

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: “ Розробка веб-додатку для планування логістики ”

Виконав: студент гр. Кн-41сп _____

Спеціальності 122 – «Комп’ютерні науки»
(шифр і назва)

Винник Дмитро Володимирович
(Прізвище та ініціали)

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

Рецензенти: к.т.н., доц. Бабич М.І.
(Прізвище та ініціали)

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти

122 – «Комп'ютерні науки»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Виннику Дмитру Володимировичу

1. Тема роботи: «Розробка веб-додатку для планування логістики»

Керівник роботи Луб Павло Миронович, к.т.н., доцент
Затверджені наказом університету 27.11.2023 року №641/к-с.

2. Строк подання студентом роботи 14.06.2024 р.

3. Початкові дані до роботи: 1. Сайти-аналоги планування маршрутів перевезень. 2. Методика побудови контекстних діаграм. 3. Методика побудови веб-додатків. 4. Науково-технічна і довідкова література.

4. Зміст розрахунково-пояснювальної записки:

1. Аналіз ІТ-сервісів планування перевезень та інструментарію для їх створення.
2. Технологія побудови проекту веб-додатку.
3. Проектування веб-додатку планування маршрутів вантажних перевезень.
4. Охорона праці та безпека в надзвичайних ситуаціях.
5. Висновки.
6. Бібліографічний список.
7. Додатки

5. Перелік презентаційного матеріалу : 1 та 2 – Тема, мета, завдання роботи; 3 – Головні визначення та класифікація веб-сторінок; 4 – Аналіз веб-сайтів та платформ; 5 – Програмні засоби для розробки веб-додатків; 6 – Структурно-функціональне моделювання та варіанти використання веб-додатку; 7 – Архітектура веб-додатку та структура папок; 8 – Розробка структури бази даних; 9 – Розробка ядра веб-додатку; 10 – Елементи роботи веб-сторінки.

6. Консультанти з розділів:

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

7. Дата видачі завдання 27.11.2023 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>27.11.2023 – 01.01.2024</i>	
2.	<i>Виконання другого розділу та формування головних показників для розрахунків</i>	<i>01.01.2024 – 01.02.2024</i>	
3.	<i>Виконання третього розділу, розрахунків та розробка листів</i>	<i>01.02.2024 – 01.03.2024</i>	
4.	<i>Написання розділу: «Охорона праці та безпека в надзвичайних ситуаціях»</i>	<i>01.03.2024 – 01.04.2024</i>	
5.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів графічної частини</i>	<i>01.05.2024 – 01.06.2024</i>	
6.	<i>Завершення роботи в цілому</i>	<i>01.06.2024 – 14.06.2024</i>	

Студент _____ Винник Д.В.
(підпис)

Керівник роботи _____ Луб П.М.
(підпис)

УДК: 004.94: 631.1

Розробка веб-додатку для планування логістики. Винник Д.В. Кафедра ІТ.
– Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 57 с. текст. част., 34 рис., 3 табл., 10 слайдів, 19 джерел.

Проаналізовано базові поняття розробки веб-сервісів та наведено класифікацію веб-сторінок. Виконано аналіз технологій формування перевезень вантажів та програмних засобів для розробки прикладних програм.

Описано технологію структурного аналізу та проектування ІС.

Виконано структурно-функціональне моделювання та варіанти використання веб-додатку.

Розроблено архітектуру веб-додатку та наведено структуру папок.

Реалізовано базу даних веб-додатку та виконано розробку її структури.

Розроблено ядро веб-додатку, наведено будову програмних модулів та представлено елементи роботи веб-сторінки в браузері.

Розроблено заходи з охорони праці та безпека в надзвичайних ситуаціях.

Ключові слова: веб-додаток, перевезення вантажів, планування, архітектура, бази даних, ядро програми, модулі, браузер.

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

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1	
АНАЛІЗ ІТ-СЕРВІСІВ ПЛАНУВАННЯ ПЕРЕВЕЗЕНЬ ТА ІНСТРУМЕНТАРІЮ ДЛЯ ЇХ СТВОРЕННЯ	7
1.1. Базові поняття розробки веб-сервісів та класифікація веб-сторінок.....	7
1.2. Аналіз технологій формування перевезень вантажів	10
1.3. Аналіз програмних засобів для розробки веб-сторінок .	15
РОЗДІЛ 2	
ТЕХНОЛОГІЯ ПОБУДОВИ ПРОЕКТУ ВЕБ-ДОДАТКУ ПЛАНУВАННЯ МАРШРУТІВ ВАНТАЖНИХ ПЕРЕВЕЗЕНЬ.....	19
2.1. Технологія структурного аналізу та проектування ІС.....	19
2.2. Структурно-функціональне моделювання та варіанти використання веб-додатку.....	21
2.3. Архітектура веб-додатку та структура папок.....	24
2.4. Реалізація бази даних веб-додатку.....	27
2.5. Розробка структури бази даних.....	28
РОЗДІЛ 3	
ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ПЛАНУВАННЯ МАРШРУТІВ ВАНТАЖНИХ ПЕРЕВЕЗЕНЬ.....	31
3.1. Розробка ядра веб-додатку	31
3.2. Розробка структури програмних модулів <i>веб-додатку</i>	33
3.3. Елементи роботи веб-сторінки в браузері.....	36
РОЗДІЛ 4.	
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ...	41
4.1. Структурно-функціональний аналіз технологічного процесу...	41
4.2. Розрахунок освітлення приміщення комп'ютерного кабінету...	42
4.3. Безпека в надзвичайних ситуаціях.....	45
ВИСНОВКИ І РЕКОМЕНДАЦІЇ.....	46
БІБЛІОГРАФІЧНИЙ СПИСОК.....	48
ДОДАТКИ.....	50

ВСТУП

Автоматизація є основною передумовою для розвитку перевезень. Із запровадженням нових операційних моделей логістика сьогодні стала набагато складнішою. Замовники очікують швидшого обслуговування та більшої прозорості. Тому, навіть малі та середні транспортні підприємства звертають свою увагу на ІТ-рішення, сподіваючись мінімізувати ручне керування, паперову роботу та складнощі. Водночас, запит ринку споживачів спрямований на те, щоб інструменти ставали розумнішими, щоб бізнес-додатки і сервіси ставали “смарт”. Те ж стосується і логістичних сервісів. За цим стоять цілком конкретні речі: електронний документообіг, електронний кабінет логіста та інші зміни логістичних бізнес-процесів.

Розумна логістика – це оптимізація інформаційних логістичних потоків, але вже – обов’язково в електронному вигляді, з дедалі більшим відходом від ручного керування в бік автоматизації, за допомогою ІТ-сервісів – і вже на більш глибокому рівні алгоритмізації процесів.

Мета роботи – створення веб-додатку планування маршрутів транспортних перевезень із мінімальним функціоналом та інтуїтивним інтерфейсом зручним для людей що малознайомі із сучасними технологіями.

Завдання кваліфікаційної роботи:

- проаналізувати ІТ-сервіси планування перевезень та інструментарій для їх створення;
- розкрити технологію побудови проекту веб-додатку;
- виконати проектування веб-додатку планування маршрутів вантажних перевезень.

РОЗДІЛ 1

АНАЛІЗ ІТ-СЕРВІСІВ ПЛАНУВАННЯ ПЕРЕВЕЗЕНЬ ТА ІНСТРУМЕНТАРІЮ ДЛЯ ЇХ СТВОРЕННЯ

1.1. Базові поняття розробки веб-сервісів та класифікація веб-сторінок

Веб-сторінка – це інформаційний ресурс доступний в мережі World Wide Web (Всесвітня павутина), який можна переглянути у веб-браузері зазвичай, ця інформація записана в форматі HTML або XHTML, і може містити гіпертекст з навігаційними гіперпосиланнями на інші веб-сторінки.

Гіпертекст – це текст для перегляду на комп'ютері, який містить зв'язки з іншими документами («гіперзв'язки» чи «гіперпосилання»); читач має змогу перейти до пов'язаних документів безпосередньо з вихідного (первинного) тексту, активізувавши посилання. Загальновідомим та яскраво вираженим прикладом сучасного гіпертексту слугують веб-сторінки, підготовлені за допомогою HTML (мова розмітки гіпертексту) й розміщені у web-мережі. Відповідно, в комп'ютерній термінології під гіпертекстом розуміється текст, сформований за допомогою мови розмітки, що потенційно містить у собі гіперпосилання. Гіперпосилання дозволяють переходити від вихідного (одного) тексту до безлічі інших текстів, розміщених у web-мережі.

Елементи веб-сторінок:

Контент – змістове наповнення веб-сторінки, доступне користувачеві:

- 1) за формою подачі: текст, зображення, аудіо, відео;
- 2) за доступністю: безкоштовний, платний;
- 3) за унікальністю: унікальний, неунікальний.

Елементи навігації – засоби для переходу на інші веб-сторінки.

Елементи дизайну – елементи структурування контенту та його форматування, оформлення сторінки.

Додаткові елементи веб-сторінок:

Банер – це невеличке рекламне зображення, що розміщується на одній сторінці, а посилається на іншу.

Форма – частина веб-сторінки, до якої користувач може записувати свої відомості та надсилати їх на сервер, де розміщується веб-сторінка.

Веб-сайт – сукупність веб-сторінок, доступних у Інтернеті, які об'єднані як за змістом, так і навігаційно.

Сайтом також називають вузол мережі Інтернет, комп'ютер, за яким закріплена унікальна IP-адреса, і взагалі будь-який об'єкт в Інтернеті, за яким закріплена адреса, що ідентифікує його в мережі (FTP-site, WWW-site тощо).

Структура сайту

- Внутрішня – дозволяє всі необхідні дані розмістити зручно і доступно в розділах та підрозділах.

- Зовнішня – визначає спосіб розміщення відомостей на сторінці при виборі кожного пункту меню, визначає дизайн сайту.

Внутрішня структура сайту

- Лінійна (послідовна) – доцільно використовувати у разі послідовного подання даних та відомостей; перегляд послідовний: від початкової до останньої сторінки сайту.

- Ієрархічна – одна сторінка головна, яка не має попередніх, решта сторінок мають лише одну попередню сторінку.

Довільна – кожна сторінка може містити посилання на довільну кількість сторінок сайту.

- Гібридна – поєднання ієрархічної і послідовної структур.

- Мережна – базується на побудові системи такої навігації сайту, коли між горизонтальними та вертикальними елементами існує взаємний зв'язок та можливості швидкого переходу з однієї сторінки на іншу без додаткової необхідності відвідування проміжних сторінок.

Головна сторінка сайту має задовольняти правило “3 по 3”, тобто 3 абзаци по 3 речення про тематику, зміст, авторів та призначення сайту. Інколи замість головної сторінки з'являється заставка – велике фото зі звуковим

супроводом або відеороликом. Такі сторінки створюють, що привернути увагу користувачів.



Рисунок 1.1 – Класифікація веб-сторінок

Основні вимоги до веб-сайтів:

- Наявність головної сторінки зі стислим каталогом усіх розділів (рубрик) сайту. З кожної сторінки сайту мають бути посилання на головну сторінку.
- Наявність швидкої і зручної навігації та пошуку.
- Наявність на головній сторінці сайту стислої інформації для відвідувачів сайту про його структуру і змістове наповнення.
- Оновлення сторінок сайту.
- За наявності великої кількості текстової інформації на сайті доречно розміщувати окрім веб-сторінок з текстами також архіви у форматі **Zip** чи **Rar**.

Сторінки сайту мають бути зручними для користувачів сайту, а також для роботи з ними.

- Інформація на сайті може бути представлена українською, російською, англійською мовами.

- Сайт повинен містити засоби зворотнього зв'язку.

Оцінювання веб-сайтів:

- Ступінь інформативності;
- Частота оновлення даних на сайті;
- Наявність інтерактивних засобів;
- Доступні мови для відтворення веб-сайту;
- Візуальна привабливість сайту;
- Зручність користування;
- Зрозумілість для користувача.

Таким чином, багато сайтів на якийсь спосіб займаються заробітком, через розміщення реклами або надання платних послуг користувачам. Однак, вони були створені спочатку не для прибутку, а для задоволення потреб користувачів, тому їх можна вважати некомерційними. Але в таких випадках границя між комерційними і некомерційними сайтами стає дещо нечіткою, і багато з цих сайтів можна назвати "умовно некомерційними".

1.2. Аналіз технологій формування перевезень вантажів

З метою забезпечення мінімізації часу та інтеграції інтересів замовників і перевізників, корисно використовувати інформаційні логістичні платформи (онлайн біржі перевезень вантажів ОБП), які дозволять у режимі онлайн, знаходити вільний транспорт для перевезення потрібного вантажу між різними країнами та спростують пошук замовників за допомогою сайту біржі.

В сучасних умовах життя є можливість спостерігати за транспортуванням вантажу в реальному часі через Інтернет з будь-яких пристроїв за допомогою маячків GPSTрекерів, або софт-трекерів, що у свою чергу є гарантією того, що перевізник буде дотримуватися визначеного маршруту. З метою проведення

аналізу онлайн-ових бірж перевезень вантажів 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 дає можливість ознайомитися із загальною інформацією щодо перевезень в середині країни та відстежує останні новини пов'язані із транспортною логістикою як у Польщі, так і Європейському союзу. 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.

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.

BursaTransport (Румунія). Це Інтернет-платформа, яка призначена для операторів автомобільних вантажних перевезень (перевізників) і виробників, які мають потребу у транспортних послугах 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) - є 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 веб-сайтів, які надають послуги планування маршрутів в Україні то їх є також достатня кількість. Проаналізуємо найвідоміші з них.

Платформа Lardi-Trans це набір інструментів для професіоналів у сфері вантажоперевезень. Пошук вантажу та транспорту, страхування, керування автопарком, електронний документообіг та проведення тендерів (рис. 1.2.).

Платформа Lardi-trans має універсальний інструмент пошуку вантажу, широкий вибір параметрів дозволяє відбирати найвигідніші пропозиції. Функція Автопошуку допоможе не пропустити вигідні пропозиції щодо вантажів та надішле повідомлення до браузеру, або Telegram. Інтеграція з Telegram дозволяє оперативно отримувати сповіщення на будь-які пристрої.

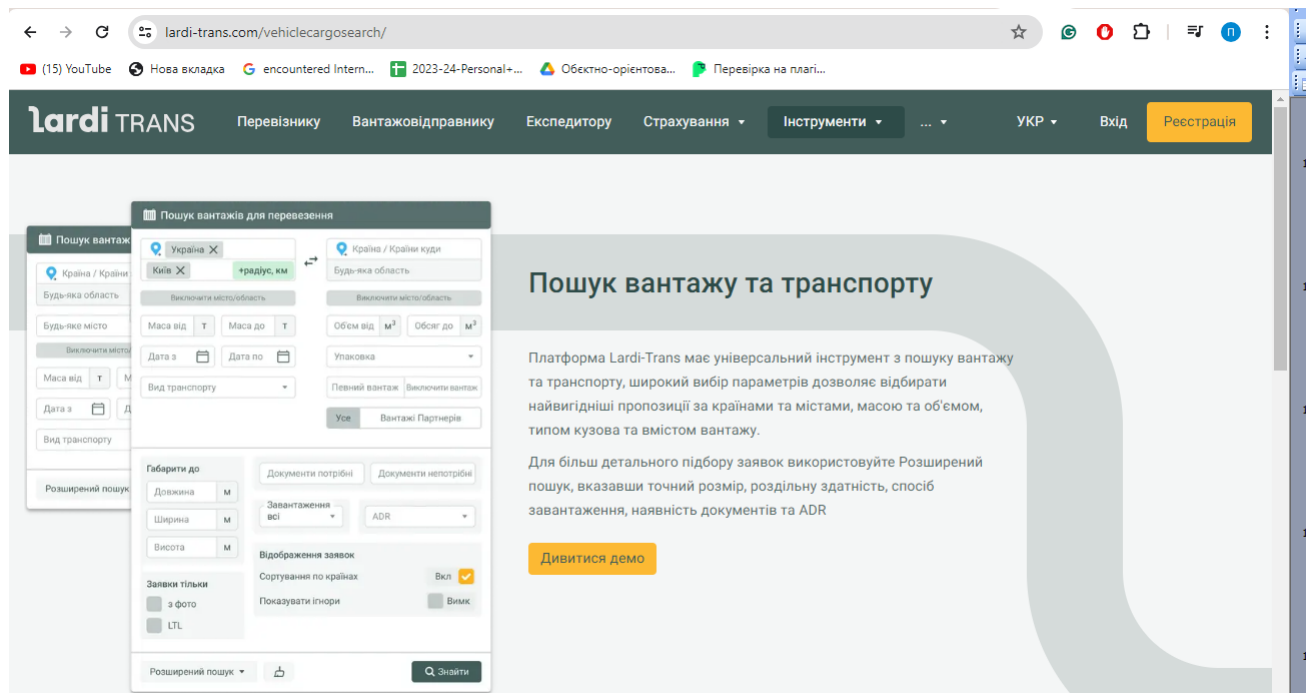


Рисунок 1.2. – Головне вікно пошуку вантажевих перевезень Lardi-trans [5]

«Ukrainian Logistic» - найкраща транспортна компанія згідно з Національним рейтингом якості товарів і послуг «Вибір Країни» 2018, 2019, 2020. 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. 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.

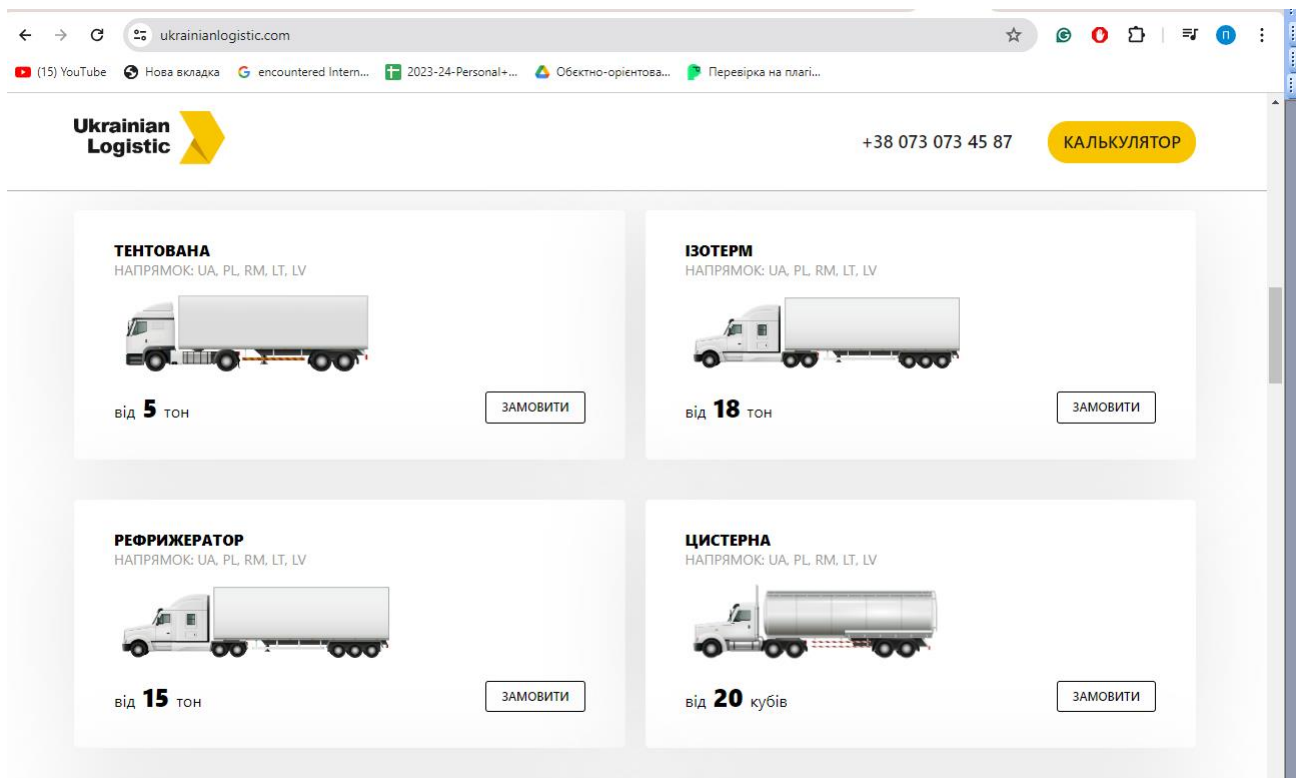


Рисунок 1.3. – Головне вікно веб-сторінки Ukrainian Logistic [2]

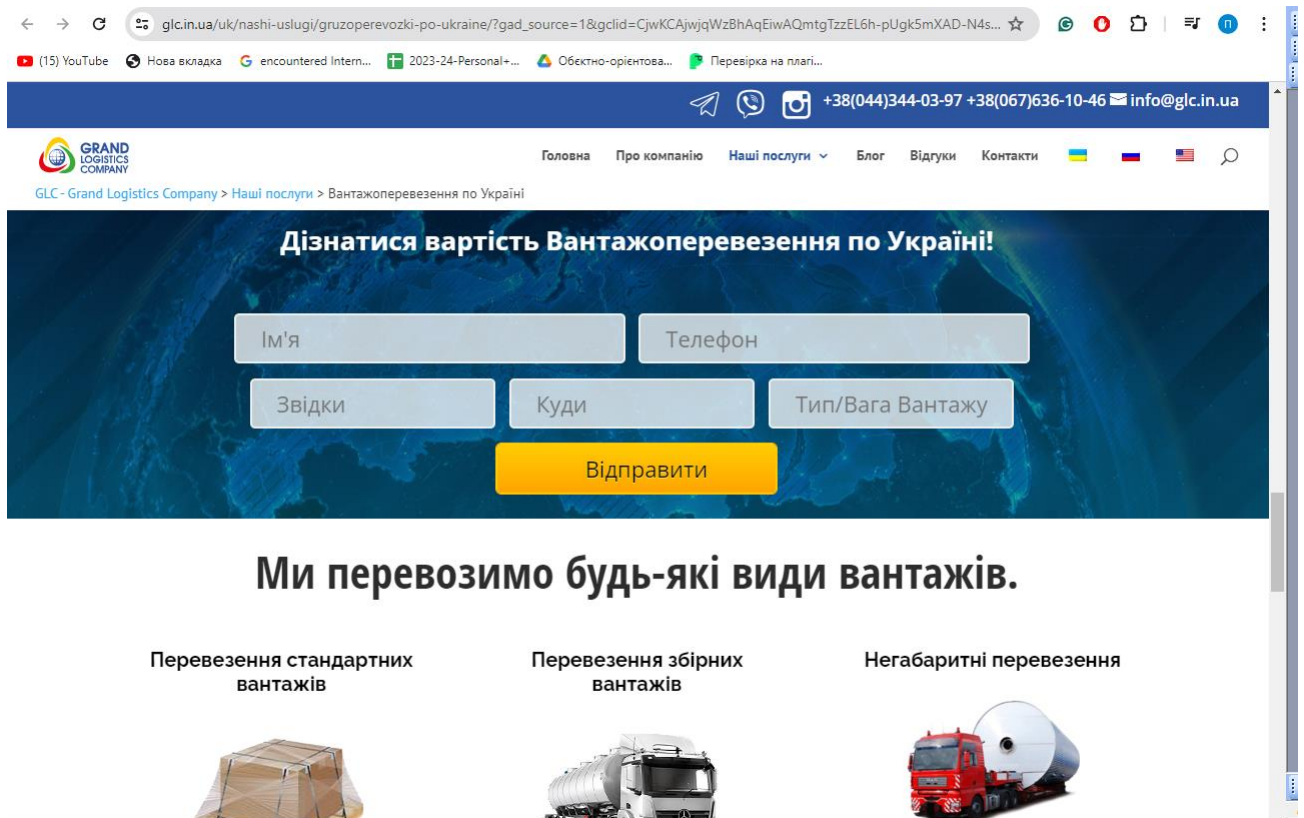
Компанія DELLA™ Вантажні перевезення успішно працює на ринку вантажних автоперевезень з 1995 р. До послуг перевізників - 20-річний досвід організації внутрішніх та міжнародних вантажних перевезень. Замовнику пропонується необхідна кількість пропозицій із перевезення вантажу від різних перевізників, та щоразу обирається тільки найбільш вигідний варіант. 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.

The screenshot shows the Della website interface. At the top, there is a navigation bar with the Della logo and the text 'Автоперевезення'. Below this, there are links for 'ДОДАТИ ВАНТАЖ', 'ДОДАТИ ТРАНСПОРТ', and 'ПОШУК ВАНТАЖІВ І ТРАНСПОРТУ'. A central banner features a petition titled 'Петиція про бронювання водіїв міжнародних перевезень' with a 'Підтримати' button. Below the banner, there are tabs for 'Вантажі Україна', 'Міжнародні Вантажі', 'Транспорт України', and 'Міжнародний Транспорт'. The main content area displays a table of cargo listings for Ukraine-Ukraine routes, including details like dates, routes (e.g., Южне (UA) — Запоріжжя (UA)), weights, and volumes. To the right of the table, there are several utility widgets: 'Потрібен перевізник?' with a call to action, 'Курси валют України' showing USD at 40,52 грн and EUR at 43,46 грн, and 'Пошук попутних вантажів' with dropdown menus for origin and destination. At the bottom of the main content area, there is a line chart titled 'Вантажні перевезення Україна. Статистика цін' showing price trends for 20-ton cargo from June to April.

Рисунок 1.4. – Головне вікно веб-сторінки Компанії DELLA™ [1]

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.



gic.in.ua/uk/nashi-uslugi/gruzoperevozki-po-ukraine/?gad_source=1&gclid=CjwKCAjwjqWzBhAqEiwAQmtgTzzEL6h-pUgk5mXAD-N4s...

Головна Про компанію Наші послуги Блог Відгуки Контакти

GLC - Grand Logistics Company > [Наші послуги](#) > Вантажоперевезення по Україні

Дізнатися вартість Вантажоперевезення по Україні!

Ім'я Телефон

Звідки Куди Тип/Вага Вантажу

Відправити

Ми перевозимо будь-які види вантажів.

Перевезення стандартних вантажів

Перевезення збірних вантажів

Негабаритні перевезення

Рисунок 1.5. – Головне вікно веб-сторінки Ukrainian Logistic [14]

Таким чином, сьогодні існують різноманітні сервіси що надають послуги із планування вантажних перевезень. 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 програмних засобів для розробки веб-сторінок

Під час розробки, для реалізації додатку використовуються різні програмні засоби. Розглянемо весь перелік програмних засобів які доцільно використати для нашої розробки (рис. 1.6), 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.



Рисунок 1.6 – Базові Інструменти розробки веб-додатків

Базовим інструментом для розробки доцільно обрати один із найпопулярніших фреймворків для розробки сайтів та веб-додатків [11] (рис. 1.7). 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 проекті, надає можливість швидкого доступу до них [13].

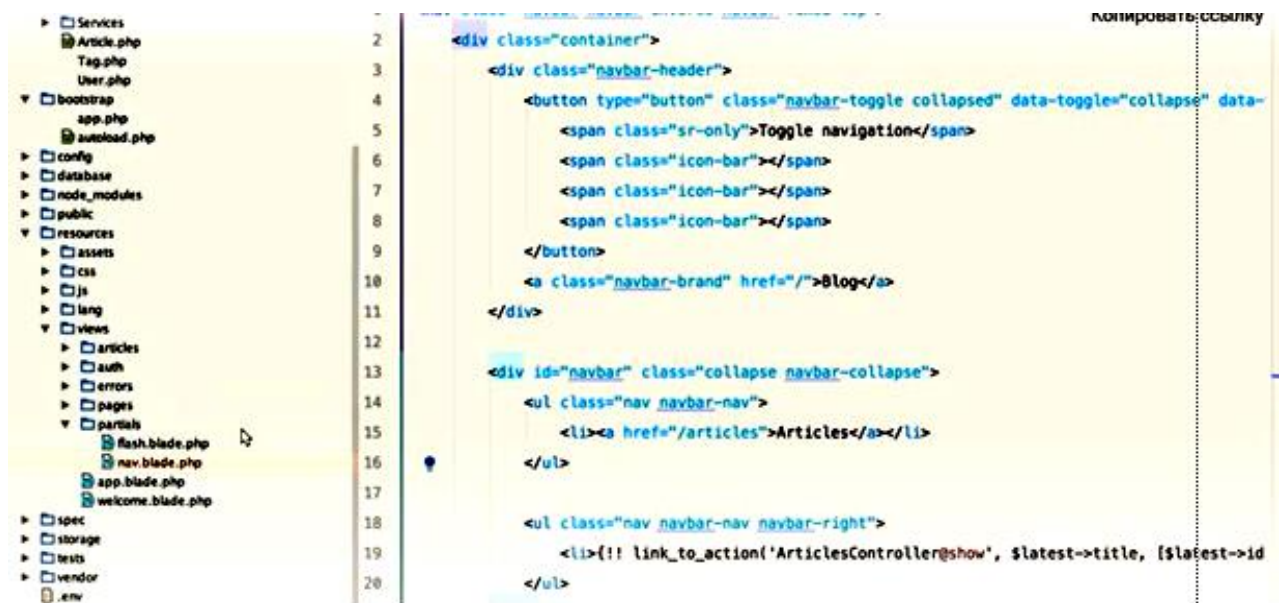


Рисунок 1.7 – Фреймворк Laravel

Для роботи із файлами в проекті доцільно застосувати текстовий редактор Notepad++. Він оснащений основними функціями та додатковими плагінами необхідних для розробки великих комерційних проектів (рис. 1.8). Дає змогу працювати з кодом відповідно до потреб користувача [17].


```

157  /**
158  * Indicates if the connection is in a "dry run".
159  *
160  * @var bool
161  */
162  protected $pretending = false;
163
164  /**
165  * All of the callbacks that should be invoked before a query is executed.
166  *
167  * @var array
168  */
169  protected $beforeExecutingCallbacks = [];
170
171  /**
172  * The instance of Doctrine connection.
173  *
174  * @var \Doctrine\DBAL\Connection
175  */
176  protected $doctrineConnection;
177
178  /**
179  * The connection resolvers.
180  *
181  * @var array
182  */
183  protected static $resolvers = [];
184
185  /**
186  * Create a new database connection instance.
187  *
188  * @param \PDO\Closure $pdo
189  * @param string $database
190  * @param string $tablePrefix
191  * @param array $config
192  * @return void
193  */
194  public function __construct($pdo, $database = '', $tablePrefix = '', array $config = [])
195  {
196      $this->pdo = $pdo;
197
198      // First we will setup the default properties. We keep track of the DB
199      // name we are connected to since it is needed when some reflective
200      // type commands are run such as checking whether a table exists.

```

Рисунок 1.8 – Notepad++

Можливість відтворення додатку на емуляції локального веб-серверу надає програмне забезпечення OpenServer, яке не менш поширене того ж Notepad, 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. 1.9).

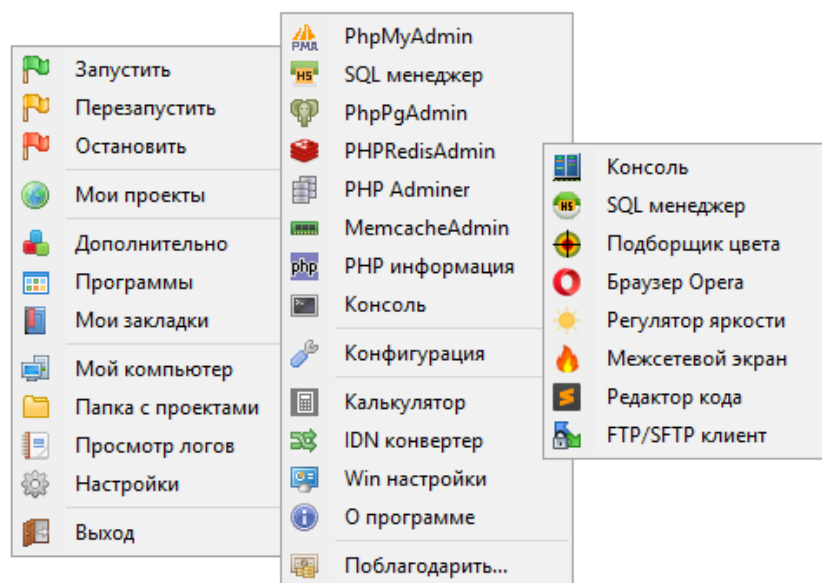


Рисунок 1.9 – Меню OpenServer

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

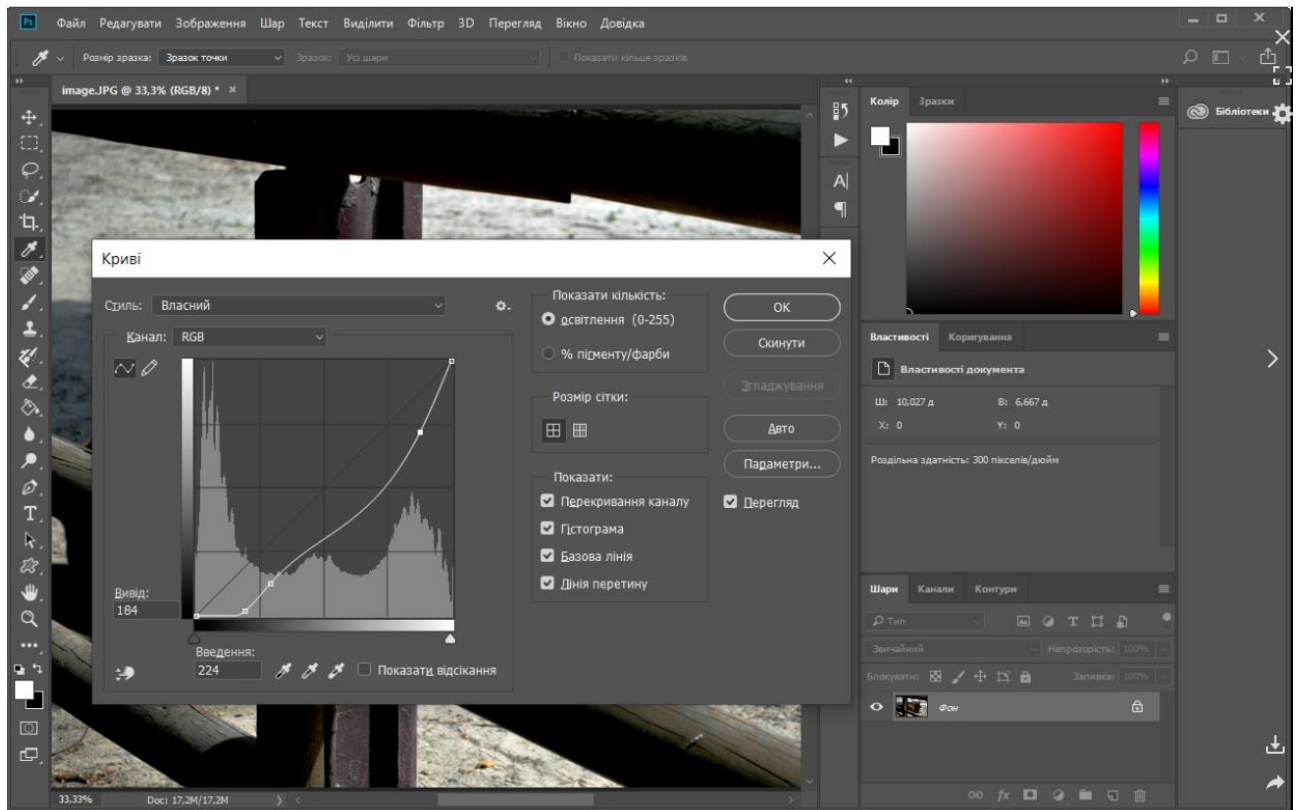


Рисунок 1.10 – Головне вікно Adobe Photoshop

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

РОЗДІЛ 2

ТЕХНОЛОГІЯ ПОБУДОВИ ПРОЕКТУ ВЕБ-ДОДАТКУ ПЛАНУВАННЯ МАРШРУТІВ ВАНТАЖНИХ ПЕРЕВЕЗЕНЬ

2.1. Технологія структурного аналізу та проектування ІС

В методології структурного аналізу традиційно використовують моделі, що показують:

- функції, які система повинна виконувати;
- процеси, що забезпечують виконання функцій;
- дані, що необхідні під час виконання функцій, і відношення між ними;
- організаційні структури, що забезпечують виконання функцій;
- матеріальні та інформаційні потоки, що виникають у ході виконання функцій.

Як інструментальні засоби структурного аналізу і проектування найбільш часто і ефективно застосовуються наступні.

BFD (Business Function Diagrams) – діаграми бізнес-функцій (функціональні специфікації).

DFD (Data Flow Diagrams) – діаграми потоків даних у нотаціях Гейна – Сарсона, Йордана та інших, що забезпечують аналіз і функціональне проектування інформаційних систем.

ERD (Entity-Relationship Diagrams) – діаграми "сутність – зв'язок" у нотаціях Чена и Баркера.

STD (State Transition Diagrams) – діаграми переходів станів, які засновані на розширеннях Уорда – Меллора (Ward – Mellor) і Хатлі (Hatley) для проектування систем реального часу.

SSD (System Structure Diagrams) – діаграми структури програмного додатка.

FDD (Functional Decomposition Diagrams) – діаграми функціональної декомпозиції.

Сімейство **IDEF** (*Integration Definition for Function Modeling*):

IDEF0 – методологія функціонального моделювання, що є складовою частиною SADT і що дозволяє описати бізнес-процес у вигляді ієрархічної системи взаємопов'язаних функцій.

IDEF1 – методологія аналізу і вивчення взаємозв'язків між інформаційними потоками в рамках комерційної діяльності підприємства.

IDEF1X – методологія інформаційного моделювання, заснована на концепції "сутність – зв'язок" Ченом. Застосовується для розробки реляційних баз даних і використовує умовний синтаксис, що спеціально розроблений для зручної побудови концептуальної схеми і забезпечує універсальне подання структури даних в рамках підприємства, незалежне від кінцевої реалізації бази даних і апаратної платформи;

IDEF3 – методологія моделювання потоків робіт підприємства, що дозволяє подати їх сценарії за допомогою опису послідовності змін властивостей об'єкта в рамках даного процесу.

IDEF4 – методологія об'єктно-орієнтованого проектування для підтримки проектів, пов'язаних з об'єктно-орієнтованими реалізаціями.

IDEF5 – методологія, що забезпечує наочне подання даних, отриманих у результаті обробки онтологічних запитів, у простій, графічній формі.

Усі перелічені засоби містять графічні і текстові засоби моделювання: перші – для зручності відображення основних компонентів моделі, другі – для забезпечення точного визначення її компонентів і зв'язків.

Методології структурного аналізу і проектування можна класифікувати наступним чином (рис. 6.1).

За відношенням до шкіл виділяють методології програмної інженерії (*Software Engineering, SE*) та інформаційної інженерії (*Information Engineering, IE*).

SE є нисхідним поетапним підходом до розробки ПЗ, що починається із загального погляду на його функціонування. Потім проводиться декомпозиція на підфункції. Процес повторюється доти, поки підфункції не стануть

достатньо малими для можливості їх кодування. У результаті виходить ієрархічна, структурована модульна програма. SE подає універсальні методології розробки ПЗ.

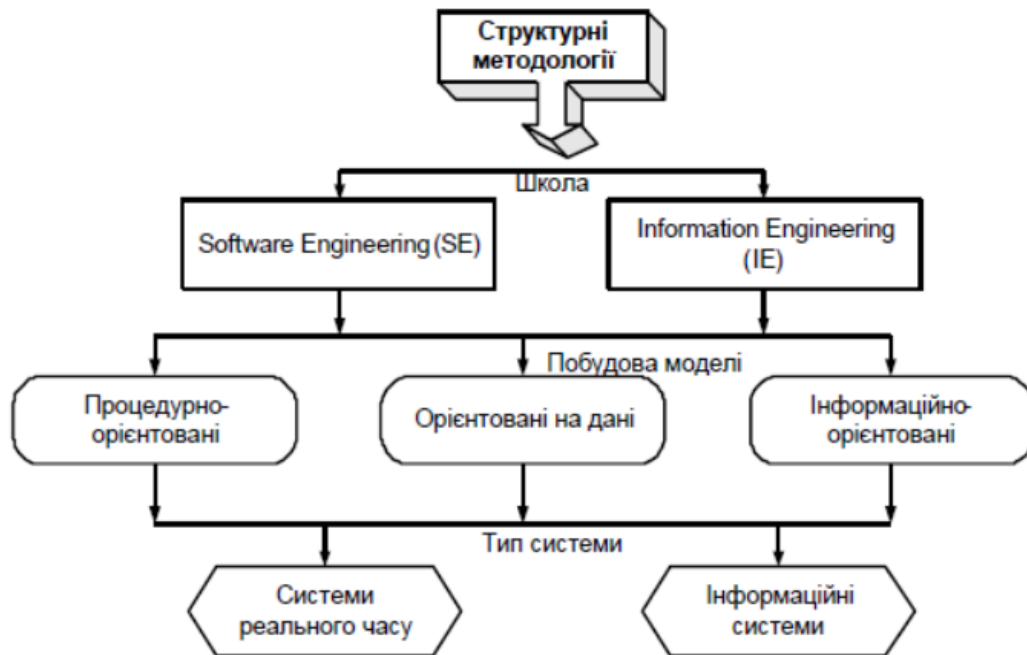


Рис. 2.1. – Класифікація структурних методологій

ІЕ представляє нові методології побудови систем взагалі і включає етапи більш високого рівня (наприклад, стратегічне планування). На етапі проектування систем за методологією SE і ІЕ аналогічні.

2.2. Структурно-функціональне моделювання та варіанти використання веб-додатку

Для опису основних процесів використано методологію функціонального моделювання, яке більш відоме як методологія IDEF0 [15]. 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 2.2-2.3.

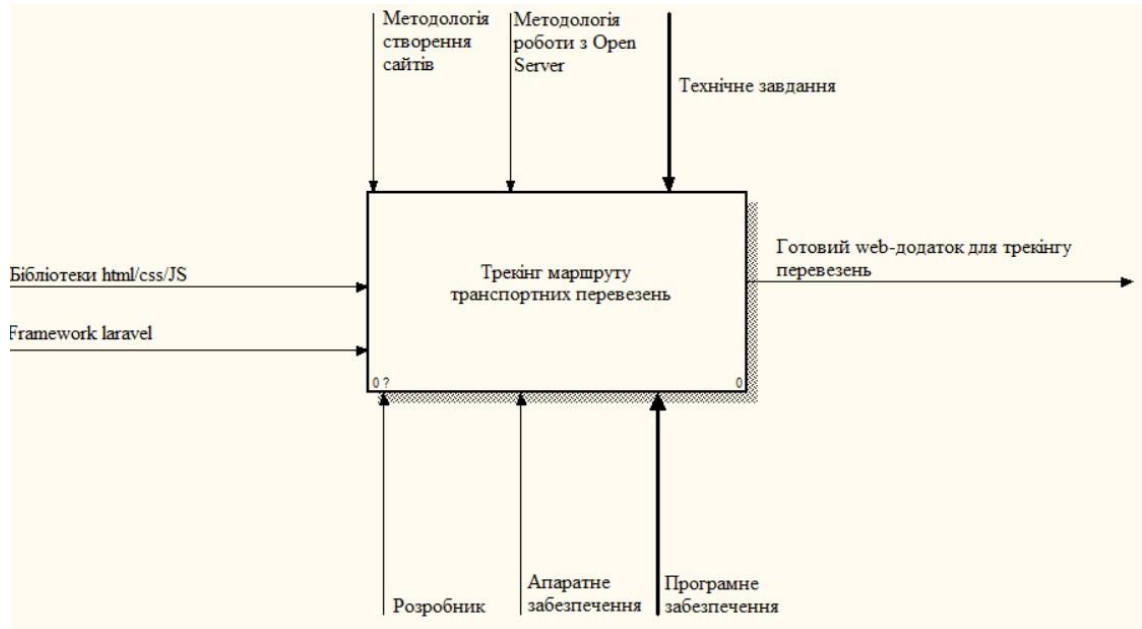


Рисунок 2.2 – Контекстна діаграма веб-сервісу

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. 2.3).

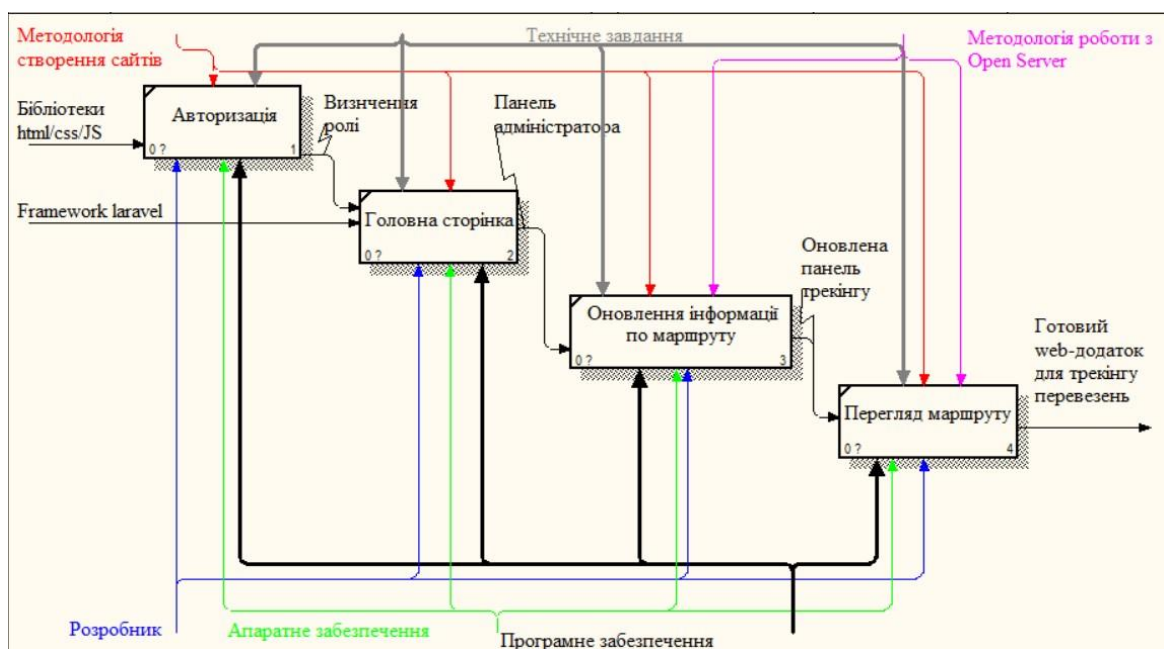


Рисунок 2.3 – Діаграма декомпозицій веб-сервісу

Моделювання варіантів використання. Завершальним етапом проектування веб-додатку є розробка діаграми класів. Така діаграма використовується для опису роботи веб-додатку, відображення користувачів та їх діяльності [15]. Use Case діаграму наведено на рисунку 2.4.

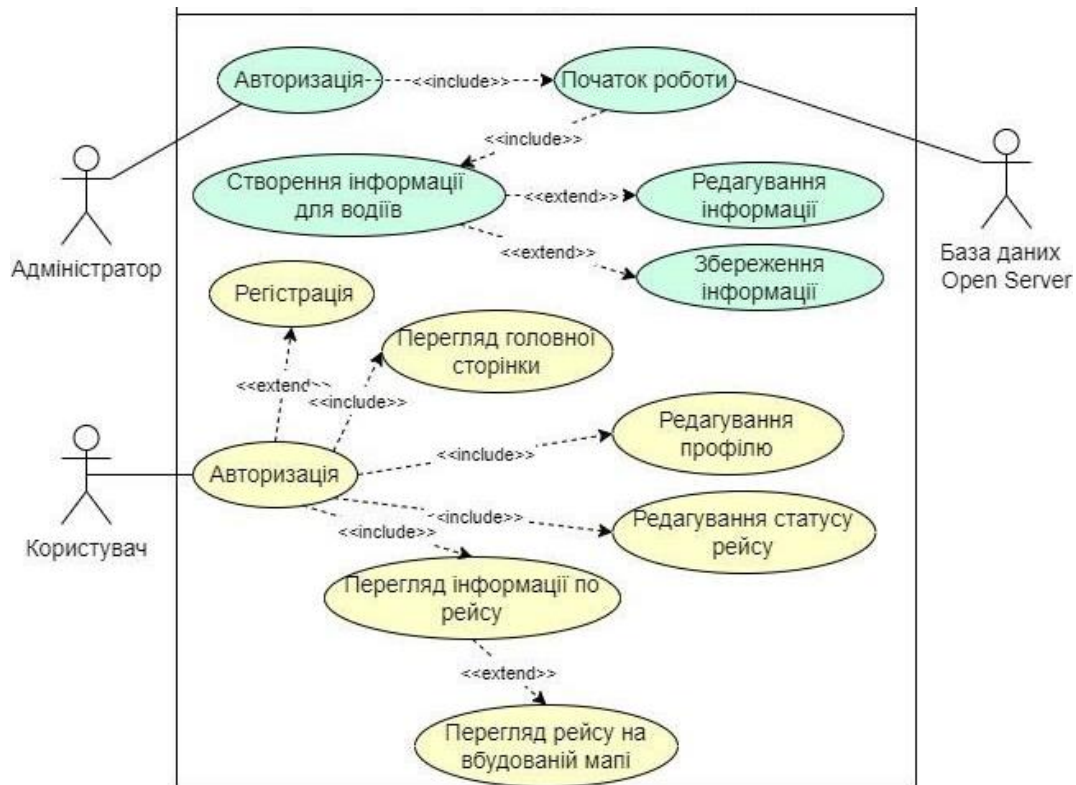


Рисунок 2.4 – Діаграма сценаріїв використання веб-додатку планування маршрутів

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.

2.3. Архітектура веб-додатку та структура папок

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 палени адміністрування сприяє легшій роботі.

Для отримання представлення програмної частини та структури сторінок додатку запропоновано його архітектуру (рис. 2.5).

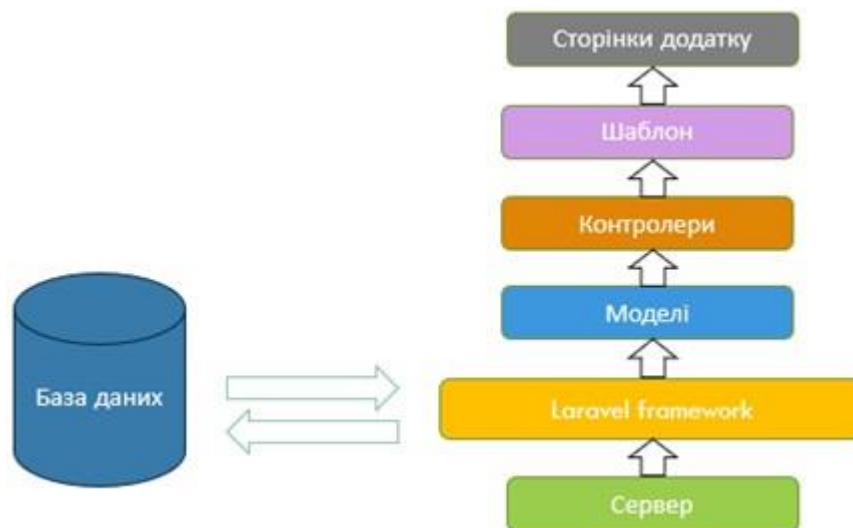


Рисунок 2.5 – Архітектура веб-додатку побудови маршрутів

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

інформація. 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. 2.6). 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.

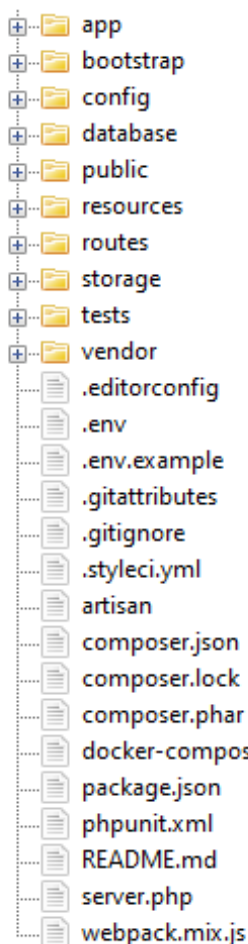


Рисунок 2.6 – Модель розробки додатків MVC

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 користувальницький 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 2.7.



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.

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.

Рисунок 2.7 –
Структура папок

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.

2.4. Реалізація бази даних веб-додатку

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 2.1 – Характеристика об'єктів бази даних

Назва об'єкту	Характеристика об'єкта
1	2
<code>failed_jobs</code>	Зберігає у собі інформацію про не виконані рейси
<code>flights</code>	Зберігає у собі інформацію про рейси

1	2
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-

Таблиця 2.2 – 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

Опис інших атрибутів бази даних веб-додатку наведено в додатку А.

2.5. Розробка структури 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).

Table Name	Field Name	Data Type	Constraints
logistic flights	id	bigint(20)	unsigned, PK
	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)	FK
status_id	int(11)	FK	
created_at	timestamp		
updated_at	timestamp		
logistic migrations	id	int(10)	unsigned, PK
	migration	varchar(255)	
	batch	int(11)	
logistic statuses	id	bigint(20)	unsigned, PK
	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, PK
	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, PK
	tokenable_type	varchar(255)	
	tokenable_id	bigint(20)	unsigned, FK
	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, PK
	uuid	varchar(255)	
	connection	text	
	queue	text	
	payload	longtext	
	exception	longtext	
failed_at	timestamp		

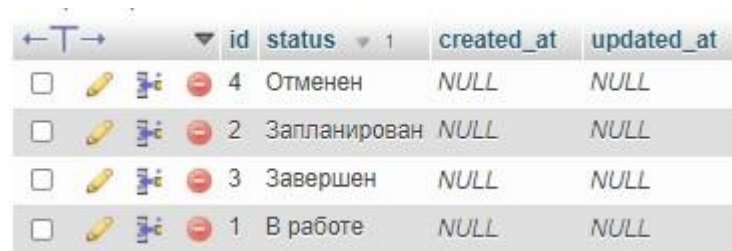
Рисунок 2.8 – Схема бази даних веб-додатку

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 2.9 – Зміст таблиці **migrations**

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

Рисунок 2.10 – Зміст таблиці **personal_access_tokens**



	id	status	created_at	updated_at
<input type="checkbox"/>	4	Отменен	NULL	NULL
<input type="checkbox"/>	2	Запланирован	NULL	NULL
<input type="checkbox"/>	3	Завершен	NULL	NULL
<input type="checkbox"/>	1	В работе	NULL	NULL

Рисунок 2.11 – 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 2.8.

Дамп бази даних наведений у додатку Б.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ПЛАНУВАННЯ МАРШРУТІВ ВАНТАЖНИХ ПЕРЕВЕЗЕНЬ

3.1. Розробка ядра веб-додатку

Першочергово слід описати конфігурацію запроектованого додатку.

Отже, файл конфігурації додатку створюється автоматично фреймворком та знаходиться у корені проекту із назвою `.env` (рис. 3.1).

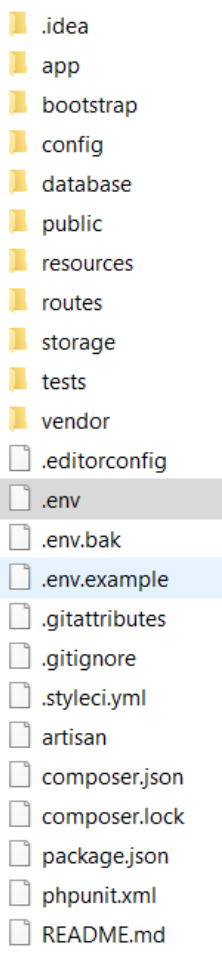
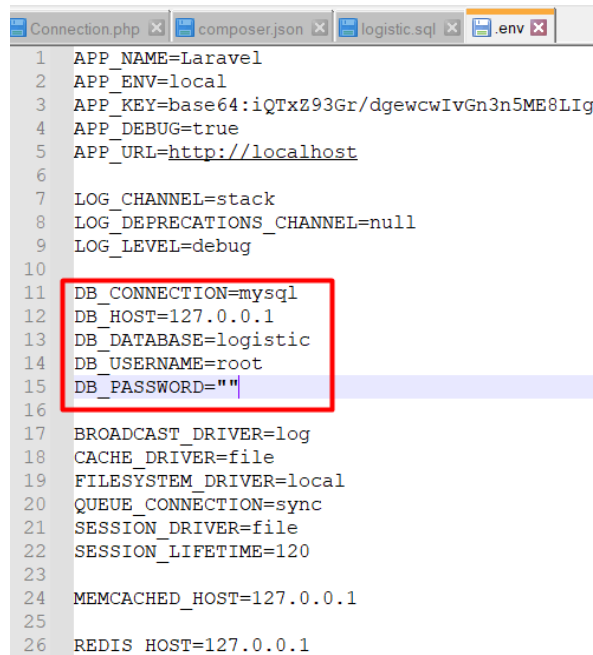


Рисунок 3.1 –
Файл конфігурації
`.env`

У цьому ж файлі 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. 3.2).

Головний клас додатку. 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 3.2 – Фрагмент файлу з підключенням до бази даних

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

В цілому, контролери:

- можуть мати доступ до даних запиту;
- можуть викликати методи моделей та інших компонентів системи з даними запиту;
- можуть використовувати уявлення для формування відповіді;
- не повинні займатися обробкою даних, це повинно відбуватися в шарі моделей;
- повинні уникати використання HTML, або інший розмітки, краще це робити в уявленнях.

Простори імен. 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. 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 ви можете 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.

Наприклад, 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`.

3.2. Розробка структури програмних модулів веб-додатку

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. 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.

Програмні модулі проекту. 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. Структуру модулів проекту зображено на рисунку 3.3.

Имя

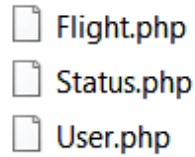


Рисунок 3.3 – Архітектура компоненту Models

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

```

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 Prepp length : 626 lines : 33 Ln : 23 Col : 1 Pos : 460

Рисунок 3.4 – Модуль User

Фрагмент коду даного модуля відображений на рисунку 3.4.

Опис створюваних блоків та загальний вигляд веб-сторінки. Перед початком розробки блогу необхідно було створити макети розроблюваних сторінок (рис. 3.5). На головній сторінці розміщено такі елементи:

- 1) шапка сайту – для відображення меню;
- 2) розміщення контенту – відобразатимуться 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 блоку є адаптивними.

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 користувальницького 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".

Архітектура даної частини MVC представлена на рисунку 3.5.

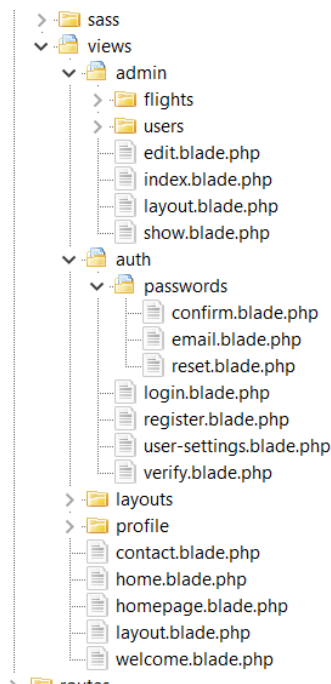


Рисунок 3.5 – Архітектура компонента **Views**

Views часто називають шаблонами. Наприклад, 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 3.6.

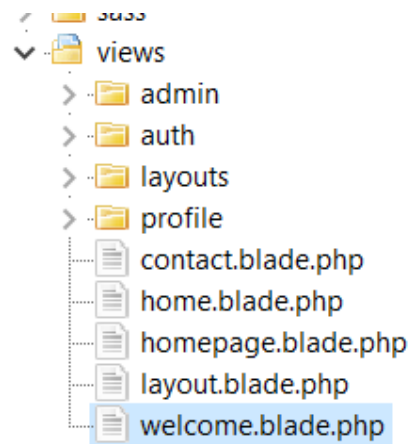


Рисунок 3.6 – Структура папки **views**



Рисунок 3.7 – Фрагмент коду шаблону

Відображається статична та динамічна частина сайту, фрагмент коду, що відповідає за контент на сайті відображено на рисунку 3.7.

3.3. Елементи роботи веб-сторінки в браузері

На рисунку 3.8 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 3.8 – Головна сторінка веб-додатку планування маршрутів

На рисунку 3.9 зображена сторінка «Контакти» на якій форма для заповнення, для зворотнього зв'язку.



Want to get in touch? Fill out the form below to send me a message and I will get back to you as soon as possible!

Name

Email address

Phone Number

Message

Рисунок 3.9 – Сторінка для зворотнього зв'язку «Контакти»

На рисунку 3.10 відображена форма реєстрації користувача сайту.

Рисунок 3.10 – Форма авторизації на сайті користувача/перевізника

На рисунку 3.11 зображена панель адміністрування. На ній можна додавати рейси, а також редагувати або видалити існуючі, додавати водіїв, а також відредагувати або видалити існуючих.

Назва маршруту	Водій	Редагувати	Видалити
Рейс Львів-Київ Мінеральна вода	Андрій Періг	Редагувати	Видалити
Рейс Київ-Львів Автозапчастини	Степан Шульга	Редагувати	Видалити

Рисунок 3.11 – Панель адміністрування «Рейси»

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

Під час редагування водія можна змінити його дані (адресу, ПІБ, номер телефону, номер транспортного засобу та причепу, водійський стаж та аватар тощо) (рис. 3.12).

The screenshot shows an admin panel titled "Адмін-панель" with a "На головну" link. The sidebar contains the following items:

- Вітаю, Admin!
- Рейси
- Водії
- Вийти

The main content area contains a form for editing driver data with the following fields:

- Email:** driver@logist.ua
- ПІБ:** Андрій Періг
- Номер телефону:** 098 48 44 117
- Номер транспортного засобу:** BC 2535 AE
- Номер причепу:** BC 2111 AE
- Водійський стаж:** 25
- Фото водія:** A file selection area with a "Виберіть файл" button and a "Файл не вибрано" status.

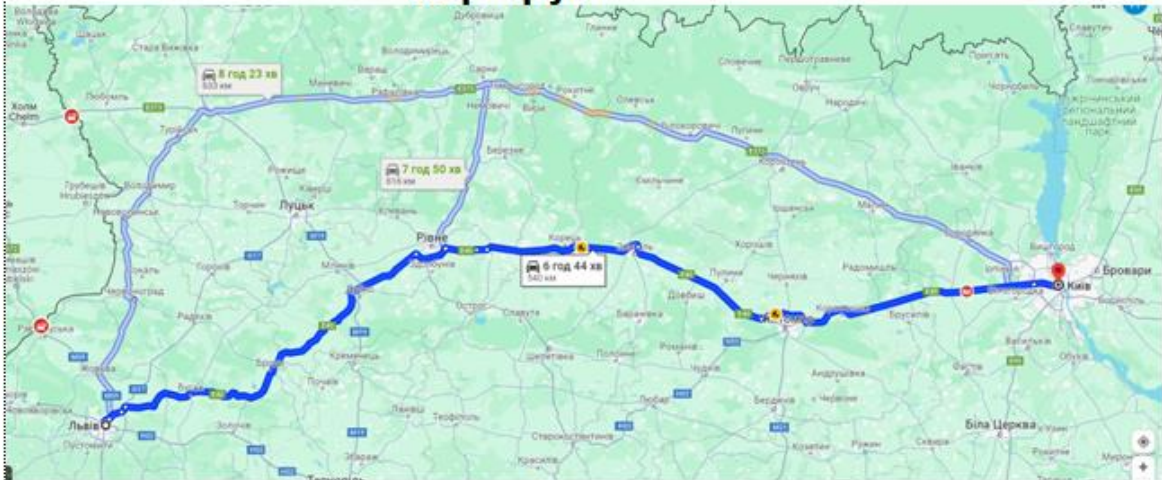
At the bottom of the form is a green "Обновити" button.

Рисунок 3.12 – Сторінка редагування даних водія

Перегляд окремого рейсу наведено на рисунку 3.13. 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 причепу, і завершальний етап – форма зміни статусу рейсу.

Місце завантаження	Львів
Місце вивантаження	Київ
Статус рейсу	Заплановано

Маршрут



Водій

Водій	Андрій Періг
Контактний номер	098 48 44 117
Номер ТЗ	BC 2535 AE

Рисунок 3.13 – Сторінка перегляду даних рейсу

В результаті усі поставлені задачі на початку проекту на розробку веб-додатку планування транспортних перевезень виконано. Реалізовано веб-додаток з користувачами, правами користувача та адміністратора, головну сторінку додатку, панель адміністратора, панель користувача (водія) тощо.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. 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. 4.1) [4].

Таблиця 4.1. 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.


```

graph TD
    NU[НУ] --> ND[НД]
    ND --> NS[НС]
    NS --> T[Т]
  
```

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи [4]:

- характеризують стан або рівень безпеки 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.

4.2. 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 [4]:

$$e_{\min} = \frac{F_b}{F_H} \cdot 100. \quad (4.1)$$

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:

$$\sum F_o = \frac{F_H \cdot e_{\min} \cdot r_o \cdot K}{100 \cdot \tau \cdot \Gamma_1}, \quad (4.2)$$

де F_H – 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:

$$N = \frac{\sum F_o}{F_o}, \quad (4.3)$$

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).

$$F_{\text{л}} = \frac{k \cdot S_{\text{п}} \cdot E}{n_{\text{л}} \cdot \eta \cdot r^2}, \quad (4.4)$$

де k – коефіцієнт запасу, $k = 1,3$; $S_{\text{п}}$ – площа підлоги, м^2 ; $S_{\text{п}} = 36 \text{ м}^2$. E – нормативна освітленість, $E = 300$ 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:

$$h_{\text{н}} = H - (h_1 + h_2), \quad (4.5)$$

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:

$$n_{\text{с}} = \frac{S_{\text{п}}}{l^2}, \quad (4.6)$$

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.

4.3. 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

Веб-сторінка – це інформаційний ресурс доступний в мережі World Wide Web (Всесвітня павутина), який можна переглянути у веб-браузері зазвичай, ця інформація записана в форматі HTML або XHTML, і може містити гіпертекст з навігаційними гіперпосиланнями на інші веб-сторінки.

З метою забезпечення мінімізації часу та інтеграції інтересів замовників і перевізників, корисно використовувати інформаційні логістичні платформи (онлайніві біржі перевезень вантажів ОБП), які дозволять у режимі онлайн, знаходити вільний транспорт для перевезення потрібного вантажу між різними країнами та спрощують пошук замовників за допомогою сайту біржі.

В сучасних умовах життя є можливість спостерігати за транспортуванням вантажу в реальному часі через Інтернет з будь-яких пристроїв за допомогою маячків GPSтрекерів, або софт-трекерів, що у свою чергу є гарантією того, що перевізник буде дотримуватися визначеного 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 користувальницький 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.

Виконуючи завдання кваліфікаційної роботи нами виконано детальний аналіз предметної сфери, проаналізовано конкурентні додатки аналоги, основні їх недоліки та переваги, описано вимоги до проєктованого веб-сервісу. Для реалізації поставленого завдання вибрано інструменти для реалізації веб-додатку, який базується на сучасних технологіях.

У 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;

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. Лехман С.Д. та ін. Запобігання аварійності і травматизму у сільському господарстві / С.Д. Лехман, В.І. Рубльов, Б.І. Рябцев. К.: Урожай, 1993. 272 с.
5. Пошук вантажу та транспорту. Lardi-trans. URL: <https://lardi-trans.com/gruz/>
6. 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.
7. Стеценко І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко ; М-во освіти і науки України, Черкас. держ. технол. ун-т. Черкаси : ЧДТУ, 2020. 399 с.
8. 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, 2010. 270 с.
9. Томашевський В. М. Моделювання систем. К.: Видавнича група ВНУ, 2015, 352 с.
10. Шилдт Г. С# 4.0: повне керівництво. М.: ТзОВ "І.Д. Вільямс", 2011. 1056 с.
11. BursaTransport. Офіційний сайт. URL: <https://www.bursatransport.com/desprebursatransport.html>
12. BURZA NÁKLADŮ 24. Офіційний сайт. URL: <http://www.burzanakladu24.cz/>

13. Cargopedia. Офіційний сайт. URL: <https://www.cargopedia.bg/>
14. Grand Logistics Company. Офіційний сайт. URL: <https://glc.in.ua/uk/>
15. IDEF0 діаграма: приклади і правила побудови [Електронний ресурс] // ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ. 2021. URL: <https://ukr.kagutech.com/3929706-idef0-diagram-examples-and-construction-rules>
16. My dello. Офіційний сайт. URL: <https://mydello.com/>
17. Notepad++. Офіційний сайт. URL: <https://notepad-plus-plus.org/>
18. Novoferm. Офіційний сайт. URL: <https://www.logistyczny.com/>
19. Sytnyk, V. F (2022). Decision support systems [Systemy pidtrymky pryiniattia rishen], KNEU, Kyiv, 614 s. [in Ukrainian]

ДОДАТКИ

Додаток А

Опис атрибутів об'єктів таблиці бази даних

Таблиця А.1 – 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

Таблиця А.2 – 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.3 – 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.4 – 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.

Лістинг файлу 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:400,700,400italic,700italic" rel="stylesheet" type="text/css" />
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300italic,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>
        @if(!isset($user))
        <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4" href="/login">Авторизація</a></li>
        <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4" href="/register">Реєстрація</a></li>
        @endif
        <li class="nav-item"><a class="nav-link px-lg-3 py-3 py-lg-4" href="/contacts">Контакти</a></li>
        @if(isset($user))
```



```

<!-- Core theme JS-->
<script src="js/scripts.js"></script>
</body>
</html>

```

Лістинг файлу 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>
        </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
залишитесь 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>
</div>
@endsection

```

Лістинг файлу 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>
                    <div class="card-body">
                        <div class="alert alert-success"
                            role="alert">
                            @if (session('status'))
                                <div class="alert alert-success"
                                    {{ session('status') }}
                                </div>
                            @endif
                            {{ ('You are logged in!') }}
                        </div>
                    </div>
                </div>
            </div>
        </div>
    @endsection

```

Лістинг файлу contact.blade.php

```

@extends('layout')

@section('content')
    <!-- Page Header-->
    <header class="masthead" style="background-image: url('assets/img/contact-
bg.jpg')">
        <div class="container position-relative px-4 px-lg-
5">
            <div class="row gx-4 gx-lg-5 justify-content-
center">
                <div class="row gx-4 gx-lg-5 justify-content-
                    <div class="col-md-10 col-lg-8 col-xl-7">
                        <div class="page-heading">
                            <h1>Contact Me</h1>
                            <span class="subheading">Have
questions? I have answers.</span>
                        </div>
                    </div>
                </div>
            </div>
        </header>
    <!-- Main Content-->
    <main class="mb-4">
        <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">
                    <p>Want to get in touch? Fill out the form below to send me a message and I will
get back to you as soon as possible!</p>

```



```

<!-- Submit success message-->
<!------>
<!-- This is what your users will
see when the form-->
>
id="submitSuccessMessage"> 3">
<!-- has successfully submitted--
<div class="d-none"
    <div class="text-center mb-
bolder"></div> sign up at
<div class="fw- To activate this form,
<br />
<a
href="https://startbootstrap.com/solution/contact-
forms">https://startbootstrap.com/solution/contact-forms</a>
    </div>
</div>
<!-- Submit error message-->
<!------>
<!-- This is what your users will see when
there is-->
->
<!-- an error submitting the form-
<div class="d-none"
id="submitErrorMessage"><div class="text-center text-danger mb- 3">Error sending
message!</div></div>
    <!-- Submit Button-->
    <button class="btn btn-primary text-
uppercase disabled" id="submitButton" type="submit">Send</button>
    </form>
    </div>
    </div>
    </div>
    </div>
</main>
@endsection

```