IMPACTEERS
FIND YOUR CALLING

Course Curriculum - DSA & System Design with Prompt Engineering

Hero Section | Data Structures and Algorithms | Interview Preparation | Low Level Design | High Level Design | Evaluations | Discussions

Data Structure Algorithms (DSA) & System Design

Week 1	Building blocks and basic algorithms	Time complexity	Standard notations Substitution method, recursion tree Master theorem	Sorting - 1	Bubble, Selection, Insertion Problem Solving	Sorting - 2	Divide and Conquer approach Merge, Quick sort Complexity analysis for all of them
		MAANG interviews	How to Leetcode MAANG interview processes Bar raiser/special rounds	Coding test	Eligibility test	Leveling test	Test from the leveling framework personalized to your level from The Great Pyramid
Week 2	Basic data structures	Searching	Binary, Ternary search Complexity analysis Modified Binary Searcha	Arrays	Theory and important functions Different types of questions Problem solving	Leveling test	Theory, Question patterns in Strings Problem solving
		Arrays and strings	Problem solving	Coding test	Weekend test on topics covered so far	Problem solving	Test from the leveling framework personalized to your level
Week 3	Basic data structures	Sets	Theory and basic functions Problem solving	Hashmaps	Theory and basic functions When to use hashmaps Problem solving	Problem solving	Miscellaneous sorting algorithms like Radix sort, Bucket sort, Counting sort overview
		Queues	Theory and basic functions When to use queues Problem solving	Coding test	Weekend test on topics covered so far	Leveling test	Test from the leveling framework personalized to your level
Week 4	Basic Data Structures and Dynamic Programming	Stacks - 1	Theory and basic functions When to use stacks Problem solving	Stacks - 2	Problem solving	Dynamic programming - 1	Identification of sub-problems Memoization and Dynamic programming
	Profile presentation	Dynamic programming - 2	Implementation of memoization Time complexity Problem solving	Coding test	Weekend test on topics covered so far	Profile presentation and interview prep	Emails, resume review, cover letter LinkedIn, GitHub, personal website Tracker sheet setup, Interview anxiety and confidence
Week 5	Dynamic programming	Dynamic programming - 3	Implementation of DP Transition from memoization to DP Time complexity and problem solving	Dynamic programming - 4	Problem solving	Dynamic programming - 5	Problem solving
		Dynamic programming - 6	Buffer session on DP	Coding test	Weekend test on topics covered so far	Leveling test	Test from the leveling framework personalized to your level
Week 6	Advanced data structures	Linked lists - 1	Theory and basic functions Problem solving	Dynamic programming - 4	Problem solving	Dynamic programming - 5	Problem solving
	Hiring manager round	Hiring manager round	Leadership principles Scenario questions. Bar raiser round Getting better at soft skills Interview closure questions	Coding test	Weekend test on topics covered so far	Leveling test	Test from the leveling framework personalized to your level
Week 7	Advanced data structures	Binary trees	Trees, Binary trees Problem solvin	Binary search trees - 1	Implementation of trees and binary search trees When to use BSTs	Binary search trees - 2	Problem solving
		Trie	Theory and basic functions When to use tries Problem solving	Coding test	Weekend test on topics covered so far	Leveling test	Test from the leveling framework personalized to your level
Week 8	Advanced data structures	Heaps - 1	Theory and basic functions Heap sort, min heap, max heap, heapify	Heaps - 2	Problem solving	Heaps - 3	Problem solving
		Job Hunt	Recruitment platforms Approaching recruiters Start working on personal project, open source contribution	Coding test	Weekend test on topics covered so far	Leveling test	Test from the leveling framework personalized to your level
Week 9	Graph algorithms	BFS and DFS - 1	DIrected, undirected, weighted graphs. How to define the input. BFS, DFS. Shortest path algorithms	BFS and DFS - 2	Problem solving	Minimum spanning trees	Minimum spanning tree Kruskal, Prims Implementation of MST
		Greedy	Greedy Implementation of greedy	Coding test	Weekend test on topics covered so far	Leveling test	Test from the leveling framework personalized to your level
Week 10	Advanced data structures	Graphs - 1	Problem solving	Graphs - 2	Problem solving	Backtrackin and recursion	Concept Problem solving
		Patterns	Types of questions Common patterns to identify the data structures from MindMap	Coding test	Weekend test on topics covered so far	Leveling test	Concept Problem solving
Week 11	Low Level Design	Object oriented programming	Git and GitHub Code structure. OOPS concepts Theory and Implementation	Introduction to design patterns	Why need design patterns 3 most useful design patterns with implementation Real life use cases	Low level design	Identify entities, design ER diagram Choose the right data structure Library management system Implementation details
		Advanced low level design	Building advanced systems Identifying and implementing the right design patterns	Hands-on	Game development (required for companies like Flipkart, Swiggy, Udaan, Razorpay)	Leveling test	Test from the leveling framework personalized to your level
Week 12	High level design concepts	Networking concepts	TCP IP, UDP protocols Client server network, P2P network DNS resolution	Database concepts	Data partitioning Indexing and database locks Master slave architecture CAP Theorem. SQL vs NoSQL	Memory management concepts	Paging and segmentation Cache eviction strategies Distributed file systems
		OS concepts on concurrency	Multithreading and concurrency Scheduling algorithms Semaphores and deadlocks Atomic transactions	Demo MAANG interview	A demo of how real interviews are conducted in MAANG companies. Level of questions and expectations	Leveling test	Test from the leveling framework personalized to your level
Week 13	High level system design interviews	System design concepts	Communication and data exchange REST APIs and AJAX Long polling, websockets and SSEs Load balancers, Caching	Approach	Requirements Estimations and components Identifying challenges with each component	Deep dive	Single point of failures Systems at scale
		Question	Designing a system like BookMyShow	System Design Test	Test on System design concepts	Leveling test	Test from the leveling framework personalized to your level
Week 14	MAANG system design interviews	Question	Designing Google Maps	Question	Designing Uber at scale	Open session	Any topic that needs to be revisited in system design
		Fireside chat	Fireside chat with a MAANG engineering manager	System Design Test	Test on System design concepts	Leveling test	Test from the leveling framework personalized to your level
Week 15	Advanced concepts for eligible learners	Union Find Topological sort	Union Find, Topological sort Problem solving, solving BFS / DFS question with union find.	Math for competitive programming	Combinatorics Primality test	Self balancing trees	Red Black Tree AVL Tree Problem solving
		Bit manipulation	Concept and applications Problem solving	Coding test series	Placement test - 4	Coding test series	Placement test - 5

Prompt Engineering Curriculum

Objective : Building Production-Ready AI Solutions

This comprehensive course equips software engineers with hands-on mastery in prompt engineering, AI integration, and automation workflows, enabling them to confidently build, deploy, and optimize AI-driven products.

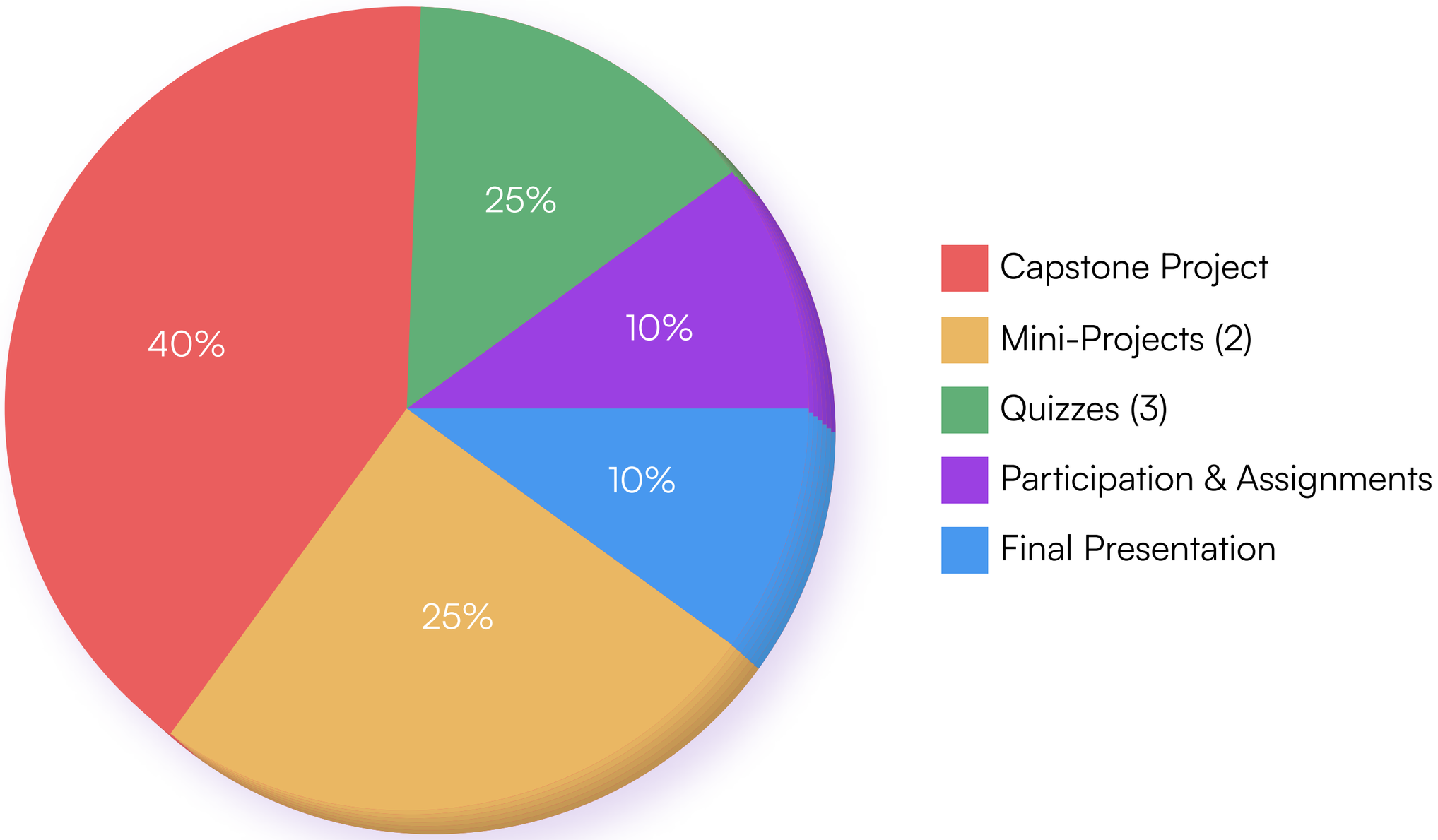
Module Structure Summary (7 Weeks)

Week	Concept	Session	Modules	Outcomes
1	Foundations of Prompt Engineering	1 - 2	LLM fundamentals, prompt architecture, and basic frameworks	**Objective:** Understand how LLMs work, the fundamentals of prompt construction, and how prompts affect model outputs. **Session 1:** Introduction to Prompt Engineering & AI Evolution • History of LLMs (GPT, Claude, Gemini) and open-source models • Transformer architecture (tokenization, embeddings) • Anatomy of a prompt: structure, temperature, tokens, stop sequences • Demo: ChatGPT API vs OpenAI Playground **Activity:** Analyze 3 different prompts and outcomes Assessment: Quiz 1 (Fundamentals) **Session 2:** Prompt Frameworks & Thought Chains • Zero-shot, few-shot, and Chain-of-Thought prompting • Role-based and task-based prompting patterns • Introducing instruction-tuning and contextual layering **Workshop:** Crafting structured prompts for code completion Outcome: Build reproducible, structured prompts for technical tasks
2	Advanced Prompt Design	3 - 5	Optimization, structured prompting, and reducing hallucinations	**Objective:** Build advanced-level prompts that optimize accuracy and reduce hallucinations. **Session 3:** Optimizing Prompt Outcomes • Multi-prompt workflows, handling ambiguity and context-depth • Role assignment and chaining reasoning, prompt compression **Session 4:** Advanced Patterns for Developers • Code generation, debugging, and refactoring via prompting • API documentation generation, prompt templates for developer workflows **Session 5:** Hands-on Lab: Prompt Engineering Toolkit Using tools (PromptLayer / LangSmith) to track and manage prompts Creating prompt libraries, integrating memory & context **Activity:** Mini-Project 1 - “AI Pair Programmer” (Build a debugging/refactoring tool) **Assessment:** Quiz 2 (Advanced Patterns)
3	Programming with Prompts & APIs	6 - 8	Integrating LLMs into codebases (OpenAI, Gemini APIs)	**Objective:** Enable developers to integrate AI into real software workflows. **Session 6:** OpenAI & Gemini API Integration Authentication, API setup, rate limits, structured inputs/outputs Cost optimization and logging strategies **Session 7:** Building Apps with Prompts Using Python/JS frameworks to call LLM APIs Converting a prompt into a working microservice (logging & error handling) **Session 8:** Building with Multiple APIs Combining models (e.g., Whisper + GPT + DALL·E) Automating developer tasks (commit summarization, doc writing) **Activity:** Mini-Project 2: “AI Documentation Assistant” Assessment: Test 1 (Integration & Code)
4	LangChain & Agentic Workflows	9 - 10	Integrating LLMs into codebases (OpenAI, Gemini APIs)	**Objective:** Build autonomous workflows and AI agents. **Session 9:** LangChain Fundamentals Chains, agents, memory, and tools (RAG implementation overview) Setting up LangChain locally, chaining multiple LLM calls **Session 10:** Building Multi-Step Agents Function calling and tool augmentation Agent frameworks for task orchestration and testing **Outcome:** Create a local AI assistant that can fetch, analyze, and summarize data automatically
5	Applied Projects	11 - 13	Developing real-world tools using industry-standard APIs (3 projects)	**Objective: Build three real-world AI tools relevant to enterprise developers.Session 11: AI Code Reviewer (Project 1)** **Prompting for style, logic, and performance feedback** **Version control integration (GitHub API)** **Session 12: Resume Builder / Job Assistant (Project 2)** **Role-based prompt design for candidate screening and application enrichment** **Session 13: Chatbot Development (Project 3)** **Integrating GPT into a web chat system** **Context windows, memory persistence, and fine-tuning options** **Outcome: Build 3 working AI-integrated tools and deploy to GitHub**
6	Responsible AI & Evaluation	14 - 15	Ethics, bias management, and rigorous performance testing	**Objective:** Learn to build safe, ethical, and production-ready AI systems. **Session 14:** AI Ethics, Bias & Compliance Bias, safety, and hallucination management OpenAI usage policies, data privacy, and enterprise compliance **Session 15:** Evaluation & Prompt Debugging A/B testing prompts and evaluation metrics for LLM quality Human feedback integration (RLHF overview) **Assessment:** Final Quiz: Responsible AI & Evaluation
7	Capstone Project & Demo	16	Building a final comprehensive, full-stack AI solutions	**Objective:** Integrate all learning into one comprehensive project. **Session 16:** Capstone Project Presentation Build a full AI-integrated app (e.g., Knowledge Assistant, Contextual Code Analyzer, LLM Workflow Automator) Final presentation to mentor panel **Deliverables:** Working prototype, Project report (architecture, prompt design), GitHub submission + video demo

IMPACTEERS
FIND YOUR CALLING

Evaluation Scheme & Deliverables

Weightage Breakdown



Key Deliverables

Successful completion requires:

01
Comprehensive Capstone Project

02
Mini-Projects (AI Pair Programmer, AI Doc Assistant)

03
Quizzes + 1 Final Test