

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

Бакалаврська кваліфікаційна робота
на тему:
«РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОКАТУ
ВЕЛОСИПЕДІВ»

Виконав: студент 4 курсу, групи 2АКІТ-216
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології

 Олександр КАБАШНИЙ
Керівник: к.т.н., доцент каф. КСУ

 Тетяна ГРИЩУК

«06» червня 2025 р.

Рецензент: к.т.н., доцент каф. АІТ

 Роман МАСЛІЙ

«06» червня 2025 р.

Допущено до захисту
Зав. кафедри КСУ

 В'ячеслав КОВТУН
«16» червня 2025р.

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 15 – Автоматизація та приладобудування
Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітньо - професійна програма – Інтелектуальні комп'ютерні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ
Д.т.н., проф.



В'ячеслав КОВТУН
“23” травня 2025 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
студенту Кабашному Олександрю Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуальної комп'ютерної системи прокату велосипедів
керівник роботи к.т.н., доц. каф. КСУ Гришук Т.В.
затверджені наказом ВНТУ № 97 від 20.03.2025 р.
2. Термін подання студентом роботи 10 червня 2025 року
3. Вихідні дані до роботи: розробити Інтернет-систему підтримки процесу оренди велосипедів з використанням засобів машинного навчання для визначення ймовірності відмови транспортного засобу та передбачення попиту оренди; основні джерела:
1 В.Б.Мокін, М.В.Дратований. Наука про дані: машинне навчання та інтелектуальний аналіз даних - Електронний навчальний посібник /Вінниця : ВНТУ, 2024. – 263 с.
2 Гришук Т. В., Юркіна О. М. Комп'ютерна програма «Маршрути громадського транспорту» : авторське свідоцтво № 133380. – Зареєстр. в Держ. реєстрі комп'ютерних програм 11.02.2025 бюлетень № 87 від 31.03.2025.
4. Зміст текстової частини: аналіз предметної області; проектування автоматизованої системи; розробка моделей машинного навчання; тестування системи; висновки
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
UML- діаграма варіантів використання системи, UML-діаграма діяльності, UML-діаграма послідовності, ER-модель бази даних, модель розрахунку ймовірності відмови велосипеда, модель прогнозу попиту оренди велосипедів, результати експерименту

1. Консультанти розділів роботи

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

2. Дата видачі завдання 27.05.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Вибір, узгодження та затвердження теми БКР	06.05.2025	08.05.2025	
2	Постановка задачі дослідження	17.05.2025	19.05.2025	
3	Аналіз об'єкта автоматизації, робота з літературою	20.05.2025	26.05.2025	
4	Розробка та налаштування програмного забезпечення, підбір архітектури та інструментів розробки	27.05.2025	03.06.2025	
5	Оформлення пояснювальної записки, графічного матеріалу і презентації	04.06.2025	08.06.2025	
6	Попередній захист БКР, доопрацювання, рецензування БКР	09.06.2025	10.06.2025	
7	Захист БКР	17.06.2025	17.06.2025	

Студент  Олександр КАБАШНИЙ
(підпис) (ім'я і ПРІЗВИЩЕ)

Керівник роботи  Тетяна ГРИЩУК
(підпис) (ім'я і ПРІЗВИЩЕ)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 109 сторінок формату А4, на яких є 44 рисунки, список використаних джерел містить 35 найменувань. Мета роботи — розробити інтелектуальну комп'ютерну систему для автоматизації процесу прокату велосипедів із використанням методів штучного інтелекту для прогнозування технічного стану велосипедів, прогнозування попиту на оренду та оптимізації обслуговування.

У теоретико-методичній частині роботи було детально розглянуто та проаналізовано предметну область та об'єкт розробки інтелектуальної комп'ютерної системи прокату велосипедів. Було висвітлено основні принципи функціонування систем прокату, можливості застосування штучного інтелекту для прогнозування технічного стану велосипедів і попиту на оренду, а також визначено критерії ефективності роботи системи. Детально розглянуто переваги та недоліки існуючих рішень у сфері прокату транспорту. У практичній частині була розроблена та протестована система для забезпечення автоматизації процесів реєстрації, вибору та оренди велосипедів, а також прогнозування їх технічного стану та попиту на послугу оренди.

Ключові слова: автоматизація, прокат велосипедів, інтелектуальна система, штучний інтелект, інформаційна система, технічний стан, попит, програмний застосунок.

ABSTRACT

The bachelor's thesis consists of 109 pages of A4 format, on which there is 44 figure, the list of used sources contains 35 names. The purpose of the work is to develop an intelligent computer system for automating the bicycle rental process using artificial intelligence methods to predict the technical condition of bicycles, forecast the demand for rental and optimize service.

In the theoretical and methodological part of the work, the subject area and the object of developing an intelligent computer system for bicycle rental were considered in detail and analyzed. The basic principles of the functioning of rental systems, the possibilities of using artificial intelligence to predict the technical condition of bicycles and the demand for rental were highlighted, and the criteria for the effectiveness of the system were determined. The advantages and disadvantages of existing solutions in the field of transport rental were considered in detail. In the practical part, a system was developed and tested to ensure automation of the processes of registration, selection and rental of bicycles, as well as forecasting their technical condition and the level of demand for the rental service.

Keywords: automation, bicycle rental, intelligent system, artificial intelligence, information system, technical condition, demand, software application.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ.....	6
1.1 Аналіз предметної області.....	6
1.2 Аналіз об'єкта автоматизації	9
1.3 Аналіз існуючих систем прокату велосипедів	11
1.4 Характеристика об'єкта оренди та умови надання послуг.....	16
1.5 Аналіз задач, які можуть бути розв'язані за допомогою методів штучного інтелекту.....	19
1.6 Постановка задачі.....	20
2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ.....	23
2.1 Аналіз функцій автоматизованої системи	23
2.2 Проектування архітектури автоматизованої системи	27
2.3 Аналіз та обґрунтування технологій розробки	29
3 РОЗРОБКА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ.....	34
3.1 Розробка моделі прогнозу технічного стану для велосипедів.....	34
3.1.1 Оцінка ймовірності відмови велосипеда згідно теорії надійності.....	34
3.1.2 Постановка задачі класифікації технічного стану велосипеда	36
3.1.3 Обґрунтування вибору набору даних для навчання моделі	38
3.1.4 Варіантний аналіз моделей для розв'язання задачі класифікації	40
3.1.5 Навчання та тестування моделей.....	41
3.2 Розробка моделі прогнозу попиту оренди велосипедів	46
3.2.1 Постановка задачі прогнозування попиту оренди велосипедів	46
3.2.2 Вибір вхідних ознак для моделі прогнозу попиту	47
3.2.3 Побудова та обробка набору даних для навчання	49

3.2.4 Критерії якості регресійних моделей	50
3.2.5 Варіантний аналіз моделей для вирішення задачі регресії.....	52
3.2.6 Навчання моделей та оцінювання якості прогнозу	54
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ	60
4.1 Вибір інструментів розробки	60
4.1.1 Вибір середовища розробки програмного забезпечення	60
4.1.2 Вибір інструментів зберігання інформації	61
4.1.3 Вибір інструментів розробки користувацького інтерфейсу	65
4.2 Розробка клієнської частини системи прокату велосипедів.....	66
ВИСНОВКИ.....	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
Додаток А(обов'язковий). Протокол перевірки на плагіат.....	83
Додаток Б(обов'язковий). Технічне завдання	85
Додаток В(довідковий). Лістинг коду	88
Додаток Г(обов'язковий). Ілюстративна частина	92

ВСТУП

У сучасних умовах розвитку урбанізації та популяризації активного способу життя все більшої популярності набувають сервіси прокату різноманітних видів транспорту. Окреме місце серед них займає оренда велосипедів, яка сприяє розвитку екологічного транспорту, покращенню фізичної активності населення та зменшенню навантаження на дорожню інфраструктуру. Особливо актуальним є створення систем прокату, які орієнтовані не лише на масове використання звичайних міських велосипедів, а й на надання користувачам доступу до різних типів велосипедів - гірських, спортивних, туристичних тощо.

Актуальність. Актуальність теми обумовлюється необхідністю впровадження інтелектуальних рішень для управління процесами прокату велосипедів різних типів через організовані пункти видачі. Відсутність системного підходу до моніторингу технічного стану велосипедів може призвести до зниження рівня безпеки та якості обслуговування клієнтів. Також важливо враховувати коливання попиту на оренду залежно від погодних умов, сезонності та дня тижня. Саме тому виникає потреба у створенні комп'ютерної системи, яка б забезпечувала не лише управління прокатом, але й інтелектуальне прогнозування технічного стану та попиту на велосипеди.

Мета дослідження полягає у підвищенні ефективності процесу прокату велосипедів.

Для досягнення поставленої мети передбачено виконання наступних завдань:

- аналіз сучасних систем прокату та вимог до прокату велосипедів різного типу;
- розробка архітектури інтелектуальної комп'ютерної системи прокату;
- реалізація програмного комплексу для реєстрації, вибору, бронювання та оренди велосипедів через точки видачі;
- створення модуля прогнозування попиту прокату велосипедів;

- створення модуля прогнозування технічного стану велосипедів на основі аналізу історії використання та зовнішніх факторів;
- тестування розробленої системи та оцінка її ефективності.

Об'єкт дослідження - процес організації прокату велосипедів різного призначення через спеціалізовані пункти видачі.

Предмет дослідження - математичні моделі та методи прогнозування технічного стану транспортних засобів та попиту оренди.

Новизна розробленої системи полягає в розробці моделей машинного навчання для прогнозування технічного стану велосипедів із урахуванням умов експлуатації, часу використання, навантажень і типу траси та для прогнозування попиту оренди. Це дозволяє своєчасно виявляти потребу в обслуговуванні або ремонті, оптимізувати витрати на технічне утримання та підвищити безпеку користувачів, а інтеграція модуля прогнозування попиту дозволяє системі не лише реагувати на замовлення, а й проактивно готуватися до пікових періодів.

Практична цінність полягає у розробці програмного забезпечення системи оренди велосипедів, що використовує моделі штучного інтелекту. Реалізація інтелектуальної системи дає змогу автоматизувати процес обліку, видачі, контролю та обслуговування велосипедів, підвищити якість сервісу для клієнтів, мінімізувати кількість аварійних ситуацій та підвищити ефективність роботи підприємств.

Апробація. Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2025)»:

Публікації: Кабашний Олександр Анатолійович «Короткий огляд розробки веб-додатку для управління процесом оренди велосипедів з використанням моделей машинного навчання», «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2025. URL:

<https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/25469/21048>

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ

1.1 Аналіз предметної області

В умовах сучасного розвитку міської інфраструктури все більшого значення набувають альтернативні засоби пересування, серед яких важливе місце посідає велосипедний транспорт. Зростаюча популярність здорового способу життя, потреба в зменшенні викидів вуглецю та бажання уникати міських заторів стимулюють розвиток сервісів прокату велосипедів. Це дозволяє жителям міст швидко й зручно долати короткі відстані, не маючи при цьому необхідності купувати власний транспортний засіб.

Сфера прокату велосипедів [1] відноситься до галузі надання послуг тимчасового користування транспортними засобами. Такий вид послуг має низку переваг як для бізнесу, так і для споживачів:

- можливість використання транспортного засобу без витрат на його придбання та обслуговування;
- економія часу за рахунок мобільності;
- сприяння розвитку екологічно чистого транспорту;
- забезпечення активного відпочинку та оздоровлення населення.

Зі збільшенням попиту на прокат велосипедів виникає необхідність удосконалення процесів управління цією діяльністю. Ефективне функціонування прокатного бізнесу можливе лише за умови чіткої організації обліку велосипедів, планування їх технічного обслуговування та взаємодії з клієнтами. Відсутність налагоджених бізнес-процесів призводить до неефективного використання транспортного парку, зниження рівня обслуговування та втрати клієнтської бази.

Організація системи прокату велосипедів включає в себе такі ключові елементи:

- формування парку велосипедів і класифікація за категоріями (гірські, дорожні, дитячі, електровелосипеди тощо);
- облік наявності велосипедів та їх технічного стану;
- реєстрація клієнтів і управління замовленнями;
- контроль за виконанням замовлень і плануванням технічного обслуговування;
- ведення фінансового обліку і аналіз ефективності роботи.

Особливої актуальності набуває автоматизація цих процесів для оптимізації управління та підвищення конкурентоспроможності бізнесу. Використання інформаційних систем дозволяє уникнути помилок у обліку, забезпечити клієнтів актуальною інформацією про доступні транспортні засоби, спростити процедуру оформлення замовлень та покращити контроль за технічним станом велосипедів.

Тепер більш детально розглянемо проблеми предметної області [2].

При відсутності автоматизації в сфері велопрокату виникають такі проблеми:

1. Відсутність централізованого обліку парку велосипедів.

Це ускладнює процес визначення наявності вільних велосипедів, їх технічного стану та місцезнаходження.

2. Труднощі в обліку замовлень.

Ручна обробка замовлень збільшує ймовірність помилок, втрати інформації та конфліктів при бронюванні велосипедів.

3. Низька ефективність управління технічним обслуговуванням.

Відсутність системи нагадувань про планове обслуговування призводить до несвоєчасного ремонту та підвищення ризику виходу велосипедів з ладу.

4. Відсутність зручних інструментів для клієнтів.

Неможливість швидко зареєструватися, переглянути наявні велосипеди або оформити замовлення онлайн знижує рівень обслуговування та лояльність клієнтів.

5. Складність у веденні фінансового обліку.

Відсутність єдиної бази даних ускладнює аналіз прибутків, витрат і оцінку рентабельності бізнесу.

Існують такі шляхи вирішення проблем предметної області [2].

Для підвищення ефективності роботи прокатного бізнесу необхідно впроваджувати інтелектуальні комп'ютерні системи, які дозволять:

- централізовано обліковувати весь парк велосипедів;
- забезпечити клієнтів актуальною інформацією про наявність велосипедів та умови їх оренди;
- надати можливість онлайн-бронювання велосипедів на зручну для клієнта дату;
- автоматизувати облік замовлень, технічного стану велосипедів і планування ремонтних робіт;
- сформувати особистий кабінет для кожного клієнта з можливістю перегляду історії замовлень і керування особистими даними;
- забезпечити адміністратора системи зручним інтерфейсом для обліку технічного обслуговування і планування ремонтів.

Розробка такої системи дозволить скоротити час обробки замовлень, мінімізувати кількість помилок у роботі персоналу, покращити обслуговування клієнтів і підвищити ефективність управління бізнесом.

Предметна область дослідження - є актуальною та перспективною для впровадження сучасних інформаційних технологій. Використання інтелектуальної комп'ютерної системи дозволяє комплексно вирішувати проблеми обліку, планування, взаємодії з клієнтами та аналізу фінансових результатів діяльності підприємства.

Це створює передумови для підвищення прибутковості бізнесу, розширення клієнтської бази та забезпечення високого рівня сервісу, що є важливим фактором конкурентоспроможності на ринку прокатних послуг.

1.2 Аналіз об'єкта автоматизації

Процеси, пов'язані з організацією прокату велосипедів, мають власну специфіку, що вимагає ретельного підходу до управління матеріальними ресурсами та взаємодії з клієнтами. Основним об'єктом автоматизації в даному випадку виступають бізнес-процеси, пов'язані з обліком велосипедів [3], формуванням і супроводом замовлень, зберіганням інформації про клієнтів і моніторингом технічного стану транспортних засобів.

На сьогодні управління прокатом велосипедів без застосування спеціалізованої системи супроводжується низкою проблем. Відсутність централізованого обліку призводить до того, що інформація про наявність велосипедів є неповною або неактуальною [2]. Це створює труднощі при обробці замовлень і часто призводить до ситуацій, коли клієнти не можуть орендувати бажаний транспортний засіб через помилки в обліку. Ручне ведення записів спричиняє зростання кількості помилок при реєстрації клієнтів і замовлень, а також ускладнює контроль за виконанням взятих зобов'язань перед споживачами послуг.

Застосування автоматизованої системи дозволяє централізувати облік ресурсів і значно спростити доступ до актуальної інформації як для адміністрації прокату, так і для клієнтів. Наявність електронної бази даних забезпечує можливість оперативного пошуку необхідної інформації, фіксації стану велосипедів, аналізу їх завантаження та історії використання. Це особливо важливо для своєчасного планування технічного обслуговування транспортних засобів, що в свою чергу позитивно впливає на безпеку та комфорт клієнтів.

Клієнтська частина системи дозволяє спростити процедуру взаємодії з сервісом[2]. Зареєстровані користувачі можуть самостійно обирати велосипеди з доступних категорій, формувати замовлення на зручну дату, переглядати власну історію прокатів і за необхідності скасовувати бронювання без участі адміністрації. Такий підхід суттєво економить час як для клієнтів, так і для персоналу прокатної

точки. Крім того, створення особистих профілів дозволяє вести повну історію взаємодії кожного користувача з сервісом, що є важливим для надання персоналізованих послуг і підвищення лояльності постійних клієнтів.

З боку адміністрації автоматизована система забезпечує спрощене управління парком велосипедів, дозволяє своєчасно отримувати інформацію про необхідність ремонту окремих одиниць транспорту, контролювати виконання замовлень і формувати звітність щодо діяльності підприємства. Керівництво отримує доступ до аналітичних даних, що дозволяють оцінити ефективність використання кожної одиниці транспорту, виявляти найбільш затребувані категорії велосипедів та приймати обґрунтовані рішення щодо оновлення або розширення парку.

Важливою характеристикою об'єкта автоматизації є чітко визначені межі автоматизації. Система має охоплювати лише ті процеси, які пов'язані з обліком ресурсів, взаємодією з клієнтами та контролем за інформацією про стан велосипедів. Фізичні процеси, пов'язані з передачею велосипедів клієнтам, їх обслуговуванням та транспортуванням, залишаються в межах відповідальності персоналу і не автоматизуються. Це дозволяє уникнути надмірного ускладнення системи та зосередитися на забезпеченні високої якості інформаційної підтримки основної діяльності прокату.

Очікуваним результатом впровадження автоматизованої системи є підвищення точності обліку та зменшення кількості помилок, покращення швидкості обробки замовлень і підвищення рівня задоволеності клієнтів сервісу. Наявність централізованої системи обліку сприятиме підвищенню прозорості діяльності підприємства, забезпечить зручний доступ до інформації для прийняття управлінських рішень та дасть можливість планувати подальший розвиток бізнесу на основі реальних даних про попит і завантаження ресурсів.

Для наочного розуміння основних компонентів [2] та взаємозв'язків у сфері прокату велосипедів доцільно розглянути схематичну структуру автоматизованої системи прокату велосипедів, яка зображена на рисунку 1.1.



Рисунок 1.1 - Схематична структуру автоматизованої системи прокату велосипедів

1.3 Аналіз існуючих систем прокату велосипедів

Розвиток інформаційних технологій значно вплинув на організацію сервісів прокату транспортних засобів, зокрема велосипедів. В умовах високої конкуренції дедалі більше компаній вдаються до впровадження автоматизованих інформаційних систем, що забезпечують облік, бронювання та управління транспортними засобами. Однак більшість таких систем орієнтовані переважно на великі міста та звичайний прокат в межах міста, який не враховує потреб окремих

користувачів, які потребують більшого вибору товарів та їх тип використання(гори, шосе і т.д.)

Тому для повного розуміння системи повноцінного велопрокату проведемо аналіз існуючих систем, які пропонують вибір як для міжміського прокату, так і для систем з більш ширшим спектром вибору товарів, які включають різний тип велосипедів для різних задач.

В якості найближчих аналогів обрано такі 3 системи:

- Veliki.ua;
- Bikenow;
- Velokratia.

Veliki.ua [4] – це популярний сервіс для прокату різного товару, таких як велосипеди, самокати та інше приладдя для відпочинку, але, як можна побачити по назві, основний вид надання послуг – це велосипеди. Також ця платформа надає послуги з ремонту транспортних засобів. Окрім перерахованих послуг, Veliki.ua пропонують школу, яка може навчити вас їздити на велосипеді і курси веломеханіків. Компанія позиціонує себе як найбільша мережа велосервісу в Україні, орієнтована як на приватних, так і на корпоративних клієнтів. Користувацький інтерфейс веб додатку наведено на рисунку 1.2.

Не зважаючи на таку різноманітність послуг, Veliki.ua має свої недоліки. Основним з них - це відсутність каталогу товарів для оренди велосипедів. Тобто для того щоб орендувати велосипед, користувачу потрібно їхати на точку видачі і вже там обирати з наявних товарів. З одного боку – це добре, так як вас зразу там проконсультують і допоможуть підібрати відповідний товар, але з іншого боку, набагато краще, якщо можна було б зразу переглянути асортимент товарів, щоб не витрачати свій дорогоцінний час, якщо вам не вдасться підібрати бажаний товар.

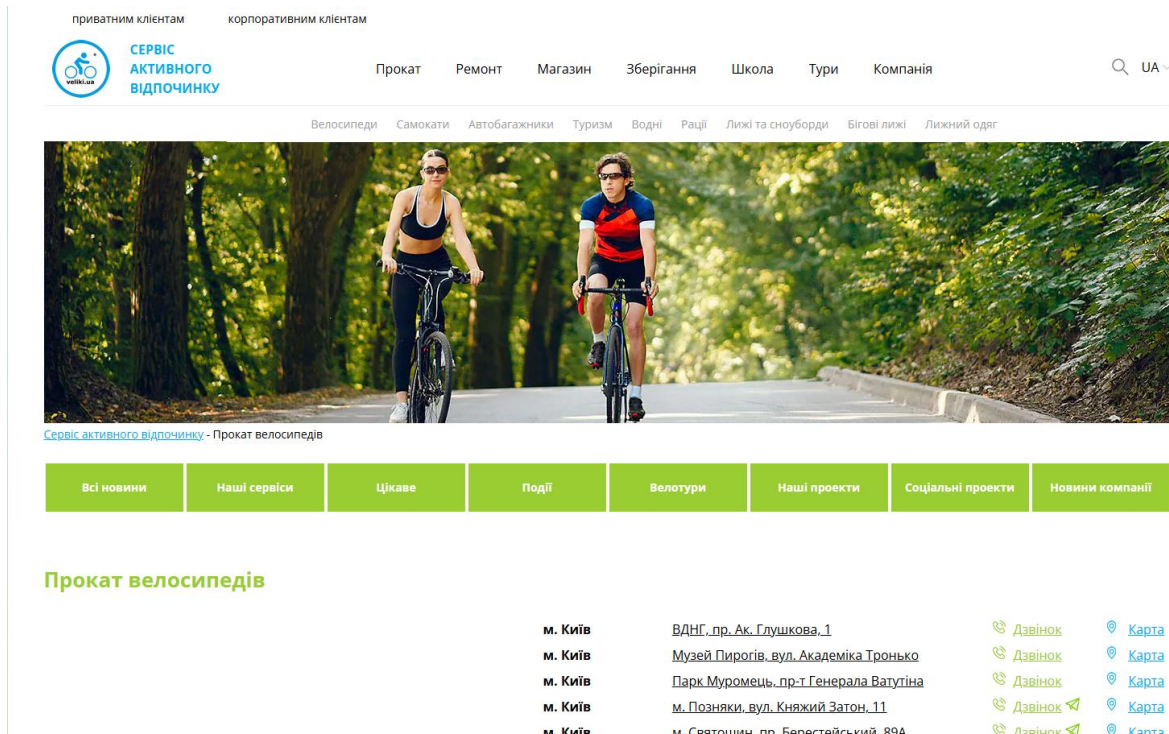


Рисунок 1.2 - Користувацький інтерфейс Veliki.ua

Також до цього можна додати відсутність змоги орендувати велосипед на бажану дату на майбутнє, так як сайт не надає функціоналу резервування транспорту на даний момент, це можна зробити тільки на точці видачі, що є не зовсім зручним. Можна ще сказати про неможливість переглядати свою історію замовлень, так як у цій системі немає змоги створити власний аккаунт. Як можна побачити на рисунку 1.2 – сайт має доволі застарілий вигляд та функціонал, що може вплинути на перше враження у нових користувачів.

Vikenow [5] – це український сервіс громадського велопрокату, який працює з 2018 року. Спочатку він функціонував під брендом Nextbike, але згодом провів ребрендинг і розширив свою діяльність, охоплюючи Київ та Львів. Це зручний сервіс для щоденних поїздок, для когось це може бути альтернативою до громадського транспорту. На відміну від попереднього сервісу, цей цілком відрізняється своєю структурою та механікою. В даній системі можна орендувати велосипед тільки на точках, локаціях за допомогою мобільного додатку скануючи

QR-Code для оплати. Такий формат є цілком зручним для короткострокових поїздки містом. Також у додатку можна відслідковувати свою історію поїздки, поповнювати баланс аккаунту, відслідковувати найближчу локацію транспорту.

Однак, попри очевидні переваги сервісу існують деякі недоліки. Основним недоліком можна вважати однотипність велосипедів. Всі велосипеди виглядають однаково і мають однакові характеристики. Крім цього Vikepow не дає змоги здійснити бронювання велосипеда на конкретну дату в майбутньому, що є важливим для тих, хто планує свої поїздки наперед. І хоча для щоденних потреб такий підхід є виправданим, сервіс обмежує використання транспорту в ситуаціях, коли треба мати гарантію наявності велосипеда і в певній локації у певний час. Щодо інтерфейсу мобільного додатку, хоч він є простим і функціональним, деякі користувачі хочуть мати більш розширені налаштування та кастомізацію свого профілю.

Отже, Vikepow є гарним одним з найкращих сервісів сучасної системи прокату велосипедів для міських короткострокових поїздки. Але через своїх недоліки, ця система може не повністю задовільняти потреби всіх категорій користувачів.

Користувацький інтерфейс мобільного застосунку наведено на рисунку 1.3.

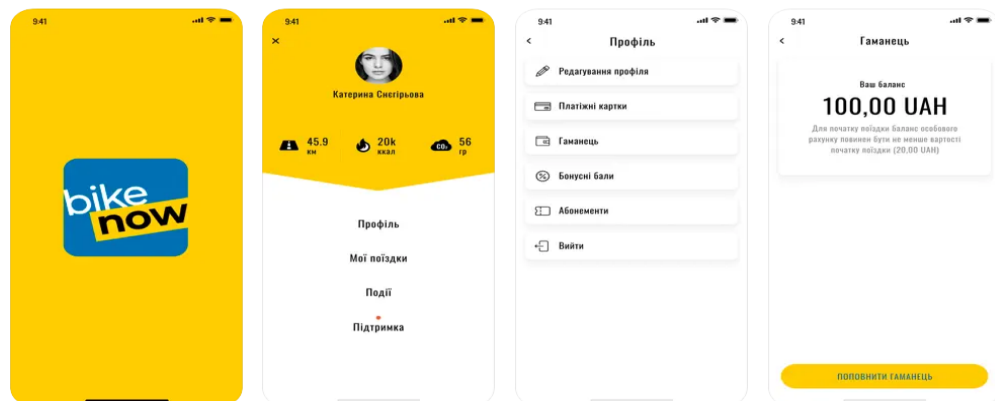


Рисунок 1.3 - Користувацький інтерфейс мобільного застосунку Vikepow

Velokratia[6] – це український сервіс для прокату велосипедів, точки яких знаходяться у популярних місцях для відпочинку, таких як ВДНГ, парк Феофанія, Межегір'я та інші. Їхня мета – це забезпечити зручний доступ для прокату у парках та туристичних локаціях.

Загалом, сервіс Velokratia є зручним рішенням для спонтанних поїздки у туристичних місцях, але для користувачів, які надають перевагу онлайн бронюванням без постійного спілкування з операторами, можуть виникнути певні незручності. Користувацький інтерфейс вебсайту наведено на рисунку 1.4.

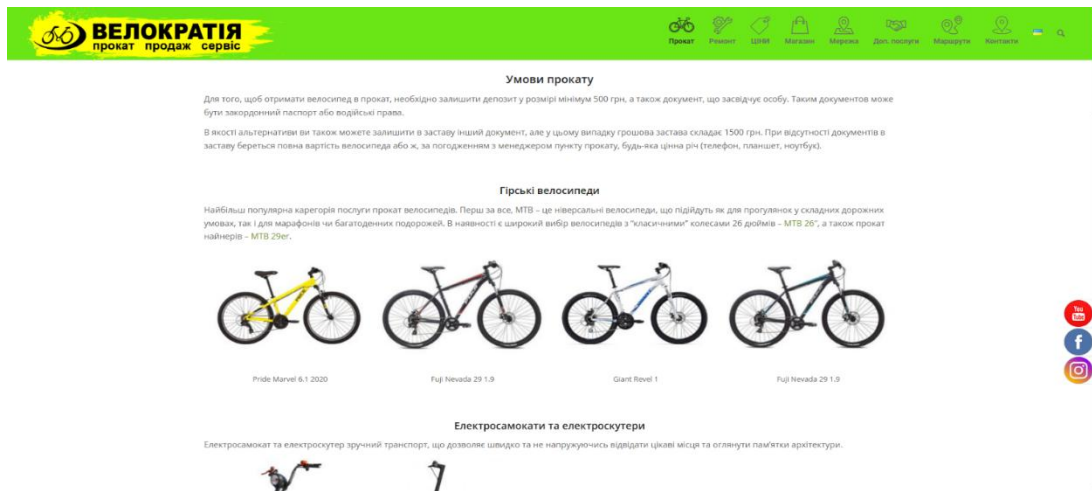


Рисунок 1.4 - Користувацький інтерфейс Velokratia

Сервіс пропонує великий спектр товару по різних категоріях: гірські, міські, дитячі. Також в наявності є додаткові аксесуари, такі як шоломи та дитячі крісла. Все це можна переглянути у каталозі товарів та прочитати детальний опис про кожний з них. Velokratia також надає послуги з доставки транспорту на замовлення. Таких підхід є зручним для корпоративних заходів чи певних користувачів, яким зручніше використовувати такий спосіб отримання товару. Крім того, компанія пропонує технічне обслуговування велосипедів, зі змогою виїзду до клієнта, що дозволяє оперативно вирішувати проблеми зі транспортом та його налаштуванням.

З мінусів цього сервісу можна відзначити неможливість забронювати товар онлайн на певну дату автоматизовано, для цього потрібно залишати заявки та зв'язуватись з менеджером для узгодження дати та кількості товару. Також до недоліків можна віднести відсутність профілю користувача через що неможливо відслідковувати свої попередні і активні замовлення, також для скасування прокату потрібно знову телефонувати менеджерам, для когось ця модель може бути не зручною.

1.4 Характеристика об'єкта оренди та умови надання послуг

Об'єктом оренди в автоматизованій системі є велосипеди різного типу, що надаються в тимчасове користування зареєстрованим клієнтам через організовані пункти видачі. Велосипеди виступають основним матеріальним ресурсом сервісу прокату, від характеристик яких залежить зручність, безпечність та ефективність користування послугою.

На сучасному ринку велопрокату використовуються моделі, які розрізняються за функціональним призначенням, технічними параметрами, способом керування, конструкцією та додатковим обладнанням. Відповідно до цього об'єкти оренди класифікуються за кількома основними ознаками.

Класифікація велосипедів за призначенням[7]:

1. Міські велосипеди - призначені для комфортного пересування по асфальтованих дорогах, мають просту конструкцію, вертикальну посадку та часто оснащуються багажниками або кошиками.
2. Гірські велосипеди (МТВ) - орієнтовані на пересічену місцевість, обладнані амортизаційною вилкою, широкими колесами з протектором і міцною рамою.

3. Шосейні велосипеди - легкі та швидкі, використовуються для їзди на великі відстані асфальтованими дорогами, мають вузькі колеса та аеродинамічну посадку.
4. Туристичні велосипеди - універсальні моделі для тривалих подорожей, часто оснащені додатковими кріпленнями для сумок, мають підвищений комфорт та надійність.
5. Дитячі велосипеди - адаптовані за розміром та вагою для дітей різного віку, часто мають додаткові колеса або обмеження швидкості.
6. Електровелосипеди - обладнані електродвигуном, що полегшує рух на складних ділянках або при втомі користувача.

Основні характеристики, що зазначаються в описі кожного велосипеда[8]:

- тип велосипеда та категорія призначення;
- розмір рами та діаметр коліс;
- кількість передач;
- тип гальмівної системи (дискові, ободні тощо);
- наявність амортизації (передньої/задньої);
- допустиме навантаження (максимальна вага користувача);
- орієнтовна висота користувача;
- рік випуску;
- виробник/бренд;
- фото товару.

Ці характеристики дозволяють користувачам підібрати велосипед відповідно до власних потреб і фізіологічних особливостей. У системі реалізується фільтрація за вказаними параметрами, що значно спрощує вибір.

Принципи формування вартості оренди базуються на ринкових умовах і включають кілька змінних[9]:

1. Тип велосипеда - електровелосипеди та гірські моделі мають вищу ціну через вартість комплектуючих та експлуатації.
2. Тривалість прокату - передбачено погодинну, денну та тижневу тарифікацію. При збільшенні терміну діє система знижок.
3. Сезонний коефіцієнт - у періоди пікового навантаження (весна–літо, вихідні дні) можливе підвищення базового тарифу.
4. Додаткові послуги - шоломи, дитячі сидіння, замки чи ліхтарі можуть додаватись до замовлення за окрему плату.
5. Тип замовлення - індивідуальне чи корпоративне (групове), що впливає на цінову політику.

Умови оренди, передбачені в системі, включають:

- реєстрацію користувача в системі з підтвердженням контактних даних;
- попереднє бронювання велосипеда через вебінтерфейс із зазначенням дати отримання та повернення;
- відповідальність за цілісність майна на період оренди згідно з умовами публічної оферти;
- штрафні санкції в разі запізнення з поверненням або пошкодження транспорту.

Таке структурування інформації про об'єкти оренди та прозорі правила користування створюють передумови для комфортної взаємодії клієнтів з сервісом, підвищують довіру до платформи та сприяють формуванню стабільної клієнтської бази.

1.5 Аналіз задач, які можуть бути розв’язані за допомогою методів штучного інтелекту

Сайти прокату велосипедів на перший погляд здаються простими - каталог, бронювання, видача. Але як тільки сервіс розширюється, зростає кількість замовлень, пунктів прокату й користувачів, - починають з’являтися питання, які не вирішиш вручну або простими умовними конструкціями в коді. Саме тут методи штучного інтелекту показують свою ефективність.

Одна з головних задач - це передбачення попиту. Скільки велосипедів орендують завтра, якщо прогнозують +25 і безвітряно? Чи вистачить транспорту у вихідні, якщо поблизу проходить масовий захід? Моделі машинного навчання, зокрема XGBoost та LSTM, дозволяють враховувати погоду, день тижня, сезонність та інші чинники, щоб сформувавши прогноз. У дослідженні [10] “Short-term bike-sharing demand forecasting incorporating multiple sources of uncertainties” ці підходи довели свою ефективність при побудові моделей короткострокового прогнозування попиту в міських умовах.

Друга важлива задача - контроль технічного стану велосипедів. Традиційно це робиться вручну: після певної кількості днів чи кілометрів транспорт направляють на огляд. Але якщо проаналізувати історію обслуговування, пробіг, типи поломок, можна навчити модель передбачати, коли варто провести сервіс до того, як щось зламається. У прикладі системи[11] “Cycling into the workshop: predictive maintenance for Barcelona's bike-sharing system” описано, як така модель впроваджена у Барселоні - і це дозволяє мінімізувати несподівані збої в роботі транспорту.

Наступна сфера - персоналізовані рекомендації. Наприклад, якщо користувач зазвичай орендує гірські велосипеди, система може запропонувати саме їх, або показати знижку саме на той тип. Алгоритми рекомендаційного типу давно

використовуються у великих сервісах, і добре себе зарекомендували навіть у прокатах транспорту. Одним із прикладів є система[12] “SAFEBIKE: A Bike-sharing Route Recommender with Availability Prediction and Safe Routing”, де рекомендації враховували не лише вподобання користувача, а й безпеку маршруту.

Ще один напрям - виявлення аномальної поведінки. Наприклад, якщо хтось орендує велосипед і довго не повертає, або часто змінює маршрут у дивний спосіб - система може зреагувати автоматично. У дослідженні[13] “Predictive and Prescriptive Performance of Bike-Sharing Demand Forecasts for Inventory Management” описано механізм виявлення нетипової активності на основі кластеризації поведінкових даних, що дозволяє адміністрації прокату оперативно реагувати на підозрілі ситуації.

Штучний інтелект не просто автоматизує повторювані дії - він дозволяє системі реагувати на ситуації гнучко і на основі даних. Його використання в прокатах велосипедів поступово стає не бонусом, а вимогою для конкурентоспроможного сервісу.

1.6 Постановка задачі

Ретельно проаналізувавши існуючі системи велопрокату, з’явилась ідея по створенню своєї системи, яка буде включати в себе багато переваг з інших моделей з мінімізуванням їх недоліків. Метою розробки є створення і в подальшому масштабування інтелектуальної комп’ютерної системи для автоматизації процесів прокату велосипедів, яка дозволить оптимізувати бізнес-процеси, підвищити якість обслуговування клієнтів і забезпечити ефективне управління ресурсами. Система має об’єднати в собі як класичний вебзастосунок для взаємодії з клієнтами та адміністраторами, так і інтелектуальну підсистему для прогнозування технічного стану велосипедів і аналізу попиту.

Загалом у межах розробки планується вирішити такі основні задачі:

1. Автоматизація процесів обліку велосипедів та організації їх прокату.
2. Забезпечення можливості реєстрації та авторизації користувачів через вебінтерфейс.
3. Надання клієнтам зручного інструменту для перегляду доступних велосипедів, оформлення замовлень на конкретну дату та керування власним профілем. Реалізація особистого кабінету користувача з можливістю перегляду історії замовлень, зміни особистих даних та скасування бронювань та кастомізацією.
4. Впровадження адміністративного інтерфейсу для управління станом парку велосипедів, перегляду історії ремонтів і планування технічного обслуговування.
5. Інтеграція інтелектуальної підсистеми для прогнозування потреби в ремонті велосипедів на основі накопичених даних.
6. Аналіз даних про попит на послуги прокату відповідно до погоди для покращення розподілу велосипедів і оптимізації роботи сервісу.

До розроблюваної системи висуваються такі основні вимоги:

1. Зручний та зрозумілий вебінтерфейс для клієнтів та адміністрації, адаптований під різні типи пристроїв.
2. Система обліку даних з надійною базою даних, що забезпечує зберігання інформації про клієнтів, велосипеди, замовлення та історію технічного обслуговування.
3. Прогнози інтелектуальної підсистеми щодо необхідності технічного обслуговування та ремонту велосипедів.
4. Можливість роботи системи при обмеженій кількості вхідних даних, з використанням як реальних, так і синтетичних даних для навчання моделей прогнозування.

5. Забезпечення безпеки персональних даних користувачів та захист системи від несанкціонованого доступу.
6. Оптимізація продуктивності системи для забезпечення швидкої обробки запитів і прогнозів у реальному часі.

Таким чином, розроблювана система має не лише автоматизувати базові бізнес-процеси прокату велосипедів, а й забезпечити підтримку прийняття різних рішень за допомогою сучасних методів аналізу даних і прогнозування. Це дозволить підвищити якість обслуговування клієнтів, покращити планування технічного обслуговування та створити умови для сталого розвитку прокатного сервісу.

2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ

2.1 Аналіз функцій автоматизованої системи

Для повноцінної роботи системи вона повинна мати такі основні функції:

Адміністративна частина:

1. Управління каталогом велосипедів. Адміністратор може додавати та забирати товар, а також редагувати їх характеристики.
2. Керування замовленнями. Їх обробка.
3. Керування історією ремонтів. Адміністратор може вносити дані про проведені ремонти та технічні огляди. Інформація про останнє обслуговування відображається у картці кожного екземпляра велосипеда.
4. Адміністративна панель. Надає змогу адміністратору ефективно управляти контентом сайту, переглядати статистику замовлень, керувати базою клієнтів і контролювати стан парку велосипедів.
5. Прогнозування технічного обслуговуванні та попиту оренди відповідно до погоди. Інтелектуальна підсистема обслуговування аналізує пробіг, історію попередніх ремонтів та інші параметри, формуючи рекомендації щодо необхідності обслуговування велосипедів. Інтелектуальна підсистема прогнозування попиту аналізує погодні умови та кількість попередніх оренд в таку погоду.

Клієнтська частина:

1. Реєстрація та авторизація. Система передбачає створення особистого кабінету з можливістю перегляду історії замовлень, скасування замовлень, редагування особистих даних.

2. **Замовлення.** Користувачі можуть оформити замовлення на прокат велосипеда на конкретну дату. Передбачена можливість попереднього бронювання, що дозволяє заздалегідь зарезервувати велосипед.
3. **Каталог.** Користувач бачить весь доступний асортимент товару, його розміри, ціни та характеристики. Він може фільтрувати товар по розміри і категорії (гірські, міські, спортивні тощо).
4. **Рекомендації.** Системи рекомендує конкретний тип товару, аналізуючи попередні замовлення.

Для опису можливостей системи зазвичай використовується UML-діаграми[14].

На рисунку 2.1 зображена діаграма варіантів використання для двох типів користувачів - звичайного користувача та адміністратора.

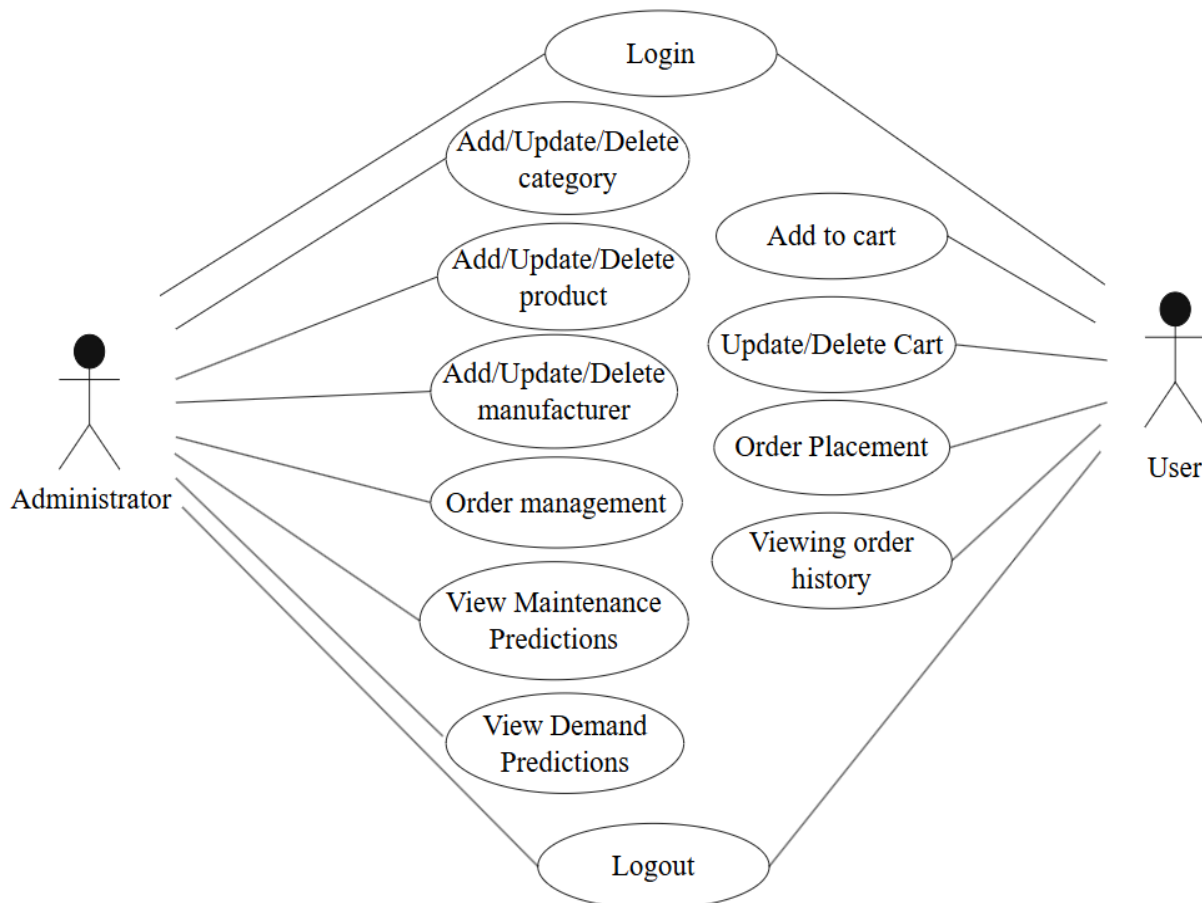


Рисунок 2.1 – UML-діаграма варіантів використання системи

Користувач має доступ до авторизації, перегляду товарів, додавання до кошика, оформлення замовлень, перегляду історії оренд та виходу з системи. Адміністратор, окрім базових дій, може керувати товарами, категоріями, виробниками, обробляти замовлення, переглядати прогноз технічного стану велосипедів та прогноз попиту. Діаграма демонструє розподіл функціональності залежно від ролі користувача.

На рисунку 2.2 діаграма діяльності відображає послідовність дій користувача та адміністратора в системі прокату велосипедів.

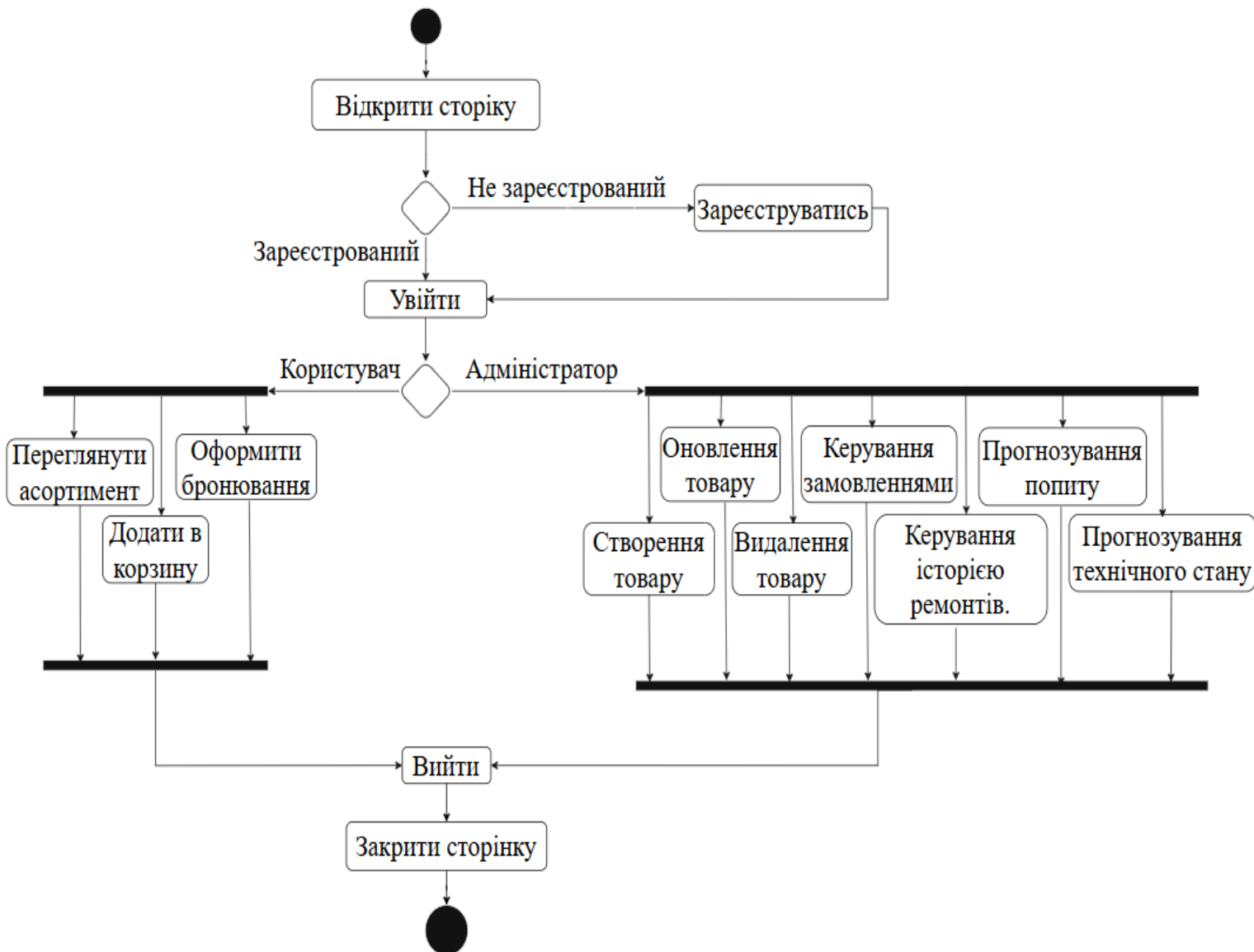


Рисунок 2.2 - Діаграма діяльності

Користувач спочатку реєструється або входить у систему, після чого може переглядати каталог, додавати велосипеди до кошика та оформлювати оренду. Адміністратор після входу отримує доступ до керування товарами (додавання, редагування, видалення), перегляду та зміни статусів замовлень, а також роботи з даними про ремонти й прогнозами технічного стану. Обидва типи користувачів завершують роботу виходом із системи.

На рисунку 2.3 водображена UML-діаграма послідовності, яка демонструє взаємодію між користувачем, інтерфейсом, системою та базою даних під час роботи з вебзастосунком прокату велосипедів.

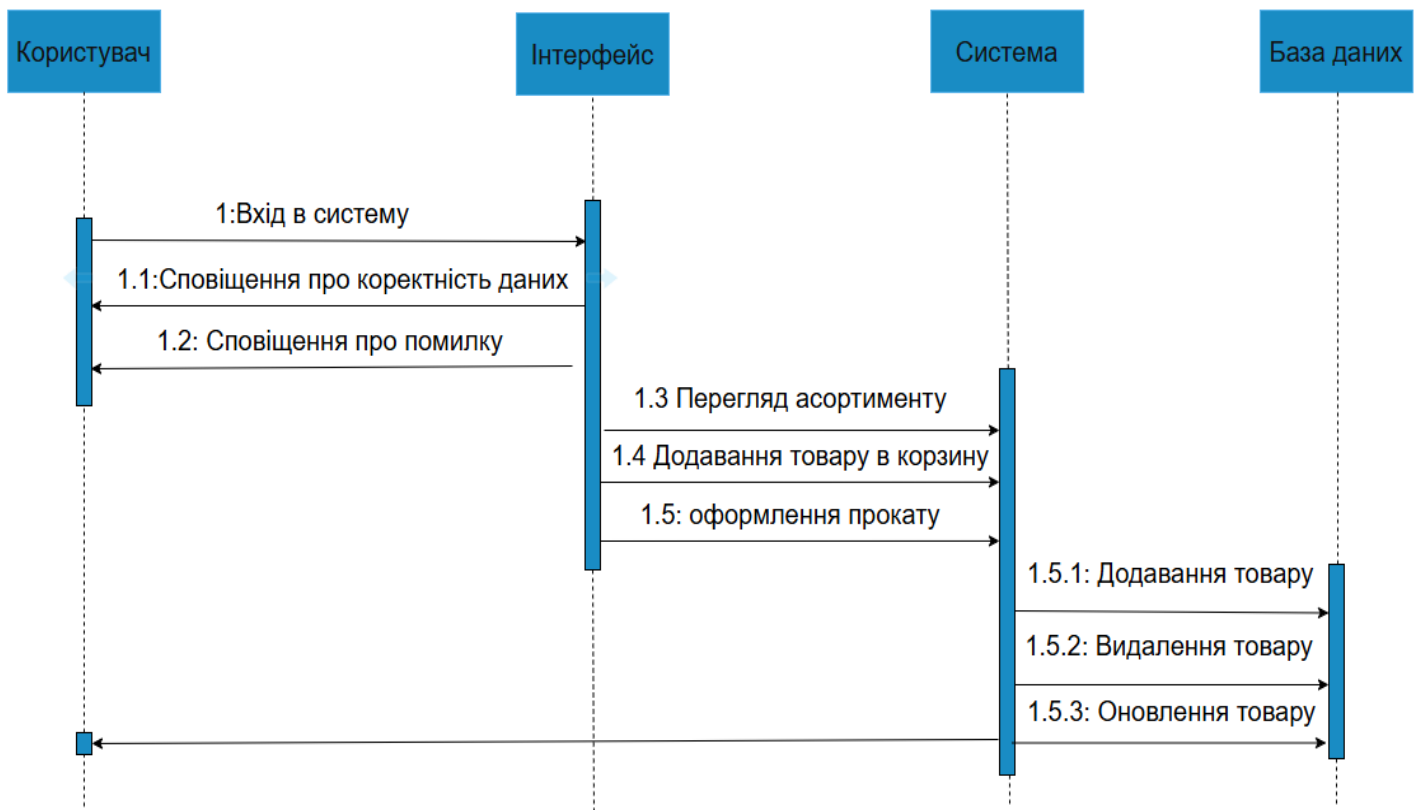


Рисунок 2.3 - Діаграма послідовності

На початку користувач проходить авторизацію через інтерфейс. Якщо дані введено правильно - надходить підтвердження; у разі помилки система генерує

відповідне повідомлення. Після входу користувач має змогу переглянути асортимент велосипедів, додати вибрані позиції до кошика та перейти до оформлення прокату.

Після підтвердження оренди система надсилає запити до бази даних для виконання трьох можливих дій: додавання нового запису про оренду, видалення попередньої інформації або оновлення даних про велосипед. Ця діаграма ілюструє, як дані проходять через усі рівні - від користувача до бази даних - і показує логіку взаємодії між компонентами системи.

Таким чином, запропонована система охоплює повний цикл обслуговування користувачів – від реєстрації на сайті та вибору велосипеда до аналізу ефективності роботи сервісу та підтримки технічного стану транспортних засобів. Це дозволяє підвищити якість обслуговування клієнтів, оптимізувати витрати на обслуговування велосипедів та покращити управління прокатним бізнесом загалом.

2.2 Проектування архітектури автоматизованої системи

Архітектурна схема є ключовим аспектом у проектуванні будь-якої складної системи. Вона визначає загальну структуру системи, взаємозв'язки між її компонентами та встановлює принципи організації системи як цілого.

Клієнт-серверна архітектура: найпоширеніший підхід, при якому функції веб-застосунку розділені між клієнтською частиною (веб-браузером або мобільним додатком) і серверною частиною. Клієнтська частина відповідає за відображення користувацького інтерфейсу та взаємодію з користувачем, тоді як серверна частина забезпечує обробку запитів, доступ до бази даних та бізнес логіку.

Модель – Подання – Шаблон(MVT)[15] - це підхід Django до організації кодової бази та робочого процесу веб-додатку. Компонентами, які складають цю архітектуру, є модель, подання та шаблон. Кожен компонент виконує певні функції,

а потім передає процес іншим компонентам для виконання своїх. Давайте коротко опишемо кожен компонент:

1. Модель: вона керує даними та взаємодіє з базою даних.
2. Подання: він діє як посередник, обробляє логіку та керує потоком даних.
3. Шаблон: він відображає вміст HTML в інтерфейсі користувача.

На рисунку 2.4 - показана модель MVT.

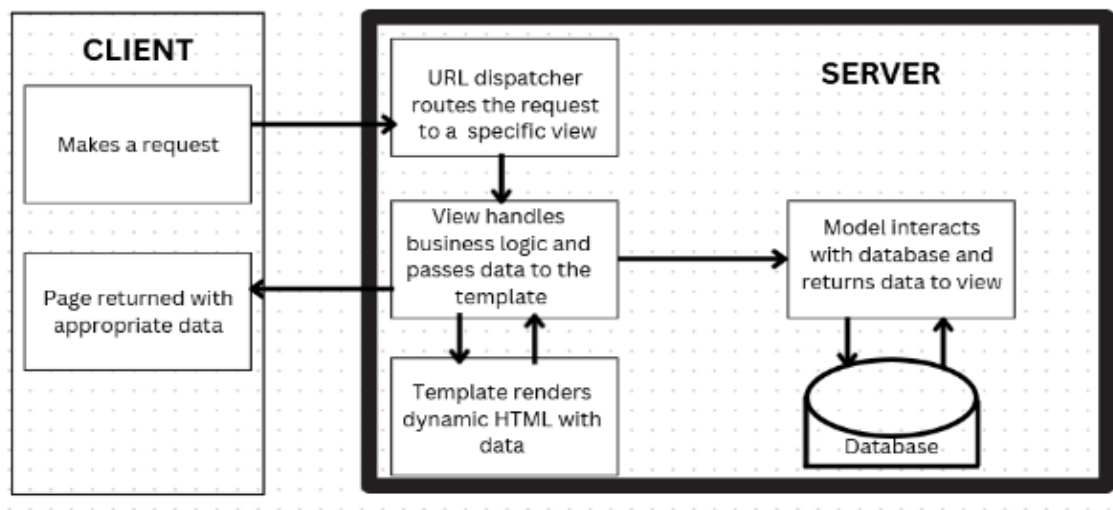


Рисунок 2.4 - модель MVT

Проектування архітектури системи прокату велосипедів розпочинається з визначення основних компонентів, які забезпечують її стабільну роботу та зручність для користувачів і адміністрації. Архітектура будується на основі класичної трирівневої моделі, яка включає клієнтський рівень, серверну частину та рівень зберігання даних.

Клієнтська частина відповідає за взаємодію з користувачами системи. Вебінтерфейс створюється таким чином, щоб навіть користувач без технічних знань міг легко знайти потрібний велосипед у каталозі, оформити замовлення або переглянути свою історію оренд. Для цього використовуються сучасні підходи до розробки інтерфейсів з урахуванням адаптивного дизайну.

Серверна частина виконує всю бізнес-логіку системи. Тут обробляються запити від клієнтів, керується процес бронювання велосипедів, обробляються дані про технічний стан транспорту і виконується взаємодія з базою даних. Також на цьому рівні розміщено інтелектуальну підсистему, яка аналізує накопичені дані і формує прогнози щодо технічного обслуговування велосипедів.

Рівень зберігання даних представлений надійною реляційною системою, яка відповідає за збереження інформації про клієнтів, велосипеди, замовлення, історію технічного обслуговування та інші важливі дані. Такий підхід дозволяє забезпечити цілісність і захищеність інформації, а також високу швидкість обробки запитів.

Спроектowana архітектура є гнучкою і дозволяє легко вносити зміни або додавати нові модулі в майбутньому без суттєвої перебудови всієї системи.

2.3 Аналіз та обґрунтування технологій розробки

При виборі технологій для реалізації системи враховувалися такі критерії, як зручність розробки, підтримка з боку спільноти, безпека, мої знання, можливість масштабування та простота інтеграції між різними компонентами.

Вибір мови програмування веб-застосунку грає велику роль і може мати значний вплив на розробку та майбутнє функціонування проекту.

При виборі мови потрібно дивитись на такі спекти:

1. Функціональність проекту та його вимоги. Конкретна мова програмування може краще підходити для реалізації конкретної системи. Наприклад вони мають в собі вже вбудовані функції, які не потрібно розробляти окремо, такі як робота з базами даних, обробка даних у реальному часі або взаємодія зі сторонніми сервісами.

2. Досвід та навички розробника. Якщо у людини вже є наявний досвід і тій чи іншій мові, це може набагато полегшити розробку веб сайту, мобільного додатку тощо.
3. Задачі. Перед вибором мови, дуже важливо поставити собі задачі, які будуть у проекті. Деякі мови програмування можуть бути кращі для веб-розробки, деякі для десктоп і мобільної розробки.

Існує багато мов які використовують для веб розробки , найпопулярніші з них є JavaScript, Python, TypeScript, Java, PHP, C# та інші [16]. Всі ці мови надають широкі можливості для розробки і створення своєї унікальної системи. Також дуже важливим етапом розробки є вибір бази даних, для зберігання та керування даними. На даний момент існують такі популярні БД[17]: Oracle, MySQL, PostgreSQL та інші.

Є два найпопулярніші типи баз даних - реляційні та нереляційні [18].

Реляційні бази даних (Relational database). Елементи в реляційній базі даних організовані як набір таблиць зі стовпцями та рядками. Цей тип баз даних визначає зв'язки бази даних у формі таблиць. Її також називають реляційною СУБД, яка є найпопулярнішим типом СУБД. Приклад бази даних системи RDBMS містить базу даних MySQL, Oracle і Microsoft SQL Server.

Нереляційні бази даних NoSQL (NoSQL database). NoSQL, або нереляційна база даних, дає змогу зберігати та маніпулювати неструктурованими (тексти, аудіо, відео, зображення) даними та напівструктурованими (даними з певними властивостями, наприклад, тегами). Бази даних NoSQL ставали популярними, оскільки вебзастосунки ставали все більш поширеними та складнішими.

Також при розробці веб застосунку важливою частиною є вибору фреймворку. Фреймворк[19] – це інструмент для розробки програмного забезпечення, що надає готову структуру та правила для створення системи. Правильно вибрані фреймворки допоможуть значно спростити і прискорити розробку веб-застосунку.

При створенні власної інтелектуальної комп'ютерної системи прокату велосипедів важливо враховувати її основні переваги, серед яких можна виділити гнучкість у налаштуванні функціоналу відповідно до конкретних потреб та можливість подальшого розширення системи без необхідності повної її перебудови. Така система дозволяє точно адаптувати процеси під особливості роботи сервісу прокату та уникнути зайвих витрат на непотрібні функції, що часто присутні у готових комерційних рішеннях.

Водночас розробка власної системи вимагає значних зусиль для написання коду, проведення тестування, впровадження й постійної підтримки працездатності. До цього слід додати можливі витрати на навчання персоналу, який буде працювати з системою, та забезпечення належного рівня захисту даних.

Приймаючи рішення про створення власної системи, необхідно ретельно оцінити реальні потреби проекту, наявні ресурси та фінансові можливості. Якщо система буде належно реалізована і враховуватиме всі поточні та перспективні вимоги, вона зможе значно покращити якість обслуговування клієнтів, оптимізувати витрати на технічне обслуговування велосипедів і сприятиме розвитку бізнесу в умовах високої конкуренції.

Для розробки своєї системи було обрано об'єктно-орієнтовану мову Python.

Python[20] - це мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Переваги та недоліки мови.

Перваги:

1. Легка читаність коду. При реалізації високонавантажених проєктів, над якими працюють відразу кілька програмістів, читабельність коду грає

вирішальне значення, тому що кожен розробник повинен розуміти, що роблять його колеги.

2. Широка область застосування. Python використовується для розробки онлайн і мобільних додатків, машинного навчання, при створенні ігор і зручна для автоматизації математичних розрахунків. Тому її можна використовувати для вирішення практично будь-яких завдань.
3. Портативність. У багатьох мовах програмування, для запуску програми на різних платформах потрібно вносити зміни в код. Python працює по іншому. Написавши код один раз, ви зможете запускати його на будь-яких пристроях без додаткових коригувань.
4. Велика база бібліотек. Працюючи з Python, розробник не залежить від зовнішніх бібліотек. Мова оснащена потужною стандартною базою функцій, які можна використовувати для вирішення комерційних завдань, що істотно спрощує роботу програміста.
5. Можливість роботи з ШІ.

Недоліки:

1. Невисока швидкість роботи. Python є інтерпретованою мовою. Тобто її код виконується без попередньої компіляції машинної мови. Тому швидкість такого виконання суттєво поступається іншим мовам, що компілюються, таким як C++, Objective-C і т.д. Ще один фактор, що «уповільнює», полягає в тому, що Python - високорівнева мова. Тобто вона більше схожа на людську, ніж на машинну, тому її обробка виконується довше.
2. Неефективне витрачання пам'яті. Python використовує динамічну типізацію, тобто перевірка типів змінних в ній виконується під час виконання, а не в момент оголошення. Це спрощує роботу розробника і робить мову програмування більш гнучкою, але збільшує споживання пам'яті та уповільнює роботу програми.

Враховуючи всі переваги і недоліки Python виявився найкращим вибором для розробки системи прокату велосипедів.

Під час вибору фреймворку для Python у мене було два варіанти –Django та Flask. Це два популярних фреймворка, які призначені для розробки веб додатків. Фреймворки Python також використовуються для розроблення API (Application Programming Interface), які дають змогу взаємодіяти з іншими додатками та сервісами.

Django[21] – один із найвідоміших і повнофункціональних фреймворків Python для веб-розробки. Він надає безліч готових інструментів для створення веб-додатків, включно з керуванням базами даних, аутентифікацією користувачів, маршрутизацією URL і багато іншого. Django також має простий синтаксис і хорошу документацію, що робить його популярним вибором для розробників усіх рівнів.

Flask[22] – це легкий і гнучкий фреймворк Python, призначений для розробки маленьких і середніх веб-додатків. Він надає базові інструменти для створення веб-додатків і дає розробникам велику свободу у виборі бібліотек та інструментів. Flask має мінімалістичний дизайн, завдяки чому підходить для простих і швидких проєктів.

Проаналізувавши їх було обрано Django, так як він здався мені більш гнучким і функціональним. До цього ж я мав досвід роботи з ним.

Отже, враховуючи всі переваги та широкі можливості, які надають Python і фреймворк Django, саме ця комбінація є оптимальним вибором для розробки вебзастосунку, що потребує високої швидкості розробки, зручного обслуговування та легкого розширення функціоналу. Завдяки Django можна швидко реалізувати складну бізнес-логіку, забезпечити безпечну роботу з даними та створити сучасний, надійний і зручний для користувачів вебсервіс, який буде доступний через Інтернет на різних пристроях і платформах.

3 РОЗРОБКА МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ

3.1 Розробка моделі прогнозу технічного стану для велосипедів

У системі прокату велосипедів реалізовано модель, що дозволяє прогнозувати ймовірність технічної несправності транспорту в найближчий період. Модель не просто визначає загальний стан, а оцінює ризик відмови на основі реальних параметрів, зібраних під час експлуатації. Це дозволяє вчасно виконувати обслуговування велосипедів та запобігати серйозним поломкам до моменту їх фактичного виникнення.

3.1.1 Оцінка ймовірності відмови велосипеда згідно теорії надійності

У сфері велопрокату важливо оцінювати надійність велосипедів для забезпечення безпеки користувачів та ефективного управління парком. Теорія надійності дозволяє аналізувати ймовірність відмови системи на основі ймовірностей відмов її компонентів.

Основні компоненти, схильні до відмов[23]:

1. Шини та камери: Проколи та зношення є найпоширенішими проблемами.
2. Гальмівна система: Знос колодок і тросів може призвести до зниження ефективності гальмування.
3. Ланцюг та трансмісія: Зношення ланцюга та зірочок впливає на передачу зусилля.
4. Підшипники: Зношення в каретці, втулках та рульовій колонці може спричинити люфти.
5. Рама та вилка: Можливі тріщини або деформації внаслідок перевантажень або ударів.

Розрахунок ймовірності відмови велосипеда[24]:

Якщо вважати, що відмови компонентів є незалежними подіями, то ймовірність безвідмовної роботи велосипеда (P) дорівнює добутку ймовірностей безвідмовної роботи кожного компонента:

$$P = P_1 \times P_2 \times \dots \times P_n,$$

де $P_1, P_2, \dots, P_n$ - ймовірності безвідмовної роботи окремих компонентів.

Відповідно, ймовірність відмови велосипеда (Q) визначається як:

$$Q = 1 - P$$

Середньостатистичні значення ймовірностей відмов компонентів:

Компоненти і ймовірність відмови за 1000 км

- шини та камери: 0.05
- гальмівна система: 0.02
- ланцюг та трансмісія: 0.03
- підшипники: 0.01
- рама та вилка: 0.005

Ці значення є орієнтовними і можуть варіюватися залежно від умов експлуатації, якості компонентів та обслуговування.

Для ефективного моніторингу стану велосипедів та аналізу причин відмов доцільно впровадити систему логування поїздок. Кожен запис повинен містити:

- ідентифікатор поїздки;
- дата та час початку і завершення;
- ідентифікатор велосипеда;

- ідентифікатор користувача.
- пройдена відстань;
- середня швидкість;
- інформація про виявлені несправності або аномалії.

Застосування теорії надійності в управлінні велопарком дозволяє прогнозувати відмови, оптимізувати обслуговування та підвищити безпеку користувачів. Інтеграція системи логування поїздок забезпечує необхідні дані для аналізу та прийняття обґрунтованих рішень.

3.1.2 Постановка задачі класифікації технічного стану велосипеда

Для забезпечення надійності прокатного сервісу та своєчасного технічного обслуговування велосипедів доцільно застосовувати методи машинного навчання для класифікації[25] технічного стану транспортних засобів. Задача полягає у визначенні, чи є велосипед придатним до подальшої експлуатації, або ж він потребує технічного втручання.

Мета класифікації - автоматично присвоїти кожному велосипеду один із двох класів:

- клас 0 - "придатний до використання";
- клас 1 - "непридатний до використання".

Вхідні ознаки (features):

кількість відмов за останні 1000 км пробігу для кожної з основних деталей:

- гальмівна система;
- ланцюг і трансмісія;
- шини;
- підшипники;
- вік велосипеда (у днях);

- кількість днів з останнього обслуговування;
- погодні умови під час експлуатації (дощ/сухо, температура, вологість);
- середня тривалість поїздок;
- загальне навантаження за останні 1000 км (наприклад, кількість поїздок).

Цільова змінна (target):

1. бінарна змінна `needs_service` (0 - не потребує обслуговування, 1 - потребує).

Загальна формалізація:

Нехай $X \in \mathbb{R}^n$ - вектор ознак для конкретного велосипеда, тоді задача полягає у побудові функції класифікації:

$$f(X) \rightarrow \{0, 1\},$$

де 0 - велосипед справний, 1 - несправний.

Особливості задачі:

1. Даний тип класифікації можна віднести до задач контролю якості в умовах експлуатації.
2. Оцінка якості класифікатора буде базуватись на метриках точності[25] (accuracy), повноти (recall) та F1-міри.
3. Через можливу дисбалансованість класів (більшість велосипедів часто перебуває у справному стані), рекомендовано застосовувати методи балансування вибірки або використання алгоритмів, стійких до дисбалансу (наприклад, RandomForest, XGBoost із параметром `scale_pos_weight`).

3.1.3 Обґрунтування вибору набору даних для навчання моделі

Щоб навчити модель прогнозувати технічний стан велосипеда, потрібно мати набір даних, у якому є реальна інформація про ремонти, пробіг, технічне обслуговування та умови експлуатації. На жаль, знайти такий готовий набір даних не вдалося.

Доступна тільки загальна інформація. Наприклад, за даними Bike in City[26], велосипед в середньому служить близько 5 років і за цей час проїжджає приблизно 15 000 км. Але там не вказується, коли саме і які деталі виходять з ладу. Також немає даних про те, як часто ці велосипеди ремонтувались і за яких умов експлуатувались.

Стосовно ДТП, у соцмережі Facebook на сторінці U-Cycle[27] вказано, що лише у 2022 році в Україні зафіксовано 1 502 наїзди на велосипедистів. Але ніде не сказано, що саме стало причиною цих пригод - технічна несправність, людський фактор чи інше.

На момент розробки моделі прогнозування технічного стану велосипедів у відкритому доступі не існує структурованих наборів даних, які містили б історію відмов компонентів велосипедів у поєднанні з супутніми факторами — пробігом, погодою, обслуговуванням тощо. Наявні лише загальні аналітичні звіти, що фіксують кількість ремонтів або аварій за певні періоди, однак ці джерела не дають змоги сформувати навчальну вибірку для моделі машинного навчання.

З огляду на це було прийнято рішення сформувати штучно згенерований набір даних, який відображає реальні сценарії експлуатації велосипедів. Було створено 10 000 прикладів, у яких збережено основні характеристики, що впливають на технічний стан:

- failures_brakes_last_1000km, failures_chain_last_1000km, failures_tires_last_1000km, failures_shifter_last_1000km — кількість відмов окремих компонентів за останні 1000 км;

- rides_in_rain — кількість поїздки під дощем;
- avg_temperature, avg_humidity — погодні умови, за яких використовувався велосипед;
- failures_last_year — кількість поломок за останній рік;
- technical_service_done — чи проводилось технічне обслуговування;
- bike_age_days, total_mileage — вік велосипеда та його загальний пробіг;
- target — цільова змінна, яка вказує на наявність або відсутність ризику поломки найближчим часом.

Таке моделювання дозволило підготувати повноцінну тренувальну вибірку для алгоритму класифікації, який оцінює ймовірність майбутніх відмов. Використання штучного набору даних на початковому етапі є доцільним рішенням, яке дозволяє запускати систему без очікування великого обсягу реальних даних, а в подальшому — донавчати модель на реальних кейсах, що накопичуються в процесі роботи платформи.

На рисунку 3.1 відображений дата сет який використовувався при тренуванні моделі ІІІ.

	failures_brakes_last_10	failures_chain_last_100	failures_tires_last_1000	failures_shifter_last_10	rides_in_rain	avg_temperature	avg_humidity	failures_last_year	technical_service_done	bike_age_days	total_mileage	target
1	2	2	2	3	5	13.403605890041643	61	2	1	1324	3230	1
2	3	3	3	1	1	24.787218256154556	56	3	1	904	2716	1
3	0	2	3	0	3	24.74524278541371	53	5	1	1489	1082	1
4	2	3	1	3	4	26.150224223622583	67	0	0	700	811	1
5	2	3	0	0	4	34.77395236333277	96	4	0	123	4396	1
6	3	2	3	3	2	16.60093129042216	45	1	1	1026	3758	1
7	0	1	1	3	5	18.959464234553028	44	1	1	365	2384	1
8	0	3	2	1	0	29.138351030760226	54	0	0	841	2626	1
9	2	2	2	2	2	22.35110069065656	33	2	1	193	1636	1
10	1	0	2	2	0	6.117716320633989	61	3	1	923	1990	1

Рисунок 3.1 - Дата сет зі згенерованими даними

Цей підхід дозволив швидко отримати реалістичний набір даних і протестувати роботу моделі без потреби чекати на накопичення реальних записів під час експлуатації системи.

3.1.4 Варіантний аналіз моделей для розв'язання задачі класифікації

У задачі класифікації технічного стану велосипеда потрібно на основі вхідних даних (пробіг, ремонт, вік велосипеда тощо) передбачити, чи є ймовірність поломки в найближчі N км. Для цього використовуються моделі машинного навчання, які вміють ділити об'єкти на два класи: придатний або непридатний.

Найпростішим варіантом є дерево рішень (Decision Tree)[28]. Воно приймає рішення на основі простих "якщо-то" умов. Наприклад, якщо пробіг понад 3000 км і давно не було ремонту - велосипед вважається непридатним. Ця модель добре підходить для невеликих задач і її легко візуалізувати, але вона може "переучуватись", тобто показувати хороші результати на тренувальних даних і погані на нових.

Більш точним і стабільним є випадкові дерева (Random Forest)[29] - це набір із багатьох дерев рішень, які працюють разом. Кожне дерево "голосує", і фінальне рішення приймається більшістю. Такий підхід дає високу точність і краще працює з випадковими шумами у даних.

Ще один підхід - градієнтний бустинг[30] (наприклад, XGBoost або LightGBM). Тут дерева навчаються одне за одним, і кожне нове дерево старається виправити помилки попереднього. Модель довше тренується, але часто показує найкращу точність. Потрібно лише правильно підібрати параметри, щоб уникнути перенавчання.

Також можна використати нейронні мережі. Наприклад, простий MLP[31] (Multilayer Perceptron). Вони імітують роботу мозку й добре працюють зі складними залежностями між ознаками. Але для них потрібно більше часу на навчання і більше даних. До того ж, результат такої моделі важче пояснити.

Для невеликих задач добре працює і метод опорних векторів (SVM)[32]. Він розділяє два класи у просторі ознак гіперплощиною. Цей метод точний, але повільний на великих об'ємах даних.

Після аналізу було вирішено обрати три моделі для експериментів:

1. Random Forest - як надійна, проста в реалізації і точна модель.
2. XGBoost - як потужна модель, яка часто дає кращі результати.
3. MLP (нейронна мережа) - щоб порівняти класичний підхід з гнучким варіантом.

Ці три варіанти дозволяють оцінити різні підходи: класичні дерева, ансамблеві методи й нейронні мережі.

3.1.5 Навчання та тестування моделей

Для ілюстрації процесу навчання використаємо класифікаційну модель `XGBClassifier` з бібліотеки `xgboost`, яка показала високу якість на задачах, де присутні складні залежності між ознаками. Алгоритм працює за принципом градієнтного бустингу по ансамблю дерев рішень і здатний точно моделювати ймовірність відмов навіть при обмеженій кількості даних.

Код тренувальння моделі:

```

1 import pandas as pd
2 import joblib
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score
5 from xgboost import XGBClassifier
6
7
8 df = pd.read_csv("D:/IHEAVEN/retheaven/diagnostics/datasets/bike_failures/failure_dataset_future_10000.csv")
9
10
11 X = df.drop(columns=["target"])
12 y = df["target"]
13
14
15 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
16
17 model = XGBClassifier(
18     use_label_encoder=False,
19     eval_metric='logloss',
20     n_estimators=300,
21     max_depth=4,
22     random_state=42
23 )
24 model.fit(X_train, y_train)
25
26
27 y_pred = model.predict(X_test)
28
29
30 print("📊 Критерії якості моделі:")
31 print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
32 print(f"Precision: {precision_score(y_test, y_pred):.4f}")
33 print(f"Recall: {recall_score(y_test, y_pred):.4f}")
34 print(f"F1 Score: {f1_score(y_test, y_pred):.4f}")
35 print("\n📄 Звіт про класифікацію:")
36 print(classification_report(y_test, y_pred))
37
38
39 joblib.dump(model, "D:/IHEAVEN/retheaven/diagnostics/bike_ai/failure_predictor_model.pkl")
40 print("📁 Модель збережено.")
41

```

Рисунок 3.2 – Тренувальна програма для прогнозу технічного обслуговування

Завантаження всіх необхідних модулів для машинного навчання, препроцесингу та збереження результатів.

```

1 import pandas as pd
2 import joblib
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score
5 from xgboost import XGBClassifier

```

Рисунок 3.3 – Завантаження бібліотек

Виконується зчитування набору даних у форматі .csv, який містить вхідні ознаки та цільову змінну (target).

```

8 df = pd.read_csv("D:/!HEAVEN/rentheaven/diagnostics/datasets/bike_failures/failure_dataset_future_10000.csv")
9

```

Рисунок 3.4 – Завантаження даних

Із таблиці видаляється колонка target (цільова змінна) — вона зберігається окремо як y, а всі інші колонки — це ознаки X.

```

10
11 X = df.drop(columns=["target"])
12 y = df["target"]
13

```

Рисунок 3.5 – Відокремлення ознак і цільового значення

Дані діляться у співвідношенні 80/20 для навчання та тестування. random_state=42 забезпечує стабільність результатів.

```

14
15 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
16

```

Рисунок 3.6 – Розділення на тренувальні і тестові дані

Створюється модель XGBoost з налаштуваннями: n_estimators=300 - кількість дерев, max_depth=4 - максимальна глибина дерева, eval_metric="logloss" - метрика логарифмічної втрати, use_label_encoder=False - вимикає старий формат кодування. Далі - навчання моделі.

```

17 model = XGBClassifier(
18     use_label_encoder=False,
19     eval_metric='logloss',
20     n_estimators=300,
21     max_depth=4,
22     random_state=42
23 )
24 model.fit(X_train, y_train)
25

```

Рисунок 3.7 - Створення та навчання моделі

Генерується прогноз (класи) для тестової вибірки. Далі обчислюються й виводяться основні метрики класифікації[25]: Точність(Accuracy), Влучність(Precision), Повнота(Recall), Середнє гармонійне(F1 - міра). Потім — повний текстовий звіт із класифікаційними результатами по кожному класу.

```

27 y_pred = model.predict(X_test)
28
29
30 print("📊 Критерії якості моделі:")
31 print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
32 print(f"Precision: {precision_score(y_test, y_pred):.4f}")
33 print(f"Recall: {recall_score(y_test, y_pred):.4f}")
34 print(f"F1 Score: {f1_score(y_test, y_pred):.4f}")
35 print("\n📄 Повний звіт:")
36 print(classification_report(y_test, y_pred))
37

```

Рисунок 3.8 – Прогноз на тестовій вибірці та виведення метрик

Навчена модель зберігається у файл .pkl, щоб у майбутньому її можна було завантажити без повторного тренування.

```

18
19 joblib.dump(model, "D:/!HEAVEN/rentheaven/diagnostics/bike_ai/failure_predictor_model.pkl")
20 print("✅ Модель збережено.")
21

```

Рисунок 3.9 – Збереження моделі

Уся логіка побудована так, щоб модель була незалежна, багаторазово використовувана і проста для повторного тренування при оновленні даних.

Для оцінки якості моделей, обраних для прогнозування технічного стану велосипеда, було проведено експеримент із використанням методу перехресної валідації (крос-валідації) на 5 фолдах. Суть цього методу полягає в тому, що вихідні дані поділяються на п'ять рівних частин: чотири з них використовуються для навчання моделі, а одна - для перевірки. Цей процес повторюється п'ять разів так, щоб кожна частина один раз стала тестовою.

Для задачі класифікації обрано F1-метрику, яка є середньозваженим значенням точності та повноти. Вона особливо корисна, коли важливо не лише правильно виявити проблемні велосипеди, а й мінімізувати кількість хибних спрацювань. У нашому випадку метрика F1 розраховувалась для класу “непридатний” - тобто для виявлення велосипедів, у яких очікується поломка на найближчі 200 км.

У тестуванні брали участь три моделі:

1. Random Forest Classifier - класична ансамблева модель на основі дерев рішень.
2. XGBoost Classifier - більш складна модель градієнтного бустингу, що має високу точність.
3. MLP (Multilayer Perceptron) - нейронна мережа, яка дозволяє враховувати складні залежності між ознаками.

```

=== Крос-валідація (F1-метрика, 5 фолдів) ===

Random Forest:
  F1-оцінки: [0.9524838  0.95276248 0.95303131 0.9567536  0.9572237 ]
  Середнє F1: 0.9545
-----

XGBoost:
  F1-оцінки: [0.98295876 0.98372514 0.98463056 0.98551679 0.98576952]
  Середнє F1: 0.9845
-----

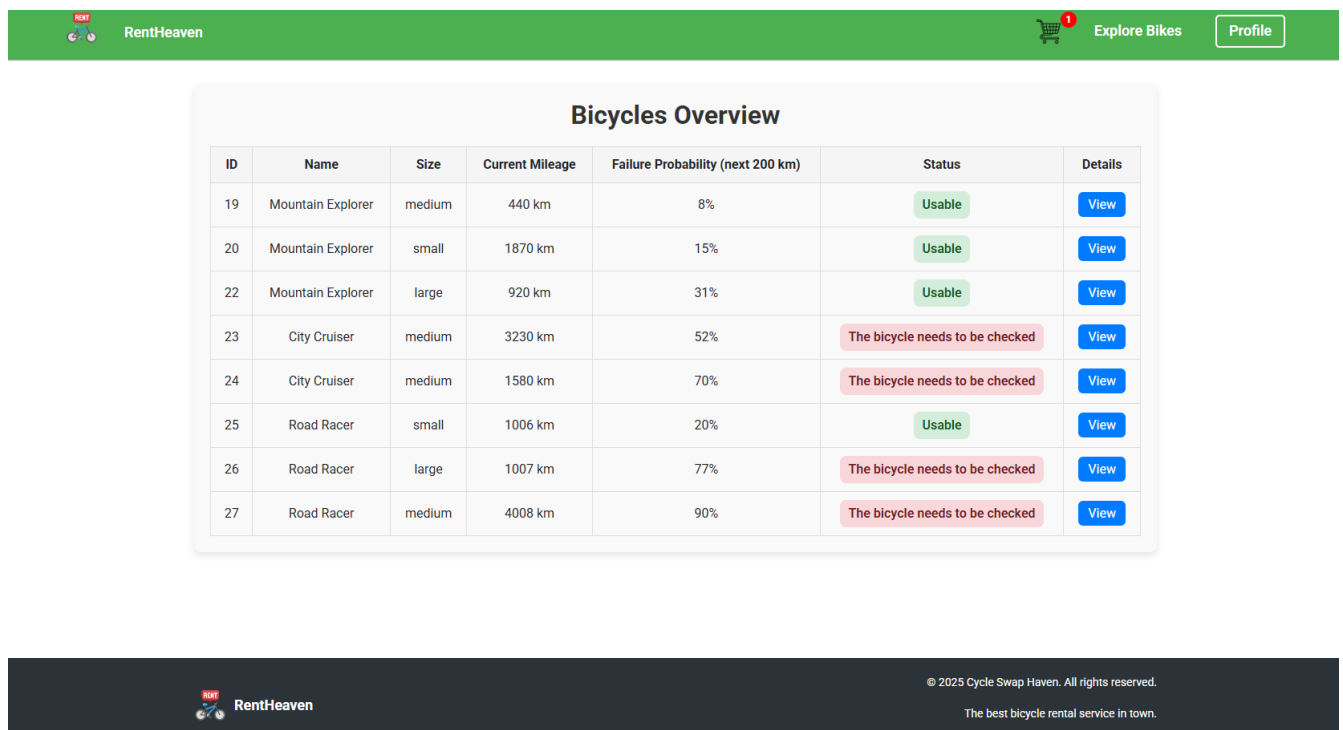
MLP:
  F1-оцінки: [0.97487437 0.96399263 0.87775953 0.89566237 0.86025916]
  Середнє F1: 0.9145
-----
  
```

Рисунок 3.10 - Результати крос-валідації (F1-метрика)

Як видно з таблиці, найкращий результат показала модель XGBoost, що є очікуваним, оскільки вона ефективно навчається на структурованих табличних даних і здатна адаптуватися до складних закономірностей. Модель Random Forest продемонструвала стабільно високі результати, а також є більш простою в налаштуванні та інтерпретації. Нейронна мережа MLP виявилася менш стабільною - значення F1 мали більший розкид залежно від фолду, хоча загальний результат залишається прийнятним.

У підсумку, для подальшого використання було обрано модель XGBoost, оскільки вона показала найвищу середню точність.

На рисунку 3.11 відображено приклад прогнозу ймовірність відмови транспортного засобу на наступні 200 кілометрів.



ID	Name	Size	Current Mileage	Failure Probability (next 200 km)	Status	Details
19	Mountain Explorer	medium	440 km	8%	Usable	View
20	Mountain Explorer	small	1870 km	15%	Usable	View
22	Mountain Explorer	large	920 km	31%	Usable	View
23	City Cruiser	medium	3230 km	52%	The bicycle needs to be checked	View
24	City Cruiser	medium	1580 km	70%	The bicycle needs to be checked	View
25	Road Racer	small	1006 km	20%	Usable	View
26	Road Racer	large	1007 km	77%	The bicycle needs to be checked	View
27	Road Racer	medium	4008 km	90%	The bicycle needs to be checked	View

Рисунок 3.11 - Приклад прогнозу ймовірність відмови транспортного засобу

3.2 Розробка моделі прогнозу попиту оренди велосипедів

Прогнозування попиту на оренду велосипедів дозволяє ефективно розподіляти транспорт між точками видачі, уникати дефіциту в пікові години та краще планувати обслуговування. Особливо актуально це для систем, де велосипеди розміщуються в організованих пунктах прокату, а не на вулиці у випадкових місцях. З огляду на це, в межах системи було реалізовано модель прогнозу, яка передбачає кількість оренд на певну дату з урахуванням погодних та календарних факторів.

3.2.1 Постановка задачі прогнозування попиту оренди велосипедів

Задача прогнозування попиту оренди велосипедів формалізується як задача регресійного аналізу, основною метою якої є передбачення кількості оренд на певну дату на основі метеоумов та календарних характеристик. На відміну від задачі класифікації, де результат обмежений певними класами, регресійна модель дозволяє отримати точне числове значення очікуваного попиту.

Попит на велосипедну оренду має низку специфічних особливостей:

1. Сезонний характер: у зимові місяці попит практично відсутній, тому аналіз обмежується періодом березень–листопад.
2. Залежність від погоди: найвищий попит фіксується за умов ясної, сухої погоди при температурі від 20 до 30°C. За дощу або знижених температур кількість оренд суттєво зменшується.
3. Календарні чинники: попит залежить від типу дня (робочий/вихідний), сезону, святкових дат, а також може змінюватися залежно від дня тижня.

Прогнозування виконується виключно для міста Вінниця, а модель орієнтована на короткотерміновий горизонт прогнозу - до 7 днів наперед, із щоденним оновленням погодних даних через OpenWeather API.

Формально, вхідними параметрами моделі є:

- температура (temp)
- вологість (humidity)
- швидкість вітру (windspeed)
- сезон (season)
- святковий день (holiday)
- робочий день (workingday)
- погодний код (weather)

Цільовою змінною (target) є кількість оренд велосипедів у межах одного дня.

Завданням моделі є забезпечення точного прогнозу попиту з метою:

- оптимізації кількості доступних велосипедів;
- планування перевезень і обслуговування;
- мінімізації випадків відмови через відсутність вільного транспорту.

3.2.2 Вибір вхідних ознак для моделі прогнозу попиту

Для побудови ефективної моделі прогнозування попиту на оренду велосипедів було відібрано набір ознак, які прямо або опосередковано впливають на рішення потенційного користувача щодо оренди транспортного засобу. Врахування саме цих факторів дозволяє враховувати змінні зовнішнього середовища, що мають статистично підтверджений вплив на динаміку попиту.

Основні ознаки, використані у моделі:

Температура (temp) - приведена до відносного масштабу шляхом ділення на 41 (максимальне значення в наборі), що дозволяє моделі однаково ефективно працювати з різними температурними діапазонами. Високі температури в межах 20–30 °C зазвичай корелюють із високим попитом.

Вологість (humidity) - виражена у відсотках. Занадто висока вологість у поєднанні з іншими факторами (наприклад, похмурістю) часто сигналізує про ризик дощу, що призводить до зменшення попиту.

Швидкість вітру (windspeed) - нормалізована діленням на 67. Надмірно сильний вітер може зробити поїздку незручною, тому ця ознака також враховується.

Сезон (season) - категоріальна ознака, яка кодується цілими числами (наприклад, 1 - весна, 2 - літо). Врахування сезону дозволяє моделі орієнтуватись на зміни в попиті, зумовлені не лише погодою, а й загальними тенденціями впродовж року.

Святковий день (holiday) - бінарна ознака (0 або 1), що вказує, чи є дата святковою. У такі дні може бути зафіксовано як зростання попиту (через дозвілля), так і зменшення (через несприятливу погоду або відсутність необхідності пересування).

Робочий день (workingday) - також бінарна ознака, яка сигналізує про тип дня. Поєднання цієї ознаки з holiday дозволяє відокремити типові будні від вікендів і свят.

Погодний код (weather) - узагальнена класифікація погодних умов:

- 1 - ясно;
- 2 - хмарно або туман;
- 3 - дощ або сніг;
- 4 - інші або екстремальні умови.

Ці ознаки обрані на основі загальних тенденцій ринку прокату велосипедів, аналізу відкритих джерел та логіки користувацької поведінки. Такий набір дозволяє моделі враховувати не лише миттєвий стан погоди, а й контекст (тип дня, сезон), що істотно підвищує точність прогнозу.

3.2.3 Побудова та обробка набору даних для навчання

Оскільки відкриті набори даних, що містять історичні дані про попит на оренду велосипедів у місті Вінниця, відсутні, було прийнято рішення про створення штучного набору даних на основі типових погодних сценаріїв та очікуваної поведінки користувачів. Для цього було реалізовано генератор даних, у якому враховувалися ключові чинники, що впливають на попит, зокрема температура, вологість, погодні умови, сезонність і тип дня.

Дані згенеровані у вигляді таблиці у форматі CSV, кожен рядок якої описує один день. Структура таблиці включає такі стовпці:

- temp - температура повітря (у °C);
- humidity - вологість (%);
- windspeed - швидкість вітру (м/с);
- season - сезон (1 - весна, 2 - літо, тощо);
- holiday - чи є день святковим (0 або 1);
- workingday - чи є день робочим (0 або 1);
- weather - погодний код;
- count - цільова змінна: згенерована кількість оренд у цей день.

Для реалізації тренування моделі у скрипті виконується попередня обробка даних:

- нормалізація ознак за допомогою StandardScaler[33] для усунення впливу різних масштабів вимірювання;
- розподіл даних на тренувальну та тестову вибірки у співвідношенні 80/20;
- навчання моделі на базі нормалізованих ознак.

Створений набір даних повністю відповідає очікуваним характеристикам попиту у місті Вінниця. Зокрема, у дні з ясною погодою та температурою понад 20 °C генерується високий попит (до 30 оренд на день), у дощову погоду - від 0 до

2, у похмуру або туманну - проміжні значення. Зимовий період не включено у набір даних, оскільки в цей час оренда велосипедів практично не здійснюється.

Таким чином, підготовлений набір даних є збалансованим і дозволяє адекватно навчити модель для подальшого застосування на реальних погодних прогнозах.

Вигляд початку дата сету можна побачити на рисунку 3.12.

	datetime	season	holiday	workingday	weather	temp	humidity	windspeed	count
1	2022-03-01 10:00:00	1	0	1	1	24.6	40	4.3	0
2	2022-03-01 11:00:00	1	0	1	1	24.6	40	4.3	1
3	2022-03-01 12:00:00	1	0	1	1	24.6	40	4.3	2
4	2022-03-01 13:00:00	1	0	1	1	24.6	40	4.3	6
5	2022-03-01 14:00:00	1	0	1	1	24.6	40	4.3	2
6	2022-03-01 15:00:00	1	0	1	1	24.6	40	4.3	8
7	2022-03-01 16:00:00	1	0	1	1	24.6	40	4.3	3
8	2022-03-01 17:00:00	1	0	1	1	24.6	40	4.3	1
9	2022-03-01 18:00:00	1	0	1	1	24.6	40	4.3	8
10	2022-03-01 19:00:00	1	0	1	1	24.6	40	4.3	0

Рисунок 3.12 - Дата сет для прогнозування попиту

3.2.4 Критерії якості регресійних моделей

Щоб оцінити, наскільки точно модель передбачає значення, використовуються спеціальні кількісні показники - критерії якості. Для регресійних задач важливо не лише отримати передбачене число, але й знати, наскільки воно близьке до фактичного значення. Тому результати моделі оцінюються за допомогою регресійних метрик, кожна з яких показує похибку прогнозу з різних боків.

Коефіцієнт детермінації (R^2)

Цей[25] показник показує, наскільки добре модель пояснює змінність залежної змінної. Його значення коливається від 0 до 1. Якщо $R^2 = 1$ - модель точно відтворює всі реальні значення, якщо $R^2 = 0$ - модель нічим не краща за просте середнє. R^2 вимірює частку дисперсії, яку пояснює модель. Формула:

$$R^2 = 1 - \frac{SS_e}{SS_t}$$

де:

SS_e - сума квадратів помилок (відхилення між передбаченим і реальним значенням),

SS_t - загальна сума квадратів (відхилення реальних значень від середнього).

Середня абсолютна похибка (MAE)

MAE(Mean Absolute Error)[25] - це середнє значення абсолютних відхилень між реальними та передбаченими значеннями. Вона показує, на скільки одиниць модель у середньому помиляється.

Формула:

$$MAE = \frac{1}{n} \sum |y - \hat{y}|$$

де:

n - кількість прикладів,

y - реальне значення,

$\hat{y}$ - передбачене значення.

Середньоквадратична похибка (RMSE)

RMSE(Root Mean Square Error)[25] враховує квадрати відхилень, а потім бере квадратний корінь із середнього. Ця метрика чутливіша до великих помилок, ніж MAE Формула:

$$MSE = \sqrt{\frac{1}{n} \sum (y - \hat{y})^2}$$

де:

- n - кількість прикладів,
- y - реальне значення,
- $\hat{y}$ - передбачене значення.

Узагальнення результатів оцінки моделі на основі кількох метрик дозволяє точніше визначити її придатність до використання. Поєднання абсолютних і відносних показників, таких як MAE, RMSE та R^2 , дає змогу не лише порівняти точність прогнозу, а й виявити чутливість моделі до відхилень. Такий підхід забезпечує об'єктивну оцінку якості побудованої регресійної моделі та дає основу для її подальшого вдосконалення.

3.2.5 Варіантний аналіз моделей для вирішення задачі регресії

Для прогнозування попиту на оренду велосипедів було протестовано кілька поширених регресійних моделей. Мета - оцінити їхню здатність враховувати вплив погодних та календарних факторів на кількість оренд. Нижче наведено короткий опис кожної з моделей та порівняльний аналіз їх ефективності.

Лінійна регресія (Linear Regression)[25] - це найпростіша модель, що описує залежність цільової змінної від набору ознак за допомогою прямої. Перевагою є швидкість навчання та інтерпретованість, проте вона не враховує нелінійні залежності.

Random Forest Regressor[25] – це ансамблева модель, що поєднує багато дерев рішень. Завдяки використанню випадковості та комбінації прогнозів, вона зменшує ймовірність перенавчання і дає стабільні результати.

XGBRegressor[25] - модель градієнтного бустингу, яка створює послідовність дерев рішень, кожне з яких виправляє помилки попередніх.

Для оцінки якості моделей, побудованих для прогнозування попиту на оренду велосипедів, було застосовано метод перехресної валідації (крос-валідації) з поділом на 5 фолдів. Такий підхід передбачає розбиття вихідного набору даних на п'ять рівних частин, у межах яких модель навчається на чотирьох підмножинах і тестується на п'ятій. Процедура повторюється п'ять разів, що дозволяє оцінити узагальнювальну здатність моделі та її стійкість до різних варіацій у даних.

Для кожної моделі регресії обчислювались показники середньої абсолютної помилки (MAE), кореня середньоквадратичної помилки (RMSE), а також коефіцієнта детермінації (R^2), які дозволяють комплексно оцінити точність прогнозування. На рисунку 3.13 показано результат крос валідації.

```

=== Крос-валідація (5 фолдів) ===

Linear Regression:
MAE: [2.76, 2.62, 2.98, 2.66, 2.87] → Середнє: 2.78
RMSE: [np.float64(3.4), np.float64(3.24), np.float64(3.49), np.float64(3.16), np.float64(3.44)] → Середнє: 3.35
R²: [0.913, 0.922, 0.916, 0.934, 0.923] → Середнє: 0.922
-----
Random Forest:
MAE: [1.94, 1.89, 1.7, 1.83, 1.79] → Середнє: 1.83
RMSE: [np.float64(2.31), np.float64(2.27), np.float64(2.07), np.float64(2.19), np.float64(2.16)] → Середнє: 2.20
R²: [0.96, 0.962, 0.97, 0.968, 0.97] → Середнє: 0.966
-----
XGBoost:
MAE: [1.95, 1.82, 1.71, 1.81, 1.82] → Середнє: 1.82
RMSE: [np.float64(2.33), np.float64(2.26), np.float64(2.05), np.float64(2.17), np.float64(2.19)] → Середнє: 2.20
R²: [0.959, 0.962, 0.971, 0.969, 0.969] → Середнє: 0.966
-----

```

Рисунок 3.13 - Результат крос валідації для прогнозу попиту оренди.

За результатами п'ятифолдної крос-валідації для оцінки моделей прогнозування попиту було порівняно три алгоритми: Linear Regression, Random Forest та XGBoost. Найнижчу середню абсолютну помилку ($MAE = 1.82$) показала модель XGBoost, що свідчить про її точність у прогнозуванні індивідуальних

значень. Водночас середній показник R^2 для XGBoost склав 0.966, що відповідає рівню Random Forest і перевищує результат лінійної регресії. Значення RMSE для XGBoost і Random Forest також практично однакові (2.20), що підтверджує високий рівень точності обох моделей.

З огляду на найнижче значення MAE та рівноцінний R^2 , модель XGBoost визначено як найбільш ефективну для задачі прогнозування попиту на оренду велосипедів у рамках даного дослідження.

3.2.6 Навчання моделей та оцінювання якості прогнозу

Навчання моделі прогнозування попиту на оренду велосипедів здійснювалося на основі згенерованого погодинного набору даних, який охоплює період з березня 2022 року по листопад 2024 року. Кожен запис у цьому наборі відповідає одній годині активного прокатного періоду (з 10:00 до 19:00) та містить метеорологічні показники (температура, вологість, швидкість вітру), тип дня, сезон, погодний код і кількість оренд у цю годину. Такий підхід дозволив врахувати як щоденні, так і внутрішньоденні закономірності в структурі попиту.

Перед подачею до моделі дані були агреговані по днях, тобто для кожної дати обчислювалися середні значення метеопараметрів, найтипівіший погодний стан, а також сумарна кількість оренд за день. Це дозволило адаптувати погодинні дані до формату щоденного прогнозу без втрати ключових тенденцій.

Для реалізації цієї задачі було використано модель XGBRegressor з бібліотеки XGBoost, яка належить до градієнтного бустингу над деревами рішень. Цей тип моделі обрано через його високу точність, здатність вловлювати складні взаємозв'язки між змінними та хорошу генералізацію на нових даних. На відміну від звичайної лінійної регресії, XGBoost дозволяє враховувати нелінійні залежності між ознаками, які часто зустрічаються у задачах, пов'язаних з погодою або людською поведінкою.

Навчання проводилось на згенерованому набору даних, який охоплює дані з березня по листопад, оскільки саме в цей період прокат є активним. Було використано 80% даних для тренування і 20% - для перевірки. Перед навчанням дані масштабувались за допомогою StandardScaler[33].

Код тренувальної моделі:

```

1 import pandas as pd
2 import joblib
3 import os
4 from xgboost import XGBRegressor
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import StandardScaler
7 from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
8 import numpy as np
9
10
11 BASE_DIR = os.path.dirname(os.path.abspath(__file__))
12 DATA_PATH = os.path.join(BASE_DIR, '../datasets/bike_sharing/generated_demand_hourly_contrast.csv')
13 MODEL_PATH = os.path.join(BASE_DIR, 'demand_model.pkl')
14 R2_PATH = os.path.join(BASE_DIR, 'demand_r2.txt')
15
16
17 df = pd.read_csv(DATA_PATH)
18
19
20 df['date'] = pd.to_datetime(df['datetime']).dt.date
21 df_daily = df.groupby('date').agg([
22     'temp': 'mean',
23     'humidity': 'mean',
24     'windspeed': 'mean',
25     'season': 'first',
26     'holiday': 'first',
27     'workingday': 'first',
28     'weather': lambda x: x.mode()[0],
29     'count': 'sum'
30 ]).reset_index()
31
32 features = df_daily[['temp', 'humidity', 'windspeed', 'season', 'holiday', 'workingday', 'weather']]
33 target = df_daily['count']
34
35 scaler = StandardScaler()
36 features_scaled = scaler.fit_transform(features)
37
38 X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.2, random_state=42)
39
40
41 model = XGBRegressor(
42     objective='reg:squarederror',
43     n_estimators=300,
44     max_depth=6,
45     learning_rate=0.05,
46     random_state=42
47 )
48 model.fit(X_train, y_train)
49
50 y_pred = model.predict(X_test)
51 r2 = r2_score(y_test, y_pred)
52 mae = mean_absolute_error(y_test, y_pred)
53 mse = mean_squared_error(y_test, y_pred)
54 rmse = np.sqrt(mse)
55
56 print(f"R² Score: {r2:.3f}")
57 print(f"MAE: {mae:.3f}")
58 print(f"RMSE: {rmse:.3f}")
59
60 joblib.dump(model, MODEL_PATH)
61 with open(R2_PATH, 'w', encoding='utf-8') as f:
62     f.write(f"R2: {r2:.3f}\nMAE: {mae:.3f}\nMSE: {mse:.3f}\nRMSE: {rmse:.3f}")
63
64 print(f"Модель збережена за адресою: {MODEL_PATH}")
65

```

Рисунок 3.14 -Тренувальна програма для прогнозу попиту оренди

Імпорт потрібних для тренування бібліотек.

```

1  import pandas as pd
2  import joblib
3  import os
4  from xgboost import XGBRegressor
5  from sklearn.model_selection import train_test_split
6  from sklearn.preprocessing import StandardScaler
7  from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
8  import numpy as np
9

```

Рисунок 3.15 – Імпорт бібліотек

Налаштування шляхів до файлів. Встановлюються абсолютні шляхи до набору даних, збереження моделей.

```

10
11  BASE_DIR = os.path.dirname(os.path.abspath(__file__))
12  DATA_PATH = os.path.join(BASE_DIR, '../datasets/bike_sharing/generated_demand_hourly_contrast.csv')
13  MODEL_PATH = os.path.join(BASE_DIR, 'demand_model.pkl')
14  R2_PATH = os.path.join(BASE_DIR, 'demand_r2.txt')
15

```

Рисунок 3.16 – Налаштування шляхів

Завантаження та перетворення погодинних даних на щоденні, групуючи записи за датою та обчислюючи середні, перші або найчастіші значення для кожного дня.

```

17  df = pd.read_csv(DATA_PATH)
18
19
20  df['date'] = pd.to_datetime(df['datetime']).dt.date
21  df_daily = df.groupby('date').agg({
22      'temp': 'mean',
23      'humidity': 'mean',
24      'windspeed': 'mean',
25      'season': 'first',
26      'holiday': 'first',
27      'workingday': 'first',
28      'weather': lambda x: x.mode()[0],
29      'count': 'sum'
30  }).reset_index()
31

```

Рисунок 3.17 – Групування даних

Завантаження і підготовка даних. Зчитуються дані з набору даних та виділяються:

- features - набір ознак, які впливають на попит;
- target - кількість оренд (count), яку необхідно передбачити.

```
32 features = df_daily[['temp', 'humidity', 'windspeed', 'season', 'holiday', 'workingday', 'weather']]
33 target = df_daily['count']
```

Рисунок 3.18 – Завантаження і підготовка даних

Масштабування ознак. Дані масштабується до середнього 0 і стандартного відхилення 1. Це покращує стабільність і швидкість навчання моделей на основі дерев.

```
22
23 scaler = StandardScaler()
24 features_scaled = scaler.fit_transform(features)
25
```

Рисунок 3.19 – Масштабування ознак

Поділ даних, побудова і навчання моделі. Дані діляться у співвідношенні 80/20. Модель тренується на X_train, а оцінка точності проводиться на X_test. Модель XGBoost навчається на навчальній вибірці. Параметри:

- n_estimators=300 - кількість дерев;
- max_depth=6 - глибина дерева;
- objective='reg:squarederror' - для регресійної задачі;
- learning_rate=0.05 – швидкість навчання;
- random_state=42 - фіксації випадковості.

```
38 X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.2, random_state=42)
39
40
41 model = XGBRegressor(
42     objective='reg:squarederror',
43     n_estimators=300,
44     max_depth=6,
45     learning_rate=0.05,
46     random_state=42
47 )
48 model.fit(X_train, y_train)
```

Рисунок 3.20 – Поділ даних, побудова та навчання моделі

Оцінка якості моделі та вивід у консоль.

```

50 y_pred = model.predict(X_test)
51 r2 = r2_score(y_test, y_pred)
52 mae = mean_absolute_error(y_test, y_pred)
53 mse = mean_squared_error(y_test, y_pred)
54 rmse = np.sqrt(mse)
55
56 print(f"R² Score: {r2:.3f}")
57 print(f"MAE: {mae:.3f}")
58 print(f"RMSE: {rmse:.3f}")

```

Рисунок 3.21 – Оцінка якості моделі

Збереження моделі і точності.

```

60 joblib.dump(model, MODEL_PATH)
61 with open(R2_PATH, 'w', encoding='utf-8') as f:
62     f.write(f"R2: {r2:.3f}\nMAE: {mae:.3f}\nMSE: {mse:.3f}\nRMSE: {rmse:.3f}")
63

```

Рисунок 3.22 – Збереження моделі і точності.

Підготовка ознак включала масштабування числових параметрів за допомогою StandardScaler, що дозволяє уникнути домінування окремих змінних у процесі навчання. Дані було поділено на тренувальну та тестову вибірки у співвідношенні 80/20. Після навчання модель протестовано на незалежній частині вибірки, результати оцінено за чотирма метриками:

- R^2 (коефіцієнт детермінації): 0.966
- MAE (середня абсолютна похибка): 1.82
- RMSE (корінь із середньоквадратичної похибки): 2.20

Високе значення R^2 свідчить про здатність моделі точно відтворювати залежності між вхідними ознаками та цільовою змінною. Низькі значення MAE та RMSE означають, що середня похибка прогнозу становить близько 2–3 оренди на день, що є прийнятним рівнем для практичного використання.

Отримана модель XGBoost була інтегрована у вебінтерфейс системи прокату для щоденного прогнозування попиту на велосипеди в місті Вінниця. Її точність і продуктивність дозволяють використовувати результати прогнозу для прийняття рішень щодо розподілу велосипедів, планування обслуговування та оптимізації роботи прокатної мережі.

Крім основного навчання на згенерованому наборі даних, модель передбачає можливість дотренування на реальних даних, які надходять безпосередньо з вебсистеми. Щодня система отримує фактичну погоду у місті Вінниця за допомогою API OpenWeather, а також фіксує реальну кількість оренд, здійснених у цей день. Ці дані зберігаються у внутрішній базі, де можуть бути використані для оновлення набору даних та періодичного донавчання моделі.

Приклад виведення прогнозу зображено на рисунку 3.23.



Рисунок 3.23 – Прогноз попиту

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ

4.1 Вибір інструментів розробки

Під час вибору інструментів для створення інтелектуальної системи прокату велосипедів необхідно враховувати технічні вимоги проекту, зручність у реалізації окремих функцій та доступність необхідної документації. Для цього проекту обирались ті засоби, які дозволяють швидко реалізовувати веб-функціонал, підключати моделі штучного інтелекту та ефективно працювати з базою даних. Існує велика кількість рішень, однак обирались ті, які найкраще відповідають задачам автоматизації процесу оренди велосипедів.

4.1.1 Вибір середовища розробки програмного забезпечення

У процесі реалізації інтелектуальної системи прокату велосипедів було проаналізовано різні варіанти середовищ розробки для вибору найзручнішого та найефективнішого. Оскільки система розробляється з використанням мови програмування Python, фреймворку Django та бази даних PostgreSQL, середовище мало забезпечувати якісну підтримку цих технологій. Було розглянуто декілька популярних інструментів: PyCharm, Sublime Text, Atom та Visual Studio Code. Зрештою, вибір зупинився саме на Visual Studio Code.

Visual Studio Code (VS Code) - це легке, кросплатформне середовище з відкритим кодом, яке має багатий функціонал, достатній як для початківців, так і для досвідчених розробників. Воно забезпечує гнучке налаштування та підтримує велику кількість розширень, що значно спрощують роботу з Python та Django. Особливо корисними в процесі розробки були розширення Python, Django, PostgreSQL, GitLens та Live Server. Окремо варто відзначити підтримку

інтегрованого терміналу, зручну навігацію по проєкту та можливість швидкого перемикання між файлами.

Порівняно з PyCharm, який має вбудовані інструменти для Django, VS Code працює швидше, не навантажує систему та не потребує ліцензії у повнофункціональній версії. Також важливу роль відіграє велика кількість онлайн-ресурсів, навчальних матеріалів та активна спільнота, що дозволяє швидко знаходити відповіді на технічні питання. Усе це робить Visual Studio Code оптимальним середовищем розробки для реалізації даного проєкту.

4.1.2 Вибір інструментів зберігання інформації

Для забезпечення надійного та ефективного зберігання інформації у рамках інтелектуальної системи прокату велосипедів було обрано систему управління базами даних PostgreSQL[34]. Це сучасна реляційна СУБД з відкритим вихідним кодом, яка відзначається високою продуктивністю, стабільністю та широкими функціональними можливостями. Її застосування дозволяє реалізовувати складні структури даних, підтримувати логічні зв'язки між об'єктами, виконувати складні агрегатні запити та гарантувати цілісність інформації.

Завдяки підтримці транзакцій, зовнішніх ключів, унікальних індексів та перевірок на рівні бази, PostgreSQL дає змогу контролювати коректність усіх операцій з даними. У контексті даної системи це особливо важливо, оскільки потрібно зберігати пов'язані дані про користувачів, замовлення, екземпляри велосипедів, записи про ремонти, логування поїздок, а також дані, що використовуються для прогнозування технічного стану транспорту.

Обрана СУБД чудово інтегрується з фреймворком Django за допомогою офіційного драйвера psycopg2, що забезпечує безперебійну роботу з ORM (об'єктно-реляційним відображенням). Це спрощує реалізацію CRUD-операцій, забезпечує високу швидкодію при виконанні запитів, а також дозволяє легко

масштабувати систему. При необхідності база даних може бути перенесена до хмарної інфраструктури або скопійована для підвищення надійності.

Вся структура таблиць системи побудована за принципами ER-моделювання, що дає змогу забезпечити логічну цілісність, узгодженість і гнучкість при подальшому розвитку функціоналу проекту. На рисунку 4.1 відображена ER-модель бази даних:



Рисунок 4.1 - ER-модель бази даних

На схемі рисунку 4.1 можна побачити, що база даних містить 10 таблиць, ось їх короткий опис:

- **CATEGORY** - таблиця для зберігання категорій велосипедів. Поля включають унікальний ідентифікатор (id), назву категорії (name) та короткий ідентифікатор (slug);
- **PRODUCT** - таблиця для зберігання інформації про моделі велосипедів. Поля включають унікальний ідентифікатор (id), назву велосипеда (name), посилання на категорію (category), ціну за годину (price_hourly), за день (price_daily) та за тиждень (price_weekly), наявність (available), опис (description), дату створення (created), дату оновлення (updated), знижку (discount), кількість замовлень (order_count), URL-ідентифікатор (slug) та зображення (image);
- **PRODUCTINSTANCE** - таблиця для зберігання окремих екземплярів велосипедів. Поля включають унікальний ідентифікатор (id), посилання на модель велосипеда (product), розмір (size), пробіг (mileage) та пункт видачі (pickup_point);
- **REPAIR** - таблиця для зберігання інформації про ремонти велосипедів. Поля включають унікальний ідентифікатор (id), посилання на екземпляр велосипеда (product_instance), дату ремонту (repair_date), тип ремонту (repair_type), пробіг на момент ремонту (mileage), опис (description), вік велосипеда у місяцях (bike_age_months), днів після останнього ремонту (days_since_last_repair), факт технічного обслуговування (technical_service_done), кількість відмов за рік (failures_last_year), середній пробіг на місяць (mileage_per_month) та коефіцієнт поломок (failure_ratio);
- **MILEAGEHISTORY** - таблиця для зберігання історії пробігу велосипедів. Поля включають унікальний ідентифікатор (id), посилання на екземпляр велосипеда (product_instance), попередній пробіг (previous_mileage), оновлений пробіг (updated_mileage) та дату оновлення (updated_at);

- **ORDER** - таблиця для зберігання замовлень. Поля включають унікальний ідентифікатор (id), користувача (user), статус замовлення (status), ім'я (first_name), прізвище (last_name), електронну пошту (email), номер телефону (phone), дату створення (created), дату оновлення (updated), факт оплати (paid), факт скасування (is_canceled) та номер замовлення користувача (user_order_number);
- **ORDERITEM** - таблиця для зберігання окремих позицій у замовленнях. Поля включають унікальний ідентифікатор (id), посилання на замовлення (order), велосипед (product), категорію (category), ціну (price), кількість (quantity), назву велосипеда (bicycle_name), дати прокату (date_from, date_to), час видачі та повернення (pickup_time, return_time), екземпляр велосипеда (product_instance) та розмір (size);
- **USER** - таблиця для зберігання даних користувачів. Поля включають унікальний ідентифікатор (id), ім'я користувача (username), ім'я (first_name), прізвище (last_name), електронну пошту (email), пароль (password), аватар (image);
- **RIDELOG** - таблиця для зберігання логів поїздки. Поля включають унікальний ідентифікатор (id), користувача (user), позицію замовлення (order_item), екземпляр велосипеда (product_instance), дату та час прокату (date_from, pickup_time) і повернення (date_to, return_time), пройдену відстань (distance_km), погодні умови (weather_condition), температуру (temperature_c), вологість (humidity_percent), швидкість вітру (wind_speed_kph) та дату запису (created);
- **PICKUPPOINT** - таблиця для зберігання точок видачі. Поля включають унікальний ідентифікатор (id), назву пункту (name) та адресу (address).
- **DEMANDFORECAST** – таблиця для зберігання прогнозованого попиту на оренду велосипедів. Поля включають унікальний ідентифікатор (id), дату прогнозу (date), значення прогнозованого попиту

(predicted_demand), фактичну кількість оренд (actual_demand), параметри погоди: температуру (temperature), вологість (humidity), швидкість вітру (wind_speed), код погодних умов (weather_code), текстовий опис погоди (weather_main), дату створення запису (created_at).

- FAILUREFORECAST – таблиця для зберігання прогнозу ймовірності відмови окремих велосипедів. Поля включають унікальний ідентифікатор (id), екземпляр велосипеда (product_instance), дату прогнозу (prediction_date), ймовірність відмови (predicted_failure_probability), прогнозовану відстань до відмови в кілометрах (prediction_distance_km), дату створення запису (created_at) та дату останнього оновлення (updated_at).

4.1.3 Вибір інструментів розробки користувацького інтерфейсу

Для створення зручного та зрозумілого користувацького інтерфейсу було вибрано класичний набір технологій[35] – HTML, CSS та JavaScript. Дані технології, на даний момент, є основою для створення сучасного інтерфейсу. За допомогою цих технологій можна отримати коректне відображення контенту, а також створення інтерактивних елементів сторінки.

HTML (HyperText Markup Language) – це спеціальна мова розмітки, яка застосовується при створенні сайтів в інтернеті. Браузери добре розуміють html і можуть інтерпретувати у зрозумілому для людини вигляді.

CSS (аббревіатура від Cascading Style Sheets, що в перекладі означає каскадні таблиці стилів) - це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи веб-сторінки), написаних мовами розмітки даних. Завдання CSS - прикрасити сайт, надати проєкту завершеного вигляду.

JavaScript – це високорівнева скриптова мова програмування, яка широко використовується для створення інтерактивних застосунків і сайтів. JS додає анімацію і спливаючі повідомлення, валідує форми, оновлює контент і змушує інтерфейс реагувати на наші дії.

4.2 Розробка клієнської частини системи прокату велосипедів

Головна сторінка є першою, що бачить користувач після переходу на сайт. Вона виконує ознайомчу функцію і допомагає зрозуміти, як працює сервіс оренди велосипедів. На рисунку 4.2 відображено головну сторінку сайту.

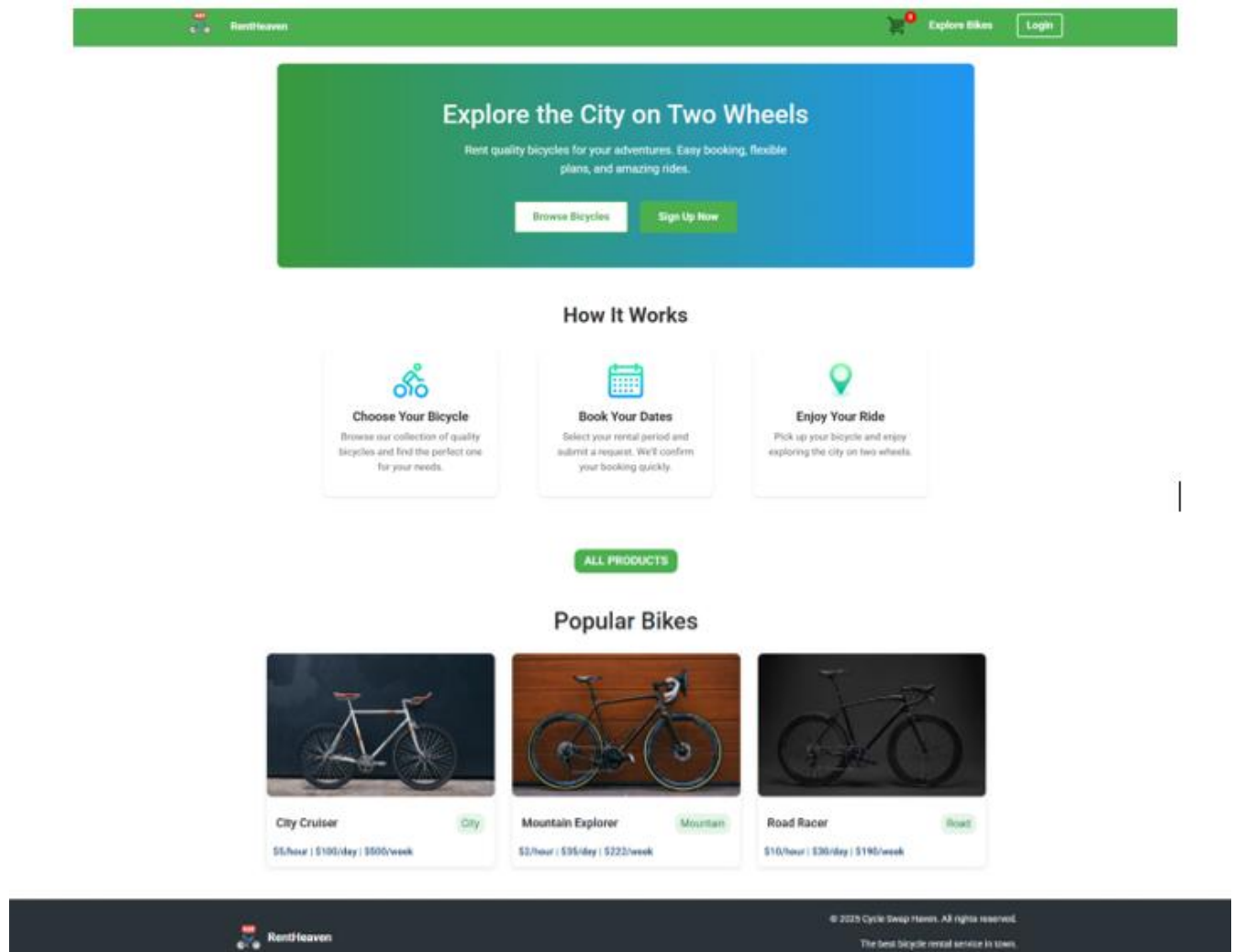


Рисунок 4.2 – Головна сторінка сайту

У верхній частині сторінки розміщено меню, яке прикріплюється до верхньої частини сайту і є на всіх сторінках, з логотипом RentHeaven, кнопкою перегляду велосипедів, іконкою кошика та кнопкою входу. Кнопка Explore Bikes веде до каталогу, Login - відкриває сторінку авторизації, а кошик відображає кількість обраних товарів.

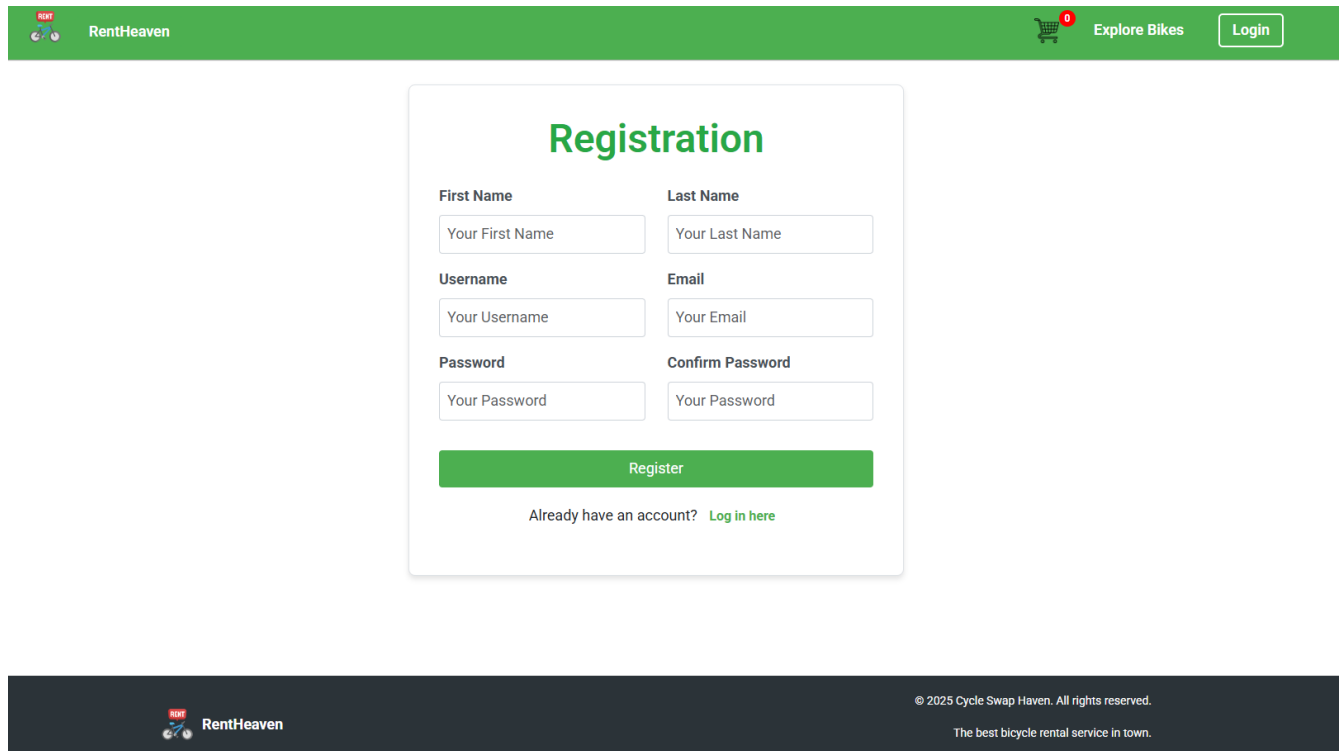
У центрі знаходиться банер із заголовком Explore the City on Two Wheels та двома кнопками: Browse Bicycles для перегляду моделей і Sign Up Now для реєстрації. Нижче розміщено розділ How It Works, який у трьох кроках пояснює, як працює система: вибір велосипеда, бронювання дат, поїздка.

У нижній частині сторінки розміщено блок Popular Bikes із картками найпопулярніших моделей. Вони містять фото, назву, тип велосипеда, ціни та кнопку All Products, яка відкриває повний каталог.

Сторінка завершується футером із логотипом, роком і підписом The best bicycle rental service in town. Інтерфейс розрахований на зручну й швидку навігацію для користувачів, які хочуть одразу перейти до вибору велосипеда.

На рисунку 4.3 відображена форма реєстрації. Ця сторінка дозволяє новому користувачу створити обліковий запис. Форма містить шість полів для заповнення: First Name - ім'я, Last Name - прізвище, Username - логін, Email - електронна пошта, Password - пароль, Confirm Password - повторення пароля

Після заповнення всіх полів потрібно натиснути кнопку Register, щоб завершити реєстрацію. Внизу сторінки є посилання Log in here, яке дозволяє перейти на сторінку входу, якщо обліковий запис уже існує.



The image shows a registration form for the RentHeaven website. The form is centered on a white background with a green border. At the top, the title "Registration" is displayed in green. Below the title, there are six input fields arranged in three rows and two columns. The first row contains "First Name" and "Last Name" fields. The second row contains "Username" and "Email" fields. The third row contains "Password" and "Confirm Password" fields. Each field has a placeholder text: "Your First Name", "Your Last Name", "Your Username", "Your Email", "Your Password", and "Your Password". Below the input fields is a green "Register" button. Underneath the button, there is a link: "Already have an account? [Log in here](#)". The form is set against a background that includes a green header with the RentHeaven logo and navigation links, and a dark footer with copyright information.

RentHeaven

Explore Bikes Login

Registration

First Name Last Name

Your First Name Your Last Name

Username Email

Your Username Your Email

Password Confirm Password

Your Password Your Password

Register

Already have an account? [Log in here](#)

RentHeaven

© 2025 Cycle Swap Haven. All rights reserved.
The best bicycle rental service in town.

Рисунок 4.3 – Форма реєстрації

На рисунку 4.4 показана форма логіну у акаунт. Сторінка логіну дозволяє користувачам увійти до свого облікового запису. У формі є два поля: Username - введення логіна, Password - введення пароля. Після введення даних потрібно натиснути кнопку Login, щоб отримати доступ до особистого кабінету та можливості оформлення замовлень. Нижче розташовані два посилання: Reset Password - для відновлення пароля, якщо користувач його забув, Create Account - перехід на сторінку реєстрації

Інтерфейс сторінки мінімалістичний, що дозволяє швидко авторизуватись без зайвих кроків.

Рисунок 4.4 – Форма логіну

Каталог є однією з найважливіших частин сайту, оскільки саме тут користувач обирає велосипед для оренди. Через каталог забезпечується доступ до повного списку товарів, їхніх характеристик, вартості оренди та доступних розмірів. Завдяки фільтрам та зручній навігації користувач може швидко знайти потрібну модель і додати її до кошика. На рисунку 4.5 відображений каталог товарів. Інтерфейс каталогу розроблений таким чином, щоб забезпечити інтуїтивну взаємодію з системою та мінімізувати кількість дій, необхідних для вибору велосипеда. Важливим аспектом є також можливість швидкого пошуку за ключовими параметрами, що особливо актуально при великій кількості доступних моделей.

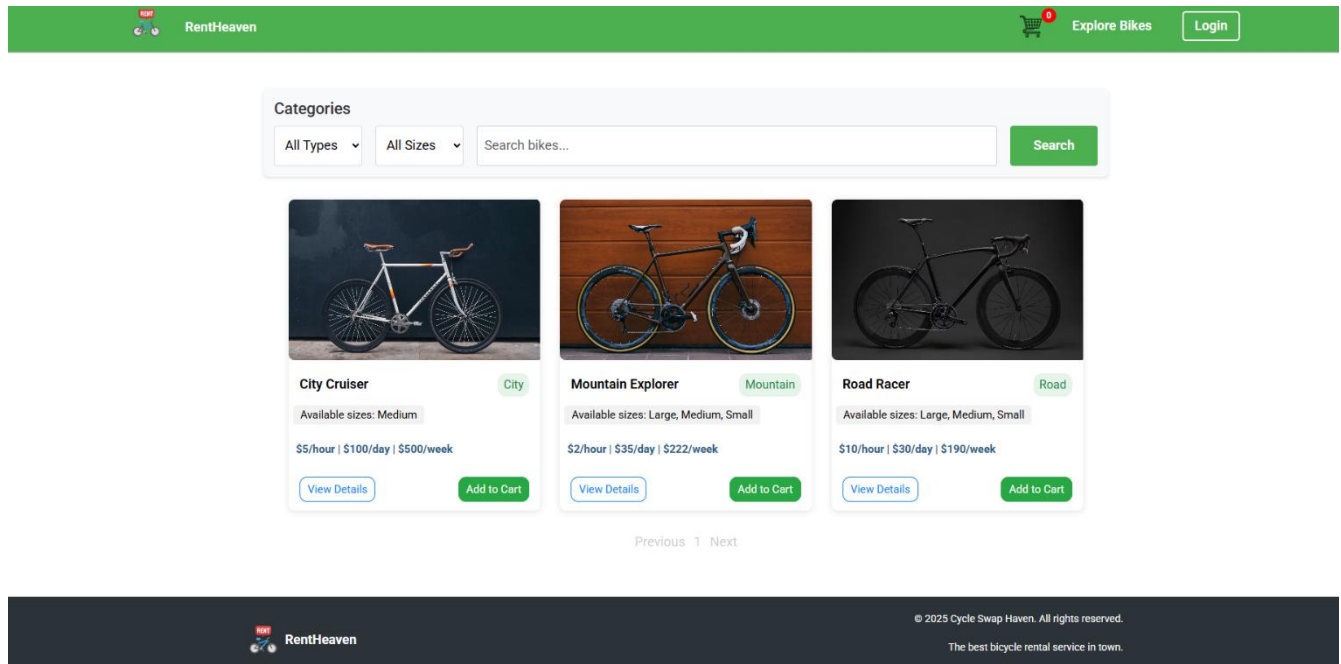


Рисунок 4.5 – Каталог велосипедів

Після завантаження сторінки каталогу відображається панель фільтрації, що складається з трьох елементів: випадаючого списку All Types для вибору типу велосипеда (наприклад, City, Road, Mountain), фільтра All Sizes для вибору розміру рами, а також поля пошуку з кнопкою Search для швидкого пошуку моделі за назвою.

Нижче розміщено картки велосипедів. Кожна картка містить:

- зображення моделі;
- назву (наприклад, City Cruiser);
- тип (позначено у вигляді бейджа справа вгорі, наприклад City);
- доступні розміри (наприклад, Available sizes: Medium);
- вартість оренди в трьох форматах: за годину, день та тиждень;
- кнопку View Details, яка відкриває сторінку з детальним описом велосипеда;
- кнопку Add to Cart, яка додає вибраний велосипед до кошика.

Також у нижній частині каталогу розміщено елементи для перемикання сторінок: Previous, Next та поточний номер сторінки.

Сторінка детального перегляду велосипеда відображена на рисунку 4.6, вона складається з двох основних частин. Ліворуч розміщено велике зображення велосипеда та текстовий опис моделі англійською мовою. В описі вказано призначення велосипеда, його особливості та технічні переваги.

Праворуч знаходиться блок бронювання. У верхній частині вказується вартість оренди за день та тиждень. Нижче розташоване поле вибору діапазону дат (Choose your date range) і два окремі поля для вибору часу видачі (Pickup time) та повернення (Return time).

Далі користувач обирає розмір велосипеда з випадаючого списку (Bike size) та кількість одиниць для замовлення. Під формою розміщено дві кнопки:

- Add to cart - додає вибраний велосипед до кошика;
- Перевірити наявність - дозволяє дізнатись, чи є велосипед у наявності на обраний період.

Нижче виводиться автоматичний розрахунок вартості оренди залежно від обраних дати й кількості. Наприкінці сторінки вказано, на якій точці видачі велосипед доступний, із зазначенням адреси та кількості доступних екземплярів.

Ця сторінка дозволяє користувачеві переглянути детальну інформацію про велосипед, налаштувати параметри оренди та одразу перейти до оформлення.

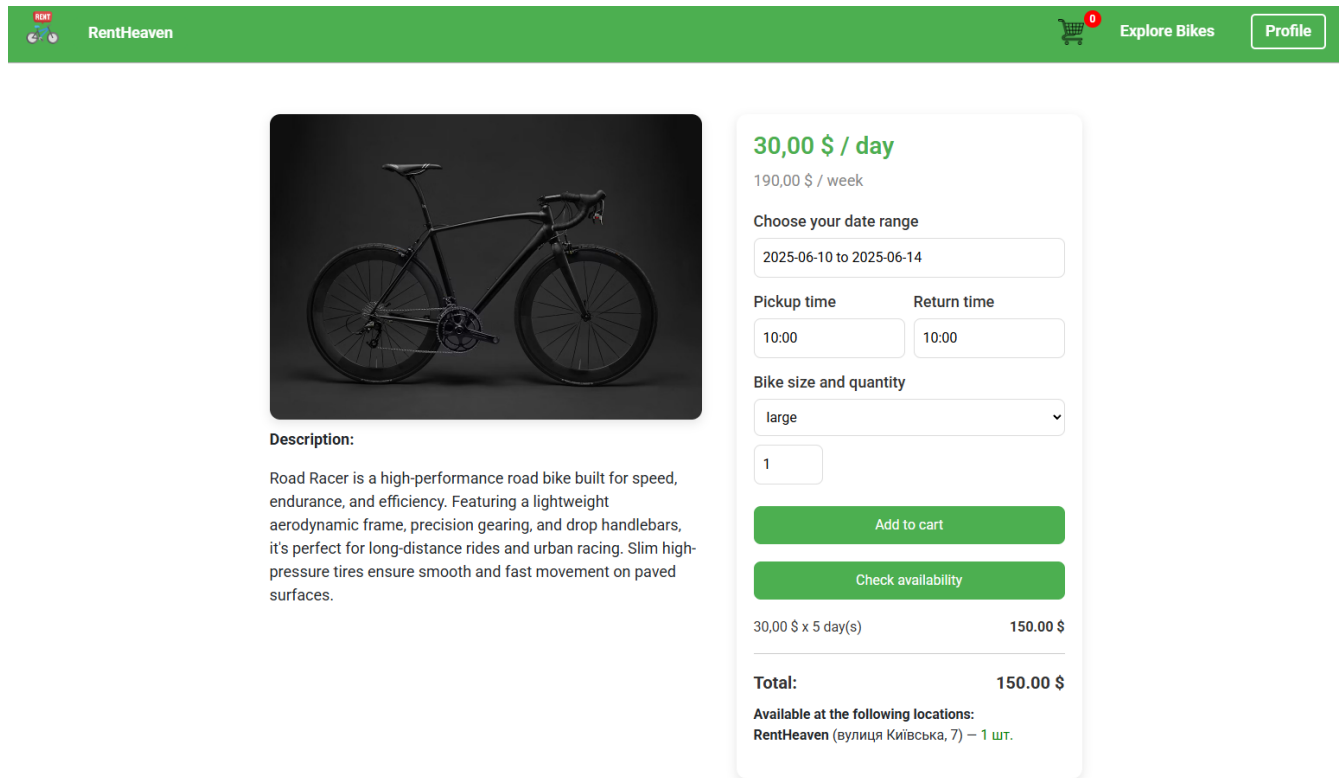


Рисунок 4.6 - Детальний перегляд велосипеда

Сторінка кошика, яка показана на рисунку 4.7, відображає усі поточні бронювання, які користувач додав до кошика. Зліва показано зображення велосипеда, його назву, короткий опис, вибраний розмір, період оренди, час видачі та повернення, кількість одиниць і ціна. Є можливість видалити бронювання через іконку з кошиком.

Праворуч знаходиться блок Booking overview, у якому повторно вказано модель, розмір, дати прокату та загальну суму до сплати. Завершується блок кнопкою Rent now, що переходить до підтвердження замовлення.

Поки велосипед знаходиться в кошику, він автоматично резервується на вибраний період. Якщо в системі доступний лише один екземпляр і хтось його вже додав у кошик, інші користувачі не можуть оформити замовлення на ці ж дати. Велосипед стане знову доступним лише після скасування або завершення відповідного замовлення. Це дозволяє уникнути конфліктів під час бронювання

одного й того ж екземпляра. Також якщо користувач не зареєстрований, то йому буде зазначено, що він зможе оформити замовлення тільки після реєстрації або логіну.

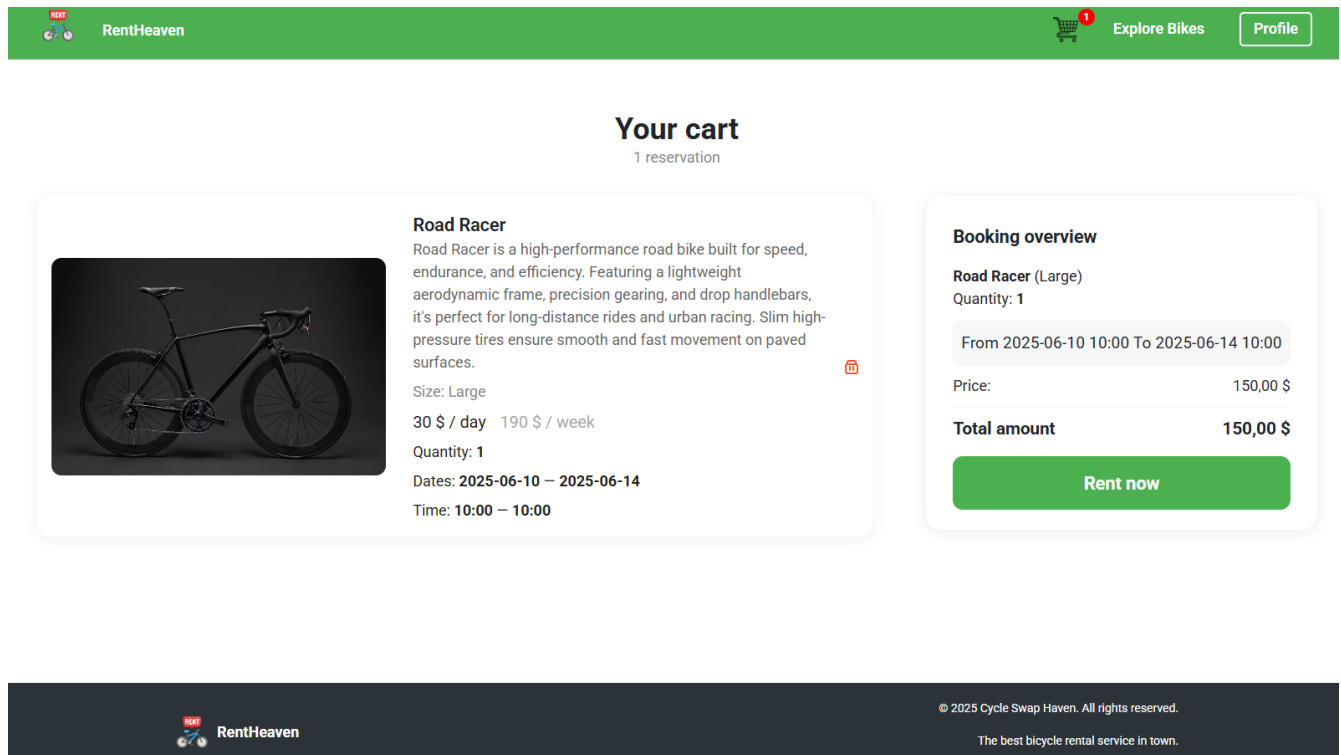


Рисунок 4.7 - Кошик

Сторінка бронювання замовлення відображена на рисунку 4.8, ця сторінка відкривається після натискання кнопки Rent now. Вона поділена на два блоки.

Ліворуч розташована форма Create Order, яка містить чотири обов'язкові поля:

- First Name - ім'я користувача
- Last Name - прізвище
- Email - електронна пошта
- Phone - номер телефону

Нижче розміщено кнопку Place Order, яка підтверджує замовлення і надсилає його на обробку. Праворуч виводиться блок Checkout, де показано короткий

підсумок замовлення: назва велосипеда, обраний розмір, кількість, дати прокату та загальна вартість.

Ця сторінка є завершальним етапом перед створенням замовлення й дозволяє користувачу перевірити правильність введених даних перед підтвердженням.

Рисунок 4.8 – Сторінка бронювання замовлення

Сторінка профілю показана на рисунку 4.9 призначена для перегляду та редагування особистих даних користувача. Ліворуч відображається аватар, який можна змінити через кнопку Вибрати файл. Після вибору нового зображення його буде оновлено після натискання кнопки Save. Також тут можна переглядати історію замовлення та скасувати бронювання. Праворуч знаходиться форма з основними полями: First Name - ім'я користувача, Last Name - прізвище, Username - логін, Email - електронна пошта. У нижній частині розміщено дві кнопки: Save - зберігає зміни в профілі, Logout - вихід з облікового запису

Ця сторінка дозволяє користувачу керувати своїми даними, оновлювати контактну інформацію або змінювати фото профілю.

Profile

 [Вибрати файл](#) файл не вибрано

First Name
OLEKSANDR

Last Name
KABASHNYI

Username
admin

Email
sasha@gmail.com

[Save](#) [Logout](#)

Your Orders

Order № 10
 Name: Road Racer | Size: Large
 Quantity: 1, Price: \$150,00
 Rental Period: 05 червня 2025 р. 10:00 – 09 червня 2025 р. 10:00
 Date of order: 05 червня 2025 р. 03:15
[Cancel Order](#)

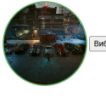
Order № 9
 Name: City Cruiser | Size: Medium
 Quantity: 1, Price: \$400,00
 Rental Period: 04 червня 2025 р. 10:00 – 07 червня 2025 р. 10:00
 Date of order: 04 червня 2025 р. 01:14
[Trip Completed](#)

Name: City Cruiser | Size: Medium
 Quantity: 1, Price: \$400,00
 Rental Period: 04 червня 2025 р. 10:00 – 07 червня 2025 р. 10:00
 Date of order: 04 червня 2025 р. 01:14

Рисунок 4.9 – Профіль користувача

Також для адміністрації є доступ до спеціального блоку «Admin Tools», де можна переглянути всі велосипеди, прогноз попиту та логування поїздок.

Profile

 [Вибрати файл](#) файл не вибрано

First Name
OLEKSANDR

Last Name
KABASHNYI

Username
admin

Email
sasha@gmail.com

[Save](#) [Logout](#)

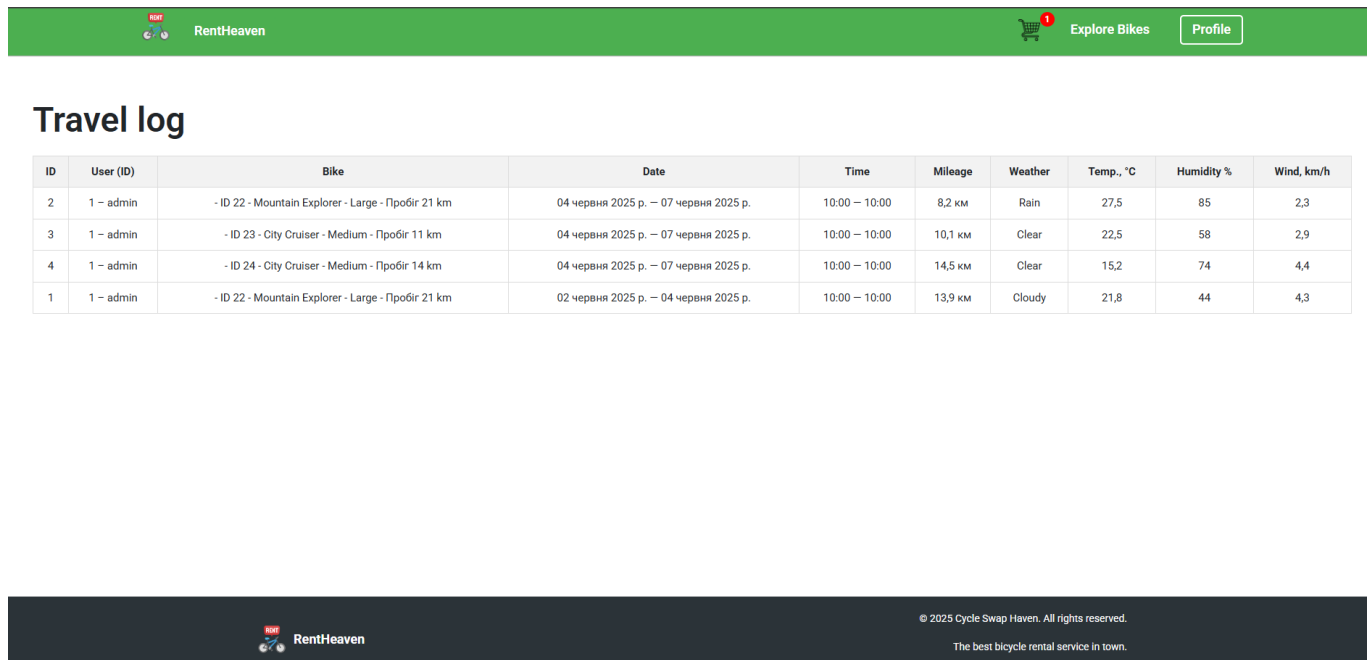
Admin Tools
[Manage Bicycles](#)
[AI Forecast](#)
[Travel log](#)

Рисунок 4.10 - Інструменти адміністратора

На рисунку 4.11 відображена сторінка, яка призначена для перегляду історії використання велосипедів. Вона подає дані у вигляді таблиці, де кожен рядок відповідає окремій завершеній поїздки користувача.

У таблиці виводяться такі колонки: ID — унікальний номер поїздки, User (ID) — ім'я користувача і його ідентифікатор, Bike — назва велосипеда, розмір та пробіг, Date — період прокату (від дати до дати), Time — час початку і завершення поїздки, Mileage — фактична відстань, яку подолав велосипед (в кілометрах), Weather — погодні умови під час поїздки, Temp., °C — температура повітря, Humidity % — вологість, Wind, km/h — швидкість вітру.

Ця сторінка надає адміністраторам або користувачам доступ до аналітики щодо використання велосипедів, а також погодних умов, що можуть впливати на технічний стан транспорту. Таблиця оновлюється автоматично після завершення кожної оренди.



ID	User (ID)	Bike	Date	Time	Mileage	Weather	Temp., °C	Humidity %	Wind, km/h
2	1 – admin	- ID 22 - Mountain Explorer - Large - Пробіг 21 km	04 червня 2025 р. – 07 червня 2025 р.	10:00 – 10:00	8,2 км	Rain	27,5	85	2,3
3	1 – admin	- ID 23 - City Cruiser - Medium - Пробіг 11 km	04 червня 2025 р. – 07 червня 2025 р.	10:00 – 10:00	10,1 км	Clear	22,5	58	2,9
4	1 – admin	- ID 24 - City Cruiser - Medium - Пробіг 14 km	04 червня 2025 р. – 07 червня 2025 р.	10:00 – 10:00	14,5 км	Clear	15,2	74	4,4
1	1 – admin	- ID 22 - Mountain Explorer - Large - Пробіг 21 km	02 червня 2025 р. – 04 червня 2025 р.	10:00 – 10:00	13,9 км	Cloudy	21,8	44	4,3

Рисунок 4.11 - Логування поїздок

Для просто перегляду замовлень, додавання нових категорій, велосипедів, ремонтів і інших можливостей, я використовую Django адміністрування. Django

адмінка (або Django admin) — це вбудована адміністративна панель, яку автоматично надає фреймворк Django. Це веб-інтерфейс "із коробки", через який можна керувати вмістом бази даних — без потреби писати додатковий код або SQL-запити. Вона доступна лише для зареєстрованих суперкористувачів (superuser). На рисунку 4.12 відображена Django адміністрування, де можна додавати все що треба для повноцінного сайту.

Django адміністрування

Адміністрування сайта

DIAGNOSTICS

Demand forecasts
+ Додати
Змінити

Repairs
+ Додати
Змінити

LOGS

Ride logs
+ Додати
Змінити

MAIN

Categories
+ Додати
Змінити

Pickup points
+ Додати
Змінити

Product instances
+ Додати
Змінити

Products
+ Додати
Змінити

ORDERS

Orders
+ Додати
Змінити

USERS

Users
+ Додати
Змінити

АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ

Групи
+ Додати
Змінити

Недавні дії

Мої дії

- ID 24 - City Cruiser - Medium - Пробір 600 km
Product instance

- ID 24 - City Cruiser - Medium - Пробір 14 km
Product instance

- ID 23 - City Cruiser - Medium - Пробір 11 km
Product instance

- ID 22 - Mountain Explorer - Large - Пробір 21 km
Product instance

- ID 20 - Mountain Explorer - Small - Пробір 5000 km
Product instance

- ID 19 - Mountain Explorer - Medium - Пробір 100 km
Product instance

Road Racer
Product

+ - ID 27 - Road Racer - Medium - Пробір 0 km
Product instance

+ - ID 26 - Road Racer - Large - Пробір 0 km
Product instance

+ - ID 25 - Road Racer - Small - Пробір 0 km
Product instance

Рисунок 4.12 – Загальний вигляд Django адміністрування

Тут можна повністю керувати сайтом, для прикладу, на рисунку 4.13, показано як відображаються замовлення. Тут адміністратор може змінювати статуси замовлення, скасовувати, видаляти.

Домівка · Orders · Orders

Почніть писати для фільтру...

DIAGNOSTICS

Demand forecasts + Додати

Repairs + Додати

LOGS

Ride logs + Додати

MAIN

Categories + Додати

Pickup points + Додати

Product instances + Додати

Products + Додати

ORDERS

Orders + Додати

USERS

Users + Додати

АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ

Групи + Додати

Виберіть order щоб змінити

Дія: ----- Вперед Обрано 0 з 10

☐	ID	FIRST NAME	LAST NAME	EMAIL	PHONE	CREATED	ВЕЛОСИПЕДИ	ПЕРІОД ОРЕНДИ	ВАРТІСТЬ	СТАТУС
☐	10	OLEKSANDR	KABASHNYI	sasha@gmail.com	0735634659	05 червня 2025 р. 03:15	1x Road Racer	2025-06-05 — 2025-06-09	150,00	✗ Скасовано
☐	9	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	04 червня 2025 р. 01:14	1x City Cruiser, 1x City Cruiser	2025-06-04 — 2025-06-07; 2025-06-04 — 2025-06-07	800,00	✓ Повернено
☐	8	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	04 червня 2025 р. 01:13	1x Mountain Explorer	2025-06-04 — 2025-06-07	140,00	✓ Повернено
☐	7	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 19:12	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	6	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 19:08	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	5	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 19:05	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	4	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 18:59	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	3	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	28 травня 2025 р. 02:36	1x City Cruiser, 1x City Cruiser	2025-05-28 — 2025-05-31; 2025-05-28 — 2025-05-31	400,00	✓ Повернено
☐	2	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	28 травня 2025 р. 01:47	1x Mountain Explorer	2025-05-27 — 2025-05-30	140,00	✓ Повернено
☐	1	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	20 травня 2025 р. 01:38	1x Mountain Explorer	2025-05-19 — 2025-05-23	175,00	✗ Скасовано

10 orders

Рисунок 4.13 – Перегляд замовлень

У цьому розділі було реалізовано основні функції програмної частини системи прокату велосипедів. Створено сайт, на якому користувач може обрати велосипед, оформити замовлення та переглянути інформацію про поїздки, а адміністратор може клегко керувати системою. Було проведено тестування роботи функціоналу — система стабільно працює та видає логічні результати.

ВИСНОВКИ

У ході виконання завдання вдалося реалізувати зручну та зрозумілу систему прокату велосипедів, яка не лише забезпечує можливість онлайн-оренди, а й автоматизує низку процесів, що раніше виконувались вручну. Користувач отримує інтуїтивний інтерфейс для бронювання, перегляду історії поїздки і керування замовленнями. Адміністратор, у свою чергу, має доступ до панелі керування велосипедами, замовленнями, логуванням поїздки та технічним станом транспорту.

Система створена на базі сучасних вебтехнологій: Django, Python, PostgreSQL, HTML, CSS і JavaScript що забезпечує стабільну роботу й можливість масштабування.

Особливу роль у роботі системи відіграють модулі штучного інтелекту. Один із них відповідає за прогнозування попиту на оренду з урахуванням погодних умов, дня тижня, сезонності та інших факторів. Інший модуль оцінює ймовірність технічної несправності велосипеда, використовуючи дані про пробіг, вік, історію поломок, погодні умови та обслуговування. Це дозволяє планувати розподіл велосипедів між точками видачі та знижувати ризик несправностей під час використання. Для реалізації цих функцій використано модель XGBoost як для задачі прогнозування попиту, так і для оцінки технічного стану. Навчання моделей проводилось на власноруч згенерованих наборах даних із подальшою можливістю донавчання на реальних прикладах у процесі використання системи. Обидві моделі показали високі результати точності.

Загалом проєкт зарекомендував себе як гнучка та ефективна основа, яку можна доповнювати новими модулями, масштабувати під більші обсяги даних та адаптувати під специфічні умови використання. Система є зручною як для користувачів, так і для адміністраторів, і має потенціал для подальшого розвитку як сучасний сервіс інтелектуального прокату.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система прокату велосипедів. Офіційна Web-сторінка Wikipedia. [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Bicycle-sharing_system
2. Golec P. Bike Rental System. Офіційна Web-сторінка ResearchGate [Електронний ресурс]. URL: https://www.researchgate.net/publication/387031352_BIKE_RENTAL_SYSTEM
3. Mrówczyńska M., Marciniak M. The Bike-Sharing System as an Element of Enhancing Sustainable Urban Mobility. Офіційна Web-сторінка MDPI. [Електронний ресурс]. URL: <https://www.mdpi.com/2071-1050/12/8/3285>
4. Офіційна Web-сторінка компанії Veliki.ua. [Електронний ресурс]. URL: <https://veliki.ua/>
5. Офіційна Web-сторінка компанії BikeNow. [Електронний ресурс]. URL: <https://bikenow.com.ua/>
6. Офіційна Web-сторінка компанії Velokratia. [Електронний ресурс]. URL: <https://velokratia.com.ua/bikes/>
7. Категорії(класифікація) велосипедів. Офіційна Web-сторінка VELOGO. [Електронний ресурс]. URL: <https://velogo.com.ua/ua/blog/klassifikatsiia-velosipedov-po-stiliu-kataniia>
8. Технічні характеристики велосипеда. Офіційна Web-сторінка VELOBUCHA. [Електронний ресурс]. URL: https://velobucha.com.ua/index.php?route=news/article&news_id=62&srsId=AfmBOorh7Y3WNc8-Dpa3J5EkIse66Zh64PDHnOsFDNOiP7qOhcEF0rSk
9. Денис Демчина. Як відкрити бізнес на прокаті велосипедів з нуля. Офіційна Web-сторінка Go Deal Brokers. [Електронний ресурс]. URL: <https://business-broker.com.ua/blog/iak-vidkryty-biznes-na-prokati-velosypediv-z-nulia-pokrokovaya-instruktsiia-dlia-pidpryiemtsiv/>

- 10.Cheng J., Correia G., Cats O., Sharif Azadeh S. Short-term bike-sharing demand forecasting incorporating multiple sources of uncertainties. Офіційний документ EPFL. [Електронний ресурс]. URL: https://trc-30.epfl.ch/wp-content/uploads/2024/09/TRC-30_paper_89.pdf
- 11.Grau-Escolano J., Bassolas A., Vicens J. Cycling into the workshop: predictive maintenance for Barcelona's bike-sharing system. Офіційний документ arXiv. [Електронний ресурс]. URL: <https://arxiv.org/pdf/2404.17217>
- 12.Zhong W., Chen F., Fu K., Lu C.-T. SAFEBIKE: A Bike-sharing Route Recommender with Availability Prediction and Safe Routing. Офіційний документ arXiv. [Електронний ресурс]. URL: <https://arxiv.org/pdf/1712.01469>
- 13.Gammelli D., Wang Y., Prak D., Rodrigues F., Minner S., Camara Pereira F. Predictive and Prescriptive Performance of Bike-Sharing Demand Forecasts for Inventory Management. Офіційний документ. [Електронний ресурс]. URL: <https://arxiv.org/pdf/2108.00858>
- 14.UML-діаграми. Офіційна Web-сторінка Evergreens.. [Електронний ресурс]. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
- 15.Django MVT структура. Офіційна Web-сторінка GeeksforGeeks. [Електронний ресурс]. URL: <https://www.geeksforgeeks.org/django-project-mvt-structure/>
- 16.Яку мову програмування вибрати? Офіційна Web-сторінка Lemon School. [Електронний ресурс]. URL: <https://lemon.school/blog/yaku-movu-programuvannya-vibrati>
- 17.Рейтинг баз даних. Офіційна Web-сторінка DB-Engines. [Електронний ресурс]. URL: <https://db-engines.com/en/ranking>
- 18.SQL чи NoSQL? Офіційна Web-сторінка AlternativeScience. [Електронний ресурс]. URL: <https://alternativescience.net/programming/242-sql-chy-nosql-os-v-chomu-pytannya/>

- 19.Що таке фреймворк? Офіційна Web-сторінка Brainlab. [Електронний ресурс].
URL: <https://brainlab.com.ua/uk/blog-uk/shho-take-frejmvork-poyasnyuyemo-prostymy-slovamy>
20. Грищук Т. В., Юркіна О. М. Комп'ютерна програма «Маршрути громадського транспорту» : авторське свідоцтво № 133380. – Зареєстр. в Держ. реєстрі комп'ютерних програм 11.02.2025 бюлетень № 87 від 31.03.2025.
- 21.Django Web Framework. Офіційна Web-сторінка MDN.
[Електронний ресурс]. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Django
- 22.Основи Flask. Офіційна Web-сторінка PNN. [Електронний ресурс].
URL:<https://pnn.com.ua/ua/blog/detail/lightweight-and-flexible-flask-fundamentals-for-software-web-development>.
- 23.Обслуговування велосипеда. Офіційна Web-сторінка БЕЛОСКЛАД.
[Електронний ресурс]. URL: <https://velosklad.ua/obslugovuvannya-velosipeda/?srsltid=AfmBOooipaUzLTsBN37Gk0gI0ksSkkZwgX4ThSYvcV-Q7n4HbbFo8eHE>
24. Л.М. Тичинська, А.А. Черепащук. Теорія ймовірностей. ч. 1. Історичні екскурси та основні теоретичні відомості : навчальний посібник – Вінниця : ВНТУ, 2010. – 112 с.
25. В.Б.Мокін, М.В.Дратований. Наука про дані: машинне навчання та інтелектуальний аналіз даних - Електронний навчальний посібник /Вінниця : ВНТУ, 2024. – 263 с.
- 26.Богдан Воеводин. Економічний ефект від велосипеду – міф? Офіційна Web-сторінка Bike in City. [Електронний ресурс]. URL: <https://bikeincity.com.ua/ekonomika/>

27. U-Cycle. Офіційна Web-сторінка на Facebook. [Електронний ресурс]. URL: <https://www.facebook.com/photo.php?fbid=1034300368725628&id=100064370241953&set=a.621761226646213>
28. Decision Tree. Офіційна Web-сторінка Geeksforgeeks. [Електронний ресурс]. URL: <https://www.geeksforgeeks.org/decision-tree/>
29. Random Forest Algorithm. Офіційна Web-сторінка Geeksforgeeks. [Електронний ресурс]. URL: <https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/>
30. Gradient boosting. Офіційна Web-сторінка Itwiki. [Електронний ресурс]. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/gradient-boosting>
31. MLP. Що таке багат шаровий перцептрон (Multilayer Perceptron, MLP) у машинному навчанні? Офіційна Web-сторінка Thetransmitted. [Електронний ресурс]. URL: <https://thetransmitted.com/adlucem/shho-take-mlp-u-mashynnomu-navchanni/>
32. Support Vector Machines(SVM). Офіційна Web-сторінка Scikit-learn. [Електронний ресурс]. URL: <https://scikit-learn.org/stable/modules/svm.html>
33. StandardScaler. Офіційна Web-сторінка Scikit-learn. [Електронний ресурс]. URL: <https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>
34. Що таке PostgreSQL і для чого використовується? Офіційна Web-сторінка Foxminded. [Електронний ресурс]. URL: <https://foxminded.ua/postgresql-shcho-tse/>
35. Основи HTML, CSS, JavaScript. Офіційна Web-сторінка Pravda.if.ua. [Електронний ресурс]. URL: <https://pravda.if.ua/html-css-i-javascript-osnovy-veb-rozrobky/>

Додаток А

Протокол перевірки на плагіат

(Обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка інтелектуальної комп'ютерної системи прокату велосипедів

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

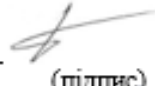
Підрозділ кафедра КСУ
(кафедра, факультет, навчальна група)

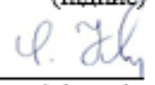
Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 6,3 %

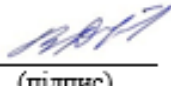
Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Завідувач кафедри КСУ  В'ячеслав КОВТУН
(прізвище, ініціали, посада) (підпис)

Гарант ОПП  Олег КОВАЛЮК
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку  Володимир ДУБОВОЙ
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник  Тетяна ГРИЩУК
(підпис) (прізвище, ініціали, посада)

Здобувач  Олександр КАБАШНИЙ
(підпис) (прізвище, ініціали)

Додаток Б
Технічне завдання
(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., проф.



В'ячеслав КОВТУН

“ 22 ” травня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання бакалаврської кваліфікаційної роботи
РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОКАТУ
ВЕЛОСИПЕДІВ

Студент групи 2АКІТ-216



Олександр КАБАШНИЙ

Керівник к.т.н., доцент



Тетяна ГРИЩУК

Вінниця 2025

1. Назва та галузь застосування

1.1. Назва – Розробка інтелектуальної комп’ютерної системи прокату велосипедів.

Частина 1. Програмна частина.

1.2. Галузь застосування – інформаційні технології у сфері сервісних послуг і транспортної логістики.

2. Підстава для проведення розробки.

Тема бакалаврської кваліфікаційної роботи затверджена наказом по ВНТУ **№_97_ від _20 березня_.2025 р.**

3. Мета та призначення розробки.

Метою розробки є створення інтелектуальної системи для управління прокатом велосипедів, що забезпечує онлайн-бронювання, контроль технічного стану та прогнозування попиту й поломок. Система призначена для автоматизації ключових процесів велопрокату: облік замовлень, обслуговування, аналітика. ШІ-модулі підвищують ефективність керування та зручність для користувачів і адміністраторів.

4. Джерела розробки.

Бакалаврська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. В.М. Дубовой. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології». Вінниця : ВНТУ, 2022. URL <https://iq.vntu.edu.ua/repository/card.php?id=5929&lang=uk>

2. Гришук Т. В., Юркіна О. М. Комп’ютерна програма «Маршрути громадського транспорту» : авторське свідоцтво № 133380. – Зареєстр. в Держ. реєстрі комп’ютерних програм 11.02.2025 бюлетень № 87 від 31.03.2025.

3. В. Б. Мокін, М. В. Дратованій. Наука про дані: машинне навчання та інтелектуальний аналіз даних Електронний навчальний посібник /– Вінниця : ВНТУ, 2024. – 263 с.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- реєстрація користувачів та онлайн-бронювання велосипедів;
- облік наявних велосипедів та їх технічного стану;

- прогноз попиту на оренду з урахуванням погоди й днів тижня;
- прогноз технічних несправностей на основі історії обслуговування;
- адміністрування замовлень;
- збереження й перегляд історії оренд у профілі користувача.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- Операційна система Windows 10;

5.2.2. Умови експлуатації системи:

- можливість стабільного функціонування системи при наявному підключенні до інтернету;
- доступ до мережі Інтернет через веб-браузер;

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми кваліфікаційної роботи.
Постановка задач дослідження «10» травня 2025 р.
2. Визначення технічних характеристик системи «12» травня 2025 р.
3. Розробка програмного забезпечення системи «30» травня 2025 р.

6.2 Графічні матеріали:

1. Розробка моделі системи «24» травня 2025 р.
2. Тестування програмного забезпечення «5» червня 2025 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «4» червня 2025 р.

- 7.2. Атестація проєкту здійснюється на попередньому захисті. Попередній захист бакалаврської кваліфікаційної роботи провести до «10» червня 2025 р.

- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист бакалаврської кваліфікаційної роботи провести до «20» червня 2025 р.

Додаток В
(ДОВІДКОВИЙ)
Лістинг коду
train_failure_model.py

```
import pandas as pd
import joblib
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score, precision_score,
recall_score, f1_score
from xgboost import XGBClassifier

df =
pd.read_csv("D:/!HEAVEN/rentheaven/diagnostics/datasets/bike_failures/failure_datase
t_future_10000.csv")

X = df.drop(columns=["failure_next_n_km"])
y = df["failure_next_n_km"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = XGBClassifier(
    use_label_encoder=False,
    eval_metric='logloss',
    n_estimators=300,
    max_depth=4,
    random_state=42
)
```

```

model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print(" Критерії якості моделі:")
print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
print(f"Precision: {precision_score(y_test, y_pred):.4f}")
print(f"Recall: {recall_score(y_test, y_pred):.4f}")
print(f"F1 Score: {f1_score(y_test, y_pred):.4f}")
print("\nПовний звіт:")
print(classification_report(y_test, y_pred))

```

```

joblib.dump(model,
"D:/!HEAVEN/rentheaven/diagnostics/bike_ai/failure_predictor_model.pkl")
print(" Модель збережено.")

```

train_demand_model.py

```

import pandas as pd
import joblib
import os
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATA_PATH = os.path.join(BASE_DIR,
'../datasets/bike_sharing/generated_demand_hourly_contrast.csv')
MODEL_PATH = os.path.join(BASE_DIR, 'demand_model.pkl')

```

```
R2_PATH = os.path.join(BASE_DIR, 'demand_r2.txt')
df = pd.read_csv(DATA_PATH)
```

```
df['date'] = pd.to_datetime(df['datetime']).dt.date
df_daily = df.groupby('date').agg({
    'temp': 'mean',
    'humidity': 'mean',
    'windspeed': 'mean',
    'season': 'first',
    'holiday': 'first',
    'workingday': 'first',
    'weather': lambda x: x.mode()[0],
    'count': 'sum'
}).reset_index()
```

```
features = df_daily[['temp', 'humidity', 'windspeed', 'season', 'holiday', 'workingday',
'weather']]
target = df_daily['count']
```

```
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

```
X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.2,
random_state=42)
```

```
model = XGBRegressor(
    objective='reg:squarederror',
    n_estimators=300,
    max_depth=6,
    learning_rate=0.05,
    random_state=42
)
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
```

```
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f'R2 Score: {r2:.3f}')
print(f'MAE: {mae:.3f}')
print(f'RMSE: {rmse:.3f}')

joblib.dump(model, MODEL_PATH)
with open(R2_PATH, 'w', encoding='utf-8') as f:
    f.write(f'R2: {r2:.3f}\nMAE: {mae:.3f}\nMSE: {mse:.3f}\nRMSE: {rmse:.3f}')

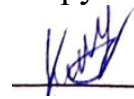
print(f'Модель збережена за адресою: {MODEL_PATH}')
```

Додаток Г
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОКАТУ ВЕЛОСИПЕДІВ

Студент групи 2АКІТ-216



Олександр КАБАШНИЙ

Керівник: к.т.н., доцент каф. КСУ



Тетяна ГРИЩУК

Схематична структура автоматизованої системи прокату велосипедів.

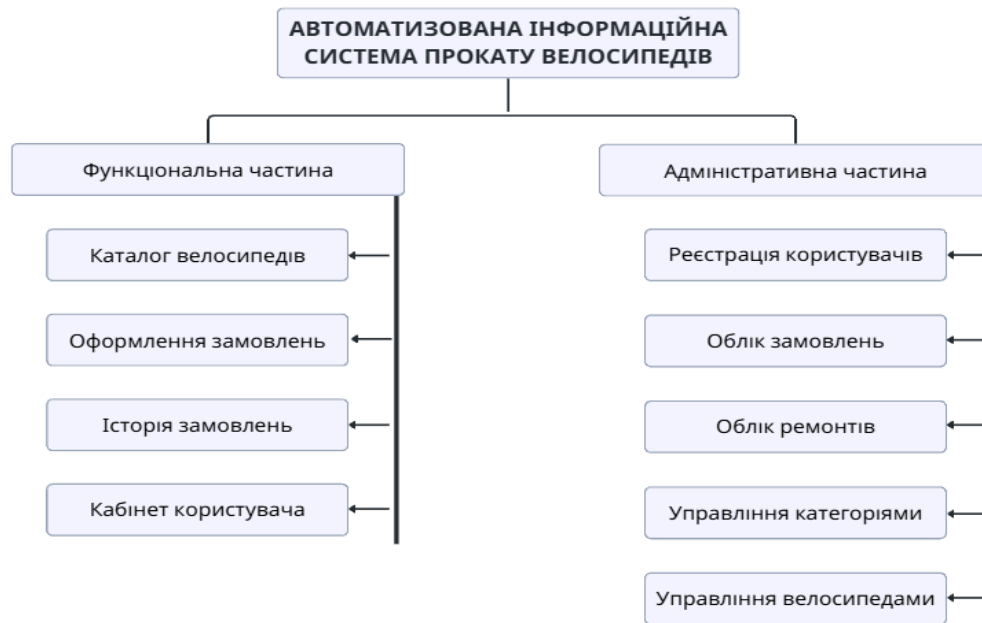


Рисунок Г.1 - Структура автоматизованої системи прокату велосипедів

UML-діаграма варіантів використання системи

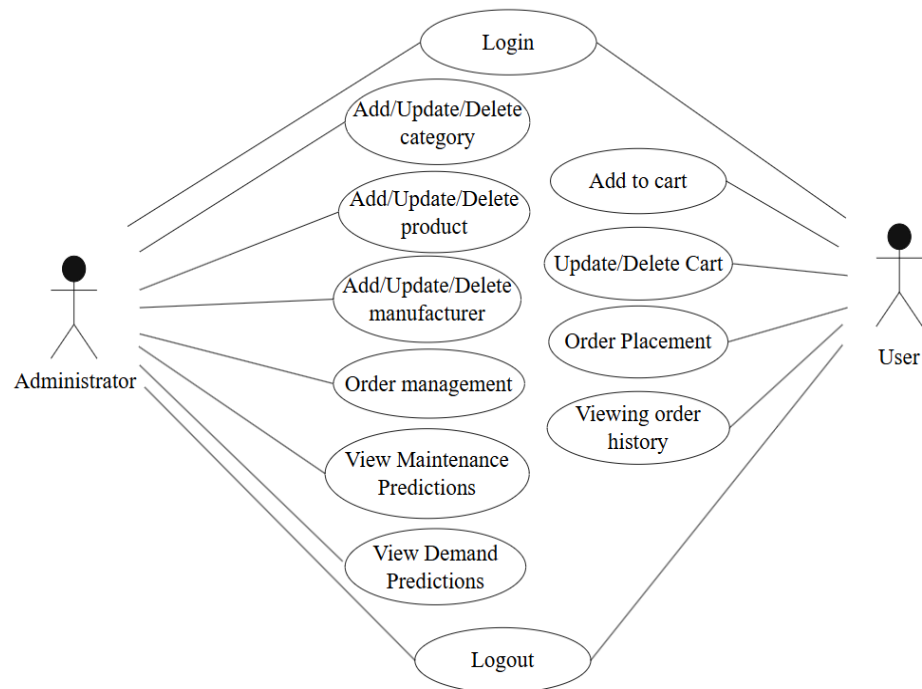


Рисунок Г.2 – UML-діаграма варіантів використання системи

Діаграма діяльності.

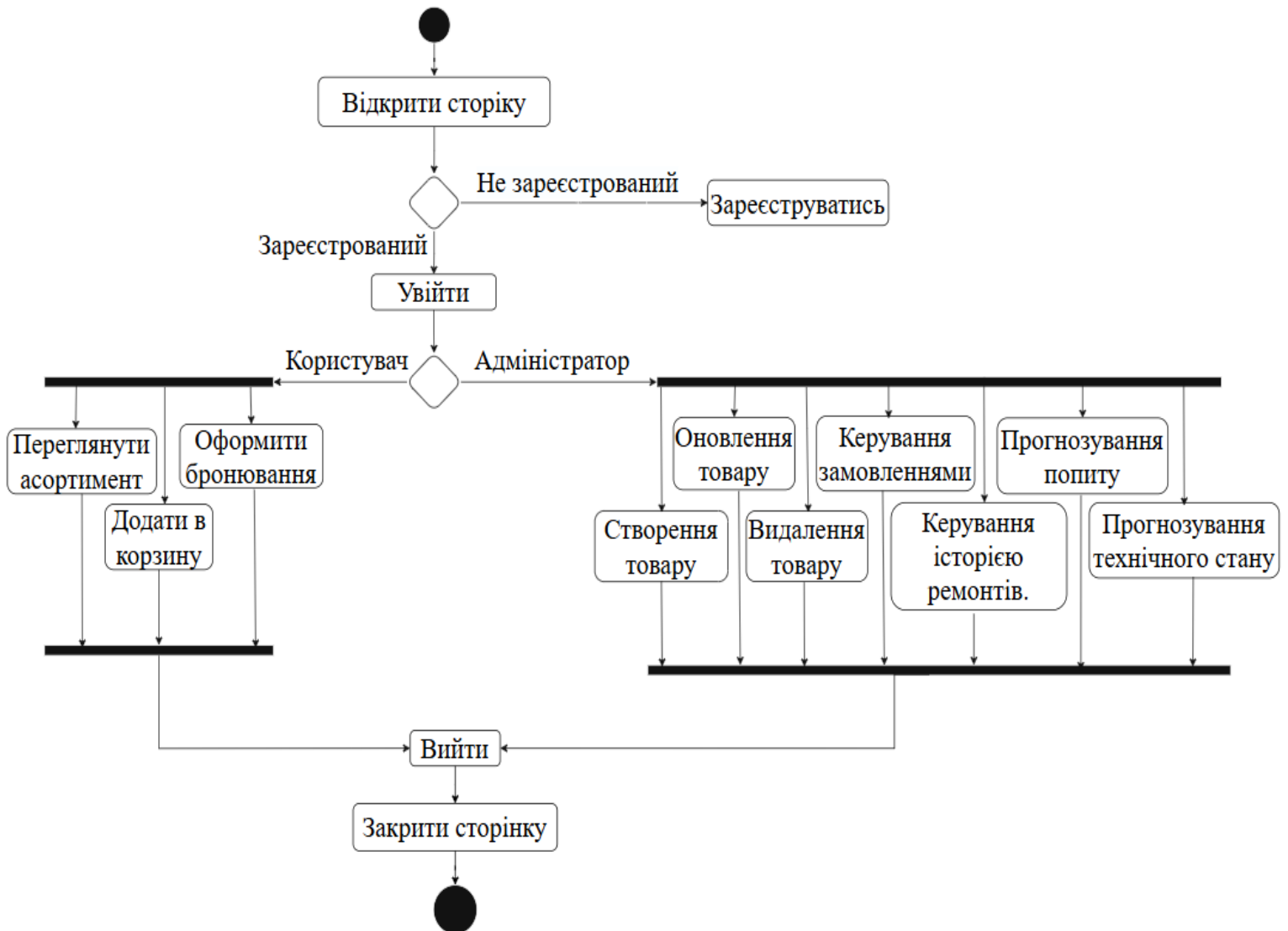


Рисунок Г.3 - Діаграма діяльності

Діаграма послідовності

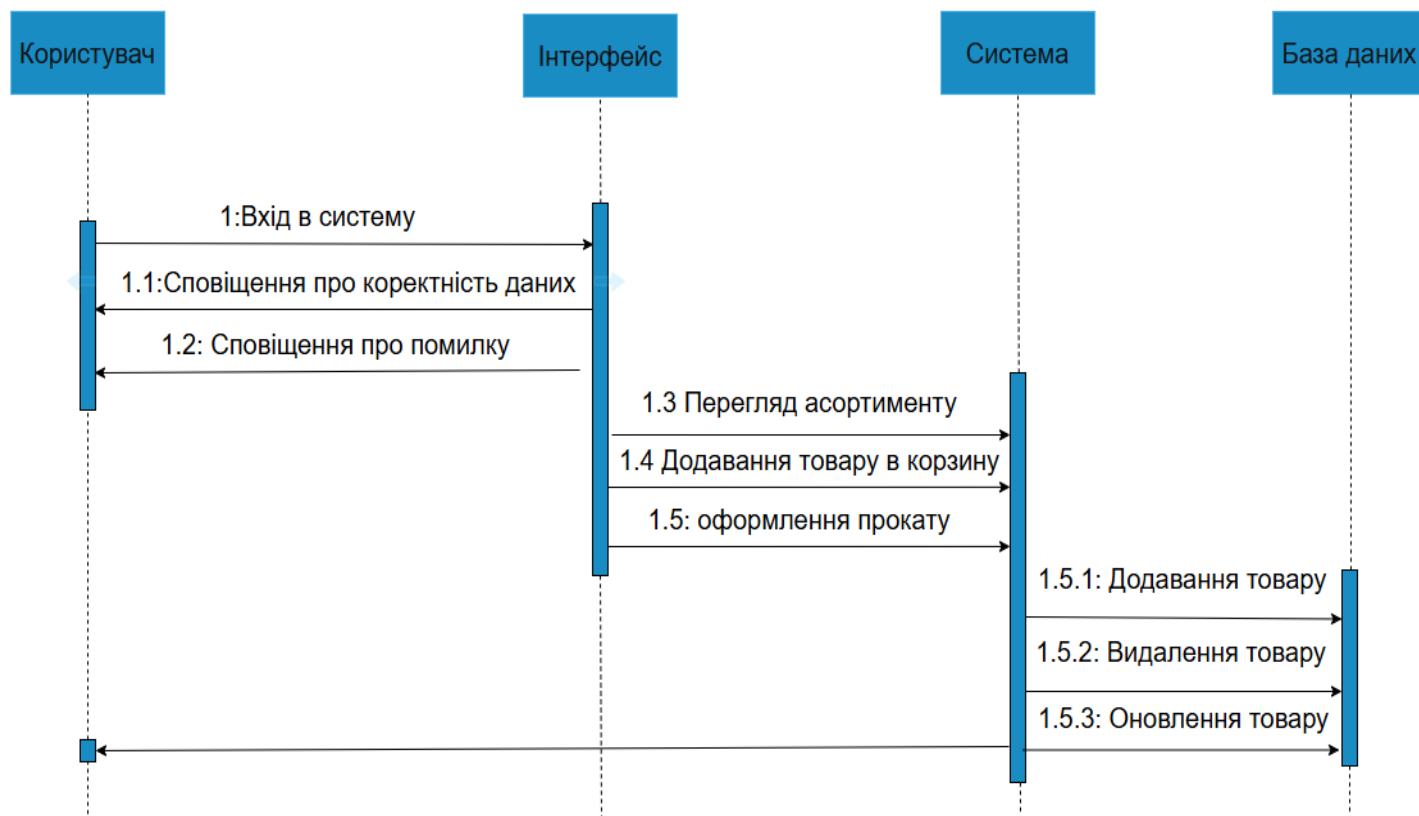


Рисунок Г.4 - Діаграма послідовності

Прогнозування відмови транспорту

Ознаки: Кількість відмов гальм за останні 1000 км, кількість відмов ланцюга за останні 1000 км, кількість відмов шин за останні 1000 км, кількість відмов перемикача передач за останні 1000 км, кількість поїздок під дощем, середня температура під час поїздок, середня вологість повітря під час поїздок, кількість поломок за останній рік, факт проведення технічного обслуговування, вік велосипеда в днях, загальний пробіг велосипеда; цільова змінна — імовірність виникнення поломки найближчим часом.

```

=== Крос-валідація (F1-метрика, 5 фолдів) ===

Random Forest:
  F1-оцінки: [0.9524838  0.95276248 0.95303131 0.9567536  0.9572237 ]
  Середнє F1: 0.9545
-----
XGBoost:
  F1-оцінки: [0.98295876 0.98372514 0.98463056 0.98551679 0.98576952]
  Середнє F1: 0.9845
-----
MLP:
  F1-оцінки: [0.97487437 0.96399263 0.87775953 0.89566237 0.86025916]
  Середнє F1: 0.9145
-----

```

Рисунок Г.5 Крос валідація трьох обраних моделей

Bicycles Overview						
ID	Name	Size	Current Mileage	Failure Probability (next 200 km)	Status	Details
19	Mountain Explorer	medium	440 km	8%	Usable	View
20	Mountain Explorer	small	1870 km	15%	Usable	View
22	Mountain Explorer	large	920 km	31%	Usable	View
23	City Cruiser	medium	3230 km	52%	The bicycle needs to be checked	View
24	City Cruiser	medium	1580 km	70%	The bicycle needs to be checked	View
25	Road Racer	small	1006 km	20%	Usable	View
26	Road Racer	large	1007 km	77%	The bicycle needs to be checked	View
27	Road Racer	medium	4008 km	90%	The bicycle needs to be checked	View

Рисунок Г.6 прогноз ймовірність відмови транспортного засобу

Прогнозування попиту оренди

Вхідні ознаки : temp(°C), humidity(%), windspeed(м/с), season (1 - весна, 2 - літо), holiday(0 або 1), workingday(0 або 1), weather(погодний код). Цільова змінна count(кількість оренд):

```

=== Крос-валідація (5 фолдів) ===

Linear Regression:
MAE: [2.76, 2.62, 2.98, 2.66, 2.87] → Середнє: 2.78
RMSE: [np.float64(3.4), np.float64(3.24), np.float64(3.49), np.float64(3.16), np.float64(3.44)] → Середнє: 3.35
R²: [0.913, 0.922, 0.916, 0.934, 0.923] → Середнє: 0.922
-----
Random Forest:
MAE: [1.94, 1.89, 1.7, 1.83, 1.79] → Середнє: 1.83
RMSE: [np.float64(2.31), np.float64(2.27), np.float64(2.07), np.float64(2.19), np.float64(2.16)] → Середнє: 2.20
R²: [0.96, 0.962, 0.97, 0.968, 0.97] → Середнє: 0.966
-----
XGBoost:
MAE: [1.95, 1.82, 1.71, 1.81, 1.82] → Середнє: 1.82
RMSE: [np.float64(2.33), np.float64(2.26), np.float64(2.05), np.float64(2.17), np.float64(2.19)] → Середнє: 2.20
R²: [0.959, 0.962, 0.971, 0.969, 0.969] → Середнє: 0.966
-----

```

Рисунок Г.7 Крос валідація трьох обраних моделей

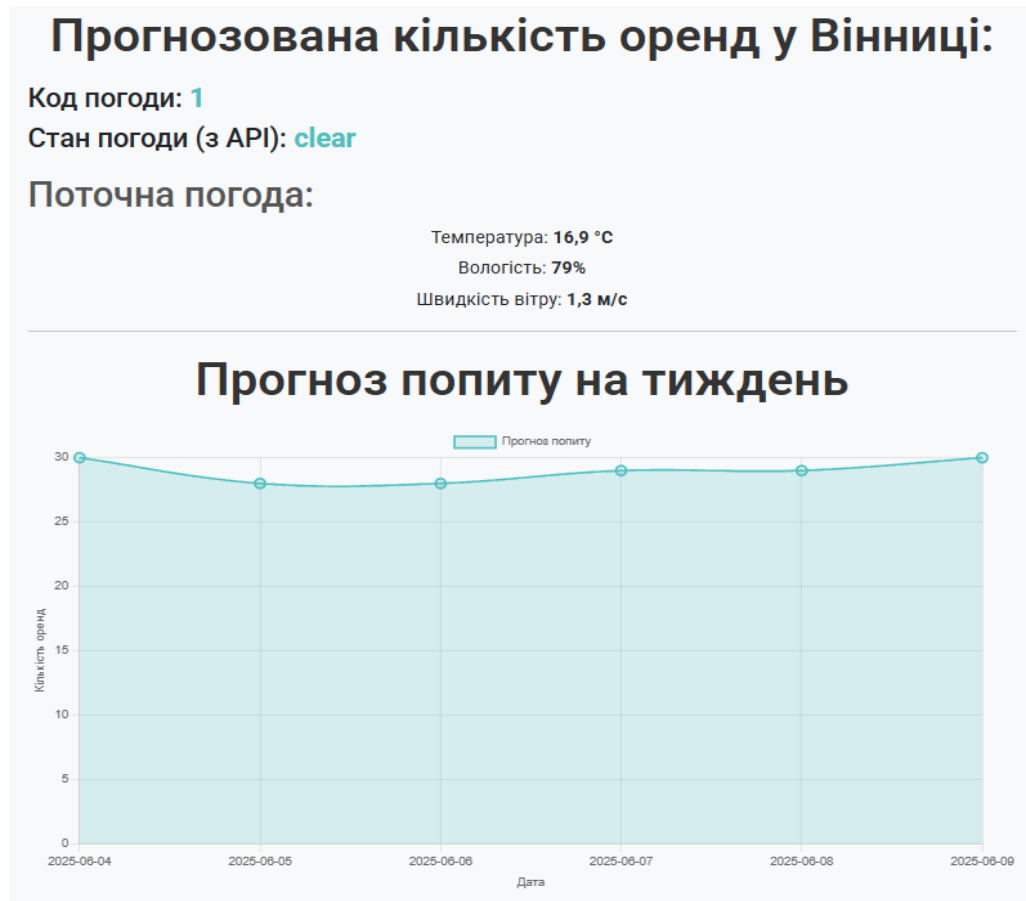


Рисунок Г.8 – Прогноз попиту

ER-модель бази даних



Рисунок Г.9 - ER-модель бази даних

Головна сторінка сайту

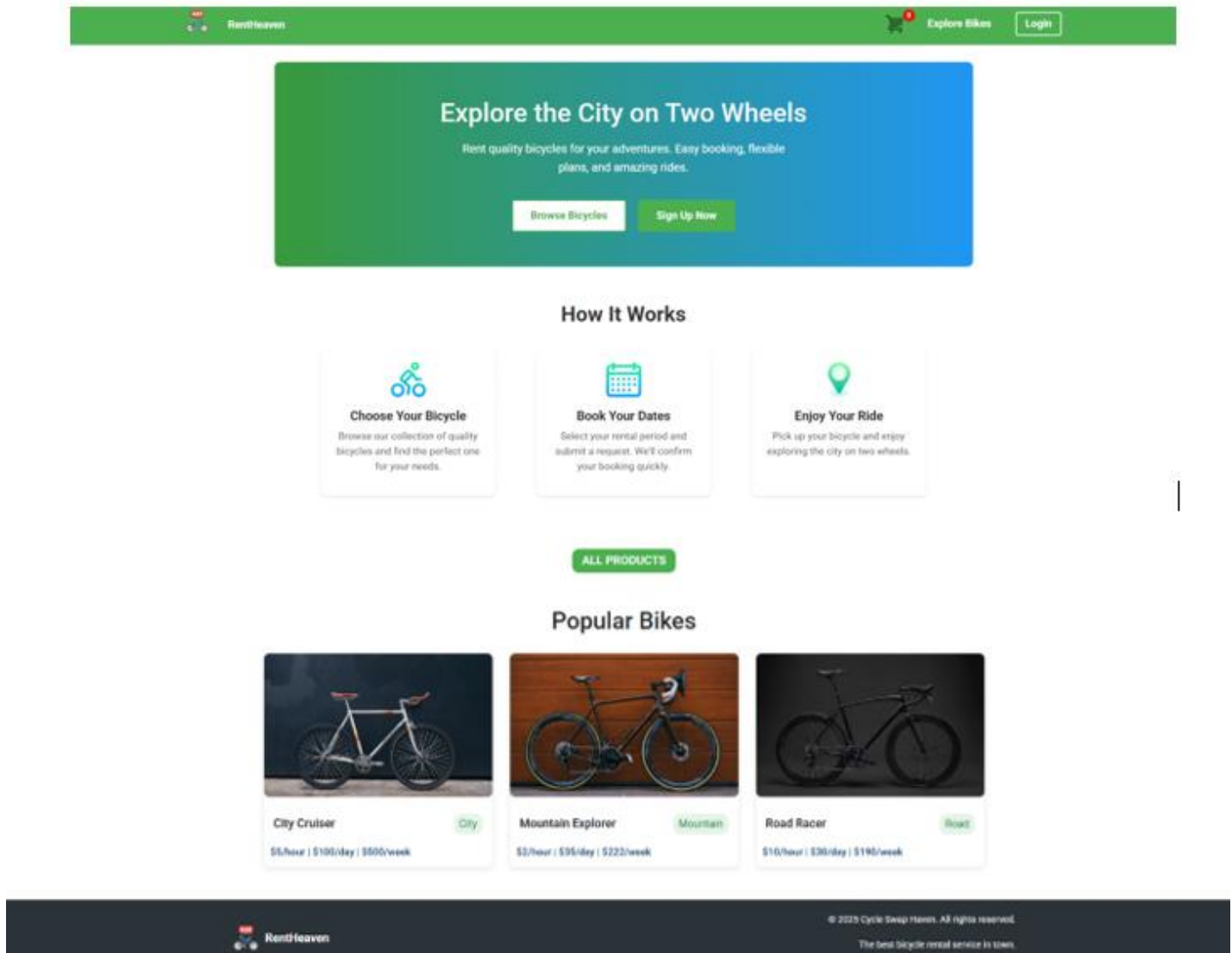
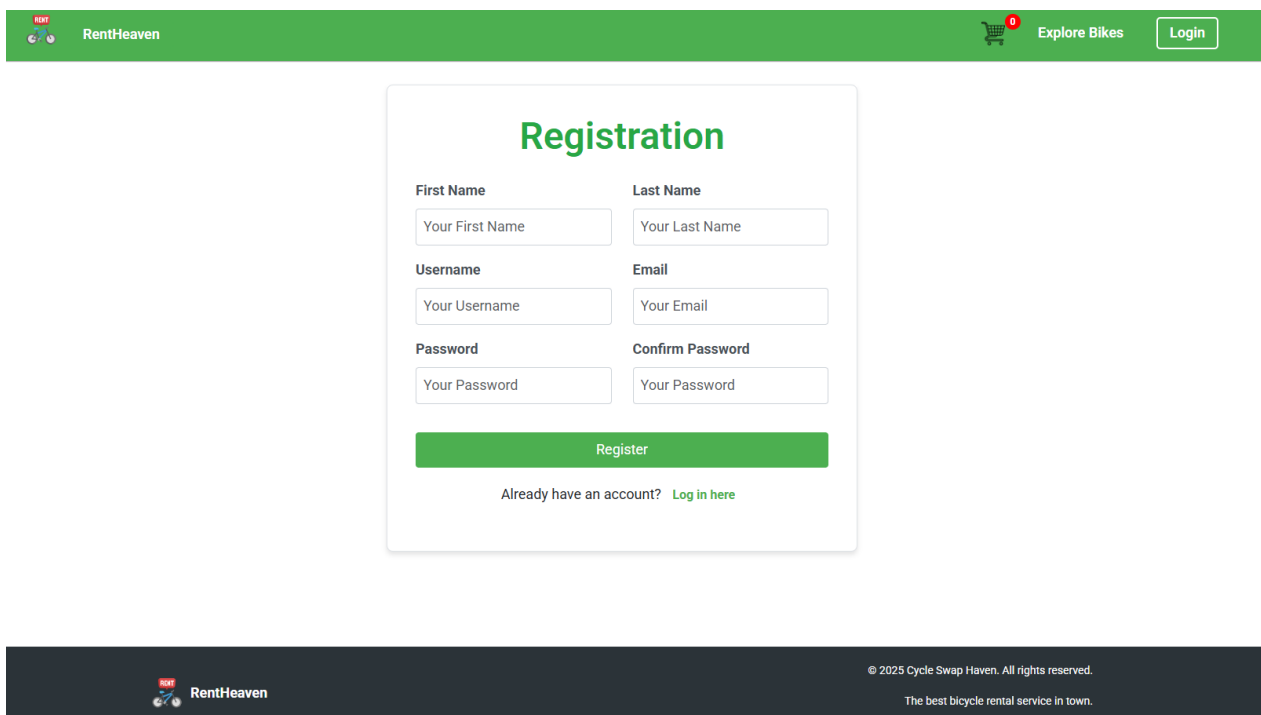


Рисунок Г.10 – Головна сторінка сайту

Форма реєстрації



The registration form is titled "Registration" in green. It is located within a white box with a light gray border. The form has a green header bar at the top with the RentHeaven logo and navigation links. The form fields are arranged in two columns: First Name, Last Name, Username, Email, Password, and Confirm Password. Each field has a placeholder text. A green "Register" button is at the bottom. Below the button is a link "Log in here" for users who already have an account. The footer of the page contains the RentHeaven logo, copyright information, and a tagline.

RentHeaven

Explore Bikes

Login

Registration

First Name

Your First Name

Last Name

Your Last Name

Username

Your Username

Email

Your Email

Password

Your Password

Confirm Password

Your Password

Register

Already have an account? [Log in here](#)

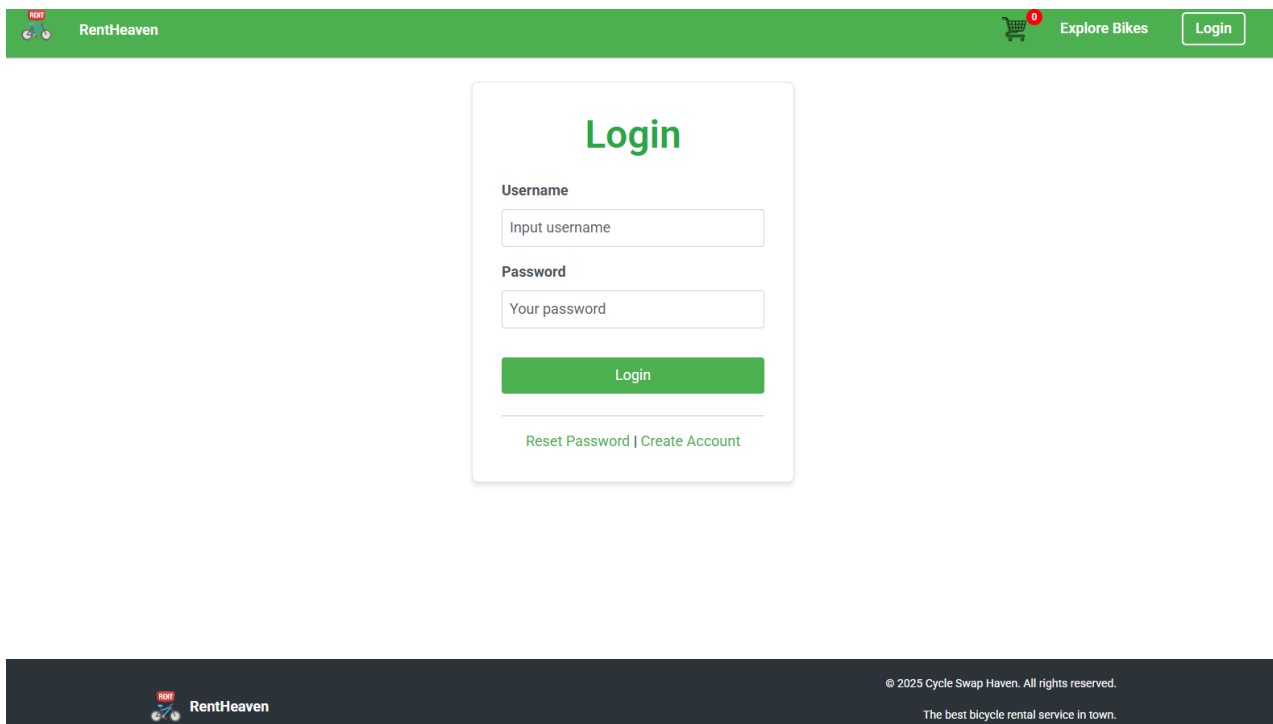
RentHeaven

© 2025 Cycle Swap Haven. All rights reserved.

The best bicycle rental service in town.

Рисунок Г.11 – Форма реєстрації

Форма логіну



The login form is titled "Login" in green. It is located within a white box with a light gray border. The form has a green header bar at the top with the RentHeaven logo and navigation links. The form fields are arranged in two columns: Username and Password. Each field has a placeholder text. A green "Login" button is at the bottom. Below the button is a link "Reset Password | Create Account" for users who need to reset their password or create a new account. The footer of the page contains the RentHeaven logo, copyright information, and a tagline.

RentHeaven

Explore Bikes

Login

Login

Username

Input username

Password

Your password

Login

[Reset Password](#) | [Create Account](#)

RentHeaven

© 2025 Cycle Swap Haven. All rights reserved.

The best bicycle rental service in town.

Рисунок Г.12 – Форма логіну

Каталог велосипедів

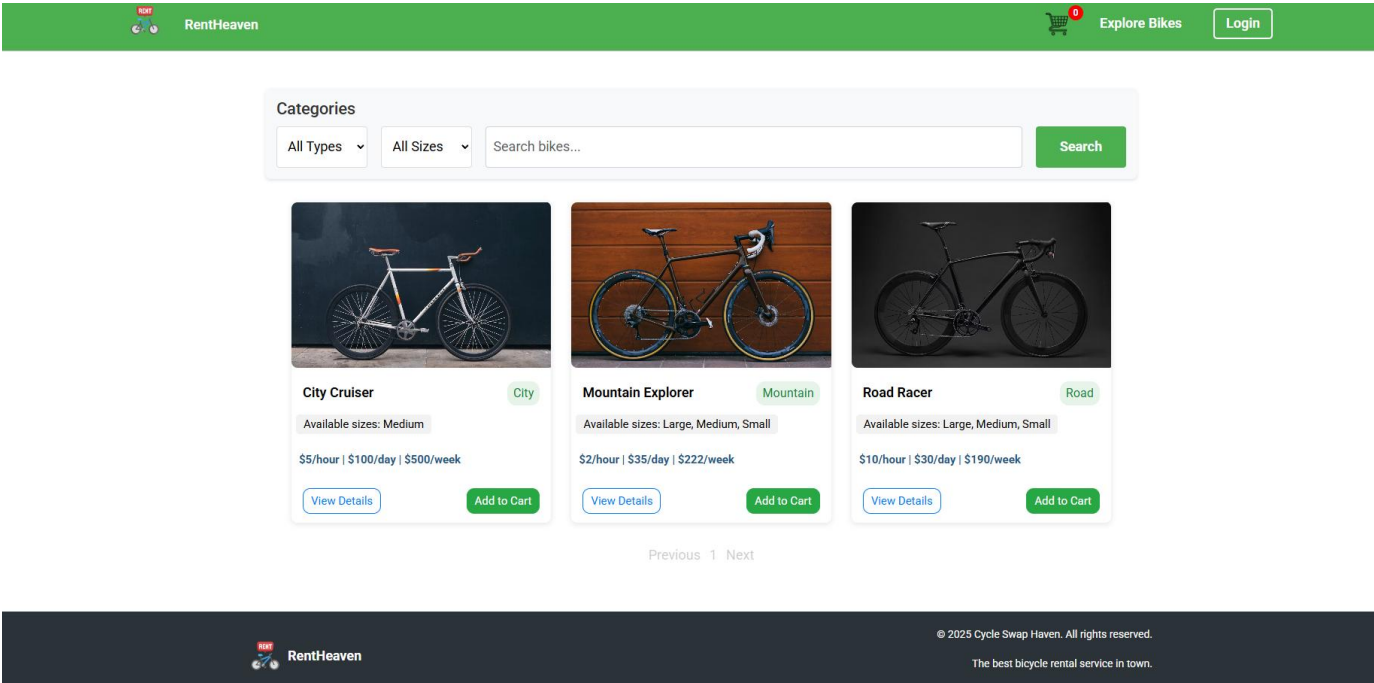


Рисунок Г.13 – Каталог велосипедів

Детальний перегляд велосипеда

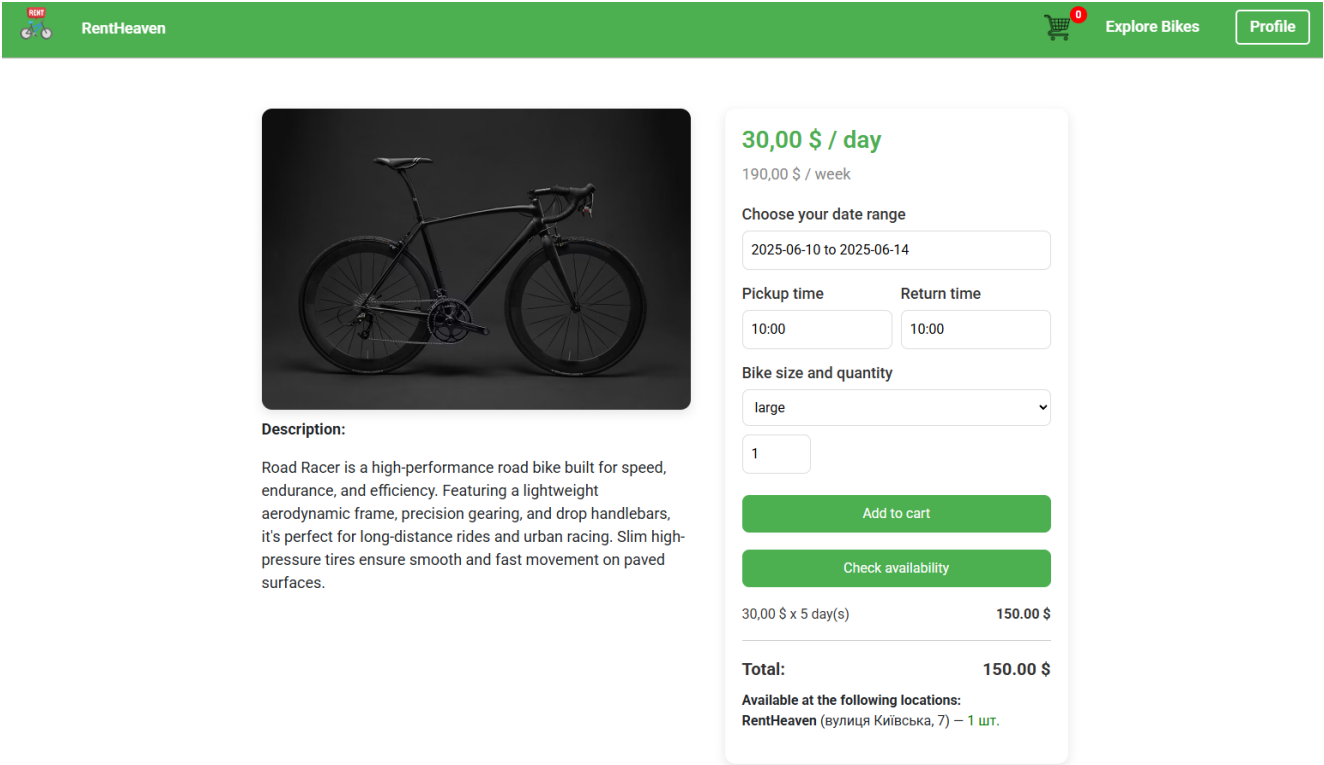


Рисунок Г.14 - Детальний перегляд велосипеда

Кошик із бронюванням

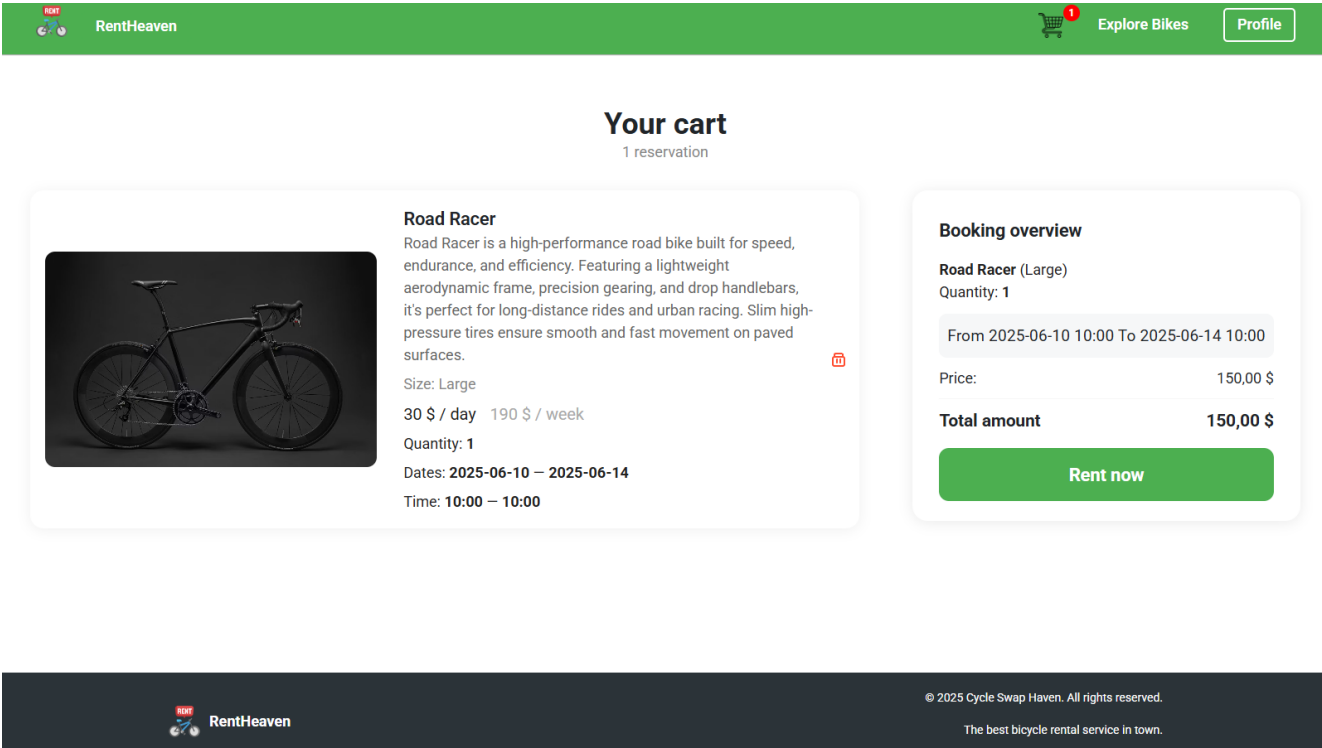


Рисунок Г.15 – Кошик

Сторінка з оформленням бронювання

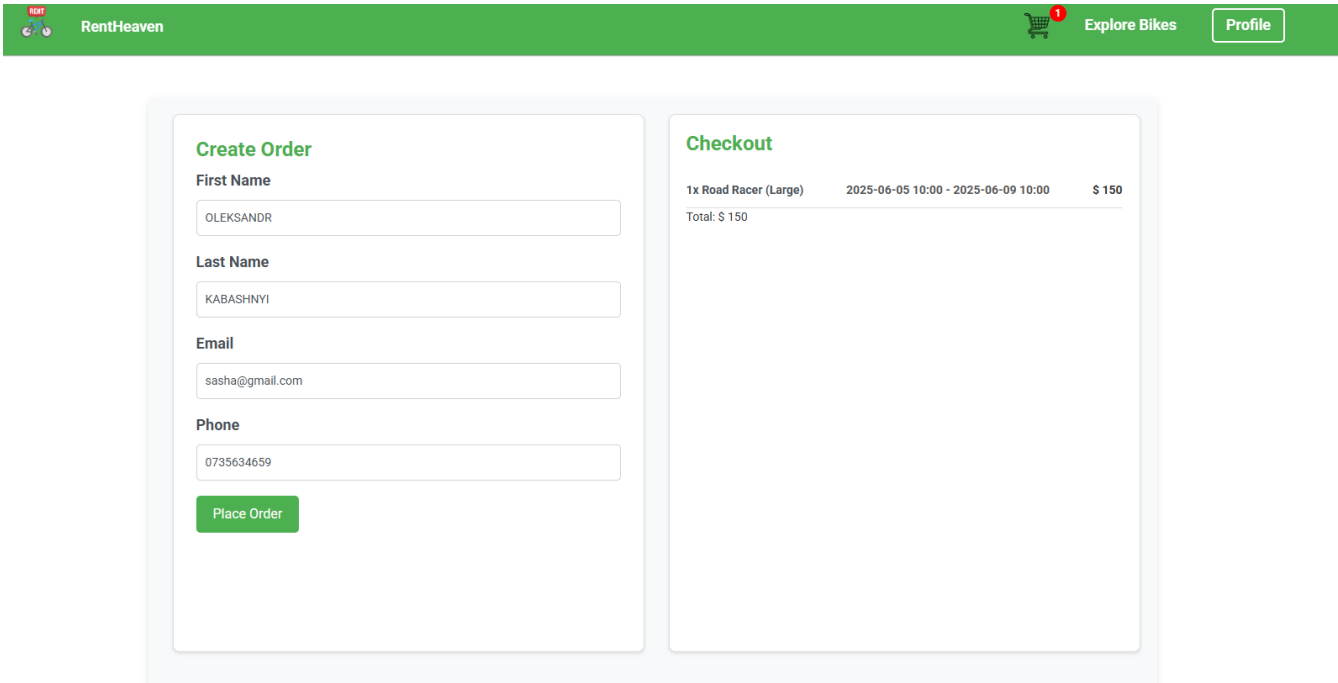


Рисунок Г.16 – Сторінка бронювання замовлення

Профіль користувача

Profile

 Вибрати файл Файл не вибрано

First Name
OLEKSANDR

Last Name
KABASHNYI

Username
admin

Email
sasha@gmail.com

[Save](#) [Logout](#)

Your Orders

Order № 10
Name: Road Racer | Size: Large
Quantity: 1, Price: \$150.00
Rental Period: 05 червня 2025 р. 10:00 – 09 червня 2025 р. 10:00
Date of order: 05 червня 2025 р. 03:15 [Cancel Order](#)

Order № 9
Name: City Cruiser | Size: Medium
Quantity: 1, Price: \$400.00
Rental Period: 04 червня 2025 р. 10:00 – 07 червня 2025 р. 10:00
Date of order: 04 червня 2025 р. 01:14 [Trip Completed](#)

Name: City Cruiser | Size: Medium
Quantity: 1, Price: \$400.00
Rental Period: 04 червня 2025 р. 10:00 – 07 червня 2025 р. 10:00
Date of order: 04 червня 2025 р. 01:14

Рисунок Г.17 – Профіль користувача

Інструменти адміністратора

Profile

 Вибрати файл Файл не вибрано

First Name
OLEKSANDR

Last Name
KABASHNYI

Username
admin

Email
sasha@gmail.com

[Save](#) [Logout](#)

Admin Tools
[Manage Bicycles](#)
[AI Forecast](#)
[Travel log](#)

Рисунок Г.18 - Інструменти адміністратора

Логування поїздок

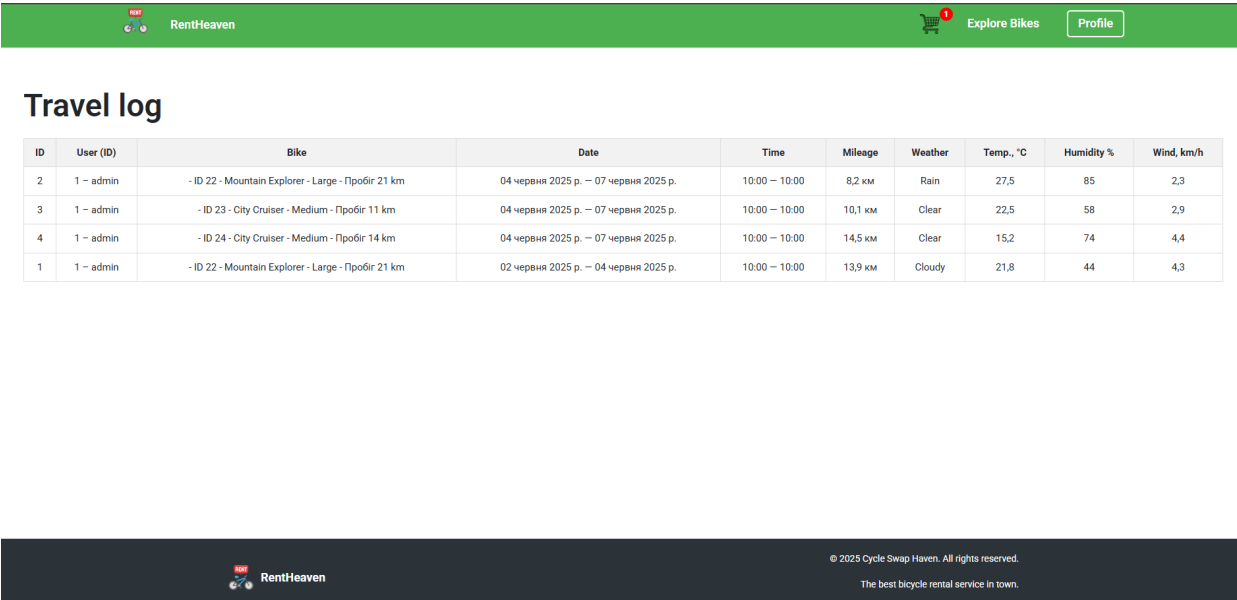
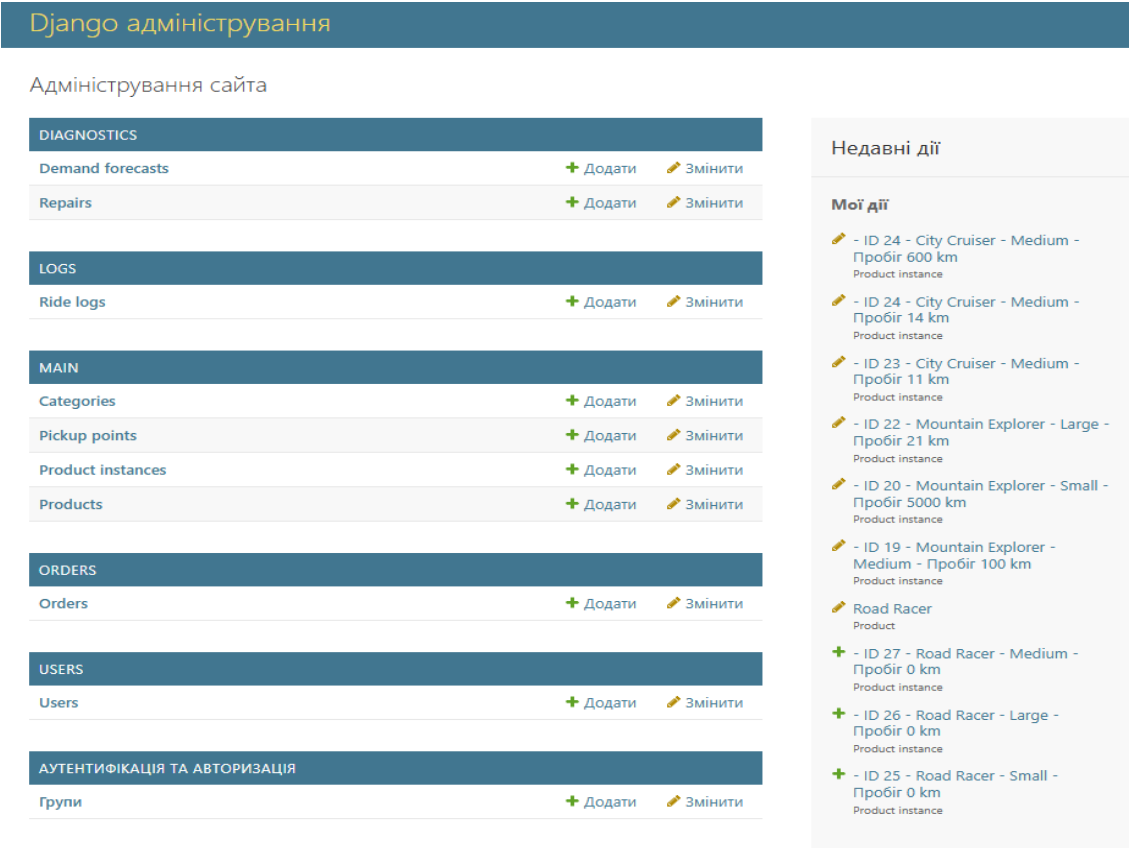


Рисунок Г.19 - Логування поїздок

Загальний вигляд Django адміністрування



Перегляд замовлень

Домівка > Orders > Orders

Почніть писати для фільтру...

DIAGNOSTICS

Demand forecasts

Додати

Repairs

Додати

LOGS

Ride logs

Додати

MAIN

Categories

Додати

Pickup points

Додати

Product instances

Додати

Products

Додати

ORDERS

Orders

Додати

USERS

Users

Додати

АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ

Групи

Додати

Виберіть order щоб змінити

Q

Пошук

Дія:

Вперед

Обрано 0 з 10

☐	ID	FIRST NAME	LAST NAME	EMAIL	PHONE	CREATED	ВЕЛОСИПЕДИ	ПЕРІОД ОРЕНДИ	ВАРТІСТЬ	СТАТУС
☐	10	OLEKSANDR	KABASHNYI	sasha@gmail.com	0735634659	05 червня 2025 р. 03:15	1x Road Racer	2025-06-05 — 2025-06-09	150,00	✗ Скасовано
☐	9	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	04 червня 2025 р. 01:14	1x City Cruiser, 1x City Cruiser	2025-06-04 — 2025-06-07; 2025-06-04 — 2025-06-07	800,00	✓ Повернено
☐	8	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	04 червня 2025 р. 01:13	1x Mountain Explorer	2025-06-04 — 2025-06-07	140,00	✓ Повернено
☐	7	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 19:12	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	6	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 19:08	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	5	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 19:05	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	4	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	02 червня 2025 р. 18:59	1x Mountain Explorer	2025-06-02 — 2025-06-04	105,00	✓ Повернено
☐	3	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	28 травня 2025 р. 02:36	1x City Cruiser, 1x City Cruiser	2025-05-28 — 2025-05-31; 2025-05-28 — 2025-05-31	400,00	✓ Повернено
☐	2	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	28 травня 2025 р. 01:47	1x Mountain Explorer	2025-05-27 — 2025-05-30	140,00	✓ Повернено
☐	1	OLEKSANDR	KABASHNYI	sasha@gmail.com	0734634669	20 травня 2025 р. 01:38	1x Mountain Explorer	2025-05-19 — 2025-05-23	175,00	✗ Скасовано

10 orders

Рисунок Г.21 – Перегляд замовлень