



ASM Essentials of NodeJS - Exam Paper

Duration: 100 minutes | Marks: 15

Question:

- Use Node.js to build a simple RESTful API and a basic user interface.
- The application must include at least 4 endpoints and perform full CRUD operations (Create, Read, Update, Delete).
- Connect to a MongoDB or MySQL or Firebase database to store data.
- Use Postman or a similar tool to test the API endpoints.
- Design a simple user interface (UI/UX) for the main functions.

Problem Statement:

Build an application to manage student information with the following requirements:

1. Student Structure:

- id: Auto-increment number (or UUID)
- name: Student's name (string, required)
- age: Student's age (number, greater than 0)
- gender: Student's gender (string, accepts only 'Male' or 'Female')
- major: Student's major (string, required)

2. Required Functions:(9 marks)

- Add a new student (POST /students)
 - This endpoint receives student information from the user (name, age, gender, major).
 - Validate the input data. If there is an error, return an appropriate error message.
 - Save the new student to the database and return the newly created student's information.
- Get list of students (GET /students)
 - Return a list of all students in the database.
 - Students can be filtered by major by passing a major parameter in the URL.
- Update student information (PUT /students/:id)
 - Update the student information by id including: name, age, gender, and major.
 - If the student does not exist, return an error message.
 - If the update is successful, return the updated student's information.
- Delete student (DELETE /students/:id)
 - Delete the student by id.
 - If the student does not exist, return an error message.
 - If deletion is successful, return a confirmation message.

3. UI/UX Requirements:(6 marks)

- Design a simple user interface with the following functionalities:
 - Student List Page: Display a list of all students in the system. Each student should display name, age, gender, and major. **(1 marks)**
 - Add New Student Form: Include fields for name, age, gender, major. When the user fills in all the information and clicks the "Add Student" button, the student should be added to the system.**(2 marks)**
 - Edit Student: Allow the user to select a student from the list and update that student's details.**(1 marks)**
 - Delete Student: Add a "Delete" button next to each student in the list. When the user clicks the "Delete" button, the student should be removed from the system.**(1 mark)**
- UX Requirements:**(1 mark)**
 - CRUD operations must be performed through the user interface.
 - Provide feedback to the user with notifications (toast messages) when a student is created, updated, or deleted successfully.
 - Use a CSS library or framework like Bootstrap to create a friendly and visually appealing interface.
 - UI components should be clearly organized with appropriate spacing.

4. Additional Requirements:

- Should be use Express.js as the main framework.
- Use MongoDB or MySQL or Firebase as the database.
- You can use libraries like mongoose for MongoDB or sequelize for MySQL.
- There must be clear error messages and proper error handling (try-catch).
- The directory structure should be clear and modular.