Lab 02

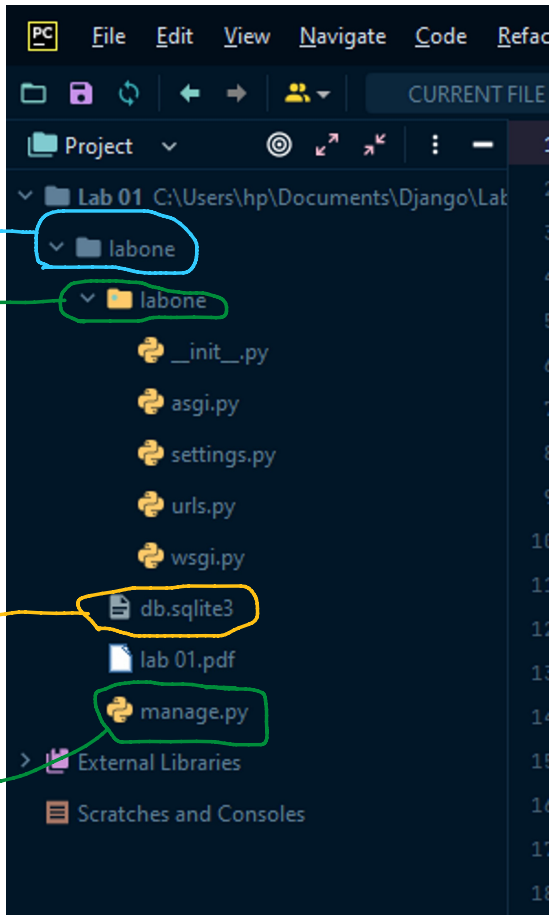
Saturday, October 1, 2022 4:32 PM

Python Django Structure

1 → Where we need to put HTML, CSS
Where database configuration goes
Where we need to put urls, and views

So let's understand the folder structure of project.

2 →



Main folder

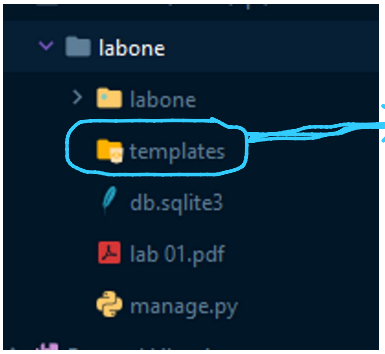
Another folder with same name as project name is, containing multiple files

A sqlite file is created when you run or migrate the project, which is default django database

This file is responsible for managing your whole django project also used to run the project or migrate the project.

3 → now if you want to use any HTML template then you have to create folder named as templates under main project folder.

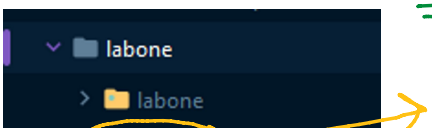
4 →



This folder is responsible of containing HTML files

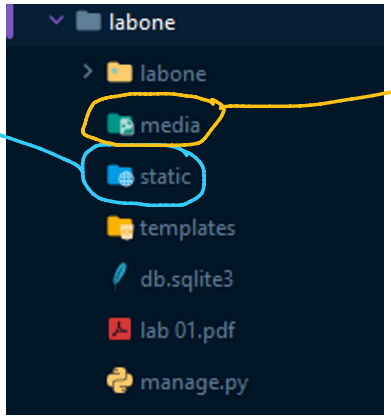
And you have to create this folder by yourself

5 → Also you have to create another folder with named as static, which contains the static files. Like css

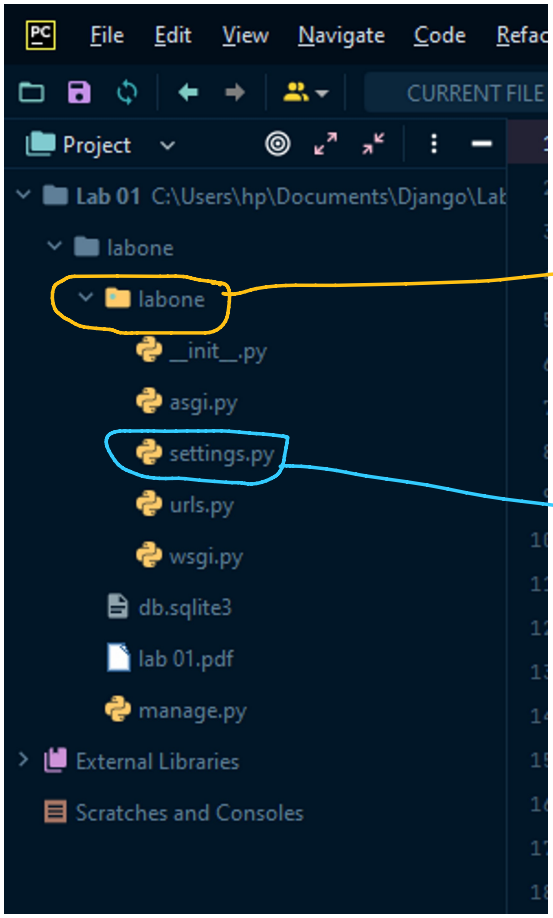


6 → You also have to create media folder which is responsible for

another folder with named as static, which contains the static files. Like css, javascript files, images, fonts etc.



You also have to create media folder which is responsible for containing dynamic files like images and other run time data



Another folder with same name as project name is, containing multiple files

This is main file which manages the database and other important stuff

What is inside in setting.py file? let's have a look

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/
```

Imported a path function

Through this path function base directory path is captured

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/


# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-d!npt3^jv$ly0ebvm_0pul9p*u9(!ju(f3!we71b(i+ryr7eo2'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
```

Imported a path function

Through this path function base directory path is captured

This is Django secrete key which is automatically generated when project is created

Debug true means when there is any error in project it will identified and if it is false then error is not identify

9

```
ALLOWED_HOSTS = []


# Application definition


INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
```

You can add here hosts name, if there is multiple then separate with commas

These are the by-default tables in the database, and its goes to installed apps

This is for the restriction of your database, means no-one can access your admin panel directly with authentication

10

```
ROOT_URLCONF = 'labone.urls'


TEMPLATES = [
    {
```

Indicates the URL file, you can see under setting file
It contains the URLS of HTML files

```
ROOT_URLCONF = 'labone.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
```

Indicates the URL file, you can see under setting file
It contains the URLS of HTML files

Here you have to add the template folder by using some function, inside the square brac

11

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

By-default database sqlite is attached here, if you can use mysql instead of default then you need to change the engine name and database name from here

12

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValida
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    }
```

In order to access the admin panel you need to add password and authentication

13

```
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True
```

Language and time zone setting

```
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True



# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field
```

Language and time zone setting

You have to give the static folder path here

14

So there is a lot of setting.py file use during the website development like

1. When you need to change or handle the database
2. If you want to add templates url
3. For static files
4. For media url

15

There is another file url.py

1. It manages all urls of your project
2. You can also create another file also to manage the urls
3. This is by default url file created by django
4. You can create all urls in this file

16

You need to create another file named as views

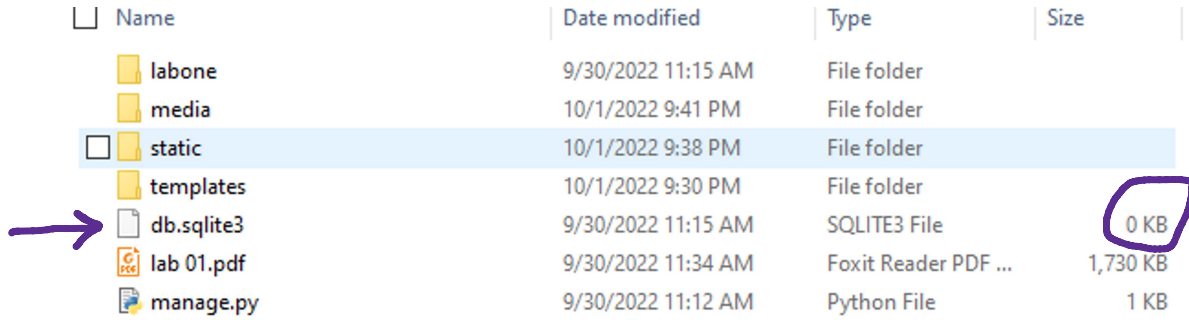
1. This file contain the functions
2. These functions is link with each url
3. Each url have separate function in views.py file
4. Views function is connected with url
5. This function have the link of specific HTML file

17

How to Migrate Default Migration

What is default migrations?

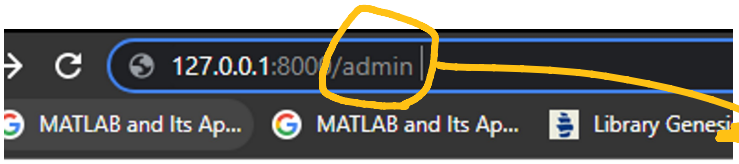
Whenever we are going to make any model, model means creating a table in data base we need to do migrations and migrate commands. Also django use sqlite3 database by defalut. You can see in the setting.py file. POINT no. 11 , You can also check in the folder there is file available named as db.sqlite3.

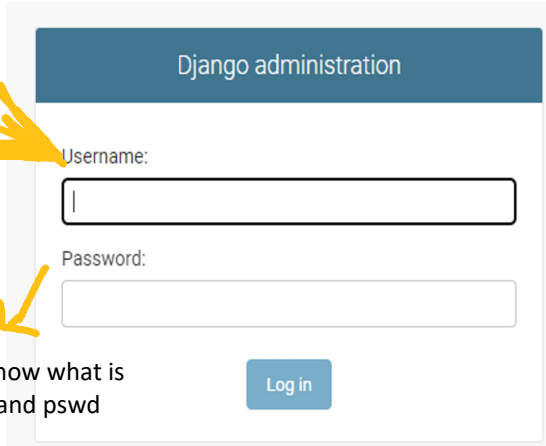


Size is 0 before migrations

So when we perform migrations a default table is created inside database, you can also see the size will be change.

When we create django project it also provide us admin panel, so lets open it





Super User is one who is allow to login and open this admin panel.

We don't know what is user name and pswd

As we didn't do migrations so that's why we are not able to create superuser.

When we do migrations then the by defalut table created and then we are able to create super user.

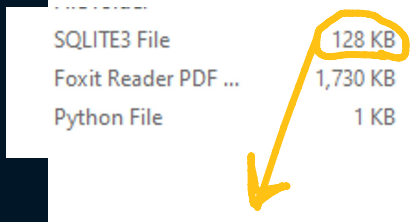
So in order to do migrations we need two commands

1. Python manage.py makemigrations
2. Python manage.py migrate

Manage.py is responsible to do migration

```
PS C:\Users\hp\Documents\Django\Lab 01> cd .\labone\
PS C:\Users\hp\Documents\Django\Lab 01\labone> python .\manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
```

So now tables created in database



You can also see the size of db.sqlite3 is changed after this

```
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

You can also see the size of db.sqlite3 is changed after this command is given

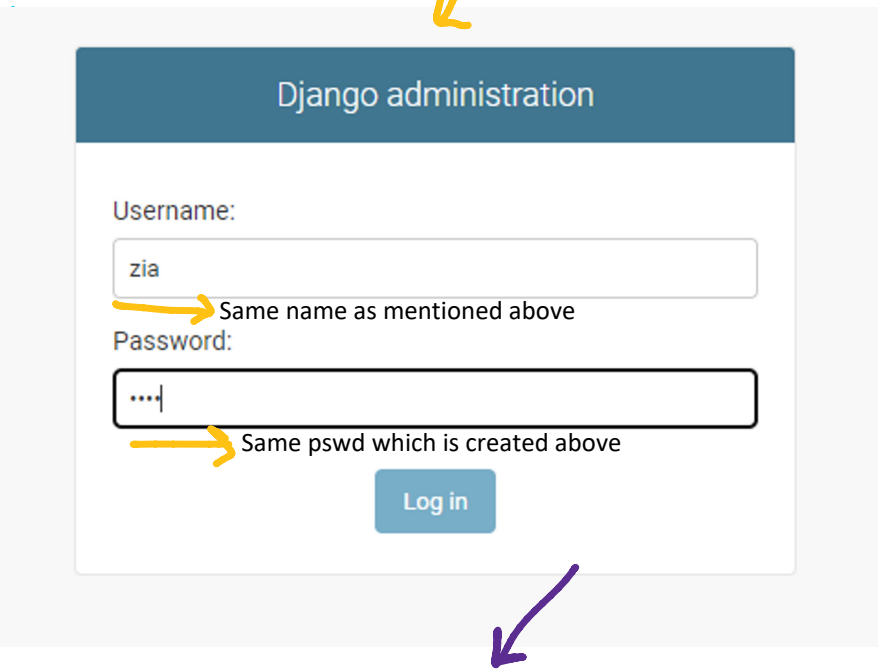
You are not able to see the tables of database, for that you need to install db browser.

How we create super user, so that we are able to open admin panel?

Command: python manage.py createsuperuser

```
PS C:\Users\hp\Documents\Django\Lab 01\labone> python .\manage.py createsuperuser
Username (leave blank to use 'hp'): zia
Email address: engrziaurrehman.kicsit@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

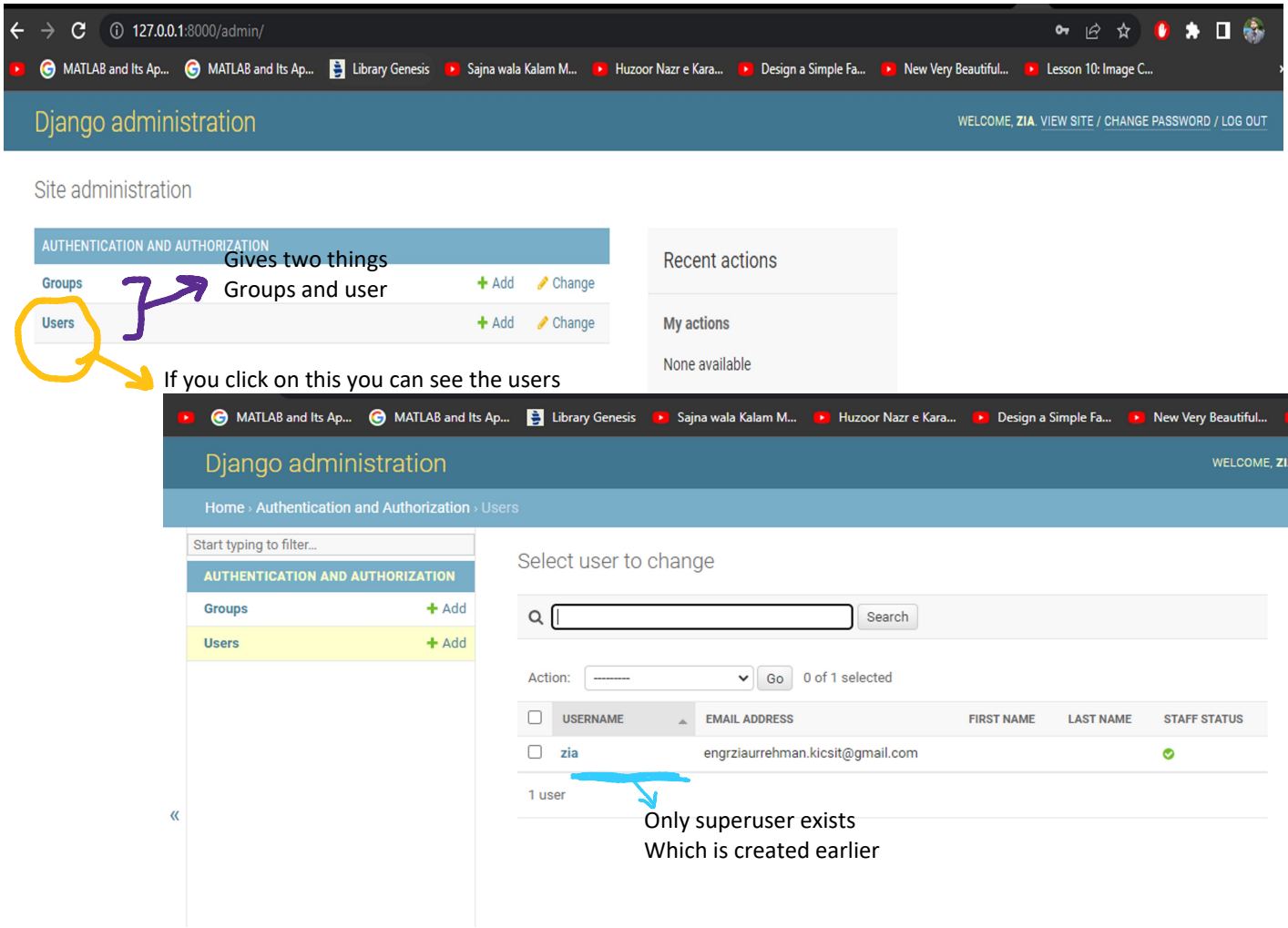
command



Same name as mentioned above

Same pswd which is created above

Admin panel



In future we have add user or groups to upload dynamic data in models so we use this user and group panel later on.

So our website is divided into two parts

1. Main website url which is display to users or public
2. Admin panel or administrative part only accessible for authorized persons

20

URLs and Views

URL's also known as Routs

For example: https://www.kicsit.com/ → shows the main page (Main routing)

This blog/ is also known as slug

https://www.kicsit.com/blog/ → shows the blog page

https://www.kicsit.com/blog/singlepage/ → shows the specific post of blog page

Detailed routing

Views : The logic is executed for different URL's (https Method)

For every specific URL there is specific View created. Means Url's connected with views

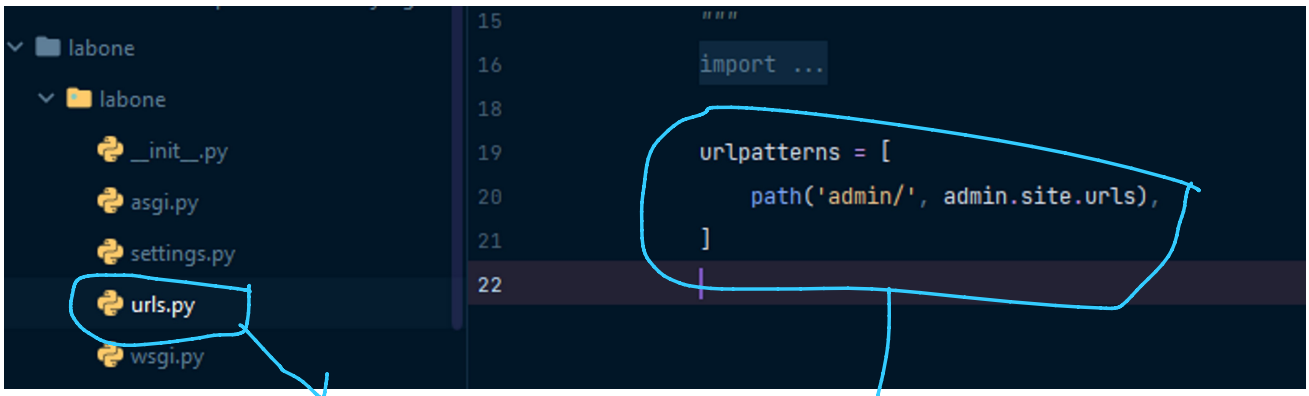
View works on https methods

Two types of Views
1. Functions
2. Class

So what did view do?
View prepares the data and send to HTML page

so the question is that how URL and Views are connected?

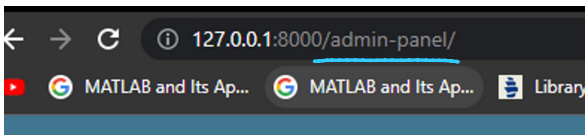
There is a file named as urls.py and another file also have to be created named as views.py.
And Views.py contains the all views or functions or classes and URL.py contain all urls



Open this file

This is the url of admin panel
Now change this and see what happens

```
urlpatterns = [
    path('admin-panel/', admin.site.urls),
]
```



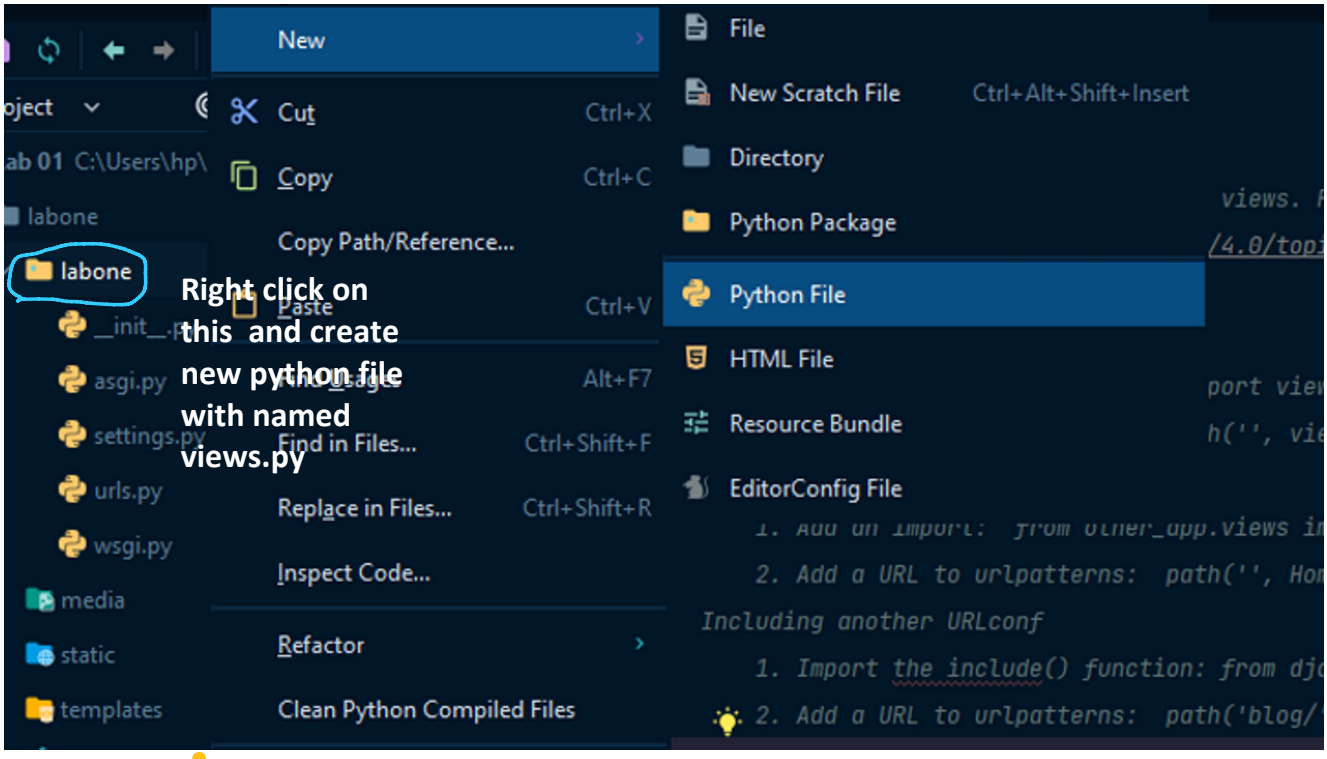
In this way you can change the url name

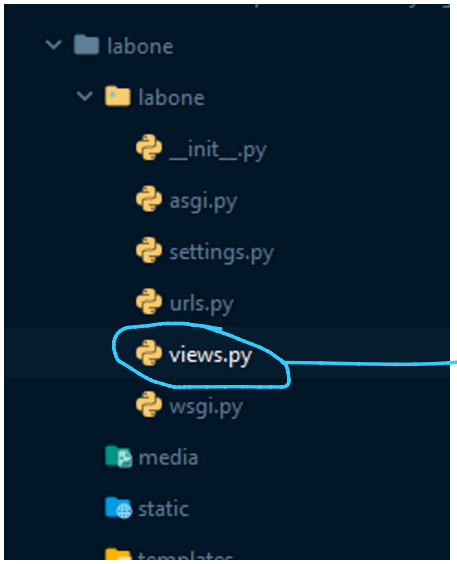
Now in order to access admin panel you have to write admin-panel

21

Creating Django URL's and Views

- So now we learn how to create a URL and view for that URL and run on browser.
- We are not going to create any HTML page at this time, only we learn how to create url and views

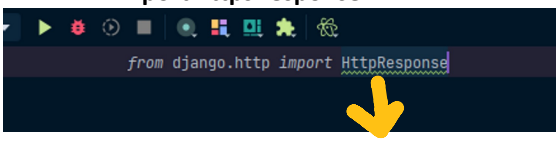




Views.py file is created

Now inside this file you have to write some code

1. **Import https response**



This would be responsible for displaying some text on browser or your response on the browser.

2. **Now you have to create function like aboutUs for new page aboutus**

```
from django.http import HttpResponse

def aboutUs(request):
    return HttpResponse("Welcome to Django Framework Development!")
```

I have created aboutUs function for aboutus page and when page opened the following string is written on there by Httpsresponse

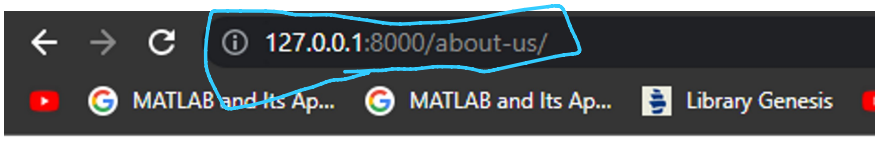
3. **The next step is you have to make the URL**

For that you have to open urls.py file

So first you have to import the views file inside the urls.py file

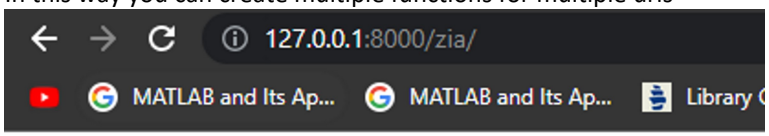
```python
from django.contrib import admin
from django.urls import path
from labone import views

urlpatterns = [
    path('admin-panel/', admin.site.urls),
    path('about-us/', views.aboutUs)
]
```



Welcome to Django Framework Development!

In this way you can create multiple functions for multiple urls



Engr. Zia Ur Rehman

```python
def aboutUs(request):
    return HttpResponse("Welcome to Django Framework Development!")
def zia(request):
    return HttpResponse("Engr. Zia Ur Rehman")
```

```python
from django.contrib import admin
from django.urls import path
from labone import views

urlpatterns = [
    path('admin-panel/', admin.site.urls),
    path('about-us/', views.aboutUs),
    path('zia/', views.zia)
]
```


from django.contrib import admin
from django.urls import path
from labone import views
2
urlpatterns = [
path('admin-panel/', admin.site.urls),
path('about-us/', views.aboutUs),
path('zia/', views.zia)
]