

BOOKMARK PLATFORM

BOOKMARK ROADMAP CHECKLIST

BOOKMARK is the July 2026 platform roadmap for a standalone digital comics and prose product built to compete with legacy storefronts while returning **85%** of each sale to the author.

FaithEngine: 100-179

Bookmark: 200-279

Revenue Split: 85/15

Policy: Human-Anchored

First-50 Complete: 40

PLATFORM

CATALOG

READER

CREATOR

SUPPORT

PDF

BLUEPRINT

Date: July 2026

Mission: Build Bookmark as a standalone creator-first storefront and reading platform, with FaithEngine powering the internal publishing stack.

Architecture: C# and Python distributed microservices for panel segmentation, multi-language translation, and Stripe Connect payouts.

Target: 24 highly granular system components across ingestion, rendering, localization, synchronization, ethics, and revenue operations.

HUMAN-ANCHORED POLICY

Performance Anchor: Human performance remains mandatory for core character narratives.

Automation Dividend: AI-driven production savings are reinvested into human talent.

Verified HITL: High-impact narratives require verifiable human-in-the-loop certification.

Operational Balance: Automation is default, but visual manual override remains mandatory for final presentation control.

STRATEGIC CONCLUSION

Monetization: No mandatory creator subscription. Default model is an 85/15 transaction split in favor of the author.

Localization: Claude 3.5 Sonnet plus retrieval-backed story bibles preserve tone, idiom, and suspense framing.

IMMEDIATE TIMELINE

Prototype path: Software scaling runs alongside proprietary comic and novel drafting.

Execution order: Use founder-authored assets to produce the first working motion-comic prototype before external studio expansion.

200-219 SITE

200. **200** Lock the public Bookmark narrative around comics, prose, and game lore on one transaction-driven platform.
201. **201** Define the site architecture for landing, catalog, title, creator, roadmap, and support pages.
202. **202** Publish the roadmap page and downloadable PDF tied to the sealed blueprint.
203. **203** Build the release shelf with status, reading-order, and monetization controls.
204. **204** Add title pages with cover, metadata, preview assets, and reading-order context.
205. **205** Add creator and Human-Anchored production-note surfaces.
206. **206** Add searchable release indexing.

- 207. **207** Add responsive media support for covers, previews, trailers, and downloads.
- 208. **208** Add preorder or purchase-intent capture tied to a real workflow.
- 209. **209** Add consistent availability states across the catalog.

220-239 APP

- 220. **220** Define whether Bookmark is a reader app, creator app, or hybrid.
- 221. **221** Lock the app shell for library, current reading, downloads, account, and settings.
- 222. **222** Add authentication only if cross-device state truly requires it.
- 223. **223** Add a library view with owned, followed, in-progress, and completed states.
- 224. **224** Add a release viewer with cover, interior pages, and issue metadata.
- 225. **225** Add reading-position memory, bookmarks, and resume behavior.
- 226. **226** Add notes or highlights only if they support the core user.
- 227. **227** Add offline download and sync rules.
- 228. **228** Add release notifications for followed titles.
- 229. **229** Add purchase, preorder, or entitlement handling tied to the catalog contract and 85/15 logic.

240-259 WORKSPACE

- 240. **240** Create Bookmark as a first-class built-in FaithEngine workspace.
- 241. **241** Define project metadata for titles, volumes, acts, releases, and publishing state.
- 242. **242** Add a workspace dashboard with active titles and blockers.
- 243. **243** Add a title editor for covers, summaries, contributors, and sequencing.
- 244. **244** Add asset lanes for covers, preview pages, PDFs, and promo art.
- 245. **245** Add preview generation for site cards and title pages.
- 246. **246** Add PDF export for the public roadmap and title packets.
- 247. **247** Add publishing actions that push approved assets into the website output safely.

248. **248** Add status gates to block incomplete releases.

249. **249** Add draft, review, scheduled, live, and archived states.

260-279 SERVICES

260. **260** Define the distributed service boundary between C#, Python, and FaithEngine orchestration.

261. **261** Add asynchronous raster extraction for CBZ, CBR, and layered archive ingestion.

262. **262** Add computer-vision panel segmentation with JSON structural output.

263. **263** Add prose document ingestion for DOCX, PDF, EPUB, and long-form manuscript parsing.

264. **264** Add continuity mapping metadata on PostgreSQL JSONB.

265. **265** Add the cinematic frame-by-frame player.

266. **266** Add the adaptive multi-layout manager.

267. **267** Add the reflowable text presentation core.

268. **268** Add the universal real-time state synchronizer.

269. **269** Add the dynamic script-to-frame coordinator.

270. **270** Add contextual localization services with retrieval grounding and glossary controls.

271. **271** Add Stripe Connect payout flows for the 85/15 split.

272. **272** Add payout accounting, audit trails, and settlement recovery paths.

273. **273** Add a Human-Anchored policy engine.

274. **274** Add verified HITL certification workflows for high-impact narratives.

275. **275** Add automation-dividend reporting tied to human talent reinvestment.

276. **276** Add sustainable-employment matching workflows.

277. **277** Add manual visual override wherever automation affects publishable output.

278. **278** Add observability, failure handling, and replay tooling for service jobs.

279. **279** Freeze the Blueprint contract only after all layers validate together.

