**Chengdu University of Technology Oxford Brookes College**

**Project Module (CHC 6096)**

**Weekly Report Sheet - 2023/2024 Academic Year**

|  |  |
| --- | --- |
| STUDENT NAME: | Calvin |
| STUDENT NUMBER: | 202018010410 |
| SUPERVISOR NAME: | Monty Mu |
| WEEK NUMBER | 9 |
| DATE: | 2023/12/2 |
| **Action plan for the current week:**  1. Implement the login function and jump to the main page:   * Redirect After Successful Registration:   - Upon successful registration, set up a redirect to the login page to enhance user experience.  - Implement server-side logic to ensure registration information is successfully stored in the database.   * Client-Side Validation Using JavaScript:   - Write JavaScript code to perform preliminary validation of user-entered data, ensuring mandatory fields are not empty.  - Validate password length, username format, etc., to enhance data input accuracy.   * Server-Side Validation:   - Create server-side validation logic to ensure login information is validated before being sent to the server.  - Check if the username and password match records in the database.   * Error Handling:   - Implement error-handling mechanisms to provide clear information to users about login failures, such as invalid username or password.  2. User Password Encryption:   * Select Encryption Algorithm:   - Choose a secure encryption algorithm such as bcrypt or Argon2 for encrypting user passwords.  - Implement server-side logic to hash user passwords, increasing password security.   * Password Encryption Process:   - During user registration, encrypt the password using the selected encryption algorithm.  - During user login, encrypt the entered password using the same algorithm and compare it with the hashed value stored in the database.   * Use of Salt:   - Use randomly generated salt values for password hashing to increase password complexity.  - Store the salt value along with the hashed password in the database.   * Password Security Policies:   - Implement password security policies, such as minimum password length, password complexity requirements, etc.  - Provide a password reset feature to securely allow users to reset their passwords if forgotten.   * Encrypted Transmission:   - Use the HTTPS protocol during user login to ensure password encryption during transmission, preventing man-in-the-middle attacks.  3. Design and implement index page:   * Design the page of index page. * Implement the search bar, menu part and personal information of index page. * Design the customer index page. * Design the merchant index page. | |
| **Challenges and issues encountered in the week:**   1. Balancing Security and User Convenience:  * Implementing strong password policies may conflict with user convenience, So websites need to strike a balance between performance and security.  1. Security of Redirect Mechanism:  * Implementing a secure redirection mechanism after successful registration requires careful handling to prevent unauthorized access or misuse. | |
| **Action plan for the next week:**  1. Redirect After log in:   * Determine the type of account the user is logged in with * If the user login type is a customer, the web page redirects to the index page with customer. * If the user login type is a merchant, the web page redirects to the index page with merchant.   2. Implement the index page with customer:   * Implement the html and css style of search bar. * Implement the html and css style of menu. * Implement the html and css style of shopping cart. * Implement the html and css style of favorites. * Implement the html and css style of personal information.   3. Implement the index page with merchant:   * Implement the html and css style of mystore. * Implement the html and css style of myorders. * Implement the html and css style of personal information. | |
| **Supervisor Feedback:** | |