

Web Development using django

Django

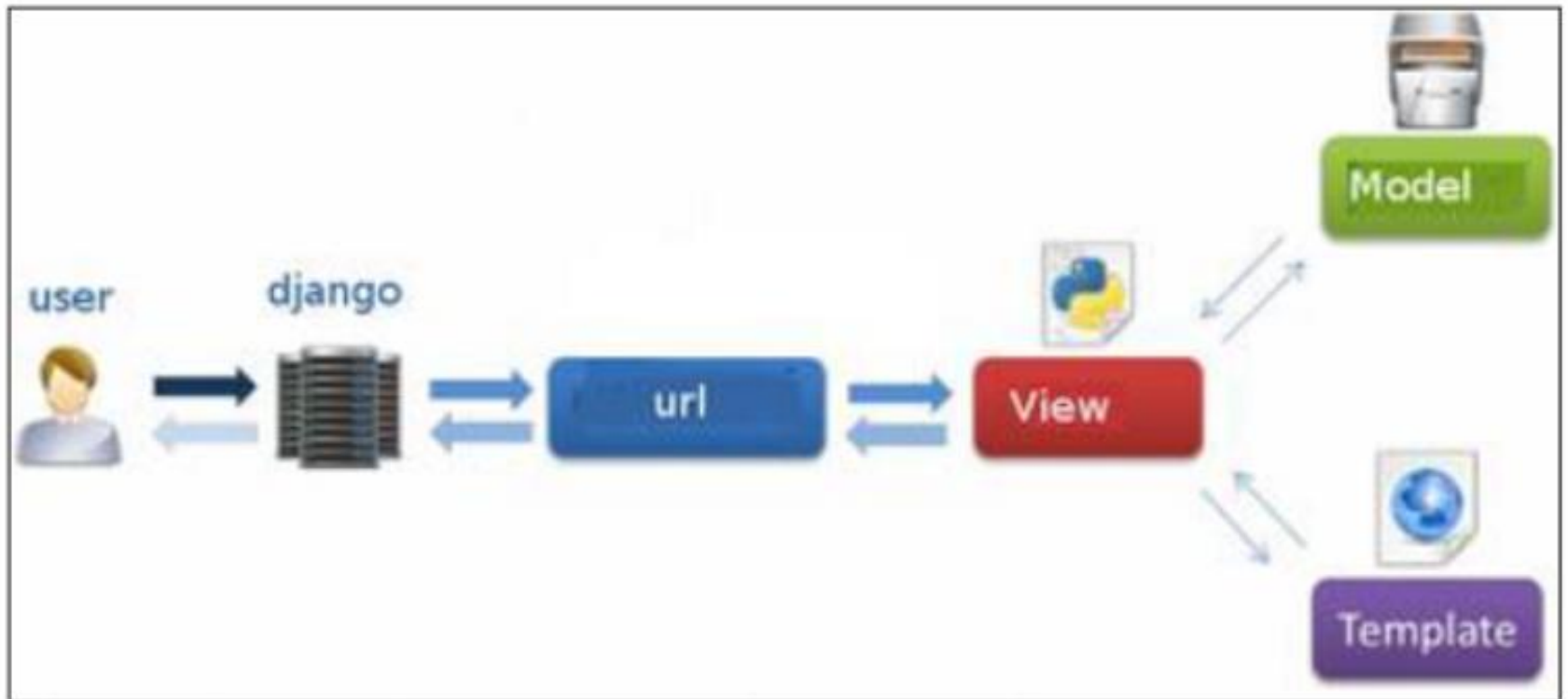
- What is Django ?
- Installation
- Application-1 : Static content
- Application-2 : Dynamic web form application

What is Django ?

- Django is a web development framework that assists in building and maintaining quality web applications.

- **Object-Relational Mapping (ORM) Support** – Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.
- **Multilingual Support** – Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.
- **Framework Support** – Django has built-in support for Ajax, RSS, Caching and various other frameworks.
- **Administration GUI** – Django provides a nice ready-to-use user interface for administrative activities.
- **Development Environment** – Django comes with a lightweight web server to facilitate end-to-end application development and testing.

DJANGO MVT Pattern



Why Django ?

- Fast
- Availability of components
- Scalable
- Secure

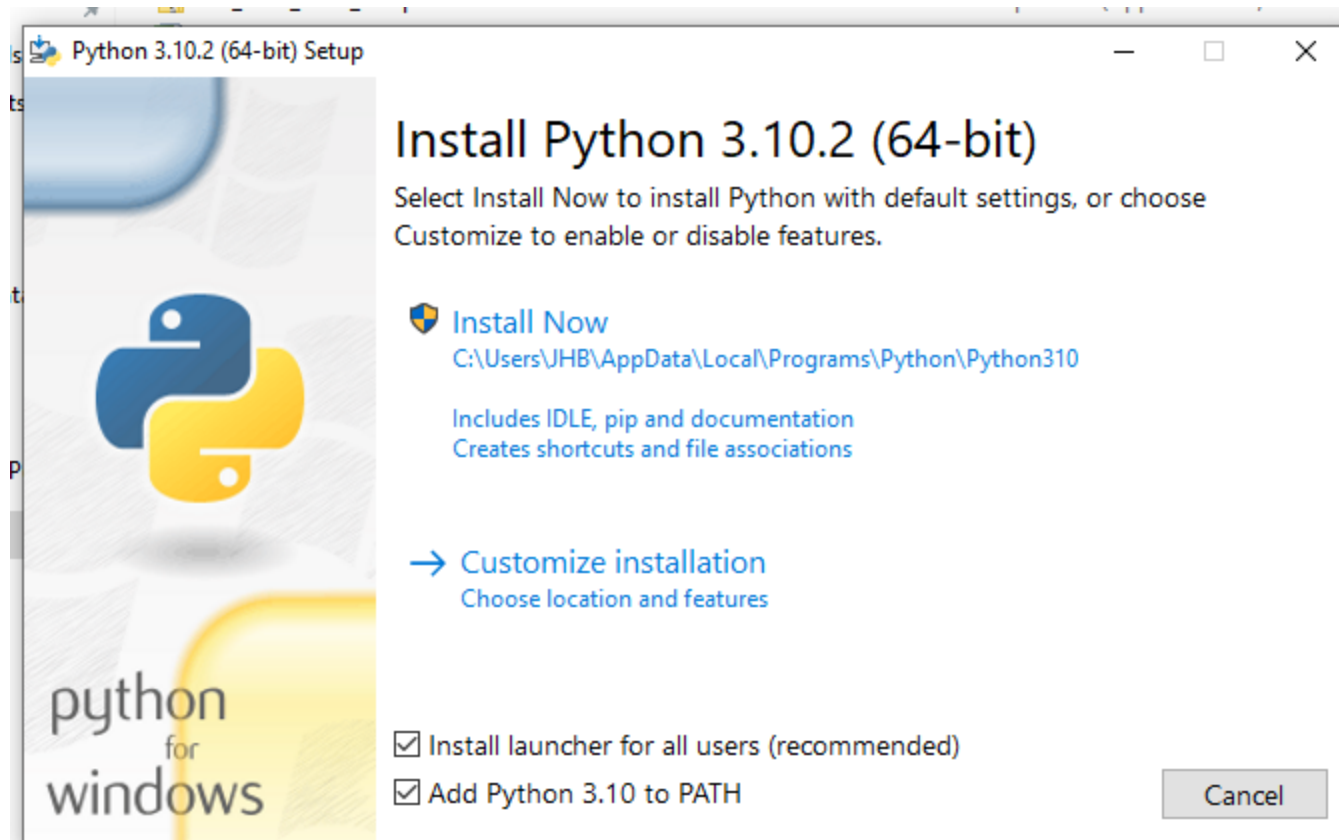
Setting up django and first app

Step-1 : Install Python



From www.python.org download latest version of Python

Step-1 : Install Python



Verify python installation

 C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.1379]  
(c) 2019 Microsoft Corporation. All rights reserved.
```

```
C:\Users\JHB>python --version  
Python 3.10.2
```

```
C:\Users\JHB>
```

Virtual Environment

- A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them. This is one of the most important tools that most of the Python developers use.

Install Virtual Environment

- `pip install virtualenvwrapper-win`

```
C:\Users\JHB>pip install virtualenvwrapper-win
Collecting virtualenvwrapper-win
  Downloading virtualenvwrapper-win-1.2.7-py3-none-any.whl (18 kB)
Requirement already satisfied: virtualenv in c:\users\jhb\appdata\loc
m virtualenvwrapper-win) (20.13.0)
Requirement already satisfied: filelock<4,>=3.2 in c:\users\jhb\appda
s (from virtualenv->virtualenvwrapper-win) (3.4.2)
Requirement already satisfied: distlib<1,>=0.3.1 in c:\users\jhb\appda
es (from virtualenv->virtualenvwrapper-win) (0.3.4)
Requirement already satisfied: platformdirs<3,>=2 in c:\users\jhb\appd
ges (from virtualenv->virtualenvwrapper-win) (2.4.1)
Requirement already satisfied: six<2,>=1.9.0 in c:\users\jhb\appdata\l
from virtualenv->virtualenvwrapper-win) (1.16.0)
Installing collected packages: virtualenvwrapper-win
Successfully installed virtualenvwrapper-win-1.2.7
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is
You should consider upgrading via the 'C:\Users\JHB\AppData\Local\Pro
upgrade pip' command.
```

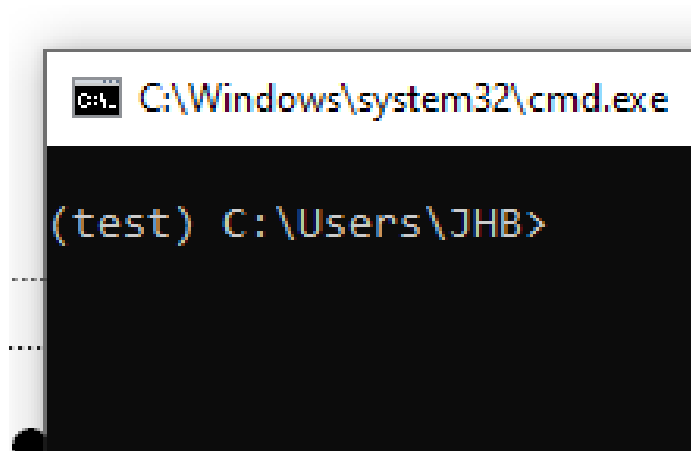
Create a new Virtual environment

- mkvirtualenv test

```
C:\Users\JHB>mkvirtualenv test
created virtual environment CPython3.10.2.
  creator CPython3Windows(dest=C:\Users\JHB\AppData\Local\pypa\virtualenv)
  seeder FromAppData(download=False, pip=b
added seed packages: pip==21.3.1, setu
activators BashActivator,BatchActivator,

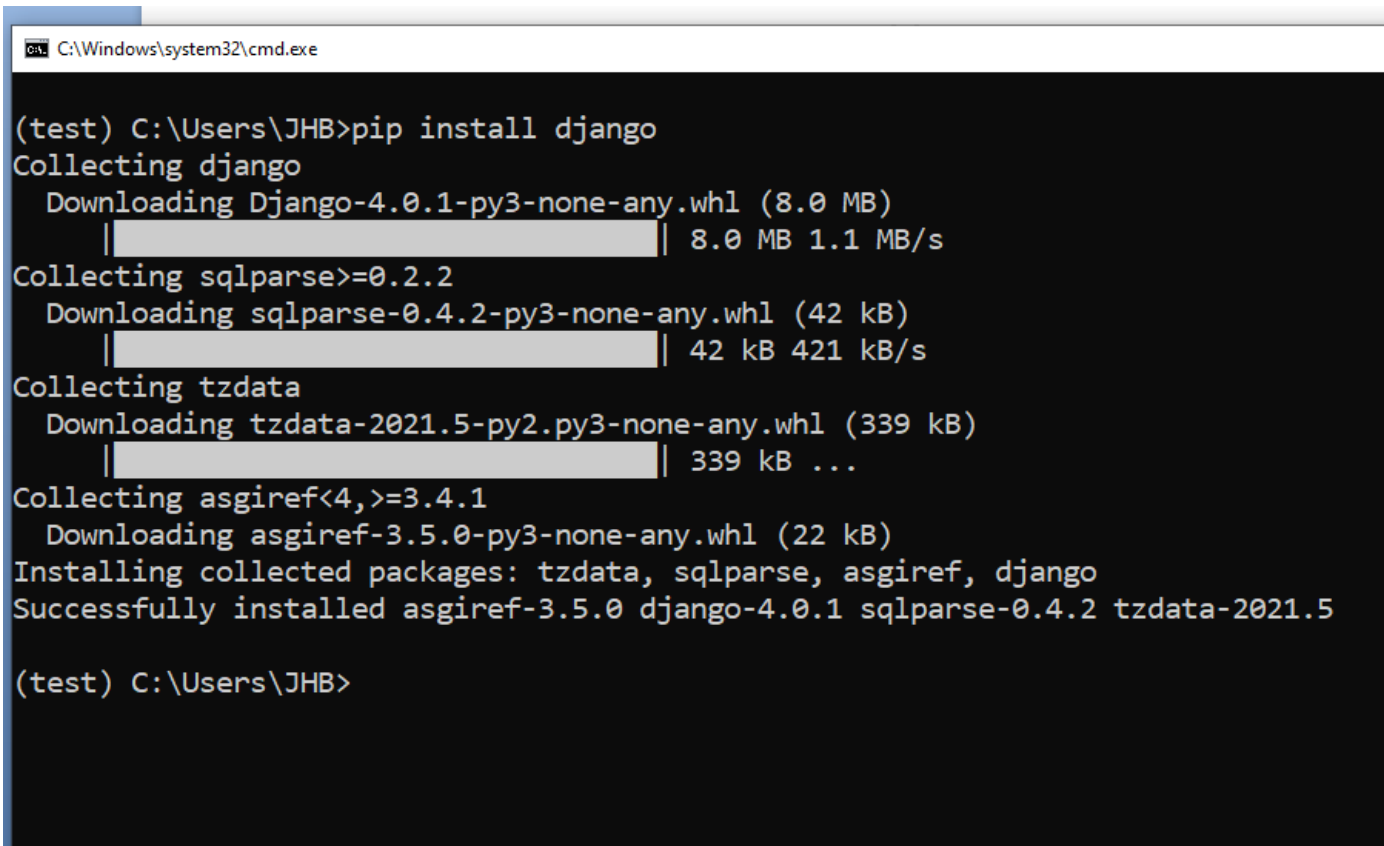
(test) C:\Users\JHB>
```

Virtual Environment



Install django

- pip install django

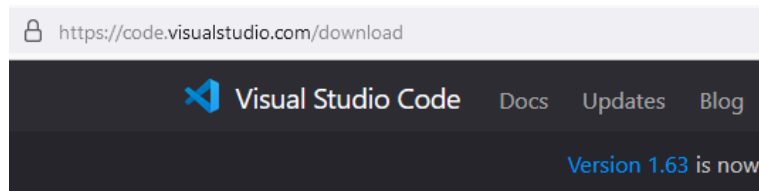


```
C:\Windows\system32\cmd.exe

(test) C:\Users\JHB>pip install django
Collecting django
  Downloading Django-4.0.1-py3-none-any.whl (8.0 MB)
    |████████████████████████████████████████| 8.0 MB 1.1 MB/s
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
    |████████████████████████████████████████| 42 kB 421 kB/s
Collecting tzdata
  Downloading tzdata-2021.5-py2.py3-none-any.whl (339 kB)
    |████████████████████████████████████████| 339 kB ...
Collecting asgiref<4,>=3.4.1
  Downloading asgiref-3.5.0-py3-none-any.whl (22 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.0 django-4.0.1 sqlparse-0.4.2 tzdata-2021.5

(test) C:\Users\JHB>
```

Install VS Code editor



Down

Free and built on c

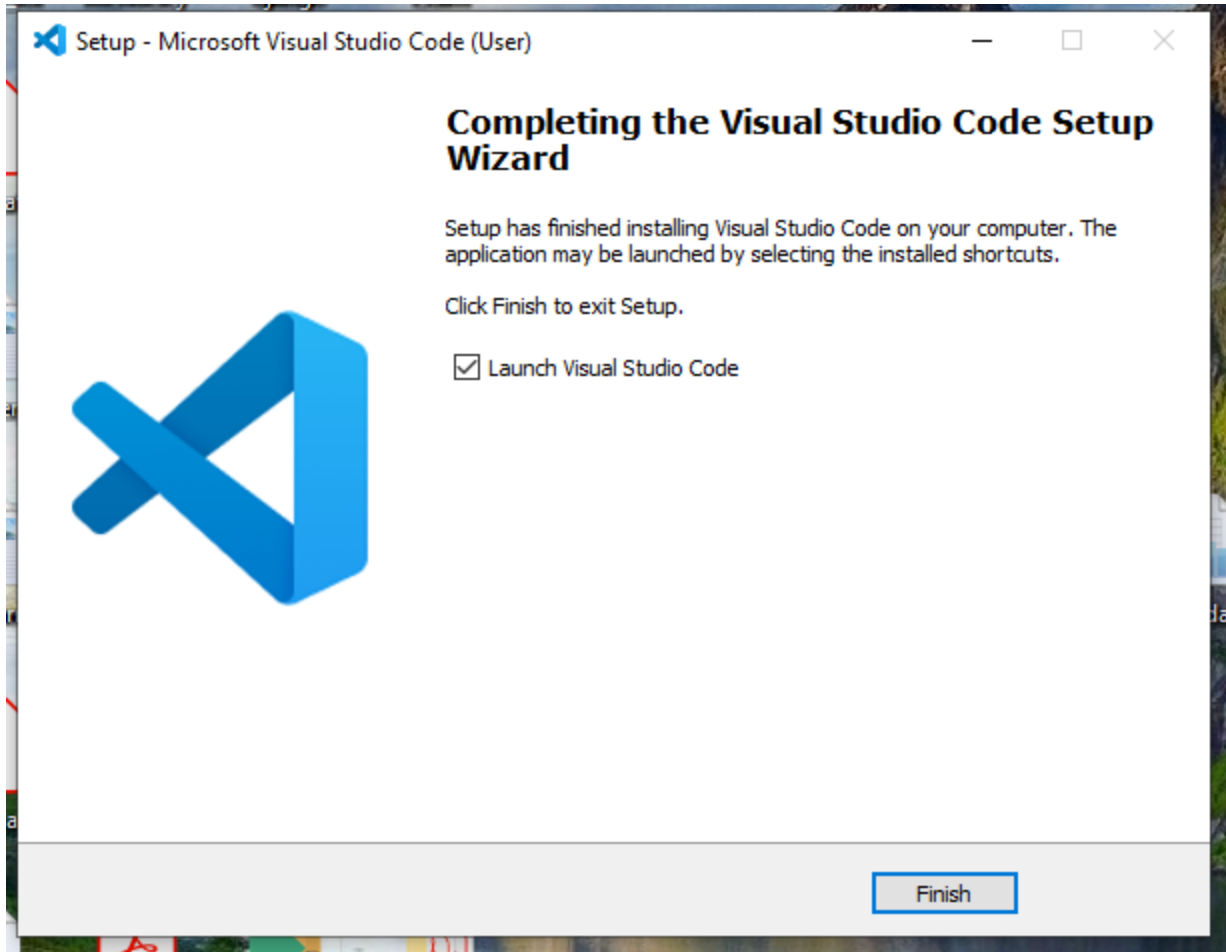


↓ Windows

Windows 7, 8, 10, 11

| | | | |
|------------------|--------|--------|-----|
| User Installer | 64 bit | 32 bit | ARM |
| System Installer | 64 bit | 32 bit | ARM |
| .zip | 64 bit | 32 bit | ARM |

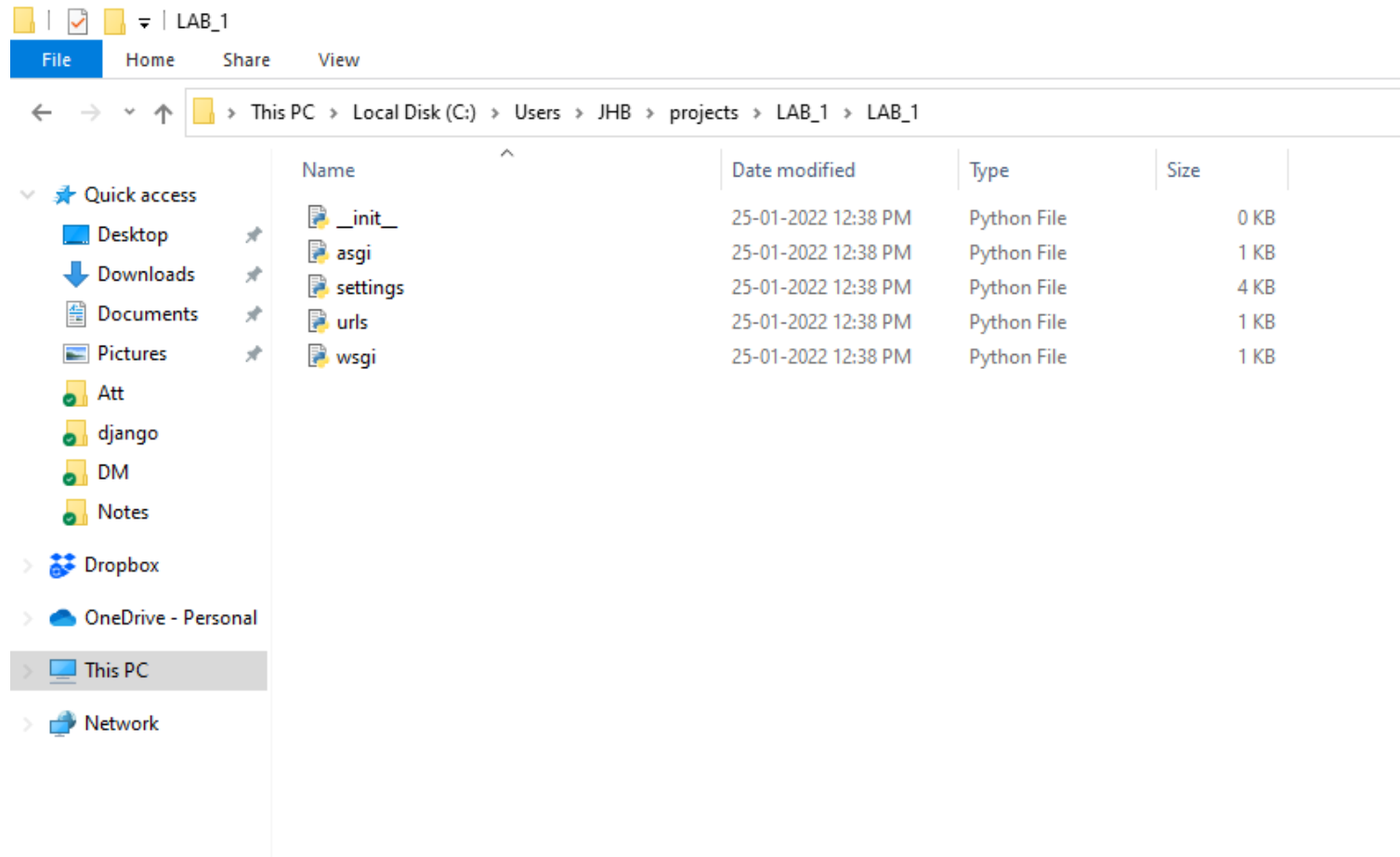
VS Code



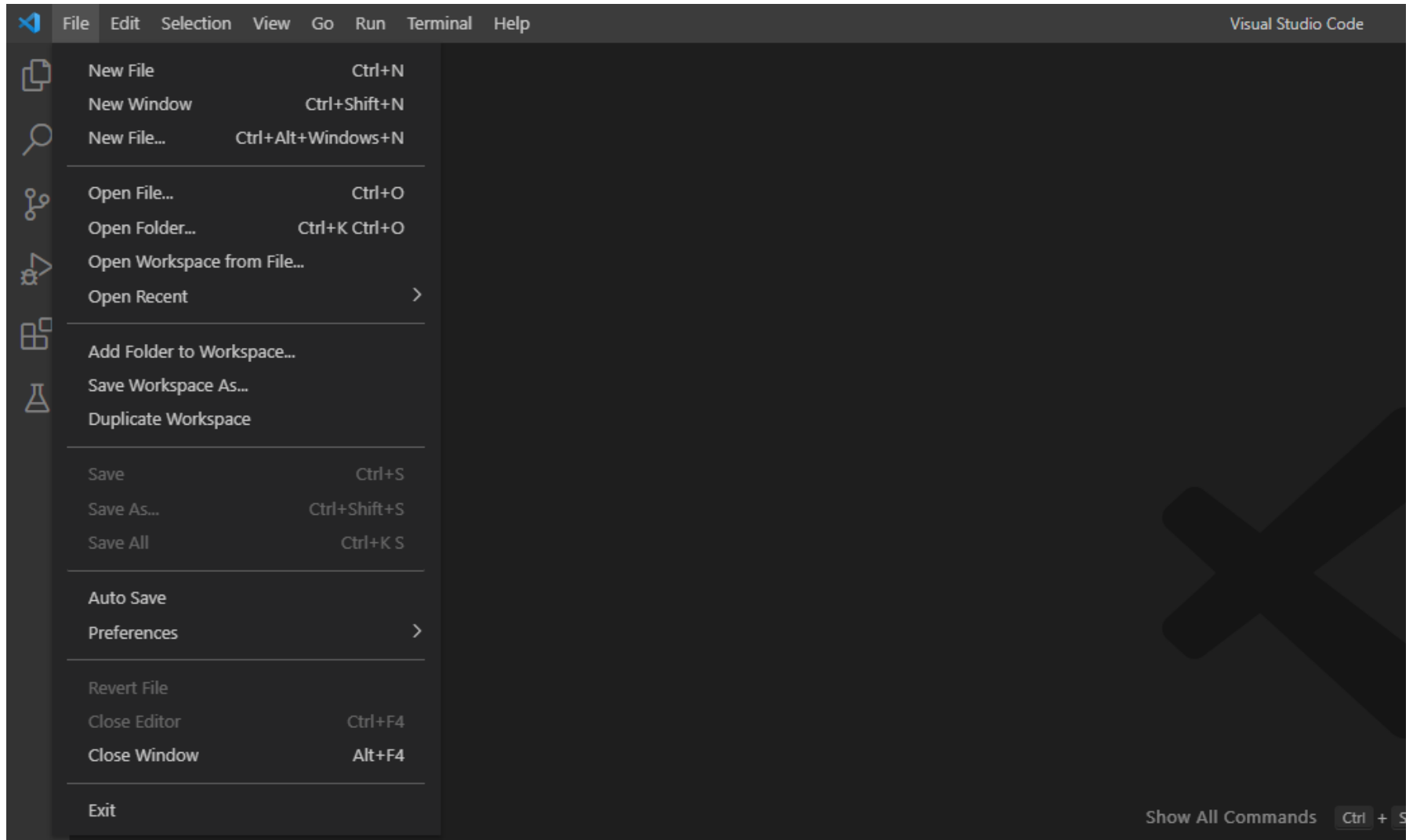
Setting up project

- `mkdir project`
- `cd project`
- `django-admin startproject LAB_1`

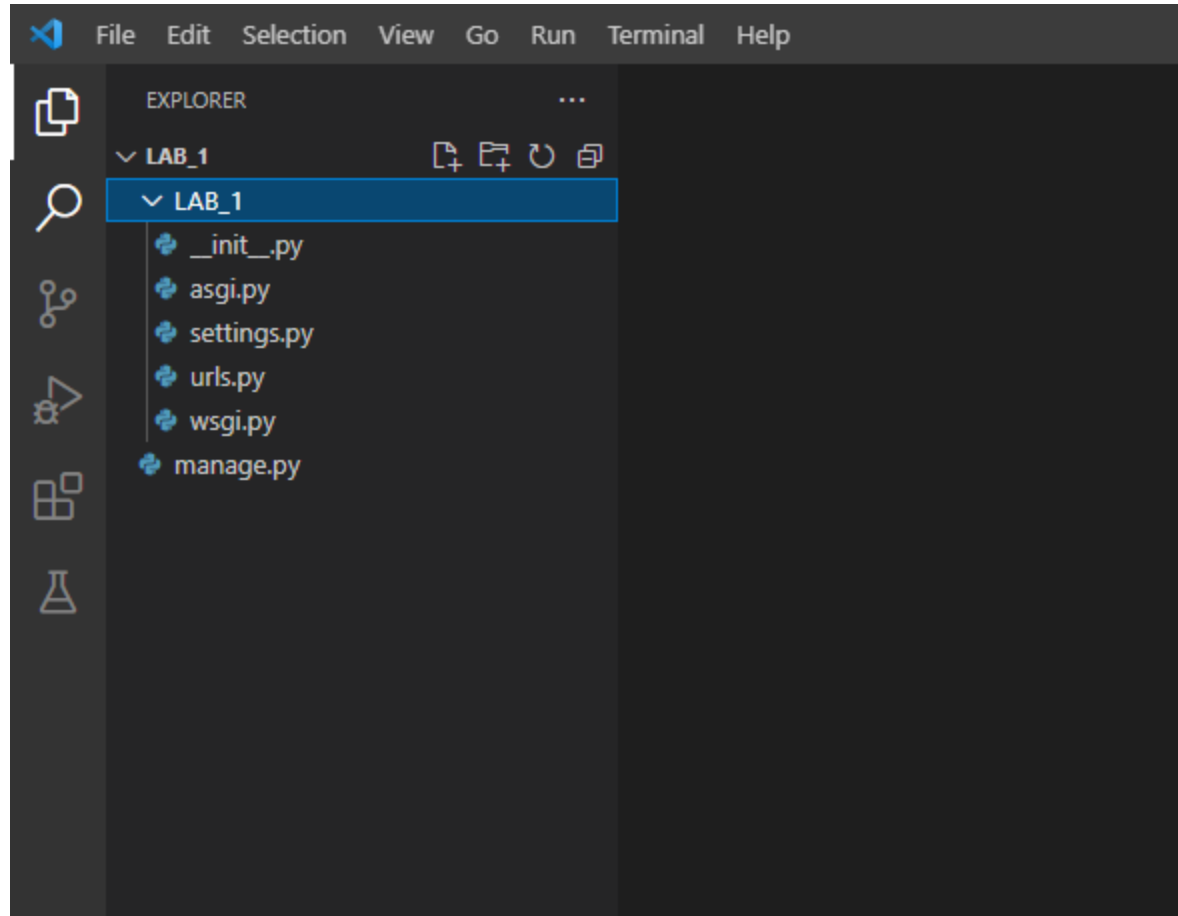
First project



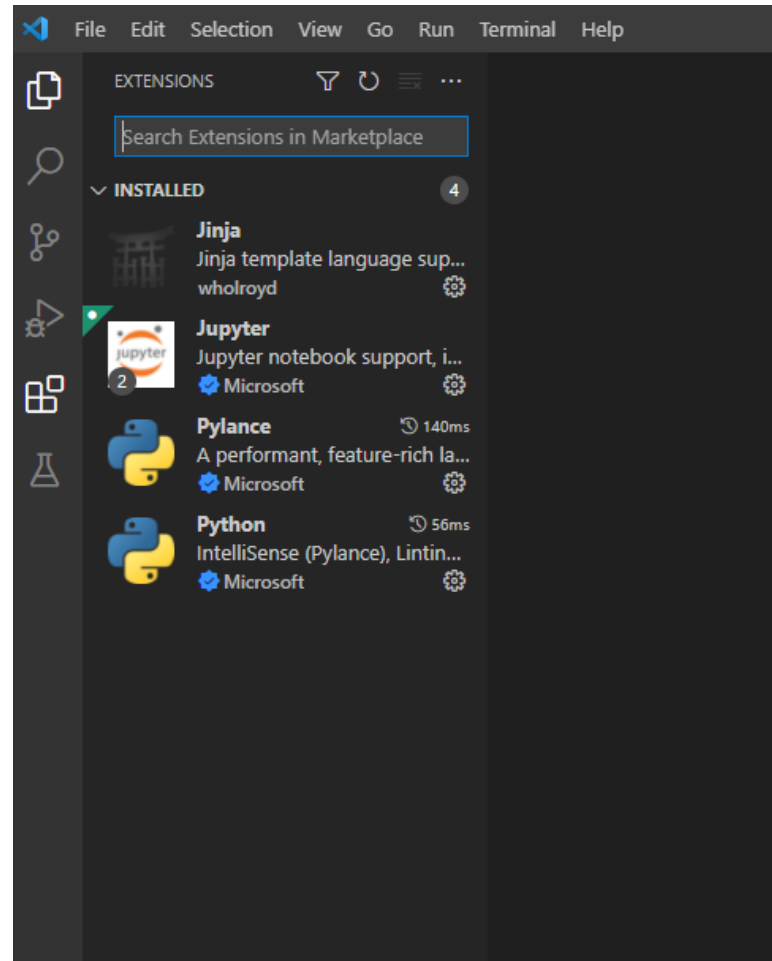
VS Code



VS Code



Python extension



Python extension

Extension: Python - LAB_1 - Visual Studio Code



Python

v2021.12.1559732655

Microsoft | 49,324,777 | ★★★★★ (461)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code f

[Reload Required](#) [Install](#) ⚙️

[Details](#) [Feature Contributions](#) [Changelog](#) [Extension Pack](#) [Runtime Status](#)

Python extension for Visual Studio Code

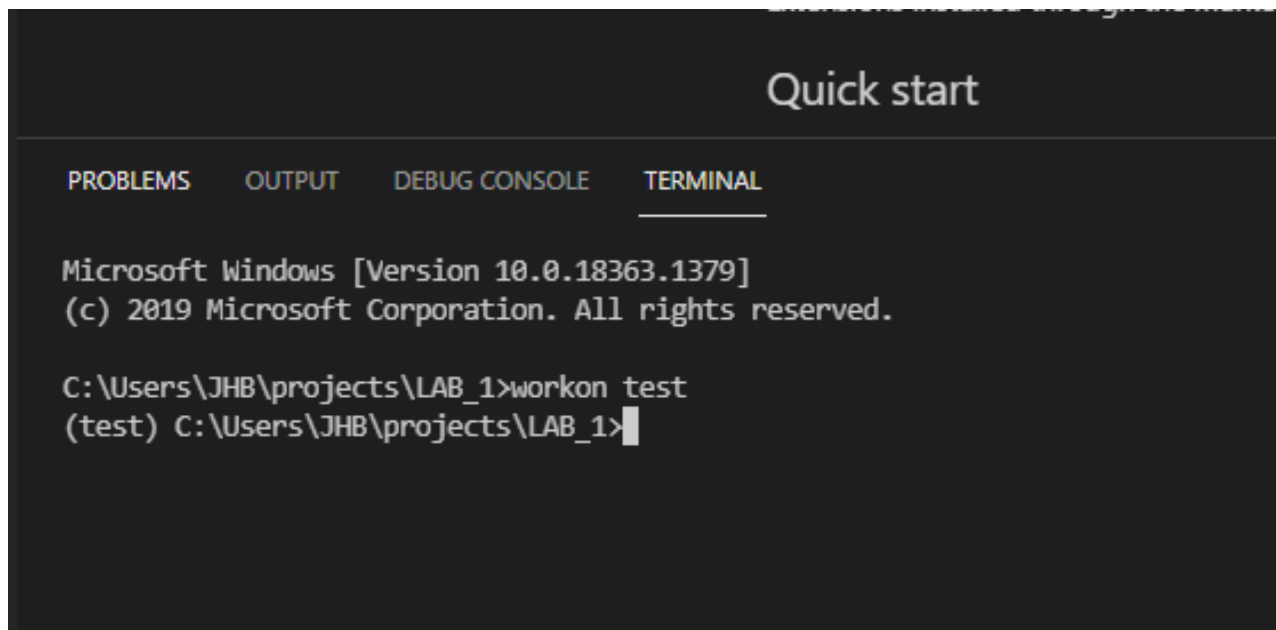
A [Visual Studio Code extension](#) with rich support for the [Python language](#) (for all [actively supported versions](#) of the language: >=3.6), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Web support

The Python extension offers [limited support](#) when running on the web (for example, on [github.dev](#)), by providing partial IntelliSense for open files on the editor.

Working with virtual environment

workon test



The screenshot shows a VS Code terminal window with a dark theme. At the top, there's a 'Quick start' header. Below it, a tab bar contains 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal content shows the Windows version and copyright information, followed by the command 'C:\Users\JHB\projects\LAB_1>workon test' and its output '(test) C:\Users\JHB\projects\LAB_1>' with a cursor at the end.

```
Quick start

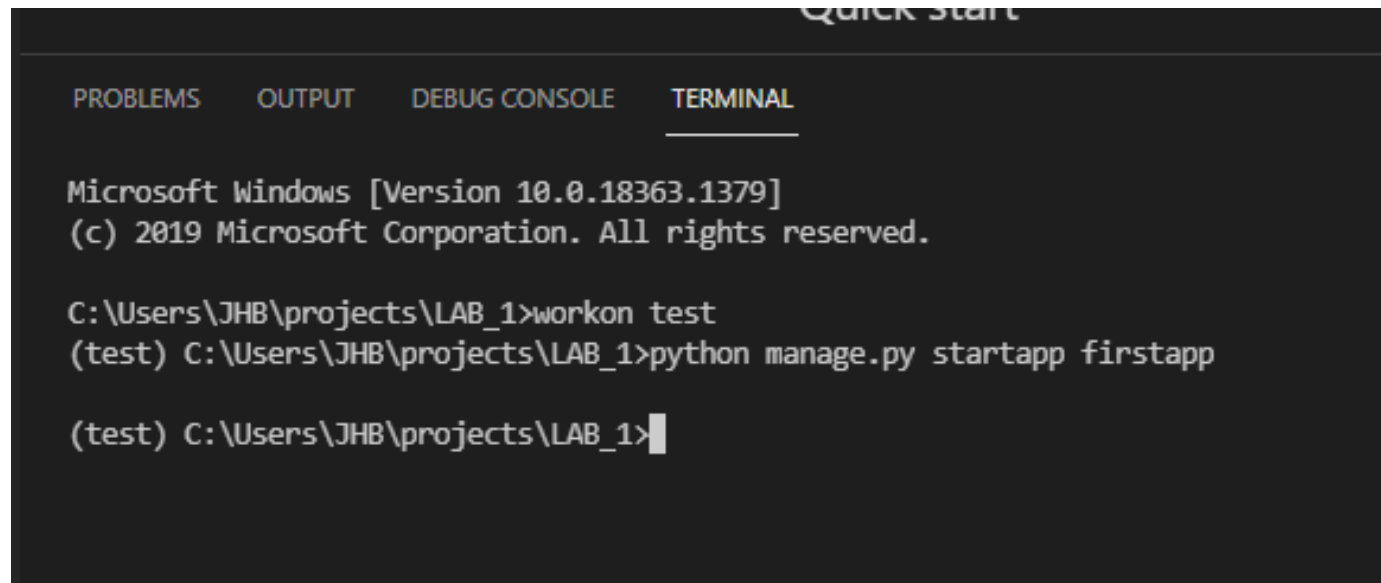
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\JHB\projects\LAB_1>workon test
(test) C:\Users\JHB\projects\LAB_1>
```


Create App

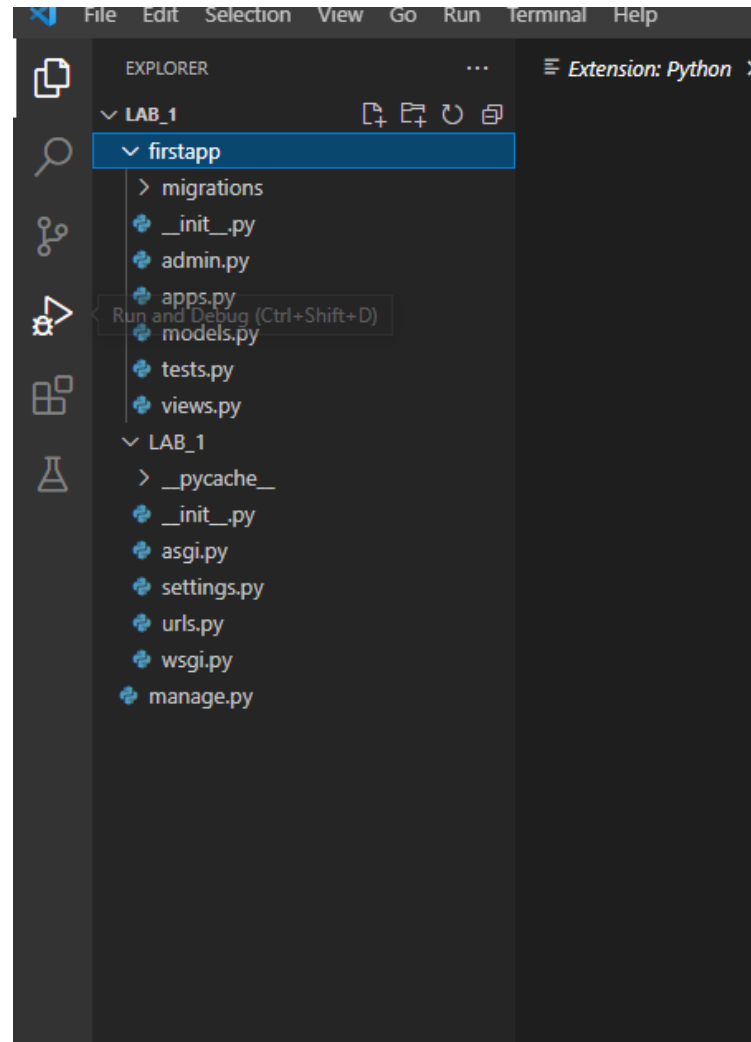
Python manage.py startapp firstapp



The image shows a screenshot of a Visual Studio Code terminal window. At the top, there is a tab labeled "Quick Start". Below the tab, there are four buttons: "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "TERMINAL" button is selected and underlined. The terminal output shows the following text:

```
Microsoft Windows [Version 10.0.18363.1379]  
(c) 2019 Microsoft Corporation. All rights reserved.  
  
C:\Users\JHB\projects\LAB_1>workon test  
(test) C:\Users\JHB\projects\LAB_1>python manage.py startapp firstapp  
  
(test) C:\Users\JHB\projects\LAB_1>
```

VS Code



Start Server

Python manage.py runserver

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply
Run 'python manage.py migrate' to apply them.
January 25, 2022 - 12:50:37
Django version 4.0.1, using settings 'LAB_1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
█
```



🔒 127.0.0.1:8000

django

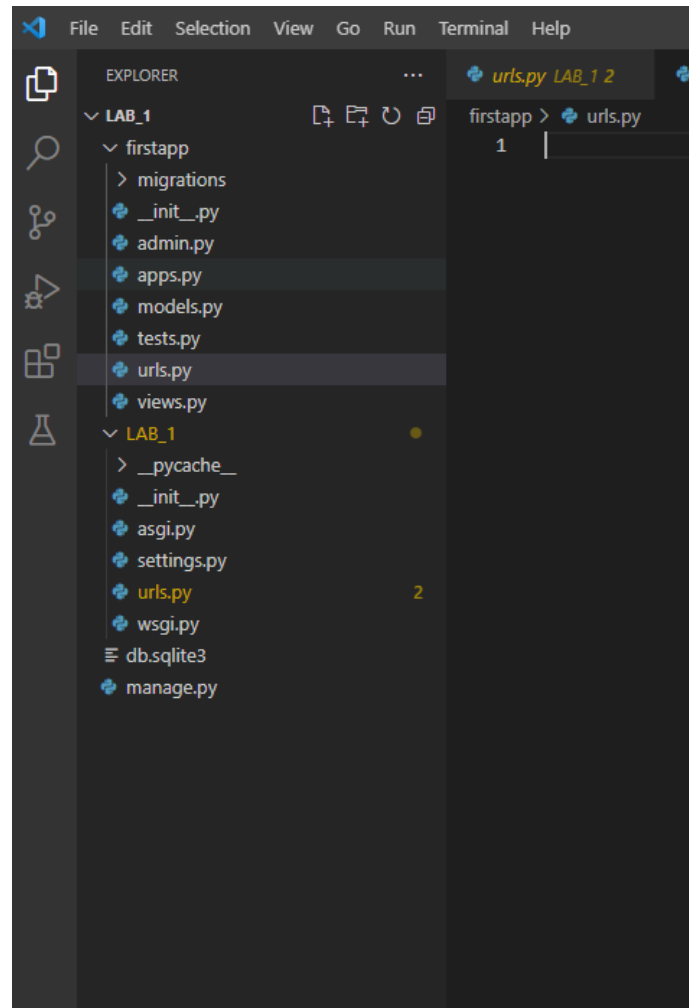
View [release notes](#) for Django 4.0



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.

Create urls.py in firstapp



Code in firstapp/urls.py

```
urls.py LAB_1 2  urls.py firstapp 2 x
firstapp > urls.py > ...
1 from django.urls import path
2 from django.urls.resolvers import URLPattern
3
4 from . import views
5
6 urlpatterns=[
7     path('', views.home, name='home')
8 ]
```

Code for firstapp/views.py

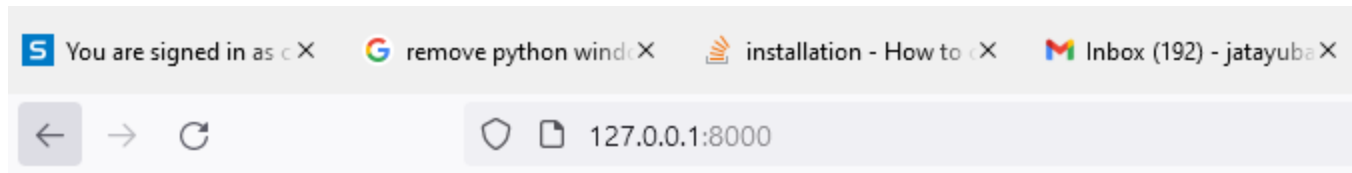
```
✓ from django.http.response import HttpResponse
  from django.shortcuts import render
  from django.http import HttpResponse
  # Create your views here.

✓ def home(request):
    return HttpResponse("Hello World");
```

Code for LAB_1/urls.py

```
16 ~ from django.contrib import admin
17  from django.urls import path
18  from django.urls.conf import include
19  from django.conf import settings
20  from django.conf.urls.static import static
21
22 ~ urlpatterns = [
23     path('admin/', admin.site.urls),
24     path('', include('firstapp.urls'))
25
26 ]
27
```

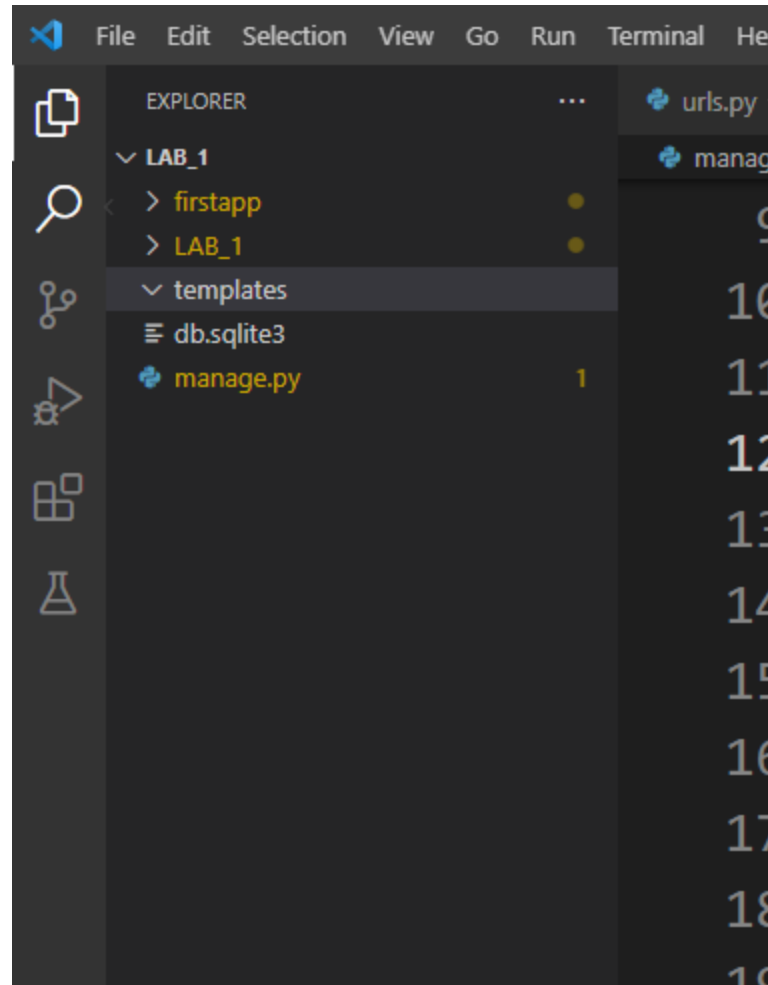

Output



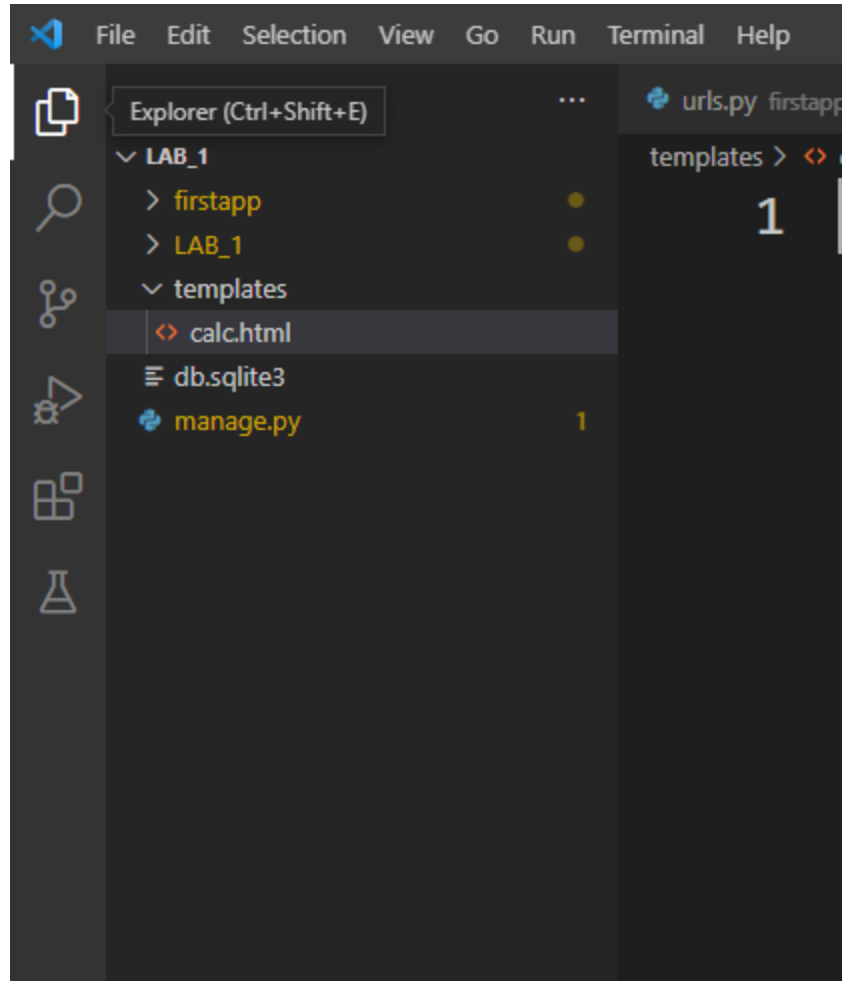
Hello World

Add Two Numbers

Add templates folder



Add calc.html file



Add templates path in settings.py of the main project

```
ROOT_URLCONF = 'LAB_1.urls'

✓ TEMPLATES = [
✓     {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
✓     'OPTIONS': {
✓         'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
```

Calc.html

```
urls.py firstapp 2 | views.py 3 | <> calc.html x | settings.py | tests.py 1 | urls.py LAB_1
templates > <> calc.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     Hello this is calc page.
11 </body>
12 </html>
```