

ASSIGNMENT TITLE: Student Registry

Objective:

Build a simple full-stack app to register and view students using:

- **NestJS** - backend
- **Next.js** - frontend
- **MySQL (XAMPP)** - database

Step-by-Step Instructions:

Part 1: Create the Database with XAMPP (MySQL)

1. Create the database:
 - Name the database: ***student_db***

Part 2: Setup the Backend (NestJS)

1. Prerequisite- Install **nodejs**
 - Download “**Windows Installer (.msi)**” with node version 22.16.0 (LTS) using the following link: <https://nodejs.org/en/download>
2. installs the NestJS CLI globally on your machine so you can use commands like **nest new**, **nest generate**
 - Use the following command : **\$ npm i -g @nestjs/cli**
3. Create a Nestjs project:
 - **\$ nest new project-name**
4. Install MySQL TypeORM packages in the project folder - used to connect your TypeScript/JavaScript code to a database
 - **\$ npm install @nestjs/typeorm typeorm mysql2**
5. Configure DB in **app.module.ts**
6. Create a **student** entity with attributes or properties of your choice, and then create endpoints for posting, retrieving, updating, and deleting student information in the database (**POST, GET, PUT, DELETE**)

Part 3: Frontend(NextJS)

1. Create a Next.js App:
 - Use this command: `$ npx create-next-app@latest`
2. Create four pages (or routes) for each functionality: posting, getting, editing and deleting students.

EXPECTING RESULTS:

- `student_db` with a `students` table
- Working NestJS backend on port 3000 with:
 - `POST /student` → creates student
 - `GET /student` → gets all students
 - `PUT /student` → update student information
 - `DELETE /student` → delete selected students
- Next.js frontend with:
 - `/post-student` page → form to post
 - `/get-students` page → display all
 - `/update-students` page → update student information
 - `/delete-students` page → delete selected students

Note:

Use the following documentations for help:

- <https://docs.nestjs.com/>
- <https://nextjs.org/docs>