Prompt Injection Attacks Tool - Technical Document

# 1. Introduction

The Prompt Injection Attacks Tool is a research project designed to investigate the susceptibility of Large Language Models (LLMs) to prompt injection attacks. This tool enables users to input prompts through a user interface, which are then manipulated by a fine-tuned Prompt Injection model based on the phi-2 model. The modified prompt is sent to the original LLM, aiming to cause the LLM to return incorrect results. The tool compares the outcomes of the original and modified prompts to showcase the effectiveness of the attack.

# 2. Architecture Overview

## 2.1 User Interface (UI)

- Input Field: Allows users to enter a prompt.  
- Send Button: Submits the prompt to the system.  
- Result Display Area: Shows both the original and attacked results side by side for comparison.

## 2.2 Prompt Injection Model

- Model Architecture: The Prompt Injection model is built by fine-tuning the phi-2 model, which has been specifically adapted to subtly alter prompts in a way that deceives the original LLM.  
- Functionality:  
 - Receives the user’s input prompt.  
 - Generates a slightly modified version of the prompt that is designed to cause the LLM to produce an incorrect or misleading response.

## 2.3 Original LLM

- Model Selection: The LLM used is the original, unmodified version that will process both the original and the attacked prompts.  
- Functionality:  
 - Processes the original prompt to produce the expected result.  
 - Processes the attacked prompt to produce the altered result.

# 3. Data Flow

1. User Input:  
 - The user enters a prompt in the UI and clicks 'Send.'  
  
2. Prompt Handling:  
 - The prompt is sent to two different pathways:  
 1. Direct Pathway: The original prompt is sent directly to the Original LLM.  
 2. Injection Pathway: The prompt is sent to the Prompt Injection model, which modifies the prompt and then forwards it to the Original LLM.  
  
3. Result Generation:  
 - The Original LLM processes both the original and modified prompts, returning two different results.  
 - The original result (from the unaltered prompt) and the attacked result (from the altered prompt) are displayed side by side in the UI.

# 4. System Components

## 4.1 Frontend (UI)

- Technologies: HTML, CSS, JavaScript.  
- Functionality: Captures user input, sends data to the backend, and displays results.

## 4.2 Backend

- Technologies: Python, Flask, PyTorch (for model handling).  
- Functionality:  
 - Handles requests from the UI.  
 - Manages interactions with both the Prompt Injection model and the Original LLM.  
 - Returns results to the frontend for display.

## 4.3 Prompt Injection Model

- Model: Fine-tuned phi-2 model.  
- Training: Trained on a dataset that focuses on manipulating prompts to cause incorrect outputs from LLMs.  
- Deployment: Hosted on a server or cloud service with GPU support.

## 4.4 Original LLM

- Model: Pre-trained LLM (e.g., Gemini, Llama2, GPT-4, BERT, etc.).  
- Deployment: Same environment as the Prompt Injection model to ensure consistency in results.