

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-наукової програми «Інформаційні технології проектування»

на тему: Web-орієнтована система підтримки роботи відділу продажів ІТ компанії

Здобувача (ки) групи ІТ.м-23. Лютого Дмитра Антоновича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Лютий Дмитро Антонович
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри інформаційних технологій,
доцент, к.т.н. Чібирик Я.І.

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ С. М. Ващенко
«___» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Лютий Дмитро Антонович

(прізвище, ім'я, по

батькові)

1 Тема проекту Web-орієнтована система підтримки роботи відділу продажів ІТ компанії

затверджена наказом по університету від «05» вересня 2023 р. № 0465-VI

2 Термін здачі студентом закінченого проекту «7» _____ грудня _____ 2023 р.

3 Вхідні дані до проекту _____ результати роботи web-орієнтованої системи _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) _____ аналіз предметної області, постановка задачі, методи дослідження, проектування web-орієнтованої системи підтримки роботи відділу продажів ІТ компанії, проектування та розробка web-орієнтованої мережі

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____ актуальність проблеми, мета проєкту, постановка задачі, функціональні вимоги до проєкту, контекстна діаграма IDEF0, декомпозиція IDEF0, схема інформаційної технології, діаграма варіантів використання, фізична модель даних, засоби реалізації, практична реалізація, висновки _____

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Огляд технологій, що застосовуються для підтримки роботи відділу продажів	до 23.09.2023	
2	Постановка задачі та формування завдань дослідження	до 04.10.2023	
3	Структурно-функціональне моделювання	до 25.11.2023	
4	Реалізація додатку з використанням Yii2, CodIgniter та MySQL	до 01.12.2023	
5	Оформлення пояснювальної записки до кваліфікаційної роботи	до 07.12.2023	

Магістрант _____

Лютий Д.А.

Керівник роботи _____

к.т.н., доц. Чібиряк Я.І.

АНОТАЦІЯ

Записка: 85 сторінок, 58 рисунків , 4 додатки, 30 використаних джерел та літератури.

Обґрунтування актуальності теми роботи: Розробка веб-систем підтримки для роботи відділу ІТ-продажів компанії обумовлена стрімким розвитком інформаційних технологій і необхідністю оптимізації процесу управління продажами ІТ-відділом. Зміни ринкових умов і конкурентний тиск вимагають від ІТ-компаній впровадження новітніх інструментів для підвищення ефективності своїх відділів продажів. Розробка системи підтримки відділу продажів на веб-платформі допоможе вирішити ці проблеми і підвищити конкурентоспроможність бізнесу.

Мета роботи: розробка web-орієнтованої системи підтримки роботи відділу продажів ІТ компанії.

Методи дослідження: аналіз та впровадження сучасних веб-технологій, вивчення та впровадження кращих практик управління продажами

Результат : Система, призначена для підтримки роботи відділу продажів, що забезпечує оптимізацію внутрішніх процесів, автоматизацію завдань, що полегшує взаємодію між співробітниками відділу, з використанням цієї системи, призведе до підвищення ефективності відділу продажів, кращого моніторингу запитів клієнтів і більш ефективної декомунізації клієнтської бази.

Ключові слова: web-система, sales, Yii2, CodeIgniter MySQL, Sales, Framework

ЗМІСТ

ВСТУП	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень та публікацій.....	8
1.2 Аналіз програмних продуктів – аналогів	12
1.3 Аналіз технологій та засобів реалізації web-системи	17
1.4 Постановка задачі	29
2 МОДЕЛЮВАННЯ РОБОТИ WEB-ДОДАТКУ	31
2.1 Структурно-функціональне моделювання.....	31
2.2 Моделювання варіантів використання	33
2.3 Проектування бази даних	35
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ	37
3.1 Ініціалізація проекту та інтеграція БД з web-системою.....	37
3.2 Реалізація серверної частини.....	42
3.3 Приклад роботи web-орієнтованої системи	45
ВИСНОВОК	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТОК А	60
ДОДАТОК Б	68
ДОДАТОК В	72
ДОДАТОК Г	82

ВСТУП

З останніми технологічними проривами веб-системи стають новою технологією. Це програмний сервіс, який використовується для полегшення онлайн-комунікації між двома електронними пристроями. Властива веб-системам здатність бути самодостатніми та самоописуваними робить їх популярними у веб-додатках.

Сучасні системи стали невід'ємною частиною бізнес-процесів завдяки розвитку Інтернету. Для створення додатків використовується ціла низка комп'ютерних мов, зокрема Java, .NET, Node.js та інші. Веб-сервіси дозволили додаткам, написаним різними мовами програмування, спілкуватися та обмінюватися даними один з одним. Використовується загальний код, зрозумілий усім додаткам, а не специфічний для платформи код, зрозумілий лише одному додатку.

Веб-системи переводять корпоративні процеси в онлайн. Вони можуть бути доступними з будь-якого місця в Інтернеті, оскільки надають функціональні можливості клієнтським програмам через HTTP. Веб-сервіси дуже корисні для забезпечення необхідної функціональності для компаній, які перенесли свою діяльність в Інтернет.

Компанії можуть звільнитися від складної, довготривалої та дорогої розробки програмного забезпечення і натомість зосередитися на завданнях з доданою вартістю та критично важливих для їхньої місії. Без зміни коду веб-сервіси, створені з внутрішніх додатків, можуть бути легко доступні для зовнішнього використання. Інкрементальна розробка за допомогою веб-сервісів є природною і простою, а оскільки веб-сервіси оголошуються і реалізуються у зрозумілому для людини форматі, виявлення і виправлення помилок спрощується. Простота інтеграції застарілих систем призводить до підвищення гнучкості та оперативності, а також до значної економії часу та коштів.

Метою дослідження є створення web-орієнтована система підтримки роботи відділу продажів ІТ компаній. Для досягнення мети необхідно вирішити наступні задачі.

- аналіз існуючих web-систем для управління продажами в ІТ галузі;
- збір і систематизація вимог для відділу продажів ІТ-компанії;
- визначення програмних засобів реалізації;
- розробка архітектури та структури web-системи;
- розробка та тестування web-системи

Об'єкт дослідження. Інформаційна підтримка роботи відділу продажів ІТ компаній.

Предмет дослідження. Інформаційна технологія підтримки роботи відділу продажів ІТ компаній.

Практична значимість. Значимість даної системи полягає в організації процесу підтримки роботи відділу продажів ІТ компаній за допомогою системи обробки даних фреймворку Yii2, а також організація процесу збереження бази даних системи шляхом генерації файлу БД під актуальні системи СУБД. Автоматизація генерації графічних даних на основі замовлень та додавання послуг.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень та публікацій

Веб-сервіси відносяться до сімейства технологій, які можуть універсально стандартизувати зв'язок між додатків з метою економічно ефективного з'єднання систем, ділових партнерів і клієнтів через всесвітню мережу Інтернет [1].

Найбільші виробники програмного забезпечення, такі як IBM, Microsoft, SAP, SUN і Oracle, впроваджують стандарти веб-сервісів і випускають продукти та інструменти з підтримкою веб-сервісів. Веб-сервіси полегшать обмеження часу, витрат і простору для пошуку, ведення переговорів і здійснення транзакцій в електронному бізнесі. Як результат, веб-сервіси змінюють спосіб, у який компанії розробляють свої додатки як сервіси та інтегруються з іншими бізнес-структурами, керують бізнес-процесами і т.д [2].

В статті Minder Chen, Andrew N. K. Chen та Benjamin B. M. Shao описується принцип інтеграції різних технологій з web-сервісами та вплив кінцевого продукту на бізнес процеси [3]. Автори акцентують на тому, що веб-сервіси покладаються на набір стандартів для підтримки сумісності між додатками, розробленими різними мовами і працюють на різних платформах або операційних системах. Один із способів зрозуміти процес роботи веб-сервісу – це розуміння стандартів веб-сервісів. Основні стандарти веб-сервісів включають SOAP (Simple Object Access Protocol - простий протокол доступу до об'єктів), WSDL (мова опису веб-сервісів), UDDI (універсальний опис, виявлення та інтеграція) та інші [4].

Веб-сервіси є природним наслідком еволюції Інтернету у відкрите середовище, яке полегшує складну взаємодію між бізнес-додатками та інформаційними технологіями. Веб-сервіси займаються проблемами забезпечення систематичної взаємодії між додатками через Інтернет та інтеграції існуючої мережевої комп'ютерної інфраструктури в Інтернет. В

статті Santosh S. Venkatraman автор зосередив увагу на бізнес процесах та можливостях розширення власного ПО для цільової аудиторії [5].

Нові ділові партнерства та інтегровані програмні додатки можуть бути пов'язані динамічно та автоматично "за потребою", оскільки веб-сервіси забезпечують повну функціональну сумісність розрізнених систем. Наприклад, компанія Johnson Controls з Мілуокі, яка виробляє системи клімат-контролю, нещодавно оголосила про нову ініціативу веб-сервісів, яка використовує Microsoft .Net для інтеграції систем, які ніколи не були пов'язані раніше не були пов'язані між собою. Johnson Controls тепер може пов'язати шкільний календар зі своєю системою клімат-контролю так, що температура в класі під час шкільних канікул може бути запрограмована на певний час. Температуру в класі під час шкільних канікул можна запрограмувати з будь-якого пристрою з доступом до Інтернету [6].

Таким чином, завдяки цій новій бізнес-моделі з'являються нові бізнес-можливості для продажу програмного забезпечення, його розробки, постачання та впровадження.

Web-орієнтовані системи підтримки роботи відділу продажів дозволяють збирати та аналізувати дані про активність клієнтів, що дає змогу краще зрозуміти їхні потреби, вподобання та поведінку. Маючи цю інформацію, є можливість адаптувати пропозиції, щоб краще відповідати потребам, що може призвести до підвищення рівня задоволеності та лояльності клієнтів.

Також дані системи корисні в сфері аналізу рівня участі співробітників і типи програм, які користуються найбільшою популярністю. Результати роботи системи можуть допомогти розробити більш ефективні програми в майбутньому та адаптувати їх до потреб цільової аудиторії [7].

Для реалізації даної задачі існують дві технології:

- клієнт-серверна архітектура;
- хмарна архітектура;

В архітектурі клієнт-сервер користувач входить на сервер, перевіряючи свою ідентифікацію за допомогою облікових даних, збережених на сервері, а не на локальному комп'ютері, ще до того, як отримати доступ до операційної системи свого комп'ютера. Тоді як доступ до хмари зазвичай відбувається без необхідності вручну вводити облікові дані, після того, як користувач увійшов на комп'ютер або інший пристрій, використовуючи локально збережені облікові дані. Обидва способи забезпечують зберігання необхідних файлів на комп'ютері користувача [8].

Архітектура клієнт-сервер зазвичай розгортається в організаціях, де контроль над комп'ютером користувача і доступом до нього, наприклад, централізовано зберігаються облікові дані користувача, оновлення операційної системи або оновлення користувацьких додатків, адмініструються і спрямовуються централізовано.

Хмарне сховище може бути прозорою функцією архітектури клієнт-сервер, хоча протилежне не є вірним, тобто архітектура клієнт-сервер не є підфункцією хмарного сховища. Залежно від хмарної архітектури, не можна сказати наскільки вона безпечна, або чи є доступ до даних користувача дійсно безпечним [9].

Основна відмінність хмарних обчислень від традиційних мереж або хостингу полягає в реалізації. Віртуалізація забезпечує широкі можливості масштабування, надаючи клієнтам практично безмежні ресурси.

У традиційній мережі сервер встановлюється в апаратному забезпеченні, і якщо є потреба масштабувати його до більшої кількості користувачів, ніж може підтримувати поточне обладнання, потрібно буде виділити більше грошей на оновлення, і все одно буде обмеження. Але в інфраструктурі хмарних обчислень кілька серверів вже є на місці на старті, потім вони використовують віртуалізацію, щоб надавати тільки ті ресурси, які потрібні конкретному користувачеві, що дає їй велику масштабованість від невеликих потреб ресурсів особистого бізнесу до великих корпоративних потреб у ресурсах. Хмарний провайдер може без проблем масштабувати ресурси, і

клієнту потрібно платити лише за те, що він використовує. У традиційній мережі потрібно платити за все: за обладнання, його встановлення, обслуговування або навіть просто орендувати його за щомісячну фіксовану ціну, навіть якщо вам потрібен лише невеликий обсяг ресурсів [10].

Таким чином, хмарна архітектура є або може бути просто ще одним різновидом архітектури клієнт-сервер, де користувач ізольований від клієнт-серверних аспектів її реалізації. Все залежить від того, хто контролює хмару і про яку хмару йдеться. Віддалене зберігання даних користувача, які модифікуються локально і доступні користувачеві незалежно від того, яку платформу він використовує для доступу до них.

Для реалізації даної задачі було обрано клієнт-серверну архітектуру через наявність в системі авторизації надання доступу за допомогою особистих даних користувача та хешуванням даних. Також доречно використовувати дану архітектуру через систему експорту бази даних шляхом виконання запитів СУБД та генерації файлу обраної БД зі збереженням на ПК користувача.

1.2 Аналіз програмних продуктів – аналогів

Для більш детального аналізу було обрано наступні web-орієнтовані системи системи, які мають найбільшу популярність та надають максимальний функціонал для аналізу даних та збору інформації:

- Zoho;
- Creatio;
- LP-System;

Zoho — це система управління взаємовідносинами з клієнтами яка ідеально відповідає потребам малого бізнесу завдяки простоті використання та невисокій ціні [11]. При цьому у даної системи менше можливостей для індивідуального налаштування. Деякі додаткові функції можна надбудувати, придбавши розширення в Zoho Marketplace. Це дозволить підлаштувати систему до індивідуальних потреб бізнесу й обійтися при цьому без розробників.

Функції Zoho:

- Базові інтеграції не потребують участі програмістів
- Різнобічна аналітика
- Прогнозування
- Сповіднення в режимі реального часу

На рисунку 1.1 представлено вигляд web-системи Zoho.

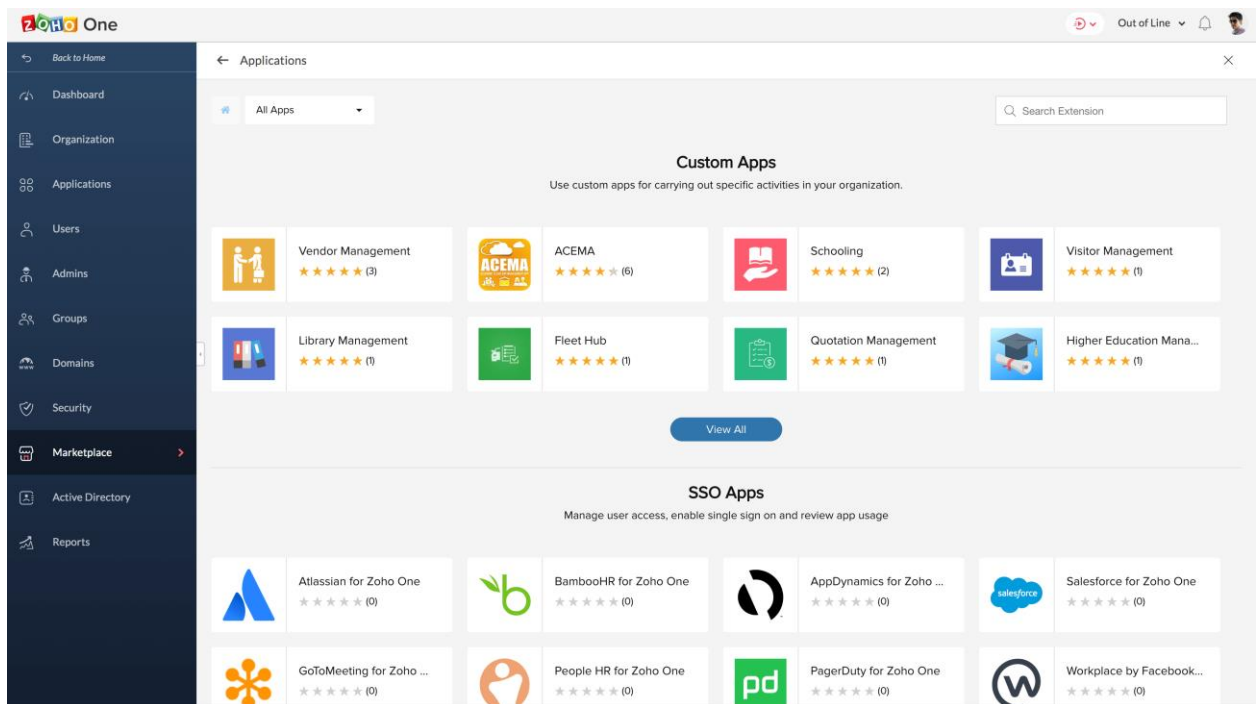


Рисунок 1.1 – Вигляд web - системи Zoho

Creatio — єдина хмарна web-система для великих та середніх компаній, яка допомагає об'єднати та прискорити процеси продажу, маркетингу та сервісу, а також операційні процеси організації. **Також** це платформа, яка містить функціонал для управління бізнес-процесами [12]. Компанією **Terrasoft** розроблено три основні продукти: для управління продажами, маркетингом і сервісом — Sales Creatio , Marketing Creatio , Service Creatio , Studio Creatio . Також клієнтам доступні десятки готових рішень в онлайн-каталозі Marketplace.

Перевагами Creatio є:

- наявність єдиної платформи;
- легкість та швидкість впровадження;
- гнучкість кастомізації та можливість самостійного налаштування;
- наявність інтелектуальних технологій;
- залучаючий інтерфейс.

На рисунку 1.2 представлено вигляд робочого середовища web-системи Creatio.

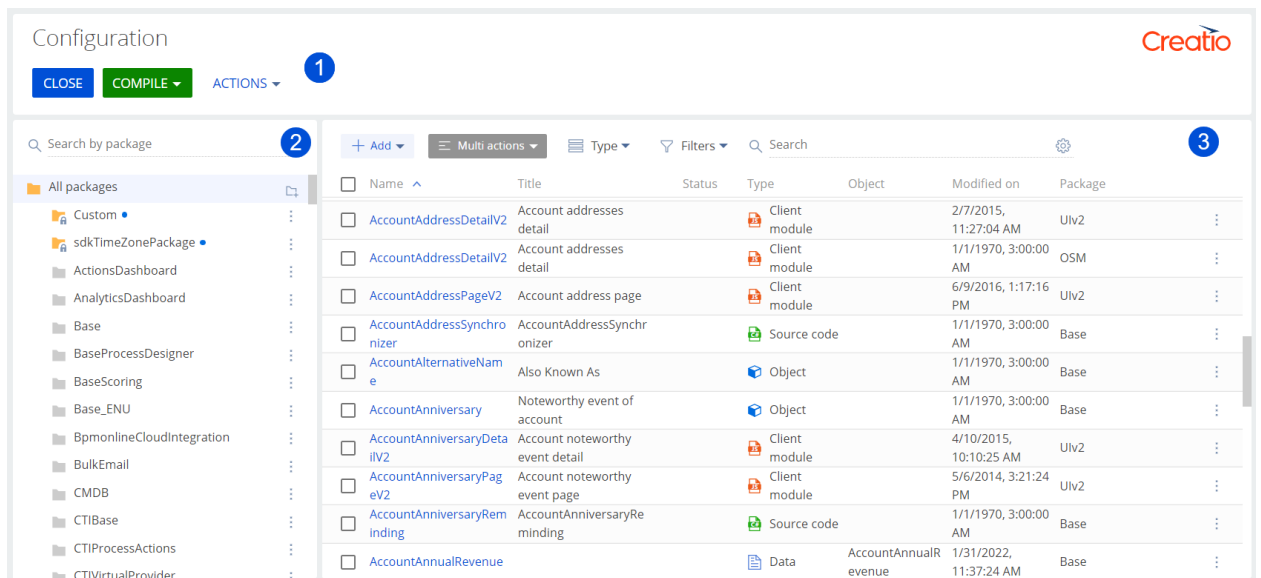


Рисунок 1.2 - вигляд робочого середовища web-системи Creatio

LP-System — web-орієнтована система для товарного бізнесу, яка створювалася з метою особистого користування [13].

Переваги LP-System:

— Легко слідкувати за замовленнями, їх статусами, а також одразу створювати ТТН, так як в LP-System наявні інтеграції із службами доставки: «Нова Пошта», «Укрпошта» та «Justin».

— Інтеграція з популярними українськими маркетплейсами Prom та Rozetka дозволяє переносити всі замовлення з них у систему без ризику їх втрати.

— Інтеграція з ПРРО Checkbox дозволяє створювати електронні чеки та одразу реєструвати їх у ДПС.

— Автозміна статусів замовлень. В залежності від статусу ТТН, замовлення буде автоматично перемішуватися по воронці продажів.

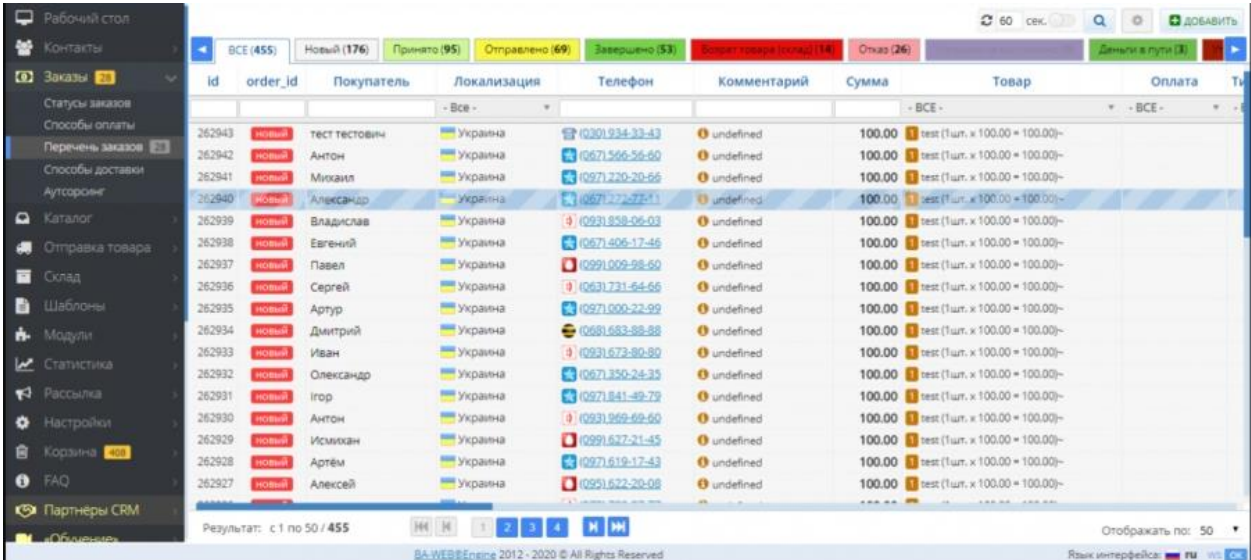
— Контроль доступів дає можливість обмежити інформацію для працівників.

— Завдяки інтеграції з сервісом «TurboSMS» легко контролювати, кому та скільки SMS відправляють працівники. Також Ви можете автоматизувати відправку повідомлень.

— В LP-System присутній свій склад, за допомогою якого зручно контролювати кількість товару, його рух та бачити, скільки товару знаходиться в замовленні.

— Відкрите API надає можливість поєднувати власні сайти із LP-System.

На рисунку 1.3 представлено вигляд робочого середовища web-системи LP-System.



The screenshot displays the LP-System web interface. At the top, there are filters for order status: ВСЕ (455), Новый (176), Принято (95), Отправлено (69), Завершено (53), В ожидании товара (114), and Отказ (26). Below the filters is a table with columns: id, order_id, Покупатель, Локализация, Телефон, Комментарий, Сумма, Товар, and Оплата. The table contains 17 rows of test data. A left sidebar menu includes options like Рабочий стол, Контакты, Заказы, Статусы заказов, Способы оплаты, Перечень заказов, Способы доставки, Авторизация, Каталог, Отправка товара, Склад, Шаблоны, Модули, Статистика, Рассылка, Настройки, Корзина, FAQ, and Партнеры CRM. The bottom of the page shows pagination: 'Результат: с 1 по 50 / 455' and 'Отображать по: 50'.

id	order_id	Покупатель	Локализация	Телефон	Комментарий	Сумма	Товар	Оплата
262943	новый	тест тестович	Украина	(030) 934-33-43	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262942	новый	Антон	Украина	(067) 566-56-60	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262941	новый	Михаил	Украина	(097) 220-20-66	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262940	новый	Александр	Украина	(067) 272-272-11	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262939	новый	Владислав	Украина	(093) 938-06-03	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262938	новый	Евгений	Украина	(067) 406-17-46	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262937	новый	Павел	Украина	(099) 009-98-60	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262936	новый	Сергей	Украина	(063) 731-64-66	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262935	новый	Артур	Украина	(097) 000-22-99	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262934	новый	Дмитрий	Украина	(068) 683-88-88	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262933	новый	Иван	Украина	(093) 673-80-80	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262932	новый	Олександр	Украина	(067) 350-24-35	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262931	новый	Ігор	Украина	(097) 841-49-79	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262930	новый	Антон	Украина	(093) 969-69-60	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262929	новый	Исмахан	Украина	(099) 627-21-45	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262928	новый	Артём	Украина	(097) 619-17-43	undefined	100.00	test (1шт. x 100.00 = 100.00)-	
262927	новый	Алексей	Украина	(095) 622-20-08	undefined	100.00	test (1шт. x 100.00 = 100.00)-	

Рисунок 1.3 - Вигляд робочого середовища web-системи LP-System

Для більш детального порівняння web-орієнтованих систем було створено порівняльну таблицю (табл 1.1) для того щоб визначити головні моменти для реалізації власної системи. Серед основних недоліків слід відзначити наступні:

1. Zoho:

Деякі користувачі відзначають, що налаштування Zoho може бути складним завданням, особливо для тих, хто не має досвіду в роботі з аналогічними системами;

Оскільки Zoho - хмарна система, вона потребує стабільного інтернет-з'єднання, щоб користувачі могли безперервно використовувати систему.

2. Creatio:

Creatio відомий своєю високою вартістю, що може бути значним обмеженням для менших компаній або стартапів;

Незважаючи на широкі можливості інтеграцій, деякі користувачі зазначають, що процес інтеграції може бути складним і вимагати додаткових зусиль.

3. LP-System:

LP-System може бути менше підходящим для великих підприємств, оскільки воно зорієнтоване на основні функції та можливості;

Оскільки LP-System в основному локальна система, це може ускладнити доступ до даних та роботу з системою в режимі онлайн для деяких користувачів.

Таблиця 1.1 – Аналіз сервісів-аналогів

Критерій	Zoho	Creatio	LP-System	IT Sales
Зручна навігація системи	+	+	+	+
Розширена функціональність	+	-	-	+
Гнучка система	+	+	+	+
Наявність системи автоматизації обчислювальних процесів	+	+	+	+
Наявність інтегрованих сервісів	-	-	-	-

Продовження таблиці 1.1 – Аналіз сервісів-аналогів

Наявність системи генерації чеків після замовлення	-	-	-	+
Наявність графічних елементів	+	+	+	+
Наявність системи експорту БД	-	-	-	+

Джерело: побудовано автором

При розробці проекту необхідно враховувати вище перераховані результати аналізу, та розробити додаток, який б усував явні недоліки.

Функціональні можливості web-орієнтованої системи наступні:

- створення звітів у графічному вигляді по виконаним замовленням;
- генерація файлу бази даних під адаптацію з іншими СУБД;
- додавання послуг для клієнтів ІТ-компаній ;
- оформлення замовлень для клієнтів .

1.3 Аналіз технологій та засобів реалізації web-системи

Web-сервіс - це стандартизований засіб зв'язку між сервером і клієнтом в Інтернеті. Будучи програмним модулем, він виконує певні завдання. У хмарних обчисленнях веб-сервіси використовуються для надання різних функціональних можливостей за запитом клієнта [14].

Серед web-сервісів можна відзначити наступні типи систем:

1. SOAP (Simple Object Access Protocol)

SOAP відомий для транспортування XML-даних у вигляді SOAP-повідомлень. Це незалежний транспортний протокол. XML-документ слідує певному структурному шаблону. Найцікавіша частина надсилання SOAP-повідомлень і веб-сервісів полягає в тому, що все відбувається через HTTP (стандартний веб-протокол) [15].

Елементи SOAP-message:

- `envelope` - кореневий елемент XML-документа. Далі він поділяється на дві частини:
- `header` - містить інформацію про маршрутизацію, тобто місце призначення клієнта для отримання документа
- `body` - містить власне повідомлення

На рисунку 1.4 представлено простий приклад SOAP-зв'язку.

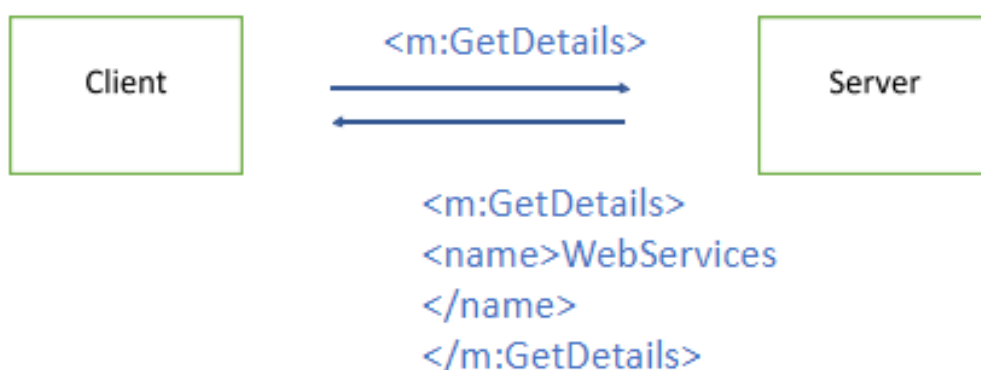


Рисунок 1.4 – приклад SOAP-зв'язку

2. WSDL (Web Services Description Language)

Перш ніж звертатися до веб-сервісу, необхідно дізнатися про його існування. Якщо сервіс не знайдено, його не можна використовувати. Крім того, клієнт повинен знати, що робить сервіс для виклику відповідного веб-сервісу [16].

Документ WSDL описує послуги, пропоновані веб-сервісом. Це дозволяє клієнту знайти сервіс і викликати його потрібним методом.

Нижче наведені аспекти щодо WSDL:

- `<message>` у визначенні WSDL, цей параметр визначає різні елементи даних для кожної операції веб-сервісу. У прикладі, показаному на рисунку вище, між додатками відбувається обмін двома повідомленнями, тобто операціями "DetailsRequest" і "DetailsResponse". DetailsRequest містить елемент DetailsID, який є рядковим типом. Аналогічно, DetailsResponse містить елемент "DetailsName" рядкового типу;
- `<portType>` цей елемент визначає операцію, яка може бути виконана веб-сервісом, в даному випадку, Details. Він приймає вхідне та вихідне повідомлення;
- `<binding>` в цьому елементі міститься протокол, який буде використовуватися. Інші деталі, такі як кодування повідомлення та простір імен, можуть бути оголошені в цьому елементі.

На рисунку 1.5 представлено ілюстрацію роботи документу WSDL.

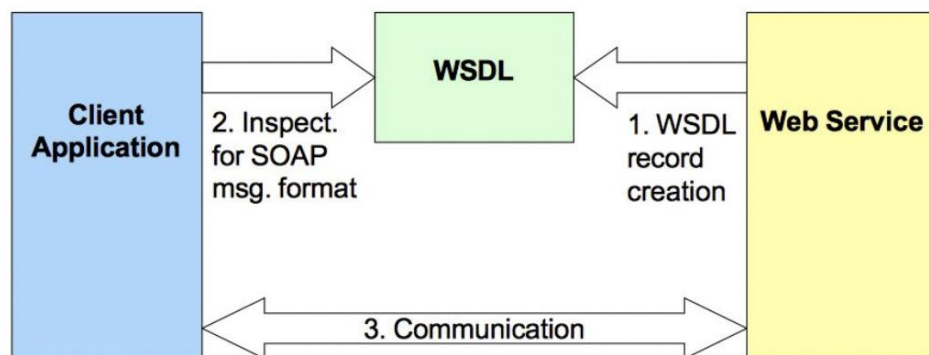


Рисунок 1.5 – ілюстрація роботи документу WSDL

Для розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній необхідно використовувати фреймворк Yii2 через наступні переваги:

1. Скорочення часу розробки:

Yii пропонує досить багато інструментів для автоматизації багатьох повторюваних завдань у проектах. Це дозволяє розробникам зосередити свій час і увагу на основних бізнес-вимогах і бізнес-логіці. Один з таких

інструментів називається "Gii". Gii допомагає в автоматичному створенні шаблонів коду для моделей, контролерів, форм, модулів та розширень. Gii повністю конфігурується, пропонуючи розробникам повну гнучкість, необхідну для створення масштабованого та кастомізованого коду.

2. Безпека:

Yii пропонує ряд механізмів безпеки, таких як XSS, міжсайтовий скриптинг, а також запобігання CSRF (підробка міжсайтових запитів). Ось як Yii справляється з трьома найпоширенішими проблемами безпеки.

Коли веб-додаток збирає дані від користувачів, наприклад, на форумах, існує ймовірність вбудовування шкідливого коду в повідомлення. Yii пропонує дуже корисний компонент, відомий як CHtmlPurifier, який видаляє шкідливі дані і гарантує, що відфільтрований вміст відповідає стандартам.

За замовчуванням Yii також пропонує перевірку CSRF (Cross Site Request Forgery - підробка міжсайтових запитів). Запобігання CSRF виникає, коли шкідливий веб-сайт змушує браузер користувача виконати небажану дію на довіреному сайті. Yii реалізує схему запобігання CSRF, яка допомагає протистояти атакам на основі POST.

Щоб уникнути Cookie-атак, Yii пропонує техніку перевірки файлів cookie, яка запобігає їх зміні. По суті, вона перевіряє HMAC (код автентифікації повідомлень на основі хешування) на наявність значень cookie, якщо вони увімкнені.

3. Кешування:

Кешування - це дуже ефективний спосіб підвищити продуктивність веб-системи. Воно допомагає зменшити час відгуку та покращує швидкість і продуктивність веб-системи. Yii дозволяє легко інтегрувати компонент кешування.

Yii надає різні компоненти кешування, такі як:

- sMemCache - memcache використовує вільну оперативну пам'ять на багатьох серверах, щоб діяти як кеш-пам'ять для часто використовуваної інформації;

- `sDummyCache` - містить фіктивний кеш, який взагалі не виконує кешування. Цей компонент використовується для спрощення коду, який повинен перевіряти наявність кешу;

- `sFileCache` - Використовує файли для зберігання кешованих даних.

4. Легка конфігурація:

Встановити Yii надзвичайно легко і просто. Базову структуру можна завантажити з сайту `Yii Framework` або завантажити через композитор.

Після завантаження структури та розпакування файлів розробникам потрібно просто з'єднати файли з базою даних, потім вказати конфігурацію бази даних, яка міститься в конфігураційному файлі, і все готово до запуску [17].

Також для організації роботи бази даних було обрано `MySQL` - безкоштовну базу даних з відкритим вихідним кодом. За допомогою `MySQL` ефективно керувати своїми даними за допомогою цієї програми, оскільки це дуже потужне, надійне та стабільне рішення [18].

Для створення продукту та перегляду результату було також використано наступні програмні продукти:

- `Visual Studio Code`;
- `Firefox`, `Google Chrome`, `Opera` для перевірки адаптації web-сервісу;
- `OpenServer` для запуску локального серверу;

Для аналізу технологій було взято інфографіку з спільноти програмістів `dou.ua`. У статті «Рейтинг мов програмування» було детально описано яким чином розробники використовують мови програмування [19].

Для прикладу на рисунку 1.4 представлено графік популярності кожної мови програмування для комерційного використання. В 2023 найпопулярнішою мовою залишається `JavaScript` — 19% розробників пишуть нею комерційні проекти. Далі — `Java` (14%) і `Python` (13%). Остання вперше серед найпопулярніших. З мінімальним відривом за `Python` йдуть `C#` і `TypeScript`.

Якою мовою пишете для роботи зараз

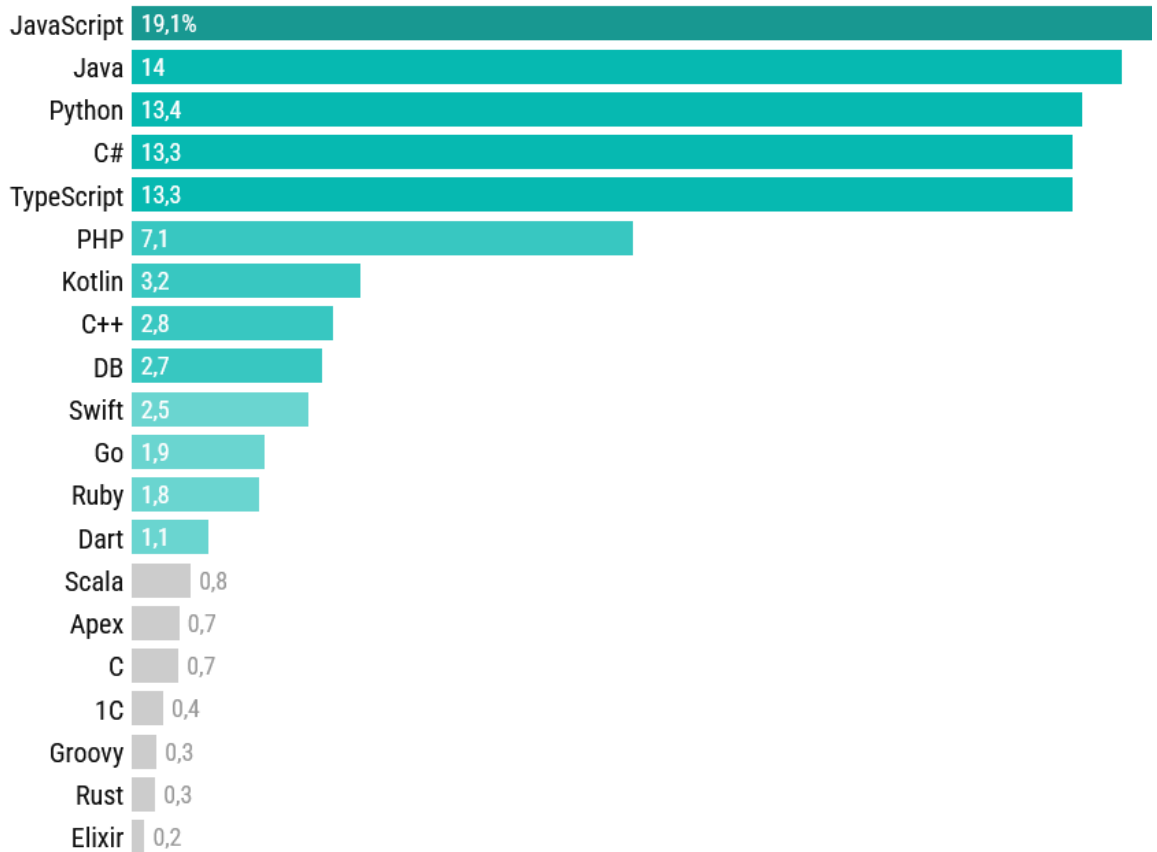


Рисунок 1.4 – Графік щодо популярності мов програмування від dou.ua

Даний графік також надає додаткову інформацію про кожну мову програмування з точки зору спільнот. Чим більш популярна мова програмування тим більше можливостей відкриваються перед розробником в плані використання для власних проектів.

Для розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній було обрано наступні технології:

- React.js;
- PHP (Yii2);
- Python.

React.js

ReactJS має невеликий розмір і створює бібліотеку JavaScript багаторазовими компонентами інтерфейсу користувача для прискорення розробки інтерфейсу користувача. Всі інтерфейсні веб-додатки та мобільні додатки, які створюються, можуть керувати своїм рівнем представлення за допомогою React JS.

Швидкість роботи DOM (Document Object Model) в інтерфейсі прикладного програмування (API) є ще однією перевагою ReactJS. Віртуальний DOM, який функціонує подібно до "дерева DOM", використовується фреймворком React для створення логічної структури для всіх сторінок, що використовуються в додатку [20].

Віртуальний DOM шукає найефективніший підхід до оновлення DOM браузера, використовуючи ці воронки та швидші запити. Ці та багато інших переваг є причинами, чому більшість розробників використовують React JS для веб-додатків.

PHP (Yii2)

PHP-фреймворк Yii2 був розроблений з метою полегшити розробникам виконання декількох завдань, таких як автентифікація, сесії, кешування та маршрутизація. Це означає, що проблеми розробки веб-додатків вирішуються за допомогою PHP-фреймворку Yii2.

Деякі компанії, що займаються веб-розробкою, надають перевагу PHP для створення фреймворку сайту. Однак Yii2 є найкращим вибором. Популярність розробки на Yii2 зростає завдяки детальному синтаксису, гнучкості та спрощенню ряду завдань.

Вибір фреймворку є складним завданням, оскільки в галузі існує кілька варіантів. У сучасному швидкоплинному середовищі треба розуміти, який фреймворк працює відразу після інсталяції на сервер. Yii2 став

найпопулярнішим фреймворком і вибором за замовчуванням для розробки складних веб-орієнтованих систем.

1. Безпека веб-ресурсів

Безпека є найбільшою проблемою, через яку більшість компаній обирають Yii2 як фреймворк для розробки своїх веб-систем. Фреймворки Yii2 мають розширені функції безпеки, за допомогою яких легко налаштувати веб-сайт. Це необхідно для підвищення безпеки та захисту сайту від хакерів.

Yii2 використовує алгоритм хешування Bcrypt, який ніколи не зберігає пароль в базі даних. За допомогою Yii2 можна вирішити проблеми міжсайтового скриптингу, SQL ін'єкцій та підміни запитів. Yii2 захищає код розробника і виправляє уразливості безпеки та інші технічні проблеми перед розгортанням веб-сайту.

2. Висока продуктивність при високому навантаженні на сервер

Коли ваш бізнес зростає, трафік на ваш сайт також зростає. Веб-сайт, розроблений на Yii2, може обробляти запити набагато швидше, ніж інші фреймворки. Yii2 використовує унікальну систему повідомлень, за допомогою якої можна відкладати певні завдання сайту, наприклад, відправку пошти, на певний час. Розробники використовують її для підтримки швидкості роботи сайту, зменшуючи витрати на хостинг.

3. Гнучкість

За допомогою фреймворку Yii2 можна розробити сайт електронної комерції або простий, але професійний B2B сайт. Він має великі попередньо встановлені бібліотеки, які дозволяють створювати та підтримувати різні розширені функції.

4. Легке обслуговування та підтримка

Сайт, створений на фреймворку Yii2, легше підтримувати, ніж інші. За допомогою Yii2 можна легко впроваджувати інші функції на розробленому сайті вже через рік. Новий розробник може легко почати свою роботу з того місця, де зупинився попередній розробник. Yii2 простіше, ніж інші

фреймворки, завдяки вбудованим функціям, архітектурі MVC та своєчасній підтримці спільноти. Це дозволяє модернізувати сайт у будь-який момент.

5. Вбудована система аутентифікації

Yii2 – простий в використанні фреймворк, тому що він має систему аутентифікації користувачів plug & play. Це дає користувачам доступ до веб-сайту в простому процесі. Yii2 має вбудований пакет аутентифікації, який надає контрольний доступ до захищених ресурсів. Yii2 пропонує спрощену систему інтеграції з поштою, яка дозволяє створювати поштові сервери. Yii2 підтримує відправку повідомлень по різних каналах і пропонує хмарні поштові сервіси, такі як Amazon SES, SMTP і багато інших.

Python

Python - це платформа з відкритим вихідним кодом. Це означає, що дана мова програмування може бути модифікована розробником відповідно до його конкретних потреб та вимог. У Python легко створювати блоки та бібліотеки, і, що більш важливо, як мову, її можна масштабувати та модернізувати, щоб відповідати технологічним змінам, що відбуваються в процесі розвитку.

Переваги Python для розробки web-систем наступні:

Легкість у вивченні. Що стосується простоти вивчення та використання, то Python дуже зручний і простий у вивченні. Крім того, він також простий у використанні і підтримує створення потужних та адаптивних веб-сайтів.

Підтримує багатозадачність. Важливою перевагою Python для стартапів та великих компаній (малих, середніх та великих) є підтримка багатозадачності. Багато організацій продовжують обирати Python для веб-розробки, оскільки завдяки своїй багатозадачності вона може одночасно працювати над складними питаннями, такими як безпека даних, що робить її однією з найнадійніших мов у світі.

Створення більшості функцій з меншою кількістю коду. Багатовимірний веб-сайт може мати велике значення у створенні такого важливого першого враження на вашого клієнта. Саме тут веб-розробка на Python виграє, оскільки вона вважається надійною та ефективною і може включати більше можливостей завдяки додатковим функціям. Ви можете швидко зробити щось, не витрачаючи надто багато часу і практично з мінімальною кількістю необхідного коду.

Має вбудовані бібліотеки, які можуть підтримувати використання штучного інтелекту у веб-розробці. Так само, як і Python, штучний інтелект та машинна мова розвиваються. Вбудовані бібліотеки, які поставляються разом з Python, підтримують програмування на основі штучного інтелекту та машинної мови, що має значний вплив на продуктивність кінцевого веб-додатку, оскільки ці бібліотеки містять Інтернет.

Легко створювати прототипи. Python підтримує розробку прототипів, які розглядаються багатьма клієнтами як перше уявлення про кінцевий продукт, що створюється. Легкість, з якою працює Python, є вагомим аргументом на користь того, щоб стати першим вибором для веб-розробки. Швидші прототипи можуть заощадити час, зменшити витрати і допомогти отримати задоволеного клієнта, і Python досить добре працює в цьому напрямку.

Проаналізувавши переваги кожної з мов програмування було створено таблицю порівняння (табл. 1.2) з врахуванням переваг та недоліків для більш детального вибору технології для реалізації поставленої задачі.

Таблиця 1.2 – Таблиця порівняння мов програмування

Критерій	React.js	PHP (Yii2)	Python
Мова програмування	JavaScript	PHP	Python
Фреймворк	React	Yii2	Django
Тип	Бібліотека для фронтенду	Фреймворк для повного стеку	Фреймворк для бекенду

Критерій	React.js	PHP (Yii2)	Python
Крива навчання	Середня	Середня до високої	Середня
Підтримка спільноти	Велика	Велика	Велика
Продуктивність	Висока (для фронтенду)	Добра (залежить від реалізації)	Добра
Масштабованість	Відмінна	Добра	Добра
Гнучкість	Висока	Середня	Висока
Швидкість розробки	Швидка	Середня	Середня
Екосистема	Багата	Міцна	Обширна
Популярні випадки використання	Односторінкові додатки (SPA)	Веб-додатки, API	Веб-додатки, наука про дані
Підтримка тестування	Добра	Добра	Добра
Безпека	Вимагає уважного оброблення стану	Безпечний, але залежить від практик	Безпечний, але залежить від практик
Тенденції спільноти	Популярний для розробки сучасних інтерфейсів	Встановлений для веб-розробки	Широко використовується в різних галузях

Джерело: побудовано автором

Для розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній необхідно використовувати фреймворк Yii2 через наступні переваги:

1. Скорочення часу розробки:

Yii пропонує досить багато інструментів для автоматизації багатьох повторюваних завдань у проектах. Це дозволяє розробникам зосередити свій час і увагу на основних бізнес-вимогах і бізнес-логіці. Один з таких інструментів називається "Gii". Gii допомагає в автоматичному створенні шаблонів коду для моделей, контролерів, форм, модулів та розширень. Gii повністю конфігурується, пропонуючи розробникам повну гнучкість, необхідну для створення масштабованого та кастомізованого коду.

2. Безпека:

Yii пропонує ряд механізмів безпеки, таких як XSS, міжсайтовий скриптинг, а також запобігання CSRF (підробка міжсайтових запитів). Ось як Yii справляється з трьома найпоширенішими проблемами безпеки

Коли веб-додаток збирає дані від користувачів, наприклад, на форумах, існує ймовірність вбудовування шкідливого коду в повідомлення. Yii пропонує дуже корисний компонент, відомий як CHtmlPurifier, який видаляє шкідливі дані і гарантує, що відфільтрований вміст відповідає стандартам.

За замовчуванням Yii також пропонує перевірку CSRF (Cross Site Request Forgery - підробка міжсайтових запитів). Запобігання CSRF виникає, коли шкідливий веб-сайт змушує браузер користувача виконати небажану дію на довіреному сайті. Yii реалізує схему запобігання CSRF, яка допомагає протистояти атакам на основі POST.

Щоб уникнути Cookie-атак, Yii пропонує техніку перевірки файлів cookie, яка запобігає їх зміні. По суті, вона перевіряє HMAC (код автентифікації повідомлень на основі хешування) на наявність значень cookie, якщо вони увімкнені.

3. Кешування:

Кешування - це дуже ефективний спосіб підвищити продуктивність веб-системи. Воно допомагає зменшити час відгуку та покращує швидкість і продуктивність веб-системи. Yii дозволяє легко інтегрувати компонент кешування.

Yii надає різні компоненти кешування, такі як:

- `sMemCache` - `memcache` використовує вільну оперативну пам'ять на багатьох серверах, щоб діяти як кеш-пам'ять для часто використовуваної інформації;

- `sDummyCache` - містить фіктивний кеш, який взагалі не виконує кешування. Цей компонент використовується для спрощення коду, який повинен перевіряти наявність кешу;

- `sFileCache` - Використовує файли для зберігання кешованих даних.

4. Легка конфігурація:

Встановити Yii надзвичайно легко і просто. Базову структуру можна завантажити з сайту Yii Framework або завантажити через композитор.

Після завантаження структури та розпакування файлів розробникам потрібно просто з'єднати файли з базою даних, потім вказати конфігурацію бази даних, яка міститься в конфігураційному файлі, і все готово до запуску [14].

Також для організації роботи бази даних було обрано MySQL - безкоштовну базу даних з відкритим вихідним кодом. За допомогою MySQL ефективно керувати своїми даними за допомогою цієї програми, оскільки це дуже потужне, надійне та стабільне рішення.

1.4 Постановка задачі

Аналіз надає можливість визначити можливості, які повинна надавати система щоб конкурувати з сервісами-аналогами. Дана web-система повинна надавати можливість моніторингу бізнес-процесів та створювати звітність на основі існуючих даних. Також повинно передбачити можливість експорту даних системи шляхом генерації датафайлу. Дана web-система повинна бути реалізована як web-додаток, який надаватиме можливість користуватися з будь якого доступного сучасного пристрою.

Web-система повинна надавати наступні можливості:

- створення звітів у графічному вигляді по виконаним замовленням;
- генерація файлу бази даних під адаптацію з іншими СУБД;
- додавання послуг для клієнтів ІТ-компаній ;
- оформлення замовлень для клієнтів .

Web-система повинна бути реалізована за допомогою клієнт-серверної архітектури та містити адміністративну частину доступ до якої отримується за допомогою логіну та паролю. Також система повинна мати інтуїтивно

зрозумілий інтерфейс, за допомогою якого користувач зможе орієнтуватися по навігації сервісу та отримувати дані з серверу з швидкою обробкою даних.

Для виконання поставленої мети необхідно вирішити наступні завдання:

- виконати аналіз систем-аналогів та визначити переваги та недоліки;

Дане завдання ставить мету визначити сучасні технології, які відносяться до даної задачі та проаналізувати шляхи покращення власної системи

- визначити функціональні та нефункціональні вимоги web-системи;
- реалізувати клієнт-серверну архітектуру;

Дане завдання в результаті вирішення має надати користувачу архітектуру, за допомогою якої користувач отримуватиме дані з мінімальною затримкою обробки, а також отримає систему, яка матиме процеси автоматизації процесів, які стосуються графічної подачі даних на основі замовлень.

- реалізувати структуру, функціонал та дизайн web-системи;
- створити логічну та фізичну модель бази даних, визначити сутності та зв'язки а також визначити систему управління базою даних та інтегрувати створену базу даних з web-сервісом;

В даному завданні необхідно визначити СУБД за допомогою якого буде створена реляційна база даних та система хешування особистих даних на основі обраної СУБД.

2 МОДЕЛЮВАННЯ РОБОТИ WEB-ДОДАТКУ

2.1 Структурно-функціональне моделювання

У теорії та практиці управління проектами стратегічне планування має вирішальне значення для успішної організації, розробки та виконання будь-якого проекту. Життєздатна стратегія проекту відіграє важливу роль для розуміння проекту як об'єкта з конкретним технічним і бізнес-змістом, спрямованого на досягнення певних цілей і реалізованого за допомогою конкретних засобів [21]; комунікації проекту із зацікавленими сторонами та прийняття рішень з техніко-економічних питань, що мають значний соціальний та екологічний вплив.

Крім того, оскільки система підтримки роботи відділу продажів ІТ компаній є складною структурою, стратегічне планування відповідає вимогам більш глибокого вивчення бізнес-функцій, так званих процесів, за допомогою яких може бути виконаний цей тип проектів. Методом, який найкраще відповідає вимогам стратегічного планування проектів, є IDEF0, техніка моделювання процесів, а також промисловий стандарт, що має широке літературне використання в аналізі та розробці проектів у сферах виробництва, аерокосмічної галузі, бізнесу, науки і техніки інформаційних систем [22].

IDEF0 - це техніка структурування та моделювання інженерних робочих процесів, що поєднує графічні та текстові об'єкти [23]. Синтаксис IDEF0 представляє ієрархію моделей процесів у структурований і систематизований спосіб, використовуючи:

- блок-схеми для визначення процесів або видів діяльності;
- стрілки для представлення потоків даних, взаємозв'язків і циклів зворотного зв'язку між процесами;
- сутності, необхідні кожному окремому процесу для його активації, що дозволяє інтегровану композицію процесів у моделі: Входи і виходи (дані або об'єкти), механізми (наприклад, матеріали, ресурси, знання та/або інші засоби)

і засоби контролю (рішення, законодавчі або регуляторні обмеження, контракти, часові плани).

На рисунку 2.1 представлена контекстна діаграма web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній.

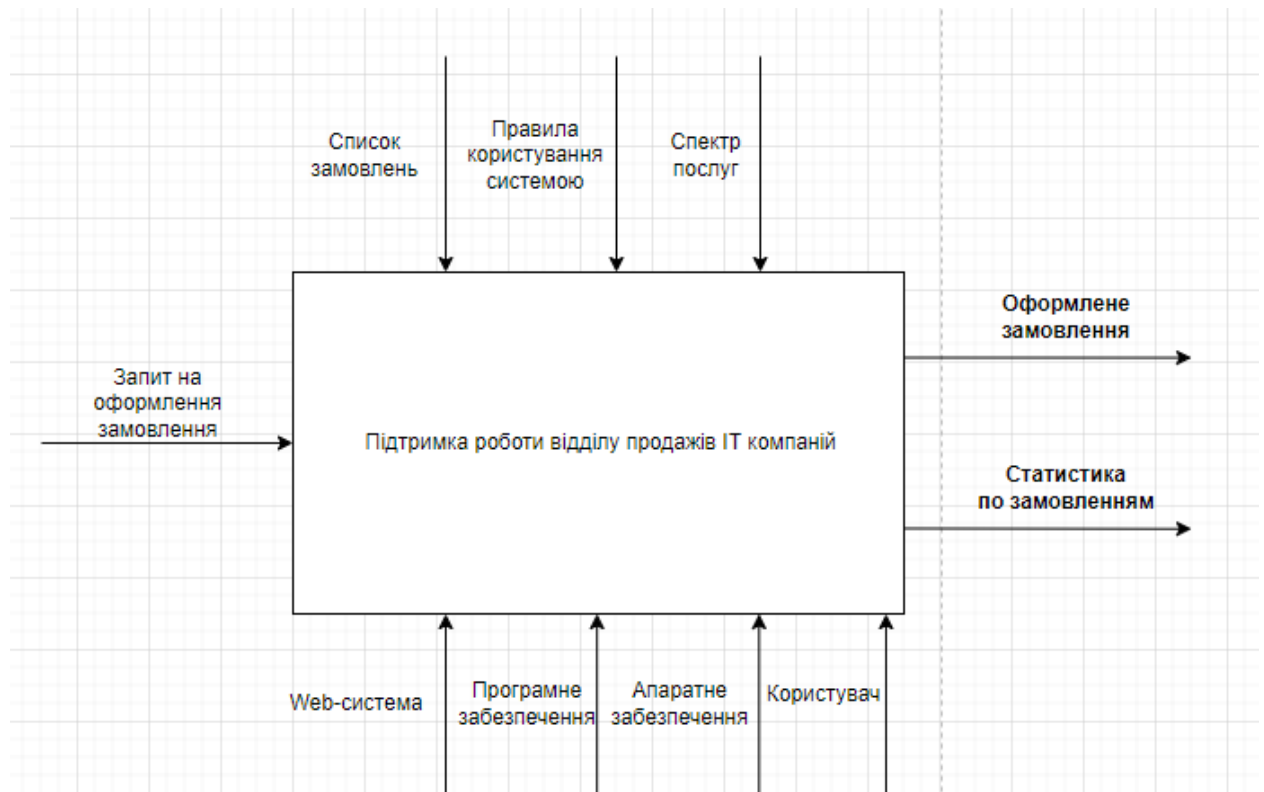


Рисунок 2.1 - Контекстна діаграма web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній

Для контекстної діаграми було визначено наступні дані:

- вхідні дані: запит на оформлення замовлення;
- вихідні дані: результат оформленого замовлення;
- управління: список транзакцій, правила користування, спектр послуг;
- механізми: програмне забезпечення, web-сервіс, технічне забезпечення.

Для більш детального розуміння контекстної діаграми було виконано декомпозицію для деталізації внутрішніх процесів (рис. 2.2).

Декомпозиція представлена наступними підпроцесами:

- Авторизація адміністратора;

- Збір даних про замовлення клієнтів;
- Оформлення замовлення;
- Створення графічних звітів (діаграми) на основі вхідних даних про замовлення.

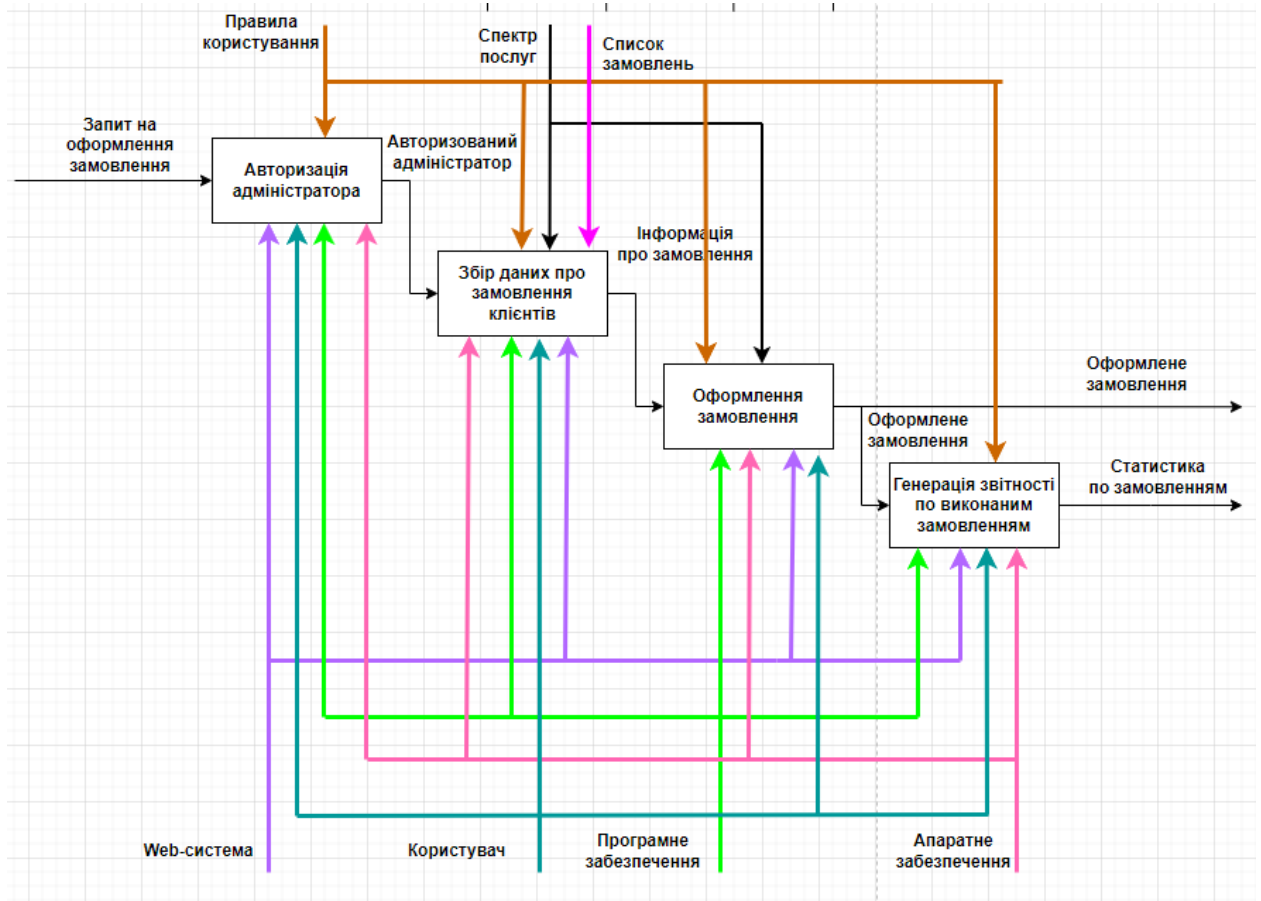


Рисунок 2.2 – Діаграма декомпозиції оформлення транзакції

2.2 Моделювання варіантів використання

Використовуючи уніфіковану мову моделювання (UML), діаграма варіантів використання допомагає зрозуміти, як користувач може взаємодіяти з системою, яку розробили. Зрештою, це має допомогти визначити та впорядкувати вимоги.

Діаграми варіантів використання UML не мають на меті надати глибокий погляд на кожен елемент розробленої системи. Натомість, вони

представляють загальний огляд того, як пов'язані між собою варіанти використання, актори та система. [24]

На рисунку 2.3 представлено діаграму варіантів використання web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній. Акторами на діаграмі варіантів використання є користувач, який може бути в ролі адміністратора або менеджера з продажів, БД та сервер.

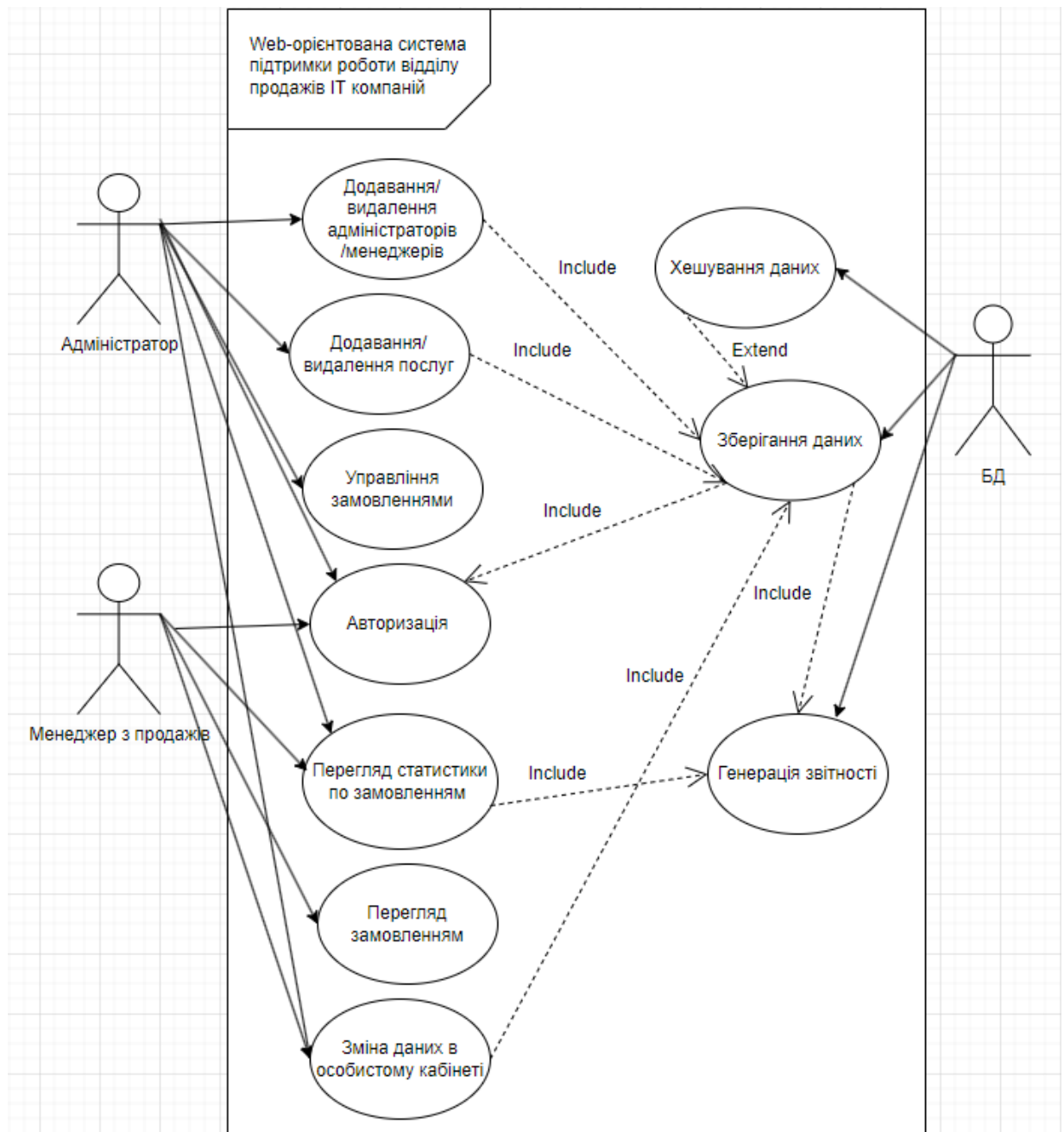


Рисунок 2.3 – Діаграма варіантів використання web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній

2.3 Проектування бази даних

MySQL – це безкоштовна база даних з відкритим вихідним кодом, яка полегшує ефективне управління базами даних шляхом їх підключення до програмного забезпечення [18]. Це стабільне, надійне та потужне рішення з розширеними можливостями:

1. Захист даних

Це одна з найпопулярніших систем управління базами даних з точки зору безпеки та надійності. Вона використовується в багатьох веб-системах, включаючи Twitter, Facebook, Joomla, Drupal і WordPress. Для обробки транзакцій підтримка та безпека даних може бути корисною для кожної організації, особливо в галузі електронної комерції [25].

2. Масштабованість на вимогу

Неймовірна масштабованість цієї платформи може допомогти в управлінні вбудованими додатками. Це справедливо навіть для великих сховищ з великими обсягами даних..

Перевага цієї платформи з відкритим вихідним кодом полягає в тому, що вона дозволяє повністю змінити її під власні потреби, що дуже корисно, якщо є веб-система, якою потрібно керувати.

3. Висока ефективність

MySQL включає унікальний механізм зберігання даних, який полегшує адміністрування системи. Крім того, можна налаштувати сервер бази даних MySQL для досягнення максимальної продуктивності. Іншими словами, навіть якщо в систему вбудована високошвидкісна система обробки даних або веб-сервіс, який отримує мільйони відвідувань щодня, то можна отримати вигоду від цієї технології [25].

4. Безперебійна робота 24/7

Система MySQL унікальна тим, що забезпечує безперервний час роботи. Крім того, вона надає безліч опцій, таких як налаштування реплікації між підлеглим і головним серверами та унікальні кластерні сервери.

5. Видатна підтримка транзакцій

MySQL знаходиться на вершині списку швидких транзакційних рішень для баз даних, доступних сьогодні. Вона має кілька особливостей, таких як послідовна та автоматизована підтримка транзакцій. Крім того, це один з найкращих варіантів для забезпечення повної цілісності даних.

6. Відмінний контроль робочого процесу

Не потрібно виділяти багато часу на налаштування та установку даної технології, оскільки завантаження та встановлення займає менше 30 хвилин. Також будь який користувач може скористатися перевагами цього рішення незалежно від платформи. Його можна використовувати для автоматизації широкого спектру процесів, включаючи управління базами даних, дизайн даних і розширення простору [25].

На рисунку 2.4 представлено схему реалізованої бази даних web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній яка була реалізована в додатку phpMyAdmin.

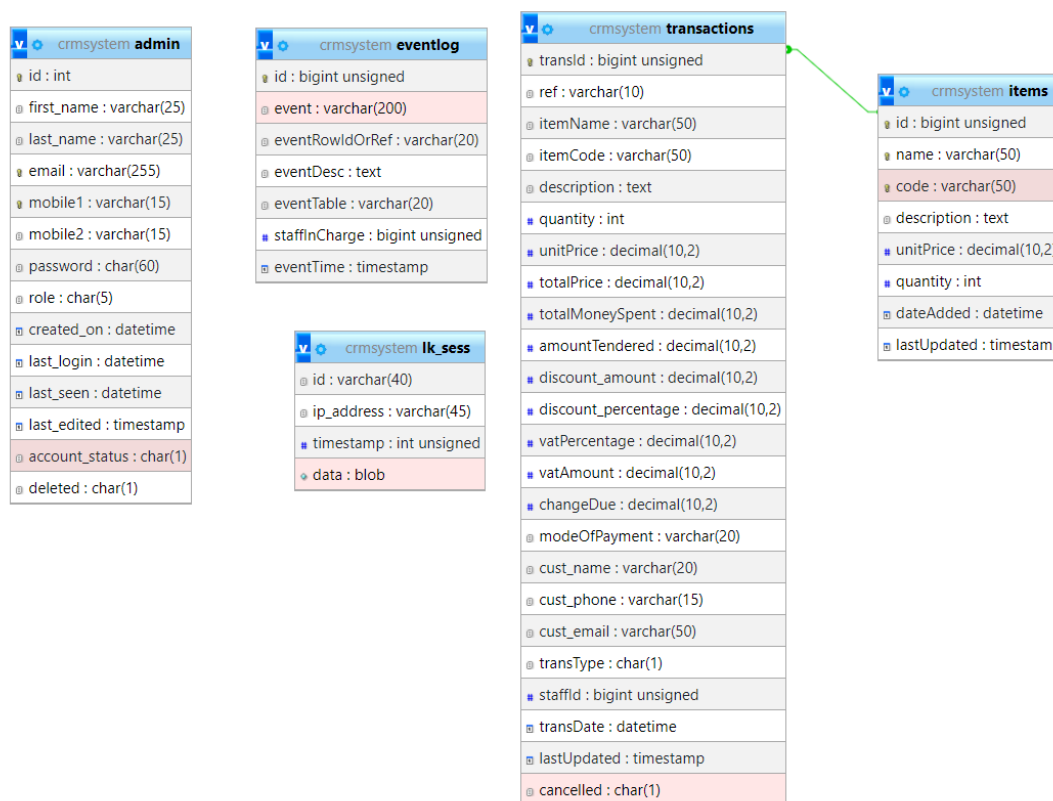


Рисунок 2.4 – Схема реалізованої бази даних web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Ініціалізація проекту та інтеграція БД з web-системою

Для ініціалізації проекту перш за все необхідно налаштувати робоче середовище. В якості локального серверу буде використовуватися OpenServer. За допомогою пакетного менеджера для керування залежностями в PHP додатку Composer необхідно виконати команду `composer create-project --prefer-dist yiisoft/yii2-app-basic basic`. Ця команда встановлює останню стабільну версію фреймворку в обрану директорію.

Після успішної установки додаток буде доступний за посиланням `http://localhost/basic/web/index.php`. На рисунку 3.1 представлено результат успішного встановлення фреймворку.

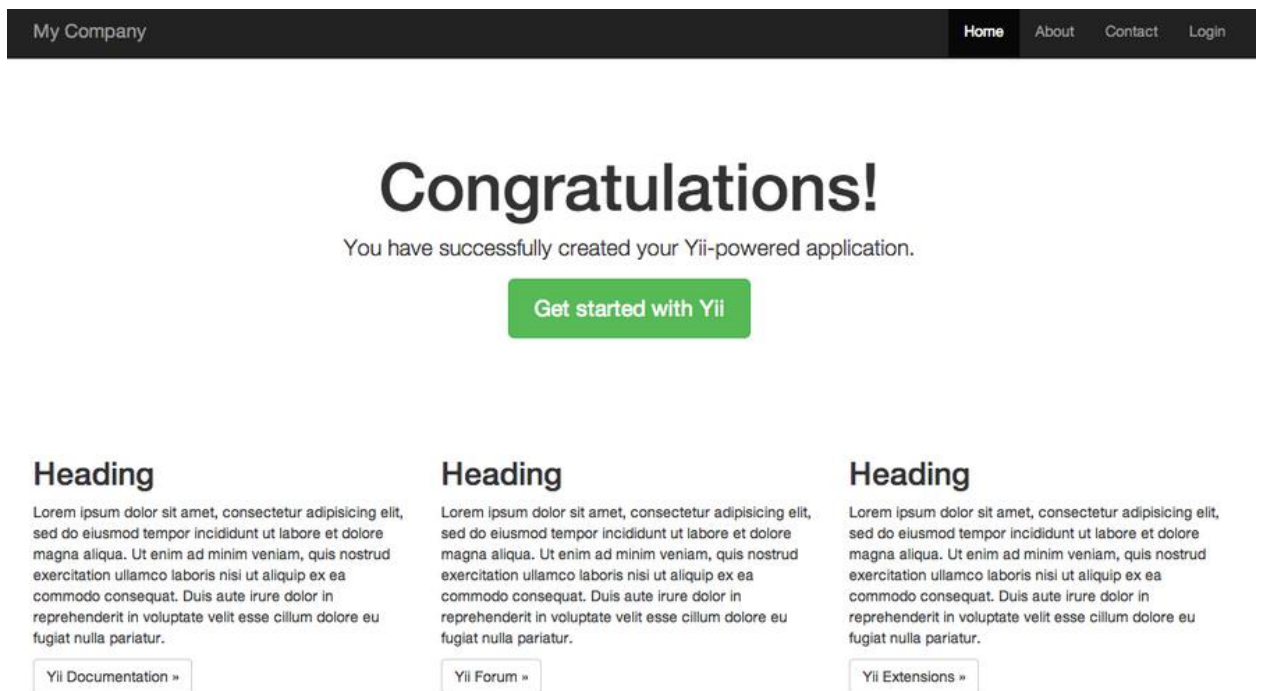


Рисунок 3.1 – Результат успішного встановлення фреймворку *Джерело:*
побудовано автором

Після цього необхідно за допомогою інструменту phpMyAdmin та мови MySQL створити базу даних web-орієнтованої системи. На рисунку 3.2 представлено вигляд бази даних itsales.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> admin	☆ 🗃️ 🔍 🗑️ 📄 🚫	5	InnoDB	latin1_swedish_ci	48.0 КиБ	-
<input type="checkbox"/> eventlog	☆ 🗃️ 🔍 🗑️ 📄 🚫	4	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> items	☆ 🗃️ 🔍 🗑️ 📄 🚫	2	InnoDB	latin1_swedish_ci	48.0 КиБ	-
<input type="checkbox"/> lk_sess	☆ 🗃️ 🔍 🗑️ 📄 🚫	0	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> transactions	☆ 🗃️ 🔍 🗑️ 📄 🚫	2	InnoDB	latin1_swedish_ci	16.0 КиБ	-
5 таблиц	Всего	13	InnoDB	utf8mb4_0900_ai_ci	144.0 КиБ	0 Байт

Рисунок 3.2 – Результат створення таблиці itsales *Джерело: побудовано автором*

Кожна таблиця містить унікальний ключ та атрибути для збереження інформації. На рисунках 3.3 – 3.7 представлено вигляд кожної таблиці бази даних itsales.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int			Нет	Нет		AUTO_INCREMENT	Ещё
2	first_name	varchar(25)	latin1_swedish_ci		Нет	Нет			Ещё
3	last_name	varchar(25)	latin1_swedish_ci		Нет	Нет			Ещё
4	email	varchar(255)	latin1_swedish_ci		Нет	Нет			Ещё
5	mobile1	varchar(15)	latin1_swedish_ci		Нет	Нет			Ещё
6	mobile2	varchar(15)	latin1_swedish_ci		Нет	Нет			Ещё
7	password	char(60)	latin1_swedish_ci		Нет	Нет			Ещё
8	role	char(5)	latin1_swedish_ci		Нет	Нет			Ещё
9	created_on	datetime			Нет	Нет			Ещё
10	last_login	datetime			Нет	Нет			Ещё
11	last_seen	datetime			Нет	Нет			Ещё
12	last_edited	timestamp		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	Ещё
13	account_status	char(1)	latin1_swedish_ci		Нет	1			Ещё
14	deleted	char(1)	latin1_swedish_ci		Нет	0			Ещё

Рисунок 3.3 – Структура таблиці «admin» Джерело: побудовано автором

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё
2	event	varchar(200)	latin1_swedish_ci		Нет	Нет			Ещё
3	eventRowIdOrRef	varchar(20)	latin1_swedish_ci		Да	NULL			Ещё
4	eventDesc	text	latin1_swedish_ci		Да	NULL			Ещё
5	eventTable	varchar(20)	latin1_swedish_ci		Да	NULL			Ещё
6	staffinCharge	bigint		UNSIGNED	Нет	Нет			Ещё
7	eventTime	timestamp		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	Ещё

Рисунок 3.4 – Структура таблиці «eventlog» Джерело: побудовано автором

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё
2	name	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё
3	code	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё
4	description	text	latin1_swedish_ci		Да	NULL			Ещё
5	unitPrice	decimal(10,2)			Нет	Нет			Ещё
6	quantity	int			Нет	Нет			Ещё
7	dateAdded	datetime			Нет	Нет			Ещё
8	lastUpdated	timestamp		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	Ещё

Рисунок 3.5 – Структура таблиці «items» Джерело: побудовано автором

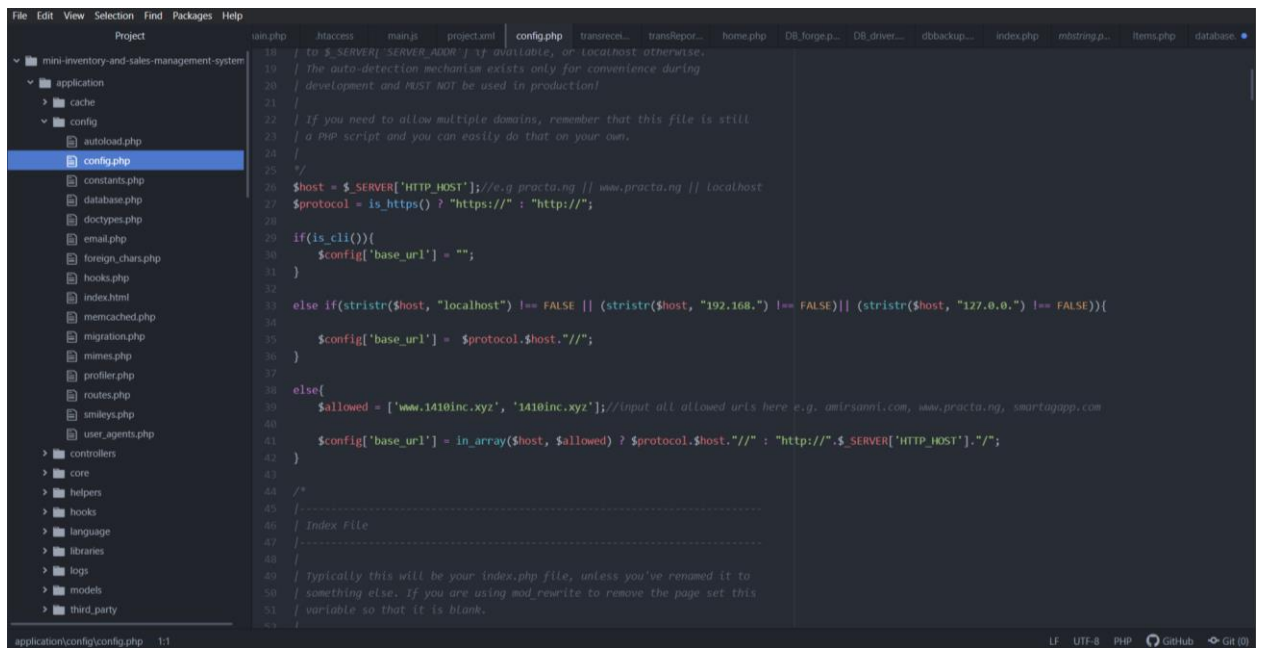
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	varchar(40)	latin1_swedish_ci		Нет	Нет			Ещё
2	ip_address	varchar(45)	latin1_swedish_ci		Нет	Нет			Ещё
3	timestamp	int		UNSIGNED	Нет	0			Ещё
4	data	blob			Нет	Нет			Ещё

Рисунок 3.6 – Структура таблиці «lk_less» Джерело: побудовано автором

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	transId	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё
2	ref	varchar(10)	latin1_swedish_ci		Нет	Нет			Ещё
3	itemName	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё
4	itemCode	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё
5	description	text	latin1_swedish_ci		Да	NULL			Ещё
6	quantity	int			Нет	Нет			Ещё
7	unitPrice	decimal(10,2)			Нет	Нет			Ещё
8	totalPrice	decimal(10,2)			Нет	Нет			Ещё
9	totalMoneySpent	decimal(10,2)			Нет	Нет			Ещё
10	amountTendered	decimal(10,2)			Нет	Нет			Ещё
11	discount_amount	decimal(10,2)			Нет	Нет			Ещё
12	discount_percentage	decimal(10,2)			Нет	Нет			Ещё
13	vatPercentage	decimal(10,2)			Нет	Нет			Ещё
14	vatAmount	decimal(10,2)			Нет	Нет			Ещё
15	changeDue	decimal(10,2)			Нет	Нет			Ещё
16	modeOfPayment	varchar(20)	latin1_swedish_ci		Нет	Нет			Ещё
17	cust_name	varchar(20)	latin1_swedish_ci		Да	NULL			Ещё

Рисунок 3.7 – Структура таблиці «transactions» Джерело: побудовано автором

Після того, як база даних була створена, необхідно підключити її до проекту. На рисунку 3.8 представлено вигляд файлу config.php. Повний лістинг коду представлено в додатку Б



```
18 | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
19 | The auto-detection mechanism exists only for convenience during
20 | development and MUST NOT be used in production!
21 |
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 |
25 |
26 | $host = $_SERVER['HTTP_HOST']; //e.g. practa.ng || www.practa.ng || localhost
27 | $protocol = is_https() ? "https://" : "http://";
28 |
29 | if(is_cli()){
30 |     $config['base_url'] = "";
31 | }
32 |
33 | else if(strpos($host, "localhost") !== FALSE || (strpos($host, "192.168.") !== FALSE) || (strpos($host, "127.0.0.") !== FALSE)){
34 |     $config['base_url'] = $protocol.$host."/";
35 | }
36 |
37 | else{
38 |     $allowed = ['www.1410inc.xyz', '1410inc.xyz']; //input all allowed uris here e.g. amirsami.com, www.practa.ng, smartajapp.com
39 |
40 |     $config['base_url'] = in_array($host, $allowed) ? $protocol.$host."/": "http://".$_SERVER['HTTP_HOST']."/";
41 | }
42 |
43 |
44 |
45 | -----
46 | Index File
47 | -----
48 |
49 | Typically this will be your index.php file, unless you've renamed it to
50 | something else. If you are using mod_rewrite to remove the page set this
51 | variable so that it is blank.
52 |
```

Рисунок 3.8 – Вигляд файлу config.php Джерело: побудовано автором

Після цього необхідно інтегрувати в проект фреймворк CodeIgniter для більш покращеного процесу створення компонентів та системи навігації (роутингу).

Для встановлення фреймворку необхідно виконати 4 етапи:

- 1) З офіційного сайту <https://codeigniter.com> необхідно завантажити та розархівувати проект в директорії створеного проекту.
- 2) Тепер необхідно завантажити папки та файли CodeIgniter на сервер.
- 3) Після цього необхідно відкрити файл config.php та встановити базову URL адресу. Також системою передбачено використання технології шифрування особистих даних, тому ключ шифрування також необхідно встановити.
- 4) Для налаштування бази даних необхідно відкрити файл database.php та встановити параметри підключення.

На рисунку 3.9 представлено вигляд файлу database.php

```
70 | The $query_builder variables lets you determine whether or not to load
71 | the query builder class.
72 | */
73 | $active_group = 'mysql';
74 | $query_builder = TRUE;
75 |
76 | //details for mysql
77 | $db['mysql'] = array(
78 |     'dsn' => 'mysql:host=localhost; dbname=crmitsales',
79 |     'hostname' => '',
80 |     'username' => 'root',
81 |     'password' => '',
82 |     'database' => '',
83 |     'dbdriver' => 'pdo',
84 |     'dbprefix' => '',
85 |     'pconnect' => FALSE,
86 |     'db_debug' => (ENVIRONMENT !== 'production'),
87 |     'cache_on' => FALSE,
88 |     'cachedir' => '',
89 |     'char_set' => 'utf8',
90 |     'dbcollat' => 'utf8_general_ci',
91 |     'swap_pre' => '',
92 |     'encrypt' => FALSE,
93 |     'compress' => FALSE,
94 |     'stricton' => FALSE,
95 |     'failover' => array(),
96 |     'save_queries' => TRUE
97 | );
98 |
99 | //details for sqlite
100 | $db['sqlite'] = array(
101 |     'dsn' => '',
102 |     'hostname' => ''
103 | );
```

Рисунок 3.9 – Вигляд файлу database.php Джерело: побудовано автором

Після того, як виконано налаштування БД в проекті, можна перейти до реалізації серверної частини

3.2 Реалізація серверної частини

На початку реалізації необхідно створити контролери та роутери, які надаватимуть кожній сторінці унікальну адресу. На рисунку 3.10 представлено створені контролери. Основна роль контролерів – обробка запитів від користувача та управління виконання дій, які пов’язані з конкретними областями функціоналу додатка. Також за допомогою механізму маршрутизації можна визначити, як HTTP запити повинні бути співставлені з іншими контролерами та діями. Для прикладу, на рисунку 3.11 представлено вигляд контролеру Administrator.php. Повний лістинг коду представлено в додатку В.

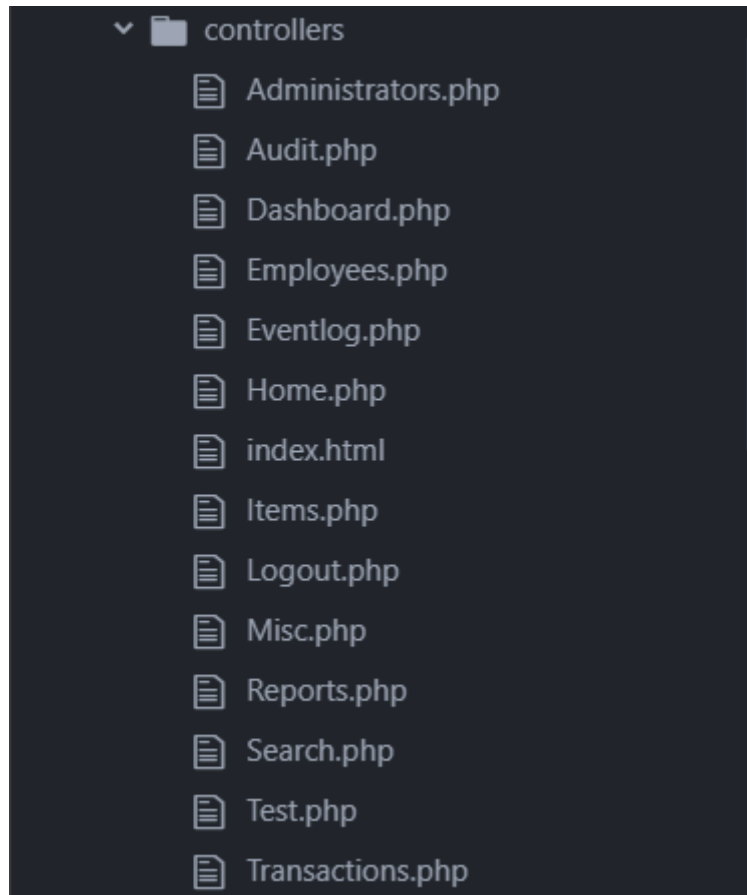


Рисунок 3.10 – Результат створених контролерів Джерело: побудовано автором

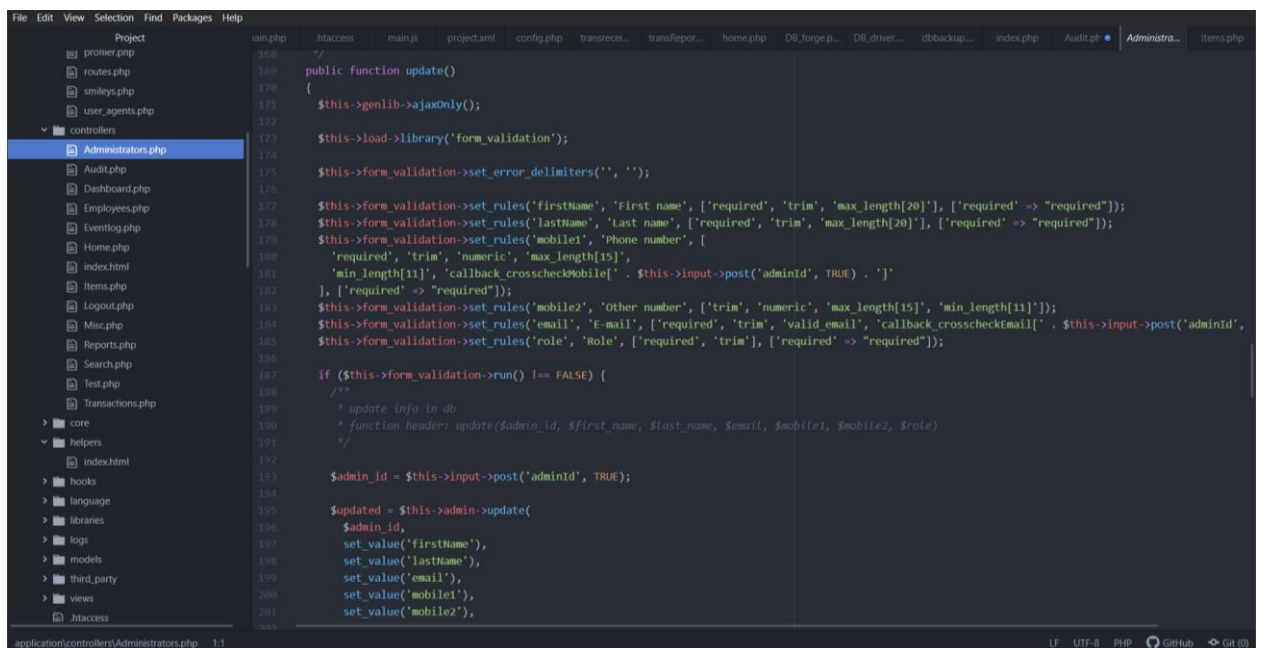
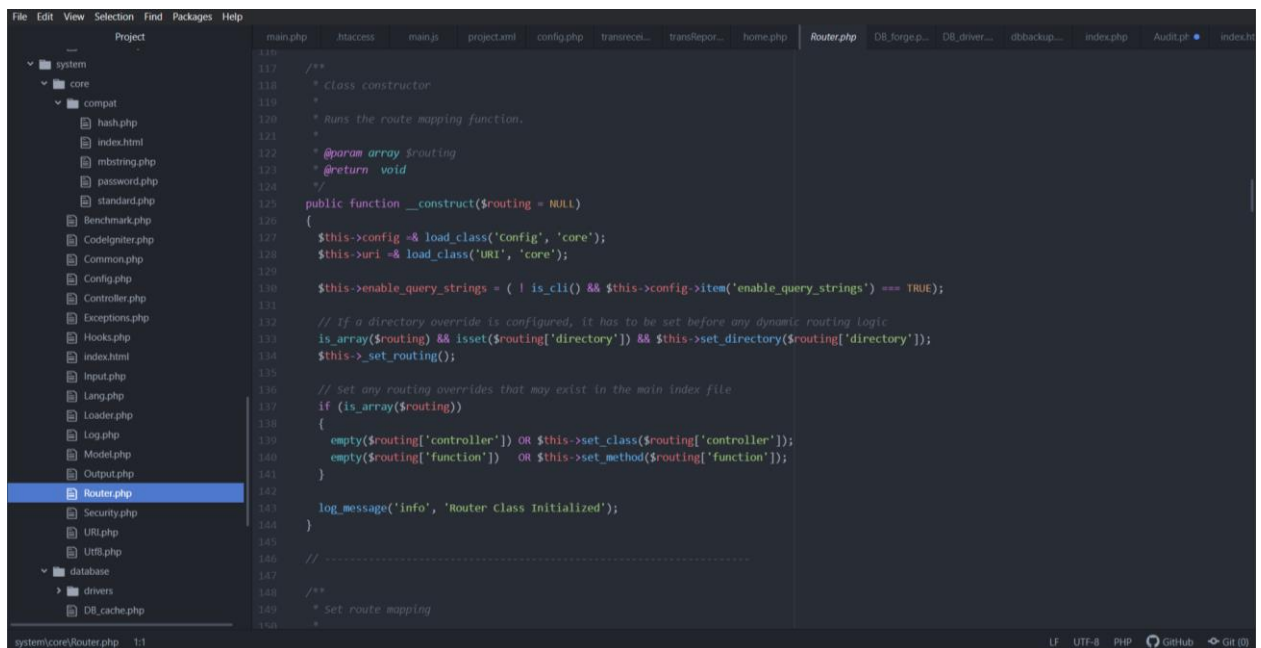


Рисунок 3.11 – Вигляд контролеру Administrator.php Джерело: побудовано автором

Також було створено роутери до контролерів для обробки URL адрес. Роутери визначають, які контролери та дії повинні бути викликані для конкретного HTTP запиту користувача. Підтримка чистого URL відіграє важливу роль в формуванні URL-адреси. Наприклад замість стандартного формату запиту даних, користувач може мати чітку URL-адресу. На рисунку 3.12 представлено результат створення системи роутингу в файлі router.php.



```
116
117
118  /**
119   * Class constructor
120   *
121   * Runs the route mapping function.
122   *
123   * @param array $routing
124   * @return void
125   */
126 public function __construct($routing = NULL)
127 {
128     $this->config =& load_class('config', 'core');
129     $this->uri =& load_class('uri', 'core');
130
131     $this->enable_query_strings = ( ! is_cli() && $this->config->item('enable_query_strings') === TRUE);
132
133     // If a directory override is configured, it has to be set before any dynamic routing logic
134     is_array($routing) && isset($routing['directory']) && $this->set_directory($routing['directory']);
135     $this->set_routing();
136
137     // Set any routing overrides that may exist in the main index file
138     if (is_array($routing))
139     {
140         empty($routing['controller']) OR $this->set_class($routing['controller']);
141         empty($routing['function']) OR $this->set_method($routing['function']);
142     }
143
144     log_message('info', 'Router Class Initialized');
145 }
146
147 // -----
148 /**
149  * Set route mapping
150  */
151
```

Рисунок 3.12 – Результат створення системи роутингу в файлі router.php

Джерело: побудовано автором

Також для автоматизації обчислювальних процесів було створено скрипти обчислень візуальних елементів. На рисунку 3.13 представлено результат створення файлу dashboard.js. Повний лістинг коду представлено в додатку Г.

```
53 $.ajax({
54   type: 'GET',
55   url: appRoot+"dashboard/earningsGraph/"+yearToFetch,
56   dataType: "html"
57 }).done(function(data){
58   var response = jQuery.parseJSON(data);
59
60   var barChartData = {
61     labels: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sept", "Oct", "Nov", "Dec"],
62     datasets: [{
63       fillColor: "rgba(255,255,255,1)", // bar color
64       strokeColor: "rgba(151,187,205,0.8)", // hover color
65       highlightFill: "rgba(242,245,233,1)", // highlight color
66       highlightStroke: "rgba(151,187,205,1)", // highlight hover color
67       data: response.total_earnings
68     }]
69   };
70
71   //show the expense title
72   document.getElementById('earningsTitle').innerHTML = "Earnings (" + response.earningsYear +)";
73
74   var earningsGraph = document.getElementById("earningsGraph").getContext("2d");
75
76   window.myBar = new Chart(earningsGraph).Bar(barChartData, {
77     responsive: true,
78     scaleGridLineColor: "rgba(255,255,255,1)",
79     scalesShowHorizontalLines: true,
80     scalesShowVerticalLines: false,
81     barStrokeWidth: 1,
82     barValueSpacing: 20
83   });
84
85   //remove the loading info
86   $("#yearAccountLoading").html("");
```

Рисунок 3.14 – Результат створення файлу dashboard.js Джерело: побудовано автором

3.3 Приклад роботи web-орієнтованої системи

Для початку користування web-орієнтованою системою користувач повинен отримати дані для входу в систему. На рисунку 3.15 представлено вигляд сторінки авторизації. Після успішної авторизації, користувач отримає повідомлення про успішну авторизацію та буде автоматично переадресований на головну сторінку системи.

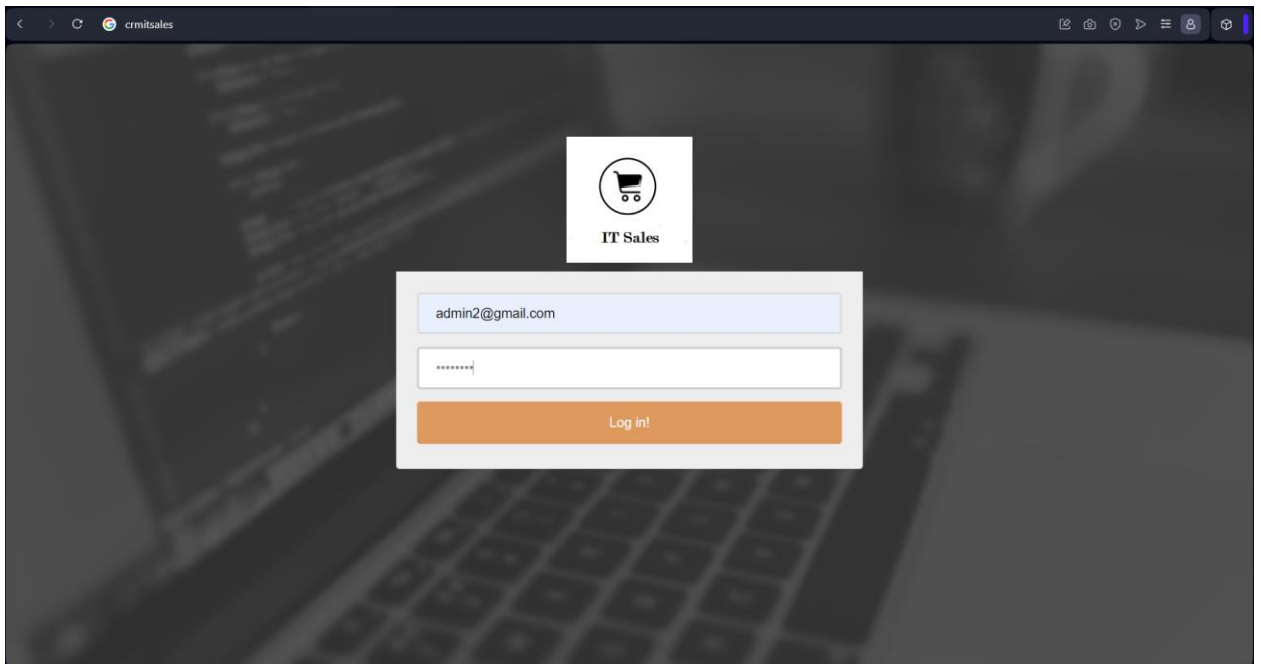


Рисунок 3.15 – Вигляд сторінки авторизації *Джерело: побудовано автором*

При успішній авторизації, користувач отримує роль адміністратора з можливостями змінювати інформацію щодо послуг та замовлення послуг.

На головній сторінці web-системи, адміністратор може переглянути узагальнену інформацію щодо замовлень послуг, а також переглянути графіки замовлень та виконати фільтрацію інформації. На рисунках 3.16 – 3.17 представлено вигляд головної сторінки з узагальненою інформацією про оформлені замовлення.

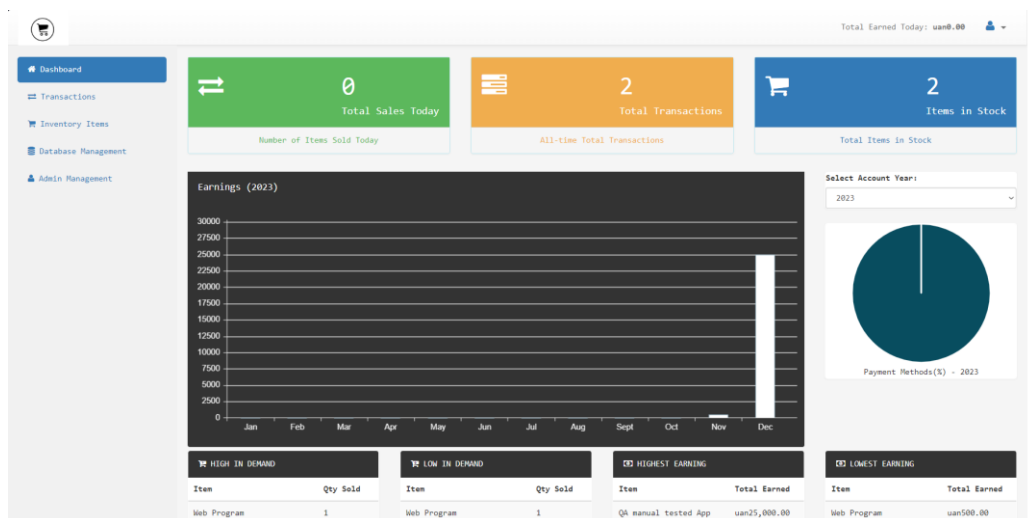


Рисунок 3.16 – Вигляд головної сторінки web-системи. Частина перша
Джерело: побудовано автором

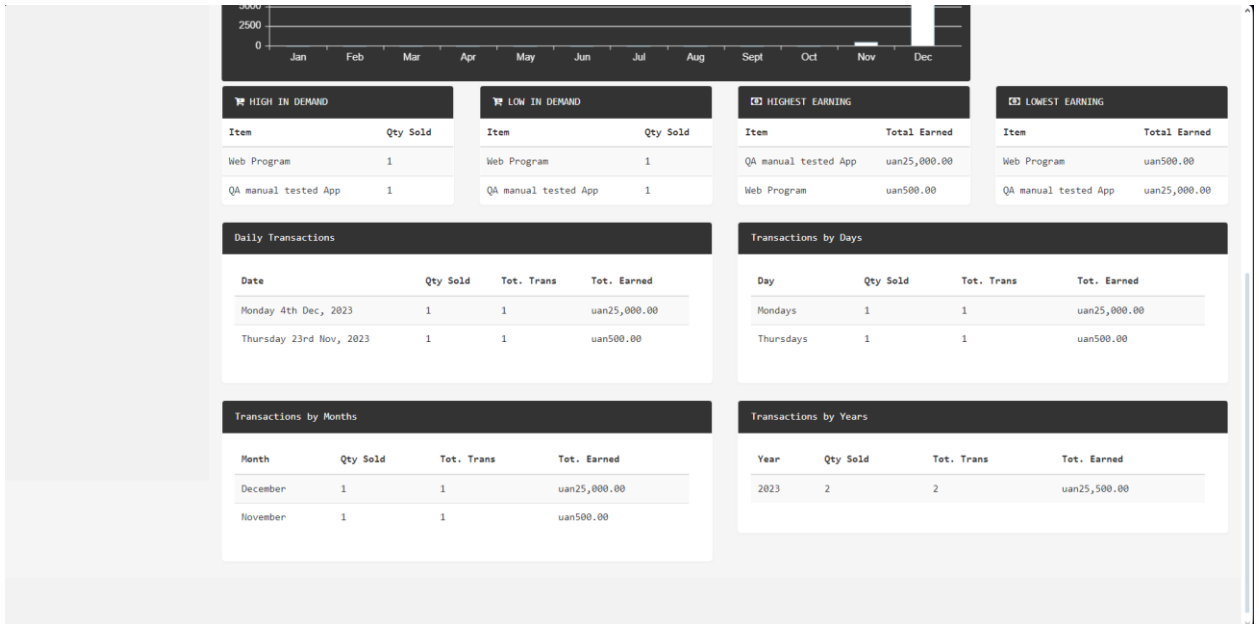


Рисунок 3.17 – Видгляд головної сторінки web-системи. Частина друга
Джерело: побудовано автором

За допомогою навігаційного меню, адміністратор може виконувати навігацію по web-системі. На рисунку 3.18 представлено вигляд навігаційного меню web-системи.

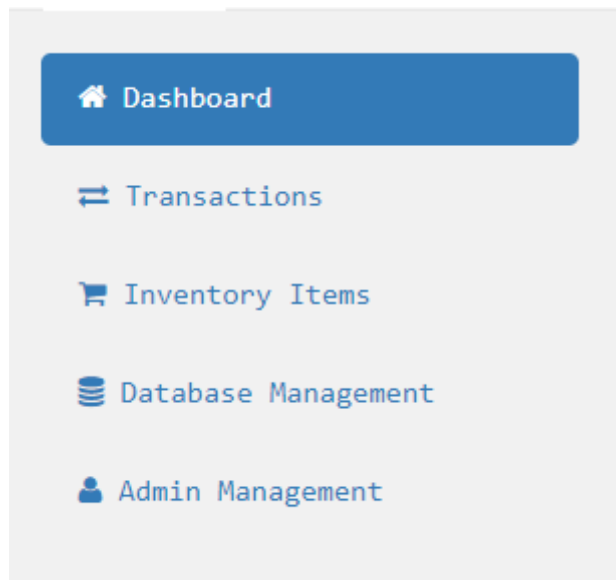


Рисунок 3.18 – Видгляд навігаційного меню web-системи *Джерело: побудовано автором*

При переході на сторінку «Transactions», адміністратор може переглянути оформлені замовлення, а також додати нові замовлення (рис 3.19). Також системою передбачено генерацію звітів в обрані рамки часу. На рисунку 3.18 представлено вигляд сторінки «Transactions».

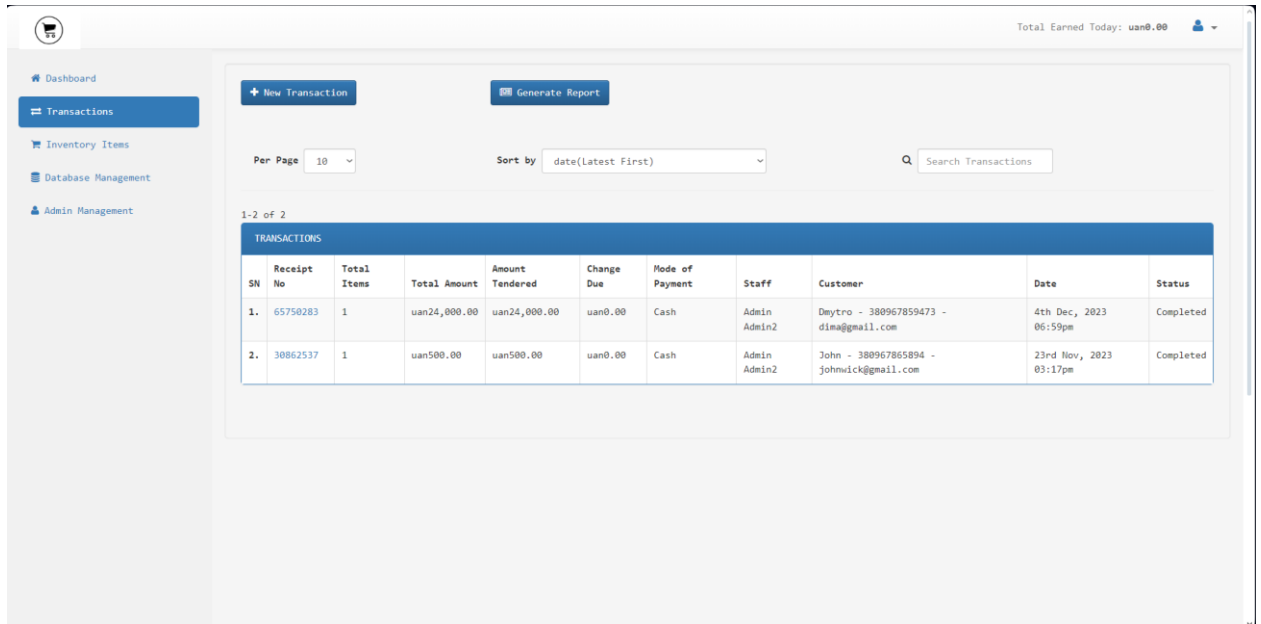


Рисунок 3.18 – Вигляд сторінки «Transactions» Джерело: побудовано автором

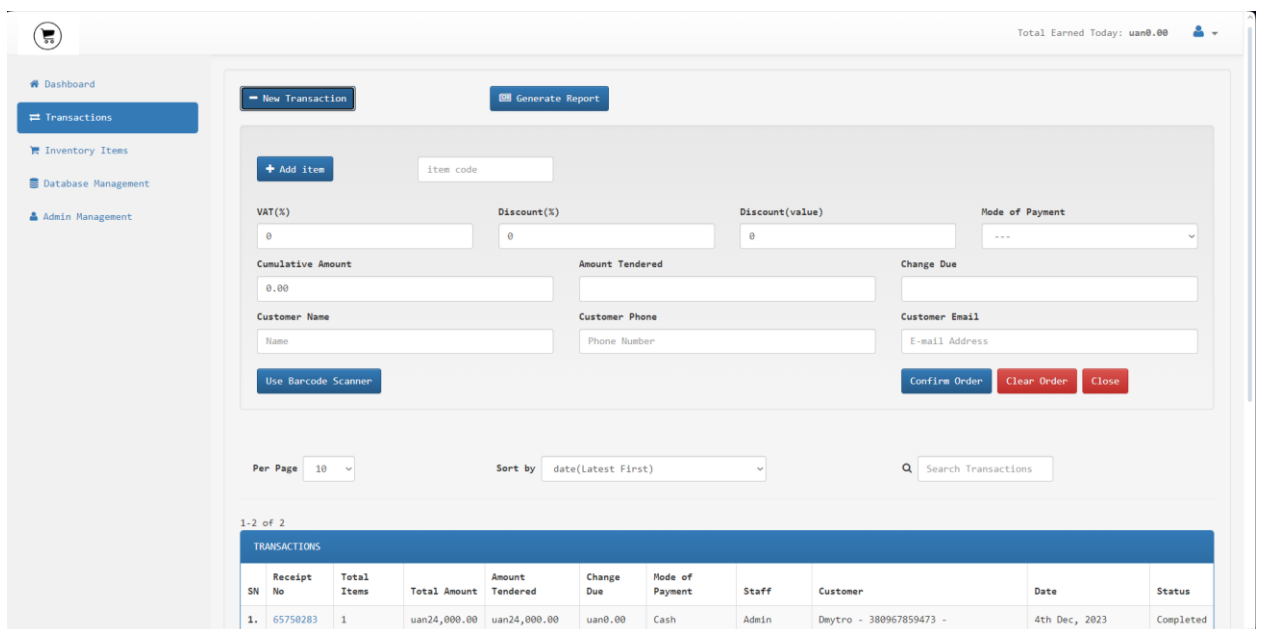
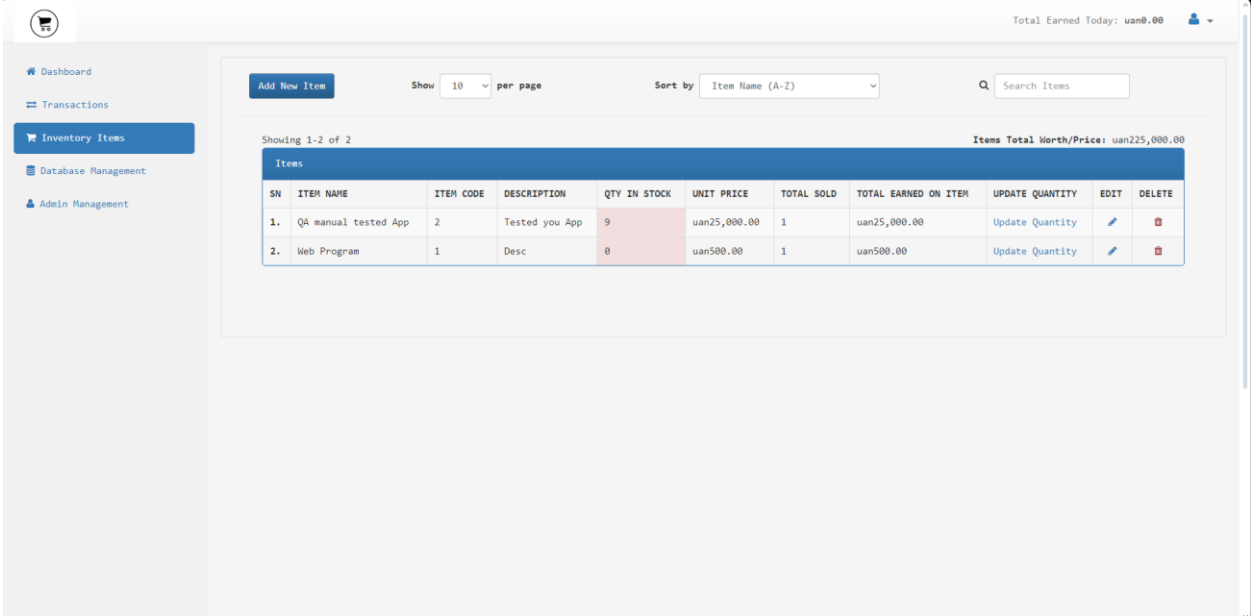


Рисунок 3.19 – Вигляд форми додавання нового замовлення Джерело: побудовано автором

На сторінці «Inventory Items» адміністратор може переглянути всі можливі послуги які може надати ІТ компанія. Також адміністратор може змінити інформацію щодо послуги а також виконати операцію додавання/видалення послуги. На рисунку 3.20 представлено вигляд сторінки «Inventory Items».



SN	ITEM NAME	ITEM CODE	DESCRIPTION	QTY IN STOCK	UNIT PRICE	TOTAL SOLD	TOTAL EARNED ON ITEM	UPDATE QUANTITY	EDIT	DELETE
1.	QA manual tested App	2	Tested you App	9	uan25,000.00	1	uan25,000.00	Update Quantity		
2.	Web Program	1	Desc	0	uan500.00	1	uan500.00	Update Quantity		

Рисунок 3.20 – Вигляд сторінки «Inventory Items» Джерело: побудовано автором

За допомогою форми додавання нової послуги адміністратор може додати нові можливості щодо використання послуг ІТ компанії. На рисунках 3.21 – 3.22 представлено форму додавання нової послуги та результат додавання.

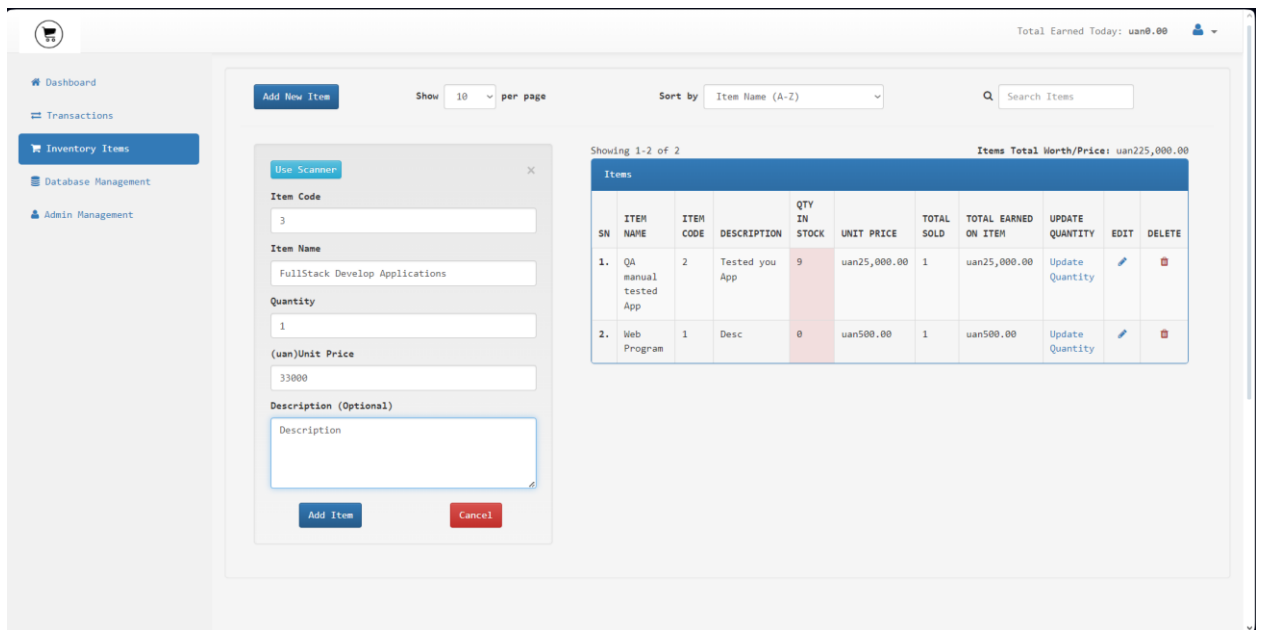


Рисунок 3.21 – Форма додавання нової послуги Джерело: побудовано автором

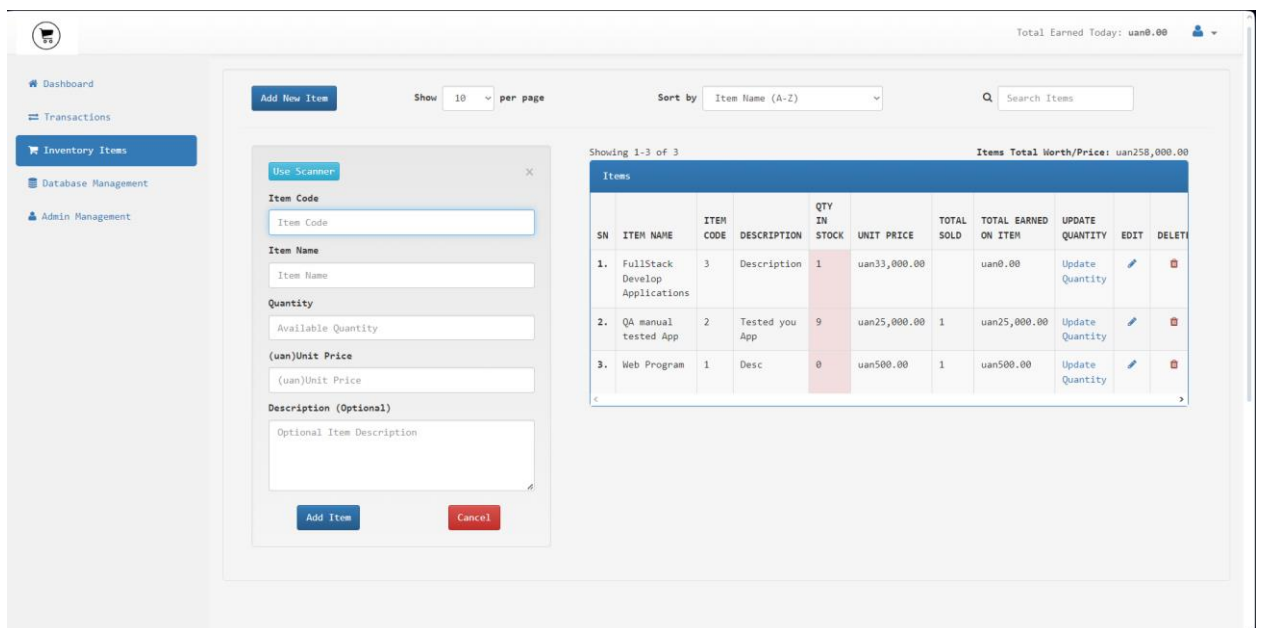


Рисунок 3.22 – Результат додавання нової послуги Джерело: побудовано автором

Для захисту інформації, яка міститься в базі даних, адміністратору надається технологія генерації бази даних з актуальною інформацією. Даний файл зберігається на особистому ПК адміністратора та має розширення .sqlite.

На рисунку 3.23 представлено результат генерації файлу БД та збереження на ПК.

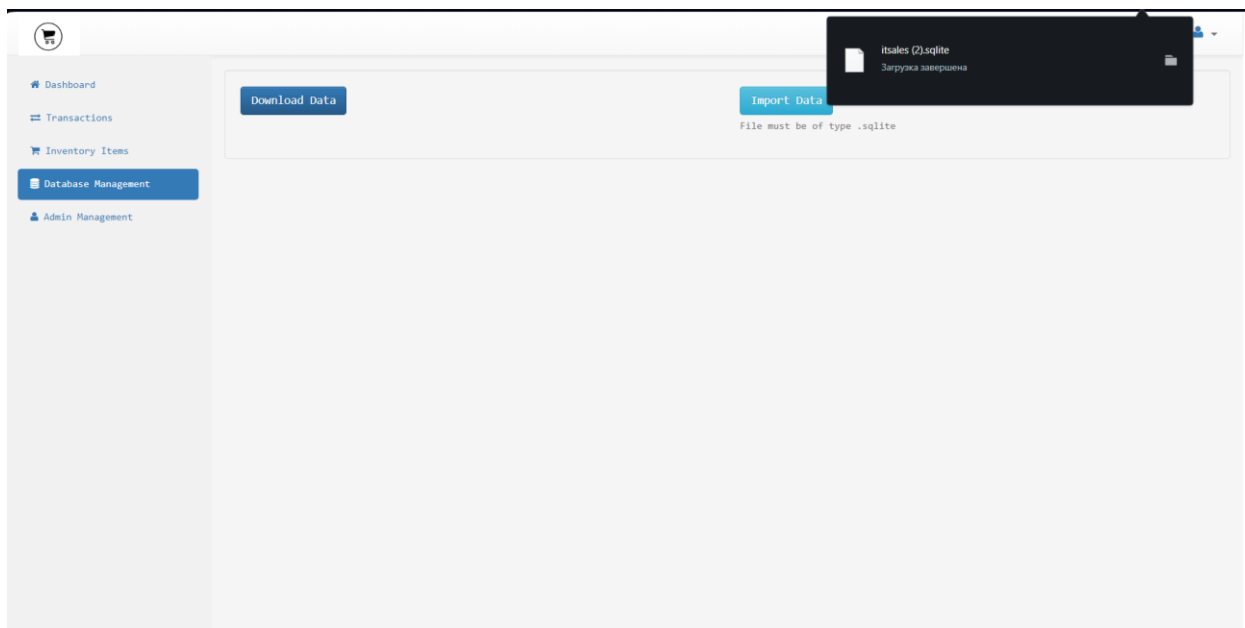


Рисунок 3.23 – Результат генерації файлу БД та збереження на ПК адміна
Джерело: побудовано автором

Також користувач може додавати інших користувачів web-системи та надавати їм права доступу. На рисунку 3.24 представлено форму додавання нового користувача з обранням ролі доступу.

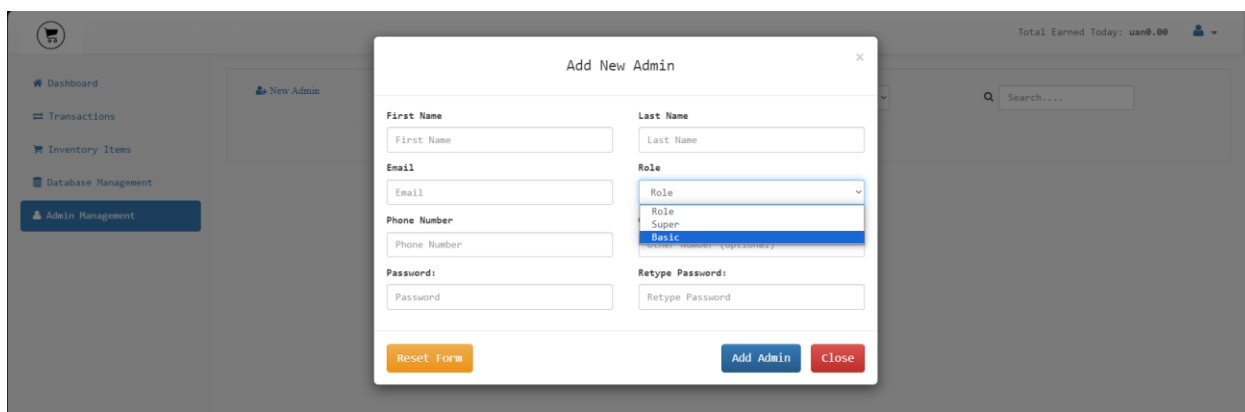


Рисунок 3.24 – Форма додавання нового користувача з наданням ролі
доступу *Джерело: побудовано автором*

3.4 Тестування та рекомендації з розгортання додатку

Для тестування було обрано панель розробника в браузері Орега. Дана панель дозволяє переглянути швидкість завантаження компонентів web-системи. На рисунку 3.25 представлено результат швидкості виконання JSON-запитів.

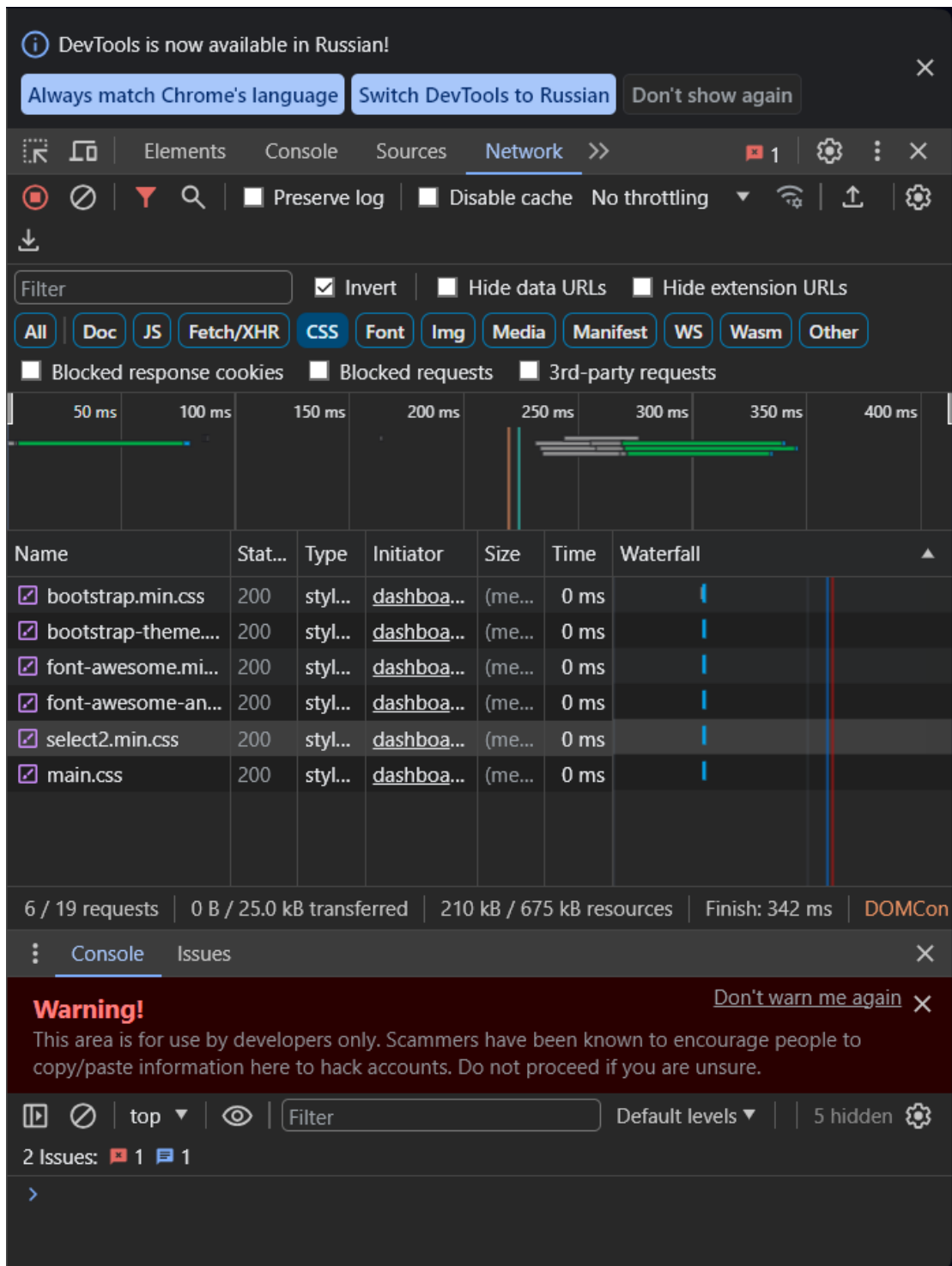


Рисунок 3.25 – Результат тестування швидкості виконання JSON-запитів

Також було протестовано швидкість виконання скриптів та підвантаження CSS файлів (рис 3.26).

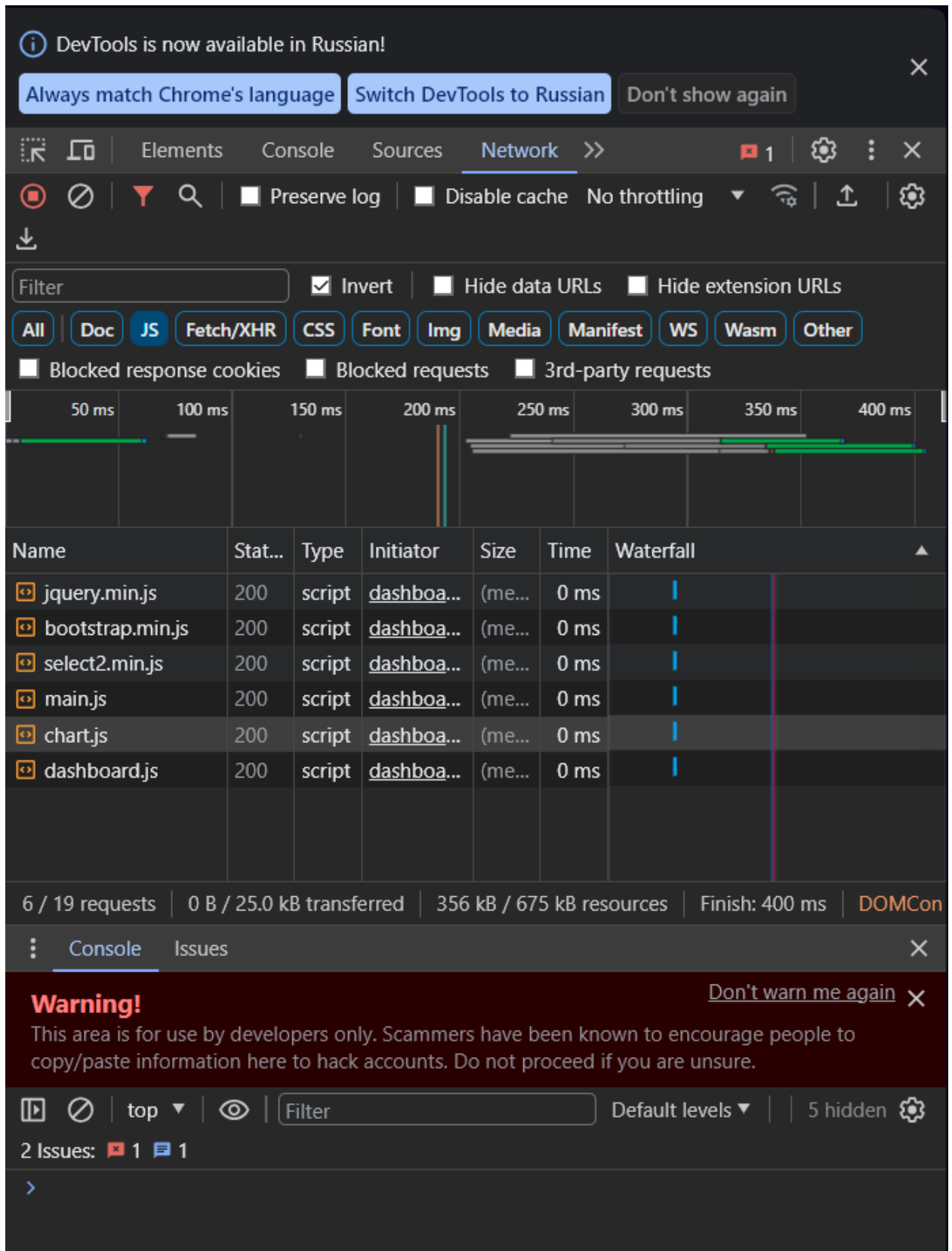


Рисунок 3.26 – Результат швидкості виконання файлів JavaScript

Для розгортання web-додатку було обрано дві хостинг-системи та створено таблицю порівняння:

- HOSTiQ;
- Hostpro;

Таблиця 3.1 – Аналіз хостинг систем для розгортання web-системи

Критерій	HOSTiQ	Hostpro
Вартість	4\$/міс	4.35\$/міс
Наявність вибору домену	-	+
Підтримка 24/7	+	+
Наявність багатопотокової системи обробки даних	-	+
Наявність безкоштовного періоду користування	+	+
Можливість вибору хмарної системи хостингу	+	+
Наявність віртуальних ОС	+	+

Джерело: побудовано автором

Враховуючи всі критерії проаналізованих хостинг-систем було визначено що Hostpro це високоякісний хостинг-провайдер який ідеально підходить для розгортання створеної web-системи. Грунтуючись на наявній інформації, цей хостинг може бути пов'язаний з надійною інфраструктурою, великим обсягом ресурсів, ефективною технічною підтримкою сервісу, який може вирішити надати повний спектр інструментів для підтримки роботи web-системи.

ВИСНОВОК

Результатом виконання кваліфікаційної роботи магістра стало дослідження розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній.

Аналіз систем-аналогів та предметної області надав результат щодо актуальності даної розробки. Було визначено функціональні та нефункціональні вимоги до додатку а також основні цілі та завдання до web-системи.

Також було визначено технології за допомогою яких буде реалізована дана система та обрано базу даних для обробки та збереження даних. За допомогою діаграми Ганта було визначено строки виконання роботи. Для запобігання виникнення ризиків під час процесу розробки було виконано роботу над управліннями ризиками.

Для більш детального управління процесом розробки було використано методики OBS та WBS. Організаційна структура проекту (OBS) визначила відповідальності учасників команди, забезпечуючи чітке розподілення обов'язків та контроль за виконанням завдань.

Робоча структура проекту (WBS) служила основою для розбиття проекту на менші, керовані частини, що сприяло кращому контролю над робочим процесом та ефективному використанню ресурсів.

Щодо актуальності розробки web-сервісу, проведений аналіз систем-аналогів та предметної області підтвердив високий попит на подібні рішення у сучасному ІТ-середовищі. Зростання обсягів даних та швидкі зміни в бізнес-середовищі підкреслили необхідність оптимізації процесів у відділі продажів через впровадження web-орієнтованої системи. Потреба у високоефективних інструментах для збільшення продуктивності та вдосконалення взаємодії з клієнтами робить дану розробку актуальною та важливою для сучасних ІТ компаній.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Importance of Customer Relationship Management in Corporate Wellness Business [Електронний ресурс] – Режим доступу до ресурсу: <https://www.corporatewellnessmagazine.com/article/the-importance-of-customer-relationship-management-crm-in-your-corporate-wellness-business> (Дата звернення 23.11.2023).
2. Importance of Customer Relationship Management [Електронний ресурс] – Режим доступу до ресурсу: <https://www.managementstudyguide.com/importance-of-crm.htm> (Дата звернення 23.11.2023).
3. Minder Chen, Andrew N. K. Chen, Benjamin B. M. Shao THE IMPLICATIONS AND IMPACTS OF WEB SERVICES TO ELECTRONIC COMMERCE RESEARCH AND PRACTICES [Електронний ресурс] – Режим доступу до ресурсу: http://www.jecr.org/sites/default/files/04_4_p01.pdf (Дата звернення 23.11.2023).
4. “Web Services and Enterprise Integration: Friends Not Foes.” [Електронний ресурс] – Режим доступу до ресурсу: http://www.accenture.com/xdoc/en/services/web/insights/web_enterprise.pdf, 2003 (Дата звернення 23.11.2023).
5. Importance of Web-system: Why it Should Top Your Business’s Day 0 Checklist [Електронний ресурс] – Режим доступу до ресурсу: <https://www.theclueless.company/importance-of-crm/> (Дата звернення 23.11.2023).
6. The importance of web service for a business operation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geodatasource.com/resources/tutorials/the-importance-of-web-service-for-a-business-operation/> (Дата звернення 24.11.2023).

7. Santosh S. Venkatraman Web-Services – The Next Evolutionary Stage of E-Business [Электронный ресурс] – Режим доступа до ресурсу: <https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1246&context=jitim> (Дата звернення 24.11.2023).
8. Difference between cloud and client/server computing [Электронный ресурс] – Режим доступа до ресурсу: <https://studyexperts.in/blog/difference-between-cloud-and-client-server-computing/> (Дата звернення 24.11.2023).
9. What is Client-Server Architecture? Everything You Should Know Автор John Terra [Электронный ресурс] – Режим доступа до ресурсу: <https://www.simplilearn.com/what-is-client-server-architecture-article> (Дата звернення 24.11.2023).
10. What is Client Server Architecture? [Электронный ресурс] – Режим доступа до ресурсу: <https://intellipaat.com/blog/what-is-client-server-architecture/> (Дата звернення 24.11.2023).
11. Огляд двадцятки найкращих web-систем для бізнесу [Электронный ресурс] – Режим доступа до ресурсу: <https://esputnik.com/uk/blog/oglyad-dvadcyatki-najkrashih-crm-sistem-dlya-biznesu> (Дата звернення 24.11.2023).
12. Web-система Creatio [Электронный ресурс] – Режим доступа до ресурсу: <https://www.creatio.com/ua/> (Дата звернення 24.11.2023).
13. Web-система LP-System [Электронный ресурс] – Режим доступа до ресурсу: <https://lp-crm.biz/> (Дата звернення 24.11.2023).
14. Prof. (Dr.) Khanna Samrat Vivekanand Omprakash Asst. Prof Mijal Mistry SIGNIFICANCE OF WEB SERVICES IN WEB APPLICATIONS [Электронный ресурс] – Режим доступа до ресурсу: <https://www.technicaljournalonline.com/ijeat/VOL%20III/IJAET%20VOL%20III%20ISSUE%20I%20JANUARY%20MARCH%202012/27%20IAET%20Vol%20III%20Issue%20I%202012.pdf> (Дата звернення 24.11.2023).

15. Simple Object Access Protocol (SOAP) [Электронный ресурс] – Режим доступа до ресурсу: <https://www.microfocus.com/documentation/silk-performer/205/en/silkperformer-205-webhelp-en/GUID-FEFE9379-8382-48C7-984D-55D98D6BFD37.html> (Дата звернення 24.11.2023).
16. What is Web Services Description Language (WSDL)? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.wallarm.com/what/what-is-web-services-description-language-wsdl> (Дата звернення 24.11.2023).
- 17.5 Business Benefits Of Yii Framework For Small Business [Электронный ресурс] – Режим доступа до ресурсу: <https://www.clariontech.com/blog/5-business-benefits-of-yii-framework-for-small-businesses> (Дата звернення 25.11.2023).
18. MySQL Documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.mysql.com/doc/> (Дата звернення 25.11.2023).
19. Рейтинг мов програмування 2023 [Электронный ресурс] – Режим доступа до ресурсу: <https://dou.ua/lenta/articles/language-rating-2023/> (Дата звернення 25.11.2023).
20. What is a DOM and Why is it Important? [Электронный ресурс] – Режим доступа до ресурсу: (Дата звернення 25.11.2023).
21. Erasmus, J.; Vanderfeesten, I.; Traganos, K.; Grefen, P. Using business process models for the specification of manufacturing operations. *Comp. Ind.* 2020, 123, 1–21.
22. Kassem, M.; Dawood, N.; Mitchell, D. A Structured Methodology for Enterprise Modelling: A case study for Modelling the Operation of a British Organization. *J. Inf. Technol. Constr.* 2011, 16, 381–410. (Дата звернення 25.11.2023).
23. Lang, A.-M.; Ellefmo, S.-L.; Aasly, K. Geo-metallurgical Flowsheet as a tool for designing and communicating the Geo-Metallurgical Programs. *Minerals* 2018, 8, 372. (Дата звернення 25.11.2023).

24. Modeling Web Application Architectures with UML [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.huihoo.com/development/rup/webapps.htm> (Дата звернення 25.11.2023).
- 25.7 Benefits of Using MySQL in Your Business [Электронный ресурс] – Режим доступа до ресурсу: <https://bootcamp.uxdesign.cc/7-benefits-of-using-mysql-in-your-business-e587e326144f> (Дата звернення 25.11.2023).
26. SMART Project Management to Achieve Business Goals [Электронный ресурс] – Режим доступа до ресурсу: <https://www.bluelinkerp.com/blog/smart-project-management-to-achieve-your-business-goals/> (Дата звернення 25.11.2023).
27. What are the advantages and disadvantages of using a WBS vs a gantt chart for different types of projects? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.linkedin.com/advice/0/what-advantages-disadvantages-using-wbs-vs-gantt> (Дата звернення 25.11.2023).
28. Importance of Gantt Charts [Электронный ресурс] – Режим доступа до ресурсу: <https://projectsly.com/importance-of-gantt-charts> (Дата звернення 25.11.2023).
29. Risk Analysis in Project Management: Steps and Benefits [Электронный ресурс] – Режим доступа до ресурсу: <https://www.indeed.com/career-advice/career-development/risk-analysis-project-management> (Дата звернення 25.11.2023).
30. Why is Risk Management Important in Project Management [Электронный ресурс] – Режим доступа до ресурсу: https://riskpublishing.com/why-is-risk-management-important-in-project-management/#google_vignette (Дата звернення 25.11.2023).

ДОДАТОК А

Деталізація мети проєкту за допомогою методу SMART

Формула SMART-цілей зазвичай асоціюється з концепцією управління за цілями Пітера Друкера. SMART-цілі заохочують використовувати конкретні завдання для деталізації того, як буде досягнута мета [26].

Потреба в формулюванні SMART-цілі має важливість в тому плані, щоб вони відповідали наступним критеріям:

Конкретність: Чи має ціль конкретні засоби та цілі?

Вимірюваність: Чи можна виміряти ціль? Як?

Здійсненна/досяжна: Які конкретні дії приведуть до досягнення цієї мети? Що потрібно зробити, щоб досягти мети?

Доречно/Реалістична: Чи стосується ця мета посадових обов'язків, команди та компанії? Чи базується вона на факторах, які знаходяться під контролем?

Обмежена в часі: Який період часу? Чи залежить він від дедлайну або кінцевої дати, чи це звичайний графік?

Постановка SMART-цілей допомагає вам зануритися в деталі, створюючи відчуття терміновості. Хоча дуже важливо слідувати широким формулюванням місії, відсутність конкретики означає, що немає способу забезпечити відповідність цілей бізнес-цілям.

Підхід SMART-цілей допомагає всім - від керівників до окремих співробітників - знати, що їм потрібно робити.

Результати деталізації мети проєкту методом SMART представлені у таблиці Б.1

Таблиця Б.1 – Формалізація мети за методом SMART

Конкретність	Розробка web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній, яка дозволить проводити аналіз замовлень послуг та надаватиме інструменти для створення звітів зі збереженням даних.
Вимірюваність	Вимірюється за кількістю оформлених транзакцій та статистикою замовлення послуг.
Досяжність	Проєкт реалізовується на основі затвердженого ТЗ та відповідно до рівня досвіду.
Значимість	На основі потреб та очікувань клієнтів, дана розробка надасть великий вплив на підвищення конкурентоспроможності в сфері web-орієнтованих систем підтримки відділу продажів.
Обмеження в часі	Проєкт має бути впроваджений та завершений впродовж визначеного терміну, згідно з розробленим графіком робіт

Work Breakdown Structure (WBS)

Структура розбиття робіт (WBS) - це ключовий результат проєкту, який організовує роботу команди на керовані частини. Структура розподілу робіт - це ієрархічна декомпозиція загального обсягу робіт, які має виконати команда проєкту для досягнення цілей проєкту та створення необхідних результатів. WBS також забезпечує необхідну основу для детальної оцінки та контролю витрат, а також для розробки та контролю графіку [26].

Також це ієрархічна деревоподібна структура, яка окреслює ваш проєкт і розбиває його на менші, більш керовані частини. Це корисний спосіб розділяти та підкорювати великі проєкти, щоб ви могли виконувати роботу швидше та ефективніше.

Структура розбиття робіт має кілька переваг на додаток до визначення та організації роботи над проектом. Розподіляючи час і кошторис витрат на конкретні розділи структури розподілу робіт, можна швидко розробити графік і бюджет проекту. Під час виконання проекту можна відстежувати виконання окремих розділів структури робіт, щоб визначити ефективність витрат за проектом і виявити питання і проблемні місця в організації проекту.

На рисунку Б.1 представлено WBS структуру проекту щодо розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній.

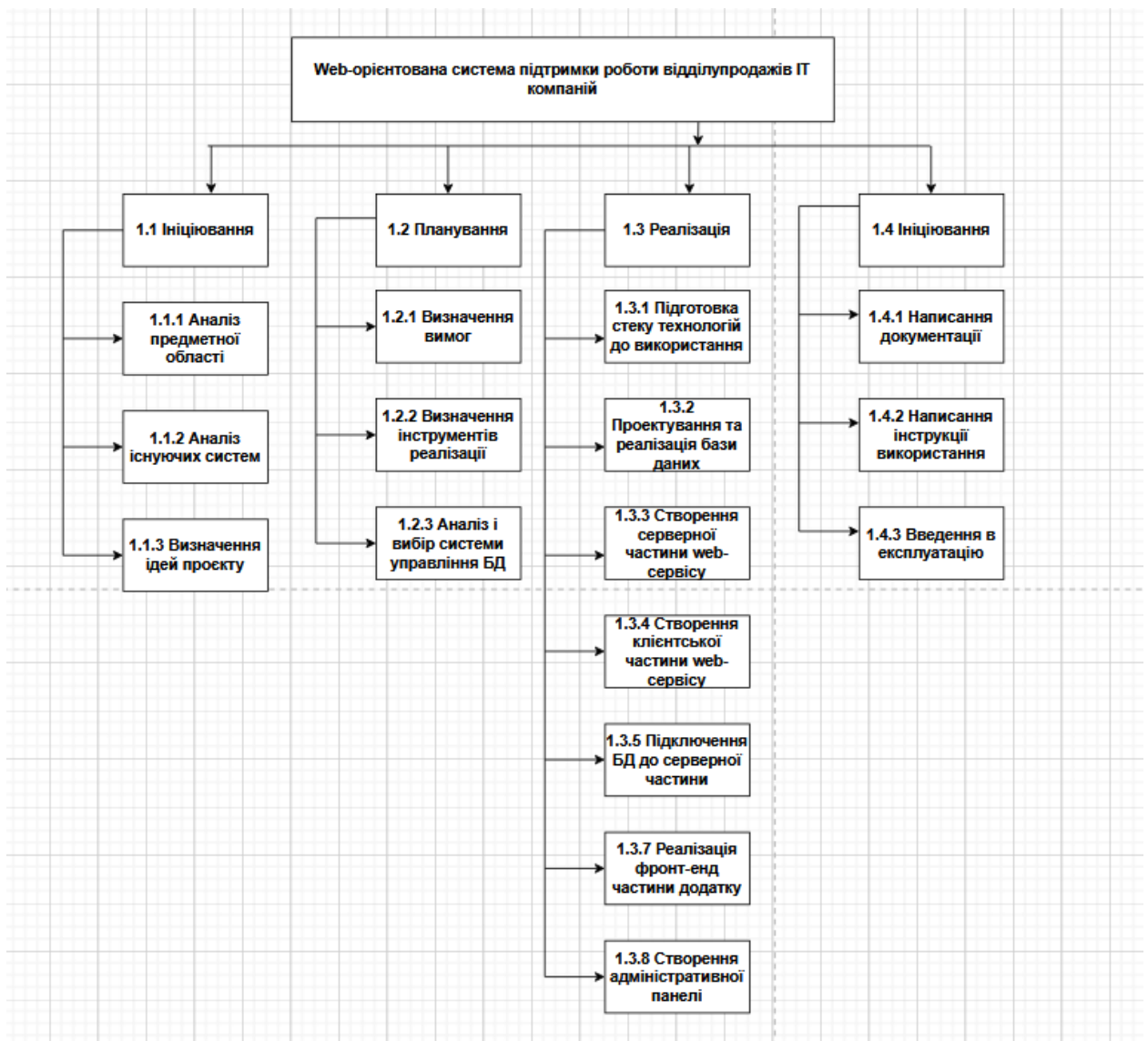


Рисунок Б.1 - WBS структура проекту щодо розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній

Організаційна структура робіт (OBS)

Організаційну структуру можна визначити як ієрархічну або структурну організацію компанії, яка повинна визначати членів команди в порядку важливості від верхівки компанії до низу, а також завдання, за які ці працівники відповідають.

Важливість даної структури заключається в визначенні і встановленні стандарту для того, хто, коли і де знаходиться особа або працівник компанії в організації. Якби не було організаційної структури, то працівники не знали б і не розуміли б, хто є керівником, а хто - підлеглим. Вона забезпечує розбивку від верхівки компанії до низу і відображає, хто за що відповідає [27].

На рисунку Б.2 представлено OBS структуру щодо розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній

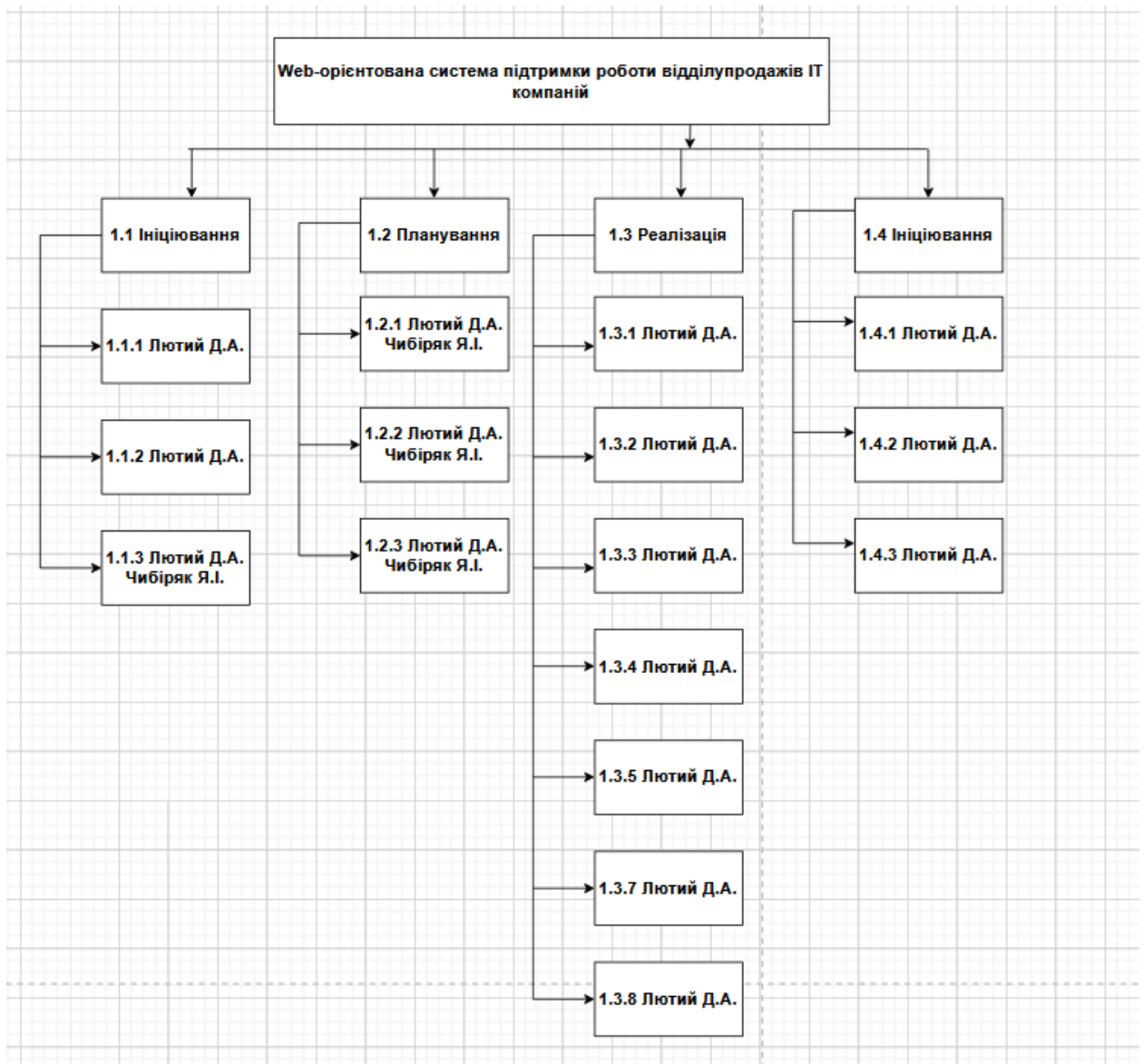


Рисунок Б.2 - OBS структура щодо розробки web-орієнтованої системи підтримки роботи відділу продажів ІТ компаній

Діаграма Ганта

Діаграми Ганта - це візуальне представлення завдань, побудованих у часі. Вони відображають важливу інформацію в проєкті, наприклад, хто за що відповідає, тривалість завдань і діяльність, що перетинається. Діаграма Ганта відображає завершення кожної роботи в проєкті за допомогою часових шкал. Ці часові рамки пояснюють, як пов'язані між собою різні завдання [28].

Діаграми Ганта показують, як робота над проектом може початися тільки після завершення іншого завдання. Менеджери можуть використовувати ці дані, щоб краще зрозуміти перешкоди в проекті і прийняти ключові рішення про те, як просунути проект вперед.

На рисунках Б.3 – Б.4 представлено діаграму Ганта з розподілом дат

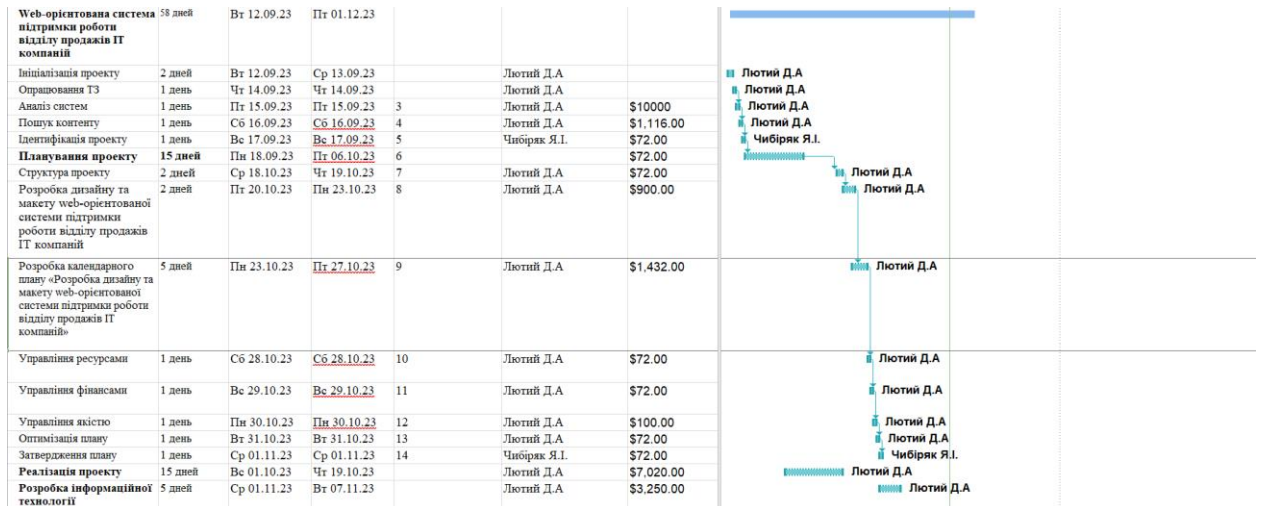


Рисунок Б.3 – Діаграма Ганта. Частина 1

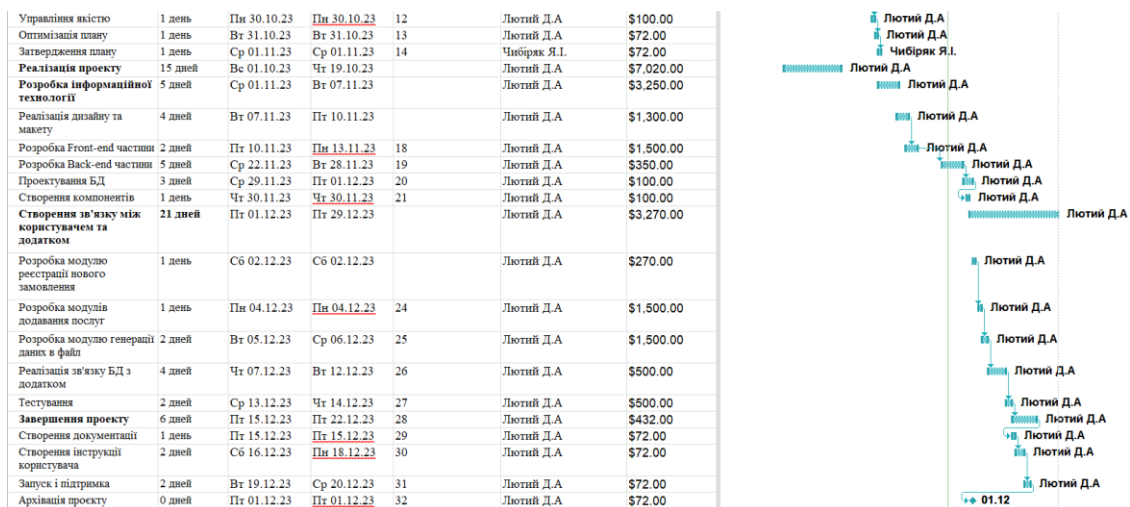


Рисунок Б.4 – Діаграма Ганта. Частина 2

Управління ризиками проекту

Аналіз ризиків в управлінні проектами - це структурований моніторинг ризиків, які можуть вплинути на терміни та якість проекту. Аналіз ризиків визначає ризики, оцінює ймовірність їх виникнення, окреслює кроки, яких слід вжити, щоб уникнути цих ризиків, і визначає, як їх вирішити, якщо вони

виникли. Успіх проекту часто залежить від якості аналізу ризиків, оскільки неочікуваний ризик може негативно вплинути на проект. Можна отримати життєво важливу і детальну інформацію з різних джерел, таких як плани проекту, прогнози продажів, фінансова звітність і протоколи безпеки [29].

Аналіз ризиків передбачає визначення кількісного та якісного впливу ідентифікованих ризиків, і потреби проекту можуть диктувати, який з них використовується [30].

Таблиця А.2. Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Проблеми з обробкою та хешуванням даних	2	4
2	Невірний вибір технологій або архітектури	3	3
3	Зміни в стратегії бізнесу або вимогах користувачів	3	5
4	Недостатня масштабованість	2	3
5	Недооцінка обсягу робіт та ресурсів	4	5
6	Відсутність резервних копій даних	2	3

Таблиця А.3 – Матриця впливу

Вірогідність виникнення	Матриця впливу				
5			3	5	
4		4			
3		6	2		
2				1	
1					

Ступінь впливу	1	2	3	4	5
----------------	---	---	---	---	---

ДОДАТОК Б

Лістинг коду config.php

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');

$host    =    $_SERVER['HTTP_HOST'];//e.g    practa.ng    ||
www.practa.ng || localhost
$protocol = is_https() ? "https://" : "http://";

if(is_cli()){
    $config['base_url'] = "";
}

else if(stristr($host,    "localhost")    !==    FALSE    ||
(stristr($host,    "192.168.")    !==    FALSE)|| (stristr($host,
"127.0.0.")    !==    FALSE)){

    $config['base_url'] = $protocol.$host."/";
}

else{
    $allowed                =                ['www.1410inc.xyz',
'1410inc.xyz'];//input    all    allowed    urls    here    e.g.
amirsanni.com, www.practa.ng, smartagapp.com

    $config['base_url'] = in_array($host,    $allowed)    ?
$protocol.$host."/":
"http://".$_SERVER['HTTP_HOST']."/";
}
```

```
$config['index_page'] = '';

$config['uri_protocol'] = 'REQUEST_URI';

$config['url_suffix'] = '';

$config['language'] = 'english';

$config['charset'] = 'UTF-8';

$config['enable_hooks'] = FALSE;

$config['subclass_prefix'] = 'MY_';

$config['composer_autoload'] = FALSE;

$config['permitted_uri_chars'] = 'a-z 0-9~%.:_\-';

$config['allow_get_array'] = TRUE;
$config['enable_query_strings'] = FALSE;
$config['controller_trigger'] = 'c';
$config['function_trigger'] = 'm';
$config['directory_trigger'] = 'd';

$config['log_threshold'] = 1;

$config['log_path'] = '';

$config['log_file_extension'] = '';
```

```
$config['log_file_permissions'] = 0644;

$config['log_date_format'] = 'Y-m-d H:i:s';

$config['error_views_path'] = '';

$config['cache_path'] = '';

$config['cache_query_string'] = FALSE;

$config['encryption_key'] =
hex2bin('31016163f8b7f43caad37fe6812048da');

$config['sess_driver'] = 'files';
$config['sess_cookie_name'] = '_1410__';
$config['sess_expiration'] = 7200;
$config['sess_save_path'] =
BASEPATH.'sessions';//'_1410__sess';
$config['sess_match_ip'] = FALSE;
$config['sess_time_to_update'] = 300;
$config['sess_regenerate_destroy'] = FALSE;

$config['cookie_prefix'] = '';
$config['cookie_domain'] = '';
$config['cookie_path'] = '/';
$config['cookie_secure'] = FALSE;
$config['cookie_httponly'] = TRUE;

$config['standardize_newlines'] = TRUE;
```

```
$config['global_xss_filtering'] = FALSE;

$config['csrf_protection'] = FALSE;
$config['csrf_token_name'] = 'csrf_test_name';
$config['csrf_cookie_name'] = 'csrf_cookie_name';
$config['csrf_expire'] = 7200;
$config['csrf_regenerate'] = TRUE;
$config['csrf_exclude_uris'] = array();

$config['compress_output'] = FALSE;

$config['time_reference'] = 'local';

$config['rewrite_short_tags'] = FALSE;

$config['proxy_ips'] = '';
```

ДОДАТОК В

Лістинг коду Administrator.php

```
<?php
defined('BASEPATH') or exit('');

class Administrators extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();

        $this->genlib->checkLogin();

        $this->genlib->superOnly();

        $this->load->model(['admin']);
    }
    public function index()
    {
        $data['pageContent'] = $this->load->view('admin/admin', '', TRUE);
        $data['pageTitle'] = "Administrators";

        $this->load->view('main', $data);
    }
    public function laad_()
    {
        //set the sort order
        $orderBy = $this->input->get('orderBy', TRUE) ?
        $this->input->get('orderBy', TRUE) : "first_name";
```



```

        $orderFormat      =      $this->input->get('orderFormat',
TRUE) ? $this->input->get('orderFormat', TRUE) : "ASC";

        //count the total administrators in db (excluding the
currently logged in admin)
        $totalAdministrators      =      count($this->admin-
>getAll());

        $this->load->library('pagination');

        $pageNumber = $this->uri->segment(3, 0);

        $limit = $this->input->get('limit', TRUE) ? $this-
>input->get('limit', TRUE) : 10;
        $start = $pageNumber == 0 ? 0 : ($pageNumber - 1) *
$limit;

        //call setPaginationConfig($totalRows, $urlToCall,
$limit, $attributes) in genlib to configure pagination
        $config      =      $this->genlib-
>setPaginationConfig($totalAdministrators,
"administrators/laad_", $limit, ['class' => 'lnp']);

        $this->pagination->initialize($config);

        //get all customers from db
        $data['allAdministrators']      =      $this->admin-
>getAll($orderBy, $orderFormat, $start, $limit);

```

```

        $data['range'] = $totalAdministrators > 0 ? ($start
+ 1) . "-" . ($start + count($data['allAdministrators']))
. " of " . $totalAdministrators : "";
        $data['links'] = $this->pagination->create_links();
//page links
        $data['sn'] = $start + 1;

        $json['adminTable'] = $this->load-
>view('admin/adminlist', $data, TRUE); //get view with
populated customers table

        $this->output-
>set_content_type('application/json')-
>set_output(json_encode($json));
    }

    public function add()
    {
        $this->genlib->ajaxOnly();

        $this->load->library('form_validation');

        $this->form_validation->set_error_delimiters(' ',
' ');

        $this->form_validation->set_rules('firstName',
'First name', ['required', 'trim', 'max_length[20]',
'strtolower', 'ucfirst'], ['required' => "required"]);

```

```

        $this->form_validation->set_rules('lastName', 'Last
name', ['required', 'trim', 'max_length[20]',
'strtolower', 'ucfirst'], ['required' => "required"]);
        $this->form_validation->set_rules(
            'email',
            'E-mail',
            ['trim', 'required', 'valid_email',
'is_unique[admin.email]', 'strtolower'],
            ['required' => "required", 'is_unique' => 'E-mail
exists']
        );
        $this->form_validation->set_rules('role', 'Role',
['required'], ['required' => "required"]);
        $this->form_validation->set_rules(
            'mobile1',
            'Phone number',
            ['required', 'trim', 'numeric', 'max_length[15]',
'min_length[11]', 'is_unique[admin.mobile1]'],
            ['required' => "required", 'is_unique' => "This
number is already attached to an admin"]
        );
        $this->form_validation->set_rules('mobile2', 'Other
number', ['trim', 'numeric', 'max_length[15]',
'min_length[11]']);
        $this->form_validation->set_rules('passwordOrig',
'Password', ['required', 'min_length[8]'], ['required'
=> "Enter password"]);
        $this->form_validation->set_rules('passwordDup',
'Password Confirmation', ['required',

```

```

'matches[passwordOrig]'], ['required' => "Please retype
password"]);
    if ($this->form_validation->run() !== FALSE) {
        $hashedPassword =
password_hash(set_value('passwordOrig'),
PASSWORD_BCRYPT);

        $inserted = $this->admin->add(
            set_value('firstName'),
            set_value('lastName'),
            set_value('email'),
            $hashedPassword,
            set_value('role'),
            set_value('mobile1'),
            set_value('mobile2')
        );
        $json = $inserted ?
            ['status' => 1, 'msg' => "Admin account
successfully created"]
            :
            ['status' => 0, 'msg' => "Oops! Unexpected server
error! Pls contact administrator for help. Sorry for the
embarrassment"];
    } else {
        //return all error messages
        $json = $this->form_validation->error_array();
        //get an array of all errors

        $json['msg'] = "One or more required fields are
empty or not correctly filled";
    }
}

```

```

        $json['status'] = 0;
    }

    $this->output-
>set_content_type('application/json')-
>set_output(json_encode($json));
    }
    public function update()
    {
        $this->genlib->ajaxOnly();

        $this->load->library('form_validation');

        $this->form_validation->set_error_delimiters(' ',
    '');

        $this->form_validation->set_rules('firstName',
    'First name', ['required', 'trim', 'max_length[20]'],
    ['required' => "required"]);
        $this->form_validation->set_rules('lastName', 'Last
    name', ['required', 'trim', 'max_length[20]'],
    ['required' => "required"]);
        $this->form_validation->set_rules('mobile1', 'Phone
    number', [
            'required', 'trim', 'numeric', 'max_length[15]',
            'min_length[11]', 'callback_crosscheckMobile[' .
    $this->input->post('adminId', TRUE) . ']'
        ], ['required' => "required"]);
    }

```

```

        $this->form_validation->set_rules('mobile2', 'Other
number', ['trim', 'numeric', 'max_length[15]',
'min_length[11]]);

        $this->form_validation->set_rules('email', 'E-
mail', ['required', 'trim', 'valid_email',
'callback_crosscheckEmail[' . $this->input-
>post('adminId', TRUE) . ']]);

        $this->form_validation->set_rules('role', 'Role',
['required', 'trim'], ['required' => "required"]);

if ($this->form_validation->run() !== FALSE) {
    $admin_id = $this->input->post('adminId', TRUE);

    $updated = $this->admin->update(
        $admin_id,
        set_value('firstName'),
        set_value('lastName'),
        set_value('email'),
        set_value('mobile1'),
        set_value('mobile2'),
        set_value('role')
    );

    $json = $updated ?
        ['status' => 1, 'msg' => "Admin info successfully
updated"]
        :
        ['status' => 0, 'msg' => "Oops! Unexpected server
error! Pls contact administrator for help. Sorry for the
embarrassment"];
    } else {

```

```

        $json = $this->form_validation->error_array();
//get an array of all errors

        $json['msg'] = "One or more required fields are
empty or not correctly filled";
        $json['status'] = 0;
    }

    $this->output-
>set_content_type('application/json')-
>set_output(json_encode($json));
}

public function suspend()
{
    $this->genlib->ajaxOnly();

    $admin_id = $this->input->post('_aId');
    $new_status = $this->genmod->gettablecol('admin',
'account_status', 'id', $admin_id) == 1 ? 0 : 1;

    $done = $this->admin->suspend($admin_id,
$new_status);

    $json['status'] = $done ? 1 : 0;
    $json['_ns'] = $new_status;
    $json['_aId'] = $admin_id;

```

```

        $this->output-
>set_content_type('application/json')-
>set_output(json_encode($json));
    }
    public function delete()
    {
        $this->genlib->ajaxOnly();

        $admin_id = $this->input->post('_aId');
        $new_value = $this->genmod->getTableCol('admin',
'deleted', 'id', $admin_id) == 1 ? 0 : 1;

        $done = $this->admin->delete($admin_id, $new_value);

        $json['status'] = $done ? 1 : 0;
        $json['_nv'] = $new_value;
        $json['_aId'] = $admin_id;

        $this->output-
>set_content_type('application/json')-
>set_output(json_encode($json));
    }
    public function crosscheckMobile($mobile_number,
$admin_id)
    {

        $adminWithNum = $this->genmod->getTableCol('admin',
'id', 'mobile1', $mobile_number);

        if ($adminWithNum == $admin_id) {

```



```

        //used for same admin. All is well.
        return TRUE;
    } else {
        $this->form_validation-
>set_message('crosscheckMobile', 'This number is already
attached to an administrator');

        return FALSE;
    }
}

public function crosscheckEmail($email, $admin_id)
{
    $adminWithEmail          =          $this->genmod-
>getTableCol('admin', 'id', 'email', $email);

    if ($adminWithEmail == $admin_id) {
        //used for same admin. All is well.
        return TRUE;
    } else {
        $this->form_validation-
>set_message('crosscheckEmail', 'This email is already
attached to an administrator');

        return FALSE;
    }
}
}

```

ДОДАТОК Г

Лістинг коду dashboard.js

```
'use strict';

$(document).ready(function() {
  checkDocumentVisibility(checkLogin);
  getEarnings();
  loadPaymentMethodChart();
  $("#earningAndExpenseYear").change(function() {
    var year = $(this).val();

    if(year) {
      $("#yearAccountLoading").html("<i
class='"+spinnerClass+"'></i> Loading...");
      getEarnings(year);
      loadPaymentMethodChart(year);
    }
  });
});

function getEarnings(year) {
  var yearToFetch = year || '';

  $.ajax({
    type: 'GET',
    url: appRoot+"dashboard/earningsGraph/"+yearToFetch,
    dataType: "html"
  }).done(function(data) {
    var response = jQuery.parseJSON(data);

    var barChartData = {
```

```

labels : ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
"Jul", "Aug", "Sept", "Oct", "Nov", "Dec"],
datasets : [{
fillColor : "rgba(255,255,255,1)",
strokeColor : "rgba(151,187,205,0.8)",
highlightFill : "rgba(242,245,233,1)",
highlightStroke : "rgba(151,187,205,1)",
data : response.total_earnings
}]
};
document.getElementById('earningsTitle').innerHTML =
"Earnings (" + response.earningsYear +)";

var earningsGraph =
document.getElementById("earningsGraph").getContext("2d
");

window.myBar = new
Chart(earningsGraph).Bar(barChartData, {
responsive : true,
scaleGridLineColor : "rgba(255,255,255,1)",
scaleShowHorizontalLines: true,
scaleShowVerticalLines: false,
barStrokeWidth : 1,
barValueSpacing : 20
});
$("#yearAccountLoading").html("");
}).fail(function(){
console.log('req failed');
});

```

```
}  
function loadPaymentMethodChart(year) {  
var yearToGet = year ? year : "";  
  
$.ajax({  
type: 'GET',  
url: appRoot+"dashboard/paymentmethodchart/"+yearToGet,  
dataType: "html",  
success: function(data) {  
var response = jQuery.parseJSON(data);  
var cash = response.cash;  
var pos = response.pos;  
var cashAndPos = response.cashAndPos;  
  
if(response.status === 1) {  
if((cash === 0 && pos === 0 && cashAndPos === 0)) {  
var paymentMethodData = [{  
value: 1,  
color:"#4D5360",  
highlight: "#616774",  
label: "No Payment"  
}];  
}  
  
else {  
var paymentMethodData = [{  
value: cash,  
color:"#084D5F",  
highlight: "#0B6B85",  
label: "Cash Only"}]  
}
```

```

    }, {
    value: pos,
    color: "#557f7c",
    highlight: "#556f7c",
    label: "POS Only"
    }, {
    value: cashAndPos,
    color: "#333",
    highlight: "pink",
    label: "Cash and POS"
    }];
}
}

else {
var paymentMethodData = [{
value: 1,
color:"#4D5360",
highlight: "#616774",
label: "No Payment"
}];
}

var ctx =
document.getElementById("paymentMethodChart").getContext
t("2d");
window.myPie = new Chart(ctx).Pie(paymentMethodData);
$("#yearAccountLoading").html("");
$("#paymentMethodYear").html("- "+response.year);
}

```

}) ;

}