WEEK-8

AIM:

List the software requirements for the installation of Django.

SOFTWARE REQUIRNMENTS:

- **Install Python:**

1. To check if python installed on your machine, open command prompt and run following command
2. python --version
3. **OUTPUT**
    - If python is installed, Python 3.10.0 (version number may change)
    - If python is NOT installed, 'python' is not recognized as an internal or external command, operable program or batch file.

- **Install Django package:**

1. Open command prompt Run any one of the following commands
    - pip install Django OR
    - py -m pip install Django
2. You can tell Django is installed and which version by running the following command
    - python -m django –version

- **Install Visual Studio Code (VS Code)**

1. To check if VS Code installed on your machine, open command prompt and run following command
2. code --version
3. **OUTPUT**
    - If vscode is installed, 1.83.1 (version number may change)
    - If vscode is NOT installed, 'code' is not recognized as an internal or external command, operable program or batch file.

WEEK-9

AIM:

Creating a Django project on windows

PROCEDURE:

1. **Create workshop folder**:

   Create a folder as C:/workshops/django/

2. **Open Command Prompt from the File Explorer :**
   Type "cmd" into the address bar and press Enter to open the Command Prompt with the path of the current folder already set.

3. **Create a Django Project**:

   Navigate to the directory where you want to create your Django project and run the following command to create a new project with the name "demo_project"

```
django-admin startproject demo_project

OR

python -m django startproject demo_project
```

4. **Navigate to the Project Directory**:

   Change your working directory to the newly created project folder

```
C:\workshops\django> cd demo_project
```

5.Initialize the Database:

Django uses a database to store information about your project. You'll need to create the database schema and tables by running the following commands:

```
python manage.py migrate
python manage.py makemigrations
```

5.Create a Superuser:

You can create a superuser to access the Django admin interface for managing your application. Run the following command and follow the prompts to set user credentials (you can set user = admin and password = admin):

```
python manage.py createsuperuser
```

6.Start the Development Server:

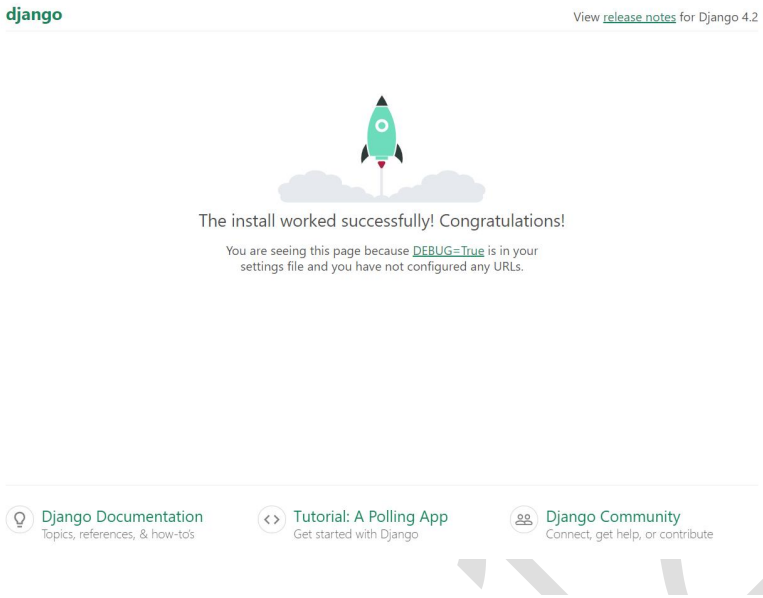
To run your Django project locally, use the development server. Run the following command

```
python manage.py runserver
```

7.Access the application in browser:

- http://127.0.0.1:8000/ (change the port number if app is running on a different port)
- http://127.0.0.1:8000/admin/ (change the port number if app is running on a different port). Log in with the superuser credentials to manage your application.

OUTPUT:


django
View release notes for Django 4.2
The install worked successfully! Congratulations!
You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.
Django Documentation
Topics, references, & how-to's
Tutorial: A Polling App
Get started with Django
Django Community
Connect, get help, or contribute

Week-10

AIM:

Create a Django app

PROCEDURE:

1. **Create app:**

```
python manage.py startapp myapp1
```

Django creates a folder named members in my project, with this content

```
demo_project
    manage.py
    myapp1
    members/
        migrations/
            __init__.py
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        views.py
```

2.Add/Update Views:

- Go to myapp1/views.py

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

def hello(request):
    return HttpResponse("Hello world!")
```

3.Define URLs

- Create a file (if not there already) named urls.py in the same folder as the views.py file in the app

```
from django.urls import path
from . import views

urlpatterns = [
  path('data/', views.hello, name='hello'),
]
```

- The urls.py file you just created is specific for the app. We have to do some routing in the root project level as well.

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('', include('myapp1.urls')),
    path('admin/', admin.site.urls),
]
```

4.Run Server:

```
py manage.py runserver
```

In the browser window, type **127.0.0.1:8000/data/** in the address bar.

OUTPUT: