DIY Django Mini Blog

GitHub Repo - https://github.com/A-N-I-T-T-A/DIY_Blog.git

Live on Render - https://diy-blog.onrender.com

Features Implemented

Minimum Requirements Implemented

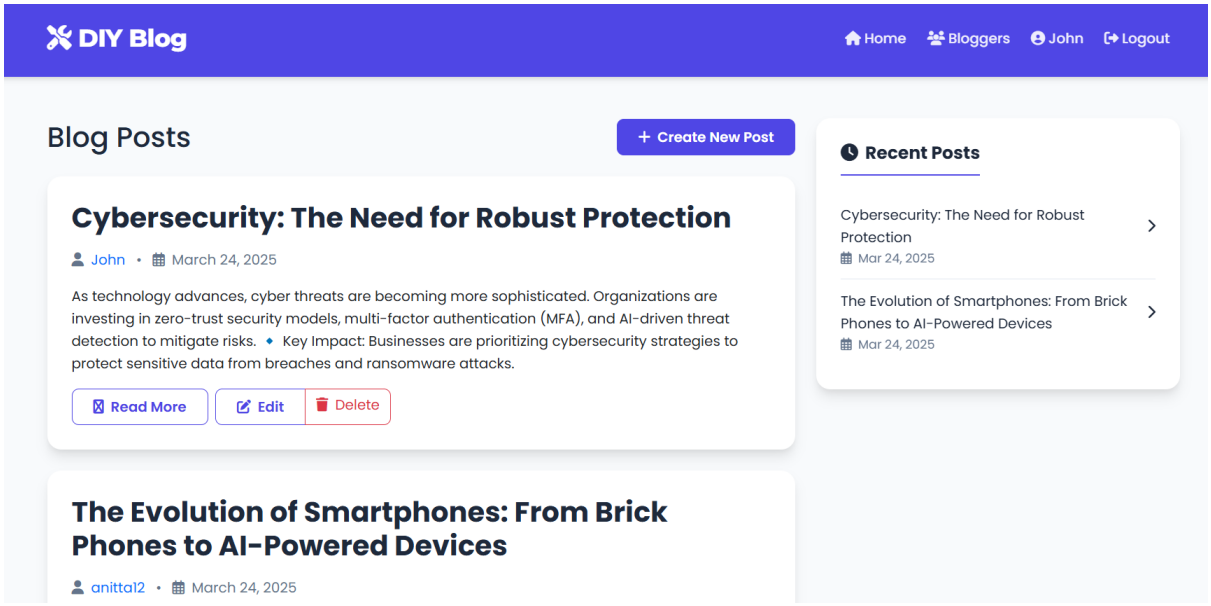
- User Authentication – Registration, Login, Logout, and Password Hashing
- Blog Post Management – CRUD (Create, Read, Update, Delete) operations
- Database Integration – Used SQLite to store posts and user data
- Templates & Styling – Applied Bootstrap for a responsive UI
- Admin Panel – Django Admin Interface for post management

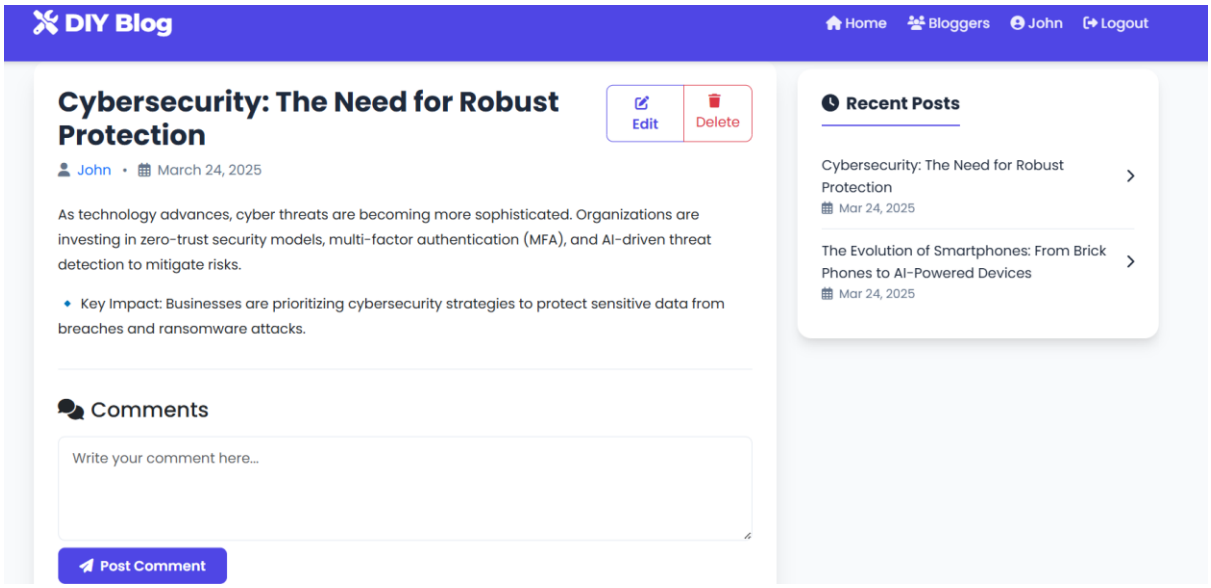
Customization & Enhancements:

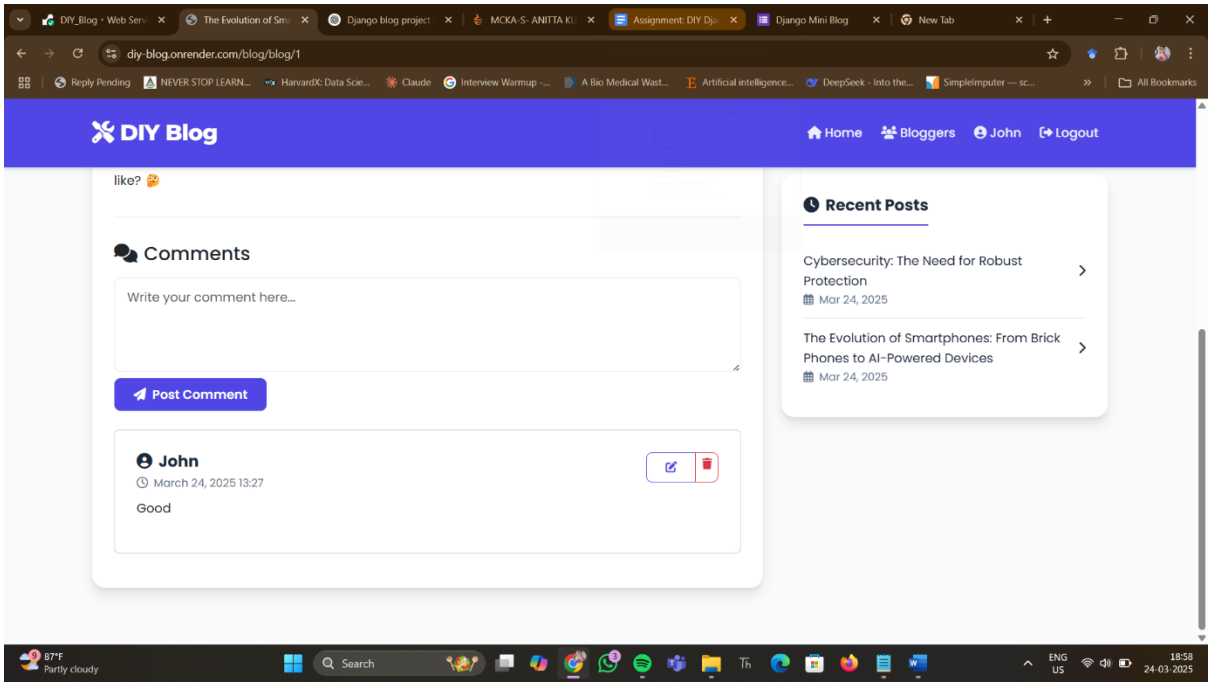
In addition to the above, I included two custom features:

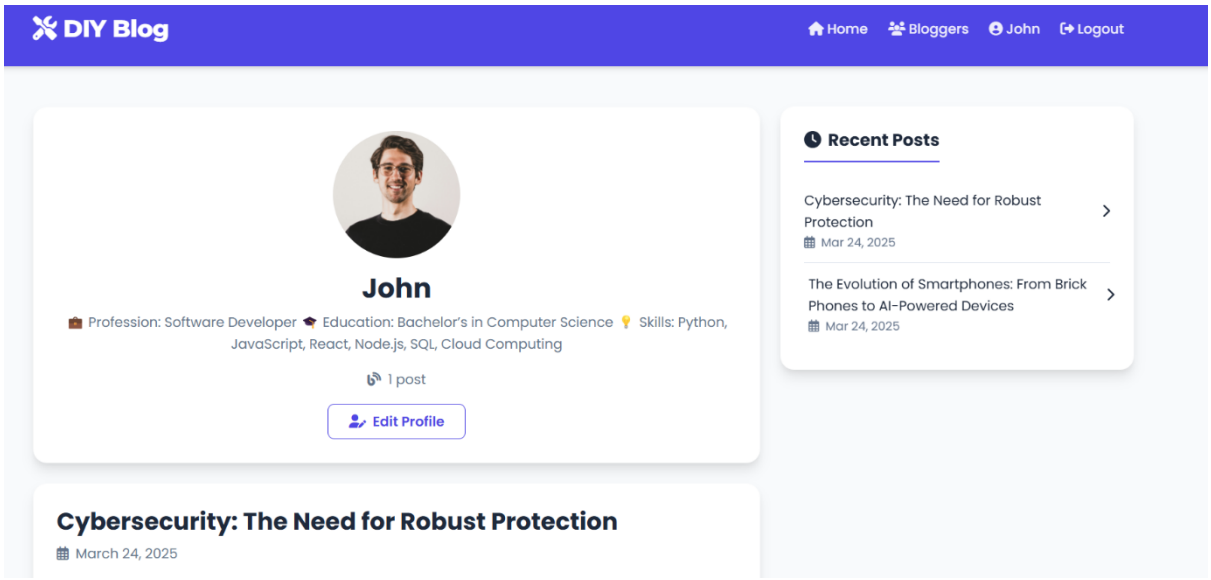
1. **Comments Section for Blog Posts**
   - Users can add, edit, and delete comments on blog posts.
   - Only logged-in users can comment.
   - Users can only edit or delete their own comments.
   - The comments are displayed below each blog post.
2. **User Profiles with Bio and Profile Pictures**
   - Each user has a profile page with a bio and profile picture.
   - Users can update their bio and upload a profile picture.
   - Django's MEDIA settings were used to handle image uploads.
3. **Recent Posts Section**
   - A "Recent Posts" section is displayed on the homepage, showcasing the latest blog posts for easy navigation.
   - This helps users quickly access the newest content without searching through the entire blog.

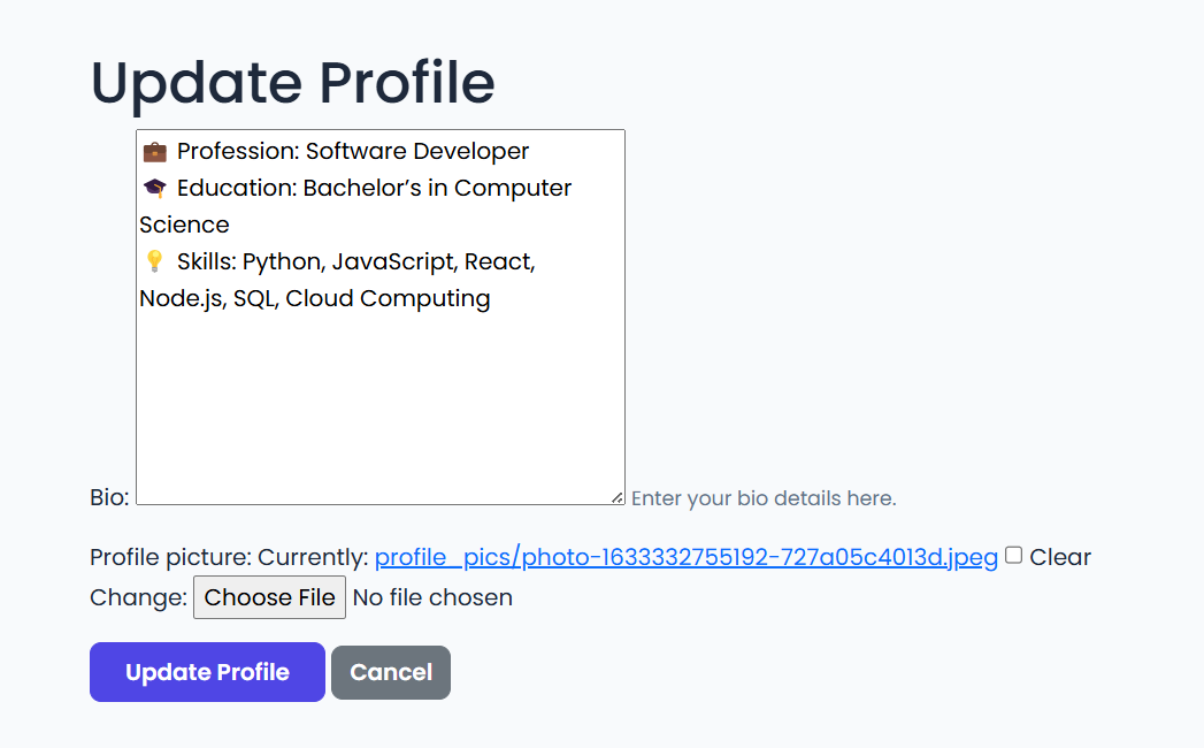
Screenshots:






DIY Blog
Home
Bloggers
John
Logout
like?
Comments
Write your comment here...
Post Comment
John
March 24, 2025 13:27
Good
Recent Posts
Cybersecurity: The Need for Robust Protection
Mar 24, 2025
The Evolution of Smartphones: From Brick Phones to AI-Powered Devices
Mar 24, 2025


DIY Blog
Home
Bloggers
John
Logout
John
Profession: Software Developer Education: Bachelor's in Computer Science Skills: Python, JavaScript, React, Node.js, SQL, Cloud Computing
1 post
Edit Profile
Recent Posts
Cybersecurity: The Need for Robust Protection
Mar 24, 2025
The Evolution of Smartphones: From Brick Phones to AI-Powered Devices
Mar 24, 2025
Cybersecurity: The Need for Robust Protection
March 24, 2025



Challenges & Solutions

1. Handling User Authentication & Access Control

Challenge: Ensuring users could securely register, log in, and manage their own content.

Solution: Used Django's built-in authentication system, including LoginRequiredMixin and UserPassesTestMixin for permissions.

2. Keeping the 'Recent Posts' Section Updated

Challenge: Ensuring the recent posts were dynamically displayed in real time.

Solution: Used Django's QuerySet method .order_by('-created_at')[:5] to retrieve and display the latest posts efficiently.

3. Using Cursor AI for Development

Challenge: Adapting to AI-assisted coding and debugging unexpected issues in auto-generated code.

Solution: Carefully reviewed AI-generated suggestions and made manual adjustments when necessary.

4. Implementing Comments with Edit/Delete Options

Challenge: Allowing users to edit and delete their own comments without affecting others.

Solution: Used ForeignKey relationships to link comments to users and posts, ensuring only the author could modify their comments.