

Case ID	Section	Title	Type	Preconditions	Steps	Expected Result	Status	Notes	Defect Link
LOGIN-001	Login	Keyboard-only login works	Accessibility	User is logged out.	1. Use Tab to reach email/password fields and submit button. 2. Submit with Enter. 3. Verify focus states.	All controls reachable and operable via keyboard with visible focus; login succeeds with valid credentials.	Pass		
LOGIN-002	Login	Screen reader reads labels and errors correctly	Accessibility	User is logged out. Screen reader available.	1. Navigate login form using screen reader. 2. Trigger an error and listen to announcement.	Fields have programmatic labels; error text is associated and announced properly.	Blocked		
LOGIN-003	Login	Cross-browser login works	Compatibility	User is logged out.	1. Repeat login on different browsers. 2. Compare behaviors.	Login works consistently on all supported browsers.	Pass		
LOGIN-004	Login	Login with valid email and password	Functional	User has a registered account and is logged out.	1. Open the Login page. 2. Enter valid email and password. 3. Submit.	User is authenticated and redirected to the intended page/account area. Session is created.	Pass		
LOGIN-005	Login	Email case-insensitive login	Functional	User has a registered account.	1. Open Login. 2. Enter registered email with different casing. 3. Enter correct password. 4. Submit.	Login succeeds regardless of email casing.	Pass		
LOGIN-006	Login	Whitespace in email is handled consistently	Validation	User has a registered account.	1. Enter email with leading/trailing spaces. 2. Enter correct password. 3. Submit.	Email is trimmed or validated consistently. If trimmed, login succeeds; if rejected, error is clear.	Pass	password field shows additional asterisks when you add whitespaces	
LOGIN-007	Login	Passwords with special characters are accepted	Functional	User account exists with a password containing special characters.	1. Open Login. 2. Enter email and special-character password. 3. Submit.	Login succeeds and password is processed correctly.	Pass		
LOGIN-008	Login	Multiple concurrent sessions behavior is consistent with policy	Functional	User has a registered account.	1. Log in on Device A. 2. Log in on Device B. 3. Verify access on both devices. 4. Log out on one device and re-check the other.	App follows defined policy (allow or restrict). If restricted, older session is invalidated consistently.	Pass		
LOGIN-009	Login	Password manager autofill works on login form	Usability	Browser password manager has saved credentials for the site.	1. Open Login. 2. Use password manager to autofill. 3. Submit.	Autofilled credentials are accepted; validation does not falsely block submission.	Pass		
LOGIN-010	Login	Invalid email format is rejected	Validation	User is logged out.	1. Open Login. 2. Enter invalid email format. 3. Enter any password. 4. Submit.	Login is blocked with field-level email validation message.	Pass		
LOGIN-011	Login	Wrong password shows generic invalid-credentials error	Security	User account exists.	1. Enter valid registered email. 2. Enter wrong password. 3. Submit.	Login fails with a generic message (no hints whether email exists). No session created.	Pass		
LOGIN-012	Login	Non-existent email shows same generic invalid-credentials error	Security	User is logged out.	1. Enter an email not registered. 2. Enter any password. 3. Submit.	Login fails with the same generic message as wrong password (prevents user enumeration).	Pass		
LOGIN-013	Login	Empty email/password shows field errors	Validation	User is logged out.	1. Open Login. 2. Leave email and/or password empty. 3. Submit.	Field-level required errors appear; no login request succeeds.	Pass	There are green ticks in registration even with empty fields	https://mofeezamasoqd.atlassian.net/browse/DEV-10
LOGIN-014	Login	Extremely long inputs are handled safely	Security	User is logged out.	1. Paste 1000+ characters into email and password fields. 2. Submit.	Inputs are constrained or rejected with a safe message. No server error or UI break.	Pass		
LOGIN-015	Login	Pasted hidden characters/newlines handled safely	Validation	User is logged out.	1. Paste an email/password containing newline or zero-width chars. 2. Submit.	Login fails gracefully with a clear message; system does not mis-handle hidden chars.	Pass		
LOGIN-016	Login	Brute-force protection triggers after repeated failures	Security	User account exists or use a test account.	1. Attempt login with wrong password repeatedly (e.g., 10+ times). 2. Observe throttling, CAPTCHA, or lockout behavior. 3. Attempt correct login afterwards.	System rate-limits or challenges suspicious attempts. Legitimate login may require cooldown or additional verification, per policy.	Fail	There is no rate limit to how many failed attempts there can be.	https://mofeezamasoqd.atlassian.net/browse/DEV-8
LOGIN-017	Login	Login response time meets Service Level Agreement	Performance	User is logged out.	1. Measure time from submit to authenticated landing across 5 runs. 2. Note slowest/fastest.	Login completes within agreed SLA	Pass		
LOGIN-018	Login	Backend timeout / 5xx shows friendly error	Resilience	User is logged out.	1. Simulate slow network or backend error 2. Submit login. 3. Observe UI.	User sees a friendly error and can retry; no crash; no sensitive details leaked.	Blocked		
LOGIN-019	Login	Password masking	Security	User is logged out.	1. Type password in login form. 2. Verify masking.	Password is masked by default	Pass		
LOGIN-020	Login	HTTPS enforced for login	Security	User is logged out.	1. Open http://automationpractice.multiformis.com via http. 2. Navigate to login. 3. Attempt login.	Login traffic is sent only over HTTPS; HTTP requests are redirected; no mixed content warnings.	Fail	HTTPS is never requested the site only runs on HTTP	https://mofeezamasoqd.atlassian.net/browse/DEV-9
LOGIN-021	Login	Session ID is regenerated after login	Security	User is logged out; tooling available to inspect cookies.	1. Capture session cookie before login. 2. Log in. 3. Capture session cookie after login.	Session ID changes after authentication; old ID cannot be used to access authenticated resources.	Blocked		
LOGIN-022	Login	Auth cookies have Secure/HttpOnly/SameSite flags	Security	User can inspect cookies in browser devtools.	1. Log in. 2. Inspect authentication/session cookies.	Cookies are configured with appropriate flags (Secure, HttpOnly, SameSite) to reduce risk of theft/CSRF.	Blocked		

Case ID	Section	Title	Type	Preconditions	Steps	Expected Result	Status	Notes	Defect Link
LOGIN-023	Login	SQL injection attempts in login inputs are handled safely	Security	User is logged out.	1. Enter SQL injection-like strings into email/password. 2. Submit.	Login fails safely; no server errors; no stack traces or DB leakage.	Pass		
LOGIN-024	Login	Logout invalidates session; back button cannot access protected pages	Security	User is logged in.	1. Log out. 2. Press browser Back to return to a previously authenticated page. 3. Refresh the page.	User cannot access protected content after logout; pages redirect to login or show logged-out state.	Fail	Back button takes user back to logged in state	https://mofeezamaso.od.atlassian.net/browse/DEV-13
LOGIN-025	Login	Session timeout logs user out after inactivity	Security	User is logged in. Session timeout configured.	1. Stay inactive until session expires. 2. Attempt an authenticated action or refresh a protected page.	User is redirected to login or receives a clear session-expired message; no authenticated access remains.	Pass		
LOGIN-026	Login	Stolen-cookie replay is limited (where testable)	Security	User can export cookies and test in another browser profile (controlled environment).	1. Log in in Browser A. 2. Copy session cookie. 3. Import cookie into Browser B. 4. Attempt to access protected pages.	System behavior matches security policy. Ideally replay is rejected or limited by additional signals; at minimum no unexpected privilege escalation occurs.	Blocked	Requires Charles Proxy to test	
LOGIN-027	Login	Error messaging and response behavior reduce account enumeration	Security	User is logged out.	1. Attempt login with a non-existent email. 2. Attempt login with an existing email + wrong password. 3. Compare messages and overall behavior (including timing).	Messages are indistinguishable and do not confirm whether an email exists; any timing differences are minimal.	Pass		
LOGIN-028	Login	Login errors are clear and consistent	Usability	User is logged out.	1. Trigger common failures (missing fields, invalid email, wrong password). 2. Review errors.	Errors are consistent, readable, and guide the user without revealing sensitive details.	Pass		
LOGIN-029	Login	Caps Lock warning	Usability	User is logged out. Caps lock indicator exists.	1. Enable Caps Lock. 2. Enter Email and password field. 3. Observe any warning.	If feature exists, user receives a clear Caps Lock warning and can dismiss it; otherwise N/A.	Fail	User is not warned about caps lock turned on	

Case ID	Section	Type	Title	Preconditions	Steps	Expected Result	Status	Notes	Defect Link
REG-001	Registration	Compatibility	Cross-browser registration	User is logged out.	1. Repeat Successful Registration on each target browser. 2. Compare Behaviors.	Registration behaves consistently and successfully across supported browsers.	Pass		
REG-002	Registration	Compatibility	Cross-device registration (Desktop/IOS/Android)	User is logged out.	1. Repeat Successful Registration on desktop, iOS, and Android. 2. Compare layout and submission.	Consistent behavior across devices; no mobile-specific blockers.	Pass		
REG-003	Registration	Functional	Successful registration (required fields only)	User is logged out.	1. Open the site. 2. Navigate to the Sign up / Create account page. 3. Fill all mandatory fields with valid data. 4. Submit the registration form.	Account is created successfully. User is authenticated (logged in) and redirected to an account/home page. A success confirmation is shown.	Pass		
REG-004	Registration	Functional	Registration with optional fields	User is logged out.	1. Open Sign up page. 2. Fill mandatory fields with valid data. 3. Fill optional fields with valid data. 4. Submit.	Account is created and all optional values are saved and visible in the user profile/account details where applicable.	Pass		
REG-005	Registration	Functional	Newsletter opt-in during registration	User is logged out. Newsletter option exists.	1. Open Sign up page. 2. Fill mandatory fields. 3. Select newsletter opt-in. 4. Submit.	Account is created. Newsletter/subscription preference is saved and reflected in account preferences (or confirmation shown).	Pass		
REG-006	Registration	Functional	Email case-insensitivity on registration/login	User is logged out.	1. Register with an email using mixed/upper case (e.g., USER@DOMAIN.COM). 2. Log out. 3. Log in using the same email in lower case (user@domain.com) and the same password.	Login succeeds. The system treats emails case-insensitively and does not create duplicates due to casing.	Pass		
REG-007	Registration	Functional	Name accepts common special characters	User is logged out.	1. Open Sign up page. 2. Enter a first/last name with characters like apostrophe/hyphen (e.g., O'Brien, Smith-Jones). 3. Complete signup.	Registration succeeds. Names render correctly in account profile and order history.	Pass		
REG-008	Registration	Functional	Unicode name support	User is logged out.	1. Register using non-ASCII characters in names (e.g., Jörg Müller). 2. View account/profile where name appears.	Registration succeeds and the name is stored/rendered correctly (no corrupted characters).	Pass		
REG-009	Registration	Usability	Browser autofill does not break validation	User is logged out. Browser has saved profile data for form autofill.	1. Open Sign up page. 2. Use browser autofill to populate fields. 3. Submit the form.	Autofilled values are accepted/rejected consistently with the same validations as manual entry; no blocked submission due to autofill quirks.	Pass		
REG-010	Registration	Functional	Double-submit prevention on registration	User is logged out.	1. Fill out the registration form with valid data. 2. Click Submit repeatedly / double-click rapidly. 3. Observe behavior and resulting account count.	Only one account is created. UI prevents repeated submissions (disabled button/spinner) and server handles idempotency safely.	Pass		
REG-011	Registration	Functional	Registration works after logout (no session leakage)	User has an existing account.	1. Log in. 2. Log out. 3. Navigate to Sign up and create a new account with a different email. 4. Verify which account is logged in.	New account is created and authenticated correctly. No data/session from previous user leaks into the new registration flow.	Pass		
REG-012	Registration	Functional	Correct post-registration redirect	User is logged out.	1. Complete registration. 2. Observe the landing page. 3. Attempt to access a page requiring authentication.	User is redirected to the intended post-registration page. Authenticated pages are accessible without re-login.	Pass		
REG-013	Registration	Performance	Registration completes within acceptable time	User is logged out.	1. Measure time from submit to success page across 5 runs on typical network. 2. Note slowest/fastest.	Registration completes within SLA (define target). UI shows progress if longer; no timeouts under normal load.	Pass		
REG-014	Registration	Resilience	Session timeout mid-form is handled gracefully	User is logged out. Session timeout configured.	1. Start filling registration. 2. Stay idle until session expires. 3. Submit the form.	User gets a clear message and is prompted to restart or re-auth; data loss minimized where possible; no confusing failures.	Pass		
REG-015	Registration	Security	Password is masked by default	User is logged out.	1. Open Sign up. 2. Type password. 3. Verify masking.	Password characters are masked by default	Pass		
REG-016	Registration	Security	Registration uses HTTPS / redirects to HTTPS	User is logged out.	1. Open http://automationpractice.multiformis.com using http. 2. Navigate to registration. 3. Submit form.	User is redirected to HTTPS for sensitive actions; no mixed content warnings; credentials are not sent over HTTP.	Fail	HTTPS is never requested the site only runs on HTTP	https://mofeezamasood.atlassian.net/browse/DEV-9
REG-017	Registration	Security	SQL injection strings are handled safely	User is logged out.	1. Enter SQL injection-like strings into text fields (e.g., John' OR '1='1). 2. Submit.	App does not error or leak DB details. Input is sanitized/escaped and stored safely (or rejected).	Pass		
REG-018	Registration	Security	Abuse prevention: rate limiting / captcha	User is logged out.	1. Attempt multiple registrations quickly from the same IP/device. 2. Observe for throttling, CAPTCHA, or blocking behavior.	System throttles or challenges suspicious behavior. User sees a clear message; legitimate users can proceed.	Blocked	Requires Charles Proxy to test	
REG-019	Registration	Usability	Inline errors are clear and actionable	User is logged out.	1. Submit the form with invalid/missing values. 2. Review error messages for each field.	Errors explain what is wrong and how to fix it; wording is consistent; no ambiguous errors.	Pass		
REG-020	Registration	Usability	Focus moves to first invalid field on submit	User is logged out.	1. Leave multiple required fields invalid. 2. Submit. 3. Observe focus/scroll behavior.	Cursor/focus moves to the first invalid field and the page scrolls to it if needed.	Pass		
REG-021	Registration	Accessibility	Keyboard-only registration works (Tab/Enter)	User is logged out.	1. Navigate form using Tab/Shift+Tab only. 2. Toggle checkboxes using Space. 3. Submit using Enter.	All controls are reachable and operable via keyboard; visible focus indicator present; registration can be completed.	Pass		
REG-022	Registration	Accessibility	Form labels and errors are screen-reader friendly	User is logged out. Screen reader available.	1. Use a screen reader to read each form field. 2. Trigger errors and listen to announcements. 3. Check associations between labels, inputs, and error text.	Fields have programmatic labels; errors are announced and associated with inputs (e.g., aria-describedby).	Pass		
REG-023	Registration	Compatibility	Mobile responsiveness of registration page	User is logged out.	1. Open registration on a small viewport (e.g., 360x640). 2. Ensure fields are readable and tappable. 3. Submit with valid data.	Layout is usable on mobile: no horizontal scrolling for form content; tap targets are adequate; no overlapping elements.	Pass		
REG-024	Registration	Validation	Duplicate email is not allowed	An account already exists with email A.	1. Open Sign up. 2. Attempt to register using email A. 3. Submit.	Registration is blocked with a clear error message (e.g., email already registered). No duplicate account is created.	Pass		
REG-025	Registration	Validation	Invalid email formats are rejected	User is logged out.	1. Open Sign up. 2. Enter invalid emails one at a time (e.g., invalidemail, user@domain, @domain.com user@com). 3. Attempt to submit each time.	Each invalid format is rejected with a clear field-level error. Form cannot be submitted with invalid email.	Pass		
REG-026	Registration	Validation	Weak password is rejected (if password policy exists)	User is logged out. Password rules are defined by the app.	1. Open Sign up. 2. Enter a weak password (e.g., 123, password). 3. Submit.	Registration is blocked and password rules are communicated clearly. No account is created.	Fail	Account submits on weak passwords	https://mofeezamasood.atlassian.net/browse/DEV-17
REG-027	Registration	Validation	Required fields empty → field-level errors	User is logged out.	1. Open Sign up. 2. Leave each required field empty one at a time (fill others valid). 3. Submit.	Submission is blocked. Each missing required field shows a clear error message and indicates how to fix it.	Pass		

Case ID	Section	Type	Title	Preconditions	Steps	Expected Result	Status	Notes	Defect Link
REG-028	Registration	Validation	Leading/trailing spaces handled consistently	User is logged out.	1. Enter values with leading/trailing spaces in text fields and email. 2. Submit. 3. If registration succeeds, verify stored values in profile.	System trims or validates consistently (per requirements). Email should not be stored with spaces; no duplicate accounts caused by whitespace.	Pass		
REG-029	Registration	Validation	Max length boundaries accepted	User is logged out.	1. For each text field, enter exactly the maximum allowed characters. 2. Submit.	Form accepts boundary values without truncation or errors; data stored intact.	Pass		
REG-030	Registration	Validation	Exceeding max length is rejected gracefully	User is logged out.	1. Enter text exceeding max length (e.g., 500+ chars) into each constrained field. 2. Submit.	Submission is blocked or field input is constrained with clear feedback. No server error/500 occurs.	Pass		
REG-031	Registration	Validation	Common valid email variants supported	User is logged out.	1. Attempt registration with email variants: user+tag@domain.com, user.name@domain.com, user_name@domain.com. 2. Submit each.	Emails are accepted if supported. If not supported, user sees a clear validation message (not a generic error).	Pass		
REG-032	Registration	Validation	Date of birth validation	User is logged out. DOB field exists.	1. Enter a future DOB. 2. Enter an implausible DOB (e.g., 1800-01-01). 3. Submit each.	Invalid DOB values are rejected with clear guidance (e.g., must be in the past, must be within realistic range).	Fail	Account is registered with invalid DOB	https://mofeezamasood.atlassian.net/browse/DEV-21

Case ID	Section	Title	Type	Preconditions	Steps	Expected Result	Status	Notes	Defect Link
SEARCH-001	Product Search	Keyboard navigation works for search, filters, sorting, pagination	Accessibility	User is on site.	1. Navigate search UI using keyboard only. 2. Apply a filter and sort. 3. Paginate results.	All interactive controls are keyboard accessible and usable with visible focus states.	Pass		
SEARCH-002	Product Search	Screen reader announces results count and filter changes	Accessibility	Screen reader available; results page includes live regions or announcements.	1. Run a search. 2. Apply a filter. 3. Listen for announcements. 4. Verify results summary is readable.	Results count and state changes are announced appropriately; UI is understandable without vision.	Pass		
SEARCH-003	Product Search	Cross-browser/device consistency for search	Compatibility	User is on site.	1. Repeat search on across target browsers/devices. 2. Compare results and layout.	Search UI and results behave consistently across supported environments.	Pass		
SEARCH-004	Product Search	Diacritics matching behavior (café vs cafe) is consistent	Functional	User is on site.	1. Search for term with diacritics. 2. Search for same term without diacritics. 3. Compare results.	Behavior matches requirements (accent-insensitive or accent-sensitive), consistently across pages.	Pass		
SEARCH-005	Product Search	Common term returns many results without performance degradation	Performance	Catalog has many matching products.	1. Search for a very common term. 2. Navigate multiple pages. 3. Measure response time and UI responsiveness.	Responses meet SLA; UI remains responsive; pagination works under load.	Pass		
SEARCH-006	Product Search	Price range filter returns only items within range	Functional	Filter controls for price range exist.	1. Perform a search that returns multiple items. 2. Apply a price min and max. 3. Verify prices in results.	All displayed products fall within selected price range; counts update correctly.	Fail	Is not applicable	
SEARCH-007	Product Search	Color filter works	Functional	Color filter exists.	1. Search for products. 2. Apply a color filter. 3. Verify results.	Only products matching the selected color (per defined logic) are shown.	Pass		
SEARCH-008	Product Search	Combined filters use intersection logic	Functional	Multiple filters exist.	1. Apply two or more filters (e.g., color + size + price). 2. Observe results and counts.	Results satisfy all applied filters (AND logic) unless requirements specify otherwise.	Pass		
SEARCH-009	Product Search	Out-of-stock indicator is correct	Functional	Some products are out of stock and UI indicates it.	1. Search/list products. 2. Find an out-of-stock item. 3. Verify label/button behavior.	Out-of-stock items are clearly labeled and purchase/add-to-cart actions behave correctly per policy.	Fail	Out of stock items are not sorted out from search filter	https://mofeezamasood.atlassian.net/browse/DEV-22
SEARCH-010	Product Search	Search by exact product name	Functional	User is on site (logged in or out).	1. Locate the search bar. 2. Enter an exact product name. 3. Submit search.	Results include the matching product. Result count and listing are accurate.	Pass		
SEARCH-011	Product Search	Search by partial keyword	Functional	User is on site.	1. Enter a partial keyword (substring) of a product name. 2. Submit.	Results include products matching the keyword; irrelevant items are minimized.	Pass		
SEARCH-012	Product Search	Search by multiple keywords	Functional	User is on site.	1. Enter multiple keywords related to a product. 2. Submit.	Results are relevant; ranking makes sense (most relevant near top).	Fail		
SEARCH-013	Product Search	Category-scoped search	Functional	Category filter exists.	1. Apply a category filter. 2. Enter a keyword. 3. Submit/search.	Results are limited to the selected category and match the keyword.	Pass		
SEARCH-014	Product Search	Search by product description keyword	Functional	Search supports description indexing.	1. Enter a keyword known to exist in a product description. 2. Submit.	Products with the keyword in descriptions are returned per requirements.	Pass		
SEARCH-015	Product Search	Search by SKU/product code	Functional	Products have SKU/codes and search supports it.	1. Enter a known product SKU/code. 2. Submit.	Exact match is returned and prioritized, or behavior matches requirements.	Pass	Search by Product Reference is not applicable.	
SEARCH-016	Product Search	Case-insensitive search	Functional	User is on site.	1. Search for a term in lowercase. 2. Search for same term in uppercase. 3. Compare results.	Results are identical (case-insensitive matching).	Pass		
SEARCH-017	Product Search	Special characters in query handled properly	Functional	User is on site.	1. Search using terms with hyphen/apostrophe (e.g., t-shirt, O'Brien). 2. Submit.	Search completes successfully; results are relevant; no errors due to special characters.	Pass		
SEARCH-018	Product Search	Autocomplete/suggestions	Usability	Autocomplete feature exists.	1. Type slowly into search bar. 2. Observe suggestions. 3. Click a suggestion.	Suggestions appear and are relevant; selecting one triggers appropriate results.	Pass		
SEARCH-019	Product Search	Empty/whitespace query behavior	Validation	User is on site.	1. Submit an empty search. 2. Submit a whitespace-only search.	App handles it gracefully: either blocks with message or shows default results per design; no errors.	Pass		
SEARCH-020	Product Search	Search → open product → back preserves state	Integration	User is on site.	1. Perform a search and apply filters/sort. 2. Open a product detail page from results. 3. Click Back to return to results.	User returns to the same results set with search term/filters/sort preserved; scroll position preserved if expected.	Pass		
SEARCH-021	Product Search	Add to cart from results	Integration	Add-to-cart is available on listing page.	1. Perform a search. 2. Add an item to cart directly from results. 3. Open cart and verify item. 4. Return to results.	Cart updates correctly; result page remains stable; item state (e.g., 'added') updates correctly.	Fail	not integrated	
SEARCH-022	Product Search	Search behavior consistent for logged-out vs logged-in users	Functional	A user account exists.	1. Search while logged out. 2. Log in. 3. Repeat the same search. 4. Compare results and UX.	Search works in both states. Any personalization differences are intentional and consistent.	Pass		
SEARCH-023	Product Search	No results shows friendly empty state	Usability	User is on site.	1. Search for a nonsense term (e.g., xyz123 nonexistent). 2. Submit.	User sees a clear 'no results' message with suggestions (e.g., check spelling, broaden search). No errors.	Pass		
SEARCH-024	Product Search	SQL injection-like query handled safely	Security	User is on site.	1. Enter SQL injection-like text into search (e.g., ' OR '1='1'). 2. Submit.	Search does not error; no stack traces; results are safe (likely no match).	Pass		

Case ID	Section	Title	Type	Preconditions	Steps	Expected Result	Status	Notes	Defect Link
SEARCH-025	Product Search	Extremely long search query handled gracefully	Security	User is on site.	1. Paste 1000+ characters into search box. 2. Submit.	App prevents or safely handles long input with clear message; no server error.	Pass		
SEARCH-026	Product Search	Special characters only query handled gracefully	Validation	User is on site.	1. Search for @#\$%^&*(). 2. Submit.	App handles gracefully: no crash; either no results or a validation message.	Pass		
SEARCH-027	Product Search	Stop-words-only query behavior is consistent	Usability	User is on site.	1. Search for common stop words (e.g., 'the and or'). 2. Submit.	Behavior matches requirements (may return broad results or show message). No errors.	Pass		
SEARCH-028	Product Search	Invalid price range (min > max or negative) is handled	Validation	Price filter exists.	1. Set min price higher than max. 2. Set negative values. 3. Apply filter.	User gets validation feedback or values auto-correct; no broken state or server errors.	Blocked	no testable	
SEARCH-029	Product Search	Pagination controls navigate correctly	Functional	Multiple pages of results exist.	1. Perform a broad search. 2. Click Next/Previous/page numbers. 3. Verify item changes and current page indicator.	Correct page loads with expected results and current page indicator updates.	Pass		
SEARCH-030	Product Search	Pagination boundaries disable appropriately	Usability	Multiple pages exist.	1. Go to first page and check Previous control. 2. Go to last page and check Next control.	Previous is disabled/hidden on first page; Next is disabled/hidden on last page; no errors when clicked.	Pass		
SEARCH-031	Product Search	Search response time meets SLA under typical use	Performance	User is on site.	1. Run 10 searches with representative keywords. 2. Measure response time from submit to results. 3. Note worst/average.	Search meets SLA; UI shows loading state; no timeouts under normal load.	Pass		
SEARCH-032	Product Search	Mobile filter UI is usable (touch targets, collapsible panels)	Compatibility	User is on mobile viewport.	1. Open results on small viewport. 2. Open filters panel. 3. Apply filters. 4. Close panel.	Filters are reachable; touch targets are adequate; panel does not block key actions; results update correctly.	Pass		
SEARCH-033	Product Search	Sort by price low to high	Functional	Sorting controls exist and products have prices.	1. Run a search with multiple results. 2. Sort by price low→high. 3. Verify order.	Prices are non-decreasing across the listing (ties handled consistently).	Fail	Sort is not functional	https://mofeezamasood.atlassian.net/browse/DEV-16
SEARCH-034	Product Search	Sort by price high to low	Functional	Sorting controls exist.	1. Run a search. 2. Sort by price high→low. 3. Verify order.	Prices are non-increasing across the listing.	Fail	Sort is not functional	https://mofeezamasood.atlassian.net/browse/DEV-16
SEARCH-035	Product Search	Sort by name A to Z	Functional	Sorting controls exist.	1. Run a search. 2. Sort by name A→Z. 3. Verify order.	Product names are in alphabetical order per locale rules.	Pass		
SEARCH-036	Product Search	Sort persists across pagination	Functional	Multiple pages of results exist.	1. Apply a sort. 2. Navigate to page 2. 3. Verify sorting still applied.	Sorting remains active across pages and URL/state reflects it if designed.	Blocked	not enough entries to test	
SEARCH-037	Product Search	Filters persist with browser back/forward	Usability	Filters exist.	1. Apply filters. 2. Open a product from results. 3. Use Back to return. 4. Use Forward if applicable.	Search term, filters, sorting, and scroll position are preserved (or restored) per expected UX.	Pass		
SEARCH-038	Product Search	Clear filters resets results state	Usability	Filters exist and clear/reset control exists.	1. Apply filters. 2. Click Clear/Reset. 3. Observe results and UI.	All filters are removed; results return to unfiltered state; UI indicates reset clearly.	Pass		