QA TEST WEB APPLICATION — TEST DOCUMENTATION

Base URL: https://qa-test-web-app.vercel.app
Github URL: https://github.com/anamarija-tomic/qa-test-web-app
Author: Anamarija Tomić
Date: 10 Feb 2026

Table of Contents

TEST PLAN

1. Introduction

This document defines the test approach, scope, environment, scenarios, test cases, and defects for the QA Test Web Application.
Testing is performed as black-box UI testing with emphasis on:

- Registration and input validation quality (positive + negative)
- Authentication/session correctness (login/logout/direct URL access)
- Forgot Password flow completeness and consistency
- Cross-viewport usability and visibility of validation feedback

2. Scope

2.1 In Scope

- **Registration** (/register.html): required fields, input validation, whitespace handling, Terms acceptance, user feedback/messages, redirect behavior
- **Login** (/index.html): authentication behavior, error handling, feedback/messages, redirect to Dashboard
- **Forgot Password** (/forgot-password.html): email validation, non-existing email handling, Security Question/Answer consistency (end-to-end completeness), feedback/messages
- **Dashboard access control** (/dashboard.html): authenticated access, direct URL access, “Last login” semantics, dashboard actions behavior (real change vs toast)
- **Logout & session handling**: session invalidation, post-logout access restriction, storage behavior (sessionStorage/localStorage)
- **Responsive behavior** (high-level): visibility/usability of critical inputs, validation messages, and key controls across Desktop/Tablet/Mobile

2.2 Out of Scope

- Backend/internal implementation details beyond observable UI behavior
- Long-term persistence guarantees (environment appears temporary/in-memory)
- Real email delivery (reset/newsletter emails)
- Advanced security testing (penetration testing, encryption review, vulnerability scanning)
- Performance/load testing

3. Acceptance Criteria (Target Product Behavior)

These represent expected behavior for a production-ready system. Deviations are logged as defects.

Registration

- Registration succeeds with valid data and redirects to Login (/index.html?registered=true).
- Required fields enforce non-empty input after trimming (whitespace-only is treated as empty).
- Invalid emails are rejected with clear email field-level feedback.
- Password policy is enforced: minimum 8 characters; must include uppercase + lowercase + number (special character optional).
- Confirm Password must match Password (mismatch blocks submission with clear feedback).
- Phone rejects letters/invalid symbols; supports reasonable length; accepts common separators (+, space, -, parentheses).
- ZIP rejects letters/whitespace-only; supports digits-only and reasonable length.
- Terms & Conditions acceptance is mandatory to register.
- Success and failure flows provide consistent user feedback (success message on success; clear error message(s) on failure).

Login / Session

- Login accepts valid credentials and redirects to Dashboard.
- Login rejects invalid credentials with accurate feedback (no false success).
- Dashboard is blocked without a valid authenticated session; direct URL access (/dashboard.html) is blocked.
- Logout fully invalidates authentication state (no access to Dashboard via direct URL after logout).
- Authentication state storage is consistent: logging in as a new user must not retain the previous user in storage or rehydrate stale identity.

Forgot Password

- Forgot Password does not confirm success for non-existing emails (must show a clear error).
- If Security Question/Answer is required in reset UI, it must be collected and stored during registration; otherwise it must not be required/present in reset flow.
- Failure scenarios provide clear submit-level feedback (not “silent”).

UI/UX (Cross-Viewport)

- Validation feedback is visible/readable on Desktop/Tablet/Mobile.
- Inputs and key controls remain visible and interactable; overlays must not obscure required inputs or critical information.
- Disabled-looking controls must not be clickable; interactive state must match visual state.

4. Test Types

- Functional testing
- Negative testing / validation testing
- UI/UX testing
- Responsive testing (Desktop/Tablet/Mobile)
- Session/auth testing (login, logout, direct URL access)

5. Test Environment

5.1 Manual Testing Environment

Item	**Details**
Application URL	https://qa-test-web-app.vercel.app
Browsers	Google Chrome (latest)
Operating System	Windows 11 Pro
Viewports	Desktop: 1920×1080; Tablet: 768–1024 px (DevTools emulation); Mobile: ≤767 px (DevTools emulation)
Tools	Browser DevTools

5.2 Automation Environment (Playwright)

Item	**Details**
Operating System	Windows 11 Pro
Node.js	v24.13.0
npm	11.6.2
Playwright	1.58.1
Framework	Playwright Test (JavaScript)
Design	Page Object Model (POM) + data-driven test data helper(s)
Reporter	Playwright HTML Reporter

Projects	chromium (Desktop Chrome): all functional/session/dashboard/registration/login/forgot-password tests + UX tests (desktop);
	tablet (iPad gen 7): UX tests only (tests/ux/);
	mobile (Pixel 5): UX tests only (tests/ux/)

Execution	workers = 1 (to reduce flakiness caused by shared state/race conditions during user creation and immediate navigation)

6. Test Scenarios

6.1 Registration

- Happy path registration
- Missing required fields
- Invalid email formats
- Weak password / policy enforcement
- Confirm password mismatch
- Phone validation
- ZIP validation
- Text-field data quality (names/city/address)
- Terms enforcement
- Whitespace-only handling (per-field)
- Feedback consistency (success vs failure)

6.2 Login

- Happy path login
- Invalid credentials
- Non-existing user
- Empty fields
- Feedback consistency
- Remember Me behavior

6.3 Forgot Password

- Existing user reset request (expects end-to-end security verification design)
- Non-existing email handling
- Security Q/A validation (if present)
- Feedback consistency (success vs failure)

6.4 Dashboard & Session

- Authenticated vs direct URL access
- Logout invalidation
- Storage-driven “zombie user” rehydration scenario
- Dashboard action buttons: real operations vs success toasts
- “Last login” data semantics

6.5 UI/UX & Responsive

- Page usability across viewports
- Visibility/readability of validation feedback across viewports
- Disabled-looking controls that remain clickable
- Overlays obstructing content/controls

7. Entry & Exit Criteria

7.1 Entry Criteria

- Application accessible via provided URL
- Required pages available
- Test environment stable
- Test data can be created via UI

7.2 Exit Criteria

- Manual test cases executed (Appendix A)
- Defects documented with severity/priority (Appendix B)
- Automation suite executed and results captured (HTML report)
- Known limitations and workarounds documented

8. Risks & Assumptions

8.1 Risks

- Responsive layout may hide errors/controls on smaller viewports
- Simulated success messages may mask real failures
- Limited validation can produce false positives in registration/login/reset flows

8.2 Assumptions

- Storage/persistence is temporary or inconsistent
- Email sending is simulated
- Testing focuses on observable UI behavior and storage-driven session logic

9. Deliverables

- This Test Plan
- Manual Test Cases (Appendix A)
- Defect Report (Appendix B)
- Automation Execution Report (Appendix C)
- Playwright automation suite + execution report (HTML) — generated locally and published as GitHub Actions artifact
- Repository README (separate deliverable)

10. Test Metrics (Current Snapshot)

- Viewport coverage: TC-UX-001 to TC-UX-006 executed on Desktop/Tablet/Mobile
- Defects logged: 26 (BUG-001 to BUG-026)

Defects by severity (current):

- Critical: 1
- High: 7
- Medium/High: 3
- Medium: 6
- Low/Medium: 4
- Low: 5

11. Manual Test Execution Summary

Manual testing covered: /index.html, /register.html, /forgot-password.html, /dashboard.html across Desktop/Tablet/Mobile.
Exact pass/fail counts were not tracked due to exploratory iterations and repeated input variations. Confirmed issues are documented in Appendix B; scripted manual test cases are documented in Appendix A.

Key findings:

- Validation weaknesses allow invalid/low-quality input to pass (email/password/confirm password/trim/sanitization/text-field quality)
- Session invalidation and storage syncing enable dashboard access after logout and "zombie user" rehydration
- Mobile viewport hides validation errors, causing silent failures and poor guidance
- Forgot Password requests Security Q/A that is never collected during registration
- Dashboard actions show success toasts without real operations; "Last login" semantics misleading

12. Automation Approach (Implemented)

Automation uses:

- Page Object Model (POM) for maintainability
- Data-driven helper for registration data generation (unique email per run)
- Project split: functional tests on Desktop Chromium; UX-only tests on tablet/mobile
- workers=1 to reduce shared-state/race flakiness after registration

Known automation constraint/workaround:

- For negative submit cases where the UI provides no submit-level failure message, automation uses a timed redirect guard: after Submit, wait ~2000ms and assert that the success redirect did NOT occur. (Impacted by BUG-026.)
- CI execution: GitHub Actions workflow runs the full suite on push/PR and uploads the Playwright HTML report as an artifact

APPENDIX A — TEST CASES

A.1 Test Case Index

Test Case ID	**Title (short)**	**Area**	**Related Bug(s)**
TC-REG-001	Register with valid data (Happy Path)	Registration	—
TC-REG-002	Registration email validation (invalid formats)	Registration	BUG-002
TC-REG-003	Password policy enforcement	Registration	BUG-003
TC-REG-004	Confirm Password must match Password	Registration	BUG-004
TC-REG-005	Phone validation (allowed characters only)	Registration	BUG-005, BUG-007
TC-REG-006	ZIP validation (numeric only)	Registration	BUG-006, BUG-007
TC-REG-007	Text fields data quality validation	Registration	BUG-008
TC-REG-008	Terms and Conditions must be required	Registration	BUG-009
TC-REG-009	Whitespace-only input treated as empty (required fields)	Registration	BUG-007, BUG-026
TC-LOG-001	Login with valid credentials (Happy Path)	Login	—
TC-LOG-002	Login rejects invalid credentials	Login	—
TC-LOG-003	Remember Me functionality (persistence behavior)	Login	BUG-010
TC-SEC-001	Logout invalidates session and blocks direct dashboard access	Session/Security	BUG-001
TC-SEC-002	New Login Overwrites Previous User Auth State (No Storage Mismatch)	Session/Security	BUG-001
TC-FP-001	Forgot Password for existing user (Happy Path expectation)	Forgot Password	BUG-013
TC-FP-002	Forgot Password for non-existing email returns error	Forgot Password	BUG-012
TC-FP-003	Forgot Password validates security question and answer	Forgot Password	BUG-013

Test Case ID	**Title (short)**	**Area**	**Related Bug(s)**
TC-DASH-001	Update Profile updates user data and Recent Activity	Dashboard	BUG-015
TC-DASH-002	Dashboard actions perform real operations	Dashboard	BUG-016
TC-DASH-003	“Last login” shows previous login time	Dashboard	BUG-017
TC-UX-001	Login page UX: content visible and usable	UX (All viewports)	BUG-019
TC-UX-002	Registration page UX: all required inputs usable	UX (All viewports)	BUG-018, BUG-020
TC-UX-003	Forgot Password page UX: controls visible and usable	UX (All viewports)	BUG-025
TC-UX-004	Dashboard UX: content readable and actions usable	UX (All viewports)	BUG-021 – BUG-024
TC-UX-005	Validation feedback visible/readable (cross-page, cross-viewport)	UX (All viewports)	BUG-014
TC-UX-006	Submit-level failure feedback is shown consistently	UX/Feedback (All viewports)	BUG-026

A.2 Baseline Test Data

A.2.1 Baseline Valid Registration Data

Field	**Value**
First Name	Ana
Last Name	Tomic
Email	ana.tomic.<unique>@example.com (unique value generated per run)
Phone	+385 91 123 4567
Street Address	Ilica 1
City	Zagreb
ZIP Code	10000
Password	Password123
Confirm Password	Password123
Terms & Conditions	Checked

A.2.2 Baseline Valid Login Credentials (created via TC-REG-001)

Field	Value
Email	<registered email from TC-REG-001>
Password	<registered password from TC-REG-001>

A.2.3 Security Question/Answer (End-to-End Expectation Reference)

Field	Value
Security Question	What was your first pet’s name?
Security Answer	Rex

Note: Registration does not collect Security Q/A; end-to-end reset verification cannot be satisfied until implemented.

A.3 Detailed Test Cases

A.3.1 Registration

A.3.1.1 TC-REG-001 — Register with Valid Data (Happy Path)

Item	Details
Objective	Verify a user can register successfully using valid inputs and is redirected to Login.
Execution Viewports	Desktop
Preconditions	Application accessible; use unique email per run.
Test Data	Baseline valid registration data (A.2.1).
Steps	1. Open /register.html 2. Populate all required fields with valid data 3. Check Terms and Conditions 4. Click “Create Account”
Expected Result	• Success message displayed (e.g., “Registration successful! Redirecting to login…”) • Redirect to /index.html?registered=true

A.3.1.2 TC-REG-002 — Registration Email Validation (Invalid Formats)

Item	Details
Objective	Verify invalid email formats are blocked and feedback is shown.
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1 for all fields except Email.

Test Data (repeat per case):

Case #	Email Input	Expected
1	test	Blocked
2	test@	Blocked
3	@test.com	Blocked
4	test@com	Blocked
5	a@@b.com	Blocked
6	hello a@b world	Blocked
7	a@b	Blocked

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Enter Email per case 4. Click “Create Account”
Expected Result	• Registration blocked • Email validation feedback shown • No redirect to /index.html?registered=true

A.3.1.3. TC-REG-003 — Password Policy Validation (Length + Complexity)

Item	Details
Objective	Verify password policy (≥8 chars, uppercase+lowercase+number)
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1 except Password/Confirm Password.

Test Data:

Case	Password	Confirm Password	Expected
1	Pass12	Pass12	Blocked (too short)
2	password123	password123	Blocked (no uppercase)
3	PASSWORD123	PASSWORD123	Blocked (no lowercase)
4	Password	Password	Blocked (no number)
5	Password123	Password123	Allowed

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Enter Password/Confirm per case 4. Click “Create Account”
Expected Result	• Blocked cases: no redirect; feedback shown • Allowed case: success + redirect to Login

A.3.1.4 TC-REG-004 — Confirm Password Must Match Password

Item	Details
Objective	Verify mismatch blocks registration.
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1 except Password/Confirm Password.

Test Data:

Case	Password	Confirm Password	Expected
1	Password123	Different123	Blocked
2	Password123	Password123	Allowed

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Enter Password/Confirm per case 4. Click "Create Account"
Expected Result	• Mismatch: blocked + feedback; no redirect • Match: success + redirect

A.3.1.5 TC-REG-005 — Phone Validation (Allowed Characters)

Item	Details
Objective	Verify Phone rejects letters/invalid symbols and enforces reasonable length.
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1 except Phone.

Test Data:

Case	Phone Input	Expected
1	abcXYZ	Blocked
2	12ab34	Blocked
3	!!!!	Blocked
4	123	Blocked (too short)
5	+123 123 123 123 123 123	Blocked (too long)
6	+385/91-123 4567	Allowed
7	+385 91 123 4567	Allowed

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Enter Phone per case 4. Click “Create Account”
Expected Result	• Blocked cases: no redirect; feedback shown • Allowed cases: success + redirect

A.3.1.6 TC-REG-006 — ZIP Validation (Numeric Only)

Item	Details
Objective	Verify ZIP rejects letters/whitespace-only and enforces length.
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1 except ZIP Code.

Test Data:

Case	ZIP Input	Expected
1	ABCD	Blocked
2	12AB	Blocked
3	12	Blocked (too short)
4	12345678910	Blocked (too long)
5	10000	Allowed

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Enter ZIP per case 4. Click “Create Account”
Expected Result	• Blocked cases: no redirect; feedback shown • Allowed case: success + redirect

A.3.1.7 TC-REG-007 — Text Fields Data Quality Validation (Names/City/Address)

Item	Details
Objective	Verify basic data quality for First Name, Last Name, City, Address.
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1; override ONE field per case.

Test Data:

Group	Case	Field	Input	Expected
A	A1	First Name	@@@	Blocked
A	A2	Last Name	@@@	Blocked
A	A3	City	@@@	Blocked
A	A4	Street Address	@@@	Blocked

Group	Case	Field	Input	Expected
B	B1	First Name	A	Blocked
B	B2	Last Name	B	Blocked
B	B3	City	Z	Blocked
B	B4	Street Address	1	Blocked
C	C1	First Name	12345	Blocked
C	C2	Last Name	12345	Blocked
C	C3	City	12345	Blocked
C	C4	Street Address	12345	Blocked
D	D1	First Name	Ana123	Blocked
D	D2	Last Name	Tomic123	Blocked
D	D3	City	Zagreb123	Blocked
D	D4	Street Address	Ilica 1	Allowed

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Replace field under test with case input 4. Click “Create Account”
Expected Result	• Invalid patterns blocked with feedback • Realistic address with alphanumeric accepted
Notes	Missing field-level validation/feedback is tracked under BUG-008 and BUG-026.

A.3.1.8 TC-REG-008 — Terms and Conditions Must Be Required

Item	Details
Objective	Verify registration is blocked if Terms not accepted.
Execution Viewports	Desktop
Preconditions	Application accessible.
Baseline	Use A.2.1; vary Terms.

Test Data:

Case	Terms Checked?	Expected
1	No	Blocked
2	Yes	Allowed

Item	Details
Steps	1. Open /register.html 2. Fill required fields with baseline valid data 3. Set Terms per case 4. Click “Create Account”
Expected Result	• Unchecked: blocked + feedback; no redirect • Checked: success + redirect

A.3.1.9 TC-REG-009 — Required Fields Reject Whitespace-Only Input (Trim Validation)

Item	**Details**
Objective	Verify required fields treat whitespace-only as empty and block submission.
Execution Viewports	Desktop
Preconditions	Application accessible.
Test Data	15 spaces (" ")
Steps	1. Open /register.html 2. Fill all required fields with baseline valid data 3. Replace ONE target field value with 15 spaces 4. Ensure Terms are checked 5. Click “Create Account”
Expected Result	• Registration blocked; no redirect to /index.html?registered=true • Clear validation feedback shown to user
Execution Notes (known product behavior)	• Field-level errors are reliably implemented only for Email/Phone/ZIP/Password/Confirm Password. • For First Name/Last Name/City/Address: missing validation/feedback is covered by BUG-008 (validation missing) and BUG-026 (submit-level failure message missing).
Automation Note (negative-case guard)	• When submit failure has no global failure message, automation uses a redirect guard: after Submit, wait ~2000ms and assert that success redirect did NOT occur.

A.3.2 Login

A.3.2.1 TC-LOG-001 — Login with Valid Credentials (Happy Path)

Item	**Details**
Objective	Verify registered user logs in and is redirected to Dashboard.
Execution Viewports	Desktop
Preconditions	User exists (created via TC-REG-001).
Test Data	A.2.2
Steps	1. Open /index.html 2. Enter valid Email + Password 3. Click “Login”
Expected Result	• Success feedback displayed • Redirect to /dashboard.html

A.3.2.2 TC-LOG-002 — Login Rejects Invalid Credentials with Accurate Feedback

Item	Details
Objective	Verify login fails with incorrect credentials and does not show false success.
Execution Viewports	Desktop
Preconditions	User exists (created via TC-REG-001).

Test Data:

Case	Email	Password	Expected
1	<registered email>	WrongPassword123	Blocked + accurate feedback
2	non.existing<currentTime>@example.com	AnyPassword123	Blocked + “user not found” / equivalent

Item	Details
Steps	1. Open /index.html 2. Enter credentials per case 3. Click “Login”
Expected Result	• Blocked cases: remain on Login; feedback accurate • Allowed case: redirect to Dashboard

A.3.2.3 TC-LOG-003 — Remember Me Functionality (Checked vs Unchecked)

Item	Details
Objective	Verify Remember Me changes persistence after browser close/reopen.
Execution Viewports	Desktop
Preconditions	User exists (created via TC-REG-001).
Test Data	A.2.2
Steps	• Case A (Checked): 1. Open /index.html 2. Check “Remember Me” 3. Login successfully and confirm Dashboard 4. Close browser 5. Reopen and navigate to /index.html (or site root) • Case B (Unchecked): 1. Open /index.html 2. Ensure “Remember Me” unchecked 3. Login successfully and confirm Dashboard 4. Close browser 5. Reopen and navigate to /index.html (or site root)
Expected Result	• Case A: session persists across browser restart; user lands on Dashboard • Case B: session does not persist after browser restart; user lands on Login

A.3.3 Session

A.3.3.1 TC-SEC-001 — Logout Invalidates Session and Restricts Dashboard Access

Item	Details
Objective	Verify logout clears authentication state and prevents /dashboard.html direct access.
Execution Viewports	Desktop
Preconditions	User is logged in and currently on Dashboard (auth state present in storage).
Steps	1. Click "Logout" 2. Confirm redirect to /index.html 3. Open /dashboard.html directly
Expected Result	• Dashboard access is blocked; user is redirected to Login or an unauthorized state is shown (no previous user state is restored).

A.3.3.2 TC-SEC-002 — New Login Overwrites Previous User Auth State (No Storage Mismatch)

Item	Details
Objective	Verify that after logout, logging in as a different user does not retain the previous user in storage and does not restore the old identity.
Execution Viewports	Desktop
Preconditions	User A exists and is logged in to Dashboard. User B exists.
Steps	1. Log in as User A and open Dashboard 2. Click "Logout" 3. Log in as User B 4. Refresh the page or open /dashboard.html directly
Expected Result	• User B remains the active session after refresh/direct access • Previous User A is not restored from localStorage/sessionStorage • No "mixed" or inconsistent identity appears across storages

A.3.4 Forgot password

A.3.4.1 TC-FP-001 — Forgot Password Reset Request (Happy Path Expectation)

Item	Details
Objective	Verify reset requires valid user verification (Security Q/A) if UI presents it.
Execution Viewports	Desktop
Preconditions	Registered user exists AND Security Q/A stored (end-to-end expectation).
Test Data	Registered email + expected Security Q/A (A.2.3).
Steps	1. Open /forgot-password.html 2. Enter registered Email 3. Select expected Security Question 4. Enter correct Security Answer 5. Click “Send Reset Link”
Expected Result	• Request accepted only when verification satisfied • Confirmation feedback displayed
Note	Current app does not collect Security Q/A during registration (BUG-013).

A.3.4.2 TC-FP-002 — Forgot Password: Non-Existing Email Handling

Item	Details
Objective	Verify reset is not confirmed for emails that do not exist.
Execution Viewports	Desktop

Test Data:

Case	Email	Expected
1	non.existing<currentTime>@example.com	Blocked + error
2	<registered email>	Proceeds (subject to verification)

Item	Details
Steps	1. Open /forgot-password.html 2. Enter Email per case 3. Click “Send Reset Link”
Expected Result	• Non-existing: error shown; no success confirmation • Existing: proceeds (subject to verification)

A.3.4.3 TC-FP-003 — Forgot Password: Security Question/Answer Validation

Item	Details
Objective	Verify reset blocks incorrect/missing Q/A and accepts correct.
Execution Viewports	Desktop
Preconditions	Registered user exists AND Security Q/A stored (end-to-end expectation).

Test Data:

Case	Security Question	Security Answer	Expected
1	Correct	Wrong answer	Blocked
2	Wrong question	Correct answer	Blocked
3	Not selected	Empty	Blocked
4	Correct	Empty	Blocked
5	Not selected	Some answer	Blocked
6	Correct	Correct	Allowed

Item	Details
Steps	1. Open /forgot-password.html 2. Enter registered Email 3. Set Q/A per case 4. Click "Send Reset Link"
Expected Result	• Incorrect/missing: blocked + feedback • Correct: allowed + confirmation
Note	Current app does not collect Security Q/A during registration (BUG-013).

A.3.5 Dashboard

A.3.5.1 TC-DASH-001 — Update Profile Updates User Data and Recent Activity

Item	Details
Objective	Verify Update Profile performs real change and updates activity timestamp.
Execution Viewports	Desktop
Preconditions	User logged in and on Dashboard.
Steps	1. Click "Update Profile" 2. Verify edit UI exists and allows changes 3. Save changes 4. Verify "Profile updated" timestamp changes
Expected Result	• Real update flow exists and persists state • Recent Activity reflects latest update

A.3.5.2 TC-DASH-002 — Dashboard Actions Perform Real Operations

Item	Details
Objective	Verify Settings/Contact Support/Download Report perform real actions or are clearly “Not implemented”.
Execution Viewports	Desktop
Preconditions	User logged in and on Dashboard.
Steps	1. Click “Settings” 2. Click “Contact Support” 3. Click “Download Report”
Expected Result	• Real operation occurs (navigate/save/ticket/download) or • Clear “Not implemented” messaging; no misleading success toast

A.3.5.3 TC-DASH-003 — “Last Login” Shows Previous Login Time

Item	Details
Objective	Verify “Last login” shows previous session login time, not current.
Execution Viewports	Desktop
Preconditions	Ability to login twice in separate sessions.
Steps	1. Log in and open Dashboard; record “Last login” 2. Log out 3. Wait 2 seconds 4. Log in again and open Dashboard 5. Check “Last login”
Expected Result	• “Last login” equals the previous session login time (Step 1), not current session time

A.3.6 UI / UX (cross-viewport)

A.3.6.1 TC-UX-001 — Login Page Usability (Cross-Viewport)

Item	Details
Execution Viewports	Desktop / Tablet / Mobile
Steps	1. Open /index.html 2. Verify: Email, Password, Login button, Remember Me checkbox, nav links visible and interactable
Expected Result	Controls visible/readable/interactable across viewports

A.3.6.2 TC-UX-002 — Registration Page Usability (Cross-Viewport)

Item	Details
Execution Viewports	Desktop / Tablet / Mobile
Steps	1. Open /register.html 2. Verify: all required inputs + Terms + Create Account visible and interactable
Expected Result	Controls usable across viewports; no overlays hiding required inputs

A.3.6.3 TC-UX-003 — Forgot Password Page Usability (Cross-Viewport)

Item	Details
Execution Viewports	Desktop / Tablet / Mobile
Steps	1. Open /forgot-password.html 2. Verify: Email + Security Question + Security Answer + Submit visible and interactable
Expected Result	Controls visible/readable/interactable across viewports

A.3.6.4 TC-UX-004 — Dashboard Usability (Cross-Viewport)

Item	Details
Execution Viewports	Desktop / Tablet / Mobile
Steps	1. Open Dashboard after login 2. Verify readability of: cards, last login, quick actions, recent activity 3. Verify quick action buttons accessible
Expected Result	Dashboard content readable; key actions accessible; overlays do not obstruct content

A.3.6.5 TC-UX-005 — Validation Feedback Visibility (Cross-Page, Cross-Viewport)

Item	Details
Execution Viewports	Desktop / Tablet / Mobile

Test Data:

Case	Page	Field	Invalid Input
1	Register	Email	test@com
2	Register	Phone	abcXYZ
3	Register	ZIP	12A
4	Register	Password	Pass12
5	Login	Email	test
6	Forgot Password	Email	test

Item	Details
Steps	1. Set viewport 2. Trigger validation error per case 3. Attempt submit if applicable
Expected Result	• Validation feedback visible/readable and guides correction across all viewports

A.3.6.6 TC-UX-006 — Submit-Level Failure Feedback Is Shown Consistently (Cross-Form)

Item	**Details**
Objective	Verify each form shows a clear submit-level error message when submission fails (not “silent”).
Execution Viewports	Desktop / Tablet / Mobile
Steps	**1. Registration:** trigger a blocked submit and observe #registerMessage **2. Login:** trigger a blocked submit and observe #loginMessage **3. Forgot Password:** trigger a blocked submit and observe #forgotPasswordMessage
Expected Result	• Submit failures show a clear global error message consistently, in addition to any field-level messages.

APPENDIX B — DEFECT REPORT

B.1 Test Environment

Item	Details
OS	Windows 11 Pro
Browsers	Google Chrome (latest)
Base URL	https://qa-test-web-app.vercel.app

B.2 Defect Summary

Bug ID	Category	Title	Severity	Priority
BUG-001	Authentication/ Session	Authentication state persists and becomes inconsistent across storages (logout not fully clearing auth + storage rehydration)	Critical	P1
BUG-002	Registration – Email	Weak email validation accepts invalid formats	High	P1/P2
BUG-003	Registration – Password	Password policy is weak (accepts too-short passwords)	High	P1/P2
BUG-004	Registration – Password	Confirm Password mismatch is not validated	High	P1
BUG-005	Registration – Phone	Phone field accepts invalid characters (letters/symbols/whitespace)	Medium/High	P2
BUG-006	Registration – ZIP	ZIP Code field accepts invalid characters (letters/whitespace)	Medium/High	P2
BUG-007	Registration – Required Fields	Whitespace-only values accepted in required registration fields (missing trim/sanitization)	High	P1/P2
BUG-008	Registration – Data Quality	Text fields lack basic validation (invalid characters / no sensible minimum length)	Medium	P2/P3
BUG-009	Registration – Compliance	Terms and Conditions acceptance is not enforced	High	P1/P2
BUG-010	Login – Feature Wiring	“Remember Me” checkbox has no functional effect	Low/Medium	P3
BUG-011	Registration – Feature Wiring	Newsletter checkbox has no observable effect	Low	P3

Bug ID	Category	Title	Severity	Priority
BUG-012	Forgot Password – User Validation	Forgot Password shows success for non-existing users (no email existence check)	Medium/High	P2
BUG-013	Forgot Password – Design Consistency	Security Q/A requested in reset flow but never collected during registration	Medium	P2/P3
BUG-014	Mobile UX / Validation	Validation error messages are hidden on mobile across the app	High	P1/P2
BUG-015	Dashboard – Functionality	Update Profile shows success but no update exists; activity timestamp not updated	High	P1/P2
BUG-016	Dashboard – Functionality	Dashboard action buttons show success toasts without real actions	Medium	P2/P3
BUG-017	Dashboard – Data Accuracy	“Last login” shows current login time instead of previous login	Medium	P2/P3
BUG-018	UI/UX – Responsive	Street Address visually hidden by overlay on tablet (input still possible)	Medium	P2/P3
BUG-019	UI/UX – Responsive	“Remember Me” hidden / not interactable on mobile	Low	P3
BUG-020	UI/UX – Responsive	Newsletter checkbox visually obstructed on mobile	Low	P3
BUG-021	UI/UX – Responsive	Rewards Points card visually obstructed on mobile	Low	P3
BUG-022	UI/UX – Responsive	“Download Report” looks disabled on mobile but is clickable	Low/Medium	P3
BUG-023	UI/UX – Responsive	Account Status card content hidden by overlay on tablet	Low/Medium	P3
BUG-024	UI/UX – Responsive	“Email verified” activity item visually obstructed on tablet	Low	P3
BUG-025	UI/UX – Responsive	Security Question dropdown hidden on mobile viewport	Low/Medium	P3
BUG-026	UX / Feedback	Submit failure has no consistent submit-level feedback across forms	Medium	P2/P3

Execution note: Unless explicitly stated otherwise, test cases are executed on Desktop (Chrome, 1920×1080). UI/UX test cases are executed on Desktop/Tablet/Mobile.

B.3 Detailed Bug Reports

B.3.1 BUG-001 — Authentication State Persists and Becomes Inconsistent Across Storages (Logout Not Fully Clearing Auth + Storage Rehydration)

URL: /dashboard.html
Preconditions: User has logged in at least once (auth state may exist in sessionStorage and/or localStorage).
Steps to Reproduce (Variant A — Post-logout direct access):

1. Log in with a valid account (User A).
2. Click **Logout**.
3. Open **/dashboard.html** directly in the address bar.

Expected Result: User is logged out; Dashboard access is blocked and user is redirected to Login (/index.html) or shown an unauthorized state.
Actual Result: Dashboard can remain accessible because authentication state persists (e.g., in localStorage) and is reused on direct navigation.
Steps to Reproduce (Variant B — Storage rehydration / identity mismatch):

1. Log in with a valid account (User A) and open Dashboard.
2. Click **Logout**.
3. Log in with a different account (User B).
4. Inspect storage values (DevTools → Application → Storage): **sessionStorage** and **localStorage**.
5. Open **/dashboard.html** directly (optional confirmation step).

Observed Behavior:

- Dashboard reads **currentUser** from **sessionStorage** and falls back to **localStorage**.
- If a value exists in one storage, the page may sync/rehydrate it into the other, effectively restoring an “authenticated” state.
- **Clearing only one storage** (localStorage or sessionStorage) can be insufficient because the remaining storage repopulates the other.
- When switching users, **sessionStorage may reflect User B** while **localStorage still contains User A** (stale identity), creating inconsistent auth state.
- In an Incognito session (no prior stored state), direct access to **/dashboard.html** correctly redirects to Login.

Expected Result (system-level):

- Logout must fully invalidate authentication state across all storages.
- Dashboard must not rehydrate stale auth data and must not “self-repair” into an authenticated state from leftover storage.
- When a new user logs in, storage must be consistent (no stale user identity remains).

Actual Result (system-level):
Auth state persists and can be rehydrated from storage, enabling unauthorized Dashboard access and stale identity conflicts.
Severity: Critical
Priority: P1
Note: Root cause appears to be incomplete logout cleanup and inconsistent "source of truth" for auth state across sessionStorage/localStorage.

B.3.2 BUG-002 — Weak Email Validation Accepts Invalid Formats

URL: /register.html
Steps: Enter invalid emails (e.g., test@com, a@@b, "hello a@b world") and submit with otherwise passing values.
Expected Result: Reject invalid emails with clear feedback.
Actual Result: Invalid emails are accepted.
Severity: High | **Priority:** P1/P2

B.3.3 BUG-003 — Password Policy is Weak (Accepts Too-Short Passwords)

URL: /register.html
Steps: Enter a 4-character password (e.g., 1234) and submit with otherwise valid data.
Expected Result: Enforce minimum 8 characters (and target complexity).
Actual Result: Password length 4 accepted.
Severity: High | **Priority:** P1/P2

B.3.4 BUG-004 — Confirm Password Mismatch is not Validated

URL: /register.html
Steps: Password=Password123, Confirm=Different123, submit.
Expected Result: Block registration; show "Passwords do not match".
Actual Result: Registration proceeds.
Severity: High | **Priority:** P1

B.3.5 BUG-005 — Phone Field Accepts Invalid Characters

URL: /register.html
Steps: Enter phone with letters/symbols/whitespace-only and submit.
Expected Result: Reject letters/invalid symbols; reject whitespace-only.
Actual Result: Invalid characters accepted.
Severity: Medium/High | **Priority:** P2

B.3.6 BUG-006 — ZIP Code Field Accepts Invalid Characters

URL: /register.html
Steps: Enter ZIP with letters/whitespace-only and submit.
Expected Result: Digits-only; reject whitespace-only.
Actual Result: Letters/whitespace accepted if min length met.
Severity: Medium/High | **Priority:** P2

B.3.7 BUG-007 — Whitespace-Only Values Accepted in Required Registration Fields

URL: /register.html
Steps: Enter spaces only in required fields (meeting minimum lengths), use email that passes current validation, submit.
Expected Result: Treat whitespace-only as empty; block registration with feedback.
Actual Result: Registration can proceed with whitespace-only values.
Severity: High | **Priority:** P1/P2

B.3.8 BUG-008 — Text Fields Lack Basic Validation (Data Quality)

URL: /register.html
Affected Fields: First Name, Last Name, City, Address
Steps: Enter invalid values (symbol-only, too-short after trim, digits-only, etc.) and submit.
Expected Result: Enforce basic data-quality rules after trimming; show field-level feedback.
Actual Result: Invalid values accepted; validation/feedback missing.
Severity: Medium | **Priority:** P2/P3

B.3.9 BUG-009 — Terms and Conditions Acceptance is not Enforced

URL: /register.html
Steps: Fill valid data, leave Terms unchecked, submit.
Expected Result: Block submission; require Terms acceptance.
Actual Result: Registration proceeds.
Severity: High | **Priority:** P1/P2

B.3.10 BUG-010 — "Remember Me" Checkbox Has No Functional Effect

URL: /index.html
Steps: Compare session persistence checked vs unchecked after browser close/reopen.
Expected Result: Observable persistence difference or clear not-implemented behavior.
Actual Result: No observable difference.
Severity: Low/Medium | **Priority:** P3

B.3.11 BUG-011 — Newsletter Checkbox Has No Observable Effect

URL: /register.html
Steps: Subscribe to newsletter, register, check for persisted preference/acknowledgement.
Expected Result: Preference stored/acknowledged.
Actual Result: No observable effect.
Severity: Low | **Priority:** P3

B.3.12 BUG-012 — Forgot Password Shows Success for Non-Existing Users

URL: /forgot-password.html
Steps: Enter non-registered email and click "Send Reset Link".
Expected Result: Show "User not found" (or equivalent); no success confirmation.
Actual Result: Success message shown.
Severity: Medium/High | **Priority:** P2

B.3.13 BUG-013 — Security Q/A Requested in Reset Flow but Never Collected During Registration

URLs: /register.html, /forgot-password.html
Steps: Register new account (no Security Q/A). Open Forgot Password (Security Q/A visible).
Expected Result: Collect Security Q/A during registration OR remove it from reset flow.
Actual Result: Reset requests information user never provided.
Severity: Medium | **Priority:** P2/P3

B.3.14 BUG-014 — Validation Error Messages Hidden on Mobile Across the App

URLs: /register.html, /index.html, /forgot-password.html
Preconditions: Mobile viewport (≤767px).
Steps: Trigger validation errors and observe error areas under fields.
Expected Result: Errors visible and readable on mobile.
Actual Result: Errors hidden; user receives no correction guidance.
Severity: High | **Priority:** P1/P2

B.3.15 BUG-015 — Update Profile Shows Success but No Update Functionality Exists

URL: /dashboard.html
Steps: Click "Update Profile"; observe no edit UI; activity time not updated.
Expected Result: Real profile update flow and timestamp updates.
Actual Result: Success toast only; no real update; UI unchanged.
Severity: High | **Priority:** P1/P2

B.3.16 BUG-016 — Dashboard Action Buttons Show Success Toasts Without Real Actions

URL: /dashboard.html
Steps: Click Settings/Contact Support/Download Report.
Expected Result: Real operations or clear "Not implemented" messaging.
Actual Result: Success toasts only; no real operations.
Severity: Medium | **Priority:** P2/P3

B.3.17 BUG-017 — "Last Login" Displays Current Login Time Instead of Previous Login

URL: /dashboard.html
Steps: Login twice in separate sessions; compare "Last login".
Expected Result: Shows previous session login time.
Actual Result: Shows current login time; misleading.
Severity: Medium | **Priority:** P2/P3

B.3.18 BUG-018 — Street Address Field Visually Hidden by Overlay on Tablet

URL: /register.html
Preconditions: Tablet viewport (768–1024px).
Steps: Open Register; observe Address covered by overlay; attempt to input.
Expected Result: Field visible and usable.
Actual Result: Field visually obscured; may still accept input.
Severity: Medium | **Priority:** P2/P3

B.3.19 BUG-019 — "Remember Me" Section Hidden / Not Interactable on Mobile

URL: /index.html
Preconditions: Mobile viewport (≤767px).
Steps: Open Login and try to interact with Remember Me.
Expected Result: Visible and interactable.
Actual Result: Hidden/obstructed or not interactable.
Severity: Low | **Priority:** P3

B.3.20 BUG-020 — Newsletter Checkbox Visually Obstructed on Mobile

URL: /register.html
Preconditions: Mobile viewport (≤767px).
Steps: Open Register; observe newsletter checkbox area; attempt interaction.
Expected Result: Control clearly visible and consistent with state.

Actual Result: Obstructed/overlaid; may still be clickable.
Severity: Low | **Priority:** P3

B.3.21 BUG-021 — Rewards Points Card Visually Obstructed on Mobile

URL: /dashboard.html
Preconditions: Mobile viewport (≤767px).
Steps: Open Dashboard; observe Rewards Points card.
Expected Result: Card readable.
Actual Result: Content obstructed by overlay.
Severity: Low | **Priority:** P3

B.3.22 BUG-022 — “Download Report” Looks Disabled on Mobile but Remains Clickable

URL: /dashboard.html
Preconditions: Mobile viewport (≤767px).
Steps: Observe disabled-like appearance; click button.
Expected Result: If looks disabled it should not execute; otherwise, it should not look disabled.
Actual Result: Clickable and shows “Report downloaded!” toast.
Severity: Low/Medium | **Priority:** P3

B.3.23 BUG-023 — Account Status Card Content Hidden By “Premium Feature” Overlay on Tablet

URL: /dashboard.html
Preconditions: Tablet viewport (768–1024px).
Steps: Open Dashboard; observe Account Status card.
Expected Result: Content readable.
Actual Result: Overlay hides content.
Severity: Low/Medium | **Priority:** P3

B.3.24 Bug-024 — “Email Verified” Activity Item Visually Obstructed on Tablet

URL: /dashboard.html
Preconditions: Tablet viewport (768–1024px).
Steps: Open Dashboard; observe “Email verified” item.
Expected Result: Readable.
Actual Result: Obstructed.
Severity: Low | **Priority:** P3

B.3.25 Bug-025 — Security Question Dropdown Hidden on Mobile Viewport

URL: /forgot-password.html
Preconditions: Mobile viewport (≤767px).
Steps: Open Forgot Password; locate Security Question section.
Expected Result: Visible and usable.
Actual Result: Hidden/obstructed.
Severity: Low/Medium | **Priority:** P3

B.3.26 Bug-026 — Submit Failure Has No Consistent Submit-Level Feedback Across Forms

URLs: /register.html, /index.html, /forgot-password.html
Preconditions: Any viewport; reproduced on Desktop and Mobile.
Steps to Reproduce:

1. Registration: Open /register.html → enter blocked input (e.g., invalid email "test") → click "Create Account" → observe submit-level message area (e.g., #registerMessage).
2. Login: Open /index.html → enter invalid email format (e.g., "test") or invalid credentials → click "Login" → observe submit-level message area (e.g., #loginMessage).
3. Forgot Password: Open /forgot-password.html → enter invalid email format (e.g., "test") or non-existing email → click "Send Reset Link" → observe submit-level message area (e.g., #forgotPasswordMessage).

Expected Result: On submission failure, a clear submit-level error message is shown consistently (e.g., "Please fix the errors above"), in addition to field-level validation.
Actual Result: No submit-level failure message is shown; forms can appear "silent" even though submission fails.
Impact/Notes: Inconsistent UX (success flows show clear global success message and/or redirect). Automation uses a redirect-guard workaround in negative cases (wait and assert no success redirect).
Severity: Medium | **Priority:** P2/P3

APPENDIX C — AUTOMATION EXECUTION REPORT

C.1 Overview

This appendix documents the automated test execution for the QA Test Web Application using Playwright (JavaScript). The automation focuses on core functional flows (Registration/Login/Forgot Password), session behavior, and cross-viewport UX checks.

C.2 Scope of Automation Execution

In scope (automated):

- Registration: happy path and selected negative validation scenarios (TC-REG-001–TC-REG-006, TC-REG-008–TC-REG-009). TC-REG-007 is excluded.
- Login: happy path, negative credentials scenarios, Remember Me behavior
- Forgot Password: validation coverage and non-existing email behavior (subject to current product limitations)
- Session: logout + direct dashboard access restrictions
- UX (cross-viewport): validation feedback visibility on Desktop/Tablet/Mobile

Out of scope (automated execution):

- Real email delivery validation (reset/newsletter)
- Backend persistence verification beyond observable UI behavior
- Performance/load testing

C.3 Test Environment (Automation)

Item Details

Operating System Windows 11 Pro

Node.js v24.13.0

npm 11.6.2

Playwright 1.58.1

Browser engine Chromium (Playwright)

Base URL: https://qa-test-web-app.vercel.app

Reporter Playwright HTML Reporter

Execution workers=1 (reduce flakiness due to shared state/race conditions during user creation and immediate navigation)

C.4 Automation Suite Design Notes

- Page Object Model (POM) is used to keep selectors and page actions centralized and maintainable.
- Test data is generated dynamically (unique email per run) to avoid collisions.

- Data-driven patterns are used for validation-heavy scenarios (e.g., multiple invalid emails/passwords/phones/ZIPs).
- The Playwright configuration defines three projects:
  - chromium (Desktop Chrome): functional/session/dashboard/registration/login/forgot-password tests + UX tests (desktop)
  - tablet (iPad gen 7): UX-only tests (tests/ux/)
  - mobile (Pixel 5): UX-only tests (tests/ux/)

C.5 Execution Commands

Install dependencies:

- npm install

Install Playwright browsers (if not installed):

- npx playwright install

Run full suite (headless):

- npm test

Run headed:

- npm run test:headed

Run Playwright UI mode:

- npm run test:ui

Run UX-only tests:

- npm run test:ux

Run smoke tests (core happy paths):

- npm run test:smoke

Generate/open HTML report:

- npm run test:report

C.6 Execution Results (Snapshot)

Execution date: 11 Feb 2026

Result summary:

- Total tests executed: 51
- Passed: 21
- Failed: 30
- Flaky/retried: 0/0

Notes on failures:

Failures are expected due to known product defects documented in Appendix B (e.g., validation weaknesses, missing failure feedback, session/storage behavior). Failures that map to known defects are considered “known issues” rather than automation defects.

C.7 Known Product Limitations Impacting Automation

- BUG-026 (submit-level failure feedback missing): In negative cases where the UI provides no global failure message, automation uses a timed redirect guard (wait ~2000ms and assert that success redirect did NOT occur).
- BUG-001 (storage-driven auth persistence/rehydration): Tests involving authentication may clear sessionStorage/localStorage pre-run to prevent cross-test contamination and "zombie user" effects.
- The application appears to use temporary or inconsistent persistence; therefore tests generate unique users and avoid assumptions about long-term retention.
- TC-REG-007 is excluded: text-field data quality validation (First/Last/City/Address) is not implemented reliably in the product (BUG-008) and failures often have no submit-level feedback (BUG-026), reducing automation signal.

C.8 Report Artifact

Playwright HTML report is generated after execution.
Report location (generated locally after execution): playwright-report/index.html
Open locally: npx playwright show-report
CI report: GitHub Actions uploads "playwright-report" as a workflow artifact (retention: 14 days).
Repository: https://github.com/anamarija-tomic/qa-test-web-app
Note: playwright-report is not committed to Git; it is generated per run and shared via CI artifacts.

C.9 Automation Coverage Summary (Mapping to Manual Test Cases)

Area	Manual Test Cases	Automated Test Cases
Registration	TC-REG-001 – TC-REG-009	TC-REG-001 – TC-REG-006, TC-REG-008 – TC-REG-009
Login	TC-LOG-001 – TC-LOG-003	TC-LOG-001 – TC-LOG-003
Session	TC-SEC-001, TC-SEC-002	TC-SEC-001, TC-SEC-002
Forgot Password	TC-FP-001 – TC-FP-001	TC-FP-002
Dashboard	TC-DASH-001 – TC-DASH-003	TC-DASH-003
UX (Cross-Viewport)	TC-UX-001 – TC-UX-006	TC-UX-005