Kyomei-MVP-PRD

Kyomei MVP: Product Requirements Document

Version: 1.0

Date: January 8, 2026

Status: Planning Phase

Target Launch: Q2 2026 (6 weeks)

Executive Summary

Kyomei (共鳴 — resonance) is a personalized anime recommendation engine built by fans for fans. Instead of endless scrolling through 10,000+ titles, Kyomei matches users with anime that *resonates with their vibe*—their personal frequency of taste.

The MVP validates the core hypothesis: **A smart recommendation system based on user taste profiles + community data surfaces better anime than random browsing.**

Target User: Anime fans (beginner to veteran) aged 13-35 who want discovery without the guessing game.

Success Metric: User completes vibe check → receives relevant recommendations → rates shows → sees improved recommendations on next visit.

Problem Statement

Current State:
- AniList has over 10000 anime titles with no intelligent personalization
- Users spend 45+ minutes scrolling before picking a show
- Generic “Top 100” lists don’t account for individual taste
- Discovery feels like gambling, the first episode is mad boring

What Kyomei Solves:
- Smart matching based on *personal taste profile*
- Transparent recommendations ("recommended because you like psychological themes")
- Continuously improving algorithm based on user ratings
- Community-informed curation (trending among similar users)

Product Vision (12-Month)

MVP (Months 1-2) → Phase 2 (Months 3-4) → Phase 3 (Months 5-12)
Discovery Loop → Community Features → Advanced ML + Scaling
Solo recommendation → User insights → Personalized rankings
Basic ratings → Social recommendations → Mobile app + integrations

Long-term: Become the trusted discovery layer for anime, powering how millions find their next favorite show.

MVP Scope: 3 Phases

PHASE 1: Foundation (MUST HAVE)

Core loop: Signup → Vibe Check → Recommendations → Rate → Better Recs

1.1 User Authentication & Account Management

Feature	Description	Priority
Sign Up	Email + password registration via BetterAuth	P0
Login	Session-based auth with secure cookie management	P0
Logout	Clear session + redirect to home	P0
User Profile	Store email, name, created_at, updated_at	P0
Password Reset	Email-based reset flow	P1

Success Criteria:
- User can register with unique email

- Sessions persist across page refreshes
- Auth errors are clear (“Email already in use”)

Technical Notes:
- BetterAuth handles JWT + session logic
- PostgreSQL stores user records
- No social login in MVP (add in Phase 2)

1.2 Vibe Check Onboarding Survey

Feature	Description	Priority
5-7 Question Form	Captures user taste profile	P0
Multi-Select Genres	User selects appealing genres (Action, Romance, Comedy, Psychological, Slice-of-Life, Thriller, Drama, Horror, Fantasy, Sci-Fi)	P0
Mood Preference	Single select: Uplifting, Intense/Dark, Relaxing, Thought-Provoking	P0
Episode Length	Short (<13), Medium (13-26), Long (26+)	P0
Animation Style	Hand-drawn, CGI, Experimental	P1
Story Complexity	Light/Fun, Moderate, Deep/Philosophical	P1
Maturity Level	SFW, PG-13, Mature	P1
Form Validation	At least 1 genre + mood required	P0
Save to Database	Store preferences in UserPreferences table	P0

Success Criteria:
- Form completes in <2 minutes
- All answers persist to database
- User can re-edit preferences anytime
- Form errors are helpful (“Select at least one genre”)

Technical Notes:
- Prefer client-side form validation (React) + server-side verification

- Map answers to recommendation algorithm (tag matching)
- Experiment with 5 vs 7 questions later (A/B test post-MVP)

1.3 Anime Catalog (Seeded Data)

Feature	Description	Priority
~300 Popular Anime	Curated seed list (popular + diverse genres)	P0
Core Metadata	Title, description, genres, tags, episode count, rating, year, poster URL	P0
Genre/Tag Tagging	Each anime tagged with 2-5 genres + optional custom tags	P0
Community Rating	Average user rating (5-star scale)	P1
Status Field	FINISHED, AIRING, UPCOMING	P1

Success Criteria:
- Database has 300+ anime with complete metadata
- No missing genres or descriptions
- All poster URLs are valid (no 404s)

Technical Notes:
- Seed from AniList API (public data) or manual CSV import
- Genres/tags normalize in database (separate table or JSON)
- Rating field updates as users rate (aggregate)

1.4 Content-Based Recommendation Engine

Feature	Description	Priority
Tag Matching Algorithm	Surface anime where genres overlap with user preferences	P0
Rating Boost	Prioritize highly-rated anime (4.5+ stars)	P0
Ranking by Relevance	Return top 10-20 recommendations sorted by match score	P0
Exclude Watched	Don’t recommend anime user already rated	P0

Feature	Description	Priority
Cold Start	First-time users get top-rated anime in preferred genres	P0

Algorithm (Pseudo-code):

```
For each unrated anime:
  1. Calculate genre overlap score (0-1, based on shared tags)
  2. Apply rating multiplier (high-rated shows scored higher)
  3. Penalize if user rated similar show poorly
  4. Sort by final score
  5. Return top 10 results
```

Success Criteria:
- Recommendation query returns <200ms
- User sees at least 3 recommendations they recognize
- Recommendations feel relevant (manually validate)

Technical Notes:
- Implement as SQL query + scoring logic (no ML needed for MVP)
- Cache recommendations for 24 hours (TanStack Query)
- Measure: CTR (click-through rate) on recommendations

1.5 Rating System

Feature	Description	Priority
5-Star Rating	User rates anime (1-5 stars, or skip)	P0
Rate from Anywhere	Rate on detail page, recommendation card, or watchlist	P0
Update Rating	User can change rating anytime	P0
Remove Rating	User can delete rating (resets recommendation)	P1
Timestamp Rating	Store when user rated (ratedAt field)	P0

Success Criteria:
- User can rate any anime in <1 second (quick interaction)

- Recommendations update after each rating (refresh on next visit)
- Ratings persist and are visible on user's profile

Technical Notes:
- Store in UserRatings table (userId, animeId, rating, ratedAt)
- Update recommendation cache after new rating
- No public rating display until Phase 2

1.6 Watchlist / Status Tracking

Feature	Description	Priority
Status Categories	Plan to Watch, Watching, Completed, Dropped	P0
Add to Watchlist	User adds anime to any category from detail page	P0
Update Status	Move anime between categories (e.g., Plan → Watching → Completed)	P0
View Watchlist	Dedicated page showing all categorized anime	P0
Progress Tracking	(Optional MVP) Track episodes watched for "Watching" category	P1
Timestamps	dateAdded, dateStarted, dateCompleted	P1

Success Criteria:
- User can organize anime across 4 categories
- Watchlist is persistent and updated in real-time
- User sees count of shows in each category (e.g., "5 Completed")

Technical Notes:
- Store in UserWatchlist table
- Watchlist is the "action layer" — where users track their queue
- Different from ratings (you can rate without watchlisting, and vice versa)

PHASE 2: Core Engagement (SHOULD HAVE)

Improve recommendation quality + add transparency

2.1 Recommendation Refinement

Feature	Description	Priority
Re-compute on Rating	Algorithm improves as user rates more anime	P1
Pattern Detection	Identify user's dominant preferences (e.g., "user loves psychological shows")	P1
Trending in Your Genre	Surface what's trending among users with similar taste	P1

Success Criteria:
- User's 5th recommendation is noticeably better than 1st
- Algorithm catches patterns (e.g., user who rates psychological shows = prioritize psychological)
- No significant latency increase as user rates more

Technical Notes:
- Update recommendation score weights based on user's rating history
- Implement lightweight collaborative filtering ("users who liked X also liked Y")

2.2 Transparency Layer

Feature	Description	Priority
Why This Rec?	Show user why they got each recommendation	P1
Hover/Click for Details	"Recommended because you like: Psychological + Dark themes"	P1
Tag Highlights	Highlight matching tags between user preferences and anime	P1

Success Criteria:
- User trusts recommendations (they understand the reasoning)
- Click-through rate increases with transparency
- User can adjust preferences if rec reasons don't align with intent

---

2.3 Dashboard & Home Feed

Feature	Description	Priority
Personalized Feed	Show user's top recommendations first	P0

Feature	Description	Priority
Quick Stats	“You’ve rated 12 anime	5 completed
Trending This Week	Top-rated anime overall (community signal)	P1
Continue Watching	Shows currently in “Watching” status	P1
Quick Access	Shortcuts to watchlist, search, profile	P0

Success Criteria:
- Dashboard loads in <1 second
- User sees actionable content above the fold
- Mobile-responsive design

2.4 Search & Browse

Feature	Description	Priority
Full-Text Search	Search anime by title	P0
Filter by Genre	Multi-select genre filters	P1
Filter by Year	Range slider (2010-2025)	P1
Filter by Status	FINISHED, AIRING, UPCOMING	P1
Filter by Rating	Min rating (3.0+, 4.0+, etc.)	P1
Sort Options	By rating, recency, title (A-Z)	P1
Search Results	Display 20 results per page with pagination	P1

Success Criteria:
- Search returns results in <500ms
- Filters are intuitive and stackable
- User can discover shows outside their vibe check preferences

Technical Notes:
- Use PostgreSQL full-text search (ILIKE) for MVP (upgrade to Elasticsearch post-MVP)
- Cache filter results for performance

2.5 Anime Detail Page

Feature	Description	Priority
Core Info	Title, poster, rating, genres, episode count, synopsis	P0
User Actions	Rate, add to watchlist, update status	P0
Your Rating	Display user's rating if they've rated it	P1
Community Stats	"87% of users rated this 4+ stars"	P1
Why Recommended?	Explanation of recommendation reasoning	P1
Related Anime	3-5 similar anime based on genres	P1

Success Criteria:
- All info loads in <1 second
- User can rate/watchlist without leaving page
- Mobile layout is readable

2.6 User Profile Page

Feature	Description	Priority
Profile Header	User name, member since, stats	P1
Stats Dashboard	Total rated, completed, dropped, avg rating	P1
Preference Editor	Edit vibe check answers anytime	P0
Rating History	User's all ratings with timestamps	P1
Quick Watchlist Link	Shortcut to their watchlist	P0

Success Criteria:
- User can see their progress over time
- Preference editing refreshes recommendations immediately
- Profile reflects all user activity

PHASE 3: Community & Advanced Features (NICE TO HAVE)

Post-MVP, only if MVP metrics are strong

3.1 Social Recommendations

Feature	Description	Priority
Similar Users	“Users like you also rated X highly”	P2
Community Trending	Most-rated anime this week (filtered by genre)	P2
User Reviews	Short-form ratings + optional text review (future)	P2

3.2 Advanced Recommendation

Feature	Description	Priority
Collaborative Filtering	Cosine similarity between user vectors	P2
A/B Testing Framework	Test multiple recommendation algorithms	P2
Feedback Loop	Track which recs lead to completed watches	P2

3.3 Mobile App

Feature	Description	Priority
React Native App	iOS/Android native experience	P3
Offline Watchlist	Access watchlist without internet	P3
Push Notifications	“New episode of your watching shows!”	P3

3.4 Third-Party Integrations

Feature	Description	Priority
AniList Import	Bulk import user’s AniList ratings	P2
MyAnimeList Import	Similar for MAL users	P2
Streaming Links	“Watch on: Netflix, Crunchyroll, etc.”	P2
Discord Bot	`/anime recommend` command	P3

3.5 Creator Tools

Feature	Description	Priority
Curated Lists	Users create public lists (“Best Slice-of-Life”)	P2
Recommendations from Lists	“Trending in Isekai” created by community	P2

User Stories (MVP Priority)

P0: Core Loop

As a new user
I want to sign up with my email
So that I can create a personalized anime profile

As a new user
I want to complete a quick vibe check (5-7 questions)
So that the system understands my anime taste

As an authenticated user
I want to see recommendations based on my vibe
So that I can find anime that matches my preferences

As an authenticated user
I want to rate anime (1-5 stars)
So that the system learns what I like

As an authenticated user
I want to add anime to my watchlist
So that I can track what I plan to watch, am watching, or have completed

As an authenticated user
I want to see why I'm getting a recommendation
So that I trust the recommendation system

As a returning user
I want to see improved recommendations after rating more anime
So that the system gets better at predicting my taste

P1: Engagement

As a user
I want to search for anime by title or genre
So that I can explore outside my personalized feed

As a user
I want to see my stats (anime watched, rating distribution)
So that I can track my progress

As a user
I want to edit my vibe check preferences anytime
So that I can refine how recommendations work

As a user
I want to see what's trending this week
So that I can discover new popular shows

P2: Community (Post-MVP)

As a user
I want to see what similar users are watching
So that I discover shows through community signal

As a user
I want to see short reviews from other users
So that I can understand why they liked/disliked a show

Non-Functional Requirements

Requirement	Target	Notes
Performance	<1s page load, <200ms API response	TanStack Query caching essential
Uptime	99.5% (MVP on single server)	Upgrade infrastructure in Phase 3

Requirement	Target	Notes
Database	PostgreSQL, 5 core tables, <10MB initial	Indexes on userId, animeId
Security	HTTPS, BetterAuth sessions, no sensitive data in logs	OWASP compliance
Scalability	100-1000 concurrent users	Upgrade to load balancer + caching in Phase 2
Browser Support	Chrome, Firefox, Safari (last 2 versions)	Mobile-responsive (not app, yet)
Accessibility	WCAG 2.1 AA (contrast, keyboard nav, alt text)	Test with accessibility tools

Success Metrics (MVP Launch)

Primary Metrics

Metric	Target	Rationale
Vibe Check Completion Rate	>80% of sign-ups	If users skip, recommendation fails
Rating Engagement	>5 ratings per user (first week)	Signal that recommendations matter
Recommendation CTR	>30% (users click recommended anime)	Validation of algorithm quality
Return Rate	>40% (users return after 1 week)	App retention signal

Secondary Metrics

Metric	Target	Rationale
Avg Recommendation Score	>3.5/5 (user rating of rec quality)	Post-MVP: ask users “how good was this rec?”
Watchlist Utilization	>60% of users use it	Core feature adoption
Search Usage	<20% of discovery (vs recommendations)	If too high = algo isn’t good enough

Out of Scope (MVP)

❌ **Real-time notifications** — Save for Phase 3

❌ **Image recognition** — Anime detection from screenshots

❌ **Payment/Subscription** — Free forever for MVP

❌ **Mobile app** — Web app only (React)

❌ **Advanced ML** — Collaborative filtering comes in Phase 2

❌ **Third-party integrations** — AniList/MAL sync in Phase 3

❌ **Streaming rights** — Don't embed/verify where to watch

❌ **User-generated content** — No reviews/comments in MVP

❌ **Social features** — Following, messaging, etc.

Technical Architecture (MVP)

Frontend

- React (TypeScript) + TanStack Query
- Tailwind CSS for styling
- tRPC client for type-safe API calls
- Pages: Auth, Onboarding, Dashboard, Anime Detail, Search, Watchlist, Profile

Backend

- Hono (edge-first, lightweight)
- tRPC routers (auth, user, anime, recommendations, ratings, watchlist)
- BetterAuth for session management

Database

- PostgreSQL (cloud-hosted: Railway, Neon, or Supabase)
- Drizzle ORM for schema + migrations
- 5 core tables: Users, UserPreferences, Anime, UserRatings, UserWatchlist

Deployment

- Frontend: Vercel (automatic from GitHub)
- Backend: Railway or Render (simple serverless)
- Database: PostgreSQL cloud provider

Timeline & Milestones

Week	Milestone	Deliverables
Week 1	Schema Design & Setup	PostgreSQL tables, Drizzle migration, monorepo structure
Week 2	Foundation Backend	BetterAuth setup, vibe check API, anime seed data
Week 3	Recommendation Engine	Tag-matching algorithm, ranking logic, tRPC procedures
Week 4	Core Frontend	Dashboard, anime detail, watchlist, search pages
Week 5	Integration & Testing	End-to-end testing, bug fixes, UX polish
Week 6	Launch & Iteration	MVP launch, metric tracking, feedback collection

---

Risks & Mitigations

Risk	Impact	Mitigation
Schema design mistakes	High — hard to fix post-deploy	Hand-test schema in PostgreSQL sandbox first
Poor recommendation quality	High — kills user engagement	Manually validate algo on 10+ test users before launch
Performance bottlenecks	Medium — affects user retention	Add database indexes on userId, animeId; use query caching
TypeScript learning curve	Medium — slows development	Block time for framework learning; pair with docs

Risk	Impact	Mitigation
Anime data incomplete	Low — can seed iteratively	Start with top 100, expand later
BetterAuth unfamiliar	Low — good docs available	Read BetterAuth guide before building auth

Definition of Done (MVP)

- [ ] All P0 features implemented and tested
- [ ] Database schema validated with 300+ anime
- [ ] tRPC end-to-end flow works (React → Hono → PostgreSQL → React)
- [ ] Recommendation algorithm manually validated (produces good results)
- [ ] Authentication fully functional (signup, login, logout, sessions)
- [ ] Mobile-responsive design verified
- [ ] TypeScript strict mode passes (no `any` types)
- [ ] 80%+ test coverage on critical paths (auth, recommendations, ratings)
- [ ] Production deployment successful
- [ ] Metrics dashboard set up for tracking

Post-MVP Roadmap (Not MVP)

Phase 2 (Months 3-4): Community & Polish

- Collaborative filtering recommendations
- User reviews / short ratings
- Advanced filters (year, episode length, animation style)
- Trending section
- Preference re-tuning based on feedback

Phase 3 (Months 5-6): Scale & Expand

- Mobile app (React Native)
- Third-party integrations (AniList, MAL import)
- Advanced analytics dashboard
- Creator tools (curated lists)
- Push notifications

Phase 4 (Months 7-12): Monetization & Growth

- Premium features (advanced stats, curated playlists)
- Partnerships with streaming platforms
- Marketing & community growth
- Discord bot, Reddit integration

Glossary

Term	Definition
Vibe Check	5-7 question survey capturing user taste profile
Tag Matching	Algorithm that matches anime genres to user preferences
CTR	Click-through rate (% of users who click recommendation)
Watchlist	User's personalized queue (Plan → Watching → Completed → Dropped)
Collaborative Filtering	Recommendation based on "users like you also liked X"
Content-Based	Recommendation based on anime metadata (genres, tags)
Cold Start	First recommendations for brand-new users (no rating history)
tRPC	Type-safe RPC framework (eliminates REST API contracts)

Questions & Decisions for Review

1. **Anime seed size:** Start with 300 or go bigger? (300 = manageable, 1000+ = overwhelming for MVP)

2. **Rating scale:** 5-star or binary (like/dislike)? (5-star gives more signal)
3. **Vibe check questions:** 5 or 7? (5 = fast, 7 = more signal)
4. **Update recommendations:** On each rating or batch overnight? (On each = real-time feel, batch = cheaper)
5. **Community rating:** Show before MVP launch? (No = less noise, yes = social proof)

Document Owner: @ David Raet

Last Updated: January 8, 2026

Next Review: When MVP alpha launches