Expectdemo.js file contains:

///<reference types=”cypress” />

describe(“Orange HRM Test”, () => {

it(“Test – Using expect()”, () => {

cy.visit(“https://opensource-demo.orangehrmlive.com/web/index.php/auth/login”);

// expect() to assert username attribute present in username text.

cy.get(‘[placeholder=”Username”]’).then(($username) => {

A screenshot of a computer program

Description automatically generated

expect($username).to.have.attr(‘name’, ’username’ )

// expect($username).to.have.attr(‘name’, ’username’ ).to.have.attr(‘placeholder’);

// if I want to type some value inside username

// if we want to interact with the element then we have to convert jquery element to // cypress object

cy.wrap($username).type(‘Admin’)

})

cy.get(‘[name=”password”]’).then(($pass) => {

expect($pass).to.have.attr(‘placeholder’, ‘Password’)

cy.wrap($pass).type(‘admin123’)

})

// login button – I want to assert the text login.

cy.get(‘[type=”submit”]’).then(($login) => {

A screenshot of a computer

Description automatically generated

expect($login).to.have.text(“ Login ”);

cy.wrap($login).click();

})

A screenshot of a computer

Description automatically generated

// Clicking on the element which opens the dropdown

cy.get(‘.oxd-userdropdown-name’).click();

// we have to iterate the 4 elements using each method

cy.get(‘.oxd-dropdown-menu’).find(‘[role=”menuitem”]’).each(($ele) => {

if($ele.text() == ‘Logout’){

cy.wrap($ele).click();

}

})

})

})

New program: make rsdemo.js file in basics folder:

///< reference types=”cypress” />

describe(‘Rahul Shetty test’, () => {

// for skipping: it.skip(‘checkbox test’, () => {

it(‘checkbox test’, () => {

cy.visit(‘https://rahulshettyacademy.com/AutomationPractice/’)

// Assert the URL

cy.url().should(‘include’, ‘AutomationPractice’)

// Checking on checkbox using check()

cy.get(‘#checkBoxOption1’).check().should(‘be.checked’); // assertion whether we have

// checked or not

cy.get(‘#checkBoxOption2’).check().should(‘be.checked’);

cy.get(‘#checkBoxOption3’).check().should(‘be.checked’);

// Unchecking the checkbox using uncheck()

cy.get(‘#checkBoxOption1’).uncheck().should(‘not.be.checked’);

cy.get(‘#checkBoxOption2’).uncheck().should(‘not.be.checked’);

cy.get(‘#checkBoxOption3’).uncheck().should(‘not.be.checked’);

});

It(‘hide/show test’, () => {

cy.visit(‘https://rahulshettyacademy.com/AutomationPractice/’)

// asserting whether the textbox is visible

cy.get(‘[name=”show-hide”]’).should(‘be.visible’)

// Clicking on the hide element

cy.get(‘#hide-textbox’).click();

// assert for the invisibility of the textbox

cy.get(‘[name=”show-hide”]’).should(‘not.be.visible’)

cy.get(‘#show-textbox’).click();

// again asserting for visibility

cy.get(‘[name=”show-hide”]’).should(‘be.visible’).type(‘Element is visible’)

// for assertion

// cy.get(‘[name=”show-hide”]’).should(‘be.visible’).type(‘Element is visible’).should(‘have.value’, ‘Element is visible’)

});

});