Django Blog Web Application – Project Documentation

Project Title: My Personal Blog WebApp

Tech Stack:

Backend: Python, Django Framework

Frontend: HTML, CSS, Bootstrap

Database: SQLite (default)

Admin Interface: Django Admin

---

1. Project Overview

This Django Blog Web Application is a full-stack web project built to manage and share blog content dynamically. It allows users to view blog posts, while admins can log in and create, edit, or delete posts from a secure backend dashboard. The project focuses on CRUD operations and Django's powerful admin interface.

---

2. Features Implemented

1. Public Features:

View All Blog Posts – Homepage displays a list of blog entries with title and summary.

Single Post View – Clicking a title leads to a full detailed blog post.

Post Categorization *(if added)* – Organize posts by category or tag.

Automatic Timestamps – Created/updated dates shown with each post.

2. Admin-Only Features:

1. ✚ **Create New Blog Post** – With rich-text body, title, date.

2. **Edit Post** – Update content, change titles or fix typos.

3. ✖ **Delete Post** – Remove outdated or incorrect posts.

4. **Admin Dashboard** – Secure backend login for managing posts and users.

3. Project Structure (Important Folders & Files)

```
myblog/
├── blog/              # Main app
│   ├── migrations/
│   ├── templates/
│   │   └── blog/
│   │       ├── index.html
│   │       ├── post_detail.html
│   ├── admin.py
│   ├── models.py
│   ├── views.py
│   ├── urls.py
│   └── forms.py        # (if forms are used)
├── myblog/            # Project config
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── db.sqlite3
└── manage.py
```

4. Models and Database

A simple Post model was created in models.py:

```python
CopyEdit
class Post(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.title
```

5. URL Routing & Views

URLs are managed in blog/urls.py, then included in the main myblog/urls.py.
Each view corresponds to a page (Homepage, Post Detail, Admin actions).

```python
CopyEdit
urlpatterns = [
    path('', views.index, name='index'),
```

```
    path('post/<int:id>/', views.post_detail, name='post_detail'),
]
```

6. Templates

Templates were built using **Django Template Language (DTL)** to dynamically load post data:

- index.html: Loops through all posts and shows summaries.
- post_detail.html: Displays a full post when clicked.
- Optional: base.html can be used for DRY layout structure.

---

7. Admin Interface

The built-in **Django Admin Panel** is used for managing posts:

- Superuser account created via createsuperuser command.
- Post model registered for admin control.
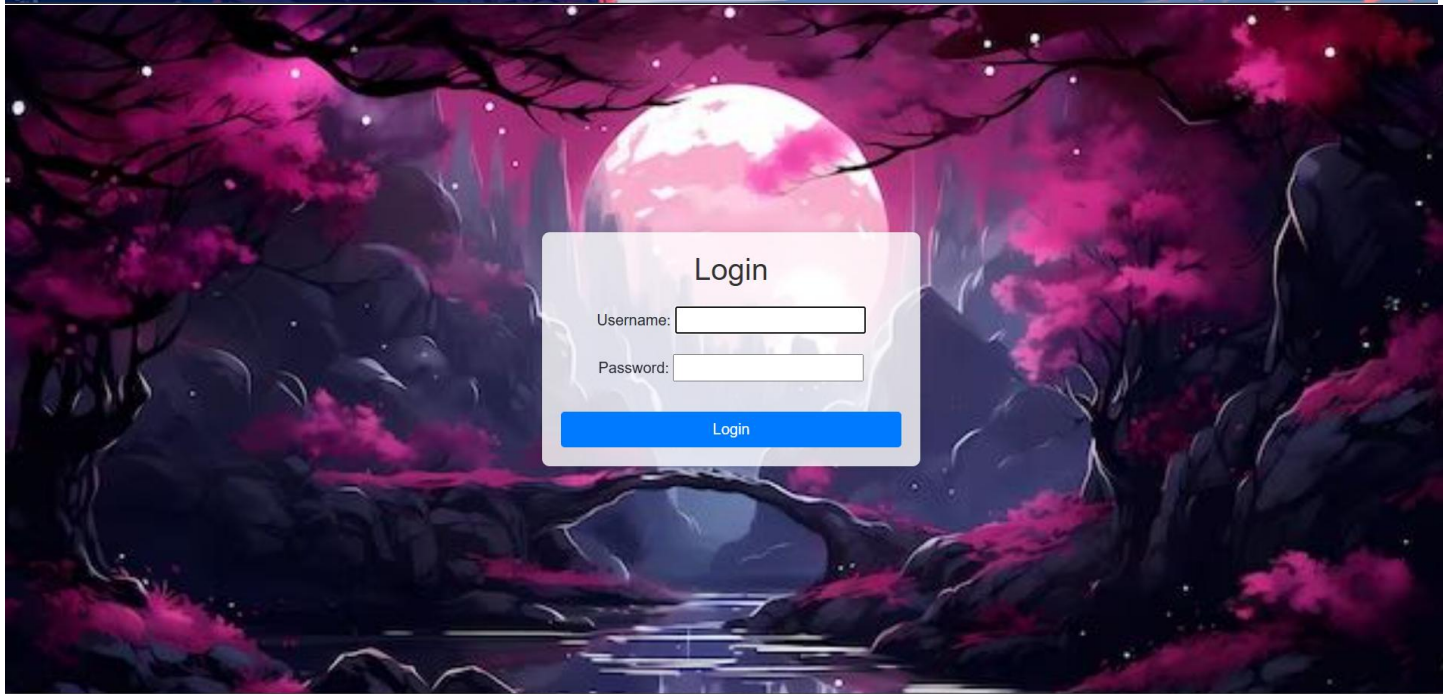- Easy to manage content without direct DB interaction.

---

8. Future Enhancements

- User Authentication System (Sign up, login, comment)
- Add comment system under each post
- Media uploads (e.g., blog thumbnails).
- Search functionality
- Deploy on platforms like Heroku or Render

---

9. Conclusion

This project helped in understanding the MVC (Model-View-Controller) pattern in Django, implementing backend logic, rendering dynamic HTML content, and managing databases using the ORM. It is a solid foundation for building more advanced Django apps.

Below are the images :

Register

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

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.
- Your password can't be entirely numeric.

Password confirmation: [   ] Enter the same password as before, for verification.

[Register]

Embark on creating your perfect, engaging blog.

SIGN UP

Django administration ◑

Username:
[   ]

Password:
[   ]

[Log in]

[Logout]

User Dashboard

[View Posts]

[Post New]

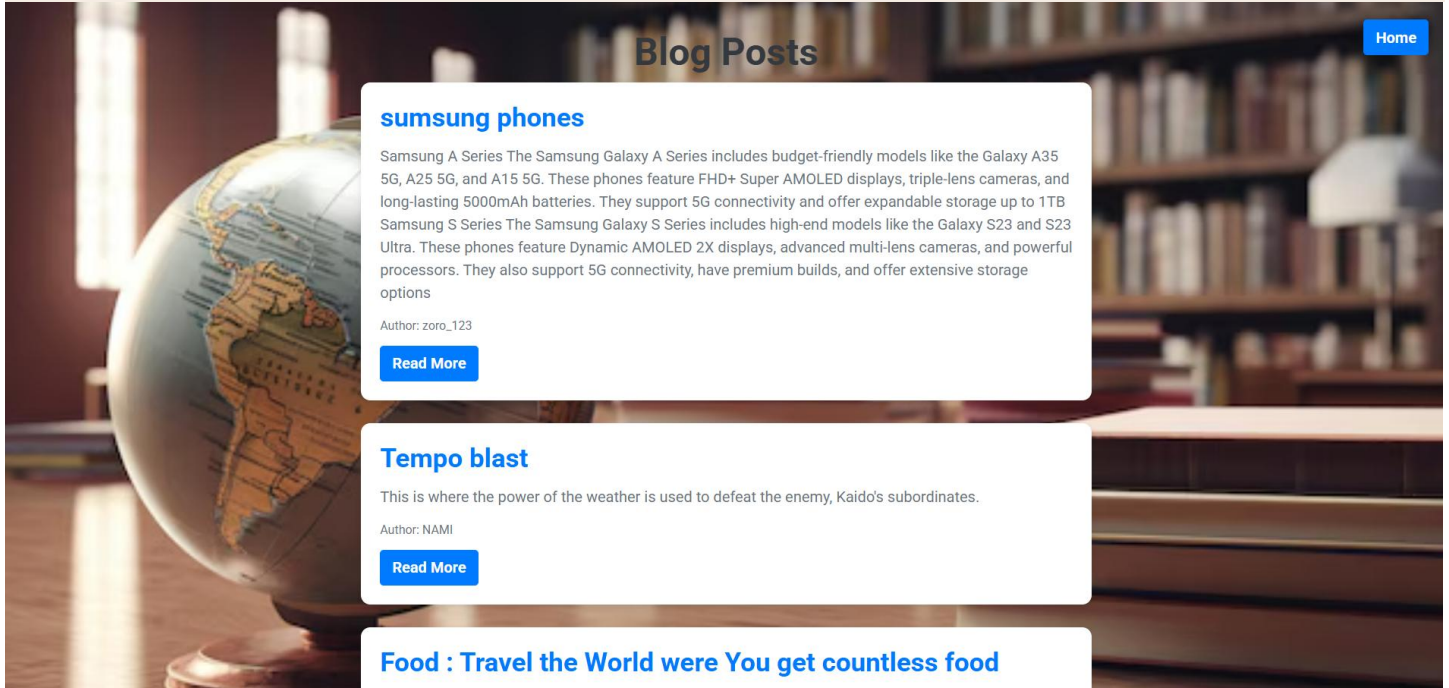| Title	Content	Actions
You have not created any posts yet.		
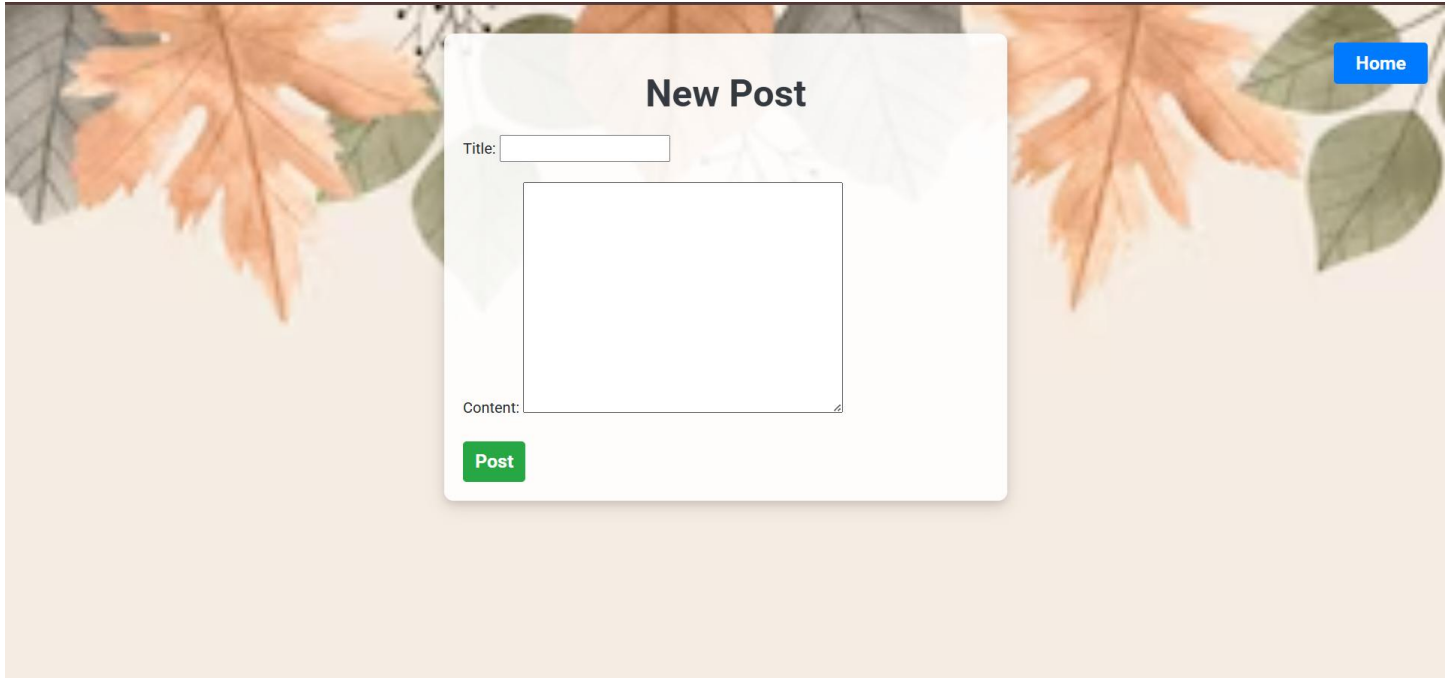
New Post

Title: [   ]

Content: [   ]

Post

Blog Posts

sumsung phones

Samsung A Series The Samsung Galaxy A Series includes budget-friendly models like the Galaxy A35 5G, A25 5G, and A15 5G. These phones feature FHD+ Super AMOLED displays, triple-lens cameras, and long-lasting 5000mAh batteries. They support 5G connectivity and offer expandable storage up to 1TB Samsung S Series The Samsung Galaxy S Series includes high-end models like the Galaxy S23 and S23 Ultra. These phones feature Dynamic AMOLED 2X displays, advanced multi-lens cameras, and powerful processors. They also support 5G connectivity, have premium builds, and offer extensive storage options

Author: zoro_123

Read More

Tempo blast

This is where the power of the weather is used to defeat the enemy, Kaido's subordinates.

Author: NAMI

Read More

Food : Travel the World were You get countless food

Food : Travel the World were You get countless food

Food is a global journey, from spicy Thai curries to Italian pasta, Japanese sushi, and Mexican tacos. Each bite offers a taste of culture, tradition, and a world of flavors.

Author: sagar

Home

Comments

thats great sagar.... happy to see your blog.....

By: luffy

July 9, 2024, 6:11 a.m.

Add a Comment

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

| Groups | + Add |
| Users | + Add |

BLOG

| Comments | + Add |
| Posts | + Add |

✅ The post "Iron Man" was changed successfully.

Select post to change

ADD POST +

Search

Action: ---------- Go 0 of 7 selected

	TITLE	AUTHOR	CREATED AT	UPDATED AT
☐	Iron Man	Tony@Stark	July 30, 2025, 1:32 p.m.	July 30, 2025, 1:32 p.m.
☐	jobs in dubai	ganesh12	July 10, 2024, 2:42 p.m.	July 10, 2024, 2:42 p.m.
☐	I am A shiprigth	Franky	July 9, 2024, 6:20 a.m.	July 9, 2024, 6:20 a.m.
☐	GUM GUM FRUIT	luffy	July 9, 2024, 6:14 a.m.	July 9, 2024, 6:14 a.m.
☐	Food : Travel the World were You get countless food	sagar	July 9, 2024, 6:08 a.m.	July 9, 2024, 6:08 a.m.
☐	Tempo blast	NAMI	July 9, 2024, 6:04 a.m.	July 9, 2024, 6:04 a.m.
☐	sumsung phones	zoro_123	July 6, 2024, 7:02 a.m.	July 6, 2024, 7:02 a.m.

7 posts