Testing Plan

JMMPlatform

Testing Types

1. **Unit Testing**
   1. **Description**: Focus on testing individual components and functions in isolation to ensure they work as expected.
   2. **Tools/Frameworks**: Jest, React Testing Library, Mocha.
2. **Integration Testing**
   1. **Description**: Test how different modules (e.g., frontend and backend) work together, focusing on data flow and communication between services.
   2. **Tools/Frameworks**: Postman
3. **User Acceptance Testing**
   1. **Description**: Validate the platform from the perspective of end-users (creators, fans, beta testers) to ensure it meets business requirements and provides a smooth user experience.
   2. **Participants**: Beta testers group.
   3. **Tools**: Manual testing with a structured checklist, supplemented by surveys for feedback.

Test Cases

- **Test Case 1: User Registration and TikTok Profile Linking**
  - **Feature**: User registration and linking TikTok accounts.
  - **Testing Type**: Unit Test, Integration Test.
  - **Steps**:
    - Fill out the registration form with valid details (username, email, password).
    - Link a TikTok profile using the provided URL.
    - Submit the form.
  - **Expected Outcome**:
    - User account is created successfully.
    - TikTok profile link is saved to the database.

Test Cases

- **Test Case 2: Competition Creation**
    - **Feature**: A creator can create a competition.
    - **Testing Type**: Integration Test, User Acceptance Testing.
    - **Steps**:
        - Log in as a creator.
        - Navigate to the "Create Competition" page.
        - Fill in competition details (title, description, rewards, dates).
        - Submit the competition.
    - **Expected Outcome**:
        - Competition is saved in the database.
        - Confirmation message is displayed.
        - Fans can see the new competition on their dashboards.

Test Cases

- **Test Case 3: Video Submission by Fans**
    - **Feature**: Fans submit videos to competitions.
    - **Testing Type**: Integration Test, User Acceptance Testing.
    - **Steps**:
        - Log in as a fan.
        - Navigate to a competition.
        - Upload a video file and add a title.
        - Submit the entry.
    - **Expected Outcome**:
        - Video is uploaded and stored successfully.
        - Status is set to "Pending" in the database.

Test Cases

- **Test Case 4: Leaderboard Updates**
    - **Feature**: Dynamic leaderboard for competition rankings.
    - **Testing Type**: Unit Test, Integration Test.
    - **Steps**:
        - Interact with the submitted videos (likes, shares, views).
        - Check the leaderboard rankings.
    - **Expected Outcome**:
        - Rankings update in real-time based on engagement metrics.
        - Leaderboard displays accurate data.

Testing Schedule

- **Week 1: Unit Testing**
  - o **Who**: Development Team.
  - o **Focus**: Core functionalities (e.g., registration, competition creation, video submission).
  - o **Time Expected**: 5 days.
- **Week 2: Integration Testing**
  - o **Who**: Development Team.
  - o **Focus**: Backend API and frontend communication, database interactions.
  - o **Time Expected**: 3 days.
- **Week 3: User Acceptance Testing**
  - o **Who**: Beta Testers (selected fan group).
  - o **Focus**: End-to-end testing of the user interface and workflows.
  - o **Time Expected**: 7 days.