Name	: Vivek Lakhlani
Roll no	: 025
Subject	: Applicatipon developed using fullstack (705)
Semester	: 7th
Division	: A
Date	: 30/7/2023

Practical Assignment : 1

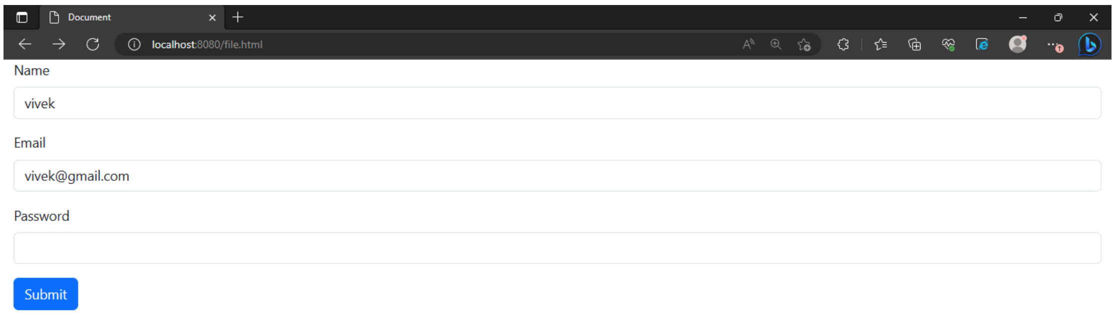
GITHUB LINK :
https://github.com/Vivek2425/Practical_Assignment_1_25_705.git

1. Develop a web server with following functionalities:
- Serve static resources.
- Handle GET request.
- Handle POST request.

Program :

```
const http = require('http');
const url = require('url');
const fs = require('fs');
const static = require('node-static');
var fileserver = new static.Server('files');
const server = http.createServer((req,res)=>{
    var url1 = url.parse(req.url, true);
    if(url1.pathname=="/file.html"){

        fileserver.serve(req,res);
    }else if(url1.pathname=="/index.html"){
        fileserver.serve(req,res);
    }else if(url1.pathname=="/processData" && req.method=="GET"){
        res.write("Name : " + url1.query.name + " Email : " +
url1.query.email + " Passowrd : " + url1.query.password);
        res.end()
    }else if(url1.pathname=="/processData" && req.method=="POST"){
        let body = '';
        req.on("data" ,chunk=>{
            body += chunk.toString();
        })
        req.on("end" ,()=>{
            res.end('ok => '+ body)
        })
    }else{
        res.end("get lost");
    }

    // req.on("/",()=>{
    //     console.log("hello world")
    // })
})
server.listen(8080,()=>{
    console.log("Server starts on port 8080 port:
http://localhost:8080")
})
```

Vivek lakhlani 025



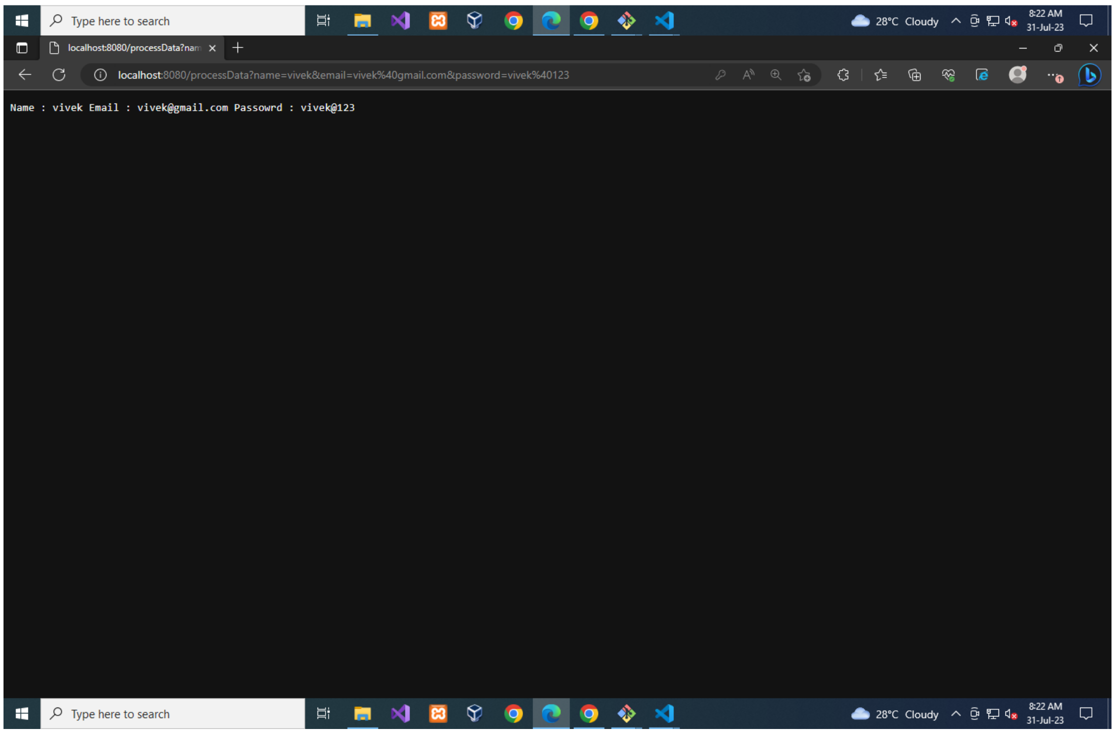
Name

vivek

Email

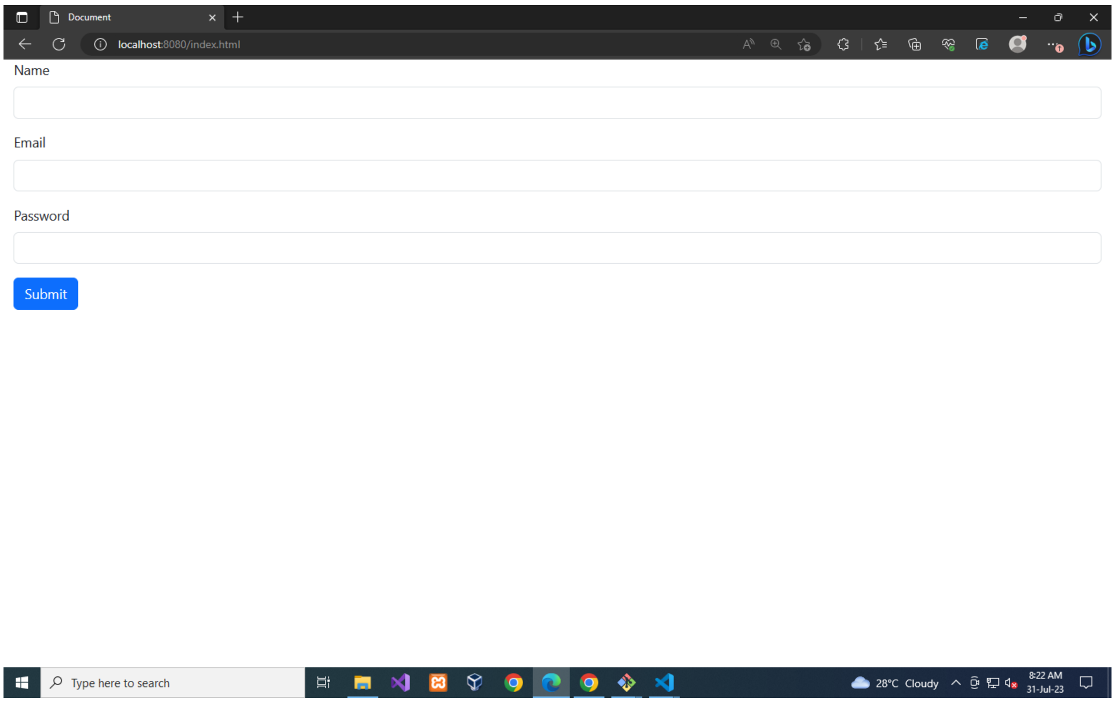
vivek@gmail.com

Password


Type here to search
localhost:8080/processData?nam
localhost:8080/processData?name=vivek&email=vivek%40gmail.com&password=vivek%40123
28°C Cloudy
8:22 AM
31-Jul-23

```
Name : vivek Email : vivek@gmail.com Passowrd : vivek@123
```

Vivek lakhlani 025

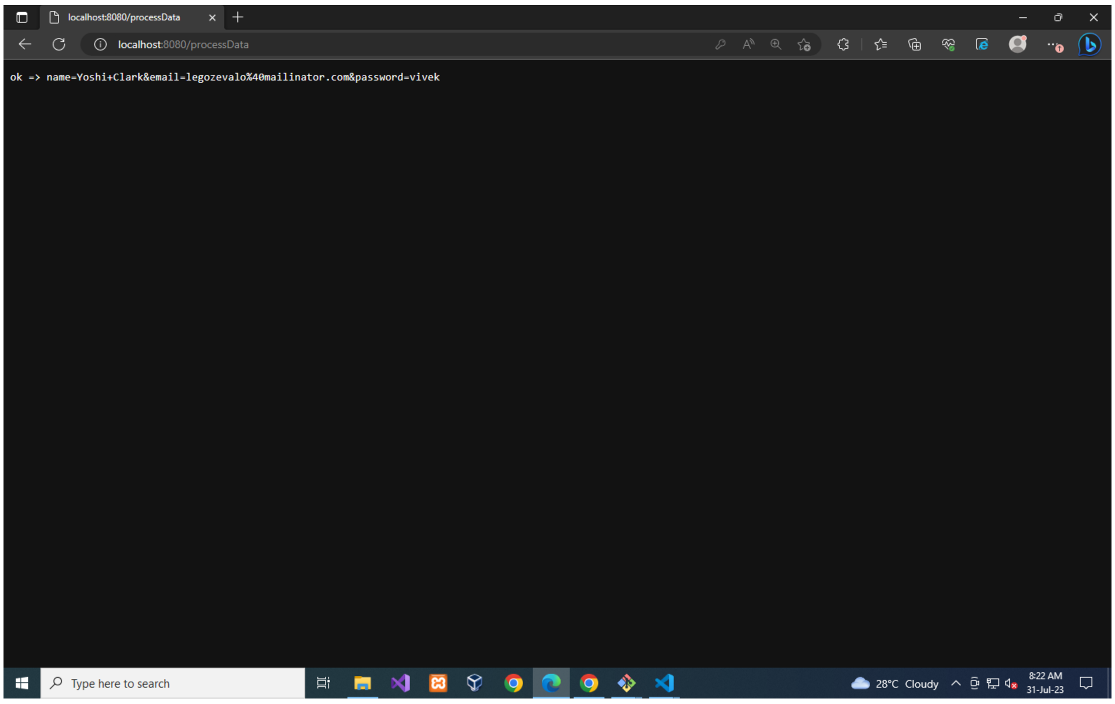


Name

Email

Password

Submit



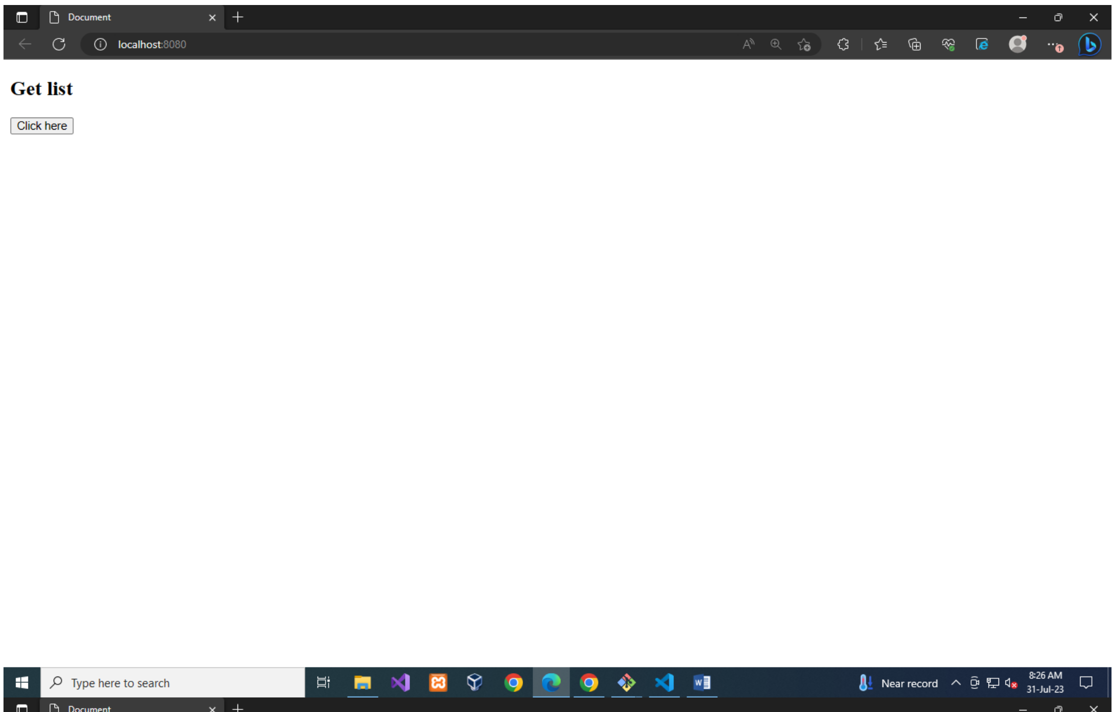
```
ok => name=Yoshi+Clark&email=legozevalo%40mailinator.com&password=vivek
```

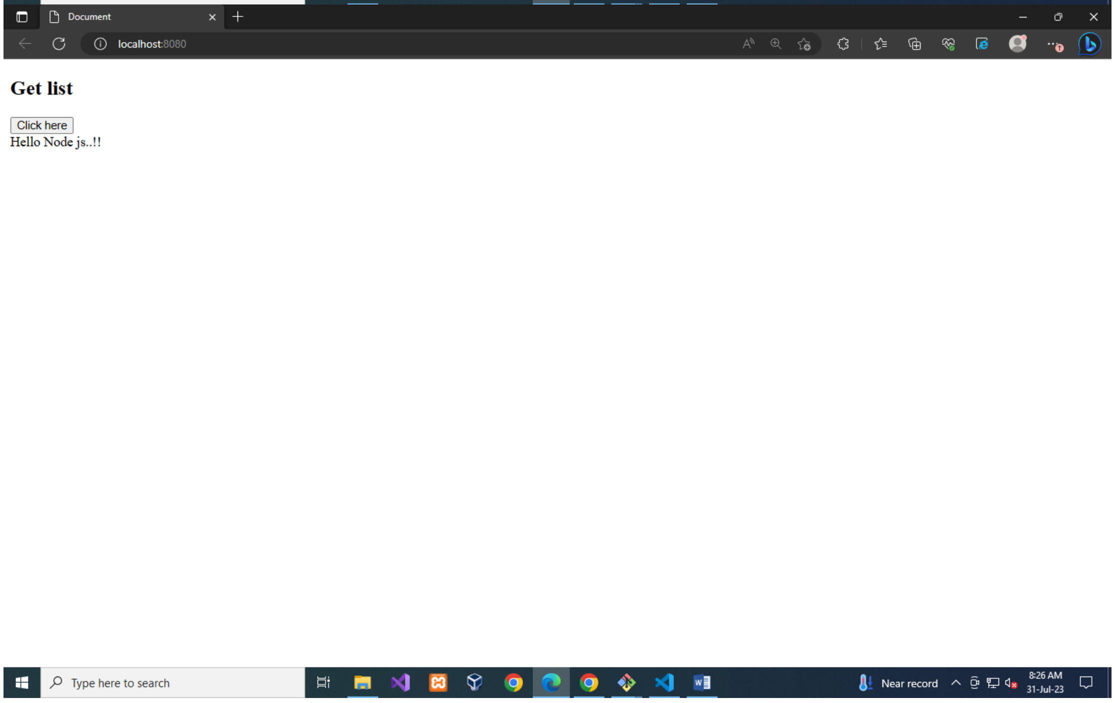
2. Develop nodejs application with following requirements:
- Develop a route "/gethello" with GET method. It displays "Hello NodeJS!!" as response.
- Make an HTML page and display.
- Call "/gethello" route from HTML page using AJAX call. (Any frontend AJAX call API can be
used.)

Program :

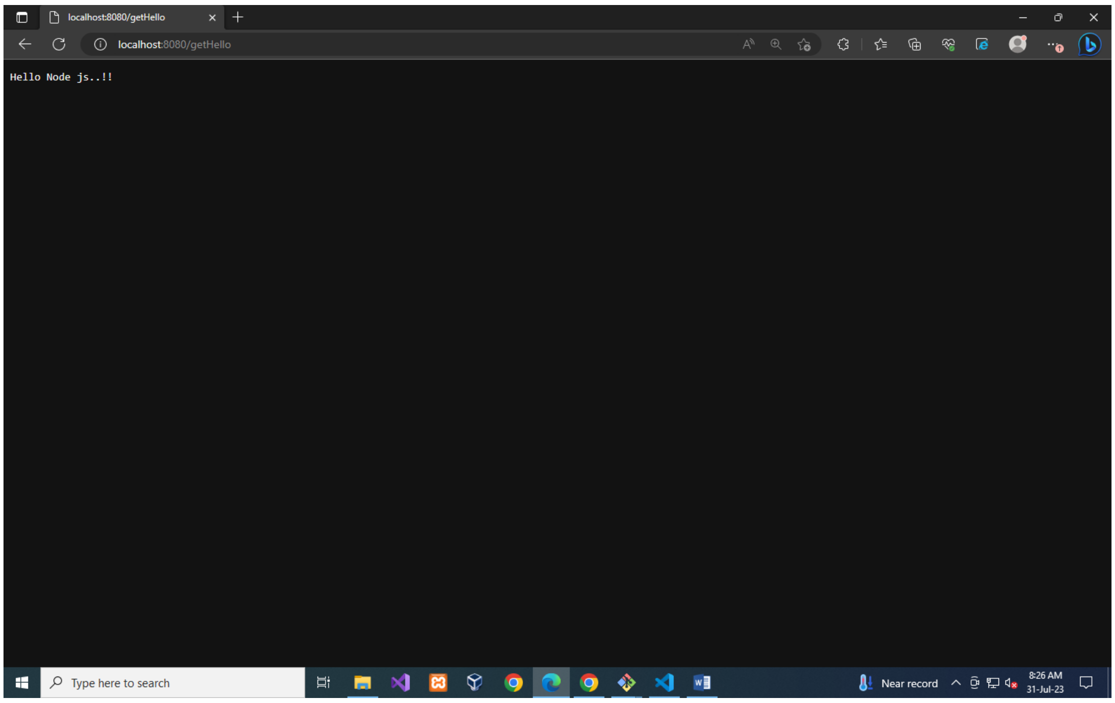
```
const http = require('http');
const url = require('url');
const static = require('node-static')
// var fileserver = static.Server('static')
// const server = http.createServer((req,res)=>{
//     var url2 = url.parse(req.url,true);
//     if(url2.pathname=="/"){
//         fileserver.serve(req,res);
//     }else if(url2.pathname == "/getHello"){
//         res.end("Hello Node js..!!");
//     }else{
//         res.end("Do nothing")
//     }
// })
// server.listen(8080,()=>{
//     console.log("http://localhost:8080");
// })

var fileserver =new static.Server('static');

const server = http.createServer((req, res) => {
   var url2 = url.parse(req.url, true);
   if (url2.pathname == "/") {
      fileserver.serve(req, res);
   } else if (url2.pathname == "/getHello" && req.method=="GET") {
      res.end("Hello Node js..!!");
      // res.end();
   } else {
      res.end("Do nothing");
   }
});

server.listen(8080, () => {
    console.log("http://localhost:8080");
 });
```

Vivek lakhlani 025


Get list


Get list

Hello Node js..!!



```
Hello Node js..!!
```

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

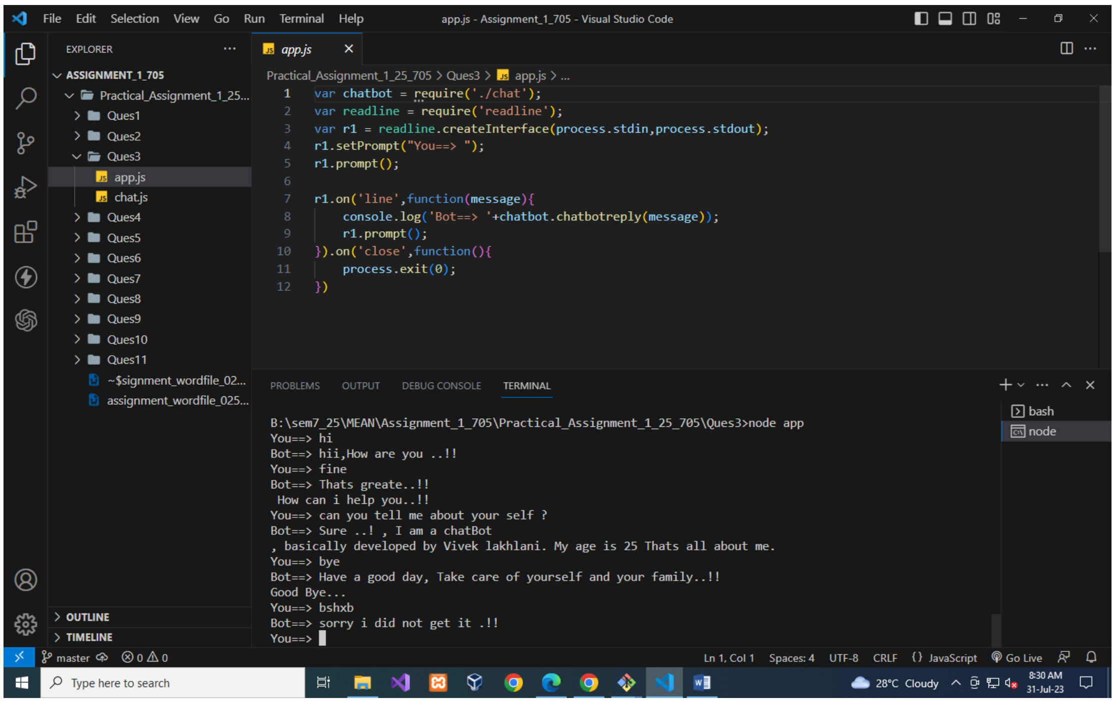
Program :

App.js

```
var chatbot = require('./chat');
var readline = require('readline');
var r1 = readline.createInterface(process.stdin,process.stdout);
r1.setPrompt("You==> ");
r1.prompt();

r1.on('line',function(message){
    console.log('Bot==> '+chatbot.chatbotreply(message));
    r1.prompt();
}).on('close',function(){
    process.exit(0);
})
```

Chat.js

```
module.exports.chatbotreply = function(message){
    this.age = 25;
    this.name = 'chatGpt';
    this.university = 'vnsgu';
    this.country = 'india';
    message = message.toLowerCase()

    if(message.indexOf('hi')  > -1){
        return "hii,How are you ..!!";
    }else if(message.indexOf('fine')  > -1){
        return "Thats greate..!!\n How can i help you..!!";
    }else if(message.indexOf('can you tell me about your self ?')  > -
1){
        return "Sure ..! , I am a chatBot \n, basically developed by
Vivek lakhlani. My age is "+this.age  + " Thats all about me.";
    }else if(message.indexOf('bye')  > -1){
        return "Have a good day, Take care of yourself and your
family..!!\nGood Bye...";
    }else{
        return "sorry i did not get it .!!";
    }
}
```



```
var chatbot = require('./chat');
var readline = require('readline');
var r1 = readline.createInterface(process.stdin,process.stdout);
r1.setPrompt("You==> ");
r1.prompt();

r1.on('line',function(message){
    console.log('Bot==> '+chatbot.chatbotreply(message));
    r1.prompt();
}).on('close',function(){
    process.exit(0);
})
```

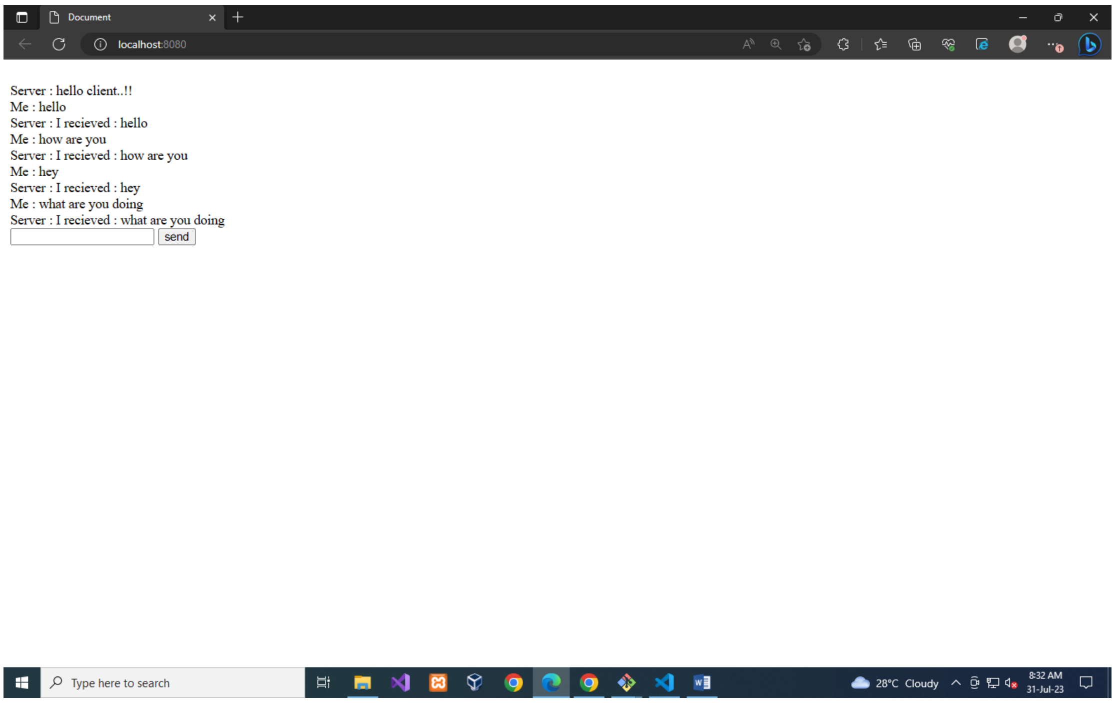
```
B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques3>node app
You==> hi
Bot==> hii,How are you ..!!
You==> fine
Bot==> Thats greate..!!
 How can i help you..!!
You==> can you tell me about your self ?
Bot==> Sure ..! , I am a chatBot
, basically developed by Vivek lakhlani. My age is 25 Thats all about me.
You==> bye
Bot==> Have a good day, Take care of yourself and your family..!!
Good Bye...
You==> bshxb
Bot==> sorry i did not get it .!!
You==> 
```

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

Program :

```
const websocket = require('ws');
const http = require('http');
const url = require('url');

const st = require('node-static');

const fileserver = new st.Server('./public');

const httpserver = http.createServer((req,res)=>{
    req.on('end',()=>{
        var get  = url.parse(req.url,true).query;
        fileserver.serve(req,res);
    }).resume();
}).listen(8080,()=>{
    console.log("http://localhost:8080");
})

const wss = new websocket.Server({server:httpserver});

wss.on('connection',(ws)=>{
    ws.send("hello client..!!");

    ws.on('message',messgae=>{
        ws.send('I recieved : ' + messgae)
    })
})
```

Vivek lakhlani 025

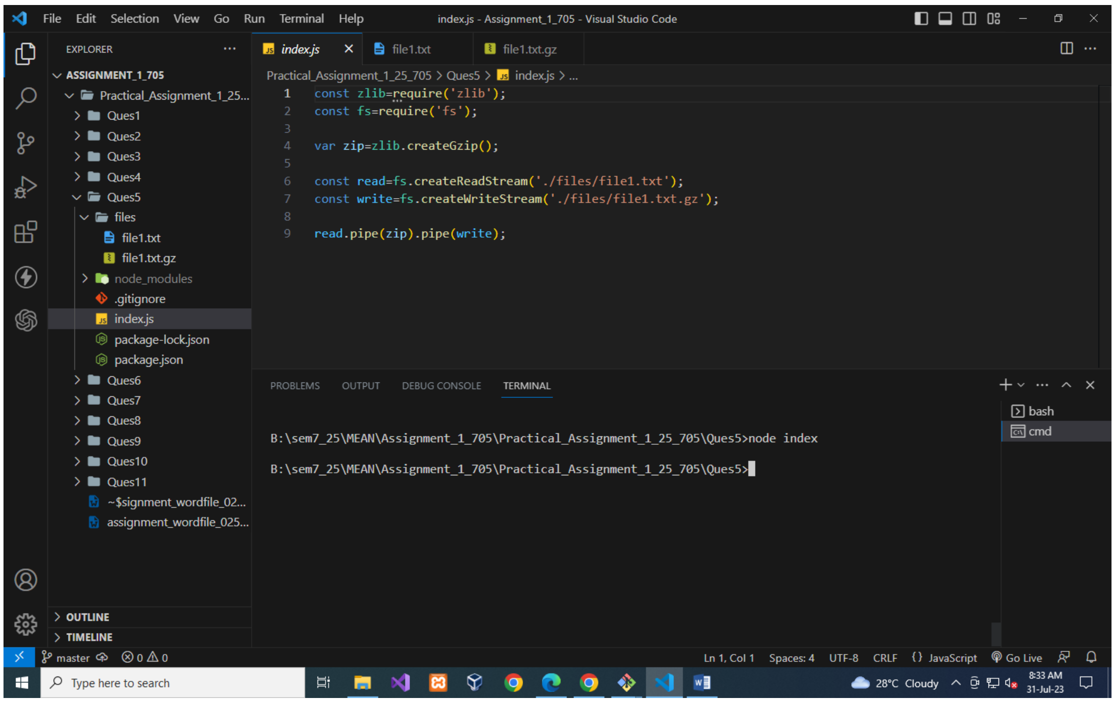


```
Server : hello client..!!
Me : hello
Server : I recieved : hello
Me : how are you
Server : I recieved : how are you
Me : hey
Server : I recieved : hey
Me : what are you doing
Server : I recieved : what are you doing
```

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

Program :

```
const zlib=require('zlib');
const fs=require('fs');

var zip=zlib.createGzip();

const read=fs.createReadStream('./files/file1.txt');
const write=fs.createWriteStream('./files/file1.txt.gz');

read.pipe(zip).pipe(write);
```



```
const zlib=require('zlib');
const fs=require('fs');

var zip=zlib.createGzip();

const read=fs.createReadStream('./files/file1.txt');
const write=fs.createWriteStream('./files/file1.txt.gz');

read.pipe(zip).pipe(write);
```

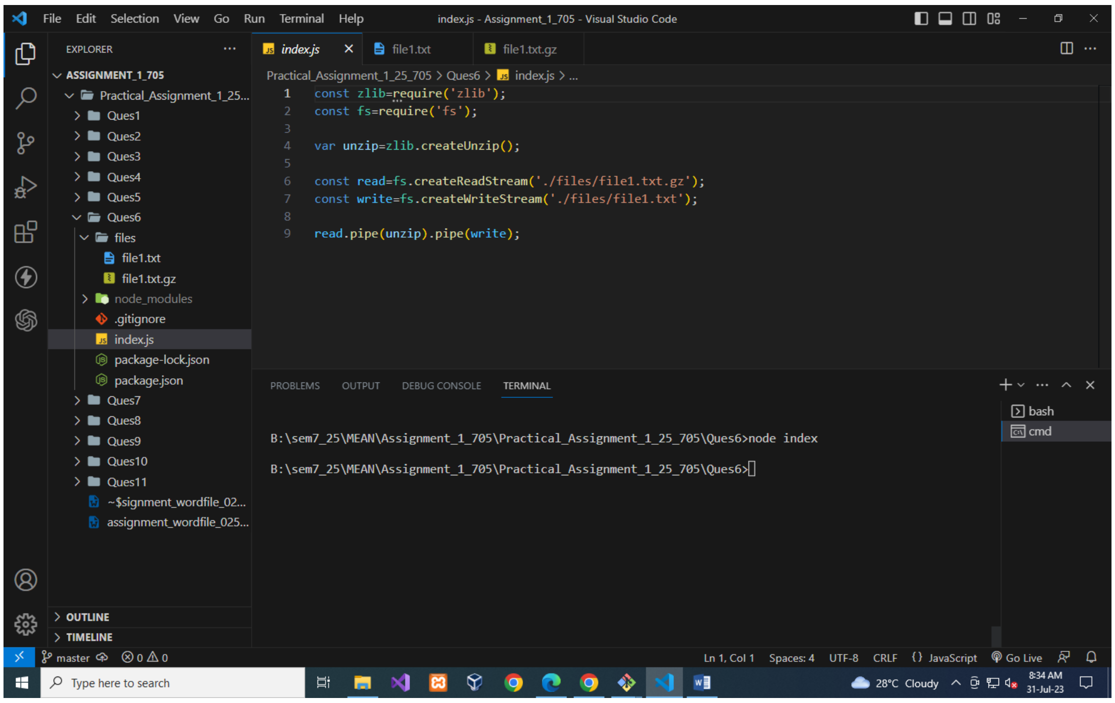
```
B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques5>node index

B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques5>
```

6. Write a program to extract a zip file.

Program :

```
const zlib=require('zlib');
const fs=require('fs');

var unzip=zlib.createUnzip();

const read=fs.createReadStream('./files/file1.txt.gz');
const write=fs.createWriteStream('./files/file1.txt');

read.pipe(unzip).pipe(write);
```

Vivek lakhlani 025



```
const zlib=require('zlib');
const fs=require('fs');

var unzip=zlib.createUnzip();

const read=fs.createReadStream('./files/file1.txt.gz');
const write=fs.createWriteStream('./files/file1.txt');

read.pipe(unzip).pipe(write);
```

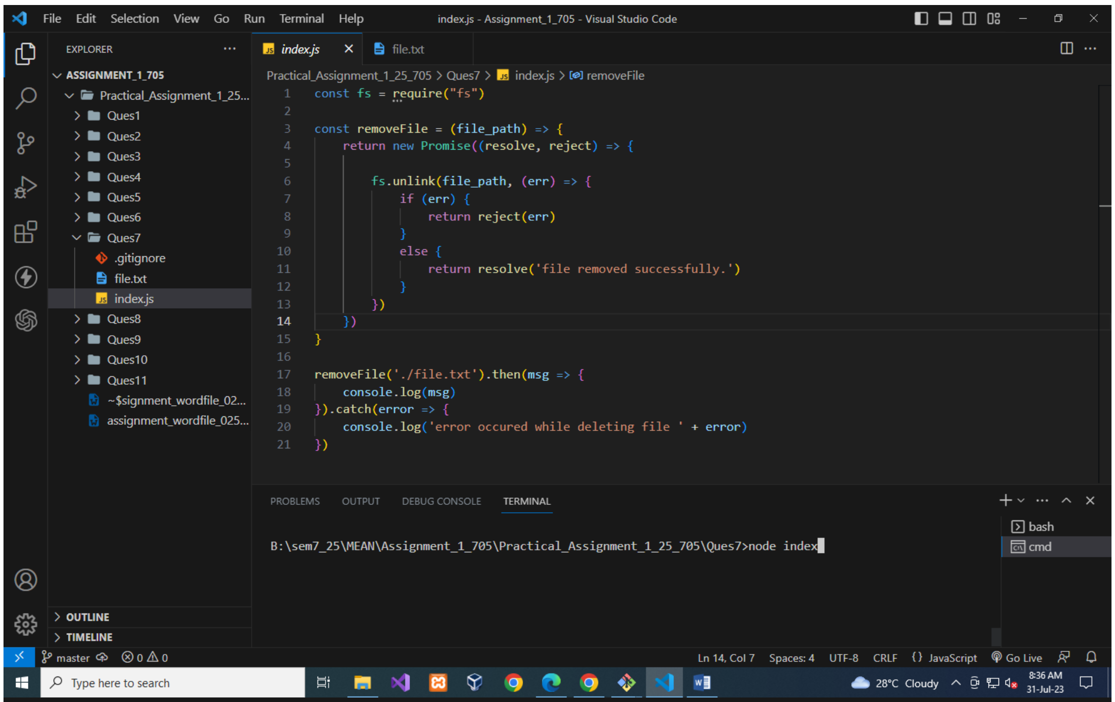
```
B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques6>node index

B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques6>
```

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

Program :

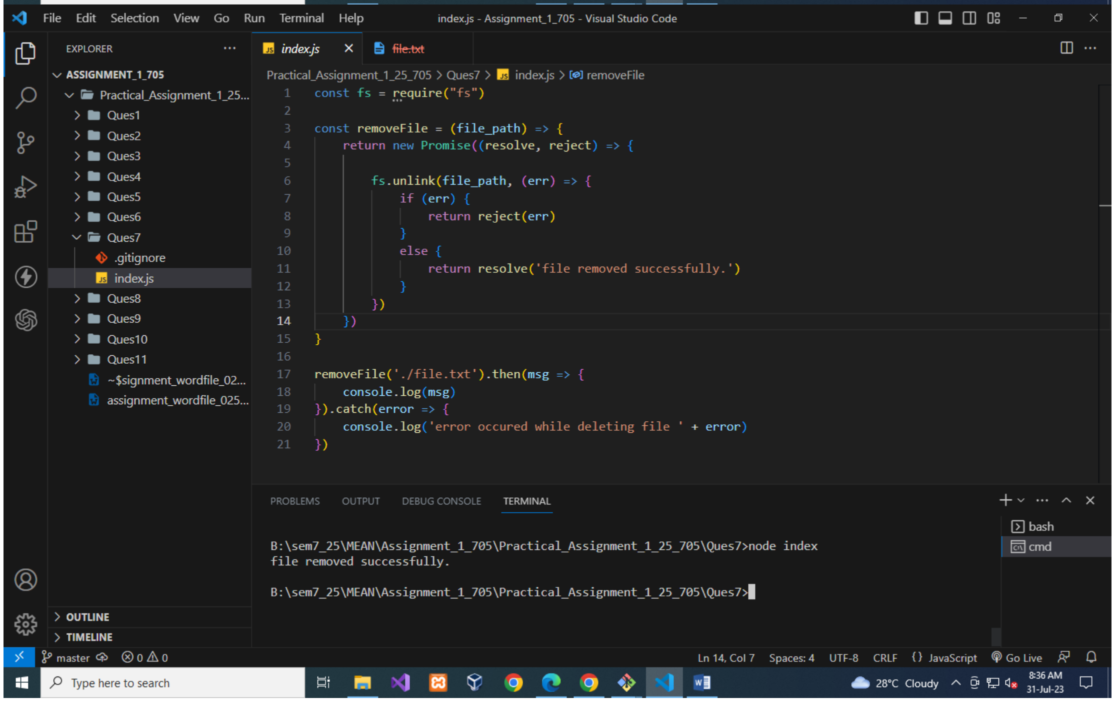
```
const fs = require("fs")

const removeFile = (file_path) => {
    return new Promise((resolve, reject) => {

        fs.unlink(file_path, (err) => {
            if (err) {
                return reject(err)
            }
            else {
                return resolve('file removed successfully.')
            }
        })
    })
}

removeFile('./file.txt').then(msg => {
    console.log(msg)
}).catch(error => {
    console.log('error occured while deleting file ' + error)
})
```

Vivek lakhlani 025



```javascript
const fs = require("fs")

const removeFile = (file_path) => {
    return new Promise((resolve, reject) => {

        fs.unlink(file_path, (err) => {
            if (err) {
                return reject(err)
            }
            else {
                return resolve('file removed successfully.')
            }
        })
    })
}

removeFile('./file.txt').then(msg => {
    console.log(msg)
}).catch(error => {
    console.log('error occured while deleting file ' + error)
})
```

```
B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques7>node index
```



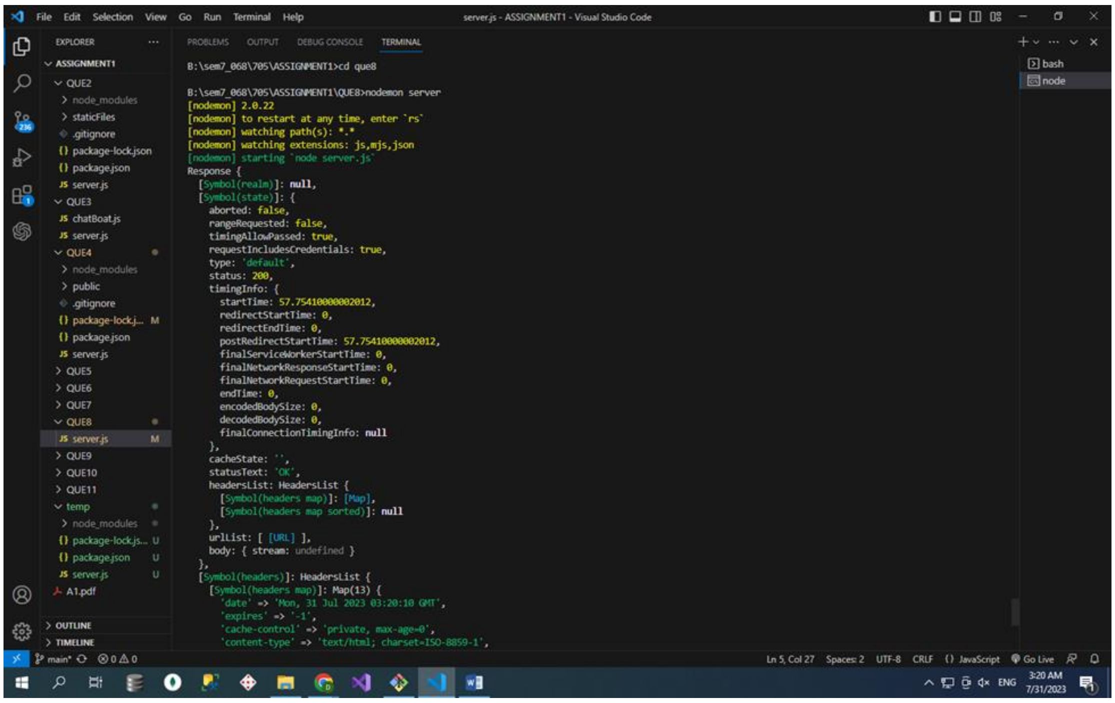
```javascript
const fs = require("fs")

const removeFile = (file_path) => {
    return new Promise((resolve, reject) => {

        fs.unlink(file_path, (err) => {
            if (err) {
                return reject(err)
            }
            else {
                return resolve('file removed successfully.')
            }
        })
    })
}

removeFile('./file.txt').then(msg => {
    console.log(msg)
}).catch(error => {
    console.log('error occured while deleting file ' + error)
})
```

```
B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques7>node index
file removed successfully.

B:\sem7_25\MEAN\Assignment_1_705\Practical_Assignment_1_25_705\Ques7>
```

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

Program :

```
(async () => {
  try {
    const response = await fetch("https://www.google.com/");
    const text = await response.text();
    console.log(text);
  } catch (error) {
    console.log(error.response.body);
  }
})();
```



```
B:\sem7_068\705\ASSIGNMENT1>cd que8

B:\sem7_068\705\ASSIGNMENT1\QUE8>nodemon server
[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Response {
  [Symbol(realm)]: null,
  [Symbol(state)]: {
    aborted: false,
    rangeRequested: false,
    timingAllowPassed: true,
    requestIncludesCredentials: true,
    type: 'default',
    status: 200,
    timingInfo: {
      startTime: 57.75410000002012,
      redirectStartTime: 0,
      redirectEndTime: 0,
      postRedirectStartTime: 57.75410000002012,
      finalServiceWorkerStartTime: 0,
      finalNetworkResponseStartTime: 0,
      finalNetworkRequestStartTime: 0,
      endTime: 0,
      encodedBodySize: 0,
      decodedBodySize: 0,
      finalConnectionTimingInfo: null
    },
    cacheState: '',
    statusText: 'OK',
    headersList: HeadersList {
      [Symbol(headers map)]: [Map],
      [Symbol(headers map sorted)]: null
    },
    urlList: [ [URL] ],
    body: { stream: undefined }
  },
  [Symbol(headers)]: HeadersList {
    [Symbol(headers map)]: Map(13) {
      'date' => 'Mon, 31 Jul 2023 03:20:10 GMT',
      'expires' => '-1',
      'cache-control' => 'private, max-age=0',
      'content-type' => 'text/html; charset=ISO-8859-1',
```

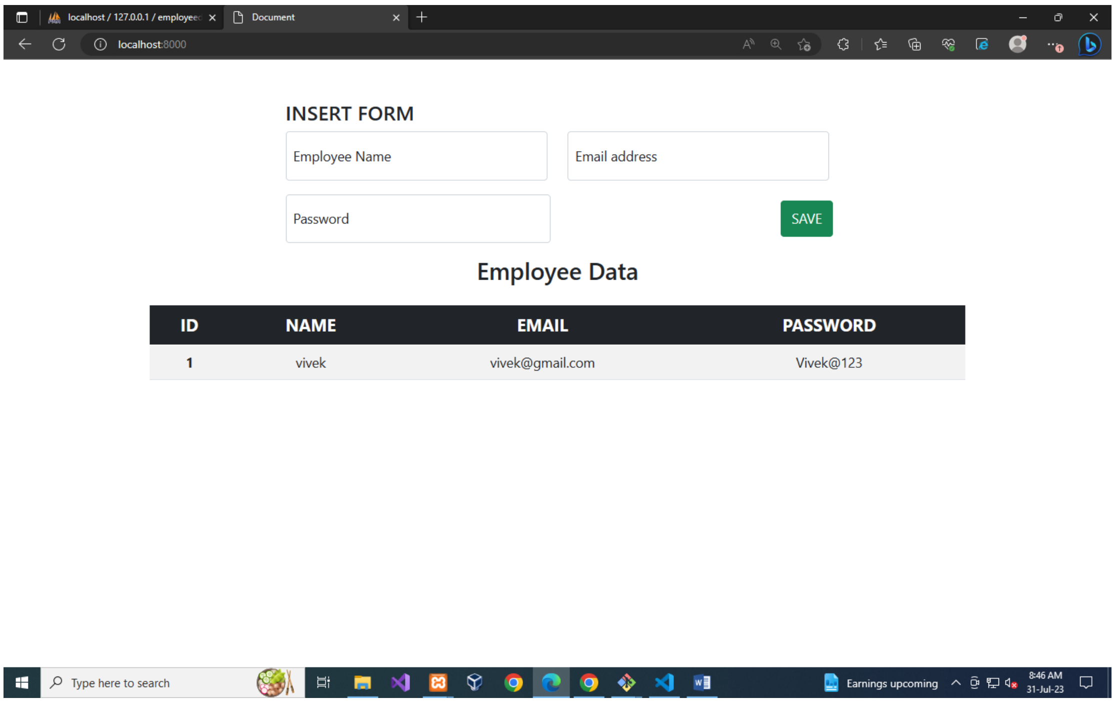
9. Write a program that connect Mysql database, Insert a record in employee table and
display all records in employee table using promise based approach.

Program :

```
const http=require("http");
const mysql=require("mysql");
const static=require("node-static");

var fileserver=new static.Server("./public");

var conn=mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"root",
    database:"employeedb"
});
conn.connect((err)=>{
    if(err){
        console.log(err);
    }else{
        console.log("connected")
    }
})

async function getData(){

}
var server=http.createServer((req,res)=>{
    console.log(req.url);
    if(req.url=="/"){
        fileserver.serve(req,res);
    }
    if(req.url=="/getData"){
        conn.query("SELECT * FROM `emptb`",(err,data)=>{
            if(err){
                return "err";
            }
            res.end(JSON.stringify(data));
        })
    }
    if(req.url=="/insert_emp_data" && req.method==="POST"){
        let data = '';
        req.on('data', (chunk) => {
            data += chunk;
        });
        req.on("end",()=>{
            var fd=JSON.parse(data);
```

Vivek lakhlani 025

```
            // console.log(fd.name)
            var sql=`INSERT INTO emptb(emp_name, emp_email, emp_pwd)
VALUES ('${fd.ename}','${fd.eEmail}','${fd.epwd}')`;
            conn.query(sql,(err,data)=>{
                if(err){
                    console.log(err);
                }else{
                    res.end("success");
                }
            })
        })
        // res.end();
    }
})

server.listen(8000,()=>{
    console.log("server listening on port 8000");
})
```



Employee Data

ID	NAME	EMAIL	PASSWORD
1	vivek	vivek@gmail.com	Vivek@123

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

Program :

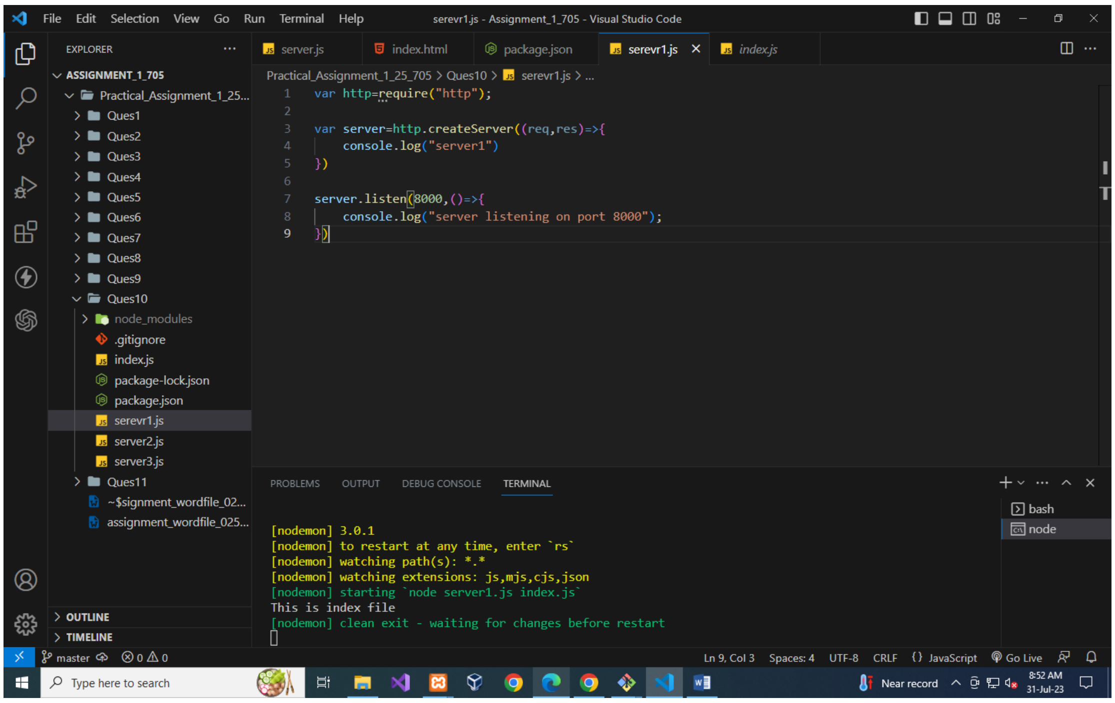
server 1:

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

var server=http.createServer((req,res)=>{
    console.log("server1")
})

server.listen(8000,()=>{
    console.log("server listening on port 8000");
})
Server2 :
var http=require("http");

var server=http.createServer((req,res)=>{
    console.log("server2")
})

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

server3 :

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

var server=http.createServer((req,res)=>{
    console.log("server3")
})

server.listen(3000,()=>{
    console.log("server listening on port 3000");
})
```

Vivek lakhlani 025

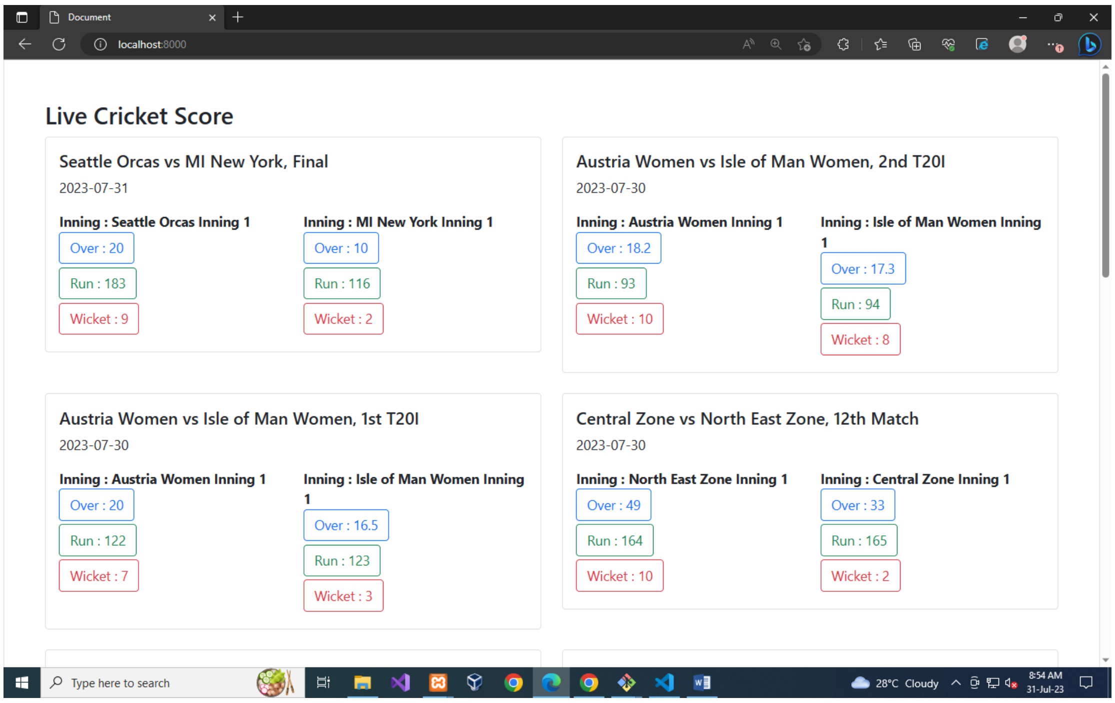

```
var http=require("http");

var server=http.createServer((req,res)=>{
    console.log("server1")
})

server.listen(8000,()=>{
    console.log("server listening on port 8000");
})
```

```
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server1.js index.js`
This is index file
[nodemon] clean exit - waiting for changes before restart
```

11. Develop an application to show live cricket score.

Program :

```
const axios = require("axios");
const http = require("http");
const static = require("node-static");
const url = require("url");
const websocket = require("ws");

var fileServer = new static.Server("./public");
var server = http.createServer((req, res) => {
    fileServer.serve(req, res);
})

var latestData = null;

server.listen(8000, () => {
    console.log("server listening on port 8000");
})

async function fetchMatchScore() {
    try {
        var response = await
axios.get("https://api.cricapi.com/v1/currentMatches?apikey=0bf9e0f5-
5333-4925-912f-5a5511d62c19&offset=0");
        return response.data;
    } catch (err) {
        console.log(err)
    }
}

var wss = new websocket.Server({ server: server });
wss.on("connection", async (ws) => {
    var data = await fetchMatchScore();
    ws.send(JSON.stringify(data));
})

async function updateDataAndBroadcast() {
    latestData = await fetchMatchScore();
    if (latestData !== null) {
        wss.clients.forEach((client) => {
            if (client.readyState === websocket.OPEN) {
                client.send(JSON.stringify(latestData));
            }
        });
    }
}

setInterval(updateDataAndBroadcast, 5000);
```



Live Cricket Score

Seattle Orcas vs MI New York, Final
2023-07-31

Inning : Seattle Orcas Inning 1
Over : 20
Run : 183
Wicket : 9

Inning : MI New York Inning 1
Over : 10
Run : 116
Wicket : 2

Austria Women vs Isle of Man Women, 2nd T20I
2023-07-30

Inning : Austria Women Inning 1
Over : 18.2
Run : 93
Wicket : 10

Inning : Isle of Man Women Inning 1
Over : 17.3
Run : 94
Wicket : 8

Austria Women vs Isle of Man Women, 1st T20I
2023-07-30

Inning : Austria Women Inning 1
Over : 20
Run : 122
Wicket : 7

Inning : Isle of Man Women Inning 1
Over : 16.5
Run : 123
Wicket : 3

Central Zone vs North East Zone, 12th Match
2023-07-30

Inning : North East Zone Inning 1
Over : 49
Run : 164
Wicket : 10

Inning : Central Zone Inning 1
Over : 33
Run : 165
Wicket : 2