Trials and Triumphs – Web Development Brief

This document is intended for the **web development team**. It outlines the **functional requirements, page flow, data handling, and implementation expectations** for the Trials and Triumphs website. Visual styling broadly follows the UI/UX brief, but this document focuses on **how things should work** rather than how they look.

Project Overview

Trials and Triumphs is a fantasy-themed team-based event website with:

- A landing page
- Team registration flow
- Character selection system
- Team status / overview page

The experience should feel **smooth, guided, and game-like**, but implementation should remain **practical and reliable**.

If a full fantasy-styled UI becomes complex to implement, a **minimal dark-themed design with stylized cards is completely acceptable**.

Tech Expectations (Flexible)

- Framework: React / Next.js (preferred but not mandatory)
- Styling:
- Tailwind CSS or equivalent utility-based styling recommended
- Dark theme by default
- State handling:
- Local state initially (can later extend to backend)
- Data source for characters:
- **Google Sheets (mandatory)**

Data Source – Google Sheets

Character Data

All character **stats, skills, and descriptions** must be fetched from a **Google Sheet**.

Expected data per character:

- Character Name
- Class (Knight, Archer, Wizard, Assassin, Bard)
- Base Stats:
- HP
- Mana
- Speed
- Strength / Utility
- Skill list (names + short description)
- Optional: sprite/image URL

  Developers may use the Google Sheets API or a published CSV/JSON endpoint.

Character values **should not be hardcoded**.

Page-by-Page Functional Requirements

---

1. Landing Page

Purpose

- Display the event title
- Establish theme

Functionality

- Static page
- Scroll or navigation button to registration section

Notes

- Background can be:
- Fantasy artwork
- OR plain black/dark gradient

No dynamic data required.

---

2. Team Registration Page

Inputs

- Team Name
- Player 1 Name

- Player 2 Name
- Player 3 Name

Functionality

- Basic form validation (non-empty fields)
- On submit:
- Store team + player data in state
- Redirect to Character Selection page

Backend persistence is **not required initially** unless explicitly added later.

---

3. Character Selection Page

Purpose

- Allow team to select characters after registration

Rules

- Total available characters:
- Knight
- Archer
- Wizard
- Assassin
- Bard
- Each team selects **3 characters**
- No duplicate character selection (unless stated otherwise later)

Functionality

- Fetch character data from Google Sheets on page load
- Display characters as cards:
- Name
- Sprite (if available)
- Short description
- Primary stat buff
- Allow selection with clear visual feedback
- Disable submission until 3 characters are selected

On submit: - Save selected characters to team state - Redirect to Team Status page

---

4. Team Status / Overview Page

Purpose

- Display the team’s final configuration

Display Per Player

For each of the 3 players:

- Assigned character
- Character sprite
- Level (default level = 1 unless specified)
- Stats fetched from Google Sheets
- Skills list

Notes

- Stats must reflect data from the sheet
- No calculations required unless specified later

Navigation Flow

1. Landing Page
2. Team Registration
3. Character Selection
4. Team Status Screen

Navigation should be **linear**, no skipping steps.

Styling Guidelines (Developer-Friendly)

If full fantasy UI is difficult to implement:

- Use **black or very dark background**
- Use **clean, stylized cards** for:
- Characters
- Players
- Simple hover effects, borders, and shadows are enough

Fantasy elements are **nice-to-have**, not blockers.

Responsiveness

- Desktop-first (event setting)
- Basic mobile responsiveness preferred
- Avoid complex animations that affect performance

Out of Scope (For Now)

- Authentication
- Payments
- Admin dashboards
- Match logic or combat calculations

Summary

The web implementation should prioritize:

- Clear flow
- Reliable data fetching from Google Sheets
- Simple, maintainable state management
- Flexibility in styling (fantasy OR minimal dark)

If it works smoothly and looks clean, it’s a win.