Dynamic website using Node JS and Express

Name:	M Badri Narayanan
Reg No:	185002018
Semester:	VI
Exercise Number:	9
Date:	April 13, 2021

Aim

To create a dynamic backend application using NodeJS, Express and MongoDB using javascript.

Procedure

- Initially, we have to create a Mongo Database.To create a database in MongoDB using javascript, start by creating a MongoClient object, then specify a connection URL with the correct IP address and the name of the database you want to create.
- MongoDB will create the database if it does not exist, and make a connection to it.
- Then create a collection using the **createCollection()** method.
- To insert a record, or document as it is called in MongoDB, into a collection, we use the **insertOne()** method.
- To insert multiple records use **insertMany()** method.The first parameter of the **insertMany()** method is an array of objects, containing the data you want to insert.
- Sort() method can be used along **toArray()** function to sort the records based on a parameter as well as view the records.
- To delete a record, or document as it is called in MongoDB, we use the **deleteOne()** method.
- The first parameter of the **deleteOne()** method is a query object defining which document to delete.
- To update a record, or document as it is called in MongoDB, we use the **updateOne()** method.
- The first parameter of the **updateOne()** method is a query object defining which document to update.
- To get the data from the user we use the **prompt()** command.

Functions Used

```
const prompt = require('prompt');

prompt.start();

prompt.get([Collection_Schema_1, Collection_Schema_2], function (err, result)
{

});

MongoClient.connect("Database URL", {useUnifiedTopology: true}, function(err, db)
{

});

dbo.createCollection("Collection Name", function(err, res)
{

});


var dbo = db.db("Database Name");
dbo.collection("Collection Name").drop(function(err, delOK)
{

});

var dbo = db.db("Database Name");
var myobj = {Collection_Schema_1: Variable_To_Be_Inserted_1, Collection_Schema_2:
↪  Variable_To_Be_Inserted_2};
dbo.collection("Collection Name").insertOne(myobj, function(err, res)
{


});

var dbo = db.db("Database Name");
var myquery = {Collection_Schema: Variable_To_Be_Deleted};
dbo.collection("Collection Name").deleteOne(myquery, function(err, obj)
{

});

var dbo = db.db("Database Name");
dbo.collection("Collection Name").find({}).toArray(function(err, result)
{

});

var dbo = db.db("Database Name");
var myquery = {Collection_Schema : Variable_To_Be_Updated};
var newvalues = { $set: {Collection_Schema_1: New_Variable_1, Collection_Schema_2 :
↪  New_Variable_2}};
dbo.collection("Collection Name").updateOne(myquery, newvalues, function(err, res)
{

});
```

Create Database Code

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/webprogramming";

MongoClient.connect(url, {useUnifiedTopology: true}, function(err, db)
{
  if (err) throw err;
  console.log("Database created!");
  db.close();
});
```

Create Collection Code

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, {useUnifiedTopology: true}, function(err, db)
{
  if (err) throw err;
  var dbo = db.db("webprogramming");
  dbo.createCollection("webprogramming", function(err, res)
  {
    if (err) throw err;
    console.log("Collection created!");
    db.close();
  });
});
```

Insertion Code

```
const prompt = require('prompt');

prompt.start();

prompt.get(['name', 'phonenumber'], function (err, result)
{
    if (err) { return onErr(err); }
    console.log('\n Data to be inserted \n');
    console.log('  Name: ' + result.name);
    console.log('  Phone Number: ' + result.phonenumber);
    name = result.name;
    phonenumber = result.phonenumber;
    var MongoClient = require('mongodb').MongoClient;
        var url = "mongodb://localhost:27017/";
        MongoClient.connect(url, {useUnifiedTopology: true}, function(err, db)
        {
                  if (err) throw err;
                  var dbo = db.db("webprogramming");
                  var myobj = { name: name, phonenumber: phonenumber};
                  dbo.collection("webprogramming").insertOne(myobj, function(err, res)
                  {
                    if (err) throw err;
                    console.log("1 document inserted");
                    db.close();
                  });
```

```
        });
});
```

Display Code

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, {useUnifiedTopology: true}, function(err, db)
{
  if (err) throw err;
  var dbo = db.db("webprogramming");
  dbo.collection("webprogramming").find({}).toArray(function(err, result)
  {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

Update Code

```
const prompt = require('prompt');

prompt.start();

prompt.get(['name', 'newname', 'newphonenumber'], function (err, result)
{
    if (err) { return onErr(err); }
    console.log('\n Name to be changed \n');
    console.log('  Name: ' + result.name);
    console.log(' Changed Name: ' + result.newname);
    console.log(' Changed Number: ' + result.newphonenumber);
    name = result.name;
    newname = result.newname;
    newphonenumber = result.newphonenumber;
    var MongoClient = require('mongodb').MongoClient;
        var url = "mongodb://localhost:27017/";
        MongoClient.connect(url, {useUnifiedTopology: true}, function(err, db)
        {
                  if (err) throw err;
                  var dbo = db.db("webprogramming");
                  var myquery = {name : name};
                  var newvalues = { $set: {name: newname, phonenumber :
                  ↪  newphonenumber}};
                  dbo.collection("webprogramming").updateOne(myquery, newvalues,
                  ↪  function(err, res)
                  {
                    if (err) throw err;
                    console.log("1 document updated");
                    db.close();
                  });
        });
});
```

Deletion Code

```
const prompt = require('prompt');

prompt.start();

prompt.get(['name'], function (err, result)
{
    if (err) { return onErr(err); }
    console.log('\n Name to be deleted \n');
    console.log('  Name: ' + result.name);
    name = result.name;
    var MongoClient = require('mongodb').MongoClient;
        var url = "mongodb://localhost:27017/";
        MongoClient.connect(url, {useUnifiedTopology: true}, function(err, db)
        {
                  if (err) throw err;
                  var dbo = db.db("webprogramming");
                  var myquery = {name: name};
                  dbo.collection("webprogramming").deleteOne(myquery, function(err,
                  ↪  obj)
                  {
                    if (err) throw err;
                    console.log("1 document deleted");
                    db.close();
                  });
        });
});
```

Output

Figure 1: **Node JS, Express and Mongo DB Pic1**

Figure 2: **Node JS, Express and Mongo DB Pic2**

```
PS C:\Users\badri\Desktop\WebProgrammingLab\Lab9> node show.js
[
  {
    _id: 608e52303a7bdd3d20f5b341,
    name: 'Badri',
    phonenumber: '8220584258'
  }
]
PS C:\Users\badri\Desktop\WebProgrammingLab\Lab9> node update.js
prompt: name:  Badri
prompt: newname:  MBadri
prompt: newphonenumber:  918220584258

 Name to be changed

  Name: Badri
 Changed Name: MBadri
 Changed Number: 918220584258
1 document updated
PS C:\Users\badri\Desktop\WebProgrammingLab\Lab9> node show.js
[
  {
    _id: 608e52303a7bdd3d20f5b341,
    name: 'MBadri',
    phonenumber: '918220584258'
  }
]
PS C:\Users\badri\Desktop\WebProgrammingLab\Lab9>
```

Result

Hence, a dynamic backend website has been implemented using NodeJS and MongoDB.