A Mini Project Report on

"RESTAURANT MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Engineering
In
Computer Engineering
By

MADHUR VINOD SHINDE (97)
HARSH JAYENDRA SAKPAL (85)
AMAN SURESH SINGH (101)
SUJAL MANOJ SINGH (103)

Under the Guidance of

Prof. Dhananjay Raut

Department of Computer Engineering
Watumull Institute of Electronics Engineering and Computer Technology
Ulhasnagar - 421003
UNIVERSITY OF MUMBAI
Academic Year 2023-2024

Approval Sheet

This Mini **Project Report** entitled ***"RESTAURANT MANAGEMENT SYSTEM" S***ubmitted **by** ***"MADHUR VINOD SHINDE" (97), "HARSH JAYENDRA SAKPAL" (85) , "AMAN SURESH SINGH" (101) & "SUJAL MANOJ SINGH" (103) i***s **approved for** the partial fulfilment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Computer Engineering*** from ***University of Mumbai***.

Prof. Dhananjay Raut
(Guide)

Prof. Nilesh Mehta
(H.O.D)

Place: Ulhasnagar

Date:

CERTIFICATE

This is to certify that the mini project entitled ***"RESTAURANT MANAGEMENT SYSTEM"*** submitted by **"MADHUR VINOD SHINDE" (97), "HARSH JAYENDRA SAKPAL" (85) , "AMAN SURESH SINGH" (101) & "SUJAL MANOJ SINGH" (103)** for the partial fulfilment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Engineering,*** to the University of Mumbai, is a bonafidework carried out during academic year 2023-2024.

Prof. Dhananjay Raut
(Guide)

Examiners:

1) _____

2) _____

Prof.Nilesh Mehta
(H.O.D)

Prof.Avinash Gondal
(Principal)

Place: Ulhasnagar

Date:

Declaration

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources.We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

MADHUR VINOD SHINDE(97)_____

HARSH JAYENDRA SAKPAL(85)_____

AMAN SURESH SINGH(101)_____

SUJAL MANOJ SINGH (103)_____

Place: Ulhasnagar

Date:

Abstract

The restaurant management system is a comprehensive gui-based application developed using java netbeans, consisting of a login page, sign-up page, and a forgot password page. After login, the main page offers six functions: add new product, update/delete product, previous days records, monthly analysis, weekly average with any range of dates, and billing page.

The system allows users to:

1. **ADD NEW PRODUCT**: Add new products to the database, facilitating seamless integration of new items into the restaurant's inventory.
2. **UPDATE/DELETE PRODUCT**: Modify product information or remove products from the database, ensuring that the inventory remains accurate and up-to-date.
3. **PREVIOUS DAYS RECORDS**: View records of a specific day selected from a calendar, providing insights into daily operations and sales.
4. **MONTHLY ANALYSIS**: Analyze product sales, showing the most sold products and their quantities over a month, facilitating strategic decision-making and inventory management.
5. **WEEKLY AVERAGE WITH ANY RANGE OF DATES**: Calculate the average quantity of products sold within a selected date range, enabling a detailed understanding of sales trends.
6. **BILLING PAGE**: Generate bills and export them to PDF, ensuring a smooth and professional billing process.

The system operates with two databases:

- **User database**: Stores ID, password, name, phone number, email, security question, and answer, ensuring secure login and data management.
- **Bill database**: Stores ID as consumer serial number, product name, quantity, and price, ensuring accurate and efficient billing processes.
- **Product database**: Stores ID of product and it’s name and price.

Content

1 .INTRODUCTION

1.1 Overview:

The Restaurant Management System is a Java NetBeans application designed to streamline restaurant management tasks. It includes a user-friendly interface with various features such as adding new products, updating or deleting existing ones, and generating bills. This system aims to provide a comprehensive solution for efficient restaurant management.

1.2 Introduction:

The Restaurant Management System simplifies restaurant operations, offering an intuitive GUI for efficient management. It covers functions from product management to billing, enhancing overall productivity and customer service. The system is designed to cater to the needs of small to medium-sized restaurants, providing them with an effective tool for managing their day-to-day operations.

1.3 Objective :

The objective is to develop a user-friendly, efficient Restaurant Management System that enables easy addition, updating, and deletion of products, and provides comprehensive data analysis for informed decision-making. The system aims to streamline restaurant operations, increase efficiency, and improve customer service by automating manual tasks and providing valuable insights into sales data.

1.4 Scope:

The system aims to provide a platform for restaurant staff to manage products effectively, generate bills effortlessly, and analyze sales data for better decision-making. It caters to the needs of small to medium-sized restaurants, offering a scalable solution that can grow with the business. The system is designed to be user-friendly and easy to navigate, making it accessible to restaurant staff with varying levels of technical expertise.

1.5 **Purpose:**

The purpose of the Restaurant Management System is to automate and simplify the day-to-day operations of a restaurant. It ensures smooth product management, accurate billing, and insightful data analysis, ultimately enhancing the restaurant's efficiency and customer satisfaction. By providing a comprehensive solution for restaurant management, the system enables restaurant owners to focus on providing excellent service to their customers.

2. Problem Definitions

2.1 Problem Statement:

The restaurant industry faces challenges in efficiently managing operations such as inventory, billing, and sales analysis. Manual methods are often prone to errors and inefficiencies, leading to decreased productivity and increased costs. There is a need for a comprehensive Restaurant Management System that automates these processes and provides accurate insights to improve decision-making and streamline operations.

Current systems lack integration and scalability, making it difficult for restaurants to adapt to changing business needs and technological advancements. Additionally, there is a lack of user-friendly interfaces and robust security measures, posing risks to data integrity and confidentiality.

Furthermore, traditional methods of data analysis often rely on manual calculations and are time-consuming, limiting the ability of restaurant owners to gain actionable insights from sales data. There is a need for a system that can automate data analysis and provide visualizations to identify trends and opportunities for improvement.

In summary, the restaurant industry requires a modern, integrated, and user-friendly Restaurant Management System that addresses the challenges of manual processes, lack of integration, scalability, data analysis, and security to improve efficiency, productivity, and decision-making.

2.2 Technology:

The Restaurant Management System is developed using Java NetBeans, offering a user-friendly Graphical User Interface (GUI) for intuitive navigation and operation. It operates with two databases: a user database for

storing login information and a bill database for managing product-related transactions.

- **Development Environment**:

1. Java NetBeans IDE: Version 12.0 or higher is recommended, providing a robust and user-friendly Integrated Development Environment (IDE) for Java.
2. Java SE Development Kit (JDK): The latest version should be installed to develop and run Java applications.

- **Database Management:**
  MySQL Database Management System (DBMS): Utilizing MySQL Community Server 8.0 or higher for efficient and secure data storage. It ensures reliability, performance, and ease of use.
  Two Databases:
    1. User Database: Stores login information, ensuring secure access to the system.
    2. Bill Database: Manages product-related transactions, ensuring accurate billing and transaction records.
    3. Product database: Stores ID of product with it name and price.

- **Platform Independence:**
  The use of Java NetBeans ensures that the system is platform-independent and can be easily deployed on various operating systems.
- **Enhanced Efficiency:**

1. The system enhances the efficiency of restaurant management tasks, providing a seamless experience for both the staff and customers.
2. It simplifies complex tasks, such as order management, inventory tracking, and reservation handling, minimizing the learning curve for users.

- **Secure Data Management:**

1. The use of two databases ensures secure data management, providing peace of mind to both restaurant staff and customers.
2. The user database ensures secure access to the system, while the bill database maintains accurate and confidential transaction records.

3. Hardware & Software Specifications

3.1 Hardware Specifications:

The Restaurant Management System requires minimal hardware specifications to operate efficiently. The system is designed to run on standard hardware commonly found in most modern computers.

- **Processor:** A minimum of a dual-core processor is recommended to ensure smooth performance. However, the system is optimized to run on various processor configurations, including Intel Core i3, i5, or i7, and equivalent AMD processors.
- **Memory (RAM):** The system requires a minimum of 4GB of RAM to operate smoothly. However, for optimal performance, 8GB or more of RAM is recommended, especially when handling large datasets for data analysis.
- **Storage:** The system's storage requirements are minimal, with approximately 500MB of available disk space needed for installation. Additional storage space may be required for storing data generated by the system, such as transaction records and reports.
- **Display:** The system is designed to run on standard display resolutions, including 1280x720 (HD) or higher. A monitor with a screen size of 15 inches or larger is recommended for better readability of the graphical user interface.
- **Input Devices:** Standard input devices such as a keyboard and mouse are required for interacting with the system. Additionally, touchscreen monitors may enhance user experience for systems deployed in restaurants with touchscreen capabilities.

3.2 Software Specifications:

The Restaurant Management System is developed using Java NetBeans,

leveraging various software components and libraries to provide its functionality. The system is designed to be platform-independent, allowing it to run on different operating systems commonly used in restaurant environments.

- **Operating System:** The system is compatible with multiple operating systems, including Windows, macOS, and Linux distributions such as Ubuntu and CentOS. It is designed to run seamlessly on both 32-bit and 64-bit architectures, ensuring broad compatibility with different hardware configurations.
- **Java Development Kit (JDK):** The system requires a compatible Java Development Kit (JDK) to compile and run Java code. It is recommended to use the latest version of JDK supported by Java NetBeans to ensure compatibility and access to the latest features and improvements.
- **Java NetBeans IDE:** The system is developed using the Java NetBeans Integrated Development Environment (IDE), which provides a user-friendly interface for developing Java applications. Java NetBeans offers various tools and utilities for designing graphical user interfaces (GUIs), writing code, debugging, and testing applications.
- **Database Management System (DBMS):** The system relies on a DBMS for storing and managing data, including user information, product details, and transaction records. Commonly used DBMS options include MySQL, PostgreSQL, and SQLite. The choice of DBMS depends on factors such as scalability, performance, and compatibility with the existing infrastructure.
- **Libraries and Frameworks**: The system may utilize additional libraries and frameworks to enhance its functionality and performance. These may include JavaFX for creating rich graphical user interfaces, Apache POI for generating PDF reports, and JFreeChart for data visualization and analysis.

The Restaurant Management System is designed to be modular and extensible, allowing for easy integration with other software components and services commonly used in restaurant environments, such as point-of-sale (POS) systems, inventory management software, and online ordering platforms.

4. SOFTWARE DESIGN

4.1 Explanation:

The software design of the Restaurant Management System encompasses various components to ensure efficient functionality and user-friendly operation. The system is structured into modules that handle different aspects of restaurant management, including product management, billing, user authentication, and data analysis.

User Interface: The graphical user interface (GUI) is designed to be intuitive and easy to navigate, with clear labels and interactive elements for seamless interaction. It provides access to different functionalities through buttons, menus, and forms, ensuring a smooth user experience.

Database Management: The system utilizes a relational database management system (DBMS) to store and manage data related to users, products, transactions, and sales analysis. The database schema is designed to optimize data storage and retrieval, ensuring efficient performance even with large datasets.

Authentication Module: The authentication module handles user login and authentication, verifying user credentials against the database to grant access to authorized users. It includes functionality for user registration, password recovery, and security features such as encryption to protect sensitive data.

Product Management: The product management module allows users to add, update, and delete products from the inventory. It includes features for specifying product details such as name, price, quantity, and category, ensuring accurate and up-to-date product information.

Billing Module: The billing module generates bills for customer transactions, calculating the total cost based on the quantity and price of products purchased. It provides functionality for printing or exporting bills in PDF format for

customer records and accounting purposes.

Data Analysis: The data analysis module analyzes sales data to provide insights into product performance and trends. It includes features for generating reports, charts, and graphs to visualize sales data over time, allowing restaurant owners to make informed decisions about inventory management and pricing strategies.

4.2 Algorithim:

1. Login Authentication Algorithm:

Algorithm: authenticateUser(username, password)
Input:
- username: Username entered by the user
- password: Password entered by the user

Output:
- Boolean value: True if the user is authenticated successfully, False otherwise

Steps:
1. Connect to the User Database.
2. Execute the SQL query to retrieve the user's information based on the entered username.
   SELECT * FROM UserDatabase WHERE Username = username
3. If no record is found, return False.
4. If a record is found, compare the password provided with the password stored in the database for the corresponding username.
5. If the passwords match, return True; otherwise, return False.

2. Add New Product Algorithm:

Algorithm: addNewProduct(productName,price)
Input:
- productName: Name of the new product
- price: Price of the new product
Output:
- Boolean: True if the new product is added successfully, False otherwise
Steps:
1. Connect to the Product Database.
2. Insert the new product into the database.
3. If the operation is successful, return True; otherwise, return False.

3. Update Product Algorithm:

Algorithm: updateProduct(productName, price)
Input:
- productName: Name of the product to be updated
- price: New price of the product
Output:
- Boolean: True if the product is updated successfully, False otherwise
Steps:
1. Connect to the Product Database.
2. Update the price of the product.
3. If the operation is successful, return True; otherwise, return False.

Algorithm: deleteProduct(productName)
Input:
- productName: Name of the product to be deleted
Output:
- Boolean: True if the product is deleted successfully, False otherwise
Steps:
1. Connect to the Product Database.
2. Select the product and click on delete then it will be deleted from database.

4. **View Records Algorithm:**

Algorithm: viewRecords(selectedDate)
Input:
- selectedDate: Date selected by the user
Output:
- Records: Transactions on the selected date
Steps:
1. Connect to the Bill Database.
2. Retrieve records based on the selected date.
3. If records are found, return them; otherwise, return an empty result.

5. **Monthly Analysis Algorithm:**

Algorithm: monthlyAnalysis(selectedMonth)
Input:
- selectedMonth: Month selected by the user
Output:
- Analysis: Monthly sales analysis
Steps:
1. Connect to the Bill Database.
2. Retrieve records based on the selected month.
3. If records are found, return the analysis; otherwise, return an empty result.

6. **Weekly Average Algorithm:**

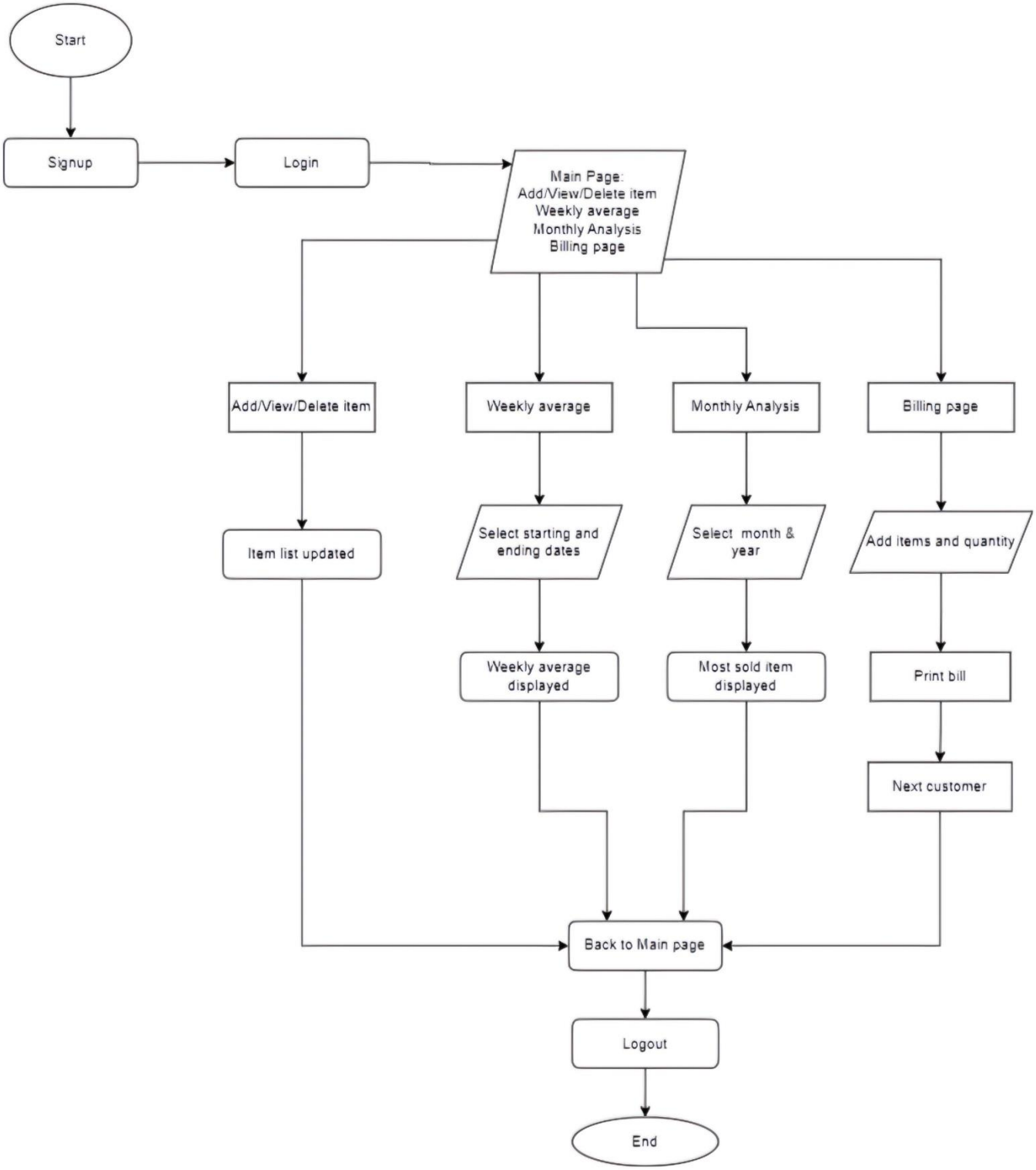
Algorithm: weeklyAverage(startDate, endDate)
Input:
- startDate: Start date selected by the user
- endDate: End date selected by the user
Output:
- Average: Weekly average of product quantity and sales within the selected date range

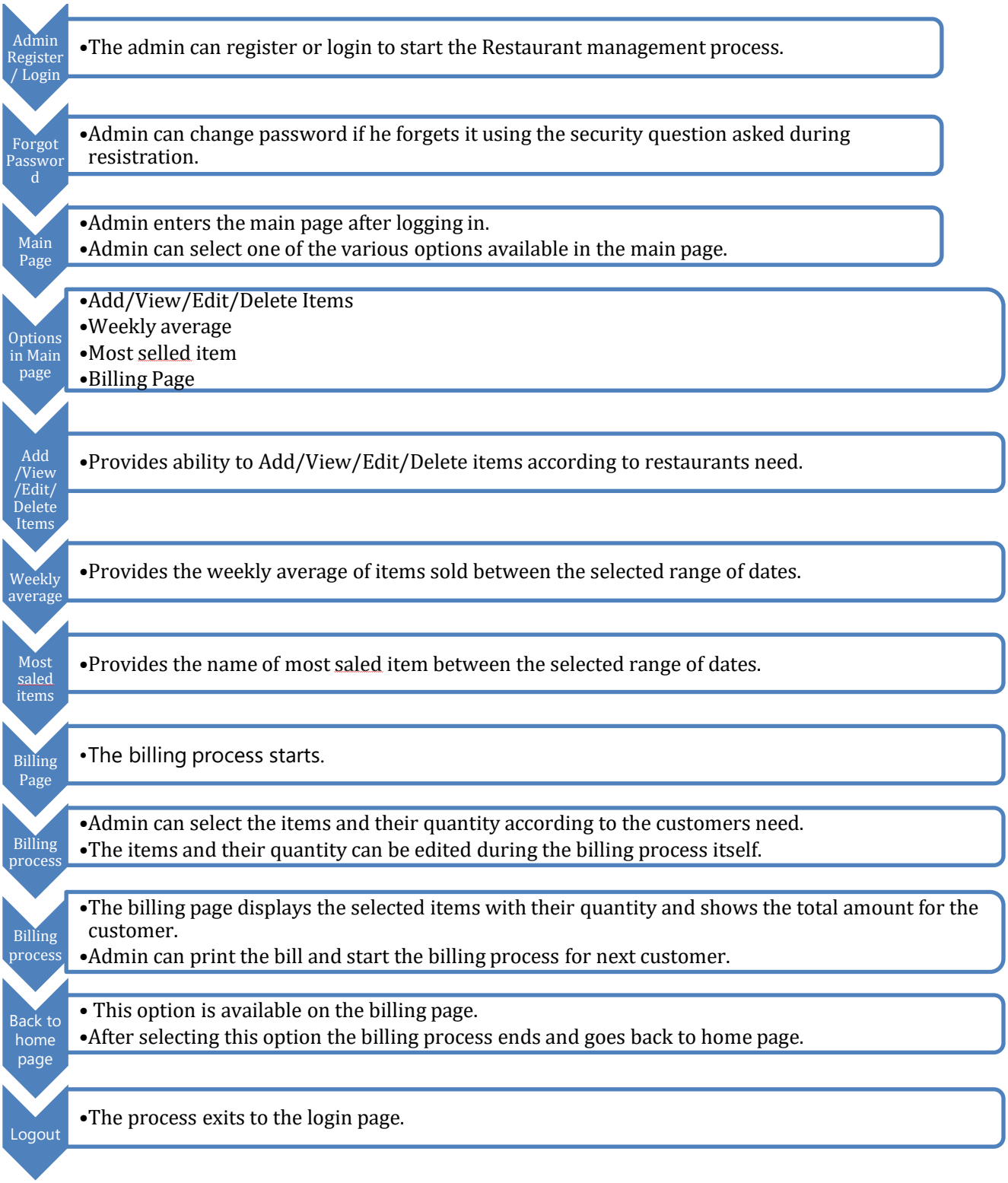
Steps:
1. Connect to the Bill Database.
2. Calculate the weekly average.
3. If records are found, return the weekly average; otherwise, return an empty result.

7. Billing Algorithm:

Algorithm: generateBill(productList)
Input:
- productList: List of products to be included in the bill
Output:
- Bill: PDF bill generated
Steps:
1. Initialize totalAmount = 0.
2. For each product in the productList:
   a. Retrieve the product details (name, quantity, price).
   b. Calculate the total price for the product (quantity * price).
   c. Add the total price to totalAmount.
3. Generate a PDF bill including the product details and the total amount.
4. Save the generated bill.
5. Print the bill.
6. End the algorithm.

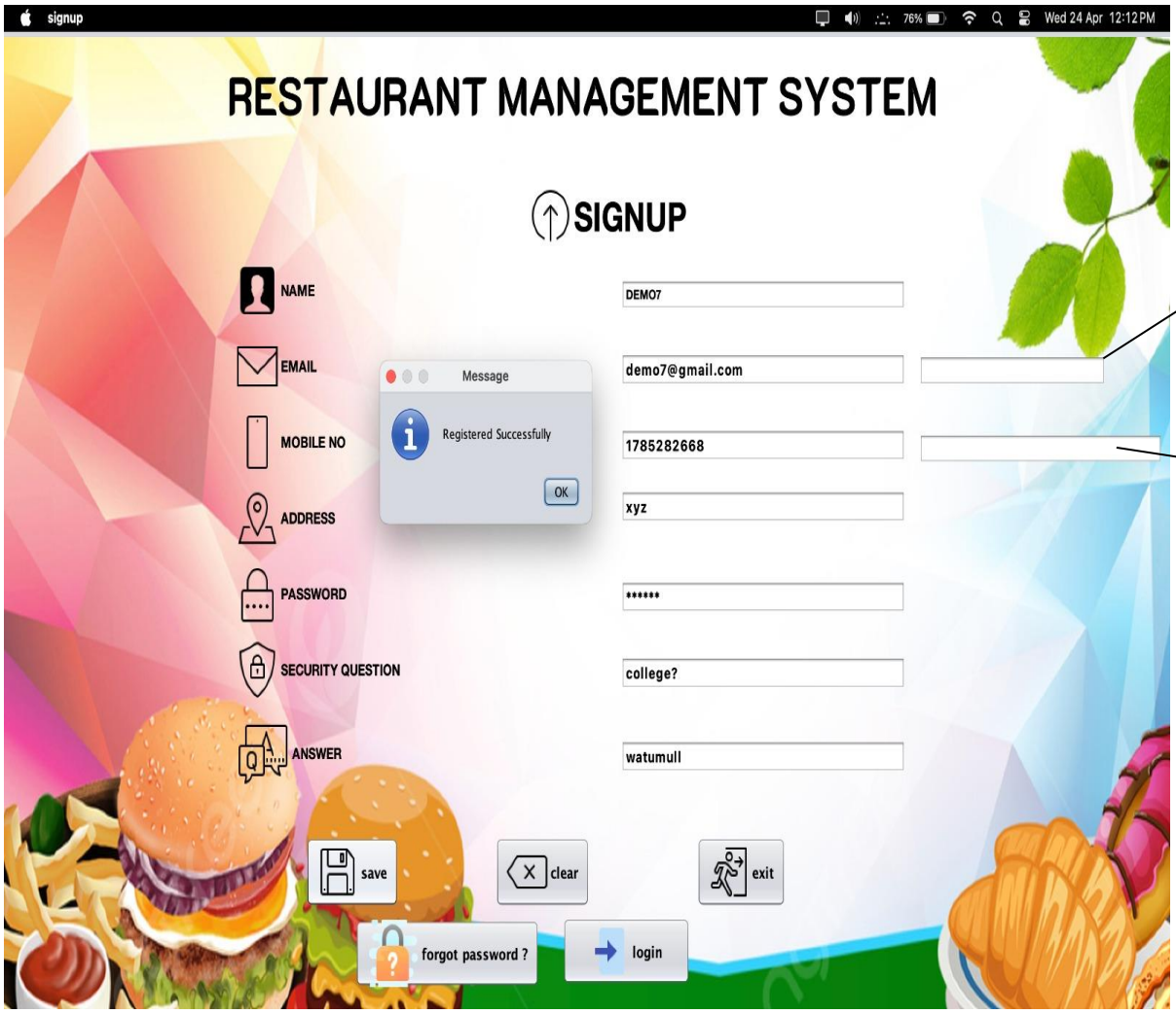
4.2 Flowchart/ER diagram:




Admin Register / Login
•The admin can register or login to start the Restaurant management process.
Forgot Password
•Admin can change password if he forgets it using the security question asked during resistration.
Main Page
•Admin enters the main page after logging in.
•Admin can select one of the various options available in the main page.
Options in Main page
•Add/View/Edit/Delete Items
•Weekly average
•Most selled item
•Billing Page
Add /View /Edit/ Delete Items
•Provides ability to Add/View/Edit/Delete items according to restaurants need.
Weekly average
•Provides the weekly average of items sold between the selected range of dates.
Most saled items
•Provides the name of most saled item between the selected range of dates.
Billing Page
•The billing process starts.
Billing process
•Admin can select the items and their quantity according to the customers need.
•The items and their quantity can be edited during the billing process itself.
Billing process
•The billing page displays the selected items with their quantity and shows the total amount for the customer.
•Admin can print the bill and start the billing process for next customer.
Back to home page
• This option is available on the billing page.
•After selecting this option the billing process ends and goes back to home page.
Logout
•The process exits to the login page.

5. RESULTS

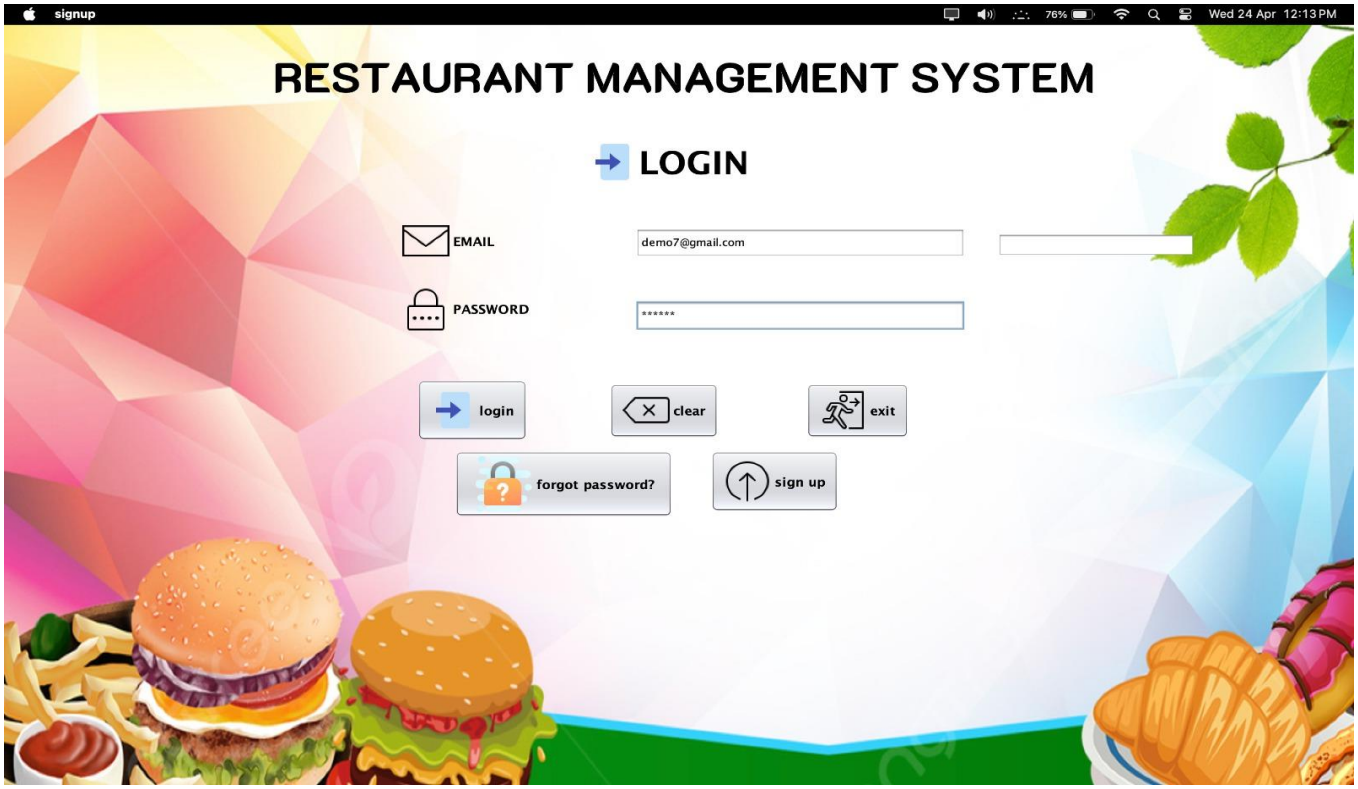
5.1 Screen Shots of test result or output:



Checking emailpattern if wrong then shows invalid email

Checking phoneno pattern if wrong then shows invalid phoneno should not excced more than 10 no's

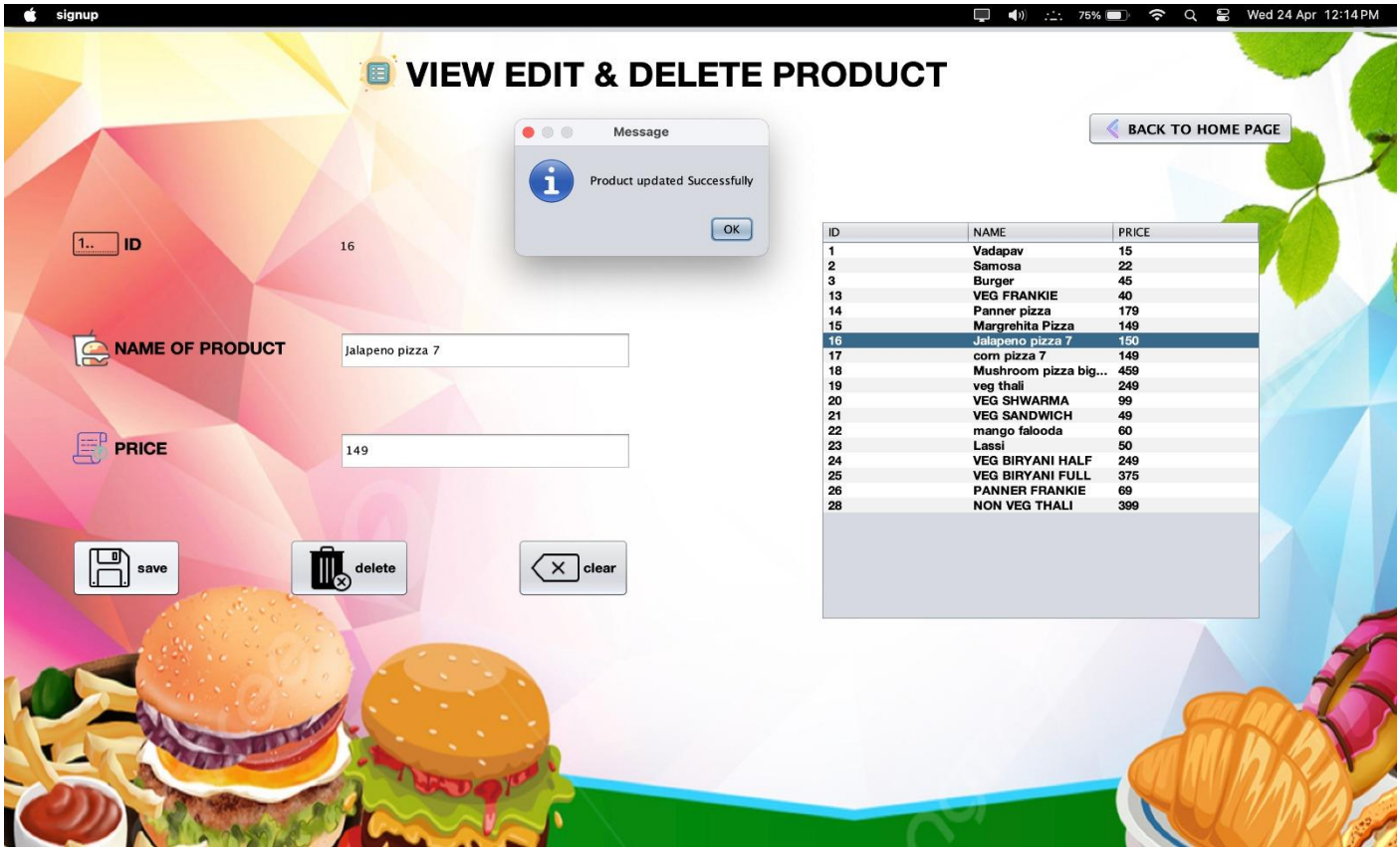
5.1.1 SIGNUP PAGE



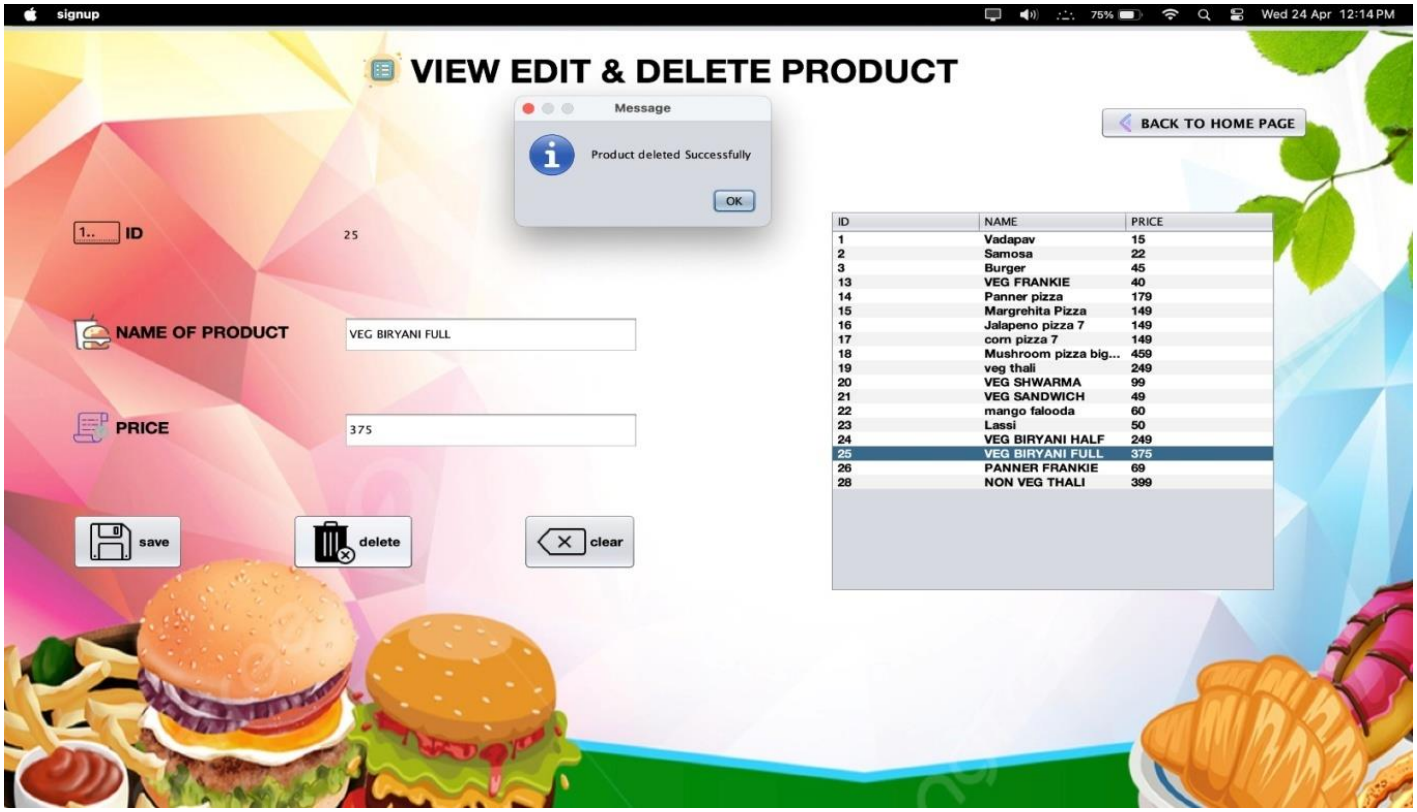
5.1.2 LOGIN PAGE

5.1.3 HOME PAGE(AFTER VALID MAIL AND PASSWORD)

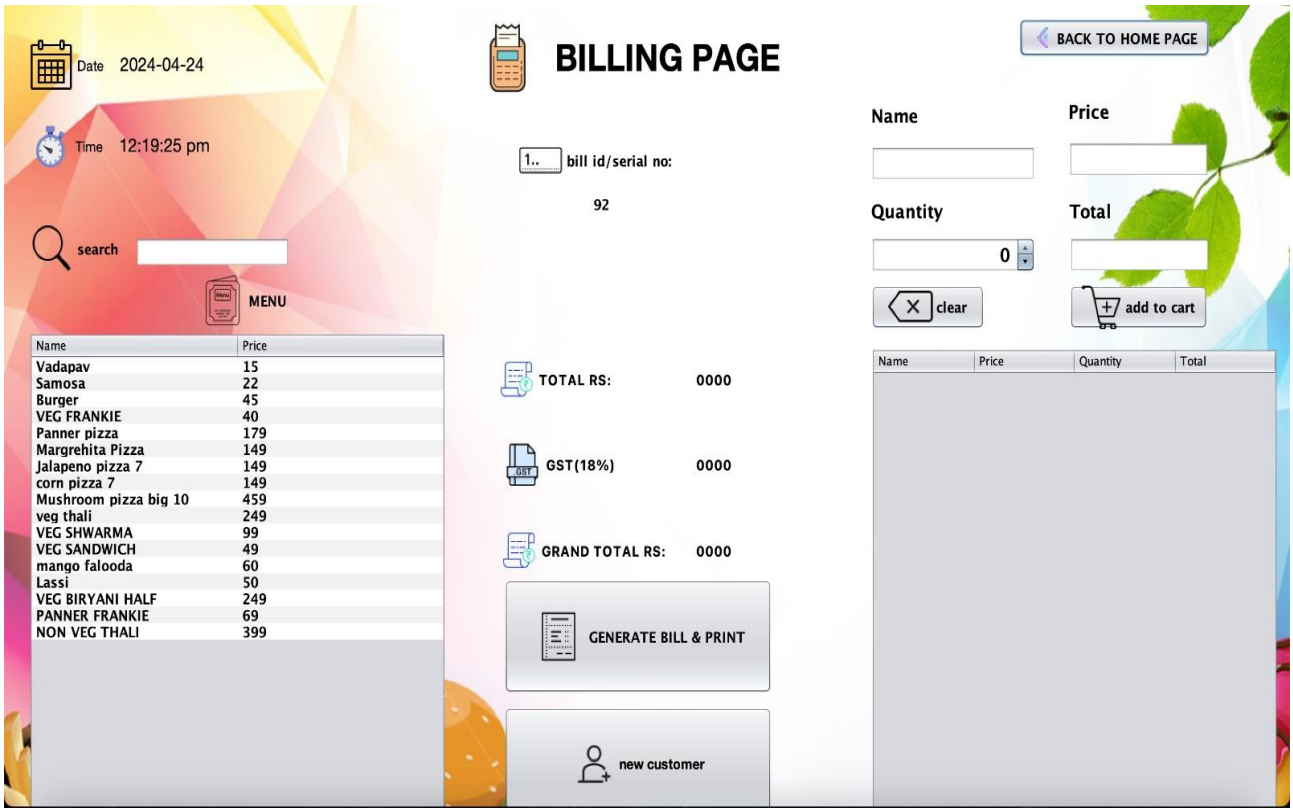
5.1.4 ADDING NEW PRODUCT (AFTER INPUT CLICK ON SAVE)



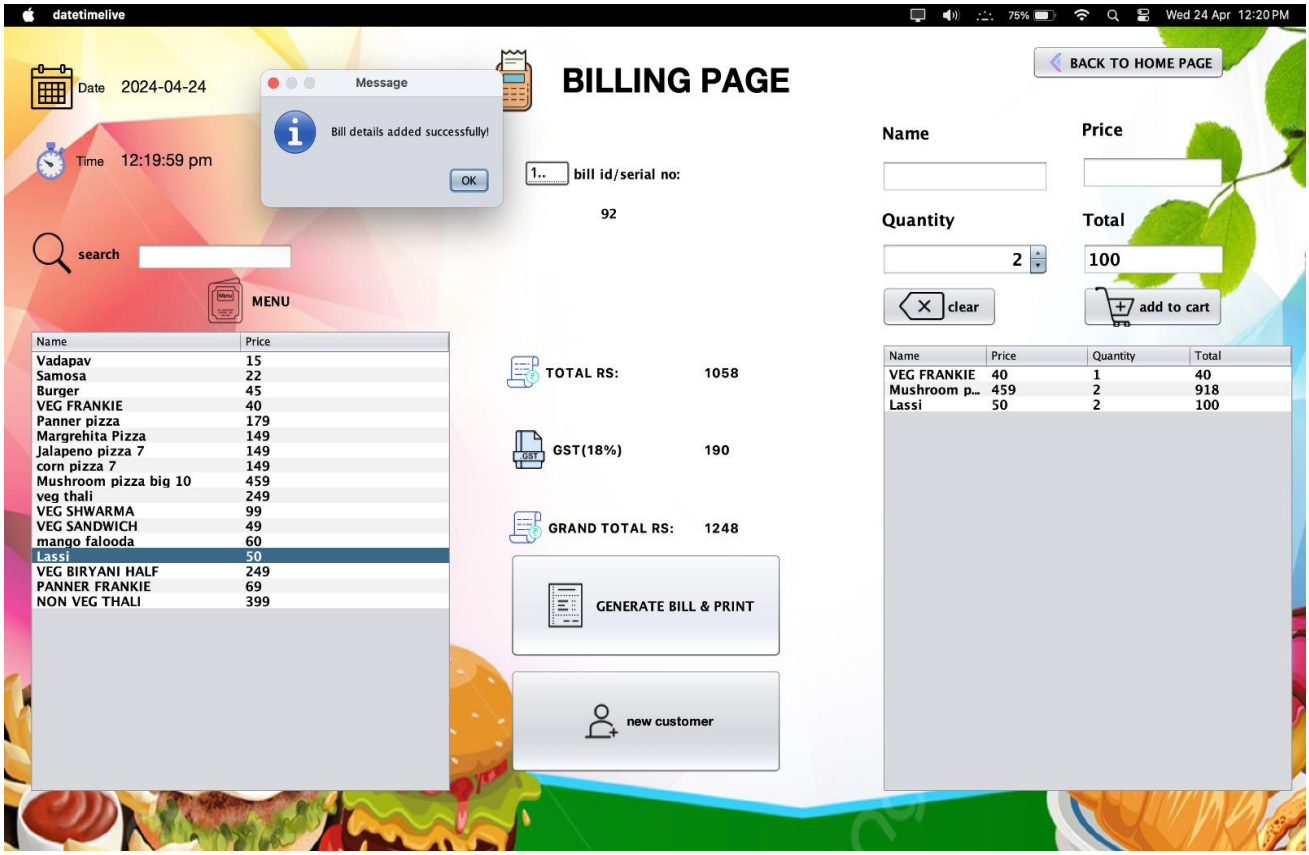
5.1.5 UPDATIG PRODUCT(SELECT PRODUCT AND CAN CHANGE NAME AND PRICE)



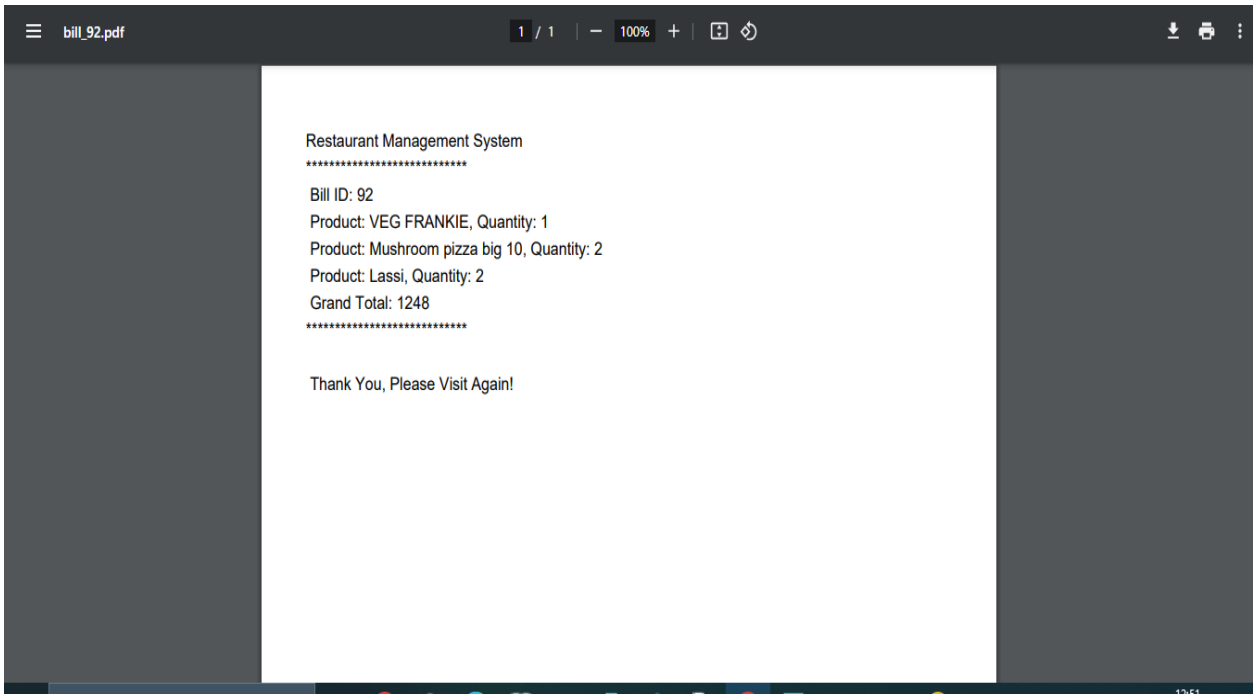
5.1.6 UPDATIG PRODUCT(SELECT PRODUCT AND CAN DELETE)



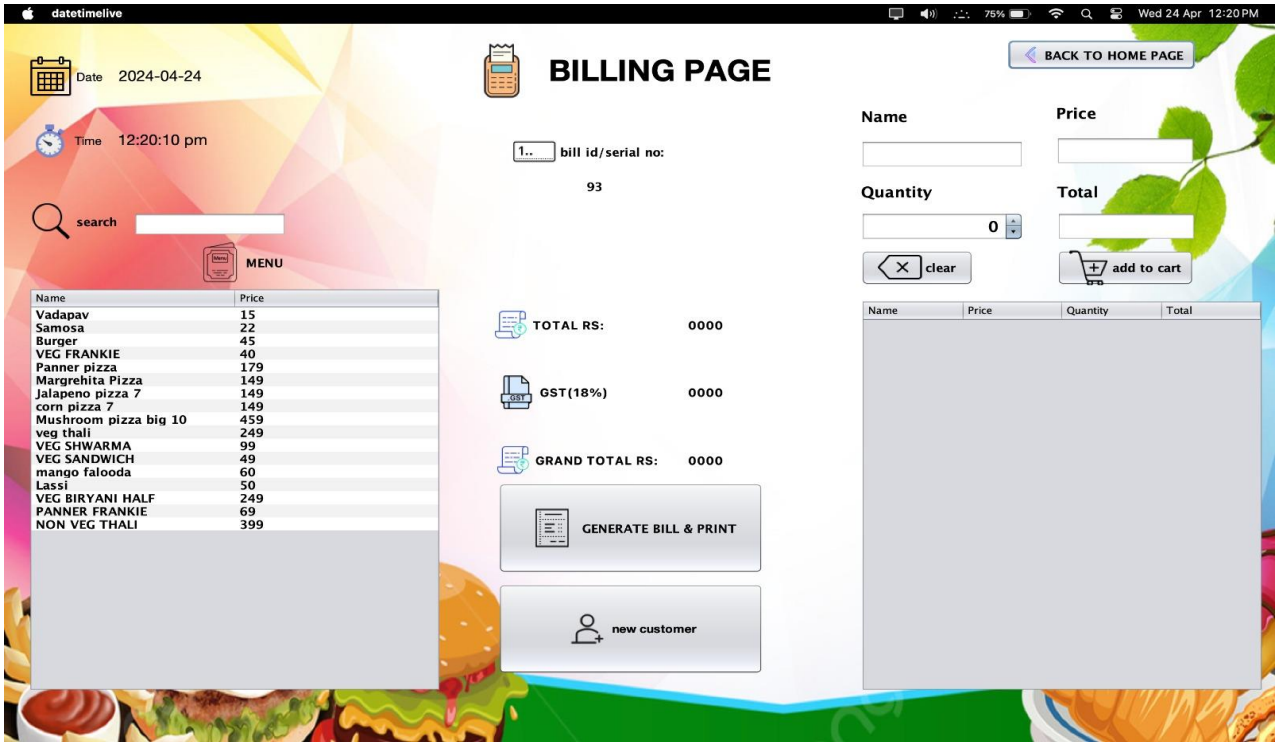
5.1.7 BILLING PAGE



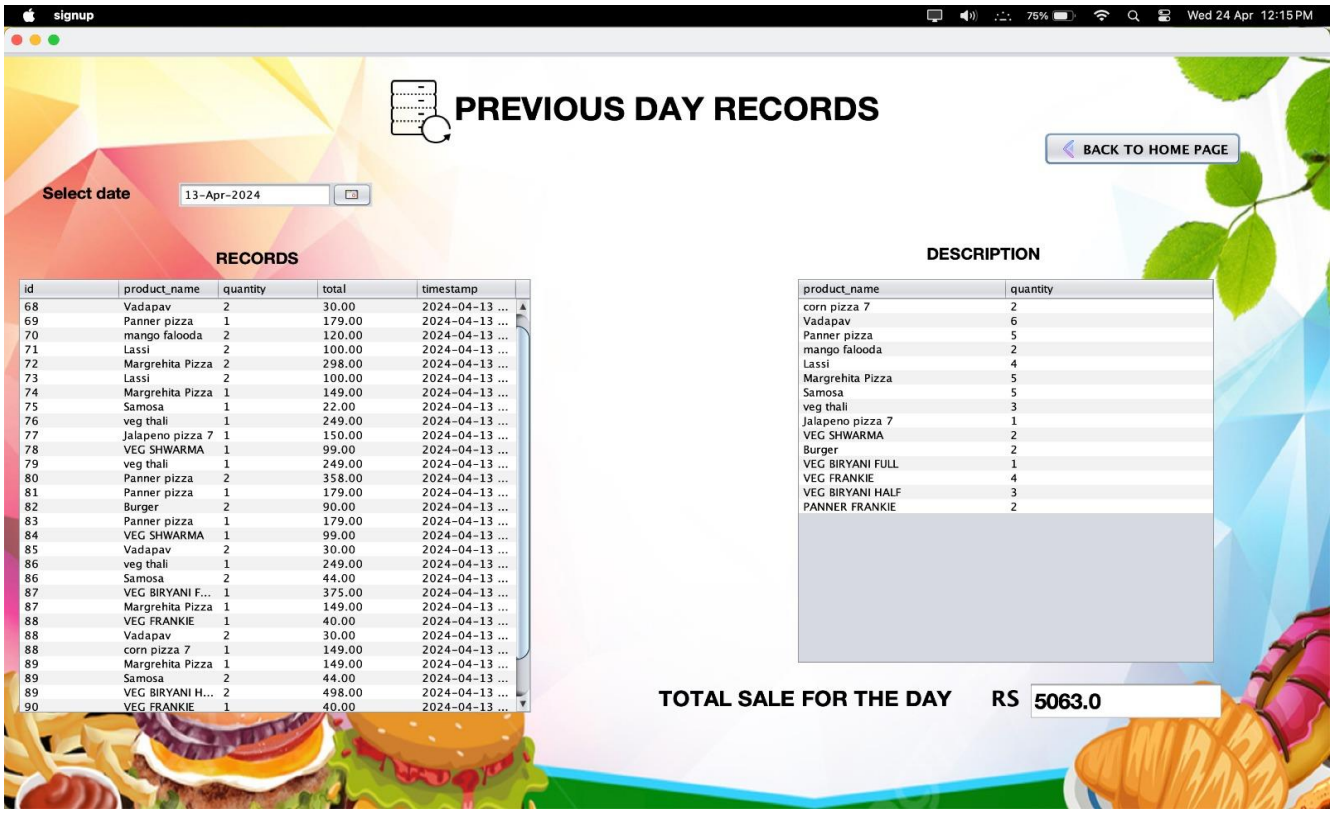
5.1.8 BILLING PAGE(FROM MENU PRODUCTS SELECTED AND ADD TO CART AFTER SEETING QUANTITY)



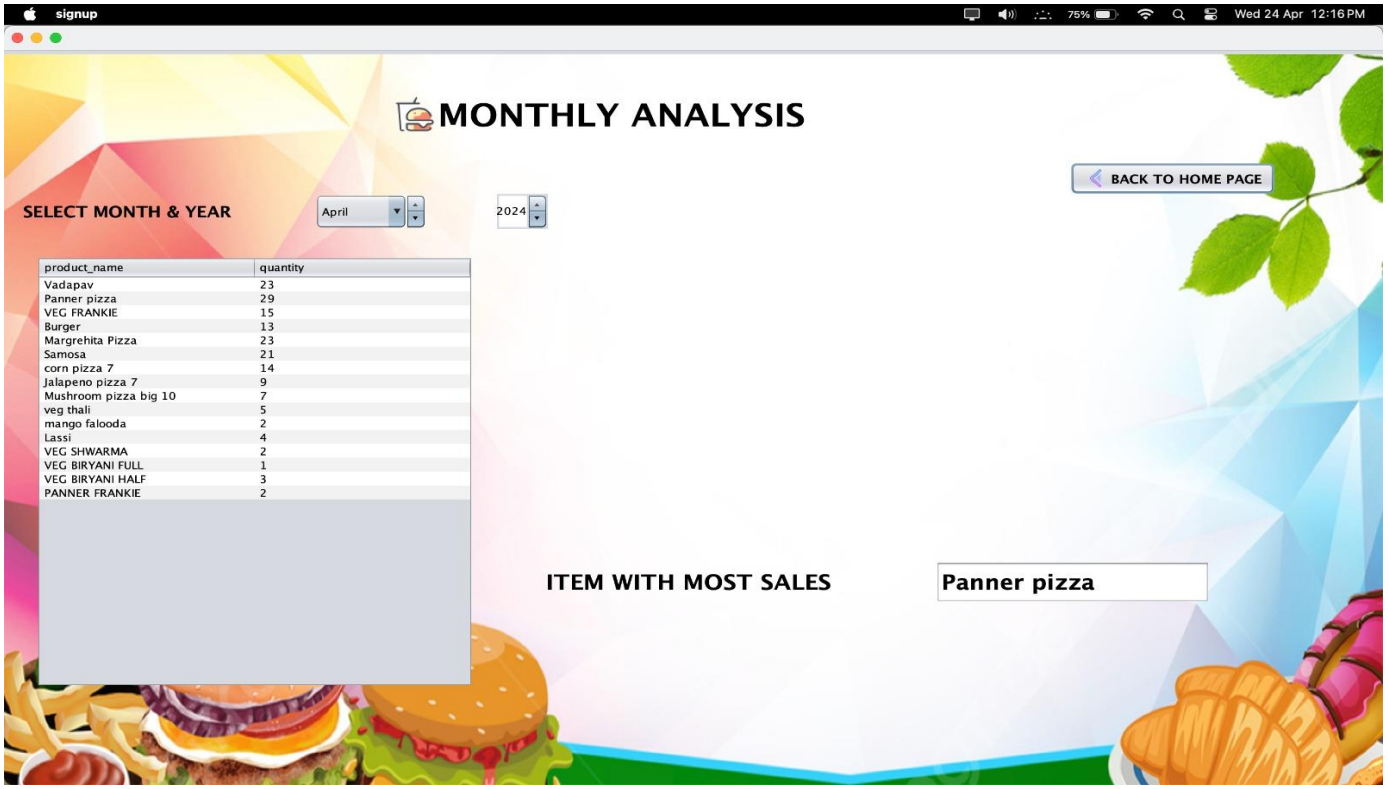
5.1.9 BILL PDF(AFTER ADDING IN CART CLICK GENERATE BILL AND PRINT AND THEN PDF IS SAVE WITH THE SAME BILL ID)



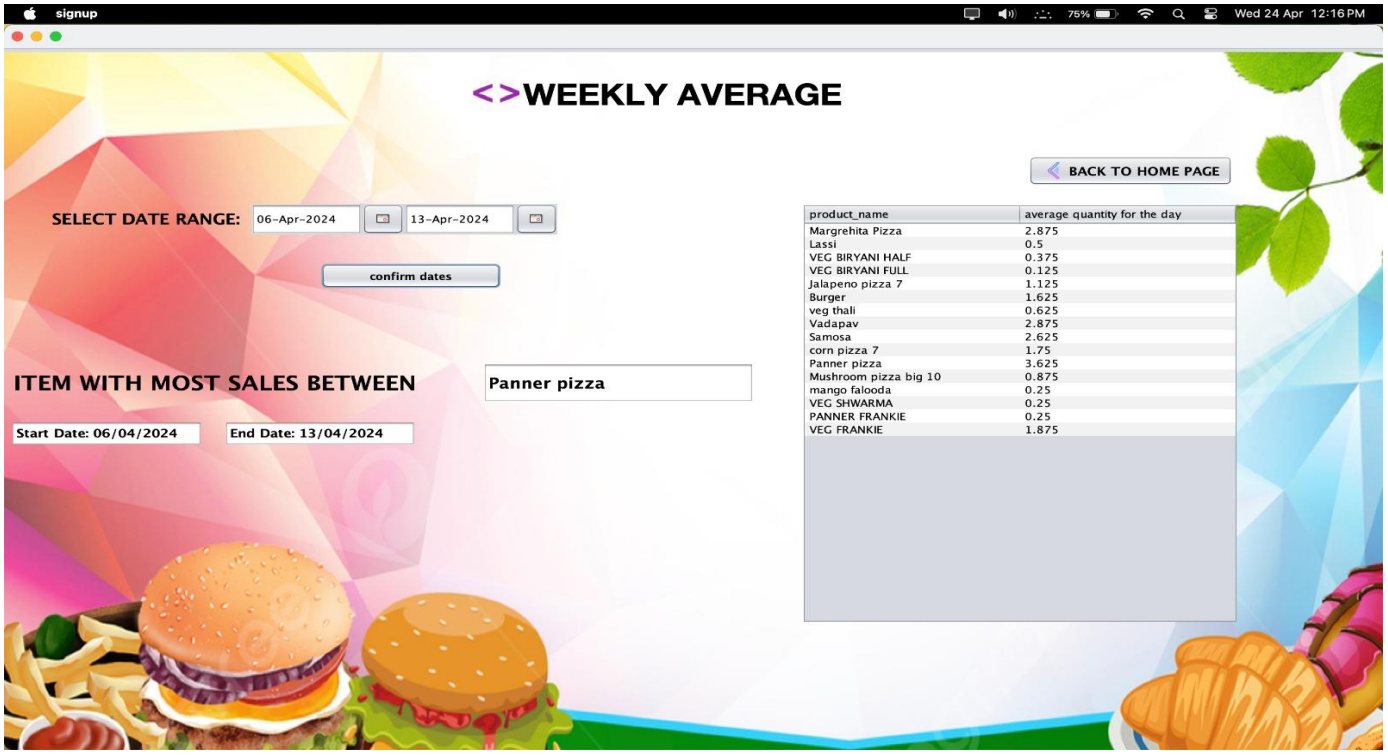
5.1.10 NEW CUSTOMER(AFTERCLICKING ON NEW CUSTOMER NEW BILL PAGE GENERETED)



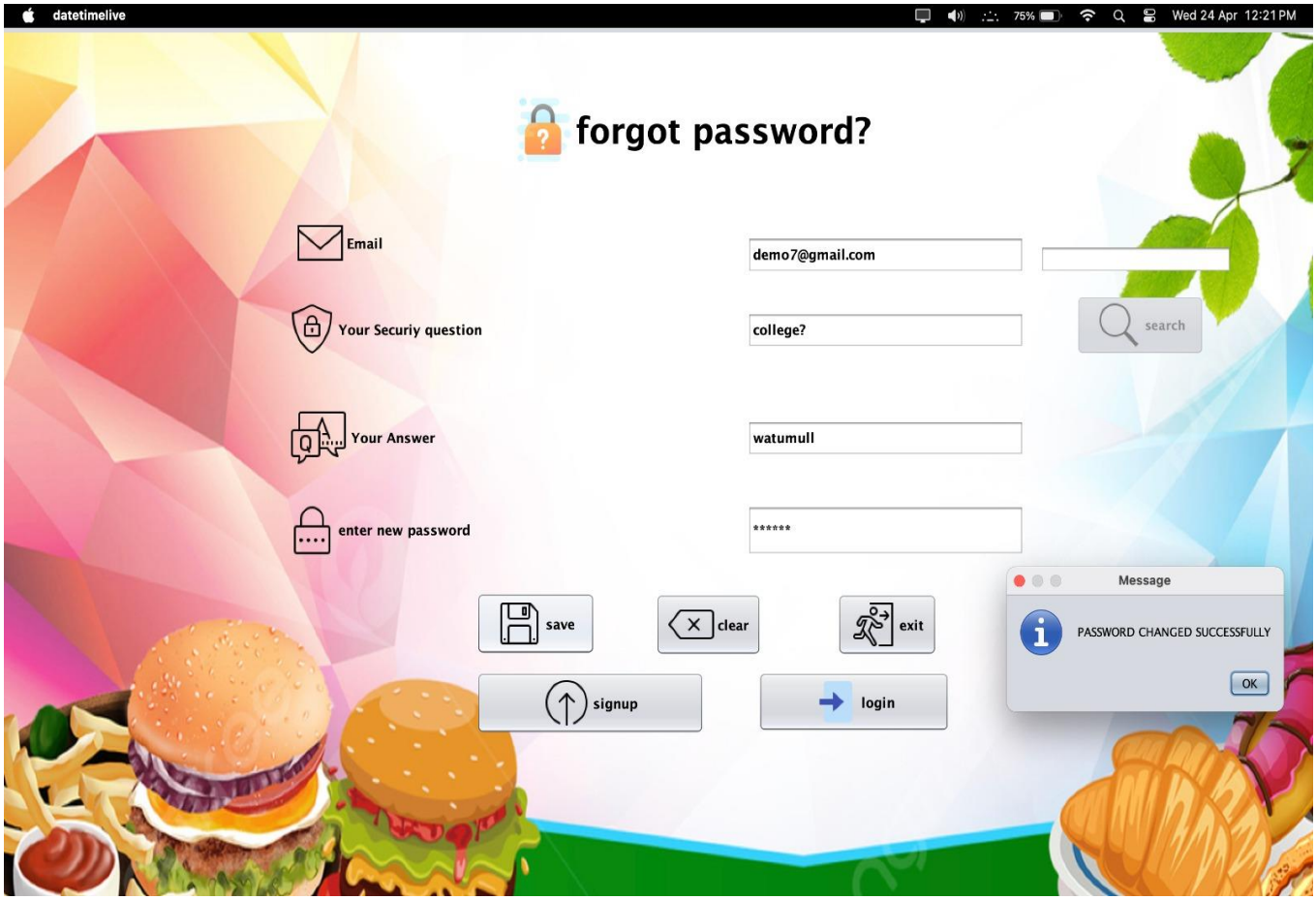
5.1.11 PREVIOUS DAY RECORDS(THIS SHOWS THAT DAY SALES WITH TOTAL QUANTITES AND ALL BILL GENERATED THAT DAY)



5.1.12 MONTHLY ANALYSIS(SELECTING MONTH IT WILL SHOW MOST SALED PRODUCT WITH QUANTITTES)



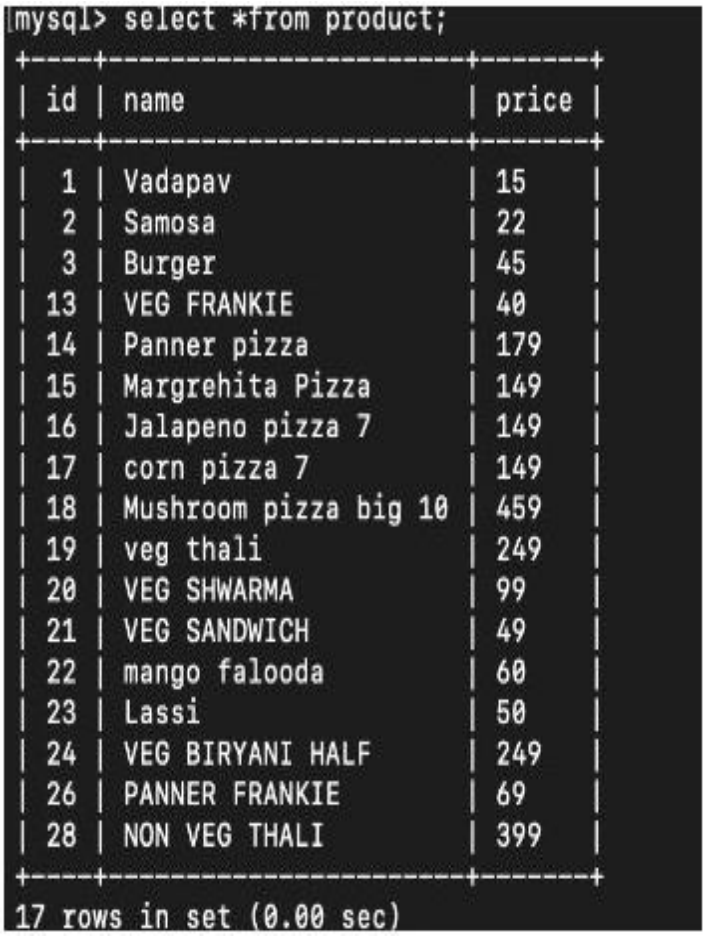
5.1.13 WEEKLY AVERAGE(CHHOSING DATE RANGE ,SHOWS THAT DAYS MOST SALED PRODUCT WITH AVERAGE QUANTITIES)



5.1.14 FOGOT PASSWORD(INPUT MAIL ID AND SEARCH FOR SECURUITY Q,AFTER GIVING RIGHT ANSWER CAN CHANGE PASSWORD)

```
mysql> select *from user;
+----+--------------+-----------------------+--------------+--------------+----------+------------------+----------+--------+
| id | name         | email                 | mobileNumber | address      | password | securityQuestion | answer   | status |
+----+--------------+-----------------------+--------------+--------------+----------+------------------+----------+--------+
|  1 | MADHUR SHINDE | madhur212004@gmail.com | 8108511335   | bhabdup east | 456789   | dog name?        | Bravo    | false  |
|  2 | AMAN SINGH   | aman12@gmail.com      | 2486346512   | bahabbhjsb   | 789456   | cartoon?         | tom      | false  |
|  3 | HARSH SAKPAL | harsh03@gmail.com     | 8722155817   | bhandupppp   | 123789   | school           | menon    | false  |
|  4 | Aryan Balani | aryan@gmail.com       | 4538445491   | ulhasnagar   | 123456   | height?          | 1.5foot  | false  |
|  5 | sumit        | sumit@gmail.com       | 9988776659   | mumbai       | 12345    | sport?           | cricket  | false  |
|  6 | DEMO         | demo@gmail.com        | 8794567894   | ulhasnagar   | 147258   | cartoon          | doremon  | false  |
|  7 | demo5        | demo5@gmail.com       | 7856445621   | dmo133134    | 789456   | ui               | 6.1      | false  |
|  8 | DEMO7        | demo7@gmail.com       | 1785282668   | xyz          | 456789   | college?         | watumull | false  |
+----+--------------+-----------------------+--------------+--------------+----------+------------------+----------+--------+
8 rows in set (0.00 sec)
```

5.1.15 USER DATABASE(IN MYSQL)



```
mysql> select *from product;
+----+-----------------------+-------+
| id | name                  | price |
+----+-----------------------+-------+
|  1 | Vadapav               | 15    |
|  2 | Samosa                | 22    |
|  3 | Burger                | 45    |
| 13 | VEG FRANKIE           | 40    |
| 14 | Panner pizza          | 179   |
| 15 | Margrehita Pizza      | 149   |
| 16 | Jalapeno pizza 7      | 149   |
| 17 | corn pizza 7          | 149   |
| 18 | Mushroom pizza big 10 | 459   |
| 19 | veg thali             | 249   |
| 20 | VEG SHWARMA           | 99    |
| 21 | VEG SANDWICH          | 49    |
| 22 | mango falooda         | 60    |
| 23 | Lassi                 | 50    |
| 24 | VEG BIRYANI HALF      | 249   |
| 26 | PANNER FRANKIE        | 69    |
| 28 | NON VEG THALI         | 399   |
+----+-----------------------+-------+
17 rows in set (0.00 sec)
```

5.1.16 PRODUCT DATABASE WITH NAME ,ID AND PRICE(IN MYSQL)

```
mysql> select *from bill;
+----+--------------------+----------+---------+---------------------+
| id | product_name       | quantity | total   | timestamp           |
+----+--------------------+----------+---------+---------------------+
|  3 | Vadapav            |        2 |   82.00 | 2024-04-09 03:48:27 |
|  4 | Panner pizza       |        1 |  500.00 | 2024-04-09 03:55:46 |
|  5 | Vadapav            |        2 |  500.00 | 2024-04-09 03:55:47 |
|  6 | VEG FRANKIE        |        2 |  500.00 | 2024-04-09 03:55:49 |
|  7 | Burger             |        3 |  500.00 | 2024-04-09 03:55:51 |
|  8 | Panner pizza       |        1 |  500.00 | 2024-04-09 03:56:58 |
|  9 | Vadapav            |        2 |  500.00 | 2024-04-09 03:56:59 |
| 10 | VEG FRANKIE        |        2 |  500.00 | 2024-04-09 03:57:00 |
| 11 | Burger             |        3 |  500.00 | 2024-04-09 03:57:01 |
| 12 | Margrehita Pizza   |        1 |  159.00 | 2024-04-09 04:01:23 |
| 13 | Burger             |        2 |  159.00 | 2024-04-09 04:01:24 |
| 14 | Samosa             |        1 |  448.00 | 2024-04-09 04:07:36 |
| 15 | Panner pizza       |        2 |  448.00 | 2024-04-09 04:07:37 |
| 16 | Vadapav            |        1 |  193.00 | 2024-04-09 04:11:58 |
| 17 | Margrehita Pizza   |        1 |  193.00 | 2024-04-09 04:12:00 |
| 18 | Margrehita Pizza   |        4 |  219.00 | 2024-04-09 04:17:36 |
| 19 | Margrehita Pizza   |        4 |  219.00 | 2024-04-09 04:18:24 |
| 20 | Vadapav            |        3 |  209.00 | 2024-04-09 04:18:41 |
| 21 | Margrehita Pizza   |        3 |  313.00 | 2024-04-09 04:20:40 |
| 22 | Burger             |        1 |   45.00 | 2024-04-09 04:35:16 |
| 23 | Panner pizza       |        1 |  224.00 | 2024-04-09 04:35:17 |
| 24 | Margrehita Pizza   |        3 |  313.00 | 2024-04-09 04:36:52 |
| 25 | Samosa             |        7 |  465.00 | 2024-04-09 19:24:48 |
| 26 | Panner pizza       |        1 |  179.00 | 2024-04-09 22:08:09 |
| 27 | Vadapav            |        3 |   70.00 | 2024-04-09 22:09:28 |
| 28 | Panner pizza       |        3 |  398.00 | 2024-04-10 02:14:04 |
| 29 | Panner pizza       |        7 | 1135.00 | 2024-04-10 05:06:15 |
| 30 | Panner pizza       |        1 |  179.00 | 2024-04-11 03:54:45 |
| 31 | VEG FRANKIE        |        3 |  379.00 | 2024-04-11 03:58:49 |
| 32 | corn pizza 7       |        1 |  149.00 | 2024-04-11 04:28:03 |
| 33 | VEG FRANKIE        |        1 |   40.00 | 2024-04-11 04:31:31 |
| 34 | Panner pizza       |        1 |  179.00 | 2024-04-11 04:34:29 |
| 35 | corn pizza 7       |        2 |  298.00 | 2024-04-11 04:34:31 |
| 36 | VEG FRANKIE        |        1 |   40.00 | 2024-04-11 04:38:09 |
| 37 | Panner pizza       |        2 |  358.00 | 2024-04-11 04:38:11 |
| 38 | Jalapeno pizza 7   |        1 |  150.00 | 2024-04-11 04:43:17 |
| 39 | Samosa             |        2 |   44.00 | 2024-04-11 04:43:18 |
| 40 | Mushroom pizza big 10 |     1 |  459.00 | 2024-04-11 04:43:19 |
| 41 | Mushroom pizza big 10 |     1 |  459.00 | 2024-04-11 04:46:56 |
| 42 | Samosa             |        2 |   44.00 | 2024-04-11 04:46:58 |
| 43 | Samosa             |        3 |  503.00 | 2024-04-11 04:46:59 |
| 44 | Jalapeno pizza 7   |        2 |  300.00 | 2024-04-11 04:48:47 |
| 45 | Margrehita Pizza   |        2 |  757.00 | 2024-04-11 04:51:00 |
| 46 | corn pizza 7       |        1 |  149.00 | 2024-04-11 04:57:51 |
| 47 | Burger             |        2 |   90.00 | 2024-04-11 04:57:53 |
| 48 | corn pizza 7       |        2 |  298.00 | 2024-04-11 04:57:54 |
| 49 | corn pizza 7       |        5 |  537.00 | 2024-04-11 04:57:54 |
| 50 | Jalapeno pizza 7   |        1 |  150.00 | 2024-04-11 05:04:16 |
| 51 | Jalapeno pizza 7   |        2 |  300.00 | 2024-04-11 05:04:18 |
| 52 | Vadapav            |        2 |   30.00 | 2024-04-11 05:04:18 |
| 53 | Panner pizza       |        1 |  179.00 | 2024-04-11 05:08:02 |
| 54 | Mushroom pizza big 10 |     1 |  459.00 | 2024-04-11 05:08:03 |
| 55 | Vadapav            |        2 |   30.00 | 2024-04-11 05:08:04 |
| 56 | Panner pizza       |        1 |  179.00 | 2024-04-11 05:08:16 |
| 57 | Jalapeno pizza 7   |        1 |  150.00 | 2024-04-11 05:14:36 |
| 58 | Mushroom pizza big 10 |     2 |  918.00 | 2024-04-11 05:14:37 |
| 59 | Panner pizza       |        1 |  179.00 | 2024-04-11 05:16:21 |
| 60 | veg thali          |        2 |  598.00 | 2024-04-11 05:16:22 |
| 61 | Jalapeno pizza 7   |        1 |  150.00 | 2024-04-11 14:21:51 |
| 62 | Samosa             |        1 |   22.00 | 2024-04-11 14:25:08 |
| 63 | corn pizza 7       |        1 |  149.00 | 2024-04-11 14:25:09 |
| 64 | VEG FRANKIE        |        2 |   80.00 | 2024-04-11 14:25:10 |
| 65 | Panner pizza       |        1 |  179.00 | 2024-04-11 15:05:18 |
| 66 | Mushroom pizza big 10 |     2 |  918.00 | 2024-04-11 15:05:20 |
| 67 | corn pizza 7       |        1 |  149.00 | 2024-04-13 01:04:35 |
| 68 | Vadapav            |        2 |   30.00 | 2024-04-13 01:04:36 |
| 69 | Panner pizza       |        1 |  179.00 | 2024-04-13 01:16:01 |
| 70 | mango falooda      |        2 |  120.00 | 2024-04-13 01:16:02 |
| 71 | Lassi              |        2 |  100.00 | 2024-04-13 01:16:03 |
| 72 | Margrehita Pizza   |        2 |  298.00 | 2024-04-13 01:16:03 |
| 73 | Lassi              |        2 |  100.00 | 2024-04-13 01:20:07 |
| 74 | Margrehita Pizza   |        1 |  149.00 | 2024-04-13 01:20:08 |
| 75 | Samosa             |        1 |   22.00 | 2024-04-13 01:20:08 |
| 76 | veg thali          |        1 |  249.00 | 2024-04-13 01:20:09 |
| 77 | Jalapeno pizza 7   |        1 |  150.00 | 2024-04-13 01:21:23 |
| 78 | VEG SHWARMA        |        1 |   99.00 | 2024-04-13 01:21:24 |
| 79 | veg thali          |        1 |  249.00 | 2024-04-13 01:52:09 |
| 80 | Panner pizza       |        2 |  358.00 | 2024-04-13 01:52:10 |
| 81 | Panner pizza       |        1 |  179.00 | 2024-04-13 01:54:15 |
| 82 | Burger             |        2 |   90.00 | 2024-04-13 01:54:16 |
| 83 | Panner pizza       |        1 |  179.00 | 2024-04-13 02:07:19 |
| 84 | VEG SHWARMA        |        1 |   99.00 | 2024-04-13 02:07:21 |
| 85 | Vadapav            |        2 |   30.00 | 2024-04-13 02:07:21 |
| 86 | veg thali          |        1 |  249.00 | 2024-04-13 02:09:21 |
| 86 | Samosa             |        2 |   44.00 | 2024-04-13 02:09:22 |
| 87 | VEG BIRYANI FULL   |        1 |  375.00 | 2024-04-13 02:18:39 |
| 87 | Margrehita Pizza   |        1 |  149.00 | 2024-04-13 02:18:40 |
| 88 | VEG FRANKIE        |        1 |   40.00 | 2024-04-13 02:20:38 |
| 88 | Vadapav            |        2 |   30.00 | 2024-04-13 02:20:39 |
| 88 | corn pizza 7       |        1 |  149.00 | 2024-04-13 02:20:40 |
| 89 | Margrehita Pizza   |        1 |  149.00 | 2024-04-13 12:58:08 |
| 89 | Samosa             |        2 |   44.00 | 2024-04-13 12:58:09 |
| 89 | VEG BIRYANI HALF   |        2 |  498.00 | 2024-04-13 12:58:10 |
| 90 | VEG FRANKIE        |        1 |   40.00 | 2024-04-13 14:03:04 |
| 90 | PANNER FRANKIE     |        2 |  138.00 | 2024-04-13 14:03:05 |
| 91 | VEG FRANKIE        |        2 |   80.00 | 2024-04-13 14:55:14 |
| 91 | VEG BIRYANI HALF   |        1 |  249.00 | 2024-04-13 14:55:19 |
| 92 | VEG FRANKIE        |        1 |   40.00 | 2024-04-24 12:19:48 |
| 92 | Mushroom pizza big 10 |     2 |  918.00 | 2024-04-24 12:19:49 |
| 92 | Lassi              |        2 |  100.00 | 2024-04-24 12:19:51 |
+----+--------------------+----------+---------+---------------------+
100 rows in set (0.00 sec)
```


5.1.17 BILL DATABASE WITH ID ,NAME,QANTITIES,TIME(IN MYSQL)

6. Future Enhancements and Future Scope of Project

6.1 Future Enhancements:

Future enhancements to the Restaurant Management System aim to further improve its functionality, usability, and scalability, keeping pace with evolving technology and industry trends. These enhancements may include:

- **Integration with Online Ordering Platforms**: Integrating the system with online ordering platforms allows restaurants to seamlessly manage both in-house and online orders from a single interface. This feature enhances customer convenience and expands the restaurant's reach to a wider audience.

- **Mobile Application Development:** Developing a mobile application version of the system enables restaurant staff to manage operations remotely, enhancing flexibility and efficiency. Mobile applications can provide features such as real-time notifications, order tracking, and inventory management on-the-go.

- **Enhanced Data Analysis**: Implementing advanced data analysis techniques such as predictive analytics and machine learning algorithms allows restaurants to gain deeper insights into customer preferences, market trends, and sales forecasting. This enables more informed decision-making and targeted marketing strategies.

- **Inventory Optimization**: Introducing inventory optimization algorithms helps restaurants optimize stock levels, minimize wastage, and reduce costs. Predictive algorithms can analyze past sales data and seasonal trends to forecast demand accurately, ensuring optimal inventory management.

- **Customer Relationship Management (CRM) Integration:** Integrating CRM functionalities allows restaurants to effectively manage customer relationships, personalize marketing campaigns, and track customer feedback. This fosters customer loyalty and enhances the overall dining experience.

- **Enhanced Security Features**: Implementing advanced security features such as biometric authentication, role-based access control, and data encryption ensures the confidentiality, integrity, and availability of sensitive information, protecting against potential security threats and data breaches.

6.2 Future Scope:

The future scope of the Restaurant Management System extends beyond its current capabilities, offering opportunities for expansion and adaptation to emerging trends and market demands. The system's future scope includes:

- **Internationalization and Localization**: Adapting the system to support multiple languages, currencies, and regulatory requirements enables its deployment in diverse geographical locations, catering to the global restaurant industry.

- **Integration with IoT Devices**: Integrating the system with Internet of Things (IoT) devices such as smart kitchen appliances, inventory sensors, and point-of-sale terminals enhances automation and real-time monitoring capabilities, improving operational efficiency and accuracy.

- **Cloud-Based Deployment**: Transitioning to a cloud-based deployment model offers scalability, flexibility, and accessibility advantages, allowing

restaurants to access the system from anywhere with internet connectivity and reducing infrastructure costs.

- **Blockchain Technology Integration**: Leveraging blockchain technology for secure and transparent transaction processing, supply chain management, and loyalty programs enhances trust, traceability, and accountability within the restaurant ecosystem.

- **Virtual Reality (VR) Dining Experience**: Incorporating virtual reality technology into the dining experience offers customers immersive and interactive dining experiences, such as virtual tours of the restaurant, interactive menus, and virtual cooking classes.

- **Sustainability Initiatives**: Implementing sustainability initiatives such as waste reduction, energy efficiency, and eco-friendly packaging solutions aligns with growing consumer preferences for environmentally conscious dining options and contributes to a positive brand image.

- **Collaborative Partnerships**: Establishing partnerships with food delivery aggregators, payment gateways, and technology providers expands the system's ecosystem, enabling seamless integration with external services and enhancing value proposition for stakeholders.

7. Conclusion

The Restaurant Management System provides an efficient and user-friendly solution for managing restaurant operations effectively, leading to improved customer satisfaction and increased profitability. This system represents a significant step forward in modernizing restaurant management, addressing the challenges encountered in traditional methods. Developed to be scalable and adaptable, the system's user-friendly Graphical User Interface (GUI) ensures easy navigation for restaurant staff, regardless of their technical expertise.

Implementing a relational database management system (DBMS) ensures efficient storage and management of data related to users, products, transactions, and sales analysis. The authentication module ensures only authorized users can access the system, with features for user registration, password recovery, and security encryption to protect sensitive data. The product management module allows adding, updating, and deleting products from the inventory, ensuring accurate and up-to-date product information.

The billing module generates bills for customer transactions, calculating the total cost based on the quantity and price of products purchased. The data analysis module analyzes sales data to provide insights into product performance and trends, allowing restaurant owners to make informed decisions about inventory management and pricing strategies. Future enhancements may include integration with online ordering platforms, mobile application development, enhanced data analysis, inventory optimization, customer relationship management (CRM) integration, and enhanced security features. In conclusion, the Restaurant Management System represents a significant step forward in modernizing restaurant management, catering to the evolving needs of the industry.

Bibliography

[1] Code-Projects.org. (n.d.). Restaurant order management system in Java and NetBeans with source code. [Online]. Available: **https://code-projects.org/restaurant-order-management-system-in-java-and-netbeans-with-source-code/**

[2] Copy Assignment. (n.d.). Restaurant management system project in Java. [Online]. Available: **https://copyassignment.com/restaurant-management-system-project-in-java/**

[3] CodeGuid. (2023, July 14). The restaurant management system project in Java Swing with source code. [Online]. Available: **https://codeguid.com/2023/07/14/the-restaurant-management-system-project-in-java-swing-with-source-code/**

Acknowledgement

We have great pleasure in presenting the mini project report on "*RESTAURANT MANAGEMENT SYSTEM*".. We take this opportunity to express our sincere thanks towards our guide Prof.Dhananjay Raut Department of Computer Engineering, Watumull Institute,for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement,support and guidance throughout the development of project..

We thank Prof. Nilesh Mehta Head of the Department,Computer Engineering, Watumull Institute for his encouragement during progress meeting and providing guidelines to write this report.

We thank Prof.Avinash Gondal Principal,Watumull Institute for his Valuable suggestions and constant support.

We also thank the entire staff of Watumull Institute of Electronics Engineering and Computer Technology for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of Watumull Institute of Electronics Engineering and Computer Technology for their encouragement.

MADHUR VINOD SHINDE (97)
HARSH JAYENDRA SAKPAL (85)
AMAN SURESH SINGH (101)
SUJAL MANOJ SINGH (103)