

Mini Project Report

of

Database Systems Lab (CSE 2262)

RESTAURANT MANAGEMNT SYSTEM

SUBMITTED

BY

NAME : ANANT AGARWAL

ROLL NO : 39

REG NO : 220962264

SECTION: CSE(AIML)-A

NAME : MEDHA CHAWLA

ROLL NO: 55

REG NO: 220962340

SECTION: CSE(AIML)-A

Department of Computer Science and Engineering

**Manipal Institute of Technology, Manipal.**

**April 2024**



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Manipal

17/04/2024

CERTIFICATE

This is to certify that the project titled RESTAURANT MANAGEMENT SYSTEM is a record of the bonafide work done by ANANT AGARWAL (Reg. No. 220962264) and MEDHA CHAWLA (Reg No. 220962340) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech.) in COMPUTER SCIENCE & ENGINEERING of Manipal Institute of Technology, Manipal, Karnataka, (A Constituent Institute of Manipal Academy of Higher Education), during the academic year 2023-2024.

Name and Signature of Examiners:

1. Dr. Anup Bhat B, Assistant Professor, Dept. of CSE
2. Dr. Vijaya Arjunan, Additional Professor, Dept. of CSE
3. Dr. T Sujithra, Associate Professor, Dept. of CSE (A2 Batch)

3. Dr. Shwetha Rai, Assistant Professor-Senior Scale, Dept. of CSE (A1 Batch)

TABLE OF CONTENTS

ABSTRACT

**CHAPTER 1: INTRODUCTION**

**CHAPTER 2: PROBLEM STATEMENT & OBJECTIVES**

**CHAPTER 3: METHODOLOGY**

**CHAPTER 4: RESULTS & SNAPSHOTS**

**CHAPTER 5: CONCLUSION**

**CHATER 6: LIMITATIONS & FUTURE WORK**

**CHAPTER 7: REFERENCES**

CHAPTER 1 : INTRODUCTION

Welcome to our Restaurant Management System project! Imagine running a restaurant without any chaos or confusion. That's exactly what we aim to achieve with our project. We've created a system that helps restaurants manage everything smoothly, from keeping track of customers to handling orders and payments, all with the help of a powerful tool called SQL.

SQL, or Structured Query Language, is like the language our system speaks to interact with its database. It helps us do a bunch of cool stuff, like adding new items to the menu, tracking how much food we have left, and making sure every order is recorded accurately.

This report presents an overview of our RMS project, highlighting its key features, SQL functionalities utilized, and the rationale behind the adoption of a database management system. With a focus on enhancing customer satisfaction, streamlining employee management, optimizing menu offerings, and ensuring seamless order processing, our RMS offers a holistic solution to the challenges faced by restaurants in today's competitive landscape.

CHAPTER 2 : PROBLEM STATEMENT AND OBJECTIVES

Problem Statement:

The restaurant industry is known for its fast-paced environment and intricate operations. However, with manual management systems or outdated software solutions, restaurants often face significant challenges in maintaining efficiency, accuracy, and customer satisfaction. Common problems include:

* Disorganized Customer Management
* Inefficient Employee Management
* Outdated Menu Management
* Tedious Order Processing
* Lack of Inventory Control
* Ineffective Payment Handling

Overall, the lack of a comprehensive and integrated management system hampers the ability of restaurants to operate efficiently, provide exceptional customer service, and stay competitive in the market.

Therefore, the primary objective of our project is to address these challenges by developing a Restaurant Management System (RMS) that integrates a robust SQL database with a user-friendly graphical user interface (GUI).

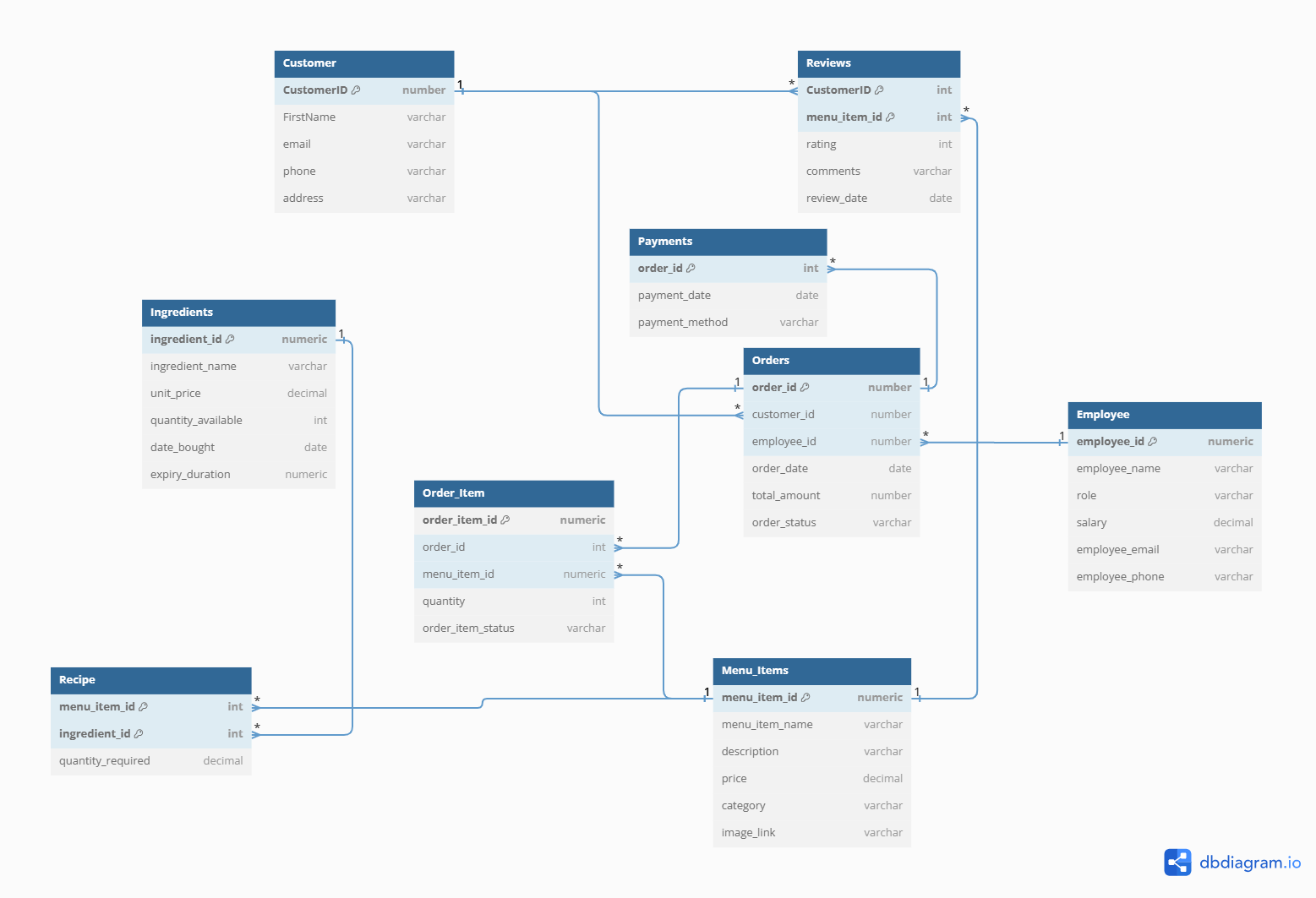
Objectives :

* Customer Satisfaction: Implement features for personalized service, order tracking, and efficient customer management to enhance the dining experience and foster long-term customer relationships.
* Employee Management: Create tools for managing employee information, roles, schedules, and payroll to optimize human resource management and ensure employee satisfaction and productivity.
* Menu Flexibility: Provide a platform for easily updating and customizing menus, including adding, modifying, and removing items, descriptions, and prices, to adapt to changing customer preferences and industry trends.
* Inventory Optimization: Develop systems for monitoring ingredient quantities, tracking usage, and generating alerts to prevent stockouts, minimize wastage, and maintain cost-effective inventory levels.
* Financial Accuracy: Implement robust payment handling mechanisms to ensure accurate and efficient processing of customer payments, supporting various payment methods and reducing errors and discrepancies.
* Technology Integration: Integrate SQL functionalities seamlessly with a user-friendly graphical user interface (GUI) to provide an intuitive platform accessible to restaurant staff with varying technical expertise.
* Data Integrity: Apply SQL constraints and transactional mechanisms to maintain data integrity, enforce relationships between database tables, and ensure the accuracy and consistency of restaurant operations.

3.METHODOLOGY

* Requirement Analysis: Conducted an in-depth analysis of restaurant management needs to identify SQL functionalities required for efficient data management, retrieval, and manipulation.
* Database Design: Employed Entity-Relationship Modeling to design a relational database schema, ensuring proper normalization and adherence to SQL best practices for efficient storage and retrieval of data.
* CRUD Operations: Utilized SQL for Create, Read, Update, and Delete operations to manage data in the database tables, ensuring seamless interaction with customer details, employee information, menu items, orders, inventory, and payments.
* Joins: Employed SQL JOIN operations to retrieve data from multiple tables, facilitating the integration of information for display in the graphical user interface (GUI) and supporting complex queries for order processing, inventory management, and reporting.
* Constraints: Applied SQL constraints such as Primary Key, Foreign Key, Unique, and Check constraints to maintain data integrity, enforce relationships between tables, and prevent anomalies in the database.
* Stored Procedures and Triggers: Leveraged SQL stored procedures for encapsulating frequently used operations and automating complex tasks, enhancing system efficiency and reducing redundancy. Additionally, employed triggers for automating actions in response to database events, such as sending notifications on reservation confirmations or updating inventory levels after order placement.
* Error Handling: Implemented robust error handling mechanisms within SQL procedures and transactions to gracefully handle exceptions and maintain system stability and data consistency.
* Security Measures: Implemented SQL security features such as user authentication, authorization, and access control to protect sensitive data and ensure compliance with data privacy regulations. Additionally, employed encryption techniques to safeguard data transmission and storage within the database.

ER-DIAGRAM



NORMALIZATION

BCNF CONVERSION

1] Customer

CustomerID -> FirstName , email , phone , address

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

2] Employee

Employee\_id -> Employee\_name , roll , salary , Employee\_email , Employee\_phone

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

3] Ingredients

Ingredients\_id -> ingredient\_name , unit\_price , quantity\_available , date\_bought , duration\_of\_expiry

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

4] Reviews

Customer\_id , Menu\_item\_id -> ratings , comment , review\_date

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

5] Order

Order\_id -> Customer\_id , Employee\_id , order\_date , total\_amount , order\_status

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

6] Order\_item

Order\_item\_id -> order\_id , menu\_item\_id , quantity , order\_item\_status

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

7] Menu\_Item

Menu\_item\_id -> menu\_item\_name , description , price , category , image\_link

8] Payment

Order\_id -> payment\_date , payment\_method

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

9] Recipe

Menu\_item\_id , ingredient-id -> quantity\_required

This is the only non-trivial functional dependency that exist on this table and LHS is primary key of the table therefore table is in BCNF.

In all the tables mentioned above alpha -> beta is the only functional dependency where alpha is a superkey .Therefore all the tables are already in BCNF and there is no need of conversion .

3NF CONVERSION

1] CUSTOMER TABLE

customer\_id -> {first\_name, email, phone, address}

Since customer\_id is a candidate key , there are no partial dependencies.

To ensure that there are no transitive dependencies, we need to examine the attributes email, phone, and address:

There are no transitive dependencies since each non-prime attribute (email, phone, address) is fully functionally dependent on the candidate key (customer\_id).

Therefore, the customer table is already in 3NF.

However, if we want to decompose the table further for better organization, we could separate the customer table into two tables:

customer\_info (customer\_id, first\_name)

customer\_contact (customer\_id, email, phone, address)

2]EMPLOYEE TABLE

EMPLOYEE\_ID -> {EMPLOYEE\_NAME, ROLE, SALARY, EMPLOYEE\_EMAIL, EMPLOYEE\_PHONE}

There are no transitive dependencies in the given set of attributes, so the table is already in 2NF. To achieve 3NF, we need to ensure that there are no non-prime attributes dependent on other non-prime attributes.

Since all attributes are dependent on the candidate key (EMPLOYEE\_ID), we don't need to decompose the table further. Therefore, the table is already in 3NF.

3] ORDER TABLE

order\_id -> {customer\_id, employee\_id, order\_date, total\_amount, order\_status}

Since all other attributes are dependent on the primary key (order\_id), the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

Since there are no transitive dependencies in the given set of attributes, the table is already in 3NF.

4] ORDER\_ITEM TABLE

order\_id, menu\_item\_id -> {quantity, order\_item\_status}

Since all other attributes are dependent on the composite key {order\_id, menu\_item\_id}, the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

Since there are no transitive dependencies in the given set of attributes, the table is already in 3NF.

5] MENU\_ITEM TABLE

menu\_item\_id -> {menu\_item\_name, description, price, category}

Since all attributes are functionally dependent on the menu\_item\_id (the candidate key), the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

There are no transitive dependencies in the given set of attributes, so the table is already in 3NF.

6] INGREDIENTS

Ingredients\_id -> ingredient\_name , unit\_price , quantity\_available , date\_bought , duration\_of\_expiry

Since all attributes are functionally dependent on the menu\_item\_id (the candidate key), the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

There are no transitive dependencies in the given set of attributes, so the table is already in 3NF.

7]RECIPE

Menu\_item\_id , ingredient-id -> quantity\_required

Since all attributes are functionally dependent on the menu\_item\_id (the candidate key), the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

There are no transitive dependencies in the given set of attributes, so the table is already in 3NF.

8] PAYMENTS

Order\_id -> payment\_date , payment\_method

Since all attributes are functionally dependent on the menu\_item\_id (the candidate key), the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

There are no transitive dependencies in the given set of attributes, so the table is already in 3NF.

9] REVIEWS

Customer\_id , Menu\_item\_id -> ratings , comment , review\_date

Since all attributes are functionally dependent on the menu\_item\_id (the candidate key), the table is already in 2NF. To achieve 3NF, we need to ensure that there are no transitive dependencies.

There are no transitive dependencies in the given set of attributes, so the table is already in 3NF.

CHAPTER 4 : RESULT AND SNAPSHOTS

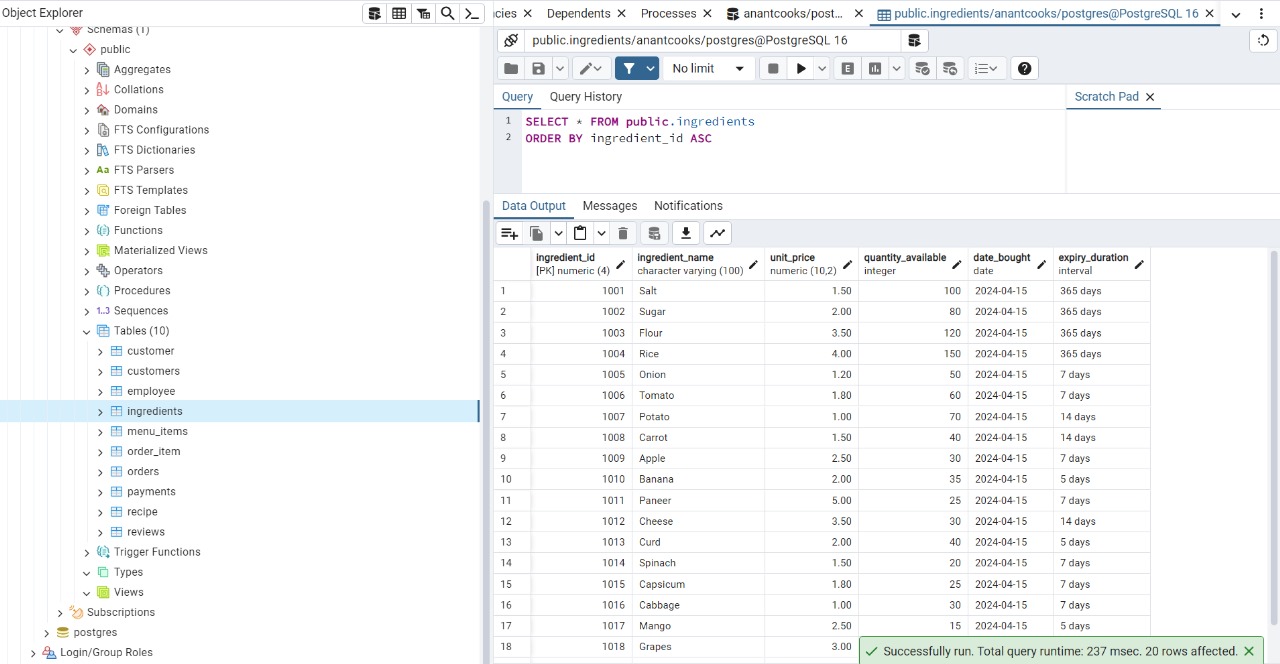
RESULT

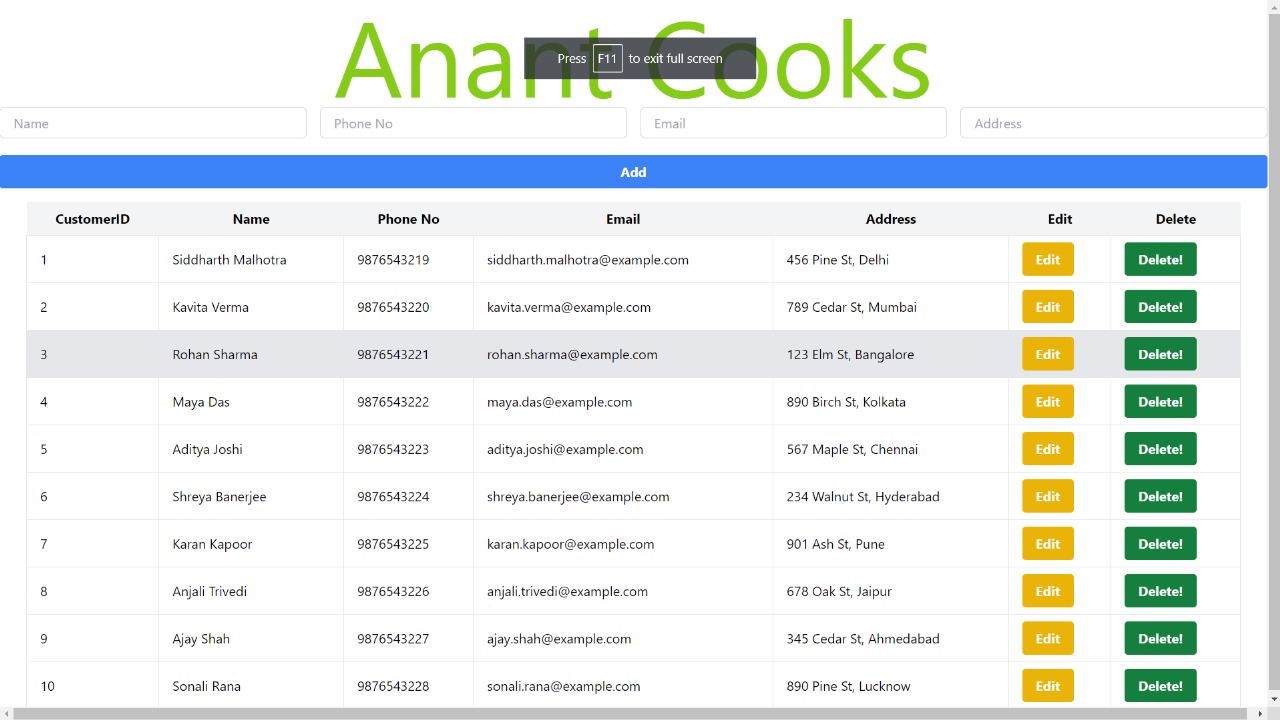
The Restaurant Management System (RMS) project has achieved significant success in addressing the challenges faced by restaurants and enhancing their operational efficiency. The key results and outcomes of the project include:

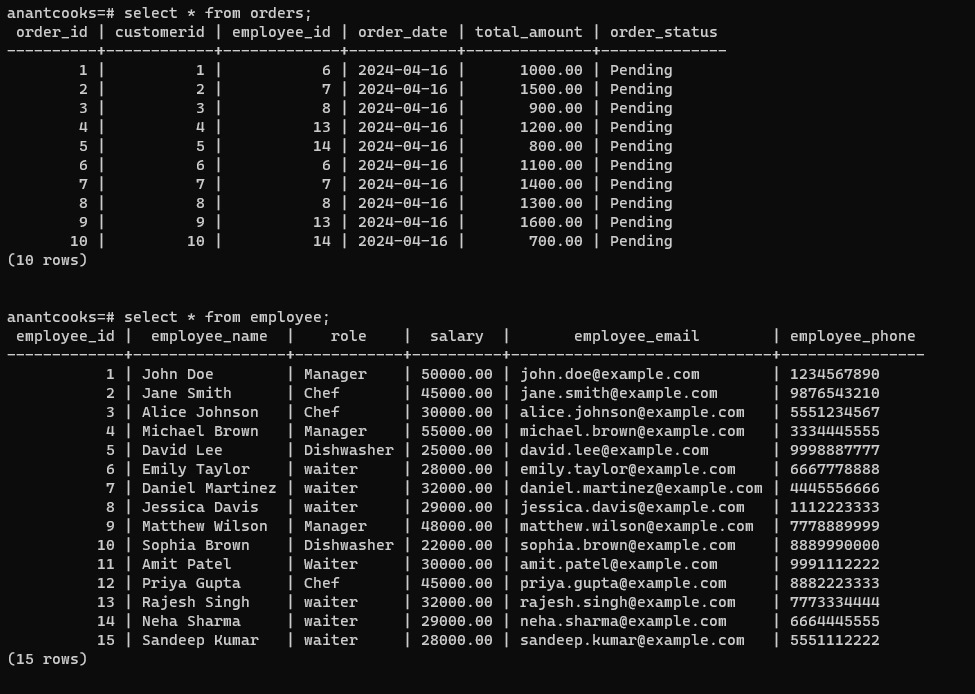
* Streamlined Operations: The implementation of the RMS has resulted in streamlined restaurant operations, allowing for efficient management of customer details, employee information, menu items, orders, inventory, and payments.
* Enhanced Customer Service: The RMS has empowered restaurants to provide personalized service and meet customer expectations more effectively. This has resulted in higher levels of customer satisfaction and increased repeat business.
* Improved Employee Management: The RMS has facilitated better management of employee information, roles and payroll, leading to improved human resource management practices. This has resulted in greater employee satisfaction and productivity.
* Dynamic Menu Management: Restaurants can now easily update and customize their menus, including adding, modifying, and removing items, descriptions, and prices. This flexibility has allowed restaurants to adapt to changing customer preferences and market trends more effectively.
* Efficient Payment Handling: With robust payment handling mechanisms, the RMS ensures accurate and efficient processing of customer payments, displaying various payment options.
* User-Friendly Interface: The graphical user interface (GUI) of the RMS is intuitive and easy to use, making it accessible to restaurant staff with varying technical expertise. This has facilitated smooth adoption and usage of the system across different restaurant environments.

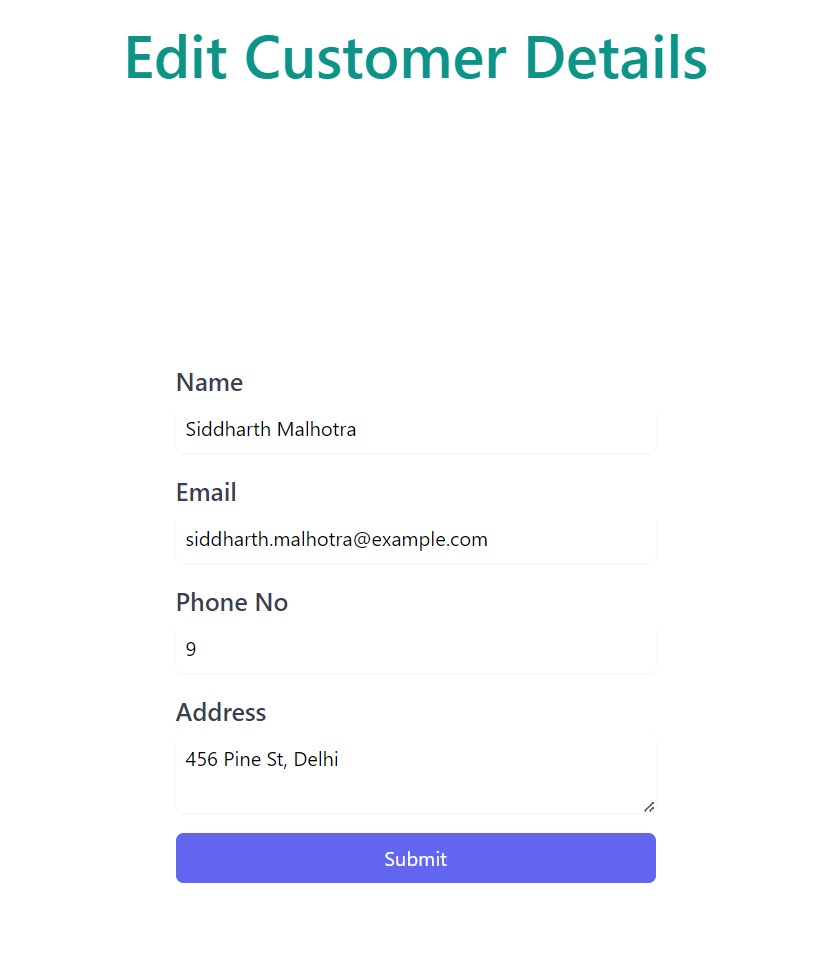
Overall, the successful implementation of the Restaurant Management System has contributed to the operational excellence, customer satisfaction, and profitability of restaurants, positioning them for continued success in the competitive food industry landscape.

SNAPSHOTS









**5**. **CONCLUSION**

In conclusion, the development and implementation of the Restaurant Management System (RMS) have marked a significant milestone in revolutionizing restaurant operations and enhancing the dining experience for both customers and restaurant staff. Through the integration of a robust SQL database with a user-friendly graphical user interface (GUI), the RMS has addressed the critical challenges faced by restaurants and provided a comprehensive solution to streamline various aspects of their operations.

The RMS has demonstrated its effectiveness in optimizing customer and employee management, menu customization, order processing, inventory control, and payment handling. By leveraging SQL functionalities such as CRUD operations, joins, constraints, transactions, stored procedures, and triggers, the system has ensured efficient data management, integrity, and reliability.

Furthermore, the user-friendly interface of the RMS has facilitated its adoption and usage by restaurant staff, regardless of their technical expertise, leading to improved productivity and operational efficiency. The system's dynamic features, including real-time inventory tracking, personalized customer service, and seamless payment processing, have resulted in enhanced customer satisfaction and loyalty.

**6. LIMITATIONS AND FUTUTRE WORK**

**LIMITATIONS**

**Normalization Overhead:** While normalization ensures data integrity and reduces redundancy, it can result in increased complexity and overhead in query execution, especially for complex joins involving multiple tables. This may impact system performance, particularly during peak hours.

**Limited Scalability:** The current design of the RMS may have limitations in scalability, particularly in accommodating a growing number of users, transactions, and data volumes. As the restaurant expands or experiences increased demand, scalability concerns may arise, requiring architectural adjustments.

**Potential for Update Anomalies:** The design of the RMS, particularly in the presence of complex relationships between entities, may be susceptible to update anomalies. Changes to data in one table may require corresponding updates in related tables, increasing the risk of inconsistencies if not handled carefully.

**Performance Impact of Triggers and Stored Procedures**: While triggers and stored procedures enhance automation and customization, they may also introduce performance overhead, particularly during high-concurrency scenarios. Complex logic within triggers or procedures can affect transaction processing times and overall system responsiveness.

Addressing these limitations may involve ongoing optimization, refinement, and adaptation of the RMS design to meet evolving business requirements and technological advancements. It is essential for stakeholders to continuously evaluate and enhance the design to ensure its effectiveness and relevance in supporting restaurant operations effectively.

FUTURE WORK

**Enhanced Reporting and Analytics**: Implement advanced reporting and analytics features within the RMS to provide insightful business intelligence for restaurant management. This could include dashboards, data visualization tools, and predictive analytics capabilities to optimize decision-making processes.

**Integration with External Systems**: Explore opportunities to integrate the RMS with external systems such as online ordering platforms, accounting software, and customer relationship management (CRM) systems. Seamless integration would improve data exchange and streamline overall restaurant operations.

**Mobile Application Development**: Develop a mobile application companion to the RMS, allowing restaurant staff to access key functionalities such as order management, inventory tracking, and customer management on-the-go. This would enhance operational flexibility and accessibility.

By incorporating these future work considerations into the project roadmap, the RMS can evolve into a comprehensive and cutting-edge solution that meets the evolving needs of the restaurant industry and delivers sustained value to stakeholders.

**7. REFERENCES**

[draw.io (diagrams.net)](https://app.diagrams.net/)

[Design Restaurant Management System | System Design - GeeksforGeeks](https://www.geeksforgeeks.org/design-restaurant-management-system-system-design/)

[Database Design of a Restaurant Management System from user story to relational Model - DEV Community](https://dev.to/lontchi12/database-design-of-a-restaurant-management-system-from-user-story-to-relational-model-2jp1)