Blog site outline

### **1. Introduction:**

* Content
  + Overview of Django and its benefits for web development.
  + Brief introduction to the blog site project.
  + Features to be implemented:
    - User authentication (login, logout, register).
    - CRUD (Create, Read, Update, Delete) operations for blog posts.
    - Categories and tags for posts.
    - Comments system for user interaction.
    - Pagination and search.

### **2. Setting Up the Environment**

* **Time:**
* **Content**:
  + Installing Python and pip.
  + Setting up a virtual environment.
  + Installing Django.
  + Creating a new Django project.
  + Running the development server.

### **3. Understanding Django Project Structure & Install App**

* **Time**:
* **Content**:
  + Explaining the Django project structure.
  + Overview of settings.py, urls.py and manage.py.
  + Creating a new app for the blog.
  + Registering the app in INSTALLED\_APPS.

### **4. Creating Models for the Blog**

* **Time**:
* **Content**:
  + Defining the **Post** model:
    - Fields: title, content, slug, author, created\_at, updated\_at, category, tags.
    - Using choices for categories (Option Example: Technology, Lifestyle).
    - Adding Meta options for ordering.
  + Explaining the relationship between models using ForeignKey and ManyToManyField.
  + Introduction to Django’s ORM.
  + Applying migrations.

### **5. Setting Up the Admin Panel**

* **Time**:
* **Content**:
  + Registering models in the admin site.
  + Create a Super User.
  + Adding some initial blog posts via the admin interface.

### **6.0. Creating Base Template**

### **6. Creating Views and Templates**

* **Time**:
* **Content**:
  + Setting up a basic home page to display all blog posts.
  + Writing views to handle the blog list.
  + Creating templates with Django template language.
  + Connecting views to urls.py.

### **7. Styling the Blog with CSS**

* **Time**:
* **Content**:
  + Integrating Bootstrap or a custom CSS file.
  + Styling the home page and blog list.
  + Adding a responsive navigation bar with links to key pages (home, login, register)

### **8. Single Blog Post Page**

* **Time**:
* **Content**:
  + Creating a detailed view for individual posts.
  + Linking blog titles on the home page to their respective detailed views.
  + Adding a "Back to Home" button.
  + Styling the single post page with CSS.

### **9. User Authentication**

* **Time**:
* **Content**:
  + Brief explanation of Django’s built-in User model.
  + Setting up authentication:
    - Creating login, logout, and register views.
    - Using Django’s built-in LoginView and LogoutView.
    - Customizing the registration process with a form.
  + Protecting views using the @login\_required decorator.

### **10. Adding and Managing Blog Posts**

* **Time**:
* **Content**:
  + Brief explanation of Django’s forms and ModelForm.
  + Creating forms for adding and editing blog posts.
  + Restricting post creation and editing to logged-in users.
  + Setting up update and delete views.
  + Validating data with is\_valid() and clean() methods.

### **11. Comments System**

* **Time**:
* **Content**:
  + Creating a Comment model linked to the Post model via ForeignKey.
  + Adding a comment form on the single post page.
  + Writing a view to handle comment submission.
  + Displaying comments under the respective blog posts.

### **12. Categories and Tags**

* **Time**:
* **Content**:
  + Implementing category as a CharField with choices.(Modify)
  + Implementing tags as a ManyToManyField.(Modify post models)
  + Displaying categories and tags on the post detail and list pages.
  + Creating views to filter posts by category or tag.
  + Adding links to filter posts dynamically.

### **13. Pagination and Search**

* **Time**:
* **Content**:
  + Adding pagination for the blog list view using Django’s Paginator.
  + Implementing a search bar to filter posts by title or content.
  + Adding search results to the home page.

### **14. Conclusion and Next Steps**

* **Time**:
* **Content**:
  + Recap of the project and features implemented.
  + Discussing potential enhancements:
    - User profiles.
    - Like, dislike system for posts.
    - Social sharing buttons.