import { text } from './Login.cy'

describe(' Access value from other cypress test', () => {

    it('Access type text from different Cypress test', () => {

        cy.visit(

            "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

        );

        cy.get('[id="input-email"]').type(text)

    })

})

describe("Performing action if element is not present-  using Cypress", () => {

  it("Open website and enter username, password", () => {

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

    cy.get(".form-group").then(($ele) => {

      if ($ele.find("#input-password1").length) {

        cy.log("-- input-passowrd1 element not present --- ");

      } else {

        cy.log("--  element is present --- ");

        cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com");

        cy.get('[id="input-password"]').type("Cypress123!!");

        cy.get('[type="submit"]').eq(0).click();

      }

    });

  });

  it("Open website and enter username, password", () => {

    cy.visit(

      "https://www.lambdatest.com/selenium-playground/bootstrap-modal-demo"

    );

    cy.get('[data-target="#myModal"]').click();

    cy.get("body").then(($popup) => {

      if ($popup.find(".ltkpopup-standard").length) {

        cy.log("--- element not present --- ");

      } else {

        cy.log("--  element is present --- ");

      }

    });

  });

});

describe('Test Alert Box', () => {

    it('Test Alert Box', () => {

        cy.visit('https://demoqa.com/alerts');

        cy.get('button#alertButton').click();

    })

})

describe("API Test using plugin cypress-plugin-api", () => {

  it("test api - GET ", () => {

    cy.api("GET", "https://reqres.in/api/users?page=2").should((response) => {

      expect(response.status).to.eq(200);

    });

  });

  it("POST API testing Using Cypress API Plugin", () => {

    cy.api("POST", "https://reqres.in/api/users", {

      name: "Anshita",

      job: "QA leader",

    }).should((response) => {

      expect(response.status).to.eq(201);

    });

  });

  it("DELETE API testing Using Cypress API Plugin", () => {

    cy.api("DELETE", "https://reqres.in/api/users/2").should((response) => {

      expect(response.status).to.eq(204);

    });

  });

});

describe("Assertion Use Case2 - If Else Condition", () => {

  beforeEach(() => {

    cy.visit(

      "https://naveenautomationlabs.com/opencart/index.php?route=account/login"

    );

  });

  it("Condition 1- Click If the element exist on the page using length >0", () => {

    cy.get('[id="input-email"]').type("cypress.test1@disposable.com");

    cy.get('[id="input-password"]').type("Cypress123!!");

    cy.get(

      "[action='https://naveenautomationlabs.com/opencart/index.php?route=account/login']"

    ).then(($value) => {

      if ($value.find(".btn.btn-primary").length > 0) {

        cy.log(" Button Exist ");

        $value.click();

      } else {

        cy.log(" Button does not exist");

      }

    });

  });

  it("Condition 2- Click If the element does not exist on the page => passing incorrect element inside find()", () => {

    cy.get('[id="input-email"]').type("cypress.test1@disposable.com");

    cy.get('[id="input-password"]').type("Cypress123!!");

    cy.get(

      "[action='https://naveenautomationlabs.com/opencart/index.php?route=account/login']"

    ).then(($value) => {

      if ($value.find(".btn.btn-primary1").length > 0) {

        cy.log(" Button Exist ");

        $value.click();

      } else {

        cy.log(" Button does not exist");

      }

    });

  });

  it("Condition 3 - If there are more than 1 elements on the page ", () => {

    cy.get(".dropdown-toggle").then(($value) => {

      if ($value.length > 0) {

        cy.log(" Inside If condition ");

        $value.first().click();

      } else {

        cy.log(" Inside else condition ");

      }

    });

  });

  it("Condition 4 -> Used length==0 where it checks if no element exist otherwise it will go to else condition", () => {

    cy.get(".dropdown-toggle").then(($value) => {

      if ($value.length == 0) {

        cy.log(" Inside If condition ");

        cy.log(" Element does not exist ");

      } else {

        cy.log(" Inside else condition ");

        cy.log(" ELement exist");

        $value.first().click();

      }

    });

  });

  it("Condition 5- Using .length() directly where it returns 0 ", () => {

    if (cy.get(".dropdown-toggle").length) {

      cy.log(" Inside If condition ");

      cy.log(" Element does not exist ");

    } else {

      cy.log(" Inside else condition ");

      cy.log(" ELement exist");

    }

  });

});

describe('Verify the background color', () => {

    it('Check background color', () => {

        cy.visit('https://www.amazon.ae/')

        cy.get('#nav-xshop').should('have.css', 'background-color', 'rgba(0, 0, 0, 0)')

    })

})

describe('Test API in Cypress', () => {

    it('Testing Bad Request', () => {

        cy.request({

            method: 'GET',

            url: 'https://reqres.in/api/unknown/23',

            failOnStatusCode: false

        }).then((response) => {

            expect(response.status).to.be.eq(404)

        })

    })

})

describe("Login to a Website with Basic Authentication", () => {

    it("1st Approach - should log in using basic authentication", () => {

        cy.visit('https://authenticationtest.com/HTTPAuth/', {

            auth: {

                username: Cypress.env('username'),

                password: Cypress.env('password')

            }

        })

        cy.contains('Login Success').should('be.visible')

    });

    it("2nd Approach - should log in using basic authentication", () => {

        cy.visit('https://user:pass@authenticationtest.com/HTTPAuth/')

        cy.contains('Login Success').should('be.visible')

    });

    it('3rd Approach - should login using basic authentication', () => {

        cy.loginBasicAuth()

        cy.contains('Login Success').should('be.visible')

    })

});

For above creates custom command

Cypress.Commands.add('loginBasicAuth', () => {

    cy.visit('https://authenticationtest.com/HTTPAuth/', {

        auth: {

            username: 'user',

            password: 'pass'

        }

    })

})

describe("Blur an element", () => {

    it(".blur() - blur off a DOM element", () => {

        cy.visit(

            "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

        );

        cy.get("#input-firstname").type("Test AB").blur();

        cy.get("#input-lastname").type("Test").focus();

    });

})

describe('Find broken links on the page', () => {

    it('verify navigation across the pages', () => {

        cy.visit('https://ecommerce-playground.lambdatest.io/index.php?route=common/home')

        let brokenLinks = 0

        let activeLinks = 0

        cy.get('a').each(($link, index) => {

            const href = $link.attr('href')

            if (href) {

                cy.request({ url: href, failOnStatusCode: false }).then((response) => {

                    if (response.status >= 400) {

                        cy.log(`\*\*\*\* link  ${index + 1} is Broken Link \*\*\* ${href} `)

                        brokenLinks++

                    }

                    else {

                        cy.log(`\*\*\* link  ${index + 1} is Active Link \*\*\*`)

                        activeLinks++

                    }

                })

            }

        }).then(($links) => {

            const totalLinks = $links.length

            cy.log(`\*\*\*\* total links \*\*\*\* ${totalLinks}`)

            cy.log(`\*\*\*\* broken links \*\*\*\* ${brokenLinks}`)

            cy.log(`\*\*\*\* active links \*\*\*\* ${activeLinks}`)

        })

    })

});

describe("Handle checkbox in Cypress", () => {

    beforeEach(() => {

        cy.visit(

            "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

        );

    })

    it(" Checbox Handling: check and uncheck ", () => {

        cy.get("label input[type='radio']").check("1", { force: true });

        cy.get("[type='checkbox']").check();

        cy.get("[type='checkbox']").uncheck();

    });

})

describe(" Assertion Use Case 1- If Else Condition", () => {

  before("Open website and enter username, password", () => {

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

  });

  it("Login into the application using credentials", () => {

    cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com");

    cy.get('[id="input-password"]').type("Cypress123!!");

    cy.get('[type="submit"]').eq(0).click();

    cy.get('[name="search"]').eq(0).type("Macbook");

    cy.get('[type="submit"]').eq(0).click();

    cy.contains("Macbook");

    cy.get(".price-new").then(($value) => {

      if ($value.length > 0) {

        cy.log(" if condition");

      } else {

        cy.log("else condition");

      }

    });

  });

});

describe("Cypress useful commands", () => {

    it("DropDown Example", () => {

        cy.visit(

            "https://chercher.tech/practice/practice-dropdowns-selenium-webdriver"

        );

        // cy.get("select#first").select("Yahoo");

        cy.get("select#animals").select("Avatar");

        cy.get("select#second").select(["Donut", "Bonda"]);

        cy.get("select#first").children("option").eq(2);

        cy.get("select#second").children("option").should("contain", "Donut", "Bonda")

    });

});

describe('Check current date', () => {

    it("Today's Date", () => {

        cy.log("date", Date.now())

        const date = new Date();

        cy.log("format date", date.getDate() + '/' + (date.getMonth() + 1) + '/' + date.getFullYear() + '-' + date.getHours() + ':' + date.getMinutes())

    })

})

describe("Hello Cypress Studio", () => {

  /\* ==== Test Created with Cypress Studio ==== \*/

  it('recorder\_test', function () {

    /\* ==== Generated with Cypress Studio ==== \*/

    cy.visit('https://bstackdemo.com/');

    cy.get('#\\31  > .shelf-item\_\_buy-btn').click();

    /\* ==== End Cypress Studio ==== \*/

    /\* ==== Generated with Cypress Studio ==== \*/

    cy.get('.buy-btn').click();

    cy.get('#username > .css-yk16xz-control > .css-1hwfws3 > .css-1wa3eu0-placeholder').click();

    cy.get('#react-select-2-option-0-0').click();

    cy.get('.css-1wa3eu0-placeholder').click();

    cy.get('#react-select-3-option-0-0').click();

    /\* ==== End Cypress Studio ==== \*/

  });

});

describe("Cypress useful commands", () => {

  it("open cart registration form", () => {

    cy.visit(

      "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

    );

    cy.get("#input-firstname").type("Test AB");

    cy.get("#input-lastname").type("AB", { delay: 1000 });

    cy.get("#input-email").type("Test.AB@disposable.com");

    cy.get("#input-telephone").type("234567890");

    cy.get("#input-password").type("testA12!!");

    cy.get("#input-confirm").type("testA12!!");

    cy.get("input[type='submit']").click();

    cy.get("label input[type='radio']").check("1", { force: true });

    cy.get("[type='checkbox']").check();

  });

  it("DropDown Example", () => {

    cy.visit(

      "https://chercher.tech/practice/practice-dropdowns-selenium-webdriver"

    );

    cy.get("select#first").select("Yahoo");

    cy.get("select#animals").select("Avatar");

    cy.get("select#second").select(["Donut", "Bonda"]);

    cy.get("select#first").select("Google", { force: true });

  });

});

it("API Test", () => {

  cy.request("GET", "https://api.coinbase.com/v2/currencies").then(

    (response) => {

      console.log("response is == ", response);

      console.log(response.status);

      assert(response.status, "200");

    }

  );

});

it("Cypress Command - Task", () => {

  cy.task("log", "hello World");

  /\*\*need to create task in nodesetupEvents function

   \* on("task", {

        log(message) {

          console.log(message);

          return null;

        },

        readDataFromExcel: (data) => {

          return readDataFromExcel(data);

        },

      }); \*/

});

it.only("Cypress Command - ScrollTo", () => {

  cy.visit(

    "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

  );

  cy.scrollTo(0, 500);

  cy.scrollTo("topRight");

  cy.scrollTo("bottom");

});

it(".focus() - focus on a DOM element", () => {

  cy.visit(

    "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

  );

  cy.get("[placeholder='Search']").type("macbook").focus();

});

it(".blur() - blur off a DOM element", () => {

  cy.visit(

    "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

  );

  cy.get("[placeholder='Search']").type("macbook").blur();

});

describe("Cypress Exception Handling", () => {

  it("Fail on status code by opening url in browser", () => {

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login/1",

      { failOnStatusCode: false }

    );

  });

  it("Fail on status code by calling api", () => {

    cy.request({

      url: "https://ecommerce-playground.lambdatest.io/index.php?route=account/login/1",

      failOnStatusCode: false,

    });

  });

  it("Test Failure when trying to find incorrect locator- error Message", () => {

    cy.on("fail", (err, runnable) => {

      console.log(err.message);

      return false;

    });

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

    cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com");

    cy.get('[id="input-password"]').type("Cypress1234!!");

    cy.get('[value="Login"]').click();

    cy.get(".error-message").should("be.visible");

    cy.log("test");

    cy.get('[name="search"]').eq(0).type("Macbook");

  });

  it("Test Failure when trying to find incorrect locator - Password", () => {

    cy.on("fail", (err, runnable) => {

      console.log(err.message);

      return false;

    });

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

    cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com");

    cy.get('[id="input-password1"]').type("Cypress1234!!");

  });

})

Fixture file .

{

  "name": {

    "first\_name": "AB",

    "last\_name": "Test"

  },

  "email": "AB12.Test@dispostable.com",

  "phone": "1234567890",

  "login": {

    "password": "Cypress123!!"

  }

}

describe('Registration Flow Using Fixture Data', () => {

    it('Enter the form details', () => {

        cy.visit('https://naveenautomationlabs.com/opencart/index.php?route=account/register')

        cy.fixture('register').then((data) => {

            cy.get('#input-firstname').type(data.name.first\_name)

            cy.get('#input-lastname').type(data.name.last\_name)

            cy.get('#input-email').type(data.email)

            cy.get('#input-telephone').type(data.phone)

            cy.get('#input-password').type(data.login.password)

            cy.get('#input-confirm').type(data.login.password)

        })

        cy.get('input[type="radio"]').eq(1).check()

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

        cy.get('.btn.btn-primary').click()

        cy.contains('Your Account Has Been Created!').should('exist')

    })

})

describe('Change browser in the test case', () => {

    beforeEach(() => {

        cy.visit("https://www.docker.com/")

    })

    it("Browser is chrome", { browser: "chrome" }, () => {

        cy.log("Chrome");

    })

    it("Browser is webkit", { browser: "Electron" }, () => {

        cy.log("Webkit");

    })

    it("Browser is Electron", { browser: "Webkit" }, () => {

        cy.log("Electron");

    })

})

describe(" Delay the text entering in Cypress", () => {

  beforeEach(() => {

    cy.visit("https://wordcounter.net/");

  })

  it("Enter username and password with delay of 0 milliseconds", () => {

    cy.get("textarea#box").type(

      "Cypress is a next generation front end testing tool built for the modern web. We address the key pain points developers and QA engineers face when testing modern applications.",

      { delay: 0 }

    );

  });

  it("Enter username and password with no delay of 0 milliseconds", () => {

    cy.get("textarea#box").type(

      "Cypress is a next generation front end testing tool built for the modern web. We address the key pain points developers and QA engineers face when testing modern applications.'"

    );

  });

});

describe(' Drag and Drop', () => {

    const capitals = ["Stockholm", "Oslo", "Seoul", "Rome", "Madrid", "Copenhagen", "Washington"];

    const countries = ["Sweden", "Norway", "South Korea", "Italy", "Spain", "Denmark", "United States"];

    beforeEach(() => {

        cy.visit('http://dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html')

    })

    function moveCapitalToCountry(dragElement, dropElement) {

        cy.get('div[id^=box]:contains(' + dragElement + ')').trigger('mousedown', { which: 1 })

        cy.wait(100)

        cy.get('div[id^=box]:contains(' + dropElement + ')').trigger('mousemove').trigger('mouseup', { force: true })

    }

    function moveBackCapital(dragElement) {

        cy.get('div[id^=box].dragableBox:contains(' + dragElement + ')').trigger('mousedown', { which: 1 })

        cy.wait(100)

        cy.get('div#dropContent').invoke('css', 'height', '200px').trigger('mousemove').trigger('mouseup', { force: true })

    }

    function verifyColor(dragElement) {

        cy.get('div[id^=box].dragableBox:contains(' + dragElement + ')').should('be.visible')

            .should('have.css', 'background-color', 'rgb(0, 255, 0)')

    }

    it('TC1 || Positive Use Case|| Drag all the capitals to the respective country section', () => {

        countries.forEach((country, index) => {

            const capital = capitals[index];

            moveCapitalToCountry(capital, country);

        });

        capitals.forEach(capital => {

            verifyColor(capital)

            moveBackCapital(capital);

        })

    })

    it('TC2 || Negative Use Case|| Drag all the capitals to the wrong countries', () => {

        moveCapitalToCountry("Stockholm", "Denmark")

        moveCapitalToCountry("Rome", "Norway")

        moveCapitalToCountry("Oslo", "Italy")

        moveCapitalToCountry("Copenhagen", "United States")

        moveCapitalToCountry("Madrid", "South Korea")

        moveCapitalToCountry("Washington", "Spain")

        moveCapitalToCountry("Seoul", "Sweden")

        capitals.forEach(capital => {

            moveBackCapital(capital);

        })

    })

    it('TC3 || Combination of Positive and Negative Use Case', () => {

        moveCapitalToCountry("Stockholm", "Sweden")

        moveCapitalToCountry("Rome", "Italy")

        moveCapitalToCountry("Oslo", "Norway")

        moveCapitalToCountry("Copenhagen", "United States")

        moveCapitalToCountry("Madrid", "Spain")

        moveCapitalToCountry("Washington", "South Korea")

        moveCapitalToCountry("Seoul", "Denmark")

        capitals.forEach(capital => {

            moveBackCapital(capital);

        })

    })

})

describe("Cypress useful commands", () => {

  it("DropDown Example", () => {

    cy.visit(

      "https://chercher.tech/practice/practice-dropdowns-selenium-webdriver"

    );

    // cy.get("select#first").select("Yahoo");

    cy.get("select#animals").select("Avatar");

    cy.get("select#second").select(["Donut", "Bonda"]);

    cy.get("select#first").children("option").eq(2);

    cy.get("select#second").children("option").should("contain", "Donut", "Bonda")

  });

});

describe('Test for multiple urls', () => {

    it('Fetch url and pass it to the next step', () => {

        cy.visit('https://naveenautomationlabs.com/opencart/')

            .get('[title="MacBook"]').click()

            .url().then((urlvalue) => {

                cy.writeFile('cypress/fixtures/url.json', { url: urlvalue });

            });

    });

    })

describe(" Read csv file using cy.readFile", () => {

    it('Read csv', () => {

        cy.readFile('cypress/fixtures/test.csv').then((data) => {

            cy.log("data === " + data + " rows" + data[0][1])

        })

    })

})

//Ref : https://docs.cypress.io/api/commands/selectfile

describe("Upload File Request", () => {

  it("TC1 - No Input Field || upload File by changing action", () => {

    cy.visit("https://www.ilovepdf.com/pdf\_to\_word");

    cy.get("a#pickfiles").selectFile("test.pdf", { action: "drag-drop" });

  });

  it.skip("TC2- Error Case- No Input Field|| upload File without changing action", () => {

    cy.visit("https://www.ilovepdf.com/pdf\_to\_word");

    cy.contains("Select PDF file").selectFile("test.pdf");

  });

  it.skip("TC3- Upload File with input field", () => {

    cy.visit("https://demoqa.com/upload-download");

    cy.get("input#uploadFile").selectFile("test.pdf");

  });

});

describe("Different approaches to find the length of elements on web page", () => {

  let length, totalCount, count;

  beforeEach(() => {

    cy.visit("https://www.tradeling.com/ae-en/catalog/audio-studio");

  });

  it("When expected length is known", () => {

    cy.get("dl.chakra-stack.css-ioc31s")

      .find('>dd:contains("United Arab Emirates")')

      .should("have.length", 40);

  });

  it("When expected length is not known", () => {

    cy.get("dl.chakra-stack.css-ioc31s")

      .find('>dd:contains("United Arab Emirates")')

      .then(($value) => {

        length = $value.length;

      });

  });

  it("Printing length - Approach1 ", () => {

    cy.log("\*\*\* length obtained using approach1 is \*\*\* " + length);

  });

  it("Using Cypress Method", () => {

    cy.get("dl.chakra-stack.css-ioc31s")

      .find('>dd:contains("United Arab Emirates")')

      .then((value) => {

        totalCount = Cypress.$(value).length;

        expect(value).to.have.length(totalCount);

      });

  });

  it("Printing length - Approach2", () => {

    cy.log("\*\*\* length obtained using approach2 is \*\*\* " + totalCount);

  });

  it("Finding Length - Approach3 Cypress Method", () => {

    count = cy

      .get("dl.chakra-stack.css-ioc31s")

      .find('>dd:contains("United Arab Emirates")')

      .its("length")

      .should("to.be", 40);

  });

});

let getText;

describe("find text of an element", () => {

    it("Alias First Test - Using Invoke Command ", function () {

        cy.visit("https://www.linkedin.com/")

        cy.get('[data-test-id="hero\_\_headline"]').invoke('text').as('textFunction');

    });

    it("Print Value - ALias => Invoke - Command ", function () {

        cy.log("===== Print Value Using Invoke Command ==== ", this.textFunction)

    })

    it("Alias Second Test - Using Wrap Command", () => {

        cy.visit("https://www.linkedin.com/", {

            failOnStatusCode: false

        })

        cy.get('[data-test-id="hero\_\_headline"]').then($value => {

            const textValue = $value.text()

            cy.wrap(textValue).as('wrapValue')

        })

    })

    it("Print Value - ALias => Wrap - Command ", function () {

        cy.log("===== Print Value Uisng Wrap Command ==== ", this.wrapValue)

    })

    it("Third Test - using variable ", () => {

        cy.visit("https://www.linkedin.com/", { failOnStatusCode: false })

        cy.get('[data-test-id="hero\_\_headline"]').then(($value) => {

            getText = $value.text()

        })

    })

    it("Print Value - Uisng Variables ", function () {

        cy.log("Print Value - Using Variables ====== ", getText)

    })

})

describe(' Test Flaky Test', () => {

    it(' Test flake features', () => {

        cy.visit('/flkyTest.html')

        cy.get('#dynamic-content').should('have.text', 'Hello, World!');

    })

})

describe("Automation using Cypress", () => {

  it("Open website and enter username, password", () => {

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

    cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com");

    cy.get('[id="input-password"]').type("Cypress123!!");

    cy.get('[type="submit"]').eq(0).click();

    cy.get('[name="search"]').eq(0).type("Macbook");

    cy.get('[type="submit"]').eq(0).click();

    cy.contains("Macbook");

  });

});

describe('Find case sensitive search on the page', () => {

    it('verify exact match case in Cypress', () => {

        cy.visit('https://ecommerce-playground.lambdatest.io/index.php?route=common/home')

        cy.contains('ADDONS', { matchCase: false }).click()

    })

});

describe('Disable Image', () => {

    it('disable image load in Cypress', () => {

        // Based on the below config, you can disable image on page load in cypress

        /\* Pass this in Cypress.config.js

        on('before:browser:launch', (browser = {}, launchOptions) => {

        console.log(launchOptions.args)

        if (browser.family === 'chromium' && browser.name !== 'electron') {

        //launchOptions.args.push('--blink-settings=imagesEnabled=false')

        return launchOptions

        }

\*/

        cy.visit('https://www.amazon.com')

    })

})

import {

  createHorde,

  gremlins,

  species,

  strategies,

  mogwais,

  randomizer,

} from "gremlins.js";

const seed = "formFiller";

describe("Form filler", () => {

  let horde;

  beforeEach(() => {

    return cy.visit("https://classic.freecrm.com/register/").then(() =>

      cy.window().then((pageWindow) => {

        horde = createHorde({

          window: pageWindow,

          species: [

            species.clicker(),

            species.toucher(),

            species.formFiller(),

            species.scroller(),

            species.typer(),

          ],

          mogwais: [mogwais.alert(), mogwais.fps(), mogwais.gizmo()],

          strategies: [

            strategies.distribution(),

            strategies.allTogether({ nb: 10000 }),

            strategies.bySpecies({ nb: 10 }),

          ],

          randomizer: new Chance(seed),

        });

      })

    );

  });

  it("should fill text input with seeded value", () => {

    return cy.wrap(horde.unleash());

  });

});

describe(" Mouse over", () => {

  /\*

  To use realHiver Command, install below plugin

  npm install cypress-real-events

  Add import "cypress-real-events" in support>e2e.js (cypress above 10)

  \*/

  it("TC1 - Mouse Over: Using Plugin realEvent ", () => {

    cy.visit("https://www.bigbasket.com/");

    cy.get(".icon.svg-header.svg-arrow-down-hover.svg-arrow-down-hover-dim"

    ).realHover("mouse");

    cy.get("a:contains('Beverages')").realHover("mouse");

  });

  it(" TC2 - Mouse Over: Using plugin realEvent", () => {

    cy.visit("https://naveenautomationlabs.com/opencart/");

    cy.get("li.dropdown>a:contains('Desktops')").realHover("mouse");

    cy.get("a:contains('PC (0)')").click();

  });

  it.only("TC3 - Mouse Over: Using Cypress inbuilt Command", () => {

    cy.visit("https://www.tradeling.com/ae-en");

    cy.get("[data-testid='all\_categories\_nav\_bar']").trigger("mouseover");

  });

});

describe('Multi-Country Application Test', () => {

    const country\_code = ['/ae/', '/uk/', '/in/', '/ca/']

     it('Should test application behavior for country', () => {

        country\_code.forEach((country) => {

            cy.visit(`https://www.apple.com${country}`)

            cy.url().should('include', country)

            cy.verifyCountry(country).then((text) => {

                cy.get('[title="Choose your country or region"]')

                    .should('have.text', text)

            })

        })

    });

});

Cypress.Commands.add('verifyCountry', (country\_code) => {

    switch (country\_code) {

        case '/in/':

            return 'India'

        case '/uk/':

            return 'United Kingdom'

        case '/ae/':

            return 'United Arab Emirates'

        case '/ca/':

            return 'Canada (English)'

        default:

            return 'United Arab Emirates'

    }

})

describe("Multi Origin in Cypress", () => {

  it("Multi Origin test using cy.origin() ", () => {

    cy.visit('https://www.magicbricks.com/')

    cy.get('a[title="Twitter"]').invoke('attr', 'target', '\_self').click({ force: true })

    cy.origin('https://twitter.com', () => {

      cy.contains('comJoined September 2009').should('be.visible')

    })

  });

  it("Error Case: When cy.origin is not passed", () => {

    cy.visit('https://www.magicbricks.com/')

    cy.contains('comJoined September 2009').should('be.visible')

  });

})

describe(" Open link in same tab", () => { });

it("loads the page using approach1", () => {

  cy.visit("https://test.io/coverage/website-testing");

  cy.get("a.cta-button\_\_button.button-cta--get-a-demo")

    .first()

    .invoke("attr", "target", "\_self")

    .click({ force: true });

  cy.go(-1)

});

it("loads the page using approach2", () => {

  cy.visit("https://test.io/coverage/website-testing");

  cy.get("button#onetrust-accept-btn-handler").click();

  cy.get("a.cta-button\_\_button.button-cta--get-a-demo")

    .first()

    .invoke("removeAttr", "target")

    .click({ force: true });

  cy.go('back')

});

it("loads the page using approach3", () => {

  cy.visit("https://test.io/coverage/website-testing");

  cy.get("button#onetrust-accept-btn-handler").click();

  cy.get('a[href="https://app.test.io/"]').click({ multiple: true });

});

describe("Pagination Test", () => {

  let numberOfPages;

  it("Pagination Example ", () => {

    cy.visit(

      "https://www.lambdatest.com/selenium-playground/table-pagination-demo"

    );

    cy.get("li#prev").as("next");

    cy.get("div.pagination-container>nav>ul")

      .find(">li")

      .then(($value) => {

        numberOfPages = $value.length;

        cy.log("page count" + numberOfPages);

        for (let i = 0; i <= numberOfPages; i++) {

          cy.get("table#table-id>tbody>tr>td")

            .eq(1)

            .then(($value) => {

              if ($value.text == "Stuart Johnston") {

                cy.log(" Name Found ");

                cy.log("page is " + i);

              } else {

                cy.log(" Name not Found ");

                cy.log(" Moving to the next page");

                cy.get("@next").click({ force: true });

              }

            });

        }

      });

  });

});

describe("Pause And Hide logs in Cypress", () => {

  it("Open website and enter username,pause the execution and pass password", () => {

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

    cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com");

    cy.pause();

   cy.get('[id="input-password"]').type("Cypress123!!");

    cy.get('[type="submit"]').eq(0).click();

  });

  it("Open website and enter username,hide the log - entered username", () => {

    cy.visit(

      "https://ecommerce-playground.lambdatest.io/index.php?route=account/login"

    );

    cy.get('[id="input-email"]').type("lambdatest.Cypress@disposable.com",{log:false});

    cy.get('[id="input-password"]').type("Cypress123!!");

  });

});

describe('PF Input Fields', () => {

    it("SERP Page", () => {

        cy.visit('https://www.propertyfinder.ae/en/buy/properties-for-sale.html')

        cy.get('[data-testid="multi-selection-autocomplete-template-input"]').type('Marina')

    })

    it("New project Page", () => {

        cy.visit('https://www.propertyfinder.ae/en/new-projects')

        cy.get('[data-testid="search-field-input"]').type('Marina')

    })

})

describe("Enter the text without passing .click()", () => {

  beforeEach(() => {

    cy.visit(

      "https://naveenautomationlabs.com/opencart/index.php?route=account/register"

    );

  });

  const dataJson = require("../ /testData.json");

  const input\_firstName = dataJson.firstName;

  const input\_lastName = dataJson.lastName;

  const input\_mobileNumber = dataJson.mobileNumber;

  const input\_password = dataJson.password;

  /\*

  Passing the input value from json file,

   passing email as random generator ,

   used .check() to select the checkbox

   used .tab() to enter tab ..

     Installed plugin => npm install -D cypress-plugin-tab

     for using tab as cypress does not have direct support for tab

  \*/

  it("Input values in the textbox and click on submit ", () => {

    cy.get("input#input-firstname")

      .type(input\_firstName)

      .tab()

      .type(input\_lastName, { log: false })

      .tab()

      .type("abc123" + Math.floor(Math.random() \* 10 + 1) + "@disposable.com")

      .tab()

      .type(input\_mobileNumber)

      .tab()

      .type(input\_password)

      .tab()

      .type(input\_password)

      .tab()

      .tab()

      .tab()

      .tab()

      .check()

      .tab()

      .click();

    cy.log("browser version ", Cypress.browser.version);

  });

});

describe('Refresh Page in Cypress', () => {

    it('Refresh page', () => {

        cy.visit('https://www.amazon.ae/')

        cy.contains('Best Sellers').should('be.visible')

        cy.reload()

        cy.contains('Best Sellers').should('be.visible')

        cy.reload(true)

        cy.contains('Best Sellers').should('be.visible')

        cy.reload({ timeout: 5000 })

        cy.contains('Best Sellers').should('be.visible')

        cy.window().then((data => {

            data.location.reload()

        }))

        cy.contains('Best Sellers').should('be.visible')

    })

})

describe("Scroll into view", () => {

  it("Infinite - scroll into view", () => {

    cy.visit("https://www.swiggy.com/");

    cy.get("#location").type("Gurgaon");

    cy.contains("Gurgaon, Haryana, India").click();

    cy.scrollTo("bottom");

    cy.contains("Pune").should("be.visible");

  });

});

import { onlyOn, skipOn } from '@cypress/skip-test'

describe("Conditionally disable test in Cypress using plugin", () => {

    it("Only on Chrome, Open URL and add product to cart", () => {

        cy.onlyOn('chrome')

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

    it("Only on Firefox, Open URL and add product to cart", () => {

        cy.onlyOn('firefox')

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

    it("Skip on Window, Open URL and add product to cart", () => {

        cy.skipOn('windows')

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

    it("Skip on MAC, Open URL and add product to cart", () => {

        cy.skipOn('mac')

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

    it("Skip on Windows, Run only on Mac and Browser Chrome", () => {

        skipOn('windows')

        //  cy.onlyOn("mac").onlyOn('chrome')

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

    it("Run on Firefox", () => {

        onlyOn("firefox")

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

    it("Skip on Firefox, Run only on Mac and Browser Chrome", () => {

        skipOn('firefox')

        cy.onlyOn("mac").onlyOn('chrome')

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    });

});

describe("Custom Commands", () => {

  it("use login method from custom commands", () => {

    cy.visit("https://bstackdemo.com/").screenshot();

    cy.get('.shelf-item\_\_buy-btn').eq(0).click();

    cy.screenshot("test3", { capture: "fullPage" });

  });

  it('should take a screenshot with timestamp', () => {

    cy.visit("https://bstackdemo.com/");

    cy.screenshotWithTimestamp('my-screenshot');

  });

});

import { onlyOn, skipOn } from '@cypress/skip-test'

describe('My Test Suite', () => {

    it('Test Case 1', { tags: 'smoke' }, () => {

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').first().click();

    })

    it('Test Case 2', { tags: 'regression' }, () => {

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').eq(1).click();

    })

    it('Test Case 3', { tags: ['smoke', 'regression'] }, () => {

        cy.visit(

            "https://bstackdemo.com/")

        cy.get('.shelf-item\_\_buy-btn').eq(2).click();

    })

})

describe('Find Todays Date', () => {

    it("Today date", () => {

        const today = new Date();

        const yyyy = today.getFullYear();

        let mm = today.getMonth() + 1; // Months start at 0!

        let dd = today.getDate();

        if (dd < 10) dd = '0' + dd;

        if (mm < 10) mm = '0' + mm;

        const formattedToday = dd + '/' + mm + '/' + yyyy;

        cy.log("===today dtae is ===" + formattedToday)

    })

})

describe('Verify Tool Tip text in Cypress', () => {

    it('Test Case 1 - Tool Tip Text', () => {

        cy.visit('https://practice.expandtesting.com/tooltips')

        cy.get('#btn1').trigger('mouseover')

        cy.get('.tooltip-inner').should('be.visible').and('have.text', 'Tooltip on top')

    })

    it('TC2 - Tool Tip Text', () => {

        cy.visit('https://open.spotify.com/')

        cy.get('[aria-label="Collapse Your Library"]').trigger('mouseover')

        cy.get('#hover-or-focus-tooltip').should('be.visible')

            .and('have.text', "Collapse Your Library")

    })

})

const screen = ['iphone-8', 'ipad-2', [1024, 768], 'macbook-13']

const orienatation = ['landscape', 'portrait']

describe('Testing different screen size and orienatation', () => {

    screen.forEach((screenView) => {

        it('Should display viewport for the screens=> Iphone-8,ipad-2 ,macbook-13', () => {

            if (Cypress.\_.isArray(screenView)) {

                cy.viewport(screenView[0], screenView[1])

                cy.log("printing viewports", screenView[0], screenView[1])

            }

            else {

                cy.viewport(screenView)

            }

            cy.visit('https://www.tradeling.com');

        })

    });

    it('Open website in iphone-xr viewport and then navigate back to the first page', () => {

        cy.viewport('iphone-xr')

        cy.visit('https://www.tradeling.com');

        cy.get('[data-testid="rfq-ctas"]').first().click();

        cy.go(-1);

    })

    it('open youtube in mobile web , perform action and again search for new text', () => {

        cy.viewport('samsung-s10')

        cy.visit("https://www.youtube.com");

        cy.get(' yt-icon-button#search-button-narrow').type('automation')

        cy.get('ul[role="listbox"]').contains("automation course").click({ force: true });

        cy.wait(5000);

        cy.get('yt-icon.topbar-icons.style-scope.ytd-masthead').click({ force: true })

        cy.get('input#search').clear().type('node js')

    })

    Cypress.\_.each(['landscape', 'portrait'], orienatation => {

        it(`Testing different ${orienatation}`, () => {

            cy.visit("https://www.dropbox.com/")

            cy.viewport('macbook-16', orienatation);

        })

    })

    it('Testing customized width and height', () => {

        cy.visit("https://www.dropbox.com/")

        cy.viewport(360, 890);

    })

    Cypress.\_.each(['macbook-15', 'samsung-note9'], viewport => {

        it(`works on ${viewport}`, () => {

            cy.visit("https://www.dropbox.com/")

            cy.viewport(viewport)

        })

    })

    it(`Config Viewports`, () => {

        cy.visit("https://www.dropbox.com/")

        cy.viewport(Cypress.config('viewportWidth'), Cypress.config('viewportHeight'))

    })

});

Package.json

{

  "name": "practice\_qa",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "Cypress\_Test": "cypress run",

    "cypress-prod-test": "npx cypress run --spec cypress/e2e/testEnv.cy.js --config-file prod.config.js",

    "cypress-stage-test": "npx cypress run --spec cypress/e2e/testEnv.cy.js --config-file stage.config.js"

  },

  "repository": {

    "type": "git",

    "url": "git+https://github.com/Anshita-Bhasin/Practise\_QA.git"

  },

  "author": "",

  "license": "ISC",

  "bugs": {

    "url": "https://github.com/Anshita-Bhasin/Practise\_QA/issues"

  },

  "homepage": "https://github.com/Anshita-Bhasin/Practise\_QA#readme",

  "devDependencies": {

    "@4tw/cypress-drag-drop": "^2.2.3",

    "@cypress/skip-test": "^2.6.1",

    "cypress": "13.2.0",

    "cypress-file-upload": "^5.0.8",

    "cypress-plugin-tab": "^1.0.5",

    "cypress-real-events": "^1.7.6",

    "playwright-webkit": "^1.25.2",

    "webpack": "^5.89.0"

  },

  "dependencies": {

    "Cypress": "npm:cypress@^12.8.1",

    "cypress-commands": "^3.0.0",

    "cypress-drag-drop": "^1.1.1",

    "cypress-plugin-api": "^2.6.1",

    "gremlins.js": "^2.2.0",

    "node-xlsx": "^0.23.0",

    "words-to-numbers": "^1.5.1",

    "xlsx": "^0.18.5"

  }

}

describe("cookies",()=>{

    it('getting login cookies', () => {

        cy.visit('https://opensource-demo.orangehrmlive.com/');

        cy.get('input[name="username"]').clear().type("Admin");

        cy.get('input[name="password"]').clear().type("admin123");

        cy.get('button[type="submit"]').click()

        cy.url().should('include',"/dashboard/")

        cy.getCookies().then((cookies) => {

            cookies.forEach((cookie) => {

              // Print the name and value of each cookie

              cy.log(`Cookie: ${cookie.name}, Value: ${cookie.value}`);

              cy.writeFile('cypress/fixtures/cookies.json', cookies);

            });

          });

    });

})

/// <reference types="cypress" />

describe("find broken links",()=>{

    it('find out broken links on homepage', () => {

               cy.findoutBrokenlinks(Cypress.env('LetMeShope'));

        })

    });

describe('globalization',()=>{

    it('handell globalization', () => {

        cy.visit("https://www.amazon.in/")

        cy.get("#icp-touch-link-country").click();

        cy.get(".a-dropdown-prompt").click();

        cy.get(".icp-2.icp-flag-2.icp-flag-2-br").click()

          // Click the element that would normally open a new tab

          cy.contains("Go to website", { matchCase: false })

  .should('exist')  // Wait for the element to exist

  .click({ force: true });

          // Optionally, check if window.open was called (if needed)

        cy.get(".icp-color-base").eq(1).should('contain.text',"Brasil")

    });

    it.only('ok', () => {

           // Visit the main page

           cy.visit("https://www.amazon.in/");

           // Click to open the country dropdown

           cy.get("#icp-touch-link-country").click();

           cy.get(".a-dropdown-prompt").click();

           cy.get(".icp-2.icp-flag-2.icp-flag-2-br").click();

           // Intercept window.open to prevent opening in a new tab

          // cy.SameTab();

           // Click the "Go to website" button

           cy.contains("Go to website", { matchCase: false }).click({ force: true });

           // Verify that the content has changed, i.e., the site navigation happened

           cy.get(".icp-color-base").should('contain.text', "Brasil");

            });

        });

        /\*\*window.open Stub:

Instead of invoking a Cypress command like cy.visit() inside the window.

open stub (which caused the issue), we directly set win.location.href = url.

This changes the page's URL in the same tab.

No Cypress Commands Inside Promises:

By using the native JavaScript redirection (win.location.href = url),

we avoid conflicts with Cypress' command queue,

preventing the CypressError related to the promise returned from cy.click().

Final Verification:

After clicking the "Go to website" button,

we confirm the navigation happened by checking that the page contains "Brasil." \*/

/// <reference types="cypress" />

describe('IFrames testing in Cypress', () => {

  const iframe = '#courses-iframe';

  beforeEach(() => {

    cy.visit('https://rahulshettyacademy.com/AutomationPractice/');

  });

  it('should prevent opening new tabs by stubbing window.open', () => {

    cy.frameLoaded(iframe);

    // Click on the first link inside the iframe

    cy.iframe().find('a[href="https://rahulshettyacademy.com/blog/"]').eq(0).invoke('removeAttr', 'target').click({ force: true });

    // Wait for the page to load

    cy.wait(5000);

    // Click on the button and prevent the new tab

    cy.iframe().find('.tcb-button-text.thrv-inline-text').eq(0).invoke('removeAttr', 'target').click({ force: true });

    // Ensure window.open was not called

  });

});

describe('localization ',()=>{

    it('handel localization', () => {

        cy.visit('https://www.amazon.in/')

        cy.get('.icp-nav-link-inner').click()

        cy.get('input[value="hi\_IN"]').click({force:true})

        cy.contains("परिवर्तन सहेजें").click({force:true})

        cy.get('div #nav-tools a ').each((ele,list)=>{

           const herf= ele.attr('href')

           if(herf.includes('hi'))

           {

            cy.log('lang chnage to local lang ,\*\*\*localization done\*\*\*')

           }

        })

    });

    it('handel localization', () => {

        cy.visit('https://www.amazon.in/')

        cy.get('.nav-template.nav-flyout-content.nav-tpl-itemList[lang="hi-IN"]').click();

    });

})

it("Pagination", () => {

    cy.visit("https://rahulshettyacademy.com/seleniumPractise/#/offers");

    findItem("Almond");

  });

  // Steps:

  // 1. Iterate through Pages - Make sure page elements are refreshed

  // 2. Iterate through the table to find the matching record

  // 3. Break all loops once the matching record is found

  function findItem(value) {

    function findInPage(pageIndex) {

      let found = false;

      // Ensure the next page button is available before clicking

      cy.get("a[aria-label='Next']").as("nextPageButton");

      // Check the table rows

      cy.get("tbody tr").each((row, index) => {

        const itemText = row.text();

        cy.log(`Row ${index}: ${itemText}`);

        // If the item is found, log its corresponding value

        if (itemText.includes(value)) {

          found = true;

          // Get the next cell in the same row for the price

          cy.get("tbody tr")

            .eq(index)

            .find("td")

            .eq(1) // Assuming the price is in the second column (adjust if necessary)

            .then((priceCell) => {

              const price = priceCell.text();

              cy.log(`Found ${value} with price: ${price}`);

            });

          // Break out of the loop by returning false

          return false;

        }

      }).then(() => {

        // If not found on this page, go to the next page and search again

        if (!found) {

          cy.get("@nextPageButton").then(($nextButton) => {

            if (!$nextButton.hasClass('disabled')) {

              cy.get("@nextPageButton").click();

              findInPage(++pageIndex);

            } else {

              // Stop when no more pages are left

              cy.log(`${value} not found after checking all pages`);

            }

          });

        }

      });

    }

    // Start searching from page 1

    findInPage(0);

  }

describe('Search Product', () => {

    it('Search and add product to cart', () => {

        cy.visit('https://www.amazon.in/')

        // Select 'Electronics' from the dropdown

        cy.get('#searchDropdownBox').select("Electronics", { force: true })

        // Type 'iphone' and press Enter

        cy.get('#twotabsearchtextbox').type("iphone{enter}")

        // Loop through the search results

        cy.get('.a-link-normal.s-underline-text.s-underline-link-text.s-link-style.a-text-normal').each((ele, index) => {

            // Get the text of each product

            const productText = ele.text();

            cy.log(productText)

            // If the desired product is found

            if (productText.includes('Apple iPhone 13 (128GB) - Starlight')) {

                // Log product found

                cy.log('Product found: Apple iPhone 13 (128GB) - Midnight')

                // Remove target attribute to prevent opening in new tab

                cy.wrap(ele).invoke("removeAttr", "target").click();

                cy.wait(2000)

                // Click 'Add to Cart' button

                cy.get('#add-to-cart-button').click();

                // Exit the loop after adding to cart

                return false;

            } else {

                cy.log('Product not found')

            }

        })

    });

});

describe('shadow dom',()=>{

    it('handel shadow dom', () => {

        cy.visit('https://www.virustotal.com/')

       // cy.get("#URL-tab",{includeShadowDom:true}).click();

       cy.get("#URL-tab").click();

    });

})

describe('set coookies',()=>{

    beforeEach(()=>{

        cy.fixture('cookies.json').then((cookies) => {

            cookies.forEach((cookie) => {

              cy.setCookie(cookie.name, cookie.value);

            });

          });

          cy.visit('https://opensource-demo.orangehrmlive.com/')

    })

    it.only('skip login', () => {

        cy.url().should('include',"/dashboard/")

    });

})

Cypress.Commands.add('findoutBrokenlinks',(url)=>{

    let brokenlinks=0;

    let activelinks=0;

    cy.visit(url);

    cy.get('a').each((ele,index,list)=>{

//cy.log(ele)

     const href= ele.attr('href')

    if(href)

    {

        cy.request({url: href, failOnStatusCode: false}).then((response)=>{

            if (response.status>= 400) {

                cy.log(` link --> ${index + 1} is BROKEN Link --> ${href} `)

                brokenlinks++;

            }

            else{

                cy.log(`link --> ${index +1} is active --> ${href}`)

                activelinks++;

            }

        })

    }

    }).then((links)=>

    {

       const totallinks= links.length

       cy.log(`totallinks is ${totallinks}`)

       cy.log(`broken links is ${brokenlinks}`)

       cy.log(`active links is ${activelinks}`)

    })

})

Cypress.Commands.add("SameTab",()=>{

    cy.window().then((win) => {

        cy.stub(win, 'open').callsFake((url) => {

            // Instead of opening a new tab, visit the URL in the same tab

            win.location.href = url;

        }).as('windowOpen');

    });

})

// cypress/support/index.js or e2e.js

Cypress.on('window:before:load', (win) => {

    // Override window.open to always open in the same tab

    cy.stub(win, 'open').callsFake((url) => {

      win.location.href = url;

    });

  });

{

  "name": "letmeshop",

  "version": "1.0.0",

  "main": "index.js",

  "scripts": {

    "test": "npx cypress   --reporter mochawesome --reporter-options reportDir=cypress/reports/mocha,reportFilename=mochawesome-report,overwrite=false,html=true,json=true",

    "cypress-test": "npx cypress run --record --parallel --key 436bbead-e9d3-48d4-b703-e09a0cfc0818 --ci-build-id '9k549i' ",

    "newman-test": " newman run postman/collection.json -e postman/QAEnv.json -r htmlextra --reporter-htmlextra-export newman/results/newman-report.html || echo 'Newman tests failed'"

  },

  "author": "",

  "license": "ISC",

  "description": "All",

  "devDependencies": {

    "@cypress-audit/lighthouse": "^1.4.2",

    "cypress": "^13.14.1",

    "cypress-cucumber-preprocessor": "^4.3.1",

    "cypress-iframe": "^1.0.1",

    "mochawesome": "^7.1.3",

    "newman": "^6.2.1",

    "newman-reporter-htmlextra": "^1.23.1"

  },

  "cypress-cucumber-preprocessor": {

    "nonGlobalStepDefinitions": false,

    "step\_definitions": "cypress/integration/Tests"

  }

}