Sticky Notes Application – Design Plan

1. Purpose of the Application

The Sticky Notes application is designed to help users create, view, edit, and delete personal notes in a simple and organized way. It functions like a digital version of physical sticky notes, allowing users to manage their ideas or reminders securely after logging in.

2. Users

There is one main type of user:

- **Authenticated User** – a registered user who can log in, create their own notes, and manage them.
  Each user can only view and modify their own notes.

3. Features

The main features of the application include:

1. **User Authentication:** Users can log in and log out of the system.
2. **Create Note:** Add a new note with a title and content.
3. **View Notes:** Display all notes belonging to the logged-in user.
4. **Edit Note:** Update the title or content of an existing note.
5. **Delete Note:** Remove a note from the system.

4. Design Pattern (MTV Structure)

The Sticky Notes application follows Django's **MTV (Model-Template-View)** pattern, which is similar to the **MVC** design pattern.

Component	Description
Model	Represents the note data stored in the database. The Note model includes fields such as title, content, created date, and user.
Template	Contains the HTML files that define what the user sees on each page, such as the list of notes, note form, and login page.
View	Contains the logic for handling requests and responses. It manages actions such as displaying notes, creating new ones, or deleting them.

5. Database Design

The application uses one main model: **Note**.
Each note belongs to one user.

Note Model Fields:

- `id`: Auto-generated primary key.
- `title`: Short text field for the note title.
- `content`: Long text field for the main body of the note.
- `created`: Date and time when the note was created.
- `user`: Foreign key linking each note to the logged-in user who created it.

Relationship:

- One **User** can have many **Notes**.
- Each **Note** belongs to one **User**.

6. User Flow

1. The user opens the application and logs in.
2. After logging in, the user is directed to the home page where all their notes are displayed.
3. The user can:
    a. Create a new note using the “New Note” button.

    b. View details of an existing note.
    c. Edit a note to update its content.
    d. Delete a note they no longer need.

4. When finished, the user can log out of the application.

7. Interface Design (Wireframe Overview)

Home Page

```
---------------------------------
| 📝 My Sticky Notes              |
| [ + New Note ]                 |
---------------------------------
| [Note 1] [Note 2] [Note 3]    |
| [Note 4] [Note 5] [Note 6]    |
---------------------------------
```

Create/Edit Note Page

```
---------------------------------
| Title: [__________]            |
| Content: [________________]    |
| [Save] [Cancel]                |
---------------------------------
(Altercations have been made for the design of the app)
```

8. Technologies Used

Component	Technology
Backend	Django Framework
Frontend	HTML, CSS (Bootstrap)
Database	SQLite or MariaDB
Authentication	Django’s built-in User model and login system

9. Summary

The Sticky Notes application is a user-friendly system built using Django. It follows the MTV design pattern, providing a clean separation between data (Model), logic (View), and presentation (Template). Users can securely manage their notes through a simple interface, making it both practical and easy to use.