

# Full Stack Application Report

## Introduction

This report provides a detailed overview of a full stack application developed with AngularJS as the frontend framework and Java Spring Boot as the backend framework. The application uses a MySQL database for performing CRUD (Create, Read, Update, Delete) operations through the Java Persistence API (JPA).

## Frontend: AngularJS

The frontend of the application is built using AngularJS, a JavaScript-based framework for building dynamic web applications.

## Initial Setup

- Node.js runtime environment
- AngularJS framework
- Angular CLI for project scaffolding and build
- Visual Studio Code editor

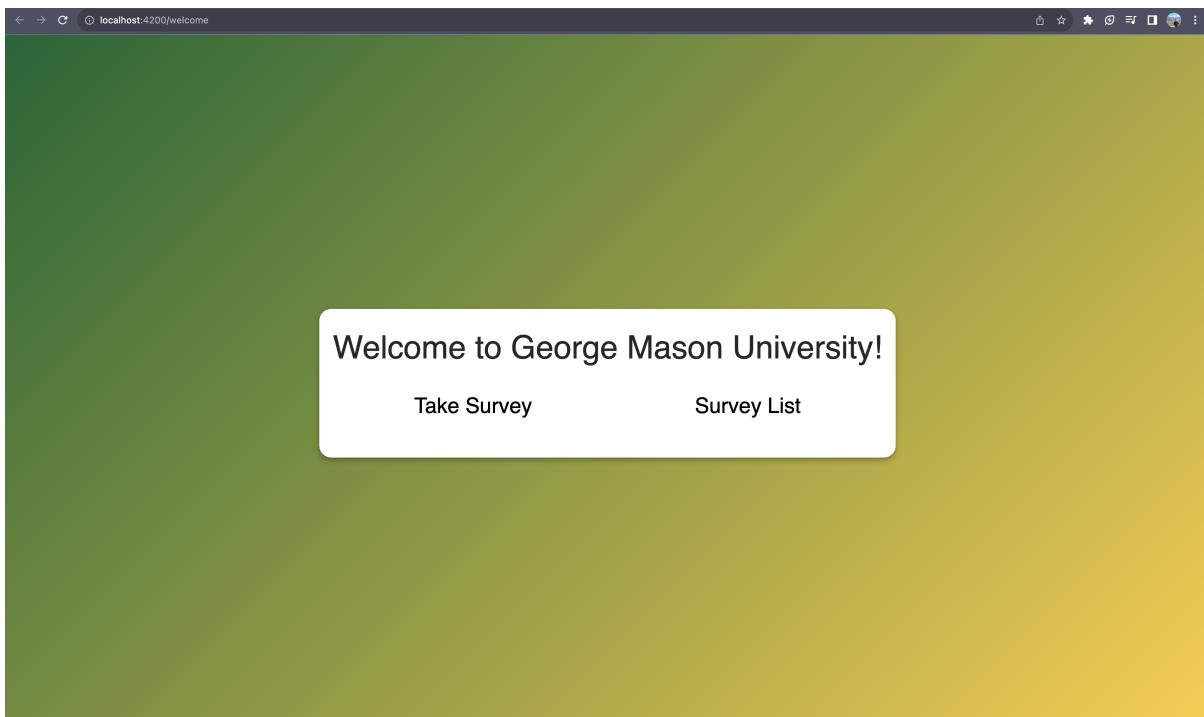
## Key Components

The application consist of various components to serve the different parts of the task.

1. Welcome Component:
  - Provides options to fill out a survey form or view a list of submitted surveys
2. Survey Form Component:
  - Form with validations to capture all required survey details. Makes API call to save data.
3. Survey List Component:
  - Calls API to retrieve all surveys and display them with options to delete or update each one.
  - Update functionality parses form data and updates the database upon submission.

- Delete operation updates the screen and removes the form data from the database.

## Screenshots



A screenshot of a web browser window showing the "Student Survey" form. The URL in the address bar is `localhost:4200/take-survey`. The form is contained within a white card-like container with a shadow. It consists of several input fields with labels and placeholder text. The fields include:

- First Name\*:
- Last Name\*:
- Username\*: (User name is required.)
- Email\*: (Email id is required.)
- Street Address\*: (Street address is required.)
- Zip Code\*
- State\*
- City\*
- URL
- Date\*
- Year of Graduation

List of Surveys	
First Name: Pranay Last Name: Sharma Email: john.doe@example.com Date: 2023-11-18	<a href="#">Update</a> <a href="#">Delete</a>
First Name: John Last Name: Doe Email: john.doe@email.com Date: 2023-01-15	<a href="#">Update</a> <a href="#">Delete</a>
First Name: Robin Last Name: kick Email: robbie@email.com Date: 2023-04-18	<a href="#">Update</a> <a href="#">Delete</a>
First Name: Sam Last Name: Cummin Email: scummin@email.com Date: 2023-10-18	<a href="#">Update</a> <a href="#">Delete</a>
First Name: Simmon Last Name: James Email: sjames22@email.com Date: 2023-02-30	<a href="#">Update</a> <a href="#">Delete</a>
<a href="#">Home</a>	

## Backend: Spring Boot

The backend was created with Spring Boot framework written in Java connecting to a MySQL database using JPA for object-relational mapping.

### Initial Setup

- Java Development Kit (JDK): Ensure JDK 8 or later is installed on the server.
- Integrated Development Environment (IDE): Use an IDE like IntelliJ or Eclipse for a better experience.

### Key APIs

1. Save Form API (`/saveform`):
  - Accepts a `SurveyForm` entity request and persists the new record in the database.
2. Get All Surveys API (`/getall`):
  - Returns a list of all survey forms stored in the database.
3. Delete Survey API (`/deletesurvey/{id}`):

- Accepts the form ID as a path variable, checks for the presence of the record, and deletes it.

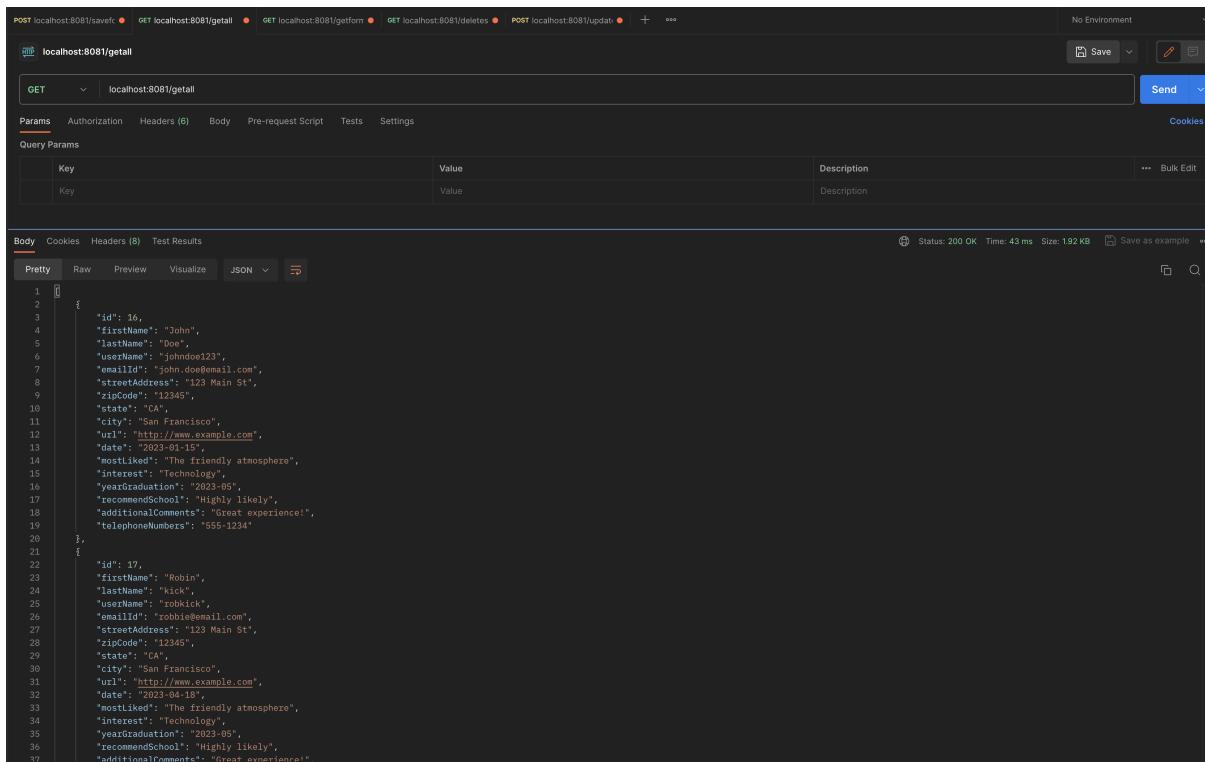
#### 4. Get Form API (`/getform/{id}`):

- Accepts the form ID as a path variable, retrieves and returns the form data as a `SurveyForm` POJO.

#### 5. Update Form API (`/updateform/{id}`):

- Accepts form data as the request body and the form ID as a path variable.
- Updates the existing record in the database.

## Screenshots



```

POST localhost:8081/savefc | GET localhost:8081/getall | GET localhost:8081/getform | GET localhost:8081/delete | POST localhost:8081/update | + | ...
No Environment
localhost:8081/getall
GET | localhost:8081/getall | Save | Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies
Query Params
Key Value Description Bulk Edit
Key Value Description
Body Cookies Headers (8) Test Results
Pretty Raw Preview Visualize JSON
1 [
2   {
3     "id": 16,
4     "firstName": "John",
5     "lastName": "Doe",
6     "userName": "johndoe123",
7     "emailId": "john.doe@email.com",
8     "streetAddress": "123 Main St",
9     "zipCode": "12345",
10    "state": "CA",
11    "city": "San Francisco",
12    "url": "http://www.example.com",
13    "date": "2023-03-15",
14    "mostLiked": "The friendly atmosphere",
15    "interest": "Technology",
16    "yearGraduation": "2023-05",
17    "recommendSchool": "Highly likely",
18    "additionalComments": "Great experience!",
19    "telephoneNumbers": "555-1234"
20  },
21  {
22    "id": 17,
23    "firstName": "Robin",
24    "lastName": "Kirk",
25    "userName": "robin.kirk",
26    "emailId": "robin.kirk@email.com",
27    "streetAddress": "123 Main St",
28    "zipCode": "12345",
29    "state": "CA",
30    "city": "San Francisco",
31    "url": "http://www.example.com",
32    "date": "2023-04-18",
33    "mostLiked": "The friendly atmosphere",
34    "interest": "Technology",
35    "yearGraduation": "2023-05",
36    "recommendSchool": "Highly likely",
37    "additionalComments": "Great experience!"
38  }
]
  
```

POST localhost:8081/saveform

localhost:8081/saveform

Body

```

1
2   "firstName": "Simon",
3   "lastName": "James",
4   "userName": "simonj123",
5   "emailId": "simonj123@mail.com",
6   "streetAddress": "123 Main St",
7   "zipCode": "12345",
8   "state": "CA",
9   "city": "San Francisco",
10  "url": "http://www.example.com",
11  "date": "2023-02-30",
12  "mostLiked": "The friendly atmosphere",
13  "interest": "Technology",
14  "yearGraduation": "2023-05",
15  "recommendSchool": "Highly likely",
16  "additionalComments": "Great experience!",
17  "telephoneNumbers": "555-1234"
18
19

```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "message": "Successfully Saved",
3   "firstName": "Simon"
4

```

Status: 200 OK Time: 19 ms Size: 308 B Save as example

POST localhost:8081/getform/16

localhost:8081/getform/16

GET

Query Params

Key	Value	Description
Key		Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "id": 16,
3   "firstName": "John",
4   "lastName": "Doe",
5   "userName": "john doe123",
6   "emailId": "john.doe@email.com",
7   "streetAddress": "123 Main St",
8   "zipCode": "12345",
9   "state": "CA",
10  "city": "San Francisco",
11  "url": "http://www.example.com",
12  "date": "2023-01-15",
13  "mostLiked": "The friendly atmosphere",
14  "interest": "Technology",
15  "yearGraduation": "2023-05",
16  "recommendSchool": "Highly likely",
17  "additionalComments": "Great experience!",
18  "telephoneNumbers": "555-1234"
19

```

Status: 200 OK Time: 24 ms Size: 682 B Save as example

POST localhost:8081/savef

localhost:8081/deletesurvey/19

DELETE

Query Params

Key	Value	Description
Key		Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "message": "Successfully deleted",
3   "firstName": "19"
4

```

Status: 200 OK Time: 41 ms Size: 304 B Save as example

The screenshot shows the Postman interface with a POST request to `localhost:8081/updateform/20`. The request body is a JSON object containing student survey data. The response status is 200 OK, indicating success.

```

1
2   "firstName": "True",
3   "lastName": "DoeUpdate",
4   "userName": "johndoe0123",
5   "emailId": "john.doe@email.com",
6   "streetAddress": "123 Main St",
7   "zipCode": "12345",
8   "state": "CA",
9   "city": "San Francisco",
10  "url": "http://www.example.com",
11  "date": "2023-01-15",
12  "mostliked": "The friendly atmosphere",
13  "interest": "Technology",
14  "yearGraduation": "2023-05",
15  "recommendSchool": "Highly Likely",
16  "additionalComments": "Great experience!",
17  "telephoneNumbers": "555-1234"
18
  
```

**Body** tab shows the JSON response:

```

1
2   "message": "Successfully Updated",
3   "firstName": "True"
  
```

## Running the Application

Backend Spring Boot app will run on port 8081 and the frontend Angular app will be served on the port 4200. The applications interact via the exposed APIs listed above.

## Conclusion

In summary, the frontend and backend applications seamlessly integrated to provide a user-friendly interface for student survey. The use of MySQL as the database ensure the data is persisted and handled efficiently. The application is light weight, modular and scalable for future enhancements.

## Team Members

- Ali Mohiuddin - G01389462
- Nishad Main - G01384035
- Pranay Sharma - G01373961
- Radhika Bilolikar - G01356549