**Cypress**

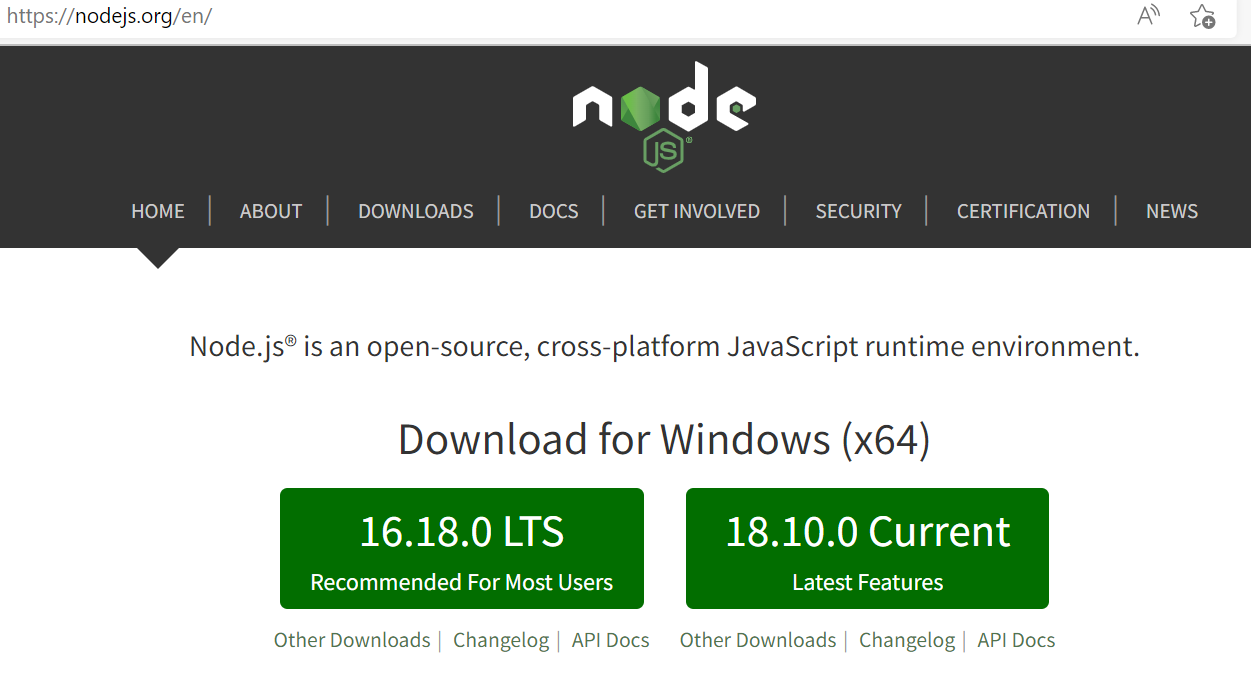
**Prerequisites:**

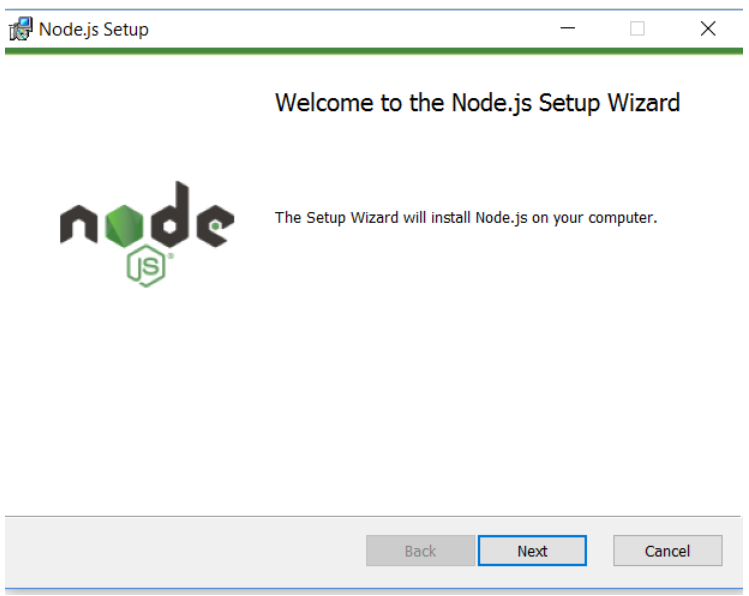
Need to install latest Nodejs.

Use the below link to download NodeJS : <https://nodejs.org/en/>

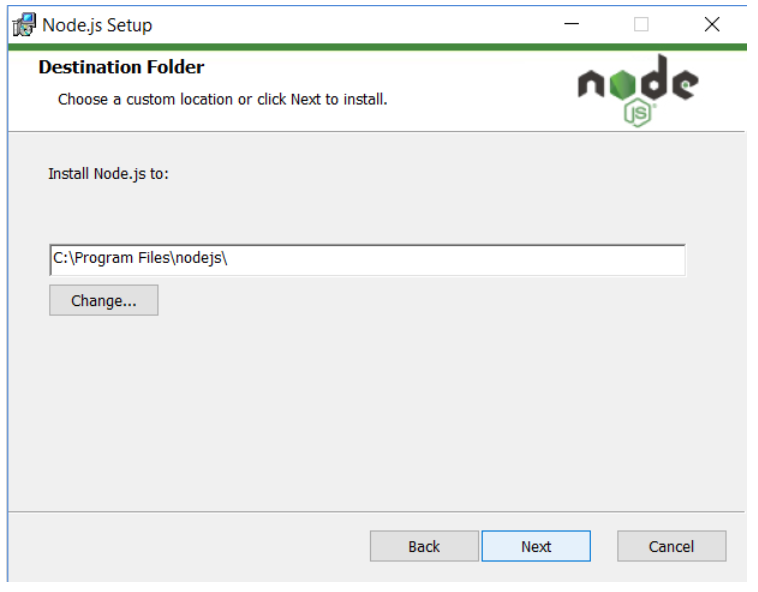
After downloading, install the it in your system .

Step :1 -> Now .msi file will be downloaded to your browser. Choose the desired location for that

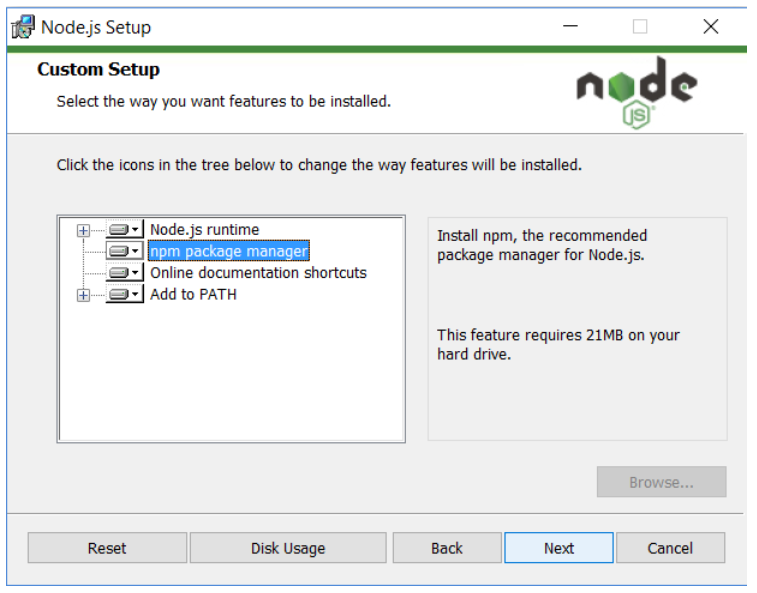




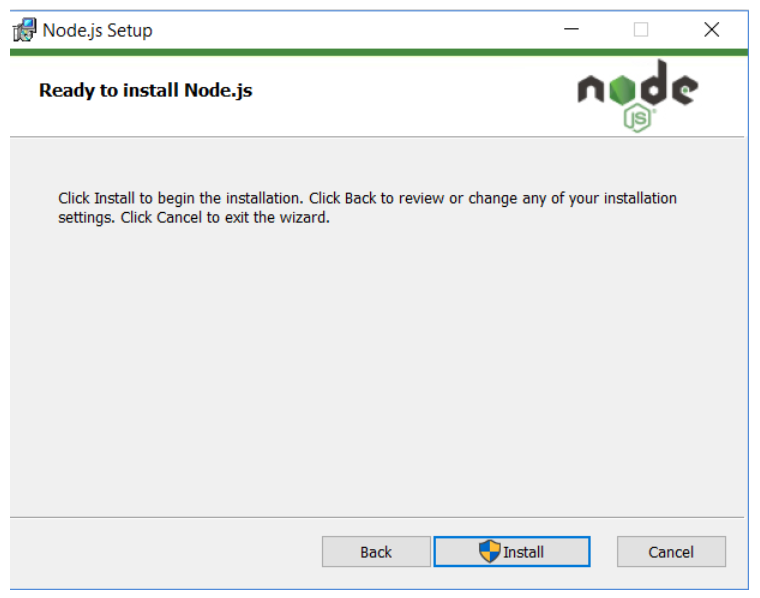
Step :2 -> Click **Next**



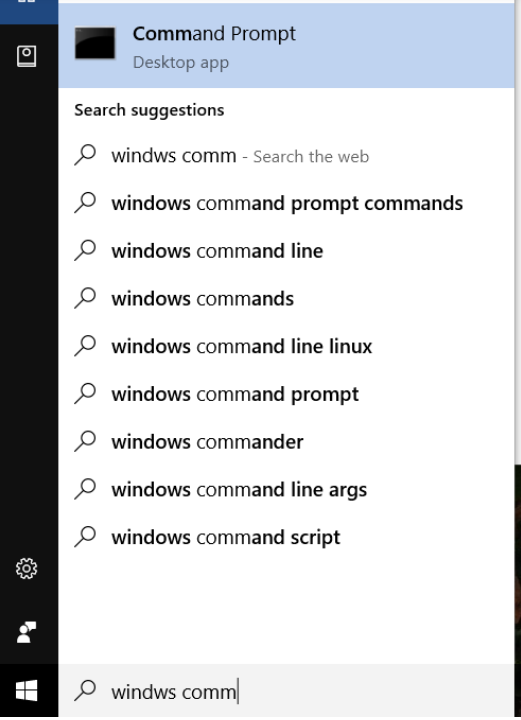
Step :3 -> Click **Next**



Step : 4 -> Click **Next**

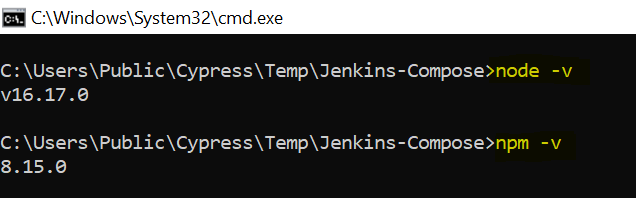


Step :5 -> Click Install



Step :6 -> Go to command prompt

1. To confirm Node installation, type node -v command
2. To confirm NPM installation, type npm -v command



Visual Studio code editor.

To download Visual studio code editor use the URL : <https://code.visualstudio.com/download>

* To setup project in visual studio use below command line in the terminal :

**npm install cypress**

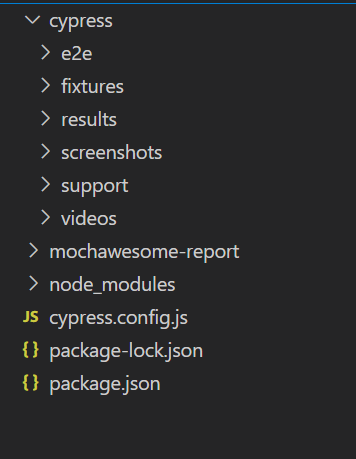
* To open cypress Test Runner in visual studio :

**npx cypress open** or **node\_modules/.bin/cypress open**

* To run project in VS use :

**npx cypress run**

* Project folder structure



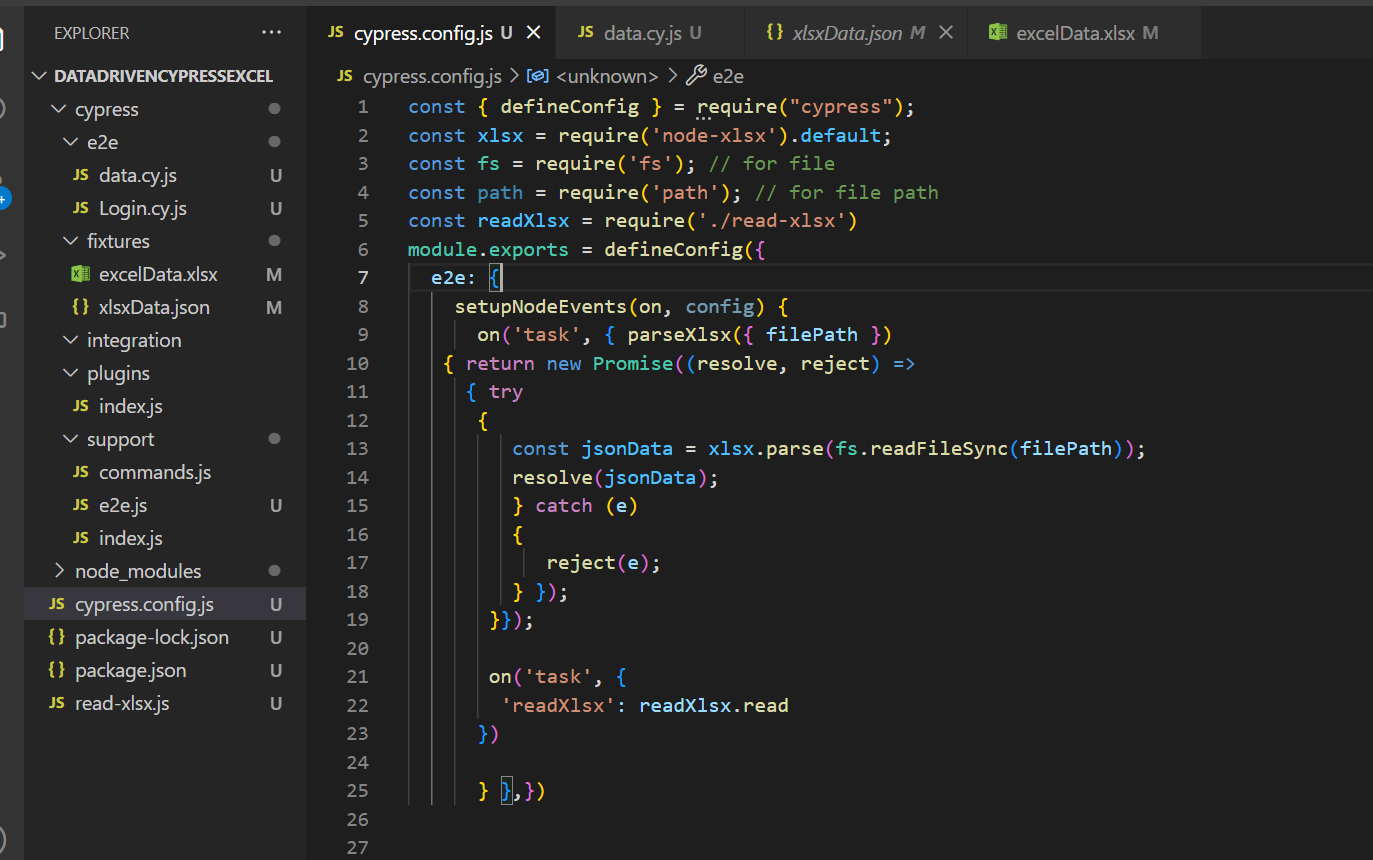
**Data Driven**

Prerequisites

Run below commands in VS terminal window

1. npm install node-xlsx --save-dev
2. store excel sheet in fixture folder

Please make below change in **cypress.config.js**



**Codes:**

**const { defineConfig } = require("cypress");**

**const xlsx = require('node-xlsx').default;**

**const fs = require('fs'); // for file**

**const path = require('path'); // for file path**

**const readXlsx = require('./read-xlsx')**

**module.exports = defineConfig({**

**e2e: {**

**setupNodeEvents(on, config) {**

**on('task', { parseXlsx({ filePath })**

**{ return new Promise((resolve, reject) =>**

**{ try**

**{**

**const jsonData = xlsx.parse(fs.readFileSync(filePath));**

**resolve(jsonData);**

**} catch (e)**

**{**

**reject(e);**

**} });**

**}});**

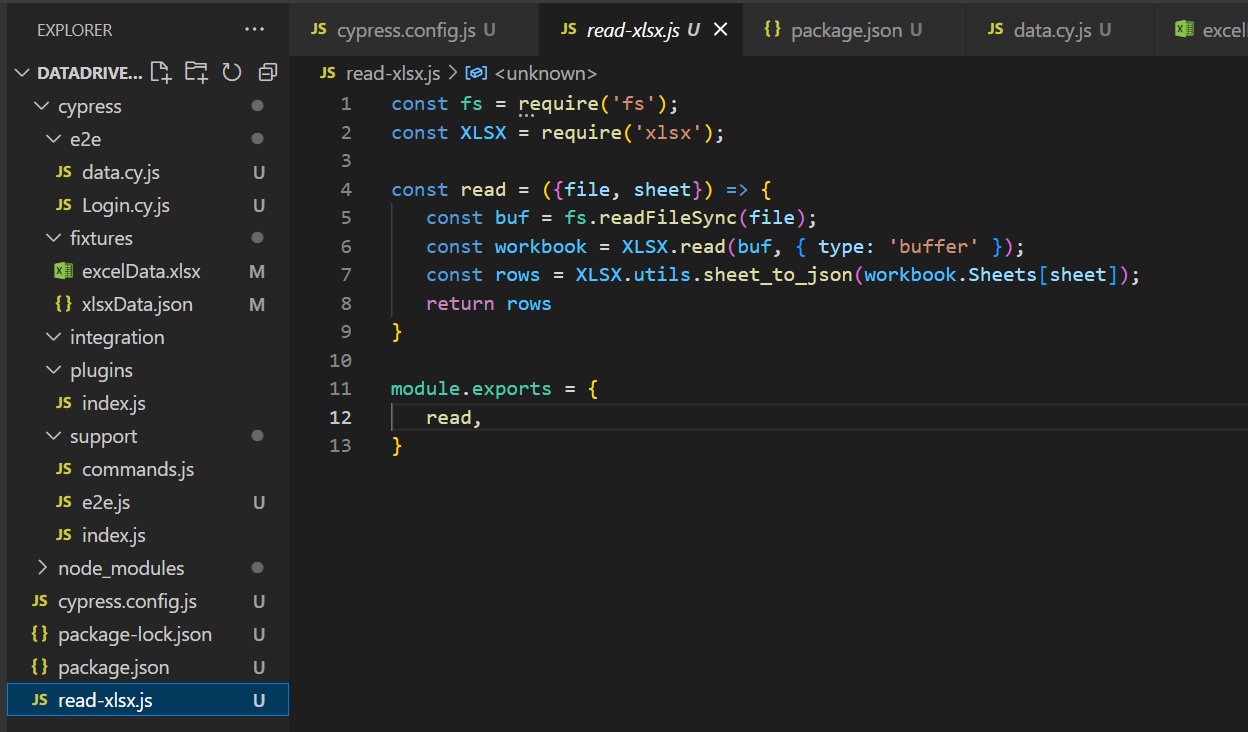
**on('task', {**

**'readXlsx': readXlsx.read**

**})**

**} },})**

**Create file with below codes.**



**Code:**

**const fs = require('fs');**

**const XLSX = require('xlsx');**

**const read = ({file, sheet}) => {**

**const buf = fs.readFileSync(file);**

**const workbook = XLSX.read(buf, { type: 'buffer' });**

**const rows = XLSX.utils.sheet\_to\_json(workbook.Sheets[sheet]);**

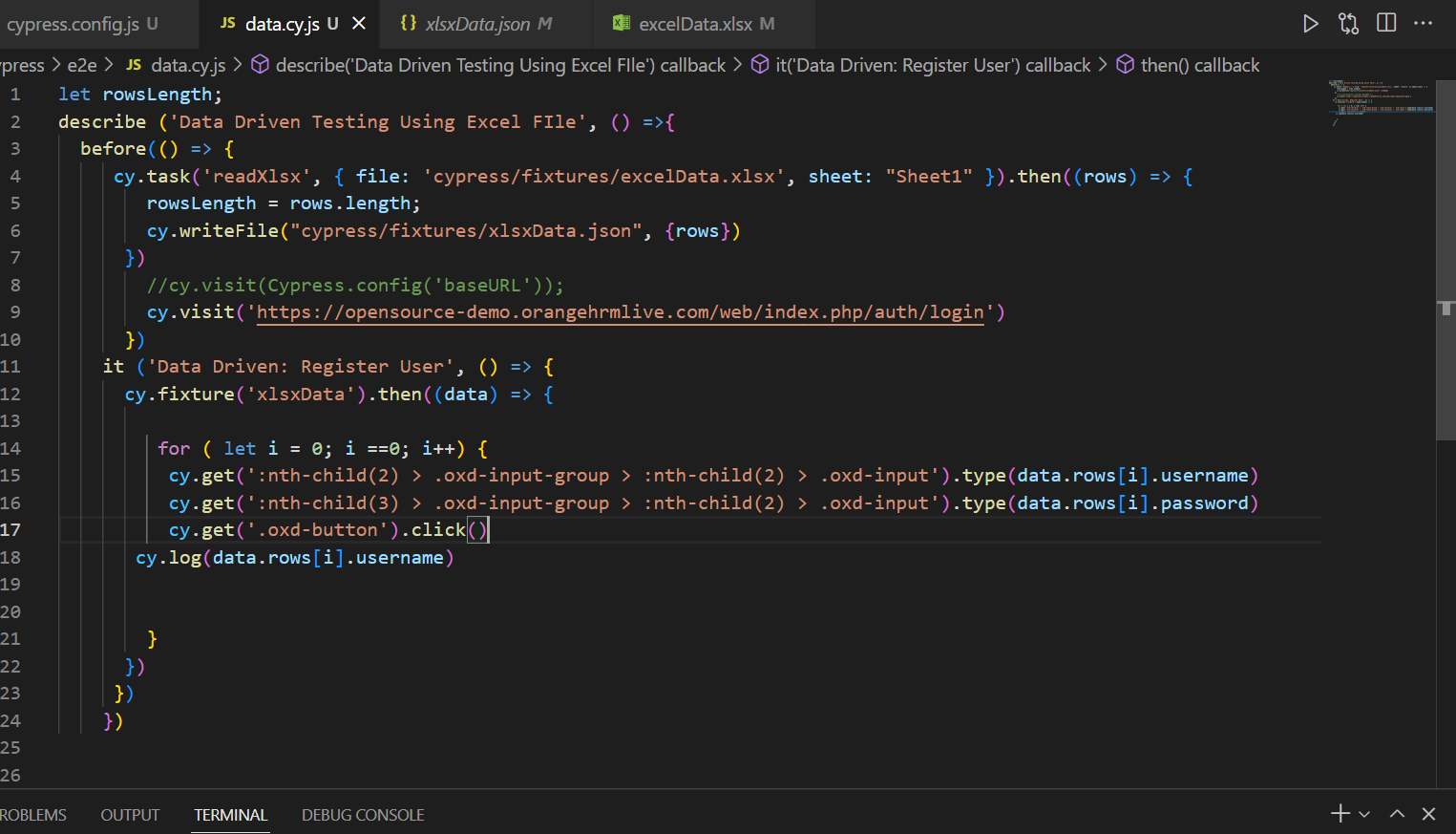
**return rows**

**}module.exports = {**

**read,**

**}**

**Create file with below codes.**



**Code:**

**let rowsLength;**

**describe ('Data Driven Testing Using Excel FIle', () =>{**

**before(() => {**

**cy.task('readXlsx', { file: 'cypress/fixtures/excelData.xlsx', sheet: "Sheet1" }).then((rows) => {**

**rowsLength = rows.length;**

**cy.writeFile("cypress/fixtures/xlsxData.json", {rows})**

**})**

**})**

**it ('Data Driven: Register User', () => {**

**cy.fixture('xlsxData').then((data) => {**

**for ( let i = 0; i ==0; i++) {**

**cy.get(':nth-child(2) > .oxd-input-group > :nth-child(2) > .oxd-input').type(data.rows[i].username)**

**cy.get(':nth-child(3) > .oxd-input-group > :nth-child(2) > .oxd-input').type(data.rows[i].password)**

**cy.get('.oxd-button').click()**

**cy.log(data.rows[i].username)**

**}**

**})**

**})**

**})**

**Report Generating**

Mochawesome

**Prerequisites**

Run below commands in VS terminal window

1. **npm install --save-dev mochawesome**
2. **npm install mochawesome-merge --save-dev**
3. **npm install mochawesome-report-generator**

Please make below change in **cypress.config.js**



Code:

reporter: 'mochawesome',  
 reporterOptions: {  
 reportDir: 'mochawesome-report',  
 overwrite: false,  
 html: false,  
 json: true,},

Run below commands in VS terminal window

1. npx cypress run --reporter mochawesome

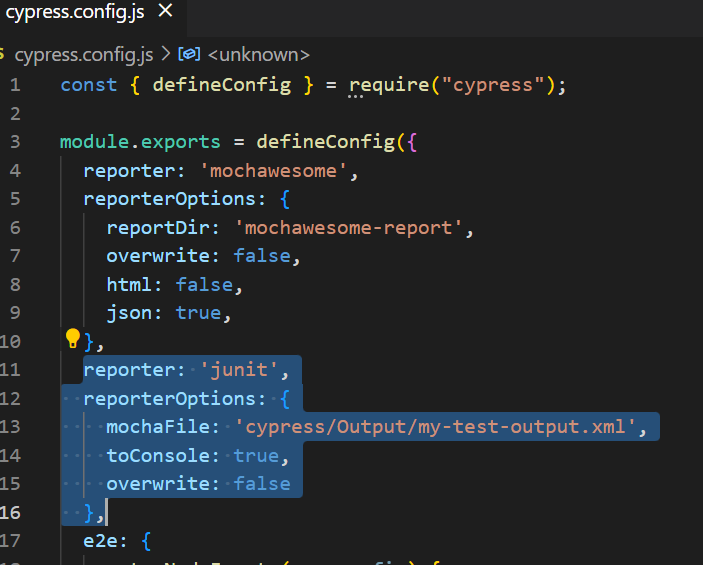
**Junit Report**

Prerequisites

Run below commands in VS terminal window

1. **npm install --save-dev cypress-multi-reporters mocha-junit-reporter**

Please make below change in **cypress.config.js**



Code:

reporter: 'junit',  
 reporterOptions: {  
 mochaFile: 'cypress/Output/my-test-output.xml',  
 toConsole: true,  
 overwrite: false  
 },

Run below commands in VS terminal window

1. npx cypress run --reporter junit

**Cucumber Reporting**

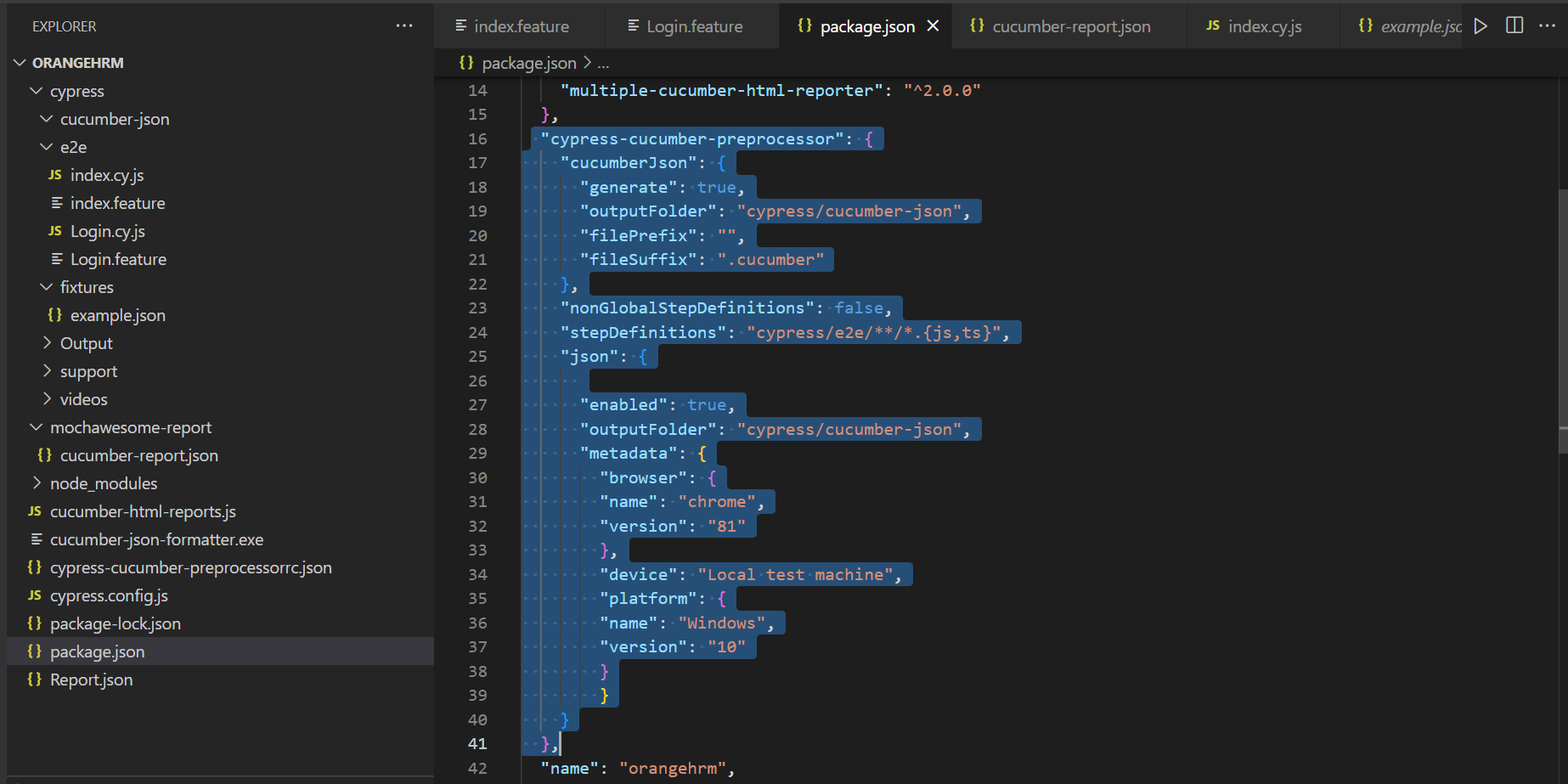
Prerequisites

**cucumber-json-formatter-windows-386** and rename as

**cucumber-json-formatter.exe** and put it in the project folder

Run below commands in VS terminal window

1. **npm install -D @badeball/cypress-cucumber-preprocessor**
2. **npm install -D @bahmutov/cypress-esbuild-preprocessor**
3. **npm install -D @esbuild-plugins/node-modules-polyfill**



Code:

"cypress-cucumber-preprocessor": {

"cucumberJson": {

"generate": true,

"outputFolder": "cypress/cucumber-json",

"filePrefix": "",

"fileSuffix": ".cucumber"

},

"nonGlobalStepDefinitions": false,

"stepDefinitions": "cypress/e2e/\*\*/\*.{js,ts}",

"json": {

"enabled": true,

"outputFolder": "cypress/cucumber-json",

"metadata": {

"browser": {

"name": "chrome",

"version": "81"

},

"device": "Local test machine",

"platform": {

"name": "Windows",

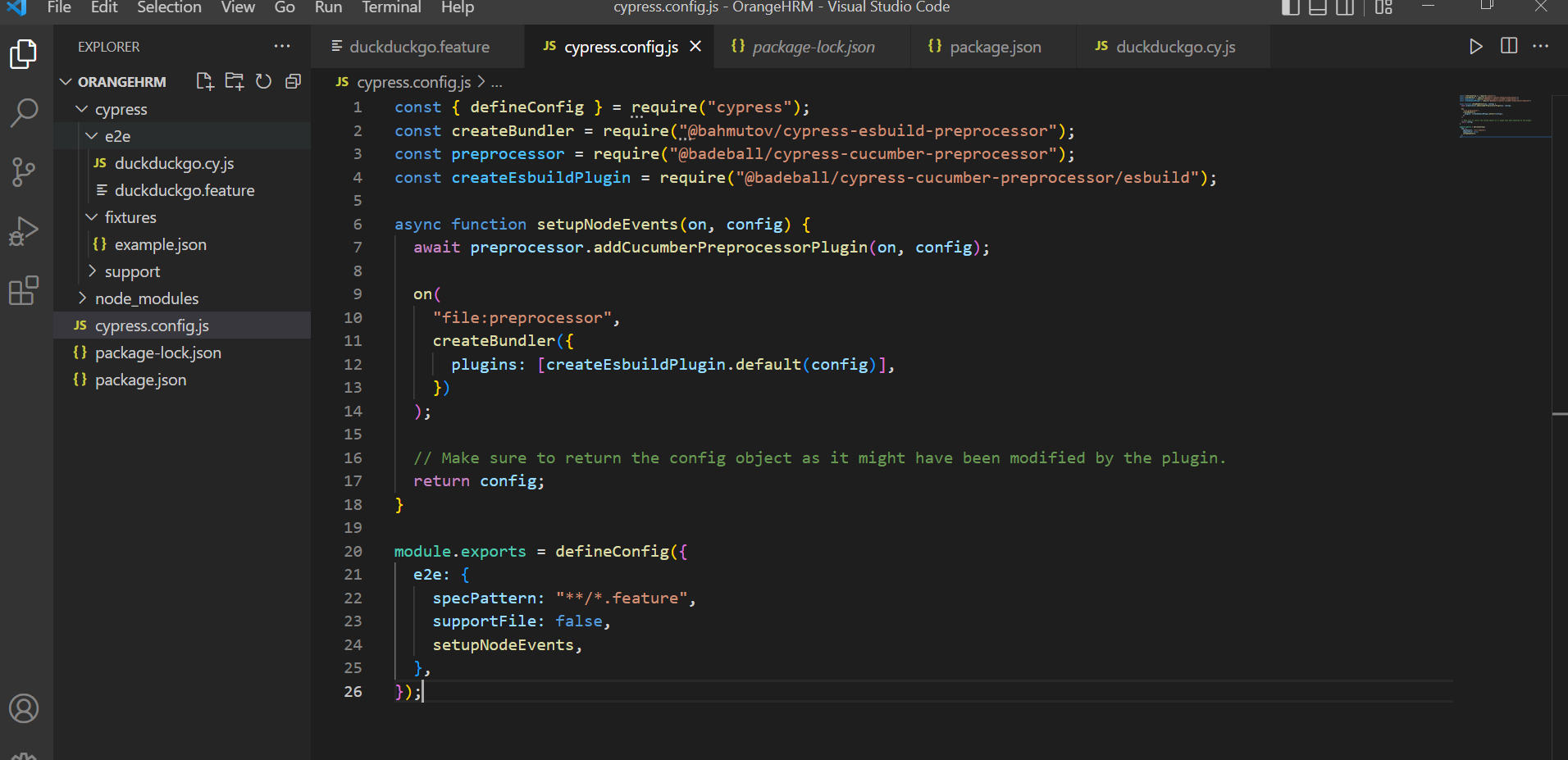
"version": "10"

}

}

}

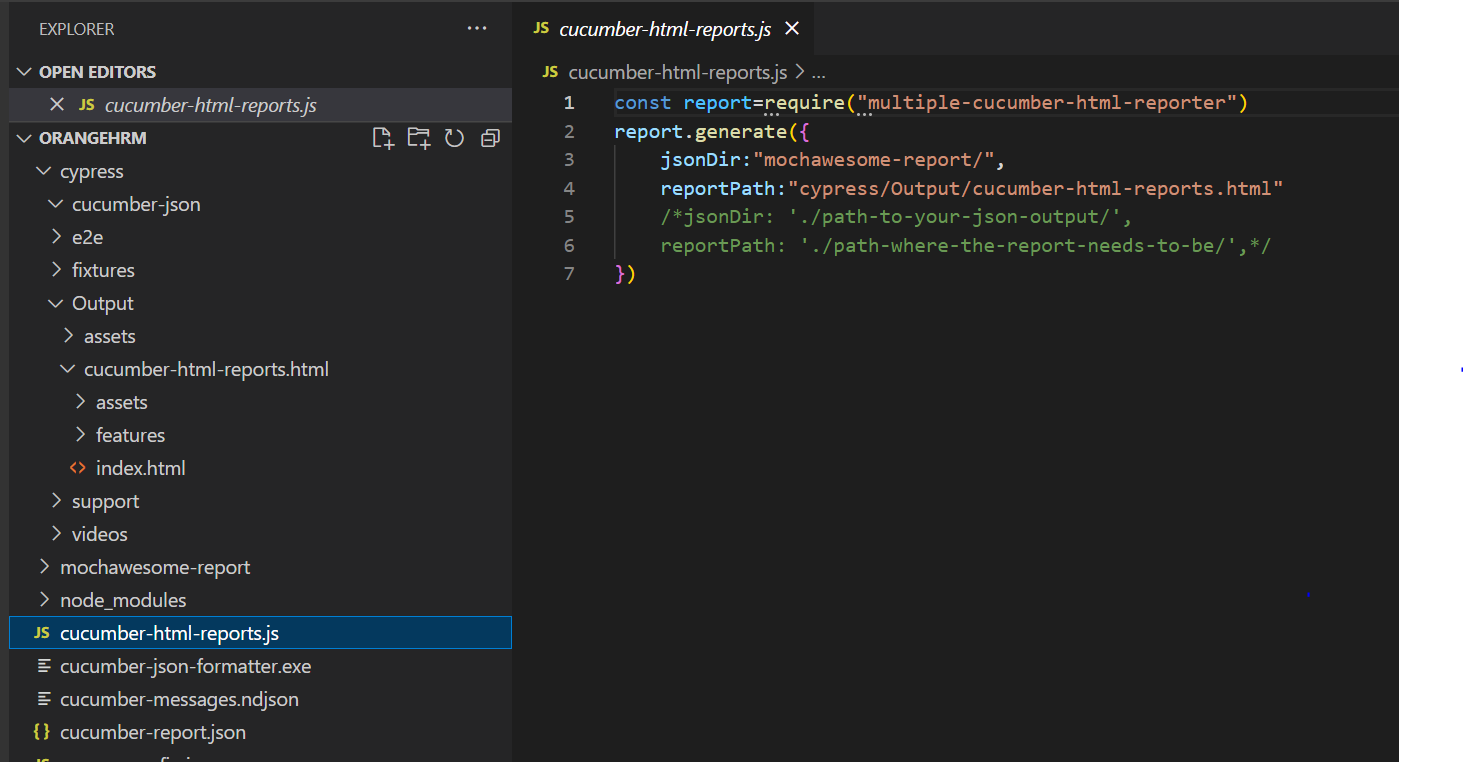
},



Code:

const { defineConfig } = require("cypress");  
const createBundler = require("@bahmutov/cypress-esbuild-preprocessor");  
const preprocessor = require("@badeball/cypress-cucumber-preprocessor");  
const createEsbuildPlugin = require("@badeball/cypress-cucumber-preprocessor/esbuild");  
  
async function setupNodeEvents(on, config) {  
 await preprocessor.addCucumberPreprocessorPlugin(on, config);  
  
 on(  
 "file:preprocessor",  
 createBundler({  
 plugins: [createEsbuildPlugin.default(config)],  
 })  
 );  
  
 // Make sure to return the config object as it might have been modified by the plugin.  
 return config;  
}  
  
module.exports = defineConfig({  
 e2e: {  
 specPattern: "\*\*/\*.feature",  
 supportFile: false,  
 setupNodeEvents,  
 },  
});

make file with below code.



Code:

**const report = require("multiple-cucumber-html-reporter")**

**report.generate({**

**jsonDir: "mochawesome-report/",**

**reportPath: "cypress/Output/cucumber-html-reports.html",**

**metadata: {**

**browser: {**

**name: "chrome",**

**version: "XX",**

**},**

**device: "Local machine",**

**platform: {**

**name: "Windows",**

**version: "11",**

**},**

**},**

**})**