**Scenario 1:**

**Positive Test Cases for Scenario 1 in Cypress**

describe("Scenario 1: Search Conditions Set Correctly", () => {

beforeEach(() => {

cy.visit("https://jp.mercari.com/");

});

it("Positive Test Case 1: Verify correct category selection", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

cy.get(".selected-category").should("contain", "Computers & Technology");

});

it("Positive Test Case 2: Verify sorting options are available and functional", () => {

cy.get("#search-bar").click();

cy.get(".sort-dropdown").should("be.visible");

cy.get(".sort-dropdown").click();

cy.get(".sort-option").should("have.length.gt", 0);

// Add more actions and assertions for sorting options

});

it("Positive Test Case 3: Check if filters can be applied and affect search results", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

cy.get(".selected-category").should("contain", "Computers & Technology");

// Apply a filter (example: Price range)

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Add assertions for filtered results

});

it("Positive Test Case 4: Test pagination functionality", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

cy.get(".selected-category").should("contain", "Computers & Technology");

// Add assertions for pagination controls

cy.get(".next-page-button").click(); // Click on next page

// Add assertions for the next page of results

});

it("Positive Test Case 5: Verify the selected category remains highlighted", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Verify that the selected category remains highlighted

cy.get(".selected-category").should("have.text", "Computers & Technology");

});

it("Positive Test Case 6: Check if the breadcrumb trail displays the selected categories correctly", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Verify that the breadcrumb trail displays the selected categories correctly

cy.get(".breadcrumb").should("contain", "Books, Music & Games > Books > Computers & Technology");

});

it("Positive Test Case 7: Confirm that the URL reflects the selected categories", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Verify that the URL reflects the selected categories

cy.url().should("include", "category=Computers+%26+Technology");

});

it("Positive Test Case 8: Verify that the product images load correctly in the search results", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Add assertions to verify product images

});

it("Positive Test Case 9: Test searching with specific keywords within selected categories", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Enter a specific keyword (e.g., "programming") and search

cy.get("#search-bar").type("programming");

cy.get(".search-button").click();

// Add assertions for search results with the specific keyword

});

it("Positive Test Case 10: Verify that the search bar displays the correct category name", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Verify that the search bar displays the correct category name

cy.get("#search-bar").should("have.value", "Computers & Technology");

});

it("Positive Test Case 11: Verify that the 'Clear' button resets all search conditions", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Apply filters and search conditions

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Click the 'Clear' button

cy.get(".clear-button").click();

// Verify that all search conditions are reset

cy.get(".selected-category").should("not.exist");

cy.get(".selected-filters").should("not.exist");

});

it("Positive Test Case 12: Verify that the search results display relevant products", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Add assertions to verify relevant search results

});

it("Positive Test Case 13: Test navigating back to the homepage after searching", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Perform a search

cy.get(".search-button").click();

// Navigate back to the homepage

cy.go("back");

// Verify that the homepage is displayed

cy.url().should("eq", "https://jp.mercari.com/");

});

it("Positive Test Case 14: Verify that the 'Sort' dropdown options are functional", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Open the 'Sort' dropdown

cy.get(".sort-dropdown").click();

// Select a sorting option (e.g., 'Price: Low to High')

cy.get(".sort-option").contains("Price: Low to High").click();

// Add assertions to verify the sorting order of search results

});

it("Positive Test Case 15: Test navigating back to the homepage after applying filters", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Apply filters (e.g., price range)

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Navigate back to the homepage

cy.go("back");

// Verify that the homepage is displayed

cy.url().should("eq", "https://jp.mercari.com/");

});

it("Positive Test Case 16: Verify that the 'Clear' button resets filters without resetting categories", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Apply filters (e.g., price range)

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Click the 'Clear' button

cy.get(".clear-button").click();

// Verify that filters are reset, but categories remain selected

cy.get(".selected-category").should("contain", "Computers & Technology");

cy.get(".selected-filters").should("not.exist");

});

it("Positive Test Case 17: Verify that the 'Clear' button is disabled when no filters are applied", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Verify that the 'Clear' button is initially disabled

cy.get(".clear-button").should("be.disabled");

});

it("Positive Test Case 18: Verify that the search bar allows users to enter long search queries", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Enter a long search query (e.g., 500 characters)

const longQuery = "A".repeat(500);

cy.get("#search-bar").type(longQuery);

// Verify that the search bar accepts the long query

cy.get("#search-bar").should("have.value", longQuery);

});

it("Positive Test Case 19: Verify that the 'Search' button works correctly", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Enter a keyword and click the 'Search' button

cy.get("#search-bar").type("programming");

cy.get(".search-button").click();

// Add assertions for search results with the specific keyword

});

it("Positive Test Case 20: Verify that the 'Search' button is disabled when no keyword is entered", () => {

cy.get("#search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

// Click 'Search' without entering a keyword

cy.get(".search-button").click();

// Verify that the 'Search' button is disabled

cy.get(".search-button").should("be.disabled");

});

});

**Negative Test Cases for Scenario 1 in Cypress**

describe("Scenario 1: Search Conditions Set Correctly", () => {

beforeEach(() => {

cy.visit("https://jp.mercari.com/");

});

it("Negative Test Case 1: Verify error message for empty filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Apply filters without setting any values

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 2: Verify error message for empty keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Click the 'Search' button without entering a keyword

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 3: Verify error for non-existent category selection", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select a non-existent category

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("NonExistentCategory").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 4: Verify error for invalid filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Apply invalid filters (e.g., enter non-numeric values)

cy.get("#price-range-min").type("abc");

cy.get("#price-range-max").type("def");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 5: Verify error for conflicting categories", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select conflicting categories

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-2-category").contains("Electronics").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 6: Verify error for invalid characters in the search bar", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter invalid characters in the search bar

cy.get("#search-bar").type("!@#$%^&\*()\_+=-");

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 7: Verify error for non-existent sorting options", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Open the 'Sort' dropdown

cy.get(".sort-dropdown").click();

// Select a non-existent sorting option

cy.get(".sort-option").contains("NonExistentSortOption").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 8: Verify error for applying filters without selecting a category", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Click 'Apply Filters' without selecting a category

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 9: Verify error for invalid price range filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select category

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

// Apply invalid price range filters

cy.get("#price-range-min").type("5000");

cy.get("#price-range-max").type("1000");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 10: Verify error for special characters in category selection", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select category with special characters

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("!@#Category").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 11: Verify error for special characters in filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Apply filters with special characters (e.g., price range)

cy.get("#price-range-min").type("!@#$");

cy.get("#price-range-max").type("!@#$");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 12: Verify error for invalid characters in the search bar", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter invalid characters in the search bar

cy.get("#search-bar").type("<script>alert('XSS')</script>");

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 13: Verify error for non-existent categories", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select non-existent categories

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("NonExistentTier1").click();

cy.get(".tier-2-category").contains("NonExistentTier2").click();

cy.get(".tier-3-category").contains("NonExistentTier3").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 14: Verify error for very long and invalid keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter a very long and invalid keyword

const longInvalidKeyword = "A".repeat(10000);

cy.get("#search-bar").type(longInvalidKeyword);

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 15: Verify error for categories with special characters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select categories with special characters

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("!@#Category").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 16: Verify error for filters with negative values", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Apply filters with negative values (e.g., price range)

cy.get("#price-range-min").type("-1000");

cy.get("#price-range-max").type("-500");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 17: Verify error for invalid special characters in search bar", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter invalid special characters in the search bar

cy.get("#search-bar").type("<script>alert('XSS')</script>");

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 18: Verify error for excessively long search query", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter an excessively long search query (e.g., 10000 characters)

const longQuery = "A".repeat(10000);

cy.get("#search-bar").type(longQuery);

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 19: Verify error for empty keyword and filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Click 'Apply Filters' without entering a keyword

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 20: Verify error for searching with invalid category names", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select invalid category names

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("InvalidCategory1").click();

cy.get(".tier-2-category").contains("InvalidCategory2").click();

cy.get(".tier-3-category").contains("InvalidCategory3").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

});

**Scenario 2:**

**Positive Test Cases for Scenario 2 in Cypress**

describe("Scenario 2: Search Conditions Set from Latest Browsing History", () => {

beforeEach(() => {

// Visit the Mercari website and set up browsing history

cy.visit("https://jp.mercari.com/");

cy.get(".search-bar").click();

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

cy.get(".tier-2-category").contains("Books").click();

cy.get(".tier-3-category").contains("Computers & Technology").click();

cy.get("#search-bar").type("javascript");

cy.get(".search-button").click();

});

it("Positive Test Case 1: Verify that the latest history entry is displayed correctly", () => {

// Verify that the latest history entry is displayed correctly

cy.get(".latest-history-entry").should("contain", "Computers & Technology / javascript");

});

it("Positive Test Case 2: Test clicking on the latest history entry", () => {

// Click on the latest history entry

cy.get(".latest-history-entry").click();

// Verify that the search conditions on the left sidebar are set correctly

cy.get(".selected-category").should("contain", "Computers & Technology");

cy.get("#search-bar").should("have.value", "javascript");

});

it("Positive Test Case 3: Verify that the keyword 'javascript' is auto-populated in the search bar", () => {

// Verify that the search bar contains the keyword 'javascript'

cy.get("#search-bar").should("have.value", "javascript");

});

it("Positive Test Case 4: Test navigating back from the search results page", () => {

// Perform a search

cy.get(".search-button").click();

// Go back to the homepage

cy.go("back");

// Verify that the keyword 'javascript' is still in the search bar

cy.get("#search-bar").should("have.value", "javascript");

});

it("Positive Test Case 5: Confirm that the browsing history entries are sorted in chronological order", () => {

// Verify that the latest history entry appears first

cy.get(".browsing-history-entry").first().should("contain", "Computers & Technology / javascript");

});

it("Positive Test Case 6: Test clearing browsing history", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Verify that the browsing history is empty

cy.get(".browsing-history").should("not.exist");

});

it("Positive Test Case 7: Test searching after clearing browsing history", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Input a new keyword and search

cy.get("#search-bar").type("laptop");

cy.get(".search-button").click();

// Add assertions for search results with the new keyword

});

it("Positive Test Case 8: Test navigating back to the homepage after searching", () => {

// Perform a search

cy.get(".search-button").click();

// Go back to the homepage

cy.go("back");

// Verify that the homepage is displayed

cy.url().should("eq", "https://jp.mercari.com/");

});

it("Positive Test Case 9: Verify that the 'Clear' button is disabled when the browsing history is empty", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Verify that the 'Clear' button is disabled

cy.get(".clear-history-button").should("be.disabled");

});

it("Positive Test Case 10: Test navigating back from a search results page with an empty browsing history", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Perform a search

cy.get(".search-button").click();

// Go back to the homepage

cy.go("back");

// Verify that the homepage is displayed

cy.url().should("eq", "https://jp.mercari.com/");

});

it("Positive Test Case 11: Test clearing browsing history with special characters in history entry", () => {

// Clear the browsing history (which includes special characters)

cy.get(".clear-history-button").click();

// Verify that the browsing history is empty

cy.get(".browsing-history").should("not.exist");

});

it("Positive Test Case 12: Verify that the 'Clear' button becomes enabled after browsing history is created", () => {

// Clear the browsing history (which includes special characters)

cy.get(".clear-history-button").click();

// Verify that the 'Clear' button becomes enabled after creating history entries

// Perform steps to create a browsing history

// Check that the 'Clear' button is now enabled

});

it("Positive Test Case 13: Test navigating back after clearing browsing history", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Navigate back to the homepage

cy.go("back");

// Verify that the homepage is displayed

cy.url().should("eq", "https://jp.mercari.com/");

});

it("Positive Test Case 14: Verify that the 'Clear' button clears history but preserves the search bar contents", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Verify that the search bar contents (e.g., 'javascript') are preserved

cy.get("#search-bar").should("have.value", "javascript");

});

it("Positive Test Case 15: Test searching after clearing browsing history and entering a new keyword", () => {

// Clear the browsing history

cy.get(".clear-history-button").click();

// Input a new keyword (e.g., 'laptop') and perform a search

cy.get("#search-bar").type("laptop");

cy.get(".search-button").click();

// Add assertions for search results with the new keyword

});

it("Positive Test Case 16: Verify that the search bar displays the correct category name", () => {

// Verify that the search bar displays the correct category name

cy.get("#search-bar").should("have.value", "Computers & Technology");

});

it("Positive Test Case 17: Verify that the 'Clear' button resets filters without resetting categories", () => {

// Apply filters (e.g., price range)

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Clear the filters using the 'Clear' button

cy.get(".clear-button").click();

// Verify that filters are reset, but categories remain selected

cy.get(".selected-category").should("contain", "Computers & Technology");

cy.get(".selected-filters").should("not.exist");

});

it("Positive Test Case 18: Verify that the 'Clear' button is enabled after applying filters", () => {

// Apply filters (e.g., price range)

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Verify that the 'Clear' button is enabled

cy.get(".clear-button").should("not.be.disabled");

});

it("Positive Test Case 19: Verify that the 'Clear' button is enabled after selecting categories", () => {

// Verify that the 'Clear' button is enabled after selecting categories

cy.get(".clear-button").should("not.be.disabled");

});

it("Positive Test Case 20: Verify that the 'Clear' button is enabled after entering a search keyword", () => {

// Verify that the 'Clear' button is enabled after entering a search keyword

cy.get("#search-bar").type("programming");

cy.get(".clear-button").should("not.be.disabled");

});

});

**Negative Test Cases for Scenario 2 in Cypress**

describe("Scenario 2: Search Conditions Set from Latest Browsing History", () => {

beforeEach(() => {

cy.visit("https://jp.mercari.com/");

});

it("Negative Test Case 1: Verify clicking on a non-existent history entry", () => {

// Click on a non-existent history entry (e.g., one that does not exist)

cy.get(".non-existent-history-entry").click();

// Verify that a page not found or error page is displayed

cy.contains("Page not found").should("be.visible");

});

it("Negative Test Case 2: Verify searching with an invalid keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter an invalid keyword (e.g., "!@#$%^") and search

cy.get("#search-bar").type("!@#$%^");

cy.get(".search-button").click();

// Verify that no search results are displayed

cy.get(".search-results").should("not.exist");

});

it("Negative Test Case 3: Verify error for clicking on a non-existent history entry", () => {

// Click on a non-existent history entry (e.g., one that does not exist)

cy.get(".non-existent-history-entry").click();

// Verify that a page not found or error page is displayed

cy.contains("Page not found").should("be.visible");

});

it("Negative Test Case 4: Verify error for searching with an invalid keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter an invalid keyword (e.g., "!@#$%^") and search

cy.get("#search-bar").type("!@#$%^");

cy.get(".search-button").click();

// Verify that no search results are displayed

cy.get(".search-results").should("not.exist");

});

it("Negative Test Case 5: Verify error for clicking on a non-clickable history entry", () => {

// Click on a non-clickable history entry (e.g., disabled or non-interactive)

cy.get(".non-clickable-history-entry").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 6: Verify error for clearing an empty browsing history", () => {

// Click on the 'Clear History' button without any browsing history

cy.get(".clear-history-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 7: Verify error for searching after clearing an empty browsing history", () => {

// Click on the 'Clear History' button without any browsing history

cy.get(".clear-history-button").click();

// Input a new keyword and search

cy.get("#search-bar").type("laptop");

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 8: Verify error for clicking on non-interactive elements in history", () => {

// Click on a history entry that is not interactive (e.g., plain text)

cy.get(".non-interactive-history-entry").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 9: Verify error for searching with a non-existent keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter a non-existent keyword and search

cy.get("#search-bar").type("NonExistentKeyword");

cy.get(".search-button").click();

// Verify that no search results are displayed

cy.get(".search-results").should("not.exist");

});

it("Negative Test Case 10: Verify error for clearing browsing history with an invalid keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter an invalid keyword and perform a search

cy.get("#search-bar").type("<script>alert('XSS')</script>");

cy.get(".search-button").click();

// Clear the browsing history

cy.get(".clear-history-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 11: Verify error for clearing browsing history with special characters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter a keyword and perform a search

cy.get("#search-bar").type("laptop");

cy.get(".search-button").click();

// Clear the browsing history (which includes special characters)

cy.get(".clear-history-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 12: Verify error for searching with special characters in keyword", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Enter a keyword with special characters and search

cy.get("#search-bar").type("<script>alert('XSS')</script>");

cy.get(".search-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 13: Verify error for clicking on non-existent sorting options", () => {

// Click on the 'Sort' dropdown

cy.get(".sort-dropdown").click();

// Select a non-existent sorting option

cy.get(".sort-option").contains("NonExistentSortOption").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 14: Verify error for clearing an empty browsing history", () => {

// Click on the 'Clear History' button without any browsing history

cy.get(".clear-history-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 15: Verify error for filters with negative values", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select category

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("Books, Music & Games").click();

// Apply filters with negative values (e.g., price range)

cy.get("#price-range-min").type("-1000");

cy.get("#price-range-max").type("-500");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 16: Verify error for invalid special characters in category selection", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select categories with invalid special characters

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("<script>alert('XSS')</script>").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 17: Verify error for special characters in filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Apply filters with special characters (e.g., price range)

cy.get("#price-range-min").type("!@#$");

cy.get("#price-range-max").type("!@#$");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 18: Verify error for non-existent categories", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select non-existent categories

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("NonExistentTier1").click();

cy.get(".tier-2-category").contains("NonExistentTier2").click();

cy.get(".tier-3-category").contains("NonExistentTier3").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 19: Verify error for empty keyword and filters", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Click 'Apply Filters' without entering a keyword

cy.get("#price-range-min").type("1000");

cy.get("#price-range-max").type("5000");

cy.get(".apply-filters-button").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

it("Negative Test Case 20: Verify error for searching with invalid category names", () => {

// Click on the search bar

cy.get("#search-bar").click();

// Select invalid category names

cy.get(".category-dropdown").click();

cy.get(".tier-1-category").contains("InvalidCategory1").click();

cy.get(".tier-2-category").contains("InvalidCategory2").click();

cy.get(".tier-3-category").contains("InvalidCategory3").click();

// Verify that an error message is displayed

cy.get(".error-message").should("be.visible");

});

});