Contents

Chapter 1

Project Proposal

1.1 Introduction

Athletes frequently seek digital guidance related to training, injuries, recovery, and health metrics. These interactions often involve sensitive personal, medical, and sports-related information. Traditional AI-powered chat systems rely heavily on cloud-based large language models (LLMs), which introduces significant privacy and data protection concerns.

The Safe Sport Advisor project proposes a privacy-first intelligent advisory system that ensures sensitive user data is never exposed to external services. The system uses a combination of local and external LLMs, guided by an intelligent routing mechanism, and stores all data securely using a NoSQL database.

1.2 Problem Statement

Most existing AI advisory systems transmit user queries directly to cloud-based LLMs without verifying whether the content contains sensitive data. This approach poses serious risks, especially in domains such as sports medicine and athlete health monitoring. There is a need for a system that can:

- Detect sensitive data in real-time
- Anonymize private information before processing
- Prevent data leakage to external AI services
- Maintain scalable and flexible data storage

1.3 Proposed Solution

The proposed solution is a privacy-preserving chat-based advisory system that acts as an intermediary between users and AI models. A local LLM is used to analyze and sanitize user input, while a NoSQL database ensures flexible and scalable storage of chat data, routing decisions, and audit logs.

1.4 Technology Stack

- Frontend: Web-based Chat Interface
- Backend: Node.js with Express
- Local LLM: Llama 3.1 via Ollama
- External LLM: OpenAI API
- Database: NoSQL Database (MongoDB)

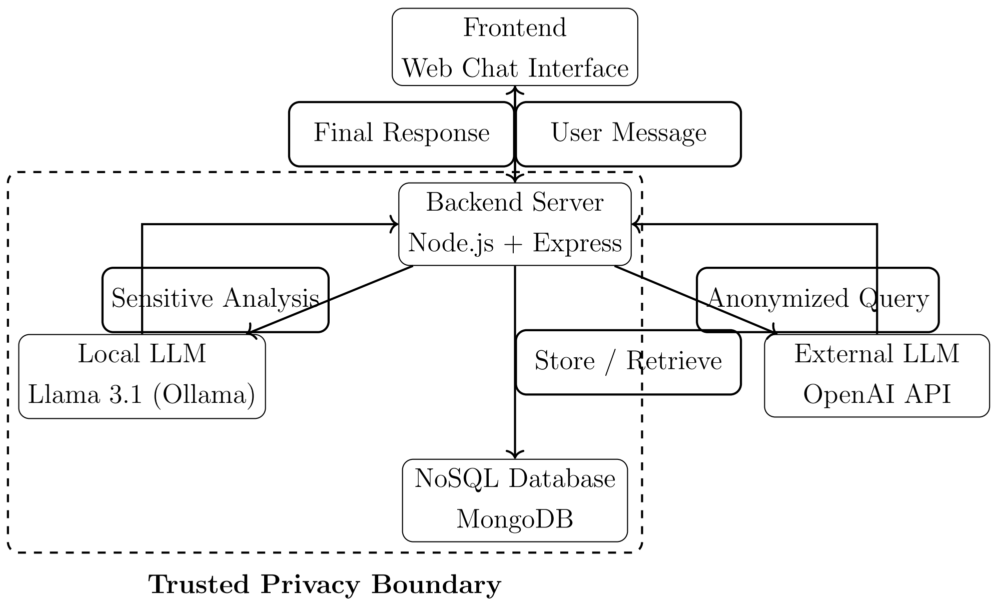
Chapter 2

System Architecture and Methodology

2.1 Technical Architecture Overview

The Safe Sport Advisor is a privacy-first intelligent advisory system designed to protect sensitive athlete information while providing accurate sports and medical guidance. The backend server acts as a trusted privacy boundary between users and artificial intelligence services. The system integrates local and external Large Language Models (LLMs) and uses a NoSQL database to provide scalable and flexible data storage.

2.2 System Architecture Diagram



2.3 Architecture Components

2.3.1 Frontend Layer

The frontend provides a chat-based interface for athletes. It is responsible only for collecting user input and displaying responses returned by the backend. No sensitive processing occurs at this layer.

2.3.2 Backend Layer

The backend, implemented using Node.js and Express, is the core control unit of the system. It handles message validation, sensitive data detection, anonymization, LLM routing, and database interaction. All privacy rules are enforced here.

2.3.3 Local LLM Layer

The local LLM runs entirely within the trusted environment. It detects sensitive personal, medical, and sports-related data and answers privacy-critical or simple queries.

2.3.4 External LLM Layer

The external LLM is used only for non-sensitive or anonymized queries that require advanced reasoning capabilities.

2.3.5 NoSQL Database Layer

The NoSQL database (MongoDB) stores user messages, anonymized data, detected entities, routing decisions, and responses in a flexible document-based format.

2.4 Detailed System Working Flow

1. The user submits a message through the frontend interface.
2. The backend receives and validates the message.
3. The message is analyzed by the local LLM to detect sensitive data.
4. If sensitive data is found, anonymization is performed.
5. The backend stores message data in the NoSQL database.
6. Intelligent routing logic selects the appropriate LLM.
7. The selected LLM generates a response.

8. The response is stored in the database.
9. The final response is returned to the frontend.

2.5 Privacy and Security Considerations

The system ensures privacy by processing sensitive data locally, anonymizing information before external communication, and maintaining a complete audit trail in the NoSQL database.

2.6 Advantages of the Proposed Architecture

- Strong privacy guarantees
- Scalable NoSQL data storage
- Modular and maintainable architecture
- Suitable for real-world deployment

Chapter 3

Methodology

3.1 Overall System Methodology

The Safe Sport Advisor follows a sequential, privacy-driven processing pipeline. Each user message passes through multiple controlled stages to ensure data security, correctness, and accountability.

3.2 Step-by-Step System Working Flow

3.2.1 Step 1: User Message Submission

The user enters a message through the frontend chat interface. The message may contain general sports questions or sensitive information such as age, injury details, or physiological measurements. The frontend sends this message to the backend using an HTTP POST request.

Key Point: No processing or decision-making is performed at the frontend level.

3.2.2 Step 2: Message Reception and Validation

The backend server receives the user message and performs basic validation, such as checking message format and length. A unique message identifier is generated for tracking purposes. The original message is temporarily stored in memory for further analysis.

3.2.3 Step 3: Sensitive Data Detection Using Local LLM

The backend forwards the user message to the local LLM running via Ollama. The local LLM analyzes the message to detect:

- Personal identifiers (e.g., age)

- Medical information (e.g., injuries, pain)
- Sports-related metrics (e.g., heart rate, training load)

The LLM returns a structured response indicating whether sensitive data is present and identifies specific sensitive entities.

Why Local LLM: Sensitive data analysis is performed locally to prevent data exposure.

3.2.4 Step 4: Anonymization of Sensitive Data

If sensitive data is detected, the backend replaces each sensitive element with a predefined placeholder (e.g., `<AGE>`, `<INJURY>`). Both the original message and the anonymized version are retained for internal processing, but only the anonymized version is allowed to leave the trusted environment.

3.2.5 Step 5: NoSQL Data Storage

The backend stores the following information in a NoSQL database:

- User identifier
- Original message
- Anonymized message
- Detected sensitive entities
- Timestamp

A document-based storage model is used, allowing flexible schema design and easy expansion.

3.2.6 Step 6: Intelligent LLM Routing Decision

The backend applies routing logic based on:

- Presence of sensitive data
- Complexity of the user query

Routing rules:

- Sensitive queries $\rightarrow$ Local LLM
- Simple non-sensitive queries $\rightarrow$ Local LLM
- Complex anonymized queries $\rightarrow$ External LLM

The routing decision and reasoning are logged in the database.

3.2.7 Step 7: Response Generation

The selected LLM generates a response:

- Local LLM responds directly for sensitive or simple queries
- External LLM processes anonymized input for complex reasoning

The generated response is returned to the backend.

3.2.8 Step 8: Response Storage

The backend stores the generated response in the NoSQL database, linked to the original message identifier. This supports traceability and auditing.

3.2.9 Step 9: Response Delivery to User

The final response is sent back to the frontend and displayed to the user in the chat interface. No internal processing details or sensitive metadata are exposed.

3.3 Privacy and Security Considerations

- Sensitive data never leaves the local environment
- External LLMs receive only anonymized input
- NoSQL database access is restricted to the backend
- Full audit trail is maintained for accountability

3.4 Advantages of Using NoSQL Database

- Flexible schema for evolving data formats
- Efficient storage of nested documents
- High scalability for large chat histories
- Faster iteration during development

3.5 Expected Outcomes

The system will provide accurate sports and medical guidance while maintaining strict privacy guarantees. The use of NoSQL databases enables scalability, and the hybrid LLM architecture ensures both security and performance.