

Django Notes App – Project Documentation

1. Project Title

Smart Notes – Django Class-Based Views Project

2. Project Overview

This project is a Notes Application built with Django. It demonstrates:

- Django project structure
- Class-Based Views (CBV)
- Authentication (Login, Logout, Signup)
- CRUD operations on Notes
- User-specific data filtering
- Django templates & Bootstrap integration

The project allows each authenticated user to create, view, update, and delete their own notes.

3. Tech Stack

- **Backend:** Django
 - **Frontend:** HTML, CSS, Bootstrap
 - **Authentication:** Django Auth System
 - **Database:** SQLite
 - **Tools:** VS Code
-

4. Features

User Authentication

- Login
- Logout
- Access control using LoginRequiredMixin

Notes Management

- Create Note
- View List of Notes
- View Single Note
- Edit Note
- Delete Note
- Notes tied to logged-in user only

Class-Based Views Used

- ListView
 - DetailView
 - CreateView
 - UpdateView
 - DeleteView
-
-

5. How the App Works

Notes Model

Each note contains:

- title
- text
- created timestamp
- user (ForeignKey)

User-Specific Notes

Using:

```
self.request.user.app.all()
```

The app shows only the logged-in user's notes.

Notes CRUD Operations

Django class-based views make CRUD operations simple.

Authentication

Handled using Django's built-in LoginView and LogoutView.

6. Key Django Code Summary

Notes Model

Stores user-assigned notes.

Notes Views

- ListView → List all logged-in user notes
- DetailView → View specific note
- CreateView → Add new note
- UpdateView → Edit note
- DeleteView → Delete note with confirmation

Home Views

Handles login + logout routes.

7. URL Endpoints

Endpoint	View	Description
/	HomeView	Welcome page
/login/	LoginInterfaceView	User login
/logout/	LogoutInterfaceView	User logout
/smart/app/	NotesListView	List of user notes
/smart/app/create/	NotesCreateView	Add new note
/smart/app/<id>/	NotesDetailView	View note
/smart/app/<id>/update/	NotesUpdateView	Edit note

8. What I Learned

- Building Django apps from scratch
 - Implementing Class-Based Views
 - Working with Django Forms
 - Managing user sessions
 - Applying Bootstrap in Django templates
 - Using LoginRequiredMixin for access restriction
 - Handling URL namespaces and template inheritance
-

9. Conclusion

This Notes App project is an excellent introduction to Django. It taught me how to implement authentication, templates, class-based views, and user-specific CRUD functionality. It lays a strong foundation for building real-world Django applications.