Project Title: Khata (RMS)

Group Members:

Muhammad Ammar Saleem   2023378

Raja Bilal Khurram   2023591

Course : CS-232 DBMS

Instructor: Sir Ahsan Shah

Table of Contents

Abstract

Inspired by the lack of any proper system in GIK we have created Khata: a dedicated Restaurant Management System designed to streamline the operations of small to medium-sized restaurants by integrating backend SQL database management with a user-friendly Python-based front-end interface. The system manages core restaurant activities such as menu management, order processing, billing, and customer tracking. Leveraging a normalized relational database schema, Khata ensures data integrity and supports efficient querying for daily restaurant operations. The Python front-end implements multiple interactive GUI screens to allow restaurant staff to seamlessly input orders, update menu items, generate bills, and review transaction history without requiring technical knowledge of SQL. This project emphasizes usability, real-time data consistency, and extensibility, making Khata a practical tool for modern restaurant management needs.

Introduction

1. Brief Introduction

Khata is a Restaurant Management System developed to optimize the daily management processes within a restaurant environment. The system is built on a relational SQL database backend that stores key information such as menu items, customer orders, billing data, and staff details. The front-end is created using Python streatmlit library providing an interactive GUI that abstracts complex database operations into simple, user-friendly workflows. The application allows restaurant staff to add or modify menu items, process customer orders, calculate bills automatically, and maintain transaction records, thereby reducing manual errors and improving operational efficiency.

2. Problem Statement

Restaurant operations often involve handling multiple tasks simultaneously, including order taking, billing, and inventory updates. Many restaurants still rely on manual or semi-automated processes, which increase the risk of errors such as incorrect billing, lost orders, or inconsistent inventory tracking. There is a need for a reliable, easy-to-use system that integrates all restaurant operations into a single platform, minimizes human error, and speeds up service delivery. Khata addresses this gap by combining a robust database with an intuitive interface tailored specifically for restaurant management.

3. Problem Overview

Restaurants typically face challenges in maintaining accurate records of menu changes, tracking customer orders efficiently, and generating timely bills. Manual record-keeping is prone to errors, delays, and data loss. Khata provides a cohesive solution by implementing a normalized SQL database schema that ensures referential integrity among tables such as Customers, Orders, Menu, and Billing. The Python GUI facilitates real-time order input and updates, allowing waitstaff to interact with the system smoothly even during peak hours. This eliminates confusion caused by paper-based order slips or disparate software tools, leading to better customer satisfaction and operational control.

---

4. Product Features and Scope

Key Features:

- **Menu Management:** Add, update, or remove menu items including name, price, category, and availability status via the Python front-end.
- **Order Processing:** Staff can enter customer orders through an interactive form; orders are recorded in the database with timestamps.
- **Billing and Payment:** Automatic bill calculation based on ordered items, with the option to print or save bills.
- **Customer Tracking:** Maintain customer information for repeat orders and loyalty tracking.
- **Inventory Linking (Future Prospect):** Basic tracking of stock levels to alert when menu items go out of stock.
- **Report Generation:** Generate sales reports and daily transaction summaries for restaurant management.
- **User-friendly GUI:** Designed for non-technical staff with clear navigation, input validation, and error handling.
- **Database Integrity:** All transactions are handled through SQL queries that ensure ACID properties for reliable data storage.
- **Multiple user roles:** we have also covered role for multiple users like we have admin, waiter, receptionist, and customer, each have different level od access.

3. State of the Art

Khata is a restaurant management system tailored for efficient handling of daily restaurant operations with a focus on ease of use, role-based access, and reliable database management.

Comparison with Existing Systems:

- Unlike large commercial POS systems (e.g., Toast, Square) which are complex and costly, **Khata** is lightweight and designed to be deployed easily in small to medium restaurant environments.
- Compared to generic open-source projects, **Khata** offers a simple Python GUI interface combined with an SQL backend, providing a balance between user-friendliness and backend robustness.

Why Khata Stands Out:

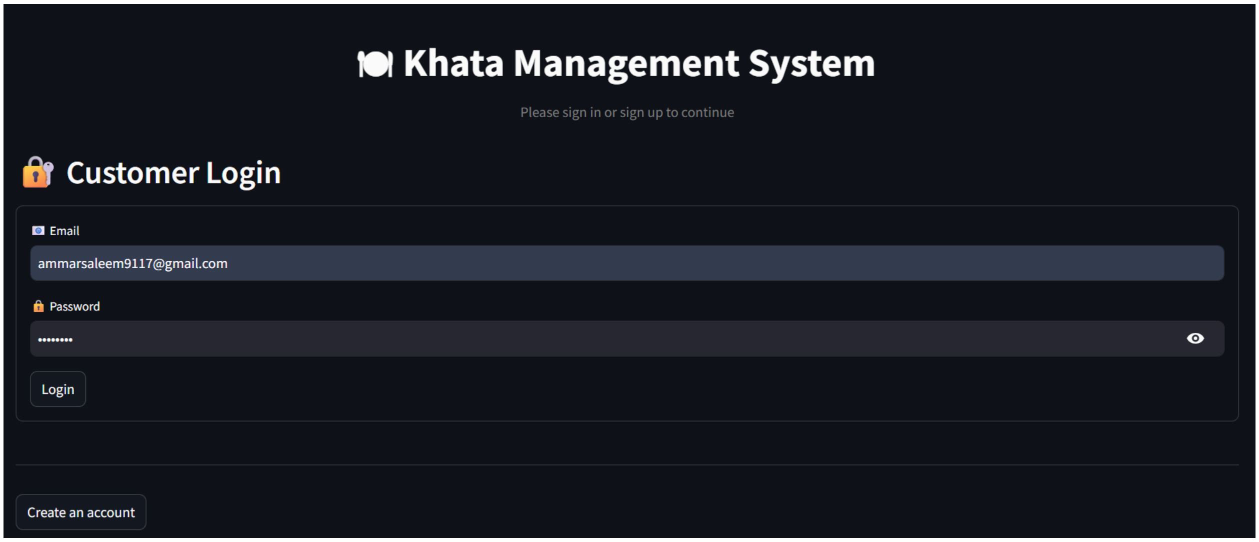
- **Multi-User Role-Based System:** Our code explicitly distinguishes between admin, waiter, receptionist and customer roles with separate access controls and interfaces.
- **Complete CRUD Operations:** Our system fully supports creating, reading, updating, and deleting menu items, orders, and users with transaction-safe database operations.
- **Order Management with Status Updates:** Orders can be created, modified, and tracked through their lifecycle.
- **Real-Time Billing:** The system calculates total bills dynamically based on ordered items and quantities.
- **Authentication and Security:** Password verification and role-based login prevent unauthorized access.
- **Python GUI:** The system uses streamlit interface providing easy navigation and form-based input validated within the code.
- **Database Integrity:** Foreign key relationships and constraints in our SQL schema ensure consistency between orders, customers, and menu items.

This focused, efficient system is designed to automate and streamline the restaurant’s front and back-end operations, with future scope for enhancements like reporting and analytics.

4. Functional Requirements

4.1 User Management

- **Login Authentication:**
  - Users log in with username and password.
  - Passwords are validated against stored hashed values in the users table.
  - The system differentiates users by their roles (admin or waiter), directing them to appropriate interfaces.



Khata Management System

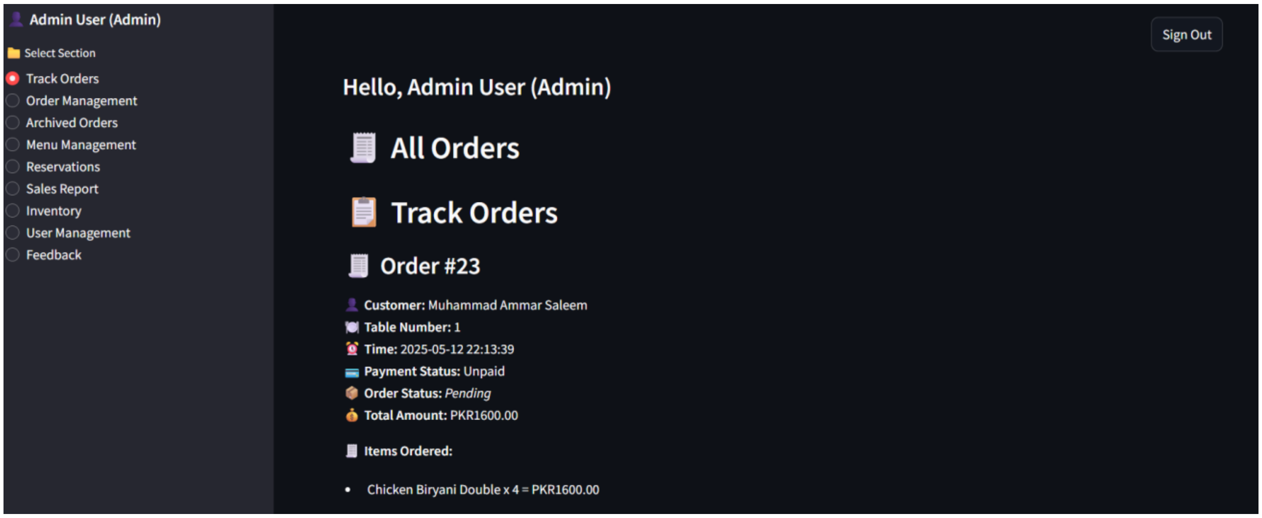
Please sign in or sign up to continue

Customer Login

Email: ammarsaleem9117@gmail.com

Password: ••••••••

- **User Role Control:**
  - **Admin:** Full privileges to manage menu, orders, and users.



Hello, Admin User (Admin)

All Orders

Track Orders

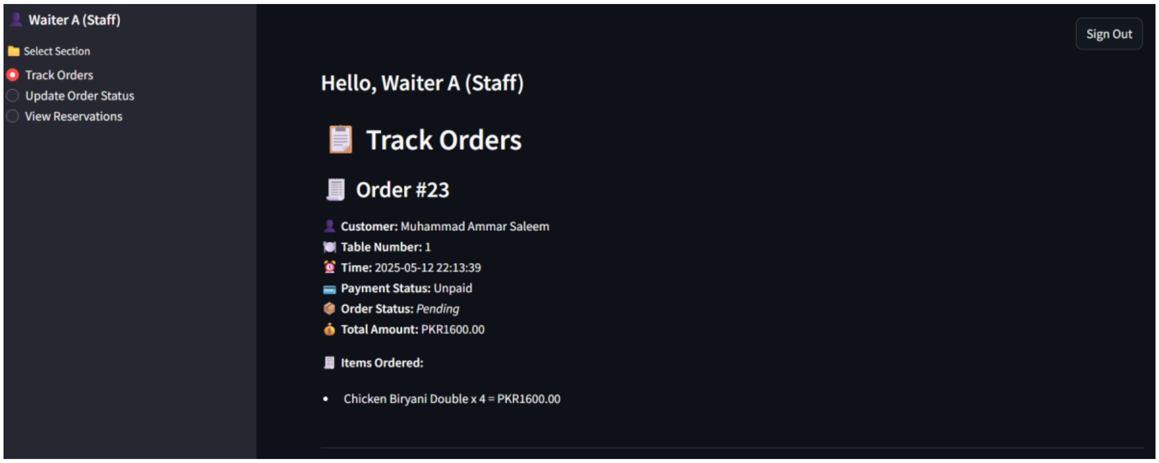
Order #23

Customer: Muhammad Ammar Saleem
Table Number: 1
Time: 2025-05-12 22:13:39
Payment Status: Unpaid
Order Status: *Pending*
Total Amount: PKR1600.00

Items Ordered:

- Chicken Biryani Double x 4 = PKR1600.00

- **Waiter:** Limited to creating orders, updating order statuses, view reservations, and generating bills.



Hello, Waiter A (Staff)

Track Orders

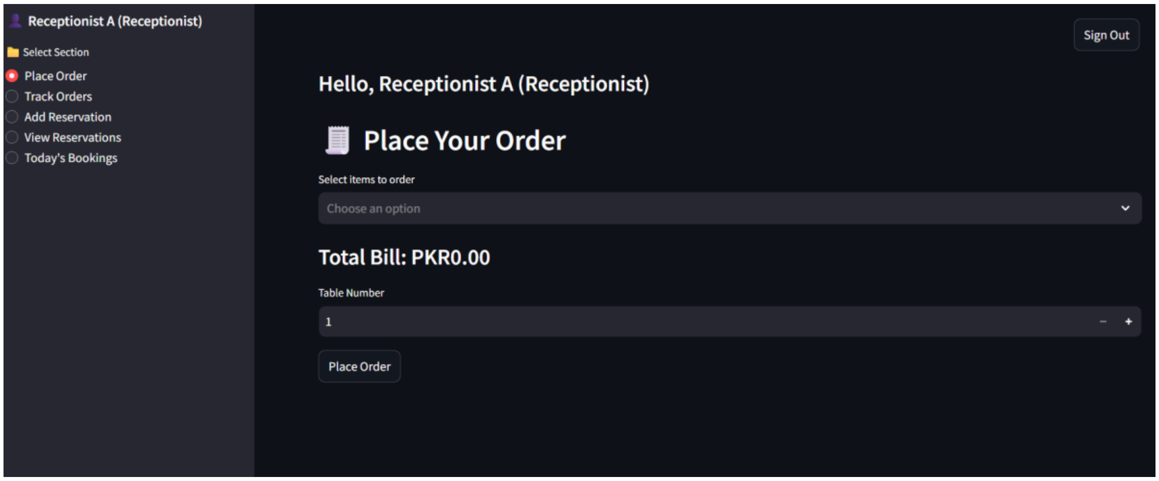
Order #23

Customer: Muhammad Ammar Saleem
Table Number: 1
Time: 2025-05-12 22:13:39
Payment Status: Unpaid
Order Status: *Pending*
Total Amount: PKR1600.00

Items Ordered:

- Chicken Biryani Double x 4 = PKR1600.00

- **Receptionist**: Handles reservations, can place new orders, track existing ones, and view today's bookings.



Hello, Receptionist A (Receptionist)

Place Your Order

Select items to order

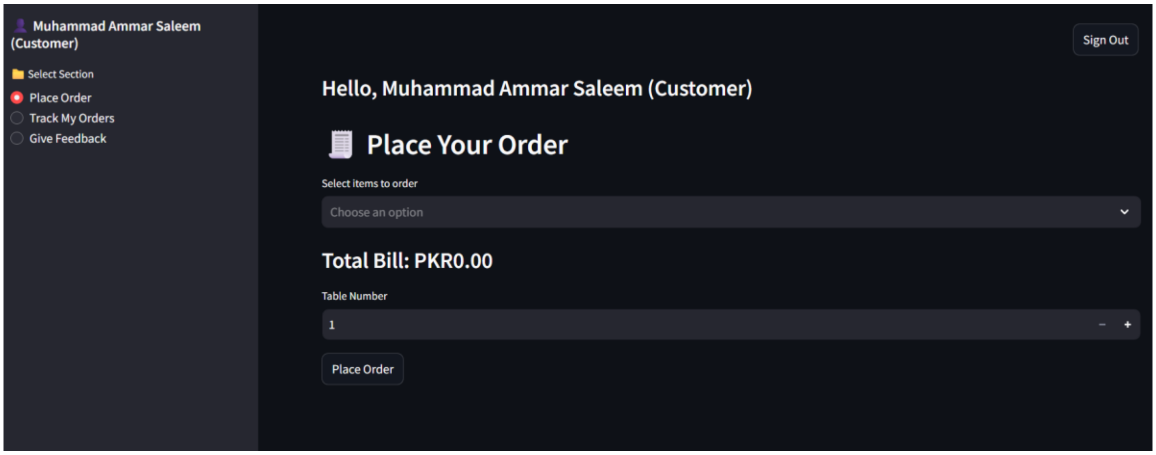
Choose an option

Total Bill: PKR0.00

Table Number

1

- **Customer**: Can place their own orders, track their order status, and submit feedback.



Hello, Muhammad Ammar Saleem (Customer)

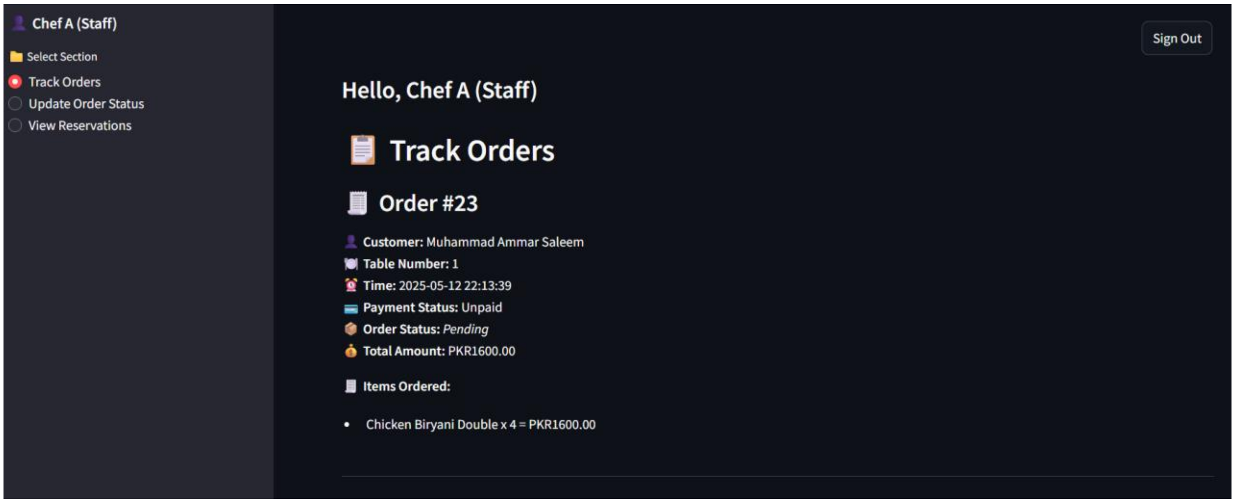
🧾 Place Your Order

Select items to order: Choose an option

Total Bill: PKR0.00

Table Number: 1

- **Chef**: Access to view and update order statuses relevant to kitchen operations only.



Hello, Chef A (Staff)

📋 Track Orders

🧾 Order #23

👤 **Customer:** Muhammad Ammar Saleem
🍽 **Table Number:** 1
⏰ **Time:** 2025-05-12 22:13:39
💳 **Payment Status:** Unpaid
📦 **Order Status:** *Pending*
💰 **Total Amount:** PKR1600.00

🧾 **Items Ordered:**

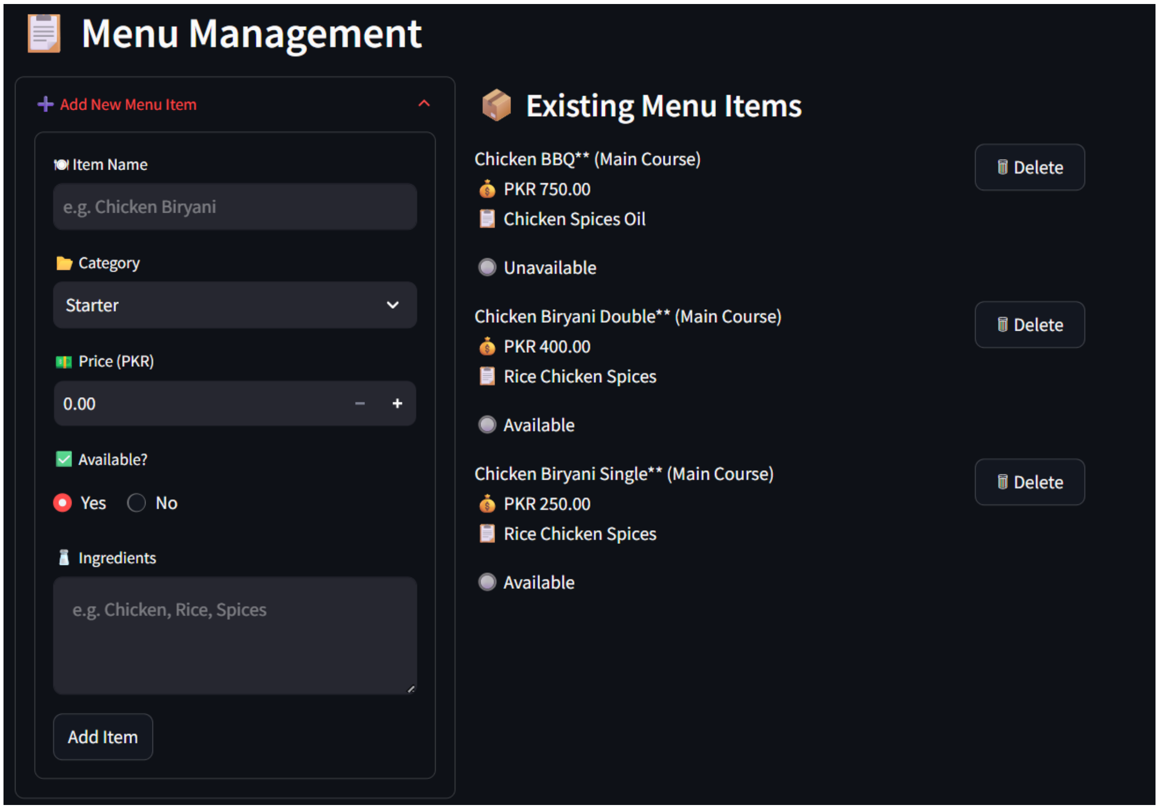
- Chicken Biryani Double x 4 = PKR1600.00

- **User CRUD:**
  - Admins can add new users, edit existing user details (including role), and delete users.

4.2 Menu Management

- **Add Menu Items:** Admin can add new menu items specifying name, category, price, and availability.
- **Edit Menu Items:** Modify existing menu item details, such as price and category.

- **Delete Menu Items:** Remove menu items from the system.
- **Menu Listing:** The interface displays menu items categorized for easy selection during order creation.



Menu Management

+ Add New Menu Item

Item Name

e.g. Chicken Biryani

Category

Starter

Price (PKR)

0.00

Available?

● Yes ○ No

Ingredients

e.g. Chicken, Rice, Spices

Existing Menu Items

Chicken BBQ** (Main Course)
PKR 750.00
Chicken Spices Oil

Unavailable

Chicken Biryani Double** (Main Course)
PKR 400.00
Rice Chicken Spices

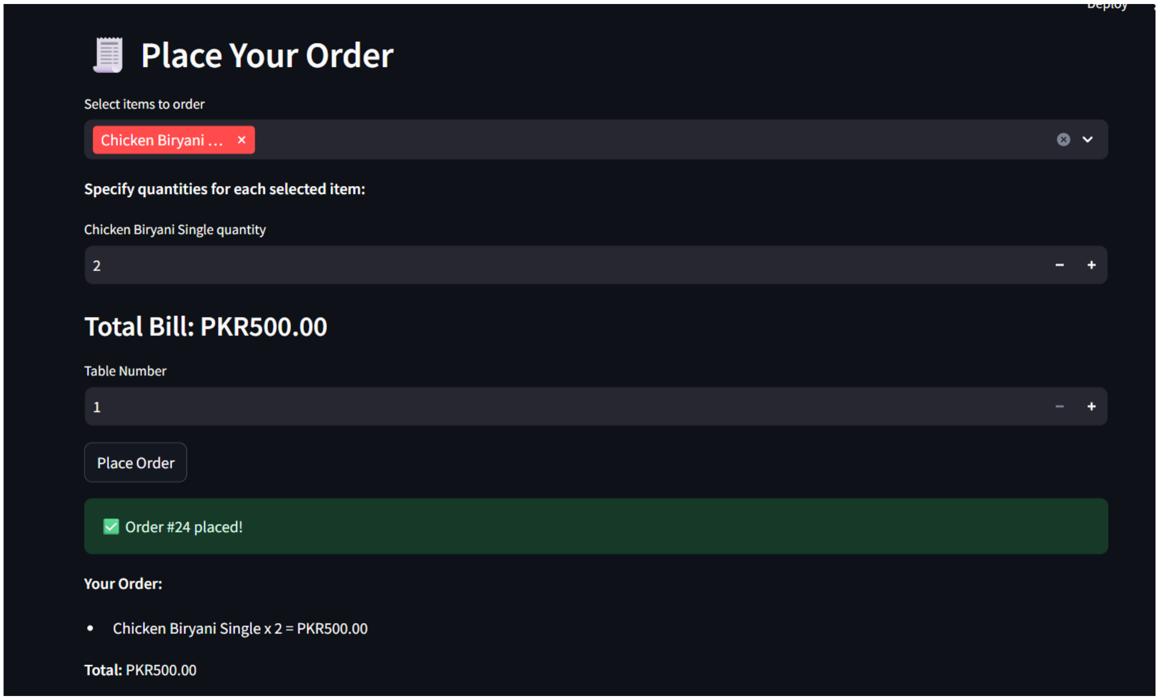
Available

Chicken Biryani Single** (Main Course)
PKR 250.00
Rice Chicken Spices

Available

4.3 Order Management

- **Order Creation:**
    - Waiters select a customer and add multiple menu items to the order.
    - Quantity input for each menu item is supported.
    - Orders are stored in the orders and order_items tables maintaining relationship with customers and menu.



🧾 Place Your Order

Select items to order

Chicken Biryani … ×

Specify quantities for each selected item:

Chicken Biryani Single quantity

2

Total Bill: PKR500.00

Table Number

1

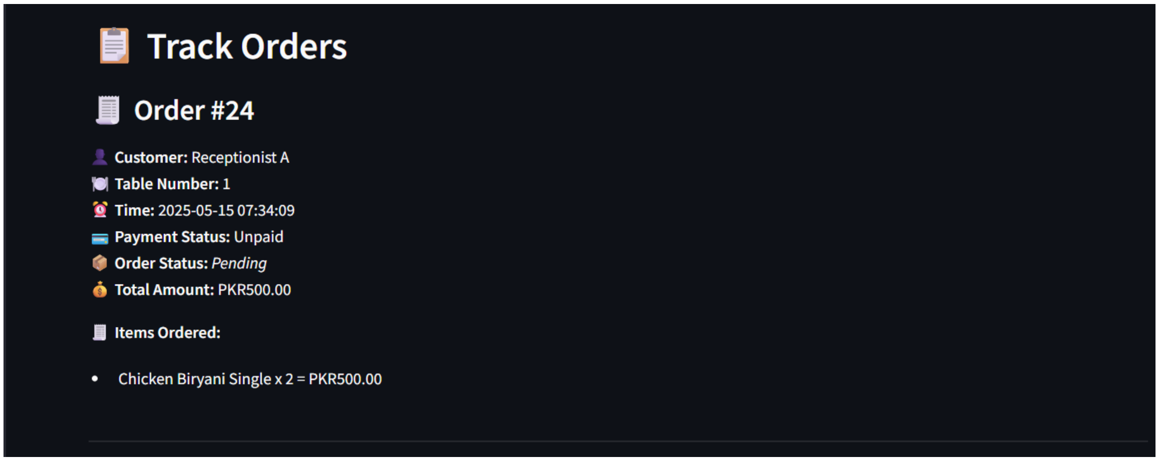
✅ Order #24 placed!

Your Order:

- Chicken Biryani Single x 2 = PKR500.00

Total: PKR500.00

- **Order Status Tracking:**
  - Each order has a status field to indicate stages such as “Pending,” “Served,” or “Paid.”
  - Waiters and admins can update order status.



📋 Track Orders

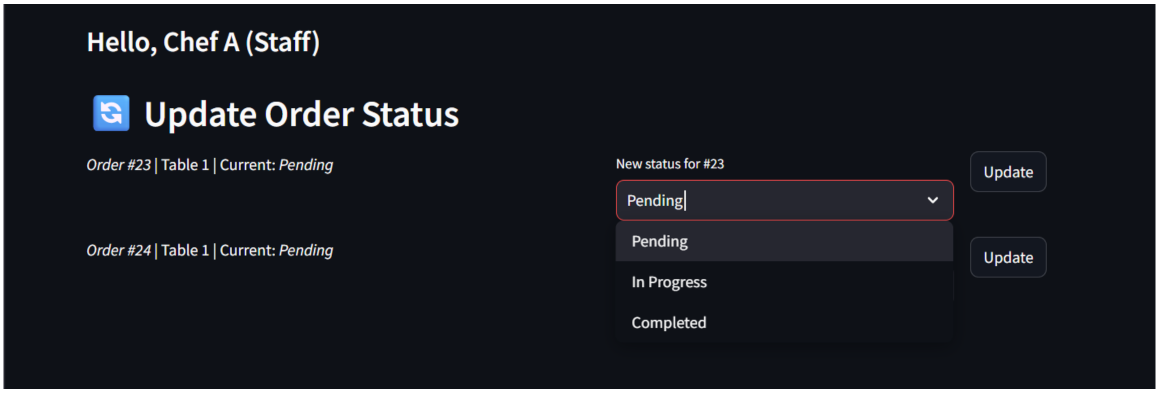
🧾 Order #24

👤 **Customer:** Receptionist A
🍽️ **Table Number:** 1
⏰ **Time:** 2025-05-15 07:34:09
💳 **Payment Status:** Unpaid
📦 **Order Status:** *Pending*
💰 **Total Amount:** PKR500.00

🧾 **Items Ordered:**

- Chicken Biryani Single x 2 = PKR500.00

- **Order Modification:**
  - Before marking an order paid, the order can be edited or cancelled.
- **Order Listing:**
  - Both roles can view a list of active orders with details including items ordered, quantities, total price, and status.



Hello, Chef A (Staff)

🔄 Update Order Status

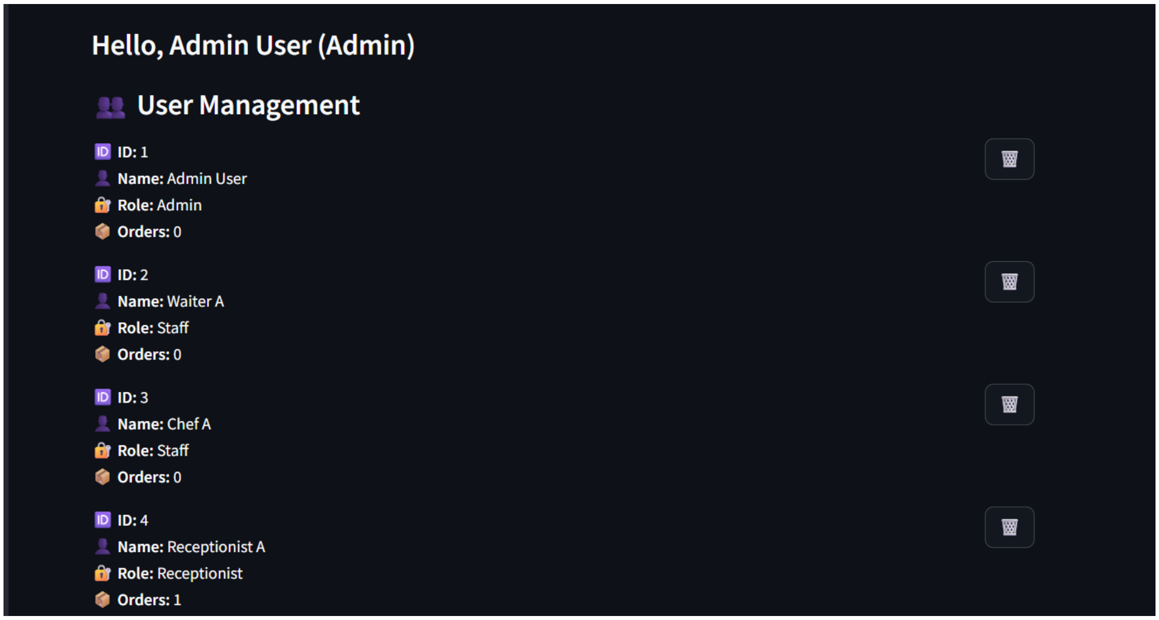
Order #23 | Table 1 | Current: *Pending*

New status for #23: Pending

Order #24 | Table 1 | Current: *Pending*

4.4 Customer Management

- **User CRUD:**
  - Add, edit, and delete customer records.
  - Customer details are linked with orders for history tracking.



Hello, Admin User (Admin)

👥 User Management

🆔 **ID:** 1
👤 **Name:** Admin User
🔐 **Role:** Admin
📦 **Orders:** 0

🆔 **ID:** 2
👤 **Name:** Waiter A
🔐 **Role:** Staff
📦 **Orders:** 0

🆔 **ID:** 3
👤 **Name:** Chef A
🔐 **Role:** Staff
📦 **Orders:** 0

🆔 **ID:** 4
👤 **Name:** Receptionist A
🔐 **Role:** Receptionist
📦 **Orders:** 1

4.5 Billing

- **Automatic Bill Calculation:**
  - Total bill for each order is computed dynamically by summing (price × quantity) for all items.
- **Bill Generation:**
  - The system can generate a printable/displayable bill for completed orders.
- Payment Marking:

4.6 User Interface Features

- **Streamlit GUI:**
  - Multiple windows/forms for login, dashboard, menu management, order taking, and billing.
- **Input Validation:**
  - Checks for empty fields, numeric values where required (e.g., quantities, prices).
- **Navigation:**
  - Role-specific menus with buttons to access allowed features only.
- **Feedback & Error Messages:**
  - User-friendly messages on errors or successful operations.

4.7 Database Features

- **Relational Schema:**
  - The database schema includes tables: users, menu, orders, order_items, and customers linked by foreign keys to enforce integrity.
- **SQLite Backend:**
  - Lightweight, file-based storage ensuring easy deployment.
- **Transactions:**
  - Use of commits after critical operations to maintain database consistency.

4.8 Additional Functionalities Observed in Code

- **Search Functionality:**
  - Ability to search menu items by name or category during order creation.
- **Logout & Session Handling:**
  - Users can log out to secure the session.
- **Basic Reporting:**
  - Display orders filtered by status (pending, completed).

5. ER Diagram

1. Entities (Correspond to Tables)

These are your main building blocks:

Entity Name	Primary Key	Description
Users	user_id	Admins, staff, receptionists, and customers
Menu_Items	item_id	Food or drink items available
Orders	order_id	Orders placed by users
Tables	table_id	Restaurant seating
Reservations	reservation_id	Table bookings
Payments	payment_id	Payments for orders
Inventory	item_id	Ingredient stock
Feedback	feedback_id	Customer feedback
OrderItems	order_item_id	Items in an order
Archived_Orders	order_id	Soft-deleted orders
Archived_Order_Items	item_id	Soft-deleted order items

2. Relationships Between Entities

Relationship	Cardinality	Type	Participation	Description
Users ↔ Orders	1 → N	One-to-Many	Partial	A user can place many orders

Relationship	Cardinality	Type	Participation	Description
Users ↔ Reservations	1 → N	One-to-Many	Partial	A user can reserve many tables
Users ↔ Feedback	1 → N	One-to-Many	Partial	A user can give many feedback entries
Orders ↔ Feedback	1 → 1	One-to-One	Partial	Each feedback belongs to one order
Orders ↔ Payments	1 → 1	One-to-One	Partial	Each order has one payment
Orders ↔ OrderItems	1 → N	One-to-Many	Total on OrderItems	Each order can have multiple items
OrderItems ↔ Menu_Items	N → 1	Many-to-One	Total	Each OrderItem refers to one menu item
Reservations ↔ Tables	N → 1	Many-to-One	Partial	A reservation is for one table
Orders ↔ Tables	N → 1	Many-to-One	Optional	Orders may be associated with a table
Inventory ↔ Supplier (not defined)	Assumed 1 → N	Optional	Not defined	Might be planned for later
Archived_Orders ↔ Archived_Order_Items	1 → N	One-to-Many	Partial	Archived mapping of

Relationship	Cardinality	Type	Participation	Description
				orders and items

3. Attributes for Each Entity

Users

- user_id (PK), name, role, contact, email, password

Menu_Items

- item_id (PK), name, category, price, availability, ingredients

Orders

- order_id (PK), user_id (FK), table_id (FK), status, total_amount, payment_status, order_time

Tables

- table_id (PK), capacity, availability

Reservations

- reservation_id (PK), user_id (FK), table_id (FK), reservation_time, status

Payments

- payment_id (PK), order_id (FK), payment_type, amount, status

Inventory

- item_id (PK), name, quantity, expiry_date, supplier_id (optional)

Feedback

- feedback_id (PK), user_id (FK), order_id (FK), rating, comments

OrderItems

- order_item_id (PK), order_id (FK), item_id (FK), quantity, total_price

Archived Orders

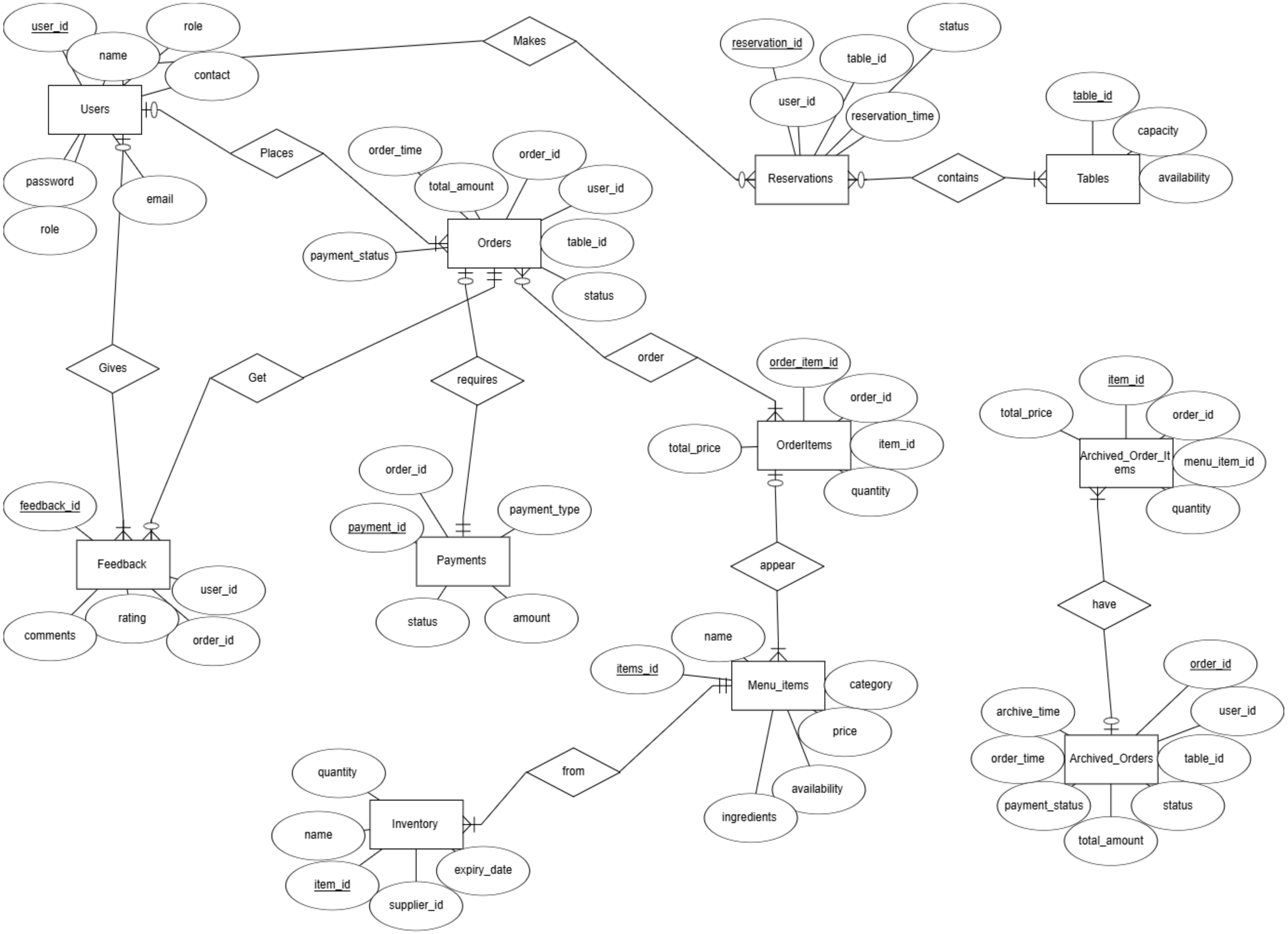
- Same as Orders but soft-deleted; useful in audit trails.

Archived Order Items

- Same as OrderItems but soft-deleted.

4. Cardinality Summary

Relationship	**Cardinality**	**Type**
User → Orders	1:N	One-to-Many
User → Reservations	1:N	One-to-Many
User → Feedback	1:N	One-to-Many
Order → Payment	1:1	One-to-One
Order → OrderItems	1:N	One-to-Many
Order → Feedback	1:1	One-to-One
OrderItem → Menu_Item	N:1	Many-to-One
Reservation → Table	N:1	Many-to-One
Order → Table	N:1 (optional)	Many-to-One


Users
user_id
name
role
contact
password
email
role
Makes
Places
Reservations
reservation_id
table_id
status
user_id
reservation_time
contains
Tables
table_id
capacity
availability
Orders
order_time
order_id
total_amount
user_id
payment_status
table_id
status
Gives
Get
requires
order
OrderItems
order_item_id
order_id
total_price
item_id
quantity
Feedback
feedback_id
user_id
rating
comments
order_id
Payments
order_id
payment_id
payment_type
status
amount
appear
Menu_items
name
items_id
category
price
availability
ingredients
from
Inventory
quantity
name
item_id
supplier_id
expiry_date
Archived_Order_Items
item_id
total_price
order_id
menu_item_id
quantity
have
Archived_Orders
order_id
archive_time
user_id
order_time
table_id
payment_status
status
total_amount

6. UI/UX Explanation

Overview

The UI of Khata is implemented in Python using the tkinter library, providing a simple, event-driven graphical interface that aligns with the workflow of restaurant staff. The interface is role-based, distinguishing between Admin and Waiter users.

---

1. **Admin Dashboard:**
   - Navigation buttons for:
     - Manage Users
     - Manage Menu
     - View & Manage Orders
     - Logout
   - Designed to give admins full control over the system.
2. **Waiter Dashboard:**
   - Navigation buttons for:
     - Create New Order
     - View Orders
     - Generate Bill
     - Logout
   - Focused on quick order entry and order tracking.
3. **User Management Form (Admin Only):**
   - Add, Edit, Delete user accounts.
   - Input validation for username, password, and role selection dropdown.
   - Displays list of existing users.
4. **Menu Management Form (Admin Only):**
   - CRUD operations on menu items.
   - Fields for item name, category (dropdown), price, availability (checkbox).

- Real-time update of menu list display.

5. **Order Creation Form (Waiter, Admin, Receptionist, Customer):**
   - Different level of access for different users
   - Add multiple items per order.
   - Submit button to save order.
   - Displays current order summary including total bill amount**.**
6. **Order Management Form:**
   - List of all orders with filters by status (Pending, Served, Paid).
   - Ability to update order status.
   - Option to cancel or modify order before payment.
7. **Billing Screen:**
   - Displays detailed bill of order with items, quantities, prices, and total.
   - Button to mark order as paid.

Conclusion:

The Khata Restaurant Management System successfully streamlines daily restaurant operations by integrating order processing, menu management, user role access, and billing into a single Python-based GUI application supported by an SQL database.

Key takeaways from the project:

- How to implement Role-Based Access Control is implemented to separate Admin and Staff functionalities.
- Real-time Order and Menu Management simplifies tasks for both staff and management.
- The Streamlit GUI provides a clean and functional interface suitable for a busy restaurant environment.
- The database schema ensures normalization, integrity, and scalability of data storage.
- All essential restaurant processes — from login to order billing — are encapsulated within the application in a secure and user-friendly manner.

The system demonstrates how a full-stack project can be implemented using Python (for frontend and backend logic) with SQLite or MySQL (for persistent data storage), forming a solid foundation for more complex restaurant automation systems.

Future Work:

While Khata meets the core needs of a small-to-medium restaurant, several enhancements can be pursued:

1. Customer Loyalty System:
   - Track returning customers and apply loyalty discounts or reward points.
2. Inventory Management:
   - Monitor stock levels of ingredients and notify when reordering is needed.
3. Analytics Dashboard:
   - Add graphical views for daily sales, most ordered items, staff performance, etc.
4. Online Order Integration:
   - Connect with a web or mobile front-end for customer self-service and takeaway orders.
5. Invoice PDF Generation:
   - Automatically generate downloadable PDF bills for customer records.
6. Multi-branch Support:
   - Add support for multiple restaurant branches with shared or distributed databases.

These additions would further increase the system's efficiency, security, and competitiveness in real-world deployment scenarios.

8. References

- Python Official Documentation – https://docs.python.org/3/
- MySQL Documentation – https://dev.mysql.com/doc/
- W3Schools SQL Tutorial – https://www.w3schools.com/sql/.
- Lecture Notes and Labs from GIK Institute – Database Management System Course (Fall 2024)