

SWE 645: Assignment 3

Angular, RESTful Web Services, and JPA

& Optional Extra Credit: Using Containerized Database

Your next assignment is to develop a single page web application using Angular2 (or later versions), RESTful Web Services and JPA/Hibernate along with Amazon Relational Database Service (Amazon RDS)/MySQL to persist and read data to/from a relational database. The application allows prospective students to fill out a survey form to provide feedback about their campus visit. It also allows users to view all surveys recorded to date. The application starts with a welcome homepage, which in essence has two links: 1) Student Survey, which allows a prospective student to fill out a survey form, with an acknowledgement and 2) List All Surveys, which allows a user to view all surveys done to date.

You may want to use Visual Studio Code to develop the Angular application and use a dynamic web project in Eclipse EE to develop RESTful web services and integration with persistence layer of your application. Please containerize your applications using at least two containers: 1) container for your Angular application, and 2) container for your RESTful APIs, using Docker technology. Deploy the containerized applications on the container orchestration platform Kubernetes or any managed Kubernetes platform that you established for the last homework. Integrate the Kubernetes platform with the CI/CD pipeline that you developed in your last homework. Also please add a hyperlink of this homework solution on your class homepage on S3.

Use Amazon RDS to provision and use a MySQL database for this homework. Refer to an additional document titled “SWE645-HW5-AppendixA-Using Amazon RDS to create MySQL Database” available on the class blackboard Course Content/Homeworks folder.

For this homework, the student survey form contains the following.

- Text boxes for first name, last name, street address, city, state, zip, telephone number, e-mail, and date of survey, which are required fields.
- Checkboxes that allow prospective students to indicate what they liked most about the campus. The checkboxes should include: students, location, campus, atmosphere, dorm rooms, and sports.
- Radio buttons that allow the prospective students to indicate how they became interested in the university. Options should include: friends, television, Internet, and other.
- A dropdown list of options that allows student to select from a list of options to select the likelihood of him/her recommending this school to other prospective students. The three options of the dropdown list are: Very Likely, Likely, Unlikely.
- A submit and cancel buttons.

This homework can be done individually or in a group of maximum 4 students.

Optional Extra Credit:

Instead of using Amazon RDS based MySQL database, use containerized MySQL database along with the concepts of Persistent Volume, Persistent Volume Claim, and Storage Class to persist/retrieve student survey data in your homework

Submission

The submission for this assignment should be through the blackboard website. I expect a zipped package containing the source files, configuration files, such as Dockerfile, Jenkinsfile, war file, and any additional packages, scripts, or files that you used. I also require a readme file which contains installation and setup instructions, including references, of the tools you used so that the TA and myself can replicate your steps and deploy and run the assignment. I also expect AWS URL of your homepage as part of readme file. Submit all source, and war file, files necessary to run the application and the installation and execution instructions in a zip file.

Please add the link of your application to your website on Amazon S3, that you created in the Part1 section of this assignment. Also, provide the URL of your homepage as well as of the application deployed on Kubernetes in readme file as part of your HW submission on the class blackboard. Please provide a video recording demonstrating the working of every part of your application and make it a part of your submission. In addition, schedule a meeting with the GTA or the professor to demo your work.

NOTE: A late assignment carries a 10% late penalty for each week it is late. Assignments are NOT accepted after being 2 weeks late. Make sure your or your group's name is on every programming artifacts so we know who it belongs to. For every source file, please include comments at the top of the program describing what the program does. This only needs to be 1 or 2 sentences. Be sure to test access and functionality to your submission before the due date.

Grading:

The following areas will be used in the basic grading of these projects:

- Does system meet the functional requirements along with proper documentation and video recording: 85 points
- Does the assignment run without errors: 13 points
- Comments: 2 points

Instant Point Deductions:

I reserve the right to deduct points instantly for the following reasons:

- The source, or binary, files are not included in the package.
- The readme file is not included in the package.
- The program doesn't run due to errors in the code.
- I spend more than 5 minutes trying to debug the assignment.
- I can't figure out how to use the assignment, and instructions are left out.

