Lesson 11 Demo 01

Implementing Hello World Unit Test with Mocha

Objective: To demonstrate unit testing in an Express.js application using Mocha to ensure robust code functionality and identify potential errors

Tools Required: Visual Studio

Prerequisites: Knowledge of JavaScript and Node.js

Steps to be followed:

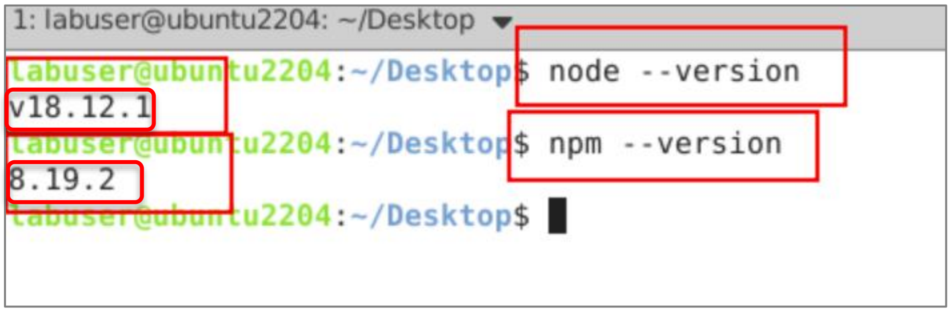
1. Check if Node.js is installed
2. Create a project structure, a package.json file, and install the Mocha module
3. Create a simple Node.js program and test file

Step 1: Check if Node.js is installed

1.1 Check if Node.js is installed successfully and run the following commands:

node --version
npm --version

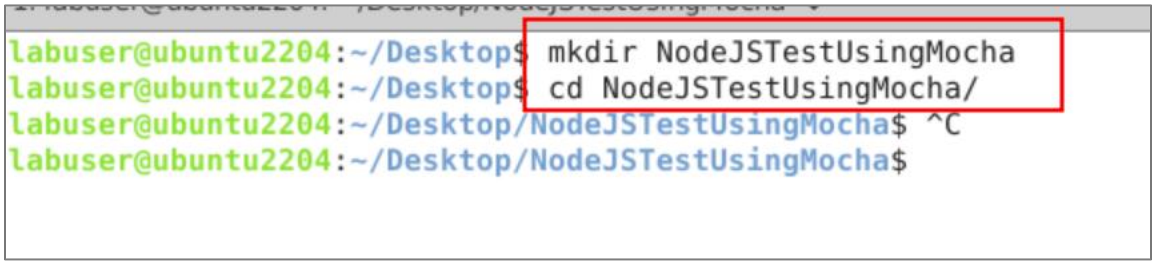
If the Node.js installation is successful, the following output will appear:



```
1: labuser@ubuntu2204: ~/Desktop
labuser@ubuntu2204:~/Desktop$ node --version
v18.12.1
labuser@ubuntu2204:~/Desktop$ npm --version
8.19.2
labuser@ubuntu2204:~/Desktop$ 
```

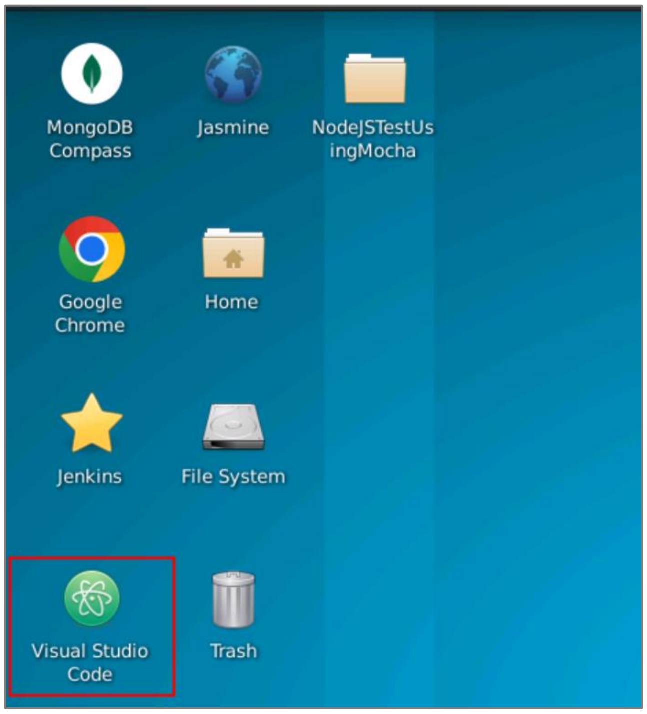
Step 2: Create a project structure, a package.json file, and install the Mocha module

2.1 Run the following code to create a directory to hold the application and make it the working directory:
mkdir NodeJSTestUsingMocha
cd NodeJSTestUsingMocha

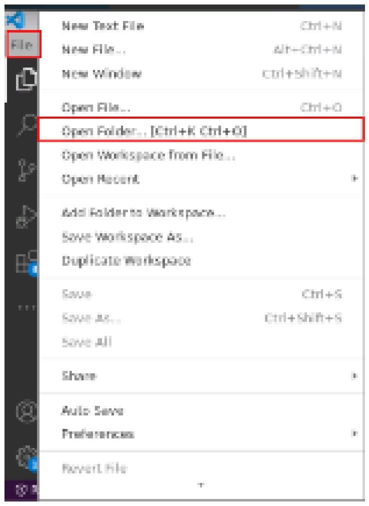


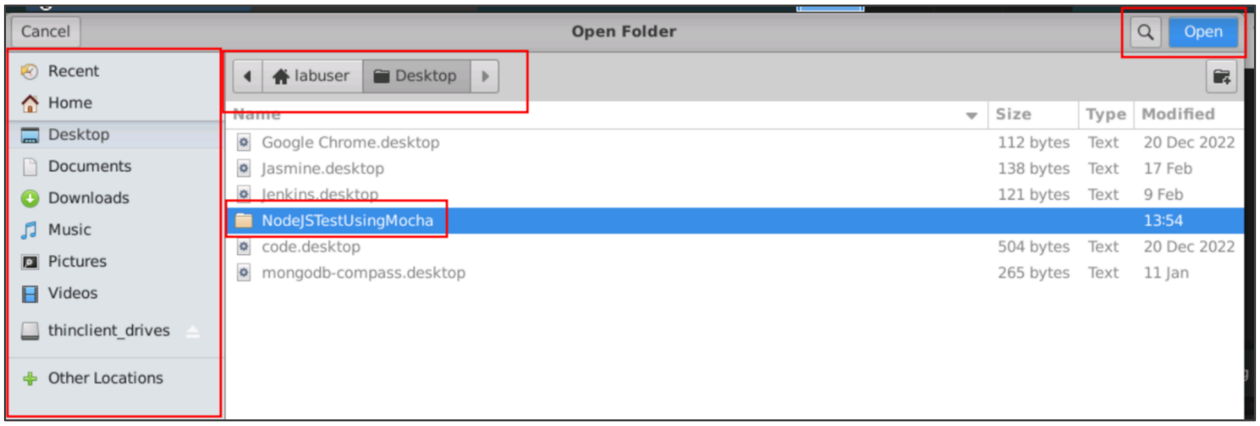
```
labuser@ubuntu2204:~/Desktop$ mkdir NodeJSTestUsingMocha
labuser@ubuntu2204:~/Desktop$ cd NodeJSTestUsingMocha/
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ ^C
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$
```

2.2 Open this folder in the VS Code IDE



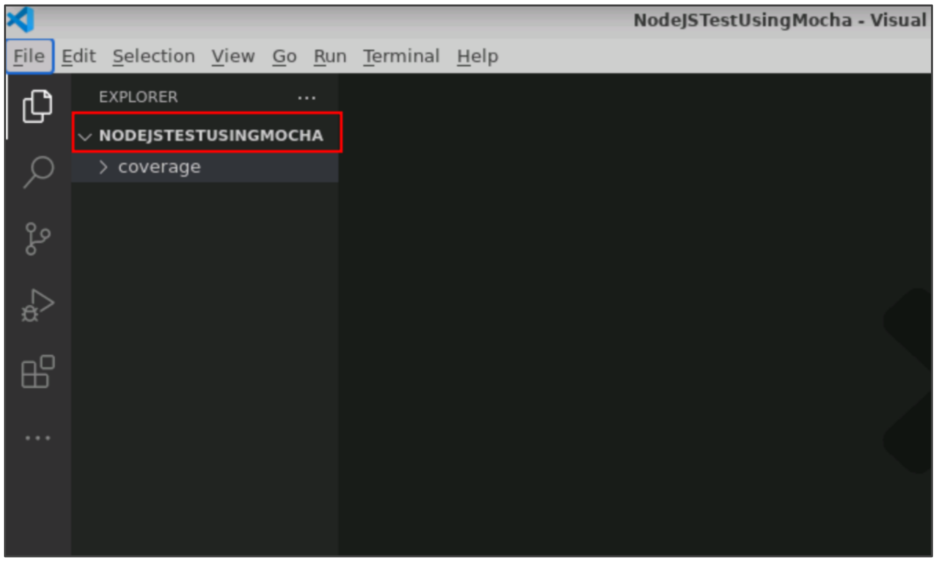
2.3 In VS Code, click **File** and select the **Open Folder** option



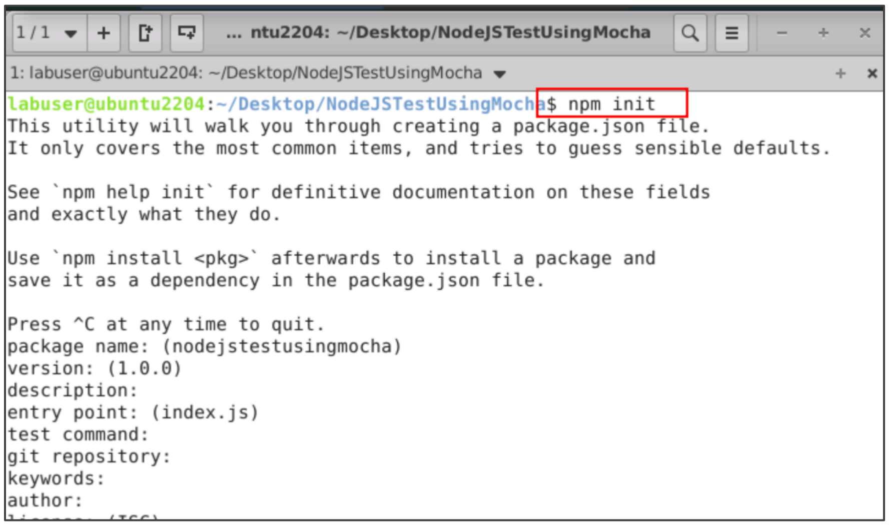


Name	Size	Type	Modified
Google Chrome.desktop	112 bytes	Text	20 Dec 2022
Jasmine.desktop	138 bytes	Text	17 Feb
Jenkins.desktop	121 bytes	Text	9 Feb
NodeJSTestUsingMocha			13:54
code.desktop	504 bytes	Text	20 Dec 2022
mongodb-compass.desktop	265 bytes	Text	11 Jan

2.4 View the empty project in VS Code IDE

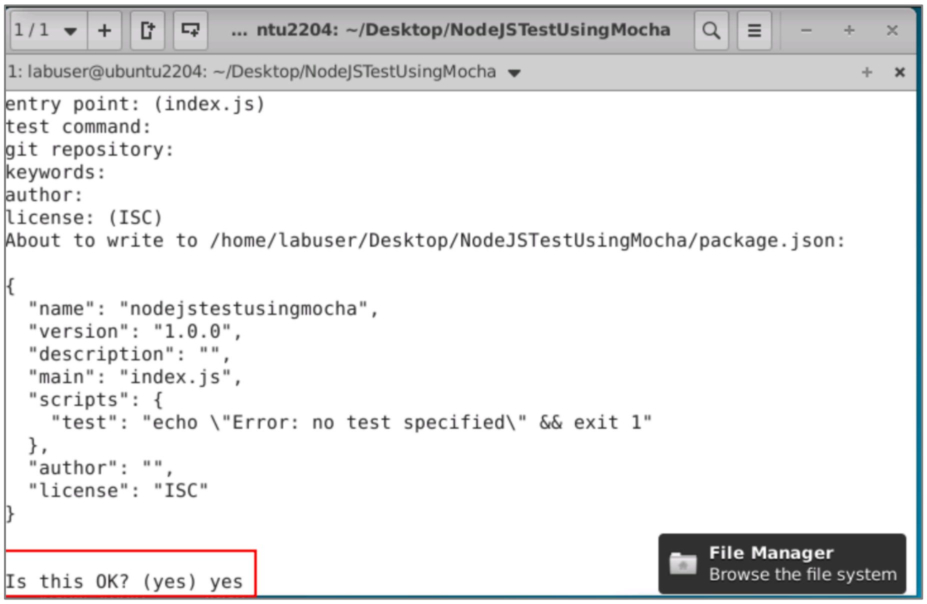


2.5 Open the terminal and run the following command to create a **package.json** file for the application: **npm init**



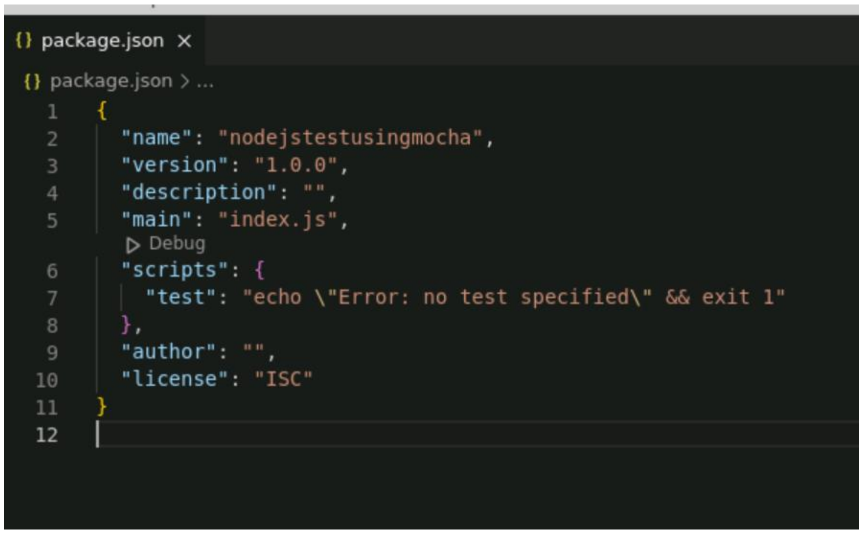
```
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejstestusingmocha)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
```

Note: Press the **enter** key for all the questions and keep the default values for now. Type **yes** and press the **enter** key.



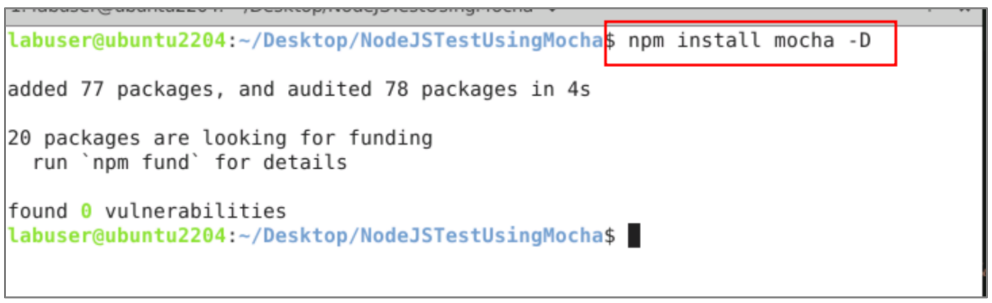
```
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/labuser/Desktop/NodeJSTestUsingMocha/package.json:

{
  "name": "nodejstestusingmocha",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
```

You will be able to view the **package.json** file.



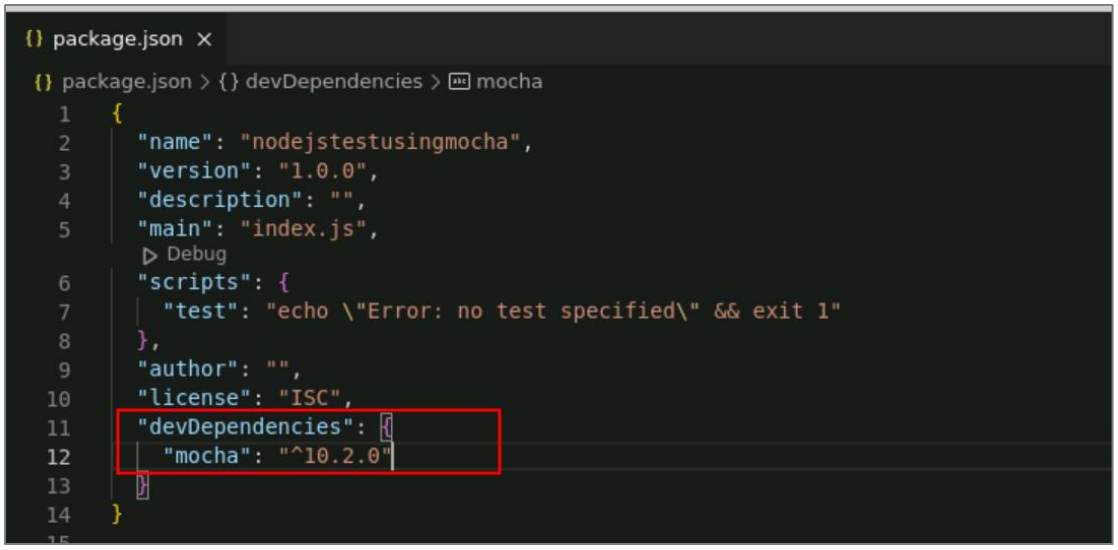
```
{} package.json ×
{} package.json > ...
{
  "name": "nodejstestusingmocha",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

2.6 Install dependencies using the **npm install mocha -D** command
(-D means testing dependencies required only in development mode.)



```
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ npm install mocha -D

added 77 packages, and audited 78 packages in 4s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$
```

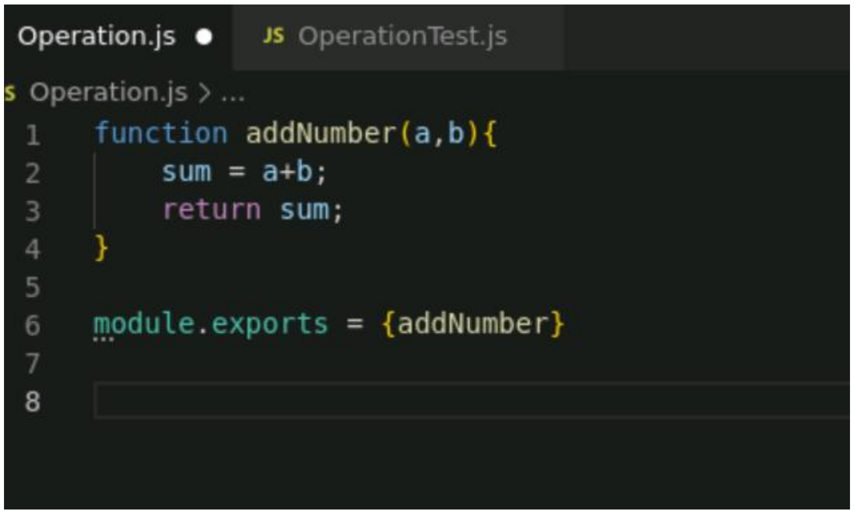
You can view these dependencies in the package.json file.



```
{} package.json ×
{} package.json > {} devDependencies > mocha
{
  "name": "nodejstestusingmocha",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "mocha": "^10.2.0"
  }
}
```

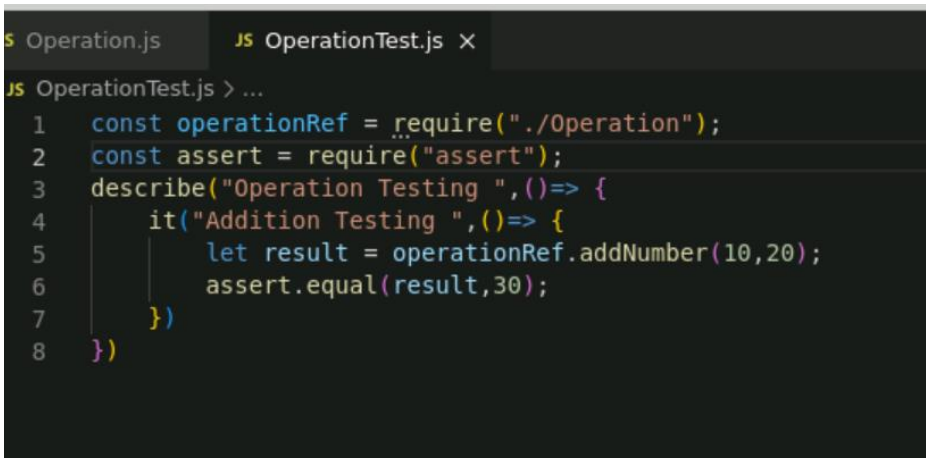
Step 3: Create a simple Node.js program and test file

3.1 Create a simple Operation.js file with the following code:



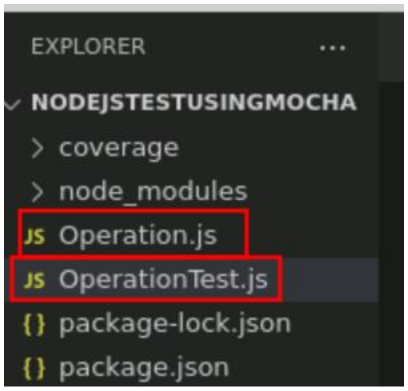
```
function addNumber(a,b){
    sum = a+b;
    return sum;
}

module.exports = {addNumber}

```

3.2 Create a simple OperationTest.js file with the following code:

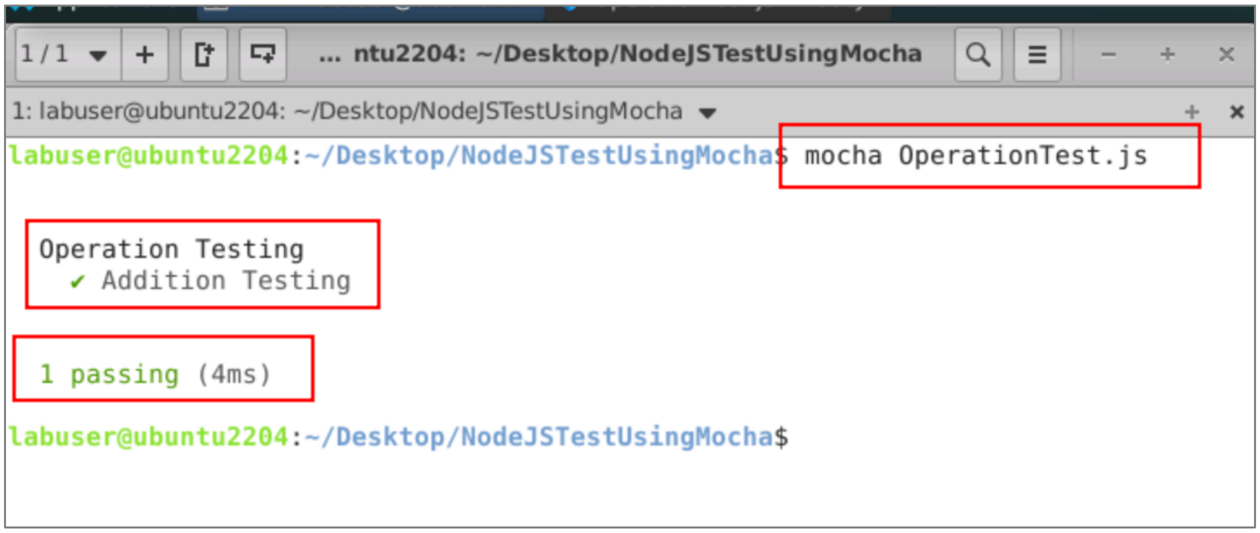


```
const operationRef = require("./Operation");
const assert = require("assert");
describe("Operation Testing ",()=> {
    it("Addition Testing ",()=> {
        let result = operationRef.addNumber(10,20);
        assert.equal(result,30);
    })
})
```

3.3 View the project structure after both files are created

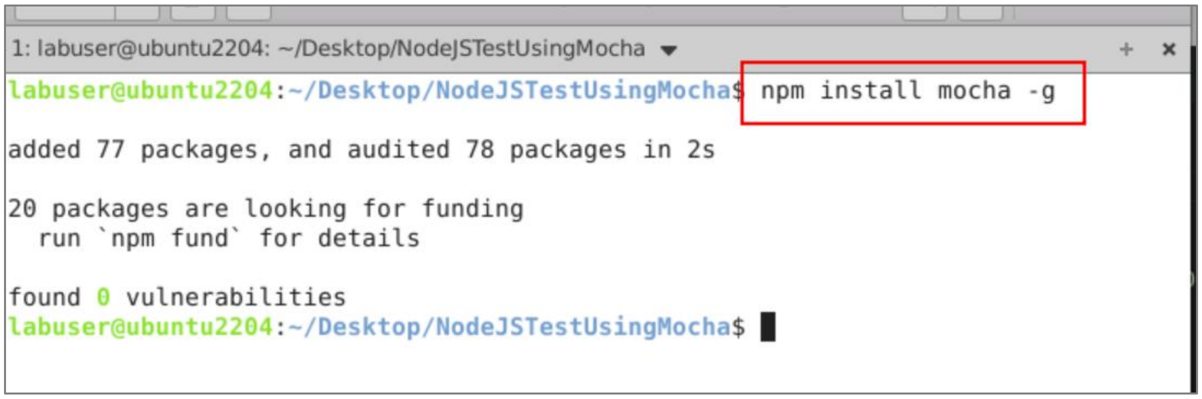


3.4 Run the following command in the **terminal** to run the file:


1 / 1
... ntu2204: ~/Desktop/NodeJSTestUsingMocha
1: labuser@ubuntu2204: ~/Desktop/NodeJSTestUsingMocha

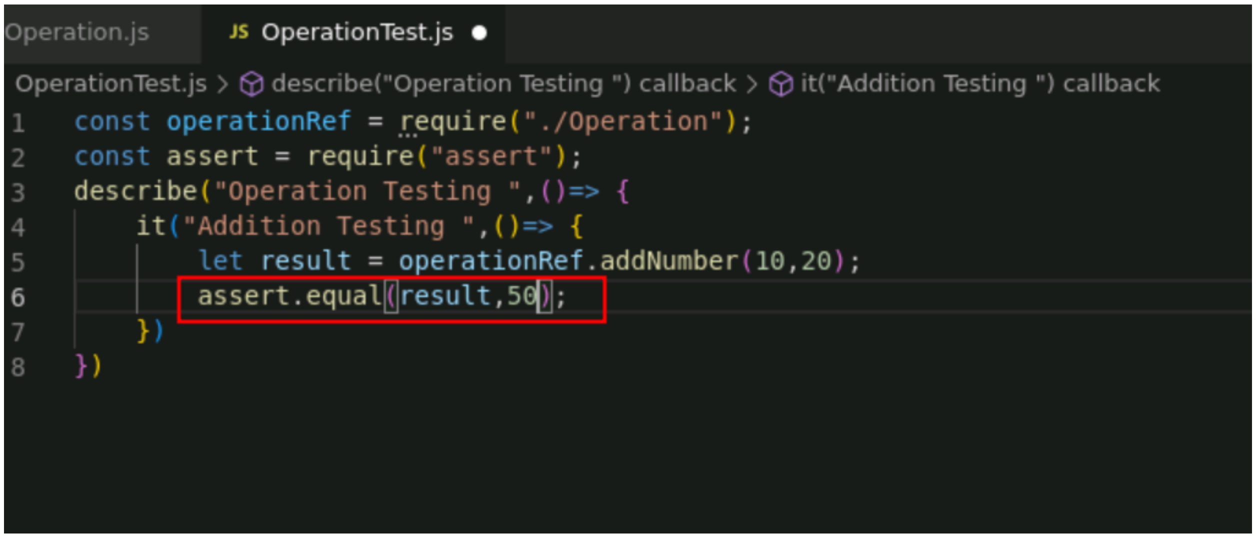
```
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ mocha OperationTest.js


  Operation Testing
    ✔ Addition Testing


  1 passing (4ms)

labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$
```

3.5 If you get any error like Mocha command not found, execute the following command to enable Mocha:
npm install mocha -g



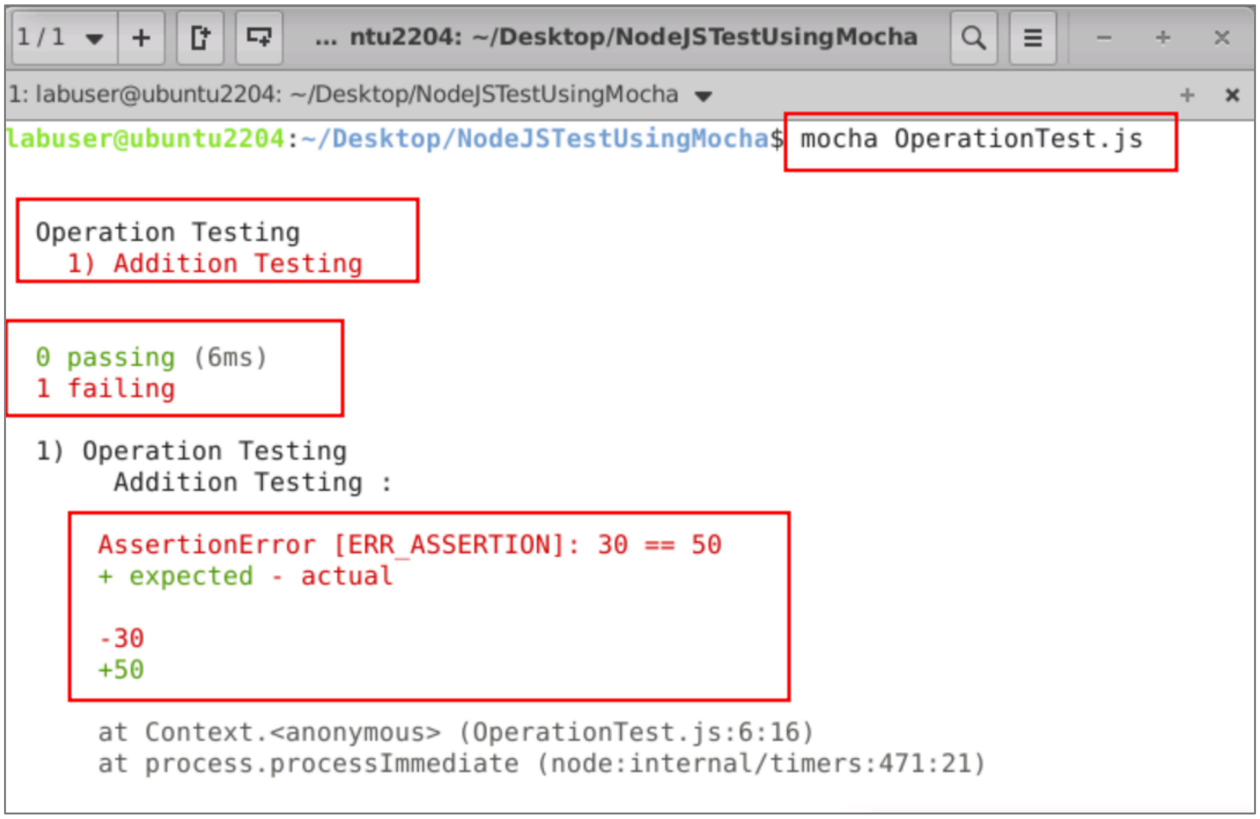
```
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ npm install mocha -g

added 77 packages, and audited 78 packages in 2s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ █
```

Note: The test will pass because the actual and expected results match. If the actual and expected values do not match, then you will get an error. So, make the necessary changes in the OperationTest.js file.


Operation.js
OperationTest.js
OperationTest.js > describe("Operation Testing ") callback > it("Addition Testing ") callback

```javascript
const operationRef = require("./Operation");
const assert = require("assert");
describe("Operation Testing ",()=> {
    it("Addition Testing ",()=> {
        let result = operationRef.addNumber(10,20);
        assert.equal(result,50);
    })
})
```

Run the test again. Here, you may expect an error.


1/1
... ntu2204: ~/Desktop/NodeJSTestUsingMocha
1: labuser@ubuntu2204: ~/Desktop/NodeJSTestUsingMocha

```
labuser@ubuntu2204:~/Desktop/NodeJSTestUsingMocha$ mocha OperationTest.js


  Operation Testing
    1) Addition Testing


  0 passing (6ms)
  1 failing

  1) Operation Testing
       Addition Testing :

      AssertionError [ERR_ASSERTION]: 30 == 50
      + expected - actual

      -30
      +50

      at Context.<anonymous> (OperationTest.js:6:16)
      at process.processImmediate (node:internal/timers:471:21)
```

By following these steps, you have successfully implemented a Hello World unit test in an Express.js application using Mocha to ensure the reliability of the code.