Project Brief: FF Content Hub

Client: Feedforward
Working title: FF Content Hub
Target launch: Monday, December 1e

1. Overview

We aim to build a **password-protected content hub** on our website powered by a **simple CMS**, with access via **Discord login**. The hub should make it easy for our members to find key documentation, tools, and archives organized by topic.

Initial architecture (Phase 1) should support:

- Vendors
- AI & governance
- Archives – pods
- Archives – meeting notes
- Tools hub (Librechat, other)

2. Objectives

- Build a **documentation-style site** with a lightweight CMS so non-technical staff can manage content.
- Use **Discord OAuth** for authentication and restrict access to our community members.
- Provide clear, extensible navigation by topic and archive type.
- Deliver something that's easy to maintain and extend in future phases.

3. Key Requirements

Authentication & Access

- Not publicly listed in main site navigation.

- “**Login with Discord**” flow; successful auth grants access to the hub.
- Preferably restrict by Discord server membership/roles if feasible in scope.

Content Structure

FF Content Hub home

- Short intro and overview.
- Navigation tiles/links for:
  - Vendors
  - AI & governance
  - Archives – pods
  - Archives – meeting notes
  - Tools hub (for Librechat instance and other tools to come)
- Optional “What’s new”/recent updates list.

Topic pages (Vendors, AI & governance, FF Tools)

- Short description of the topic.
- List or grid of content items (articles, templates, links, tools).
- Support for tags or basic filters if low effort.

Archives

- Pods archive: list of pod-related content, sortable by date; optional filter by pod and tags.
- Meeting notes archive: searchable/browsable list of meeting summaries with date, title, tags, and link to detail.

Tools hub (LibreChat, etc)

- Dedicated section or page with description and either:
  - Embedded chat widget, or
  - Prominent link out to existing chat experience.

CMS Functionality

Admins should be able to:

- Create/edit/delete content items with:
  - Title, summary, body (rich text/Markdown), topic, type, tags, publish status.
- Assign each item to one or more topics (Vendor, AI governance, etc.).
- Publish/unpublish without code changes.
- Reorder navigation labels/sections for Phase 1.

UX & Technical

- Clean, responsive design (desktop + mobile).
- Fast page loads (static or SSR preferred).
- Reasonable accessibility (semantic HTML, alt text, focus states).
- Secure handling of Discord OAuth tokens.

We are flexible on stack (e.g., Next.js + headless CMS) and hosting; please propose an approach you're comfortable supporting.

4. Deliverables

By **Dec 15**, we expect:

1. **Working, deployed FF Content Hub** with Discord login and Phase 1 topics implemented.
2. **CMS configuration** with example content populated for each section.

3. **Source code** in a repo we can access.

4. **Short admin guide** (written doc or Loom) explaining:

    - How to log in to the CMS

    - How to add/edit/publish content

    - How to manage navigation/sections

5. Brief notes on known limitations and recommended Phase 2 enhancements (if any).

5. Process & Checkpoints

Between project start and Dec 15, we'd like:

- Quick **requirements/architecture confirmation** at project start.

- **Wireframes or low-fi mockups** for the hub home, topic page, archives views, Libre chat, and login state (1–2 review rounds).

- **Staging/beta deployment** before final launch for testing and feedback.

- Time for **1–2 iteration cycles** based on our feedback.