



SYSTEM CATALOG 02 ● HOSTING PANEL

Hosting Panel Catalog

A local hosting control panel for domains, websites and site files - built to show the machine's real state instead of keeping a second copy of the truth.

fbpanel

single Node.js program

live machine readings

prototype - episode two

Reference implementation for the FB Server project. The panel is designed for a home-network server and provides a visual alternative to terminal-first hosting administration.



What FB-PANEL is

02

**"Nothing on screen is invented:
every number is read from the machine."**

FB-PANEL is a hosting control panel for one server. After signing in, six sections let the owner give the server a domain name, create a website and manage that website's files without opening a terminal.

Its shape is familiar if you have used cPanel or Plesk, but this implementation is intentionally small: one readable program, local tools, a narrow privilege model and an activity log for changes.

6 sections

4 currently working

local network

**ONE PROGRAM****Node.js web server**

Serves every page itself. There is no separate front end to drift out of sync.

**REAL TOOLS****System helpers**

Website, DNS, WordPress, file-permission and theme work is delegated to named tools.

**LIVE STATE****The machine is authority**

Most displayed state is read directly from nginx, services, disks and files rather than duplicated in the panel.

Sign in, stay local, leave a trail

03

SERRA / OPERATIONAL CONTEXT

The FB Server crew aesthetic is used here as the visual language around the panel, while the panel itself remains a practical administrative tool. Security is intentionally narrow because a successful sign-in can take a website offline.

Password storage

The password itself is never stored. A deliberately slow derivation process is used with a unique random value for the account.

Constant-time comparison

Sign-in checks are compared in a way intended not to reveal information through how quickly a wrong password is rejected.

Signed browser ticket

The browser receives a server-signed ticket that page scripts cannot read, is not sent to other sites and expires after 12 hours of inactivity.

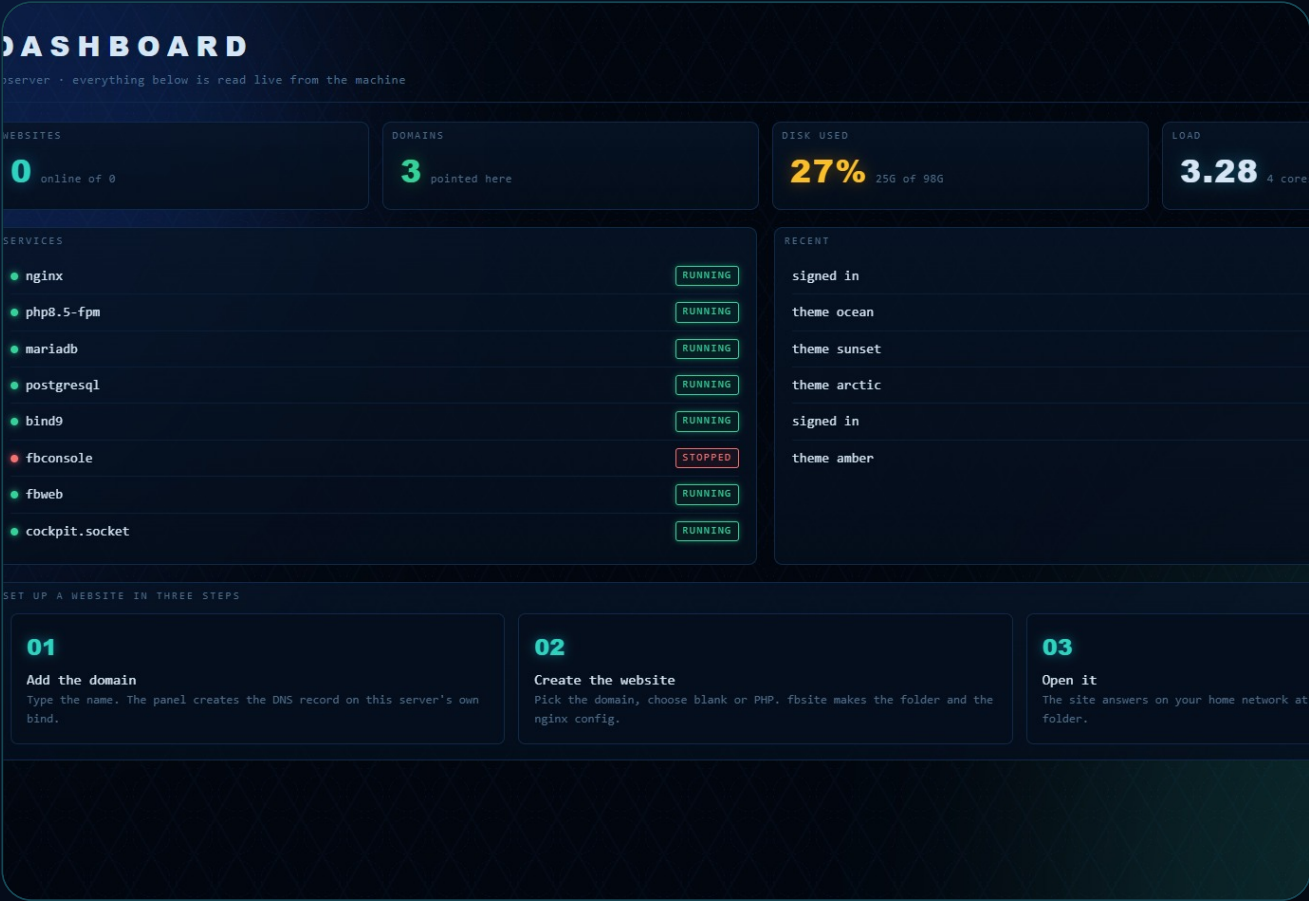
Failed attempts are logged

Rejected sign-ins are written to the activity log with the username that was tried.

Owner password setup: the password is read from standard input rather than command-line arguments, so it does not appear in terminal history, the process list or on screen. Minimum length: eight characters.



The first page after sign-in



FB PANEL DASHBOARD · example capture from 20 September 2026

Websites

Shows how many sites are switched on out of how many exist. A disabled site still exists; it simply stops answering.

Domains

Shows how many names are pointed at this server.

Disk used

Displays the percentage full with the actual figures beside it.

Load

Shows how hard the machine is working, together with core count and uptime so the load figure has context.

Example screenshot: 0 of 0 websites online, 3 domains, 27% disk used and load 3.28 on 4 cores. These are snapshot values, not fixed product specifications.

SERVICES

Eight components are checked directly: nginx, PHP, MariaDB, PostgreSQL, bind9, fbconsole, fbweb and Cockpit. Green means running, red means stopped.

RECENT

The last six activity-log entries show what the panel changed, who requested it and when. This is the panel's memory of itself.

Names this server answers to

05

ADDING A DOMAIN

Adding a domain records the name in the panel and creates a DNS record on the server's own name server so devices on the home network can find the site by name.

The name is checked against a strict pattern for length, allowed characters and overall domain shape. Invalid or duplicate names are refused before anything is created.

READING THE DNS STATUS

`bind` · A `record` means the local DNS record was created successfully. `not on bind` means the name is in the panel list but the DNS record was not made.

REAL INTERNET DOMAINS

The panel creates records on this server. A public domain you own is still pointed here from the registrar where the domain was purchased. That is a separate external step.

REMOVING ONE SAFELY

Removal asks for confirmation, removes the DNS record and removes the name from the panel list. The old DNS zone file is preserved in a folder for removed zones.

If a website is still using the domain, the panel refuses the removal and tells you to remove the website first.

Design principle: the panel says when a record was not created. It does not hide the condition or silently pretend the operation succeeded.

Sites hosted on this server

MEASURED, NOT REMEMBERED

The Websites page does not keep its own website list. Each time you open it, the panel reads nginx's configuration folder, opens each configuration it owns and reports what it finds.

Only configurations marked as created by the site tool are offered for management. Hand-written configurations are left alone.

ENABLE / DISABLE

Disabling takes the site offline without touching its contents. Enabling puts it back.

REMOVE DOES NOT MEAN DESTROY

Site files are moved into a folder for removed sites rather than deleted. For WordPress, the database is exported first so the site's content remains recoverable.

HOW TYPE IS DETECTED

A WordPress configuration file in the folder identifies WordPress. A PHP handler in nginx identifies PHP. Neither means a plain site. The enabled state comes from the same nginx link structure the web server itself uses.

Blank

Creates a plain website for HTML files.

PHP

Creates a site with PHP handling enabled.

WordPress

Installs WordPress plus its database, user, configuration and fresh security keys.

Before creation: the panel lists what will happen - folder creation, nginx configuration, testing, reload and any WordPress-specific work. Afterward, the real tool output is printed on screen.

A file manager that stays inside one website

07

SITE SCOPE

The site picker chooses the website you are working in. Everything below that point is intended to exist inside that site's folder and nowhere else.

THE WALL

Every file operation resolves the requested path to its real final location - following shortcuts and attempts to go up a level - then checks that the final destination is genuinely inside the selected site's folder.

If it is outside the site, the operation stops.

Why this approach: suspicious path text is not stripped or guessed at. The real destination is resolved first, then the boundary is checked.

REQUEST

User picks a path

Browse, edit, upload, rename or move to trash.



RESOLVE

Find the real destination

Follow links and normalize parent-directory movement.



CHECK

Inside selected site?

Only then is the operation allowed to continue.

Browsing

Folders first, then files, alphabetically, with type, size and last-change time.

Editing

Text files under 1 MB can be edited in the browser. Ctrl+S saves and Tab indents.

Uploading

Drag and drop or upload button. Up to 200 MB per file, one at a time.

Private folders hidden

The trash and saved-version folders are intentionally omitted from the normal file listing.

Versions, trash and quiet permission repair

08

EVERY SAVE KEEPS THE PREVIOUS FILE

Before a text file is overwritten, the old version is copied into a versions folder and named with the date and time.

An edit that breaks the site can be rolled back from the saved copy.

THERE IS NO DELETE BUTTON

Trashing moves an item into a trash folder inside the same site and records where it came from and when.

Restoring puts it back. If the old location is occupied, the restored item is renamed instead of overwriting.

PERMISSIONS ARE HANDLED QUIETLY

Before a write, the panel checks whether it can write to the website folder. If not, it asks a dedicated file-permission tool to correct ownership.

If permission repair fails, the panel reports the failure rather than half-completing the operation.

Rename protection

Names beginning with an underscore are refused because panel-private folders use that convention. Renaming over an existing name is also refused.

Trash is deliberately not emptied in the UI

Items remain in trash until a person empties the folder from a terminal. Permanent deletion is kept as an explicit human action outside the panel.

Reversible by default.
Creation can be automated; destruction stays deliberate.

FB Panel inside the wider FB Server system

09



THE CORE EIGHT · FB Server visual universe



SERRA AND THE CREW

The characters are part of the FB Server presentation layer and story world. The panel remains a real machine interface beneath that aesthetic.

ACCOUNT

The owner can change the password by providing the current password and entering the new one twice. The panel also shows the account list and last sign-in time.

SHARED THEME SYSTEM

FB Panel uses the same theme library as the other FB Server screens. A selection changes immediately in the browser and is also sent to the server as the shared theme.

Dynamic links: the panel works out the server's current network address itself for the three outgoing server links, so they do not depend on a manually typed address.

Crew Dashboard

Link out to the server's live operational display.

Pipeline

Link out to the server's separate pipeline screen.

Cockpit

Link out to the system's own management console.

What the panel can change - and how that power is contained

10

The panel itself runs as an ordinary program. Tasks that genuinely need administrator power are delegated to a short, named list of tools. Each tool receives only the right needed for that one task.

SITE TOOL

Make, enable, disable and remove websites.

DNS TOOL

Add and remove local DNS records.

WORDPRESS TOOL

Install or remove WordPress, its database and related files.

FILES TOOL

Correct website-folder ownership so files can be edited.

THEME TOOL

Set the theme shared by the machine's FB Server screens.

NO ARBITRARY COMMAND PATH

The program does not build free-form commands from what is typed into forms. The helper program is named separately from its arguments, narrowing what user input can cause the machine to do.

EVERY CHANGE ENTERS THE ACTIVITY LOG

Operations are written down with the account name and time, giving the panel a trace of its own actions.

BOUNDARY

The panel is not intended for the public internet.

The source document places the risk at the sign-in boundary: a local-network password opens access to actions that can take a website offline. That is why sessions expire, failed attempts are logged and the panel is designed for the house network.

Put a WordPress site on the server

01

Add the domain

Open Domains, enter the name and add it.
Check that the DNS column reports a bind A record before continuing.

02

Create the website

Open Websites, choose New website, select the domain and choose WordPress.

03

Read the real output

The panel displays what the underlying tools actually printed. The final output tells you where to continue.

04

Finish WordPress

Open the new site and complete WordPress's own setup. The panel stops at the boundary of what it owns.

05

Work on the files

Use Files to upload, edit or organize the site.
Saved versions and trash provide two recovery paths if something goes wrong.

HOSTING PANEL

PANEL

- DASHBOARD
- DOMAINS
- WEBSITES
- DATABASES
- FILES
- EMAIL

SERVER

- CREW DASHBOARD
- PIPELINE
- COCKPIT

YOU

fbserver • everything below is read live from the machine

SIGNED IN AS ADMIN

WEBSITES

0online of 0

DOMAINS

3pointed here

DISK USED

27%25G of 98G

LOAD

3.284 cores • up 9 h

SERVICES

- nginxRUNNING
- php8.5-fpmRUNNING
- mariadbRUNNING
- postgresqlRUNNING
- bind9RUNNING
- fbconsoleSTOPPED
- fbwebRUNNING
- cockpit.socketRUNNING

RECENT

- signed in2026-09-20 15:41
- theme ocean2026-09-20 15:41
- theme sunset2026-09-20 15:41
- theme arctic2026-09-20 15:41
- signed in2026-09-20 15:40
- theme amber2026-09-20 15:40

FB PANEL CATALOG • WORDPRESS WORKFLOW

DOMAIN → SITE → WORDPRESS → FILES → RECOVERY

What is not built yet

12

The panel deliberately shows unfinished areas instead of hiding them. Database and Email are visible in the sidebar with a SOON label because they are planned, not complete.

One owner

No customer accounts yet, so there is no per-customer site access model.

No certificates

Sites are served without encryption. The source document treats this as acceptable only for the home network, not for public internet deployment.

Home network only

Remote internet access is explicitly outside this version's scope.

No database page

WordPress can receive a database automatically, but manual database management is not yet implemented.

No email

Email management is not implemented in this prototype.

One site per domain

This is a stated design rule in the current implementation.

DATABASES - SOON

The machine already runs database engines and WordPress site creation can create what it needs. The missing part is a human-facing management page.

EMAIL - SOON

The source document treats mail hosting as a later problem because operating a mail server well is substantially harder and has abuse risks if done poorly.

Not unfinished by accident.
The interface says where the current build ends.



FB SERVER ECOSYSTEM • HOSTING CONTROL

Local hosting, visible decisions.

FB Panel is the management side of the project: domains, websites and site files in a graphical interface, while the Crew Dashboard remains the read-first operational screen.

live machine state

narrow privileges

logged changes

recovery by default

"Nothing on screen is invented:
every number is read from the machine."



CATALOG IDENTITY

FB Panel shares the FB Server visual universe with Serra and the agents. The operational claims here come from the supplied running-panel documentation and dashboard capture.

