MOUNT LEARNER ACADEMY

Bajaranagar, Dobhi, Jaunpur, 222148

SESSION 2025-26

A Project Report on

Restaurant Management System

For CBSE 2026 Examination

[As a part of the Computer Science Course(083)]

SUBMITTED BY:

SHREYANSH JAISWAL

Class: XII 'A'

Roll no:

SUBMITTED TO:

SAURABH KUMAR

PGT (Computer Science)

CERTIFICATE

THIS IS TO CERTIFY THAT SHREYANSH JAISWAL STUDYING IN CLASS XII A, HAS SATISFACTORILY COMPLETED A PROJECT WITH THE TITLE (**Restaurant Management System**) UNDER THE GUIDANCE OF Mr. SAURABH KUMAR, PGT (COMPUTER SCIENCE) DURING THE ACADEMIC YEAR 2025-26 IN PARTIAL FULFILLMENT OF "COMPUTER SCIENCE" PRACTICAL EXAMINATION OF CENTRAL BOARD OF SECONDARY EXAMINATION (CBSE)

Internal Examiner :

_____

External Examiner :

_____

PRINCIPAL :

_____

ACKNOWLEDGEMENT

I warmly acknowledge the continuous encouragement and timely suggestions offered by our Principal, Mr. Abhishek Singh. I extend my hearty thanks for giving me the opportunity to make use of the facilities available in the campus to carry out the project successfully.

I am highly indebted to Mr. **SAURABH KUMAR** (PGT Computer Science) for the constant supervision, providing necessary information, and supporting in completing the project. I would like to express my gratitude towards them for their kind cooperation and encouragement.

Finally, I extend my gratefulness to one and all who are directly or indirectly involved in the successful completion of this project work.

NAME: SHREYANSH JAISWAL

CLASS: XII 'A'

SIGN:

INTRODUCTION

The Restaurant Management System is a Python–MySQL(SQLITE3) based project designed to simplify restaurant operations. It provides modules for managing the menu, customer records, and order processing.

The system allows CRUD operations such as adding, updating, deleting, and viewing menu items as well as customer details. Orders placed by customers are linked with their records and stored in a database, ensuring easy retrieval and tracking.

The system also generates bills with GST calculation and provides a daily sales report to analyze revenue. By integrating Python with MySQL, the project demonstrates database connectivity, data handling, and practical use of SQL queries.

PYTHON

Python is a versatile and widely used programming language with features that make it a favorite among beginners and professionals alike. Here's why Python stands out:

1. Simplicity and Readability:

- Python's clean syntax emphasizes readability, making it easy to learn and write.
- Ideal for beginners, as the code structure is intuitive and less complex than many other languages.

2. Wide Range of Applications:

- Used in web development, data science, artificial intelligence, machine learning, automation, and game development.
- Supports a variety of domains with specialized libraries like Django, NumPy, Pandas, and TensorFlow.

3. Flexibility in Programming Paradigms:

- Allows procedural, object-oriented, and functional programming, catering to diverse problem-solving approaches.

4. Extensive Library and Community Support:

- Comes with a robust standard library, reducing the need to write code from scratch for many tasks.
- Backed by a large and active community, ensuring continuous updates, resources, and troubleshooting assistance.

5. Efficiency and Productivity:

- Speeds up development with features like dynamic typing and automatic memory management.
- Facilitates rapid prototyping and efficient handling of complex tasks.

SYSTEM IMPLEMENTATION

Hardware used:

- PC (Processor: Intel(R) Core(TM) i3, Ram: 8.00 GB)
- Laptop

Software used:

- OS: Microsoft Windows® 11
- Python IDLE (Latest Version)
- Webbrowser

Technologies Used:

- Python – Programming Language
- datetime Module – Used to handle order timestamps and daily sales reports.
- **os Module** (optional):- Used for clearing the console to improve user interface.
- Tabulate **module** (External):- Displays results in **tabular format** for better readability.
- sqlite3 module:- Provides SQLite database connectivity.

APPLICATION:

RESTAURANT MANAGEMENT SYSTEM

The **Restaurant Management System** is a Python-based application integrated with an SQLite3 database. It is designed to simplify and automate common restaurant operations such as **menu management, customer management, order processing, and generating daily sales reports**.

The system maintains three main tables in the database: **Menu**, **Customer**, and **Orders**. Users can add new menu items, store customer details, and record orders by linking customers with selected food items. The project also calculates and displays a **daily sales report**, showing the total number of items sold and the revenue generated for the day.

For better presentation, the project uses the **tabulate module** to display records in a structured tabular format instead of raw text output. This improves readability and provides a more professional interface.

SOURCE CODE

SOURCE CODE ONLINE FILE LINK:-

https://github.com/ethicslearn/Restaurant-Management-System-Python-MySQL-.git

```python
import sqlite3
from datetime import datetime
from tabulate import tabulate    # For pretty tables


# ---------------- DATABASE CONNECTION ---------------- #
db = sqlite3.connect("restaurant.db")
cursor = db.cursor()


# ---------------- CREATE TABLES ---------------- #
cursor.execute('''
CREATE TABLE IF NOT EXISTS Menu (
    ItemID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
    Price REAL NOT NULL,
    Category TEXT
)
''')
```

```python
cursor.execute('''
CREATE TABLE IF NOT EXISTS Customer (
    CustomerID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name TEXT NOT NULL,
    Phone TEXT,
    Email TEXT
)
''')

cursor.execute('''
CREATE TABLE IF NOT EXISTS Orders (
    OrderID INTEGER PRIMARY KEY AUTOINCREMENT,
    CustomerID INTEGER,
    ItemID INTEGER,
    Quantity INTEGER,
    OrderDate TEXT,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (ItemID) REFERENCES Menu(ItemID)
)
''')

db.commit()
```

```python
# --------------- FUNCTIONS --------------- #

# Add Menu Item
def add_menu_item():
    name = input("Enter Item Name: ")
    price = float(input("Enter Item Price: "))
    category = input("Enter Category: ")
    cursor.execute("INSERT INTO Menu (Name, Price, Category) VALUES (?, ?, ?)"
        , (name, price, category))
    db.commit()
    print("✅ Item added successfully!")


# View Menu
def view_menu():
    cursor.execute("SELECT * FROM Menu")
    items = cursor.fetchall()
    if items:
        print("\n------ MENU ------")
        print(tabulate(items, headers=["ItemID", "Name", "Price", "Category"],
            tablefmt="fancy_grid"))
    else:
        print("⚠ No menu items found.")
```

```python
# Add Customer
def add_customer():
    name = input("Enter Customer Name: ")
    phone = input("Enter Phone: ")
    email = input("Enter Email: ")
    cursor.execute("INSERT INTO Customer (Name, Phone, Email) VALUES (?, ?,
        ?)", (name, phone, email))
    db.commit()
    print("✅ Customer added successfully!")

# View Customers
def view_customers():
    cursor.execute("SELECT * FROM Customer")
    customers = cursor.fetchall()
    if customers:
        print("\n------ CUSTOMERS ------")
        print(tabulate(customers, headers=["CustomerID", "Name", "Phone",
            "Email"], tablefmt="fancy_grid"))
    else:
        print("⚠ No customers found.")
```

```python
# Place Order
def place_order():
    view_customers()
    cust_id = int(input("Enter Customer ID: "))
    view_menu()
    item_id = int(input("Enter Item ID: "))
    qty = int(input("Enter Quantity: "))
    order_date = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    cursor.execute("INSERT INTO Orders (CustomerID, ItemID, Quantity,
        OrderDate) VALUES (?, ?, ?, ?)",
                    (cust_id, item_id, qty, order_date))
    db.commit()
    print("✅ Order placed successfully!")
```

```python
# View Orders
def view_orders():
    cursor.execute('''
    SELECT Orders.OrderID, Customer.Name, Menu.Name, Orders.Quantity, Orders
        .OrderDate
    FROM Orders
    JOIN Customer ON Orders.CustomerID = Customer.CustomerID
    JOIN Menu ON Orders.ItemID = Menu.ItemID
    ''')
    orders = cursor.fetchall()
    if orders:
        print("\n------ ORDERS ------")
        print(tabulate(orders, headers=["OrderID", "Customer", "Item",
            "Quantity", "OrderDate"], tablefmt="fancy_grid"))
    else:
        print("⚠ No orders found.")


# Daily Sales Report
def daily_sales_report():
    today = datetime.now().strftime("%Y-%m-%d")
    cursor.execute('''
    SELECT Menu.Name, SUM(Orders.Quantity), SUM(Orders.Quantity * Menu.Price)
    FROM Orders
    JOIN Menu ON Orders.ItemID = Menu.ItemID
    WHERE date(OrderDate) = ?
    GROUP BY Menu.Name
    ''', (today,))
    report = cursor.fetchall()
    if report:
        print(f"\n📊 Sales Report for {today}")
        print(tabulate(report, headers=["Item", "Total Sold", "Revenue (₹)"],
            tablefmt="fancy_grid"))
    else:
        print("⚠ No sales found for today.")


# ---------------- MAIN MENU ---------------- #
def main():
    while True:
        print("\n===== RESTAURANT MANAGEMENT SYSTEM =====")
        print("1. Add Menu Item")
        print("2. View Menu")
        print("3. Add Customer")
        print("4. View Customers")
        print("5. Place Order")
        print("6. View Orders")
        print("7. Daily Sales Report")
        print("8. Exit")

        choice = input("Enter choice: ")
```

```python
        if choice == "1":
            add_menu_item()
        elif choice == "2":
            view_menu()
        elif choice == "3":
            add_customer()
        elif choice == "4":
            view_customers()
        elif choice == "5":
            place_order()
        elif choice == "6":
            view_orders()
        elif choice == "7":
            daily_sales_report()
        elif choice == "8":
            print("👋 Exiting...")
            db.close()
            break
        else:
            print("❌ Invalid choice! Try again.")

# Run program
if __name__ == "__main__":
    main()
```

PROGRAM STARTUP MENU:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: █
```

Adding ITEM Menu:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: 1
Enter Item Name: pissa
Enter Item Price: 250
Enter Category: snacks
✅ Item added successfully!
```

View Item Menu:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: 2

------ MENU ------
```

ItemID	Name	Price	Category
1	roti	5	chapati
2	Pissza	250	Fast Food
3	pissa	250	snacks

Adding Customer Data:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: 3
Enter Customer Name: Raj
Enter Phone: 2581735791
Enter Email: raj123@gmail.com
✅ Customer added successfully!
```

VIEWING CUSTOMER DATA:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: 4

------ CUSTOMERS ------
```

CustomerID	Name	Phone	Email
1	Raj	2581735791	raj123@gmail.com

Placing order:-

```
------ CUSTOMERS ------
```

CustomerID	Name	Phone	Email
1	Raj	2581735791	raj123@gmail.com

```
Enter Customer ID: 1

------ MENU ------
```

ItemID	Name	Price	Category
1	roti	5	chapati
2	Pissza	250	Fast Food
3	pissa	250	snacks

```
Enter Item ID: 2
Enter Quantity: 1
✅ Order placed successfully!
```

Viewing orders:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: 6

------ ORDERS ------
```

OrderID	Customer	Item	Quantity	OrderDate
1	Raj	Pissza	1	2025-08-29 20:57:55

Daily Sales Reports:-

```
===== RESTAURANT MANAGEMENT SYSTEM =====
1. Add Menu Item
2. View Menu
3. Add Customer
4. View Customers
5. Place Order
6. View Orders
7. Daily Sales Report
8. Exit
Enter choice: 7
```

📊 Sales Report for 2025-08-29

Item	Total Sold	Revenue (₹)
Pissza	1	250

REFERENCES

In order to work on this project titled - Restaurant Management System, the following books and websites are referred by me during the various phases of development of the project.

1. Computer science with PYTHON (Sumita Arora) - Textbook for class XII
2. www.python.com
3. https://www.geeksforgeeks.org
4. https://www.w3schools.com
5. ⊕ GitHub - ethicslearn/Restaurant-Manageme...

Other than the above-mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project