**1. Overview**

**Product Name**: Simple Prompt Optimizer (Powered by LLM)  
**Objective**: Provide a minimal, single-screen web application where the user enters a brief prompt, and an LLM automatically transforms it into a detailed, domain-specific prompt. The user can then copy the advanced prompt or optionally use it immediately for further querying.

**2. Key Goal**

* **Leverage an LLM for Prompt Transformation**  
  Instead of manually crafting rules or templates, the system will pass the user’s short prompt to a behind-the-scenes LLM trained (or fine-tuned) to output a more comprehensive “prompt-on-a-prompt.” This ensures users do not need to adjust settings; the LLM itself applies the latest best practices in prompt engineering.

**3. Scope**

1. **In Scope**
   * A single text input for the user’s basic prompt.
   * Automated calling of an LLM (“transformer LLM”) to produce the advanced prompt.
   * Displaying the advanced prompt to the user for copying or additional use.
   * (Optional) One-click button to send the advanced prompt to the same or another LLM (“target LLM”) for final answers.
2. **Out of Scope**
   * Complex user interfaces with domain selections or toggles.
   * Detailed analytics, dashboards, or user management.
   * Support for multiple LLM providers at launch (we focus on one provider or model).

**4. User Flow**

1. **Prompt Input**
   1. User navigates to the single-page application.
   2. User enters a simple query (e.g., “Explain the benefits of meditation”).
2. **Prompt Transformation**
   1. The application sends the user input to the “transformer LLM.”
   2. The LLM returns a more detailed, specialized prompt that follows best-practice prompt engineering patterns (role prompting, structured instructions, constraints, etc.).
3. **Result Display**
   1. The application shows the optimized prompt to the user in a read-only text area.
   2. The user can copy the advanced prompt to use elsewhere.
   3. *(Optional)* The user can click “Generate Answer” to immediately send the optimized prompt to the same or another LLM, displaying the final response in the interface.

**5. Functional Requirements**

| **Requirement** | **Description** | **Priority** |
| --- | --- | --- |
| Single Prompt Input Field | Minimal text box; user pastes or types a short prompt. | High |
| LLM-Based Prompt Transformation | The system calls a “transformer LLM” endpoint to generate the complex prompt. | High |
| Display Optimized Prompt | Show the transformed prompt in a read-only or minimal editing area. | High |
| (Optional) LLM Query | Allow one-click usage of the advanced prompt on the same or another LLM to get a final response. | Medium |

**6. Non-Functional Requirements**

1. **Performance**
   * The transformation call to the LLM should typically complete within a few seconds, dependent on the LLM provider’s API.
2. **Security**
   * HTTPS for all in-transit data.
   * Keep prompt logs secure or do not store them at all if privacy is required.
3. **Scalability**
   * Only a small set of users expected. Standard hosting solutions are sufficient.
4. **Maintainability**
   * Minimal codebase; primarily a front-end with a simple back-end call to the LLM’s API.
5. **Reliability**
   * Ensure graceful error handling if the LLM API fails or times out.

**7. Implementation Plan**

1. **Phase 1: Core Functionality (1–2 weeks)**
   * Implement the front-end with a single text box and a “Transform Prompt” button.
   * Connect to the chosen “transformer LLM” API.
   * Display the LLM-generated advanced prompt.
2. **Phase 2: Optional Features (1–2 weeks)**
   * Add a “Generate Answer” button that forwards the optimized prompt to the same or another LLM and displays the response.
   * Incorporate basic error handling and possibly a loading indicator.
3. **Phase 3: Feedback & Refinements (Ongoing)**
   * Gather feedback from the small user group.
   * Adjust the prompt transformation logic or the LLM’s instructions if the generated prompts need improvement.

**8. Success Criteria**

* **Ease of Use**: Users only see a single field for their prompt; no options or complex steps.
* **Improvement in Prompt Quality**: The LLM-generated advanced prompt consistently yields better final responses (verified by user feedback).
* **Minimal Maintenance**: The tool runs reliably with little overhead, given the small user base.

**Conclusion**

This revised PRD emphasizes a minimalistic approach where **an LLM is used to transform a simple user prompt into a sophisticated, domain-specific prompt**. By hiding the complexity of prompt engineering from the user, this product ensures easy adoption and consistently high-quality prompts.

| **Model ID** | **Model Name** |
| --- | --- |
| llama-3.3-70b | Llama 3.3 70B |
| llama-3.2-3b | Llama 3.2 3B |
| dolphin-2.9.2-qwen2-72b | Dolphin 2.9.2 Qwen2 72B |
| llama-3.1-405b | Llama 3.1 405B |
| qwen-2.5-coder-32b | Qwen 2.5 Coder 32B |
| deepseek-r1-llama-70b | DeepSeek R1 Llama 70B |
| deepseek-r1-671b | DeepSeek R1 671B |
| qwen-2.5-vl | Qwen 2.5 VL 72B |