Department of Computer Science and Engineering

S.G.Shivanirudh , 185001146, Semester VI

23 April 2021

UCS1611 - Internet Programming Lab

Ex 08: Programs using Node.js

Console non-blocking I/O

Objective:

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.

Code:

greetings.txt:

```
Hi
Hey
Hello
Vanakkam
Hola
Yo
What's up
Welcome
```

JavaScript:

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

var greetings = []

fs.readFile("greetings.txt", function(err, info){
    if(err){
        console.log("ERROR:File not found!");
        return 1;
    }
    greetings = info.toString().split("\n");
});
console.log("Completed reading file.");

const read = require('readline').createInterface({
    input: process.stdin,
    output: process.stdout
})

read.question("What is your name?\n", (name) => {
    console.log(`\n${greetings[Math.floor(Math.random() *
    greetings.length)]} ${name}`);
    read.close();
})
```

Output:

```
 ➜ node greetprog.js
 Completed reading file.
 What is your name?
 shiva

 Hola shiva
```

```
➜ node greetprog.js
Completed reading file.
What is your name?
shiva

Welcome shiva
```

```
➜ node greetprog.js
Completed reading file.
What is your name?
shiva

Yo shiva
```

Server greeting

Objective:

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.

Code:

greetings.txt:

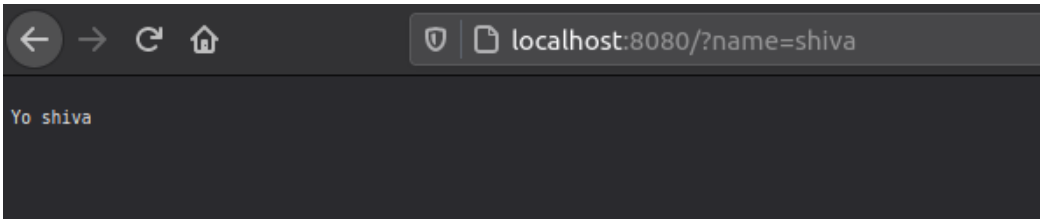
```
Hi
Hey
Hello
Vanakkam
Hola
Yo
What's up
Welcome
```

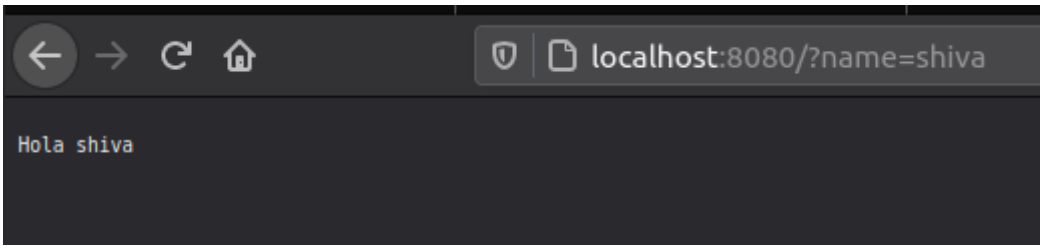
JavaScript:

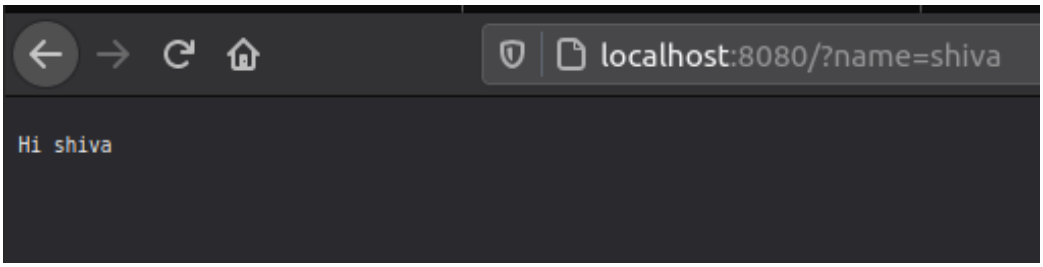
```
var fs = require("fs");
var http = require("http");
var url = require("url");

var greetings = []

fs.readFile("greetings.txt", function(err, info){
    if(err){
        console.log("ERROR:File not found!");
        return 1;
    }
    greetings = info.toString().split("\n");
```

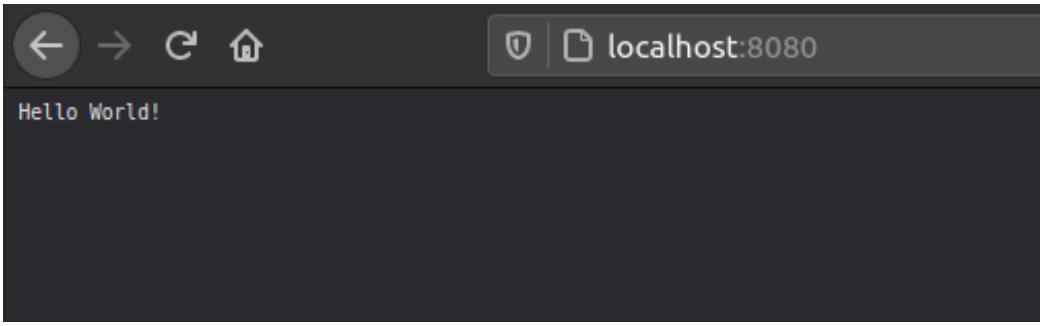
```
13 });
14 console.log("Completed reading file.");
15
16 http.createServer(function (req, res) {
17
18     var query = url.parse(req.url, true).query;
19     if(query.name){
20         var greet = greetings[Math.floor(Math.random() *
       greetings.length)];
21         res.write(`\n${greet} ${query.name}`);
22     }
23     else{
24         res.write('Hello World!');
25     }
26     res.end();
27 }).listen(8080);
```

Output:


localhost:8080/?name=shiva
Yo shiva


localhost:8080/?name=shiva
Hola shiva


localhost:8080/?name=shiva
Hi shiva


localhost:8080
Hello World!

Server render HTML

Objective:

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

Code:

HTML:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>Book Category</title>
    <style>
        body{
    font-size: 16pt;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
    background: rgb(21,27,14);
    background: linear-gradient(90deg, rgba(21,27,14,1) 0%, rgba(40,4,65,1) 44%, rgba(61,7,196,1) 100%);
    color: white;
    height: 100vh;
    text-align: center;
}

h1{
    font-size: 50pt;
    text-align: center;
    color: white;
}
```

```
25
26 table{
27     display: block;
28     margin-left: auto;
29     margin-right: auto;
30     width: 25%;
31     text-align: center;
32     border-spacing: 25px;
33 }
34
35
36     </style>
37 </head>
38
39 <body>
40     <h1>BOOKS LIST</h1>
41     <hr><br>
42
43     <table align="center">
44         <tr>
45             <th>Title</th>
46             <th>Author</th>
47             <th>Publisher</th>
48             <th>Country</th>
49         </tr>
50         <tr>
51             <td>The Hobbit </td>
52             <td>J. R. R. Tolkien</td>
53             <td>Houghton Mifflin</td>
54             <td>USA</td>
55         </tr>
56         <tr>
57             <td>Slinky Malinki</td>
58             <td>Lynley Dodd</td>
59             <td>Mallinson Rendel</td>
60             <td>NZ</td>
61         </tr>
62         <tr>
63             <td>Hairy Maclary from Donaldson's Dairy</td>
64             <td>Lynley Dodd</td>
65             <td>Mallinson Rendel</td>
66             <td>NZ</td>
67         </tr>
68         <tr>
69             <td>How to Lie with Statistics</td>
```

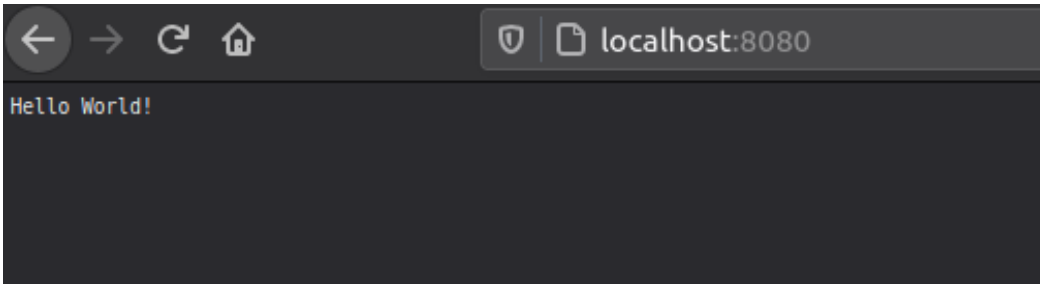
```
            <td>Darrell Huff</td>
            <td>W. W. Norton</td>
            <td>USA</td>
        </tr>
        <tr>
            <td>Mechanical Harry</td>
            <td>Bob Kerr </td>
            <td>Mallinson Rendel</td>
            <td>NZ</td>
        </tr>
        <tr>
            <td> My Cat Likes to Hide in Boxes</td>
            <td>Lynley Dodd </td>
            <td>Mallinson Rendel</td>
            <td>NZ</td>
        </tr>
    </table>
</body>

</html>
```

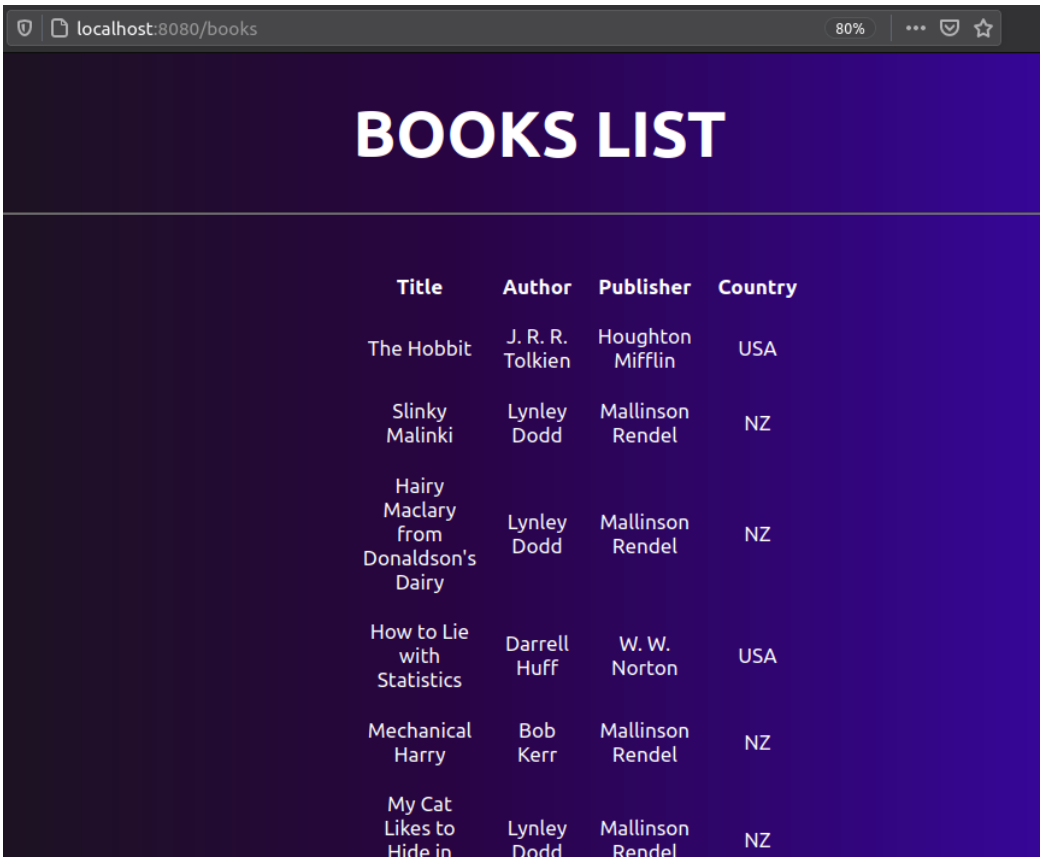
JavaScript:

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

var contents = "";

fs.readFile("display.html", function(err, info){
    if(err){
        console.log("ERROR:File not found!");
        return 1;
    }
    contents = info.toString();
});
console.log("Completed reading file.");

http.createServer(function (req, res) {

    var query = url.parse(req.url, true).pathname;
    if(query==="/books"){
```

```
20          res.write(`${contents}`);
21      }
22      else{
23          res.write('Hello World!');
24      }
25      res.end();
26  }).listen(8080);
```

Output:




localhost:8080/books
80%
BOOKS LIST
Title
Author
Publisher
Country
The Hobbit
J. R. R. Tolkien
Houghton Mifflin
USA
Slinky Malinki
Lynley Dodd
Mallinson Rendel
NZ
Hairy Maclary from Donaldson's Dairy
Lynley Dodd
Mallinson Rendel
NZ
How to Lie with Statistics
Darrell Huff
W. W. Norton
USA
Mechanical Harry
Bob Kerr
Mallinson Rendel
NZ
My Cat Likes to Hide in
Lynley Dodd
Mallinson Rendel
NZ

MongoDB

Objective:

Create a DB withthe following details using Mongodb. Wrtie a node.js programto do the following operations:Add, Delete, Update, Search.

Code:

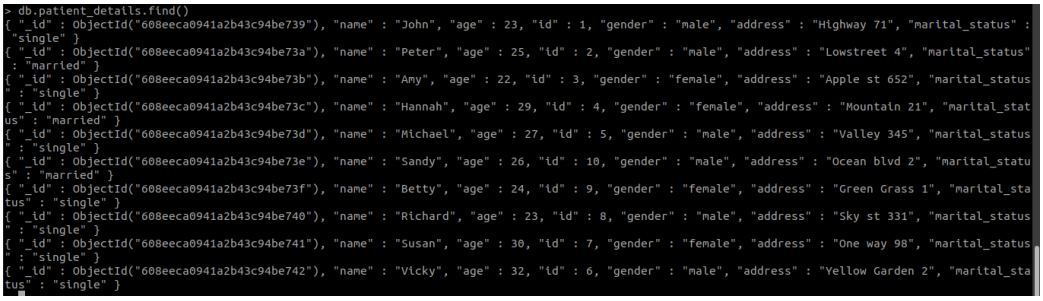
JavaScript:

```
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', age: 23, id: 1, gender:'male', address: '
   Highway 71', marital_status:'single'},
    { name: 'Peter', age: 25, id: 2, gender:'male', address:
   'Lowstreet 4', marital_status:'married'},
    { name: 'Amy', age: 22, id: 3, gender:'female', address:
   'Apple st 652', marital_status:'single'},
    { name: 'Hannah', age: 29, id: 4, gender:'female',
   address: 'Mountain 21', marital_status:'married'},
    { name: 'Michael', age: 27, id: 5, gender:'male', address
   : 'Valley 345', marital_status:'single'},
    { name: 'Sandy', age: 26, id: 10, gender:'male', address:
    'Ocean blvd 2', marital_status:'married'},
    { name: 'Betty', age: 24, id: 9, gender:'female', address
   : 'Green Grass 1', marital_status:'single'},
    { name: 'Richard', age: 23, id: 8, gender:'male', address
   : 'Sky st 331', marital_status:'single'},
    { name: 'Susan', age: 30, id: 7, gender:'female', address
   : 'One way 98', marital_status:'single'},
    { name: 'Vicky', age: 32, id: 6, gender:'male',  address:
    'Yellow Garden 2', marital_status:'single'},
  ];
```

```
19
20    //Add
21    dbo.collection("patient_details").insertMany(myobj,
       function(err, res) {
22      if (err) throw err;
23      console.log("Number of documents inserted: " + res.
       insertedCount);
24      db.close();
25    });
26
27    var del_query = { address: 'Mountain 21' };
28    dbo.collection("patient_details").deleteOne(del_query,
       function(err, obj) {
29      if (err) throw err;
30      console.log("1 document deleted");
31      db.close();
32    });
33
34    var upd_query = { address: "Valley 345" };
35    var newvalues = { $set: {name: "Mickey", address: "Canyon
       123" } };
36    dbo.collection("patient_details").updateOne(upd_query,
       newvalues, function(err, res) {
37      if (err) throw err;
38      console.log("1 document updated");
39      db.close();
40    });
41
42    var search_query = { gender: /^male/ };
43    dbo.collection("patient_details").find(search_query).
       toArray(function(err, result) {
44    if (err) throw err;
45    console.log(result);
46    db.close();
47      });
48
49  });
```

Output:

```
1 document deleted
[]
1 document updated
Number of documents inserted: 10
```



```
> db.patient_details.find()
{ "_id" : ObjectId("608eeca0941a2b43c94be739"), "name" : "John", "age" : 23, "id" : 1, "gender" : "male", "address" : "Highway 71", "marital_status" :
 "single" }
{ "_id" : ObjectId("608eeca0941a2b43c94be73a"), "name" : "Peter", "age" : 25, "id" : 2, "gender" : "male", "address" : "Lowstreet 4", "marital_status"
 : "married" }
{ "_id" : ObjectId("608eeca0941a2b43c94be73b"), "name" : "Amy", "age" : 22, "id" : 3, "gender" : "female", "address" : "Apple st 652", "marital_status
" : "single" }
{ "_id" : ObjectId("608eeca0941a2b43c94be73c"), "name" : "Hannah", "age" : 29, "id" : 4, "gender" : "female", "address" : "Mountain 21", "marital_stat
us" : "married" }
{ "_id" : ObjectId("608eeca0941a2b43c94be73d"), "name" : "Michael", "age" : 27, "id" : 5, "gender" : "male", "address" : "Valley 345", "marital_status
" : "single" }
{ "_id" : ObjectId("608eeca0941a2b43c94be73e"), "name" : "Sandy", "age" : 26, "id" : 10, "gender" : "male", "address" : "Ocean blvd 2", "marital_statu
s" : "married" }
{ "_id" : ObjectId("608eeca0941a2b43c94be73f"), "name" : "Betty", "age" : 24, "id" : 9, "gender" : "female", "address" : "Green Grass 1", "marital_sta
tus" : "single" }
{ "_id" : ObjectId("608eeca0941a2b43c94be740"), "name" : "Richard", "age" : 23, "id" : 8, "gender" : "male", "address" : "Sky st 331", "marital_status
" : "single" }
{ "_id" : ObjectId("608eeca0941a2b43c94be741"), "name" : "Susan", "age" : 30, "id" : 7, "gender" : "female", "address" : "One way 98", "marital_status
" : "single" }
{ "_id" : ObjectId("608eeca0941a2b43c94be742"), "name" : "Vicky", "age" : 32, "id" : 6, "gender" : "male", "address" : "Yellow Garden 2", "marital_sta
tus" : "single" }
```