Requirements Document

Prompt Battle WebGame

Introduction

This document defines the functional and quality requirements for the Prompt Battle WebGame. The project's purpose is to create a lightweight but educational web-based game where players compete by writing prompts for a given AI-generated image. Their submissions are compared to the hidden original prompt, and scores are calculated based on similarity.

The requirements serve as the foundation for design, realisation, and validation, ensuring the game can be delivered within the semester scope while remaining testable, transparent, and aligned with the learning outcomes.

Scope

The scope of this iteration is a minimal viable version of Prompt Battle that supports both private-room play and an optional asynchronous daily challenge. It includes the full game loop: lobby, round setup, prompt drafting with a character limit, scoring pipeline, and results reveal with explanation. Out of scope for this version are matchmaking, monetisation, and advanced social features.

Functional Requirements

Game Flow

- **FR1. Room Management**
    - Users can create a private room and receive a join code.
    - Users can join an existing room using a code.
    - The host can start rounds once all participants are ready.
- **FR2. Daily Challenge**
    - Users can access an asynchronous "daily challenge" without creating a room.
    - All submissions in daily mode are compared against the same prompt.
- **FR3. Round Setup**
    - Each round presents one image with a hidden source prompt.
    - A countdown timer and character limit are shown to all players.
- **FR4. Prompt Submission**
    - Players can enter a prompt up to the character limit.
    - A live character counter is displayed.
    - Submissions lock automatically at time zero.

- **FR5. Scoring**
    - All submitted prompts are collected server-side.
    - The system compares each prompt against the hidden original prompt.
    - A similarity score from 0–100 is calculated for each player.
    - Rankings are generated based on score.
- **FR6. Results Reveal**
    - After scoring, the original prompt is revealed.
    - A results screen displays each player's prompt, score, and highlight of matching/missing entities.
    - The leaderboard shows ranking for the round.
- **FR7. Feedback and Data Persistence**
    - Players can give quick feedback on the fairness of the score.
    - Results are saved for later analysis and validation.

Quality Requirements

- **QR1. Usability**
    - Interface is clear and responsive on both desktop and mobile browsers.
    - Character counter and timer are visible at all times during drafting.
    - Results screen highlights matched and missed elements in an understandable way.
- **QR2. Transparency**
    - Scoring method is documented and shown to users.
    - Players see which words/entities influenced their score.
- **QR3. Performance**
    - Submissions are processed within 2 seconds for up to 8 players.
    - Rounds complete without server lag or delays.
- **QR4. Reliability**
    - Disconnected players can rejoin within a grace period.
    - If a player does not rejoin, their round is marked as forfeit.
- **QR5. Security**
    - Original prompts are never revealed before the results stage.
    - Rooms are protected by unique join codes.
    - User input is filtered for harmful or banned terms.
- **QR6. Privacy**
    - Only temporary nicknames are required to join.
    - No personal data is stored unless explicitly created by a teacher for classroom sessions.

MoSCoW Prioritisation

Must Have

- Room creation and join via code.
- Display of image and hidden source prompt per round.
- Character limit and countdown for drafting.
- Prompt submission with auto-lock at timer end.
- Server-side scoring and ranking.
- Results reveal with original prompt and highlight of overlaps.
- Basic feedback and persistence of round results.

Should Have

- Daily challenge mode.
- Export of round summary as text/CSV for analysis.
- Anti-cheat checks (no early reveal, no duplicate copy of original prompt).

Could Have

- Embedding similarity as a tiebreaker.
- Public leaderboard for daily challenge.
- Teacher mode with restricted environment (classroom sessions).

Will Not Have (in this iteration)

- Public matchmaking.
- Friends and chat system.
- Monetisation (ads, subscriptions).
- Native mobile apps.

Validation Approach

Each requirement will be validated through:

- Unit testing (for scoring and token matching).
- End-to-end testing (for full game flow).
- Usability testing (pilot sessions with classmates).
- Feedback surveys (player trust in scoring fairness).