

## BACKEND: MERN STACK BOOTCAMP

Assignment No: 02

Name: Romaisa Amjad

Date: 6/4/2025

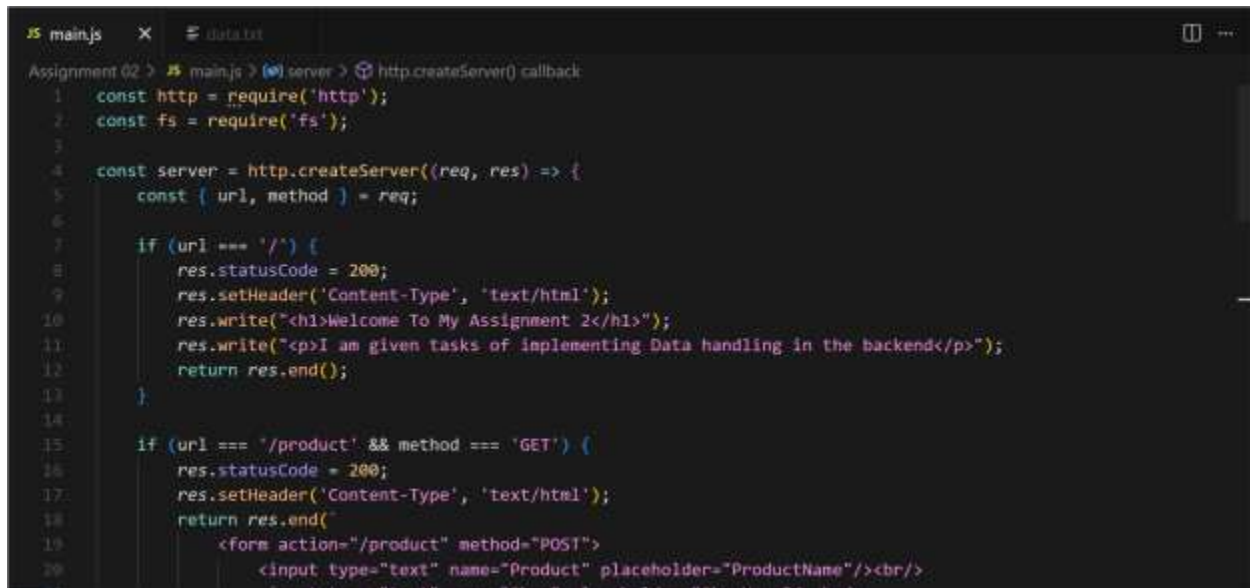
### Question:

Create a basic web server using core Node JS that can listen to requests and response to it. The server should have the following responses on different routes

1. **/:** Return a heading and a paragraph.
2. **GET: /product:** Return a form having an input field of **ProductName** and **ShopName**.
3. Hit **POST: /product** on form submission to send a post request to **/product**
4. **POST /product:** Parse the request data and save it in a file and also return html showing that data.

Handle the rest of routes by sending **404** status code and **Page Not Found** response.

### Code Implementation:

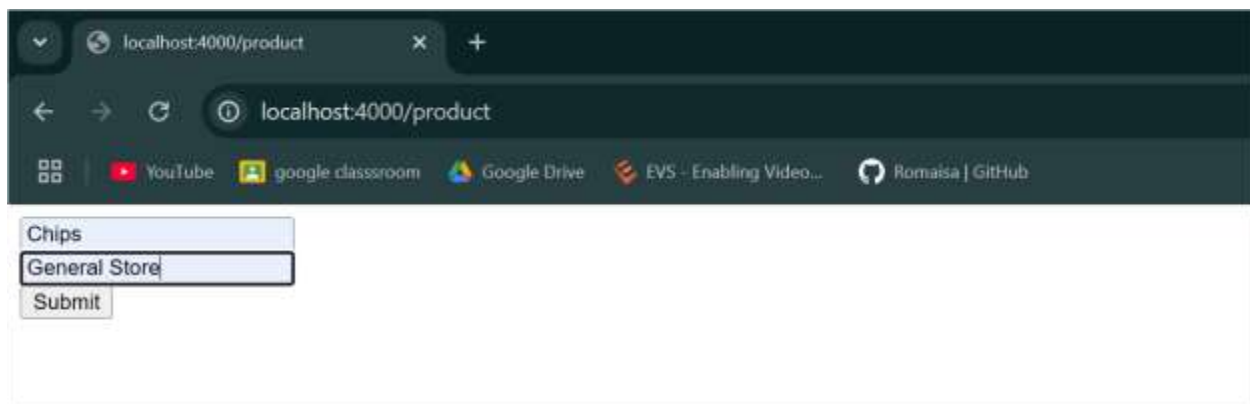
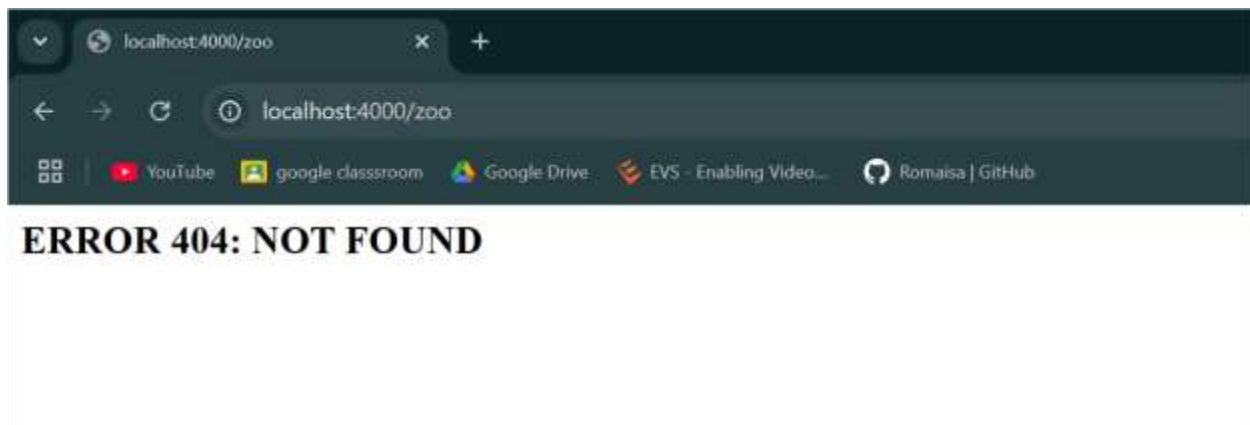
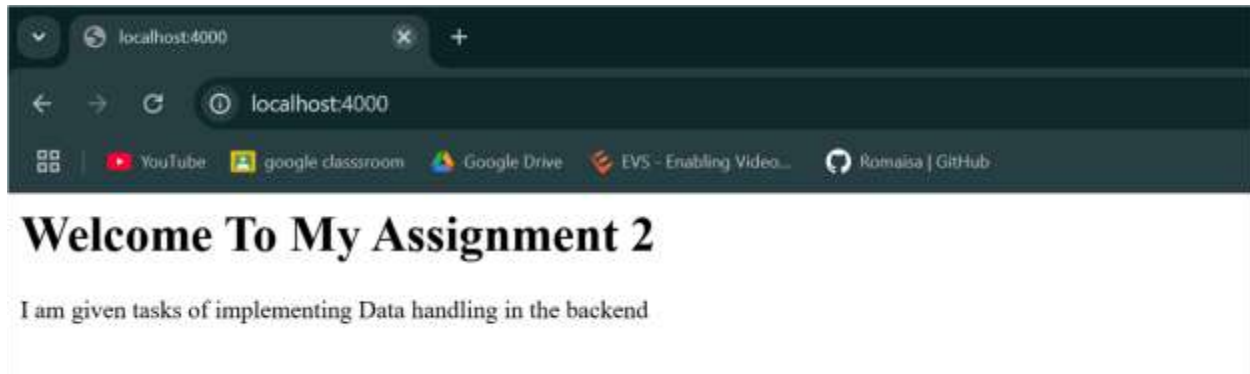


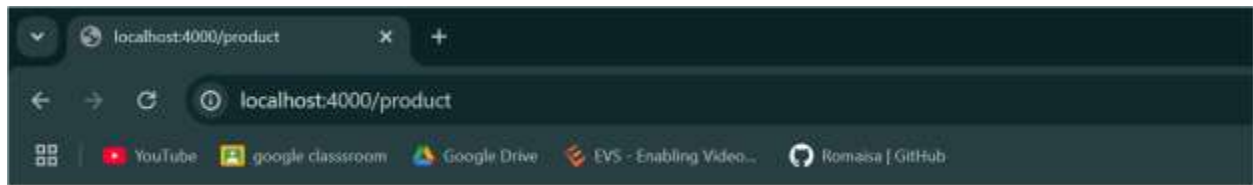
```
main.js x data.txt
Assignment 02 > main.js > server > http.createServer() callback
1  const http = require('http');
2  const fs = require('fs');
3
4  const server = http.createServer((req, res) => {
5    const { url, method } = req;
6
7    if (url === '/') {
8      res.statusCode = 200;
9      res.setHeader('Content-Type', 'text/html');
10     res.write("<h1>Welcome To My Assignment 2</h1>");
11     res.write("<p>I am given tasks of implementing Data handling in the backend</p>");
12     return res.end();
13   }
14
15   if (url === '/product' && method === 'GET') {
16     res.statusCode = 200;
17     res.setHeader('Content-Type', 'text/html');
18     return res.end(
19       "<form action='/product' method='POST'>
20         <input type='text' name='Product' placeholder='ProductName'><br/>
21         <input type='text' name='Shop' placeholder='ShopName'><br/>
22         <input type='submit' value='Submit'>
23       </form>
24     );
25   }
26
27   res.statusCode = 404;
28   res.setHeader('Content-Type', 'text/html');
29   res.write("<h1>Page Not Found</h1>");
30   return res.end();
31 }
```

```
main.js x data.txt
Assignment 02 > main.js > server > http.createServer() callback
4 const server = http.createServer((req, res) => {
21     <input type="text" name="Shop" placeholder="ShopName"/><br/>
22     <input type="submit"/>
23     </form>
24     ');
25 }
26
27 if (url === '/product' && method === 'POST') {
28     res.statusCode = 200;
29     const data = [];
30
31     req.on("data", (chunk) => {
32         data.push(chunk);
33     });
34
35     req.on("end", () => {
36         const result = Buffer.concat(data).toString();
37         const parse = new URLSearchParams(result); // for individual key value
38         const productname = parse.get("Product");
39         const shopname = parse.get("Shop");
```

```
main.js x data.txt
Assignment 02 > main.js > server > http.createServer() callback
4 const server = http.createServer((req, res) => {
35     req.on("end", () => {
40         fs.writeFileSync("data.txt", result);
41         res.setHeader("Content-Type", "text/html");
42         res.end(`<h2>Your Data:<br/>Prouct Name:${productname}<br/>Shop Name:${shopname}</h2>`);
43     });
44
45     return;
46 }
47
48 else{
49     res.setHeader("Content-Type", "text/html");
50     res.statusCode = 404;
51     res.end(`<h2>ERROR 404: NOT FOUND</h2>`);
52 }
53 });
54
55 server.listen(4000);
56
```

## Output:





**Your Data:**

**Product Name:Chips**

**Shop Name:General Store**

