introduction

* free, open source framework to build web apps with python
* other framworks: flask, tornado, bottle, falcon, hug
* react, angular, vue generate webpage on client
* django, asp.net, express generate webpage on server
* API: application program interface

django features

* admin site
* object relational mapper (ORM)
* authentication
* caching data

Setting up

* install python
* run ‘pip3 install pipenv’ in cmd
* use code editor: vs code
* install ‘python’ extension in vs code
* now, install VScode

First app (secondtry)

* navigate to dir using cd command in cmd
* now, let’s create virtual environment called env
* python -m venv env
* env\Scripts\activate
* pip install django
* django-admin startproject secondtry
* in the above code, put ‘.’ at the end to stop creating additional folders
* cd secondtry
* now, open secondtry project folder in VScode (one exists with the env)
* python manage.py startapp hello
* here, hello is the name of the app (type in cmd)
* go to settings.py in secondtry in Vscode
* add ‘hello’ to end of list of INSTALLED APPS
* to start the app, go to cmd and type the below code
* python manage.py runserver
* open URL in browser, Ex: <http://127.0.0.1:8000/>
* ctrl + c to stop the server
* now, to create user you have to first create the database
* python manage.py migrate
* python manage.py createsuperuser
* now, add username, email and pw
* now, start server as follows
* python manage.py runserver
* go to URL/admin Ex: <http://127.0.0.1:8000/admin> and enter username and pw
* in Vscode go to hello > views.py
* Insert below code
* def index(request):

    return render(request, 'hello/index.html')

* create templates folder under hello and create subfolder hello under templates
* create index.html in hello subfolder
* include below code as html file content
* <h1>Hello world!</h1>
* go to urls.py in secondtry dir and add below code to import
* from hello.views import index
* and add below code in the urlpatterns code section
* path('hello/', index)
* if you leave path empty (Ex: path(‘’, index)), it’ll set default path as <http://127.0.0.1:8000/>
* the server restarts automatically everytime we make a change
* go to [127.0.0.1:8000/hello/](http://127.0.0.1:8000/hello/)
* django project is successful!

second app (storefront)

* install python, pip, pipenv, Vscode, python Vscode extension
* create folder and install django inside (virtual environment is automatically created, Ex: C:\Users\Dilshan\.virtualenvs\storefront-fqKi16xk)
* we can use python interpreter inside the environment
* pipenv shell
* let’s start our project
* django-admin startproject storefront .
* python manage.py runserver
* server is running now!

using terminal in Vscode

* view -> command palette -> search ‘python interpreter’ -> enter interpreter path
* code below code in cmd to find interpreter path (Ex: C:\Users\Dilshan\.virtualenvs\storefront-fqKi16xk)
* pipenv –venv
* Put it in Vscode with \Scripts\python at the end (Ex: C:\Users\Dilshan\.virtualenvs\storefront-fqKi16xk\Scripts\python)
* view -> terminal
* terminal automatically activates the environment
* now the terminal is ready to use!
* python manage.py runserver
* the server is running!

creating the app (storefront)

* in Vscode terminal:
* python manage.py startapp playground
* storefront -> settings.py -> put app name under INSTALLED\_APPS section (Ex: playground)

writing views

* takes a request and returns a response (request handler)
* playground -> views.py
* Enter below code in views.py
* from django.shortcuts import render

from django.http import HttpResponse

def say\_hello(request):

    return HttpResponse('Hello World!')

mapping URLs to views

* create newfile in playground dir called urls.py (we can use any name)
* write below code in that file:

from django.urls import path

from . import views

#this is an url configuration module

urlpatterns = [

    path('hello/', views.say\_hello)

]

* now go to storefront -> urls.py and modify the code as follows:

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('playground/', include('playground.urls'))

]

* here, any url that has playground/ is sent to playground app
* in browser, go to [127.0.0.1:8000/playground/hello/](http://127.0.0.1:8000/playground/hello/)
* this section is successful!

using templates

* in django, request handler is called view and view (conventionally used term) is called template
* create ‘templates’ dir inside playground dir
* create hello.html file inside templates folder
* <h1>Hello World!</h1>
* now, modify the views.py file in playground dir as follows:

from django.shortcuts import render

from django.http import HttpResponse

def say\_hello(request):

    return render(request, 'hello.html')

* go to [127.0.0.1:8000/playground/hello/](http://127.0.0.1:8000/playground/hello/)
* now, hello.html content is being rendered!

more on using templates

* modify views.py with a dictionary as follows:

from django.shortcuts import render

from django.http import HttpResponse

def say\_hello(request):

    return render(request, 'hello.html', {'name': 'Dilshan'})

* now, go to hello.html and modify it as follows:

<h1>Hello {{name}}</h1>

* Now, refresh the browser!
* It’ll display ‘Hello Dilshan’
* We can further modify hello.html file
* {%if name%}

<h1>Hello {{name}}</h1>

{%else%}

<h1>Hello World!</h1>

{%endif%}

* refresh the page!

debugging django apps

* you can use run and debug in Vscode or django debug toolbar (search google)
* to use django debug toolbar: in Vscode terminal,

pipenv install django-debug-toolbar

* add ‘debug\_toolbar’ to INSTALLED\_APPS in settings.py
* then, add below code to modify urls.py in storefront:

import debug\_toolbar

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('playground/', include('playground.urls')),

    path('\_debug\_/', include(debug\_toolbar.urls))

]

* to enable middleware, add below code in top of middleware section in settings.py:

'debug\_toolbar.middleware.DebugToolbarMiddleware'

* now, to configure internal IPs add below code in settings.py:

INTERNAL\_IPS = [

    '127.0.0.1'

]

* now, modify the hello.html to make a proper html file:

<html>

<body>

    {% if name %}

    <h1>Hello {{name}}</h1>

    {% else %}

    <h1>Hello World!</h1>

    {%endif%}

</body>

</html>

* refresh the browser page to get the django debug toolbar!

how to hide secret key in django

* create .env file in storefront dir
* copy secret key from settings.py to .env file
* remove spaces and ‘’ signs from secret key
* similarly, copy DEBUG value from settings and remove spaces
* it should look like this:

SECRET\_KEY=django-insecure-pdc8+z#d#h1mgshqjm2&q5d^724ws@ln1rw)scn2&=2u9a=ns7

DEBUG=True

* open Vscode terminal and run this:

pip install python-decouple

* remove secret key and DEBUG from settings and add below code parts:

from decouple import config

SECRET\_KEY = config(‘SECRET\_KEY’)

DEBUG = config(‘DEBUG’, cast = bool)

* when uploading to github check .gitignore and select python

building a data model

* models are used to store and retrieve data
* relationships -> one to one, one to many, many to many
* data models should have minimal coupling, high cohesion (focus)

installing app

* pipenv install storefront