

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО
РЕСУРСУ КОМПАНІЇ ПЕРЕВЕЗЕНЬ ВАНТАЖУ

SOFTWARE IMPLEMENTATION OF THE INFORMATION
RESOURCE OF THE CARGO TRANSPORTATION COMPANY

Виконав: студент 4 курсу, групи КНТ-219

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

КОТЛЯРОВ К.К.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання «04» квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

Підпис Кирило КОТЛЯРОВ
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

Михайло КОЦУР
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
86 с., 39 рис., 3 табл., 4 дод., 25 джерел.

ІНФОРМАЦІЙНИЙ РЕСУРС, ПЕРЕВЕЗЕННЯ, ВАНТАЖ, РОЗРОБКА, ВЕБЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Об'єкт дослідження – процес програмної реалізації інформаційного ресурсу компанії перевезень вантажу.

Предмет дослідження – програмні засоби для реалізації інформаційного ресурсу компанії перевезень вантажу.

Метою роботи є програмна реалізація інформаційного ресурсу компанії перевезень вантажу у вигляді кроссплатформного вебзастосунку.

Матеріали, методи та технічні засоби: мова програмування PHP та система керування вмістом WordPress, засоби проєктування UML.

Результати. Розроблено інформаційний ресурс компанії перевезень вантажу.

Практична цінність роботи полягає у програмній реалізації інформаційного ресурсу компанії перевезень вантажу у вигляді кроссплатформного вебзастосунку.

Висновки. Реалізовано інформаційний ресурс компанії перевезень вантажу у вигляді кроссплатформного вебзастосунку.

Галузь використання – компанії, що потребують перевезення вантажу, люди, що змінюють житло.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 86 pages, 39 figures, 3 tables, 4 appendixes, 25 sources.

INFORMATION RESOURCE, TRANSPORTATION, CARGO, DEVELOPMENT, WEB APPLICATION, PROGRAMMING LANGUAGE, SOFTWARE

The object of research is the process of software implementation of the information resource of a cargo transportation company.

The subject of the research is software tools for implementing the information resource of a cargo transportation company.

The aim of the work is to programmatically implement the information resource of the cargo transportation company in the form of a cross-platform web application.

Materials, methods, and technical tools: PHP programming language and WordPress content management system, UML design tools.

Results. An information resource of the cargo transportation company has been developed.

The practical value of the work lies in the software implementation of the information resource of the cargo transportation company in the form of a cross-platform web application.

Conclusions. Implemented an information resource of the cargo transportation company in the form of a cross-platform web application.

Industry of use: companies that need cargo transportation, people who change their homes.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області	12
1.1 Бізнес з надання послуг перевезення вантажів.....	12
1.2 Використання застосунків для сфери логістики	17
1.3 Порівняння існуючих аналогів.....	18
1.3.1 Сервіс «Uber Freight»	20
1.3.2 Сервіс «Webfleet»	22
1.3.3 Сервіс «Нова пошта»	24
1.3.4 Порівняння існуючих аналогів	25
1.4 Постановка завдання	27
1.4.1 Функціональність системи	27
1.4.2 Вимоги до інтерфейсу.....	28
1.4.3 Вимоги до продуктивності	28
1.4.4 Вимоги до безпеки	28
1.4.5 Експлуатаційні вимоги	29
1.4.6 Не функціональні вимоги до продукту	29
1.5 Висновки за розділом 1	29
2 Матеріали і методи	30
2.1 Вибір мови програмування	30
2.2 Вибір та обґрунтування системи керування вмістом.....	33
2.3 Структурна схема розробки.....	36
2.4 Висновки за розділом 2	37
3 Основні рішення щодо реалізації компонентів системи	38
3.1 Проектування моделей системи	38
3.2 Проектування інтерфейсу	48
3.3 Висновки за розділом 3	51
4 Експлуатація, тестування та експериментальне дослідження програми.....	52

4.1 Призначення й умови застосування програми	52
4.1.1 Вимоги до інтерфейсу.....	53
4.2 Умови виконання програми.....	53
4.3 Розгортання застосунку.....	54
4.3.1 Звернення до програми.....	56
4.3.2 Вхідні й вихідні дані	57
4.4 Методика та результати тестування	57
Висновки	60
Перелік джерел посилання	61
Додаток А Технічне завдання	64
Додаток Б Опис програми	69
Додаток В Текст програми	74
Додаток Д Слайди презентації	80

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

СКБД – система керування базами даних;

MVC – model-view-controller;

SEO – пошукова оптимізація;

UML – Unified Modeling Language.

ВСТУП

Вебзастосунки відіграють важливу роль в управлінні інформаційними ресурсами компаній, що займаються вантажоперевезеннями. Перевезення вантажів-це складний процес, який вимагає ретельного планування та управління для забезпечення безпечної та ефективної доставки вантажів. Вебзастосунок може стати потужним інструментом для компаній, що займаються вантажоперевезеннями, для оптимізації своєї діяльності, покращення досвіду клієнтів та оптимізації бізнес-показників [1]–[3].

Однією з головних переваг вебпрограми для компаній, що займаються вантажоперевезеннями, є те, що вона надає централізований центр для управління інформацією, пов'язаною з операціями з перевезення вантажів. Сюди входить інформація про поставки, маршрути, розклад, ціни та багато іншого. За допомогою вебпрограми вантажні компанії можуть відстежувати свою діяльність у режимі реального часу, що дозволяє їм приймати швидкі та обґрунтовані рішення. Це може допомогти підвищити операційну ефективність і швидкість реагування на потреби клієнтів [1]–[3].

Наприклад, вебзастосунок можна використовувати для відстеження місця відправлення в режимі реального часу, дозволяючи вантажним компаніям вносити корективи в маршрути та розклади за необхідності. Він також може бути використаний для управління рівнем запасів, гарантуючи, що компанії, що займаються вантажоперевезеннями, мають під рукою необхідну кількість запасів для задоволення попиту клієнтів. Крім того, вебзастосунок можна використовувати для управління інформацією про ціни та виставлення рахунків, спрощуючи процес виставлення рахунків та гарантуючи, що з клієнтів стягується точна та своєчасна плата [1]–[3].

Вебзастосунок також може покращити досвід клієнтів, надаючи їм легкий доступ до інформації про їх відправлення та статус вантажу. Клієнти можуть використовувати вебзастосунок для відстеження своїх відправлень у режимі реального часу, отримання оновлень статусу доставки та управління

інформацією про свій обліковий запис. Це може підвищити задоволеність і лояльність клієнтів, що призведе до збільшення доходів бізнесу [1]–[3].

Крім того, вебзастосунок може надати компаніям, що займаються вантажоперевезеннями, цінні дані та інформацію про їх діяльність. Збираючи та аналізуючи дані про постачання, маршрути та поведінку клієнтів, компанії можуть визначити сфери вдосконалення та оптимізувати свою діяльність, щоб зменшити витрати та підвищити продуктивність. Наприклад, вебзастосунок може допомогти компаніям, що займаються вантажоперевезеннями, виявити сфери неефективності у своїй діяльності та вжити коригувальних заходів для оптимізації своїх процесів. Вони також можуть аналізувати дані про поведінку та уподобання клієнтів для розробки цільових маркетингових стратегій, які збільшують дохід та утримують клієнтів [1]–[3].

Крім підвищення операційної ефективності та якості обслуговування клієнтів, вебзастосунок також може допомогти забезпечити безпечне і надійне проведення операцій з перевезення вантажів. За допомогою вебзастосунку компанії з перевезення вантажів можуть вести актуальний облік своїх операцій і дотримуватися нормативних вимог. Це може допомогти мінімізувати ризики та відповідальність компанії [1]–[3].

На закінчення слід зазначити, що вебзастосунок для інформаційного ресурсу компаній з перевезення вантажів має важливе значення для підвищення операційної ефективності, поліпшення якості обслуговування клієнтів і оптимізації бізнес-показників. За допомогою вебпрограми вантажні компанії можуть централізовано та ефективно керувати інформацією, пов'язаною з відправленнями, маршрутами, розкладом, ціноутворенням тощо. Вони також можуть надати клієнтам легкий доступ до інформації про їх постачання та підвищити точність та своєчасність виставлення рахунків. Крім того, вебзастосунок може надавати цінні дані та аналітичну інформацію для оптимізації бізнес-показників та забезпечення безпечного та надійного виконання операцій з перевезення вантажів [1]–[3].

Отже актуальною задачею залишається програмна реалізація інформаційного ресурсу компанії перевезень вантажу.

Для виконання такої задачі необхідно виконати наступні завдання:

- проаналізувати предметну область послуг з перевезення вантажу;
- дослідити існуючі аналоги інформаційних ресурсів компаній перевезень вантажу;
- визначивши переваги та недоліки існуючих аналогів сформувати перелік функціональних вимог до нового інформаційного ресурсу, що буде якісно відрізнятися від аналогів;
- визначити технічні засоби програмної реалізації нового рішення;
- виконати проєктування розробки нового застосунку;
- на основі виконаного проєктування реалізувати новий інформаційний ресурс компанії перевезень вантажу;
- дослідити та протестувати нове рішення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Перший розділ буде присвячено дослідженню проблемної області надання послуг з перевезення вантажу. Відповідно, окрему увагу буде приділено аналізу існуючих інформаційних ресурсів таких компаній, що розміщені в Інтернеті. Дослідивши їх та визначивши головні переваги та недоліки буде сформовано ряд вимог до нового застосунку, що буде розроблятися. Таким чином новий інформаційний ресурс матиме усі шанси стати конкурентоспроможним та якісним.

1.1 Бізнес з надання послуг перевезення вантажів

Сучасний транспортний бізнес охоплює широкий спектр послуг-від персональної відправки та послуг з переміщення та зберігання до великомасштабних вантажних операцій. Залежно від рівня вашої зацікавленості як майбутнього власника нового бізнесу, у вас є безліч варіантів [4]–[6].

Категорія транспортних компаній широка, що означає, що можна спеціалізуватися на одному виді бізнесу або ключовій галузі.

Транспортний бізнес – це вид бізнесу, який займається перевезенням товарів або пасажирів з одного місця в інше. Вони можуть надавати послуги окремим пасажиром, іншим компаніям або глобальним торговим партнерам. Транспортний бізнес може спеціалізуватися на ряді видів діяльності, включаючи індивідуальні послуги з організації поїздок, перевезення споживчих товарів або міжнародну доставку витратних матеріалів і продуктів [4]–[6].

Найчастіше Транспортний бізнес класифікується за типом клієнтської бази, яку він обслуговує, або за типом звичайних послуг, які він надає [4]–[6].

Види транспортного бізнесу.

Для цілей цього керівництва ми розділимо транспортні підприємства на три основні категорії [4]–[6].

Видами транспортного бізнесу є [4]–[6]:

- особистий транспорт: сюди входять індивідуальні послуги, що надаються окремим особам або невеликим групам, включаючи такі послуги, як таксі, можливість спільного катання та лімузини;

- місцевий транспорт: ця категорія обслуговує регіональні ринки, перевозячи такі товари, як худоба, матеріали, споживчі товари тощо. Можливості можуть бути комерційними або приватними, залежно від матеріалів, що розповсюджуються [4]–[6];

- глобальний транспорт: глобальні транспортні компанії охоплюють аспекти інших категорій, але в міжнародному масштабі. Міжнародні постачальники можуть пропонувати морську або повітряну доставку вантажів.

Для дотримання вимог кожної категорії може знадобитися унікальне знання регіональних, федеральних або навіть міжнародних законів. Виходячи з інтересів та досвіду, можна відповідно спланувати свої подальші кроки [4]–[6].

Основні кроки до відкриття вашої транспортної компанії.

Нехтування будь-яким із цих важливих елементів може призвести до стресу та проблем пізніше, тому важливо розставити пріоритети у створенні міцного фундаменту. Нижче наведено вісім кроків, які можна зробити, щоб створити свою компанію з надання транспортних послуг [4]–[6].

Крок 1. Визначитися з конкретною транспортною нішею. Коли починаєте свій транспортний бізнес, перший крок полягає в тому, кого і що будете обслуговувати. Потрібно буде відповісти на питання: яку нішу слід обрати.

Найприємніше в першому кроці – це те, що можливості практично безмежні. Можна заснувати що завгодно - від компанії з прокату велосипедів до логістичної корпорації або служби медичного транспорту. Найважливішим елементом є обрання однієї області і дізнатися про неї все, що можна [4]–[6].

Якщо нема впевненості, що вибрати – можна провести деякі дослідження щодо попиту та пропозиції у регіоні. Забезпечення вирішення конкретної та актуальної потреби чи проблеми гарантує, що буде постійна клієнтська база, коли настане час відкритись [4]–[6].

Крок 2. Вибудовування бажаної бізнес-моделі. Після того, як обрану нішу і досліджено про неї все, що можна, потрібно буде перейти до фази бізнес-моделі. Це час, коли створюється власна бізнес-структура і починається заповнення оперативної інформації про свій бізнес [4]–[6].

Крок 3. Подання заявки на отримання ліцензій та дозволів. Коли починається власна транспортна діяльність, потрібно мати відповідну ліцензію. Чому транспортним підприємствам потрібно більше дозволів, ніж іншим типам компаній – в багатьох сценаріях доведеться працювати з пасажирями, людьми та іншими видами цінних вантажів [4]–[6].

Якщо спеціалізуватися на управлінні вантажоперевезеннями або операціях, департамент транспорту має певний перелік вимог, яких слід дотримуватися. Цей список в основному базується на вазі та розмірі того, що перевозите [4]–[6].

Крок 4. Визначення бюджету та фінансові очікування. Залежно від обраного виду транспортного бізнесу витрати можуть бути мінімальними або значними. Управління службою таксі з одним транспортним засобом напевно обійдеться набагато дешевше, ніж повноцінним логістичним парком, але будь-який власник бізнесу може спланувати ці витрати [4]–[6].

Потрібно отримати кредит, щоб розпочати роботу. Адміністрація малого бізнесу-чудовий ресурс, і місцеві банки або кредитні спілки також можуть допомогти. Слід мати чіткий бізнес-план, складений на папері, оскільки багато кредиторів попросять переглянути цей план, перш ніж розглядати нову заявку на отримання кредиту [4]–[6].

Також ніколи не буває занадто рано ставити фінансові цілі та орієнтири. Потрібно розглянути:

- який дохід потрібно підтримувати, щоб покрити операційні витрати;
- скільки інвестувати у витратні матеріали, обладнання та робочу силу;

- конкретна сума заборгованості бізнесу за кредитами або іншими витратами;
- яку суму грошей ви готові вкласти в маркетинг і рекламу.

Крок 5. Робити покупки і створювати свій автопарк. Правильний вибір обладнання може забезпечити прибутковість, надавши новому бізнесу професійний вигляд з першого дня. Розмір та якість транспортного засобу мають значення, особливо коли перевозять товари та матеріали від імені інших людей. Поява на роботі із занадто маленьким або невідповідним транспортним засобом (не вистачає холодильника, місця і т.д.) може погано відбитися на вашій компанії [4]–[6].

Ось кілька речей, які слід враховувати при прийнятті рішень про покупку транспортних засобів і створенні свого автопарку:

- чи знадобиться вам перевозити будь-яких пасажирів;
- наскільки важкі матеріали, які ви будете перевозити;
- який тип місцевості Вам потрібно подолати в звичайній поїздки;
- якого зносу слід очікувати на регулярній основі;
- чи існують економічні чи екологічні варіанти на вибір.

Всі ці фактори надзвичайно важливі при створенні транспортного бізнесу.

Крок 6. Впровадження надійної практики найму персоналу. Практика найму у компанії може змінюватися в міру зростання організації, але завжди потрібно включати стандартні процедури найму у свій бізнес-план. Це допомагає формалізувати процес і гарантує, що дотримуєтесь місцевих державних правил та положень щодо працевлаштування [4]–[6].

Крок 7. Слідкування за технічним обслуговуванням та повторними сертифікаціями. Коли є власником транспортного бізнесу, потрібно починати думати про майбутнє. Незважаючи на те, що витратні матеріали, транспортні засоби та автопарк можуть бути абсолютно новими, тривале використання може вплинути як на безпеку, так і на якість. Як будуть проводитися планове

технічне обслуговування, щоб клієнти не відчували жодних перебоїв в обслуговуванні [4]–[6].

На додаток до регулярних перевірок, сертифікати та ліцензії вашої компанії можуть вимагати продовження з різною періодичністю. Щоб залишатися відповідно до законодавчих вимог та підтримувати безперебійне функціонування вашого бізнесу, вам потрібно переконатися, що вони завжди актуальні [4]–[6].

Незважаючи на те, що дрібні деталі легко втрачаються в суєті повсякденних операцій, встановлення повторюваних нагадувань або призначення конкретного працівника відповідальним за рутинні перевірки можуть запобігти багатьом проблемам [4]–[6].

Помилки, яких слід уникати при відкритті транспортного бізнесу.

Ігнорування правил сусідніх регіонів та місць: оскільки здійснюється перевезення, природно, може знадобитися перетинати державні кордони або працювати в інших юрисдикціях. Слід переконатися, що усвідомлюються правила та вимоги до ліцензування в кожному регіоні чи місці, де ведеться бізнес [4]–[6].

Забувши застрахувати кожен аспект бізнесу: хоча може здатися виснажливим покривати витрати на кожен товар, який перевозиться, водія, якого наймаєте, або пасажирів, якому допомагаєте, аварії трапляються. Страхування загальної відповідальності може заповнити прогалини в ситуаціях, коли у вас немає конкретних вимог до полісу [4]–[6].

Неможливість включити контракти та застереження: ділові контракти можуть гарантувати, що буде отримана оплата за надані послуги. Крім того, ці важливі документи задають тон тому, чого слід очікувати під час виконання роботи чи проекту. Якщо обслуговуються пасажирів, відмови можуть юридично захистити вас у разі непорозуміння або нещасного випадку [4]–[6].

У цій галузі цілком можливо почати з малого і розширювати масштаби — аж до пропозиції глобальних послуг клієнтам по всьому світу [4]–[6].

1.2 Використання застосунків для сфери логістики

Спеціалізовані програмні засоби (ПЗ) для підприємств з перевезення вантажів можуть бути дуже корисними з точки зору підвищення операційної ефективності та загальної продуктивності [7]–[9].

Ці програми розроблені спеціально для потреб компаній, що займаються вантажоперевезеннями, і можуть забезпечувати функціональність, що виходить за рамки програмного забезпечення загального призначення.

Ось кілька способів використання спеціалізованих застосунків [7]–[9].

Спосіб 1. Планування завантаження: спеціалізовані програми можуть бути використані для оптимізації планування завантаження, що є процесом визначення найкращого способу завантаження вантажу в транспортний засіб для максимальної ефективності та безпеки. ПЗ для планування завантаження може враховувати багато факторів, таких як розмір і вага вантажу, кількість зупинок на маршруті та тип використовуваного транспортного засобу для створення оптимального плану [7]–[9].

Спосіб 2. Планування маршруту: спеціалізовані програми також можуть бути використані для оптимізації планування маршруту, що включає визначення найбільш ефективного маршруту доставки. ПЗ для планування маршруту може враховувати такі фактори, як трафік, дорожні умови та розташування пунктів прийому та доставки, щоб сформувати оптимальний маршрут [7]–[9].

Спосіб 3. Відстеження та моніторинг: спеціалізовані програми можуть використовуватися для відстеження відправлень у режимі реального часу. Це може включати GPS-відстеження транспортних засобів та вантажів, а також сповіщення про будь-які несподівані події, такі як нещасні випадки або затримки. Програми відстеження та моніторингу можуть надати компаніям, що займаються вантажоперевезеннями, інформацію, необхідну для швидкого прийняття рішень та інформування клієнтів.

Спосіб 4. Управління складом: спеціалізовані програми можуть використовуватися для управління складськими операціями, включаючи управління запасами, обробку замовлень та відстеження товарів. Ці програми можуть надати компаніям, що займаються вантажоперевезеннями, можливість керувати рівнем запасів і відстежувати переміщення товарів на складах, та за їх межами [7]–[9].

Спосіб 5. Управління перевезеннями: спеціалізовані програми можуть використовуватися для управління процесом Експедирування вантажів, який включає координацію переміщення вантажу з одного місця в інше. Програмне забезпечення для управління перевезеннями може надати компаніям, що займаються вантажоперевезеннями, можливість керувати кількома відправленнями одночасно та відстежувати стан кожного відправлення [7]–[9].

Спосіб 6. Безпека та відповідність вимогам: спеціалізовані програми також можуть використовуватися для забезпечення дотримання правил техніки безпеки та інших вимог. Наприклад, деякі програми можуть відстежувати години роботи водія та переконатися, що вони не перевищують законодавчих обмежень, тоді як інші можуть відстежувати технічне обслуговування транспортних засобів та забезпечувати належне обслуговування транспортних засобів [7]–[9].

В цілому, спеціалізовані програми можуть надати підприємствам з перевезення вантажів інструменти, необхідні їм для оптимізації своєї діяльності, підвищення ефективності та забезпечення відповідності нормативним вимогам. Використовуючи спеціалізовані програми, підприємства з перевезення вантажів можуть отримати конкурентну перевагу, надаючи своїм клієнтам більш якісний сервіс і знижуючи витрати [7]–[9].

1.3 Порівняння існуючих аналогів

Спеціалізовані застосунки для підприємств з перевезення вантажів розроблені з урахуванням специфічних потреб компаній, що працюють у сфері

вантажоперевезень і логістики. Основні ідеї, що лежать в основі цих додатків, включають:

Підвищення операційної ефективності: спеціалізовані програми можуть допомогти компаніям, що займаються вантажоперевезеннями, оптимізувати свої операції, автоматизуючи ручні процеси та скорочуючи час, необхідний для виконання таких завдань, як планування завантаження, маршрут та управління запасами.

Покращення досвіду клієнтів: спеціалізовані програми можуть допомогти компаніям, що займаються вантажоперевезеннями, забезпечити краще обслуговування клієнтів, покращивши спілкування та прозорість. Відстеження та моніторинг відправлень у режимі реального часу можуть надавати клієнтам актуальну інформацію про стан їх вантажу, що може сприяти зміцненню довіри та лояльності.

Зниження витрат: спеціалізовані програми можуть допомогти вантажоперевезенням зменшити витрати, оптимізуючи завантаження та планування маршрутів, покращуючи управління запасами та мінімізуючи ризик аварій та затримок.

Забезпечення відповідності: спеціалізовані програми можуть допомогти підприємствам з перевезення вантажів забезпечити дотримання правил техніки безпеки та інших вимог. Наприклад, деякі програми можуть відстежувати години роботи водія та переконатися, що вони не перевищують законодавчих обмежень, тоді як інші можуть відстежувати технічне обслуговування транспортних засобів, щоб забезпечити належний догляд за транспортними засобами.

Підвищення прибутковості: підвищуючи операційну ефективність, покращуючи обслуговування клієнтів, знижуючи витрати та забезпечуючи відповідність, спеціалізовані програми можуть допомогти вантажоперевезенням підвищити прибутковість.

В цілому, основна ідея спеціалізованих застосунків для компаній, що займаються вантажоперевезеннями, полягає в наданні комплексного рішення,

яке вирішує унікальні завдання, з якими стикаються компанії, що працюють в сфері вантажоперевезень і логістики. Ці програми призначені для оптимізації операцій, поліпшення обслуговування клієнтів, зниження витрат і забезпечення відповідності вимогам - і все це з метою підвищення прибутковості.

1.3.1 Сервіс «Uber Freight»

Uber Freight (рис. 1.1) – підрозділ Uber Technologies, що пропонує цифрову платформу, що об'єднує вантажовідправників і перевізників для вантажних перевезень [10], [11]. Ось коротка історія та пояснення того, як це працює:

Uber Freight був запущений в 2017 році з метою оцифрування та спрощення галузі вантажних перевезень. Сервіс побудований на тій же технології, що і платформа для організації поїздок Uber, використовуючи алгоритми і машинне навчання для підбору вантажовідправників і перевізників і забезпечуючи прозорість і ефективність процесу [10], [11].

Вантажовідправники: вантажовідправники можуть використовувати платформу Uber Freight для бронювання вантажу. Вони вказують місця отримання та доставки, тип вантажу та будь-яку іншу відповідну інформацію. Потім платформа генерує список доступних перевізників, які мають кваліфікацію для перевезення вантажу [10], [11].

Перевізники: перевізники можуть використовувати платформу для пошуку та бронювання доступних вантажів. Вони можуть здійснювати пошук вантажів в залежності від їх місця розташування і доступності, а також переглядати детальну інформацію про кожен вантаж, включаючи місця отримання і доставки, тип вантажу і умови оплати.

Відповідність: вантажна платформа Uber використовує машинне навчання та алгоритми для відповідності вантажовідправників та перевізників на основі таких факторів, як доступність, розташування та тип вантажу. Після встановлення відповідності платформа обробляє всю логістику, включаючи

координацію прийому та доставки, обробку платежів та відстеження відправлення в режимі реального часу [10], [11].

Прозорість: платформа забезпечує прозорість протягом усього процесу, дозволяючи як вантажовідправникам, так і перевізникам відстежувати відправлення в режимі реального часу, переглядати платіжні реквізити та отримувати повідомлення про будь-які проблеми або затримки [10], [11].

Оплата: Uber Freight обробляє всю обробку платежів, включаючи виставлення рахунків-фактур. Оплата перевізникам здійснюється безпосередньо через платформу, причому платежі зазвичай обробляються протягом семи днів з моменту доставки [10], [11].

Uber Freight пропонує ряд переваг вантажовідправникам і перевізникам, включаючи більшу ефективність, підвищену прозорість і доступ до більш широкої мережі перевізників і вантажовідправників. Платформа також пропонує такі функції, як цілодобова підтримка 7 днів на тиждень, відстеження в режимі реального часу та миттєві ціни. Зосереджуючись на цифровізації та спрощенні галузі вантажних перевезень, Uber Freight швидко став головним гравцем у секторі логістики та перевезень [10], [11].

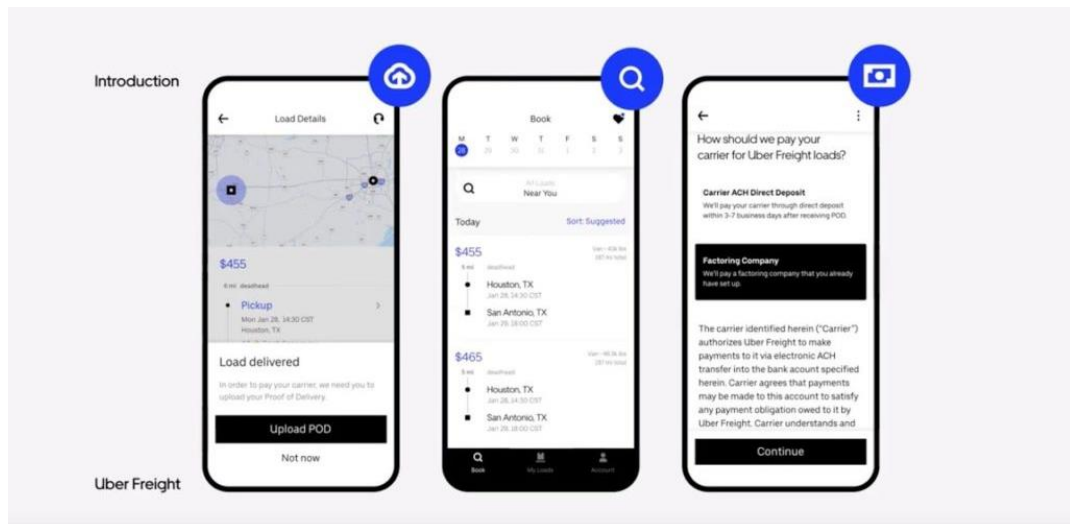


Рисунок 1.1 – Приклади інтерфейсу платформи Uber Freight [10]

1.3.2 Сервіс «Webfleet»

Webfleet (рис. 1.2) – це система управління автопарком, яка допомагає компаніям керувати своїм автопарком. Ось коротка історія та пояснення того, як це працює [12], [13].

Компанія Webfleet була заснована в 1991 році в Нідерландах як постачальник навігаційних систем. З часом компанія розширила свої послуги, включивши програмне забезпечення для управління автопарком, і в 2006 році її придбав TomTom, голландський виробник навігаційних систем. Сьогодні Webfleet є провідним постачальником рішень для управління автопарком, що обслуговує підприємства всіх розмірів у таких галузях, як транспорт, логістика та будівництво [12], [13].

GPS-відстеження: Webfleet використовує GPS-відстеження для відстеження місцезнаходження та переміщення транспортних засобів у автопарку. Ці дані передаються в режимі реального часу в програмне забезпечення Webfleet, яке надає підприємствам інформацію про роботу їх автопарку в режимі реального часу [12], [13].

Моніторинг транспортних засобів: Webfleet надає підприємствам детальну інформацію про їхні транспортні засоби, включаючи споживання палива, продуктивність двигуна та графіки технічного обслуговування. Ця інформація може бути використана для визначення областей, які потребують вдосконалення, та оптимізації роботи автопарку [12], [13].

Поведінка водія: Webfleet також відстежує поведінку водія, надаючи інформацію про такі фактори, як швидкість, прискорення та гальмування. Ця інформація може бути використана для підвищення безпеки водія і зниження витрати палива [12], [13].

Маршрутизація та диспетчеризація: Webfleet надає підприємствам інструменти для оптимізації маршрутизації та диспетчеризації. Програмне забезпечення можна використовувати для планування найефективніших

маршрутів руху транспортних засобів, враховуючи такі фактори, як дорожній рух, погода та дорожні умови [12], [13].

Аналітика та звітність: Webfleet надає підприємствам інструменти аналітики та звітності, які дозволяють їм аналізувати ефективність автопарку та визначати сфери для вдосконалення. Програмне забезпечення може генерувати детальні звіти про такі фактори, як витрата палива, поведінка водія та використання транспортного засобу [12], [13].

Інтеграція: Webfleet може бути інтегрований з іншими програмними системами, що дозволяє підприємствам оптимізувати свою діяльність і підвищити ефективність. Наприклад, програмне забезпечення може бути інтегроване з системами бухгалтерського обліку, що дозволяє без проблем виставляти рахунки-фактури [12], [13].

Webfleet пропонує підприємствам ряд переваг, включаючи підвищення ефективності, зниження витрати палива та підвищення безпеки. Програмне забезпечення надає підприємствам інформацію про роботу їх автопарку в режимі реального часу, дозволяючи їм приймати обґрунтовані рішення і оптимізувати свою діяльність. Завдяки широкому набору функцій і перевіреному досвіду успішної роботи Webfleet стала провідним постачальником рішень для управління автопарком [12], [13].



Рисунок 1.2 – Приклад інтерфейсу сервісу «Webfleet» для різних платформ [12]

1.3.3 Сервіс «Нова пошта»

«Нова пошта» (рис. 1.3) – українська кур'єрська та логістична компанія, що надає широкий спектр послуг з доставки вантажів. Ось коротка історія та пояснення того, як це працює [14], [15].

«Нова пошта» була заснована в 2001 році в Україні як місцева кур'єрська служба. З часом компанія розширила свої послуги, включивши міжнародні перевезення, складування та логістику. Сьогодні Нова пошта – одна з найбільших кур'єрських і логістичних компаній в Україні, що має понад 4000 відділень і автопарк з більш ніж 12 000 транспортних засобів [14], [15].

Кур'єрські послуги: Нова пошта надає кур'єрські послуги як приватним особам, так і підприємствам. Клієнти можуть планувати прийом і доставку онлайн або через Мобільний додаток компанії, а також відстежувати свої відправлення в режимі реального часу [14], [15].

Доставка: Нова Пошта пропонує широкий спектр послуг з доставки, включаючи експрес-доставку, стандартну доставку та міжнародну доставку. Компанія має мережу з більш ніж 4000 відділень по всій Україні, що спрощує клієнтам відправку та отримання посилок [14], [15].

Складування та логістика: Нова Пошта надає складські та логістичні послуги для бізнесу, включаючи зберігання, управління запасами та виконання замовлень. Компанія має в своєму розпорядженні мережу складів і розподільних центрів по всій Україні, що дозволяє їй надавати ефективні та економічні логістичні рішення [14], [15].

Технологія: Нова Пошта використовує передові технології для поліпшення своєї діяльності та надання кращого сервісу клієнтам. Компанія розробила ряд програмних засобів і мобільних додатків, які дозволяють клієнтам відстежувати свої відправлення, планувати прийом і доставку, а також керувати своїми обліковими записами онлайн [14], [15].

Стійкість: Нова Пошта прагне до сталого розвитку і реалізувала ряд ініціатив щодо зниження впливу на навколишнє середовище. Компанія

інвестувала в електричні та гібридні транспортні засоби, впровадила енергоефективні системи освітлення та опалення на своїх складах і запустила програму утилізації пакувальних матеріалів [14], [15].

Нова Пошта пропонує клієнтам ряд переваг, включаючи швидку і надійну доставку, гнучкі варіанти доставки і широкий спектр послуг для задоволення їх потреб. Завдяки своїй великій мережі відділень і передовим технологіям Нова пошта стала лідером у сфері кур'єрської доставки та логістики в Україні [14], [15].

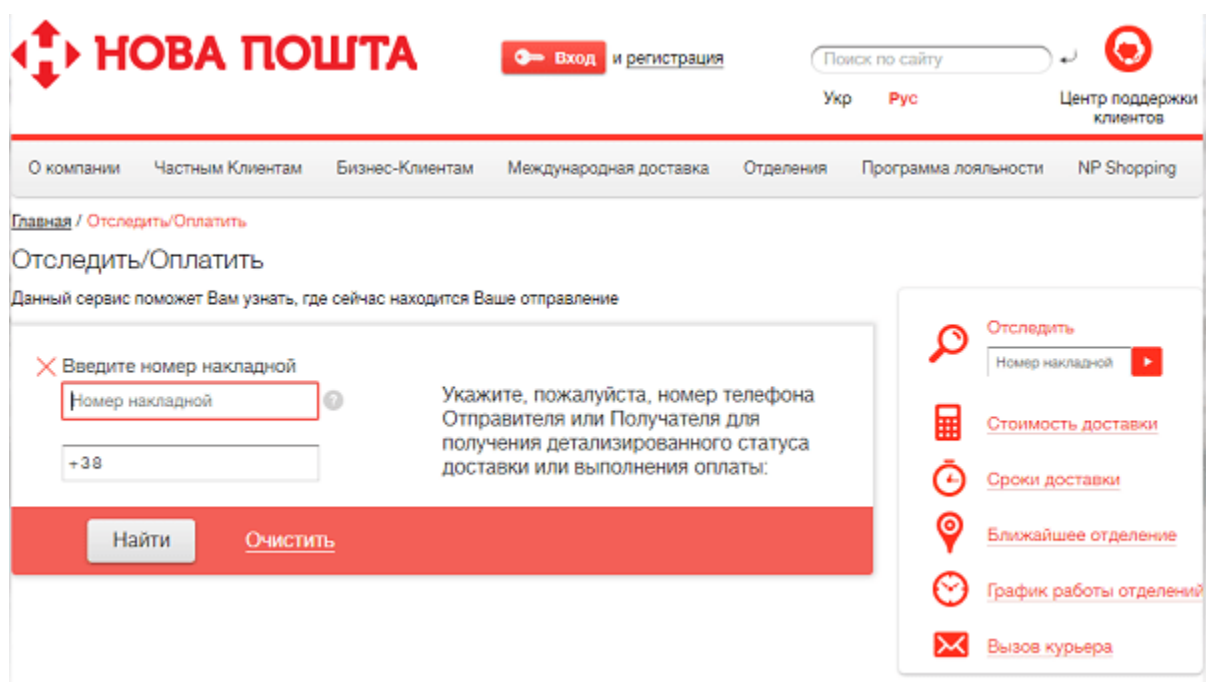


Рисунок 1.3 – Приклад інтерфейсу «Нова пошта» [14]

1.3.4 Порівняння існуючих аналогів

Виконаємо порівняння існуючих аналогів (табл. 1.1) за наступними критеріями:

- перелік доступних путівок – чи надає ресурс перелік доступних для бронювання путівок (коли є можливість забронювати не тільки житло, а ще й додаткові послуги);

- перелік доступних авіарейсів – чи надає ресурс перелік доступних авіарейсів, щоб дістатися до пункту призначення;
- бронювання на основі попередньої оплати – чи можливо виконати бронювання за попередньою оплатою (частковою або повною);
- адаптивність та швидкість роботи сайту – оцінка якості роботи ресурсу (наскільки швидко відкриваються та працюють сторінки ресурсу).

При порівнянні можна побачити, що більшість ресурсів не надають інформацію про доступні путівки – виключно в режимі бронювання місця в готелі. Проте часто важливими для клієнтів є саме додаткові послуги та комплекси послуг, що можуть надаватися під час подорожі. Також не усі ресурси надають можливість забронювати місце на авіарейс, що може бути не зручним та в подальшому буде накладати додаткову комісію на користувача. Нарешті, деякі існуючі платформи відрізняються дуже повільною швидкістю роботи, що значно уповільнює роботу з ними.

Тому актуальною задачею залишається розробка інформаційного ресурсу для туристичного агентства у вигляді вебзастосунку із реалізацією усього функціоналу та зручним інтерфейсом.

Таблиця 1.1 – Програмне забезпечення для підтримки освітньої діяльності

Критерії порівняння	Uber Freight	Webfleet	Нова Пошта	TransLogistic
Робота в Україні	Ні	Ні	Так	Так
Прямий зв'язок замовника із водієм	Так	Ні	Ні	Так
Трекінг в реальному часі	Так	Та	Так	Так
Калькулятор вартості доставки	Ні	Ні	Так	Так
Рейтингування перевізників	Так	Ні	Ні	Так

1.4 Постановка завдання

Головне завдання: реалізувати транспортний вебсервіс призначений для розміщення замовлень, та ефективної комунікації між замовниками та водіями. Експлуатаційне призначення системи це використання її власниками вантажу та власниками вантажних автівок [16], [17].

1.4.1 Функціональність системи

Функціональні вимоги водія:

- реєстрація в системі;
- авторизація як водій;
- додавання транспорту;
- видалення транспорту;
- зв'язок водія з замовником;
- можливість створення рейтингу водію;
- можливість брати замовлення водію;
- створення калькулятора для розрахунку вартості роботи водія;
- додавання мапи для перегляду кілометражу між точками

перевезення.

Функціональні вимоги замовника:

- реєстрація в системі;
- авторизація як замовник;
- додавання замовлення;
- видалення замовлення;
- зв'язок замовника з водієм;
- можливість створення рейтингу замовника;
- можливість вибору транспорту замовникові.

Функціональні вимоги співробітників платформи:

- технічна підтримка;
- видалення некоректних заявок замовників та водіїв.

1.4.2 Вимоги до інтерфейсу

Інтерфейс повинен бути зручний, як для користувача так і для адміністратора:

- мінімальні затрати ресурсів користувача при вводі, модифікації та перегляді даних;
- максимальна взаємодія користувача з системою.

1.4.3 Вимоги до продуктивності

Система повинна працювати цілодобово без збоїв та помилок. У разі виникнення збою системи, повинна адекватно на них реагувати, виводити відповідні повідомлення користувачам та зберігати усю вже наявну інформацію про клієнтів та путівки і відновлюватися у найкоротші терміни.

У випадку некоректних дій користувача система повина адекватно реагувати та надавати підказки користувачу.

1.4.4 Вимоги до безпеки

Система повина бути захищеною від посягань зловмисників на персональні дані користувачів. Кожен акаунт повинен бути прив'язаний до електронних скриньок користувачів системи.

Для використання можливостей системи користувач повинен мати один з відомих браузерів останніх версій.

1.4.5 Експлуатаційні вимоги

Експлуатаційними вимогами є:

- Інтернет;
- сервер;
- персональний комп'ютер (ПК);
- актуальний браузер.

1.4.6 Не функціональні вимоги до продукту

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;
- розгорнута серверна операційна система (ОС);
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі засоби:

- ПК з IBM-сумісною архітектурою;
- процесор із тактовою частотою від 2,0 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet;
- встановлений браузер.

1.5 Висновки за розділом 1

Цей розділ було присвячено дослідженню проблемної області надання послуг з перевезення вантажу. Окрему увагу було приділено аналізу існуючих інформаційних ресурсів таких компаній, що розміщені в Інтернеті. Дослідивши їх переваги та недоліки було сформовано ряд вимог до нового застосунку, що буде розроблятися. Таким чином новий інформаційний ресурс матиме усі шанси стати конкурентоспроможним та якісним.

2 МАТЕРІАЛИ І МЕТОДИ

Вище було досліджено проблемну області надання послуг з перевезення вантажу. Відповідно, окрему увагу було приділено аналізу існуючих інформаційних ресурсів таких компаній, що розміщені в Інтернеті. Дослідивши їх та визначивши головні переваги та недоліки було сформовано ряд вимог до нового застосунку, що буде розроблятися. Таким чином новий інформаційний ресурс матиме усі шанси стати конкурентоспроможним та якісним.

В цьому розділі буде досліджуватися вибір технології матеріалів розробки: вибір засобів та технологій розробки. Так буде обрано мову програмування, програмну надбудову для спрощення процесу розробки та шаблон проєктування. Вирішення цих завдань на основі аналізу та дослідження допоможе розв'язати поставлене завдання другого розділу.

2.1 Вибір мови програмування

PHP, що розшифровується як гіпертекстовий препроцесор, є серверною мовою сценаріїв, яка широко використовується для веброзробки. Незважаючи на те, що PHP існує вже більше 25 років, він все ще залишається популярним вибором для проєктів веброзробки, особливо для створення динамічних та інтерактивних вебзастосунків [16]–[20].

Ось кілька причин, чому PHP все ще залишається актуальною та важливою ідеєю для веброзробки сьогодні.

Простий у використанні: PHP легко вивчити та використовувати навіть для розробників з невеликим досвідом програмування. Мова базується на C та Perl, що означає, що розробники можуть швидко зрозуміти його синтаксис та структуру. Крім того, в PHP існує велика і активна спільнота розробників, які створили безліч ресурсів, фреймворків і бібліотек, які допомагають розробникам швидко приступити до роботи [16]–[20].

Економічність: PHP – це мова з відкритим кодом, що означає, що вона вільна у використанні та розповсюдженні. PHP не має ліцензійних зборів, що робить його економічно вигідним вибором для бізнесу та розробників. Крім того, оскільки PHP використовується так широко, існує безліч хостинг-провайдерів, які пропонують доступні та надійні послуги хостингу для програм на базі PHP [16]–[20].

Сумісність між платформами: PHP – це крос-платформна мова, що означає, що її можна використовувати в різних операційних системах, таких як Windows, macOS та Linux. Це спрощує розробникам створення додатків, які можуть бути розгорнуті на широкому спектрі серверів і платформ [16]–[20].

Гнучкість: PHP – це дуже гнучка мова, яку можна використовувати для створення широкого спектру вебзастосунків, від невеликих вебсайтів до масштабних корпоративних систем. Він має безліч вбудованих функцій і комплектуючих, які дозволяють легко створювати складні і динамічні додатки. Крім того, існує безліч сторонніх фреймворків та бібліотек, які можна використовувати для розширення функціональності PHP [16]–[20].

Підтримка спільноти: PHP має велику та активну спільноту розробників, які сприяють її розробці, підтримці та підтримці. Ця спільнота створила багато ресурсів, таких як документація, форуми та навчальні посібники, які можуть допомогти розробникам вивчати та використовувати PHP. Крім того, спільнота розробила безліч сторонніх інструментів, бібліотек та фреймворків, які можна використовувати для розширення функціональності PHP та підвищення його продуктивності [16]–[20].

Інтеграція з іншими технологіями: PHP можна легко інтегрувати з іншими технологіями, такими як HTML, CSS, JavaScript та SQL. Це дозволяє створювати додатки, які використовують широкий спектр технологій, інструментів та фреймворків. Крім того, PHP може бути інтегрований з популярними системами управління вмістом, такими як WordPress, Joomla та Drupal, що полегшує створення та управління вебсайтами та програмами.

На закінчення відзначимо, що PHP як і раніше залишається актуальною і важливою ідеєю для веброзробки і сьогодні. Він простий в освоєнні і використанні, економічний, кроссплатформенно сумісний, гнучкий і має велике і активне співтовариство розробників. Крім того, він може бути легко інтегрований з іншими технологіями та фреймворками, що дозволяє створювати широкий спектр вебзастосунків. Незважаючи на появу нових технологій та фреймворків веброзробки, PHP залишається популярним вибором для створення динамічних та інтерактивних вебзастосунків [16]–[20].

Проведемо порівняння мов програмування для реалізації вебзастосунків, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні (табл. 2.1):

- об’єктно-орієнтовна концепція – підтримка мовою програмування концепції об’єктно-орієнтовного програмування;
- робота з системами керування базами даних (СКБД) – наявність механізмів для роботи із базами даних та СКБД;
- простота вивчення – простота вивчення мови програмування та вхід в розробку;
- гнучкість та масштабованість – гнучкість мови програмування для розробки застосунку та можливості подальших підтримки та масштабованості.

Таблиця 2.1 – Порівняння мов програмування

Критерій	PHP	JavaScript
Об’єктно-орієнтовна концепція	Повністю підтримує	Відносна підтримка
Робота з СКБД	Повністю реалізована	Вимагає додаткових надбудов
Простота вивчення	Проста	Проста
Гнучкість та масштабованість	Висока	Висока гнучкість, але відносна масштабованість

З таблиці видно, що у якості альтернативи PHP є JavaScript, проте аналог потребує додаткових надбудов для роботи із базами даних. Більш того, об’єктно-орієнтовне програмування в JavaScript має не класичне

представлення, тому іноді це призводить до множини помилок. Саме тому PHP для звичайного інтерактивного вебзастосунку є хорошим вибором.

2.2 Вибір та обґрунтування системи керування вмістом

WordPress – це популярна і широко використовувана система управління контентом (CMS), яка використовується для створення різних вебсайтів, від простих блогів до складних сайтів електронної комерції. WordPress – це ПЗ з відкритим кодом, написане на PHP і використовує базу даних MySQL. Незважаючи на те, що WordPress існує вже більше 18 років, він все ще є актуальною та важливою ідеєю для веброзробки і сьогодні.

Ось кілька причин, чому WordPress залишається популярним вибором для веброзробки [21]–[23].

Простота використання: WordPress розроблений таким чином, щоб бути простим у використанні навіть для людей, які практично не мають досвіду веброзробки. Платформа зручна у використанні та інтуїтивно зрозуміла, з простим інтерфейсом, який дозволяє користувачам створювати та керувати вмістом без необхідності знати, як кодувати. Крім того, WordPress має велику та активну спільноту користувачів, які створили безліч ресурсів, таких як навчальні посібники та форуми, які можуть допомогти новачкам розпочати роботу з платформою [21]–[23].

Гнучкість: WordPress має високу гнучкість і налаштованість, що дозволяє створювати широкий спектр вебсайтів і додатків. Платформа пропонує тисячі безкоштовних та преміальних тем та плагінів, які можна використовувати для додавання функціональності та покращення зовнішнього вигляду вебсайту. Крім того, WordPress можна легко інтегрувати з іншими технологіями та платформами, що дозволяє створювати складні та динамічні програми [21]–[23].

Оптимізований для пошукової оптимізації (SEO): WordPress розроблений таким чином, щоб бути зручним для пошукових систем, а це

означає, що він оптимізований для рейтингу в пошукових системах. Платформа включає в себе безліч вбудованих функцій SEO, таких як спеціальні постійні посилання, метадані та XML-карти сайту, які можуть допомогти покращити видимість вебсайту на сторінках результатів пошуку. Крім того, існує безліч сторонніх SEO-плагінів, які можна використовувати для подальшої оптимізації вебсайту для пошукових систем [21]–[23].

Безпека: WordPress серйозно ставиться до безпеки та має безліч вбудованих функцій безпеки, які можуть допомогти захистити вебсайти від атак. Крім того, існує безліч сторонніх плагінів безпеки, які можна використовувати для подальшого підвищення безпеки вебсайту. Платформа також має велику та активну спільноту розробників, які регулярно випускають оновлення та виправлення для усунення вразливостей безпеки [21]–[23].

Мобільна чуйність: зі збільшенням використання мобільних пристроїв для доступу до Інтернету важливо, щоб вебсайти були адаптовані до мобільних пристроїв. WordPress розроблений таким чином, щоб бути адаптивним до мобільних пристроїв, що означає, що вебсайти, створені на платформі, автоматично коригують свій макет та дизайн відповідно до розміру екрана пристрою, який використовується для їх перегляду. Крім того, багато тем і плагінів спеціально розроблені для роботи з мобільними пристроями [21]–[23].

Підтримка спільноти: WordPress має велику та активну спільноту користувачів та розробників, які сприяють його розробці, обслуговуванню та підтримці. Ця спільнота створила безліч ресурсів, таких як документація, форуми та навчальні посібники, які можуть допомогти користувачам ефективно вивчати та використовувати WordPress. Крім того, існує безліч сторонніх плагінів та тем, створених членами спільноти, які можуть допомогти розширити функціональність WordPress [21]–[23].

Можливості електронної комерції: WordPress можна використовувати для створення вебсайтів електронної комерції, при цьому доступно багато сторонніх плагінів для додавання функціональності електронної комерції. Одним з найпопулярніших плагінів електронної комерції для WordPress є

WooCommerce, який дозволяє користувачам створювати власний інтернет-магазин з такими функціями, як списки товарів, кошики для покупок та обробка платежів [21]–[23].

На закінчення, WordPress залишається популярною і важливою ідеєю для веброзробки сьогодні завдяки своїй простоті використання, гнучкості, оптимізованості для SEO, безпеки, чуйності мобільних пристроїв, підтримці спільноти і можливостям електронної комерції. WordPress - це ідеальна платформа для створення широкого спектру вебсайтів і додатків, від простих блогів до складних сайтів електронної комерції. Незважаючи на появу нових технологій і платформ веброзробки, WordPress продовжує розвиватися і вдосконалюватися, завдяки регулярним оновленням і додаванню нових функцій спільнотою розробників [21]–[23].

Порівняльна характеристика WordPress з іншими найпоширенішими CMS для розробки вебзастосунку у таблиці (табл. 2.2).

Серед критеріїв порівняння будемо розглядати наступні:

- вимоги до хостингу – вимоги зі сторони застосунку та CMS до хостингу, де розгортається застосунок;
- впорядкованість застосунку – наявність чіткої архітектури для спрощення підтримки;
- можливість масштабованості – можливість та простота масштабованості попередньо розробленого застосунку, на основі використання сторонніх модулів;
- робота з СКБД – простота та захищеність роботи із СКБД;
- підтримка та сторонні розширення – підтримка зі сторони спільноти та оцінка кількості якісних сторонніх розширень для спрощення розробки додаткового функціоналу.

Таблиця 2.2 – Порівняння фреймворків

	WordPress	Joomla
Вимоги до хостингу	Середні	Середні
Впорядкованість застосунку	Чітка архітектура	Ускладнена архітектура
Можливість масштабованості	Можливо	Можливо, але ускладнено
Робота з СКБД	Проста, захищена	Проста, потребує додаткового захисту
Підтримка та сторонні розширення	Активна спільнота, багато розширень	Менш активна спільнота, не багато розширень

З порівняння можна зробити висновок, що WordPress – продуктивна CMS, яка залишається однією з найпопулярніших цілком заслужено. Вона надає широкий функціонал для розробників та значно спрощує цей процес.

2.3 Структурна схема розробки

Загальна схема системи за обраною темою на рисунку (рис. 2.1).

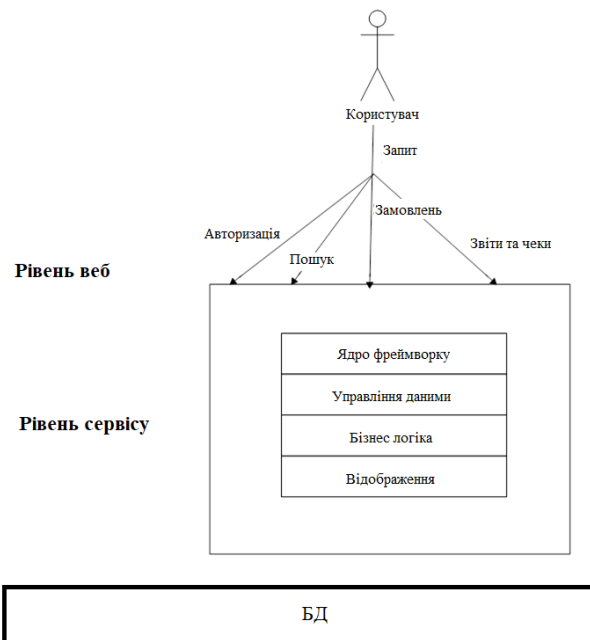


Рисунок 2.1 – Загальна схема системи за обраною темою

Архітектура вебзастосунку базується на моделі Модель-вид-контролер (MVC) (рис. 2.2).

Модель здійснює роботу з базою даних та саме у моделях повинні бути зосереджені методи роботи з даними.

Контролер оброблює дані з моделі, якщо вони потрібні, і генерує Вид з цими даними, де інформація доноситься до користувача з відповідними графічними елементами. Взаємодія користувача з додатком відбувається через Контролер.

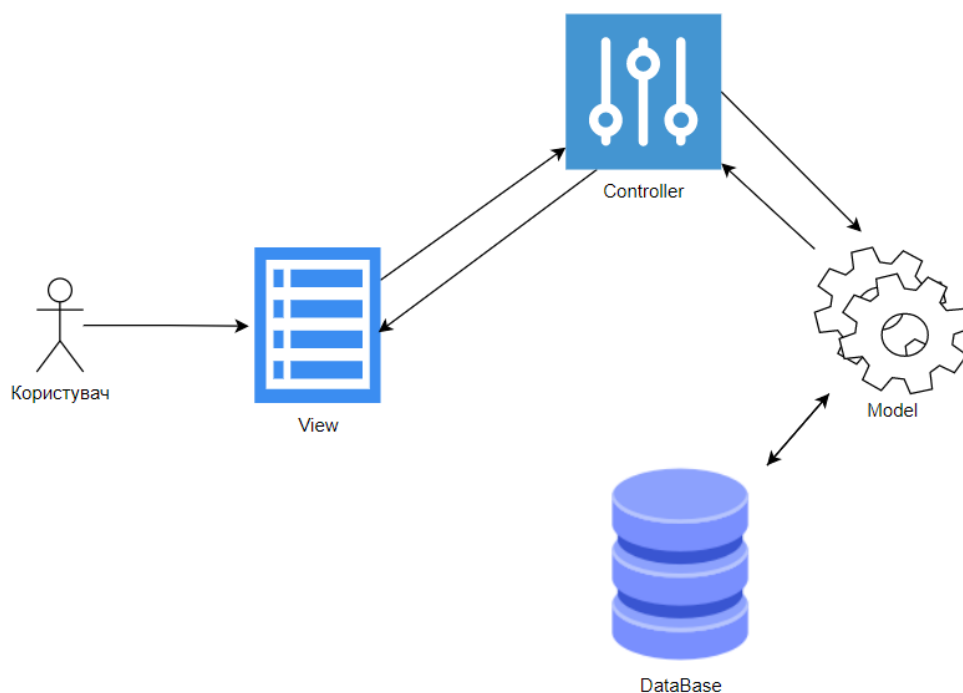


Рисунок 2.2 – Архітектура вебзастосунку

2.4 Висновки за розділом 2

В цьому розділі було досліджено питання вибору технології матеріалів розробки: обрано засоби та технології розробки. Так було обрано мову програмування, програмну надбудову для спрощення процесу розробки та шаблон проєктування. Вирішення цих завдань на основі аналізу та дослідження допоможе розв'язати поставлене завдання другого розділу.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проектування моделей системи

Unified Modeling Language (UML) – це мова візуального моделювання, яка використовується для розробки програмного забезпечення [24], [25]. Вона пропонує стандартизовану нотацію, яка допомагає розробникам створювати моделі, зрозумілі як технічним, так і нетехнічним зацікавленим сторонам. UML – це потужний інструмент, який можна використовувати для проектування та документування програмних систем, включаючи вебпрограми.

Що стосується проектування вебзастосунків, UML-діаграми використовуються для моделювання поведінки та структури системи. Нижче наведено деякі діаграми UML, які можна використовувати для розробки вебзастосунків [24], [25].

Діаграма класів (рис. 3.1): ця діаграма представляє статичну структуру системи та взаємозв'язки між різними об'єктами. Діаграма класів забезпечує візуальне представлення класів, атрибутів та методів системи [24], [25].

Діаграма прецедентів (рис. 3.2): ця діаграма використовується для визначення функціональних вимог до системи. У ній зображені різні дійові особи і їх взаємодія з системою. Діаграми випадків використання застосовуються для визначення та опису основних завдань, які користувачі виконують всередині системи [24], [25].

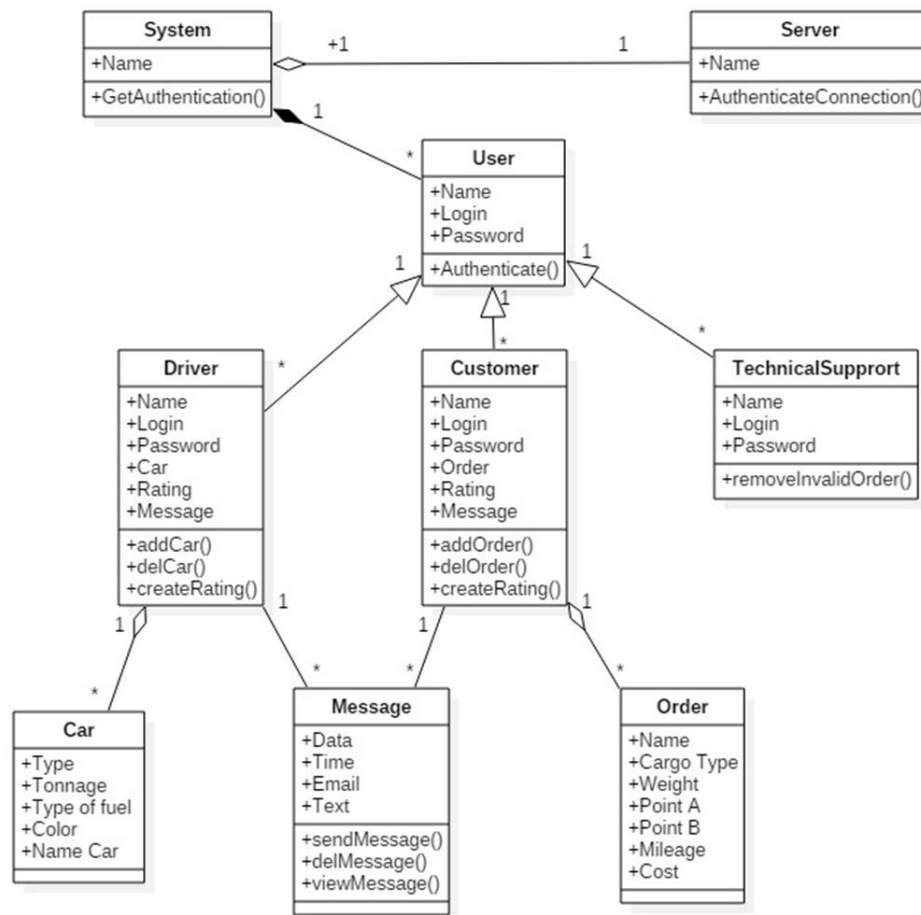


Рисунок 3.1 – Діаграма класів

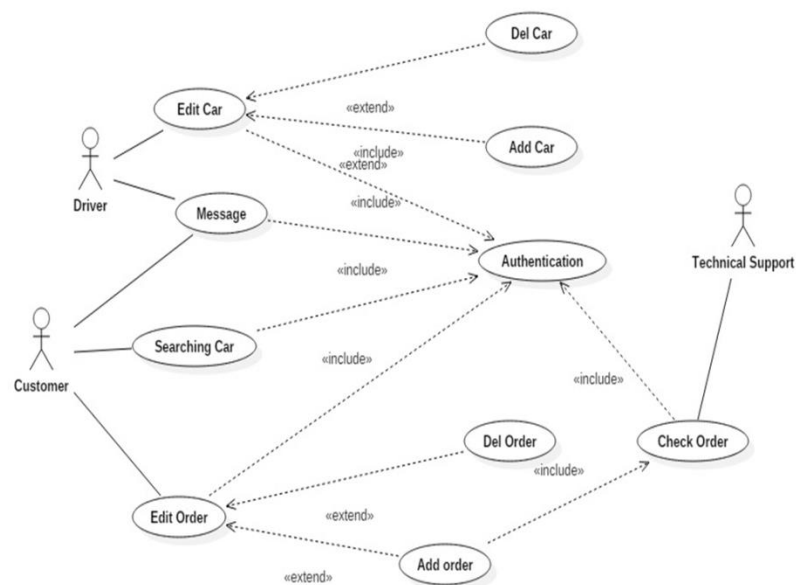


Рисунок 3.2 – Діаграма прецедентів

Діаграма послідовності (рис. 3.3): ця діаграма зображує взаємодію між різними об'єктами та послідовність подій у системі. Він використовується для моделювання динамічної поведінки системи [24], [25].

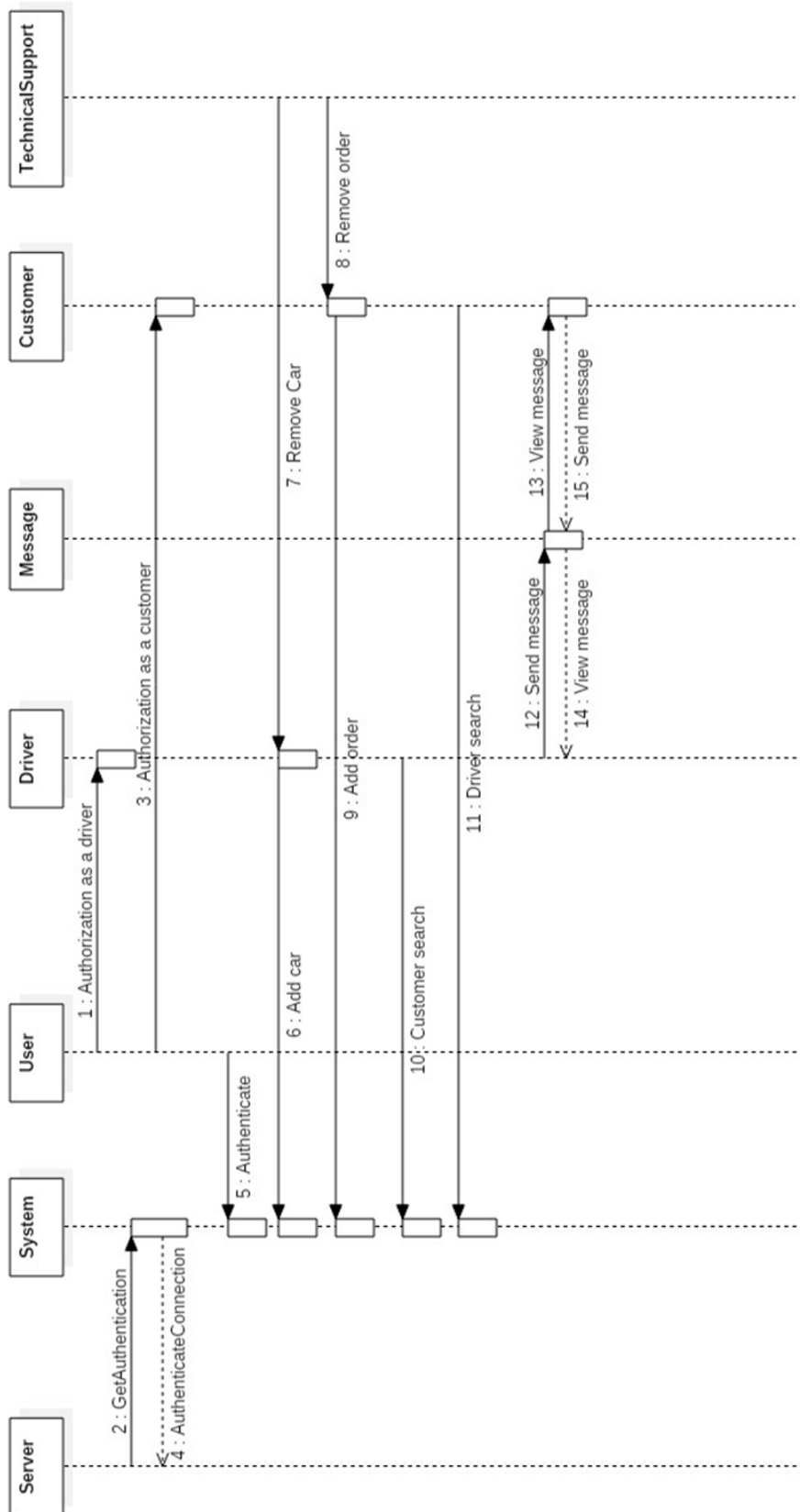


Рисунок 3.3 – Діаграма послідовностей

Діаграма дій (рис. 3.4): ця діаграма використовується для моделювання потоку дій у системі. На ньому зображені різні дії та послідовність, в якій вони відбуваються [24], [25].

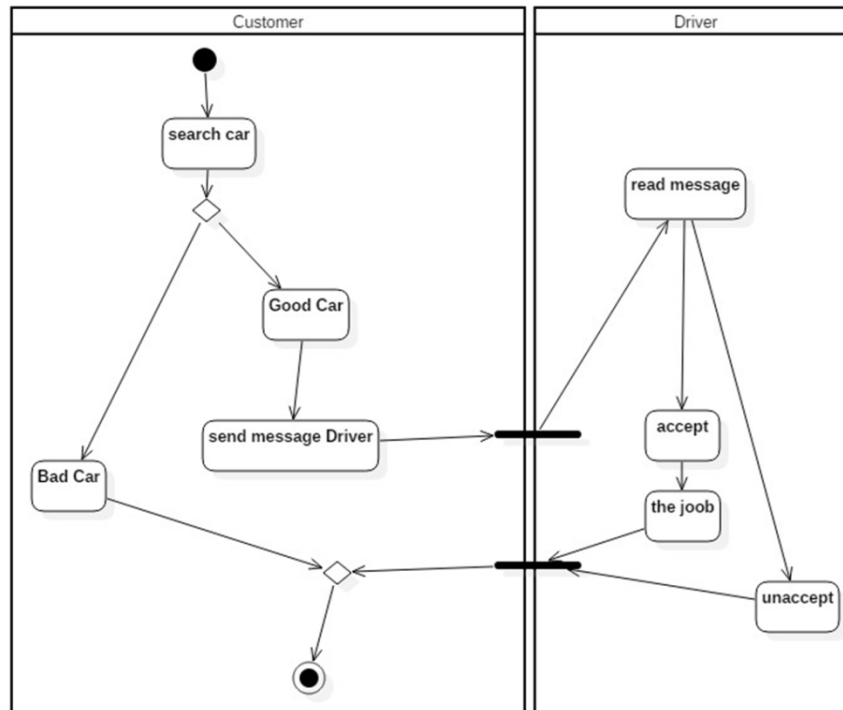


Рисунок 3.4 – Діаграма діяльності

Схема компонентів: на цій схемі представлені фізичні та логічні компоненти системи. Це показує, як пов'язані різні компоненти системи та як вони взаємодіють між собою [24], [25].

Схема розгортання: ця схема представляє фізичне розгортання системи. На ній зображені різні апаратні та програмні компоненти системи та те, як вони підключені [24], [25].

Використовуючи UML для проектування вебзастосунків, важливо почати зі схеми випадків використання. Ця діаграма допоможе визначити основних учасників системи та основні завдання, які вони виконують. Діаграма КЛАСІВ використовується для визначення моделі даних системи. У ньому зображені різні класи і їх взаємини. Діаграма послідовності використовується

для моделювання різних взаємодій між об'єктами в системі. Це допомагає визначити основні випадки використання та їх послідовність [24], [25].

Діаграма дій використовується для моделювання потоку дій у системі. Це забезпечує візуальне представлення різних видів діяльності та того, як вони пов'язані. Схема компонентів використовується для моделювання різних компонентів системи. Це показує, як з'єднані різні компоненти та як вони взаємодіють між собою. Нарешті, схема розгортання використовується для моделювання фізичного розгортання системи. На ньому зображені різні апаратні та програмні компоненти та їх з'єднання [24], [25].

На закінчення, UML – це потужний інструмент, який можна використовувати для проектування вебзастосунків. Він забезпечує стандартизовану нотацію, яка допомагає створювати моделі, зрозумілі як технічним, так і нетехнічним зацікавленим сторонам. Діаграми UML використовуються для моделювання поведінки та структури системи. Використовуючи UML для розробки вебзастосунків, важливо почати з діаграми випадків використання, а потім перейти до діаграми класів, діаграми послідовностей, діаграми дій, діаграми компонентів та діаграми розгортання. Використовуючи UML, розробники можуть створювати вебзастосунки, які добре розроблені, прості для розуміння та супроводу [24], [25].

У програмній інженерії фаза фізичного проектування передбачає перетворення логічного проекту програмної системи у фізичну реалізацію. UML – це мова візуального моделювання, яка часто використовується для фізичного проектування при розробці програмного забезпечення. Нижче наведено кроки, пов'язані з використанням UML [24], [25].

Ідентифікація компонентів та їх фізичне розташування: першим кроком у фізичному проектуванні є ідентифікація різних компонентів системи та їх фізичного розташування. Це включає в себе відображення логічного проекту на фізичну інфраструктуру. Наприклад, у логічному проекті може бути зазначено, що система має три рівні - рівень презентації, рівень додатків та рівень бази даних. Етап фізичного проектування включатиме визначення фізичного

розташування кожного рівня-наприклад, на яких серверах вони будуть розміщені [24], [25].

Зіставлення логічних компонентів з фізичними компонентами: наступним кроком є зіставлення логічних компонентів з фізичними компонентами. Наприклад, у логічному проекті може існувати клас, що представляє інтерфейс користувача. У фізичній реалізації цей клас може бути реалізований як HTML-сторінка, що обслуговується вебсервером [24], [25].

Використовуйте діаграми розгортання: діаграми розгортання UML можна використовувати для моделювання фізичної архітектури системи. На діаграмі розгортання Показані фізичні вузли та їх взаємозв'язки. Вузли-це фізичні об'єкти, на яких розміщені компоненти. Наприклад, вузол може представляти сервер або робочу станцію. Компоненти – це програмні артефакти, які виконуються на вузлах. Діаграми розгортання можна використовувати, щоб показати, як компоненти розподіляються по різних вузлах [24], [25].

Використовуйте діаграми компонентів (рис. 3.5): діаграми компонентів можна використовувати для моделювання фізичної структури системи. Діаграми компонентів показують компоненти та їх взаємозв'язки. Компоненти-це модульні будівельні блоки системи. Їх можна розглядати як автономні функціональні одиниці. У фізичній реалізації компоненти можуть бути реалізовані у вигляді бібліотек, виконуваних файлів або сценаріїв. Діаграми компонентів можна використовувати, щоб показати, як різні компоненти з'єднані між собою [24], [25].

Використання діаграми пакетів (рис. 3.6): діаграми пакетів можна використовувати для організації різних компонентів системи. Пакети використовуються для групування пов'язаних компонентів разом. Наприклад, усі компоненти, пов'язані з інтерфейсом користувача, можуть бути згруповані разом у пакет під назвою "інтерфейс користувача". Діаграми пакетів можуть використовуватися для відображення залежностей між різними пакетами у базі даних [24], [25].

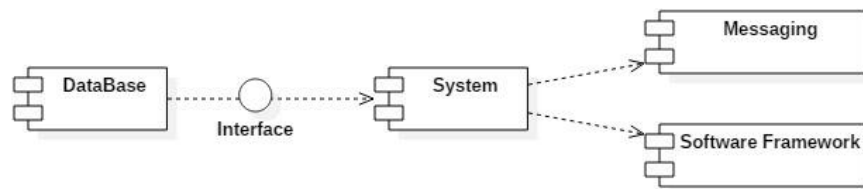


Рисунок 3.5 – Діаграма компонентів

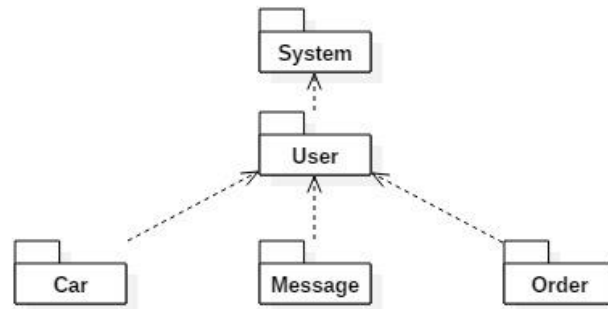


Рисунок 3.6 – Діаграма пакетів

На закінчення, UML може бути використаний для фізичного проектування шляхом ідентифікації компонентів та їх фізичного розташування, відображення логічних компонентів фізичним компонентам, використання діаграм розгортання для моделювання фізичної архітектури, використання діаграм компонентів для моделювання фізичної структури та використання діаграм пакетів для організації різних компонентів. Використовуючи UML для фізичного проектування, розробники програмного забезпечення можуть створювати добре розроблені, зрозумілі програмні системи, які легко обслуговуються та масштабуються [24], [25].

Логічне проектування з використанням UML передбачає створення високорівневого абстрактного уявлення розроблюваної системи, незалежного від будь-якої конкретної технології або реалізації. Мета полягає в тому, щоб зафіксувати функціональні вимоги системи та те, як ці вимоги виконуються компонентами системи та їх взаємодією. Нижче наведено кроки, пов'язані з логічним проектуванням за допомогою UML [24], [25].

Визначення випадків використання: першим кроком у логічному проектуванні є визначення випадків використання системи. Випадки

використання - це зовнішні взаємодії між системою та її користувачами або іншими системами. Вони описують поведінку системи з точки зору користувача. Випадки використання зазвичай відображаються на діаграмах випадків використання, які показують учасників (користувачів або інші системи) та випадки використання, в яких вони беруть участь [24], [25].

Створення моделі домену: наступним кроком є створення моделі домену. Модель домену - це концептуальна модель, яка представляє об'єкти, концепції та взаємозв'язки в проблемній області. Він описує проблемну область незалежно від будь-якого конкретного рішення чи реалізації. Моделі домену зазвичай відображаються на діаграмах класів, які показують класи, їх атрибути та їх взаємозв'язки [24], [25].

Визначення меж системи: межа системи визначає, що знаходиться всередині системи і що знаходиться за її межами. Межі системи зазвичай визначаються випадками використання та учасниками, які беруть участь у них. Межа системи важлива, оскільки вона допомагає визначити компоненти, які потрібно розробити, та інтерфейси між ними [24], [25].

Ідентифікація компонентів системи (рис. 3.7): як тільки межі системи визначені, наступним кроком є ідентифікація компонентів системи. Системний компонент - це модульний будівельний блок, який виконує певну функцію в системі. Компоненти зазвичай представлені на діаграмах компонентів, які показують інтерфейси між компонентами та їх залежності [24], [25].

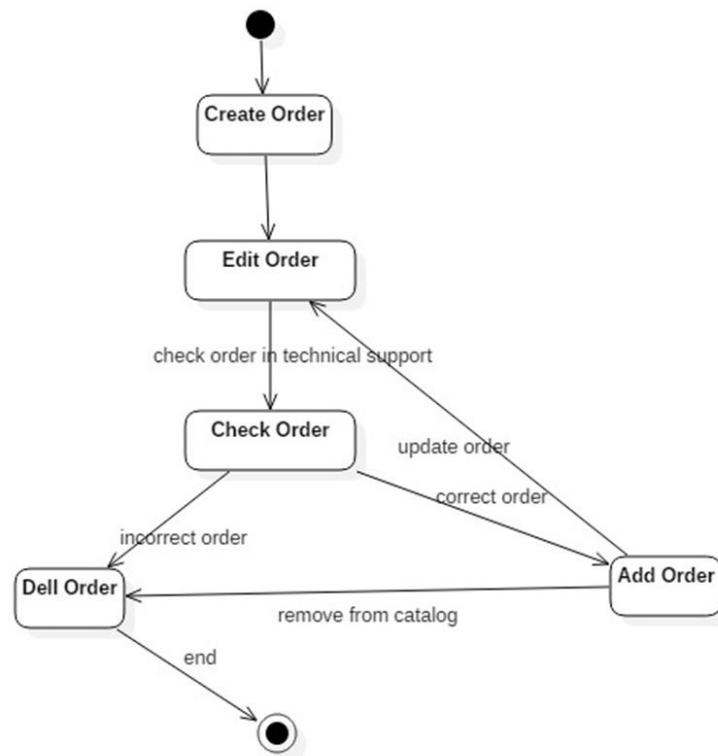


Рисунок 3.7 – Діаграма станів

Визначення поведінки системи: наступним кроком є визначення поведінки компонентів системи. Це передбачає вказівку методів або операцій, які виконують компоненти, та повідомлень або подій, які їх запускають. Поведінка компонентів системи зазвичай відображається на діаграмах послідовності, які показують взаємодію між об'єктами з часом [24], [25].

Визначення архітектури системи (рис. 3.8): після того, як компоненти системи та їх поведінка визначені, наступним кроком є визначення архітектури системи. Архітектура системи визначає структуру системи високого рівня та те, як компоненти організовані та взаємодіють між собою. Архітектура системи зазвичай представлена на діаграмах компонентів та кооперацій [24], [25].

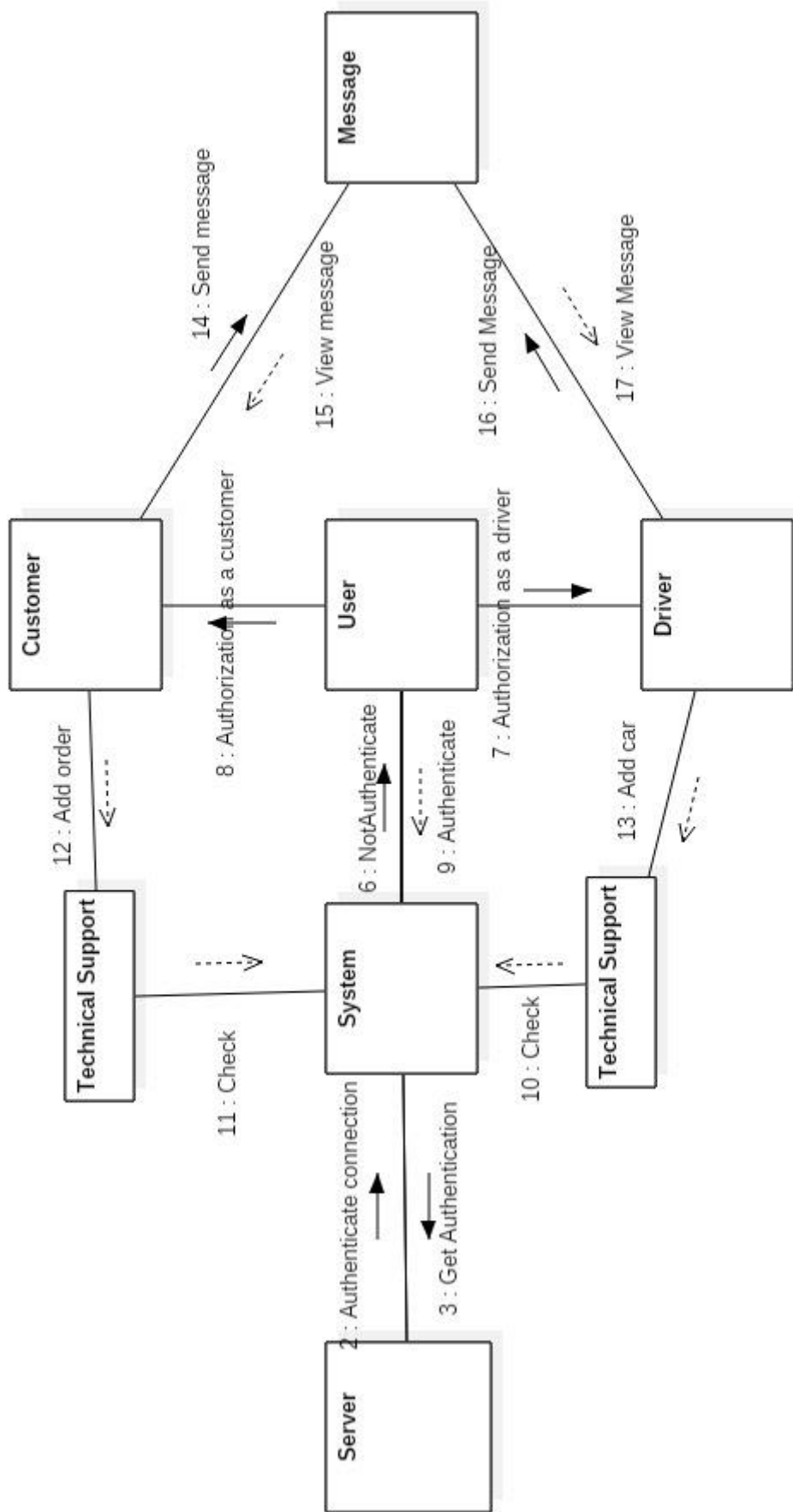


Рисунок 3.8 – Діаграма кооперації

На закінчення, логічне проектування за допомогою UML включає визначення випадків використання, створення моделі домену, визначення меж системи, ідентифікацію компонентів системи, визначення поведінки системи та визначення архітектури системи. Використовуючи UML для логічного проектування, розробники програмного забезпечення можуть створити абстрактне представлення системи високого рівня, незалежне від будь-якої конкретної реалізації чи технології. Це допомагає гарантувати, що система відповідає функціональним вимогам користувачів і зацікавлених сторін і може бути легко підтримана і розширена в майбутньому [24], [25].

3.2 Проектування інтерфейсу

Прототипи інтерфейсу наведено на рисунках нижче.

На рис. 3.9 зображено можливий дизайн головної сторінки сервісу



Рисунок 3.9 – Головна сторінка сервісу

На рис. 3.10 наведено вікно реєстрації



The registration window features a header with a small icon of two figures shaking hands and the title "РЕЄСТРАЦІЯ НА СЕРВІСІ". Below the header, there are nine input fields for registration details: "ПРИЗВИЩЕ", "ІМ'Я", "ПО БАТЬКОВІ", "LOGIN", "ПАРОЛЬ", "ПІДТВЕРДЖЕННЯ ПАРОЛЮ", "e-mail", "МОБІЛЬНИЙ ТЕЛЕФОН", and "КОД ПІДТВЕРДЖЕННЯ". At the bottom center, there is a button labeled "ЗАРЕЄСТРУВАТИСЯ".

Рисунок 3.10 – Вікно реєстрації

На рис. 3.11 наведено вікно додавання нового автомобіля (водія)



The "ДОДАВАННЯ АВТОМОБІЛЯ" window contains several input fields: "ТИП МАШИНИ", "ВАНТАЖОПІДЙОМНІСТЬ", "ЦІНА ЗА ГОДИНУ РОБОТИ", "ВИД ПАЛИВА", "МЕЖІ РОБОТИ ВОДІЯ", and "РІК ВИПУСКУ МАШИНИ". To the right of these are fields for "e-mail" and "тел.". A large box on the left contains a red plus sign and the text "ДОДАТИ ФОТО АВТОМОБІЛЯ". At the bottom right, there is a button labeled "ДОДАТИ".

Рисунок 3.11 – Вікно додавання нового автомобіля (водія)

На рис. 3.12 наведено вікно додавання нового замовлення

ДОДАВАННЯ ЗАМОВЛЕННЯ

ТИП МАШИНИ

ТИП ВАНТАЖУ

ЦІНА

ТОЧКА А
(вказати адресу завантаження вантажу)

ТОЧКА Б
(вказати адресу розвантаження вантажу)

МАСА ВАНТАЖУ
(вказати в кг/тонн)

e-mail

тел.

ДОДАТИ

Рисунок 3.12 – Додавання нового замовлення

На рис. 3.13 зображено сторінку технічної підтримки

ЗАМОВЛЕННЯ	АВТОМОБІЛІ	Admin
Замовлення 1		
Замовлення 2		
ДОДАТИ		ВИДАЛИТИ

Рисунок 3.13 – Сторінка технічної підтримки

3.3 Висновки за розділом 3

Використовуючи засоби проєктування мови UML, було проведено фізичне та логічне проєктування вебзастосунку. Спираючись на визначені принципи розробки графічного інтерфейсу було прийнято рішення, щодо подальшої реалізації ПЗ.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Підставою для розроблення є проблема з вантажоперевезеннями. Ця проблема досить велика в тому плані що є замовники, яким просто треба щоб їх вантаж перевезли до міста призначення, але не можуть знайти водія з нормальною автівкою. А є водії, які в наш час маючи своє авто не мають можливості заробляти гроші по якимось причинам і самою глобальною причиною є те, що не має наприклад сервісу який би був простий і зручний у використанні, та за допомогою якого можна було б шукати ці замовлення. Ось ця проблема і стала підставою для розробки системи.

Транспортний вебсервіс «Вантажні перевезення». Цей сервіс вирішить проблему з вантажоперевезенням. Сервіс надає можливість користувачам виступати, як зі сторони замовника, якому потрібно перевезти свій вантаж з однієї точки в іншу, тому він може вибрати водія, який виконає замовлення, так і зі сторони водія, який в свою чергу може взяти замовлення на перевезення вантажу та виконати його.

Функціональні вимоги водія:

- реєстрація в системі;
- авторизація як водій;
- додавання транспорту;
- видалення транспорту;
- зв'язок водія з замовником;
- можливість створення рейтингу водію;
- можливість брати замовлення водію;
- створення калькулятора для розрахунку вартості роботи водія;
- додавання мапи для перегляду кілометражу між точками

перевезення.

Функціональні вимоги замовника:

- реєстрація в системі;
- авторизація як замовник;
- додавання замовлення;
- видалення замовлення;
- зв'язок замовника з водієм;
- можливість створення рейтингу замовника;
- можливість вибору транспорту замовникові.

Функціональні вимоги співробітників платформи:

- технічна підтримка;
- видалення некоректних заявок замовників та водіїв.

4.1.1 Вимоги до інтерфейсу

Інтерфейс повинен бути зручний, як для користувача так і для адміністратора.

- мінімальні затрати ресурсів користувача при вводі, модифікації та перегляді даних;
- максимальна взаємодія користувача з системою.

4.2 Умови виконання програми

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;
- розгорнута серверна операційна система (ОС);
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК з IBM-сумісною архітектурою;
- процесор із тактовою частотою від 2,0 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet;
- встановлений браузер.

4.3 Розгортання застосунку

Поетапне розгортання вебсерверу та вебзастосунку зображено на рисунках, зазначених нижче.

На рис. 4.1 наведено вікно програми XAMPP Control Panel.

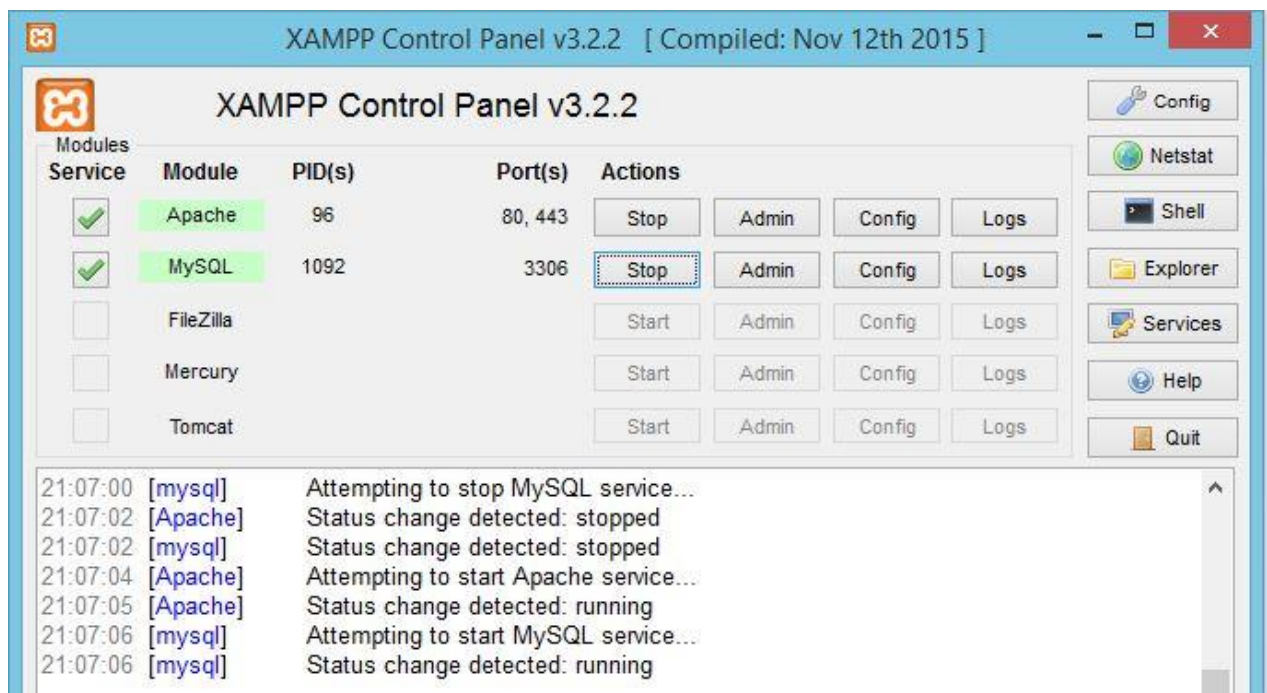


Рисунок 4.1 – XAMPP Control Panel

На рис. 4.2 зображено каталог розміщення Open Card шаблону системи.

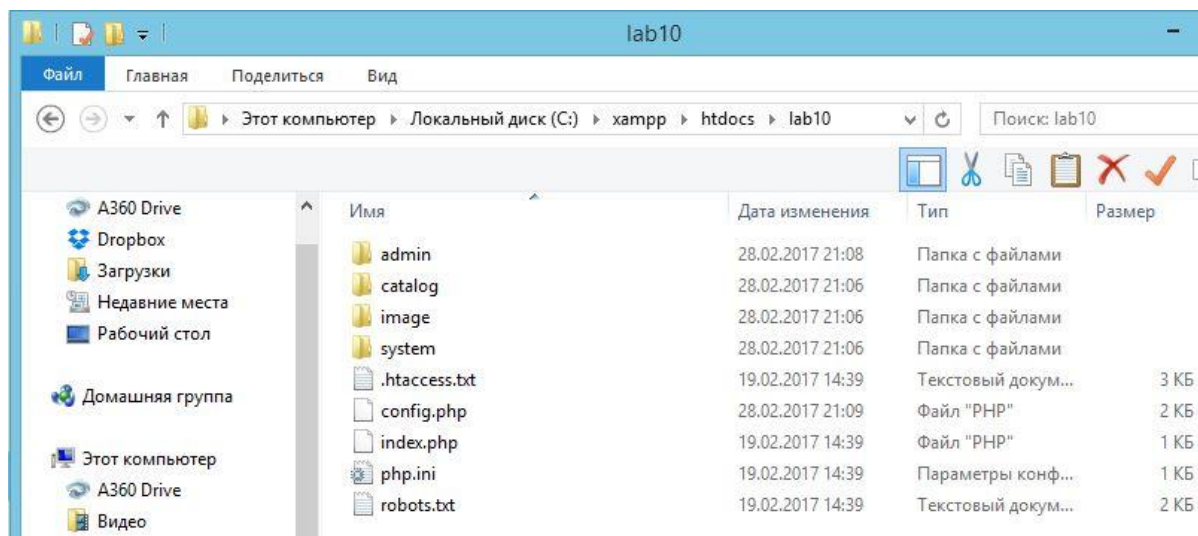


Рисунок 4.2 – Каталог розміщення Open Card шаблону системи

На рис. 4.3 зображено вікно створеної бази даних на сервері.

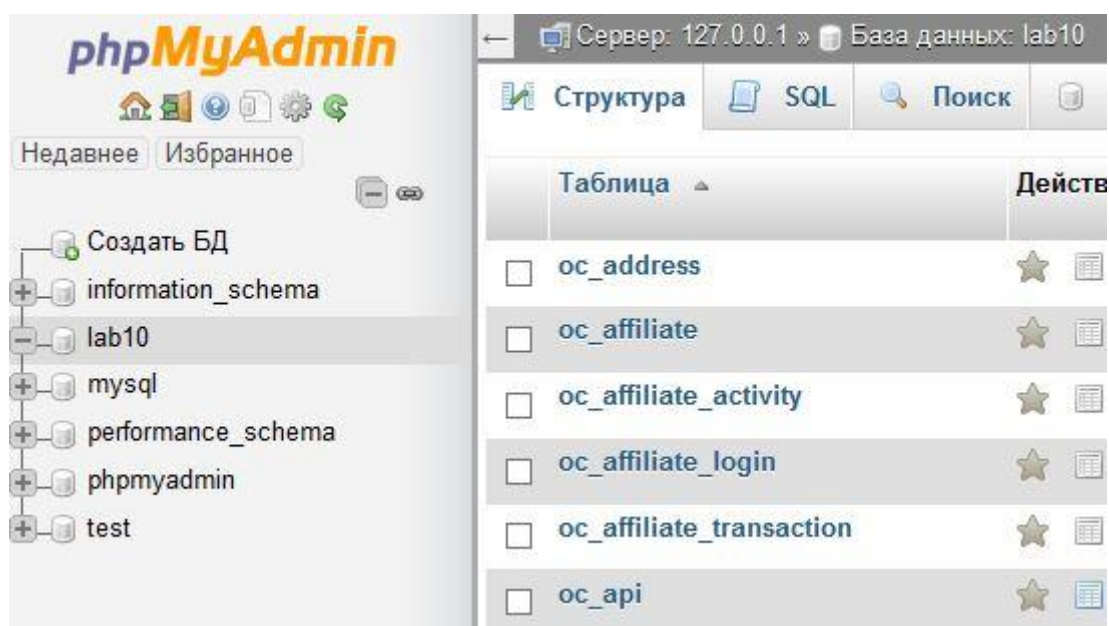


Рисунок 4.3 – Створена база даних на сервері

На рис. 4.4 наведено розгорнуте дерево бази даних для застосунку.

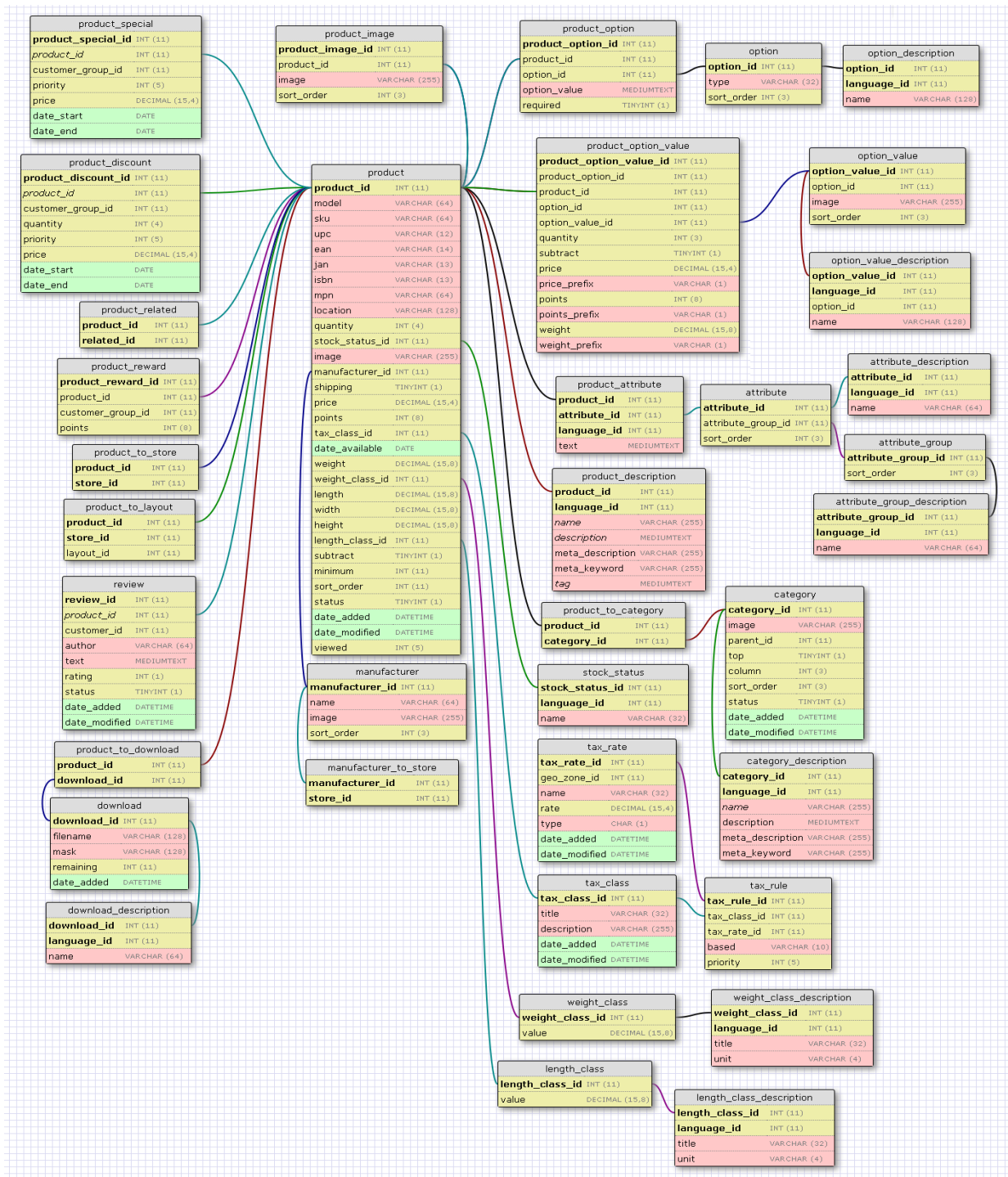


Рисунок 4.4 – Розгорнута база даних для застосунку

4.3.1 Звернення до програми

Звертання до програми від користувача організовано через відправлення запиту про перехід на сайт, використовуючи адресний рядок у веббраузері.

Також звертання до програми відбувається, коли адміністратор змінює налаштування або виконує операції з користувачами.

4.3.2 Вхідні й вихідні дані

Вхідними даними для застосунку є база даних, що включає у себе таблиці із даними про користувачі, тематичні дописи сайту та особливості організації застосунку.

Вихідними даними є відповіді на запити користувача, які надаються системою на вебсторінках застосунку.

4.4 Методика та результати тестування

Тестування вебзастосунків в різних браузерях необхідно для забезпечення коректної і послідовної роботи програми на різних платформах і пристроях. Ось кілька причин, чому важливо тестувати веб - програми в різних браузерах.

Сумісність браузера: веббраузери можуть відображати вебсторінки по-різному залежно від їх функцій та реалізації, що призводить до проблем сумісності. Тестування вебзастосунків у різних браузерах гарантує, що програма працює коректно та послідовно у всіх популярних веббраузерах, включаючи Chrome, Firefox, Safari, Edge та Internet Explorer.

Досвід користувачів: користувачі можуть використовувати різні браузери для доступу до вебпрограми, а тестування гарантує, що програма виглядає та поводиться однаково у всіх них. Це покращує користувацький інтерфейс і гарантує, що користувачі зможуть отримати доступ до програми незалежно від бажаного браузера.

Доступність: різні браузери підтримують різні спеціальні функції, такі як перетворення тексту в мову, зчитувачі з екрана та навігація клавіатурою. Тестування вебзастосунків у різних браузерах гарантує, що програма доступна

для користувачів з обмеженими можливостями та відповідає відповідним рекомендаціям щодо доступності.

Безпека: різні браузері мають різні функції безпеки та налаштування, а тестування вебзастосунків у різних браузерах гарантує, що програма є безпечною та не має вразливостей чи проблем, які могли б скомпрометувати дані користувачів.

Продуктивність: різні браузері мають різні характеристики продуктивності, а тестування вебзастосунків у різних браузерах гарантує оптимальну роботу програми у всіх браузерах та пристроях.

Частка ринку: різні браузері мають різну частку ринку, а тестування вебзастосунків у різних браузерах гарантує, що програма працює коректно та послідовно для більшості користувачів, незалежно від бажаного браузера.

На закінчення, тестування вебзастосунків у різних браузерах важливо для забезпечення сумісності браузерів, покращення користувацького досвіду, забезпечення доступності, підвищення безпеки, оптимізації продуктивності та охоплення ширшої аудиторії. Це необхідно для створення високоякісного вебзастосунка, який відповідає потребам користувачів та зацікавлених сторін.

Сумісність з браузерами:

На рис. 4.5 наведено сумісність з браузером Google Chrome.

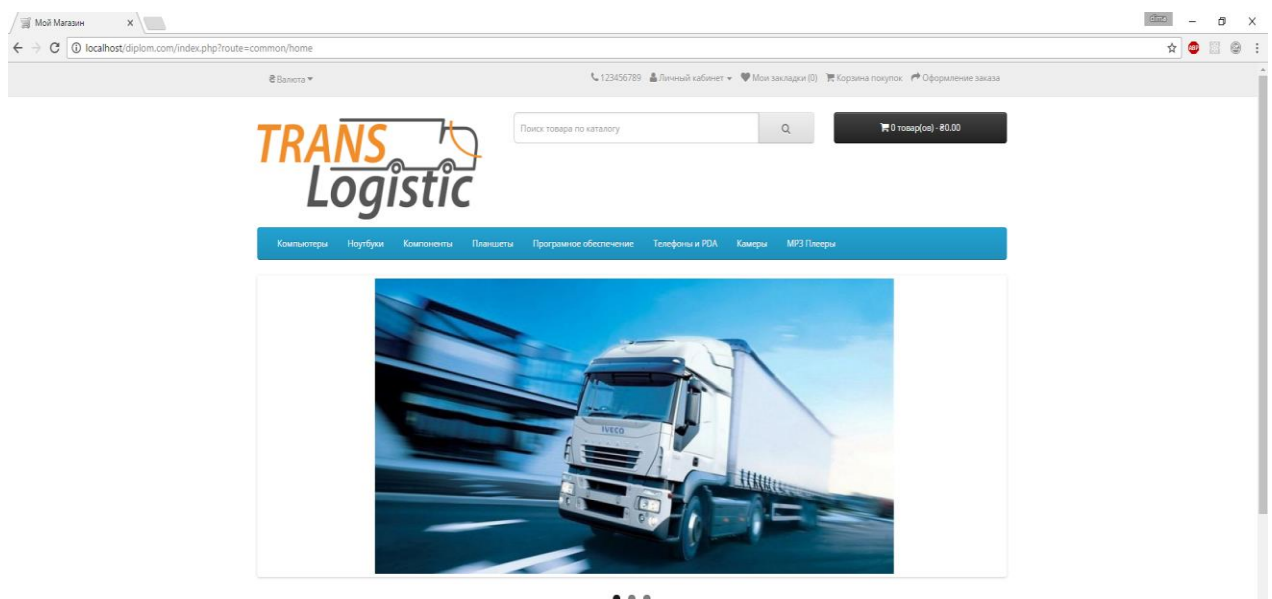


Рисунок 4.5 – Сумісність з браузером Google Chrome

На рис. 4.6 наведено сумісність з браузером Microsoft Edge.

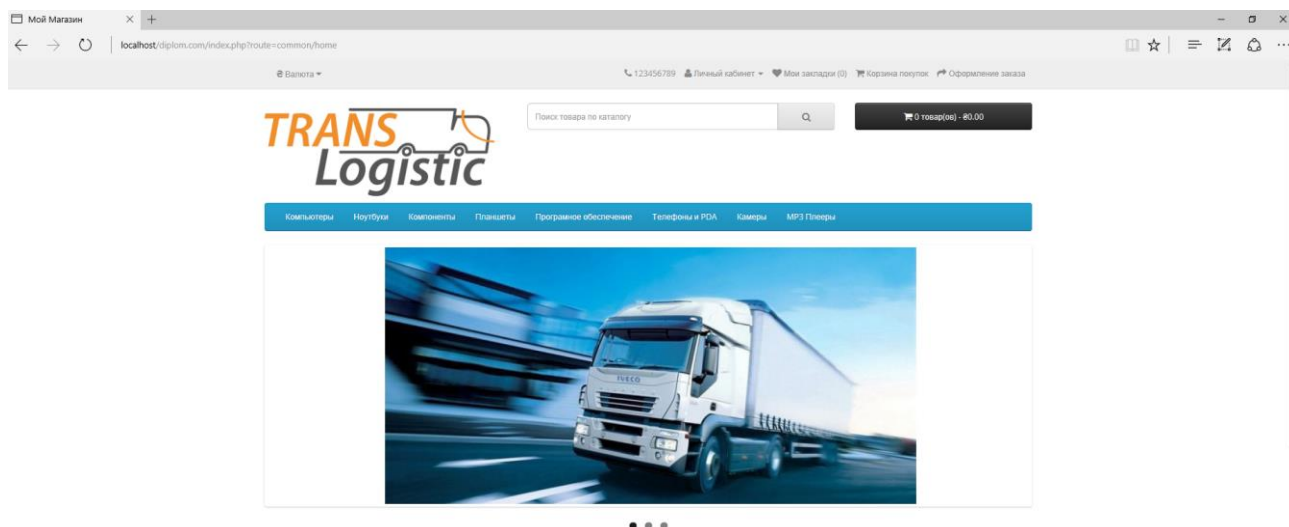


Рисунок 4.6 – Сумісність з браузером Microsoft Edge

На рис. 4.7 наведено сумісність з браузером Opera.

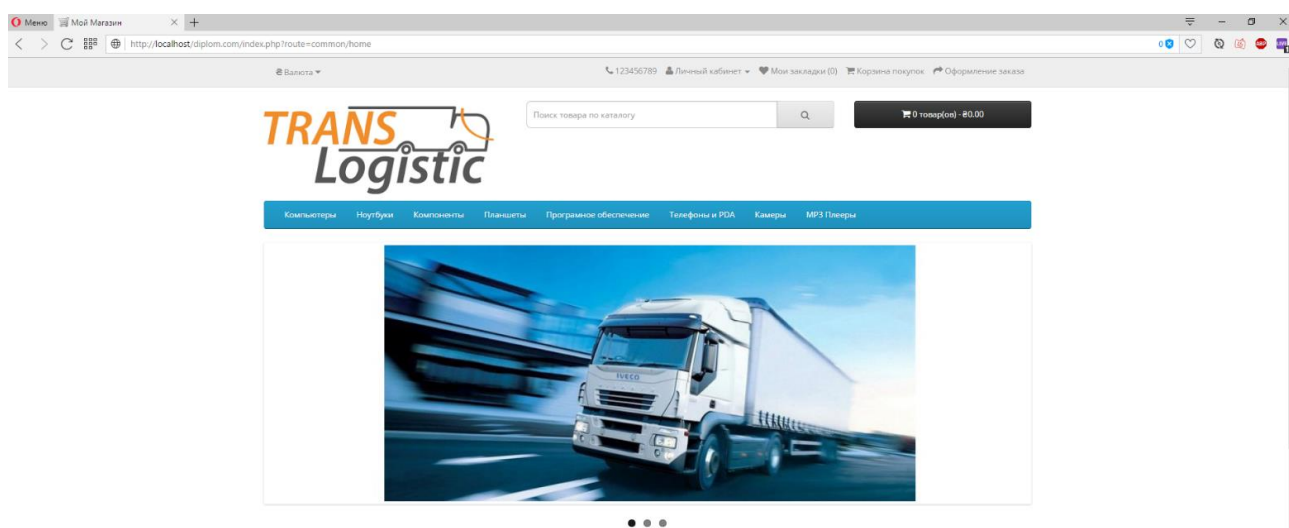


Рисунок 4.7 – Сумісність з браузером Opera

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було реалізовано головну мету – програмну реалізацію інформаційного ресурсу компанії перевезень вантажу. Для цього було виконано ряд завдань:

- проаналізувати предметну область послуг з перевезення вантажу;
- дослідити існуючі аналоги інформаційних ресурсів компаній перевезень вантажу;
- визначивши переваги та недоліки існуючих аналогів сформувати перелік функціональних вимог до нового інформаційного ресурсу, що буде якісно відрізнятися від аналогів;
- визначити технічні засоби програмної реалізації нового рішення;
- виконати проєктування розробки нового застосунку;
- на основі виконаного проєктування реалізувати новий інформаційний ресурс компанії перевезень вантажу;
- дослідити та протестувати нове рішення.

Розроблена структура вебзастосунку була реалізована із використанням PHP CMS – WordPress.

Був розроблений сценарій та проведено тестування розробленого вебзастосунку з використанням сценарію тестування на основі основної логіки використання застосунку. Отриманий результат свідчить про те, що тестування дало позитивний результат.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аляб'єва О. М. Теоретичні основи формування сутності поняття "логістика" та "логістичне управління" [Текст] / О. М. Аляб'єва // Науковий вісник Ужгородського національного університету. Серія : Міжнародні економічні відносини та світове господарство. - 2016. - Вип. 8(1). - С. 5-7.
2. Кудринський Р. Б. Логістика сучасних процесів транспортного забезпечення [Текст] / Р. Б. Кудринський, Л. А. Савченко, І. І. Махмудов // Механізація та електрифікація сільського господарства. - 2017. - Вип. 5. - С. 253-262.
3. Кустріч Л. О. Логістика та інновації: концепція, стратегія управління та взаємодія [Текст] / Л. О. Кустріч, М. О. Гоменюк // Бізнес Інформ. - 2021. - № 1. - С. 89-96
4. Бабайлов В. К. Логістика – це методологія [Текст] / В. К. Бабайлов, Я. С. Левченко // Проблеми і перспективи розвитку підприємництва. - 2021. - № 2. - С. 4-14
5. Ніколюк О. В. Логістика функціональних областей: теоретичні підходи [Текст] / О. В. Ніколюк, К. Ю. Соколюк, Л. І. Димбовська // Економіка харчової промисловості. - 2021. - Т. 13, Вип. 4. - С. 34-40.
6. Гуштан Т. В. Логістика в діяльності торговельних підприємств України, що виходять на ринки ЄС [Текст] / Т. В. Гуштан // Український журнал прикладної економіки. - 2021. - Т. 6, № 1. - С. 269-279.
7. Магер Є. В. Розробка ІТ-проекту "Логістика" на основі хмарних технологій [Текст] / Є. В. Магер, Ф. В. Рудь, Б. Л. Ткаченко, В. В. Фіалковський, Ю. В. Триус // STE Workshop Proceedings. - 2013. - Vol. 1. - С. 20-21.
8. Федченко С. В. Комерційна логістика торгового підприємства [Текст] / С. В. Федченко // Молодий вчений. - 2020. - № 10(2). - С. 205-208.
9. Запотоцький С. П. Транспортна логістика: основні виклики для України [Текст] / С. П. Запотоцький, В. А. Паренюк // Вісник Київського

національного університету імені Тараса Шевченка. Географія. - 2016. - Вип. 2. - С. 42-47.

10. Uber Freight [Electronic resource]. – Access mode: <https://www.uberfreight.com/>

11. There's a 30 percent chance the truck next to you on the highway is empty—a sign of inefficiency in an aging industry. Uber Freight's Lior Ron is on a quest to change that [Electronic resource]. – Access mode: <https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/logistics-disruptors-replicating-ubers-success-in-the-trucking-industry>

12. Webfleet [Electronic resource]. – Access mode: https://www.webfleet.com/en_us/webfleet/

13. A brief guide to the history of telematics [Electronic resource]. – Access mode: https://media.webfleet.com/legacy/ttt/lib/assets/ttt/doc/whitepapers/en_gb/ttt-history-of-telematics.pdf

14. Нова Пошта [Електронне джерело] – Режим доступу: <https://novaposhta.ua/>

15. Інформація про Нова Пошта [Електронне джерело] – Режим доступу: <https://forbes.ua/profile/nova-poshta-260>

16. Gozhyj A. P. Method for developing and modelling composite web-services [Text] / A. P. Gozhyj, I. A. Kalinina, V. A. Gozhyj // Herald of advanced information technology. - 2022. - Vol. 5, no. 3. - С. 185-197.

17. Eko C. E. Design and implementation of a fault tolerant web-based examination system for developing countries [Text] / C. E. Eko, I. E. Eteng, E. E. Essien // Eastern-European journal of enterprise technologies. - 2022. - № 1(2). - С. 58-67.

18. Is it worth being a PHP developer in 2023. Scope of PHP developers [Electronic resource]. – Access mode: <https://www.linkedin.com/pulse/worth-being-php-developer-2023-scope>

19. Are PHP Web Development Services Any Good for Building Websites in 2023? Benefits and Real Examples [Electronic resource]. – Access mode: <https://www.altamira.ai/blog/build-websites-with-php/>

20. PHP in 2023 [Електронне джерело] – Режим доступу: <https://stitcher.io/blog/php-in-2023>

21. Bogoslov I. A. Linking Facebook to Wordpress for educational purposes – a proposed architecture sustaining social learning [Text] / I. A. Bogoslov, M. R. Georgescu, A. G. Pitic // Bulletin of Taras Shevchenko Kyiv National University. Economy. - 2020. - Issue 5. - P. 6-11.

22. How to become a WordPress Developer in 2023 [Electronic resource]. – Access mode: <https://aarontjones.co.uk/blog/support/how-to-become-a-wordpress-developer-in-2023/>

23. Актуальність WordPress: чому з ним треба працювати у 2023 році [Електронне джерело] – Режим доступу: <https://dou.ua/forums/topic/42889/>

24. Automated Prototyping of User Interfaces based on UML Scenarios [Electronic resource]. – Access mode: <http://www.iro.umontreal.ca/~keller/Publications/Papers/2006/ase-2006.pdf>

25. UML Activity Diagram Semantics and Automated GUI Prototyping [Electronic resource]. – Access mode: <https://core.ac.uk/download/pdf/55336476.pdf>

ДОДАТОК А
Технічне завдання

Вступ

Підставою для розроблення є проблема з вантажоперевезеннями. Ця проблема досить велика в тому плані що є замовники, яким просто треба щоб їх вантаж перевезли до міста призначення, але не можуть знайти водія з нормальною автівкою. А є водії, які в наш час маючи своє авто не мають можливості заробляти гроші по якимось причинам і самою глобальною причиною є те, що не має наприклад сервісу який би був простий і зручний у використанні, та за допомогою якого можна було б шукати ці замовлення. Ось ця проблема і стала підставою для розробки системи.

Транспортний вебсервіс «Вантажні перевезення». Цей сервіс вирішить проблему з вантажоперевезенням. Сервіс надає можливість користувачам виступати, як зі сторони замовника, якому потрібно перевезти свій вантаж з однієї точки в іншу, тому він може вибрати водія, який виконає замовлення, так і зі сторони водія, який в свою чергу може взяти замовлення на перевезення вантажу та виконати його.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація інформаційного ресурсу компанії перевезень вантажу», затверджене наказом Національного університету «Запорізька політехніка» № 87 від 04 квітня 2022 р.

А.2 Основні вимоги до програми, що розробляється

Головне завдання: реалізувати транспортний вебсервіс призначений для розміщення замовлень, та ефективної комунікації між замовниками та водіями. Експлуатаційне призначення системи це використання її власниками вантажу та власниками вантажних автівок.

A.2.1 Вимоги до функціональних характеристик

Функціональні вимоги водія:

- реєстрація в системі;
- авторизація як водій;
- додавання транспорту;
- видалення транспорту;
- зв'язок водія з замовником;
- можливість створення рейтингу водію;
- можливість брати замовлення водію;
- створення калькулятора для розрахунку вартості роботи водія;
- додавання мапи для перегляду кілометражу між точками

перевезення.

Функціональні вимоги замовника:

- реєстрація в системі;
- авторизація як замовник;
- додавання замовлення;
- видалення замовлення;
- зв'язок замовника з водієм;
- можливість створення рейтингу замовника;
- можливість вибору транспорту замовникові.

Функціональні вимоги співробітників платформи:

- технічна підтримка;
- видалення некоректних заявок замовників та водіїв.

A.2.2 Вимоги до інтерфейсу програми

Інтерфейс повинен бути зручний, як для користувача так і для адміністратора:

- мінімальні затрати ресурсів користувача при вводі, модифікації та перегляді даних;
- максимальна взаємодія користувача з системою.

A.2.3 Вимоги до надійності

Система повина бути захищеною від посягань злоумисників на персональні дані користувачів. Кожен акаунт повинен бути прив'язаний до електронних скриньок користувачів системи.

Для використання можливостей системи користувач повинен мати один з відомих браузерів останніх версій.

A.2.4 Умови експлуатації

Експлуатаційними вимогами є:

- Інтернет;
- сервер;
- персональний комп'ютер (ПК);
- актуальний браузер.

A.2.5 Вимоги до складу та параметрів технічний засобів

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;

- розгорнута серверна ОС;
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК з IBM-сумісною архітектурою;
- процесор із тактовою частотою від 2,0 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet;
- встановлений браузер.

A.3 Вхідні дані до роботи

Вхідними даними для застосунку є база даних, що включає у себе таблиці із даними про користувачі, тематичні дописи сайту та особливості організації застосунку.

Вихідними даними є відповіді на запити користувача, які надаються системою на вебсторінках застосунку.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Підставою для розроблення є проблема з вантажоперевезеннями. Ця проблема досить велика в тому плані що є замовники, яким просто треба щоб їх вантаж перевезли до міста призначення, але не можуть знайти водія з нормальною автівкою. А є водії, які в наш час маючи своє авто не мають можливості заробляти гроші по якимось причинам і самою глобальною причиною є те, що не має наприклад сервісу який би був простий і зручний у використанні, та за допомогою якого можна було б шукати ці замовлення. Ось ця проблема і стала підставою для розробки системи.

Транспортний вебсервіс «Вантажні перевезення». Цей сервіс вирішить проблему з вантажоперевезенням. Сервіс надає можливість користувачам виступати, як зі сторони замовника, якому потрібно перевезти свій вантаж з однієї точки в іншу, тому він може вибрати водія, який виконає замовлення, так і зі сторони водія, який в свою чергу може взяти замовлення на перевезення вантажу та виконати його.

Б.2 Функціональне призначення

Функціональні вимоги водія:

- реєстрація в системі;
- авторизація як водій;
- додавання транспорту;
- видалення транспорту;
- зв'язок водія з замовником;
- можливість створення рейтингу водію;
- можливість брати замовлення водію;
- створення калькулятора для розрахунку вартості роботи водія;
- додавання мапи для перегляду кілометражу між точками перевезення.

Функціональні вимоги замовника:

- реєстрація в системі;
- авторизація як замовник;
- додавання замовлення;
- видалення замовлення;
- зв'язок замовника з водієм;
- можливість створення рейтингу замовника;
- можливість вибору транспорту замовникові.

Функціональні вимоги співробітників платформи:

- технічна підтримка;
- видалення некоректних заявок замовників та водіїв.

Б.3 Опис логічної структури

Загальна схема системи за обраною темою (рис. Б.1).

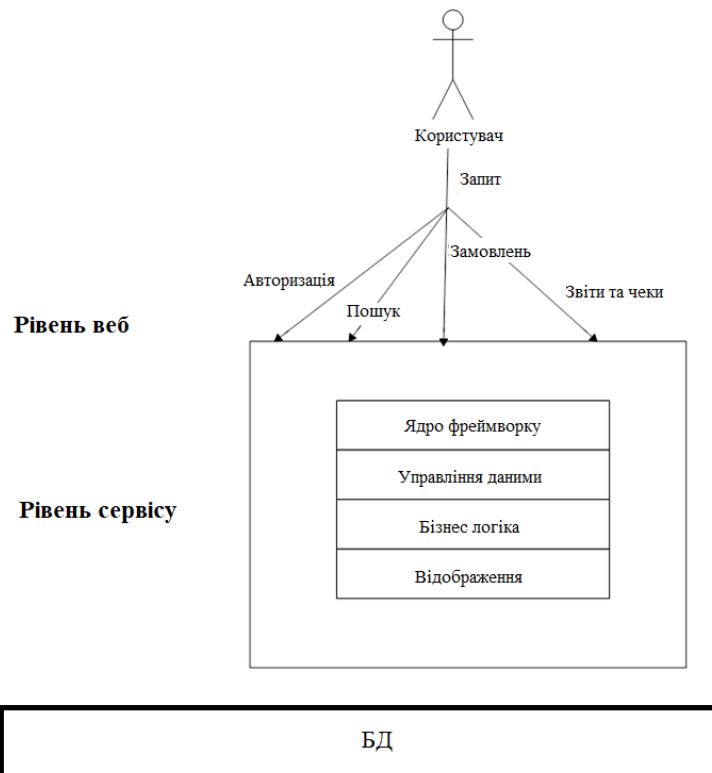


Рисунок Б.1 – Загальна схема системи за обраною темою

Архітектура вебзастосунку базується на моделі Модель-вид-контролер (MVC) (рис. Б.2).

Модель здійснює роботу з базою даних та саме у моделях повинні бути зосереджені методи роботи з даними.

Контролер оброблює дані з моделі, якщо вони потрібні, і генерує Вид з цими даними, де інформація доноситься до користувача з відповідними графічними елементами. Взаємодія користувача з додатком відбувається через Контролер.

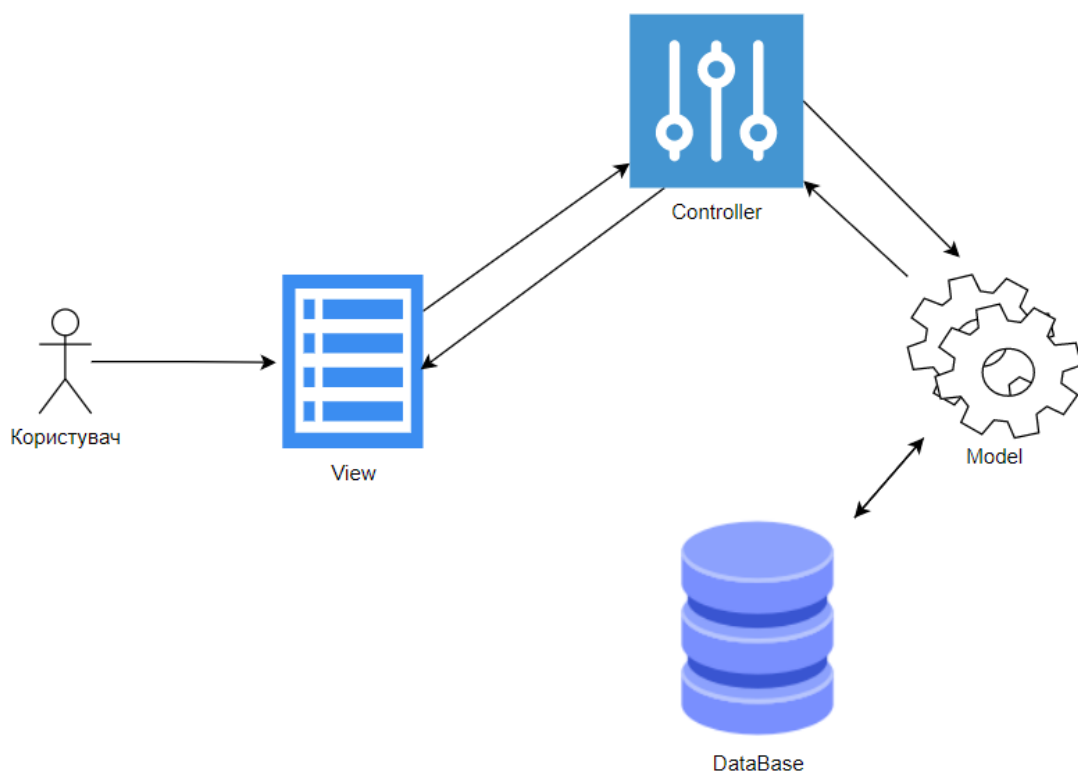


Рисунок Б.2 – Архітектура вебзастосунку

Б.4 Використані технічні засоби

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;
- розгорнута серверна ОС;

- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК з IBM-сумісною архітектурою;
- процесор із тактовою частотою від 2,0 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet;
- встановлений браузер.

Б.5 Виклик і завантаження

Звертання до програми від користувача організовано через відправлення запиту про перехід на сайт, використовуючи адресний рядок у веббраузері. Також звертання до програми відбувається, коли адміністратор змінює налаштування або виконує операції з користувачами.

Б.6 Вхідні та вихідні дані

Вхідними даними для застосунку є база даних, що включає у себе таблиці із даними про користувачі, тематичні дописи сайту та особливості організації застосунку.

Вихідними даними є відповіді на запити користувача, які надаються системою на вебсторінках застосунку.

ДОДАТОК В
Текст програми

```

<?php
//This script will handle login
session_start();

// check if the user is already logged in
if(isset($_SESSION['us_login']))
{
    header("location: welcome.php");
    exit;
}
require_once "config.php";

$us_login = $pswrd = "";
$error = "";

// if request method is post
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if(empty(trim($_POST['us_login'])) ||
    empty(trim($_POST['pswrd'])))
    {
        $error = "Please enter us_login + pswrd";
    }
    else{
        $us_login = trim($_POST['us_login']);
        $pswrd = trim($_POST['pswrd']);
    }

    if(empty($error))
    {
        $sql = "SELECT id, us_login, pswrd FROM users WHERE
us_login = ?";
        $stmt = sql_conn_prepare($conn, $sql);
        sql_conn_stmt_bind_param($stmt, "s",
$params_us_login);
        $params_us_login = $us_login;

        // Try to execute this statement
        if(sql_conn_stmt_execute($stmt)){
            sql_conn_stmt_store_result($stmt);
            if(sql_conn_stmt_num_rows($stmt) == 1)
            {

```

```

sql_conn_statemnt_bind_result($statemnt, $id, $us_login,
$hashed_pswrd);

if(sql_conn_statemnt_fetch($statemnt))
{
    if(pswrd_verify($pswrd,
$hashed_pswrd))
    {
        // this means the pswrd is
correct. Allow user to login
        session_start();
        $_SESSN["us_login"] =
$us_login;
        $_SESSN["id"] = $id;
        $_SESSN["loggedin"] = true;

        //Redirect user to welcome
page
        header("location:
welcome.php");
    }
}

}

}

}

?>

<!doctype html>
<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">

        <!-- Bootstrap CSS -->

```

```

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1
/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">

```

```

<title>PHP login system!</title>
</head>
<body>
<navi class="navibar navibar-expand-lg navibar-dark
bg-dark">
  <a class="navibar-brand" href="#">Php Login System</a>
  <button class="navibar-toggler" type="button" data-
toggle="collapse" data-target="#navibarnaviDropdown"
aria-controls="navibarnaviDropdown" aria-
expanded="false" aria-lbl="Toggle naviigation">
    <span class="navibar-toggler-icon"></span>
  </button>
  <div class="collapse navibar-collapse"
id="navibarnaviDropdown">
    <ul class="navibar-navi">
      <li class="navi-item active">
        <a class="navi-link" href="#">Home <span
class="sr-only">(current)</span></a>
      </li>
      <li class="navi-item">
        <a class="navi-link"
href="Registration.php">Registration</a>
      </li>
      <li class="navi-item">
        <a class="navi-link" href="login.php">Login</a>
      </li>
      <li class="navi-item">
        <a class="navi-link"
href="logout.php">Logout</a>
      </li>
    </ul>
  </div>
</navi>

<div class="container mt-4">
<h3>Please Login Here:</h3>

```

```

<hr>

<form action="" method="post">
  <div class="form-group">
    <lbl for="exampleInputEmail1">us_login</lbl>
    <input type="text" name="us_login" class="form-
control" id="exampleInputEmail1" aria-
describedby="emailHelp" placeholder="Enter us_login">
  </div>
  <div class="form-group">
    <lbl for="exampleInputpswrd1">pswrd</lbl>
    <input type="pswrd" name="pswrd" class="form-
control" id="exampleInputpswrd1" placeholder="Enter
pswrd">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input"
id="exampleCheck1">
    <lbl class="form-check-lbl"
for="exampleCheck1">Check me out</lbl>
  </div>
  <button type="submit" class="btn btn-
primary">Submit</button>
</form>

</div>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS
-->
  <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.
14.7/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJK6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
js/bootstrap.min.js" integrity="sha384-

```

```

JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDs4x0xI
M+B07jRM" crossorigin="anonymous"></script>
<header>
    <div class="main">
        <ul>
            <li><a
href=" ../HTML/index.html">Home</a></li>
            <li class="active"><a
href=" ../HTML/login.html">Login</a></li>
            <li><a
href=" ../HTML/course.html">Course</a></li>
            <li><a
href=" ../HTML/gallery.html">Gallery</a></li>
            <li><a
href=" ../HTML/about.html">About</a></li>
            <li><a
href=" ../HTML/about.html.html">about.htmltml</a></li>
        </ul>
    </div>
</header>
</body>
</html>

```

ДОДАТОК Д
Слайди презентації

**Міністерство освіти та науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів**

Програмна реалізація інформаційного ресурсу компанії перевезень вантажу

Виконав: Кирило КОТЛЯРОВ

ст. групи КНТ-219

Керівник: Михайло КОЦУР

к.т.н., доцент

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

- * Об'єкт дослідження – процес програмної реалізації інформаційного ресурсу компанії перевезень вантажу.
- * Предмет дослідження – програмні засоби для реалізації інформаційного ресурсу компанії перевезень вантажу.
- * Метою роботи є програмна реалізація інформаційного ресурсу компанії перевезень вантажу у вигляді кросплатформного веб-застосунку.

2

Рисунок Д.2 – Слайд 2

Порівняння існуючих аналогів				
Критерії порівняння	Uber Freight	WebFleet	Нова Пошта	TransLogistic
Робота в Україні	Ні	Ні	Так	Так
Прямий зв'язок замовника із водієм	Так	Ні	Ні	Так
Трекінг в реальному часі	Так	Та	Так	Так
Кількість факторів вартості доставки	Ні	Ні	Так	Так
Рейтингування перевізників	Так	Ні	Ні	Так

3

Рисунок Д.3 – Слайд 3

Постановка завдання

Головне завдання: реалізувати транспортний веб-сервіс призначений для розміщення замовлень, та ефективної комунікації між замовниками та водіями. Експлуатаційне призначення системи це використання її власниками вантажу та власниками вантажних автівок.

4

Рисунок Д.4 – Слайд 4

Порівняння мов програмування		
Критерій	PHP	JavaScript
Об'єктно-орієнтовна концепція	Повністю підтримує	Відносна підтримка
Робота з СКБД	Повністю реалізована	Вимагає додаткових надбудов
Простота вивчення	Проста	Проста
Гнучкість та масштабованість	Висока	Висока гнучкість, але відносна масштабованість

5

Рисунок Д.5 – Слайд 5

Порівняння CMS		
	WordPress	Joomla
Вимоги до хостингу	Середні	Середні
Впорядкованість застосунку	Чітка архітектура	Ускладнена архітектура
Можливість масштабованості	Можливо	Можливо, але ускладнено
Робота з СКБД	Проста, захищена	Проста, потребує додаткового захисту
Підтримка та сторонні розширення	Активна спільнота, багато розширень	Менш активна спільнота, не багато розширень

6

Рисунок Д.6 – Слайд 6

Загальна схема застосунку

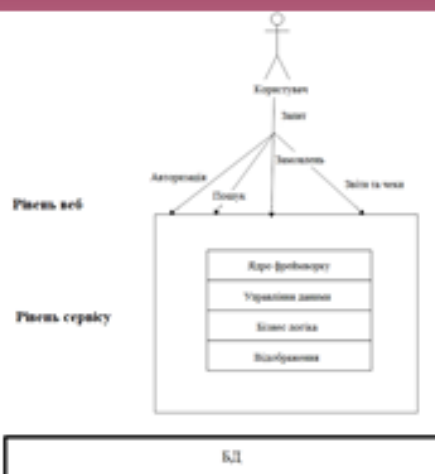


Рисунок Д.7 – Слайд 7

Діаграма класів

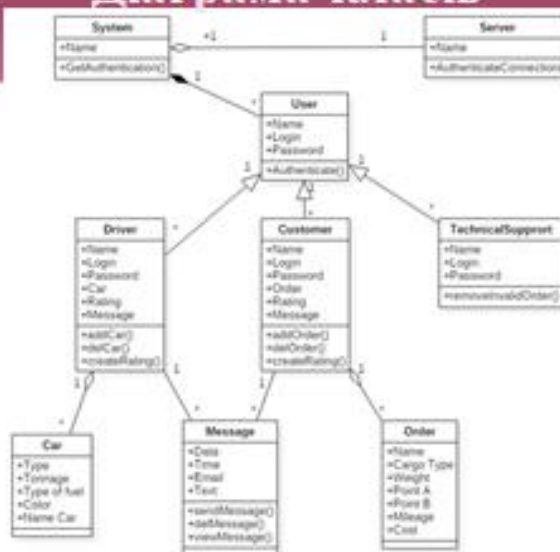


Рисунок Д.8 – Слайд 8

Діаграма прецедентів

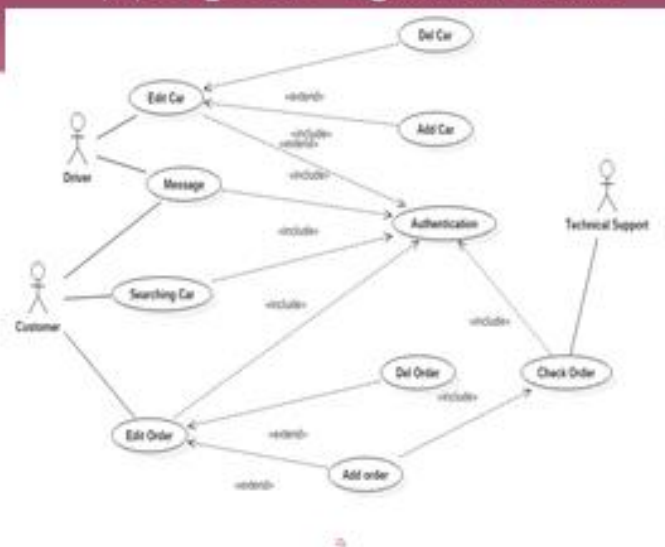


Рисунок Д.9 – Слайд 9

Діаграма кооперації

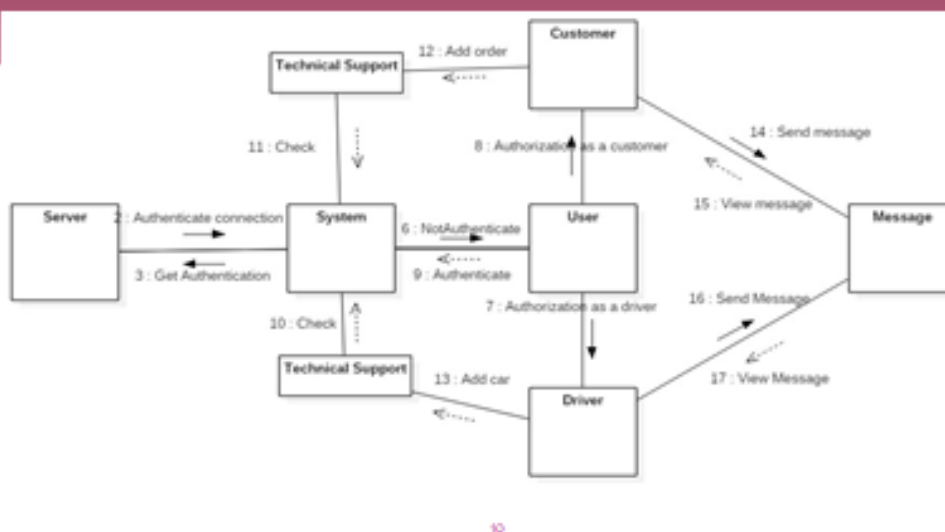


Рисунок Д.10 – Слайд 10



Рисунок Д.11 – Слайд 11

Висновки

- ♦ В ході виконання дипломної кваліфікаційної роботи бакалавра було реалізовано головну мету – програмну реалізацію інформаційного ресурсу компанії перевезень вантажу. Для цього було виконано ряд завдань:
- ♦ проаналізувати предметну область послуг з перевезення вантажу;
- ♦ дослідити існуючі аналоги інформаційних ресурсів компаній перевезень вантажу;
- ♦ визначивши переваги та недоліки існуючих аналогів сформулювати перелік функціональних вимог до нового інформаційного ресурсу, що буде логічно відрізнятися від аналогів;
- ♦ визначити технічні засоби програмної реалізації нового рішення;
- ♦ виконати проектування розробки нового застосунку;
- ♦ на основі виконаного проектування реалізувати новий інформаційний ресурс компанії перевезень вантажу;
- ♦ дослідити та протестувати нове рішення.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок Д.12 – Слайд 12