

**કૉમ્પ્યુટર વિજ્ઞાન વિભાગ**

**વ્યવસ્થાપન અને પ્રોદ્યોગિકી વિજ્ઞાન વિદ્યાશાખા**

**ગૂજરાત વિદ્યાપીઠ, અમદાવાદ – ૩૮૦૦૦૯**

**UI & API Automation Excercise**

**પ્રસ્તુતકર્તા**

(**પરીક્ષા નંબર – 212308003)**

**એમ.સી.એ. વર્ષ -** ૨ **સત્ર –** 4

**એમ.સી.એ સત્ર-4 પ્રશ્નપત્ર નં. -MCA-304ના અભ્યાસક્રમના ભાગ રૂપે તૈયાર કરેલ**

**પરિયોજના અહેવાલ વર્ષ -2024-25**

**માર્ગદર્શક**

**આંતરિક માર્ગદર્શક  
શ્રીમતી રુષિતા શાહ**

**કમ્પ્યુટર વિજ્ઞાન વિભાગ** **ગૂજરાત વિદ્યાપીઠ,**

**અમદાવાદ- 380009.**

**બાહ્ય માર્ગદર્શક**

**શ્રીમતી કૃપાલી ચાવડા   
 Manager of DeepCoder Pvt. Ltd Thaltej Cross, Road,અમદાવાદ-** **380054**

**Acknowledgement**

ગૂજરાત વિદ્યાપીઠમાં આવેલ કોમ્પ્યુટર વિજ્ઞાન વિભાગમાં કોમ્પ્યુટર વિનીયોગ પારંગત (એમ.સી.એ.) વર્ષઃ૨ સત્રઃ4 ના અભ્યાસક્રમના ભાગરૂપે **“UI & Automation Practice”** બનાવેલ છે.

અધ્યાપક ડૉ. અજય પરીખ ના માર્ગદર્શન અને તેમના અનુભવની મદદ દ્વારા **“UI & API Automation Practice”**નું કાર્ય કરેલ છે. તેમના માર્ગદર્શન હેઠળ ઘણુ શીખવા મળેલ છે. તેમણે આપેલા કિંમતી સમય આપવા બદલ અમે તેમ નો આભાર માનીએ છિયે

**NuWave Communicatins** સંસ્થાના શ્રી ના માર્ગદર્શન માટે આભાર માનીએ છિયે.

પરિયોજનાના કાર્યની તક આપવા બદલ માટે અમે કોમ્પુટર વિજ્ઞાન વિભાગનો આભાર માનીએ છિયે. જાણે અજાણે આ પરિયોજનના કાર્યમાં મને મદદ કરનાર દરેક જણનો અમે આભાર માનીએ છિયે.

**અનુક્રમણિકા**

|  |  |  |
| --- | --- | --- |
| **ક્રમ** | **વિગતો** | **પૃષ્ઠ નંબર** |
| **1** | **ગૂજરાત વિદ્યાપીઠનો પરિચય** |  |
| **2** | **કંપનીનો પરિચય** |  |
| **3** | **પ્રસ્તાવના** |  |
| **5** | **Project Profile** |  |
| **6** | **હાલની સિસ્ટમ** |  |
| **7** | **Project Flow Diagram** |  |
| **8** | **Backend Flow** |  |
| **9** | **Project Test Flow Diagram** |  |
| **10** | **Test Scenarios** |  |
| **11** | **Test Case** |  |
| **12** | **API Testing (Postman)** |  |
| **13** | **Bug Sheet** |  |
| **14** | **Test Scripts (Java Selenium)** |  |
| **15** | **Bug Images** |  |

**ગૂજરાત વિદ્યાપીઠનો પરિચય**

ગૂજરાત વિદ્યાપીઠની સ્થાપના રાષ્ટ્રપિતા મહાત્મા ગાંધીજી દ્વારા ઇ.સ.૧૯૨૦ માં કરવામાં આવી હતી. ગૂજરાત વિદ્યાપીઠની સ્થાપના “રાષ્ટ્રીય વિદ્યાપીઠ” તરીકે થઇ હતી. મહાત્મા ગાંધીજી તેમના જીવન પર્યન્ત વિદ્યાપીઠના કુલાધિપતિ રહયા હતા. હાલ ગૂજરાત વિદ્યાપીઠના કુલાધિપતિ તરીકે શ્રી. આચાર્ય દેવવ્રત કાર્યરત છે.

ગૂજરાત વિદ્યાપીઠમાં પ્રાર્થના, સમૂહજીવન, અંબર-કાંતણ, પદયાત્રાને પણ અભ્યાસના આધારભૂત સ્ત્રોત ગણીને અભ્યાસમાં આવરી લેવામાં આવેલ છે. અને સમગ્ર વિશ્વમાં પોતાની એક આગવી ઓળખ ઉભી કરવાના રસ્તે હમેશા તત્પર રહેતી સંસ્થા છે.

ગૂજરાત વિદ્યાપીઠમાં વિવિધ વિભાગ જેમાં કૃષિ વિજ્ઞાન કેન્દ્ર, આદિવાસી ભવન, આરોગ્ય કેન્દ્ર, બાઈબલ ખંડ, ગ્રામસેવા કેન્દ્ર, ભાષા ભવન, અનુસ્નાતક વિભાગનો સમાવેશ કરવામા આવ્યો છે. સમુહજીવન, કાંતણ, પદયાત્રા એ ગૂજરાત વિદ્યાપીઠની વિશેષ પ્રવૃતિઓ છે. ખાદીનો પહેરવેશ એ ગૂજરાત વિદ્યાપીઠની એક આગવી ઓળખ છે.

**કંપનીનો પરિચય**

**NuWave Communications**

Name of Organization **NuWave Communications**

Type of Organization IT services, consulting, and

Business solution

Year of Registration 2015

Address

Website                                                    https://nuwave.com/

**DeepCoder Pvt. Ltd**. એક સોફ્ટવેર ડેવેલોપમેન્ટ કંપની છે. કંપની મોબાઈલ અને વેબ એપ ડેવલપમેન્ટની સેવાઓ પૂરી પાડે છે. કંપની ઈ-કોમર્સ, ઓટોમોબાઈલ્સ, હેલ્થ, ફાર્મસી, રિયલ એસ્ટેટ, એજ્યુકેશન, લો, ટ્રાવેલ, હેલ્પડેસ્ક અને હોસ્પિટાલિટી, લોજિસ્ટિક, ફાયનાન્સ, આઈટી સેવાઓ વગેરે જેવા સોલ્યુશન્સ પૂરા પાડે છે.

કંપની મોબાઈલ એપ માટે Flutter , android & ios ટેકનોલોજીનો ઉપયોગ કરે છે. અને વેબ એપ ડેવેલોપપમેન્ટ માટે રિએક્ટ, nodejs , Neatjs, PHP expressjs , mongodb અને AWS જેવી ટેક્નોલોજી નો ઉપયોગ કરે છે.

સાગર જસાણી કંપનીના સીઈઓ અને સ્થાપક છે. અને હરી પટેલ કંપનીના સહ-સ્થાપક તરીકે કાર્યરત છે. કંપનીના કર્મચારીઓની સંખ્યા 50 થી 60 છે.

**પ્રસ્તાવના**

**AutomationExercise** એ ડેમો ઇ-કોમર્સ વેબસાઇટ છે જે QA અને ઓટોમેશન ટેસ્ટિંગનો અભ્યાસ કરવા માટે રચાઈ છે. આ પ્લેટફોર્મ વાસ્તવિક શોપિંગ અનુભવ આપે છે અને વિવિધ પ્રકારના ટેસ્ટિંગ સ્નેરીયો માટે ઉપયોગી છે.

**મુખ્ય સુવિધાઓ**

* **રજિસ્ટ્રેશન અને લોગિન:**  
  યુઝર્સ એકાઉન્ટ બનાવી, લોગિન કરી અને પ્રોફાઇલ મેનેજ કરી શકે છે.
* **પ્રોડક્ટ બ્રાઉઝિંગ:**  
  કેટેગરી અનુસાર પ્રોડક્ટ્સ શોધી અને ફિલ્ટર કરી શકાય છે.
* **કાર્ટ મેનેજમેન્ટ:**  
  કાર્ટમાં વસ્તુઓ ઉમેરવી, કાઢવી અને તેમના ક્વોન્ટિટી બદલવી શક્ય છે.
* **ચેકઆઉટ અને પેમેન્ટ (સિમ્યુલેટેડ):**  
  ઓર્ડર પ્લેસ કરવા માટે બિલિંગ માહિતી આપી શકાય છે.
* **સંપર્ક ફોર્મ:**  
  પ્રશ્નો કે ફીડબેક મોકલવા માટે ફોર્મ ઉપલબ્ધ છે.
* **સબ્સ્ક્રિપ્શન:**  
  ન્યુઝલેટર્સ માટે સબ્સ્ક્રાઇબ કરવાની સગવડ છે.

**પ્રોજેક્ટનો ઉપયોગ**

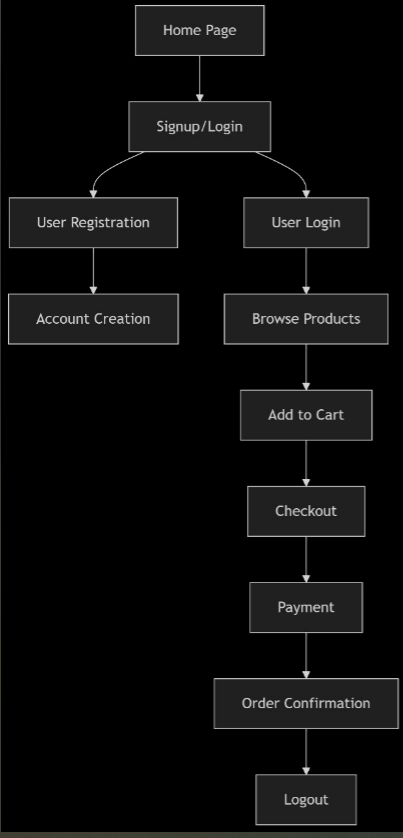
આ પ્રોજેક્ટ મદદ કરે છે:

* Selenium, Cypress, Playwright જેવા ટૂલ્સ વડે ઓટોમેશન પ્રેક્ટિસમાં
* વિવિધ યુઝર ફ્લોઝ માટે ટેસ્ટ કેસ લખવા અને ચલાવવા
* UI, વેલિડેશન અને એરર હેન્ડલિંગ ચકાસવા
* API ટેસ્ટિંગ અને CI/CD પાઇપલાઇન ઈન્ટિગ્રેશનના અભ્યાસ માટે

**Project Profile: -**

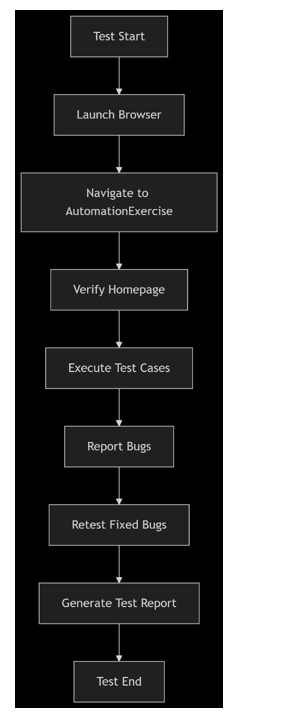
|  |  |
| --- | --- |
| **Project Title:** | **Automation Exercise** |
| **Project Duration:** | **[Jan 2025 to April 2025]** |
| **Testing Team:** | **Makwana Isha [212308003]** |
| **Testing Type:** | **manual testing and automation testing** |
| **Website Category:** | **User side** |
| **manual Tool:** | **Jira, TestMo** |
| **automation Tool:** | **IntelliJ, java selenium, Postman** |
| **Project Guide:** | **Pro. Dhiren Patel** |
| **Team Member** | **1** |

7. **Project Flow Diagram**



**8. Backend System Flow:**

**9. Project Test Flow Diagram**



**10. Test Scenarios for AutomationExercise.com**

|  |  |
| --- | --- |
| **Scenario Id** | **Test Scenario** |
| TS001 | Verify the home page loads successfully |
| TS002 | Verify the navigation bar is visible |
| TS003 | Verify navigation to Signup/Login page |
| TS004 | Verify login with valid credentials |
| TS005 | Verify login with invalid credentials |
| TS006 | Verify signup with new user |
| TS007 | Verify error for existing email during signup |
| TS008 | Verify contact form with valid data |
| TS009 | Verify error on submitting empty contact form |
| TS010 | Verify navigation to Products page |
| TS011 | Verify product search functionality |
| TS012 | Verify product details page |
| TS013 | Verify add to cart from product list |
| TS014 | Verify navigation to cart |
| TS015 | Verify remove item from cart |
| TS016 | Verify cart is empty message |
| TS017 | Verify proceed to checkout button |
| TS018 | Verify place order with logged in user |
| TS019 | Verify test cases page loads |
| TS020 | Verify subscription on homepage |
| TS021 | Verify scroll up button functionality |
| TS022 | Verify logout functionality |

11. **Test Case**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Testcase Id** | **Test Scenario** | **Test Step** | **Expected Result** | **Actual Result** | **Status** | **Tested By** | **Priority** |
| TC001 | Verify the home page loads successfully | Navigate to home page | Home page displays correctly with all sections visible | Home page loaded successfully with all sections visible | Pass | Isha Makwana | High |
| TC002 | Verify the navigation bar is visible | Check for navigation bar | Navigation bar should be visible with all menu items | All menu items in the navigation bar appeared as expected | Pass | Isha Makwana | High |
| TC003 | Verify navigation to Signup/Login page | Click on 'Signup / Login' | 'Login to your account' and 'New User Signup!' should be visible | Login and signup prompts were correctly displayed | Pass | Isha Makwana | High |
| TC004 | Verify login with valid credentials | Enter valid email and password and click Login | User is logged in successfully | User login was successful with valid credentials | Pass | Isha Makwana | High |
| TC005 | Verify login with invalid credentials | Enter invalid email/password and click Login | Error message is displayed | Appropriate error message appeared for invalid login | Pass | Isha Makwana | High |
| TC006 | Verify signup with new user | Enter name and new email and click Signup | Redirected to account information page | User redirected to account page after signup | Pass | Isha Makwana | High |
| TC007 | Verify error for existing email during signup | Enter already registered email | Error message 'Email Address already exist!' is displayed | System displayed a message indicating the email already exists | Pass | Isha Makwana | High |
| TC008 | Verify contact form with valid data | Fill all fields and click Submit | Success message should appear | A confirmation message appeared successfully | Pass | Isha Makwana | High |
| TC009 | Verify error on submitting empty contact form | Click Submit with empty fields | Error or validation messages appear | Validation messages were shown as required | Pass | Isha Makwana | Medium |
| TC010 | Verify navigation to Products page | Click on 'Products' | Products page should be displayed | Products page rendered correctly as expected | Pass | Isha Makwana | High |
| TC011 | Verify product search functionality | Search for a product | Relevant search results are displayed | Search results were relevant and  correctly displayed | Pass | Isha Makwana | High |
| TC012 | Verify product details page | Click on 'View Product' | Product detail page should be visible | Detailed view of the product was accessible | Pass | Isha Makwana | High |
| TC013 | Verify add to cart from product list | Click 'Add to cart' on a product | Product should be added to cart | Item was added to the cart without any issue | Pass | Isha Makwana | High |
| TC014 | Verify navigation to cart | Click on 'Cart' | Cart page is displayed with added products | Cart displayed the correct added items | Pass | Isha Makwana | High |
| TC015 | Verify remove item from cart | Click 'X' on product in cart | Product should be removed from cart | Selected item was removed successfully from the cart | Pass | Isha Makwana | Medium |
| TC016 | Verify cart is empty message | Empty the cart | 'Cart is empty!' message should appear | Empty cart notification appeared as expected | Pass | Isha Makwana | Medium |
| TC017 | Verify proceed to checkout button | Click on 'Proceed To Checkout' | User navigates to checkout page | User was directed to the checkout screen | Pass | Isha Makwana | High |
| TC018 | Verify place order with logged in user | Login and place order | Order confirmation message should appear | Order confirmation was shown after placing the order | Pass | Isha Makwana | High |
| TC019 | Verify test cases page loads | Click on 'Test Cases' | Test Cases page should be visible | Test cases page opened and loaded properly | Pass | Isha Makwana | Low |
| TC020 | Verify subscription on homepage | Enter email and click arrow in footer | Success message 'You have been successfully subscribed!' is shown | User saw a message confirming successful subscription | Pass | Isha Makwana | Medium |
| TC021 | Verify scroll up button functionality | Scroll down and click 'Scroll Up' | Page should scroll to top | Scroll-to-top functionality worked as intended | Pass | Isha Makwana | Low |
| TC022 | Verify logout functionality | Click 'Logout' | User is redirected to login page | User was successfully redirected to the login page | Pass | Isha Makwana | High |

12. **API Testing (Postman)**

**API 1: Get All Products List**

* **Endpoint:** GET /api/productsList
* **Full URL:** https://automationexercise.com/api/productsList
* **Description:** Retrieves a complete list of all available products.
* **Request Headers:**
  + **Content-Type:** application/json *(optional for GET)*
* **Request Body:** Not required
* **Success Response:**
  + **Status Code**: 200 OK
* **Response Body:**

{

    “responseCode”: 200,

    “products”: [

        {

            “id”: 1,

            “name”: “Blue Top”,

            “price”: “Rs. 500”,

            “brand”: “Polo”,

            “category”: {

                “usertype”: {

                    “usertype”: “Women”

                },

                “category”: “Tops”

            }

        },

…

]

}

**Test Script for the API:**

// Store JSON response body

**var** jsonData = pm.response.json();

// Assert that the HTTP response code is 200

pm.test("HTTP response code is as expected 200", **function** () {

    pm.response.to.have.status(200);

});

// Assert that the "responseCode" is 200 in the response body

pm.test("responseCode in response body is as expected 200", **function** () {

    pm.expect(jsonData.responseCode).to.eql(200);

});

// Assert that the HTTP response code and the 'responseCode' in the response body are the same

pm.test("HTTP Response Code matches the 'responseCode' in the response body", **function** () {

    pm.expect(pm.response.code).to.eql(jsonData.responseCode);

});

// Assert "products" attribute exists

pm.test("response contains 'products' attribute", **function** () {

    pm.expect(jsonData).to.have.property('products');

});

// \*\* Verify all expected attributes are present in 'products' array \*\*

pm.test("'products' array is not empty", **function** () {

    pm.expect(jsonData.products).to.be.an('array').that.is.not.empty;

});

// Check if 'id' exists in the products

pm.test("'products' contains 'id' attribute", **function** () {

    pm.expect(jsonData.products[0]).to.have.property('id');

});

// Check if 'name' exists in the products

pm.test("'products' contains 'name' attribute", **function** () {

    pm.expect(jsonData.products[0]).to.have.property('name');

});

// Check if 'price' exists in the products

pm.test("'products' contains 'price' attribute", **function** () {

    pm.expect(jsonData.products[0]).to.have.property('price');

});

// Check if 'brand' exists in the products

pm.test("'products' contains 'brand' attribute", **function** () {

    pm.expect(jsonData.products[0]).to.have.property('brand');

});

// Check if 'category' exists in the products

pm.test("'products' contains 'category' attribute", **function** () {

    pm.expect(jsonData.products[0]).to.have.property('category');

});

// Check if 'category' exists in the products

pm.test("'products' contains 'category' attribute", **function** () {

    pm.expect(jsonData.products[0]).to.have.property('category');

});

// Check if 'usertype' exists in the category

pm.test("'category' contains 'usertype' attribute", **function** () {

    pm.expect(jsonData.products[0].category).to.have.property('usertype');

});

// Check if 'usertype' exists in the category

pm.test("'category' contains 'category' attribute", **function** () {

    pm.expect(jsonData.products[0].category).to.have.property('category');

});

**Test Results:**

|  |  |
| --- | --- |
| Test Results | |
| PASSED | HTTP response code is as expected 200 |
| PASSED | responseCode in response body is as expected 200 |
| PASSED | HTTP Response Code matches the 'responseCode' in the response body |
| PASSED | response contains 'products' attribute |
| PASSED | 'products' array is not empty |
| PASSED | 'products' contains 'id' attribute |
| PASSED | 'products' contains 'name' attribute |
| PASSED | 'products' contains 'price' attribute |
| PASSED | 'products' contains 'brand' attribute |
| PASSED | 'products' contains 'category' attribute |
| PASSED | 'products' contains 'category' attribute |
| PASSED | 'category' contains 'usertype' attribute |
| PASSED | 'category' contains 'category' attribute |

**API 2: Get All Brands List**

* **Endpoint:** GET /api/brandsList
* **Full URL:** https://automationexercise.com/api/brandsList
* **Description:** Returns a list of all available product brands.
* **Request Headers:**
  + Content-Type: application/json *(optional for GET)*
* **Request Body*:***Not required
* **Success Response:**
  + **Status Code:** 200 OK
* **Response Body:**

{

    "responseCode": 200,

    "brands": [

        {

            "id": 1,

            "brand": "Polo"

        },

        {

            "id": 2,

            "brand": "H&M"

        },

...

]

}

**Test Script for the API:**

// Store JSON response body

**var** jsonData = pm.response.json();

// Assert that the HTTP response code is 200

pm.test("HTTP response code is as expected 200", **function** () {

    pm.response.to.have.status(200);

});

// Assert that the "responseCode" is 200 in the response body

pm.test("responseCode in response body is as expected 200", **function** () {

    pm.expect(jsonData.responseCode).to.eql(200);

});

// Assert that the HTTP response code and the 'responseCode' in the response body are the same

pm.test("HTTP Response Code matches the 'responseCode' in the response body", **function** () {

    pm.expect(pm.response.code).to.eql(jsonData.responseCode);

});

// Assert "brands" attribute exists

pm.test("response contains 'brands' attribute", **function** () {

    pm.expect(jsonData).to.have.property('brands');

});

// Assert that 'brands' array is not empty

pm.test("'brands' array is not empty", **function** () {

    pm.expect(jsonData.brands).to.be.an('array').that.is.not.empty;

});

jsonData.brands.forEach(**function** (item) {

    pm.test(`Brand ID is as expected: ${item.id}`, **function** () {

        pm.expect(item.id).to.be.a('number');

    });

    pm.test(`Brand name is valid for ID`, **function** () {

        pm.expect(item.brand).to.be.a('string').that.is.not.empty;

    });

});

**Test Results:**

|  |  |
| --- | --- |
| Test Results | |
| PASSED | HTTP response code is as expected 200 |
| PASSED | responseCode in response body is as expected 200 |
| PASSED | HTTP Response Code matches the 'responseCode' in the response body |
| PASSED | response contains 'brands' attribute |
| PASSED | 'brands' array is not empty |
| PASSED | Brand ID is as expected: 1 |
| PASSED | Brand name is valid for ID |
| PASSED | Brand ID is as expected: 2 |
| PASSED | Brand name is valid for ID |

**13. Bug Sheet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bug Id** | **Date** | **Module Name** | **Bug Description** | **Priority** | **Status** | **Bug Posted By** |
| BUG001 | April 02, 2025 | Home Page | Broken image on homepage slider | Medium | Open | Isha Makwana |
| BUG002 | April 03, 2025 | Signup/Login | Login allows access with incorrect password | High | In Progress | Isha Makwana |
| BUG003 | April 04, 2025 | Contact Us | Contact form submits with empty required fields | Critical | Open | Isha Makwana |
| BUG004 | April 05, 2025 | Products | Product image not loading on detail page | Medium | Fixed | Isha Makwana |
| BUG005 | April 06, 2025 | Cart | Cart total miscalculates price with multiple quantities | High | Open | Isha Makwana |
| BUG006 | April 07, 2025 | Checkout | Order places without selecting payment method | Critical | Open | Isha Makwana |
| BUG007 | April 08, 2025 | Subscription | No success message after subscribing with valid email | Low | Closed | Isha Makwana |
| BUG008 | April 09, 2025 | Scroll | Scroll-up button not working in Firefox | Medium | In Progress | Isha Makwana |
| BUG009 | April 10, 2025 | Test Cases | Test cases page displays 404 error | High | Open | Isha Makwana |
| BUG010 | April 11, 2025 | Logout | User remains logged in after clicking logout | Critical | Open | Isha Makwana |
| BUG011 | April 12, 2025 | Cart | Product not removed from cart after clicking delete icon | Medium | Fixed | Isha Makwana |
| BUG012 | April 13, 2025 | Checkout | Checkout crashes on selecting shipping option | High | Open | Isha Makwana |
| BUG013 | April 14, 2025 | Products | Add to cart button unresponsive for featured items | Medium | Open | Isha Makwana |
| BUG014 | April 15, 2025 | Home Page | Navigation bar overlaps with content on small screens | Low | In Progress | Isha Makwana |
| BUG015 | April 16, 2025 | Signup/Login | No validation on signup form fields | High | Open | Isha Makwana |
| BUG016 | April 17, 2025 | Products | Slow load time for product detail page | Medium | Closed | Isha Makwana |
| BUG017 | April 18, 2025 | Cart | Quantity update doesn't reflect in cart | Medium | Open | Isha Makwana |

14**. Test Scripts (Key Functionalities):**

**1. Add To Cart Test (Selenium WebDriver -java):**

* **CartPage.java**

package pages;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
  
public class CartPage {  
 WebDriver driver;  
 By Products = By.*xpath*("(//a[@href='/products'])[1]");  
 By deleteItem = By.*xpath*("//tr[@id='product-1'] //a[@class='cart\_quantity\_delete']");  
 By clickHereForBuyProduct = By.*xpath*("//u[normalize-space()='here']");  
  
 public CartPage(WebDriver driver) {  
 this.driver = driver;  
 }  
  
 public void removeItemFromCart() throws InterruptedException  
 {  
 Thread.*sleep*(2000);  
 driver.findElement(deleteItem).click();  
 }  
  
 public void clickToHereForBuyProducts() throws InterruptedException  
 {  
 Thread.*sleep*(2000);  
 driver.findElement(clickHereForBuyProduct).click();  
 }  
  
  
 private void scrollToElement(By locator) {  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 js.executeScript("arguments[0].scrollIntoView(true);", driver.findElement(locator));  
 }  
  
 // You can add cart validation methods here  
}

* **HomePage.java**

package pages;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
  
public class HomePage {  
 WebDriver driver;  
  
 By expandWomen = By.*xpath*("//div[@id='accordian'] //a[@href='#Women']");  
 By expandMen = By.*xpath*("//div[@id='accordian'] //a[@href='#Men']");  
 By expandKids = By.*xpath*("//div[@id='accordian'] //a[@href='#Kids']");  
  
 By topsSubCategory = By.*xpath*("//div[@id='Women'] //a[contains(text(),'Tops')]");  
 By sareeSubCategory = By.*xpath*("//div[@id='Women'] //a[contains(text(),'Saree')]");  
  
 public HomePage(WebDriver driver) {  
 this.driver = driver;  
 }  
  
 public void selectWomenTops() throws InterruptedException {  
 scrollToElement(expandWomen);  
 driver.findElement(expandWomen).click();  
 driver.findElement(topsSubCategory).click();  
 Thread.*sleep*(2000);  
 }  
  
 public void selectWomenSaree() throws InterruptedException {  
 scrollToElement(expandWomen);  
 driver.findElement(expandWomen).click();  
 driver.findElement(sareeSubCategory).click();  
 Thread.*sleep*(2000);  
 }  
  
 private void scrollToElement(By locator) {  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 js.executeScript("arguments[0].scrollIntoView(true);", driver.findElement(locator));  
 }  
  
  
}

* **ProductPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import java.util.List;

public class ProductPage {

WebDriver driver;

By addProductIntoCart = By.xpath("//a[contains(@class, 'add-to-cart')]");

By viewCart = By.xpath("//u[text()='View Cart']");

public ProductPage(WebDriver driver) {

this.driver = driver;

}

public void addFirstProductToCart() throws InterruptedException {

List<WebElement> buttons = driver.findElements(addProductIntoCart);

WebElement firstButton = buttons.getFirst();

Thread.sleep(2000);

// Hover over the first product's parent or the button itself

Actions actions = new Actions(driver);

actions.moveToElement(firstButton).perform();

Thread.sleep(1000); // Wait for any animation or hover effects

actions.click(firstButton).perform();

Thread.sleep(2000);

}

public void addSecondProductToCart() throws InterruptedException {

scrollToElement(addProductIntoCart);

List<WebElement> buttons = driver.findElements(addProductIntoCart);

WebElement secondButton = buttons.get(1);

Thread.sleep(2000);

// Hover over the first product's parent or the button itself

Actions actions = new Actions(driver);

actions.moveToElement(secondButton).perform();

Thread.sleep(1000); // Wait for any animation or hover effects

actions.click(secondButton).perform();

Thread.sleep(2000);

}

public void clickViewCart()

{

driver.findElement(viewCart).click();

}

private void scrollToElement(By locator) {

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].scrollIntoView(true);", driver.findElement(locator));

}

}

* **AutomationExerciseTest.java**

import io.github.bonigarcia.wdm.WebDriverManager;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.\*;

import pages.\*;

import java.time.Duration;

public class AutomationExerciseTest {

WebDriver driver;

HomePage homePage;

ProductPage productPage;

CartPage cartPage;

@BeforeClass

public void setup() {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

driver.manage().window().maximize();

driver.get("https://www.automationexercise.com/");

homePage = new HomePage(driver);

productPage = new ProductPage(driver);

cartPage = new CartPage(driver);

}

@Test

public void cart\_AddProduct\_RemoveProduct\_AddSameProductAgain() throws InterruptedException {

// Step 1-2: Go to Women > Tops

homePage.selectWomenTops();

// Step 3: Add first product

productPage.addFirstProductToCart();

// Step 4: View Cart

productPage.clickViewCart();

// Step 5: remove item from cart

cartPage.removeItemFromCart();

// Step 6: click here to buy product

cartPage.clickToHereForBuyProducts();

// Step 7: Again select deleted product

homePage.selectWomenTops();

// Step 8: Add product in cart

productPage.addFirstProductToCart();

// Step 9: View Cart

productPage.clickViewCart();

}

@AfterClass

public void tearDown() {

driver.quit();

}

}