Jpad Consultancy

App Health Check

An independent review of an app built with AI tools

App reviewed	TradeQuote Portal (sample)
Prepared for	Sample Trading Ltd (a fictional client)
Built with	ChatGPT and Cursor
Date of review	14 October 2026
Prepared by	A Jpad engineer
Report version	1.0

Confidential. This report has been prepared for the client named above. You are free to share it with your own advisers and developers.

Summary

Overall: not ready for customers yet, but close.

TradeQuote Portal does what it was built to do and the design is sensible. However, two problems should be fixed before any customer uses it: customer records can be reached without logging in, and some quotes are calculated wrongly. Both are quick to fix. The other findings can be handled after launch.

Ratings at a glance

Area	Rating	In one line
Security and data protection	**Fix first**	Customer records can be opened without logging in, and a secret key is stored in the code.
Correct results	**Needs work**	7 of 50 real quotes came out wrong, 2 of them by more than £100.
Reliability	**Needs work**	Fine for a few users, slow with more, and it fails silently if payments are down.
Hosting, backups and cost	**Looks good**	Sensible set-up and low cost. A backup restore has not yet been tested.
Code quality	**Needs work**	The same pricing rules are copied into five places and there are no automated tests.
Ownership and licences	**Looks good**	The client owns everything and all components are fine to use commercially.

Fix these first

1. Make customer records require a login. Right now anyone with a quote link can see a customer's name, address and order history.
2. Replace the payment provider key, which is stored in plain text in the code, and treat the old one as exposed.
3. Stop the discount being applied twice on orders over £5,000, which overcharged customers by more than £100 in our tests.

What we reviewed

We reviewed the app the way a careful engineer would before taking responsibility for it. Our engineers read and judged every finding in this report.

What we looked at	How
The code	Read-only access to the repository, line by line in the sensitive areas.
The running app	A test copy, used as an ordinary customer and as an administrator.
Your real examples	50 past quotes supplied by the client, re-calculated and compared.
Hosting and backups	Screenshots of the hosting dashboard and backup settings.
Licences and ownership	The list of components the app depends on, and who owns each account.

What this review does not cover. It is not a full penetration test, a legal opinion, or a guarantee that the app is free of faults. It reflects the app as it was on the date of review and the materials supplied to us.

How we rate each area

Fix first	A problem that could harm customers or the business. Fix before the app is used by real people.
Needs work	Not dangerous today, but likely to cause trouble as the app grows. Plan to fix soon.
Looks good	In good shape. Nothing needed beyond routine care.

Findings

Each area is rated, explained in plain English, and followed by what to do and how long it takes.

Fix first	**Security and data protection**
What we found	• Anyone who has a customer's quote link can see that customer's name, address and order history without logging in. We confirmed this on three records. • The key for the payment provider is stored in the code in plain text. • Admin pages hide their buttons from ordinary users, but the server accepts the same requests from any logged-in user. • Good: passwords are stored using a strong, standard method.
Why it matters	Customer personal data could be seen by people who should not see it. That is a business risk and a data protection matter.
What to do	• Require login on the server for every customer record. • Move the payment key out of the code and replace it, treating the old one as exposed. • Check admin permissions on the server.
Effort	About 2 days.

Needs work	**Correct results**
What we found	We re-calculated 50 of your real past quotes. 43 matched exactly. • 5 differed by pennies because VAT was rounded on the total, where your accounts round each line. • 2 were wrong by more than £100, because the trade discount was applied twice on orders over £5,000.
Why it matters	Customers could be over- or undercharged, and the mistakes would not be obvious on screen.
What to do	• Round VAT per line. • Fix the double discount. • Add automated tests using these 50 real quotes so it cannot happen again.
Effort	About 1 day.

Needs work	**Reliability**
What we found	• Works well for a handful of users. With 200 quotes loaded, the quote list took 9 seconds to appear. • If the payment provider is unavailable the app shows a blank page. • Nobody is alerted when errors happen.
Why it matters	Slow pages and silent failures are usually noticed by customers first.
What to do	• Show a friendly message and retry when payments are down. • Speed up the quote list. • Set up error alerts.
Effort	About 1.5 days.

Looks good	**Hosting, backups and cost**
What we found	• Hosted with a reputable provider, with daily automatic backups switched on. • Current running cost is about £25 a month, and should stay low even at ten times the usage. • One note: nobody has tested restoring from a backup.
Why it matters	A backup that has never been restored is a hope, not a plan.
What to do	• Test one restore and write down the steps.
Effort	About half a day.

Needs work	**Code quality**
What we found	• The same pricing rules are copied into five places, so fixing one does not fix the others. This is the cause of the double-discount bug. • There are no automated tests. • Parts of the app were generated at different times and in different styles, so names and structure are inconsistent.
Why it matters	Another developer would find it slow and risky to change, and each change could break something unseen.
What to do	• Move pricing rules into one place. • Add automated tests. • Tidy names and structure.
Effort	About 2 days.

Looks good	**Ownership and licences**
What we found	• The code is in a repository owned by the client's company account. • All third-party components use licences that allow commercial use. • One note: the AI tool subscription sits on an employee's personal account.
Why it matters	If that person leaves, the history of how the app was built could be lost.
What to do	• Move the subscription to a company account.
Effort	Under 1 hour.

Fix list

In the order we would tackle them. Effort is working time for one of our engineers.

#	What to fix	Area	When	Effort
1	Require login on the server for every customer record	Security	**Before launch**	1 day
2	Replace the payment provider key and remove it from the code	Security	**Before launch**	0.5 day
3	Stop the discount being applied twice on orders over £5,000	Correct results	**Before launch**	0.5 day
4	Check admin permissions on the server, not just on screen	Security	**Before launch**	0.5 day
5	Round VAT per line, as your accounts do	Correct results	**Within a month**	0.5 day
6	Handle payment outages gracefully and add error alerts	Reliability	**Within a month**	1 day
7	Speed up the quote list	Reliability	**Within a month**	0.5 day
8	Test restoring a backup, and write down the steps	Hosting	**Within a month**	0.5 day
9	Move pricing rules into one place and add automated tests	Code quality	**Within 3 months**	2 days
10	Move the AI tool subscription to a company account	Ownership	**Within 3 months**	Under 1 hour

Total: about 7 days of work. Items 1 to 4 come to about 2.5 days.

Your options

Option	What is included	Fixed price	Timescale
A. Fix the essentials	Items 1 to 4: everything that should be done before customers use the app.	Quoted in writing	Agreed before we start
B. Fix everything	All 10 items on the list, so the app is dependable and easy for any developer to maintain.	Quoted in writing	Agreed before we start
C. Leave it as it is	No work from us. The security and calculation problems remain, so we would not advise putting it in front of customers.	None	None

Our recommendation: Option A now, so the app is safe to use, then Option B when convenient. Every option has a fixed price agreed in writing before we start. You are under no obligation to hire us, and this report is yours to give to any other developer.

What happens next

1. We arrange a call to walk you through the report and answer your questions.
2. You decide which option, if any, suits you.
3. If you would like us to do the work, we confirm the fixed price and timescale in writing before we start anything.

Contact

Your Jpad engineer will share their direct contact details when they send your report.