**1. Install dependencies**

bash

CopyEdit

npm init -y

npm install express mongodb body-parser

**2. Basic folder structure**

lua

CopyEdit

project/

│-- index.js

│-- package.json

**3. index.js — Express + MongoDB CRUD**

javascript

CopyEdit

const express = require('express');

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

const { MongoClient, ObjectId } = require('mongodb');

const app = express();

app.use(bodyParser.json());

// MongoDB connection URI

const uri = "mongodb://localhost:27017"; // Change to your MongoDB URL

const client = new MongoClient(uri);

// Database and collection

let db, usersCollection;

async function connectDB() {

try {

await client.connect();

db = client.db('testdb');

usersCollection = db.collection('users');

console.log("✅ Connected to MongoDB");

} catch (error) {

console.error("❌ DB connection failed", error);

}

}

connectDB();

// CREATE - Add new user

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

try {

const result = await usersCollection.insertOne(req.body);

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

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// READ - Get all users

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

try {

const users = await usersCollection.find().toArray();

res.json(users);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// READ - Get single user by ID

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

try {

const user = await usersCollection.findOne({ \_id: new ObjectId(req.params.id) });

if (!user) return res.status(404).json({ message: "User not found" });

res.json(user);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// UPDATE - Update user by ID

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

try {

const result = await usersCollection.updateOne(

{ \_id: new ObjectId(req.params.id) },

{ $set: req.body }

);

res.json(result);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// DELETE - Remove user by ID

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

try {

const result = await usersCollection.deleteOne({ \_id: new ObjectId(req.params.id) });

res.json(result);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Start server

const PORT = 3000;

app.listen(PORT, () => {

console.log(`🚀 Server running on http://localhost:${PORT}`);

});

**4. Test the API**

You can test using **Postman** or curl.