Game Hub - Development Roadmap & Future Plans

Version: 1.0.0 **Last Updated:** December 27, 2025 **Current Phase:** Phase 3 (Intelligence)

Table of Contents

Phase Overview

```
Phase 1 (Foundation)     ████████████████████ 100%  ✓ Complete
Phase 2 (Multi-Platform) ████████████████████ 100%  ✓ Complete
Phase 3 (Intelligence)   ███████░░░░░░░░░░░░░  35%  ● In Progress
Phase 4 (Social)         █░░░░░░░░░░░░░░░░░░░   5%  ○ Planned
Phase 5 (Journal)        ░░░░░░░░░░░░░░░░░░░░   0%  ○ Planned
```

Phase	Focus Area	Status
1	Foundation	Complete
2	Multi-Platform	Complete
3	Intelligence	In Progress (35%)
4	Social	Planned (5%)
5	Journal	Planned (0%)

Phase 1: Foundation (Complete)

Goal: Build the core infrastructure for game library management with Steam as the primary platform.

Deliverables

Steam Library Sync

Status: ✅ Complete

- Full Steam Web API integration
- Rate limiting (200 requests/5 minutes)
- Owned games and playtime fetching
- Achievement sync with privacy detection
- Real-time "currently playing" detection

- Auto-completion detection (100% achievements)

Key Files:

- `src/lib/steam/client.ts`
- `src/lib/actions/steam/sync.ts`
- `src/app/api/auth/steam/`

IGDB Metadata Enrichment

Status: ✅ Complete

- OAuth token authentication with Twitch
- 50-day token caching
- Cover art fetching
- Game descriptions
- Developer/publisher information
- Genre categorization
- Platform-specific release dates
- Field locking to preserve user edits

Key Files:

- `src/lib/igdb/client.ts`
- `src/lib/actions/games/enrichment.ts`

Manual Backlog Management

Status: ✅ Complete

- 6 game statuses (unplayed, playing, played, completed, finished, on_hold)
- 3 priority levels (high, medium, low)
- Kanban-style UI
- Collapsible columns
- Search and filter functionality
- Quick "Start Playing" action

Key Files:

- `src/app/(dashboard)/backlog/page.tsx`

Per-Game Notes & Status

Status: ✅ Complete

- Notes field (unlimited text)
- Personal rating (1-10 scale)
- Completion percentage tracking
- Custom tags (up to 10 per game)
- Physical copy indicator
- Hidden game toggle

Key Files:

- `src/components/modals/GameFormModal.tsx`
- `src/lib/actions/games/crud.ts`

Phase 2: Multi-Platform (Complete)

Goal: Expand platform support to PlayStation, Xbox, and Epic Games with unified tracking.

Deliverables

PlayStation Network Sync

Status: ✅ Complete

- NPSSO token authentication
- OAuth flow with auto-refresh
- Trophy fetching (bronze, silver, gold, platinum)
- Playtime data via getUserPlayedGames API
- Profile import (PSN ID, avatar, trophy level)
- Platform normalization (PS5, PS4, PS3, PS Vita, PSP)
- User search for friend comparison

Key Files:

- `src/lib/psn/client.ts`
- `src/lib/actions/psn/`
- `src/components/settings/PsnSettings.tsx`

Xbox Live Integration

Status: ✅ Complete

- OpenXBL API authentication
- Achievement fetching with gamerscore
- Library sync with device tracking
- Profile import (gamertag, avatar, gamerscore)
- Filters PC-only games
- User search by gamertag
- Cross-generation support (360, One, Series X|S)

Key Files:

- `src/lib/xbox/client.ts`
- `src/lib/actions/xbox/`
- `src/components/settings/XboxSettings.tsx`

Epic Games Integration

Status: ✅ Complete

- OAuth authorization code flow
- Library sync with pagination
- Metadata enrichment via IGDB
- DLC/non-game content filtering
- Account linking/unlinking
- Sync logs integration

Note: Epic Games API does not provide playtime or achievement data.

Key Files:

- `src/lib/epic/client.ts`
- `src/lib/actions/epic.ts`
- `src/components/settings/EpicSettings.tsx`

Cross-Platform Friend Comparison

Status: ✅ Complete

- Search users on PSN, Xbox, Steam
- Side-by-side statistics comparison
- Common games detection
- Progress comparison on shared games
- Platform-specific stats display
- Self-comparison prevention

Key Files:

- `src/lib/actions/compare/`
- `src/app/(dashboard)/friends/page.tsx`
- `src/components/compare/`

Unified Achievement Tracking

Status: ✅ Complete

- Cross-platform achievement aggregation
- Per-platform breakdown (Steam, PSN, Xbox)
- Perfect games tracking (100% completion)
- Almost complete games (90-99%)
- Hall of Fame display
- Automatic status updates on 100% completion

Key Files:

- `src/lib/actions/achievements.ts`
- `src/app/(dashboard)/achievements/page.tsx`
- `src/components/achievements/`

Phase 3: Intelligence (In Progress)

Goal: Add smart features for backlog prioritization and data analytics.

Progress: 35% Complete

Completed Features

Priority Queue System

Status: ✅ Complete

- Three priority levels with visual distinction
  - High: "Play Next" (Red)
  - Medium: "When Ready" (Amber)
  - Low: "Someday" (Blue)
- Kanban-style layout
- Collapsible columns

- Quick actions for game management
- Database support (`priority` column)

Custom Tags

Status: ✅ Complete

- Tag array storage (`tags TEXT[]`)
- Tag input/display in game forms
- Filter backlog by tags
- Special 'adult' tag handling
- Up to 10 tags per game

In Progress Features

Custom Lists

Status: 🔶 Schema Only (No UI)

Completed:

- Database tables (`custom_lists` , `list_games`)
- RLS policies implemented
- Sort order support

Needed:

- List management page UI
- Create/edit list modals
- Add games to lists functionality
- List filtering in library
- Public list sharing

Database Schema:

```sql
CREATE TABLE custom_lists (
  id UUID PRIMARY KEY,
  user_id UUID REFERENCES profiles(id),
  name TEXT NOT NULL,
  description TEXT,
  is_public BOOLEAN DEFAULT false,
  created_at TIMESTAMPTZ,
  updated_at TIMESTAMPTZ
);

CREATE TABLE list_games (
  id UUID PRIMARY KEY,
  list_id UUID REFERENCES custom_lists(id),
  game_id UUID REFERENCES games(id),
  sort_order INTEGER DEFAULT 0,
  UNIQUE(list_id, game_id)
);
```

Playtime Analytics

Status: 🔶 Partial Implementation

Completed:

- `game_sessions` table for session tracking
- `getUserStats()` for basic aggregations
- Session recording during Steam sync
- `daily_playtime_summary` view
- Recharts dependency installed

Needed:

- Playtime charts and graphs
- Genre breakdown visualizations
- Platform distribution charts
- Activity heatmaps (GitHub-style)
- Weekly/monthly/yearly trends
- Session history browser

Planned Features

Smart Recommendations

Status: ❌ Not Started

Planned Approach:

1. Analyze user's completion patterns
2. Consider genre preferences
3. Weight by playtime habits
4. Factor in backlog priority
5. Surface "What to play next" suggestions

Implementation Ideas:

- Score games based on:
  - Genre match with completed games
  - Similar playtime to finished games
  - High priority but low completion
  - Recently purchased/added
- Display recommendations on dashboard
- "Quick Pick" feature for indecisive moments

Phase 4: Social (Planned)

Goal: Build social features for gaming communities and friend interactions.

Progress: 5% Complete

Planned Features

Persistent Friend Lists

Status: ❌ Not Started

Planned:

- Save friends from comparison searches

- Track friends across platforms
- Quick-access friend comparison
- Friend activity indicators

Database Design:

```
CREATE TABLE friends (
  id UUID PRIMARY KEY,
  user_id UUID REFERENCES profiles(id),
  friend_platform TEXT NOT NULL,
  friend_platform_id TEXT NOT NULL,
  friend_username TEXT NOT NULL,
  friend_avatar_url TEXT,
  nickname TEXT,
  created_at TIMESTAMPTZ,
  UNIQUE(user_id, friend_platform, friend_platform_id)
);
```

Activity Feeds

Status: ❌ Not Started

Planned:

- Track significant gaming events:
  - Game completion
  - Achievement unlocked
  - New game added
  - 100% completion
  - Started playing
- Display feed on dashboard
- Optional friend activity

Database Design:

```
CREATE TABLE activities (
  id UUID PRIMARY KEY,
  user_id UUID REFERENCES profiles(id),
  activity_type TEXT NOT NULL,
  user_game_id UUID REFERENCES user_games(id),
  metadata JSONB,
  created_at TIMESTAMPTZ
);
```

Leaderboards

Status: ❌ Not Started

Planned:

- Global leaderboards:
  - Most games completed
  - Highest achievement percentage

    - Most playtime
    - Most perfect games
- Friend leaderboards
- Opt-in/opt-out privacy controls

Session Planning

Status: ❌ Not Started

Planned:

- Schedule gaming sessions
- Invite friends
- Multiplayer game matching
- Calendar integration
- Reminders

Release Calendar

Status: 🔶 Data Available

Existing:

- `release_date` field in games table
- IGDB provides release date data

Needed:

- Calendar UI component
- Upcoming releases view
- Wishlist integration
- Release notifications

Price Tracking

Status: ❌ Not Started

Planned:

- Track prices across stores
- Price history charts
- Sale alerts
- Price drop notifications
- Wishlist price monitoring

Potential Data Sources:

- Steam Store API
- IsThereAnyDeal API
- PlayStation Store API
- Xbox Store API

Phase 5: Journal (Planned)

Goal: Create a personal gaming journal with rich media support.

Progress: 0% Complete

Planned Features

Screenshot Aggregation

Status: ❌ Not Started

Planned:

- Import screenshots from Steam
- Manual screenshot upload
- Organize by game
- Gallery view
- Tag screenshots

Technical Considerations:

- Storage: Supabase Storage or external CDN
- Compression for thumbnails
- Lazy loading for galleries
- Max file size limits

Gaming Journal

Status: ❌ Not Started

Planned:

- Per-game journal entries
- Rich text editor (Markdown support)
- Session notes
- Thoughts and reviews
- Progress milestones
- Attach screenshots to entries

Database Design:

```sql
CREATE TABLE journal_entries (
  id UUID PRIMARY KEY,
  user_id UUID REFERENCES profiles(id),
  user_game_id UUID REFERENCES user_games(id),
  title TEXT,
  content TEXT NOT NULL,
  entry_type TEXT DEFAULT 'note',
  session_id UUID REFERENCES game_sessions(id),
  created_at TIMESTAMPTZ,
  updated_at TIMESTAMPTZ
);

CREATE TABLE journal_screenshots (
  id UUID PRIMARY KEY,
  journal_entry_id UUID REFERENCES journal_entries(id),
  screenshot_url TEXT NOT NULL,
  caption TEXT,
  sort_order INTEGER DEFAULT 0
);
```

Timeline Views

Status: ❌ Not Started

Planned:

- Visual timeline of gaming history
- Activity markers:
  - Games started
  - Games completed
  - Achievements earned
  - Journal entries
  - Sessions played
- Filter by game, platform, date range
- Export timeline as image/PDF

Data Sources:

- `game_sessions` table
- `user_games.created_at`
- `user_games.completed_at`
- `journal_entries`

Technical Debt & Improvements

Current Technical Debt

Issue	Priority	Description
Xbox Playtime	Low	OpenXBL API doesn't provide playtime data
Epic Achievements	Low	Epic Games has no achievement system
Steam Rate Limits	Medium	Achievement sync limited in comparisons
Stats Page UI	High	Currently placeholder, needs charts
Custom Lists UI	Medium	Schema exists, no frontend

Planned Improvements

Performance Optimizations

- □ Implement virtual scrolling for large libraries
- □ Add Redis caching for API responses
- □ Optimize image loading with blur placeholders
- □ Implement service worker for offline support

Code Quality

- □ Increase test coverage
- □ Add E2E tests with Playwright
- □ Implement error boundary improvements
- □ Add request retry logic for API failures

User Experience

- □ Add onboarding flow for new users
- □ Implement keyboard shortcuts
- □ Add bulk edit operations
- □ Create mobile-optimized views

Accessibility

- □ Complete ARIA label audit
- □ Improve keyboard navigation
- □ Add screen reader announcements
- □ Ensure color contrast compliance

Future Platform Integrations

Planned Integrations

Nintendo (Limited)

Priority: Medium **Challenge:** No official public API

Possible Approaches:

- Manual entry with IGDB enrichment
- Import from Nintendo Account (web scraping, not recommended)
- Third-party services if available

Games Supported:

- Switch, Switch 2
- Wii U
- 3DS/DS
- Older consoles (manual only)

GOG Galaxy

Priority: Low **Status:** Manual entry currently supported

Potential:

- GOG Galaxy 2.0 database integration (local)
- GOG.com API (limited)

EA App

Priority: Low **Status:** Manual entry currently supported

Challenge: No public API for library access

Ubisoft Connect

Priority: Low **Status:** Manual entry currently supported

Challenge: No public API

Battle.net

Priority: Low **Status:** Manual entry currently supported

Potential:

- Blizzard API for game ownership
- Achievement tracking for supported games

Integration Wishlist

Platform	API Available	Status
Nintendo	No	Manual only
GOG	Limited	Manual only
EA App	No	Manual only
Ubisoft	No	Manual only
Battle.net	Yes	Planned
itch.io	Yes	Potential
Humble Bundle	Yes	Potential

Release Timeline

Q1 2026 Goals

- □ Complete Stats Page visualizations
- □ Implement Custom Lists UI
- □ Add Activity Heatmaps
- □ Basic recommendation algorithm

Q2 2026 Goals

- □ Persistent Friend Lists
- □ Activity Feed
- □ Release Calendar UI
- □ Improved mobile experience

Q3-Q4 2026 Goals

- □ Screenshot integration
- □ Gaming Journal
- □ Timeline Views
- □ Price Tracking
- □ Leaderboards

Contributing

Game Hub is open to community contributions. Priority areas:

1. **Stats Page Charts** - Help build the analytics visualizations
2. **Custom Lists UI** - Create the list management interface
3. **Mobile Optimization** - Improve responsive design
4. **Accessibility** - Ensure WCAG compliance
5. **Testing** - Add unit and integration tests

Roadmap last updated December 27, 2025