|  |  |  |
| --- | --- | --- |
| **Name: Ankush H V** | **SRN: PES1UG21CS091** | **Section: B** |
| **Date:29 Nov 2022** | **Unit 4 Assignment Exercise** |
| PROBLEM STATEMENT1 (for odd number SRNs): | | |
| 1. Create a MongoDB database **Bangalore\_City** that has a collection of **Companies** having different documents (such as comp\_name, comp\_rank, comp\_website,comp\_location,employee\_size, comp\_revenue  etc)for each company. Create a server listening to 8081. Use the POST method to insert document in to the database. Query the database to retrieve the newly inserted document. | | |
| 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. | | |
| SAMPLE SCREENSHOT OF OUTPUT (Just for your reference) | | |
| SAMPLE1: | | |
| PROGRAM | | |
| **Server.js:**  var http = require('http');  var MongoClient = require('mongodb').MongoClient;  var mongoUrl = "mongodb://localhost:27017";  http.createServer(function(req, res) {  console.log("The request type is: " + req.method);  if(req.method == "GET") {  // let company = req.body;  MongoClient.connect(mongoUrl, function(err, db) {  if (err) throw err;  var citydb = db.db("Bangalore\_City");  citydb.collection("Companies").find({}).toArray(function(err, result) {  if (err) throw err;  console.log(result);  res.writeHead(200, { 'Content-Type': 'application/json' });  //write the content of the file to response body  res.write(JSON.stringify(result));  db.close();  res.end();  });  });  }  else { // method is POST  // console.log(req.body);  req.on('data', function(data) {  console.log("The data is: " + data);  company = JSON.parse(data);  });  req.on('end', function() {  try {  MongoClient.connect(mongoUrl, function(err, db) {  if (err) throw err;  var citydb = db.db("Bangalore\_City");  citydb.collection("Companies").insertMany(company, function(err, response) {  if (err) throw err;  console.log(response.insertedCount + " documents inserted");  db.close();  res.end("Data inserted:\n" + company);  // res.end("\nMessage: " + JSON.stringify(response));  });  });  } catch (error) {  res.end("Error: " + error);  }  });  }  }).listen(8081, function() {  console.log("Server is listening on port 8081");  });  **Client.js:**  var http = require('http');  var fetch = require('node-fetch');  var url = require('url');  // gettting the data from the server  fetch('http://localhost:8081/api', {  method: 'GET',  })  .then(res => res.json())  .then(res => {console.log("Response received on GET\n")  console.table(res)});  // posting the data to the server  fetch('http://localhost:8081/api', {  method: 'POST',  body: JSON.stringify([{  "compName": "IBM",  "compRank": 2,  "compWebsite": "www.ibm.com",  "compAddress": "Nagavara, Bangalore",  "empCount": 10000,  "cmpRevenue": 1000000000},  {  "compName": "TCS",  "compRank": 3,  "compWebsite": "www.tcs.com",  "compAddress": "Nagavara, Bangalore",  "empCount": 10000,  "cmpRevenue": 1000000000  },{  "compName": "Wipro",  "compRank": 4,  "compWebsite": "www.wipro.com",  "compAddress": "Nagavara, Bangalore",  "empCount": 10000,  "cmpRevenue": 1000000000  },{  "compName": "Accenture",  "compRank": 5,  "compWebsite": "www.accenture.com",  "compAddress": "Nagavara, Bangalore",  "empCount": 10000,  "cmpRevenue": 1000000000  }]),  headers: { 'Content-Type': 'application/json' },  })  .then(res => {console.log("\nResponse Received on POST:\n")  console.log(res);}); | | |
| SCREENSHOT OF YOUR OUTPUT | | |
|  | | |

