

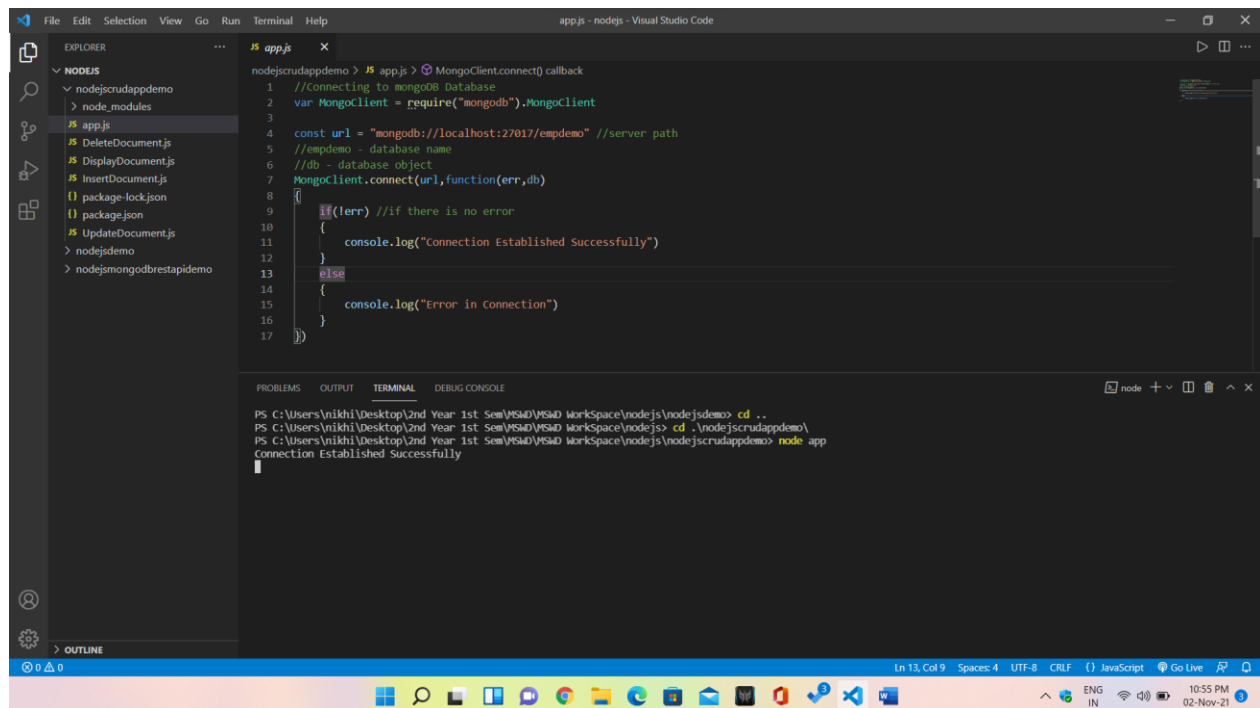
Skill 9

ID: 2000030384

Name: Jatla Nikhil Sai Lalith

Section No: S09

App.js:

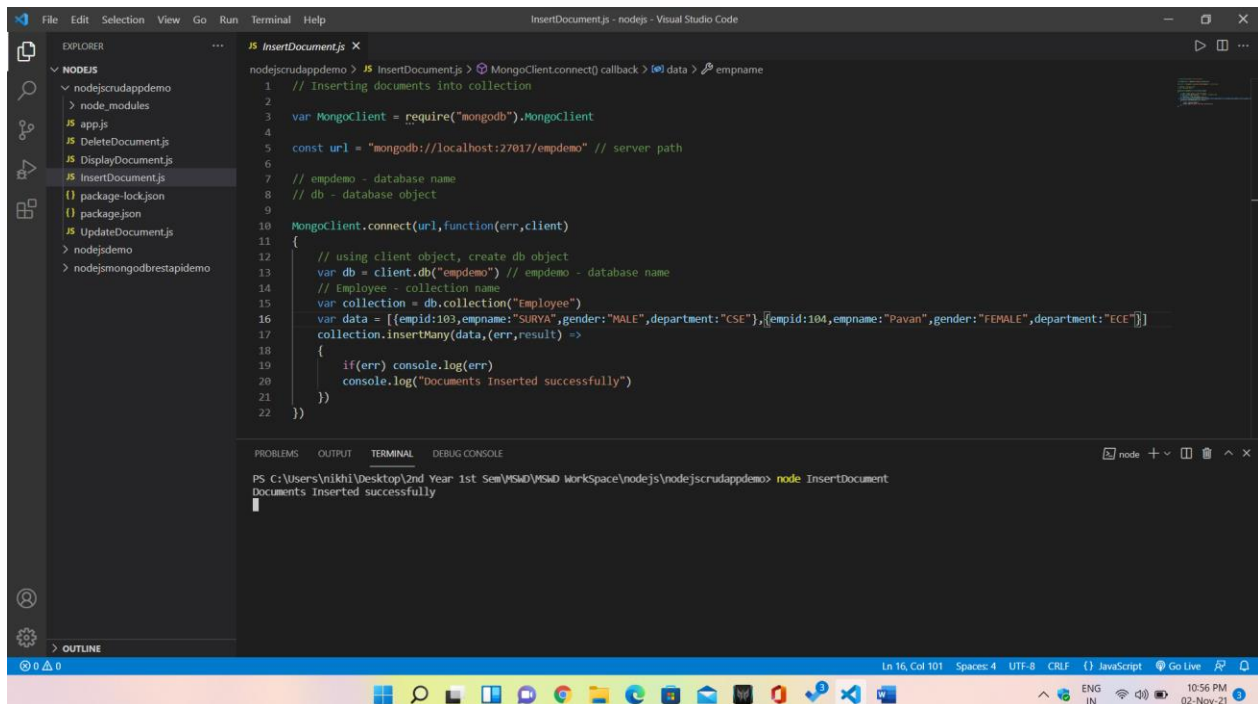


The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project structure with a 'node_modules' directory and several JavaScript files. The code editor displays the 'app.js' file, which contains a MongoDB connection script. The terminal shows the command 'node app' being executed, resulting in the message 'Connection Established Successfully'.

```
nodejscrudappdemo > .\app.js > MongoClient.connect() callback
1 //Connecting to mongoDB Database
2 var MongoClient = require("mongodb").MongoClient
3
4 const url = "mongodb://localhost:27017/empdemo" //server path
5 //empdemo - database name
6 //db - database object
7 MongoClient.connect(url,function(err,db)
8 {
9   if(!err) //if there is no error
10   {
11     console.log("Connection Established Successfully")
12   }
13   else
14   {
15     console.log("Error in Connection")
16   }
17 })
```

```
PS C:\Users\nikhil\Desktop\2nd Year 1st Sem\MSAD\MSAD Workspace\nodejs\nodejsdemo> cd ..
PS C:\Users\nikhil\Desktop\2nd Year 1st Sem\MSAD\MSAD Workspace\nodejs> cd .\nodejscrudappdemo\
PS C:\Users\nikhil\Desktop\2nd Year 1st Sem\MSAD\MSAD Workspace\nodejs\nodejscrudappdemo> node app
Connection Established Successfully
```

InsertDocument.js:

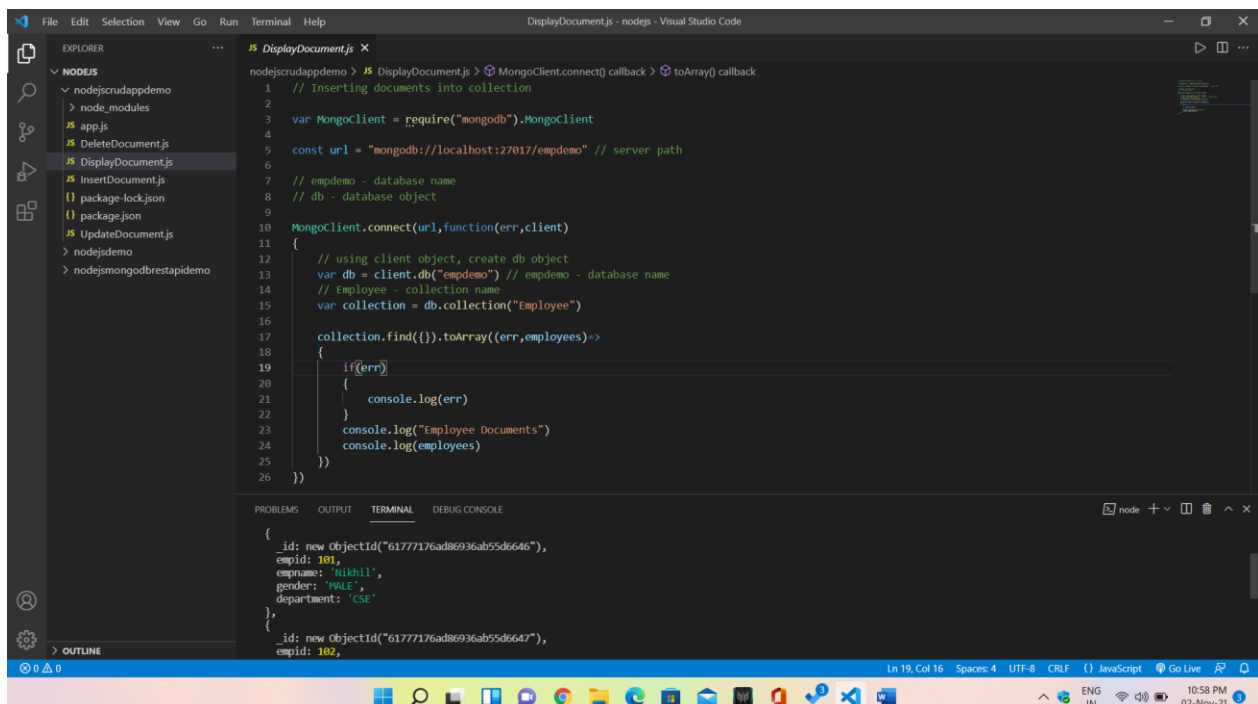


The screenshot shows the Visual Studio Code editor with the file `InsertDocument.js` open. The file explorer on the left shows the project structure. The code in `InsertDocument.js` connects to a MongoDB database named `empdemo` and inserts two documents into the `Employee` collection. The terminal at the bottom shows the command `node InsertDocument` being executed, resulting in the output `Documents Inserted successfully`.

```
1 // Inserting documents into collection
2
3 var MongoClient = require("mongodb").MongoClient
4
5 const url = "mongodb://localhost:27017/empdemo" // server path
6
7 // empdemo - database name
8 // db - database object
9
10 MongoClient.connect(url,function(err,client)
11 {
12     // using client object, create db object
13     var db = client.db("empdemo") // empdemo - database name
14     // Employee - collection name
15     var collection = db.collection("Employee")
16     var data = [{empid:103,empname:"SURYA",gender:"MALE",department:"CSE"},{empid:104,empname:"Pavan",gender:"FEMALE",department:"ECE"}]
17     collection.insertMany(data,(err,result) =>
18     {
19         if(err) console.log(err)
20         console.log("Documents Inserted successfully")
21     })
22 })
```

PS C:\Users\nikhi\Desktop\2nd Year 1st Sem\VS&D\VS&D workspace\nodejs\nodejscrudappdemo> node InsertDocument
Documents Inserted successfully

DisplayDocument.js:



The screenshot shows the Visual Studio Code editor with the file `DisplayDocument.js` open. The file explorer on the left shows the project structure. The code in `DisplayDocument.js` connects to a MongoDB database named `empdemo` and retrieves all documents from the `Employee` collection. The terminal at the bottom shows the command `node DisplayDocument` being executed, resulting in the output of the retrieved documents.

```
1 // Inserting documents into collection
2
3 var MongoClient = require("mongodb").MongoClient
4
5 const url = "mongodb://localhost:27017/empdemo" // server path
6
7 // empdemo - database name
8 // db - database object
9
10 MongoClient.connect(url,function(err,client)
11 {
12     // using client object, create db object
13     var db = client.db("empdemo") // empdemo - database name
14     // Employee - collection name
15     var collection = db.collection("Employee")
16
17     collection.find().toArray((err,employees)=>
18     {
19         if(err)
20         {
21             console.log(err)
22         }
23         console.log("Employee Documents")
24         console.log(employees)
25     })
26 })
```

{
 _id: new ObjectId("61777176ad86936ab55d6646"),
 empid: 101,
 empname: 'Nikhil',
 gender: 'MALE',
 department: 'CSE'
},
{
 _id: new ObjectId("61777176ad86936ab55d6647"),
 empid: 102,

Output:

File Edit Selection View Go Run Terminal Help

DisplayDocument.js - nodejs - Visual Studio Code

EXPLORER

- nodejscrudappdemo
 - node_modules
 - app.js
 - DeleteDocument.js
 - DisplayDocument.js
 - InsertDocument.js
 - package-lock.json
 - package.json
 - UpdateDocument.js
 - nodesdemo
 - nodejsmongodbrestapidemo

OUTLINE

DisplayDocument.js

```

nodejscrudappdemo > JS DisplayDocument.js > MongoClient.connect() callback > toArray() callback
1 // Inserting documents into collection
2
3 var MongoClient = require("mongodb").MongoClient
4
5 const url = "mongodb://localhost:27017/empdemo" // server path
6
7 try {
8   // Inserting documents into collection
9   MongoClient.connect(url, function(err, client) {
10     if (err) {
11       console.log("Error connecting to MongoDB: " + err);
12       return;
13     }
14     console.log("Connected to MongoDB");
15     // Inserting documents into collection
16     const db = client.db("empdemo");
17     const collection = db.collection("employees");
18     // Inserting documents into collection
19     collection.insertMany([
20       {
21         _id: new ObjectId("61777176ad86936ab55d6646"),
22         empid: 101,
23         empname: 'Nikhil',
24         gender: 'MALE',
25         department: 'CSE'
26       },
27       {
28         _id: new ObjectId("61777176ad86936ab55d6647"),
29         empid: 102,
30         empname: 'SURYA',
31         gender: 'MALE',
32         department: 'CSE'
33       },
34       {
35         _id: new ObjectId("61777176ad86936ab55d6648"),
36         empid: 103,
37         empname: 'Pavan',
38         gender: 'MALE',
39         department: 'CSE'
40       },
41       {
42         _id: new ObjectId("61777176ad86936ab55d6649"),
43         empid: 104,
44         empname: 'Pavani',
45         gender: 'FEMALE',
46         department: 'ECE'
47       }
48     ], function(err, result) {
49       if (err) {
50         console.log("Error inserting documents: " + err);
51       } else {
52         console.log("Documents inserted successfully: " + result);
53       }
54     });
55     client.close();
56   });
57 } catch (err) {
58   console.log("Error: " + err);
59 }

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\nikhi\Desktop\2nd Year 1st Sem\VSAD\VSAD Workspace\nodejs\nodejscrudappdemo> node DisplayDocument.js

Employee Documents

```

{
  _id: new ObjectId("61777176ad86936ab55d6646"),
  empid: 101,
  empname: 'Nikhil',
  gender: 'MALE',
  department: 'CSE'
},
{
  _id: new ObjectId("61777176ad86936ab55d6647"),
  empid: 102,
  empname: 'SURYA',
  gender: 'MALE',
  department: 'CSE'
},
{
  _id: new ObjectId("61777176ad86936ab55d6648"),
  empid: 103,
  empname: 'Pavan',
  gender: 'MALE',
  department: 'CSE'
},
{
  _id: new ObjectId("61777176ad86936ab55d6649"),
  empid: 104,
  empname: 'Pavani',
  gender: 'FEMALE',
  department: 'ECE'
}

```

Ln 19, Col 16 Spaces: 4 UTF-8 CRLF JavaScript Go Live

10:58 PM 02-Nov-21

DeleteDocument.js:

The screenshot displays the Visual Studio Code interface with a file named `DeleteDocument.js` open. The code is as follows:

```

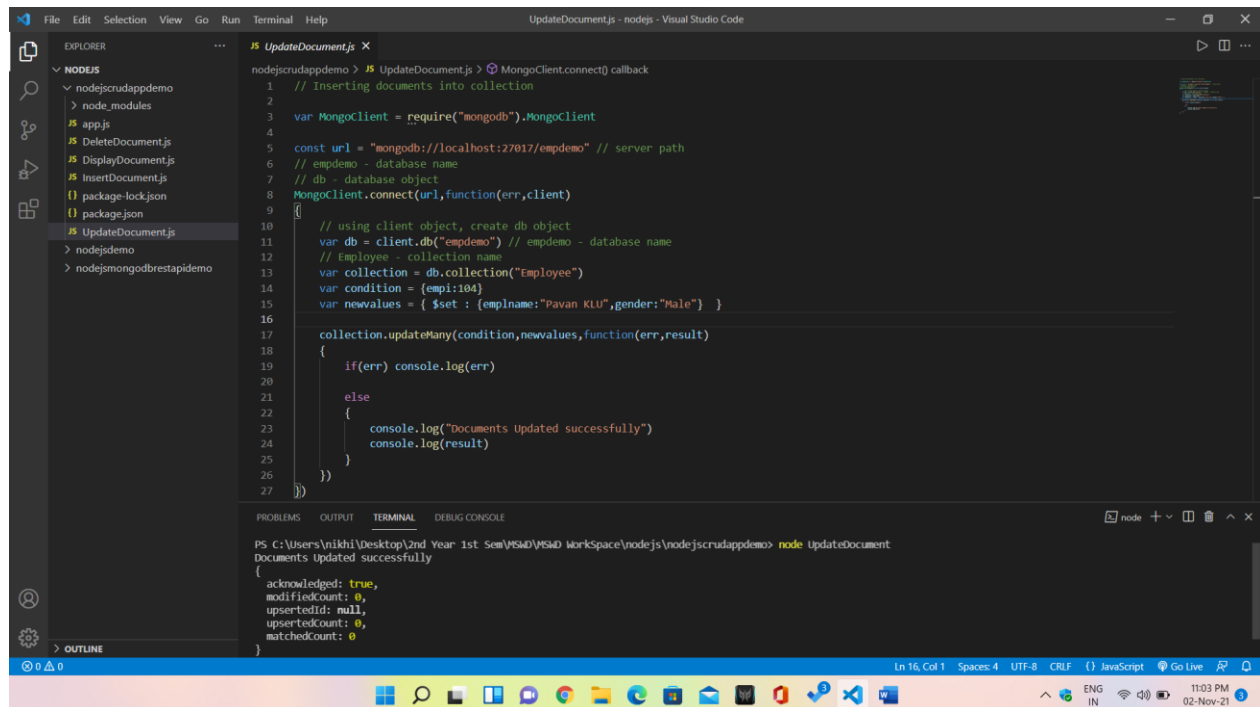
1 // Inserting documents into collection
2
3 var MongoClient = require("mongodb").MongoClient
4
5 const url = "mongodb://localhost:27017/empdemo" // server path
6 // empdemo - database name
7 // db - database object
8 MongoClient.connect(url,function(err,client)
9 {
10     // using client object, create db object
11     var db = client.db("empdemo") // empdemo - database name
12     // Employee - collection name
13     var collection = db.collection("Employee")
14     var condition = {empid:104}
15
16     collection.deleteMany(condition,function(err,result)
17     {
18         if(err) console.log(err)
19
20         else
21         {
22             console.log("Documents Deleted successfully")
23             console.log(result)
24         }
25     })
26 })

```

The Explorer sidebar on the left shows the project structure with files like `node_modules`, `app.js`, `DeleteDocument.js`, `DisplayDocument.js`, `InsertDocument.js`, `package-lock.json`, `package.json`, `UpdateDocument.js`, `nodesdemo`, and `nodesmongodbrestapidemo`.

The Terminal at the bottom shows the command `node DeleteDocument.js` being executed, resulting in the output: `Documents Deleted successfully`.

UpdateDocument.js:



The screenshot displays the Visual Studio Code editor with the file `UpdateDocument.js` open. The file is part of a project named `nodejscrudappdemo`. The code in `UpdateDocument.js` is as follows:

```
1 // Inserting documents into collection
2
3 var MongoClient = require("mongodb").MongoClient
4
5 const url = "mongodb://localhost:27017/empdemo" // server path
6 // empdemo - database name
7 // db - database object
8 MongoClient.connect(url,function(err,client)
9 {
10     // using client object, create db object
11     var db = client.db("empdemo") // empdemo - database name
12     // Employee - collection name
13     var collection = db.collection("Employee")
14     var condition = {empid:104}
15     var newvalues = { $set : {empname:"Pavan KLU",gender:"Male"} }
16
17     collection.updateMany(condition,newvalues,function(err,result)
18     {
19         if(err) console.log(err)
20     else
21     {
22         console.log("Documents Updated successfully")
23         console.log(result)
24     }
25 })
26 })
27 }
```

The terminal at the bottom shows the command `node UpdateDocument` being executed. The output indicates that the documents were updated successfully, with the following details:

```
PS C:\Users\nikhi\Desktop\2nd Year 1st Sem\VSAD\VSAD Workspace\nodejs\nodejscrudappdemo> node UpdateDocument
Documents Updated successfully
{
  acknowledged: true,
  modifiedCount: 0,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 0
}
```

The status bar at the bottom of the editor shows the current cursor position at line 16, column 1, and the file is encoded in UTF-8 with CRLF line endings. The system tray at the bottom right indicates the time is 11:03 PM on 02-Nov-21.