Database Management System UE21CS351A

PROJECT

REPORT

PES CANTEEN PRE-BOOKING SYSTEM

SEMESTER: **5**

SECTION: **A**

TEAM NUMBER: **23**

TEAM: ***SRN:***

1. *AARYAN SHETTY*   *PES1UG21CS010*

2. *ADITHYA MAHESH*   *PES1UG21CS032*

1. *Mini World/Application Description:*

The Canteen Prebooking Management System is a web application designed to streamline the process of ordering food from various canteens within a university campus. It caters to both customers and canteen owners, providing a user-friendly interface for placing orders, managing menus, and handling transactions. Users can log in, view menus from different canteens, place orders, and confirm them. Canteen owners have additional functionalities, including viewing current orders, adding menu items, and tracking total sales. By implementing a canteen pre-booking system, the project aims to:

Reduce Overcrowding: By allowing users to pre-order food, the system aims to reduce long queues and overcrowding in the canteens during peak hours.

Save Time: Users can save valuable time by pre-booking their meals, minimizing waiting times and allowing for quick and convenient meal pickups.

Enhance Ordering Convenience: The system will make it easier for users to browse menus, place orders, and make payments through a user-friendly interface, accessible via web.

Improve Canteen Management: Canteen staff will benefit from streamlined order processing and better inventory management, leading to increased efficiency and reduced wastage.

2. *User Requirement Specification :*

1. Customer Requirements:
    i.   User Authentication:
        * Users (both customers and canteen owners) can log in using their credentials (email and password).
        * New users can sign up by providing necessary details.
    ii. Customer Functions:
        * Customers can view menus from different canteens.
        * Customers can place food orders by selecting items and specifying quantities.
        * Confirmation of placed orders is available to customers.
2. Canteen Owner Requirements:
    i.   User Authentication:
        * Users (both customers and canteen owners) can log in using their credentials (email and password).
        * New users can sign up by providing necessary details.
    ii. Canteen Owner Functions:
        * Canteen owners can view the current orders for their canteens.
        * Canteen owners can add new menu items with details such as item name, description, and price.
        * Total sales for a canteen can be viewed by canteen owners.
3. Order Management:
    i.   Placed Orders:
   * Placed orders are stored in a database with relevant details.
    ii. User Confirmation:
   * Users (customers) can confirm placed orders.

3. *List of Functionalities of the Application:*

1. Login and Signup:
    - Users can log in with their credentials.
    - New users can sign up by providing necessary information.
2. Customer Functions:
    - Customers can view menus from different canteens.
    - Customers can place food orders by selecting items and specifying quantities.
    - Confirmation of placed orders is available to customers.
3. Canteen Owner Functions:
    - Canteen owners can view the current orders for their canteens.
    - Canteen owners can add new menu items with details such as item name, description, and price.
    - Total sales for a canteen can be viewed by canteen owners.
4. Order Management:
    - Placed orders are stored in a database with relevant details.
    - Users (customers) can confirm placed orders.
5. User Management:
    - Users can be deleted from the system.
6. Database Interaction:
    - The system interacts with a MySQL database to store and retrieve user data, menus, orders, and other relevant information.

4. ER DIAGRAM
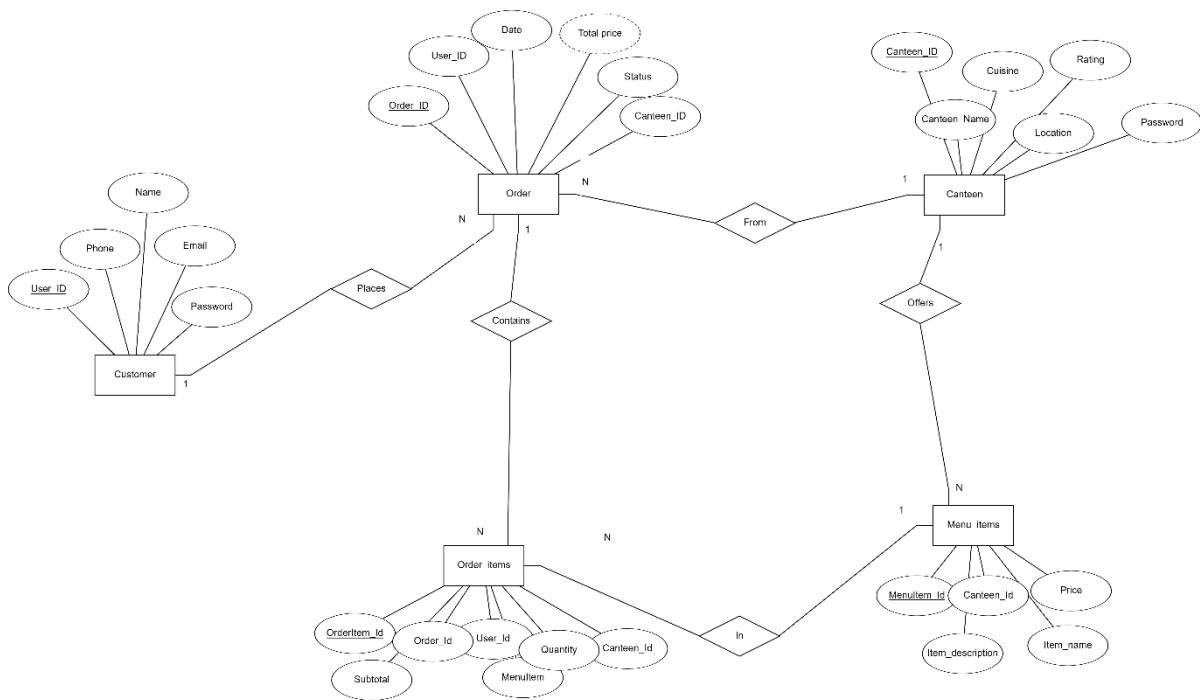
5. RELATIONAL SCHEMA:
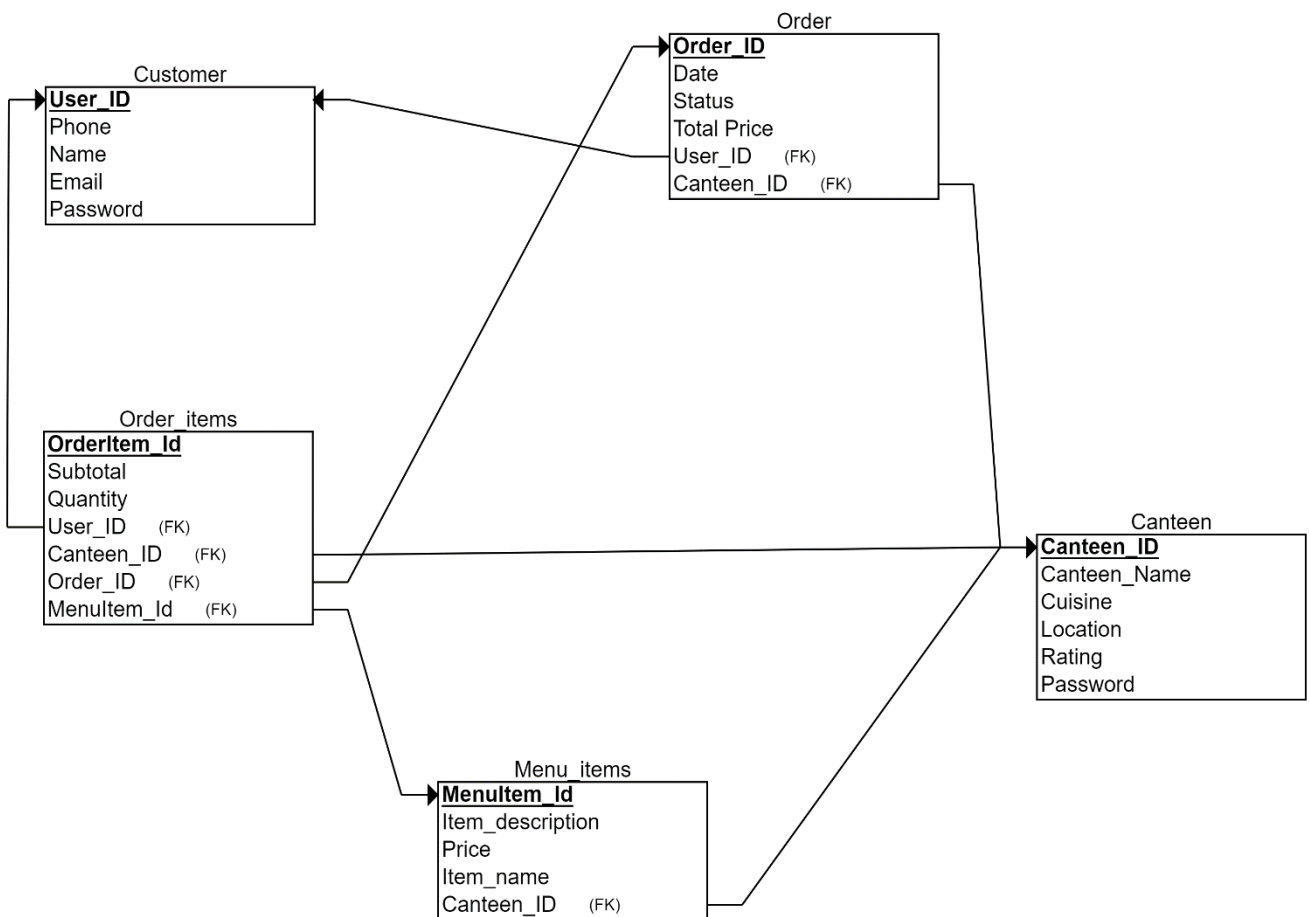
6. *SCREENSHOTS ACCORDING TO THE RUBRICS:*

i. NESTED QUERY:

```python
# Get all order_items for the canteen
get_canteen_orders_query = """
SELECT OI.orderitem_id, OI.menuitem_id, MI.item_name, MI.price, OI.quantity, OI.subtotal, OI.user_id
FROM order_items OI
INNER JOIN menu_items MI ON OI.menuitem_id = MI.menuitem_id
WHERE OI.canteen_id = %s AND OI.order_id IN (
    SELECT order_id
    FROM `order`
    WHERE status = 'Confirmed'
)
"""
cursor.execute(get_canteen_orders_query, (canteen_id,))
canteen_orders = cursor.fetchall()
```

ii. JOIN QUERY;

```python
get_user_orders_query = """
SELECT OI.orderitem_id, OI.menuitem_id, MI.item_name, MI.price, OI.quantity, OI.subtotal, OI.canteen_id
FROM order_items OI
INNER JOIN menu_items MI ON OI.menuitem_id = MI.menuitem_id
WHERE OI.user_id = %s
"""
cursor.execute(get_user_orders_query, (user_id,))
user_orders = cursor.fetchall()
```

iii. AGGREGATE QUERY:

```python
get_total_sales_query = """
SELECT O.canteen_id, C.canteen_name, SUM(O.total_price) AS total_sales
FROM `order` O
INNER JOIN canteen C ON O.canteen_id = C.canteen_id
WHERE O.canteen_id = %s
GROUP BY O.canteen_id, C.canteen_name
"""
cursor.execute(get_total_sales_query, (canteen_id,))
total_sales = cursor.fetchall()
```

iv. TRIGGERS:

```
DELIMITER //
CREATE TRIGGER before_order_insert
BEFORE INSERT ON `order`
FOR EACH ROW
BEGIN
    SET @order_count = (SELECT COUNT(*) FROM `order`) + 1;
    SET NEW.order_id = CONCAT('O_', @order_count);
END;
//
DELIMITER ;
```

v. PROCEDURES:

```
DELIMITER //
CREATE PROCEDURE ConfirmOrder(IN userId VARCHAR(10))
BEGIN
    UPDATE `order` SET
        total_price = (
            SELECT SUM(subtotal)
            FROM order_items
            WHERE order_id = `order`.order_id
        ),
        status = 'Confirmed'
    WHERE user_id = userId;
END //
DELIMITER ;
```

vi. FUNCTIONS:

```
CREATE FUNCTION CalculateTotalPrice(order_id VARCHAR(10)) RETURNS DECIMAL(5, 2) READS SQL DATA
BEGIN
    DECLARE total DECIMAL(5, 2);

    SELECT SUM(subtotal) INTO total
    FROM order_items
    WHERE order_id = order_id;

    RETURN COALESCE(total, 0);
END //

DELIMITER ;
```

7. _SCREENSHOTS:_

Menu

Choose an option
- Login
- Signup
- Place Order
- Confirm Order
- Delete User

PESU CANTEEN PRE BOOKING MANAGEMENT

Confirm Order

Your Orders:

Buns - Quantity: 2, Subtotal: ₹20.00

Paneer Butter Masala - Quantity: 1, Subtotal: ₹150.00

Butter Naan - Quantity: 1, Subtotal: ₹30.00

Roti - Quantity: 1, Subtotal: ₹15.00

Masala Dosa - Quantity: 2, Subtotal: ₹160.00

Confirm Order

Menu

Choose an option
- Login
- Signup
- View Current Orders
- Add Menu Item
- View Total Sales
- Delete User

PESU CANTEEN PRE BOOKING MANAGEMENT

View Current Orders

Current Orders:

Paneer Butter Masala - Quantity: 1, Subtotal: ₹150.00, User: U_1

Butter Naan - Quantity: 1, Subtotal: ₹30.00, User: U_1

Roti - Quantity: 1, Subtotal: ₹15.00, User: U_1

Menu

Choose an option
- Login
- Signup
- View Current Orders
- Add Menu Item
- View Total Sales
- Delete User

PESU CANTEEN PREBOOKING MANAGEMENT

Add Menu Item

Item Name

Roti

Item Description

Roti is an Indian flatbread made from whole wheat flour, commonly used to complement various curries and dishes.

Item Price

30.00   − +

Add Menu Item

Menu item added successfully!

View Total Sales

Total Sales for Your Canteen:

Canteen ID: C_5, Canteen Name: F-block mess, Total Sales: ₹195.00

Delete User

Delete User

User deleted successfully!