**EXTRA CREDIT ASSIGNMENT - GROUP PROJECT**

**DOCUMENTATION**

**Team Members:**

**Deepika Hemant Tendulkar - G01451861**

**Sanjana Vegesna – G01447619**

**Sai Abhishek Nemani - G01462099**

**1. Overview**

The Student Survey Website is a web application for creating, managing, and viewing student surveys. The application consists of:

- **Frontend**: Built with Vue.js, managing user interactions.

- **Backend**: Built with Java Spring Boot, handling business logic and data operations.

- **Database**: Amazon RDS for data storage.

The frontend and backend are deployed on separate Google Kubernetes Engine (GKE) clusters.

**Architecture:-**

**1. Frontend:**

- Technologies:

* Vue.js
* HTML5, CSS3, JavaScript
* Vue Router (for navigation)

- Key Components:

- HomePage.vue: Main landing page with navigation buttons.

- SurveyForm.vue: Allows users to fill and submit surveys.

- SurveyList.vue: Displays a list of surveys from the database with options to edit or delete.

- Routing: Managed by index.js in the route folder.

- Deployment: Deployed on a GKE cluster.

**2. Backend:**

- Technologies:

* Java (JDK 11+)
* Spring Boot Framework
* Spring Data JPA (for database access)

- Functionality:

- Handles API requests from the frontend.

- Interfaces with Amazon RDS for data storage and retrieval.

- Deployment: Deployed on a separate GKE cluster.

**3. Database:**

**Amazon RDS:**

Stores survey data.

Used for persisting submitted surveys and providing data for listing.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Frontend Components**

**1. HomePage.vue**

Landing page of the application.

- Features:

- Two buttons:

1. Navigate to SurveyList Page.

2. Navigate to SurveyForm Page.

A screenshot of a survey form

Description automatically generated

A screen shot of a computer program

Description automatically generated

2. SurveyForm

Allows users to fill out and submit survey forms.

- Features:

- Dynamic form fields for capturing survey details.

- Submit button:

- Sends a POST request to the backend API.

- Backend saves the submitted data to Amazon RDS.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

3. SurveyList

Displays a list of all surveys stored in the database.

- Features:

- Fetches all survey data from the backend API.

- Provides options to:

1. Edit: Update an existing survey.

2. Delete: Remove a survey from the database.

Screens screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

4. Routes (index.js)

- Manages the navigation within the Vue.js application.

- Defined Routes:

- /: Renders HomePage.vue.

- /survey-list: Renders SurveyList.vue.

- /survey-form: Renders SurveyForm.vue.

A screen shot of a computer program

Description automatically generated

**5. App.vue**

This is the root Vue component. It contains a <router-view /> tag that dynamically loads child components based on the route.

A screenshot of a computer program

Description automatically generated

Backend Overview

**Backend Structure**

1. **Controller**:

* Handles API requests for survey operations.
* Example: POST /api/surveys to save survey data.

1. **Service**:

* Contains the business logic for saving and retrieving survey data.

1. **Repository**:

* Uses Spring Data JPA for database operations.

A screenshot of a computer program

Description automatically generated

**APIs**

- Survey Creation:

- Method: POST

- Endpoint: /api/v2/surveys

- Description: Saves a new survey to the database.

- Fetch Survey List:

- Method: GET

- Endpoint: /api/v2/surveys

- Description: Retrieves all surveys from the database.

- Update Survey:

- Method: PUT

- Endpoint: /api/v2/surveys/{id}

- Description: Updates an existing survey.

- Delete Survey:

- Method:DELETE

- Endpoint: /api/v2/surveys/{id}

- Description: Deletes a survey by ID.

**Deployment**

1. Frontend Deployment

- Platform: Google Kubernetes Engine (GKE)

- Steps:

- Build the Vue.js application using npm run build.

- Create a Docker image for the frontend.

- Deploy the image to a GKE cluster using Kubernetes manifests.

External IP for front-end(vue.js) : http://34.41.5.17/

2. Backend Deployment

- Platform: Google Kubernetes Engine (GKE)

- Steps:

- Package the Spring Boot application using Maven .

- Create a Docker image for the backend.

- Deploy the image to a separate GKE cluster using Kubernetes manifests.

External IP for Back-end(spring app) : http://34.27.237.200/api/v2/surveys

3. Database Configuration

- Platform: Amazon RDS

- Configuration:

- Ensure network connectivity between the clusters and the RDS instance by modifying the inbound rules in the security groups tab to allow traffice from the clusters.

Access the Application:

Our application is deployed on <http://34.41.5.17/>

**How to Run the Code**

**Backend Setup**

1. mvn clean package
2. java -jar target/student-survey-0.0.1-SNAPSHOT.jar

**Frontend Setup**

1. Navigate to the vue-frontend folder.
2. Install dependencies:

npm install

npm install -g @vue/cli

vue create vue-frontend

npm install vue-router@next

npm install axios

1. Start the Vue app:

npm run serve