

Code Review: Car Rental App

Project: car_rental

Branch: main

Date: November 18, 2025

1. The Big Picture

We looked at the car_rental website code. It's mostly JavaScript and CSS (the stuff that makes it look good).

The Gist: The code is "Okay" overall. It technically passed the automated check, but that shouldn't be the goal.

But, there are three big red flags:

- No Tests:** The computer gave the testing score a **0.0%**. This means there are zero automated safety checks running. If we change something, we won't know if we broke something else until a user finds it. (SQ doesn't detect our tests, so we must re-write them)
- Bugs:** There are **23 bugs** hiding in the code that need fixing, so the app doesn't crash.
- Security:** We found **8 "Security Hotspots"**. These aren't definitely hacks waiting to happen, but they are suspicious things that a human needs to double-check.

2. The Report Card

Category	Score	What it means
Will it break? (Reliability)	C	It has 23 bugs , so yeah, it might break.
Is it safe? (Security)	A	No confirmed holes, which is good.
Security Review	E	There are 8 suspicious spots nobody has looked at yet.

Is it messy? (Maintainability)	A	The code is actually pretty clean and easy to read.
Tests	Failed	0.0% . Our tests aren't detected by SQ

3. What Needs Fixing (The Details)

The Scary Stuff (Do this first)

There are a few things that looked important:

- Passwords in the Code:

The scanner thinks it found passwords written directly into the files (specifically in Login.test.js and Register.test.js). Even if these are just for testing, it's bad practice to write passwords in code files.

- The Search Bar Could Freeze:

There is a complicated text-search command (Regex) in Register.test.js that is written in a way that could accidentally freeze the whole browser if someone types something weird.

The Annoying Stuff (Bugs)

- **23 Bugs:** These are logical errors. For example, there are spots where the code logic is backwards ("Unexpected negated condition") or where we are ignoring errors instead of handling them.

The Messy Stuff (Code Smells)

There are **174 "Code Smells"**. These aren't bugs, just sloppy writing according to SQ:

- **Missing Checks:** We aren't checking if the data being passed around is the right type (missing "props validation").
- **Hard to Read:** Some text colors are too light and hard to read against the background.
- **Clicking things:** We made some things clickable (like div tags) that aren't actually buttons, which confuse screen readers.

- **Leftover Junk:** There is code that has been commented out (turned off) but left in the files. We should just delete it.

4. What is this app made of?

It's a standard React website.

- It has **Components** like a Footer, Navbar, and Charts.
- It has **Pages** for Admin stuff, Payments, Bookings, and a Home page.
- It looks like there *are* test files in the folders (like `Footer.test.js`), but they aren't running, which is why the score is 0%.

5. Next Steps (To-Do List)

Here is what I would do if I were you:

1. **Check those passwords:** Go look at the Security Hotspots and make sure we aren't saving real passwords in them.
2. **Fix the text search:** Fix that "Regex" issue in the register test file, so the app doesn't.
3. **Turn on the tests:** We have test files, but the system says **0.0%**. We need to figure out why they aren't counting.
4. **Clean up:** Delete the unused code and old imports to make things

6. Summarized by screenshots:

main

4.4k Lines of Code · Version 1.0 · Set as homepage

✓ Passed

Last analysis 2 minutes ago

New Code

Overall Code

Security

0 Open issues

A

Reliability

100 Open issues

C

Maintainability

168 Open issues

A

Accepted issues

0

B

Coverage

0.0%

D

Duplications

4.9%

E

On 854 lines to cover.

On 5.2k lines.

Security Hotspots

8

E

Coverage

Overview

Overall Code

Coverage

0.0%

Lines to Cover

854

Blocker

0

Uncovered Lines

854

High

4

Line Coverage

0.0%

Medium

91

Low

73

Info

0

Software Quality

Security

0

Reliability

100

Maintainability

168

Add to selection Ctrl + click

Severity

Blocker

0

High

4

Medium

91

Low

73

Info

0

Duplications	
Overview	
Overall Code	
Density	4.9%
Duplicated Lines	254
Duplicated Blocks	4
Duplicated Files	4