**Simple Blog Project**

Create a simple blogging platform with two applications: **Blog** and **Users**.

**Requirements**

**Models**

1. **Blog Application**
   * **Post**
     + Title (CharField)
     + Content (TextField)
     + Created\_at (DateTimeField)
     + Author (ForeignKey to User)
   * **Comment**
     + Content (TextField)
     + Created\_at (DateTimeField)
     + Post (ForeignKey to Post)
     + Author (ForeignKey to User)
2. **Users Application**
   * **Profile**
     + User (OneToOneField to User)
     + Bio (TextField)
     + Profile Picture (ImageField)
3. **Follow**
   * Follower (ForeignKey to User)
   * Following (ForeignKey to User)

**Views**

1. **Blog Views**
   * List all posts (GET)
   * View a single post (GET)
   * Create a new post (GET/POST)
   * Edit a post (GET/POST)
   * Delete a post (POST)
   * List comments for a post (GET)
   * Add a comment to a post (POST)
2. **User Views**
   * User registration (GET/POST)
   * User profile view (GET)
   * Edit user profile (GET/POST)
   * Follow a user (POST)
   * Unfollow a user (POST)

**Templates**

1. **Blog Templates**
   * post\_list.html - List of all blog posts.
   * post\_detail.html - Single post view with comments.
   * post\_form.html - Form for creating/editing a post.
2. **User Templates**
   * registration.html - User registration form.
   * profile.html - User profile page.

**Steps to Implement**

1. **Set Up Django Project**
   * Create a new Django project.
   * Create the two applications: blog and users.
2. **Define Models**
   * Implement the models as outlined above in models.py for each application.
3. **Migrate Database**
   * Run migrations to create database tables.
4. **Create Views**
   * Implement the views in views.py for both applications.
5. **Set Up URLs**
   * Create URL routing for each view in urls.py for both applications.
6. **Create Templates**
   * Design and create the templates for each view in a templates folder for each app.
7. **Static Files and Media**
   * Configure static files and media files for user-uploaded profile pictures.
8. **Test the Application**
   * Ensure all views are functional and templates render correctly.

**Additional Features (Optional)**

* Implement user authentication (login/logout).
* Add pagination to the post list.
* Implement search functionality for posts.