

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЬЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти
на тему: “ **Проектування веб-додатку для планування і
відслідковування автотранспортних процесів** ”

Виконав: студент гр. Іт-41

Спеціальності 126 – «Інформаційні системи та
технології»

(шифр і назва)

Герега Андрій Богданович

(Прізвище та ініціали)

Керівник: к.т.н., доц. Луб П.М.
(Прізвище та ініціали)

Рецензенти: _____
(Прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ
ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЖИЦЬКОГО

ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти
126 – «Інформаційні системи та технології»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 202_ р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Герезі Андрію Богдановичу

1. Тема роботи: «Проектування веб-додатку для планування і відслідковування автотранспортних процесів»

Керівник роботи Луб Павло Миронович, к.т.н., доцент
Затверджені наказом університету 123/к-с від 25.02.2025.

2. Строк подання студентом роботи 09.06.2025 р.

3. Початкові дані до роботи: 1. Додатки для планування маршрутів перевезень. 2. Програмне забезпечення диспетчеризації перевезень. 3. Методика побудови контекстних діаграм. 4. Науково-технічна і довідкова література.

4. Зміст розрахунково-пояснювальної записки:

1. Інформаційні системи та технології для управління перевезеннями.
2. Проектування веб-додатку для вантажних автоперевезень.
3. Розробка веб-додатку для планування і відслідковування автотранспортних процесів.
4. Практичне застосування веб-додатку.
5. Охорона праці та безпека в надзвичайних ситуаціях.
6. Висновки.
7. Бібліографічний список.
8. Додатки

5. Перелік презентаційного матеріалу : 1 та 2 – Тема, мета, завдання роботи; 3 – Аналіз веб-сайтів, веб-додатків та їх типів; 4 – Аналіз платформ логістики та бірж перевезень; 5 – Аналіз диспетчерського програмного забезпечення; 6 – Вибір інструментів для розробки; 7 – Функціональне моделювання основних процесів веб-додатку; 8 – Архітектура веб-додатку та структура папок; 9 – Опис атрибутів та розробка бази даних; 10 – Розробка конфігурації системи; 11 – Функціонал та результати використання веб-додатку.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	<i>Луб П.М., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри інженерної механіки</i>		

7. Дата видачі завдання 25.02.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Написання першого розділу та означення головних завдань роботи</i>	<i>25.02.2025 – 01.04.2025</i>	
2.	<i>Виконання другого розділу, опис інструментів та формування головних показників для розробки</i>	<i>01.04.2025 – 01.05.2025</i>	
3.	<i>Виконання третього розділу, реалізація мети роботи</i>	<i>01.04.2025 – 01.05.2025</i>	
4.	<i>Виконання четвертого розділу та представлення результатів розробки</i>	<i>01.04.2025 – 01.05.2025</i>	
4.	<i>Написання розділу: «Охорона праці та безпека в надзвичайних ситуаціях»</i>	<i>01.05.2025 – 01.06.2025</i>	
5.	<i>Завершення оформлення розрахунково-пояснювальної записки та презентаційного матеріалу</i>	<i>01.05.2025 – 01.06.2025</i>	
6.	<i>Завершення роботи в цілому</i>	<i>01.06.2025 – 09.06.2025</i>	

Студент _____ Герега А.Б.
(підпис)

Керівник роботи _____ Луб П.М.
(підпис)

УДК: 004.42 + 629.4

Проектування веб-додатку для планування і відслідковування автотранспортних процесів. Герєга А.Б. Кафедра ІТ. – Дубляни, Львівський НУВМБ, 2025.

Кваліфікаційна робота: 62 с. текст. част., 38 рис., 5 табл., 11 слайдів, 25 джерел.

У роботі розглядаються основні типи веб-додатків та принципи їхньої роботи, зокрема у контексті логістичних платформ і бірж перевезень вантажу. Проведено аналіз сучасного диспетчерського програмного забезпечення, що використовується в автотранспортних компаніях, з акцентом на їхню функціональність, переваги та недоліки.

Описано процес планування робіт зі створення веб-сервісу для логістики, вибір відповідних інструментів для розробки та побудову функціонального моделювання основних бізнес-процесів. Застосовується модель розробки додатків MVC, що дозволяє чітко розділити логіку, інтерфейс та дані.

Розкрито атрибути бази даних, побудову її загальної схеми та конфігурації системи. Наведено маршрутизацію потоків у межах додатку, логіку обробки запитів і відповідей веб-сервера, а також взаємодію між програмними модулями.

Розроблено додаток та продемонстровано його основний функціонал, результати та ефективність використання у сфері вантажоперевезень.

Розроблено заходи з охорони праці та безпека в надзвичайних ситуаціях.

Ключові слова: веб-додаток, проектування, архітектура, автотранспортні процеси.

Keywords: web application, cargo transportation, planning, architecture, databases, application core, modules, browser.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1.	
ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ ДЛЯ УПРАВЛІННЯ ПЕРЕВЕЗЕННЯМИ.....	7
1.1. Типи веб-додатків та принцип їх роботи.....	7
1.2. Аналіз логістичних платформ та бірж перевезень вантажу.....	11
1.3. Аналіз диспетчерського програмного забезпечення автотранспортних компаній.....	16
РОЗДІЛ 2.	
ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ ВАНТАЖНИХ АВТОПЕРЕВЕЗЕНЬ.....	22
2.1. Планування робіт із створення ІТ-сервісу.....	22
2.2. Вибір інструментів для розробки веб-додатку.....	23
2.3. Функціональне моделювання основних процесів веб-додатку...	27
2.4. Модель розробки додатків MCV.....	29
РОЗДІЛ 3.	
РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ І ВІДСЛІДКОВУВАННЯ АВТОТРАНСПОРТНИХ ПРОЦЕСІВ.....	33
3.1. Опис атрибутів бази даних.....	33
3.2. Розробка загальної схеми бази даних.....	34
3.3. Розробка загальної конфігурації системи.....	36
3.4. Маршрутизація потоків додатку.....	38
3.5. Логіка обробки запитів та відповідей веб-сервера.....	39
РОЗДІЛ 4.	
ПРАКТИЧНЕ ЗАСТОСУВАННЯ ВЕБ-ДОДАТКУ.....	41
4.1. Поєднання та взаємодія програмних модулів.....	41
4.2. Функціонал та результати використання веб-додатку.....	43
РОЗДІЛ 5.	
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ...	48
5.1. Структурно-функціональний аналіз технологічного процесу...	48
5.2. Розрахунок освітлення приміщення комп'ютерного кабінету...	49
5.3. Безпека в надзвичайних ситуаціях.....	52
ВИСНОВКИ І РЕКОМЕНДАЦІЇ.....	53
БІБЛОГРАФІЧНИЙ СПИСОК.....	55
ДОДАТКИ.....	57

ВСТУП

Розвиток інформаційних технологій (ІТ) зумовив впровадження систем, онлайн-сервісів та веб-додатків, які покращують комунікації між виробниками та замовниками, партнерами та сприяють налаштуванню зв'язків у бізнес процесах. В сфері вантажних перевезень створюються такі додатки та системи, які частину інформації переводять в електронний вигляд та опрацьовують дані завдяки відповідним алгоритмам, спеціалізованим інтерфейсам та базам даних. Використання інтуїтивно простого інтерфейсу усуває пряму залежність людини від суб'єктивізму, помилок водія, менеджера, їх взаємозалежності. Водночас, аналітики-менеджери мають можливість завчасно обробляти дані, зберігати їх в базах даних, з яких потім водій отримує інструкції та формує щоденні завдання, виконує прямі робочі обов'язки, відходячи від обміну даними зі своїм менеджером через відповідні ІТ інструменти.

Сьогодні, запит учасників ринку спрямований на використання інструментів які пришвидшують бізнес-процеси та усувають незаплановані витрати. Те ж стосується і логістичних сервісів. За цими словами стоять цілком конкретні речі: електронний документообіг, електронний кабінет логіста, Іт-сервіси планування і моніторингу перевезень та інші зміни логістичних бізнес-процесів тощо.

Мета роботи – проектування веб-додатку для планування і відслідковування вантажних перевезень із підтримкою обробки запитів та відповідей веб-сервера.

Завдання кваліфікаційної роботи:

- аналіз інформаційних систем та технології для управління перевезеннями;
- спроектувати веб-додаток для планування і відслідковування автотранспортних процесів;
- показати практичність розробленого веб-додатку.

РОЗДІЛ 1.

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ ДЛЯ УПРАВЛІННЯ ПЕРЕВЕЗЕННЯМИ

1.1. Типи веб-додатків та принцип їх роботи

Кількість підприємств, що використовують Інтернет як канал зв'язку із клієнтами та працівниками зростає щороку. Всесвітня мережа дозволяє обмінюватись інформацією зі своєю цільовою аудиторією, проводити безпечні транзакції. Проте ефективна взаємодія можлива лише тоді, коли бізнес зможе швидко збирати та обробляти дані про клієнтів, відповідати на запити тощо. Звичайні сайти нездатні впоратися з усіма завданнями бізнесу, тому на допомогу приходять веб-додатки.

Веб-додатки – це інтерактивні застосунки, що складаються з двох частин:

1) завантажується у браузер, коли користувач набирає відповідну команду.

2) розміщується на веб-сервері, що містить усі дані, закладені у додатку. Він отримує запит від користувача, знаходить інформацію та відправляє у браузер.

Для більшості клієнтів веб-додатки ззовні нічим не відрізняються від сайтів. Але якщо останні носять більш інформативний характер, веб-додатки мають велику кількість вбудованих функцій – від роботи з графікою і таблицями, до оформлення замовлень онлайн тощо.

Наведемо переваги web-додатків:

- ✓ веб-додатки не потребують інсталяції у пам'ять пристрою;
- ✓ користувач може не думати про оновлення програми, оскільки воно відбувається автоматично й централізовано;
- ✓ усі користувачі отримують доступ до однієї й тієї ж версії вебзастосунку, що дозволяє уникнути непорозумінь;

- ✓ веб-додаток розробляється з урахуванням того, щоб його можна було відкрити з будь-якого браузера та ОС;
- ✓ веб-додатки зменшують витрати як для бізнесу, так і для кінцевого користувача, оскільки потребують менше ресурсів (робочої сили, обладнання) для роботи й обслуговування;
- ✓ застосунки прості у користуванні, не потребують спеціальних знань та вмінь, що дозволяє працювати з різною аудиторією;
- ✓ веб-додатки рентабельні, оскільки дешевші за класичні програмні рішення, що дозволяє їх використовувати компаніям з обмеженим бюджетом.

Веб-додаток можна легко підлаштувати під потреби будь-якого бізнесу.

Водночас, для роботи з веб-сайтом як і з веб-додатком використовують браузер, але ці два інструменти мають суттєві відмінності (табл. 1.1.):

Веб-сайт – це певна кількість веб-сторінок, переважно інформаційного змісту. Щоб ними користуватися, авторизація необов’язкова, проте взаємодія з ними значно обмежена. Споживач може лише скористатися пошуком, опублікувати відгук, підписатися на новини, надіслати листа власнику сайту.

Таблиця 1.1. – Основні відмінності веб-сайту від веб-додатку

№ п/п	Назва показника	Веб-додаток (Інтерактивний додаток)	Веб-сайт
1	Мета створення	Взаємодія з кінцевим споживачем	Донести до споживача інформацію
2	Доступ	Має певні обмеження (може знадобитись авторизація)	Мають доступ усі відвідувачі
3	Взаємодія з користувачем	Юзер може не лише ознайомитись з інформацією, але й зробити певні маніпуляції (наприклад, працювати з текстом)	Користувач може лише ознайомитись з контентом, лишити відгук
4	Структура	Веб-додаток має складнішу структуру у порівнянні з сайтом	
5	Компіляція	Потрібна. Крім того, якщо у веб-додаток будуть внесені зміни, компіляцію потрібно буде зробити ще раз	Не потрібна, навіть якщо згодом сайт зазнає змін

Веб-додаток – містить різні функції, залежно від потреб компанії. Щоб використовувати його можливості у повному обсязі, зазвичай потрібно

обов'язково пройти авторизацію. Система буде виокремлювати користувача від інших клієнтів, зберігати особисті дані тощо.

Для запуску веб-додатків потрібні браузер та веб-сервер. Додаток працює за такою схемою:

- Користувач надсилає запит до веб-серверу, що складається з центру керування застосунком (логіки) та бази даних.
- Центр керування отримує запит й скеровує його до бази даних.
- Інформація витягується з бази даних й передається у центр керування.
- Вебсервер відправляє клієнту інформацію – й вона з'являється на дисплеї користувача.

Виділяють три види веб-додатків:

- односторінковий додаток (SPA);
- багатосторінковий веб-додаток (MPA).
- PWA (прогресивний веб-застосунок).

Додатки SPA (*Single Page Application*) – види веб-додатків для бізнесу, що завантажують лише одну HTML-сторінку. Вона оновлюється автоматично за допомогою JavaScript, тому нема потреби перезавантажувати сторінку. Так, наприклад, відбувається, коли приходить лист на пошту Gmail. Сама сторінка не оновлюється, потрібний елемент (лист) просто підвантажується. Процес відбувається швидко, без затримок.

Таким чином, SPA: 1) забезпечує високу швидкість, оскільки усі елементи завантажуються з першого разу, й під час виконання різних дій на сторінці дані просто змінюються; 2) майже уся робота відбувається у браузері, без звернень до сервера, що дозволяє значно зекономити час; 3) дані можна швидко закешувати – застосунок відправляє лише один запит, збирає дані, після чого може працювати офлайн.

Серед мінусів: 1) необхідність постійного інтернет-з'єднання, без якого використовувати застосунок майже неможливо; 2) додаткове навантаження на браузер. Якщо сторінка виявитися важкою, старі та слабкі пристрої можуть “зависнути”; 3) браузер обов'язково має підтримувати JavaScript; 4) можуть

виникнути проблеми з SEO-оптимізацією, оскільки пошукові системи не зможуть проіндексувати усі модулі.

Веб-додаток МРА (*Multi Page Application*) – це багатосторінкові веб-додатки, що працюють за класичною схемою, тобто при постійній взаємодії з веб-сервером. Будь-яка зміна провокує оновлення сторінки. Такі додатки важчі за SPA, тому довше відкриваються.

Плюси веб-додатку МРА: 1) ідеально підходить для інтернет-магазинів з великими каталогами; 2) сторінки можна легко оптимізувати під ключові запити; 3) застосунок можна використовувати у браузерях, що не підтримує JavaScript.

До недоліків відносять: 1) МРА складніші, дорожчі та масштабніші за SPA; 2) розробка веб-додатку МРА потребує більше часу, а відповідно і коштує дорожче.

PWA (*Progressive Web App*) можна назвати гібридом веб-сторінки та мобільного додатка. Йдеться про можливість відкрити веб-додаток через браузер на десктопі чи смартфоні. Також його можна встановити на робочий екран гаджета, що дозволяє працювати автономно, без доступу до мережі. При цьому додаток PWA зберігає чимало функцій, типових для нативного мобільного додатку. Візуально макет веб-додатку майже не відрізняється від мобайлу.

Переваги PWA: 1) можна працювати одразу з кількома ОС, на смартфоні, ПК, терміналі у торговому залі; 2) висока швидкість встановлення й роботи; 3) швидка розробка – для створення PWA не треба робити окремий сайт, достатньо додати новий модуль для вже наявного; 4) оновлення відбувається автоматично; 4) для встановлення не потрібно звертатись в магазин додатків.

Недоліки PWA: 1) Firefox, Edge та деякі інші браузери погано працюють з цією технологією; 2) через обмеження iOS розробникам PWA зазвичай доступні не всі нативні функції пристроїв від Apple.

1.2. Аналіз логістичних платформ та бірж перевезень вантажу

Пришвидшення вирішення завдань із співпраці замовників й перевізників відбувається завдяки інформаційним логістичним платформам (онлайнві біржі перевезень вантажів ОБП), які сьогодні у режимі онлайн дозволять знаходити транспорт для міждержавних перевезень, спрощують пошук замовників тощо завдяки використанню спеціалізованих веб-сайтів.

Сучасні технології дють можливість спостерігати за транспортуванням вантажів в реальному часі, зокрема, через Інтернет за допомогою маячків GPSTрекерів, або софт-трекерів тощо. Це в свою чергу є гарантією того, що перевізник буде дотримуватися узгодженого маршруту.

Проаналізуємо онлайн-біржі вантажних перевезень Східної Європи, розглянемо наступні логістичні платформи:

1. Novoferm. (Польща).
2. BURZA NÁKLADŮ 24. (Чехія).
3. Cargopedia. (Болгарія).
4. BursaTransport. (Румунія).
5. MyDello. (Естонія).

Novoferm. (Польща). Сайт ознайомлює із загальною інформацією щодо перевезень в середині країни та відстежує останні новини пов'язані із транспортною логістикою як у Польщі, так і Європейському союзі. На сайті є опції ознайомлення з вантажем, який в дорозі, зі складськими інвестиціями і транзакціями, а також дає можливість проаналізувати інформацію щодо ланцюгів постачання.

BURZA NÁKLADŮ 24 (Чехія) – це онлайн-платформа, яка пропонує ефективні рішення з транспортування й об'єднує пропозиції та потреби транспортних компаній, постачальників, виробників, які потребують вантажу чи транспорту. Біржа дає можливість всім користувачам знайти доступні та актуальний транспорт чи вантажі в базі даних. Деякі транспортні компанії

пропонують безкоштовний транспорт на зворотний маршрут чи на нові маршрути.

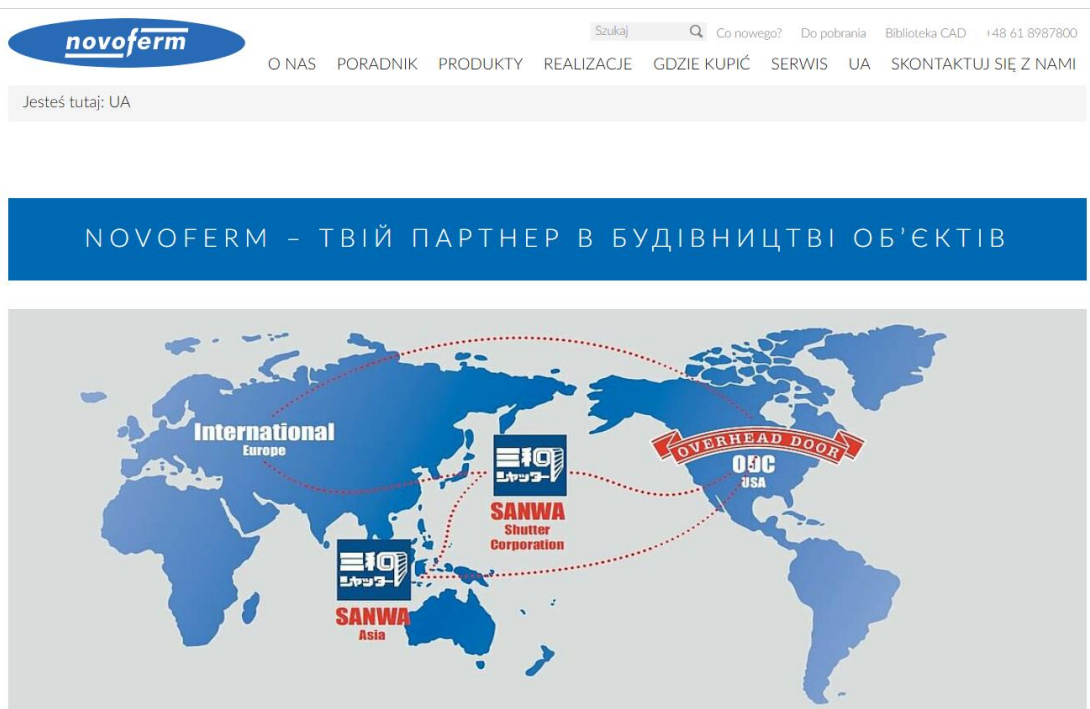


Рисунок 1.1. – Сайт Novoferm для транспортної логістики в Польщі

Cargopedia (Болгарія). Міжнародна біржа вантажів, яка доступна більш ніж 10 мовами, і надає вільний доступ всім користувачам, яким потрібний вантаж чи транспорт. З метою залучення нових клієнтів реєстрація та розміщення оголошень на сайті здійснюється на безкоштовній основі з доступом до всіх основних функцій сайту.

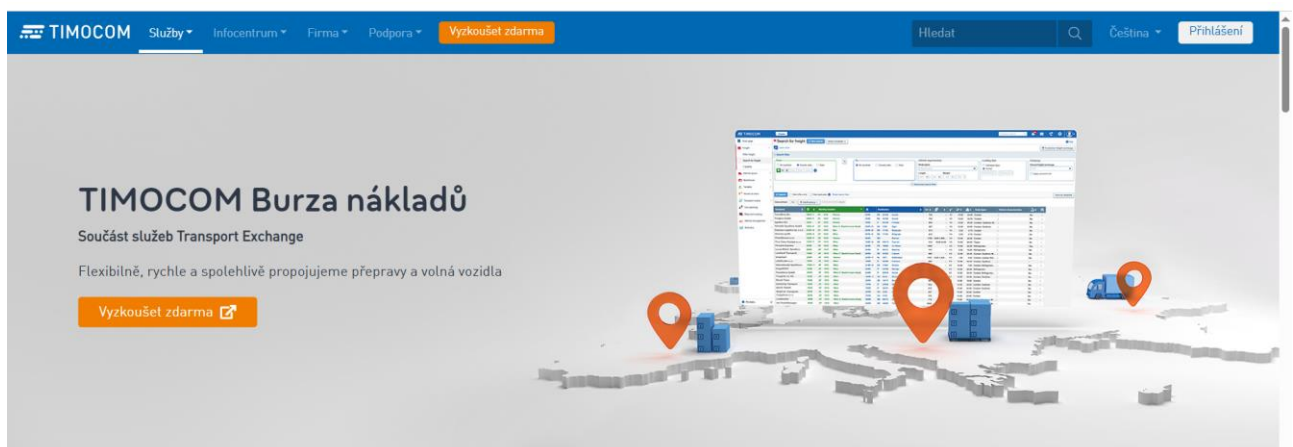


Рисунок 1.2. – Онлайн-платформа для взаємодії транспортних компаній в Чехії

BursaTransport (Румунія). Це Інтернет-платформа, яка призначена для операторів автомобільних вантажних перевезень (перевізників) і виробників, які реалізують транспортні послуги і не бажають користуватися послугами експедиторів. На цей час клієнтська база нараховує понад 16000 клієнтів і постійно розширюється в Центральній і Південно-Східній Європі.

MyDello (Естонія) – і одним із найзручніших рішень щодо здійснення організації перевезення вантажів у короткострокові терміни, починаючи з прозорого та зручного вибору транспорту онлайн, замість незліченних електронних листів та телефонних дзвінків.

Щодо то їх є також достатня кількість. Проаналізуємо також найвідоміші веб-сайти, через які реалізують транспортні послуги в Україні.

Зокрема, платформа Lardi-Trans – це набір інструментів для професіоналів у сфері вантажоперевезень. Пошук вантажу та транспорту, страхування, керування автопарком, електронний документообіг та проведення тендерів (рис. 1.3).

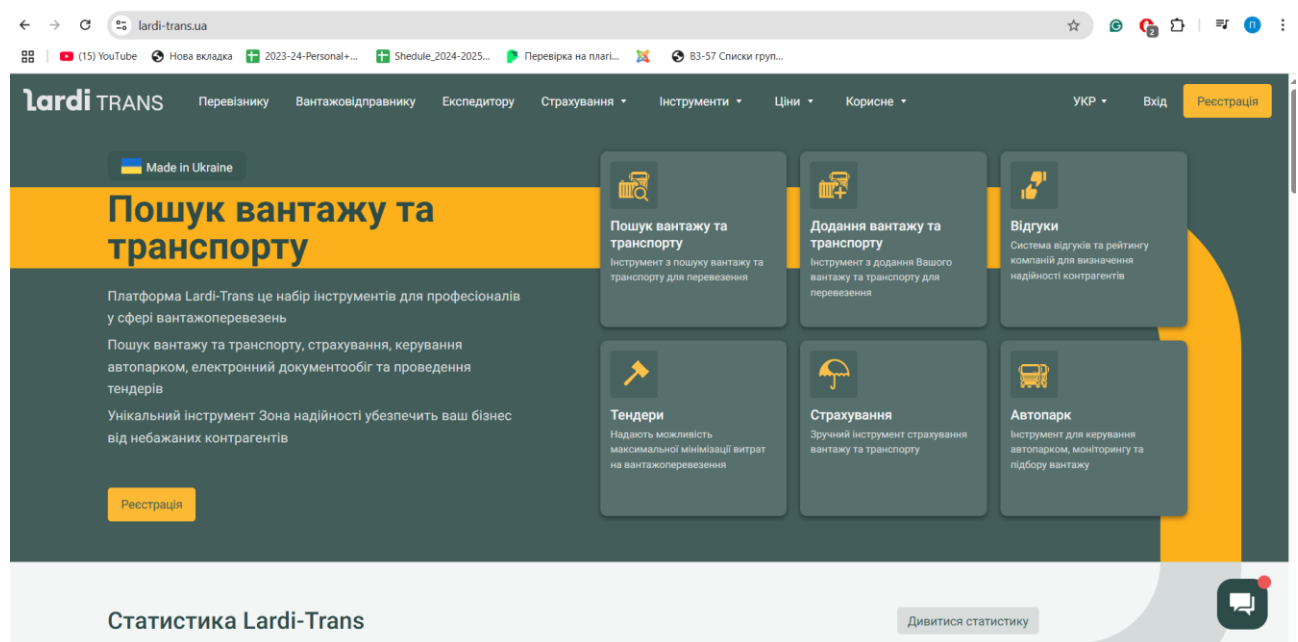


Рисунок 1.3. – Веб-сайт для планування вантажних перевезень Lardi-trans [6]

Lardi-trans використовує сучасний інструмент пошуку вантажу, широкий вибір параметрів дозволяє відбрати найвигідніші пропозиції. Функція

Автопошуку не пропускає вигідні пропозиції перевезень та надішле повідомлення до браузера, або Telegram. Інтеграція з Telegram дозволяє оперативно отримувати сповіщення на будь-які пристрої.

«Ukrainian Logistic» – одна з кращих транспортних компаній згідно із Національним рейтингом якості товарів і послуг «Вибір Країни» 2018, 2019, 2023. Ефективно витрачайте свій час, організацію перевезення делегуйте на відповідальних професіоналів. Мотивація компанії спрямована на «Гру в довгу» з клієнтами та партнерами, фокус діяльності спрямований на три напрямки - ціна, відносини, довіра.

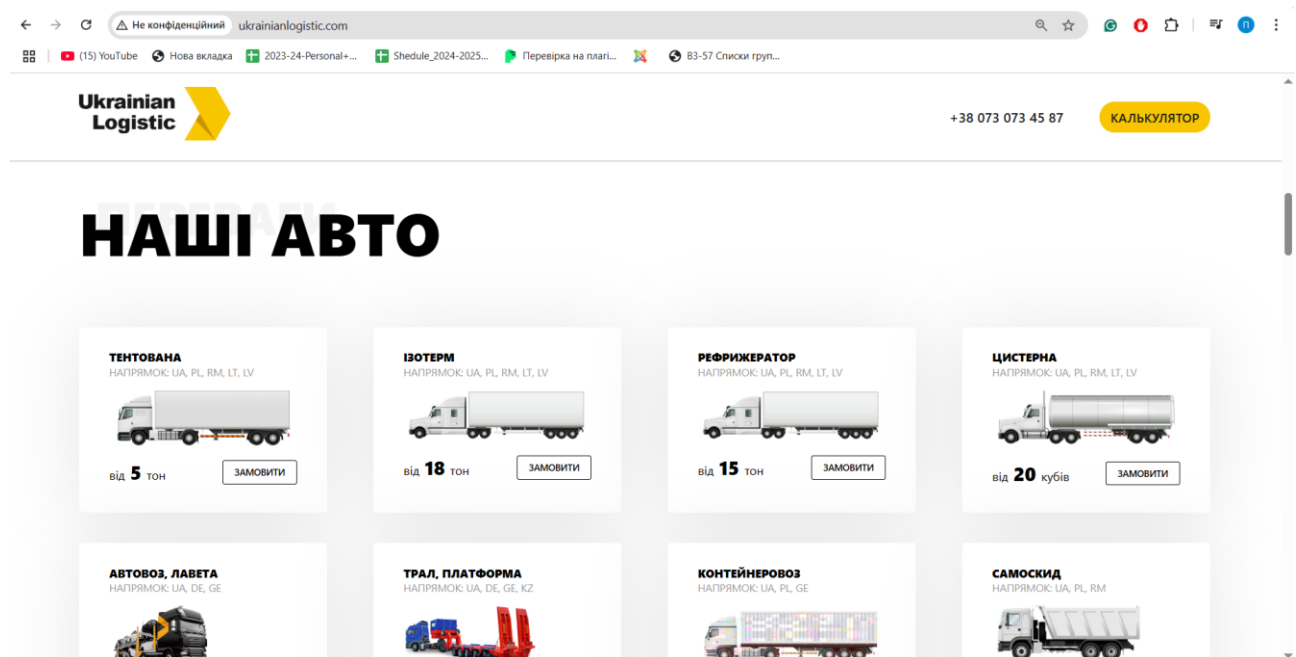


Рисунок 1.4. – Головне вікно сервісу Ukrainian Logistic [2]

Алгоритм роботи із замовниками перевезень наступний: 1. Підбір авто яке стане на рейс. 2. Підписання договорів і страхування вантажу. 3. Авто їде на завантаження. 4. Під час перевезення надається інформація, де їде і коли буде авто. 5. Вивантаження, передача оригіналів документів. 6. Вантажники за необхідності. 7. Акт виконаних робіт.

Компанія DELLA™ вантажні перевезення успішно працює на ринку вантажних автоперевезень з 1995 р. До послуг перевізників – 20-річний досвід організації внутрішніх та міжнародних вантажних перевезень. Замовнику

пропонується необхідна кількість пропозицій із перевезення вантажу від різних перевізників, та щоразу обирається тільки найбільш вигідний варіант.

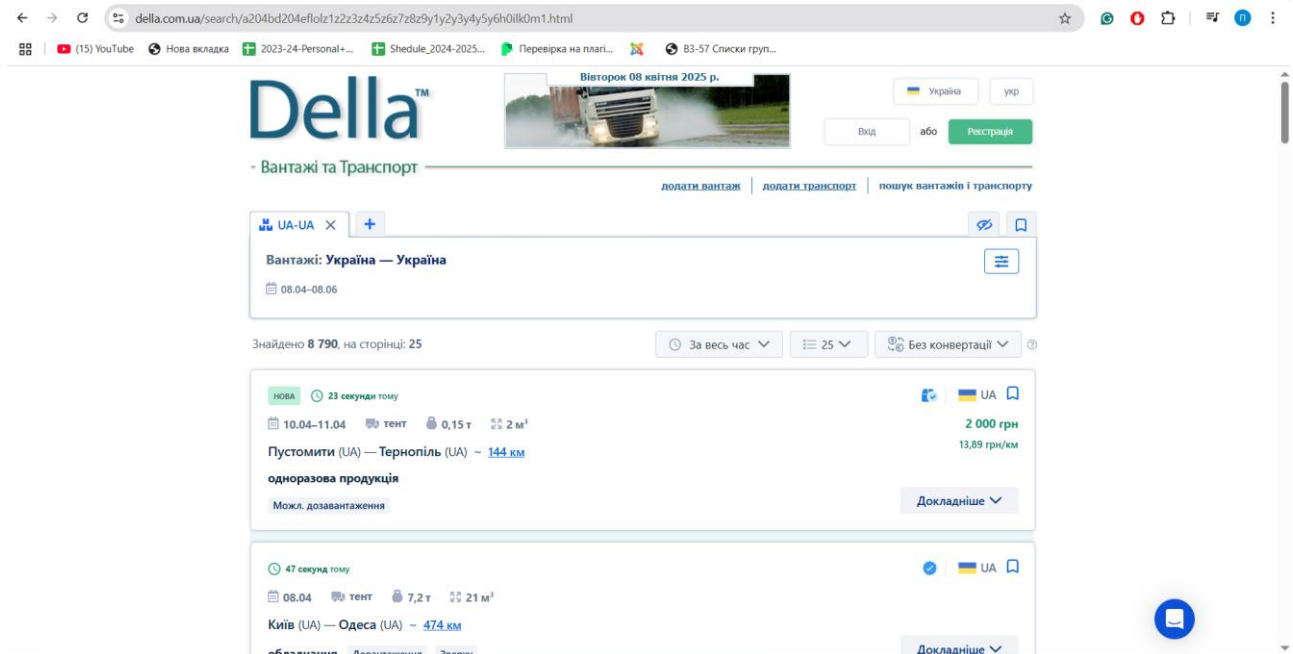


Рисунок 1.5. – Головне вікно веб-сторінки Компанії DELLA™ [1]

Всі перевізники DELLA (компанії, що професійно займаються вантажоперевезеннями) проходять обов'язкову реєстрацію, надаючи державні документи на свій вид діяльності, наприклад, ліцензію на міжнародні автоперевезення або ліцензію на внутрішні перевезення. Веб-сервіс дає можливість знайти перевізника на сайті в реальному часі, надається інформація про прямих Замовників перевезень вантажів та замовлення на вантажні перевезення тощо.

Grand Logistics Company - працює на ринку транспортно-експедиторських послуг України з 2007 року і надає комплекс послуг в сфері вантажоперевезень. Веб-сервіс забезпечує швидкий розрахунок вартості перевезення, оперативну доставку вантажів, індивідуальний підхід та гнучке ціноутворення, якісне митно-брокерське обслуговування, вигідні ставки страхування та допомогу у вирішенні складних ситуацій. Компанія спеціалізується на доставці вантажів будь-яких розмірів та всіма видами транспорту Україною, країнами СНД та країнами ЄС.

Таким чином, сьогодні існують різноманістні сервіси що надають послуги із планування вантажних перевезень.

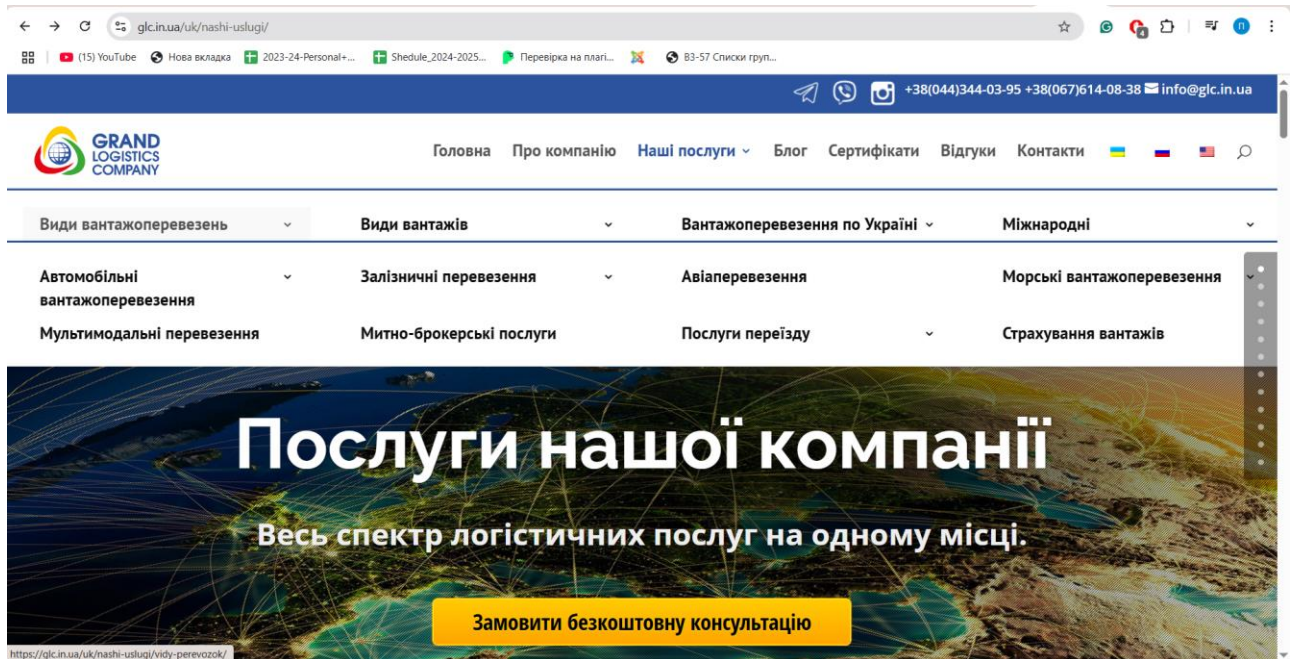


Рисунок 1.6. – Головне вікно веб-сторінки Grand Logistics Company [18]

Їх головні пропозиції скеровані на те щоб підібрати замовника, побудувати маршрут, запропонувати транспортний засіб, забезпечити документальну підтримку, страхування як вантажів так і машини й водія тощо.

1.3. Аналіз диспетчерського програмного забезпечення автотранспортних компаній

Хмарні технології сьогодні дозволяють користувачам завантажувати дані на віддалений сервер і зберігати їх у хмарі. Тоді, доступ до даних буде з будь-якого місця і можна контролювати та управляти своїм автопарком, або вантажем. Це зручно для систем диспетчеризації вантажоперевезень, які поєднують мобільні пристрої у вантажівках і диспетчерське програмне забезпечення в базовому офісі. Програмне забезпечення відстежує кожну вантажівку на маршруті за допомогою GPS-пристроїв, встановлених у

транспортних засобах [14]. Водночас, транспортні засоби можуть отримувати навігацію та коригувати маршрут, щоб водії не заблукали.

Таким чином, програмне забезпечення для диспетчеризації вантажоперевезень скероване на забезпечення бездоганного зв'язку між водіями та диспетчерами, управління маршрутами, створення звітності точною та своєчасною, сприяння швидким фінансовим операціям, відстеження активів, управління автопарком та транспортними засобами тощо.

Проаналізовані нами постачальники найкраще підходять для автотранспортних компаній та вантажних брокерів, більшість з них мають досвід роботи в цій галузі протягом декількох років.

ProTransport обслуговує автотранспортну галузь вже понад 15 років, надаючи універсальне програмне забезпечення "все-в-одному", яке можна налаштувати для автопарків або транспортно-брокерських компаній будь-якого розміру. З ним отримується диспетчеризація, облік, безпека, аналітика та технічне обслуговування, об'єднані в єдине ціле.

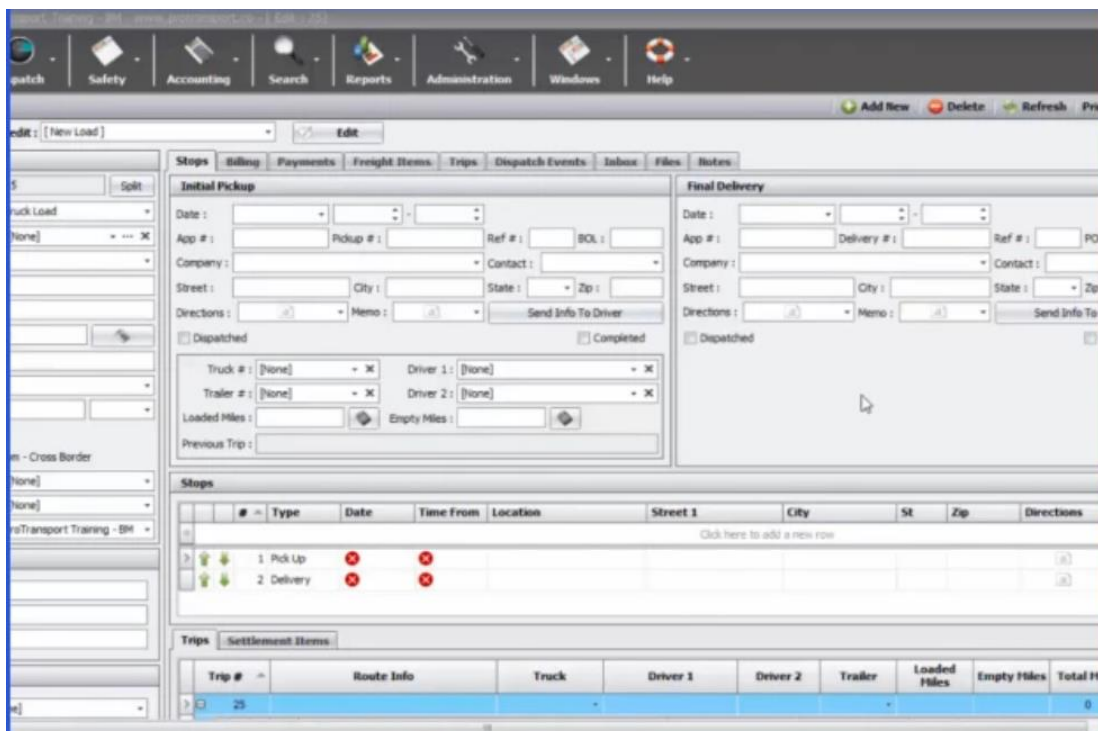


Рисунок 1.7. – Програмне рішення ProTransport для управління автопарком

Рішення ProTransport можна інтегрувати з різними програмами та інструментами, включаючи GPS, QuickBooks, постачальників паливних карток, IFTA тощо. ProTransport є простим і зручним у використанні завдяки кольоровому інтерфейсу та інтуїтивно зрозумілій інформаційній панелі. Всі операції автопарку об'єднані в єдину систему, що дозволяє швидко вимірювати продуктивність і приймати стратегічні рішення на основі отриманих результатів.

Onfleet – це надійне рішення для міжнародних кур'єрських компаній, роздрібних торговців, дистриб'юторів, стартапів, що займаються доставкою на вимогу, та багатьох інших. Програмне рішення орієнтоване в першу чергу на доставку "останньої милі" і працює вже у понад 90 країнах світу.



Рисунок 1.8. – Програмне рішення Onfleet для міжнародних кур'єрських компаній

Onfleet дозволяє легко маршрутизувати, відправляти, відстежувати та аналізувати доставку, спілкуватися з водіями та надавати клієнтам оновлення в режимі реального часу за допомогою SMS-повідомлень. Більше того, він

пропонує підтвердження доставки разом з інструментами збору відгуків, які допоможуть вам зробити вашу доставку ще кращою.

TruckingOffice – це програмне забезпечення, розроблене власником автопарку для організації виставлення рахунків та обліку власної компанії, було створено з урахуванням усіх нюансів бізнесу вантажоперевезень. Однак воно вийшло за рамки рішення для сімейного бізнесу батька і сина, для якого було створено спочатку.

Орієнтоване в основному на невеликі автотранспортні компанії, програмне забезпечення TruckingOffice тепер полегшує ефективну диспетчеризацію, облік і звітність IFTA, дозволяє менеджерам тримати руку на пульсі технічного обслуговування автопарку і витрат. Завдяки дотриманню графіків, власники та оператори автопарків збільшують дохід на мілью, зменшують кількість придорожніх ремонтів, а також спрощують виставлення рахунків та фінансові питання!

1053	04/03/12	04/06/12	Doe	0101	Best Customer	Canton, OH	San Antonio, TX	zx987654	\$1,830.75	View Edit Clone Delete
1055	04/01/12	04/06/12	Campbell	0102	Customer	New York, NY	Los Angeles, CA		\$4,500.35	View Edit Clone Delete
1054	03/21/12	03/27/12	Campbell	0102	Best Customer	Canton, OH	New York, NY	wx710101	\$1,500.00	View Edit Clone Delete
1052	03/07/12	03/27/12	Doe	0101	Factory	Houston, TX	Houston, TX	326106	\$1,954.66	View Edit Clone Delete
1051	03/12/12	03/15/12	Doe	0101	Best Customer	Canton, OH	San Antonio, TX	zx987654	\$1,830.75	View Edit Clone Delete
1050	02/29/12	03/02/12	Campbell	0102	Ship Yard	Norfolk, VA	Cleveland, OH	131619	\$829.95	View Edit Clone Delete
1047	01/26/12	02/03/12	Campbell	0102	Best Customer	Canton, OH	Richmond, VA	AB123456	\$1,242.74	View Edit Clone Delete
1049	01/28/12	01/28/12	Campbell	0102	EMPTY MOVE	Cleveland, OH	Franklin, VA			View Edit Clone Delete
1046	01/24/12	01/27/12	Doe	0101	Factory	Houston, TX	Cleveland, OH	326106	\$1,954.66	View Edit Clone Delete
1045	01/23/12	01/25/12	Campbell	0102	Ship Yard	Norfolk, VA	Cleveland, OH	131619	\$829.95	View Edit Clone Delete
1043	01/20/12	01/22/12	Campbell	0102	Best Customer	Canton, OH	Richmond, VA	468676	\$1,142.74	View Edit Clone Delete
1044	01/18/12	01/22/12	Doe	0101	Best Customer	Canton, OH	San Antonio, TX	zx987654	\$4,182.42	View Edit Clone Delete
1042	01/10/12	01/14/12	Doe	0101	Factory	Houston, TX	Canton, OH	791346	\$2,340.09	View Edit Clone Delete
1041	01/09/12	01/11/12	Campbell	0102	Broker	Norfolk, VA	Cleveland, OH	134679	\$691.63	View Edit Clone Delete
1039	01/05/12	01/07/12	Campbell	0102	Best Customer	Canton, OH	Richmond, VA	00091730264	\$1,086.25	View Clone
1040	01/03/12	01/07/12	Doe	0101	Best Customer	Canton, OH	San Antonio, TX	95100357	\$3,993.83	View Clone
1038	12/20/11	12/22/11	Campbell	0102	Broker	Norfolk, VA	Cleveland, OH	53719041	\$608.63	View Edit Clone Delete
1036	12/19/11	12/22/11	Doe	0101	Factory	Houston, TX	Cleveland, OH	br95422648	\$1,628.88	View Edit Clone Delete
1037	12/14/11	12/16/11	Campbell	0102	Best Customer	Canton, OH	Richmond, VA	264895301	\$1,303.60	View Edit Clone Delete
1035	12/12/11	12/16/11	Doe	0101	Best Customer	Canton, OH	San Antonio, TX	bx316497	\$4,719.98	View Edit Clone Delete
1034	12/09/11	12/12/11	Campbell	0102	Broker	Norfolk, VA	Cleveland, OH	vb16497	\$960.27	View Clone
1033	12/05/11	12/07/11	Campbell	0102	Best Customer	Canton, OH	Richmond, VA	wx98462	\$1,086.25	View Edit Clone Delete
1032	11/26/11	12/01/11	Doe	0101	Factory	Houston, TX	Cleveland, OH	wx35791	\$2,280.42	View Edit Clone Delete
1031	11/24/11	11/28/11	Campbell	0102	Customer	New York, NY	Cleveland, OH	zx9764	\$866.36	View Edit Clone Delete

Рисунок 1.9. – Програмне рішення TruckingOffice для обліку вантажоперевезень

Axon Trucking Software з 1982 року використовується в галузі вантажоперевезень, налічує понад 16 000 користувачів і 54 мільйони введених даних про вантажі. Рішення найкраще підходить для перевізників вантажівок/LTL, автоперевізників, тваринників, нафтопромисловців та вантажовідправників з великими вантажами.

Axon Trucking можна інтегрувати з програмним забезпеченням для обліку пробігу, маршрутизації та картографування, що дозволяє відстежувати в режимі реального часу, управляти автопарком і паливом, технічним обслуговуванням транспортних засобів та звітувати за допомогою зручного і простого у використанні мобільного додатку.

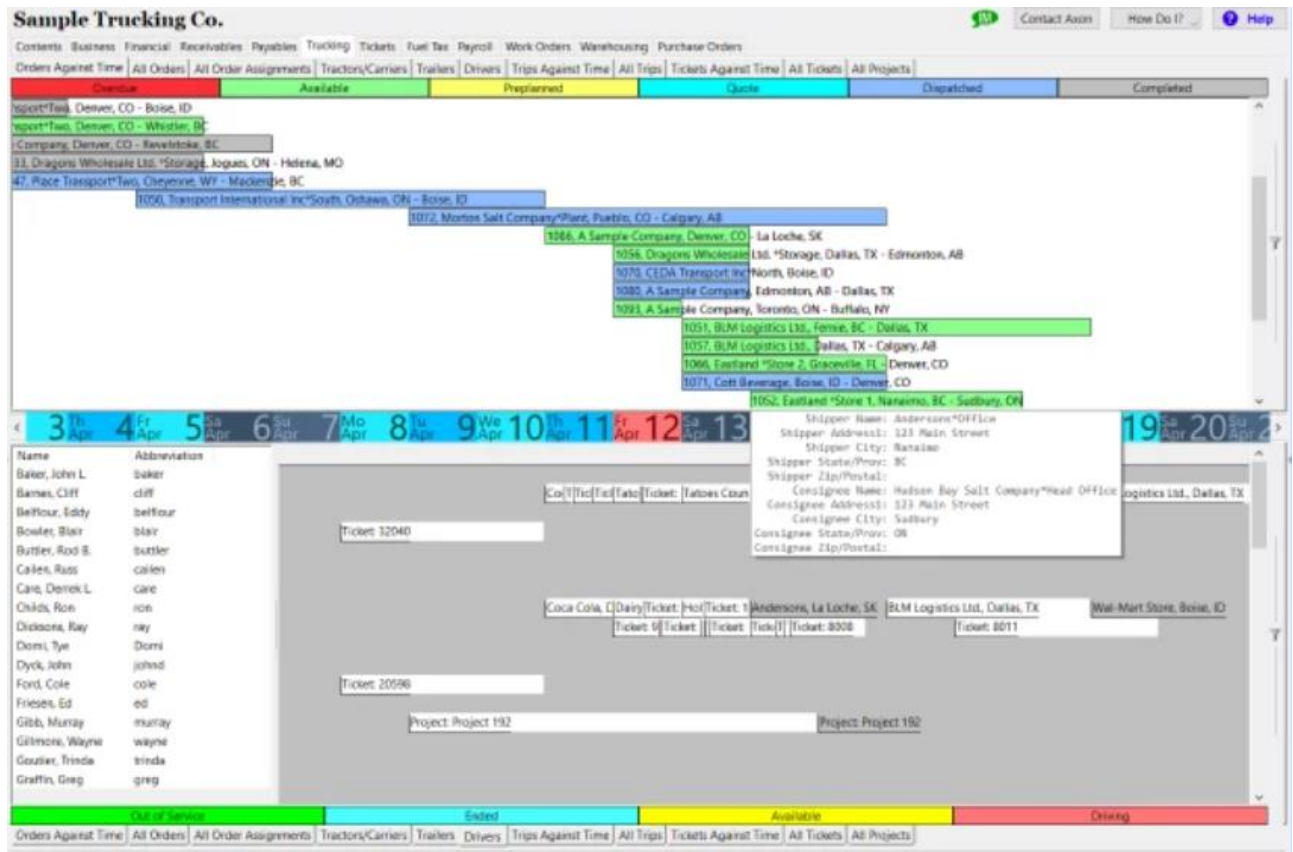


Рисунок 1.10. – Axon Trucking для перевезення великих вантажів

McLeod LoadMaster та PowerBroker з 1985 пропонує диспетчерські рішення для автотранспортних компаній, сторонніх логістичних компаній, компаній-вантажовідправників, перевізників, автопарків. На сьогоднішній день компанія має близько тисячі активних користувачів у транспортній сфері.

Крім того, продукти McLeod можуть бути інтегровані з рішеннями, які допомагають управляти документообігом, іміджевими документами, автоматизувати бізнес-процеси, покращити маркетинг та обслуговування клієнтів, управляти та вимірювати продуктивність автопарку.

The screenshot displays the 'Driver - McLeod Software - Company TMS' window. It features a top toolbar with icons for navigation and data management. Below the toolbar, there are input fields for driver details: Driver code (BUKA), Driver type (Owner), Name (BURN), Address (635 BROADFIELD CIR.), City/State/Zip (DOUGLAS, GA 31535), and Latitude/Longitude. To the right, there are fields for 'Pay to' (MCLT), 'Name' (MCLEOD TRUCKING), and 'Address' (3200 Tenth Street). A 'City/State/Zip' field shows 'Montgomery AL 36117'. Below these fields is a tabbed interface with tabs for General, Safety, Comments (2), Profile, Dispatch, Contacts, and Operations. The 'Performance' tab is active, showing a table with columns: Date, Unsafe, Fatigued, Controlled Substance, Fitness, Vehicle, Cargo, and Crash. The table contains data for various dates, with some cells showing red and green arrows indicating trends. At the bottom, there is a 'Display' button and a status bar showing 'Record 1 of 1 in list'.

Service Failures	Date	Unsafe	Fatigued	Controlled Substance	Fitness	Vehicle	Cargo	Crash
OS&D/Claims	12/31/1969	21	4	0	0.1111	6.5714	0	0
Accidents	12/31/1969	14	4	0	0	19.4	0	0
Scorecard	12/31/1969	14	2.8261	0	0.0435	11.0769	0	0
BASIC	12/31/1969	14	2.8261	0	0.0435	11.07...	0	0
	12/31/1969	14	3.3333	0	0	14.375	0	0
	12/31/1969	14	2.8261	0	0.0435	11.0769	0	0

Рисунок 1.11. – McLeod диспетчерські рішення для автотранспортних компаній

Наскрізне управління вантажними автомобілями або вантажами. Термін "диспетчеризація" не означає лише управління водіями та диспетчерські операції на початку маршруту. Диспетчерські рішення дозволяють отримати уявлення про всю подорож і можливі проблеми на маршруті, більш того, маршрути можуть бути оновлені та оптимізовані на ходу, щоб скоротити час доставки і прибуття. Точні дані для своєчасного прийняття рішень. Завдяки програмному забезпеченню для логістики вантажних перевезень можна створювати резервні копії та зберігати дані в одному місці і мати доступ до них за допомогою будь-якого пристрою з доступом до Інтернету.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ ВАНТАЖНИХ АВТОПЕРЕВЕЗЕНЬ

2.1. Планування робіт із створення IT-сервісу

Як зазначалося вище, метою роботи є розробити проект веб-додатку для планування і відслідковування маршрутів автотранспортних перевезень. Цей веб-додаток спрощуватиме комунікацію у бізнес-процесах між водієм транспортного засобу, та менеджером, який знаходить рейс, бере його в роботу та планує його маршрут тощо [24].

Основним інструментом для планування змісту структури робіт служить WBS (**Work Break Structure**) – представлення проекту, виконане у вигляді ієрархічної структури робіт, що досягається за допомогою послідовної декомпозиції. Інструмент спрямований на детальне планування, оцінку вартості, визначення та розподіл персональної відповідальності виконавців та інші – тобто, на основні роботи і результати, що визначають зміст проекту [10].

Як правило, на верхньому рівні вказується сам проект, під ним (на першому рівні) – основні результати, кожен з яких, в свою чергу, деталізується, тобто наступний рівень завжди менше попереднього за обсягом робіт і, як правило, включає 2 і більше пакетів робіт. При цьому в різних гілках WBS може бути різна кількість рівнів в залежності від потрібного ступеня деталізації. Повна WBS структура наведена на рисунку 2.1.

Після побудови WBS структури необхідно створити OBS – це структура виконавців, тобто хто та за який етап розробки буде відповідати [23]. Список виконавців, котрі братимуть участь у розробці даного проекту формується відповідно до рекомендацій відділу HR. Для оцінки тривалості робіт, з розробки веб-додатку, розподілу задач по часовим проміжкам на основі ієрархічної структури робіт по проекту, та з урахуванням неробочих днів як правило будують діаграму Ганта.

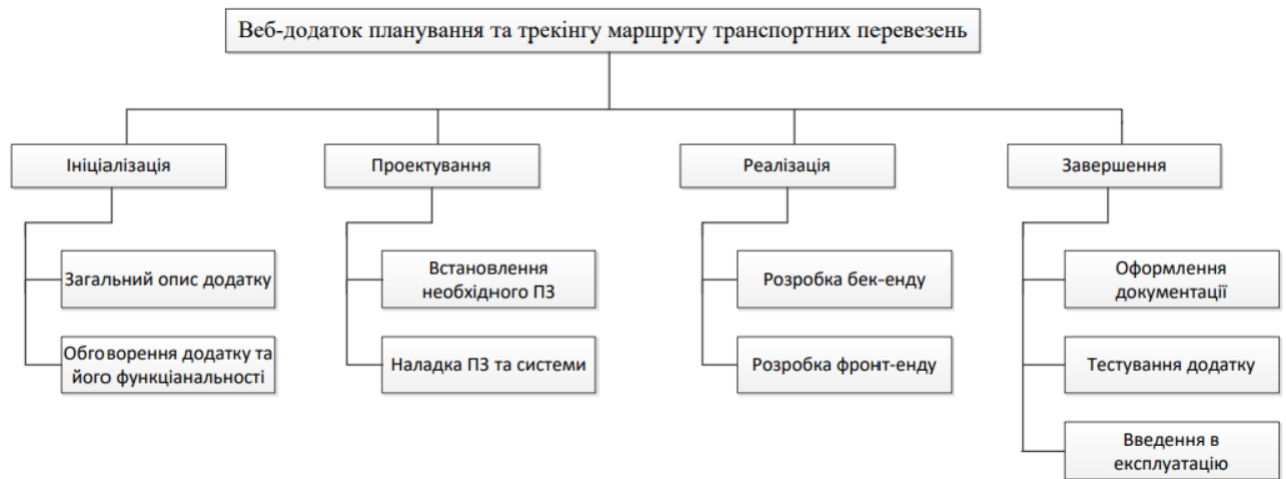


Рисунок 2.1 – WBS структура проекту

Діаграма є планом розподілу робіт згідно календарного розкладу та представляє собою горизонтально розміщенні відрізки на часовій шкалі, де кожен відрізок відповідає своїй окремій задачі та займає певний слот часу в календарі. Сумарна "довжина" діаграми показує загальну тривалість розробки проекту і т.д.

2.2. Вибір інструментів для розробки веб-додатку

Під час розробки, для реалізації додатку використовуються різні програмні засоби. Розглянемо весь перелік програмних засобів які доцільно використати для нашої розробки (рис. 2.2).



Рисунок 2.2 – Базові Інструменти розробки веб-додатків

А саме:

1. Adobe Photoshop для створення макету;

2. Notepad++ для роботи з кодом;
3. OpenServer для розгортки сайту на локальному сервері;
4. PhpMyAdmin для роботи з БД;
5. Фреймворк Laravel для розробки веб-додатку.

Для створення графічних елементів, їх редагування доцільно використовувати Adobe Photoshop, можливостей його достатньо для редагування елементів додатку.

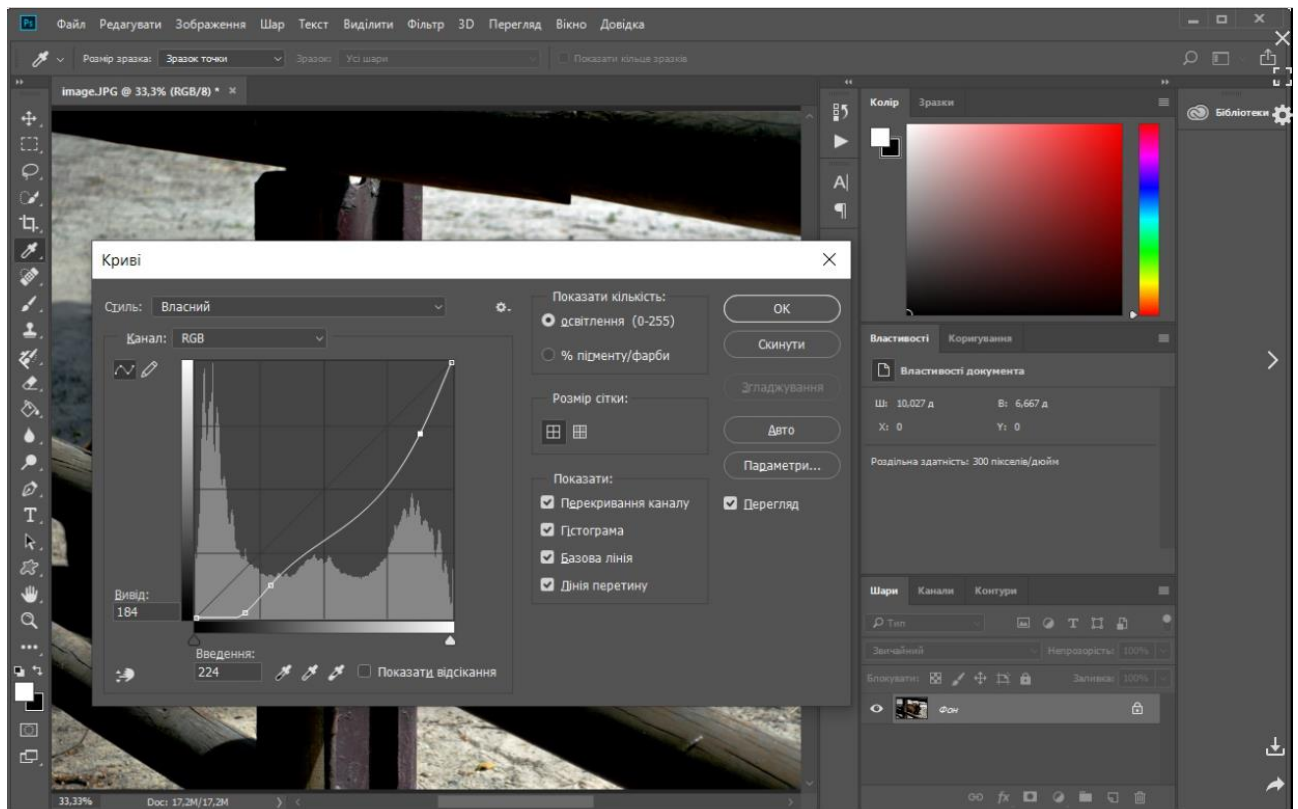


Рисунок 2.3 – Головне вікно Adobe Photoshop

Для роботи із файлами в проєкті доцільно застосувати текстовий редактор Notepad++. Івін оснащений основними функціями та додатковими плагінами необхідних для розробки великих комерційних проєктів (рис. 2.4). Дає змогу працювати з кодом відповідно до потреб користувача [21].

Можливість відтворення додатку на емуляції локального веб-серверу надає програмне забезпечення OpenServer, яке не менш поширене того ж Notepad.

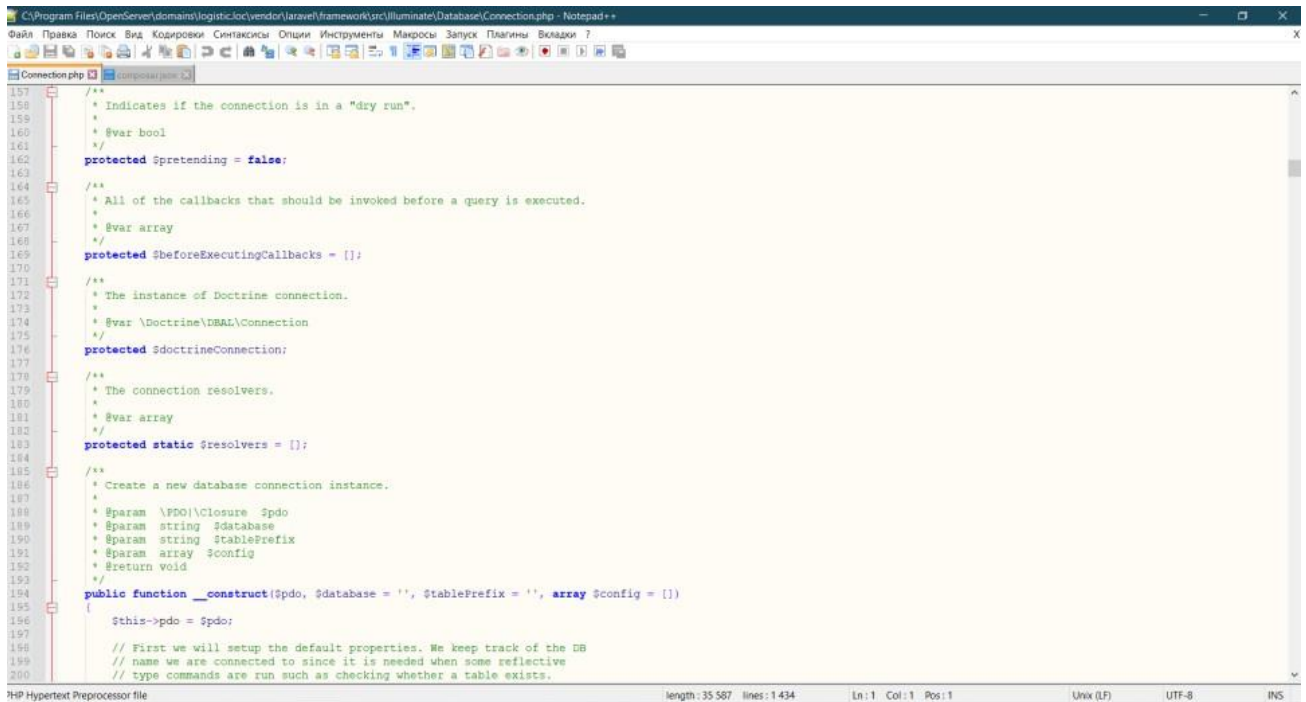


Рисунок 2.4 – Notepad++

OpenServer включає ретельно підібраний набір серверного програмного забезпечення, а також зручну і керуючу утиліту, яка володіє потужними можливостями з адміністрування і налаштування всіх доступних компонентів (рис. 2.5).

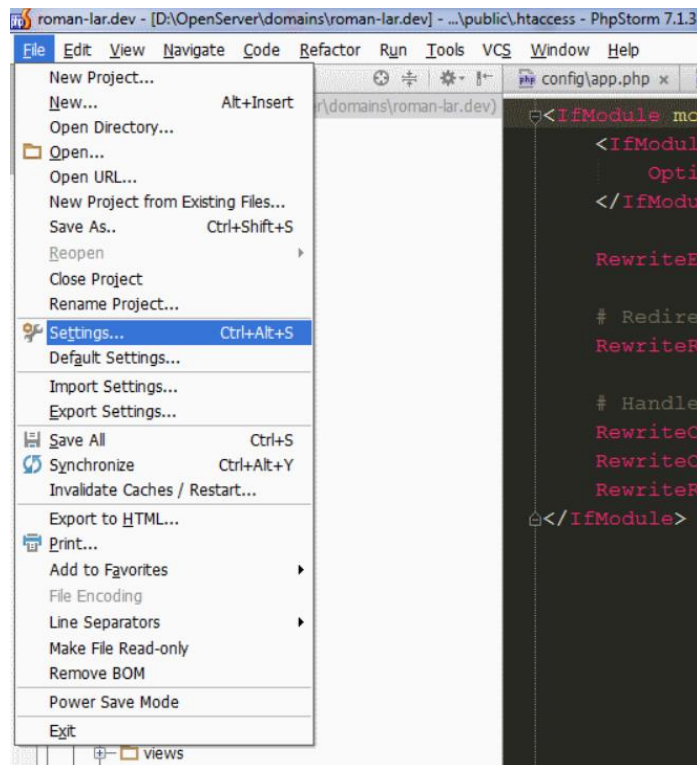


Рисунок 2.5 – Меню налаштування OpenServer

PHP – це популярна мова програмування, особливо серед веб-розробників. Автором початкової версії є Рasmus Лердорф, ідея якого полягала у розробці набору інструментів спрощення процесу створення динамічних веб-сторінок. Незважаючи на те, що сучасний PHP є мовою загального призначення, найчастіше його використовують як серверний інструмент для генерації HTML-коду, який потім інтерпретується веб-браузером.



Фактично це мова програмування з відкритими вихідними кодами, над розвитком якого працюють програмісти-ентузіаста з усього світу. Вона має простий синтаксис, частково схожий на Java та C++. За статистикою, кожен шостий програмний продукт створено на PHP.

Базовим інструментом для розробки доцільно обрати один із найпопулярніших фреймворків для розробки сайтів та веб-додатків [11] (рис. 2.6). Таке рішення прийнято через його поширеність та наявність матеріалів для вивчення, а також він має достатню простий інтерфейс для використання та освоєння. Laravel дозволяє використовувати зручну структуру файлів в проєкті, надає можливість швидкого доступу до них [13].

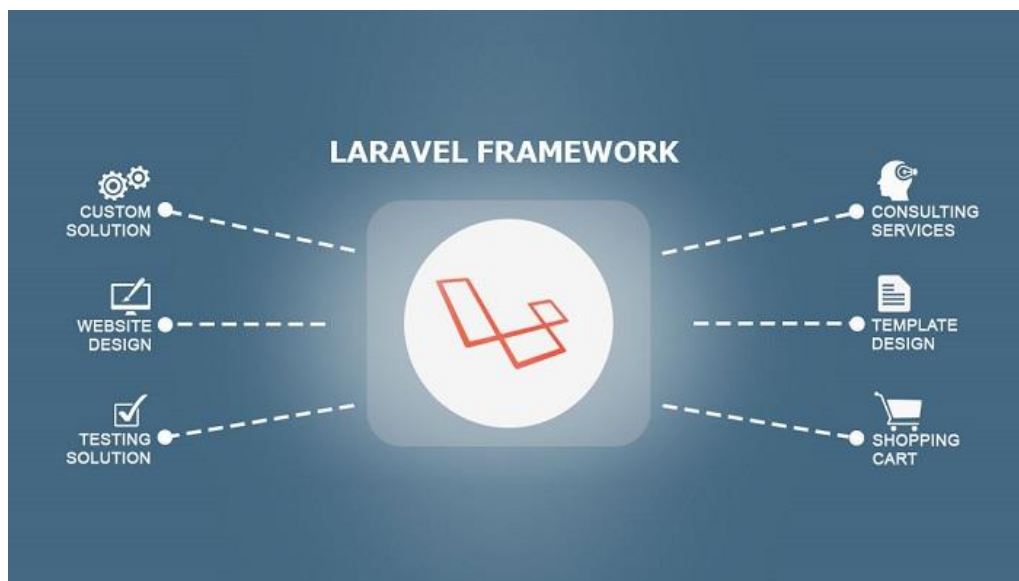


Рисунок 2.6 – Фреймворк Laravel

За допомогою описаних засобів можна реалізувати функціональні вимоги додатку: розробити функціонал, створити макет, локально розгорнути додаток тощо.

2.3. Функціональне моделювання основних процесів веб-додатку

Для опису основних процесів використано методологію функціонального моделювання, яке більш відоме як методологія IDEF0 [19]. Ця методологія використовується, для схематичного представлення всіх інструментів та ресурсів, які використані під час роботи над додатком. Також діаграми побудовані на цій методології можуть містити відображення бізнес- процесів, які вирішуватиме розробляємий продукт.

Контекстна діаграма веб-додатку планування та трекінгу транспортних перевезень та відповідно його декомпозиція, яку було реалізовано в блочному вигляді, для опису роботи створеного веб-додатку, наведено на рисунках 2.7-2.8.

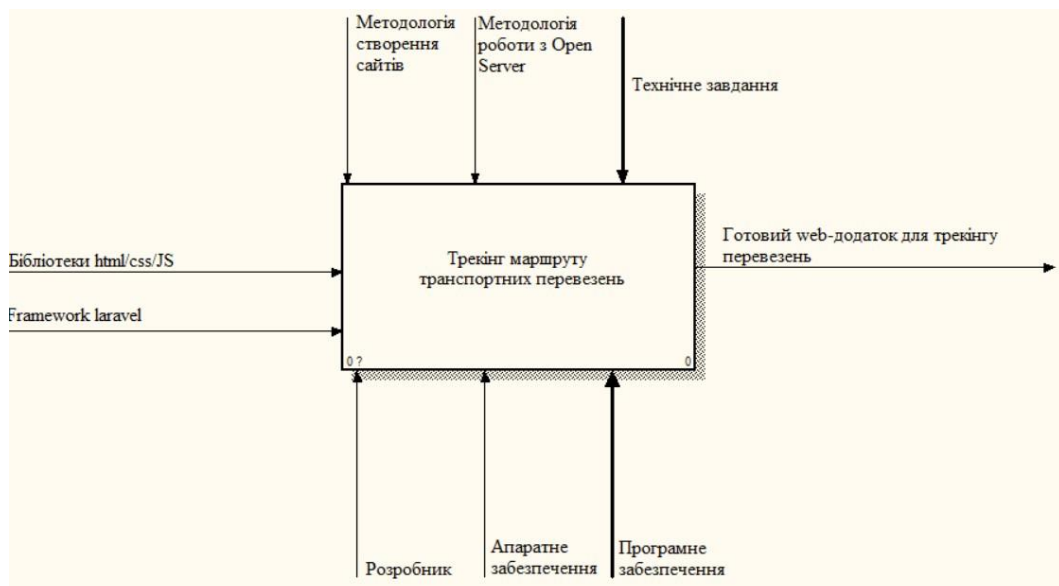


Рисунок 2.7 – Контекстна діаграма веб-сервісу

Декомпозиція контекстної діаграми наглядно відображає основні етапи роботи веб-додатку. Декомпозиція містить не тільки основні блоки, а й

матеріальні та інформаційні ресурси, які позначені стрілками (рис. 2.8).

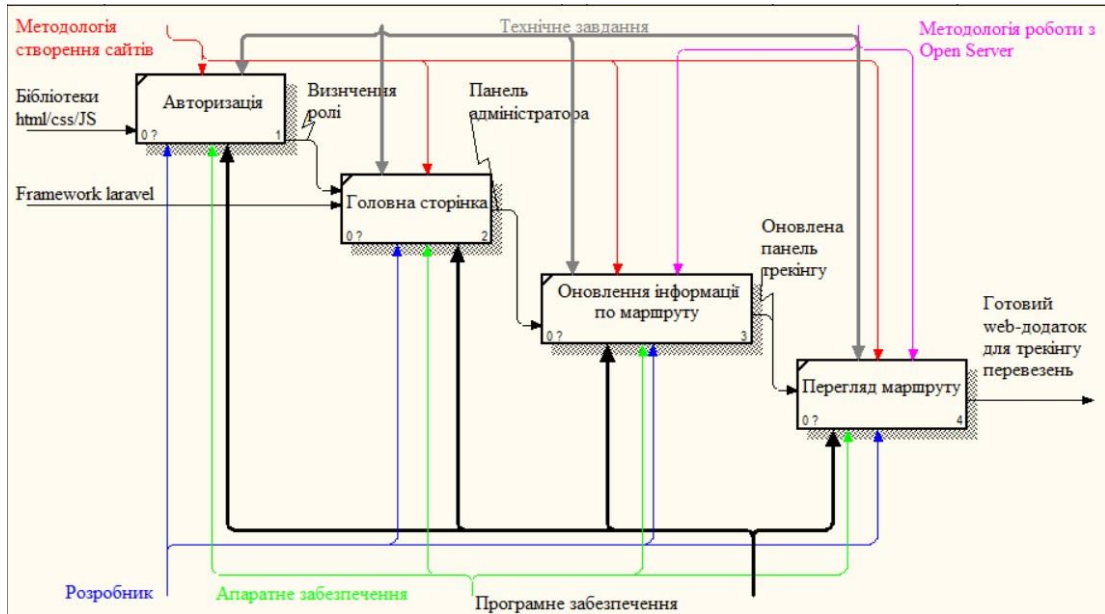


Рисунок 2.8 – Діаграма декомпозиції веб-сервісу

Завершальним етапом проектування веб-додатку є розробка діаграми класів. Така діаграма використовується для опису роботи веб-додатку, відображення користувачів та їх діяльності [19]. Use Case діаграму наведено на рисунку 2.9.

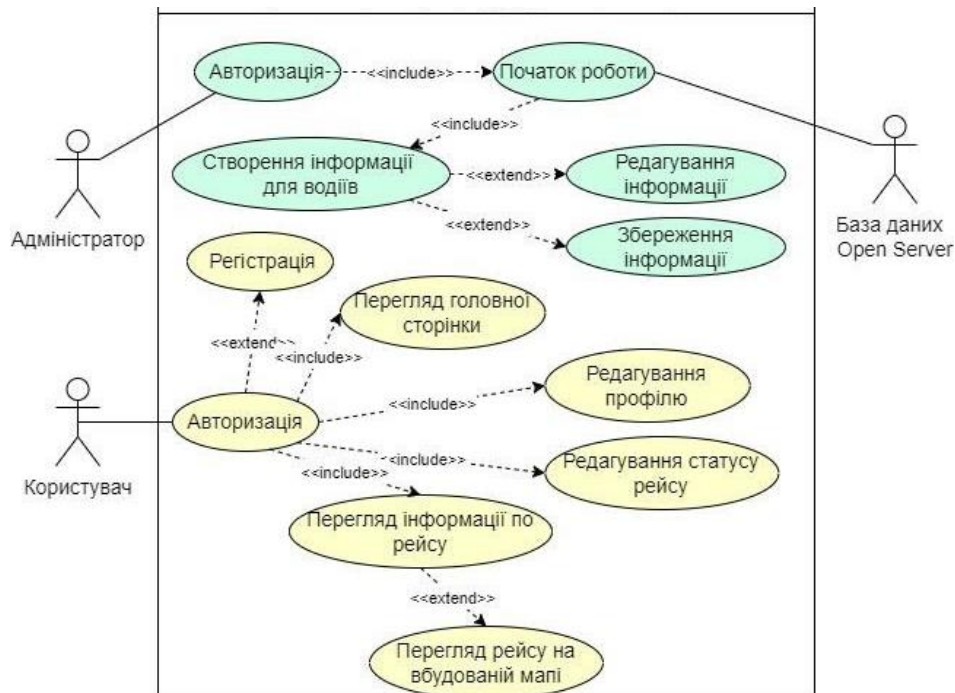


Рисунок 2.9 – Діаграма сценаріїв використання веб-додатку планування маршрутів

В розроблюваному додатку можуть діяти один, або більше користувачів – з правами адміністратора та один, або більше користувач з правами користувача – це залежить від поля відповідальності.

Для початку користувачеві з правами адміністратор доступна буде головна сторінка та панель адміністратора де він може створювати рейси, керувати їми та редагувати, та редагувати дані користувачів з правами користувач. Для користувача з правами користувач доступна головна сторінка та панель водія де він може працювати з рейсами, переглядати на мапі їх маршрут, брати їх в роботу та керувати своїми даними користувача.

2.4. Модель розробки додатків MCV

PHP-фреймворк Laravel є одним із найпоширеніших фреймворків у веб-розробці. Розробники віддають перевагу цьому фреймворку, поступово відходячи від конкуруючих рішень, наприклад Yii, Zend та інших. Laravel поширюється на безкоштовній основі, та має відкритий код. Функціонал Laravel є простим для розуміння та використання. Більшість його функцій, спираючись на загальноприйняті стандарти написання коду, роблять його інтуїтивно зрозумілим, працюють не вимагаючи додаткових налаштувань. Для фреймворку наведено повний пакет документації, яка має регулярні оновлення.

Laravel дає широку можливість створення проектів різної масштабності, потіжність фреймворку надає варіанти вирішення різноманітних задач, незалежно від складності. Та також надає змогу виконувати тестування, зміну функціоналу сайту, або ж його оновлення. Простота інтерфейсу палені адміністрування сприяє легшій роботі.

Для отримання представлення програмної частини та структури сторінок додатку запропоновано його архітектуру (рис. 2.10).

Архітектура додатку основана на базі даних вона складається з реляційної бази даних з підтримкою MySQL, в якій зберігається уся

інформація.

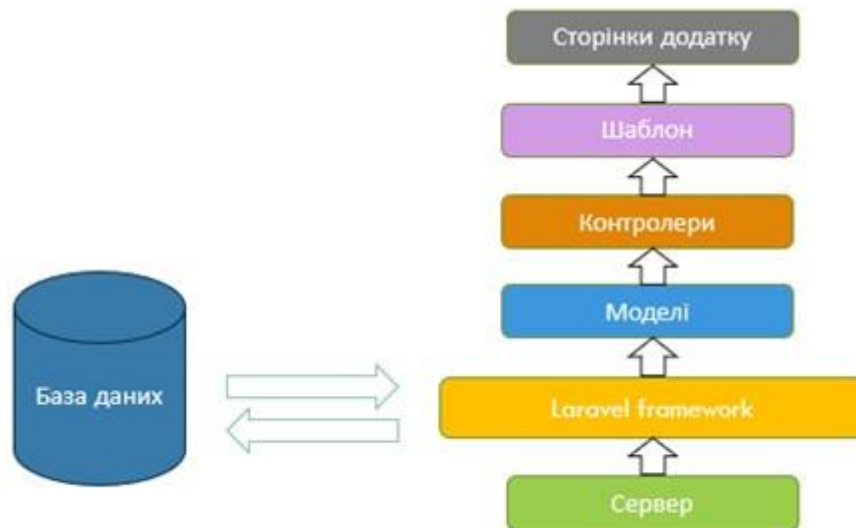


Рисунок 2.10 – Архітектура веб-додатку побудови маршрутів

Для роботи з базами даних фреймворк Laravel використовує моделі даних. На моделі дані передаються на контролери, котрі відповідають за їх обробку. На наступному етапі дані з контролерів відправляються на шаблони. Така послідовність дає змогу відобразити сторінки створеного додатку.

Laravel використовує шаблон проектування Модель-Представлення-Контролер (MVC, Model-View-Controller), який широко застосовується в веб-програмуванні.

MVC – це модель розробки додатків, яка дозволяє легко створювати додатки (рис. 2.11). М – модель, яка представляє собою ядро програми, і дозволяє підключитися до бази даних та виконувати всі завдання CRUD, виконувати перевірки та багато іншого. Але воно не містить ніякої інформації про дизайн та перегляд додатків.

V – це представлення, яке містить конструкції та погляди заявки. Це стосується того, як дані, які надходять від моделі, будуть показуватися користувачам.

C – це контролер, який допомагає підключити модель та переглянути разом. Іншими словами, він використовується для зв'язку між класом "Модель" та "Представлення", так що дані, що надходять з моделі, можуть бути легко

інтегровані в клас перегляду. Користувачі взаємодіють з контролером лише для виконання необхідних операцій.



Рисунок 2.11 – Модель розробки додатків MVC

MVC призначений для поділу бізнес-логіки і призначеного для користувача інтерфейсу, щоб розробники могли легко змінювати окремі частини програми, не зачіпаючи інші. В архітектурі MVC модель надає дані і правила бізнес-логіки, уявлення відповідає за користувацький інтерфейс (наприклад, текст, поля введення), а контролер забезпечує взаємодію між моделлю і представленням.

Структура папок. Структура Laravel-додатку за замовчуванням - відмінна відправна точка як для великих, так і для маленьких додатків. Але, звичайно, можна вільно організувати додаток як завгодно. Laravel не накладає ніяких обмежень на те, де буде розміщений який-небудь клас, поки Composer буде в змозі автоматично завантажувати цей клас.

Структура папок додаток наведена на рисунку 2.12.

Папка `app` містить код ядра додатку, майже всі класи блогу будуть знаходитися в цій папці.

Папка `bootstrap` містить файли, які завантажують фреймворк і налаштовують автозавантаження. Також в папці `bootstrap` знаходиться папка

cache, яка містить згенеровані фреймворком файли для оптимізації продуктивності – наприклад, кеш-файли маршрутів і сервісів.

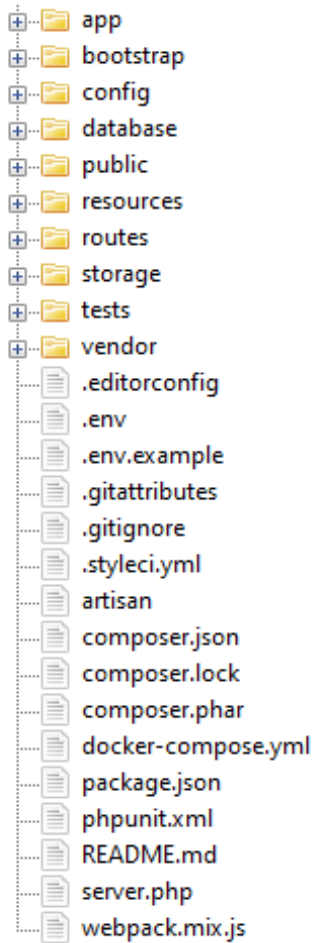


Рисунок 2.12 –
Структура папок

Папка config, як свідчить її назва, містить всі конфігураційні файли додатку.

Папка database містить міграції і класи для наповнення початковими даними БД.

Папка public містить файл index.php, який є вхідний точкою для всіх запитів, що надходять в додаток. Також ця папка містить ресурси, такі як зображення, JavaScript, CSS.

Папка resources містить уявлення, а також сирі,

некомпільовані ресурси, такі як LESS, SASS, JavaScript. А також тут знаходяться всі «мовні» файли.

Папка routes містить всі визначення маршрутів додатку. За замовчуванням в Laravel вбудовано три файли маршрутів web.php, api.php і console.php.

Папка storage містить скомпільовані Blade-шаблони, файл-сесії, кеші файлів і інші файли, створювані фреймворком. Ця папка ділиться на підпапки app, framework і logs. В папці app можна зберігати будь-які файли, які генеруються додатком. В папці framework зберігаються створювані фреймворком файли і кеш. А в папці logs знаходяться файли журналів програми.

Папка tests містить автотест. Спочатку там вже є приклад PHPUnit. Клас кожного тесту повинен мати в імені суфікс Test. В проекті автотест не використовувався.

Папка vendor містить Composer-залежності.

РОЗДІЛ 3.

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ ПЛАНУВАННЯ І ВІДСЛІДКОВУВАННЯ АВТОТРАНСПОРТНИХ ПРОЦЕСІВ

3.1. Опис атрибутів бази даних

В нашій розробці використовується база даних з ім'ям «logistic». Для її адміністрування використовується додаток «PhpMyAdmin». Дані для підключення до БД: логін – root, пароль – root.

Характеристика об'єктів та атрибутів – база даних веб-додатку виконана в середовищі MySQL. База даних складається з певної групи таблиць, котрі можуть містити дані. Кожна таблиця має деякі поля, які, залежно від типу інформації, яка буде зберігатися в цих полях, – матимуть певний тип даних.

Таблиця 3.1 – Характеристика об'єктів бази даних

Назва об'єкту	Характеристика об'єкта
1	2
failed_jobs	Зберігає у собі інформацію про не виконані рейси
flights	Зберігає у собі інформацію про рейси
migrations	Зберігає у собі файли, які призначені для створення таблиць баз даних
password_resets	Зберігає у собі інформацію про скидання паролю
personal_access_tokens	Зберігає у собі інформацію про токен авторизації
statuses	Зберігає у собі інформацію про статуси виконання рейсу
users	Зберігає у собі інформацію про зареєстрованих користувачів веб-додатку

Таблиця 3.2 – Опис атрибутів об'єктів таблиці flights

Назва атрибута	Характеристика атрибута
id	Порядковий номер запису
name	Назва рейсу
manager	Ім'я відповідального менеджера
phone	Номер телефону менеджера
cargo	Категорія вантажу
completion_date	Запланована дата виконання
route	Маршрут перевезення
loading_place	Місто завантаження
uploading_place	Місто розвантаження
user_id	Порядковий номер назначеного водія
status_id	Порядковий номер статусу рейсу
created_at	Дата створення запису
updated_at	Дата останньої модифікації запису

Таблиця 3.3 – Опис атрибутів об'єктів таблиці password_resets

Назва атрибута	Характеристика атрибута
id	Порядковий номер запису
email	Почтова адреса користувача
token	Токен авторизації користувача
created_at	Дата створення запису

Опис інших атрибутів бази даних веб-додатку наведено в [додатку А](#).

3.2. Розробка загальної схеми бази даних

Для створення таблиць використанні стандартні SQL конструкції. Під час розробки таблиць використовувалась цілісність даних для атрибутів, зв'язків та об'єктів. Щоб реалізувати обмеження цілісності зв'язків було використано групу методів під назвою "методи обмеження зовнішнього ключа", в яких NOT NULL сповіщує про обов'язковість зв'язку.

Для отримання цілісності атрибутів зазначені значення типів даних, їх розміри, також встановлені умови на значення атрибутів та визначені обов'язкові значення атрибутів. Для цілісності відношень було застосовано обмеження цілісності первинного ключа (атрибут первинного ключа не NULL і він не може дублюватися в межах відношення).

logistic.flights id : bigint(20) unsigned name : varchar(255) manager : varchar(255) phone : varchar(255) cargo : varchar(255) completion_date : date route : text loading_place : varchar(255) uploading_place : varchar(255) user_id : int(11) status_id : int(11) created_at : timestamp updated_at : timestamp	logistic.migrations id : int(10) unsigned migration : varchar(255) batch : int(11) logistic.statuses id : bigint(20) unsigned status : varchar(255) created_at : timestamp updated_at : timestamp logistic.password_resets email : varchar(255) token : varchar(255) created_at : timestamp	logistic.users id : bigint(20) unsigned name : varchar(255) phone : text auto_number : text trailer_number : text is_admin : int(11) avatar : text email : varchar(255) email_verified_at : timestamp password : varchar(255) remember_token : varchar(100) created_at : timestamp updated_at : timestamp experience : int(11)	logistic.personal_access_tokens id : bigint(20) unsigned tokenable_type : varchar(255) tokenable_id : bigint(20) unsigned name : varchar(255) token : varchar(64) abilities : text last_used_at : timestamp created_at : timestamp updated_at : timestamp	logistic.failed_jobs id : bigint(20) unsigned uuid : varchar(255) connection : text queue : text payload : longtext exception : longtext failed_at : timestamp
--	--	---	---	--

Рисунок 3.1 – Схема бази даних веб-додатку

	id	migration	batch
☐	1	2014_10_12_000000_create_users_table	1
☐	2	2014_10_12_100000_create_password_resets_table	1
☐	3	2019_08_19_000000_create_failed_jobs_table	1
☐	4	2019_12_14_000001_create_personal_access_tokens_table	1
☐	5	2021_11_14_014122_create_flights_table	2
☐	6	2021_11_14_015249_create_status_table	3

Рисунок 3.2 – Зміст таблиці **migrations**

id	tokenable_type	tokenable_id	name	token	abilities	last_used_at	created_at	updated_at
----	----------------	--------------	------	-------	-----------	--------------	------------	------------

Рисунок 3.3 – Зміст таблиці **personal_access_tokens**

	id	status	created_at	updated_at
☐	4	Отменен	NULL	NULL
☐	2	Запланирован	NULL	NULL
☐	3	Завершен	NULL	NULL
☐	1	В работе	NULL	NULL

Рисунок 3.4 – Зміст таблиці **statuses**

Схему бази даних для роботи веб-додатку було реалізовано в додатку phpMyAdmin 5.0.2, який поширюється з відкритим кодом, та розроблений на мові програмування PHP і використовується для адміністрування СУБД MySQL. Схему БД наведено на рисунку 3.1.

Дамп бази даних наведений у додатку Б.

3.3. Розробка загальної конфігурації системи

Файл конфігурації додатку створюється автоматично фреймворком та знаходиться у корені проекту з назвою .env (рис. 3.5).

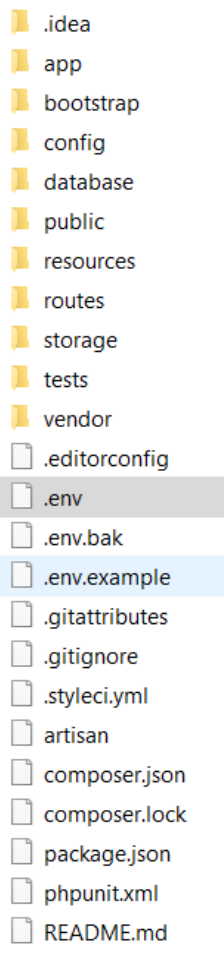


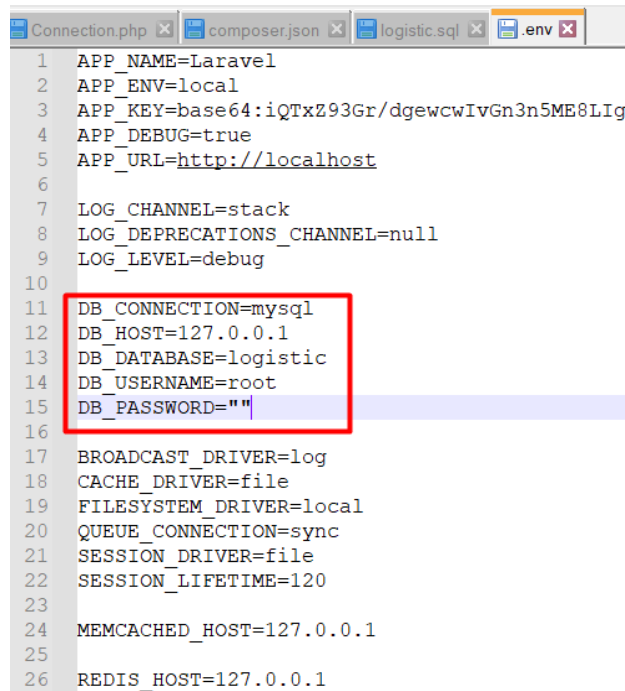
Рисунок 3.5 –
Файл конфігурації
.env

Тут відбувається підключення проекту до бази даних, вказуючи клас під'єднання (DB_CONNECTION), посилання на хост (DB_HOST), назву бази даних (DB_DATABASE), користувача (DB_USERNAME), пароль (DB_PASSWORD) та тип кодування (charset) (рис. 3.6).

Замість того, щоб визначати всю логіку обробки запитів у вигляді замикань в файлах маршрутів, можна організувати її за допомогою класів контролерів. Контролери можуть групувати пов'язану з обробкою HTTP-запитів логіку в окремий клас.

Контролери є частиною MVC архітектури. Це об'єкти класів, що відповідають за обробку запиту і генерування відповіді. Після обробки запиту додатками, контролери проаналізують вхідні дані, передадуть їх в моделі, вставляють результати моделі в уявлення, і в кінцевому підсумку згенерують вихідні відповіді.

Контролери складаються з дій, які є основними блоками, до яких може звертатися кінцевий користувач і запитувати виконання того чи іншого функціоналу. У контролері може бути одне або декілька дій.



```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:iQTxZ93Gr/dgewcwIvGn3n5ME8LIg
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_DATABASE=logistic
14 DB_USERNAME=root
15 DB_PASSWORD=""
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 FILESYSTEM_DRIVER=local
20 QUEUE_CONNECTION=sync
21 SESSION_DRIVER=file
22 SESSION_LIFETIME=120
23
24 MEMCACHED_HOST=127.0.0.1
25
26 REDIS_HOST=127.0.0.1

```

Рисунок 3.6 – Фрагмент файлу з підключенням до бази даних

Основним контролером додатку є Controller. Розташований він в папці `app\Http\Controllers`.

Основні завдання контролерів: 1) можуть мати доступ до даних запиту; 2) можуть викликати методи моделей та інших компонентів системи з даними запиту; 3) можуть використовувати уявлення для формування відповіді; 4) не повинні займатися обробкою даних, це повинно відбуватися в шарі моделей; 5) повинні уникати використання HTML, або інший розмітки, краще це робити в уявленнях.

Простори імен служать для логічного угруповання імен класів, щоб їх можна було відрізнити від інших, навіть якщо їх імена співпадають. Важливо пам'ятати, що при визначенні маршруту контролера не треба вказувати повний простір імен контролера, а тільки ту частину імені класу, яка слідує за «коренем» простору імен – `App\Http\Controllers`. Тому що `RouteServiceProvider` завантажує файли маршрутів разом з групою маршрутизації, що містить кореневе простір імен контролера.

Автозавантаження дозволяє вам використовувати класи без явного підключення їх файлів за допомогою `require()` або `include()`. Таким

чином, тільки ті класи, які дійсно використовуються будуть завантажені і можна почати використовувати будь-який клас не замислюючись про те, чи був він підключений чи ні. За замовчуванням, в файлі `application/start.php` вже визначено автозавантаження для папок `application/models` і `application/libraries`.

Завантажувач використовує ім'я класу для визначення шляху до його файлу. Всі імена файлів наводяться до нижнього регістру. Хоча, клас `User` всередині `application/models` повинен бути визначений в файлі `user.php`.

Можна також використовувати вкладені папки якщо класи містяться в просторі імен. Наприклад, клас `Entities\User` може міститися у файлі `application/models/entities/user.php`.

3.4. Маршрутизація потоків додатку

У Laravel маршрутизація призначена для направлення запиту до відповідного контролера. Маршрути додатки можуть бути визначені в файлі `app/Http/routes.php`. Механізм маршрутизації наведено на рисунку 4.3.

Крок 1. Спочатку потрібно виконати кореневу URL-адресу програми.

Крок 2. Виконана URL-адреса буде відповідати методу в файлі `route.php`. У нашому випадку вона буде відповідати кореневій URL ('/'). Таким чином буде виконана відповідна функція.

Крок 3. Функція `resources/views/welcome.blade.php` викликає файли шаблону. Потім дана функція викликає функцію `view()` з аргументом «welcome» без використання `blade.php`.

Часто в додатку також потрібно отримати параметри, пов'язані з URL-адресою. Для цього необхідно відповідним чином змінити код в файлі `routes.php`. Існує два способи, за допомогою яких можна отримати параметри, передані з URL-адресою:

- 1) обов'язкові параметри;
- 2) додаткові параметри.

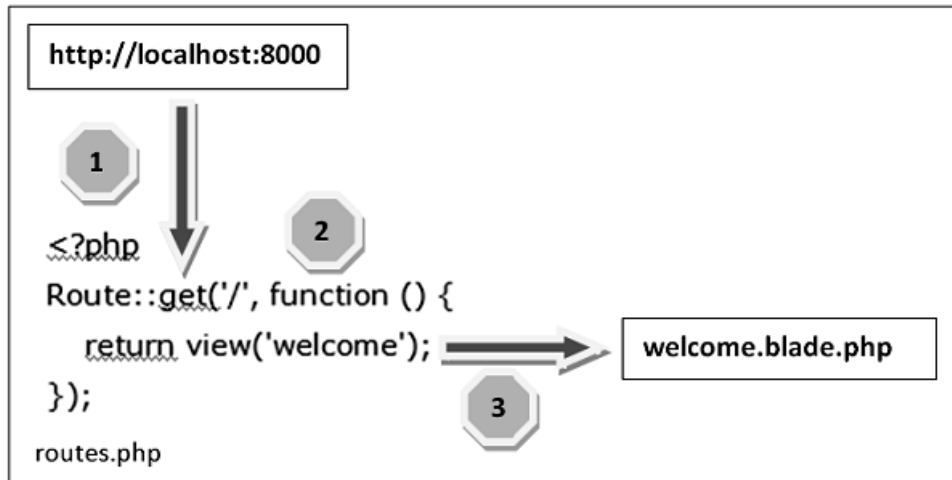


Рисунок 4.3 – Маршрутизація потоків в PHP-фреймворку Laravel

Обов'язкові параметри повинні бути обов'язково присутніми в URL-адресі. Наприклад, можна отримати ідентифікатор з URL-адреси, щоб виконати з ним певні дії. Незалежно від аргументів, які ми передаємо після кореневої URL-адреси (`http://localhost:8000/ID/5`), цей параметр буде зберігатися в `$id`, і можна використовувати його для подальшої обробки, проте тут він просто виводиться. Таким чином, можна передати його в уявлення або в контролер для подальшої обробки.

3.5. Логіка обробки запитів та відповідей веб-сервера

Вхідна точка для всіх запитів до додатка – файл `public/index.php`. Всі запити направляються в цей файл налаштуваннями веб-сервера (Apache/Nginx). Файл `index.php` містить досить мало коду. Швидше, він просто відправна точка для завантаження всього іншого фреймворка.

Файл `index.php` завантажує згенероване за допомогою Composer визначення автозавантажувача, а потім витягує екземпляр Laravel-додатку зі скрипта `bootstrap/app.php`. Перша дія самого Laravel – створення

екземпляра додатка/сервіс-контейнера. Далі вхідний запит надсилається або в HTTP-ядро, або в ядро консолі, в залежності від типу цього запиту. Ці ядра служать центральним місцем, через яке проходять усі запити.

HTTP-ядро успадковує клас `Illuminate\Foundation\Http\Kernel`, який визначає масив завантажувачів `bootstrappers`, які будуть запуснені перед виконанням запиту. Ці завантажувачі налаштовують обробку помилок, налаштовують ведення журналів, визначають середу додатку і виконують інші завдання, які треба виконати перед самою обробкою запиту.

HTTP-ядро також визначає список посередників HTTP, через які повинні пройти всі запити, перш ніж будуть оброблені додатком. Ці посередники обробляють читання і запис HTTP-сесії, визначаючи, чи знаходиться додаток в режимі обслуговування, перевіряють CSRF-послідовність, і т.п. Принцип дії методу `handle` HTTP-ядра дуже простий: отримати `Request` і повернути `Response`. Слід уявити ядро як велику чорну коробку, яка представляє собою все додаток. Треба наповнювати його HTTP-запитами і воно буде повертати HTTP-відповіді.

Одна з найважливіших дій ядра при завантаженні – завантаження постачальників послуг для додатку. Усі постачальники послуг налаштовуються в конфігураційному файлі `config/app.php` в масиві `providers`. Спочатку буде викликаний метод `register` для всіх постачальників, а коли всі вони будуть зареєстровані, буде викликаний метод `boot`.

Постачальники послуг відповідають за початкову завантаження всіляких компонентів фреймворка: БД, чергу, помилки під час введення і маршрутизація. Постачальники послуг – найважливіший елемент всього процесу початкового завантаження `Laravel`, так як вони відповідають за завантаження і налаштування всіх можливостей, необхідних фреймворку.

Коли додаток завантажено, і всі постачальники послуг зареєстровані, запит `Request` буде переданий роутера для відправки. Роутер відправить запит по маршруту або контролера, а також запустить посередника, маршруту.

РОЗДІЛ 4.

ПРАКТИЧНЕ ЗАСТОСУВАННЯ ВЕБ-ДОДАТКУ

4.1. Поєднання та взаємодія програмних модулів

Дуже важливо розуміти, що модель, як шар, існує незалежно від фреймворка, HTTP і веба в цілому. Вся інша система може звертатися до моделей, але моделі не можуть знати і не знають нічого про середовище, в якій вони виконуються. Модель в MVC – це шар додатку, що відповідає за зв'язок з предметною областю. У ньому знаходиться вся бізнес-логіка додатку. Для простоти реалізації, сюди включають механізми для роботи з базою даних.

Програмні модулі. В проекті створені моделі які відповідають кожній таблиці в базі даних. Майже всі вони є нащадками класу Model так як взаємодіють з базою даних і вносять до неї зміни.

Структура модулів проекту включає три елементи:

- 1) Flight.php;
- 2) Status.php;
- 3) User.php.

User – модуль працює з таблицею user в базі даних. Містить методи, що дозволяють ідентифікувати нашого автора.

Фрагмент коду даного модуля відображений на рисунку 4.1.

Опис блоків та загальний вигляд веб-додатку. Перед початком розробки блоку необхідно створити макети сторінок (рис. 4.1). На головній сторінці розміщено елементи:

- 1) шапка сайту – для відображення меню;
- 2) розміщення контенту – відображатимуться створені статті.

Справа розміщені два блоки – блок з полем для пошуку статей та блок з переліком категорій блогу. Під час розробки шаблону головного вікна використано фреймворк Bootstrap, за допомогою якого оформлені усі блоки, завдяки чому сторінки блогу є адаптивними.

```

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    protected $guarded = [];

    protected $hidden = [
        'password',
        'remember_token',
    ];

    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    public function flights()
    {
        return $this->hasMany(Flight::class);
    }
}

```

pertext Prepr length : 626 lines : 33 Ln : 23 Col : 1 Pos : 460

Рисунок 4.1 – Модуль User

Меню, стилі кнопок, а також відступи, таблиці тощо оформлені із використанням вказаного фреймворку. Один із основних компонентів моделі MVC це View. "View" – компонент служить для відтворення даних в виді користувацького інтерфейсу, перетворює дані отримані від "Model and Controller", у форму, зручну для інтерфейсу користувача. В Web-додатках "View" – головним чином HTML-сторінка. "View" отримує дані від "Моделі" (через "Контролер") і направляє їх в інтерфейс користувача. Дана компонента ні за яких обставин не може стати причиною зміни бази даних, вона служить тільки для виведення даних, отриманих від "Моделі".

Архітектуру MVC наведено на рисунку 4.2.

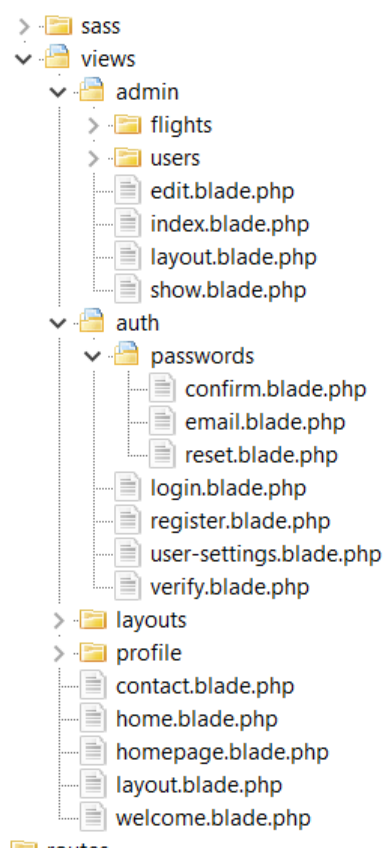


Рисунок 4.2 – Архітектура компоненти Views

Views ідентифікують як шаблони. Зокрема, для відтворення фронтальної частини сайту, в папці Views міститься шаблон – `welcome.blade.php` в якому підключено всі стилі, віджети, головне меню сайту. Структура папки `views` зображена на рисунку 4.3.

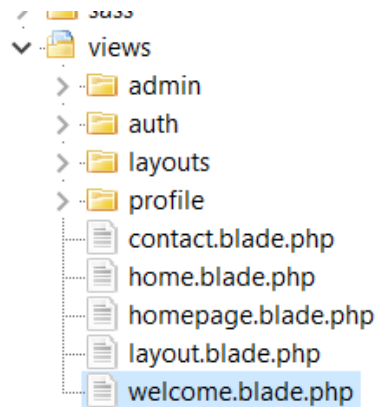


Рисунок 4.3 – Структура папки `views`



Рисунок 4.4 – Код шаблону (фрагмент)

Відображається статична та динамічна частина сайту, фрагмент коду, що відповідає за контент на сайті відображено на рисунку 4.4.

4.2. Функціонал та результати використання веб-додатку

На рисунку 4.5 зображена головна сторінка, на якій можна бачити назву сторінки, меню сайту, та текстову частину.



Чому саме Ми?

- Ми гарантовано знаходимо замовлення, Ви не залишитеся без роботи!
 - Авторизація водія та машини!
 - Вантаж застраховано!

Рисунок 4.5 – Веб-додаток для планування автотранспортних процесів

На рисунку 4.6 зображена сторінка «Контакти» на якій форма для заповнення, для зворотнього зв'язку.



Рисунок 4.6 – Сторінка для зворотнього зв'язку «Контакти»

На рисунку 4.7 відображена форма Авторизації для реєстрації користувача веб-додатку.

Реєстрація

Email
Введіть Ваш email

ПІБ
Введіть Ваше прізвище та ім'я

Номер телефону
Введіть Номер телефону

Номер транспортного засобу
Введіть Номер транспортного засобу

Номер причепа
Введіть Номер причепа

Водійський стаж
Введіть Водійський сатж

Пароль
Введіть пароль

Зареєструватися [На головну](#)

Рисунок 4.7 – Авторизація замовника/перевізника

На рисунку 4.8 зображена вікно налаштування та адміністрування. В ньому можна додавати рейси, а також редагувати або видалити існуючі, додавати водіїв, а також відредагувати або видалити існуючих.

Адмін-форма [На головну](#)

Вітаю, Admin!

Створити рейс

Рейси

Водії

Вийти

Назва маршруту	Водій	Редагувати	Видалити
Ужгород-Львів Мінеральна вода	Андрій Герєга	Редагувати	Видалити
Львів-Київ Автозапчастини	Леонід Кучма	Редагувати	Видалити
Київ-Суми Побутова техніка	Степан Герєга	Редагувати	Видалити

Рисунок 4.8 – Вікно налаштування та адміністрування Рейсів перевезення

Під час редагування рейсу можна змінити назву маршруту, менеджера, контактний номер менеджера, вантаж, дату виконання, маршрут, місця завантаження та вивантаження тощо.

Під час редагування водія можна змінити його дані (адресу, ПІБ, номер телефону, номер транспортного засобу та причепа, водійський стаж та аватар тощо) (рис. 4.9).

Адмін-форма На головну

**Вітаю,
Admin!**

Рейси

Водії

Вийти

Email

ПІБ

Номер телефону

Номер транспортного засобу

Номер причепа

Водійський стаж

Фото водія

Файл не вибрано

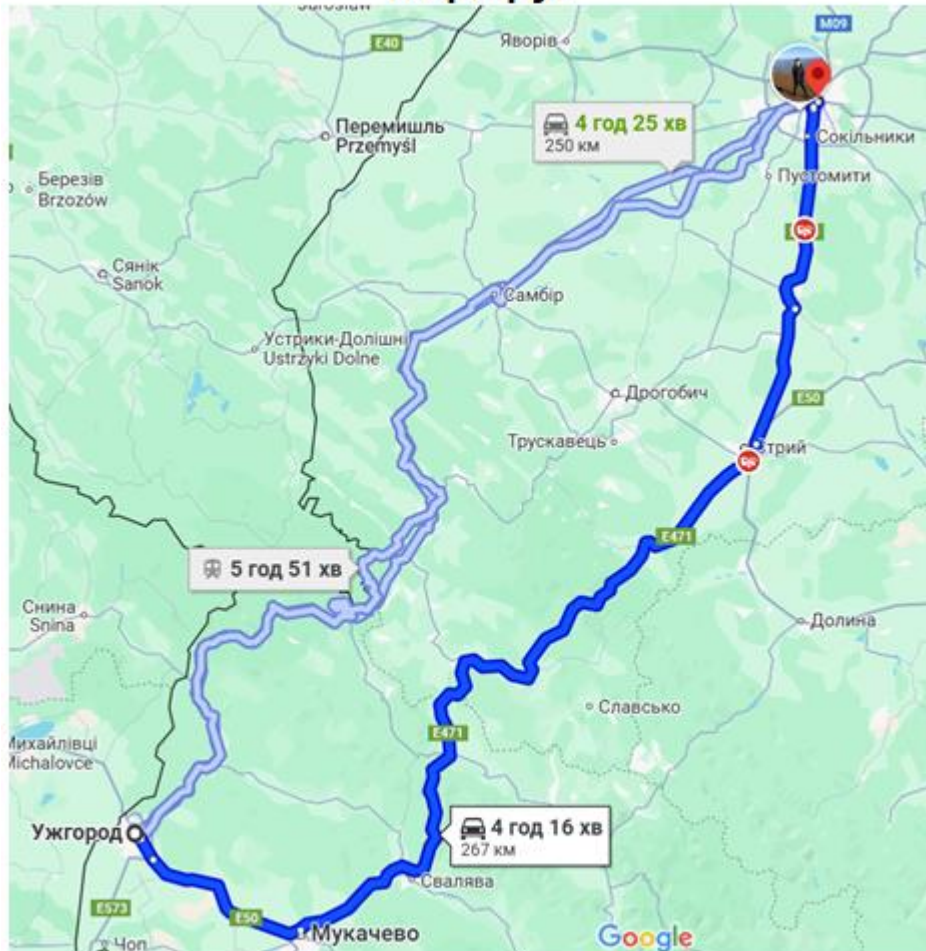
Рисунок 4.9 – Адмін-форма внесення даних водія

Перегляд окремого рейсу наведено на рисунку 4.10. Заголовком показано назву рейсу, у вигляді таблиці вказується ім'я менеджера, його контактний номер, номер рейсу, вантаж, дату виконання, місто завантаження та розвантаження, статус рейсу тощо. Виведена форма карти гугл, з прокладеним

маршрутом, блок інформації про водія, його повне ім'я, номер телефону, номер машини та причепа, і завершальний етап – форма зміни статусу рейсу.

Місце завантаження	Ужгород
Місце вивантаження	Львів
Статус рейсу	Заплановано

Маршрут



Водій

Водій	Андрій <u>Герега</u>
Контактний номер	+380972709349
Номер ТЗ	BC 7117 <u>СВ</u>

Рисунок 4.10 – Сторінка перегляду даних рейсу

Таким чином, поставлені задачі ініціювання та запуску проекту щодо розробки веб-додатку планування автотранспортних процесів виконано. Реалізовано веб-додаток з користувачами, правами користувача та адміністратора, головну сторінку додатку, панель адміністратора, панель користувача (водія) тощо.

РОЗДІЛ 5.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Структурно-функціональний аналіз технологічного процесу

Розробка та вживання ефективних заходів запобігання аварійним і травмонебезпечним ситуаціям можливі лише при завчасному виявленні тих небезпек, з яких починаються процеси їх формування. Оскільки небезпечні умови не завжди завчасно можна виявити, а для вивчення небезпечних дій іноді потрібно багато часу, щоб зібрати статичний матеріал, то і методи виявлення цих небезпек повинні бути відповідно диференційовані (табл. 5.1) [5].

Таблиця 5.1. Моделі формування та виникнення травмонебезпечних і аварійних ситуацій

Вид робіт, виробничий підрозділ, робоче місце, виробниче обладнання, склад агрегату	Виробнича безпека			Можливі наслідки	Заходи запобігання небезпечним ситуаціям
	Небезпечна умова (НУ)	Небезпечна дія (НД)	Небезпечна ситуація (НС)		
Виконання робіт із електрообладнанням	Не вимкнено живлення. Відсутність заземлення.	Нехтування правилами ТБ	Ураження струмом	Травма (Т)	Проведення повторного інструктажу з ТБ. Розробка нових способів захисту. Встановлення заземлення.
НД ↓ НУ → НС → Т					

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи [5]:

- характеризують стан або рівень небезпеки обладнання, які використовуються.

- сприяють виникненню технологічних помилок обслуговуючого персоналу впродовж виробничого процесу;
- створювати умови та можливість проникнення працівника в небезпечну зону;
- приводять до виникнення небезпечних дій (внаслідок низького рівня професійної підготовки працівників та організації навчання з охорони праці).

Моделі формування та виникнення травмонебезпечних і аварійних ситуацій в комп'ютерному кабінеті представлено у вигляді моделі формування та виникнення травмонебезпечних і аварійних ситуацій – табл. 5.1.

5.2. Розрахунок освітлення приміщення комп'ютерного кабінету

Освітленість виробничих приміщень може бути штучною і природною. Природне освітлення при правильному обладнанні найбільш сприятливе для людини. Основні вимоги для освітлення наступні:

- освітлення повинне бути достатнім для швидкого і легкого розпізнання об'єктів роботи;
- освітлення повинно бути рівномірне без різких тіней;
- джерело світла не повинно осліплювати працівника;
- рівень освітленості не повинен обмежуватись часом.

Природне освітлення забезпечується обладнанням вікон (бокове освітлення) фонарів і світильних покриттів приміщень (верхнє освітлення). Природне освітлення нормується коефіцієнтом природної освітленості. Коефіцієнт природної освітленості – це процентне відношення фактичної освітленості F_v в будь-якій точці приміщення до освітленості F_n розсіяної світлом небозводу точки, яка лежить на відкритій місцевості. Розрахунок природного освітлення через бокові вікна по нормам освітленості ведеться для

самої дальньої від вікон точки, тобто знаходять мінімальне значення стик коефіцієнта природної освітленості [5]:

$$e_{\min} = \frac{F_b}{F_n} \cdot 100. \quad (5.1)$$

Значення коефіцієнта природної освітленості визначається не менше чим в п'яти точках. Значення коефіцієнта природної освітленості для сільськогосподарських виробничих приміщень в даному випадку ремонтній майстерні, беремо $e_{\min} = 5\%$.

Розрахунок природного освітлення зводиться до визначення площі світлових променів.

Сумарну площу світлових променів $\sum F_o (m^2)$ по коефіцієнту природної освітленості для бокових променів визначаємо по формулі:

$$\sum F_o = \frac{F_n \cdot e_{\min} \cdot r_o \cdot K}{100 \cdot \tau \cdot \Gamma_1}, \quad (5.2)$$

де F_n – площа підлоги, m^2 ; $e_{\min}$ – величина мінімального коефіцієнта природного освітленості; τ – загальний коефіцієнт світловикористання віконного отвору із врахуванням його забруднення, $\tau = 0,25$; r_o – світлова характеристика вікна, $r_o = 9,5$; Γ_1 – коефіцієнт, який враховує підвищення освітленості за рахунок світла, яке відбивається від стін і стелі, $\Gamma_1 = 1,2$; K – коефіцієнт, який враховує затінення вікон сусідніми приміщеннями і загорожею, $K = 1$.

$$\sum F_o = \frac{36 \cdot 0,5 \cdot 9,5 \cdot 1}{100 \cdot 0,25 \cdot 1,2} = 5,7 m^2.$$

Кількість світлових променів визначимо:

$$N = \frac{\sum F_o}{F_o}, \quad (5.3)$$

де F_o – площа вікна згідно стандарту, m^2 .

$$N = \frac{5,7}{6} = 0,95.$$

Приймаємо кількість вікон – одне вікно.

При розрахунку природного освітлення найбільш поширеним і простим є метод світлового потоку. При цьому методі розраховуємо світловий потік $F_{\text{л}}$ (Лк), який повинна випромінювати кожна лампа (при заданій кількості ламп).

$$F_{\text{л}} = \frac{k \cdot S_{\text{п}} \cdot E}{n_{\text{л}} \cdot \eta \cdot r^2}, \quad (5.4)$$

де k – коефіцієнт запасу, $k = 1,3$; $S_{\text{п}}$ – площа підлоги, м^2 ; $S_{\text{п}} = 36 \text{ м}^2$. E – нормативна освітленість, $E = 300 \text{ Лк}$; $n_{\text{л}}$ – кількість встановлених ламп, $n_{\text{л}} = 6$ од; η – коефіцієнт використання світлового потоку, $\eta = 0,25$; r – коефіцієнт нерівномірності освітленості, $r = 0,545$.

Коефіцієнт запасу (K) враховує можливість забруднення світильників пилом, що залежить від характеру виробництва.

Розрахунок штучного освітлення починаємо із визначення висоти розташування світильника і їх кількості. Висоту $h_{\text{н}}$ (м) розташування світильників над робочим місцем знаходимо за формулою:

$$h_{\text{н}} = H - (h_1 + h_2), \quad (5.5)$$

де H – висота приміщення, м; h_1 – віддаль від підлоги до освітлювальної поверхні, м; h_2 – віддаль від стелі до світильника, м.

$$h_{\text{н}} = 4,5 - (2,2 + 1,5) = 0,8 \text{ м.}$$

При симетричному розміщенні світильників по вершинах квадратів їх кількість визначається за формулою:

$$n_{\text{с}} = \frac{S_{\text{п}}}{l^2}, \quad (5.6)$$

де l – віддаль між світильниками, м.

Підставивши значення отримаємо:

$$n_{\text{с}} = \frac{36}{9} = 4 \text{ од.}$$

Тоді світловий потік буде становити

$$F_{\text{л}} = \frac{1,3 \cdot 36 \cdot 300}{4 \cdot 0,25 \cdot 0,545} = 2576,2 \text{ Лк.}$$

При світловому потоці 2576,2 Лк для заданої лампи вибираємо тип і потужність.

Вибираємо тип лампи – люмінесцентну, потужністю 40Вт.

5.3. Безпека в надзвичайних ситуаціях

Забезпечення захисту населення і території у разі загрози і виникнення надзвичайних ситуацій є одним з най важливіших завдань держави.

Захист населення є системою загально державних заходів, які реалізуються центральними і місцевими органами виконавчої влади, виконавчими органами влад, органами управління з питань надзвичайних ситуацій та цивільного захисту населення, підпорядкованими їм системами, та підприємств, що забезпечують виконання організаційних, інженерно – технічних, санітарно – гігієнічних, проти епідемічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій.

Загрози життєво важливих інтересів громадян, держави, суспільства поділяють на зовнішні та внутрішні, виконують під час надзвичайних ситуацій техногенного та природного характеру та воєнних конфліктах.

Принципи захисту впливають з основних положень Женевської конвенції щодо захисту жертв війни та додаткових протоколів до неї, можливого характеру воєнних дій, реальних можливостей держави щодо створення матеріальної бази захисту. З метою захисту населення, зменшення втрат та шкоди економіці в разі виникнення надзвичайних ситуацій має право проводитись спеціальний комплекс заходів.

Оповіщення та інформування, яке досягається завчасним створенням і підтримкою в постійній готовності загально державної, територіальних та об'єктивних систем оповіщення населення.

ВИСНОВКИ І РЕКОМЕНДАЦІЇ

Кількість підприємств, що використовують Інжернет як канал зв'язку із клієнтами та працівниками зростає щороку. Всесвітня мережа дозволяє обмінюватись інформацією зі своєю цільовою аудиторією, проводити безпечні транзакції. Проте ефективна взаємодія можлива лише тоді, коли бізнес зможе швидко збирати та обробляти дані про клієнтів, відповідати на запити тощо. Звичайні сайти нездатні впоратися з усіма завданнями бізнесу, тому на допомогу приходять веб-додатки.

Для запуску веб-додатків потрібні браузер та веб-сервер. Застосунок працює за такою схемою: 1) користувач надсилає запит до веб-серверу, що складається з центру керування застосунком (логіки) та бази даних; 2) центр керування отримує запит й скеровує його до бази даних; 3) інформація витягується з бази даних й передається у центр керування; 4) вебсервер відправляє клієнту інформацію – й вона з'являється на дисплеї користувача.

Сучасні технології дють можливість спостерігати за транспортуванням вантажів в реальному часі, зокрема, через Інтернет за допомогою маячків GPSTрекерів, або софт-трекерів тощо. Це в свою чергу є гарантією того, що перевізник буде дотримуватися узгодженого маршруту.

PHP-фреймворк Laravel є одним із найпоширеніших фреймворків у веб-розробці. Розробники віддають перевагу цьому фреймворку, поступово відходячи від конкуруючих рішень, наприклад Yii, Zend та інших. Laravel поширюється на безкоштовній основі, та має відкритий код. Функціонал Laravel є простим для розуміння та використання. Більшість його функцій, спираючись на загальноприйняті стандарти написання коду, роблять його інтуїтивно зрозумілим, працюють не вимагаючи додаткових налаштувань.

MVC призначений для поділу бізнес-логіки і призначеного для користувача інтерфейсу, щоб розробники могли легко змінювати окремі частини програми, не зачіпаючи інші. В архітектурі MVC модель надає дані і правила бізнес-логіки, уявлення відповідає за користувацький інтерфейс

(наприклад, текст, поля введення), а контролер забезпечує взаємодію між моделлю і представленням.

База даних веб-додатку виконана в середовищі MySQL. База даних складається з певної групи таблиць, котрі можуть містити дані. Кожна таблиця має деякі поля, які, залежно від типу інформації, яка буде зберігатися в цих полях, – матимуть певний тип даних.

Відповідно до завдань кваліфікаційної роботи детально проаналізовано предметну сферу, проаналізовано конкурентоспроможні додатки-аналоги, основні їх недоліки та переваги, описано вимоги до проєктованого веб-сервісу тощо. Для реалізації поставленого завдання вибрано інструменти для реалізації веб-додатку, який базується на сучасних технологіях.

У розробленому веб-додатку реалізовано такі основні можливості для користувача як підтримка роботи додатку з 2 ролями користувачів, можливість створення рейсу, можливість створення водія, можливість редагування та видалення рейсу, можливість редагування та видалення водія, можливість планування маршруту рейсу, можливість редагування профілю водія, під час авторизації під користувачем, можливість зміни статусу рейсу водієм;

Перевірка функціональності та практичне значення розробки полягає в створенні веб-додатку, який дає змогу адміністратору внести в базу даних вантаж, запланувати маршрут та занести його в базу даних, назначити відповідного користувача(водія), цей запис може отримати користувач додатку (водій), та відповідно реалізувати запланований маршрут, отримати супроводжуючу документацію.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Автоперевезення. Компанія DELLA™. URL: <https://della.com.ua>
2. Вантажні перевезення. Ukrainian Logistic. URL: <https://ukrainianlogistic.com/>
3. Голуб Б.М. С#. Концепція та синтаксис. Навч. посібник / Б.М. Голуб, Львів: Видавничий центр ЛНУ імені Івана Франка, 2016. 136 с.
4. Дмитриченко М. Ф. Основи теорії транспортних процесів і систем : навчальний посібник для студентів ВНЗ напрямку “Транспортні технології” / М-во освіти і науки України ; М. Ф. Дмитриченко, Л. Ю. Яцківський, С. В., Ширяєва, В. З. Докуніхін. К.: Слово, 2019. 336 с.
5. Пархоменко І. І., Покровська О. В., Івашко С. В. Організація управління охороною праці: навч. посіб. / за ред. І. І. Пархоменко, О. В. Покровської. Київ: ВПЦ «Київський університет», 2023. 256 с.
6. Пошук вантажу та транспорту. Lardi-trans. URL: <https://lardi-trans.com/gruz/>
7. Редзюк А.М. Автомобільний транспорт України: стан, проблеми, перспективи розвитку / Державний автотранспортний науководослідний проектний інститут; Монографія за Заг. ред. А. М. Редзюка. К.: ДП "ДержавтотрансНДІпроект", 2015. 400 с.
8. Спірін О. М. Зміст навчального матеріалу спецкурсу "Хмарні інформаційно-аналітичні технології у науково-дослідному процесі" / О. М. Спірін, О. А. Одуд // Інформаційні технології і засоби навчання. 2016. Т. 52, Вип. 2. С. 108-120.
9. Стеценко І.В. Моделювання систем: навч. посіб. / І.В. Стеценко; М-во освіти і науки України, Черкас. держ. технол. ун-т. Черкаси : ЧДТУ, 2020. 399 с.
10. Структура розбиття робіт (Work Breakdown Structure – WBS). 2023. URL: <https://www.maxzosim.com/struktura-rozbittia-robot/>
11. Тимченко А.А. Основи системного проектування та системного аналізу

складних об'єктів: Підручник для студентів вищих закладів освіти / За ред. В.І. Бикова. К.: Либідь, 2020. 270 с.

12. Томашевський В. М. Моделювання систем. К.: Видавнича група BVH, 2015, 352 с.
13. Шилдт Г. C# 4.0: повне керівництво. М.: ТЗОВ "І.Д. Вільямс", 2021. 1056 с.
14. 5 найкращих диспетчерських програмних забезпечень для логістичних компаній. URL: <https://stfalcon.com/uk/blog/post/5-best-trucking-dispatch-software-for-logistic-companies>
15. BursaTransport. Офіційний сайт. URL: <https://www.bursatransport.com/desprebursatransport.html>
16. BURZA NÁKLADŮ 24. Офіційний сайт. URL: <http://www.burzanakladu24.cz/>
17. Cargopedia. Офіційний сайт. URL: <https://www.cargopedia.bg/>
18. Grand Logistics Company. Офіційний сайт. URL: <https://glc.in.ua/uk/>
19. IDEF0 діаграма: приклади і правила побудови // ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ. 2021. URL: <https://ukr.kagutech.com/3929706-idef0-diagram-examples-and-construction-rules>
20. My dello. Офіційний сайт. URL: <https://mydello.com/>
21. Notepad++. Офіційний сайт. URL: <https://notepad-plus-plus.org/>
22. Novoferm. Офіційний сайт. URL: <https://www.logistyczny.com/>
23. Organization Breakdown Structure (OBS). 2020. URL: <https://uplandsoftware.com/psa/resources/glossary/organization-breakdown-structure-obs/>
24. SMART Goals. 2023. URL: <https://www.mindtools.com/pages/article/smart-goals.htm>
25. Sytnyk, V. F (2022). Decision support systems, KNEU, Kyiv, 614 s.

ДОДАТКИ

Додаток А.

Опис атрибутів об'єктів таблиці бази даних

Таблиця А.1 – Опис атрибутів об'єктів таблиці personal_access_tokens

Назва атрибута	Характеристика атрибута
id	Порядковий номер запису
tokenable_type	Тип токена
tokenable_id	Порядковий номер токена
name	Ім'я користувача
token	Токен
abilities	Права для токена
last_used_at	Дата останньої авторизації
created_at	Дата створення запису
updated_at	Дата редагування запису

Таблиця А.2 – Опис атрибутів об'єктів таблиці statuses

Назва атрибута	Характеристика атрибута
id	Порядковий номер запису
status	Тип статусу: «Відмінено», «Заплановано», «Завершено», «В роботі»
created_at	Дата створення запису
updated_at	Дата редагування запису

Таблиця А.3 – Опис атрибутів об'єктів таблиці users

Назва атрибута	Характеристика атрибута
id	Порядковий номер запису
name	Ім'я водія
phone	Мобільний номер водія
auto_number	Номер тягачу
trailer_number	Номер причепа
avatar	Інформація про аватар водія
email	Почта водія
email_verified_at	Запис чи перевірена почта водія
password	Зашифрований пароль
remember_token	Запам'ятовувати чи ні токен
created_at	Дата створення запису
updated_at	Дата редагування запису
experience	Стаж (досвід) водія

Додаток Б. ФРАГМЕНТ КОДУ ВЕБ-ДОДАТКУ.

Файл *layout.blade.php*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>Clean Blog - Start Bootstrap Theme</title>
    <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
    <!-- Font Awesome icons (free version)-->
    <script
        src="https://use.fontawesome.com/releases/v5.15.4/js/all.js"
crossorigin="anonymous"></script>
    <!-- Google fonts-->
    <link
        href="https://fonts.googleapis.com/css?family=Lora:400,700,400it
alic,700italic" rel="stylesheet" type="text/css" />
    <link
        href="https://fonts.googleapis.com/css?family=Open+Sans:300itali
c,400italic,600italic,700italic,800italic,400,300,600,700,800"
        rel="stylesheet"
type="text/css" />
    <!-- Core theme CSS (includes Bootstrap)-->
    <link href="/css/styles.css" rel="stylesheet" />
</head>
<body>

<!-- Navigation-->
<nav class="navbar navbar-expand-lg navbar-light" id="mainNav">
    <div class="container px-4 px-lg-5">
        <a class="navbar-brand" href="/">LogisticGroup</a>
        <button class="navbar-toggler" type="button" data-bs- toggle="collapse"
data-bs-target="#navbarResponsive" aria- controls="navbarResponsive" aria-
expanded="false" aria- label="Toggle navigation">
            Menu
            <i class="fas fa-bars"></i>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
            <ul class="navbar-nav ms-auto py-4 py-lg-0">
                <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4"
href="/">Головна</a></li>
                <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4"
href="/login">Авторизація</a></li>
                <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4"
href="/register">Реєстрація</a></li>
                <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4"
href="/contacts">Контакти</a></li>
                <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4"
href="/admin">Адмін-панель</a></li>
                <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4"
href="/profile">Особистий кабінет</a></li>
                <li class="nav-item"><a class="nav-link px- lg-3 py-3 py-lg-
4" href="/logout">Вийти</a></li>
            </ul>
        </div>
    </div>
</nav>
```

```

        @endif()
    </ul>
</div>
</div>
</nav>

@yield('content')

<!-- Footer-->
<footer class="border-top">
    <div class="container px-4 px-lg-5">
        <div class="row gx-4 gx-lg-5 justify-content-center">
            <div class="col-md-10 col-lg-8 col-xl-7">
                <ul class="list-inline text-center">
                    <li class="list-inline-item">
                        <a href="#">
                            <span class="fa-stack fa-lg">
                                <i class="fas fa-circle fa-stack-2x"></i>
                                <i class="fab fa-twitter fa-stack-1x fa-inverse"></i>
                            </span>
                        </a>
                    </li>
                    <li class="list-inline-item">
                        <a href="#">
                            <span class="fa-stack fa-lg">
                                <i class="fas fa-circle fa-stack-2x"></i>
                                <i class="fab fa-facebook-f fa-stack-1x fa-inverse"></i>
                            </span>
                        </a>
                    </li>
                    <li class="list-inline-item">
                        <a href="#">
                            <span class="fa-stack fa-lg">
                                <i class="fas fa-circle fa-stack-2x"></i>
                                <i class="fab fa-github fa-stack-1x fa-inverse"></i>
                            </span>
                        </a>
                    </li>
                </ul>
            </div>
            <div class="small text-center text-muted fst-italic">Мини-Б
            соқ.мөрежа</div>
        </div>
    </div>
</footer>

<!-- Bootstrap core JS-->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootst
rap.bundle.min.js"></script>
<!-- Core theme JS-->
<script src="js/scripts.js"></script>
</body>
</html>

```


Файл *homepage.blade.php*

```

@extends('layout')

@section('content')
    <!-- Page Header-->
    <header class="masthead" style="background-image:
url('/assets/img/background.jpeg')">
        <div class="container position-relative px-4 px-lg-
5">

center">

        <div class="row gx-4 gx-lg-5 justify-content-

            <div class="col-md-10 col-lg-8 col-xl-7">
<div class="site-heading">
    <h1>LogisticGroup</h1>
    <span class="subheading">Ви легко знайдете роботу з нами!</span>
    </div>
    </div>
    </div>
</header>
<!-- Main Content-->
<div class="container px-4 px-lg-5">
    <div class="row gx-4 gx-lg-5 justify-content-center">
        <div class="col-md-12 col-lg-12 col-xl-7">
            <!-- Post preview-->
            <h1 class="text-center">Чому ми?</h1>
            <div class="post-preview">
                <ul class="mt-4">
                    <li>Ми гарантовано знайдемо замовлення, ви не
залишитеся без роботи</li>
                    <li>Вантаж застраховано</li>
                    <li>Гарантована амортизація автомобіля</li>
                    <li>Оплачуване житло</li>
                    <li>Все в ваших руках! Ви можете прийняти зручні для Вас замовлення в своєму особистому
кабінеті.</li>
                </ul>
            </div>
        </div>
    </div>
</div>
@endsection

```

Файл *home.blade.php*

```

@extends('layouts.app')
@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header">{{
__('Dashboard')
}}</div>

role="alert">
<div class="card-body">
    @if (session('status'))
        <div class="alert alert-success"

            {{ session('status') }}

        </div>
    @endif

    {{ ('You are logged in!') }}
</div>

        </div>
    </div>
</div>
</div>
@endsection

```