A REPORT ON

“Role-Based Blogging & Article Review Portal”

Diploma in Advanced Computing

Infoway Technologies

August-2025

SUBMITTED BY

Akshada Kshirsagar

Shweta Avhad

Vidya Thange

ABSTRACT

The Role-Based Blogging and Article Review Platform is a web-based application designed to facilitate collaborative content creation and controlled publishing through a secure and structured role-based access control mechanism. The platform supports multiple user roles—Admin, Editor, Author, Reviewer, and Reader—each with clearly defined permissions and responsibilities to ensure proper workflow management and system security. This role segregation allows the platform to maintain content quality, accountability, and efficient collaboration among users. Authors are responsible for creating, editing, and submitting articles for review. Editors evaluate submitted content, provide feedback, and approve or reject articles based on quality and relevance. Administrators manage user accounts, assign roles, and configure platform settings to ensure smooth system operation. Reviewers and readers enhance platform interactivity by engaging with published content through comments, ratings, and feedback, thereby fostering a collaborative and knowledge-sharing environment. The application adopts a hybrid backend architecture to improve modularity and scalability. Java-based services are used exclusively for authentication-related functionalities, including user registration, login, logout, and secure token generation. All core business logic, content management operations, role authorization, and workflow handling are implemented using ASP.NET Core Web API, ensuring efficient processing and maintainability. The frontend is developed using React.js, offering a responsive, dynamic, and user-friendly interface that enhances the overall user experience. By separating authentication from business logic and employing modern web technologies, the platform achieves improved security, flexibility, and scalability. The system is suitable for real-world applications such as educational portals, research publication platforms, corporate blogging systems, and community-driven content platforms, providing a robust solution for structured digital content creation and management.

Contents

Pseudo Code

Evaluation Criteria

Mathematical Model

5. **Software Requirements Specification**

6. **Hardware Requirements**

7. System Architecture

Data Flow Diagram

UML Diagrams

Use Case Diagram

Package Diagram

Sequence Diagram

Activity Diagram

7. **Test Specification**

Implementation

Types Of Testing

Unit testing

Integration testing

Functional test

System Test

White Box Testing

Black Box Testing

Acceptance Testing

Test cases and Test Results

Analysis of System Modules

9. Data Table and Discussion

Discussion

Application

Advantages

Data Tables

10. Results and Discussion

Analysis

Model Train

Efficiency Calculation Parameters

Registration Page

User Sign Up

Login Page

Home Page

Result

11. Conclusion

12. Future Scope

List of Figures

List of Tables

Chapter 1

Introduction

In today's digital era, online content creation and publishing have become essential components of knowledge sharing, communication, and information dissemination. Blogging platforms and content management systems are widely used for publishing articles, research content, technical blogs, and opinion-based writings. However, many existing systems lack structured review mechanisms, proper role-based access control, and secure content moderation, which often results in low content quality, security issues, and inefficient collaboration among users. The Role-Based Blogging and Article Review Platform is designed to address these challenges by providing a secure, scalable, and collaborative web-based solution for structured content creation and moderated publishing. The platform enables multiple users to interact within a controlled environment, where each user is assigned a specific role such as Admin, Editor, Author, Reviewer, or Reader, with clearly defined permissions. This role-based approach ensures that users can perform only authorized actions, thereby improving system security, accountability, and operational efficiency. In this platform, authors are responsible for creating and submitting articles, editors review and approve content to maintain quality standards, and administrators manage user accounts, roles, and platform configurations. Reviewers and readers actively participate by providing comments, feedback, and ratings on published articles, which promotes engagement and continuous improvement of content. The structured workflow ensures that articles are reviewed before publication, reducing the risk of misinformation and enhancing the credibility of the platform. To achieve modularity and scalability, the application follows a hybrid backend architecture. Authentication-related functionalities such as user registration, login, and logout are handled by a dedicated Java-based authentication service, ensuring secure access to the system. All core business logic, including article management, review workflows, role authorization, and content moderation, is implemented using ASP.NET Core Web API. This separation of concerns

improves maintainability, allows independent updates to system components, and enhances overall performance.

Basic Concept

The Role-Based Blogging and Article Review Platform is designed to provide a secure and structured environment for content creation, review, and publication using role-based access control. The core concept of the system is to divide responsibilities among different user roles such as Admin, Editor, Author, Reviewer, and Reader, ensuring that each user can perform only the actions permitted by their role. Authors can create and submit articles, editors review and approve content before publication, and administrators manage users and platform settings. Reviewers and readers can interact with published articles through comments and ratings, promoting quality feedback and engagement. The system uses a modern web-based architecture with a React.js frontend and an ASP.NET Core Web API backend, ensuring efficient communication, scalability, and security. This approach improves content quality, maintains publishing integrity, and enables collaborative content management in an organized and controlled manner.

Application

1. Educational blogging and research portals
2. Online article publishing platforms

Chapter 2

Problem Statement and Scope

Problem Statement

In the current digital environment, many blogging and content publishing platforms allow users to publish content with minimal control over quality, security, and accountability. Most existing systems lack a structured review and approval process, leading to the publication of unverified or low-quality content. Additionally, inadequate role-based access control often results in unauthorized actions, poor content moderation, and inefficient collaboration among users such as authors, reviewers, and editors..

Features

The system provides role-based access control, secure authentication, article management, review and approval workflow, user management, comments and ratings, and a responsive web interface.

Scope of the project

- The scope of the Role-Based Blogging and Article Review Platform defines the functional boundaries and capabilities of the system.
- The project focuses on providing a secure and structured web-based solution for collaborative content creation and moderated publishing

using role-based access control. Improvements and adaptations for various types of image manipulations.

Goals

The primary goal of the Role-Based Blogging and Article Review Platform is to provide a secure, organized, and efficient system for creating, reviewing, and publishing digital content using role-based access control. The project aims to streamline the entire content lifecycle by enabling authors to submit articles, editors to review and approve content, and administrators to manage users and platform settings effectively. Another important goal is to ensure content quality and integrity through a structured review process while encouraging user engagement through comments and ratings. The system is designed to be scalable, user-friendly, and secure, leveraging modern web technologies such as React.js and ASP.NET Core Web API to support collaborative content management and controlled publishing in a web-based environment..

Objective

- To develop a web-based platform that supports collaborative content creation and moderated publishing.
- To implement secure role-based access control for Admin, Editor, Authors, Reviewer and Reader only.

Technology Used

The **Role-Based Blogging and Article Review Platform** is developed using a modern web technology stack that ensures security, scalability, and efficient content management. The system follows a layered architecture with clear separation between frontend, backend services, and database.

1. React.js

React.js is used to develop the user interface of the application.

It provides a component-based architecture for building reusable and dynamic UI components.

Enables fast rendering through the Virtual DOM.

Supports role-based UI rendering (Admin, Editor, Author, Reviewer, Reader).

Communicates with the backend using RESTful APIs via HTTP requests.

2. HTML and CSS

Frontend Development. HTML and CSS are used to design the frontend of the web application. HTML structures the content, while CSS styles the interface, ensuring a user-friendly and responsive design. This combination allows users to interact with the system intuitively, enhancing the overall user experience.

3. ASP.NET Core Web API

- ASP.NET Core Web API is used for implementing all business logic and content management.
- Handles article creation, editing, approval, publishing, commenting, and rating.
- Provides role-based authorization using JWT tokens.
- Ensures high performance, security, and scalability.
- Implements RESTful APIs for communication with the frontend.

Chapter 3

Research Methodology

Steps to carry out project Work

1. Admin Workflow

- Admin logs in using JWT-based authentication
- Manages user accounts (create, update, delete).
- Assigns and modifies roles (Admin, Editor, Author, Reviewer, Reader).
- Monitors overall system activity and platform settings.
- Ensures security, access control, and smooth system operation

2. Author Workflow

- Author logs in to the system.
- Creates new articles using the article editor.
- Submits articles for review.
- Views article status (Pending, Approved, Rejected).
- Edits and resubmits articles if revisions are requested.

3. Editor Workflow

- Editor logs in with editor privileges.
- Views list of submitted articles.
- Reviews article content for quality and policy compliance.
- Approves articles for publication or rejects them with feedback.
- Publishes approved articles to the platform.

4. Reviewer Workflow

- Reviewer logs in to the system.
- Views published articles.
- Adds comments and provides ratings based on content quality.
- Helps maintain content standards through feedback

5. System Workflow (Backend Interaction)

- React.js frontend sends requests to ASP.NET Core Web API.
- JWT validates user identity and role.
- API executes business logic based on role permissions.
- Data is stored and retrieved from the MySQL database.

Scope

The scope of the Role-Based Blogging and Article Review Platform includes the design and development of a secure web-based system that supports structured content creation, review, and publishing through role-based access control. The platform enables authors to create and submit articles, editors to review and approve content, and administrators to manage users and system configurations. Readers and reviewers can access published articles and interact through comments and ratings. The system integrates JWT-based authentication for secure user access and uses ASP.NET Core Web API to handle all business logic and content management, with a React.js frontend providing a responsive user interface. The project focuses on ensuring data security, content quality, and efficient collaboration while allowing scalability for future enhancements such as advanced analytics, notification systems, and content recommendation features.

Overall Description

The Role-Based Blogging and Article Review Platform is a comprehensive web-based system designed to streamline content creation, review, and publishing in a controlled and collaborative environment. The platform follows a **role-based access control (RBAC) model**, where each user is assigned a specific role—Admin, Editor, Author, Reviewer, or Reader—with predefined permissions. This ensures that only authorized users can perform certain actions, maintaining system security and content integrity.

The system uses **JWT (JSON Web Token) authentication** to secure registration, login, and logout processes, while the **ASP.NET Core Web API** handles all business logic, including article management, reviews, comments, ratings, and role-based operations. The **React.js frontend** provides an intuitive and responsive user interface, enabling users to interact with the platform seamlessly.

Document Overview

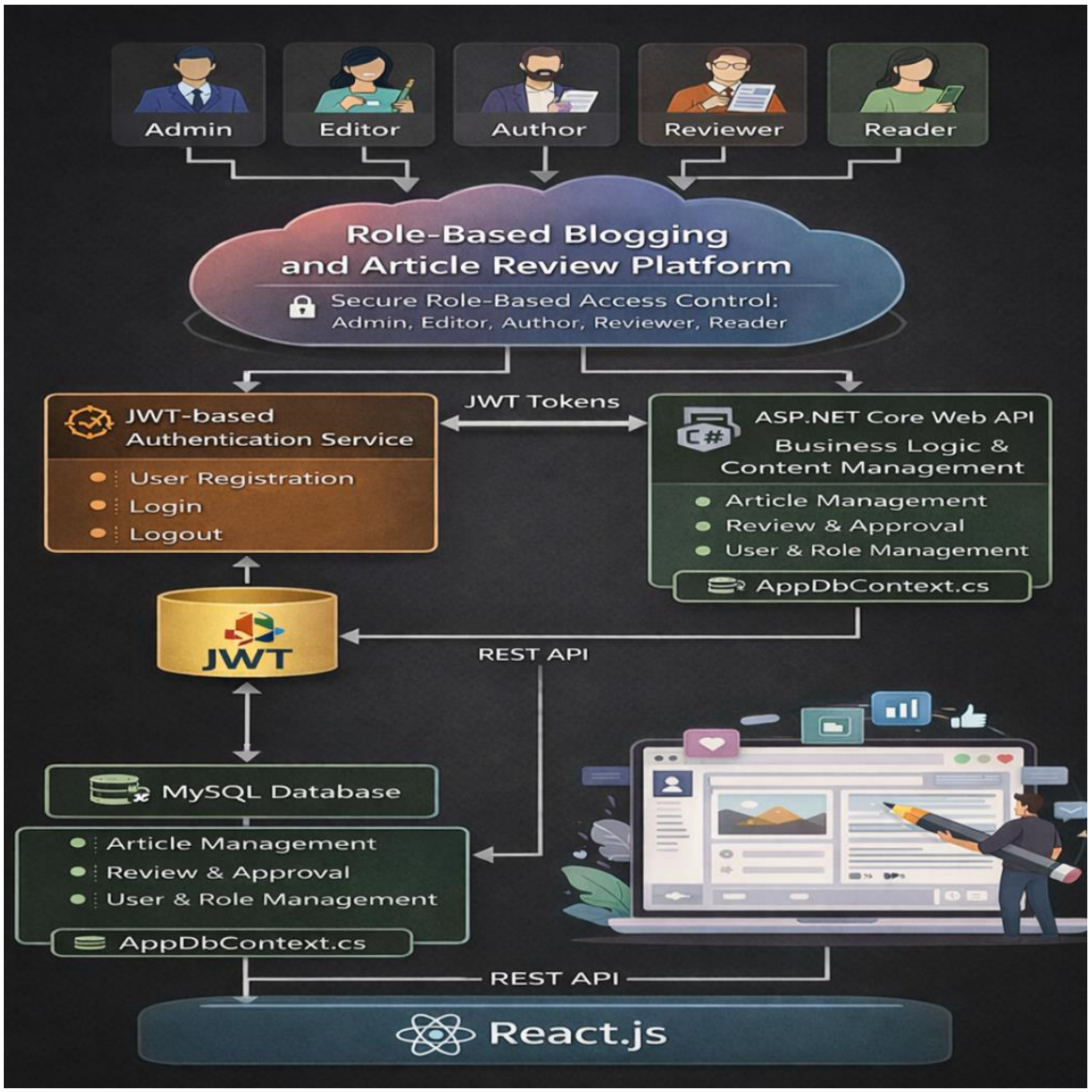
This document provides a detailed description of the **Role-Based Blogging and Article Review Platform**, including its objectives, functionality, and architecture. It presents the system's **basic concept**, **goals**, **scope**, and **applications**, providing a clear understanding of the purpose and benefits of the platform.

The document outlines the **technologies used**, such as React.js for the frontend, ASP.NET Core Web API for the backend, JWT for authentication, and MySQL for database management. It describes the **workflow**, including role-wise responsibilities of Admin, Editor, Author, Reviewer, and Reader, and explains how articles are created, reviewed, approved, and published.

Additionally, the document includes **system diagrams** (Use Case, Sequence, Activity, and Package diagrams) to visualize architecture and interactions. It also details the **implementation steps, testing methodology, and deployment process**, providing a complete guide for understanding, developing, and evaluating the project.

This overview serves as a **reference for developers, evaluators, and users**, ensuring clarity in system design, functionality, and operational workflow.

Chapter 4



SYSTEM ARCHITECTURE

Chapter 5

Software Requirements

- Operating system : Windows 7 Onwards
    - Technology : ASP.NET Core
- IDE : Microsoft Visual Studio code.
    - Front End: HTML,CSS, Bootstrap and React.js
    - Database : MySQL

Hardware Requirements

- Processor : Intel I3/I5 Processor.
- Hard Disk : Min 500GB.
- RAM : 8GB

Use Case Diagram

The use case diagram illustrates the role-based interactions within the blogging and article review platform. Each user role—Admin, Editor, Author, Reviewer, and Reader—has specific responsibilities. Admins manage users and platform settings, Authors submit articles, Editors edit and approve content, Reviewers provide feedback through article reviews, and Readers access and engage with published content. The diagram highlights how each role contributes to the workflow of article creation, review, approval, and publication.

Figure 6.5: Usecase

Package Diagram

The diagram illustrates the system architecture of the Role-Based Blogging and Article Review Platform. The Auth Service (ASP.NET Core) handles secure authentication using JWT, while the Controllers & Services layer manages API requests from the React.js frontend. Business logic processes article management, reviews, and ratings, which are handled by the Content Management layer. The architecture ensures secure, scalable, and organized communication between frontend and backend components.

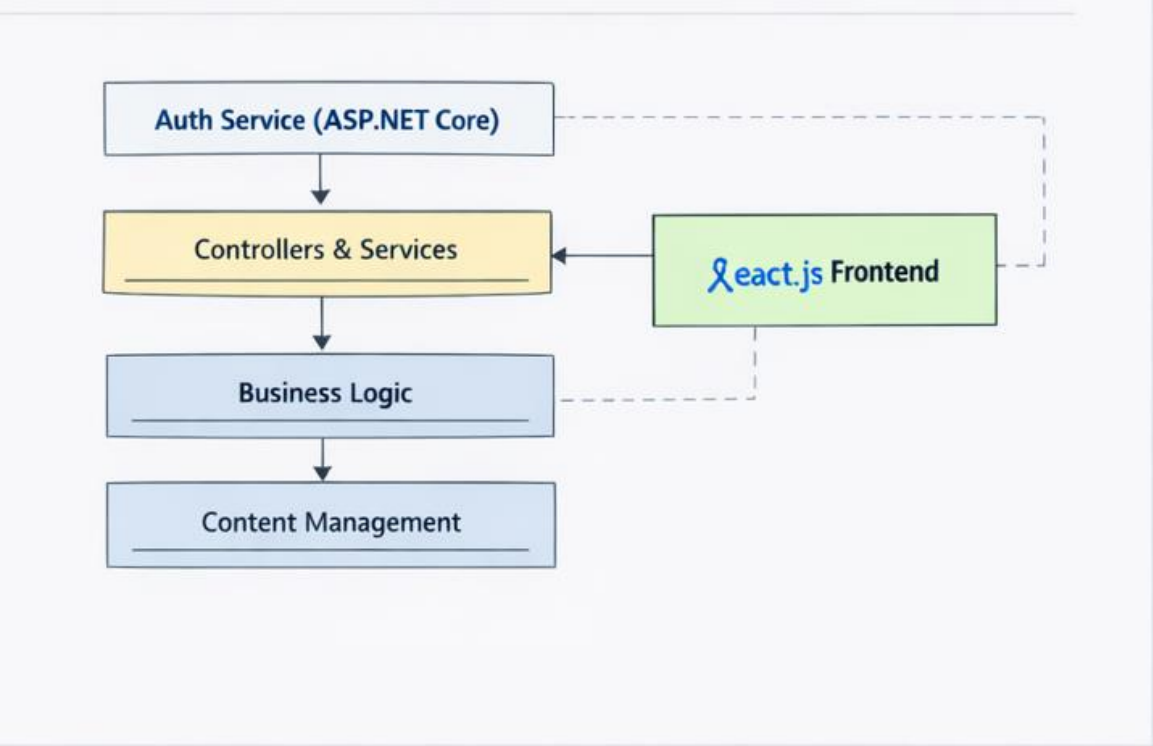


Figure 6.7: Package Diagram

Sequence Diagram

The sequence diagram depicts the flow of interactions in a hybrid deep learning model for detecting phony faces using CNNs. The user starts the procedure by uploading a facial image. The system then pre-processes the image, extracts features with CNNs, and passes them to the hybrid module. The hybrid module uses the retrieved features to determine if the face is real or false. Finally, the classification results are displayed on the user interface.

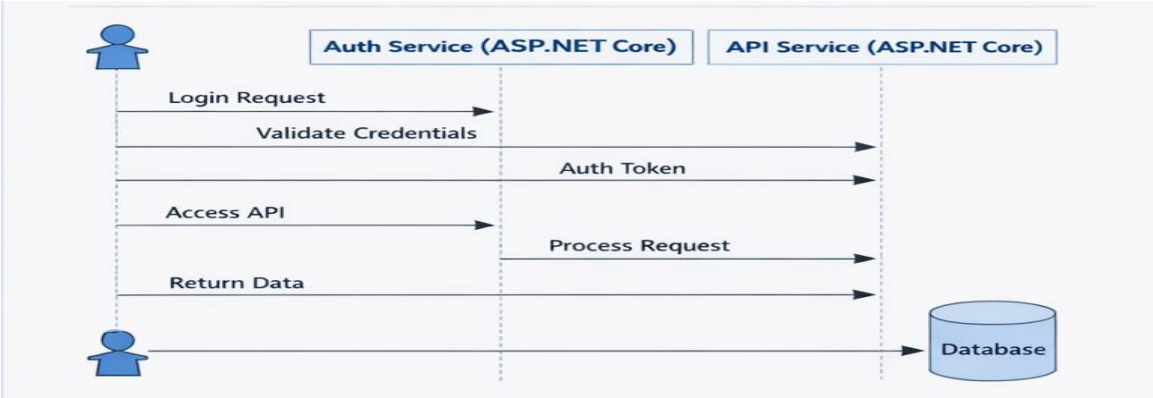


Figure 6.8: Sequence diagram

Activity Diagram

The activity diagram depicts the process of a hybrid deep learning model for fake face detection with CNNs. Users begin by logging onto the system. They then input a facial image, which goes through pre-processing to verify quality. If the image quality is satisfactory, features are retrieved with CNNs and fed into the hybrid model for classification. The categorization result, which indicates whether the face is real or artificial, is displayed to the user and saved in a database for future reference.

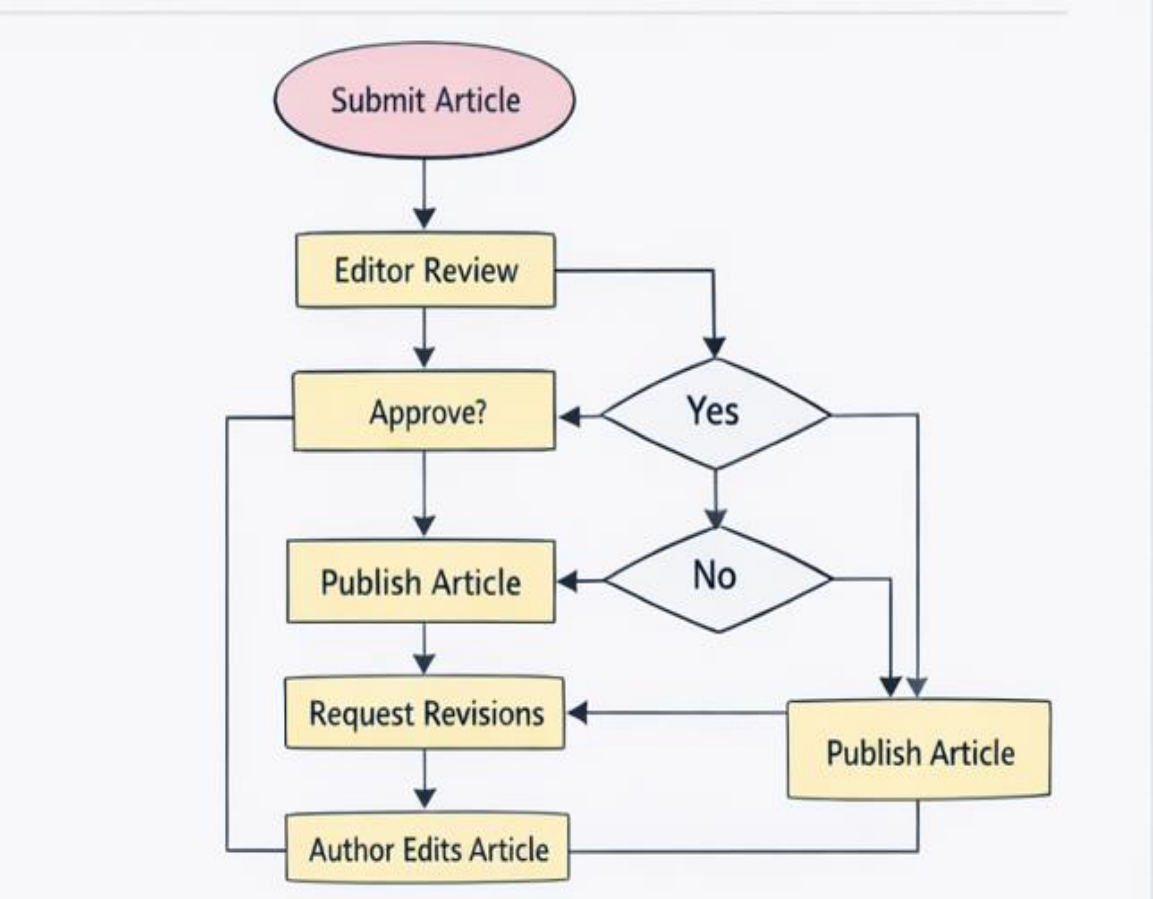


Figure 6.9: Activity Diagram

Chapter 6

Test Specification

Implementation

The implementation of test specification defines the testing strategy used to verify that the **Role-Based Blogging and Article Review Platform** functions correctly according to the specified requirements. The testing process ensures system reliability, security, correctness, and performance across different user roles.
Testing was carried out at multiple levels to validate individual components as well as the complete system. Each test case was designed with clear **test objectives, input conditions, expected results, and actual outcomes** to ensure traceability and accuracy.

Types Of Testing

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is

a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose.
It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully.

No defects encountered.

Chapter 7

Data Table and Discussion

Discussion

Data Tables

1. Registration Table

Field Name	Data Type	Description
user id	INT (Primary Key)	Unique identifier for user
full _name	VARCHAR(255)	User's full name
email	VARCHAR(255) (Unique)	User's email
password	VARCHAR(255)	Hashed password

Table 1: Registration Table Structure

2. Login Table

Field Name	Data Type	Description
user _id	INT (Primary Key)	Unique identifier for user
email	VARCHAR(255)	User email for login
password	VARCHAR(255)	Hashed password

Table 2: Login Table Structure

Chapter 8

Results and Discussion

Analysis

The **Role-Based Blogging and Article Review Platform** was successfully designed, implemented, and tested according to the specified requirements. All core functionalities of the system operated as expected during testing, demonstrating the effectiveness of the proposed architecture and role-based access control mechanism.

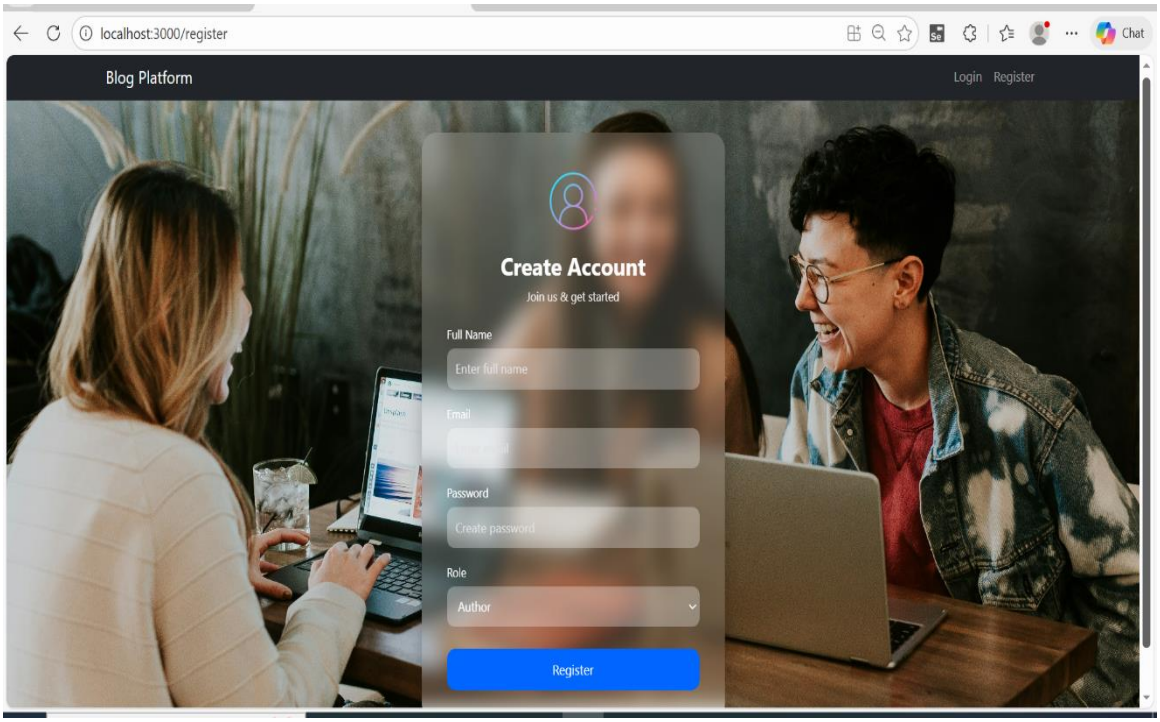
The system correctly enforced **role-based permissions**, ensuring that users could access only the features assigned to their respective roles. Authors were able to create, edit, and submit articles, while editors could review, approve, or reject submitted content. Administrators successfully managed users, roles, and platform settings. Reviewers and readers interacted with published articles through comments and ratings without compromising system security.

Integration between the frontend and backend services was smooth, with REST APIs providing reliable data exchange and acceptable response times. Authentication and authorization mechanisms functioned correctly, preventing unauthorized access and ensuring secure user sessions. The React.js frontend delivered a responsive and user-friendly interface, improving overall usability.

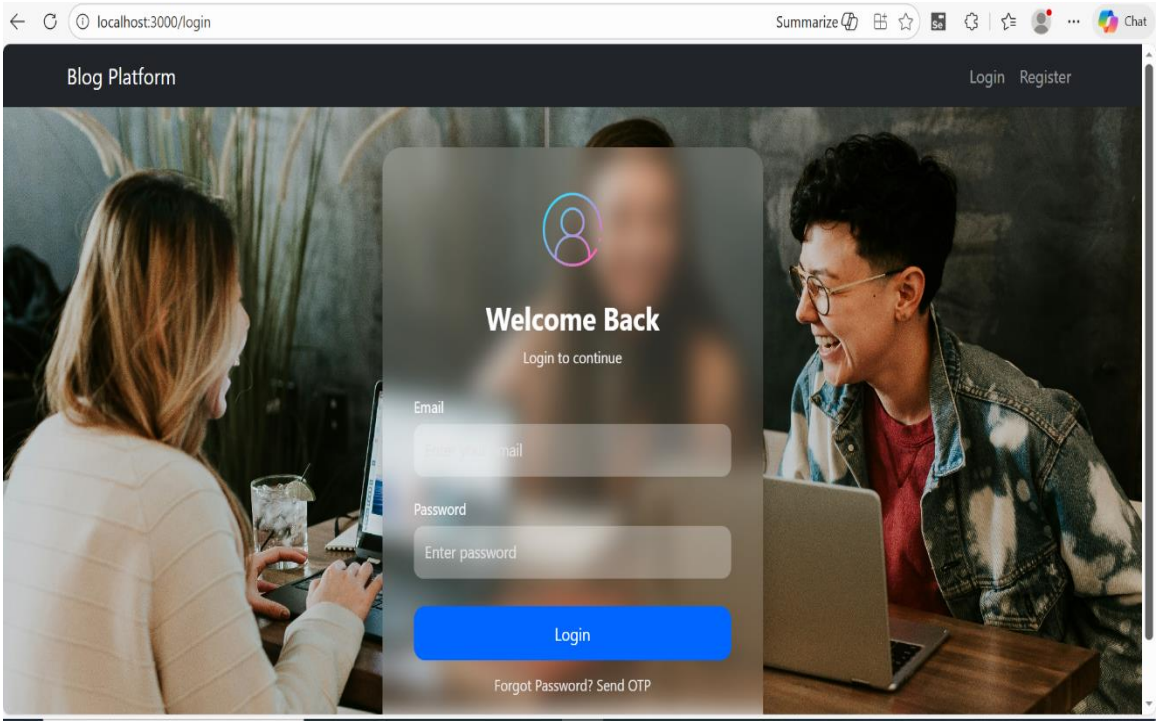
Performance testing showed that the system handled normal user loads efficiently, with minimal delays in page loading and API responses. Security testing confirmed that sensitive operations were protected and invalid access attempts were appropriately restricted.

Overall, the results indicate that the system meets its functional and non-functional requirements.

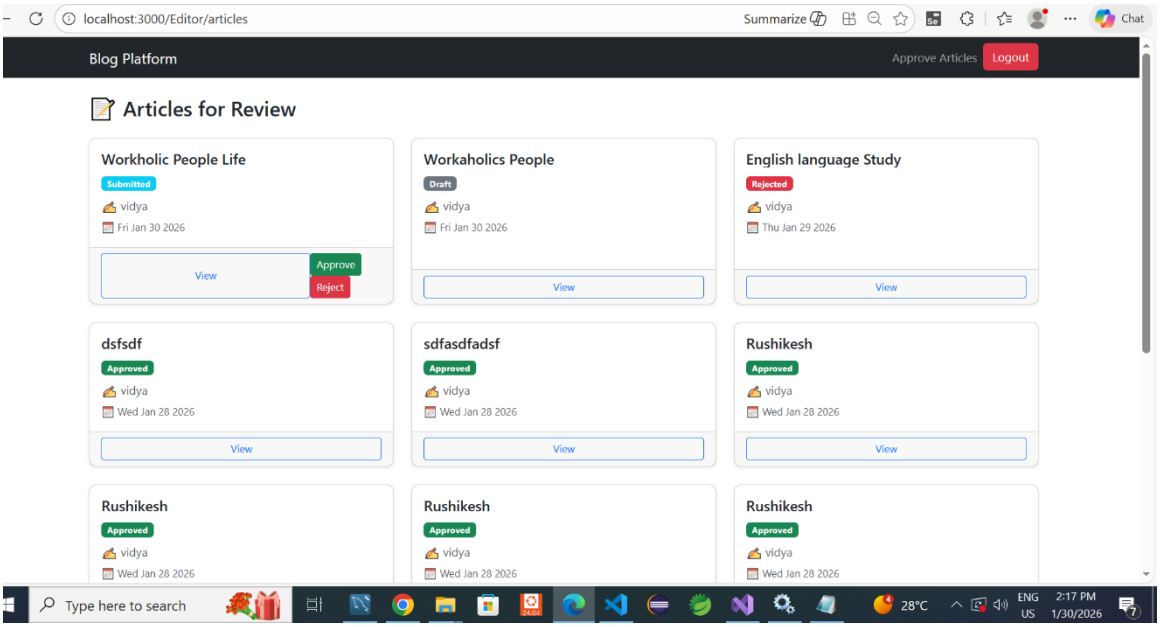
1. User Registration



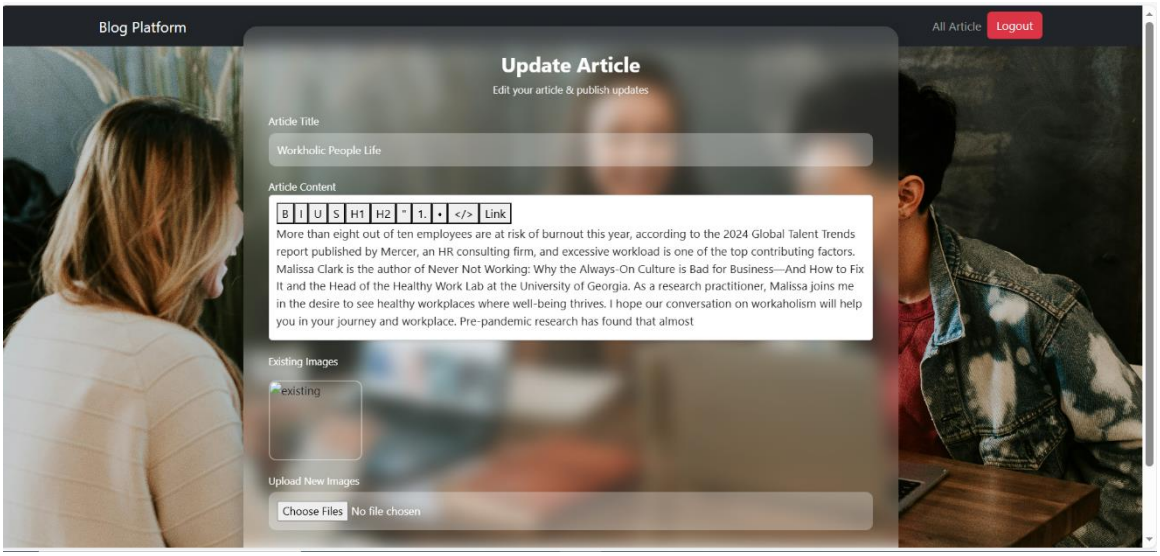
2. Login Page



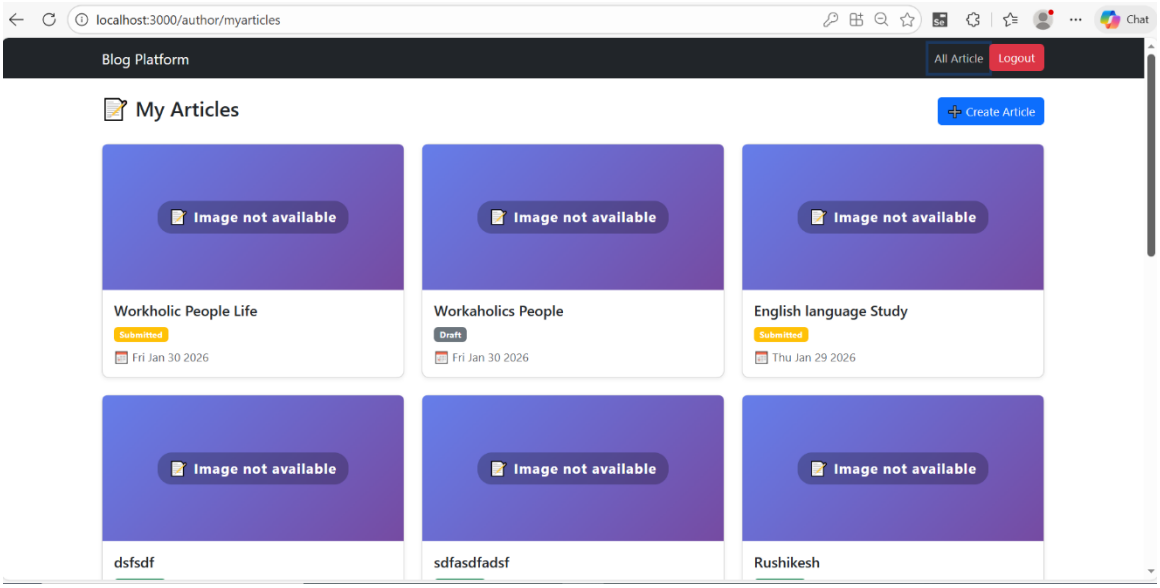
Review Page

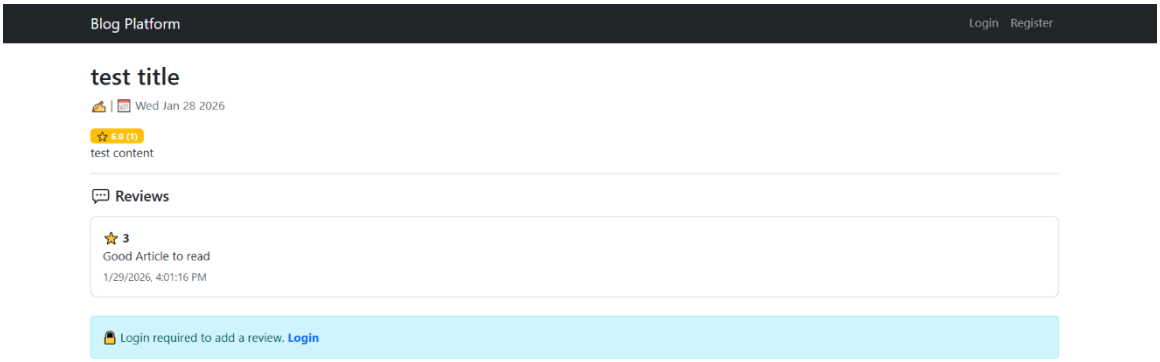

localhost:3000/Editor/articles
Summarize
Chat
Blog Platform
Approve Articles
Logout
Articles for Review
Workholic People Life
Submitted
vidya
Fri Jan 30 2026
View
Approve
Reject
Workaholics People
Draft
vidya
Fri Jan 30 2026
View
English language Study
Rejected
vidya
Thu Jan 29 2026
View
dsfsdf
Approved
vidya
Wed Jan 28 2026
View
sdfasdfadsf
Approved
vidya
Wed Jan 28 2026
View
Rushikesh
Approved
vidya
Wed Jan 28 2026
View
Rushikesh
Approved
vidya
Wed Jan 28 2026
Rushikesh
Approved
vidya
Wed Jan 28 2026
Rushikesh
Approved
vidya
Wed Jan 28 2026
Type here to search
28°C
ENG
US
2:17 PM
1/30/2026

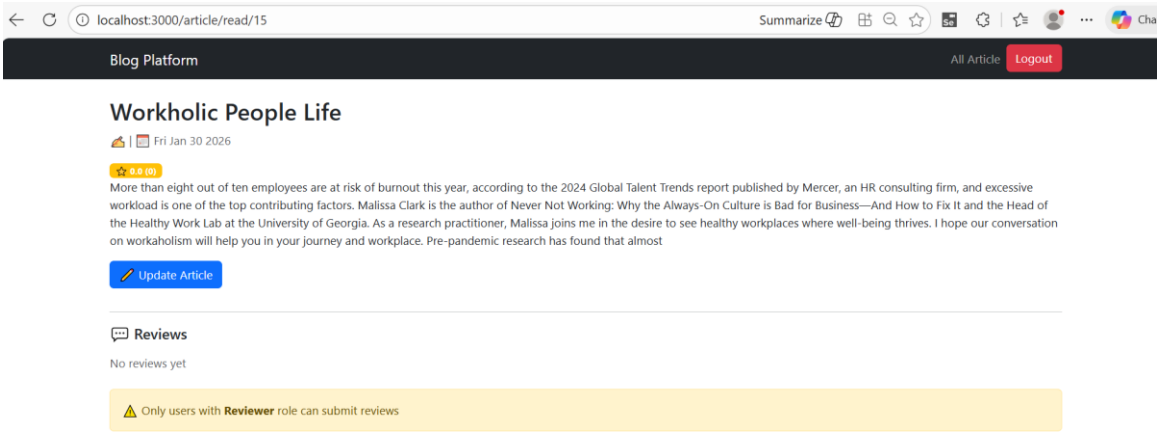
Update Article Page

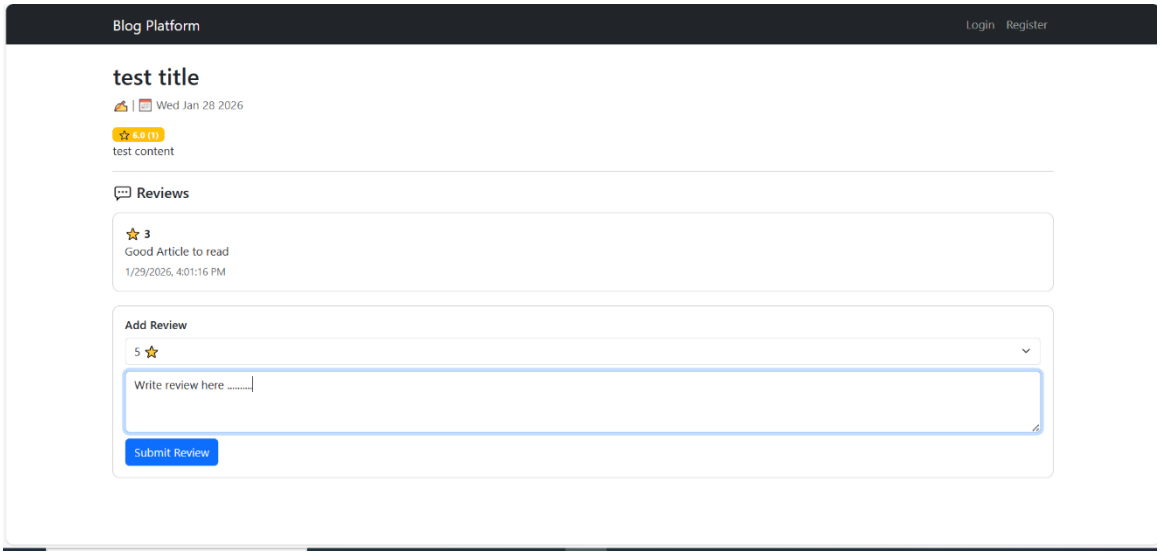


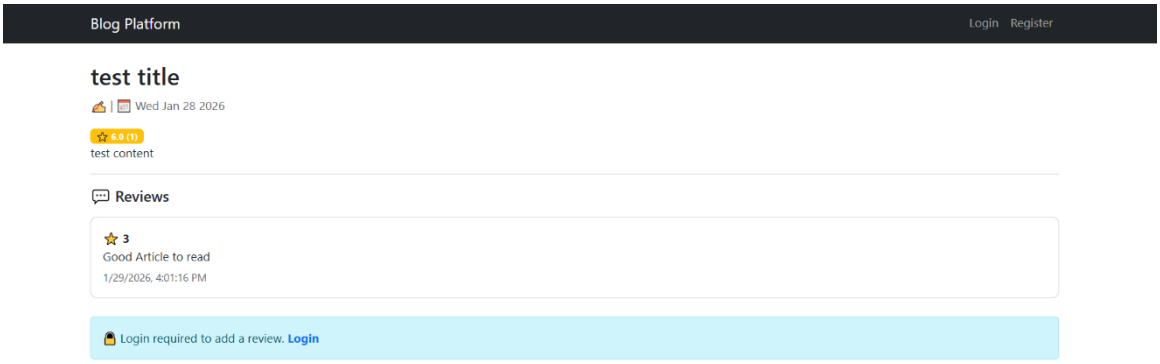
Result




Blog Platform
Login
Register
test title
Wed Jan 28 2026
6.0 (1)
test content
Reviews
3
Good Article to read
1/29/2026, 4:01:16 PM
Login required to add a review. Login


localhost:3000/article/read/15
Summarize
Blog Platform
All Article
Logout
Workholic People Life
Fri Jan 30 2026
0.0 (0)
More than eight out of ten employees are at risk of burnout this year, according to the 2024 Global Talent Trends report published by Mercer, an HR consulting firm, and excessive workload is one of the top contributing factors. Malissa Clark is the author of Never Not Working: Why the Always-On Culture is Bad for Business—And How to Fix It and the Head of the Healthy Work Lab at the University of Georgia. As a research practitioner, Malissa joins me in the desire to see healthy workplaces where well-being thrives. I hope our conversation on workaholism will help you in your journey and workplace. Pre-pandemic research has found that almost
Update Article
Reviews
No reviews yet
Only users with Reviewer role can submit reviews


Blog Platform
Login
Register
test title
Wed Jan 28 2026
5.0 (1)
test content
Reviews
3
Good Article to read
1/29/2026, 4:01:16 PM
Add Review
5
Write review here .....
Submit Review


Blog Platform
Login
Register
test title
Wed Jan 28 2026
5.0 (1)
test content
Reviews
3
Good Article to read
1/29/2026, 4:01:16 PM
Login required to add a review. Login

Chapter 10

Conclusion

The Role-Based Blogging and Article Review Platform was successfully designed and implemented to provide a secure and structured environment for collaborative content creation and moderated publishing. The system effectively enforces role-based access control, ensuring that each user operates within clearly defined permissions and responsibilities.

The platform supports a complete article lifecycle, including content creation, review, approval, and publication, which improves content quality and accountability. Testing results confirm that the system meets all functional requirements, offers reliable performance, and ensures secure user interactions. Overall, the project delivers a scalable and user-friendly solution suitable for real-world blogging and content management applications.

Future scope

The future scope of the Role-Based Blogging and Article Review Platform includes several enhancements to improve functionality, scalability, and user experience. Advanced features such as AI-based article recommendations and automated content moderation can be integrated to enhance content quality and personalization. Real-time notifications for article status updates, comments, and reviews can be added to improve user engagement. The system can be extended to support mobile applications and multi-language content to reach a wider audience. Additionally, deployment on cloud platforms with advanced analytics, enhanced security mechanisms, and scalability support can make the platform suitable for large-scale, real-world publishing environments.