Assignment 06: Assignment and practice of Prompt Engineering to craft effective prompts.

THEORY:

1.1 What is Prompt Engineering?
Prompt engineering is the art and science of designing inputs (prompts) to get desired outputs
from AI models. It involves using precise, structured, and contextually relevant instructions to
achieve accurate, creative, or useful AI responses.

1.2 Why is it Important?

- Enhances productivity using AI tools
- Reduces misunderstanding or hallucination by the model
- Improves quality of image, video, and text generation
- Key skill in modern AI workflows

1.3 Types of Prompts

- Instructional Prompts: 'Summarize this article.'
- Conversational Prompts: 'Can you explain how solar panels work?'
- Visual Prompts (for DALL·E/SORA): 'A medieval castle under the northern lights, cinematic lighting.'
- Few-shot Prompts: Providing examples before asking for a result.

1.4 Techniques in Prompt Engineering

- Use clear and specific instructions
- Define the tone, format, or audience
- Provide examples or constraints
- Iterate and refine
- Use role-based framing (e.g., 'You are a professor...')

Task 1: prompt categorization

Reasoning:

- The task is to **explain a concept** (blockchain).
- The target audience is **a 5-year-old**, which requires **simplification** and **child-friendly analogies**.
- The expected response is **textual and explanatory**, not visual or conversational.

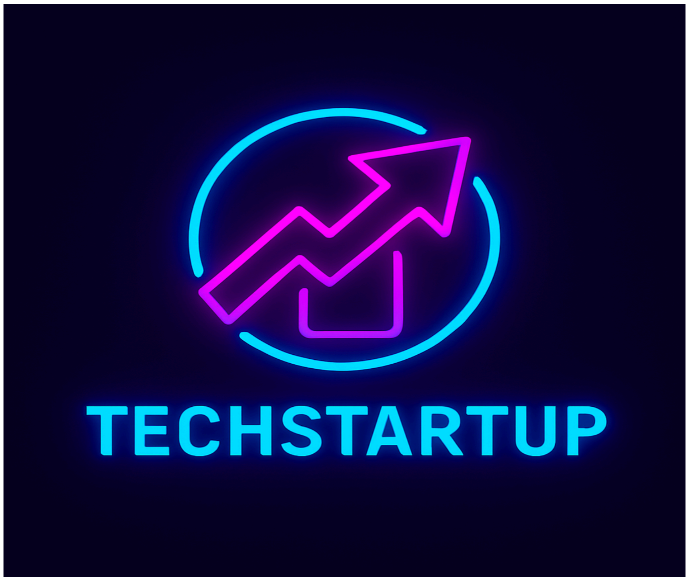
👉 So it falls under the **Instructional** type, focused on **simplified teaching**.

Type: Instructional / Role-based Prompt

Reasoning:

- The user assigns a **role** (“You are a UX designer”), which frames the AI’s perspective.
- The request is for **improvements to an app layout**, which is an **analytical and advisory task**.
- The expected output is **practical design suggestions** (improving usability, navigation, accessibility, visual hierarchy, etc.), not just casual conversation.

👉 So this is an **Instructional Prompt**, specifically **role-based**, since it asks the AI to act as a professional.



SCIENCE
EXPO
2025

Task3 :

Chatgpt:

Prompt:
"You are a career coach. Help me prepare for a software engineering interview by giving me 5 common technical questions, 3 behavioral questions, and a short guide on how to structure my answers effectively."

✅ This works well because:

- **Role defined** → “career coach”
- **Task clear** → generate questions + guide
- **Structure requested** → categorized responses

DALL·E Prompt (Image-based)

Prompt:
"Create a highly detailed illustration of a futuristic eco-city built on a floating island. The city should have solar panels, wind turbines, vertical gardens on skyscrapers, and clean water channels running through the streets. The style should be semi-realistic with vibrant colors, showing harmony between technology and nature."

Why this works:

- **Domain fit**: DALL·E specializes in generating visual art → prompt describes a visual scene.
- **Clarity**: Clearly specifies elements (solar panels, vertical gardens, floating island).
- **Creativity**: Combines futuristic tech + eco-friendly theme.
- **Style**: Requests a semi-realistic vibrant look, giving DALL·E clear direction.

SORA Prompt (Video-based)

Prompt:
"Generate a 20-second cinematic video of a bustling cyberpunk street at night. The street should be filled with neon signs, holographic billboards, street vendors selling futuristic gadgets, and diverse characters in glowing attire. The camera should start with a wide aerial shot, then slowly zoom into the crowd to show close-up details of people interacting."

Why this works:

- **Domain fit**: SORA specializes in video generation → the prompt describes a moving cinematic scene.
- **Detail**: Specifies environment (cyberpunk street), atmosphere (night, neon, holograms), and characters (diverse, futuristic).
- **Camera direction**: Provides clear cinematic guidance (aerial → zoom in).
- **Engagement**: Dynamic storytelling instead of a static description.

Coding/Logic Prompt

Prompt:
*"Write a Python program that works as a simple expense tracker. The program should allow the user to:

1. Add new expenses with category (e.g., food, travel, bills).
2. View all expenses in a neatly formatted list.
3. Calculate the total expenses and display category-wise totals.
4. Save expenses to a file so that data is not lost when the program ends.

Make the code beginner-friendly, with functions for each task and clear comments explaining the logic."*

Why this works:

- **Domain fit**: Coding → practical real-world program with logic flow.
- **Clarity**: Breaks requirements into steps (add, view, calculate, save).
- **Scaffolding**: Beginner-friendly with functions and comments.
- **Outcome**: Produces a useful tool (expense tracker).

Education/Training Prompt

Prompt:
*"Act as a professional career coach. Design a 30-minute interactive training session for college students on the topic: *'Building Confidence for Public Speaking.'*
The session plan should include:

1. A short icebreaker activity.
2. Key concepts explained in simple terms.
3. At least two interactive exercises (like role-play or peer feedback).
4. A reflection task to help students apply what they learned.
   Make it engaging, practical, and student-friendly."*

Why this works:

- **Domain fit**: Education/training → focuses on structured learning.
- **Clarity**: Specifies time (30 mins), audience (college students), and topic (public speaking).
- **Engagement**: Includes activities, not just theory.
- **Outcome**: Helps learners practice and reflect.