

CW15-2: Django – Models – Admin – Views – Templates

Project 2: Django Blogging Platform

Build a blogging platform using Django, focusing on URL routing, views, templates, and template inheritance. The project will have the following structure and style:

1. Create the Django Project:

Use the `django-admin startproject` command to create a new Django project named "BlogPlatform".

2. Create the Data Models:

Define the following data models in the `models.py` file of the "BlogApp" directory:

- **Post Model:** Fields: title, content, author, publication date, and any other relevant information.
- **Category Model:** Fields: name, description, and any other relevant information.
- **Comment Model:** Fields: post, author, content, date, and any other relevant information.
- **Author Model:** Fields: name, bio, and any other relevant information for an author.

3. Establish the Relationships between Models:

Each post can belong to only one category.

Each category can have multiple posts.

Each post can have multiple comments.

Each comment belongs to a single post.

Each author can have multiple comments.

Each comment is written by a single author.

Each post is written by a single author.

Each author can write multiple posts.

Each comment belongs to a single post.

Each post can have multiple comments.

4. Register Models in the Django Admin Panel:

- In `admin.py` file in your app directory Register each model.
- Define the `__str__` for each model to see a meaningful representation in the admin panel.

All of your templates must be responsive to different sizes using the Bootstrap framework

5. Create Views:

Open the views.py file in the "BlogApp" directory. Import the necessary modules and models.

- **post_list:** Show all of the posts with a link to view of post details.
- **post_details:** Show a specific post and all of the information.
- **category_list:** Show all of the categories with a link to view of category details.
- **category_details:** Show a specific category and all of the information.
- **author_list:** Show all of the authors with a link to view of author details.
- **author_details:** Show a specific author and all of the information.

6. Configure URL Routing:

- Open the urls.py file in the project directory.
- Define URL patterns to map the views to specific URLs. You can use these patterns:

posts/	categories/	authors/
posts/<int:pk>/	categories/<int:pk>/	authors/<int:pk>/

7. Create Templates and Implement Template Inheritance:

Create a templates directory in the "BlogApp" directory.

- Create a base template called "base.html" to define the common html layout and structure.
- Create a template called "post_list.html" to render the list of blog posts.
- Create a template called "post_details.html" to display the details of a specific blog post.
- Create a template called "category_list.html" to render the list of categories.
- Create a template called "category_details.html" to display the details of a specific category.
- Create a template called "author_list.html" to render the list of authors.
- Create a template called "author_details.html" to display the details of a specific author.
- Ensure that the templates are responsive and can adapt to different screen sizes using a responsive framework like Bootstrap.

8. Implement Template Inheritance:

- Inherit the base template "base.html" in the child templates.
- Use template blocks to override specific sections of the base template with content specific to each model.
- Use **for**, **if** blocks if necessary. Example **if** data was not provided, show **No Data Found**

All of your templates must be responsive to different sizes using the Bootstrap framework