# Student Grade System - Expanded Beginner's Guide

## 1. Introduction

This project is a simple CRUD (Create, Read, Update, Delete) application that manages students and their grades. It uses React for the frontend, Spring Boot for the backend, and H2 as the database.

CRUD means:
- Create: Add a new student
- Read: View student list
- Update: Edit student details
- Delete: Remove a student

React handles the user interface, Spring Boot handles the API logic, and H2 stores data.

## 2. Prerequisites

You need the following installed before starting:
- Java 8 (check with: java -version)
- Apache Maven (check with: mvn -v)
- Node.js and npm (check with: node -v and npm -v)
- IDE: IntelliJ/Eclipse (for backend) and VS Code (for frontend)
- Browser: Chrome/Edge recommended

[screenshot: checking versions of Java, Maven, Node.js in terminal]

## 3. Backend (Spring Boot)

The backend exposes REST APIs for CRUD operations.

Important files:
- Student.java (Entity): Defines student fields mapped to DB columns.
- StudentRepository.java: Extends JpaRepository for auto CRUD methods.
- StudentController.java: REST APIs to create, read, update, delete.
- application.properties: Configures H2 DB.

Example flow: React sends JSON → Spring Boot saves in H2 DB.

[screenshot: backend running in terminal with Tomcat started]

## 4. Frontend (React)

The frontend displays the UI for interacting with students.

Important files:
- App.js: Root component, loads StudentForm and StudentList.
- StudentForm.js: Input fields for name, email, course, grade → POST request.
- StudentList.js: Fetches and displays students → GET, PUT, DELETE requests.

- api.js: Configures Axios with backend base URL.

[screenshot: React UI showing student form and list]

## 5. Running the Project

Step 1: Start backend
cd backend
mvn spring-boot:run

You should see 'Tomcat started on port 8080'.
Check in browser: http://localhost:8080/api/students → []

[screenshot: backend started successfully]

Step 2: Start frontend
cd frontend
npm install
npm start

Browser opens http://localhost:3000.
Try adding a student → it should appear in the list.

[screenshot: adding a student in React form]
[screenshot: student displayed in list after save]

## 6. Testing with H2 Console

Spring Boot includes H2 console to view the DB.
- URL: http://localhost:8080/h2-console
- JDBC URL: jdbc:h2:mem:studentsdb

[screenshot: H2 console showing students table]

## 7. Common Errors & Fixes

- npm blocked in PowerShell → Run: Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
- 'PORT' not recognized → Use cross-env package or remove PORT=3000
- ECONNREFUSED → Backend not running, start Spring Boot first

[screenshot: example error message in terminal]

## 8. Next Steps

Enhancements you can try:
- Add pagination and search in frontend
- Add validation annotations in backend (@NotNull, @Email)
- Switch DB from H2 to MySQL/Postgres
- Add authentication with Spring Security
- Deploy to AWS or Heroku

[screenshot: optional deployment diagram]