Task 2

1&2

SQL Injection Prevention:

SQL queries are prepared and bound to parameters when querying the database for user data.

Password Security:

The code uses password\_verify to check if the entered password matches the hashed password stored in the database, which is the recommended way to verify passwords securely.

Session Management:

The code starts a session and stores the user's ID in the session after a successful login.

Both files incorporate several security measures to protect against common web security threats.

4.

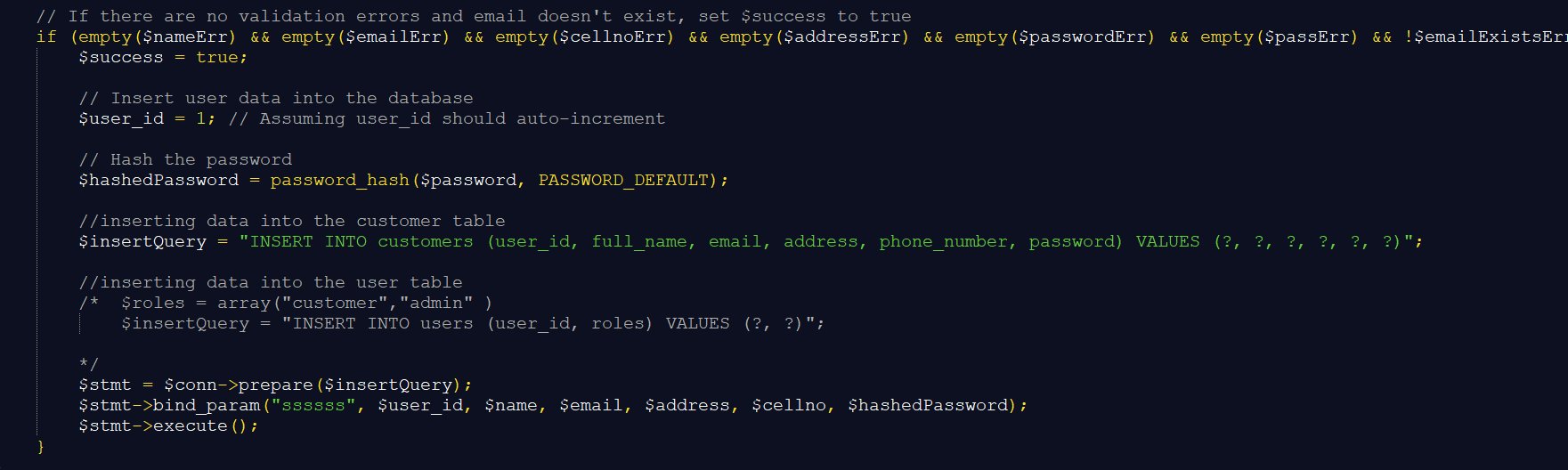
SQL queries are prepared and bound to parameters. The email validation includes a database check to ensure the email doesn't already exist.

We also hashed the passwords before it is sent in the database.

We also manage our sessions for user authentication, ensuring that only authenticated users can access certain features.

Made sure that the data is inserted without revealing it in the URL by using post method for forms.

In the createAccountForm.php file, hashing input from user:



Task 3

*KASI DINER Ordering system*

# Introduction

In response to the challenges faced by patrons in our township restaurants due to long queues and extended waiting times, we propose the implementation of a Online Ordering System. This system aims to provide a convenient and efficient platform for customers to place orders online and streamline the order processing workflow for restaurant administrators.

# System Overview

KasiDiner will comprise two main user roles: customers and administrators. Customers can browse restaurant menus, customize their orders, and place them through the online platform. Administrators will manage and process these orders.

# How to Implement

3.1 Technology Stack

The system will be implemented using a web-based framework, incorporating technologies such as HTML, CSS, JavaScript for the front-end, and PHP for the back-end. A MySQL database will be used for data storage.

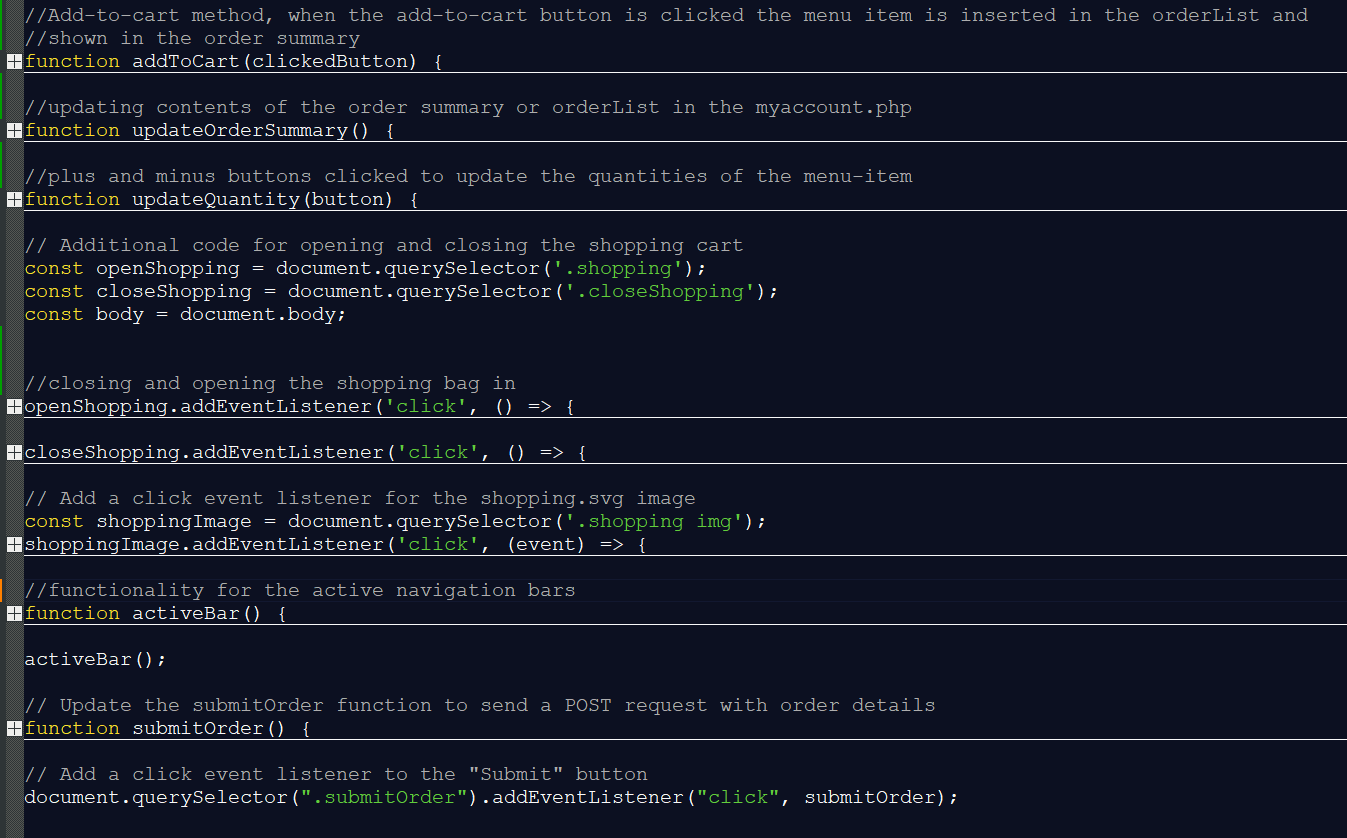
3.2 System Architecture

The architecture will follow a client-server model. The client-side, accessible by customers, will be a userfriendly web interface. The server-side will handle order processing, user authentication, and database interactions.

# How the Features Work

4.1 Order Placement

Customers can access the online menu, select items, and customize orders. The system will provide realtime updates on their order, the order will get submitted to the administrator for processing. Customers can contribute to the system by providing reviews and ratings for restaurants and specific dishes, enhancing the overall user experience.



In the placed\_orders.php makes sure that the orders placed showup in the admin dashboard, so that the admin can processes them.A computer code with many colorful lines

Description automatically generated with medium confidence

4.3 User Accounts

Customers and administrators will have individual accounts for secure logins. Account information will be stored securely, and password hashing will be implemented for enhanced security.

# Security

5.1 User Authentication

Secure login mechanisms will be implemented to ensure that only authorized users can access the system.

5.2 Data Encryption

Sensitive data, especially during transactions, will be encrypted using industry-standard protocols to safeguard user information.

5.3 Authorization

Role-based access controls will be enforced to ensure that users have appropriate permissions within the system.

# User Contribution

Customers can contribute to the system by providing reviews and ratings for restaurants and specific dishes, enhancing the overall user experience.

# Report and Representation

Administrators will have access to reporting features, allowing them to analyze order trends, track popular dishes, and make informed business decisions on what to improve.

**Section A: Overall Functionality and Development**

Order Placement

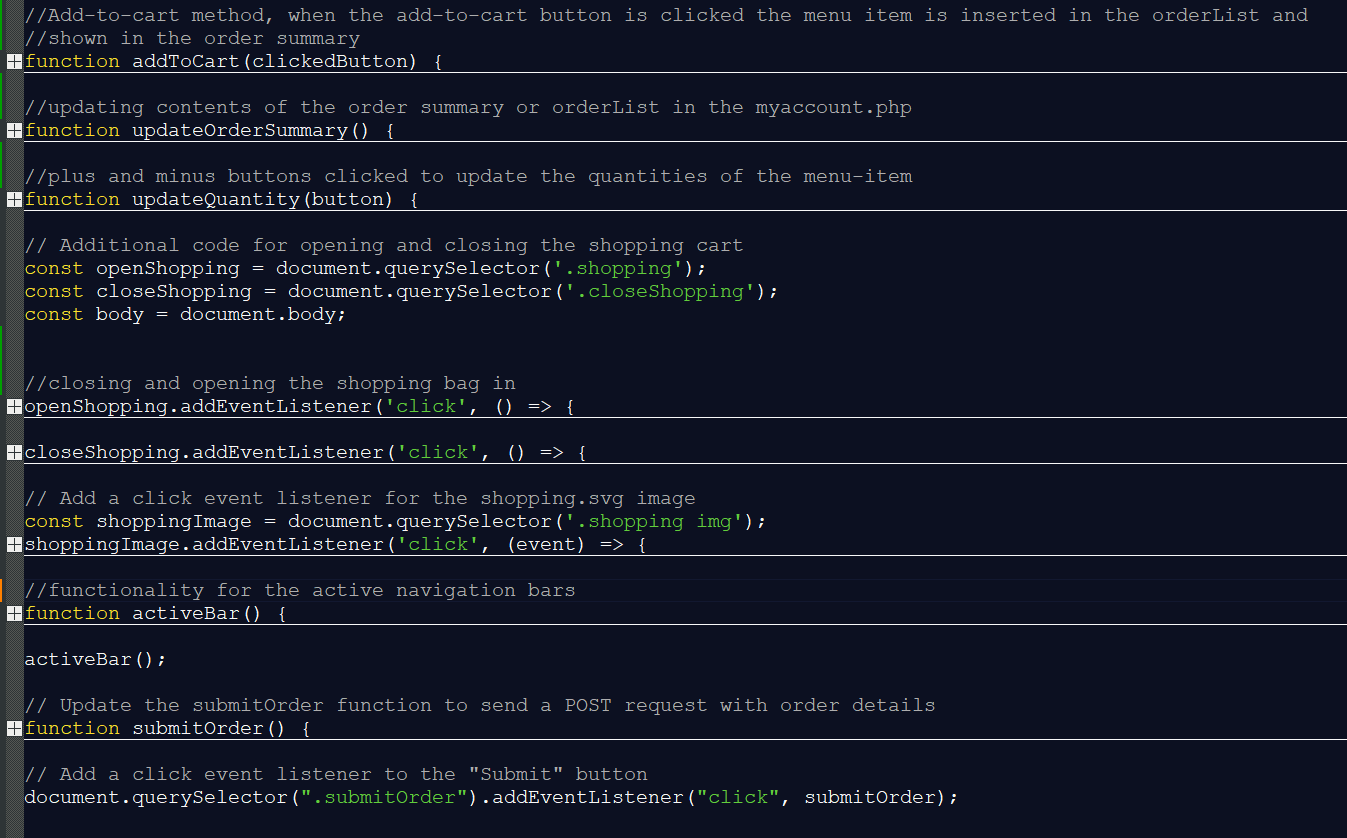
Customers can access the online menu, select items, and customize orders. The system will provide realtime updates on their order, the order will get submitted to the administrator for processing. Customers can contribute to the system by providing reviews and ratings for restaurants and specific dishes, enhancing the overall user experience. These are included in the placed\_order.php, show.php, reviews.php, myaccount.php, myaccount.js and signin.php

This is how the order placement works with its code:

The user clicks the plus/minus buttons to update the quantity of the order item, when the add-to-cart button is clicked the order item is placed in the order summary and order list. When the user clicks the submit button the orderList is sent to the placed\_order.php to the admin

A screenshot of a menu

Description automatically generated



In the placed\_orders.php makes sure that the orders placed showup in the admin dashboard, so that the admin can process them.A computer code with many colorful lines

Description automatically generated with medium confidence

User Accounts

Customers and administrators will have individual accounts for secure logins. Account information will be stored securely, and password hashing will be implemented for enhanced security. This occurs in the signin.php and admin\_login.php. The admin and customer pages used different styles since the admin page is more formal and there is limited access of users than the customer page.

The admin login page:  
A screenshot of a computer

Description automatically generated

The customer login page:

A screenshot of a computer

Description automatically generatedThe admin can also add products to the new products and delete them:  
A screenshot of a computer

Description automatically generated

**Section B: Contributions of Each Team Member**

**Sinenjabulo's Contributions:**

**Security Measures:**

Implemented user authentication mechanisms to ensure secure access to the web application.

Conducted thorough security assessments, identifying, and addressing potential vulnerabilities.

Integrated encryption protocols to protect sensitive user data.

Styling (CSS):

Crafted a visually appealing and responsive user interface using CSS.

Ensured consistent and intuitive design elements throughout the web pages.

Collaborated on the creation of a style guide for a cohesive design language.

Form Validation:

Implemented client-side form validation to enhance user experience and prevent data entry errors.

Validated user inputs on the client side to ensure data integrity before submission.

User Experience (UX):

Contributed to the design of user workflows, focusing on creating a seamless and intuitive experience.

Implemented interactive elements and transitions to improve overall UX.

**Zamokuhle's Contributions:**

**PHP Functionalities:**

Developed server-side functionalities using PHP to process user requests and interactions.

Implemented dynamic content generation based on user inputs and database interactions.

**Database Creation:**

Designed and created the database schema to store user information, messages, or any relevant data.

Implemented relational database concepts for efficient data management.

**HTML Components:**

Structured and coded HTML components to create the foundational elements of web pages.

Ensured proper integration of HTML with PHP for dynamic content rendering.

**Backend Logic:**

Implemented backend logic for data retrieval, manipulation, and presentation to the frontend.

Collaborated on error handling and server-side validations for a strong application.

**Section C: Key Takeaways and Feedback**

6. Key Takeaways and Feedback

**a. Experiences learned working in a group project:**

We gained valuable experiences collaborating on our group project. Working together allowed us to understand the dynamics of teamwork, appreciate diverse perspectives, and improve our communication skills.

**b. Things we wished we could have done better, especially on our project:**

Looking back, we understand the importance of thorough project planning. Establishing precise milestones and deadlines would have allowed us to better organize our time. Additionally, consistent check-ins would have been instrumental in identifying and resolving challenges at an earlier stage in the process.

**c. What we found to be most challenging about the project and the module:**

The most challenging aspect for us was coordinating our schedules. Organizing conflicting priorities and ensuring everyone was on the same page required extra effort. Balancing individual responsibilities and seamlessly integrating our contributions proved to be a significant task.

**d. What aspects of the module need improvement to make the learning experience better:**

Given the overwhelming workload within a tight timeframe, we think the module could be improved by providing clearer guidelines on group dynamics and project expectations. Additional resources or workshops on project management could greatly benefit future students facing similar time constraints.

**e. Did the course meet our expectations? If yes, how, if not, what was lacking?**

Yes, the course met our expectations by providing a thorough understanding of web development concepts. The course imparted knowledge on both front-end and back-end aspects of web development. We gained insights into designing user interfaces (front end) as well as working with databases and server-side scripting (back end). This holistic understanding has equipped us with a comprehensive skill set for web development.

**f. Things we wished were improved to make the module achieve its intended purpose:**

To optimize the module, we propose incorporating the latest frameworks in web development. This enhancement would streamline the learning process and provide us with up-to-date tools, making web development more efficient and aligned with industry practices.

**g. What aspects of the module did we find interesting?**

We found it particularly enjoyable to explore the world of emerging technologies and their practical applications. Additionally, the hands-on experience of analyzing the page structure of real websites added a fun and interactive element to our learning journey.

**h. Suggestions for improvement. If we were to assume the lecture role, how would we improve the module?**

If we were in charge, we'd suggest giving the module an extra week. This would help us absorb the material more thoroughly, especially the complex parts. Having that additional time could also allow for more hands-on practice, making the learning experience richer. It's not just about covering content but really understanding it, and a bit more time could make a big difference for both us students and the instructors.

**i. Did we get the support we needed from both the lecturer and our allocated tutor? Explain your answer in either case.**

Yes, we did receive support from both the lecturer and our allocated tutor.