**NAME: Janhavi Sharad Shedge**

**AIM: To implement JWT Authentication.**

**Description:**

**jwt:** JWT is a JSON based security token for API Authentication. In its compact form, JSON Web Tokens consist of three parts separated by dots (.), which are:

Header

Payload

Signature

Therefore, a JWT typically looks like the following.

xxxxx.yyyyy.zzzzz

JSON Web Token (JWT) is a popular user authentication standard, used to securely exchange information online. This token is made up of three components – a header that specifies the algorithm used to encrypt the contents of the token; a payload that contains “claims” (information the token securely transmits); and a signature that can be used to verify the authenticity of the information.

* **Installing all the required modules**
  + pip install flask\_jwt
  + pip install werkzeug==2.0.0
  + pip install jinja2==3.0.3
  + pip install flask==2.0.0

from flask import Flask

from flask\_jwt import JWT,jwt\_required,current\_identity

from werkzeug.security import safe\_str\_cmp

class User(object):

def \_\_init\_\_(self,id,username,password):

self.id=id

self.username=username

self.password=password

def \_\_str\_\_(self):

return"User(id='%s')"%self.id

users=[

User(1,'janhavi','shedge'),

User(2,'user2','iwonttell'),

]

username\_table={u.username:u for u in users}

userid\_table={u.id: u for u in users}

def authenticate(username,password):

user=username\_table.get(username,None)

if user and safe\_str\_cmp(user.password,password):

return user

def identity(payload):

user\_id=payload['identity']

return userid\_table.get(user\_id,None)

app=Flask(\_\_name\_\_)

app.debug=True

app.config['SECRET\_KEY']='super-secret'

jwt=JWT(app,authenticate,identity)

@app.route('/protected')

@jwt\_required()

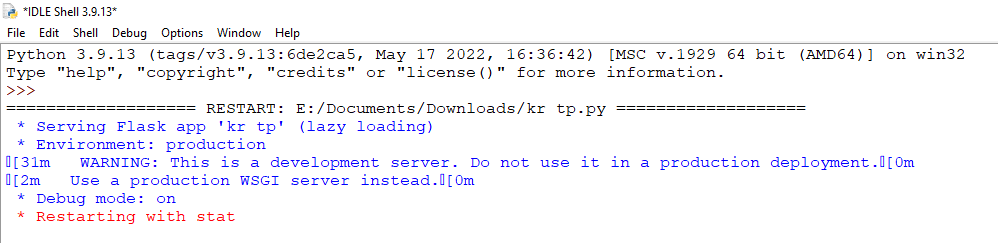
def protected():

return'%s'%current\_identity

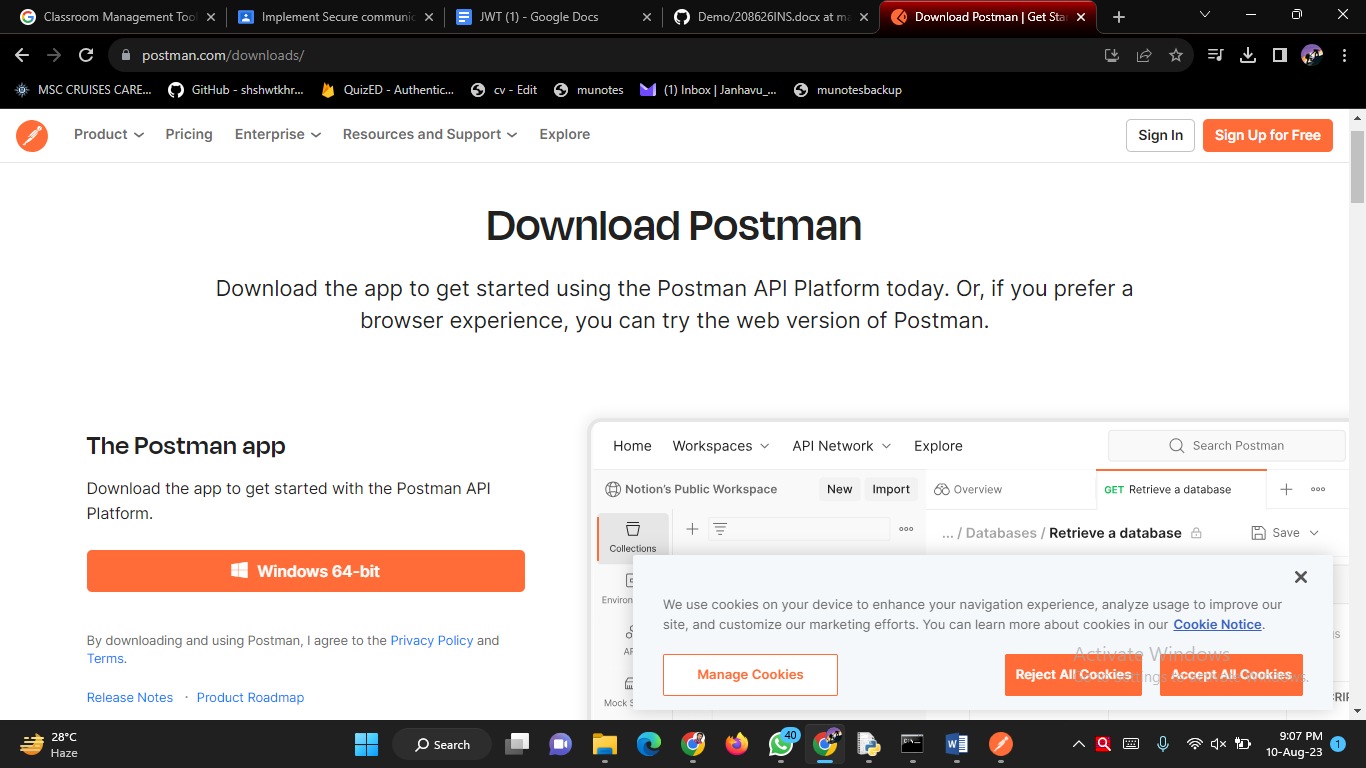
if \_\_name\_\_=='\_\_main\_\_':

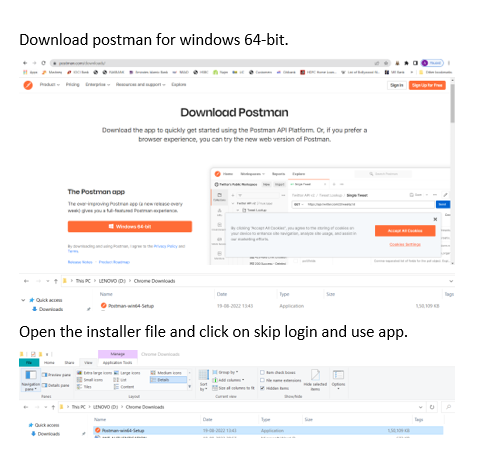
app.run()

**Output:**



Download Postman for Windows 64 bit





Click on **+** sign and paste the url: **http://127.0.0.1:5000/auth**

