ASSIGNMENT:01

Name : Belim Atiya Mohammed Shoeb

Roll no. : 002

Sem : 7th (M.Sc(IT))

Subject : 701-Application Development using full stack

Q-1: Develop a web server with following functionalities:

- Serve static resources.

- Handle GET request.

- Handle POST request.

Answer : 01

- **Q-1.js**

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

const static = require("node-static");

const fileserver = new static.Server('./files');

const server = http.createServer(
  function (req, res) {
    console.log("requsted url :" + req.url);
    var url1 = url.parse(req.url, true);
    if (url1.pathname == "/Q-1.html") {

      fs.readFile("./files/Q-1.html", (err, data) => {
        if (err) {
```

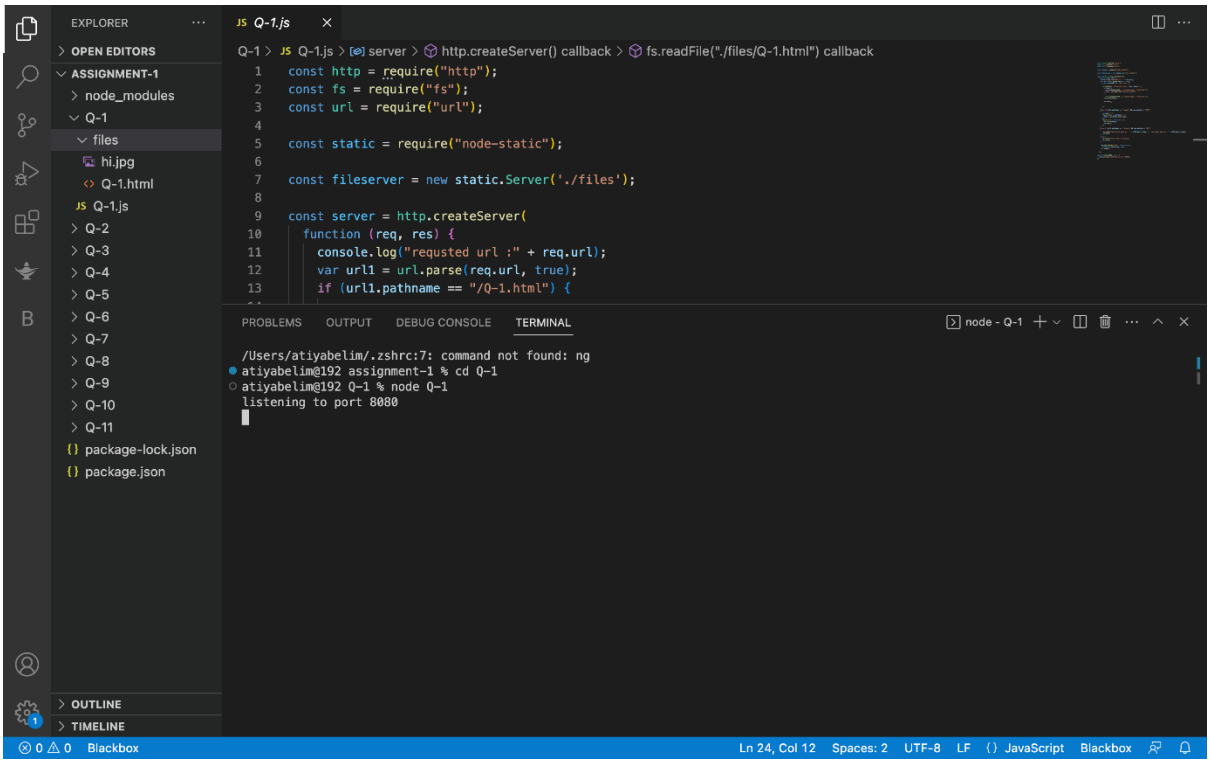
```
            res.writeHead(404, { 'content-type': "text/html" })
           return  res.end("404 file not found")
          }

            res.writeHead(200, { 'content-type': "text/html" })
          res.write(data);

          res.end();


        });
      }
      else if (url1.pathname == "/submit" && req.method == "POST")
     {
        let body = '';
        req.on('data', postdata => {
          body += postdata.toString();
        });
        req.on('end', function () {
          res.write(body);
          res.end();
        });
      }
      else if (url1.pathname == "/process" && req.method == "GET")
     {
        res.write("your first name is :  " + url1.query.fname + "  your
last name is : " + url1.query.lname);
        res.end();
      }
      else {
        res.write("error while running");
        res.end();
      }

      req.addListener('end', function () {
        fileserver.serve(req, res);
      }).resume();

    });

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

- **Q-1.html**

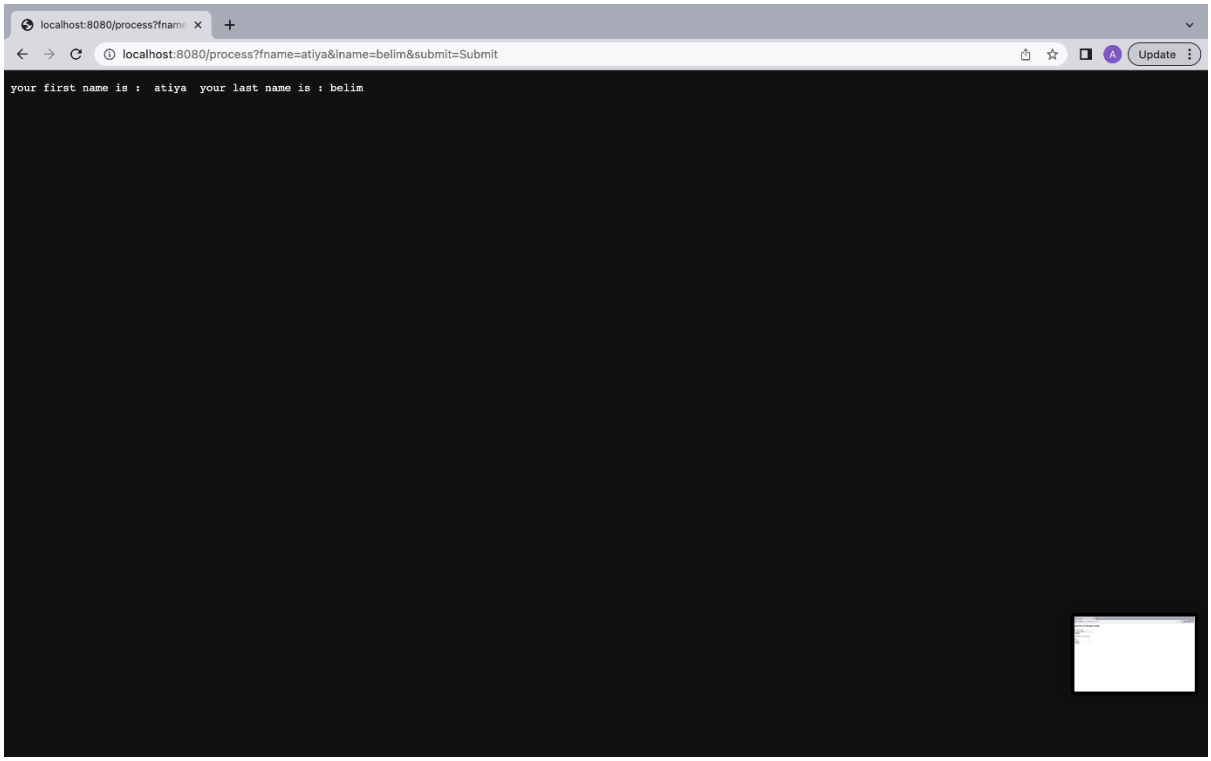
```
<!DOCTYPE html>
```

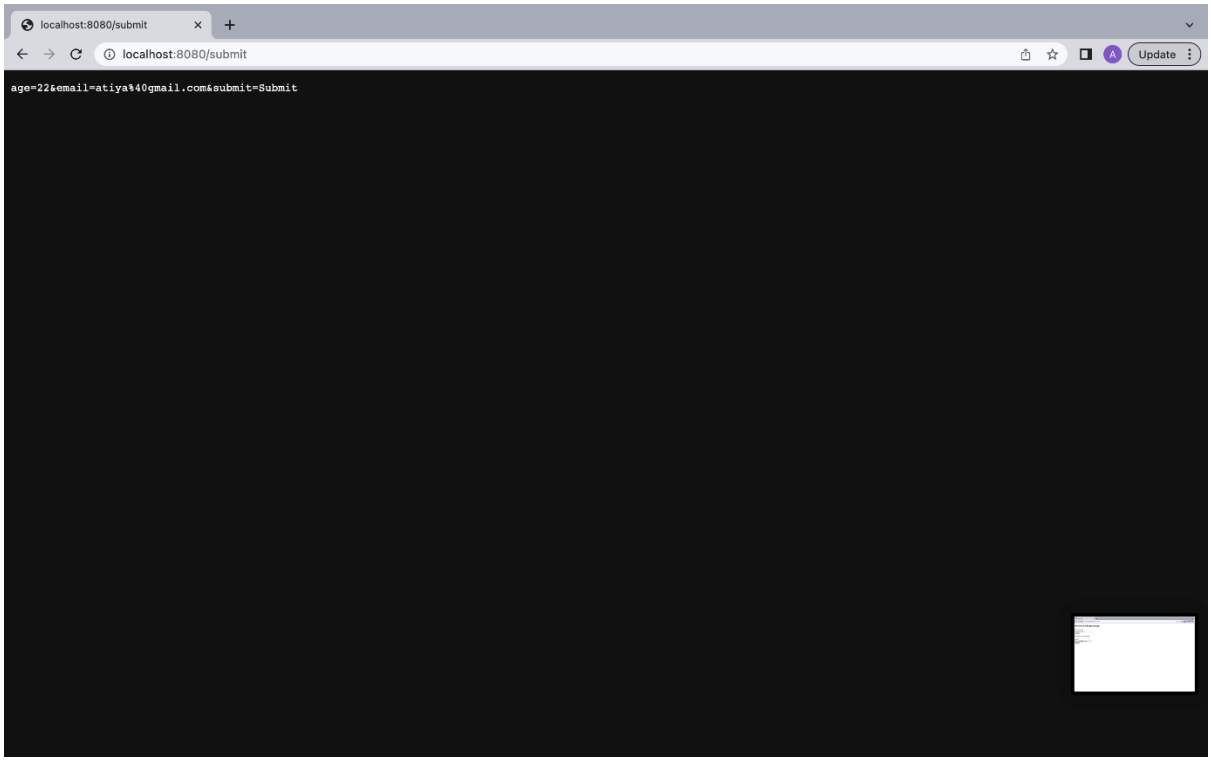
```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
   <h2>this form is with get message</h2>
    <form action="/process" method="get">
        first name:<input type="text" name="fname" id="fname"><br>
        last name:  <input type="text" name="lname" id="lname"><br>
        <input type="submit" name="submit" id="submit">
    </form>
   <p> this form is by post method</p>
    <form action="/submit" method="POST">
        age:<input type="text" name="age" id="age"><br>
        Email:  <input type="text" name="email" id="email"><br>
        <input type="submit" name="submit" id="submit">
    </form>


</body>
</html>
```

➢ OUTPUT




Document
localhost:8080/Q-1.html
Update
this form is with get message
first name: atiya
last name: belim
Submit
this form is by post method
age: 22
Email: atiya@gmail.com
Submit


localhost:8080/process?fname
localhost:8080/process?fname=atiya&lname=belim&submit=Submit
Update
your first name is : atiya your last name is : belim



Q-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.

Answer : 02

- **Q-2.js**

```
const http = require('http');
const fs=require('fs');
http.createServer((req, res) => {
    if (req.method === 'GET') {
        if (req.url === '/') {
            res.end("Home Page");
        }
        if (req.url === '/gethello') {
            fs.readFile('./files/q-2.html',(err,data)=>{
```

```
                if(err)
                {
                    return res.send("errorrr!!");
                }
                else{
                    res.writeHead(200,{
                        'Content-Type':'text/html'});
                    res.write(data);
                    return res.end();
                }
            })
        }
        if (req.url === '/ajaxcall') {
            fs.readFile('./files/ajax.html',(err,data)=>{
                if(err)
                {
                    return res.send("error!!");
                }
                else{
                    res.writeHead(200,{
                        'Content-Type':'text/html'});
                    res.write(data);
                    return res.end();
                }
            })
        }
    }

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

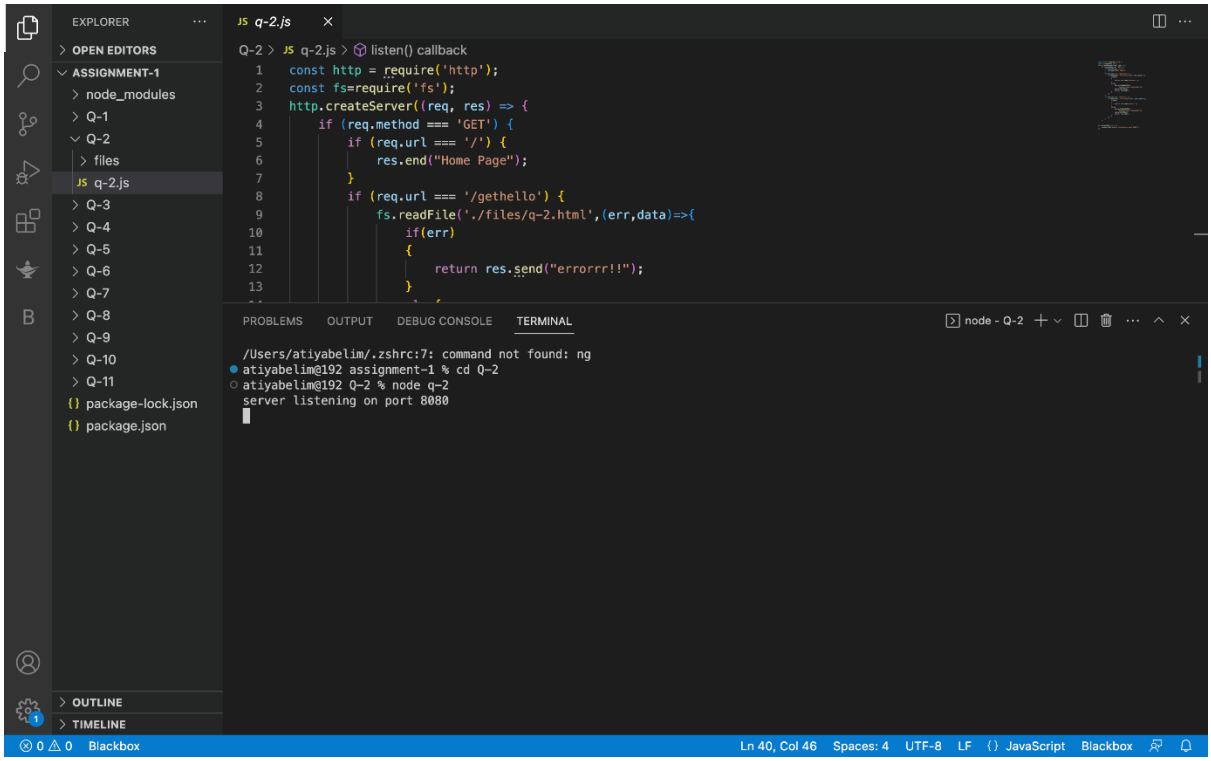
- **Q-2.html**

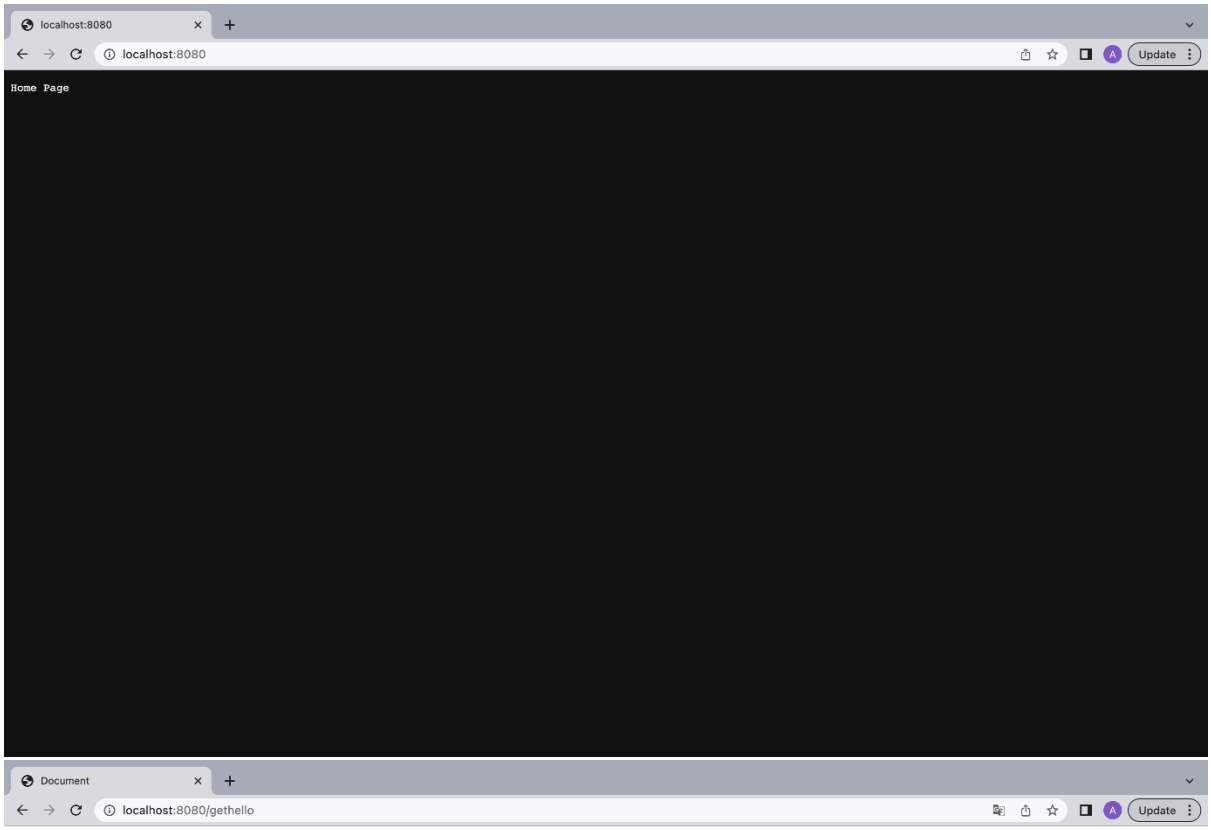
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
</head>
<body>
    <h4>Hello NodeJS!!</h4>
</body>
</html>
```

- **Ajax.html**

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>


<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>
```

➢ **OUTPUT**


EXPLORER
OPEN EDITORS
ASSIGNMENT-1
node_modules
Q-1
Q-2
files
q-2.js
Q-3
Q-4
Q-5
Q-6
Q-7
Q-8
Q-9
Q-10
Q-11
package-lock.json
package.json
OUTLINE
TIMELINE
q-2.js
Q-2 > q-2.js > listen() callback
1 const http = require('http');
2 const fs=require('fs');
3 http.createServer((req, res) => {
4 if (req.method === 'GET') {
5 if (req.url === '/') {
6 res.end("Home Page");
7 }
8 if (req.url === '/gethello') {
9 fs.readFile('./files/q-2.html',(err,data)=>{
10 if(err)
11 {
12 return res.send("errorrr!!");
13 }
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
node - Q-2
/Users/atiyabelim/.zshrc:7: command not found: ng
atiyabelim@192 assignment-1 % cd Q-2
atiyabelim@192 Q-2 % node q-2
server listening on port 8080
0 0 Blackbox
Ln 40, Col 46
Spaces: 4
UTF-8
LF
JavaScript
Blackbox



Hello NodeJS!!

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

Answer : 03

- Q-3.js

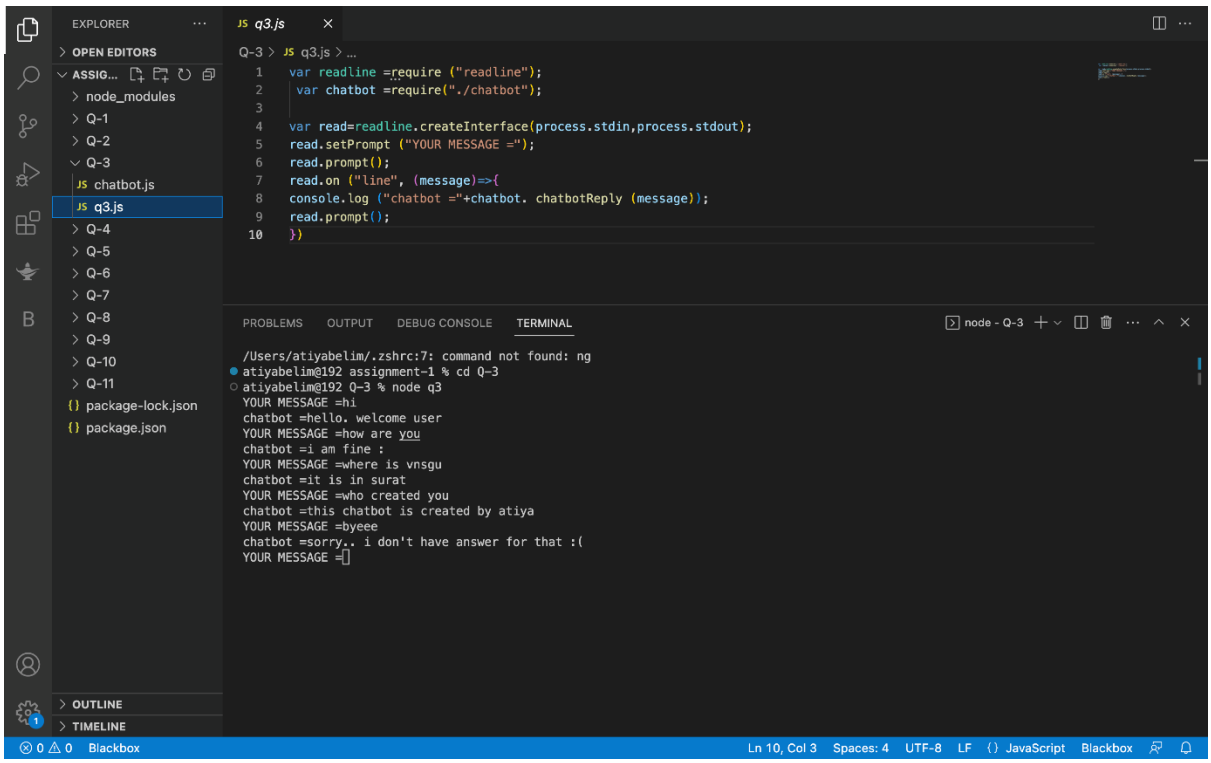
```
var readline =require ("readline");
 var chatbot =require("./chatbot");

var read=readline.createInterface(process.stdin,process.stdout);
read.setPrompt ("YOUR MESSAGE =");
read.prompt();
read.on ("line", (message)=>{
console.log ("chatbot ="+chatbot. chatbotReply (message));
read.prompt();
})
```

- Chatbot.js

```
module.exports .chatbotReply=function(message)
{
message. toLowerCase ();
```

```
if (message.indexOf ("hello") >-1 || message.indexOf ("hey") >-1 ||
message.indexOf  ("hi") >-1)
{
return "hello. welcome user";
}
else if(message.indexOf ("how") >-1 || message.indexOf ("are") >-1 )
{
return "i am fine :"
}
else if (message.indexOf ("where") >-1 || message.indexOf("vnsgu")>-1 )
{
return "it is in surat";
}
else if(message.indexOf ("created" )>-1 || message.indexOf ("chatbot")>-1 )
{
return "this chatbot is created by atiya";
}
else{
return "sorry.. i don't have answer for that :(";
}
}
```

➢ **OUTPUT**



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

Answer : 04

- Q-4.js

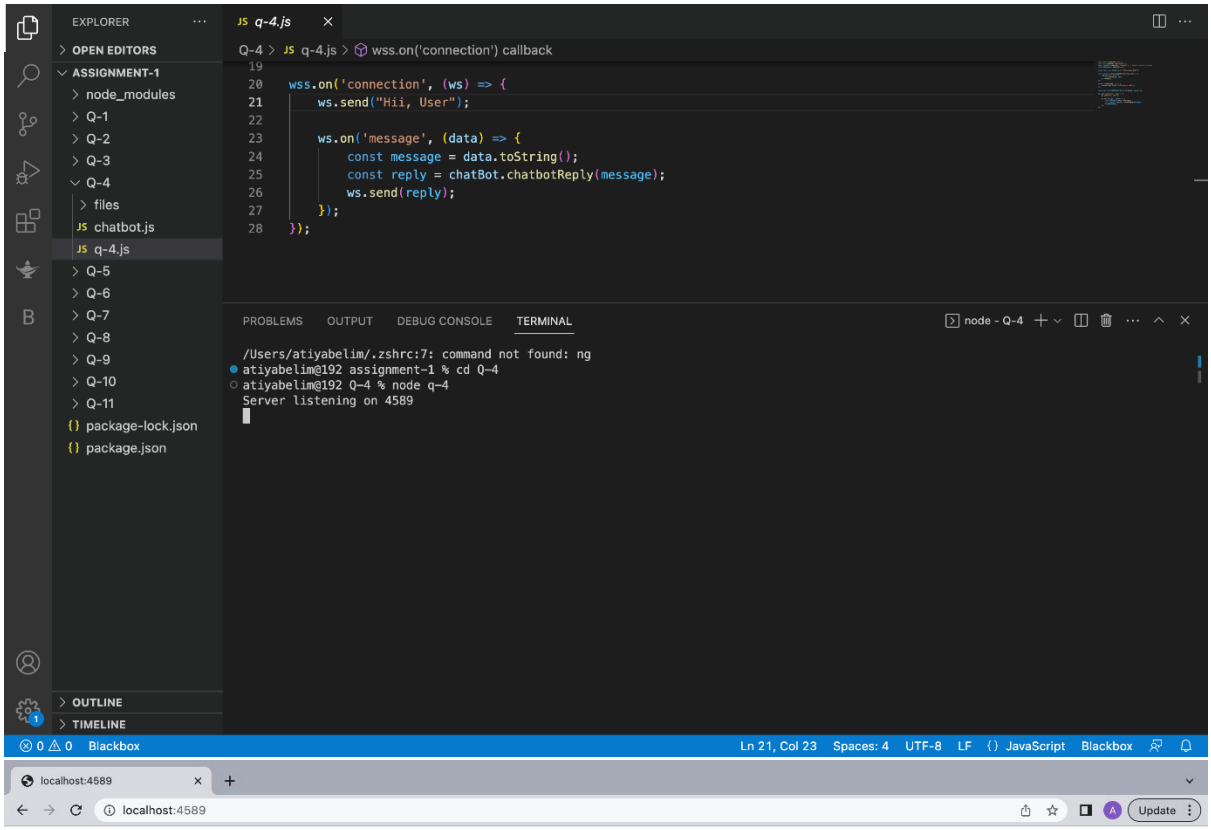
```
const http = require('http');
const st = require('node-static');
const chatBot = require('./chatbot'); // Import chatbot.js module
const WebSocket = require('ws');

const file = new st.Server('./files/index.html');

const server = http.createServer((req, res) => {
    req.on('end', () => {
        file.serve(req, res);
    }).resume();
});

server.listen(4589, () => {
    console.log("Server listening on 4589");
});

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

wss.on('connection', (ws) => {
    ws.send("Hii, User");

    ws.on('message', (data) => {
        const message = data.toString();
        const reply = chatBot.chatbotReply(message);
        ws.send(reply);
    });
});
```

- Index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8">
        <title></title>
    </head>

    <body>
        <h1>WebSocket Chat Bot</h1>
```

```
        <div id="chat">
            <div id="messages"></div>
            <input type="text" id="inputMessage"
                placeholder="Type your message here..." />
            <button onclick="sendMessage()">Send</button>
        </div>

        <script>
        const ws = new WebSocket('ws://localhost:4589');

        ws.onmessage = (event) => {
            displayMessage("Server: "+event.data);
        };

        function sendMessage() {
            const inputMessage = document.getElementById('inputMessage');
            const message = inputMessage.value;
            inputMessage.value = '';

            displayMessage('You: ' + message);
            ws.send(message);
        }

        function displayMessage(message) {
            const messagesDiv = document.getElementById('messages');
            const messageDiv = document.createElement('div');
            messageDiv.textContent = message;
            messagesDiv.appendChild(messageDiv);
        }
    </script>
    </body>

</html>
```

- **OUTPUT**



WebSocket Chat Bot

Server: Hii, User
You: hello
Server: hello. welcome user
You: who created you
Server: this chatbot is created by atiya
You: how are you
Server: i am fine :
You: where is vnsgu
Server: it is in surat
You: bye
Server: sorry.. i don't have answer for that :(

Type your message her Send

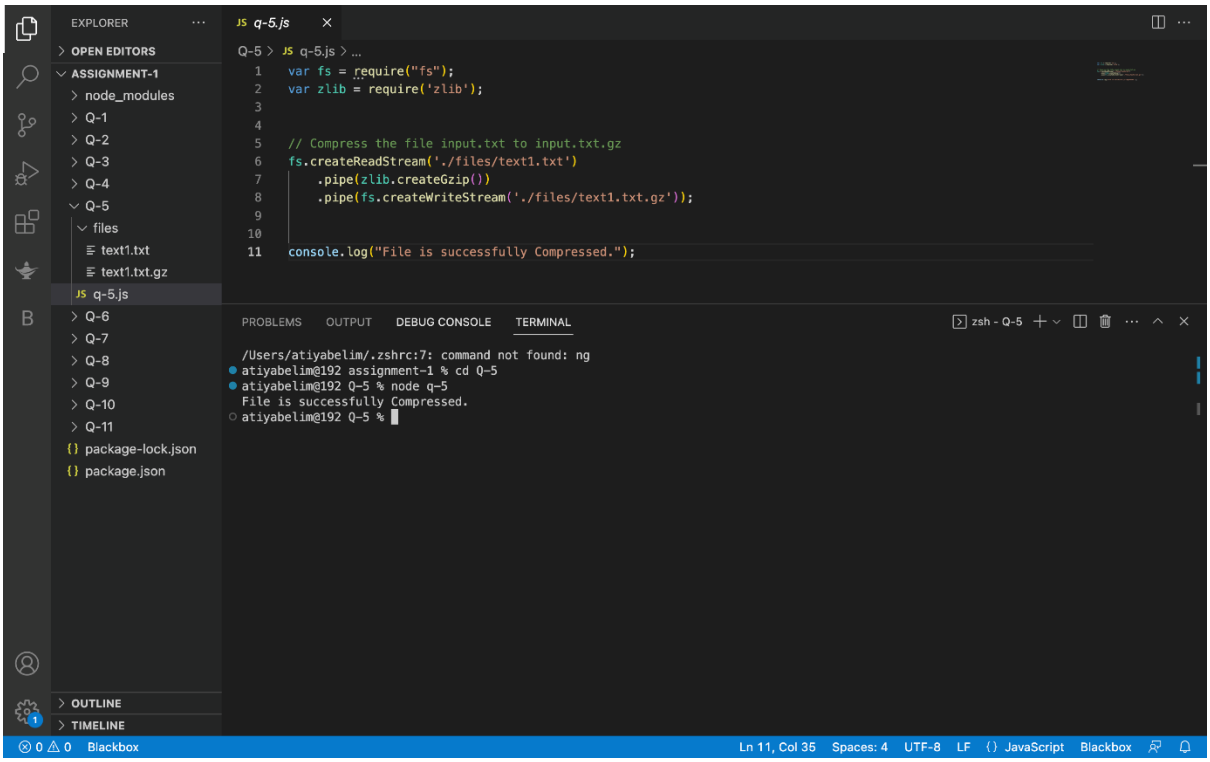
Q-5. Write a program to create a compressed zip file for a folder.

Answer : 05

- Q-5.js

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

// Compress the file input.txt to input.txt.gz
fs.createReadStream('./files/text1.txt')
    .pipe(zlib.createGzip())
    .pipe(fs.createWriteStream('./files/text1.txt.gz'));

console.log("File is successfully Compressed.");
```

- **OUTPUT**



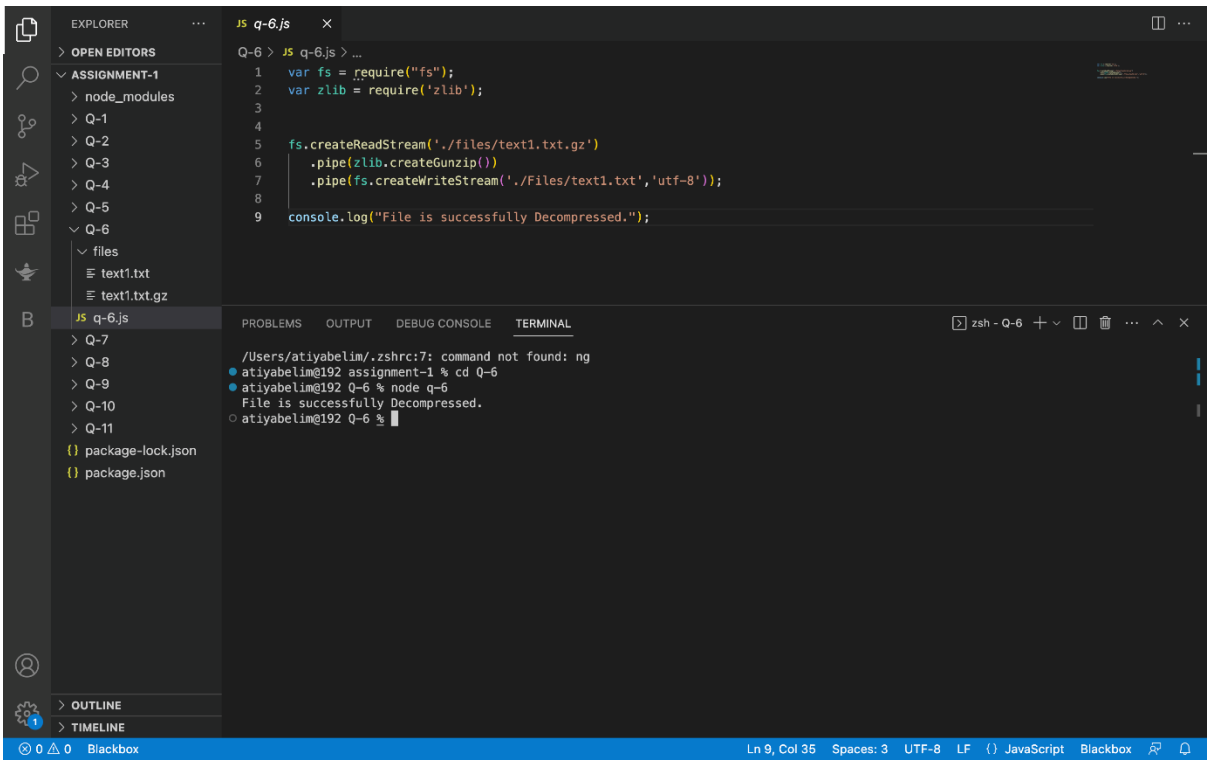
Q-6. Write a program to extract a zip file.

Answer : 06

- q-6.js

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

fs.createReadStream('./files/text1.txt.gz')
   .pipe(zlib.createGunzip())
   .pipe(fs.createWriteStream('./Files/text1.txt','utf-8'));

console.log("File is successfully Decompressed.");
```

- **OUTPUT**



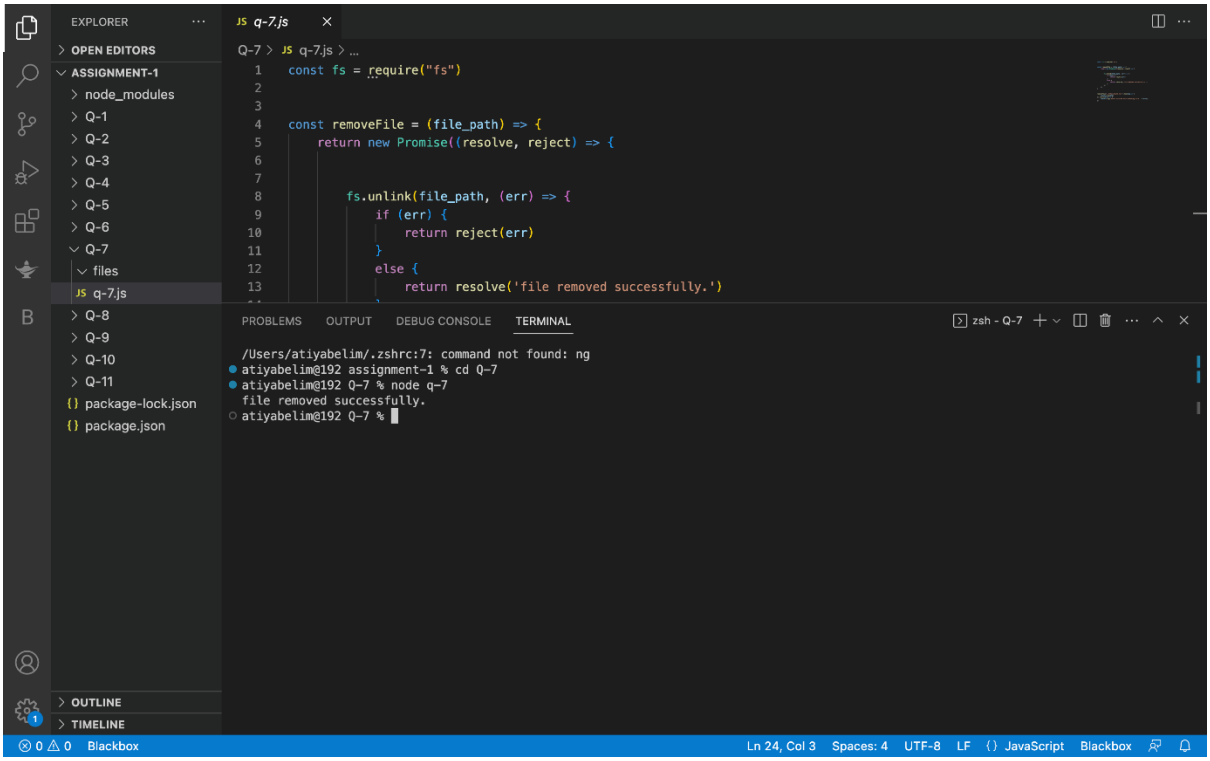
Q-7. Write a program to promisify fs.unlink function and call it.

Answer : 07

- q-7.js

```
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('./files/text1.txt').then(msg => {
    console.log(msg)
}).catch(error => {
    console.log('error occured while deleting file ' + error)
})
```

- **OUTPUT**



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

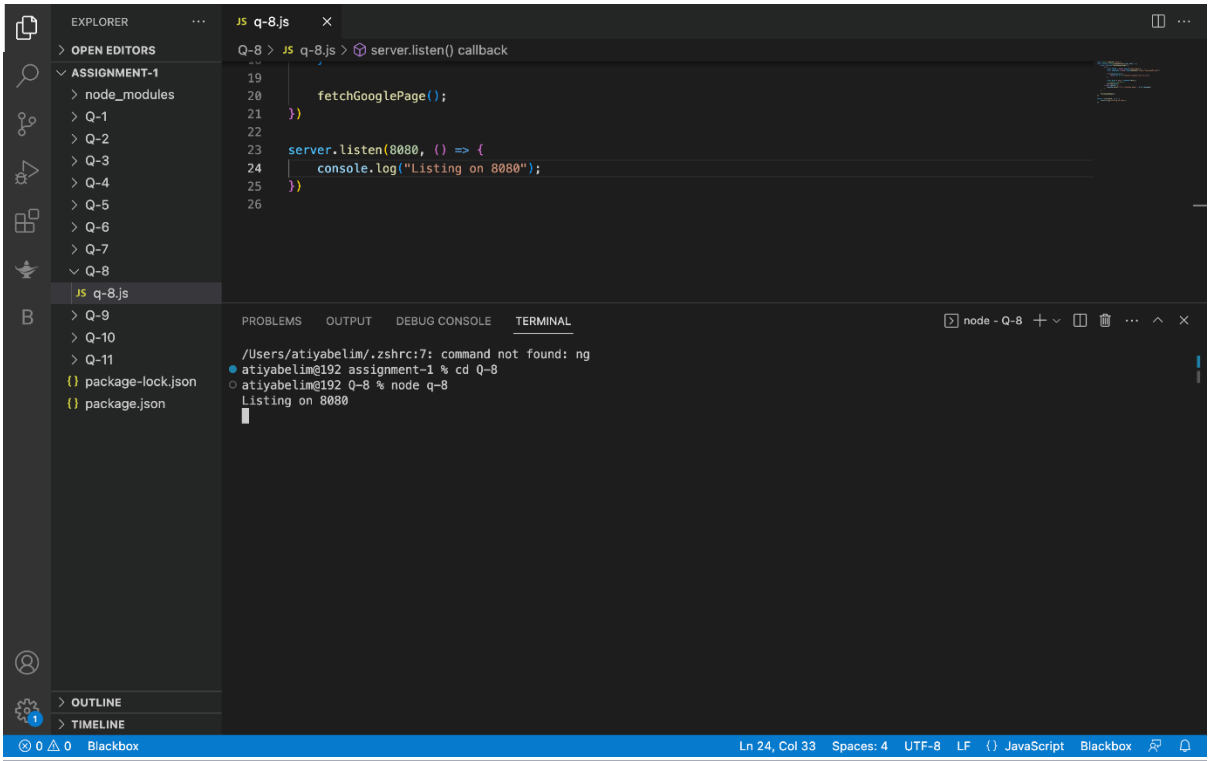
Answer : 08

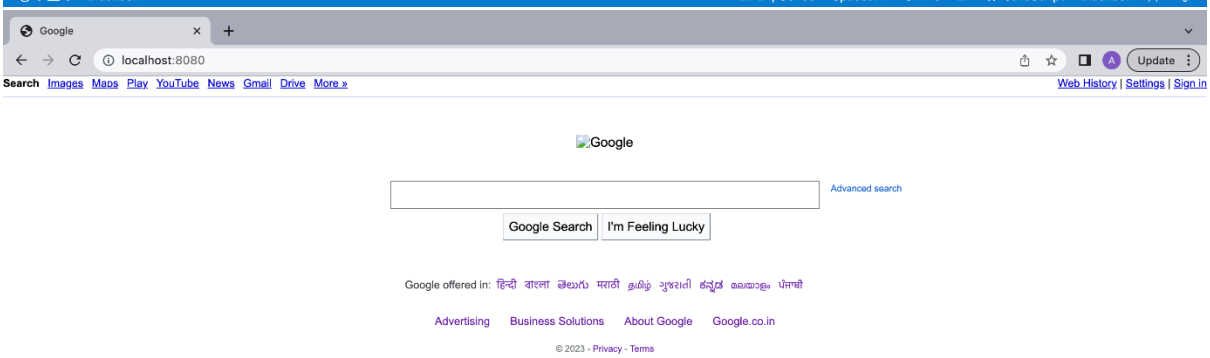
- Q-8.js

```
const http = require('http');
const server = http.createServer((req, res) => {
    async function fetchGooglePage() {
        try {
            const fetch = await import('node-fetch');
            const response = await
fetch.default('https://www.google.com');

            if (!response.ok) {
                throw new Error('Network response was not ok');
            }

            const data = await response.text();
            // console.log(data);
            res.end(data);
        } catch (error) {
            console.error('Error fetching data:', error.message);
        }
```

```
    }

    fetchGooglePage();
})

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

```

- **OUTPUT**


EXPLORER
q-8.js
OPEN EDITORS
ASSIGNMENT-1
node_modules
Q-1
Q-2
Q-3
Q-4
Q-5
Q-6
Q-7
Q-8
q-8.js
Q-9
Q-10
Q-11
package-lock.json
package.json
OUTLINE
TIMELINE
Q-8 > q-8.js > server.listen() callback
19
20 fetchGooglePage();
21 })
22
23 server.listen(8080, () => {
24 console.log("Listing on 8080");
25 })
26
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
node - Q-8
/Users/atiyabelim/.zshrc:7: command not found: ng
atiyabelim@192 assignment-1 % cd Q-8
atiyabelim@192 Q-8 % node q-8
Listing on 8080
0 0 Blackbox
Ln 24, Col 33
Spaces: 4
UTF-8
LF
JavaScript
Blackbox


Google
localhost:8080
Update
Search
Images
Maps
Play
YouTube
News
Gmail
Drive
More »
Web History
Settings
Sign in
Google
Advanced search
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
Advertising
Business Solutions
About Google
Google.co.in
© 2023 - Privacy - Terms

Q-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.

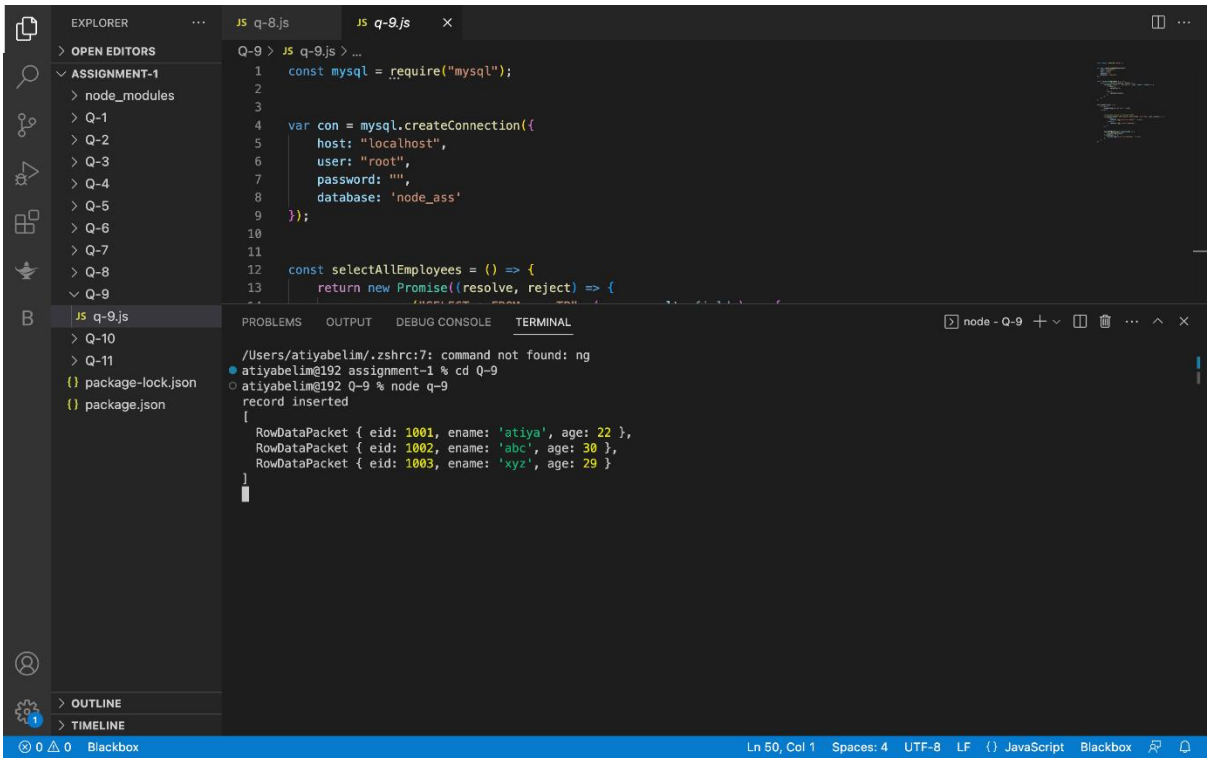
Answer : 09

- Q-9.js

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

var con = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "",
    database: 'node_ass'
});

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

con.connect((err) => {
    if (err) {
        console.log("er at con: " + err)
    } else {

        //inserting record in employee table
        con.query("INSERT INTO emp_TB values(1003,'xyz',29)", (err,
result) => {
            if (err) {
                console.log("error at insert: " + err)
            } else {
                console.log("record inserted")
            }
        })

        selectAllEmployees().then(result => {
            console.log(result)
        }).catch(err => {
```

```
                console.log("error at fetching: " + err)
            })
        }
    })

```

- **OUTPUT**



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

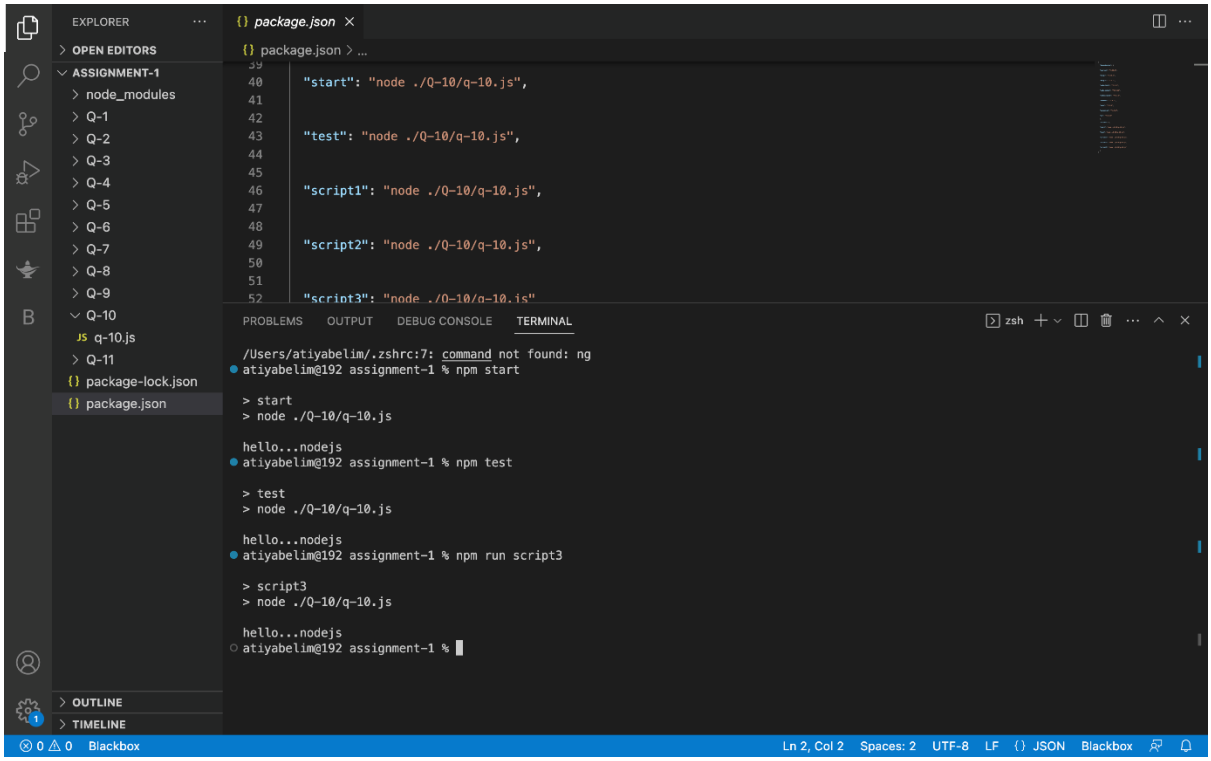
Answer : 10

- **Package.json**

```
{

  "dependencies": {


  "express": "^4.18.2",


  "mysql": "^2.18.1",


  "mysql2": "^3.5.2",


```

```
    "node-fetch": "^3.3.2",


    "node-static": "^0.7.11",


    "nodejs-mysql": "^0.1.3",


    "nodemon": "^3.0.1",


    "save": "^2.9.0",


    "socket.io": "^4.7.1",


    "ws": "^8.13.0"

    },

    "scripts": {


    "start": "node ./Q-10/q-10.js",


    "test": "node ./Q-10/q-10.js",


    "script1": "node ./Q-10/q-10.js",


    "script2": "node ./Q-10/q-10.js",


    "script3": "node ./Q-10/q-10.js"

    }
  }

```

➢ **OUTPUT**



Q-11. Develop an application to show live cricket score.

Answer : 11