UCS 1611 - Internet Programming Lab

Exercise 8: Programs using Node.js

Swetha Saseendran
CSE-C
185001183

Learning objective:

a. Write a Node.js program that reads all the greetings from the file greetings.txt, asks the user "What is your name?", then prints a random greeting followed by the given name. Make sure to check for the case where the file doesn't exist! For example, if the greeting is "Hey", then the program will print "Hey, Joe" to the console, then pick some other greeting and do the same until finished. Use Non-blocking I/O.

b. Write a Node.js program that reads all the greetings as before. When all the greetings are loaded, it creates a server listening on port number 8080. On request, it checks for whether there is a name value in the query string. If there isn't, the value of query.name will be undefined. In other words, if you access http://localhost:8080/?name=Mike, then your browser should just display something like "Hello, Mike" when the page loads.

greetings.txt

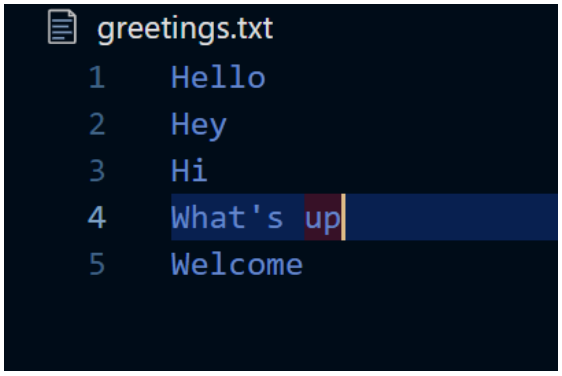
Hello
Hey
Hi
What's up
Welcome

c. Create a web server using node.js which listens for clients request. Once the client request the server, the server returns a web page which contains a list of books and its details in table format.

d. Create a DB with the following details using Mongodb: Database Name: Patient_Details Table Schema: Name, age, ID, gender, address, marital status, Date of Visit Wrtie a node.js program to do the following operations: Add, Delete, Update, Search.

Code:

greetings.txt



assign8.1.js

```
var fs = require("fs");
var readline = require("readline");

var rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout
});

if (!fs.existsSync("greetings.txt")) {
    console.log("File not found");
    rl.close();
} else {
    var greetingsFile = fs.readFileSync("greetings.txt");
    var greetings_string = greetingsFile.toString();
    var greetings = greetings_string.split("\n");

    rl.question("What is your name? ", function(name) {
        for (var i = 0; i < 1; i++) {
            var n = Math.floor(Math.random() * greetings.length);
            console.log(greetings[n]);
            console.log(name);
        }
        rl.close();
    });
}
```

assign8.2.js

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

var greetings;
```

```
fs.readFile("greetings.txt", function(err, body) {
  if (err === null) {
    greetings = body.toString().split("\n");
    if (greetings.slice(-1) == "") {
       greetings.pop();
    }

    var server = http.createServer(function(req, res) {
      res.writeHead(200);
      var query = url.parse(req.url, true).query;
      var name = query.name;
      var greeting = greetings[Math.floor(Math.random() * greetings.length)];
      if (name) {
        res.end(greeting + ", " + name);
      } else {
        res.end(greeting);
      }
    });
    server.listen(8080);
  } else {
    console.log(err);
  }
});
```

assign8.html

```
<!DOCTYPE html>
<html>

<head>
    <title>Simple Node Server</title>
</head>
<style>
    body {
        font-family: 'Source Sans Pro', sans-serif;
        font-size: 100%;
        background-color: #23b4cc;
    }

    div {
        box-sizing: border-box;
        padding: 3vw;
        margin-top: 30%;
        background: rgba(0, 0, 0, 0.78);
        max-width: 700px;
        margin: 0 auto;
        height: 200px;
    }
```

```
    table {
        border-collapse: collapse;
        color: aliceblue;
    }

    thead {
        color: #23b4cc;
    }
</style>

<body>
    <div>
        <table border="1px solid black" align="center">
            <thead>
                <tr>
                    <td><strong>Attribute</strong></td>
                    <td><strong>Value</strong></td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Name</td>
                    <td>Harry Potter</td>
                </tr>
                <tr>
                    <td>Author</td>
                    <td>JK Rowling</td>
                </tr>
                <tr>
                    <td>Description</td>
                    <td>Harry Potter is a series of seven fantasy novels writt
en by British author, J. K. Rowling. The novels chronicle the lives of a young
 wizard, Harry Potter. </td>
                </tr>
                <tr>
                    <td>Genre</td>
                    <td>Fantasy</td>
                </tr>
            </tbody>
        </table>
    </div>
</body>

</html>
```

assign8.3.js

```
var http = require('http');
var fs = require('fs');
var url = require('url');
http.createServer(function(request, response) {
    var pathname = url.parse(request.url).pathname;
    console.log("Request for " + pathname + " received.");
    fs.readFile('.' + pathname, function(err, data) {
        if (err) {
            console.log(err);
            response.writeHead(404, { 'Content-Type': 'text/html' });
        } else {
            response.writeHead(200, { 'Content-Type': 'text/html' });
            response.write(data.toString());
        }
        response.end();
    });
}).listen(8080);
console.log('Server running at http://127.0.0.1:8080/');
```

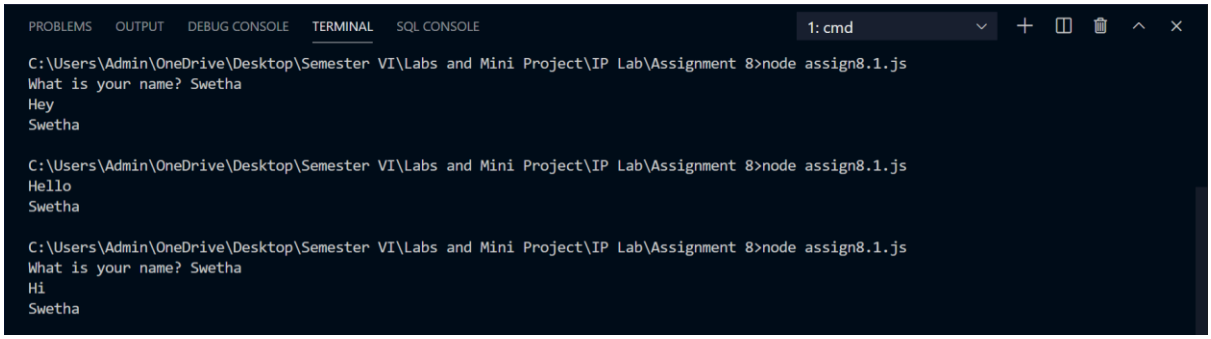
assign8.4.js

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

MongoClient.connect(url, {useUnifiedTopology: true }, function(err, db)
{
    if (err)
        throw err;
    console.log("Database Connected");
    var dbObject = db.db("Patient_Details");
    var myobj = {
        Name:'Swetha Saseendran',
        Age:21,
        ID:4,
        Gender:'Female',
        Address:'Kodambakkam',
        Marital_Status:'Single',
        DateOfVisit: Date()
    };

    //* INSERT
    dbObject.collection("patients").insertOne(myobj, function(err, res)
    {
        if (err) throw err;
        console.log("Inserting record");
        dbObject.collection('patients').find().toArray(function(err,res)
        {
```

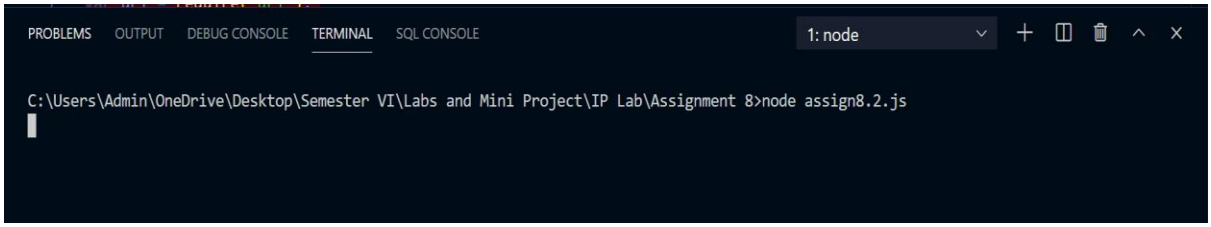
```
            if(err) throw err;
            console.log(res);
        });

        //* UPDATE
        var upd_url = { Name:"Swetha Saseendran" };
        var upd_values = { $set: {Marital_Status:"Married"} };
        dbObject.collection("patients").updateOne(upd_url, upd_values, functio
n(err, res)
        {
            if (err)
                throw err;
            console.log("Updated");

            //* SEARCH
            dbObject.collection('patients').find().toArray(function(err,res)
            {
                if(err)
                    throw err;
                console.log(res);
                console.log("Record Found");

                //* DELETE

                dbObject.collection('patients').deleteOne({Name:'murali'},func
tion(err,res)
                {
                    if(err)
                        throw err
                    console.log('Deleted');
                    db.close();
                });

            });

        });

    });
});
```

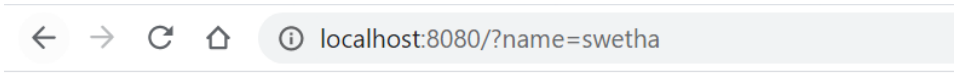
Output:

assign8.1.js

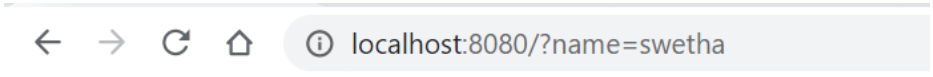


assign8.2.js

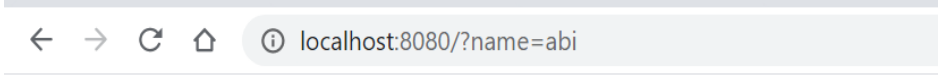





```
Hey
, swetha
```

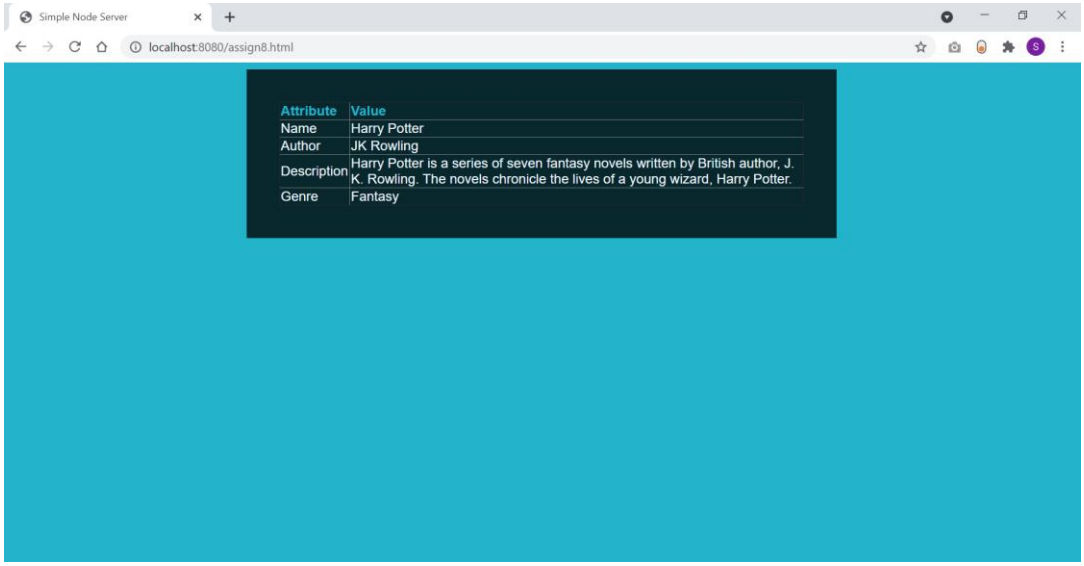



```
Hello
, swetha
```

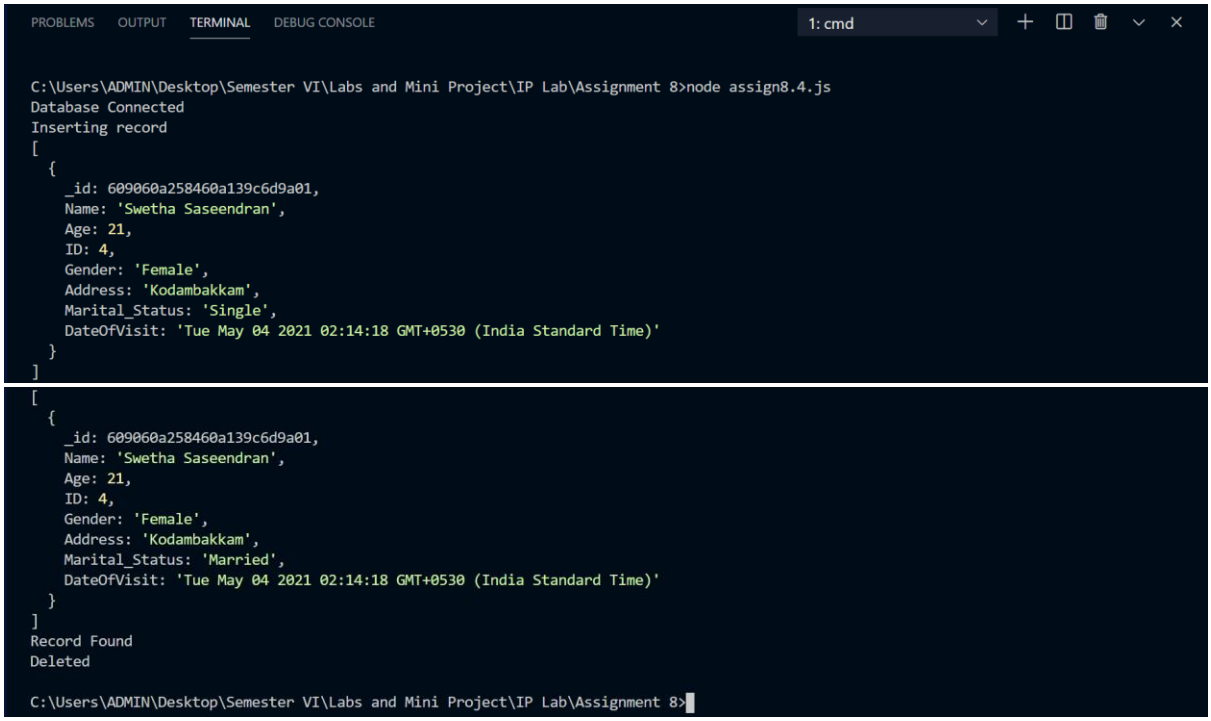



```
Hi
, abi
```

assign8.3.js



assign8.4.js



Learning Outcome:

- Understood how NodeJS is architected to allow high scalability with asynchronous code.
- Build an HTTP server using the core modules in NodeJS.
- Learnt to use stream I/O to efficiently serve the web pages and render them.
- Learnt to Interface to a MongoDB database and modify/ retrieve data.
- Learnt file operations and how to handle them using NodeJS.