Practical Assignment -1

Name: Tailor kruti P

Roll No: 76

Semester : 7th

Subject : Application development using Node.js

1. Develop a web server with following functionalities:

- Serve static resources.

- Handle GET request.

- Handle POST request.

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

const server = http.createServer((req,res)=>{
    var ul = url.parse(req.url,true);
    if(req.url == "/hi"){
        var path = "./index.html";
        fs.readFile(path,(err,data)=>{
            if(err){
                res.writeHead(404,{'Content-Type':'text/html'});
                return res.end("404 page not found");
            }
            res.writeHead(200,{'Content-Type':'text/html'});
            res.write(data);
            res.end();
        });
    }
    else if(ul.pathname=="/process" && req.method=="GET")
    {
        res.write("First name :" +ul.query.fname+ "\n Last Name:"
+ul.query.lname);
        res.end();
    }
    else if(ul.pathname=="/process" && req.method=="POST")
    {
        let body = '';
        req.on('data',chunk=>{
            body+=chunk.toString();
        });
        req.on(end,()=>{
            req.end(body);
        });
    }
});

server.listen(8080);
console.log("Server listening on port 8080");
```

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

const server = http.createServer((req,res)=>{
    var ul = url.parse(req.url,true);
    if(req.url == "/hi"){
        var path = "./index.html";
        fs.readFile(path,(err,data)=>{
            if(err){
                res.writeHead(404,{'Content-Type':'text/html'});
                return res.end("404 page not found");
            }
            res.writeHead(200,{'Content-Type':'text/html'});
            res.write(data);
            res.end();
        });
    }
    else if(ul.pathname=="/process" && req.method=="GET")
    {
        res.write("First name :" +ul.query.fname+ "\n Last Name:" +ul.query.lname);
        res.end();
    }
    else if(ul.pathname=="/process" && req.method=="POST")
    {
        let body = '';
        req.on('data',chunk=>{
```

```
PS C:\Users\krutii\OneDrive\Documents\node> cd Q1
PS C:\Users\krutii\OneDrive\Documents\node\Q1> node server
Server listening on port 8080
node:events:491
      throw er; // Unhandled 'error' event
      ^

Error: listen EADDRINUSE: address already in use :::8080
```


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

const server = http.createServer((req,res)=>{
    var ul = url.parse(req.url,true);
    if(req.url == "/hi"){
        var path = "./index.html";
        fs.readFile(path,(err,data)=>{
            if(err){
                res.writeHead(404,{'Content-Type':'text/html'});
                return res.end("404 page not found");
            }
            res.writeHead(200,{'Content-Type':'text/html'});
            res.write(data);
            res.end();
        });
    }
    else if(ul.pathname=="/process" && req.method=="GET")
    {
        res.write("First name :" +ul.query.fname+ "\n Last Name:" +ul.query.lname);
        res.end();
    }
    else if(ul.pathname=="/process" && req.method=="POST")
    {
        let body = '';
        req.on('data',chunk=>{
```


```
    at emitErrorNT (node:net:1767:8)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21) {
  code: 'EADDRINUSE',
  errno: -4091,
  syscall: 'listen',
  address: '::',
  port: 8080
}
```

2. Develop nodejs application with following requirements: -

Develop a route "/gethello" with GET method. It displays "Hello NodeJS!!" as response.

- Make an HTML page and display.

- Call "/gethello" route from HTML page using AJAX call. (Any frontend AJAX call API can be used.)

```
const http= require('http');
const fs= require('fs');
http.createServer((req,res)=>{
    if(req.method=="GET"){
        if(req.url=="/"){
            res.end("Home page");
        }
        else if(req.url =="/gethello")
        {
            fs.readFile("./hello.html",(err,data)=>{
                if(err)
                {
                    return res.send("something went wrong!!");
                }
                else{
                    res.writeHead(200,{'Content-Type':'text/html'});
                    res.write(data);
                    return res.end();
                }
            })
        }
        else(req.url=="/ajaxcall")
        {
            fs.readFile('./ajaxcall.html',(err,data)=>{
                if(err){
                    return res.send("Something went wrong");
                }
                else{
                    res.writeHead(200,{'Content-Type':'text/html'});
                    res.write(data);
                     return res.end();
                }
            })

        }
    }
}).listen(8080,()=>{
    console.log("Server listeing on port 8080");
})
```

Ajaxcall.html

```
<!DOCTYPE html>
<html>
    <body>
        <div id = "page_content"></div>
        <button onclick="loadData()""> Fetch Page </button>
        <script>
            function loadData(){
                var xhttp = new XMLHttpRequest();
                xhttp.onreadystatechange =function(){
                    if(this.readyState == 4 && this.status ==200){
                        document.getElementById("page_content").innerHTML=this
.responseText;

                    }
                };
                xhttp.open("GET","/gethello",true);
                xhttp.send();
            }
        </script>
    </body>
</html>
```

Hello.html

```
<!DOCTYPE html>
<html>
    <body>
        <h3> Hello Node JS !!!</h3>
    </body>
</html>
```

Output:

```js
const http= require('http');
const fs= require('fs');
http.createServer((req,res)=>{
    if(req.method=="GET"){
        if(req.url=="/"){
            res.end("Home page");
        }
        else if(req.url =="/gethello")
        {
            fs.readFile("./hello.html",(err,data)=>{
                if(err)
                {
                    return res.send("something went wrong!!");
                }
                else{
                    res.writeHead(200,{'Content-Type':'text/html'});
                    res.write(data);
                    return res.end();
                }
            })
        }
        else(req.url=="/ajaxcall")
        {
            fs.readFile('./ajaxcall.html',(err,data)=>{
                if(err){
                    return res.send("Something went wrong");
                }
                else{
```

```
PS C:\Users\kruti\OneDrive\Documents\node\Q2> node server
Server listeing on port 8080
PS C:\Users\kruti\OneDrive\Documents\node\Q2> node server
Server listeing on port 8080
```

```js
                    res.write(data);
                    return res.end();
                }
            })
        }
        else(req.url=="/ajaxcall")
        {
            fs.readFile('./ajaxcall.html',(err,data)=>{
                if(err){
                    return res.send("Something went wrong");
                }
                else{
                    res.writeHead(200,{'Content-Type':'text/html'});
                    res.write(data);
                    return res.end();
                }
            })

        }
    }
}).listen(8080,()=>{
    console.log("Server listeing on port 8080");
})
```

```
PS C:\Users\kruti\OneDrive\Documents\node\Q2> node server
Server listeing on port 8080
PS C:\Users\kruti\OneDrive\Documents\node\Q2> node server
Server listeing on port 8080
```

localhost:8080

localhost:8080

Home page

30°C
Mostly cloudy

Search

ENG
US

17:34
31-07-2023

localhost:8080/gethello

localhost:8080/gethello

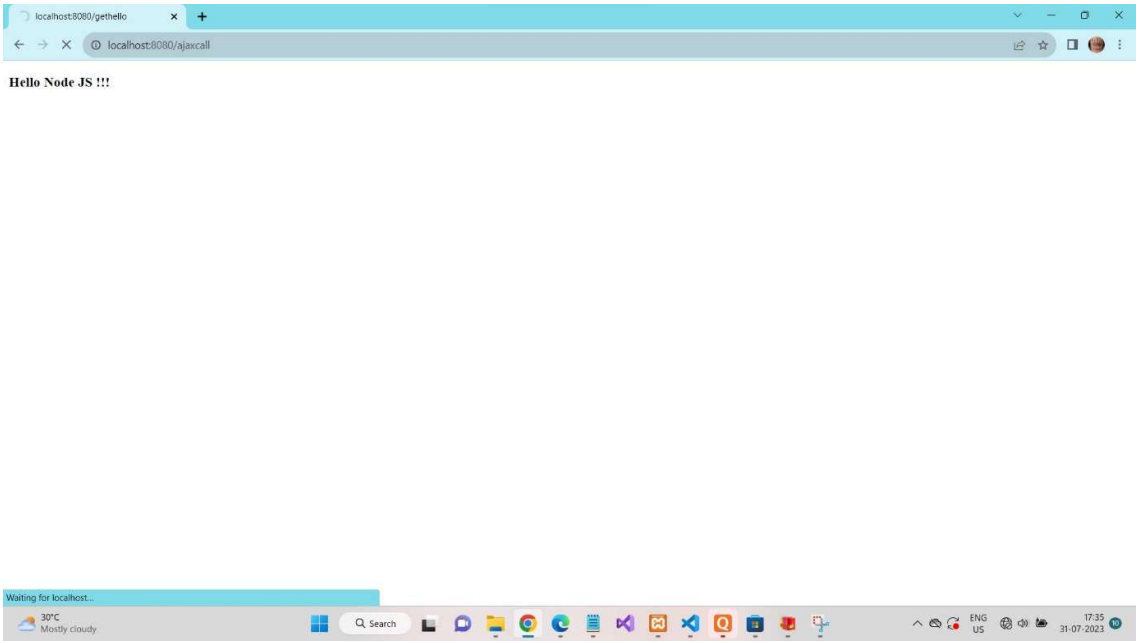
Hello Node JS !!!

30°C
Mostly cloudy

Search

ENG
US

17:34
31-07-2023



3. Develop a module for domain specific chatbot and use it in a command line application.

Chatboat.js

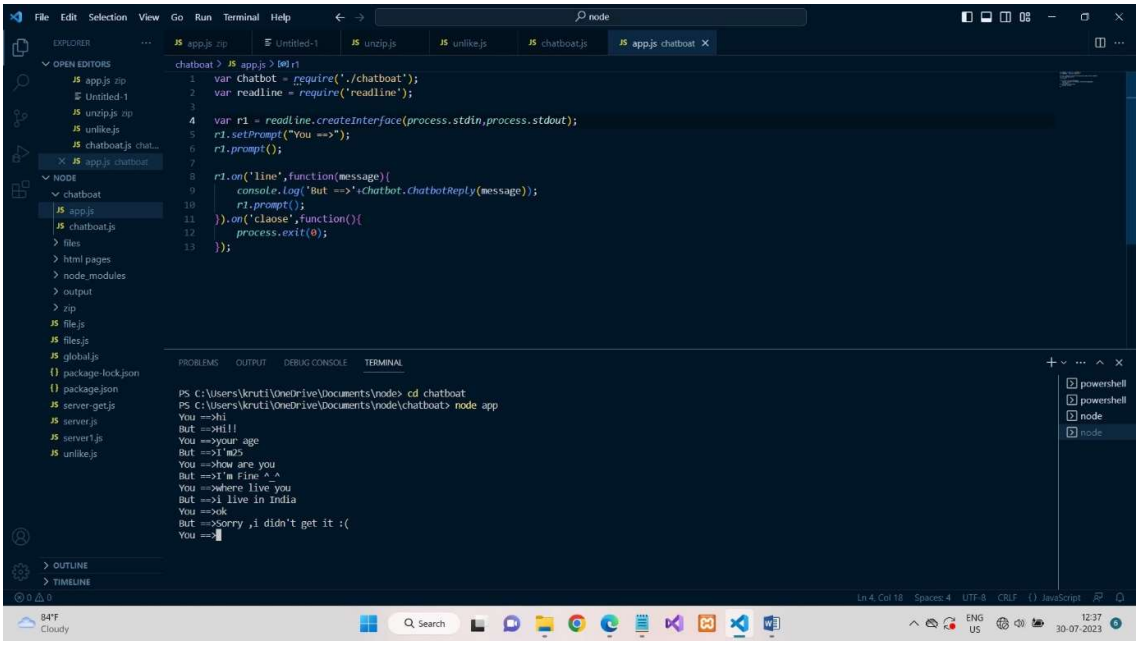
```
module.exports.ChatbotReply = function(message){
    this.Bot_Age = 25;
    this.Bot_Name = "Kruti";
    this.Bot_University = "VNSGU";
    this.Bot_Country = "India";

    message = message.toLowerCase()
    if(message.indexOf("hi")>-1 || message.indexOf("hello")>-1 ||
message.indexOf("Welcome")>-1){
        return "Hi!!";
    }
    else if(message.indexOf("age")>-1 && message.indexOf("Your")){
        return "I'm" + this.Bot_Age;
    }
    else if(message.indexOf("how")>-1 && message.indexOf("are") &&
message.indexOf("you")){
        return "I'm Fine ^_^"
    }
    else if(message.indexOf("where")>-1 && message.indexOf("live") &&
message.indexOf("you")){
        return "i live in " +this.Bot_Country;
    }
```

```
    return "Sorry ,i didn't get it :(";
}
```

App.js

```
var Chatbot = require('./chatboat');
var readline = require('readline');

var r1 = readline.createInterface(process.stdin,process.stdout);
r1.setPrompt("You ==>");
r1.prompt();

r1.on('line',function(message){
    console.log('But ==>'+Chatbot.ChatbotReply(message));
    r1.prompt();
}).on('claose',function(){
    process.exit(0);
});
```

Output:



4. Use above chatbot module in web based chatting of websocket.

Websoket.js

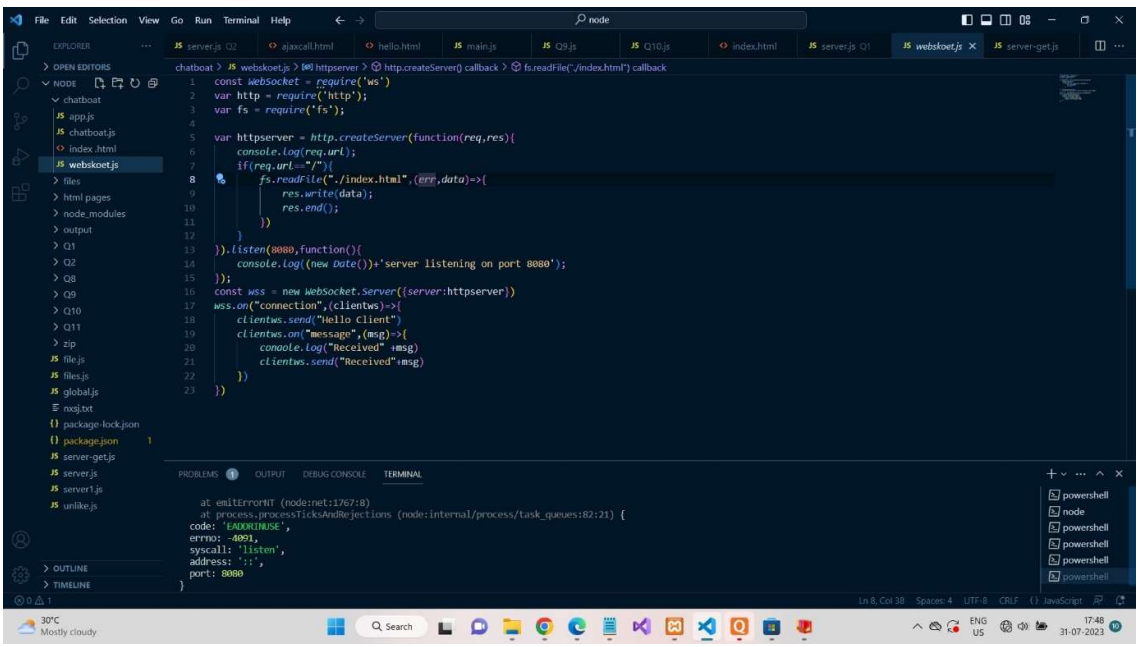
```
const WebSocket = require('ws')
var http = require('http');
var fs = require('fs');

var httpserver = http.createServer(function(req,res){
    console.log(req.url);
    if(req.url=="/"){
        fs.readFile("./index.html",(err,data)=>{
            res.write(data);
            res.end();
        })
    }
}).listen(8080,function(){
    console.log((new Date())+'server listening on port 8080');
});
const wss = new WebSocket.Server({server:httpserver})
wss.on("connection",(clientws)=>{
    clientws.send("Hello Client")
    clientws.on("message",(msg)=>{
        conaole.log("Received" +msg)
        clientws.send("Received"+msg)
    })
})
```

Index.html

```
<!DOCKTYPE html>
<html>
    <body>
        <script langauge="javascript">
            var ws = new WebSocket('ws://lialhost:8080');
            ws.addEventListener("message",function(e){
                var msg = e.data;
                document.getElementById('chatlog').innerHTML+='<br>Server:'+ms
g;
            });
            function sendMessage(){
                var message = document.getElementById('message').value;
                document.getElementById('chatlog').innerHTML+='<br>Me:'+messag
e;
                ws.send(message);
                document.getElementById('message').value="";
            }
        </script>
        <h2>
```

```
            Data From Server
        </h2>
        <div id = "chatlog"></div></hr>
        <h2>Data from Client</h2>
        <input type ="text" id = "message"/>
        <input type ="button" id="b1" onclick="sendMessage()" value ="send"/>
    </body>
</html>
```

Output:

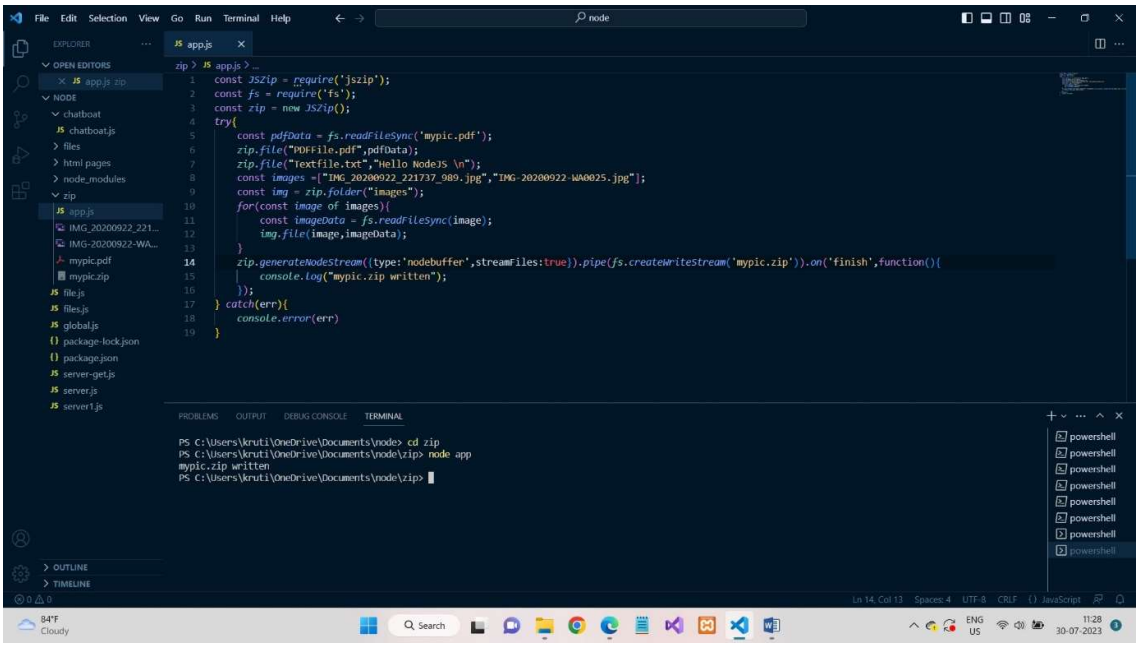


5. Write a program to create a compressed zip file for a folder.

```
const JSZip = require('jszip');
const fs = require('fs');
const zip = new JSZip();
try{
    const pdfData = fs.readFileSync('mypic.pdf');
    zip.file("PDFFile.pdf",pdfData);
    zip.file("Textfile.txt","Hello NodeJS \n");
    const images =["IMG_20200922_221737_989.jpg","IMG-20200922-WA0025.jpg"];
    const img = zip.folder("images");
    for(const image of images){
        const imageData = fs.readFileSync(image);
```

```
        img.file(image,imageData);
    }
    zip.generateNodeStream({type:'nodebuffer',streamFiles:true}).pipe(fs.creat
eWriteStream('mypic.zip')).on('finish',function(){
        console.log("mypic.zip written");
    });
} catch(err){
    console.error(err)
}
```

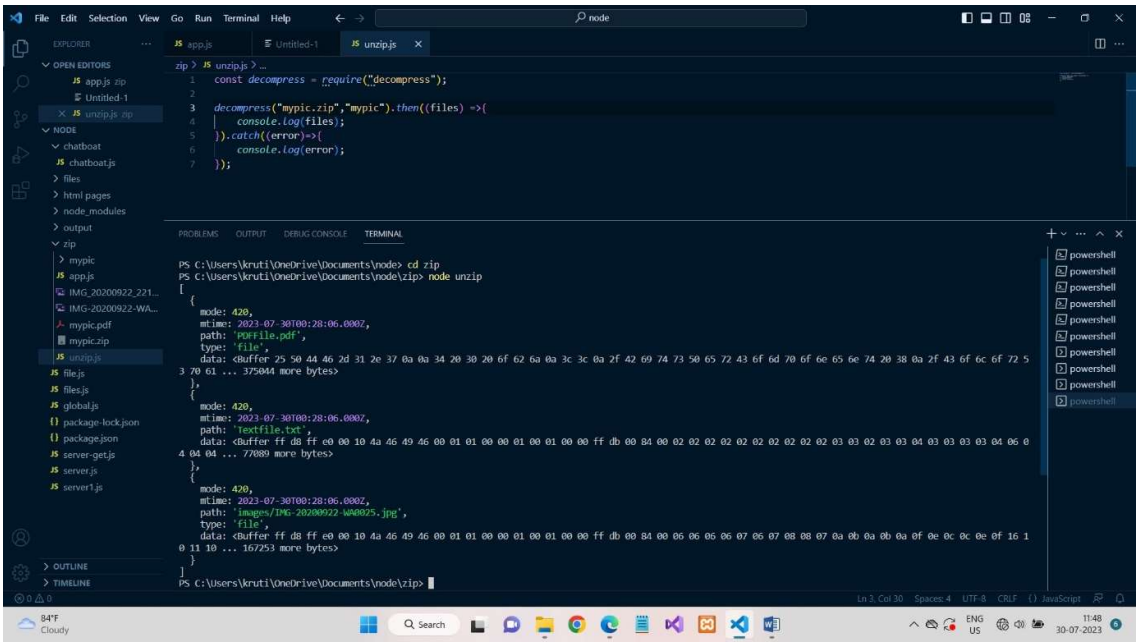
Output:



6. Write a program to extract a zip file.

```
const decompress = require("decompress");

decompress("mypic.zip","mypic").then((files) =>{
    console.log(files);
}).catch((error)=>{
    console.log(error);
});
```

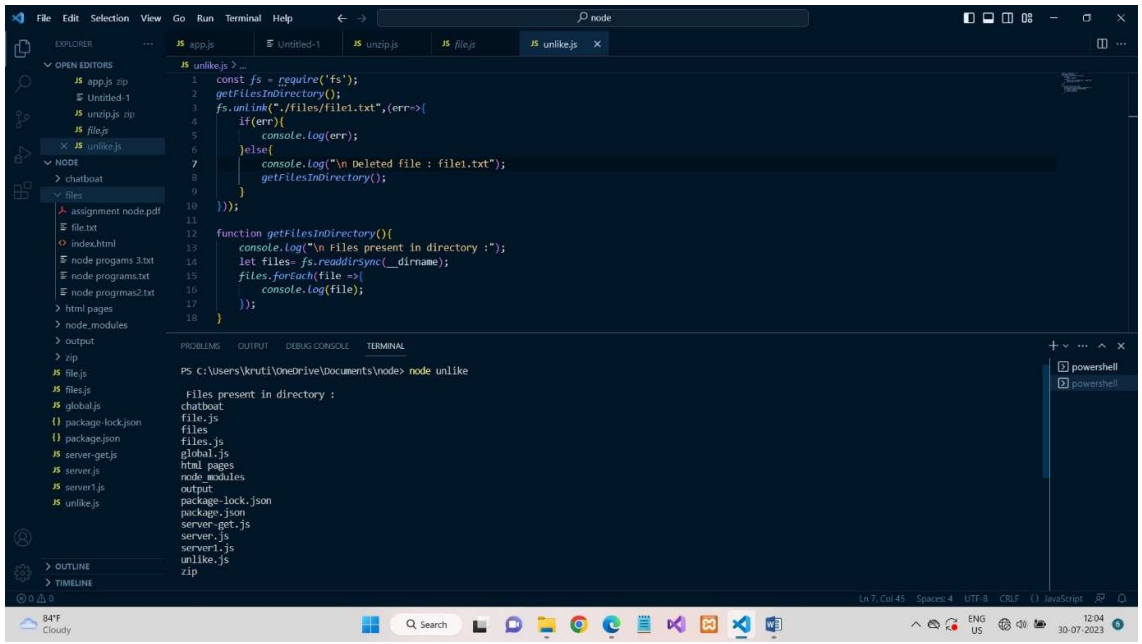
Output:

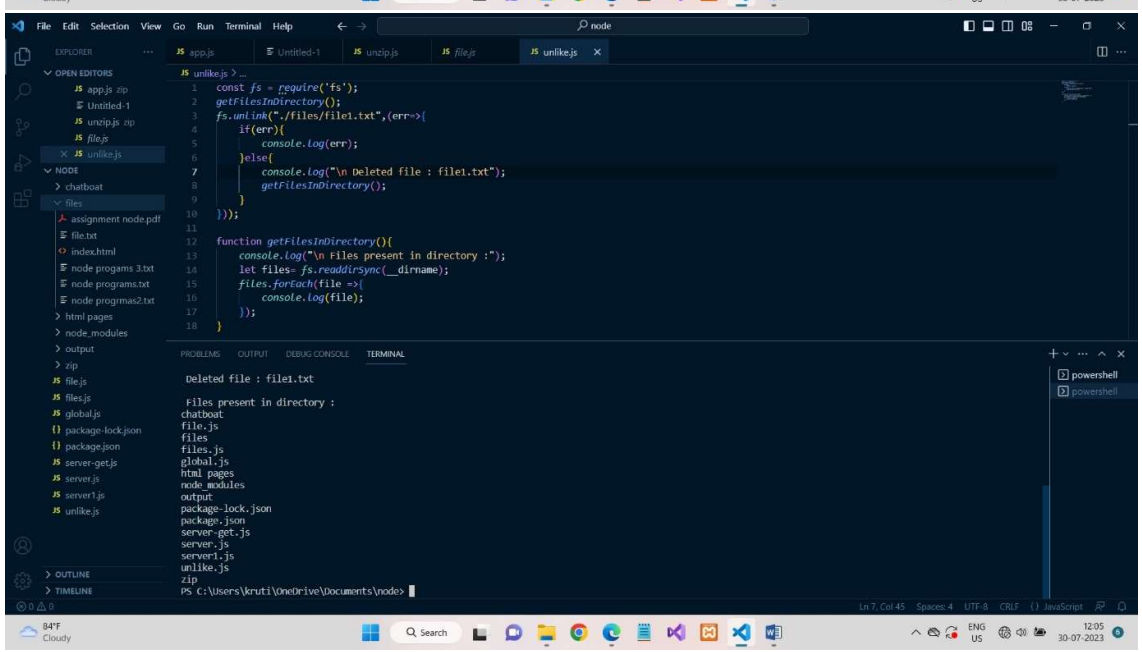


7. Write a program to promisify fs.unlink function and call it.

```
const fs = require('fs');
getFilesInDirectory();
fs.unlink("./files/file1.txt",(err=>{
    if(err){
        console.log(err);
    }else{
        console.log("\n Deleted file : file1.txt");
        getFilesInDirectory();
    }
}));

function getFilesInDirectory(){
    console.log("\n Files present in directory :");
    let files= fs.readdirSync(__dirname);
    files.forEach(file =>{
        console.log(file);
    });
}
```

Output:



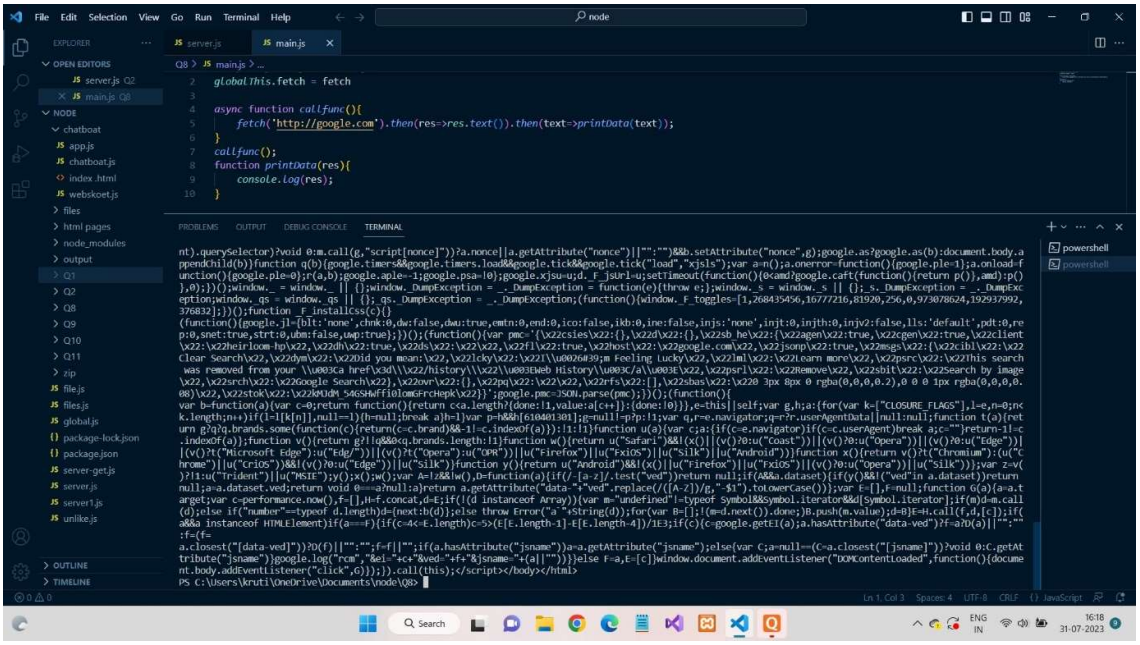


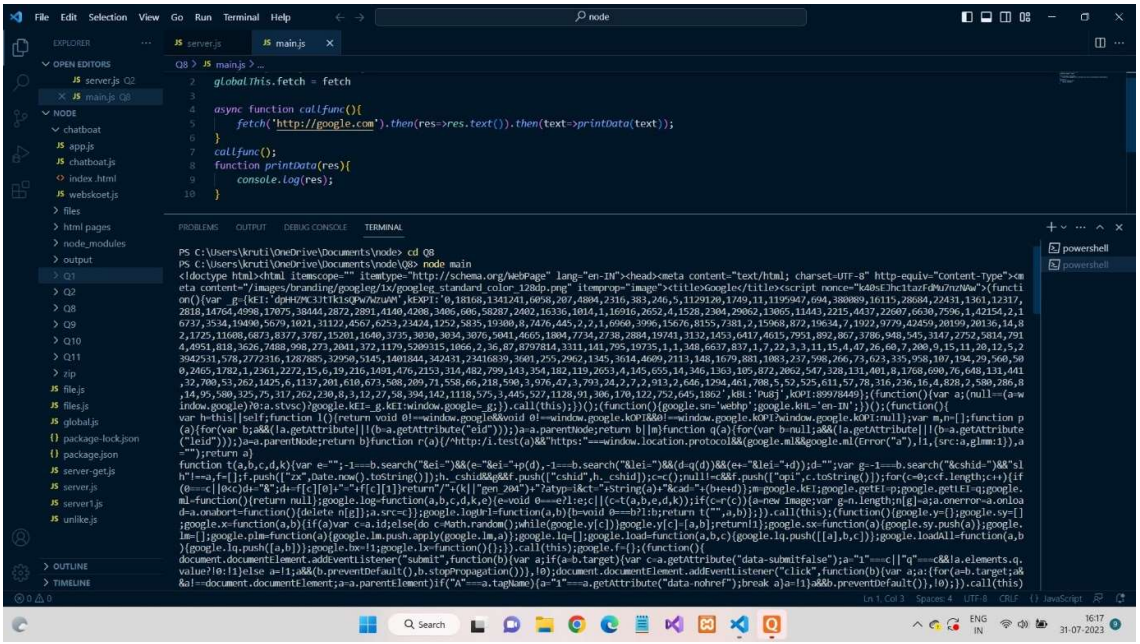
8. Fetch data of google page using note-fetch using async-await model.

```
//import fetch from 'node-fetch'
globalThis.fetch = fetch

async function callfunc(){
    fetch('http://google.com').then(res=>res.text()).then(text=>printData(text
));
}
callfunc();
function printData(res){
```

```
    console.log(res);
}
```

Output:

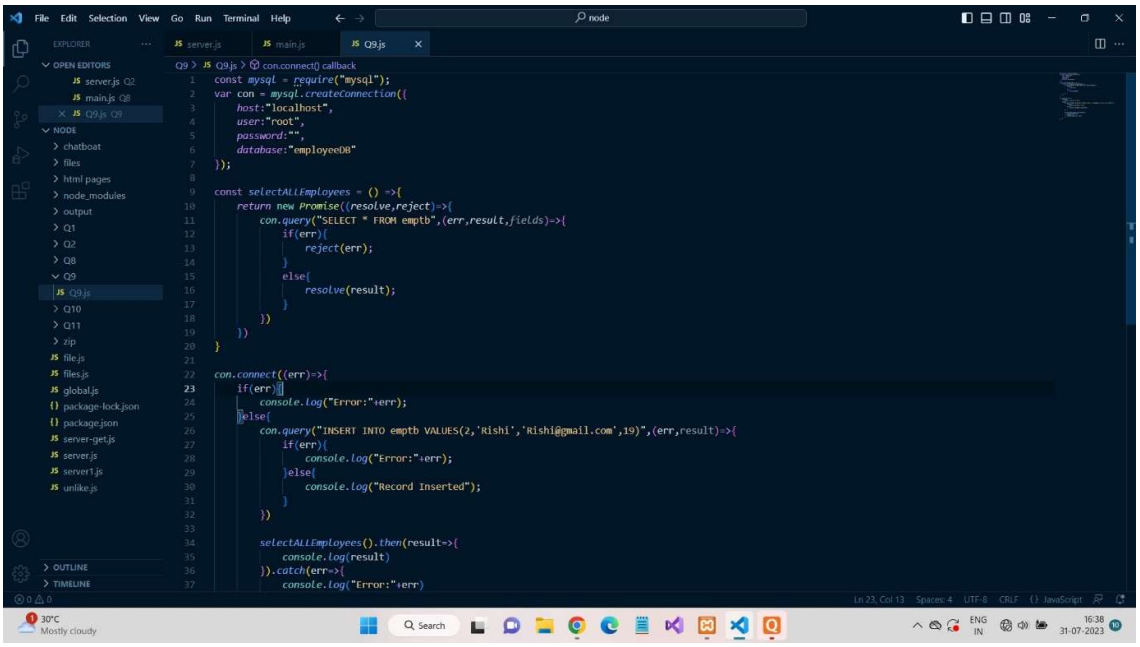




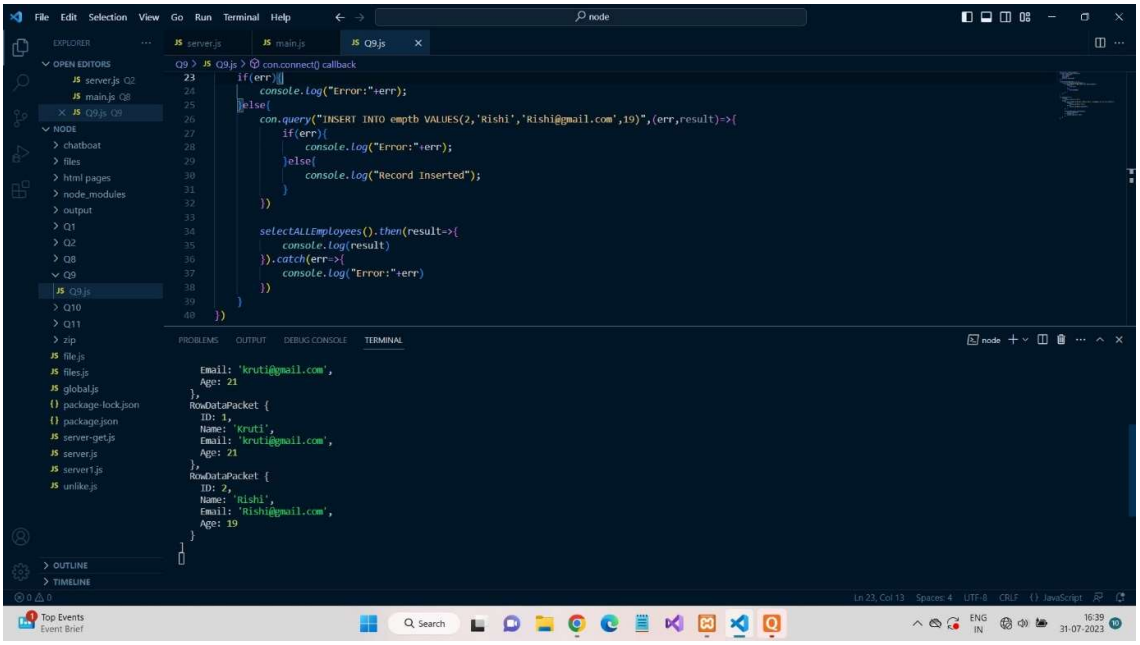
9. Write a program that connect Mysql database, Insert a record in employee table and display all records in employee table using promise based approach.

```
const mysql = require("mysql");
var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"employeeDB"
});

const selectALLEmployees = () =>{
    return new Promise((resolve,reject)=>{
        con.query("SELECT * FROM emptb",(err,result,fields)=>{
            if(err){
                reject(err);
            }
            else{
                resolve(result);
            }
        })
    })
}

con.connect((err)=>{
    if(err){
        console.log("Error:"+err);
    }else{
        con.query("INSERT INTO emptb
VALUES(2,'Rishi','Rishi@gmail.com',19)",(err,result)=>{
            if(err){
                console.log("Error:"+err);
            }else{
                console.log("Record Inserted");
            }
        })

        selectALLEmployees().then(result=>{
            console.log(result)
        }).catch(err=>{
            console.log("Error:"+err)
        })
    }
})
```

Output:



```javascript
const mysql = require("mysql");
var con = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"employeeDB"
});

const selectALLEmployees = () =>{
    return new Promise((resolve,reject)=>{
        con.query("SELECT * FROM emptb",(err,result,fields)=>{
            if(err){
                reject(err);
            }
            else{
                resolve(result);
            }
        })
    })
}

con.connect((err)=>{
    if(err){
        console.log("Error:"+err);
    }else{
        con.query("INSERT INTO emptb VALUES(2,'Rishi','Rishi@gmail.com',19)",(err,result)=>{
            if(err){
                console.log("Error:"+err);
            }else{
                console.log("Record Inserted");
            }
        })

        selectALLEmployees().then(result=>{
            console.log(result)
        }).catch(err=>{
            console.log("Error:"+err)
```



```
    Email: 'kruti@gmail.com',
    Age: 21
  },
  RowDataPacket {
    ID: 1,
    Name: 'Kruti',
    Email: 'kruti@gmail.com',
    Age: 21
  },
  RowDataPacket {
    ID: 2,
    Name: 'Rishi',
    Email: 'Rishi@gmail.com',
    Age: 19
  }
]
```

10. Set a server script, a test script and 3 user defined scripts in package.json file in your nodejs application.

Script 1:

```javascript
console.log("This is the script 1");
```

Script 2:

```
console.log("This is the script 2");
```

Script 3:

```
console.log("This is the script 3");
```

Test:

```
console.log("This is a test  script !!");
```

package.json :

```
{
  "name": "node",
  "version": "1.0.0",
  "main": "file.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js",
    "test1": "test.js",
    "UD script1": "script1.js",
    "UD script2": "script2.js",
    "UD script3": "script3.js",
    "server": "Q10.js"
  },
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "decompress": "^4.2.1",
    "jszip": "^3.10.1",
    "mysql": "^2.18.1",
    "node-fetch": "^3.3.2",
    "package-lock.json": "^1.0.0",
    "package.json": "^2.0.1",
    "pakage.json": "^1.0.0",
    "ws": "^8.13.0"
  }
}
```