**Program 6**

**ExpressJS – Database, RESTful APIs**

a Write a program to connect MongoDB database using Mangoose and perform CRUD operations.

**Step 1: Install Required Packages**

Run in terminal:

1. node -v
2. npm -v
3. npm install express mongoose
4. npm list mongoose
5. **mongod –version**
6. npm init -y
7. node 6a1.js

**Step 2: 6a1.js**

const express = require('express');

const mongoose = require('mongoose');

const app = express();

app.use(express.json()); // for JSON data

// 1️⃣ Connect to MongoDB

mongoose.connect('mongodb://127.0.0.1:27017/mydb')

.then(() => console.log('✅ MongoDB Connected'))

.catch(err => console.log(err));

// 2️⃣ Create Schema & Model

const UserSchema = new mongoose.Schema({

name: String,

email: String

});

const User = mongoose.model('User', UserSchema);

// 3️⃣ CRUD Routes

// CREATE (POST)

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

const user = new User(req.body);

await user.save();

res.send(user);

});

// READ (GET)

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

const users = await User.find();

res.send(users);

});

// UPDATE (PUT)

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

const user = await User.findByIdAndUpdate(req.params.id, req.body, { new: true });

res.send(user);

});

// DELETE (DELETE)

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

await User.findByIdAndDelete(req.params.id);

res.send({ message: 'User deleted' });

});

// 4️⃣ Start Server

app.listen(3000, () => console.log('🚀 Server running on port 3000'));

**Step 3: Run the Project**

**node 6a1.js**

✅ You can test the API using **Postman**

**Create User: POST** → [**http://localhost:3000/users**](http://localhost:3000/users)

JSON Body: { "name": "John", "email": "john@test.com" }

* **Get Users:** GET → [**http://localhost:3000/users**](http://localhost:3000/users)
* **Update User:** PUT → [**http://localhost:3000/users/<id**](http://localhost:3000/users/%3cid)**>**
* **Delete User:** DELETE → **[http://localhost:3000/users/<id](http://localhost:3000/users/%3cid)>**

**Explanation:**

**Step 1: Create a User (POST)**

**Request:**

POST

**http://localhost:3000/users**

Body (JSON):

**{**

**"name": "Alice",**

**"email": "alice@test.com"**

**}**

**Response Example:**

{

"\_id": "66a90f4a8c1b3f18f92c1a23",

"name": "Alice",

"email": "alice@test.com",

"\_\_v": 0

}

**Step 2: Update the User (PUT)**

Use the \_id from the response above in your **PUT request**:

PUT http://localhost:3000/users/66a90f4a8c1b3f18f92c1a23

**Body (JSON):**

**{**

**"name": "Alice Updated",**

**"email": "alice.new@test.com"**

**}**

**Step 3: Check Update**

Then you can **GET** all users to confirm:

GET http://localhost:3000/users

**b. Write a program to develop a single page application using RESTful APIs.**

very simple and small ExpressJS SPA program using RESTful APIs with a mock in-memory database (array). This example will:

* Serve a single HTML page (SPA)
* Provide RESTful API endpoints (GET and POST)
* Allow adding and displaying items dynamically

**1. Install dependencies**

npm init -y

npm install express body-parser

**2. 6b.js**

const express = require('express');

const app = express();

app.use(express.json());

app.use(express.static('public'));

let items = ["Apple", "Banana"];

app.get('/api/items', (req, res) => res.json(items));

app.post('/api/items', (req, res) => { items.push(req.body.name); res.json(items); });

app.listen(3000, () => console.log('http://localhost:3000'));

**3. public/index.html (Single Page App)**

Create a folder named **public** in your own created folder and save this file as index.html in public folder

<!DOCTYPE html>

<html>

<body>

<h2>Items</h2>

<ul id="list"></ul>

<input id="item" placeholder="New item">

<button onclick="addItem()">Add</button>

<script>

async function load(){

const res = await fetch('/api/items');

document.getElementById('list').innerHTML = (await res.json()).map(i=>`<li>${i}</li>`).join('');

}

async function addItem(){

await fetch('/api/items',{method:'POST',headers:{'Content-Type':'application/json'},body:JSON.stringify({name:document.getElementById('item').value})});

document.getElementById('item').value='';

load();

}

load();

</script>

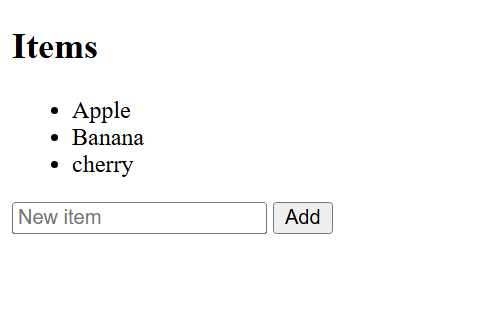
</body>

</html>

**Run**

**node 6b.js**

**Output:**

****