Patristic Library Website - Epics and User Stories

I'll break down this project into logical epics with detailed user stories. This structure will help you build the site systematically.

Epic 1: Site Foundation & Architecture

Goal: Establish the basic technical infrastructure and project structure

Stories:

1. **Set up project structure**
   - As a developer, I need to create the initial project folder structure so that HTML files, CSS, and assets are organized logically
   - Acceptance criteria: Folders for pages, works, styles, scripts, and images exist
2. **Create design system**
   - As a developer, I need to establish consistent typography, colors, and spacing so that the site has a cohesive look
   - Acceptance criteria: CSS file with variables for fonts, colors, and common components defined
3. **Build responsive layout framework**
   - As a user, I want the site to work on mobile, tablet, and desktop so that I can read on any device
   - Acceptance criteria: CSS grid/flexbox system that adapts to different screen sizes
4. **Plan data structure for content**
   - As a developer, I need to organize how fathers, works, and parts are structured so that navigation logic is clear
   - Acceptance criteria: JSON structure or clear folder hierarchy documented for all content

Epic 2: Home Page Development

Goal: Create the main entry point listing all Church Fathers

Stories:

1. **Design and build home page layout**
    - As a visitor, I want to see a clear, attractive home page so that I understand what the site offers
    - Acceptance criteria: Header with site title, introduction text, and layout for father list
2. **Create Church Fathers list display**
    - As a user, I want to see all Church Fathers listed alphabetically so that I can easily find who I'm looking for
    - Acceptance criteria: Alphabetical list with clear visual hierarchy, possibly grouped by era or region
3. **Add search/filter functionality**
    - As a user, I want to search or filter the list of fathers so that I can quickly locate a specific person
    - Acceptance criteria: Search box that filters the list in real-time as I type
4. **Implement father profile cards/links**
    - As a user, I want each father's name to be clickable and show basic info so that I know what I'm clicking on
    - Acceptance criteria: Each father has a clickable element with name, dates, and brief descriptor

Epic 3: Author Works Page

Goal: Display all works by a selected Church Father

Stories:

1. **Create author page template**
    - As a developer, I need a reusable template for author pages so that all fathers' pages are consistent
    - Acceptance criteria: HTML template with sections for bio, portrait/icon, and works list
2. **Build author biography section**
    - As a user, I want to see biographical information about the Church Father so that I have context for their writings
    - Acceptance criteria: Section displaying dates, location, key facts, and historical context

3. **Display complete works list**
    - As a user, I want to see all available works by this author so that I can choose what to read
    - Acceptance criteria: Organized list of works, possibly categorized by type (treatises, letters, homilies, etc.)
4. **Add breadcrumb navigation**
    - As a user, I want to see where I am in the site hierarchy so that I can easily navigate back
    - Acceptance criteria: Breadcrumb showing Home > Author Name

Epic 4: Work Display System

Goal: Present individual works with proper navigation for multi-part texts

Stories:

1. **Create single-page work template**
    - As a developer, I need a template for works that are single documents so that they display cleanly
    - Acceptance criteria: Template that displays title, metadata, and full HTML content with good typography
2. **Build multi-part work index page**
    - As a user viewing a multi-part work, I want to see a table of contents so that I can navigate to specific sections
    - Acceptance criteria: Index page listing all chapters/books/parts with descriptions if available
3. **Implement work reader interface**
    - As a user, I want a clean reading experience with good typography so that I can focus on the text
    - Acceptance criteria: Readable font sizes, line spacing, margins; no distracting elements
4. **Add work metadata display**
    - As a user, I want to see information about the work (date, context, translation info) so that I understand what I'm reading
    - Acceptance criteria: Metadata section showing relevant details about the work

Epic 5: Navigation & User Experience

Goal: Enable smooth movement throughout the site

Stories:

1. **Implement consistent header/navigation**
    - As a user, I want a consistent navigation bar so that I can always get back to the home page or search
    - Acceptance criteria: Header present on all pages with logo/site name and key navigation links
2. **Add previous/next navigation for multi-part works**
    - As a user reading through a multi-part work, I want next/previous buttons so that I can read sequentially
    - Acceptance criteria: Navigation buttons at top and bottom of each part
3. **Create "back to works list" functionality**
    - As a user, I want an easy way to return to the author's works list so that I can explore other writings
    - Acceptance criteria: Clear button/link to return to author page from any work
4. **Build site-wide search**
    - As a user, I want to search across all content so that I can find specific topics or quotes
    - Acceptance criteria: Search function that returns relevant fathers, works, or text passages

Epic 6: Content Integration

Goal: Connect all existing HTML content files to the site structure

Stories:

1. **Create content mapping spreadsheet**
    - As a developer, I need a master list of all works and their file locations so that I can link everything correctly
    - Acceptance criteria: Spreadsheet listing each father, their works, parts (if applicable), and file paths
2. **Standardize HTML work files**
    - As a developer, I need all HTML work files to follow consistent formatting so that they display properly

    - Acceptance criteria: Script or process to wrap existing HTML in site template, add headers/footers
3. **Link all single-part works**
    - As a developer, I need to connect all single-document works to their author pages so that users can access them
    - Acceptance criteria: All links from author pages to works are functional
4. **Link all multi-part works**
    - As a developer, I need to create index pages and link all parts of multi-part works so that navigation works
    - Acceptance criteria: All multi-part works have index pages with working links to each part

Epic 7: Enhancement Features

Goal: Add features that improve the reading and research experience

Stories:

1. **Implement reading preferences**
    - As a user, I want to adjust text size and theme (light/dark mode) so that I can read comfortably
    - Acceptance criteria: Controls for text size and theme that persist across sessions
2. **Add bookmarking capability**
    - As a user, I want to bookmark my place in a work so that I can return to where I left off
    - Acceptance criteria: Bookmark button that saves current location (using local storage)
3. **Create cross-reference system**
    - As a user, I want to see references to related works or passages so that I can explore connections
    - Acceptance criteria: Links within texts to related passages or works
4. **Build "About" and "How to Use" pages**
    - As a new visitor, I want to understand the site's purpose and how to navigate it so that I can use it effectively
    - Acceptance criteria: About page with mission, credits, and usage instructions

Epic 8: Testing & Launch Preparation

Goal: Ensure quality and prepare for public release

Stories:

1. **Conduct cross-browser testing**
   - As a developer, I need to verify the site works in all major browsers so that all users can access it
   - Acceptance criteria: Site tested in Chrome, Firefox, Safari, and Edge
2. **Perform accessibility audit**
   - As a developer, I need to ensure the site is accessible so that users with disabilities can use it
   - Acceptance criteria: WCAG 2.1 AA compliance verified with proper headings, alt text, keyboard navigation
3. **Optimize performance**
   - As a user, I want pages to load quickly so that I have a smooth experience
   - Acceptance criteria: Images optimized, CSS/JS minified, page load times under 3 seconds
4. **Create deployment plan**
   - As a developer, I need a hosting solution and deployment process so that the site can go live
   - Acceptance criteria: Domain selected, hosting configured, deployment checklist created

---

Suggested Development Order

1. Start with **Epic 1** (foundation)
2. Build **Epic 2** (home page) with sample data
3. Create **Epic 3** (author page) for 2-3 test fathers
4. Develop **Epic 4** (work display) with sample works
5. Implement **Epic 5** (navigation) across existing pages
6. Execute **Epic 6** (content integration) systematically
7. Add **Epic 7** features (enhancements) based on priority
8. Complete **Epic 8** (testing and launch)

This structure allows you to build incrementally and have a working prototype early, which you can then expand with actual content and refinements.