Name:Himanshu Dhomane

Roll no:47

RegNo:21070297

Practical-6

Aim: **Develop an express web application that can interact with REST API to**

**perform CRUD operations on student data. (Use postman).**

**Theory:**

**Step 1: Set up your Express project**

1. Install Node.js if you haven't already.

2. Create a new directory for your project.

3. Initialize a new Node.js project using `npm init`.

4. Install Express using `npm install express`.

5. Create your main server file, e.g., `server.js`.

**Step 2: Create your REST API endpoints**

You'll create endpoints for CRUD operations:

- Create: POST `/api/students`

- Read: GET `/api/students`

- Update: PUT `/api/students/:id`

- Delete: DELETE `/api/students/:id`

**Step 3: Implement CRUD operations**

// index.js

const express = require('express');

const fs = require('fs').promises;

const bodyParser = require('body-parser');

const path = require('path');

const app = express();

const PORT = process.env.PORT || 3000;

const dataFilePath = path.join(\_\_dirname, 'data.json');

// Middleware

app.use(bodyParser.json());

// Helper functions

const readDataFromFile = async () => {

try {

const data = await fs.readFile(dataFilePath, 'utf8');

return JSON.parse(data);

} catch (error) {

return [];

}

};

const writeDataToFile = async (data) => {

await fs.writeFile(dataFilePath, JSON.stringify(data, null, 2));

};

// Routes

app.get('/todos', async (req, res) => {

try {

const todos = await readDataFromFile();

res.json(todos);

} catch (error) {

res.status(500).json({ error: 'Internal Server Error' });

}

});

app.post('/todos', async (req, res) => {

const { task, completed } = req.body;

try {

const todos = await readDataFromFile();

const newTodo = { id: Date.now(), task, completed };

todos.push(newTodo);

await writeDataToFile(todos);

res.status(201).json(newTodo);

} catch (error) {

res.status(400).json({ error: 'Bad Request' });

}

});

app.put('/todos/:id', async (req, res) => {

const { id } = req.params;

const { task, completed } = req.body;

try {

let todos = await readDataFromFile();

const index = todos.findIndex(todo => todo.id.toString() === id);

if (index !== -1) {

todos[index] = { id: parseInt(id), task, completed };

await writeDataToFile(todos);

res.json(todos[index]);

} else {

res.status(404).json({ error: 'Todo not found' });

}

} catch (error) {

res.status(400).json({ error: 'Bad Request' });

}

});

app.delete('/todos/:id', async (req, res) => {

const { id } = req.params;

try {

let todos = await readDataFromFile();

const index = todos.findIndex(todo => todo.id.toString() === id);

if (index !== -1) {

todos = todos.filter(todo => todo.id.toString() !== id);

await writeDataToFile(todos);

res.sendStatus(204);

} else {

res.status(404).json({ error: 'Todo not found' });

}

} catch (error) {

res.status(500).json({ error: 'Internal Server Error' });

}

});

// Start the server

app.listen(PORT, () => {

console.log(Server is running on port ${PORT});

});tion, etc.

Screenshot: