

AI-Assisted Coding Training

Duration: 2 Days

Description

The purpose of this training is to clearly understand LLM fundamentals (token, prompt, embedding/vector), develop a hands-on example using a basic RAG approach, and teach tool/service integration via the Model Context Protocol (MCP). The training aims to provide a production-oriented mindset with enterprise-scale best practices such as security, versioning, testing, and observability.

Learning Outcomes

- Understand token, prompt, and embedding/vector concepts
- Integrate local or enterprise services with LLMs via MCP and build a design-to-code prototype using FigmaMCP
- Apply security, versioning, testing, and observability best practices in production-focused workflows

Audience

- Backend, Frontend, and Mobile Developers
- Product and Prototyping Teams
- Teams adopting LLM-assisted development within IDEs

Outline

LLM Fundamentals & Hands-on Coding

- Core concepts: LLMs, tokens, embeddings/vectors
- Context window and token budget management
- Prompt design (roles, formats, examples) and evaluation
- Basic RAG: indexing, similarity search, retrieval flow
- Code generation and refactoring
- Test generation (unit/fixtures) and validation
- Security and content filtering (PII masking)
- Cost and performance optimization

Prompts, IDE Integrations & Best Practices

- Advanced prompt patterns and anti-patterns
- IDE integrations (VS Code, Cursor, Windsurf)
- Rule sets and style guides
- Practical workflows for code and test generation
- Performance and cost optimization
- Collaboration and review workflows
- Project: Git-integrated design-to-code application

Prerequisites

Basic programming knowledge is required.