***Day 7 – Github Workflows, Authentication***

1. **Github Workflow**

This GitHub Actions **workflow** is designed to **automatically run Playwright tests** when code is pushed to or a pull request is made against the main or master branches, specifically for files in the particular directory.

**Key Features:**

* **Triggers**: On push and pull\_request to main or master.
* **Environment**: Runs on the latest Ubuntu environment.
* **Working Directory**: All commands execute in Playwright\_UST/Day\_7\_GithubWorkflow.
* **Parallel Execution**: Tests are sharded across **4 jobs** (shardIndex 1–4), improving speed.
* **Setup**:
  + Checks out the code.
  + Sets up Node.js.
  + Installs dependencies and Playwright browsers.
* **Test Execution**: Runs npx playwright test with sharding.
* **Reporting**: Uploads test reports (blob reports) as artifacts, with 1-day retention.

name: Playwright Tests

on:

  push:

    branches: [ main, master ]

    paths:

            - 'Playwright\_UST/Day\_7\_GithubWorkflow/\*\*'

  pull\_request:

    branches: [ main, master ]

    paths:

            - 'Playwright\_UST/Day\_7\_GithubWorkflow/\*\*'

jobs:

  playwright-tests:

    timeout-minutes: 60

    runs-on: ubuntu-latest

    defaults:

      run:

        working-directory: Playwright\_UST/Day\_7\_GithubWorkflow

    strategy:

      fail-fast: false

      matrix:

        shardIndex: [1, 2, 3, 4]

        shardTotal: [4]

    steps:

    - uses: actions/checkout@v4

    - uses: actions/setup-node@v4

      with:

        node-version: lts/\*

    - name: Install dependencies

      run: npm ci

    - name: Install Playwright browsers

      run: npx playwright install --with-deps

    - name: Run Playwright tests

      run: npx playwright test --shard=${{ matrix.shardIndex }}/${{ matrix.shardTotal }}

      working-directory: Playwright\_UST/Day\_7\_GithubWorkflow

    - name: Upload blob report to GitHub Actions Artifacts

      if: ${{ !cancelled() }}

      uses: actions/upload-artifact@v4

      with:

        name: blob-report-${{ matrix.shardIndex }}

        path: blob-report

        retention-days: 1

1. **Authentication(Setup, user.json)**

To **reuse login state** across Playwright test runs, avoiding repeated logins and speeding up test execution.

1. **Login Script** (global-setup.ts or similar):
   * Runs once before tests.
   * Logs in using UI or API.
   * Saves the session (cookies + storage) to a file like user.json.
2. **Test Configuration** (playwright.config.ts):
   * Loads user.json using storageState to apply the authenticated session to your test context.

Once user.json is saved, any test that uses storageState will be pre-authenticated.

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

import path from 'path';

import fs from 'fs';

import { Certificate } from 'crypto';

const authFile = path.join(\_\_dirname, '../playwright/.auth/user.json');

const { username, password } = JSON.parse(fs.readFileSync(authFile, 'utf-8'));

setup('authenticate', async ({ page }) => {

  // Perform authentication steps. Replace these actions with your own.

  await page.goto('https://github.com/login');

  await page.getByLabel('Username or email address').fill(username);

  await page.getByLabel('Password').fill(password);

  await page.locator('[value="Sign in"]').click();

  // Wait until the page receives the cookies.

  //

  // Sometimes login flow sets cookies in the process of several redirects.

  // Wait for the final URL to ensure that the cookies are actually set

  await page.waitForURL('https://github.com');

  // End of authentication steps.

  await page.context().storageState({ path: authFile });

});