Django Task Four

Task 1: Set Up Your Environment

- Create a virtual environment for your Django project.
- Activate the virtual environment.
- Install Django using **pip install django**.

Task 2: Create a Django Project and App

- Use the Django CLI to generate a new project named **blog_project**.
- Create a new app named **blog_app** within the project.

Task 3: Define Models

- In the **blog_app** directory, define a **Post** model in the **models.py** file.
- Include fields such as **title**, **content**, **author**, and **published_date**.

Task 4: Set Up Admin Panel

- Register the **Post** model in the admin panel.
- Create a superuser to access the admin interface.

Task 5: Create Views and Templates

- Implement views to list all blog posts and to display individual posts.
- Design templates (e.g., **post_list.html** and **post_detail.html**) to render the blog posts.

Task 6: Define URLs and Connect Views

- Configure URL patterns in **urls.py** for listing and displaying posts.
- Link these URLs to the respective views.

Task 7: Implement Create Post Functionality

- Develop a view to add new blog posts.
- Design a form in a template (**create_post.html**) to add new posts.
- Handle form submissions to create new blog posts.

Task 8: Integrate Prefetching

- Identify related objects in your models (e.g., comments, tags).
- Implement **prefetch_related** in views where related objects are queried to minimize database hits.

Task 9: Connect Templates and Views

- Use Django's **render** function to link views with their respective templates.
- Pass necessary context data to the templates for rendering.

Task 10: Test Your Blog with Prefetching

- Run the development server (**python manage.py runserver**) and navigate to the blog's URL.
- Test functionality: Create posts via the admin panel and the 'create post' form. Ensure posts display correctly and efficiently.