**Notes from:**

**https://www.codementor.io/@olawalealadeusi896/restful-api-with-python-flask-framework-and-postgres-db-part-1-kbrwbygx5**

**Set Up Virtual Environment**

1. python3.9 -m venv env
2. source env/bin/activate

* To exit the venv, use commend “deactivate”. To reactivate, run “source env/bin/activate”

Install Dependencies

1. pip install flask==1.1.2
2. pip install pytest==6.1.1

Set up backend folder structure:

1. Backend > API > create \_\_init\_\_.py
2. Backend > tests > create \_\_init\_\_.py

Set up first test to test flask app:

1. from api.app import app
2. def test\_index():
3. tester = app.test\_client()
4. response = tester.get("/", content\_type="html/text")
5. assert response.status\_code == 200
6. assert response.data == b"Storage Rates"

Create postgres database:

$ createdb **[database name]**

Access db to confirm:

$ **psql [database name]**

Set system environment variables:

$ ex**port** FLASK\_ENV=development

$ ex**port** DATABASE\_URL= postgres://name:password@houst:**port**/blog\_api\_db

Create Flask app

from flask import Flask, request

from .config import app\_config

def create\_app(env\_name):

"""Creates app"""

# app initialization

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

app.config.from\_object(app\_config[env\_name])

@app.route('/', methods=['GET'])

def index():

return 'Your first endpoint is working!'

return app

app = create\_app()

app.run(debug = True)

Create an entry point, run.py, in the root folder. When you call this file, it will create a Flask object:

import os

from api.app import create\_app

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

env\_name = os.getenv('FLASK\_ENV')

app = create\_app(env\_name)

# run app

app.run(debug = True)

Execute python run.py and check that the Flask app runs. Go to the URL and check it returns your endpoint

Create your models. A model determines the logical structures of a database. Simply put, it determines how tables would look like on the database. Models define how records can be manipulated or retrieved in the database. Sqlalchemy is an ORM(Object Relational Mapper). An ORM is a database sql abstraction that makes it easy to carry out sql operations on relational database. With ORM, instead of writing raw sql queries(e.g to retrieve all rows from our User's table) we could do something like - model.query.all().

In models folder, create an \_\_init\_\_.py file:

from flask\_sqlalchemy import SQLAlchemy

# initialize our db

db = SQLAlchemy()

Add a RateModel.py with a class RateModel that inherits from the db object. Name our table and set the columns, constructor, save method, delete method, and class methods to extract data

Go back to app.py and import db and initialize it into the app

Migrate our model changes to the db with a migration script and MigrateCommand

Run migrations initialization with db init command:

python manager.py d binit