

1. Model (models/studentModel.js)

Use an array to simulate a database.

```
// In-memory data
let students = [
  { id: 1, name: "Alice", age: 20, course: "BSIT" },
  { id: 2, name: "Bob", age: 21, course: "BSCS" }
];

class StudentModel {
  static findAll() {
    return students;
  }

  static findById(id) {
    return students.find(student => student.id === id);
  }

  static create(newStudent) {
    newStudent.id = students.length + 1;
    students.push(newStudent);
    return newStudent;
  }

  static update(id, updatedData) {
    const index = students.findIndex(s => s.id === id);
    if (index !== -1) {
      students[index] = { ...students[index], ...updatedData };
      return students[index];
    }
    return null;
  }

  static delete(id) {
    const index = students.findIndex(s => s.id === id);
    if (index !== -1) {
      return students.splice(index, 1);
    }
    return null;
  }
}

module.exports = StudentModel;
```

2. Controller (controllers/studentController.js)

```
const StudentModel = require("../models/studentModel");

// GET all students
exports.getAllStudents = (req, res) => {
  res.json(StudentModel.findAll());
};

// GET student by ID
exports.getStudentById = (req, res) => {
  const student = StudentModel.findById(Number(req.params.id));
  if (student) {
    res.json(student);
  } else {
    res.status(404).json({ error: "Student not found" });
  }
};

// POST create new student
exports.createStudent = (req, res) => {
  const newStudent = StudentModel.create(req.body);
  res.status(201).json(newStudent);
};

// PUT update student
exports.updateStudent = (req, res) => {
  const updated = StudentModel.update(Number(req.params.id), req.body);
  if (updated) {
    res.json(updated);
  } else {
    res.status(404).json({ error: "Student not found" });
  }
};

// DELETE student
exports.deleteStudent = (req, res) => {
  const deleted = StudentModel.delete(Number(req.params.id));
  if (deleted) {
    res.json({ message: "Student deleted successfully" });
  } else {
    res.status(404).json({ error: "Student not found" });
  }
};
```

3. Routes (routes/studentRoutes.js)

```
const express = require("express");
const router = express.Router();
const studentController = require("../controllers/studentController");

router.get("/", studentController.getAllStudents);
router.get("/:id", studentController.getStudentById);
router.post("/", studentController.createStudent);
router.put("/:id", studentController.updateStudent);
router.delete("/:id", studentController.deleteStudent);

module.exports = router;
```

4. Main App (app.js)

```
const express = require("express");
const studentRoutes = require("./routes/studentRoutes");

const app = express();
app.use(express.json());

app.use("/students", studentRoutes);

app.listen(3000, () => {
  console.log("Server running at http://localhost:3000");
});
```

In-Class Activity

- Create a model that holds bio/hobbies/skills in arrays.
- Create a controller that fetches data from the model.
- Use routes to map endpoints to controllers.

Improve the project to use EJS and forms for creating new students, display list of students, updating etc.