COMP639 Studio Project – Summer School
Individual Assignment

Worth: **40%**

Due: **Wed, 6th December 2023 @ 5:00pm**

Late Penalty: Work not received by due time attracts an immediate penalty of up to 25% of the marks available. No work will be accepted after **Fri, 8th December 2023 5:00pm**.

A Simple Holiday House Rental System

Develop a Flask Python Web App for a Holiday House Rental System. The application should include a login system and separate dashboards for three user roles: customer, staff and admin. Users should be able to login to the system and access their respective dashboards as well as to perform specific actions related to holiday house rental operations. This is a simplified version without the booking and rental functionalities as the focus is on providing different level of access for different user roles.

User Login

You will need to:

- Create a login page where users can enter using a username and password.
- Implement a password hashing and salting techniques to ensure secure storage of user passwords.
- Provide a new user registration functionality that allows new customer users to register and create a customer account themselves. In this process they will provide their details and set a unique username and password.

User Roles and Access Control

- Define three user roles: customer, staff and admin.
- Implement a role-base access control system that restricts access to certain pages or features based on the user's role.
- Customers should be able to manage their own profile (update personal information and change password) and view all the holiday houses available for rent. When viewing holiday houses, customers should be able to click on each holiday house to get detailed information.
- Staff members should be able to manage their own profile (update personal information and change password), view customer profiles and manage holiday houses (add, update and delete holiday houses).
- Admin should have full access to the system and the ability to manage customers (add, update and delete customers), staff (add, update and delete staff) and holiday houses (add, update and delete holiday houses) and view the list of all holiday houses.

Data Requirements

- Customer profile (name, customer number, address, email, phone number)
- Staff profile (name, staff number, email, phone number, date joined)

- Holiday Houses (house id, house address, number of bedrooms, number of bathrooms, maximum occupancy, rental per night, house image)
- The database should contain at least 5 customers, 3 staff, one admin and 20 different holiday houses.
- Design and implement a database schema and populate it with data to meet these requirements.

Dashboard Pages

- Create separate dashboard pages for each user role.
- Design and implement a visually appealing user interface for each dashboard.
- Customise the functionality and features available on each dashboard based on the associated user role.

Deliverables

- A fully functional Flask Python web application for a holiday house rental system with a login system and a role-based dashboards.

Project Requirements

You must

- Use only Python & Flask, Bootstrap CSS, JavaScript, MySQL. Do not use SQLAlchemy or ReactJS (or other similar technologies) in your solution.
- Create a private GitHub repository that contains:
    - All Python, HTML, images and any other required files for the web app
    - A requirements.txt file showing the required pip packages.
    - MySQL scripts for creating and populating the database.
    - README file with comments.
    - Your repository must have a .gitignore file and therefore not have a copy of your virtual environment.
    - Add lincolnmac (computing@lincoln.ac.nz) as a collaborator to your private GitHub repository.
- Host your system (including database) using pythonAnywhere
    - Add lincolnmac as your “teacher” via the site configuration.
- Submit via the link on LEARN COMP639 Page the **COMP639 Web App Hand-In Sheet**. This includes details of:
    - Your PythonAnywhere URL.
    - Your GitHub repository URL.
    - Usernames and Passwords for Customers, Staff and Admin for testing purposes.
    - Confirmation that certain files have been saved in you GitHub repository.

Marking

Criteria	Marks
Overall Structure and Organization: • Clear and logical structure of the web application. • Proper use of templates, modules, and components. • Adherence to coding and naming conventions.	**10**
Consistent and visually appealing design across all pages. • Responsive layout that adapts to different screen sizes. • Intuitive and user-friendly interface for easy navigation and interaction.	**10**
Database and Data Management: • Proper database schema design and implementation. • Correct usage of database models and relationships. • Appropriate handling of database queries, updates, and deletions. • Data integrity and security considerations.	**10**
Features: • Add, update and delete holiday house rentals (5) • Add, update and delete customers and staff (10) • Register new customer users (5) • Change password (5) • List all holiday house rentals (10) • Login and logout (10) • Dashboard specific based on user role (5)	**50**
Error Handling and Validation: • Proper validation and error handling throughout the application. • Proper use of form validation and error handling. • Appropriate display of error messages and user feedback.	**10**
Documentation and Readability: • Well-structured and readable code with proper comments. • Compliance with coding style guidelines.	**10**