

Web State & Completion Audit

An owned website decomposed into verified states, gaps, dependencies and completion evidence.

15	7	3	0
OBSERVED STATES	COMPLETION MOVES	STATE CLASSES	CLIENT SYSTEMS USED

Executive result

An ambiguous 'is the site finished?' question becomes a register of verified source states, gaps, intentional boundaries, live-check residuals and acceptance evidence.

What this sample proves

A web project can be managed by observable state rather than a vague design label. Implementation, verification and governance work remain distinguishable.

Use boundary

The audited surface is owned by Lilambd. This is not a client audit, security assessment, uptime certificate or claim about current production behavior.

State transformation

01 Chaos

Routes, metadata, language semantics, downloads, mobile behavior and third-party links exist in different states and are easily collapsed into 'done' or 'not done.'

02 Operation

Source observation → route and dependency register → state classification → severity → completion condition → EOS/human boundary.

03 Usable state

A completion map shows what can be fixed deterministically, what needs live evidence and which decisions remain intentional governance boundaries.

Uncompressed residual

Live responses, cross-device behavior, external destinations and aesthetic judgment require fresh observation.

Observable proof

- 15 observable website states
- Verified / gap / live-check separation
- Seven acceptance conditions
- EOS completion and human judgment split

Portfolio integrity

Independently constructed self-directed technical work using public or synthetic inputs. Not a reproduced marketplace brief, confidential client engagement, paid result or claim of deployment.