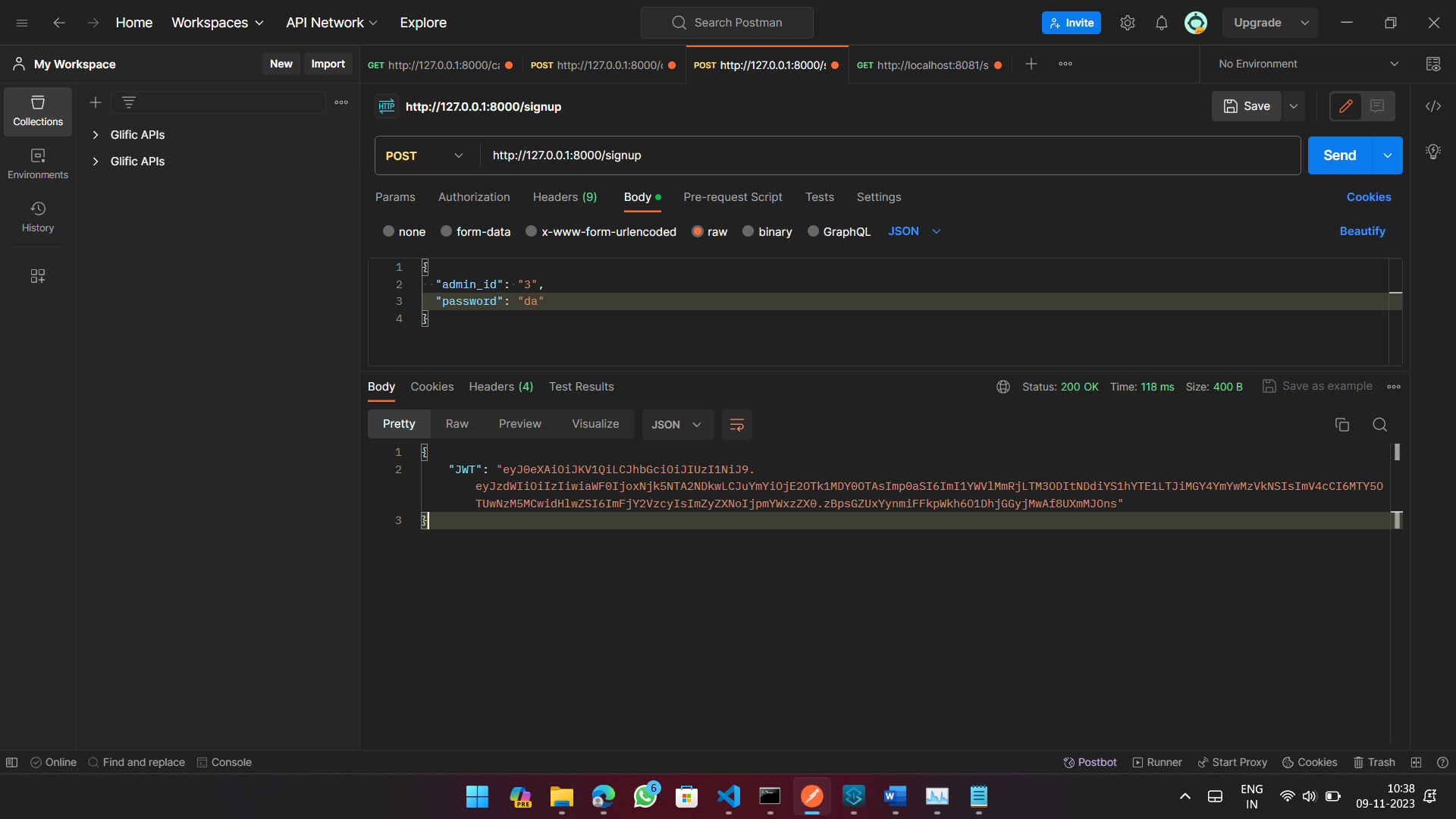
|  |  |
| --- | --- |
| Name: | Darshan Jain |
| UID: | 2022701005 |
| Experiment No: | 9 |
| Title: | Design secured Web application using web token |

Screenshots:



from fastapi import  APIRouter, Depends

from src.models.cars\_model import Cars

from src.database.cars\_db import get\_all\_cars, add\_cars, update\_cars, delete\_cars, cars\_model, login\_admin, create\_admin

from pydantic import BaseModel

from fastapi\_jwt\_auth import AuthJWT

router = APIRouter(

    prefix="",

    tags=["Application"],

    responses={404: {"description": "Not found"}},

)

class Admin(BaseModel):

    admin\_id: str

    password: str

@router.get("/cars")

def get\_all(Authorize: AuthJWT = Depends()):

    Authorize.jwt\_required()

    admin\_id = Authorize.get\_jwt\_subject()

    return get\_all\_cars(admin\_id)

@router.get("/cars/{id}")

def get\_cars(id, Authorize: AuthJWT = Depends()):

    Authorize.jwt\_required()

    admin\_id = Authorize.get\_jwt\_subject()

    return cars\_model(id, admin\_id)

@router.post("/cars")

def create\_cars(cars: Cars, Authorize: AuthJWT = Depends()):

    Authorize.jwt\_required()

    admin\_id = Authorize.get\_jwt\_subject()

    add\_cars(cars.id, cars.name, cars.price, cars.company, admin\_id)

    return {"SUCCESS": "cars added successfully"}

@router.put("/cars")

def cars\_update(cars: Cars, Authorize: AuthJWT = Depends()):

    Authorize.jwt\_required()

    admin\_id = Authorize.get\_jwt\_subject()

    update\_cars(cars.id, cars.name, cars.price, cars.company, admin\_id)

    return {"SUCCESS": "cars updated successfully"}

@router.delete("/cars")

def cars\_delete(id, Authorize: AuthJWT = Depends()):

    Authorize.jwt\_required()

    admin\_id = Authorize.get\_jwt\_subject()

    delete\_cars(id, admin\_id)

    return {"SUCCESS": "cars deleted successfully"}

@router.post('/login')

def login(admin: Admin, Authorize: AuthJWT = Depends()):

    if login\_admin(admin.admin\_id, admin.password):

        return {"ERROR": "Invalid Admin"}

    # subject identifier for who this token is for example id or username from database

    access\_token = Authorize.create\_access\_token(subject=admin.admin\_id)

    return {"JWT": access\_token}

@router.post('/signup')

def signup(admin: Admin, Authorize: AuthJWT = Depends()):

    create\_admin(admin.admin\_id, admin.password)

    access\_token = Authorize.create\_access\_token(subject=admin.admin\_id)

    return {"JWT": access\_token}