

TCGManager

TCGManager is a high-performance, cross-platform TCG portfolio tracker and collection management system. Built for speed and reliability, it allows collectors to manage their assets across multiple games like Riftbound, Magic: The Gathering, Lorcana, One Piece, and Yu-Gi-Oh!.

The Stack

This project is built with a focus on **low latency** and **concurrency**:

- **Backend:** Go (Golang) with Gin – Leveraging go routines for high-speed API aggregation and price updates.
 - **Frontend:** React – A lightning-fast, reactive interface for a seamless "digital binder" experience.
 - **Database:** PostgreSQL – The gold standard for reliable, relational card metadata.
 - **Infrastructure:** Nginx – Serving as a robust Load Balancer and Reverse Proxy.
 - **Performance:** k6 and hey – Integrated load testing to ensure stability during market volatility.
-

Key Features

- **Multi-Game Support:** Unified database for all major TCGs.
 - **Live Market Integration:** Displays current estimated market prices for specific card variants.
 - **Status Management:** Tag cards as *Available*, *Reserved*, *Sold*, or *Vaulted*.
 - **Advanced Filtering:** Deep-search by rarity, type, mana cost, or custom tags.
-

Roadmap

Phase 1: Foundation & Infrastructure

- [x] Setup Docker Compose environment (Nginx, Postgres, Go Backend).
- [x] Implement database schema and migrations for the core TCG logic.
- [x] Implement JWT-based user authentication (Register/Login).

Phase 2: Core Data & Collection

- [x] Seed database with initial game, set, and card data.
- [x] Build basic CRUD API endpoints for searching and viewing cards.
- [x] Implement collection management (add to collection, set status to trade/reserved).
- [x] Develop Frontend UI for collection viewing and filtering.

Phase 3: Polish

- [x] Build Admin Dashboard to monitor components

- [x] Add automated card import from APIs
 - [x] Perform backend load testing and optimize SQL queries.
 - [x] Backend redundancy
 - [x] Integrate mock (or real) market prices for card variants.
-

Getting Started

Prerequisites

- Docker & Docker Compose installed on your machine.

Installation & Run

1. Clone the repository:

```
bash git clone https://github.com/oldonaax/TCGManager.git cd TCGManager
```

2. Configure Environment:

Copy the example environment file and adjust the values if needed:

```
bash cp .env.example .env
```

3. Start the System:

```
bash docker compose up -d --build --scale backend-api=2
```

4. Access the application:

5. Frontend UI: <http://localhost>

6. Backend API (via Nginx): <http://localhost/api>

7. Admin Dashboard: <http://localhost:8081>
-

License

This project is licensed under the **MIT License** – see the [LICENSE](#) file for details.

Built by collectors, powered by Go.