Exercise 11

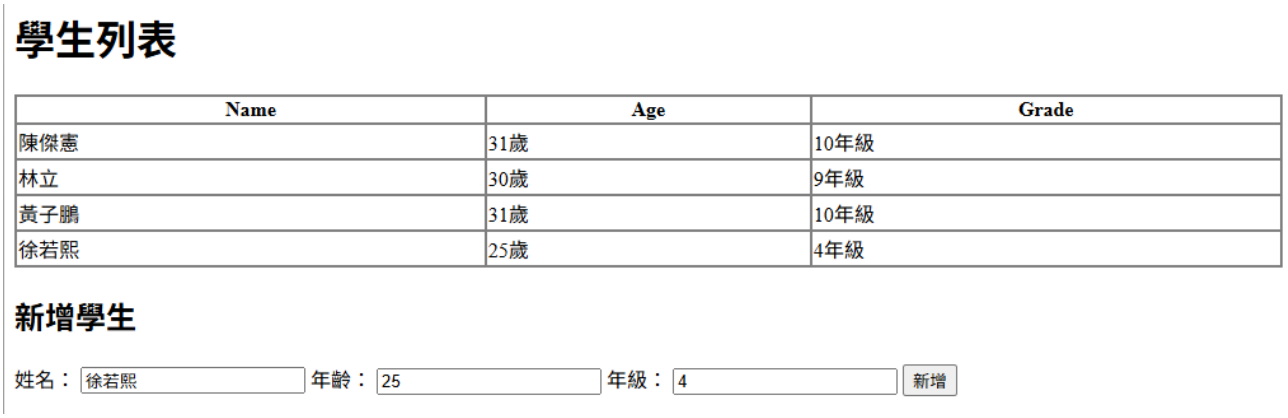
Student Management System

You have been given a simple student management system frontend, and your task is to implement the backend using **Express** and **Mongoose** to establish a RESTful API.

Frontend: You are provided with an HTML file (index.html) that includes:

- An area to display the list of students.
- A form for **adding new students**, including input fields for name, age, and grade.
- fetch is used to make API calls to interact with the backend.

Note that “index.html” should be placed in the "public" subdirectory for an Express project.



Backend:

- Please create a Node.js application and set up a backend server using the Express framework.
- Please connect to a MongoDB database using Mongoose for database operations.
- Please define a model for student data, including attributes such as **name (String), age (Number), and grade (String)**.
- Please implement two API endpoints:
  - GET /students: Return a list of all students.
  - POST /students: Accept data for a new student, add it to the database, and return the newly added student data.

Notes:

- You can assume that the frontend is correctly set up to make API requests to the backend.
- Please ensure that the local MongoDB server is running, and the backend should successfully connect to the database.
- Please zip your folder and submit it to TronClass.

Additional Challenge (Bonus):

- Implement a DELETE /students/:id API endpoint to delete a specific student.
- Implement a PUT /students/:id API endpoint to update a specific student.
- Change the local MongoDB to Atlas.
- Deploy the Web app to Render (render.com).