DATABASE MANAGEMENT SYSTEM
PROJECT PHASE 3

Assigned Topic: Article Submission System

Gizem Tuğuz
Ömer Faruk Yılmaz

Table of Contents

1. Introduction

This project aims to develop a dynamic article submission and management system that serves authors, reviewers, administrators, and general users. Designed to streamline the process of submitting, reviewing, and accessing articles, it offers personalized content based on user interests, incorporates feedback mechanisms, and provides insights like the most read articles and popular authors. By focusing on user engagement and data-driven recommendations, this system intends to enhance the digital content experience, fostering a vibrant community around knowledge sharing and discovery.

2. Application Program Code and Database Dump

Application Program Code

The application program code consists of various modules and scripts that implement the functionalities of the Article Submission System. The main components of the codebase include:

User Interface (UI) Module: This module handles the front-end part of the application, providing users with an interactive interface to submit, review, and read articles.

Backend Server Module: This module manages the server-side operations, including user authentication, data processing, and communication between the frontend and the database.

Database Interaction Module: This module contains scripts and functions to interact with the database, perform CRUD (Create, Read, Update, Delete) operations, and execute advanced queries.

Notification System: This part of the code handles sending automated notifications to users regarding article status updates, feedback, and other relevant communications.

Database Dump

The database dump file contains the complete schema and data of the Article Submission System database. This includes:

User Tables: Storing information about users, including their profiles, roles (authors, reviewers, administrators), and reading histories.

Article Tables: Storing details about articles submitted to the system, including titles, content, submission dates, and statuses.

Feedback Tables: Storing feedback and scores provided by reviewers and readers for each article.

Notification Tables: Managing notifications sent to users about various events in the system.

These components together form the backbone of the Article Submission System, enabling it to function as intended.

3. How to Run the Program

To run the Article Submission System application, follow these steps:

1. **Database Setup:**
  - Install the DBMS (MySQL, Microsoft SQL Server, or PostgreSQL).
  - Import the database dump file using the following command:
  ```
  mysql -u [username] -p [database_name] < [path_to_dump_file]
  ```
  - Ensure the database server is running.

2. **Application Setup:**
  - Install necessary dependencies (e.g., using `pip` for Python or `npm` for Node.js).
  - Navigate to the application directory.
  - Configure the database connection settings in the application configuration file.

3. **Running the Application:**
  - Start the application using the appropriate command (e.g., `python app.py` or `npm start`).
  - Open a web browser and go to `http://localhost:[port_number]`.

4. **Testing the Application:**
  - Log in using the provided test accounts.
  - Navigate through the application to explore different features.
  - Use the admin account to manage users and content.

5. **Advanced Queries:**
  - Access the advanced queries section to view the results of predefined queries.

4. Detailed Report

4.1. System Functionalities

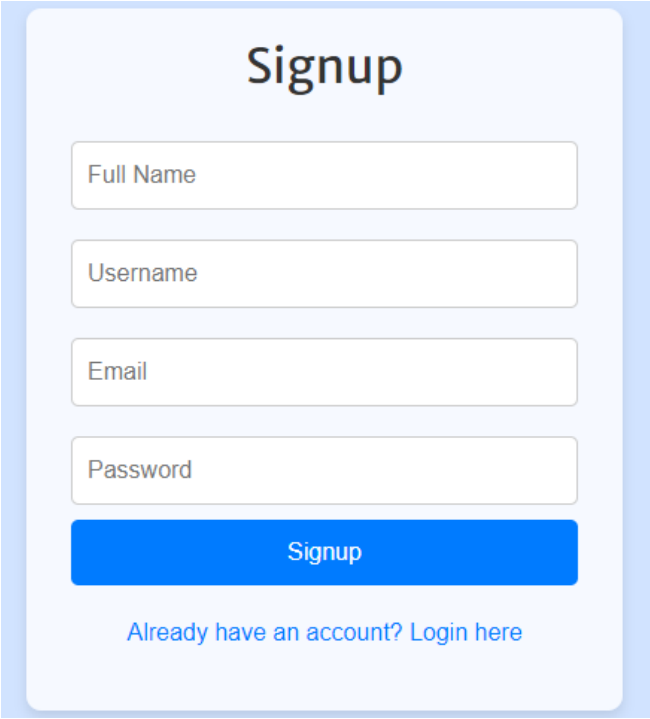
The Article Submission System is designed to cater to various user roles including authors, reviewers, administrators, and general users. Below is a detailed description of the functionalities provided by the system along with screenshots of the user interfaces.

User Roles and Functionalities

General Users:

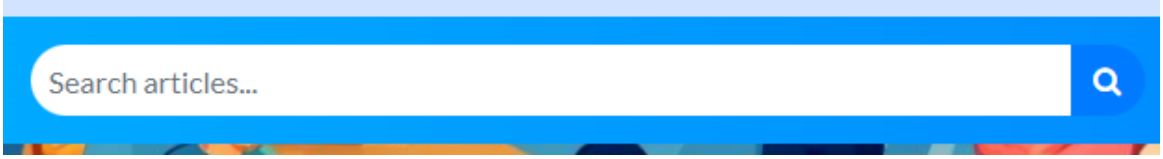
Create and Manage Account:

Users can create an account, log in, and update their profiles, including profile pictures and personal information.



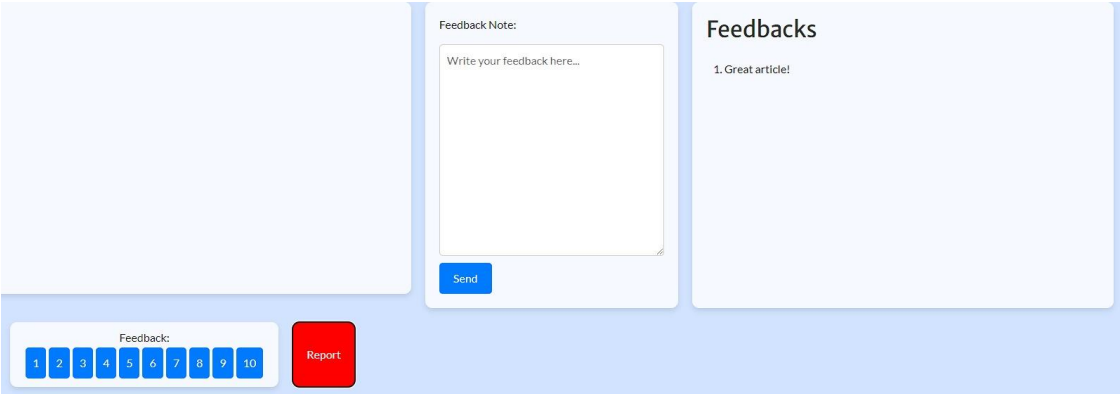
Read Articles:

Users can browse and read articles from all authors.



Send Feedback on Articles:

After reading an article, users can provide feedback to the author with a score from 1 to 10 and a feedback note.

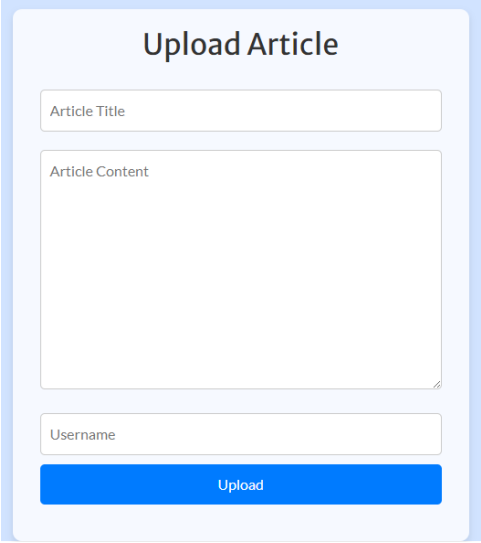


Report Issues:

Users can report issues to administrators if needed.

Authors:

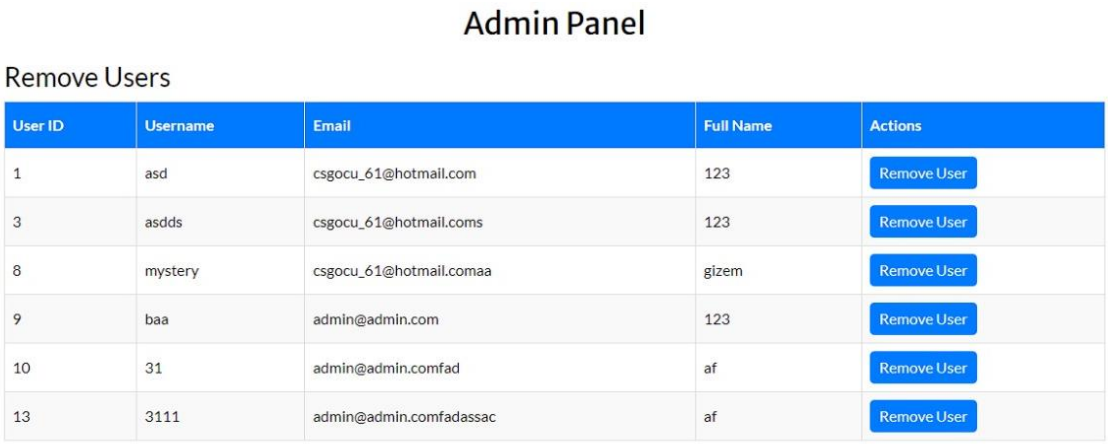
Submit Articles: Authors can submit their articles to the system by using their username.



Administrators:

User Management:

Administrators manage user accounts, including creation, role assignments, and resolving access issues.



Screenshot: User Management Page

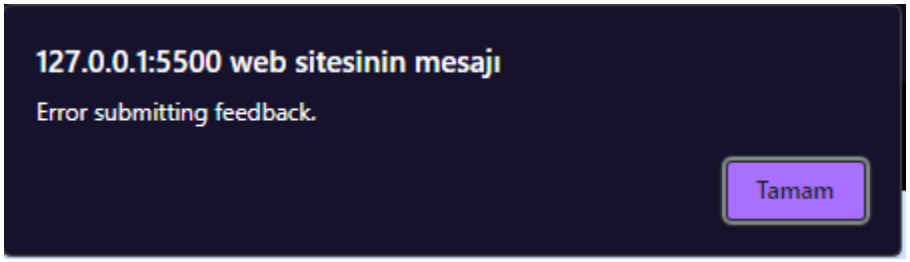
Role Flexibility:

The system accommodates users with multiple roles (e.g., an author who is also a reviewer).

Key Features:

Notification System:

An automated notification system for status updates.

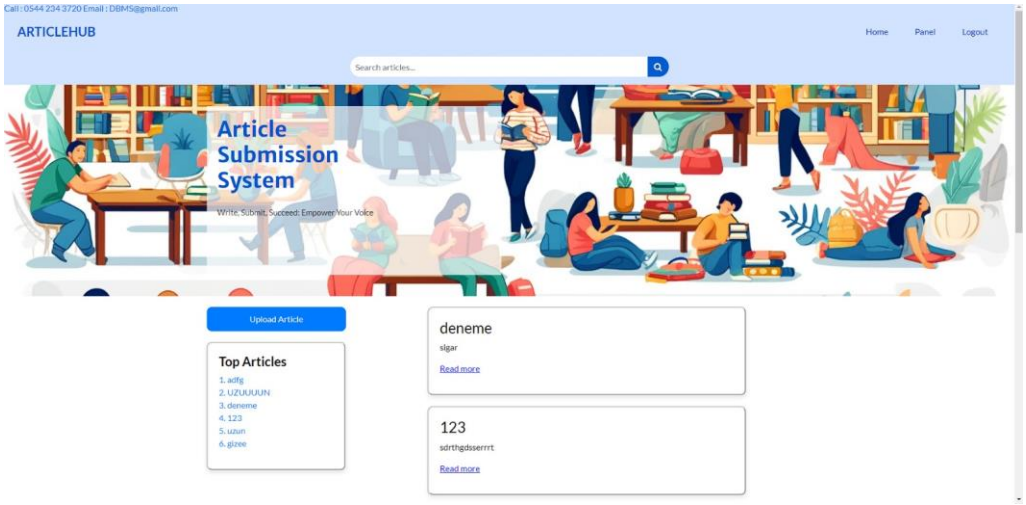


Search and Filtering:

Advanced search functionality for administrators and editors to filter submissions, reviews, and user accounts based on various criteria.

Data Insights:

The system provides insights such as the most read article of the month, most viewed author, articles with the highest scores, etc., displayed on the users' main page.

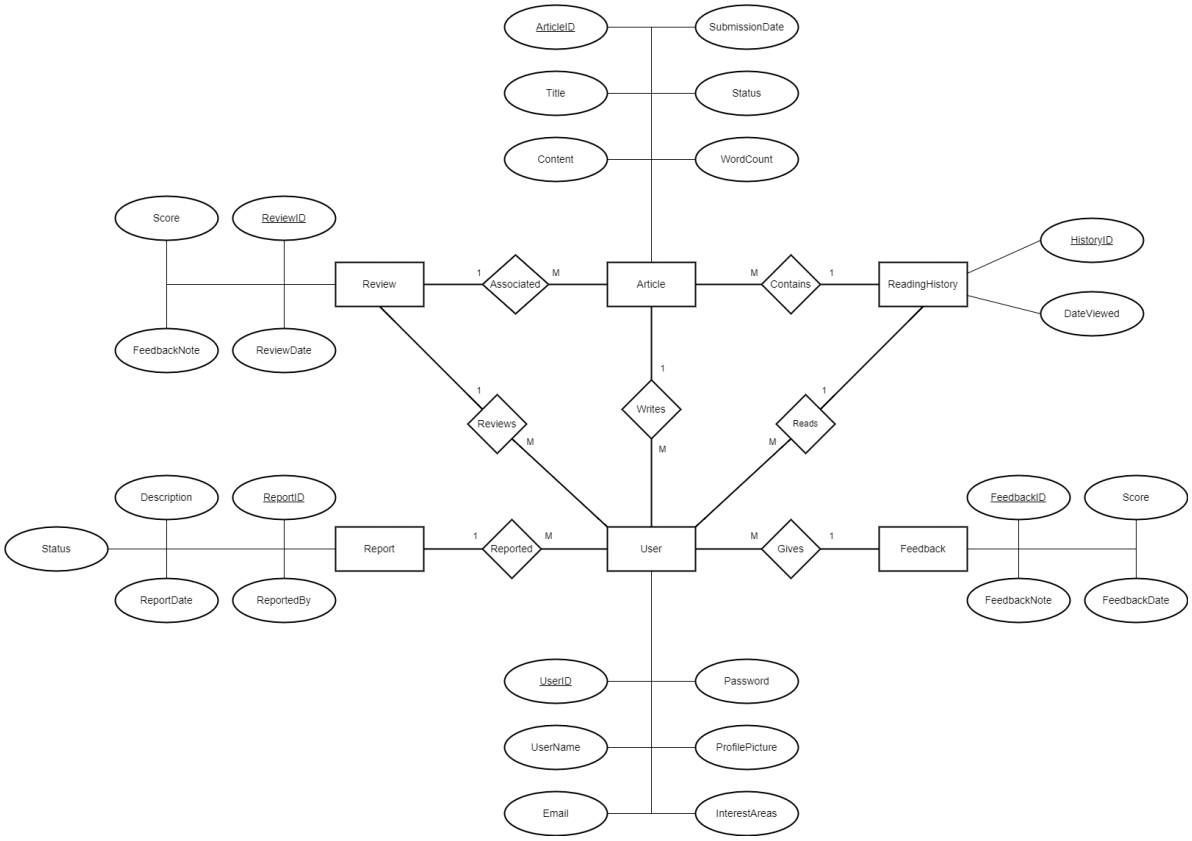


By integrating these functionalities, the Article Submission System aims to provide a comprehensive and engaging platform for managing and interacting with digital content.

4.2. Advanced Queries

- What is the most read article of the month?
- What is the article with the highest feedback score?
- Top 5 read articles in each category this month.
- The most popular author based on monthly reads.
- The young author of the month based on read numbers.

4.3. Database Schema



4.4. Testing and Results

The testing phase of the ArticleHub project was comprehensive and aimed at ensuring the application functions as intended. The testing procedures included unit testing, integration testing, and user acceptance testing. These tests were designed to verify the functionality, usability, and reliability of the system.

Testing Procedures

Unit Testing: Each module and function in the application was tested individually. For instance, the user authentication module was tested to ensure that users could sign up, log in, and manage their profiles securely. The article management module was tested to verify that authors could create, edit, and submit articles without issues.

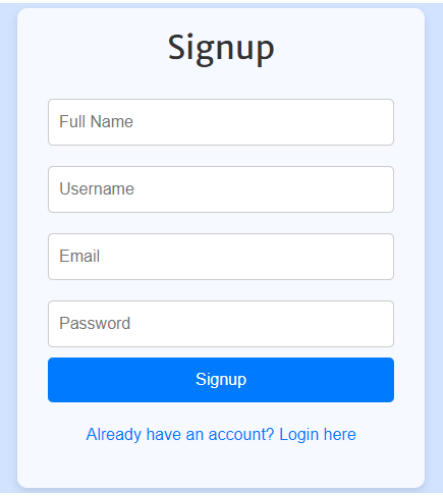
Integration Testing: This phase involved testing the interaction between different modules. For example, we tested the integration of the article submission module with the feedback and rating system to ensure that users could provide feedback on published articles and that this feedback was correctly stored and displayed.

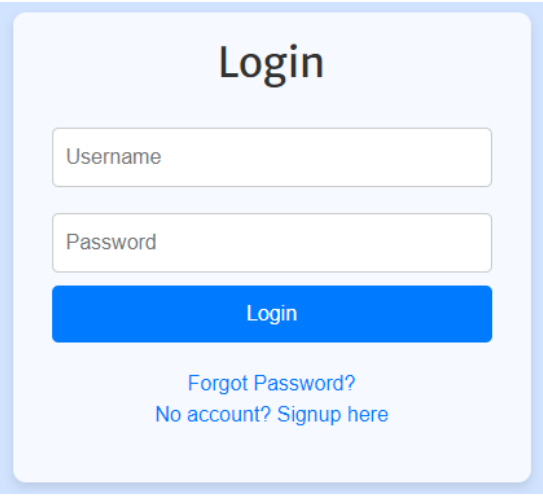
Testing Results

The testing procedures confirmed that the application functions as intended. Below are some key results from the testing phase, along with screenshots demonstrating the application's functionality:

User Registration and Login:

Result: Users can successfully register and log in.

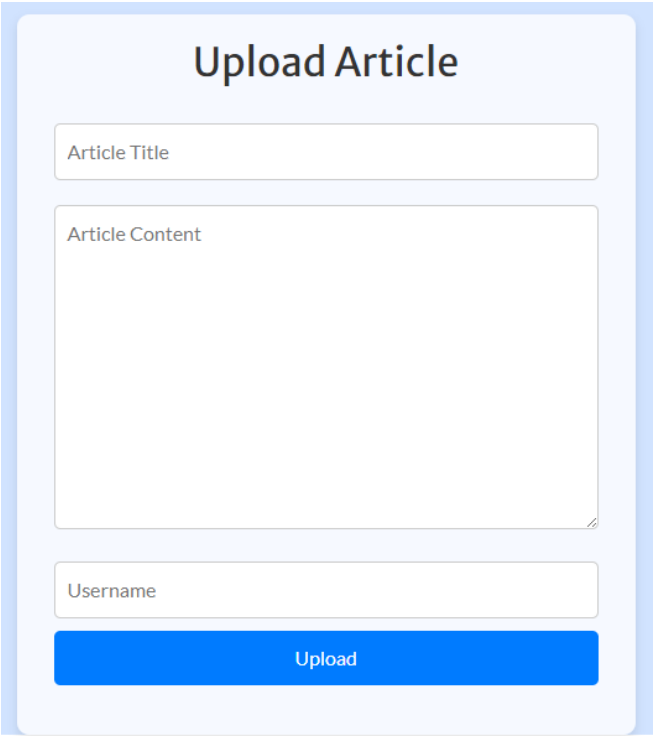




Description: The registration form captures user details, including username, password, email, and profile picture. The login screen verifies user credentials.

Article Creation and Submission:

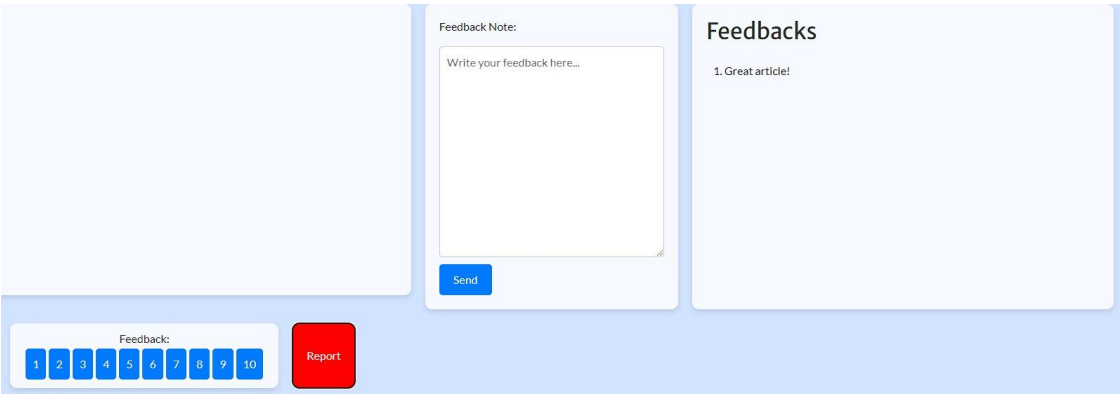
Result: Authors can create, edit, and submit articles.



Description: The article creation form allows authors to input the title, content, and status. Articles can be saved as drafts or submitted for review.

Feedback and Rating:

Result: Users can provide feedback and rate articles.



Description: The feedback form captures the user's rating and comments. Feedback is linked to the specific article and user.

Issue Reporting:

Result: Users can report issues related to articles.

Description: The reporting form allows users to describe issues with articles. Reports are sent to administrators for review.

Overall, the testing phase demonstrated that the ArticleHub application meets its requirements and functions as expected. The successful implementation of user registration, article management, feedback and rating systems and issue reporting confirms the application's reliability and usability. The provided screenshots offer visual evidence of the application working as intended.

4.5. Conclusion

The ArticleHub project successfully created an article submission and management system. Key achievements include a well-structured relational database schema for managing users, articles, feedback, and reports. The system supports secure user authentication, role-based access control, and various article statuses, facilitating a smooth workflow from creation to publication. Users can provide detailed feedback and ratings, which enhance content quality and engagement.

The development process offered valuable lessons in software development and project management. Designing a scalable and efficient database schema was crucial. Focusing on user experience and incorporating regular feedback improved the UI/UX. Implementing secure authentication and data protection practices built a trustworthy platform. An iterative development approach allowed for continuous testing and timely adjustments, ensuring a stable application. Collaboration with team members Gizem Tuğuz and Ömer Faruk Yılmaz was key to the project's success, highlighting the importance of effective communication and task delegation. Addressing challenges like technical issues and scope changes required adaptability and problem-solving skills, contributing to personal and professional growth.

In conclusion, the ArticleHub project achieved its goals and provided a valuable learning experience. The knowledge and skills gained will benefit future endeavors in software development and project management.