**🧩 System Architecture (High-Level Components)**

pgsql

CopyEdit

Users ───┐

├──▶ [ User Portal (Space) ]

│ ├── Profile

│ ├── My Websites

│ ├── Create New Website

│ └── Admin (Website Config / Template Editor)

│

└──▶ [ Authentication Service ]

└── Invite-only Signup (Google, GitHub, Email)

Sys Admin ──▶ [ Admin Dashboard ]

├── All Users

├── Invite Hierarchy

├── Website Publish Requests

└── Notifications Panel

Both ──▶ [ Backend Service (API) ]

├── CSV Upload & SQLite Conversion

├── Schema Editor & FTS Config

├── Template Save / Load

└── Publish Request System

└──▶ [ Hosting via Fly.io (Datasette Instances per User Website) ]

**🌐 User Portal — After Signup**

**🧑‍💻 Auth Options:**

* Invite-only
* Google, GitHub, or email-based signup
* Each user has a **dedicated space (dashboard)**

**🧭 Dashboard Tabs:**

1. **Profile**
   * View/change personal info
   * Change password (or link/unlink OAuth)
2. **My Websites**
   * List of user-created datasets/websites
   * Status: Draft | Pending Review | Published | Archived
3. **Create New Website**
   * Name → becomes subdomain (e.g., airquality.edgi.cloud)
   * Upload CSV or URL → auto-convert to SQLite
   * Choose icon, description, tags
   * Click “Create”
4. **Schema & Search Config (per table)**
   * FTS toggle (checkbox per column)
   * Foreign key mapping
   * Data type editing
5. **Website Template Editor (admin.html → template.html)**
   * HTML editor with preview
   * Save & submit for publishing
6. **Publish Request**
   * Button: “Submit to Admin”
   * Confirmation: “Your request has been sent. You’ll be notified.”

**👩‍💼 Admin Dashboard**

**🔒 Accessible to System Administrators only**

**Views:**

* **User Management**
  + View all users
  + Invite new user
  + View invite tree (who invited whom)
* **Website Requests**
  + View all pending publish requests
  + Review template, schema config
  + Approve / Reject / Send Feedback
* **Notifications Panel**
  + Activity log
  + Schema errors, new submissions
* **Global Settings**
  + Tweak global template defaults
  + Manage system-wide announcements

**📦 Data Import & Management**

**Allowed Inputs:**

* **Upload CSV** (drag & drop or file picker)
* **Import CSV from URL**

**Automated Steps:**

1. Convert to SQLite
2. Create tables in user’s dedicated DB
3. Auto-detect schema
4. Allow user to edit types, FTS, FK

**🚀 Website Publishing Flow**

mermaid

CopyEdit

flowchart TD

A[User Uploads CSV] --> B[SQLite Conversion]

B --> C[Schema Config & FTS Setup]

C --> D[Customize Template (template.html)]

D --> E[Submit Publish Request]

E --> F[Admin Reviews]

F -->|Approve| G[Website Deployed to Fly.io]

F -->|Reject| H[Feedback Returned to User]

**🖼️ Visual/UX Considerations**

* 🪄 **Clean, minimal dashboard UI** using TailwindCSS + Heroicons
* ✨ Modals for upload, FTS config, and submit request
* 🧠 Use JSON schema + YAML config behind-the-scenes
* 📦 Folder per user on /data/{username}/db.sqlite
* 🔗 Each site: https://{subdomain}.edgi.cloud

**🔌 Optional Enhancements**

* 🔍 Built-in preview mode (before publishing)
* 🧪 Version history for website template
* 📊 Simple analytics (visits, downloads per dataset)
* 🗂️ Tag-based or category browsing (air, water, soil, etc.)
* 🤝 Federated search across all public datasets

**✍️ Invite-Only Access Model**

**Onboarding Flow:**

1. Admin generates invite code
2. User visits signup/ with invite
3. Chooses auth method (Google, GitHub, email)
4. Redirected to personal dashboard (their "space")

**💡 Platform Inspiration Summary**

| **Platform** | **Borrowed Ideas** | **Relevance to EDGI** |
| --- | --- | --- |
| **Datasette Cloud** | Isolated DBs per user, Fly.io deploy, CSV import | Core infrastructure model |
| **Turso** | Scalable SQLite, copy-on-write | Long-term horizontal scaling |
| **Supabase** | RBAC, real-time, API-first | Inspiration for multi-user dashboard |
| **Airtable** | Friendly UI, roles | Optional future roadmap for community editors |

Also we can Add database size limits and quotas per user.

Provide UI for uploading datasets with validation steps.

Build monitor database usage and activity logs in admin dashboard.

User authentication at least confirmation of the email? (git, google account is ok)

**Updated Workflow**

Based on your architecture, here’s the refined localhost workflow:

1. **Signup**:
   * Users visit /register with an invite code (invite-only model).
   * Enter username, email, password, role, and invite\_code.
   * Redirected to /login upon successful registration.
2. **Login**:
   * Log in via /login to access the user portal.
3. **User Portal (Dashboard)**:
   * **Index Page (/)**: Displays “Profile”, “My Websites”, and “Create New Website” sections.
   * **Profile**: View/edit username, email, and change password via /change-password.
   * **My Websites**: Lists user databases with status (Draft, Pending, Published, Archived) and links to edit (/template/<db\_id>).
   * **Create New Website**: Access /create-database to upload CSV/SQLite, set icon/description/tags, and create a database in /data/username/db\_name.db.
   * **Template Editor**: Use /template/<db\_id> to customize website content (title, header\_image, etc.) and submit for publishing.
4. **System Admin Dashboard**:
   * Accessible via /system-admin for role="admin" users.
   * Includes:
     + **User Management**: View users, invite new users, see invite hierarchy.
     + **Publish Requests**: Review, approve/reject, and provide feedback on website submissions.
     + **Notifications**: View activity logs (e.g., database creation, publish requests).
5. **Website Publishing**:
   * Users submit websites via /publish-request from /template/<db\_id>.
   * Admins review requests in /system-admin, updating status to Published or Rejected.

**Analysis and Suggestions**

1. **Registration**:
   * The current register.html (artifact\_version\_id="2a67c0cc-1f55-4aec-bb31-72549366ced1") includes email and invite\_code, but your provided datasette\_admin\_panel.py doesn’t validate invite\_code. We’ll add this to support invite-only signup.
2. **User Portal**:
   * The index page (index.html) should serve as the user dashboard, with clear sections for “Profile”, “My Websites”, and “Create New Website”.
   * create\_database.html needs fields for description, icon, and tags, with a modal for upload confirmation.
   * template.html (renamed from admin.html) should be database-specific and include a publish request button.
3. **System Admin Dashboard**:
   * Create system\_admin.html for user management, invite generation, and publish request handling.
   * Add an invites table to portal.db for invite codes and an activity\_logs table for monitoring.
4. **UX Improvements**:
   * Use TailwindCSS modals for dataset uploads and publish requests.
   * Add validation feedback (e.g., file size, format) in create\_database.html.
   * Implement database quotas (e.g., 5 databases, 50MB each).
5. **Current File Issues**:
   * datasette\_admin\_panel.py lacks invite-only signup, publish requests, and system admin functionality.
   * admin.html is portal-wide; it should be template.html for user databases.
   * index.html needs a dashboard layout with profile and website sections.