Unit Testing the eCommerce Applications-1 Assignment Solution

Create a promise and callback functions and also their test function and test it.

Create a new file named async.test.js and write your promise and test function:

```
async.test.js U ✕

async.test.js > ⊗ test("Testing Callback Function") callback > ⊗ callback
1 ∨ /**
2     * Testing Promise
3     */
4
5 ∨ function promiseTestFunc(){
6 ∨     return new Promise((resolve, reject) =>{
7             resolve("Promise Tested");
8         });
9     }
10
11 ∨ test("Testing Promise", ()=>{
12 ∨     return promiseTestFunc().then((msg)=>{
13             expect(msg).toBe("Promise Tested");
14         })
15     });
16
17
```

Similarly write a callback and its test function:

```
18    /**
19     *
20     * Testing a callback function
21     */
22    test("Testing Callback Function", done=>{
23        function callback(data) {
24            try{
25                expect(data).toBe("Called");
26                done();
27            }catch(error){
28                done(error);
29            }
30        }
31
32        fetchData(callback);
33    })
34
```

```
34
35
36    function fetchData(callback){
37        setTimeout(()=>{
38            callback("Called")
39        },2000);
40    }
```

Test Result:

```
PS D:\Mohit\Relevel\Backend\testing-using-jest> npm test async.test.js

> testing-using-jest@1.0.0 test
> jest "async.test.js"

 PASS  ./async.test.js
  √ Testing Promise (3 ms)
  √ Testing Callback Function (2007 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        2.865 s, estimated 3 s
Ran all test suites matching /async.test.js/i.
PS D:\Mohit\Relevel\Backend\testing-using-jest> []
```

Create another file with a function in it and also, it's test file and run the test.

Create a file named calculator.js with some function as show below:

```
README.md U        JS calculator.js U  X

JS calculator.js > ...
  1    function add(...num){
  2        return num.reduce((acc, summ) => acc + summ, 0);
  3    }
  4
  5    function multiply(...num){
  6        return num.reduce((acc, summ) => acc * summ, 1);
  7    }
  8
  9    function substract(minuend, subtrahend ){
 10        return minuend - subtrahend;
 11    }
 12
 13    function divide(dividend, divisor ){
 14        return dividend / divisor;
 15    }
 16
 17    module.exports = { add, multiply, substract, divide };
```

Now create a test file corresponding to it:

```
calculator.test.js U ✕
calculator.test.js > ...
1    const { add, multiply, substract, divide }  = require('./calculator');
2
3
4    test('testing the add method',()=>{
5        expect(add(1,2,3,4)).toBe(10);
6    });
7
8    test('testing the substract method',()=>{
9        expect(substract(6, 8)).toBe(-2);
10   });
11
12   test('testing the multiply method',()=>{
13       expect(multiply(6, 8, 0)).toBe(0);
14   });
15
16   test('testing the divide method',()=>{
17       expect(divide(10, 2)).toBe(5);
18   });
```

Test Result:

```
PS D:\Mohit\Relevel\Backend\testing-using-jest> npm test calculator.test.js

> testing-using-jest@1.0.0 test
> jest "calculator.test.js"

 PASS  ./calculator.test.js
   √ testing the add method (3 ms)
   √ testing the substract method (1 ms)
   √ testing the multiply method
   √ testing the divide method

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        1.047 s
Ran all test suites matching /calculator.test.js/i.
PS D:\Mohit\Relevel\Backend\testing-using-jest>
```

Please find the github link for reference.