Project Requirements Document

1. Project Overview

Project Name: Grammys Prediction Market MVP

Objective: Develop a web application that allows users to place bets on Grammy nominees, withdraw partial stakes, and claim winnings based on the outcomes. The app will simulate a prediction market for the Grammys awards, focusing on providing an engaging and user-friendly experience.

2. Goals and Objectives

- **User Engagement**: Provide an interactive platform for users to engage with the Grammys by predicting winners.
- **Ease of Use**: Ensure the platform is intuitive, with seamless navigation and clear instructions.
- **Security**: Implement robust authentication and authorization mechanisms to protect user data and transactions.
- **Scalability**: Build the application with future growth in mind, allowing for easy addition of new features and markets.
- **Compliance**: Adhere to legal and regulatory requirements related to running a prediction market.

3. Functional Requirements

3.1 User Accounts

- Users can sign up and log in using Google OAuth
- Users have a profile displaying their balance, bets, and transaction history.

3.2 Market Management

- Display a list of active and upcoming markets (e.g., different award categories).
- Each market has a title, description, expiration date, and a list of outcomes (nominees).

3.3 Betting

- Users can place bets on any active market before its expiration date.
- Users can specify the amount they want to bet on a particular nominee.
- Users can view their current stakes on each outcome.

3.4 Stake Management

- Users can withdraw a partial amount from their stake on any outcome before the market expires.

- Users can view their remaining stakes after withdrawal.

3.5 Market Resolution

- Admins can resolve markets by selecting the winning nominees after the Grammys results are announced.
- The system calculates winnings based on the total pool and individual stakes.

3.6 Withdrawals

- Users can withdraw their winnings after a market is resolved.
- If no winning bets exist, users can claim a refund of their stakes.

3.7 Admin Functions v2

- Admins can create new markets and add outcomes (nominees).
- Admins can resolve markets and manage allowed callers (if in restricted mode).

4. Non-Functional Requirements

4.1 Performance

- The app should load pages within 2 seconds under normal load.
- Database queries should be optimized for quick response times.

4.2 Security

- Use HTTPS for all communications.
- Securely handle user authentication tokens.
- Protect against common web vulnerabilities (e.g., SQL injection, XSS).

4.3 Usability

- The UI should be intuitive and accessible.
- Provide clear error messages and guidance for user actions.

4.4 Scalability

- Design the architecture to handle an increasing number of users and markets.
- Implement pagination or lazy loading for lists with many items.

4.5 Maintainability

- Code should be well-documented and follow best practices.
- Use a consistent coding style and structure.

5. Constraints and Assumptions

- The MVP will focus on a single event (the Grammys) and may expand later.
- Users' balances are simulated and do not involve real money transactions.
- The app will not handle real currency but will use virtual tokens for betting.

6. Risks and Mitigations

- **Legal Risks**: Running a prediction market may have legal implications.
  - *Mitigation*: Consult legal counsel to ensure compliance; consider offering the app as a simulation without real money.
- **Security Risks**: Potential for unauthorized access or data breaches.
  - *Mitigation*: Implement robust authentication, authorization, and regular security audits.
- **Performance Risks**: High traffic during peak times may affect performance.
  - *Mitigation*: Optimize code, use caching, and consider scalable hosting solutions.