



AUG-
DEC

Unit 4: NODE JS & Mongo DB

2022

ASSIGNMENT

Name: <u>Tejas Venugopalan</u>	SRN: PES1UG22AM177	Section: C
	Date: 04-12-2023	Unit 4 Assignment Exercise
<u>PROBLEM STATEMENT1 (for odd number SRNs):</u>		

Create a Node.js application that interacts with a MongoDB database named Bangalore_City. This database contains a collection of restaurants, and each restaurant document should include the following entities:

rest_name: Name of the restaurant. rest_id:
Unique identifier for the restaurant. rest_addr:
Address of the restaurant. rest_reviews: Grade
assigned to the restaurant. Food_menu: Menu
items offered by the restaurant.

Implement CRUD operations for the "restaurants" collection:

1. Create a new restaurant:
Design a route that allows the addition of a new restaurant to the collection. Users should be able to provide details such as rest_name, rest_addr, rest_grade, rest_reviews, and Food_menu.
2. Read all restaurants:
Create a route that retrieves and displays all restaurants in the collection.
3. Update a restaurant's grade:
Design a route that allows users to update the rest_grade of a specific restaurant based on the rest_id.

OBJECTIVE

The objective of this exercise is to test the student on back end frame work and storage Node JS with Mongo DB. It evaluates the student's knowledge Node Js App, modules, Node Js ,HTTP modules, Reading and writing to Mongo DB through Node Js.

PREREQUISITE

In order to complete this exercise, the student needs to understand the fundamentals of JavaScript, Mongo DB Operations with Nodejs modules.



AUG-
DEC



AUG-
DEC

Unit 4: NODE JS & Mongo DB

2022

PROGRAM

```
JS index.js X index.html
JS index.js > ...
1  const express = require('express');
2  const mongoose = require('mongoose');
3  const bodyParser = require('body-parser');
4
5  const app = express();
6  const PORT = process.env.PORT || 3000;
7
8  // Connect to MongoDB
9  mongoose.connect('mongodb://localhost/Bangalore_City', {
10    useNewUrlParser: true,
11    useUnifiedTopology: true,
12  });
13
14  // Create a schema for the restaurant
15  const restaurantSchema = new mongoose.Schema({
16    rest_name: String,
17    rest_id: { type: String, unique: true },
18    rest_addr: String,
19    rest_reviews: Number,
20    food_menu: [String],
21  });
22
23  // Create a model based on the schema
24  const Restaurant = mongoose.model('Restaurant', restaurantSchema);
25
26  // Middleware to parse JSON requests
27  app.use(bodyParser.json());
28
29  // Create a new restaurant
30  app.post('/restaurants', async (req, res) => {
31    try {
32      const { rest_name, rest_id, rest_addr, rest_reviews, food_menu } = req.body;
33
34      const newRestaurant = await Restaurant.create({
35        rest_name,
36        rest_id,
37        rest_addr,
38        rest_reviews,
39        food_menu,
40      });
```



AUG-
DEC

Unit 4: NODE JS & Mongo DB

2022

Unit 4: NODE JS & Mongo DB

```
ex.js > ...
  try {
    const { rest_name, rest_id, rest_addr, rest_reviews, food_menu } = req.body;

    const newRestaurant = await Restaurant.create({
      rest_name,
      rest_id,
      rest_addr,
      rest_reviews,
      food_menu,
    });

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

// Get all restaurants
app.get('/restaurants', async (req, res) => {
  try {
    const restaurants = await Restaurant.find();
    res.json(restaurants);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

// ... (Other routes for updating and deleting)

// Serve HTML file
app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});

// Start the server
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:\${PORT}`);
});
```



AUG-
DEC

Unit 4: NODE JS & Mongo DB

2022

Unit 4: NODE JS & Mongo DB

TERMINAL EXECUTION FOR RUNNING SERVER...

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\User\Documents\WT_ASSIGNMENT> node index.js
(node:5324) [MONGODB DRIVER] Warning: useUrlParser is a deprecated option: useUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:5324) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
PS C:\Users\User\Documents\WT_ASSIGNMENT> node index.js
(node:10208) [MONGODB DRIVER] Warning: useUrlParser is a deprecated option: useUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:10208) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
PS C:\Users\User\Documents\WT_ASSIGNMENT> node index.js
(node:7972) [MONGODB DRIVER] Warning: useUrlParser is a deprecated option: useUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:7972) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
```



AUG-
DEC

Unit 4: NODE JS & Mongo DB

2022

Unit 4: NODE JS & Mongo DB

SCREENSHOT OF YOUR OUTPUT

Restaurant Details

Restaurant Name: Restaurant ID: Restaurant Address: Restaurant Reviews: Food Menu (comma-separated):

Restaurant ID to Update: Updated Restaurant Name: Updated Restaurant Address: Updated Restaurant Reviews:
 Updated Food Menu (comma-separated):

All Restaurants:

- THE GOURMET KITCHEN - 123 Main Street, Cityville,BANGLORE - Reviews: 25 - Menu: VEG PIZZA, NON - VEG PIZZA, AND OTHERS

Unit 4: NODE JS & Mongo DB

MongoDB Compass - localhost:27017/Bangalore_City.restaurants

Connect Edit View Collection Help

localhost:27017 Documents Bangalore_City.r...

My Queries Databases Search

Bangalore_City restaurants

LoginFormPractice admin config local sk

Bangalore_City.restaurants 0 DOCUMENTS 2 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query Explain Reset Find Options

ADD DATA EXPORT DATA 1-1 of 1

```
{
  "_id": ObjectId("655ccd01694c2f45fdfcd62e"),
  "rest_name": "THE GOURMET KITCHEN",
  "rest_id": "789456",
  "rest_addr": "123 Main Street, Cityville,BANGLORE",
  "rest_reviews": 25,
  "food_menu": Array (3)
  __v: 0
}
```

OUTPUT IN DATABASE..



AUG-
DEC

Unit 4: NODE JS & Mongo DB

2022