**ДИПЛОМНА РОБОТА**

**ДР.122.042.008.ПЗ**

**КВІТКА НАЗАР ТАРАСОВИЧ**

**2024**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ «Інженерна інфраструктура та комп'ютерні науки»

ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ «КОМП’ЮТЕРНІ НАУКИ»

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до дипломної роботи освітнього ступеня «фаховий молодший бакалавр»

за спеціальністю 122 «Комп’ютерні науки»

(освітня програма 122 «Комп’ютерні науки»)

на тему:

**«Web застосунок підприємства автосервісу»**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студента | 4 | | курсу | | П-42 | групи | | | |
| Квітки Назара Тарасовича | | | | | | | | | |
| (ПІБ) | | | | | | | | | |
| Керівник | | Габрійчук Наталя Ігорівна | | | | | | | |
| Рецензент | | Можаровський Сергій Володимирович | | | | | | | |
|  | | | | | | | | | |
| Національна шкала | | | | |  | | | | |
| Кількість балів | | | |  | | | Оцінка: ECTS | |  |
|  | | | |  | | |  | |  |
| Члени комісії | | | |  | | | | Ісаєв А.М. | |
|  | | | | (підпис) | | | | (прізвище та ініціали) | |
|  | | | |  | | | | Габрійчук Н.І. | |
|  | | | | (підпис) | | | | (прізвище та ініціали) | |
|  | | | |  | | | | Устименко Л.М. | |
|  | | | | (підпис) | | | | (прізвище та ініціали) | |

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ «Інженерна інфраструктура та комп'ютерні науки»

ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ «КОМП’ЮТЕРНІ НАУКИ»

**«ЗАТВЕРДЖУЮ»**

Голова циклової комісії спеціальності

«Комп’ютерні науки»

Сергій МОЖАРОВСЬКИЙ

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_202\_\_ р.

**ЗАВДАННЯ**

**на дипломну роботу**

Здобувач вищої освіти: **КВІТКА Назар Тарасович**

Керівник роботи: **гАБРІЙЧУК Наталя Ігорівна**

Тема роботи: **«Web застосунок підприємства автосервісу»**, затверджена наказом закладу вищої освіти від «27» грудня 2023 р., №532у.

Вихідні дані для роботи:

Об'єкт дослідження: веб-орієнтовані системи для автосервісів.

Предмет дослідження: розробка алгоритму для веб-застосунку, який автоматизує процеси управління автосервісом, зокрема відстеження замовлень клієнтів, управління запасами, планування графіку обслуговування, аналіз типових сценаріїв використання та визначення функціональних вимог, розробка стратегій впровадження веб-застосунків для автосервісу, включаючи планування, ресурси, організаційні аспекти та оцінку ефективності, дослідження правових аспектів розробки та використання веб-застосунків для автосервісу, таких як захист даних, правила конфіденційності, відповідність законодавству тощо.

Аспекти дослідження:

1. Аналіз існуючих веб-орієнтованих систем для автосервісів.

2. Вивчення потреб автосервісів у веб-застосунках та існуючих проблем.

3. Розробка алгоритмів для ефективного управління автосервісом через веб-застосунок.

4. Додаткові функціональні можливості.

5. Проектування та реалізація інтерфейсу веб-застосунку з урахуванням потреб користувачів та специфіки автосервісів.

6. Мультиплатформеність та мобільність: розширення веб-застосунку на мобільні платформи та адаптація веб-застосунку для роботи на мобільних пристроях.

7. Тестування та оцінка продуктивності розробленого веб-застосунку.

Очікуваний результат: розроблений веб-застосунок для автосервісу, що відповідає високим стандартам ергономіки та забезпечує максимальний комфорт для користувачів. Зрозумілий та легкий у користуванні інструмент автоматизує основні процеси, такі як запис на обслуговування за допомогою онлайн-форми зворотного зв'язку, планування графіка роботи, перегляд наявних автозапчастин, переглядати різні види робіт та послуг, які надаються автосервісом та їх приблизна ціна, також можна з легкістю отримати інформацію про кожен вид послуги. Завдяки йому, персонал автосервісу може ефективно керувати замовленнями, а клієнти - отримати швидкий та якісний сервіс без зайвих зусиль.

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Консультант | Завдання  видав | Завдання прийняв |
| 1 | Наталя ГАБРІЙЧУК | 06-02-2024 | 06-02-2024 |
| 2 | Наталя ГАБРІЙЧУК | 29-03-2024 | 29-03-2024 |
| 3 | Наталя ГАБРІЙЧУК | 18-04-2024 | 18-04-2024 |

Консультанти з дипломної роботи із зазначенням розділів, що їх стосується:

**Календарний план**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Етап роботи | Термін виконання | Примітка |
| 1 | Аналіз потреб користувачів | 06 лютого 2024 | Виконано |
| 2 | Вивчення потреб автосервісів у веб-застосунках та існуючих проблем. | 15 лютого 2024 | Виконано |
| 3 | Планування розробки | 27 лютого 2024 | Виконано |
| 4 | Визначення функціональності додатку | 9 березня 2024 | Виконано |
| 5 | Проектування інтерфейсу користувача та дизайну проекту | 16 березня 2024 | Виконано |
| 6 | Створення проекту, налаштування необхідних компонентів та бібліотек | 29 березня 2024 | Виконано |
| 7 | Розробка алгоритмів для ефективного управління автосервісом через веб-застосунок. | 08 квітня 2024 | Виконано |
| 8 | Проектування клієнтської сторони додатку | 18 квітня 2024 | Виконано |
| 9 | Додаткові функціональні можливості | 27 квітня 2024 | Виконано |
| 10 | Розробка мультиплатформеності та мобільності | 08 травня 2024 | Виконано |
| 11 | Тестування та налагодження | 17 травня 2024 | Виконано |
| 12 | Попередній захист роботи | 06 червня 2024 |  |

Здобувач фахової передвищої освіти\_\_\_\_\_\_\_\_\_\_ Назар КВІТКА

Керівник \_\_\_\_\_\_\_\_\_\_ Наталя ГАБРІЙЧУК

**РЕФЕРАТ**

Записка 51 ст., 12 рис., 2 додатки, 16 джерел.

Об'єктом дослідження є процес розробки веб-сайту для автосервісу. Предметом дослідження є методи та технології розробки веб-додатків, зокрема HTML, CSS, JavaScript та інші веб-технології.

Метою дипломної роботи є розробка веб-сайту для автосервісу, який забезпечить зручний доступ до інформації про послуги, ціни, розташування та результати виконаних робіт.

Для досягнення мети роботи використовувалися методи системного аналізу, проектування програмного забезпечення, а також методи тестування та оптимізації веб-додатків. Було проведено аналіз вимог користувачів, створено проектну документацію, реалізовано програмний код та проведено тестування.

Розроблений веб-сайт для автосервісу відповідає поставленим вимогам та завданням. Він забезпечує зручний доступ до інформації та покращує взаємодію з клієнтами. Подальший розвиток проекту може включати додавання нових функцій, інтеграцію з соціальними мережами та мобільними додатками.

ABSTRACT

Note 51 pages, 12 figures, 2 appendices, 16 sources.

The object of research is the process of developing a website for a car service. The subject of research is the methods and technologies of web application development, including HTML, CSS, JavaScript and other web technologies.

The aim of the thesis is to develop a website for a car service, which will provide convenient access to information about services, prices, location and results of the work performed.

To achieve the goal of the work, methods of system analysis, software design, as well as methods of testing and optimizing web applications were used. Analysis of user requirements was carried out, project documentation was created, software code was implemented and testing was carried out.

**ЗМІСТ**

[ВСТУП 6](#_Toc169196598)

[РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ 9](#_Toc169196599)

[1.1. Постановка основної задачі розробки програмного продукту 9](#_Toc169196600)

[1.2. Огляд та порівняння аналогічних систем 12](#_Toc169196601)

[1.3. Вибір та обґрунтування технологій розробки програмного продукту 13](#_Toc169196602)

[РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ 17](#_Toc169196603)

[2.1. Функціональні можливості та діаграма варіантів використання програмного продукту 17](#_Toc169196604)

[2.2. Проектування діаграм активностей, послідовностей та класів 19](#_Toc169196605)

[2.3. Проектування бази даних програного продукту 22](#_Toc169196606)

[2.4. Етапи розробки системи. Розробка алгоритму конвертації програмного продукту 24](#_Toc169196607)

[РОЗДІЛ 3. КЕРІВНИЦТВО ВСТАНОВЛЕННЯ ТА КОРИСТУВАННЯ ПРОГРАМНИМ ПРОДУКТОМ. ОСОБЛИВОСТІ ТА РЕЗУЛЬТАТИ ТЕСТУВАННЯ 31](#_Toc169196608)

[3.1. Керівництво встановлення програмного продукту 31](#_Toc169196609)

[3.2. Керівництво користування програмним продуктом 33](#_Toc169196610)

[3.3. Керівництво користування web-сайтом 35](#_Toc169196611)

[Техніко-економічне обґрунтування 38](#_Toc169196612)

[Охорона праці 41](#_Toc169196613)

[ВИСНОВКИ 44](#_Toc169196614)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46](#_Toc169196615)

[ДОДАТКИ 47](#_Toc169196616)

# ВСТУП

В сучасному світі автомобільна промисловість та послуги автосервісу постійно розвиваються, а це вимагає постійного удосконалення процесів та технологій. Зокрема, зростає попит на веб-застосунки, що спрямовані на полегшення обслуговування автотранспорту та забезпечення зручності для клієнтів.

Об'єктом нашого дослідження є розробка веб-застосунку для автосервісу, що має на меті автоматизацію ключових процесів та полегшення роботи персоналу автосервісу. Отже, об'єктом дослідження є цей комплексний веб-застосунок, який охоплює різноманітні аспекти обслуговування автомобілів та спрямований на оптимізацію робочих процесів та забезпечення зручності для всіх зацікавлених сторін.

Мета даної роботи полягає у розробці ефективного мультиплатформеного веб-застосунку, який забезпечуватиме легкий доступ до інформації про сам автосервіс, його розташування та його послуги, їх вартість та умови надання. Крім того, він має оптимізувати роботу персоналу, автоматизувати процеси прийому замовлень та знизити час їх обробки, що має також забезпечити зручність та ефективність для всіх сторін - від клієнтів до персоналу.

Завдання нашого дослідження включають аналіз сучасних тенденцій у розробці веб-застосунків для автосервісу, визначення вимог користувачів, розробку та тестування прототипу веб-застосунку, проведення оцінки ефективності та користувацького задоволення веб-застосунком після впровадження, аналіз результатів та визначення можливостей для подальшого вдосконалення.

Для досягнення поставлених завдань використовувалися такі методи дослідження, як аналіз літературних джерел, опитування та спостереження користувачів, розробка прототипу та його тестування, проведення оцінки результатів впровадження веб-застосунку, аналіз отриманих даних та визначення можливостей для подальшого вдосконалення та розвитку.

В контексті наукового використання результатів досліджень, отриманих у цій роботі, є можливість їх подальшого застосування в інших галузях, де використовуються веб-застосунки для оптимізації робочих процесів і покращення взаємодії з клієнтами. Результати цього дослідження можуть стати основою для подальших розробок у сфері розвитку веб-технологій для автосервісів та суміжних галузей.

У даній роботі було проведенно аналіз потреб та вимог користувачів, розробка функціоналу та інтерфейсу веб-застосунку, а також в тестування та впровадження цього застосунку у практику автосервісу. Також було зроблено значний внесок у пошук та аналіз наукової літератури з питань веб-розробки та автосервісного бізнесу, що сприяло підготовці даної роботи.

Зазначені вище завдання дослідження, такі як аналіз, проектування, розробка та впровадження веб-застосунку, вирішені з великою уважністю та дбайливістю, з метою досягнення поставленої мети дослідження і вирішення зазначених завдань у відповідності до них.

У вступній частині дипломної роботи визначено актуальність проблеми, обґрунтовано її значення та визначено мету та завдання дослідження. Наукове обґрунтування використання результатів досліджень і визначення особистого внеску також надані.

Структура дипломної роботи складатиметься з таких основних елементів:

1. Вступ: У цьому розділі визначено актуальність проблеми, визначена мета та завдання дослідження, обґрунтовано наукове використання результатів та внесок автора. Також розкрита структура роботи та обґрунтована послідовність її структурних елементів.

2. Теоретична частина: В цьому розділі проводиться огляд літературних джерел, аналіз попередніх досліджень та встановлення теоретичних засад, які допоможуть в розумінні проблеми та вирішенні завдань.

3. Аналіз поточного стану: В даному розділі проводиться аналіз сучасного стану справ у сфері автосервісу та веб-застосунків, виявлення проблемних

моментів та потреб користувачів.

4. Розробка веб-застосунку: У цьому розділі розглядається процес розробки веб-застосунку, включаючи проектування інтерфейсу, вибір технологій та розробку програмного забезпечення.

5. Тестування та впровадження: В даному розділі проводиться тестування розробленого веб-застосунку, оцінка його ефективності та підготовка до впровадження.

6. Висновки та рекомендації: У цьому розділі формулюються висновки на основі отриманих результатів дослідження та надаються рекомендації щодо подальших заходів.

7. Список використаних джерел: У цьому розділі наводяться всі використані в роботі джерела літератури та інформації.

8. Додатки: У даному розділі наводяться додаткові матеріали, які доповнюють основний зміст роботи, такі як код програмного забезпечення, додаткові таблиці, графіки тощо.

Ця структура обрана з метою логічної та послідовної побудови роботи, що дозволить читачеві чітко розуміти основні змістові елементи та результати дослідження.

# РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

**1.1. Постановка основної задачі розробки програмного продукту**

Програмний продукт має на меті створити зручне та ефективне середовище для управління процесом, автоматизацію ключових процесів та полегшення роботи персоналу автосервісу, оптимізації робочих процесів і покращення взаємодії з клієнтами. Його функціональність включатиме можливість легкого доступу користувачів web-додатку до всіх робіт, які може виконати автосервіс, їх ціну, приклади роботи, та його розташування.

Програмний продукт спрямований на забезпечення потреб автовласників у зручному та ефективному способі знаходження та запису на послуги автосервісу. Цільова аудиторія програмного продукту включає:

1. Автовласники:

* Особи, які володіють автомобілями та потребують регулярного технічного обслуговування, ремонту або інших послуг автосервісу.
* Клієнти, які цінують зручність та ефективність у процесі вибору автосервісу та запису на обслуговування.

1. Адміністратори автосервісу:

* Персонал автосервісу, який відповідає за прийом та обробку заявок на обслуговування.
* Менеджери, які відстежують роботу майстерень та розподіляють завдання між працівниками.

1. Майстри автосервісу:

* Фахівці, які здійснюють технічне обслуговування, діагностику та ремонт автомобілів.
* Спеціалісти, які виконують різноманітні роботи з ремонту автотранспорту.

1. Менеджмент автосервісу:

* Власники або керівники автосервісу, які відповідають за стратегічне управління бізнесом та прийняття стратегічних рішень.

1. Інтересовані особи:

* Люди, які шукають інформацію про роботи та послуги, що пропонуються автосервісом.
* Потенційні клієнти, які можуть бути зацікавлені у взаємодії з автосервісом у майбутньому.

Ця аудиторія має різні потреби та очікування від програмного продукту, включаючи зручність використання, доступ до інформації про послуги та ціни, можливість запису на обслуговування онлайн та легкий доступ до контактної інформації автосервісу. Розуміння цих потреб допоможе в розробці продукту, який задовольнить очікування користувачів і забезпечить їм позитивний досвід користування.

Веб-додаток було розроблено з використанням HTML для структури сторінок, CSS для оформлення та стилізації, а також JavaScript для динамічного взаємодії з користувачем і обробки подій. Це забезпечило швидку розробку, покращену ергономіку та зручність використання для кінцевих користувачів. Це допомогло створити додаток з урахуванням крос-платформеності, щоб забезпечити його сумісність з різними веб-браузерами та пристроями різних платформ та діагоналей екрану.

Програмний продукт призначений для автомайстерні, що здійснює ремонт, технічне обслуговування та інші послуги автомобільного сервісу, що дозволяє додатку відображати інформацію про послуги, що надаються, їх ціни, а також змінювати їх відповідно до внутрішньої політики та зміни умов ринку.



Рисунок 1.1.1 Логотип автомайстерні

У процесі створення web-додатку було розроблено логотип з назвою автомайстерні з однойменною назвою «Автогрант», що підвищить впізнаваність автомайстерні і дозволить в майбутньому мати більше коло клієнтів, за допомогою піару та реклами.

Програмний продукт дозволяє відображати інформацію про розташування та контактні дані. Користувачі можуть скористатися програмним продуктом для запису на діагностику або інші послуги онлайн. Програмний продукт забезпечує зручність і простоту процесу запису, а також містить контактну інформацію кожної автомайстерні, щоб користувачі могли зв'язатися з ними для отримання додаткової інформації або уточнень

Успішність програмного продукту може бути виміряна збільшенням кількості клієнтів, які користуються можливістю онлайн запису на діагностику. Це свідчить про зручність та ефективність системи бронювання.

Інформація про доступні послуги та їх ціни може стати більш доступною та привабливою для користувачів.

Однією з ключових метрик успіху є покращення рейтингу задоволеності користувачів програмним продуктом. Це може бути виміряно через зворотний зв'язок від клієнтів, відгуки на соціальних медіа або спеціально проведені опитування.

Успішність програмного продукту також може бути визначена збільшенням обсягу замовлень послуг автомайстерні через онлайн-платформу, підвищення конверсії відвідувачів в клієнти, які здійснюють запис на обслуговування після відвідування веб-сайту автомайстерні. Це свідчить про збільшення впливу програмного продукту на покращення бізнес-процесів.

## 1.2. Огляд та порівняння аналогічних систем

У процесі дослідження аналогічних програмних рішень для автосервісу я виявив ряд недоліків:

1. Відсутність онлайн запису на діагностику автомобіля. Функціонал онлайн-запису на сервіс дозволить клієнтам зручно планувати свій візит до сервісного центру, обираючи зручний для них час і послуги.
2. Відсутність зворотного зв’язку з автосервісом. Можливість отримати консультацію в режимі реального часу через чат або телефон дозволить клієнтам отримати необхідну інформацію швидко та ефективно.
3. Мала кількість інформації про сам автосервіс, його послуги і ціни на них.
4. Відсутність управління запасними частинами: облік наявності та замовлення запасних частин для виконання ремонтних робіт.
5. Інформація про акції та знижки: Публікація актуальних акцій, спеціальних пропозицій та знижок дозволить клієнтам економити гроші та використовувати вигідні пропозиції.
6. Відсутність результатів виконаної роботи. Це може призвести до неприємних ситуацій для клієнтів, які можуть не бути готові до додаткових витрат чи труднощів у процесі обслуговування.
7. Відсутність інформації про розташування автосервісу. Наприклад, не завжди вказується точна адреса або карта проїзду до сервісного центру, що ускладнює процес знаходження та доїзду для клієнтів.

Розроблений додаток інтегрує в собі всі необхідні функції для зручного та ефективного обслуговування автосервісу. Він вирішує низку проблем, які спостерігаються на більшості існуючих сайтів автосервісів, містить у собі повний набір функцій, які роблять процес обслуговування автомобілів максимально зручним та ефективним для клієнтів.

## 1.3. Вибір та обґрунтування технологій розробки програмного продукту

Зважаючи на функціональні та технічні вимоги проекту, використання технологій HTML, CSS і JavaScript має такі переваги:

1. Функціональні вимоги:

* HTML та CSS використовуються для створення зручного й зрозумілого користувацького інтерфейсу, який включає в себе форми для введення даних, елементи керування системою, навігація між блоками системи та механізми і можливості які утворюють єдине ціле..
* JavaScript дозволяє реалізувати динамічний інтерфейс, повний різноманітніх анімацій, валідацію введених даних, а також взаємодію з сервером для обробки запитів без перезавантаження сторінки.
* JavaScript використовується для реалізації запису на діагностику, та зворотнього зв’язку користувача та працівників автосервісу, а також для динамічної зміни контенту на сторінці, наприклад, для відображення різних акцій у залежності від дати або інших умов..
* HTML, CSS і JavaScript використовуються для створення візуального вигляду рекламних блоків та створення веб-сторінок з інформацією про послуги, ціни та контактну інформацію.

1. Технічні вимоги:

* HTML, CSS і JavaScript використовуються для створення добре структурованих і підтримуваних веб-сторінок, що сприяє масштабованості проекту.
* Використання легких та оптимізованих HTML, CSS та JavaScript коду сприяє покращенню швидкодії веб-сторінок.
* HTML, CSS і JavaScript є простими та кросплатформовими технологіями, які працюють на різних пристроях і браузерах. Ці технології дозволяють створювати інтуїтивно зрозумілий та зручний для користувачів інтерфейс за допомогою відповідного візуального оформлення і взаємодії з користувачем.

Оцінка продуктивності та швидкодії використання HTML, CSS і JavaScript порівняно з іншими технологіями може бути обґрунтована наступним чином:

1. Продуктивність:

* HTML, CSS і JavaScript - це технології, що мають простий синтаксис та легкість в освоєнні. Розробка веб-сторінок за допомогою цих технологій дозволить бути швидкою і ефективною, оскільки не потрібно великої кількості коду для досягнення бажаного результату.

1. Здатність оптимально реалізувати функціональність продукту:

* HTML відповідає за структуру сторінок, CSS - за їхній вигляд, а JavaScript - за взаємодію з користувачем. Ця комбінація технологій дозволяє ефективно втілити різноманітні функціональність продукту, включаючи анімацію, валідацію форм, динамічне оновлення контенту і багато іншого.

1. Переваги HTML, CSS і JavaScript:

* HTML - це мова розмітки, яка дозволяє створювати структуру сторінок і визначати їхній зміст.
* CSS - це мова стилів, яка відповідає за зовнішній вигляд сторінок і їхнє оформлення. Вона дозволяє легко керувати виглядом елементів і розміщенням на сторінці.
* JavaScript - це мова програмування, яка використовується для створення інтерактивних елементів на веб-сторінках. Вона дозволяє реалізувати різноманітні функції, такі як анімація, взаємодія з користувачем, обробка подій і багато іншого.

Загалом, використання HTML, CSS і JavaScript дозволяє ефективно втілити функціональність продукту і забезпечити його високу продуктивність та швидкодію.

Для HTML, CSS і JavaScript існує велика спільнота розробників, яка активно підтримує ці технології. Вони широко використовуються в різних веб-проектах різного масштабу - від невеликих особистих веб-сайтів до великих корпоративних додатків.

HTML, CSS і JavaScript є стабільними та надійними технологіями, які вже багато років використовуються в розробці веб-додатків.

Один з ключових аспектів при виборі технологій для розробки програмного продукту - це їхня здатність масштабуватися та підтримуватися у майбутньому. Розглянемо можливості масштабування та підтримки HTML, CSS і JavaScript:

1. Масштабованість проекту: HTML, CSS і JavaScript є відкритими технологіями, які підтримують масштабування проектів різної складності. Завдяки їхній архітектурі та екосистемі можна легко розвивати та розширювати функціональність веб-додатків.
2. Підтримка спільнотою розробників: HTML, CSS і JavaScript мають велику та активну спільноту розробників, яка постійно вносить удосконалення та виправлення у вже існуючі технології. Це забезпечує підтримку та розвиток технологій у майбутньому.
3. Технологічний стек: HTML, CSS і JavaScript є основою веб-розробки та використовуються в поєднанні з різними фреймворками та бібліотеками, що робить їх гнучкими та адаптивними до змін у технологічному стеку.
4. Сумісність та стандартизація: HTML, CSS і JavaScript базуються на міжнародних стандартах, що забезпечує їхню сумісність та стабільність у майбутньому. Це дозволяє забезпечити сумісність з майбутніми версіями браузерів та зберегти працездатність програмного продукту протягом тривалого періоду часу.

Усі ці фактори роблять HTML, CSS і JavaScript надійними та масштабованими технологіями для розробки програмного продукту у майбутньому. Вони дозволяють ефективно розвивати та підтримувати веб-додатки відповідно до зростаючих потреб користувачів та ринкових вимог.

Враховуючи ретельний огляд технічних та функціональних вимог до проекту, можна визначити ряд ключових критеріїв, які впливають на вибір технологій розробки. У даному випадку, вимоги такі як можливість реєстрації користувачів, онлайн запис на діагностику, зворотній зв'язок через чат або телефон, публікація акцій та знижок, керування запасними частинами та повний огляд інформації про послуги і ціни - це фундаментальні функції, які визначають користувацький досвід.

Щодо технічних вимог, важливо врахувати, що система повинна бути масштабованою для обробки великого потоку даних та користувачів. Безпека також є ключовим аспектом, особливо коли маємо справу з конфіденційною інформацією клієнтів. Швидкодія та ефективність взаємодії з користувачами - це також важливі технічні вимоги, які визначають якість та прийняття продукту.

Однак вибір технологій не зупиняється лише на вимогах. Важливо також дослідити застосування обраних технологій в інших проектах. HTML, CSS і JavaScript широко використовуються веб-розробниками по всьому світу, що свідчить про їхню популярність та стабільність. Ці технології є основою веб-розробки та мають велику підтримку спільноти розробників, що забезпечує стале вдосконалення та розвиток.

Оцінюючи можливості масштабування та підтримки обраних технологій у майбутньому, важливо врахувати їхню гнучкість та активність розвитку. HTML, CSS і JavaScript відомі своєю гнучкістю та можливістю легкої інтеграції з іншими технологіями. Крім того, їхні стандарти постійно оновлюються, що дозволяє використовувати найновіші функції та можливості для розробки продукту.

Таким чином, на основі огляду та аналізу вимог, вивчення застосування та оцінки масштабування та підтримки, можна зробити висновок, що використання технологій HTML, CSS і JavaScript для розробки програмного продукту є обґрунтованим та перспективним рішенням.

# РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

## 2.1. Функціональні можливості та діаграма варіантів використання програмного продукту

Функціональні можливості та основні операції:

1. Забезпечення онлайн доступу до послуг автосервісу.
2. Оптимізація взаємодії між клієнтами і сервісним центром.
3. Надання актуальної інформації про послуги та їх вартість.
4. Публікація акцій, спеціальних пропозицій та знижок.
5. Онлайн запис на діагностику автомобіля.
6. Зворотній зв'язок з автосервісом через чат або телефон.
7. Керування запасними частинами.

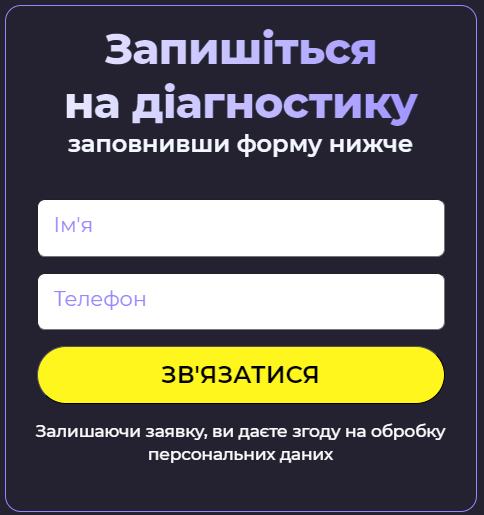


Рисунок 2.1.1 Поле запису для зворотного зв’язку працівників автосервісу з клієнтами

Аналіз вимог:

1. Технічні вимоги: масштабованість, безпека, швидкодія, кросплатформенність, зручний інтерфейс.
2. Функціональні вимоги: реєстрація користувачів, онлайн запис на діагностику, зворотний зв'язок, публікація акцій, управління запасними частинами, огляд інформації про автосервіс.

Основними авторами проекту є різноманітні зацікавлені сторони, які сприяють успішному завершенню розробки програмного продукту. Насамперед, це замовники – власники автосервісу, які визначили необхідність створення інноваційної системи для поліпшення обслуговування клієнтів та управління внутрішніми процесами. Користувачі – клієнти автосервісу, які користуються онлайн записом на послуги, отримують консультації та переглядають актуальні акції та пропозиції. Розробники – команда програмістів, відповідальна за технічну реалізацію проекту, включаючи фахівців з HTML, CSS та JavaScript. Менеджери проекту координують роботу команди, контролюють строки та забезпечують відповідність проекту вимогам замовників. Співробітники автосервісу, такі як адміністратори і менеджери, інтегрують нову систему у свої робочі процеси, забезпечують її належне функціонування та підтримку. Спільно всі ці автори вносять свій вклад у створення ефективного і зручного програмного продукту, який задовольняє потреби як клієнтів, так і співробітників автосервісу.



Рисунок 2.1.2 Діаграма варіантів використання

## 2.2. Проектування діаграм активностей, послідовностей та класів

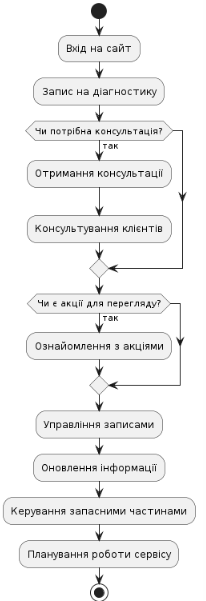
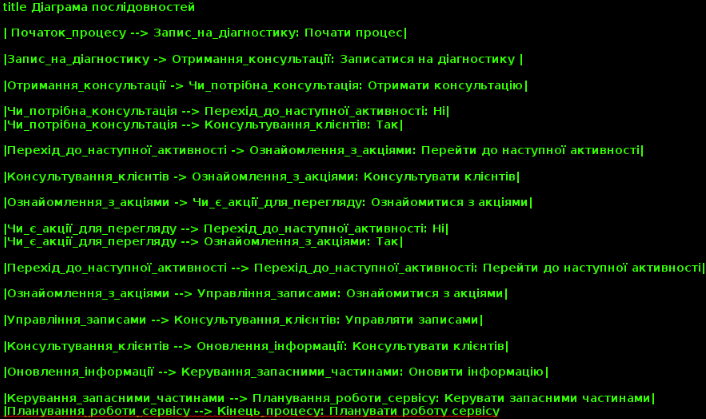


Рисунок 2.2.1 Діаграма активностей

Діаграма активностей дозволяє візуалізувати потік роботи в системі автомайстерні від початку до кінця, включаючи всі важливі активності та рішення, які відбуваються під час обслуговування клієнтів. Це допоможе вам краще зрозуміти процеси, виявити потенційні вузькі місця та оптимізувати роботу системи.

Рисунок 2.2.2 Діаграма послідовностей

Діаграма послідовності, намальована згідно з моделлю PlantUML, ілюструє послідовні етапи взаємодії користувача з системою. Починаючи з ініціації процесу реєстрації для зворотнього зв’язку, користувач крок за кроком просувається через різноманітні функції, такі як запис на діагностику, отримання консультації та перегляд акцій. Ця діаграма відображає логічний порядок операцій і допомагає зрозуміти взаємозв'язок між різними етапами в процесі взаємодії користувача з системою. Вона використана для аналізу та вдосконалення процесів в системі, а також для навчання нових користувачів процедур використання системи.

Створена діаграма класів моделює структуру об'єктно-орієнтованої системи для проекту автомайстерні. У цій діаграмі представлено шість класів (Клас\_1 до Клас\_6), кожен з яких має власні атрибути та методи. Зв'язки між класами вказують на взаємозв'язки та взаємодію між різними частинами системи. Ця діаграма допомагає уявити структуру програми та її компоненти, що дозволяє краще розуміти взаємодію між ними та потреби системи в цілому.

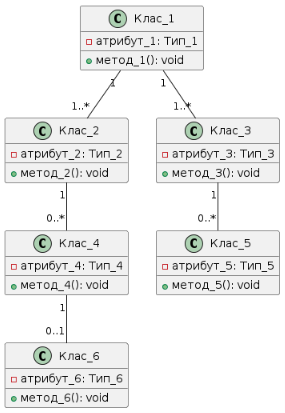


Рисунок 2.2.3 Діаграма класів

Створена діаграма класів моделює структуру об'єктно-орієнтованої системи для проекту автомайстерні. У цій діаграмі представлено шість класів (Клас\_1 до Клас\_6), кожен з яких має власні атрибути та методи. Зв'язки між класами вказують на взаємозв'язки та взаємодію між різними частинами системи. Ця діаграма допомагає уявити структуру програми та її компоненти, що дозволяє краще розуміти взаємодію між ними та потреби системи в цілому.

У процесі розробки програмного продукту для автомайстерні було створено кілька ключових діаграм, які допомагають детально зрозуміти функціональні та структурні аспекти системи:

1. Діаграма варіантів використання (Use Case Diagram)
2. Діаграма активностей (Activity Diagram)
3. Діаграма послідовностей (Sequence Diagram)
4. Діаграма класів (Class Diagram)

## 2.3. Проектування бази даних програного продукту

Проектування бази даних є критичним етапом у розробці програмного продукту, оскільки саме база даних забезпечує зберігання та обробку інформації. У нашому випадку, додаток збирає та зберігає інформацію про користувачів, яку вони надають на сторінці сайту для зворотного зв’язку, а саме їх ім'я та телефон. Нижче наведено кроки, які були пророблені у проектуванні бази даних для цього додатка.

На першому етапі важливо чітко визначити, які саме дані будуть збиратися, оброблятися та зберігатися. У нашому випадку це:

* Ім'я користувача (тип даних: текст)
* Номер телефону (тип даних: текст)

Крім цього, було враховано можливі обмеження та вимоги до валідації даних, такі як формат номера телефону.

На основі визначених вимог створюється модель даних. Основними сутностями будуть користувачі. Структура бази даних може бути візуалізована за допомогою схеми сутність-відношення (ER-діаграма).

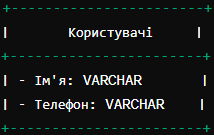


Рисунок 2.3.1 ER-діаграма

На етапі нормалізації база даних перевіряється для зменшення дублювання та підвищення ефективності. У нашому випадку структура вже є досить простою, тому нормалізація включає лише перевірку, що кожен запис є унікальним і немає зайвих повторень.

Встановлення ключів та обмежень є важливим для забезпечення цілісності даних:

* Первинний ключ (Primary Key): Так як база даних проста і складається з одного типу даних, можна додати автоматично згенерований унікальний ідентифікатор для кожного користувача.
* Обмеження: Валідація формату номера телефону.

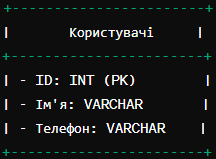


Рисунок 2.3.2 Структура таблиці

Розглядаються можливі запити до бази даних і додаються індекси для прискорення доступу до даних. У нашому випадку можна додати індекс на колонку "Телефон", якщо пошук за телефоном буде частим.

Безпека даних є ключовим аспектом проектування бази даних:

* Управління правами доступу: Встановлення ролей та прав доступу для захисту даних користувачів.
* Шифрування: Використання шифрування для зберігання конфіденційної інформації.
* Резервне копіювання: Розробка стратегії регулярного резервного копіювання даних.

Після завершення проектування бази даних важливо провести тестування на реальних даних. Це включає:

* Тестування продуктивності: Перевірка швидкості виконання запитів.
* Тестування цілісності даних: Переконання, що всі обмеження та правила валідації працюють правильно.
* Оптимізація: Внесення змін на основі результатів тестування для підвищення продуктивності та надійності.

Ці кроки допомогли створити структуровану, ефективну та безпечну базу даних для зберігання інформації про користувачів, яка буде використовуватися в нашому додатку. Правильне проектування бази даних є запорукою успішної роботи програмного продукту та задоволення вимог користувачів.

## 2.4. Етапи розробки системи. Розробка алгоритму конвертації програмного продукту

Метою цього розділу є детальний опис етапів розробки системи та алгоритму конвертації даних. Розробка програмного забезпечення є складним та багатоетапним процесом, який вимагає систематичного підходу для забезпечення якісного та ефективного результату. У цьому розділі буде висвітлено ключові етапи, які були виконані для успішної реалізації системи, а також процес розробки алгоритму конвертації даних.

Процес розробки системи та алгоритму конвертації складається з кількох важливих етапів, кожен з яких має свої специфічні завдання та результати. Нижче наведено ключові етапи цього процесу:

1. Аналіз вимог:

* Збір та аналіз вимог від замовників або користувачів системи.
* Визначення функціональних та нефункціональних вимог.
* Створення специфікації вимог.

1. Проектування:

* Розробка архітектури системи, що включає визначення основних компонентів та їх взаємодію.
* Проектування інтерфейсу користувача, орієнтованого на зручність та інтуїтивність.
* Визначення структури бази даних та розробка ER-діаграми.
* Визначення основних алгоритмів та логіки роботи системи.

1. Розробка:

* Реалізація системи відповідно до проектних специфікацій.
* Написання коду та інтеграція компонентів.
* Використання системи контролю версій для управління кодом.
* Документування процесу розробки та створення технічної документації.

1. Тестування:

* Проведення різних видів тестування (функціонального, інтеграційного, навантажувального).
* Виявлення та виправлення помилок.
* Перевірка системи на відповідність вимогам та стабільність роботи.

1. Впровадження:

* Підготовка середовища для розгортання системи.
* Міграція даних, якщо необхідно.
* Навчання користувачів та забезпечення технічної підтримки.
* Запуск системи в експлуатацію та моніторинг її роботи.

1. Розробка алгоритму конвертації даних:

* Аналіз вихідного та цільового форматів даних.
* Проектування та реалізація алгоритму конвертації, який забезпечить коректне перетворення даних з одного формату в інший.
* Тестування алгоритму на різних наборах даних для перевірки його працездатності.
* Оптимізація алгоритму для підвищення ефективності.
* Інтеграція алгоритму в програмний продукт та забезпечення його підтримки.

Розглядаючи кожен з цих етапів детальніше, було отримано чітке уявлення про те, як структуровано та послідовно розробити програмні системи та алгоритми конвертації даних. Це дозволило не лише досягти високої якості кінцевого продукту, але й забезпечити його надійність та ефективність в умовах реального використання.

Аналіз вимог є одним із найважливіших етапів розробки програмного забезпечення. На цьому етапі відбувається збір, аналіз та узагальнення вимог до системи, що дозволяє визначити її основну функціональність та забезпечити відповідність очікуванням користувачів.

Процес збору вимог включає кілька важливих кроків:

1. Проведення інтерв'ю з замовником або користувачами:

* Основним методом збору вимог є проведення інтерв'ю з замовником або потенційними користувачами системи. Під час інтерв'ю вивчаються потреби та очікування користувачів, обговорюються основні функції та особливості майбутньої системи.
* Важливо задавати питання, що допоможуть розкрити деталі щодо бажаних можливостей, обмежень та пріоритетів.

1. Аналіз документації:

* Додатково до інтерв'ю, необхідно проаналізувати наявну документацію, яка може включати технічні завдання, бізнес-процеси, існуючі системи та їх функціональні можливості. Це допомагає отримати більш детальне розуміння контексту проекту та специфічних вимог.

1. Створення списку функціональних та нефункціональних вимог:

* Після збору інформації необхідно систематизувати отримані дані у вигляді списку вимог. Функціональні вимоги описують конкретні функції та можливості системи, такі як можливість реєстрації користувачів, введення даних тощо. Нефункціональні вимоги включають характеристики, що визначають якість роботи системи, такі як продуктивність, безпека, масштабованість та зручність використання.

1. Документування очікувань користувачів:

* Важливим аспектом аналізу вимог є чітке документування очікувань користувачів. Це включає опис бажаного досвіду використання системи, основних сценаріїв використання та кінцевих цілей, яких користувачі прагнуть досягти за допомогою системи.

1. Визначення критичних вимог:

* Необхідно виділити критичні вимоги, без яких система не може бути визнана успішною. Ці вимоги мають найвищий пріоритет і повинні бути виконані в першу чергу. Критичні вимоги можуть включати основні функції системи, вимоги до безпеки, збереження даних та інші важливі аспекти.

1. Створення специфікації вимог:

* На основі зібраних та проаналізованих даних створюється специфікація вимог, яка є офіційним документом, що описує всі необхідні вимоги до системи. Специфікація повинна бути чіткою, повною та легко зрозумілою для всіх зацікавлених сторін.

1. Розробка діаграм використання (use case diagrams):

* Для візуалізації функціональних вимог до системи використовуються діаграми використання (use case diagrams). Ці діаграми допомагають зобразити взаємодію користувачів із системою, визначити основні сценарії використання та їх взаємозв'язок. Діаграми використання є корисним інструментом для комунікації з замовником та командою розробників, забезпечуючи спільне розуміння функціональних можливостей системи.

Аналіз вимог є основою для успішної розробки програмного забезпечення. Чітке визначення вимог дозволило уникнути непорозумінь та забезпечило відповідність кінцевого продукту очікуванням користувачів. Процес збору, аналізу та документування вимог є критичним для досягнення високої якості та ефективності розроблюваної системи.

Проектування є важливим етапом розробки системи, який включає створення архітектури, проектування інтерфейсу користувача, визначення структури бази даних та розробку внутрішньої логіки системи. Цей етап забезпечує фундамент для подальшої розробки, гарантуючи, що всі компоненти системи працюватимуть узгоджено та ефективно.

Під час створення архітектури системи проводилось визначення компонентів системи та їх взаємодії. На цьому етапі розроблялись архітектурні діаграми, такі як діаграми компонентів UML, які дозволили візуалізувати структуру системи та взаємозв'язки між компонентами.

Розробка інтерфейсу користувача включала створення макетів (wireframes) та прототипів інтерфейсу для визначення розташування елементів та їх функціональності. Також визначаються вимоги до UX/UI, які включають зручність використання та естетичний вигляд інтерфейсу.

Для визначення структури бази даних розроблялась ER-діаграма, яка відображає сутності, їх атрибути та взаємозв'язки між ними. Також визначаються основні операції та алгоритми системи, що включають в себе операції CRUD (створення, читання, оновлення, видалення).

Проектування є важливим етапом розробки системи, який забезпечує чітку та зрозумілу архітектуру, інтуїтивний інтерфейс користувача та ефективну структуру бази даних. Виконання всіх кроків цього етапу дозволило підготувати основу для подальшої реалізації та тестування системи, забезпечуючи високу якість кінцевого продукту.

На етапі розробки системи було написано код та втіленно функціональність, що була розроблена на попередніх етапах. Цей етап включає в себе розподіл завдань, використання систем контролю версій, реалізацію функціональності та документування коду.

Під час реалізації функціональності важливо враховувати вимоги, що були визначені на попередніх етапах, а також дотримуватися принципів чистого коду та кращих практик програмування.

Документування коду є не менш важливим елементом на етапі розробки. Кожен написаний блок коду повинен бути чітко задокументованим для полегшення розуміння його призначення та логіки роботи.

Етап розробки є ключовим у створенні програмного продукту. Забезпечення ефективної роботи команди розробників, використання сучасних інструментів розробки та документування коду дозволяє забезпечити якість та ефективність процесу розробки.

На етапі тестування розроблена система піддається ретельному аналізу та перевірці на відповідність вимогам. План тестування розробляється для визначення стратегії, обсягу та ресурсів, що будуть використовуватися під час тестування. Функціональне тестування спрямоване на перевірку відповідності системи встановленим вимогам та її коректності з точки зору користувача. Інтеграційне тестування дозволяє оцінити взаємодію окремих компонентів системи та їх працездатність у єдиному середовищі. Тестування продуктивності та навантажувальне тестування відображають ефективність системи під час роботи з великим обсягом даних або при великому навантаженні. Усі виявлені під час тестування помилки документуються у спеціальних звітах, після чого проводиться їх аналіз та усунення.

Після успішного завершення тестування система готується до впровадження в реальне середовище. Підготовка середовища для впровадження включає перевірку готовності технічних засобів та програмного забезпечення до роботи з системою. Якщо це необхідно, проводиться міграція даних з попередньої системи. Користувачі отримують навчання щодо роботи з новою системою на спеціальних навчальних сесіях. Після впровадження системи проводиться моніторинг її роботи для виявлення можливих проблем та їх вирішення.

Перший крок - аналіз вихідного та цільового форматів даних, що включає вивчення їх структури та визначення особливостей. Наступним етапом є проектування алгоритму, де розробляється логіка перетворення даних, забезпечення цілісності та структури даних, а також оптимізація швидкості конвертації.

Після проектування алгоритму йде його тестування, в якому створюються тестові набори даних та проводиться перевірка роботи алгоритму як на тестових, так і на реальних даних. Наступним етапом є оптимізація та вдосконалення алгоритму, що включає аналіз результатів тестування та внесення відповідних покращень для підвищення його ефективності.

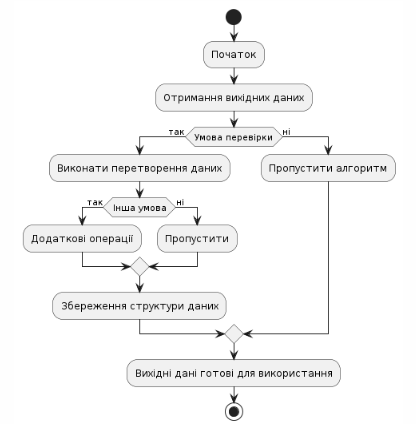


Рисунок 2.4.1 Алгоритм конвертації

У розділі "Етапи розробки системи" було розглянуто ключові етапи в процесі розробки програмного продукту. Розглядаючи кожен етап, від аналізу вимог і проектування до розробки, тестування, впровадження та підтримки, було виявлено важливість кожного з них у забезпеченні успішного завершення проекту.

# РОЗДІЛ 3. КЕРІВНИЦТВО ВСТАНОВЛЕННЯ ТА КОРИСТУВАННЯ ПРОГРАМНИМ ПРОДУКТОМ. ОСОБЛИВОСТІ ТА РЕЗУЛЬТАТИ ТЕСТУВАННЯ

## 3.1. Керівництво встановлення програмного продукту

Програмний продукт "AutoService" - це комплексна система для управління автомайстернею, призначена для автоматизації ключових процесів та покращення ефективності в управлінні автосервісом. Ця система дозволяє автомайстерням ефективно керувати своєю діяльністю, забезпечуючи підвищену продуктивність та зручність у роботі.

"AutoService" - це інтегрована платформа, яка надає повний спектр інструментів для управління автомайстернею. Система включає в себе модулі для управління клієнтами, записів на обслуговування, складського обліку, фінансового обліку та звітності. Основна мета програмного продукту - спростити рутинні процеси автомайстерні, збільшити швидкість обслуговування клієнтів, зменшити помилки та оптимізувати роботу всього підприємства.

Переваги користування додатком для користувачів сайту:

1. Зручний доступ: Користувачі мають зручний доступ до сервісу у будь-який час та з будь-якого пристрою через веб-версію або мобільний додаток.
2. Простота використання: Інтуїтивний інтерфейс додатку дозволяє легко записуватися на обслуговування, переглядати інформацію про послуги та здійснювати оплату.
3. Оповіщення та нагадування: Додаток може надсилати користувачам сповіщення та нагадування про наближені терміни обслуговування або акції, що дозволяє їм завчасно планувати свій візит до автомайстерні.
4. Онлайн-запис: Можливість записуватися на обслуговування онлайн без необхідності дзвінків або відвідування автомайстерні особисто, що забезпечує зручність та ефективність обслуговування.
5. Історія обслуговування: Користувачі мають можливість переглядати історію своїх відвідувань, обслуговування та оплати, що дозволяє їм вести облік своєї автомобільної діяльності.
6. Акції та спеціальні пропозиції: Додаток може повідомляти користувачів про акції, знижки та спеціальні пропозиції, що дозволяє їм економити кошти на обслуговування автомобіля.
7. Зручна оплата: Можливість здійснювати оплату послуг безпосередньо через додаток за допомогою різних способів оплати, таких як кредитні картки або електронні платіжні системи.

Всі ці переваги роблять використання додатку "AutoService Management System" зручним та ефективним для користувачів, допомагаючи їм зекономити час та зусилля при обслуговуванні своїх автомобілів.

Перелік ключових функцій та можливостей:

1. Управління клієнтами: система дозволяє зберігати дані про клієнтів, їх автомобілі та історію обслуговування.
2. Запис на обслуговування: клієнти можуть записатися на потрібний сервіс онлайн або через додаток.
3. Складський облік: ведення обліку запасних частин та інвентарю автомайстерні.
4. Звітність: генерація звітів про продуктивність, фінансовий стан та інші показники діяльності автомайстерні.

AutoService" допомагає автомайстерні оптимізувати свою роботу, забезпечуючи ефективне управління та забезпечення високої якості обслуговування для клієнтів. Всі ці переваги роблять використання додатку "AutoService Management System" зручним та ефективним для користувачів, допомагаючи їм зекономити час та зусилля при обслуговуванні своїх автомобілів.

## 3.2. Керівництво користування програмним продуктом

Керівництво користування програмним продуктом "AutoService Management System" призначене для надання користувачам докладних інструкцій щодо ефективного використання функцій та можливостей програми. Це допоможе користувачам максимально скористатися перевагами системи та забезпечить їх комфортну роботу з програмним продуктом.

"AutoService Management System" - це інтегрована платформа для управління автомайстернею, яка надає повний спектр інструментів для управління клієнтами, записів на обслуговування, складського обліку, фінансового обліку та звітності. Програма допомагає оптимізувати роботу автомайстерні, підвищити ефективність та зручність обслуговування клієнтів. У цьому розділі користувачі знайдуть інформацію про загальне призначення програми та її основні функції, а також важливі посилання або інструкції щодо швидкого старту.

Для кожної функції або операції системи надаються детальні покрокові інструкції. Це включає:

* Реєстрацію клієнтів і запис на обслуговування.
* Управління складом запасних частин та інвентарем.
* Ведення фінансового обліку та створення звітів.
* Керування обліковими записами співробітників та розподіл завдань.

Кожен розділ містить пояснення параметрів та опцій програми, а також зображення, якщо це можливо, для візуального пояснення процесів. Це дозволяє користувачам легко розібратися у функціоналі програми та використовувати її ефективно.

Розділ надає реальні сценарії та приклади, які демонструють використання програми для різних завдань або ситуацій. Наприклад:

* Запис клієнта на діагностику та наступне обслуговування.
* Ведення обліку використаних запчастин та їх поповнення.
* Генерація фінансових звітів для аналізу доходів та витрат автомайстерні.

Кожен приклад ілюструє використання конкретних функцій програми на реальних випадках, що допомагає користувачам краще зрозуміти, як застосовувати програму у своїй повсякденній роботі.

Підтримка та контактна інформація містить інформацію про те, як користувачі можуть звертатися за підтримкою або допомогою у разі виникнення проблем. Надаються посилання на веб-сайт програми, форуми підтримки та контактні дані служби підтримки. Це дозволяє користувачам швидко отримувати необхідну допомогу та вирішувати будь-які питання, що можуть виникнути під час роботи з програмою.

У системі "AutoService Management System" всі оновлення та покращення функціоналу виконуються автоматично на серверному рівні. Це означає, що користувачам не потрібно здійснювати будь-які дії для отримання нових версій програми або її оновлень.

Користувачі завжди працюють із найактуальнішою версією системи, яка автоматично оновлюється командою розробників. Всі нові функції, виправлення помилок та покращення продуктивності стають доступними без необхідності завантажувати або встановлювати оновлення. Їм не потрібно турбуватися про сумісність, встановлення патчів або технічні питання, пов’язані з оновленнями.

Інформація про оновлення системи, нові функції та покращення буде надаватися через регулярні повідомлення на сайті та у розсилці новин. Таким чином, користувачі завжди будуть в курсі нових можливостей, що з’являються у системі.

Користувачам надається інформація щодо налаштувань програми для оптимальної роботи та забезпечення безпеки даних. Рекомендації включають управління правами доступу, налаштування резервного копіювання даних та інші заходи безпеки, що допомагають захистити інформацію користувачів та забезпечити стабільну роботу системи.

Цей розділ є важливим документом, який допомагає користувачам зрозуміти, як правильно використовувати програмний продукт "AutoService Management System" та отримати максимальну користь від його функцій.

## 3.3. Керівництво користування web-сайтом

Тестування програмного продукту "AutoService Management System" є критично важливим етапом розробки, який допомагає виявити можливі проблеми та перевірити працездатність системи перед її випуском для широкого загалу. В рамках цього етапу були виконані наступні види тестування:

1. Функціональне тестування

* Функціональне тестування включало перевірку всіх основних функцій системи, таких як запис на діагностику, отримання консультації, ознайомлення з акціями, управління записами та оновлення інформації. Були проведені тести для переконання, що кожна функція відповідає вимогам специфікації та працює коректно. Наприклад, перевірялася можливість перегляду цін та послуг автосервісу, перегляду розташування сервісу на карті, а також перегляду результатів виконаних робіт.



Рисунок 3.3.1 Приклад перегляду ціни на ремонт КПП

1. Інтеграційне тестування

* Інтеграційне тестування було проведено для перевірки взаємодії між різними компонентами системи. Зокрема, перевірялося, як модуль запису на діагностику взаємодіє з модулем управління записами, а також як інформація про акції відображається на головній сторінці сайту. Тестування забезпечило узгодженість і правильну роботу всіх компонентів у комплексі.

1. Тестування продуктивності

* Тестування продуктивності включало вимірювання швидкості та реакції системи під навантаженням. Було протестовано, як система обробляє одночасні запити на запис на діагностику, консультації та перегляд інформації про послуги. Це дозволило переконатися, що система може працювати ефективно навіть під великим навантаженням.

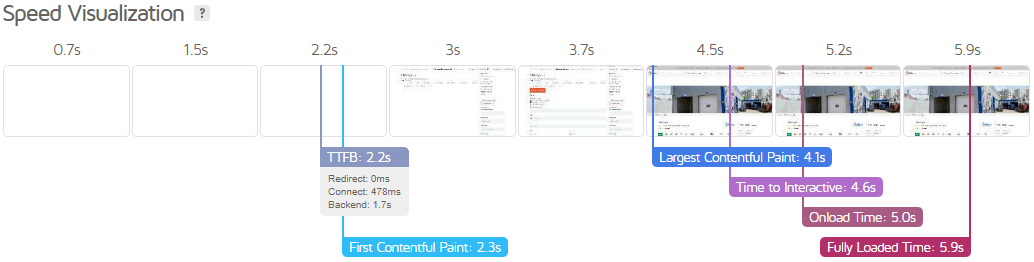


Рисунок 3.3.2 Результат тесту продуктивності

1. Тестування безпеки

* Тестування безпеки було спрямоване на виявлення потенційних уразливостей та їх усунення. Було проведено тестування на проникнення для виявлення можливих загроз безпеці даних користувачів. Наприклад, перевірялися можливості несанкціонованого доступу до даних клієнтів та їх захист під час передачі.

1. Тестування сумісності

* Тестування сумісності включало перевірку роботи системи на різних браузерах та пристроях. Було протестовано сумісність із основними веб-браузерами (Google Chrome, Mozilla Firefox, Microsoft Edge) та різними операційними системами (Windows, macOS, Android, iOS). Це забезпечило правильне відображення та функціонування системи на різних платформах.

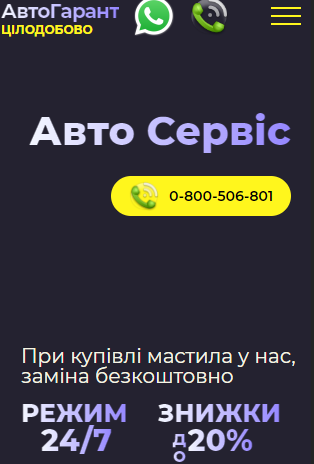


Рисунок 3.3.3 приклад відображення головної сторінки на телефоні

1. Тестування відновлення

* Тестування відновлення перевіряло здатність системи до відновлення після відмов або аварій. Були проведені тести на відновлення даних після збоїв системи та перевірка стабільності роботи після відновлення. Це дозволило переконатися, що система може швидко та ефективно відновлюватися після будь-яких несподіваних ситуацій.

1. Тестування користувацького інтерфейсу

* Тестування користувацького інтерфейсу включало оцінку зручності та інтуїтивності інтерфейсу для кінцевих користувачів. Були проведені тестові сесії за участю реальних користувачів, які перевіряли зручність навігації, зрозумілість інструкцій та доступність основних функцій. Результати цього тестування дозволили внести необхідні покращення в інтерфейс для підвищення зручності використання.

Тестування програмного продукту "AutoService Management System" допомогло забезпечити високу якість та надійність системи перед її випуском. Завдяки ретельному тестуванню всі критичні функції були перевірені, а система виявилася здатною ефективно працювати під різними умовами. Це забезпечує користувачам стабільну та зручну платформу для взаємодії з автосервісом.

# Техніко-економічне обґрунтування

Метою розділу “Техніко-економічне обґрунтування” є аналіз прийнятих технічних і програмних рішень з економічної точки зору. Цей розділ спрямований на обґрунтування доцільності впровадження нового програмного продукту, а також оцінку трудових, матеріальних і фінансових витрат при створенні та впровадженні системи "AutoService Management System".

Впровадження системи "AutoService Management System" є доцільним з огляду на її здатність автоматизувати процеси управління автосервісом, що дозволяє підвищити ефективність роботи та покращити обслуговування клієнтів. Система надає зручний інтерфейс для користувачів, що дозволяє їм швидко та легко отримати необхідну інформацію, записатися на діагностику, переглянути послуги та ціни, а також знайти автосервіс на карті.

Трудові витрати, для розробки системи були залучені наступні фахівці:

* Проектний менеджер (1 особа) – 6 місяців.
* Розробники програмного забезпечення (2 особи) – 6 місяців.
* Тестувальники (1 особа) – 3 місяці.
* Дизайнер інтерфейсу (1 особа) – 2 місяці.
* Системний адміністратор (1 особа) – 2 місяці.

Матеріальні витрати:

* Витрати на апаратне забезпечення (сервери, робочі станції).
* Ліцензії на програмне забезпечення (системи управління базами даних, інструменти розробки).
* Витрати на інтернет та інші комунікаційні послуги.

Фінансові витрати:

* Заробітна плата працівників.
* Витрати на офісне приміщення та комунальні послуги.
* Непередбачені витрати (10% від загального бюджету).

Розрахунок собівартості впровадження системи:

Загальні витрати на розробку:

* Заробітна плата працівників:
* Проектний менеджер: ₴28000 \* 6 місяців = ₴168,000
* Розробники: ₴15000 \* 2 особи \* 6 місяців = ₴180,000
* Тестувальники: ₴45000 \* 3 місяці = ₴135,000
* Дизайнер: ₴14000 \* 2 місяці = ₴28,000
* Системний адміністратор: ₴18000 \* 2 місяці = ₴36,000
* Матеріальні витрати: ₴5,000
* Ліцензії та програмне забезпечення: ₴10,000
* Комунальні та інші послуги: ₴20,000
* Непередбачені витрати: 10% від загального бюджету = ₴58,200

Загальна собівартість впровадження системи: ₴640,200.

Вартість обслуговування системи. Щомісячні витрати на обслуговування включають:

* Заробітна плата системного адміністратора: ₴18,000
* Витрати на оновлення апаратного та програмного забезпечення: ₴35,000
* Комунальні та інші послуги: ₴40,000
* Загальні щомісячні витрати на обслуговування системи: ₴93,000

Очікується, що впровадження системи дозволить зменшити трудові витрати на управління автосервісом на 20%, що в середньому складає ₴78,000 щомісячно. Також система підвищить ефективність обслуговування клієнтів, що може збільшити доходи автосервісу на 10%, що в середньому складає ₴117,000 щомісячно.

Додаткові доходи та зекономлені витрати на місяць: ₴195,000 Час окупності системи: ₴640,200 / ₴195,000 ≈ 3,2 місяців.

Економічний аналіз показав, що впровадження системи "AutoService Management System" є доцільним та ефективним. Система забезпечує автоматизацію багатьох процесів, що дозволяє зменшити трудові витрати та підвищити доходи автосервісу. Час окупності системи складає приблизно 3,2 місяців, що є прийнятним терміном для інвестицій в подібні технологічні рішення. Подальша робота над проектом може включати оптимізацію процесів та розширення функціональності системи, що ще більше підвищить її ефективність та дохідність.

# Охорона праці

Охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, лікувально-профілактичних, реабілітаційних та інших заходів і засобів, спрямованих на збереження життя, здоров'я та працездатності людини в процесі трудової діяльності. В контексті дипломної роботи, присвяченої розробці веб-сайту для управління автосервісом, питання охорони праці охоплюють умови праці працівників автосервісу, зокрема тих, хто працює з комп'ютерною технікою.

Сучасні вимоги охорони праці акцентують увагу на створенні безпечного та здорового робочого середовища, що є особливо важливим у сфері інформаційних технологій, де працівники часто проводять багато годин перед комп'ютером. Впровадження системи управління автосервісом вимагає ретельного аналізу та дотримання вимог охорони праці для забезпечення комфортних та безпечних умов роботи.

В контексті управління автосервісом, основні завдання включають обробку інформації про послуги, клієнтів та інші дані, що вимагає використання комп'ютерної техніки. Основне обладнання, яке використовується для вирішення цих завдань, включає комп'ютери, сервери та інші інформаційно-комунікаційні технології.

Приміщення, в якому розташовані робочі місця працівників, повинно відповідати вимогам виробничої санітарії та гігієни праці. Це передбачає наявність відповідної системи вентиляції, оптимального мікроклімату, низького рівня забрудненості повітря, адекватного освітлення, контролю рівня шуму та забезпечення електробезпеки.

Зокрема, параметри мікроклімату в приміщенні повинні підтримуватися на рівні, який забезпечує комфортні умови праці: температура повітря в межах 18-22°C, відносна вологість 40-60%, швидкість руху повітря не більше 0,2 м/с. Рівень освітленості на робочих місцях з комп'ютерами повинен бути не менше 500 люкс. Для зменшення шуму слід використовувати звукоізоляційні матеріали та пристрої.

Робота з комп'ютерною технікою супроводжується певними шкідливими та небезпечними факторами, такими як іонізаційне випромінювання, електромагнітні поля, тривале навантаження на зоровий апарат та інші. Тому важливо забезпечити правильну організацію робочого місця користувача ПК, що включає зручне розташування монітора, клавіатури та миші, належне освітлення та ергономічне крісло.

Основні заходи техніки безпеки включають:

* Використання моніторів з низьким рівнем випромінювання.
* Регулярні перерви в роботі для зменшення навантаження на очі та хребет.
* Використання спеціальних підставок для ніг.
* Забезпечення належного освітлення робочого місця.

Одним з важливих аспектів охорони праці є забезпечення належного рівня освітленості робочих місць. В умовах автосервісу, де працівники часто працюють з комп'ютерами, освітлення відіграє ключову роль у забезпеченні комфортних умов праці. Розрахунок освітленості можна виконати за формулою:

3.1

де: *E* — необхідна освітленість (люкс),

*F* — світловий потік лампи (люмен),

— коефіцієнт запасу (рекомендується 1.5-1.7),

— коефіцієнт використання світлового потоку,

*S* — площа приміщення (м²),

*Z* — кількість ламп.

Для приміщення площею 20 м² з використанням світлодіодних ламп потужністю 800 люмен та коефіцієнтом використання світлового потоку 0.6, необхідна кількість ламп буде розрахована таким чином:

(3.2)

Для забезпечення освітленості на рівні 500 люкс необхідно:

(3.3)

(3.4)

Отже, необхідно використовувати щонайменше 2 лампи.

Пожежна безпека є важливим аспектом охорони праці, особливо в приміщеннях, де використовується комп'ютерна техніка. Основні небезпечні фактори, що можуть спричинити пожежу, включають короткі замикання, перегрів обладнання та необережне поводження з електроприладами.

Для забезпечення пожежної безпеки необхідно:

* Встановити системи автоматичного пожежогасіння та сигналізації.
* Обладнати приміщення вогнегасниками та іншими засобами пожежогасіння.
* Проводити регулярні інструктажі з пожежної безпеки для працівників.
* Забезпечити вільний доступ до евакуаційних виходів.

Ці заходи дозволять швидко виявити та знешкодити вогонь, а також забезпечити безпеку працівників у разі пожежі.

Розробка та впровадження системи управління автосервісом потребують ретельного врахування вимог охорони праці. Це включає створення комфортних та безпечних умов для працівників, особливо тих, хто працює з комп'ютерною технікою. Виконані розрахунки та запропоновані заходи допоможуть забезпечити належний рівень охорони праці, що сприятиме ефективній та безпечній роботі всіх співробітників автосервісу.

# ВИСНОВКИ

Результати розробки системи "AutoService Management System" демонструють успішне досягнення поставлених цілей та вимог. В ході виконання дипломної роботи була створена та впроваджена система, яка забезпечує автоматизацію процесів управління автосервісом, покращуючи ефективність роботи і надаючи зручний інтерфейс для користувачів.

Основними досягненнями проекту є:

1. Розробка функціонального веб-сайту, який дозволяє користувачам записуватись на діагностику, отримувати консультації, переглядати інформацію про послуги та ціни, а також ознайомлюватись з акціями автосервісу. Система надає можливість переглядати результати виконаних робіт та розташування автосервісу на карті.
2. Використання сучасних технологій для забезпечення стабільної роботи системи, зокрема, застосування технологій для зберігання та обробки даних, забезпечення безпеки та захисту даних користувачів.
3. Проведення комплексного тестування системи, яке включало функціональне, інтеграційне, продуктивне, безпекове та інші види тестування. Це дозволило виявити та усунути критичні помилки, забезпечивши високу якість та надійність програмного продукту.

Основні показники, отримані при розробці проекту:

1. Покращення ефективності роботи автосервісу за рахунок автоматизації процесів запису на діагностику та консультації.
2. Забезпечення зручного доступу до інформації про послуги та ціни автосервісу для користувачів.
3. Зниження часу на обробку записів та запитів від клієнтів завдяки інтегрованій системі управління записами.
4. Підвищення задоволеності клієнтів через зручний та інтуїтивно зрозумілий інтерфейс користувача.

Подальша робота над проектом може включати наступні напрямки:

1. Розширення функціональності системи шляхом додавання нових модулів, таких як управління запасами запчастин, ведення історії обслуговування автомобілів та інтеграція з платіжними системами для онлайн-оплати послуг.
2. Покращення користувацького інтерфейсу, щоб зробити його ще більш зручним та привабливим для користувачів, зокрема, за рахунок адаптації для мобільних пристроїв.
3. Розширення можливостей аналітики та звітності для надання керівництву автосервісу детальних звітів про ефективність роботи, завантаженість сервісу та інші важливі показники.
4. Інтеграція з CRM-системами для поліпшення управління взаємодією з клієнтами та маркетинговими кампаніями.

Подальший розвиток системи є доцільним, оскільки він дозволить підвищити конкурентоспроможність автосервісу, забезпечити більш якісне обслуговування клієнтів та оптимізувати внутрішні процеси. Впровадження нових функцій та можливостей сприятиме залученню нових клієнтів і підвищенню лояльності існуючих.

Таким чином, результати дипломної роботи свідчать про успішне виконання поставлених задач та досягнення запланованих цілей. Подальший розвиток системи "AutoService Management System" обіцяє значні перспективи для покращення обслуговування клієнтів та підвищення ефективності роботи автосервісу.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білик, С. І. (2016). Веб-програмування. Частина 1: HTML і CSS. Навчальний посібник. Харків: ХНУРЕ.
2. Wilcox J.C. A new algorithm for strapped-down inertial navigation / J.C. Wilcox // IEEE Trans. Aerospace and Electr. Syst., – 1967. – Vol.3. No 5. – P. 796–802.
3. Git Documentation. (2023). Retrieved from <https://git-scm.com/doc>
4. Eloquent JavaScript. (2023). Retrieved from <https://eloquentjavascript.net/>
5. Docker Documentation. (2023). Retrieved from <https://docs.docker.com/>
6. W3C. (2023). HTML5. Retrieved from <https://www.w3.org/TR/html52/>
7. Мельник, М. А. (2017). Основи веб-технологій: HTML, CSS, JavaScript. Львів: ЛНУ імені Івана Франка.
8. W3C. (2023). CSS: Cascading Style Sheets. Retrieved from <https://www.w3.org/Style/CSS/>
9. Robbins, J. N. (2018). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (5th Edition). O'Reilly Media.
10. Meyer, E. A. (2017). CSS: The Definitive Guide (4th Edition). O'Reilly Media.
11. Скляр, О. М., & Литвин, О. М. (2018). Основи веб-програмування: HTML, CSS, JavaScript. Львів: Видавництво Львівської політехніки.
12. Дзьоба, О. В. (2019). Створення та розвиток веб-сайтів: HTML, CSS, JavaScript, CMS. Київ: КПІ імені Ігоря Сікорського.
13. Гудима, А. О., & Шевчук, В. В. (2020). Розробка веб-додатків: HTML, CSS, JavaScript, PHP. Київ: КНУ імені Тараса Шевченка.
14. MDN Web Docs. (2023). HTML: HyperText Markup Language. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTML>
15. MDN Web Docs. (2023). CSS: Cascading Style Sheets. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/CSS>
16. Lawson, B., & Sharp, R. (2011). Introducing HTML5 (2nd Edition). New Riders.

# ДОДАТКИ

**Додаток А**

**Частина коду головної сторінки проекту з файлу index.html:**

<header class="header">

  <div class="container header\_\_container">

    <div class="header\_\_phone-box">

      <div class="header\_\_inner">

        <a href="#" class="header\_\_logo">АвтоГарант</a>

        <a href="https://api.whatsapp.com/send/?phone=0-800-506-801" class="header\_\_sotial"><img src="img/whatsapp.png" alt="Ватсап"></a>

        <a href="tel:0-800-506-801" class="header\_\_sotial"><img src="img/phone.png" alt="Телеграмм"></a>

      </div>

      <p class="header\_\_text">ЦІЛОДОБОВО</p>

    </div>

        <div class="header\_\_menu menu">

            <div class="menu\_\_icon">

                <span></span>

                <span></span>

                <span></span>

            </div>

            <nav class="menu\_\_body">

                <ul class="menu\_\_list">

                    <li class="menu\_\_item"><a href="index.html" class="menu\_\_link">Головна</a></li>

                    <li class="menu\_\_item"><a href="painting.html" class="menu\_\_link">Фарбування</a></li>

          <li class="menu\_\_item"><a href="price.html" class="menu\_\_link">Ціни</a></li>

                    <li class="menu\_\_item"><a href="#contacts" class="menu\_\_link">Контакти</a></li>

                </ul>

            </nav>

        </div>

  </div>

</header>

**Додаток Б**

**Частина коду з файлу main.js**

document.addEventListener("DOMContentLoaded", () => {

          const accordions = document.querySelectorAll(".accordion");

          accordions.forEach((el) => {

            el.addEventListener("click", (e) => {

              const self = e.currentTarget;

              const control = self.querySelector(".accordion\_\_control");

              const content = self.querySelector(".accordion\_\_content");

              self.classList.toggle("open");

              // если открыт аккордеон

              if (self.classList.contains("open")) {

                control.setAttribute("aria-expanded", true);

                content.setAttribute("aria-hidden", false);

                content.style.maxHeight = content.scrollHeight + "px";

              } else {

                control.setAttribute("aria-expanded", false);

                content.setAttribute("aria-hidden", true);

                content.style.maxHeight = null;}

            });

          });

        });

const menu = document.querySelector(".menu\_\_body");

        const menuBtn = document.querySelector(".menu\_\_icon");

        const body = document.body;

        if (menu && menuBtn) {

          menuBtn.addEventListener("click", (e) => {

            menu.classList.toggle("active");

            menuBtn.classList.toggle("active");

            body.classList.toggle("lock");

          });

          menu.addEventListener("click", (e) => {

            if (e.target.classList.contains("menu\_\_body")) {

              menu.classList.remove("active");

              menuBtn.classList.remove("active");

              body.classList.remove("lock");}

          });

          menu.querySelectorAll(".menu\_\_link").forEach((link) => {

            link.addEventListener("click", () => {

              menu.classList.remove("active");

              menuBtn.classList.remove("active");

              body.classList.remove("lock");});

          });

        }

const anchors = document.querySelectorAll('a[href\*="#"]');

        anchors.forEach((anchor) => {

          anchor.addEventListener("click", (event) => {

            event.preventDefault();

            const blockID = anchor.getAttribute("href").substring(1);

            document.getElementById(blockID).scrollIntoView({

              behavior: "smooth",

              block: "start",});

          });

        });