

Assignment 06:

Assignment and practice of Prompt Engineering to craft effective prompts.

Task 1: Prompt Categorization

Categorize the following prompts into types (Instructional, Conversational, Visual, etc.) and explain your reasoning:

1. - 'Generate a logo for a tech startup using neon colors.'
 - a. Company name "Future Developers And Builders"
2. - 'Explain blockchain to a 5-year-old.'



Okay, imagine this:

You and your friends have a **magic notebook** 📖.

- Whenever someone gives or takes a toy, you **all write it down** in the notebook.
- But here's the rule: once you write something, **nobody can erase it**—not even with the world's strongest eraser!
- Everyone has a **copy of the notebook**, so if someone tries to cheat and say, "I never gave you my toy," the others can check their notebooks and say, "Nope, it's right here!"

👉 That magic notebook is like a **blockchain**:

- The pages are the **blocks**.
- The whole notebook is the **chain**.
- And because everyone has the same copy, it's **super hard to cheat**.

 You and 3 friends are playing with candies.

1. Trading starts:

- You give 2 candies to your friend.
- Everyone writes it down in their notebook:
“You gave 2 candies to Sam.”

2. More trades:

- Sam gives 1 candy to Mia.
- Again, everyone writes it down.

3. No erasing allowed:

- Once it's written, it **stays forever**. No one can cheat and say,
“I didn't give the candy!”

4. Copies everywhere:

- Each friend keeps their own notebook.
- If someone lies, the others can show the truth from their notebooks.

 That's blockchain:

- Every candy trade = a **transaction**.
- A notebook page = a **block**.
- All the pages stuck together = the **chain**.
- Everyone having a copy = it's **safe and fair**.
-

3 - 'You are a UX designer. Suggest improvements to this app layout.'

Thanks for sharing the app flow diagram 🙌

I see it's an **e-commerce checkout journey** (Shop → Product Page → Cart → Login/Guest → Checkout → Confirmation).

Here are my **UX improvement suggestions**:

Shop & Product Page

1. Product Discovery

- Add filters (price, category, rating) to reduce cognitive load.
- Include a **search bar** at the top for faster navigation.

2. Product Page

- Prioritize visuals: larger product image with zoom.
 - Use **sticky “Add to Cart” button** so users don't scroll to find it.
 - Show **reviews & ratings** right below the product description to build trust.
-

Cart (View Cart)

3. Cart Preview

- Add a **mini cart dropdown/side panel** so users don't always leave the product page.
- Show **estimated delivery & savings** (if discounts apply).

4. Cart Clarity

- Use **clear CTAs**:
 - “Continue Shopping” (secondary button).
 - “Checkout” (primary, high-contrast button).
 - Highlight the **total amount** at the top-right of the cart summary.
-

Sign In / Guest Checkout

5. Login Friction

- Don't force login. Default to **guest checkout** with a subtle option to log in.
 - Enable **social logins (Google, Facebook, Apple)** to reduce drop-offs.
-

Checkout Confirmation

6. Form Design

- Split long forms into **steps**: Shipping → Payment → Review.
- Use **address auto-complete** (Google Maps API).

7. Payment Options

- Show multiple payment methods upfront.
 - Default to **most used option (UPI/Wallet in India, Cards elsewhere)**.
-

Final Confirmation

8. Post-Purchase Trust

- Show **order summary + tracking info** clearly.
 - Add “**Continue Shopping**” CTA after confirmation.
 - Provide **customer support link/chatbot** in case of issues.
-

Visual & Interaction Design

- Maintain **consistent button styles** (primary = checkout, secondary = edit/update).
- Add **progress indicators** (Step 1 of 4 → Step 2 of 4, etc.) so users know where they are.
- Use **contrasting colors** for CTAs, but softer backgrounds for text-heavy areas.
- Mobile-first: Ensure cart and checkout are optimized for small screens.

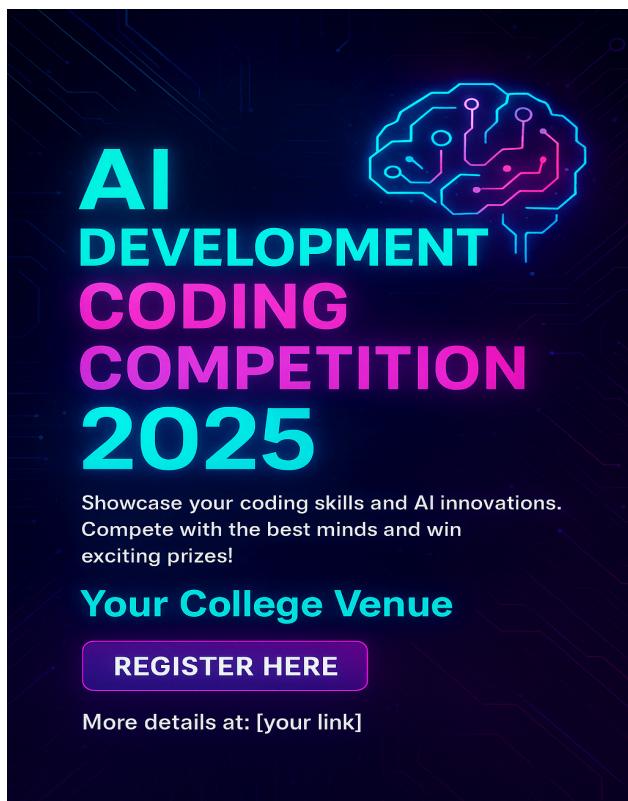
✨ In short:

- Reduce friction in checkout.
- Highlight clarity in cart.
- Make buttons & progress obvious.
- Provide trust signals (reviews, secure payments).

Task 2: Refinement Practice

Prompt for image creation :-

Generate a flyer for ai development coding competition with vibrant colours including AI relevancy and as a coordinator i wont it to be noticed by almost every clg student



Task 3: Prompt Design Exercise

Prompt (text based) :-

generate a short text para for application of ai at various fields such as education at schools and other fields

Result

Artificial Intelligence (AI) is rapidly transforming different sectors of our lives by bringing smarter, faster, and more efficient solutions to everyday challenges. In the field of education, AI is revolutionizing the way students learn and teachers teach. Schools are increasingly adopting AI-powered tools to provide personalized learning experiences tailored to each student's pace and ability. Intelligent tutoring systems and virtual assistants can help learners solve problems, clarify doubts instantly, and even recommend additional resources. Teachers also benefit from AI through automated grading, attendance tracking, and data-driven insights into student performance, which allows them to focus more on teaching and mentoring.

Beyond the classroom, AI applications extend to a wide range of industries. In healthcare, AI assists doctors in early disease detection, medical imaging, drug discovery, and personalized treatment plans, improving both speed and accuracy. In agriculture, AI-driven systems help monitor crop health, predict weather conditions, and optimize irrigation, leading to higher productivity and sustainability. Finance relies heavily on AI for fraud detection, risk analysis, and smart investment decisions. Meanwhile, industries use AI-powered automation and robotics for precision manufacturing, quality control, and predictive maintenance. Overall, AI is shaping a future where efficiency, accuracy, and accessibility become the norm across all fields

2. Prompt for DALLE image generation : generate a pic for showing stelite



3. Prompt for code generation :- generate c++ code logic for traffic signal scheduling

```
#include <iostream>
#include <thread> // for sleep_for
#include <chrono> // for seconds
```

```
using namespace std;
```

```
using namespace std::chrono;
```

```
void trafficSignal(string direction, int greenTime, int yellowTime, int redTime) {
```

```
    cout << "\n==== " << direction << " Signal Control ===\n";
```

```
// Green signal

cout << "GREEN light ON for " << greenTime << " seconds\n";
this_thread::sleep_for(seconds(greenTime));

// Yellow signal

cout << "YELLOW light ON for " << yellowTime << " seconds\n";
this_thread::sleep_for(seconds(yellowTime));

// Red signal

cout << "RED light ON for " << redTime << " seconds\n";
this_thread::sleep_for(seconds(redTime));

}

int main() {

    // Example timings (in seconds)

    int greenTime = 5;

    int yellowTime = 2;

    int redTime = 7;

    while (true) {
```

```
    trafficSignal("North-South", greenTime, yellowTime, redTime);

    trafficSignal("East-West", greenTime, yellowTime, redTime);

}

return 0;
}
```

4.Prompt for an ai educator bot:-

generate a tutorial for explaining llm to an ai srtudent in which u would educate as a friend

Result

Alright, let's do this like I'm sitting next to you in class, just two friends geeking out over AI. I'll explain **LLMs (Large Language Models)** step by step, so it doesn't feel like a lecture but more like a story you'll remember.

Tutorial: Understanding LLMs (Large Language Models)

1. Start with the Big Picture

Imagine you and I are chatting right now (which we are 😊). I'm giving you responses that *make sense* because I've read tons of books, articles, and conversations before.

That's exactly what an **LLM** does — it has read huge amounts of text

(the “training data”) and learned patterns of how words usually come together.

Think of it like a super-smart parrot  — but not just copying words. It has actually learned the *structure* of language, so it can create new sentences that sound natural.

2. What Makes It "Large"?

The “large” in **LLM** is about two things:

- **Massive training data** → billions of words, everything from Wikipedia to research papers.
- **Millions or billions of parameters** → these are like tiny knobs in a giant machine. The model tunes them during training to “understand” language.

So, the more parameters, the better it can capture complex relationships between words.

3. How Does It Work?

Let’s say I start a sentence:

“AI in the future will...”

Your brain already guesses: “*change jobs, help doctors, become powerful,*” etc.

An LLM does the same — it predicts the **next word** based on all the words before it. It's like playing the world's biggest "fill in the blanks" game 🎯.

This is powered by something called a **Transformer** (no, not Optimus Prime 🤖, but close). Transformers are neural networks that are great at paying attention to context — not just the last word, but the whole sentence/paragraph.

4. Why Is It So Powerful?

Because once a model learns language patterns, it can:

- **Answer questions** (like an encyclopedia that talks back).
- **Translate languages** (English → Hindi, French → Spanish, etc.).
- **Summarize text** (long article → short notes).
- **Generate code, poetry, or stories.**

Basically, if it's *language*, an LLM can handle it.

5. Where Do You Use It?

- **Chatbots** (like me 😎).
- **Search engines** (Google, Bing use them under the hood).

- **Healthcare** (summarizing patient records).
- **Education** (personalized tutors).

LLMs are not just about talking — they're becoming tools for every industry.

6. The Friend's Tip

Don't think of LLMs as "magical geniuses." They don't *think* like humans; they just **predict really well**. The real magic is how prediction at scale feels like understanding.

If you get this, you're already ahead of most beginners.

 So bro, next time someone asks you what an LLM is, just tell them:
 "*It's a machine trained on tons of text that learns to predict the next word. With enough data and parameters, it can write, answer, and even code like a human.*"

Would you like me to make this tutorial more **visual** (like a small diagram/flowchart of how an LLM predicts words), so you can use it in your class notes?