Assessments

1)Playwright - JPetStore:

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

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

await page.goto('https://jpetstore.aspectran.com/catalog/');

await page.locator('#QuickLinks').getByRole('link', { name: 'Fish' }).click();

await page.getByRole('link', { name: 'FI-SW-01' }).click();

await page.getByRole('link', { name: 'Add to Cart' }).first().click();

await page.getByRole('link', { name: 'Proceed to Checkout' }).click();

await page.getByRole('button', { name: 'Login' }).click();

await page.getByRole('button', { name: 'Continue' }).click();

await page.getByRole('button', { name: 'Confirm' }).click();

await page.locator('#BackLink').click();

await page.locator('#QuickLinks').getByRole('link', { name: 'Birds' }).click();

const test1 = await page.getByRole('cell', { name: 'AV-CB-01' }).textContent();

const aaa = test1?.trim();

console.log(aaa);

console.log(await page.getByRole('cell', { name: 'Amazon Parrot' }).textContent());

const test2 = await page.getByRole('cell', { name: 'AV-SB-02' }).textContent();

const bbb=test2?.trim();

console.log(bbb);

console.log(await page.getByRole('cell', { name: 'Finch' }).textContent())

await expect(page).toHaveTitle(/JPetStore Demo/);

await page.pause();

});

2)API Testing: Postman

Validate the data type in the Test tab:

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

var data = pm.response.json();

pm.test("data type validation", function () {

pm.expect(data[0].id).to.be.a("number");

pm.expect(data[0].name).to.be.a("string");

pm.expect(data[0].grade).to.be.a("number");

pm.expect(data[0].mark).to.be.an("object");

pm.expect(data[0].mark.maths).to.be.a("number");

pm.expect(data[0].mark.physics).to.be.a("number");

pm.expect(data[0].mark.chemistry).to.be.a("number");}

);

3)SQL

select \* from FG\_user

select \* from FG\_party\_contact

select ID,User\_Id from FG\_user

select ID from FG\_party\_contact

select DISTINCT fg\_user.id,FG\_user.User\_id,FG\_party\_contact.id

from fg\_user

inner join FG\_party\_contact on fg\_user.party\_id=fg\_party\_contact.party\_id

select DISTINCT REFERENCE as EMAIL

from FG\_party\_contact

where TYPE\_CODE='EMAIL'

select DISTINCT REFERENCE as MOBILE

from FG\_party\_contact

where TYPE\_CODE='MOBILE';

select DISTINCT fg\_user.id,FG\_user.User\_id,FG\_party\_contact.id,REFERENCE as MOBILE

from fg\_user

inner join FG\_party\_contact on fg\_user.party\_id=fg\_party\_contact.party\_id

where TYPE\_CODE='MOBILE';

select DISTINCT fg\_party\_contact.REFERENCE as MOBILE,fg\_user.id,FG\_user.User\_id,FG\_party\_contact.id,

(select DISTINCT reference from fg\_party\_contact

where TYPE\_CODE='EMAIL' and fg\_user.party\_id=fg\_party\_contact.party\_id) as EMAIL

from fg\_user

inner join FG\_party\_contact on fg\_user.party\_id=fg\_party\_contact.party\_id

where TYPE\_CODE='MOBILE';

4)Playwright – Amazon

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

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

await page.goto('https://www.amazon.in/');

await page.getByPlaceholder('Search Amazon.in');

await page.keyboard.press('Enter');

await page.getByPlaceholder('Search Amazon.in').fill('android phone');

await page.keyboard.press('Enter');

await page.locator("img[data-image-index='1']").click();

await page.getByPlaceholder('Search Amazon.in').fill('decor');

await page.keyboard.press('Enter');

await page.locator("img[data-image-index='1']").click();

await page.pause();

});

5)Playwright – UltimateQA

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

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

await page.goto('https://ultimateqa.com/automation/');

await page.locator('#menu-home-page-menu').getByRole('link', { name: 'Professional Services' }).click();

await page.goBack();

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

await page.goBack();

await page.getByRole('link', { name: 'Success Stories' }).click();

//await page.locator('video').first().textContent();

//await page.getByLabel('Close').click();

//await page.goto('https://ultimateqa.com/testimonials/');

console.log(await page.locator("a[href='https://courses.ultimateqa.com/enroll/808620?et=free\_trial']").textContent());

//await page.goto('https://ultimateqa.com/automation/');

await page.pause();

});

1. SQL query:

select \* from fg\_product\_req;

select SUB\_TYPE\_CODE, CUSTOMER\_CRN, APPLICANT\_USER\_CRN,APPLICATION\_DATE, CREATED\_ON from fg\_product\_req;

select SUB\_TYPE\_CODE, CUSTOMER\_CRN, APPLICANT\_USER\_CRN,APPLICATION\_DATE, CREATED\_ON

from fg\_product\_req

where SUB\_TYPE\_CODE='FG\_PRODUCT\_REQ\_TRADE' AND CREATED\_ON='14-09-23 11:05:53.000000000 AM';

select SUB\_TYPE\_CODE, CUSTOMER\_CRN, APPLICANT\_USER\_CRN,APPLICATION\_DATE, CREATED\_ON

from fg\_product\_req

where SUB\_TYPE\_CODE='FG\_PRODUCT\_REQ\_TRADE' OR CREATED\_ON between '14-09-23 10:00:00.000000000 AM' AND '14-09-23 12:40:00.000000000 PM';

select SUB\_TYPE\_CODE, CUSTOMER\_CRN, APPLICANT\_USER\_CRN,APPLICATION\_DATE, CREATED\_ON

from fg\_product\_req

where CREATED\_ON between '14-09-23 10:00:00.000000000 AM' AND '14-09-23 05:40:00.000000000 PM';

select SUB\_TYPE\_CODE, CUSTOMER\_CRN, APPLICANT\_USER\_CRN,APPLICATION\_DATE, CREATED\_ON

from fg\_product\_req

where SUB\_TYPE\_CODE='FG\_PRODUCT\_TRADE' OR CREATED\_ON between '14-09-23 10:00:00.000000000 AM' AND '14-09-23 05:40:00.000000000 PM';

select count(SUB\_TYPE\_CODE),CUSTOMER\_CRN,APPLICANT\_USER\_CRN, APPLICATION\_DATE

FROM fg\_product\_req

group by CUSTOMER\_CRN,APPLICANT\_USER\_CRN, APPLICATION\_DATE;

select SUB\_TYPE\_CODE, STAGE\_CODE,STATUS\_CODE,CREATED\_ON FROM fg\_product\_req ;

SELECT max(prod\_name\_desc), stage\_code, max(status\_code), max(prod\_name\_desc), count(customer\_crn),count(last\_updated\_by) FROM fg\_product\_req

GROUP BY stage\_code;

select stage\_code, status\_code,count(customer\_crn),

case

when SUB\_TYPE\_CODE='FG\_PRODUCT\_REQ\_TRADE' then 'Instant Trade' else 'Instant CMS' end as Product

from fg\_product\_req

/\*where created\_on between '14-09-23 10:00:00.000000000 AM' AND '14-09-23 05:40:00.000000000 PM'\*/

where created\_on like '14-09-23\_%'

group by stage\_code, sub\_type\_code, status\_code