МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Економічний факультет
Кафедра економіко-математичного моделювання та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА/ПРОЄКТ
на здобуття освітнього ступеня бакалавра

на тему: «**Розробка системи для організації та управління вантажними перевезеннями: модуль карта»**

Виконав: студент 4 курсу, групи КН-41
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Мороз Олександр Миколайович

Керівник:
Красюк Богдан Віталійович

Рецензент:
Гаврильчик Леонід Сергійович

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри економіко-математичного моделювання та інформаційних технологій _____ (проф., д.е.н. Кривицька О.Р.)

Протокол № ____ від « ___ » _____ 2022 р.

Острог, 2022

Міністерство освіти і науки України
Національний університет «Острозька академія»

Факультет: *економічний*

Кафедра: *економіко-математичного моделювання та інформаційних технологій*

Спеціальність: 122 Комп'ютерні науки
Освітньо-професійна програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри економіко-математичного моделювання
та інформаційних технологій

_____ Ольга КРИВИЦЬКА

«____» _____20__ р.

ЗАВДАННЯ

на кваліфікаційну роботу/проєкт студента

Мороза Олександра Миколайовича

1. Тема роботи: Розробка системи для організації та управління вантажними перевезеннями: модуль карта.

керівник проєкту Красюк Богдан Віталійович.

Затверджено наказом ректора НаУОА від 29 жовтня 2021 року №110.

2. Термін здачі студентом закінченої роботи/проєкту: 03 червня 2022 року

3. Вихідні дані до роботи/проєкту: дана робота полягає у розробці системи для організації та управління вантажними перевезеннями компанії, та контролю складу водіїв.

4. Перелік завдань, які належить виконати: розробка серверної частини, розробка бази даних, розробка інтерфейсу, огляд наявних аналогів.

5. Перелік графічного матеріалу: рисунки, таблиці.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Красюк Б. В.	01.12.2021р.	01.12.2021р.
2	Красюк Б. В.	01.12.2021р.	01.12.2021р.
3	Красюк Б. В.	01.12.2021р.	01.12.2021р.

7. Дата видачі завдання: 01.12.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
1.	Постановка технічного завдання	до 01.12.2021	
2.	Розробка архітектури проекту, погодження функціоналу	до 20.12.2021	
3.	Розробка бази даних	до 08.01.2022	
4.	Розробка макетів інтерфейсу веб застосунку	до 24.02.2022	
5.	Розробка серверної частини проєкту	до 15.05.2022	
6.	Розробка клієнтської частини проєкту		
7.	Попередній захист кваліфікаційної роботи/проєкту	до 01.06.2022р.	
8.	Здача кваліфікаційної роботи/проєкту на кафедру	03.06.2022 р.	

Студент: _____ Олександр МОРОЗ

Керівник кваліфікаційної роботи/проєкту: _____ Богдан КРАСЮК

(підпис)

АНОТАЦІЯ
кваліфікаційної роботи/проєкту
на здобуття освітнього ступеня бакалавра

Тема: Розробка системи для організації та управління вантажними перевезеннями: модуль карта.

Автор: Мороз Олександр Миколайович

Науковий керівник: Красюк Богдан Віталійович, *старший викладач.*

Захищена «……»…… 20__ року.

Пояснювальна записка до кваліфікаційної роботи: *53 с., 33 рис., 5 табл., 12 джерел.*

Ключові слова: база даних, серверна частина, клієнтська частина.

Короткий зміст праці:

Термін «логістика» був вперше застосований у військових операціях і стосувався насамперед процесів забезпечення армійських підрозділів. Останніми роками даний термін набув широкого вжитку в бізнесі і тепер асоціюється саме з корпоративною лексикою. Сучасне уявлення про логістику сильно відрізняється від оригінального. Подальшим розвитком логістики є термін управління ланцюгом поставок, що є свідченням зростаючого розуміння компаніями важливості координації всіх функцій і бізнес процесів. Німецький учений Пфоль вважає, що логістика — це процес планування, реалізації і контролю ефективних та економних з огляду на витрати переміщення та зберігання матеріалів, напівфабрикатів і готової продукції, а також одержання інформації про постачання товарів від місця виробництва до місця споживання згідно з вимогами клієнтури.

І наша команда вирішила зробити цей процес більш швидким та автоматизованим, завдяки застосуванню у цьому процесі додатку на якому адміністратор зможе контролювати та відстежувати етапи доставки в інтуїтивно зрозумілому додатку, який не потребує особливої підготовки кадрів для початку робіт. І також бере на себе навантаження зберігати інформацію про транспортування та статус роботи і переліком вантажу.

The term "logistics" was first used in military operations and referred primarily to the processes of providing military units. In recent years, this term has become widely used in business and is now

associated with corporate vocabulary. The modern idea of logistics is very different from the original. A further development of logistics is the term supply chain management, which is evidence of the growing understanding of companies of the importance of coordinating all functions and business processes. German scientist Pfol believes that logistics is a process of planning, implementing and controlling efficient and economical given the cost of moving and storing materials, semi-finished and finished products, as well as obtaining information on the supply of goods from production to consumption according to customer requirements.

And our team decided to make this process faster and more automated, thanks to the use of an application in which the administrator can control and track the stages of delivery in an intuitive application that does not require special training to get started. And also takes on the load of storing information about transportation and work status and cargo list.

Зміст

Вступ

Ми живемо в той час, коли безперервний розвиток інтернету та інформаційних технологій в цілому зробили так, що навіть маленька компанія,не кажучи вже про великі корпорації, державні установи, має свій веб-застосунок. Фреймворк - це каркас веб-застосунку. Він зобов'язує програміста будувати архітектуру програми відповідно до певної логіки. Зазвичай середовище надає можливість для подій, сховищ і з'єднань даних. Як і машина, рама забезпечує готовий каркас, кузов і двигун. Можна додавати, знімати або замінювати певні деталі за умови, що автомобіль залишається в робочому стані.

Більшість фреймворків пишуться за допомогою JavaScript. Сьогодні більше 90 відсотків веб-застосунків створенні на базі саме JavaScript фреймворків. Незважаючи на гарний базовий захист у всіх сучасних фреймворках, все одно є лазійки, за допомогою яких хакери отримують доступ до важливих даних, та інформації про користувачів, банківські рахунки, тощо. На сьогоднішній день є досить багато аналогів транспортних систем , але вони є досить застарілими і не зручними. Будь-який користувач надає перевагу саме зручності за стосунку, тобто інтуїтивно зрозумілий інтерфейс стоїть на першому місці. Потрібно було вдало підібрати палітру кольорів та стилі застосунку, досить часто користувачі звертають увагу на зовнішній вигляд, якщо він не буде відповідати вимогам, додаток не буде користуватися такою популярністю.

Вантажні перевезення є невід’ємною частиною сучасного життя, з кожним днем кількість перевезень збільшується, відповідно потрібно усе це контролювати. Власники компаній задаються питанням яким чином контролювати усі замовлення та водіїв які здійснюють ці перевезення. Отже , ми створили за стосунок для того щоб допомогти власникам вантажних компаній досить легко контролювати усі замовлення які надходять у компанію.

Наш веб-застосунок складається з декількох частин:

Перш за все потрібно зареєструватися та увійти у наш додаток за допомогою логіну та паролю. Відразу ми потрапляємо на головну сторінку додатку де ми можемо побачити уже наявні замовлення , а також створити нові замовлення заповнивши усі відповідні поля , а також вказати вид вантажу та його вартість. Після створення замовлення, стан цього замовлення змінюється на “у дорозі”. Для того щоб додати водія, а також авто на якому він пересувається і категорію водійських прав, ми повинні натиснути на “Додати водія” у горі веб-застосунку. Ми потрапляємо на сторінку де ми повинні вказати Прізвище , Ім’я, По-батькові водія якого додаємо, а також авто на якому він буде здійснювати це замовлення , і категорію наявних водійських прав.

Звичайно виникають випадки коли водій повинен змінити транспортний засіб, для цього ми повинні натиснути на “Зв’язок водія та авто” у горі веб-застосунку , де ми зможемо побачити список усіх наявних водіїв у базі даних ,а також на якому транспортному засобі вони пересуваються ,та список усіх наявних водійських категорій у водія.

РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Опис предметного середовища

Visual Studio Code

Visual Studio Code — засіб для створення, редагування сучасних веб застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

За основу для Visual Studio Code використовуються напрацювання вільного проекту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовує браузерний рушій Chromium і Node.js. Примітно, що про використання напрацювань вільного проекту Atom і на сайті Visual Studio Code, і в прес-релізі, і в офіційному блозі не згадується.

Переваги

- Зручний інтерфейс.
- Велика різноманітність плагінів.
- Наявність командного рядка та вбудованих відгадчиків.
- Підсвітка синтаксису.
- Інтеграція з Git Hub, Visual Studio.
- Можливість командної роботи.

GitHub

Один з найбільших веб сервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub Inc (раніше Logical Awesome).

Сервіс безкоштовний для проектів з відкритим вихідним кодом з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проектів пропонуються різні платні тарифні плани.

Переваги

- Є можливість прямого додавання нових файлів в свій репозиторій через веб інтерфейс сервісу.
- Вкладені списки завдань у файлах.
- Візуалізація геопросторових даних.
- Можливість збереження попередніх версій проекту.
- GitHub добре підходить для проектів з відкритим кодом. Це корисно для створення власного іміджу розробника або компанії. Завдяки великій кількості користувачів проект бачить велика аудиторія.

Тестування та налагодження

Zoho bugtracker

Отже, для тестування нашого додатку було вирішено використати ручне тестування. Для того щоб оформляти усі баги в прийнятному вигляді було вирішено використовувати Zoho Bugtracker

Переваги Zoho bugtracker

- Безкоштовне користування
- Легкий та зрозумілий інтерфейс
- Можливість додавання вспливаючих сповіщень

codeSandbox

В якості програмного забезпечення для тестування було взято ресурс codeSandbox За допомогою нього можна тестувати та налагоджувати додаток у будь якій версії. На будь-якій операційній системі та без жодних затрат. CodeSandbox є досить зручним у використанні та для виконання колективної роботи є можливість імпорту файлів з

Github. При тестуванні додатку є можливість зміни версії залежних пакетів для того, щоб зрозуміти чи причина проблеми у версії чи у не правильному синтаксису коду.

Переваги codeSandrbox:

- Зручний у використанні інтерфейс, який не вимагає додаткових навичок та знань.
- Для початку не потрібно ніяких затрат.
- Можливість легкого встановлення пакетів різних версій.
- Легко інтегрується з GiTHub.
- Має вбудовані навчальні курси.
- Підтримує ручне тестування.
- Дозволяє тестувати окремі модулі без зміни основного проекту.

Стек технологій:

Back-end:

1. Firebase cloud database
2. Firebase Auth

Front-end:

1. Vue (Vue cli в ролі зборки)
2. Vuex
3. Antd
4. Bootstrap

Firebase

Це по своїй суті набір інструментів, які розробники можуть використовувати створюючи і змінюючи додатки в залежності від своїх потреб. Це і сервер, і база даних, і хостинг, і аутентифікація в одній платформі. Так, Firebase, Realtime, Database надає розробникам API, який синхронізує дані додатки між клієнтами і зберігає їх в хмарному сховищі. Додаток підключається до бази даних через WebSocket, який

відповідає за синхронізацію даних протягом усього сеансу. Розробники, які використовують цю платформу отримують доступ до сервісів за допомогою, яких вони можуть розробляти свої продукти, що дозволяє сконцентруватися безпосередньо на випуску якісного продукту. Деякі з найпопулярніших функцій це база даних: Firestore Database, Authentiaction service, Realtime database. Ці сервіси дозволяють проводити аналітику, зберігання інформації, авторизацію користувачів та багато іншого. Оскільки сервіси знаходяться в хмарі - розробники можуть почергово виконувати масштабування своїх продуктів без проблем. Перевагами платформи мобільних та веб-застосунків FireBase є підтримка спільного хостингу, бази даних, сховища мультимедійних типів даних, систему аутентифікації для веб та мобільних додатків та обмін повідомлень Firebase надає в режимі реального часу базу даних та бекенд як службу.

Переваги

- Безкоштовний початковий план.
- Швидкість розробки.
- Працює на платформі Google.
- Не потребує використання сервера.
- Моніторинг помилок.
- Безпека.

Vue.js

JavaScript фреймворк для розробки інтерфейсів, призначених для користувача та односторінкових web-додатків. Зараз він займає третє за популярністю місце після Angular та React. Це потужний та універсальний фреймворк, створений для швидкого прототипу складних динамічних та інтерактивних призначених для користувача інтерфейсів. Vue.js — поки ще молода технологія тому в її базовій збірці є тільки необхідний мінімум для розробки. При цьому фреймворк дуже гнучкий та легко інтегрується з іншими бібліотеками javascript. Можливість

підключати рішення тільки для потрібних функцій робить web-додатки на Vue.js легкими, продуктивними та масштабованими. Vuex — одне з доповнень до фреймворку, що спрощує, автоматизує та систематизує обробку даних.

Vuex

офіційне доповнення до фреймворку Vue.js, що легко інтегрується з ним і крім своєї основної функції управління станом, також дає можливість створювати зліпки стану даних, а також інструменти для тестування та налагодження web-додатку.

Antd

Це повноцінна дизайн-система, візуальна мова. Зі своїми принципами та бібліотекою компонентів. Проект підтримується розробниками з Alibaba Group. Antd. Має досить різноманітний набір компонентів який дозволяє без витрат часу стилізувати свій додаток, та використати певні стильові рішення у своєму проекті.

Bootstrap

Це один із самих популярних інструментів, які використовуються при створенні сайтів і веб-застосунків. Bootstrap включає в себе багато різних компонентів для веб сайтів: веб форми, кнопки, таблиці, панелі навігацій та є аналогом Antd.

Переваги

- Створення адаптивних сайтів (коректне відображення на мобільних пристроях).
- Легкий в освоєнні.
- Пропонує достатню кількість готових стилістичних рішень.
- Кросс-браузерність (однакове відображення на всіх браузерах).

Figma

це хмарний, багатоплатформовий сервіс для дизайнерів інтерфейсів і web-розробників з яким можна працювати безпосередньо в браузері. І це лише одне

з важливих переваг платформи. Що таке Figma з точки зору функціоналу? Це зручний графічний редактор, в якому можливо створювати: прототипи web-сайтів і додатків:

- окремі елементи інтерфейсу: іконки, кнопки, форми і багато іншого.
- векторні зображення та ілюстрації, інше.

Figma є потужним інструментом, який має значні переваги над іншими програмами, що донедавна використовувались у проектуванні та малюванні дизайну сайтів.

Друга особливість Figma – це повноцінна робота не тільки в десктопной версії сервісу, але і в браузері. Обидва варіанти вимагають постійного інтернет-підключення.

Огляд наявних аналогів.

1С: TMS Логістика. Управління перевезеннями - програма для планування та обліку діяльності транспортних компаній. Дозволяє автоматизувати багато процесів транспортної логістики, включаючи документообіг та організацію мультимодальних вантажоперевезень. Галузеве рішення складається з модулів "планування транспортування", "АРМ водія", "управління майном автопарку" та "ГЛОНАСС/GPS моніторинг", які функціонують на базі загальної інформаційно-аналітичної платформи (СУБД). Програмне забезпечення встановлюється на ПК, модуль "АРМ водія" - на мобільні пристрої з ОС Android. Система управління вантажоперевезеннями призначена для транспортно-експедиційних підприємств, які використовують власний транспорт або замовляють послуги сторонніх компаній. Програма буде корисна будь-яким фірмам зі службами доставки, оскільки дозволяє планувати та контролювати всі деталі внутрішньо міських, міжміських та міжнародних перевезень товарів.

4logist –це хмарна система для логістичних компаній, яка дозволяє оптимізувати діяльність експедиторів та перевізників. З її допомогою можна впорядкувати всі процеси всередині підприємства, що відносяться до сфери,

спростити роботу з документами і знизити людський фактор, а також проаналізувати ефективність команди в реальному часі. Сервіс орієнтований на діяльність транспортних та експедиторських компаній. Його функціонал дозволяє збільшувати ефективність роботи співробітників завдяки тому, що автоматизується частина їх рутинних завдань, наприклад створення документів, організація бази контрагентів, планування завантаження машин.

Cargo.24 – хмарний сервіс для планування та контролю вантажних перевезень, що є системою управління базами даних. Програма дозволяє автоматизувати складання до 40 видів аналітичних та управлінських звітів, що формуються на основі інформації з транспортних заявок. Вбудовані інструменти служать для комплексного аналізу діяльності компанії, забезпечуючи доступ до будь-яких документів через Інтернет.

Програмне забезпечення розроблено для транспортно-експедиційних та логістичних компаній. Системою можуть користуватися менеджери, складські працівники, логісти, бухгалтери, кадровики та керівники, які мають різні права доступу. Програма підійде фірмам із кількома підрозділами, оскільки функціонує на базі хмарного хостингу. Для клієнтів передбачені окремі облікові записи, через які можна створювати заявки на відправку вантажів.

Сервіс призначений для збирання, зберігання та обробки інформації про бізнес-процеси транспортних компаній. У базу даних можна заносити дані про транспорт, маршрути, співробітників, замовників. Співробітники складу можуть заповнювати накладні, скануючи штрих-коди вантажів. Зібрані відомості використовуються системою для автоматичного формування документів, проведення фінансового аналізу та створення звітів. Серед можливостей – sms-інформування про статус доставки, друк накладних, розрахунок зарплати персоналу.

ГрузоПлан - це веб-система для перевізників, в якій можна без зайвих зусиль вести документообіг, накопичувати та обробляти заявки, роздавати права доступу до

системи менеджерам, швидко розраховувати їхню заробітну плату, а також вести базу замовників та інших перевізників.

Сервіс працює в російських умовах, зберігає дані на європейських серверах, захищаючи їх від проникнення третіх осіб, зберігає історію перевезень і не вимагає якихось спеціальних комп'ютерів, для його роботи потрібний лише пристрій з виходом в Інтернет.

Розробники спробували реалізувати системний підхід при збиранні та обробці заявок, забезпечити прозорість процесу від їх додавання до прибуття вантажу у потрібне місце.

Постановка задачі

Створення системи для контролю вантажних перевезень. Проект має назву «**Big Transport**». Клієнтська частина була представлена на vue.js. Для авторизації та баз даних був використаний Google сервіс під назвою firebase, а саме сервіси “firebase auth”, “firebase real time database”, ” firebase, firestore, database”.

Адміністратор:

- Реєструється у додатку.
- Створює нові замовлення вантажних перевезень.
- Додає водіїв у додаток.
- Додає водійські категорії при створенні водія.
- Керує станом перевезення.
- Приєднує виконавця до транспорту.

Роль адміністратора

А. сторінка входу та реєстрації.

Для входу у додаток потрібна обов'язкова аутентифікація користувача, для цього перш за все потрібно зареєструватися та використовувати логін пароль при реєстрації для входу у додаток .

Б. Головна сторінка.

На головній сторінці відображаються наявні замовлення перевезень. З можливістю створення та редагування замовлень, а також наповнення вантажу предметами перевезення. Також є можливість перегляду та контролю стану перевезення.

В. Додавання водіїв.

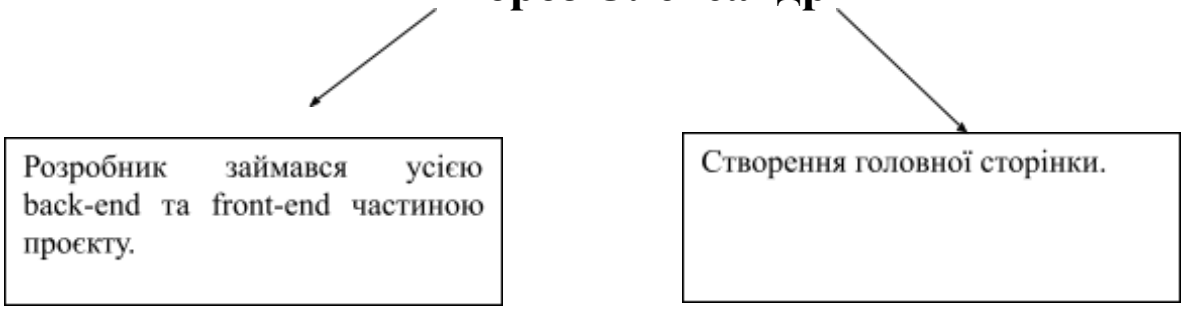
На сторінці додавання водіїв є можливість додавання нових водіїв у систему, а також додавання водійських категорій для кожного водія.

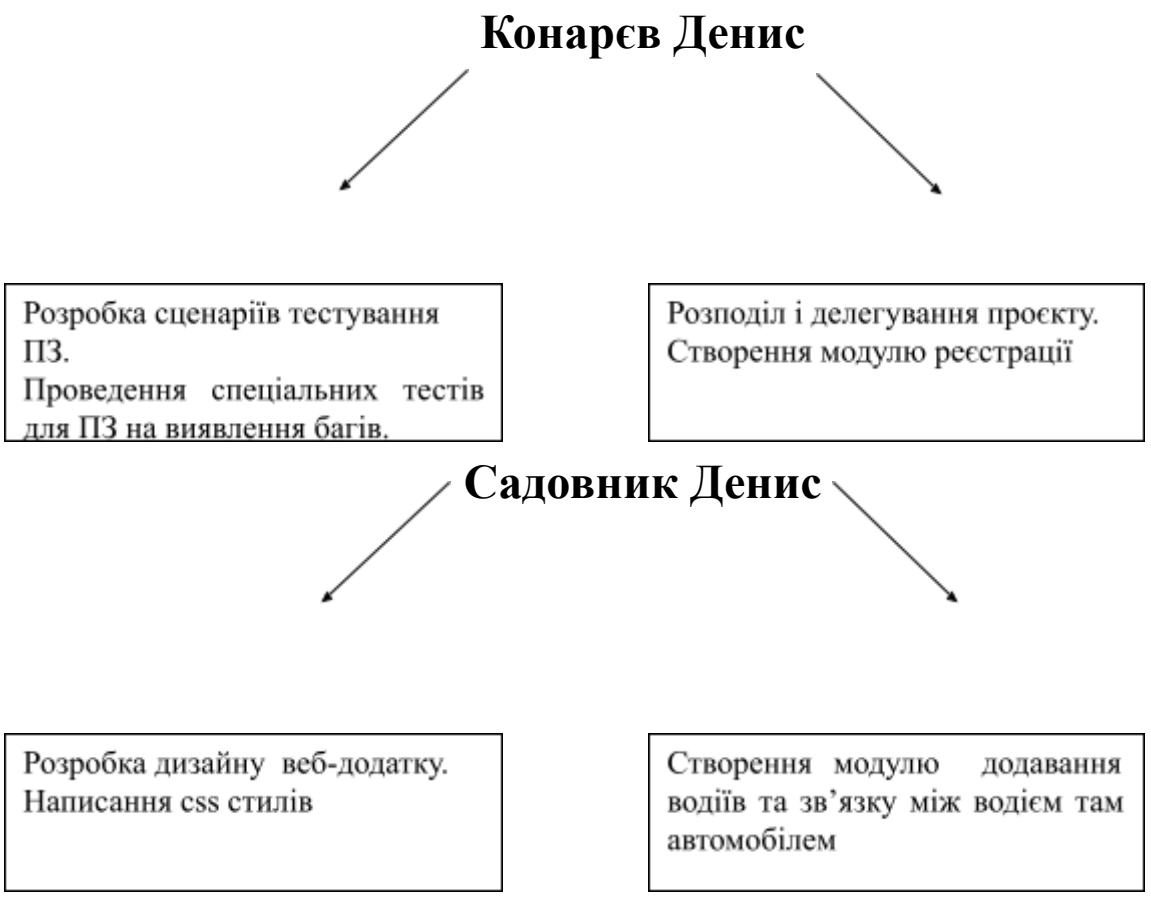
Г. Сторінка приєднання виконавця до транспорту.

На цій сторінці встановлюється зв'язок між водієм та транспортним засобом також була створена наступна UML діаграма.

Відповідальні особи

Мороз Олександр





Організація роботи

Ми як команда розробників отримали замовлення на роботу від нашого керівника (університету).На початку, ми обговорили з керівником технічне завдання та те як ми будемо його виконувати, і для більш ефективного контролю виконання завдання ми вибрали методологію Scrum

Scrum – це методика, яка допомагає командам вести спільну роботу. Як спортивна команда готується до вирішальної гри (до речі, scrum — англ. «битва», елемент гри в регбі), так і команда співробітників компанії повинна отримувати уроки з набутого досвіду, освоювати принципи самоорганізації, працюючи над вирішенням проблеми, та аналізувати свої успіхи та провали, щоб постійно вдосконалюватися. Scrum сприяє цьому.

Методику Scrum найчастіше застосовують команди розробників додатків, але принципи та досвід її використання можна застосувати до командної роботи будь-якого роду. Це одна із причин такої популярності методики. Scrum часто представляють як платформу управління проектами за методикою Agile. Учасники команди Scrum проводять збори, використовують спеціальні інструменти та беруть на себе особливі ролі, щоб організувати роботу та керувати нею.

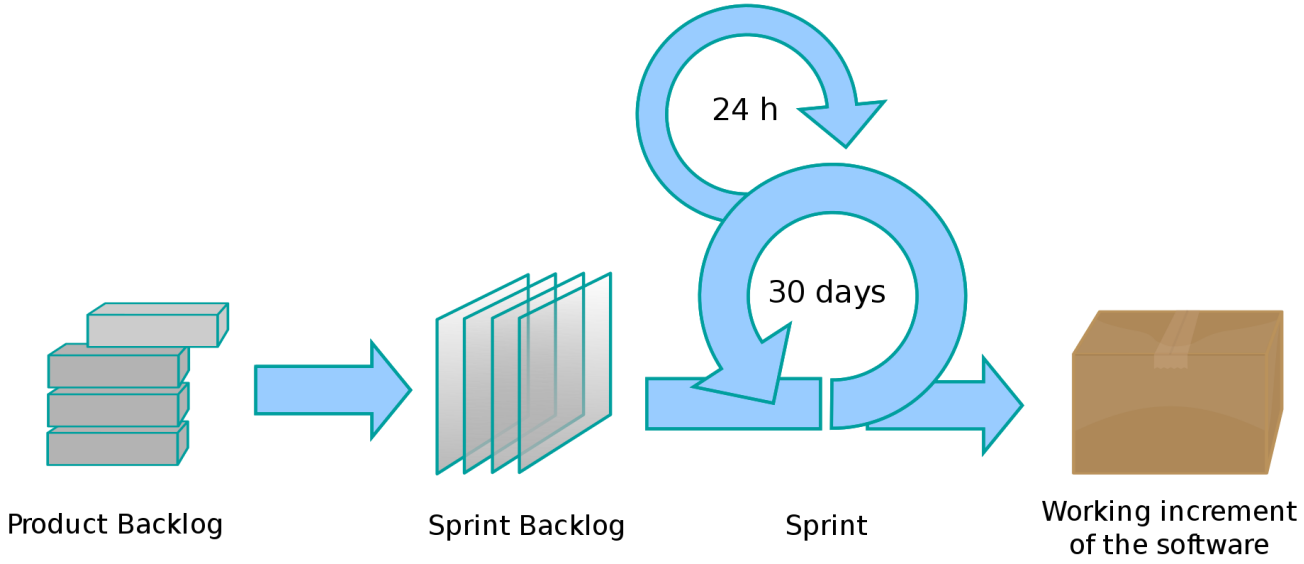


Рисунок 1.1 Структурна схема методології Scrum

Процеси та події в Scrum

Основною частиною процесу Scrum являється спринт. Спринт — це короткий часовий інтервал який є однаковий протягом усього життя продукту , протягом якого команда scrum виконує заданий обсяг роботи. Спринти лежать в основі методологій scrum та agile, і правильний вибір спринтів допоможе вашій agile-команді випускати якісніше програмне забезпечення без зайвого головного болю.

Перед початком кожного Sprint проводиться Sprint Planning який знаменує початок спринту. На цьому етапі проводиться формування Sprint Backlog, що містить

завдання (Story, Bugs, Tasks), які мають бути виконані у поточному спринті. Кожен спринт повинен мати мету, яка є мотивуючим фактором і досягається за допомогою виконання завдань із Sprint Backlog.

Щодня проводиться Daily Scrum, на якому кожен член команди відповідає на запитання «що я зробив учора?», «що я планую зробити сьогодні?», «Які перешкоди на роботі я зустрів?». Завдання Daily Scrum — визначення статусу і прогресу роботи над Sprint, раннє виявлення перешкод, вироблення рішень зі зміни стратегії, необхідних досягнення цілей Sprint.

По закінченню Sprint виробляються Sprint Review та Sprint Retrospective, завдання яких оцінити ефективність (продуктивність) команди в минулому спринті, спрогнозувати очікувану ефективність (продуктивність) у наступному спринті, виявленні наявних проблем, оцінки ймовірності завершення всіх необхідних робіт по продукту та інше .

У класичному Скрамі існує 3 базові ролі:

- Власник продукту
- Скрам-майстер
- Команда розробки

Власник продукту - повинен не тільки розуміти клієнта, але й мати концепцію цінності, яку Scrum-команда постачає клієнту. З іншого боку, власник продукту визначає баланс між потребами інших зацікавлених сторін у створенні. Таким чином, власник продукту повинен врахувати всі ці вхідні дані та розставити пріоритети у роботі. Мабуть, це його найважливіший обов'язок, оскільки суперечливі пріоритети та розпливчасті установки не лише знижують ефективність команди, але й можуть порушити важливі довірчі відносини між бізнесом та командою розробників.

Рисунок 1.2 Власник продукту в методології Scrum

Scrum-майстер - це роль, яка поєднує всі елементи та забезпечує ефективну реалізацію методики Scrum. На практиці це означає, що такий співробітник допомагає власнику продукту визначати цінність, команді розробників – постачати цінність, а Scrum-команді – ставати краще. Scrum-майстер - це лідер-слуга, який не тільки формує лояльний стиль керівництва, але щодня його практикує.

Він допомагає власнику продукту краще розуміти та виражати цінність, керувати беклогом, планувати роботу з командою та розбивати цю роботу для найбільш ефективного навчання. Команді розробників Scrum-майстер допомагає з самоорганізацією, налаштовує учасників на досягнення кінцевого результату. Крім того, Scrum-майстер служить організації в цілому, допомагаючи їй зрозуміти суть методики Scrum та створити для неї сприятливе середовище.

Команда розробників – це люди, які виконують роботу. Спершу можна подумати, що під командою розробників маються на увазі лише технічні спеціалісти. Але це не завжди так. Згідно з керівництвом по Scrum, команда розробників може складатися з людей різних професій, у тому числі дизайнерів, письменників, програмістів і т.д. Уявіть, що у вас є будинок і ви наймаєте розробника. Він розробляє проект та виконує роботу. Так, він може класти цеглу, займатися водопроводом і навіть копати величезні ями, але ця людина все одно просто «розробник». Це означає, що роль

«розробника» в Scrum означає учасника команди, який має певні навички, необхідні для виконання роботи в команді.

Рисунок 1.3 Схема Команди розробників методології Scrum

Висновки до розділу 1

Отже, ми ознайомилися з предметним середовищем, зважили усі переваги та недоліки програмного забезпечення яке наявне у вільному користуванні. Визначилися з яким будемо працювати та встановили завдання яке будемо виконувати. Ми вирішили, що дана програма буде досить актуальна у даний момент часу так, як вантажні перевезення зараз потрібні усім без винятку компаніям. Логістика компаній суттєво ускладнилась у періоди війни та карантинних обмежень і ми хочемо допомогти її полегшити. Ми створили український полегшений аналог системи транспортних перевезень, який у подальшому буде вдосконалюватися. Так як онлайн торгівля зростає і доставка великогабаритних товарів збільшується логістичні навантаження також збільшуються і тому актуальність нашого додатка росте з кожним днем. На ринку праці є нестача кваліфікованих логістів, які використовують 1С та інші аналоги, ми представляємо додаток, який не потребує широких знань у IT сфері.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

Вибір бази даних

Structured query language - Мова структурованих запитів, зазвичай скорочена як SQL, є мовою програмування для конкретного домену, яка використовується для зберігання, обробки та отримання даних у СУБД (Реляційна система управління базами даних).Він в основному використовується для управління структурованими даними, коли ми маємо взаємозв'язок між різними сутностями та змінними даних.

NoSQL (також стосується не лише SQL, не-SQL або нереляційних) - це база даних, яка дає вам спосіб керувати даними, що знаходяться в нереляційній формі, тобто яка не структурована в табличному вигляді. NoSQL дедалі більше набирає популярності, оскільки використовується в додатках великих даних та в реальному часі. Їх структури даних повністю відрізняються від структур реляційних баз даних.

SQL проти NoSQL Безпека

Перш за все, важливо знати значення безпеки баз даних. Щоб база даних зберігала інформацію в безпечному режимі, потрібно забезпечити конфіденційність, цілісність та доступність, що в сукупності називається ЦРУ.

Конфіденційність означає, що доступ до даних можуть мати лише уповноважені користувачі або системи. Цілісність - це точність та узгодженість даних протягом їхнього терміну служби, а доступність означає, що дані повинні бути доступними, коли це потрібно.

Більшість реляційних баз даних на базі підприємств або SQL, таких як Oracle і MSSQL мають вбудовані сильні функції безпеки. Вони дотримуються властивостей ACID, які забезпечують безпечні та надійні транзакції баз даних.СУБД також має такі функції, як захист на основі ролей, контроль доступу за допомогою дозволів на рівні користувача, зашифровані повідомлення, підтримка контролю доступу до рядків і стовпців тощо. Однак ці функції безпеки потребують значної плати за ліцензування та впливають на швидкість доступу до даних .

Різниця між SQL та NoSQL

Sql

- Бази даних SQL є вертикально масштабованими. Їх можна масштабувати, збільшуючи апаратну ємність (ЦП, ОЗУ, SSD тощо) на одному сервері.
- Бази даних SQL - це в основному реляційні бази даних (СУБД).
- Старі технології.
- Бази даних SQL складаються з таблиць у формі рядків і стовпців і повинні суворо дотримуватися стандартних визначень схем.
- Вони є кращим варіантом для програм, які потребують багаторядкових транзакцій.
- Вони мають добре розроблену заздалегідь визначену схему для структурованих даних.
- Бази даних SQL належним чином відповідають властивостям ACID (атомність, послідовність, ізоляція та довговічність).
- Додавання нових даних у базу даних SQL вимагає внесення деяких змін, таких як засипка даних, зміна схем.
- Відмінна підтримка постачальників та підтримка спільноти доступна для всіх баз даних SQL.
- Найкраще підходить для додатків із високим рівнем транзакцій.
- Не підходить для ієрархічного зберігання даних.
- Приклад баз даних SQL: MySQL, Oracle, MS-SQL, SQLite.

NoSQL

- Бази даних NoSQL можна масштабувати горизонтально. Їх можна масштабувати, додаючи до інфраструктури більше серверів для управління великим навантаженням і зменшення купи.
- Бази даних NoSQL - це в основному нереляційні або розподілені бази даних.
- Порівняно молода технологія.

- Бази даних NoSQL можуть базуватися на документах, парах ключ-значення, графіках або стовпцях, і вони не повинні дотримуватися стандартних визначень схем.
- Вони мають динамічну схему для неструктурованих даних. Дані можна гнучко зберігати, не маючи заздалегідь визначеної структури.
- Бази даних NoSQL підтримують денормалізовану схему.
- Дешевший масштаб у порівнянні з реляційними базами даних.
- Не підходить для складних запитів, оскільки в NoSQL немає стандартного інтерфейсу для обробки запитів.
- Запити в NoSQL не такі потужні, як запити SQL.
- Він називається UnQL, і синтаксис використання мови неструктурованих запитів буде відрізнятися від синтаксису до синтаксису.
- Бази даних NoSQL найкраще підходять для ієрархічного зберігання даних, оскільки вони слідують методу пари ключ-значення для зберігання даних.
- Вони класифікуються на основі способу зберігання даних як сховище ключ-значення, сховище документів, сховище графіків, сховище стовпців та сховище XML.
- Бази даних NoSQL належним чином відповідають теоремі Brewers CAP (узгодженість, доступність та допуск на розділи).
- Нові дані можна легко вставити в бази даних NoSQL, оскільки вони не вимагають жодних попередніх кроків
- Для баз даних NoSQL доступна лише обмежена підтримка спільноти.
- Ви можете використовувати NoSQL для важких транзакційних цілей. Однак це не найкраще підходить для цього.
- Підходить для ієрархічного зберігання даних та зберігання великих наборів даних (наприклад, великі дані).
- Приклади баз даних NoSQL: MongoDB, Apache CouchDB, Redis, HBase,Firebase.

Проектування системи (проектування бази даних

Оскільки сайт написано з допомогою платформи FireBase, база даних якій подається разом с сервісом є найкращим вибором для інформаційної системи. Firebase залишаються чутливими навіть у режимі офлайн, оскільки Firebase Realtime Database SDK зберігає дані.

Cloud Firestore – це гнучка, масштабована хмарна база даних від Firebase і Google Cloud Platform для інтернету, мобільних платформ і серверних додатків. Як і Firebase Realtime Database вона синхронізує дані між клієнтськими додатками за допомогою слухачів реального часу, а також пропонує підтримку офлайн режиму для мобільних платформ і веб.

База даних реального часу: Замість типових HTTP-запитів в базі даних Firebase Realtime використовується синхронізація даних кожний раз, коли дані змінюються, будь-який приєднаний пристрій отримує це оновлення в мілісекунди.

Створення бази даних

Проміжні сутності: id, title, categoriya, description, published, billerCity, Biller Country, BillerStreetAddress, BillerZipCode, clientCity, ClientCounry, ClientEmail, ClientName, ClientStreedAdress, ClientZipCode, InvoiceDate, InVoiceDate, Unix, InvoiceDraft, InvoiceId, InvoicePaid, Invoice Pending, InvoiceTotal

Сутність **users**

Має такі поля:

id – користувача.

email – поштова скринька користувача.

Password – пароль користувача.

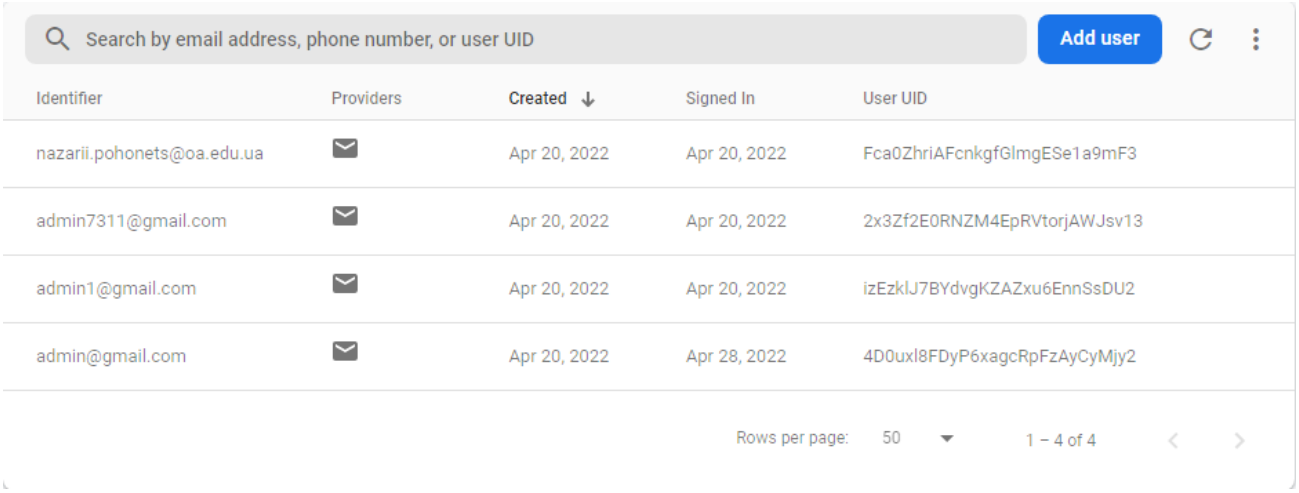


Рис. 2.1. Структура таблиці users.

Сутність **orders**

Має такі поля:

Car – Назва авто.

Dvig – який двигун.

Kb – Кб.м Двигуна.

Rozxod – Розхід двигуна на 100 км.

clientCity – Місто замовника.

clientCountry – Країна замовника.

clientEmail – Електронна скринька замовника.

clientName – ПІБ замовника.

ClientMobile – мобільний номер замовника.

invoiceDate" – дата відправки.

invoiceDateUnix – дата відправки в Unix системі.

id" – айді замовлення.

invoicePaid – поле ідентифікації стану вантажу.

invoicePending – поле ідентифікації стану вантажу.

paymentDueDate – орієнтована дата прибуття.

paymentDueDateUnix – орієнтована дата прибуття В Unix системі.

productDescription – коментарі до перевезення.

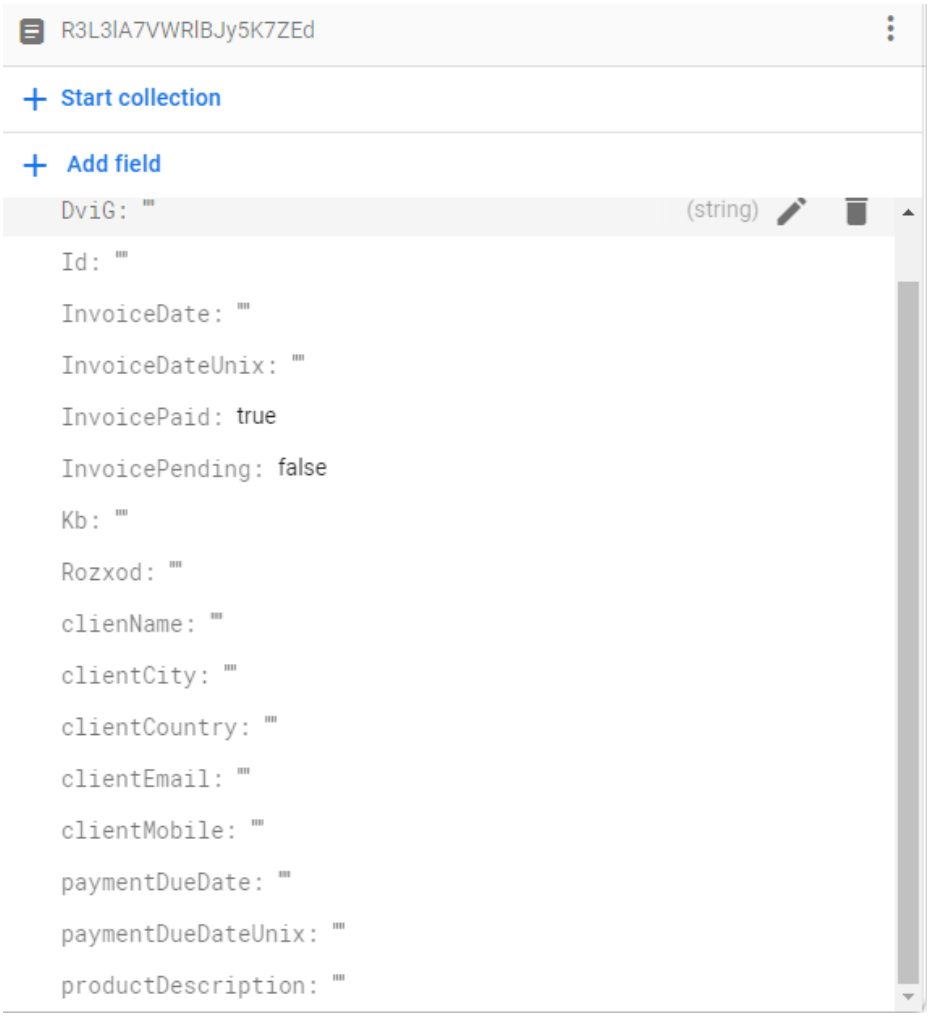


Рис. 2.2. Структура таблиці orders.

Сутність **invoiceItemList**

Має такі поля:

id" – айді товару.

itemName – назва товару.

price – ціна товару.

qty – кількість штук товару.

total – розрахована загальна ціна.

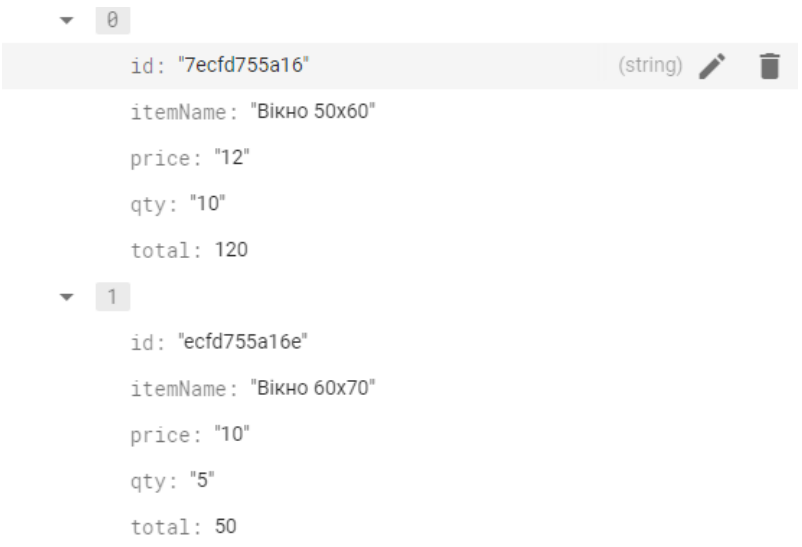


Рис. 2.3. Структура таблиці invoiceItemList.

Сутність **drivers**

Має такі поля:

Title – Прізвище Ім’я По-батькові водія.

Description – Назва автомобіля закріплена за водієм.

Categoriya – Водійська категорія водія.

Published – змінна для контролю стану опублікованості.

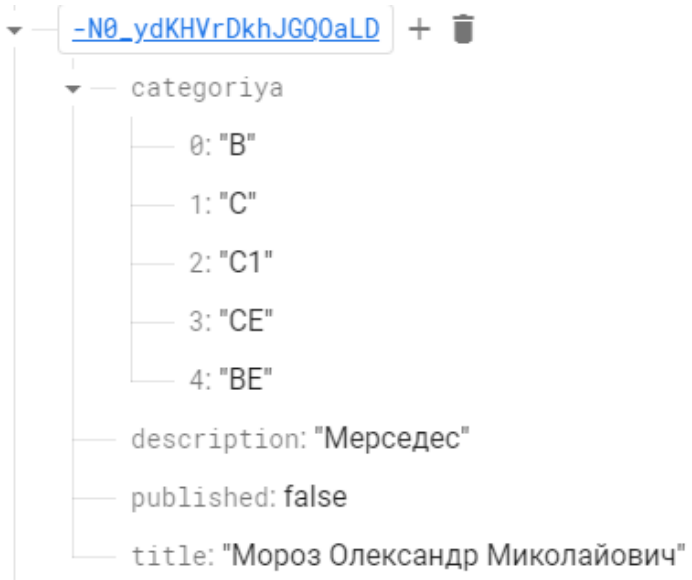


Рис. 2.4. Структура таблиці drivers.

Висновок до розділу 2

Отже, в цьому розділі ми проаналізували предметне середовище, та по результатам нашого дослідження визначилися яким способом будемо контролювати якісь продукту, та виконувати вимоги які перед нами ставлять користувачі.

Також створили базу даних на сервісі firebase яка є досить простою у підключенні та зручною у користуванні. Для підключення firebase потрібно встановити модуль firebase, а також використовувати різні плагіни для підключення окремих її частин (auth, realtime, database, firestore, database). Та виділили для себе багато переваг у користуванні цим сервісом.

РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

Засоби розробки

JavaScript – мова програмування, що дозволяє реалізувати ряд складних рішень в web-документах. Вона допомагає зробити сторінки сайту більш інтерактивними, обробляє дії користувачів сайту. Це об'єктно-орієнтована клієнтська мова, яка підтримується додатками, що працюють з дизайном сайту. JavaScript став ще більш популярним в середовищі девелоперів, коли з'явилася AJAX-технологія, що призвело до нового етапу в розробці сайтів.

Ось лише кілька окремих прикладів застосування технології:

- відображення контенту, який періодично оновлюється (інтерактивні карти);
- створення якісної анімації і графічних об'єктів у форматі 2D/3D;
- опція прокрутки відеозапису в медіа програвачі.

Поряд з HTML і CSS, javascript – третій важливий блок, на основі якого будується більшість стандартних веб-інтерфейсів. Ядро JavaScript включає цілий ряд функцій, які дають наступні можливості:

- Зберігати дані в змінних;
- Активувати частину коду у відповідності з певними сценаріями, які здійснюються на сторінці сайту;
- Створювати контент, який оновлюється автоматично;
- Керувати мультимедійними можливостями (працювати з відео, анімувати зображення).

Саме тому мова настільки популярна серед розробників. Але ще більше можливостей дає функціонал, який доступний як надбудова щодо основних складових Javascript. Мова йде про інтерфейси прикладного програмування (API), які істотно розширюють набір інструментів розробника.

API – це готові модулі коду, які допомагають програмісту в реалізації деяких складних завдань. Зазвичай такі «заготовки» діляться на браузерні і API третіх розробників.

До браузерних API-інтерфейсів відносяться:

- API - інтерфейс DOM (Document Object Model).
- Модулі геолокації.
- API Canvas і WebGL.
- Аудіо та відео API.

Vue – JavaScript фреймворк з відкритим вихідним кодом для створення користувацьких інтерфейсів. Легко інтегрується в проекти з використанням інших JavaScript бібліотек. Може функціонувати як веб-фреймворк для розробки односторінкових додатків в реактивному стилі.

Переваги використання Vue.js як фреймворку для сайту:

- велике співтовариство;
- посилений HTML. Це означає, що Vue.js має багато характеристик схожих з Angular, це завдяки використанню різних компонентів також допомагає в оптимізації HTML блоків;
- детальна документація. Vue.js має дуже детальну документацію, яка може прискорити процес навчання для розробників і заощадити багато часу на розробку програми, використовуючи тільки базові знання HTML і JavaScript;
- усуває деякі невідповідності між браузерами;
- корисно для вибору і ітерації над наборами вузлів DOM;
- підтримка різних браузерів;
- малий розмір плагінів;
- масштабування. Vue.js може допомогти в розробці досить великих шаблонів багаторазового використання, які можуть бути зроблені майже за той же час, що і більш прості;

Недоліки використання Vue.js як бібліотеки для сайту:

- регулярні оновлення, які змінюють існуючу поведінку;
- недолік ресурсів. Vue.js як і раніше займає досить невелику частку ринку в порівнянні з React або Angular, що означає, що обмін знаннями в цьому середовищі все ще перебуває на початковій стадії;
- якість плагінів дуже мінлива;
- ризик надмірної гнучкості. Іноді у Vue.js можуть виникнути проблеми при інтеграції в величезні проекти, і поки ще немає досвіду можливих рішень, але вони обов'язково з'являться найближчим часом;
- продуктивність зазвичай набагато повільніше, ніж еквівалентний (добре продуманий) код без jQuery.

Vue повністю підходить і для створення складних односторінкових програм (SPA, Single-Page, Applications), якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

Опис програмної реалізації

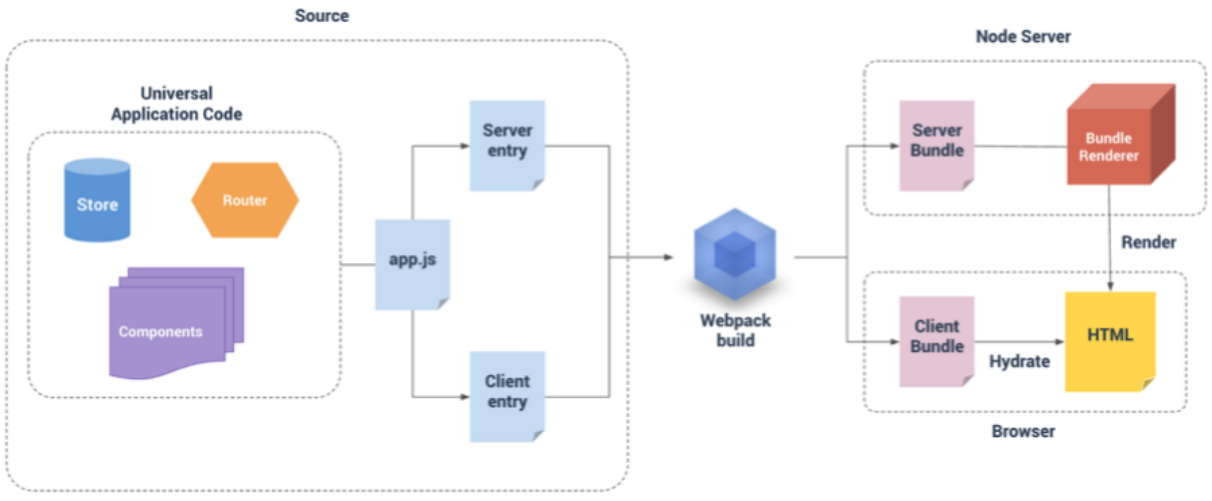


Рис 3.1 Структура додатку.

Фронтенд

Single-page application, SPA

Односторінковий застосунок також відомий як **односторінковий інтерфейс** (single-page interface, SPI) – в перекладі означає як додаток на одній сторінці. Це веб додаток який для роботи загружає усі javascript - файли, а також файли стилів додатка CSS. Існують додатки які можуть завантажувати сотні таких скриптів, але це не означає що сторінка буде завантажуватись повільніше, тому що ці скрипти завантажуються тільки по мірі необхідності, тобто тільки тоді коли цей компонент буде використаний. У такому типі за стосунку ми нікуди не переходимо, а все виконується на одній сторінці у динамічному форматі.

MVVM - (Model-View-ViewModel). шаблон проектування, що застосовується під час проектування архітектури застосунків (додатків). Публічно вперше був представлений Джоном Госсманом (John Gossman) у 2005 році як модифікація шаблону Presentation Mode.

MVVM використовується для відокремлення моделі та її відображення. Необхідністю цього є надання можливості змінювати їх незалежно одну від одної. Наприклад, розробник працює над логікою роботи з даними, а дизайнер — з користувацьким інтерфейсом. MVVM була створена з метою поділу праці дизайнера і програміста.

Шаблон MVVM ділиться на три частини:

- Модель (Model), як і в класичному шаблоні MVC, модель являє собою фундаментальні дані, що необхідні для роботи застосунку.
- Вигляд (View) як і в класичному шаблоні MVC, вигляд — це графічний інтерфейс, тобто вікно, кнопки тощо.
- Модель вигляду (ViewModel, що означає «Model of View») з одного боку є абстракцією вигляду, а з іншого надає обгортку даних з моделі, які мають зв'язуватись.

Фактично ViewModel призначена для того, щоб:

- Здійснювати зв'язок між моделлю та вікном.
- Відслідковувати зміни в даних, що зроблені користувачем.
- Відпрацьовувати логіку роботи View (механізм команд).

MVVM являє собою наступну та покращену версію MVC. Говорячи про модель MVVM, ми повинні зачепити також і структуру MVC.

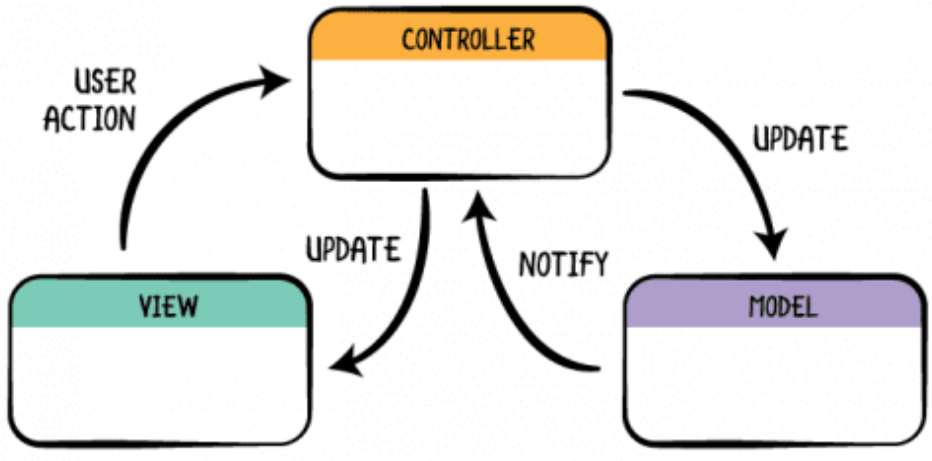


Рис 3.2 Структура MVC.

MVC складається з трьох компонентів:

- Модель.
- Вид.
- Контролер.

Модель: простими словами, модель містить дані про програму. Тут вказується вся інформація, яка має бути важливою для відображення, її вимоги щодо доступу та інших перевірок.

Перегляд: Перегляд відображає дані в компоненті модель. Будь-яка відповідь від користувача також розпізнається та надсилається до компонента Controller. Контролер відповідає за надання даних, присутніх у моделі, компоненту «Вид» та інтерпретацію відповідей користувачів, які розпізнаються компонентом «Перегляд».

Для створення свого інтерфейсу розробники vue js надихнулись патерном MVVM який є братом MVC та створили свій патерн на основі MVVM. Ця конструкція має в

собі обробку даних на сервері і відображення результатів на стороні клієнта в додатку Vue.

На допомогу цьому був створений патер управління станом Vuex. Цей патерн служить для централізованого зберігання та обробки даних для всіх компонентів, також гарантує, що стан змінних може змінитись тільки передбачуваним варіантом. За допомогою такого способу ми можемо створювати мутація для наших компонентів що створить реактивність для нашого додатку.

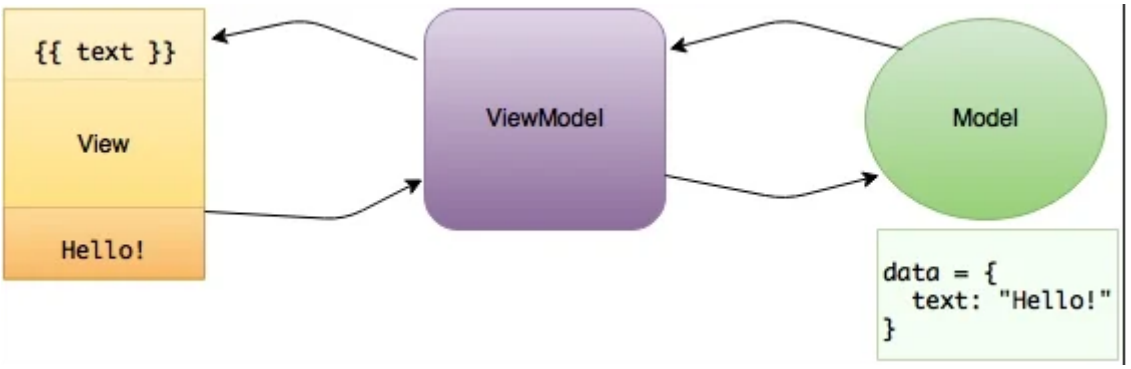


Рис 3.2 приклад комунікації Vue.

На малюнку 3.2 ми можемо побачити що MVVM View спочатку прив’язує події до Model через своєрідний міст ViewModel, а Model керує View через прив’язки даних.

Бекенд

Так як явного використання Back-end у нас не має усі функції цього виконують наші Api з Google сервісу Firebase. Вони є досить гнучкими та зручними у підключенні та використанні. Компанія також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, та Backbone.js. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері.

Firebase – це одне з BaaS-рішень (Backend as a Service), яке дає розробнику масу можливостей.

Firebase Authentication – це Api сервісу firebase який з легкістю дозволяю додати авторизацію та реєстрацію у ваш додаток. Головна особливість цього сервісу полягає у тому що розробнику не потрібно відволікатися на Back-end частину. Додаток підключається до бази даних через WebSocket, який відповідає за синхронізацію даних протягом усього сеансу. Сервіс дозволяє виконувати Авторизація за допомогою багатьох способів

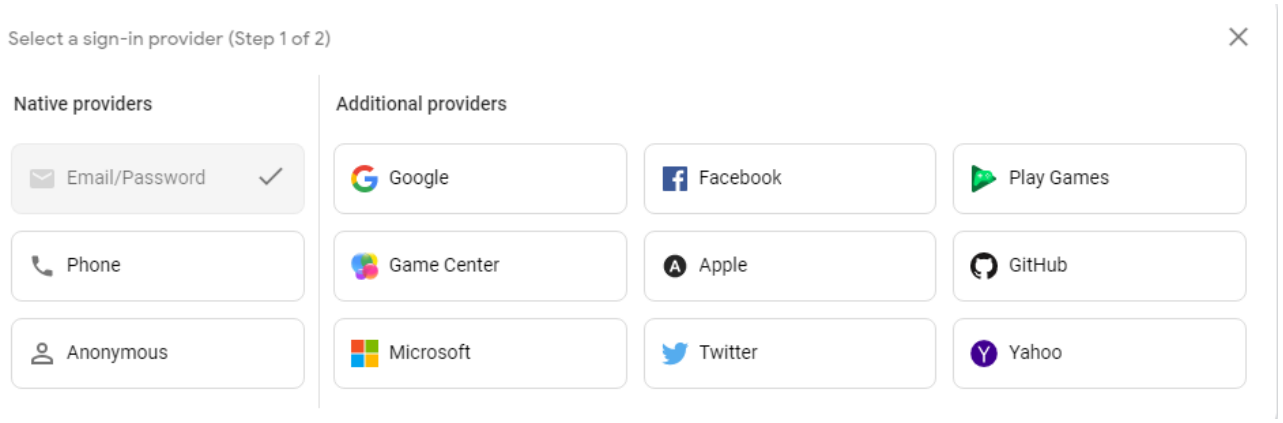


Рис 3.3 Способи авторизації firebase.

Як можемо бачити у нашому проекті ми використали спосіб авторизації за допомогою Email/Password. Також тут можна вибирати чи потрібно підтвердити електронну скриньку чи ні у нашому проекті ніякого підтвердження не потрібно. Наявні лише дві умови для проходження реєстрації це:

- Пароль більше 7 символів.
- Правильна електронна скринька (наявність @).

Зручність усього цього полягає у тому що для виконання реєстрації потрібно лише одна функція сервіс firebase у своїй back-end частині сам створює потрібну базу даних для зберігання логіну та захищеного паролю.

Firebase realtime Database – це сервіс для зберігання і синхронізації даних в реальному часі. Це дозволяє змінювати данні у реальному часі для всіх користувачів, які будуть ним користуватись. Це вид бази даних в яких ми працюємо не з таблицями як у Mysql sql, а з документами і тому швидкість набагато вища чим в реляційних базах даних.

Отже, наша база даних орієнтована на документи тому нам її не потрібно підготовлювати створювати поля типи полів і так далі. І з легкістю можемо спостерігати за статистикою підключення до нашої бази даних.

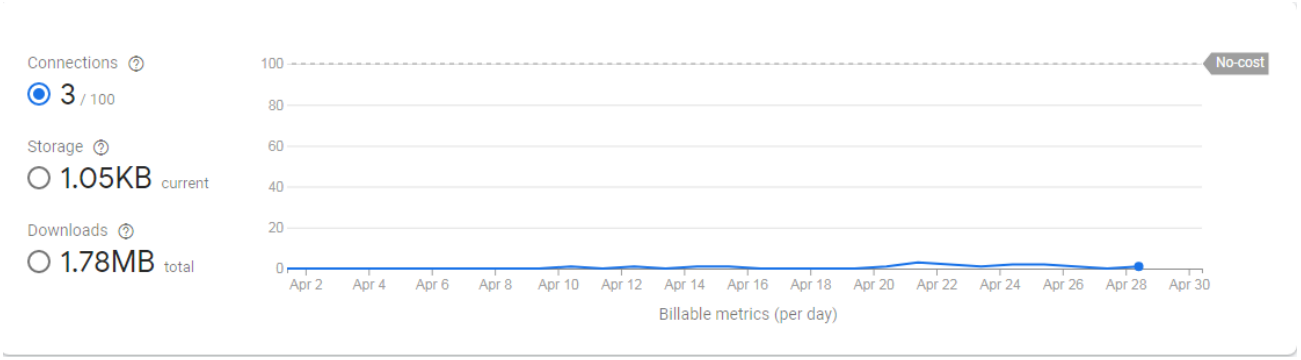


Рис 3.4 статистика Realtime Database.

Firebase Firestore Database – це гнучка база даних яка з легкістю масштабується і як і Realtime Database, вона синхронізує ваші дані між клієнтськими додатками за допомогою слухачів реального часу і також підтримує офлайн режим для мобільних платформ і веб застосунків. Модель даних cloud firestore підтримує гнучкі ієрархічні структури даної. Зберігає дані в документах, які в свою чергу зберігаються в колекціях. В документи можуть бути вкладені об’єкти та під колекції.

Також перевагою цього є те що Cloud firestore підтримує офлайн режим, база даних кешує дані які ваш додаток часто використовує, і по цьому додатку може писати, читати, і робити виклики коли ваш пристрій знаходиться офлайн. Коли знову підключитесь до Firestore всі локальні зміни приміняться у хмарі.

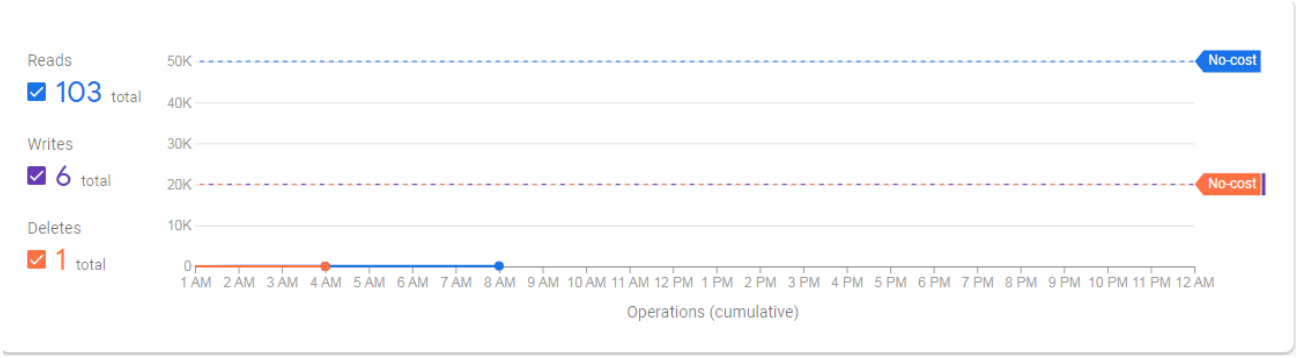


Рис 3.5 статистика Firestore Database.

Керівництво користувача

Для того щоб реалізувати вибрану нами тему, ми вирішили створити роль адміністратора.

Для входження у додаток потрібно спочатку пройти реєстрацію.

Для цього у потрібні поля потрібно ввести свої облікові данні (Електронна скринька та пароль).

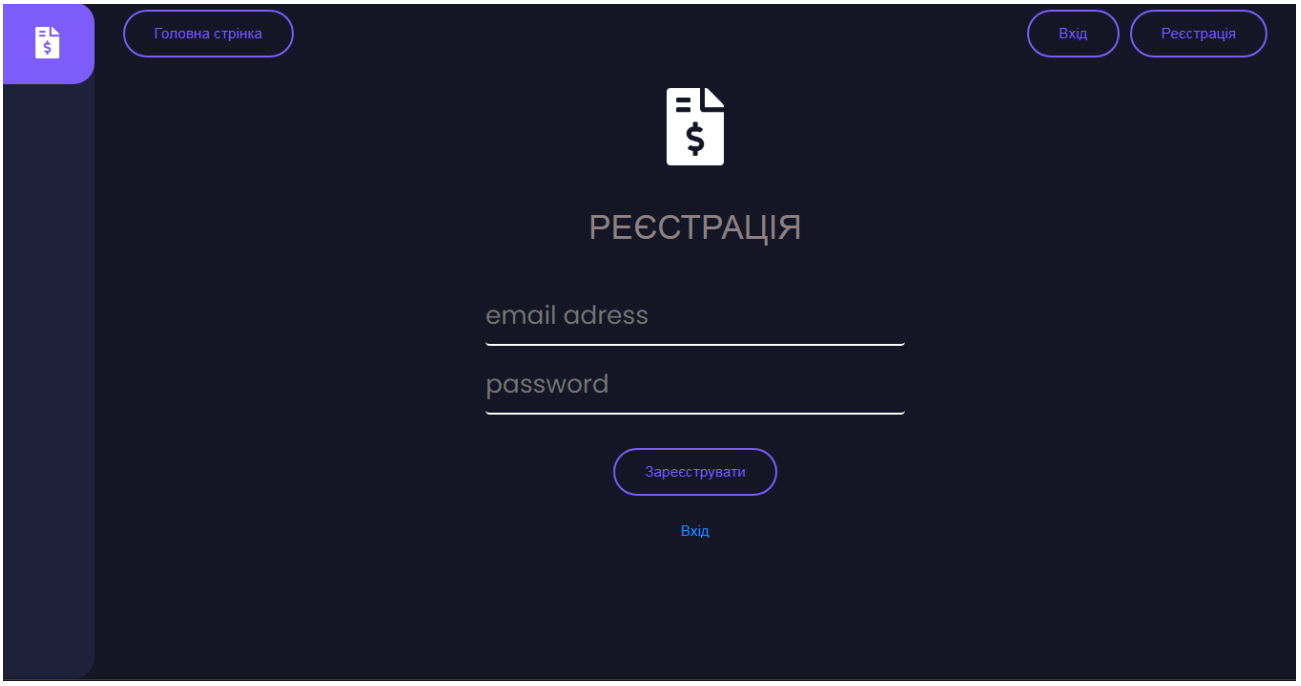


Рис 4.1 сторінка реєстрації.

Далі переходячи по посиланню «Головна сторінка» з’являється модальне вікно з проханням пройти авторизацію. Натиснувши “Ок” вас направляє на сторінку входу. На цій сторінці користувач вводить свої особисті данні (поштова скринька та пароль), які він створював при реєстрації. У разі введення не вірних даних з’явиться помилка.

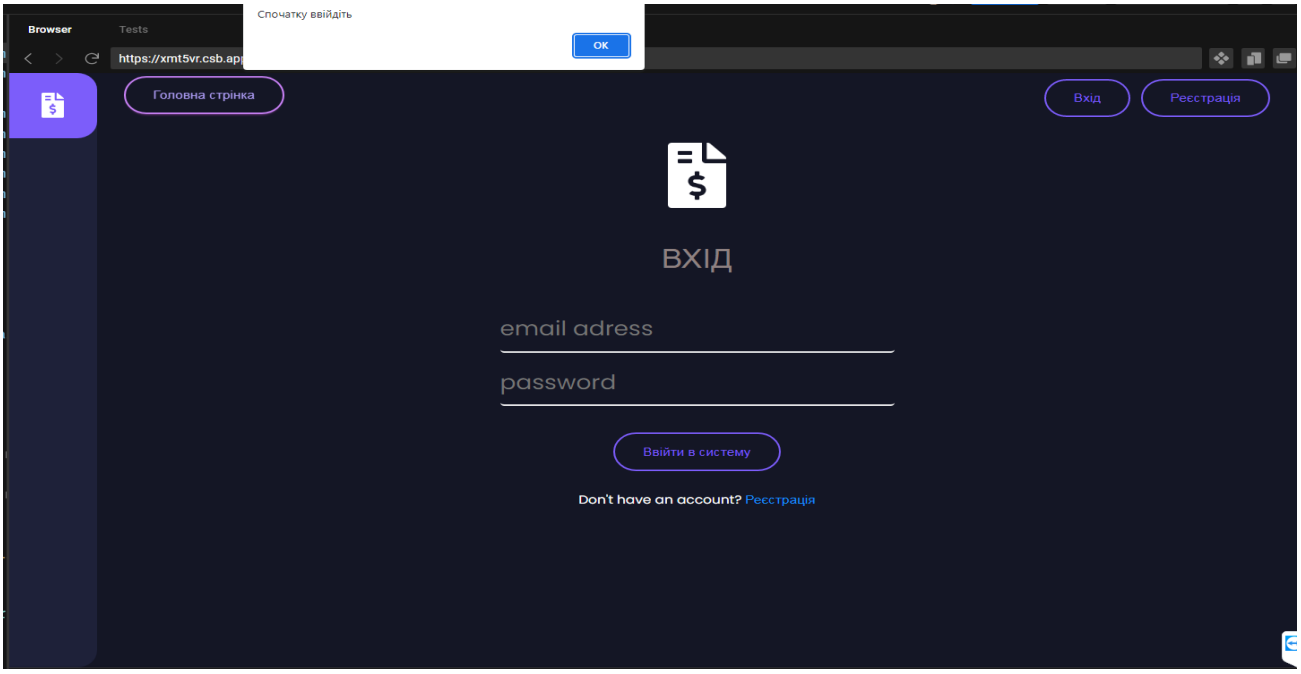


Рис 4.2 вікно помилки входу.

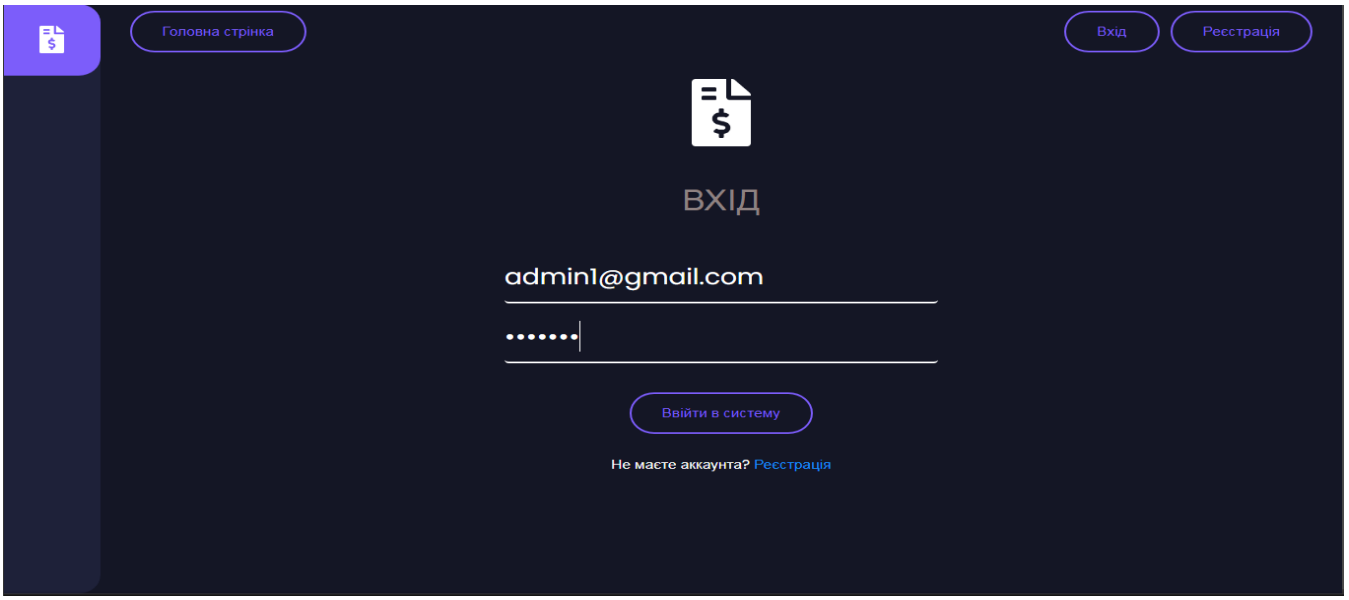


Рис 4.3 пропонується ввійти в систему

Можливості адміністратора

Після того як користувач увійшов в свій акаунт, він може побачити інформацію про свої замовлення перевезень.

Користувач бачить список замовлень, а також основні данні з них (ПІБ замовника, дата прибуття товару, оголошена вартість товару і стан вантажу).

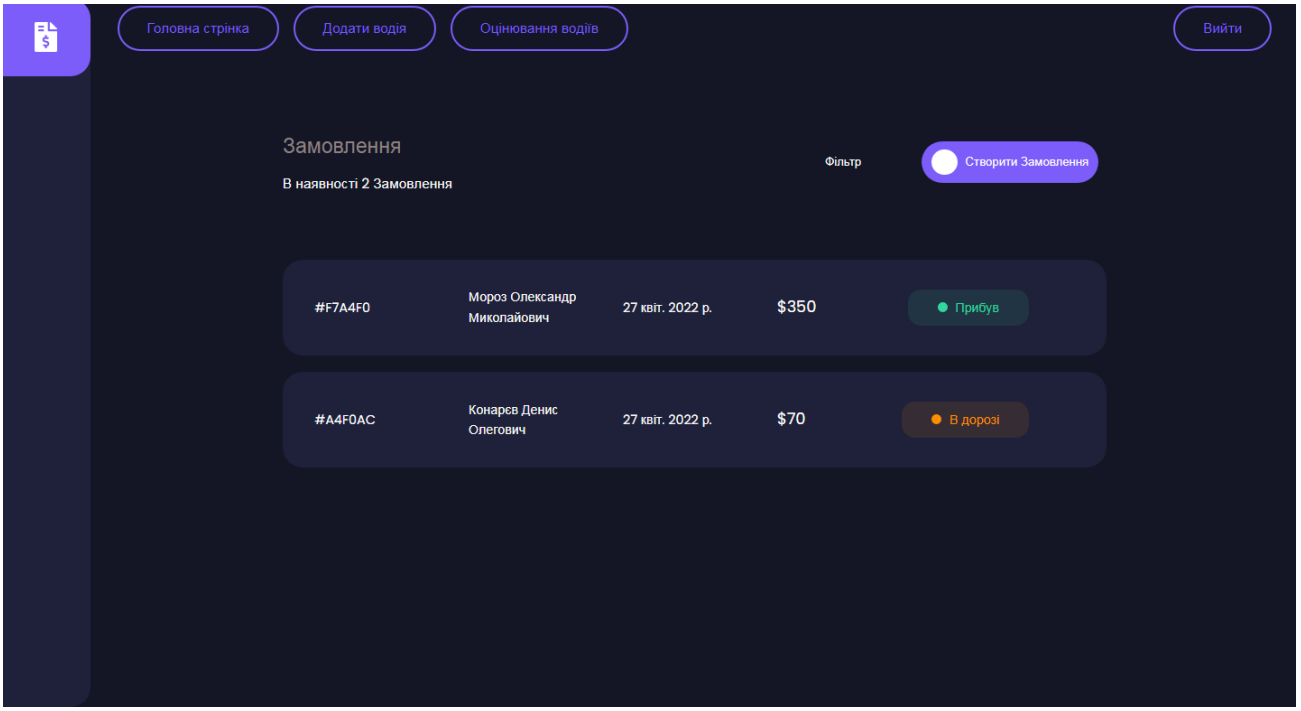


Рис. 4.4 Сторінка «Замовлення».

На Головній сторінці можливо створити нове замовлення, для цього потрібно натиснути кнопку ”створити замовлення”.

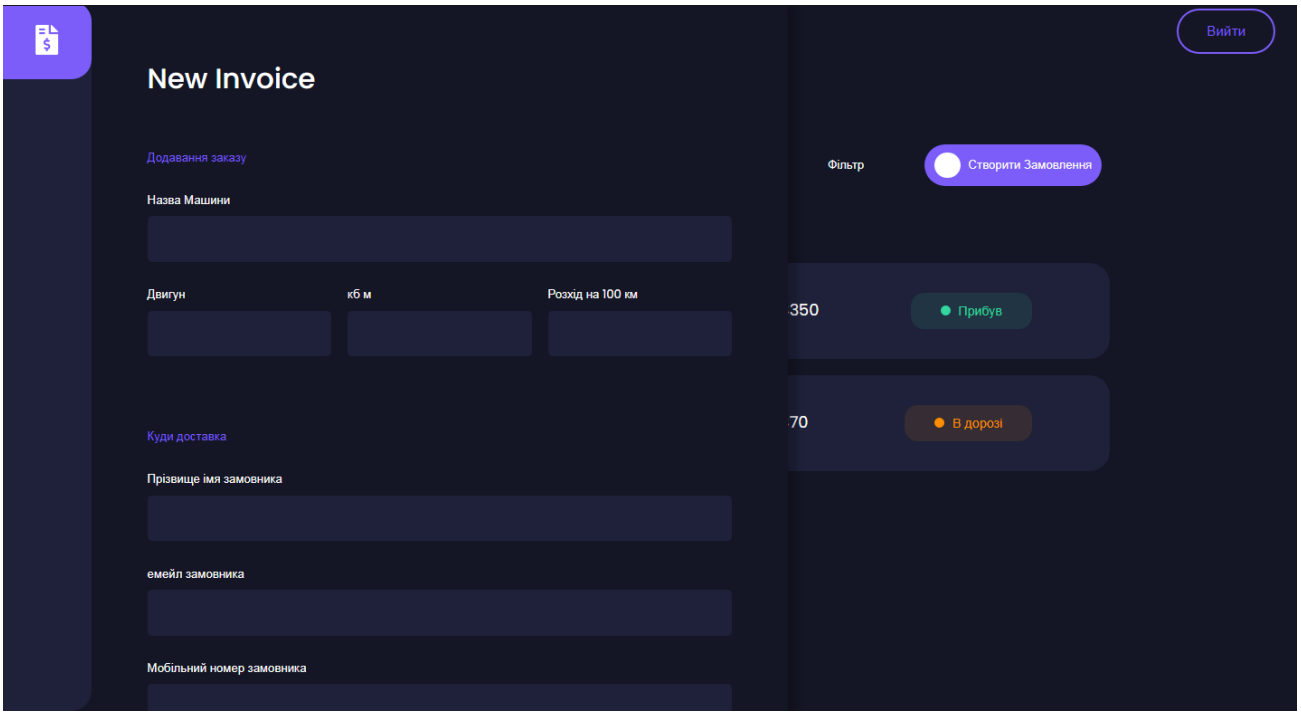


Рис. 4.5 Сторінка «Створення замовлення».

Також на сторінці створення замовлення ми можемо додавати товар у наш вантаж для цього натискаємо кнопку “Додати новий вантаж” і вписуємо у поля назву вантажу кількість одиниць і ціну за одиницю і отримуємо загальну суму.

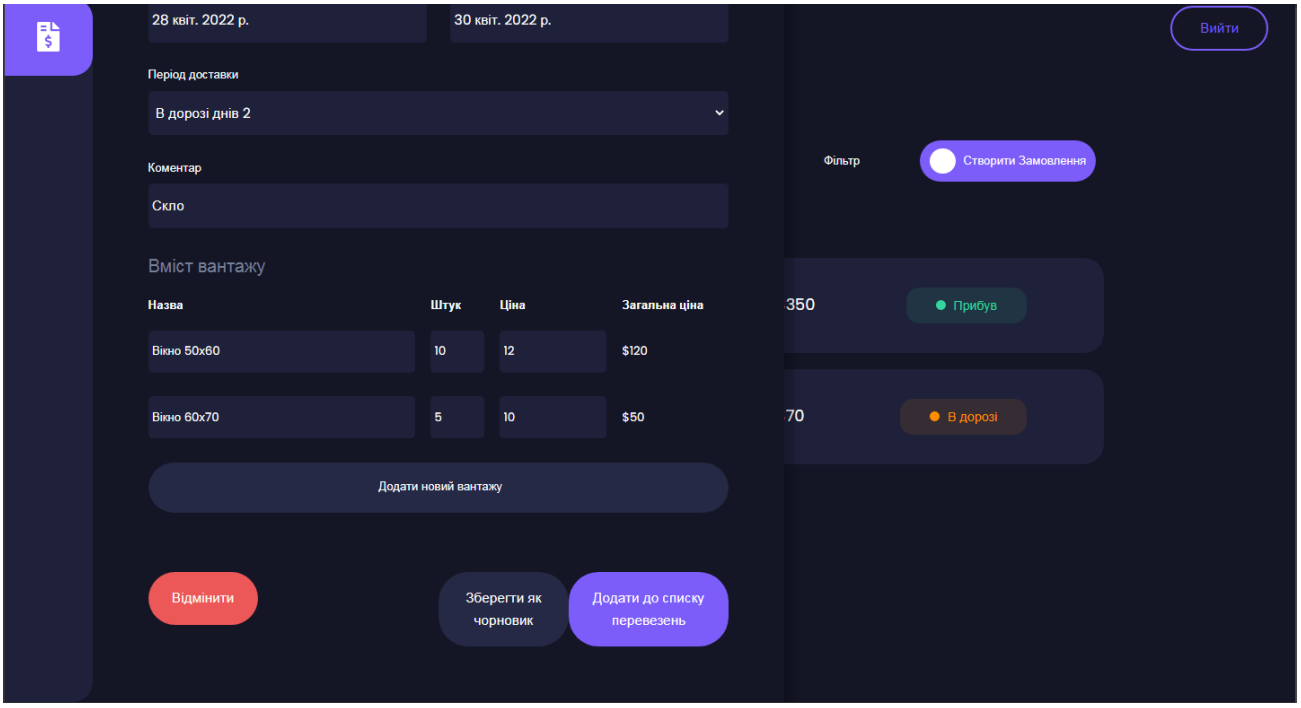


Рис. 4.6 Сторінка «Створення замовлення та додавання вантажу».

Отже, після того як ми заповнили усі поля, ми можемо або додати вантаж у список замовлень, або ж зберегти його як чорновик, для подальшого додавання у список замовлень.

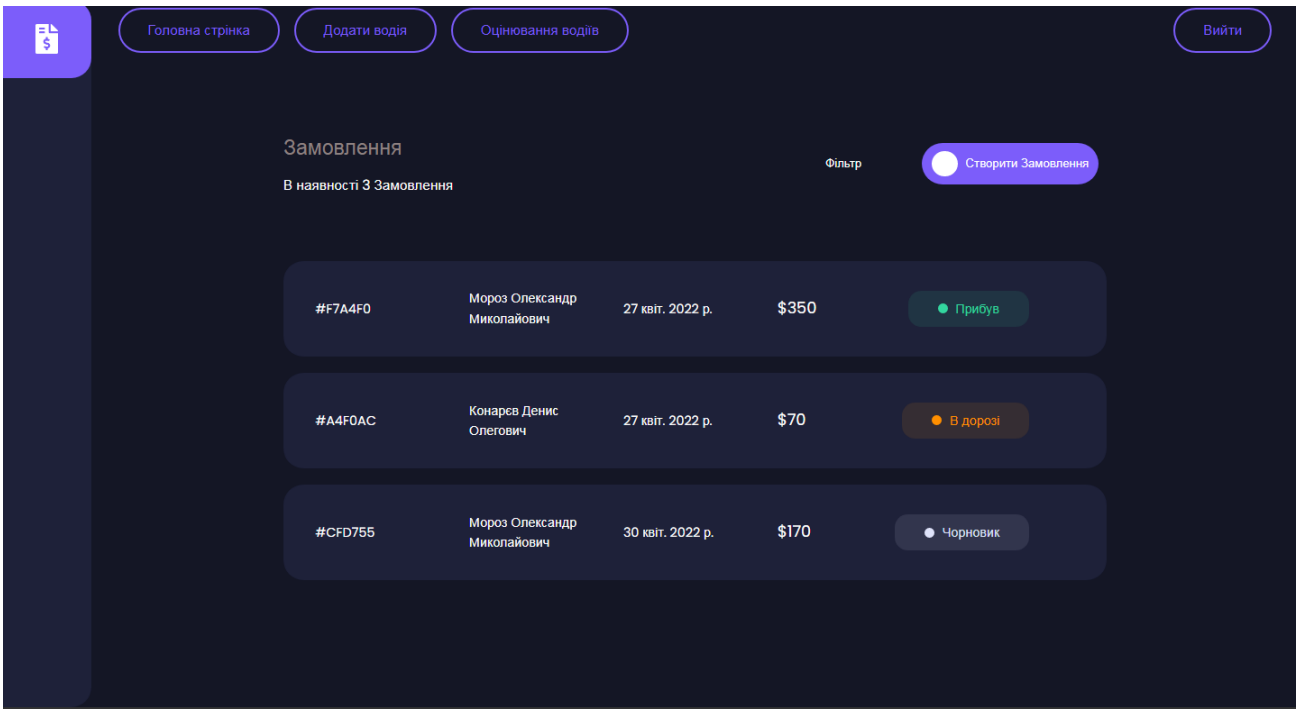


Рис. 4.7 Сторінка «Додавання замовлення до чорновика».

Якщо ми випадково натиснули у іншій частині сайту для виходу, з’явиться вікно, для запиту чи ми точно хочемо покинути сторінку “створення замовлень”.

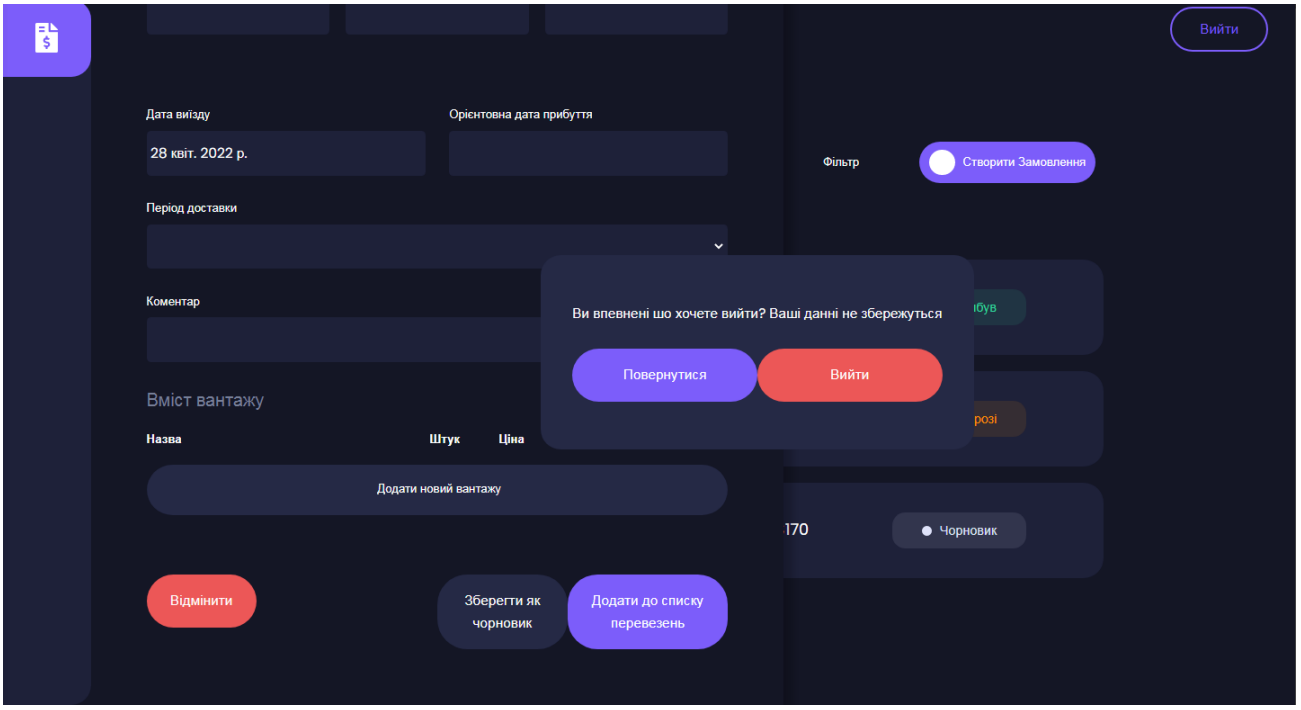


Рис. 4.8 Сторінка « Запит для виходу».

Також при натисканні на замовлення ми переходимо на сторінку Детальна інформація. Натиснувши кнопку “Відредагувати ” з’явиться сторінка з полями для редагування.

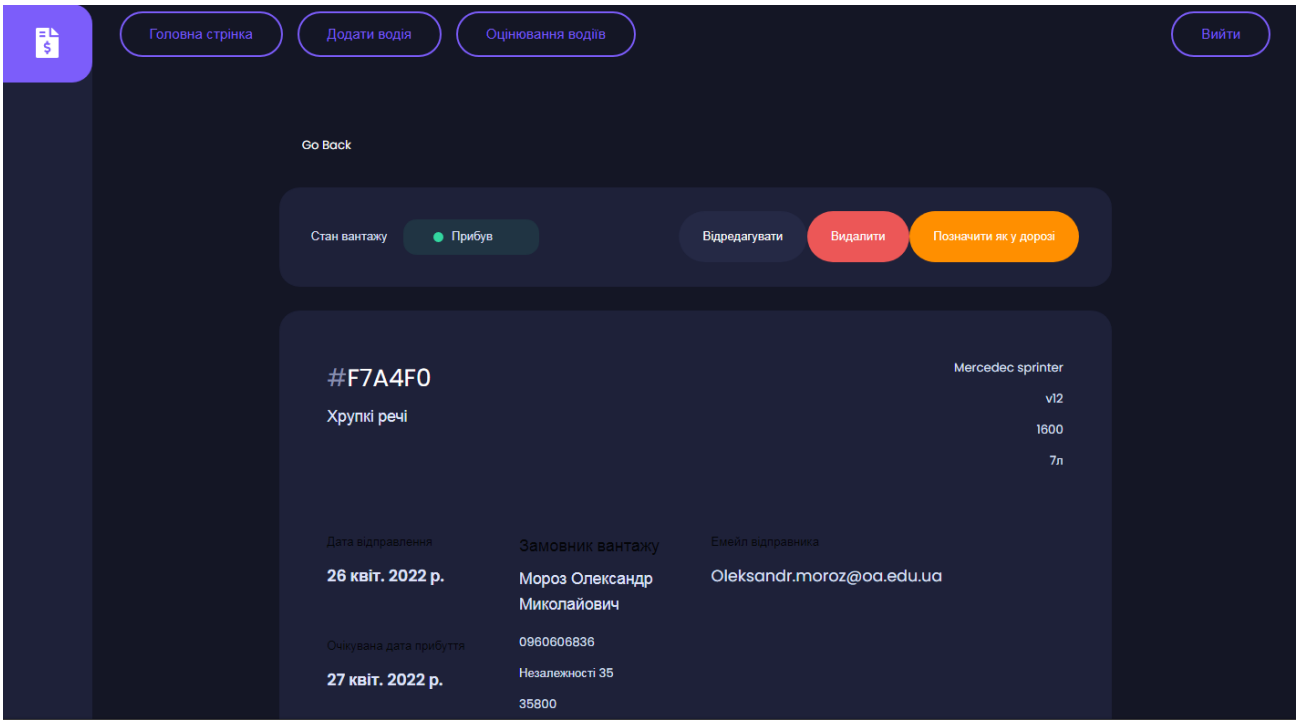


Рис. 4.9 Сторінка «детальніше».

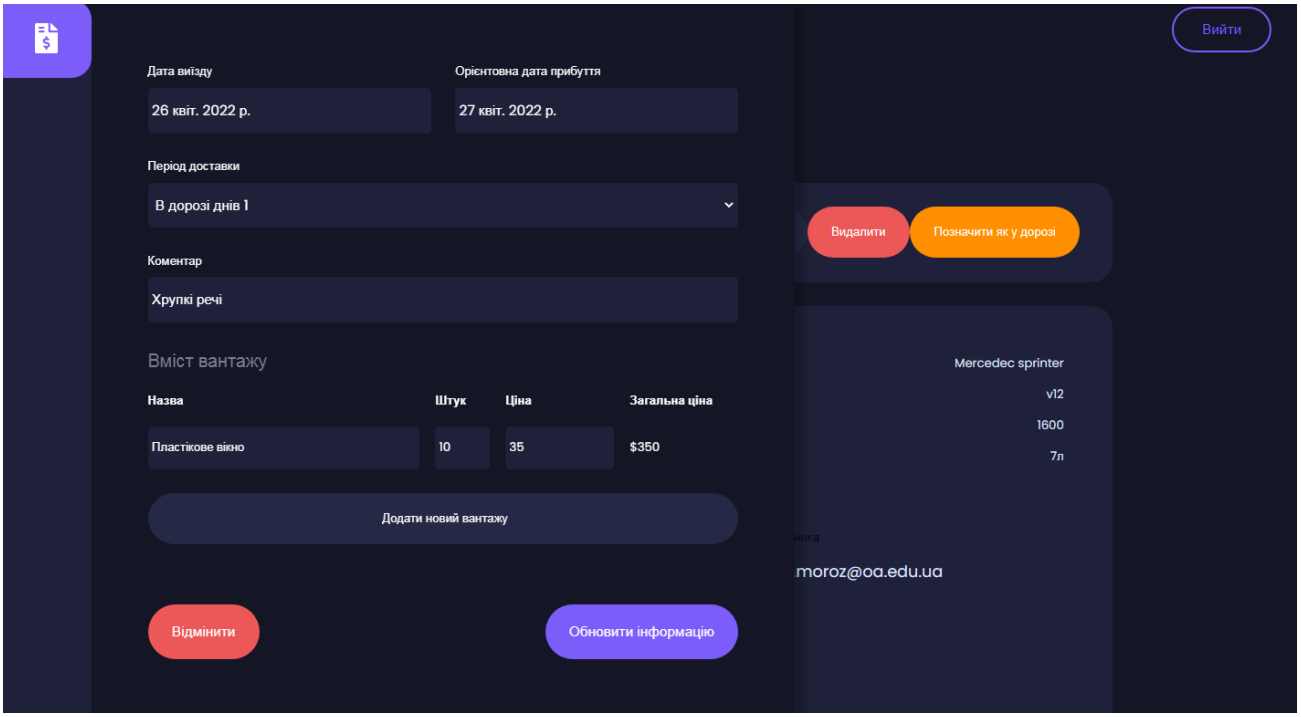


Рис. 4.10 Сторінка «Редагування».

Натиснувши обновити інформацію, відредагована інформація приміниться.

Також на сторінці редагування ми можемо змінити стан вантажу (В дорозі або прибув) Для цього потрібно натиснути кнопку “Позначити як прибув” для стану-прибув, в іншому випадку натиснути кнопку “Позначити як В дорозі” для стану – В дорозі.

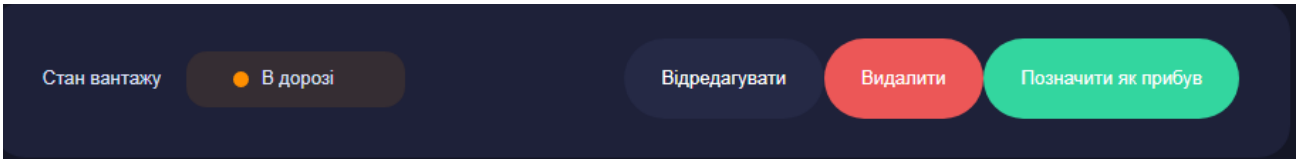


Рис. 4.11 «Стан в дорозі».

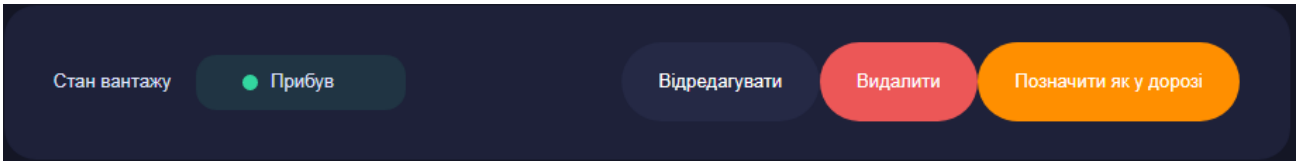


Рис. 4.12 « Стан Прибув».

Для додавання водіїв у компанію потрібно натиснути на кнопку “Додати водія”.

На цій сторінці нам потрібно ввести дані водія (Прізвище, Ім’я, По-Батькові), а також вибрати категорію водійського посвідчення наявного у цього водія.

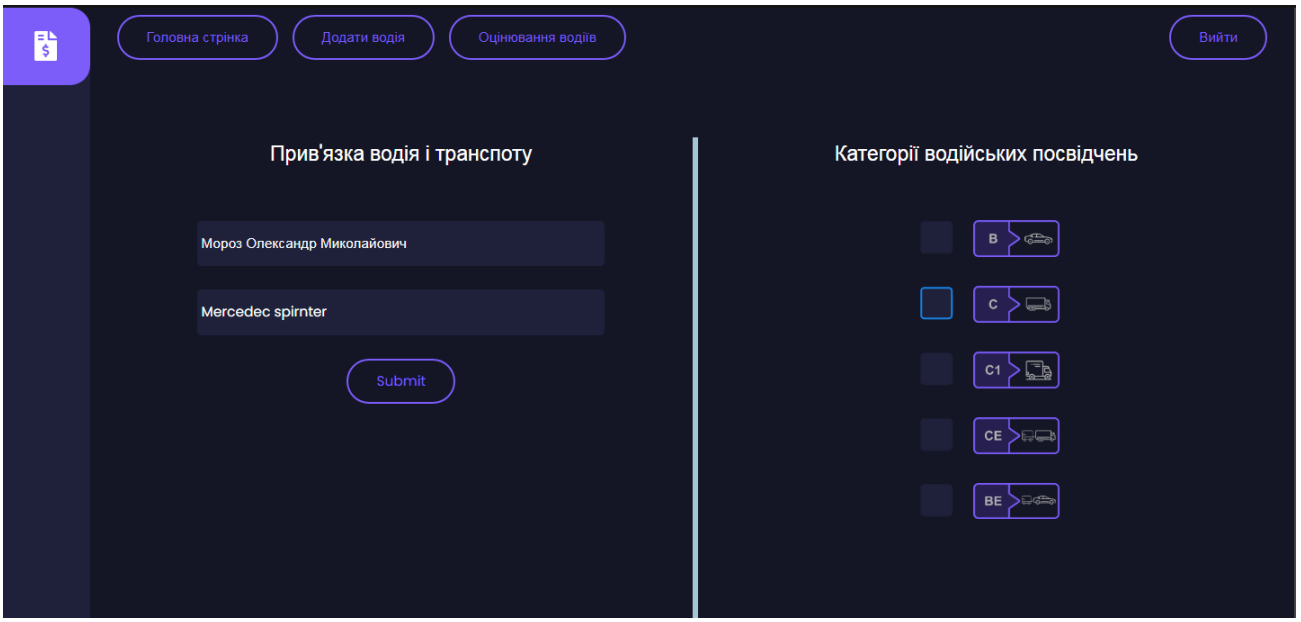


Рис. 4.13 Сторінка “Додати водія”.

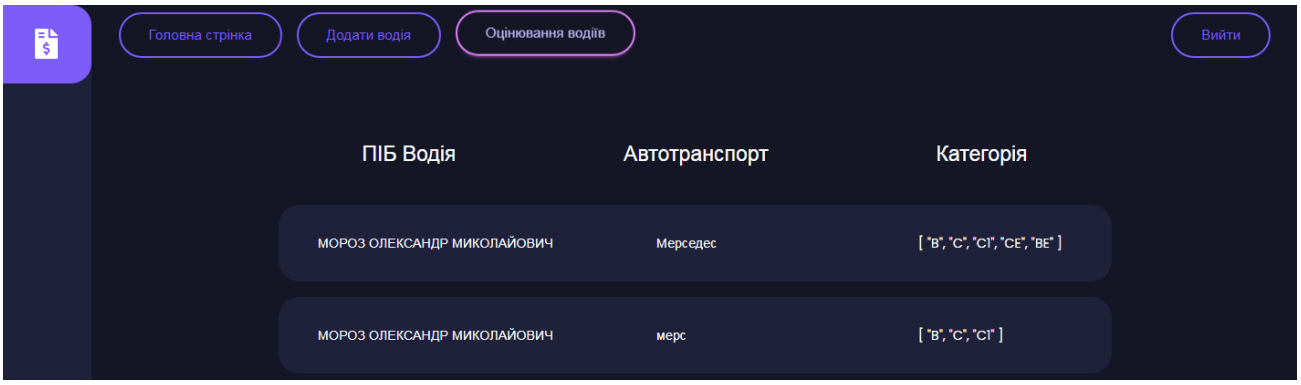


Рис. 4.14 Сторінка “Зв’язку Водія та Авто”.

На сторінці “Зв’язку водія та авто” ми бачимо на якому авто переміщується водій та з якою категорією водійського посвідчення. Натиснувши на водія чи авто в низу сторінки з‘являється редагування цього зв’язку. Для поновлення інформацію натискаємо кнопку “Поновити”

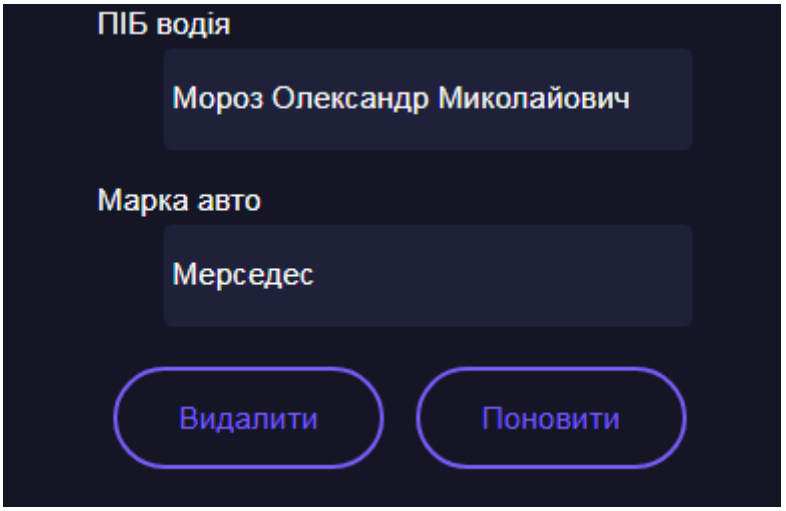


Рис. 4.15 Розділ «Редагування зв’язку водія та авто».

Після проведення усіх операцій натискаємо на кнопку “Вихід” яка знаходиться у верхньому правому куті.

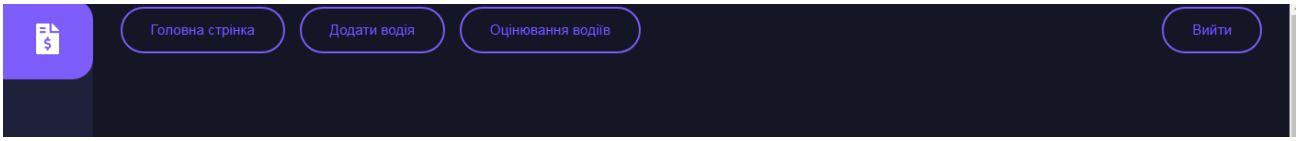


Рис. 4.16 «Вихід з додатку».

Тестування

Так як проект не є великим, отже було прийнято рішення використовувати ручне тестування

Ручне тестування - це тип тестування програмного забезпечення де тестуються випадки вручну без використання будь-яких засобів автоматизації. Тому тестувальники фактично сидять за екраном програми, вводять тестові випадки і дивляться, яким буде результат. Таким чином, ручне тестування є найпримітивнішою формою тестування програмного забезпечення. Ми використали його для пошуку помилки в додатках та програмних системах. Як мінімум, ми повинні спочатку протестувати кожну нову або модифіковану програму, перш ніж ми зможемо виконувати автоматичне тестування. Таким чином ми визначаємо можливість тестування програмного забезпечення. Ручне тестування програмного забезпечення вимагає більше зусиль для перевірки роботи програми.

Під час такої перевірки ми використовували codeSandbox для запуску окремих модулів програми . Для формування тест кейсів використовувався сервіс bugtracker zoho він дозволяє описувати та призначати відповідальних за усунення помилок.

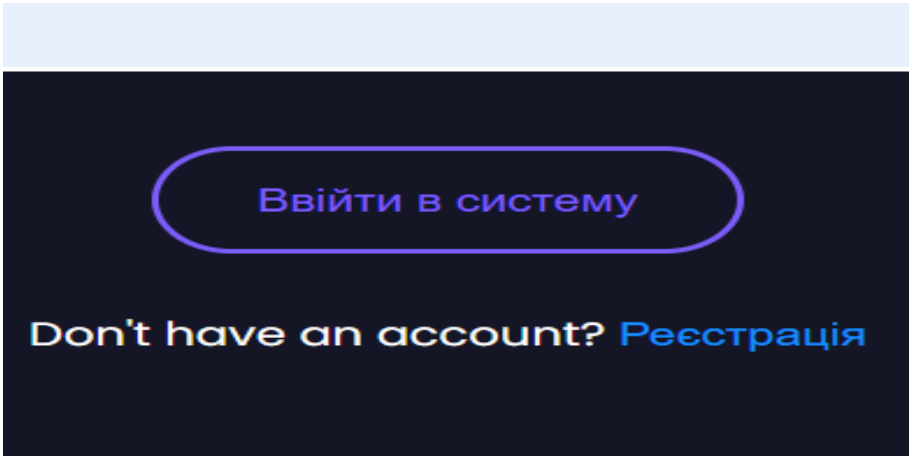


Рис. 4.17 «приклад текстового багу».

Після того як баг був виявлений ми оформляємо цей баг відповідно до вимог на Zoho Bugtracker.

Вибираємо кому ми надсилаємо цей баг репорт ,в нашому випадку це наш розробник фронт-енд частини, вибираємо дату коли замітили баг, коли цей баг відтворюється(завжди , чи за певних обставин).

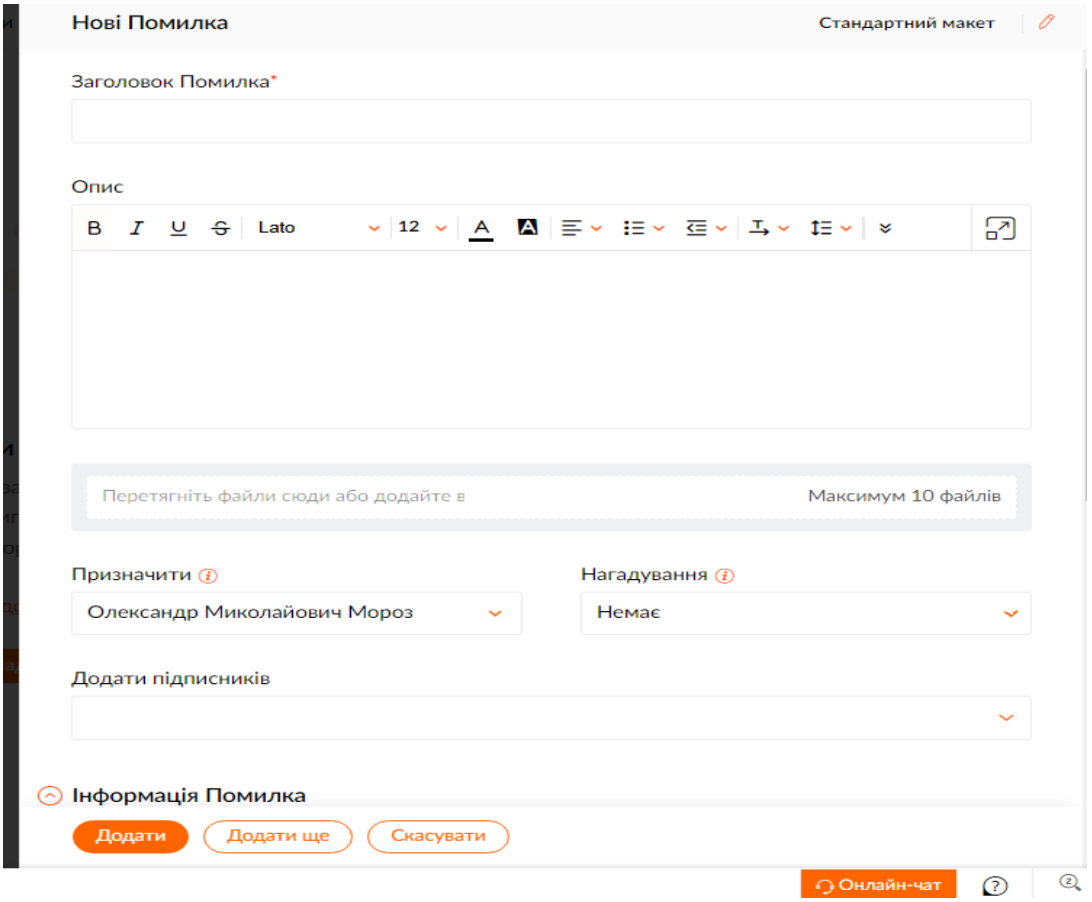


Рис. 4.18 «приклад текстового багу».

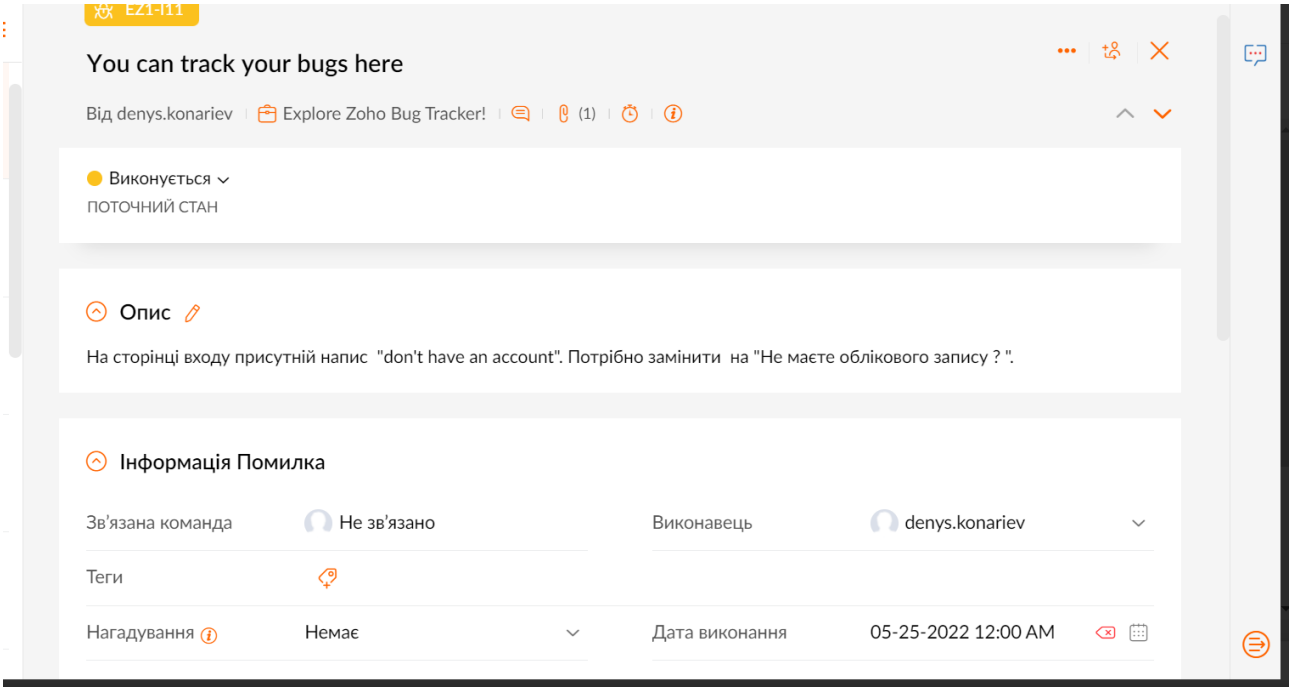


Рис. 4.19 «приклад готового тестового багу ».

Після опису багу команда приступала до ліквідування помилки. Якщо баг був виправлений ,статус роботи позначався як завершений.

Рис. 4.19 «Успішно усунутий баг ».

Висновок до розділу 3

Отже, в даному розділі ми описали засоби, які використовуємо у нашому проекті, а також визначилися чому ці засоби найкраще для нас підходять. Javascript є найбільш популярна мова програмування для створення динамічних за стосунків. У розділі опис програмної реалізації ми детально поговорили в про моделі, які використовуються у нашому проекті, а також дослідили переваги та недоліки програмних засобів які були задіяні. Ми використали сервіс Firebase який з легкістю дав нам можливість реалізації Back-end без відволікання на неї.

Загальні Висновки

В даній дипломній роботі був створений додаток транспортної компанії HTTP за допомогою WEB-технологій, яка була створена на фреймворку Vue js. Ця система буде розрахована на створення вантажних замовлень, їх контролю та виконання. Додавання водіїв які працюють в цій компанії створення їх особової карти для того щоб бачити прізвище та ім’я водія і водійську категорію. А також для створення зв’язку між транспортним засобом та водієм який на ньому пересувається. Система має ергономічний та досить вдало підібраний дизайн та структуру. Користувачів веб-додатку має лише одну роль адміністратор. Адміністратор керує серверною частиною сайту, базою даних. Розроблюючи даний додаток було вирішено такі задачі: досліджено та проаналізовано існуючі аналоги, розроблено оригінальний дизайн, вивчено механізм роботи на фреймворку Vue js, змодельована структура бази даних, розроблено власний демонстраційний продукт.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. HTTPS-Vue.js URL: https://vue3js.cn/vuex/ru/#что-такое-“паттерн-управления-состоянием” (дата звернення 12.03.2022).

2. Що таке MVC URL: https://uk.education-wiki.com/3886982-what-is-mvc (дата звернення 12.03.2022).

3. Вікіпедія Model-View-ViewModel URL: https://uk.wikipedia.org/wiki/Model-View-ViewModel (дата звернення 14.03.2022).

4. Вікіпедія Git Hub URL: https://uk.wikipedia.org/wiki/GitHub (дата звернення 5.04.2022).

5. Вікіпедія Visual Studio Code URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code (дата звернення 05.04.2022).

6. Хабр URL: https://habr.com/ru/company/simbirsoft/blog/416925/ (дата звернення 07.04.2022).

7. Вікіпедія Firebase URL: https://uk.wikipedia.org/wiki/Firebase (дата звернення 07.04.2022).

8. Вікіпедія Node.js URL: https://uk.wikipedia.org/wiki/Node.js (дата звернення 07.04.2022).

9. Вікіпедія Bootstrap URL: https://uk.wikipedia.org/wiki/Bootstrap (дата звернення 08.04.2022).

10. Вікіпедія пісочниця URL: https://uk.wikipedia.org/wiki/Пісочниця_(комп%27ютерна_безпека) (дата звернення 08.04.2022).

11. 1С логістика URL: https://solutions.1c.ru/catalog/tms/features (дата звернення 10.04.2022).

12. SQl vs Nosql URL: https://uk.myservername.com/sql-vs-nosql-exact-differences(дата звернення 14.04.2022)