Campus Connect

Project By:

Budharam Suresh (21BCS043)

Rohit Verma (21BCS059)

Nitika (21BCS059)

Shivani (21BCS114)

Under Supervision of

Dr. Priyanka

Project Documentation

1. Purpose and Objective:

- The **Discussion Portal** aims to create a user-friendly and interactive platform for students to post queries and for faculty to provide answers. The primary goal is to facilitate academic discussions beyond the classroom, enhancing learning and communication between students and instructors.

2. Technology Stack:

- **Frontend Technologies**:
  - **HTML**: Used to structure the web pages.
  - **CSS**: For styling and designing a user-friendly interface.
  - **JavaScript**: For client-side scripting to handle dynamic content and interactivity.
- **Backend Technologies**:
  - **PHP (Hypertext Preprocessor)**: The primary server-side scripting language used to handle requests, process user input, and interact with the database.
  - **MySQL**: The database management system used to store all user data, questions, answers, and other portal information.
- **Server**: The portal is deployed on a **PHP-compatible server** that can interact with MySQL to provide the necessary backend processing.

3. System Architecture:

- The project follows a **Three-Tier Architecture**:
  1. **Client Layer (Frontend)**: This layer is responsible for user interaction. It consists of web forms, buttons, and other graphical elements that allow users (students, faculty, and admin) to perform tasks such as login, posting questions, or answering queries.

2. **Application Layer (Backend)**: PHP scripts handle the business logic, processing requests from the client and interacting with the database.

3. **Data Layer (Database)**: MySQL stores all the necessary information, including user accounts, questions, answers, and posts.

- **Security**: The separation of layers enhances security by ensuring that sensitive operations (like database transactions) are handled exclusively in the backend.

4. Core Features & Modules:

4.1 User Roles:

- **Admin**:
  - Responsible for managing the entire portal, including user accounts, posts, and handling flagged content.
  - Admin can add, update, delete users and posts, moderate discussions, and handle reported posts.
- **Faculty**:
  - Can log in, view student queries, post answers, and interact with students.
  - Faculty have options to update their profiles and respond to posts.
- **Students**:
  - Can post questions, view answers, and interact with other students and faculty.
  - They can update their profiles, view and edit their posts, and participate in discussions.

4.2 Key Functionalities:

- **Registration and Authentication**:
  - Users register using their **college credentials**. Admin approves new faculty members, ensuring that only authorized users have access.
  - Students and faculty can log in using Google authentication or manually with their college ID and password.
- **Post Management**:
  - Students can post new questions and reply to others' queries.
  - Faculty can answer queries, while admins can delete inappropriate posts flagged by users.
- **Search Functionality**:
  - Users can search for previously posted questions to check if their queries have already been answered.
  - Search filters are implemented to sort questions by topics, dates, or relevance.

- **Upvote/Downvote System**:
  - An optional voting mechanism allows users to upvote or downvote questions and answers to highlight helpful responses.

6. Database Design:

- **Tables**:
  - **Users Table**: Stores user credentials, including IDs, passwords, and roles (admin, faculty, student).
  - **Posts Table**: Contains all the posted questions and answers, including metadata such as timestamps and user information.
  - **Forum Table**: Manages the different forums (e.g., programming, general knowledge) where students can post their questions.
- **CRUD Operations** (Create, Read, Update, Delete):
  - Users (students and faculty) can create posts, read others' posts, update their own posts, and the admin has the ability to delete posts.

8. Design Decisions & Tradeoffs:

- **Multi-Screen Design**: The decision to have separate screens for admin, faculty, and student roles was made to improve encapsulation and data privacy. This ensures that users only see the information relevant to their role.
- **Button Navigation**: Instead of links, buttons are used to improve visibility and accessibility. This ensures that users can navigate the portal easily and reduces the chances of confusion or navigation errors.

11. Execution and Runtime Environment:

- The system can run on any device supporting PHP and MySQL.
- **Browsers**: The portal is compatible with all modern web browsers (Chrome, Firefox, Safari).

Conclusion:

This Discussion Portal provides a robust, scalable, and interactive solution for student-instructor engagement. It is designed with extensibility in mind, allowing for future upgrades like adding online assignments or real-time notifications.