SWE-642 group:

Govind Sharma, Rachna Latkar, Bantwal Shreyas Mallya, Anjali Sharma

--------------------------------------------------------------------------------------------------------------------

SWE-642 Homework #3 Summary

Contents

[ReadMe 1](#_Toc141531413)

[Statement of contribution 1](#_Toc141531414)

[AWS MySQL RDS Database 2](#_Toc141531415)

[Spring Boot Rest API Application 7](#_Toc141531416)

[We created Maven spring boot starter project with spring web, spring data jpa and my sql driver dependencies. 7](#_Toc141531417)

[Spring Boot Start Class with annotation @SpringBootApplication: 7](#_Toc141531418)

[SurveyRepository: 7](#_Toc141531419)

[Survey Entity 8](#_Toc141531420)

[SurveyService 10](#_Toc141531421)

[SurveyController 11](#_Toc141531422)

[Angular2+ (Angular 12) Frontend Framework Application 12](#_Toc141531423)

[Spring Boot Rest API - get, post, put, delete tests via postman 17](#_Toc141531424)

[Integration of Rest API with Frontend Angular Application 23](#_Toc141531425)

[Running of the project 23](#_Toc141531426)

[References 28](#_Toc141531427)

# ReadMe

###### **The submission zip contains:**

1. setup-instructions document
2. all source files including Rest API, Angular Frontend, war file

# Statement of contribution

All group members did setup of various components involved in the homework including spring boot restful web services, Angular2+ frontend environment setup, web application setup, analysis, design, development, integration with Rest API and testing end to end. Then we compiled the documentation, reviewed, edited, and exported as PDF final submission. The following are specific focus areas for each member.

Shreyas Mallya: Worked on spring boot initial setup and implementation. Reviewed and edited the document.

Rachna: Worked on front-end application and testing. Reviewed and edited the document.

Govind Sharma: Worked on spring boot rest API, frontend integration with Rest API and testing. Reviewed and edited final document.

Anjali Sharma: Worked on AWS MySQL database setup, integration RDS MySQL with Rest API and implementation of frontend and back end. Took a first cut at the submission documentation.

# AWS MySQL RDS Database

1. Login to console

<https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;s3-import=false>

1. In Services, search for RDS
2. Click Create database
3. Choose MySQL

A screenshot of a computer

Description automatically generated

1. Keep default Engine Version
2. Choose Free tier.

A screenshot of a computer

Description automatically generated

1. Under Settings, Provide unique name for database instance and setup admin master password.

A screenshot of a computer

Description automatically generated

1. Make sure MySQL port is open or for the purpose of the project, you may allow All Traffic.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Keep default selections.
2. See estimated cost ~$15/month
3. Hit Create database
4. After some time, you will see, database Instance is running

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Spring Boot Rest API Application

We created Maven spring boot starter project with spring web, spring data jpa and my SQL driver dependencies.

Maven pom with required dependencies:

A screen shot of a computer program

Description automatically generated

## Spring Boot Starter Class with annotation @SpringBootApplication:

**/hw3-rest/src/main/java/com/stil/hw3/Hw3Application.java**

A screen shot of a computer code

Description automatically generated

## SurveyRepository:

**/hw3-rest/src/main/java/com/stil/hw3/repository/SurveyRepository.java**

A screenshot of a computer code

Description automatically generated

## Survey Entity

**/hw3-rest/src/main/java/com/stil/hw3/entity/Survey.java**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

## SurveyService

**/hw3-rest/src/main/java/com/stil/hw3/service/SurveyService.java**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

## SurveyController

**/hw3-rest/src/main/java/com/stil/hw3/controller/SurveyController.java**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

# Angular2+ (Angular 12) Frontend Framework Application

* Create new angular project for homework
* ng new hw3-web
* Router with two links with survey being default route when application runs:
* src\app\app-routing.module.ts

A screenshot of a computer code

Description automatically generated

* src\app\app.component.html

A screen shot of a computer code

Description automatically generated

* src\app\app.module.ts

A screenshot of a computer program

Description automatically generated

* src\app\survey.service.ts

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

* There are two controllers one for survey form functionality and other for listing all surveys coming from MySql database via Rest API.

A screenshot of a computer

Description automatically generated

* src\app\survey\survey.component.ts
* Survey component calls service injected in constructor which in turn calls Rest API to persist survey form data

A screen shot of a computer

Description automatically generated

* src\app\surveysuccess\surveysuccess.component.ts
* This controller calls service to fetch all surveys so far exist in the database. As user successfully completes the form and submits, new row is added for “List All Surveys” page.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

# Spring Boot Rest API - get, post, put, delete tests via postman

We used postman to verify Rest API application is working as intended, persisting to MySql database and getting data from Rest API deployed at <http://localhost:9191/surveys>.

#### Add One/Single Survey:

A screenshot of a computer

Description automatically generated

#### Add List of Surveys:

A screenshot of a computer

Description automatically generated

#### Get All Surveys:

A screenshot of a computer

Description automatically generated

#### Get Survey By Id:

A screenshot of a computer

Description automatically generated

#### Get Survey By First Name:

A screenshot of a computer

Description automatically generated

#### Delete Survey By Id:

A screenshot of a computer

Description automatically generated

#### Update Survey:

A screenshot of a computer

Description automatically generated

# Integration of Rest API with Frontend Angular Application

Angular frontend application calls Rest API post and get methods mapped to the URLs. Angular controller calls global service to post survey or get all surveys. Please see screen shots in Angular frontend application description.

# Running of the project

Please follow the steps below to make sure all components involved in the project are up and running in the order.

Check the database is running as identified in the Rest API application.properties file. I am using DBeaver for AWS RDS setup.

A screenshot of a computer

Description automatically generated

Start spring boot restful webservices project first. Look for starter class which is annotated with “@SpringBootApplication” 🡪right click 🡪run as Java Application

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Verify by get URL or by using postman (API testing tool explained above in Rest API section)

<http://localhost:9191/surveys>

You should see Json is returned listing all surveys.

A close-up of a white background

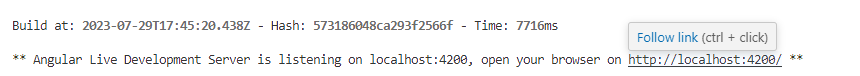
Description automatically generated

##### After successfully starting and verifying Rest API application, go to angular project home , perform below steps

* npm install (do this first time so that dependencies come to your local environment)
* ng serve (wait for it to compile and provide URL)

A screenshot of a computer

Description automatically generated



* Follow the URL it gives after successful compilation

<http://localhost:4200/> which will redirect to <http://localhost:4200/survey> as we have setup default route path as survey form (explained in Angular frontend project setup)

* At the top right hand side, you see two links, Survey Form and List All Surveys

A screenshot of a survey form

Description automatically generated

* Click on link “List All Surveys” to view all surveys.

A screenshot of a computer

Description automatically generated

##### Validation:

* firstName, lastName, Street Address, city, state, zip, phone, email and date of survey are required fields and have proper validation by Angular2+ validation mechanism such as required, invalid, format matching, touched.
* Form submission button is disabled until form is validated with all required values and proper format.
* If a field is touched and does not have valid input, the application will provide appropriate error message.

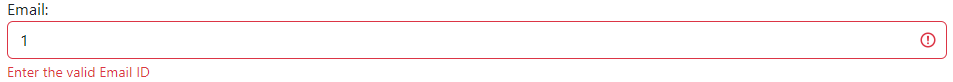
A screenshot of a survey form

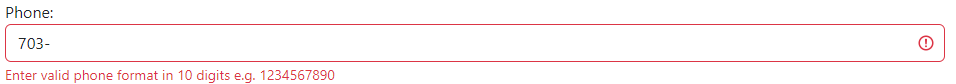
Description automatically generated





* If the field is touched and the value entered is not in proper format as expected, you will get appropriate message such as enter valid Email ID.





#### Example valid form with Submit Feedback enabled when form has all required information:

After successful submission, the application will acknowledge student feedback with confirmation message.

A blue square with black text

Description automatically generated

A screenshot of a survey

Description automatically generated

A screenshot of a computer

Description automatically generated

##### Trouble shoot connection issue with API and database:

Since the angular2+ frontend connects to Restful API application at (<http://localhost:9191/addSurvey>) for posting the survey to API and persisting to database, if you see submit is taking too long, restart spring boot API application again. The connection might be broken.

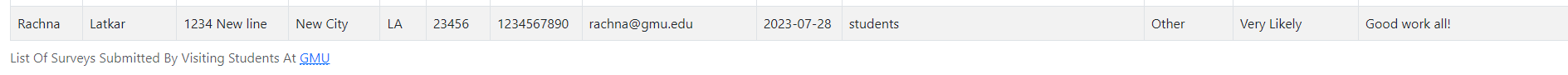
Once successful submission, you can verify that survey persists in database by calling get all surveys API URL from browser or from postman/insomnia.

<http://localhost:9191/surveys>

you will see newly persist survey will be added as last row with your entries.

You may also verify by calling web application “List All Surveys”

<http://localhost:4200/surveysuccess> , new submitted survey will be listed as last row.



Clear button is provided to rest form to its original state by clearing user entries.

# References

SWE-642 course lectures

<https://spring.io/>

https://angular.io/

<https://docs.aws.amazon.com/>