

Q.1 Django Views and URLs Implementation (Recipe Sharing Platform)

Your task is to implement views and URLs for a recipe-sharing platform using Django. You will create views to display recipe listings, recipe details, and a user profile. Additionally, you will define URL patterns to map these views to appropriate URLs.

Requirements:

1. Create Django view functions to handle the following tasks:

- Display a list of recipe.
- Display details of a recipe .
- User Profile.

2. Define URL patterns to map the views to appropriate URLs.

- The list of recipes should be accessible at the URL `"/recipe/"`.
- Individual recipe details should be accessible at the URL `"/recipe/<recipe_id>/"`.
- Adding recipe by user should be handled by a URL such as `"/user/<recipe_id>/"`.

3. Implement template files to render the views appropriately.

4. Ensure that the views and URLs are correctly configured and tested to work as expected.

The following tasks are to be fulfilled:

1. Implement view functions to handle the tasks mentioned above.
2. Define URL patterns in the project's `urls.py` file to map the views to appropriate URLs.
3. Create template files to render the views' content.
4. Ensure that the views retrieve data from sample data (define as per your own) as necessary and pass it to templates for rendering. And Write the settings also that have to done for the project

Q2: Blog Management System - URLs and Views Implementation

◆ **Scenario:** Create a blogging platform where users can read blogs, view details, and create posts.

- Implement views for **listing blogs**, **viewing blog details**, and **creating a new blog post**.
- Define URL patterns like:
 - `/blogs/` → List of blogs.
 - `/blog/<blog_id>/` → Blog details.
 - `/blog/new/` → Create a new blog.
- Implement **template inheritance** to share a common base template.

Q3: E-Commerce Product Management

◆ **Scenario:** Develop an e-commerce system to list products, show product details, and allow users to add products.

- Implement views for **listing products**, **viewing product details**, and **adding new products**.
- Define URLs:

- /products/ → List all products.
 - /product/<product_id>/ → Show product details.
 - /product/add/ → Add new products.
- Use Django **forms** to handle product addition.

Q4: Online Course Management System

◆ **Scenario:** Create a system where users can browse courses, view course details, and register.

- Implement views for **listing courses**, **showing course details**, and **user registration**.
- Define URLs:
 - /courses/ → List all courses.
 - /course/<course_id>/ → Course details.
 - /register/ → User registration.
- Use Django's **authentication system** for login/logout.

Q5: Library Management System

◆ **Scenario:** Implement a library system where users can search books, view book details, and borrow books.

- Implement views for **listing books**, **showing book details**, and **borrowing books**.
- Define URLs:
 - /books/ → List all books.
 - /book/<book_id>/ → Book details.
 - /borrow/<book_id>/ → Borrow a book.
- Use **Django models** to represent books.

Q6: Event Booking System

◆ **Scenario:** Users should be able to browse events, view event details, and book tickets.

- Implement views for **listing events**, **event details**, and **booking events**.
- Define URLs:
 - /events/ → List all events.
 - /event/<event_id>/ → Event details.
 - /event/book/<event_id>/ → Book an event.
- Display **available seats** dynamically in the template.

Q7: Social Media Platform - User Profile and Posts

◆ **Scenario:** Create a social media app where users can post updates, view profiles, and see posts.

- Implement views for **displaying posts**, **user profiles**, and **adding new posts**.
- Define URLs:
 - /posts/ → Show all posts.
 - /post/<post_id>/ → Post details.
 - /user/<user_id>/ → User profile.
- Implement **template inheritance** for consistent UI.

Q8: Movie Review System

◆ **Scenario:** Develop a system where users can browse movies, read reviews, and add their own reviews.

- Implement views for **listing movies**, **viewing movie details**, and **adding reviews**.
- Define URLs:
 - `/movies/` → Show all movies.
 - `/movie/<movie_id>/` → Movie details.
 - `/review/add/<movie_id>/` → Add a review.
- Use **Django forms** to handle reviews submission.
-

Q9: Job Portal - Job Listings and Applications

◆ **Scenario:** Build a job portal where users can browse jobs, view details, and apply.

- Implement views for **listing jobs**, **viewing job details**, and **applying for a job**.
- Define URLs:
 - `/jobs/` → List all jobs.
 - `/job/<job_id>/` → Job details.
 - `/apply/<job_id>/` → Apply for a job.
- Store **job applications** in the database.

Q10: Online Examination System

◆ **Scenario:** Develop a platform where students can take exams and see results.

- Implement views for **listing exams**, **taking an exam**, and **showing results**.
- Define URLs:
 - `/exams/` → List all exams.
 - `/exam/<exam_id>/` → Take an exam.
 - `/results/` → View results.
- Use **Django forms** for quiz submissions.
-