

Article Submission System

Project Concept:

The **Article Submission System** is a web-based platform developed as the final project for the **Database Management Systems** course. The platform allows users to **submit, read, and rate articles** while also providing features to rank users based on their activity. It includes an **admin panel** for managing content and users. Built using **MySQL** for database management, **HTML/CSS** for front-end design, and **Python** for back-end functionality, this project combines usability, scalability, and efficiency.

Core Features:

1. User Features:

- **User Registration and Login:**
 - Login with password hashing for security.
- **Article Submission:** Users can write and submit articles for approval.
- **Article Reading and Voting:**
 - Users can browse and read articles.
 - Articles can be rated using a star-based or thumbs-up voting system.
- **User Rankings:**
 - **Best Readers:** Users ranked based on the number of articles read and rated.
 - **Best Scored Authors:** Users ranked by the total ratings their articles have received.

2. Admin Panel:

- **Article Management:**
 - Review, approve, or reject articles submitted by users.
 - Edit or delete articles when necessary.
- **User Management:** Manage user accounts, including banning or unblocking users.
- **System Metrics:**
 - View user activity statistics.
 - Track article performance, including views, votes, and ratings.

3. Dynamic Ranking System:

- **Best Readers:**
 - Calculated based on the number of articles a user has read and rated.
- **Best Scored Authors:**
 - Based on the total score accumulated by the articles submitted by a user.
- Rankings are updated dynamically and displayed on a leaderboard.

4. Database Design (MySQL):

- **Tables:**
 - **Users:** Stores user data, including roles, activity stats, and credentials.
 - **Articles:** Tracks article content, author, status, and rating.
 - **Ratings:** Logs votes and comments for each article.
 - **Activity:** Tracks user interactions like articles read and rated.
- **Relationships:**
 - Articles are linked to users via foreign keys.
 - Ratings are linked to both articles and users for ranking calculations.

5. Front-End (HTML/CSS):

- **Responsive Design:** Optimized for both desktop and mobile users.
- **Leaderboards:** Dedicated pages for **Best Readers** and **Best Scored Authors**.
- **Article Details:** Includes title, content, author info, and user ratings.

6. Back-End (Python):

- Developed using **Flask**.
- Handles authentication, database queries, and ranking algorithms.
- **Session Management:** Ensures logged-in users can interact securely with the system.

Additional Features:

- **Real-Time Updates:** Dynamic refresh of leaderboards and article scores.
- **Comment System:** Readers can leave comments on articles to engage with authors.
- **Email Integration:** Notification emails for article approvals and new leaderboards.

Tools and Technologies:

- **MySQL:** Handles user, article, and ranking data efficiently.
- **HTML/CSS/Bootstrap:** Ensures a clean and responsive user interface.
- **Python (Flask):** Manages back-end logic and API routes.

This system not only provides a functional platform for article submissions but also introduces **gamification** elements through rankings like **Best Readers** and **Best Scored Authors**, encouraging user engagement and competition. This combination of features ensures an interactive and rewarding user experience.