

# AI MTG Assistant – research-driven refinements and feature ideas

## 1. Observations from existing tasks

The user's to-do list already covers core features for an initial **deck-building chatbot** that integrates Scryfall (caching results for autocomplete and pricing), hardened CSV parsing, a **cost-to-finish** tool that compares target decks against a user's collection, basic chat logging, Zod-based error handling, and mobile UI polish. These P0 tasks address reliability and baseline UX.

- **Caching & rate limiting** – Scryfall explicitly recommends staying below ~10 requests per second, including `User-Agent` / `Accept` headers and using bulk data for heavy queries <sup>1</sup>. Tools like the Scryfall MCP server emphasise a 75-ms delay between calls and 24-hour caching, with separate TTLs for card data and prices <sup>2</sup>. Your plan to use Supabase as a 24-hour cache and a debounced API client aligns with this guidance.
- **CSV parsing** – streaming parsers such as `csv-parser` handle ~90k rows per second, support custom headers, trimming and mapping values, and can normalise quotes and whitespace <sup>3</sup>. Using this along with Zod validation will harden import functionality.
- **Price and API restrictions** – Cardmarket and TCGplayer have restricted API access. Cardmarket will release **read-only price and product data** rather than full APIs <sup>4</sup>, and TCGplayer is no longer granting new API keys <sup>5</sup>. This means your assistant should rely on Scryfall (or third-party services like JustTCG) for price data and use affiliate links or manual CSV price snapshots for monetisation.

## 2. Modern tools and competitor features

### 2.1 All-in-one mobile apps

Many mobile collection managers and deck builders (e.g., **ManaBox**, **Dragon Shield's Card Manager**, **Lion's Eye**) offer features that your assistant could emulate:

App/tool	Key features from research
<b>ManaBox</b>	Multi-market price view (CardMarket, TCGplayer, Card Kingdom, Cardhoarder); camera-based card scanning; offline search; collection management; deck builder; deck charts; deck simulator and playtester; news feed; multi-language and multi-currency search; trade-fairness tool; integrated rule book <sup>6</sup> .
<b>Dragon Shield Card Manager</b>	Camera scanner with <b>translation</b> in 11 languages; tracks collection value; social features allowing users to <b>connect with friends, share collections, duplicate decks</b> ; built-in news feed <sup>7</sup> . A recent update adds a <b>"Sold Cards"</b> list which tracks the sale price of cards, calculates gains/losses over time and provides statistics (top performers, earnings rate, etc.) <sup>8</sup> . It also adds new sorting/grouping options by mana value, set, colour, card type, etc. <sup>9</sup> .

App/tool	Key features from research
<b>Lion's Eye (MTG Scanner)</b>	Fast search by name/oracle text/CMC/set/colour identity; dynamic sorting (recently added, rarity, etc.); <b>PRO SCAN</b> for auto-scanning cards into your collection; advanced search using Scryfall syntax; system-wide shortcuts for card images/rulings; sample opening-hand simulator; remote play features to share cards during online games <sup>10</sup> .
<b>EDH Decklist Combo Finder / Commander Spellbook</b>	Allows users to paste a decklist and returns <b>all combos</b> contained in the deck and combos that are only missing one piece. Data is sourced from the Commander Spellbook and is updated daily <sup>11</sup> . The tool categorises combos by effectiveness, type and the number of missing pieces, and suggests alternative cards to optimise a deck <sup>12</sup> .

**Lesson:** players expect scanning, offline search, deck simulation, social sharing and up-to-date combo/price analysis. Matching these capabilities will set a high baseline for your assistant.

## 2.2 AI-driven analysis and meta-awareness

Open source projects demonstrate how machine learning can analyse metagame trends and card interactions:

- **MTG AI Deck Builder** – a research project that pulls Standard-legal cards from Scryfall and uses multiple analysis approaches for **deck archetype classification**. It offers:
  - *Pattern-based analysis* that uses regex patterns to detect mechanics and synergies <sup>13</sup> .
  - *Keyword/statistical analysis* that derives card types, subtypes, keywords and references without assumptions <sup>14</sup> .
  - *Semantic analysis* using pre-trained sentence transformers to generate embeddings for card text and cluster cards/decks based on similarity <sup>15</sup> .
  - *Deck name analysis* to extract archetypes and themes from deck names using TF-IDF and vocabulary matching <sup>16</sup> .
- A `consolidated_meta_analysis.py` script that merges all approaches to provide a unified metagame view <sup>17</sup> .
- **Visual support for synergy** – a recent academic paper proposes computing **synergy scores** by analysing card effects and co-occurrence networks within elite player decks. Cards are represented as nodes and their synergies as edges; the resulting 3D graph helps users understand synergistic relationships <sup>18</sup> . This shows how interactive visualisations can reduce information overload and make synergy discovery intuitive.
- **Price prediction models** – projects like Tsukalos' MTG price predictor extract card attributes from Scryfall and TCGplayer price history to train a neural network that predicts future prices <sup>19</sup> . An article on Data-Surge further explains how LLMs can regress card text to price, using distilbert to predict card value and emphasising the use of MLflow for model tracking <sup>20</sup> . This suggests that an assistant could eventually offer **trend forecasts** and alerts when a card's price is likely to rise.

## 2.3 AI assistant examples

Existing AI chatbots such as MagicGPT showcase typical capabilities: real-time card insights, deck building advice, and strategic gameplay tips <sup>21</sup>. They handle queries about counter-strategies, metagame updates, and can provide recommendations across formats <sup>22</sup>. Although they offer general guidance, they lack personalised suggestions based on a user's collection or budget.

## 3. Ideas to enrich your assistant

### 3.1 Enhanced card input and collection management

1. **Scan & translate physical cards** – implement a mobile/web scanning feature (e.g., via a React Native or JS library using the device camera). Scan results should be matched using Scryfall's fuzzy search, and the tool should offer **auto-scan** (continuous scanning) similar to Lion's Eye. Provide translation of foreign-language cards by showing the English oracle text and set name. <sup>7</sup>
2. **Advanced collection tracking** – allow users to mark cards as owned, wishlisted, sold or for trade. Keep a separate **"sold cards"** list to track sale price, current market value and profit/loss over time <sup>8</sup>. Include sorting/grouping by mana value, set, acquisition date, sold date, colour, type, etc., mirroring Dragon Shield's new sorting options <sup>9</sup>. Display collection statistics (total value, top cards, distribution by colour/rarity).
3. **Social features** – optional friend connections so users can share decks, collections or wishlists. Let players duplicate friends' decks and collaborate on builds. Provide privacy controls like private collections but public decks. Support comments or up-votes on decklists, similar to Moxfield community features.
4. **Import/export flexibility** – along with CSV, support scanning of decklist images (OCR) and imports from popular platforms (Moxfield, Archidekt, AetherHub). For exports, provide JSON, CSV and text outputs and shareable links.
5. **Mobile UX considerations** – maintain sticky chat composers and enlarge tap targets, as planned. Include offline mode for collection viewing and deck building like ManaBox, caching data locally with service workers.

### 3.2 Deck analysis, combo detection and metagame awareness

1. **Multi-layered deck analysis** – combine several analysis techniques:
2. **Statistical analysis**: compute mana curve, colour distribution, card type ratios, ramp vs removal counts, and generate charts.
3. **Rule-based pattern matching**: detect known combos (e.g., Kiki-Jiki + Pestermite) and synergies using regex patterns on oracle text (inspired by `analyze_meta_old_try_to_parse.py`) <sup>23</sup>.
4. **Keyword extraction**: automatically identify mechanics (e.g., "treasure", "flicker") and group cards accordingly <sup>14</sup>.
5. **Semantic similarity**: embed card text using a sentence transformer and cluster similar cards to suggest unconventional synergies <sup>15</sup>.
6. **Combo finder integration** – integrate an EDH combo search similar to Commander Spellbook. When a user provides a decklist, show combos present in the deck and combos missing one

piece, with step-by-step instructions <sup>11</sup> . Provide links to rules/gatherer pages and highlight budget versions of combo pieces.

7. **Synergy visualisation** – build an interactive graph that displays cards as nodes and edges weighted by synergy scores computed via co-occurrence and effect analysis. Use the study's approach of representing card synergies in a 3D graph <sup>18</sup> . Users can explore clusters (e.g., aristocrats package) to understand why cards work together. This can be toggled to a 2-D view for mobile.
8. **Metagame and archetype insights** – scrape top decklists from sources like MTGGoldfish or EDHREC (within terms of service) and classify them into archetypes. Provide a `Meta snapshot` panel showing representation by archetype, color pair, and new cards. Use unsupervised or semi-supervised learning to cluster decks, as in the AI Deck Builder project <sup>24</sup> . Update this snapshot periodically and let users compare their deck against the current meta.
9. **EDH power-level and format legality checks** – implement power-level estimation (similar to edhpowerlevel.com) by evaluating average mana value, presence of combos, tutors and mass removal. Check deck legality across formats using Scryfall and the banlist dataset; highlight illegal cards and suggest replacements.

### 3.3 Personalised AI coaching and multi-agent orchestration

1. **Personas and context-aware suggestions** – leverage the provided `player_personas.json` to tailor recommendations. For example, a **Budget Brewer** persona emphasises low-cost alternatives, commons and uncommons and prioritises fun themes. Adjust queries accordingly (e.g., suggest “Evolving Wilds” instead of fetch lands, “Diabolic Tutor” instead of “Vampiric Tutor”) `[player_personas_dataset]` . Provide toggles for competitive vs casual, tribal vs combo and budget caps.
2. **Role-based AI agents** – adopt a multi-agent framework like CrewAI. For instance:
  3. *Researcher Agent*: fetches up-to-date card rulings, combos and price data.
  4. *Strategist Agent*: analyses deck strength, mana curve and synergy, suggests swaps and strategies.
  5. *Rules Lawyer Agent*: answers rules queries using the comprehensive rule book and identifies interactions that require judge calls.
  6. *Reviewer Agent*: summarises analysis, explains reasoning and suggests improvements in plain language.

CrewAI emphasises role-based agents with tasks and tools <sup>25</sup> , while LlamaIndex's agentic retrieval modes can automatically choose between different data sources for queries <sup>26</sup> . This architecture can orchestrate complex tasks like “build me a budget deck around [[Alela, Artful Provocateur]] using my collection”.

1. **Explainable AI suggestions** – whenever the AI recommends a card, accompany it with a brief explanation (“adds redundancy for infinite mana combo”) and, if appropriate, highlight budget or colour-identity constraints. This aligns with the research suggestion to provide transparent AI rather than black-box results <sup>27</sup> .
2. **Simulation and playtesting** – integrate a **hand simulator** and **virtual playtest** environment similar to ManaBox's deck simulator. Allow users to draw sample hands, shuffle, and simulate the

first few turns. For advanced simulation, consider AI agents that play sample games against standard archetypes.

3. **Natural language & advanced search** – support both Scryfall syntax and plain-language queries. An adaptive search engine can interpret queries like “cheap blue flyers with evasion” or “3-drop ramp spells in green” and convert them into Scryfall search strings. Provide suggestions and highlight filters (mana cost, set, format) similar to Lion’s Eye’s advanced search <sup>28</sup>.
4. **Multi-language support** – localise card names, UI and search terms for major languages (English, Spanish, French, German, Japanese, etc.), leveraging Scryfall’s `multiverse_ids` and translation. Provide card rulings and summarised text in the user’s language.

### 3.4 Pricing and monetisation

1. **Cached price history and alerts** – build a cron job (P2) to download Scryfall’s bulk data nightly to a `cards` table keyed by `oracle_id`. Store daily price snapshots so you can show trends and compute average or median prices over time. Provide **alerts** when the price of a card exceeds user-defined thresholds or shows a spike (use a simple moving average or integrate the price predictor described earlier <sup>19</sup>).
2. **Price predictions** – experiment with simple regression models using card text, rarity and historical prices to forecast short-term price changes (drawing on Data-Surge’s LLM regression example <sup>20</sup>). Even if predictions are approximate, a “trend indicator” (↑, ↓, stable) can guide users.
3. **Affiliate and pro subscription** – continue with the plan for a **Pro** tier where heavy operations (full cost-to-finish, deep simulations, meta analysis) require a subscription. For free users, display ads and provide links to purchase cards via affiliate partners (Cardmarket/TCGplayer if allowed). Add donation links (Patreon/Ko-fi) and optional digital goods (custom card sleeves or tokens) as suggested in your research.
4. **Trade fairness calculator** – incorporate a tool that compares the total value of cards offered by each party (using cached prices) and warns if a trade is lopsided. This is a feature in ManaBox <sup>29</sup>.

### 3.5 Analytics, logging and quality assurance

1. **Event analytics** – build a simple events table or integrate PostHog to track when users open the app, save decks, upload CSVs, compute costs, or submit feedback. Use the data to prioritise features and refine UI. Keep analytics optional to respect privacy.
2. **Chat history and session restoration** – your plan to store chat threads and messages is good. Expose this via a **History** dropdown so users can revisit previous sessions and continue the conversation.
3. **Rule enforcement and moderation** – implement a profanity filter and basic moderation for public comments or chat (shoutbox). Add rate limiting to prevent spam and ensure safety. Use hashed user IDs when storing messages for privacy.

4. **Testing** – expand your Playwright smoke tests to include scanning (simulate file uploads), deck analysis and cost computation. Use automated tests to confirm that caching works and that rate-limiting prevents hitting Scryfall's 429 errors. Add unit tests for CSV parsing (handling quotes, duplicates, blank lines) and for persona adjustments.
5. **Security and RLS** – as planned, ensure all reads/writes use your server routes protected by Supabase row-level security. Provide a “RLS test” route in development to confirm isolation.

## 4. Stretch goals and future work (P2 and beyond)

- **Comprehensive rules engine** – integrate the full Magic Comprehensive Rules into your assistant, enabling detailed questions like “How does *Morph* interact with *Urza, Lord High Artificer*?” Provide step-by-step resolution and link to relevant rules sections.
- **AI opponent for playtesting** – train or fine-tune a lightweight model to play specific archetypes against the user's deck in a sandbox environment. This could start with scripted heuristics (e.g., always play ramp on turn two, counter spells strategically) and eventually incorporate reinforcement learning.
- **Speech interfaces** – once mobile scanning is stable, experiment with voice commands (e.g., “Add two Arcane Signets”, “Show me removal spells under three mana”) to enable hands-free deck building.
- **Tournament helper** – add features for tournament players: pairings and standings integration, life counters with poison and commander damage, round timer, and note-taking. Link to official tournament policies and penalty guides.
- **Alternate format support** – support special formats like Pauper, Oathbreaker, Artisan and Cube drafting. Provide suggestions for building and drafting cubes, including average mana curves, card roles (removal, finisher) and synergy packages.

## 5. Summary

Combining the P0 and P1 tasks with the **modern features** and **AI-driven insights** highlighted above will position your MTG assistant as a comprehensive platform rather than a basic chatbot. Start by solidifying the Scryfall cache, CSV imports and cost-to-finish computation. Then gradually integrate **scanning, deck analysis with combo detection, metagame awareness, personalised AI personas, and social features**. Respect API rate limits, use caching and consider nightly bulk data syncs. Where heavy operations are needed, gate them behind a **Pro tier** while offering basic features for free. Ultimately, the goal is to create a **smart, explainable and community-oriented assistant** that helps players brew, optimise, and enjoy Magic: The Gathering.

---

<sup>1</sup> I'm having trouble accessing the Scryfall API, or I'm blocked · Scryfall Magic The Gathering Search  
<https://scryfall.com/docs/faqs/i-m-having-trouble-accessing-the-scrryfall-api-or-i-m-blocked-17>

<sup>2</sup> Scryfall MCP 서버 | MCP Servers · LobeHub  
<https://lobehub.com/ko/mcp/bmurdock-scrryfall-mcp>

<sup>3</sup> csv-parser - npm  
<https://www.npmjs.com/package/csv-parser>

<sup>4</sup> The State of Cardmarket 2024 | Cardmarket  
<https://www.cardmarket.com/en/Insight/Articles/the-state-of-cardmarket-2024>

- 5 TCGplayer APIs Developer Community and Documentation  
<https://developer.tcgplayer.com/>
- 6 29 ManaBox  
<https://manabox.app/>
- 7 Digital Card Manager for TCG players  
<https://www.dragonshield.com/en-us/card-manager>
- 8 9 New Features Now Available on Card Manager - Dragon Shield  
<https://about.dragonshield.com/gaming-inspiration/new-features-now-available-on-card-manager/>
- 10 28 MTG Scanner - Lion's Eye for Windows Pc & Mac: Free Download (2023) | Pcmacstore.com  
<https://pcmacstore.com/en/software/1546754798/mtg-scanner-lion-s-eye>
- 11 12 EDH Decklist Combo Finder  
<https://combo-finder.com/>
- 13 14 15 16 17 23 24 mtg\_ai\_deck\_builder/README.md at main · georgejeh/mtg\_ai\_deck\_builder · GitHub  
[https://github.com/georgejeh/mtg\\_ai\\_deck\\_builder/blob/main/README.md](https://github.com/georgejeh/mtg_ai_deck_builder/blob/main/README.md)
- 18 27 132569.pdf  
<https://www.scitepress.org/Papers/2025/132569/132569.pdf>
- 19 MTG-price-predictor/README.md at main · Tsukalos/MTG-price-predictor · GitHub  
<https://github.com/Tsukalos/MTG-price-predictor/blob/main/README.md>
- 20 Predicting Magic the Gathering's Card Power Using LLMs | Data Surge | data-surge  
<https://medium.com/data-surge/predicting-magic-and-the-gathering-card-power-from-price-using-llms-92b1bb8dd757>
- 21 22 MagicGPT - Real-Time MTG AI Companion [100% Free, No Login Required]  
<https://www.ai4chat.co/gpt/magicgpt>
- 25 Introduction - CrewAI  
<https://docs.crewai.com/en/introduction>
- 26 RAG is dead, long live agentic retrieval — LlamaIndex - Build Knowledge Assistants over your Enterprise Data  
<https://www.llamaindex.ai/blog/rag-is-dead-long-live-agentic-retrieval>