MOUNTAINS OF THE MOON UNIVERSITY
Department of Computer Science
Central Test

BIT 1203 Internet Technologies and Advanced Website

This will be submitted and presented individually on 8th October 2024 starting from 8:00AM to 6pm (First Shift 8:00am to 1pm, second shift 2pm to 6pm)

Min Project

You are required to design and develop a comprehensive web application that allows users to log in and manage data of CS and IT Year 2 related to students, course units, and staff members in an academic system. The application should incorporate **Bootstrap (BS)** for front-end styling, **JavaScript (JS)** for client-side interactivity and data visualization, and **server-side development** using a language of your choice (e.g., PHP) to interact with a **MySQL database**.

Functional Requirements

1. **Login System**
   - Implement a user authentication system where users can log in using a username and password.
   - Ensure secure handling of user credentials (e.g., password hashing).
   - Redirect the user to a personalized dashboard upon successful login. Display an error message for invalid login attempts.
2. **Dashboard**
   - After login, users should be presented with a dashboard displaying key statistics related to students, courses, and staff.
   - Use **Bootstrap** to create a responsive and user-friendly layout with components such as cards, tables, and grids.
   - Integrate **JavaScript** (or a charting library like **Chart.js**) to display graphical representations (e.g., bar charts, line charts, pie charts) of the data, such as the number of students enrolled in each course or the distribution of staff across departments.
3. **Data Management Forms**
   - Develop **forms** that allow users to **add** data about;
     - **Students**: Including fields such as name, student RegNo, Gender email, enrolled courses, etc. (Find what is relevant)
     - **Course Units**: Including fields such as course name, course code, credit units, etc.
     - **Staff**: Including fields such as name, staff ID, Gender, department, assigned courses, etc.
   - The forms should be integrated with the **MySQL database**, allowing users to **submit** data which is then stored securely.

4. **CRUD Operations (Create, Read, Update, Delete**
    - Implement full **CRUD functionality** for managing students, course units, and staff:
        - **Create**: Allow users to add new entries for students, course units, and staff.
        - **Read**: Display the existing data in a tabular format, allowing users to view details of all students, course units, and staff.
        - **Update**: Provide functionality to **edit** and update existing records.
        - **Delete**: Allow users to **delete** records from the database.
5. **Search and Filter Functionality:**
    - Implement a **search bar** or **filtering options** that allow users to:
        - Search for students by name, RegNo or ID, Gender, or enrolled course.
        - Filter courses based on department or course code.
        - Search staff by name, ID, or department.
    - The search functionality should be **real-time** and allow users to easily find and manage the desired records.
6. **Additional Features**
    - Include a **logout** option that securely ends the user session.
    - Ensure proper **validation** of all input fields on both the client-side (using JavaScript) and server-side to prevent SQL injection and other security vulnerabilities.
    - Implement session management to ensure data security and protect against unauthorized access.

Technical Requirements

1. **Data Collection method: -** Utilize data collections methods and Tools to collect real and relevant data from the users or stakeholders (in this case CS staff and Students)
2. **Bootstrap (BS)**: -Use Bootstrap to ensure the application is responsive and user-friendly.
3. **JavaScript (JS)**: -Use JavaScript (or libraries such as Chart.js) to implement data visualization and enhance user interactions.
4. **Server-Side Development**: - Use a server-side language such as **PHP, or Python** to handle user authentication, form submissions, and data retrieval from the MySQL database.
5. **MySQL Database**: Use **MySQL** to store all relevant data, including users, students, courses, and staff.
6. **Security Considerations**: Ensure secure handling of data, including password hashing, input validation, and protection against SQL injection and XSS attacks.

Assessment Criteria

1. Evidence of data collection (e.g., Tool, datasets, etc) (5 marks)
2. Secure and functional working **Login System** (5 marks)
3. A well-structured, **responsive dashboard** with Bootstrap components and data visualization (10 marks)
4. Functional **data entry forms** for students, courses, and staff (10 marks)
5. Complete **CRUD operations** (Create, Read, Update, Delete) for managing data (10 marks)
6. **Search and filtering functionality** for easy data retrieval (5 marks)

7. **Validation and security** (including password handling, input validation, session management) (5 marks)

NOTE: ***All this assessment scores will base on how a student presents and explains the components to the Examiner. (Understand your code).***