# Module – 5 (DB and Python Framework)

## 1. Why Django should be used for web-development? Explain how you can create a project in Django?

### Ans :-

-> Django is an open-source framework for backend web applications based on Python one of the top web development languages.

-> Its main goals are simplicity, flexibility, reliability, and scalability. And when introduced to what is Django, Python and its features open up in a new way.

-> High scalability:- Django is highly scalable and can handle large amounts of traffic and data.

-> Security:- Django provides a lot of built-in security features, such as protection against SQL injection, cross-site scripting, and clickjacking.

-> Versatility:- Django can be used to build a variety of web applications, including content management systems, social networking sites, e-commerce platforms, and more.

-> Community:- Django has a large and active community of developers who contribute to its development and provide support to other developers.

**~~~ To create a project in Django, you can follow these steps:-**

-> First, let's create a new virtual environment for this project.

-> We can now install Django:- pip install django

->Create a project:- To create a new Django project, run the following command:- django-admin startproject projectname.

-> Create an app:- Once you have created your project, you can create an app within the project by running the following command: python manage.py startapp appname. Replace "appname" with the name of your app.

-> Define your models:- Models are the way that Django represents data. You can define your models in the models.py file within your app.

-> Create database tables:- Once you have defined your models, you can create the database tables by running the following command: python manage.py makemigrations followed by python manage.py migrate.

-> Create views:- Views are the way that Django handles HTTP requests. You can define your views in the views.py file within your app.

-> Create templates:- Templates are the way that Django renders HTML. You can create your templates in the templates directory within your app.

-> Define URLs:- URLs are the way that Django maps URLs to views. You can define your URLs in the urls.py file within your app.

-> Run the development server:- Finally, you can run the development server by running the following command: python manage.py runserver.
This will start the development server at http://127.0.0.1:8000/.

## 2. How to check installed version of django?

### Ans :-

~~> Open a terminal or command prompt.

-> Type the following command and press Enter:-

python -m django --version

~~> Another way to check Django version for python3 is by using the below command :-

django-admin --version

## 3. Explain what does django-admin.py make messages command is used for?

### Ans :-

Manage.py in Django is a command-line utility that works similar to the djangoadmin command. The difference is that it points towards the project's settings.py file.

This manage.py utility provides various commands that you must have while working with Django.

Some of the most commonly used commands are :-

- python manage.py startapp
- python manage.py makemigrations
- python manage.py migrate
- python manage.py runserver

## 4. What is Django URLs?make program to create django urls

## Ans :-

~~> URL is a path through which a specific web-based application and one particular page in that web application can be reached. So for any web-oriented application setting these url paths is a very key necessity. Django manages the necessary URLs in the urls.py section of the framework, following various techniques to maintain them throughout the application. The methods used to keep the URLs organized in Django are discussed below.

-> Example:-

```
# myapp/urls.py

from django.urls import path

from . import views

urlpatterns = [

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

   path('about/', views.about, name='about'),

   path('contact/', views.contact, name='contact'),

]
```

**->** In this example:

-> We import the path function from django.urls and the views module from the current application.

-> We define a list called urlpatterns, which contains instances of the path function.

-> Each path function takes three arguments:

-> The first argument is the URL pattern as a string.

-> The second argument is the view function that should handle the request.

**5. What is a QuerySet? Write program to create a new Post object in database:**

## Ans :-

-> A QuerySet is a collection of data from a database.

-> A QuerySet is built up as a list of objects.

-> QuerySets makes it easier to get the data you actually need, by allowing you to filter and order the data at an early stage.

-> In this tutorial we will be querying data from the Member table.

-> Three types of Queryset :-

**1.** Django QuerySet - Get Data

**2.** Django QuerySet - Filter

**3.** Django QuerySet - Order By

### → models.py :-

```
From django.db import models

Class userinfo(models.model):

Fristname=models.charfield(max_length=20)

Lastname=models.charfield(max_length=20)

Mobile=models.bigintegerfield()

dob=models.DateField()

email=models.EmailField()
```

→ **views.py :-**

```
From django.shortcuts import render
From .models import userinfo
Def index(request):
If request.method==’POST’:
        Newuser=userdataform(request.POST)
        If newuse.is_valid():
            Newuser.save()
           Print(“data has been saved”)
 Else:
        Print(newuser.errors)
 Return render (request, ‘index.html’)
```

## → templates/ index.html :-

```
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Welcome</title>
 <link rel="icon" type="image/x-icon" href="/static/imgs/favicon_app.png">
</head>
<body style="background-color: rgb(179, 179, 23);">
 <center><h1>Welcome to Database Project</h1></center>
 <hr>
```

```
<form method="post">
{% csrf_token %}
<input type="text" name="firstname" placeholder="Firstname"><br><br>
<input type="text" name="lastname" placeholder="Lastname"><br><br>
<input type="number" name="mobile" placeholder="Mobile"><br><br>
<input type="text" name="dob" placeholder="DOB"><br><br>
<input type="text" name="email" placeholder="Email"><br><br>
<input type="submit" value="Save">
<input type="reset">
</form>
<hr>
</body>
</html>
```

Note: In the example we use the .all() method to get all the records and fields of the table.

**6. Mention what command line can be used to load data into django ?**

To load data into Django you have to use the command line Django-admin.py loaddata.

The command line will searches the data and loads the contents of the named fixtures into the database.