**CAFE MANAGEMENT SYSTEM**

**USING PYTHON TKINTER AND SQLITE**

A project dissertation submitted to Bharathidasan University

in partial fulfillment of the requirements

for the award of the Degree of

**MASTER OF SCIENCE IN DATA SCIENCE**

***Submitted by***

**RAGAVI S**

**235229149**

***Guided by***

**Ms.G.Rajalaxmi, M.Sc.,**

**Assistant Professor**



**DEPARTMENT OF DATA SCIENCE**

**BISHOP HEBER COLLEGE (AUTONOMOUS)**

(Nationally Reaccredited at the ‘A++’ Grade by NAAC with the CGPA of 3.69 out of 4)

(Recognized by UGC as “College with Potential for Excellence”)

(Affiliated to Bharathidasan University)

**TIRUCHIRAPPALLI 620017**

**OCTOBER 2023**

**DECLARATION**

I here by declare that the project work presented is originally done by me under the guidance of **G.Rajalakshmi, B.sc., M.sc., Assistant Professor, Department of Data Science , Bishop Heber College (Autonomous), Tiruchirapalli-620 017** and has not been included in any other thesis/project submitted for any other degree

**Name of the candidate : RAGAVI S**

**Register Number : 235229149**

**Batch : 2023-2025**

**Signature of the Candidate**

**DEPARTMENT OF DATA SCIENCE**

**BISHOP HEBER COLLEGE (AUTONOMOUS)**

(Nationally Reaccredited at the ‘A’ Grade by NAAC with the CGPA of 3.58 out of 4)

(Recognized by UGC as “College with Potential for Excellence”)

(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI 620017

**Date:**

**Course Title: Programming and Data Structure with Python**

**Course Code: P23DS101**

**BONAFIDE CERTIFICATE**

This is to certify that the project work titled “TITLE” is a bonafide record of the project work done by Name of the student, Register Number, in partial fulfillment of the requirements for the award of the degree of MASTER OF SCIENCE IN DATA SCIENCE during the period 2021-2023.

The Viva-Voce examination for the candidate Name of the Student, Register Number, was held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Signature of the HOD Signature of the Guide**

**Examiners:**

**1.**

**2.**

**Acknowledgement**

The satisfaction that accompanies the successful completion of this project would be in complete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. I consider myself privileged to express gratitude and respect towards all those who guided us through the completion of this project.

I convey thanks to my project guide G.Rajalakshmi Assistant professor Department of Data Science for providing encouragement, constant support and guidance which was of a great help to complete this project successfully.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn Datascience. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

**Abstract:**

The "**Cafe Management System**" using Python with Tkinter for the GUI and SQLite for data storage is a practical and educational project. This project leverages a user-friendly GUI (Graphical User Interface) and integrates with a SQL (Structured Query Language) database to enhance efficiency, reduce errors, and centralize applicant data. In this project, you'll build a basic system for managing cafe orders, including adding items, displaying orders, and calculating the total bill.

**CAFÉ MANAGEMENT SYSTEM**

**Introduction:**

The Cafe Management System is a comprehensive software solution designed to streamline the operations of a cafe or coffee shop. This project aims to create an efficient, user-friendly, and versatile cafe management application using Python, the Tkinter library for the graphical user interface (GUI), and SQLite for data storage.

**Dependencies:**

The project will be developed using the following technology stack:

Python: The primary programming language for developing the application's core logic.

Tkinter: A Python library for creating the graphical user interface, ensuring cross-platform compatibility.

SQLite: A lightweight, embedded relational database system for storing menu items and customer orders.

**Existing system:**

**Objectives:**

The primary objectives of the Cafe Management System project are as follows:

Menu Management: Create a digital menu that includes items, descriptions, and prices.

Order Management: Allow cafe staff to take customer orders, specify quantities, and calculate the total bill.

**Data Persistence:**

Store menu items and customer orders in an SQLite database for historical records.

User-Friendly Interface: Develop an intuitive and visually appealing GUI to enhance user experience.

**Features:**

The Cafe Management System will offer the following key features:

Menu Display: Display a comprehensive list of menu items, including names, descriptions, and prices.

Order Entry: Enable cafe staff to select menu items, specify quantities, and add them to the customer's order.

Order List: Maintain a real-time list of items in the current order, including quantity and total price.

Total Bill Calculation: Automatically calculate the total bill based on the items and quantities in the order.

Data Storage: Store menu items and customer orders in an SQLite database for future reference.

Menu Management: Provide an administrative interface to update menu items and prices.

**Budget:**

The Cafe Management System project is anticipated to have a minimal budget as it primarily involves software development. Potential budgetary considerations may include:

Development Environment: Costs related to acquiring or maintaining a development machine (if required).

Miscellaneous Expenses: Small expenses associated with testing, documentation, and project-related activities.

**Project Timeline:**

The Cafe Management System project will follow the following timeline:

Week 1: Project Planning and Database Setup

Define project objectives, requirements, and milestones.

Create the SQLite database schema for menu items and orders.

Set up the basic project structure.

Week 2: User Interface Development

Design and implement the graphical user interface using Tkinter.

Develop menu display, order entry, and order list components.

Week 3: Functionality Implementation

Implement order calculation logic.

Develop menu management functionality for administrators.

Ensure seamless data persistence in the SQLite database.

Week 4: Testing and Refinement

Conduct extensive testing, including unit testing and user acceptance testing.

Refine the user interface for optimal usability and aesthetics.

Address and resolve any bugs or issues identified during testing.

Week 5: Documentation and Presentation

Document the project comprehensively, including code documentation and user guides.

Prepare a presentation to demonstrate the project's functionality and features.

Finalize the project report and related materials.

**Source code:**

import sqlite3

conn = sqlite3.connect("cafe.db")

cursor = conn.cursor()

# Create a table for menu items

cursor.execute('''CREATE TABLE IF NOT EXISTS menu (

item\_id INTEGER PRIMARY KEY,

item\_name TEXT,

item\_price REAL)''')

# Create a table for orders

cursor.execute('''CREATE TABLE IF NOT EXISTS orders (

order\_id INTEGER PRIMARY KEY,

item\_id INTEGER,

quantity INTEGER,

total\_price REAL)''')

conn.commit()

conn.close()

This code creates a database named "cafe.db" and two tables: "menu" for storing cafe menu items and "orders" for recording customer orders.

**Build the Tkinter GUI**

import tkinter as tk

import sqlite3

# Function to add an item to the order

def add\_item():

selected\_item = menu\_listbox.get(tk.ACTIVE)

quantity = quantity\_entry.get()

if selected\_item and quantity.isdigit():

item\_id = menu\_items[selected\_item]

item\_price = menu\_prices[selected\_item]

quantity = int(quantity)

total\_price = item\_price \* quantity

# Add the order to the database

conn = sqlite3.connect("cafe.db")

cursor = conn.cursor()

cursor.execute("INSERT INTO orders (item\_id, quantity, total\_price) VALUES (?, ?, ?)",

(item\_id, quantity, total\_price))

conn.commit()

conn.close()

# Update the order list

order\_listbox.insert(tk.END, f"{selected\_item} x{quantity}")

# Update the total bill

total\_bill.set(f"Total Bill: ${calculate\_total\_bill()}")

# Function to calculate the total bill

def calculate\_total\_bill():

conn = sqlite3.connect("cafe.db")

cursor = conn.cursor()

cursor.execute("SELECT SUM(total\_price) FROM orders")

total = cursor.fetchone()[0]

conn.close()

return total if total else 0.0

# Create the main window

window = tk.Tk()

window.title("Cafe Management System")

# Create a list of menu items and their prices

menu\_items = {}

menu\_prices = {"Coffee": 2.5, "Tea": 2.0, "Cake": 3.0, "Sandwich": 5.0}

# Create a menu listbox

menu\_listbox = tk.Listbox(window)

for item in menu\_prices.keys():

menu\_items[item] = len(menu\_items) + 1

menu\_listbox.insert(tk.END, item)

menu\_listbox.pack()

# Create an entry for quantity

quantity\_label = tk.Label(window, text="Quantity:")

quantity\_label.pack()

quantity\_entry = tk.Entry(window)

quantity\_entry.pack()

# Create a button to add items to the order

add\_button = tk.Button(window, text="Add to Order", command=add\_item)

add\_button.pack()

# Create a listbox to display the order

order\_listbox = tk.Listbox(window)

order\_listbox.pack()

# Create a label to display the total bill

total\_bill = tk.StringVar()

total\_bill.set(f"Total Bill: ${calculate\_total\_bill()}")

total\_bill\_label = tk.Label(window, textvariable=total\_bill)

total\_bill\_label.pack()

# Start the Tkinter main loop

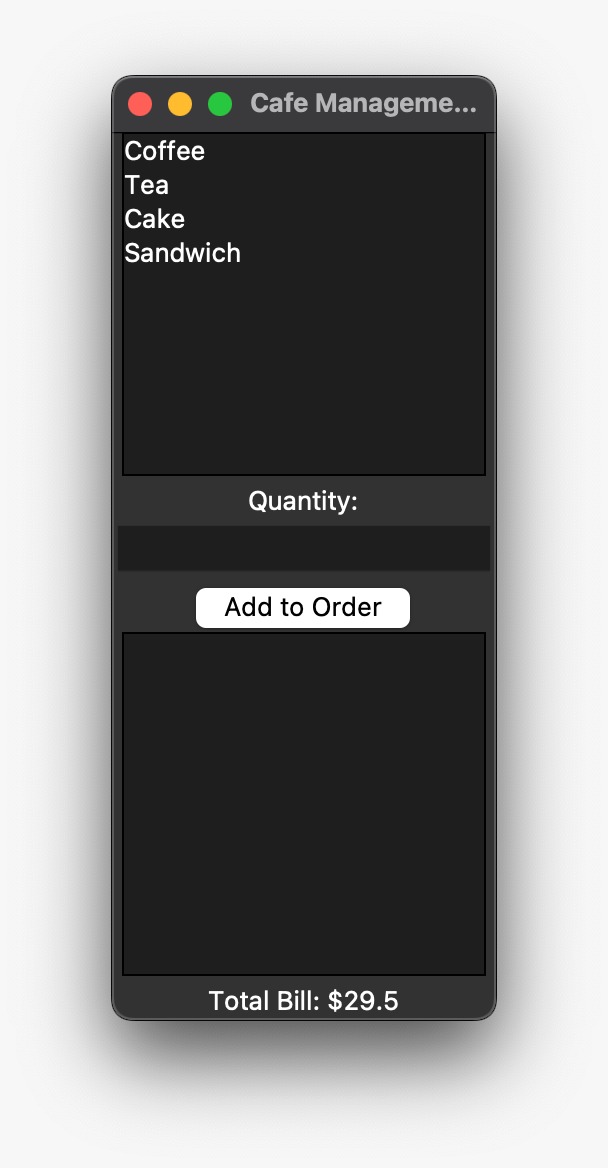
window.mainloop()

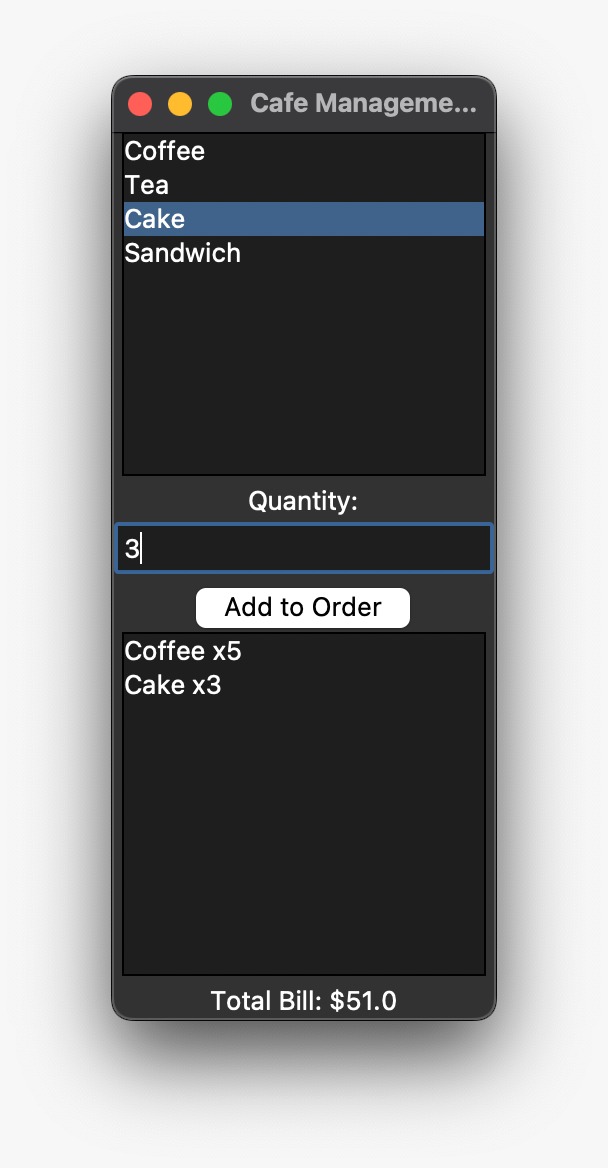
This code sets up a Tkinter window for the cafe management system. Users can select items from the menu, enter the quantity, and add items to the order. The orders are displayed in a listbox, and the total bill is calculated and displayed at the bottom.

Run the Application Run the Python script, and you should see the GUI for the Cafe Management System. Users can interact with the interface to add items to the order and see the total bill.

This is a basic example, and you can further enhance the project by adding features like order history, removing items from orders, or customizing the menu.

**Output:**





**Conclusion:**

The Cafe Management System project endeavors to deliver a powerful and user-friendly solution for cafe and coffee shop management. By utilizing Python, Tkinter, and SQLite, the project aims to meet the diverse needs of cafe businesses while providing a rich and interactive user experience.

This proposal outlines the fundamental aspects of the "Cafe Management System" project, including objectives, features, technology stack, project timeline, and budget. We look forward to the opportunity to develop a robust and valuable solution for cafe owners and staff.