SUPERMARKET MANAGEMENT SYSTEM

MINI PROJECT REPORT

Submitted by

DINESH S	**231801034**
IRAIYANBU ST	**231801061**
HARISH TUTU YT	**231801050**

In partial fulfillment for the award of the degree of

BACHELOR OF

ENGINEERING IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS) THANDALAM

CHENNAI-602105

2024 - 2025

BONAFIDE CERTIFICATE

Certified that this project report "**SUPERMARKET MANAGEMENT SYSTEM**" is the bonafide work of **"DINESH S (231801034), HARISH TUTU YT (231801050), IRAIYANBU ST(231801061)"**

who carried out the project work under my supervision.

Submitted for the Practical Examination held on _____

SIGNATURE

Dr. GNANASEKAR J M
Head of the Department, Artificial intelligence and data Science,Rajalakshmi EngineeringCollege (Autonomous),Chennai-602105

SIGNATURE

Mr Thiyagarajan
Associate Professor, Artificial Intelligence and DataScience, Rajalakshmi Engineering College,(Autonomous), Thandalam, Chennai-602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

ABSTRACT

The Supermarket Management System aims to streamline supermarket operations by automating inventory, sales, customer data, and supplier management.
Key features include product management, sales processing, customer profiles, supplier tracking, and reporting.
It uses a relational database for structured data storage, ensuring integrity and minimizing redundancy.
SQL queries facilitate efficient data retrieval and manipulation.
By implementing this system, supermarkets can enhance efficiency, reduce operational costs, and improve customer service through quick transaction processing.
Overall, this project highlights the importance of effective database management in optimizing supermarket operations and fostering growth.

1. INTRODUCTION:

A Supermarket Management System (SMS) streamlines key supermarket operations like inventory management, sales tracking, customer relationship management, and financial reporting. It enhances efficiency, improves customer satisfaction, and boosts profitability by providing real-time data and automation for better decision-making and smoother daily operations.

Objectives of the Supermarket Management System

1. **Inventory Management :** The system allows for real-time tracking of stock levels, automatic reordering of products, and efficient management of perishable goods to reduce waste.

2. **Sales Tracking** : By automating the sales process, the SMS provides accurate sales data, helping managers identify trends, forecast demand, and make informed pricing decisions.

3. **Customer Relationship Management** : The system can store customer data, preferences, and purchase history, enabling personalized marketing strategies and improved customer service.

4. **Financial Reporting** : Automated financial reporting features help in monitoring sales performance, managing expenses, and generating insights for strategic decision-making.

5. **Employee Management** : The system can also assist in managing employee schedules, performance tracking, and payroll processing, contributing to overall operational efficiency.

MODULES:

A supermarket management system (SMS) requires a robust database to efficiently store and manage various types of data. Here are some essential modules that you can consider incorporating into your SMS using a database management system:

Core Modules:

- Product Management
- Customer Management
- Sales and Billing
- Employee Management

Additional Modules:

- Reporting and Analytics
- Supplier Management
- Promotions and Discounts
- Integrations

Database Considerations:

- Choose a suitable RDBMS (MySQL, PostgreSQL, SQL Server)
- Normalize data for integrity
- Implement robust security measures
- Design for scalability

2. SURVEY OF TECHNOLOGIES:

2.1 Software Description :

This project utilizes a combination of software tools to create a comprehensive and efficient supermarket management system:

- **Database Management System (DBMS):** MySQL is chosen as the DBMS for its reliability, performance, and widespread use in various applications.
- **Integrated Development Environment (IDE):** PyCharm is selected as the IDE for its Python-specific features, code completion, debugging tools, and seamless integration with MySQL.
- **Web Framework:** Flask is employed as a lightweight and flexible web framework to build the user interface and handle HTTP requests.

2.2 Languages :

- **SQL:** Structured Query Language is used to interact with the MySQL database, defining data structures, performing queries, and managing data integrity.
- **Python**: A versatile programming language is used to develop the backend logic, implement business rules, and interact with the MySQL database through the SQLAlchemy ORM.

2.2.1 MONGO DB :

- Flexible Schema: MongoDB's schema-less design allows easy adaptation to changing data needs, such as adding new product attributes or customer details.
- High Scalability: Handles large volumes of transactions and inventory data efficiently using horizontal scaling with sharding.
- Real-Time Insights: Offers real-time analytics and reporting, enabling quick access to sales trends, stock levels, and customer behavior.
- Embedded Data: Speeds up queries by embedding related data, like sale items within a sales document, reducing the need for joins.

2.2.2 Python :

- Develop the backend logic: Implementing business rules, calculations, and data processing tasks.
- Interact with the MySQL database: Using the SQLAlchemy Object-Relational Mapper (ORM) to map Python objects to database tables, simplifying data access and manipulation.
- Create the user interface: Building the web interface using Flask, allowing users to interact with the system and perform various tasks.
- Integrate with other systems: Connecting the supermarket management system with external systems like accounting software or inventory management tools.

3.REQUIREMENTS AND ANALYSIS :

3.1 Requirement Specification

Functional Requirements :

- **Product Management:**
  - Add, edit, and delete products
  - Track inventory levels and reorder points
  - Manage product categories and suppliers
  - Handle product expiration dates
- **Customer Management:**
  - Create, update, and view customer profiles
  - Track customer purchase history and loyalty points
  - Manage customer addresses and contact information
- **Sales and Billing:**
  - Process sales transactions (cash, card, digital wallets)
  - Generate invoices and receipts
  - Handle discounts, promotions, and coupons
- **Employee Management:**
  - Manage employee information (roles, salaries, contact details)
  - Schedule shifts and track time and attendance
  - Assign access controls and permissions

- **Reporting and Analytics:**
    - Generate sales reports (daily, weekly, monthly)
    - Analyze inventory levels and trends
    - Track customer behavior and loyalty
    - Evaluate employee performance

Non-Functional Requirements:

- **Performance:** The system should handle a large number of transactions efficiently.
- **Scalability:** The system should be able to accommodate future growth and increased data volumes.
- **Security:** Sensitive customer and financial data should be protected.
- **Usability:** The user interface should be intuitive and easy to navigate.
- **Reliability:** The system should be reliable and have minimal downtime.

3.2 Hardware and Software Requirements :

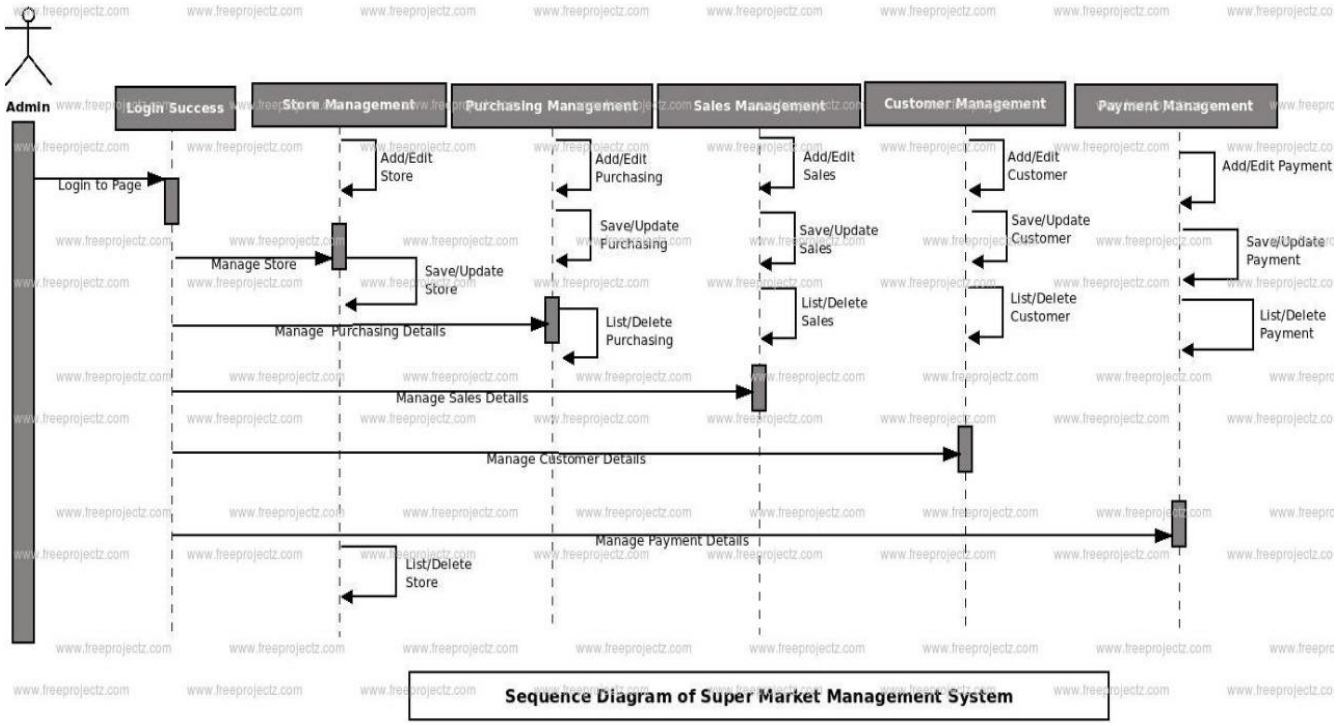
Hardware :

- **Server:** A powerful server with sufficient CPU, RAM, and storage to handle the database and application workload.
- **Network:** A reliable network connection to allow access to the system from different locations.
- **POS Terminals:** Point-of-sale terminals for processing sales transactions.

Software :

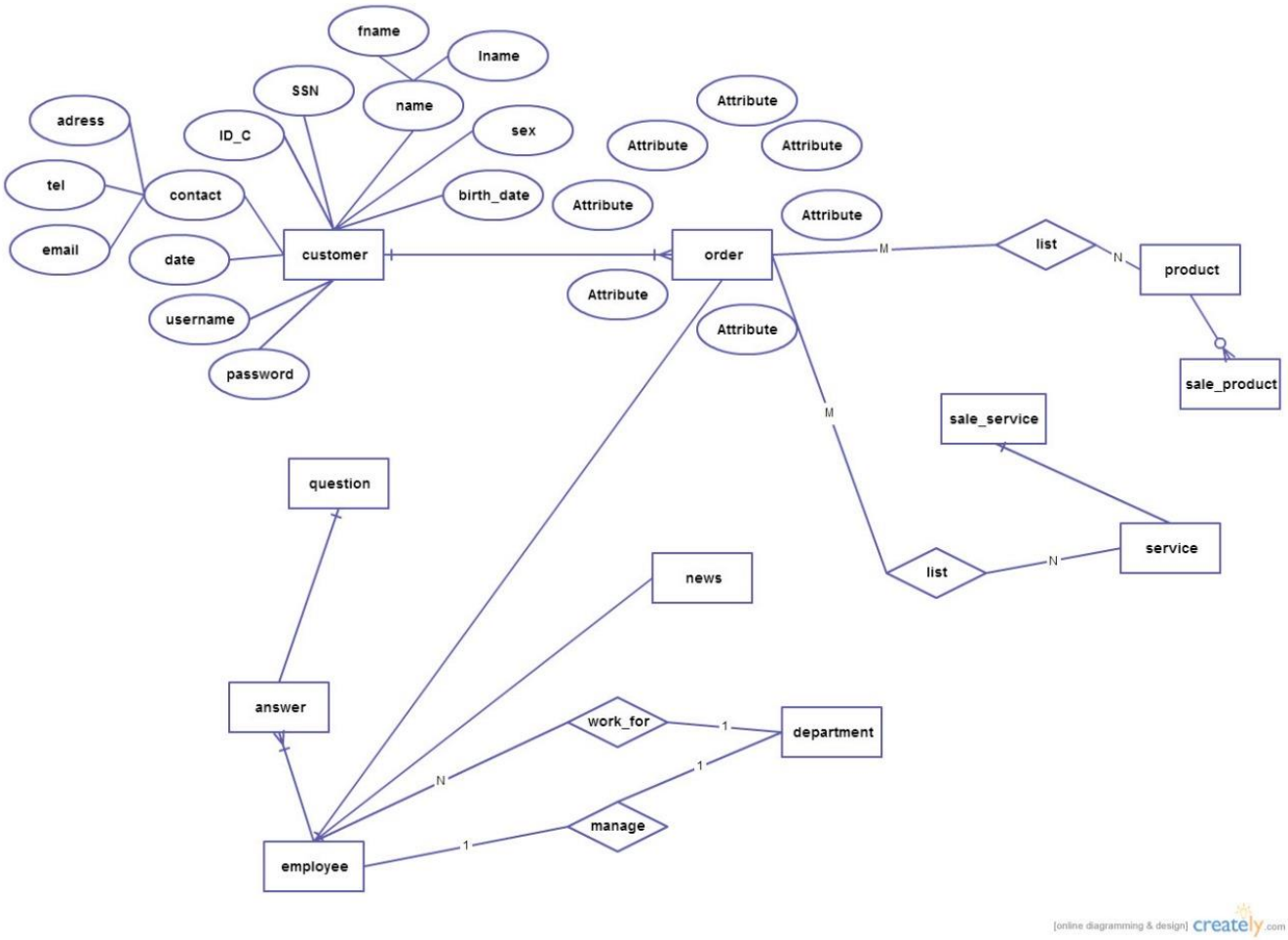
- **Database Management System (DBMS):** MySQL, PostgreSQL, or SQL Server.
- **Integrated Development Environment (IDE):** PyCharm or Visual Studio Code.
- **Web Framework:** Flask or Django.
- **Operating System:** Linux or Windows.
- **Web Server:** Apache or Nginx.

3.3 ARCHITECTURE DIAGRAM :



Sequence Diagram of Super Market Management System

3.4 ER DIAGRAM :



4. PROGRAM CODE

```
# ----------------SUPERMARKET MANAGEMENT SYSTEM-------


items = []

while True:

    display = input('Press enter to continue.')

    print('-----------------Welcome to the supermarket-----------------')

    print('1. View items\n2. Add items for sale\n3. Purchase items\n4. Search items \n5. Edit items\n6. Exit')

    choice = input('Enter the number of your choice : ')


    if choice == '1':

        print('-----------------View Items-----------------')

        print('The number of items in the inventory are : ', len(items))

        while len(items) != 0:

            print('Here are all the items available in the supermarket.')

            for item in items:

                for key, value in item.items():

                    print(key, ':', value)
```

```
            break
    elif choice == '2':


        print('------------------Add items-----------------')
        print('To add an item fill in the form')


        # while True:
        #     try:
        #     number_items = int(input('Enter the number of items you want to add in the inventory : '))


        #         break
        #     except ValueError:
        #         print('Number of items should only be in digits')
        # for num in range(number_items):


        item = {}
        item['name'] = input('Item name : ')
        while True:
            try:
                item['quantity'] = int(input('Item quantity : '))
                break
```

```
            except ValueError:


                print('Quantity should only be in digits')
        while True:

            try:
                item['price'] = int(input('Price $ : '))
                break


            except ValueError:
                print('Price should only be in digits')
        print('Item has been successfully added.')
        items.append(item)


    elif choice == '3':
        print('------------------purchase items------------------')
        print(items)
        purchase_item = input('which item do you want to purchase? Enter name : ')
        for item in items:
```

```
        if purchase_item.lower() == item['name'].lower():
            if item['quantity'] != 0:
                print('Pay ', item['price'], 'at checkout counter.')
                item['quantity'] -= 1
            else:
                print('item out of stock.')



    elif choice == '4':
        print('------------------search items------------------')
        find_item = input('Enter the item\'s name to search in inventory
: ')
        for item in items:


            if item['name'].lower() == find_item.lower():
                print('The item named ' + find_item + ' is displayed below
with its details')
                print(item)
            else:
                print('item not found.')
```

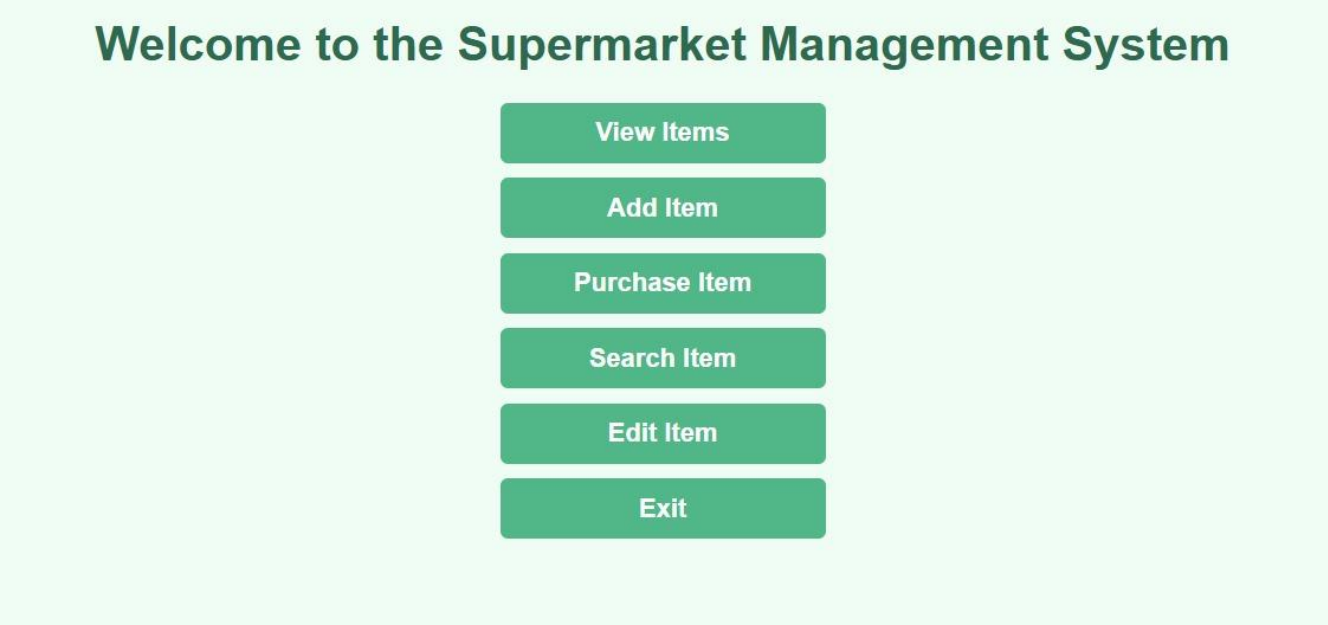
```
    elif choice == '5':
        print('------------------edit items-----------------')
        item_name = input('Enter the name of the item that you want to
edit : ')
        for item in items:


            if item_name.lower() == item['name'].lower():
                print('Here are the current details of ' + item_name)
                print(item)
                item['name'] = input('Item name : ')


                while True:
                    try:


                        item['quantity'] = int(input('Item quantity : '))
                        break
                    except ValueError:
                        print('Quantity should only be in digits')
                while True:
                    try:
```

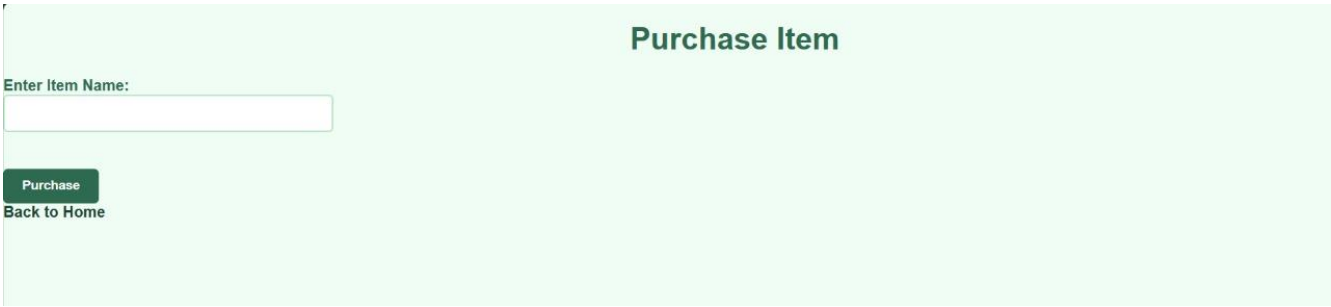
```
                item['price'] = int(input('Price $ : '))
                break
            except ValueError:
                print('Price should only be in digits')
        print('Item has been successfully updated.')
        print(item)
    else:
        print('Item not found')

elif choice == '6':
    print('------------------exited-----------------')
    break

else:
    print('You entered an invalid option')
```

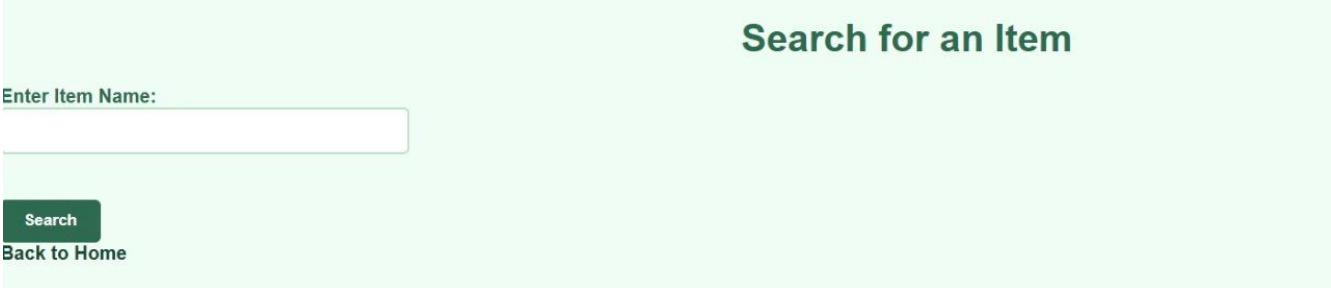
DATA SETS :

	ProductNa	Brand	Price	DiscountPr	Image_Url	Quantity	Category	SubCatego	Absolute_Url
2	Onion (Loc	Fresho	69.75	52	https://wv	2 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/40075537/fresho-onion-2-kg/
3	Onion	Fresho	174.35	130	https://wv	5 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/1201414/fresho-onion-5-kg/
4	Onion (Loc	Fresho	34.87	26	https://wv	1 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/10000148/fresho-onion-1-kg/
5	Onion	Fresho	69.74	52	https://wv	2 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/1201413/fresho-onion-2-kg/
6	Onion (Loc	Fresho	174.37	129	https://wv	5 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/10000150/fresho-onion-5-kg/
7	Farm Eggs	Fresho	110	78	https://wv	12 pcs	Eggs, Meat	Farm Eggs	https://www.bigbasket.com/pd/40072320/fresho-farm-eggs-table-tray-medium-antibiotic-residue-free-12-pcs/
8	Farm Eggs	Fresho	210	181	https://wv	30 pcs	Eggs, Meat	Farm Eggs	https://www.bigbasket.com/pd/150502/fresho-farm-eggs-table-tray-medium-antibiotic-residue-free-30-pcs/
9	Potato (Lo	Fresho	50.32	38.75	https://wv	1 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/40048457/fresho-potato-new-crop-1-kg/
10	Potato	Fresho	100.65	77.5	https://wv	2 kg	Fruits & Ve	Potato, Or	https://www.bigbasket.com/pd/40162469/fresho-potato-new-crop-2-kg/
11	Eggs - Reg	Fresho	250	250	https://wv	5x6 pcs	Eggs, Meat	Farm Eggs	https://www.bigbasket.com/pd/1205938/fresho-eggs-regular-5x6-pcs/
12	Farm Eggs	Fresho	50	38	https://wv	6 pcs	Eggs, Meat	Farm Eggs	https://www.bigbasket.com/pd/281204/fresho-farm-eggs-regular-medium-antibiotic-residue-free-6-pcs/
13	Eggs - Reg	Fresho	100	76	https://wv	2x6 pcs	Eggs, Meat	Farm Eggs	https://www.bigbasket.com/pd/1203477/fresho-eggs-regular-2x6-pcs/
14	Eggs - Reg	Fresho	200	200	https://wv	4x6 pcs	Eggs, Meat	Farm Eggs	https://www.bigbasket.com/pd/1205933/fresho-eggs-regular-4x6-pcs/
15	Atta/Godil	Aashirvaad	119	118	https://wv	2 kg	Foodgrains	Atta Whol	https://www.bigbasket.com/pd/161826/aashirvaad-atta-whole-wheat-2-kg-pouch/

5.RESULTS AND DISCUSSION :







Items in Inventory

tomato - Quantity: 1 - Price: $0.01

potato - Quantity: 3 - Price: $20.0

Back to Home

Add a New Item

Name:

Quantity:

Price:

Add Item

Back to Home

Edit an Item

Current Name:

New Name:

New Quantity:

New Price:

Update Item

Back to Home

6. CONCLUSION:

Conclusion for Supermarket Management System in DBMS

- In conclusion, the Supermarket Management System (SMS) leverages a Database Management System (DBMS) to streamline various supermarket operations, including inventory management, sales tracking, and customer relation.

- This system enhances efficiency by providing real-time data access, enabling informed decision-making and responsiveness to customer needs. Its robust security measures protect sensitive information, ensuring data integrity and compliance with regulations.

- Additionally, the system's scalability allows it to adapt to changing business requirements, making it a valuable tool for improving customer satisfaction and driving profitability. Overall, the SMS represents a significant advancement in modern supermarket management, optimizing operations and resource allocation effectively.

7.REFERENCES :

- **Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems*. 7th Edition. Pearson.**

- **Gonzalez, J., & Palacios, J. (2018). "Analyzing Customer Preferences in Supermarkets Using Data Mining Techniques." *Journal of Retailing and Consumer Services*, 45, 345-354.**

- **Khan, M. A., & Qureshi, M. A. (2013). "Retail Management Information System: An Effective Tool for Supermarket Management." *International Journal of Scientific & Engineering Research*, 4(5), 1-5.**

- **Akhter, A., & Dey, S. (2019). "Development of a Smart Supermarket Management System." *International Journal of Computer Applications*, 975, 8887.**