

Project Charter

BUDT748 Fall 2024

**PROJECT NAME**      Football Paradise - Stream Management Platform

Specify the official name of the project to be undertaken.

TEAM MEMBERS

List the names and roles of all team members.

ID	NAME	ROLE
1	Ian Delannes-Molka	Business Systems analyst
2	Ilyas Nur	Data Analyst, Software Developer
3	Shruthi	Quality Assurance Analyst
4	Dheeraj Lagudu	IT Systems Analyst
5	Raunak Singh Chhabra	Software Developer
6	Piyali Bedagkar	Scrum Master
7	Rajat Nirwan	PM, UI & UX

BACKGROUND

Provide a brief overview of the project context and rationale.

Why is this project important? What led to the initiation of this project?

This project is an exciting leap into the future of football fandom, designed to bring fans closer to the game they love with a cutting-edge, interactive platform. Born out of the skyrocketing demand for immersive online sports experiences, it's set to revolutionize how football enthusiasts connect with live action, videos, and exclusive content. Leveraging the latest in web technology, this platform will be a game-changer, offering fans a vibrant, personalized space that stands out in the digital sports arena. It's not just a project—it's the future of football, right at your fingertips.

MISSION

State the fundamental purpose of the team.

What is the primary goal the team aims to achieve?

The primary goal of the team is to create an innovative and user-centric platform that elevates the football fan experience by providing seamless access to live streams, videos, and personalized content. The team aims to deliver a cutting-edge digital hub that not only meets the evolving needs of football enthusiasts but also sets a new standard for interactivity and engagement in the sports media landscape.

## OBJECTIVES

List specific, measurable outcomes the team should achieve.

What are the key deliverables? What are the success criteria?

### Key Deliverables:

1. **Centralized Streaming Management Platform:** A user-friendly dashboard for managing live stream links, accessing metadata, and viewing upcoming streams.
2. **Video Storage and Management System:** Secure and organized storage for live-stream videos and pre-recorded content.
3. **Health Monitoring Features:** Real-time monitoring of live stream link health, with alerts for downtime or performance issues.
4. **Reporting and Analytics:** Tools for generating usage reports, tracking link performance, viewer engagement, and content popularity.

### Success Criteria:

1. **Platform Functionality and Efficiency:** The platform must be fully functional, providing seamless content management and live streaming.
2. **Client Satisfaction:** Positive feedback on platform usability, features, and overall satisfaction from clients and users.
3. **Platform Stability:** Consistent system uptime, quick response times, and minimal error rates to maintain a seamless user experience.
4. **Security:** Robust security measures with continuous monitoring, ensuring no vulnerabilities or data breaches occur.

## DELIVERY SCHEDULE

List the feature deliveries planned.

ID	Sprint	Dates: Start/Finish	Deliverables/Features/Tasks
1	1	8/26-9/9	Project Initiation and Planning
2	2	9/9-9/23	UX/UI Design and Environment Setup
3	3	9/23-10/7	Front-End Development – HTML & CSS
4	4	10/7-10/21	Front-End Development – JavaScript & React Setup
5	5	10/21-11/4	React Component Development
6	6	11/4-11/18	Testing and Debugging
7	7	11/18-12/2	Deployment and Monitoring

### COMMUNICATION PLAN

Outline how the team will communicate internally and with stakeholders.

What are the preferred communication channels and frequency?  
Who are the key stakeholders to be informed?

Preferred communication channels: WhatsApp & Microsoft Teams.

Frequency: During Sprint Start and Retrospective

Stakeholder: Gaurang Manjrekar

### RISK MANAGEMENT

Identify potential risks and mitigation strategies.

What potential obstacles could arise? How will these be managed or mitigated?

#### Potential Obstacles and Mitigation Strategies

##### 1. Time Constraints

- Obstacle: The project must be completed within a specified timeframe, which could limit the scope of development and testing phases.

- Mitigation Strategy: Implement a clear project timeline with well-defined milestones. Prioritize critical features and functionalities, and use Agile methodologies to allow for iterative development and flexibility in managing the scope.

## **2. Resource Limitations**

- Obstacle: Limited access to specialized software tools, development environments, or hardware may impact the ability to implement certain features or conduct comprehensive testing.
- Mitigation Strategy: Optimize the use of available resources by leveraging open-source tools and cloud-based services. Plan resource allocation carefully and consider outsourcing specialized tasks if necessary.

## **3. Technical Challenges**

- Obstacle: Integration with external databases, APIs, or services may present challenges, especially if access is restricted or the technology is unfamiliar to the team.
- Mitigation Strategy: Conduct thorough technical research and feasibility studies before implementation. Establish contingency plans, such as alternative solutions or mock data for testing, and ensure team members receive adequate training on new technologies.

## **4. Security Risks**

- Obstacle: The platform could be vulnerable to security breaches, data leaks, or other cybersecurity threats, especially with live streaming and user data involved.
- Mitigation Strategy: Implement robust security protocols, including encryption and secure authentication. Use established security frameworks and ensure continuous monitoring for vulnerabilities.

## **5. Stakeholder Expectations**

- Obstacle: Misalignment between stakeholder expectations and project deliverables could lead to dissatisfaction or project delays.
- Mitigation Strategy: Maintain clear and consistent communication with stakeholders throughout the project. Conduct regular check-ins and updates to ensure that all parties are aligned with the project's progress and scope.

## **6. Platform Scalability and Performance**

- Obstacle: As the platform grows, it may face challenges related to scalability and maintaining performance under high user loads.
- Mitigation Strategy: Design the platform with scalability in mind from the outset, using cloud infrastructure like AWS for dynamic scaling. Implement performance testing and optimization techniques to ensure the platform can handle increased traffic.

## Appendix:

### Deliverables (Backlog Chart)

Release	Sprint	Product Backlog
1	1	<b>Feature 01:</b> Team Charter/Roadmap <b>Feature 02:</b> Release & Stories Preparation
	2	<b>Feature 03:</b> Design UX/UI for Design for Website <b>Feature 04:</b> Receive feedback / Refine design <b>Feature 05:</b> Present the Final UX/UI Design to the client
2	3	<b>Feature 06:</b> Configure Development Environments for all team members <b>Feature 07:</b> Configure Team Git Repository <b>Feature 08:</b> Ensure all tools/software licenses are good to go configured (Teams, VSCode, etc.)
	4	<b>Feature 09:</b> Develop front-end structure and backbone using HTML & CSS (I) <b>Feature 10:</b> Develop front-end structure and backbone using HTML & CSS (II)
3	5	<b>Feature 11:</b> Setup interactive and dynamic user interface using Javascript and React <b>Feature 12:</b> Develop interactive and dynamic user interface using Javascript and React (I)
	6	<b>Feature 13:</b> Develop interactive and dynamic user interface using Javascript and React (II) <b>Feature 14:</b> Testing & Debugging <b>Feature 15:</b> Deployment & Monitoring

Sprint	Duration	Task	Start Date	End Date	Duration (Days)
Sprint 1	1 Week	Project kickoff meeting (PM)	2024-09-04	2024-09-04	1
Sprint 1	1 Week	Define user personas and journey maps (UI/UX)	2024-09-05	2024-09-06	2
Sprint 1	1 Week	Create wireframes for the platform (UI/UX)	2024-09-07	2024-09-08	2
Sprint 1	1 Week	Set up project repository and basic documentation (PM)	2024-09-09	2024-09-09	1
Sprint 1	1 Week	Prepare AWS environment for development (Back-end)	2024-09-10	2024-09-11	2
Sprint 1	1 Week	Develop high-fidelity mockups (UI/UX)	2024-09-12	2024-09-14	3
Sprint 1	1 Week	Review and finalize UI/UX designs (PM, UI/UX)	2024-09-15	2024-09-16	2
Sprint 2	2 Weeks	Set up front-end frameworks (React.js, etc.) (Front-end)	2024-09-17	2024-09-18	2
Sprint 2	2 Weeks	Basic front-end structure and routing (Front-end)	2024-09-19	2024-09-21	3
Sprint 2	2 Weeks	Design database schema and ERD (Back-end)	2024-09-22	2024-09-26	5
Sprint 2	2 Weeks	Develop RESTful APIs for basic operations (Back-end)	2024-09-27	2024-10-01	5
Sprint 3	2 Weeks	Continue front-end integration with static data (Front-end)	2024-10-02	2024-10-06	5
Sprint 3	2 Weeks	Integrate front-end with live APIs (Front-end, Back-end)	2024-10-07	2024-10-10	4
Sprint 3	2 Weeks	Implement user authentication and session management (Back-end)	2024-10-11	2024-10-13	3
Sprint 4	2 Weeks	Design and implement key UI components (Front-end)	2024-10-14	2024-10-16	3
Sprint 4	2 Weeks	Set up AWS S3 for video storage (Back-end)	2024-10-17	2024-10-19	3
Sprint 4	2 Weeks	Implement health monitoring features (Back-end)	2024-10-20	2024-10-23	4
Sprint 5	3 Weeks	Complete all front-end components (Front-end)	2024-10-24	2024-10-26	3
Sprint 5	3 Weeks	Conduct usability testing and gather feedback (UI/UX)	2024-10-27	2024-10-29	3
Sprint 5	3 Weeks	Optimize cloud infrastructure (AWS) (Back-end)	2024-10-30	2024-11-01	3
Sprint 5	3 Weeks	Perform final integration testing (All)	2024-11-02	2024-11-04	3
Sprint 6	2 Weeks	Deploy the platform on AWS (Back-end)	2024-11-05	2024-11-06	2
Sprint 6	2 Weeks	Finalize all documentation (PM, UI/UX)	2024-11-07	2024-11-08	2
Sprint 6	2 Weeks	Prepare launch plan (PM)	2024-11-09	2024-11-11	3
Sprint 7	2 Weeks	Monitor platform performance post-launch (All)	2024-11-12	2024-11-14	3
Sprint 7	2 Weeks	Conduct a retrospective meeting (All)	2024-11-15	2024-11-15	1
Sprint 7	2 Weeks	Handover and project closure (PM)	2024-11-16	2024-11-17	2

ID	Feature	Priority	Status
1	Project Kickoff & Planning	High	Planned
2	UX/UI Design	High	Planned
3	Environment Setup	Medium	Planned
4	HTML & CSS Development	Medium	Planned
5	JavaScript & React Setup	Medium	Planned
6	React Component Development	High	Planned
7	Testing & Debugging	High	Planned
-	Deployment & Monitoring	Medium	Planned

Backlog Chart Sample

Product Backlog

Release

Sprint 1 Backlog

Sprint 2 Backlog