PyCharm-Django-Pytest-Learning

1. SetUp

- Open Terminal
- Create new folder <*myDjangoProjects*>
- cd to **<***myDjangoProjects*>
- Open Pycharm and select ***New Project***
    - Set location to */home/barrie/myDjangoProjects/***<*new project name*>** Avoid using the word ***project*** in Django Project name
    - Choose New Environment Using ***VirtualEnv***
    - Set location to */home/barrie/myDjangoProjects/***<*new project name*>**/venv/<***new project name***>
    - Set Base Interpreter to /usr/bin/python3.9
    - Deselect all other boxes
    - Click on Create
- In Terminal,
    - Should see command prompt prefixed by <***new project name***>
    - Run ***pip install django***
    - Run ***django-admin startproject <new_django_project_name>***

      <new_django_project_name> = _____
    - To avoid confusion, rename folder <***new_django_project_name>*** to ***src*** using ***mv <new_django_project_name> src***
    - cd ***src***

- Folder **src** should contain ***manage.py*** and ***<new_django_project_name>*** folders

- Run ***python manage.py migrate***

- Run ***python manage.py createsuperuser***

- Run ***python manage.py startapp [my_new_App>***

  For convenience, use the suffix ***App*** on the new app name

<my_new_App> = _____

- Run ***python manage.py runserver*** and check that localhost:8000 shows Django default home page.

2. PyCharm

- From the ***src*** folder run ***pycharm .*** and PyCharm should open

- Check that the python interpreter is set correctly

- Add a configuration to run the ***runserver*** command

  - Select Run , Edit Configurations and Create Configuration

  - Name = ***runserver***
  - Script Path = path to ***manage.py***
  - Parameters=***runserver***
  - Select ***Create***

- Add the extensions: Material Theme UI, Djaneiro and PyCrunch

- Open the Terminal

  - Run

    pip install pytest, pytest-django, pytest-cov pytest-pythonpath

3. Start Coding Basic App

In ***settings.py***

1add ***[my_new_App]*** to INSTALLED_APPS section

2add the following in the TEMPLATES section within the DIRS list

```
path.join(BASE_DIR,'templates'


STATICFILES_DIRS = [
    os.path.join(BASE_DIR,'static')
]


MEDIA_URL = '/images/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'static/images')
```

In ***[my_new_App]*** folder

1***[new_modol_name]*** =

In ***models.py*** create ***[new_modol_name]*** model class

```
from django.db import models
class [new model name](models.Model):
    name = models.CharField(max_length=50)
    def __str__(self):
     return self.name
```

In ***views.py*** create views to list and add ***[new_model_name]***

```
from django.shortcuts import render
from .models import Snippet
def [new_modol_name]_home_view(request):
    context = {}
    return render(request, 'snippet/home.html', context)

def [new_modol_name]_list_view(request):
    # TODO: query database for all items
    context = {}
    return render(request, 'snippet/list.html', context)

def [new_modol_name]_add_view(request):
    context = {}
    return render(request, 'snippet/add.html',context)
```

In ***admin.py*** register ***[new_modol_name]*** model

```
from django.contrib import admin
from .models import [new_modol_name]
admin.site.register([new_modol_name])
```

In ***urls.py*** create paths to list and add views

```
from django.contrib import admin
from django.urls import path
```

```
from snippetsApp.views import snippet_add_view,snippet_list_view, snippet_home_view

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', [new_model_name]_home_view, name='home'),

    path('add/',[new_model_name]_add_view, name='add'),

    path('list/',[new_model_name]_list_view, name='list')

    ]
```

In ***src*** folder create ***templates*** folder

In ***templates*** folder create a new folder ***[new_model_name]*** with a name that reflects the model name

In ***[model_name]*** folder create the files base.html, home.html, ***add_[new_model_name]*** .html, ***list_[new_model_name]*** html

Add the following lines to ***base.html***

```
{% load static %}
<html lang="en">
 <head>
   <meta charset="UTF-8">
    <meta name="viewport"
         content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/
bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/
bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
   <link rel="stylesheet" href="{% static 'css/style.css' %}">
<title>Base</title>
</head>
<body>
 {% block content %}

 {% endblock %}
</body>
```

```
</html>
```

Add inital html code to home, add and list html files according to te following code

```
{% extends '[new_model_name]/base.html' %}
{% block content %}
<h1>[Page Title]</h1>
{% endblock %}
```

In ***[my_new_App]*** folder create ***static*** folder

In ***static*** folder create ***css*** folder

In ***css*** folder create the file ***style.css***

Delete the ***tests.py*** file in the ***[my_new_App]*** folder

In Terminal, run ***python manage.py makemigrations***

In Terminal, run ***python manage.py migrate***

Using localhost:8000/admin, add the first ***[new_modol_name]***

4. Setting up Pytest

Add ***pytest.ini*** file to ***src*** folder

Add the following 3 lines to the ***pytest.ini*** file

```
[pytest]

addopts = -v -p no:warnings --nomigrations --cov=.  --cov-report=html

DJANGO_SETTINGS_MODULE = ***</new_django_project_name>***.settings`
```

In the Terminal, from the ***src*** folder, run ***pytest*** to check that PyTest is working

In the ***src*** folder create the ***.coveragerc*** file and add the following lines

```
[run]
omit =
 manage.py,
 *wsgi.py,
 *asgi.py,
 *migrations/*,
 *__init__.py,
 *admin.py,
*apps.py,
 *tests/*,
*settings*,
 *urls.py,
 *venv*
```

In the ***[my_new_App]/tests/*** folder create the file ***test_example.py*** with the following content

```
import pytest
from django.test import SimpleTestCase
class TestExample(SimpleTestCase):
 def test_example_pass(self):
  result = "Test Example"
 expected_result = "Test Example"
 assert result == expected_result, "Should pass test"
def test_example_fail(self):
```

```
result = "Test Example"

expected_result = "XXXXXX"

assert result != expected_result, "Should pass test"
```

5. Test App

From the Terminal, run ***Pytest*** . Both tests should pass.

View the ***htmlcov/index.html*** in a browser and check the coverage statistics

Create Tests for urls, models, views, and forms

6. Selenium

pip install selenium

pip install django-selenium-login

In settings.py, add

SELENIUM_LOGIN_START = ‘/login’

In Templates, check that the tags for inputs and buttons have names OR id set so selenium can reference via get_element_by_name or get_element_by_id