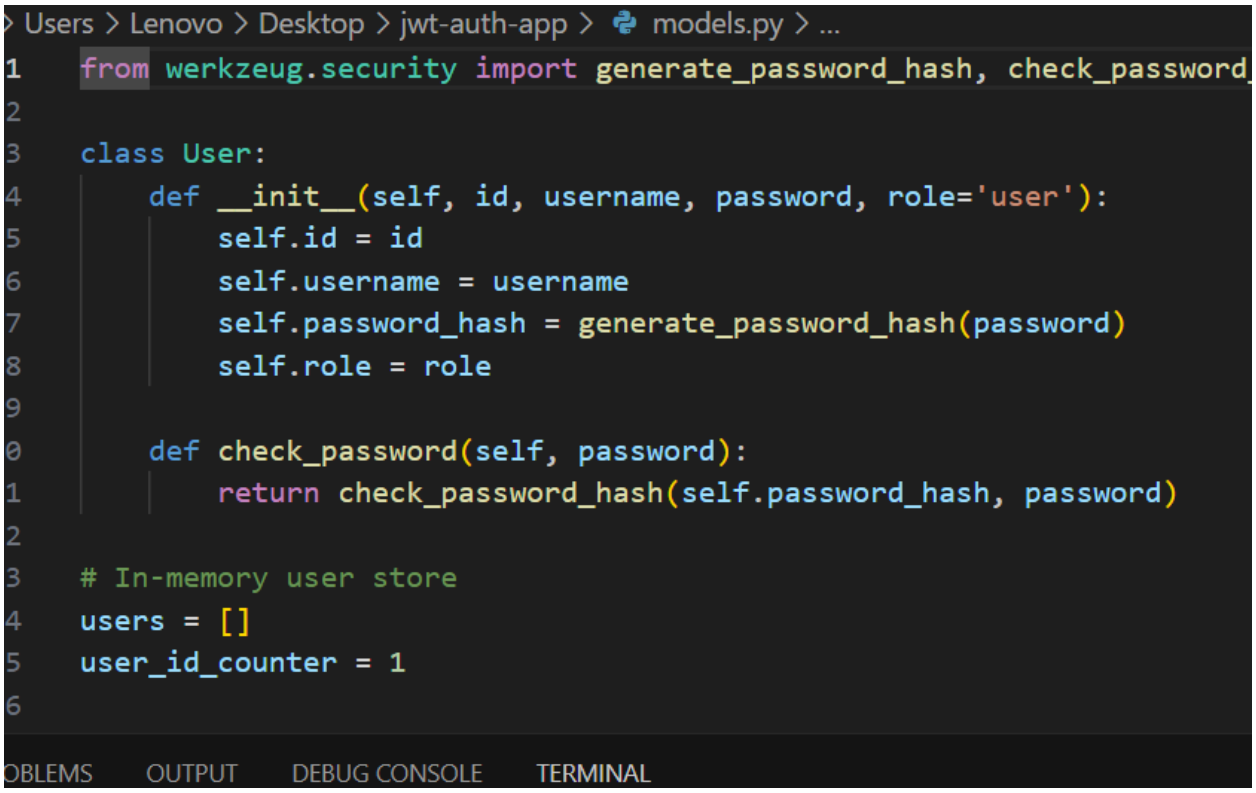
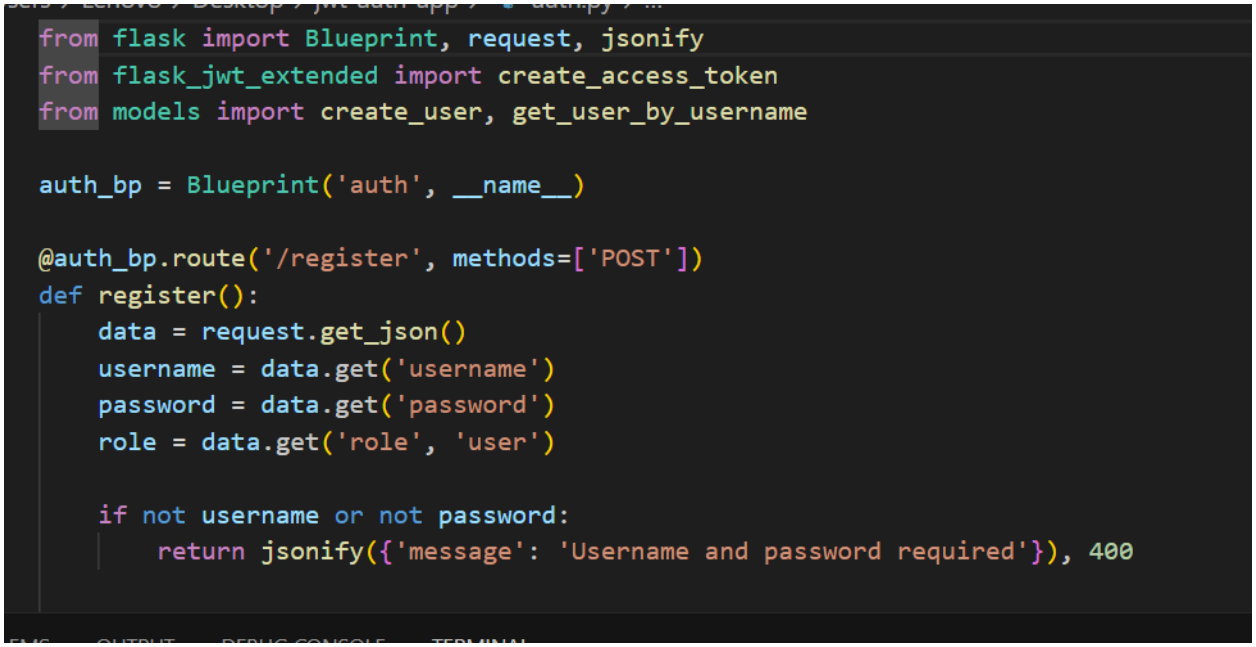
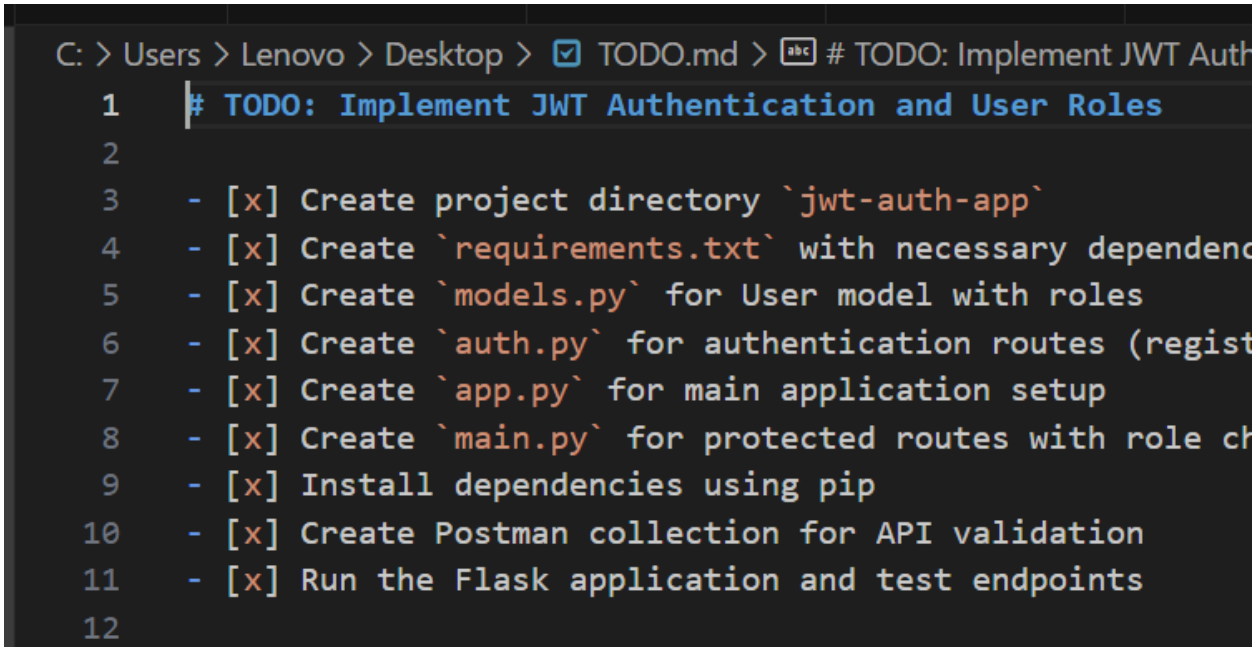
Experiment no 07

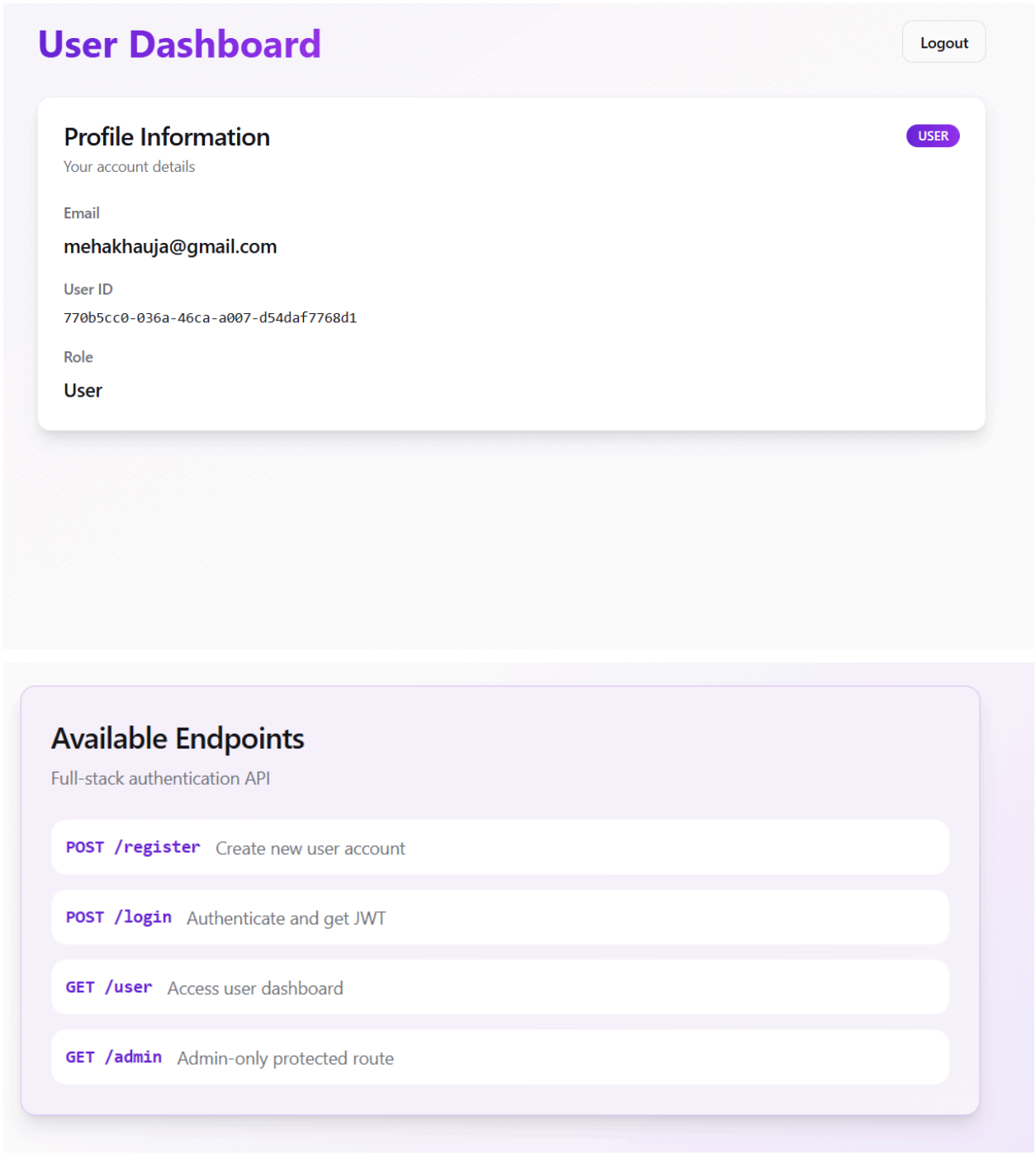
Validating RESTful APIs using Postman.






requirements.txt
models.py
auth.py
main.py
app.py
postman_collection.json
TODO
C: > Users > Lenovo > Desktop > jwt-auth-app > postman_collection.json > ...
{
"info": {
"name": "JWT Auth API Validation",
"description": "Postman collection for validating the JWT authentication API endpoints",
"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
},
"item": [
{
"name": "Register User",
"request": {
"method": "POST",
"header": [
{
"key": "Content-Type",
"value": "application/json"
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
powershell


C: > Users > Lenovo > Desktop > TODO.md > # TODO: Implement JWT Auth
TODO: Implement JWT Authentication and User Roles
- [x] Create project directory `jwt-auth-app`
- [x] Create `requirements.txt` with necessary dependenc
- [x] Create `models.py` for User model with roles
- [x] Create `auth.py` for authentication routes (regist
- [x] Create `app.py` for main application setup
- [x] Create `main.py` for protected routes with role ch
- [x] Install dependencies using pip
- [x] Create Postman collection for API validation
- [x] Run the Flask application and test endpoints


User Dashboard
Logout
Profile Information
Your account details
USER
Email
mehakhauja@gmail.com
User ID
770b5cc0-036a-46ca-a007-d54daf7768d1
Role
User
Available Endpoints
Full-stack authentication API
POST /register Create new user account
POST /login Authenticate and get JWT
GET /user Access user dashboard
GET /admin Admin-only protected route