Practical Assignment 1

Name: Isha Dadawala

Roll No.: 008

Semester: 7th

Subject: [702] Application Development using Full Stack

Submission of: Practical Assignment 1

1. Develop a web server with following functionalities:

- Serve static resources.

- Handle GET request.

- Handle POST request.

Solution:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Form</title>

</head>

<body>

    <h1>GET Form</h1>

        <form action="/process" method="GET">

            Username: <input type="text" name="uname"><br/>

            Age: <input type="text" name="age"><br/>

            <input type="submit" value="Submit">

        </form>

    <h1>POST Form</h1>

        <form action="/process" method="POST">

            Username: <input type="text" name="uname"><br/>

            Age: <input type="text" name="age"><br/>\

            <input type="submit" value="Submit">

        </form>

</body>

</html>

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 = "./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("Username: " + ul.query.uname + " \nAge: " + ul.query.age);

        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);

        });

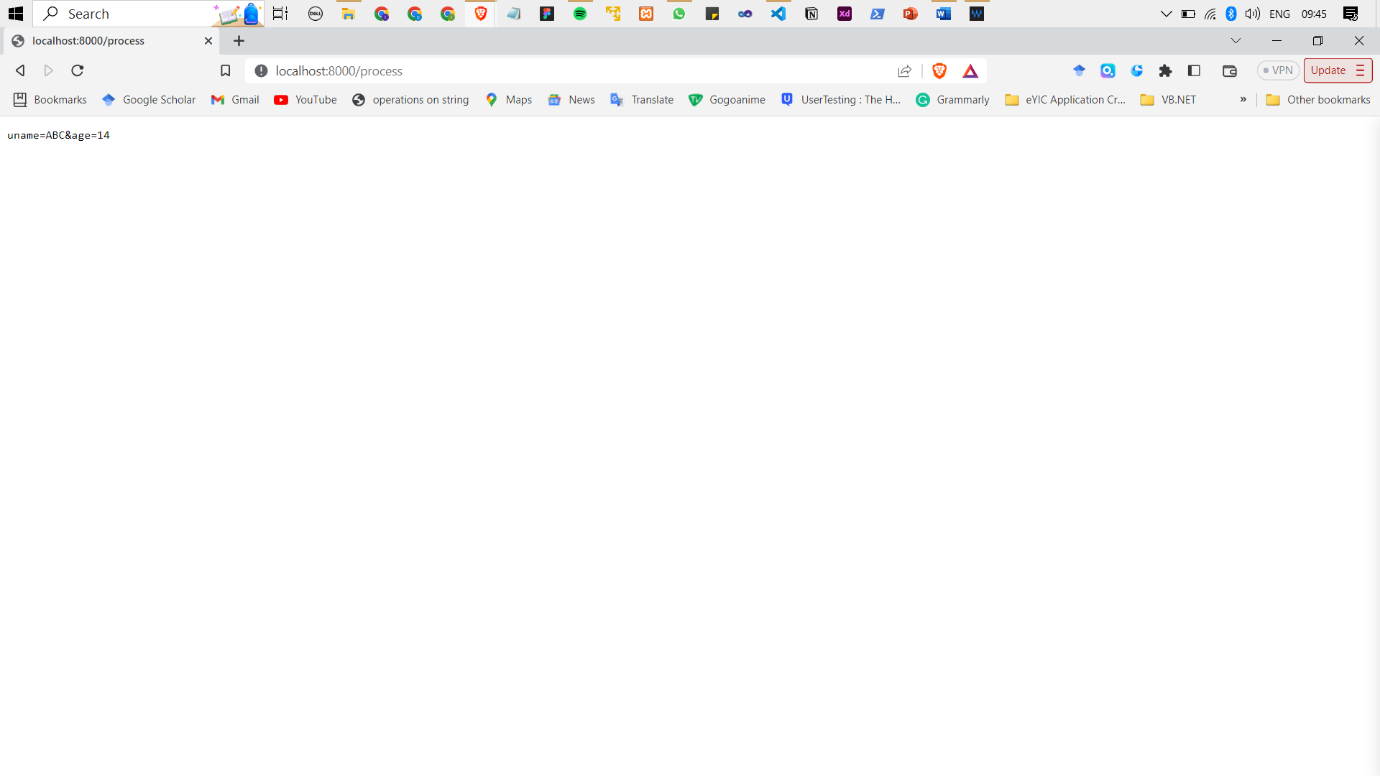
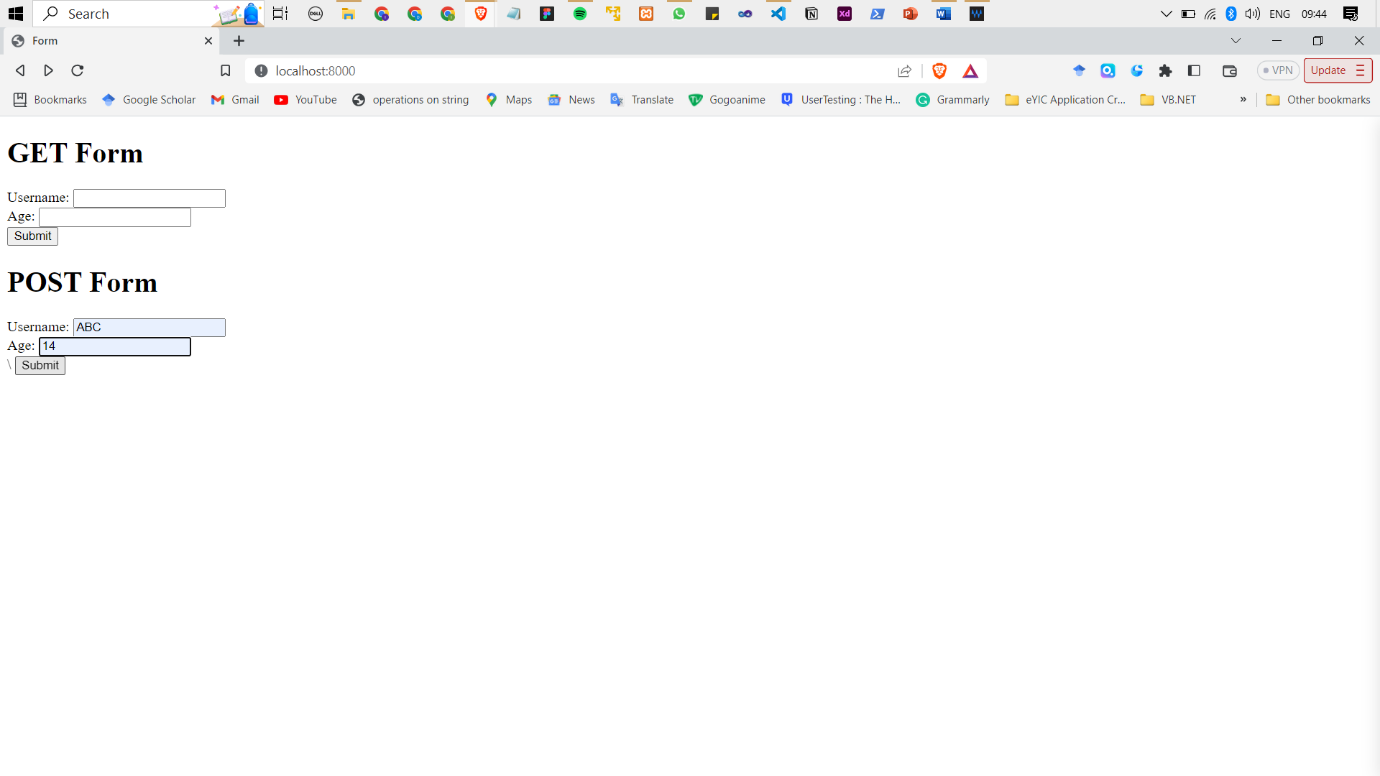
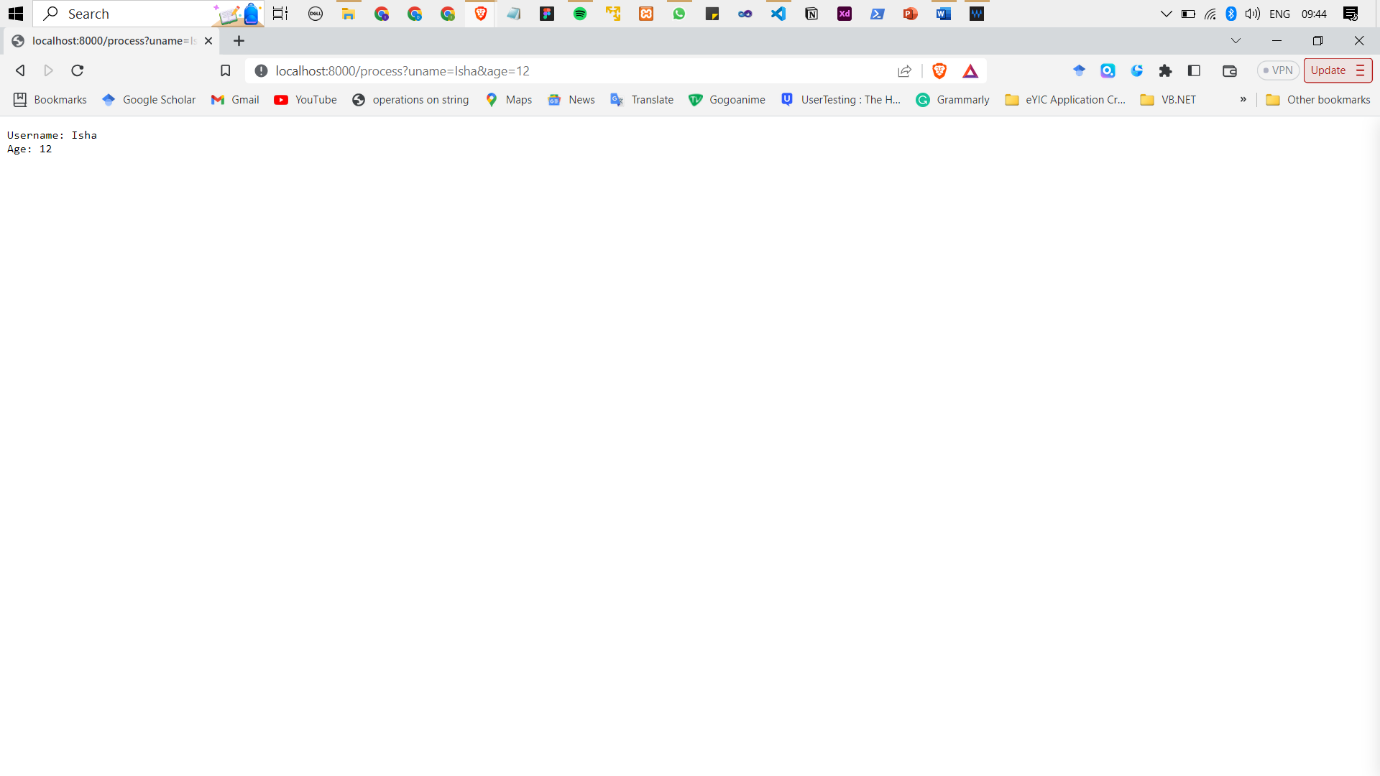
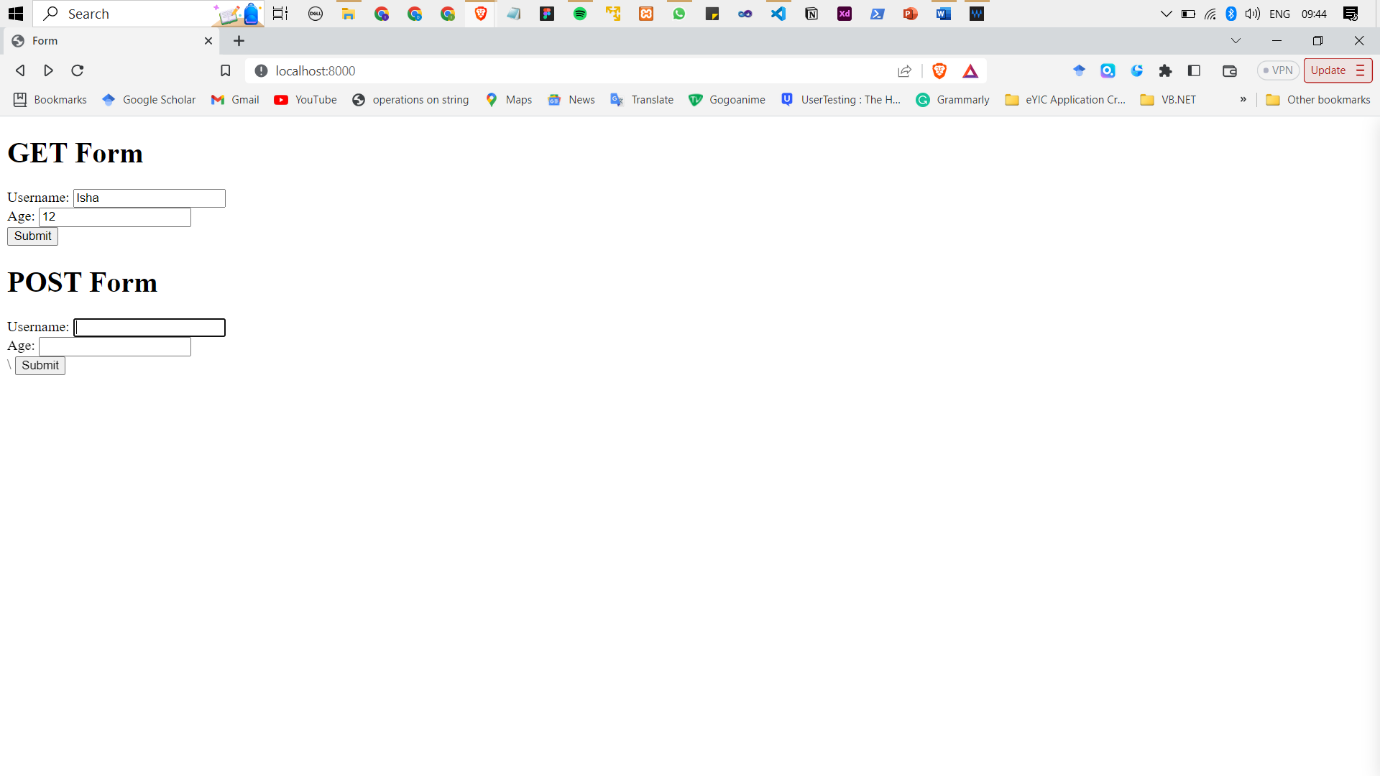
    }

});

server.listen(8000);

    console.log("Server running on: 8000");

Output:



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.)

Solution:

server.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/file1.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 === '/call') {

            fs.readFile('./files/index.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 running on: 8000");

})

In the files folder within the directory:-

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>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>

file1.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>

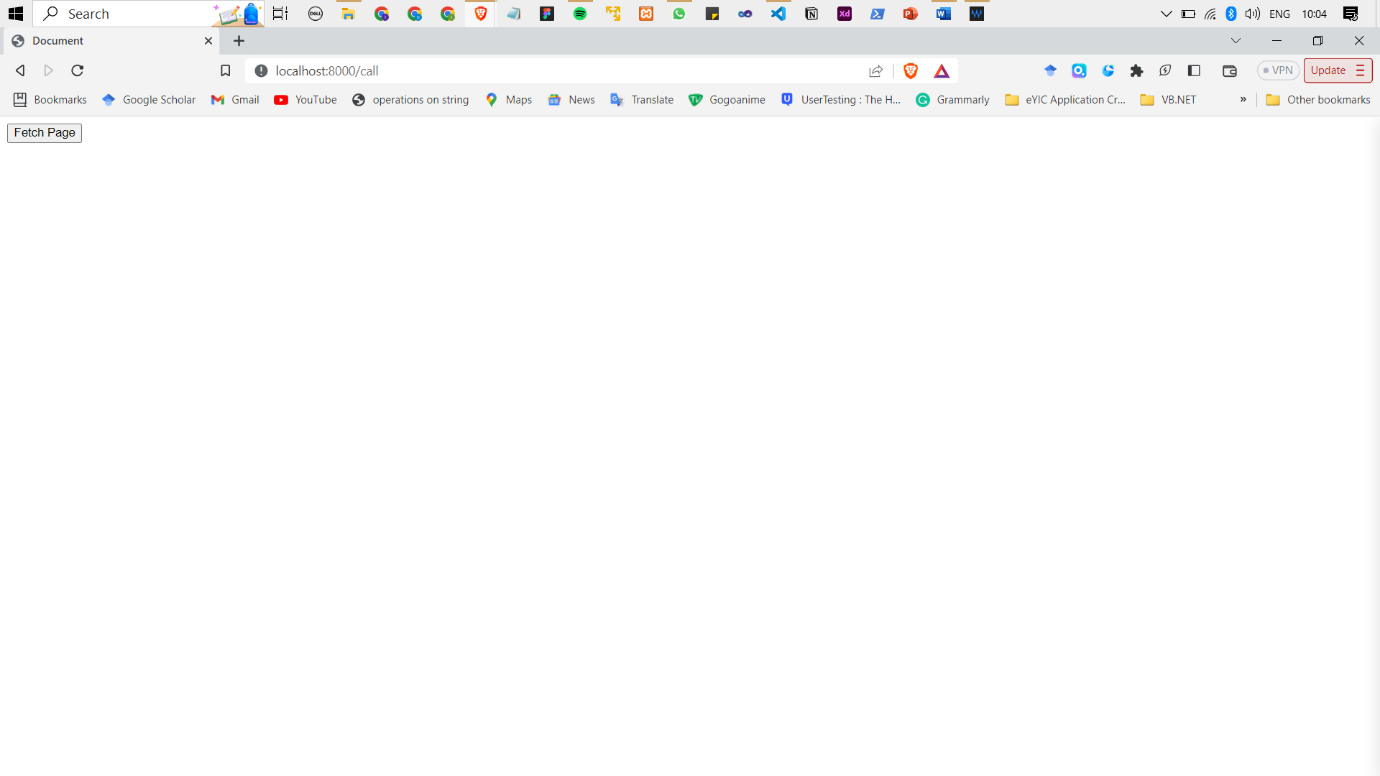
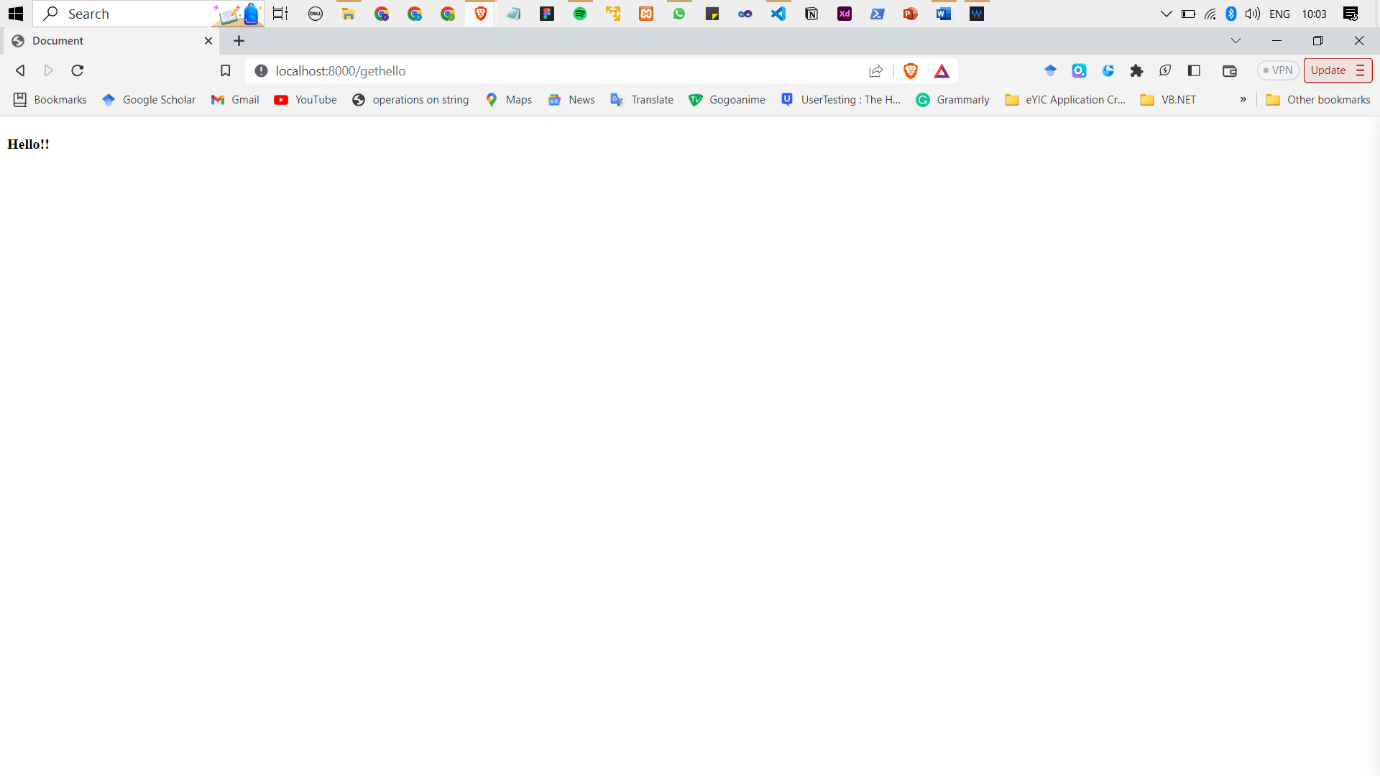
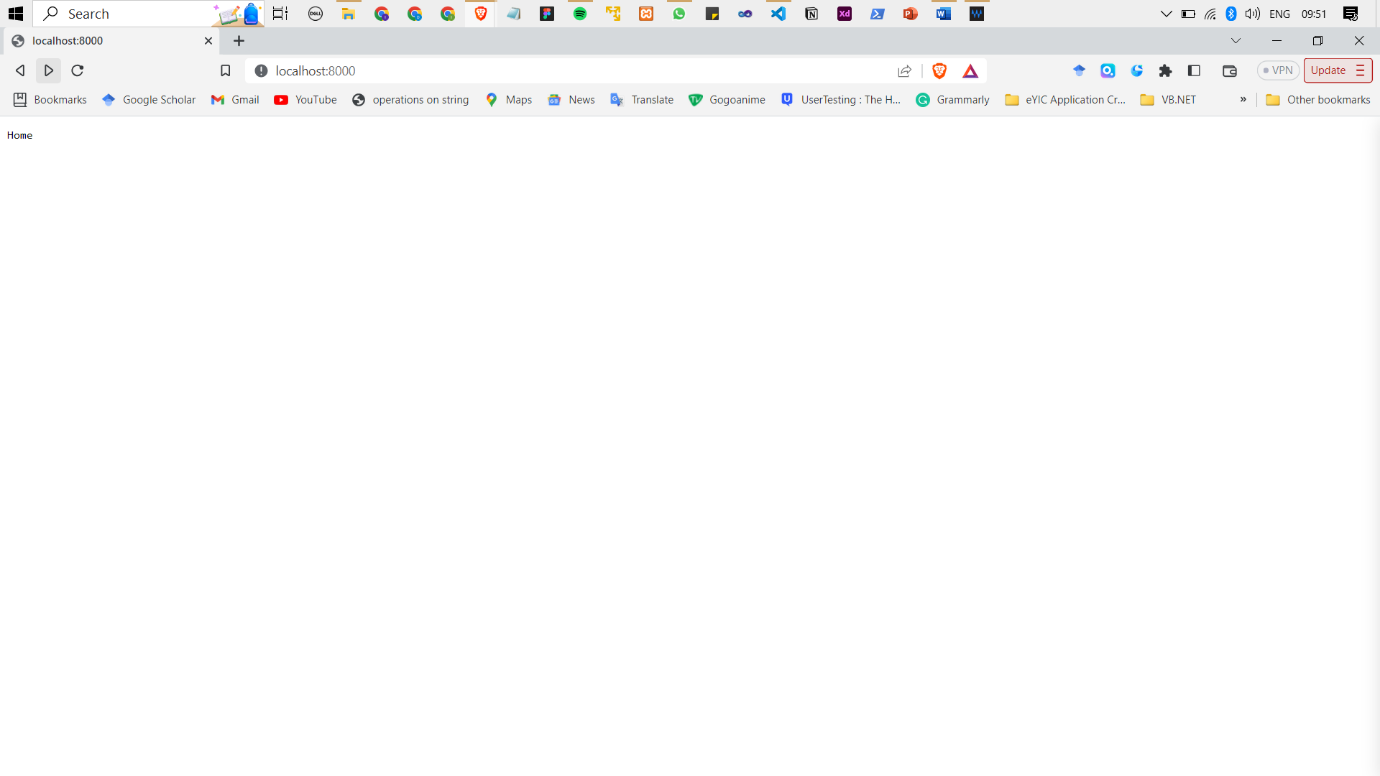
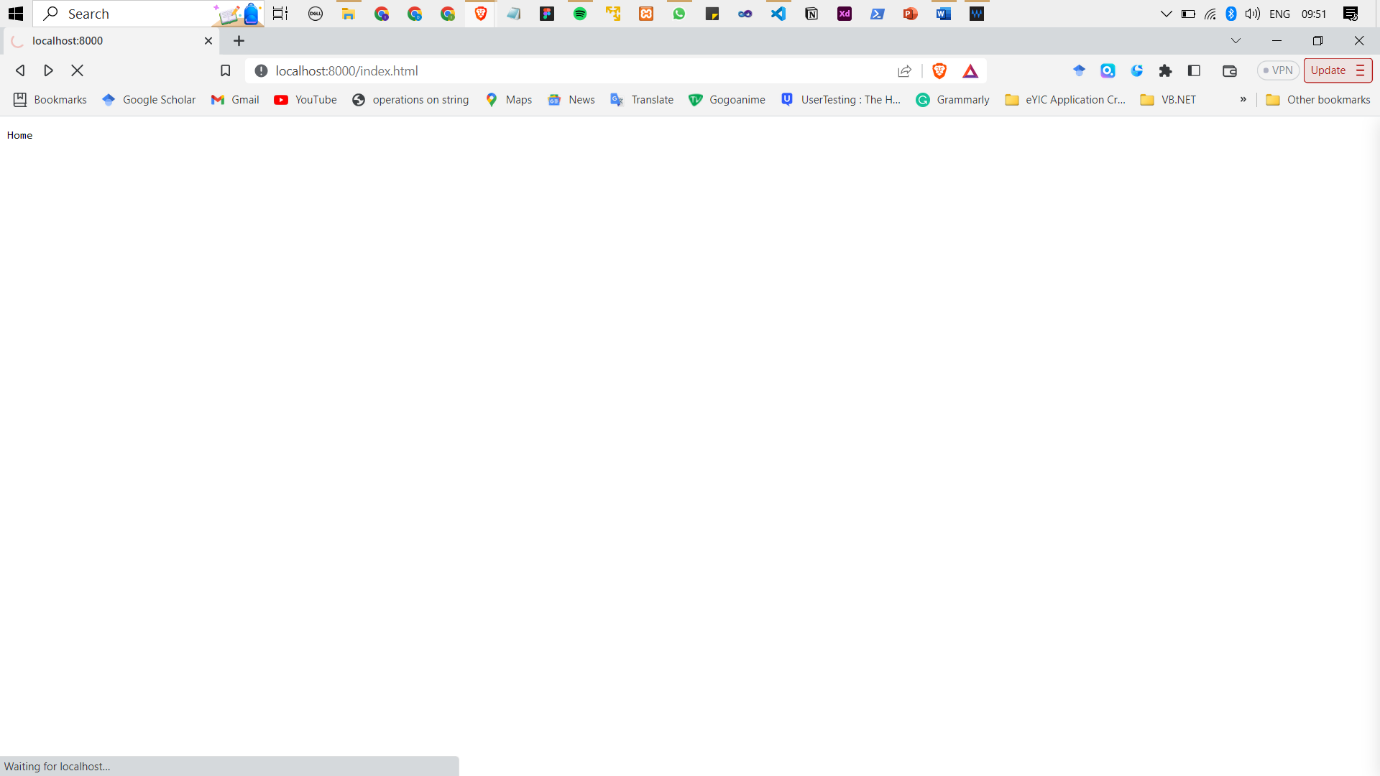
<body>

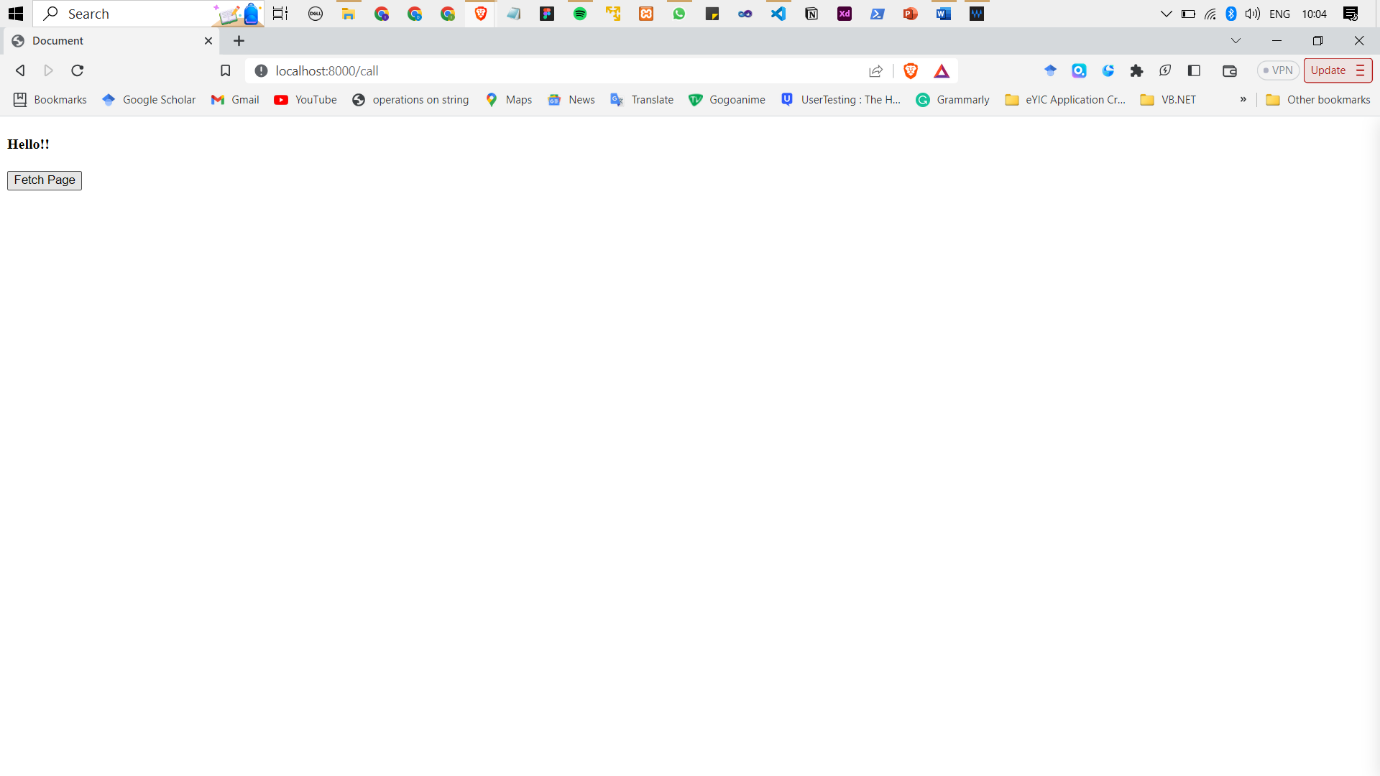
    <h4>Hello!!</h4>

</body>

</html>

Output:





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

Solution:

app.js

const readline = require("readline");

const { handleUserInput } = require("./chatbot");

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout,

});

function startChat() {

  rl.question(

    "Chatbot: Hi there! Please type in whatever you want to ask me. ",

    (userInput) => {

      if (userInput.trim() === "exit") {

        console.log("Chatbot: Bye! Have a great day!");

        rl.close();

      } else {

        const response = handleUserInput(userInput);

        console.log(`Chatbot: ${response}`);

        startChat();

      }

    }

  );

}

startChat();

chatbot.js

function handleUserInput(userInput) {

    const input = userInput.toLowerCase();

    if (input.indexOf("hello") !== -1 || input.indexOf("hi") !== -1) {

      return "Hi there! Please type in whatever you want to ask me.";

    } else if (input.indexOf("bye") !== -1 || input.indexOf("exit") !== -1) {

      return "Bye! Have a great day!";

    } else if (input.indexOf("what is your name") !== -1) {

      return "I am Chatbot";

    } else if (input.indexOf("what time is it") !== -1) {

      const currentTime = new Date().toLocaleTimeString();

      return `The current time is: ${currentTime}`;

    } else if (input.indexOf("how are you") !== -1) {

      return "I am a chatbot & i don't feel anything, but thanks for asking!";

    } else if (input.indexOf("tell me a joke") !== -1) {

      return "Why did a scarecrow win a Nobel? He was outstanding in his field!";

    } else {

      return "I'm sorry, I didn't understand that.";

    }

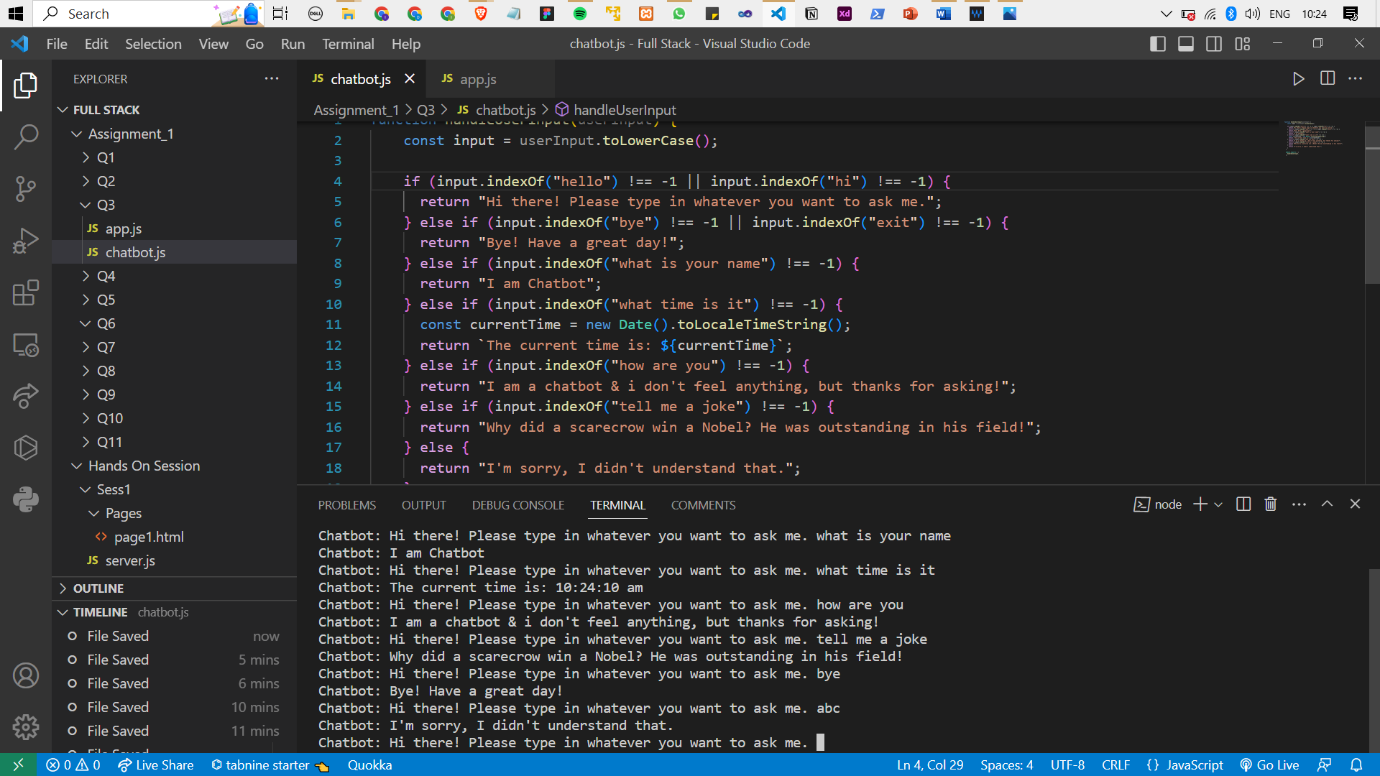
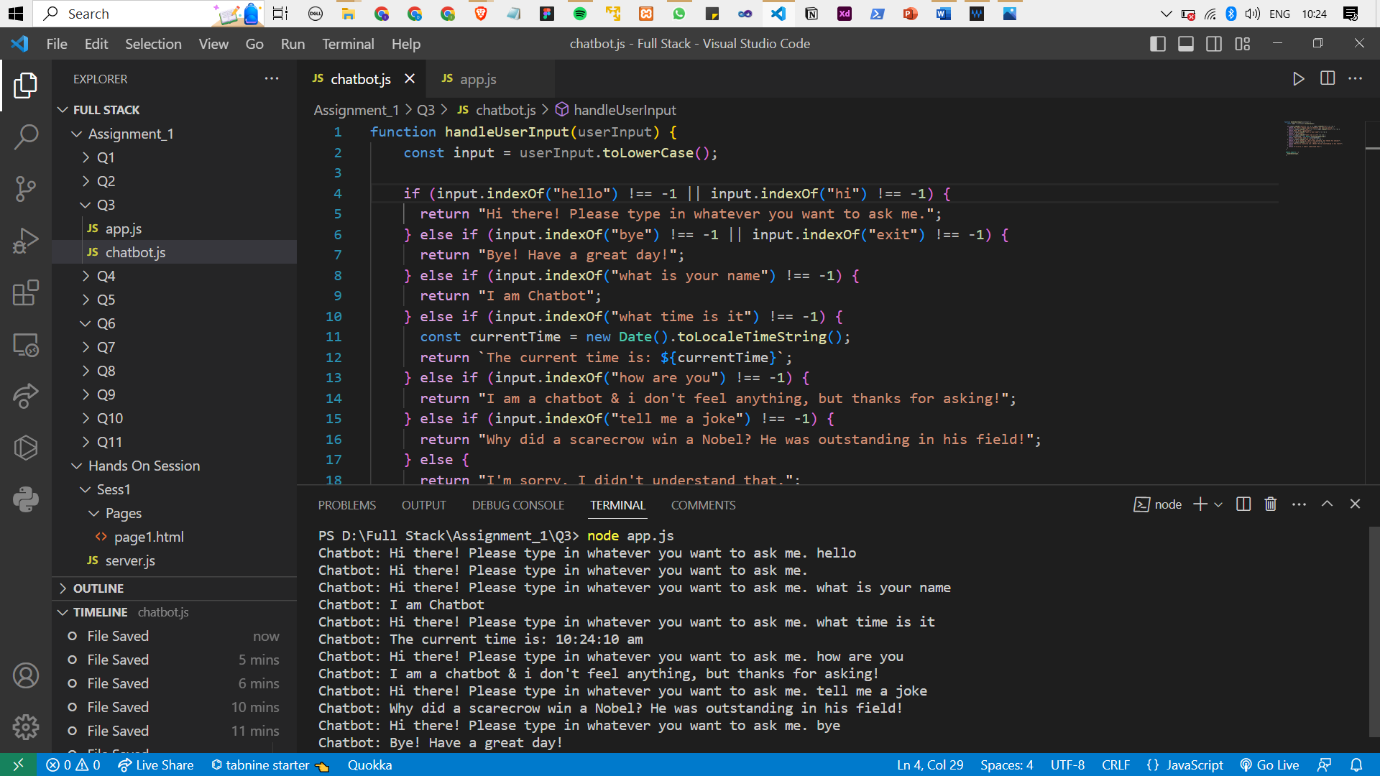
  }

  module.exports = {

    handleUserInput,

  };

Output:



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

Solution:

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("./files/index.html",(err,data)=>{

        response.write(data)

        response.end();

    })

  }

}).listen(8000, function() {

    console.log((new Date()) +

      ' Server running on: 8000 ');

});

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)

  })

})

In the files folder within the directory:-

index.html:

<!DOCTYPE html >

<html>

    <body>

<script language="javascript">

var ws = new WebSocket('ws://localhost:8000');

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>