

# 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:**

1. **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)
2. **Bugs:** There are **23 bugs** hiding in the code that need fixing, so the app doesn't crash.
3. **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 <b>23 bugs</b> , so yeah, it might break.
Is it safe? (Security)	A	No confirmed holes, which is good.
Security Review	E	There are <b>8 suspicious spots</b> nobody has looked at yet.

<b>Is it messy? (Maintainability)</b>	<b>A</b>	The code is actually pretty clean and easy to read.
<b>Tests</b>	<b>Failed</b>	<b>0.0%.</b> 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:**

✓

Quality Gate ⓘ  
**Passed**

Last analysis 2 minutes ago

- New Code
- Overall Code

<b>Security</b> 0 Open issues <div>A</div>	<b>Reliability</b> 100 Open issues <div>C</div>	<b>Maintainability</b> 168 Open issues <div>A</div>
<b>Accepted issues</b> 0 <small>Valid issues that were not fixed</small> <div>🕒</div>	<b>Coverage</b> 0.0% <small>On 854 lines to cover.</small> <div>🔴</div>	<b>Duplications</b> 4.9% <small>On 5.2k lines.</small> <div>🔄</div>
<b>Security Hotspots</b> 8 <div>E</div>		

Coverage	Software Quality ⓘ
Overview	Security 0
Overall Code	Reliability 100
Coverage 0.0%	Maintainability 168
Lines to Cover 854	Add to selection <b>Ctrl + click</b>
Uncovered Lines 854	Severity ⓘ
Line Coverage 0.0%	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	