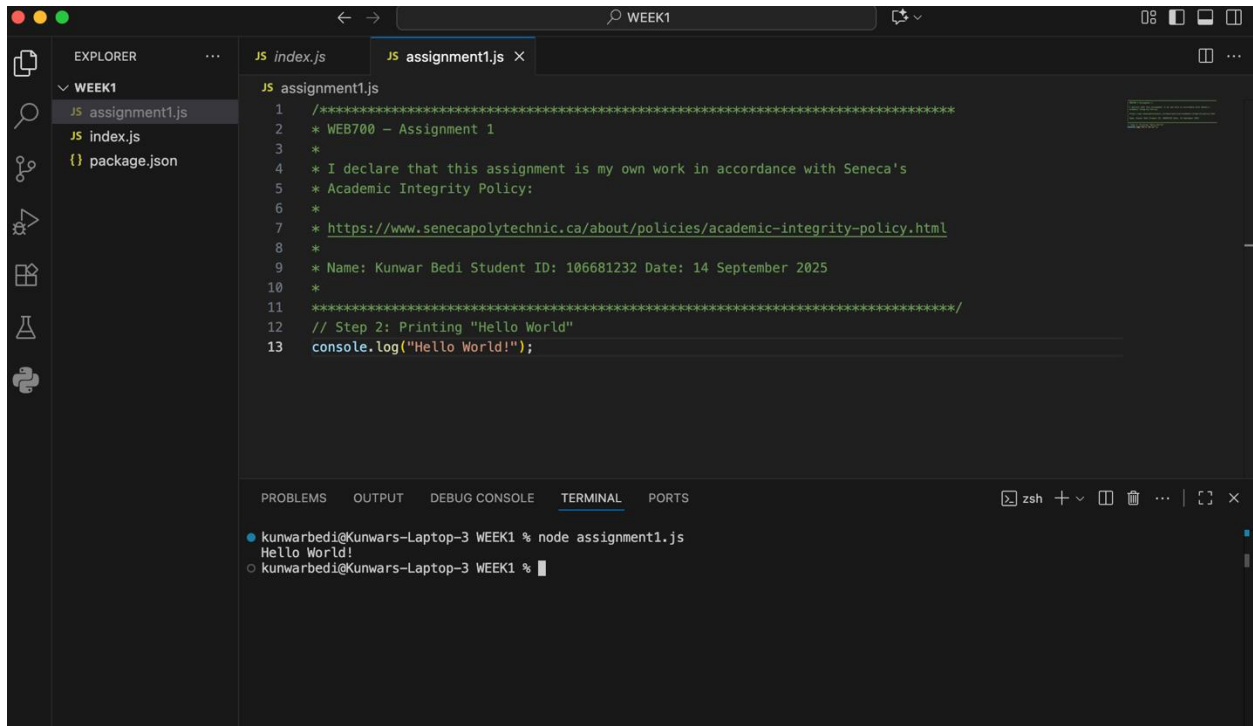


# ASSIGNMENT 1

Kunwar Bedi- 106681232

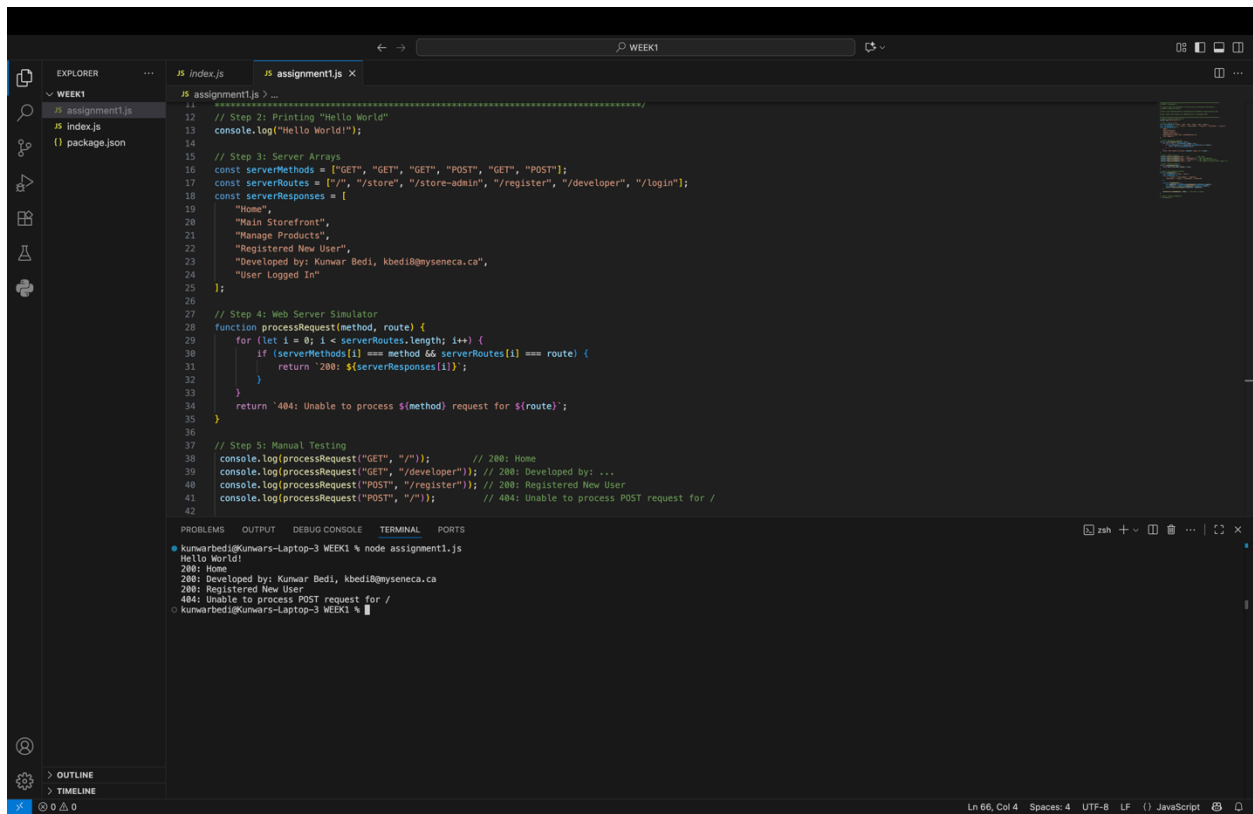


The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'WEEK1' containing 'assignment1.js', 'index.js', and 'package.json'. The main editor window is open to 'assignment1.js', which contains the following code:

```
1  /*****
2  * WEB700 – Assignment 1
3  *
4  * I declare that this assignment is my own work in accordance with Seneca's
5  * Academic Integrity Policy:
6  *
7  * https://www.senecapolytechnic.ca/about/policies/academic-integrity-policy.html
8  *
9  * Name: Kunwar Bedi Student ID: 106681232 Date: 14 September 2025
10 *
11 *****/
12 // Step 2: Printing "Hello World"
13 console.log("Hello World!");
```

Below the editor, the TERMINAL panel is active, showing the command `node assignment1.js` being executed, which results in the output `Hello World!`.

“Hello World”



```
11 // =====
12 // Step 2: Printing "Hello World"
13 console.log("Hello World!");
14
15 // Step 3: Server Arrays
16 const serverMethods = ["GET", "GET", "GET", "POST", "GET", "POST"];
17 const serverRoutes = ["/", "/store", "/store-admin", "/register", "/developer", "/login"];
18 const serverResponses = [
19     "Home",
20     "Main Storefront",
21     "Manage Products",
22     "Registered New User",
23     "Developed by: Kumar Bedi, kbedi@myseneca.ca",
24     "User Logged In"
25 ];
26
27 // Step 4: Web Server Simulator
28 function processRequest(method, route) {
29     for (let i = 0; i < serverRoutes.length; i++) {
30         if (serverMethods[i] === method && serverRoutes[i] === route) {
31             return `200: ${serverResponses[i]}`;
32         }
33     }
34     return `404: Unable to process ${method} request for ${route}`;
35 }
36
37 // Step 5: Manual Testing
38 console.log(processRequest("GET", "/")); // 200: Home
39 console.log(processRequest("GET", "/developer")); // 200: Developed by: ...
40 console.log(processRequest("POST", "/register")); // 200: Registered New User
41 console.log(processRequest("POST", "/")); // 404: Unable to process POST request for /
42
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
kunwarbedi@Kumwars-Laptop-3 WEEK1 % node assignment1.js
Hello World!
200: Home
200: Developed by: Kumar Bedi, kbedi@myseneca.ca
200: Registered New User
404: Unable to process POST request for /
kunwarbedi@Kumwars-Laptop-3 WEEK1 %
```

Before testRequests()

```
15 // Step 3: Server Arrays
16 const serverMethods = ["GET", "GET", "GET", "POST", "GET", "POST"];
17 const serverRoutes = ["/", "/store", "/store-admin", "/register", "/developer", "/login"];
18 const serverResponses = [
19   "Home",
20   "Main Storefront",
21   "Manage Products",
22   "Registered New User",
23   "Developed by: Kunwar Bedi, kbedi@myseneca.ca",
24   "User Logged In"
25 ];
26
27 // Step 4: Web Server Simulator
28 function processRequest(method, route) {
29   for (let i = 0; i < serverRoutes.length; i++) {
30     if (serverMethods[i] === method && serverRoutes[i] === route) {
31       return `200: ${serverResponses[i]}`;
32     }
33   }
34   return `404: Unable to process ${method} request for ${route}`;
35 }
36
37 // Step 5: Manual Testing
38 console.log(processRequest("GET", "/")); // 200: Home
39 console.log(processRequest("GET", "/developer")); // 200: Developed by: ...
40 console.log(processRequest("POST", "/register")); // 200: Registered New User
41 console.log(processRequest("POST", "/")); // 404: Unable to process POST request for /
42
43 // Step 6: Utility Function
44 function getRandomInt(max) {
45   return Math.floor(Math.random() * max);
46 }
47
48 // Step 6: Test Requests Function
49 function testRequests() {
50   const testMethods = ["GET", "POST"];
51
52   // ... (rest of the function code) ...
53 }
54
55 // ... (rest of the file code) ...
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
kunwarbedi@Kumars-Laptop-3 WEEK1 % node assignment1.js
Hello World!
200: Home
200: Developed by: Kunwar Bedi, kbedi@myseneca.ca
200: Registered New User
404: Unable to process POST request for /
200: Home
404: Unable to process POST request for /notFound2
404: Unable to process POST request for /notFound2
404: Unable to process POST request for /notFound2
404: Unable to process POST request for /developer
404: Unable to process GET request for /login
404: Unable to process POST request for /developer
404: Unable to process GET request for /notFound2
404: Unable to process POST request for /store
```

After testRequests()