COMPUTER SCIENCES

FINAL YEAR PROJECT

Titled As

SMART AND INTELLIGENT INVENTORY STORE

SUBMITTED BY

AREEJ ILYAS 11447

MUHAMMAD IBRAHIM 10857

Under the supervision of

DR NOMAN BAIG

KARACHI PAKISTAN

19/09/2023

TABLE OF CONTENT

Contents

Figure Table

Chapter 4

Chapter 5

Chapter 6

Chapter 7

ABSTRACT

This project is aimed at developing a web-based application named Smart and Intelligent Inventory Store for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate sales and inventory report daily or weekly based. This project is categorized individual aspects for the sales and inventory management system.

In this system we are solving different problems like managing the stocks, sales, purchases and their records, alerts while touching bottom line of stock, future stock management planning etc.

Without proper inventory control, a large retail store may run out of stock on an important item. In our project we are solving this problem. We will provide a feature that customer will see what inventory store needs in which situations.

ACKNOWLEDGEMENT

In the name of Allah, the most Gracious and the Most Merciful.

Peace and blessing of Allah be upon Prophet Muhammad ﷺ

First, praise of Allah, for giving us this opportunity, the strength and the patience to complete our FYP finally, after the challenges and difficulties. We would like to thank our supervisor ***NOMAN BAIG*** for his guidance, motivation and most his significant contribution in this project, expert and for giving us the opportunity to work on this project. We would also like to thanks our parents for financial and moral support and our friends who have helped and motivated us throughout. May Allah reward them all abundantly Ameen

DEDICATION

This project is prepared in the partial fulfillment of the requirement for the degree of Bachelor in Computer Information System (BCIS). The satisfaction and success of completion of this task would be incomplete without heartfelt thanks to people whose constant guidance, support and encouragement made this work successful. On doing this undergraduate project we have been fortunate to have help, support and encouragement from many people we would like to acknowledge them for their cooperation.

Our first thanks goes to **KIET UNIVERSITY KARACHI** for designing such a worthy syllabus and preparing us for doing this project. Our next batch of thanks goes to the CoSIC Faculty of KIET without whose help our project would have been impossible. Our very sincere and heartfelt thanks go to Mr. Dr Noman Baig, our project supervisor who constantly guided us through the project time period. Without his guidance, our project would have been impossible. Last but not the least we want to thank every direct and indirect hands that were involved in completion of this project.

With Regards

Areej Ilyas (SID:11447)

Muhammad Ibrahim (SID:10857)

SEMESTER	FALL 2022	YEAR	2022-2023

TITLE OF PROPOSED PROJECT

SMART AND INTELLIGENT INVENTORY STORE

Project Category (choose one)	Product based

SUPERVISOR INFORMATION

Supervisor Name:	DR. NOMAN BAIG	Organization/ Designation	
Contact No:		email:	

STUDENT(S) INFORMATION

S#	Student ID	Name	
1	11447	Areej Ilyas	
Contact No:	03035845961	email:	8074.alpha@gmail.com
2	10857	Muhammad Ibrahim	
Contact No:	03000047075	email:	926289isa@gmail.com

PROJECT AREA/TOOLS

Tools Required:	Visual Studio SQL Server Visual Studio Code/ Jupyter
Area/Specialization:	AI(ML) and Data Science+ ASP.net

SUMMARY OF PROPOSED PROJECT (MAXIMUM 300 WORDS)

Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate inventory report daily or weekly based.

This purposed system will predict your sales using ML and create warnings seeing remaining stock and their expiry using data science in your inventory.

PROJECT OBJECTIVE(S)/OUTCOMES
To develop an application that deals with the day-to-day requirement of any organization and the easy management of the inventory. To provide details information about the stock balance and making the stock manageable and simplify the use of inventory in the organization

FUNCTIONAL FEATURES
Login/Signup/forget/Change Password Product Management Monitor and Categories and their Products Manage Product Pricing Sales and Stocks Management Monitor Purchasing Transaction Analysis

CHAPTER-1: INTRODUCTION

1.1 Introduction

A web-based application named Smart and Intelligent Inventory Store for managing the inventory system of any organization.

This web-based application is based on the management of stock of an organization/stores. The application contains general organization profile, sales details, Purchase details and the remaining stock that are presented in the organization.

There is a provision of updating the inventory also. This application also provides the remaining balance of the stock as well as the details of the balance of transaction. Each new stock is created and entitled with the named and the entry date of that stock and it can also be updating any time when required as per the transaction or the sales is returned in case.

1.2 Literature Review

Products are considered as the business resources for the organization. This includes managing the product with appropriate way to review any time as per the requirement. Therefore, it is important to have a computer based IMS which has the ability to generate reports, maintain the balance of the stock, details about the purchase and sales in the organization. Before developing this application, we came up with 2 Inventory Management System existing in the market, which helps to give the knowledge for the development of our project. This application software is only used by the large organization and there was no concepts of managing future stock problems but so we came up with the application which can be used by the small company for the management of their stock in the stores and stock houses and also we add a feature that can predict some future planning hints etc. After analyzing the other inventory management system, we decided to include some of common and key features that should be included in every inventory management system so we decided to include those things that help the small organization in a way or other.

1.2 Problem Statement

After analyzing many existing IMS, we have now the obvious vision of the project to be developed. Before we started to build our project, team had many challenges. We defined our problem statement as:

- **Overstocking**: This occurs when a company has an excessive amount of inventory on hand, which can result in increased storage costs, spoilage, and other issues.
- **Understocking:** When a company does not have enough inventory to meet demand, it can result in lost sales and customer dissatisfaction.
- **Stockouts:** When a company runs out of a specific item, it can result in lost sales and customer dissatisfaction.
- **Excess inventory:** This occurs when a company has an excessive amount of inventory of a particular item that is not selling, resulting in higher storage costs and lower profits.

1.3 Purposed System

1.3.1 Features

Inventory management systems are computer-based applications that assist businesses in tracking, managing, and organizing their inventory. Inventory management systems have the following features:

- **Product cataloguing:** The ability to create a digital catalogue of all inventory products, including product details such as name, description.
- **Stock tracking** refers to the ability to track and monitor the quantity of each product in inventory as well as the location of the products within the warehouse.
- **Order processing** entails the ability to create and manage orders, as well as generate invoices.
- **Purchase order management** entails the ability to create as well as receiving and processing vendor invoices.
- **Sales management system** includes all sales history make sales and view records.
- **Reporting and analysis:** The ability to generate reports and analyses inventory data such as sales data, stock levels, and supplier performance.

1.3.2 Scope

Inventory management is the process of tracking and managing a company's stock of goods, materials, and products. Overall, inventory management aims to keep a sufficient supply of goods on hand to meet customer demand while minimizing the costs associated with holding excess or obsolete inventory.

Chapter No 2. Requirements Analysis

2.1 IMS Requirements

We began our investigation by determining the organization's need for IMS. Initially, we limited our research to the general reasons that an Inventory Management System was required. We used various techniques to collect data that can clearly show us the overall picture of the application. Interviewing developers, visiting online websites that are presented as templates, and visiting some organizations to see their IMS application were the techniques we used.

The application's goal is to manage the organization's inventory management function. Once automated, all functions can be managed effectively and the organization can gain a competitive advantage. The following additional details are provided in the Scope section to supplement the discussion of business requirements:

- Helps to search the specific product and remaining stock.
- Details information about the product sales and purchase.
- Brief Information of the organization today's status in terms of news, number of present inventories as per the date entered.
- It helps to identify the total presented inventory in the company
- To know the balance and details of sales distributed in specific date.
- There is proper transaction management of inventory.
- All transactions have specific entry date along with quantity and rate.
- Only admin can sign-up and new users in the page

2.2 User Requirements

User requirements are categorized by user type:

2.2.1 Admin Requirements

- Able to create new sales and purchase entries along with date.
- Able to edit the entry as per entry.
- Able to add, modify and delete the stock entry.
- Able to add new employees
- Able to see all the daily activities by individual employee.

2.2.2 Employee Requirements

- Able to create new sales and purchase entries along with date.
- Able to add new products, categories and subcategories along with their details.
- Able to check available stocks.
- Able to change password.

Chapter No 3. Methodology

3.1 Technologies Used

3.1.1 Database

A database is a collection of data that has been structured in a way that allows it to be easily accessed, managed, and updated. An inventory system is a system for managing and tracking the inventory of goods or materials. A database can be used to store and manage inventory system data.

3.1.1.1 Relational Database

A relational database is a type of database in which data is stored and organized in tables. Tables are made up of rows and columns, where each row represents a distinct record and each column represents a specific piece of data within that record.

A relational database may be used in an inventory system to store information about the items being tracked, such as their names, descriptions, quantities on hand, and locations. It could also be used to keep track of suppliers, orders, and customers.

One of the primary benefits of using a relational database for an inventory system is the ease with which related pieces of data can be linked and relationships formed.

3.1.1.2 Primary Keys

A primary key in a database is a column or set of columns that uniquely identifies each row in a table. It is a method of uniquely identifying each row in a table, allowing it to be referred to and accessed in a database. Primary keys can be composed of a single column or several columns. A composite key is a primary key that is made up of multiple columns.

Here are some properties of primary keys:

- They must contain unique values.
- They cannot contain null values.
- They cannot be changed.

3.1.1.3 Foreign Keys

A foreign key is a field in a database that refers to the primary key of another table. The foreign key's purpose is to ensure the data's referential integrity. Foreign keys are an important tool for ensuring a database's integrity and consistency. They enable you to enforce table relationships and ensure that data is entered and organized correctly.

3.1.1.4. Unique Constraints

A unique constraint is a rule that prevents duplicate values in a column or set of columns in a database table. When a unique constraint is defined on a column or set of columns, all values in those columns must be unique across all rows in the table. If an attempt is made to insert a duplicate value into a column with a unique constraint, the database will return an error and the insert will be rejected.

Here are some examples of when you might use a unique constraint:

- You have a customer_id column in a customer's table and you want to ensure that each customer has a unique ID.
- You have a username column in a user's table and you want to ensure that each user has a unique username.

3.2.1 .Net Framework/ASP.Net

Microsoft created the.NET framework as a software development platform. It comes witha large library of coded solutions to common programming problems, as well as a virtual machine that manages the execution of programs written specifically for the.NET framework. ASP.NET is a web development framework included in the.NET framework.It is used to create web applications and allows you to write server-side code in languages such as C#.

3.2.2 CSS/HTML

HTML (Hypertext Markup Language) is a markup language that is commonly used to create web pages. It allows you to structure and format web content, such as headings, paragraphs, lists, and links. CSS (Cascading Style Sheets) is a stylesheet language used to describe the appearance and formatting of an HTML document. It is used to specify things like colours, font styles, and layout and is used to control the appearance of web pages.

3.2.3 Ajax/J-Query/Java-Script

AJAX (Asynchronous JavaScript and XML) is a web application development technique that allows a web page to communicate with a server and update itself without having to reload the entire page. It operates by combining JavaScript and XML (a markup language for structured data).

This is how it works: AJAX requests are sent by a web page to a server, instructing it to perform a specific task or retrieve data. The server processes the request and returns a response to the web page, which can then update itself without having to reload the entire page. This enables the development of web applications that are faster, more responsive, and more interactive.

AJAX has evolved into an important tool for developing modern web applications, and it is now widely used in a variety of contexts such as social networking, online shopping, and data visualization.

3.2.4 Python

Python is a widely used programming language in the fields of artificial intelligence and machine learning. Python is well-suited for these tasks for several reasons:

Python has a large and active developer community, which means there are many open-source libraries and frameworks available for AI and ML projects.

Python's syntax is simple and readable, making it easier to learn and use than other programming languages.

Following is list of pythons used in our project:

3.2.5 Flask API

Flask is a lightweight Python web framework with useful tools and features for developing web applications. It is classified as a microframework because it does not necessitate the use of any specific tools or libraries. It has a small and easily extensible core.

Flask's support for developing APIs is one of its key features (Application Programming Interfaces). An API is a set of rules that defines how two software systems should communicate with one another. A Flask API is a lightweight Python web framework that provides tools and features for developing APIs.

3.3 Development Tools

3.3.1 Visual Studio

Microsoft Visual Studio is a comprehensive software development suite that includes a wide range of tools and features for developing web, cloud, and mobile applications. It includes an integrated development environment (IDE), which allows developers to write, debug, and deploy code in a single location.

Some features of Visual Studio include:

- Code editing and debugging tools
- Support for a wide range of programming languages, including C++, C#, and Visual Basic
- A variety of project templates for creating different types of applications, such as web applications, mobile apps, and console applications
- Integration with version control systems such as Git
- Tools for building and deploying applications to the cloud

3.3.2 My SQL Server

Microsoft SQL Server is a relational database management system (RDBMS) that Microsoft developed and marketed. It is commonly used in conjunction with web-based applications to store and manage data used by other applications.

SQL Server is compatible with a variety of programming languages and development frameworks, including.NET, Java, PHP, and Python. It can store data in various formats, including structured, semi-structured, and unstructured data.

Some features of SQL Server include:

- Support for transactions and data integrity
- Support for stored procedures and triggers
- Data security and encryption
- Scalability and high availability
- Support for data warehousing and business intelligence

3.3.3 Jupyter

Jupyter is a free and open-source web app that lets you create and share documents with live code, equations, visualizations, and narrative text. It is frequently employed in data analysis, machine learning, and scientific computing.

Jupyter notebooks are interactive, which means you can run code and see the results immediately. Because they allow users to explore and experiment with data and models, they are a popular choice for data science and machine learning tasks.

Jupyter is a powerful tool that is widely used in the data science and machine learning communities, as well as anyone who works with data or code.

Chapter No 4. System Design

4.1 Use-Case Diagram

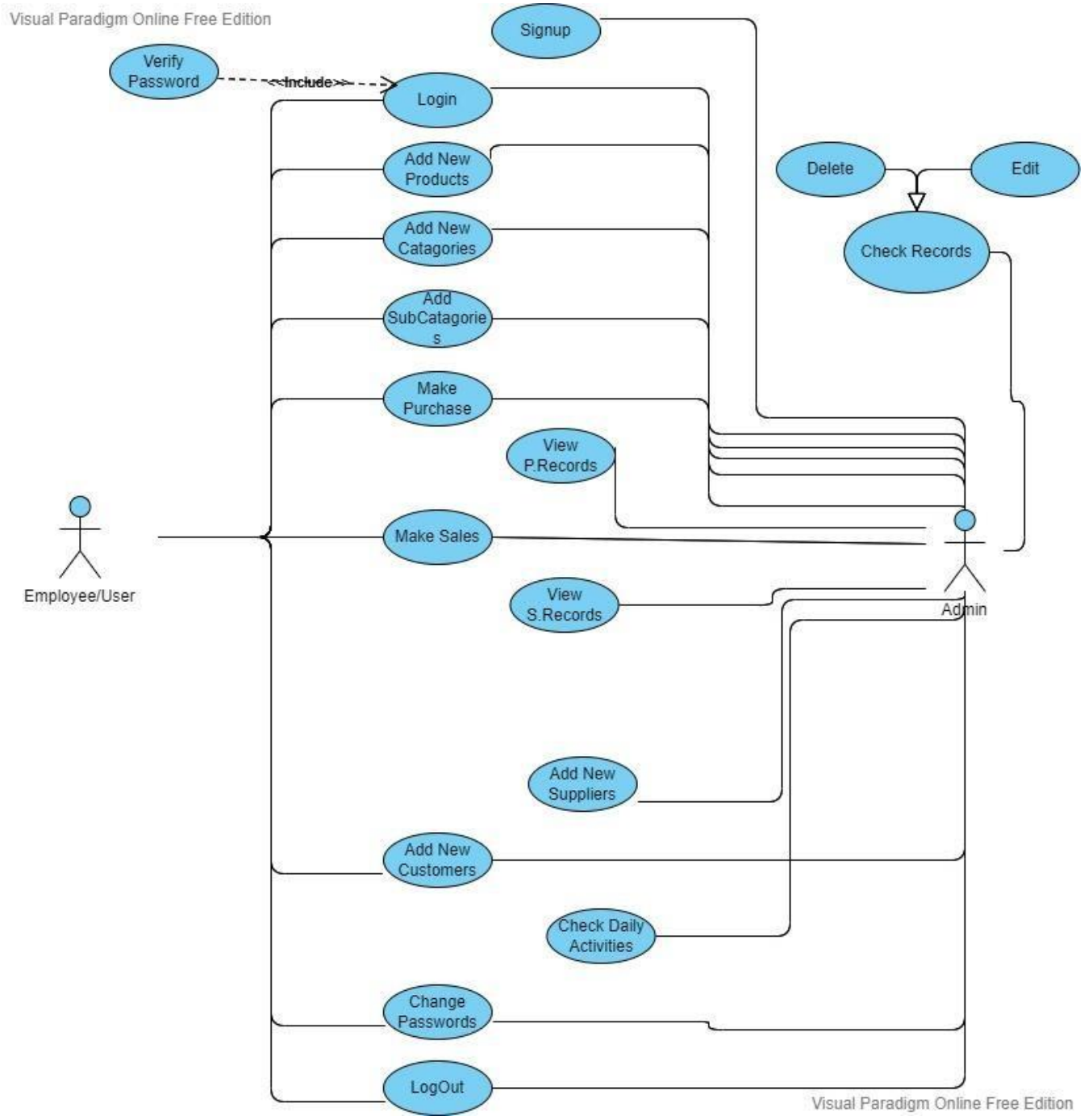


Figure 4. 1 Use Case Diagram

4.2 Sequence Diagram

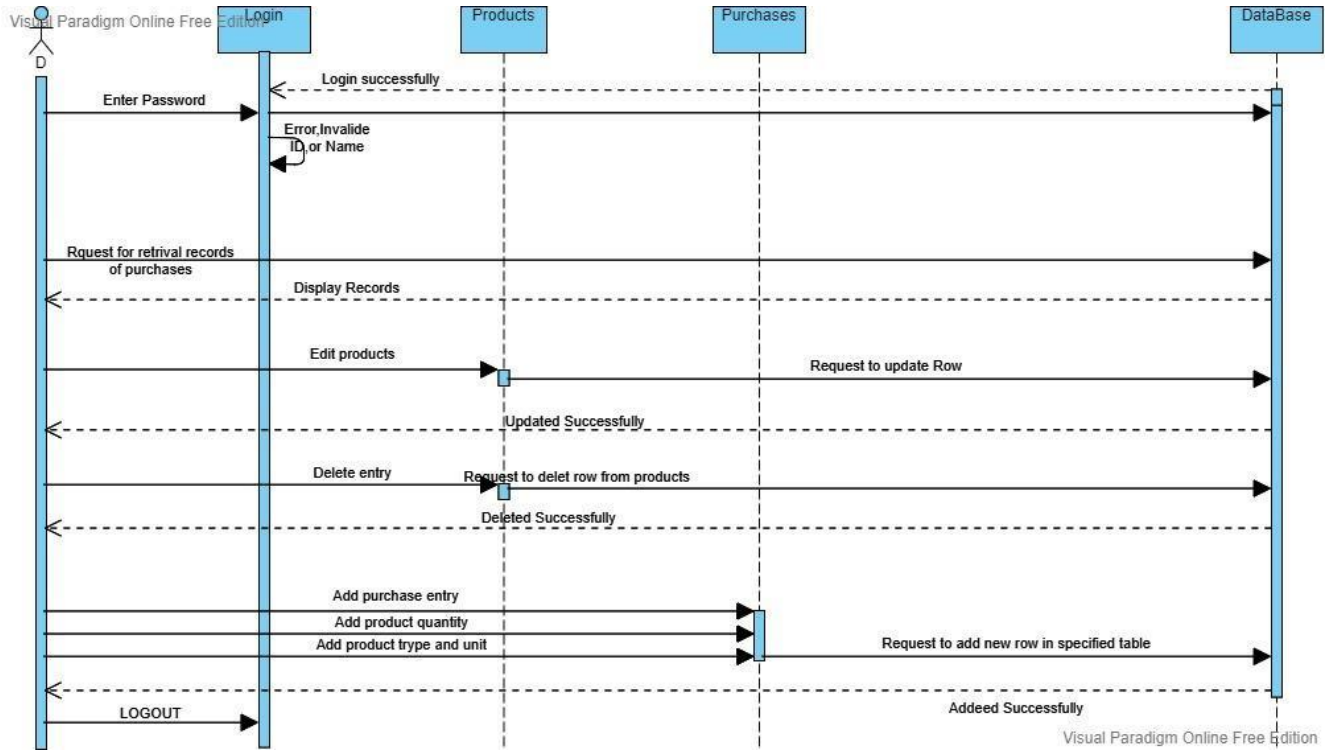


Figure 4. 2 Sequence Diagram

4.3 DATABASE MODEL

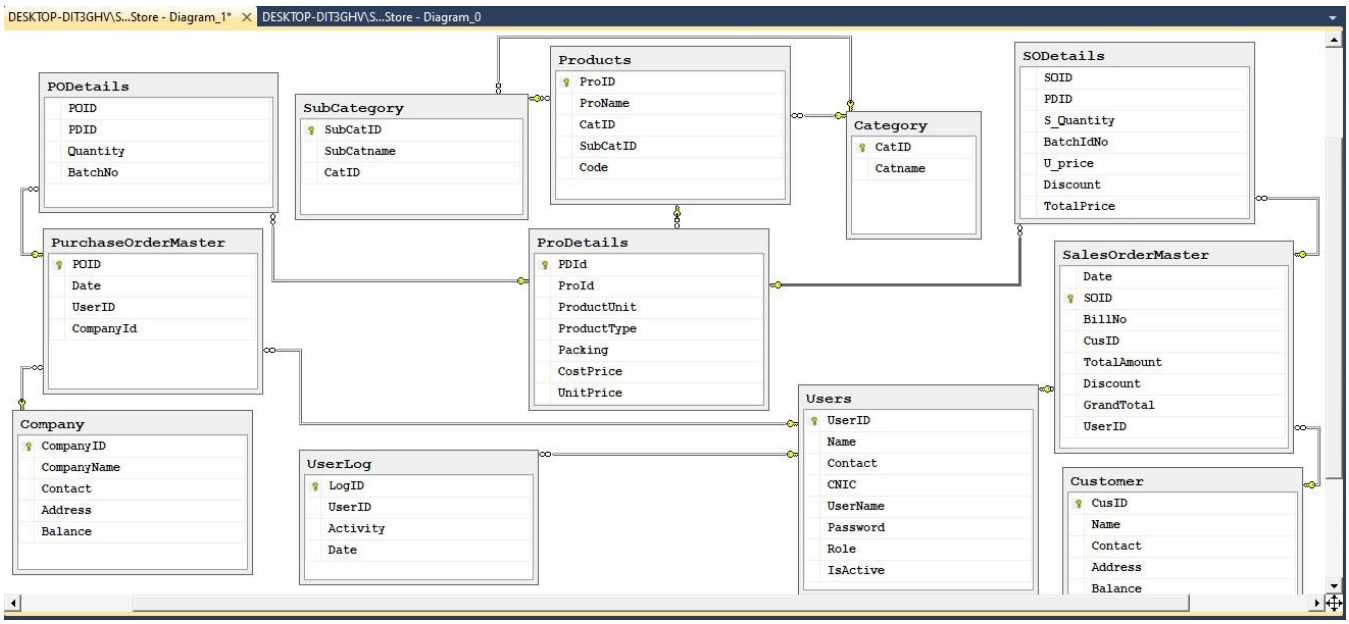


Figure 4. 3 Database Model

4.4 Activity Diagram

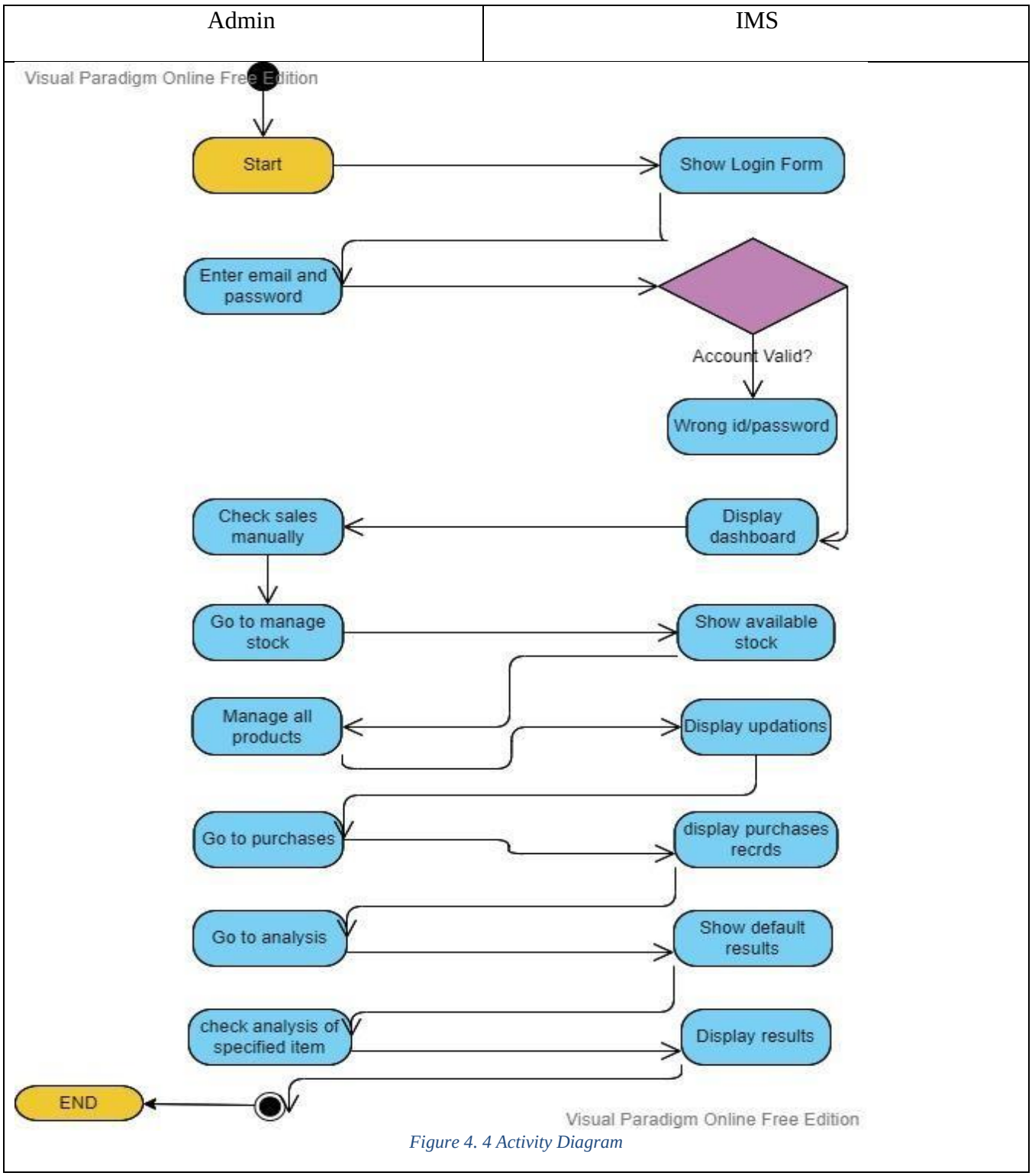


Figure 4. 4 Activity Diagram

Chapter no 5.

5.1 Project planning

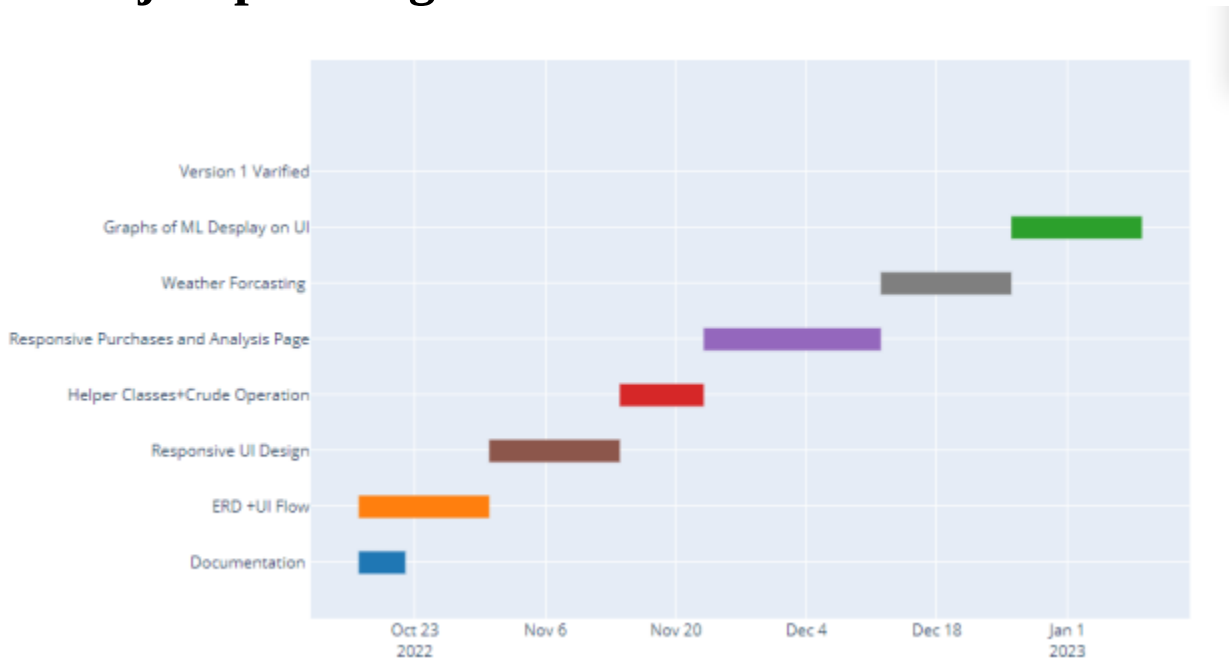


Figure 5. 1 Planning Version 1

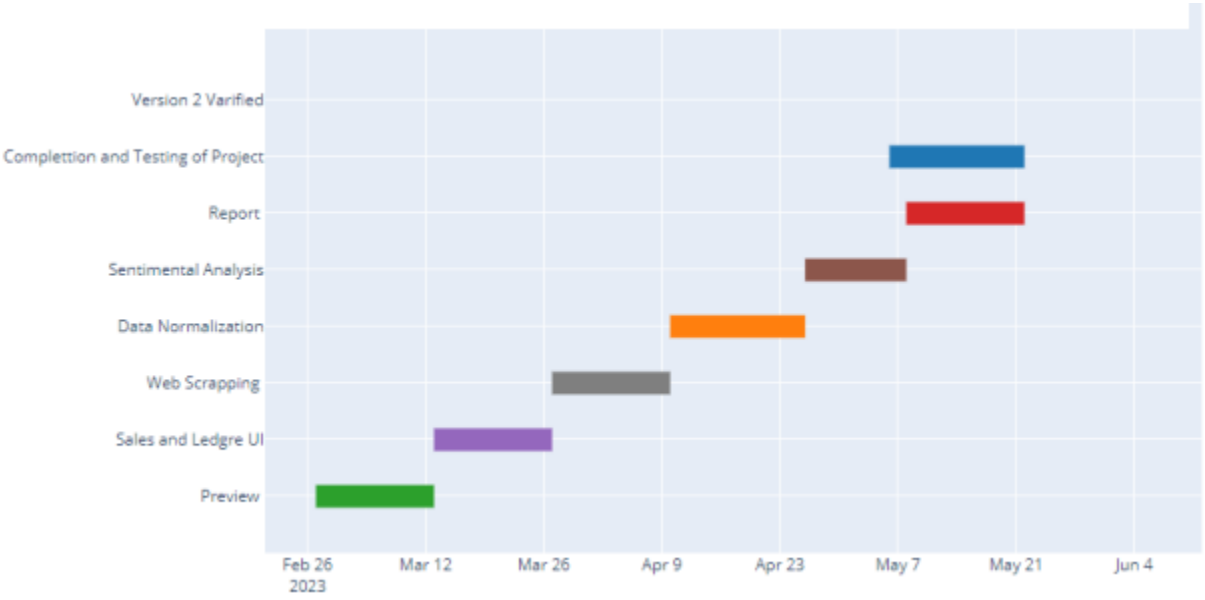


Figure 5. 2 Planning Version 2

5.2 Test Cases

TEST CASE # 1 LOGIN

Test Case Title:	*Login with Right Passwords and Id*
Preconditions:	Employee or admin is logging to personal account
Actions:	Enter correct id and Password
Expected Results:	Dashboard Displayed
Tested By:	Areej Ilyas
Result:	Pass

Test Case Title:	*Login with Wrong Password*
Preconditions:	Employee or admin going to login personal account.
Actions:	Enter correct id but wrong password
Expected Results:	Error displayed about wrong password
Tested By:	Areej Ilyas
Result:	Pass

TEST CASE # 2 SIGN-UP

Test Case Title:	*Sign-up with correct information*
Preconditions:	User is going to create his account as admin
Actions:	Enter information like first and last name, username, password, phone no, and cnic in correct format.
Expected Results:	Displayed dashboard
Tested By:	Areej Ilyas
Result:	Pass

Test Case Title:	*Sign-up with correct information but wrong phone no format*
Preconditions:	User is going to create his account as admin
Actions:	Enter correct information as first and last name, username, password and cnic in correct format but phone no in wrong format.
Expected Results:	Displayed and error that phone no is not in correct format.
Tested By:	Areej Ilyas
Result:	Pass

Chapter No 6. GUI

6.1 Login

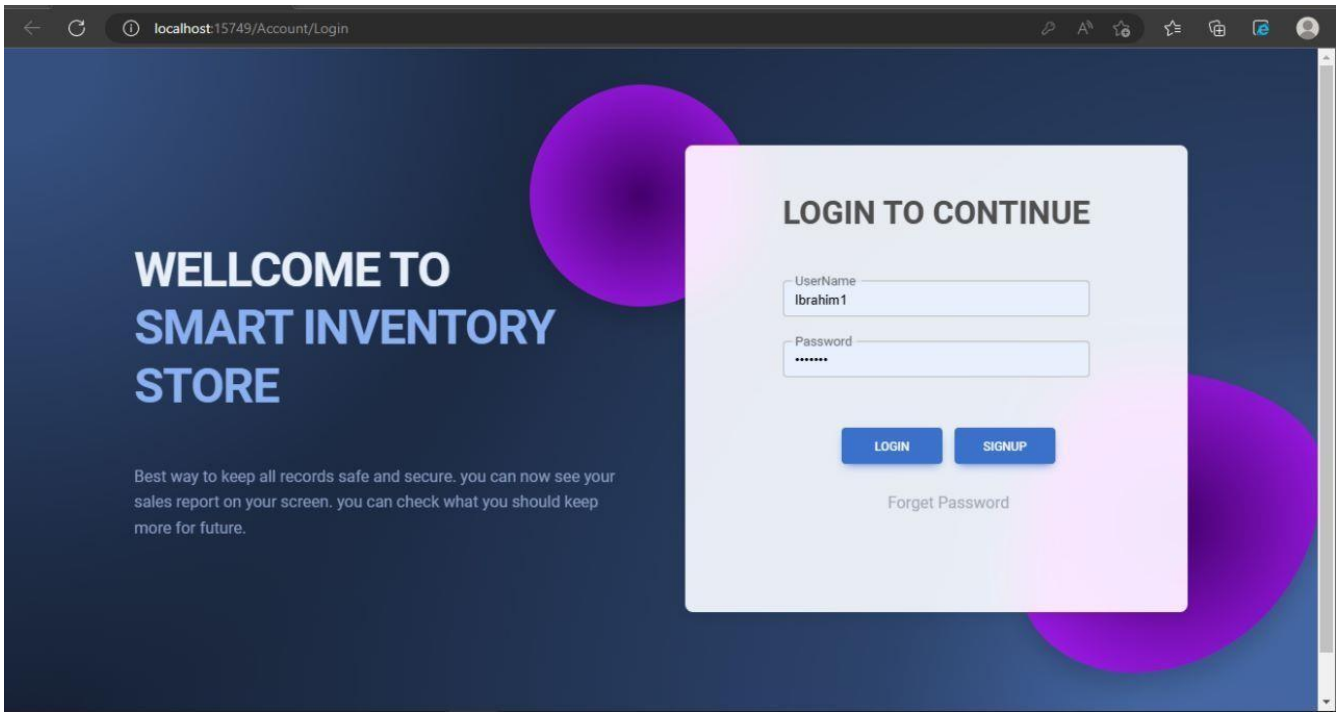


Figure 6. 1 Login Form

6.2 Sign-up

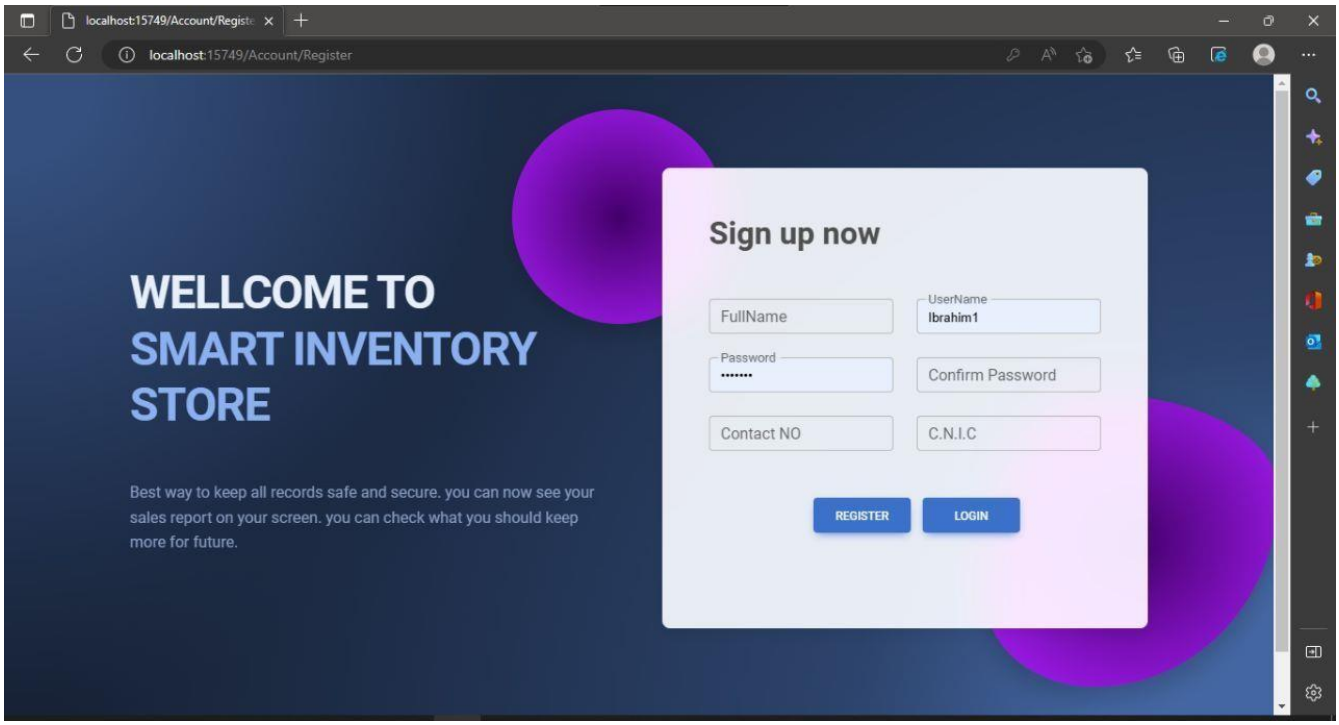


Figure 6. 2 Signup Form

6.3 Dashboard

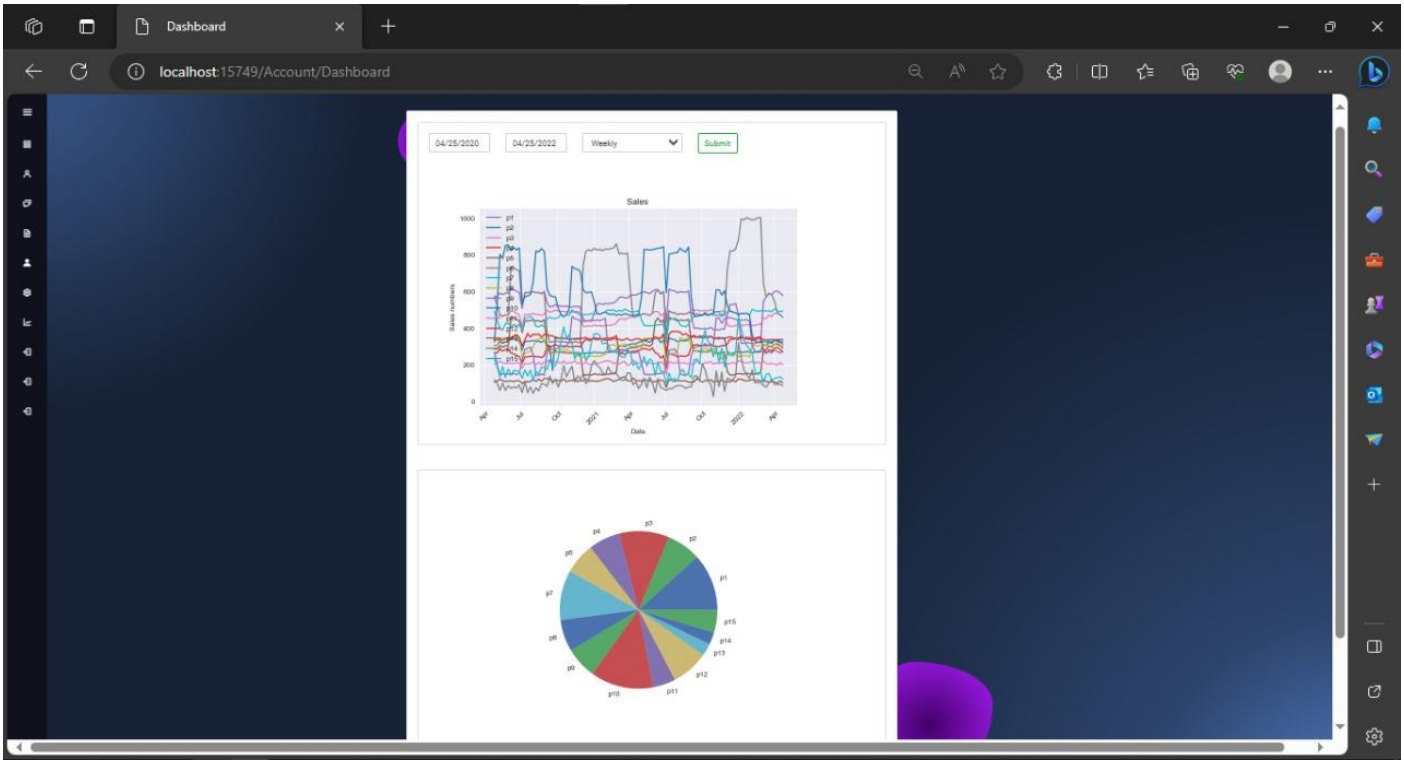


Figure 6. 3 Dashboard(1)

6.4 Manage-Stock

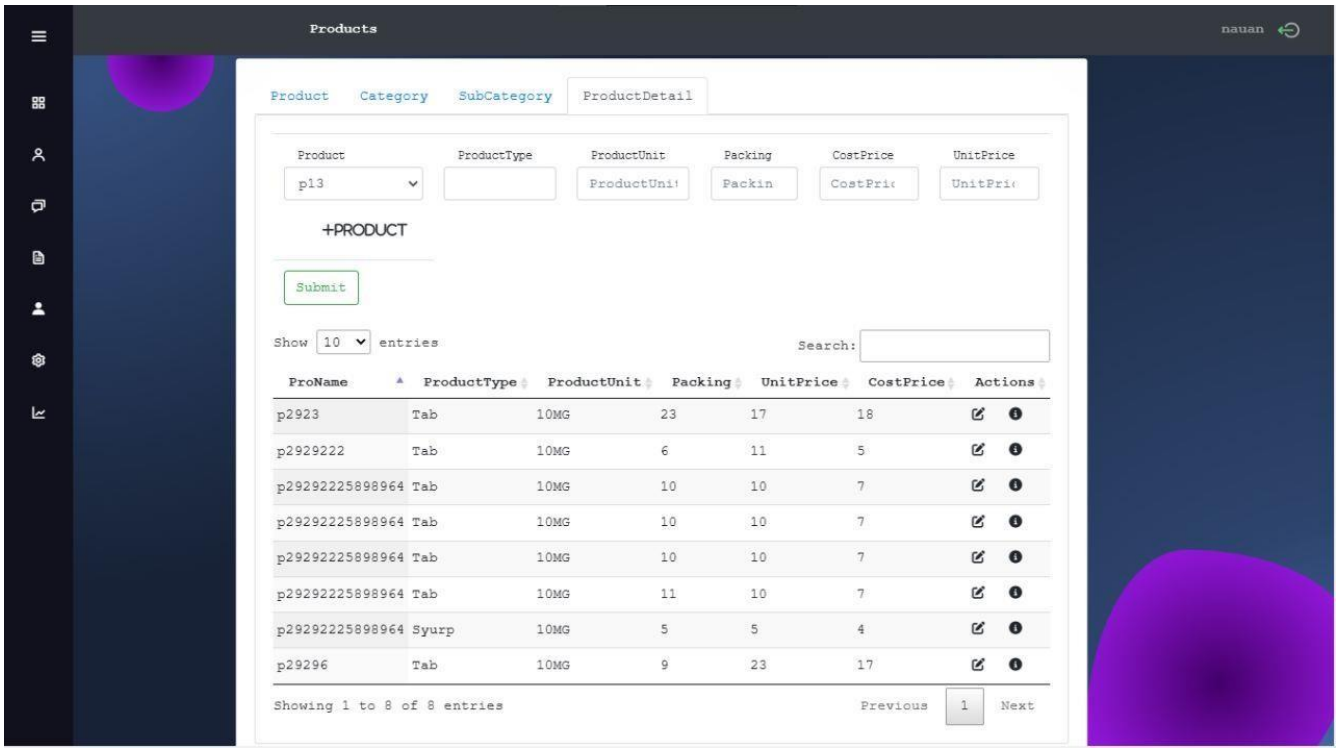


Figure 6. 4 Manage Stock (1)

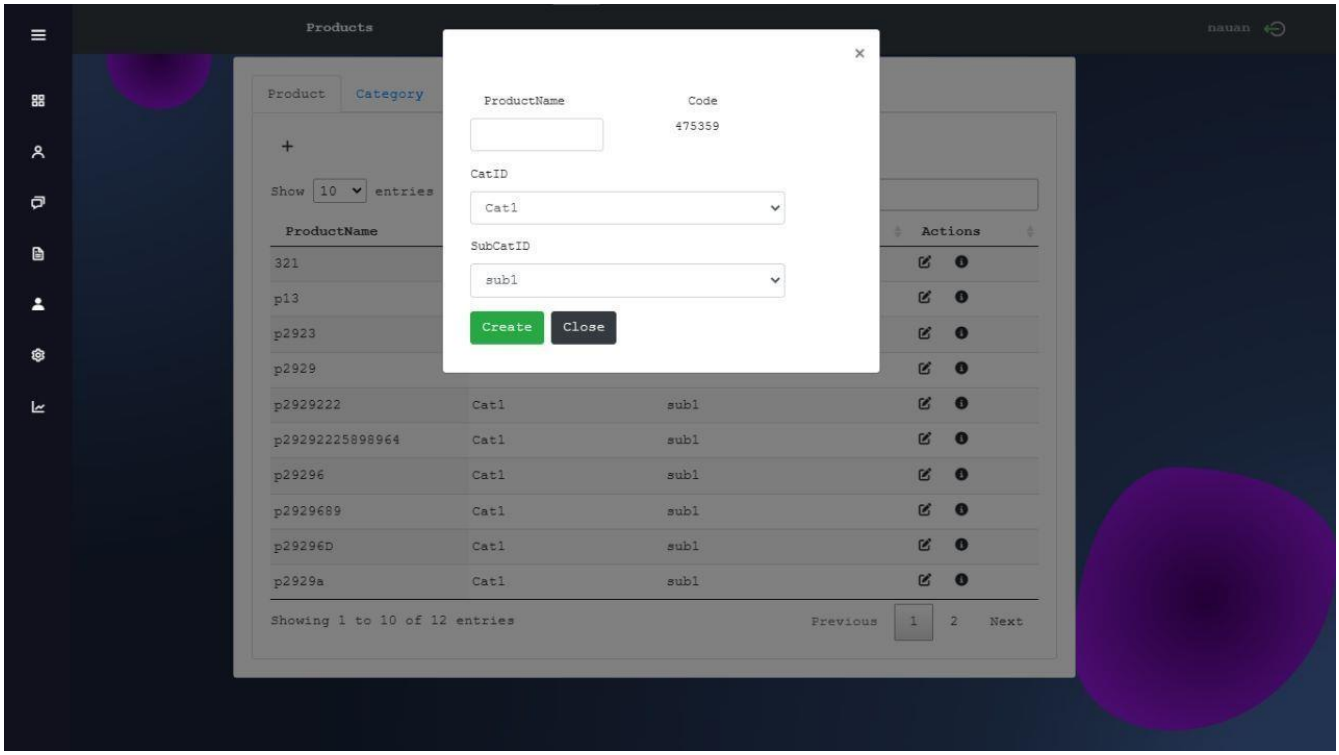


Figure 6. 4 Manage Stock (2)

6.5 Purchases

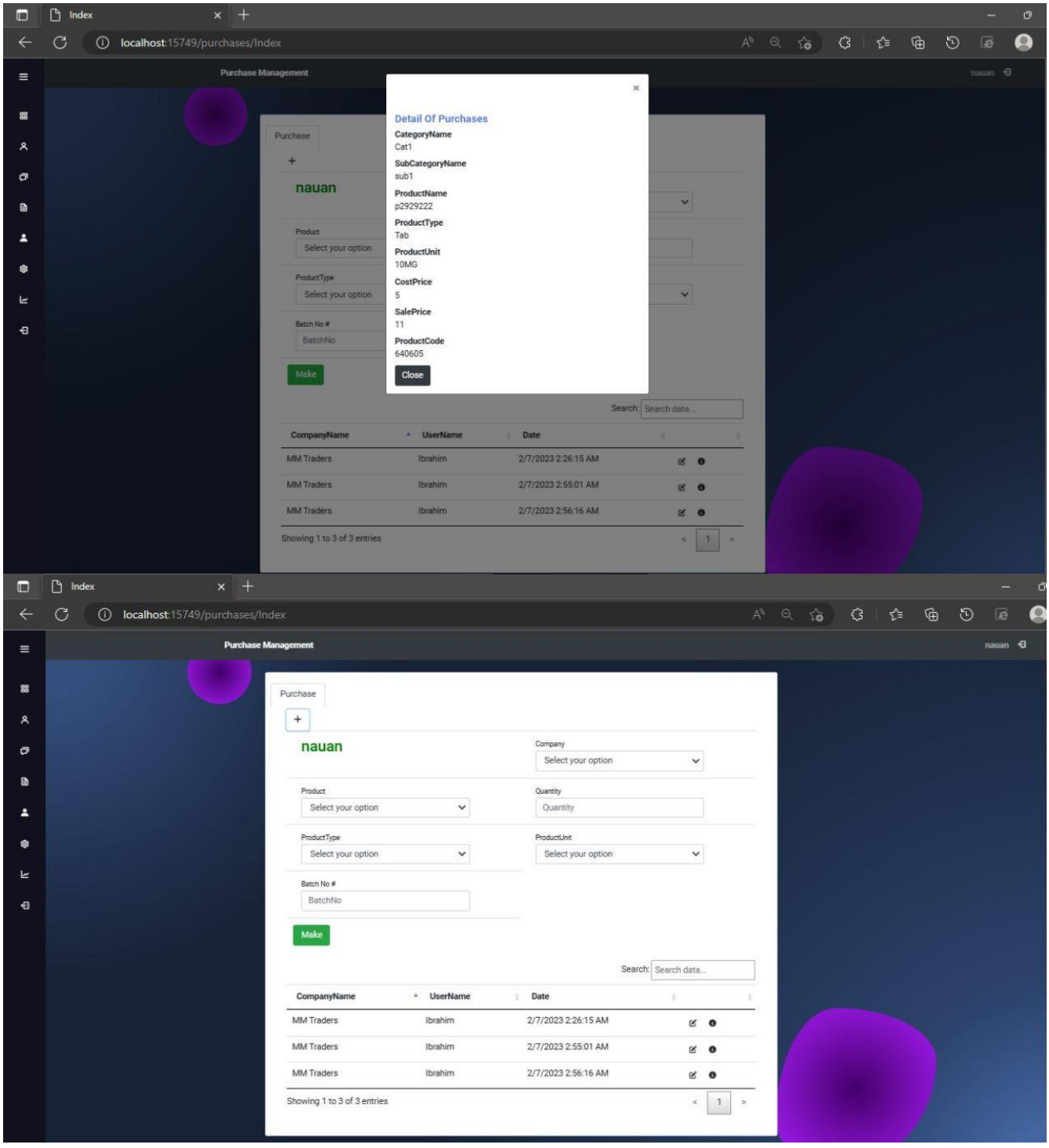


Figure 6. 5 Purchase Management

6.7 Sales

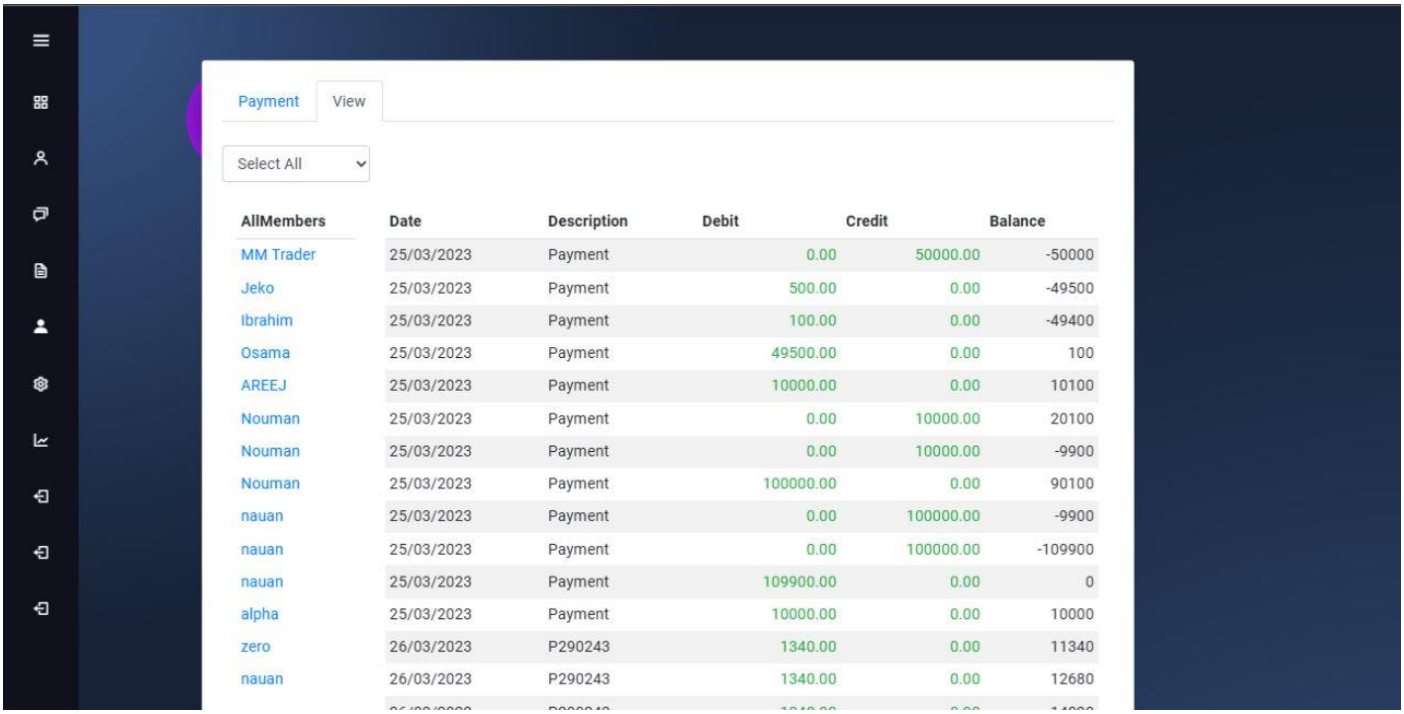


Figure 6. 6 Sales Management

6.8 Setting

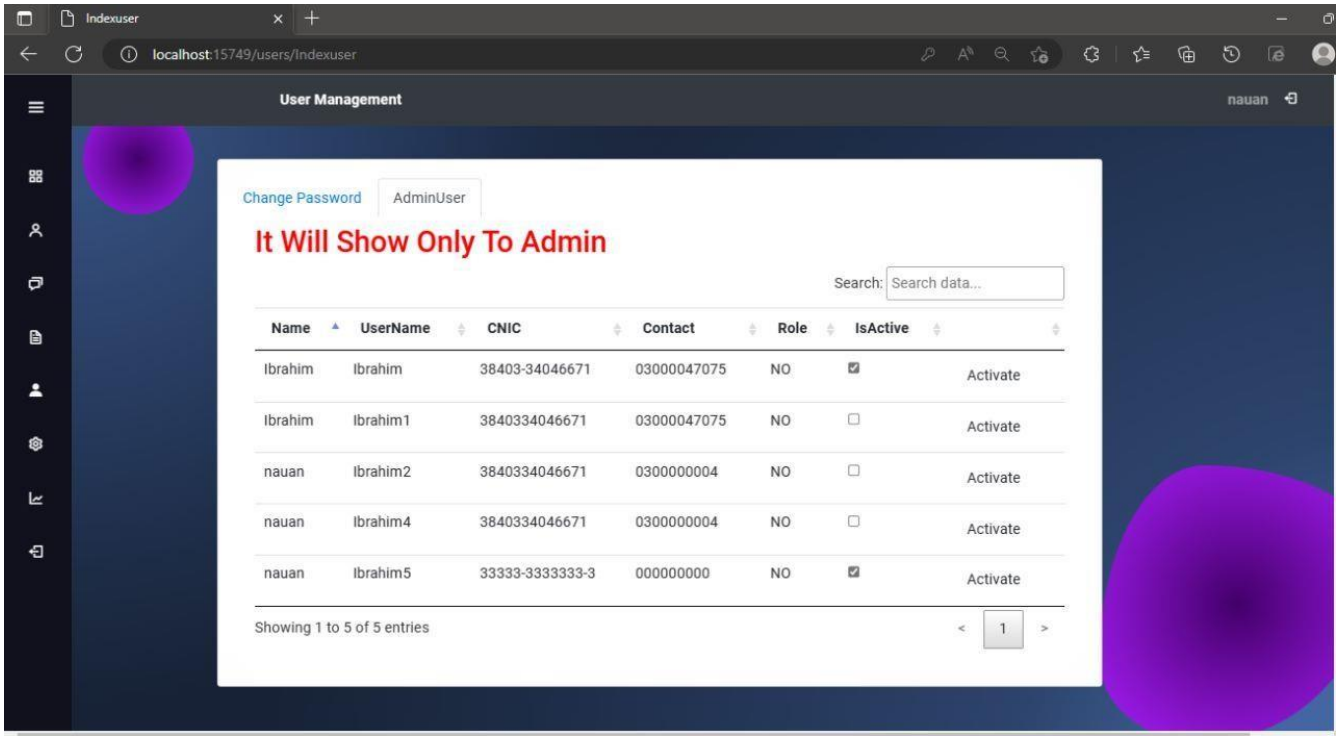


Figure 6. 7 Setting

6.9 Analysis

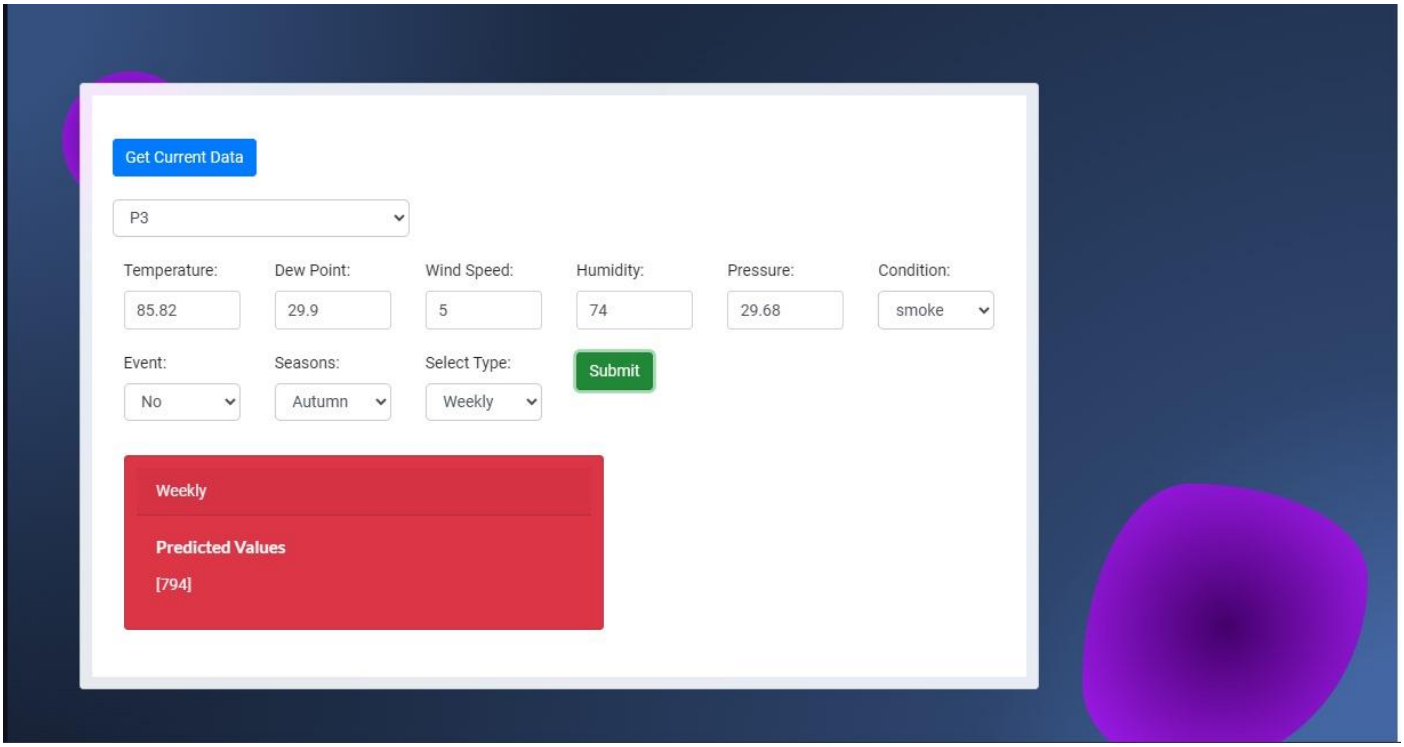


Figure 6. 8 Analysis

6.10 Ledger

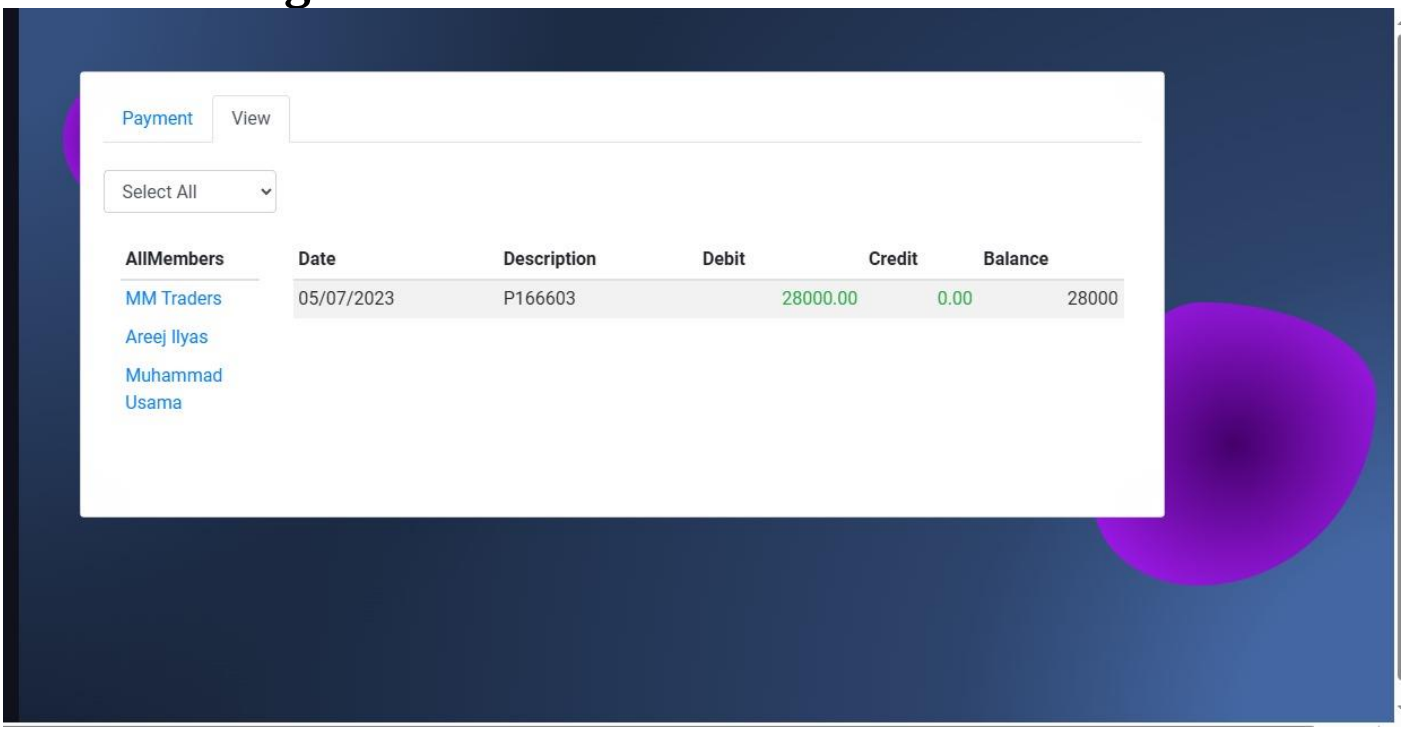


Figure 6. 9 Ledger System

Chapter No 7. Finding & Results

7.1 Data-Set

A dataset is a collection of data, typically in tabular form (rows and columns), used for a specific purpose such as analysis or machine learning. A dataset can be saved in a variety of formats, including CSV (Comma Separated Values) files and SQL databases. A dataset is commonly divided into two subsets: a training set for training a machine learning model and a test set for evaluating the model's performance.

With the help of this dataset, companies, producers, and retailers may better comprehend the connection between product demand and temperature and adjust their production plans accordingly.

7.1.1 Description

The dataset includes sales information for a particular product or product category coupled with historical temperature records. Insights into how temperature variations affect shopper behavior and purchasing habits are provided by the vast variety of temperature values provided across various geographic regions and time periods. Businesses can learn important information about how events and temperature affect sales by analyzing this dataset, which will help them plan their marketing, manufacturing, and inventory needs effectively.

7.1.2 Key Features of Dataset

Temperature Information: The dataset contains temperature readings that were regularly taken (hourly, daily, or monthly) at numerous sites. Instruments that are dependable and accurate are used to record temperature measurements, offering a wide range of temperature circumstances.

Sales Information: The dataset includes sales information for the particular product or product category under consideration, corresponding to each temperature record. These sales data points are an accurate reflection of customer demand as it was seen during the relevant temperature period.

Time-Stamped Data: Data that has been time-stamped allows for the investigation of temporal patterns and the effects of seasonality on sales based on temperature changes. Each data entry in the dataset has a timestamp attached to it.

Events Data: This dataset includes data on significant events or occurrences that may have an impact on consumer behavior and sales patterns in addition to temperature and sales statistics. Holidays, festivals, athletic events, and other noteworthy occurrences that have been found to affect purchasing decisions might all be included in this list. Businesses can

obtain insights into how these events interact with one another by incorporating this data.

7.1.3 Source

Weather Station: The temperature dataset utilized in this "Sales Forecasting Dataset for Future Production" was created by comparing weather station records with time-stamped sales data. Timestamped historical sales data were gathered from [http://www.kaggle.com], +while accurate weather stations in [certain locations/regions] provided the appropriate temperature data. A comprehensive temperature dataset was constructed by comparing the timestamps of sales data with the corresponding temperature readings, allowing the analysis of temperature's impact on sales trends.

Calendar: This "Sales Forecasting Dataset for Future Production" contains an events dataset that was created by matching timestamped sales data with key events on a calendar website. The calendar website offered a thorough listing of public holidays, celebrations, athletic events, and other pertinent occasions. We built an events dataset that depicts the impact of these factors on customer behavior and sales patterns by matching the timestamps from the sales data with the corresponding events on the calendar.

7.1.4 Data Collection

Data is collected using web scrapping technique for weather station and calendar.

7.1.4.1 Web-Scrapping

In order to gather temperature data from the indicated weather stations and pertinent event data from the specified calendar websites, we used web scraping techniques. For specified time periods, such as hourly, daily, or monthly readings, we have collected temperature values.

7.1.4.2 Data Integration

Combine the temperature data, sales data, and events dataset into a unified dataset. Merge the datasets using common fields such as timestamps or location identifiers. Ensure that the data is organized in a structured and coherent manner for further analysis and modeling.

7.1.5 Data Cleaning and Processing

7.1.5.1 Adjusting Entities

We found a difference between the frequency of the temperature and sales data in our dataset. The sales data was recorded as a single quantity every hour, whereas the temperature data was recorded as two entities per hour. We changed the temperature data by taking the average of the two temperature entries for each hour in order to align the two variables. This modification made sure that the number of entities per hour for the temperature data and the sales data were equal. By applying this correction, we were able to attain data consistency and accurately model and analyses the relationship between temperature and sales. This strategy keeps the dataset's integrity intact and guarantees accurate results for our analysis.

7.1.5.2 Removing Special Characters

In the dataset preparation phase, particular attention was given to cleaning the temperature column to ensure accurate and consistent values for subsequent analysis. During the data cleaning process, it was observed that the temperature column contained ASCII characters, including the Fahrenheit symbol (°F), which caused issues in data manipulation and analysis. To address this, the ASCII characters were systematically removed from the temperature column using appropriate techniques. Specifically, the 'str. replace ()' function in Python was applied, replacing the Fahrenheit symbol with an empty string. This ensured the removal of the unwanted ASCII characters, allowing for seamless data processing and analysis. The changes were validated, and the temperature column was subsequently reviewed to confirm the successful elimination of the ASCII characters. These steps were documented to maintain transparency and facilitate reproducibility in the dataset preparation process

7.1.5.3 Replacing Values

We used the existence of public holidays as an indicator for events in our calendar dataset. The days that coincided to public holidays were given a value of 1, treating them as event days. On the other hand, we assigned a value of 0 to all other days, signifying the absence of an event. We can now account for the significance of public holidays in our study thanks to this change. We can explore the effects of these special days on a variety of areas by depicting events in this way, including sales trends, customer behavior, and any other pertinent factors. This strategy makes our analysis more thorough and guarantees that the impact of public holidays is properly considered.

7.1.5.4 Adding Column

We grouped the dataset into 28 sub conditions based on temperature ranges to analyses the link between temperature and wind speed. We may evaluate the combined effects of temperature and wind speed on many aspects of interest because each sub condition reflects a certain temperature range. This classification gives us a finer grasp of the interactions between various temperature conditions and wind speed in our analysis. Considering these 28 sub conditions allows us to more thoroughly examine temperature-wind speed dynamics, spot trends, and possibly get insights that might not be apparent when considering temperature and wind speed as continuous variables.

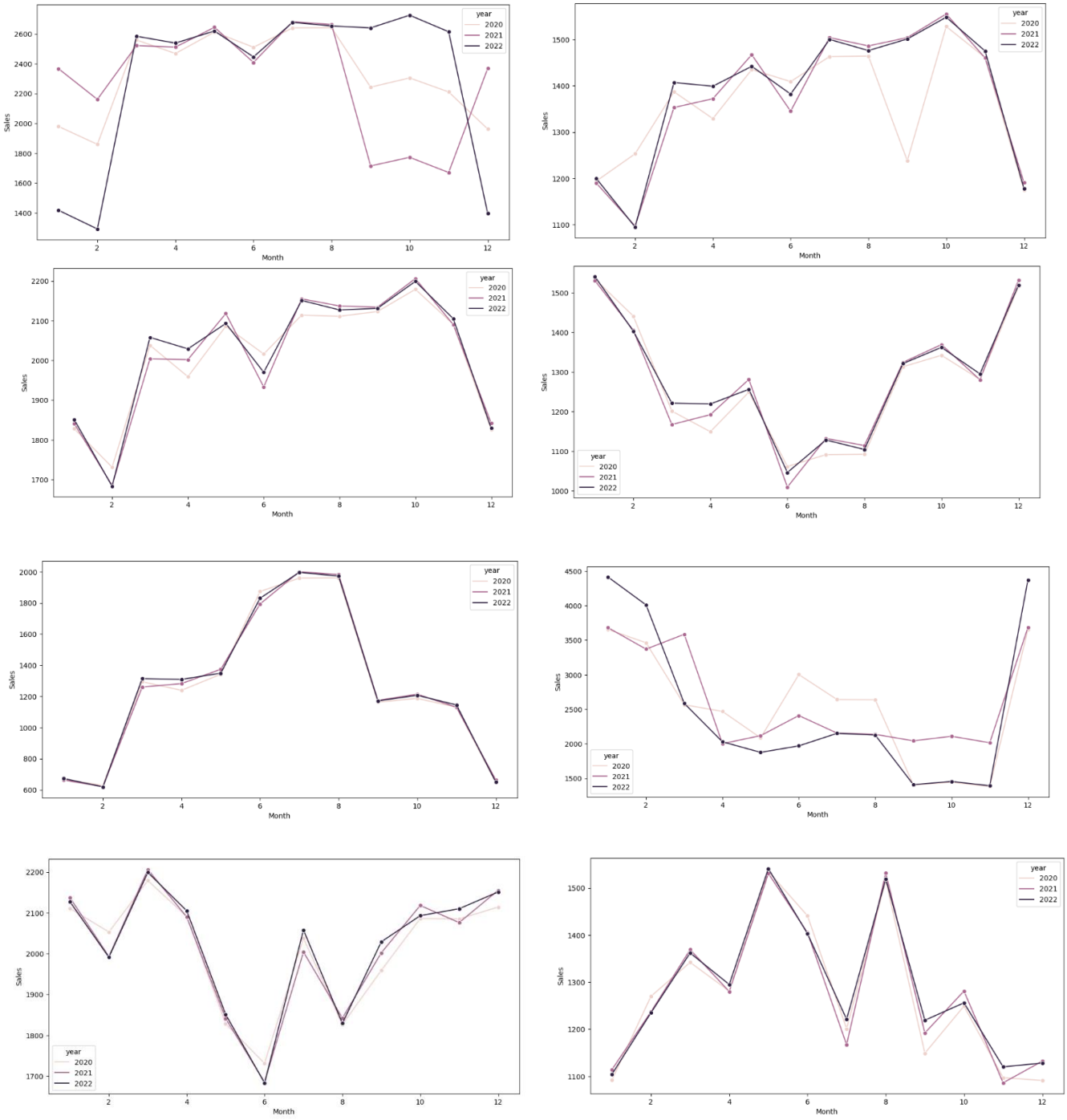
We converted the temperature and wind speed conditions, which were previously represented as strings, into numerical values to aid in analysis. In order to do this, we converted the 28 sub conditions into numbers between 1 and 28. Since numerical values support mathematical operations and comparisons, this translation enables quantitative analysis and modelling.

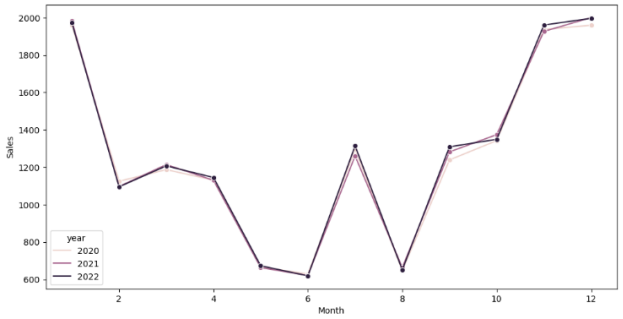
7.1.6 Graphs

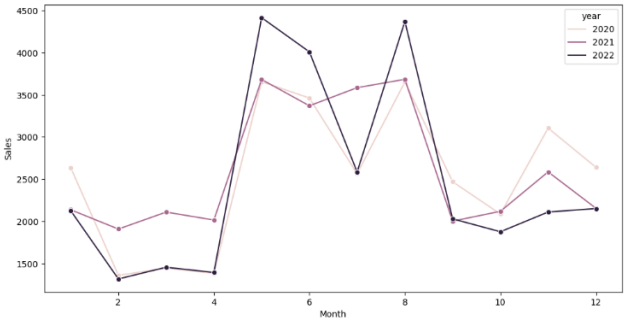
Here are some graphs that are performed for EDA process

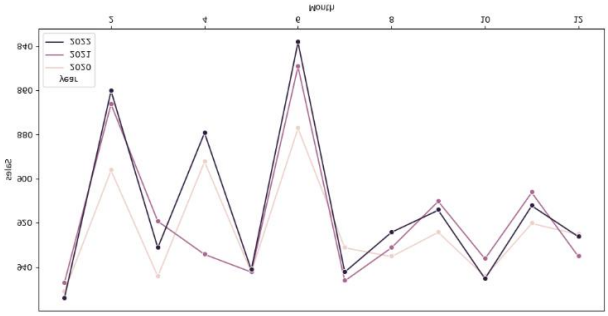
7.1.6.1 Line Graph

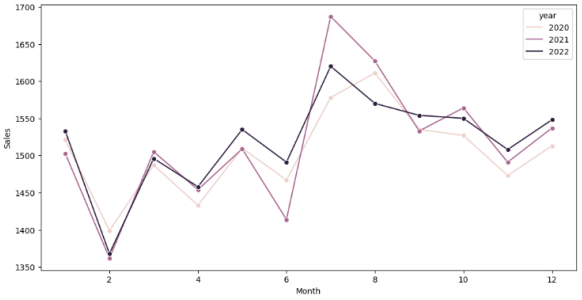
This line graph is showing the all sales of complete dataset as each medicine separately

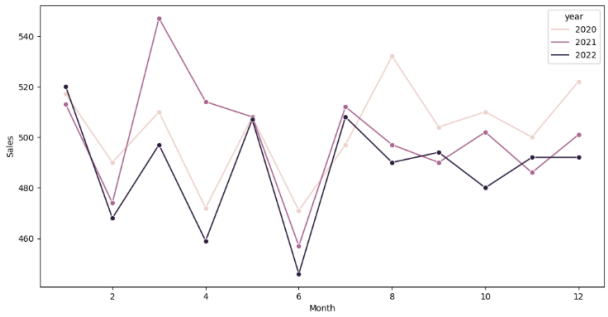


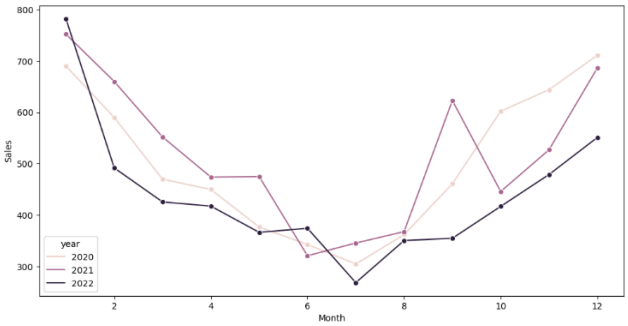

Sales
Month
year
2020
2021
2022


Sales
Month
year
2020
2021
2022




Sales
Month
year
2020
2021
2022


Sales
Month
year
2020
2021
2022


Sales
Month
year
2020
2021
2022

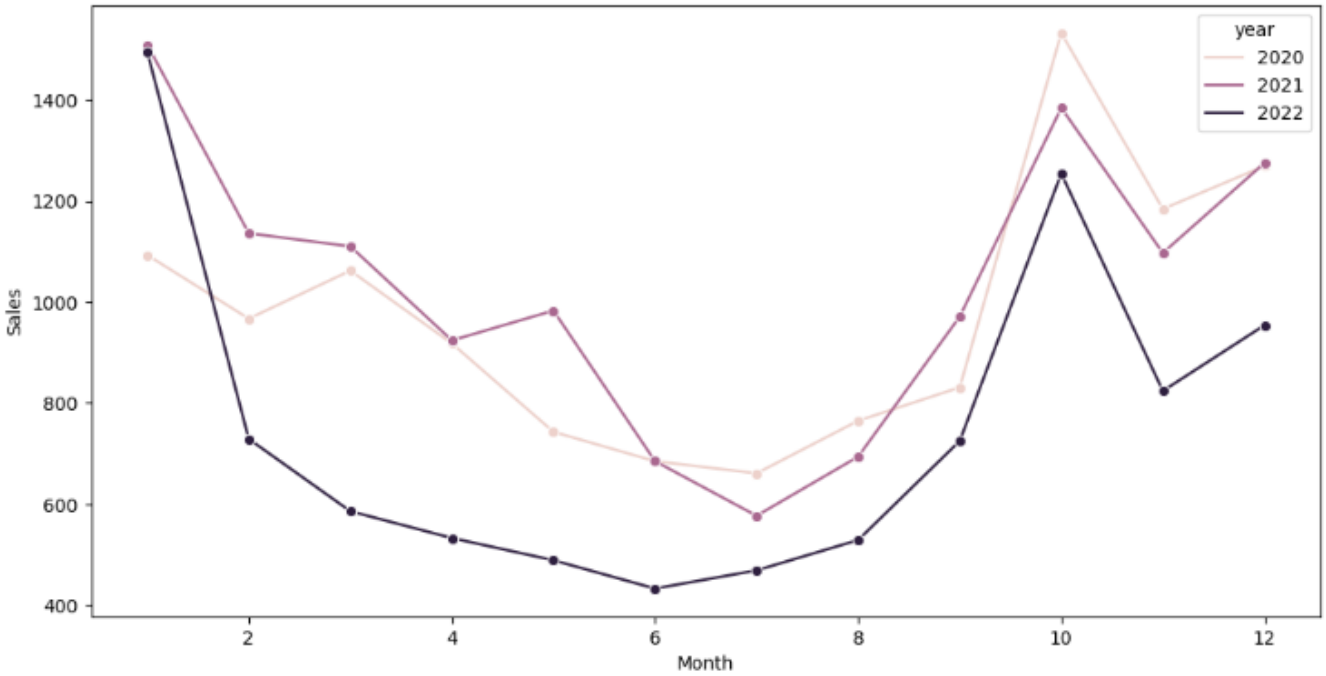


Figure 7. 1 Line Graphs of all products respectively

7.1.6.2 Pie Chart

This pie chart is showing the difference between sales of different medicines of the dataset.

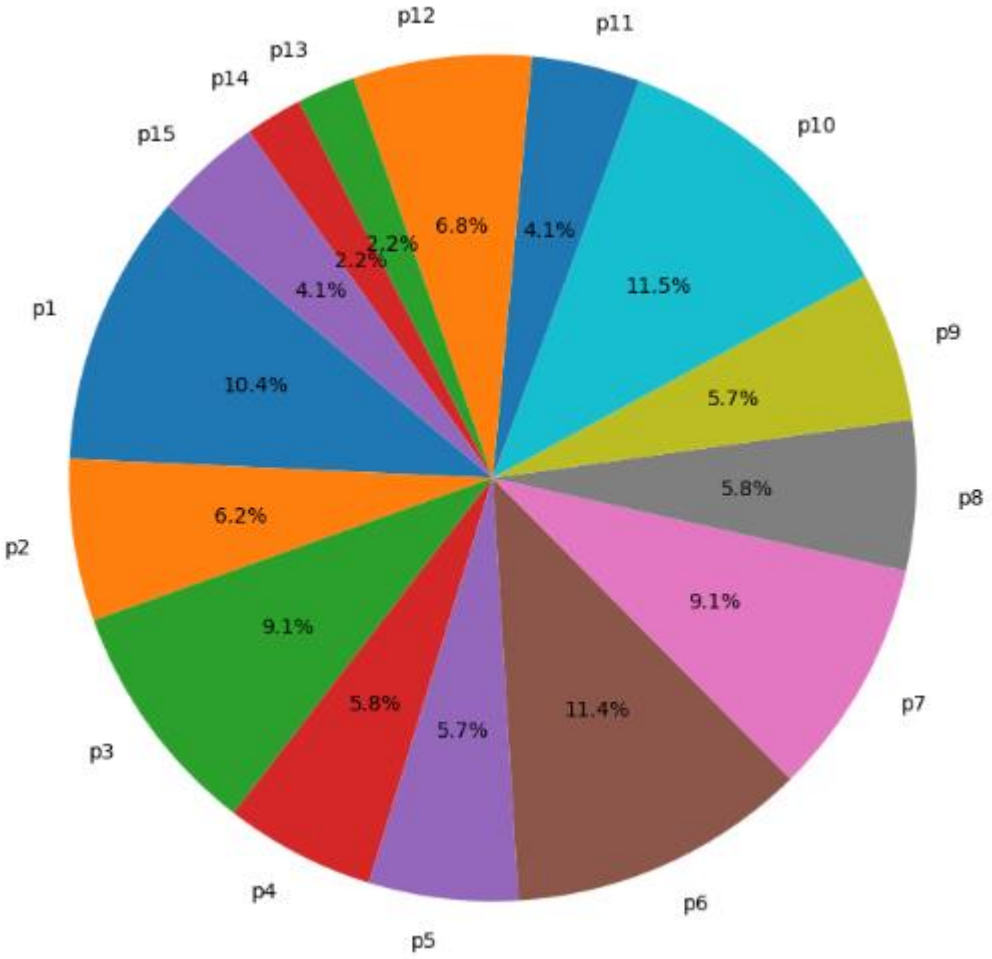


Figure 7. 2 all products sales differences

7.1.6.3 Co-Relations

Correlation is a statistical measure that describes the relationship between two variables. It indicates how closely the values of one variable are associated with the values of another variable. Correlation is used to determine the strength and direction of the linear relationship between two variables.

Positive correlation (r > 0): As one variable increases, the other variable tends to increase as well. A value of +1 indicates a perfect positive correlation.

Negative correlation (r < 0): As one variable increases, the other variable tends to decrease. A value of -1 indicates a perfect negative correlation.

No correlation (r ≈ 0): There is no discernible linear relationship between the variables. However, it's important to note that there could still be other types of relationships (e.g., nonlinear relationships) that are not captured by the correlation coefficient.

Following are correlation of R06 with all its dependent variables/columns. Some of them are

negative some are positive co-relations.

Some correlations are below to describe the nature of independent variables.

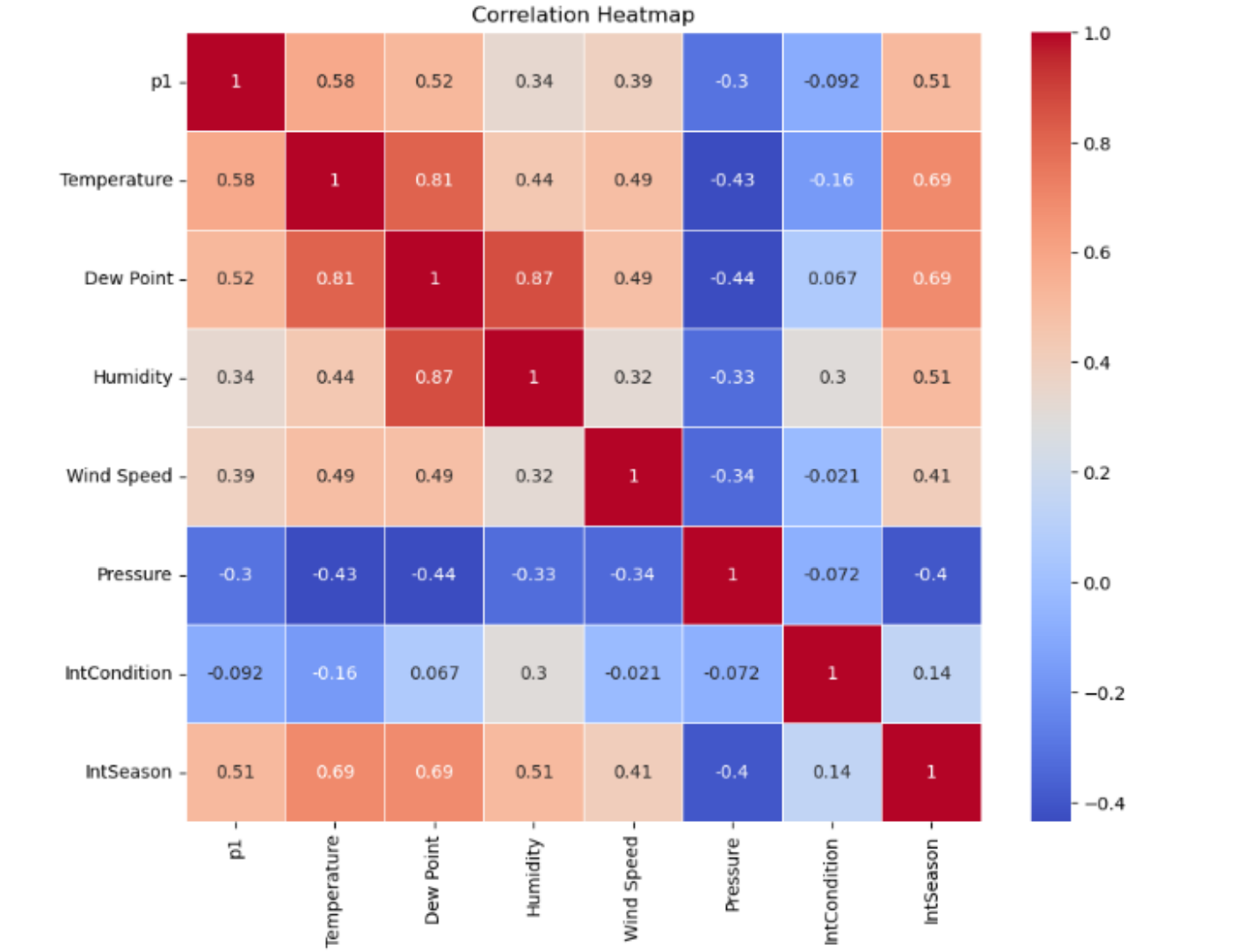


Figure 7. 3 Relation of p1 with all its independent variables

This figure is describing that best relation of target variable(p1) is with temprature, dewpoint and season of the years that means that these independent variabes are directly effecting the results of the model. Relations of these variables are below seperatly.

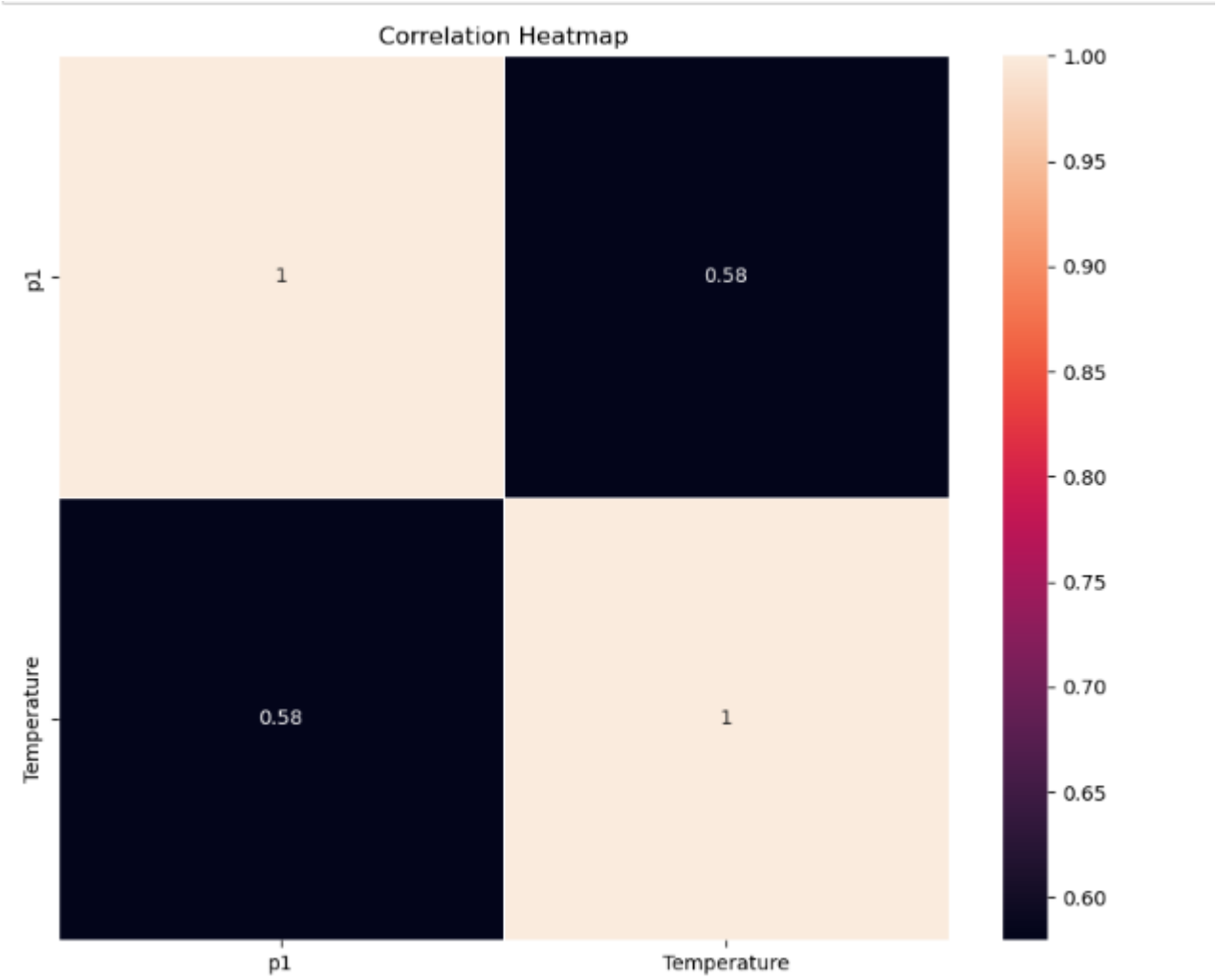


Figure 7.3. 1 relation between p1 and temperature

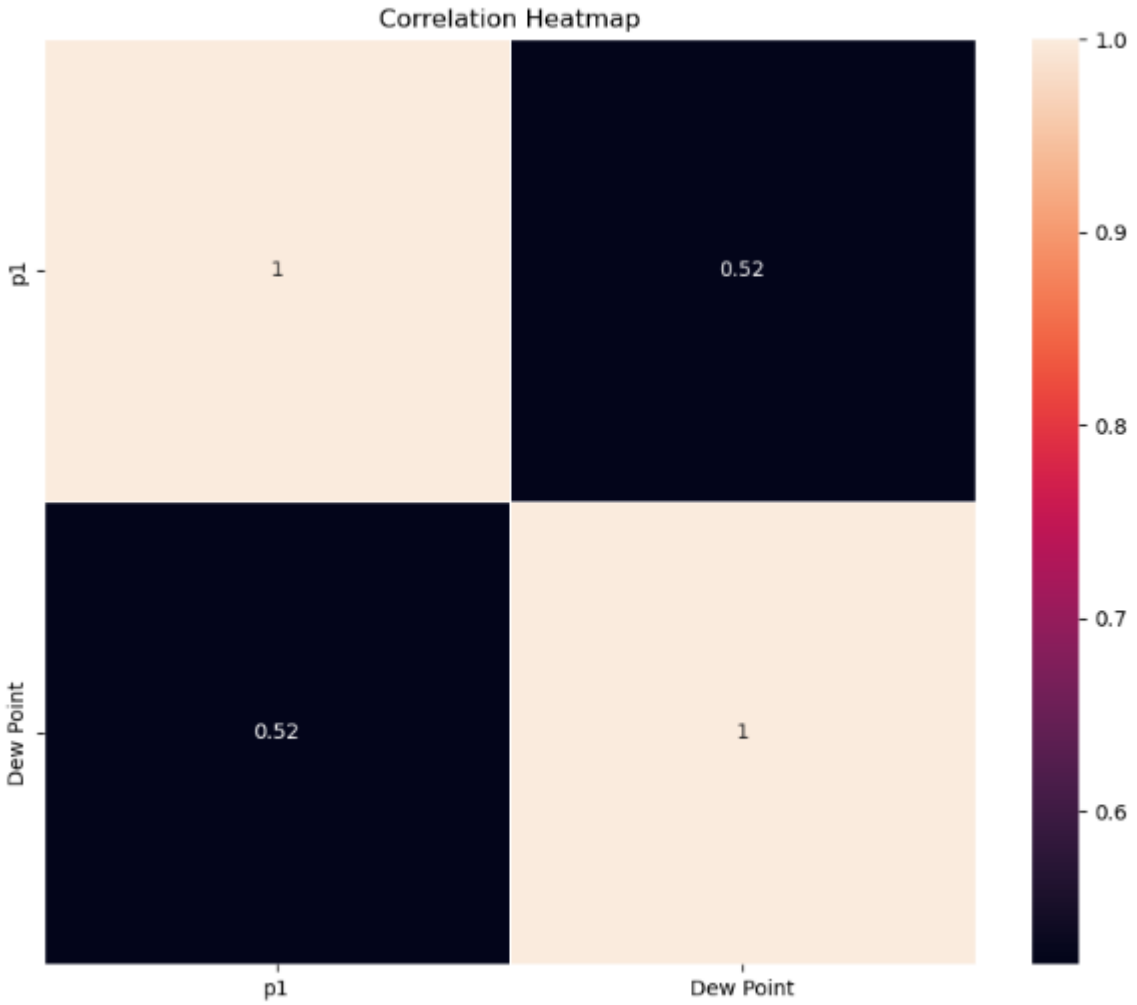


Figure 7.3. 2 relation between p1 and dew point

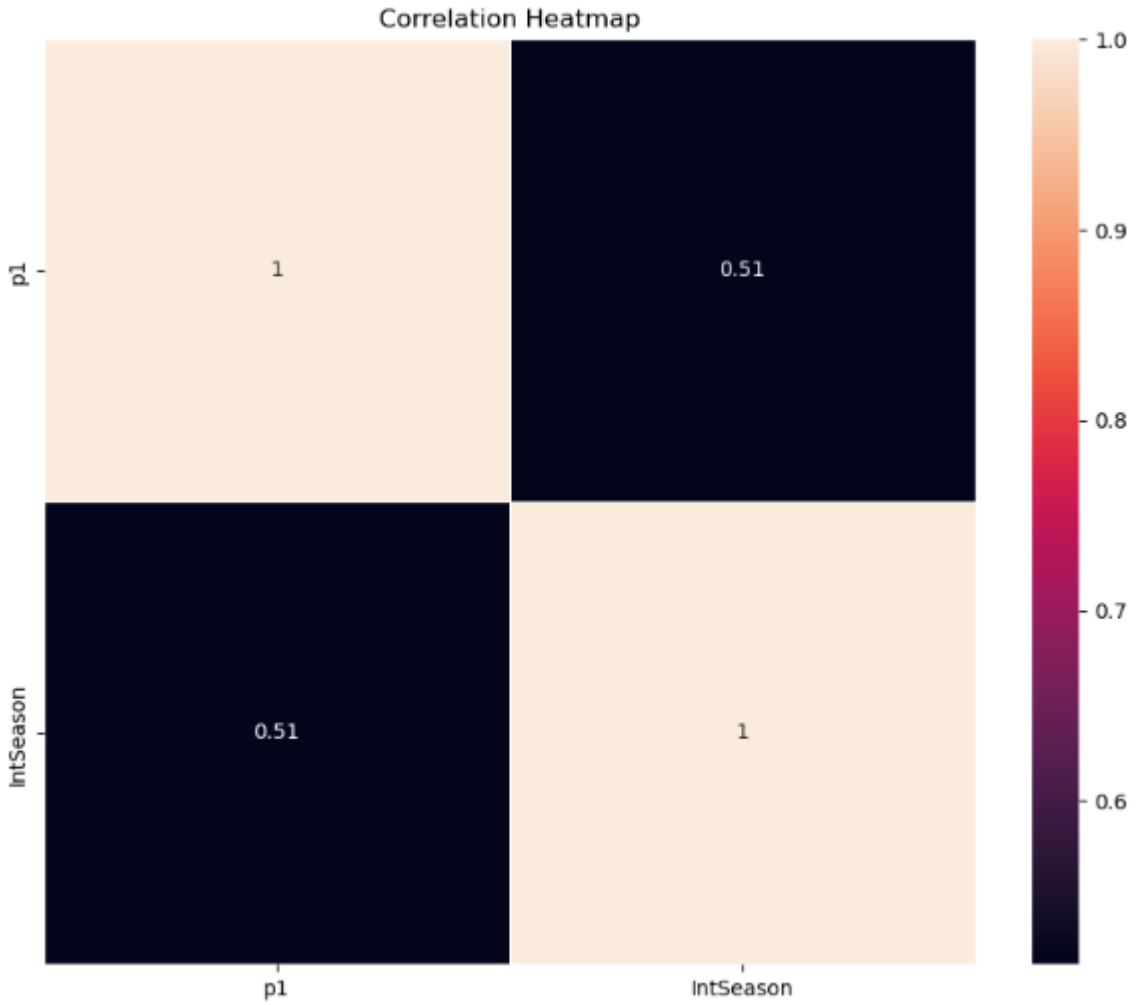


Figure 7.3. 3 relation between season and p1

7.1.7 Training and Testing

7.1.7.1 Data Split

To facilitate accurate model development and evaluation, the dataset was divided into training and testing sets using a 70:30 ratio. The training set, comprising 70% of the dataset, was utilized for model training, parameter optimization, and capturing underlying data patterns. This larger subset provided sufficient data for building robust predictive models. The remaining 30% of the data constituted the testing set, which was kept separate and untouched throughout the model development phase. This smaller portion of the dataset served as an unbiased evaluation set, allowing for a comprehensive assessment of model performance and its ability to generalize to new, unseen data instances. Employing the 70:30 split ratio aimed to strike an appropriate balance between model training and validation, ensuring the reliability and effectiveness of our predictive models in real-world scenarios

7.1.7.2 Models

"For our predictive modeling tasks, three different algorithms were employed: Linear Regression, Random Forest, and Decision Trees. These algorithms were chosen based on their suitability for the specific objectives of our project.

Linear Regression, a classical statistical modeling technique, was utilized to establish linear relationships between the input features and the target variable. It provided insights into the magnitude and direction of the impact that each feature had on the target variable.

Random Forest is an ensemble machine learning technique that combines multiple decision trees to make more accurate predictions. It's known for its high predictive accuracy and robustness against overfitting. By aggregating the outputs of individual decision trees, Random Forest can handle both classification and regression tasks, making it a popular choice for various applications, including data classification, feature importance analysis, and handling large and complex datasets.

Decision Trees were also used as a standalone algorithm for modeling. Decision Trees are non-parametric models that split the dataset based on various attributes, creating a tree-like structure to make predictions. They are particularly effective in capturing interactions between features and handling both categorical and continuous variables.

By employing these three algorithms, we aimed to leverage the strengths of each approach and compare their performance in our predictive modeling tasks. This allowed us to gain a comprehensive understanding of the data and select the most effective algorithm for accurate predictions in real-world scenarios

7.1.7.3 Models Comparison

R2 ranges from 0 to 1. A value of 0 indicates that the model does not explain any of the variance in the target variable, while a value of 1 indicates that the model explains all of the variance. In practice, R2 values are typically between 0 and 1, but negative values are possible if the model performs worse than a simple average.
Higher R2 values indicate that the model explains a larger proportion of the variance in the target variable. For example, an R2 of 0.8 means that the model explains 80% of the variance. Generally, higher R2 values are preferred as they indicate better predictive power.

Another way to choose best model according to our dataset result is that if the ***difference between RMSE and MAE*** is least it indicates best performance on the dataset

	R2
Linear Regression	0.4584
Decision Tree	0.8488
Random Forest	0.8672

After analyzing above results, we can say that Random Forest is best for our dataset for further predictions.

1. Its R2 is highest among all other model.

7.2 Conclusion

We observed that there are some products which are completely not depending on the temperature and event and after it we have categorized them as products which are not sold due to increase or decrease situation of weather or any event cannot affect their sales.

Some products totally depend on weather, temperature and events that's means their sales completely effect by weather and events occurring.

Some products are also found that partially depend on weather and events.

A API

API For PIE Chart

```
CORS(app)
@app.route('/image')def
serve_image():

    CleanDataset=pd.read_csv('New_Final.csv')
    dailydataset=Dailydata(CleanDataset) columns =
    dailydataset.iloc[:, 0:8]
    sizes = columns.sum(axis=0)
    plt.ion()
    plt.clf()

    plt.figure(figsize=(8, 5))
    sizes.plot.pie()
    buf = io.BytesIO()

    # Save the figure to the BytesIO object
    plt.savefig(buf, format='png')

    # Seek to the beginning of the BytesIO object
    buf.seek(0)

    # Create a Flask Response object
    response = Response(buf.getvalue(), mimetype='image/png')
    plt.ioff()
    # Send the Response object to the client
    return response
```

API FOR DASHBOARD GRAPHS

```
@app.route('/GetGraphs',methods=['GET', 'POST']) def dashboardgraphs():
CleanDataset=pd.read_csv('New_Final.csv') if request.method == 'POST':
content = request.json
startDate=content["startDate"]
endDate=content["endDate"] types=content["type"]
#3 attributes startdate enddate datetype if(types=='Weekly'):
dailydataset=Dailydata(CleanDataset) elif(types=='Monthly'):
dailydataset=Monthlydata(CleanDataset) else:
dailydataset=Yearlydata(CleanDataset) plotgraph(dailydataset,startDate,endDate) with
open("plot.png", "rb") as img_file:
my_img = base64.b64encode(img_file.read()) base64_string = my_img .decode('utf-8')
else:
dailydataset=Dailydata(CleanDataset) plotgraph(dailydataset,'2017-06-01','2017-07-13') with
open("plot.png", "rb") as img_file:
my_img = base64.b64encode(img_file.read()) base64_string = my_img .decode('utf-8')
return jsonify(base64_string)
```

CURRENT PREDICTIONS

https://api.openweathermap.org/data/2.5/weather?lat=24.9&lon=67.0&appid=677094fd637b92220b1819ef46fb5185&units=imperial

Refrences

- https://www.w3schools.com/python/python_ml_linear_regression.asp
- https://www.geeksforgeeks.org/linear-regression-python-implementation/
- https://www.sciencedirect.com/science/article/pii/S0920410521008998
- https://www.researchgate.net/publication/346821916_A_Study_on_Gradient_Boosting_Algorithms_for_Development_of_AI_Monitoring_and_Prediction_Systems
- https://www.kaggle.com/datasets/milanzdravkovic/pharma-sales-data
- https://www.wunderground.com/history/daily/pk/karachi/OPKC/date
- https://www.researchgate.net/publication/346821916_A_Study_on_Gradient_Boosting_Algorithms_for_Development_of_AI_Monitoring_and_Prediction_Systems
- https://repository.upenn.edu/cgi/viewcontent.cgi?article=1186&context=marketing_papers
- https://www.sciencegate.app/keyword/408087
- https://www.timeanddate.com/holidays/pakistan/2013?hol=1