

Tt	TC ID	Test Case Title	Prerequisites	Steps to Reproduce	Expected Result	🔗	Priority	Test Type	Tt	Notes
	TC-AUTH-001	Verify successful user registration with valid data.	User is not already registered; Browser is open at demoblaze.com.	1. Click on "Sign up" in the navigation bar. 2. Enter a unique Username (e.g., tester_2025). 3. Enter a valid Password. 4. Click the "Sign up" button.	An alert should appear saying "Sign up successful." The user data is saved in the system.		high	Functional Testing		Ensure the username is truly unique to avoid "User already exists" error during testing.
	TC-AUTH-003	Verify successful login with valid registered credentials.	User has a pre-existing account.	1. Click on "Log in" in the navigation bar. 2. Enter the registered Username and Password. 3. Click the "Log in" button.	The login modal should close, and "Welcome [Username]" should appear in the navbar.		high	Functional Testing		Check if the session is maintained correctly after navigating between categories.
	TC-CART-032	Validate system behavior when adding a product without being logged in.	User is in Guest mode (Not logged in).	1. Select any product (e.g., Samsung Galaxy s6). 2. Click the "Add to cart" button.	The product should be added to the cart, and a confirmation alert "Product added" should appear. (Demoblaze allows guest carting).		medium	Boundary		In most e-commerce apps, this is allowed, but it's crucial to check if the cart merges upon login.
	TC-CART-030	Verify cart data persistence after browser refresh	At least one item is added to the cart.	1. Go to the "Cart" page and verify the item is listed. 2. Refresh the browser page (F5).	The item should still be visible in the cart table with correct price and title.		high	Regression Testing		This ensures that cart data is stored in Local Storage or IndexedDB
	TC-CHCK-038	Validate "Place Order" failure when the 'Name' field is empty.	Items are present in the cart	1. Click "Place Order" in the cart page. 2. Fill in all details (Country, City, Card, Month, Year). 3. Leave the "Name" field empty. 4. Click "Purchase".	An alert should appear: "Please fill out Name and Creditcard." No order should be placed.		high	Boundary Value Analysis (BVA)		Critical for data integrity; the backend should not process orders with null identity values.
	TC-NAV-006	Verify that selecting 'Phones' category displays only mobile devices.	User is on the homepage.	1. Locate the 'Categories' menu on the left sidebar. 2. Click on the 'Phones' link.	The product grid should update to show only phone-related items (e.g., Samsung Galaxy, iPhone, Nexus).		high	Functional Testing		Check if the active category is visually highlighted in the sidebar.
	TC-NAV-007	Verify navigation to Product Description Page (PDP).	Products are visible on the home screen.	Click on the title or image of "Samsung galaxy s6".	The user should be redirected to a new page showing the product image, price, and detailed description.		high	Functional Testing		Ensure the price on this page matches the price on the listing page.
	TC-UI-008	Verify 'Next' and 'Previous' buttons for product pagination.	User is on the home page.	1. Scroll to the bottom of the product list. 2. Click the 'Next' button. 3. After the new page loads, click 'Previous'.	Next' should load a new set of products. 'Previous' should return the user to the initial list.		medium	UI/UX Testing		Verify that 'Previous' is disabled on the first page.
	TC-CART-009	Verify adding different products to the cart in one session.	User is on the product listing.	1. Add a 'Phone' to the cart. 2. Navigate back to Home, select a 'Laptop', and add it to the cart. 3. Navigate to the 'Cart' page.	Both items should be listed in the cart table with their respective prices.		high	Functional Testing		Critical for e-commerce business logic.
	TC-CHCK-010	Validate 'Year' field in Place Order modal for future dates.	Items are in the cart.	1. Open 'Place Order' modal. 2. Enter "2020" (a past year) in the Year field. 3. Complete other fields and click 'Purchase'.	The system should ideally reject past years for credit card validation.		medium	Boundary Value Analysis (BVA)		Demoblaze is a simple demo, so it might accept it, which can be reported as a logical bug.
	TC-API-011	Verify API response for valid login credentials	Postman tool is open; API endpoint is known.	Send a POST request to /login with valid JSON payload (username/password)	tatus code 200 OK should be returned with a valid Auth		high	API Testing		Part of the API test suite mentioned in your CV.
	TC-UI-012	Verify all carousel banners load correctly.	Homepage is loaded.	1. Observe the slider at the top. 2. Wait for it to rotate or use the arrows to slide.	No broken images should appear; each banner should be clear and high-resolution.		low	UI Testing		Checked for broken image links (Status 404)
	TC-AUTH-013	Validate registration failure with an existing username.	A username is already registered (e.g., admin).	1. Click 'Sign up'. 2. Enter the existing username and a password. 3. Click 'Sign up'.	An alert should appear: "This user already exists."		medium	Negative Testing		Essential for database integrity.
	TC-CART-014	Verify the accuracy of the 'Total' price in the cart.	Two or more items are in the cart.	1. Add items worth \$360 and \$700. 2. Go to the 'Cart' page.	The 'Total' field should display "1060".		high	Functional Testing		Manual calculation must match the system's output.
	TC-REG-015	Verify that 'Close' button works for all modals (Login/Signup/Contact).	Any modal is open.	Click the 'Close' button or the 'X' at the top right.	The modal should disappear, returning the user to the main page without a refresh.		low	Regression Testing		Tested to ensure no UI "freezing" occurs after closing modals.

Tt	TC ID	Test Case Title	Prerequisites	Steps to Reproduce	Expected Result	🔗	Priority	Test Type	Tt	Notes
	TC-MOB-016	Verify the navigation menu functionality on mobile viewport.	Access to Chrome DevTools (Mobile Emulator).	1. Switch to Mobile View (e.g., iPhone 12). 2. Click the 'Hamburger' (three lines) icon in the top right.	The menu should expand smoothly, showing all navigation links (Home, Contact, About, Cart, Log in, Sign up).		high	Mobile Testing / UI		Critical for ensuring a seamless end-user experience on handheld devices.
	TC-CART-017	Verify that deleting an item updates the cart contents and total price.	Two items added to the cart.	1. Navigate to the 'Cart' page. 2. Click 'Delete' next to one product.	The item should disappear immediately, and the 'Total' price should decrease accordingly.		high	Functional Testing		Ensures the shopping cart management logic works correctly as stated in your project experience.
	TC-CHCK-018	Validate system response to invalid Month/Year format during checkout.	User is in the 'Place Order' modal.	1. Fill in Name and Card details. 2. Enter "13" in the Month field and "99" in the Year field. 3. Click 'Purchase'	The system should ideally validate the date format or prevent submission with impossible months.		medium	Boundary Value Analysis / Negative Testing		Used to identify potential gaps in data validation
	TC-API-019	Verify API response payload for category list retrieval.	Postman tool is configured for Demoblaze API.	Send a GET request to the categories endpoint.	Status 200 OK. The JSON payload structure must contain categories: 'phone', 'notebook', and 'monitor'.		medium	API Testing		Validates response codes and payload structures
	TC-UI-020	Verify that the 'About Us' video modal functions correctly.	User is on the homepage.	1. Click 'About us' in the navbar. 2.Click the 'Play' button in the modal.	The video should start playing without errors; the 'Close' button should stop the video and close the modal.		low	Functional / UI Testing		Essential for verifying all core flows mentioned in your automation framework.
	TC-NAV-021	Verify successful message submission through the 'Contact' modal.	Browser is open at demoblaze.com	1. Click 'Contact' in the navbar. 2. Enter Email, Name, and Message. 3. Click 'Send message'.	An alert should appear: "Thanks for the message!!"		medium	Functional Testing		Part of the end-to-end testing cycles you conducted.
	TC-AUTH-022	Validate login failure with unregistered credentials.	A username that has never been registered	1.Open Login modal. 2. Enter random username and password. 3. Click 'Log in'.	An alert should appear: "User does not exist."		high	Negative Testing		Ensures early identification of defects in the authentication flow.
	TC-UI-023	Verify that the price of an item is consistent between Home, PDP, and Cart.	A specific item is chosen (e.g., Nexus 6).	1.Note the price on Home page. 3. Open Product page. 2. Add to Cart and check Cart page.	Price must remain \$650 across all three screens.		high	Regression Testing		Verifies data integrity across system outputs.
	TC-CART-024	Verify the 'Cart' page display when no items have been added.	New session; Cart is empty.	Click 'Cart' in the navbar.	The cart table should be empty, and the 'Total' should be blank or 0.		low	Functional Testing		Checks the initial state of the shopping cart module.
	TC-API-025	Verify API response when deleting a product by ID	Postman tool; valid item ID.	Send a POST/DELETE request to remove an item.	Status 200 OK or 204 No Content.		medium	API Testing		Implements automated test runs for API regression
	TC-CB-026	Verify UI layout consistency between Chrome and Firefox browsers	Both Chrome and Firefox browsers installed.	1. Open demoblaze.com in Chrome and Firefox simultaneously. 2. Compare the alignment of the navbar, product cards, and footer.	The UI should be identical in both browsers; no overlapping text or misaligned buttons		medium	Cross-Browser Testing		Supports your skill in ensuring seamless end-user experiences across different environments.
	TC-SEC-027	Validate system resistance to SQL Injection in the login fields.	User is on the Login modal.	1. In the username field, enter: ' OR '1'='1'. 2. Enter any password and click 'Log in'.	The system should not log the user in; it should treat the input as a plain string or show an error.		Critical	Security Testing		Vital for protecting user data
	TC-PERF-028	Verify that the homepage loads within acceptable time limits.	Clear browser cache.	1. Open browser and enter URL. 2. Use DevTools (Network tab) to measure 'Load' time.	The page should be fully interactive within 3 seconds on a standard 4G/High-speed connection.		medium	Non-Functional / Performance Testing		Related to your ability to identify potential risks and gaps in performance.
	TC-UI-029	Verify the 'About Us' modal can be closed using the 'Escape' key.	About Us' modal is open.	Press the 'ESC' key on the keyboard	The modal should close immediately.		low	UI/UX Testing		Testing keyboard accessibility, which is part of ensuring a comprehensive U

Tt	TC ID	Test Case Title	Prerequisites	Steps to Reproduce	Expected Result	🔗	Priority	Test Type	Tt	Notes
	TC-NAV-030	Verify system behavior when no products exist in a category.	Use a direct link to a non-existent category (if applicable).	Select a category that is expected to be empty (if any).	The page should display a clear "No products found" message rather than a 404 error or a blank screen.		medium	Negative Testing		Checks for graceful error handling, a key QA practice
	TC-API-031	Verify API response for checkout request without a valid Auth token.	Postman tool; User is not logged in.	Send a POST request to the purchase endpoint without the Authorization header.	Status 401 Unauthorized should be returned.		high	API Security Testing		validating response codes and error handling in Postman.
	TC-CART-032	Verify adding the same product multiple times to the cart.	User is on a product page.	1. Click 'Add to cart'. 2. Click 'Add to cart' again for the same product. 3. View Cart.	The cart should list the product twice or update the quantity (Demoblaze usually lists them as separate entries).		high	Functional Testing		Core business logic for an e-commerce site
	TC-UI-033	Verify that all buttons use the consistent 'Success' (Green) color theme.	Access to different pages (Home, Cart, Product).	Inspect the 'Log in', 'Add to Cart', and 'Purchase' buttons.	All action buttons should follow the site's design system (matching Hex codes).		low	UI Testing		Ensures adherence to quality standards as part of your Agile ceremony participation
	TC-SEC-034	Verify the Contact Form handles HTML tags safely.	Contact' modal is open.	1. In the message field, enter: <h1>Test</h1>. 2. Submit the form.	The system should treat the input as text, not render the HTML header.		medium	Security Testing		Checks for XSS (Cross-Site Scripting) vulnerabilities.
	TC-REG-035	Verify all navbar links remain functional after a user logs in.	User just logged in successfully.	Click on 'Home', 'About Us', and 'Cart' after the 'Welcome' message appears.	All links should redirect to the correct pages without breaking the session.		high	All links should redirect to the correct pages without breaking the session.		Essential to ensure no "regression" bugs were introduced by the login state.
	TC-API-036	Verify that the product list API responds within 500ms	Postman; Collection Runner.	1. Send a GET request to /entries. 2. Check the 'Time' in the Postman response window.	Time should be less than 500ms		medium	API Performance Testing		Demonstrates your advanced use of Postman for efficient regression testing.
	TC-UI-037	Verify manual navigation of the homepage carousel via side arrows.	Homepage is open.	1. Click the 'Next' (Right) arrow on the slider. 2. Click the 'Previous' (Left) arrow.	The slider should move to the next/previous image immediately.		low	Functional / UI Testing	Notes	
	TC-AUTH-038	Validate registration with special characters in the username.	Sign up modal is open.	1. Enter username: user!@#\$. 2. Enter password and click 'Sign up'.	The system should either accept it (if allowed) or provide a clear validation error about character limits.		low	Negative Testing	Notes	
	TC-UI-039	Verify the alignment of Pic, Title, Price, and Delete columns in the cart.	Items are in the cart.	Navigate to 'Cart'.	Headers should match the data rows perfectly; text should not be cut off.		low	UI Testing	Notes	
	TC-AUTH-040	Verify that a user can log out successfully while being on the 'Cart' page.	User is logged in and on the /cart.html page.	Click 'Log out'.	The user should be logged out and ideally redirected to the homepage or the login button should reappear.		high	Functional Testing	Notes	
	TC-DB-041	Verify that a new user record is correctly inserted into the database.	Access to the database/backend logs; New user registered.	1. Register a new user via the UI. 2. Execute: SELECT * FROM users WHERE username = 'new_tester';	The database should return one row containing the correct username and a hashed password.		high	Database Testing		MySQL and Database testing.
	TC-NAV-042	Verify system stability when viewing items with extremely high prices.	User is on the Laptops category	1. Click on "MacBook Pro" or the most expensive item. 2. Check if the price renders correctly without UI breaking.	The price (\$1100) should be formatted correctly with the "\$" symbol.		low	Functional Testing		Ensures seamless end-user experience for premium products
	TC-SEC-043	Verify that a Guest user cannot proceed to checkout without data.	Fresh browser session; No items in cart.	1. Manually type demoblaze.com/cart.html in the address bar. 2. Click 'Place Order'.	While the page opens, the "Place Order" button should ideally be disabled or show a message that the cart is empty.		medium	Security Testing		Part of identifying gaps and potential risks

Tt	TC ID	Test Case Title	Prerequisites	Steps to Reproduce	Expected Result	🔗	Priority	Test Type	Tt	Notes
	TC-E2E-044	Verify the complete flow from Login to Order Confirmation.	Registered user credentials.	Login -> Add "Iphone 6" -> Go to Cart -> Place Order -> Fill Data -> Purchase.	Every step should transition correctly, ending with a success alert and an empty cart.		Critical	End-to-End (E2E) Testing		This is the core flow automated in Java/Selenium framework.
	TC-DB-045	Verify that cart records are removed from the DB/Session after successful purchase.	Check the cart table in the DB for the current session ID.	The records associated with the completed order should be deleted or marked as "purchased".	The records associated with the completed order should be deleted or marked as "purchased".		high	Database Testing	Notes	
	TC-UI-046	Verify that all footer information (About Us, Get in Touch) is visible and accurate.	Scroll to the bottom of any page.	Inspect the footer section.	Contact info and the "Product Store" logo should be clearly visible and correctly aligned.		low	UI Testing	Notes	
	TC-REG-047	Verify that the system handles browser alerts correctly during signup.	Signup modal is open.	Submit an empty signup form.	A browser alert "Please fill out Username and Password" must appear and be dismissible.		medium	Regression Testing	Notes	
	TC-NAV-048	Validate the "Contact Message" field with a very large text input (1000+ characters).	Contact modal is open.	Paste a very long text into the 'Message' area and send.	The system should either truncate the text or handle the large payload without a 500 Server Error.		low	Negative Testing	Notes	
	TC-API-049	Verify API response for a single product using a valid ID.	Postman tool; Product ID (e.g., 1)	Send GET request to /view?id=1.	Status 200 OK; Response must contain "Samsung galaxy s6".		medium	API Testing	Notes	
	TC-UI-050	Verify that the "Home" link in the navbar is highlighted when on the homepage.	User is on the main landing page.	Observe the 'Home' link style.	The 'Home' link should have an "active" CSS class (e. g., bold or different color)		low	UI Testing	Notes	

TT	Test Case ID	Test Case Title	Actual Result	🔍 Status	TT Defect ID
	TC-001	User Registration (Valid)	Account created; alert "Sign up successful" appeared.	Pass	-
	TC-002	Duplicate Registration	Alert "This user already exists" appeared.	Pass	-
	TC-003	User Login (Valid)	Welcome [User] displayed in navbar.	Pass	-
	TC-004	Login (Invalid Password)	Alert "Wrong password" appeared.	Pass	-
	TC-005	Login (Empty Fields)	Alert "Please fill out Username and Password" appeared.	Pass	-
	TC-006	Category Filter (Phones)	Only phones displayed correctly.	Pass	-
	TC-007	Category Filter (Laptops)	Only laptops displayed correctly.	Pass	-
	TC-008	Category Filter (Monitors)	Only monitors displayed correctly.	Pass	-
	TC-009	Product Details Page (PDP)	Images and description match the selected product.	Pass	-
	TC-010	Carousel Auto-rotation	Images slider moves every 5 seconds.	Pass	-
	TC-011	Add to Cart (Single Item)	Confirmation alert appeared; item added.	Pass	-
	TC-012	Add to Cart (Guest Mode)	Item added even without login.	Pass	-
	TC-013	Multiple Items in Cart	All items listed correctly in the cart table.	Pass	-
	TC-014	Cart Total Calculation	Sum matched individual item prices.	Pass	-
	TC-015	Delete Item from Cart	Item removed; total price updated.	Pass	-
	TC-016	Cart Persistence (Refresh)	Items stayed in cart after F5 refresh.	Pass	-
	TC-017	Empty Cart State	Cart table is empty for new users.	Pass	-
	TC-018	Place Order Modal (UI)	Form opened with all fields (Name, Card, etc).	Pass	-
	TC-019	Checkout (Empty Name)	System allowed purchase even with empty Name.	Fail	BUG-001
	TC-020	Checkout (Empty Card)	Purchase processed without credit card number.	Fail	BUG-002
	TC-021	Purchase Success Alert	"Thank you for your purchase!!" alert appeared.	Pass	-
	TC-022	Cart Clear after Purchase	Cart became empty after successful checkout.	Pass	-
	TC-023	Contact Form Submission	Alert "Thanks for the message!!" appeared.	Pass	-
	TC-024	Contact Form (Invalid Email)	System accepted invalid email format (no @).	Fail	BUG-003
	TC-025	About Us Video Playback	Video plays and pauses correctly in modal.	Pass	-

Tt	Test Case ID	Test Case Title	Actual Result	☑ Status	Tt Defect ID
	TC-026	Logout Functionality	Session ended; navbar returned to guest state.	Pass	-
	TC-027	Browser Back (After Logout)	Could access user profile via Back button.	Fail	BUG-004
	TC-028	SQL Injection (Login)	Malicious input rejected; no login allowed.	Pass	-
	TC-029	XSS in Contact Message	Script tag <script> executed in the alert.	Fail	BUG-005
	TC-030	Mobile View (Hamburger Menu)	Menu expands/collapses on mobile screen.	Pass	-
	TC-031	Responsive Images (Mobile)	Product images are stretched on iPhone SE.	Fail	BUG-006
	TC-032	Cross-Browser (Firefox)	UI layout is consistent with Chrome.	Pass	-
	TC-033	API: Login Endpoint	Status 200 OK returned in Postman.	Pass	-
	TC-034	API: Get Products List	JSON payload contains correct data structure.	Pass	-
	TC-035	API: Unauthorized Purchase	Status 401 Unauthorized for guest checkout.	Pass	-
	TC-036	DB: User Insert (SQL)	Record found in 'users' table after signup.	Pass	-
	TC-037	DB: Cart Table Update	Items saved in DB under the correct session ID.	Pass	-
	TC-038	Navbar Links (Post-Login)	All links (Home, Cart) work after login.	Pass	-
	TC-039	Checkout Year (Future Date)	System accepted years like 2026/2027.	Pass	-
	TC-040	Checkout Year (Past Date)	System accepted year "2015" for credit card.	Fail	BUG-007
	TC-041	Pagination (Next Button)	Next page of products loaded correctly.	Pass	-
	TC-042	Pagination (Prev Button)	Returned to previous set of products.	Pass	-
	TC-043	Footer Information	Contact and social info visible on all pages.	Pass	-
	TC-044	Modal Close (ESC Key)	Modals closed when pressing Escape key.	Pass	-
	TC-045	Page Load Time (Home)	Loaded in 2.1s (using Network Tab).	Pass	-
	TC-046	Logo Redirection	Clicking logo returns user to homepage.	Pass	-
	TC-047	Password Masking	Characters hidden while typing in login.	Pass	-
	TC-048	Special Chars in Username	Handled correctly by registration system.	Pass	-
	TC-049	API: Delete Item	Status 200 OK when removing product via ID.	Pass	-
	TC-050	End-to-End Flow	Full cycle (Login -> Cart -> Purchase) passed.	Pass	-

T#	Defect ID	Summary (Bug Title)	🔍 Severity	Priority	Type	Suggested Fix
	BUG-001	Purchase processed with empty "Credit Card" field	Critical	High	Functional	Add Mandatory field validation & Regex for 16-digit CC numbers.
	BUG-002	Session remains active after Logout when using Back button	High	High	Security	Implement Cache-Control: no-store and clear Session Tokens on logout.
	BUG-003	Internal Server Error (500) when Name exceeds 500 chars	Critical	High	Functional	Add Character Limit (e.g., 50) and Server-side Input Validation.
	BUG-004	Contact Form accepts invalid email format (e.g., "test.com")	Low	Medium	Functional	Implement Front-end Email Validation (e.g., type="email" or Regex).
	BUG-005	XSS Vulnerability: Script executes in Contact Message alert	High	High	Security	Sanitize HTML inputs and use Output Encoding to prevent script injection.
	BUG-006	Place Order allows purchase with a past Year (e.g., 2015)	Medium	High	Functional	Add Date Validation logic to ensure Year >= Current Year.
	BUG-007	Hamburger menu icon overlaps with Logo on iPhone SE view	Medium	Medium	UI/UX	Adjust CSS Media Queries and Z-index for the navigation bar.
	BUG-008	About Us video continues playing after closing the modal	Medium	Low	UI/UX	Add a JavaScript Event Listener to pause/stop video on modal hide.
	BUG-009	API returns 500 Error instead of 404 for non-existent ID	High	Medium	API	Handle null responses and return 404 Not Found status code.
	BUG-010	Carousel images aspect ratio is distorted on small screens	Low	Low	UI	Use CSS object-fit: cover to maintain image proportions.
	BUG-011	Next button in pagination is functional even on last page	Low	Medium	Functional	Disable the "Next" button when current_page == total_pages.
	BUG-012	Log in modal doesn't close when clicking outside the box	Low	Low	UI/UX	Enable "Dismiss on backdrop click" in the Modal component settings.
	BUG-013	Database does not update "Last Login" timestamp correctly	Medium	Medium	Database	Add a Trigger or Update Query in the backend during login success.
	BUG-014	Items in Cart are lost if the user clears browser session	Medium	High	Functional	Use LocalStorage or IndexedDB to persist cart data for guests.
	BUG-015	Prices in "Cart" do not match Prices on "Home" page	High	High	Functional	Ensure the Cart API pulls prices dynamically from the Products Table.
	BUG-016	Social media icons in Footer lead to 404 (Broken Links)	Low	Medium	UI	Update the <a> tag href attributes with valid URLs.
	BUG-017	User can sign up with a username consisting only of spaces	Medium	Medium	Functional	Apply .trim() to the input and check for length > 0.
	BUG-018	API Response time for /entries exceeds 2 seconds	Medium	Medium	Performance	Optimize DB queries and implement Pagination or Caching for products.
	BUG-019	Cart "Total" shows incorrect sum for items > 10	High	High	Functional	Fix the floating-point calculation logic in the JavaScript total function.
	BUG-020	Delete button in Cart lacks a confirmation message	Low	Medium	UI/UX	Add a window.confirm() or a Modal before executing deletion.
	BUG-021	Sign up password field shows plain text during typing	High	High	Security	Change input type from text to password.
	BUG-022	Contact message sends "undefined" if Name field is null	Medium	Medium	Functional	Add a conditional check to prevent sending if fields are null.
	BUG-023	Search/Filter fails to display results for "Monitors" category	High	High	Functional	Check the DB query filter for the "Monitors" string mapping.
	BUG-024	Text overlapping in "About Us" modal on tablet view	Low	Low	UI	Use Responsive Flexbox or Grid for better content flow.
	BUG-025	System allows adding a product with \$0 price via API injection	Critical	High	API/Security	Add Server-side price verification before processing orders.
	BUG-026	User registration allows duplicate emails (if email used)	Medium	Medium	Database	Add a UNIQUE constraint to the email column in the database.

Tt	Defect ID	Summary (Bug Title)	Severity	Priority	Type	Suggested Fix
	BUG-027	Broken image placeholder for "Sony vaio i7" on Firefox	Medium	Medium	UI	Ensure cross-browser compatibility for image rendering/paths.
	BUG-028	Success alert "Thank you" stays on screen indefinitely	Low	Low	UI/UX	Add a setTimeout function to auto-hide the alert after 5 seconds.
	BUG-029	API Post request to /addcart fails without error message	High	Medium	API	Return a clear JSON Error Message and proper status code (e.g., 400).
	BUG-030	Home navbar link is not highlighted while on homepage	Low	Low	UI	Use a JavaScript function to add the .active class to the current link.