Official website: https://www.cypress.io/

https://docs.cypress.io/guides/overview/why-cypress

Note: cypress only supports `css selectors`

Syntax: #id, .class

To write css for parent to child – use space

Cypress Step-by-Step Installation:

What is Node.js?

Node.js is an open-source, cross-platform, backend JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

Pre-requisites:

1. Download Node.js

2. Download Visual Studio Code

Step1: Create a new project with "Package.json"

1. *A "package.json" is a JSON file at the root of a JavaScript/Node project.*
2. *It holds metadata relevant to the project and is used for managing project dependencies.*
3. *From <root_folder> where package.json is present from there hit "npm install" to install all dependencies*

Note:

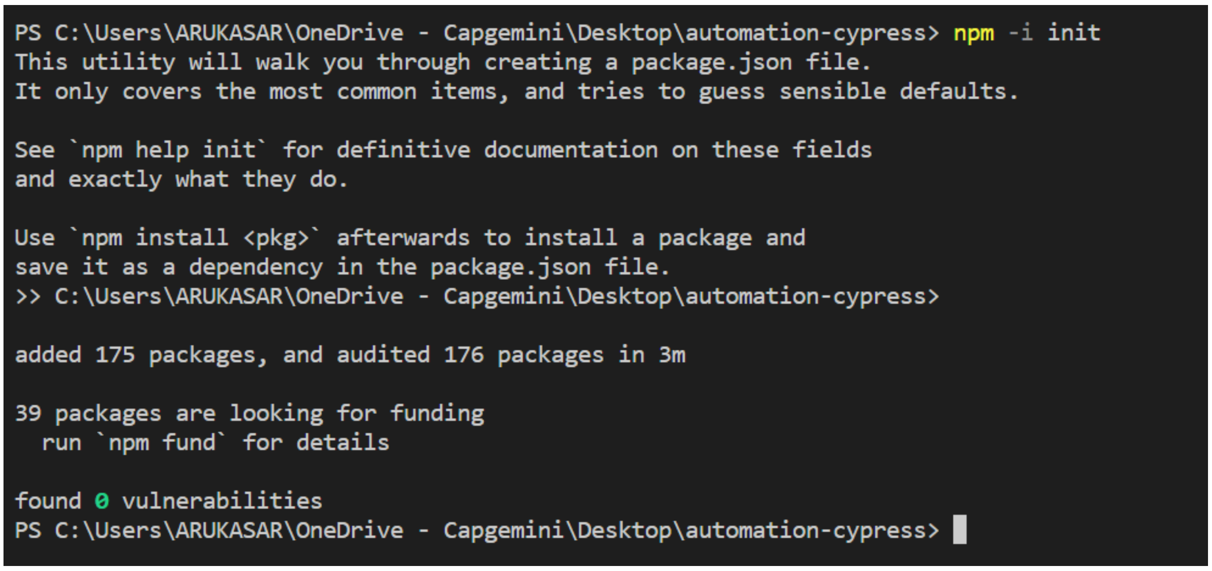
- Make a New Directory -- mkdir automation-cypress.
- Change Directory -- cd..
- Create a "package.json" file using the "npm -i init" command. And do not forget to hit enter.
- To update npm use the command "npm install -g npm"

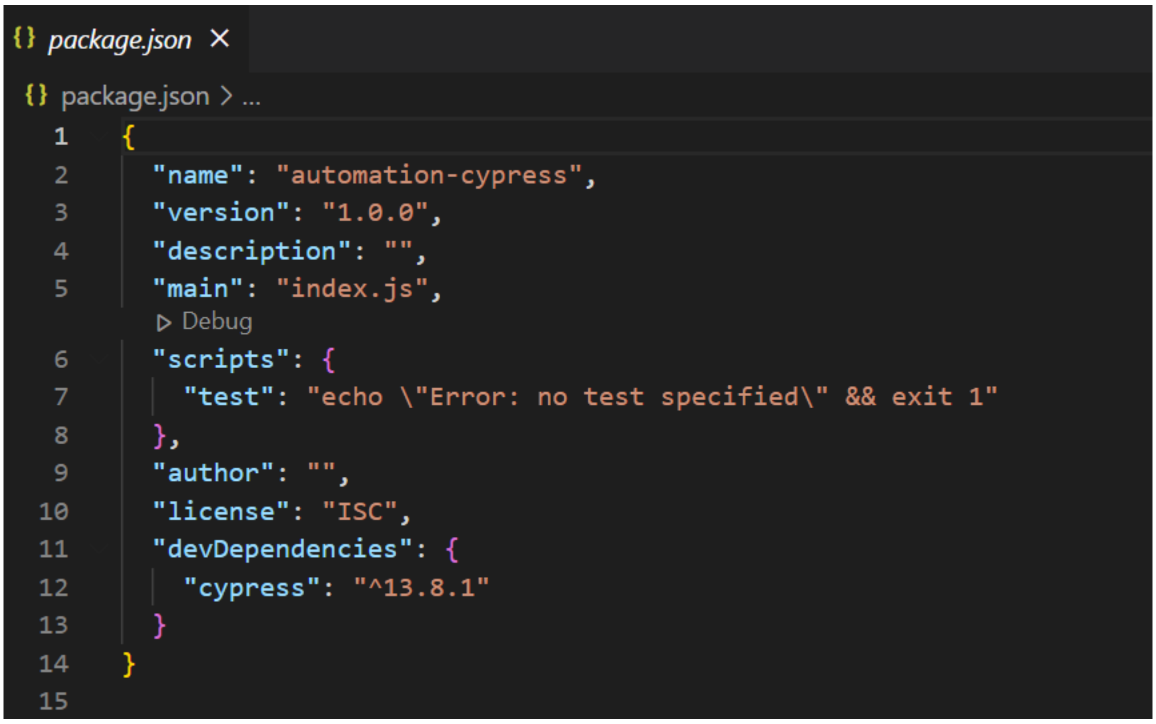
Install Cypress:

Installing: "npm install"

Install Cypress via npm [if cypress version is given in package.json]: "npm install cypress"

To save or create the entry in package.json use the command "npm install <package name> --save-dev": npm install cypress --save-dev


```
PS C:\Users\ARUKASAR\OneDrive - Capgemini\Desktop\automation-cypress> npm -i 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.
>> C:\Users\ARUKASAR\OneDrive - Capgemini\Desktop\automation-cypress>

added 175 packages, and audited 176 packages in 3m

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

found 0 vulnerabilities
PS C:\Users\ARUKASAR\OneDrive - Capgemini\Desktop\automation-cypress>
```

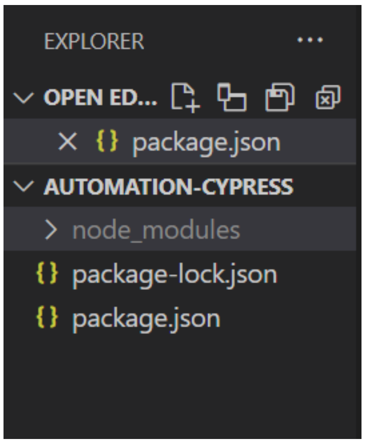


```
{
  "name": "automation-cypress",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "cypress": "^13.8.1"
  }
}
```

After Step1

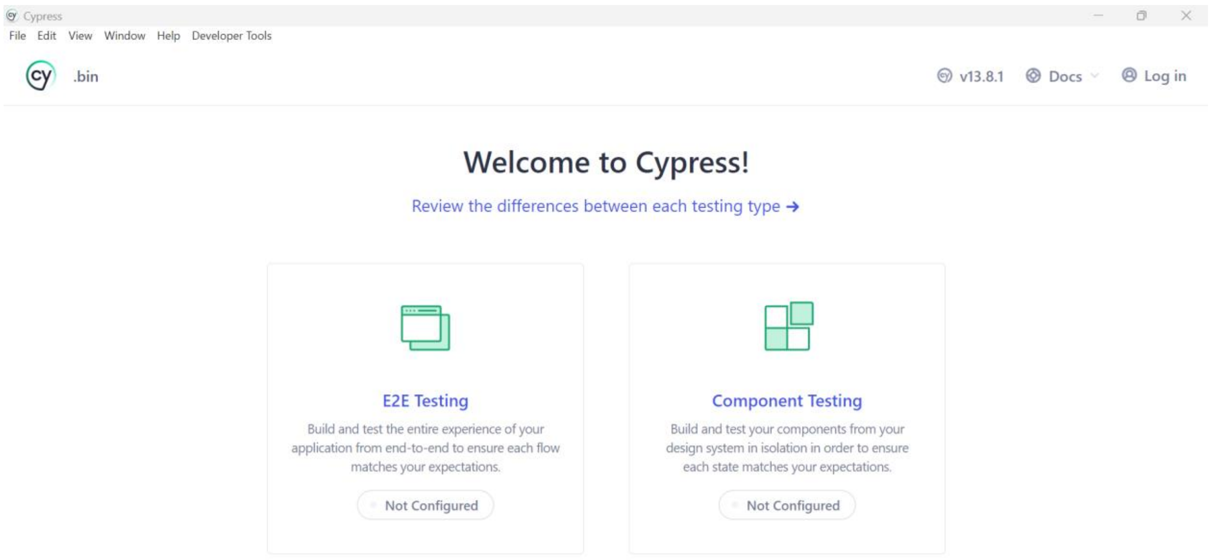
Step2: “package-lock.json” will create automatically

Step3: The “node_modules” folder also will be created automatically.

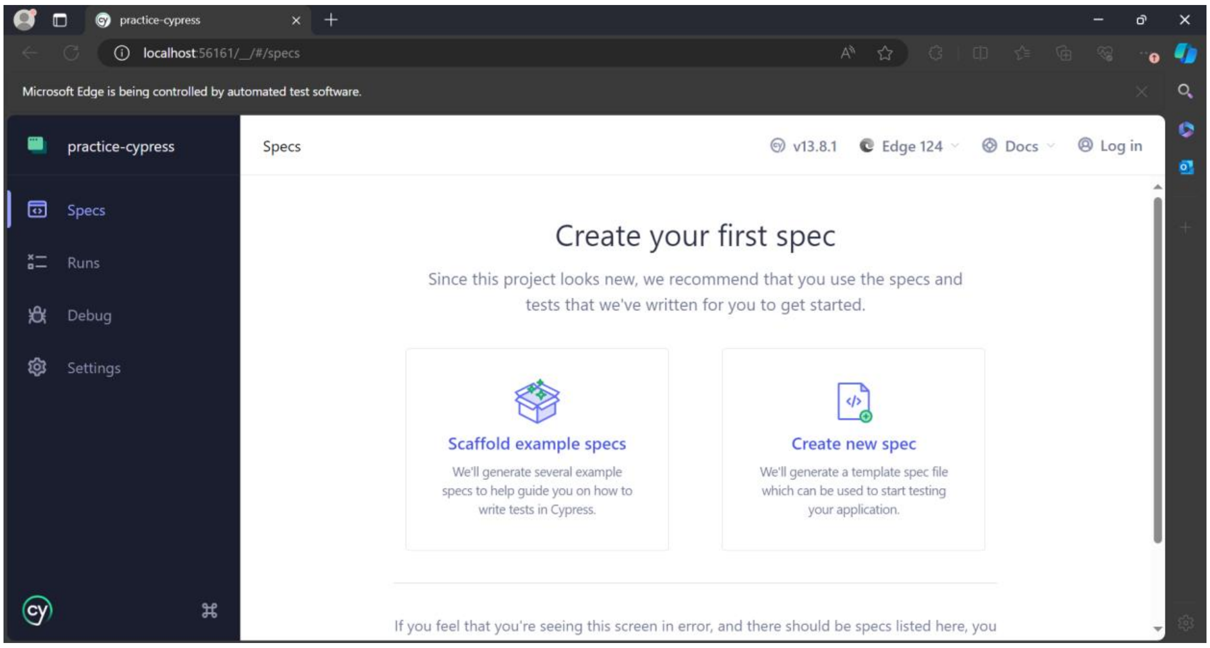


After Step2 and 3

Step4: To run cypress use the command "npx cypress open" or refer: https://docs.cypress.io/guides/getting-started/opening-the-app


Cypress
File Edit View Window Help Developer Tools
.bin
v13.8.1
Docs
Log in
Welcome to Cypress!
Review the differences between each testing type →
E2E Testing
Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.
Not Configured
Component Testing
Build and test your components from your design system in isolation in order to ensure each state matches your expectations.
Not Configured

– *Click on E2E Testing and then continue.*


practice-cypress
localhost:56161/__/#/specs
Microsoft Edge is being controlled by automated test software.
practice-cypress
Specs
Runs
Debug
Settings
Specs
v13.8.1
Edge 124
Docs
Log in

Create your first spec

Since this project looks new, we recommend that you use the specs and tests that we've written for you to get started.

Scaffold example specs

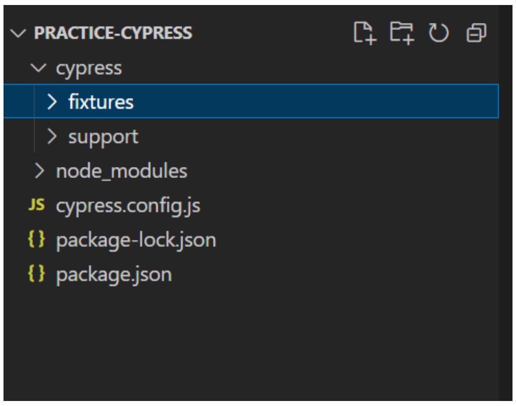
We'll generate several example specs to help guide you on how to write tests in Cypress.

Create new spec

We'll generate a template spec file which can be used to start testing your application.

If you feel that you're seeing this screen in error, and there should be specs listed here, you

– Cypress folder is created automatically:



★ Write our first test case:

Write test cases inside cypress folder which was created automatically.

Save the test cases file using `.js` extension

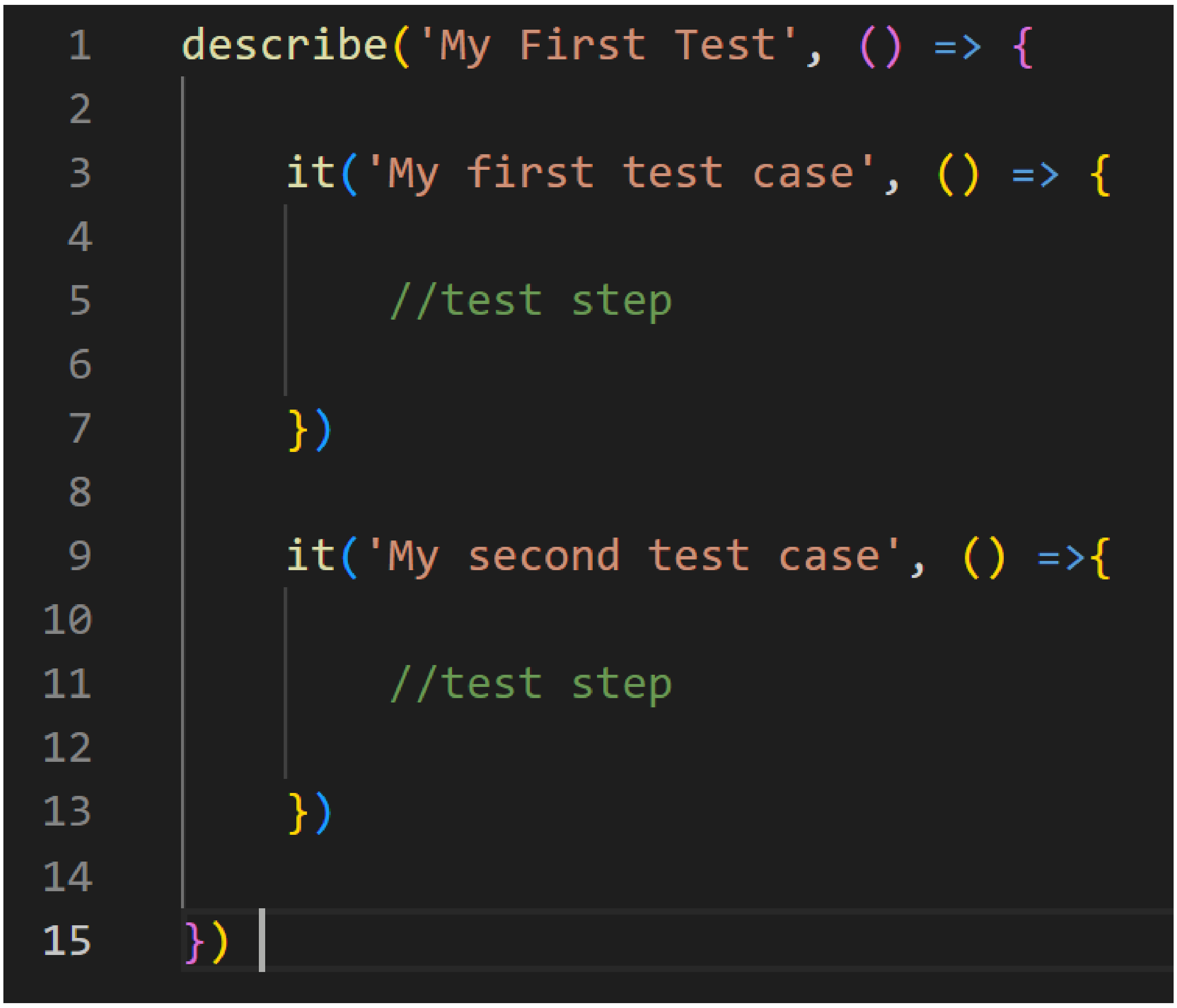
Now it's time to write your first test. We're going to:

- Write your first passing test.
- Update it so it fails.
- Watch Cypress reload in real time.

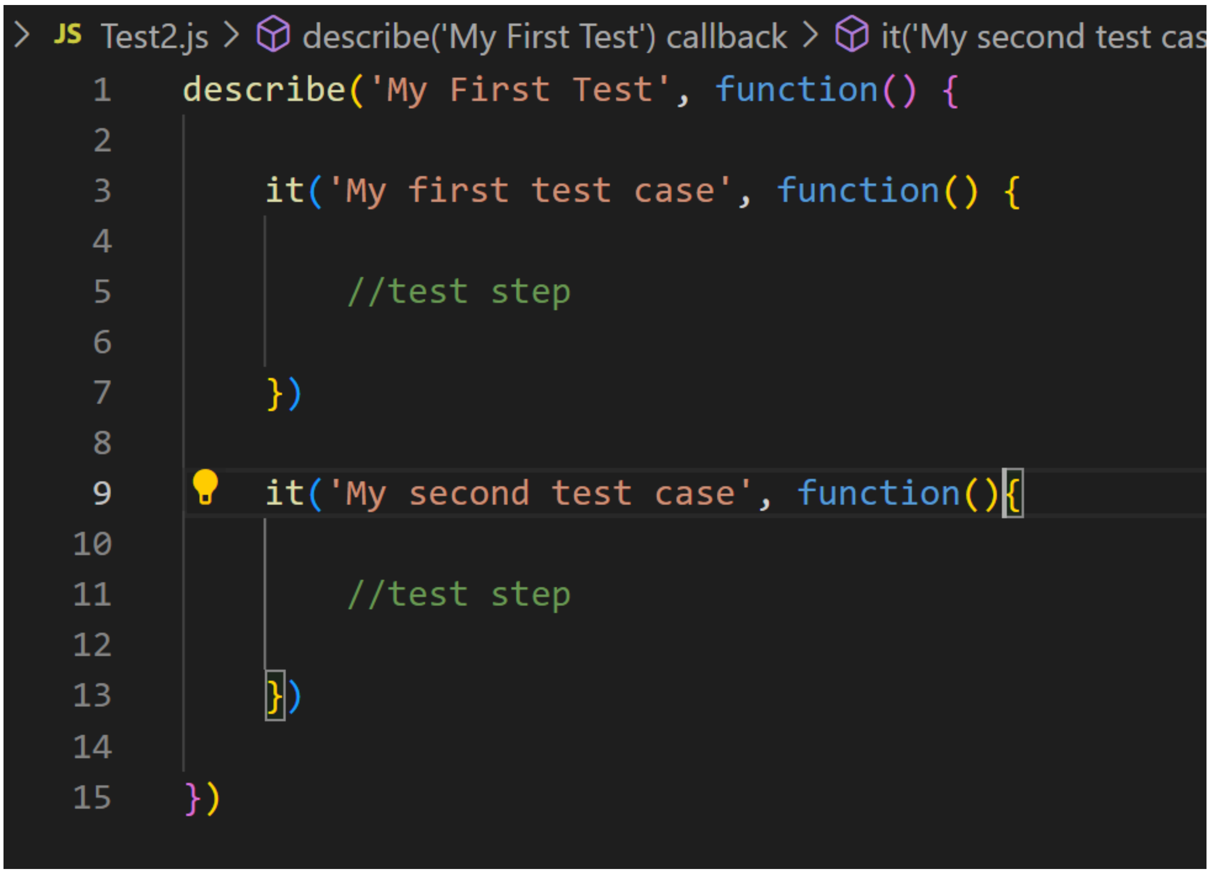
```
/*Describe take two arguments
1. About description
2. Function that wraps all its blocks

it
1. description
2. function
```

```
 For example:
 describe('My First Test', () => {
    it('My first test case', () => {
        //test step
    })
    it('My second test case', () =>{
        //test step
    })
})

*/
```

Syntax:



```
describe('My First Test', () => {

    it('My first test case', () => {

        //test step

    })

    it('My second test case', () =>{

        //test step

    })

})
```

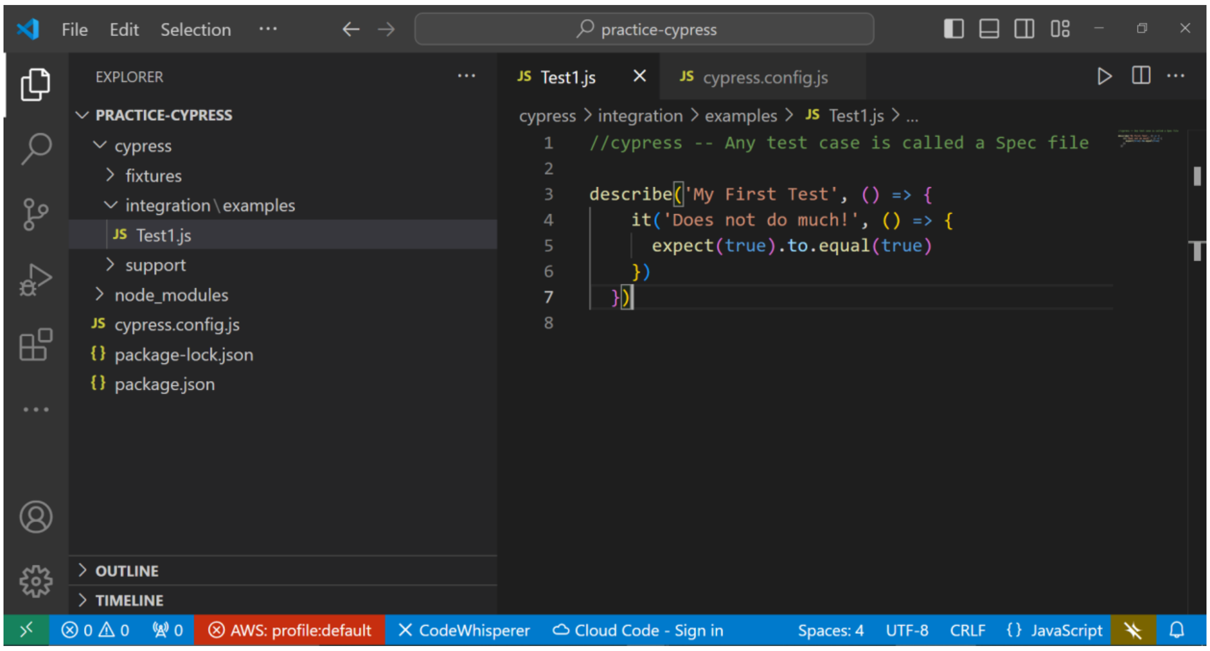


```
> JS Test2.js > describe('My First Test') callback > it('My second test cas
describe('My First Test', function() {

    it('My first test case', function() {

        //test step

    })

    it('My second test case', function(){

        //test step

    })

})
```

Open your favorite IDE and replace the contents of your spec with the code below.

```
//cypress -- Any test case is called a Spec file
```

In this dummy test case: Although it doesn't do anything useful, this is our first passing test! ✓

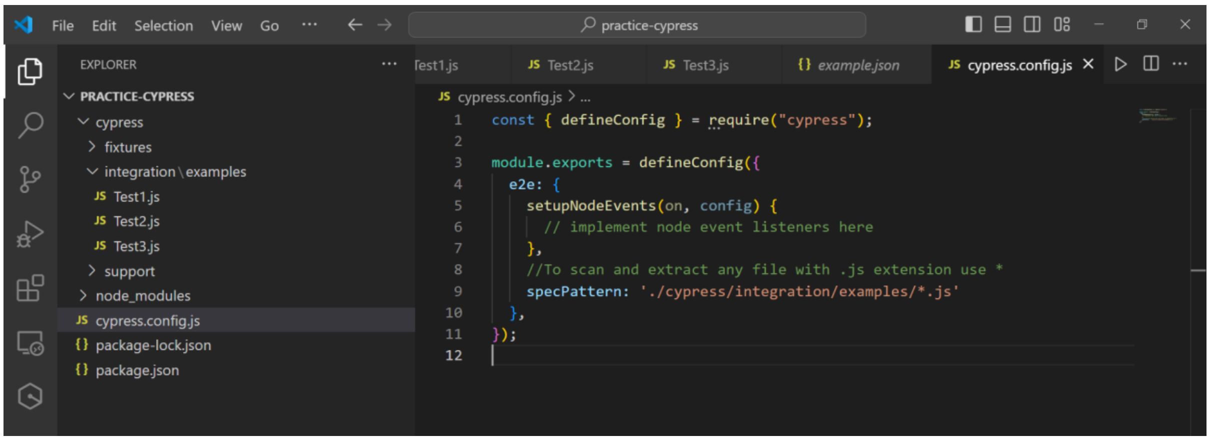


```
//cypress -- Any test case is called a Spec file

describe('My First Test', () => {
    it('Does not do much!', () => {
      expect(true).to.equal(true)
    })
  })

```

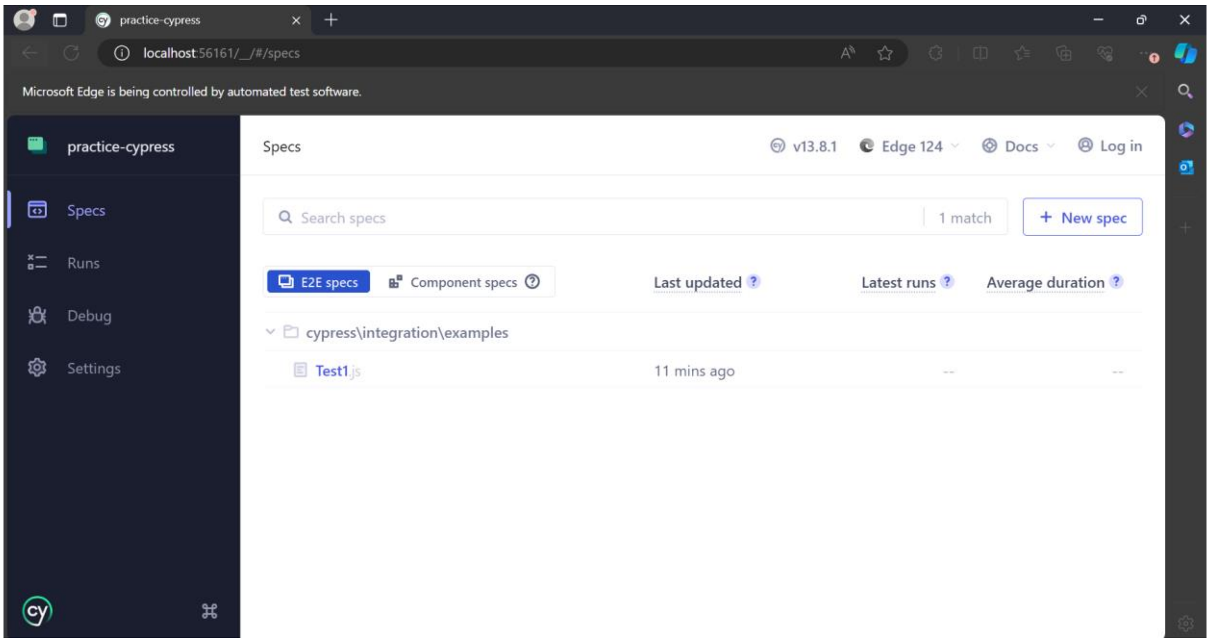
To tell the test runner where your test cases have been written we need to give path in "config.json file"

Add new field: `//To scan and extract any file with .js extension use *`

```
    specPattern: './cypress/integration/examples/*.js
'
```



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

module.exports = defineConfig({
  e2e: {
    setupNodeEvents(on, config) {
      // implement node event listeners here
    },
    //To scan and extract any file with .js extension use *
    specPattern: './cypress/integration/examples/*.js'
  },
});
```

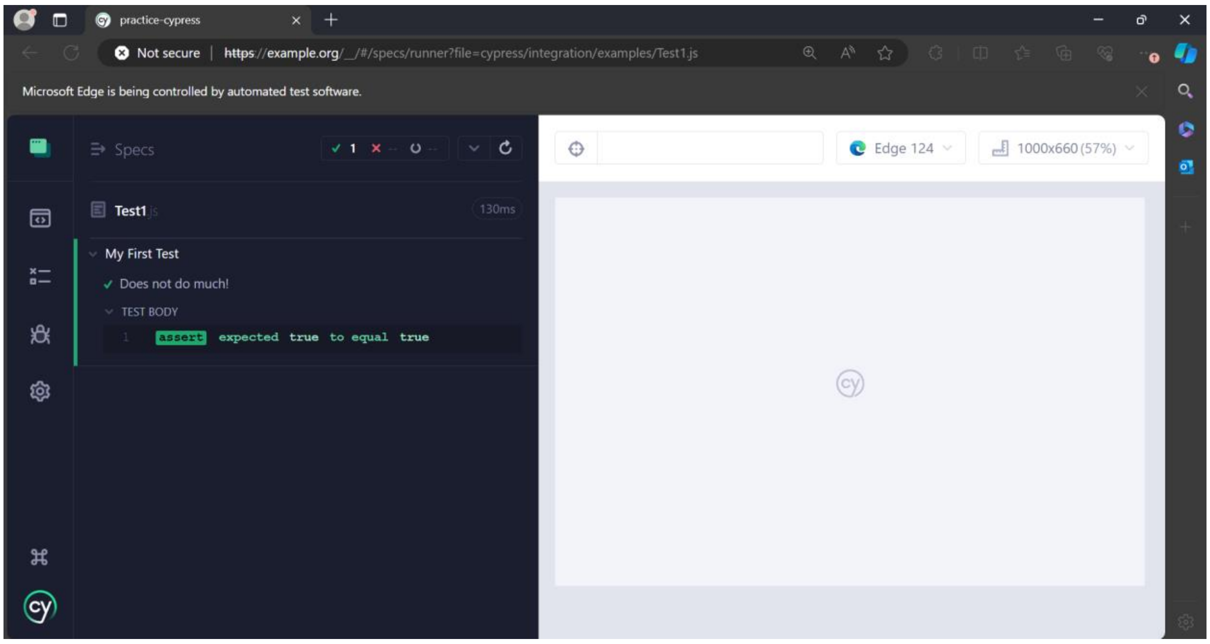


Microsoft Edge is being controlled by automated test software.

practice-cypress — Specs — v13.8.1 — Edge 124

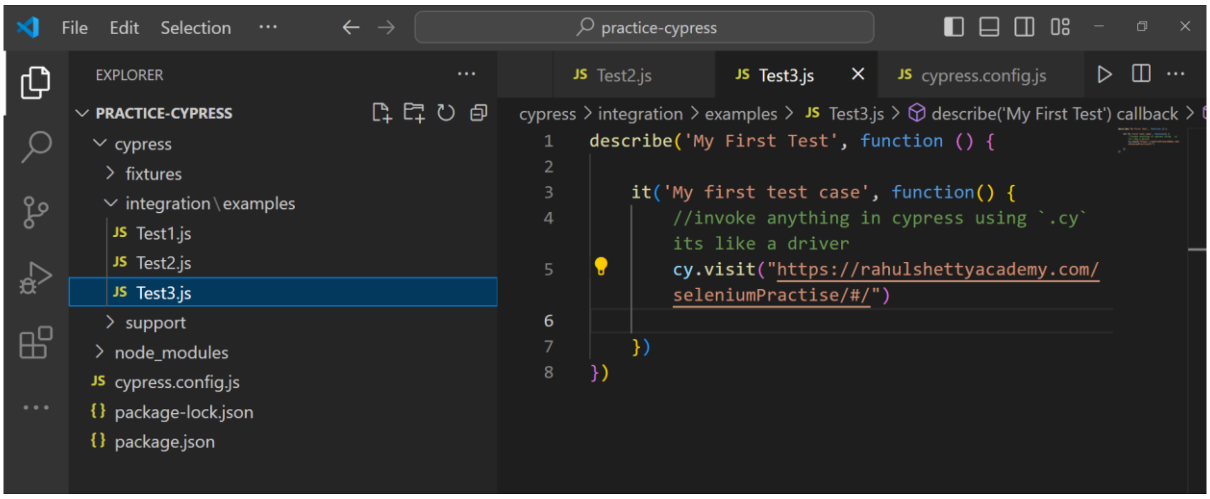
cypress\integration\examples

	Last updated	Latest runs	Average duration
Test1.js	11 mins ago	--	--

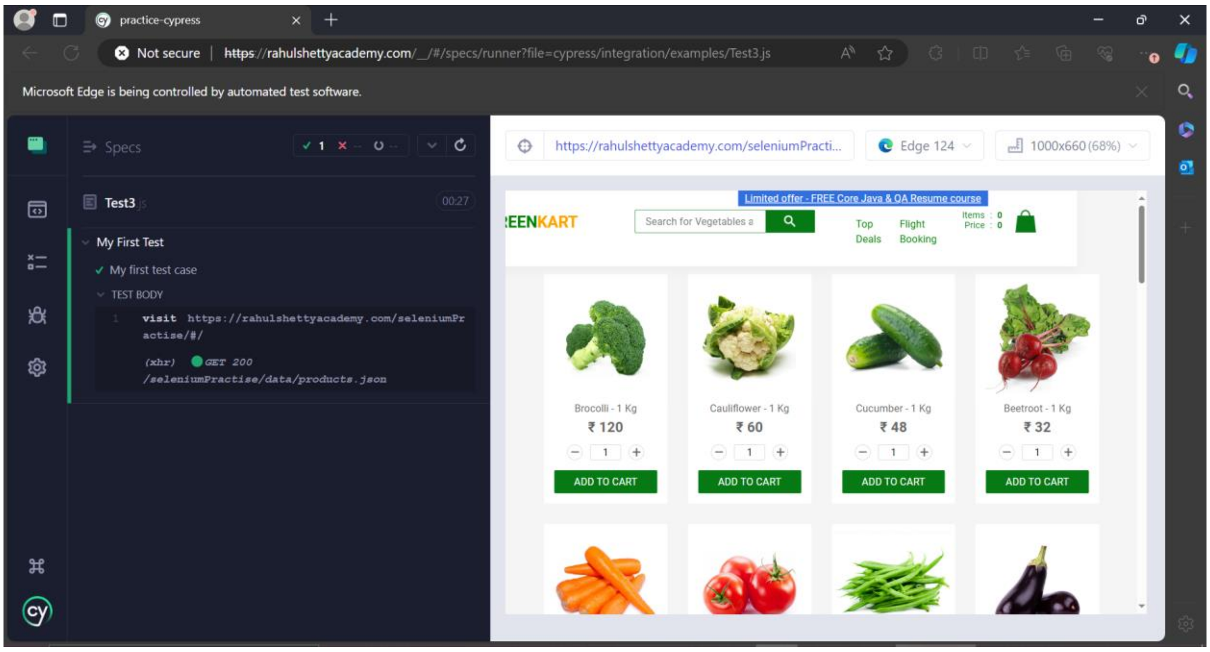

practice-cypress
Not secure | https://example.org/__/#/specs/runner?file=cypress/integration/examples/Test1.js
Microsoft Edge is being controlled by automated test software.
Specs
Test1.js
130ms
Edge 124
1000x660 (57%)

```
My First Test
✓ Does not do much!
TEST BODY
1  assert expected true to equal true
```

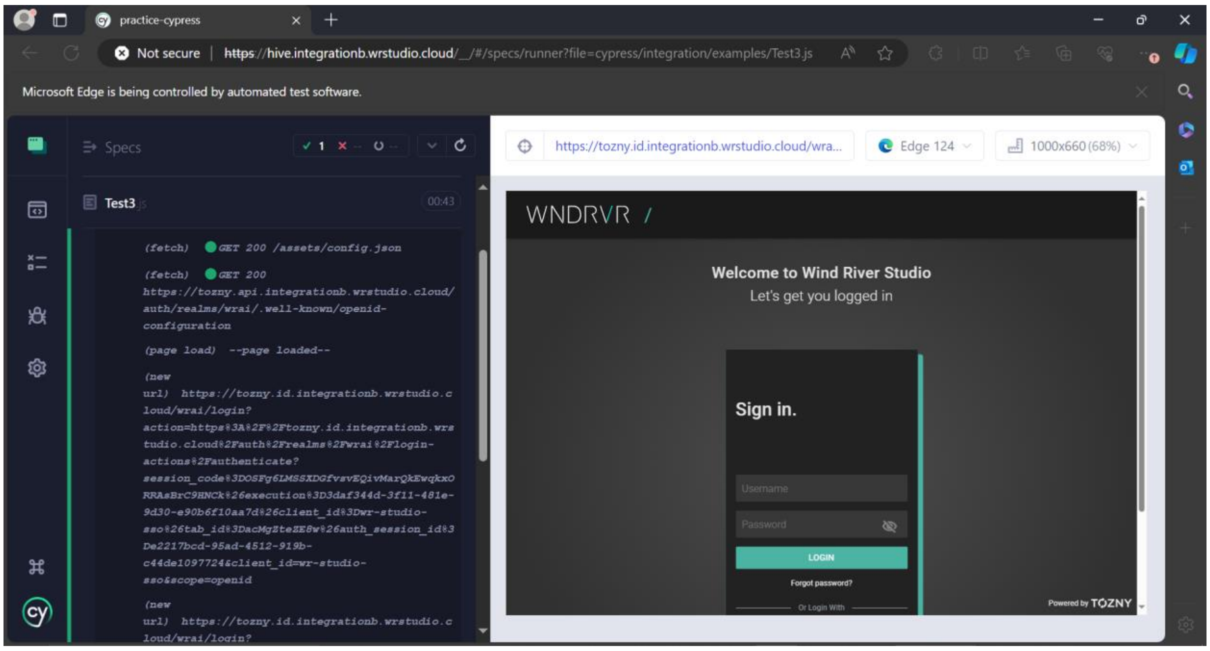
First test case:



```
describe('My First Test', function () {

    it('My first test case', function() {
        //invoke anything in cypress using `.cy`
        its like a driver
        cy.visit("https://rahulshettyacademy.com/
        seleniumPractise/#/")

    })
})
```



```
Microsoft Edge is being controlled by automated test software.

Test3.js                                             00:27
My First Test
  ✓ My first test case
  TEST BODY
    visit  https://rahulshettyacademy.com/seleniumPractise/#/
    (xhr)  ● GET 200 /seleniumPractise/data/products.json
```

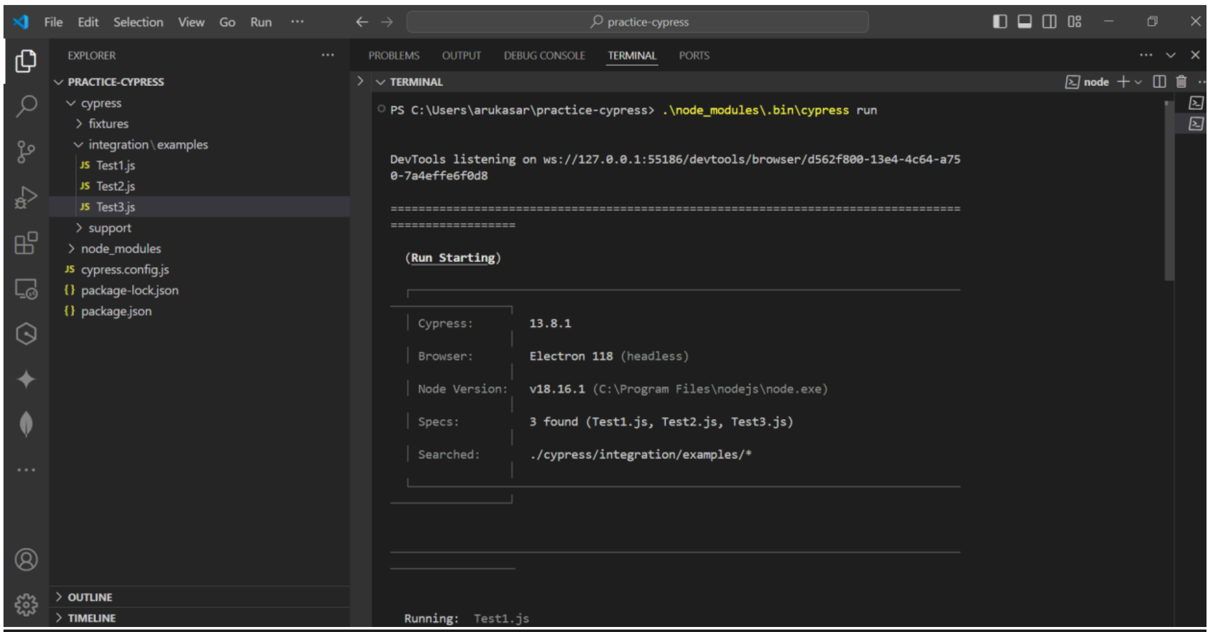


```
Microsoft Edge is being controlled by automated test software.

Test3.js                                             00:43
    (fetch)  ● GET 200 /assets/config.json
    (fetch)  ● GET 200
    https://tozny.api.integrationb.wrstudio.cloud/
    auth/realms/wrai/.well-known/openid-
    configuration
    (page load)  --page loaded--
    (new
    url)  https://tozny.id.integrationb.wrstudio.c
    loud/wrai/login?
    action=https%3A%2F%2Ftozny.id.integrationb.wrs
    tudio.cloud%2Fauth%2Frealms%2Fwrai%2Flogin-
    actions%2Fauthenticate?
    session_code%3DOSFg6LMSSXDGfvsvEQivMarQkEwqkxO
    RRAsBrC9HNCk%26execution%3D3daf344d-3f11-481e-
    9d30-e90b6f10aa7d%26client_id%3Dwr-studio-
    sso%26tab_id%3DacMgZteZE8w%26auth_session_id%3
    De2217bcd-95ad-4512-919b-
    c44de1097724&client_id=wr-studio-
    sso&scope=openid
    (new
    url)  https://tozny.id.integrationb.wrstudio.c
    loud/wrai/login?
```

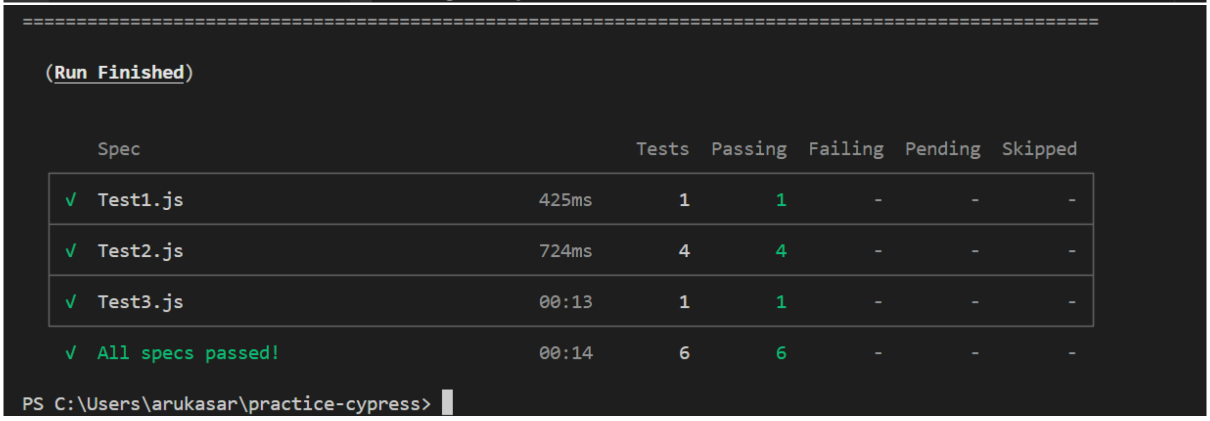
How to run tests from cmd?

.\node_modules\.bin\cypress run

It will execute all the test cases without opening any browser automatically.



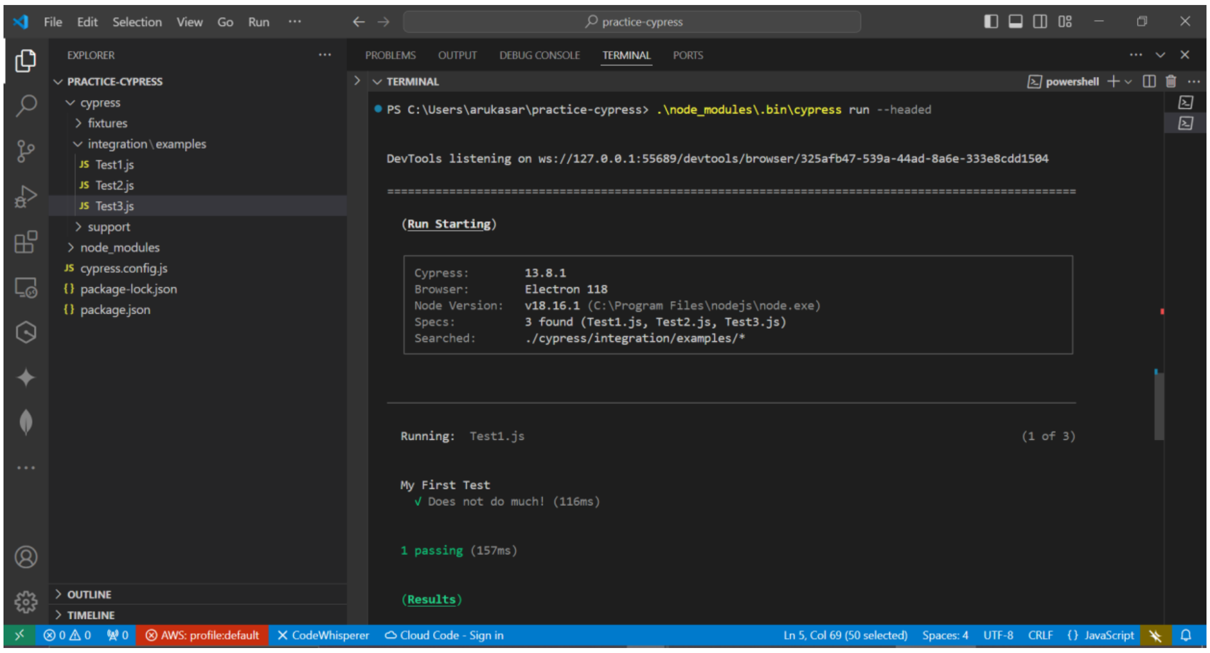
```
PS C:\Users\arukasar\practice-cypress> .\node_modules\.bin\cypress run

DevTools listening on ws://127.0.0.1:55186/devtools/browser/d562f800-13e4-4c64-a75
0-7a4effe6f0d8

==================================================================================
=================

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:        13.8.1                                                         │
  │ Browser:        Electron 118 (headless)                                        │
  │ Node Version:   v18.16.1 (C:\Program Files\nodejs\node.exe)                    │
  │ Specs:          3 found (Test1.js, Test2.js, Test3.js)                         │
  │ Searched:       ./cypress/integration/examples/*                               │
  └────────────────────────────────────────────────────────────────────────────────┘

────────────────────────────────────────────────────────────────────────────────────

  Running:  Test1.js
```



```
====================================================================================================

  (Run Finished)


       Spec                                              Tests  Passing  Failing  Pending  Skipped
  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ √  Test1.js                                 425ms        1        1        -        -        - │
  ├────────────────────────────────────────────────────────────────────────────────────────────────┤
  │ √  Test2.js                                 724ms        4        4        -        -        - │
  ├────────────────────────────────────────────────────────────────────────────────────────────────┤
  │ √  Test3.js                                 00:13        1        1        -        -        - │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘
    √  All specs passed!                        00:14        6        6        -        -        -

PS C:\Users\arukasar\practice-cypress>
```

.\node_modules\.bin\cypress run –headed

It will automatically invoke electron browser and run all the test cases after completion it will close itself.



```
PS C:\Users\arukasar\practice-cypress> .\node_modules\.bin\cypress run --headed

DevTools listening on ws://127.0.0.1:55689/devtools/browser/325afb47-539a-44ad-8a6e-333e8cdd1504

====================================================================================================

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:        13.8.1                                                                         │
  │ Browser:        Electron 118                                                                   │
  │ Node Version:   v18.16.1 (C:\Program Files\nodejs\node.exe)                                    │
  │ Specs:          3 found (Test1.js, Test2.js, Test3.js)                                         │
  │ Searched:       ./cypress/integration/examples/*                                               │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘


────────────────────────────────────────────────────────────────────────────────────────────────────

  Running:  Test1.js                                                                        (1 of 3)


  My First Test
    √ Does not do much! (116ms)


  1 passing (157ms)


  (Results)
```

```
.\node_modules\.bin\cypress run --browser <name>
```

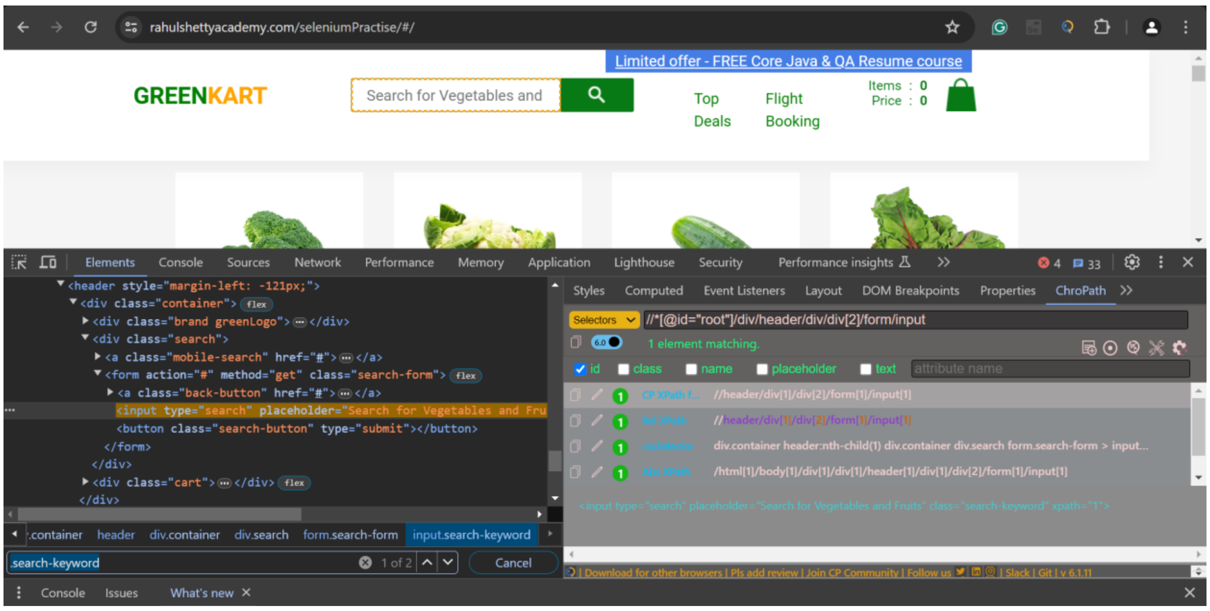
____Framework:

```
By hitting: cypress open

cypress
1. fixtures -- responsible to store test data, if your test case have some data, it
should be extracted from external files for ex: xml, json etc.

2. e2e -- this is used to write test cases

3. plugin -- not much used, listeners like maximize window and give error message, to
handle cypress events etc.

4. support -- write reusable methods, so that e2e test cases can use it.
file Support/commad.js

cypress.cofig.json -- auto-generated when you open cypress 1st time. Configuration
file for entire framework.8

For example:
//To scan and extract any file with .js extension use *
specPattern: './cypress/integration/examples/*.js'
```

____

Locators [CSS Selector only]

Extension--chropath



```
<header style="margin-left: -121px;">
  <div class="container"> flex
    <div class="brand greenLogo">…</div>
    <div class="search">
      <a class="mobile-search" href="#">…</a>
      <form action="#" method="get" class="search-form"> flex
        <a class="back-button" href="#">…</a>
        <input type="search" placeholder="Search for Vegetables and Fru
        <button class="search-button" type="submit"></button>
      </form>
    </div>
    <div class="cart">…</div> flex
  </div>
```

.container header div.container div.search form.search-form input.search-keyword

.search-keyword   1 of 2

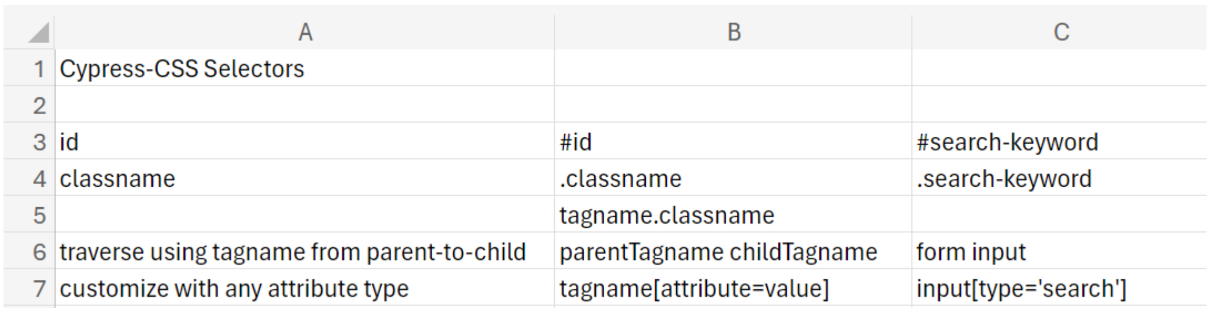
//*[@id="root"]/div/header/div/div[2]/form/input

1 element matching.

CP XPath f...	//header/div[1]/div[2]/form[1]/input[1]
Rel XPath	//header/div[1]/div[2]/form[1]/input[1]
cssSelector	div.container header:nth-child(1) div.container div.search form.search-form > input...
Abs XPath	/html[1]/body[1]/div[1]/div[1]/header[1]/div[1]/div[2]/form[1]/input[1]

```
<input type="search" placeholder="Search for Vegetables and Fruits" class="search-keyword" xpath="1">
```

| Download for other browsers | Pls add review | Join CP Community | Follow us | Slack | Git | v 6.1.11

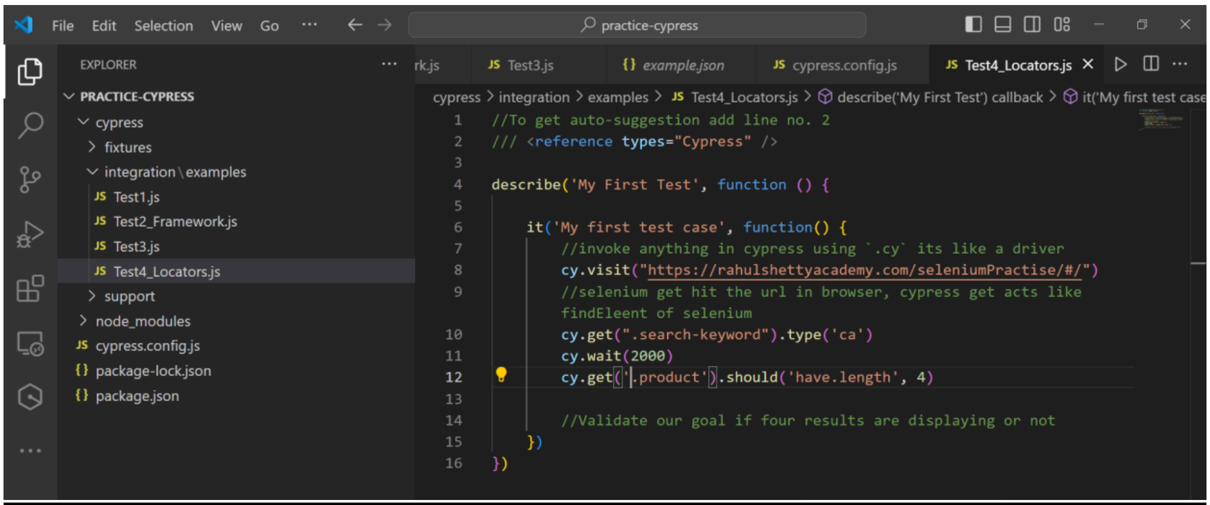


	A	B	C
1	Cypress-CSS Selectors		
2			
3	id	#id	#search-keyword
4	classname	.classname	.search-keyword
5		tagname.classname	
6	traverse using tagname from parent-to-child	parentTagname childTagname	form input
7	customize with any attribute type	tagname[attribute=value]	input[type='search']

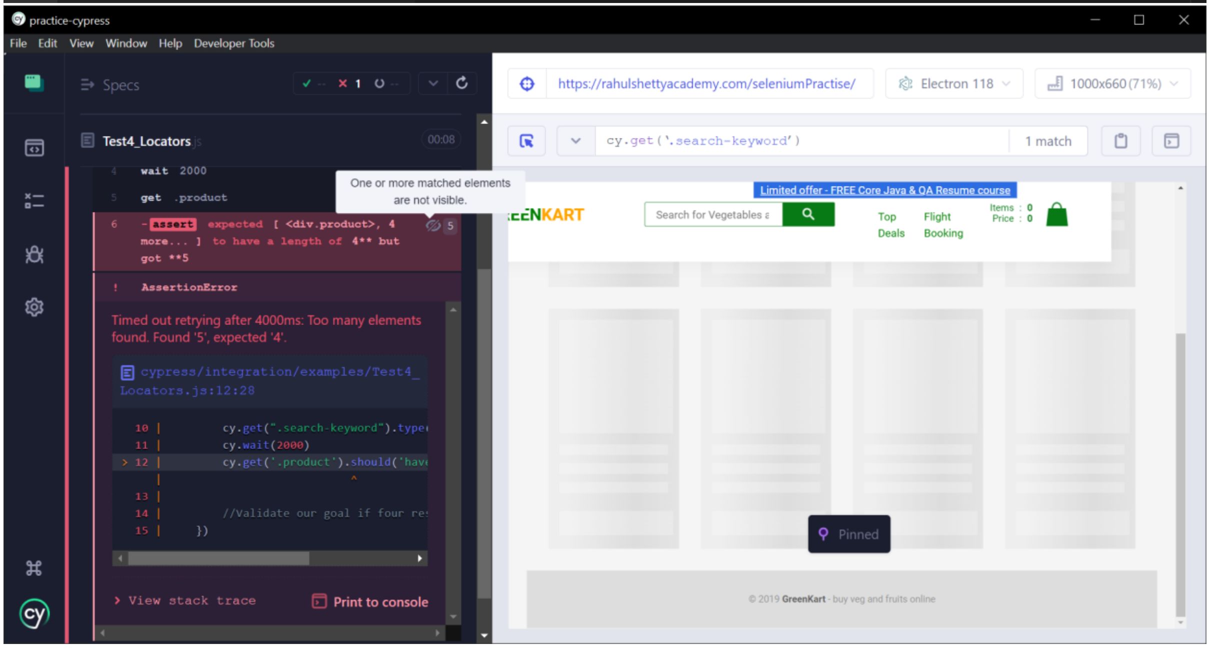
To get auto-suggestion under testcase file write:

```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />
```

With Locator TestCase:



```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My First Test', function () {

    it('My first test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/")
        //selenium get hit the url in browser, cypress get acts like findEleent of selenium
        cy.get(".search-keyword").type('ca')
        cy.wait(2000)
        cy.get('.product').should('have.length', 4)

        //Validate our goal if four results are displaying or not
    })
})
```

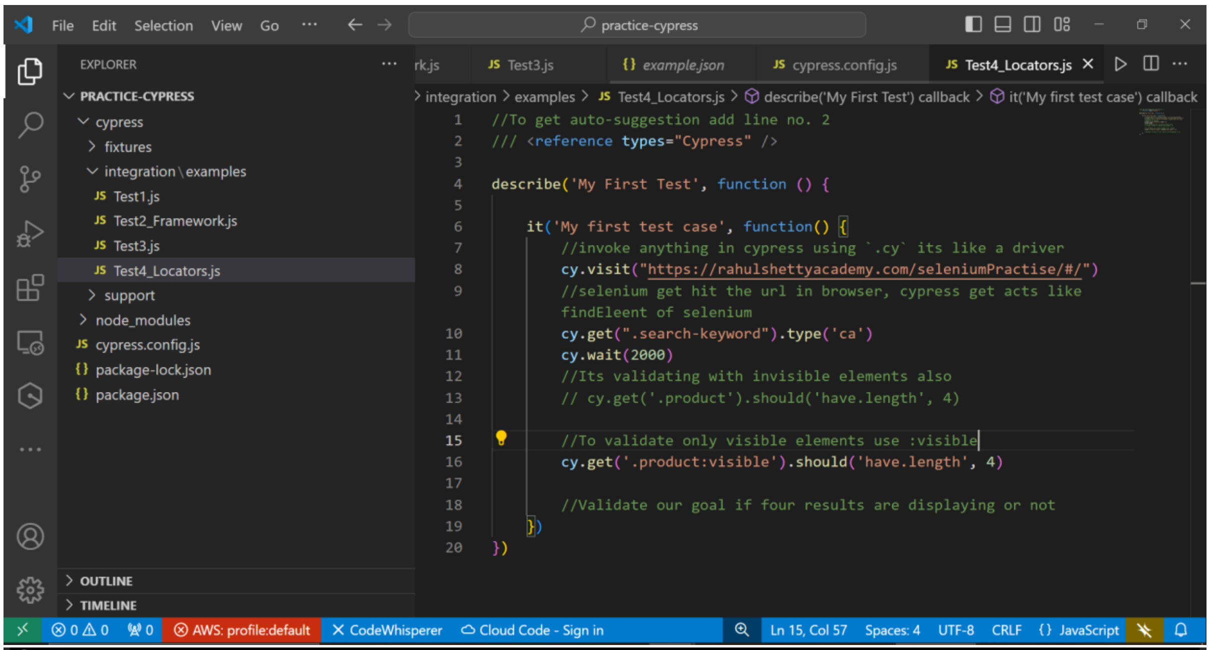


```
Test4_Locators.js                                   00:08
  4  wait  2000
  5  get  .product
  6  -assert  expected [ <div.product>, 4 more... ] to have a length of 4** but got **5
  !  AssertionError

Timed out retrying after 4000ms: Too many elements found. Found '5', expected '4'.

cypress/integration/examples/Test4_Locators.js:12:28

  10 |         cy.get(".search-keyword").type
  11 |         cy.wait(2000)
> 12 |         cy.get('.product').should('have
     |                            ^
  13 |
  14 |         //Validate our goal if four re
  15 |     })

> View stack trace                    Print to console
```

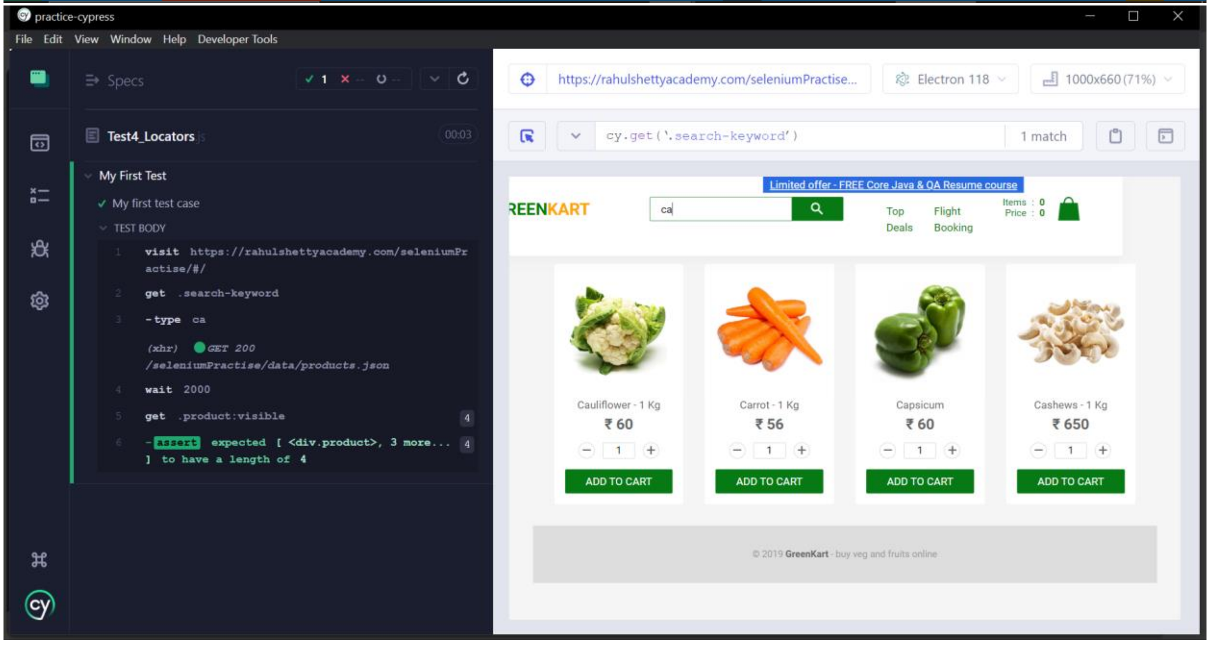
One or more matched elements are not visible.

https://rahulshettyacademy.com/seleniumPractise/   Electron 118   1000x660 (71%)

cy.get('.search-keyword')   1 match



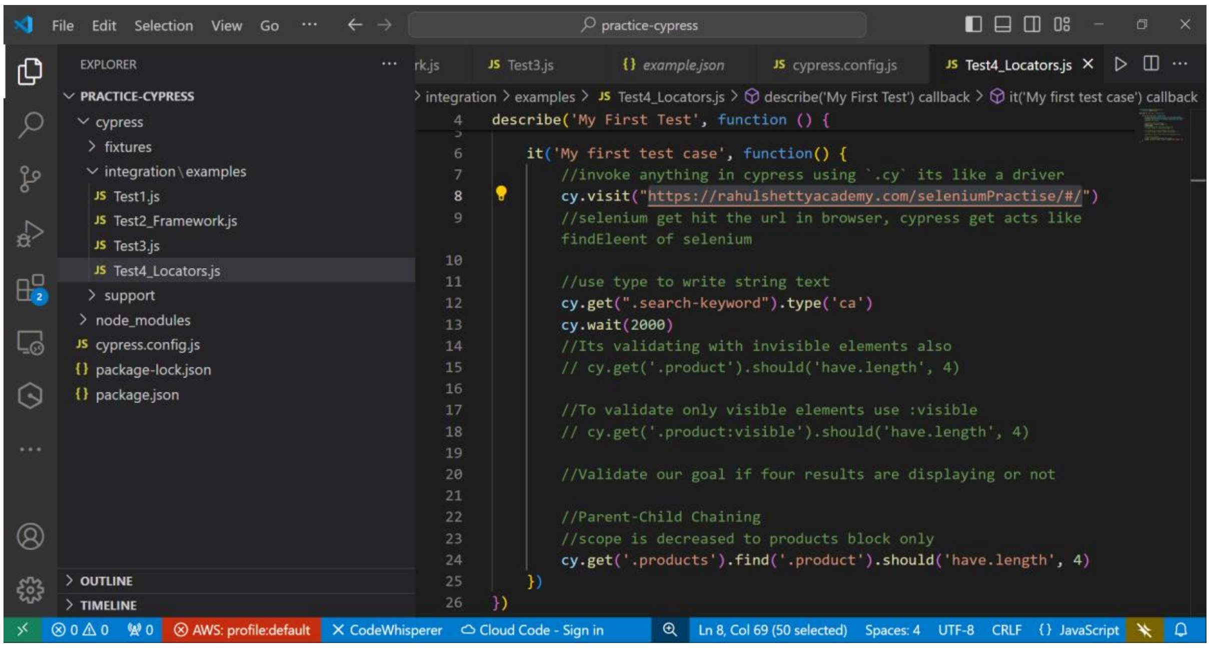
```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My First Test', function () {

    it('My first test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/")
        //selenium get hit the url in browser, cypress get acts like
        findEleent of selenium
        cy.get(".search-keyword").type('ca')
        cy.wait(2000)
        //Its validating with invisible elements also
        // cy.get('.product').should('have.length', 4)

        //To validate only visible elements use :visible
        cy.get('.product:visible').should('have.length', 4)

        //Validate our goal if four results are displaying or not
    })
})
```



```
Test4_Locators.js                                  00:03
My First Test
  ✓ My first test case
    TEST BODY
    1  visit https://rahulshettyacademy.com/seleniumPractise/#/
    2  get .search-keyword
    3  -type ca
       (xhr) ● GET 200 /seleniumPractise/data/products.json
    4  wait 2000
    5  get .product:visible                                 4
    6  -assert expected [ <div.product>, 3 more... ] to have a length of 4    4
```

——

Parent-Child Chaining



```
describe('My First Test', function () {

    it('My first test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/")
        //selenium get hit the url in browser, cypress get acts like
        findEleent of selenium

        //use type to write string text
        cy.get(".search-keyword").type('ca')
        cy.wait(2000)
        //Its validating with invisible elements also
        // cy.get('.product').should('have.length', 4)

        //To validate only visible elements use :visible
        // cy.get('.product:visible').should('have.length', 4)

        //Validate our goal if four results are displaying or not

        //Parent-Child Chaining
        //scope is decreased to products block only
        cy.get('.products').find('.product').should('have.length', 4)
    })
})
```

First UI Test Case: Test4_Locators.js

```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My First Test', function () {

    it('My first test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/")
        //selenium get hit the url in browser, cypress get acts like
findEleent of selenium

        //use type to write string text
        cy.get(".search-keyword").type('ca')
        cy.wait(2000)
        //Its validating with invisible elements also
        // cy.get('.product').should('have.length', 4)

        //To validate only visible elements use :visible
        // cy.get('.product:visible').should('have.length', 4)

        //Validate our goal if four results are displaying or not

        /*Parent-Child Chaining
        scope is decreased to products block only*/
        // cy.get('.products').find('.product').should('have.length', 4)

        /*
        methods: contains, equal, find and get
        Click on Add to Cart
        and resolving the promise
        */

        cy.get('.products').find('.product').eq(2).contains('ADD TO 
CART').click().then(function(){
            console.log("Hello Cypress")
        })

        /* Req: Grab all the product name and add only capsicum when its found
        iterate where capsicum is present
        Method: each*/

        /*
        To RE-USE LOCATORS EVERYTIME
        // Aliases: part of optimization, to act as a variable
        cy.get('.products').as('productsLocator')
        //Before
        cy.get('.products').find('.product').should('have.length', 4)
        //After
        cy.get('@productsLocator').find('.product').should('have.length', 4)
        */

        // Aliases: part of optimization, to act as a variable
        cy.get('.products').as('productsLocator')

        cy.get('@productsLocator').find('.product').each(($e1, index, $list)
=> {

           const textVEG = $e1.find('h4.product-name').text()
           if(textVEG.includes('Cashew'))
           {
            //with .find click method is depreciated so wrap it using 
cy.wrap() method
            cy.wrap($e1).find('button').click()

            console.log('HelloWorld')
           }
        })

        //DO NOT USE
        /*
        const logo = cy.get('.brand')
        //To print something we use method: cy.log()
        cy.log(logo.text())
        */
        //this is to assert if logo text is correctly displayed
        cy.get('.brand').should('have.text', 'GREENKART')

        //this is to print in logs
        //DO USE
       cy.get('.brand').then(function(logoelement)
       {
        //To print something we use method: cy.log()
        cy.log(logoelement.text())
    })



    })
})
```

Test5.js

```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My Second Test', function () {

    it('My second test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/")

        cy.get(".search-keyword").type('ca')
        cy.wait(2000)

        cy.get('.products').as('productsLocator')

        cy.get('@productsLocator').find('.product').each(($e1, index, $list)
=> {

           const textVEG = $e1.find('h4.product-name').text()
           if(textVEG.includes('Cashews'))
           {
            cy.wrap($e1).find('button').click()
           }
        })
        cy.get('.cart-icon').click()
        // Locator of PROCEED TO CHECKOUT--('button[css="1"]')
        cy.contains('PROCEED TO CHECKOUT').click()
        cy.contains('Place Order').click()
    })
})
```
___

My Understanding:

Here's how you can structure your Cypress test suites and API endpoints for the mentioned modules:

API Endpoints:

- **architectures.js**: Defines API endpoints related to architectures.

- **bsp.js**: Contains API endpoints for BSP (Board Support Packages).

- **networkInterface.js**: Includes API endpoints related to network interfaces.

Alternative URLs:

- **routes.js**: Contains alternative URLs for the vLab module.

Page Objects:

- **Common Components**:

  - **navBar.js**: Defines navigation bar elements with locators and keywords.

  - **sideNav.js**: Contains locators and keywords for the side navigation.

- **Pages**:

  - **common**: Page elements common across different modules.

  - **vLab**: Page elements specific to the vLab module.

  - **vLabAdministration**: Page elements related to vLab administration.

Test Cases:

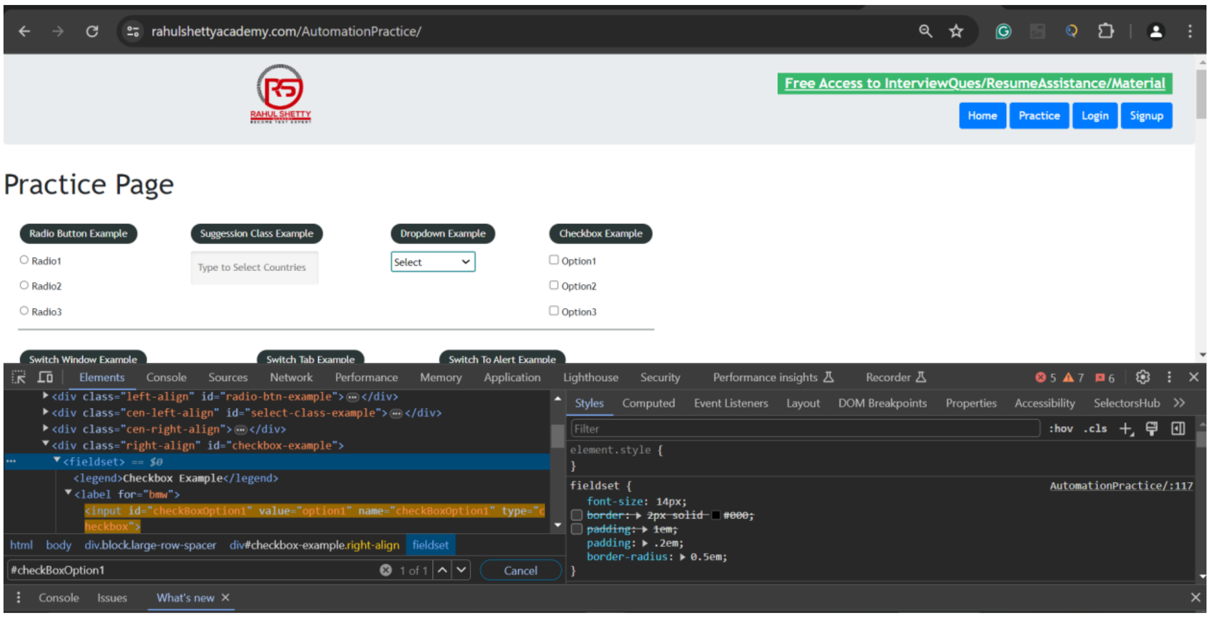
- **testSuites**: Contains test suites covering various scenarios.

- **vLabAdministration**: Test cases specifically for vLab administration.

- **vLabMenu**: Test cases related to the vLab menu functionality.

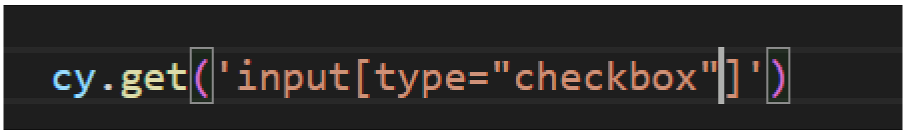
In your Cypress setup, you can organize your code according to these modules and folders for easier navigation and maintenance.

How to write CSS for checkboxes



```
<div class="left-align" id="radio-btn-example">⋯</div>
<div class="cen-left-align" id="select-class-example">⋯</div>
<div class="cen-right-align">⋯</div>
<div class="right-align" id="checkbox-example">
  <fieldset> == $0
    <legend>Checkbox Example</legend>
    <label for="bmw">
      <input id="checkBoxOption1" value="option1" name="checkBoxOption1" type="checkbox">
```

```
element.style {
}
fieldset {                                AutomationPractice/:117
    font-size: 14px;
    border: 2px solid #000;
    padding: 1em;
    padding: .2em;
    border-radius: 0.5em;
}
```



```
cy.get('input[type="checkbox"]')
```

```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My Third Test', function () {

    it('My Third test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

        /*verify if the checkbox is check or not
        validate if option1 is clicked or not*/

        //First: It will check option1

cy.get('#checkBoxOption1').check().should('be.checked').and('have.value','opti
on1')

        //Second: It will uncheck option1
        cy.get('#checkBoxOption1').uncheck().should('not.be.checked')

        //Third: It will check option2 and 3
        cy.get('input[type="checkbox"]').check(['option2','option3'])

        //Four: It will uncheck option2 and 3
        cy.get('input[type="checkbox"]').uncheck().should('not.be.checked')




    })
})
```

When dealing with dynamic dropdowns, if the options disappear after typing an individual text into a search box, you can use the inspect option in your browser's developer tools to identify the CSS path of the elements.

```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My Fourth Test', function () {

    it('My Fourth test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

       /*
       dynamic dropdowns -- means options  will be displayed based on inputs
provides
       [tagname: select] static dropdowns -- means we can select from the
given suggestions
       */

       //Static Dropdowns
       cy.get('select').select('option2').should('have.value', 'option2')

        //Dynamic dropdowns
        cy.get('#autocomplete').type('ind')
        cy.get('.ui-menu-item div').each(($el, index, $list) => {

        if($el.text()==="India")
        {
            $el.click()
        }
    })
    //verifying if its have value as india or not
    cy.get('#autocomplete').should('have.value', 'India')

    })
})
```

Hide and Show

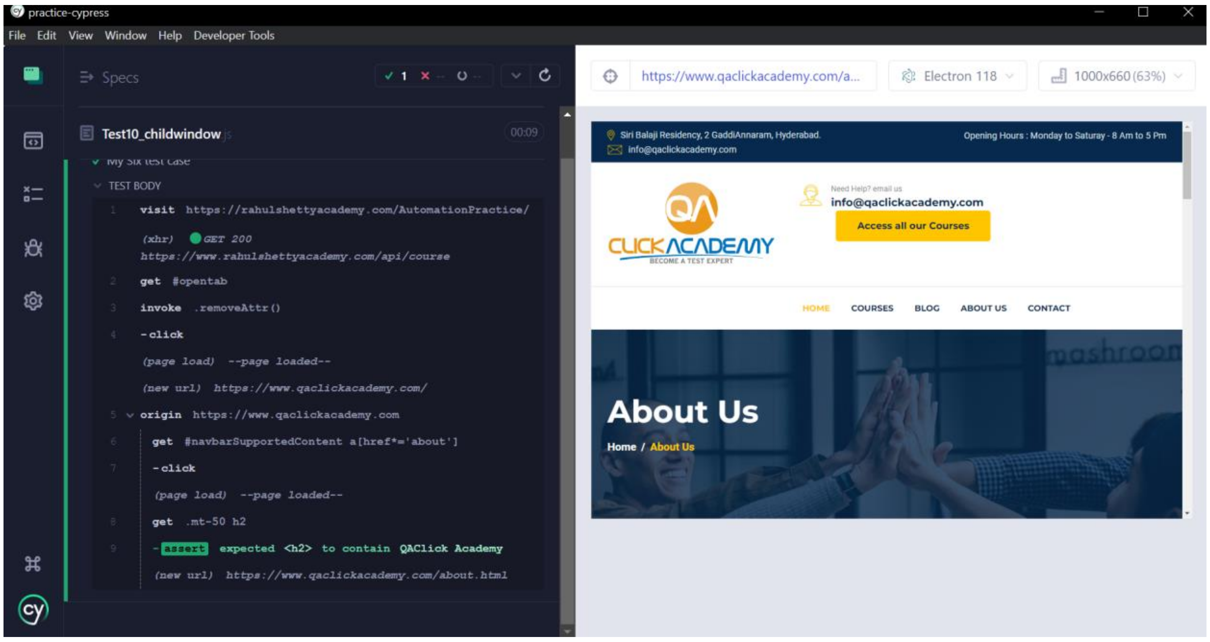
```
//To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My Third Test', function () {

    it('My Third test case', function() {
        //invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

       cy.get("#displayed-text").should('be.visible')
       cy.get("#hide-textbox").click()

       cy.get("#displayed-text").should('not.be.visible')
       cy.get("#show-textbox").click()
       cy.get("#displayed-text").should('be.visible')


    })
})
```

Alert:

```
// To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My Fifth Test', function () {

    it('My Fifth test case', function() {
        // invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

        // cypress auto accepts alerts
        cy.get('#alertbtn').click()
        cy.get('#confirmbtn').click()

        // window:alert

        cy.on('window:alert', (str)=>{

            // how to compare two str
            expect(str).to.equal('Hello , share this practice page and share 
your knowledge')
        })
        // window:confirm
        cy.on('window:confirm', (str)=>{

            // how to compare two str
            expect(str).to.equal('Hello , Are you sure you want to confirm?')
        })


    })
})
```

Navigate to another domain

```
// To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My six Test', function () {

    it('My Six test case', function() {
        // invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

        // Because every child url have target attribute in they html
        cy.get('#opentab').invoke('removeAttr','target').click()

        /*
        to further automate things on the next page do follow the thing below:
        cy.origin("<next page url>", ()=>
        {
            //code
        })
            */


        cy.origin("https://www.qaclickacademy.com", ()=>
        {
            cy.get("#navbarSupportedContent a[href*='about']").click()
            cy.get(".mt-50 h2").should('contain','QAClick Academy')
        })


    })
})
```


practice-cypress

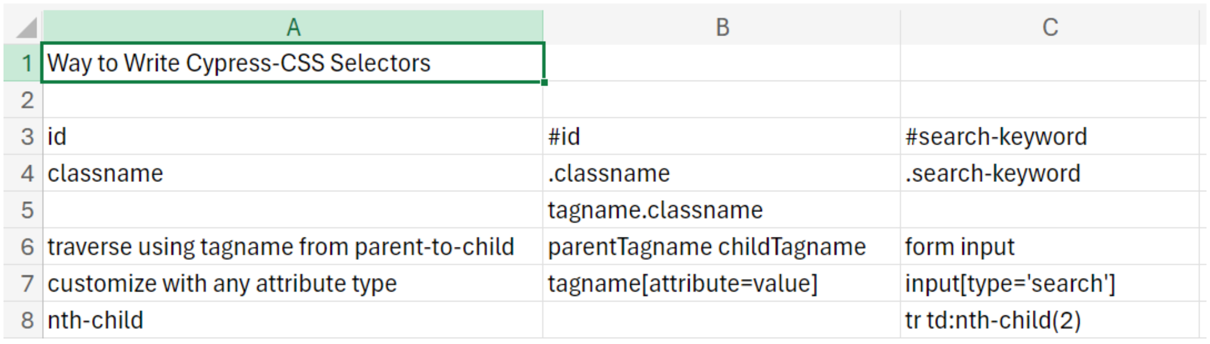
Test10_childwindow.js 00:09

My Six test case

TEST BODY

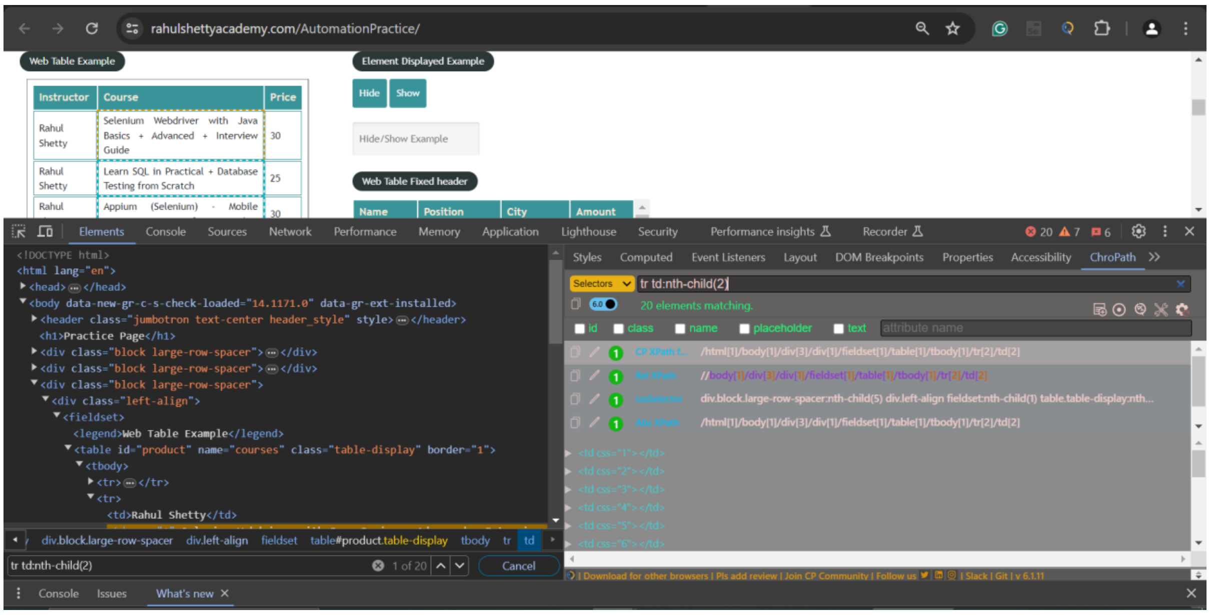
```
visit https://rahulshettyacademy.com/AutomationPractice/
(xhr) GET 200 https://www.rahulshettyacademy.com/api/course
get #opentab
invoke .removeAttr()
-click
(page load) --page loaded--
(new url) https://www.qaclickacademy.com/
origin https://www.qaclickacademy.com
  get #navbarSupportedContent a[href*='about']
  -click
  (page load) --page loaded--
  get .mt-50 h2
  -assert expected <h2> to contain QAClick Academy
  (new url) https://www.qaclickacademy.com/about.html
```

File Edit View Window Help Developer Tools
Specs
https://www.qaclickacademy.com/a...
Electron 118
1000x660 (63%)
Siri Balaji Residency, 2 GaddiAnnaram, Hyderabad.
info@qaclickacademy.com
Opening Hours : Monday to Saturay - 8 Am to 5 Pm
Need Help? email us
info@qaclickacademy.com
Access all our Courses
HOME COURSES BLOG ABOUT US CONTACT
About Us
Home / About Us



	A	B	C
1	Way to Write Cypress-CSS Selectors		
2			
3	id	#id	#search-keyword
4	classname	.classname	.search-keyword
5		tagname.classname	
6	traverse using tagname from parent-to-child	parentTagname childTagname	form input
7	customize with any attribute type	tagname[attribute=value]	input[type='search']
8	nth-child		tr td:nth-child(2)

Tables:


rahulshettyacademy.com/AutomationPractice/

```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body data-new-gr-c-s-check-loaded="14.1171.0" data-gr-ext-installed>
  ▶<header class="jumbotron text-center header_style" style>…</header>
   <h1>Practice Page</h1>
  ▶<div class="block large-row-spacer">…</div>
  ▶<div class="block large-row-spacer">…</div>
  ▼<div class="block large-row-spacer">
    ▼<div class="left-align">
      ▼<fieldset>
         <legend>Web Table Example</legend>
        ▼<table id="product" name="courses" class="table-display" border="1">
          ▼<tbody>
            ▶<tr>…</tr>
            ▼<tr>
               <td>Rahul Shetty</td>
```

div.block.large-row-spacer div.left-align fieldset table#product.table-display tbody tr td

tr td:nth-child(2) 1 of 20 Cancel

Selectors: tr td:nth-child(2)

20 elements matching

/html[1]/body[1]/div[3]/div[1]/fieldset[1]/table[1]/tbody[1]/tr[2]/td[2]

//body[1]/div[3]/div[1]/fieldset[1]/table[1]/tbody[1]/tr[2]/td[2]

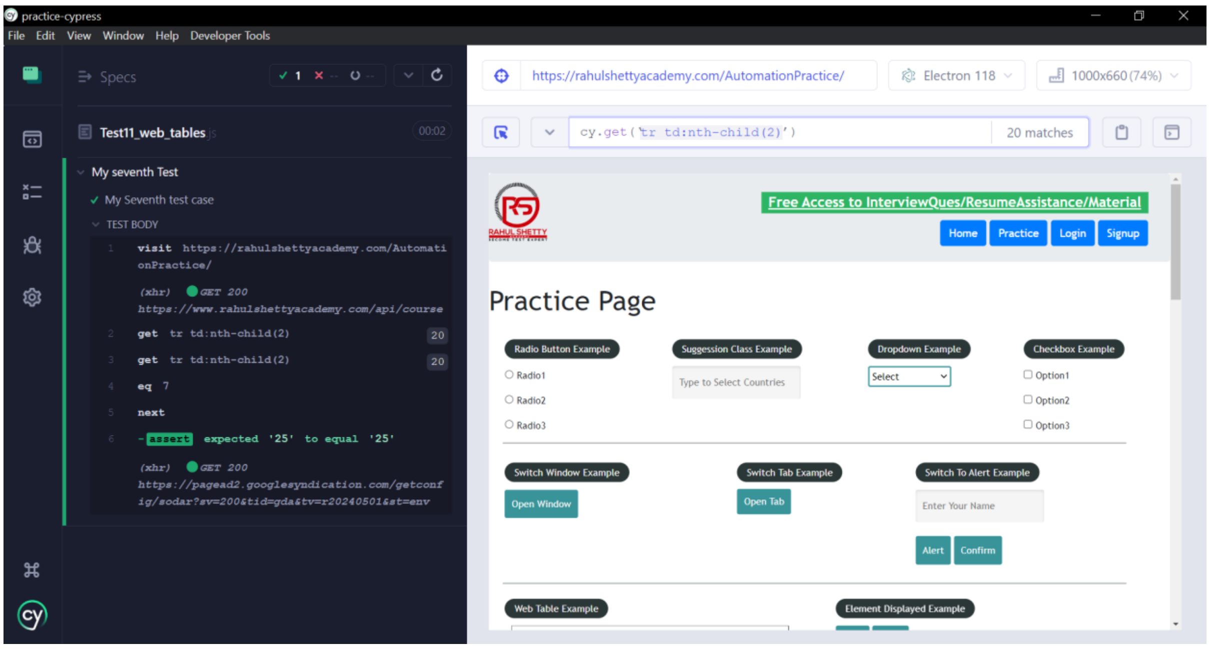
div.block.large-row-spacer:nth-child(5) div.left-align fieldset:nth-child(1) table.table-display:nth...

/html[1]/body[1]/div[3]/div[1]/fieldset[1]/table[1]/tbody[1]/tr[2]/td[2]

Web Table Example
Instructor
Course
Price
Rahul Shetty
Selenium Webdriver with Java Basics + Advanced + Interview Guide
30
Rahul Shetty
Learn SQL in Practical + Database Testing from Scratch
25
Rahul
Appium (Selenium) - Mobile
30
Element Displayed Example
Hide
Show
Hide/Show Example
Web Table Fixed header
Name
Position
City
Amount
Elements Console Sources Network Performance Memory Application Lighthouse Security Performance insights Recorder
Styles Computed Event Listeners Layout DOM Breakpoints Properties Accessibility ChroPath
id class name placeholder text attribute name
Console Issues What's new

```
// To get auto-suggestion add line no. 2
/// <reference types="Cypress" />
```

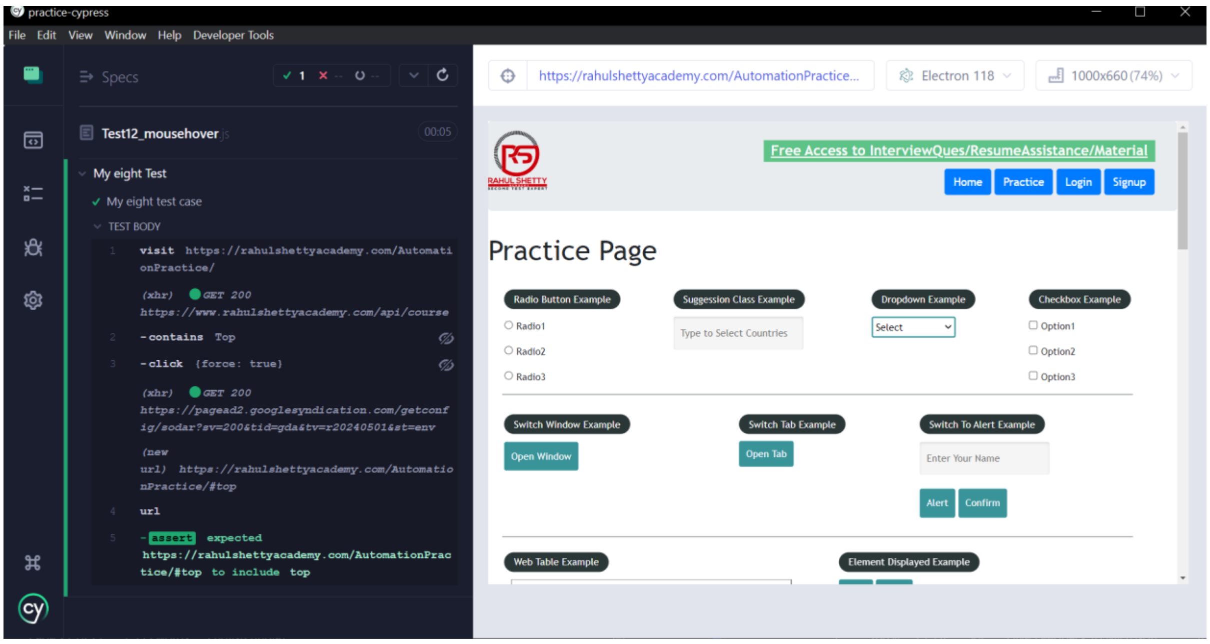
```
describe('My seventh Test', function () {

    it('My Seventh test case', function() {
        // invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

        //get the specific column data

        cy.get('tr td:nth-child(2)').each(($e1, index, $list) => {

            const text = $e1.text()
            if(text.includes('Python'))
            {
                // how to find next sibling using method `.next()`
                cy.get('tr td:nth-
child(2)').eq(index).next().then(function(price)
                {
                    const priceText= price.text()
                    expect(priceText).to.equal('25')
                })
            }

        })
    })
})
```



```
visit https://rahulshettyacademy.com/AutomationPractice/
(xhr) GET 200 https://www.rahulshettyacademy.com/api/course
get tr td:nth-child(2)        20
get tr td:nth-child(2)        20
eq 7
next
-assert expected '25' to equal '25'
(xhr) GET 200 https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gda&tv=r20240501&st=env
```

Mouse-Hover

```
// To get auto-suggestion add line no. 2
/// <reference types="Cypress" />

describe('My eight Test', function () {

    it('My eight test case', function() {
        // invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

        /*
        //give the parent near element css
        cy.get('div.mouse-hover-content').invoke('show')
        cy.contains('Top').click()
        cy.url().should('include', 'top')
        */

        // forcefully clickinh hidden pop-ups
        cy.contains('Top').click({force:true})
        cy.url().should('include', 'top')
    })
})
```

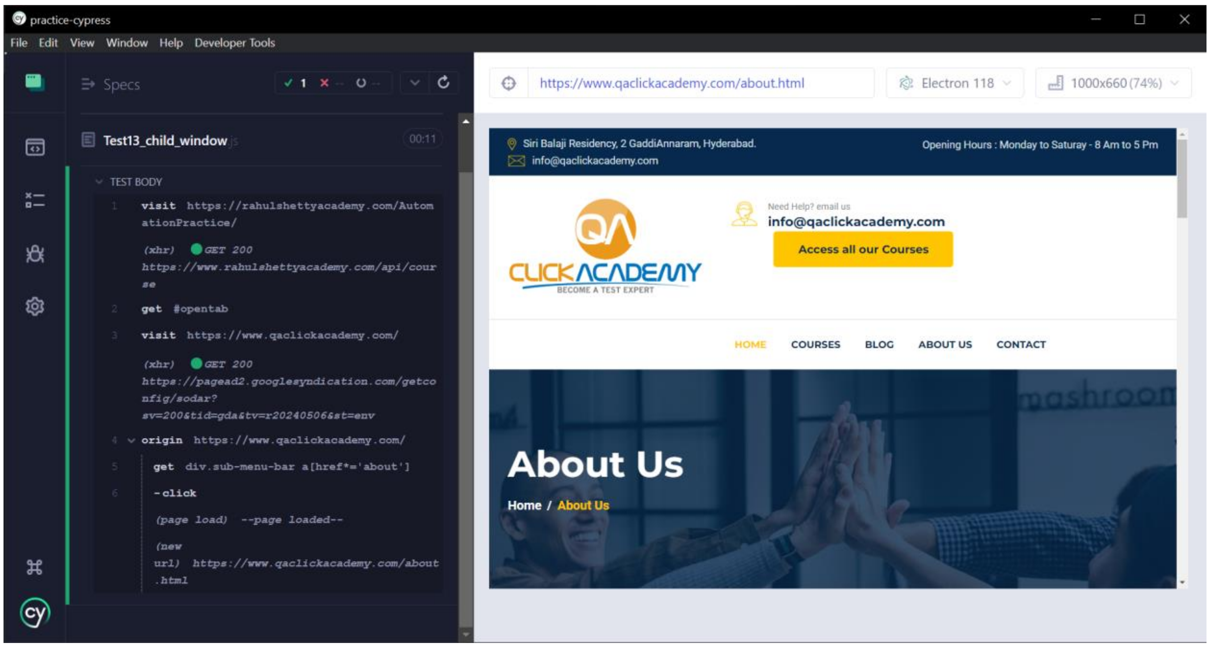


```
Test12_mousehover.js   00:05
My eight Test
✓ My eight test case
TEST BODY
visit https://rahulshettyacademy.com/AutomationPractice/
(xhr) ● GET 200 https://www.rahulshettyacademy.com/api/course
-contains Top
-click {force: true}
(xhr) ● GET 200 https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gda&tv=r20240501&st=env
(new url) https://rahulshettyacademy.com/AutomationPractice/#top
url
-assert expected https://rahulshettyacademy.com/AutomationPractice/#top to include top
```

Child Window:

```
// To get auto-suggestion add line no. 2
/// <reference types="Cypress" />
```

```
describe('My eight Test', function () {

    it('My eight test case', function() {
        // invoke anything in cypress using `.cy` its like a driver
        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")

        /*
        directly hit the url without clicking on any button
        or user href tag to visit that url
        1. To click url
        2. To visit href link
        */

        /*
        open window

        Here we are concatenating cypress method with prop method prop is not
a cypress method, so first we are resolving this promise
        */

        cy.get('#opentab').then(function(e1){
            const url =e1.prop('href')
            //using .visit can redirect you to the particular url, you cant
perform automation
            cy.visit(url)

            //inside func only: new url page can be automated
            cy.origin(url, ()=>
        {
            cy.get("div.sub-menu-bar a[href*='about']").click()
        })
        })

    })
})
```



Test13_child_window.js 00:11

TEST BODY

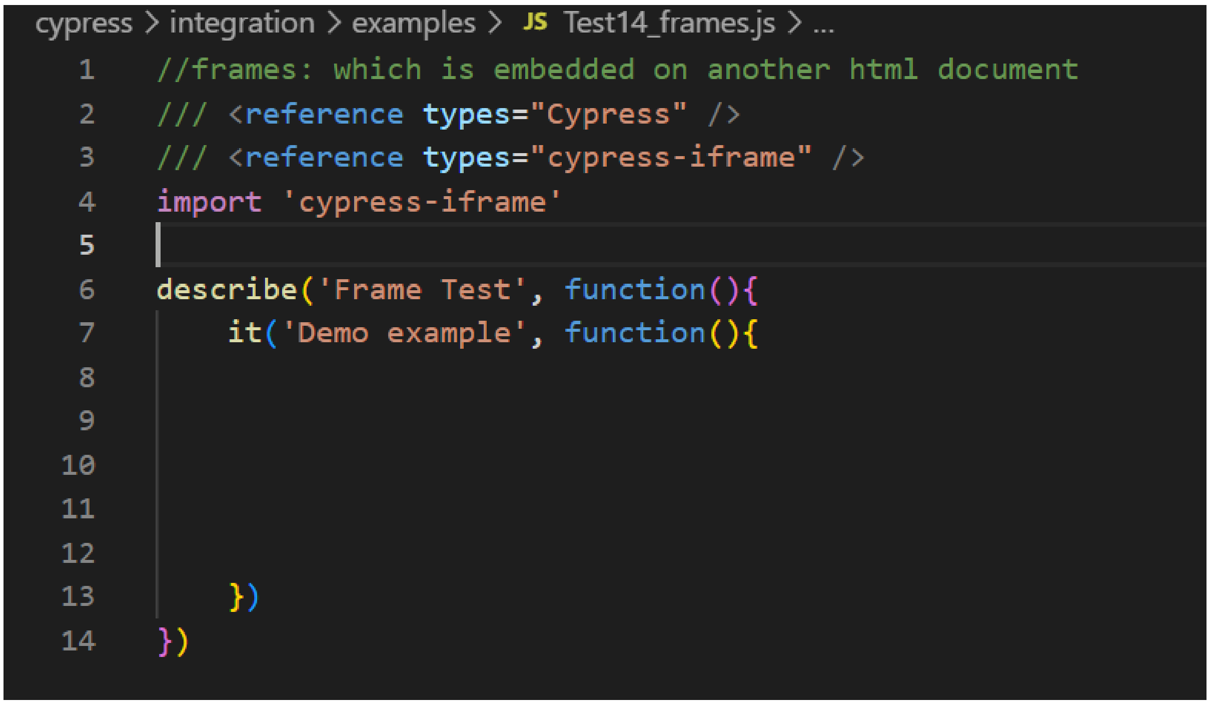
```
1  visit  https://rahulshettyacademy.com/AutomationPractice/
   (xhr)  ● GET 200 https://www.rahulshettyacademy.com/api/course
2  get  #opentab
3  visit  https://www.qaclickacademy.com/
   (xhr)  ● GET 200 https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gda&tv=r20240506&st=env
4  origin  https://www.qaclickacademy.com/
5    get  div.sub-menu-bar a[href*='about']
6    -click
     (page load)  --page loaded--
     (new url)  https://www.qaclickacademy.com/about.html
```

https://www.qaclickacademy.com/about.html

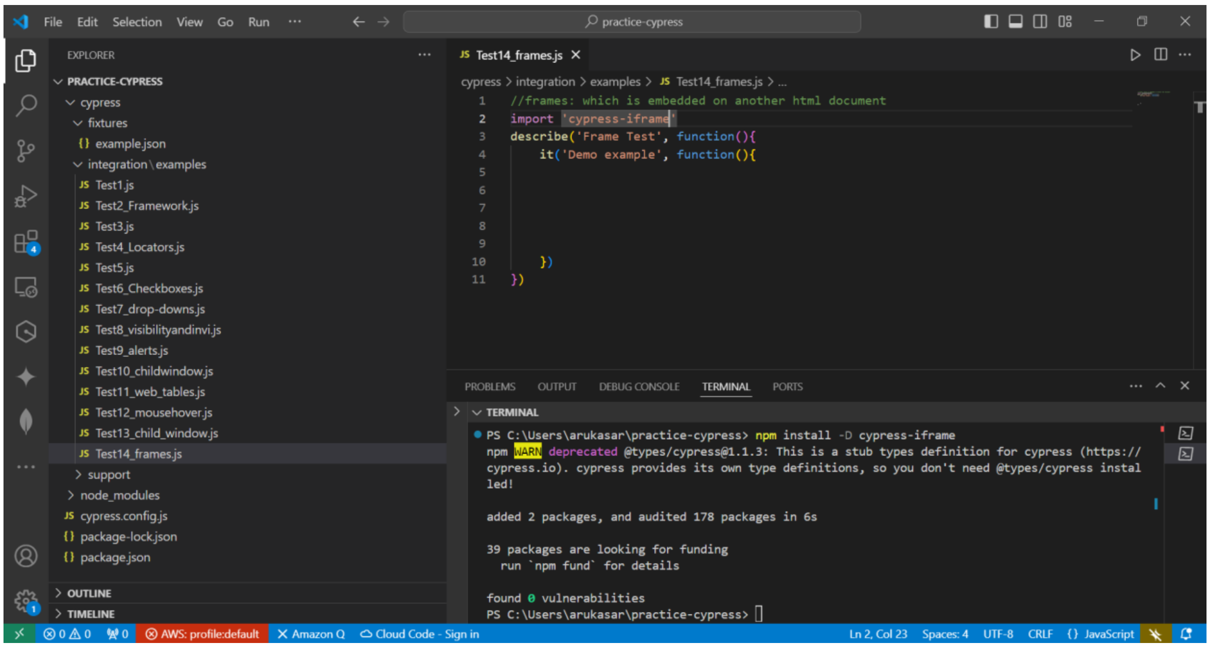
Frame

Command to install iframe: `npm install -D cypress-iframe`

Note: Do not forget to import


cypress > integration > examples > JS Test14_frames.js > ...

```javascript
//frames: which is embedded on another html document
/// <reference types="Cypress" />
/// <reference types="cypress-iframe" />
import 'cypress-iframe'

describe('Frame Test', function(){
    it('Demo example', function(){





    })
})
```



```
//frames: which is embedded on another html document
import 'cypress-iframe'
describe('Frame Test', function(){
    it('Demo example', function(){






    })
})
```

```
PS C:\Users\arukasar\practice-cypress> npm install -D cypress-iframe
npm WARN deprecated @types/cypress@1.1.3: This is a stub types definition for cypress (https://
cypress.io). cypress provides its own type definitions, so you don't need @types/cypress instal
led!

added 2 packages, and audited 178 packages in 6s

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

found 0 vulnerabilities
PS C:\Users\arukasar\practice-cypress> 
```

```
//frames: which is embedded on another html document
/// <reference types="Cypress" />
/// <reference types="cypress-iframe" />
import 'cypress-iframe'

describe('Frame Test', function(){
    it('Demo example', function(){

        cy.visit('https://rahulshettyacademy.com/AutomationPractice/')
        cy.frameLoaded('#courses-iframe')
        cy.iframe().find("a[href*='mentorship']").eq(0).click()

        // cy.iframe().find("h1[class*='pricing-
title']").should('have.length',2)







    })
})
```