Django Framework

💡 **Tools and Technologies Used :**

1. Python
2. Django
3. OpenSSL

Issue

> Understanding Django Framework :

- Establishing a User Login Authentication Page.
- Implementing HTTPS for hosting.

💭 Platforms

> The Environments used to test the Project:

- Windows 10
- Kali Linux

🛫 Implementation

> **Then, answer these questions:**
>
> ▼ Installing Python:
>
> ```
> a. Download and install python3.11 from
>     https://www.python.org/
>
> b. Add Python to the path in Environment Variables
>
> c. Use the following code to create virtual environment
>    python3.11 -m venv myenv
>
> d. Activate the scripts ; go to the scripts folder
>          ./activate
>
> e.Install the requirements.txt
>     pip install -r requirements.txt
> ```
>
> - The virtual environment is already included into the git file, hence we can ignore this.

▼ Configuring Django:

```
a.Create a django Project
      django-admin startproject *AnyName

b. Use this to open and migrate the server
    python manage.py migrate

    Then run the server
    python manage.py runserver

c. Start the django app
     django-admin startapp testing

d. Now we need to register this file in settings.py

        '*appname',

e. Create and connect url.py and view.py
   We are trying to connect the  app
   to django project.

    Open urls.py in the project and  add

        from django.urls import path,include
        path('',include('testing.urls'))

        Now we will be able to create urls in the app itself

f. Update views to add the required pages.

g.Create a subfolder with the app Name and create a Templates folder here.
   Inside the folder , create
              'index.html'
              'register.html'
              'my-login.html'
              'dashboard.html'

h. Implement access restrictions
   After setting up the app,make sure to restrict access to the dashboard to only logged-in users

i. Use Django  forms to perform the register and login authentication functions.

j. I have created a Super User using

            python manage.py createsuperuser

   Now we run the migrate command , before starting the server

           python manage.py makemigrations
            python manage.py migrate
```

▼ OpenSSl

```
a.Install OpenSSl to create,sign certificate and keys.

b.We are hosting a development server from our localhost environment.

c. So, if we want to host using https , we can implement it using a self-signed certificate or
   create a local CA to sign the certificate.

d.In this project, I created a local Certificate Authority (CA) and used
  it to sign certificates for mylocal domain.

    1. To create a Local CA :

        openssl genpkey -algorithm RSA -out rootCA.key
        openssl req -x509 -new -key rootCA.key -out rootCA.crt

    2. Generate a Certificate Signing Request (CSR):
       For each domain we want a certificate , we need to create a CSR

        openssl req -new -key localhost.key -out localhost.csr -subj "/CN=localhost"

    3. Sign the CSR with the Local CA:
       Now use the local CA we created to sign the CSR and generate a certificate

       openssl x509 -req -in localhost.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out localhost.c

e.But while implementing the Django app ,  the browser still shows warning as the local CA is unrecognized.
  We can import the rootCA.crt to the browser's managed certificates place it in Trusted Root Certification
  The browser still gives warning  but exempts.
  So now , we understood how to implement using https.
```

Functionality:

```
1. Users can register in the webpage , and their details would be stored

2. Username , Email ID  would be checked in the server , if there are any matching instances if so ,
   they would be asked to change the Username or the Email accordingly.

3. As do not store Raw Passwords, Django encrypts them using SHA256 , then asks to enter the password again ,
   to encrypt and verify the both hash outputs.

4. In the login page , if the Users have registered they can log-in using their credentials.
   So , after they login they will be redirected to the Dashboard. They can also  log-out from the dashboard,
   after they will be redirected to the home page.

5. The Super User can have change/access their admin privileges in the admin page
   which is  the website path +'/admin'.
```

Working:

- **Windows**

```
1. Navigate to the `myenv/scripts` directory using the `cd` command, then execute:

                             ./activate

2. Next, proceed to the Test Folder and run the following Command:

      python manage.py runsslserver --cert localhost.crt --key localhost.key
```

- Bash

```
1. Open the Testing Folder, navigate to the Test Folder and run the following Command:

      python manage.py runsslserver --cert localhost.crt --key localhost.key
```