

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка
до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«Розробка програмного забезпечення для прогнозування
продажів товарів мовою Python»**

Виконав: студент 4 курсу, групи ПД-41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

_____ Бондар Д.В.
(прізвище та ініціали)

Керівник _____ Трінтіна Н.А.
(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти -«Бакалавр»
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інженерії програмного
забезпечення

Негоденко О.В.

“ _____ ” _____ 2022 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
Бондар Дмитро Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для прогнозування продажів товарів мовою Python»

Керівник роботи: к.т.н., доцент Трінтіна Н.А.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «18» лютого 2022 року №_____.

2. Строк подання студентом роботи 03.06.2022 року

3. Вхідні дані до роботи:

3.1. Річні дані про продажі товарів для магазинів Walmart

3.2. Методи розробки веб-додатка

3.3. Існуючі інструменти для розробки веб-додатка

3.4. Науково-технічна література

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1. Аналіз програмного забезпечення для прогнозування продажів

4.2. Огляд засобів програмної реалізації додатку

- 4.4. Опис використаних технологій
- 4.3. Опис розробки додатку
- 4.5. Висновки
5. Перелік демонстраційного матеріалу (назва основних слайдів)
- 5.1. Титульний слайд
- 5.2. Мета, об'єкт та предмет дослідження
- 5.3. Аналоги
- 5.4. Технічне завдання
- 5.5. Програмні засоби реалізації
- 5.6. Архітектура бази даних. 5.10. Архітектура плеєра тестових сценаріїв
- 5.7. Схема прецедентів
- 5.8. Діаграма активності — створення Dataset
- 5.9. Діаграма активності — створення прогнозу (Predict)
- 5.10. Демонстрація застосування.
- 5.11. Апробація результатів дослідження
- 5.12. Висновки
6. Дата видачі завдання 11.04.22

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04.22	Виконано
2	Вимоги до системи	11.04.22	Виконано
3	Аналіз і підготовка даних для прогнозування	20.04.22	Виконано
4	Створення моделі для прогнозування	25.04-30.04	Виконано
5	Концепція та архітектура програмного забезпечення	30.04-10.05	Виконано
6	Розробка веб додатка	10.05-15.05	Виконано
7	Вступ, висновки, реферат	15.05-20.05	Виконано
8	Розробка обов'язкових демонстраційних матеріалів	20.05-25.05	Виконано
9	Попередній захист роботи	31.05.22	
10	Здача роботи	3.06.22	

Студент _____

(підпис)

(прізвище та ініціали)

Керівник роботи _____

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 62 с., 36 рис., 16 джерел.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОДАЖІВ ТОВАРІВ МОВОЮ ПРОГРАМУВАННЯ PYTHON, ВЕБ-ДОДАТОК.

Об'єктом дослідження — веб-додаток, серверні можливості для роботи з файлами. Дані про прогнозування продажів корпорації Walmart.

Завданням роботи — розробка веб-додатка, зберігання і обробка даних на сервері для створення прогнозу продажів за даними корпорації Walmart мовою програмування Python.

Предмет дослідження — розробка веб-додатка, зберігання і обробка даних на сервері для створення прогнозу продажів за даними корпорації Walmart мовою програмування Python.

Метою роботи — розробка веб-додатка для зберігання і обробки даних на сервері для спрощення прогнозування продажів за даними корпорації Walmart мовою програмування Python.

Методика дослідження: Аналіз даних про продажі в різних відділах магазинів, визначити ключові фактори, які відіграють певну роль у стимулюванні продажів, і використовують їх для розробки моделі яка допоможе прогнозування продажів з певною точністю. Обрати стек технологій із забезпеченням коректної працездатності продукту, за можливості підтримки його розробником з метою ефективного додавання нового функціоналу або виправлення виявлених недоліків.

У роботі проведено аналіз існуючих додатків, таких як Streamline, Oracle Demantra, SAP, Anaplan, SPOTIO, InsightSquared.

Проаналізувавши особливості обробки та розрахунку даних для прогнозування продажів виведено алгоритми для створення даних і прогнозу. Реалізована для веб-додатку та забезпечувати наступні функції:

- Завантаження, обробка і зберігання даних про продажі.

- Можливість створення даних для моделі прогнозування на основі сирих даних.
- Можливість створення прогнози про продажі.
- Можливість аналізувати дані.
- Можливість представлення даних у вигляді графіків та діаграм;

Загальною проблемою цих продуктів є собі вартість і в основному частина додатків дає тільки поверхневу оцінку про прогнозування. Інша частина дає детальні прогнози але немає функціонала для зберігання і обробки даних. Відповідно для Walmart розроблено індивідуальне рішення.

Практична значущість результатів: Даний продукт може бути використаний для прогнозування продажів Walmart.

Даний продукт може бути використаний для бізнесів для прогнозування продажів.

Галузь використання — прогнозування продажів.

ЗМІСТ

ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОДАЖІВ	14
1.1. ОГЛЯД ІСНУЮЧИХ ДОДАТКІВ STREAMLINE, ORACLE DEMANTRA, SAP, ANAPLAN, SPOTIO І INSIGHTSQUARED	14
РОЗДІЛ 2. ОГЛЯД ЗАСОБІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ	16
2.1. ВЕБ-ДОДАТОК	16
2.2. МОВИ ПРОГРАМУВАННЯ ДЛЯ ВЕБ-ДОДАТКІВ.....	16
2.3. ВЕБ ФРЕЙМВОРКИ НА МОВІ PYTHON	18
2.3.1. <i>Django</i>	18
2.3.2. <i>Flask</i>	19
2.3.3. <i>Pyramid</i>	19
2.4. ЗБЕРІГАННЯ ДАНИХ	20
2.4.1. <i>Реляційні і не реляційні бази даних</i>	20
2.4.2. <i>Огляд реляційних Баз даних</i>	21
2.5. ЧЕРГА ЗАВДАНЬ.....	22
2.6. ТЕСТУВАННЯ ПО	22
2.7. КЛІЄНТСЬКА ЧАСТИНА ДОДАТКА.....	22
2.7.1. <i>UI-фреймворки</i>	23
2.7.2. <i>Bootstrap</i>	24
РОЗДІЛ 3. ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	25
3.1. ПЛАТФОРМИ, ФРЕЙМОКРИ ЯКІ ВИКОРИСТОВУЮТЬСЯ ДЛЯ ЯДРА ДОДАТКУ	25
3.1.1. <i>Python</i>	25
3.1.2. <i>Flask і Jinja</i>	25
3.1.3. <i>Що означає «мікро»?</i>	26
3.1.4. <i>ORM та SQLAlchemy</i>	27
3.1.5. <i>PostgreSQL</i>	29
3.2. ЧЕРГА ЗАВДАНЬ CELERY	30
3.3. ТЕСТУВАННЯ PYTEST	31
3.4. ОПИС ЗАСОБІВ ДЛЯ РОЗРОБКИ ВЕБ-СТОРІНКИ	31
3.4.1. <i>Bootstrap</i>	31
3.4.2. <i>Bootswatch</i>	31
3.4.3. <i>Charts – Google developer</i>	32
3.4.4. <i>Flask-Admin</i>	33
3.4.5. <i>Flask-BasicAuth</i>	33

3.5. ІНСТРУМЕНТИ ДЛЯ ОБРОБКИ ДАНИХ. PANDAS, STATSMODELS, SCIPY, MATPLOTLIB, SEABORN, NUMPY.....	33
---	----

РОЗДІЛ 4. ОПИС АНАЛІЗУ ДАНИХ І СТВОРЕННЯ ПРОГНОЗУ ДЛЯ WALMART.....36

4.1. АНАЛІЗ ДАНИХ ПРОДАЖІВ WALMART.....	36
4.1.1. <i>Набір Даних Walmart</i>	36
4.1.2. <i>Аналіз даних для моделі прогнозування</i>	39

РОЗДІЛ 5. ОПИС РОЗВРОБКИ ДОДАТКУ42

5.1. ОПИС РОЗРОБКИ ВЕБ-ДОДАТКА.....	42
5.2. ОПИС РОЗРОБКИ ТЕСТІВ ДЛЯ ПЕРЕВІРКИ КОРЕКТНОЇ РОБОТИ ДОДАТКА.	43
5.3. ОПИС РЕАЛІЗАЦІЇ СХОВИЩА ДАНИХ POSTGRESQL.....	44
5.4. ОПИС РОЗРОБКИ ВЕБ-ФРЕЙМВОРКА FLASK.....	46
5.5. НАЛАШТУВАННЯ ЧЕРГИ ЗАВДАНЬ CELERY.....	49
5.6. БЕЗПЕКА І ДОСТУПИ.....	50
5.7. ОПИС РЕАЛІЗАЦІЯ ЗАВДАННЯ ОБРОБКИ ДАНИХ ПРО ПРОДАЖІ DATASET.	51
5.8. ОПИС РЕАЛІЗАЦІЯ ЗАВДАННЯ ПРОГНОЗУВАННЯПРОДАЖІВ (PREDICT)	53
5.9. ОПИС ПАНЕЛІ КЕРУВАННЯ В РЕЖИМІ АНАЛІТИКИ.....	54

ВИСНОВКИ.....56

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ORM – Object-Relational Mapping

SDK – Software Development Kit

API – Application Programming Interface

CSV – Comma-separated values

ШІ – Штучний Інтелект

SQL – Structured query language

JS – Java Script

UI –User Interface

QA – Quality Assurance

ПЗ – Програмне забезпечення

СУБД – Система Управління Базами Даних

HTTP – HyperText Transfer Protocol

Basic-Auth – Basic access authentication

ВСТУП

Для бізнесів як показує практика в динамічному світі де що дня змінюється ситуація на прибуток. Щоб аналізувати продажі і робити прогнози потрібні зручні інструменти.

Бізнеси які займаються торгівлею товарів стрімко розвиваються у світі займають значну частину економіки. Попит бізнесів залежать від поставок, ситуації на ринку, закупівлі товарів і багато інших факторів. В бізнесі багато залежить від різних факторів і їх потрібно враховувати, щоб обрати гарну модель і підлаштуватись до умов ринку.

Великі компанії мають змогу для залучати фахівців для аналітики, аналізу тощо. Відповідно фахівці роблять розрахунки, використовують всі інструменти аналізують ринку тощо.

Часто для розрахунку прогнозу продажів фахівці використовують інструменти ПЗ як Microsoft Excel, веб-додатки, хмарні технології тощо. Це дозволяє дати оцінку своїй ситуації на сьогодні та оцінити на майбутнє.

Загальною проблемою цих продуктів є собі вартість і в основному частина додатків дає тільки поверхневу оцінку про прогнозування. Інша частина дає детальні прогнози, але немає функціонала для зберігання та обробки даних.

Тому я обрав розробити додаток для обробки аналізу і прогнозування продажів товарів на основі історичних даних, наприклад про магазини, їх відділи, категорії товарів, постачальників та брендах, що дозволяє робити вичерпний аналіз та приймати обґрунтовані рішення для прикладу бізнесу було обрано дані про корпорацію Walmart.

Завдяки цьому ми матимем:

- Збільшення швидкості прийняття рішень.
- Прозора аналітика для бізнес-процесів.
- Перетворення існуючих даних у бізнес-ідеї.

- Аналітик матиме змогу більше зосереджений на аналізі і прогнозі, опис товарів, продажів, тенденцію для бізнесів.
- Бізнес зможе уникнути перебоїв в продажах тощо.

Зустрічі з огляду конвеєра також стають легкими, оскільки ви повністю впевнені в даних про продажі. За рахунок чього спеціаліст швидко, може, визначити, які угоди слід відкласти, а які призупинити. Крім того, ви також знаєте, яких представників тренувати, оскільки можете зрозуміти причини їхніх показників продажів.

У дипломі розглянуто розробка додатка для прогнозування продажів який я зробив для Walmart.

Walmart є одним з найбільших роздрібних продавців у світі, і це дуже важливо для них мають точні прогнози щодо їх продажу в різних відділах.

Ступінь вивчення проблеми: дуже важливо для **Walmart** мають точні прогнози щодо їх продажу в різних відділах. Оскільки факторів може бути багато що може вплинути на продажі для кожного відділу.

Метою роботи — Розробка веб-додатка для зберігання і обробки даних на сервері для спрощення прогнозування продажів за даними корпорації Walmart мовою програмування Python.

Об'єкт дослідження – Веб-додаток, серверні можливості для роботи з файлами. Дані про прогнозування продажів корпорації Walmart.

Предмет дослідження – Розробка веб-додатка, зберігання і обробка даних на сервері для створення прогнозу продажів за даними корпорації Walmart мовою програмування Python.

Завданням роботи є Розробка додатка для обробки, аналізу і прогнозування продажів на основі історичних даних.

Методика дослідження: Аналіз історичних даних про продажі в різних відділах магазинів, визначити ключові фактори, які відіграють певну роль у стимулюванні продажів, і використовують їх для розробки моделі яка допоможе прогнозування продажів з певною точністю. Обрати стек

технологій із забезпеченням коректної працездатності продукту, за можливості підтримки його розробником з метою ефективного додавання нового функціоналу або виправлення виявлених недоліків.

Враховуючи вимоги до специфіки програмного продукту, найкращим рішенням є веб-додаток, що має клієнт-серверну архітектуру, де клієнтська сторона відповідає за роботу та логіку програмного продукту, а серверна – за обчислення. Такий тип технології дає можливість використання складних обчислень, не призводячи до необхідності введення великої кількості даних локально.

Потрібно мати процес для прогнозування продажів і обробки даних у фоновому режимі на сервері, а клієнт отримує тільки дані для аналізу.

Відповідно ми матимем сторінки для завантаження, обробки і зберігання даних про продажі.

Сторінки для створення даних для моделі прогнозування на основі сирих даних. Також сторінки створення прогнози про продажі.

Сторінки для аналізу даних з можливістю представлення даних у вигляді графіків та діаграм тощо.

Наукова новизна роботи: Таким чином, наукова новизна полягає в створенні веб додатка з залученням серверних можливостей.

Практична значущість результатів: Даний продукт може бути використаний для бізнесів з продажу товарів.

РОЗДІЛ 1. АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ ПРОДАЖІВ

В сучасних умовах якісне управління економічними системами можливе тільки на основі використання ефективної моделі прогнозування, що забезпечить гарний прибуток.

На сьогодні, на ринку існує багато різних автоматизованих систем для прогнозування товарів. Це можуть бути платформи, або комбіновані системи з командою аналітиків, або з одним аналітиком тощо.

1.1. Огляд існуючих додатків **Streamline, Oracle Demantra, SAP, Anaplan, SPOTIO i InsightSquared**

Streamline є провідною у світі програмною платформою для прогнозування продажів для малих, середніх і великих підприємств. Ця програмна платформа на основі штучного інтелекту допомагає виробникам, роздрібним торговцям і дистриб'юторам.

Oracle Demantra — дозволяє організації точно прогнозувати та планувати попит. Також попередньо встановлені ієрархії, робочі таблиці та робочі процеси для встановлення базового прогнозу на основі комбінації кількісних або якісних даних. Demand Management пропонує необмежену кількість ієрархій для багатовимірного аналізу з можливостями динамічної звітності.

SAP Інтегроване — ПЗ для бізнес-планування пропонує рішення для планування, з інструменти фінансового прогнозування, бюджетування, планування сценаріїв, консолідації та інструментів співпраці.






Переваги використання SAP проста у використанні, в основному налаштовується в Excel.

Anaplan — ПЗ для прогнозування продажів від Anaplan дозволяє генерувати керовані даними прогнози доходу в режимі реального часу, що покращує прийняття рішень у вашій організації.

Ви можете спрогнозувати продажі за географічним розташуванням, лінійкою продуктів або обліковим записом з будь-яким рівнем деталізації – наприклад, за містом чи штатом, за специфічними артикулами продуктів або за певним набором облікових записів.

InsightSquared — ПЗ для прогнозування продажів на основі штучного інтелекту InsightSquared спрощує подання прогнозів у вашій команді, надаючи вам більше часу, щоб зосередитися на прийнятті рішень. Крім того, оновлення на основі штучного інтелекту дозволяють легко згортати прогнози для всієї команди та гарантувати, що всі на шляху до досягнення цілі.

Таблиця 1.

 Streamline	 SAP ERP	 Anaplan	 INSIGHTSQUARED	 ORACLE Demantra
Тільки для windows.	налаштовується в Excel	Націлена для роботи на Американському у ринку	прогнозування продажів на основі штучного інтелекту	Деталізувач прогноза. Складний інтерфейс.

На ринку для бізнесу продажів існує ПЗ базуючись на демо версіях цих ПЗ. Можна сказати що перші три додатки для бізнесу основному вони дають поверхневу оцінку аналізу і прогнозу оскільки вони націлені більше на роботи з кроками поставок тощо. SPOTION, InsightSquared і Anaplan вони чудово описують і підходять для точних прогнозування продажів і гарно описують підхід прогнозування. Також гарний підхід для прогнозування продажів використовувати **III** як реалізовано в InsightSquared.

Оглянувши ПЗ для прогнозів продажів я розроблю свій додаток з моделювання формули для Walmart прогнозів відповідно в нашій роботі це буде продемонстровано.

РОЗДІЛ 2. ОГЛЯД ЗАСОБІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ

Для розробки додатка на тему прогнозування продажів товарів для малого та середнього **найкращий спосіб це веб-додаток**. Головне завдання це обробка, зберігання і робота з великим об'ємом даних для цього найкраще підходить віддалений сервер.

2.1. Веб-додаток

Це комп'ютерна програма, яка виконує будь-яку функцію, використовуючи веб-браузер.

Для розробки програми веб-додатки використовують комбінацію сценарію на стороні сервера (ASP, Python, PHP тощо) та сценарію на стороні клієнта (HTML, Javascript, CSS). Сценарій на стороні клієнта займається поданням інформації, тоді як сценарій на стороні сервера займається всіма важкими речами, такими як зберігання та отримання інформації.

2.2. Мови програмування для веб-додатків.

З рейтинг мов програмування 2022 з сайту **dou.ua**. В Україні за галузю застосування, то ситуація у бекенді програмуванні використовують такі мови програмування: Java популярна, C# на третьому місці PHP

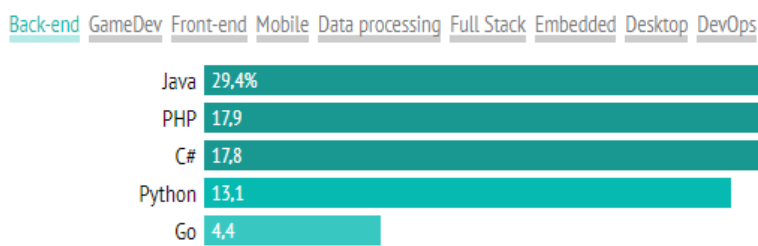


Рисунок 2.1 — Рейтинг мов програмування 2022 в галузі **Back-end** з сайту **Dou.ua**.

Back-end GameDev Front-end Mobile Data processing Full Stack Embedded Desktop DevOps



Рисунок 2.2 — Рейтинг мов програмування 2022 в галузі **Data processing** з сайту **DoU.ua**.

Відповідно для розробки серверної частини і ядра додатка **я обрав популярну мову програмування Python.**

Головні переваги Python — велика спільнота програмістів і швидкість розробки. Завдяки лаконічній мові програмування і бібліотекам це дає можливість проекту на Python бути дешевими та швидкими в розробці.

Чистий синтаксис і лаконічність. Лаконічний, читабельний код легше підтримувати та виправляти.

Бібліотеки в Python

Стандартна бібліотека Пітона спрощує виконання багатьох рутинних завдань програміста і дає доступ системному функціоналу написаних модулів на мові C.

Python не є швидкою мовою для розрахунків великою кількістю даних. Python популярна мова програмування в Data Sense завдяки тому що Python має доступ до модулів написаних на C. Python має велику кількість математичних бібліотек написаних на c такі як: numpy, Scikit-learn, pandas тощо.

В проект можна підключати зовнішні бібліотеки. Вони покривають велику кількість сфери розробки – від корпоративних систем, машинного навчання, e-commerce і візуалізації статистики.

Python підтримує асинхронний код. Модель асинхронна підвищує продуктивність веб-додатків. Виходячи з досвіду Python навіть підходить для

високонавантажених проектів і сайтів, де потрібно швидкість запитів обробки складних.

2.3. Веб фреймворки на мові Python

Головною метою веб фреймворків це дати готовий функціонал і інструменти для розробки веб додатка. Протоколи обміну даних типу HTTP, обмін даних, оформлення даних тощо.

В Python для розробки веб-додатків має багато різних фреймворків: Flask, Django, Pyramid. Найпопулярніші згідно з рейтингами на Python це є Flask і Django, Pyramid.

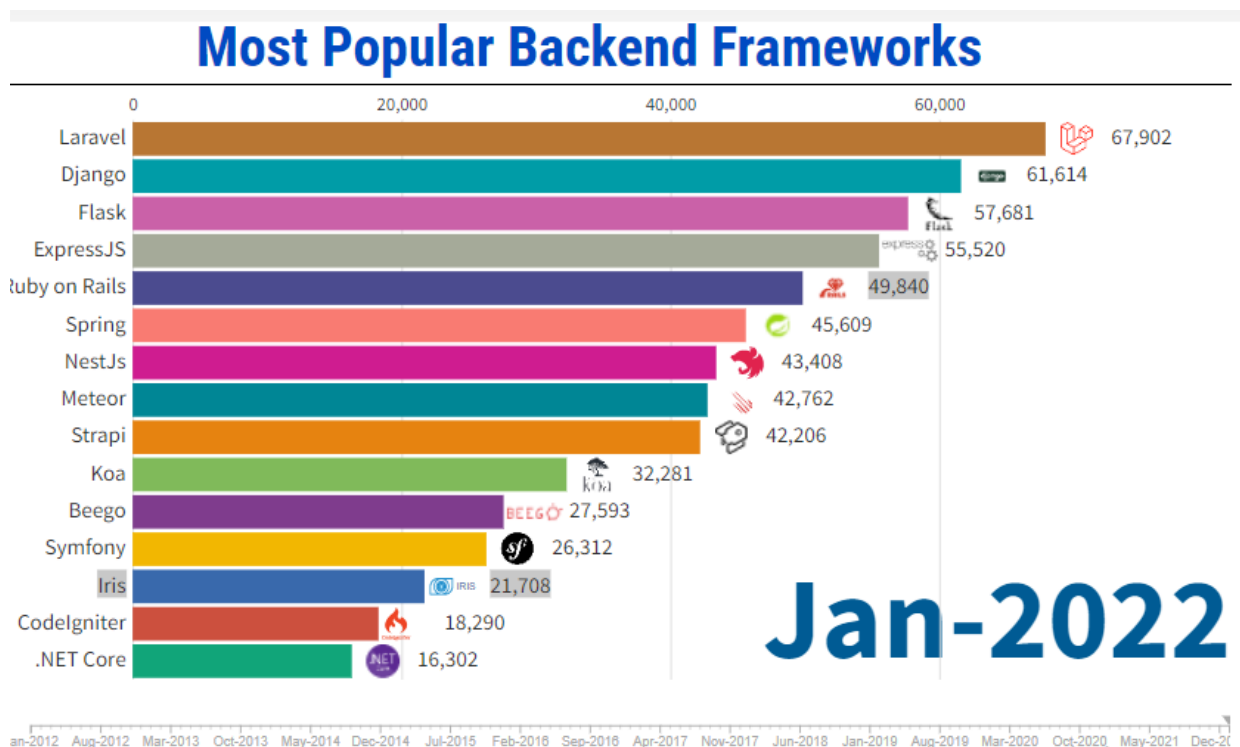


Рисунок 2.3 — Рейтинг популярності веб-фреймворків на 2022 рік згідно з сайтом statisticsanddata.org

2.3.1. Django

Це фреймворк, який використовується більшістю розробників Python. Це повна композиція веб-додатків, яка пропонує вам безліч функцій; Він

включає маршрутизацію URL-адрес, ORM, механізм шаблонів, аутентифікацію, міграцію схеми бази даних та багато іншого.

Крім того, Django є надзвичайно ефективним фреймворком Python. Його універсальність робить його більш відомим серед користувачів. Django володіє приблизно 14,2% частки світового бізнесу і займає десяте місце за поширеністю фреймворк розробки Python.

2.3.2. Flask

Це ще один провідний фреймворк, коли справа доходить до веб-розробки. Цей фреймворк Python пропонує широкий спектр розширень. Більшість користувацького агентства веб-розробки Python використовує цю структуру, щоб миттєво допомогти спрощеному API для веб-сервера.




Вбудоване налагодження є ще однією чудовою функцією, яку ви отримуєте разом із Flask. Деякі з неминучих функцій, які ви отримуєте в цьому фреймворку Python, — це ефективна обробка запитів HTTP, розробка вбудованого сервера, допомога при модульному тестуванні. Flask навіть відомий миттєвим розгортанням; лише через кілька секунд він готовий до запуску. Після Django Flask є 11-м за поширеністю фреймворком.

2.3.3. Pyramid

Найпопулярніший фреймворк веб-пакета Python завжди неповний без Pyramid. Pyramid пропонує широкий спектр функцій для створення складних або простих додатків; Pyramid пропонує вам широкий спектр підтримки, щоб збільшити свій потенціал. Є навіть багато програмістів Python, які люблять Pyramid за її розширюваність.

Деякі з унікальних функцій, які ви отримуєте в Pyramid, — це декоратор функцій, гнучкі авторизації, пакет з одним файлом, засоби візуалізації та багато іншого. Як фреймворк Python, Pyramid також дає розробникам можливість вибирати будь-яку мову шаблонів, бібліотеки генерації та рівні бази даних.

Таблиця 2.

		 Flask	 Pyramid™
Адмін панель	+	-	-
Веб-Фреймворк	+	+	+
База даних	+	-	-
Безпека	+	-	-
Шаблонізатор	+	+	+
Гнучкість	-	+	+
Продуктивність	+	+	+

Для нашого додатка не потрібна сильний двигун оскільки не матиме великий функціонал. Головною метою є обробка і зберігання даних для цього важливо два головних пункти це продуктивність і гнучкість. **Тому для розробки веб-додатка був обраний Flask.**

2.4. Зберігання даних

У веб-додатках зазвичай потрібна база даних, яка є організованою колекцією даних. Бази даних використовуються для зберігання та підтримки постійних даних, які можна ефективно отримувати та керувати ними. **реляційна і не реляційна.**

2.4.1. Реляційні та не реляційні бази даних

Існують два види база даних **Реляційна і не реляційна.**

Головна перевага реляційної це структура і зберігання даних. З таким підходом бази даних зазвичай швидкі і надійніші. А не реляційна — вони не мають таблиці дані зберігаються в об'єктах які можна модифікувати. Тому база є гнучкішою і динамічною.

Відповідно нам головне це обробка і розрахунки даних тому для додатка обрано реляційна база даних.

2.4.2. Огляд реляційних Баз даних.

Сьогодні популярними базами даних є: MySQL, PostgreSQL, SQLite, MongoDB та ін.

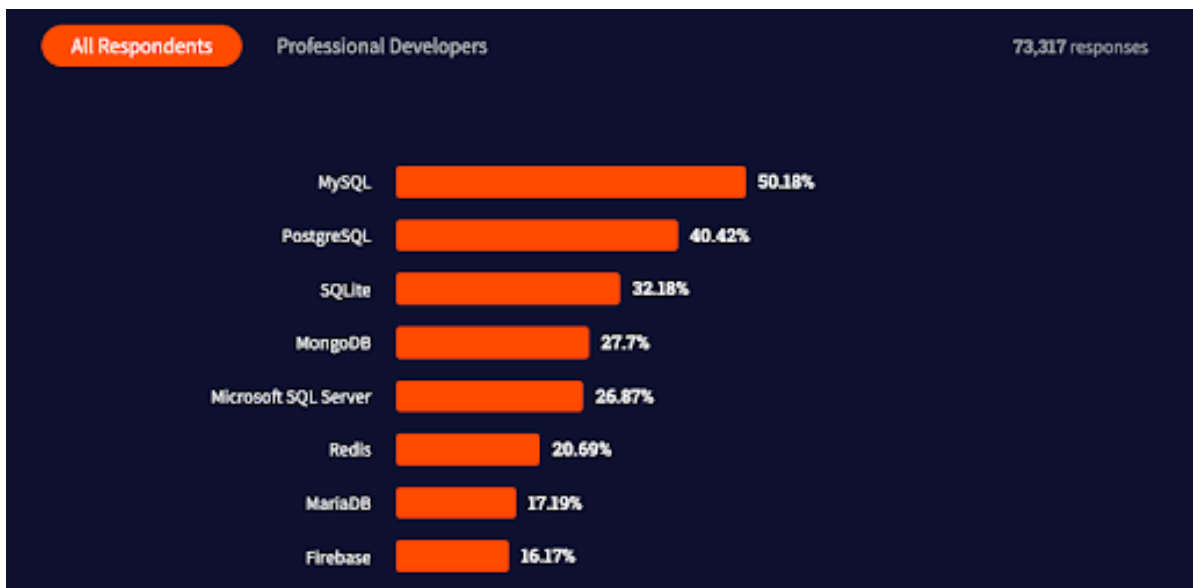


Рисунок 2.4 — Опитування на Stack Overflow “найкращі бази даних 2022 р.”

Розглянемо першу трійку. SQLite одразу виключаємо, оскільки її не використовують для веб-додатків. MongoDB не реляційна тому для нашого додатку також не підходить тому **будемо обирати з MySQL і PostgreSQL**.

Встановлення Postgres за замовчуванням зазвичай працює краще, ніж MySQL за замовчуванням (але ви можете налаштувати MySQL для компенсації). MySQL має деякі дивні параметри за замовчуванням (наприклад, для кодування символів і зіставлення).

Postgres дуже розширюваний. Він підтримує ряд розширених типів даних, недоступних в MySQL (геометричні/ГІС, типи мережеских адрес, JSONB, який можна індексувати, власний UUID, часові позначки з

урахуванням часового поясу). Якщо цього недостатньо, ви також можете додати власні типи даних, оператори та типи індексів.

Так шляхом індексації ми матимем зручніший і спосіб оперуванням даних для нашого додатка, тому для реалізації бази даних було обрано PostgreSQL

2.5. Черга завдань.

Для роботи з завданнями які можуть займати великий час потрібна черга яка буде виконувати це завдання на сервері в фоновому режимі який не пов'язаний з веб-клієнтом. Для python це найпопулярнішою є celery.

2.6. Тестування ПО

Для тестування роботи можна використовувати два основних підходи це мануальне тестування і unit тести. Мануальне тестування це ручна перевірка функціоналф на програми. Unit test — це метод тестування коду, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яка може бути протестованою.

В Python для роботи з Unit test користуються є дві популярні бібліотеки **unittests** і **pytest**. Судячи з можливостей pytest це нова бібліотека і вона зараз набирає велику популярність вона дає зручність і гнучкість для тестування коду та інших можливості яких нема в стандартній бібліотеці. Для проекту відповідно обрано pytest.

2.7. Клієнтська частина додатка

На сьогодні дуже рідко використовують чисту верстку на HTML, CSS Javascript. Дуже велика частка веб-ресурсів використовують **UI-фреймворки.**

2.7.1. UI-фреймворки

Фреймворк створений для полегшення роботи веб-розробника та веб-дизайнера, пришвидшення розробки та попередження максимальної кількості помилок.

Більшість сучасних фреймворків за стандартом де-факто, містять так звані CSS файли-нормалізатори, призначення яких "онулення"(скидання) CSS до стандартного значення. Це дозволяє отримувати майже однаковий вигляд(дизайн) веб-сторінки, сайту в різних браузерах та ОС.

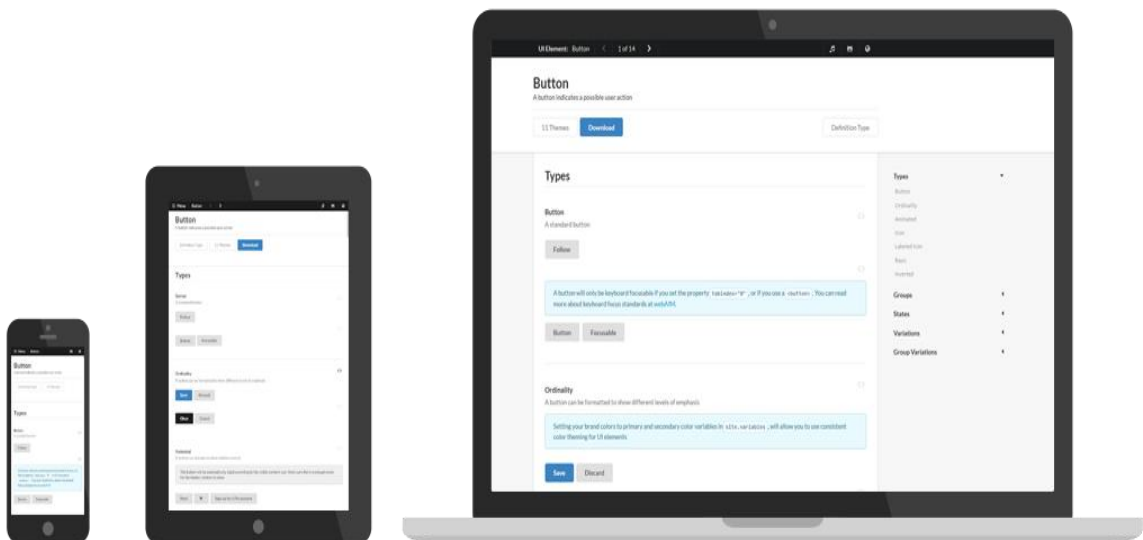


Рисунок 2.5 — Вигляд веб-сторінки на різних дисплеях за допомогою UI фреймворку(Bootstrap)

На сьогодні популярні фреймворки: Bootstrap, Semantic UI, UIKit та ще багато інших не так популярні.

Таблиця 3:

	Bootstrap	UIKit	Semantic UI
Елементи адаптовані під різні типи дисплеїв	мобільний, ноутбук, монітор, планшети	мобільний, ноутбук, монітор, часи, планшети	мобільний, ноутбук, монітор
Легке налаштування	Так	Так	Так
Таблиці	Так	Так	Так
Структура	Сітка 12 стовбців	Адаптується на основі grid атрибутів	Адаптується на основі grid атрибутів

Порівнявши фреймворки можна сказати що на сьогодні всі ці три фреймворки є популярні оскільки всі вони мають схожий функціонал. Відповідно можна обрати який подобається. Для нашого випадку **я обрав Bootstrap.**

2.7.2. Bootstrap

Це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript.

Оглянув засоби програмної реалізації веб-додатку можна зробити висновок що перелічені технології в комплексі з Python, це гарне рішення для розробки нашого додатка.

РОЗДІЛ 3. ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

3.1. Платформи, фреймворки які використовуються для ядра додатка.

3.1.1. Python

Це інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом.

Бібліотеки в Python

Пакетів Python (PyPI) містить тисячі модулів сторонніх розробників для Python. Як стандартна бібліотека Python, так і модулі, надані спільнотою, надають безмежні можливості.

- Розробка веб та Інтернет;
- Доступ до бази даних;
- Графічні інтерфейси;
- Наукові й математика;
- Освіта;
- Мережне програмування;
- Розробка ігор;

Стандартна бібліотека Python підтримує багато Інтернет-протоколів: HTML, JSON, XML, IMAP, FTP і багато інших протоколів.

Python підтримує асинхронний код з версії 3.5. Асинхронний додаток продовжує працювати і обробляти запити під час очікування відповіді від сервера.

3.1.2. Flask і Jinja

Це веб-фреймворк, це модуль Python, який дозволяє легко розробляти веб-додатки. Він має невелике ядро, яке легко розширити: це мікрофреймворк, який не містить ORM (Об'єктний реляційний менеджер) або подібні функції.

Він має багато цікавих функцій, таких як маршрутизація URL-адрес, механізм шаблонів. Це структура веб-додатків WSGI.

Flask для роботи з сторінками зазвичай Flask використовує **Jinja**. **Jinja** — це швидкий, виразний, розширюваний механізм шаблонів. Спеціальні заповнювачі в шаблоні дозволяють писати код, схожий на синтаксис Python. Потім шаблону передаються дані для відтворення остаточного документа.

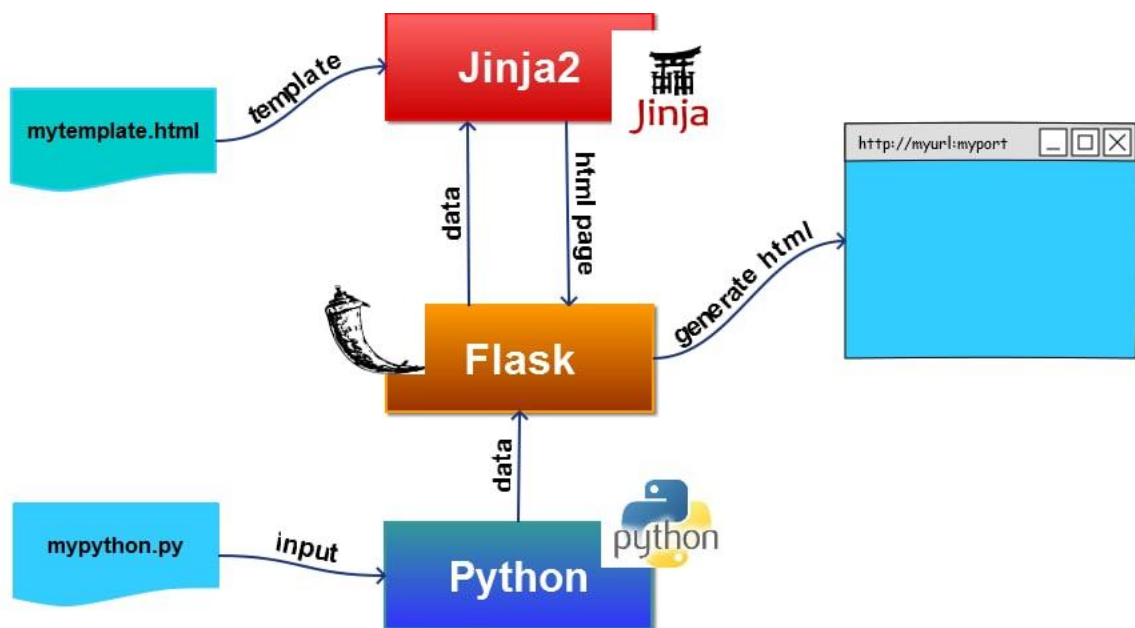


Рисунок 3.1 — схема як взаємодіють Python, Flask і Jinja2.

3.1.3. Що означає «мікро»?

«Мікро» не означає, що вся ваша веб-програма має вміститись в один файл Python (хоча це, безумовно, може), а також не означає, що Flask не має функціональних можливостей. «Мікро» в мікрофреймворку означає, що Flask прагне зробити ядро простим, але розширюваним. Flask не буде приймати багато рішень за вас, наприклад, яку базу даних використовувати. Ті рішення, які він приймає, наприклад, який механізм шаблонів використовувати, легко змінити. Все інше залежить від вас, так що Flask може бути всім, що вам потрібно, і тим, що вам не потрібно.

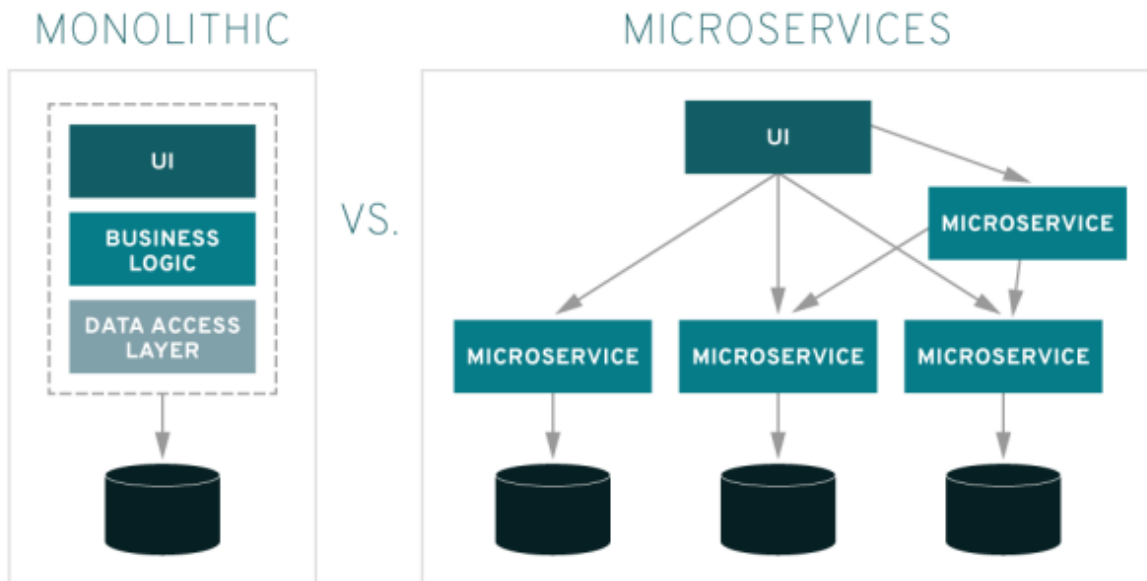


Рисунок 3.2 — Мікросервісна архітектура.

За замовчуванням Flask не включає рівень абстракції бази даних, перевірку форми чи щось інше, де вже існують різні бібліотеки, які можуть це обробляти. Натомість Flask підтримує розширення для додавання такої функціональності до вашої програми, як якщо б вона була реалізована в самому Flask. Численні розширення забезпечують інтеграцію бази даних, перевірку форм, обробку завантаження, різні технології відкритої аутентифікації тощо. Колба може бути «мікро», але вона готова до використання у виробництві для різних потреб.

3.1.4. ORM та SQLAlchemy

Об'єктно-реляційне відображення (ORM) — це техніка програмування, в якій дескриптор метаданих використовується для підключення об'єктного коду до реляційної бази даних. Об'єктний код пишеться на мовах об'єктно-орієнтованого програмування (ООП), таких як Java або C#. ORM перетворює дані між системами типів, які не можуть співіснувати в реляційних базах даних і мовах ООП.

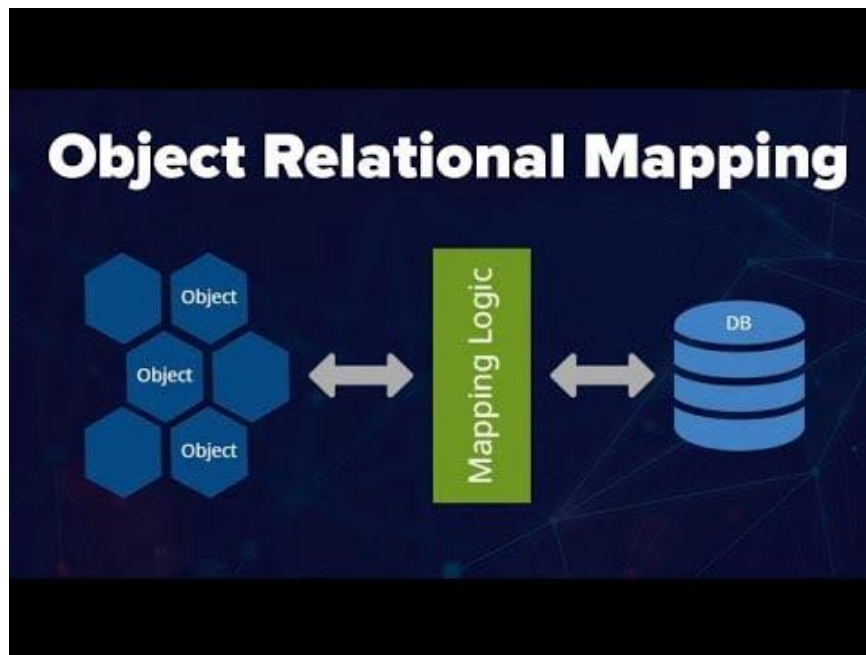


Рисунок 3.3 — Схема роботи ORM.

ORM усуває невідповідність об'єктного коду та реляційної бази даних. Вибираючи найкраще програмне рішення, розробники повинні повністю розуміти середовище та вимоги до дизайну.

Переваги ORM також включають:

- Спрощена розробка, автоматизує перетворення об'єкта в таблицю в об'єкт, що дає до зниження витрат на розробку та обслуговування.
- Менше коду в порівнянні з вбудованим SQL.
- Прозоре кешування об'єктів на рівні програми, що покращує продуктивність системи.
- Оптимізоване рішення, швидшим і простіше в обслуговуванні.

SQLAlchemy — Це набір інструментів Python SQL та Object Relational Mapper, який надає розробникам додатків повну потужність та гнучкість SQL.

SQLAlchemy найбільш відомий своїм об'єктно-реляційним картографом (ORM), додатковим компонентом, який надає шаблон відображення даних, де класи можуть бути зіставлені з базою даних різними способами, що дозволяє об'єктній моделі та схемі бази даних розвиватися в чітко відокремлений шлях із самого початку.

```

from sqlalchemy import Column
from sqlalchemy import ForeignKey
from sqlalchemy import Integer
from sqlalchemy import String
from sqlalchemy.orm import declarative_base
from sqlalchemy.orm import relationship

Base = declarative_base()

class User(Base):
    __tablename__ = "user_account"

    id = Column(Integer, primary_key=True)
    name = Column(String(30))
    fullname = Column(String)

    from sqlalchemy.orm import Session

    with Session(engine) as session:

        spongebob = User(
            name="spongebob",
            fullname="Spongebob Squarepants",
            addresses=[Address(email_address="spongebob@sqlalchemy.org")],
        )
        sandy = User(
            name="sandy",
            fullname="Sandy Cheeks",
            addresses=[
                Address(email_address="sandy@sqlalchemy.org"),
                Address(email_address="sandy@squirrelpower.org"),
            ],
        )
        patrick = User(name="patrick", fullname="Patrick Star")

        session.add_all([spongebob, sandy, patrick])

        session.commit()

```

Рисунок 3.4 — Приклад коду моделі і записи на Python (SQLAlchemy)

```

BEGIN (implicit)
INSERT INTO user_account (name, fullname) VALUES (?, ?)
[...] ('spongebob', 'Spongebob Squarepants')
INSERT INTO user_account (name, fullname) VALUES (?, ?)
[...] ('sandy', 'Sandy Cheeks')
INSERT INTO user_account (name, fullname) VALUES (?, ?)
[...] ('patrick', 'Patrick Star')
INSERT INTO address (email_address, user_id) VALUES (?, ?)
[...] ('spongebob@sqlalchemy.org', 1)
INSERT INTO address (email_address, user_id) VALUES (?, ?)
[...] ('sandy@sqlalchemy.org', 2)
INSERT INTO address (email_address, user_id) VALUES (?, ?)
[...] ('sandy@squirrelpower.org', 2)
COMMIT

```

Рисунок 3.5 — Приклад запису оригінальною мовою SQL

3.1.5. PostgreSQL

Це потужна система об'єктно-реляційних баз даних з відкритим вихідним кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями, які безпечно зберігають і масштабують найскладніші робочі навантаження даних. Витоки PostgreSQL сягають 1986 року в рамках

проекту POSTGRES в Каліфорнійському університеті в Берклі і має понад 30 років активної розробки на базовій платформі.

PostgreSQL намагається відповідати стандарту SQL, якщо така відповідність не суперечить традиційним функціям або може призвести до неправильних архітектурних рішень.

PostgreSQL постачається з багатьма функціями, які допомагають розробникам створювати програми, адміністраторам захищати цілісність даних і створювати відмовостійкі середовища, а також допомагають вам керувати даними незалежно від того, наскільки великий чи малий набір даних. Окрім того, що PostgreSQL є безкоштовним і відкритим вихідним кодом, він дуже розширюваний. Наприклад, ви можете визначати власні типи даних, створювати власні функції, навіть писати код з різних мов програмування без перекомпіляції бази даних!

3.2. Черга завдань Celery

Celery є асинхронною чергою завдань із відкритим кодом або чергою завдань, яка базується на розподіленому повідомленні. Хоча він підтримує планування, його увагу приділяється операціям у режимі реального часу.



Рисунок 3.6 — Приклад запису до черги завдань Celery

3.3. Тестування Pytest

Це платформа для тестування ПЗ, на мові програмування Python. Його можна використовувати для написання різних типів тестів ПЗ, включаючи модульні тести, інтеграційні тести, наскрізні тести та функціональні тести.

3.4. Опис засобів для розробки веб-сторінки

3.4.1. Bootstrap

Це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript.

3.4.2. Bootswatch

Це колекція стилів яка має уже всі готові компоненти для верстки сайту написані фреймворком *Bootstrap*. В колекції є 25 тем з різними ми стилями, кольорами, оформленням. за замовченням ці стилі інтегровані в *Flask-admin*.

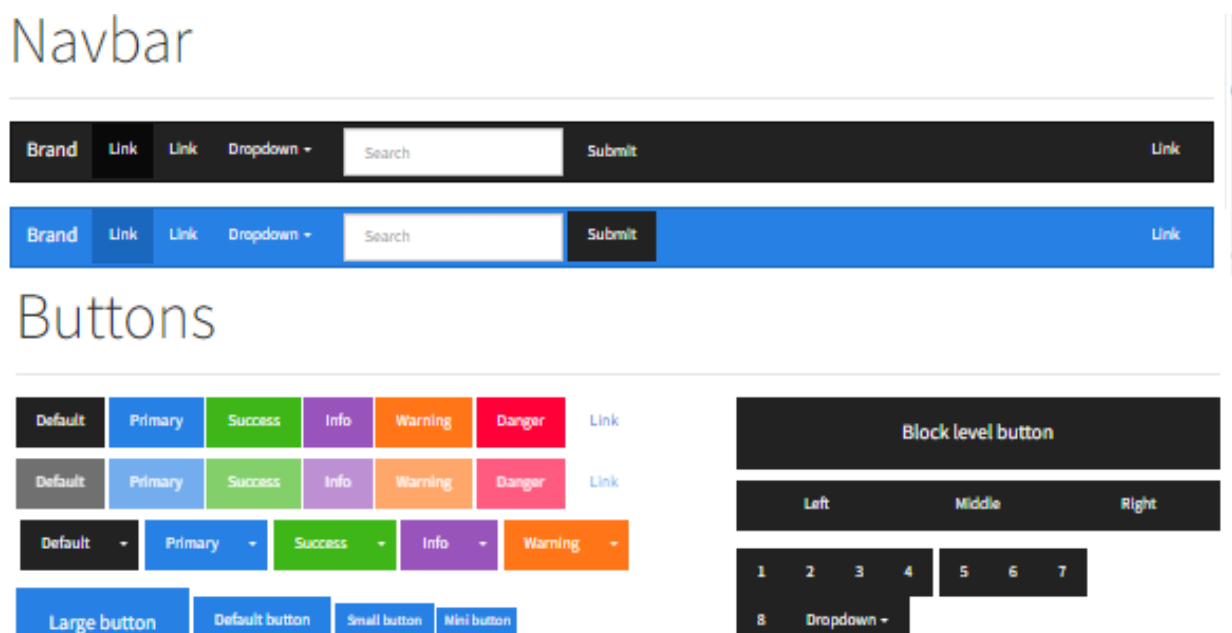


Рисунок 3.7 — Приклад елементів теми COSMO колекції Bootswatch

3.4.3. Charts – Google developer

Це інтерактивна веб-служба, яка створює графічні діаграми з наданої користувачем інформації. Користувач надає дані та специфікації форматування, виражені в JavaScript, вбудованому у веб-сторінку;

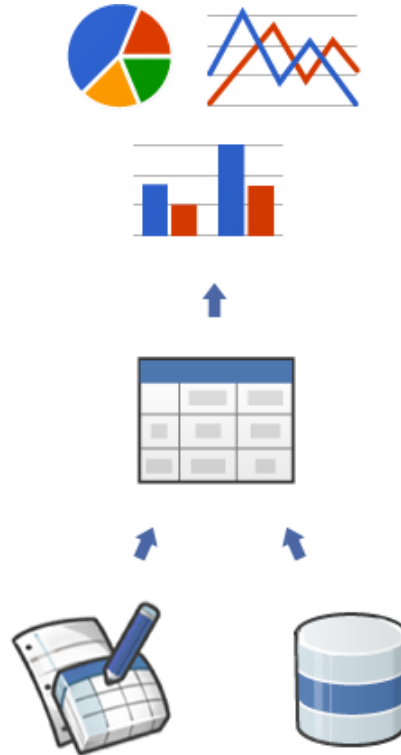


Рисунок 3.8 — Принцип роботи **Google Charts**

Дані беруться з бази даних на основі даних генерується таблиця, після таблиці бібліотека відображає готовий графік.

Створений графік можна динамічно змінити значення, працювати додавати свої дані тощо.

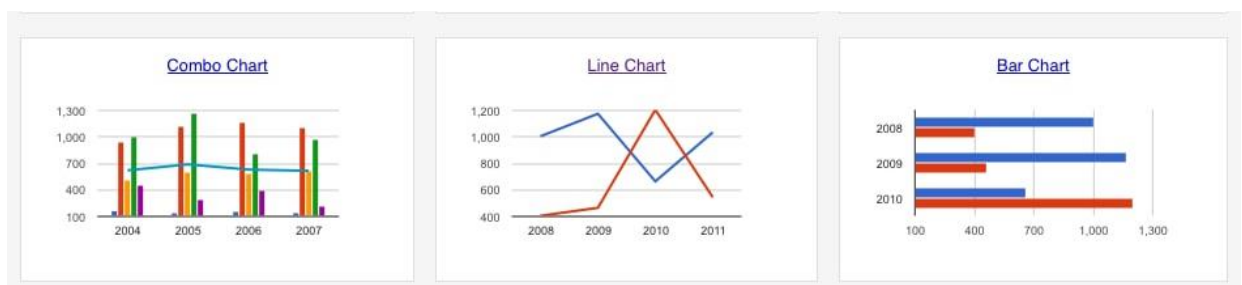


Рисунок 3.9 — Приклад графіків **Google Charts**

3.4.4. Flask-Admin

Це просте у використанні розширення *Flask* з батареїками, яке дозволяє додавати інтерфейси адміністратора до програм Flask. Він натхненний пакетом *django-admin*, але реалізований таким чином, що розробник має повний контроль над зовнішнім виглядом, відчуттям і функціональністю отриманої програми.

Готовий Flask-Admin чудово грає з різними **ORM**, у тому числі: **SQLAlchemy**, **MongoEngine**, **pymongo** і **Peewee**.

3.4.5. Flask-BasicAuth

це розширення Flask, яке забезпечує спосіб захистити певні сторінки або всю вашу програму за допомогою базової аутентифікації доступу HTTP.

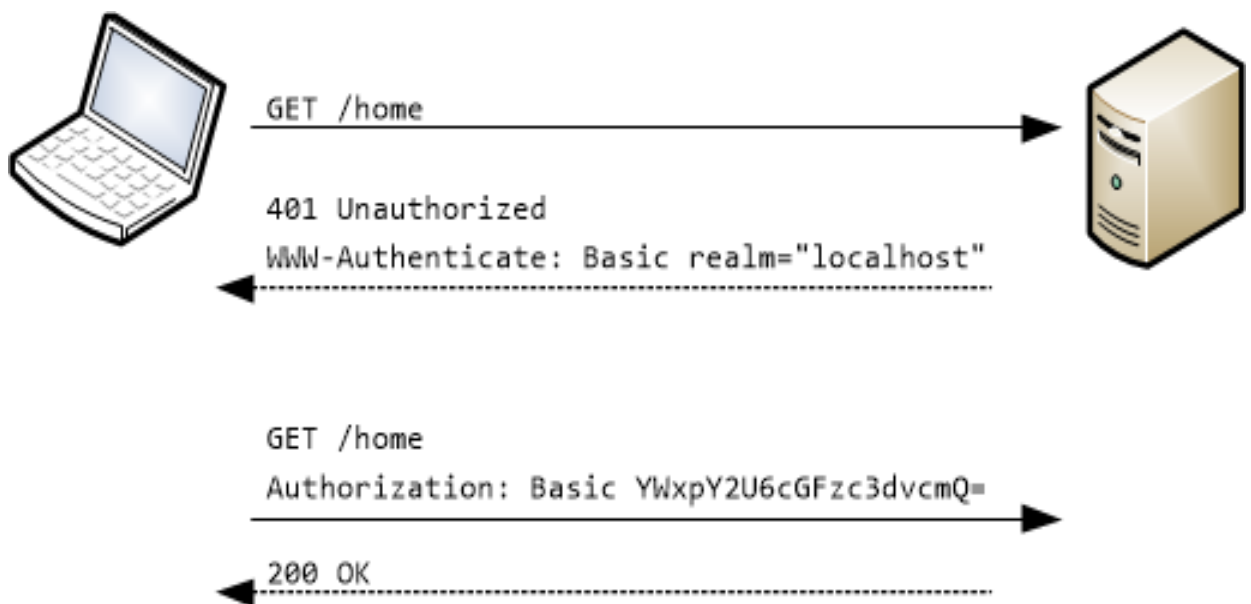


Рисунок 3.10 — Схема роботи Basic-Auth.

3.5. Інструменти для обробки даних. Pandas, Statsmodels, SciPy, Matplotlib, Seaborn, numpy.

Pandas — Це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та маніпулювання даними з відкритим вихідним кодом,

Основні моменти бібліотеки

Швидкий та ефективний об'єкт DataFrame для маніпулювання даними з інтегрованим індексуванням;

Інструменти для читання та запису даних між структурами даних у пам'яті та різними форматами: CSV та текстові файли, Microsoft Excel, бази даних SQL та швидкий формат HDF5;

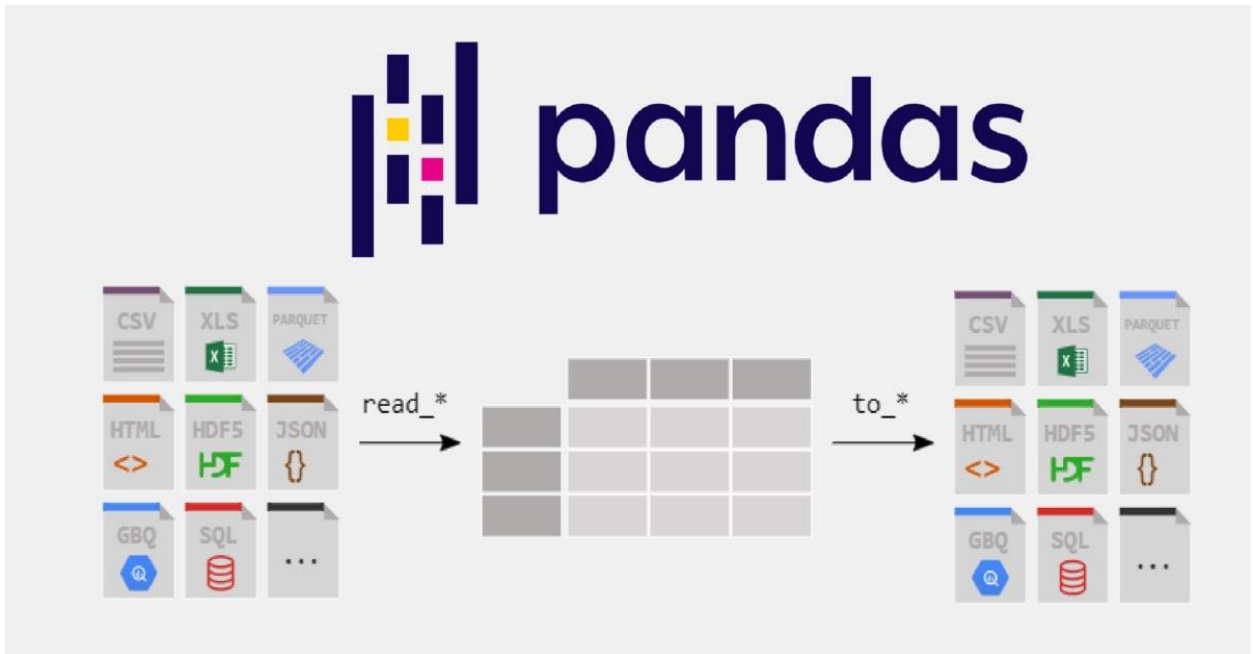


Рисунок 3.11 — Принцип роботи pandas.

Агрегування або перетворення даних за допомогою потужної групи за допомогою механізму, що дозволяє виконувати операції розділення-застосовування-об'єднання над наборами даних;

Функціональність часових рядів : генерація діапазону дат і перетворення частоти, статистика рухомого вікна, зсув і відставання дати. Навіть створюйте зміщення часу для певної області та об'єднуйте часові ряди без втрати даних;

Високо оптимізований для продуктивності , з критичними шляхами коду, написаними на Cython або C.

Python з pandas використовується в широкому спектрі наукових і комерційних областей, включаючи фінанси, економіку, статистику, рекламу, веб-аналітику тощо.

Statsmodels — це модуль Python, який надає класи та функції для оцінки багатьох різних статистичних моделей, а також для проведення статистичних тестів та дослідження статистичних даних. Для кожного оцінювача доступний великий список статистичних даних результатів. Результати перевіряються з існуючими пакетами статистичних даних, щоб переконатися, що вони правильні.

Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами.

Matplotlib — це повна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій на Python.

Seaborn — це бібліотека візуалізації даних Python на основі matplotlib. Він забезпечує високорівневий інтерфейс для малювання привабливої та інформативної статистичної графіки.

SciPy — це програмне забезпечення з відкритим вихідним кодом для математики, науки та інженерних розрахунків.

РОЗДІЛ 4. Опис аналізу даних і створення прогнозу для Walmart.

Для розробки алгоритму спершу було проведено аналіз даних, щоб зрозуміти послідовність створення даних для даних для моделі прогнозування і самого прогнозу

4.1. Аналіз даних продажів Walmart

Аналіз прогнозування продажів я використав набір даних «Прогнозування продажів у магазині Walmart». Дані, які були доступні на Kaggle. У цьому наборі даних ми маємо щотижневі дані про продажі для 45 магазинів і 99 департаментами на 3 роки. Крім того, у нас були специфічні магазини та географія. таку інформацію, як розмір магазину, рівень безробіття, температура, рекламні знижки тощо.

Використовуючи ці фактори, потрібно було розробити регресійну модель, яка може прогнозувати продажі. Основні проблеми, з якими я зіткнувся під час аналізу великий набір даних, що призвело до кількох обчислювальних проблем, через які було змінено наш підхід для вирішення проблеми.

Використавши кілька різних дослідницьких аналізів, щоб визначити ключові змінні для нашої регресії рівняння, такі як графіки кореляції, теплові карти, гістограми тощо.

4.1.1. Набір Даних Walmart

Для розробки моделі прогнозування продажів я використав набір даних «Прогнозування продажів у магазині Walmart». Дані, які були доступні на Kaggle. У цьому наборі даних маємо щотижневі дані про продажі для 45 магазинів і 99 департаментами на 3 роки.

Крім того, протягом року Walmart проводить кілька рекламних заходів. Ці знижки передують видатним святам, чотирма найбільшими з яких

є суперкубок, День праці, День подяки та Різдво. Тижні, включаючи ці святкові дні, оцінюються в п'ять разів більше, ніж не святкові тижні. Частиною завдання, яке представляє цей конкурс, є моделювання впливу уцінок на ці святкові тижні за відсутності повних/ідеальних історичних даних.

stores.csv

Цей файл містить інформацію про 45 магазинів із зазначенням типу та розміру магазину.

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	2010-02-05	24924.50	False
1	1	1	2010-02-12	46039.49	True
2	1	1	2010-02-19	41595.55	False
3	1	1	2010-02-26	19403.54	False
4	1	1	2010-03-05	21827.90	False

Рисунок 4.1 — Приклад даних файл **store.csv**.

train.csv

Це історичні дані про навчання, які охоплюють від 2010-02-05 до 2012-11-01. У цьому файлі ви знайдете такі поля:

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	2010-02-05	24924.50	False
1	1	1	2010-02-12	46039.49	True
2	1	1	2010-02-19	41595.55	False
3	1	1	2010-02-26	19403.54	False
4	1	1	2010-03-05	21827.90	False

Рисунок 4.2 — Приклад даних файл **train.csv**.

Містить такі поля:

- Store - номер магазину
- Dept - номер відділення
- Date – тиждень
- Weekly_Sales - продажі для даного відділу в даному магазині
 - IsHoliday - чи є тиждень особливим святковим

test.csv

Цей файл ідентичний train.csv, за винятком того, що ми приховали щотижневі продажі. Ви повинні передбачити продажі для кожної трійки магазину, відділу та дати в цьому файлі.

features.csv

Цей файл містить додаткові дані щодо діяльності магазину, відділу та регіону на вказані дати.

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	2010-02-05	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
2730	16	2010-02-05	19.79	2.580	NaN	NaN	NaN	NaN	NaN	189.381697	7.039	False
5460	31	2010-02-05	39.05	2.572	NaN	NaN	NaN	NaN	NaN	210.752605	8.324	False
3640	21	2010-02-05	39.05	2.572	NaN	NaN	NaN	NaN	NaN	210.752605	8.324	False
4550	26	2010-02-05	9.55	2.788	NaN	NaN	NaN	NaN	NaN	131.527903	8.488	False

Рисунок 4.3 — Приклад даних файл **features.csv**.

Він містить такі поля:

- Store - номер магазину
- Date – тиждень
- Temperature - середня температура в регіоні
- Fuel_Price - вартість палива в регіоні
- MarkDown1-5 – анонімні дані, пов'язані з рекламними уцінками, які працює Walmart. Дані MarkDown доступні лише після листопада 2011 року і доступні не для всіх магазинів постійно. Будь-яке відсутнє значення позначається NA.

- CPI - індекс споживчих цін
- Unemployment - рівень безробіття
- IsHoliday - чи є тиждень особливим святковим

Для зручності чотири свята припадають на наступні тижні в наборі даних (не всі свята є в даних):

Супербоул: 12 лютого, 10, 11 лютого, 10 лютого, 12, 8 лютого, 13

День праці: 10 вересня, 10 вересня, 9 вересня, 11, 7 вересня, 12, 6 вересня -13

День подяки: 26 листопада 10, 25 листопада 11, 23 листопада 12, 29 листопада 13

Різдво: 31 грудня 10, 30 грудня 11, 28 грудня, 12, 27 грудня -13

4.1.2. Аналіз даних для моделі прогнозування.

Перший крок, я зосередився на об'єднанні наборів даних, щоб отримати всі фактори, які можуть нам допомогти прогнозування тижневих продажів. Я побудував нормалізовану гістограму.

Проблема з даними які не вказані на результат залежності впливають, щоб додати ці колонки ми замінили пусті значення на 0 і також прописали певні випадки для об'єднання полів. Розділили дані на певні групи неділя, квартал, місяць, сезон, рік. Для кращого апелювання даних.

Наступний крок, я створили повну модель, що складається з більшості доступних змінних для полегшення цього я створили змінну з затримкою на основі часових проміжків (щотижневий лаг продажів), вважаємо що це допоможе охопити властиві атрибути певного типу магазину та відділу.

Крім того, я створив фіктивні змінні з категорійних значень, наведених у наборі даних (наприклад, відділ і тип). Основною метою їх створення було врахування деяких відділів які дуже відрізняються від інших. Якщо ми уважніше поглянемо на Теплокарту.

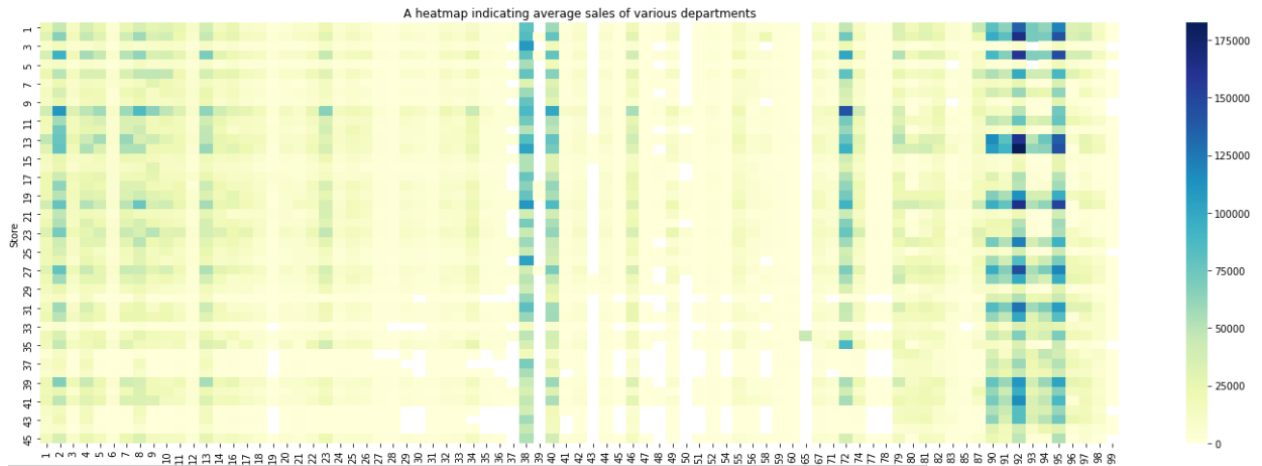


Рисунок 4.4 — Теплова Карта залежності магазинів від відділів.

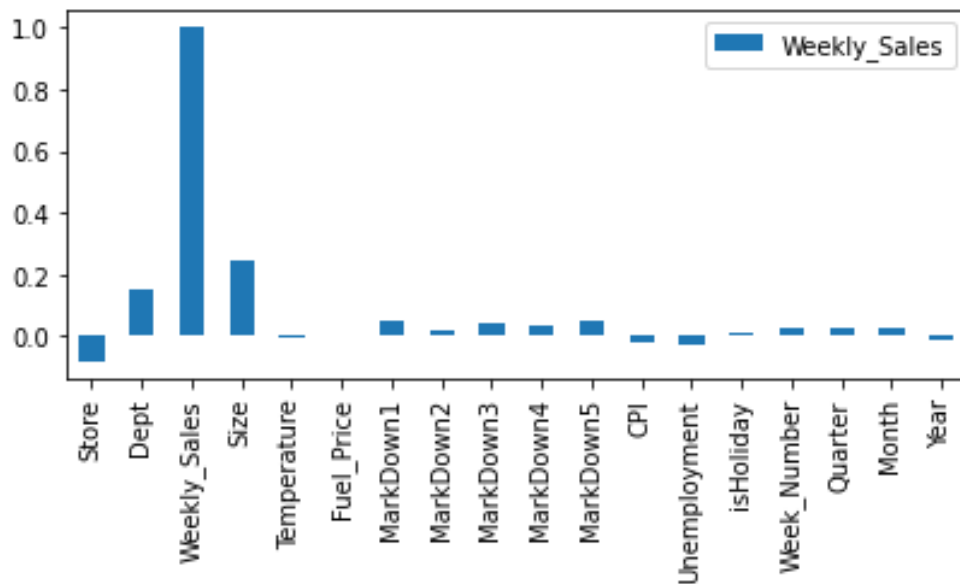


Рисунок 4.5 — Графік кореляції полів прогнозу порівняно з іншими полями.

Ми можемо з огляду на теплову карту упевненістю зробити висновок, що відділи 38, 95 і 92 грають ключову роль у прогнозуванні майбутніх продажів. Тому ми вирішили включити їх у наш регресійні змінні.

Ми також зробили різну описову статистику. Ми побачили кореляційну матрицю і знайшли співвідношення відділу з тижневими продажами.

З аналізу даних і їхній залежності. Знайшли рівняння регресії.

Для перевірки формули я розділив дані на


```
#segregating the data
data_test = master_df_new_var[master_df_new_var.Year == 2012]
data_train = master_df_new_var[master_df_new_var.Year != 2012]
```

Рисунок 4.6 — Розділення даних на тестові для перевірки даних і даних для навчання.

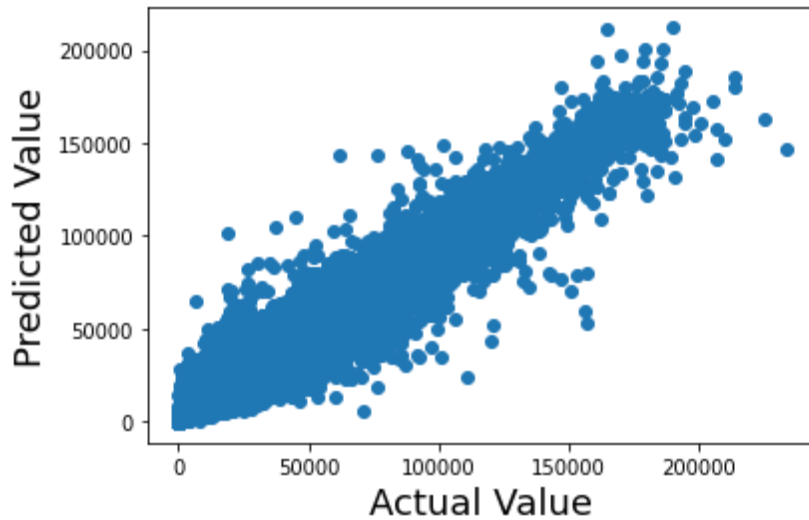


Рисунок 4.7 — Діаграма розкиду фактичного значення проти прогнозованого значення.

Згідно з графіком ми бачимо що формула виведена має збіг зі справжніми даними.

Аналіз часу також потребує займає багато часу. Я подолав це, розділивши партії на міні-партії.

Відповідно ми можемо зробити висновок що процес прогнозування ми розділимо на 2 частини це підготовка даних для моделі прогнозування тобто структуровані дані Dataset. І наступна частина на основі формули та даних структурованих для навчання ми будемо створювати прогноз(Predict). Ці задачі будуть реалізовані у Celery.

РОЗДІЛ 5. ОПИС РОЗРОБКИ ДОДАТКУ

Відповідно перед розробкою додатка проведеного огляду та аналізу у розділі 4. Я описав технічне завдання для створення нашого додатку.

Технічне завдання

- Розробка веб-додатка використовуючи фреймворк **Flask**.
- Архітектура бази даних **PostgreSQL**.
- Реалізація завдання обробки даних про продажі.
- Реалізація завдання прогнозування продажів.
- Створення авторизації для безпеки.
- Розробка UI для зручної роботи.
- Візуалізація даних.
- Візуалізація графіків.

5.1. Опис розробки Веб-додатка

У веб-додаток розроблена архітектура додатка на основі мікро-сервісів.

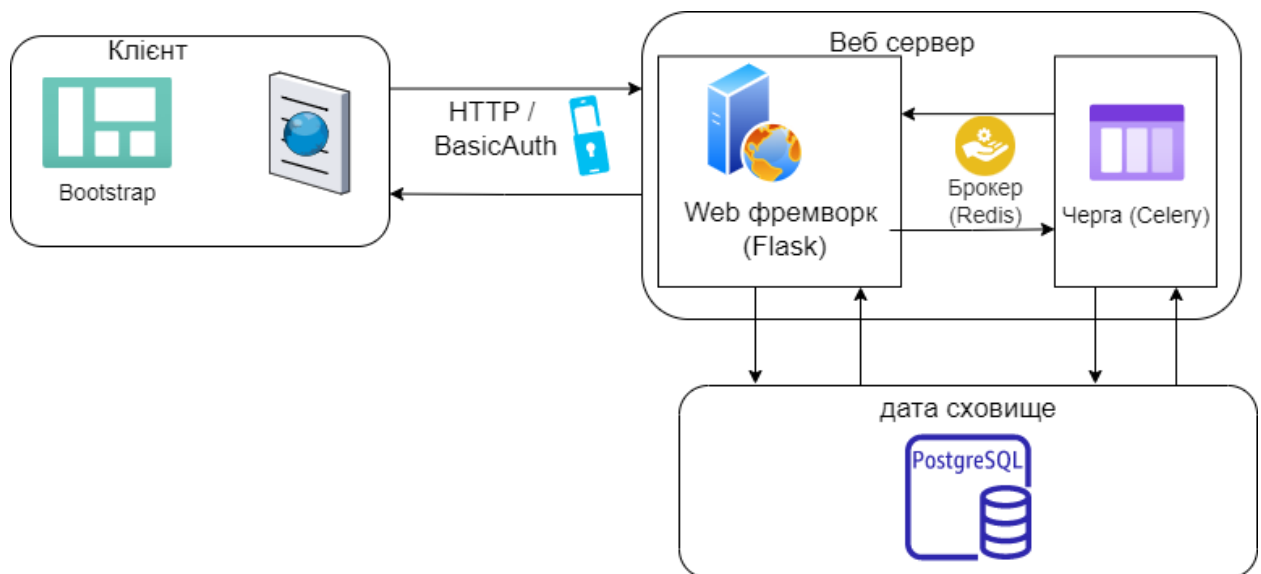


Рисунок 5.1 — Архітектура веб-додатка і зв'язки між мікро-сервісами.

Клієнт браузер який приймає HTTP запити де відповідь отримуємо згенеровану веб-сторінку HTML і за допомогою UI-фреймвоку Bootstrap браузер відображає дані.

Відповідно веб фреймворку Flask до якого під'єднано панель Flask-Admin також залучили Flask-BasicAuth для реалізації аутентифікації до нашого додатка.

Також на сервері створений сервіс черги завдань Celery який отримує задачі від веб фреймворку. Комунікація відбувається за допомогою брокера Redis який також розташований на сервері.

Створенно і під'єднана окреме сховище даних PostgreSQL. База даних має зв'язок з веб-фреймворком і окремо з чергою завдань Celery. Зв'язок це транзакція sql запитів.

Перед реалізацією першої будь яких частин кода створював unit tests використовуючи бібліотекою pytest.

5.2. Опис розробки тестів для перевірки коректної роботи додатка.

Для створення тестів для написання unit tests я використовую бібліотеку pytest. Pytest рахує за тести всі файли з префіксом test та функції або класи які мають також префікс test у назві.

Відповідно дотримуючись правил я створив папку tests з деревом яка повторює дерево папок проекту.

Також для повноцінного використання unit test розумно задіяти функціонал Fixtures які спростять нам тестування код.

```
1 |
2 | def test_index(client):
3 |     response = client.get("/")
4 |     assert response.status_code == 200
```

Рисунок 5.2 — Приклад unit test з fixture з проекта

На картинках продемонстровано створені **fixtures**, наприклад client це fixture яка повертає об'єкт клієнта який функції для HTTP запиті до нашого Flask. Відповідно за допомогою різних порівнянь використовуючи магічне слово **aserrt**.

5.3. Опис реалізації сховища даних PostgreSQL

У веб-додаток розроблена архітектура бази даних. Таблиці і зв'язки між таблицями.

Ми маємо таблиці Rawdata, Dataset, Predict, File

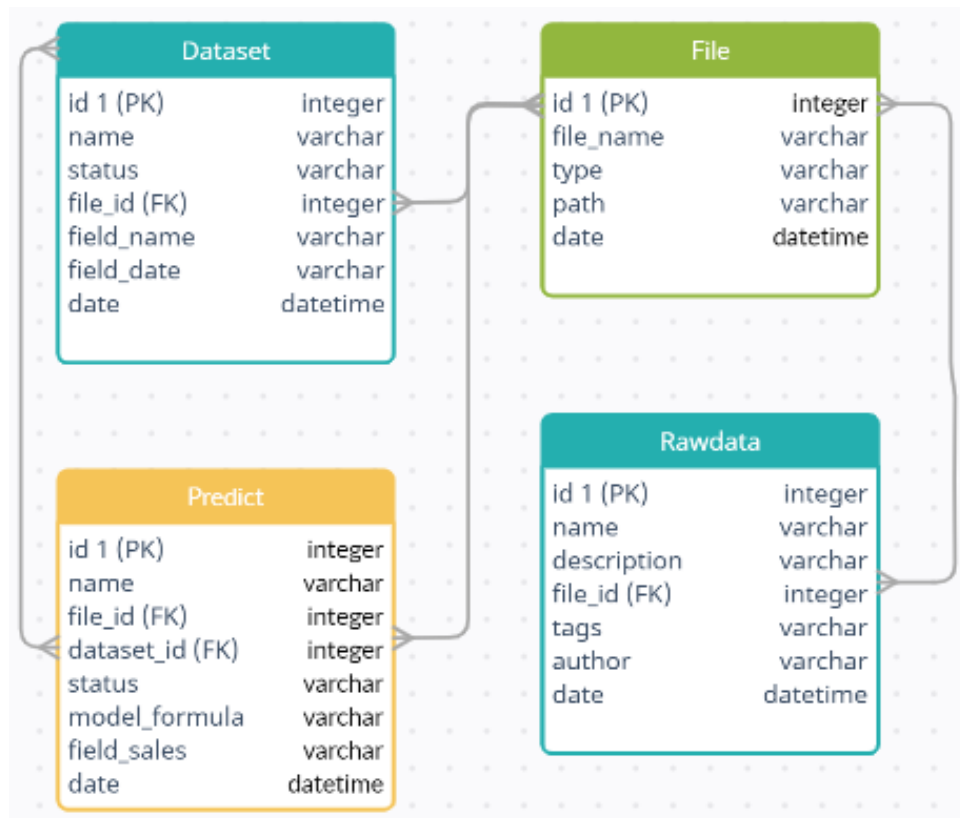


Рисунок 5.3 — таблицями і зв'язки між таблицями.

Таблиці і їхні поля:

Сирі Дані(Rowdata)

- **name** – назва даних
- **description** – опис даних
- **author** – автор даних (магазин, відділ тощо)
- **tags** – назви для класифікації даних
- **file_id** – посилання на файл з даними
- **date** – дата створення

Структуровані Дані (Dataset)

- **name** – назва даних
- **status** – стан процесу (В процесі, завершено, помилка)
- **file_id** – посилання на файл з структурованими даними
- **date** – дата створення

Прогнози(Predict)

- **name** – назва даних
- **status** – стан процесу (В процесі, завершено, помилка)
- **file_id** – посилання на файл з результатом прогнозу
- **dataset_id** – посилання на створені дані для тренування
- **model_formula** – посилання на файл з результатом прогнозу
- **field_sales** – поле для прогнозування
- **date** – дата створення

Файли (File):

- **field_name** – назва файлу
- **path** – путь до файлу
- **type** – тип файлу(Dataset, Rowdata, Predict)
- **date** – дата створення

Така архітектура бази даних з таблицями і зв'язками дає нам можливість повноцінно працювати надалі з даними для прогнозів і обробки у веб у фоновому режимі.

5.4. Опис розробки веб-фреймворка Flask

Flask тут я реалізував запити HTTP також під'єднано панель Flask-Admin разом з Flask-Charts. В результаті у панель керування додано розділи: Головна, Файли, Сирі дані, Структуровані дані, Прогнози.

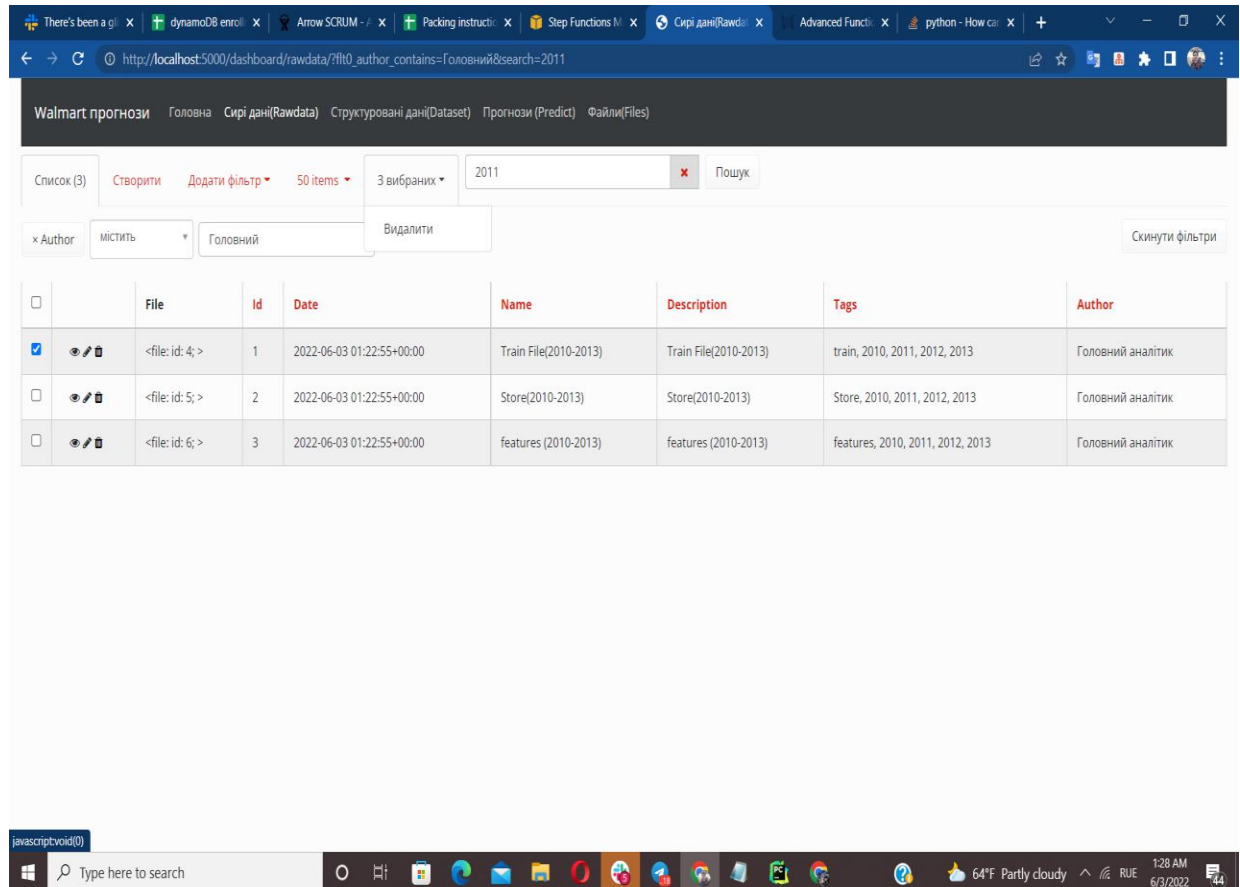


Рисунок 5.5 — Сторінка панель керування

Сторінки створенні за допомогою класу `ModelView` цей клас реалізовує для моделі одразу декілька сторінок(CRUD) це редагувати, список, видалити, створити, деталі і додатковий функціонал такий як пошук, фільтр даних тощо.

На сторінці списку на кожній сторінці доступна навігація для редагування, видалення і детального перегляду даних.

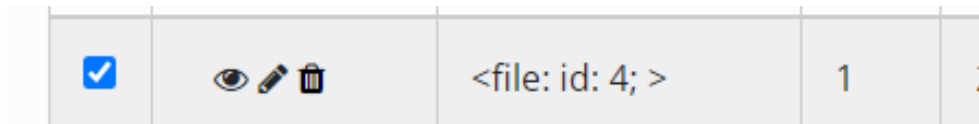


Рисунок 5.4 — Посилання на редагування, деталі і видалення даних.

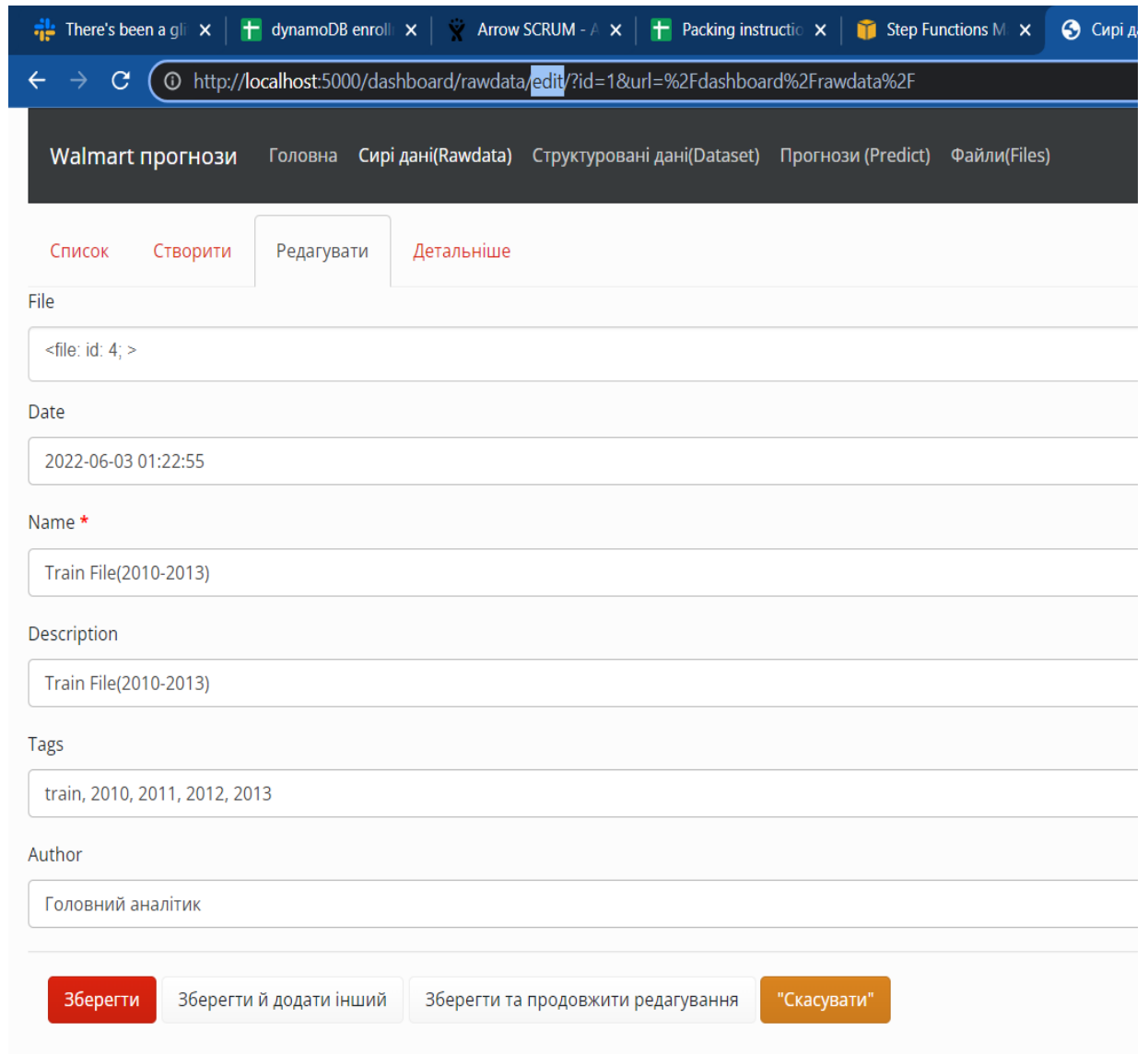


Рисунок 5.6 — Сторінка редагувати Сирі дані(Rawdata).

Реалізовані запити додатка:

- **/dashboard/** – Головна сторінка

Головна сторінка – де продемонстровані коротко дані інформація по даним і прогнозам також продемонстровані основні графіки для прогнозів.

- **/dashboard/rawdata/(CRUD)** – Сирі Дані (**Rawdata**)

Базові сторінки для роботи з сирими даними.

- / - список
- /create Створити
- /edit Редагувати
- /detail Деталі
- /delete Видалити

- **/dashboard/dataset/(CRUD)** – Структуровані Дані (**Dataset**)

Базові сторінки для роботи з структуровані даними.

- / - список
- /create Створити
- /edit Редагувати
- /delete Видалити

- **/dashboard/predict/(CRUD)** – Прогнози (**predict**)

Базові сторінки для роботи прогнозами продажів.

- / - список
- /create - Створити
- /edit - Редагувати
- /delete - Видалити

- **/dashboard/file/(CRUD)** – Файли (**File**)

Базові сторінки для керування файлами додатка можна видаляти завантажувати тощо.

- / - список
- /create - Створити
- /edit - Редагувати
- /delete - Видалити
- /download – Скачати файл.

5.5. Налаштування Черги завдань Celery

Для роботи з чергою завдань Celery ми під'єднали брокер Redis. Redis є посередником від клієнта це наш Flask і чергою завдань Celery.

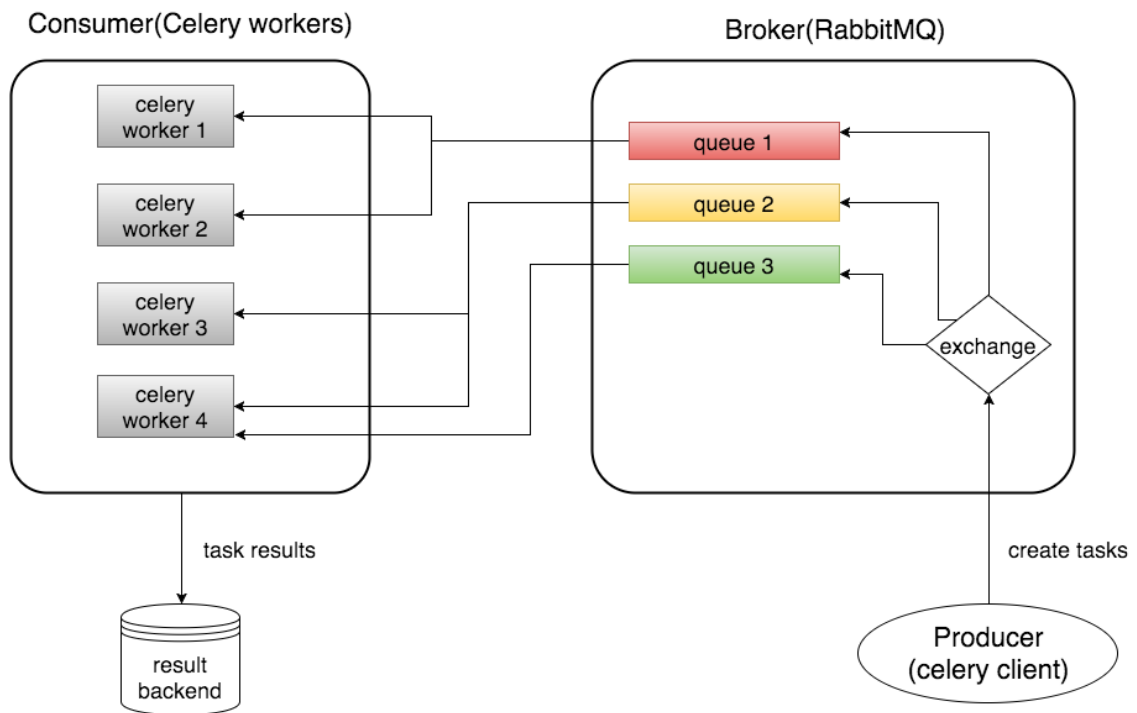


Рисунок 5.7 — Схема черги завдань.

Щоб використати паралельність роботи завдань виділено 4 обробники та 3 черги для нашого додатка.

- 2 обробники для завдань для обробки даних.
- 2 для завдань для прогнозування.
- Також виділено 3 черги.

Як видно зі схеми то для перша черга відправляє завдання на два обробники - це завдання для прогнозування. Для обробки даних створено 2 черги та 2 обробника що дає нам гарний розподіл навантаження і в результаті маємо гарну продуктивність даної моделі.

5.6. Безпека і доступи

Так як додаток розробляється для бізнесу і так як додаток працює з даними про фінанси і продажі тощо відповідно потрібно забезпечити рівень додатка. Додаток є індивідуальним сервісом відповідно нам базова аутентифікація на Basic-Auth рівні HTTP буде достатньо. Basic-Auth — це метод, який агент користувача HTTP надає ім'я користувача та пароль під час виконання запити.

До веб-ядра Flask підключено модуль Flask-BasicAuth і задано відповідні налаштування.

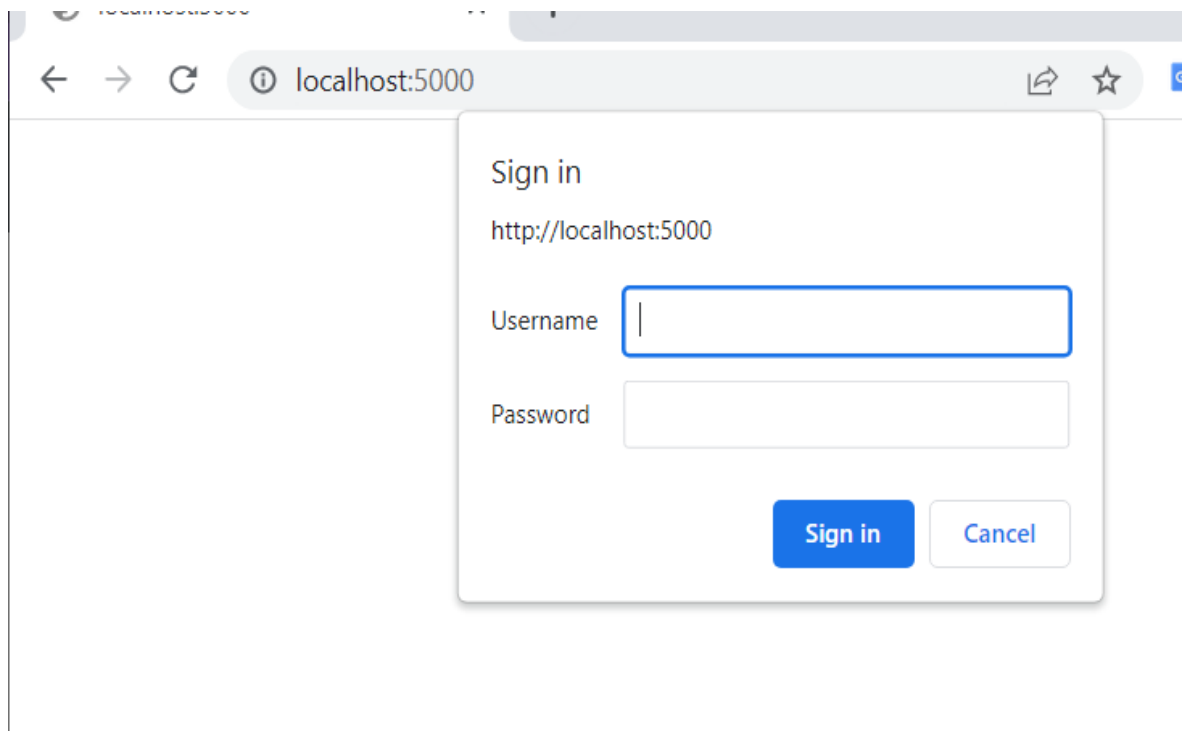


Рисунок 5.8 — авторизації користувача до додатка.

Відповідно ми маємо метод авторизації в систему що забезпечує обмежений рівень доступ і рівень захисту до веб-додатка.

5.7. Опис реалізація завдання обробки даних про продажі Dataset.

На основі проведеного аналізу даних Walmart в розділі 4 – ми описали алгоритм для обробки даних про продажі.(Dataset).

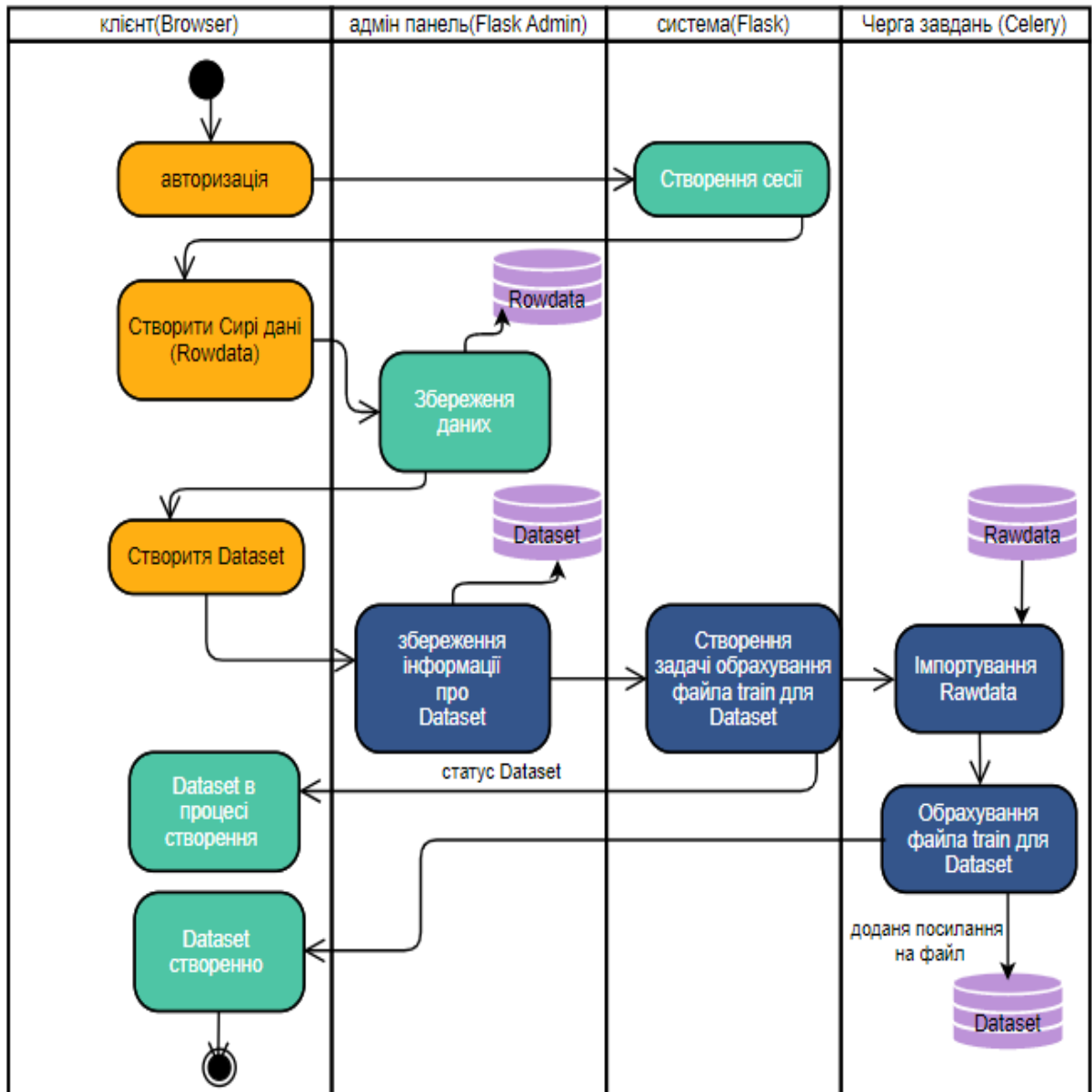


Рисунок 5.9 — Діаграма активності — створення Dataset.

Згідно з послідовності продемонстровано що перш потрібно отримати доступ до панелі керування.

Далі уже користувач може обрати додати сирі дані додавши і записавши їх в таблицю.

Після запису сирих даних ми маємо змогу згенерувати Dataset. Це й кров виглядає як форма з вказаними налаштуваннями для початку обробки даних.

Далі запускається задача `create_train` в черзі завдань Celery. Після результат записується в файл на сервері і приходиться користувачу повідомлення про завершення задачі.

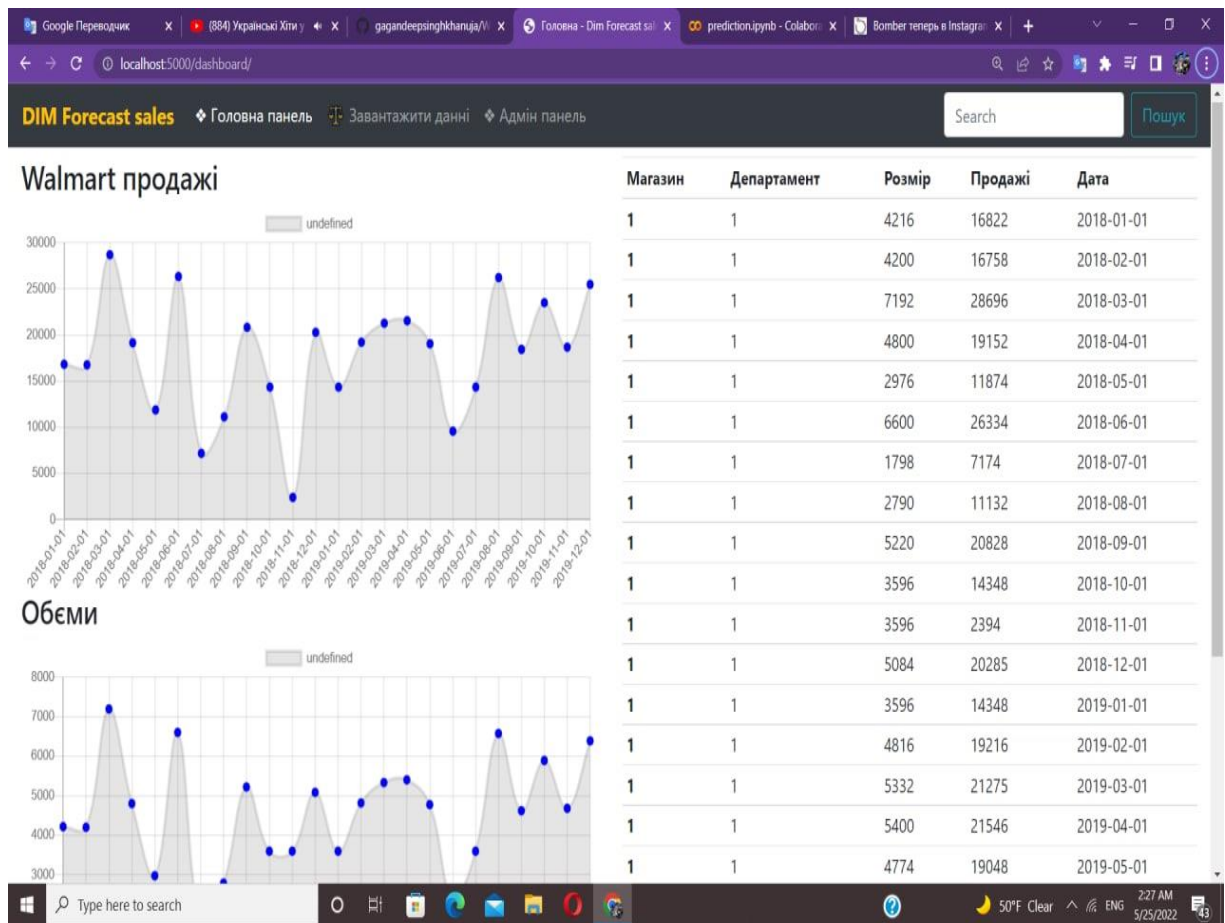


Рисунок 5.10 — Приклад сторінки Dataset даних в режимі аналізу.

Після завершення задачі обробки даних може перейти в режим аналізу і переглянути дані в таблиці також для аналізу маємо графіки які створенні за допомогою сервісу **Google Charts**.

5.8. Опис реалізація завдання прогнозування продажів (Predict)

На основі проведеного аналізу даних Walmart в розділі 4 – ми описали алгоритм для створення прогнозу (Predict).

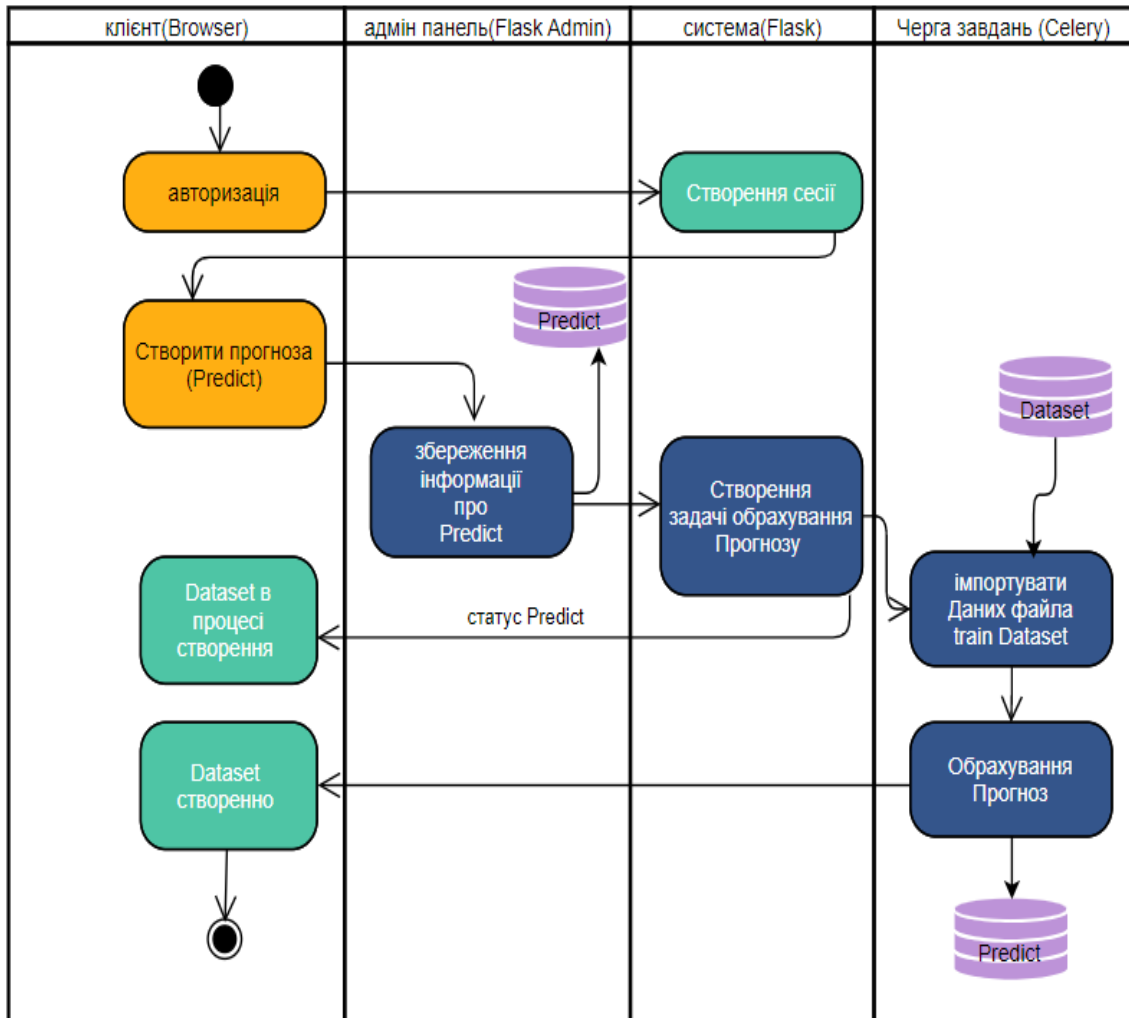


Рисунок 5.11 — Діаграма активності — створення прогнозу (Predict)

Згідно з послідовностями бачим що першим потрібно отримати доступ до панелі керування.

Далі уже користувач може обрати створити прогноз додавши і записавши їх в таблицю. Це й крив виглядає як форма з вказаними налаштуваннями де потрібно обрати створений файл Dataset додати формулу для прогнозу і відповідно сам файл з даними для прогнозу.

Далі запускається задача **make_predict** в черзі завдань Celery. Після результату записується в файл з результатом на сервері і приходиться користувачу повідомлення про завершення задачі.

Після завершення задачі обробки даних може перейти в режим аналізу і переглянути дані в таблиці також для аналізу маємо графіки які створенні за допомогою сервісу **Google Charts**.

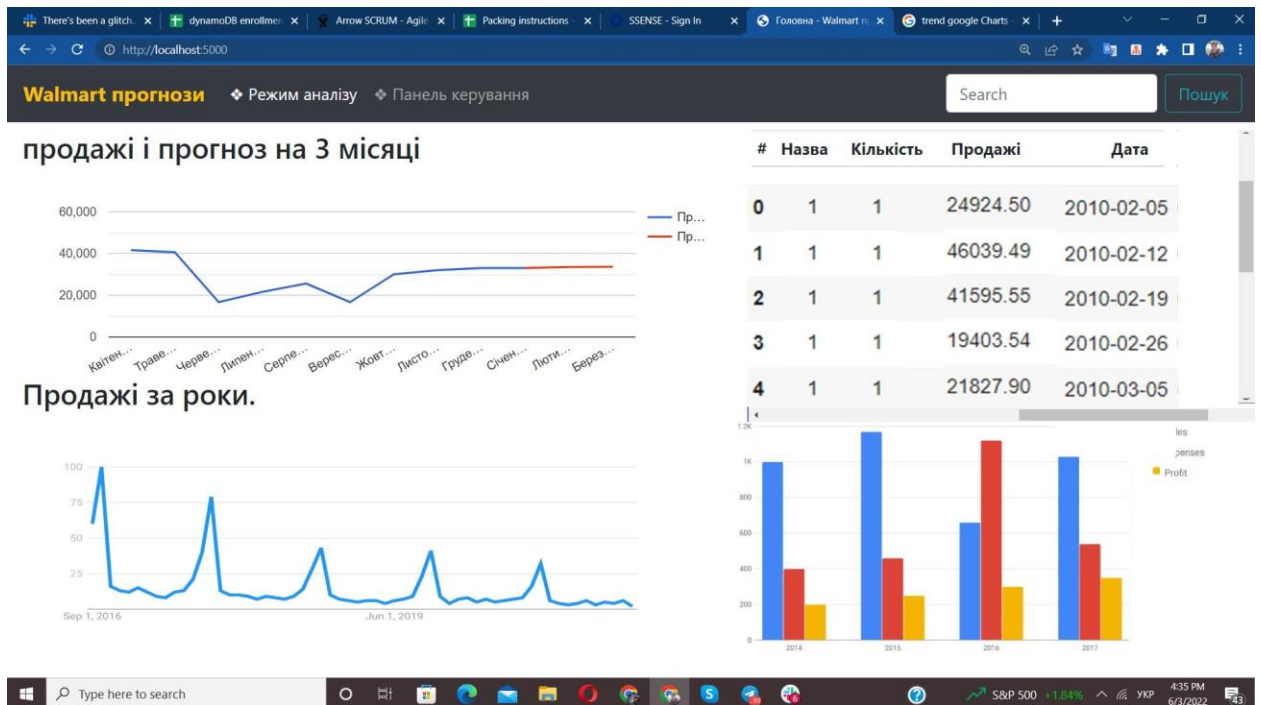


Рисунок 5.12 — Приклад сторінки Predict даних в режимі аналізу.

Аналітик має змогу оцінювати продажі прогнозу робити певні висновки тощо аналізувати якщо певної звітності замало аналітика є можливість перейти на файл з результатом прогнозу і скачати файл з прогнозами в форматі CSV.

5.9. Опис панелі керування в режимі аналітики.

На цій сторінці реалізовано вхід на отримання даних і різні варіанти відображення, наприклад різні варіації графіків таблиць тощо.

Графіки створюються за допомогою під'єднаної бібліотеки Flask-Charts. В ній описаний клас Chart який і задаються різні формати налаштувань для відображення графіків.

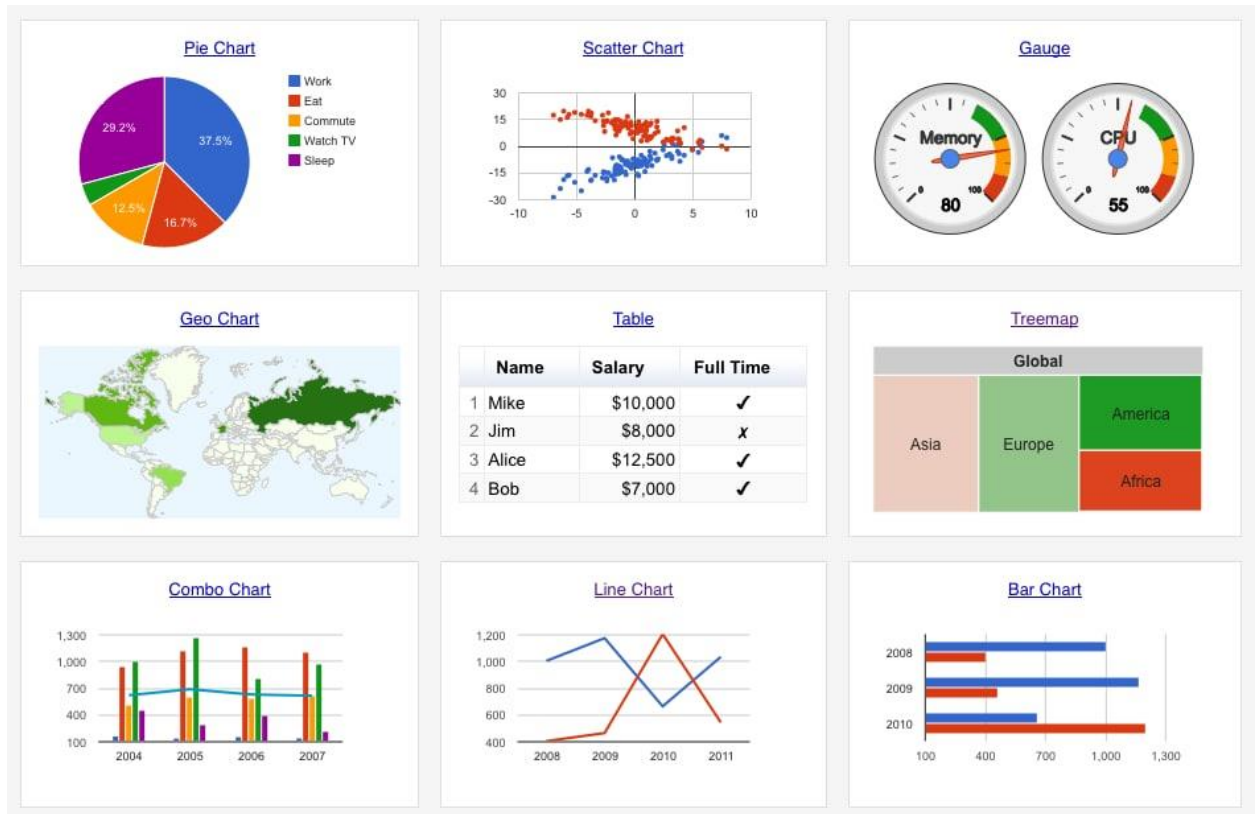


Рисунок 5.13 — Приклад доступних діаграм для демонстрації.

Відповідно ми маємо зручний інтерфейс з інструментами для аналітика й аналізу.

ВИСНОВКИ

З виконаної роботи можна зробити певні висновки, про розробку веб-додатка для зберігання і обробки даних на сервері для спрощення прогнозування продажів за даними корпорації Walmart мовою програмування Python.

Оглянувши існуючих додатки які існують на ринку для прогнозування продажів товарів, а саме: **Streamline, Oracle Demantra, SAP, Anaplan, SPOTIO і InsightSquared**. Можна сказати що додатки: **Streamline, Oracle Demantra, SAP** – основному вони дають поверхневу оцінку аналізу і прогнозу оскільки вони націлені більше на роботи з кроками поставок тощо. **SPOTION, InsightSquared і Anaplan** вони чудом описують і підходять для точних прогнозування продажів і гарно описують підхід прогнозування. Також гарний підхід для прогнозування продажів використовувати **III** оскільки описано в **InsightSquared**.

Оглянувши засоби для програмної реалізації рішення було обрано рішення реалізації програмного продукту, найкращим рішенням є **веб-додаток**, що має **клієнт-серверну** архітектуру, де клієнтська сторона відповідає за роботу та логіку програмного продукту, а серверна – за обчислення.

Було акцентовано розділити додаток на **мікро-сервіси**.

Було обрано наступні програмні засоби для реалізації:

- Мови програмування – **Python, JS**
- Фреймворки – **Flask, Bootstrap, Google Charts**
- Засоби роботи з файлами і для обробки даних – **Pandas, Statsmodels, SciPy, Matplotlib, Seaborn, numpy**.
- База даних – **PostgreSQL**
- Тестування – **Unit Tests**
- Черга завдань – **Celery**

Для розробки було спроектовано **архітектура веб-додатка і зв'язки між ними мікро сервісами**. Також було спроектована **архітектура бази даних PostgreSQL**.

Для забезпечення безпечного доступу до додатка було задіяно **BasicAuth**. Також було розглянуто використання **Pytest** для тестування кода.

Клієнт браузер відображення даних було реалізовано за допомогою UI-фреймвока **Bootstrap**.

Реалізовано в додатку веб-фреймворк **Flask** до якого під'єднано панель **Flask-Admin** також задіяно **Flask-BasicAuth** для реалізації аутентифікації до нашого додатка.

Також на сервері створений сервіс черги завдань **Celery**. Зв'язок відбувається за допомогою брокера **Reddit** який також розташований на сервері.

Створення і під'єднана окремо сховище даних **PostgreSQL**.

Використовуючи і проаналізувавши опубліковані дані про продажі товарів **Walmart** магазинів за **2010-2013 рік** на сайті **Kaggle**. Було розроблено алгоритм створення структурованих даних для тренування моделі. Відповідно на основі структурованих даних виведено алгоритм створення прогнозу.

Для демонстрації результатів прогнозу у вигляді графіків було задіяно застосунок **Google Charts**.

Для перегляду результатів існує в розділі деталі в про прогноз доступний **режим аналізу** який відображає дані і графіки про ці дані.

Мій додаток націлений для аналітиків **Walmart**. Це доволі зручне рішення для спрощення прогнозування продажів за даними корпорації **Walmart**.

Також реалізовано як сховище сирих даних які можуть завантажувати, наприклад: завідувач магазинів або департаментів тощо що **спрощує значно збір даних для аналітика**.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Прогнозна аналітика (Predictive Analytics). <https://techexpert.ua/solutions-it/proгнозна-analytyka/>
2. Demand Management Overview. https://docs.oracle.com/cd/E18727_01/doc.121/e05179/T493571T493575.htm
3. Solutions to intelligently control your digital supply chain | Kinaxi <https://www.kinaxis.com/en/solutions>
4. What is SAP? <https://www.sap.com/ukraine/about/company/what-is-sap.html>
5. Walmart Recruiting - Store Sales Forecasting <https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting/overview>
6. About Python. <https://www.python.org/about/>
7. Рейтинг мов програмування 2022. С# обійшов Java, TypeScript зрівнявся з PHP, а Dart — найбільш комфортна мова <https://dou.ua/lenta/articles/language-rating-2022/>
8. Most Popular Backend Frameworks – 2012/2022 <https://data/most-popular-backend-frameworks-2012-2022/>
9. HTML Introduction. https://www.w3schools.com/html/html_intro.asp
10. CSS Introduction. https://www.w3schools.com/css/css_intro.asp
11. About JavaScript. https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
12. Overview Bootstrap 4. <https://getbootstrap.com/docs/4.0/layout/overview/>
13. Welcome to Flask's documentation. <https://flask.palletsprojects.com/en/2.1.x/>
14. What is PostgreSQL? <https://www.postgresql.org/about/>

15.Результати Stack Overflow's Developer Survey 2020.

<https://dou.ua/forums/topic/30700/>

16.About SQLAlchemy <https://www.sqlalchemy.org/>

Додаток А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка програмного забезпечення для прогнозування продажів товарів мовою Python

Виконав студент 4 курсу,
група ПД-41
Бондар Дмитро Володимирович
Керівник роботи

Трінтіна Наталя Альбертівна






Київ – 2020

К.Т.Н., ДОЦЕНТ

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – Розробка веб-додатка для зберігання і обробки даних на сервері для спрощення прогнозування продажів за даними корпорації Walmart мовою програмування Python.
- **Об'єкт дослідження** – Веб-додаток, серверні можливості для роботи з файлами. Дані про прогнозування продажів корпорації Walmart.
- **Предмет дослідження** – Розробка веб-додатка, зберігання і обробка даних на сервері для створення прогнозу продажів за даними корпорації Walmart мовою програмування Python.

АНАЛОГИ

 Streamline	 SAP ERP	 Anaplan	 INSIGHTSQUARED	 ORACLE Demantra
Тільки для windows.	налаштовується в Excel	Націлена для роботи на Американському ринку	прогнозування продажів на основі штучного інтелекту	Деталізація прогноза. Складний інтерфейс.

3

ВИСНОВКИ

Створено веб-додок для зберігання і обробки даних на сервері для спрощення прогнозування продажів за даними корпорації Walmart мовою програмування **Python**.

1. Створено веб-додаток використовуючі фреймворк **Flask**.
2. Реалізована архітектура баз даних на **PostgreSQL**.
3. Підєднано панель адміністратора **Flask-Admin**.
4. Реалізовано завдання обробки даних і прогнозування продажів на сервері використовуючи чергу **Celery**.
5. Реалізовано авторизації для безпеки **BaseAuth**.
6. Реалізовані інтерфей використовуючі фреймворк Bootstrap і графікі - **Google Developers**.

13

ДЯКУЮ ЗА УВАГУ!