A WordPress database error can take a site offline without warning. Visitors may see “Error establishing a database connection,” a blank page, or a server error. While you investigate, a clear maintenance page can reassure customers, protect your business’s reputation, and give people a way to get in touch.

The quickest safe approach is usually to serve a small, static maintenance page that does not depend on WordPress or its database. Start with reversible checks, save a record of what you see, and make a backup before changing configuration. This guide walks through rapid triage, database credentials, hosting health, the basics of wp-config.php, and options for server-level maintenance or a DNS temporary switch.

Start with Fast, Reversible Checks

Before editing files or changing settings, find out whether the problem is ongoing, widespread, or limited to one device or network. A few minutes of careful checking can prevent unnecessary changes during an outage.

1. **Check the site in a private browser window.** Try both the homepage and a page that normally loads content. This helps rule out a stale browser cache or a logged-in-only issue.
2. **Check from another connection.** Use a phone with Wi-Fi turned off, or ask a trusted colleague to test. If the site works elsewhere, you may be seeing a local network or DNS issue.
3. **Record the exact message.** Write down the error exactly as displayed, the time and time zone, the affected URL, and whether the WordPress login page also fails. Avoid publishing technical details that could expose sensitive information.
4. **Look for a recent change.** Note recent deployments, plugin or theme updates, hosting changes, password rotations, or domain migrations. Don’t start by installing a “repair” plugin; a database outage may prevent it from working, and adding software makes the incident harder to diagnose.
5. **Check your hosting dashboard.** Look for an outage notice, database status, disk usage warnings, and resource limits. Start with read-only checks wherever possible.

Scratchpad example: “2026-10-15, 09:20 UTC — homepage returns ‘Error establishing a database connection’; wp-admin shows the same; mobile network also affected; hosting dashboard reports database service unavailable.” Keep this record as the incident develops.

Put Up a Static Maintenance Page

A **static maintenance page** is a plain HTML file that can load without WordPress, PHP, plugins, or a database. It is often the fastest option when the site's application is broken but the web server and file access are still available.

Prepare a simple page

Create a file named maintenance.html on your computer. Keep it lightweight and include only information visitors need: that the site is temporarily unavailable, that you're working on it, and a working contact method. Don't promise a restoration time unless you can support it.

We'll be back shortly

Our website is temporarily unavailable while we investigate a technical issue.

For urgent assistance, call [phone number] or email [email address].

Use a phone number or email address that does not depend on the affected website. Don't include account details, database errors, or internal troubleshooting information in the public message.

Serve the page without WordPress

If your hosting panel provides a maintenance-page or static-site setting, use that first and note how to switch it off. Otherwise, an administrator with file access may be able to place a static file in the site's document root and temporarily configure the web server to serve it. The right method depends on the host and server; avoid copying generic rewrite rules into production without understanding their effect.

On Apache, rewrite directives may be in .htaccess; on Nginx, they are commonly in server configuration managed by the host. A misplaced rule can block the site or interfere with verification and support access. If you edit a configuration file, download a copy first and make one change at a time. Confirm that the maintenance page loads over HTTPS and that your contact details display correctly.

Approach Useful when Watch out for Hosting-panel maintenance setting Your provider offers a simple on/off control Confirm whether it serves a static page or relies on WordPress Static HTML with server-level maintenance The web server and file access work, but WordPress does not Back up configuration and keep a clear rollback path DNS temporary switch The original web server is unavailable and you control a ready alternate destination DNS changes take time to propagate and may affect email if records are changed carelessly

Use Server-Level Maintenance Carefully

Server-level maintenance means the web server, hosting platform, or an upstream service serves the maintenance response instead of relying on the WordPress application. This is particularly useful during database errors because WordPress itself may be unable to render a maintenance page or run a plugin.

If the site is behind a content delivery network (CDN), check whether it offers a maintenance or custom error-page feature. A CDN may be able to serve a prepared page while the origin server is being repaired. Test the result from a separate connection, since a cached copy of the old site may otherwise mislead you.

Some WordPress guides recommend creating a root-level .maintenance file. That file is intended for WordPress update maintenance and may not display the page you want during a database failure. Don't assume it will replace a server-level maintenance page.

Fix — What changed: A static page or server-side rule temporarily replaces normal site responses. **How to roll back:** Remove the temporary rule or turn off the hosting-panel setting, then restore the saved configuration if needed. Test the live site after rollback.

Check Database Credentials in wp-config.php

If the database service is available, incorrect credentials are a common cause of WordPress connection errors. The file wp-config.php, usually in the WordPress installation directory, contains the settings WordPress uses to connect. Access it through your host's file manager or secure file transfer—not by making it publicly accessible or sharing its contents in a support forum.

The relevant constants typically look like this:

define('DB_NAME', 'database_name'); define('DB_USER', 'database_user'); define('DB_PASSWORD', 'database_password'); define('DB_HOST', 'localhost');

Your values may differ. In particular, DB_HOST is not always localhost; some hosts provide a hostname, port, or socket value. Use the values shown in your hosting account rather than guessing.

1. **Back up the file.** Download a copy of wp-config.php before editing. Store it securely because it contains secrets.
2. **Compare the settings with the hosting dashboard.** Confirm the database name, database username, password, and host. If the database password was recently changed, WordPress may still have the old one.
3. **Check for accidental formatting changes.** Look for missing quotes, semicolons, or copied spaces. Don't paste passwords into chat or email unless you have an approved secure process.
4. **Change only what you can verify.** If the host confirms the correct database password, update only the relevant setting, save the file, and test once. Avoid changing multiple values at the same time.

Do not share your database password, full configuration file, or database export publicly. If you suspect credentials have been exposed, arrange a controlled password rotation with the hosting provider and update WordPress securely.

Fix — What changed: A verified database connection value was corrected in wp-config.php. **How to roll back:** Restore the private backup if the edit causes a new problem, then verify the correct values with the host before trying again.

Check Hosting and Server Health

Correct credentials won't help if the database service is stopped, overloaded, or unable to accept connections. Check the hosting dashboard and logs for the database's status, recent restarts, resource limits, disk space, and connection limits. Review error logs around the timestamp in your scratchpad, but redact passwords and other sensitive details before sharing excerpts.

- **Database service:** Is it reported as running or experiencing an incident?
- **Storage:** Is the account, database, or server out of disk space? Full storage can prevent writes and disrupt services.
- **Resources:** Are CPU, memory, or database connection limits being reached?
- **Recent changes:** Did a migration, backup job, deployment, or traffic spike happen just before the outage?

- **Other sites and services:** Are other applications on the same account affected? This can help distinguish a WordPress-specific issue from a server-wide incident.

Don't restart services or run database repair commands just to "see if it helps." Those actions can interrupt other sites or make recovery harder. Before any repair operation, make sure a restorable backup exists and understand whether the operation changes data. If the dashboard shows a platform outage or you lack permission to inspect the service, send the host a specific, evidence-based request.

A useful message to your hosting provider

Instead of a vague request, provide the domain, exact error, first observed time and time zone, whether the issue affects the whole site, and what you already checked. Ask whether the database service is healthy and whether logs show connection failures, resource limits, or a recent restart. Do not send passwords.

When a DNS Temporary Switch Makes Sense

A **DNS temporary switch** can point visitors to a separate static maintenance site when the original server cannot serve any page. It is a fallback, not the first move: you need control of the domain's DNS and an already prepared alternate destination. DNS changes can take time to reach users because resolvers cache records, so some visitors may continue to see the old site for a while.

Before changing DNS, record the current records and values. Change only the website records you intend to change, and take care not to alter mail-related records such as MX records or email authentication settings. After the incident, restore the original website records and verify both the site and email. If you use a CDN, load balancer, or managed DNS service, follow its documented process.

Fix — What changed: A web record temporarily points to a prepared alternate page. **How to roll back:** Restore the saved original record values, then check the site from multiple networks as DNS caches update.

Restore the Site and Verify It

Once database access is restored, remove the maintenance layer deliberately rather than assuming the site is fixed. Test the homepage, a representative inner page, the login page, and any important enquiry or booking form.

Check from a private window and another connection. Confirm the site uses HTTPS and that no maintenance rules or alternate DNS records remain.

Keep your scratchpad with the incident notes: exact error messages, timestamps, changes made, and the outcome of each test. This makes it easier to identify the real cause if the issue returns and provides a useful record for your hosting provider or developer.

Quick Outage Checklist

- Capture the exact error, affected URLs, and timestamps.
- Check from a private window and a second network.
- Serve a static maintenance page that does not depend on WordPress or the database.
- Back up wp-config.php and server configuration before editing.
- Verify database credentials against the hosting dashboard; don't guess.
- Check database status, disk space, resource limits, and relevant logs.
- Use a DNS temporary switch only when the original server cannot serve a page and a prepared alternate is available.
- For every change, record what changed and how to roll back.
- After recovery, test key pages and forms, then remove temporary settings.

A calm, reversible response is usually more effective than a rushed repair. Put up a static https://fantasticlawyers.com/2026/10/11/best-criminal-lawyers-brisbane/ maintenance page, preserve the evidence, and investigate the database connection one verified step at a time. Visitors get a clear message while you work toward a safe restoration.