Lab Exercise 03 Víctor Vega Sobral

F1 Strategy Assistant

The F1 Strategy Assistant is a specialized chat application built to provide Formula 1 enthusiasts with expert insights on race strategies, technical regulations, and F1 history, alongisde simple explanations of complex concepts of this sport, in order to help new fans familiarize with the sport.

The app also provides the user the option to select from their local LLMs.

Tools and Technologies Selection

Core technologies

1. **Streamlit** is selected as the primary framework for several reasons. It has rapid prototypiing capabilites for data-focused applications, with built-in widgets that simplify UI development. Moreover, and most important, it´s Python-based, allowing easy integration with other libraries.
2. **LM Studio** is the chosen LLM backend, becuase it provides an easy API integration that allows us to connect easily the model through endpoints. It also allows experimenting with different model sizes and capabilities, apart from reducing latency and removing dependency of cloud-base services.
3. **Python Libraries** needed:
    1. `requests`: handles the API communication with LM Studio.
    2. `uuid`: generates unique identifiers for the chat sessions,
    3. `datetime`: used for timestamping the conversations.
    4. `json`: used for processing API responses.

Development Process

First Step: Initial Setup and Requirements Analysis

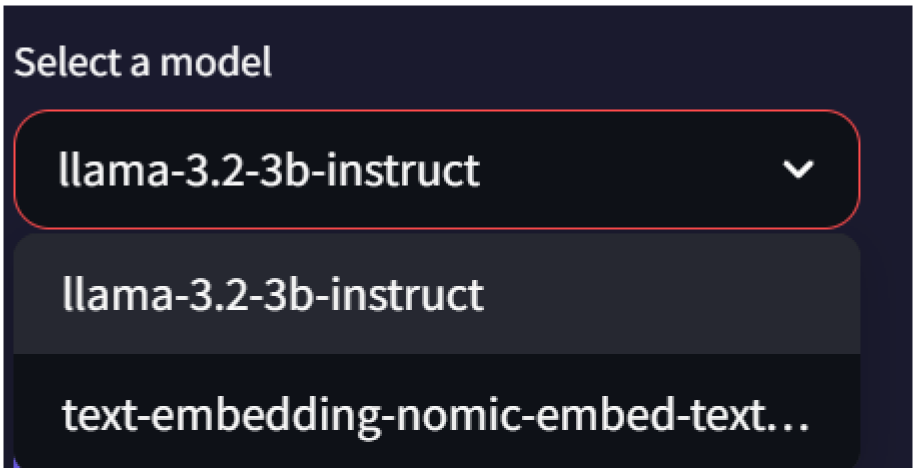
First step involded is defining the scope and requirements for the F1 Strategy Assistant. The application needed to:

- Provide expert-level F1 Knowledge through an LLM (in our case, LLama 3.2).
- Support multi-session chat management.
- Present information in a F1-themed interface.
- Allow customization of model parameters.

This step is crucial for defining what we want to do with the LLM and also the parts that our application will have. The purpose is to mimic ChatGPT-liked chat applications with some of the parts that are common in them (chat history, model selection, etc).

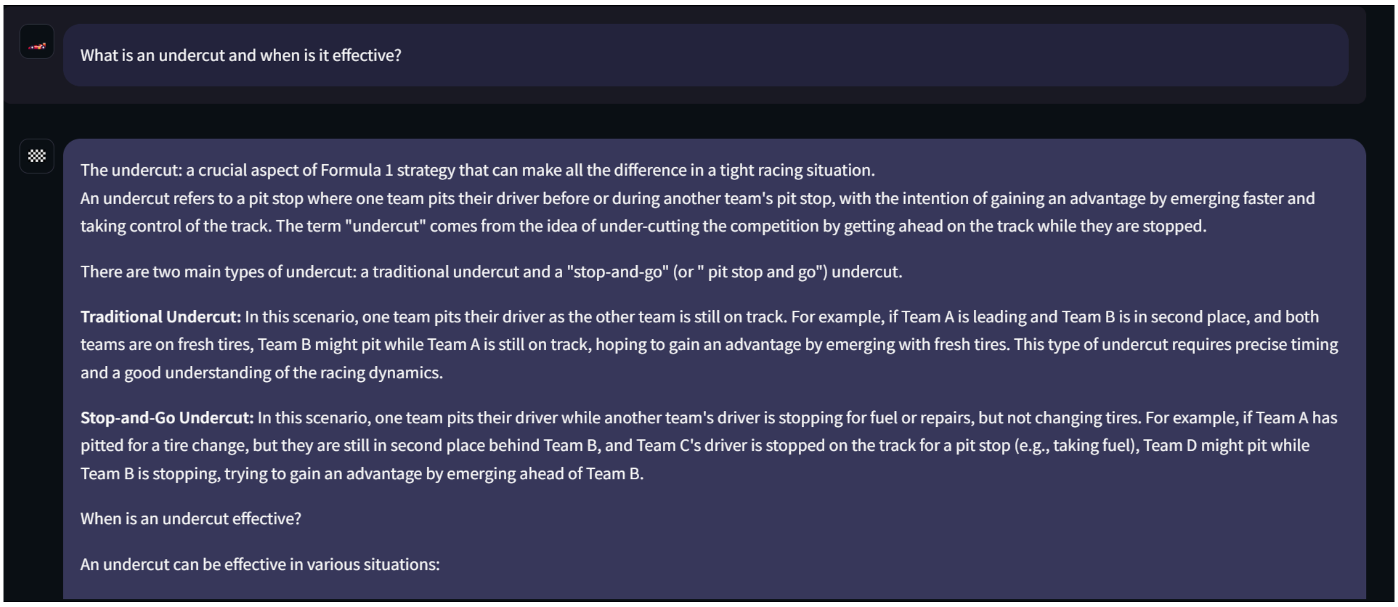
Second step: backend integration with LM Studip

LM studio was set up to serve LMs locally, and API integration was implemented to discover the available models through endpoint `v1/models`, send queries to the selected model through a POST petition with our `payload` dictionary and finally parsing to JSON format the model response.



Step 3: Defining the F1 Expert System Prompt

A critical aspect was crafting a specialized system prompt to transform a general-purpose language model into an F1 expert. This prompt provides context and boundaries for the model's responses.



What is an undercut and when is it effective?

The undercut: a crucial aspect of Formula 1 strategy that can make all the difference in a tight racing situation.
An undercut refers to a pit stop where one team pits their driver before or during another team's pit stop, with the intention of gaining an advantage by emerging faster and taking control of the track. The term "undercut" comes from the idea of under-cutting the competition by getting ahead on the track while they are stopped.

There are two main types of undercut: a traditional undercut and a "stop-and-go" (or " pit stop and go") undercut.

Traditional Undercut: In this scenario, one team pits their driver as the other team is still on track. For example, if Team A is leading and Team B is in second place, and both teams are on fresh tires, Team B might pit while Team A is still on track, hoping to gain an advantage by emerging with fresh tires. This type of undercut requires precise timing and a good understanding of the racing dynamics.

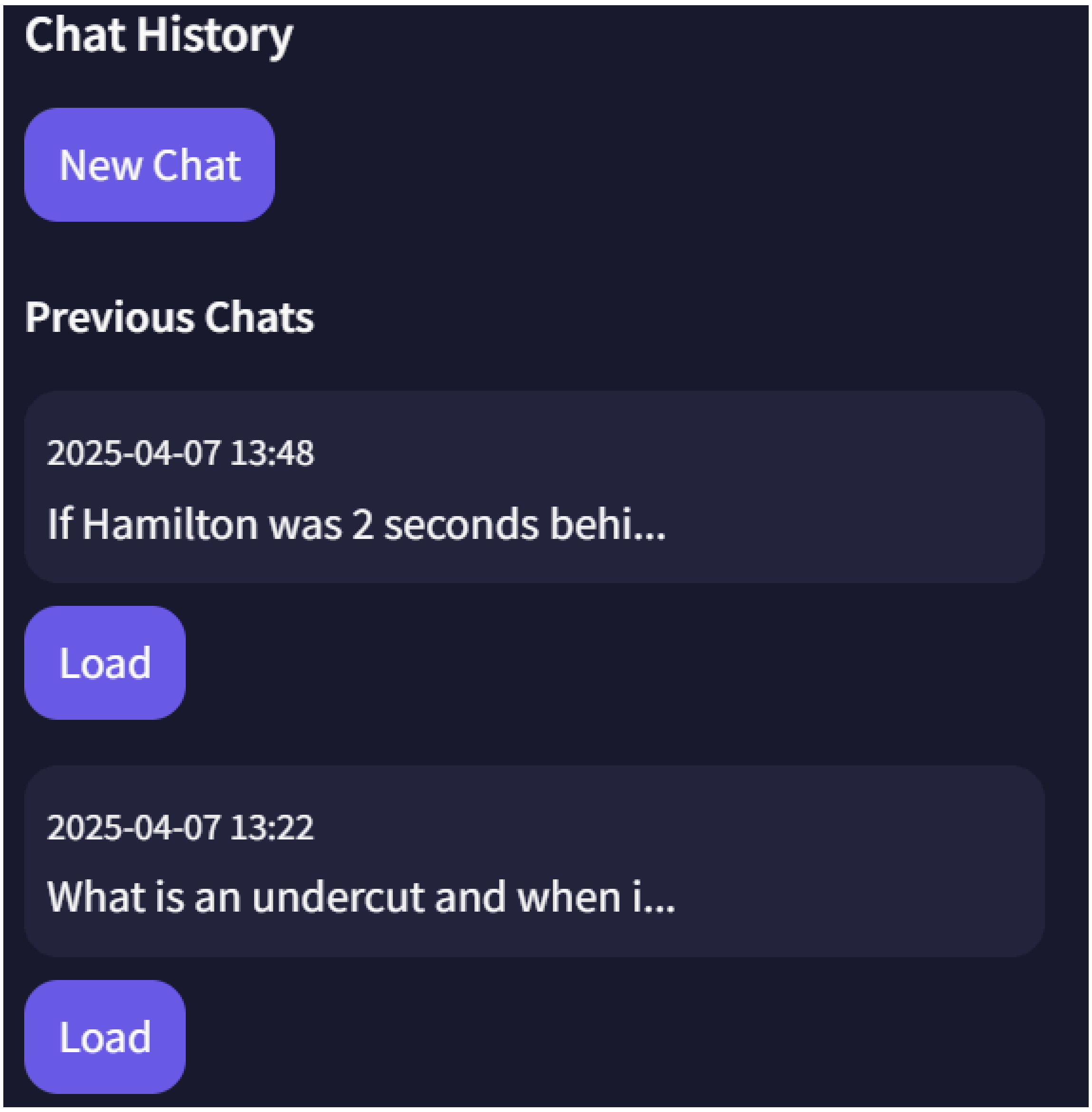
Stop-and-Go Undercut: In this scenario, one team pits their driver while another team's driver is stopping for fuel or repairs, but not changing tires. For example, if Team A has pitted for a tire change, but they are still in second place behind Team B, and Team C's driver is stopped on the track for a pit stop (e.g., taking fuel), Team D might pit while Team B is stopping, trying to gain an advantage by emerging ahead of Team B.

When is an undercut effective?

An undercut can be effective in various situations:

Step 4: Implementing Chat History Management

To support multiple conversations, a robust chat history management system was implemented using Streamlit's session state capabilities and unique identifiers for each chat.



2025-04-07 13:48

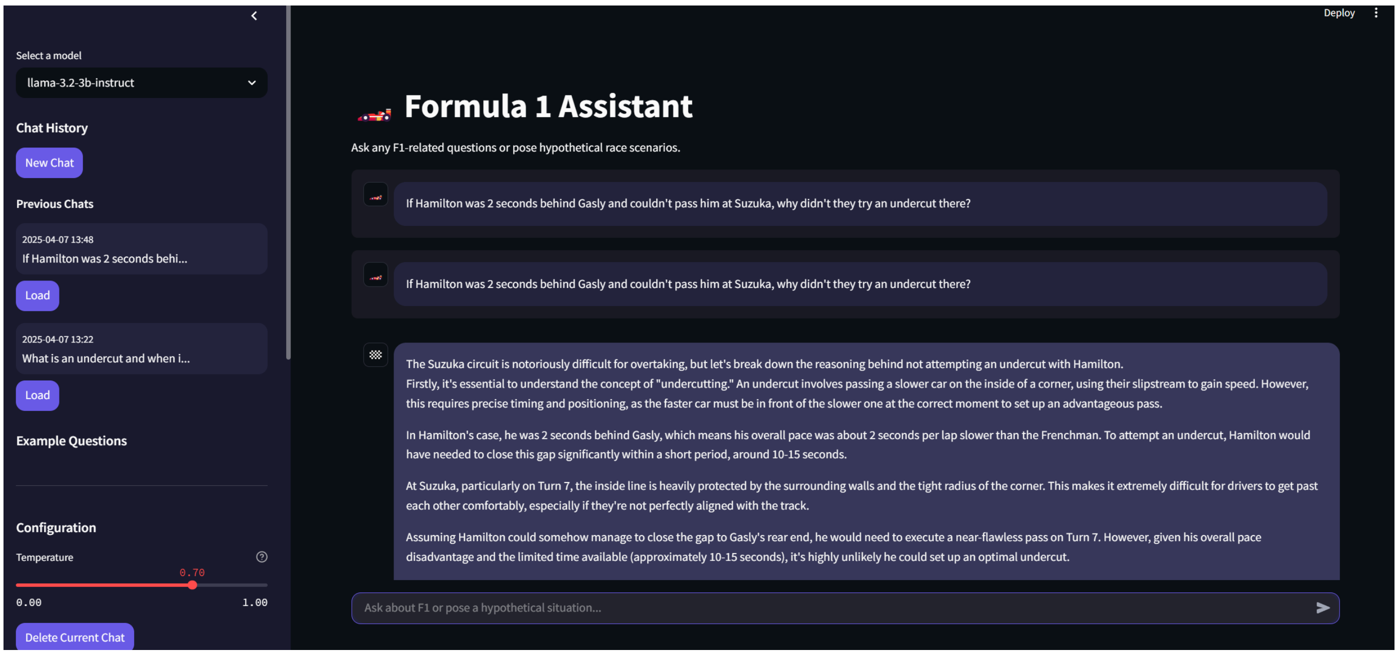
If Hamilton was 2 seconds behi...

2025-04-07 13:22

What is an undercut and when i...

Step 5: Designing a Formula 1 Themed UI

To create an immersive F1 experience, custom CSS styling was applied to Streamlit's default components, creating a dark theme with purple/blue accents similar to modern F1 team colors and digital dashboards.



Formula 1 Assistant

Ask any F1-related questions or pose hypothetical race scenarios.

If Hamilton was 2 seconds behind Gasly and couldn't pass him at Suzuka, why didn't they try an undercut there?

If Hamilton was 2 seconds behind Gasly and couldn't pass him at Suzuka, why didn't they try an undercut there?

The Suzuka circuit is notoriously difficult for overtaking, but let's break down the reasoning behind not attempting an undercut with Hamilton.
Firstly, it's essential to understand the concept of "undercutting." An undercut involves passing a slower car on the inside of a corner, using their slipstream to gain speed. However, this requires precise timing and positioning, as the faster car must be in front of the slower one at the correct moment to set up an advantageous pass.

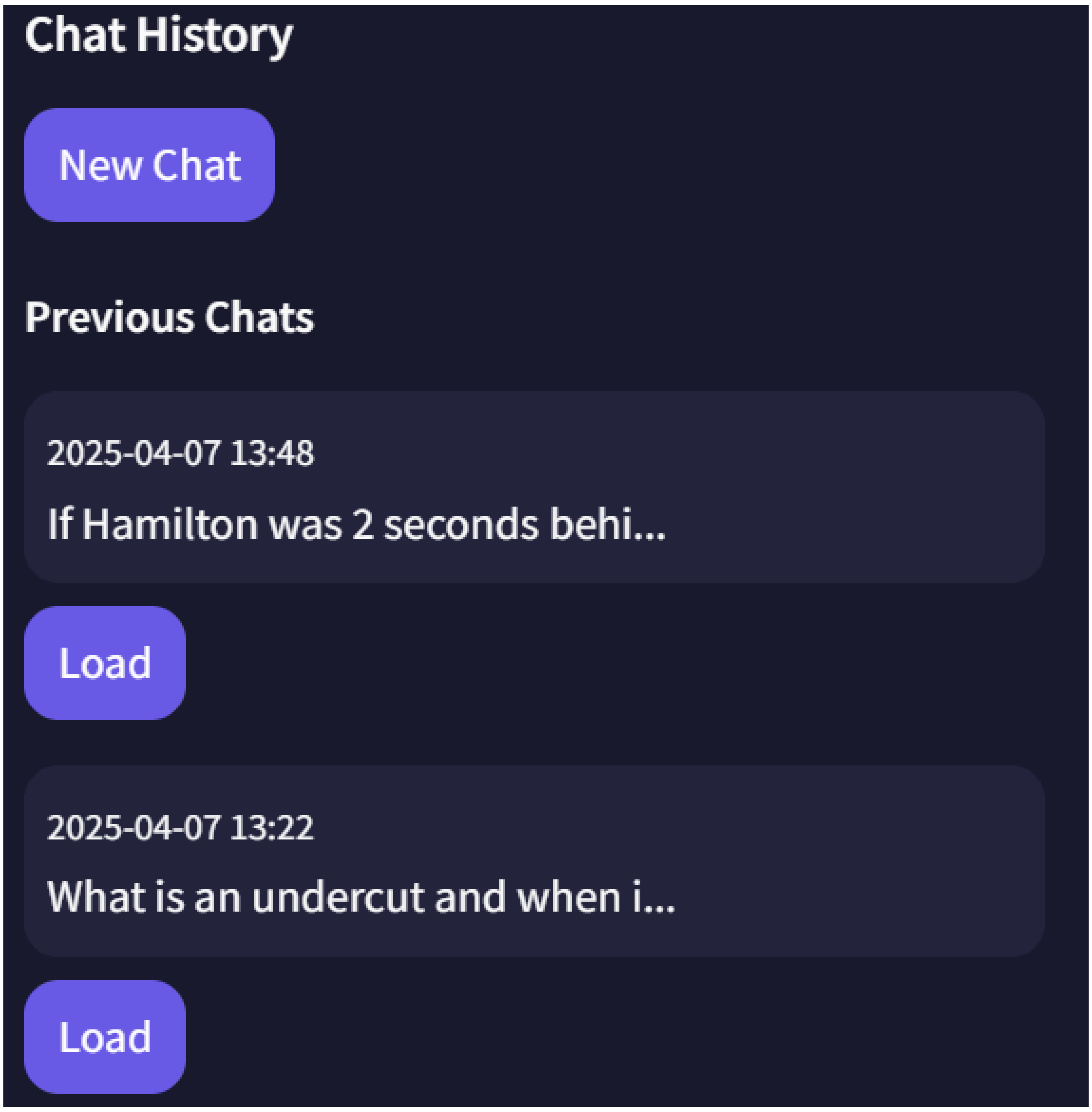
In Hamilton's case, he was 2 seconds behind Gasly, which means his overall pace was about 2 seconds per lap slower than the Frenchman. To attempt an undercut, Hamilton would have needed to close this gap significantly within a short period, around 10-15 seconds.

At Suzuka, particularly on Turn 7, the inside line is heavily protected by the surrounding walls and the tight radius of the corner. This makes it extremely difficult for drivers to get past each other comfortably, especially if they're not perfectly aligned with the track.

Assuming Hamilton could somehow manage to close the gap to Gasly's rear end, he would need to execute a near-flawless pass on Turn 7. However, given his overall pace disadvantage and the limited time available (approximately 10-15 seconds), it's highly unlikely he could set up an optimal undercut.

Step 6: Implementing the Chat Interface

The chat interface was built to display messages with F1-themed icons (race car for user, checkered flag for assistant) and provide an input field for user questions. This can be seen again



2025-04-07 13:48

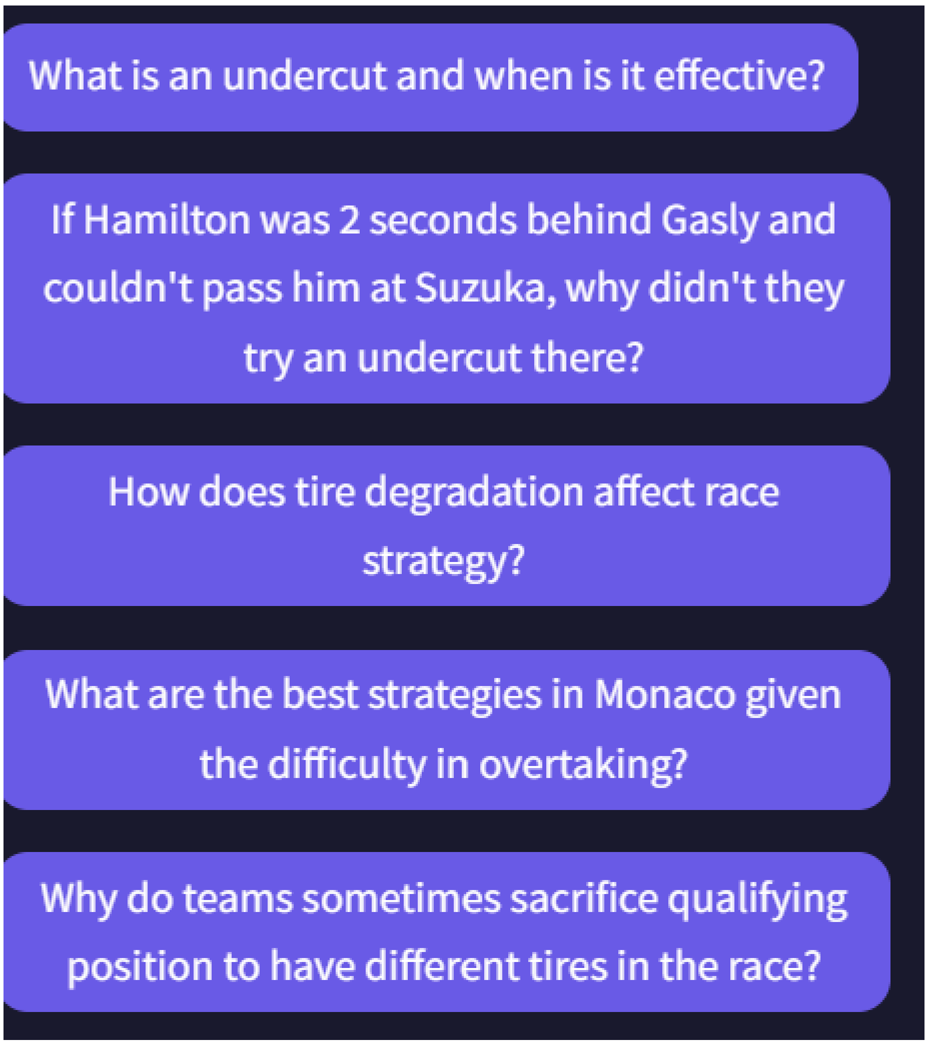
If Hamilton was 2 seconds behi...

2025-04-07 13:22

What is an undercut and when i...

Step 7: Adding Example Questions for User Engagement

To help users understand the assistant's capabilities, a set of example questions was added to the sidebar, with each example functioning as a clickable button.



What is an undercut and when is it effective?

If Hamilton was 2 seconds behind Gasly and couldn't pass him at Suzuka, why didn't they try an undercut there?

How does tire degradation affect race strategy?

What are the best strategies in Monaco given the difficulty in overtaking?

Why do teams sometimes sacrifice qualifying position to have different tires in the race?

Development Challenges & Solutions

Challenge 1: Local Model Management

Problem : Integrating with locally hosted language models required handling connectivity issues and model loading states.

Solution : Implemented robust error handling in the API communication functions and added fallback options for when models are unavailable.

Challenge 2: Maintaining Chat State

Problem : Streamlit refreshes the entire application on each interaction, which can make maintaining chat state difficult.

Solution : Leveraged Streamlit's session state to persist chat histories and manage multiple conversations effectively.

Challenge 3: F1-Specific Knowledge

Problem : General language models may not have specialized F1 strategy knowledge, or might provide overly generic responses.

Solution : Created a detailed system prompt that frames the model as an F1 expert and includes specific instructions for analyzing race scenarios.

Conclusion

The development of the F1 Strategy Assistant demonstrates how specialized LLM applications can provide value in specific domains. By combining Streamlit's rapid development capabilities with LM Studio's flexible model hosting, a powerful tool for F1 enthusiasts was created with relatively simple code.

The application shows that even without custom model training, properly prompted language models can provide domain-specific expertise when framed with the right context and user interface.