

Mini Blog – Project Report

1. Introduction

The **Mini Blog** is a web-based blogging platform built using the Django framework. This project allows users to create, manage, and interact with blog posts while providing a simple and user-friendly interface. The application ensures proper user authentication, CRUD operations, and database integration, making it a complete blogging system.

2. Minimum Requirements Implementation

2.1 User Authentication

- Users can register, log in, and log out securely.
- Django's built-in authentication system is used for secure password storage.
- Session management ensures only authenticated users can create or manage posts.

2.2 Post Management (CRUD Operations)

- Users can Create, Read, Update, and Delete their blog posts.
- Only the author of a post can edit or delete it.
- Posts are displayed in a list format with details such as the title, content, and date published.

2.3 Database Integration

- SQLite is used as the default database to store user details and blog posts.
- Django's ORM (Object-Relational Mapping) handles all database interactions.

2.4 Templates & Styling

- The frontend is designed using Bootstrap for a clean and responsive layout.
- Navigation bar includes links for Home, Login, Register, and Logout.

2.5 Admin Panel

- Django's admin interface is enabled for managing blog posts.
- Admins can view, edit, and delete all posts, ensuring content moderation.

3. Custom Features & Enhancements

Beyond the minimum requirements, I have implemented two additional features:

3.1 Comments Section for Blog Posts

- Users can leave comments on blog posts to engage in discussions.

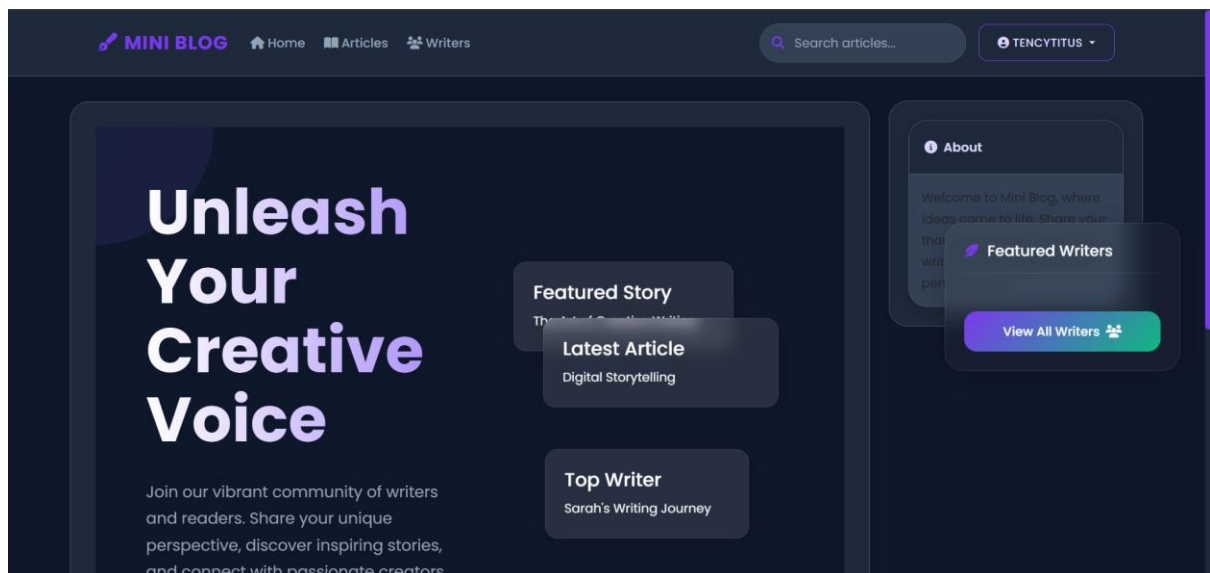
- Comments are linked to users, showing who posted them and when.
- A comment moderation system allows post authors to delete inappropriate comments.

3.2 Like/Dislike System

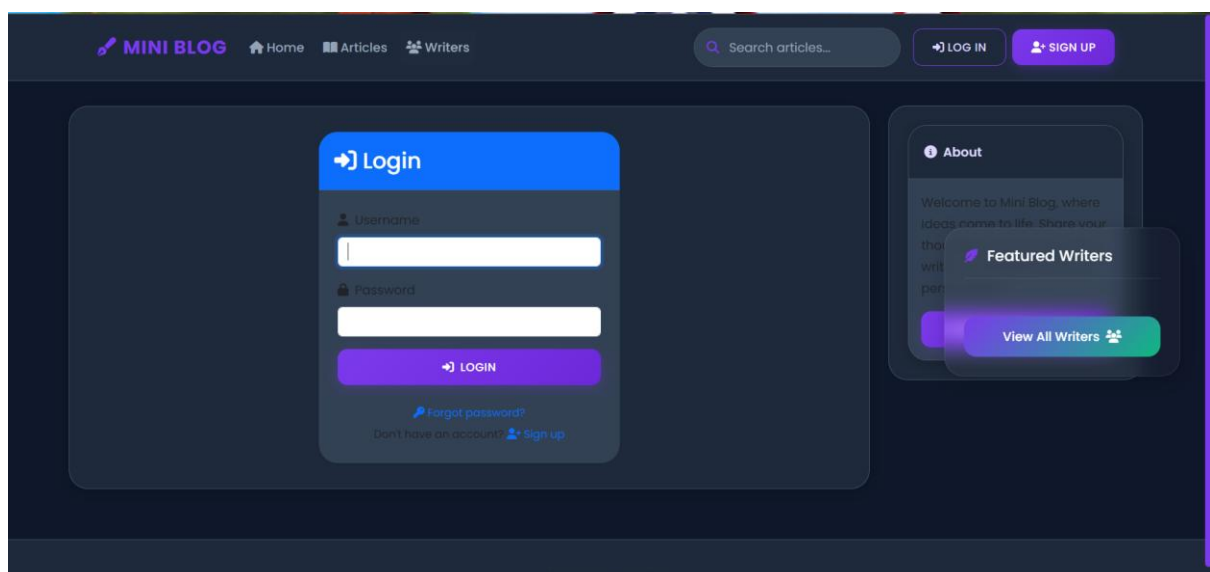
- Users can like or dislike blog posts, adding engagement features.
- Each post displays the total number of likes and dislikes.
- Users can toggle their reaction (e.g., change a like to a dislike).

4. Screenshots

4.1 Home Page



4.2 User Authentication (Login & Register)



MINI BLOG

HomeArticlesWriters

Search articles...

LOG INSIGN UP

Create an Account

Username

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email Address

First Name

Last Name

OptionalOptional

Password

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.

About

Welcome to Mini Blog, where ideas come to life. Share your thoughts, connect with fellow writers.

Featured Writers

View All Writers

4.3 Blog Posts

MINI BLOG

HomeArticlesWriters

Search articles...

TENCYTITUS

Blog Posts

Explore our collection of insightful articles written by our community.

WRITE NEW POST

CCodeMaster

March 25, 2025 • 3 comments

Frontend Development with Modern Tools

Modern frontend development involves various tools and frameworks. Understanding HTML5, CSS3, and JavaScript is essential. Popular

WWebDev

March 25, 2025 • 3 comments

Understanding Database Design

Good database design is crucial for building scalable applications. This includes proper normalization, indexing, and understanding

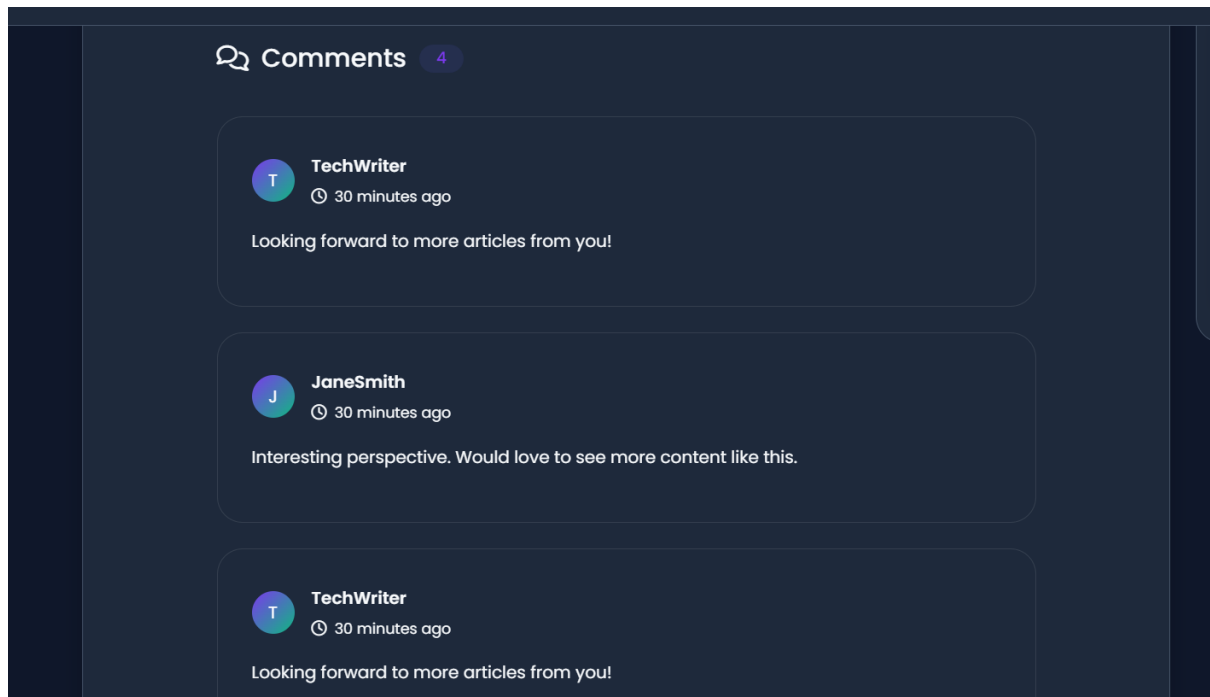
About

Welcome to Mini Blog, where ideas come to life. Share your thoughts, connect with fellow writers, and explore diverse perspectives.

Featured Writers

View All Writers

4.4 Comments Section



4.5 Blog Writers



5. Challenges & Learnings

5.1 User Authentication Issues

- **Challenge:** Preventing users from accessing posts without logging in.
- **Solution:** Used Django's login required decorator and session management.

5.2 Database Migration Issues

- **Challenge:** Changes in models required frequent migrations.
- **Solution:** Used python manage.py makemigrations and migrate commands carefully.

5.3 Handling AJAX for Like/Dislike

- **Challenge:** Making likes/dislikes update in real time without reloading the page.
- **Solution:** Implemented AJAX requests to handle likes/dislikes dynamically.

6. Conclusion

This project provided hands-on experience with Django, database management, and user authentication. By adding comments and a like/dislike system, the blog is now more interactive. Future improvements could include user profiles with avatars, post categories, and a search bar to enhance the user experience further.

7. Submission Links

- **GitHub Repository:** <https://github.com/TencyTitus/mini-blog.git>