HOTEL TAJ RESTAURANT MANAGEMENT SYSTEM

Project Report

1. PROJECT OVERVIEW

Project Name: Hotel Taj Restaurant Management System

Technology: Python 3.x with Tkinter GUI

Purpose: To create an interactive menu and order management system for a restaurant that allows customers to view items and add them to their order with real-time bill calculation and visual animations.

Duration: [Project Duration]

Team: [Student/Developer Name]

2. INTRODUCTION

Hotel Taj Restaurant Management System is a Python-based desktop application designed to streamline the ordering process at a restaurant. The system provides a user-friendly interface where customers can browse the restaurant menu, add items to their order, and view the running total with engaging visual feedback. This application improves the ordering experience with an attractive design and animated price updates.

3. FEATURES

- □ **Clean GUI Interface** - Visually appealing design with attractive styling
- □ **Menu Display** - Shows all restaurant items with prices in an organized format
- □ **Add to Order** - One-click buttons to add items to the shopping cart
- □ **Order List** - Dynamically updates the list of selected items
- □ **Bill Calculation** - Automatically calculates and displays the total amount
- □ **Visual Animations** - Animated color effects on the total for enhanced UX
- □ **Multiple Selections** - Supports adding the same item multiple times
- □ **Real-time Updates** - Price updates instantly as items are added

4. MENU ITEMS

Item	Price (Rs.)
Pizza	150
Burger	100
Salad	250
Cold Coffee	125
Momo	120
Dosa	140

Total Menu Items: 6

Price Range: Rs. 100 - Rs. 250

Average Price: Rs. 147.50

5. SYSTEM REQUIREMENTS

Hardware

- Processor: Intel Core i3 or equivalent
- RAM: 2 GB minimum
- Storage: 100 MB free space

Software

- Python 3.x
- Tkinter (pre-installed with Python)
- Operating System: Windows, macOS, or Linux

6. HOW TO RUN

```
1. Ensure Python 3.x is installed on your system
2. Save the script as: hotel_taj.py
3. Open terminal/command prompt
4. Run: python hotel_taj.py
5. The application window will open automatically
6. Click "Add to Order" buttons to add items
7. Order list and total price update automatically
```

7. HOW IT WORKS

1. **Welcome Message** - Main window displays a greeting and restaurant name
2. **Menu Dictionary** - Stores all restaurant items and their prices
3. **BakeryApp Class** - Manages the entire application interface
4. **Add Button** - When clicked, adds the item to the order list
5. **Total Calculator** - Updates the bill amount in real-time
6. **Order Display** - Shows selected items with individual prices
7. **Animation** - Color toggle animation on total for visual confirmation

---

8. KEY COMPONENTS

Components Used:

- **Tkinter Widgets:** Frame, Label, Button, Listbox, Text
- **Data Structure:** Python Dictionary for menu storage
- **Logic:** Event-driven programming with callback functions

Code Structure:

```
hotel_taj.py
├── Menu Dictionary
├── BakeryApp Class
│   ├── __init__()
│   ├── display_menu()
│   ├── add_to_cart()
│   ├── animate_total()
│   └── update_total()
└── Main Execution
```

Key Methods:

- **add_to_cart()** - Handles item addition and total updates
- **animate_total()** - Creates color toggle animation on the total price label
- **update_total()** - Recalculates and displays the total amount

---

9. TESTING

Feature	Status
Menu Display	□ Working

Feature	Status
Add to Order	□ Working
Bill Calculation	□ Working
Order List Update	□ Working
Price Animation	□ Working
Multiple Selections	□ Working
UI Responsiveness	□ Working

10. STRENGTHS

□ Clean and attractive interface design

□ Fast performance with minimal resource usage

□ Engaging visual feedback with animations

□ Lightweight and portable application

□ Easy to use for restaurant staff and customers

□ Supports multiple item selections

□ Easy to modify and extend

11. LIMITATIONS

⚠ No item quantity selection (adds fixed quantity per click)

⚠ Cannot remove items from the order

⚠ No data persistence (order lost when app closes)

⚠ No payment gateway integration

⚠ No database backend

⚠ No order history or logging

⚠ Single restaurant menu only

12. FUTURE ENHANCEMENTS

1. **Quantity Selection** - Allow users to specify quantities for each item
2. **Remove Items** - Add delete button for each order item
3. **Clear Order** - Reset all items with one click
4. **Bill Printing** - Generate and print receipts
5. **Order History** - Save and retrieve previous orders
6. **Database Integration** - Store orders in SQLite/MySQL
7. **Payment Gateway** - Integrate online payment methods
8. **Discount System** - Apply promotional codes and discounts
9. **Admin Panel** - Manage menu items, prices, and restaurant settings
10. **Search Function** - Quick search for menu items

11. **Table Management** - Assign orders to specific tables
12. **Multi-Restaurant Support** - Manage multiple restaurant branches

---

13. USER INTERACTION

Order Process:

1. Application window opens showing welcome message
2. Restaurant menu displays with item names and prices
3. "Add to Order" buttons appear next to each item
4. Click any "Add to Order" button to add that item
5. Order list below updates with the new item
6. Total price label updates with smooth color animation
7. Continue adding items as needed
8. Order list shows all selected items with prices

Visual Feedback:

- Color animation on total price when items are added
- Dynamic order list updates in real-time
- Clear layout for easy navigation

---

14. CONCLUSION

The Hotel Taj Restaurant Management System is a simple yet effective solution for restaurant ordering. The application successfully demonstrates core concepts of Python GUI programming with Tkinter. While the current version meets basic ordering requirements, implementing the suggested enhancements would transform it into a complete point-of-sale system suitable for commercial restaurant use.

Recommendations:

- Deploy as a desktop application for in-store use
- Expand with database connectivity for order tracking
- Add payment integration for streamlined transactions
- Develop table management features for multi-table restaurants
- Create an admin panel for menu and pricing management
- Integrate with kitchen display systems (KDS)

---

15. ACKNOWLEDGMENTS

Special thanks to Tkinter documentation and Python community for providing excellent resources and libraries that made this project possible.

Project Completion Date: November 2025

Report Generated: 24 November 2025
