SWE 645 Assignment 3
(Mahbubul Alam Palash & Taseef Rahman)

Set UP of Java Rest API:
A Maven java Dynamic Web project has been created to implement the Java Rest API to enter new survey information and retrieve all the survey information. The Java Web project uses java JPA to enter and retrieve values to and from the mysql database deployed in a Google Kubernetes Engine container.

A java web project was created first then it was converted to jpa then to Maven Project.All the required dependencies were provided to maven by the configuration file **pom.xml** and all database credentials were provided using **persistence.xml.**

The url pattern to access the rest api was provided in **web.xml.** And the url to access the rest api are:

Getting all surveys:
http://35.224.162.205/surveywebjpa-RestAPI/rest/surveys/allsurvey

To enter new Survey data:

http://35.224.162.205/surveywebjpa-RestAPI/rest/surveys/new

Webhook & Jenkins continuous integration to GKE:

The project is then pushed to a repository in github and a pipeline is created in jenkins using the webhook from the repository. A Dockerfile to containerize the application along with Jenkins file is written and added to github repository. Then the image is deployed in GKE as **swe645final** and a service to expose to the outside world named

swe645final is also created.

After each commit in github a new image is built, pushed to docker hub and the image in GKE container is updated with a new image.

After deploying in GKE angular app can use the links to access the Rest APIS for survey.

Setup for angular app-

Import HttpClientModule and ReactiveFormsModule into app.module.ts.
Then we can generate 2 components -first and second which will act as links to fill new student form and get previous forms.

Now in first.component.ts, we can declare our formgroups and other variables.

In this case, the form control values are- first_name, last_name, address,city, state, zip , tel_num, email,dos, likings,likelihood and interest.

For example, it contains a list named likes where the values are 'Very likely', 'Likely', 'Unlikely' to help us make a dropdown from these values.

When the create button is clicked, we can call REST API's post method to create a new student. Before that we have to check if the values that are being passed are valid or not.

In the second component, we just have the functionalities for a button that calls REST API's get method to obtain previously submitted forms.

Webhook & Jenkins continuous integration to GKE:

The project is then pushed to a repository in github and a pipeline is created in jenkins using the webhook from the repository. A Dockerfile to containerize the application along with Jenkins file is written and added to github repository. Then the image is deployed in GKE as **swe645angular-app** and a service to expose to the outside world named **swe645angular-app** is also created. After each commit in github a new image is built, pushed to docker hub and the image in GKE container is updated with a new image.

The links of the Angular app deployed in GKE is :
http://35.188.166.111

Video link for CI/CD angular pipeline-
https://drive.google.com/file/d/1qU1SNW2R2AqPcbhMnuQklnCkax-xgtBz/view?usp=sharing(*note that the browser cache needs to be cleared to see the change taking place*)
Video link for REST Angular implementation-
https://drive.google.com/file/d/1lBMOAZvC-eO32Ntd_BniB5qg3z0-v3LQ/view?usp=sharing

Extra Credit:

Instead of using Amazon RDS based MySQL database, we have used containerized MySQL database along with the concepts of Persistent

Volume, Persistent Volume Claim, and Storage Class to persist/retrieve student survey data in our homework