

COMP 3133

Lab02 – ScreenShots

1) Terminal Output

```
PS C:\Sem - 6\COMP - 3133\Lab - 2\lab2_mocha_test\lab2_mocha_test> npm test
① PS C:\Sem - 6\COMP - 3133\Lab - 2\lab2_mocha_test\lab2_mocha_test> npm test

> lab2_mocha_test@1.0.0 test
> mocha

Calculator Test Cases
  ADD
ADD PASS
  ✓ PASS: add(5,2) should be 7
  1) FAIL: add(5,2) should be 8
  SUB
SUB PASS
  ✓ PASS: sub(5,2) should be 3
  2) FAIL: sub(5,2) should be 5
  MUL
MUL PASS
  ✓ PASS: mul(5,2) should be 10
  3) FAIL: mul(5,2) should be 12
  DIV
DIV PASS
  ✓ PASS: div(10,2) should be 5
  4) FAIL: div(10,2) should be 2

  4 passing (9ms)
  4 failing
```

```
1) Calculator Test Cases
  ADD
    FAIL: add(5,2) should be 8:

      AssertionError: expected 7 to equal 8
      + expected - actual

      -7
      +8

      at Context.<anonymous> (test\calculator.test.js:17:25)
      at process.processImmediate (node:internal/timers:505:21)

2) Calculator Test Cases
  SUB
    FAIL: sub(5,2) should be 5:

      AssertionError: expected 3 to equal 5
      + expected - actual

      -3
      +5

      at Context.<anonymous> (test\calculator.test.js:31:25)
      at process.processImmediate (node:internal/timers:505:21)
```

```
3) Calculator Test Cases
  MUL
    FAIL: mul(5,2) should be 12:

      AssertionError: expected 10 to equal 12
      + expected - actual

      -10
      +12

      at Context.<anonymous> (test\calculator.test.js:45:25)
      at process.processImmediate (node:internal/timers:505:21)
```

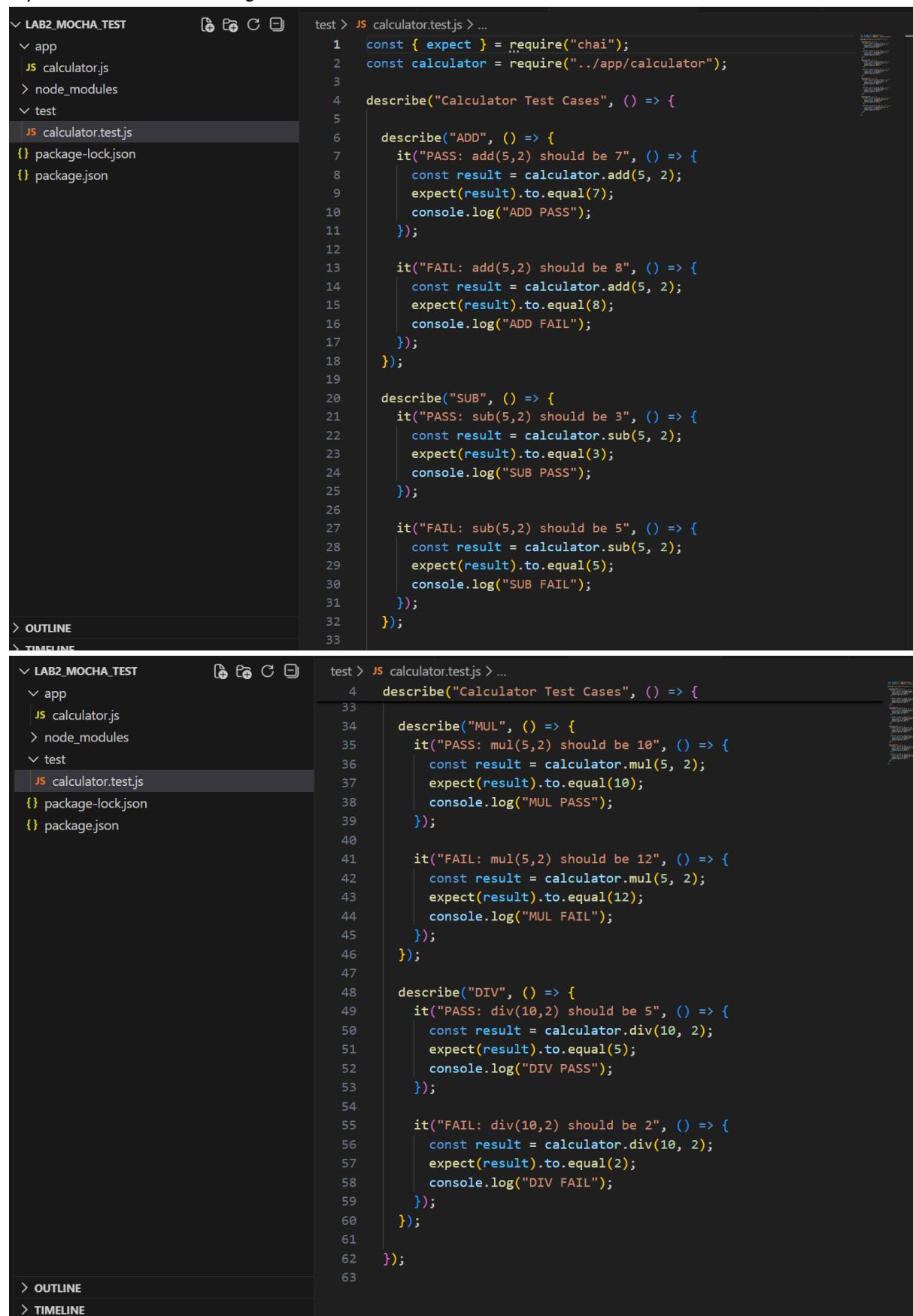
```
4) Calculator Test Cases
  DIV
    FAIL: div(10,2) should be 2:

      AssertionError: expected 5 to equal 2
      + expected - actual

      -5
      +2

      at Context.<anonymous> (test\calculator.test.js:59:25)
      at process.processImmediate (node:internal/timers:505:21)
```

2) calculator.test.js



```
test > JS calculator.test.js > ...
1  const { expect } = require("chai");
2  const calculator = require("../app/calculator");
3
4  describe("Calculator Test Cases", () => {
5
6    describe("ADD", () => {
7      it("PASS: add(5,2) should be 7", () => {
8        const result = calculator.add(5, 2);
9        expect(result).to.equal(7);
10       console.log("ADD PASS");
11     });
12
13     it("FAIL: add(5,2) should be 8", () => {
14       const result = calculator.add(5, 2);
15       expect(result).to.equal(8);
16       console.log("ADD FAIL");
17     });
18   });
19
20   describe("SUB", () => {
21     it("PASS: sub(5,2) should be 3", () => {
22       const result = calculator.sub(5, 2);
23       expect(result).to.equal(3);
24       console.log("SUB PASS");
25     });
26
27     it("FAIL: sub(5,2) should be 5", () => {
28       const result = calculator.sub(5, 2);
29       expect(result).to.equal(5);
30       console.log("SUB FAIL");
31     });
32   });
33 }

> OUTLINE
> TIMELINE
```



```
test > JS calculator.test.js > ...
4  describe("Calculator Test Cases", () => {
33
34    describe("MUL", () => {
35      it("PASS: mul(5,2) should be 10", () => {
36        const result = calculator.mul(5, 2);
37        expect(result).to.equal(10);
38        console.log("MUL PASS");
39      });
40
41      it("FAIL: mul(5,2) should be 12", () => {
42        const result = calculator.mul(5, 2);
43        expect(result).to.equal(12);
44        console.log("MUL FAIL");
45      });
46    });
47
48    describe("DIV", () => {
49      it("PASS: div(10,2) should be 5", () => {
50        const result = calculator.div(10, 2);
51        expect(result).to.equal(5);
52        console.log("DIV PASS");
53      });
54
55      it("FAIL: div(10,2) should be 2", () => {
56        const result = calculator.div(10, 2);
57        expect(result).to.equal(2);
58        console.log("DIV FAIL");
59      });
60    });
61  });
62 });

> OUTLINE
> TIMELINE
```

3) calculator.js

The screenshot shows a code editor interface with two panes. The left pane displays a file tree for a project named 'LAB2_MOCHA_TEST'. The tree includes a 'app' folder containing 'calculator.js', a 'node_modules' folder, a 'test' folder with 'calculator.test.js', and 'package-lock.json' and 'package.json' files. The right pane shows the content of 'calculator.js' with syntax highlighting for JavaScript code. The code defines four export functions: add, mul, div, and sub.

```
app > JS calculator.js > ...
1  exports.add = function(i, j) {
2    ...
3    return i + j;
4  };
5  exports.mul = function(i, j) {
6    ...
7    return i * j;
8  };
9  exports.div = function(i, j) {
10   ...
11  return i / j;
12 };
13 exports.sub = function(i, j) {
14   ...
15  return i - j;
};
```