IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

A Python Tkinter-Based Dual-Role Food Ordering and Management System

Mr. Sujal Nimsarkar[1] and Prof. Pooja Pimpalshende[2]
Student, Department of Computer Science and Engineering (Data Science)[1]
Supervisor, Department of Computer Science and Engineering (Data Science)[2]
Tulsiramji Gaikwad Patil College of Engineering & Technology, Nagpur, India
nimsujal.250104@gmail.com

Abstract: *This research presents the design and implementation of a desktop-based food ordering and management system built using Python's Tkinter graphical interface and SQLite database engine. The system provides separate interfaces for customers and sellers, allowing both roles to interact with the platform according to their requirements. Customers can browse dishes, apply live search filters, adjust quantities, place orders, track their order status, and provide feedback. Sellers can manage menus, monitor incoming orders in real time, update order status, and modify their restaurant details. The application integrates dynamic theming, image handling, and interactive widgets to enhance user experience. The system offers a lightweight, offline-compatible solution suitable for small-scale restaurants and college projects, eliminating the dependence on costly online systems. Experimental testing demonstrates that the system is responsive, stable, and efficient for typical food-ordering operations*

Keywords: Python Tkinter, Food Ordering System, SQLite Database, Offline Application, Restaurant Management, Graphical User Interface (GUI)

I. INTRODUCTION

In today's world, digital technology has changed the way people order and receive food. Earlier, customers had to either visit restaurants or call on the phone to place an order. This often-caused problems such as wrong orders, waiting time, communication issues, and lack of proper records. With the increasing use of computers and mobile devices, people now prefer easy, fast, and convenient methods of ordering food. Because of this, online platforms like Zomato, Swiggy, and Uber Eats have become very popular. These platforms offer menu browsing, online payments, live order tracking, and delivery updates. However, they also come with several challenges, especially for small food shops, local restaurants, and college canteens. They require internet connection, paid registration, high commission charges, and technical knowledge. Many small businesses cannot afford or manage these requirements.

To solve these problems, there is a need for a simple and low-cost system that does not depend on the internet. A desktop-based food ordering system is a suitable solution for such environments. Desktop applications are easy to install, require less setup, do not need hosting or servers, and can work offline. They are helpful for small shops, hostels, canteens, or college projects where owners or students do not have access to advanced technologies. Python is an excellent programming language for building such systems. It is simple to understand, easy to code, and provides many useful libraries. One of the most commonly used libraries for creating graphical user interfaces (GUI) in Python is Tkinter. Tkinter allows developers to design attractive application windows with buttons, labels, frames, images, and input fields. It is lightweight and does not require extra installations, which makes it perfect for student projects.

For storing data like user details, order history, restaurant information, and menu items, SQLite is used. SQLite is a very simple and powerful database that works without a server. It stores data in a single file and can be easily connected with Python. This makes it ideal for offline desktop applications where internet and cloud storage are not available. With the combination of Python Tkinter and SQLite, it becomes possible to create a complete food ordering system that is fast, simple, and effective.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

The main aim of this project is to develop a Food Ordering and Management System that can be used by both customers and sellers. Customers can browse items, add them to the cart, place orders, track their order status, and give ratings or reviews. Sellers can add new dishes, update prices, remove items, accept or reject orders, and view customer feedback. Since the system works offline, sellers can manage everything directly from their computer without depending on the internet. This makes it highly useful for small businesses that want a digital solution without extra costs. Another important feature of the system is the user-friendly interface. The application is designed in such a way that even non-technical users can easily understand how to use it. All screens such as login, sign-up, menu view, cart, checkout, and order status are built with clarity and simplicity. The system also supports theme switching, where users can choose between light mode and dark mode as per their comfort. This improves accessibility and gives a modern touch to the application.

From a student point of view, developing such a system helps in learning many important concepts like GUI development, database handling, file management, event-driven programming, and creating user-friendly applications. It also helps students understand how real-life applications work behind the scenes. This project can also be helpful for internship work, final-year projects, and academic demonstrations.

Overall, this introduction explains why food ordering systems are important today, especially in small businesses and educational environments. It also highlights the role of Python, Tkinter, and SQLite in creating an offline desktop application. The project not only solves common problems faced by restaurant owners but also provides a learning opportunity for students to work on a practical and useful software system. This research contributes a simple, cost-effective, and efficient solution that can be easily used, modified, and expanded according to future needs.

II. SYSTEM DESIGN

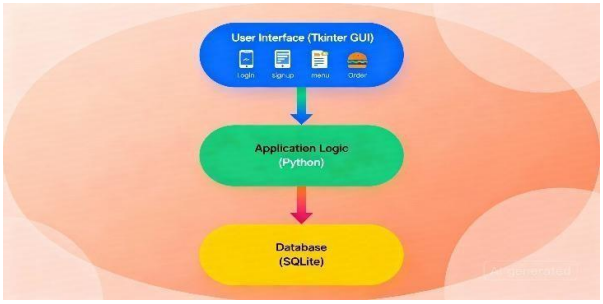

Fig.1 System Design of project

1. Presentation layer: This layer is built using Python Tkinter and provides all graphical interfaces such as login, menu display, cart, order tracking, and seller dashboard. It includes separate views for customers and sellers and supports light/dark themes.
2. Business logic layer: This layer manages the core operations such as login verification, cart updates, menu management, order processing, order status updates, image handling, review processing, and automatic dashboard refreshing. It connects the UI with the database.
3. Data layer: Implemented using, this layer stores all persistent data such as user accounts, menu items, orders, restaurant details, and reviews. It ensures fast and secure offline data management.

Overall workflow: Customers browse the menu, add items to the cart, place orders, and track status. Sellers manage dishes, view incoming orders, and update order statuses. All interactions pass through the business logic layer and are stored in the SQLite database.

III. RESEARCH METHODOLOGY

The research methodology followed in this project is based on a developmental and experimental approach aimed at designing, implementing, and testing an offline Food Ordering and Management System using Python Tkinter and SQLite.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

1) Requirement Analysis: The requirements of both customers and sellers were identified. Customer needs included menu viewing, image-based browsing, cart handling, order placement, order tracking, and review submission. Seller requirements included menu management, order monitoring, updating order statuses, and viewing customer feedback. Both functional and non-functional requirements were documented for clarity.

2) System Design: A three-layer architecture was planned, consisting of the Presentation Layer, Business Logic Layer, and Data Layer. Tkinter was selected for GUI development due to its simplicity and offline compatibility, while SQLite was chosen for lightweight local data storage. Modules such as authentication, customer operations, seller functions, order handling, theme management, and database operations were defined.

3) Implementation: The system was developed module-wise using Python 3.x. Tkinter components were used to design the interface, including frames, labels, buttons, images, and entry fields. Business logic was added to manage login, menu updates, cart operations, order placement, review handling, and dashboard refreshing. SQLite tables were created for users, dishes, orders, and feedback. Image handling and theme switching were also integrated.

4) Testing: Functional testing was performed for each module to ensure that all features worked correctly. Integration testing was conducted after merging modules to verify smooth interaction between components. Performance testing checked speed, responsiveness, and offline efficiency. User acceptance testing (UAT) ensured that the interface was easy to understand and operate.

5) Evaluation: The system was evaluated on usability, accuracy, responsiveness, and reliability. Test results indicated that the system performed efficiently in offline environments and met the needs of small restaurants, canteens, and college projects by providing smooth order management and reduced manual errors.

IV. RESULT AND DISCUSSION

The developed Food Ordering and Management System was tested with different users to check its performance and ease of use. The results showed that the system worked smoothly in offline mode and responded quickly during menu browsing, searching items, placing orders, and updating order status.

Customers found the interface simple to understand because the menu, images, buttons, and screens were clearly arranged. The theme-switching option also improved comfort for users. On the seller side, features like adding dishes, editing menus, and checking new orders worked correctly and helped reduce manual mistakes.

The SQLite database handled all data operations efficiently, such as storing user details, menu items, orders, and reviews. Since the system does not require internet, it is useful for small shops, college canteens, and places with weak connectivity.

Overall, the system improved the ordering process, reduced confusion, and made communication between customers and sellers easier. The project also helped in understanding practical concepts of GUI development, database handling, and real-life system implementation.

V. CONCLUSION

The Food Ordering and Management System developed using Python, Tkinter, and SQLite successfully demonstrates how a simple, offline, and user-friendly desktop application can replace traditional manual ordering methods used by small restaurants, canteens, and local food vendors. The system provides a smooth platform where customers can browse menus with images, place orders, track their order status, and submit reviews, while sellers can efficiently manage dishes, update menus, and process incoming orders in real time. All core functionalities—such as login, cart operations, menu handling, order management, and review submission—were implemented effectively through a modular design approach.

The use of SQLite ensures secure and reliable storage of data without requiring internet or server configuration, making the system affordable and accessible for small businesses. Likewise, the Tkinter interface allows simple and intuitive navigation, making the application suitable even for non-technical users. Testing results show that the system performs well, responds quickly, and handles multiple orders without difficulties. The project achieves its main goal of providing a practical, cost-effective solution that improves accuracy, reduces order confusion, and enhances communication between customers and sellers.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

Overall, the project demonstrates the potential of Python-based GUI applications in solving realworld problems. It provides valuable learning experience in software development, database management, and user interface design. This system can be further improved by adding online payment integration, mobile app support, cloud connectivity, and delivery tracking, making it suitable for larger-scale businesses in the future.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to everyone who supported me throughout the development of this research work. I am deeply thankful to Prof. Pooja Pimpalshende, my guide, for her continuous guidance, valuable suggestions, and encouragement during the completion of this project. Her expert advice and constructive feedback played a crucial role in shaping this research.

I would also like to thank the Department of Computer Science and Engineering (Data Science), Tulsiramji Gaikwad Patil College of Engineering & Technology, Nagpur, for providing the necessary resources and a supportive environment for carrying out this work.

REFERENCES

[1]. M. Grayson and T. Nelson, "Development of GUI-Based Python Applications Using Tkinter," Journal of Software Engineering and Applications, vol. 14, no. 3, pp. 125–139, 2021.

[2]. A. Kumar and R. Singh, "SQLite as a Lightweight Database Solution for Local Applications," International Journal of Computer Applications, vol. 176, no. 32, pp. 22–27, 2020.

[3]. S. Patel and N. Verma, "A Comparative Study of Online Food Ordering Systems," International Journal of Advanced Research in Computer Science, vol. 10, no. 5, pp. 15–21, 2019.

[4]. K. Bansal and S. Gupta, "Design and Analysis of Offline Food Ordering Systems for Small Enterprises," International Journal of Computer Trends and Technology, vol. 70, no. 1, pp. 45–52, 2022.

[5]. P. Shrestha, "Improving User Experience in Desktop Applications Using Python Tkinter Framework," International Journal of Innovative Research in Computer Science & Technology, vol. 8, no. 4, pp. 97–103, 2020.

[6]. R. Mishra and A. Sharma, "Development of a Desktop-Based Restaurant Management and Billing System Using Python Tkinter," International Journal of Advanced Computer Science and Applications, vol. 12, no. 7, pp. 210– 217, 2021.