ASSIGNMENT 1

CS304 - MODERN DEVELOPMENT LAB

Done By:

2004 - NARAYAN BANDODKER

2054 - VIBHAV SINAI PISSURLENKAR

SEMESTER 5 - BATCH (2020-23)

Node Js - MongoDb

Connection :

1) Installation of db package : npm install mongodb
2) Database creation

```
var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/mydb";

MongoClient.connect(url, function(err, db) {

 if (err) throw err;

 console.log("Database created!");

 db.close();

});
```

3) Mongodb insertion

```
var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";


MongoClient.connect(url, function(err, db) {

 if (err) throw err;

 var dbo = db.db("mydb");

 var myobj = [
```

```
        { name: 'John', address: 'Highway 71'},
        { name: 'Peter', address: 'Lowstreet 4'},
        { name: 'Amy', address: 'Apple st 652'},   ];
dbo.collection("customers").insertMany(myobj, function(err, res) {
        if (err) throw err;
        console.log("Number of documents inserted: " + res.insertedCount);
        db.close();
});
});
```

Combining NodeJS and NoSQL- Why MongoDB is the best choice?

1. Node.js is popularly being used in web applications because it lets the application run while it is fetching data from the backend server. It is asynchronous, event-driven and helps to build scalable web applications. Even though Node.js works well with a MySQL database, the perfect combination is a NoSQL like MongoDB wherein the schema need not be well-structured. MongoDB represents the data as a collection of documents rather than tables related by foreign keys. This makes it possible for the varied types of data dealt over the internet to be stored decently and accessed in the web applications using Node.js.
2. MongoDB is a distributed database which allows ad-hoc queries, real-time integration, and indexing efficiently.
3. MongoDB allows geospatial queries and is text-search enabled.

Reference :

https://www.w3schools.com/nodejs