1. **Button Functionality Code**

import {test, expect, selectors} from '@playwright/test' ;

test('select\_ex1', async({page}) => {

    await page.goto('https://ultimateqa.com/simple-html-elements-for-automation/');

     const link =await page.locator('xpath=(//\*[@class="et\_pb\_button et\_pb\_promo\_button"])[1]');

    // let var2 = link.getAttribute('href');

    let var3 = await page.locator('//\*[@id="post-909"]/div/div[1]/div/div[3]/div/div[1]/div[2]/div[2]/a').getAttribute('href');

    console.log("Button label name is: ", var3);

    await link.click();

    await await expect(page).toHaveTitle('Button success - Ultimate QA');

});

1. **CheckBox Functionality Code**

import {test, expect, selectors} from '@playwright/test' ;

test('CheckBox', async({page}) => {

    await page.goto('https://ultimateqa.com/simple-html-elements-for-automation/');

    //await page.pause();

      //Counting Values

      const element=await page.locator("//div[text()='Select a checkbox and validate that they are selected']");

      const options=await page.$$("//input[@type='checkbox']");

      console.log("Number of Radio Buttons:", options.length);

      //For Fetching Drop Down List

    //let status=false;

    for(let i=0;i< await options.count(); i++)

    {

        const list= options.length(i);

        console.log(await list.textContent());

    }

    //Code For Single Checkbox

    // await page.locator("//input[@value='Bike']").click();

    // await page.waitForTimeout(3000)

    // await page.check("//input[@value='Car']");

    // await page.waitForTimeout(3000)

    // await expect(page.locator("//input[@value='Bike' and @type='checkbox']")).toBeChecked();

    // await expect(page.locator("//input[@value='Bike' and @type='checkbox']").isChecked()).toBeTruthy();

    // await page.waitForTimeout(5000)

//Multiple CheckBox

    const checkBox = [

        "//input[@value='Bike' and @type='checkbox']",

        "//input[@value='Car' and @type='checkbox']"

    ];

    //Multiple select

    for (const locator of checkBox) {

        await page.click(locator);

    }

    //If Checked then uncheck the check box

    await page.waitForTimeout(3000);

    for (const locator of checkBox) {

        if(await page.locator(locator).isChecked())

        await page.locator(locator).uncheck();

    }

    await page.waitForTimeout(6000);

});

1. Count The Locators from the Page

import {test, expect, selectors} from '@playwright/test' ;

import { count, log } from 'console';

test('DropDown', async({page}) => {

    await page.goto('https://ultimateqa.com/simple-html-elements-for-automation/');

    //Counting Text Boxes From The Page

    const textBox= page.locator("//\*[@class='input' and @type='text']");

    const textBoxCount= await textBox.count();

    console.log("Text Box Count Is: " + textBoxCount);

    //Count Drop down groups

    const dropDown= await page.locator("//select");

    console.log("Drop Down Count: "+ await dropDown.count());

    //Count of Drop down values

    const dropDownValues= await page.locator("//option");

    console.log("Drop Down Values Count: "+ await dropDownValues.count());

    await expect(page.locator("//option")).toHaveCount(4);

    //Count of Radio Buttons Count

    const radioButtons= await page.$$("//\*[@type='radio']");

    //console.log("Radio Button Count: "+radioButtons.cou;

    console.log("Value is:" ,count(radioButtons));

    //Count of Check Box

    const checkBox= await page.$$("//\*[@type='checkbox']");

    const checkBoxCount= await checkBox.length;

    console.log("Check Box Count: "+checkBoxCount);

    //Count of List .menu-item

    const listC= await page.$$("//li");

    //await listC.waitFor();

    const ListCount= listC.length

    console.log("List Count: "+ListCount);

    //Count of Buttons

    const button = await page.locator("//button[@type='submit' or @id='button1']");

    //await button.waitFor();

    const buttonCount = await button.count();

    console.log("Button Count Is: " + buttonCount);

    //Links Count

    const links= await page.locator("//\*[@href='/link-success/' or @href='#']");

    const linksCount= await links.count();

    console.log("Links Count Is: "+linksCount);

});

1. Date Picker Code

import {test, expect, selectors} from '@playwright/test' ;

import { count, log } from 'console';

test('Date Picker', async({page}) => {

    await page.goto('https://testautomationpractice.blogspot.com/');

   const Year= "2024";

   const Month= "June";

   const Day= "17";

   await page.click('#datepicker');

   let currentYear;

   let currentMonth;

   while(true)

   {

     currentYear= await page.locator('.ui-datepicker-year').textContent();

     currentMonth= await page.locator('.ui-datepicker-month').textContent();

     if(currentYear == Year && currentMonth == Month)

     {

        break;

     }

     await page.locator('[title="Next"]').click();

   }

   await page.click(`//a[@class='ui-state-default'][text()='${Day}']`);

   await page.waitForTimeout(6000)

});

1. List the Number Of Options

import {test, expect, selectors} from '@playwright/test' ;

test('SampleDemo', async({page}) => {

    await page.goto('https://freelance-learn-automation.vercel.app/signup');

    // const element1= page.locator("#state");

    // const list=element1.locator("option");

    // //const list1 = page.locator('list > .component');

    // await expect(list).toHaveCount(37);

    await page.waitForTimeout(3000)

    const options=await page.$$("#state option");

    console.log("Number of options:", options.length);

    await page.waitForTimeout(3000)

});

1. Drop Down Value Code

import {test, expect, selectors} from '@playwright/test' ;

import { clear, log } from 'console';

    //await page.waitForSelector("#Country-1-list li");

    //const value1=page.locator("#select-ui-input--Country-1");

test('Dropdown1', async({page}) => {

    const expectedValue="Armenia";

    const expectedValue1="Albania";

    await page.goto('https://www.telerik.com/support');

    const value1=await page.locator("#select-ui-input--Country-1");

    await value1.type("Armenia",{delay:1000});

    await page.waitForSelector("//li[contains(@class,'select-ui')]/mark[1]");

    const StateOptions= await page.locator("//li[contains(@class,'select-ui')]/mark[1]");

    const options=await page.$$("#Country-1-list li");

    console.log("Number of options:", options.length);

    for(let i=0;i<options.length;i++)

    {

        const value= await StateOptions.textContent();

        if(value===expectedValue)

        {

            await StateOptions.click();

            await page.waitForTimeout(4000);

            await value1.clear();

            await value1.scrollIntoViewIfNeeded();

            await page.waitForTimeout(4000);

           await value1.type("Albania",{delay:1000});

        }

        else if(value===expectedValue1)

        {

            await StateOptions.click();

            bre

            await page.waitForTimeout(4000);

        }

    }

    //  //Counting Values

    //  const options=await page.$$("#Country-1-list li");

    //  console.log("Number of options:", options.length);

    //  //Printing all values using for loop

    //  for(let i=0; i<options.length; i++)

    //  {

    //     let element= options[i];

    //     let value= await element.textContent();

    //     console.log("Value from drop down:" +value);

    //  }

    await page.waitForTimeout(5000)

});

II) Drop down Value Code 2.

import {test, expect, selectors} from '@playwright/test' ;

import { clear, log } from 'console';

test('Auto Suggest', async({page}) => {

    await page.goto('https://www.telerik.com/support');

    test.setTimeout(160000);

     //Counting Values and clicking on the Text Box for selection

     const options=await page.$$("//ul[@class='select-ui-list']/child::li");

     console.log("Number of options:", options.length);

     const value1= page.locator("#select-ui-input--Country-1");

     await value1.scrollIntoViewIfNeeded();

     await value1.waitFor();

     let element;

     let value;

     //Printing all values using for loop

     for(let i=0; i<options.length; i++)

     {

        await value1.click();

        element=options[i];

        await element.click();

        value= await element.textContent();

        console.log("Selected Value from drop down:" +value);

     }

});

1. Locator Selection Code

// @ts-check

const { test, expect } = require('@playwright/test');

test('has title', async ({ page }) => {

  await page.goto('https://playwright.dev/');

  // Expect a title "to contain" a substring.

  await expect(page).toHaveTitle(/Playwright/);

});

test('get started link', async ({ page }) => {

  await page.goto('https://playwright.dev/');

  // Click the get started link.

  await page.getByRole('link', { name: 'Get started' }).click();

  // Expects page to have a heading with the name of Installation.

  await expect(page.getByRole('heading', { name: 'Installation' })).toBeVisible();

});

1. HTML Table Code

import {test, expect, selectors} from '@playwright/test' ;

test('HTMLTable', async({page}) => {

    await page.goto('https://ultimateqa.com/simple-html-elements-for-automation/');

    const table= await page.locator("#htmlTableId");

    await table.scrollIntoViewIfNeeded();

    await page.waitForTimeout(4000)

    //Finding the Columns Count from the Table

    const columns= await table.locator("tbody tr th");

    console.log("Number of columns:",await columns.count());

    expect(await columns.count()).toBe(3);

    //Finding the rows Count from the Table

    const rows= await table.locator("tbody tr");

    console.log("Number of rows:",await rows.count());

    expect(await rows.count()).toBe(4);

    let row;

    let tds;

    //Print all the Products Name

    for(let i=0; i<await rows.count(); i++)

    {

        row=rows.nth(i);

        tds= row.locator('td');

        for (let j=0; j<await tds.count();j++)

        {

            console.log(await tds.nth(j).textContent());

        }

    }

    await page.waitForTimeout(6000)

});

1. Radio Button Code

import {test, expect, selectors} from '@playwright/test' ;

test('RadioButton', async({page}) => {

    await page.goto('https://ultimateqa.com/simple-html-elements-for-automation/');

    //await page.pause();

      //Counting Values

      const element=await page.locator("//\*[@class='et\_pb\_blurb\_description']");

      const options=await page.$$("//input[@type='radio']");

      console.log("Number of Radio Buttons:", options.length);

      //For Fetching Drop Down List

    let status=false;

    for(const option of options)

    {

        console.log(await option.getAttribute('value'));

    }

    await page.locator("//input[@value='female']").click();

    //await page.check("//input[@value='female']");

    //await page.pause();

    await page.waitForTimeout(3000)

    await expect(page.locator("//input[@value='female']")).toBeEnabled();

    await expect(page.locator("//input[@value='female']")).toBeVisible();

    await expect(page.locator("//input[@value='female']").isChecked()).toBeTruthy();

    await page.waitForTimeout(1000)

    await page.locator("//input[@value='male']").click();

    await expect(page.locator("//input[@value='male']")).toBeEnabled();

    await page.waitForTimeout(1000)

    await page.locator("//input[@value='other']").click();

    await expect(page.locator("//input[@value='other']")).toBeEnabled();

    await page.waitForTimeout(3000)

    //await expect(page.locator("//input[@value='male']").isChecked()).toBeFalsy();

});

1. Read Excel Code
2. var XLSX= require("xlsx");
3. var workbook= XLSX.readFile("data/Customer.xlsx");
4. let worksheet= workbook.Sheets[workbook.SheetNames[0]];
5. for(let index= 2; index < 8; index++)
6. {
7. const id= worksheet[`A${index}`].v;
8. const name= worksheet[`B${index}`].v;
9. console.log({
10. id:id, name:name
11. })
12. }

9. Read PDF Code

const { test, expect } = require('@playwright/test');

const fs = require('fs');

// Define a test using Playwright's `test` function

test('verify content', async ({ page }) => {

  // Navigate to a URL that serves a PDF file

  await page.goto('https://onlinetestcase.com/pdf-file/');

  await page.waitForTimeout(6000)

  // Wait for the download event and click on a link to download the PDF file

  const [download] = await Promise.all([

    page.waitForEvent('download'),

    await page.click("//h2[@class='elementor-heading-title elementor-size-large'][text()='Download 500 KB PDF File']"),

    await page.waitForTimeout(6000)

  ]);

  // Use the suggested filename from the download event to save the file

  const suggestedFileName = download.suggestedFilename();

  const filePath = 'ExportData/' + suggestedFileName;

  await download.saveAs(filePath);

  // Use the 'pdf-parse' module to extract the text from the PDF file

  var pdf = require('pdf-parse');

  var dataBuffer = fs.readFileSync('./ExportData/sample.pdf');

  await pdf(dataBuffer).then(function(data) {

    fs.writeFileSync('./ExportData/actual.txt', data.text);

  });

  // Read the expected and actual values from the saved files

  let expected\_export\_values = fs.readFileSync('./ExportData/expected.txt', 'utf-8');

  let actual\_export\_values = fs.readFileSync('./ExportData/actual.txt', 'utf-8');

  // Use the `expect` function from Playwright to assert that the values match

  expect(expected\_export\_values).toMatch(actual\_export\_values);

});

10. Drop Down With Capturing List

import {test, expect, selectors} from '@playwright/test' ;

test('DropDown', async({page}) => {

    await page.goto('https://ultimateqa.com/simple-html-elements-for-automation/');

    const element= page.locator('xpath=//div[@class="et\_pb\_blurb\_description"]/select');

    await element.scrollIntoViewIfNeeded();

    //Counting Values

    const options=await page.$$("option");

    console.log("Number of options:", options.length);

    //For Fetching Drop Down List

        //console.log(await options.textContent());

        //await element.click();

        const result=await element.selectOption({index: 1});

        console.log("Selected Drop Down Value 1: " +result);

        await page.waitForTimeout(1000)

        //Second Selection

        const result1=await element.selectOption({index: 2});

        const text1= await page.$eval("option", ele => ele.value);

        console.log("Selected Drop Down Value 2 : " +result1);

        await page.waitForTimeout(1000)

        //Second Selection

        const result2=await element.selectOption({index: 3});

        const text2= await page.$eval("option", ele => ele.value);

        console.log("Selected Drop Down Value 3 : " +result2);

        await page.waitForTimeout(1000)

    await page.waitForTimeout(6000)

});

11. Web Table Code

import {test, expect, selectors} from '@playwright/test' ;

test('Web Table', async({page}) => {

    await page.goto('https://testautomationpractice.blogspot.com/');

    const table= await page.locator("#productTable");

    await table.scrollIntoViewIfNeeded();

    //Finding the Columns Count from the Table

    const columns= await table.locator("thead tr th");

    console.log("Number of columns:",await columns.count());

    expect(await columns.count()).toBe(4);

    //Finding the rows Count from the Table

    const rows= await table.locator("tbody tr");

    console.log("Number of rows:",await rows.count());

    expect(await rows.count()).toBe(5);

   //select product using reusbale function

   await selectProduct(rows, page, 'Product 1');

   await selectProduct(rows, page, 'Product 3');

    await page.waitForTimeout(6000)

});

async function selectProduct(rows, page, name)

{

    const matchRow= rows.filter({

        has: page.locator('td'),

        hasText: name

    })

   await matchRow.locator('input').check();

}

12. Auto Suggest Drop Down Values

import {test, expect, selectors} from '@playwright/test' ;

import { log } from 'console';

test('Auto Suggest Drop Down', async({page}) => {

    const searchValue= "Armenia";

    const expectedValue= "Armenia";

    await page.goto('https://www.telerik.com/support');

    await page.locator("#select-ui-input--Country-1").type(searchValue, {delay:1000});

    const options=await page.locator("#Country-1-list li");

     const countOptions= options.length;

     await page.waitForSelector("//li[contains(@class,'select-ui')]/mark[1]");

    const StateOptions= await page.$$("//li[contains(@class,'select-ui')]/mark[1]");

     for(let i=0; i<countOptions; i++)

     {

        const text=await StateOptions.textContent();

        if(text.includes("Armenia"))

        {

            await StateOptions.click();

            break;

        }

     }

});