ReadMe

**Assignment**: Authentication process By using Fast api and MongoDB  
  
**Brief:-**

In this assignement at first we connect MongoDB database with Python and create authentication precedure by using FastAPI on server.  
  
**Imported Module :-**  
1) mongoclient() function of Pymongo module used for connecting Python and MongoDB.  
2) FastAPI() function used from connecting with Fastapi and Python.Depends function and HTTPerror case are also utilized from this module.  
3) Basemodel of Pydantic module used for designing model.  
4) CryptContext() function from passlib.context module used for converting plain password into hashing password.  
5) OAuth2PasswordBearer of fastapi.security module used for creating token after successful authentication. OAuth2PasswordRequestForm of fastapi.security module used for displaying GUI for authenticaton .   
6) pymongo.errors module were use for importing duplication error exception.  
  
**Description:-**  
In this project,there is one database whose name is Userinfo and collection name is Proo.

App is Fastapi instance is used to create endpoints.  
Get and post are Http request used for creating endpoints in fastapi.

bcrypt function are used for hashing.

oauth\_scheme is OAuth2PasswordBearer instance which is used to create tokens after successful sign\_in.

Defined functions are:-

Hashing fun():

It return hash formed password.

Sign\_up fun():

In this function, we take input from user by using fastapi on local server(127.0.0.1:8000) and stores in database base.In this we use indexing in mongodb to store only identical username and also store password in database in hashed format.If user try to insert same username then it return error.

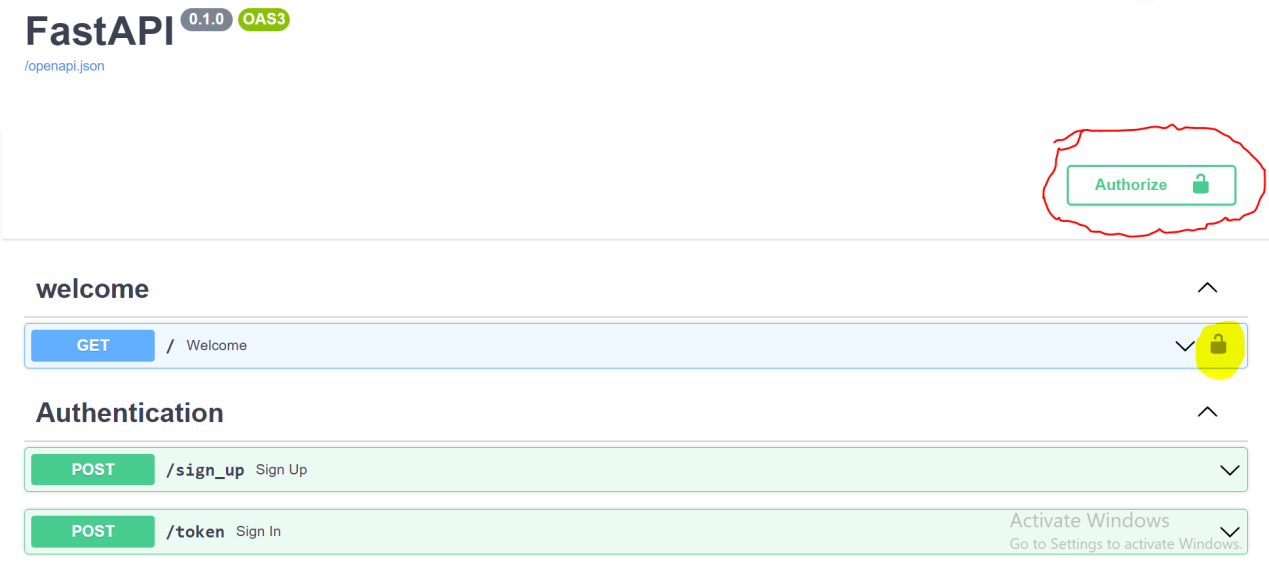
Authenticate fun(): it will verify by using verify function of cryptconcept and return boolen value to sign\_in function.

Sign\_in fun():

In this module, we take input by using Auth2PasswordRequestForm and transfer credentials to authenticate function and authenticate fun return boolean expression and according to token generate if credentials are correct.

**Working :**

At first,execute program by using Uvicorn server

1. Uvicorn main:app --reload
2. Open 127.0.0.1:8000/docs by using any browser.
3. 

Here padlock sign represent that, it’s private.

**welcome;**

we use authentication , we need to use registered username with password for get access.

**Sign\_up ;**

In this we insert unique username and password in hashed format in database

**Token;**

in this we have sign\_in function which create token when authentication successful.