Sports Insight Bot — Documentation

1. Introduction

In recent years, sports analytics and digital fan engagement have grown rapidly. Users increasingly seek instant access to sports-related information such as match insights, tactical analysis, and general sports knowledge. Traditional sports websites and platforms often require users to manually browse through multiple pages, which can be time-consuming and inefficient.

The Sports Insight Bot is a full-stack, chatbot-based web application designed to simplify access to sports information through a conversational interface. By allowing users to interact using natural language and optionally upload sports-related images (such as match or pitch screenshots), the system significantly enhances accessibility and user experience.

This project demonstrates the practical implementation of modern software engineering concepts, including full-stack development, API integration, multimodal AI usage, cloud deployment, and modular system design.

2. Proposed Solution

The proposed solution is an AI-powered sports chatbot system that enables users to interact with a conversational agent to receive sports-related insights, tactical explanations, and contextual analysis.

The solution consists of the following components:

Backend System

The backend is implemented using FastAPI (Python) and is responsible for:

- Handling user requests (text and image inputs)
- Processing image uploads and converting them into contextual descriptions
- Generating responses using a multimodal vision-language model accessed via OpenRouter
- Maintaining limited conversational context to ensure smooth and coherent dialogue flow

The backend exposes RESTful API endpoints that can be easily consumed by any frontend client.

Frontend System

The frontend is a lightweight, non-flashy web interface built using HTML, CSS, and JavaScript. It provides:

- A clean chat-based interface
- Support for text input and image uploads
- Real-time interaction with the backend via API calls

System Architecture

The application follows a client–server architecture, where:

- The frontend communicates with the backend using HTTPS API requests
- OpenRouter is used as the API gateway to access the AI model
- Frontend and backend are developed and deployed independently

This modular design ensures scalability, maintainability, and ease of future enhancement.

3. Unique Selling Proposition (USP)

The key differentiating features of the Sports Insight Bot include:

- Conversational Interface
  Users interact using natural language, eliminating the need for manual navigation or form-based searches.
- Multimodal Capability
  The system supports both text-based queries and image-based sports analysis, enhancing analytical depth.
- Context-Aware Responses
  Limited conversational memory allows the chatbot to maintain context, resulting in a more natural and flowing interaction.
- Monorepo Architecture
  Both frontend and backend code are maintained in a single repository, improving transparency, evaluation, and version control.
- Cloud Deployment
  The project demonstrates real-world deployment using:
  - Render for backend services
  - Vercel for frontend hosting

- Scalable Design
  The loosely coupled frontend and backend can be scaled, modified, or replaced independently.

These features make the project both technically sound and practically relevant.

4. Target Users

The Sports Insight Bot is designed for a broad range of users:

- Sports Enthusiasts
  Users seeking quick tactical insights and sports-related explanations.
- Students and Learners
  Individuals interested in sports knowledge, analytics, or AI-based systems.
- Casual Fans
  Users who follow sports occasionally and prefer an intuitive, conversational interface.
- Academic Evaluators
  Faculty members assessing full-stack development skills, system design, and deployment practices.

The system requires minimal technical expertise, making it accessible to non-technical users as well.

5.Kano Model

The Kano Model is used to classify system features based on their impact on user satisfaction. For the Sports Insight Bot, features are categorized as follows:

1. Must-Have Features (Basic Requirements)

These features are essential for system functionality. Their absence would result in user dissatisfaction.

- Text-to-Text Conversational Response
  The chatbot must correctly process user queries and generate meaningful sports-related responses.

2. Performance Features (One-Dimensional Requirements)

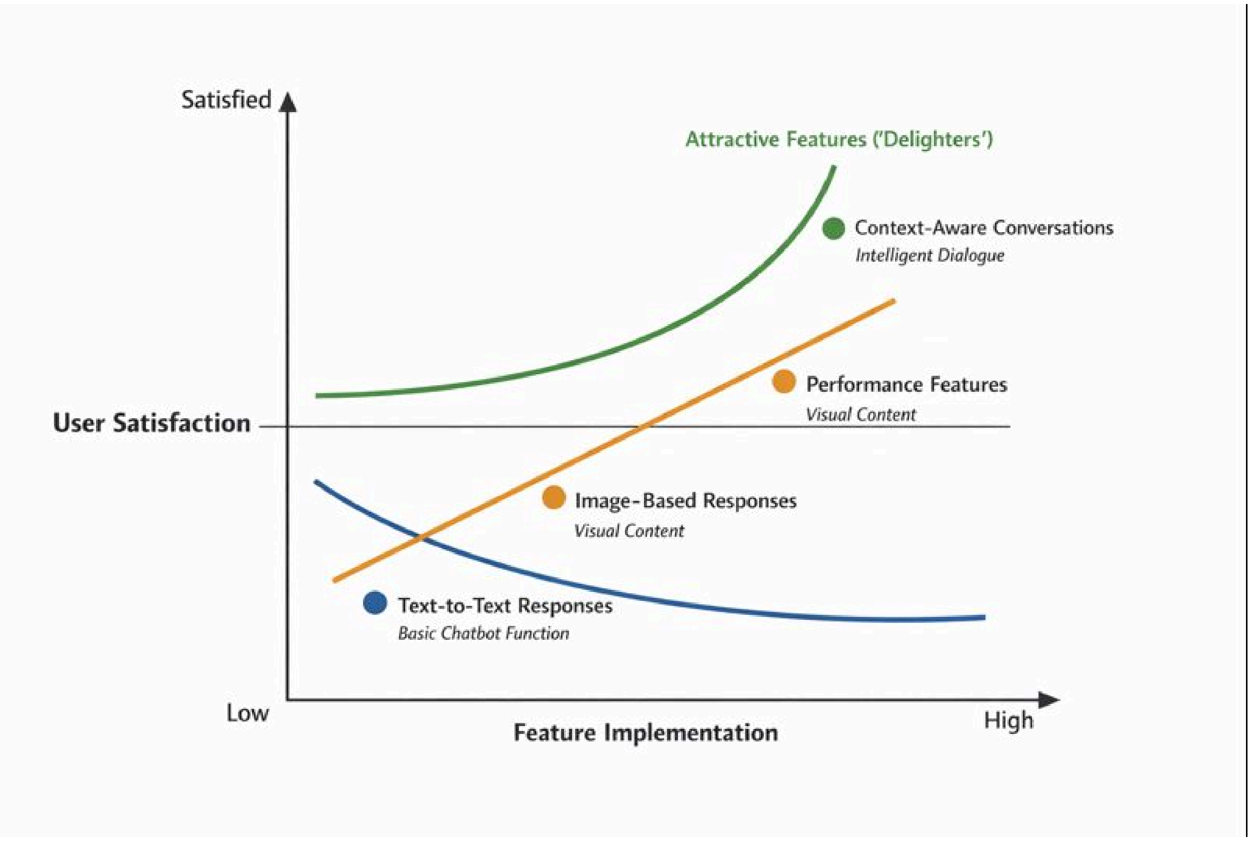
User satisfaction increases proportionally with the quality of these features.

- Image-Based Analysis
  The ability to analyze uploaded sports images and provide contextual insights enhances usefulness.

3. Attractive Features (Delighters / Excitement Factors)

These features are not explicitly expected but significantly improve user experience.

- Context Awareness and Intelligent Conversation Flow
  Maintaining conversational memory improves realism and engagement.



6. Future Scope

The project can be extended in several directions:

- Live Sports Data Integration
  Integration with real-time sports APIs for live scores, player statistics, and match updates.
- User Authentication
  Personalized experiences through user accounts and session-based memory.

- Analytics Dashboard
  Visualization of user interactions, popular queries, and system usage patterns.
- Enhanced AI Capabilities
  Improved memory handling, streaming responses, and advanced tactical modeling.

These enhancements can significantly increase the system's real-world applicability.

7. Conclusion

The Sports Insight Bot successfully demonstrates the development of a full-stack, AI-powered chatbot application. By combining a scalable FastAPI backend with a responsive web-based frontend, the system effectively addresses the need for quick, accessible, and intelligent sports information.

The use of a monorepo structure, cloud deployment, and modular system design makes the project well-suited for academic evaluation as well as future real-world expansion. Overall, the project reflects a strong understanding of full-stack development, system architecture, AI integration, and user-centric design principles.