DAY - 1

INTRODUCTION TO PROMPT ENGINEERING

On the first day of training, we were introduced to the concept of Prompt Engineering, a key skill for effectively interacting with AI language models such as ChatGPT. In simple terms, prompt engineering is the art of giving clear and specific instructions to an AI to get the most useful and accurate responses.

We discussed how the structure and wording of a prompt can greatly influence the quality of the AI's output.

REAL-WORLD USE CASES

Prompt engineering is being widely adopted in various fields. Some practical applications we discussed include:

- Writing and editing emails and content
- Generating creative writing and educational materials
- Summarizing documents and extracting key points
- Providing customer service or virtual assistance
- Generating ideas or even debugging code

This showed us how valuable and versatile prompt engineering can be in both personal and professional settings.

ANATOMY OF A GREAT PROMPT

We learned that a good prompt has a clear structure. A well-designed prompt usually includes:

- A defined task
- Proper context
- Specific constraints (such as tone or length)
- A clear desired output format

This structure helps the AI understand exactly what is expected and respond accordingly.

•

TYPES OF PROMPTS

We explored several different types of prompts:

Instructional Prompt – Direct instructions to complete a task.

- Role-Based Prompt – Asking the AI to take on a specific role or persona.
- Few-Shot Prompt – Giving 1-2 examples before asking for a similar response.
- Zero-Shot Prompt – Providing no examples, just a well-structured instruction.
- Chain-of-Thought Prompt – Encouraging the AI to reason step-by-step.
- Constraint-Based Prompt – Setting rules such as word count or tone.
- Reframing Prompt – Changing the way a question is asked to get better results.

PROMPT WRITING FORMULAS

We also discussed prompt writing formulas that make it easier to structure tasks efficiently. Some common formulas included:

- Task + Context + Output Format
- Act as [Role] + Task + Constraints
- Using brackets and clear separators for multi-step instructions

This help streamline the prompt-writing process and ensure clarity.

ACTIVITY: PROMPT MASTER CHALLENGE

An engaging activity titled "Prompt Master Challenge" was conducted. We were given several small tasks and asked to create prompts for them. The tasks included:

- Writing an apology email for a late delivery
- Generating a blog intro for *"Morning routines for students"*
- Creating a quiz on *World War 2*
- Explaining AI to a 6th grader

I chose task (a) and my original prompt was:

“Actually, I want to write an apology email for a late delivery of a parcel related to clothes. Actually, I had to deliver it to the customer on 29th June 2025, but due to weather conditions I was not able to deliver it on time. The delivery was delayed by one day.”

This prompt was rated 8.5/10 by ChatGPT for clarity, specificity, and effectiveness. I then refined the prompt using the principles we had learned and compared the responses. This helped me understand how small changes in wording can improve the outcome significantly.

COMMON PROMPT MISTAKES

We discussed frequent mistakes made while crafting prompts:

- Being too vague or general
- Not defining the goal or expected format
- Overloading the prompt with too much information
- Ignoring the context or target audience

Avoiding these mistakes can lead to much better and usable results from the AI.

PRO-PROMPTING TIPS

Some advanced tips were shared to improve prompt performance:

- Use step-by-step format for complex tasks
- Ask for structured output (e.g., lists, paragraphs, tables)
- Assign roles to improve the tone and expertise
- Include examples when necessary

PROMPTING AS A CAREER SKILL

Prompt engineering is now being recognized as a 21st-century skill, especially valuable in careers related to:

- AI and Machine Learning
- Digital Marketing and Content Creation
- EdTech and E-learning
- Software Development and Automation

The ability to communicate effectively with AI systems is becoming increasingly important in both technical and non-technical fields.

•

KEY TAKEAWAYS

- Clear and well-structured prompts lead to more accurate responses.
- Understanding different prompt types allows more control over output.

  Practicing and refining prompts helps develop better communication with AI tools.

HOMEWORK

Two tasks were assigned as part of homework:

1. Good Prompt Table – Rewrite a set of poorly written prompts into better ones and identify the type of each.
2. Prompt Rewriting – Pick one of our previously used prompts, rewrite it using today's learning, and compare the results and ratings.

CONCLUSION

Day 1 of the training gave us a solid introduction to Prompt Engineering, including both theory and hands-on practice. From understanding prompt types to writing and refining our own, the session laid the foundation for developing strong AI interaction skills. I'm excited to apply this knowledge further in the upcoming sessions.

DAY - 2

INTRODUCTION TO GENERATIVE AI

On Day 2 of the training, we explored the concept of Generative Artificial Intelligence (Generative AI). We learned that Generative AI refers to a class of models that can create new content—such as text, images, audio, code, and more—based on the data they were trained on. These models are capable of learning patterns and generating human-like outputs, making them powerful tools for creativity, automation, and productivity.

Examples of generative outputs include:

- Text (e.g., essays, code, scripts)
- Images (e.g., artworks, visual designs)
- Audio (e.g., synthetic voices, music)
- Video (e.g., animation, scene generation)

UNDERSTANDING LLMs, DIFFUSION MODELS & MULTIMODAL MODELS

We were introduced to the basic architecture and function of the following foundational model types:

- **Large Language Models (LLMs):** These are deep learning models trained on massive amounts of text data to understand and generate human language. Examples include GPT (Generative Pre-trained Transformer), Claude, Gemini, and LLaMA.
- **Diffusion Models:** These are mainly used for image generation tasks. They start with random noise and gradually transform it into a coherent image. DALL·E by OpenAI uses similar mechanisms to create realistic images from text prompts.
- **Multimodal Models:** These models can process and combine multiple types of data inputs—such as text, images, and audio. Gemini and GPT-4o (Omni) are examples of such models, as they can handle text and visual inputs simultaneously.

TOOLS & PLATFORMS INTRODUCED

We explored several cutting-edge tools and platforms powered by generative AI. Each of these tools serves a specific purpose:

a. ChatGPT (OpenAI)

- Mainly used for document generation, summaries, ideation, and conversational AI.

- Model in use: GPT-4o (Omni) – a powerful, multimodal model capable of understanding and generating both text and images, with improved speed and cost efficiency.

b. Gemini (by Google DeepMind)

- A strong LLM with capabilities in reasoning, coding, and multimodal input.
- Model in use: Gemini 1.5 Flash – optimized for speed and cost, suitable for prompt engineering and real-time responses.

c. DALL·E

- Specialized in image generation from textual descriptions.
- Also supports inpainting (editing parts of an existing image).

d. Co-Pilot (GitHub + OpenAI)

- AI assistant designed for code suggestions and completion.
- Integrated into development environments like VS Code for real-time help. **e. CodeX**
- Developed by OpenAI for code generation and natural language-to-code transformation.
- Ideal for automating repetitive coding tasks.

f. Claude (Anthropic)

- A conversational AI known for safe, ethical, and nuanced responses.
- Built for high-quality reasoning, summarization, and assistance.

g. LLaMA (Meta AI)

- Lightweight open-source models for various tasks, suitable for local deployment.
- Popular in academic research and customized AI applications.

h. Whisper & Google Speech Recognition

- Whisper (OpenAI): A speech-to-text model capable of transcribing audio with high accuracy.
- Google Speech-to-Text: A robust, cloud-based API used for real-time voice recognition and transcription tasks.

CONCLUSION

Day 2 provided a well-rounded overview of foundational AI models and the tools built upon them. We not only learned how different architectures like LLMs and diffusion models function, but also gained hands-on awareness of the tools that leverage these technologies in the real world.

The exposure to models such as GPT-4o, Gemini 1.5 Flash, and tools like DALL·E and Whisper gives us a strong starting point for building creative, technical, and practical applications powered by generative AI.

DAY – 3

UNDERSTANDING LARGE LANGUAGE MODELS (LLMs)

On Day 3, we delved deeper into the core of generative AI by understanding Large Language Models (LLMs) — the driving force behind tools like ChatGPT, Claude, Gemini, and others. A Large Language Model is an artificial intelligence model trained on massive datasets to understand and generate human-like language. These models use statistical patterns to perform tasks like translation, summarization, reasoning, and content creation.

We learned that LLMs work by predicting the next word in a sentence, using context and prior knowledge acquired during training.

KEY TERMINOLOGIES

Several important terms were introduced, which are essential to understand the working of LLMs:

- **Token:** A piece of text (word, sub-word, or character) used as input for processing by the model.
- **Parameter:** The weights or values learned during training, which influence how the model responds to a prompt. Modern LLMs have billions of parameters.
- **Prompt:** The input provided to the model to guide its output.
- **Fine-Tuning:** The process of training a pre-trained model further on a specific dataset to tailor its behaviour.
- **Inference:** The phase where a trained model generates output based on new input.

ARCHITECTURE OF A LLM – WHAT'S INSIDE A TRANSFORMER

We studied the inner workings of a Transformer, the foundational architecture behind most LLMs. Key components include:

- **Self-Attention Mechanism:** Allows the model to focus on different words in a sentence relative to each other, enhancing understanding of context.
- **Feed-Forward Neural Networks:** These layers help in transforming the attentionbased representations into meaningful output.
- **Positional Encoding:** Since transformers do not process tokens in order, positional encoding provides information about the position of each token in the sequence.

This architecture is the backbone of advanced LLMs like GPT-4, Gemini, and Claude.

TRAINING OF LLMs

We learned about the three key stages involved in training LLMs:

a. Pre-Training:

The initial phase where the model is trained on a vast corpus of general text data. The goal is to learn language structure and general knowledge.

b. Fine-Tuning:

A more focused training phase on specific data or tasks to customize the model's behavior.

c. RLHF (Reinforcement Learning with Human Feedback):

This method involves human reviewers scoring the model's outputs to help it learn preferences, reduce bias, and align better with human values.

APPLICATIONS OF LLMs

LLMs are powering a wide range of applications, such as:

- Virtual assistants and chatbots
- Document summarization
- Code generation and completion
- Language translation
- Creative writing and ideation
- Medical and legal document analysis

LIMITATIONS OF LLMs

While powerful, LLMs come with certain limitations:

- They can generate inaccurate or fabricated information (hallucinations).
- They may not retain context over long conversations.
- High computational requirements make them expensive to train and deploy.
- Their outputs can be sensitive to prompt phrasing.

ETHICAL CONCERNS

We also touched upon the ethical implications of using LLMs:

- **Bias and Fairness:** Models may inherit biases from their training data.
- **Misinformation:** Risk of spreading false information or fake news.
- **Privacy:** Potential to leak personal or sensitive information if trained on unfiltered data.
- **Over-Reliance:** Users may become overly dependent on AI-generated content without verifying accuracy.

CONCLUSION

Day 3 provided an insightful understanding of the technical foundations of LLMs, from their architecture and training to their capabilities and limitations. This session laid a solid theoretical base for working more effectively with AI tools and prompts, helping us become more conscious of both the power and responsibility that come with using such technologies.

DAY – 4

CHATGPT PLAYGROUND – OVERVIEW & FEATURES

On Day 4, we explored the ChatGPT Playground, a flexible environment provided by OpenAI that allows users to experiment with prompts, adjust model behaviour, and generate various types of outputs. The Playground acts as a sandbox for testing and refining prompts in a more controlled, customized setting.

We studied several important configuration options within the Playground:

- **Temperature:** Controls the creativity or randomness of the response. Lower values make the output more focused and deterministic, while higher values make it more diverse and imaginative.
- **Max Tokens:** Sets the limit for the length of the model's response.
- **Tools:** Includes functions like code generation, data analysis, browsing, DALL·E image generation, and file handling.
- **Output Formats:** We also learned how ChatGPT can return output in multiple structured forms such as:
  - JSON object
  - JSON schema
- **Plain text**

  Step-by-step explanations

This session helped us understand how prompt configuration affects the outcome, especially in technical or structured tasks.

SYSTEM PROMPT VS USER PROMPT

We studied the distinction between two fundamental components of AI-based communication:

- **System Prompt:** Sets the behaviour or persona of the AI (e.g., “You are a helpful assistant” or “You are an expert Python developer”).
- **User Prompt:** The actual input or query from the user (e.g., “Generate a Python program for the Fibonacci series”).

This separation is important as it allows us to control not just the *task* but also *how* the AI responds.

SHIFT TO GEMINI AI STUDIO

Since the ChatGPT Playground is a paid service, we transitioned to using Google's Gemini AI Studio, which provides similar functionality for free. The Gemini AI Studio supports both text and multimodal inputs and gives us the ability to define system roles and test prompts with different configurations.

We explored it thoroughly and carried out structured experiments to understand the effect of temperature and format.

PRACTICAL EXPERIMENTS ON GEMINI STUDIO

We performed a task using different prompt types and configurations:

Scenario 1:

- **System Prompt:** You are an expert Python developer.
- **User Prompt:** Provide me the code of Fibonacci series and also explain me in step-by -step manner.
- **Format:** Code Execution
- **Temperature:** 1
- **Response Time:** ~40 seconds

Scenario 2:

- **User Prompt (modified):** Provide me the code to find the Fibonacci series and also explain me in step-by-step manner.
- **Format:** Structured
- **Temperature:** 1.5
- **Response Time:** ~15 seconds

These experiments gave us a practical understanding of:

- How temperature affects creativity and response length.
- How structured formats influence clarity and response speed.
- How a well-defined system prompt helps improve technical accuracy.

CONCLUSION

Day 4 was a **hands-on and highly practical** session, offering a deeper insight into how prompt-based AI systems can be tuned using platforms like ChatGPT Playground and Gemini

AI Studio. We not only learned about model configuration and prompt structuring, but also experienced the **impact of prompt design** on speed, output style, and usefulness.

The understanding of system prompts, user prompts, and temperature settings will be especially valuable as we move into more advanced prompt engineering techniques in the upcoming days.

DAY – 5

BUILDING A WEBPAGE SUMMARIZER USING PYTHON

On Day 5, we transitioned from prompt writing and AI platforms into practical implementation using Python. The session was focused on how to integrate Generative AI capabilities into coding projects using APIs and external libraries.

The first task was to build a **Website Summarizer Tool** that takes a webpage URL as input and generates a markdown-formatted summary. The core steps included:

1. **Generating API Key from Google AI Studio:**
   We began by generating a working API key to access generative capabilities from Google's AI services.
2. **Coding the Summarizer:**
   - We wrote a Python program that accepts a dynamic user input (i.e., the URL of any website).
   - Using the Beautiful Soup 4 library, we extracted all the internal and external links from the page.
   - The webpage content was then summarized using the API, and the output was structured in Markdown format.
   - The summarized result was saved as a file to the system for future access and documentation.

This hands-on project helped solidify our understanding of how to integrate AI-generated summaries into real-world applications, especially useful for content researchers, bloggers, and analysts.

IMAGE GENERATION USING API – HUGGING FACE & STABILITY AI

The second task involved dynamic image generation using external APIs:

- We generated API keys for Hugging Face and Stability AI.
- The Python code was written in such a way that users can input prompts dynamically, and based on the prompt, the AI will generate an image.
- The image response was decoded using the Base64 library to retrieve and display the generated visual content.

This task demonstrated how AI can respond not only with text but also visual content, opening doors to creative applications such as content design, marketing automation, and digital art tools.

HOMEWORK TASK

For the day's homework, we were asked to enhance the website summarizer in the following way:

- Reframe the summarized content into a new format using typewriter-style output.
- Store the reframed output in a separate file, different from the original summary.
- Make sure the content appears as if it was typed manually—possibly adding visual structure or delays to simulate the behaviour of a typewriter.

This extension challenged us to combine text formatting, file handling, and creative output simulation in Python.

CONCLUSION

Day 5 was a highly technical and productive day that focused on the real-world application of AI APIs through Python. We not only learned to build functional tools such as a web summarizer and dynamic image generator, but also gained confidence in working with libraries like Beautiful Soup and Base64, and integrating external AI services through API keys.

This hands-on coding experience provided a strong foundation in how AI can be embedded in custom software projects, making it a valuable learning milestone in our training journey.

DAY – 6

FAMILIARIZATION WITH GOOGLE AI STUDIO

On Day 6, the focus of our training was to explore and understand the various functionalities and capabilities of Google AI Studio, a powerful platform that enables developers and learners to build, test, and deploy AI models interactively. This session was essential for bridging the gap between theoretical prompt design and its practical implementation in real-world tools.

KEY FEATURES AND OBSERVATIONS

1. FREELY ACCESSIBLE

We were introduced to Google AI Studio's major advantage—it is freely available to all users, making it an ideal choice for students, developers, and educators looking to experiment with AI capabilities without subscription fees.

2. MODEL COMPARISON MODE

We explored the ability to toggle between two major models:

- **Gemini 1.5 Flash** – Fast, lightweight, and cost-effective.
- **Gemini 1.5 Pro** – More powerful and capable of deeper reasoning and multimodal understanding.

The comparison feature helps evaluate the difference in speed, output quality, and interpretation between the models for the same prompt.

3. GOOGLE DRIVE INTEGRATION

Every project and asset—whether input data, prompt history, or generated output—can be seamlessly saved and retrieved from Google Drive, enabling easier collaboration and cloudbased management.

4. GROUNDING WITH GOOGLE SEARCH

A very impactful feature is its ability to ground answers with live web search, making the output more current, reliable, and accurate, especially for factual or real-time questions.

5. AUDIO CAPABILITIES

We tested the speech recognition and audio input/output features, allowing us to interact using voice, and also generate voice outputs for AI-generated content.

6. SCREEN SHARING

The platform supports screen sharing, which is highly useful in collaborative development and during walkthroughs or demonstrations.

7. WEBCAM INTEGRATION

Users can also integrate their webcam feed into the project, allowing for more immersive and interactive multimodal experiences.

8. EMPTY PROJECT INITIALIZATION

We learned how to create blank (empty) projects and build from scratch by defining system roles, setting response formats, and linking APIs or media inputs.

9. API KEY MANAGEMENT

Google AI Studio allows you to generate and manage multiple API keys, which are required to authenticate and access the model via different applications or environments.

10. SYSTEM PROMPT VS USER PROMPT

We revisited the distinction:

- **System Prompt** sets the model's behaviour or role (e.g., “You are a helpful language tutor.”)
- **User Prompt** is the actual instruction given by the user (e.g., “Summarize this article in simple terms.”)

Understanding and configuring both properly helps guide the model effectively.

11. TEXT, IMAGE, AUDIO, AND VIDEO INTERACTIONS

The platform supports:

- **Generating** content (e.g., writing, art, sound, video)
- **Analysing** uploaded media, such as:
    - What an image depicts
    - What is happening in a video
    - Transcribing and understanding audio clips

This multimodal capability makes it a powerful tool for both content creation and understanding.

12. STREAMING RESPONSES

We learned about the streaming option, which lets you view the AI’s response in real-time as it's being generated. This is helpful for understanding the reasoning process and improving user interaction.

13. SDK AVAILABILITY

Google provides an SDK (Software Development Kit) for developers to integrate Gemini into their own apps and platforms. This opens opportunities for building AI-powered applications and workflows.

CONCLUSION

Day 6 was an exploratory and application-rich session. We gained hands-on experience with the powerful features of Google AI Studio, including working with different model types, handling multimodal inputs and outputs, managing projects via Google Drive, and interacting with real-time web-grounded AI responses.

This knowledge not only strengthens our practical skills but also sets the stage for building and deploying AI-driven applications across various domains such as education, content generation, and automation.

DAY – 7

SPEECH-TO-TEXT CONVERSION USING WHISPER API

On Day 7, we shifted focus to the integration of audio input with AI, by learning how to develop a Speech-to-Text Conversion System using Python. The objective was to capture user speech through a microphone and convert it accurately into text using the Whisper model, provided via Hugging Face's API.

This project helped us understand the practical application of voice recognition, and how speech can be transcribed dynamically using AI.

DEPENDENCIES USED

We installed and implemented the following Python libraries:

- **sound device** – to access the system microphone and record real-time audio.
- **SciPy** – for handling and saving audio data.
- **Ip widgets** – for creating an interactive user interface in Jupyter Notebooks.

These tools helped us build a flexible and interactive environment for recording and analyzing voice inputs.

TWO RECORDING MODES IMPLEMENTED

We developed the functionality to handle two different modes of speech recording:

a. Record for a Fixed Duration

- The user sets a predefined duration (e.g., 5 seconds), speaks into the microphone, and the tool automatically stops recording after the time ends.

b. Record Until Stop (8-Second Timeout)

- The system listens for a maximum of 8 seconds or until silence is detected.
- This approach is more natural for open-ended speech and spontaneous inputs.

SPEECH TRANSCRIPTION USING WHISPER MODEL

After recording, the audio was passed to Whisper, a robust speech recognition model developed by OpenAI and integrated using an API key from Hugging Face.

Whisper returned the transcribed text based on the audio input, and we then evaluated:

- How accurately it captured spoken words