Exercise: Creating a Student Information System

Objective: To create a RESTful API that can perform CRUD (Create, Read, Update, Delete) operations on a student database.

Tools Required: Node.js, Express.js, MongoDB, Mongoose

Tasks:

Set up the project:

Create a new Node.js project, installing the necessary dependencies (express, mongoose, etc.).

Set up your Express app and create your server file (usually app.js or server.js).

Model the Data:

Define a Student Mongoose model with the following properties: name (String), age (Number), major (String), createdDate, updatedDate.

Create the Database Connection:

Use Mongoose to connect to a MongoDB database (this can be locally hosted or in the cloud using MongoDB Atlas).

Create CRUD Routes:

Set up the following Express routes:

GET /students: Fetch all students

GET /students/:id: Fetch a single student by id

POST /students: Add a new student

PUT /students/:id: Update a student by id

DELETE /students/:id: Delete a student by id

Implement Route Handlers:

Implement the logic for each route. Use async/await and try/catch for error handling.

The GET routes should return the requested student(s).

The POST route should add a new student to the database and return the created student data.

The PUT route should update an existing student's data and return the updated data.

The DELETE route should remove a student from the database and return a confirmation message.

Testing:

Use a tool like Postman to test all your routes and ensure they are working as expected.

Deliverable: Submit the link to your Github repository with the complete project code, screen shot and video explaining your code. The repository should include all source code files, a README.md file with instructions on how to run the project, and any other necessary documentation.

TIPS: YOU CAN USE TODOs PROJECT AS REFERENCE