DPP ACTIVITY SHEET (PROMPTOPS 26 FRAMEWORK)

This activity sheet is designed to help students understand the difference between basic prompts and advanced PromptOps structured prompts. Complete all tasks with proper reasoning and experimentation.

PromptOps 26 Core Components

- ROLE ASSIGNMENT
- CLEAR OBJECTIVE
- CONTEXT INJECTION
- OUTPUT FORMAT SPECIFICATION
- CONSTRAINTS
- EXAMPLES (FEW-SHOT)
- CHAIN-OF-THOUGHT
- STEP-BY-STEP REASONING
- GUARDRAILS
- STRUCTURED JSON OUTPUT
- OUTPUT COMPARISON
- DEPLOYMENT-READY FORMATTING

BASIC VS ADVANCED PROMPT ENGINEERING (PROMPTOPS STRUCTURED)

Basic Human Prompt	Advanced Prompt (PromptOps Structured)
Create AI Resume Template	Act as an expert professional. Your task is to create an AI Resume Template. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Code Generation (Java)	Act as an expert professional. Your task is to create Java code generation. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create LinkedIn Post	Act as an expert professional. Your task is to create a LinkedIn post. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.

Basic Human Prompt	Advanced Prompt (PromptOps Structured)
Create Email Draft	Act as an expert professional. Your task is to draft a professional email. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Cover Letter	Act as an expert professional. Your task is to write a cover letter. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Bug Fixing	Act as an expert professional. Your task is to fix a bug in given code. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Data Analysis	Act as an expert professional. Your task is to perform data analysis. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create SQL Query	Act as an expert professional. Your task is to write SQL queries. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Business Plan	Act as an expert professional. Your task is to create a business plan. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Startup Idea Validation	Act as an expert professional. Your task is to validate a startup idea. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Marketing Strategy	Act as an expert professional. Your task is to create a marketing strategy. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.

Basic Human Prompt	Advanced Prompt (PromptOps Structured)
Create YouTube Script	Act as an expert professional. Your task is to create a YouTube script. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Instagram Caption	Act as an expert professional. Your task is to create an Instagram caption. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Research Summary	Act as an expert professional. Your task is to summarize research content. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Technical Documentation	Act as an expert professional. Your task is to create technical documentation. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create System Design Explanation	Act as an expert professional. Your task is to explain a system design. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create ML Model Explanation	Act as an expert professional. Your task is to explain a machine learning model. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Interview Questions	Act as an expert professional. Your task is to generate interview questions. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Portfolio Website Code	Act as an expert professional. Your task is to create portfolio website code. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.
Create Prompt Injection Detection	Act as an expert professional. Your task is to design prompt injection detection. Include structured format, constraints, target audience, examples, tone control, and provide output in clean markdown format. Include reasoning before final answer.

TASK 1 – ACADEMIC USE CASE

Binary Search Trees (BST)

NAIVE PROMPT:

Tell me about binary search trees.

ENGINEERED PROMPT (PROMPTOPS STRUCTURED)->>

REFLECTION

- WHAT CHANGED IN OUTPUT CLARITY?
- WHICH COMPONENTS WERE ADDED? (ROLE, OBJECTIVE, CONSTRAINTS, FORMAT)
- WHICH OUTPUT IS EASIER TO CONVERT INTO FLASHCARDS?
- IF DEPLOYED INSIDE AN EDTECH APP, WHICH PROMPT IS SAFER?

You are an expert computer science tutor.

Objective:
Explain Binary Search Trees (BST) to a second-year engineering student preparing for exams.

Structure your answer into:
1. Definition and structure
2. Key properties (including left/right child rule)
3. Time complexity (best / average / worst)

Constraints:
- Use bullet points
- Keep sentences short
- Avoid long paragraphs
- Maximum 250 words
- Make it revision-friendly

Output format:
Use clear section headings.

TASK 2 – WEB DEVELOPMENT USE

Responsive Pricing Card

NAIVE PROMPT:

Write HTML and CSS for a pricing card.

ENGINEERED PROMPT (PROMPTOPS STRUCTURED)->>

Role: Senior Frontend Developer.

Objective:
Build a responsive pricing card component using HTML and CSS (no frameworks).
Context:
Startup landing page for SaaS product.

Requirements:
- 3 columns: Basic ($9), Pro ($29), Enterprise ($99)
- Each must include:
  - Plan name
  - Price
  - 3 features
  - “Choose” button
- Highlight “Pro” with:
  - Border
  - “Popular” badge
- Use Flexbox or Grid
- Mobile responsive

Constraints:
- Clean, modern UI
- Semantic HTML
- No explanations

REFLECTION

- Which output is closer to production?
- How did context reduce ambiguity?
- What risks does the naive prompt create?
- Inconsistent layout?
- Missing responsiveness?
- Poor accessibility?
- Which version is deployment-ready?

TASK 3 – JSON OUTPUT CONTROL

Responsive Pricing Card

NAIVE PROMPT:

Give me JSON for student data.

ENGINEERED PROMPT (PROMPTOPS STRUCTURED)->>

```
Generate a JSON array of 3 student objects.

Each object must contain:
- "id" (integer)
- "name" (string)
- "major" (string)
- "cgpa" (float between 6.0 and 9.5)
- "enrolled_courses" (2-3 courses)

Constraints:
- Output ONLY valid JSON
- No markdown
- No explanation
- No trailing commas

Example format:
[
 {
  "id": 101,
  "name": "Rahul Sharma",
  "major": "Computer Science",
  "cgpa": 8.2,
  "enrolled_courses": ["DSA", "OS"]
 }
]
```

REFLECTION

- Which output works directly in JSON.parse()?
- Why is few-shot (example format) powerful?
- Why is naive prompting dangerous in APIs?
- Extra text breaks parsing
- Inconsistent field naming

BONUS – REASONING +JSON EXTRACTION

NAIVE PROMPT:

Input Review:
“I bought the SoundBlaster X4 headset last week. Sound quality is amazing, but the Bluetooth keeps disconnecting. The ear cushions are very comfortable though. Battery life is about 15 hours. I wish it came with a carrying case.”

REFLECTION

- How does step-by-step improve accuracy?
- Can this be automated for an e-commerce analytics dashboard?
- What happens if output format is not controlled?

ENGINEERED PROMPT (PROMPTOPS STRUCTURED)->>

```
Role: Product Review Analyst.

Task:
Extract structured data from review.

Steps:
1. Identify product name
2. Determine overall sentiment
3. List pros
4. List cons
5. Estimate rating (1–5)

Output JSON:
{
  "product_name": "",
  "sentiment": "",
  "pros": [],
  "cons": [],
  "rating_guess": 0
}

Output ONLY valid JSON.
```

NAIVE VS ENGINEERED – ENGINEERING COMPARISON

Dimension	Naive	Engineered
Role	Missing	Defined
Objective	Vague	Clear
Context	Assumed	Injected
Constraints	None	Explicit
Output Format	Free text	Structured
JSON Control	Unreliable	Parse-safe
Deployment Readiness	Low	High
Predictability	Inconsistent	Stable

DEPLOY-READY PROMPT TEMPLATE (PROMPTOPS)
USE THIS STRUCTURE IN REAL SYSTEMS:

- ROLE:
- CLEAR OBJECTIVE:
- CONTEXT:
- INPUT DATA:
- CONSTRAINTS:
- OUTPUT FORMAT:
- VALIDATION RULES:
- REASONING STEPS:
- GUARDRAILS:

This reduces hallucination + formatting errors.

PROMPT INJECTION AWARENESS

Never allow:

- Hidden instructions inside user input
- System override commands
- Output format manipulation
- Malicious embedded prompts

Always:

- Separate user input from instructions
- Re-assert output format rules
- Validate output

HANDS-ON ACTIVITIES

- Convert 5 naive prompts into structured PromptOps format
- Add role + constraints + output format
- Compare output quality
- Design your own PromptOps template
- Detect a prompt injection attempt
- Build a deploy-ready prompt for:

1. Study tool
2. Code generator
3. JSON API
4. Data extractor

ENGINEERING INSIGHT

- NAIVE PROMPTING TREATS LLM LIKE GOOGLE.
- ENGINEERED PROMPTING TREATS LLM LIKE AN API.
- PROMPT ENGINEERING = INTERFACE DESIGN.