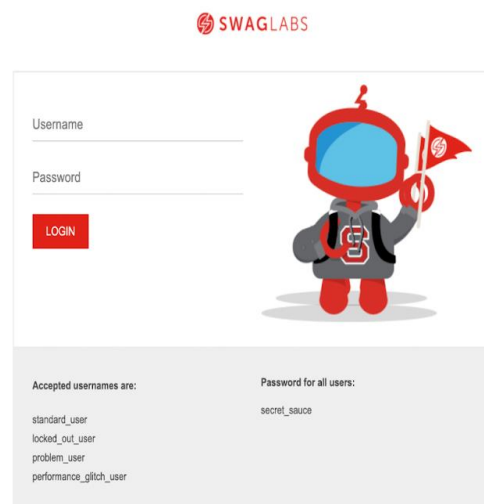


CSS Selectors – Classwork Solution

Classwork – 1 (CSS Path)

1. Find user name element using CSS selector by ID
2. Find user name element using CSS selector by className
3. Find element which has input tag and placeholder as Username
4. Find element which has input tag and placeholder as Username
5. Find the immediate next sibling of username input element using sibling mechanism
6. Then login manually using the credentials
7. Find all Add to cart button [use approach of selecting element with multiple class]
8. Find all hyperlinks (a tags) where href value ends with ".html"
9. Find hyperlink which navigates to saucelabs.com by using "saucelabs.com" as substring for href attribute



Answers:

1. `input[id='user-name']`
2. `input[class='form_input']`
3. `input[placeholder='Username']`
4. `input[placeholder='Username']`
5. `input[placeholder='Username'] + input`
- 6 --- login manually step
7. `.btn_primary.btn_inventory`
8. `a[href$='.html']`
9. `a[href*='saucelabs.com']`

Code:

```
import {test} from "@playwright/test"
test.only('login to sauce demo web app', async({page})=>{
  // 1. load the url
  await page.goto("https://www.saucedemo.com/v1/index.html")
  // 2. find the selectors
  const userNameSelector:string = "#user-name"
  const userPasswordSelector:string = "#password"
  const loginButtonSelector:string = "#login-button"

  //3. enter the username
  /**
   * i need username locator
   * how do i find the username locator?
   * by using the selector of username
   */

  await page.locator(userNameSelector).fill("standard_user")
  //3. enter the password
  await page.locator(userPasswordSelector).fill("secret_sauce")
  //4. click on login button
  await page.locator(loginButtonSelector).click()
  //5. we will make sure that the page.loaded
  await page.waitForTimeout(10000)
})

/**
 * global scopre
 * test is a fixture
 * page, contet, confifguration
 * //setup
 * launches browser
 * create context
 * const page
 *
 *
 * //teardown
 * close the context
 * saving the screenshot
 * trace
 *
 *
 */
```