

# TASK: Create CRUD Application Using Spring Boot

 Estimated Time: 2–3 Hours

 Objective: Build a basic REST API using Spring Boot that performs CRUD operations on a database table.

---

## Task Overview

You need to create a **Spring Boot backend application** that allows you to:

- Create records
- Read records
- Update records
- Delete records

This will be a **REST-based CRUD application** using Spring Boot and PostgreSQL (or H2/MySQL if preferred).

---

## Module to Implement

**Entity: Student**

### Fields

Field Name	Type
id	Long (Primary Key)
name	String
email	String
age	Integer

course      String

---

## Step-by-Step Tasks

---

### Task 1 Create Spring Boot Project

Create a Spring Boot project using:

- Spring Web
  - Spring Data JPA
  - PostgreSQL Driver (or H2 for local testing)
- 

### Task 2 — Create Entity Class

Create a `Student` entity:

- Annotate with `@Entity`
  - Use `@Id` and `@GeneratedValue`
  - Map fields to database columns
- 

### Task 3 — Create Repository

Create a repository interface:

```
StudentRepository extends JpaRepository<Student, Long>
```

This should handle basic DB operations automatically.

---

### Task 4 — Create Service Layer

Create `StudentService` with methods:

- `addStudent(Student student)`
  - `getAllStudents()`
  - `getStudentById(Long id)`
  - `updateStudent(Long id, Student student)`
  - `deleteStudent(Long id)`
- 

## Task 5 — Create REST Controller

Create `StudentController` with endpoints:

Method	Endpoint	Operation
POST	<code>/students</code>	Add student
GET	<code>/students</code>	Get all students
GET	<code>/students/{id}</code>	Get student by ID
PUT	<code>/students/{id}</code>	Update student
DELETE	<code>/students/{id}</code>	Delete student

Use:

- `@RestController`
  - `@RequestMapping`
  - `@PostMapping, @GetMapping, @PutMapping, @DeleteMapping`
- 

## Task 6 — Configure Database

Configure DB in `application.properties`:

- Database URL
- Username
- Password
- JPA properties

Ensure table is auto-created.

---

## Task 7 — Test APIs

Test all endpoints using:

- Postman

Verify:

- Data is saved
- Data is fetched correctly
- Update & delete works