

Rento QA Audit Report (Nov 10 2025)

Feature	Role	PASS/ FAIL	Steps to Reproduce	URL / Route	Scre
Auth & Session – sign-in/out, protected routes, session persistence	Tenant / Landlord	PASS (with caveats)	1. Navigate to <code>/auth/sign-in</code> and submit valid email/password. 2. After a successful login you are redirected to <code>/dashboard</code> . 3. Open another tab and navigate to a protected route (e.g., <code>/dashboard</code>); session is persisted via Supabase session cookies. 4. Click your avatar and choose Sign out (calls <code>supabase.auth.signOut()</code> and then redirects to <code>/</code>).	<code>/auth/sign-in</code> and <code>/auth/sign-up</code> pages	
Dark mode / Theming – colour contrast and theme toggle	All	PARTIAL PASS	1. Log in and click the sun/moon icon in the header to toggle themes. 2. Inspect text contrast in both light and dark themes.	Global layout (<code>components/header.tsx</code> , <code>ThemeProvider</code>)	

Feature	Role	PASS/ FAIL	Steps to Reproduce	URL / Route	Score
Listings – create/edit/publish	Landlord	PASS	1. Log in as landlord and navigate to <code>/listings/new</code> . 2. Fill out the form (title, address, price, description, photos). 3. Attempt to save with missing fields to verify required-field errors. 4. Save as draft or publish; confirm the listing appears on the listing page and can be edited.	<code>/listings/new</code> and <code>/listings/[slug]/edit</code>	
Browse & Filters – search by location, price, beds/baths; filter persistence; no layout shift	Tenant	PASS	1. Navigate to <code>/browse</code> . 2. Use the filter drawer (city, min/max rent, beds, baths, type, pet-friendly, furnished, verified). 3. Apply filters; note the URL query parameters update (e.g., <code>?city=Toronto&min=1000&max=2500&beds=2</code>). 4. Reload or use the browser back button; filters remain applied and results persist. 5. Switch between grid and map views; ensure the results list does not shift unexpectedly.	<code>/browse</code>	

Feature	Role	PASS/ FAIL	Steps to Reproduce	URL / Route	Screen
Favorites – favourite/un-favourite properties with persistence	Tenant	PARTIAL PASS	1. While browsing listings, click the heart icon on a property card to favourite it. 2. Refresh the page; the favourite state should persist. 3. Open the property detail page; click heart to toggle. 4. Navigate to <code>/dashboard/favorites</code> to view all saved homes.	<code>/api/favorites</code> (API route), <code>/dashboard/favorites</code>	
Applications – tenant applies, landlord updates status	Tenant ↔ Landlord	FAIL (critical)	1. As a tenant, open a listing and submit an application (message, deposit, etc.). 2. Log in as the landlord and navigate to <code>/applications</code> . The new application appears under Submitted . 3. Attempt to update status to Reviewed/Accepted/Rejected. 4. Tenant checks <code>/applications</code> → recent applications to see status.	<code>/applications</code> page, <code>/api/applications/update</code>	

Feature	Role	PASS/ FAIL	Steps to Reproduce	URL / Route	Screen
Tours & Scheduling – request, confirm/reschedule, prevent double-booking	Tenant ↔ Landlord	FAIL (critical)	1. As a tenant, click Request Tour on a property and pick a date/time. 2. As landlord, open <code>/tours</code> → scheduled tours and try to confirm/reschedule the tour. 3. Attempt to schedule another tour at the same date/time to test double-booking prevention.	<code>/tours</code> page, <code>app/(app)/tours/actions.ts</code>	
Messaging / Chat – real-time threads and messages	Tenant ↔ Landlord	PARTIAL FAIL	1. Send a message as tenant from a property page; a thread is created or reused via <code>/api/messages</code> . 2. Landlord replies in <code>/messages</code> ; the conversation shows new messages with timestamps. 3. Check that pressing Enter sends and Shift+Enter inserts a newline. 4. Reload; messages should persist, and unread counts reset.	<code>/messages</code> page; <code>/api/messages</code>	

Feature	Role	PASS/ FAIL	Steps to Reproduce	URL / Route	Screen
Dashboards – home pages for each role	Tenant / Landlord	PARTIAL FAIL	1. Log in as tenant and open <code>/dashboard</code> . 2. The page should show counts for favourites, applications, upcoming tours and recent messages. 3. Log in as landlord; the dashboard should show stats for properties, pending applications and tours.	<code>/dashboard</code>	
Accessibility – labels, aria, focus management, contrast, keyboard nav	All	PARTIAL PASS	Evaluate random pages (Browse, Sign-in, Listing form) for: 1. Presence of labels and <code>aria-describedby</code> on form fields; 2. Focus styles; 3. Screen reader semantics; 4. Keyboard navigation.	Global	
Performance – LCP, lazy loading, CLS	All	PARTIAL PASS	1. Measure Listing Details page on throttled 4G. 2. Observe largest contentful paint (LCP) and layout shift.	<code>/property/[slug]</code>	

Feature	Role	PASS/ FAIL	Steps to Reproduce	URL / Route	Score
Connection & Data Integrity – Supabase env, RLS, API calls	All	FAIL (critical)	Inspect code and migrations to verify correct service-key usage, RLS coverage and API security.	Repo analysis	
Environment / Secrets – .env handling and secret leakage	All	PASS	Check repository for .env.local and keys.	Repo	

Prioritized Defects and Suggested Fixes

Severity	Issue	Evidence / Explanation	Suggested Fix
Critical	RLS update policies missing for applications and tours – landlords cannot update statuses; all updates fail.	The migration only defines select/insert policies for these tables ¹⁰ . The UI calls <code>supabase.from('applications').update(...)</code> and <code>.update()</code> for tours ¹¹ , which is denied under row-level security.	Add <code>UPDATE</code> policies to allow landlords to update status for their own properties and tenants to view updates. Use <code>ALTER POLICY</code> or <code>CREATE POLICY</code> with appropriate <code>WITH CHECK</code> conditions. Alternatively, implement server actions using the service role key.

Severity	Issue	Evidence / Explanation	Suggested Fix
Critical	Tours double-booking & timezone – requests ignore conflicts and timezone; reschedule fails due to RLS.	The tours action inserts without checking overlapping <code>scheduled_at</code> ; timezone is not persisted; no RLS update policy.	Add a unique index on <code>(landlord_id, scheduled_at)</code> or implement conflict checks in server action; store timezone; add <code>UPDATE</code> policy; surface meaningful error messages.
Critical	Applications status update errors hidden – UI uses Supabase client; errors are not surfaced.	<code>ApplicationsClient</code> calls <code>.update(...)</code> without catching errors, so users see no feedback when update fails ¹¹ .	Add try/catch around status update calls; show toast on failure; update via server action (API route) to centralize validation and error handling.
High	Messages bypass secure API route – client calls Supabase directly, skipping captcha/CSRF.	<code>sendMessage</code> function sends messages directly via Supabase ¹⁴ , ignoring the robust <code>/api/messages</code> route ¹³ .	Remove client Supabase calls; always route messages through <code>/api/messages</code> . Consolidate validation and presence updates there.
High	Dashboard unhandled errors – <code>Promise.all</code> lacks error handling.	In <code>dashboard/page.tsx</code> there is no <code>try/catch</code> , so any failing data fetch crashes the entire page ¹⁵ .	Wrap data fetching in <code>try/catch</code> ; show fallback UI for each section; continue rendering other widgets.
High	Insufficient accessibility on some forms and dark-mode contrast	Some inputs lack <code>aria-describedby</code> (e.g., applications form), and dark-mode text contrast falls below WCAG.	Audit forms with an accessibility checker (axe); add hidden hints linked via <code>aria-describedby</code> for errors/help; increase contrast for secondary text and ghost buttons in dark mode. Provide a skip-to-content link.

Severity	Issue	Evidence / Explanation	Suggested Fix
Medium	Performance: large hero images & missing WebP/AVIF	Listing details page loads full-resolution images without <code>priority</code> or WebP conversions.	Use <code>next/image</code> with <code>priority={true}</code> for the hero image and specify <code>sizes</code> . Convert uploaded images to modern formats (WebP/AVIF) on upload or use Supabase storage transformation.
Medium	Rate limiting gaps – server actions bypass API rate limits.	Only some API routes implement rate limiting. Actions like tours scheduling and applications update call Supabase directly.	Centralize writes through API routes that enforce rate limits; implement per-user limits in Supabase functions for tours and applications.
Low	No debounce for city search – typing in the city field triggers queries on each keystroke.	<code>FiltersSheet</code> sets city search value instantly; <code>getMany</code> runs on each change.	Debounce city input (e.g., 300 ms) before applying filters; update query string only when user clicks Apply .
Low	Missing attachments in chat – file uploads not implemented.	<code>docs/CHAT_ATTACHMENTS_TODO.md</code> acknowledges missing attachments.	Implement chat attachments per the TODO: file input component, <code>messages_attachments</code> table, supabase storage, UI to preview attachments.
Low	No HTML sanitisation in listings description	Landlords can input HTML markup; there is no sanitisation.	Sanitize description with a library (DOMPurify) before saving or rendering; restrict allowed tags.

1 2 [page.tsx](#)

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/app/auth/sign-in/page.tsx>

3 [page.tsx](#)

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/app/browse/page.tsx>

4 6 [BrowseClient.tsx](#)

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/app/browse/BrowseClient.tsx>

5 16 **FiltersSheet.tsx**

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/components/FiltersSheet.tsx>

7 **FavoriteButton.tsx**

<https://github.com/ShashidhaReddy3333/RentoH/blob/main/components/ui/FavoriteButton.tsx>

8 **route.ts**

<https://github.com/ShashidhaReddy3333/RentoH/blob/main/app/api/favorites/route.ts>

9 **applications.ts**

<https://github.com/ShashidhaReddy3333/RentoH/blob/main/lib/data-access/applications.ts>

10 **setup.sql**

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/supabase/setup.sql>

11 **ApplicationsClient.tsx**

[https://github.com/ShashidhaReddy3333/RentoH/blob/main/app/\(app\)/applications/ApplicationsClient.tsx](https://github.com/ShashidhaReddy3333/RentoH/blob/main/app/(app)/applications/ApplicationsClient.tsx)

12 **actions.ts**

[https://github.com/ShashidhaReddy3333/RentoH/blob/main/app/\(app\)/tours/actions.ts](https://github.com/ShashidhaReddy3333/RentoH/blob/main/app/(app)/tours/actions.ts)

13 **route.ts**

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/app/api/messages/route.ts>

14 **messages.ts**

<https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/lib/data-access/messages.ts>

15 **page.tsx**

[https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/app/\(app\)/dashboard/page.tsx](https://github.com/ShashidhaReddy3333/RentoH/blob/ea31503f3668a225cda5516ab0fde0f8ef13748f/app/(app)/dashboard/page.tsx)

17 **server.ts**

<https://github.com/ShashidhaReddy3333/RentoH/blob/main/lib/supabase/server.ts>

18 **env.ts**

<https://github.com/ShashidhaReddy3333/RentoH/blob/main/lib/env.ts>