Sample Legacy Health Check report: a 2014 CakePHP order portal

This is an illustrative sample. Harlow Ridge Supply is not a real company, and nothing here comes from a client engagement. Any resemblance to a real business is coincidental. The findings show the kind of thing a Health Check turns up and how we report it. The prices are examples; yours depend on what we find.

LEGACY HEALTH CHECK

Harlow Ridge Supply

Legacy PHP application · Prepared by Skyhawk Labs

A regional industrial-supply distributor with about 60 staff. Its customers place and track orders through a web portal built in 2014 by a freelance developer who stopped responding in 2021.

APPLICATION

Customer order portal, built 2014

FRAMEWORK

CakePHP 2.6

PHP

5.6

DATABASE

MySQL 5.5, 4.1 GB

HOSTING

One virtual server, set up in 2015

SIZE

About 48,000 lines in 410 files

Where you stand

The portal works, and the business logic in it is sound. It does not need a rewrite. What it needs is the basic safety net that was never put in place: a backup that restores, a copy of the code somewhere other than the live server, and software that still receives security fixes.

Five things should be dealt with now. The most pressing is that there is no usable backup: if the server failed today, the most recent copy of your order data we could restore is 14 months old. The second is a deadline you did not set: your host retires this server plan on March 31.

Everything urgent can be fixed in one Stabilization Sprint while the portal keeps running. The larger framework upgrade can then be planned in steps instead of being forced by an emergency.

5	5	5
URGENT	TO PLAN FOR	CAN WAIT

What's urgent

These carry real risk today or have a deadline attached. Each can be fixed while the portal keeps running.

U1 There is no backup that can be restored — $900

What it means. If the server fails or is compromised, you lose every order, customer account and invoice entered in the last 14 months.

What we found. A nightly backup job exists, but it has been failing since the disk filled up. The last complete database backup is 14 months old. Backups are stored on the same server as the portal, so a server failure takes them with it. We tried to restore the most recent complete backup to a test machine and it failed on a corrupted table.

The fix. Free the disk space, repair the backup job, send an encrypted copy to separate storage every night, and add an alert when a backup fails. Then run a full restore and time it, so you know it works and how long it takes.

U2 PHP and the database stopped receiving security fixes at the end of 2018, and the host is retiring the server — $3,200

What it means. Newly discovered security holes in the software under your portal are never patched. Separately, your hosting company has given notice that this server plan ends on March 31.

What we found. The server runs PHP 5.6 and MySQL 5.5. Neither has had a security release since the end of 2018. The operating system is also past its support date. The host's end-of-life notice was sent to an email address nobody at Harlow Ridge monitors.

The fix. Build a new server on supported software, move the portal to the newest PHP version CakePHP 2 can run on as an interim step, test it against a copy of your data, and switch over outside business hours with the old server kept as a fallback for two weeks.

U3 The only copy of the code is on the live server — $600

What it means. There is no history of what changed or when, no safe place to test a change, and no way back if an edit breaks something.

What we found. The code is not in version control. Files on the server have been edited in place, most recently in 2021. Three files have backup copies with names like orders.php.old sitting beside them, and one of those is reachable from the web.

The fix. Put the code into a private repository you own, remove the stray copies, and set up a staging copy of the portal so every future change is tested before it goes live.

U4 Three pages build database queries from what the visitor types — $1,400

What it means. Someone who knows the technique could read or change data they should not see, including other customers' orders, without logging in as them.

What we found. The order search, the customer login and the invoice lookup each put text from the request straight into a database query. This is the classic "SQL injection" weakness. The rest of the portal uses the framework's safe query methods; these three were written by hand.

The fix. Rewrite the three queries to use the framework's safe methods, add tests that prove the weakness is closed, and review the server logs for signs it was used.

U5 Passwords are stored in a form that is easy to reverse — $1,100

What it means. If the customer table were ever copied, most passwords could be recovered within hours, and customers who reuse passwords elsewhere would be exposed too.

What we found. Customer and staff passwords are stored with an outdated method (unsalted SHA-1). Six staff members also share one administrator login, so there is no record of who changed what.

The fix. Move to modern password storage, upgrading each account automatically the next time its owner logs in. Give each staff member their own login and retire the shared one.

What to plan for

None of these will hurt you this month. Each should be scheduled within the year.

P1 The framework upgrade, one step at a time — $2,400

What it means. CakePHP 2 is no longer maintained and cannot run on current PHP. Until the framework moves, the portal stays tied to old software.

What we found. The portal is on CakePHP 2.6. The safe route is 2.6 to 2.10, then 3, 4 and 5, with a working release at each step. About a fifth of the code uses features that changed between versions; the rest should move mechanically. Two third-party plugins have no maintained successor and need replacing.

The fix. Do the first step (2.6 to 2.10) now, which is low risk and unlocks the interim PHP move. We will quote each later step as a fixed price once the one before it is done.

P2 Key accounts are in a former contractor’s name — $500

What it means. If the domain or the security certificate lapses, or the payment gateway needs a change, nobody at the company can act.

What we found. The domain registration, the hosting account and the payment gateway’s developer keys are registered to the original developer’s personal email address. The security certificate is renewed by hand and expires in 71 days.

The fix. Transfer each account to a company-owned address, record who holds what, and make certificate renewal automatic.

P3 The order history page takes nine seconds for large customers — $700

What it means. Your biggest customers have the slowest experience, and it gets slower every month as orders accumulate.

What we found. The orders table has 2.3 million rows and is searched by customer and date, but it has no index on either. The page also runs one extra query for every order it lists.

The fix. Add the two missing indexes and load each page of orders in one query. We measured under half a second on a copy of your data with both changes.

P4 Uploaded invoices can be opened by anyone with the link — $800

What it means. Invoice PDFs are not protected by the login, and their addresses follow a guessable pattern.

What we found. Invoices are saved in a public folder and named by invoice number. Opening one does not check who is asking.

The fix. Move the files out of the public folder and serve them through the portal, which checks that the invoice belongs to the logged-in customer.

P5 The nightly inventory import fails silently — $600

What it means. When the import stops, the portal shows yesterday’s stock levels and customers order items you do not have.

What we found. A scheduled script pulls stock levels from your warehouse system each night. It is not documented anywhere, it lives in the former developer’s home folder on the server, and it failed on 11 of the last 90 nights without anyone being told.

The fix. Move the script into the codebase, document it, and send an alert when a run fails or does not happen.

What can wait

- About 6,000 lines of code are no longer reachable from any page and can be removed during the upgrade.
- There are no automated tests. We add them around each area we touch, so coverage grows with the work.
- Email is sent by the server itself, so some messages land in spam. Moving to a mail service is a half-day job when convenient.
- The portal is hard to use on a phone. That is a design project, not a risk, and is outside this report.
- Error messages show technical detail to visitors. This is fixed as part of the server move.

Costs at a glance

ITEM	PRIORITY	FIXED PRICE
U1. There is no backup that can be restored	Urgent	$900
U2. PHP and the database stopped receiving security fixes at the end of 2018, and the host is retiring the server	Urgent	$3,200
U3. The only copy of the code is on the live server	Urgent	$600
U4. Three pages build database queries from what the visitor types	Urgent	$1,400
U5. Passwords are stored in a form that is easy to reverse	Urgent	$1,100
All urgent items		**$7,200**
P1. The framework upgrade, one step at a time	Plan for	$2,400
P2. Key accounts are in a former contractor's name	Plan for	$500
P3. The order history page takes nine seconds for large customers	Plan for	$700
P4. Uploaded invoices can be opened by anyone with the link	Plan for	$800
P5. The nightly inventory import fails silently	Plan for	$600
All items to plan for		**$5,000**

Each price is fixed. Items can be done together or one at a time, by us or by any developer you choose.

What we looked at

Application

- Code condition and architecture
- Security exposure and vulnerabilities
- PHP and framework version risk
- Hosting and backup situation
- Bus-factor risk: who knows what

Database

- Performance and query efficiency
- Indexing and optimization gaps
- Backup integrity, tested by restoring
- Version risk and upgrade path
- Data safety and recovery readiness

We reviewed all 410 first-party files. We did not assess the warehouse system the portal imports from, or the payment gateway's own configuration, because we had no access to either.

How this review was done

Senior engineers led this review and used AI tooling to read the whole codebase instead of a sample. An engineer verified every finding in this report. Your code was not used to train models, and no production data or customer records were sent to a model, only schema and query statistics.

What happens next

This report is yours to keep, and you can take it to any developer. If you would like us to do the work, the 5 urgent items make up one Stabilization Sprint with a fixed price of $7,200. Book it within 60 days and the $2,500 Health Check fee comes off, leaving $4,700.