• Overview of Django: Web development framework.

**What is Django?**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Created by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

**Key Features of Django**

1. **MVC Architecture**: Django follows the Model-View-Controller (MVC) architectural pattern, though it refers to it as Model-View-Template (MVT). This pattern promotes separation of concerns, making it easier to manage and maintain code.
2. **Built-in Admin Interface**: One of Django's standout features is its automatically generated admin interface, which allows for quick and easy management of application data.
3. **ORM (Object-Relational Mapping)**: Django includes a powerful ORM that lets you interact with your database using Python code instead of raw SQL. This makes database operations more intuitive and less error-prone.
4. **Template System**: Django's template system allows you to define your HTML layout separately from your business logic, promoting a clear separation of concerns and easier management of your codebase.
5. **URL Routing**: Django provides a clean and easy way to map URL patterns to views using a routing system. This helps in managing URLs of your application in a concise and understandable manner.
6. **Security Features**: Django includes several built-in security features to help you protect your application from common vulnerabilities like SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and clickjacking.
7. **Scalability**: Django's design promotes scalability, making it suitable for both small and large applications. It can handle high traffic loads and large amounts of data efficiently.
8. **Community and Documentation**: Django has a large and active community, offering extensive documentation and numerous third-party packages that extend its functionality.

**Basic Workflow in Django**

1. **Setting Up the Project**:

pip install django

django-admin startproject myproject

cd myproject

1. **Creating an App**:

python manage.py startapp myapp

1. **Defining Models**:
   * Define models in myapp/models.py to represent the database schema.

* python

from django.db import models

class Member(models.Model):

name = models.CharField(max\_length=100)

email = models.EmailField()

mobile = models.CharField(max\_length=15)

def \_\_str\_\_(self):

return self.name

1. **Applying Migrations**:
   * Create and apply database migrations to reflect the model changes.

python manage.py makemigrations

python manage.py migrate

1. **Creating Views**:
   * Define views in myapp/views.py to handle requests and return responses.

* python

from django.shortcuts import render

def index(request):

return render(request, 'myapp/index.html')

1. **URL Routing**:
   * Map URLs to views in myapp/urls.py and include them in the project's urls.py.

* python

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index'),

]

* + Include the app's URLs in the project's urls.py:
* python

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('myapp.urls')),

]

1. **Templates**:
   * Create HTML templates in the templates directory and render them in views.

html

<!-- myapp/templates/myapp/index.html -->

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

</head>

<body>

<h1>Welcome to Digital Society</h1>

</body>

</html>

1. **Running the Development Server**:
   * Start the Django development server to view your application.

bash

python manage.py runserver

**Conclusion**

Django is a versatile and powerful framework that simplifies the process of building web applications. Its built-in features and adherence to best practices make it a great choice for both beginners and experienced developers.

• Advantages of Django (e.g.,scalability,security).

• Django vs. Flask comparison: Which to choose and why.