**SWE642 – HW3: Build survey form using Angular and Spring Boot**

**Team Member: Aniket Pandey, Siddhanth Kalyanpur, Adel Alkhamisy**

**Step 1: Write POST and GET api using Spring Tool Suite and connect to SQL**

* Following are the two APIs

Graphical user interface, text, application, email

Description automatically generated

* + GET http://{baseurl}/api/v1.0/surveys
  + POST http://{baseurl}/api/v1.0/surveys

Body

{

"firstName": "John",

"lastName": "Doe",

"email": "john.doe@mail.com",

"address": "9155 Walnut St",

"city": "Bentonville",

"state": "AR",

"zip": "22031",

"telephone": "8188884114",

"dateOfSurvey": "04-28-2022",

"recommendation": "Very Likely",

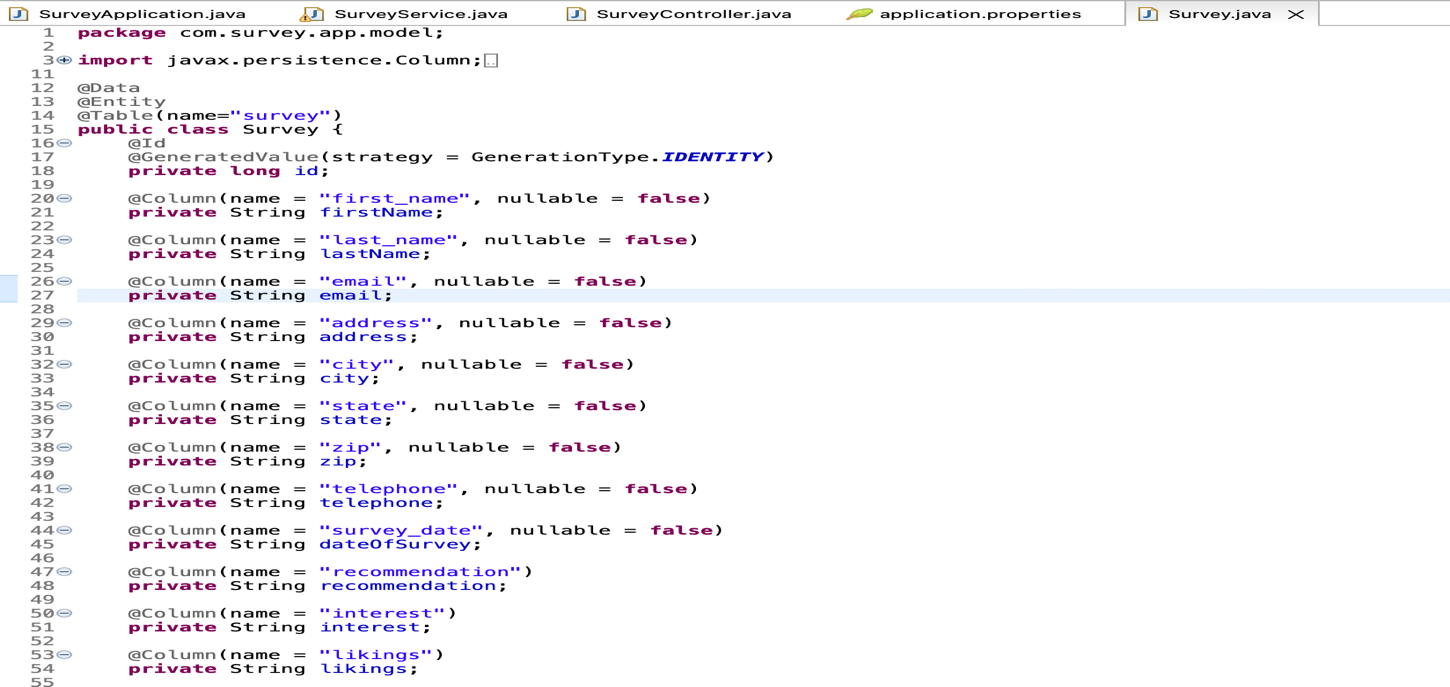
"interest": "Friends",

"likings": "Students",

"comment": "Good University",

}

* The Data Model looks like the following fields



* Write the API and create MySQL database using AWS RDS
* The API is written following the standard design pattern. First we have a controller layer which calls the service layer and finally the service layer calls the database layer to store data in database.
* The API project structure looks like below

A picture containing text

Description automatically generated

* + Package for running the application: survey.app
  + Package for controller: survey.app.controller
  + Package if any exception is thrown: survey.app.exception
  + Package for specifying contract to be followed by service layer: survey.app.interfaces
  + Package for data model: survey.app.model
  + Package for JPA and Hibernate configuration: survey.app.repository
  + Package for service layer: survey.app.service
* The JPA and Hibernate helps in auto creation of table and its column in database.
* We hosted our database on AWS RDS.
* Hosting database on RDS is comparatively easy just follow the step by step process as mentioned in the AWS documentation.
* Copy the database endpoint, port number, username and password
* Using JDBC connect to database and write the following lines in application.properties

Graphical user interface, text, application, website

Description automatically generated

* The api can be tested using Postman
* GET API

**Graphical user interface, text, application, email

Description automatically generated**

* POST API

**Graphical user interface, text, application, email

Description automatically generated**

* The Database record can be viewed using MySQLWorkbench

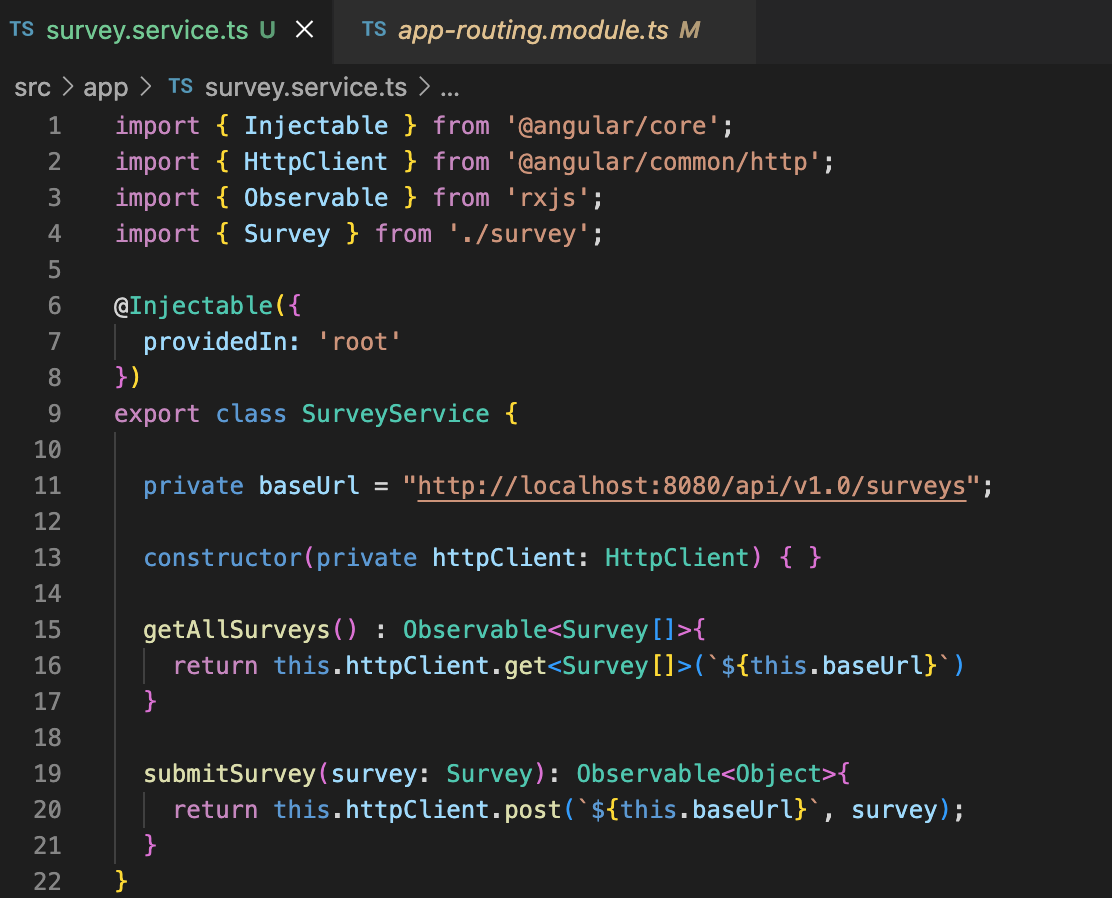
**Graphical user interface, text, application

Description automatically generated**

* Now right click on the pom.xml file and inside the Goal on the next step write “clean install” and then continue with the JAR packaging. The JAR file will be present inside target folder by default.

**Step 2: Create Angular application**

* Create a service call and make GET and POST calls to the api using HttpClient
* The url should be the url of the spring-boot application api on which the api is hosted. Check the image below with baseUrl.



* Following is the walk through of Angular project

Text

Description automatically generated

* + Class for data binding: survey.ts
  + Service for API calls: survey.service.ts
  + Submit form component: submit-survey
  + Get list of all the surveys: survey-list

**Step 3: Host API and UI**

* First host the API by running you application as spring-boot app.
* Second step is to run the UI by using command ng serve.
* Now open your browser and go to the URL: <http://localhost:4200>
* You should see a page as below.

Graphical user interface, text

Description automatically generated

* This is the homepage it has two navigation options one to view list of surveys done till date and the other is for submitting new survey

Graphical user interface, application, Word

Description automatically generated

* The above image displays list of surveys

Graphical user interface, application

Description automatically generated

* The above image displays the survey form with all the fields.
* As seen below after the form is submitted a toaster with acknowledgement is displayed.

Graphical user interface, application

Description automatically generated

* The UI calls the API can also be verified from the browser inspect.

Graphical user interface, application, Word

Description automatically generated