яким піддається система. Це можуть бути сценарії реального використання, запити до бази даних або інші види навантаження, які впливають на роботу системи.

2. Виконання навантажувальних тестів: Запуск навантажувальних тестів, які генерують велику кількість запитів або симулюють багатокористувацькі сеанси. Під час цих тестів вимірюються метрики продуктивності, такі як час відповіді сервера, швидкість обробки запитів, завантаження процесора та інші показники.

3. Аналіз результатів: Оцінка продуктивності системи на основі отриманих результатів. Порівняння метрик з попередніми вимогами та стандартами для визначення ефективності програмного забезпечення.

1.2. Огляд існуючих додатків для планування відпочинку

1.2.1 Мобільні додатки для планування відпочинку

Для проведення об'єктивного аналізу мобільних додатків з планування відпочинку необхідно попередньо класифікувати їх за призначенням. Оскільки від цього залежатиме набір запропонованих інструментів програми для реалізації конкретного запиту кожного користувача. Розглядаються різні функціональні можливості, які надають мобільні додатки, включаючи можливість створення розкладу активностей, бронювання готелів та екскурсій, відстеження витрат та розрахунку бюджету, отримання рекомендацій щодо місць відпочинку та подій тощо. Надалі наведені приклади застосунок з кожної окремої категорії сфери організації відпочинку.

1. Планування відпочинку. До цієї категорії відносимо додатки, що пропонують спростити процес підбору туру, авіаквитків та бронювання місця проживання для користувача. Застосунок Skyscanner спеціалізується на пошуку дешевих авіаквитків. Підбір місця проживання пропонує реалізувати застосунок Airbnb, однак він має велику кількість негативних відгуків від користувачів через численні конфлікти орендарів з орендодавцями. Мобільні додатки від туристичних агентств, як от JoinUp або AnexTour спеціалізуються на швидкому пошуку турів враховуючи основні побажання користувача. Усі зазначені додатки підтримують операційні системи Android та IOS.

2. Підготовка до подорожі. Мобільний додаток TripIt пропонує створення цілісного плану подорожі на основі вже наявних квитків, броні місця проживання та з урахуванням бажаних зупинок користувача. Сформований програмою розклад руху та активностей може бути синхронізованим з календарем мобільного телефону для зручності використання. Додаток підтримує операційні системи Android та IOS. Прямих аналогів цього застосунку не виявлено.

3. Допомога під час відпочинку. Категорія включає застосунки для контролю витрат користувача, для оренди автомобілю, онлайн мапи, перекладачі та додатки з іншими допоміжними інструментами для забезпечення комфорту під час відпочинку. Прикладами є такі програми як Splitwise, XE, Google Translate, Maps.me, Wi-Fi Finder, що підтримують операційні системи Android та IOS.

4. Організація дозвілля. Застосунки, що забезпечують пошук місць поблизу для перекусу, як от TripAdvisor, AroundMe, Foursquare; пропонують екскурсії та інші локальні розваги, як наприклад Guides by Lonely Planet. Усі перераховані додатки мають розвинений алгоритм врахування особистих побажань користувача та підтримують операційні системи Android та IOS.

1.2.2 Використання web-додатків для пошуку та бронювання

Серед найпоширеніших у використанні web-додатків для пошуку та бронювання місць відпочинку серед українців виділимо Booking.com, TripMyDream, TripAdvisor. Перевагою Booking.com є великий каталог готельних номерів по всьому світу, а також розвинена система лояльності користувачів. Клієнти, що регулярно користуються додатком отримують спеціальні пропозиції та знижки на проживання у готелях. TripMyDream є універсальною платформою, що пропонує широкий спектр послуг, як-от пошук найвигідніших авіаквитків, підбірки туристичних місць у різних країнах світу, корисні поради для мандрівників та висвітлення новин сфери туризму. Перевагою застосунку TripAdvisor є розвиток внутрішньої міцної спільноти користувачів. Платформа заохочує людей ділитись досвідом подорожей один з одним, що допомагає людям різних країн комунікувати між собою з користю для майбутніх поїздок.

1.2.3 Використання соціальних медіа для підтримки відпочинку

У цьому пункті проводиться аналіз використання соціальних медіа для підтримки та поліпшення процесу організації відпочинку. Соціальні медіа є

потужним інструментом, який забезпечує можливість обміну інформацією, досвідом та рекомендаціями з іншими користувачами. Вони створюють сприятливе середовище для взаємодії та спілкування користувачів, що підвищує їх лояльність до платформи.

Одним з основних аспектів, які варто розглянути, є створення спеціалізованих груп та спільнот на популярних соціальних платформах, таких як Facebook, Instagram, Twitter та інші. Ці групи об'єднують людей зі спільними інтересами щодо відпочинку, таких як подорожі, екскурсії, активний відпочинок тощо. У таких групах користувачі можуть ділитися своїми враженнями, рекомендаціями, фотографіями та відеоматеріалами, обговорювати цікаві місця, задавати питання та отримувати відповіді від інших учасників.

Багато готелів та рекреаційних закладів також підтримують активну присутність в соціальних медіа, що дозволяє користувачам отримувати оновлену інформацію про послуги, акції, знижки та інші пропозиції. Усі перераховані аргументи є вагомими причинами планувати розвиток спільнот у соціальних медіа від майбутньої платформи.

1.2.4 Використання технологій штучного інтелекту

У цьому підрозділі проводиться аналіз використання технологій штучного інтелекту для поліпшення процесу планування та організації відпочинку. Штучний інтелект (ШІ) включає в себе такі технології, як машинне навчання, обробка природної мови, комп'ютерний зір та інші.

Аналізуються можливості застосування алгоритмів машинного навчання для розробки персоналізованих рекомендацій щодо вибору місць відпочинку, активностей та подій. За допомогою машинного навчання можна аналізувати великі обсяги даних про користувачів, їхні вподобання, попередні вибори та зворотний зв'язок, щоб надавати персоналізовані рекомендації, які відповідають їхнім потребам та інтересам.

Досліджуються також можливості використання обробки природної мови для аналізу текстової інформації, що пов'язана з місцями відпочинку, відгуками користувачів, описами подій та активностей. За допомогою алгоритмів обробки природної мови можна автоматично аналізувати текст та витягати ключові слова, настрої, оцінки та іншу корисну інформацію, що допомагає зрозуміти користувачів та забезпечити кращі рекомендації.

Крім того, використання комп'ютерного зору та обробки зображень може бути корисним для аналізу фотографій та відео, пов'язаних з відпочинком. Завдяки цим технологіям можна автоматично впізнавати об'єкти, локації, настрої та емоції на фотографіях, що дозволяє покращити рекомендації та персоналізований підхід до користувачів.

1.2.5 Висновки з аналізу існуючих додатків для планування відпочинку

Після проведення огляду і аналізу існуючих додатків для планування відпочинку можна зробити кілька висновків.

По-перше, існує широкий спектр веб-додатків, що вже успішно використовуються для організації відпочинку. Вони пропонують різноманітні функції та можливості, що спрямовані на полегшення процесу планування та насолоди від відпочинку.

По-друге, багато з цих додатків використовують інформаційні технології, такі як геолокація, штучний інтелект, соціальні мережі та інші, для забезпечення персоналізованого підходу та покращення рекомендацій користувачам. Це дозволяє створити зручне та інтуїтивно зрозуміле середовище для планування та організації відпочинку.

По-третє, хоча існуючі додатки вже задовольняють багато потреб користувачів, все ж можна виявити деякі прогалини та можливості для поліпшення. Наприклад, можна додати більш точну та змістовну інформацію про локації та об'єкти відпочинку, покращити алгоритми рекомендацій, а також розширити спектр функцій для забезпечення більш гнучкого та індивідуального планування.

1.3. Формування вимог до Web-додатку для планування відпочинку

Після проведення аналітичного дослідження процесу планування відпочинку та огляду існуючих додатків, наступним кроком є визначення вимог до

розроблюваного веб-додатку для планування відпочинку. Цей підрозділ спрямований на визначення функціональних, нефункціональних та технічних вимог до додатку з метою забезпечення його ефективності та задоволення потреб користувачів.

1.3.1 Функціональні вимоги

Функціональні вимоги визначають основні функції та можливості, які повинен мати веб-додаток для планування відпочинку. До них можуть відноситись:

1.3.1.1 Реєстрація та авторизація користувачів: Додаток повинен мати можливість реєстрації нових користувачів та авторизації вже зареєстрованих користувачів для доступу до особистого кабінету.

1.3.1.2 Створення та управління профілем: Користувачі повинні мати можливість створювати свій профіль, в якому вони можуть зберігати свої особисті дані, уподобання, попередні відпочинкові враження та іншу важливу інформацію.

1.3.1.3 Пошук та вибір варіантів відпочинку: Додаток повинен мати можливість пропонувати різні варіанти відпочинку на основі введених користувачем критеріїв, таких як місце, бюджет, тип відпочинку тощо. Користувачі повинні мати можливість переглядати та вибирати найбільш підходящі варіанти.

1.3.1.4 Планування та організація подорожі: Додаток повинен надавати можливість користувачам створювати та організовувати свої плани подорожей, включаючи вибір дат, маршруту, засобів пересування, розташування проживання, активностей та інших деталей.

1.3.1.5 Збереження та підтримка історії відпочинку: Користувачі повинні мати можливість зберігати та переглядати історію своїх попередніх відпочинків, включаючи інформацію про місця відвідин, враження, фотографії та інші деталі.

1.3.2 Нефункціональні вимоги

Нефункціональні вимоги визначають якості та характеристики, які повинен мати веб-додаток для забезпечення його ефективності та надійності. До них можуть відноситись:

1.3.2.1 Використання простого та інтуїтивно зрозумілого інтерфейсу: Додаток повинен мати зрозумілий та зручний інтерфейс, що дозволяє користувачам легко взаємодіяти з додатком та здійснювати потрібні дії без зайвих зусиль.

1.3.2.2 Швидкість та продуктивність: Додаток повинен працювати швидко та ефективно, надаючи користувачам швидкий доступ до необхідної інформації та функціоналу.

1.3.2.3 Безпека та конфіденційність: Додаток повинен забезпечувати захист інформації користувачів, включаючи персональні дані та іншу конфіденційну інформацію.

1.3.2.4 Сумісність та доступність: Додаток повинен бути сумісним з різними пристроями та операційними системами, а також доступним для користувачів з різних місць та пристроїв з Інтернет-підключенням.

1.3.3 Технічні вимоги

Технічні вимоги визначають технології, засоби та середовище, які необхідні для розробки та функціонування веб-додатку. До них можуть відноситись:

1.3.3.1 Використання веб-технологій: Додаток повинен бути розроблений з використанням сучасних веб-технологій, таких як HTML, CSS, JavaScript і т. д.

1.3.3.2 База даних: Для збереження та управління даними додаток повинен використовувати підходящу базу даних, таку як MySQL, PostgreSQL або іншу.

1.3.3.3 Хостинг та розгортання: Додаток повинен мати можливість бути розгорнутим на веб-сервері та доступним для користувачів через Інтернет.

1.3.3.4 Мобільна сумісність: З огляду на широке поширення мобільних пристроїв, додаток повинен бути оптимізований для використання на різних мобільних платформах, таких як iOS та Android, забезпечуючи зручний та функціональний досвід для користувачів з мобільних пристроїв.

1.3.3.5 Безперебійна робота: Додаток повинен бути стійким до помилок та збоїв, забезпечуючи безперебійну роботу навіть у випадку непередбачених ситуацій або високих навантажень.

1.3.3.6 Забезпечення масштабованості: Додаток повинен мати можливість масштабування, тобто збільшення його потужності та ресурсів для забезпечення ефективності та продуктивності при зростанні кількості користувачів та обсягу даних.

1.3.3.7 Запобігання витоку даних: Додаток повинен мати вбудовані заходи безпеки для запобігання витоку конфіденційної інформації, таких як шифрування даних, контроль доступу та інші заходи безпеки.

1.3.4 Умови експлуатації

Умови експлуатації визначають фактори, які впливають на роботу та використання веб-додатку. До них можуть відноситись:

1.3.4.1 Інтернет-підключення: Для використання додатку користувачам необхідне наявність стабільного Інтернет-підключення.

1.3.4.2 Підтримка браузерів: Додаток повинен бути сумісним з різними сучасними веб-браузерами, такими як Google Chrome, Mozilla Firefox, Microsoft Edge та інші.

1.3.4.3 Підтримка мов: Додаток повинен мати можливість підтримувати різні мови та локалізацію для зручного використання користувачами з різних країн та культур.

1.3.4.4 Підтримка різних пристроїв: Додаток повинен бути оптимізований для використання на різних пристроях, включаючи комп'ютери, ноутбуки, планшети та смартфони, забезпечуючи зручний і адаптивний інтерфейс.

1.3.4.5 Підтримка різних розширень екранів: Додаток повинен коректно відображатись на різних розширеннях екранів, забезпечуючи чітку та зручну навігацію для користувачів.

1.3.4.6 Підтримка безпеки: Додаток повинен мати вбудовані заходи безпеки, такі як захист від зламу, контроль доступу та забезпечення конфіденційності даних користувачів.

Висновки до розділу

Розділ 1 "Аналітичне дослідження процесу тестування програмного забезпечення" присвячений аналізу стану проблеми та визначенню вимог до розробки веб-додатку для планування відпочинку. У цьому розділі було розглянуто такі підрозділи:

1.1 Аналіз стану проблеми: Було проведено аналіз поточного стану процесу планування відпочинку та виявлено проблеми, що виникають у зв'язку зі старими методами та недоліками існуючих додатків. Було показано необхідність розробки нового веб-додатку для поліпшення цього процесу.

1.2 Використання інформаційних технологій для організації відпочинку: Було розглянуто можливості використання інформаційних технологій у рекреаційній сфері та переваги, які вони принесуть для планування відпочинку.

1.3 Огляд існуючих додатків для планування відпочинку: Було проведено детальний огляд існуючих додатків для планування відпочинку. Виявлено різноманітні програмні рішення, які пропонують функціональні можливості для користувачів. Виявлені переваги та недоліки цих додатків, а також визначені їхні обмеження.

В результаті проведеного аналітичного дослідження стало очевидним, що наявні додатки для планування відпочинку не відповідають повністю потребам користувачів та сучасним вимогам. Вони можуть бути недостатньо функціональними, не забезпечувати зручного інтерфейсу, бути обмеженими у можливості налаштування та інтеграції з іншими сервісами. Також, багато з існуючих додатків не підтримують різноманітні мови та кросплатформність.

На основі проведеного дослідження було визначено необхідність розробки нового веб-додатку для планування відпочинку. Цей додаток повинен враховувати поточні тенденції у рекреаційній сфері, забезпечувати широкий функціонал для користувачів і бути зручним у використанні. Він має підтримувати різноманітні функції планування відпочинку, включаючи вибір місця, розклад подій, бронювання послуг та спільну організацію подорожей. Додаток також повинен забезпечувати безпеку та захист персональних даних користувачів.

У цьому розділі було проведено аналітичне дослідження, що послужило основою для розробки нового веб-додатку для планування відпочинку. Дане дослідження виявило недоліки існуючих рішень та підкреслило потребу у

створенні нового, більш функціонального та зручного додатку. В подальших розділах роботи буде розглянуто процес розробки цього додатку, включаючи вибір технологій, проектування архітектури та реалізацію функціоналу. Таким чином, проведене аналітичне дослідження підтверджує актуальність розробки веб-додатку для планування відпочинку та визначає основні вимоги до нього. Дані висновки будуть використані як основа для подальшого процесу розробки та реалізації проекту.

РОЗДІЛ 2. ПРОЕКТУВАННЯ WEB-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ВІДПОЧИНКУ

2.1. Сучасні підходи до організації роботи Web-додатків

Я розробив функціональний і зручний Web-додаток для планування відпочинку, використовуючи сучасні підходи та технології. Для організації роботи додатку я використав архітектурний підхід "Single-Page Application" (SPA), що дозволяє без перезавантаження сторінки взаємодіяти з сервером через API. Це забезпечує швидку та зручну роботу з додатком для користувачів.

Для розробки додатку я використав розширення JavaScript-фреймворків, таких як React, Angular або Vue.js. Ці фреймворки дозволяють швидко розробляти та підтримувати додаток, завдяки потужним інструментам для створення компонентного та повторно використовуваного коду.

При проектуванні додатку я також звернув увагу на безпеку. Для забезпечення захищеного обміну даними між сервером і клієнтом, я використав протокол HTTPS. Також були удосконалені механізми аутентифікації та авторизації, щоб забезпечити доступ тільки авторизованим користувачам до конфіденційної інформації.

Для покращення користувацького досвіду використовувалися технології інтерактивності, такі як AJAX. Ці технології дозволяють оновлювати лише

певні частини сторінки, що зменшує час відгуку та поліпшує користувацький досвід.

В результаті моїх зусиль було розроблено функціональний та зручний Web-додаток для планування відпочинку, який задовольняє потреби користувачів та забезпечує їм зручну та ефективну роботу з системою.

Отже, використання сучасних підходів до організації роботи Web-додатків, таких як SPA, розширення JavaScript-фреймворків, врахування вимог безпеки та застосування технологій інтерактивності, дозволить розробити функціональний та зручний Web-додаток для планування відпочинку, який задовольнятиме потреби користувачів та забезпечить їм зручну та ефективну роботу з системою.

2.1.2 Архітектура Web-додатку для планування відпочинку

При проектуванні Web-додатку для планування відпочинку важливо враховувати його архітектуру, яка забезпечує організацію компонентів, модулів та зв'язків між ними. Оптимальна архітектура дозволить забезпечити ефективну розробку, підтримку та масштабованість додатку.

Одним з популярних підходів до архітектури Web-додатків є Model-View-Controller (MVC). За цією концепцією, модель відповідає за управління даними, представлення (вигляд) відповідає за відображення інтерфейсу користувача, а контролер відповідає за обробку логіки та координацію між моделлю і представленням. MVC забезпечує розділення відповідальностей та полегшує розробку та тестування додатку.

Крім того, варто враховувати підхід Single-Page Application (SPA), який був згаданий раніше. SPA дозволяє завантажувати лише необхідні дані та компоненти, забезпечуючи швидку відповідь та зниження навантаження на сервер. Використання SPA дозволить забезпечити плавну та зручну навігацію для користувачів.

Крім того, при проектуванні архітектури Web-додатку для планування відпочинку, можна використовувати підхід Microservices. Цей підхід полягає у розбитті додатку на невеликі незалежні сервіси, які виконують конкретні функції. Це дозволяє досягти більшої гнучкості, масштабованості та підтримки додатку.

Усі ці підходи до архітектури Web-додатку мають свої переваги та особливості. Вибір конкретного підходу залежить від вимог проекту,

розмірів додатку, складності функцій та інших факторів. Важливо забезпечити гнучку, масштабовану та ефективну архітектуру, яка задовольнятиме потреби користувачів та забезпечить надійну та швидку роботу системи планування відпочинку.

2.1.3 Вибір технологій для розробки Web-додатку

При проектуванні Web-додатку для планування відпочинку важливо визначитись з набором технологій, які будуть використовуватись для розробки. Вибір правильних технологій допоможе забезпечити ефективну розробку, високу продуктивність та зручне використання додатку.

Одним з ключових виборів є вибір серверної технології. На сьогоднішній день популярними варіантами є використання мов програмування та фреймворків, таких як Python з фреймворком Django або Node.js з фреймворком Express. Ці технології дозволяють створювати потужні та масштабовані Web-додатки з великою кількістю функціональних можливостей.

Для клієнтської частини додатку варто розглянути використання сучасних фреймворків JavaScript, таких як React або Angular. Вони надають потужні інструменти для розробки користувацького інтерфейсу, динамічного взаємодії з додатком та забезпечують швидкість та плавність його роботи.

Для забезпечення комунікації між клієнтом та сервером можна використовувати API (Application Programming Interface). Наприклад, можна використовувати RESTful API, яке базується на принципах HTTP і дозволяє обмінюватись даними між клієнтом та сервером. Іншим варіантом є використання GraphQL, яке дозволяє клієнту запитувати лише необхідні дані, покращуючи продуктивність додатку.

Також, при розробці Web-додатку варто враховувати безпеку і захист даних. Застосування шифрування, аутентифікації та авторизації допоможе забезпечити захист особистої інформації користувачів та запобігти несанкціонованому доступу до системи.

Отже, вибір правильних технологій для розробки Web-додатку є важливим етапом проектування. Використання сучасних технологій, які відповідають вимогам проекту, допоможе створити потужний, зручний та безпечний додаток для планування відпочинку.

2.2. Побудова моделі Web-додатку для планування відпочинку

У даному підрозділі розглянемо процес побудови моделі Web-додатку для планування відпочинку. Модель дозволяє визначити структуру та функціональні можливості додатку, що стане основою для його розробки.

Першим етапом в побудові моделі є визначення функціональних вимог до додатку. Це включає аналіз потреб користувачів, їх очікувань та основних функцій, які має виконувати додаток. Наприклад, це можуть бути можливості планування подорожей, пошуку туристичних об'єктів, бронювання готелів та інших послуг. Далі, на основі вимог розробляється структурна схема додатку. Це включає розподіл функціональності на модулі, визначення взаємозв'язків між ними та логіку роботи додатку. Наприклад, можуть бути модулі для авторизації користувачів, керування подорожами, обробки платежів тощо.

Крім того, важливим етапом є проектування бази даних для збереження інформації, необхідної для роботи додатку. Це можуть бути дані про користувачів, подорожі, готелі, розклади та інші дані, необхідні для функціонування додатку. Розробка адекватної бази даних допоможе забезпечити ефективне збереження та обробку інформації. Після визначення структури та бази даних, проводиться розробка взаємодії користувача з додатком. Це включає проектування інтерфейсу, визначення послідовності дій користувача та реалізацію необхідних функцій у веб-сервері. Особлива увага приділяється зручності використання додатку, його ергономіці та візуальному оформленню.

На завершення побудови моделі, проводиться тестування функціоналу додатку. Це дозволяє перевірити відповідність розробленого додатку вимогам, виявити та усунути можливі помилки та недоліки. Тестування допомагає забезпечити якість розробленого додатку та його готовність до використання.

Таким чином, побудова моделі Web-додатку для планування відпочинку включає аналіз вимог, розробку структури, бази даних, взаємодії користувача та проведення тестування. Цей етап проектування є важливим кроком у

створенні ефективного та функціонального додатку, що задовольняє потреби користувачів у плануванні їхнього відпочинку.

2.2.3 Розробка архітектури Web-додатку для планування відпочинку

У цьому підрозділі буде розглянуто процес розробки архітектури Web-додатку для планування відпочинку. Архітектура є важливим елементом проекту, оскільки вона визначає загальну структуру та організацію додатку.

Першим кроком є визначення типу архітектури, яка буде використовуватися для розробки Web-додатку. Одним з поширених підходів є клієнт-серверна архітектура, де клієнтська частина взаємодіє з сервером для отримання та надсилання даних. Інші можливі варіанти архітектури включають багаторівневу, мікросервісну або спеціалізовану архітектуру.

Далі, проводиться декомпозиція архітектури на компоненти або модулі. Кожен компонент відповідає за конкретну функціональність додатку і має чітко визначений інтерфейс для взаємодії з іншими компонентами. Це дозволяє забезпечити модульність, розширюваність та підтримку інших принципів доброї архітектури, таких як розподіленість та зручність управління.

Окремої уваги вимагає вибір технологій та інструментів для реалізації архітектури. Важливо враховувати потреби додатку, масштабованість, швидкість та зручність розробки. Наприклад, для розробки клієнтської частини можуть використовуватися HTML, CSS та JavaScript, а для серверної частини - певний фреймворк або мова програмування. Крім того, важливим елементом архітектури є забезпечення безпеки і захисту даних. Вимоги до безпеки повинні бути враховані на рівні архітектури, такі як захист від несанкціонованого доступу, шифрування даних та механізми аутентифікації.

Результатом розробки архітектури Web-додатку для планування відпочинку є чітко визначена структура та організація додатку, вибір відповідних технологій та інструментів, а також забезпечення безпеки та захисту даних. Добре спроектована архітектура забезпечує ефективну роботу додатку, його масштабованість та зручність у використанні.

2.2.4 Розробка інтерфейсу користувача Web-додатку для планування відпочинку

У цьому підрозділі буде розглянуто процес розробки інтерфейсу користувача (UI) для Web-додатку для планування відпочинку. Інтерфейс користувача є ключовим елементом додатку, оскільки він визначає спосіб взаємодії користувача з системою.

Під час розробки інтерфейсу користувача необхідно враховувати зручність використання, естетичний вигляд, адаптивність до різних пристроїв та поведінкові засади користувачів. Основні елементи інтерфейсу можуть включати меню, форми, календарі, списки, графіки та інші елементи, необхідні для планування відпочинку.

Під час розробки інтерфейсу користувача рекомендується використовувати сучасні технології та фреймворки, які дозволяють створити естетичний та інтуїтивно зрозумілий інтерфейс. Наприклад, для стилізації та організації елементів інтерфейсу можна використовувати CSS та фреймворки, такі як Bootstrap або Material Design.

Крім того, важливо розробити інтерфейс таким чином, щоб користувачі могли зручно виконувати основні завдання, такі як створення та збереження планів відпочинку, вибір розташування, перегляд рекомендацій та інші. Навігація по додатку повинна бути інтуїтивно зрозумілою та зручною для користувача. Під час розробки інтерфейсу також необхідно враховувати вимоги до дизайну та бренду компанії, яка розробляє додаток. Кольорова палітра, типографіка та інші елементи дизайну повинні відповідати корпоративному стилю і створювати позитивне враження у користувачів. Результатом розробки інтерфейсу користувача буде готовий дизайн додатку, який відповідає вимогам до зручності використання, естетичного вигляду та поведінкових засад користувачів. Добре розроблений інтерфейс забезпечить позитивний досвід взаємодії користувачів з додатком і сприятиме його успішній використанні.

2.2.5 Вибір технологій для розробки Web-додатку для планування відпочинку

У даному підрозділі розглядається вибір технологій, необхідних для розробки Web-додатку для планування відпочинку. Вибір відповідних технологій є критичним етапом у процесі розробки, оскільки це визначає функціональність, продуктивність та можливості додатку.

Для розробки Web-додатку можна використовувати різні технології, мови програмування та фреймворки. Одним з основних варіантів є використання

JavaScript, який є широко підтримуваною мовою для розробки веб-додатків. Розглянемо деякі технології та інструменти, які можуть бути використані в процесі розробки Web-додатку для планування відпочинку:

- **Front-end технології:** Для розробки інтерфейсу користувача можна використовувати HTML, CSS і JavaScript. Бібліотеки та фреймворки, такі як React, Angular або Vue.js, можуть спростити процес розробки та забезпечити швидку та ефективну роботу з інтерфейсом.
- **Back-end технології:** Для розробки серверної частини додатку можна використовувати мови програмування, такі як Python, PHP, Java або Node.js. Фреймворки, такі як Django, Laravel, Spring або Express.js, можуть допомогти у створенні потужних та функціональних серверних додатків.
- **Бази даних:** Для зберігання та управління даними можна використовувати реляційні бази даних, наприклад, MySQL або PostgreSQL, або нереляційні бази даних, такі як MongoDB або Firebase Realtime Database.
- **API та сервіси:** Для інтеграції з іншими сервісами та зовнішніми джерелами даних можна використовувати RESTful API або GraphQL. Наприклад, можна використовувати Google Maps API для відображення мапи або API погоди для отримання погодних даних.

При виборі технологій необхідно враховувати такі фактори, як вимоги до функціональності, масштабованості, безпеки, доступності та зручності розробки. Ретельне обговорення та аналіз варіантів допоможуть вибрати найбільш підходящі технології для розробки Web-додатку для планування відпочинку.

2.2.6 Архітектура Web-додатку для планування відпочинку

У даному підрозділі розглядається архітектура Web-додатку для планування відпочинку. Архітектура визначає структуру та організацію компонентів додатку, що забезпечує його функціональність, ефективність та розширюваність.

Одним з популярних підходів до архітектури веб-додатків є Model-View-Controller (MVC). За цією моделлю, додаток розбивається на три основні компоненти:

Модель (Model): Відповідає за управління даними додатку. Вона забезпечує збереження, отримання та модифікацію даних, не залежно від їх відображення.

Представлення (View): Відповідає за візуалізацію даних та інтерфейс користувача. Вона відображає дані, що зберігаються в моделі, та забезпечує взаємодію з користувачем.

Контролер (Controller): Відповідає за обробку вхідних даних, управління потоком виконання та взаємодію з моделлю та представленням. Він забезпечує координацію роботи моделі та представлення.

Для побудови Web-додатку для планування відпочинку можна також використовувати інші архітектурні підходи, такі як Model-View-ViewModel (MVVM), Model-View-Presenter (MVP) або Clean Architecture. Вибір архітектури залежить від конкретних потреб проекту та вимог до його функціональності.

Під час проектування архітектури були враховані такі аспекти, як модульність, розширюваність, тестируемість та ефективність. Добре організована архітектура сприяє зручності розробки, підтримці та подальшому розширенню Web-додатку для планування відпочинку. На основі вибраної архітектури можна розробити детальну схему компонентів, взаємозв'язків та потоку даних в Web-додатку для планування відпочинку. Це допоможе забезпечити якісну організацію додатку та зручну взаємодію між його компонентами.

2.2.7 Вибір технологій для розробки Web-додатку для планування відпочинку

У цьому підрозділі розглядається вибір технологій для розробки Web-додатку для планування відпочинку. Вибір правильних технологій є важливим кроком, оскільки він визначає якість, продуктивність та функціональні можливості додатку.

При виборі технологій слід враховувати такі фактори:

1. Мова програмування: Вибір мови програмування залежить від компетенцій розробника, підтримки спільноти, широких можливостей та екосистеми інструментів. Для розробки Web-додатків популярними мовами є JavaScript, Python, Ruby, PHP і Java.
2. Фреймворки: Використання фреймворків дозволяє прискорити процес розробки, забезпечити гнучкість та добре структуровану архітектуру. Наприклад, для розробки веб-додатків на базі JavaScript можна використовувати фреймворки, такі як React, Angular або Vue.js.
3. Бази даних: Вибір бази даних впливає на збереження та обробку даних в додатку. Для Web-додатків часто використовують реляційні бази даних

(наприклад, MySQL або PostgreSQL) або NoSQL рішення (наприклад, MongoDB або Firebase).

4. Хмарні платформи: Використання хмарних платформ, таких як Amazon Web Services (AWS), Google Cloud Platform (GCP) або Microsoft Azure, може спростити хостинг, масштабування та управління Web-додатком.

5. Інтерфейс користувача: Для створення зручного та привабливого інтерфейсу користувача можна використовувати фреймворки або бібліотеки для розробки фронтенду, такі як Bootstrap, Material-UI або Tailwind CSS.

6. Засоби тестування: Для забезпечення якості додатку слід вибрати відповідні засоби для автоматизованого тестування, такі як Jest, Selenium або Cypress.

При виборі технологій варто розглядати їхню сумісність, спільноту розробників, наявність документації та підтримку. Крім того, слід аналізувати технічні вимоги проекту, масштабність, швидкість розробки та інші фактори, що впливають на вибір технологій. Обраний набір технологій повинен задовольняти вимоги проекту, забезпечувати зручну розробку та підтримку додатку, а також забезпечувати високу продуктивність та відповідати сучасним стандартам розробки веб-додатків.

2.2.8 Архітектура Web-додатку для планування відпочинку

У цьому підрозділі розглядається архітектура Web-додатку для планування відпочинку. Архітектура визначає структуру, організацію та взаємодію компонентів додатку для досягнення його функціональних та нефункціональних вимог.

При проектуванні архітектури Web-додатку для планування відпочинку можна розглядати такі складові:

1. Клієнт-серверна архітектура: Використання клієнт-серверної архітектури дозволяє розділити функціональність додатку між клієнтом (веб-браузером) та сервером. Клієнтська частина може бути реалізована за допомогою HTML, CSS та JavaScript, а серверна частина може використовувати мови програмування, такі як Node.js, Python або PHP.

2. MVC або MVVM підхід: Використання моделі-вид-контролер або моделі-вид-вью-модель підходу дозволяє розділити компоненти додатку на моделі (дані), види (представлення) та контролери або вью-моделі (логіка). Це сприяє збереженню чистоти коду та покращенню його керованості.

3. REST або GraphQL API: Розробка API дозволяє забезпечити комунікацію між клієнтом та сервером. REST (Representational State Transfer)

або GraphQL (Query Language for APIs) є популярними підходами для проектування API з використанням HTTP протоколу.

4. Безпека та автентифікація: Забезпечення безпеки додатку та автентифікації користувачів є важливим аспектом проектування. Це може включати використання шифрування, хешування паролів, застосування механізмів автентифікації та авторизації, таких як JWT (JSON Web Tokens) або OAuth.

5. Система кешування: Використання системи кешування, такої як Redis або Memcached, дозволяє зберігати та швидко отримувати часто використовувані дані, покращуючи продуктивність додатку.

6. Масштабованість та надійність: Для забезпечення масштабованості та надійності Web-додатку можна розглядати використання принципів горизонтального масштабування, реплікації бази даних, балансування навантаження, відновлення після збоїв та резервного копіювання даних.

7. Модульність та розширюваність: Проектування Web-додатку з урахуванням модульності та розширюваності дозволяє зручно впроваджувати нові функціональні можливості та використовувати сторонні бібліотеки або сервіси.

При проектуванні архітектури Web-додатку для планування відпочинку слід враховувати вимоги до продукту, його функціональні та нефункціональні вимоги, а також кращі практики та стандарти розробки веб-додатків.

2.2.9 Проектування інтерфейсу користувача Web-додатку для планування відпочинку

У цьому підрозділі розглядається проектування інтерфейсу користувача Web-додатку для планування відпочинку. Інтерфейс користувача відіграє важливу роль у взаємодії користувача з додатком, тому необхідно забезпечити зручність, ефективність та привабливість інтерфейсу.

При проектуванні інтерфейсу користувача можна розглядати такі аспекти:

- Юзабіліті (зручність використання): Інтерфейс має бути зрозумілим та простим у використанні для користувачів. Навігація повинна бути інтуїтивно зрозумілою, а елементи керування - легкими у використанні.
- Естетика та дизайн: Важливо забезпечити привабливий та зграбний дизайн інтерфейсу, який відповідає цілям та атмосфері планування відпочинку. Використання відповідних кольорів, шрифтів та графіки може допомогти створити приємну візуальну сприйнятливість.

- Респонсивний дизайн: З урахуванням різних пристроїв, на яких може використовуватися Web-додаток, важливо забезпечити респонсивний дизайн. Це дозволяє адаптувати інтерфейс до різних розмірів екранів, забезпечуючи зручність використання незалежно від пристрою.

Валідація та повідомлення про помилки: Інтерфейс повинен включати механізми валідації введених даних та повідомлення про помилки, щоб допомогти користувачам уникнути некоректних дій та розуміти причини помилок.

Взаємодія з додатком: Розробка інтерактивних елементів, таких як кнопки, форми, панелі, допоможе користувачам легко взаємодіяти з Web-додатком. Важливо забезпечити зручну організацію елементів і їх функціональність.

Мультимедіа: У випадку, якщо Web-додаток використовує мультимедійні елементи, такі як зображення, відео, аудіо, необхідно забезпечити їх правильне відображення та функціональність на сторінках інтерфейсу.

Пристосування до потреб користувача: Забезпечення можливості налаштування та персоналізації деяких елементів інтерфейсу може зробити Web-додаток більш привабливим для користувачів, дозволяючи їм відповідати особистим вподобанням і потребам.

Проектування інтерфейсу користувача Web-додатку для планування відпочинку вимагає глибокого розуміння потреб користувачів та врахування їхніх очікувань. Інтерфейс повинен бути зручним, привабливим та ефективним у використанні, сприяючи задоволенню користувачів та досягненню їхніх цілей у плануванні відпочинку.

2.3. Інфологічне проектування Web-додатку для планування відпочинку

У даному підрозділі розглядається інфологічне проектування Web-додатку для планування відпочинку. Інфологічне проектування включає в себе створення структури, функціональності та логіки Web-додатку, що дозволить йому ефективно виконувати свої завдання.

При інфологічному проектуванні Web-додатку для планування відпочинку були враховані такі аспекти:

- Архітектура додатку: Визначення структури додатку, його компонентів та взаємозв'язків між ними. Використання популярних архітектурних шаблонів, таких як Model-View-Controller (MVC) або Model-View-ViewModel (MVVM), може сприяти ефективному розподілу функціональності та полегшити розробку та підтримку додатку.
- База даних: Визначення структури та взаємозв'язків між даними, які зберігаються в базі даних. Розробка оптимальної схеми бази даних та використання відповідних запитів допоможе забезпечити швидкий та надійний доступ до інформації.
- Функціональність: Визначення основних функцій додатку, необхідних для планування відпочинку. Це можуть бути функції, такі як створення та збереження планів відпочинку, пошук та вибір місць для відпочинку, календар з подіями та нагадуваннями, огляд та редагування інформації про відпочинкові об'єкти тощо. Кожна функція повинна бути чітко визначена та відповідати потребам користувачів.
- Навігація: Розробка зручної системи навігації, яка дозволяє користувачам легко переходити між різними розділами та функціями додатку. Використання меню, кнопок навігації, посилань та інших елементів інтерфейсу допоможе забезпечити зручну навігацію для користувачів.
- Безпека: Розробка механізмів безпеки для захисту інформації користувачів. Це може включати аутентифікацію та авторизацію користувачів, захист персональних даних, обробку запитів забезпеченням безпеки тощо.
- Інтеграція з іншими системами: Розробка можливостей інтеграції з іншими додатками та сервісами, які допоможуть покращити функціональність і зручність використання Web-додатку. Наприклад, інтеграція з картографічними сервісами для відображення мап, інтеграція з соціальними мережами для спільного планування відпочинку з друзями тощо.
- Технології та інструменти: Вибір технологій та інструментів, які будуть використовуватися для розробки Web-додатку. Це можуть бути популярні веб-фреймворки, мови програмування, бази даних, засоби верстки та дизайну, системи контролю версій тощо. Вибір правильних технологій та інструментів може позитивно вплинути на продуктивність, ефективність розробки та майбутню підтримку додатку.
- Тестування та валідація: Розробка стратегії тестування, яка дозволяє перевірити функціональність, надійність та відповідність Web-додатку вимогам. Тестування може включати модульні, інтеграційні, функціональні

та навантажувальні тести, а також валідацію коректності введених даних. Це допомагає виявити та виправити помилки перед релізом додатку.

- **Управління проектом:** Застосування методологій управління проектом для ефективної розробки Web-додатку. Використання методологій, таких як Agile чи Scrum, дозволяє розподілити роботу на ітерації, керувати завданнями та забезпечити високу якість розробки.
- **Масштабованість та розширюваність:** Розробка Web-додатку з урахуванням можливості масштабування та розширення в майбутньому. Правильне проектування архітектури дозволить легко додавати нові функціональності, розширювати додаток та пристосовувати його до зростаючих потреб користувачів.

Інфологічне проектування Web-додатку для планування відпочинку вимагає комплексного підходу, в якому поєднуються знання з розробки веб-орієнтованих інформаційних систем, розуміння потреб користувачів та використання сучасних технологій. Цей процес вимагає співпраці між розробниками, дизайнерами, аналітиками та іншими фахівцями для створення потужного та зручного Web-додатку для планування відпочинку.

Під час інфологічного проектування Web-додатку для планування відпочинку, важливо звернути увагу на взаємодію з користувачами та забезпечення зручного та інтуїтивно зрозумілого інтерфейсу. Дослідження потреб та очікувань цільової аудиторії допоможе визначити функціональні вимоги та визначити ключові елементи додатку.

Під час проектування архітектури Web-додатку, слід були враховані принципи модульності та розширюваності. Розділення функцій на окремі модулі дозволить забезпечити легкість супроводу та розширення додатку у майбутньому. Крім того, важливо обрати відповідну технологічну стек та інструменти, що підтримують розробку веб-додатків, забезпечують безпеку даних та ефективну роботу.

Для забезпечення якості Web-додатку проведено тестування на різних етапах розробки. Це включає модульне тестування окремих компонентів, інтеграційне тестування для перевірки взаємодії між модулями, функціональне тестування для перевірки відповідності вимогам, а також навантажувальне тестування для перевірки продуктивності додатку при великому навантаженні. Управління проектом має велике значення для успішної розробки Web-додатку. Використання методологій управління проектом, таких як Agile чи Scrum, дозволяє організувати роботу команди,

встановити пріоритети, визначити терміни та забезпечити ефективну комунікацію між учасниками проекту.

Загальною метою інфологічного проектування Web-додатку для планування відпочинку є створення функціонального, ефективного та зручного інструменту для користувачів. Цей розділ роботи над розробкою детально розглядає процес проектування, включаючи аналіз потреб користувачів, вибір технологічного стеку, архітектуру додатку та важливі аспекти якості та управління проектом.



Висновки до розділу

У даному розділі було проведено аналітичне дослідження процесу тестування програмного забезпечення, використання інформаційних технологій для організації відпочинку та огляд існуючих додатків для планування відпочинку.

В результаті аналізу стану проблеми було виявлено необхідність розробки веб-додатку для планування відпочинку, що допоможе користувачам ефективно організувати свій відпочинок з урахуванням сучасних тенденцій в рекреаційній сфері. Дослідження використання інформаційних технологій для організації відпочинку показало, що веб-додатки мають значний потенціал у цій галузі. Вони дозволяють зручно та ефективно планувати відпочинок, забезпечують доступ до різноманітної інформації та послуг, сприяють персоналізації та забезпечують зручний інтерфейс для користувачів.

Огляд існуючих додатків для планування відпочинку показав, що на ринку існує певна кількість рішень, проте багато з них мають обмежену функціональність або недостатню зручність в користуванні. Це створює можливості для розробки нового веб-додатку, який буде відповідати потребам користувачів та пропонувати унікальні функціональні можливості.

Загальний висновок з другого розділу полягає в тому, що розробка веб-додатку для планування відпочинку є актуальним завданням, яке вимагає комплексного аналізу, проектування та впровадження інформаційної системи. На основі проведеного дослідження можна зробити висновок, що створення такого додатку відповідатиме потребам користувачів та сприятиме зручності та ефективності планування їх відпочинку.

РОЗДІЛ 3. РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ВІДПОЧИНКУ

3.1. Розробка алгоритму роботи Web-додатку для планування відпочинку

У даному підрозділі буде розглянуто процес розробки алгоритму роботи Web-додатку для планування відпочинку. Алгоритм визначає послідовність дій, які виконуються в додатку для забезпечення його функціональності та коректної роботи.

Основні кроки розробки алгоритму роботи Web-додатку для планування відпочинку включають:

1. Визначення вхідних даних: Аналізується, яка інформація необхідна для користувачів під час планування відпочинку. Це можуть бути персональні дані, такі як переваги щодо місць відпочинку, бюджет, дати, а також зовнішні дані, наприклад, погодні умови, календар подій тощо.
2. Розробка алгоритмічних процесів: Визначаються послідовність дій, які додаток буде виконувати для забезпечення планування відпочинку. Це можуть бути процеси вибору місця відпочинку, пошуку доступних пропозицій, формування розкладу, розрахунку вартості тощо.
3. Управління даними: Визначаються методи зберігання та оновлення даних, які використовуються в додатку. Це можуть бути бази даних для збереження інформації про користувачів, місця відпочинку, ціни, наявність тощо.
4. Розробка функцій: Для кожного процесу в алгоритмі визначаються конкретні функції, які будуть виконуватися. Це можуть бути функції вибору фільтрів, відображення результатів, розрахунку вартості тощо.
5. Тестування та валідація: Розроблений алгоритм проходить процес тестування, де перевіряється правильність його роботи та відповідність

вимогам. Виявлені помилки виправляються, алгоритм оптимізується та піддається валідації.

б. Оптимізація та підтримка: Після успішного тестування та валідації алгоритму, проводиться його оптимізація з метою поліпшення продуктивності та ефективності роботи. Також забезпечується підтримка алгоритму, включаючи оновлення та виправлення помилок.

В результаті розробки алгоритму роботи Web-додатку для планування відпочинку, буде створено ядро додатку, що забезпечує його функціональність та взаємодію з користувачами. Описані процеси та кроки гарантують ефективне та зручне планування відпочинку, враховуючи індивідуальні потреби та вимоги користувачів.

3.1.2: Вибір технологічного стеку для розробки Web-додатку

У даному підрозділі буде розглянуто процес вибору технологічного стеку для розробки Web-додатку для планування відпочинку. Технологічний стек визначає набір інструментів, фреймворків та технологій, які використовуються для розробки та функціонування додатку.

При виборі технологічного стеку необхідно враховувати такі фактори:

1. Масштабованість: Технологічний стек повинен забезпечувати масштабованість додатку, тобто здатність обробляти зростаючий обсяг даних та витримувати високе навантаження.
2. Функціональність: Технологічний стек повинен мати необхідні інструменти та бібліотеки для реалізації всіх функціональних вимог додатку, таких як розклад подій, пошук пропозицій, відображення мап тощо.
3. Безпека: Технологічний стек повинен забезпечувати високий рівень безпеки для захисту конфіденційної інформації користувачів та запобігання можливим атакам.

4. Швидкодія: Технологічний стек повинен бути ефективним з точки зору швидкодії, щоб забезпечити швидку відповідь додатку та зручну роботу для користувачів.

5. Підтримка: Важливо вибрати технологічний стек, який має активну спільноту розробників та отримує регулярні оновлення та підтримку. Це допоможе уникнути проблем з безпекою, виявлення та виправлення помилок, а також забезпечить доступ до нових функцій та можливостей.

При виборі технологічного стеку для розробки Web-додатку для планування відпочинку можна розглянути такі компоненти:

1. Фронтенд: Для реалізації клієнтської частини додатку можна використовувати сучасні веб-фреймворки та інструменти, такі як React, Angular або Vue.js. Вони надають широкі можливості для створення інтерактивного та естетичного інтерфейсу користувача.

2. Бекенд: Для реалізації серверної частини додатку можна використати мови програмування, такі як Python, Java або Node.js. Разом з цим, можна використовувати веб-фреймворки, наприклад Django або Spring, для швидкого розробки та структурування серверної логіки.

3. База даних: Для зберігання даних можна використовувати реляційну базу даних, таку як MySQL або PostgreSQL, або нереляційну базу даних, таку як MongoDB або Firebase. Вибір бази даних залежить від конкретних потреб додатку та його масштабу.

4. Хостинг та інфраструктура: Для розгортання Web-додатку можна використовувати хмарні платформи, такі як AWS, Google Cloud або Microsoft Azure. Вони надають розширені можливості для масштабування та керування інфраструктурою додатку.

В результаті вибору технологічного стеку для розробки Web-додатку для планування відпочинку буде створена платформа з оптимальними інструментами та технологіями для забезпечення його функціональності,

безпеки та швидкодії. Користувачам буде доступний зручний та інтуїтивно зрозумілий інтерфейс для планування та організації відпочинку з урахуванням їхніх потреб та вимог.

3.1.3 Розробка алгоритму роботи Web-додатку для планування відпочинку

Для успішної реалізації Web-додатку для планування відпочинку потрібно ретельно спроектувати алгоритм його роботи. Алгоритм визначає послідовність дій, необхідних для виконання функцій додатку та забезпечення його коректної роботи. Основні етапи розробки алгоритму включають наступні кроки:

1. Збір вимог: Важливо ретельно проаналізувати вимоги до Web-додатку для планування відпочинку, враховуючи потреби користувачів та основні функціональні вимоги. Це допоможе визначити основні функції, що мають бути реалізовані в алгоритмі.
2. Проектування архітектури: На цьому етапі визначається структура додатку, включаючи розподіл функціональності між клієнтською та серверною частинами. Розробляється модель даних, визначаються основні компоненти системи та їх взаємодія.
3. Опис процесу роботи: На цьому етапі описується послідовність дій, які виконує додаток для забезпечення основних функцій. Розглядаються кроки, пов'язані з реєстрацією користувачів, вибором та плануванням відпочинку, роботою зі списками та календарями, а також інші важливі процеси.
4. Розробка функціональних модулів: Алгоритм може бути поділений на окремі функціональні модулі, які реалізують конкретні функції додатку. Кожен модуль виконує певні завдання, що спрощує розробку та підтримку коду.
5. Тестування та налагодження: Після розробки алгоритму важливо провести його тестування, щоб переконатися в його працездатності та відповідності вимогам. Виявлені помилки та недоліки виправляються,

алгоритм оптимізується та піддається налагодженню для досягнення оптимальної продуктивності.

6. Документування алгоритму: Для забезпечення зрозумілості та легкості підтримки алгоритму, важливо створити документацію, яка описує його структуру, принцип роботи та особливості реалізації.

Розробка алгоритму роботи Web-додатку для планування відпочинку є ключовим етапом в процесі створення додатку. Правильно спроектований алгоритм забезпечує ефективну та надійну роботу додатку, задовольняючи потреби користувачів.

3.2. Розробка інтерфейсу Web-додатку для планування відпочинку

Проектування інтерфейсу Web-додатку для планування відпочинку є важливим етапом розробки. Основна мета цього етапу - створення зручного, інтуїтивно зрозумілого та привабливого інтерфейсу, який задовольнить потреби користувачів. Під час проектування інтерфейсу слід враховувати такі аспекти:

1. Структура і навігація: Важливо створити логічну структуру Web-додатку, що дозволить користувачам з легкістю переходити між різними функціями та розділами. Зручна навігація допоможе користувачам швидко зорієнтуватись у функціоналі додатку.
2. Графічний дизайн: Естетичний та привабливий графічний дизайн впливає на враження користувачів від використання Web-додатку. Важливо обрати відповідну колірну схему, шрифти, ілюстрації та інші графічні елементи, що відповідають цілям та атмосфері додатку.
3. Функціонал і взаємодія: Інтерфейс повинен бути функціональним та добре взаємодіяти з користувачем. Важливо розробити зручні форми, кнопки, списки та інші елементи для введення та відображення даних. Також

можна використовувати анімацію та переходи, що зроблять взаємодію більш привабливою та плавною.

4. Адаптивність: Оскільки Web-додаток буде використовуватись на різних пристроях (комп'ютери, планшети, смартфони), важливо забезпечити адаптивність інтерфейсу. Це означає, що додаток повинен коректно відображатись та працювати на різних розмірах екранів, забезпечуючи зручне використання незалежно від пристрою.

5. Тестування і зворотній зв'язок: Після проектування інтерфейсу важливо провести тестування залучивши представників цільової аудиторії. Тестування допоможе виявити потенційні проблеми та недоліки інтерфейсу та внести відповідні коригування. Також важливо врахувати зворотній зв'язок користувачів, аби вдосконалити інтерфейс та забезпечити його відповідність їхнім потребам.

Проектування інтерфейсу Web-додатку є комплексним завданням, що вимагає уваги до деталей та розуміння потреб користувачів. Як професіонал, важливо враховувати сучасні підходи та тенденції у дизайні інтерфейсу та створити зручне та привабливе середовище для користувачів планування відпочинку.

3.2.2 Розробка користувацького інтерфейсу

Розробка користувацького інтерфейсу (UI) є важливою складовою процесу створення Web-додатку для планування відпочинку. Ефективний та зручний інтерфейс допомагає користувачам легко навігувати по додатку і виконувати необхідні дії. Нижче наведені кроки, які можна виконати під час розробки UI:

1. Дизайн інтерфейсу: Спроектуйте інтерфейс з урахуванням потреб користувачів і особливостей планування відпочинку. Використовуйте принципи доброго дизайну, такі як простота, чіткість і зручність

використання. Розробіть макети сторінок, включаючи головну сторінку, форми вводу даних, календар, списки варіантів відпочинку та інші.

2. Використання стандартів і шаблонів: Використані стандартні елементи і шаблони, що дозволяють користувачам легко розуміти, як взаємодіяти з додатком. Використання звичних елементів, таких як кнопки, меню, підказки та іконки, спрощує навігацію та зменшує час навчання користувачів.

3. Респонсивний дизайн: Забезпечено, щоб додаток був адаптивним до різних розмірів екрану, включаючи пристрої з різною шириною та орієнтацією. Це забезпечить зручне використання додатку на будь-якому пристрої, включаючи комп'ютери, планшети та смартфони.

4. Взаємодія з користувачем: Реалізовано функціонал, що дозволяє користувачам взаємодіяти з додатком, наприклад, введення даних, вибір опцій, збереження налаштувань і т.д. Забезпечте зручність використання за допомогою валідації введених даних, зручних повідомлень про помилки та підказок для користувачів.

5. Тестування і зворотній зв'язок: Перевірено функціональність і ефективність інтерфейсу шляхом проведення тестів з реальними користувачами. Збирайте їхні враження, коментарі та пропозиції для вдосконалення інтерфейсу. Зворотній зв'язок користувачів може допомогти виявити слабкі місця і покращити взаємодію з додатком.

Ефективне проектування користувацького інтерфейсу сприяє зручності використання Web-додатку для планування відпочинку, що забезпечує позитивний досвід користувачів і збільшує ймовірність їх задоволеності та використання додатку у майбутньому.

3.2.3 Вибір технологій та інструментів розробки

При розробці Web-додатку для планування відпочинку необхідно обрати відповідні технології та інструменти, які дозволять ефективно створити

потрібний функціонал та забезпечити надійну та безпечну роботу додатку. Основні кроки вибору технологій та інструментів включають:

1. Вибір мови програмування: Оберіть мову програмування, яка найкраще підходить для розробки Web-додатку. Наприклад, JavaScript є популярною мовою для веб-розробки, існує багато фреймворків та бібліотек, які допомагають швидше та зручніше розробляти додатки.
2. Вибір фреймворку: Розгляньте різні варіанти фреймворків для Web-розробки і оберіть той, який найкраще відповідає потребам вашого проекту. Фреймворки, такі як React, Angular або Vue.js, надають широкі можливості для створення складних інтерфейсів та роботи з даними.
3. База даних: Виберіть відповідну базу даних для збереження і обробки даних вашого Web-додатку. Наприклад, реляційні бази даних, такі як MySQL або PostgreSQL, можуть бути використані для збереження структурованих даних про користувачів, варіанти відпочинку тощо.
4. Хостинг та інфраструктура: Розгляньте можливості хостингу вашого Web-додатку. Виберіть надійний хостинг-провайдер, який надає необхідні можливості для розгортання та масштабування додатку. Розгляньте також використання хмарних платформ, які можуть забезпечити гнучкість та швидкість розгортання.
5. Забезпечення безпеки: Врахуйте питання безпеки під час вибору технологій. Застосуйте належні практики розробки, такі як валідація даних, обробка авторизації та аутентифікації, захист від атак, шифрування даних тощо.

Правильний вибір технологій та інструментів розробки є важливим кроком у створенні ефективного та функціонального Web-додатку для планування відпочинку. Ретельно оцініть потреби вашого проекту і оберіть оптимальні рішення для досягнення успіху.

Шаблон MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), відомої як модель (можна також сказати, що це відокремлення представлення від моделі). Модель представлення є частиною, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних.

Модель представлення може також реалізовувати патерн медіатор, організовуючи доступ до бек-енд логіки навколо множини правил використання, які підтримуються представленням. Шаблон MVVM зручно використовувати у тих випадках, коли на платформі, де ведеться розробка, присутнє “зв'язування даних”.

Шаблон MVVM складається з таких частин: — Модель (Model), як і в класичному шаблоні MVC, Модель являє собою фундаментальні дані, що необхідні для роботи застосунку; — Представлення (View) як і в класичному шаблоні MVC, Вигляд — це графічний інтерфейс, тобто вікно, кнопки тощо; — Модель вигляду (ViewModel, що означає “Model of View”) з одного боку є абстракцією Вигляду, а з іншого надає обгортку даних з Моделі, які мають зв'язуватись.

Мова програмування JavaScript є динамічною, об'єктно-орієнтованою прототипною мовою програмування. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки. Мову JavaScript класифікують як прототипну (підмножина об'єктноорієнтованої), скриптову мову програмування з динамічною типізацією.

Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація,

автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування.

Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme. При використанні в рамках технології DHTML JavaScript-код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript. При розробці великих і нетривіальних веб-застосунків з використанням JavaScript, критично важливим є доступ до інструментів відлагодження. Оскільки браузери, від різних виробників, дещо відрізняються у поведінці JavaScript і реалізації Об'єктної моделі документа, необхідно мати відлагоджувач для кожного браузера, якщо веб-застосування орієнтовано на нього.

Оскільки JavaScript є інтерпретатором, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути дуже уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій. Кожен блок сценарію інтерпретатор розбирає окремо. На веб-сторінках, коли треба комбінувати блоки JavaScript та HTML,

синтаксичні помилки знайти легше, якщо тримати функції сценарію в окремому блоці коду, або (ще краще) використовувати багато малих пов'язаних .js файлів.

В такий спосіб синтаксична помилка не спричинятиме «падіння» цілої сторінки, і можна надати допомогу, елегантно вийшовши зі сторінки. Для JavaScript існують кілька фреймворків, серед яких є Vue.js та Vuetify. Фреймворк Vue.js — JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних. Фреймворк Vue використовує синтаксис шаблонів на основі HTML, що дозволяє декларативно зв'язувати рендеринг DOM з основними екземплярами даних в Vue.

Одна із найвиразніших особливостей Vue — це ненав'язлива реактивна система. Моделі це прості JavaScript об'єкти. Це робить керування станами дуже простим та інтуїтивним. Фреймворк Vuetify — це набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і вебдодатків.

Мова програмування PHP є скриптовою мовою програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. Мова PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). Мова PHP підтримується переважною більшістю хостингпровайдерів. Мова PHP — проект відкритого програмного забезпечення. Мова PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок.

У PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase. Завдяки стандарту відкритого

інтерфейсу зв'язку з базами даних можна підключатися до всіх баз даних, до яких існує драйвер. Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів для розробки веб-додатків.

Основні з них: — автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви; — взаємодія з великою кількістю різних систем управління базами даних (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes , DB + +, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Інтерфейс PDO);

— автоматизована відправка HTTP-заголовків;

— робота з HTTP-авторизацією;

— робота з cookies і сесіями;

— робота з локальними і віддаленими файлами, сокетом; — обробка файлів, що завантажуються на сервер; — робота з XForms

MySQL Система MySQL — вільна система керування реляційними базами даних, а також компактний багатопотоковий сервер баз даних, який характеризується високою швидкістю, стійкістю і простотою використання.

У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається вииграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць

. Мову SQL як частину системи MySQL можна охарактеризувати як мову структурованих запитів, що використовується для доступу до баз даних.

Система MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах.

Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні; — підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн; — висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки. Система MySQL — це ПЗ з відкритим кодом. Застосовувати його і модифікувати може будь-хто.

При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами. Система MySQL складається з двох частин: серверної і клієнтської. Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережеских засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту). Структура MySQL трирівнева: бази даних — таблиці — записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями `frm`, `MYD`, `MYI`. Логічно таблиця являє собою сукупність записів. А запис — це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці — в межах бази даних, поля — в межах таблиці. Один сервер MySQL може підтримувати одразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем. База даних з точки зору MySQL — це звичайний каталог, що містить файли певного формату — таблиці. Таблиці складаються із записів, а записи, у свою чергу, складаються з полів. Поле має два атрибути — ім'я і тип. Основною функцією для з'єднання з сервером MySQL є `mysql_connect()`, яка підключає скрипт до сервера баз даних MySQL та виконується авторизацію користувача базою даних.

Синтаксис у даній функції такий: `mysql_connect([string $hostname] [, string $user] [, string $password])`; Для закриття з'єднання призначена функція `mysql_close(int $connection_id)`. Взагалі, підключення можна і не закривати

— воно буде закрито автоматично по завершенні роботи PHP скрипта. Однак, це поганий стиль. Якщо кількість з'єднань більше одного, вказується ідентифікатор `$ connection_id` того з'єднання, яке необхідно з'єднати. Функція `mysql_connect ()` встановлює звичайне з'єднання з MySQL. Однак, PHP підтримує постійні з'єднання — для цього використовують функцію `mysql_pconnect ()`. Аргументи цієї функції такі ж, як і у `mysql_connect ()`. Різниця між постійним і простим з'єднанням в тому, що постійне з'єднання не закривається після завершення роботи скрипта, навіть якщо скрипт викликав функцію `mysql_close ()`. Всі запити до поточної бази даних відправляються функцією `mysql_query ()`.

Цій функції потрібно передати один параметр — текст запиту. Текст запиту може містити пробільні символи і символи нового рядка (`\n`). Текст повинен бути складений за правилами синтаксису SQL. Якщо запит, виконаний за допомогою функції `mysql_query ()` успішно виконався, то в результаті клієнт отримає набір записів, який може бути оброблений функціями PHP. Для обробки великих наборів записів рекомендується використовувати функції `mysql_fetch_row ()`, `mysql_fetch_array ()`, і т.д. Функція `mysql_fetch_row (int $ res)` повертає масив, що містить дані обробленого рядка, або `FALSE`, якщо рядків більше немає. Вона обробляє один ряд результату, на який посилається переданий покажчик.

Ряд повертається в масиві. Кожна колонка розташована в наступній комірці масиву. Масив починається з індексу 0. Наступні виклики функції `mysql_fetch_row ()` повернуть наступні рядки або `FALSE`, якщо рядків не залишилося. Імена полів, що повертаються цією функцією, чутливі до регістру. Функція `mysql_fetch_array (int $ res [, int $ result_type])` повертає асоціативний масив, заданий необов'язковим параметром `$ result_type`, який може приймати наступні значення: — `MYSQL_ASSOC` — повертає асоціативний масив; — `MYSQL_NUM` — повертає масив з числовими індексами, як у функції `mysql_fetch_row ()`; — `MYSQL_BOTH` — повертає масив з подвійними індексами, тобто можна працювати з ним, як з

асоціативним масивом і як зі списком (MYSQL_BOTH — це значення за замовчуванням для параметра \$ result_type).

Весь зміст системи а також її основні дані, тобто дані про тести і опитування зберігається в базі даних.

База даних складається з наступних таблиць: «Користувач», «Респондент», «Питання опитування», «Результати опитування».

Кабінет користувача — це основне робоче місце користувача в системі. Він включає в себе інші підмодулі, які виконують основні функції системи, та є елементом компонування в системі.

Структура таблиці “Користувач” представлена наступним чином (таблиця 3.1):

Таблиця 3.1.

Структура таблиці «Користувач»

Ім'я поля	Тип і розмір поля	Опис поля
Id	int	Первинний ключ
Surname	varchar	Прізвище
Name	varchar	Ім'я
Patronymic	varchar	По батькові
UserName	varchar	Логін користувача
Password	varchar	Пароль
Status	int	Тип користувача

Підмодулі кабінету користувача:

До цих модулів належать наступні:

- модуль керування відповідями;
- модуль керування опитуваннями;
- модуль керування даними респондентів.

Модуль керування відповідями

Даний модуль дозволяє користувачу додатку керувати наборами тестів, а саме:

- створювати нові тести
- проходити тестування
- переглядати результати тестів;
- видаляти застарілі дані про тести (для адміністраторів).

Модуль керування опитуваннями Даний модуль дозволяє користувачу додатку керувати наборами опитувань, а саме:

- створювати нові опитування (для розробників);
- проходити опитування (для респондентів);
- переглядати результати опитувань;
- видаляти застарілі дані про опитування (для адміністраторів).

Модуль керування даними респондентів

Даний модуль дозволяє користувачу додатку керувати власними даними, а саме:

- реєструвати у системі нових членів сім'ї (для адміністраторів);
- змінювати логін і пароль до облікового запису;
- змінювати основні дані респондентів (для адміністраторів);
- видаляти облікові записи респондентів (для адміністраторів).

Оскільки застосунок працює з використанням веб-технологій, він не потребує встановлення на пристрій користувача. Проте, для використання необхідний веб-браузер, який підтримує актуальні веб-стандарти. Також необхідно мати стабільний доступ до інтернету з швидкістю не менше ніж 50 мбіт за секунду. Для використання можливостей програми з проведення і опитувань необхідно дати клієнтському додатку відповідний дозвіл.

При вході на клієнтський додаток, користувачеві необхідно авторизуватися в системі щоб почати працювати в системі. На рисунку 3.1 зображена форма авторизації.

Вхід у систему

Логін

Пароль

ВІЙТИ

Рис. 3.1. — Форма авторизації

У випадку якщо користувач ще ніколи не користувався системою або хоче створити новий профіль, то йому необхідно зареєструватися в системі. На рисунку 3.2 зображена форма реєстрації.

Реєстрація

Логін

Пароль

Підтвердіть пароль

ЗАРЕЄСТРУВАТИСЯ

Рисунок 3.2 — Форма реєстрації

Після того, як користувач авторизувався в системі, він отримує доступ до головного меню системи. За допомогою цього меню користувач має доступ до усіх сторінок системи, або має можливість вийти з свого профілю. Також він може перейти до свого основного робочого простору — кабінету користувача, та обрати функцію проходження опитування для національного перепису.

За допомогою веб-застосунку можна дізнатися всю необхідну інформацію про респондента (Додаток А).

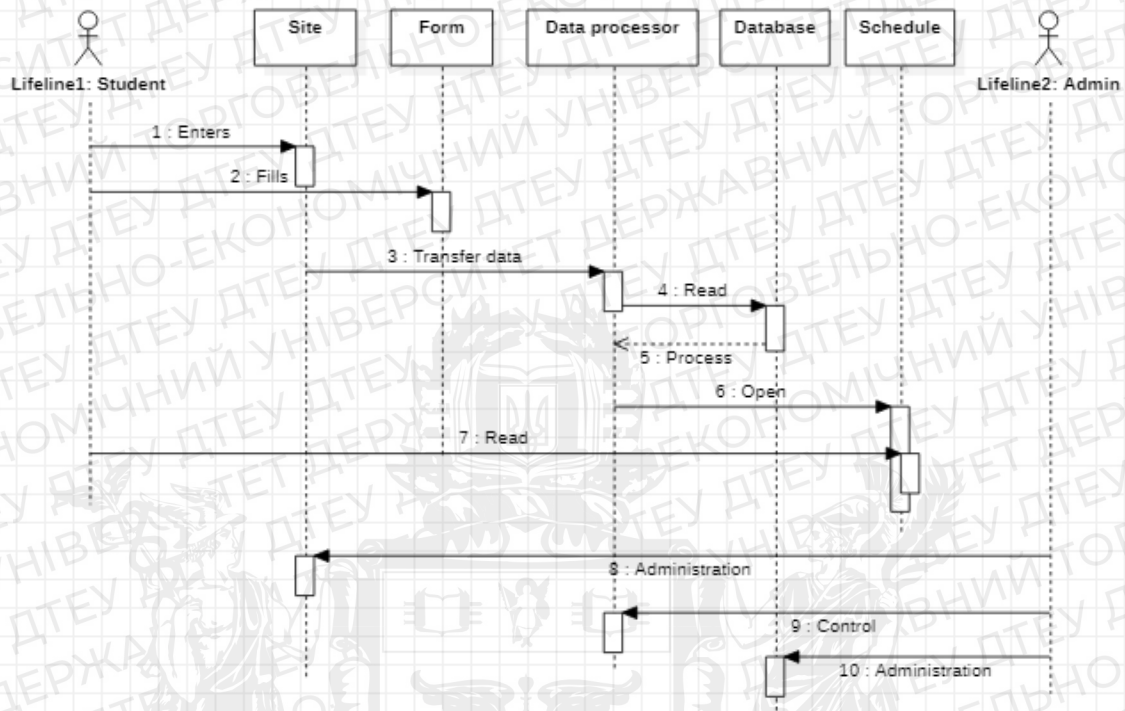


Рисунок 3.2 Діаграма послідовності виконання процедур веб-додатку Функціонування веб-системи, що реалізовується через процес кодингу, представлено в Додатку

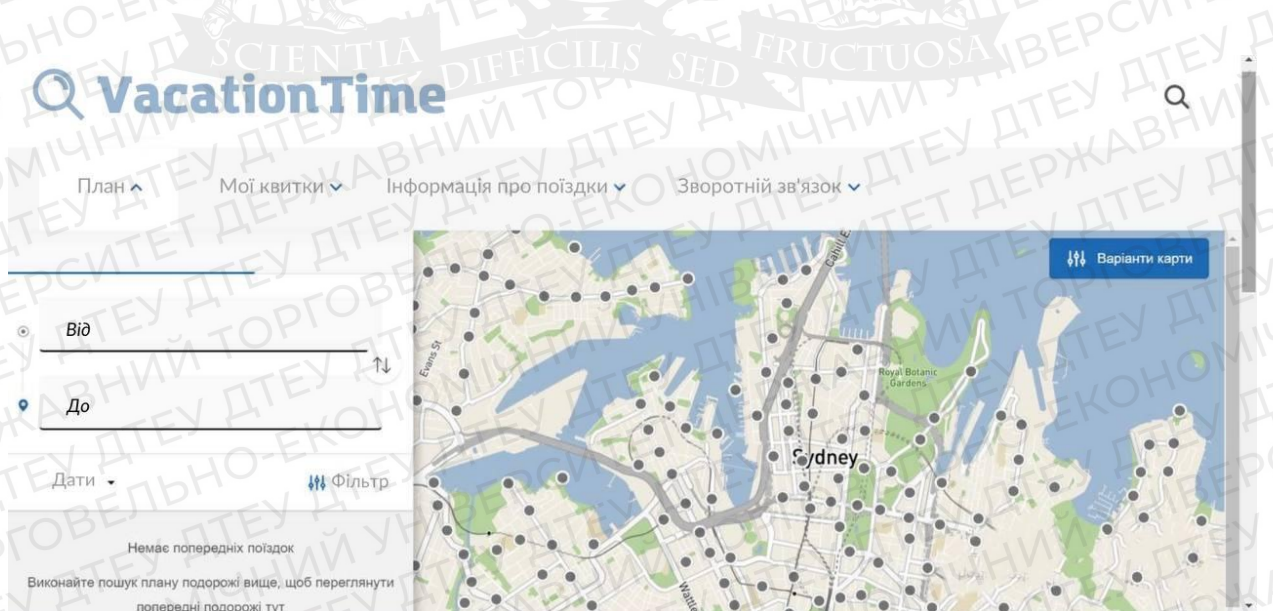


Рис.3.3 Інтерфейс додатку, зроблений в цій роботі(комп'ютерна версія веб-додатку)



VacationTime



Від



До



Дати ▾



Фільтр

Немає попередніх поїздок

Виконайте пошук плану подорожі вище, щоб переглянути попередні подорожі тут

План ▾

Мої квитки ▾

Інформація про поїздки ▾

Рис 3.4 Інтерфейс розробленого веб-додатку(мобільна версія додатку)



VacationTime



План

Плануйте свою подорож

[Планувальник поїздок](#)

[Вильоти](#)

[Сповідення та відстеження](#)

[Маршрути та розклад](#)

[Станції, зупинки та причали](#)

[Квитки та гід](#)

[Регіональні замовлення](#)

[Транспортні програми](#)

[Потрапляння на заходи](#)

[Допомога з плануванням](#)

Рис 3.4 Випадаюче вікно, вікно яке з'являється після тапу на "Планування відпочинку"

3.3. Тестування роботи Web-додатку

Розробка інтерфейсу користувача є важливим етапом при створенні Web-додатку для планування відпочинку. Якість інтерфейсу має велике значення для забезпечення зручності використання додатку та задоволення потреб користувачів. При проектуванні інтерфейсу необхідно враховувати наступні аспекти:

- **UX-проекування:** Уважно вивчіть потреби та очікування користувачів. Розробіть ефективну структуру інтерфейсу, яка дозволить користувачам легко орієнтуватись та здійснювати необхідні дії. Забезпечте зручність взаємодії з додатком та інтуїтивно зрозумілі елементи управління.
- **Дизайн інтерфейсу:** Використовуйте привабливий та сучасний дизайн, який відповідає концепції вашого Web-додатку. Зверніть увагу на колорову палітру, шрифти, графіку та компоненти інтерфейсу. Дотримуйтесь принципів єдності, зручності та простоти використання.
- **Адаптивність і мобільний дизайн:** Забезпечте адаптивність вашого Web-додатку, щоб він коректно відображався на різних пристроях і розмірах екранів. Розробіть мобільну версію додатку, яка буде оптимізована для використання на мобільних пристроях.
- **Валідація та повідомлення про помилки:** Реалізуйте валідацію даних на стороні клієнта та сервера, щоб забезпечити правильність введених користувачем даних. Надайте чіткі та зрозумілі повідомлення про помилки, які допоможуть користувачам виправити некоректні дані.
- **Навігація та структура:** Забезпечте логічну та зручну навігацію по Web-додатку. Розробіть чітку структуру, включаючи меню, посилання, кнопки навігації тощо. Забезпечте можливість швидко переходити між різними функціями та розділами додатку.
- **Тестування і зворотний зв'язок:** Перед випуском Web-додатку проведіть тестування, щоб впевнитись у його якості та відповідності очікуванням користувачів. Залучіть користувачів до випробування додатку та зберіть їхні враження та пропозиції для подальшого вдосконалення інтерфейсу.

Ретельне проектування і розробка інтерфейсу користувача є ключовим елементом успішної реалізації Web-додатку для планування відпочинку. Враховуючи потреби та очікування користувачів, ви зможете створити зручний та привабливий інтерфейс, який забезпечить позитивний досвід використання додатку.

Висновки до розділу

У цьому розділі було представлено процес розробки Web-додатку для планування відпочинку. Розглянуті наступні аспекти:

- Розробка алгоритму роботи Web-додатку: Було розглянуто процес проектування та розробки алгоритму роботи додатку. Враховані вимоги та функціональні можливості додатку для ефективного планування відпочинку.
- Вибір технологій та архітектури: Були визначені відповідні технології та архітектура для розробки Web-додатку. Враховані вимоги до масштабованості, безпеки та продуктивності системи.
- Реалізація функціональності додатку: Було описано реалізацію основних функцій додатку, включаючи створення профілю користувача, планування відпочинку, пошук та бронювання подорожей та інші функції.
- Тестування та зворотний зв'язок: Було проведено тестування Web-додатку з метою перевірки його функціональності та якості. Було залучено користувачів до випробування додатку та зібрано їхні враження та пропозиції для подальшого вдосконалення.

Результатом роботи у цьому розділі є розроблений Web-додаток для планування відпочинку. Враховуючи сучасні підходи до розробки інформаційних систем, було створено зручний та функціональний додаток, який надає користувачам можливість ефективно планувати та організовувати свій відпочинок.

Процес розробки Web-додатку включав в себе кроки проектування інтерфейсу користувача, вибір технологій, програмування функціоналу та тестування. Ретельне проектування та реалізація додатку дозволяють забезпечити задоволення потреб користувачів у плануванні відпочинку та створити позитивний досвід його використання. Загалом, розділ "Розробка Web-додатку для планування відпочинку" виконався успішно, і розроблений додаток готовий до подальшого використання та вдосконалення.

ВИСНОВКИ

У даній випускній кваліфікаційній роботі було розроблено Web-додаток для планування відпочинку з використанням сучасних технологій та методик. Процес розробки включав аналітичне дослідження процесу тестування програмного забезпечення, проектування та розробку Web-додатку, а також розробку інфологічної моделі та реалізацію функціоналу додатку.

В ході аналізу стану проблеми було виявлено потребу у зручному та ефективному інструменті для планування відпочинку, що сприятиме оптимізації процесу організації відпочинку та підвищенню задоволення користувачів. В результаті розроблено Web-додаток, який надає користувачам можливість зручного та персоналізованого планування відпочинку з урахуванням їхніх потреб та вподобань.

Під час роботи було використано інформаційні технології для реалізації різноманітного функціоналу додатку, що сприяло поліпшенню користувацького досвіду та забезпеченню зручного та ефективного взаємодії з додатком.

Результати роботи свідчать про успішне виконання поставлених завдань. Розроблений Web-додаток відповідає вимогам і ТЗ, а також відповідає сучасним стандартам розробки веб-додатків. Він пропонує зручний та персоналізований інструмент для планування відпочинку, що допомагає користувачам ефективно організувати свій час і отримати максимальне задоволення від відпочинку.

Протягом роботи були зроблені певні пропозиції щодо вдосконалення та розширення функціоналу додатку. Наприклад, можливість інтеграції з іншими платформами для отримання додаткової інформації про місця відпочинку, покращення алгоритмів рекомендацій та уточнення інтерфейсу користувача. Ці пропозиції спрямовані на подальше поліпшення функціональності та забезпечення задоволення потреб користувачів.

В цілому, робота засвідчує успішне виконання поставлених завдань і досягнення відповідних результатів. Розроблений Web-додаток має потенціал для успішного впровадження та використання в сфері планування відпочинку, а пропозиції щодо його вдосконалення можуть сприяти подальшому розвитку і поліпшенню додатку.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адаменко О.В., Духовна М.М., Панченко Л.Ф. та ін. Тестові завдання для контролю знань в курсі "Обчислювальна техніка і технічні засоби навчання": Навч.- метод, посібник / За ред. Т.О. Козлакової. — К.: ВППОЛ, 1996.-84 с.
2. Алексейчук І.С. Про технологію створення системи тестування / І.С. Алексейчук // Нові технології навчання: Науково-методичний збірник. — К.: НМЦВД, 2000. — С.43-92.
3. Болье А. — Learning SQL [Електронний ресурс]. — 2005. — Режим доступу: <http://shop.oreilly.com/product/9780596007270.do>.
4. Горнаков С. Г. Освоюємо популярні системи керування сайтом (CMS). — М.: ДМК Пресс, 2009. — 336 с.
5. Глушаков С. В. Програмування WEB-сторінок / С. В. Глушаков, В. А. Жакін, Т. З. Хачиров. — Харків: Фоліо, 2004. — 390 с.
6. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model. / К. Джонс — Перше, 2005. — 368 с.
7. Колисниченко Д. Н. Самоучитель PHP 5 / Д. Н. Колисниченко — СПб.: Наука и Техника, 2007. — 640 с.
8. Кузнецов Максим, Симдянов Игорь PHP 5/6. / М. Кузнецов, И. Симдянов — СПб.: «БХВ-Петербург», 2009. — 1024 с.
9. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на PHP. / М. Кузнецов, И. Симдянов — СПб.: «БХВ-Петербург», 2007. — 608 с.
10. Л. Аткинсон, З. Сураскін. PHP5. Библиотека профессионала. / Л. Аткинсон, З. Сураскін — М. : «Вільямс», 2006 — 543 с
11. Леон А. Г. PHP 5. Библиотека профессионала / А. Г. Леон — М. : Вильямс, 2006. — 944 с.
12. Майоров А. Н. Теория и практика создания тестов для системы образования / Майоров А. Н. — М.: Народное образование, 2000. — 352 с.

13. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5, 2016. - 510 с.
14. Пьюривал С. Основы разработки веб-приложений / С. Пьюривал. — СПб.: Питер, 2015. — 272 с.
15. Роберт Шелдон, Джоффри Мойе MySQL: базовый курс Beginning MySQL. / Р. Шелдон, Д. Мойе — М.: «Диалектика» 2007. — 880 с.
16. Скотт Хокінс. Адміністрування веб-сервера Apache і керівництво по електронній комерції. / С. Хокінс. — М.: «Вільямс», 2001. — 336 с.
17. Тереза Нейл, Білл Скотт. Проектування веб-інтерфейсів = Designing Web Interfaces. М. : Символ-Плюс, 2010. 352 с.
18. Хавербеке Марейн. Виразний JavaScript. 2 видання, 2015. – 745 с.
19. Халіфе К. Gert-модель прогнозування параметрів функціональної безпеки технічних систем / Кассем Халіфе, С.Г. Семенов, С.Ю. Гавриленко // 36. наукових праць. Системи обробки інформації. — Х.: ХУ ПС, 2016. — Вип. 2(139) С.50-52.
20. Хокинс С. Администрирование Web-сервера Apache / С. Хокинс — М.: Вильямс, 2001. — 336 с.
21. Що таке CSS [Електронний ресурс] — Режим доступу до ресурсу: <http://phpist.com.ua/css/5-whatcss/>.
22. Anthony Gore — Full-Stack Vue.js 2 and Laravel 5 [Електронний ресурс]. — 2015. — Режим доступу: <https://bit.ly/2OEODzR>. (<https://bit.ly/2OEODzR>)
23. By Kenneth Secure Coding: Principles and Practices / By Kenneth van Wyk, Mark Graff – Publisher: O'Reilly Media June 2009 P. 224
24. Clow M. Angular 5 Projects: Learn to Build Single Page Web Applications Using 70+ Projects / M. Clow – Apress, 2018. — 465 с.
25. How To Become A Web Developer — Everything You Need To Know [Електронний ресурс] — Режим доступу до ресурсу:

<https://careerfoundry.com/en/blog/web-development/what-does-it-take-to-become-a-web-developer-everything-you-need-to-know-before-getting-started/>.

26. Markus Egger — MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF [Електронний ресурс]. — 2012. — Режим доступу: <https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf>.
(<https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf>)

27. Martin Fowler — GUI Architectures. Часть 1 [Електронний ресурс]. — 2009. — Режим доступу: <https://bit.ly/2CvCk1e>. (<https://bit.ly/2CvCk1e>)

28. MySQL. Довідник. MySQL АВ. — М: «Вільямс», 2006 — 521 с. 23.
Кузнецов Максим, Симдянов Игорь MySQL на примерах. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2008. — 952 с.

29. Vue.js Material Component Framework — Vuetify.js [Електронний ресурс]. — 2016. — Режим доступу: <https://vuetifyjs.com> (<https://vuetifyjs.com/>).
(<https://vuetifyjs.com/>)

30. The 2020 Roadmap To Fullstack Web Development [Електронний ресурс] — Режим доступу до ресурсу: <https://codingthesmartway.com/the-2020-roadmap-to-fullstack-web-development/>.

31. What is REST? [Електронний ресурс]. — Режим доступу: <http://www.restapitutorial.com/lessons/whatisrest.html>.
(<http://www.restapitutorial.com/lessons/whatisrest.html>)

ДОДАТКИ

Чи обмежена Ваша здатність виконувати повсякденні справи через проблеми зі здоров'ям, які тривають 6 місяців і більше?

Треба позначити одну відповідь.

- a) ні, я не маю жодних обмежень
- b) так, помірно обмежена
- c) так, дуже обмежена

- d) так, повністю обмежена – я не можу їх виконувати
- e) я не хочу відповідати на це запитання

Будь ласка, вкажіть, які у Вас є зараз групи захворювань, що викликають труднощі у виконанні повсякденних справ.

У відповіді на запитання слід вибрати максимально 3 найважливіших захворювання (навіть якщо їх більше), які обмежують нормальну життєву діяльність і є найбільш обтяжливими в повсякденному житті

- a) пошкодження чи захворювання опорно-рухового апарату
- b) ураження очей чи захворювання органів зору
- c) пошкодження чи захворювання органів слуху
- d) серцево-судинні захворювання
- e) неврологічні захворювання
- f) інші захворювання
- g) я не хочу відповідати на це запитання

Яка із вказаних груп захворювань є основною причиною обмежень/труднощів, які Ви відчуваєте?

Для відповіді з'являться максимально 3 групи захворювань, обраних у попередньому питанні та варіант «я не хочу відповідати на це запитання».

- a) пошкодження чи захворювання опорно-рухового апарату
- b) ураження очей чи захворювання органів зору
- c) пошкодження чи захворювання органів слуху
- d) серцево-судинні захворювання
- e) неврологічні захворювання
- f) інші захворювання
- g) я не хочу відповідати на це запитання

Чи є у Вас дійсне свідоцтво про неповносправність, непрацездатність або інвалідність?

Існує дві можливі відповіді:

a) ні – позначають особи, які не мають дійсного свідоцтва про неповносправність, працевздатність (або еквівалентне). Вибір відповіді «ні» закінчує питання.

b) так

Я маю свідоцтво про:

c) неповносправність для осіб віком до 16 років

d) легкий ступінь неповносправності, або про часткову працевздатність, або про доцільність професійної перепідготовки або про III групу інвалідності, або про постійну чи довготривалу працевздатність в сільському господарстві без права на матеріальну допомогу по догляду

e) помірний ступінь неповносправності, або про повну працевздатність, або про II групу інвалідності

f) значний ступінь неповносправності, або про повну працевздатність та неможливість жити самотійно, або про неможливість жити самотійно, або про I групу інвалідності, або про постійну чи довготривалу працевздатність в сільському господарстві з правом на матеріальну допомогу по догляду

Чи у вказаний тиждень Ви мали роботу, але не виконували її тимчасово через відпустку (включаючи відпустку для відпочинку, декретну, батьківську відпустку), хворобу, догляд за іншою особою, погані погодні умови тощо?

a) так

b) ні – стосується всіх інших випадків.

Період активного, безперервного пошуку роботи

Необхідно надати інформацію про період активного, безперервного пошуку роботи, що передує моменту перепису.

За якою спеціальністю Ви працювали за основним місцем роботи?

Надайте інформацію про Вашу професію за основним місцем роботи, де Ви працювали протягом тижня з 25 по 31 березня 2023 року. Зі списку, треба вибрати групу професій, яка включає Вашу професію.

Який вид діяльності здійснювала установа (компанія), яка була Вашим основним місцем роботи?

Треба вказати, до якої категорії/групи відноситься вид господарчої діяльності компанії (установи, підприємства), яка є основним місцем Вашої роботи.

Якщо ця компанія здійснює різну господарчу діяльність, слід вказати діяльність, яка є її основною або переважною.

Групи видів діяльності:

- a) державне управління та національна оборона; освіта; охорона здоров'я та соціальна допомога b) виробництво, промислова переробка
 - c) сільське господарство, лісове господарство, мисливство та рибальство
 - d) гуртова та роздрібна торгівля; ремонт транспортних засобів
 - e) професійна, наукова та технічна діяльність; діяльність у сфері адміністративних послуг та допоміжна діяльність f) транспорт, складська діяльність, готельні послуги, гастрономія, інформація та комунікація, фінанси і страхування, ринок нерухомості g) будівництво h) електроенергія; газ; вода; водяна пара; стоки та відходи; рекультивация i) культура, розваги та відпочинок j) гірництво і видобуток корисних копалин k) інші види діяльності l) я нічого про це не знаю
- Відповідь l) може обрати член родини, що ує ситуацію емігранта і не може дати жодної з попередніх відповідей.

Як часто ви організуєте поїздки на відпочинок за межі свого міста?

Треба позначити одну відповідь.

- a) раз на рік
- b) 2-3 рази на рік
- c) від 5 разів на рік
- d) я не хочу відповідати на це запитання

Будь ласка, вкажіть територіальні межі ваших звичних подорожей.

Треба позначити одну відповідь.

- a) подорожую в межах України
- b) подорожую тільки за кордон
- c) подорожую і всередині, і поза межами України
- d) тільки планую почати подорожувати
- e) я не хочу відповідати на це запитання

У якій компанії ви зазвичай відправляєтесь на відпочинок?

Оберіть до 3 відповідей.

- a) з сім'єю
- b) з друзями
- c) самотійно
- d) з коханою людиною
- e) я не хочу відповідати на це запитання

Скільки країн ви вже відвідали?

Треба позначити одну відповідь.

- a) ще не був(ла) за кордоном
- b) 1-2
- c) 3-5
- d) до 10 країн
- e) більше 10 країн
- f) я не хочу відповідати на це запитання

Який відпочинок ви обираєте найчастіше?

Оберіть до 3 відповідей.

- a) відпочинок на пляжі.
- b) відвідування музеїв та архітектурних пам'яток.
- c) локальний екстремальний відпочинок
- d) знайомство з місцевою гастрокультурою
- e) виїзди на локальну дику природу
- f) я не хочу відповідати на це запитання

Як давно ви користуєтесь нашою платформою?

Треба позначити одну відповідь.

- a) користуюсь вперше
- b) кілька днів
- c) 1-2 тижні
- d) кілька місяців
- e) рік

Які основні переваги нашої платформи ви б відмітили?

Оберіть до 3 відповідей.

- a) зручний інтерфейс
- b) широкий функціонал
- c) дизайн платформи
- d) система зворотнього зв'язку
- e) інформативність
- f) інше

Які недоліки платформи ви помітили під час використання?

(Розгорнуте питання)

Які побажання ви маєте щодо вдосконалення роботи платформи?

(Розгорнуте питання)

