01/05/21 Vanathi
Ex. 8 185001188
CSE C

Programs using Node.js

Learning Objective -

To write Node.js programs for the given tasks - (a) to get a name from the console and display a randomized greeting, (b) to get a name from query string and display a randomized greeting on the webpage, (c) to implement a client - server program where the server returns a table with book details on the client's request, (d) to use MongoDB for add, delete, update and search operations in a patient database.

Programs -

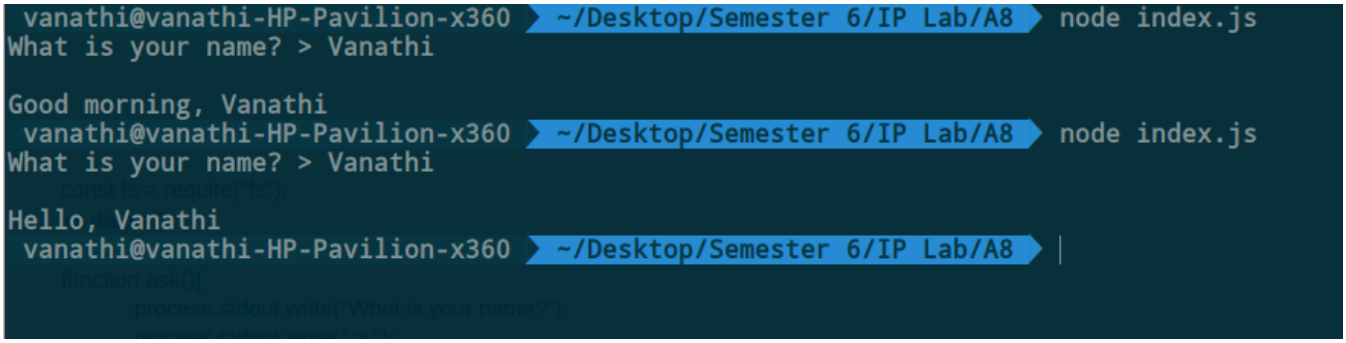
(a) index.js -

```
const fs = require("fs");
function ask(){
	process.stdout.write("What is your name?");
	process.stdout.write(" > ");
}
ask();

process.stdin.on("data", function(name) {
	fs.readFile('greetings.txt', function(err, data){
		const greetings = data.toString().split(/\r?\n/);
		const select = Math.floor(Math.random() * 6);
		process.stdout.write(`\n${greetings[select]}, ${name}`);
	});
});
```

greetings.txt -

```
Hello
Hi
Hey
Good morning
Good afternoon
Good evening
```

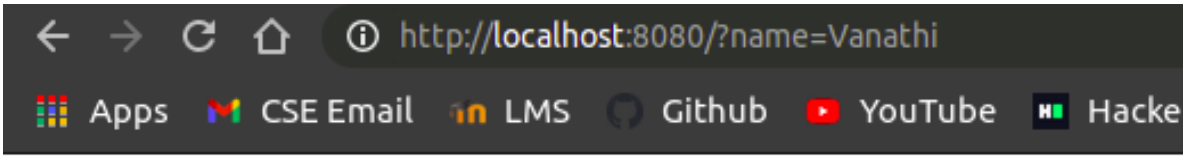
Output -



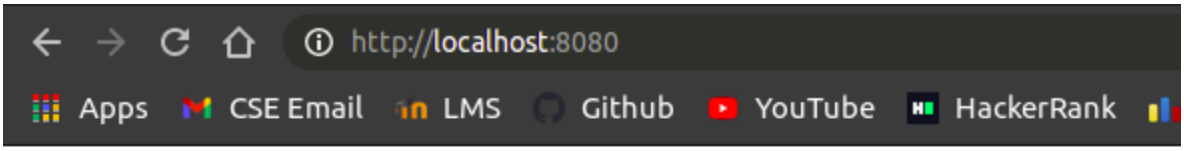
(b) index.js -

```
const http = require("http");
const url = require("url");
const fs = require("fs");

http.createServer(function(req, res){
    const query = url.parse(req.url,true).query
    let name = query.name;
    fs.readFile('greetings.txt', function(err, data){
            const greetings = data.toString().split(/\r?\n/);
            const select = Math.floor(Math.random() * 6);
            if(name == null)
                name = "";
            res.writeHead(200, {'Content-Type': 'text/html'});
            res.write(`<html><body><h1>\n${greetings[select]}
${name}!</h1></body></html>`);
            res.end();
    });

}).listen(8080)
```

Outputs -



Hello Vanathi!



Hello !

(c) server.js -

```
const http = require('http');
const fs = require('fs');
const url = require('url');

http.createServer( function (req, res) {
      const pathname = url.parse(req.url).pathname;
      fs.readFile(pathname.substr(1), function (err, data) {
                  if (err) {
```

```
                console.log(err);
                res.writeHead(404, {'Content-Type': 'text/html'});
        }

        else {
                res.writeHead(200, {'Content-Type': 'text/html'});
                res.write(data.toString());
        }
                res.end();
    });
}).listen(8080);
```

client.js -

```
const http = require('http');
const options = { host: 'localhost', port: '8080', path: '/index.html' };

function callback(response) {
    var body = '';
    response.on('data', function(data) {
        body += data;
    });
    response.on('end', function() {
        console.log(body);
    });
}
const req = http.request(options, callback);
req.end();
```

index.html -

```
<!DOCTYPE html>
<html>
<head>
  <title>Book Details</title>
  <style>
      table {
         border-collapse: collapse;
         width: 50%;
      }

      th, td {
         text-align: left;
         padding: 8px;
      }

      tr:nth-child(even) {
         background-color: aquamarine;
      }
   </style>
</head>
```

```
<body>
 <table>
  <thead>
   <h2>Book details</h2>
  </thead>
  <tbody>
   <tr>
    <th>Title</th>
    <th>Author</th>
    <th>Genre</th>
   </tr>
   <tr>
    <td>Ender's Game</td>
    <td>Orson Scott Card</td>
    <td>Science Fiction</td>
   </tr>
   <tr>
    <td>The LOTR : The Two Towers</td>
    <td>JRR Tolkien</td>
    <td>Fantasy</td>
   </tr>
   <tr>
    <td>Matilda</td>
    <td>Roald Dahl</td>
    <td>Children</td>
   </tr>
   <tr>
    <td>Artemis Fowl</td>
    <td>Eoin Colfer</td>
    <td>Young Adult</td>
   </tr>
  </tbody>
 </table></body></html>
```

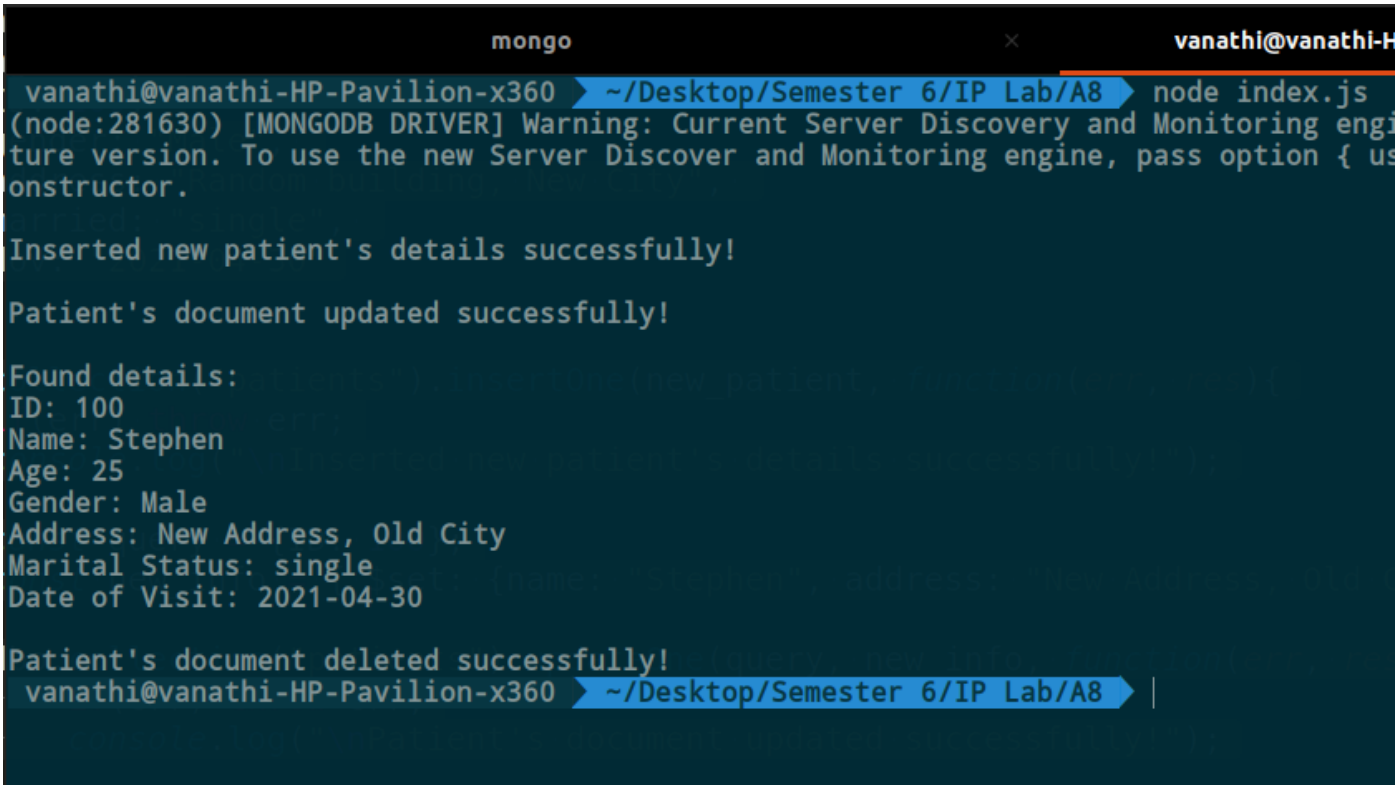
Output -

```
vanathi@vanathi-HP-Pavilion-x360 ~/Desktop/Semester 6/IP Lab/A8 node client.js
<!DOCTYPE html>
<html>
<head>
        <title>Book Details</title>
        <style>
        table {
            border-collapse: collapse;
            width: 50%;
        }

        th, td {
            text-align: left;
            padding: 8px;
        }

        tr:nth-child(even) {
            background-color: aquamarine;
        }
    </style>
</head>
<body>
        <table>
                <thead>
                        <h2>Book details</h2>
                </thead>
                <tbody>
                        <tr>
                                <th>Title</th>
                                <th>Author</th>
                                <th>Genre</th>
                        </tr>
                        <tr>
                                <td>Ender's Game</td>
                                <td>Orson Scott Card</td>
                                <td>Science Fiction</td>
                        </tr>
                        <tr>
                                <td>The LOTR : The Two Towers</td>
                                <td>JRR Tolkien</td>
                                <td>Fantasy</td>
                        </tr>
                        <tr>
                                <td>Matilda</td>
                                <td>Roald Dahl</td>
                                <td>Children</td>
                        </tr>
                        <tr>
                                <td>Artemis Fowl</td>
                                <td>Eoin Colfer</td>
                                <td>Young Adult</td>
                        </tr>
                </tbody>
        </table>
</body>
</html>
```

(d) index.js -

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

MongoClient.connect(url, function(err, db){
    var dbo = db.db("Patient_Details");
    const new_patient = {
            name: "Steve",
            age: 25,
            ID: 100,
            gender: "Male",
            address: "Random building, New City",
            married: "single",
            dov: "2021-04-30"
    }
    dbo.collection("patients").insertOne(new_patient, function(err, res){
            if(err) throw err;
            console.log("\nInserted new patient's details successfully!");

            const query = {ID: 100};
            const new_info = { $set: {name: "Stephen", address: "New Address, Old City" } };

            dbo.collection("patients").updateOne(query, new_info, function(err, res){
                    if(err) throw err;
                    console.log("\nPatient's document updated successfully!");

                    dbo.collection("patients").findOne({ name: "Stephen" }, function(err, result)
{
                            if(err) throw err;
                            console.log(`\nFound details:\nID: ${result.ID}\nName:
${result.name}\nAge: ${result.age}\nGender: ${result.gender}\nAddress:
${result.address}\nMarital Status: ${result.married}\nDate of Visit: ${result.dov}`);

                            dbo.collection("patients").deleteOne(query, function(err, obj) {
                                    if (err) throw err;
                            console.log("\nPatient's document deleted successfully!");
                                    db.close();
                            });
                    });
            });
    });
});
```

Output -



Learning Outcomes -

1. I successfully implemented all the programs using Node.js and MongoDB. I was able to install the required modules and use them.
2. I learnt about the various things we can do with Node.js such as file system operations, client request handling and database connections & operations.
3. I also successfully created a connection to the no-sql database and was able to modify the patients collection in the patient_details db. I learnt all the various methods available for MongoDB in Node.js
4. Overall, it was a good learning experience about Node.js and MongoDB