Pre-requisites (covered)

- Shell, Git, Python, HTML, CSS

Introduction to Django

Brief

- Django
- Project
- App
- Views and URLs

Why Django?

Companies that use

- Instagram
- Mozilla
- Spotify

Advantages

- versatile
- secure
- lot of support
- lot of stuff provided out of box => authentication, authorization
- rapid development
- complete framework

Other framework => MVC architecture, model view controller

Django => MVT architecture

- Model
- View
- Template

FAQ

- If you do not know all of the python, it's okay
- Dev in django => side by side we will be learning a lot more in Python

You will be having regular assignments from now.

- takes roughly 1 hour to complete for today's assignment

Next few sessions

- views

- templates
- models and databases
- forms and authentication

End goal: Build a Trello like application

Creating a new project

- starting code will be provided by django
- `django-admin startproject first_project`

File structure

manage.py

allowing us to interact with django

- start the server
- make/run the migrations (clear later on)

first_project/settings.py

- It contains configurations and variables

first_project/wsgi.py

- Allows us to deploy our application, [web sever gateway interface]

first_project/urls.py

- Allows us to map our URLs to the views, url mapper

first_project/init.py

- first_project is a package

Run server

`python3 manage.py runsever`

http - hyper text transfer protocol browser will make a request server responds with the response

Applications

Project has multiple applications

- allows for reuse of code
- plug and play

Create our first app

```
python3 manage.py startapp first_app
```

Files that got created

1. admin.py => register our model classes for the app
2. apps.py => app specific configuration
3. models.py => data models related info, python classes that map to tables in db
4. tests.py => write some unit tests
5. views.py => create functions that return responses to requests made onto the server

Folder

1. migrations => stores DB specific info

Respond with Hello World

first_project/urls.py

- path('', views.index) For localhost:8000/ this path go to the index method in views to fetch the response
- path('about', views.about_me) localhost:8000/about will go to about_me method in views to fetch the response
- urls.py simply redirects a request to the correct view
- view takes the request, generates and returns the response

Name the path

- path('', views.index, name='index')

Allow us to import this path and redirect at some places in the code

=> Static Routing covered till now

Dynamic Routing

- age in the url
- print the age on the page

```
path('<age>', views.age_page, name='age'),
```

```
def age_page(request, age):
  return HttpResponse(
    """
    <h1 style='margin:5rem'>
    """
    +
    age
    +
    """
    </h1>
    """
  )
```

include statement, segregating the urls of app

- path('first_app/', include('first_app.urls'))
- will match first_app in the url like localhost:8000/first_app/{keyword}
- and then send the remaining keyword to first_app.urls
- all the paths checked in the sequence

MVT architecture

- Model
- View ==> looked this in the today's session
- Template

We will see in further sessions how views interact with models and templates.

Assignment

- Demonstration
- Contributing towards PSP
- Try to finish before the next session
- Takes roughly 1-2 hours max

Code

first_project/urls.py

```
from django.contrib import admin
from django.urls import path
from first_app import views
from django.urls import include

urlpatterns = [
    # 127.0.0.1:8000/{keyword}, keyword
    # path('', views.index, name='index'),
    # path('about', views.about_me, name='about_me'),
    # path('<age>', views.age_page),
    path('', include('first_app.urls')),
    path('admin/', admin.site.urls),
]
```

first_app/urls.py

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

```
# url patterns specific to the application

urlpatterns = [
  path('', views.index, name='index'),
  path('about', views.about_me, name='about_me'),
  path('<age>', views.age_page),
  path('11', views.about_me),
]
```

first_app/views.py

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

# Create your views here.

def index(request):
  # request comes here
  # job of view is to responds to the request
  return HttpResponse("<h1> Hello world </h1>")


def age_page(request, age):
  return HttpResponse(
    """
    <h1 style='margin:5rem'>
    """
    +
    age
    +
    """
    </h1>
    """
  )


def about_me(request):
  return HttpResponse(
    """
    <h1> About Me </h1>
    <p>
      About me text.
    </p>
    """
  )
```