`Assignment (Introduction to Django)

1. **What is Django, and why is it popular? Explain with some real time examples.**

Django is a python web framework which uses MVT (Model - View - Template), unlike other web frameworks like spring or express, this web framework gives a lot of flexibility and is easy to assess by pre-configuring the difficult part. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

* Provides built-in features like authentication, ORM, and admin panel for Fast Development:
* Protects against SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
* Used in scalable applications handling millions of users exampl instagram

#### Real-time examples

1. Instagram – Uses Django to manage millions of users, media files, and real-time notifications.
2. Mozilla – Uses Django for scalability in its websites and documentation platforms.
3. Spotify – Uses Django for recommendation engines and data analytics.
4. **List out Django's key features and explain every feature with some real time examples.**

* Django offers numerous features that make it stand out. The MVT architecture ensures a structured approach to web development by separating data handling (Model), business logic (View), and user interface (Template).
* The Object-Relational Mapper (ORM) simplifies database interactions by allowing developers to work with Python objects instead of raw SQL queries. The built-in admin interface enables easy management of database records without additional development.
* Django’s security mechanisms protect applications from common threats such as SQL injection and cross-site scripting (XSS). The framework is also highly scalable, making it ideal for applications requiring high performance, such as Instagram, which manages millions of user comments.
* Django’s Django REST Framework (DRF) allows developers to build APIs seamlessly, making it a preferred choice for backend development in mobile and web applications.

1. **What is the MVT architecture?**

The MVT (Model-View-Template) architecture is Django’s core design pattern.

* The Model represents the database structure and defines the data fields.
* The View handles business logic and processes user requests
* The Template is responsible for presenting data in an HTML format.

This structure ensures better organization and maintainability of the code. For instance, in a blogging application, the Model stores blog posts, the View retrieves posts from the database, and the Template displays them on a webpage. This separation allows developers to modify one part without affecting the others, improving efficiency.

1. **How does Django handle scalability?**

Django is designed for scalability, making it suitable for applications with high traffic and large datasets. It achieves scalability through database optimization, supporting various relational databases like PostgreSQL and MySQL. Django also integrates caching mechanisms like Redis and Memcached to store frequently accessed data and reduce database load. Additionally, Django supports horizontal scaling, enabling applications to run across multiple servers to handle increased user traffic. Companies like Instagram rely on Django’s scalability features to manage their vast user base, large media files, and real-time notifications.

1. **What is the role of manage.py?**

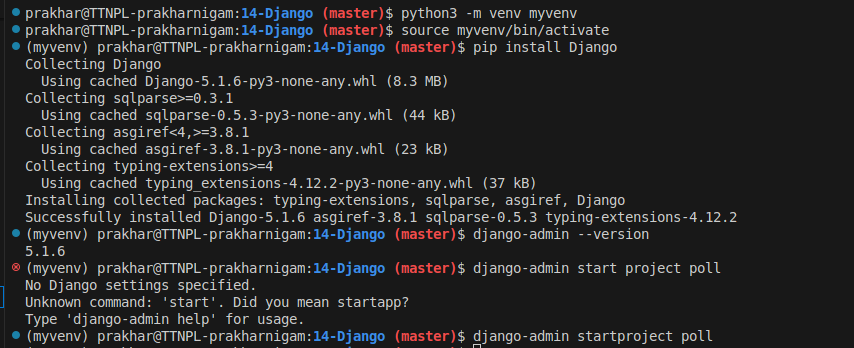
The manage.py file plays a crucial role in Django projects by offering a command-line interface for various administrative tasks. It is used to start the Django development server (runserver), create and apply database migrations (makemigrations) and generate a superuser account (createsuperuser). Additionally, it provides access to the Django shell for executing database queries and testing code snippets.

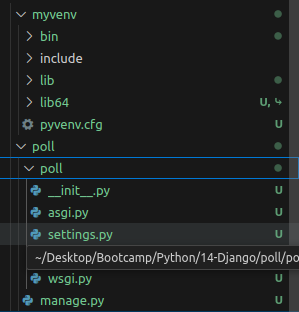
By using manage.py, developers can efficiently manage their projects without manually writing repetitive commands, streamlining the development process.

1. **Create a virtual env in your local system and install Django (5.x.x) to create a project. List out all the steps and explain them.**

I followed following steps:

* First, i used venv to create a virtual environment
* Then i activated the virtual environment by running /bin/activate
* Once i was in the virtual environment, i installed Django using pip
* By default it downloaded the latest version, however just to be sure, I checked the version as well
* Then using ‘django-admin’, I started a project called poll

****

****

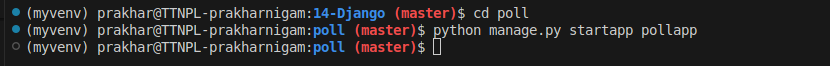
Project created

1. **Create a new Django project in the recently created env.**

Since this question has been answered in the last question, i moved forward to demonstrate here how i created an app in my project.

Following the last question, I changed my directory to 14-Django/Poll to make use of manage.py to create an app

I used the command startapp <appnamee> to create a new app



In my poll project an app called pollapp was created

