****

**University of Macau**

**CISC 3003 Web Programming**

**Project Assignment – Individual Portion**

DC229567 SUN SHENGHE

# Abstract

As life speeds up, finding time to sit down for a meal becomes increasingly challenging. That's where food ordering platforms come in handy. Our website, a collaboration credited to Vijay Thapa, encompasses both frontend and backend web application design.

For website administrators, managing their platform is a breeze. They can effortlessly handle tasks like adding, updating, or deleting items from the food menu through an intuitive interface. Additionally, admin account management is streamlined, ensuring smooth operations.

On the customer side, finding and ordering their desired food is a simple process. Our website offers a user-friendly interface where customers can easily search for their preferred dishes and place orders hassle-free.

Built with a blend of HTML, CSS, JavaScript, and PHP, our website is versatile and efficient. MySQL is employed for database management, with stringent measures in place to protect against SQL injection, safeguarding the integrity of our data.

In summary, our website serves as a valuable tool for administrators, simplifying website management tasks, while also saving time for customers by offering a seamless ordering experience.

# List of Services

* **Administrator**
  1. Access the admin page through login credentials.
  2. Manage admin accounts by adding, updating, or removing them.
  3. Handle food categories by adding, updating, or removing them, including:
     1. Manage the featured category display on home page
     2. Manage the food category display on all food page
  4. Manage food items by adding, updating, or removing them, including:

1. Manage the featured food display on home page
2. Manage the foods display on foods page
   1. Update the customers’ orders
      1. Update the info of the orders (quantities, status, customer information)
   2. View data (number of categories, foods, orders, total revenue) on admin page

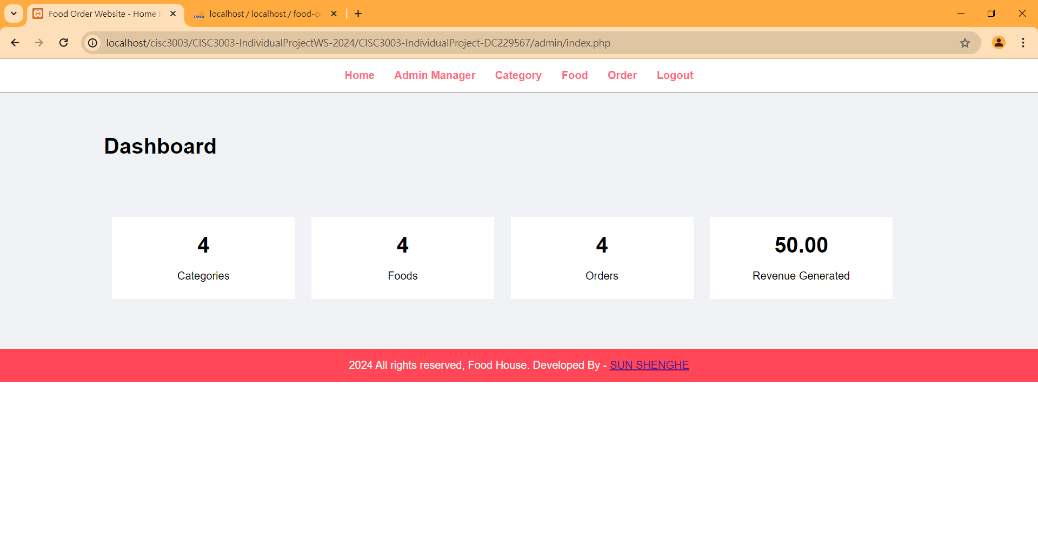
* **User**
  1. Search the food in the website
  2. Make orders on different foods
* **System**
  1. Protect the database from SQL injection

# List of Tasks

1. Employ session management to restrict access to admin pages unless users have logged in.
2. Develop index.php as an admin dashboard, showcasing key website statistics such as the number of categories, foods, orders, and total revenues.
3. Implement measures to prevent users from accessing unauthorized pages by ensuring the presence of necessary query strings.
4. Manage Admin Accounts
5. Create manage-admin.php to give an interface that lists all of the admin accounts and the information in the database with links to add, update, delete, password changing pages.
6. Create update-admin.php to update the information of the accounts in database, and send back a message of success or fail.
7. Create delete-admin.php to delete the accounts from database, and send back a message of success or fail.
8. Create add-admin.php to add accounts to database, and send back a message of success or fail.
9. Manage Categories
   1. Create manage-category.php to give an interface that lists all of the categories and the information in the database with links to add, update, delete pages.
   2. Create update-category.php to update the information of the categories in database, and send back a message of success or fail.
   3. Create delete-category.php to delete the categories from database, and send back a message of success or fail.
   4. Create add-category.php to add categories to database, and send back a message of success or fail.
10. Manage Foods
    1. Create manage-food.php to give an interface that lists all of the foods and the information in the database with links to add, update, delete pages.
    2. Create update-food.php to update the information of the foods in database, and send back a message of success or fail.
    3. Create delete-food.php to delete the foods from database, and send back a message of success or fail.
    4. Create add-food.php to add foods to database, and send back a message of success or fail.
11. Manage Orders
    1. Establish manage-order.php to provide an interface listing all orders and associated information from the database, including links redirecting to the update page.
    2. Develop update-order.php to facilitate updating order information within the database, notifying users of success or failure through appropriate messages.
12. Website
    1. List at most 3 featured and active categories on home page
    2. List at most 6 featured and active foods on home page
    3. List all active categories on categories page
    4. List all active foods on foods page
    5. User can search the foods by inputting keywords
    6. User can make orders to different foods
13. Protect the database from SQL injection

# Accomplishment

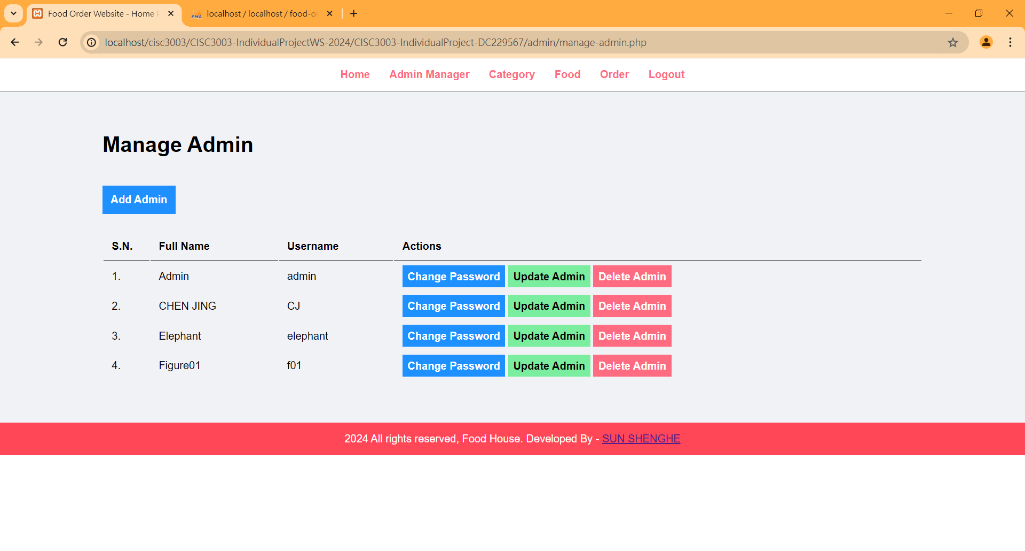
1. Create index.php as a dashboard page for admin with displaying information



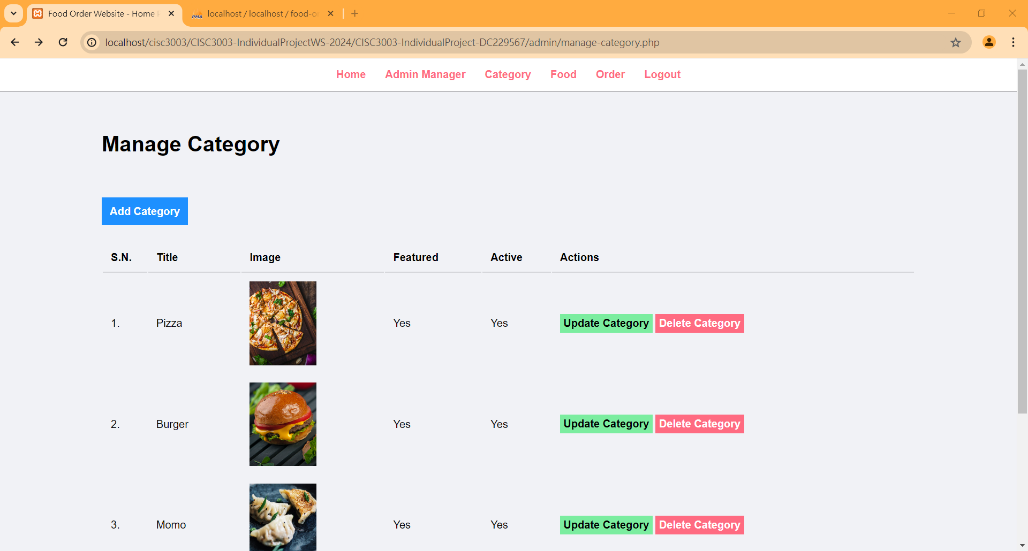
1. Prevent users from directing to different pages without necessary query strings

* For example, if we direct to category-foods.php without “category\_id” query strings, it will redirect to the home page.

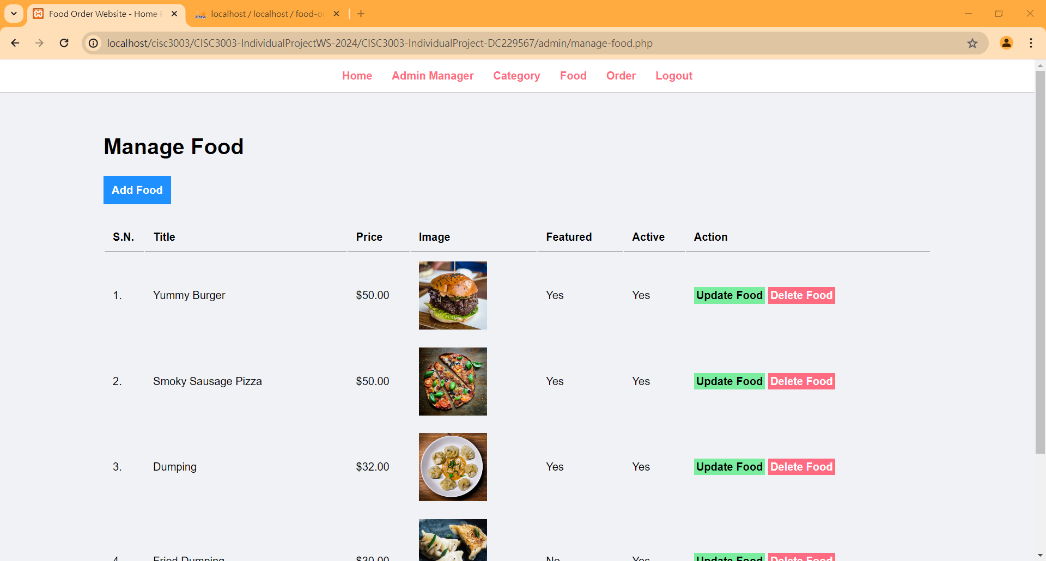
1. Manage Admin Accounts



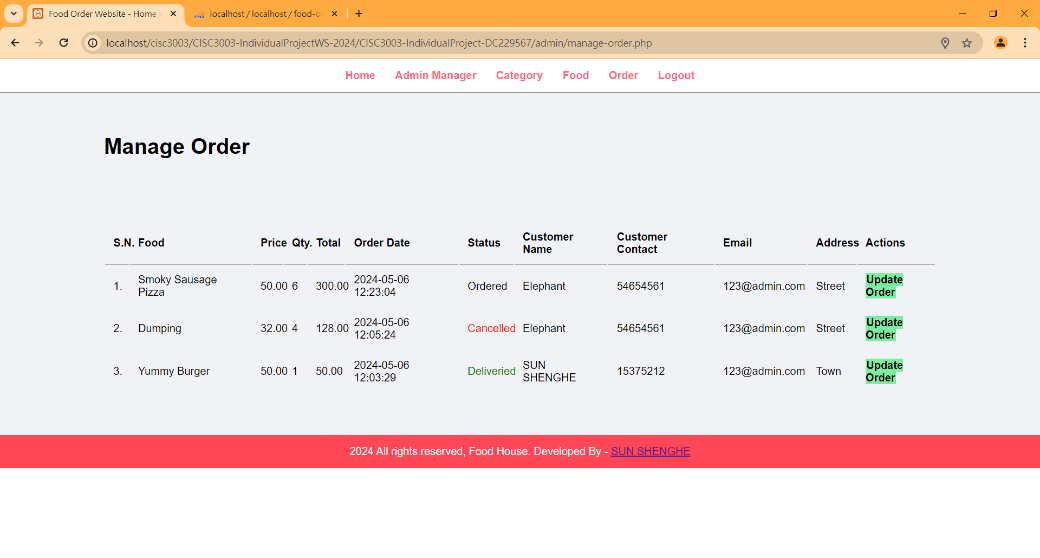
1. Manage Categories



1. Manage Foods



1. Manage Orders



1. Website

# Incomplete

1. Enhance database security to mitigate SQL injection risks. Certain inputs, such as those in manager-add.php and manager-update.php, remain vulnerable. Malicious input poses a significant threat to the database integrity and must be addressed promptly.
2. Address username duplication issue. Currently, admin account usernames can be duplicated, leading to login complications, where the same username and password may result in different account IDs. Implement measures to prevent username duplication and ensure a seamless login experience.
3. Implement validation to prevent the creation of accounts with empty information. Currently, accounts can be created with blank full names, usernames, and passwords, which is undesirable. Implement checks to ensure that all necessary information is provided during account creation to maintain data integrity and system functionality.