Web Development using djangos

Django

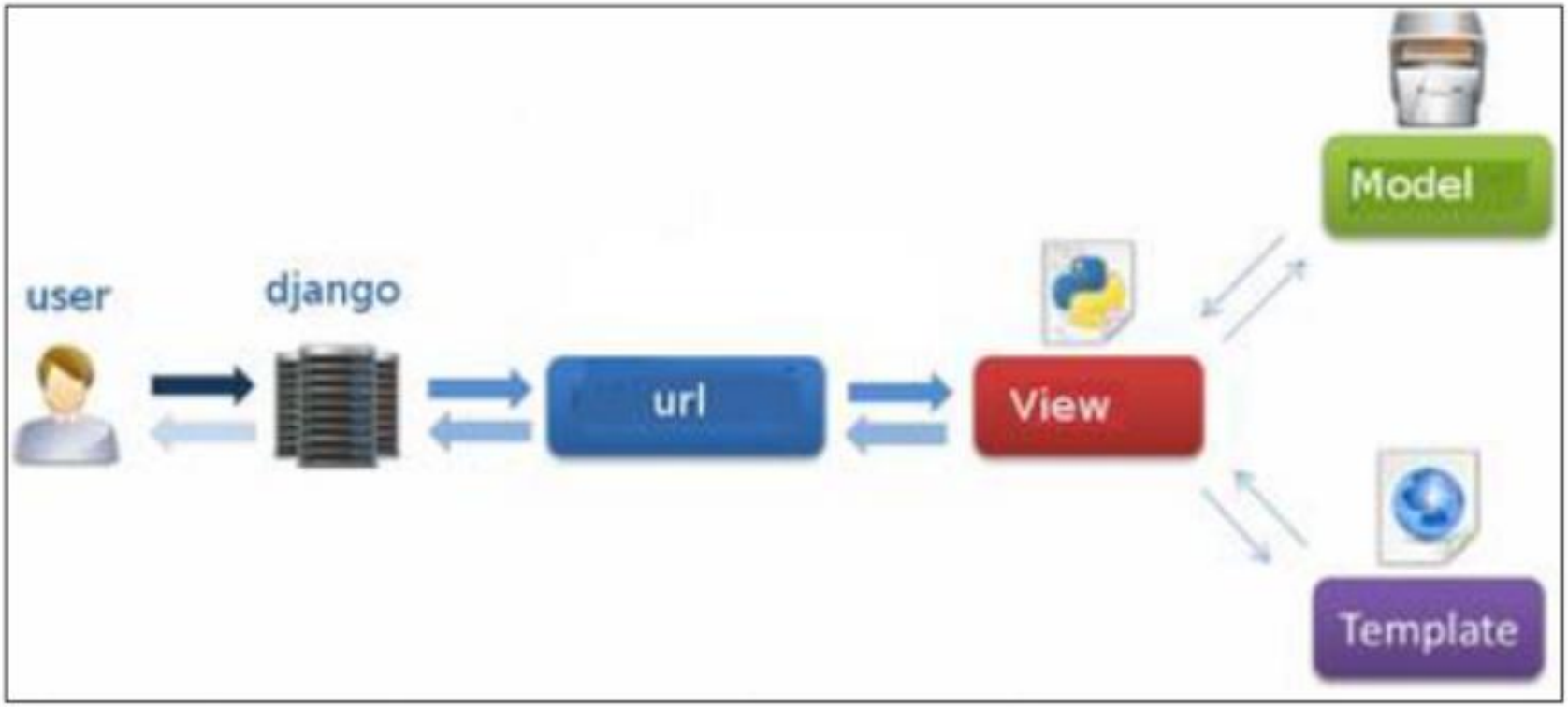
- What is Django ?
- Installation
- Application-1 : Static content
- Application-2 : Dynamic web form application

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 ?

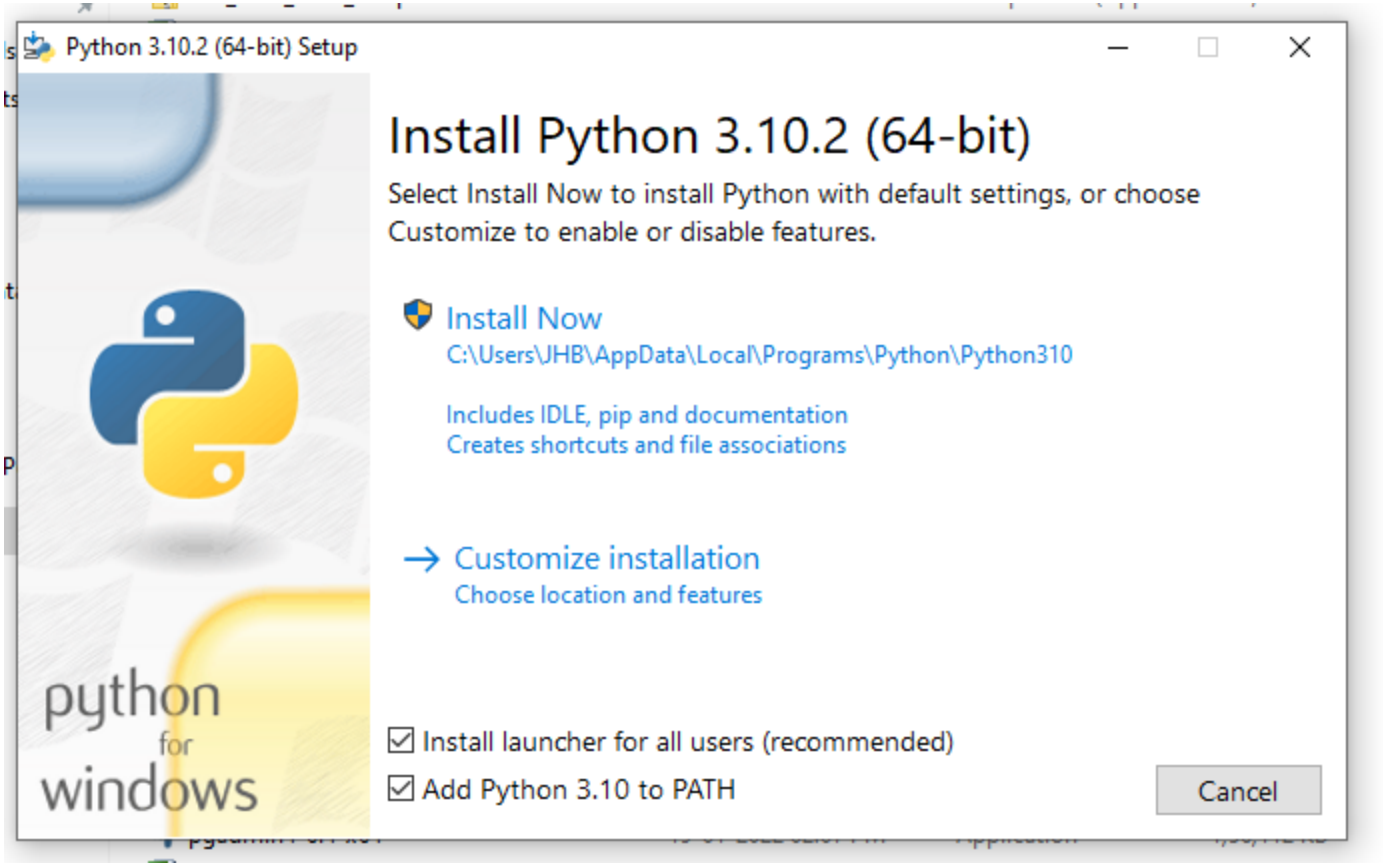
- Fast
- Availability of components
- Scalable
- Secure

Setting up django and first app

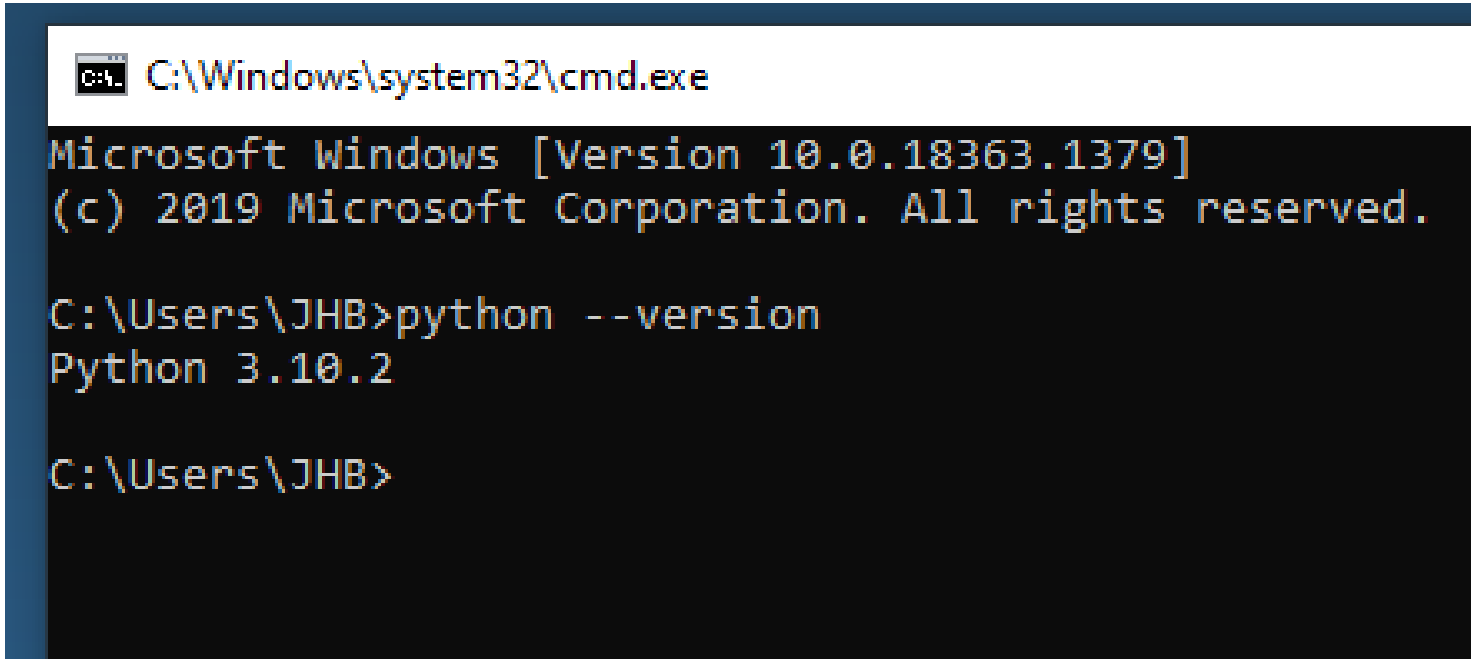
Step-1 : Install Python

From www.python.org download latest version of Python

Step-1 : Install Python



Verify python installation



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.

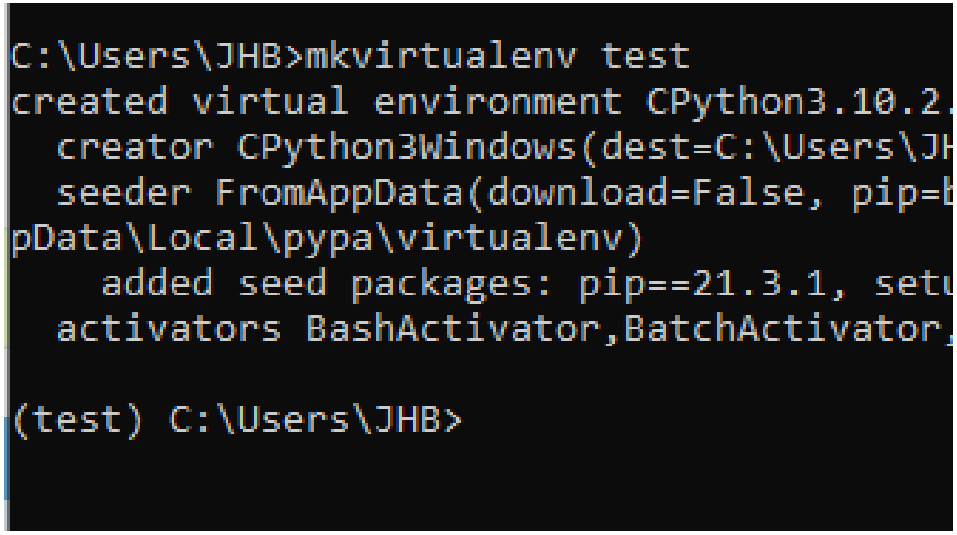
Install Virtual Environment

- pip install virtualenvwrapper-win

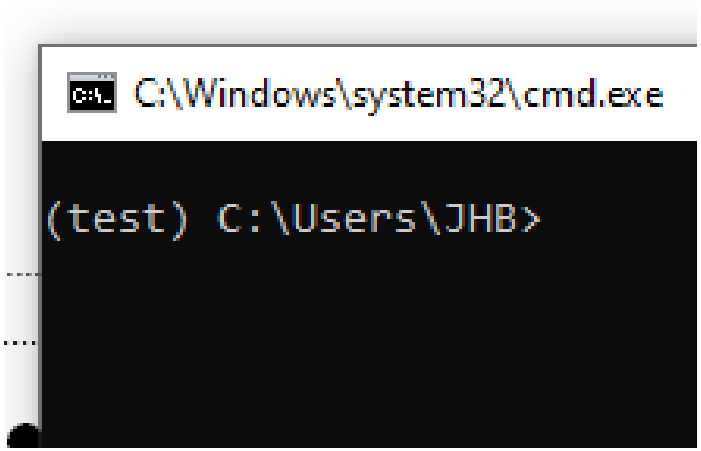
```
C:\Users\JHB>pip install virtualenvwrapper-win
Collecting virtualenvwrapper-win
  Downloading virtualenvwrapper_win-1.2.7-py3-none-any.whl (18 kB)
Requirement already satisfied: virtualenv in c:\users\jhb\appdata\loca
m virtualenvwrapper-win) (20.13.0)
Requirement already satisfied: filelock<4,>=3.2 in c:\users\jhb\appdat
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\Prog
upgrade pip' command.
```

Create a new Virtual environment

- mkvirtualenv test

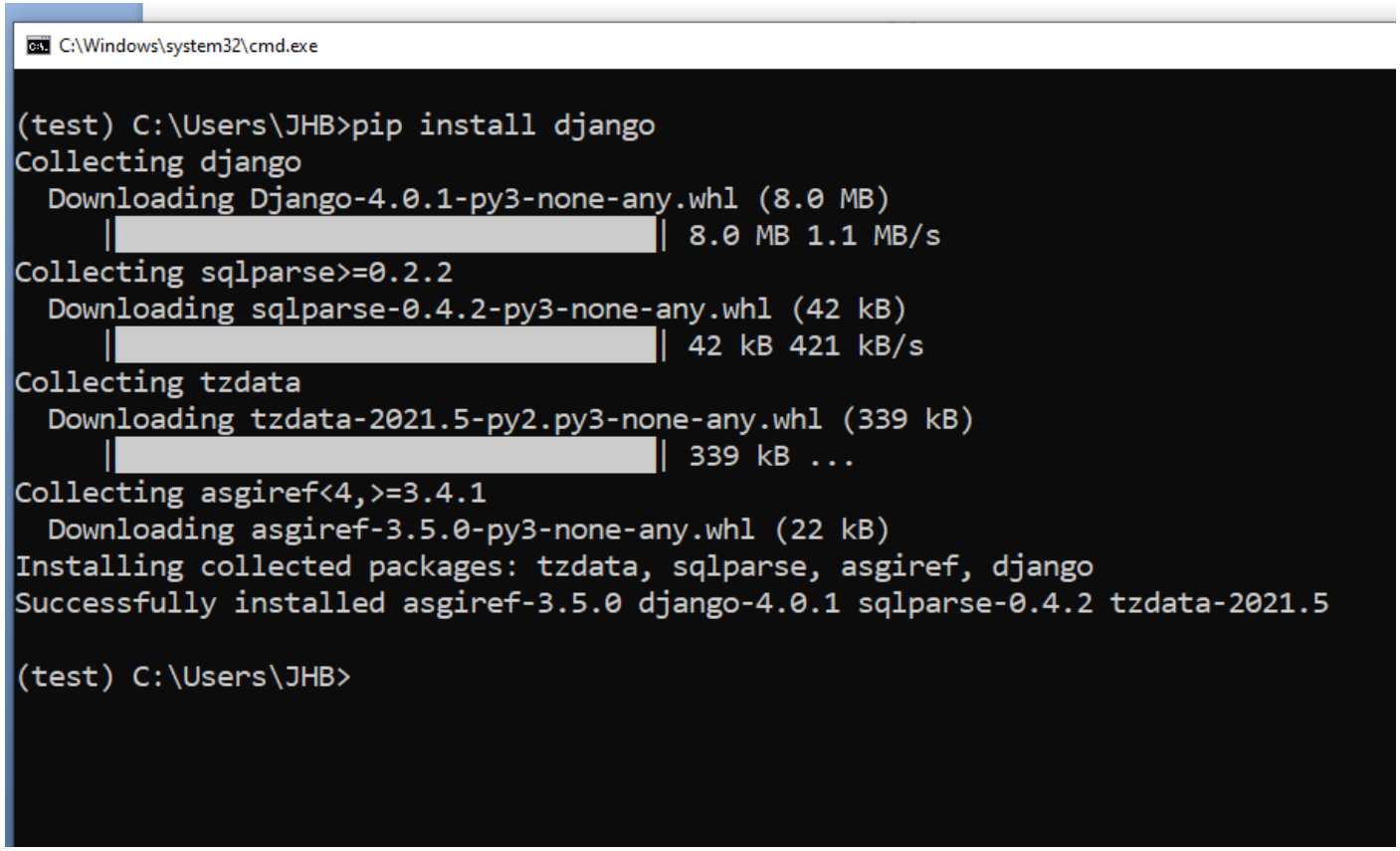


Virtual Environment

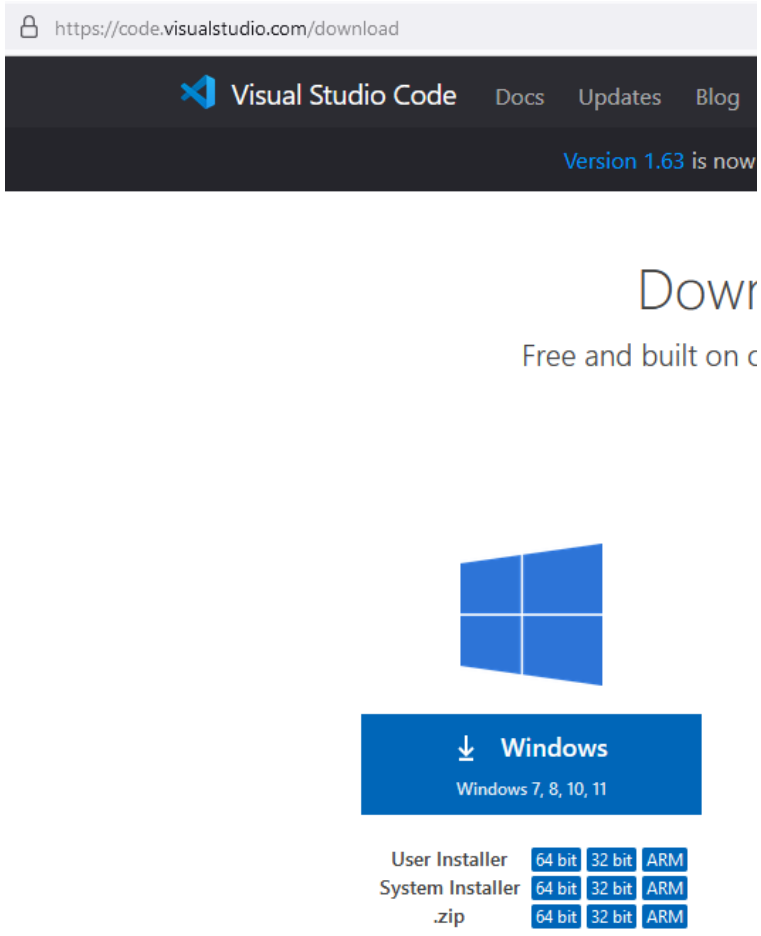


Install django

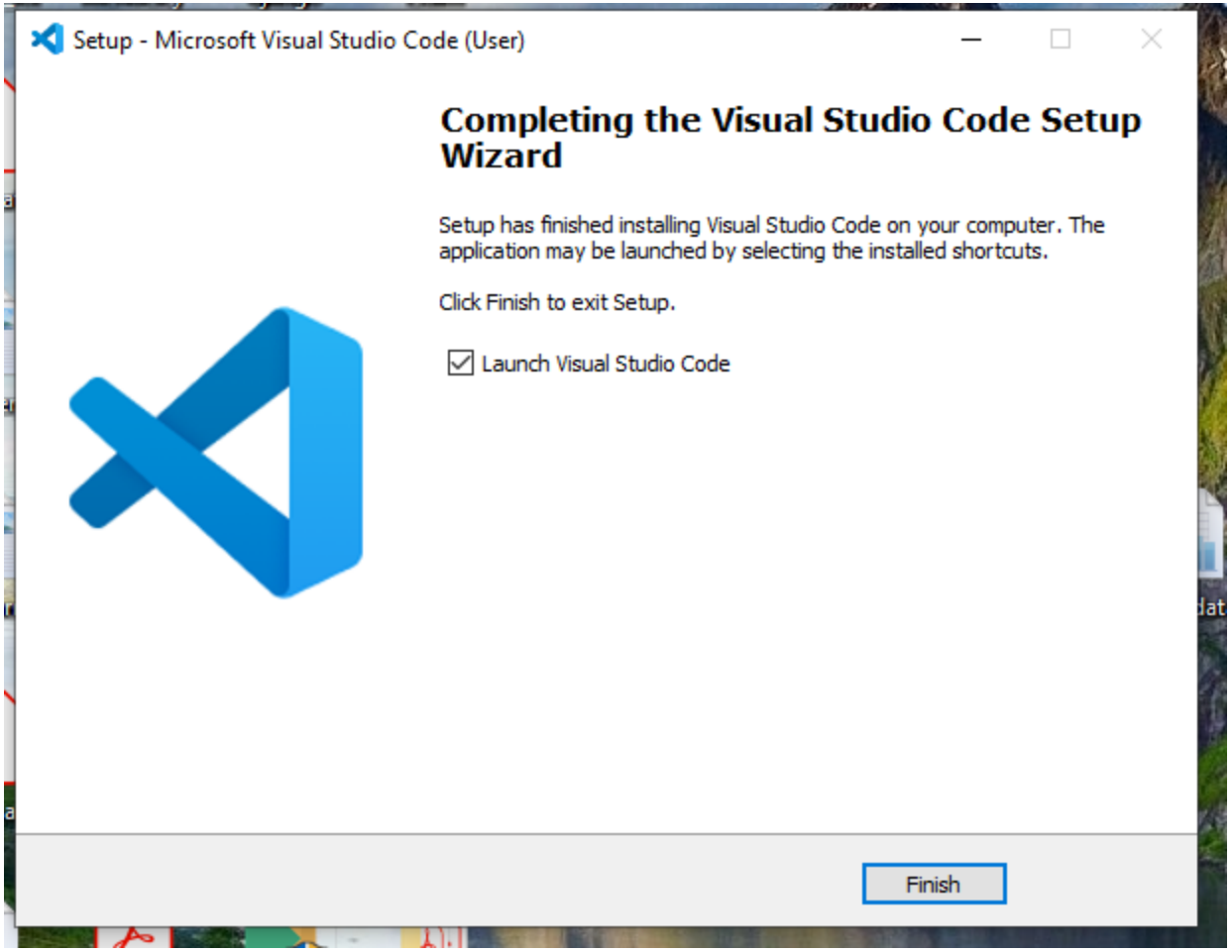
- pip install django



Install VS Code editor



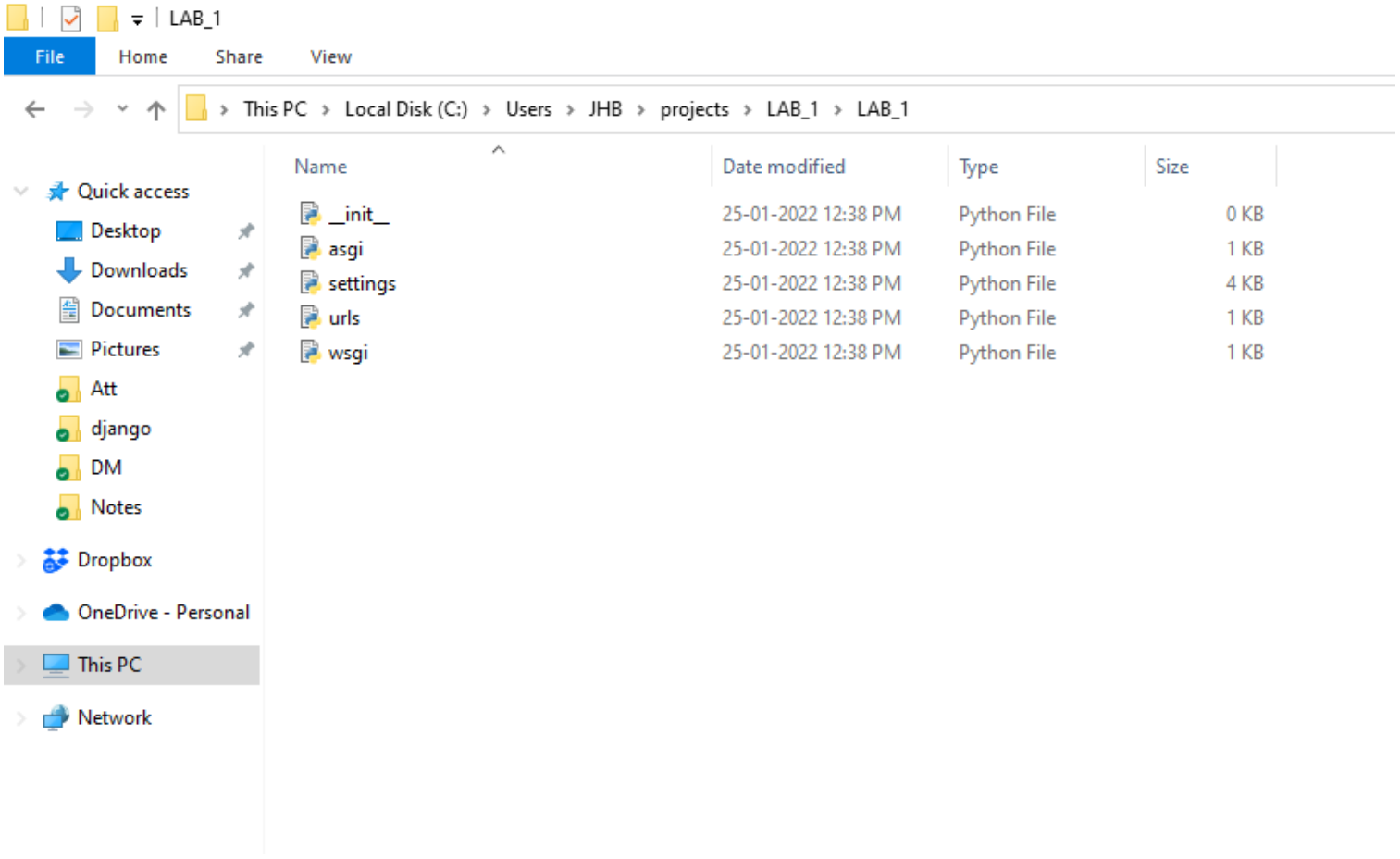
VS Code


Setup - Microsoft Visual Studio Code (User)
Completing the Visual Studio Code Setup Wizard
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Visual Studio Code
Finish

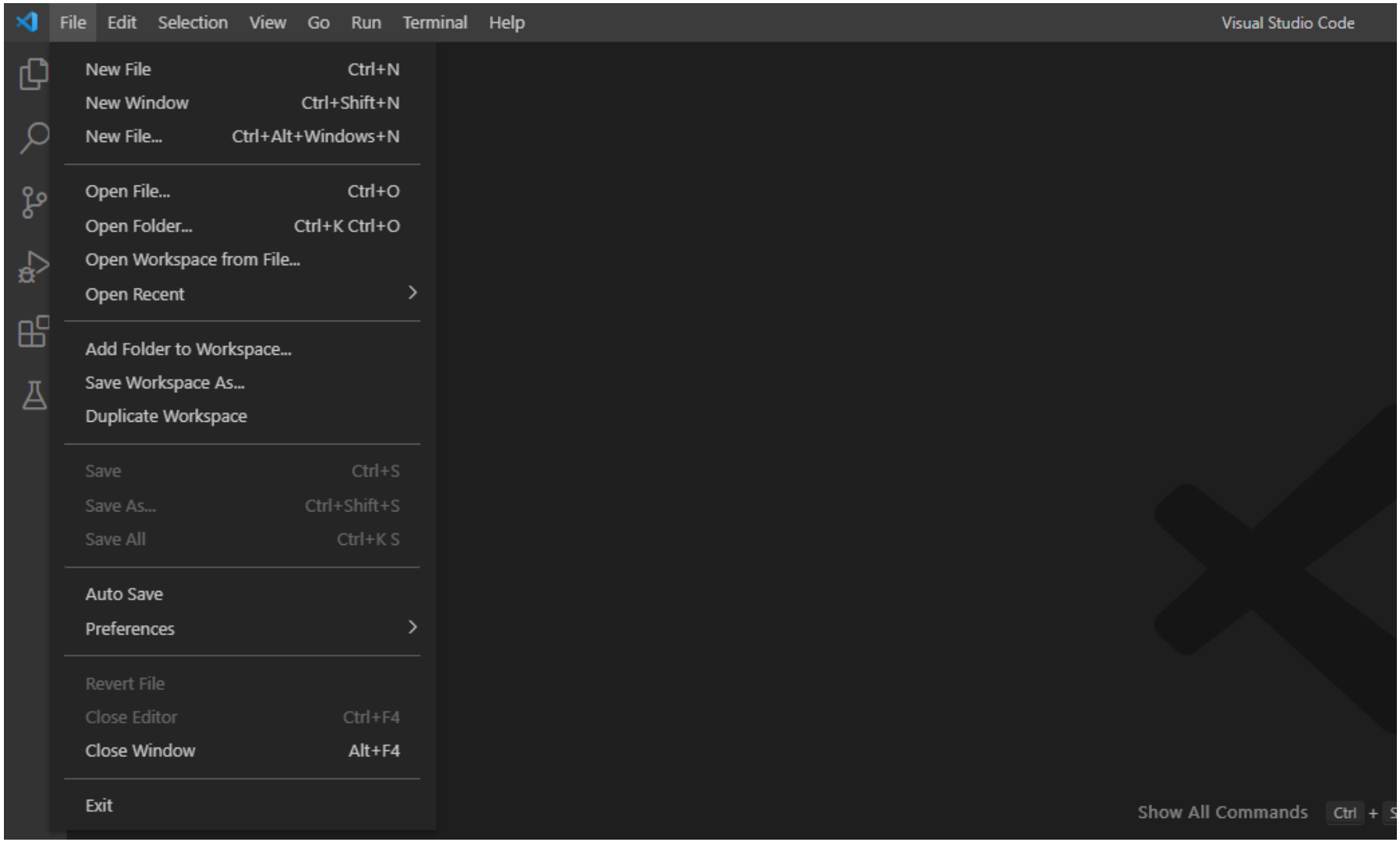
Setting up project

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

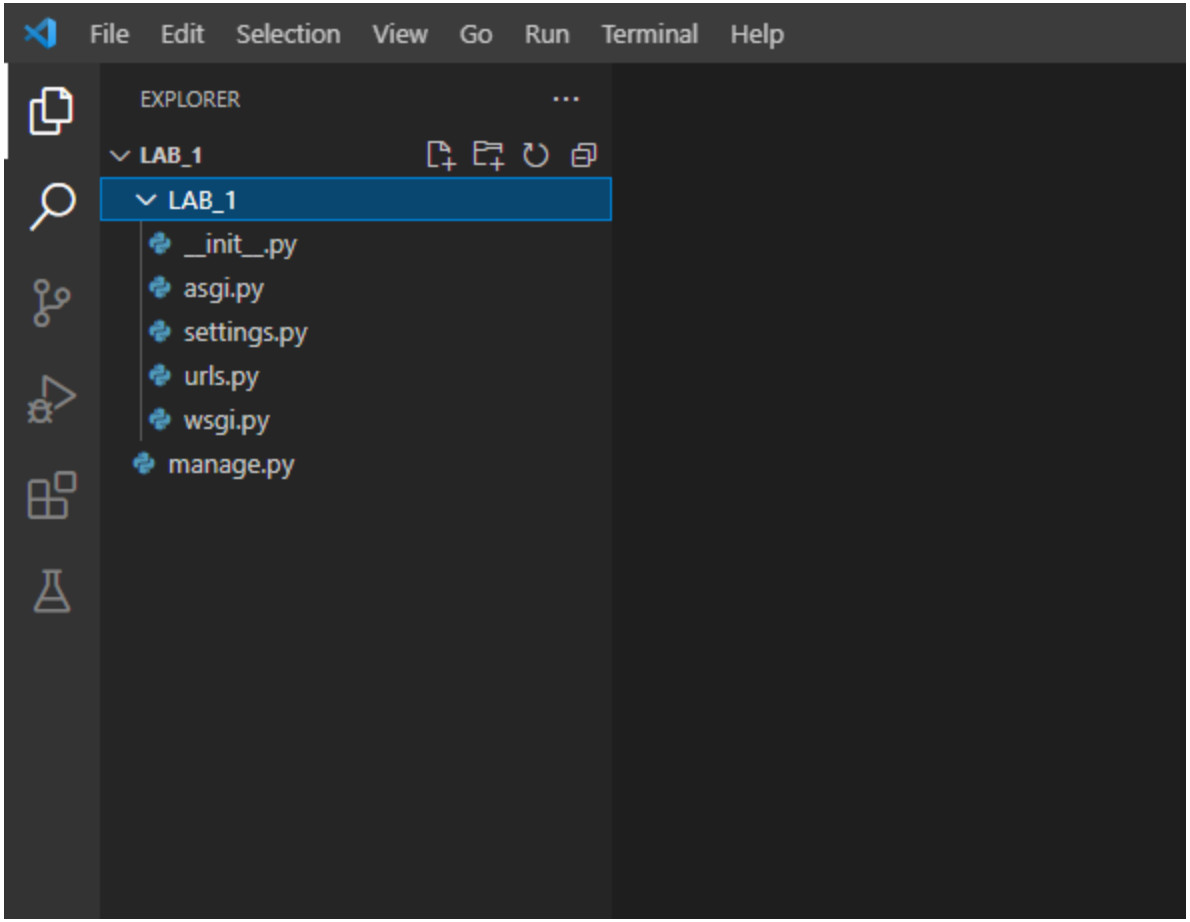
First project



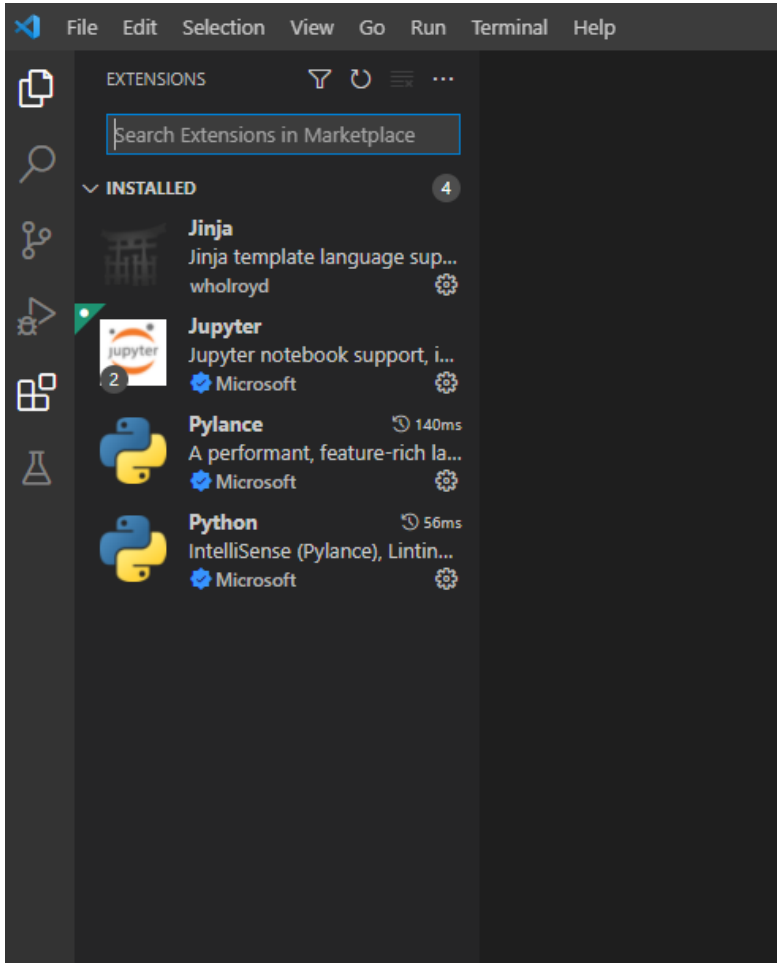
VS Code


File
Edit
Selection
View
Go
Run
Terminal
Help
Visual Studio Code
New File Ctrl+N
New Window Ctrl+Shift+N
New File... Ctrl+Alt+Windows+N
Open File... Ctrl+O
Open Folder... Ctrl+K Ctrl+O
Open Workspace from File...
Open Recent
Add Folder to Workspace...
Save Workspace As...
Duplicate Workspace
Save Ctrl+S
Save As... Ctrl+Shift+S
Save All Ctrl+K S
Auto Save
Preferences
Revert File
Close Editor Ctrl+F4
Close Window Alt+F4
Exit
Show All Commands Ctrl +

VS Code


File
Edit
Selection
View
Go
Run
Terminal
Help
EXPLORER
LAB_1
LAB_1
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
manage.py

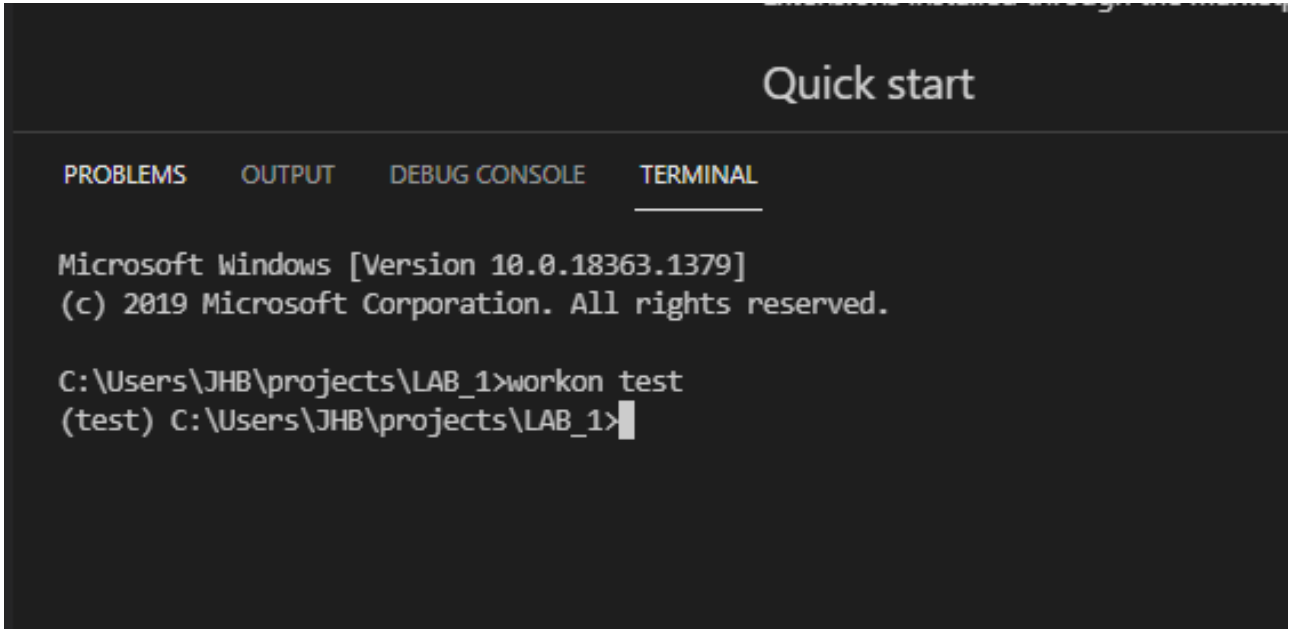
Python extension



Python extension

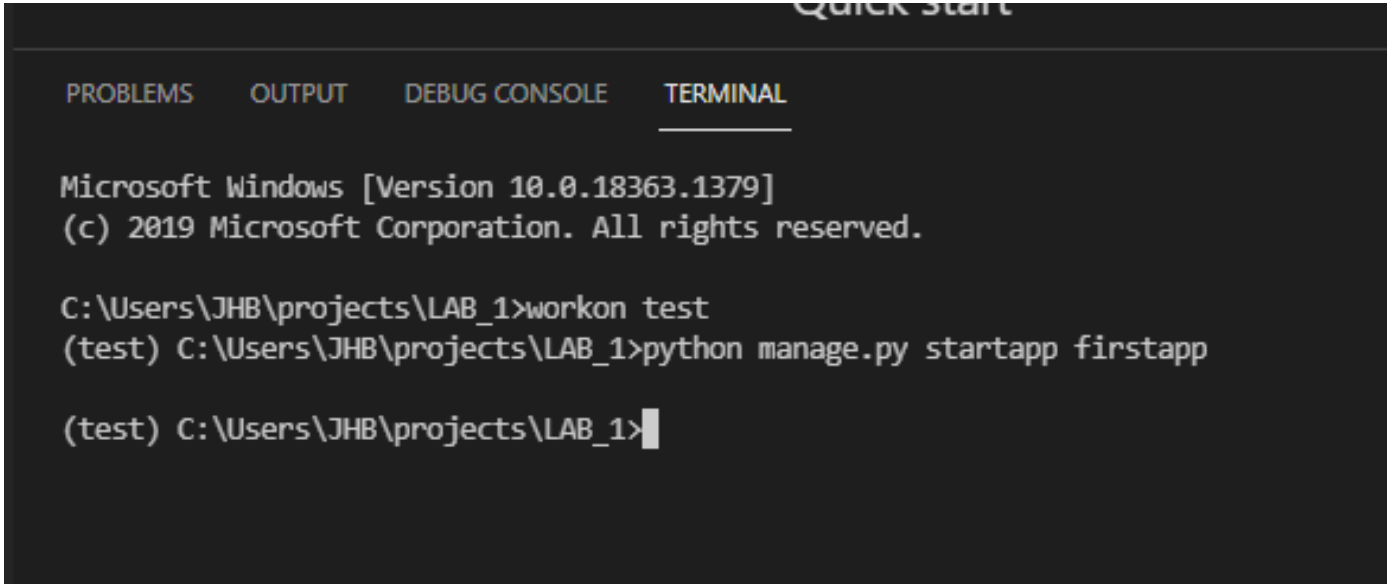
Working with virtual environment

workon test

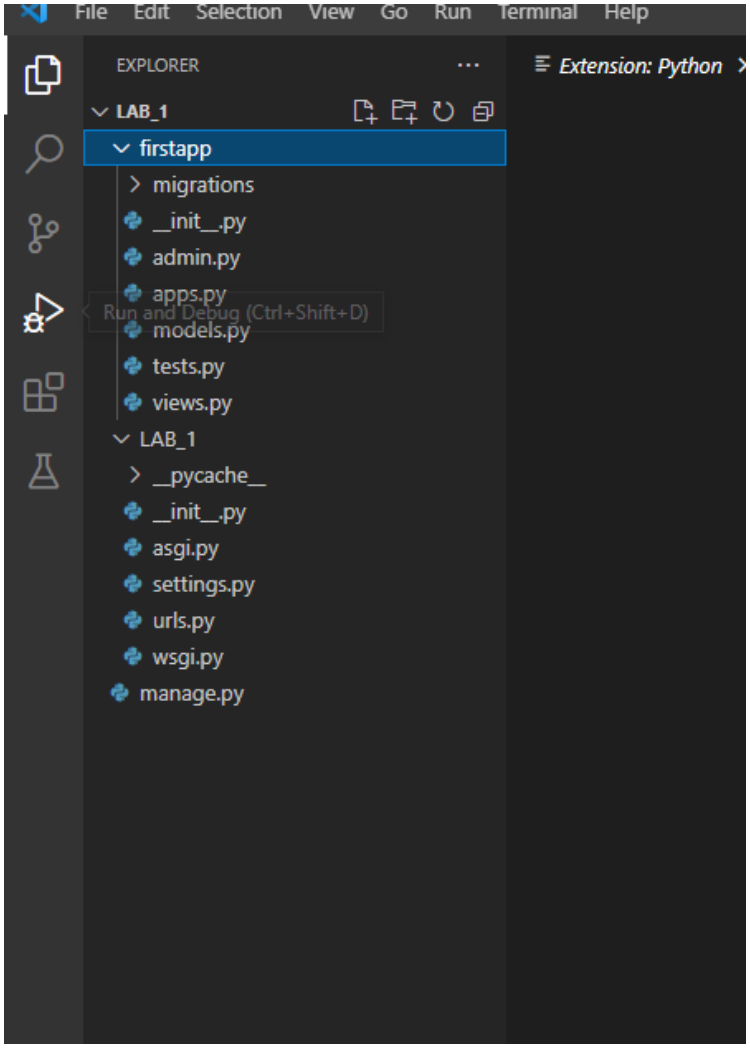


Create App

Python manage.py startapp firstapp

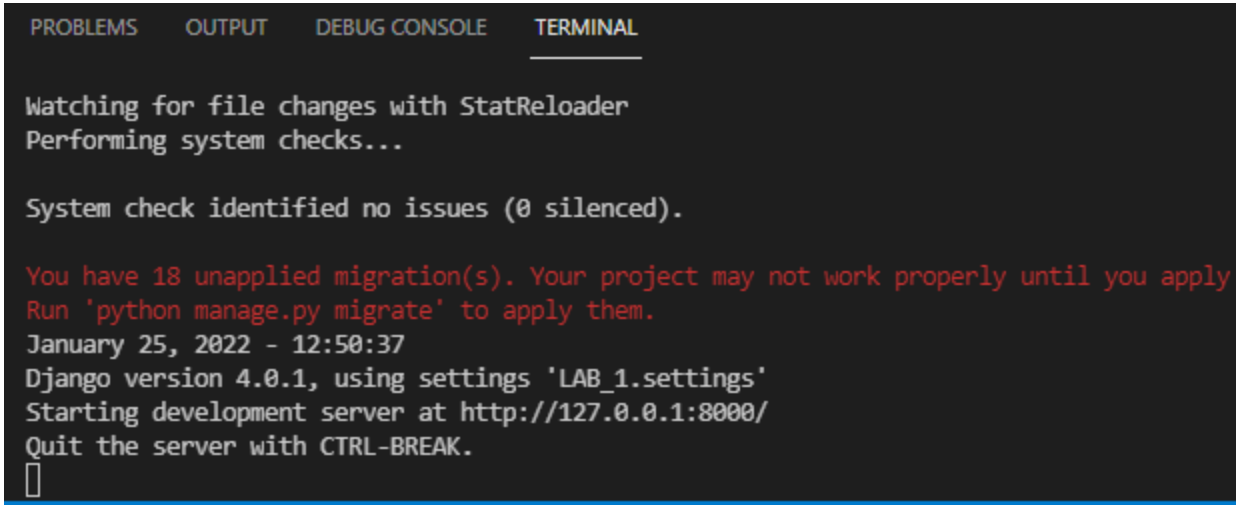


VS Code


File Edit Selection View Go Run Terminal Help
EXPLORER
Extension: Python
LAB_1
firstapp
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
LAB_1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
manage.py
Run and Debug (Ctrl+Shift+D)

Start Server

Python manage.py runserver



127.0.0.1:8000

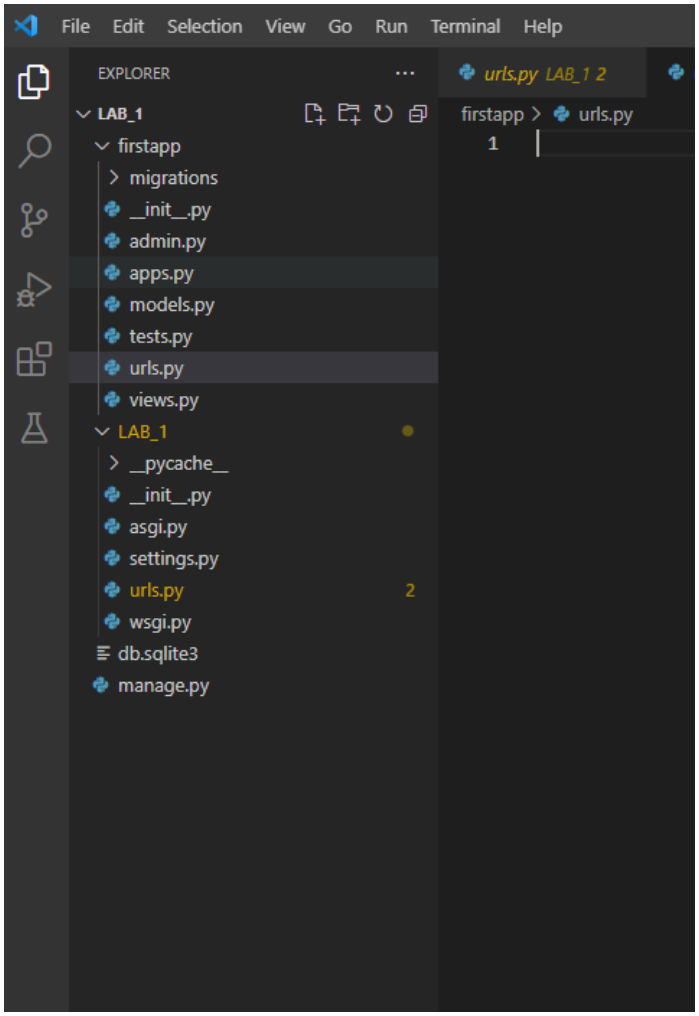
django

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

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

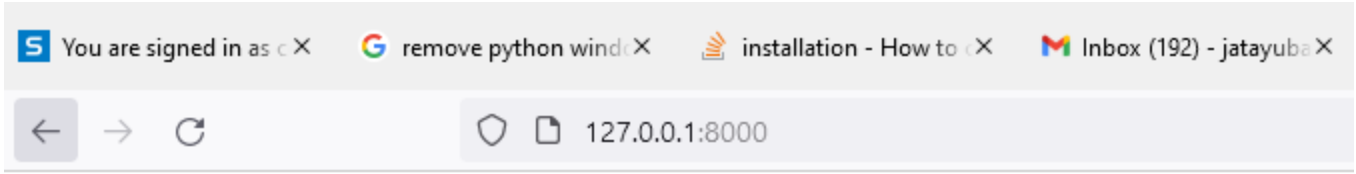
```
from django.http.response import HttpResponse
from django.shortcuts import render
from django.http import HttpResponse
# Create your views here.

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
```

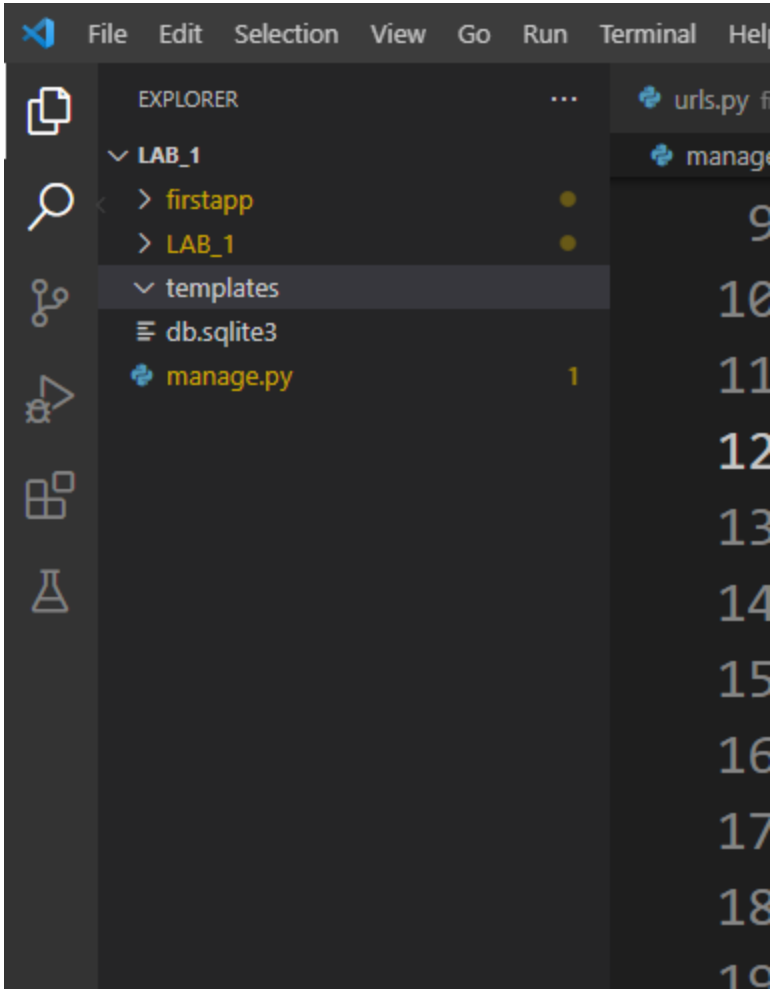
Output



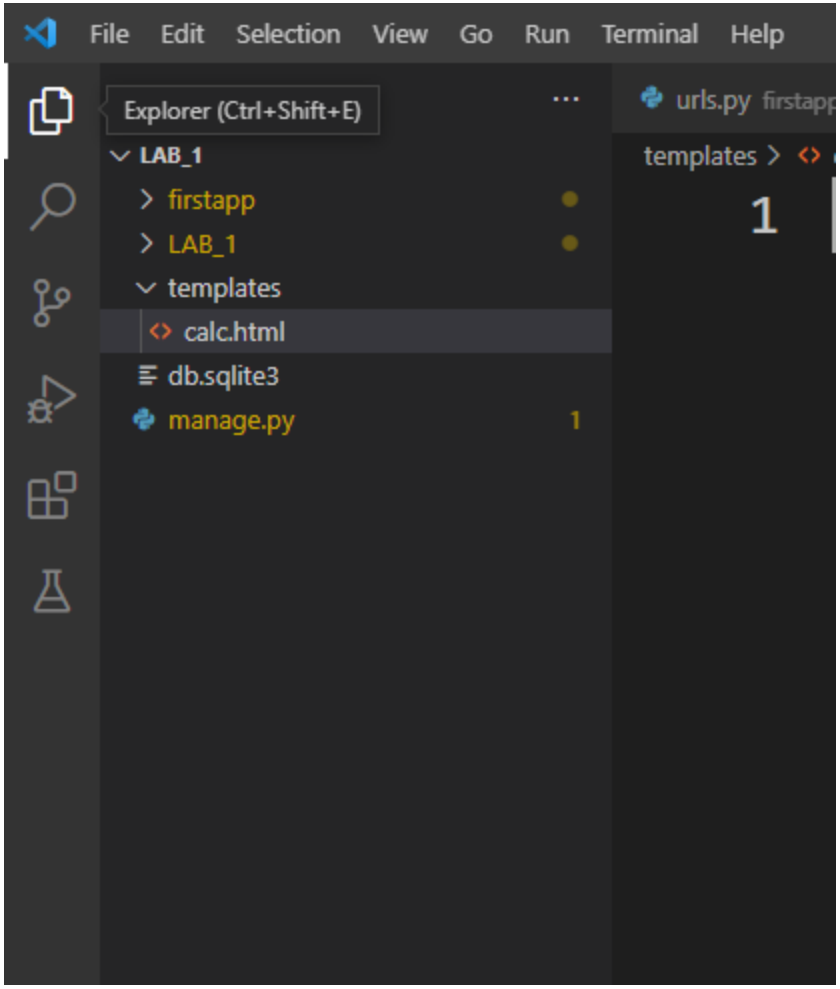
Hello World

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 | 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>
```