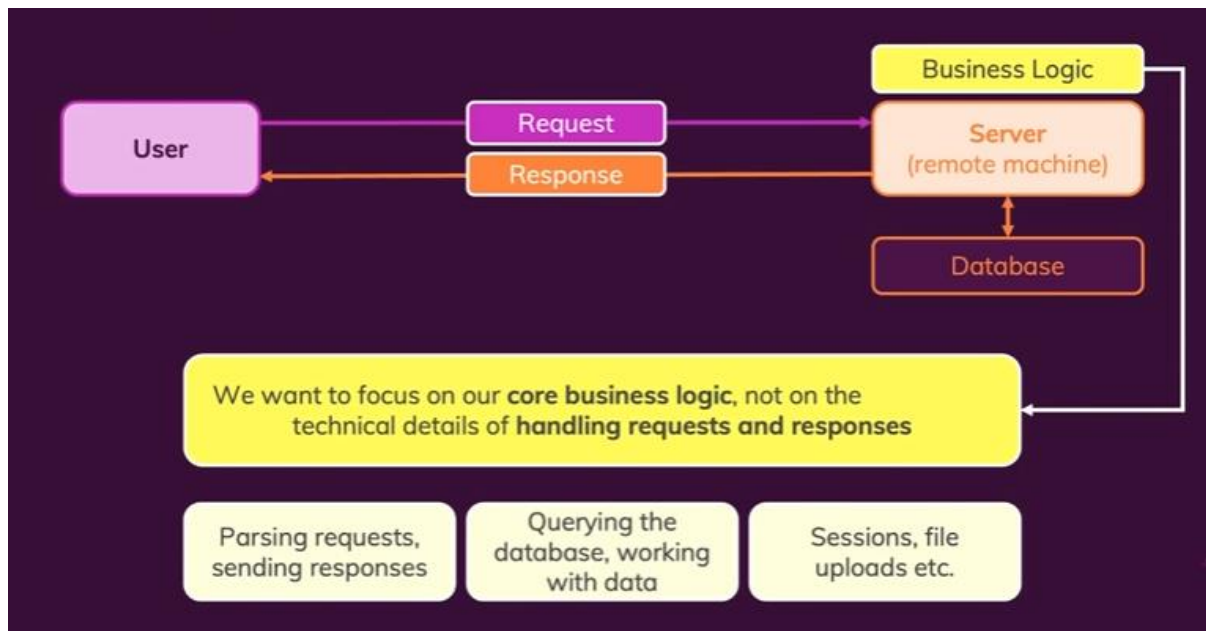


## Lab-2 Task

### 1. Comparative Study on MVC and MVT architectures.

---

#### Why Django?



#### Configuring and Installation of Django

##### 1. Install Python

###### Django version

###### Python versions

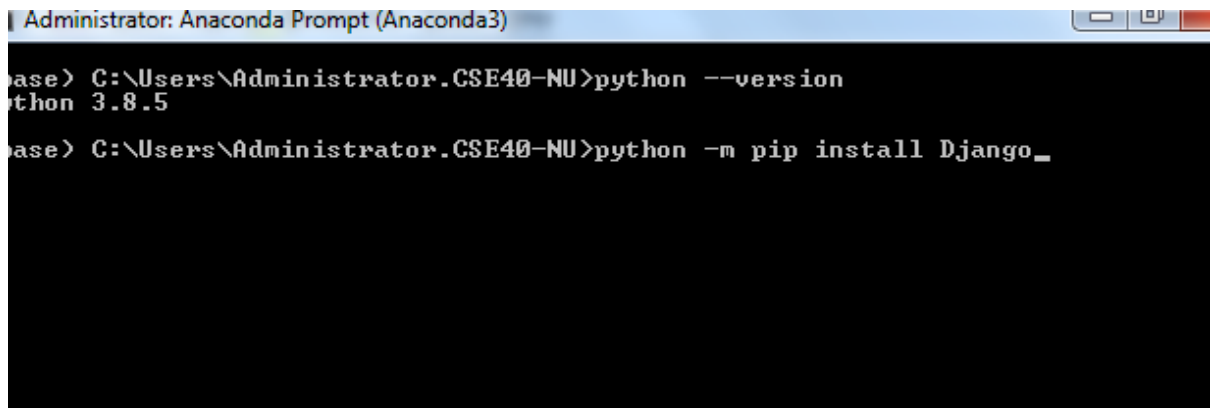
2.2	3.5, 3.6, 3.7, 3.8 (added in 2.2.8), 3.9 (added in 2.2.17)
3.0	3.6, 3.7, 3.8, 3.9 (added in 3.0.11)
3.1	3.6, 3.7, 3.8, 3.9 (added in 3.1.3)
3.2	3.6, 3.7, 3.8, 3.9

For each version of Python, only the latest micro release (A.B.C) is officially supported. You can find the latest micro version for each series on the [Python download page](#).

The latest version of Python 3 is recommended.

## 2. Open prompt (Ensure you have internet connection working)

Check version of Python, with the command *python --version*



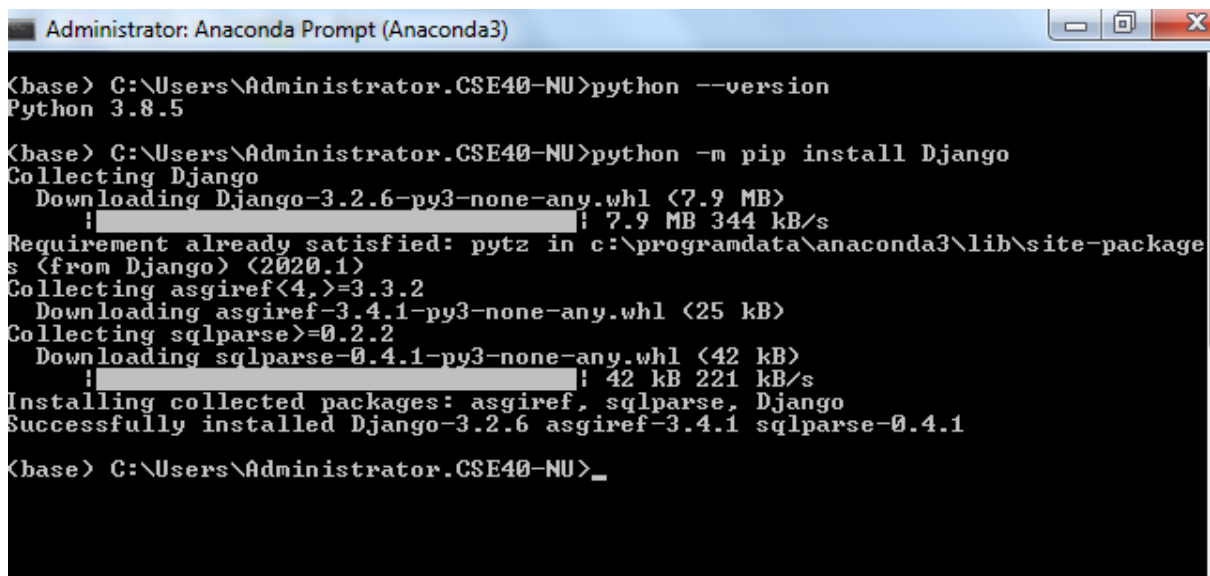
```
Administrator: Anaconda Prompt (Anaconda3)

(base) C:\Users\Administrator.CSE40-NU>python --version
Python 3.8.5

(base) C:\Users\Administrator.CSE40-NU>python -m pip install Django_
```

For installing Django use command, *python -m pip install Django*

Once Django is installed, following things will appear in command prompt:



```
Administrator: Anaconda Prompt (Anaconda3)

(base) C:\Users\Administrator.CSE40-NU>python --version
Python 3.8.5

(base) C:\Users\Administrator.CSE40-NU>python -m pip install Django
Collecting Django
  Downloading Django-3.2.6-py3-none-any.whl (7.9 MB)
    |#####| 7.9 MB 344 kB/s
Requirement already satisfied: pytz in c:\programdata\anaconda3\lib\site-packages (from Django) (2020.1)
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.1-py3-none-any.whl (42 kB)
    |#####| 42 kB 221 kB/s
Installing collected packages: asgiref, sqlparse, Django
Successfully installed Django-3.2.6 asgiref-3.4.1 sqlparse-0.4.1

(base) C:\Users\Administrator.CSE40-NU>_
```

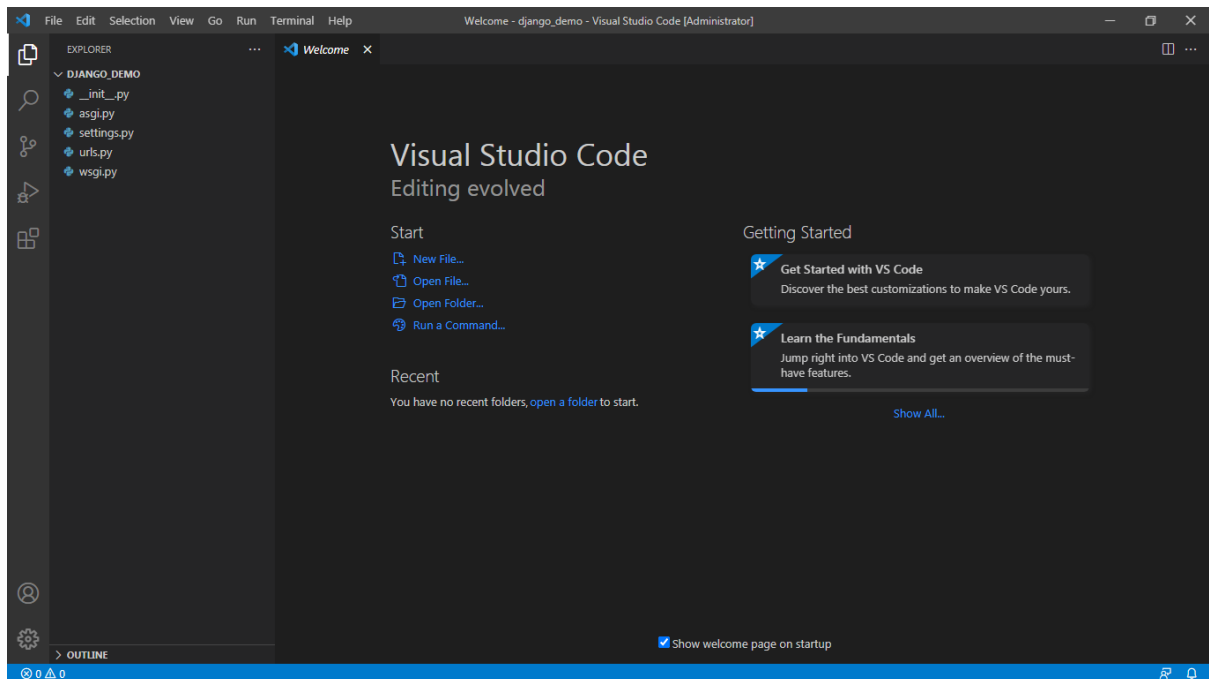
After this, use following command (This will create a new project):

*django-admin startproject django\_demo*

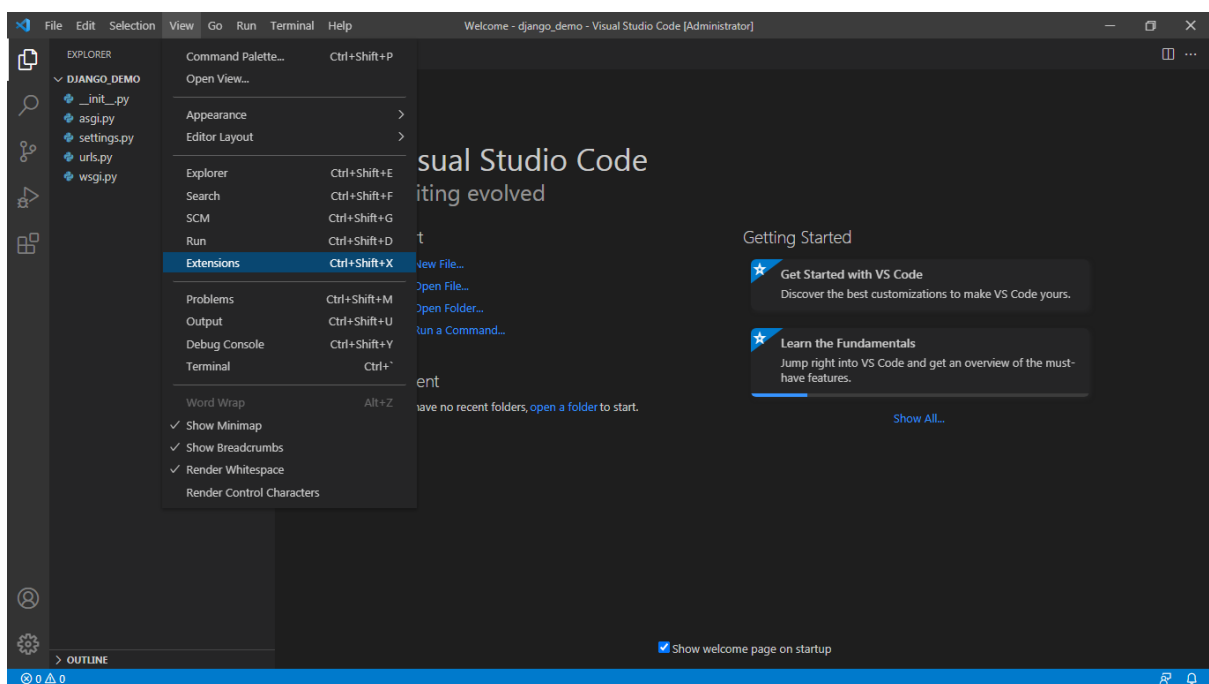
### 3. Installing Editor (pycharm or visual studio code can be used. It is recommended to use visual studio code i.e vscode)

Go to the link: <https://code.visualstudio.com/> and install visual studio code in your system.

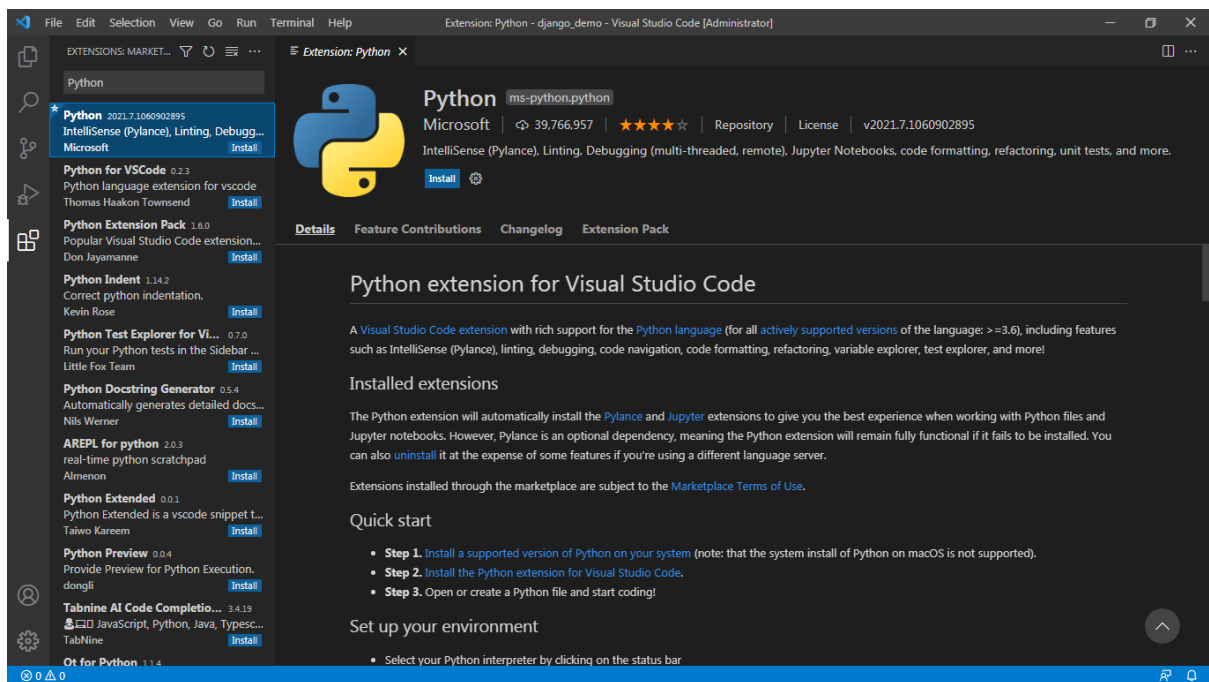
After successful installation of vscode, open it and go to the project folder created: i.e django\_demo



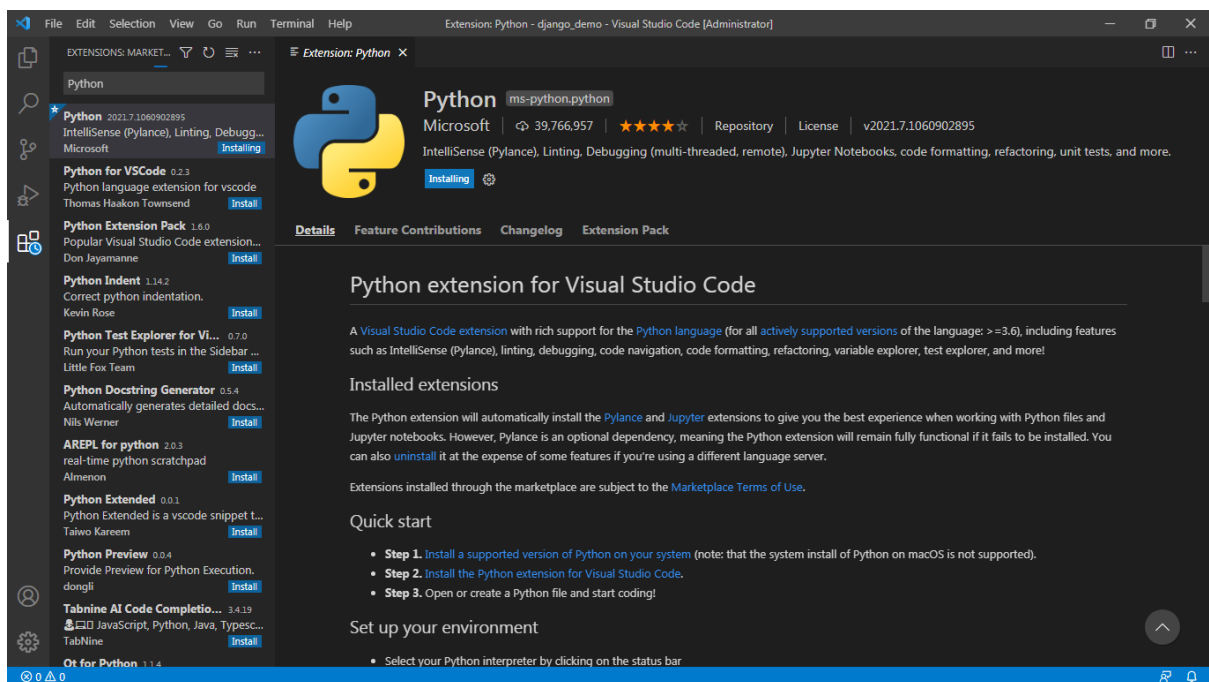
Click on View menu option and select Extensions:



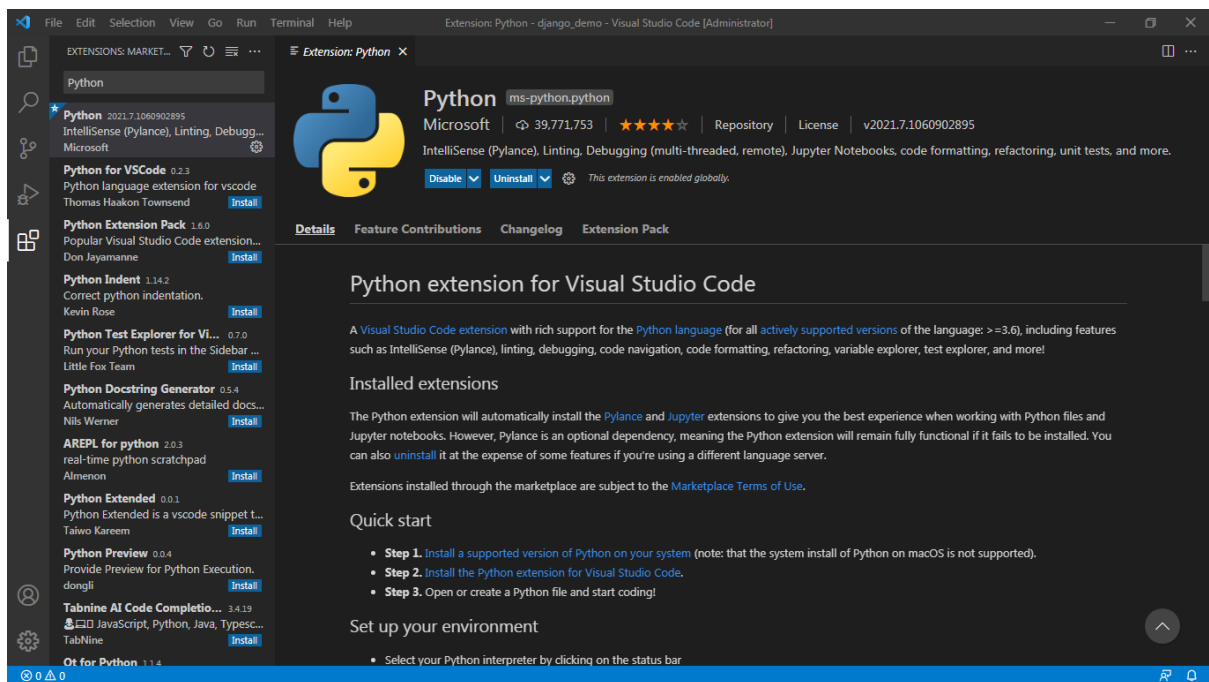
From the available extensions, select the Python extension and install it:



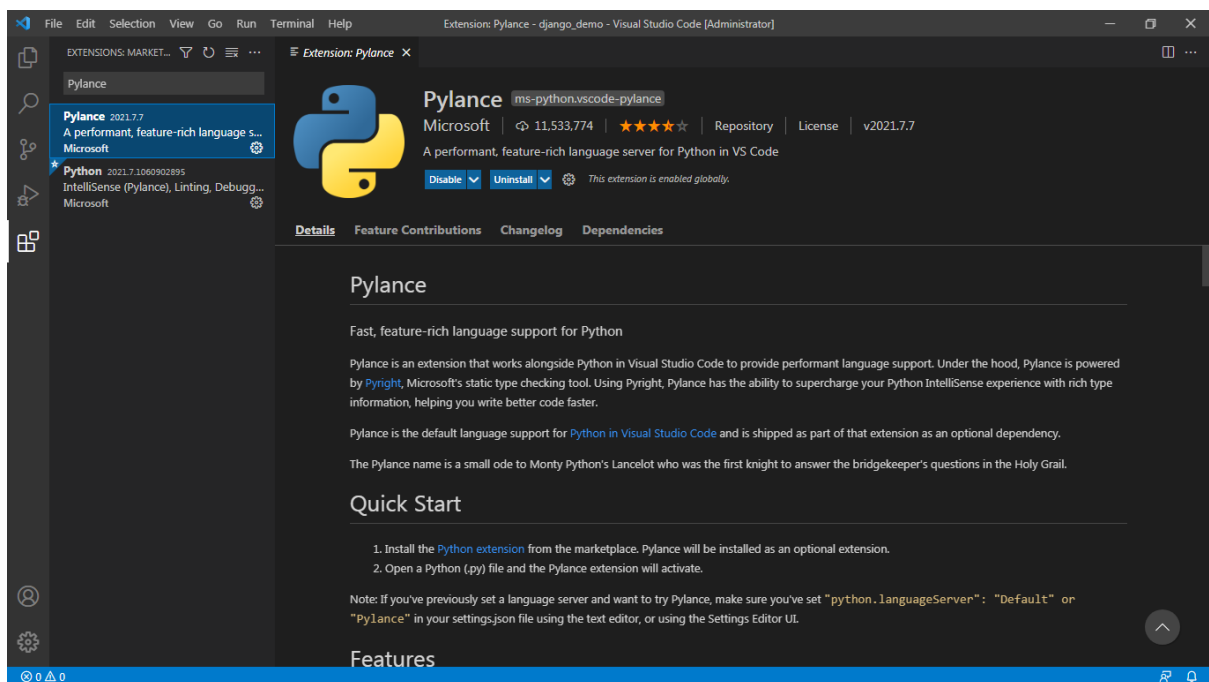
Once you click on install, it will start installing the Python extension:



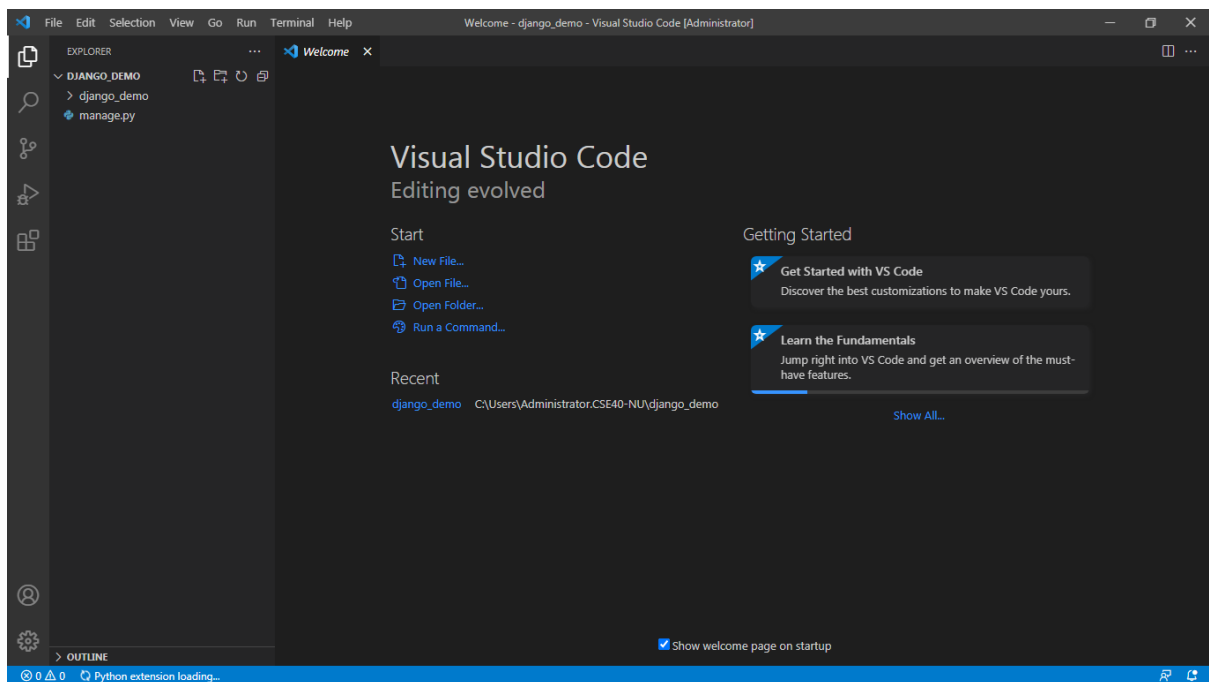
After it is successfully installed, following will appear;



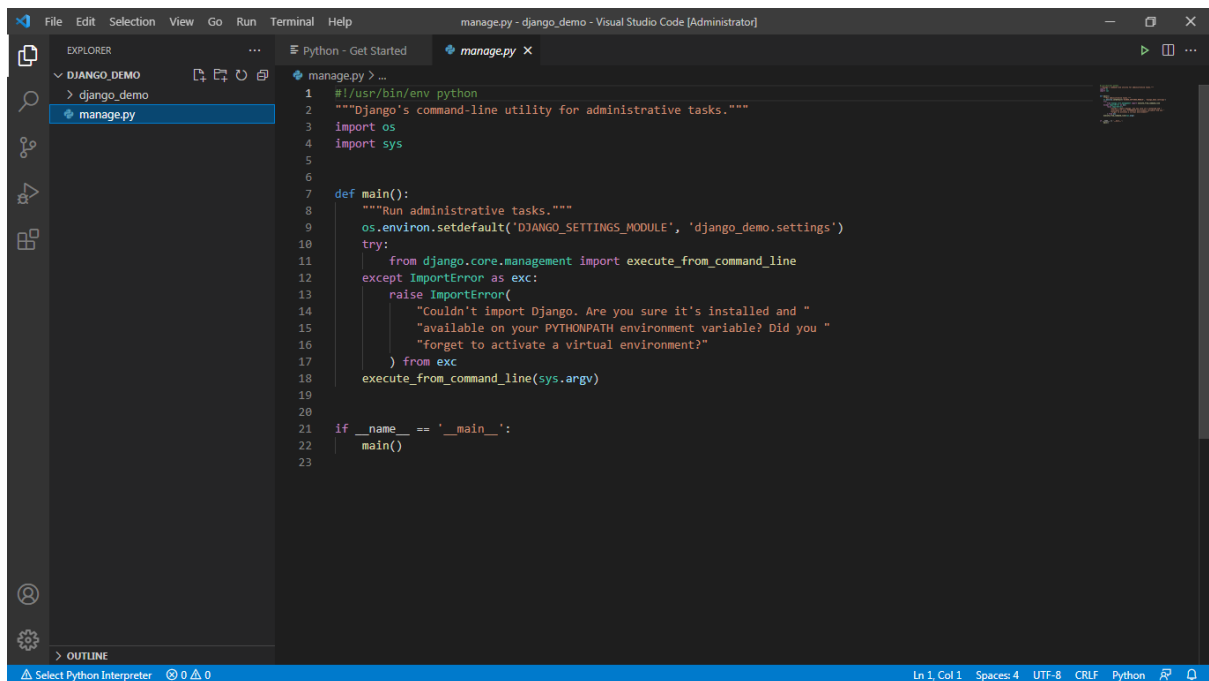
Then install Pylance extension:



#### 4. Go back to the folder i.e django\_demo

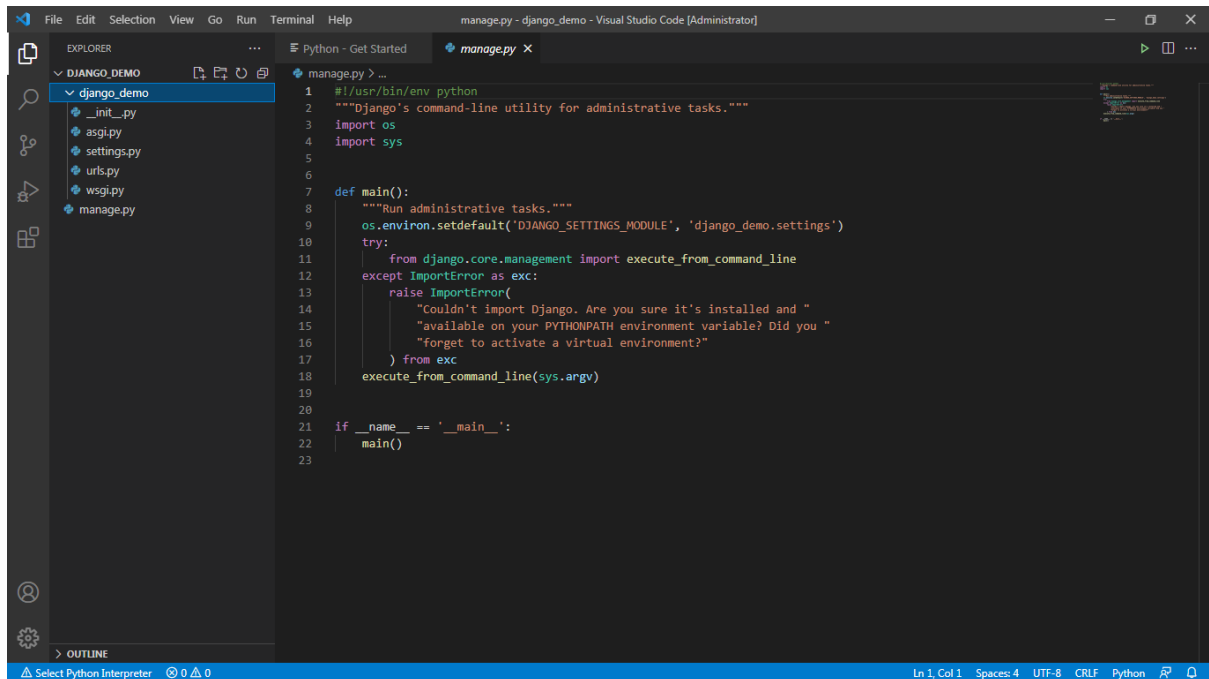


Select the python interpreter (choose the interpreter that you used to install the Django package)i.e default path and then Click on manage.py



manage.py file was created when you used django-admin command to create the django\_demo project

Now select the `django_demo` subfolder inside the `django_demo` main folder, the following files will appear:



The screenshot shows the Visual Studio Code interface with the `django_demo` project open. The Explorer sidebar on the left shows the project structure:

- `DIANGO_DEMO`
  - `django_demo`
    - `__init__.py`
    - `asgi.py`
    - `settings.py`
    - `urls.py`
    - `wsgi.py`
    - `manage.py`

The main editor displays the `manage.py` file, which is a Django management utility script. The code is as follows:

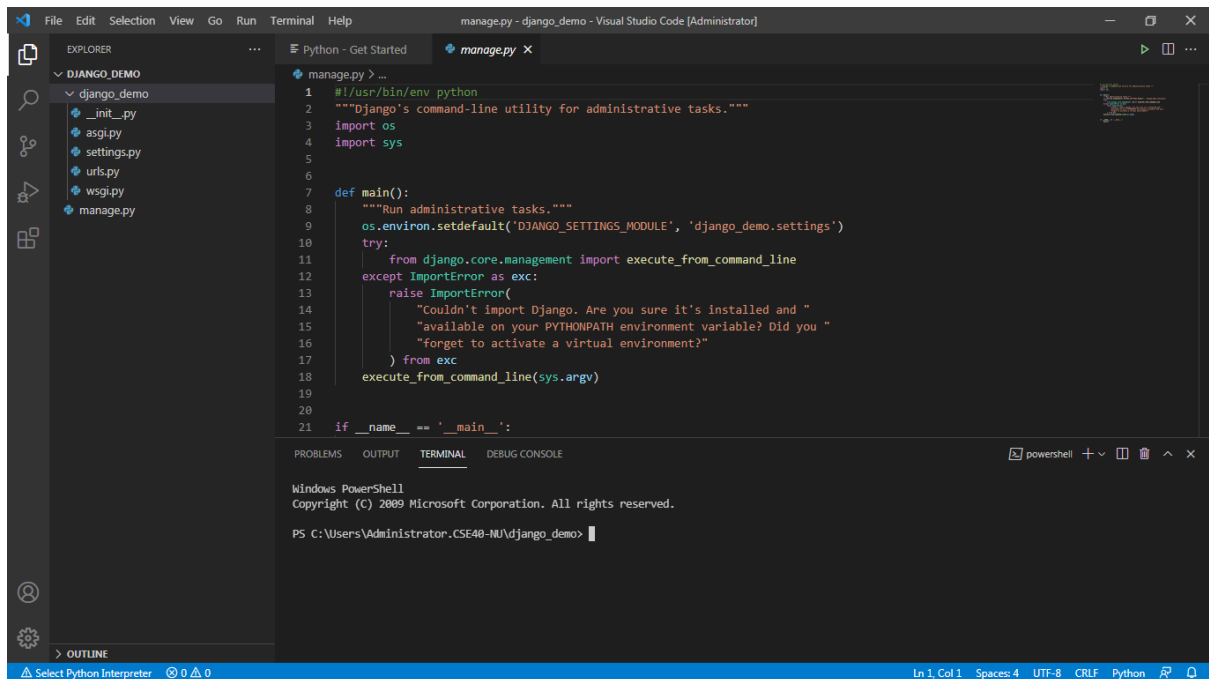
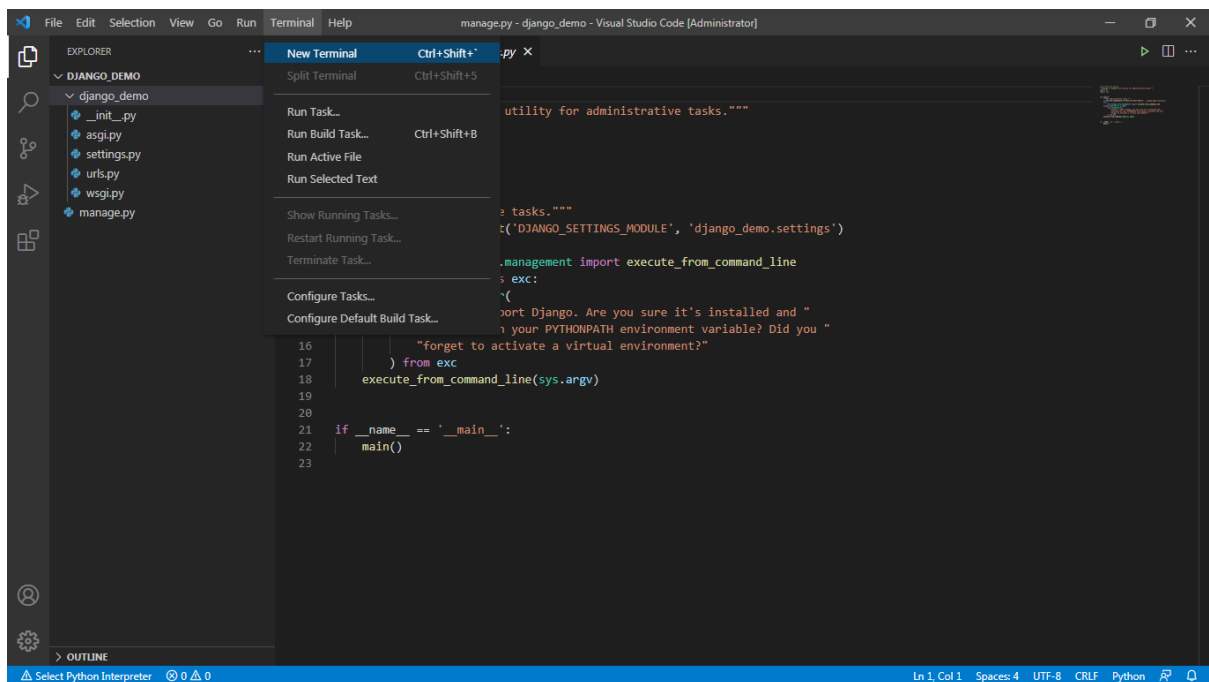
```
1 #!/usr/bin/env python
2 """Django's command-line utility for administrative tasks."""
3 import os
4 import sys
5
6
7 def main():
8     """Run administrative tasks."""
9     os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'django_demo.settings')
10    try:
11        from django.core.management import execute_from_command_line
12    except ImportError as exc:
13        raise ImportError(
14            "Couldn't import Django. Are you sure it's installed and "
15            "available on your PYTHONPATH environment variable? Did you "
16            "forget to activate a virtual environment?"
17        ) from exc
18    execute_from_command_line(sys.argv)
19
20
21 if __name__ == '__main__':
22     main()
23
```

The `wsgi.py` and `asgi.py` files are related to deployment of web applications. This will be discussed later on.

The `settings.py` helps in configuration of the project, keep the default settings in the file as it is.

The `urls.py` file is of significant importance, as it will decide which url's / routes your web application will support. It is related to different pages that the web application will load.

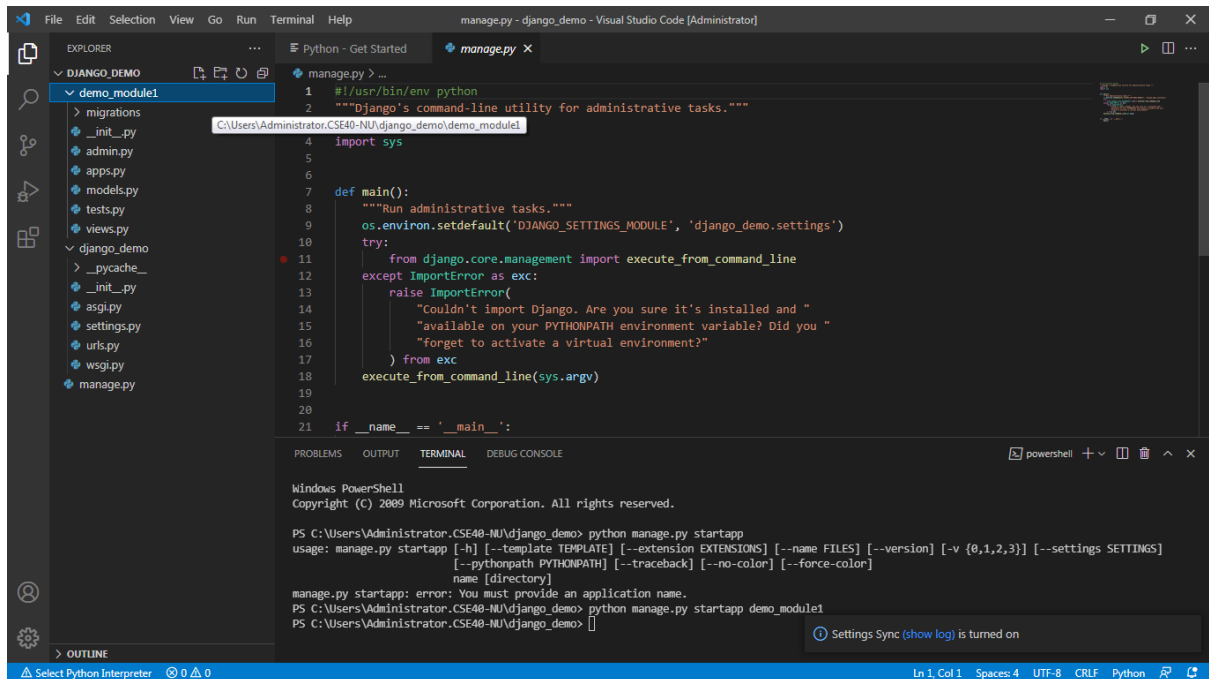
## 5. Open Terminal from vscode:



This can be used to run various commands.



Now in the terminal, run *python manage.py startapp demo\_module1* command  
This will help in creating a module/app for the Django based web application that you want to create.



The screenshot shows the Visual Studio Code interface with a Django project named 'DIANGO\_DEMO'. The Explorer panel on the left shows the project structure, including a 'demo\_module1' directory. The main editor displays the 'manage.py' file, which contains the following code:

```
1 #!/usr/bin/env python
2 """Django's command-line utility for administrative tasks."""
3
4 import sys
5
6
7 def main():
8     """Run administrative tasks."""
9     os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'django_demo.settings')
10    try:
11        from django.core.management import execute_from_command_line
12    except ImportError as exc:
13        raise ImportError(
14            "Couldn't import Django. Are you sure it's installed and "
15            "available on your PYTHONPATH environment variable? Did you "
16            "forget to activate a virtual environment?"
17        ) from exc
18    execute_from_command_line(sys.argv)
19
20
21 if __name__ == '__main__':
22     main()
```

The terminal at the bottom shows the command prompt and the execution of the 'python manage.py startapp demo\_module1' command, which successfully creates the 'demo\_module1' app.

All these files will be discussed in details in next lecture and lab sessions.