## 

**CSE- 482**

**Internet and Web Technologies**

**Project Report on**

Online Canteen Order & Queue System

**Submitted to**

Faculty- Md. Naqib Imtiaz Hussain

Lab Instructor- Shabbir Ahmed

**Submitted by**

Chowdhury Nafis Faiyaz 1931841642

Ayman Ibne Hakim 2013364042

Zahirul Islam Nahid 2013421042

Md. Labib Hasan 2011068642

May 30, 2023

**Contents:**

[**Executive Summary: 3**](#_kfkr1r44wcsl)

[**Introduction 3**](#_cx307ih2860g)

[**Background and Product Context 3**](#_cyg3tyoascec)

[**Technologies used: 4**](#_4uuhxn6ivcvt)

[Frontend technologies: 4](#_t7bcn5nmdc1h)

[Backend technologies: 4](#_jyjagwmzs07l)

[**Implementation plan 5**](#_7fyuwru3v0t3)

[First Checkpoint (week 1): 5](#_1gez2pj8ti9b)

[Requirement Analysis 5](#_91obyqnrheaw)

[Second checkpoint (week 2-3): 5](#_l1j9md90a25)

[Overall Front end structure 5](#_hqhqnlis738d)

[Third checkpoint (week 4-7) 5](#_45lj4hbzca3t)

[Database table formation 5](#_f3js4wb9iqlx)

[Backend- connecting front end with database 5](#_s5wydxggun1w)

[Fourth checkpoint (week 8-10) 6](#_pgw2k4or4w4e)

[Implementation of ajax 6](#_njmt6edgx5cj)

[**Gantt Chart: 6**](#_u21qhm5l65on)

[**Testing Protocols: 7**](#_3gvyws7mu9r4)

[**Use Case and User Story 8**](#_6eme9jbn16nx)

[Use Case 1: User Login 8](#_gup8hb7m3c4q)

[Use Case 2: Order Food 8](#_e96qd9fv175u)

[Use Case 3: View Pending Order 9](#_6ktenl48focp)

[Use Case 4: Receive Payment 9](#_5p00idh4b4a1)

[Use Case 5: Call Next Queue 10](#_ilrvwrehop5r)

[Use Case 6: Refund Order 10](#_ljx4b4aivelj)

[Use Case 7: Place order for customer 11](#_f6dgdr4q20t4)

[Use Case 8: Create Account 11](#_xt3x1ssckfo1)

[Use Case 9: Delete account 12](#_g5pq7b6blf4u)

[Use Case 10: Get sales info 12](#_gynybzhpnvsz)

[Use Case 11: Update Menu 13](#_gxbnwf6h3o9b)

[**Exceptional Flow: 13**](#_c095a1log24s)

[Conclusion: 13](#_56lzano5t92o)

## 

## **Executive Summary:**

This project proposal outlines the development of a universal digital portal designed to automate the food ordering process in the canteen. The primary objective is to introduce a user-friendly digital platform that enables students to conveniently order and pay for their food, eliminating the need for cash transactions and reducing waiting times through an efficient queue management system. This report presents the project's objectives, features, and implementation plan to address the current challenges faced by canteens in managing food orders and payments, and the admin privileges which our system offers.

## **Introduction**

The canteen food ordering system plays a critical role in catering to the dietary needs of students while fostering a streamlined and efficient process. However, the traditional manual system often leads to long queues, cash handling issues, and increased waiting times. This project proposes the development of a universal digital portal that revolutionizes the canteen experience by automating the ordering process, introducing a secure digital payment system, and implementing a queue management system.

## **Background and Product Context**

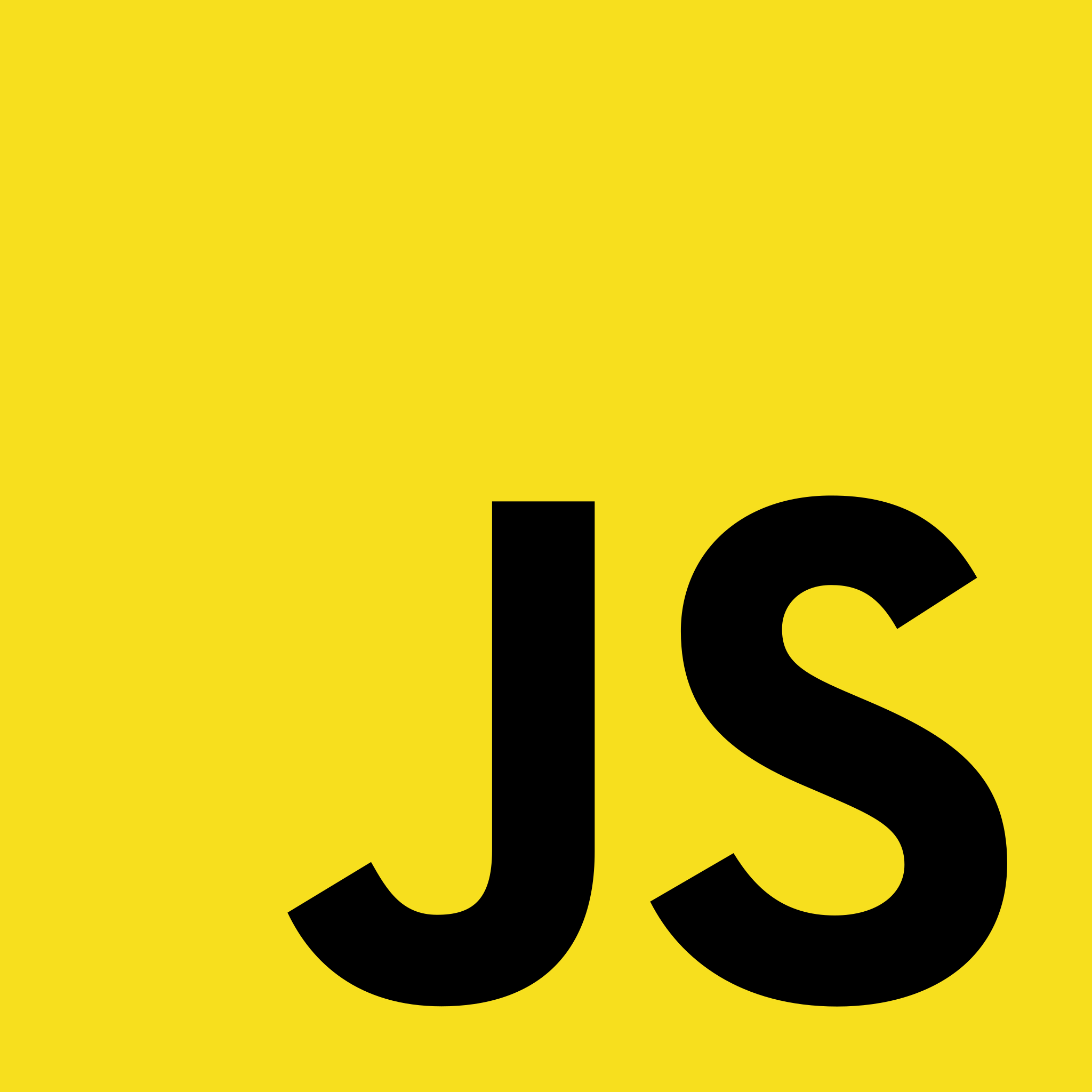
The canteen at NSU faces challenges such as long queues, frustrated customers, limited seating, and the unavailability of change, causing inconvenience and delays for students. To address these issues, a digital solution is proposed to automate the food ordering process and improve the overall customer experience.

Product Overview:

The proposed solution is a web application designed to streamline food ordering in the canteen. The application will display the available food items on both user and canteen screens, allowing students to select their desired food and make online payments securely through a payment gateway. Upon successful payment, users will receive a digital receipt with a unique queue number. Canteen staff will serve the food based on the queue number, ensuring efficient order delivery. The system will lock the queue number upon submission, preventing further processing.

By implementing this digital portal, the canteen aims to eliminate long queues, minimize waiting times, improve customer satisfaction, and facilitate cashless transactions, ultimately enhancing the overall dining experience for students.

## **Technologies used:**

### 

### **Frontend technologies:**

For the front-end development of the digital portal, a combination of HTML, Tailwind CSS, and JavaScript will be utilized. HTML (Hypertext Markup Language) will be used as the foundation for structuring the web pages, defining the content, and organizing the elements. Tailwind CSS, a highly customizable CSS framework, will be employed for efficient and rapid styling of the user interface, ensuring a visually appealing and responsive design. JavaScript will provide the necessary interactivity and functionality, enabling dynamic elements, event handling, form validation, and communication with the backend. Together, these technologies will empower the development team to create a seamless and intuitive user experience for students accessing the digital portal.

### **Backend technologies:**

The backend of the digital portal will be developed using PHP for server-side scripting and MySQL as the database management system. PHP will handle user requests, process data, and generate dynamic web pages, while MySQL will be used to store and manage the data related to menu items, user profiles, orders, and transactions. This combination of PHP and MySQL provides a robust and efficient backend solution, ensuring secure data storage, seamless integration with the front-end, and smooth operation of the digital portal.

## **Implementation plan**

### First Checkpoint (week 1):

##### Requirement Analysis

### Second checkpoint (week 2-3):

##### Overall Front end structure

1. Landing Page design
2. Login design
3. Super Admin Dashboard
4. Super Admin CRUD Table & Form Design
5. Staff Dashboard for Taking Money Payment, Receipt Print Point
6. Staff Dashboard for Food Server Point
7. Customer Login
8. Customer Dashboard
9. Queue Display Page

### Third checkpoint (week 4-7)

##### Database table formation

1. Super Admin Table
2. Customer Table
3. User Table
4. Payment Table
5. Food List Table
6. Que Table
7. Balance Table

##### Backend- connecting front end with database

1. Connecting Database
2. User Login Session Setup
3. Admin CRUD with database
4. Staff and Customers CRUD with database
5. Live Food Search
6. Instantiation of frontend interface as required.

### Fourth checkpoint (week 8-10)

##### Implementation of ajax

1. Live Queue Fetch
2. Integration of “Stripe”- Online payment Gateway API
3. Implementation of service worker
4. Implementation of lazy loading using javascript libraries
5. Testing responsiveness of the frontend UI
6. Debugging

## **Gantt Chart:**

|  | | **Week** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Checkpoint** | **Deliverable** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | **Requirement analysis** |  |  |  |  |  |  |  |  |  |  |
| 2 | **Front end UI development** |  |  |  |  |  |  |  |  |  |  |
| 3 | **Backend Functionalities and database development** |  |  |  |  |  |  |  |  |  |  |
| 4 | **Live search & queue display, Service worker, lazy loading, payment API integration** |  |  |  |  |  |  |  |  |  |  |

### 

## 

## **Testing Protocols:**

## 

## **Use Case and User Story**

| Use Case 1: User Login |
| --- |
| Actor: Customer, Food Server, Cashier, Super Admin  Scenario:  1. Input user ID and password.  2. Check user ID and password (User Authentication).  Exception:  1. Required page not found.  2. No database connection.  **Precondition:**  1.URL of the login page from the device.  **Post Condition:**   1. Successful login- Logging in to the system. 2. Unsuccessful login- Stay on the same page showing an error message. |

As a customer, food server, cashier, or super admin, I should be able to log in to my account.

\*Customer refers to students, faculties, and other staff working under the university.

| Use Case 2: Order Food |
| --- |
| **Actor:** Customer  **Scenario:**  1. Select available food from the menu.  2. Select payment method.  **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. Customers signed in. 2. URL of the order food page. 3. Customer has selected at least 1 item.   **Post Condition:**   1. Successful order- Send order confirmation message. 2. Unsuccessful order- Stay on the same page showing an error message. |

As a customer, I should be able to order food.

| Use Case 3: View Pending Order |
| --- |
| **Actor:** Food Server  **Scenario:**  1. View the list of orders pending.  **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. Food Server signed in. 2. URL of the pending order page.   **Post Condition:**   1. Show list of orders pending. |

As a food server, I should be able to view pending orders.

| Use Case 4: Receive Payment |
| --- |
| **Actor:** Food Server, Cashier  **Scenario:**  1. Select the pending payment.  2. Clear payment after receiving it.  **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. Food Server or cashier signed in. 2. URL of the accept payment page. 3. There is at least 1 payment due.   **Post Condition:**   1. Payment successful- Send a message to customer and food server/ cashier stating payment successful.\ 2. Payment Unsuccessful- Stay on the same page and show an error message. |

As a food server/ cashier, I should be able to receive payment.

| Use Case 5: Call Next Queue |
| --- |
| **Actor:** Food Server  **Scenario:**  1. Call the next customer.  **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. Payment for the specific order has been cleared.   **Post Condition:**   1. Stay on the same page and show the next order details. |

As a food server, I should be able to call the next queue.

| Use Case 6: Refund Order |
| --- |
| **Actor:** Food Server  **Scenario:**   1. Select the order to be refunded. 2. Make payment to the customer.   **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. There is at least 1 pending order. 2. Customer had already paid for the order.   **Post Condition:**   1. Payment successful- Send a message to customer and food server/ cashier stating payment successful. 2. Payment Unsuccessful- Stay on the same page and show an error message. |

As a food server, I should be able to refund orders.

| Use Case 7: Place order for customer |
| --- |
| **Actor:** Cashier  **Scenario:**   1. Select the order. 2. Generate receipt. 3. Process payment   **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. Cashier is logged in to the system. 2. Customer has ordered and paid.   **Post Condition:**   1. Payment successful- Send a message to customer and food server/ cashier stating payment successful. 2. Payment Unsuccessful- Stay on the same page and show an error message. |

As a cashier, I should be able to place orders for customers.

| Use Case 8: Create Account |
| --- |
| **Actor:** Super Admin  **Scenario:**   1. Select "Create Account" option 2. Select user type (Customer, Server, or Cashier) 3. Fill out required user details (name, email, password, etc.) 4. Submit the form to create the new account.   **Precondition:**   1. Super Admin logged in to their account on the system   **Exception:**  1. Required page not found.  2. No database connection.  **Post Condition:**   1. New account has been created and stored into the database. |

As a super admin, I should be able to create accounts for Customers, servers, and cashiers.

| Use Case 9: Delete account |
| --- |
| **Actor:** Super Admin  **Scenario:**   1. Selects "Delete/Block Account" option 2. Searches for the account to be deleted/blocked using their email address or unique ID 3. Confirms the account to be deleted/blocked 4. Submits the form to delete/block the account   **Exception:**  1. Required page not found.  2. No database connection.  **Precondition:**   1. The Super Admin is logged in to their account on the system   **Post Condition:**  The account has been successfully deleted/blocked from the system |

As a super admin, I should be able to delete/ block existing accounts of customers, servers, and cashiers.

| Use Case 10: Get sales info |
| --- |
| **Actor:** Super Admin  **Scenario:**   1. Selects "Sales Information" option 2. Selects a date range to view sales data 3. System generates sales report for the selected date range 4. Super Admin views the sales report   **Precondition:**   1. Super Admin is logged in to their account on the system.   **Exception:**   1. If there are no sales data for the selected date range. 2. If there is an issue with generating the sales report |

As a super admin, I should be able to get sales info.

| Use Case 11: Update Menu |
| --- |
| **Actor:** Super Admin  **Scenario:**   1. Selects "Update Food Menu" option 2. Edits food items or prices as necessary 3. Saves changes to the food menu     **Precondition:**   1. Super Admin logged in to their account on the system.   **Post Condition:**   1. The food menu has been successfully updated and the changes are reflected in the system |

As a super admin, I should be able to update the menu.

Alternate Flow:

* If a customer encounters an issue with their order, they can request a refund or speak to the food server or cashier for assistance. The food server or cashier can refund the order if necessary.
* If the queue display screen or mobile app is unavailable, the food server can call out the order numbers manually.

## Exceptional Flow:

If the system experiences technical issues, the food server or cashier can handle orders manually until the issue is resolved. The super admin can also troubleshoot the system and make necessary updates.

## Conclusion:

The proposed digital portal for food ordering in the NSU canteen aims to improve operational efficiency, reduce waiting times, and enhance the overall dining experience for students. By implementing this solution, the canteen can eliminate manual processes, introduce cashless transactions, and provide a streamlined queue management system, ultimately benefiting both the canteen staff and the student community.