**Updated System Architecture and Specification**

Below is the revised specification, incorporating the single portal.db approach, simplified table management, and homepage creation with unique URLs.

**High-Level Components**

1. **Frontend (HTML Templates with Jinja2 and Tailwind CSS)**:
   * **Templates**: index.html, login.html, register.html, dashboard.html, create\_database.html, manage\_databases.html, create\_homepage.html, system\_admin.html, change\_password.html.
   * **Purpose**: Provides interfaces for user authentication, database creation, table management (CSV upload/delete), homepage customization, and admin tasks.
   * **Features**:
     + **Navigation**: Consistent across all pages, linking to /dashboard, /manage-databases, /system-admin (for admins), /profile, /change-password, /logout.
     + **Manage Databases**: Displays user’s tables in portal.db (filtered by {username}\_{db\_name}\_\*), allows CSV uploads/deletions, save draft, and publish.
     + **Homepage Customization**: Edits title, description, header image, and footer for published databases, stored in web\_content.
     + **Alerts**: Success/error messages for user feedback.
   * **Styling**: Tailwind CSS with custom styles (.card, .btn, .alert, .upload-area) for consistency.
2. **Backend (Python with Datasette)**:
   * **File**: datasette\_admin\_panel.py.
   * **Purpose**: Handles HTTP requests, authentication, table management in portal.db, homepage content storage, and Datasette publishing.
   * **Key Functions**:
     + index\_page: Renders the homepage with featured published databases and statistics.
     + login\_page, register\_page, change\_password\_page, logout\_page: Manage user authentication (open registration, no invites).
     + dashboard\_page: Displays user profile and list of owned databases with links to manage or edit homepage.
     + create\_database: Creates a new logical database in portal.db by inserting metadata into databases and creating a table from a CSV (e.g., user1\_air\_quality\_data).
     + manage\_databases: Shows user’s tables, allows CSV uploads/deletions (draft only), and handles save draft/publish.
     + add\_table: Adds a new table to portal.db from a CSV (e.g., user1\_air\_quality\_newtable).
     + delete\_table: Deletes a user’s table from portal.db (draft only).
     + publish\_database: Sets status to Published, triggers a Datasette instance scoped to user’s tables.
     + create\_homepage: Updates homepage content in web\_content for published databases.
     + delete\_database: Removes database metadata and tables from portal.db (draft only).
     + system\_admin\_page: Manages users and portal content (for admins).
   * **Dependencies**: bcrypt, sqlite\_utils, pandas, bleach.
3. **Database (SQLite - portal.db)**:
   * **Tables**:
     + users: Stores user accounts (user\_id, username, password\_hash, role, email, created\_at).
     + databases: Tracks logical databases (db\_id, user\_id, db\_name, website\_url, status, created\_at).
     + admin\_content: Stores portal global content (section, content, updated\_at, updated\_by).
     + web\_content: Stores homepage content for published databases (db\_id, section, content, updated\_at, updated\_by).
     + activity\_logs: Logs user actions (log\_id, user\_id, action, details, timestamp).
     + **User Data Tables**: Created dynamically in portal.db (e.g., user1\_air\_quality\_data, user1\_water\_quality\_metrics) from CSV uploads.
   * **Purpose**: Centralizes all data (user accounts, database metadata, user tables, homepage content) in a single SQLite database.
4. **File System**:
   * Stores header images for homepages (e.g., data/{username}/{db\_name}\_header.jpg).
   * No separate .db files; all data tables are in portal.db.
   * Requires write permissions for image uploads.
5. **Authentication and Authorization**:
   * Uses signed ds\_actor cookies for session management (httponly, samesite="lax").
   * Role-based access: system\_admin for managing users and portal content, system\_user for creating/managing databases and homepages.
   * CSRF protection via ds\_csrftoken cookies and form tokens.
6. **External Integration**:
   * Publishes Datasette instances (e.g., user1-air\_quality.datasette-portal.fly.dev) using Fly.io, scoped to user-specific tables in portal.db.
   * Uses ODbL licensing for data.

**System Workflow**

1. **User Registration**:
   * Users register at /register (no invite required), creating a system\_user or system\_admin account in users.
   * Registration is logged in activity\_logs.
2. **Database Creation**:
   * Users navigate to /create-database or /manage-databases.
   * /create-database: Uploads a CSV, creates a table in portal.db (e.g., user1\_air\_quality\_data), and adds metadata to databases with user\_id as owner.
   * /manage-databases: Creates a default database (my\_database) if none exists, or manages existing ones.
   * Metadata includes db\_name, website\_url, status (Draft/Published).
3. **Table Management**:
   * At /manage-databases, users see their tables (filtered by {username}\_{db\_name}\_\*).
   * **Add Table**: Uploads a CSV, creates a new table in portal.db (e.g., user1\_air\_quality\_newtable).
   * **Delete Table**: Removes a table (draft only).
   * **Save Draft**: Placeholder (can save description to web\_content).
   * Changes are logged in activity\_logs.
4. **Publishing**:
   * Users publish via /manage-databases, setting status to Published in databases.
   * Triggers a Datasette instance (e.g., via Fly.io) scoped to the user’s tables (e.g., --table-filter=user1\_air\_quality\_\*).
   * The website\_url (e.g., user1-air\_quality.datasette-portal.fly.dev) points to the Datasette instance, rendering the homepage from web\_content.
5. **Homepage Creation**:
   * For published databases, users customize the homepage at /create-homepage/{db\_id}.
   * Updates web\_content with title, description, header\_image, footer.
   * Homepage is accessible via website\_url.
6. **Database Deletion**:
   * Users delete draft databases via /manage-databases, removing metadata from databases and tables from portal.db.
   * Logged in activity\_logs.
7. **Admin Management**:
   * system\_admin users manage users and portal content at /system-admin.
   * Actions logged in activity\_logs.

**System Specifications**

* **Platform**: Datasette (Python-based).
* **Database**: Single SQLite file (portal.db) for all data (users, metadata, user tables, homepage content).
* **Frontend**: Jinja2 templates with Tailwind CSS.
* **Authentication**: Cookie-based with bcrypt password hashing, CSRF protection.
* **File Storage**: Local file system for header images (data/{username}/).
* **Deployment**: Fly.io for published Datasette instances, scoped to user tables.
* **Security**: Input sanitization (bleach), HTTPS-only cookies (secure=True), CSRF protection (samesite="lax").
* **Constraints**:
  + Max 5 databases per user.
  + Max 10 tables per database.
  + Max file size: 50MB for CSVs, 5MB for images.
  + Supported file types: .csv, .jpg, .png.
  + Single owner per database (user\_id in databases).