TrendTrivia - Comprehensive Product Requirements Document

Version: 1.0
Last Updated: May 20, 2025

1. Introduction & Overview

1.1. Document Purpose

This document outlines the overall product vision, goals, target audience, features, and phased development plan for TrendTrivia. It serves as a strategic guide for building an engaging, AI-powered news quiz application.

1.2. Product Name

TrendTrivia

1.3. Product Vision

To be the go-to platform for individuals seeking a fun, engaging, and informative way to stay updated with and test their knowledge on current events and trends across various domains, powered by intelligent content generation.

1.4. Core Problem/Opportunity

Staying informed about rapidly changing news and trends can be time-consuming and often passive. Traditional news consumption lacks engagement and a way to actively test understanding. TrendTrivia aims to transform this by gamifying news consumption, making learning active, enjoyable, and rewarding.

1.5. Overall Product Goals (User-centric)

- **Inform & Educate:** Provide users with a dynamic way to learn about and recall key information from recent news and happenings.
- **Engage & Entertain:** Offer an enjoyable and interactive quiz experience that motivates users to return and test their knowledge regularly.
- **Empower Knowledge:** Help users feel more informed and confident about their understanding of current events in topics they care about.
- **Personalize (Future):** Allow users to tailor their quiz experience to their interests and track their progress over time.

1.6. Target Audience

- Individuals interested in a fun, quick, and accessible way to stay updated and test

their knowledge on current news and trends across various topics (e.g., Finance, Technology, Startups, Entertainment).
- Lifelong learners and trivia enthusiasts.

2. Guiding Principles

- **User-Centric Design:** Prioritize a simple, intuitive, and enjoyable experience for the end-user.
- **AI-Powered Innovation:** Leverage AI to create relevant, timely, and diverse quiz content.
- **Iterative Development (Start Small, Grow Smart):** Begin with a focused MVP and incrementally add value based on learning and user feedback.
- **Quality Content:** Strive for accuracy and relevance in quiz questions and contextual information.
- **Accessibility:** Design for a broad audience, considering accessibility standards.
- **(Developer Context) Practical, Real-World Learning:** Focus on applying concepts in a way that mirrors real product development.
- **(Developer Context) Free & Open Source First:** Prioritize free tiers, open-source software, and low-cost solutions for development and deployment, especially in early stages.
- **(Developer Context) Keep it Simple (KISS):** Avoid unnecessary complexity, especially in initial phases.

3. Product Roadmap & Phased Releases

This roadmap outlines a phased approach to developing TrendTrivia, starting with a foundational MVP and progressively adding features and capabilities.

3.1. Phase 1: MVP1 - Foundation & Core Quiz Experience

- **3.1.1. Goals for MVP1:**
  - Validate the core quiz concept with a single topic (Finance).
  - Establish the foundational technical architecture.
  - Implement basic AI-assisted MCQ generation.
  - Deliver a simple, playable quiz experience with local high score tracking.
- **3.1.2. Key Features for MVP1:**
  - **Topic Focus:** Single topic - "Finance."
  - **AI Content Generation (Assisted):**
    - Manually sourced finance news snippets as input.
    - LLM API calls to generate 1-2 MCQs per snippet, including correct answer,

        plausible distractors, and a brief summary/context.
        - Initial Question Bank stored as a JSON file.
    - **Quiz Interface:**
        - Simple, clean UI for question display, option selection.
        - Display one question at a time.
    - **Gamification:**
        - Timer for quiz/questions (exact logic TBD).
        - Scoring based on correct answers.
        - Immediate feedback on answers (correct/incorrect, show correct answer, show summary).
        - Display final score.
    - **High Score:**
        - Store and display the user's single highest score for the "Finance" topic using browser cookies.
    - **Basic UI:** Landing page, quiz page, score page.
- **3.1.3. Success Metrics for MVP1:**
    - Successful deployment to a public URL (e.g., AIproductPM.com/TrendTrivia).
    - Users can complete a quiz session on "Finance."
    - AI-generated questions are successfully presented and are generally coherent.
    - High score functionality works as expected.
    - Positive qualitative feedback on the core quiz experience (even if from a small test group).
- **3.1.4. Non-Goals for MVP1:**
    - User accounts or authentication.
    - Multiple topics.
    - Automated news sourcing.
    - Large-scale, fully automated daily question generation.
    - Advanced gamification (leaderboards, badges).
    - Mobile-specific native apps.

3.2. Phase 2: MVP2 - Expansion & Engagement

- **3.2.1. Goals for MVP2:**
    - Expand content diversity by introducing more topics and automating news sourcing.
    - Enhance AI content generation capabilities.
    - Improve question bank management.
    - Introduce basic user accounts for persistent score tracking and a more

personalized experience.

- **3.2.2. Key Features for MVP2:**
  - **Multiple Topics:** Introduce 1-2 additional topics (e.g., "Technology," "Startups").
    - Allow users to select a topic for their quiz.
  - **Automated News Sourcing:**
    - Integrate free News APIs to fetch recent news articles/summaries for selected topics.
  - **Enhanced AI Content Generation:**
    - Refine prompts for better quality MCQs and summaries.
    - Potentially increase the number of questions generated from each news item.
    - Explore techniques for more regular updates to the question bank.
  - **Database for Question Bank:**
    - Migrate question storage from JSON file to a scalable, free-tier cloud database (e.g., PostgreSQL, MongoDB).
  - **Basic User Accounts:**
    - Simple registration and login (e.g., email/password).
    - Persistent storage of user scores across sessions and devices.
    - Display user's overall high scores per topic.
  - **UI/UX Improvements:** Based on feedback from MVP1.
  - **Time-based Filtering (Exploratory):** Allow users to choose news recency (e.g., "Last Week," "Last Month") for quizzes if feasible with news sources and AI.
- **3.2.3. Success Metrics for MVP2:**
  - Successful integration of multiple topics.
  - Demonstrably improved and more automated question generation pipeline.
  - Users can create accounts and have their scores saved persistently.
  - Increase in user engagement (e.g., number of quizzes played, session duration).

3.3. Future Vision (Beyond MVP2)

- **3.3.1. Potential Future Features & Capabilities:**
  - **Advanced Gamification:**
    - Global and friend-based leaderboards.
    - Badges, achievements, and points systems.
    - Daily challenges or streaks.
  - **Personalization:**

        - User profiles with preferences.
        - Quiz recommendations based on interests or performance.
        - Personalized learning paths or feedback.
    - **Content Expansion:**
        - Wider range of topics (e.g., Entertainment, Science, World News).
        - Different question types (e.g., true/false, fill-in-the-blanks, image-based questions if relevant).
        - User-contributed questions (with moderation).
    - **Community Features:**
        - Ability to challenge friends.
        - Discussion forums or comments related to quiz topics.
    - **Advanced AI & Analytics:**
        - AI to dynamically adjust quiz difficulty.
        - Deeper analysis of news content for more nuanced questions.
        - Analytics on user performance to provide insights.
    - **Monetization (Optional & Long-term):**
        - Premium features (e.g., ad-free experience, exclusive content).
        - Sponsorships for specific topic quizzes.
    - **Mobile Applications:** Native iOS and Android apps for an optimized mobile experience.
    - **Admin Dashboard:** Tools for content management, user management, and analytics.

4. Detailed Feature Specifications (Conceptual - will be detailed per phase)

4.1. Core Quiz Engine

- **Functionality:** Handles question fetching, display logic, answer processing, scoring, timer management, and progression through the quiz.
- **Key Aspects:** Randomization of questions, clear presentation of questions and options, immediate feedback mechanisms.

4.2. AI-Powered Content Generation

- **Functionality:** System for sourcing news/information and using LLMs to generate MCQs, correct answers, distractors, and contextual summaries.
- **Evolution:**
    - MVP1: Manual news snippets, basic LLM prompting, JSON storage.
    - MVP2: Automated news API integration, refined prompting, database storage.

    - Future: Advanced news analysis, diverse question types, dynamic difficulty.

4.3. Gamification Elements

- **Functionality:** Features designed to increase user engagement and enjoyment.
- **Evolution:**
    - MVP1: Timer, basic scoring, local high score (cookies).
    - MVP2: Persistent scores with user accounts.
    - Future: Leaderboards, badges, points, challenges.

4.4. User Interface & Experience (UI/UX)

- **Principles:** Clean, intuitive, responsive, and visually appealing.
- **Key Components:** Landing page, topic selection (MVP2+), quiz interface, score display, user profile (MVP2+).

4.5. User Account Management (MVP2+)

- **Functionality:** Registration, login, profile management, secure storage of user data and scores.

4.6. Content Management (Future)

- **Functionality:** Tools for managing topics, reviewing/editing AI-generated questions, monitoring content quality.

5. AI Integration Strategy

5.1. MVP1 Approach

- Focus on validating the core concept of LLM-based MCQ generation from manually selected finance news snippets.
- Prioritize simplicity and feasibility with free-tier LLM APIs.
- Store generated questions in a local JSON file.

5.2. Evolution & Scalability (Post-MVP1)

- Integrate news APIs for automated content sourcing across multiple topics.
- Develop more sophisticated prompt engineering techniques for higher quality and variety of questions.
- Implement a robust pipeline for regular (e.g., daily) updates to a database-backed question bank.
- Explore different LLMs or fine-tuning (if feasible and cost-effective in later stages) for specialized content generation.
- Monitor AI model performance and implement mechanisms for quality control.

6. Technology Stack Considerations (Overall & Evolving)

The specific technologies will be chosen based on the guiding principles, particularly focusing on free tiers, open-source options, and learning value (for the developer). The stack will be discussed and decided upon collaboratively for each phase.

- **Frontend:**
  - Options: React, Vue.js, Svelte, or Plain HTML/CSS/JavaScript.
  - Considerations: Learning curve, community, performance, suitability for interactive UIs.
- **Backend:**
  - Options: Python (Flask or FastAPI), Node.js (Express.js).
  - Considerations: Ease of AI/News API integration, scalability, developer familiarity.
- **AI Model API (LLM):**
  - Options: Google Gemini, Hugging Face Inference API, other platforms with accessible free tiers.
  - Considerations: Free tier limits, API capabilities, ease of use, quality of output.
- **News API:**
  - Options: NewsAPI.org, Finnhub.io, GNews, etc. (focus on free tiers with relevant content).
  - Considerations: Data relevance, update frequency, API limits, terms of service.
- **Database:**
  - MVP1: JSON file.
  - MVP2+: Cloud-based free tiers like Supabase (PostgreSQL), Firebase Firestore (NoSQL), MongoDB Atlas (NoSQL).
  - Considerations: Scalability, query needs, ease of use, free tier limitations.
- **Deployment Platform:**
  - Options: Vercel, Netlify, Heroku (check current free tier), Google Cloud Run/App Engine, AWS EC2 free tier.
  - Considerations: Ease of deployment, cost, support for chosen stack.
- **Version Control:** Git & GitHub.

7. Open Questions & Areas for Future Exploration

- Optimal free-tier LLM and News APIs for sustained use.
- Specifics of timer logic (per question vs. per quiz).
- Detailed schema for the question bank (JSON and future database).

- Comprehensive error handling strategies for external API calls.
- Security best practices for API key management and user data (especially with user accounts).
- Strategies for ensuring the factual accuracy and reducing bias in AI-generated content.
- User feedback mechanisms for continuous improvement.
- Scalability considerations for handling more users, topics, and questions.