**Classroom Activity: Implement Error handling.**

* **Update index.js for error handling.**

const express = require('express');

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

const app = express();

const port = 3000;

// Middleware

app.use(bodyParser.json());

// In-memory data store

let restaurants = [

  { id: 1, name: 'Restaurant One', location: 'Location One', cuisine: 'Cuisine One' },

  { id: 2, name: 'Restaurant Two', location: 'Location Two', cuisine: 'Cuisine Two' }

];

// Routes

// GET route to retrieve all restaurants

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

  res.json(restaurants);

});

// POST route to add a new restaurant

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

  const newRestaurant = req.body;

  newRestaurant.id = restaurants.length ? restaurants[restaurants.length - 1].id + 1 : 1;

  restaurants.push(newRestaurant);

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

});

// PUT route to update an existing restaurant by ID

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

  const restaurantId = parseInt(req.params.id, 10);

  const updatedRestaurant = req.body;

  let restaurant = restaurants.find(r => r.id === restaurantId);

  if (restaurant) {

    restaurant = { ...restaurant, ...updatedRestaurant };

    restaurants = restaurants.map(r => (r.id === restaurantId ? restaurant : r));

    res.json(restaurant);

  } else {

    res.status(404).json({ message: 'Restaurant not found' });

  }

});

// DELETE route to delete a restaurant by ID

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

  const restaurantId = parseInt(req.params.id, 10);

  const restaurantIndex = restaurants.findIndex(r => r.id === restaurantId);

  if (restaurantIndex !== -1) {

    restaurants.splice(restaurantIndex, 1);

    res.status(204).end();

  } else {

    res.status(404).json({ message: 'Restaurant not found' });

  }

});

// Start the server

app.listen(port, () => {

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

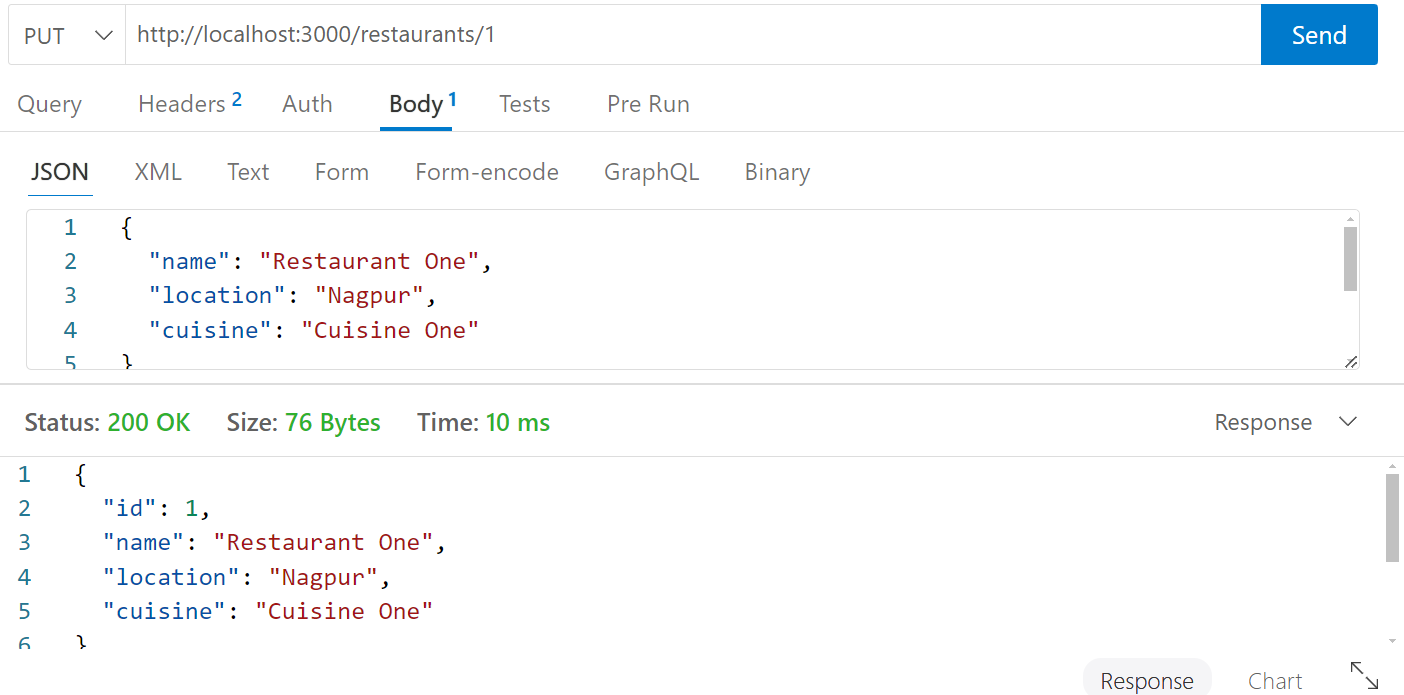
});

* **Start the server using Nodemon:**

npm run dev

* **Output**

Restaurant with id =1 exists so it will update it



If restaurant is not found it will give error message

