1. **Design a test to check that all elements on screen are correct, such as positioning, colour, text etc.**

const { Builder, By, Key, until } = require('selenium-webdriver');

async function runTests() {

  const driver = await new Builder().forBrowser('chrome').build();

  try {

    // Part 1: Verify correct elements on the home page

    await driver.get('<https://subscription.packtpub.com/register'>);

    const headerText = await driver.fi ndElement(By.css('header')).getText();

    console.log(`Header Text: ${headerText}`);

    test('Verify Page Elements', async t =>

{

    //Checking the title of the page

    const pageTitle = await t.title;

    console.log('Page Title:', pageTitle);

    // Checking the presence and attributes of an element

    const createAccountButton = Selector('#create-account-button');

    console.log('Is Create Account Button displayed?',

   await createAccountButton.exists);

    console.log('Create Account Button Text:',

    await createAccountButton.innerText);

    //Checking CSS properties of an element

    console.log('Create Account Button Color:',

   await createAccountButton.getStyleProperty('color'));

1. **Design a test to check that the top nav bar, and their sub options go to the correct pages. Don’t worry about testing all of the options on the browse menu.**

**2a Bonus points for doing this in a loop**

// Test top nav bar and sub-options

    const topNavBarOption = await driver.findElement(By.linkText('TopNavOption'));

    await topNavBarOption.click();

    const currentPageTitle = await driver.findElement(By.css('h1')).getText();

    console.log(`Current Page Title: ${currentPageTitle}`);

1. **In the “your suggested titles” section, design automated tests to click the titles in the carousel below the main title, and ensure that they appear correctly as the main title.**

// Part 3: Test suggested titles section

    const carouselTitle = await driver.findElement(By.css('.carousel-title'));

    await carouselTitle.click();

    const mainTitle = await driver.findElement(By.css('.main-title')).getText();

    console.log(`Main Title: ${mainTitle}`);

1. **Design a test that will click browse In the top nav, then click view all books. This should take you to the browse page. Click to clear any filters that are already set, and then click to set the 2021 filter for publication date.   
   Following on from this, automate a test that will type the following words into the search bar, and check that all titles found include that search text.**

**Python**

**Paint**

**Secure**

**Tableau**

// Test browse and filter functionality

    const browseLink = await driver.findElement(By.linkText('Browse'));

    await browseLink.click();

    const viewAllBooksLink = await driver.findElement(By.linkText('View All Books'));

    await viewAllBooksLink.click();

// 2021 filter

    const filterButton = await driver.findElement(By.id('filterButton'));

    await filterButton.click();

    const year2021Filter = await driver.findElement(By.css('.filter-option[data-year="2021"]'));

    await year2021Filter.click();

    const searchBar = await driver.findElement(By.id('searchBar'));

    await searchBar.sendKeys('Test Automation', Key.RETURN);

    const searchResults = await driver.findElements(By.css('.search-result'));

    for (const result of searchResults) {

      const resultText = await result.getText();

      console.log(`Search Result: ${resultText}`);

  }

  } finally {

    await driver.quit();

  }

}

runTests();