DAY 1 — Generative AI Foundations + Insight Agent

---

1. Understanding Generative AI

Subtopics

1. What is Generative AI?
2. LLMs (Large Language Models) explained simply
3. What GenAI can do (practical use cases)
4. How LLMs interpret text
5. Limitations & risks (in simple terms)

Groups Benefitted

- **Non-Technical (Primary):** ⭐⭐⭐⭐⭐
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐

---

2. Basic Prompting

Subtopics

1. What is a prompt?
2. Structure of a good prompt
3. Writing clear instructions
4. Summaries, rewrites, insights
5. Templates for daily use
6. Fixing poor prompts (debugging)

Groups Benefitted

- **Non-Technical (Primary):** ⭐⭐⭐⭐⭐
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers:** ⭐⭐

---

3. Intermediate Prompting

Subtopics

1. Extracting insights
2. Identifying issues
3. Asking GenAI to structure information
4. Using GenAI for text → action items
5. Creating reusable prompt flows

Groups Benefitted

- **Non-Technical (Primary):** ⭐⭐⭐⭐⭐
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐

---

4. Multi-Step Insight Extraction

Subtopics

1. Summary → insights → issues → actions
2. Prompt chaining (no coding)
3. Structured output (tables, bullets, checklists)
4. Multi-step reasoning through prompts
5. Designing insight frameworks

Groups Benefitted

- **Non-Technical:** ⭐⭐
- **Techno-Functional (Primary):** ⭐⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐

---

5. Insight Agent Architecture

Subtopics

1. What is an Insight Agent?
2. LLM reasoning workflow
3. Understanding Tools (Parser, Insight Generator)
4. JSON-mode outputs
5. Creating an automated Insight system

Groups Benefitted

- **Non-Technical:** ⭐ (high-level understanding)

- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers (Primary):** ⭐⭐⭐⭐⭐

---

6. Hands-On Build: Insight Agent API

Subtopics

1. LangChain basics
2. Creating insight tools
3. Structured output parsing
4. Creating `/insight` API endpoint
5. Testing & validating results

Groups Benefitted

- **Non-Technical:** — (not required)
- **Techno-Functional:** ⭐⭐ (observe for awareness)
- **Developers (Primary):** ⭐⭐⭐⭐⭐

DAY 2 — Communication & Automation Agent

---

1. Natural Language Understanding for Everyone

Subtopics

1. How AI understands emotions
2. Tone analysis
3. Intent detection
4. Message simplification

Groups Benefitted

- **Non-Technical (Primary):** ⭐⭐⭐⭐⭐
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐

---

2. Text Rewriting & Communication Generation

Subtopics

1. Making text clearer
2. Making messages professional
3. Creating polite responses
4. Fixing grammar, tone, clarity
5. Response templates (copy/paste)

Groups Benefitted

- **Non-Technical (Primary):** ⭐⭐⭐⭐⭐
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers:** ⭐⭐

---

3. Classification & Prioritization

Subtopics

1. Categorizing text
2. Priority scoring
3. Extracting tasks
4. When to escalate
5. Structured JSON outputs

Groups Benefitted

- **Non-Technical:** ⭐⭐
- **Techno-Functional (Primary):** ⭐⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐⭐

---

4. Communication Workflow Design

Subtopics

1. Message → summary → priority → reply → task
2. Automated decision flow
3. Multi-step prompting logic
4. Action extraction

Groups Benefitted

- **Non-Technical:** ⭐⭐
- **Techno-Functional (Primary):** ⭐⭐⭐⭐⭐

- **Developers:** ⭐⭐⭐

---

5. Building Communication Agent

Subtopics

1. Agent tools
2. ClassifierTool
3. ReplyGenerationTool
4. TaskExtractionTool
5. `/communication-agent` API

Groups Benefitted

- **Non-Technical: —**
- **Techno-Functional:** ⭐⭐
- **Developers (Primary):** ⭐⭐⭐⭐⭐

DAY 3 — RAG (Retrieval Augmented Generation) Knowledge System

---

1. RAG for Beginners (NT First)

Subtopics

1. Why LLMs need context
2. What is RAG in simple language
3. How RAG improves accuracy
4. Ask questions → get answers → show citations

Groups Benefitted

- **Non-Technical (Primary):** ⭐⭐⭐⭐⭐
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐

---

2. Document Preparation (TF Focus)

Subtopics

1. PDF → text
2. Cleaning content
3. Chunking data
4. Metadata importance

Groups Benefitted

- **Non-Technical:** ⭐⭐
- **Techno-Functional (Primary):** ⭐⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐⭐

---

3. Embeddings (TF + DEV Focus)

Subtopics

1. What are embeddings?
2. Semantic meaning
3. Similarity search

Groups Benefitted

- **Non-Technical:** ⭐ (basic awareness)
- **Techno-Functional:** ⭐⭐⭐⭐
- **Developers (Primary):** ⭐⭐⭐⭐⭐

---

4. RAG Pipeline Logic (TF + DEV)

Subtopics

1. Retrieve relevant chunks
2. Construct context
3. Generate grounded answer
4. Provide citation sources

Groups Benefitted

- **Non-Technical:** ⭐⭐ (conceptual)
- **Techno-Functional (Primary):** ⭐⭐⭐⭐⭐
- **Developers:** ⭐⭐⭐⭐⭐

---

5. Build & Deploy RAG API (Developer Track)

Subtopics

1. Vector DB configuration
2. Retriever + LLM Builder
3. `/rag/query` endpoint
4. Accuracy testing
5. Guardrails

Groups Benefitted

- **Non-Technical: —**
- **Techno-Functional:** ⭐⭐
- **Developers (Primary):** ⭐⭐⭐⭐⭐

Section A - Prompt

---

1. What is a Prompt?

A **prompt** is the instruction you give to an AI model.
It tells the model:

- **What you want**
- **How you want it**
- **In what format**
- **With what style, tone, or constraints**

Think of it as **talking to a very smart intern**:
The clearer your instructions, the better the output.

Examples:

❌ *Bad prompt:*
"Explain AI."

✅ *Good prompt:*
"Explain Generative AI in simple terms for first-year students, with 3 examples and a relatable analogy."

---

2. Structure of a Good Prompt

A powerful prompt usually contains these sections:

A. Role / Persona

Tell the AI *who* it should act as.
Example:
"You are a senior UX designer."
"You are a Python instructor."

B. Task

What exactly should the AI do?
Example:
"Create a 1-page explanation…"

C. Context

Background information the AI must know.
Examples:
“The audience is non-technical.”
“The company sells HR software.”

D. Constraints / Rules

Define limits, formatting, tone, word count, etc.
Examples:

- “Keep it within 150 words.”
- “Use simple English.”
- “Return output in bullet points.”

E. Output Format

Decide how you want the answer delivered.
Examples:

- JSON
- Table
- Bullet points
- Step-by-step

Prompt Skeleton (Reusable Template):

```
You are a [role]. Your task is to [objective].
Context: [background].
Constraints: [tone, style, rules].
Output format: [table/JSON/bullets].
```

---

3. Writing Clear Instructions

To get the best output:

✓ Be specific

Instead of: “Write a report,”
Say: “Write a 300-word report summarizing the meeting in formal tone.”

✓ Avoid ambiguity

Use numbered points, lists, constraints.

✓ Define the boundaries

Example:
“Do NOT include technical jargon.”

✓ Provide examples

AI learns from patterns.

✓ Break complex tasks

Use step-by-step instruction blocks.

✓ Give corrections

If you don’t like the answer, refine the instruction:
“Rewrite it with simpler language.”

---

4. Summaries, Rewrites, Insights

These are the most common everyday GenAI tasks:

A. Summaries

AI can condense long content while keeping meaning intact.

Examples:

- “Summarize this text in 5 bullet points.”
- “Convert this into an executive summary.”

B. Rewrites

AI can rewrite content for:

- Tone (formal, friendly, persuasive)
- Audience (students, executives)
- Platform (email, LinkedIn, presentation)

Example:
“Rewrite this paragraph for a CEO in a confident and concise tone.”

C. Insights

AI can extract hidden outcomes, patterns, or meaning.

Examples:

- “Extract the key insights.”
- “Highlight risks and opportunities.”
- “What decisions can be made based on this?”

---

5. Templates for Daily Use

Useful prompt templates for every role:

Email Writing

```
Rewrite this email to sound professional and concise:
[PASTE EMAIL]
```

Meeting Summary

```
Summarize this meeting into action items, decisions, and risks.
```

Presentation Slide Content

```
Convert this into a 3-slide summary with bullets:
[TEXT]
```

Brainstorming

```
Give me 10 creative ideas for ______ targeting ______.
```

Research

```
Explain this concept in simple terms with an example:
[TOPIC]
```

JSON Output

```
Return the information in JSON format with fields:
title, summary, key_points, actions.
```

Role-based Expert Help

```
Act as a senior data architect and explain how to design a pipeline for
this use case.
```

---

6. Fixing Poor Prompts (Debugging)

If the output is not good, check for common issues:

❌ Not enough context

Fix: Add details, examples, audience, industry.

❌ Vague task

Fix: Make the task specific and measurable.

❌ Wrong format

Fix: Define the format clearly (“return in table format”).

❌ Missing constraints

Fix: Add tone, length, perspective, rules.

Debugging Template

Use this whenever output feels wrong:

```
The response is not what I expected.
Rewrite it using these corrections:
1. [Correction 1]
2. [Correction 2]
3. [Correction 3]
Keep the output in [format].
```

Self-check Prompt

```
Review your own answer and improve clarity, accuracy, and structure.
Return the revised version only.
```

Section B - INTERMEDIATE PROMPTING

---

1. Extracting Insights

GenAI can read long text, meetings, reports, customer feedback, or emails and extract **hidden meaning**, **patterns**, and **signals**.

What "insights" mean

Insights are:

- Conclusions not directly written but implied
- Trends or patterns across multiple sentences
- Meaningful signals that support decision-making

Examples of what GenAI can extract

- Customer pain points
- Productivity blockers
- Bottlenecks in a workflow
- Reasons behind delays
- Opportunities for improvement
- Root causes buried inside long text

Example Prompt

```
Extract the top 5 insights from the text below.
Focus on patterns, pain points, and underlying reasons.
Return in bullet format.

[PASTE TEXT]
```

Output You Can Expect

- "Customers are unhappy with delivery delays."
- "Teams lack visibility on project timelines."
- "Most issues arise due to missing documentation."

---

2. Identifying Issues

GenAI can scan text and quickly identify **problems**, **risks**, **gaps**, or **inconsistencies**.

Types of issues GenAI can find

- Operational issues
- Missing information
- Conflicting data
- Duplication
- Errors in logic or process
- Risks and blockers
- Compliance gaps

Example Prompt

```
Identify issues, risks, and missing information in the text below.
Categorize them into: operational, strategic, communication, and data
issues.

[TEXT]
```

Possible Output

- Operational Issue: “No clear ownership of tasks.”
- Communication Issue: “Stakeholders are informed late.”
- Data Issue: “Metrics mentioned without sources.”

---

3. Asking GenAI to Structure Information

Unstructured text → structured output
This is one of the strongest use cases for GenAI across all job roles.

What GenAI can structure

- Emails → tables
- Reports → summaries
- Notes → sections
- Requirements → user stories
- Policies → checklists
- Long paragraphs → bullet points
- Meeting transcription → action items & decisions

Common Structure Formats

- Bullet points
- Tables
- JSON
- Checklists
- Flowcharts
- Step-by-step procedures
- SOP format

Example Prompt

```
Structure the following content into:
1. Summary
2. Key Points
3. Action Items
4. Risks
5. Open Questions

[TEXT]
```

Why this is powerful

- Faster reading
- Better communication
- Easier auditing
- Quick decision-making
- Saves hours of manual formatting

---

4. Using GenAI for Text → Action Items

GenAI can convert **confusing, unstructured** text into **precise, actionable tasks**.

What GenAI can do

- Convert meeting notes into action items
- Turn project updates into next steps
- Identify tasks from emails
- Generate owner + deadline automatically
- Assign task priority
- Convert problem descriptions into solutions

Key Components of Action Items

1. **Task** — What must be done
2. **Owner** — Who should do it
3. **Deadline** — By when
4. **Priority** — High/Medium/Low

Example Prompt

```
Convert the below conversation into clear action items.
Include: task, owner, deadline, and priority.

[TEXT]
```

Example Output

- **Task:** Update API documentation
  Owner: Sam
  Deadline: Tomorrow EOD
  Priority: High

---

5. Creating Reusable Prompt Flows

A **prompt flow** is a reusable sequence of prompts that creates a consistent, reliable outcome — like a mini-workflow powered by GenAI.

Why Prompt Flows Matter

- Save time
- Fix quality inconsistencies
- Standardize team output
- Automate repetitive work
- Increase accuracy
- Reduce thinking effort

Common Prompt Flow Examples

1. **Email → Summary → Action Items → Final Draft**
2. **Raw Notes → Structuring → Insights → Risks → Report**
3. **Customer Review → Sentiment → Root Cause → Recommendation**
4. **Idea → Expansion → Prioritization → Roadmap**
5. **Meeting Transcript → Key Points → Stakeholder Summary → MOM**

Example of a Reusable Prompt Flow

Step 1 — Summarize

```
Summarize the text below in 5 bullet points.
```

Step 2 — Extract Issues

```
From the summary, identify issues and their root causes.
```

Step 3 — Suggest Solutions

```
Recommend practical solutions for each issue.
```

Step 4 — Generate Action Items

```
Convert solutions into actionable tasks with owner, deadline, and priority.
```

Step 5 — Final Formatting

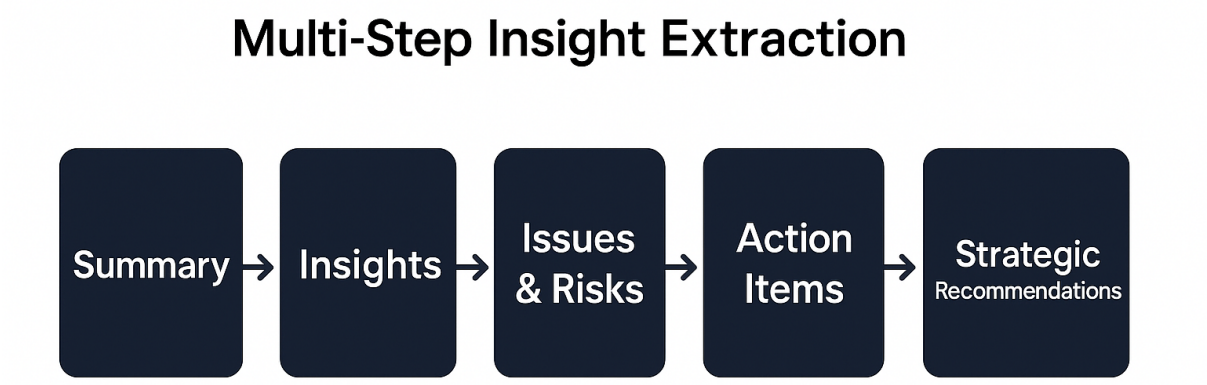
```
Format the final result into a neat table.
```

Result

A complete end-to-end workflow powered entirely by GenAI.

Section C – Multi-Step Insight Extraction

Multi-step insight extraction means using GenAI to transform **raw text → structured intelligence** by guiding it through sequential prompts.



---

1. Summary → Insights → Issues → Actions

This is the core pipeline for turning text into decisions.

Stage 1 — Summary

Condenses the text into the *essence*:

- What happened?
- What is important?
- What are the key points?

Example Prompt:

```
Summarize the text below into 5 bullet points.
[TEXT]
```

Stage 2 — Insights

Insights = hidden meaning behind the summary.

AI identifies:

- Patterns
- Root causes
- Implications
- Trends

Example Prompt:

```
Extract the top insights from the summary. Focus on hidden meaning,
implications, and patterns.
```

Stage 3 — Issues

AI flags:

- Risks
- Gaps
- Missing information
- Contradictions
- Operational bottlenecks

Example Prompt:

```
Identify all issues and risks mentioned or implied.
```

Stage 4 — Actions

AI converts problems into:

- Action items
- Owners
- Deadlines
- Priorities

Example Prompt:

```
Convert the issues into action items with Task, Owner, Deadline, Priority.
```

Why this matters

It transforms any:

- Meeting notes
- Emails
- Reports
- Customer feedback
- Chat transcripts

into a **decision-ready output**.

---

2. Prompt Chaining (No Coding)

Prompt chaining means connecting prompts **step-by-step**, where each output becomes the next input.

Example Flow

1. **Prompt 1:** Summarize
2. **Prompt 2:** Generate insights from the summary
3. **Prompt 3:** Identify issues from insights
4. **Prompt 4:** Convert issues into action items
5. **Prompt 5:** Format everything nicely

No code.
Just **human reasoning + sequence of prompts**.

Why it works

LLMs follow context *within the same conversation* — so the chain creates a compounding effect.

Typical Use Cases

- Project management
- Business analysis
- Requirement extraction
- Creating MOMs
- Risk analysis
- Strategy documentation

---

3. Structured Output (Tables, Bullets, Checklists)

GenAI can convert any unstructured text into **clean, readable formats**.

Supported formats

- **Tables**
- **Bullet lists**
- **Numbered lists**
- **Checklists**
- **Multi-section reports**
- **JSON**
- **Decision matrices**

Example Prompt

```
Structure the final output into:
1. Summary
2. Insights
3. Issues
4. Actions (as a table)
5. Recommendations
```

Why Structured Output is Powerful

- Saves hours of formatting
- Makes communication clear
- Allows stakeholders to digest quickly
- Perfect for automation and workflows

---

4. Multi-Step Reasoning Through Prompts

This is where the AI behaves like an analyst, not just a text generator.

How it works

You instruct the model to:

- Think step by step
- Avoid jumping to conclusions
- Break tasks into logical blocks
- Show reasoning before conclusion

Reasoning Prompt Examples

A. Chain-of-Thought Style Instruction

```
Think step-by-step. First summarise, then extract insights, then list
issues, then propose actions.
Do not skip steps.
```

B. Analytical Reasoning Instruction

```
Before answering, identify the underlying causes, assumptions, and
implications in the text.
```

C. Decision-Making Instruction

```
Based on the insights, predict 3 possible outcomes and their probabilities.
```

This shows the **real intelligence** of GenAI — not just content creation.

---

5. Designing Insight Frameworks

An insight framework is a **reusable template** to extract intelligence from any text.

Common Insight Frameworks

A. 4-Block Framework

1. Summary
2. Insights
3. Issues
4. Actions

B. 5-Lens Framework

1. What happened?
2. Why did it happen?
3. What is the impact?
4. What are the risks?
5. What should be done next?

C. Strategic Framework

1. Themes
2. Patterns
3. Gaps
4. Opportunities
5. Recommendations

D. Stakeholder Framework

1. What matters to Management?
2. What matters to Operations?
3. What matters to Customers?
4. What matters to Developers?

Example Insight Framework Prompt

```
Using the 4-block insight framework, analyze the below text:

Framework:
1. Summary
2. Insights (hidden meaning)
3. Issues (explicit + implied)
4. Actions with owners

[TEXT]
```

Why Insight Frameworks Matter

- Make AI output predictable
- Improve quality of reasoning
- Standardize analysis across a team
- Faster decision-making

Multi step Insight Extraction

You are an expert business analyst.
Analyze the text I provide using multi-step reasoning.
Follow these steps exactly and do not skip any stage:

1. Summarize the text into 5 key bullet points.
2. Extract the top insights (hidden patterns, root causes, implications).
3. Identify issues and risks that are mentioned or implied.
4. Convert issues into clear action items with:
   - Task
   - Suggested Owner
   - Priority (High/Medium/Low)
   - Deadline
5. Recommend 3 strategic improvements based on everything above.
6. Present the final answer in this structure:

Summary
[5 bullets]

Insights
[hidden meaning]

Issues & Risks
[list]

Action Items
| Task | Owner | Priority | Deadline |

Strategic Recommendations
[3 points]

Here is the text for analysis:
[PASTE TEXT HERE]