CODE GRAPGERS PYTHON TEST TASK

Create a flask API using flask-restful and Postgres DB with the following requirements

Auth APIS:
1. create a signup API which stores the user in the Postgres DB (first name (required), last_name, email (required, unique and email validation), username(required and unique) password(required and at least 10 characters validation) it should include basic auth and validation
2. create a login API with username and password which included basic auth and also handles if user not found (the response should include jwt_token with user_id as payload, it will be needed for jwt authentication for below APIs)

Jobs APIS:
all jobs API's should include jwt authentication, the user_id will be fetched from jwt_token

1. create job API which will save job in the postures DB (job title, job description, job_rate, latitude, longitude, is_active(for soft delete), user_id(from jwt_token), job_created, job_updated)
2. update job API which will update job in the postures DB (job title, job description, job_rate, latitude, longitude, is_active(for soft delete), user_id(from jwt_token), job_created, job_updated)
3. delete API which soft delete job in the postures using is_active flag in DB
4. get jobs API list which will return the jobs of the user (it should include filtration by radius(kilometers) using the latitude, longitude)

++ POINTS:
Swagger documentation of the above APIs
Basic unit testing of the APIs

HELPING MATERIAL:
https://github.com/flask-restful/flask-restful
https://github.com/flasgger/flasgger
https://github.com/vimalloc/flask-jwt-extended
https://github.com/miguelgrinberg/Flask-HTTPAuth
https://github.com/pallets/flask-sqlalchemy
https://geoalchemy.readthedocs.io/en/0.5/tutorial.html
https://flask.palletsprojects.com/en/2.0.x/testing/