

Cloud based enterprise systems lab (15B28CI581)

Assignment week 7

Patil Amit Gurusidhappa

19104004

B11

1. WAP to print hello world on port number 9090. Access these contents from another system via web browser.

Sever.js

```
const express = require('express')
const app = express()
const port = 9090

app.listen(port, () => console.log(`Q1 app listening on port ${port}!`))

app.get('/', (req, res) => {
  res.send("Hello World");
})
```

Package.json

```
{
  "name": "q1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "nodemon server.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.3",
    "nodemon": "^2.0.15"
  }
}
```

Output



2. Create a new module that returns your name (e.g. Manish). Further, include this module in a new file to append your name with the “Hello” keyword. E.g. Hello Manish Server.js

```
const express = require('express')
const app = express()
const port = 9090

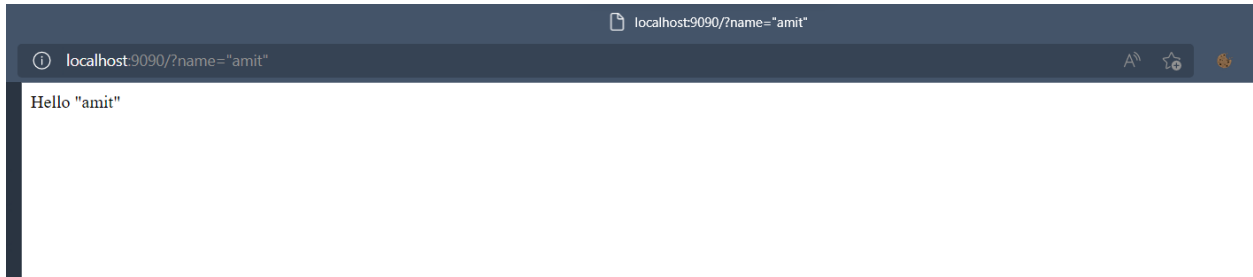
app.listen(port, () => console.log(`Q2 app listening on port ${port}!`))

app.get('/', (req, res) => {
  name = req.query.name;
  res.send(`Hello ${name}`);
})
```

Package.json

```
{
  "name": "q2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "nodemon server.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.3"
  }
}
```

Output



3. Create a new file abc.txt and include “I love my India” in this file. Create the file using

all three methods. Further (i) append the content “My name is” at end of abc.txt file

(ii) replace the contents and create new data in this file.

Server.js

```
const express = require('express')
const fs = require('fs')
const app = express()
const port = 9090

app.listen(port, () => console.log(`Q2,3,4 app listening on port ${port}!`))
app.get('/file', (req, res) => {
  fs.writeFile('abc.txt', "I love my India", function (err) {
    if (err) {

      return;
    }
    console.log(`File Created Successfully`);
  })

  fs.appendFile('abc.txt', 'My name is Amit', function (err) {
    if (err) throw err;
    console.log('Append content Success');
  });

  fs.writeFile('abc.txt', "New content after replacing", function (err)
  {
    if (err) {
      res.send("filed to replace content of the file");
      return;
    }
  })
})
```

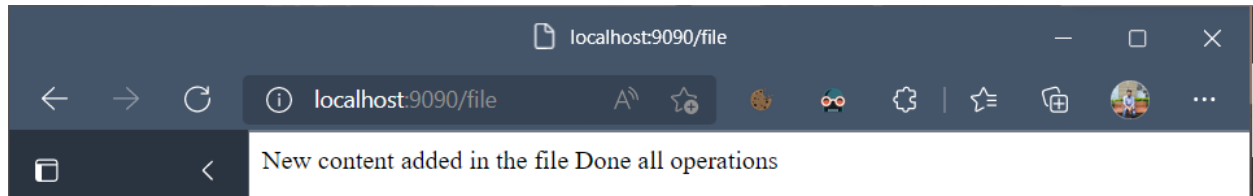
```

        console.log(`New content added in the file`);

    })
    res.send("New content added in the file\n Done all operations");
})

```

Output



4. WAP to explore the functionality of rename and delete the file. Also, consider the case

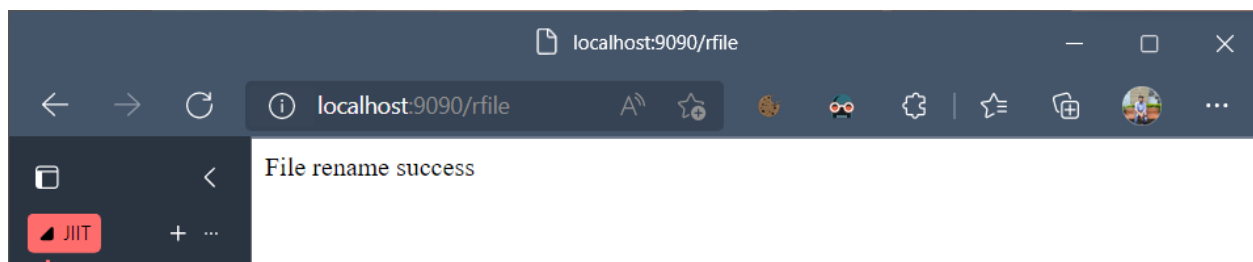
where (i) file does not exist (ii) Rename/deletion is not permitted.

Rename file

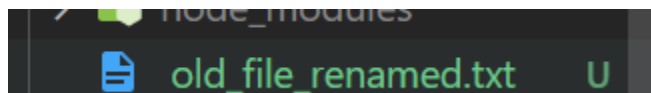
```

app.get('/rfile', (req, res) => {
    fs.rename("old_file.txt", " old_file_renamed.txt", function(err) {
        res.send("File Cant be renamed");
        throw err;
        return;
    })
    res.send("File rename success");
})

```

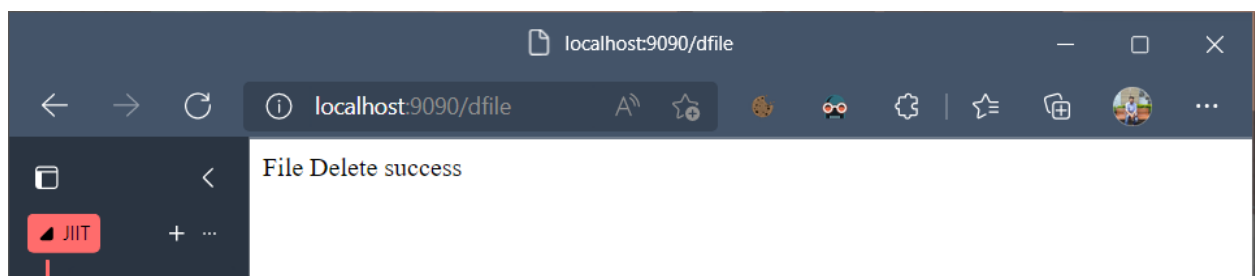


File Renamed



Delete File

```
app.get('/dfile', (req, res) => {  
  fs.unlink("file_to_be_deleted.txt", function(err) {  
    res.send("File Cant be deleted");  
    return;  
  })  
  res.send("File Delete success");  
})
```



5. Write a program to sort the numeric array using bubble sort using REPL terminal.
(Insert code and screenshot in file)

```
function bblSort(arr) {  
  
  for (var i = 0; i < arr.length; i++) {  
  
    for (var j = 0; j < (arr.length - i - 1); j++) {  
  
      if (arr[j] > arr[j + 1]) {  
  
        var temp = arr[j]  
        arr[j] = arr[j + 1]  
        arr[j + 1] = temp  
      }  
    }  
  }  
}
```

```
app.get('/bsort', (req, res) => {  
  
  // This is our unsorted array  
  var arr = [234, 43, 55, 63, 5, 6, 235, 547];  
  
  var copyarr=arr;  
  // Now pass this array to the bblSort() function  
  bblSort(arr);  
  res.send(`Initla array arr ${copyarr} After bubble sort ${arr}`);  
  
})
```

