CSE302: Database Systems (Section No.06) [Summer 25]

Project Report

EWU Cafeteria and Payment System

https://github.com/Jafreen-ewu/EWU-Cafeteria.git
https://github.com/FatemaNourNishi/EWU_CafeteriaProject
https://github.com/sanjidajaman847/ewu-cafeteria.git

Submitted by:

Student ID	Student Name	Contribution Percentage
2021-1-60-018	Mst. Jafreen Jahan	34%
2023-1-60-256	Fatema Akter Nishi	33%
2023-1-60-162	Sanjida Jaman	33%

1.Project Description

The EWU Cafeteria and Payment system is a computer-based project tailored to East West University with the aim of enhancing the management of the cafeteria and simplification of services offered to students and personnel. The current system in most of the university cafeterias such as that of EWU is manual. Students tend to wait in long queues to take food orders, to pay with cash alone and cafeteria staff manually books all their transactions. Although this conventional method is effective, it tends to pose a number of challenges that comprise of lengthy queues, inaccurate or absent orders, confusion when it comes to managing payments and lack of organization on stocking of food. Such issues reduce the efficiency and convenience of the cafeteria service to both the students and the employees.
In order to address these problems, our project proposes implementation of a web-based cafeteria management system which is directly linked to a centralized database. This system offers a number of useful functions. Students are able to access the menu online see the food items they have and order them at their own convenience. Payments are also facilitated by the system both in cash and digital wallet and make the process flexible and modern. The database keeps all the orders and payment information safely, and therefore, minimizes human error and guarantees appropriate record-keeping. Moreover, the system also automatically restocks the stock of food items without any confusion over the availability of items after each order.

This project is simple in design but very impactful in solving real-life cafeteria problems. It shows how database systems may be utilized not just in data storage, but also in the day-to-day work. With the help of this system, the cafeteria can be faster and more reliable and organized. Also, the given project opens up new opportunities to improve it with the addition of a login and registration system, an administration panel of the cafeteria workers, the inclusion of mobile payment options, including bKash or Nagad, and the creation of detailed sales statistics. Overall, the EWU Cafeteria & Payment System is an excellent example of how technology and databases can improve everyday university life.

2. Key Features

The system has several important features. Each one is explained below in simple words:

1.Show Menu

- The home page shows all food items available in the cafeteria.
- It also shows the price and how many items are in stock.
- Example: Chicken Biryani – 150 Taka – Stock: 45

2.Place Order

- A student can choose an item, select quantity, and place an order.
- The system saves the order in the database.

3.Wallet Payment

- Each student has a wallet account.

- When the student pays from the wallet, the system checks the balance.
- If enough money is there, the balance goes down.

4.Cash Payment

- If a student wants, they can still pay with cash.
- The order is saved as "Cash payment".

5.Stock Update

- When someone buys food, the stock decreases automatically.
- Example: Stock of Coffee is 100 → after buying 2 cups, it becomes 98.

6.Wallet Top-Up

- Students can add money to their wallet balance.
- In the demo, we just use a simple form. In real life, it can connect to bKash/Nagad.

7.Payment Records

- Every payment is saved in the database.
- Later, both students and staff can see these records.

8.Future Admin Role

- In the future, an Admin can manage menu items (add, update, delete).
- Admin can also check sales reports.

3. Database Design

3.1 E-R Model

Entities:
- User (User_ID, Name, Student_ID, Email, Phone, Role)
- Wallet (Wallet_ID, Balance, FK: User_ID)
- Menu (Item_ID, Item_Name, Price, Stock)
- Order (Order_ID, User_ID, Total_Amount, Payment_Method, Order_Date)
- Order_Item (Order_Item_ID, Order_ID, Item_ID, Quantity, Price)
- Payment (Payment_ID, Order_ID, Amount, Status, Payment_Date)

Relationships:
- User ↔ Wallet (1:1)
- User ↔ Order (1:N)
- Order ↔ Order_Item (1:N)
- Order ↔ Payment (1:1)
- Menu ↔ Order_Item (1:N)

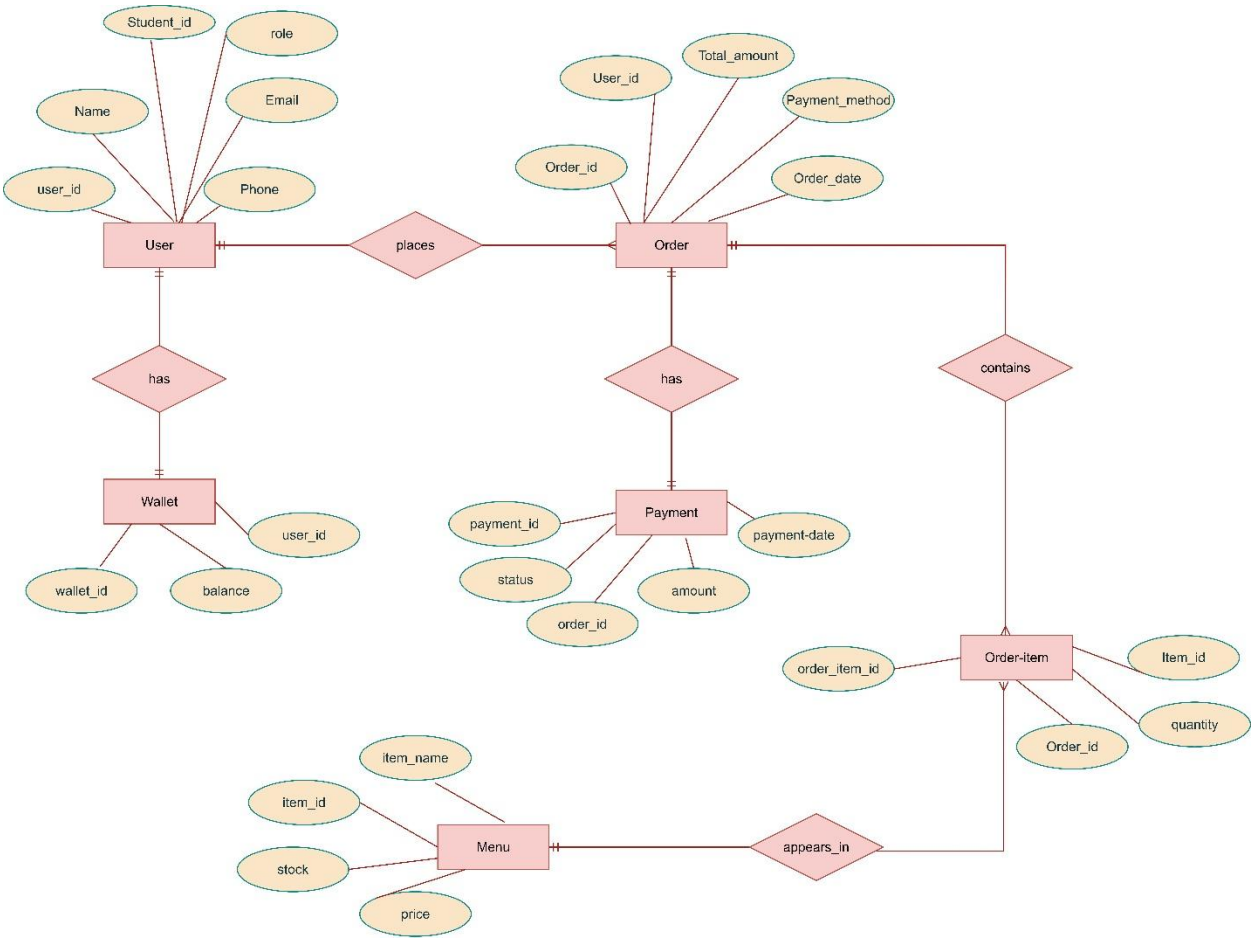

Fig:1 ER-diagram

3.2 Relational Data Model

- Users(User_ID PK, Student_ID, Name, Email, Phone, Role)
- Wallet(Wallet_ID PK, User_ID FK, Balance)
- Menu(Item_ID PK, Item_Name, Price, Stock)
- Orders(Order_ID PK, User_ID FK, Total_Amount, Payment_Method, Order_Date)
- Order_Items(Order_Item_ID PK, Order_ID FK, Item_ID FK, Quantity, Price)
- Payments(Payment_ID PK, Order_ID FK, Amount, Status, Payment_Date)

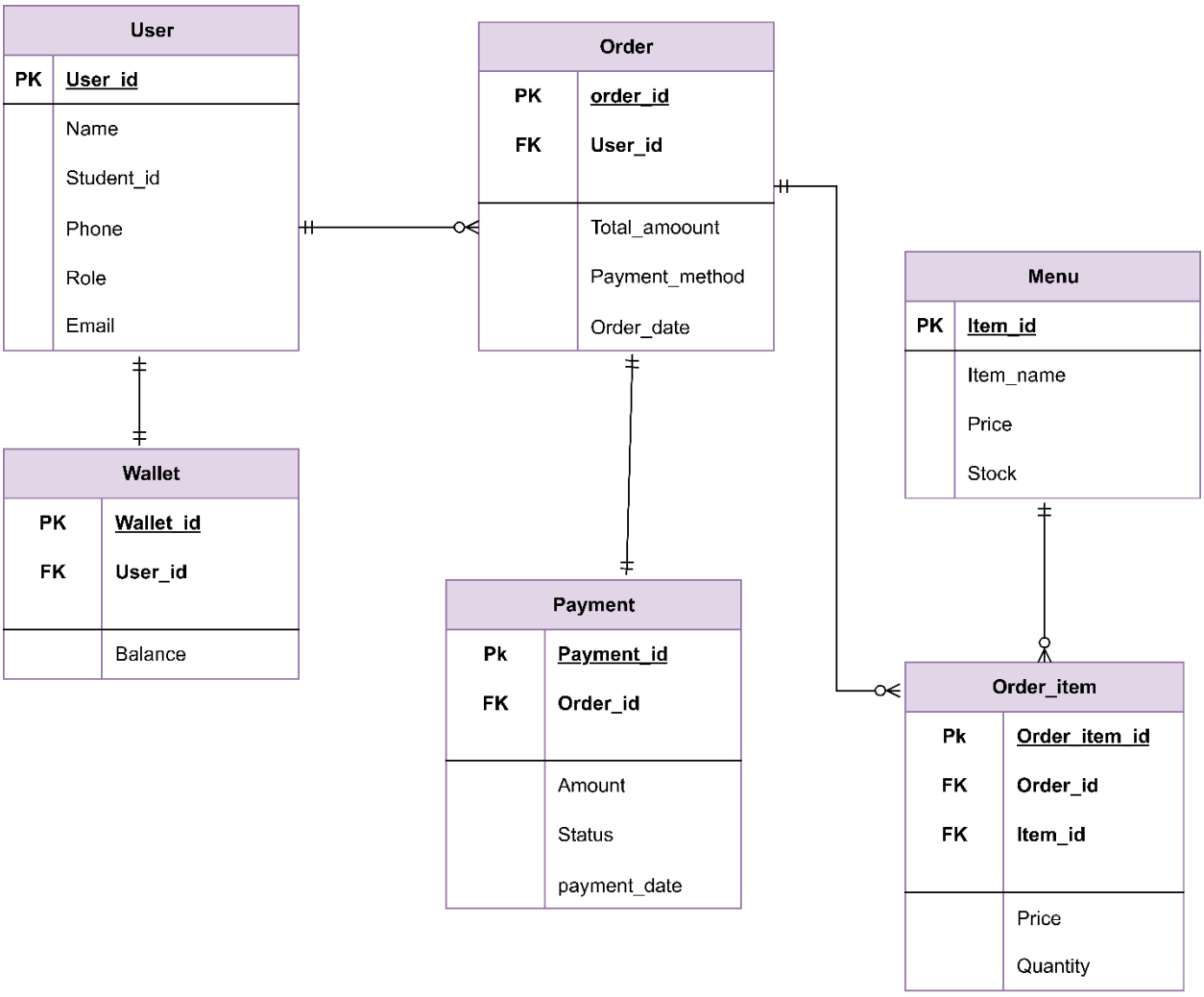

Fig:2 Relational Model

4. Tools & Technologies Used

- **MySQL** → Used for database management. It supports referential integrity, indexing, and foreign key constraints.
- **PHP** → Used for backend logic, connecting to the database, and processing transactions.
- **HTML & CSS** → Used for the frontend GUI (menu display, wallet page, payment history).
- **Apache Server (XAMPP)** → Provides the local server environment where PHP and MySQL run together.
- **phpMyAdmin** → Simplifies MySQL management with a web-based interface.
- **Draw.io** → Used for drawing ER and schema diagrams.
- **VS Code** → Used as the main code editor

5. Role Assignment

- **User (Student/Staff):**

- Can see menu.
- Can order food.
- Can pay by wallet or cash.
- Can check wallet balance.
- Can check payment history.

- **Admin (Future Work):**
  - Can add/edit/remove food items.
  - Can check all sales records.
  - Can control stock.

6. GUI Screenshot

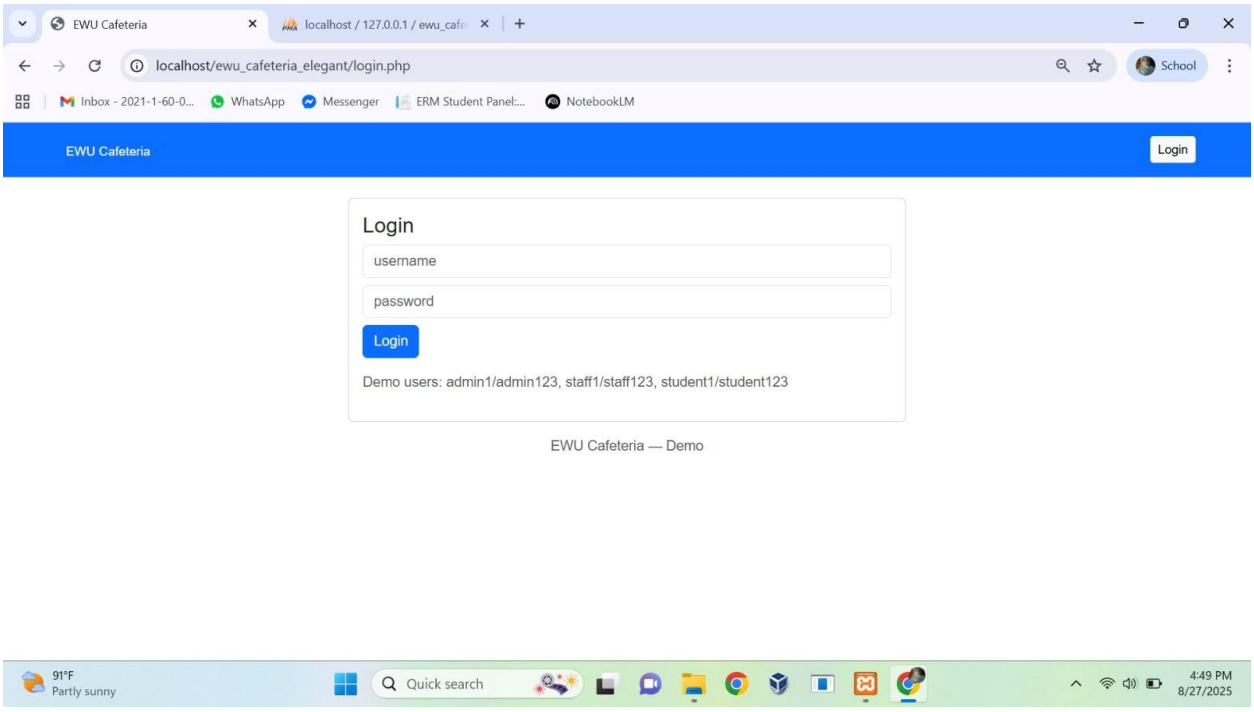

Fig:1 Login Page

The entry point to the system where users enter their username and password. It lists demo accounts for testing different roles (admin, staff, student).

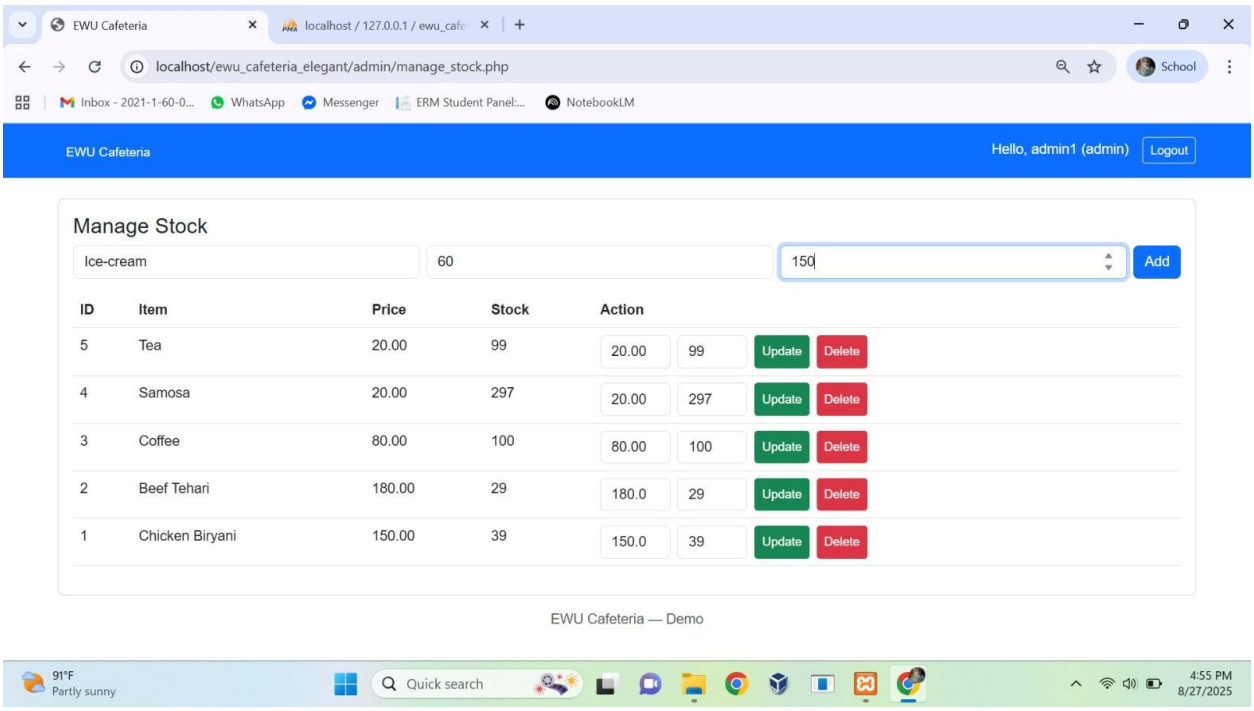

Fig:2 Manage Stock Page

This is the main interface for an administrator to control the menu inventory. It lists all available items (like Chicken Biryani, Coffee, Tea) with their ID, Price, and current Stock level. For each item, there are "Update" and "Delete" buttons to modify or remove them. The top section with "Name | Price | Stock | Add" is a form to input new items into the system. This appears to be a confirmation screen shown immediately after a new item (in this case, "Ice-cream") has been successfully added to the stock list. It displays the updated list, confirming the new item is now

included.

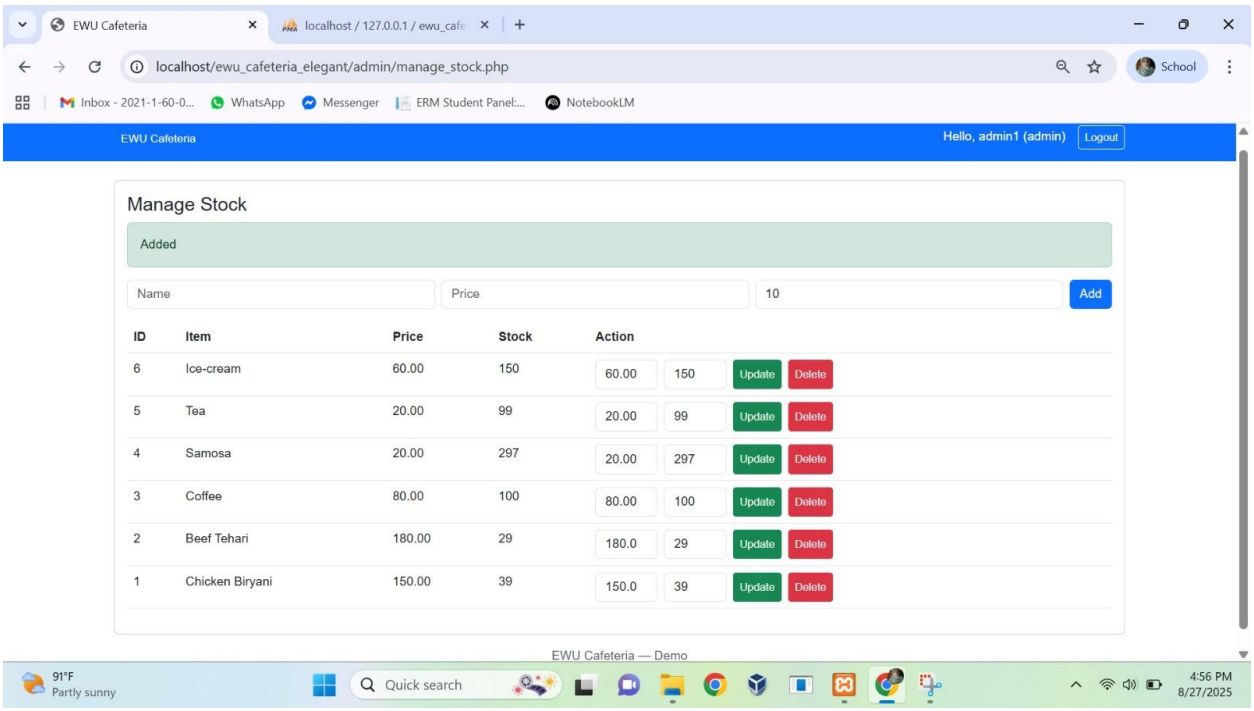

Fig:3 Added" Confirmation Page

This appears to be a confirmation screen shown immediately after a new item (in this case, "Ice-cream") has been successfully added to the stock list. It displays the updated list, confirming the new item is now included.

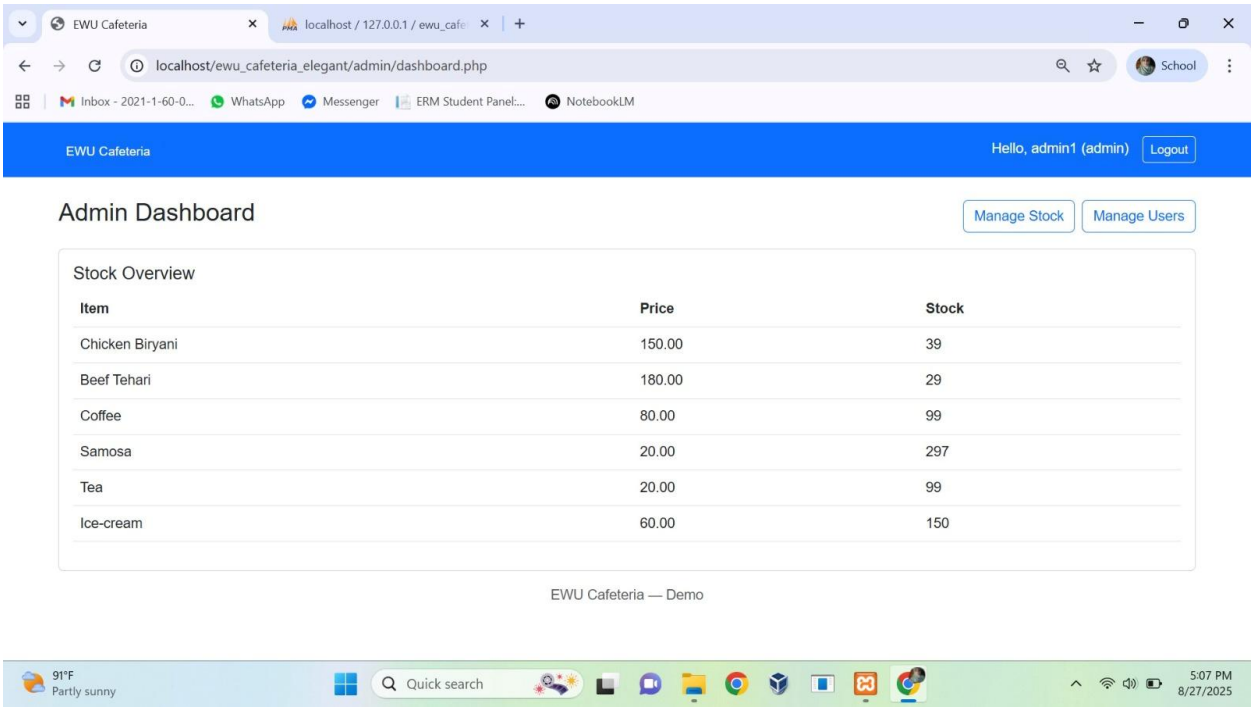

Fig:4 Admin Dashboard - Stock Overview

This is a dashboard view(As shown in fig:4), likely the home page for an admin user. It provides a quick, at-a-glance summary of the current inventory, showing each item alongside its price and stock count. It's a simplified view compared to the full "Manage Stock" page.

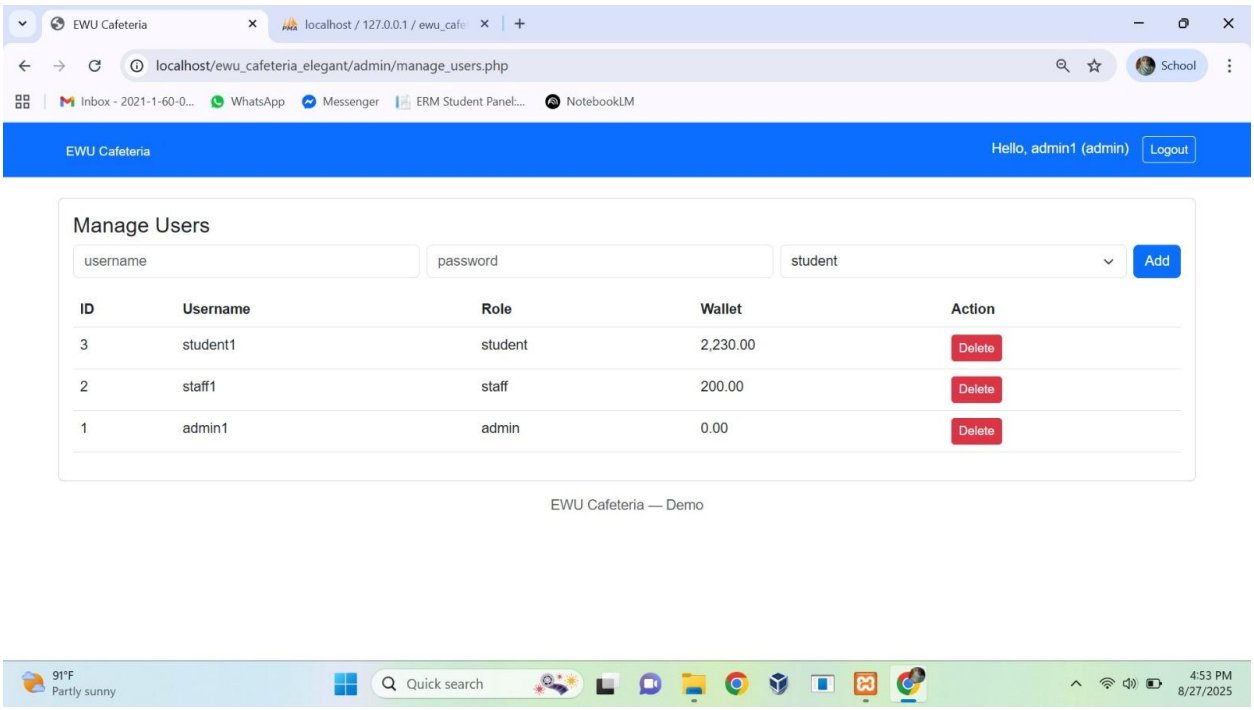

Fig:5 Manage Users Page

This interface allows an administrator to manage user accounts for the system. It lists all users (e.g., admin1, staff1, student1), their roles, and their current wallet balance. The "Delete" action allows for removing users. The form at the top is for adding new users.

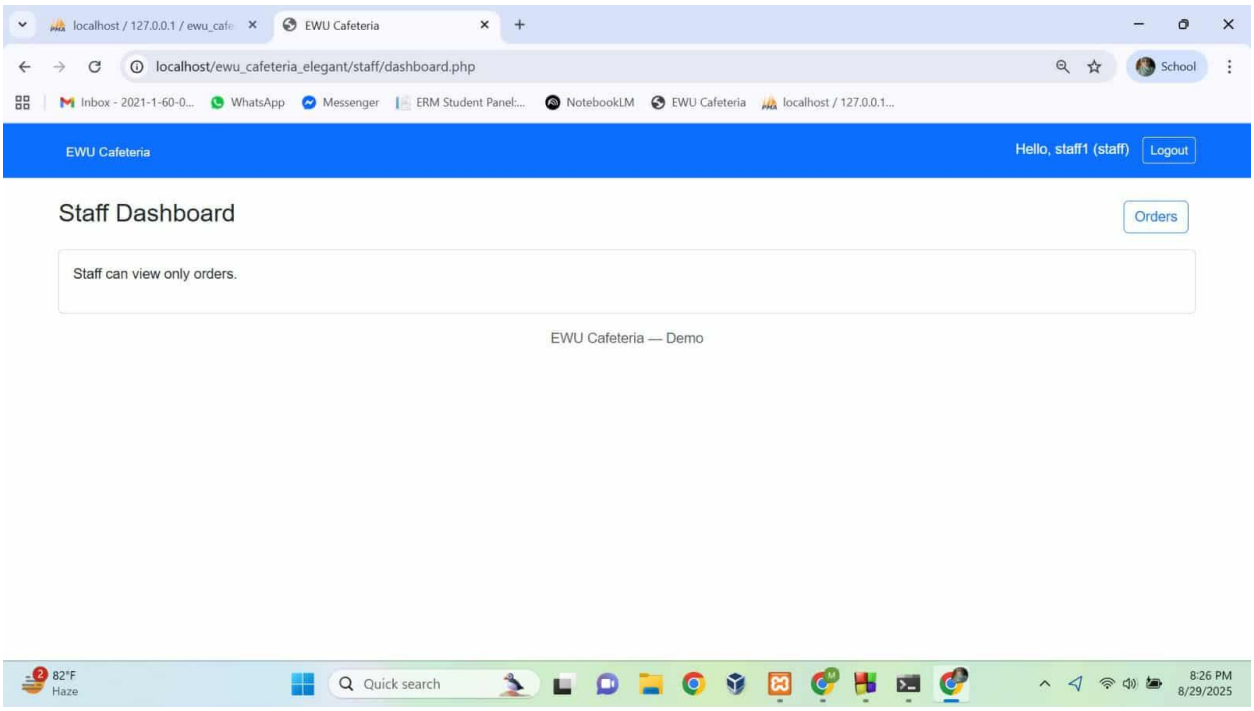

Fig:6 Staff Dashboard

The home screen for a user with a "staff" role. Its purpose is described as allowing staff to view orders.

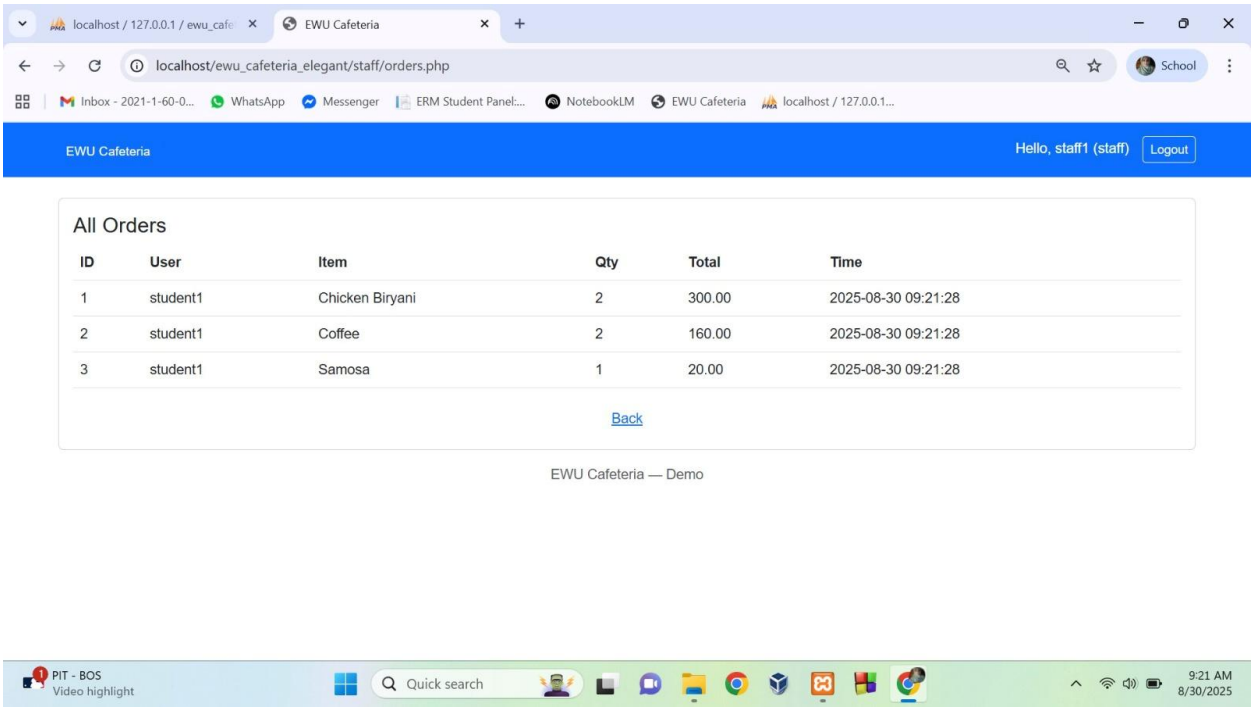

Fig:7 All Orders Page

Every order placed in the system. It shows the order ID, user who placed it, the item, quantity, total cost, and the exact timestamp. This view is likely for admins or staff to monitor activity.

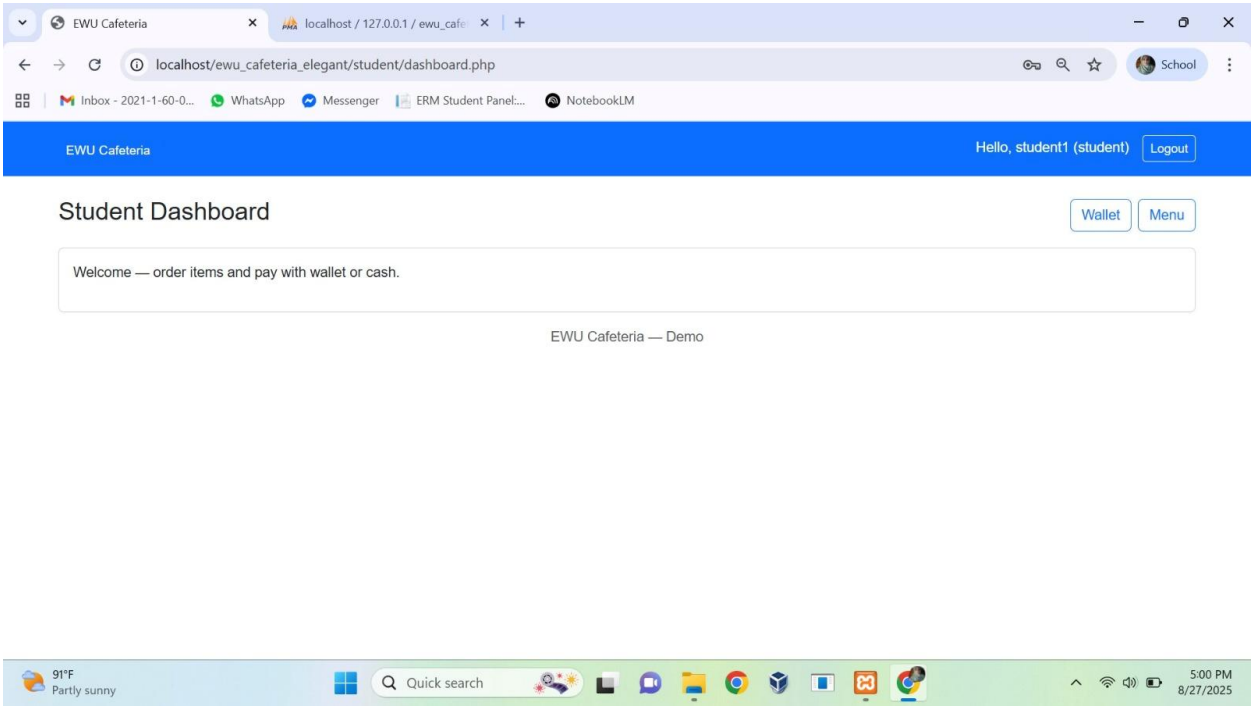

Fig: 8 Student Dashboard

The home screen for a user with a "student" role. It welcomes the student and explains that they can order items and pay using their wallet balance or cash.

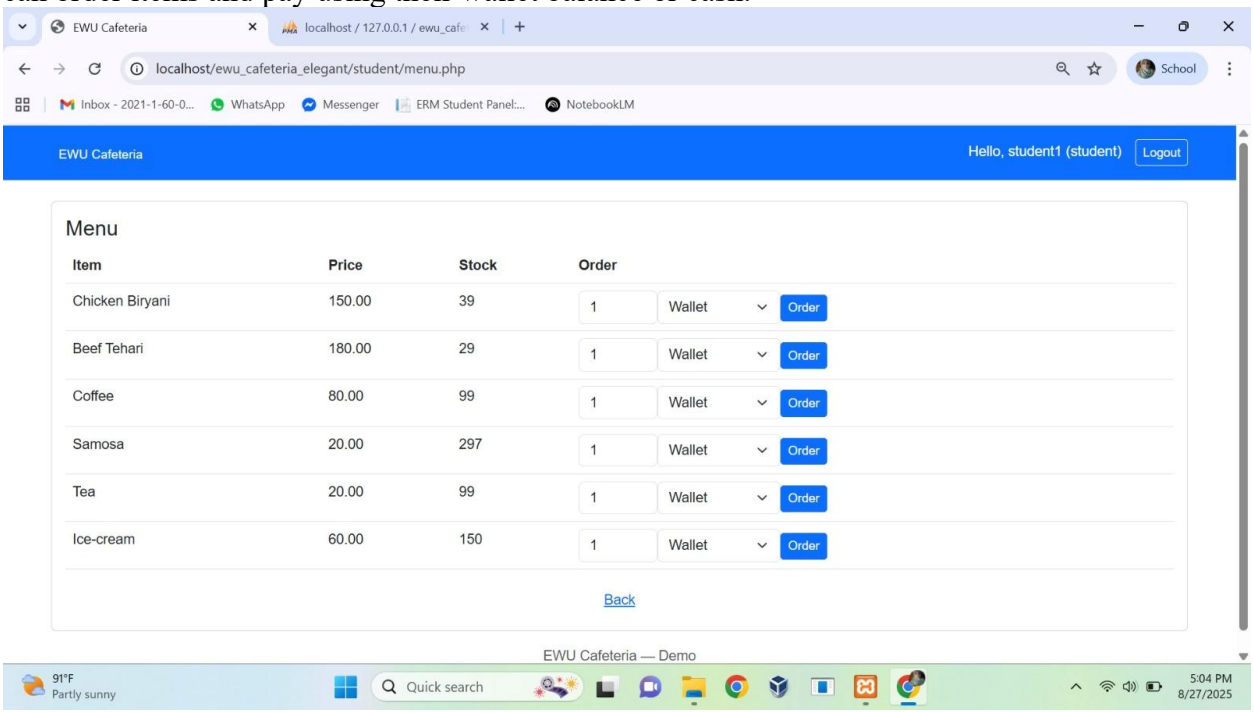

Fig:9 Student Menu

The main menu for a student to place orders. It lists all available food items, their price, current stock, and an "Order" button (with a quantity selector, defaulted to 1) for each item.

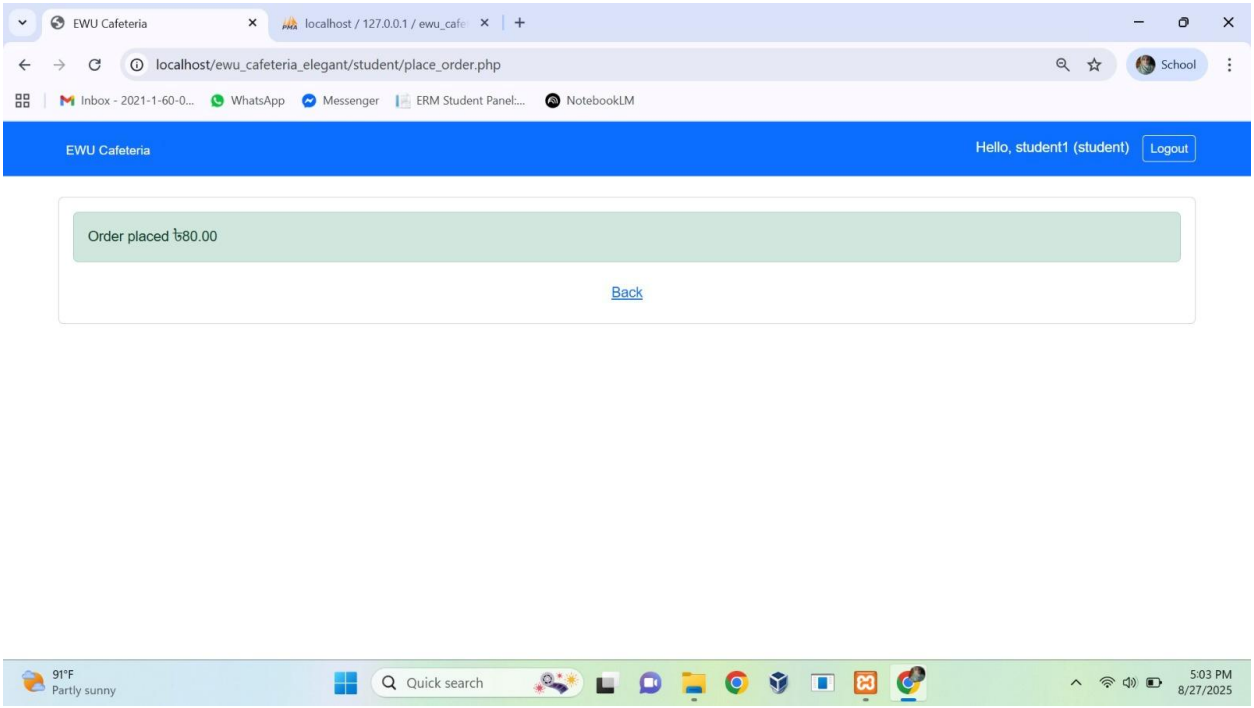

Fig:10 Order Confirmation Page

Fig:11 shown a confirmation screen. A student after successfully placing an order. It displays the message "Order placed" and the total amount charged (80.00).

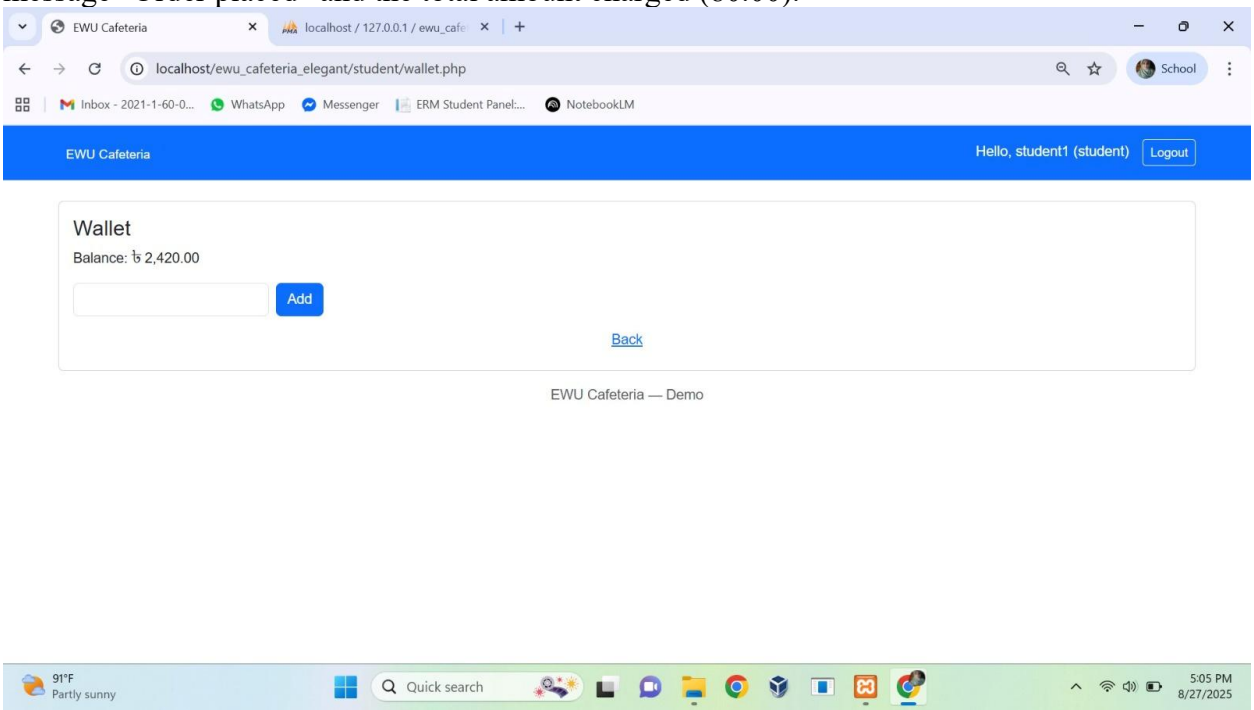

Fig:11 Student Wallet Page

This is the student's wallet view. It shows their current balance ($2,420.00 in one, $2,230.00 in the other) and has a button to "Add" funds. This is likely where a staff member would top up the student's balance.

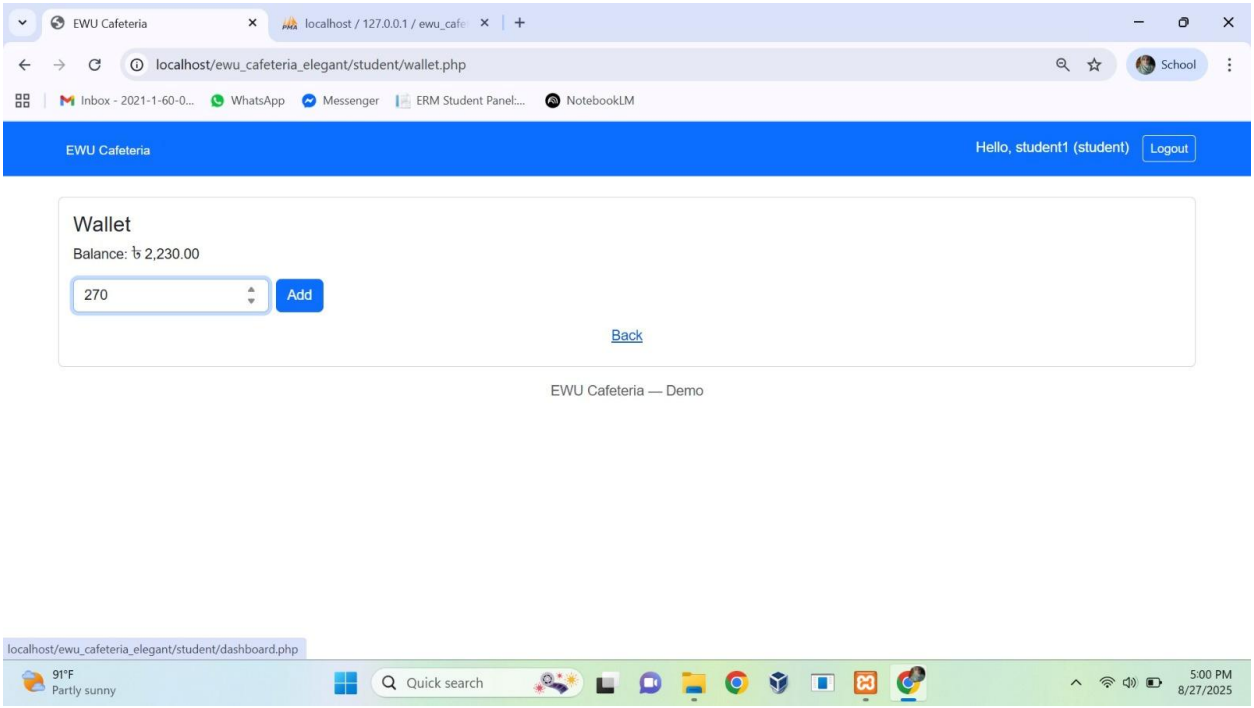

Fig:12 Student Wallet Page

This image shows a student's wallet page within the EWU Cafeteria management system. The interface displays the current balance of $2,230.00 and provides a functionality to add funds, as indicated by the pre-filled amount of "270" and an "Add" button. This allows the student to top up their account balance for making purchases. A "Back" button is also present for returning to the previous dashboard page.

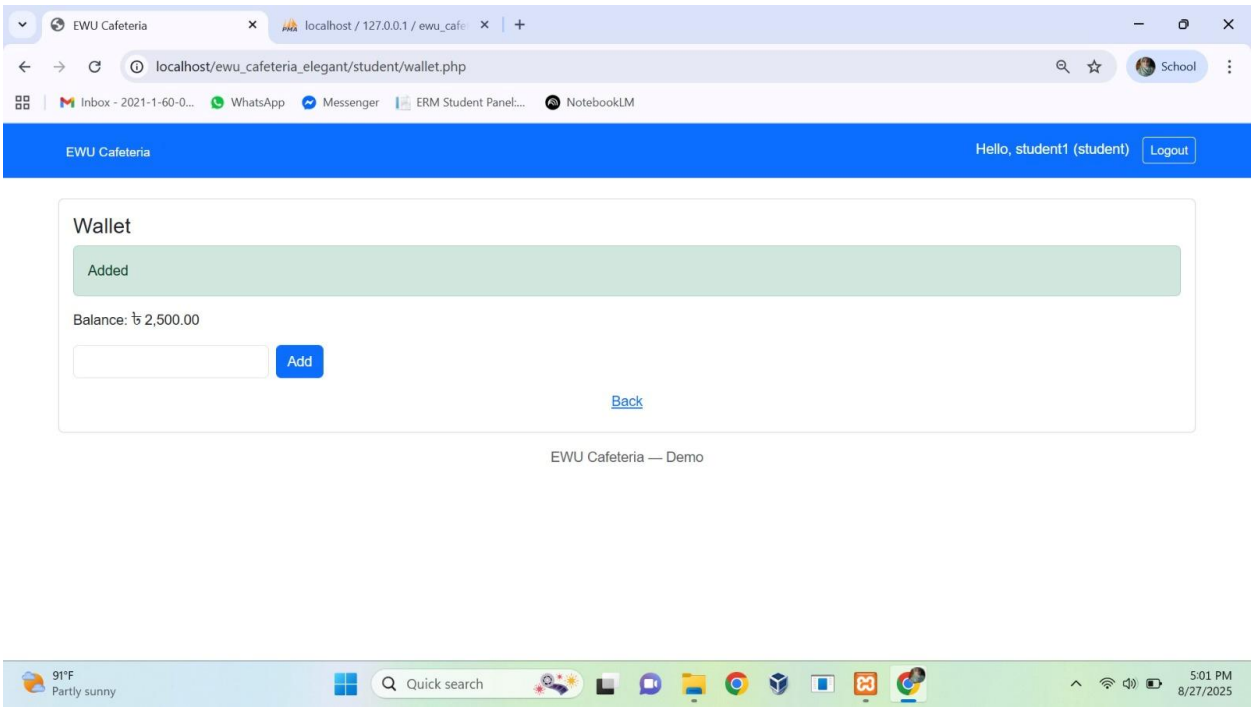

Fig:13 Student Wallet page

A confirmation message is displayed once the amount is added successfully. As shown in the fig:14 the balance is now ৳ 2,500.00 after adding money. Additionally, there is a Back link to return to the previous page.

7. Conclusion

This project helped us understand how a database system works in real life. It was not just a classroom exercise but also a practical solution to cafeteria problems. Through this project, we learned how to make ER diagrams, convert them into database tables, and write SQL queries for create, read, update, and delete (CRUD) operations. We also gained experience in connecting PHP code with a MySQL database. However, we faced some challenges such as learning how to set up XAMPP and phpMyAdmin, fixing SQL errors in PHP code, ensuring that the wallet balance does not go below zero, and handling stock updates after each order. In the future, we plan to improve the system by adding a login and registration feature, creating an admin panel for cafeteria staff, integrating online payment apps like bKash or Nagad, and generating daily and monthly sales reports.

8. References

- MySQL Docs: https://dev.mysql.com/doc/
- PHP Docs: https://www.php.net/manual/en/
- XAMPP: https://www.apachefriends.org/
- Draw.io for diagrams