DEVELOP A PROMPT-BASED APPLICATION TAILORED TO THEIR PERSONAL NEEDS, FOSTERING CREATIVITY AND PRACTICAL PROBLEM-SOLVING SKILLS WHILE LEVERAGING THE CAPABILITIES OF LARGE LANGUAGE MODELS.

## ****1. Abstract****

This project presents the development of a **prompt-based AI application** that adapts to user needs, stimulates creativity, and enhances practical problem-solving skills. By leveraging **Large Language Models (LLMs)** such as GPT, Claude, and Gemini, the system allows users to generate context-specific insights, ideas, and solutions. The application personalizes prompts according to the user’s profile, goals, and activity mode—fostering both critical and creative thinking through interactive AI collaboration.

## ****2. Objectives****

To design a **prompt-driven system** that responds to personalized user goals.

To foster **creative thinking** and **analytical reasoning** through structured prompt templates.

To enable **multi-model AI integration**, comparing outputs across LLMs.

To encourage **reflective learning** through feedback and re-prompting mechanisms.

## ****3. Key Features****

| **Feature** | **Description** |
| --- | --- |
| **Personalized Prompting Engine** | Customizes prompts based on user preferences, profession, or learning goals. |
| **Creativity Mode** | Generates imaginative ideas (e.g., storylines, campaigns, innovations). |
| **Problem-Solving Mode** | Uses reasoning and chain-of-thought prompts for analytical solutions. |
| **Multi-AI Compatibility** | Supports GPT, Gemini, and Claude API calls for comparative responses. |
| **Feedback Loop** | Allows users to refine or regenerate prompts for improved accuracy. |

4. System Architecture

+-------------------------------------------------------+

| USER INTERFACE (CLI/UI) |

| - Input: topic, goal, or problem |

| - Output: creative idea / analytical solution |

+-------------------------------------------------------+

↓

+-------------------------------------------------------+

| PERSONALIZATION MODULE |

| - Reads user profile, interests, and context |

| - Selects suitable prompt template |

+-------------------------------------------------------+

↓

+-------------------------------------------------------+

| PROMPT GENERATION ENGINE |

| - Builds structured prompts based on mode |

| - Injects examples, roles, or reasoning cues |

+-------------------------------------------------------+

↓

+-------------------------------------------------------+

| LLM INTERFACE LAYER |

| - Sends prompts to GPT / Gemini / Claude APIs |

| - Fetches, formats, and compares responses |

+-------------------------------------------------------+

↓

+-------------------------------------------------------+

| OUTPUT FEEDBACK & REFLECTION |

| - Displays results |

| - Asks user for improvement / re-prompting |

+-------------------------------------------------------+

## ****5. Implementation Plan****

| **Phase** | **Task** | **Tools/Tech** |
| --- | --- | --- |
| **Phase 1** | Define user profile and personalization logic | Python, JSON |
| **Phase 2** | Develop prompt generation module | Custom prompt templates |
| **Phase 3** | Integrate LLM APIs | OpenAI, Google Gemini SDK |
| **Phase 4** | Add comparative response & feedback system | Python CLI / Streamlit |
| **Phase 5** | Test with user scenarios | AI-assisted evaluation |

6. Sample Python Prototype

prompt\_based\_app.py

A personalized prompt-based AI application for creativity and problem-solving

"""

import os

import openai

import google.generativeai as genai

# --- Configuration ---

openai.api\_key = os.getenv("OPENAI\_API\_KEY")

genai.configure(api\_key=os.getenv("GOOGLE\_API\_KEY"))

# --- User Profile ---

user\_profile = {

"name": "Koushi",

"profession": "Student",

"interest": "Environmental innovation and storytelling",

"goal": "Generate new ideas for eco-awareness and education"

}

# --- Prompt Builder ---

def build\_prompt(user\_input, mode="creative"):

if mode == "creative":

return f"""

You are a creative AI mentor guiding {user\_profile['name']} ({user\_profile['profession']}).

Interest: {user\_profile['interest']}

Goal: {user\_profile['goal']}

Task: {user\_input}

Respond with imagination and originality while keeping it practical.

"""

elif mode == "problem-solving":

return f"""

You are an analytical AI tutor helping {user\_profile['name']} ({user\_profile['profession']}).

Focus on clarity, logic, and step-by-step reasoning.

Problem: {user\_input}

Provide a structured and feasible solution.

"""

else:

return user\_input

# --- LLM Interface ---

def query\_gpt(prompt):

try:

response = openai.ChatCompletion.create(

model="gpt-4o-mini",

messages=[{"role": "user", "content": prompt}],

temperature=0.7

)

return response.choices[0].message["content"]

except Exception as e:

return f"GPT Error: {e}"

def query\_gemini(prompt):

try:

model = genai.GenerativeModel("gemini-1.5-flash")

response = model.generate\_content(prompt)

return response.text

except Exception as e:

return f"Gemini Error: {e}"

# --- Application Execution ---

def run\_prompt\_app():

print("\n✨ PERSONALIZED PROMPT-BASED AI APPLICATION ✨")

print("Choose Mode:\n1. Creative Mode\n2. Problem-Solving Mode\n")

choice = input("Enter mode (1/2): ").strip()

mode = "creative" if choice == "1" else "problem-solving"

user\_input = input("\nEnter your topic or problem: ").strip()

prompt = build\_prompt(user\_input, mode=mode)

print("\n🔍 Processing your request...\n")

gpt\_output = query\_gpt(prompt)

gemini\_output = query\_gemini(prompt)

print("\n==================== GPT RESPONSE ====================")

print(gpt\_output)

print("\n=================== GEMINI RESPONSE ==================")

print(gemini\_output)

if \_\_name\_\_ == "\_\_main\_\_":

run\_prompt\_app()

## ****7. Example Test Scenarios****

| **Scenario** | **Input** | **Mode** | **Expected Output** |
| --- | --- | --- | --- |
| 1 | “Design a school campaign to reduce plastic waste.” | Creative | AI suggests innovative awareness activities and slogans. |
| 2 | “How can AI improve recycling management?” | Problem-Solving | Logical, stepwise plan integrating sensors and data analytics. |
| 3 | “Write a short eco-poem for students.” | Creative | Generates an original, age-appropriate poem on sustainability. |

## ****8. Evaluation Metrics****

| **Metric** | **Description** |
| --- | --- |
| **Relevance** | Alignment with user’s goal and context |
| **Creativity** | Novelty and imagination in responses |
| **Practicality** | Feasibility and logical structure |
| **Personalization** | Adaptation to user’s profile and mode |
| **Diversity** | Variety across different AI model responses |

## ****9. Applications****

**Education:** AI-assisted personalized learning or creative writing support.

**Environment:** Generating awareness campaigns and sustainable solutions.

**Entrepreneurship:** Ideation and problem-solving for startups.

**Creative Arts:** Story writing, poetry, and idea generation.

**Professional Training:** Decision-making and case-based reasoning.

## ****10. Conclusion****

This project successfully demonstrates how **prompt engineering** and **LLMs** can be harnessed to create **personalized AI companions** that inspire creativity and logical reasoning.  
By combining structured prompting, multi-model integration, and user-centered design, the system encourages users to engage in **reflective, purposeful interactions** with AI—making technology a true partner in learning and innovation.