const express = require('express');

const mongoose = require('mongoose');

const prompt = require('prompt-sync')();

const app = express();

const port = 3000;

// Connect to MongoDB

mongoose.connect('mongodb+srv://Niran\_jana\_2604:12345@cluster0.k9vdgjh.mongodb.net/day?retryWrites=true&w=majority' {

useNewUrlParser: true,

useUnifiedTopology: true,

});

// Define MongoDB schema and model

const legSchema = new mongoose.Schema({

source: String,

destination: String,

cost: Number,

});

const Leg = mongoose.model('Leg', legSchema);

class Route {

constructor() {

this.legs = [];

}

addLeg(source, destination, cost) {

const leg = new Leg({ source, destination, cost });

this.legs.push(leg);

}

getTotalCost() {

return this.legs.reduce((totalCost, leg) => totalCost + leg.cost, 0);

}

async save() {

await Promise.all(this.legs.map((leg) => leg.save()));

}

}

app.use(express.json());

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

const { source, destination, cost } = req.body;

const leg = new Leg({ source, destination, cost });

await leg.save();

res.sendStatus(201);

});

app.get('/total-cost', async (req, res) => {

const legs = await Leg.find();

const totalCost = legs.reduce((total, leg) => total + leg.cost, 0);

res.json({ totalCost });

});

app.listen(port, () => {

console.log(`Server listening on port ${port}`);

});

async function main() {

const route = new Route();

const numLegs = prompt('Enter the number of legs in the route:');

for (let i = 1; i <= numLegs; i++) {

const source = prompt(`Enter the source city for leg ${i}:`);

const destination = prompt(`Enter the destination city for leg ${i}:`);

const cost = parseFloat(prompt(`Enter the cost for leg ${i}:`));

route.addLeg(source, destination, cost);

}

const totalCost = route.getTotalCost();

console.log('Total cost of the trip:', totalCost);

await route.save();

console.log('Route saved to MongoDB');

}

main();