

PART III — Website PRD

Elivate Book Club — Product Requirements Document

1. Overview

Build a modern, fast, secure, and editorial-grade website that serves as the public home for Elivate Book Club. The platform should communicate cultural authority, integrity, and long-term vision from the first interaction. It must support curated book discovery, author engagement, editorial content, and community-facing features with minimal operational overhead.

2. Goals

- Establish a credible literary platform that attracts both serious readers and serious authors.
 - Provide a seamless pipeline for book submissions, editorial review, and member engagement.
 - Create a scalable structure that can evolve into a multi-branch ecosystem (publishing, events, AI tools).
 - Ensure the website becomes the authoritative source for all Elivate Book Club information and activity.
-

3. Target Users

Readers

Need a clear way to discover books, follow the reading cycle, join discussions, and consume editorial output.

Authors

Need a trustworthy environment to submit work, reach a genuine audience, and gain visibility through ethical curation.

Admins

Need a single-login interface to update content, manage posts, oversee book cycles, and adjust events.

Moderators

Need tools to review discussion content, enforce standards, and manage reported activity.

Visitors

Need fast clarity on what Elivate stands for and why they should join.

4. User Stories

- As a **reader**, I want to browse curated book lists and join discussions without friction.
 - As an **author**, I want to submit my book and understand the review process clearly.
 - As a **moderator**, I want to approve posts, maintain standards, and guide community culture.
 - As a **visitor**, I want a clear understanding of Elivate's mission before deciding to commit.
 - As an **admin**, I want to update content quickly without navigating complex systems.
-

5. Key Features

A. Homepage

- Featured Book of the Month
- Brand mission section
- Highlights of upcoming events or author features
- Clear CTAs: Join, Explore, Read More

B. Book Voting / Nomination Tool

- Monthly poll with pre-curated options
- Member nomination system
- Admin panel for selection and review cycles

C. Discussion System

- Threaded conversations per book and per event
- Nested replies
- Moderation controls and content flags
- Clean, distraction-free reading experience

D. Event Calendar

- Public-facing events listing
- Event detail pages (description, date, host, location/virtual link)
- No backend booking system required

E. Author Portal (Public Page)

- Submission guidelines
- Explanation of ethical selection process
- Benefits of participation
- Highlight of featured authors

F. CMS

- Full editorial system for blogs, essays, interviews, and dispatches
- Quick-edit capability
- SEO metadata controls
- Draft and publish workflows
- Scalable structure for long-term growth

G. Newsletter Integration

- Simple description of how newsletters work
- Frontend signup form integrated with your external system
- No internal mailing backend needed

H. Analytics

- User behavior tracking
- Book engagement metrics
- Traffic sources
- Retention indicators

I. Accessibility

- WCAG 2.1 AA requirements
 - Keyboard navigability
 - Color contrast compliance
 - Alt-text enforced in the CMS
-

6. Design & UX Requirements

Visual Style

- Literary-modern aesthetic
- Strong typography hierarchy
- Black-and-white base with a defined accent color for clarity and brand consistency

UX Principles

- Simplicity
- Immediate context and clarity
- Mobile-first workflows

- Optimized navigation focused on discovery, reading cycle, and joining

Typography

- Serif for reading content (classic, literary)
 - Sans-serif for interface elements (modern, clean)
 - Balanced whitespace for readability
-

7. Technical Requirements

Framework

Next.js or Astro

- Both are optimal for performance and SEO.
- Next.js offers more built-in ecosystem support, making it the more practical choice.

CMS

Sanity or Contentful

- Sanity recommended for flexibility, developer-first workflow, and customizable schemas.
- Contentful if you prefer a more enterprise-structured environment.

Hosting

Vercel

- Fast, stable, global edge network
- Perfect fit with Next.js
- Built-in CI/CD

Database

PostgreSQL (with Prisma ORM if using Node)

Authentication

Single admin login only.

Clerk/Auth0/Firebase are unnecessary.

A simple password-protected admin route or minimal auth provider is sufficient.

Performance

- Static rendering where possible
 - CDN caching
 - Pre-rendered pages for books, posts, and story content
-

8. Testing & QA

Testing phase intentionally minimized due to the time-sensitive nature of the website.

Basic smoke tests for:

- Page loads
- CMS publishing
- Event entries
- Discussion flow

No extended QA cycles required.

9. Launch Plan

Direct launch.

No soft launch or staged rollout required.