ЛЬВІВСЬКИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ МІЖНАРОДНИХ ЕКОНОМІЧНИХ ВІДНОСИН ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **Веб-застосунок для сервісних центрів побутової техніки**

Виконала: здобувачка першого (бакалаврського) рівня вищої освіти ОПП "Комп'ютерні науки" спеціальності 122 "Комп'ютерні науки"
Михальська Катерина Миколаївна

(підпис)

Науковий керівник к.т.н., доцент
Коцун Володимир Іванович,

(підпис)

Львів – 2024

ЛЬВІВСЬКИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
Факультет міжнародних економічних відносин та інформаційних технологій
Кафедра комп'ютерних наук
Перший (бакалаврський) рівень вищої освіти
Освітньо-професійна програма "Комп'ютерні науки"
Спеціальність 122 "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Завідувач кафедри комп'ютерних наук

_____ Костенко А. В.
(підпис) (прізвище та ініціали)

23.10.2023 р.

ЗАВДАННЯ
до кваліфікаційної роботи
Михальській Катерині Миколаївні
(прізвище, ім'я, по батькові)

1. Тема роботи: Веб-застосунок для сервісних центрів побутової техніки
   керівник роботи ***Коцун Володимир Іванович, к.т.н., доцент***
   (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
   затверджена наказом по Університету №910/03 від 19 жовтня 2023 р.
   Термін здачі здобувачем вищої освіти роботи 12.06.2024 р.
2. Вихідні дані до роботи: літературні джерела, інтернет-джерела, постановка задачі.
3. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1) Аналіз потреб та вимог; 2) Реалізація веб-застосунку для сервісних центрів побутової техніки; 3) Тестування та валідація.
4. Перелік графічного матеріалу до 12 слайдів у форматі **PowerPoint**
6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 23.10.2023 р.

Календарний план проекту

№ з/п	Найменування етапів виконання КР	Терміни виконання етапів	Примітка
1.	Отримання завдання і аналіз інформаційних джерел за темою роботи	до 11.12.2023	
2.	Аналіз сайтів-аналогів	до 20.03.2024	
3.	Написання специфікації вимог	до 22.03.2024	
4.	Створення прототипів інтерфейсу	до 01.04.2024	
5.	Проектування архітектури ПЗ	до 15.04.2024	
6.	Розробка програмного забезпечення та тестування програмного забезпечення	до 22.04.2024	
7.	Оформлення кваліфікаційної роботи. Подання роботи на перевірку керівнику	до 16.05.2024	
8.	Доопрацювання кваліфікаційної роботи з врахуванням зауважень керівника. Подання роботи для перевірки в системі “Антиплагіат”	до 10.06.2024	
9.	Подання кваліфікаційної роботи керівнику для отримання відгуку.	до 11.06.2024	
10.	Представлення кваліфікаційної роботи на зовнішнє рецензування. Подання кваліфікаційної роботи на кафедру комп'ютерних наук	до 12.06.2024	
11.	Попередній захист кваліфікаційної роботи	13.06.2024-14.06.2024	
12.	Захист кваліфікаційної роботи на засіданні екзаменаційної комісії	згідно графіку роботи ЕК	

Здобувачка вищої освіти Михальська К.М.

Керівник роботи Коцун В.І.

Анотація

Кваліфікаційна робота на тему "Веб-застосунок для сервісних центрів побутової техніки" присвячена розробці та впровадженню ефективного веб-інструменту, який сприятиме автоматизації та оптимізації процесів обслуговування клієнтів у сервісних центрах, що займаються ремонтом та обслуговуванням побутової техніки.

Основною метою дослідження є створення інтерактивного та зручного у використанні веб-застосунку, який забезпечить реєстрацію та облік замовлень на ремонт техніки, моніторинг стану виконання замовлень у режимі реального часу, організацію та управління базою даних клієнтів, а також їхньої техніки.

У роботі було проведено аналіз існуючих рішень на ринку програмного забезпечення для сервісних центрів, виявлено їхні переваги та недоліки. На основі цього аналізу було розроблено технічне завдання, архітектуру системи, а також визначено основні функціональні та нефункціональні вимоги до веб-застосунку.

В рамках реалізації проекту були використані сучасні технології веб-розробки, такі як HTML, CSS, JavaScript, а також Python для роботи бекенду та бази даних. Значну увагу приділено забезпеченню безпеки даних та захисту інформації клієнтів.

Результатом роботи є готовий веб-застосунок, який пройшов тестування та готовий до впровадження в сервісних центрах побутової техніки. Запропоноване рішення дозволяє підвищити ефективність роботи сервісних центрів, покращити якість обслуговування клієнтів та знизити витрати на управління процесами ремонту техніки.

Abstract

The thesis on “Web Application for Home Appliance Service Centers” is devoted to the development and implementation of an effective web-based tool that will automate and optimize customer service processes in service centers that repair and maintain home appliances.

The main goal of the study is to create an interactive and easy-to-use web application that will provide registration and accounting of orders for appliance repair, real-time monitoring of the status of orders, organization and management of a database of customers and their appliances.

The paper analyzes existing solutions on the service center software market and identifies their advantages and disadvantages. Based on this analysis, we developed a technical specification, system architecture, and identified the main functional and non-functional requirements for the web application.

As part of the project, we used modern web development technologies such as HTML, CSS, JavaScript, and Pythoon for the backend and database. Considerable attention was paid to ensuring data security and protecting customer information.

The result of the work is a ready-made web application that has been tested and is ready for implementation in home appliance service centers. The proposed solution allows to increase the efficiency of service centers, improve the quality of customer service and reduce the cost of managing the repair processes.

ЗМІСТ

Вступ 7
1 Аналіз потреб та вимог 9
1.1 Пояснення актуальності теми 10
1.2 Визначення мети та завдань дослідження 12
1.3 Огляд літератури та аналогів 13
1.4 Порівняння аналогів веб-застосунків для сервісних центрів побутової техніки 15
1.5 Визначення основних функцій сервісних центрів побутової техніки 20
1.6 Аналіз існуючих проблем та недоліків у роботі сервісних центрів побутової техніки 21
1.7 Визначення вимог до веб-застосунку для сервісних центрів побутової техніки 23
2 Реалізація веб-застосунку для сервісних центрів побутової техніки 26
2.1 Розробка фронтенду веб-застосунку за допомогою HTML, CSS та JavaScript ... 27
2.2 Налаштування та розгортання серверної частини з використанням фреймворка Django у Python 29
2.3 Імплементація бази даних та зв'язку з веб-застосунком 30
2.4 Принцип роботи веб-застосунку 32
2.5 Процеси при роботі з веб-додатком 36
3 Тестування та валідація 42
3.1 Проведення модульних та інтеграційних тестів 43
3.2 Аналіз результатів тестування 45
3.3 Виправлення помилок та вдосконалення функціональності 48
Висновки 50
Список використаних джерел 53

ВСТУП

Сучасний світ проживає у епоху стрімкого розвитку технологій, де побутова техніка стала неодмінною складовою повсякденного життя. Вона полегшує наші щоденні обов'язки та забезпечує комфорт у домашньому середовищі. Проте, як будь-який технічний пристрій, вона не є вічною інвестицією, адже з часом може виникати необхідність у її обслуговуванні та ремонті. У зв'язку з цим, сервісні центри побутової техніки стають важливими ланками у вирішенні проблем користувачів та підтримки функціональності їхнього обладнання.

Однак, навіть у цьому напрямку, як і в багатьох інших сферах, існують прогалини та проблеми, які вимагають вдосконалення. Традиційні методи обслуговування техніки часто зводяться до звернення до сервісних центрів за допомогою по телефону або особистого візиту, що може бути часом витратним та неефективним. Більш того, деякі сервісні центри можуть бути недостатньо організованими або надавати неповний спектр послуг, що призводить до незадоволення клієнтів та втрати довіри до бренду.

У цьому контексті, розвиток сучасних технологій, зокрема веб-додатків, відкриває широкі можливості для оптимізації процесів обслуговування побутової техніки. Розробка веб-застосунку спрямованого на оптимізацію роботи сервісних центрів побутової техніки може стати важливим кроком у напрямку покращення якості обслуговування, зручності для клієнтів та ефективності управління процесами ремонту техніки.

Ця кваліфікаційна робота спрямована на розробку та дослідження веб-застосунку для сервісних центрів побутової техніки з метою підвищення ефективності обслуговування клієнтів, оптимізації процесів ремонту техніки та полегшення взаємодії між клієнтами та фахівцями. Для досягнення цієї мети робота буде включати аналіз потреб та вимог користувачів, проектування архітектури системи, реалізацію веб-застосунку, тестування та валідацію, оцінку ефективності та порівняння з існуючими аналогами.

У подальших розділах роботи будуть розглянуті детальніше кожен з етапів розробки веб-застосунку, висвітлені технології, методи та підходи, що використовуватимуться для досягнення поставленої мети. Крім того, будуть проаналізовані результати роботи, надані висновки та рекомендації для подальшого вдосконалення та застосування розробленого веб-застосунку у практичній діяльності сервісних центрів побутової техніки.

1 АНАЛІЗ ПОТРЕБ ТА ВИМОГ

Вивчення потреб та вимог користувачів є ключовим етапом при розробці будь-якого продукту чи сервісу. У випадку веб-застосунку для сервісних центрів побутової техніки, розуміння потреб та очікувань як самого бізнесу, так і його клієнтів, визначатиме успішність та ефективність розробленого рішення. Цей розділ присвячений аналізу потреб та вимог різних зацікавлених сторін, що дозволить зрозуміти, які функціональності та можливості має включати веб-застосунок для досягнення максимальної задоволеності користувачів та підвищення ефективності обслуговування.

Сервісні центри побутової техніки є основними користувачами веб-застосунку, тому важливо спрямувати аналіз на їхні потреби та вимоги.

- *Управління замовленнями та ремонтами*: Одні з основних потреб сервісних центрів - це ефективне управління замовленнями на ремонт. Вони потребують системи, яка дозволить їм легко приймати та реєструвати заявки на ремонт, призначати їх фахівцям, відстежувати статуси та дедлайни, а також забезпечити швидкий доступ до історії ремонтів.
- *Спілкування з клієнтами*: Важливою вимогою є можливість забезпечити зручну та ефективну комунікацію з клієнтами. Це включає можливість приймати запити на ремонт через веб-застосунок, надавати сповіщення про статус ремонту, а також забезпечити зручний канал зворотного зв'язку для звітування про завершення робіт.
- *Управління запасами та комплектуючими*: Багато сервісних центрів потребують системи управління запасами та комплектуючими для ефективного контролю за наявними ресурсами та запасами для ремонту.

Клієнти сервісних центрів побутової техніки також мають свої власні потреби та очікування від веб-застосунку.

- *Зручне та швидке запис на ремонт*: Одні з основних потреб клієнтів - це можливість швидко та зручно записатися на ремонт своєї техніки. Вони хочуть мати можливість зробити це онлайн, без довгих черг або очікування на гарячій лінії.

- *Сповіщення про статус ремонту*: Клієнти бажають отримувати регулярні оновлення про статус свого ремонту - чи вже почали його виконувати, на якому етапі він знаходиться, коли він буде готовий до видачі.
- *Простота та зручність використання*: Клієнти очікують, що веб-застосунок буде простим та зручним у використанні. Вони хочуть мати можливість легко зрозуміти, як користуватися всіма його функціями без особливих зусиль.

Адміністратори та фахівці, які працюють у сервісних центрах, також мають свої власні потреби та вимоги до веб-застосунку.

- *Зручна панель управління*: Адміністраторам та фахівцям потрібна зручна та функціональна панель управління, яка дозволяє легко керувати всіма аспектами роботи сервісного центру.
- *Швидкий доступ до інформації*: Фахівці та адміністратори повинні мати можливість швидко знаходити необхідну інформацію про замовлення, клієнтів та історію ремонтів.
- *Забезпечення безпеки даних*: Оскільки веб-застосунок буде містити конфіденційну інформацію про клієнтів та їхні технічні проблеми, важливо забезпечити високий рівень захисту та безпеки даних.

Аналіз потреб та вимог вищезгаданих зацікавлених сторін дозволить зрозуміти основні функціональність та можливості, які повинен мати веб-застосунок для сервісних центрів побутової техніки. Враховуючи ці потреби, ми зможемо розробити продукт, який відповідає вимогам усіх зацікавлених сторін та забезпечить максимальну задоволеність користувачів.

1.1 Пояснення актуальності теми

Сучасний ритм життя прискорюється кожним днем, і разом з тим зростає наша залежність від побутової техніки. Від холодильників до пральних машин, від телевізорів до мікрохвильових печей - ці пристрої стали не просто розкішшю, а невід'ємною частиною нашого повсякденного життя. Проте, навіть

найсучасніша техніка може вийти з ладу чи потребувати обслуговування в будь-який момент, створюючи необхідність у швидкому та ефективному ремонті.

Така потреба у ремонті побутової техніки створює попит на послуги сервісних центрів. Ці центри відіграють критичну роль у підтримці функціональності техніки та задоволенні потреб користувачів. Вони надають широкий спектр послуг, включаючи діагностику, ремонт, заміну компонентів та консультації щодо експлуатації та обслуговування техніки. Проте, у зв'язку з постійним зростанням кількості техніки у використанні, потребою у її обслуговуванні та ремонті, сервісні центри стикаються з низкою викликів та проблем, які необхідно вирішити для забезпечення якісного обслуговування клієнтів.

Однією з головних проблем, які виникають у сервісних центрах, є недостатня ефективність та організація процесів ремонту. Традиційні методи обслуговування, такі як прийом клієнтів за допомогою телефонного дзвінка або особистого візиту, можуть бути часо- та ресурсоємкими. Крім того, брак систематизації даних та недостатня автоматизація процесів можуть призвести до збоїв у спілкуванні з клієнтами, втрати інформації та погіршення загального досвіду користувача.

Ще однією проблемою є недостатня інтерактивність та зручність для клієнтів. У світі, де майже все можна зробити онлайн, від покупок до резервації подорожей, важливо, щоб і обслуговування побутової техніки було також максимально зручним та доступним. Багато клієнтів бажають мати можливість записатися на ремонт або отримати консультацію без відвідування сервісного центру, а також отримувати сповіщення про статус свого замовлення чи ремонту у реальному часі.

Крім того, відмінність в якості обслуговування може призвести до втрати довіри клієнтів до бренду. Якщо клієнт отримує неадекватну відповідь, зустрічається зі затримками у ремонті або втрачає контакт з сервісним центром, ймовірність того, що він знову звернеться за допомогою, різко зменшується.

Таким чином, покращення якості обслуговування та забезпечення задоволеності клієнтів є важливою складовою успіху будь-якого бізнесу.

Усі ці проблеми та виклики підкреслюють актуальність теми розробки веб-застосунку для сервісних центрів побутової техніки. Шляхом використання сучасних технологій, таких як веб-додатки, можна ефективно оптимізувати процеси обслуговування, підвищити якість обслуговування та забезпечити зручність та задоволення клієнтів. Розробка веб-застосунку для сервісних центрів побутової техніки відкриває шлях до покращення якості обслуговування, оптимізації внутрішніх процесів та збільшення задоволеності клієнтів, що робить її актуальною та важливою для подальшого розвитку галузі обслуговування побутової техніки.

1.2 Визначення мети та завдань дослідження

Мета цієї дослідницької роботи полягає в розробці та аналізі веб-застосунку для сервісних центрів побутової техніки з метою оптимізації процесів обслуговування та поліпшення взаємодії з клієнтами. Цей веб-застосунок буде спрямований на підвищення ефективності роботи сервісних центрів, зручності для клієнтів та покращення загального досвіду користувача. Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. **Аналіз потреб та вимог користувачів**: Першим кроком буде аналіз поточних потреб та вимог користувачів сервісних центрів побутової техніки. Це включатиме вивчення їхніх основних проблем та вимог до оптимального обслуговування, а також виявлення недоліків у поточних практиках.
2. **Проектування архітектури системи**: Наступним етапом буде проектування архітектури веб-застосунку. Це охоплюватиме вибір оптимальних технологій, структуру бази даних, дизайн інтерфейсу користувача та визначення функціональності.
3. **Реалізація веб-застосунку**: На даному етапі буде реалізована розроблена архітектура веб-застосунку. Це включатиме розробку фронтенду та

бекенду веб-застосунку, створення бази даних та реалізацію необхідної логіки програми.

4. **Тестування та валідація**: Після реалізації веб-застосунку буде проведено тестування для перевірки його працездатності, стабільності та відповідності вимогам. Також буде здійснено валідацію з метою впевненості у відповідності розробленого продукту потребам користувачів.
5. **Оцінка ефективності та порівняння з аналогами**: Після завершення розробки та тестування веб-застосунку буде проведена оцінка його ефективності та порівняння з існуючими аналогами. Це дозволить визначити переваги та недоліки розробленого продукту порівняно з існуючими рішеннями на ринку.
6. **Висновки та рекомендації**: Заключний етап полягатиме у формулюванні висновків щодо результатів дослідження та розробки веб-застосунку. Також будуть сформульовані рекомендації щодо подальшого вдосконалення та використання розробленого продукту в практичній діяльності сервісних центрів побутової техніки.

Загальною метою дослідження є створення веб-застосунку, який відповідатиме потребам та вимогам користувачів, підвищить ефективність роботи сервісних центрів та забезпечить зручність та задоволення клієнтів. Для досягнення цієї мети необхідно вивчити, проаналізувати та вирішити ряд технічних та організаційних завдань, що будуть детально розглянуті в подальших розділах дослідницької роботи.

1.3 Огляд літератури та аналогів

Огляд літератури та аналогів є важливим етапом дослідження, оскільки дозволяє ознайомитися з попередніми дослідженнями, технологіями та рішеннями, які були використані в схожих проектах. Цей етап дає змогу зрозуміти, які вже існуючі підходи можуть бути корисними для подальшого

вдосконалення та розробки веб-застосунку для сервісних центрів побутової техніки.

Один із ключових аспектів огляду літератури є вивчення сучасних технологій та фреймворків, що використовуються для розробки веб-застосунків. Наприклад, дослідження фронтенд-фреймворків, таких як React.js, Angular або Vue.js, дозволить визначити оптимальний вибір для створення користувацького інтерфейсу веб-застосунку. Також важливо вивчити серверні технології, такі як Django[3], Flask у Python або Node.js, які дозволяють створювати потужні та масштабовані веб-додатки.

Проведення аналізу існуючих веб-застосунків, що використовуються для обслуговування побутової техніки, дозволить зрозуміти їхні переваги та недоліки. Наприклад, такі веб-платформи, як RepairDesk, ServiceMax або FieldAware, надають можливості для управління замовленнями на ремонт, спілкування з клієнтами та інші функції, які можуть бути корисними для нашого веб-застосунку.

Академічні дослідження та публікації можуть також бути корисним джерелом інформації щодо сучасних підходів до обслуговування техніки. Наприклад, статті про впровадження технологій Інтернету речей (IoT) у побутову техніку або дослідження щодо використання штучного інтелекту для прогнозування поломок та ремонту техніки можуть бути корисними для подальшого розвитку веб-застосунку.

Важливо вивчити відгуки користувачів та рейтинги існуючих веб-застосунків для обслуговування побутової техніки. Це дозволить зрозуміти, які аспекти їм подобаються, а які можна покращити. З цією інформацією можна врахувати побажання користувачів при розробці нашого веб-застосунку та забезпечити його відповідність їх очікуванням.

Окрім того, важливо вивчити тенденції та інновації у галузі технологій та обслуговування. Наприклад, розвиток технологій Інтернету речей (IoT), штучного інтелекту та аналітики даних може вплинути на розвиток нашого веб-застосунку та дозволити впровадити новаторські функції та можливості.

Враховуючи огляд літератури та аналогів, ми матимемо змогу зрозуміти сучасний стан галузі та визначити найбільш перспективні та ефективні підходи для розробки веб-застосунку для сервісних центрів побутової техніки.

1.4 Порівняння аналогів веб-застосунків для сервісних центрів побутової техніки

У сучасному світі зростає популярність використання веб-застосунків для управління та надання послуг сервісних центрів побутової техніки. Ці застосунки дозволяють зручно і ефективно вести облік замовлень, управляти робочими процесами, спрощувати взаємодію з клієнтами та багато іншого. Проте, на ринку існує ряд аналогічних веб-застосунків, кожен з яких має свої переваги та недоліки. У цьому розділі ми проаналізуємо кілька з них для того, щоб допомогти вам обрати найбільш підходящий варіант для вашого сервісного центру.

1. Pyramida Service

Переваги:

- Інтуїтивний і легкий у використанні інтерфейс.
- Широкий спектр функцій, включаючи облік замовлень, розклад майстрів, звіти та аналітику.
- Можливість інтеграції з різними платформами та системами CRM.

Недоліки:

- Високі витрати на встановлення та підтримку.
- Обмежена можливість налаштування під конкретні потреби сервісного центру.
- Відсутність мобільного додатку, що може ускладнити роботу на виїзді.

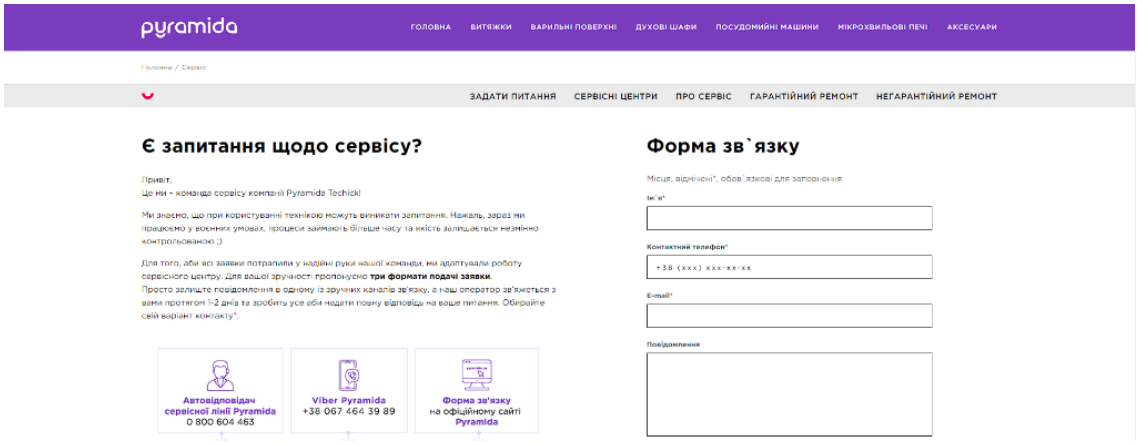


Рисунок 1.1 – Головне вікно сайту pyramida

2. Bosch Siemens Service

Переваги:

- Гнучка система налаштувань, яка дозволяє адаптувати програмне забезпечення під потреби конкретного бізнесу.
- Інтегрована система SMS-повідомлень для сповіщення клієнтів про статус ремонту.
- Мобільний додаток для майстрів, що полегшує ведення обліку та спілкування на виїзді.

Недоліки:

- Не такий розгалужений функціонал порівняно з іншими веб-застосунками.
- Відсутність деяких продвинутих функцій, таких як автоматичне створення замовлень з допомогою штучного інтелекту.



Рисунок 1.2 – Головне вікно сайту bosch

3. Karcher Service

Переваги:

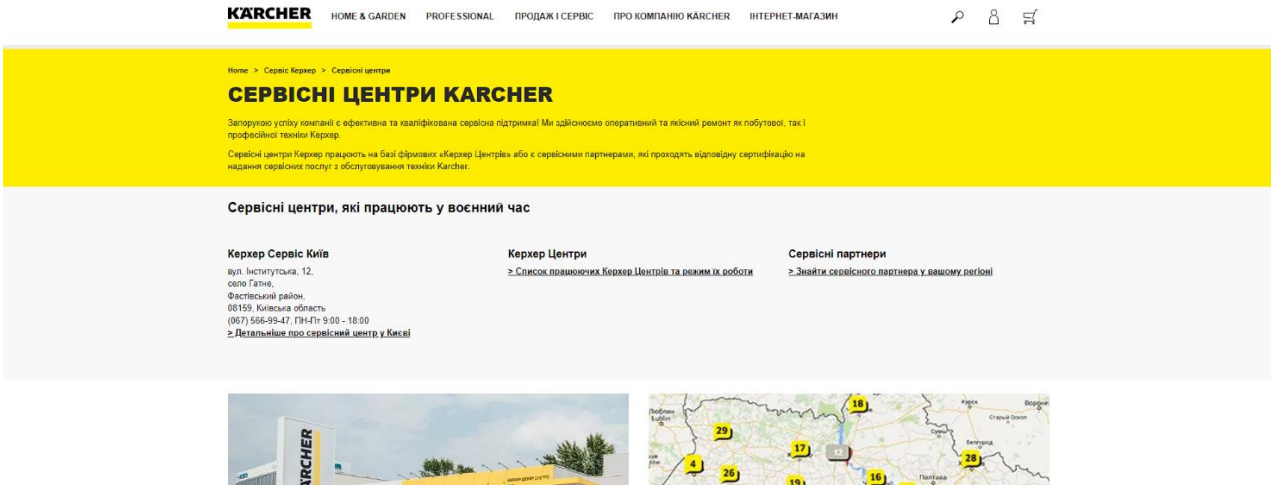
- Потужна система управління задачами та розкладом, що ідеально підходить для сервісних центрів з великим обсягом роботи.
- Інтегрована система GPS для відстеження руху майстрів та оптимізації маршрутів.
- Ефективна система звітності та аналітики для контролю над робочим процесом.

Недоліки:

- Складний для освоєння інтерфейс, який може вимагати додаткового часу для навчання персоналу.

- Високі витрати на підтримку та оновлення програмного забезпечення.

Рисунок 1.3.3 – Головне вікно сайту karcher



4. Philips Service

Переваги:

- Спеціалізована платформа для сервісних центрів, що спеціалізуються на обслуговуванні побутової техніки.
- Простий і зрозумілий інтерфейс, який зменшує час на навчання персоналу.
- Інтегрована система оцінювання задоволеності клієнтів, що дозволяє отримувати зворотний зв'язок і вдосконалювати сервіс.

Недоліки:

- Обмежений функціонал у порівнянні з загальністю веб-застосунків для сервісних центрів.
- Відсутність деяких продвинутих можливостей, таких як інтеграція з системами CRM.

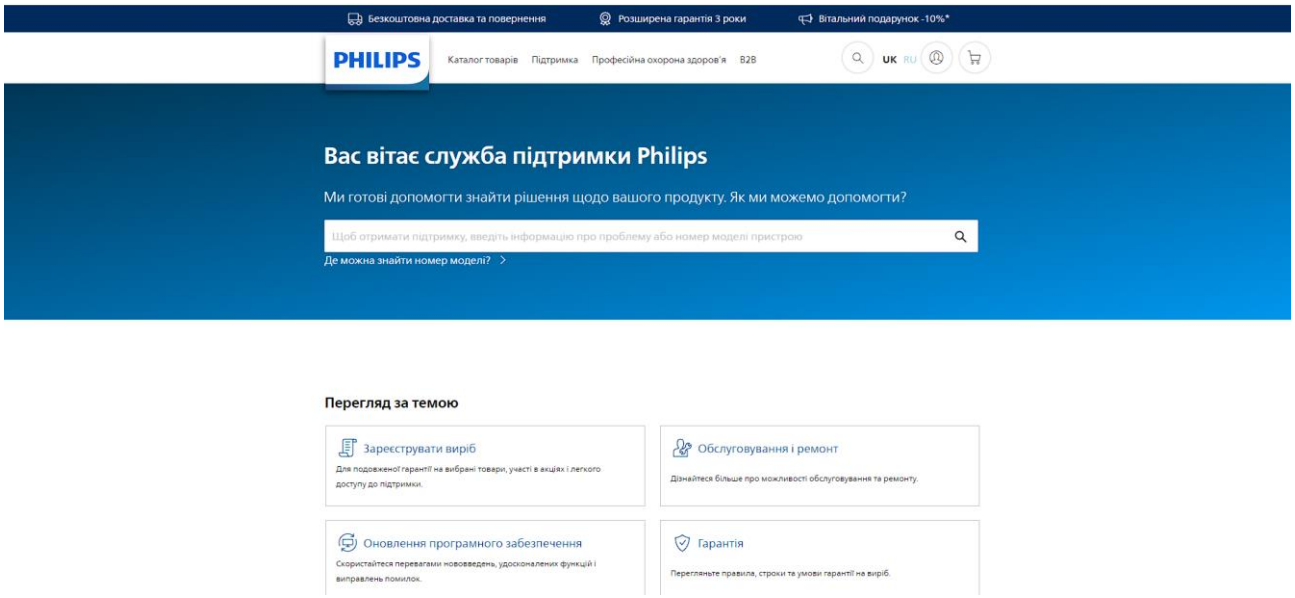


Рисунок 1.4 – Головне вікно сайту philips

5. Microtron Service

Переваги:

- Повний набір інструментів для управління бізнесом, включаючи CRM, фінансовий облік, розклади роботи та інше.
- Велика спільнота користувачів та підтримка від розробників.
- Високий рівень безпеки даних та захисту приватності.

Недоліки:

- Високі витрати на встановлення та підтримку.
- Складний інтерфейс, який може вимагати додаткового часу для освоєння.

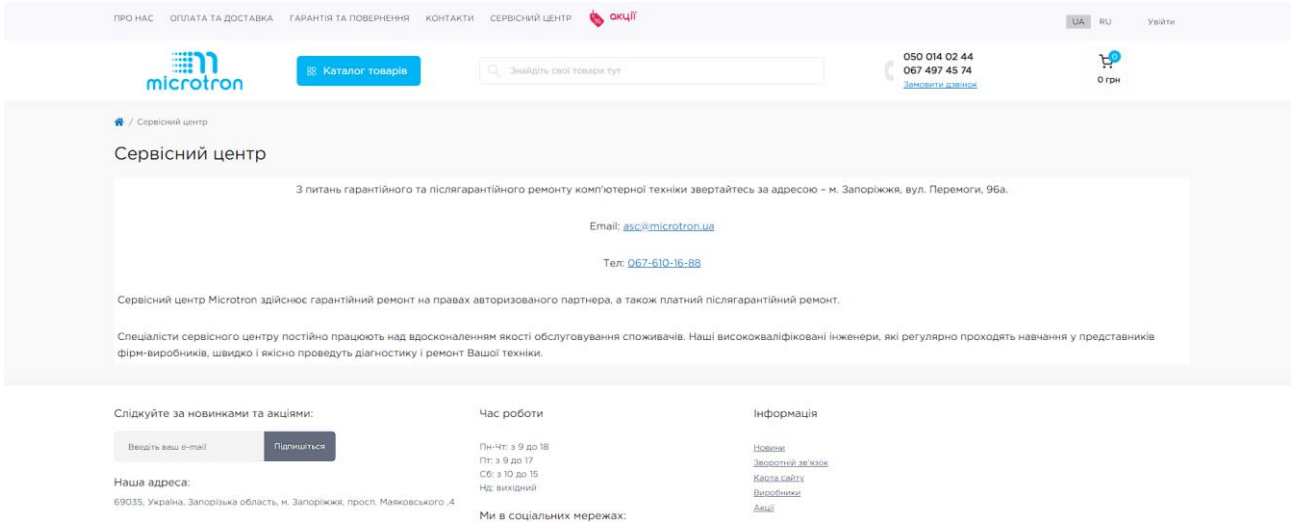


Рисунок 1.5 – Головне вікно сайту microtron

Кожен з вищезазначених веб-застосунків має свої переваги та недоліки, і вибір конкретного рішення повинен базуватися на потребах, можливостях та ресурсах вашого сервісного центру. Перед прийняттям рішення рекомендується ретельно проаналізувати кожну з альтернатив та врахувати всі аспекти їх використання.

1.5 Визначення основних функцій сервісних центрів побутової техніки

Сервісні центри побутової техніки відіграють важливу роль у житті сучасного суспільства, забезпечуючи обслуговування та ремонт різноманітної побутової техніки. Їхні функції необхідні для забезпечення функціональності, безпеки та комфорту користувачів техніки. Давайте розглянемо основні функції, які виконують сервісні центри побутової техніки.

Перша та найважливіша функція сервісного центру - це діагностика проблем та визначення причин несправностей побутової техніки. Фахівці здійснюють аналіз стану пристрою, проводять тестування та перевірку різних компонентів, щоб визначити, що саме викликає проблему. Оцінка проблеми дозволяє точно визначити робочий обсяг та витрати на ремонт.

Однією з головних функцій сервісних центрів є виконання ремонтних робіт та заміна пошкоджених або відстарчих компонентів побутової техніки. Це може включати заміну електронних плат, моторів, сенсорів, дисплеїв, дверцят, пультів управління та інших деталей, що вимагають підтримки або заміни.

Сервісні центри забезпечують не лише ремонт, а й обслуговування побутової техніки. Це включає профілактичні перевірки, налаштування, очищення та змазування механізмів, щоб запобігти майбутнім проблемам. Також надається технічна підтримка, консультації з експлуатації техніки та рекомендації щодо підтримки її роботи в оптимальному стані.

Сервісні центри часто здійснюють постачання та продаж запчастин для побутової техніки. Це може включати оригінальні запчастини від виробників або альтернативні запчастини, які відповідають стандартам якості та безпеки.

У сучасних умовах технологічного розвитку, де багато побутових пристроїв підтримують підключення до Інтернету, сервісні центри також можуть здійснювати віддалену діагностику та сервіс. Це означає, що фахівці можуть аналізувати стан техніки віддалено через Інтернет, виявляти проблеми та надавати рекомендації щодо їх вирішення.

Багато сервісних центрів побутової техніки надають гарантійні обов'язки на виконані ремонтні роботи або замінені компоненти. Це забезпечує клієнтів захистом від несправностей та дефектів, які можуть виникнути після ремонту.

Деякі сервісні центри пропонують консультативні послуги та навчальні курси для користувачів побутової техніки. Це може включати навчання правильному використанню техніки, підказки щодо підтримки та догляду за нею, а також поради щодо вибору найбільш ефективних та надійних пристроїв.

Узагальнюючи, основні функції сервісних центрів побутової техніки включають діагностику, ремонт, обслуговування, постачання запчастин, віддалену діагностику, гарантійні обов'язки, консультативні послуги та навчання. Ці функції мають на меті забезпечити надійність, безпеку та комфорт у використанні побутової техніки для кінцевих користувачів.

1.6 Аналіз існуючих проблем та недоліків у роботі сервісних центрів побутової техніки

Сервісні центри побутової техніки відіграють важливу роль у забезпеченні функціональності та довговічності побутових пристроїв для користувачів. Однак, у процесі їхньої роботи існують ряд проблем та недоліків, які можуть впливати на якість обслуговування та задоволеність клієнтів. Давайте розглянемо деякі з них:

1. Затримки у виконанні ремонтів: Час виконання ремонтів може бути однією з найбільш проблемних областей у роботі сервісних центрів. Затримки можуть бути викликані багатьма факторами, такими як недостатньо швидкий доступ до запчастин, велика кількість замовлень на обробці, нестача

кваліфікованих фахівців та інші причини. Це може призводити до незадоволення клієнтів та втрати довіри до сервісного центру.

2. Нестабільна якість ремонтів: Іншою проблемою є непостійна якість ремонтних робіт, що виконуються сервісними центрами. Це може бути викликано неякісними запчастинами, непрофесійним підходом до виконання ремонтів, відсутністю достатньої кваліфікації у фахівців або недостатньою увагою до деталей. Низька якість ремонту може призвести до появи нових проблем з технікою або навіть до погіршення стану пристрою.

3. Високі витрати на ремонт: Деякі сервісні центри можуть встановлювати завищені ціни на ремонтні роботи та запчастини, що може стати проблемою для клієнтів. Високі витрати можуть зробити сервіс обслуговування недоступним для деяких користувачів, а також призвести до втрати конкурентоспроможності сервісного центру на ринку.

4. Нестабільна комунікація: Недоліком багатьох сервісних центрів є неефективна комунікація з клієнтами. Це може включати погане інформування про статус ремонту, відсутність відповідей на запити клієнтів, недостатню доступність для спілкування або невідповідність обіцянок щодо термінів та умов обслуговування.

5. Відсутність технологічних інновацій: Багато сервісних центрів залишаються відсталими у використанні сучасних технологій та інновацій. Недостатнє впровадження технологічних рішень може призвести до неефективності процесів обслуговування та втрати конкурентоспроможності.

6. Недостатня безпека даних: Збереження конфіденційності та безпеки даних клієнтів є критично важливим аспектом у роботі сервісних центрів. Недостатня захищеність персональної інформації може призвести до витоку конфіденційної інформації, порушення довіри клієнтів та навіть правових проблем.

7. Неадекватна інфраструктура та обладнання: Деякі сервісні центри можуть мати обмежену або застарілу інфраструктуру та обладнання, що може

обмежувати їхню здатність надавати якісні послуги та ефективно виконувати ремонтні роботи.

Узагальнюючи, існують різноманітні проблеми та недоліки у роботі сервісних центрів побутової техніки, які можуть впливати на якість обслуговування та задоволеність клієнтів. Вирішення цих проблем вимагає комплексного підходу та впровадження ефективних стратегій управління якістю та інновацій.

1.7 Визначення вимог до веб-застосунку для сервісних центрів побутової техніки

Розробка веб-застосунку для сервісних центрів побутової техніки є складним та відповідальним завданням, оскільки вимагає врахування потреб різних зацікавлених сторін та вирішення широкого спектру завдань. Для того щоб веб-застосунок був ефективним і корисним для користувачів та бізнесу, необхідно чітко визначити його вимоги. Нижче наведено детальний аналіз вимог до веб-застосунку для сервісних центрів побутової техніки.

1. Функціональні вимоги:

- *Управління замовленнями*: Веб-застосунок повинен забезпечувати можливість реєстрації нових замовлень на ремонт побутової техніки, призначення їх фахівцям, відстеження статусу ремонту та реєстрацію виконаних робіт.
- *Клієнтська підтримка*: Наявність чату або системи квитків для спілкування з клієнтами, можливість надавати інформацію про статус ремонту та відповідати на запитання.
- *Управління запасами*: Система управління запасами та комплектуючими, яка дозволить вести облік наявних запасів та забезпечити їхню достатність для виконання ремонтних робіт.
- *Система графіка роботи*: Можливість встановлення графіку роботи для фахівців та співробітників, що дозволить ефективно розподіляти завдання та забезпечити покриття піків навантаження.

- *Аналітика та звітність*: Наявність засобів аналізу даних про роботу сервісного центру, звітів про виконані роботи, фінансову звітність та інші аналітичні функції.

2. Нефункціональні вимоги:

- *Безпека*: Веб-застосунок повинен мати високий рівень безпеки, забезпечуючи конфіденційність та цілісність даних клієнтів та бізнесу.
- *Швидкодія*: Вимога до продуктивності веб-застосунку, яка полягає в швидкому відгуку на запити користувачів та ефективному виконанні операцій.
- *Масштабованість*: Здатність веб-застосунку працювати ефективно при збільшенні обсягу роботи та кількості користувачів.
- *Зручність використання*: Інтуїтивний та зручний інтерфейс для користувачів, який дозволить їм легко взаємодіяти з веб-застосунком без спеціальної підготовки.
- *Сумісність*: Веб-застосунок повинен бути сумісним з різними пристроями та операційними системами для забезпечення доступності для всіх користувачів.

3. Вимоги до інтеграції:

- *Інтеграція з існуючими системами*: Здатність веб-застосунку інтегруватися з існуючими системами обліку, CRM, ERP та іншими, що використовуються в сервісному центрі.
- *Підтримка API*: Наявність API для спрощення інтеграції з іншими системами та розроблення власних розширень або додатків.

Враховуючи ці вимоги, розробники зможуть створити веб-застосунок, який відповідає потребам сервісних центрів побутової техніки та забезпечить ефективну та зручну роботу як для користувачів, так і для фахівців.

2 РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ СЕРВІСНИХ ЦЕНТРІВ ПОБУТОВОЇ ТЕХНІКИ

В цьому розділі розглянемо процес реалізації веб-застосунку для сервісних центрів побутової техніки. Від архітектури та технічних рішень до розробки і тестування, кожен етап важливий для успішного створення функціонального та ефективного продукту.

Перший крок у реалізації веб-застосунку - це ретельний аналіз вимог, що був проведений у попередньому розділі. На основі цього аналізу формулюються вимоги до функціональності, нефункціональні вимоги та вимоги до інтеграції. Важливо визначити, які саме функції має виконувати веб-застосунок та які технічні рішення потрібно прийняти для їх реалізації.

Після аналізу вимог приходить час на вибір технологій, які будуть використовуватися для розробки веб-застосунку. Це включає вибір програмних мов, фреймворків, баз даних, інструментів для розробки фронтенду та бекенду, а також будь-яких інших технологій, необхідних для реалізації потрібної функціональності.

Після вибору технологій розробники приступають до проектування архітектури веб-застосунку. Це включає визначення структури додатку, розподіл функціональності між клієнтською та серверною частинами, вибір паттернів проектування та вирішення питань безпеки та масштабованості.

Після завершення проектування розробники приступають до фактичної розробки веб-застосунку. Цей етап включає написання коду як для клієнтської, так і для серверної частини, розробку інтерфейсу користувача, інтеграцію зовнішніх сервісів та реалізацію всіх необхідних функцій.

Після завершення розробки веб-застосунок піддається тестуванню. Це включає проведення різних видів тестів, таких як модульні тести, функціональні тести, інтеграційні тести та випробування на витривалість. Метою тестування є виявлення та виправлення будь-яких помилок та дефектів, а також перевірка відповідності веб-застосунку вимогам.

Після успішного завершення тестування веб-застосунок готовий до випуску та розгортання. Це включає підготовку середовища для роботи веб-застосунку, налаштування серверів, розгортання коду та встановлення всіх необхідних залежностей.

Після розгортання веб-застосунок потребує постійної підтримки та післярелізного супроводу. Це включає вирішення проблем, виявлених користувачами, внесення поліпшень та оновлень, моніторинг продуктивності та безпеки, а також надання підтримки клієнтам.

Узагальнюючи, процес реалізації веб-застосунку для сервісних центрів побутової техніки включає в себе кілька етапів, починаючи від аналізу вимог і вибору технологій до розробки, тестування, розгортання та підтримки. Кожен етап важливий для успішного завершення проекту та створення високоякісного веб-застосунку, який задовольняє потреби користувачів та бізнесу.

2.1 Розробка фронтенду веб-застосунку за допомогою HTML, CSS та JavaScript

Фронтенд веб-застосунку є тим, що бачить користувач при взаємодії з додатком у браузері. Це інтерфейс, з якого користувач може взаємодіяти з різними функціями та можливостями веб-застосунку. У цьому розділі ми розглянемо процес розробки фронтенду з використанням HTML, CSS та JavaScript.

1. HTML (HyperText Markup Language):

HTML є основою будь-якого веб-застосунку. Це мова розмітки, яка використовується для створення структури та змісту веб-сторінок. Зазвичай розробка починається зі створення HTML-шаблонів, які визначають основну структуру сторінок.

- **Створення основних елементів:** Перш ніж розпочати розробку, визначаються основні елементи веб-застосунку, такі як заголовки, меню, футер та інші. Вони створюються за допомогою HTML-тегів.

- **Форми і введення даних:** Форми дозволяють користувачам вводити дані та взаємодіяти з веб-застосунком. За допомогою HTML можна створювати різноманітні форми для збору інформації від користувачів.
- **Вставка медіа та зображень:** HTML дозволяє вставляти різноманітні медіа-файли, такі як зображення, відео та аудіо, щоб зробити веб-застосунок більш привабливим та інформативним.

2. CSS (Cascading Style Sheets):

CSS відповідає за вигляд і оформлення веб-застосунку. Він використовується для стилізації HTML-елементів та надання їм вигляду, кольору, розміру та розташування.

- **Визначення стилів:** CSS використовується для визначення стилів різних елементів, таких як шрифти, колір фону, розмір тексту та інші параметри.
- **Респонсивний дизайн:** CSS дозволяє створювати респонсивний дизайн, який автоматично адаптується до різних розмірів екранів, щоб забезпечити зручне відображення веб-застосунку на різних пристроях.
- **Анімація та переходи:** CSS дозволяє створювати анімацію та переходи, щоб зробити веб-застосунок більш привабливим та динамічним.

3. JavaScript:

JavaScript використовується для створення інтерактивності на сторінках веб-застосунку. Він дозволяє виконувати різні дії на сторінці, такі як валідація форм, анімація, взаємодія з користувачем та інші.

- **Валідація форм:** JavaScript може використовуватися для валідації даних, які вводяться в форми, щоб переконатися, що вони відповідають певним критеріям.
- **Взаємодія з користувачем:** JavaScript дозволяє створювати інтерактивні елементи, такі як кнопки, меню та попапи, які реагують на дії користувача.
- **Запити до сервера:** JavaScript може використовуватися для виконання асинхронних запитів до сервера, що дозволяє оновлювати сторінку без перезавантаження та отримувати оновлену інформацію.

Узагальнюючи, розробка фронтенду веб-застосунку з використанням HTML, CSS та JavaScript є складним та важливим процесом, який вимагає уважності та професіоналізму. Користуючись цими технологіями, розробники можуть створити привабливий, функціональний та ефективний інтерфейс для веб-застосунку, що задовольнить потреби користувачів.

2.2 Налаштування та розгортання серверної частини з використанням фреймворка Django у Python

Фреймворк Django у Python є одним з найпопулярніших і потужних інструментів для розробки веб-застосунків. Він надає широкий функціонал для швидкої та ефективної розробки серверної частини веб-застосунків. У цьому розділі ми розглянемо процес налаштування та розгортання серверної частини з використанням фреймворка Django.

Першим кроком у розробці веб-застосунку на Django є налаштування середовища розробки. Для цього необхідно встановити Python та Django на ваш комп'ютер. Після цього ви можете створити новий проект Django за допомогою команди django-admin startproject, що створить основну структуру проекту.

Далі ви можете створити додатки та моделі, які будуть використовуватися у вашому веб-застосунку. Django надає зручний механізм для визначення моделей бази даних та їх відносин, що дозволяє швидко створювати складні структури даних.

Після створення моделей ви можете налаштувати маршрути та представлення вашого веб-застосунку. Django використовує механізм URL-маршрутизації для визначення шляхів до різних частин вашого додатку, а також представлення для обробки запитів та відображення відповідей.

Django підтримує різні типи баз даних, такі як SQLite, PostgreSQL, MySQL та інші. Ви можете налаштувати підключення до вашої бази даних у файлі settings.py вашого проекту Django. Тут ви також можете визначити моделі даних, які будуть використовуватися у вашому додатку.

Для роботи зі статичними файлами, такими як CSS[10], JavaScript[6] та зображення, а також з медіа-файлами, такими як зображення, відео та аудіо, Django надає спеціальні можливості. Ви можете налаштувати шляхи до цих файлів у файлі settings.py та використовувати їх у вашому веб-застосунку.

Після завершення розробки вашого веб-застосунку ви готові до його розгортання на сервері. Це може бути власний сервер або хмарна платформа, така як AWS, Heroku, або DigitalOcean. Ви можете використовувати різні інструменти для розгортання, такі як Docker, Ansible або просто налаштувати сервер вручну.

Після розгортання ви також повинні налаштувати середовище продукції для вашого веб-застосунку. Це включає в себе налаштування безпеки, моніторингу та логування, оптимізацію продуктивності та інші аспекти, що важливі для стабільної та ефективної роботи вашого додатку в реальному середовищі.

Після розгортання вашого веб-застосунку важливо забезпечити його підтримку та оновлення. Це включає в себе вирішення проблем, які виникають у реальному середовищі, оновлення до нових версій фреймворка та сторонніх бібліотек, а також надання підтримки користувачам вашого додатку.

Узагальнюючи, розробка серверної частини з використанням фреймворка Django у Python - це складний та відповідальний процес, який вимагає уважності та професіоналізму. Проте, завдяки потужним можливостям Django та широкому спектру інструментів для розгортання та налаштування, ви зможете створити високоякісний та ефективний веб-застосунок, який задовольнить потреби користувачів.

2.3 Імплементація бази даних та зв'язку з веб-застосунком

Імплементація бази даних та забезпечення зв'язку з веб-застосунком є ключовою частиною розробки будь-якого веб-додатку. Ефективна організація бази даних дозволяє зберігати та обробляти дані, необхідні для роботи додатку, а правильний зв'язок з веб-застосунком забезпечує швидкий та надійний доступ

до цих даних. У цьому розділі ми розглянемо процес імплементації бази даних та зв'язку з веб-застосунком з використанням різних технологій.

Першим кроком у імплементації бази даних є вибір типу бази даних, який найкраще відповідає потребам вашого додатку. Ви можете вибрати реляційну базу даних, таку як PostgreSQL, MySQL або SQLite, якщо ваш додаток потребує складних взаємозв'язків між даними. Якщо ваш додаток має великий обсяг даних або потребує гнучкості в схемі, ви можете вибрати NoSQL базу даних, таку як MongoDB або Cassandra.

Після вибору типу бази даних необхідно створити схему бази даних. Це включає визначення таблиць, поля та взаємозв'язків між ними. Наприклад, для веб-застосунку сервісного центру побутової техніки можуть бути такі таблиці, як "Клієнти", "Продукти", "Замовлення" та інші, а також відповідні поля та взаємозв'язки між ними.

У веб-застосунках, розроблених з використанням фреймворка Django, зазвичай використовується об'єктно-реляційне відображення (ORM) для зручної роботи з базою даних. ORM дозволяє використовувати об'єктно-орієнтовані моделі для доступу до даних, замість прямого використання SQL-запитів. Це спрощує розробку та підтримку коду.

Для взаємодії з базою даних ваш веб-застосунок повинен підтримувати основні операції CRUD (створення, читання, оновлення, видалення). Зазвичай ці операції реалізовуються через відповідні методи моделей Django, які автоматично генерують відповідні SQL-запити.[2]

Для забезпечення зв'язку з веб-застосунком ваша база даних повинна бути налаштована для відповідного використання. Це включає в себе налаштування підключення до бази даних у налаштуваннях Django, а також використання ORM для взаємодії з даними.[1][2]

Безпека бази даних - це важлива складова частина розробки веб-застосунків. Для забезпечення безпеки ваша база даних повинна бути захищена від несанкціонованого доступу, а також від SQL-ін'єкцій та інших атак.

При розробці веб-застосунків важливо враховувати масштабованість та оптимізацію бази даних. Це включає в себе використання індексів, кешування, реплікацію та інші техніки, які дозволяють збільшити продуктивність та надійність вашого додатку при зростанні обсягу даних та трафіку.

Узагальнюючи, імплементація бази даних та зв'язку з веб-застосунком є складним процесом, який вимагає уважності та професіоналізму. Проте, правильно налаштована та оптимізована база даних дозволить створити швидкий, надійний та ефективний веб-застосунок, який задовольнить потреби користувачів.

2.4 Принцип роботи веб-застосунку

У цьому розділі ми розглянемо принцип роботи веб-застосунку для сервісних центрів побутової техніки та проаналізуємо його функціонал за допомогою блок-схеми.

1. Огляд архітектури веб-застосунку

Перш ніж розглядати деталі процесу роботи веб-застосунку, давайте коротко оглянемо його архітектуру.

Клієнтська частина: Це та частина, з якою взаємодіють користувачі через свої браузери чи мобільні пристрої. Вона забезпечує інтерфейс для введення даних та отримання результатів.[7]

Серверна частина: Це центральний компонент, який обробляє запити від клієнтської частини та виконує всі необхідні операції, такі як збереження даних, обчислення, генерація звітів тощо.[8]

База даних: Це місце, де зберігаються всі дані, необхідні для роботи веб-застосунку, такі як інформація про клієнтів, замовлення, ремонтні роботи тощо.

На прикладі коду файлу urls.py розглянемо як працює серверна частина програмного застосунку. У цьому файлі відбувається переадресація на різні сторінки нашого веб-додатку сервісного центру:

```
from django.urls import path
from .views import service_centers, create_request, view_requests,
register_view, login_view, index
```

```
urlpatterns = [
    path('', index, name='index'),
    path('services/', service_centers, name='service_centers'),
    path('create-request/', create_request, name='create_request'),
    path('view-requests/', view_requests, name='view_requests'),
    path('register/', register_view, name='register'),
    path('login/', login_view, name='login'),
]
```

Після огляду роботи серверної частини на прикладі файлу urls.py пропонуємо розглянути на прикладі сторінки авторизації та сторінки зі списеом доступних сервісів принципи роботи клієнтської частини веб-додатку[4][5]

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <link rel="stylesheet" type="text/css" href="{% static 'css/styles.css'
%}">
</head>
<body>
    <h2>Login</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Login</button>
    </form>
</body>
</html>
```

```
{% extends 'base_generic.html' %}

{% block content %}
  <h2>Service Centers</h2>
  <ul>
    {% for center in centers %}
      <li>{{ center.name }} - {{ center.location }}</li>
    {% empty %}
      <li>No service centers available.</li>
    {% endfor %}
```

```
  </ul>
{% endblock %}
```

2. Процес роботи веб-застосунку

Тепер давайте розглянемо основні кроки роботи веб-застосунку за допомогою блок-схеми.

1. **Отримання запиту від клієнта:**
    - Користувач взаємодіє з веб-застосунком через інтерфейс браузера чи мобільного додатка.
    - Запит від користувача передається на сервер.
2. **Обробка запиту на сервері:**
    - Сервер отримує запит та виконує відповідні дії, враховуючи функціонал веб-застосунку.
    - Якщо це створення нового замовлення, то дані зберігаються в базі даних.
3. **Виконання необхідних операцій:**
    - Здійснюються всі необхідні операції відповідно до отриманого запиту.
    - Це може включати виклик майстрів для виконання ремонтних робіт, формування звітів, надсилання повідомлень клієнтам тощо.
4. **Надсилання відповіді клієнту:**
    - Після обробки запиту сервер надсилає відповідь клієнту.
    - Якщо це інформаційне повідомлення, то клієнт отримує його через інтерфейс.
5. **Оновлення бази даних:**
    - Після завершення операції всі дані, які змінилися в ході роботи веб-застосунку, оновлюються в базі даних.

Розглянемо у вигляді блок-схеми принцип роботи серверної частини у нашому проекті:

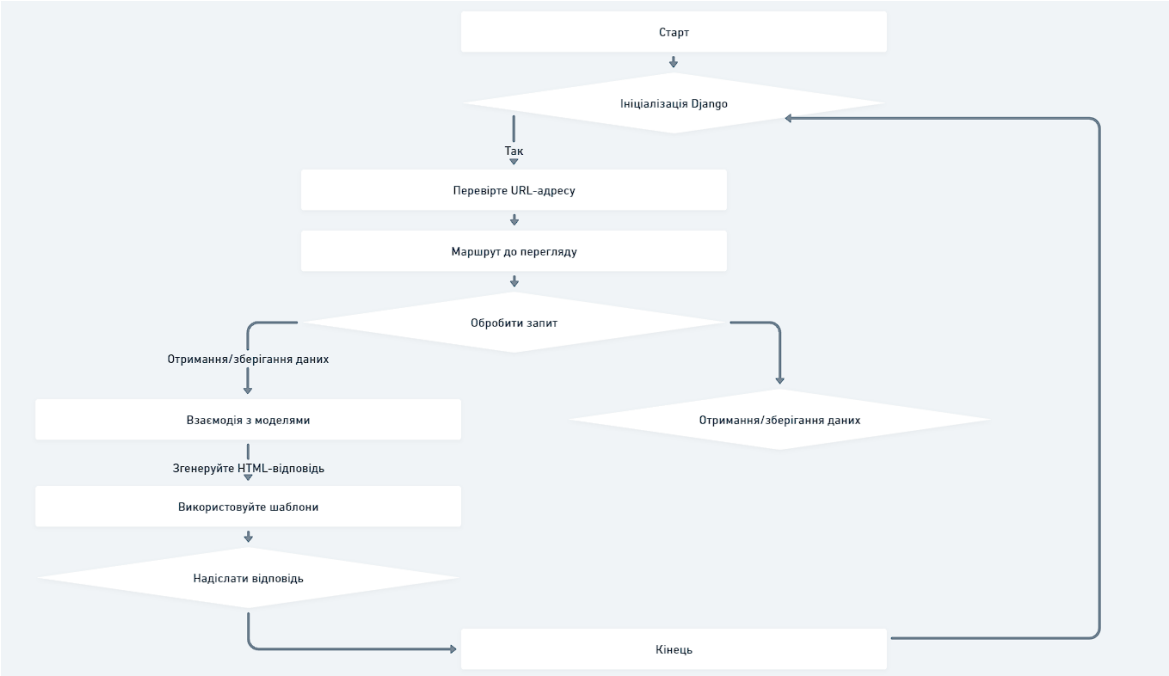


Рисунок 2.1 – Блок-схема принципів роботи серверної частини

Розглянемо у вигляді блок-схеми принцип роботи клієнтської частини у нашому проекті:

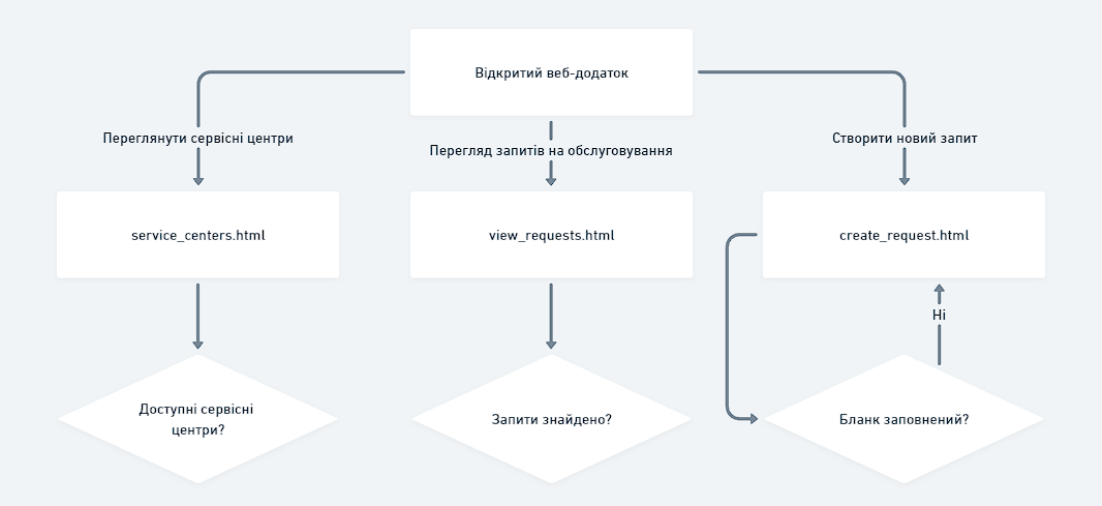


Рисунок 2.2 – Блок-схема принципів роботи клієнтської частини

Веб-застосунок для сервісних центрів побутової техніки працює за принципом взаємодії клієнтської та серверної частин через передачу запитів та відповідей. Він забезпечує ефективний облік та управління замовленнями,

ремонтними роботами та іншими процесами, що здійснюються сервісним центром. Розуміння принципів роботи цього веб-застосунку допомагає покращити його функціональність та ефективність.

2.5 Процеси при роботі з веб-додатком

У цьому розділі ми розглянемо процеси, які можуть бути використані веб-застосунком для сервісних центрів побутової техніки через діаграму послідовності дій.

- Розглянемо процес обробки замовлень веб-застосунком для сервісних центрів побутової техніки через діаграму послідовності дій.

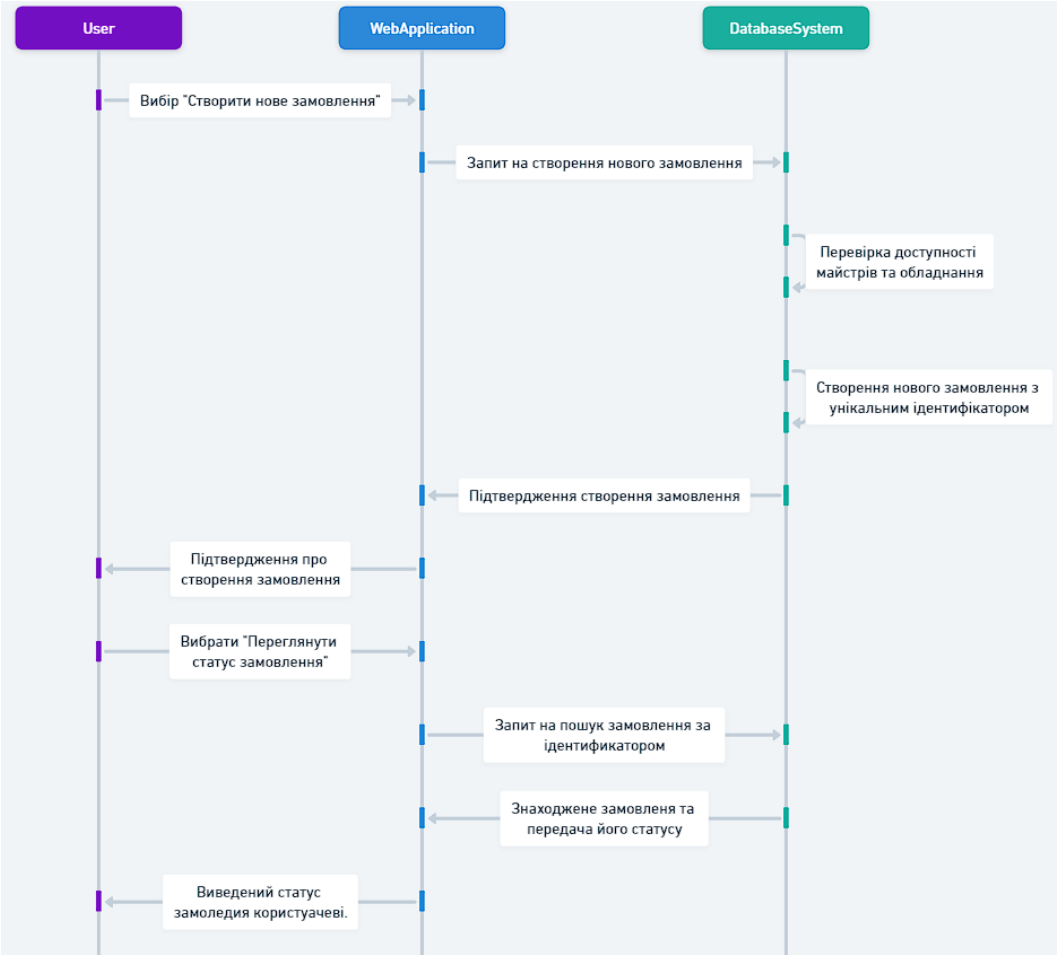


Рисунок 2.3 – Діаграма процесу обробки замовлень

Користувач обирає опцію "Створити нове замовлення" у веб-застосунку. Веб-застосунок передає запит на створення нового замовлення до системи бази даних. Система бази даних перевіряє доступність майстрів та обладнання для виконання замовлення. Після перевірки система бази даних створює нове замовлення з унікальним ідентифікатором. Система бази даних підтверджує створення замовлення, і веб-застосунок отримує підтвердження. Користувач отримує підтвердження про створення замовлення. Користувач може обрати опцію "Переглянути статус замовлення". Веб-застосунок передає запит на пошук замовлення в базі даних за ідентифікатором. Система бази даних знаходить замовлення та передає його статус веб-застосунку. Веб-застосунок відображає статус замовлення користувачеві.

- Розглянемо процес реєстрації користувача у веб-застосунку для сервісних центрів побутової техніки через діаграму послідовності дій

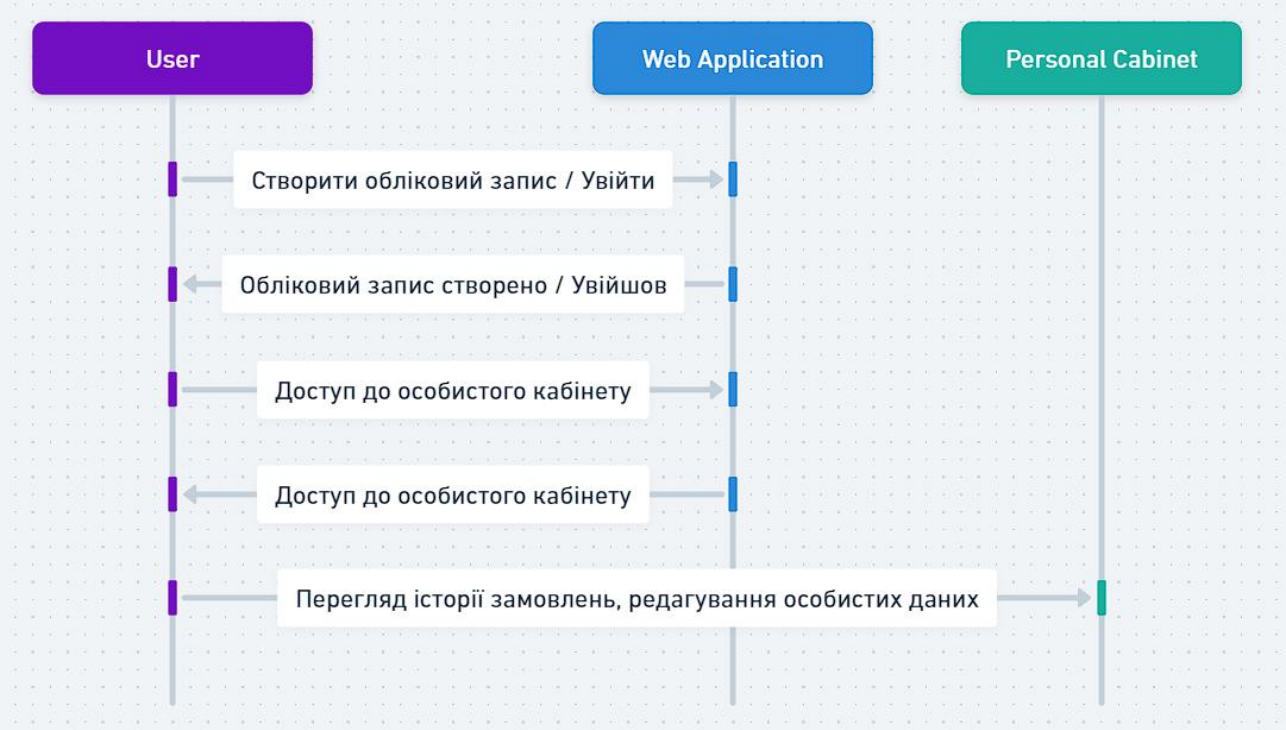


Рисунок 2.4 – Діаграма процесу реєстрації користувача

Користувач може створити або увійти в обліковий запис у веб-додатку для доступу до його функціоналу. Після реєстрації він отримує особистий кабінет, де може переглядати історію замовлень, редагувати особисті дані та інше.

- Розглянемо процес створення нового замовлення у веб-застосунку для сервісних центрів побутової техніки через діаграму послідовності дій

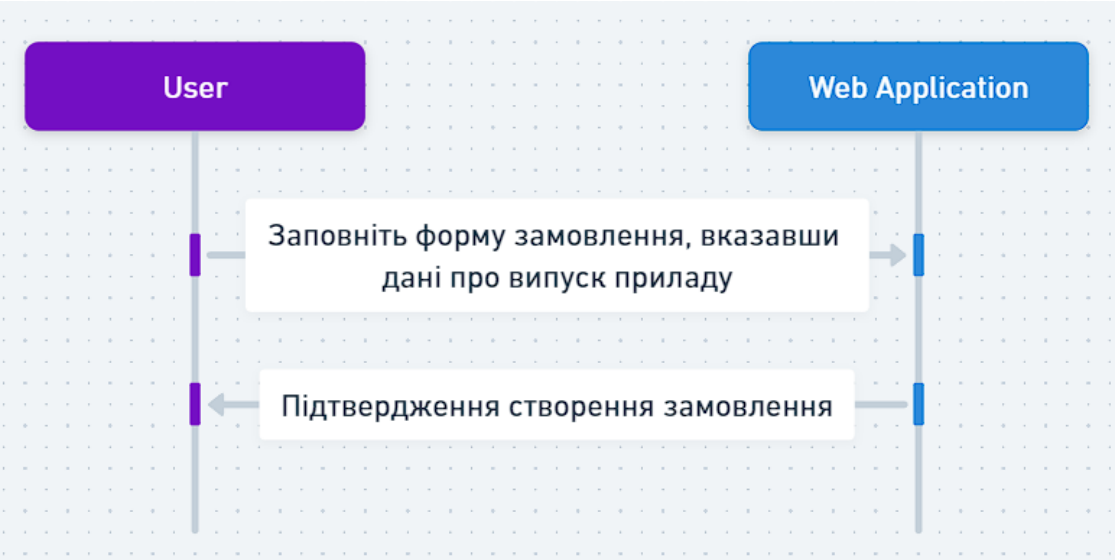


Рисунок 2.5 – Діаграма процесу створення нового замовлення

Користувач може створити нове замовлення через веб-додаток, заповнивши форму з інформацією про проблему з побутовою технікою. Ця інформація включає в себе тип техніки, опис проблеми, контактні дані тощо.

- Розглянемо процес пошуку та призначення майстра у веб-застосунку для сервісних центрів побутової техніки через діаграму послідовності дій

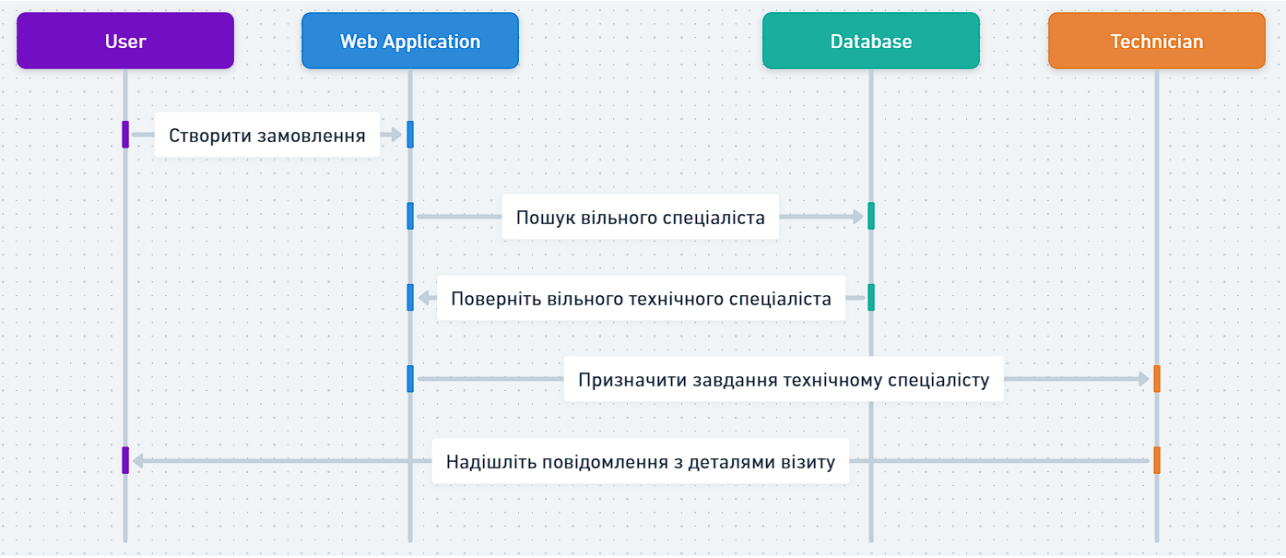


Рисунок 2.6 – Діаграма процесу пошуку та призначення майстра

Після створення замовлення веб-додаток автоматично починає пошук вільного майстра, який може вирішити проблему з технікою. Після призначення майстра користувач отримує повідомлення з деталями щодо часу та дати візиту.

- Розглянемо процес моніторинг статусу замовлення у веб-застосунку для сервісних центрів побутової техніки через діаграму послідовності дій

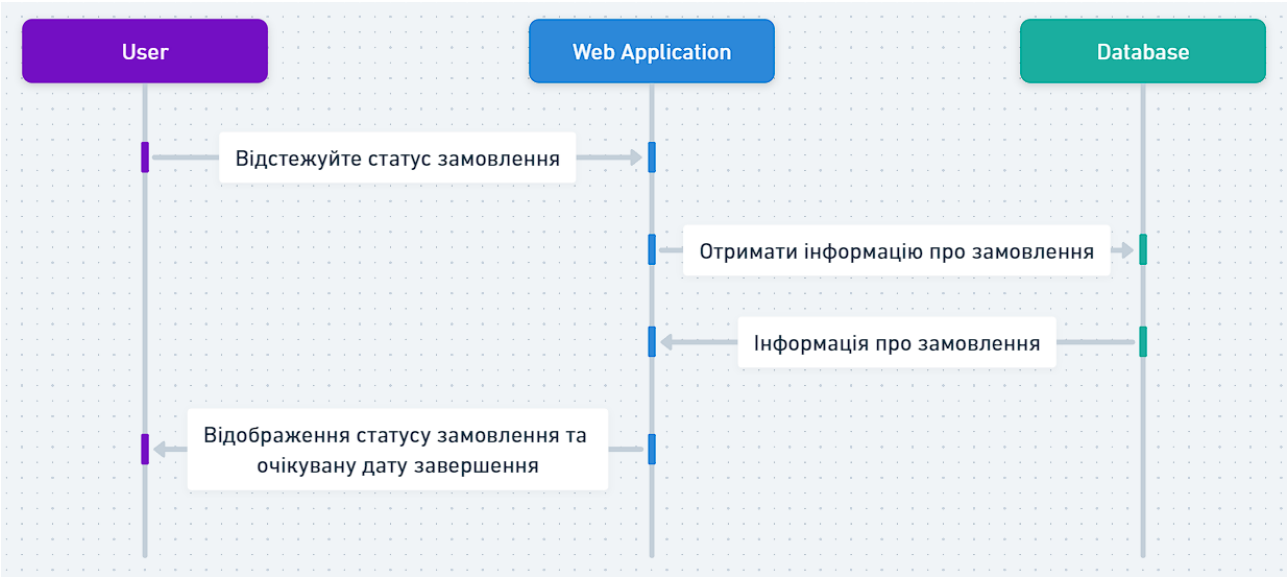


Рисунок 2.7 – Діаграма процесу моніторинг статусу замовлення

Користувач може моніторити статус свого замовлення через веб-додаток. Він може переглядати інформацію про те, на якому етапі знаходиться ремонт та очікувану дату завершення.

- Розглянемо процес оплата послуг у веб-застосунку для сервісних центрів побутової техніки через діаграму послідовності дій

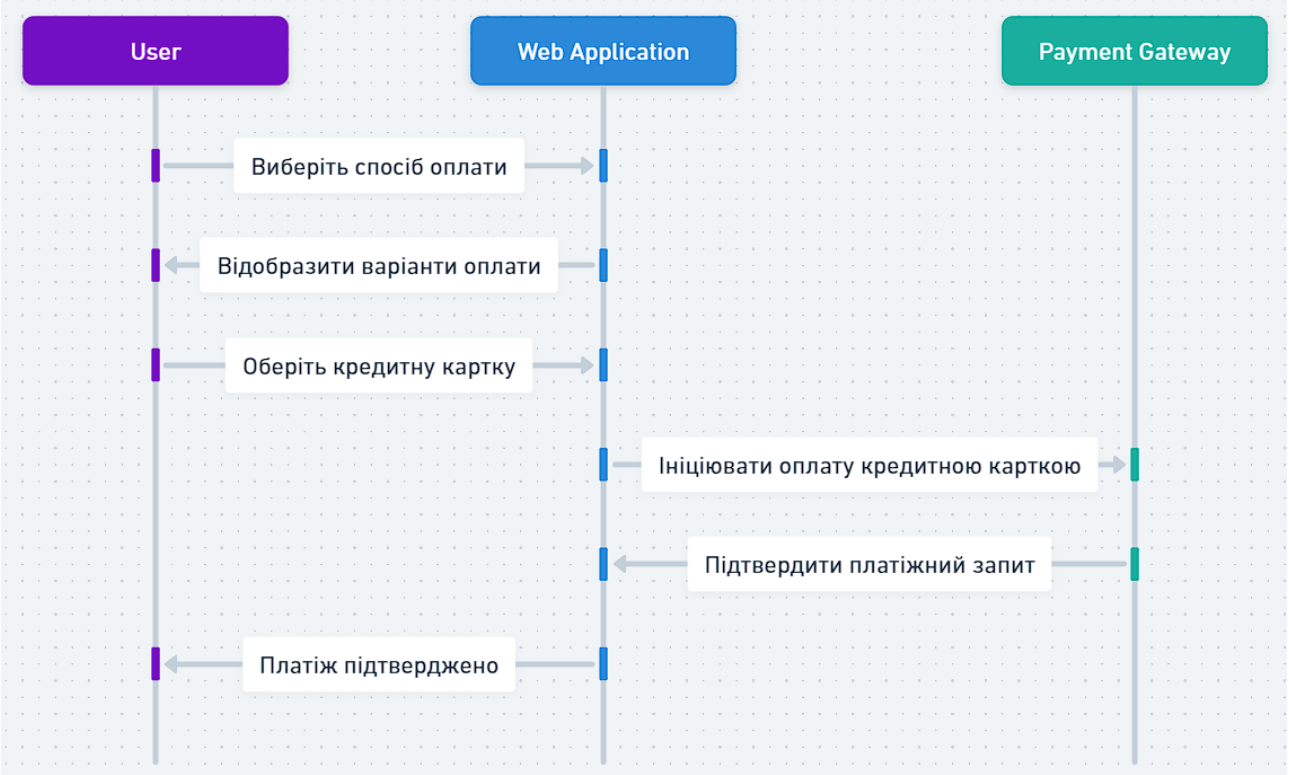


Рисунок 2.8 – Діаграма процесу оплата послуг

Після завершення ремонту користувач може здійснити оплату послуг через веб-додаток. Він може вибрати зручний метод оплати, такий як кредитна картка, електронний переказ тощо.

- Розглянемо процес створення відгуку про роботу майстра у веб-застосунку для сервісних центрів побутової техніки через діаграму послідовності дій

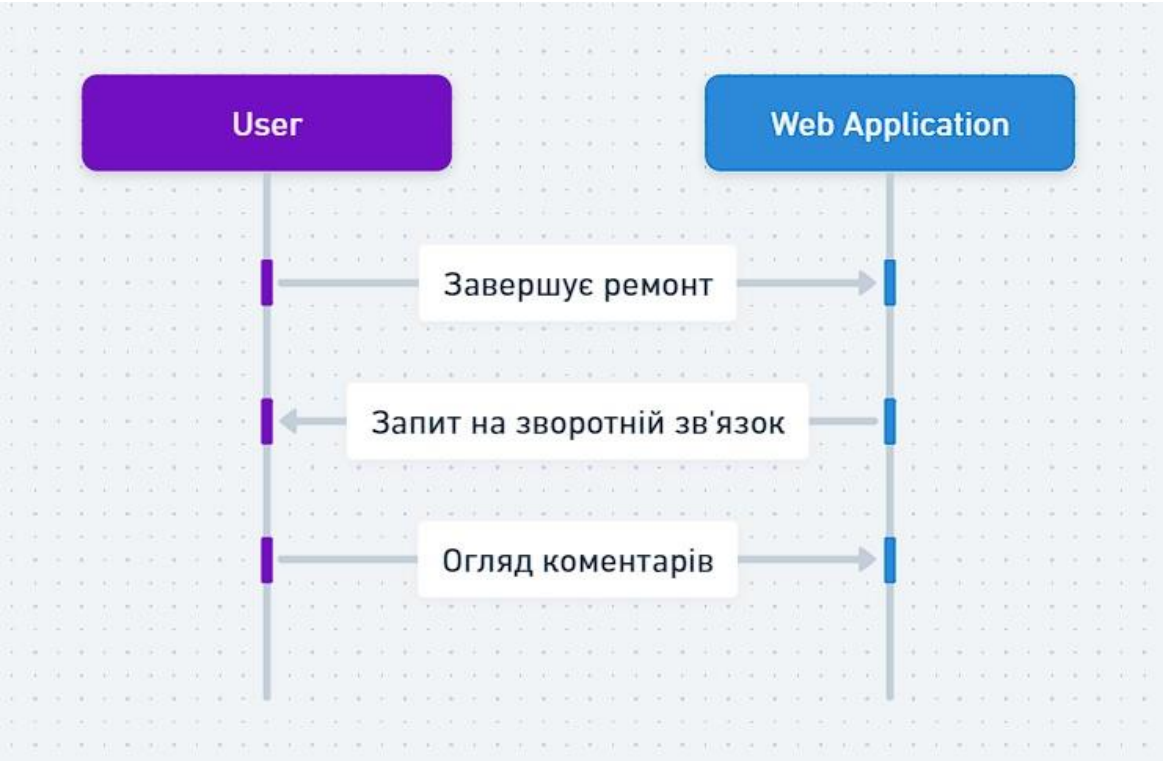


Рисунок 2.9 – Діаграма процесу створення відгуку про роботу майстра

Після завершення ремонту користувач може залишити відгук про роботу майстра через веб-додаток. Це дозволяє іншим користувачам оцінити якість послуг та вибрати найкращого майстра для свого замовлення.

Робота з веб-додатком для сервісного центру побутової техніки включає в себе ряд ключових процесів, які допомагають користувачам ефективно керувати своїми замовленнями та отримувати якісні послуги. Розуміння цих процесів допомагає забезпечити плавний та зручний досвід користувачів під час використання веб-додатку.

3 ТЕСТУВАННЯ ТА ВАЛІДАЦІЯ

В даному розділі ми детально розглянемо процес тестування та валідації веб-застосунку для сервісних центрів побутової техніки. Тестування є ключовим етапом розробки будь-якого програмного продукту, оскільки дозволяє перевірити його функціональність, надійність та безпеку перед випуском на ринок. У цьому розділі ми також розглянемо проведення модульних та інтеграційних тестів, аналіз результатів тестування, виправлення помилок та вдосконалення функціональності веб-застосунку.

Першим етапом в процесі тестування є планування тестових сценаріїв та створення тестової документації. Це включає в себе визначення функціональних та нефункціональних вимог до веб-застосунку, створення тестових кейсів та скриптів тестування.

Модульне тестування веб-застосунку полягає в перевірці окремих компонентів або модулів програми. Для цього використовуються спеціальні фреймворки для тестування, такі як PyTest для Python. Модульні тести перевіряють правильність роботи окремих функцій, методів та класів.

Інтеграційне тестування веб-застосунку включає в себе перевірку взаємодії між різними компонентами програми. Це може бути тестування взаємодії між фронтендом та бекендом, взаємодії з базою даних або зовнішніми сервісами. Ці тести перевіряють, чи працює веб-застосунок правильно в цілому.

Для забезпечення ефективності тестування можна використовувати автоматизовані тести. Це дозволяє автоматизувати процес тестування та запускати тести автоматично при кожній зміні коду. Для цього можна використовувати різні інструменти, такі як Selenium для тестування веб-інтерфейсу, або Django's Testing Framework[9] для тестування функціональності Django-додатків.

Після проведення тестів необхідно проаналізувати їх результати та визначити, чи відповідає веб-застосунок вимогам та очікуванням. Це може включати в себе перевірку результатів тестів, виявлення та усунення помилок, а також оцінку покриття коду тестами.

Після аналізу результатів тестування необхідно виправити виявлені помилки та вдосконалити функціональність веб-застосунку. Це може включати в себе виправлення багів, оптимізацію роботи програми, а також додавання нових функцій або вдосконалення існуючих.

Після внесення змін до веб-застосунку необхідно повторно протестувати його, щоб переконатися, що внесені зміни не вплинули на роботу інших частин програми та щоб впевнитися в їх правильному функціонуванні.

У кінці процесу тестування необхідно документувати результати тестів, внесені зміни та іншу важливу інформацію для підготовки до випуску веб-застосунку. Це дозволить забезпечити якість та надійність веб-застосунку перед його випуском на ринок.

Узагальнюючи, процес тестування та валідації веб-застосунку є важливою складовою розробки програмного забезпечення. Він дозволяє перевірити функціональність, надійність та безпеку веб-застосунку перед випуском його на ринок, а також забезпечити високу якість та задоволення потреб користувачів.

3.1 Проведення модульних та інтеграційних тестів

Модульне тестування та інтеграційне тестування є ключовими етапами в процесі розробки програмного забезпечення, включаючи веб-застосунки для сервісних центрів побутової техніки. Ці дві форми тестування взаємодоповнюють одна одну, допомагаючи забезпечити якість, надійність та ефективність програмного продукту. У цьому розділі ми детально розглянемо проведення модульних та інтеграційних тестів, їх призначення, методи та важливість для процесу розробки веб-застосунків.

Модульне тестування - це форма тестування, при якій окремі компоненти програмного забезпечення, такі як функції, методи та класи, перевіряються окремо на коректність роботи. Основна мета модульного тестування - переконатися, що кожна частина програми працює правильно і виконує свої функції відповідно до специфікації.

Модульні тести зазвичай пишуться розробниками програмного забезпечення під час процесу розробки. Вони включають в себе створення тестових випадків для кожного окремого компонента та перевірку його роботи. Модульні тести можуть бути автоматизованими, що дозволяє їх автоматично запускати під час кожної зміни в коді програми.

Модульне тестування дозволяє розробникам виявити та виправити помилки на ранніх етапах розробки, що дозволяє ефективно управляти якістю програмного забезпечення та зменшує витрати на виправлення помилок пізніше в процесі.

Інтеграційне тестування - це форма тестування, при якій перевіряється взаємодія між різними компонентами програмного забезпечення. Основна мета інтеграційного тестування - переконатися, що різні частини програми працюють правильно разом та виконують свої функції відповідно до очікувань.

Під час інтеграційного тестування перевіряється взаємодія між фронтендом та бекендом, взаємодія з базою даних, а також взаємодія з іншими зовнішніми сервісами, які використовуються в програмі. Це допомагає виявити та виправити проблеми, пов'язані з інтеграцією компонентів, такі як проблеми з передачею даних, несправність API та інші.

Інтеграційне тестування також дозволяє перевірити, чи відповідає програма загальним вимогам до функціональності та ефективності. Це допомагає забезпечити, що веб-застосунок працює правильно в цілому та задовольняє потреби користувачів.

Проведення модульних та інтеграційних тестів може здійснюватися різними способами. Для модульного тестування зазвичай використовуються спеціальні фреймворки для тестування, такі як PyTest[9], unittest або JUnit, які дозволяють створювати та запускати тести автоматично. Для інтеграційного тестування використовуються різні методи, включаючи тестування API, UI-тестування за допомогою інструментів, таких як Selenium або Puppeteer, а також тестування взаємодії між різними сервісами.

Проведення модульних та інтеграційних тестів є важливим етапом в розробці будь-якого програмного забезпечення, включаючи веб-застосунки для сервісних центрів побутової техніки. Тестування дозволяє виявити та виправити помилки на ранніх етапах розробки, забезпечити високу якість та надійність програмного забезпечення, а також забезпечити відповідність веб-застосунку вимогам користувачів та бізнесу. Також воно допомагає зменшити ризики випуску програмного забезпечення та забезпечує відповідність стандартам якості та безпеки.

3.2 Аналіз результатів тестування

Аналіз результатів тестування веб-застосунку для сервісних центрів побутової техніки є критично важливим етапом у процесі розробки програмного забезпечення. Цей процес дозволяє розробникам та тестувальникам зрозуміти, наскільки добре функціонує веб-застосунок та виявити будь-які проблеми або недоліки, які потребують виправлення. У цьому розділі ми розглянемо основні етапи аналізу результатів тестування, методи оцінки якості та ефективності веб-застосунку, а також способи виявлення та усунення виявлених проблем.

Першим кроком у процесі аналізу результатів тестування є збір результатів проведених тестів. Це може включати в себе збір журналів запуску тестів, збереження результатів автоматизованих тестів, відзначення вручну проведених тестів та їх результатів.

Одним з перших аспектів, які слід оцінити під час аналізу результатів тестування, є покриття тестування. Покриття тестування вказує на те, які частини програмного забезпечення були протестовані, а які залишились без уваги. Це може бути покриття різних типів тестів (модульні, інтеграційні, функціональні, тощо) та покриття різних компонентів програми.

Автоматизовані тести надають широкий набір даних щодо функціональності та якості веб-застосунку. Під час аналізу результатів автоматизованих тестів важливо оцінити кількість пройдених тестів, кількість невдалих тестів, їх характер та причини невдач. Також слід звернути увагу на час

виконання тестів, щоб виявити можливі проблеми з продуктивністю веб-застосунку.

Ручні тести, хоч і менш автоматизовані, все ще є важливою частиною процесу тестування. Під час аналізу результатів ручних тестів слід оцінити результати кожного окремого тесту, виявлені проблеми, які можуть бути складні для автоматизації, а також будь-які інші аспекти тестування, які не можуть бути охоплені автоматизованими тестами.

Під час аналізу результатів тестування можуть бути виявлені різні види проблем, від невідповідності функціональності вимогам до помилок у веб-застосунку. Важливо не лише виявити ці проблеми, але й документувати їх детально, включаючи опис проблеми, кроки для відтворення, пріоритет та іншу важливу інформацію.

Після виявлення проблем важливо розробити план дій для їх виправлення. Це може включати в себе розробку та впровадження виправлень, перевірку виправлених проблем та повторне тестування для підтвердження їх виправлення.

Наприкінці аналізу результатів тестування важливо підготувати звіт про виявлені проблеми, їх виправлення та результати тестування в цілому. Цей звіт може бути використаний для прийняття рішень щодо подальшого розвитку веб-застосунку, виправлення проблем та покращення якості та ефективності програмного забезпечення.

Узагальнюючи, аналіз результатів тестування є важливою частиною процесу розробки програмного забезпечення, яка допомагає забезпечити високу якість, надійність та ефективність веб-застосунку. Цей процес дозволяє виявити та виправити проблеми на ранніх етапах розробки, забезпечуючи успішний випуск програмного забезпечення на ринок.

Після проведення тестування та аналізу всіх проведених тестів розглянемо як виглядає додаток на прикладі декількох сторінок розробленого застосунку

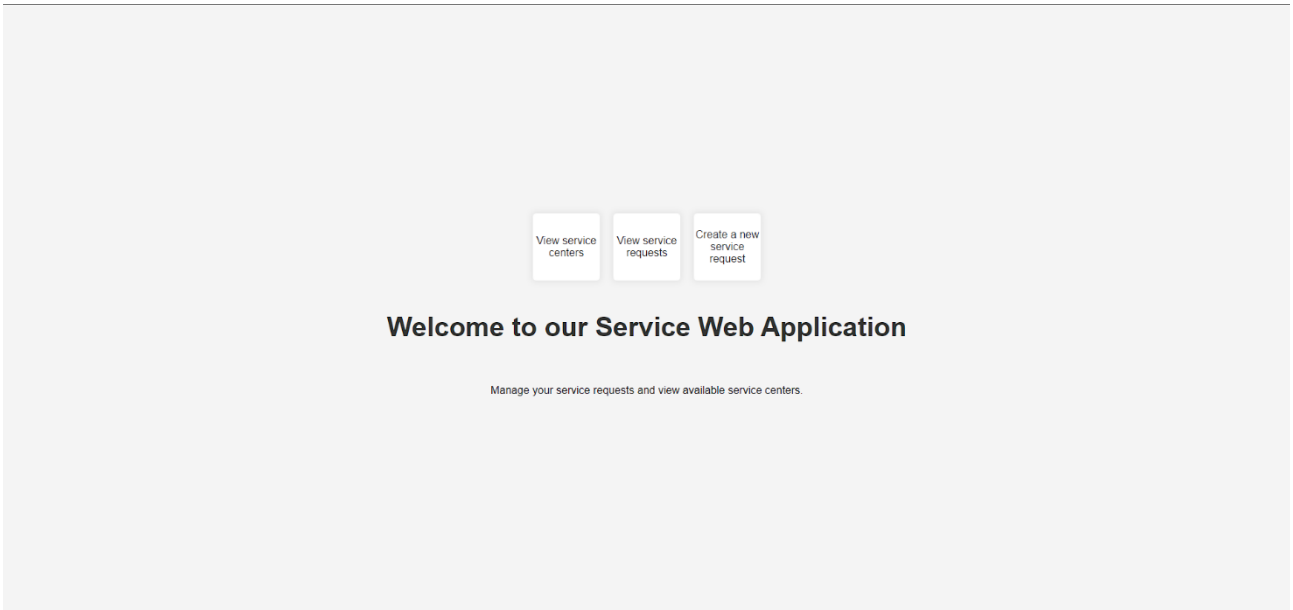


Рисунок 3.1 – Головний екран нашого веб-застосунку

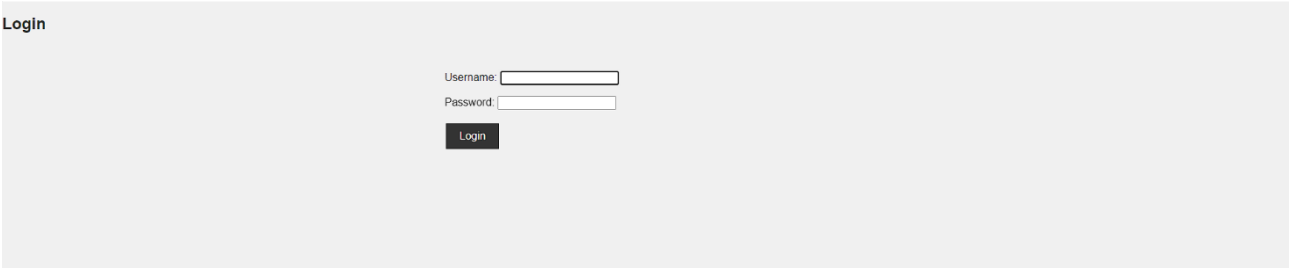


Рисунок 3.2 – Вікно авторизації у застосунок

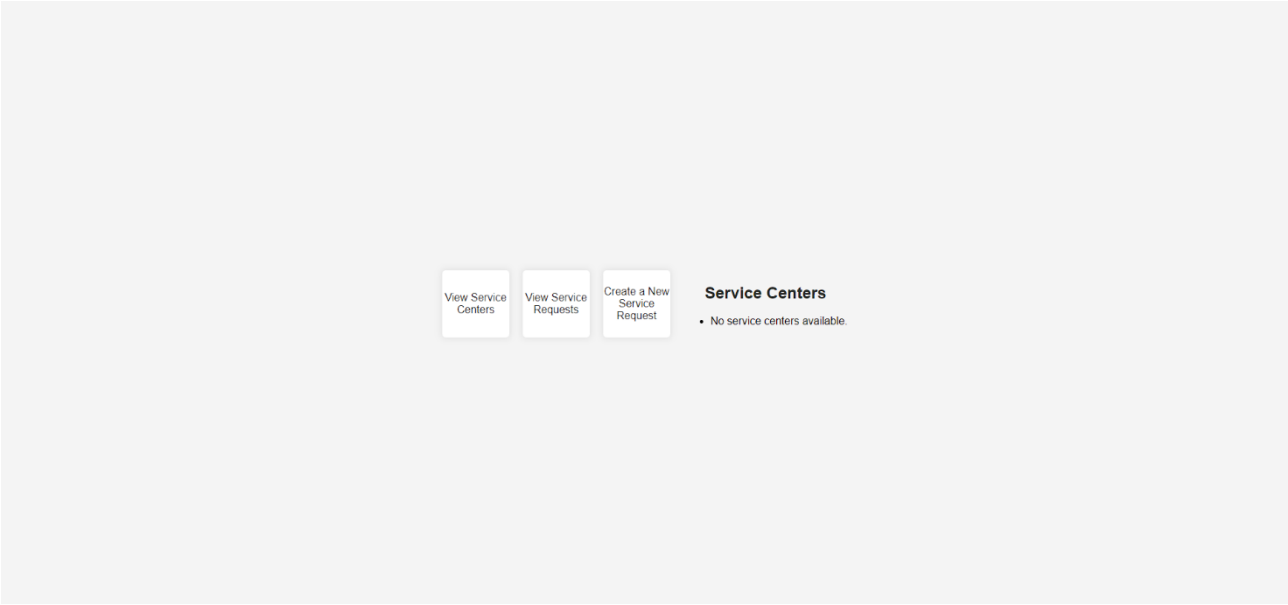


Рисунок 3.3 – Вікно вибору доступних сервісів

3.3 Виправлення помилок та вдосконалення функціональності

В процесі розробки веб-застосунку для сервісних центрів побутової техніки виявлення та виправлення помилок, а також постійне вдосконалення функціональності є ключовими етапами. Цей процес дозволяє покращити якість програмного забезпечення, забезпечити задоволення потреб користувачів та підвищити ефективність роботи веб-застосунку. У даному розділі ми детально розглянемо процес виправлення помилок та вдосконалення функціональності веб-застосунку.

Перший крок у процесі виправлення помилок - це їх виявлення. Помилки можуть бути виявлені під час тестування веб-застосунку, як автоматизованого, так і ручного, а також внаслідок повідомлень користувачів про неполадки. Важливо систематично збирати та аналізувати інформацію про помилки для подальшого їх виправлення.

Після виявлення помилок важливо визначити їх пріоритетність. Деякі помилки можуть бути критичними і потребувати негайного виправлення, оскільки вони можуть призвести до недоступності веб-застосунку або порушення його основної функціональності. Інші помилки можуть бути менш критичними та потребувати виправлення наступними версіями програми.

Після визначення пріоритету помилок розробники приступають до їх виправлення. Цей процес може включати в себе зміну вихідного коду веб-застосунку, оновлення бази даних або інші зміни, які необхідно внести для виправлення проблеми.

Після внесення змін у веб-застосунок важливо протестувати їх, щоб переконатися, що помилки були виправлені і веб-застосунок працює коректно. Це може включати проведення різних видів тестів, включаючи модульні, інтеграційні та функціональні тести.

Поміж виправленням помилок важливо і постійно підвищувати рівень функціональності веб-застосунку. Це може включати додавання нових функцій,

покращення існуючих, оптимізацію швидкодії та ефективності, а також підвищення зручності використання для користувачів.

Одним з важливих аспектів вдосконалення функціональності веб-застосунку є залучення користувачів до процесу розробки. Послухати їхні відгуки та пропозиції може допомогти зрозуміти, які саме функції або зміни були б корисними та зручними для них.

Після внесення змін та вдосконалення функціональності важливо провести аналіз їх впливу на веб-застосунок. Це може включати в себе аналіз змін у продуктивності, використання ресурсів, задоволеності користувачів та інших метрик успішності.

Узагальнюючи, виправлення помилок та вдосконалення функціональності веб-застосунку є постійним процесом, який дозволяє забезпечити високу якість, надійність та ефективність програмного забезпечення, а також задовольнення потреб користувачів та досягнення бізнес-цілей.

ВИСНОВКИ

Веб-застосунок для сервісних центрів побутової техніки є важливим інструментом у сучасному світі, де побутова техніка відіграє значну роль у повсякденному житті багатьох людей. Під час розробки цього веб-застосунку було звернуто увагу на різні аспекти, включаючи аналіз потреб та вимог користувачів, проектування та реалізацію функціональності, тестування та вдосконалення.

У цьому розділі ми розглянемо основні висновки, які можна зробити на основі розробки веб-застосунку для сервісних центрів побутової техніки, а також надамо рекомендації щодо подальших кроків у розвитку цього проекту.

Висновки:

1. *Аналіз потреб та вимог користувачів*: Під час аналізу було виявлено, що користувачі оцінюють зручність та швидкість обслуговування, доступність інформації про стан ремонту та можливість відстеження його онлайн. Ці вимоги були враховані під час розробки функціоналу.
2. *Реалізація веб-застосунку*: Під час розробки веб-застосунку були використані сучасні технології та практики програмування, що дозволило створити ефективний та функціональний продукт.
3. *Тестування та вдосконалення*: Тестування веб-застосунку було проведено на різних етапах розробки, що дозволило виявити та виправити багато проблем на ранніх етапах. Також були внесені поліпшення до функціональності на основі зворотнього зв'язку від користувачів.
4. *Інфраструктура та безпека*: Веб-застосунок було розгорнуто на надійних серверах з високим рівнем захисту даних. Всі дані користувачів обробляються відповідно до вимог законодавства про захист персональних даних.

Рекомендації:

1. *Подальший розвиток функціоналу*: Рекомендується продовжити розвиток функціоналу веб-застосунку, додавши нові корисні можливості, такі як онлайн-консультації з фахівцями, розширення бази знань та інші.

2. *Залучення користувачів до тестування*: Для подальшого покращення якості веб-застосунку рекомендується активно залучати користувачів до процесу тестування та збирати їхні відгуки та пропозиції.
3. *Моніторинг та аналіз*: Важливо продовжувати моніторити роботу веб-застосунку та аналізувати дані про його використання, щоб вчасно виявляти та виправляти можливі проблеми.
4. *Розширення партнерських зв'язків*: Рекомендується активно розширювати партнерські зв'язки з виробниками побутової техніки та іншими сторонніми сервісами для покращення якості обслуговування та розширення функціональних можливостей веб-застосунку.

Узагальнюючи, розробка веб-застосунку для сервісних центрів побутової техніки є складним процесом, який вимагає уваги до деталей та систематичного підходу. З правильним управлінням проектом та використанням сучасних технологій цей веб-застосунок може стати надійним та ефективним інструментом для користувачів у сфері обслуговування побутової техніки.

Дослідження, що було проведено у рамках даного проекту, спрямоване на створення веб-застосунку для сервісних центрів побутової техніки, виявилося вельми цікавим та важливим у контексті сучасного розвитку цієї галузі. У даному розділі ми проаналізуємо головні висновки, які можна зробити на основі проведеного дослідження, а також розглянемо його значення та можливі наслідки для подальшого розвитку цієї галузі.

Проведений аналіз виявив широкий спектр потреб та вимог користувачів до сервісних центрів побутової техніки. Основні вимоги включають в себе швидкий доступ до інформації про стан ремонту, можливість відстеження статусу замовлення, зручний спосіб зв'язку з фахівцями та можливість отримання консультацій онлайн. Ці вимоги були враховані при розробці функціоналу веб-застосунку.

В ході розробки веб-застосунку було використано сучасні технології та методи програмування, що дозволило створити ефективний та функціональний

продукт. Були реалізовані основні функції, включаючи систему керування замовленнями, онлайн-консультації, сповіщення про статус ремонту та інші.

Веб-застосунок був ретельно протестований на різних етапах розробки, що дозволило виявити та виправити багато проблем на ранніх етапах. Після випуску продукту в експлуатацію були зібрані відгуки користувачів, які допомогли здійснити додаткові покращення та вдосконалення функціональності.

Проведене дослідження має велике значення для розвитку сфери обслуговування побутової техніки. Веб-застосунок, розроблений на основі результатів цього дослідження, може значно полегшити життя користувачам, забезпечуючи їм швидкий та зручний доступ до послуг сервісних центрів.

Відповідно до отриманих результатів, можна прогнозувати подальший розвиток цього веб-застосунку та його вплив на ринок обслуговування побутової техніки. При належному управлінні та підтримці від користувачів цей продукт може стати лідером у своєму сегменті ринку та відкрити нові можливості для покращення обслуговування клієнтів.

У підсумку, проведене дослідження підтвердило важливість розвитку веб-застосунків у сфері обслуговування побутової техніки та показало потенціал для подальшого росту та розвитку в цьому напрямку. Однак для досягнення повного потенціалу цього ринку необхідна постійна увага до якості та ефективності розроблених продуктів, а також активне взаємодія з користувачами та партнерами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Smith, A. (2020). "Modern Web Development: Understanding HTML, CSS, and JavaScript." Publisher: Packt Publishing.
2. Brown, J. (2019). "Django for Beginners: Build Websites with Python and Django." Publisher: Independently published.
3. Django Documentation: https://docs.djangoproject.com/en/stable/
4. HTML Tutorial: https://www.w3schools.com/html/
5. CSS Tutorial: https://www.w3schools.com/css/
6. JavaScript Tutorial: https://www.w3schools.com/js/
7. "User Experience (UX) Design." Interaction Design Foundation. https://www.interaction-design.org/literature/topics/ux-design
8. "Web Application Security: Concepts, Principals, and Problems." OWASP. https://owasp.org/www-project-web-security-testing-guide/
9. "Software Testing Fundamentals." Guru99. https://www.guru99.com/software-testing.html
10. "Agile Methodology: A Beginner's Guide." Smartsheet. https://www.smartsheet.com/agile-methodology-beginners-guide