SQUARE

A PROJECT REPORT

Submitted by

SARAN JOSHUA S	2024239012
KAILASH N	2024239026
ARAVIND V	2024239027
VETRISELVAN K	2024242018
SHESHA N	2024242022

for the course of

XT3351

OBJECT-ORIENTED PROGRAMMING USING C++

DEPARTMENT OF MATHEMATICS

ANNAUNIVERSITY : CHENNAI 600025

NOVEMBER – 2025

SQUARE

ABSTRACT

CEG SQUARE is an integrated canteen management and food-ordering platform developed using C++ and the Qt framework, with SQLite as the database backend. The system is designed to digitalize and streamline food service operations within educational institutions, providing a unified solution for both students and vendors. It delivers a dual-interface architecture where each user group interactswith a customized workflow tailored to its needs.

For students, CEG SQUARE offers an intuitive interface that enables real-time browsing of menus, seamless cart management, secure order placement, and continuous order-status tracking. This digital approach eliminates long queues, reduces waiting times, and improves overall user convenience. On the vendor side, the system includes comprehensive shop-management utilities such as inventory control, menu updates, order processing workflows, and financial analytics dashboards. These modules support efficient decision-making, better inventory forecasting, and enhanced service delivery.

The application is built upon a robust relational database structure that ensures data consistency, integrity, and efficient transaction handling. By replacing traditional manual ordering methods, CEG SQUARE minimizes communication gaps, reduces operational errors, and improves resource utilization. The inclusion of analytical insights further empowers vendors to understand sales patterns, optimize offerings, and enhance business performance.

Overall, CEG SQUARE serves as a modern, scalable, and user-centered solution that greatly enhances the canteen experience for students while providing vendors with valuable operational intelligence. Its reliability, ease of use, and efficient data-driven workflow make it a transformative system for campus-based food service management.

TABLE OF CONTENTS		
CHAPTER NO.	**TITLE**	**PAGE NO.**
1.	**OVERVIEW**	**1**
2.	**OBJECTIVE OF THE PROJECT**	**2**
3.	**FEASIBILITY STUDY**	
	TECHNICAL FEASIBILITY	**3**
	OPERATIONAL FEASIBILITY	**4**
	DATA FEASIBILITY	**5**
4.	**PROJECT FEATURES CORE MODULES**	**6**
5.	**CORE FUNCTIONALITY**	**7**
6.	**IMPLEMENTATION**	**8**
7.	**SCREENSHOTS**	**9**
8.	**CONCLUSION**	**13**
	REFERENCES	**14**

1. Overview

The traditional canteen system in educational institutions often suffers from inefficiencies like long queues, manual order taking leading to errors, lack of real-time menu updates, and minimal sales insights for vendors.

CEG SQUARE addresses these challenges by introducing a centralized digital platform. The project automates the entire lifecycle of food ordering, from menu discovery to order completion and analytics.

CEG SQUARE is a comprehensive digital canteen management solution designed specifically for educational institutions.

It serves as a bridge between students and vendors, facilitating a smooth, transparent, and efficient transaction process.

The system revolutionizes traditional canteen operations by providing a fully automated, role-based platform that bridges the gap between students and food vendors.

The system is built with a client-server architecture model where the Qt-based GUI client and the SQLite database are encapsulated within a single desktop application, ensuring ease of deployment and reliability.

2. OBJECTIVE OF THE PROJECT

The primary objective of CEG SQUARE is to develop a reliable and user-friendly software solution that modernizes canteen operations. The key goals are:

- **To Digitize Order Management: Eliminate paper-based orders and manual calculations by providing a digital platform for order placement and processing.**
- **To Enhance User Experience: Provide students with a convenient way to browse available food items, place orders from multiple vendors, and track their status in real-time.**
- **To Empower Vendors: Equip vendors with tools to manage their shop profile, update their product catalog, control inventory, and process orders efficiently.**
- **To Provide Data-Driven Insights: Generate financial reports and analytics for vendors, helping them understand sales trends and make informed business decisions.**
- **To Ensure System Reliability: Build a robust and scalable application using modern software engineering practices to ensure data integrity and consistent performance.**

3. FEASIBILITY STUDY

3.1 TECHNICAL FEASIBILITY

The project is technically feasible due to the careful selection of technologies:

- **C++ and Qt Framework: C++ offers high performance for database operations, while Qt provides a rich set of libraries for building cross-platform, native-looking graphical user interfaces. The signal-slot mechanism simplifies event handling.**
- **SQLite Database: As a serverless, self-contained SQL database engine, SQLite is ideal for a desktop application. It requires zero configuration and provides a reliable way to store and manage relational data like users, products, and orders.**
- **Development Tools: Mature and free tools like Qt Creator and the GNU compiler chain were used, making the development process smooth and cost-effective.**

3.2 OPERATIONAL FEASIBILITY

The system is highly operable and addresses the core needs of its users:

- **For Students: The interface is intuitive, resembling modern e-commerce applications, which reduces the learning curve.**
- **For Vendors: The tab-based dashboard logically separates tasks (shop setup, product management, order processing, finance), making daily operations straightforward.**
- **Administrative Perspective: The system automates most tasks, reducing the administrative burden of managing orders and financial records manually.**

3.3 DATA FEASIBILITY

The data requirements for the system are well-defined and efficiently managed:

- **Structured Data Model: The database schema is normalized across five core tables (users, shops, products, orders, order_items), minimizing redundancy and ensuring data consistency.**
- **Efficient Handling: SQLite competently handles all data operations, including complex JOINs for generating order histories and financial reports. The use of prepared statements ensures security and performance.**

4. PROJECT FEATURES & CORE MODULES

The system is architecturally divided into two main modules:

A. Student Module:

- **User Authentication: Secure login for students.**
- **Product Catalog: Dynamic browsing of all available food items from different shops with prices and categories.**
- **Shopping Cart Management: Add/remove items, adjust quantities, and view a running total. Includes validation to ensure orders are from a single shop.**
- **Order Placement & Tracking: Place orders and monitor their status (Pending, Preparing, Completed) in real-time through a color-coded history table.**

B. Vendor Module:

- **Shop Registration: Vendors can register their shop by providing a name, slot number, and description.**
- **Product Management: Add, and manage (soft-delete) products, including setting names, prices, and categories.**
- **Order Processing Dashboard: View incoming orders, see customer details and ordered items, and update order status through an Accept→Preparing→Complete workflow.**
- **Financial Analytics: View key metrics including Total Revenue, Today's Revenue, Total Orders, and Completed Orders. Access a detailed payment history.**

5.Core Functionality

The system implements a role-based access control with two main user types:

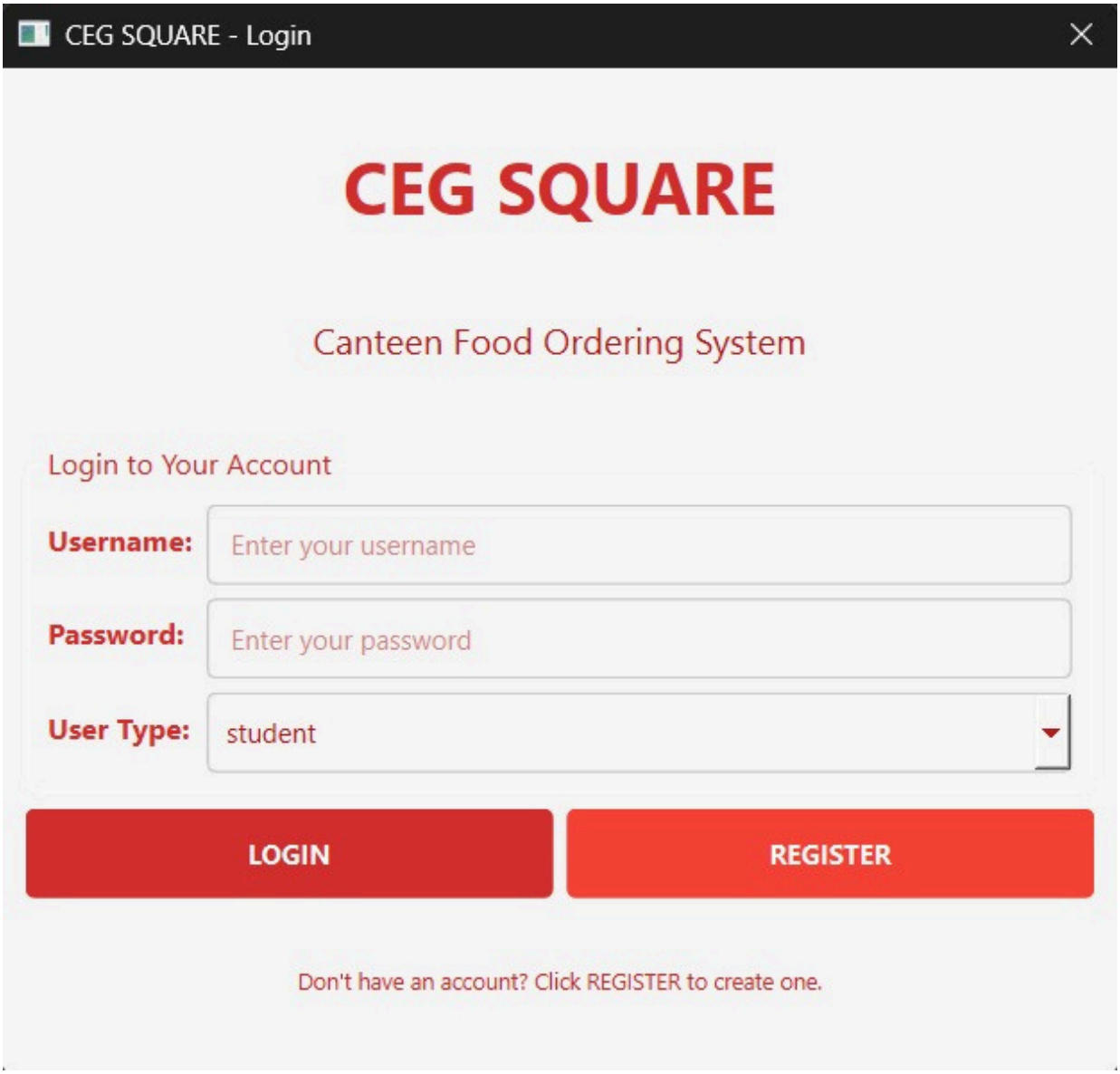
1. **Students - Can:**
    - **Browse available food items from multiple shops**
    - **Add items to cart with quantity selection**
    - **Place orders from a single shop per transaction**
    - **View order history with status tracking**
    - **Real-time cart management with total calculation**
2. **Vendors - Can:**
    - **Register their shop with slot allocation**
    - **Manage product catalog (add/remove items)**
    - **Process incoming orders (accept/prepare/complete)**
    - **Track financial metrics and revenue analytics**
    - **Monitor order statistics and payment history**

6. *IMPLEMENTATION*

The implementation followed a structured software development lifecycle:

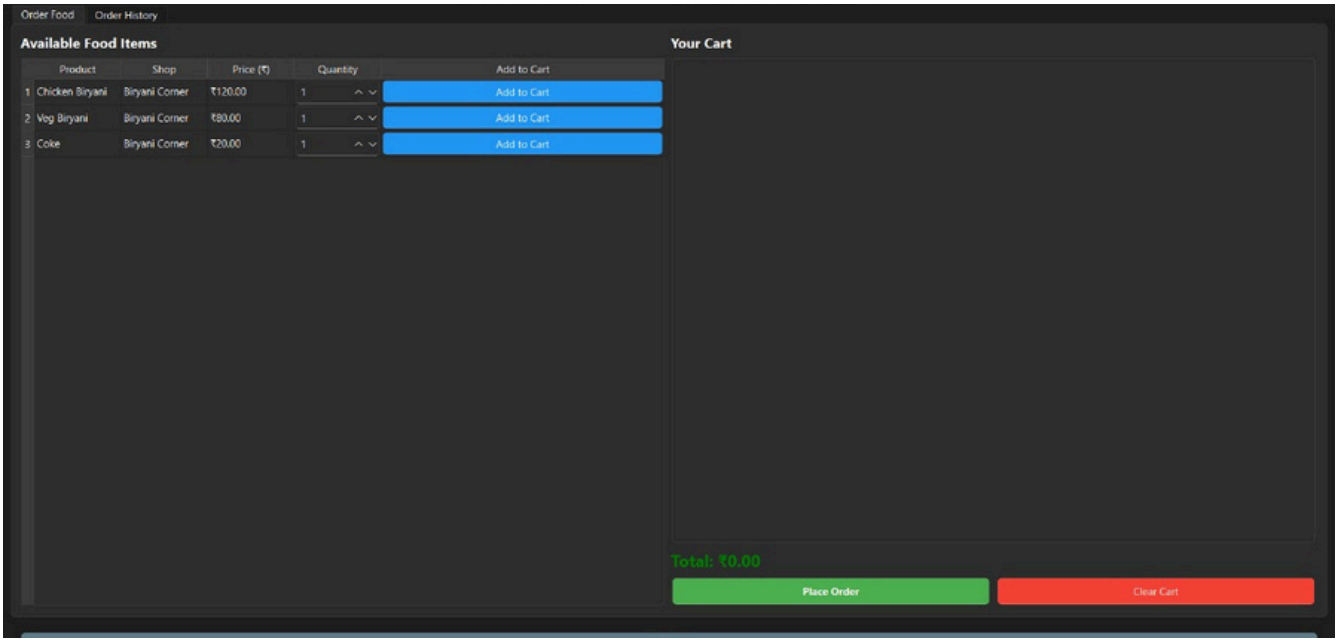
- **Technology Stack: C++17, Qt 5/6, SQLite3.**
- **Architecture: Model-View-Controller (MVC) pattern was implied, with UI forms (View), C++ classes for business logic (Controller), and the database layer (Model).**
- **Database Design: A relational schema was implemented with proper primary and foreign key relationships to maintain referential integrity. Sample data was pre-populated for testing.**
- **Key Algorithms & Logic:**
  - **Singleton Pattern: Used in the DatabaseManager class to ensure a single, global point of access to the database.**
  - **Cart Management Logic: Implemented using a QVector of CartItem structures, with logic to validate single-shop orders.**
  - **Dynamic UI Generation: Buttons for "Add to Cart" and order actions are dynamically created and connected to slots using Qt's signal-slot mechanism.**
- **Testing: Each module was tested iteratively for functionality, including user registration, login, product addition, order placement, and status updates.**

7.SCREEN SHOTS



a).Login Dialog

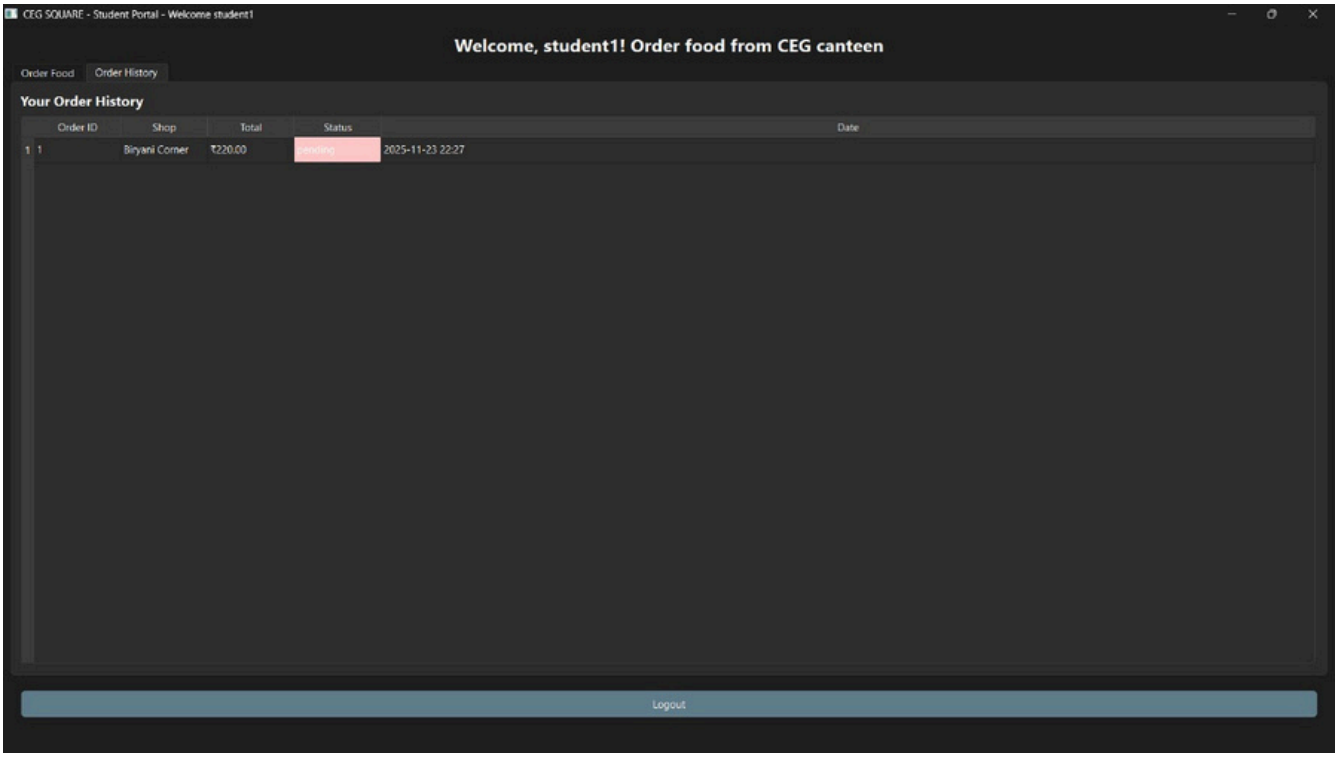
Shows the user authentication interface with role selection for students and vendors.



b).Student Ordering Interface

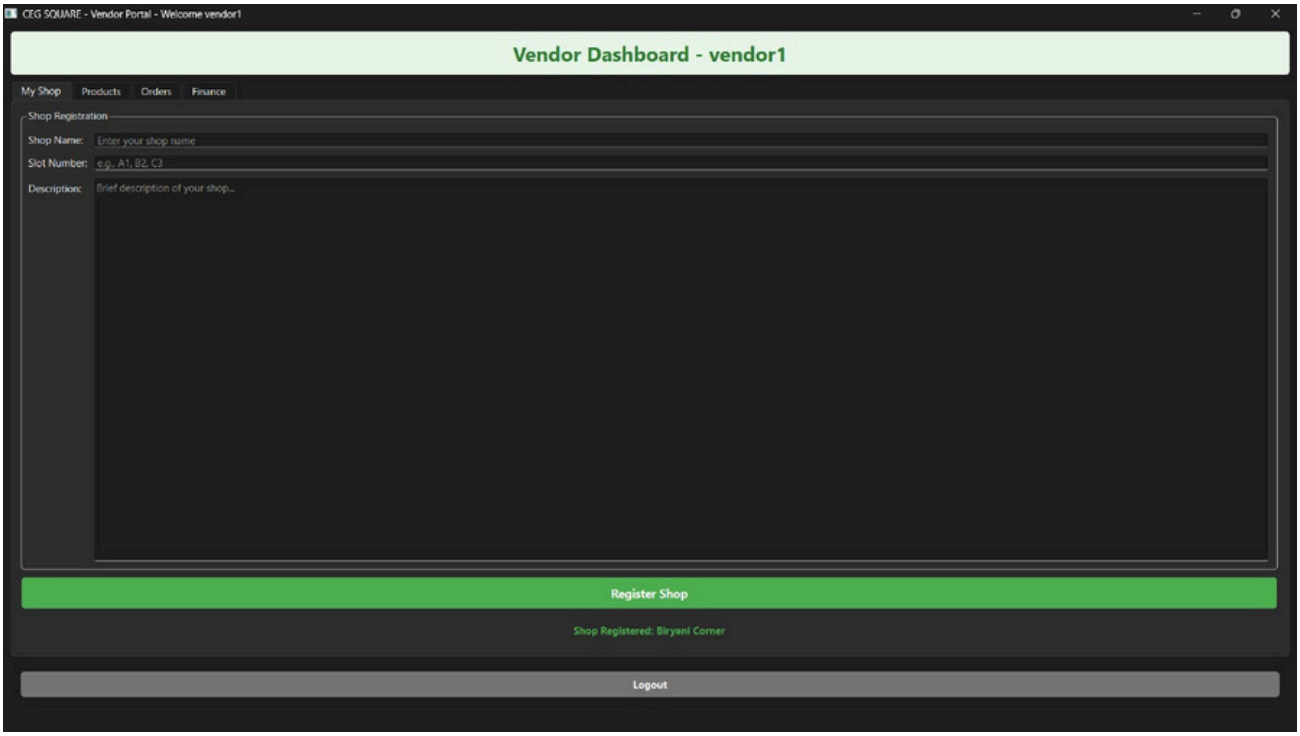
Displays the product catalog, shopping cart, and order managementcontrols for students

.



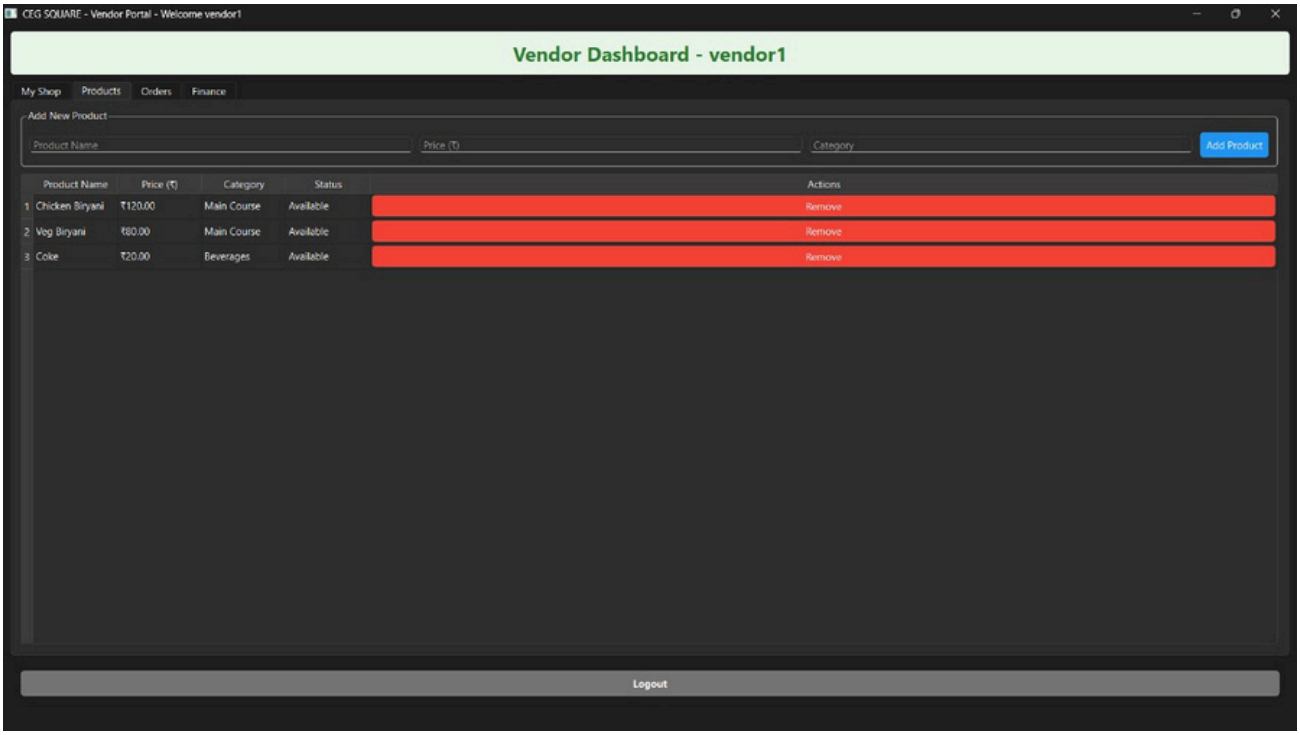
c).Student Window

History Tab - Showing the order history with detailed information including order status and timestamps.



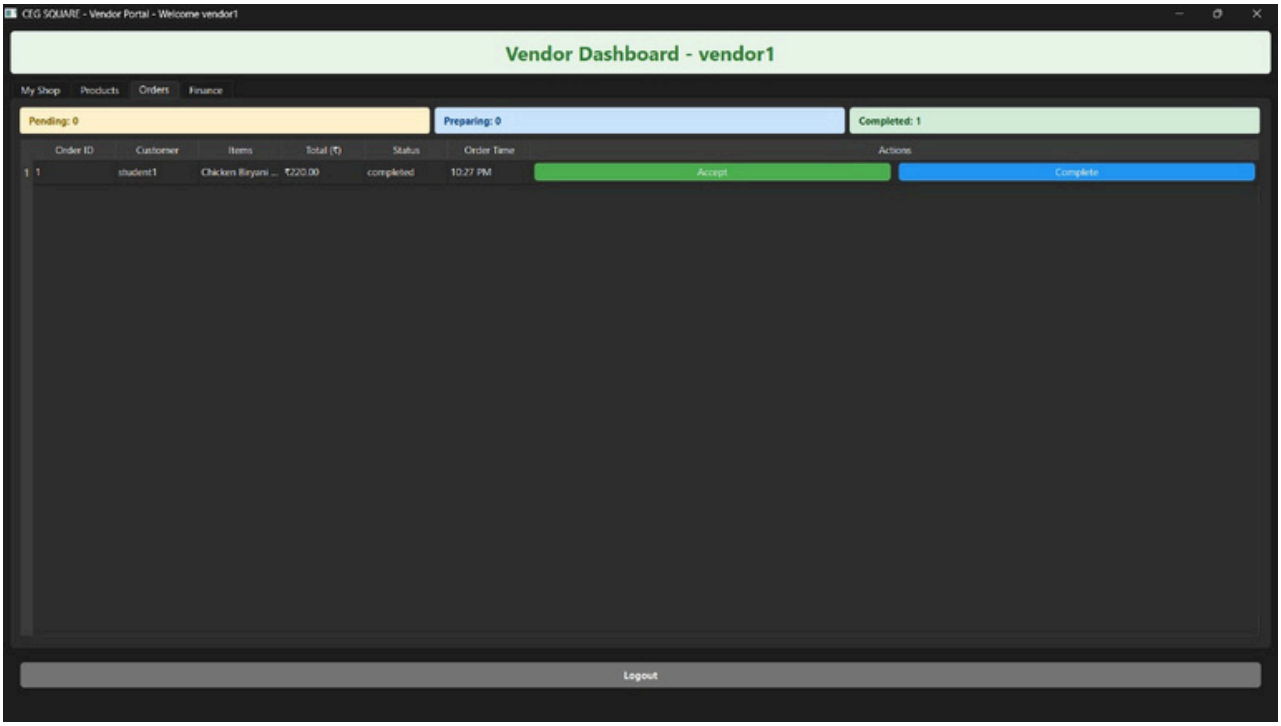
d).Vendor Shop Registration

Illustrates the form where vendors register their shop details and business information.



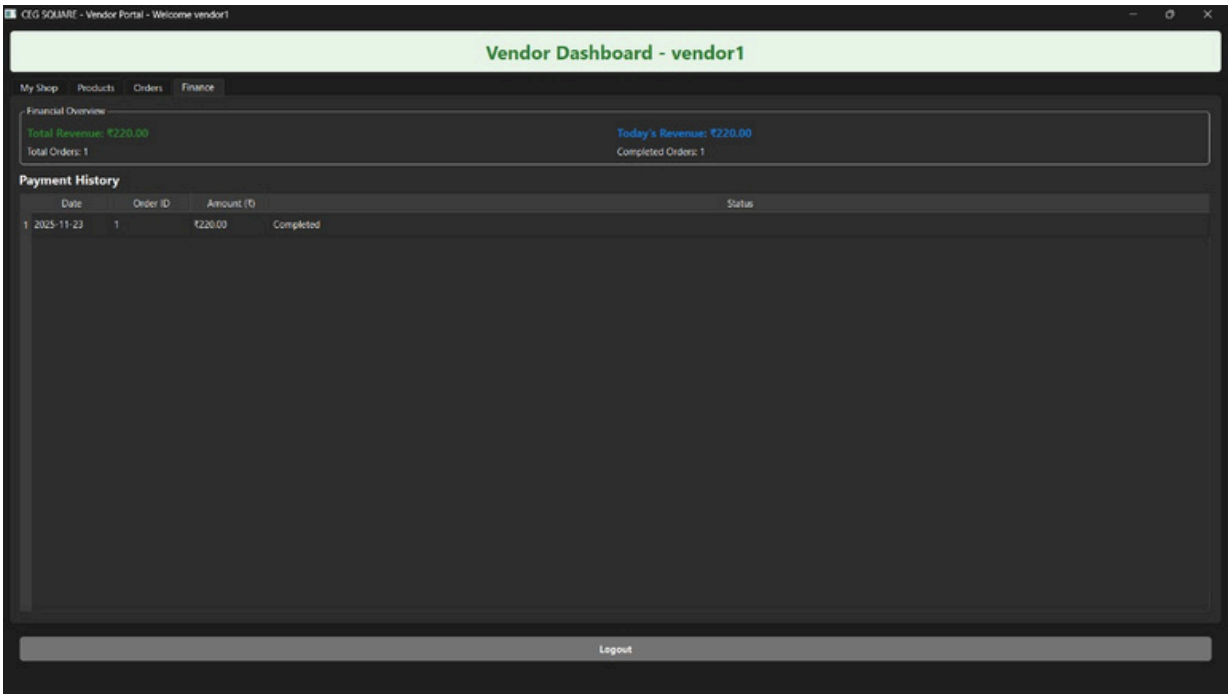
e).Vendor Product Management

Shows the vendor's product catalog with options to add and manage menu items.



f).Vendor Order Processing

Depicts the order queue where vendors can view and update order statuses.



g).Vendor Finance Dashboard

Presents financial analytics including revenue reports and payment history.

8.CONCLUSION

The CEG SQUARE project successfully demonstrates the design and implementation of a full-featured canteen management system. All primary objectives have been met: the system provides a seamless ordering experience for students, a powerful management tool for vendors, and effectively digitizes the core operations of a canteen. The use of C++ and Qt resulted in a performant and responsive application, while the SQLite database reliably handles all data persistence needs. The project stands as a testament to the practical application of software engineering principles, including object-oriented design, database normalization, and UI/UX considerations. Future work could focus on integrating online payment gateways, developing a mobile companion app, and adding push notifications for order updates.

9. REFERENCES

- **Qt Documentation. https://doc.qt.io/**
- **SQLite Documentation. https://www.sqlite.org/docs.html**
- **Stroustrup, B. (2013). The C++ Programming Language (4th Edition). Addison-Wesley Professional.**
- **Blanchette, J., & Summerfield, M. (2008). C++ GUI Programming with Qt 4 (2nd Edition). Prentice Hall.**
- **AI ASSISTANCE : https://chat.deepseek.com/**

THANK YOU...