

GROUP MEMBERS:Sushmitha Konduru G01456225
RamyaSri Bhattaram G01443966
Chandana Konda G01449289
Vaishnav sai vivek varma yarakaraju G01464960

DOCUMENTATION

Introduction

This application is a Spring Boot-based CRUD (Create, Read, Update, Delete) system for managing student data. It exposes RESTful APIs to handle operations on a Student entity, stores data in a database, and integrates with an Angular front-end running on <http://localhost:4200>.

Features

Create: Add a new student.

Read:Fetch all students.

Update: Modify details of an existing student.

Delete: Remove a student record.

Components

Entity Layer (Student):

Represents the Student entity with fields such as name, address, email, preferences, and other survey details.

Annotated with JPA annotations to map the object to the database table.

Repository Layer (StudentRepository):

Extends JpaRepository to provide built-in CRUD operations for the Student entity.

Service Layer (StudentService):

Handles business logic for CRUD operations.

Includes validation and transformation of data before interacting with the database.

Controller Layer (StudentController):

Exposes RESTful endpoints for each CRUD operation.

Handles HTTP requests and responses for the API.

How It Works

Back-End Setup:

GROUP MEMBERS:Sushmitha Konduru G01456225
RamyaSri Bhattaram G01443966
Chandana Konda G01449289
Vaishnav sai vivek varma yarakaraju G01464960

Implemented a Student class annotated with JPA to define the database schema.
Created a StudentRepository extending JpaRepository to abstract database operations.
Built StudentService to encapsulate the business logic for CRUD operations and validate the input data.
Designed StudentController with RESTful endpoints to handle API requests from the front-end.

Front-End Integration:

Enabled @CrossOrigin(origins = "http://localhost:4200") in the controller and service layers to allow requests from the Angular application.
Provided endpoints for the Angular front-end to send and receive data for survey submissions and display.

Validation:

Ensured email format, phone number, and zip code validity using regex in Angular's reactive forms.
Checked for the existence of a student before updating or deleting to avoid invalid operations.