Assignment- 1

Name : Iwala Rohan KamleshBhai.

Roll No : 18.

Sem : 7th.

Subject : Application Development With Full Stack.(705).

Date : 30-Jul-23.

Github Link :- https://github.com/Rohaniwala/18_Rohan_iwala_PA1_705

1. Develop a web server with following functionalities:

- Serve static resources.

- Handle GET request.

- Handle POST request.

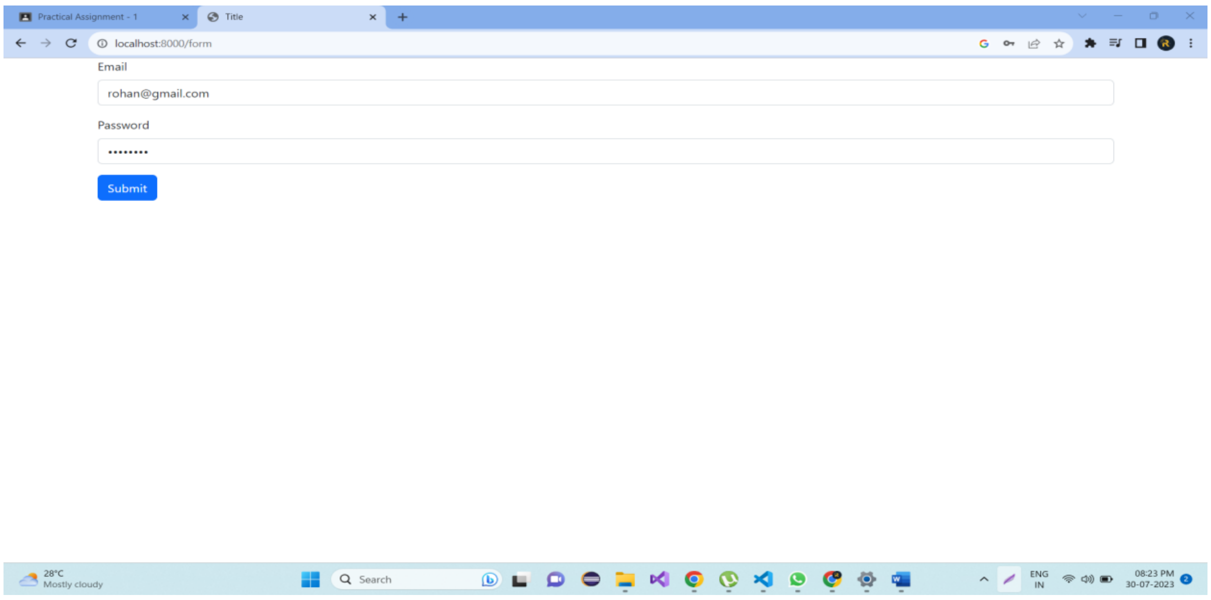
Index.html

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

<head>
    <title>Title</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <!-- Bootstrap CSS v5.2.1 -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">

</head>

<body>
    <div class="container">
        <form action="/submit" method="post">
            <div class="mb-3">
                <label for="email" class="form-label">Email</label>
                <input type="email" class="form-control" name="email"
id="email" placeholder="Entar Email here ...">
            </div>
            <div class="mb-3">
                <label for="password" class="form-label">Password</label>
                <input type="password" class="form-control" name="password"
id="password"
                    placeholder="Entar Password here ...">
            </div>
            <button type="submit" class="btn btn-primary">Submit</button>
        </form>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
"
        integrity="sha384-
oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
crossorigin="anonymous">
        </script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"
        integrity="sha384-
7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz"
crossorigin="anonymous">
        </script>
</body>

</html>
```

Index.js

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


http.createServer((req, res) => {
    const responseObject = { API: req.url, METHOD: req.method, Body:
(req.body) ? req.body : {} };
    if (req.url === '/' && req.method == 'GET') {
        res.setHeader("Content-Type", "application/json");
        res.write(JSON.stringify(responseObject));
        res.end();
    }
    else if (req.url === '/data' && req.method == 'GET') {
        res.setHeader("Content-Type", "application/json");
        res.write(JSON.stringify(responseObject));
        res.end();
    }
    else if (req.url === '/form' && req.method == 'GET') {
        fs.readFile('./index.html', 'utf8', (err, data) => {
            if (err) {
                res.writeHead(500, { 'Content-Type': 'text/plain' });
                res.end('Internal Server Error');
            } else {
                res.writeHead(200, { 'Content-Type': 'text/html' });
                res.end(data);
            }
        });
    }
    else if (req.url === '/submit' && req.method == 'POST') {
        let body = '';
        req.on('data', (chunk) => {
            body += chunk.toString();
        });
        req.on('end', () => {
            const formData = new URLSearchParams(body);
            const email = formData.get('email');
            const password = formData.get('password');
            responseObject.Body = {
                email: email,
                password: password
            }
            res.write(JSON.stringify(responseObject));
            res.end();
        });
    }
    else {
        res.write(JSON.stringify(responseObject));
        res.end();
    }
}).listen(8000, () => {
    console.log("server i listening on port 8000");
    console.log("localhost:8000/");
})
```

Output:



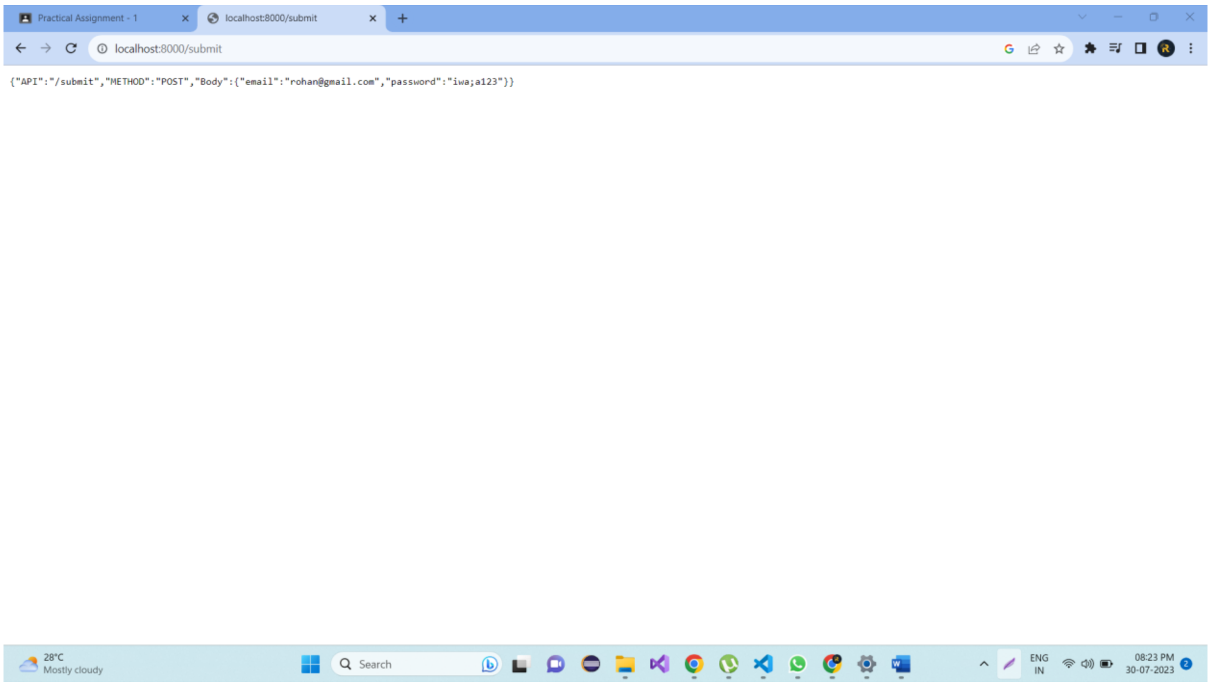
Email

rohan@gmail.com

Password

••••••••

Submit



```
{"API":"/submit","METHOD":"POST","Body":{"email":"rohan@gmail.com","password":"iwa;a123"}}
```

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.)

Ajaxcall.html

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <div id="content-Page">

    </div>
    <button onclick="fetchdata()">Fecth Data</button>
```

```html
</body>
<script>
    function fetchdata() {
        var httpRequest = new XMLHttpRequest();
        httpRequest.onreadystatechange = function () {
            if (httpRequest.readyState == 4 && httpRequest.status == 200) {
                document.getElementById("content-Page").innerHTML =
httpRequest.responseText
            }
        };
        httpRequest.open("GET", '/gethello', true);
        httpRequest.send();
    }
</script>

</html>
```

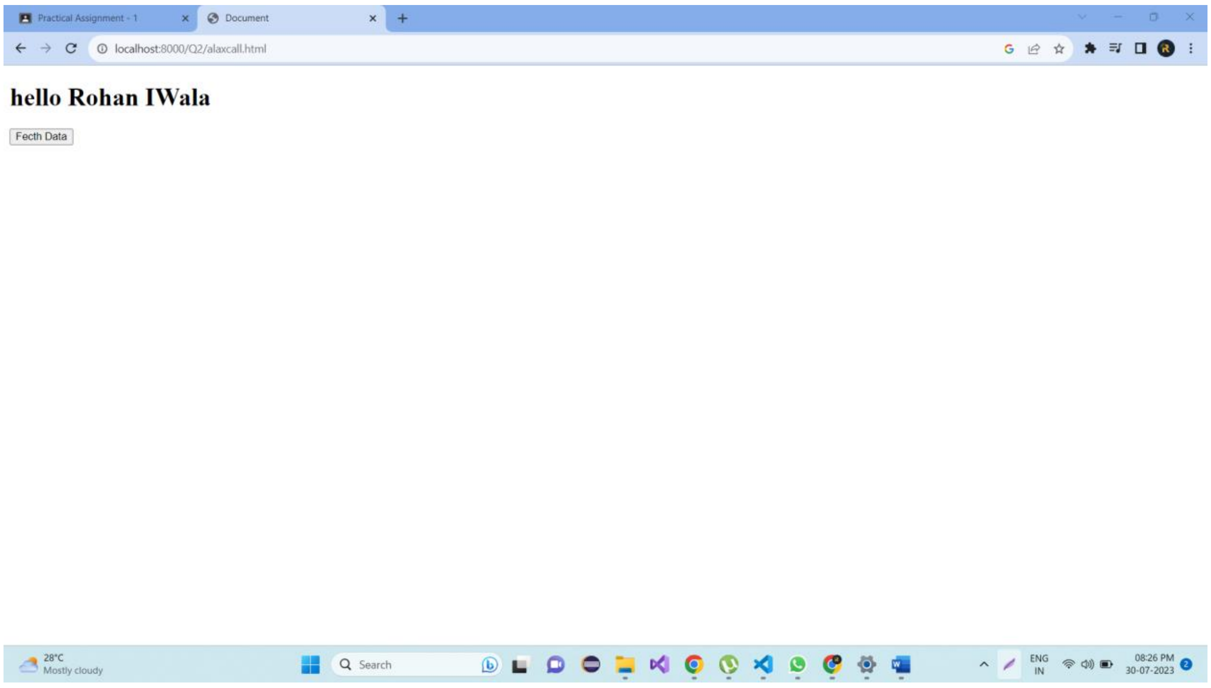
Hello.html

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <h1>hello Rohan IWala</h1>
</body>

</html>
```

Server.js

```javascript
var http = require('http');
var fs = require('fs');
http.createServer((req, res) => {
    if (req.method === 'GET' && req.url === '/') {
        res.end("hello Page 1");
    }

    if (req.url === '/gethello') {
        fs.readFile('./hello.html', (err, data) => {
            if (err) {
                fs.write("page not found");
                fs.end();
            }
            else {
                res.writeHead(200, { 'Content-Type': 'text/html' });
                res.write(data);
                res.end();
            }
        })
    }
    if (req.url === '/Q2/alaxcall.html') {
        fs.readFile('./alaxcall.html', (err, data) => {
            if (err) {
                fs.write("pagenotfound");
                fs.end();
            }
            else {
                res.writeHead(200, { 'Content-Type': 'text/html' });
                res.write(data);
                res.end();
            }
        })
    }
    // console.log("hello");
}).listen(8000);
```

Output:


localhost:8000/Q2/alaxcall.html

hello Rohan IWala

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

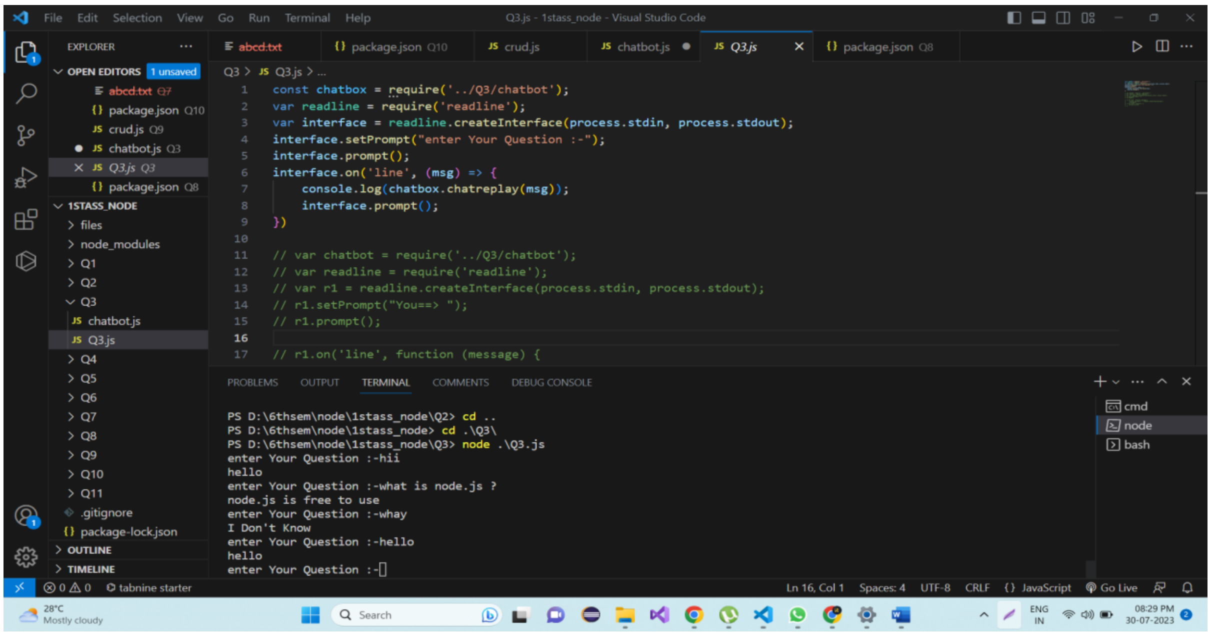
Chatbot.js

```
module.exports.chatreplay = (message) => {
    if (message.toLowerCase().indexOf('hii') > -1 ||
message.toLowerCase().indexOf('hello') > -1) {
        return "hello";
    }
    else if (message.toLowerCase().indexOf('What is Node.js ?') > -1 ||
message.toLowerCase().indexOf('node.js') > -1) {
        return "node.js is free to use";
    }
    else {
        return "I Don't Know";
    }
}
```

Q3.js

```
const chatbox = require('../Q3/chatbot');
var readline = require('readline');
var interface = readline.createInterface(process.stdin, process.stdout);
interface.setPrompt("enter Your Question :-");
interface.prompt();
interface.on('line', (msg) => {
    console.log(chatbox.chatreplay(msg));
    interface.prompt();
})
```

Output:



```
const chatbox = require('../Q3/chatbot');
var readline = require('readline');
var interface = readline.createInterface(process.stdin, process.stdout);
interface.setPrompt("enter Your Question :-");
interface.prompt();
interface.on('line', (msg) => {
    console.log(chatbox.chatreplay(msg));
    interface.prompt();
})

// var chatbot = require('../Q3/chatbot');
// var readline = require('readline');
// var r1 = readline.createInterface(process.stdin, process.stdout);
// r1.setPrompt("You==> ");
// r1.prompt();

// r1.on('line', function (message) {
```

```
PS D:\6thsem\node\1stass_node\Q2> cd ..
PS D:\6thsem\node\1stass_node> cd .\Q3\
PS D:\6thsem\node\1stass_node\Q3> node .\Q3.js
enter Your Question :-hii
hello
enter Your Question :-what is node.js ?
node.js is free to use
enter Your Question :-whay
I Don't Know
enter Your Question :-hello
hello
enter Your Question :-
```

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

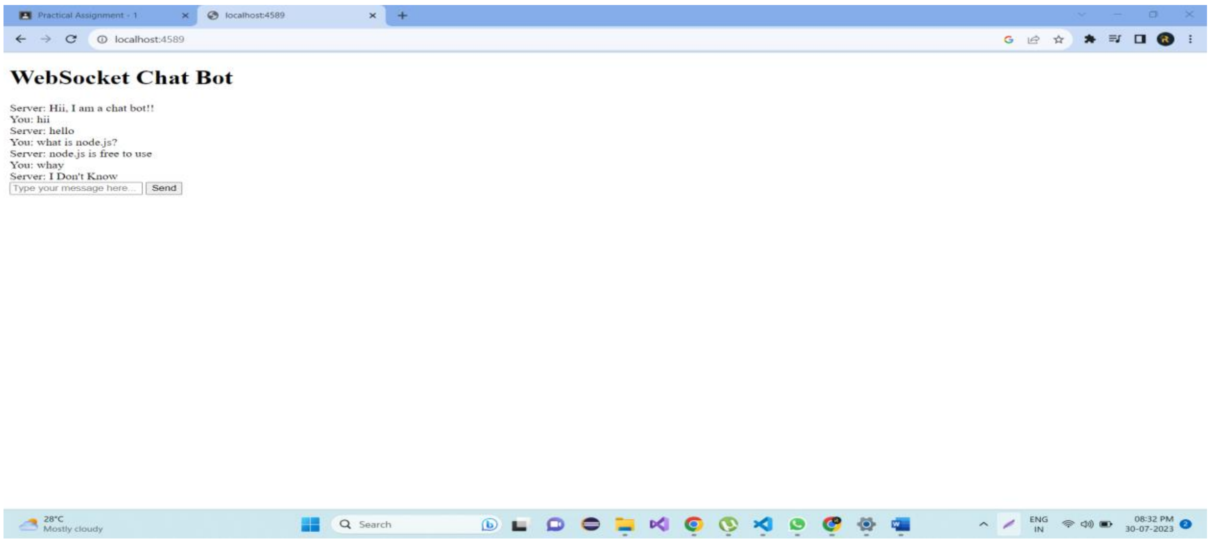
Q4.html

```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>
```

Mainchat.js

```
const http = require('http');
const st = require('node-static');
const chatBot = require('../Q3/chatbot'); // Import chatbot.js module
const WebSocket = require('ws');

const file = new st.Server('./Q4.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, I am a chat bot!!");

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

Output:



WebSocket Chat Bot

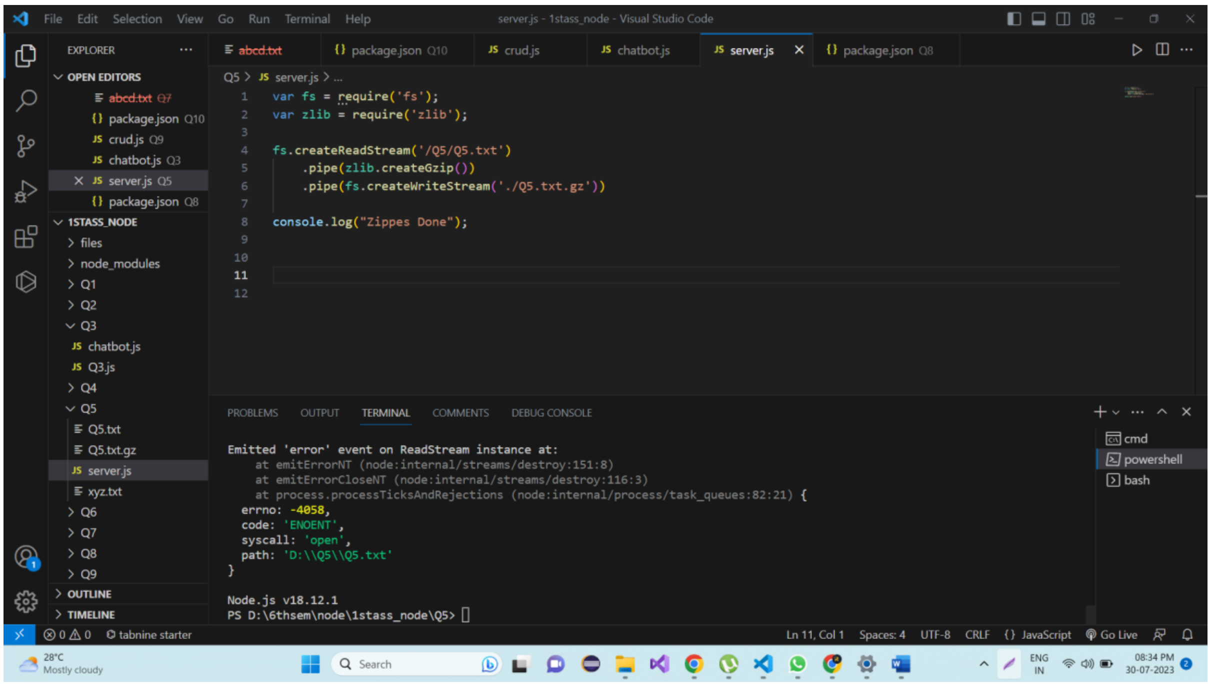
```
Server: Hii, I am a chat bot!!
You: hii
Server: hello
You: what is node.js?
Server: node.js is free to use
You: whay
Server: I Don't Know
```

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

Server.js

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

fs.createReadStream('/Q5/Q5.txt')
    .pipe(zlib.createGzip())
    .pipe(fs.createWriteStream('./Q5.txt.gz'))

console.log("Zippes Done");
```

output:



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

fs.createReadStream('/Q5/Q5.txt')
    .pipe(zlib.createGzip())
    .pipe(fs.createWriteStream('./Q5.txt.gz'))

console.log("Zippes Done");
```

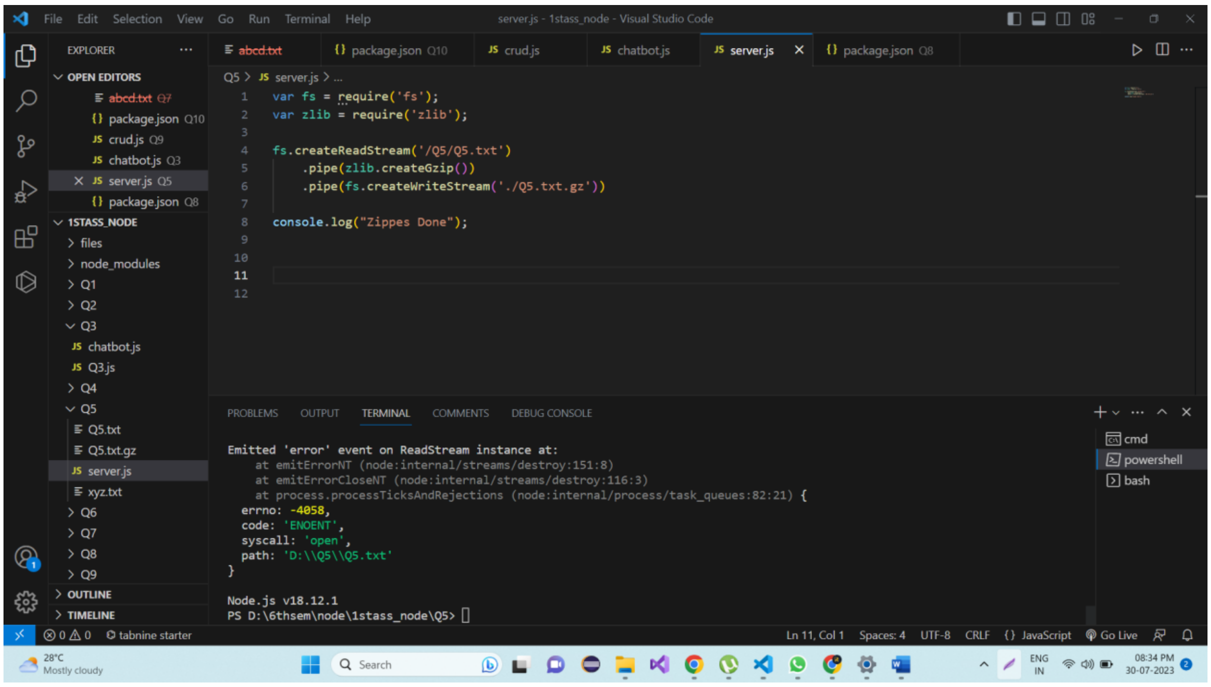
```
Emitted 'error' event on ReadStream instance at:
    at emitErrorNT (node:internal/streams/destroy:151:8)
    at emitErrorCloseNT (node:internal/streams/destroy:116:3)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21) {
  errno: -4058,
  code: 'ENOENT',
  syscall: 'open',
  path: 'D:\\Q5\\Q5.txt'
}

Node.js v18.12.1
PS D:\6thsem\node\1stass_node\Q5> 
```

6. Write a program to extract a zip file.

Decom.js

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

fs.createReadStream('../Q5/Q5.txt.gz')
    .pipe(zlib.createGunzip())
    .pipe(fs.createWriteStream('../Q5/xyz.txt', 'utf-8'))

console.log("dcompress done")
```

output:



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

fs.createReadStream('/Q5/Q5.txt')
    .pipe(zlib.createGzip())
    .pipe(fs.createWriteStream('./Q5.txt.gz'))

console.log("Zippes Done");
```

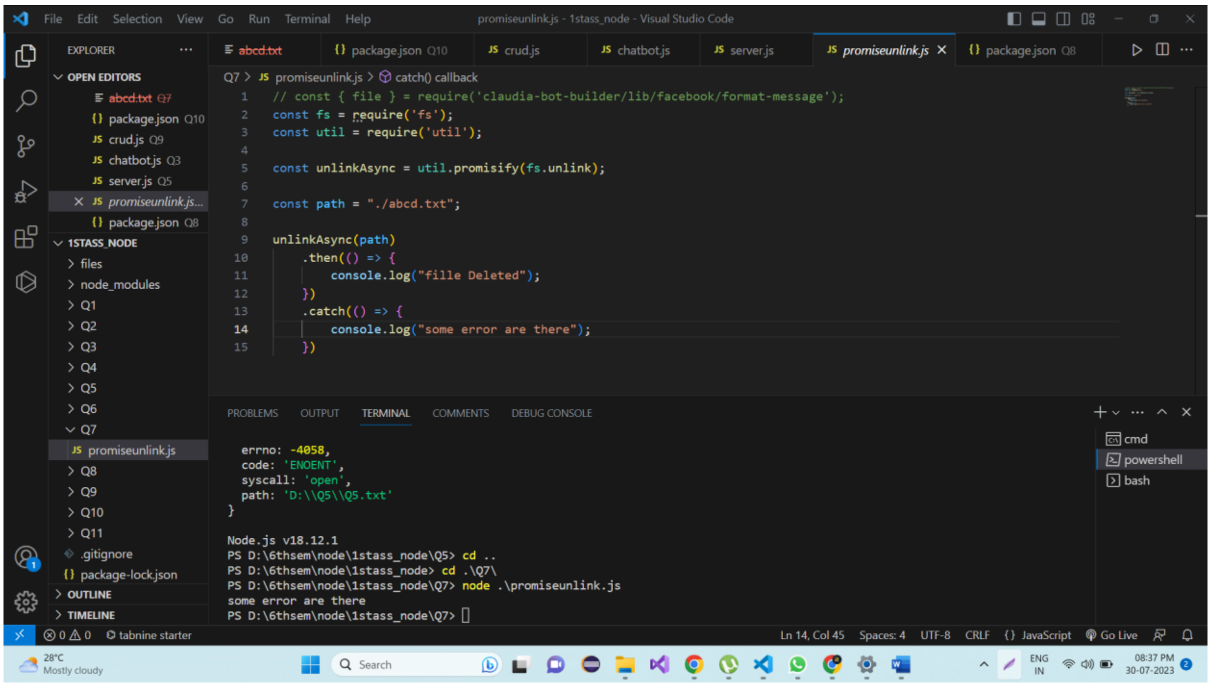
```
Emitted 'error' event on ReadStream instance at:
    at emitErrorNT (node:internal/streams/destroy:151:8)
    at emitErrorCloseNT (node:internal/streams/destroy:116:3)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21) {
  errno: -4058,
  code: 'ENOENT',
  syscall: 'open',
  path: 'D:\\Q5\\Q5.txt'
}

Node.js v18.12.1
PS D:\6thsem\node\1stass_node\Q5>
```

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

Promiseunlink.js

```
// const { file } = require('claudia-bot-builder/lib/facebook/format-
message');
const fs = require('fs');
const util = require('util');

const unlinkAsync = util.promisify(fs.unlink);

const path = "./abcd.txt";

unlinkAsync(path)
    .then(() => {
        console.log("fille Deleted");
    })
    .catch(() => {
        console.log("some error are there");

    })
```

Output:



```javascript
// const { file } = require('claudia-bot-builder/lib/facebook/format-message');
const fs = require('fs');
const util = require('util');

const unlinkAsync = util.promisify(fs.unlink);

const path = "./abcd.txt";

unlinkAsync(path)
    .then(() => {
        console.log("fille Deleted");
    })
    .catch(() => {
        console.log("some error are there");
    })
```

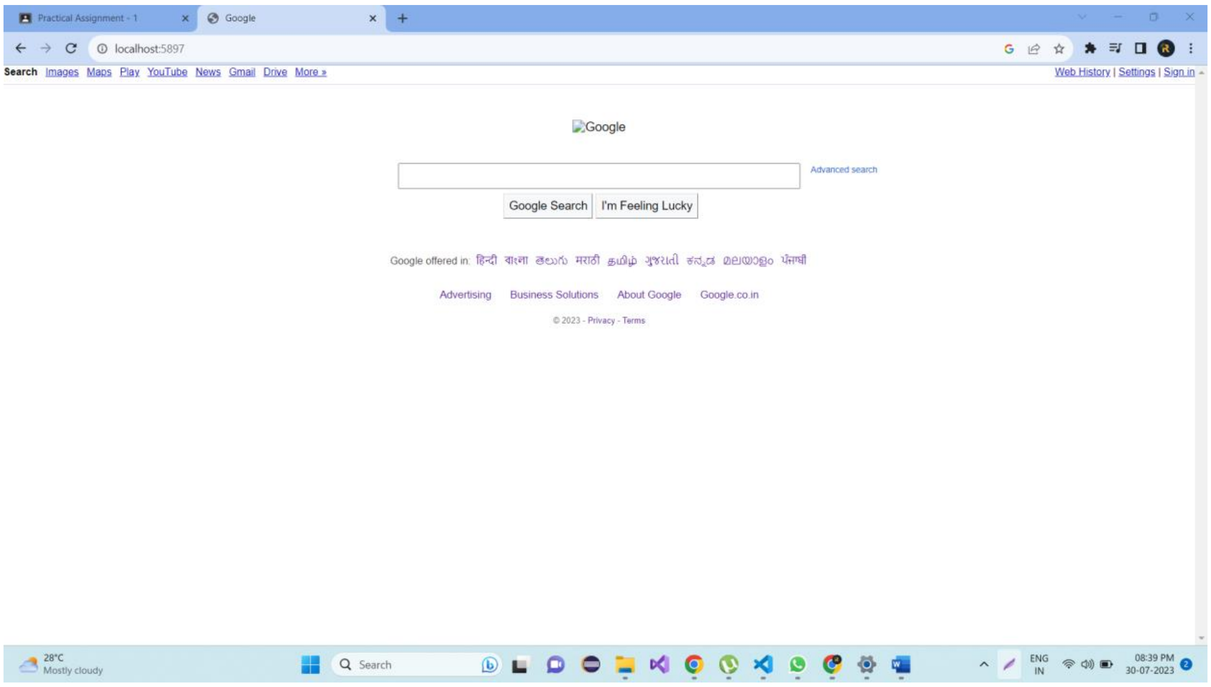
```
  errno: -4058,
  code: 'ENOENT',
  syscall: 'open',
  path: 'D:\\Q5\\Q5.txt'
}

Node.js v18.12.1
PS D:\6thsem\node\1stass_node\Q5> cd ..
PS D:\6thsem\node\1stass_node> cd .\Q7\
PS D:\6thsem\node\1stass_node\Q7> node .\promiseunlink.js
some error are there
PS D:\6thsem\node\1stass_node\Q7> 
```

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

Q8.js

```javascript
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(5897, () => {
    console.log("Listing on 5897");
})
```

Output:


Practical Assignment - 1
Google
localhost:5897
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
28°C Mostly cloudy
Search
ENG IN
08:39 PM 30-07-2023

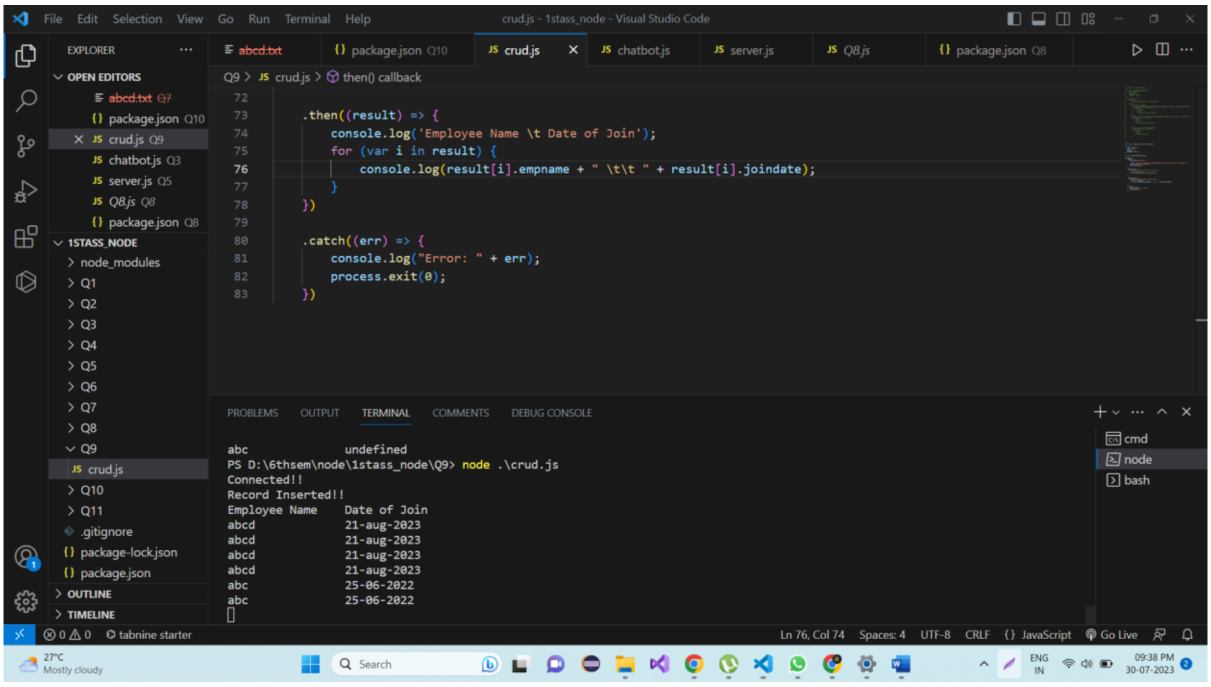
9. Write a program that connect Mysql database, Insert a record in employee table and display all records in employee table using promise based approach.

Crud.js

```
const mysql = require('nodejs-mysql').default;

const conn = ({
    host: "localhost",
    user: "root",
    password: "",
    database: "Node_ass"
});

const db = mysql.getInstance(conn);

db.connect()
    .then(() => {
        console.log(`Connected!!`)

        var sql = "INSERT INTO employeetb (empid,empname,joinDate) VALUES
(201,'abc','25-06-2022')";
        console.log("Record Inserted!!");
        return db.exec(sql);
    })
```

```javascript
    .then((display) => {
        // var sqlDisplay = "SELECT * FROM employeetb";
        // console.log(display);
        return db.exec("SELECT * FROM employeetb");

    })

    .then((result) => {
        console.log('Employee Name \t Date of Join');
        for (var i in result) {
            console.log(result[i].empname + " \t\t " + result[i].joindate);
        }
    })

    .catch((err) => {
        console.log("Error: " + err);
        process.exit(0);
    })
```

Output:



```javascript
    .then((result) => {
        console.log('Employee Name \t Date of Join');
        for (var i in result) {
            console.log(result[i].empname + " \t\t " + result[i].joindate);
        }
    })

    .catch((err) => {
        console.log("Error: " + err);
        process.exit(0);
    })
```

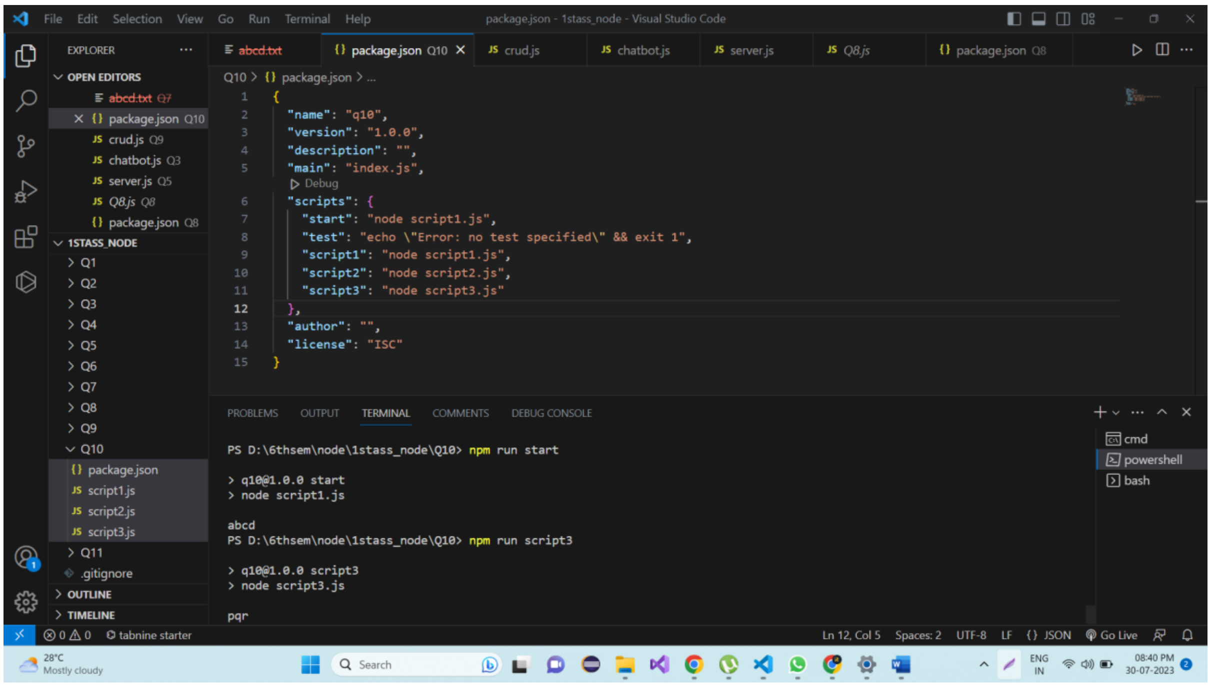
```
abc                 undefined
PS D:\6thsem\node\1stass_node\Q9> node .\crud.js
Connected!!
Record Inserted!!
Employee Name       Date of Join
abcd                21-aug-2023
abcd                21-aug-2023
abcd                21-aug-2023
abcd                21-aug-2023
abc                 25-06-2022
abc                 25-06-2022
```

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

Pacakage.json

```json
{
  "name": "q10",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node script1.js",
    "test": "echo \"Error: no test specified\" && exit 1",
    "script1": "node script1.js",
    "script2": "node script2.js",
    "script3": "node script3.js"
  },
  "author": "",
  "license": "ISC"
}
```

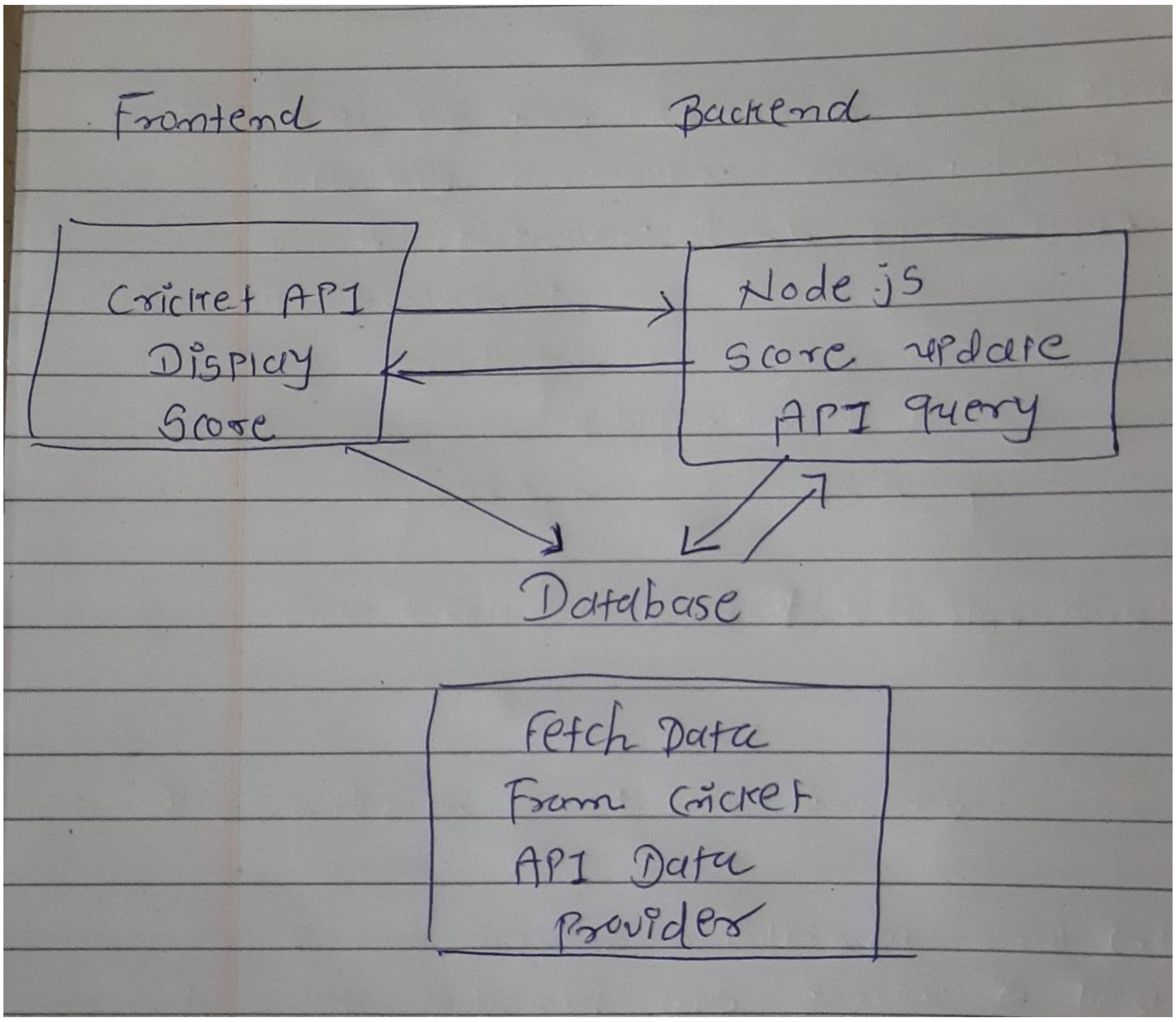
Output:



```json
{
  "name": "q10",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node script1.js",
    "test": "echo \"Error: no test specified\" && exit 1",
    "script1": "node script1.js",
    "script2": "node script2.js",
    "script3": "node script3.js"
  },
  "author": "",
  "license": "ISC"
}
```

```
PS D:\6thsem\node\1stass_node\Q10> npm run start

> q10@1.0.0 start
> node script1.js

abcd
PS D:\6thsem\node\1stass_node\Q10> npm run script3

> q10@1.0.0 script3
> node script3.js

pqr
```

11. Develop an application to show live cricket score.


Frontend
Backend
Cricket API
Display
Score
Node js
Score update
API query
Database
Fetch Data
From Cricket
API Data
Provider