Solvei8 Backend Assignment

Name: G Rushivardhan

Phone number: 9603366515

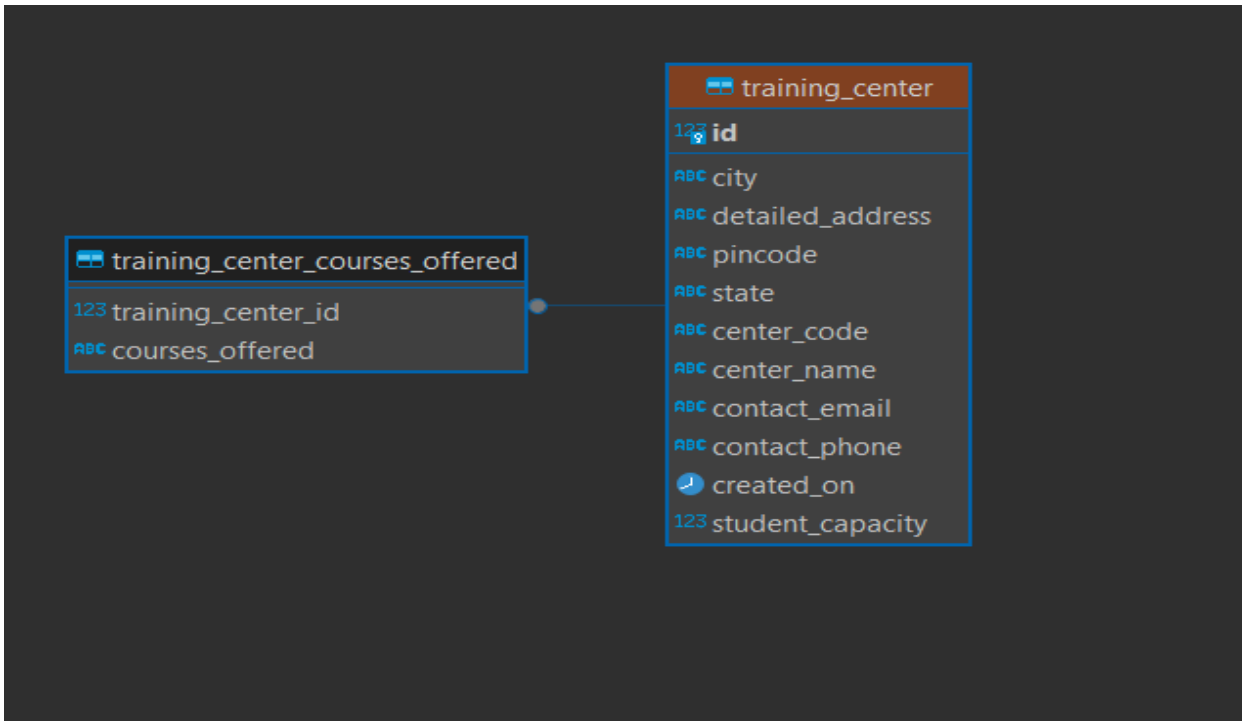
Email: rushivardhan18@gmail.com

Traini8: Problem Statement

Assume that you are working at a startup Traini8. You must make an MVP for a registry for Govt funded Training Centers with the following minimum requirements. In the MVP, you must create a Spring or Play project with support for two APIs, which are explained below.

Database details: PostgreSQL as the relational database management system for storing data.

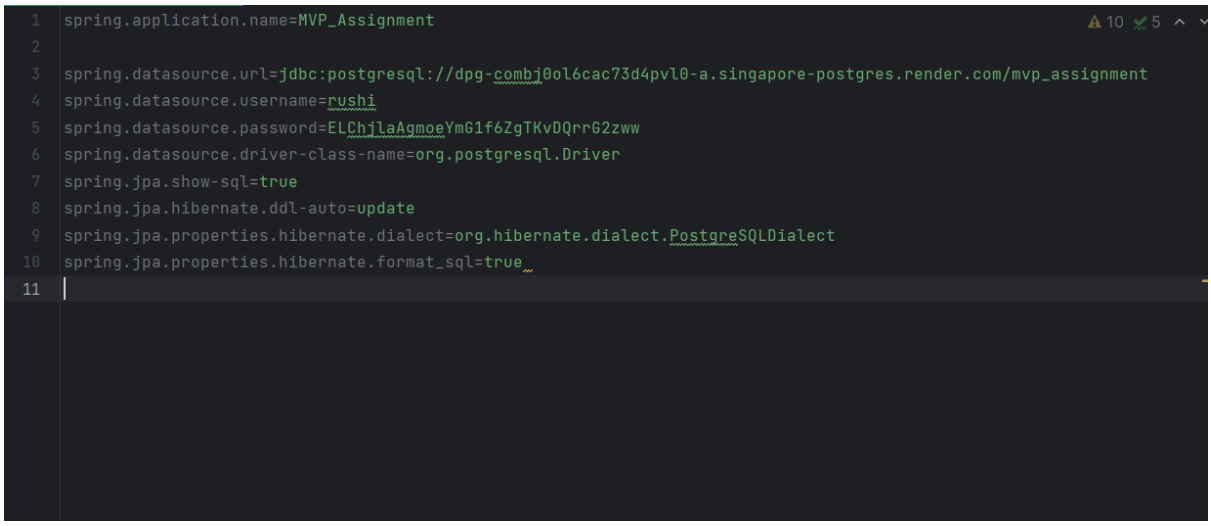
ERD:



Explanation:

Firstly, I created a spring boot project “Assignment” using “Spring Initializer” where I added all the required dependencies for my project and downloaded as a jar folder.

Then In Assignment project I added all the packages and classes required for it. Then added the Postgres properties in application properties file, and all other requirements.



```
spring.application.name=MVP_Assignment

spring.datasource.url=jdbc:postgresql://dpg-combj0ol6cac73d4pvl0-a.singapore-postgres.render.com/mvp_assignment
spring.datasource.username=rushi
spring.datasource.password=ELChjlaAqmoeYmG1f6ZgTKvDQrrG2zww
spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.properties.hibernate.format_sql=true
```

Entities:

- Training center: this contains all the fields given in the requirements along with a embedded address class.

- Address: this is a embedded class which encapsulates the fields of a address and helps in easy modifications and creation rather than single field in the main entity.

Functionality:

- As the given instructions in the assignment, when a new centre is created it will valid the given email and phone number whether they are in correct format or not.

- CenterName should be below 40, CenterID should be of length 12characters, email and phone number should bein correct format if wrong will return error.

Technologies Used:

1. Java Spring Boot for building the RESTful API.
2. PostgreSQL for Database management.
4. Postman for API documentation.
5. Render for Postgres

API Endpoints:

1.CreateCenters: Creates the centers based on the payload given in the assignment

METHOD: POST
URL: http://localhost:8080/api/training-centers/Create

Payload:


```
        {
  "centerName": "BBA Training Center",
  "centerCode": "ETA275TRR759",
  "address": {
     "detailedAddress": "BBA Training,Madhapur Street, Hyderabad",
     "city": "Hyderabad",
     "state": "Telangana",
     "pincode": "500001"
  },
  "studentCapacity": 500,
  "coursesOffered": [
     "Course 1",
     "Course 2",
     "Course 3",
     "Course 4"
  ],
  "contactEmail": "BBA@example.com",
  "contactPhone": "8921113517"
}
```


RESPONSE: Status code: 200

And will return the complete payload.

2) Getallcenters: Will return all the centres from the database.

METHOD: GET

URL: http://localhost:8080/api/training-centers/GetallCentres

RESPONSE:


```
                {
    "id": 7,
    "centerName": "Alpha2 Training Center",
    "centerCode": "ETC15ATRR759",
    "address": {
        "detailedAddress": "Alph2a Training,Madhapur Street, Hyderabad",
        "city": "Hyderabad",
        "state": "Telangana",
        "pincode": "500001"
    },
    "studentCapacity": 500,
    "coursesOffered": [
        "Course 1",
        "Course 2",
        "Course 3",
        "Course 4"
    ],
    "createdOn": "2024-04-27T17:35:27.618+00:00",
    "contactEmail": "ATranin2g@example.com",
    "contactPhone": "9711113597"
},
{
    "id": 8,
    "centerName": "Alpha Training Center",
    "centerCode": "ETA15ATRR759",
```

```
    "address": {
        "detailedAddress": "Alpha Training,Madhapur Street, Hyderabad",
        "city": "Hyderabad",
        "state": "Telangana",
        "pincode": "500001"
    },
    "studentCapacity": 500,
    "coursesOffered": [
        "Course 1",
        "Course 2",
        "Course 3",
        "Course 4"
    ],
    "createdOn": "2024-04-27T17:35:53.433+00:00",
    "contactEmail": "ATranin2@example.com",
    "contactPhone": "9711113517"
}
```


3)UpdateCenterByID: Update the center through ID. It will check the fields from the body and only change the fields while keeping the other fields same.

METHOD: PUT

URL: http://localhost:8080/api/training-centers/UpdateCenterById/7

REQUEST BODY:


```
                {
 "address": {
    "detailedAddress": "New Training center,Hitech Street,Hitech city,Training City",
    "city": "Hyderabad",
    "state": "Telangana",
    "pincode": "500001"
```

```
    },
    "studentCapacity": 10000
}
```


Response:


```
            {
    "id": 7,
    "centerName": "Alpha2 Training Center",
    "centerCode": "ETC15ATRR759",
    "address": {
        "detailedAddress": "New Training center,Hitech Street,Hitech city,Training City",
        "city": "Hyderabad",
        "state": "Telangana",
        "pincode": "500001"
    },
    "studentCapacity": 10000,
    "coursesOffered": [
        "Course 1",
        "Course 2",
        "Course 3",
        "Course 4"
    ],
    "createdOn": "2024-04-27T17:35:27.618+00:00",
    "contactEmail": "ATranin2g@example.com",
    "contactPhone": "9711113597"
}
```


4) DeleteCenterByID: deletes the center by ID.

METHOD: DELETE

URL: http://localhost:8080/api/training-centers/DeleteCenterByID/1

RESPONSE: STATUS CODE: 204

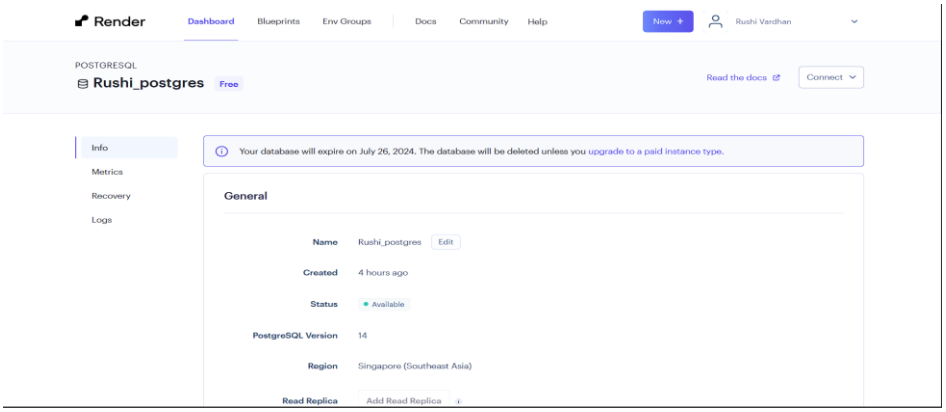
5)DELETEALL: DELETE's all the centers.

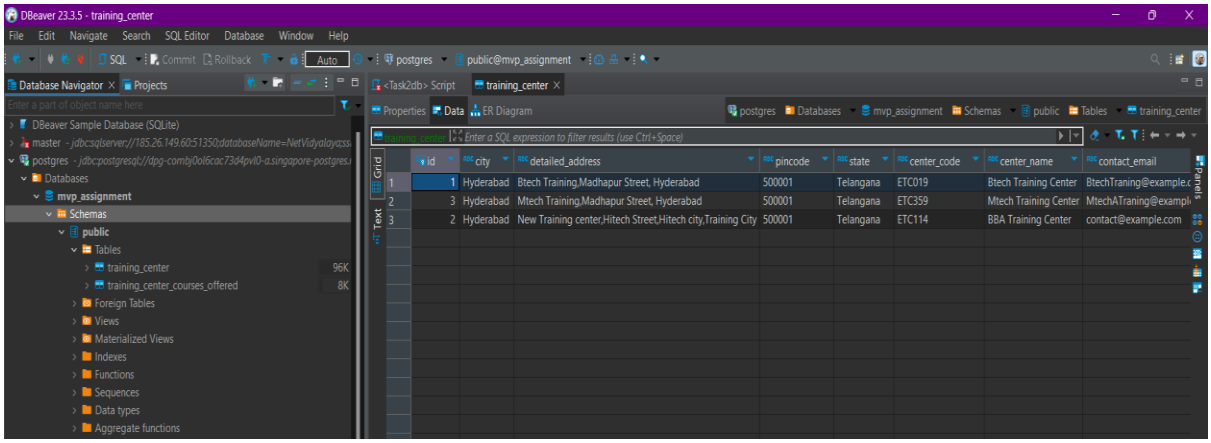
METHOD: DELETE
URL: http://localhost:8080/api/training-centers/DeleteAllCenters

RESPONSE: STATUS_CODE: 204

I have deployed Postgres using render and created a docker image for easy accessing of the endpoints.

POSTGRES DEPLOYMENT:





Please find the attached postman collection in the project folder.

Assignment Submitted To: **Buyogo**

Student Name: G Rushivardhan

College Name: Woxsen University

Email: rushivardhan18@gmail.com

Phno:9603366515