INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY

APPLICATION DEVELOPMENT

FRAMEWORK

PRACTICAL - 1

Het Mamtora

20BCE513

B.Tech Computer Science

Batch – E4

PRACTICAL - 1

AIM : : Introduction to MVC and MVT architecture. Comparative study of MVC and MVT. Explore the Django application Structure.

Introduction to MVC and MVT architecture

1. **MVC Architecture**

   Model View Controller, known as MVC separates the code as three components. MVC separates the business logic and presentation layer from each other. It was traditionally used for desktop graphical user interfaces (GUIs). Nowadays, MVC architecture has become popular for designing web applications as well as mobile apps.

   - **MODEL -** This layer deals with data-related logic. For example, it can retrieve, change and save data to the database.
   - **VIEW -** We can call it the presentation layer. It is responsible for collecting data from the model or user and presenting it. In a web application, everything that is displayed in the browser falls under View.
   - **CONTROLLER -** It controls the data flow and interaction between view and model. For example, a controller, based on a request or action, will collect data from a database with the help of Model and send it to the user through Views.
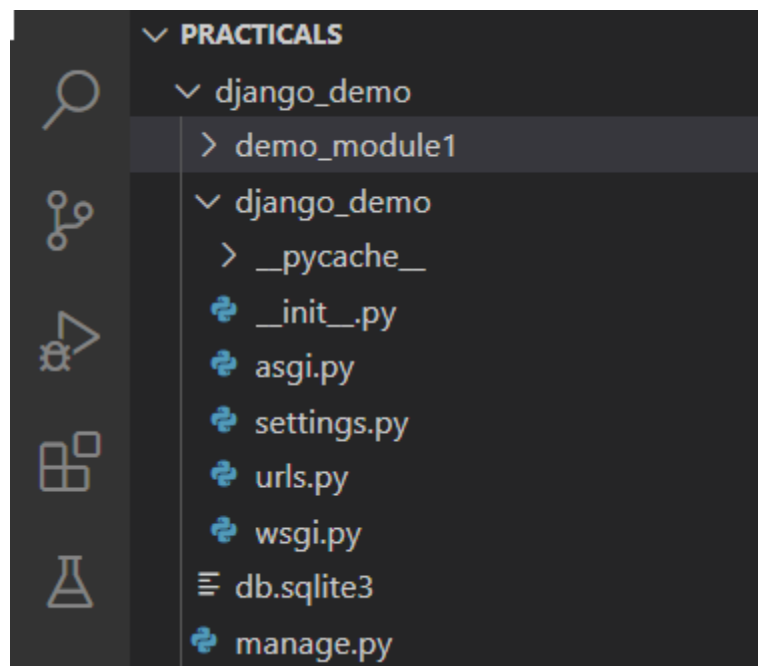
2. **MVT Architecture**

   Model View Template, widely known as MVT is another design pattern similar to MVC. Like MVC, the MVT design pattern also separates the code into three parts.

   - **MODEL -** Same as the model in MVC. It contains the code responsible for dealing with data and databases.
   - **VIEW -** In MVT design pattern, the View is decides what data should be displayed.
   - **TEMPLATE -** Templates are used to specify a structure for an output. Data can be populated in a template using placeholders. It defines how the data is presented. An example is a Generic list view that we can use to display a set of records from the database.
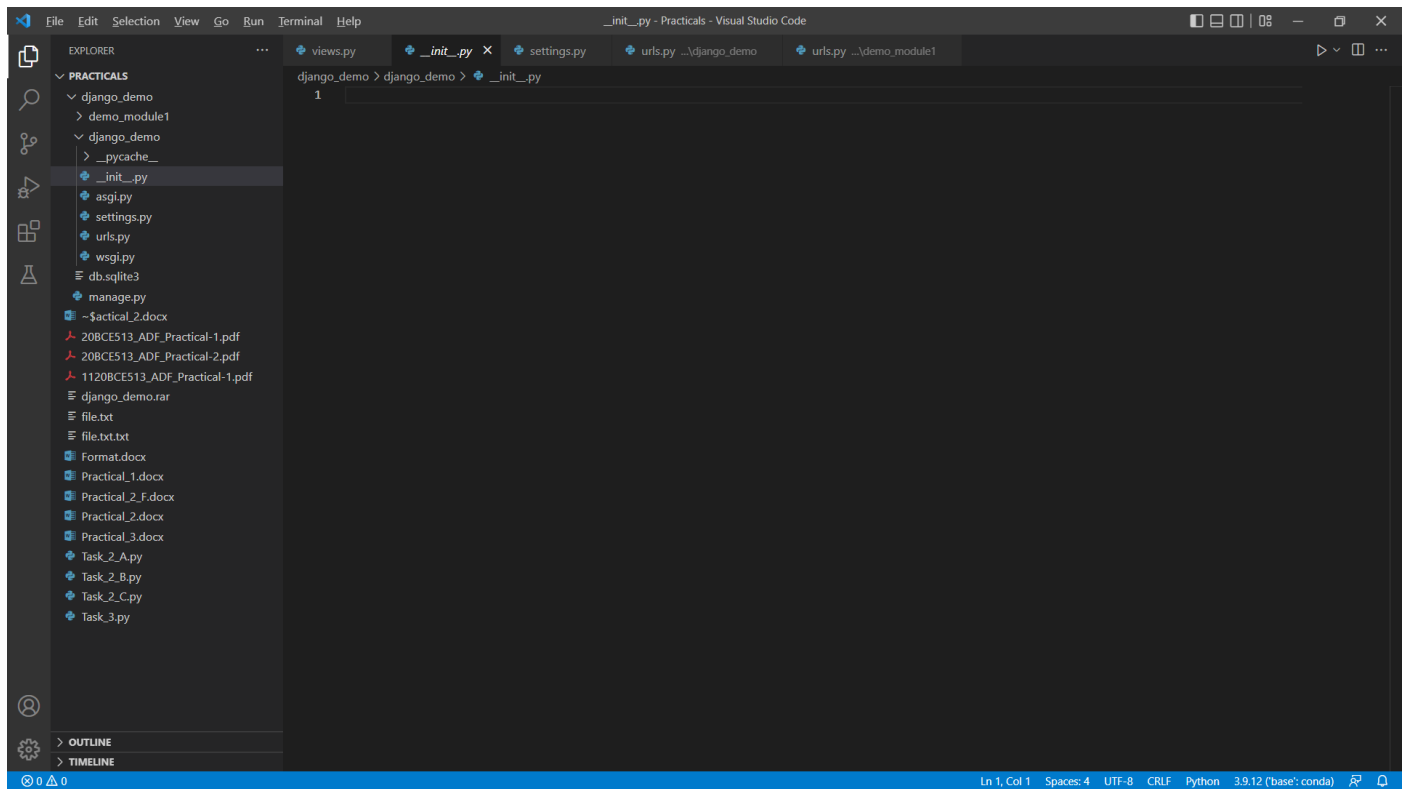
Comparative study of MVC and MVT architecture.

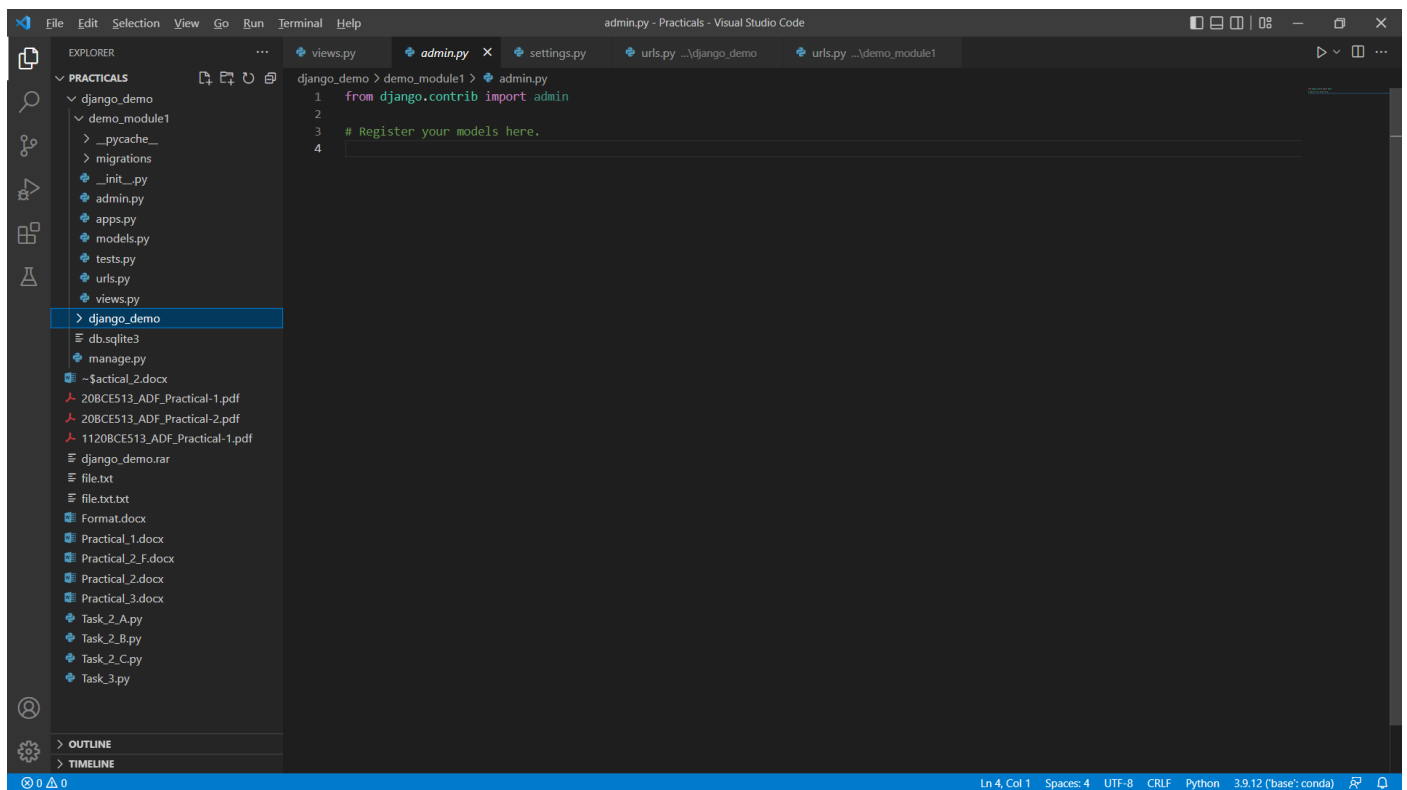| MVC	MVT
View tells how the user data will be presented.	Templates are used in MVT for that purpose.
Suitable for development of large applications but not for slower applications.	Suitable for both small and large applications.
It doesn't involves mapping of URL's	URL pattern mapping takes place.
Control specific code is to be written.	Controller part is manages by framework itself.
It is tightly coupled.	It is loosely coupled.
Flow is defined hence is east to understand. Eg. Spring MVC, ASP.net MVC	Flow is sometimes harder to understand. Eg. Django.

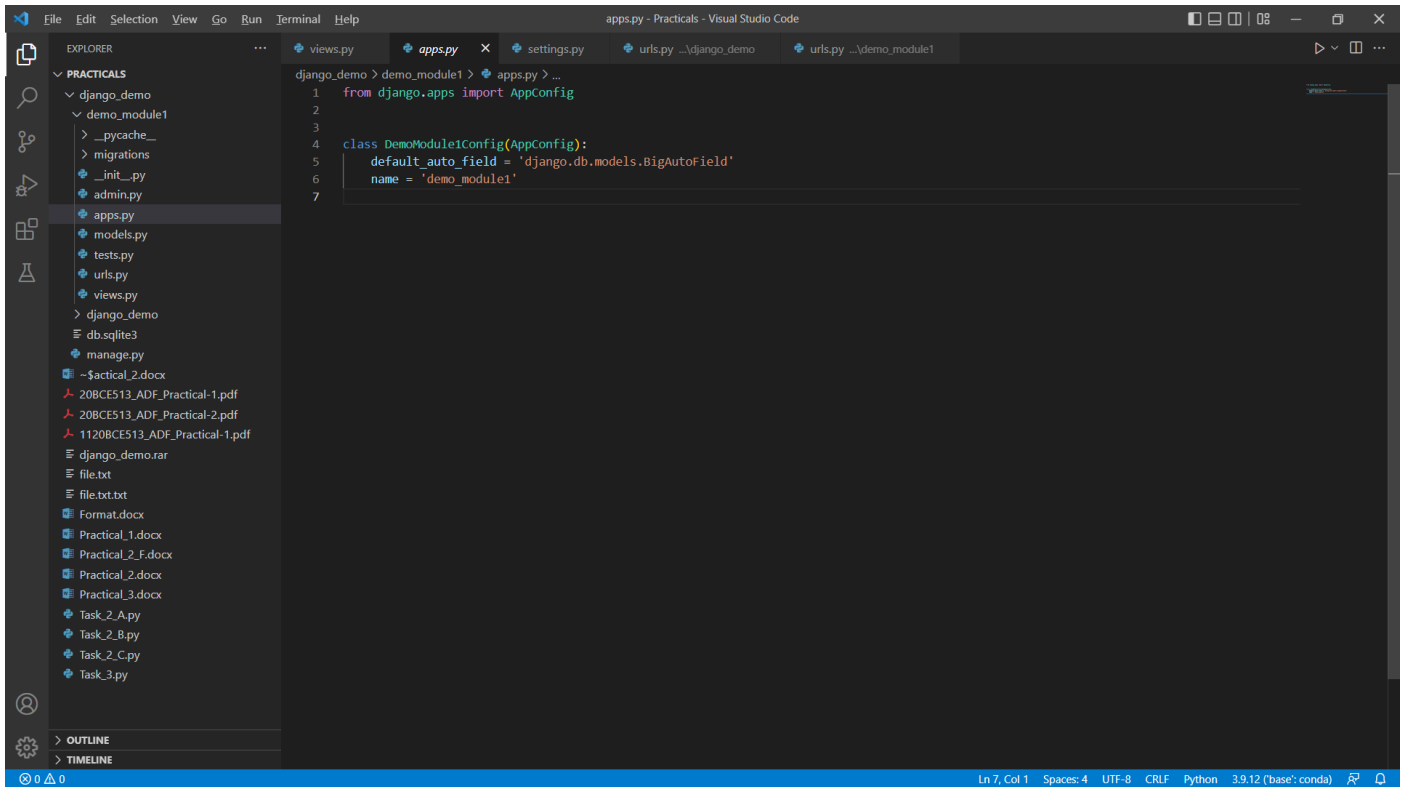**Explore the Django application structure**
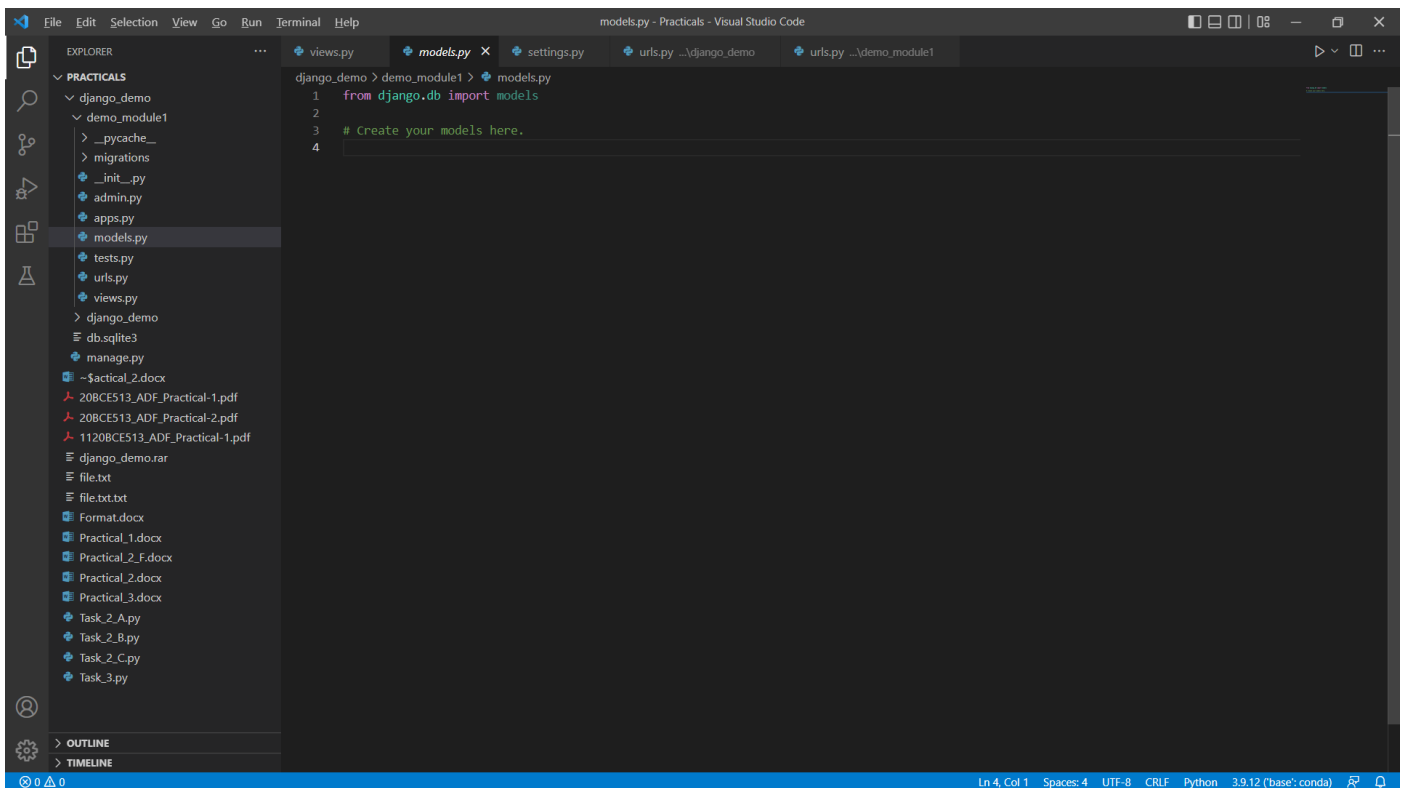
__init__.py

Admin.py


```python
from django.contrib import admin

# Register your models here.
```

Apps.py

```python
from django.apps import AppConfig


class DemoModule1Config(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'demo_module1'
```

Models.py

```python
from django.db import models

# Create your models here.
```

Tests.py

```python
from django.test import TestCase

# Create your tests here.
```

Views.py

```python
from django.http import HttpResponse
from django.shortcuts import render

# Create your views here.
def index(request):
    return HttpResponse('Hello World')
```