

PRACTICAL ASSIGNMENT -1

Name : kanbi himanshi a.

Roll no : 021

Div : A

Subject : Full stack web development using node.js

Semester : 7th(M.sc.IT)

Github link:

<https://github.com/himanshikanbi/Assignment1.git>

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

- Serve static resources. - Handle GET request.
- Handle POST request.

CODE:

Server.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 === '/')
  {
    var path = "./public/index.html";
    fs.readFile(path, (err, data) => {
      if(err)
      { res.writeHead(404, {'Content-Type': 'text/html'});

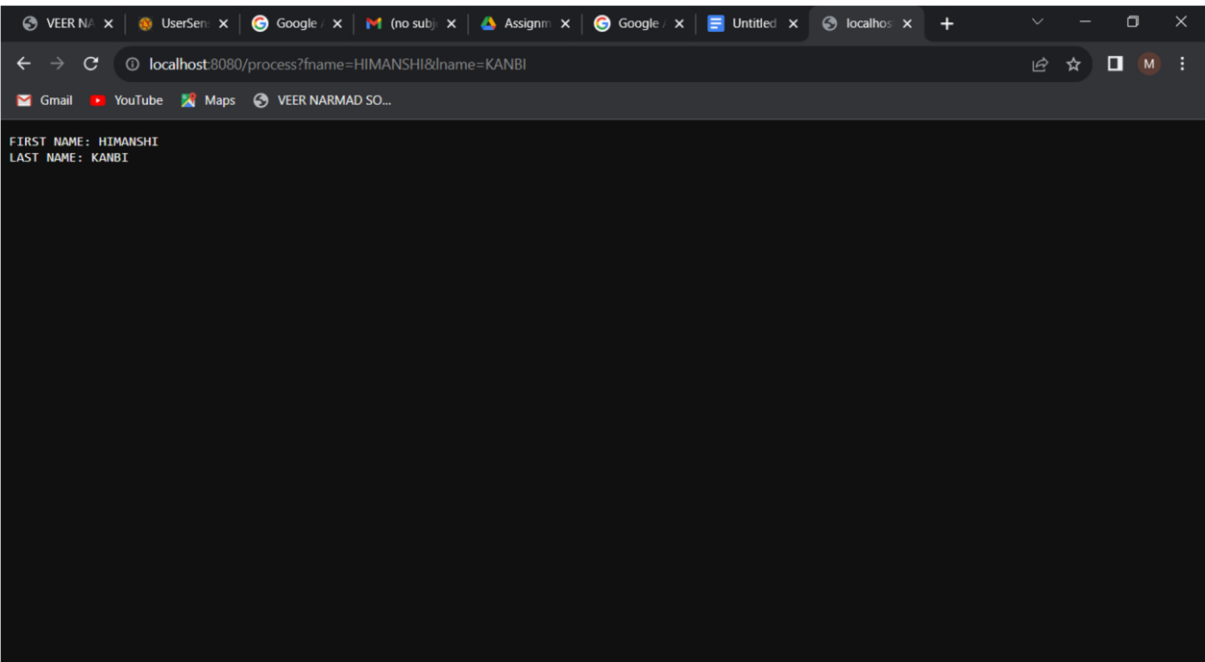
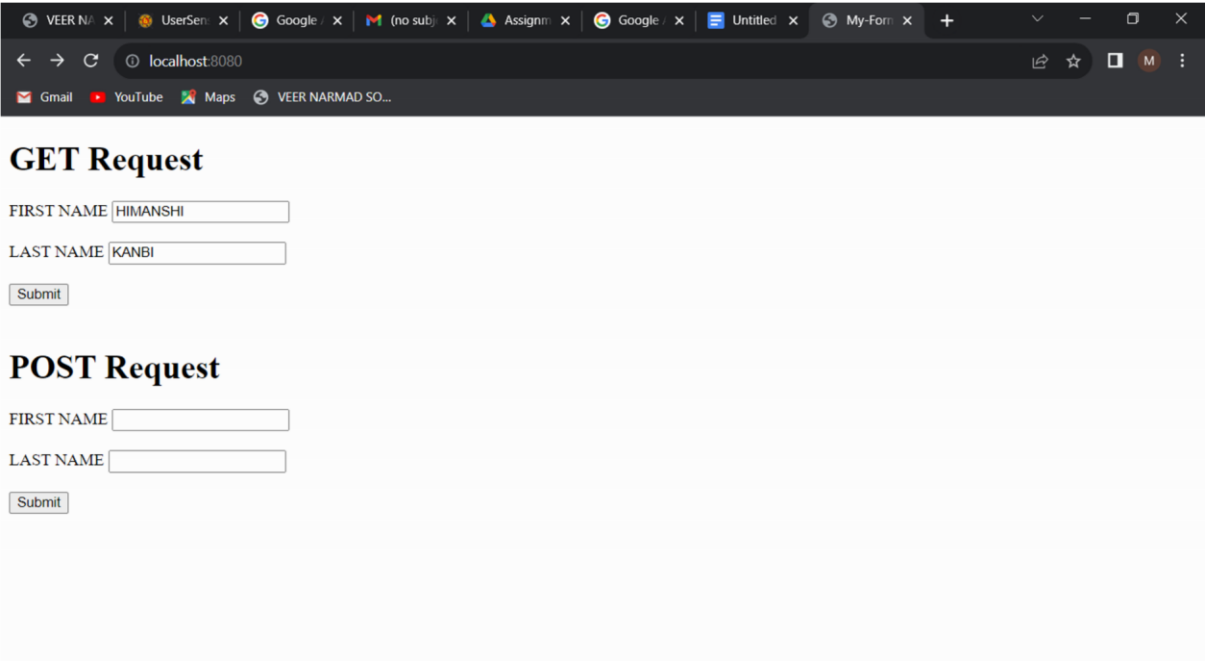
        return res.end("404 page not found...");
      }
      res.writeHead(200, {'Content-Type': 'text/html'});
      res.write(data);
      res.end();
    });
  }
  else if(ul.pathname === '/process' && req.method === 'GET')
  {
    res.write("FIRST NAME: " + ul.query.fname + " \nLAST NAME: " +
ul.query.lname);
    res.end();
  }
  else if(ul.pathname === '/process' && req.method === 'POST')
  {
    let body = '';
    req.on('data', chunk => {
      body += chunk.toString();
    });
    req.on('end', () => {
      res.end(body);
    });
  }
});
```

```
    });  
  }  
});  
server.listen(8080);  
  
console.log("The Server is Running on 8080");
```

Index.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>My-Form</title>  
</head>  
<body>  
  <h1>GET Request</h1>  
  <form action="/process" method="GET">  
    FIRST NAME <input type="text" name="fname"/><br/><br/>  
    LAST NAME <input type="text" name="lname"/><br/><br/>  
    <input type="submit" value="Submit"/><br/><br/>  
  </form>  
  <h1>POST Request</h1>  
  <form action="/process" method="POST">  
    FIRST NAME <input type="text" name="fname"/><br/><br/>  
    LAST NAME <input type="text" name="lname"/><br/><br/>  
    <input type="submit" value="Submit"/>  
  </form>  
</body>  
</html>
```

OUTPUT:



GET Request

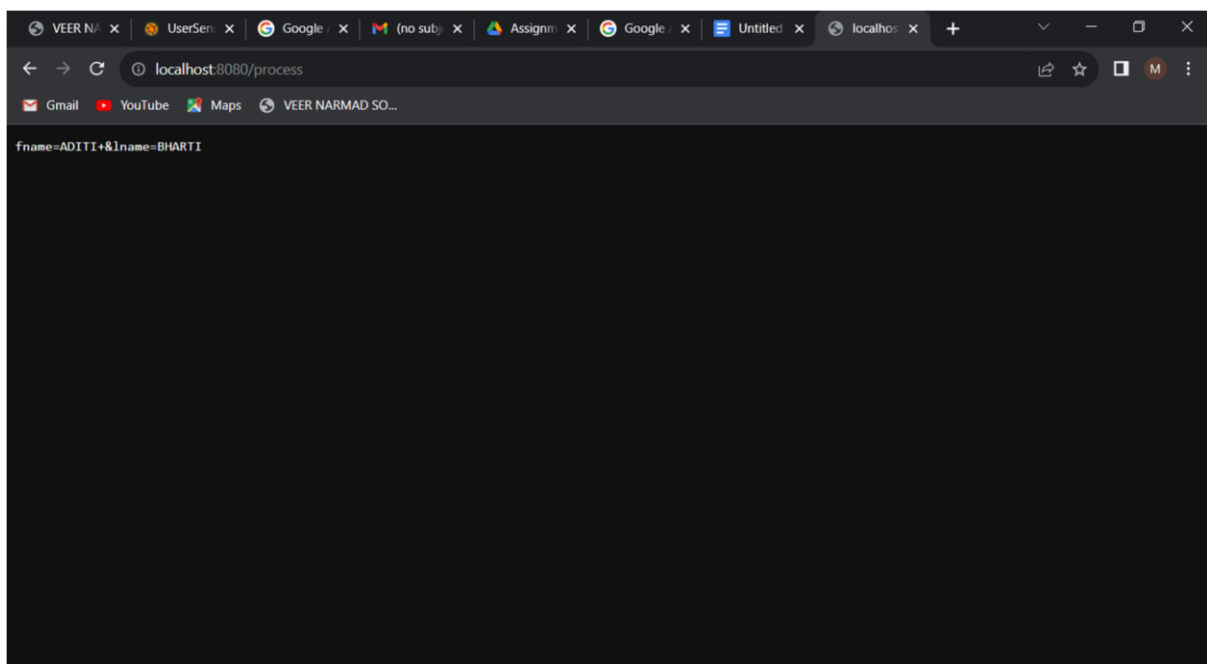
FIRST NAME

LAST NAME

POST Request

FIRST NAME

LAST NAME



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.)

CODE:

server.js

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

```
const fs=require('fs');
http.createServer((req, res) => {
  if (req.method === 'GET') {
    if (req.url === '/') {
      res.end("NODE JS");
    }
    if (req.url === '/gethello') {
      fs.readFile('./public/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();
        }
      })
    }
    if (req.url === '/ajaxcall') {
      fs.readFile('./public/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(8000, () => {
  console.log("server listening on port 8000");
```

```
})
```

Home.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>

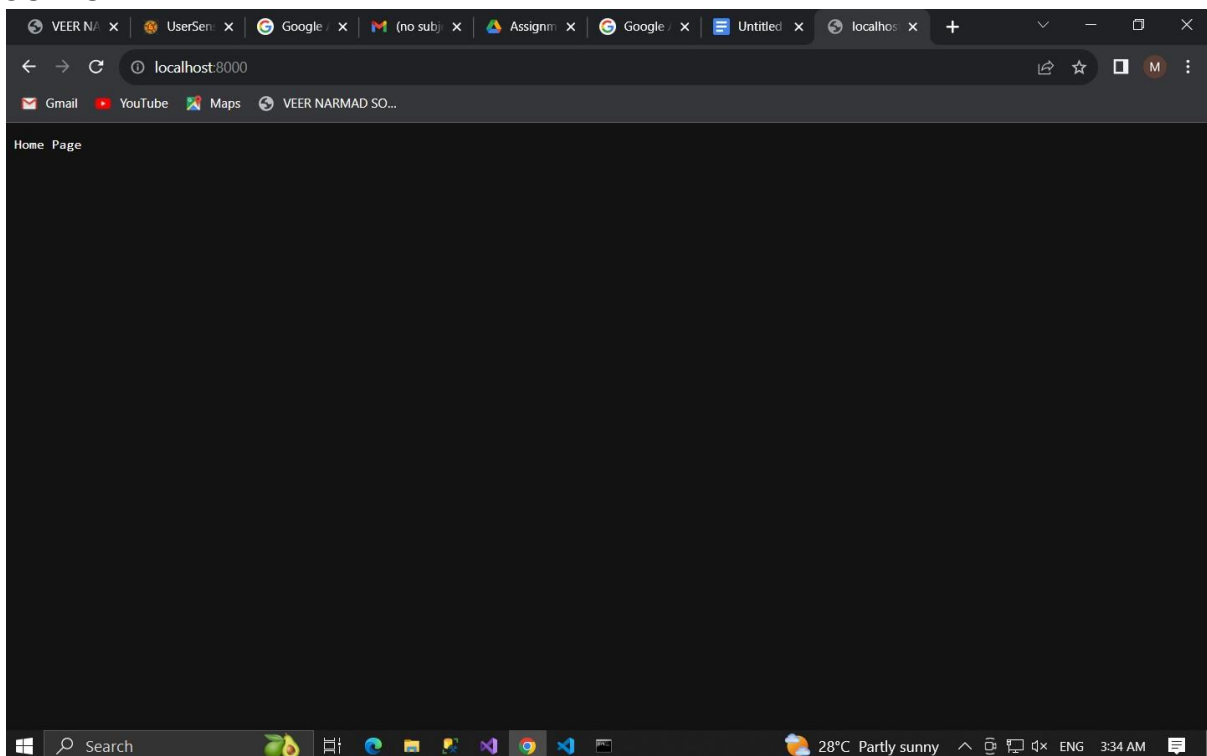
<body>
  <div id="page_content">

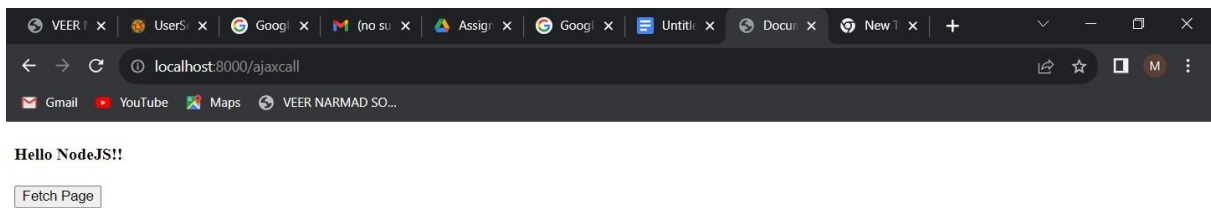
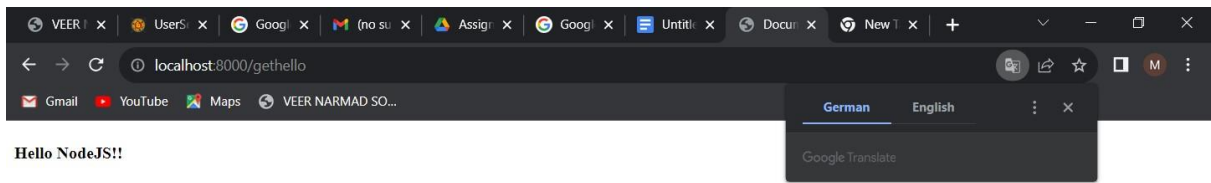
  </div>
  <button onclick="loadData()">Fetch Page</button>
  <script>
    function loadData() {
      var xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
```

```
        document.getElementById("page_content").innerHTML =
this.responseText;
    }
};
xhttp.open("GET", "/gethello", true);
xhttp.send();
}
</script>
</body>

</html>
```

OUTPUT:





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

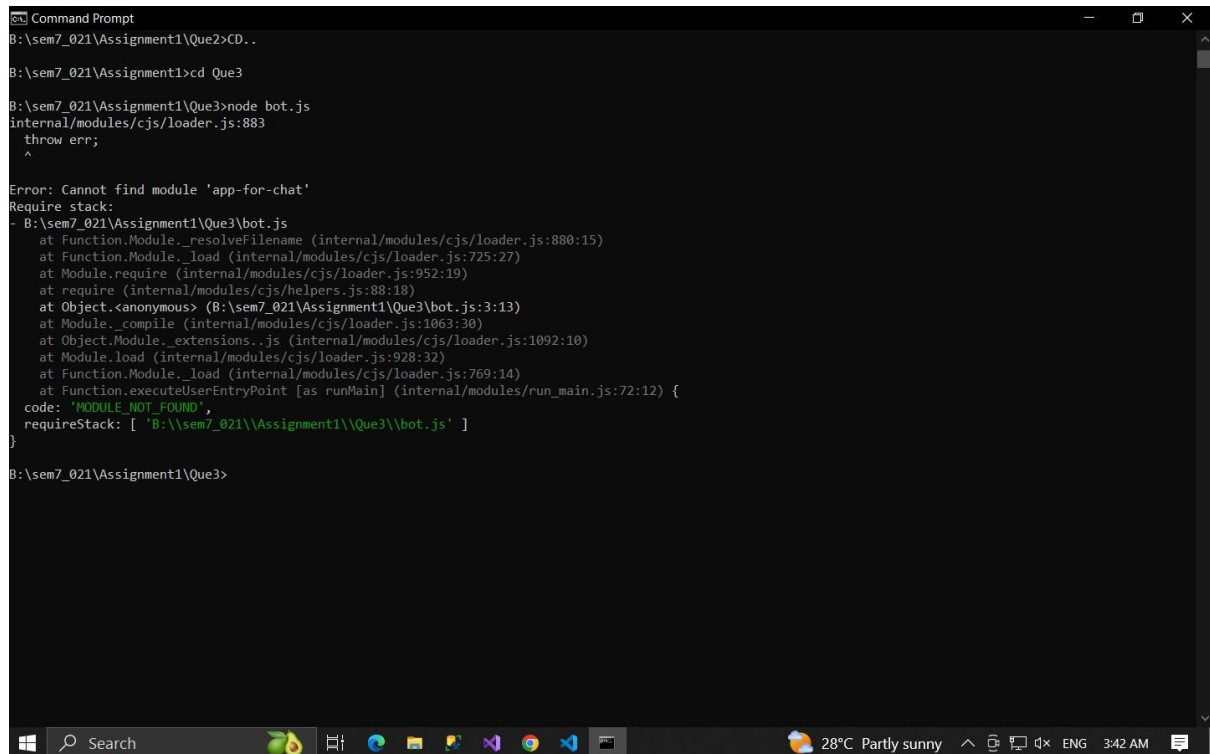
CODE:

```
const readline = require('readline');
```

```
var rl = readline.createInterface({input: process.stdin, output:
process.stdout, terminal: false});

const app = require('app-for-chat');
```

OUTPUT:



```
Command Prompt
B:\sem7_021\Assignment1\Que2>CD..

B:\sem7_021\Assignment1>cd Que3

B:\sem7_021\Assignment1\Que3>node bot.js
internal/modules/cjs/loader.js:883
  throw err;
  ^

Error: Cannot find module 'app-for-chat'
Require stack:
- B:\sem7_021\Assignment1\Que3\bot.js
    at Function.Module._resolveFilename (internal/modules/cjs/loader.js:880:15)
    at Function.Module._load (internal/modules/cjs/loader.js:725:27)
    at Module.require (internal/modules/cjs/loader.js:952:19)
    at require (internal/modules/cjs/helpers.js:88:18)
    at Object.<anonymous> (B:\sem7_021\Assignment1\Que3\bot.js:3:13)
    at Module._compile (internal/modules/cjs/loader.js:1063:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1092:10)
    at Module.load (internal/modules/cjs/loader.js:928:32)
    at Function.Module._load (internal/modules/cjs/loader.js:769:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:72:12) {
  code: 'MODULE_NOT_FOUND',
  requireStack: [ 'B:\\sem7_021\\Assignment1\\Que3\\bot.js' ]
}

B:\sem7_021\Assignment1\Que3>
```

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

CODE:

Server.js

```
const WebSocket = require('ws')

var http = require('http');
var fs = require('fs');

var httpserver = http.createServer(function(request, response)
{
  if(request.url=="/")
  {
    fs.readFile("./public/index.html", (err,data)=>{
      response.write(data)
      response.end();
    })
  }
}).listen(8080, function() {
  console.log((new Date()) +
```

```

        ' Server is listening on port 8080');
    });
const wss=new WebSocket.Server({server:httpserver})
wss.on("connection", (clientws)=>{
    clientws.send("Hello Client")

    clientws.on("message", (msg)=>{
        console.log("Received "+msg)
        clientws.send("Received "+ msg)
    })
})
})

```

Index.js

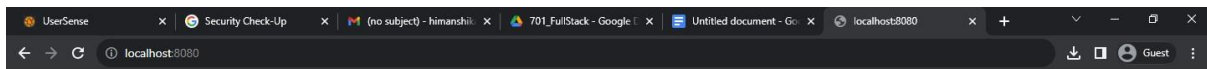
```

<!DOCTYPE html >
<html>
    <body>
<script language="javascript">
var ws = new WebSocket('ws://localhost:8080');
ws.addEventListener("message", function(msg1) {
    var msg = msg1.data;
    document.getElementById('chatlog').innerHTML+='<br>Server: ' + msg;
});

function sendMessage(){
    var message = document.getElementById('message').value;
    document.getElementById('chatlog').innerHTML+='<br> Me: ' +
    message; ws.send(message);
}
</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:



Data from server

Server: Hello Client
Me: hello
Server: Received hello

Data from client



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

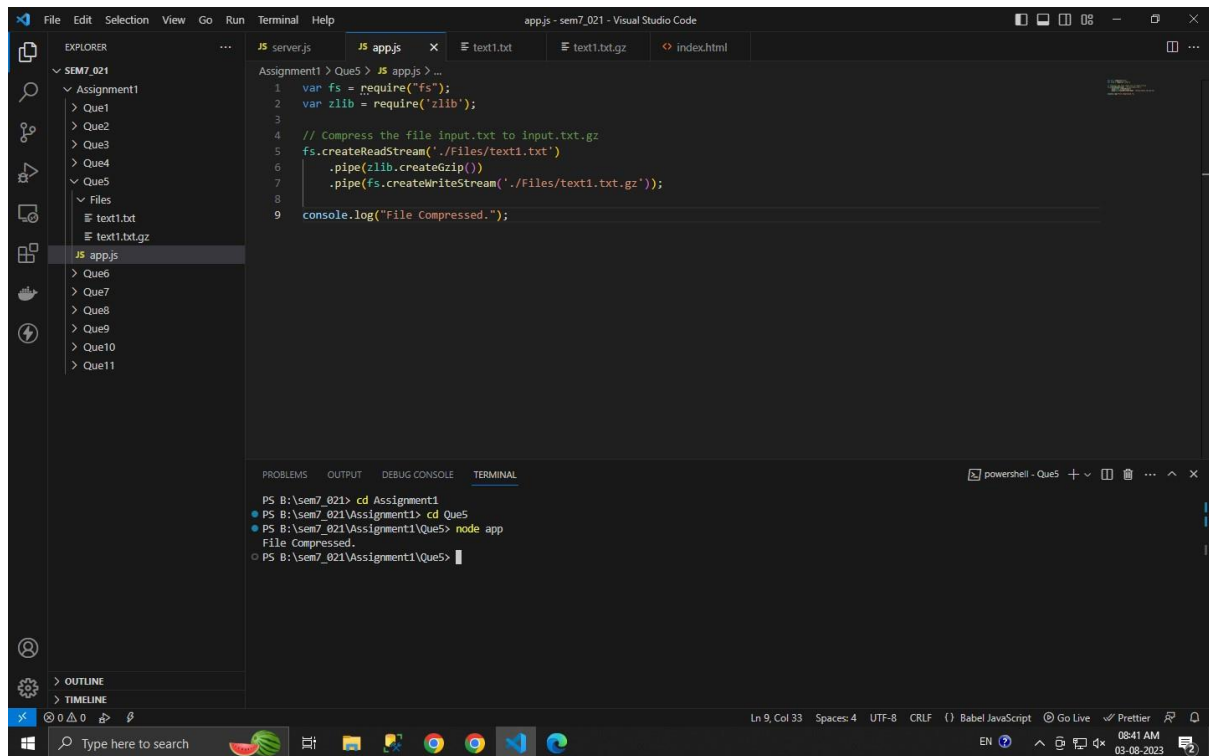
CODE:

```
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 Compressed.");
```

OUTPUT:



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

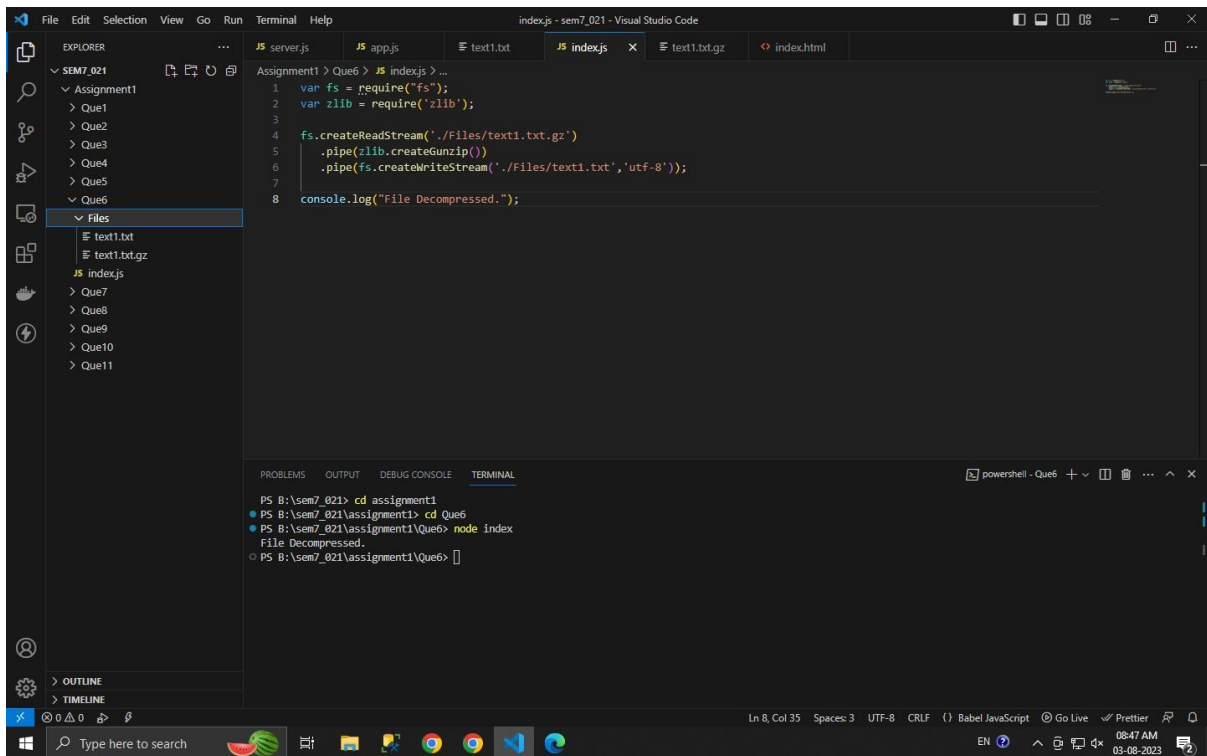
CODE:

```
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 Decompressed.");
```

OUTPUT:



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

CODE:

```
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:

```
main.js - sem7_021 - Visual Studio Code

1  const fs = require("fs")
2
3  const removeFile = (file_path) => {
4      return new Promise((resolve, reject) => {
5
6          fs.unlink(file_path, (err) => {
7              if (err) {
8                  return reject(err)
9              }
10             else {
11                 return resolve('file removed successfully.')
12             }
13         })
14     })
15 }
16
17 removeFile('./files/text1.txt').then(msg => {
18     console.log(msg)
19 }).catch(error => {
20     console.log('error occured while deleting file ' + error)
21 })

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Error: Cannot find module 'B:\sem7_021\assignment1\Que7\index'
    at Module._resolveFilename (node:internal/modules/cjs/loader:1048:15)
    at Module._load (node:internal/modules/cjs/loader:901:27)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:83:12)
    at node:internal/main/run_main_module:23:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}

Node.js v20.4.0
PS B:\sem7_021\assignment1\Que7> node main
file removed successfully.
PS B:\sem7_021\assignment1\Que7>
```

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

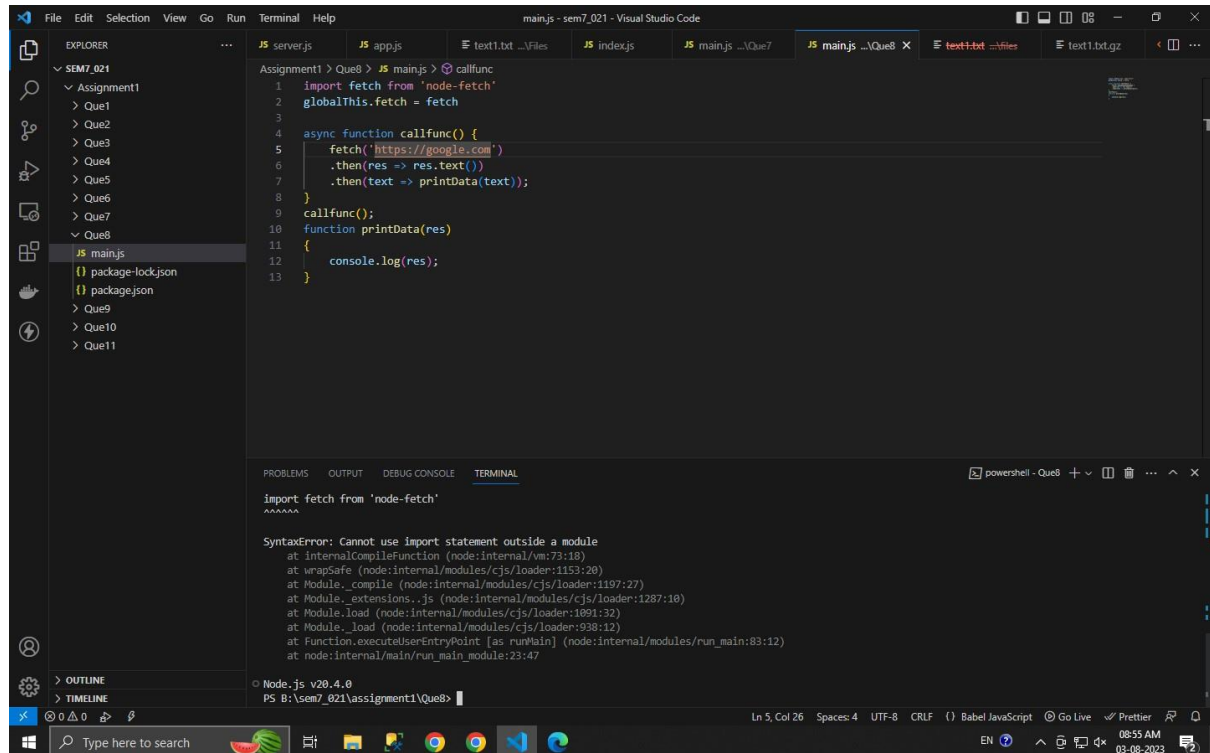
CODE:

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

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

function printData(res)
{
    console.log(res);
}
```

OUTPUT:



```
1 import fetch from 'node-fetch'
2 globalThis.fetch = fetch
3
4 async function callfunc() {
5   fetch('https://google.com')
6     .then(res => res.text())
7     .then(text => printData(text));
8 }
9 callfunc();
10 function printData(res)
11 {
12   console.log(res);
13 }
```

SyntaxError: Cannot use import statement outside a module
at internalCompileFunction (node:internal/vm:73:18)
at wrapSafe (node:internal/modules/cjs/loader:1153:20)
at Module._compile (node:internal/modules/cjs/loader:1197:27)
at Module._extensions..js (node:internal/modules/cjs/loader:1287:10)
at Module.load (node:internal/modules/cjs/loader:1091:32)
at Module._load (node:internal/modules/cjs/loader:938:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:83:12)
at node:internal/main/run_main_module:23:47

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.

CODE:

```
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);
      }
    });
  });
}
```



```

    }
  })
})

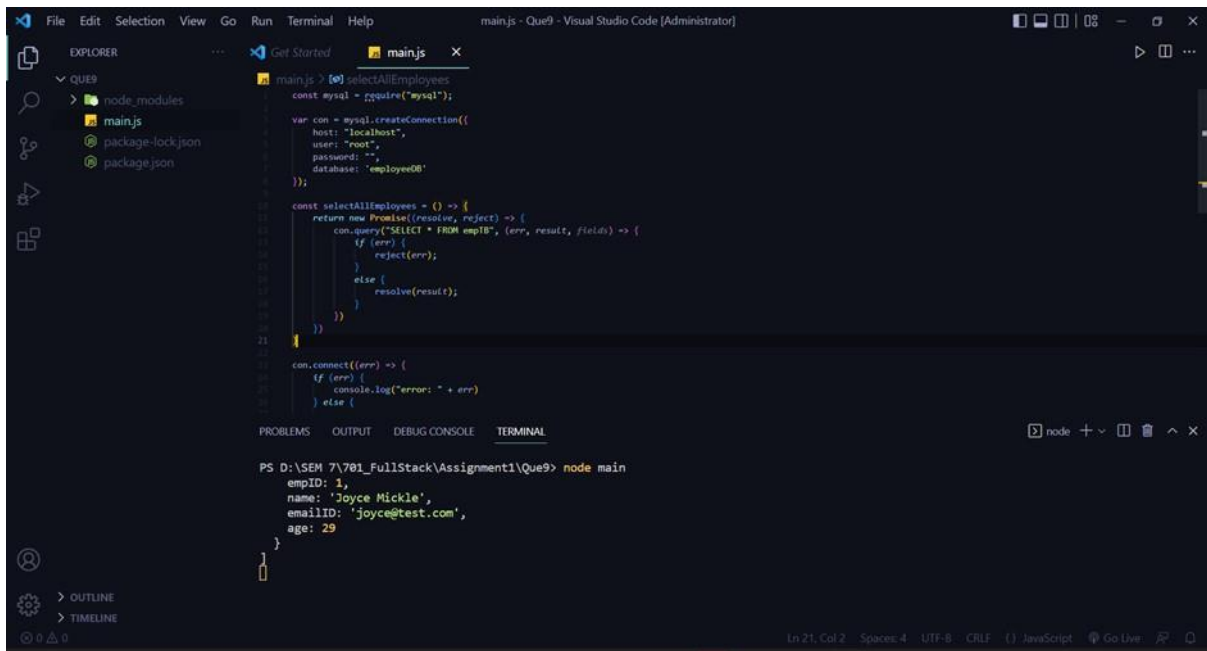
con.connect((err) => {
  if (err) {
    console.log("error: " + err)
  } else {

    //inserting record in employee table
    con.query("INSERT INTO empTB values(null,'Joyce
Mickle','joyce@test.com',29)", (err, result) => {
      if (err) {
        console.log("error: " + err)
      } else {
        console.log("record inserted")
      }
    })

    //getting all data from employee table using promise based
    approach
    selectAllEmployees().then(result => {
      console.log(result)
    }).catch(err => {
      console.log("error: " + err)
    })
  }
})

```

OUTPUT:



```
main.js > [0] selectAllEmployees
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 emp1b", (err, result, fields) => {
      if (err) {
        reject(err);
      } else {
        resolve(result);
      }
    })
  })
}

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

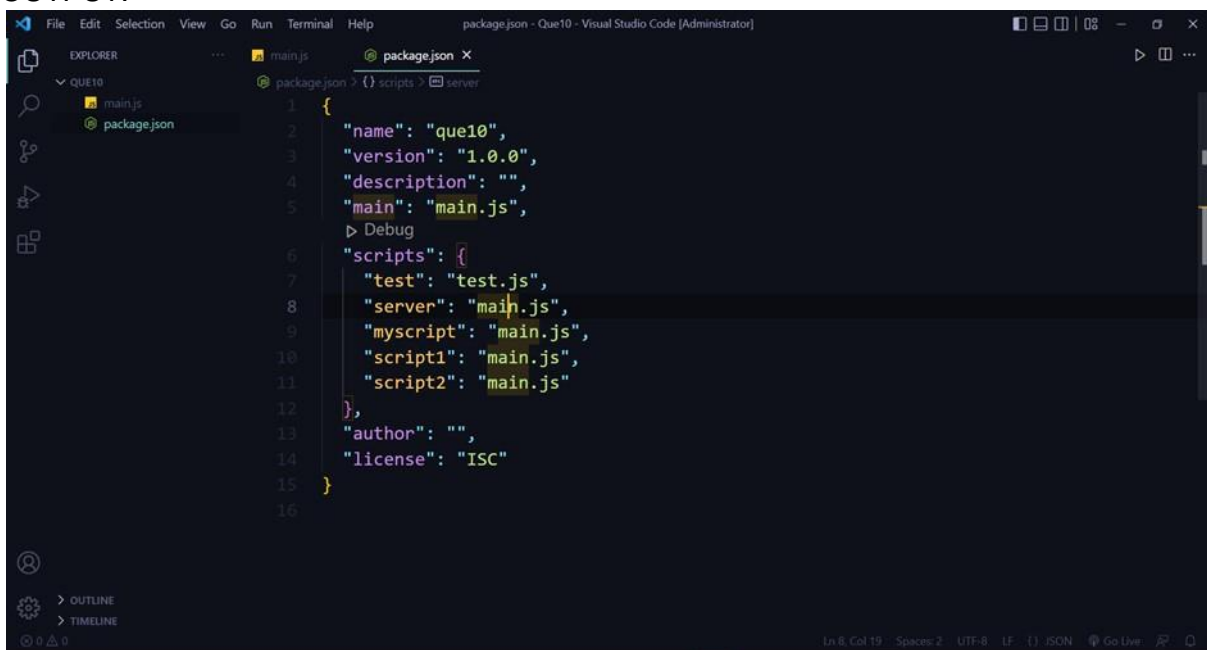
PS D:\SEM 7\701_FullStack\Assignment1\Que9> node main
empID: 1,
name: 'Joyce Mickle',
emailID: 'joyce@test.com',
age: 29
}
```

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

CODE:

```
console.log("Hello From Question 10!");
```

OUTPUT:



```
package.json > {} scripts > server
1 {
2   "name": "que10",
3   "version": "1.0.0",
4   "description": "",
5   "main": "main.js",
6   "scripts": {
7     "test": "test.js",
8     "server": "main.js",
9     "myscript": "main.js",
10    "script1": "main.js",
11    "script2": "main.js"
12  },
13   "author": "",
14   "license": "ISC"
15 }
```

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

CODE:

```
const express = require('express');
const app = express();
```

```

const http = require('http').createServer(app)
const PORT = process.env.PORT || 5000
var data;
const socket = require('socket.io')(http)
const fetch=require('node-fetch')
var

data=fetch("https://cricket-
api.vercel.app/cric.php?url=https://www.cric
buzz.com/live-cricket-scores/32057/3rd-test-india-tour-of-england-
2022")
).then(res => res.json())
.then(json => {
  data=json
});
socket.on('connection',function(s){
  console.log("Helloo "+ data)
  s.emit('message',data);

})
http.listen(PORT, () => {
  console.log(`Listening on port ${PORT}`);
})
app.use(express.static(__dirname+'/public'))
app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html')
})

```

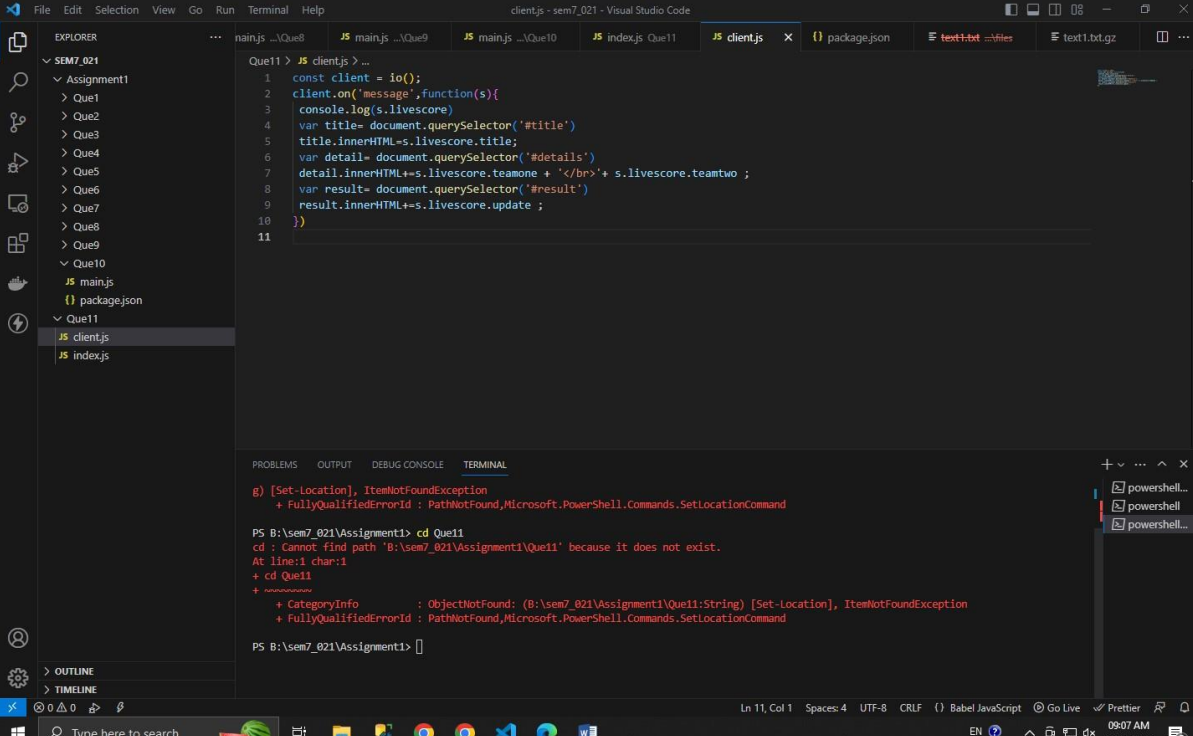
Client.js

```

const client = io();
client.on('message',function(s){
  console.log(s.livescore)
  var title= document.querySelector('#title')
  title.innerHTML=s.livescore.title;
  var detail= document.querySelector('#details')
  detail.innerHTML+=s.livescore.teamone + '</br>'+ s.livescore.teamtwo
  ;
  var result= document.querySelector('#result')
  result.innerHTML+=s.livescore.update ;
})

```

OUTPUT:



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with a folder named 'SEM7_021' containing subfolders 'Assignment1' and 'Que11'. The 'client.js' file is selected in the 'Que11' folder. The main editor area shows the content of 'client.js', which is a JavaScript script using jQuery to interact with an API and update DOM elements. The Terminal window at the bottom shows a PowerShell session where the user attempts to run 'cd Que11', resulting in a 'PathNotFound' error because the directory does not exist.

```
1 const client = io();
2 client.on('message',function(s){
3   console.log(s.livescore)
4   var title= document.querySelector('#title')
5   title.innerHTML=s.livescore.title;
6   var detail= document.querySelector('#details')
7   detail.innerHTML+=s.livescore.teamone + '<br>' + s.livescore.teamtwo ;
8   var result= document.querySelector('#result')
9   result.innerHTML+=s.livescore.update ;
10 })
11
```

g) [Set-Location], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand

PS B:\sem7_021\Assignment1> cd Que11
cd : Cannot find path 'B:\sem7_021\Assignment1\Que11' because it does not exist.
At line:1 char:1
+ cd Que11
+ ~~~~~
+ CategoryInfo : ObjectNotFound: (B:\sem7_021\Assignment1\Que11:String) [Set-Location], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand

PS B:\sem7_021\Assignment1> []

