C237 Team Contribution Planner

As part of your project planning, your team is required to complete the table below. This helps ensure a clear division of responsibilities and provides clarity during your presentation. The completed table must be included as part of your final submission.

Please fill in each column as follows:

* A team member may be responsible for more than one feature. You may add extra rows under the same name if needed.
* **Example:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Team Member Name** | **Feature Assigned** | **Description of Work** | **Route(s) Used** | **SQL Queries Involved** |
| **Student Full Name (Student ID)** | **The main feature/function the student is responsible for.** | **Describe what you worked on for this feature.** | **The Express route(s) created or handled: E.g. /yourRouteHere** | **Any SQL actions used**  **E.g. SELECT, INSERT, etc.** |
| Mary Tan (24012345) | Add New Record | Form to create new item and handle database insert. | /addItem | INSERT INTO items |
| Peter Ho (24054321) | View List | Display all items retrieved from database. | /items | SELECT \* FROM items |

**Complete the table below for your team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Team Member Name** | **Feature Assigned** | **Description of Work** | **Route(s) Used** | **SQL Queries Involved** |
| Kong Li Ting  (24003443) | Login & Signup | Add password validation: need to include one uppercase, one lowercase, one digit, and one special character. (unlimited length allowed) Used SHA1 hashing for password storage and flash messages for user feedback during login and registration  Create a payment.ejs to let user pay for their item and create paymentSucces.ejs for confirmation for their payment | /register  /login  /dashboard  /logout  /index  /payment  /processPayment  /PaymentSuccess | SELECT \* FROM user WHERE email = ? AND password = SHA1(?)  INSERT INTO user (username, email, password, address, contact, role) VALUES (?, ?, SHA1(?), ?, ?, ?) |
| Low Hew Yin, Alicia  (24008484) | View/List Item  Create Item | Create food cart to display food items that user has added in his cart allowing them to review their selected items before placing an order  Add-to-cart route will handle adding items from menu to user’s cart  Once they go to cart table, user can place order and go order confirmation for last confirm before go payment  Create add button in the inventory for only admin to add new food items or update menu inventory ensuring the foot list stays current    Users can save their frequently ordered or favourite food items to a favourites list, making it easier and more convenient to place future orders  Users are allowed to remove favourites either from the favourites list or directly from the menu | /cart  /add-to-cart  /placeOrder  /orderConfirmation  /addInventory  /favourites  /addfavourites  /removefavourite | SELECT \* FROM menuItems WHERE idmenuItems = ?  INSERT INTO orders (iduser, name, total\_amount, order\_date) VALUES (?, ?, ?, NOW())  INSERT INTO orderItems (idorder, idmenuItems, quantity, price) VALUES ?  SELECT \* FROM orders WHERE idorder = ?  SELECT oi.\*, m.name AS foodName FROM orderItems oi JOIN menuItems m ON oi.idmenuItems = m.idmenuItems WHERE oi.idorder = ?    INSERT INTO menuItems (name, image, quantity, price, category) VALUES (?, ?, ?, ?,?)  SELECT mi.\*, uf.created\_at as favourited\_at FROM menuItems mi JOIN user\_favourite uf ON mi.idmenuItems = uf.idmenuItems WHERE uf.iduser = ? ORDER BY uf.created\_at DESC  SELECT \* FROM user\_favourite WHERE iduser = ? AND idmenuItems = ?  DELETE FROM user\_favourite WHERE iduser = ? AND idmenuItems = ? |
| Brandie  (24006767) | View/List Items | Menu to view different culture of food and drinks  Food to view each individual food item  View user is used go view all the user information such as email, contact, address, role  Edit user is to edit user information except for the password  Delete user is to remove user from database  Increase quantity and decrease quantity is use for adding multiple of the same product to cart at once | /menu  /food  /inventory  /viewUser  /editUser  /deleteUser  /increaseQuantity  /decreaseQuantity | SELECT \* from menuItems WHERE idmenuItems = ?  SELECT idmenuItems,name,image,price,category from menuItems  SELECT \* FROM user  SELECT id, username, email, contact, address, role FROM user WHERE id = ?'  Update user SET username =?, email =? , contact = ? ,address = ? , role = ? WHERE id = ?';  DELETE FROM user WHERE id = ? |
| Ryan Danish (24006799) | Edit Item  Delete Item | Order/Inventory can be updated if there is any need  Users & Admin can delete items they don’t want | /editCart  /editInventory  /deleteCart  /deleteInventory | SELECT \* FROM menuItems WHERE idmenuItems = ?  'UPDATE menuItems SET name =?,image = ?, quantity =? , price = ?, category = ? WHERE idmenuItems = ?'  'SELECT \* FROM menuItems WHERE idmenuItems = ?'  ‘DELETE FROM menuItems WHERE idmenuItems = ?’ |
| Zafar  (24000058) | Search/Filter  View/List items | View/List items  - Implemented a category search bar in menu.ejs for users to filter menu items by food category. - Modified /menu route in app.js to handle req.query.category and run dynamic SQL. - Ensured input persistence and handled empty results. - Pop-up confirmation done through UI (e.g., confirm() JS) for delete actions. | /menu | SELECT \* DISTINCT category FROM menuItems WHEREORDER BY category = ? SELECT DISTINCT category FROM menuItems ORDER BY category |
| Xinyi | View/List Item/App  Update |  | /about  /contact  /failpayment |  |