

1.

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Selam>node -v
v14.17.5

C:\Users\Selam>node -v
v16.13.0

C:\Users\Selam>
```

2.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Selam>node -v
v14.17.5

C:\Users\Selam>node -v
v16.13.0

C:\Users\Selam>npm install -g typescript

added 1 package, and audited 2 packages in 15s

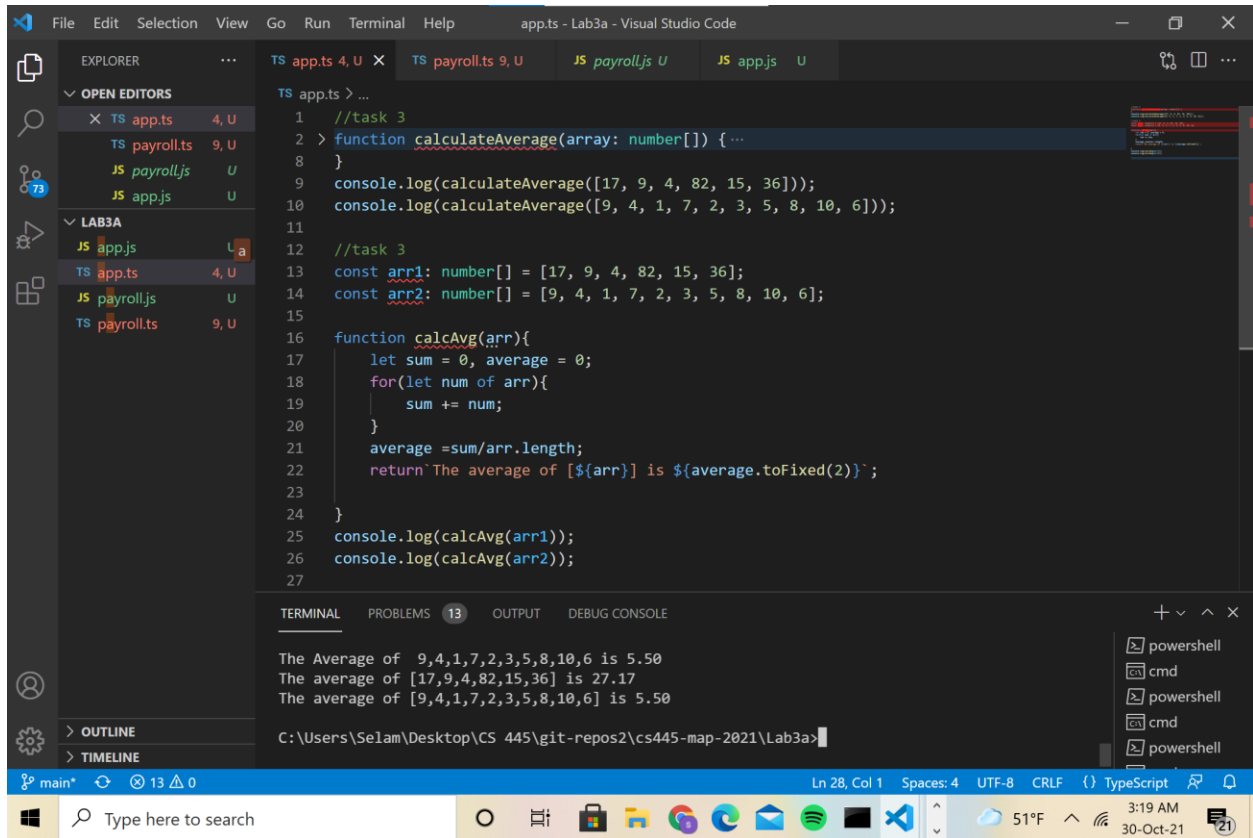
found 0 vulnerabilities

npm notice
npm notice New patch version of npm available! 8.1.0 -> 8.1.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.1.2
npm notice Run npm install -g npm@8.1.2 to update!
npm notice

C:\Users\Selam>tsc -v
Version 4.4.4

C:\Users\Selam>
```

3

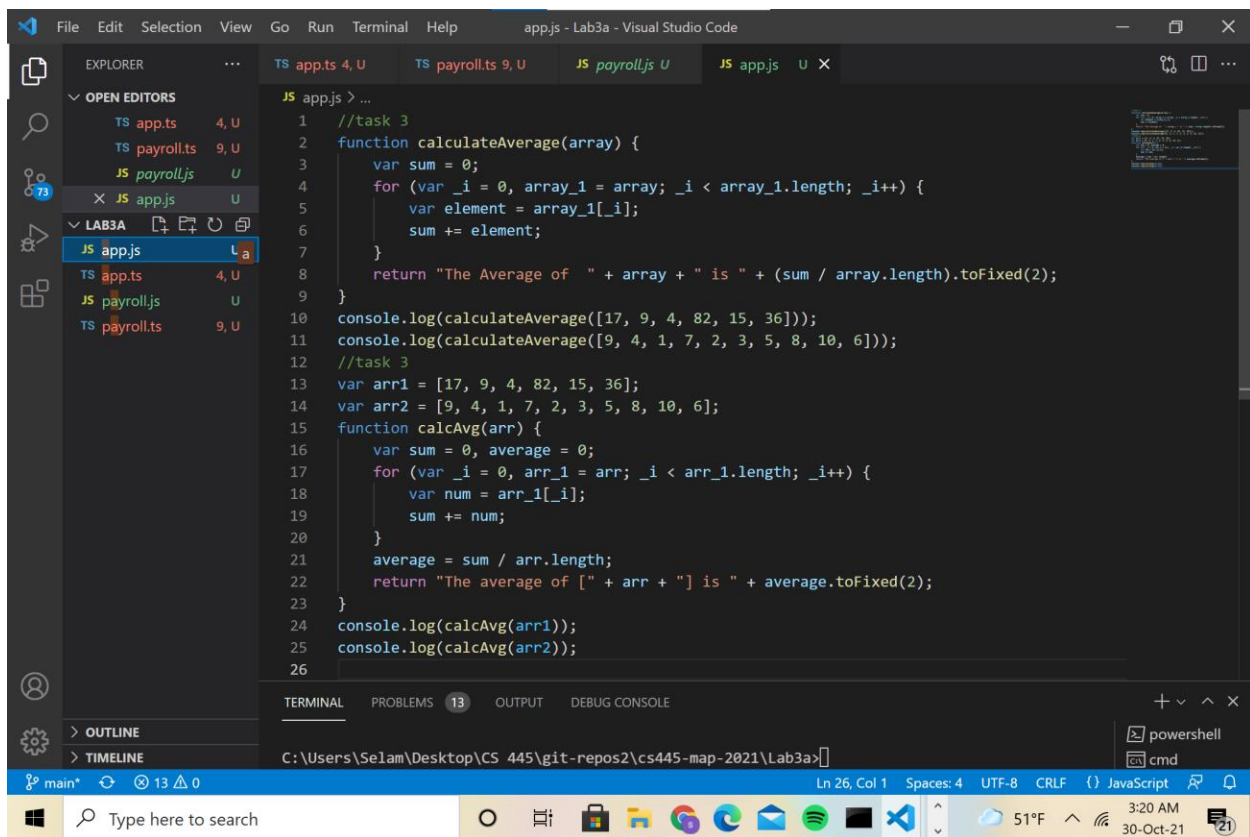


The screenshot displays the Visual Studio Code interface with the following components:

- Explorer:** Shows the project structure with files `app.ts`, `payroll.ts`, `payroll.js`, and `app.js`.
- Editor:** Displays the content of `app.ts`, which includes a function `calculateAverage` and a `calcAvg` function. The code is as follows:

```
1 //task 3
2 > function calculateAverage(array: number[]) { ...
8 }
9 console.log(calculateAverage([17, 9, 4, 82, 15, 36]));
10 console.log(calculateAverage([9, 4, 1, 7, 2, 3, 5, 8, 10, 6]));
11
12 //task 3
13 const arr1: number[] = [17, 9, 4, 82, 15, 36];
14 const arr2: number[] = [9, 4, 1, 7, 2, 3, 5, 8, 10, 6];
15
16 function calcAvg(arr){
17     let sum = 0, average = 0;
18     for(let num of arr){
19         sum += num;
20     }
21     average =sum/arr.length;
22     return`The average of [${arr}] is ${average.toFixed(2)}`;
23 }
24
25 console.log(calcAvg(arr1));
26 console.log(calcAvg(arr2));
27
```
- Terminal:** Shows the output of the program execution:

```
The Average of 9,4,1,7,2,3,5,8,10,6 is 5.50
The average of [17,9,4,82,15,36] is 27.17
The average of [9,4,1,7,2,3,5,8,10,6] is 5.50
```
- Taskbar:** Shows the Windows taskbar with the search bar and various application icons.



4.

This screenshot shows the Visual Studio Code editor with the file payroll.js open. The Explorer sidebar on the left shows the project structure with files app.ts, payroll.ts, payroll.js, and app.js. The main editor area displays the payroll.js code, which defines two tax calculation functions, taxPayroll1 and taxPayroll2, and a taxRate object. The code uses console.log to output the results of the calculations. The status bar at the bottom indicates the current line and column (Ln 1, Col 1) and the file encoding (UTF-8).

```
1 var taxPayroll;
2 (function (taxPayroll) {
3   taxPayroll[taxPayroll["FICA"] = 10] = "FICA";
4   taxPayroll[taxPayroll["STATE"] = 5] = "STATE";
5   taxPayroll[taxPayroll["MEDICARE"] = 12] = "MEDICARE";
6 })(taxPayroll || (taxPayroll = {}));
7 console.log(taxPayroll.FICA, taxPayroll.STATE, taxPayroll.MEDICARE);
8 //222
9
10 var taxPayroll2;
11 (function (taxPayroll2) {
12   taxPayroll2[taxPayroll2["FICA"] = 0] = "FICA";
13   taxPayroll2[taxPayroll2["STATE"] = 1] = "STATE";
14   taxPayroll2[taxPayroll2["MEDICARE"] = 2] = "MEDICARE";
15 })(taxPayroll2 || (taxPayroll2 = {}));
16 ;
17 var tFICA = taxPayroll2.FICA;
18 var tSTATE = taxPayroll2.STATE;
19 var tMEDICARE = taxPayroll2.MEDICARE;
20 console.log("FICA tax: " + tFICA + ", STATE taxPayroll2: " + tSTATE + ", MEDICARE tax:" +
21 var taxRate;
22 (function (taxRate) {
23   taxRate[taxRate["FICA"] = 10] = "FICA";
24   taxRate[taxRate["STATE"] = 5] = "STATE";
25   taxRate[taxRate["MEDICAR"] = 12] = "MEDICAR";
26 })(taxRate || (taxRate = {}));
27 ;
28 var taxNum1 = taxRate.FICA;
29 var taxNum2 = taxRate.STATE;
30 var taxNum3 = taxRate.MEDICAR;
31 console.log("FICA tax: " + taxNum1 + ", STATE taxPayroll2: " + taxNum2 + ", MEDICARE tax:
```

This screenshot shows the Visual Studio Code editor with the file payroll.ts open. The Explorer sidebar on the left shows the project structure with files app.ts, payroll.ts, payroll.js, and app.js. The main editor area displays the payroll.ts code, which defines a taxRate enum and uses it to calculate the FICA, STATE, and MEDICARE taxes. The code uses console.log to output the results of the calculations. The status bar at the bottom indicates the current line and column (Ln 25, Col 17) and the file encoding (UTF-8).

```
17 let tMEDICARE: taxPayroll2 = taxPayroll2.MEDICARE;
18
19 console.log("FICA tax: " + tFICA + ", STATE taxPayroll2: " + tSTATE + ", MEDICARE tax:" +
20
21 enum taxRate{
22   FICA = 10,
23   STATE = 5,
24   MEDICAR = 12
25 };
26
27 let taxNum1: taxRate = taxRate.FICA;
28 let taxNum2: taxRate = taxRate.STATE;
29 let taxNum3: taxRate = taxRate.MEDICAR;
30 console.log("FICA tax: " + taxNum1 + ", STATE taxPayroll2: " + taxNum2 + ", MEDICARE tax:
```

The terminal window at the bottom shows the output of the code execution. It displays the results of the tax calculations for both taxPayroll1 and taxPayroll2, and the results of the taxRate enum. The terminal output is as follows:

```
10 5 12
FICA tax: 0, STATE taxPayroll2: [object Object], MEDICARE tax:2
FICA tax: 10, STATE taxPayroll2: 5, MEDICARE tax:12

C:\Users\Selam\Desktop\CS 445\git-repos2\cs445-map-2021\Lab3a>tsc payroll

C:\Users\Selam\Desktop\CS 445\git-repos2\cs445-map-2021\Lab3a>node payroll
10 5 12
FICA tax: 0, STATE taxPayroll2: 1, MEDICARE tax:2
FICA tax: 10, STATE taxPayroll2: 5, MEDICARE tax:12

C:\Users\Selam\Desktop\CS 445\git-repos2\cs445-map-2021\Lab3a>
```

