

The Visual Testing Revolution!



Andrew Knight

Automation Panda
Applitools Developer Advocate
Test Automation University Director

Applitools

Automated Visual Testing



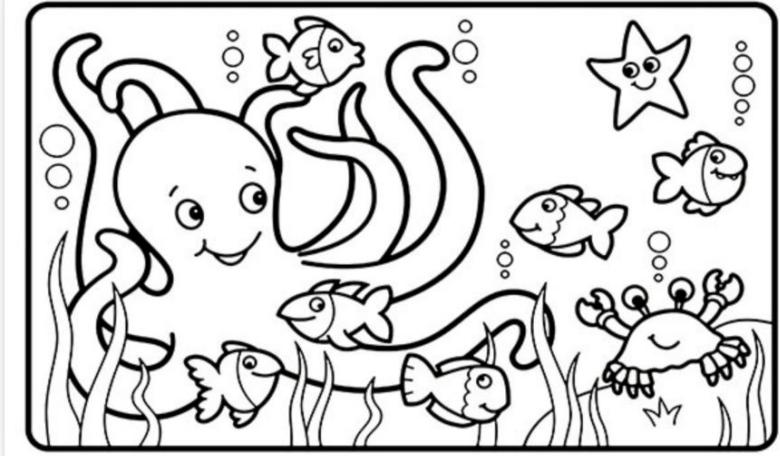
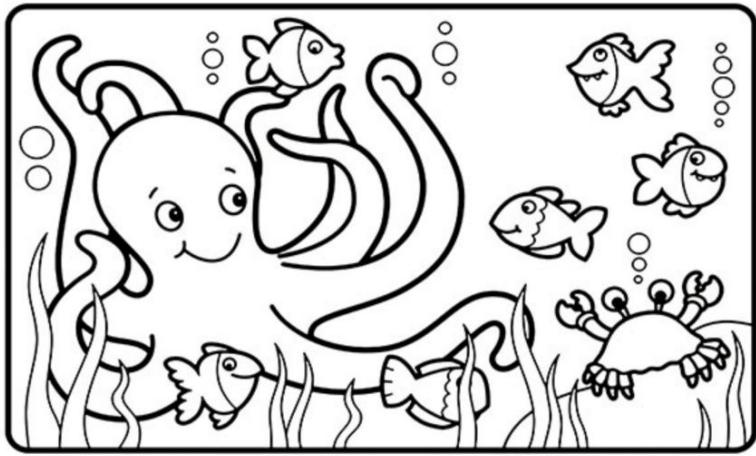
The image shows a promotional landing page for Test Automation University. At the top left is the logo 'TAU TEST AUTOMATION UNIVERSITY Powered by Applitools'. Below it is the tagline 'Become a test automation superstar!'. A large yellow button with the text 'All Courses Free!' is prominently displayed. In the center, there's a stylized graphic of a blue ocean wave at the bottom, transitioning to white land above. Two flags are planted in the land: a larger teal flag with 'TAU' and a smaller black flag with a colorful emblem.

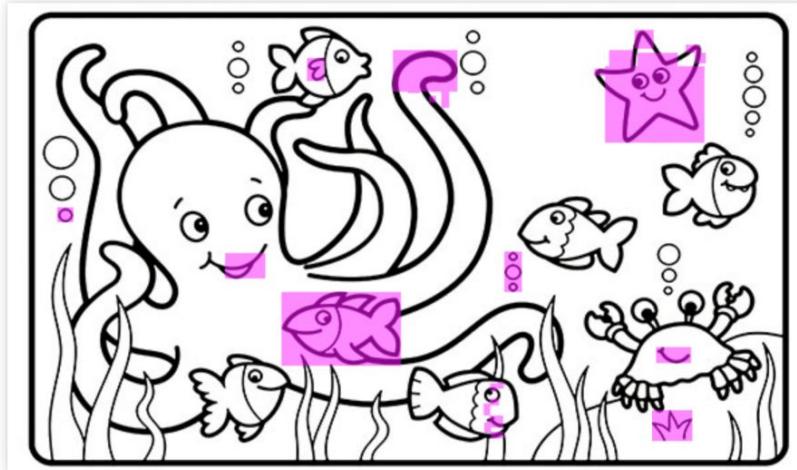
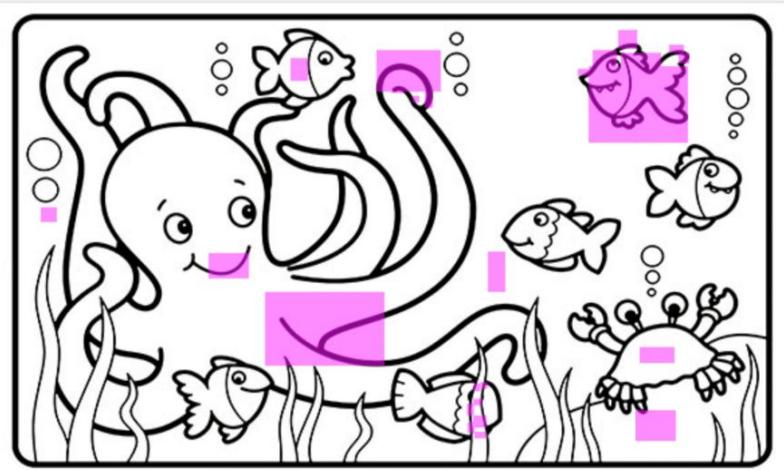
TAU TEST
AUTOMATION
UNIVERSITY
Powered by Applitools

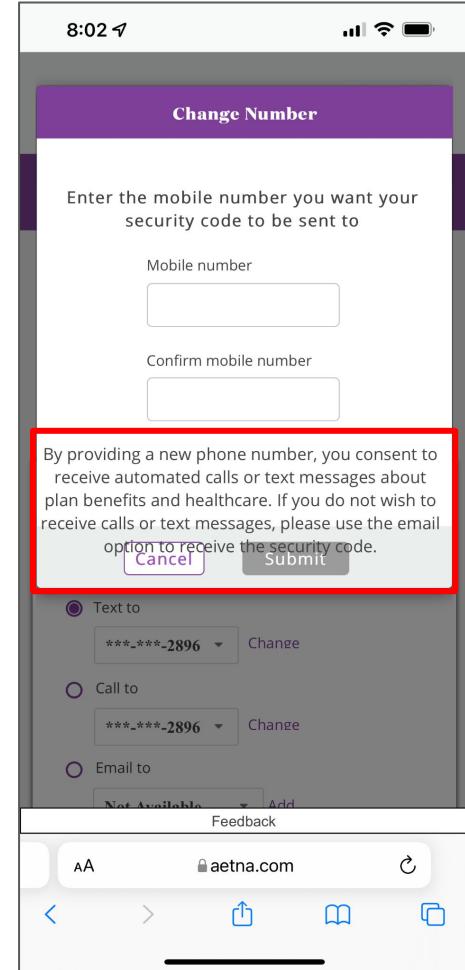
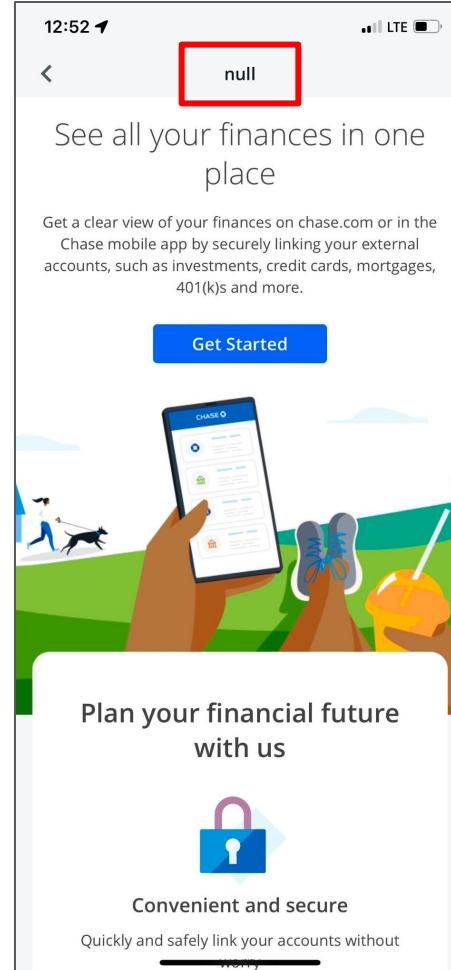
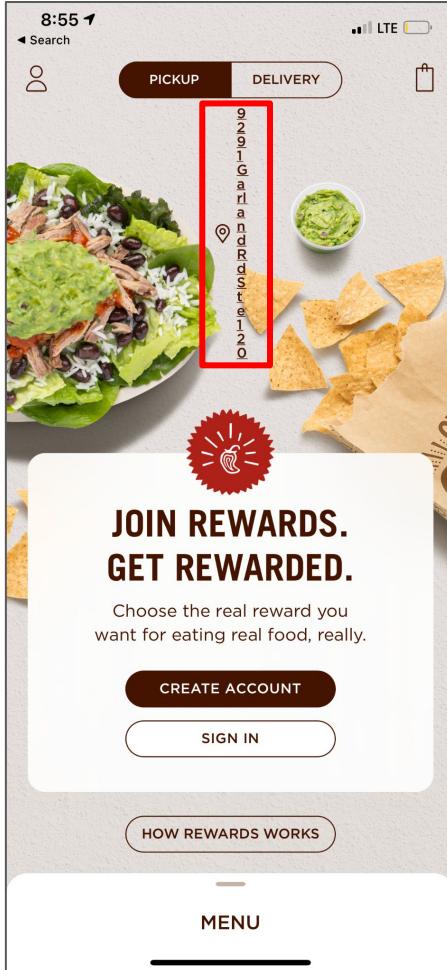
Become a test automation superstar!

All Courses Free!

testautomationu.applitools.com



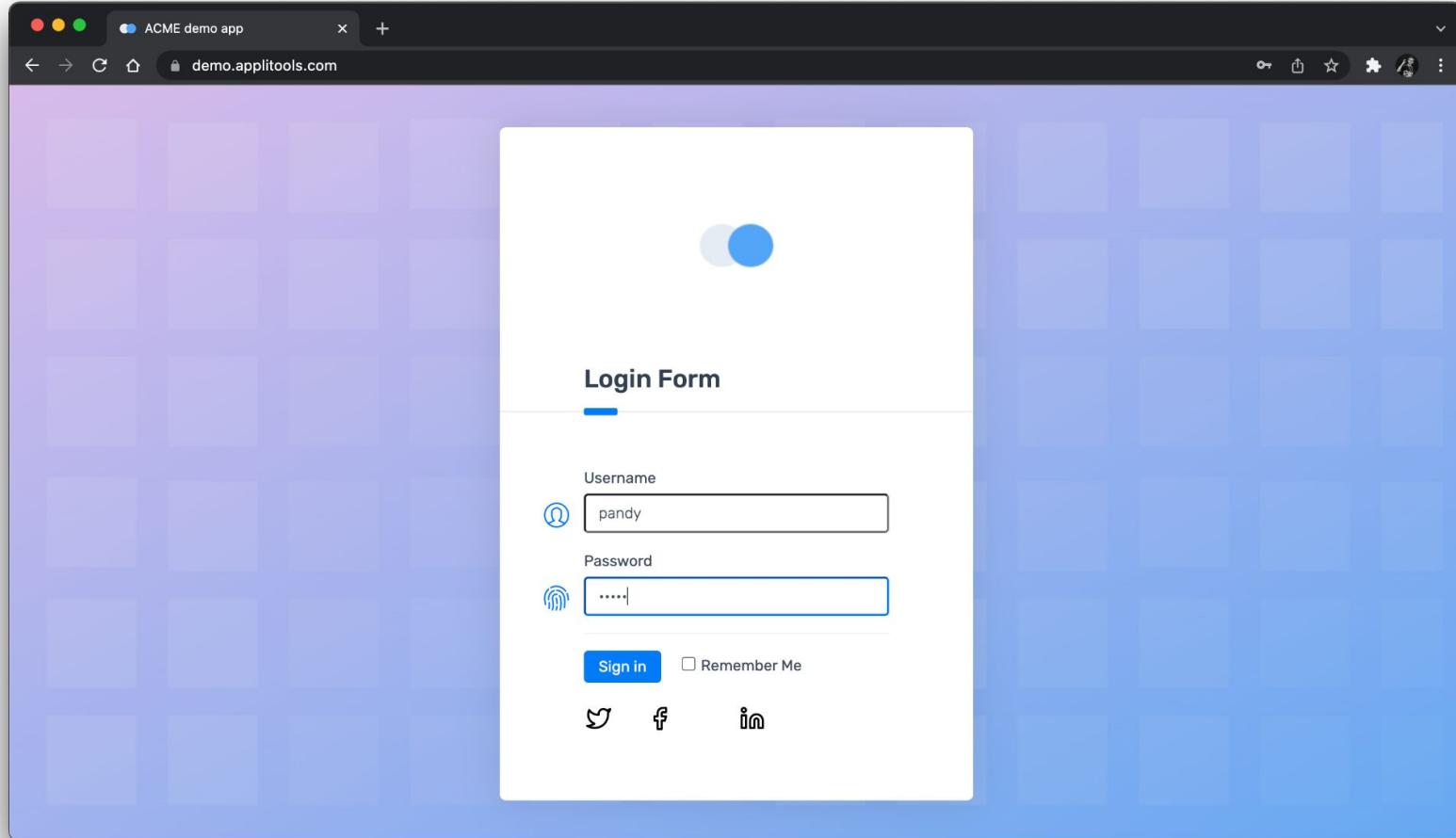




Visual Testing

Functional
~~Visual~~ Testing

**Visual testing is easier
than traditional test automation.**

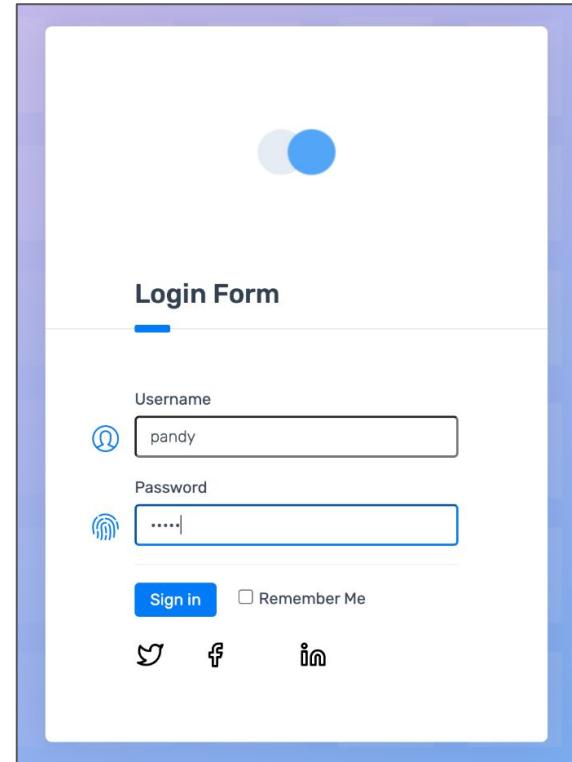


The screenshot shows a web browser window with the title "ACME demo app" and the URL "demo.applitools.com/app.html". The page has a blue header bar with the "ACME" logo and a search bar. On the left, there's a sidebar with sections for "CARD TYPES" (Credit cards, Debit cards) and "LENDING" (Loans, Mortgages). The main content area features a red banner at the top stating "Your nearest branch closes in: 30m 5s". Below it is a "Financial Overview" section with four boxes: "Total Balance" (\$350), "Credit Available" (\$17,800), "Due Today" (\$180), and buttons for "View Statement", "Request Increase", and "Pay Now". The "Recent Transactions" section lists four entries:

STATUS	DATE	DESCRIPTION	CATEGORY	AMOUNT
● Complete	Today 1:52am	Starbucks coffee	Restaurant / Cafe	+ 1,250 USD
● Declined	Jan 19th 3:22pm	stripe Stripe Payment Processing	Finance	+ 952.23 USD
● Pending	Yesterday 7:45am	MailChimp Services	Software	- 320 USD
● Pending	Jan 23rd 2:27pm	Shopify product	Shopping	+ 17.99 USD

Login test case

1. Load the login page.
2. Verify that the login page loads correctly.
3. Log into the app.
4. Verify that the main page loads correctly.



```
public class LoginTest {  
  
    private WebDriver driver;  
    private WebDriverWait wait;  
  
    @Test  
    public void login() {  
        loadLoginPage();  
        verifyLoginPage();  
        performLogin();  
        verify MainPage();  
    }  
  
    // ...  
}
```

```
private void loadLoginPage() {  
    driver.get("https://demo.applitools.com");  
}
```

```
private void waitForAppearance(By locator) {  
    wait.until(d -> d.findElements(locator).size() > 0);  
}  
  
private void verifyLoginPage() {  
    waitForAppearance(By.cssSelector("div.logo-w"));  
    waitForAppearance(By.id("username"));  
    waitForAppearance(By.id("password"));  
    waitForAppearance(By.id("log-in"));  
    waitForAppearance(By.cssSelector("input.form-check-input"));  
}
```

```
private void performLogin() {  
    driver.findElement(By.id("username")).sendKeys("andy");  
    driver.findElement(By.id("password")).sendKeys("i<3pandas");  
    driver.findElement(By.id("log-in")).click();  
}
```

```

private void verify MainPage() {
    // Check various page elements
    waitForAppearance(By.cssSelector("div.logo-w"));
    waitForAppearance(By.cssSelector("div.element-search.autosuggest-search-activator > input"));
    waitForAppearance(By.cssSelector("div.avatar-w img"));
    waitForAppearance(By.cssSelector("ul.main-menu"));
    waitForAppearance(By.xpath("//a/span[.='Add Account']"));
    waitForAppearance(By.xpath("//a/span[.='Make Payment']"));
    waitForAppearance(By.xpath("//a/span[.='View Statement']"));
    waitForAppearance(By.xpath("//a/span[.='Request Increase']"));
    waitForAppearance(By.xpath("//a/span[.='Pay Now']"));

    // Check time message
    assertTrue(Pattern.matches(
        "Your nearest branch closes in:( \\\d+[hms])+",
        driver.findElement(By.id("time")).getText()));

    // Check menu element names
    var menuElements = driver.findElements(By.cssSelector("ul.main-menu li span"));
    var menuItems = menuElements.stream().map(i -> i.getText().toLowerCase()).toList();
    var expected = Arrays.asList("card types", "credit cards", "debit cards", "lending", "loans", "mortgages");
    assertEquals(expected, menuItems);

    // Check transaction statuses
    var statusElements = driver.findElements(By.xpath("//td[./span[contains(@class, 'status-pill')]]/span[2]"));
    var statusNames = statusElements.stream().map(n -> n.getText().toLowerCase()).toList();
    var acceptableNames = Arrays.asList("complete", "pending", "declined");
    assertTrue(acceptableNames.containsAll(statusNames));
}

```

```

private void verify MainPage() {
    // Check various page elements
    waitForAppearance(By.cssSelector("div.logo-w"));
    waitForAppearance(By.cssSelector("div.element-search.autosuggest-search-activator > input"));
    waitForAppearance(By.cssSelector("div.avatar-w img"));
    waitForAppearance(By.cssSelector("ul.main-menu"));
    waitForAppearance(By.xpath("//a/span[.='Add Account']"));
    waitForAppearance(By.xpath("//a/span[.='Make Payment']"));
    waitForAppearance(By.xpath("//a/span[.='View Statement']"));
    waitForAppearance(By.xpath("//a/span[.='Request Increase']"));
    waitForAppearance(By.xpath("//a/span[.='Pay Now']"));

    // Check time message
    assertTrue(Pattern.matches(
        "Your nearest branch closes in:( \\\d+[hms])+",
        driver.findElement(By.id("time")).getText()));

    // Check menu element names
    var menuElements = driver.findElements(By.cssSelector("ul.main-menu li span"));
    var menuItems = menuElements.stream().map(i -> i.getText().toLowerCase()).toList();
    var expected = Arrays.asList("card types", "credit cards", "debit cards", "lending", "loans", "mortgages");
    assertEquals(expected, menuItems);

    // Check transaction statuses
    var statusElements = driver.findElements(By.xpath("//td[./span[contains(@class, 'status-pill')]]/span[2]"));
    var statusNames = statusElements.stream().map(n -> n.getText().toLowerCase()).toList();
    var acceptableNames = Arrays.asList("complete", "pending", "declined");
    assertTrue(acceptableNames.containsAll(statusNames));
}

```

```

private void verify MainPage() {
    // Check various page elements
    waitForAppearance(By.cssSelector("div.logo-w"));
    waitForAppearance(By.cssSelector("div.element-search.autosuggest-search-activator > input"));
    waitForAppearance(By.cssSelector("div.avatar-w img"));
    waitForAppearance(By.cssSelector("ul.main-menu"));
    waitForAppearance(By.xpath("//a/span[.='Add Account']"));
    waitForAppearance(By.xpath("//a/span[.='Make Payment']"));
    waitForAppearance(By.xpath("//a/span[.='View Statement']"));
    waitForAppearance(By.xpath("//a/span[.='Request Increase']"));
    waitForAppearance(By.xpath("//a/span[.='Pay Now']"));

    // Check time message
    assertTrue(Pattern.matches(
        "Your nearest branch closes in: ( \\d+[hms])+",
        driver.findElement(By.id("time")).getText()));

    // Check menu element names
    var menuElements = driver.findElements(By.cssSelector("ul.main-menu li span"));
    var menuItems = menuElements.stream().map(i -> i.getText().toLowerCase()).toList();
    var expected = Arrays.asList("card types", "credit cards", "debit cards", "lending", "loans", "mortgages");
    assertEquals(expected, menuItems);

    // Check transaction statuses
    var statusElements = driver.findElements(By.xpath("//td[./span[contains(@class, 'status-pill')]]/span[2]"));
    var statusNames = statusElements.stream().map(n -> n.getText().toLowerCase()).toList();
    var acceptableNames = Arrays.asList("complete", "pending", "declined");
    assertTrue(acceptableNames.containsAll(statusNames));
}

```

```

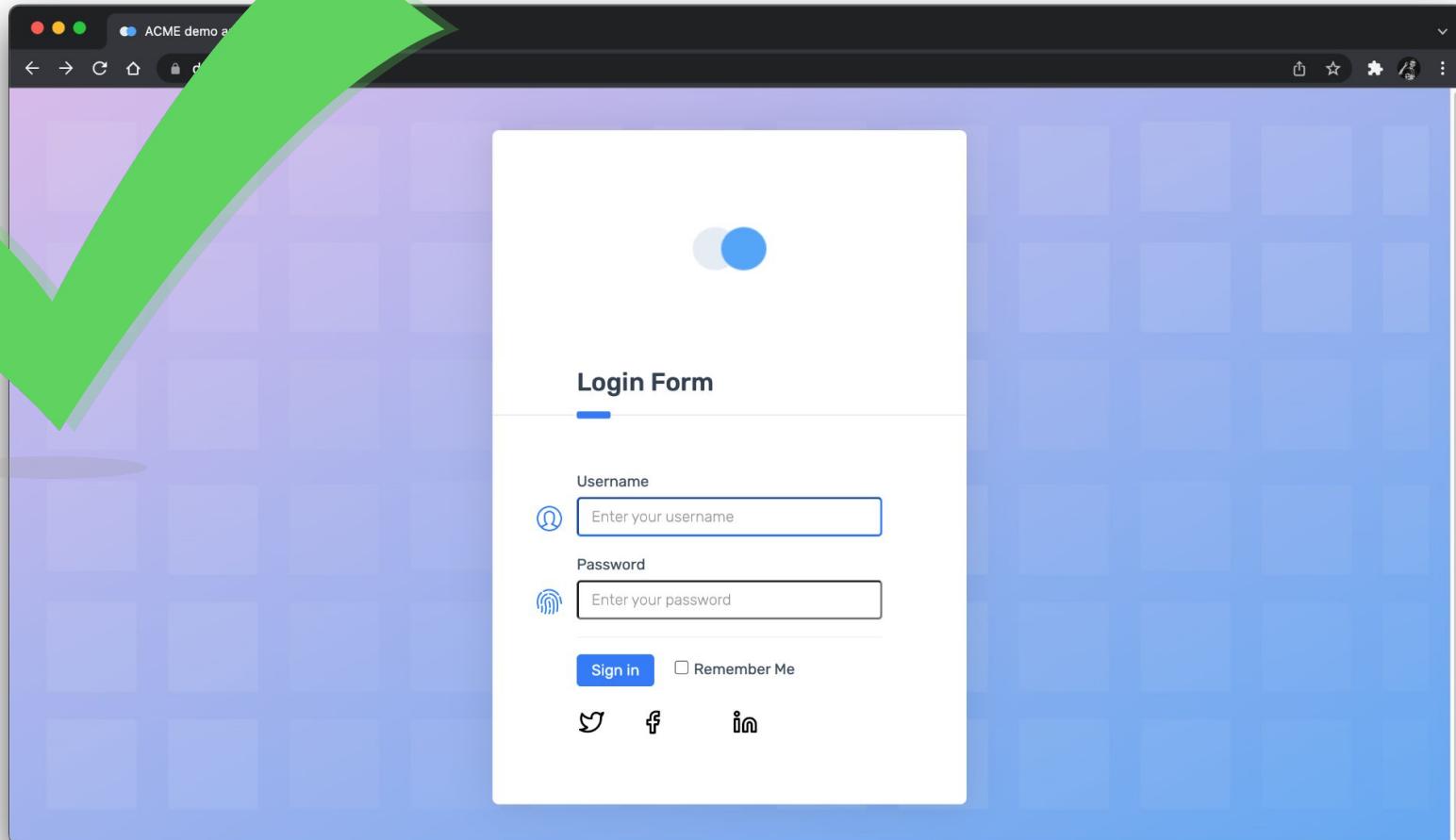
private void verify MainPage() {
    // Check various page elements
    waitForAppearance(By.cssSelector("div.logo-w"));
    waitForAppearance(By.cssSelector("div.element-search.autosuggest-search-activator > input"));
    waitForAppearance(By.cssSelector("div.avatar-w img"));
    waitForAppearance(By.cssSelector("ul.main-menu"));
    waitForAppearance(By.xpath("//a/span[.='Add Account']"));
    waitForAppearance(By.xpath("//a/span[.='Make Payment']"));
    waitForAppearance(By.xpath("//a/span[.='View Statement']"));
    waitForAppearance(By.xpath("//a/span[.='Request Increase']"));
    waitForAppearance(By.xpath("//a/span[.='Pay Now']"));

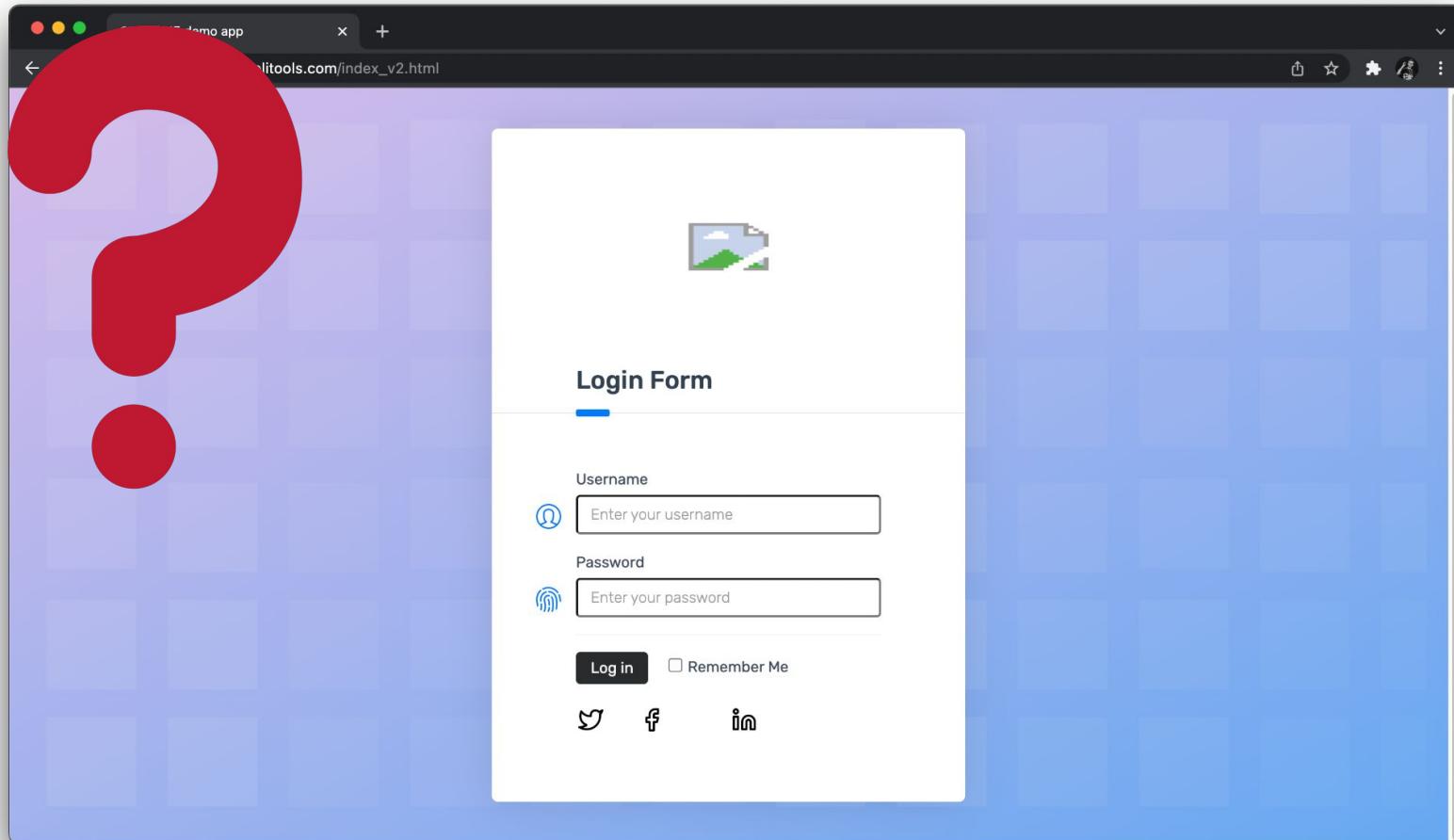
    // Check time message
    assertTrue(Pattern.matches(
        "Your nearest branch closes in:( \\\d+[hms])+",
        driver.findElement(By.id("time")).getText()));

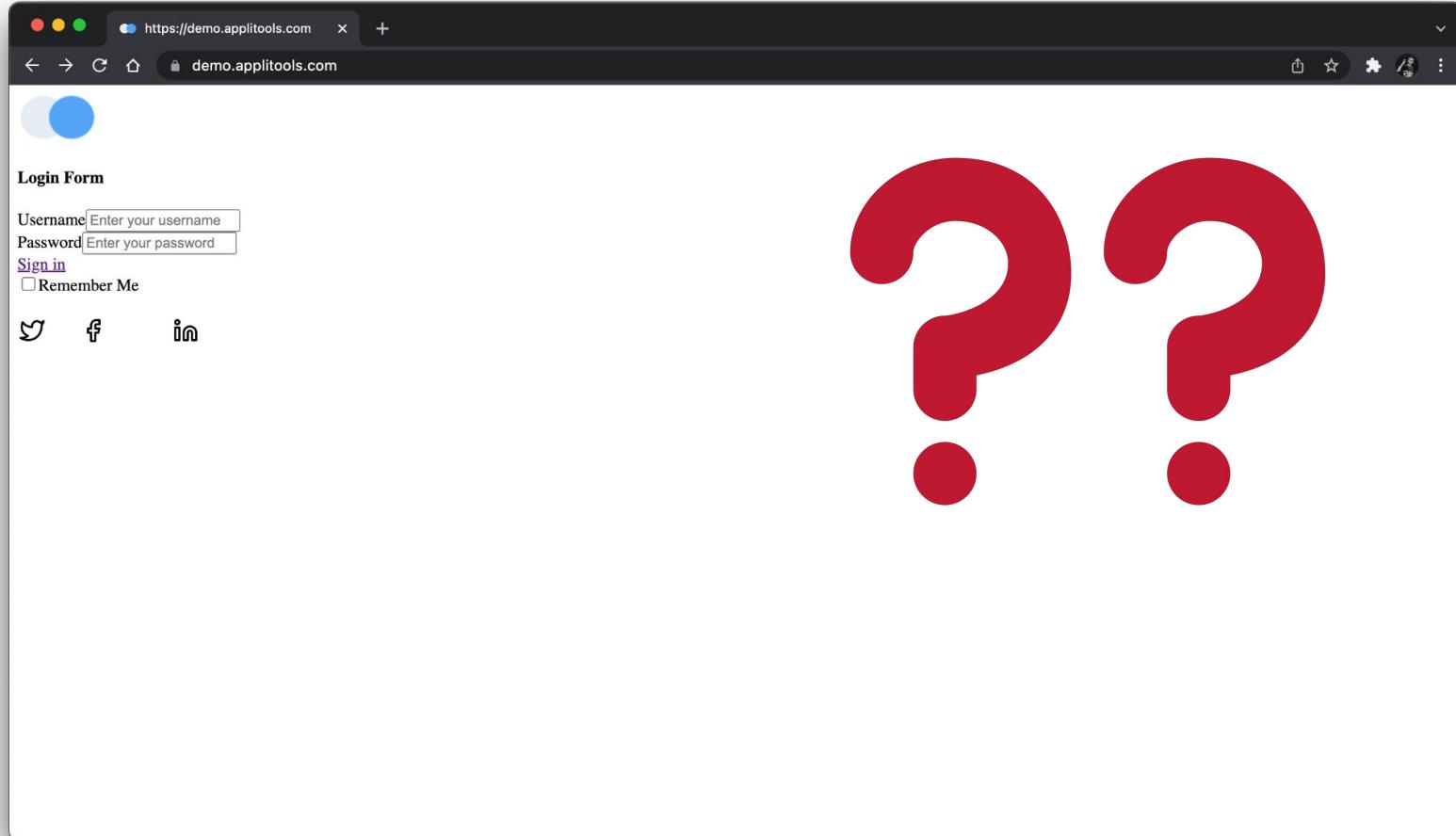
    // Check menu element names
    var menuElements = driver.findElements(By.cssSelector("ul.main-menu li span"));
    var menuItems = menuElements.stream().map(i -> i.getText().toLowerCase()).toList();
    var expected = Arrays.asList("card types", "credit cards", "debit cards", "lending", "loans", "mortgages");
    assertEquals(expected, menuItems);

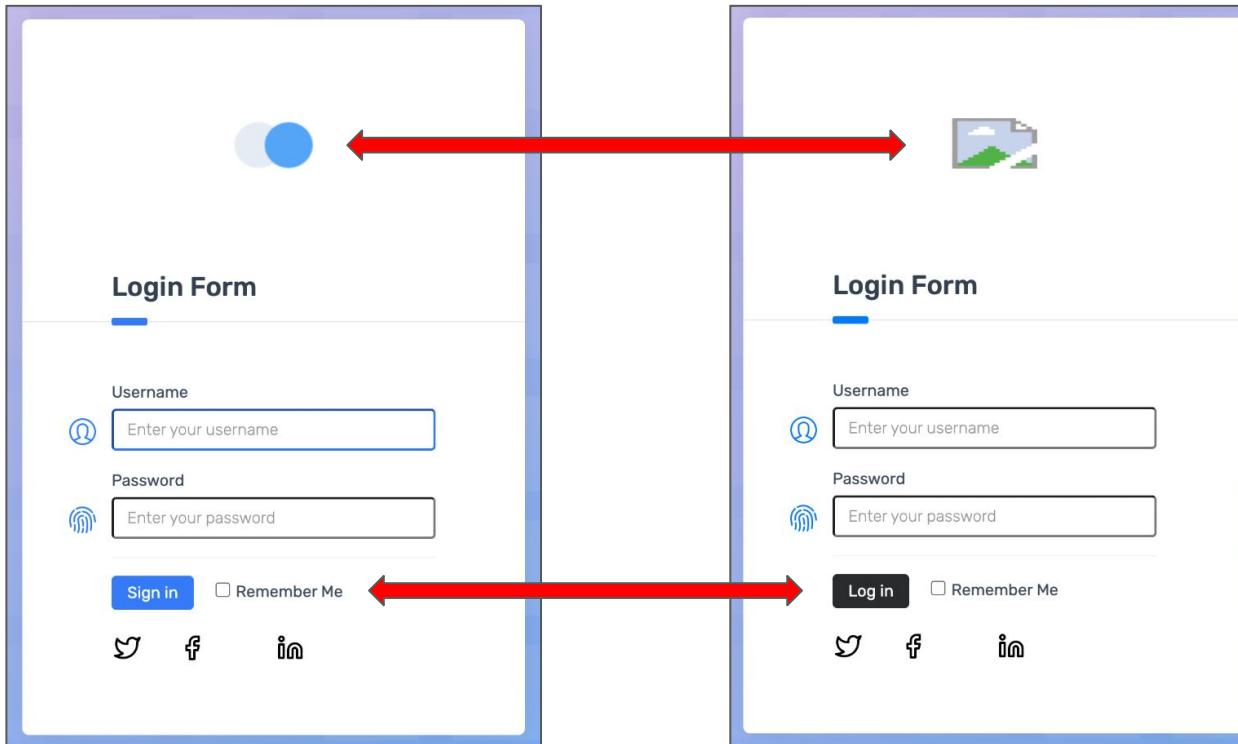
    // Check transaction statuses
    var statusElements = driver.findElements(By.xpath("//td[./span[contains(@class, 'status-pill')]]/span[2]"));
    var statusNames = statusElements.stream().map(n -> n.getText().toLowerCase()).toList();
    var acceptableNames = Arrays.asList("complete", "pending", "declined");
    assertTrue(acceptableNames.containsAll(statusNames));
}

```









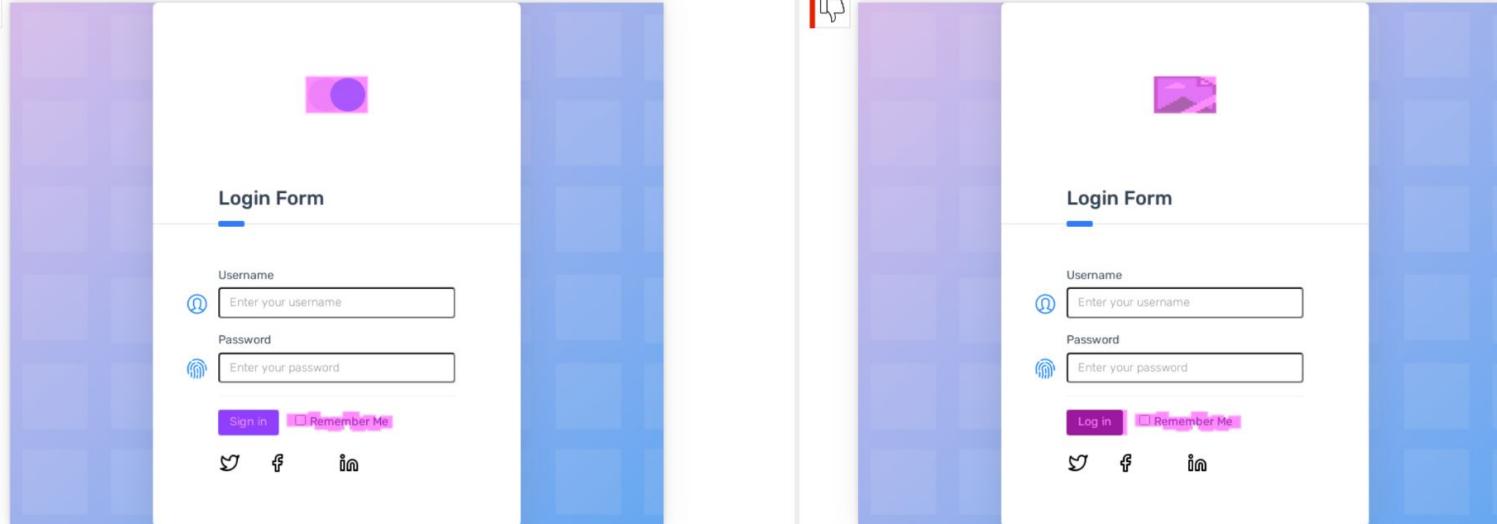
Step 1/2: Login page

Test: A visual login test

VIEW HIGHLIGHT DIFFS ANNOTATIONS AUTO MAINTENANCE ACCESSIBILITY

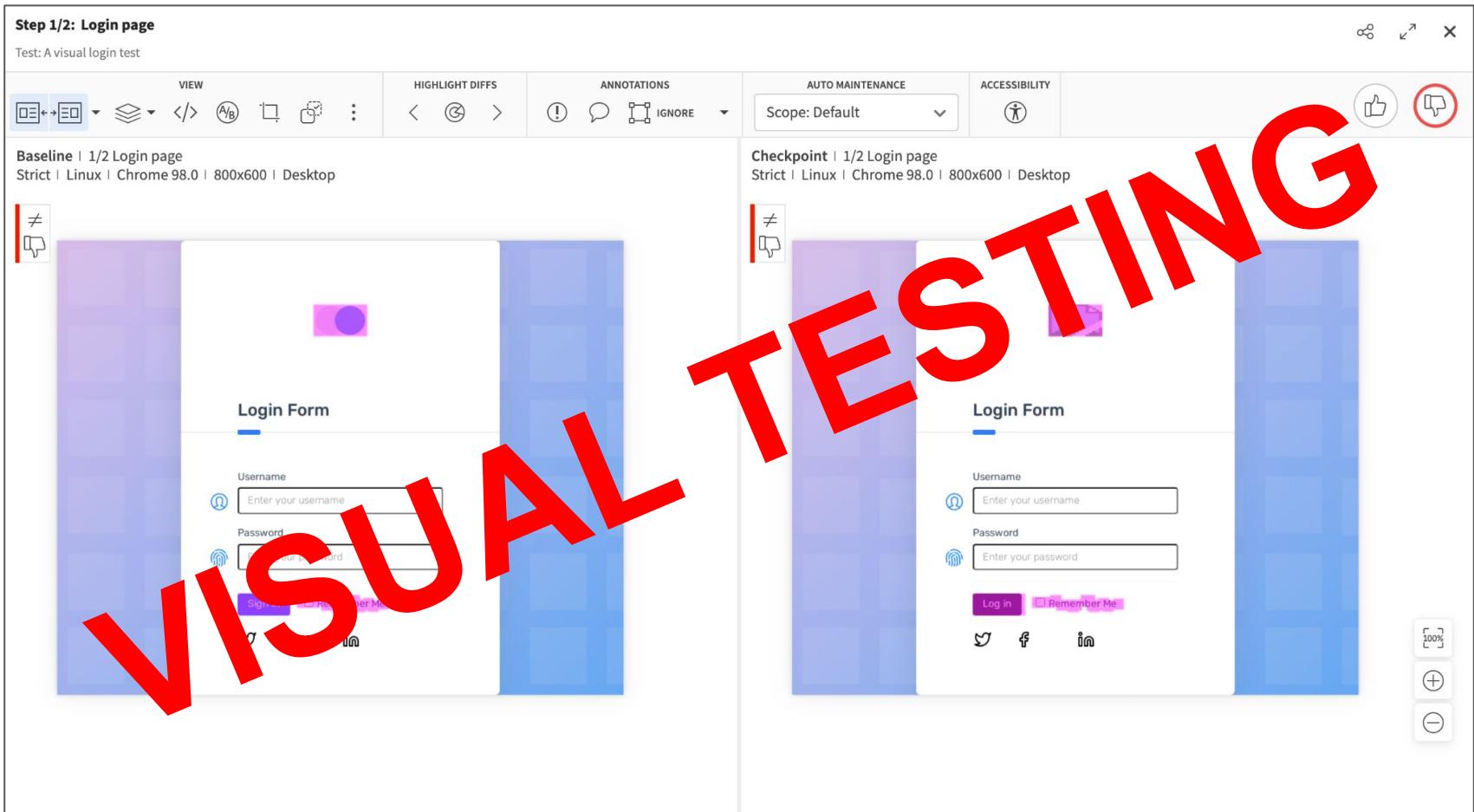
Baseline | 1/2 Login page
Strict | Linux | Chrome 98.0 | 800x600 | Desktop

Checkpoint | 1/2 Login page
Strict | Linux | Chrome 98.0 | 800x600 | Desktop



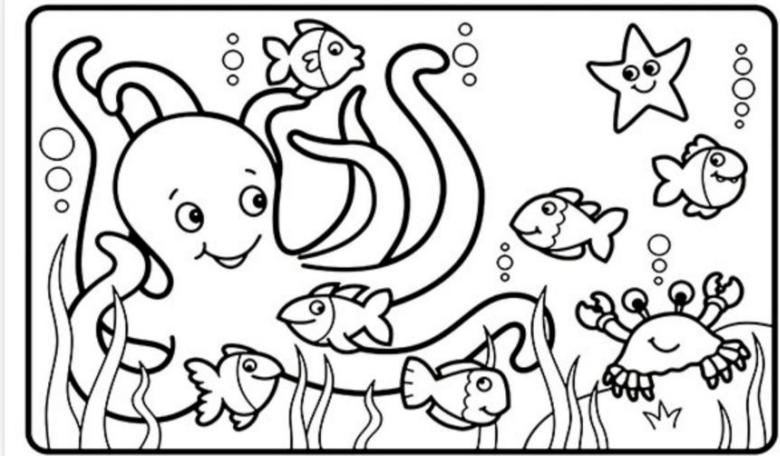
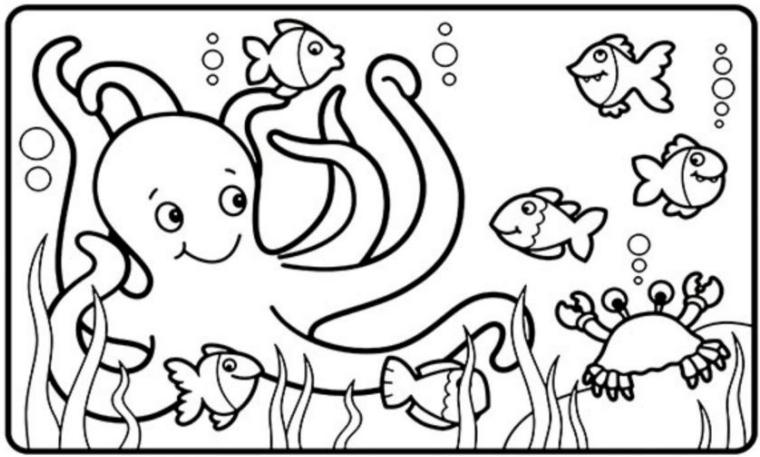
Annotations:  

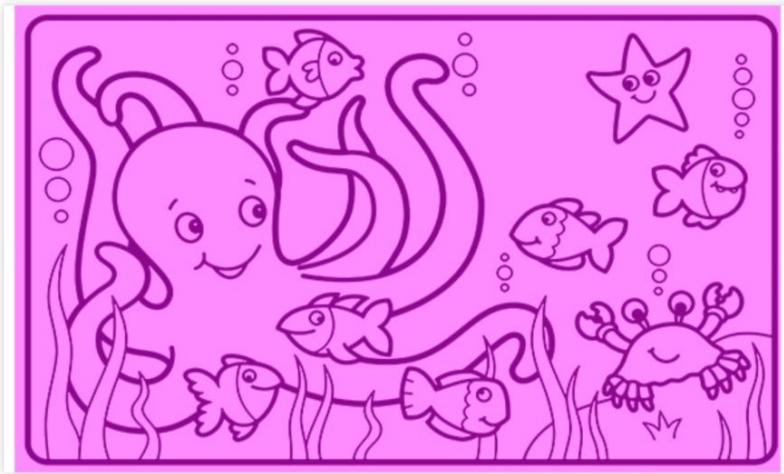
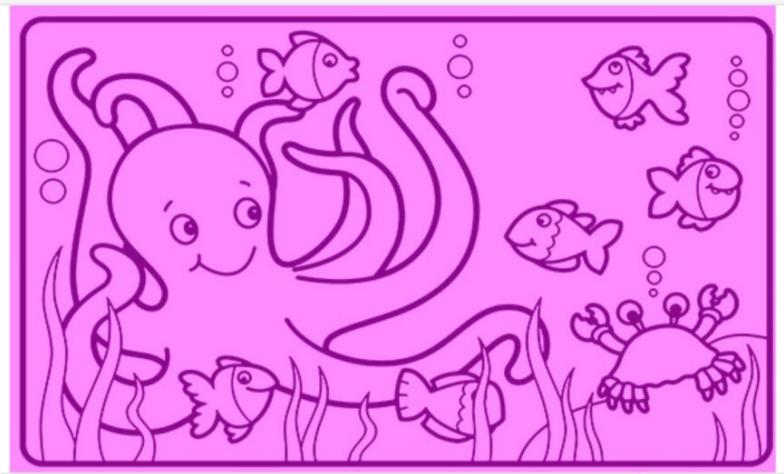
Annotations:   



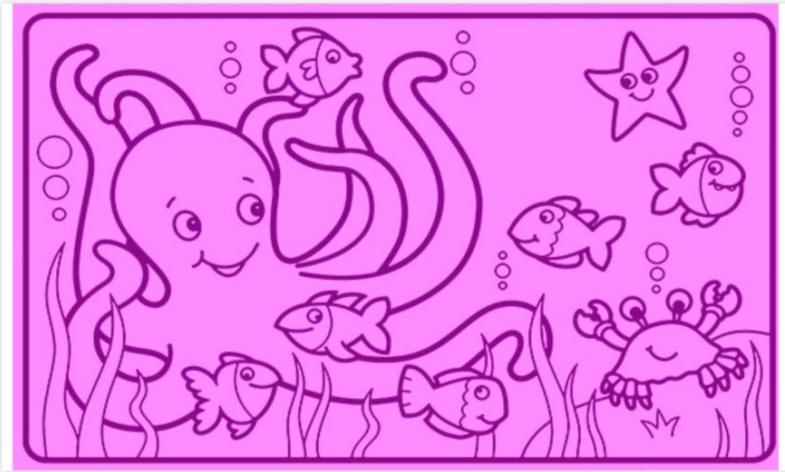
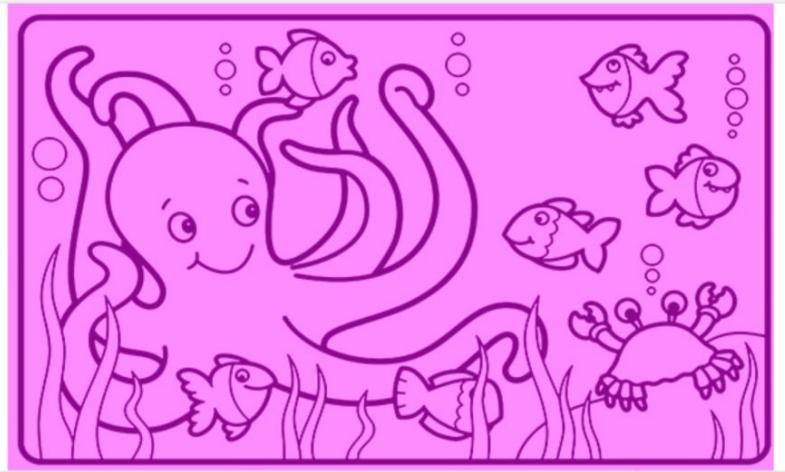
Advantage #1:

Visual testing covers
everything on a page.

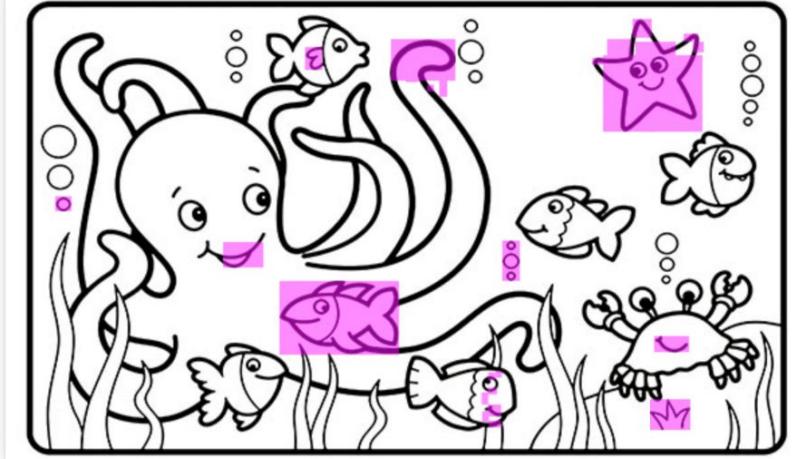
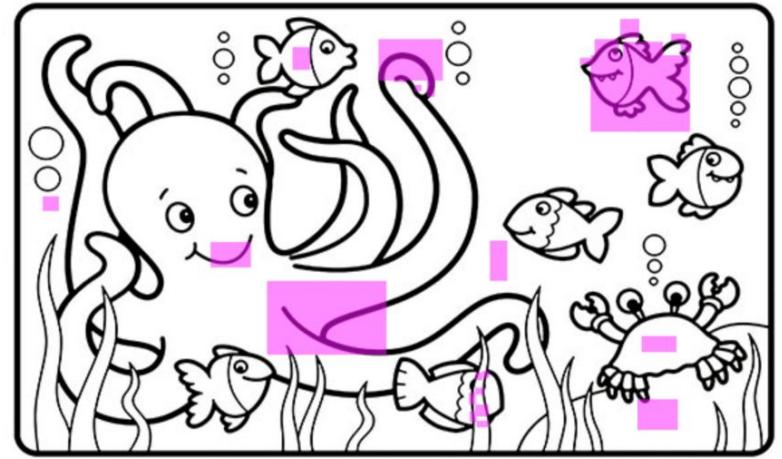




Pixel-to-Pixel (“Exact”) Matching



Visual AI Matching



Advantage #2:

Visual AI focuses on
meaningful changes to avoid noise.

Sign up | AppliTools

auth.appliTools.com/users/register

appliTools

VISUALLY PERFECT

Visually test all your apps

- Find visual bugs across browsers, screen sizes, mobile devices, and operating systems
- Use AI to eliminate false positives that frustrate your team
- One call to our API eliminates multiple checkpoints in your test scripts
- Supports over 40 testing tools: Selenium, Appium, and many more

Questions?

Talk to a visual testing expert at +1 650 680 1000

Trusted by hundreds of companies

intuit  SAP

Create your free account

No obligation. No credit card required.

GITHUB SIGNUP

or

First name Last name
andyipk247@gmail.com
Your API key will be sent here

.....

Company

Country

By creating an account, you agree to our terms of service and acknowledge your data will be used in accordance with AppliTools' privacy policy.

[CREATE ACCOUNT](#)

 Privacy - Terms

```
<dependency>
    <groupId>com.applitools</groupId>
    <artifactId>eyes-selenium-java3</artifactId>
    <version>3.211.0</version>
    <scope>test</scope>
</dependency>
```

```
private WebDriver driver;
private VisualGridRunner runner;
private Eyes eyes;

@BeforeEach
public void setUpVisualAI() {
    driver = new ChromeDriver();
    runner = new VisualGridRunner();
    eyes = new Eyes(runner);

    Configuration config = eyes.getConfiguration();
    config.setApiKey(System.getenv("APPLITOOLS API KEY"));
    config.setBatch(new BatchInfo("A Visual Testing Revolution"));
    config.addBrowser(800, 600, BrowserType.CHROME);

    eyes.setConfiguration(config);
}
```

```
@Test
public void login() {
    try {
        eyes.open(driver, "Applitools Demo App", "A visual login test");

        loadLoginPage();
        verifyLoginPage();
        performLogin();
        verify MainPage();

        eyes.closeAsync();
    }
    finally {
        eyes.abortAsync();
    }
}
```

```
private void verifyLoginPage() {
    // waitForAppearance(By.cssSelector("div.logo-w")) ;
    // waitForAppearance(By.id("username")) ;
    // waitForAppearance(By.id("password")) ;
    // waitForAppearance(By.id("log-in")) ;
    // waitForAppearance(By.cssSelector("input.form-check-input")) ;

    eyes.check(Target.window().fully().withName("Login page"));
}
```

```
private void verifyMainPage() {
    // Check various page elements
    waitForAppearance(By.cssSelector("div.logo-w"));
    waitForAppearance(By.cssSelector("div.element-search.autosuggest-search-activator > input"));
    waitForAppearance(By.cssSelector("div.avatar-w img"));
    waitForAppearance(By.cssSelector("ul.main-menu"));
    waitForAppearance(By.xpath("//a/span[.='Add Account']"));
    waitForAppearance(By.xpath("//a/span[.='Make Payment']"));
    waitForAppearance(By.xpath("//a/span[.='View Statement']"));
    waitForAppearance(By.xpath("//a/span[.='Request Increase']"));
    waitForAppearance(By.xpath("//a/span[.='Pay Now']"));

    // Check time message
    assertTrue(Pattern.matches(
        "Your nearest branch closes in: ( \\d+[hms])+",
        driver.findElement(By.id("time")).getText()));

    // Check menu element names
    var menuElements = driver.findElements(By.cssSelector("ul.main-menu li span"));
    var menuItems = menuElements.stream().map(i -> i.getText().toLowerCase()).toList();
    var expected = Arrays.asList("card types", "credit cards", "debit cards", "lending", "loans", "mortgages");
    assertEquals(expected, menuItems);

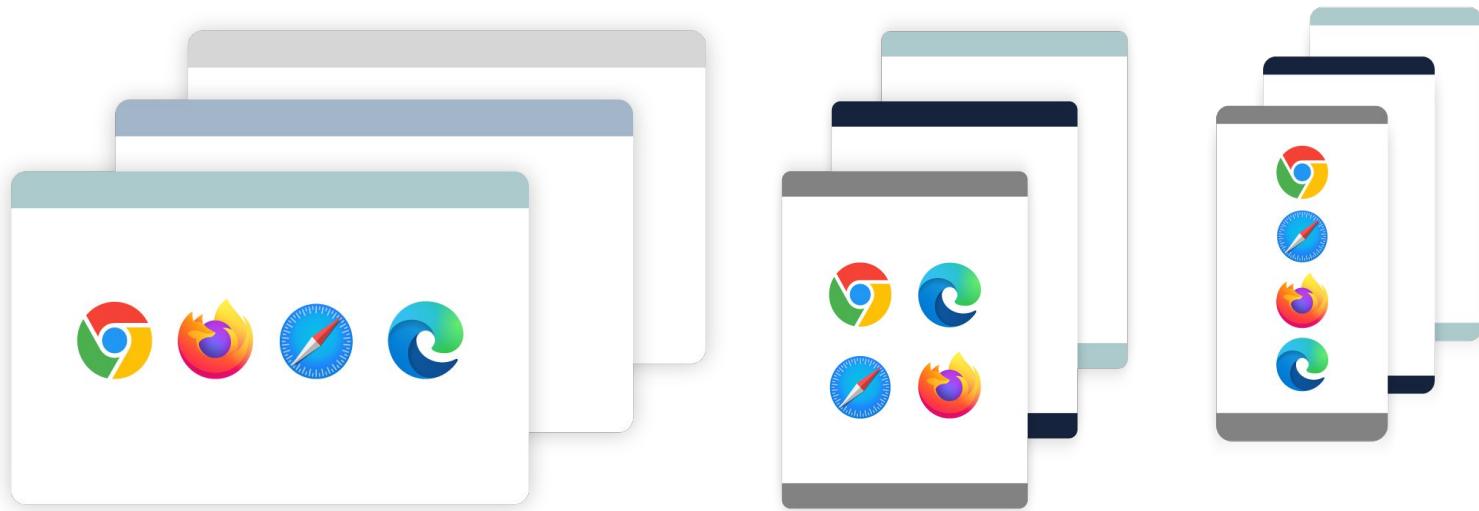
    // Check transaction statuses
    var statusElements = driver.findElements(By.xpath("//td[./span[contains(@class, 'status-pill')]]/span[2]"));
    var statusNames = statusElements.stream().map(n -> n.getText().toLowerCase()).toList();
    var acceptableNames = Arrays.asList("complete", "pending", "declined");
    assertTrue(acceptableNames.containsAll(statusNames));
}
```

```
eyes.check(Target.window().fully()
    .withName("Login page"));
```

Advantage #3:

A **snapshot** is worth a thousand assertions.

Cross-Browser and Cross-Device Testing

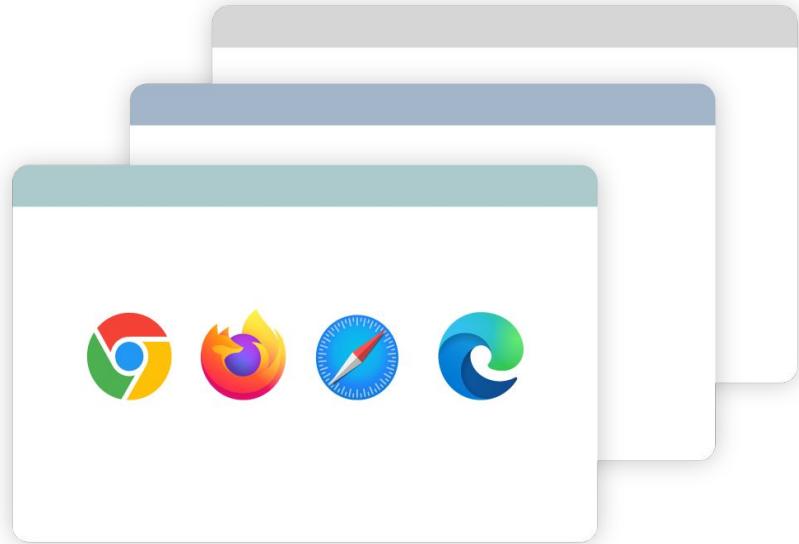




Screenshot



Snapshot

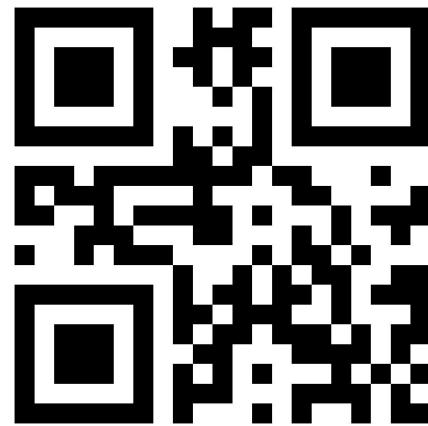


Advantage #4:

Visual snapshots enable lightning-fast
cross-platform testing.

Let's demo!

<https://github.com/AutomationPanda/visual-testing-revolution>



When should a team adopt visual testing?

From the start.

Advantage #5:

Visual testing makes
functional testing **easier.**



Let's keep in touch!

My Blog

AutomationPanda.com



My Twitter

@AutomationPanda



My LinkedIn

andrew-leland-knight



My Puppy

IG: suki_thefrenchie_



5 Key Advantages of Visual Testing

1. Visual testing covers **everything** on a page.
 2. Visual AI focuses on **meaningful changes** to avoid noise.
 3. A **snapshot** is worth a thousand assertions.
 4. Visual snapshots enable lightning-fast **cross-platform testing**.
 5. Visual testing makes functional testing **easier**.
-



<https://automationpanda.com/2022/06/01/modernizing-software-quality-assurance-with-visual-testing/>

<https://github.com/AutomationPanda/visual-testing-revolution>

