MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Project Report

on

Customer Relationship Model with Analysis Features

Submitted By:

Parn Jain

0901AD211035

Shantanu Dixit

0901AD211056

Faculty Mentor:

Ms. Gaurisha Sisodia

Assistant Professor, Centre for Artificial Intelligence

CENTRE FOR ARTIFICIAL INTELLIGENCE

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Parn Jain** (0901AD211035) and **Shantanu Dixit (0901AD211056)** has submitted the project report titled **Customer Relationship Model with Analysis Features under** the mentorship of **Assistant Professor Gaurisha Sisodia**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Artificial Intelligence and Data Science** from Madhav Institute of Technology and Science, Gwalior.

23/11/23

Ms. Gaurisha Sisodia
Faculty Mentor
Assistant Professor
Centre for Artificial Intelligence

23/11/23

Dr. Rajni Ranjan Singh
Coordinator
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in **Artificial Intelligence and Data Science** at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Ms. Gaurisha Sisodia**, Assistant Professor, Centre for Artificial Intelligence

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Parn Jain
0901AD211035
III Year
Centre for Artificial Intelligence

Shantanu Dixit
0901AD211056
III Year
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Ms.Gaurisha Sisodia**, **Assistant Professor**, Centre for Artificial Intelligence, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Parn Jain
0901AD211035
III Year,
Centre for Artificial Intelligence

Shantanu Dixit
0901AD211056
III Year,
Centre for Artificial Intelligence

ABSTRACT

The Customer Relationship Management (CRM) Analysis System is a versatile web-based project designed to streamline data management and provide meaningful insights into customer interactions and product sales. The system incorporates three interconnected tables: one for customer details, another for product information, and a third for recording customer purchases.

The primary goal of this project is to offer a user-friendly interface for inputting and retrieving data efficiently. Customers, products, and purchase records are organized in separate tables, ensuring a well-structured database for easy access to information.

The Analysis page of the web application presents a range of visualizations that help users grasp key business trends. The system provides dynamic graphs, such as State vs. Quantity, City vs. Quantity, and Date (month and year) vs. Quantity, offering users a visual representation of sales data for strategic decision-making.

Moreover, the system generates generalized reports, presented in the form of interactive cards, covering aspects like top customers, states, cities, and best-selling products. These reports are designed to provide users with a quick overview of essential performance indicators without imposing rigid criteria.

The CRM Analysis System is intended to empower businesses with a tool that enhances data-driven decision-making. By facilitating a deeper understanding of customer behavior, regional sales patterns, and product popularity, the system aims to optimize CRM strategies and contribute to overall operational efficiency. Its user-friendly design and flexible analytics make it a valuable asset for businesses looking to harness the power of data for sustainable growth.

Keyword: CRM, Django, Analysis, Business, Data Analysis, Graphs, Website, HTML, CSS, JavaScript,

सार:

ग्राहक संबंध प्रबंधन (CRM) विश्लेषण प्रणाली एक बहुपयोगी वेब-आधारित परियोजना है जिसका उद्देश्य डेटा प्रबंधन को सरल बनाना और ग्राहक संबंध और उत्पाद बिक्री में मार्गदर्शन करना है। प्रणाली में तीन एक-दूसरे से जुड़े तालिकाएँ शामिल हैं: ग्राहक विवरणों के लिए एक, उत्पाद जानकारी के लिए दूसरा, और ग्राहक खरीदी की जानकारी के लिए तीसरा।

इस परियोजना का मुख्य उद्देश्य डेटा प्रबंधन को सुगम बनाए रखना और गहरे बिजनेस बुद्धिमत्ता को प्रदान करना है। प्रणाली एक उपयोगकर्ता-मित्र संवेदनशील इंटरफेस प्रदान करती है जिसमें डेटा को सरलता से दर्ज करने और पुनः प्राप्त करने की सुविधा है। ग्राहक, उत्पाद, और खरीदी की जानकारी अलग-अलग तालिकाओं में संग्रहित की जाती है, जिससे जानकारी को आसानी से पहुँचने की सुनिश्चित किया गया है।

वेब एप्लिकेशन की विश्लेषण पृष्ठ पर उपयोगकर्ताओं को कुंजीय व्यापकताओं को समझने में मदद करने के लिए विभिन्न चित्रों का सीधा प्रदर्शन करता है। प्रणाली गतिविधियों की रणनीतिक निर्धारण में राजनीतिक निर्णय के लिए राजस्व डेटा की दृष्टि प्रदान करने वाले राज्य बनाम मात्रा, शहर बनाम मात्रा, और तिथि (महीना और वर्ष) बनाम मात्रा जैसे ग्राफ प्रदान करता है।

इसके अतिरिक्त, प्रणाली उपयोगकर्ताओं को सरलता से विभिन्न प्रमुख प्रदर्शन सूचियों के साथ (बिना कड़ा नियम लगाए) उपयोगकर्ता, राज्य, शहर, और सबसे बिकने वाले उत्पादों जैसे पहले 10 विवरण उत्पन्न करती है। इन रिपोर्ट्स का उद्देश्य उपयोगकर्ताओं को चुनौतीपूर्ण मानकों के बिना महत्वपूर्ण प्रदर्शन सूचियों की त्वरित अवलोकन प्रदान करना है।

CRM विश्लेषण प्रणाली का उद्देश्य व्यापारों को एक ऐसे टूल के साथ सशक्त करना है जो डेटा द्वारा निर्धारित निर्णय लेने की क्षमता में सुधार करता है।

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The aim of this project is to create a user-friendly CRM Analysis System that helps businesses manage and understand their data better. We want to make it easy for users to input and find information about customers, products, and purchases. The system will have a special Analysis page with easy-to-read graphs that show important trends in sales. Our goal is to provide a tool that lets businesses make smart decisions based on these trends. We also aim to make the system flexible, so users can customize reports based on what they need. Ultimately, we want the CRM Analysis System to empower businesses with insights into customer behavior, regional sales, and popular products, leading to better customer relationships and overall success..

1.1 Project Aim:

The primary aim of the CRM Analysis System project is to revolutionize data management and decision-making in businesses. By offering a user-friendly interface and a well-structured database, the system aims to streamline the organization and retrieval of customer details, product information, and purchase records. The dynamic visualizations on the Analysis page provide users with key insights into business trends, while the flexible analytics and interactive reports cater to evolving business needs. Ultimately, the project seeks to deepen the understanding of customer behaviour, regional sales patterns, and product popularity, empowering businesses with the tools needed for optimized CRM strategies and sustainable growth..

1.3 Objective:

The objectives of this report are multi-faceted, aimed at providing a comprehensive understanding of the CRM Analysis System and its impact on modern business practices. Firstly, the report seeks to elucidate the technical underpinnings of the system, delving into its architecture, data management structures, and user interface design. It aims to highlight the efficiency and user-friendliness of the system in organizing and retrieving customer data. Furthermore, the report endeavors to showcase the practical utility of the Analysis page by dissecting the dynamic visualizations it offers, elucidating their role in aiding strategic decision-making. With a keen focus on the flexibility and adaptability of analytics, the report aims to underscore the system's capacity to meet diverse business needs. Ultimately, the report aspires to contribute valuable insights into how the CRM Analysis System empowers businesses, optimizing customer relationship management strategies, and fostering sustainable growth through data-driven decision-making.

CHAPTER 2: LITERATURE SURVEY

The literature survey for the Customer Relationship Management (CRM) Analysis System delves into several key areas to inform its design and functionality. Firstly, an examination of CRM systems offers insights into established practices for managing customer interactions. Studies in this domain explore the impact of CRM tools on customer satisfaction, loyalty, and overall business performance. Understanding the nuances of effective CRM systems is crucial for developing a platform that optimally addresses customer relationship needs.

Secondly, a review of literature on data management in the context of CRM provides essential guidance. This involves investigating best practices in structuring customer-related data, database design, and data quality assurance. Efficient data organization is fundamental for the CRM Analysis System's goal of facilitating easy access to information. Insights from this literature domain contribute to the system's architecture, ensuring a well-structured database that supports seamless data retrieval and analysis.

Finally, the literature survey extends to research on data visualization techniques, user interface design for data-driven systems, and the role of analytical tools in business decision-making. Understanding how visualizations contribute to user comprehension, principles of user interface design, and the impact of analytical tools on strategic decision-making informs the design of the Analysis page. By synthesizing findings from these diverse literature domains, the CRM Analysis System aims to be a user-friendly, efficient, and insightful tool for businesses seeking to optimize their customer relationship strategies and overall operational efficiency through data-driven insights.

CHAPTER 3: SYSTEM SPECIFICATION

3.1 Hardware Specification

Since the hardware is an important part while developing a web project, it's necessary to find hardware requirements.

Processor–

The minimum level of the required processor for this platform is Pentium IV with 800 MHz processing speed. Since the time taken for processing the instruction depends on the processor's power, it is very important to choose the required processor.

RAM– For a higher speed of processing, it also depends on the memory. Therefore, for better performance minimum RAM should be 2 GB.

Hard disk– In modern days, a vast amount of data is generated daily from internet platforms. So, a good size hard disk is required for the storage of the processed data.

Cache Memory– The access time of the operation tasks mainly depends on the cache memory. Therefore, the recommended cache memory is 1000 KB.

3.2 Software Specification

Our system should meet the following minimum specifications

OS — Ubuntu, Windows, Mac OS

We will be using HTML, CSS, JAVASCRIPT, Python, Django and Matplotlib and we also need an application **Visual**

Studio Code and PyCharm as a code editor.

CHAPTER 4: METHODOLOGY

4.1 DATABASE MAKING:

We have made a SQL database so as to store the information of Customer , Products and Order Records

3 Tables are created and connected with each other using Foreign key

Code is done in models.py file in Django

```python
from django.db import models

class Product(models.Model):
    name = models.CharField(max_length=100)
    quantity_available = models.IntegerField(default=0)  # Added field for quantity available

    def __str__(self):
        return self.name

class Records(models.Model):
    Name = models.CharField(max_length=50)
    Email = models.EmailField(max_length=100)
    phone = models.CharField(max_length=50)
    city = models.CharField(max_length=50)
    address = models.CharField(max_length=100)
    state = models.CharField(max_length=50)
    products = models.ManyToManyField(Product, through='OrderProduct')

    def __str__(self):
        return self.Name

class OrderProduct(models.Model):
    OrderData = models.DateField(default='2023-01-01')  # Moved the OrderData field here
    record = models.ForeignKey(Records, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    quantity = models.IntegerField()
    unit = models.CharField(max_length=10)

    def __str__(self):
        return f"{self.record.Name} - {self.product.name}"
```

Fig 4.1.1

3 Tables are as follows :-

1.)

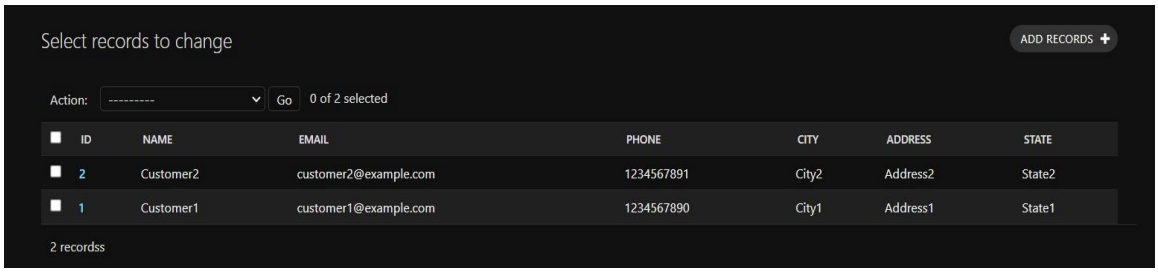

Fig 4.1.2

2.)

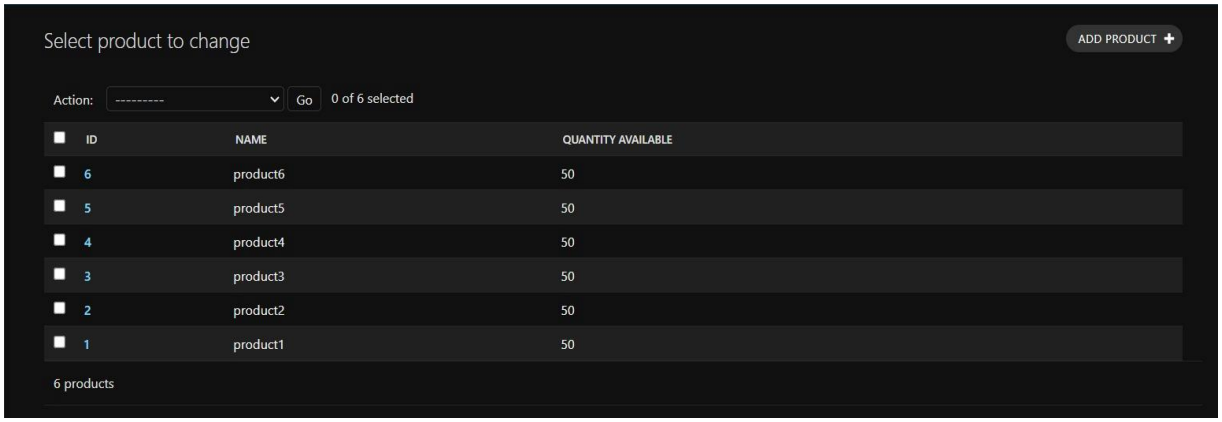

Fig 4.1.3

3.)

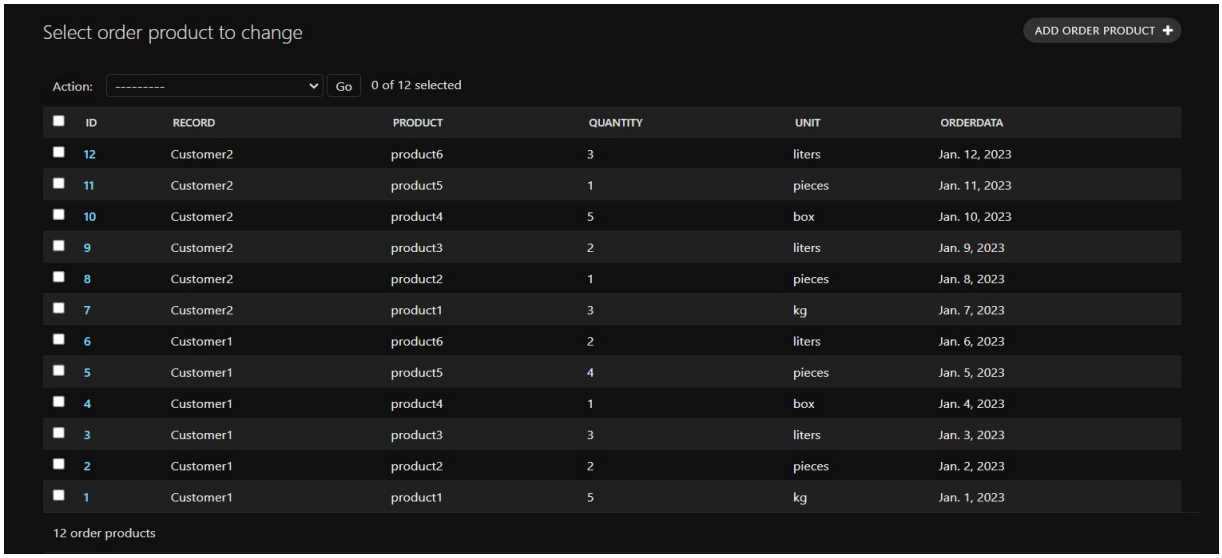

Fig 4.1.4

4.2 ADDING DATA:

A form is made in HTML and connected it to Django and SQL using Django Formset , views functions handles the data from front-end and send it to SQL to store in database in the form of Tables

Fig 4.2.1

An add product button is created using JavaScript to dynamically create the form to add more products since it depends on the customer how many different products they buy

To fill the bulk sample data, we use SQL queries :-

```
-- Insert products
INSERT INTO website_product (name,quantity_available,OrderData) VALUES
  ('product1',50),
  ('product2',50),
  ('product3',50),
  ('product4',50),
  ('product5',50),
  ('product6',50);

-- Insert customers
INSERT INTO website_records (Name, Email, phone, city, address, state) VALUES
  ('Customer1', 'customer1@example.com', '1234567890', 'City1', 'Address1', 'State1'),
  ('Customer2', 'customer2@example.com', '1234567891', 'City2', 'Address2', 'State2');

-- Insert order products
INSERT INTO website_orderproduct (record_id, product_id, quantity, unit,OrderData) VALUES
  -- For Customer1
  (1, 1, 5, 'kg','2023-01-01'),
  (1, 2, 2, 'pieces','2023-01-02'),
  (1, 3, 3, 'liters','2023-01-03'),
  (1, 4, 1, 'box','2023-01-04'),
```

Fig 4.2.2

4.3) Handling Requests

Handling requests in web development involves directing client requests to the appropriate backend components through routing, extracting necessary data, executing backend logic, and generating responses. Middleware functions may intervene for tasks like authentication or logging. Effective error handling and security measures, such as input validation, are essential for a reliable and secure request-handling process.

We have written code to handle requests in views.py file:

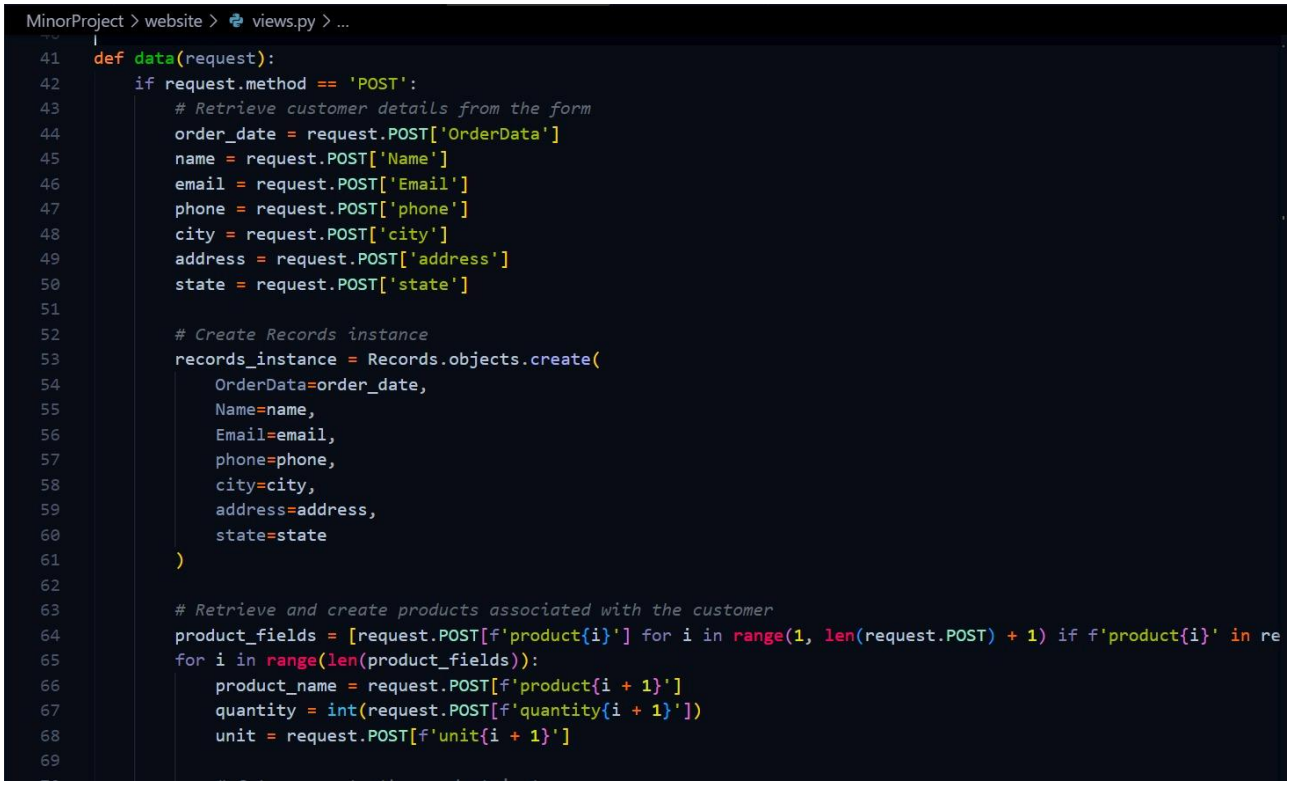

Fig 4.3

4.4) Data analysis

Data analysis involves examining and interpreting datasets using statistical and computational methods to extract meaningful insights and support decision-making. Techniques like descriptive statistics and data visualization are employed to identify patterns and trends, facilitating informed conclusions and predictions.

After making database and collecting data , we applied analysis on data

The code to analysis data is written in analysis.py file, Data from SQL is extracted using SQL cursor and analysis steps are applied on data :

MinorProject > website > analysis.py > fetch_records

```python
import os
import sys
import django
sys.path.append("D:/College/5th Sem/Minor Project/MinorProject")  # Replace with the correct path to your project roo
os.environ['DJANGO_SETTINGS_MODULE'] = 'MinorProject.settings'

from django.db import connection
import matplotlib.pyplot as plt
def fetch_records():
    with connection.cursor() as cursor:
        # Execute a raw SQL query to fetch records data
        cursor.execute("SELECT * FROM website_records")

        # Fetch all rows from the result set
        records = cursor.fetchall()
        print(type(records))
        # print(records)

        # Process the records data
        for record in records:
            print(record)

def fetch_products():
    with connection.cursor() as cursor:
        # Execute a raw SQL query to fetch products data
        cursor.execute("SELECT * FROM website_product")

        # Fetch all rows from the result set
        products = cursor.fetchall()
```

Fig 4.4

4.5) Data Visualization

Data visualization is the presentation of data in graphical or visual formats to aid in understanding complex patterns, trends, and relationships within datasets. It leverages charts, graphs, and other visual elements to communicate information effectively. Data visualization is essential for making data more accessible and actionable, allowing users to grasp insights quickly and make informed decisions.

We wrote code for data visualization using matplotlib in analysis.py

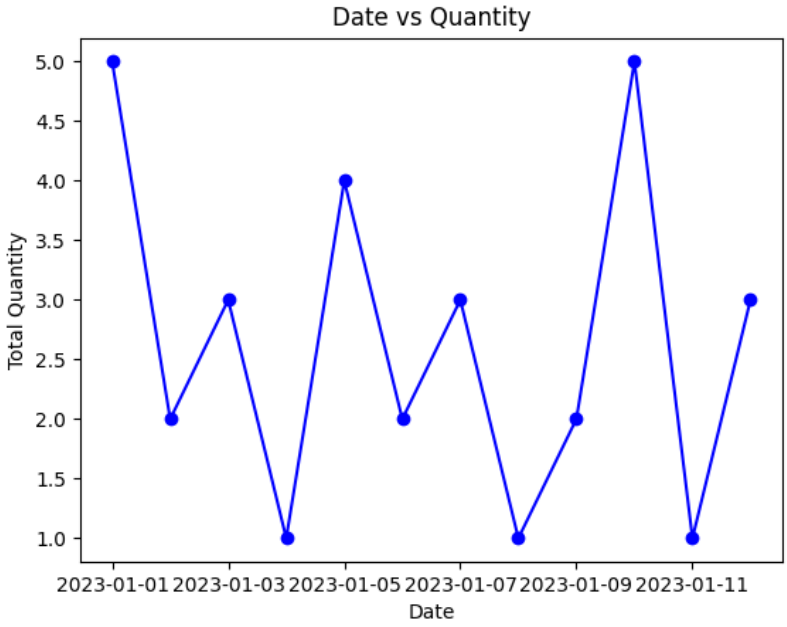

Fig 4.5.1

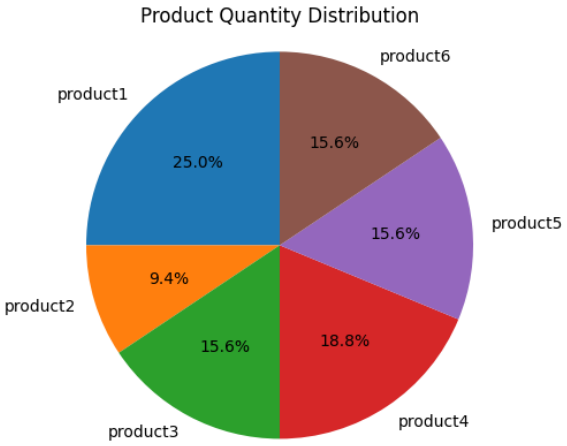

Fig 4.5.2

4.6) User Interface

User Interface (UI) refers to the visual elements and interactive components through which users interact with a software application or system. It includes design elements such as buttons, menus, and layouts, aiming to create an intuitive and user-friendly experience. UI design focuses on enhancing usability and accessibility, ensuring that users can navigate and interact with the application seamlessly. The primary goal is to provide a visually appealing and efficient interface that enhances the overall user experience.

In Template folder we created our HTML pages and in Static folder we created CSS files.

HTML:

MinorProject > Templates > base.html > html > body > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css" integrity="

    <title>{% block title %}  {% endblock title %}</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <a class="navbar-brand" href="/">Minor Project</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link" href="/">Home <span class="sr-only">(current)</span></a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/about">About</a>
          </li>

          <li class="nav-item">
            <a class="nav-link" href="/contact">Contact us</a>
```

Fig 4.6.1

CSS:

```
MinorProject > Static > # style.css > ...
1   body {
2       background: #343a40;
3       /* color: #fff; */
4       color: #0FFCF0;
5
6       font-family: Arial, sans-serif;
7       margin: 0;
8       padding: 0;
9   }
10
11
12  h1 {
13      /* color: white; */
14      color: #0FFCF0;
15      text-align: center;
16      padding-bottom: 50px;
17  }
18
19  p {
20      margin-bottom: 15px;
21  }
22
23  /* Add styles to limit the max-width of the images */
24  .card-container {
25      /* margin-top: 5px; */
26      display: flex;
27      justify-content: space-between;
28  }
29
30  /* Card styles */
```

Fig 4.6.2

JavaScript:

```
<script>
    $(document).ready(function(){
        var productCount = 1;

        // Function to add product fields dynamically
        function addProductFields() {
            var productFieldsHtml = `
                <div class="mt-3">
                    <label for="product${productCount}">Product ${productCount}:</label>
                    <input type="text" class="form-control" id="product${productCount}" name="product${productCou
                    <label for="quantity${productCount}">Quantity:</label>
                    <input type="number" class="form-control" id="quantity${productCount}" name="quantity${produc
                    <label for="unit${productCount}">Unit:</label>
                    <select class="form-control" id="unit${productCount}" name="unit${productCount}" required>
                        <option value="L">L</option>
                        <option value="Kg">Kg</option>
                        <option value="Box">Box</option>
                    </select>
                </div>
            `;
            $('#productFields').append(productFieldsHtml);
            productCount++;
        }

        // Event listener for adding product fields
        $('#addProductBtn').click(function() {
            addProductFields();
        });
```

Fig 4.6.3

CHAPTER 5: TECHNIQUES AND TOOLS

Technologies and tools refer to the various software, hardware, and other resources that are used to accomplish a specific task or achieve a particular goal. They are often used interchangeably, but generally, technologies refer to the broader range of resources and methods used to achieve a specific outcome, while tools refer to specific software applications, hardware devices, or other resources used to carry out specific tasks.

5.1 Technologies used:

- **Python**

  Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy-tolearn syntax emphasizes readability and therefore reduces the cost of program maintenance.

- **Libraries**
  - **Pandas**

    Panda is an open-source library designed primarily for working quickly and logically with relational or labeled data. It offers a range of data structures and procedures for working with time series and numerical data. The NumPy library serves as the foundation for this library. Pandas is quick and offers its users exceptional performance & productivity.

  - **Matplotlib**

    Matplotlib is a widely used Python library for creating visualizations, including charts, graphs, and plots. It provides a wide range of tools for data visualization, making it a popular choice in the field of machine learning (ML). Matplotlib supports a wide range of plots, including line plots, scatter plots, bar plots, histograms, and more.

- **Backend or Server**
  - In the development of the CRM Analysis System, Django stands as the backbone of our backend infrastructure, providing a robust and scalable framework for efficient data management and user interactions. Django, a high-level Python web framework, was chosen for its versatility, speed, and adherence to best practices in web development.

  - The heart of our backend lies in Django's Model-View-Controller (MVC) architecture, where models define the structure of our database, views handle the presentation and user interface logic, and controllers manage the flow of data between the model and view. This modular approach ensures a clear and organized codebase, facilitating easier maintenance and future scalability.

- One of Django's strengths is its Object-Relational Mapping (ORM) system, which seamlessly translates between our database schema and Python objects. This simplifies database interactions, making it easier to store and retrieve customer details, product information, and purchase records.

- Furthermore, Django's built-in admin panel streamlines the management of our database, allowing for easy addition, modification, and deletion of data. This admin interface not only enhances the efficiency of our development process but also ensures that the system remains user-friendly for administrators.

- Our use of Django extends beyond data management. It forms the foundation for implementing user authentication, securing our system against unauthorized access. Additionally, Django's REST framework facilitates the creation of a robust Application Programming Interface (API), enabling seamless communication between the frontend and backend components of our web application.

- In conclusion, Django serves as the linchpin of the CRM Analysis System's backend, providing a reliable and flexible framework that empowers our project with efficient data handling, user authentication, and API functionality. Its integration reinforces our commitment to a streamlined, scalable, and secure web application tailored to meet the dynamic needs of modern businesses.

- **Frontend or User Interface:**
  - **HTML:**

    HTML (Hypertext Markup Language) is a markup language used for creating web pages and other information that can be displayed in a web browser. HTML is the backbone of the World Wide Web and is used to structure and display content such as text, images, and multimedia on websites.HTML consists of a series of elements that define the structure of a web page.
  - **CSS:**

    CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in HTML (Hypertext Markup Language). CSS is used to define the visual appearance of a web page, including its layout, colours, fonts, and other styling options.CSS works by separating the content of a web page from its presentation
  - **JavaScript:**

    JavaScript (JS) is a programming language used to create dynamic and interactive web pages. It is widely used in web development for creating client-side scripts that run in web browsers, and server-side scripts that run on web servers. JavaScript is a high-level, object-oriented

programming language that supports eventdriven, functional, and imperative programming styles.

5.2 Tools used:

VS code: Visual Studio Code, often abbreviated as VS Code, is a free, open-source code editor developed by Microsoft. It is designed for building and debugging modern web and cloud applications and supports a wide range of programming languages, including Python, JavaScript, and TypeScript. VS Code provides a range of features, including syntax highlighting, code completion, and debugging tools.

CHAPTER 6: CONCLUSION

In the culmination of the CRM Analysis System project, we have successfully crafted a dynamic and user-friendly solution designed to propel businesses into the realm of data-driven decision-making. Our journey began with a fundamental goal: to streamline data management and empower users with meaningful insights. Through meticulous implementation and strategic choices, we leveraged the power of Django as the backend framework, providing a solid foundation for our system's architecture.

The Analysis page, adorned with dynamic graphs, has emerged as a visual powerhouse, unraveling key business trends at a glance. This visual representation, coupled with flexible analytics and interactive reports, affords users the agility to adapt their analyses based on evolving business needs.

Our commitment to user-friendliness is evident in every facet of the system, from the intuitive interface for inputting and retrieving data to the seamless integration of Django's ORM and REST framework for efficient data handling and API functionality.

As we reflect on the project's journey, it becomes clear that the CRM Analysis System is not merely a technological solution but a catalyst for transformative business strategies. By deepening the understanding of customer behaviour, regional sales patterns, and product popularity, our system equips businesses with the tools needed to optimize CRM strategies and foster sustainable growth.

In conclusion, the CRM Analysis System is not just a project; it's a testament to the potential of data in steering businesses towards success. With a blend of intuitive design, dynamic analytics, and the robustness of Django, we present a tool poised to revolutionize how businesses harness the power of their data for strategic decision-making and long-term viability.

Reference

https://docs.python.ore/3

https://matplotlib.org/stable/index.html

https://stackoverflow.com/

https://pypi.org/project/wordeloud/

https://www.w3schools.com/

www.kaggle.com

https://colab.google/