**Requirements:**

1. Create a simple customer data model,
2. Implement an API that will perform CRUD operations on the customer data model, and
3. Prepare documentation to guide a user on how to get the above running, and how to use it

**Python**

Use Python 3.6+, e.g. via pyenv and pipenv, and any framework of your choice (we strongly prefer Flask). Use any libraries you like, but do not use boilerplate code.

**Customer data model**

Create a data model in PostgreSQL

Customers (

id,

name,

dob,

updated\_at

)

**Build a basic RESTful JSON API**

Create RESTful API endpoints returning JSON so that a user can perform CRUD (create, read, update, delete) actions to the customers table. In addition, also implement a list action. List should take in a number **n as a GET parameter** that returns **n** youngest customers ordered by date of birth.

You may want to create a Postman collection containing a few example calls to these endpoints. <https://www.getpostman.com/>

**Authentication**

The API endpoints should not be publicly accessible, use JWT to implement authentication.

(*Optional credit*) replay attacks are possible with JWT since the auth data is stored client-side. Implement a simple method to block the recycling (replays) of old sessions. This process need not be comprehensive, just share an explanation of the potential pitfalls of what you’ve designed.

**Deployment or packaging** (optional + additional credit)

Prepare a docker image or docker-compose file that provides a means of easy deployment.

The build process should include the necessary initialization of Postgres and any misc. configuration.