Faysal Aziz

AI Accelerated Development

From everyday AI assistance to MCP-connected, agentic applications in one intensive day

FA-0050 · one-day

Agentic & Generative AI

Introduction

Modern software teams need more than prompt-writing skills. They must know which AI tool fits a task, how to give it reliable context, when to permit autonomous action, and how to prevent generated code from reaching production unchecked. This one-day course reflects the current ecosystem: Gemini supports multimodal and native image-generation workflows; Claude Code provides repository-based development capabilities, permissions, subagents and MCP connectivity; and ChatGPT supports MCP-backed tools and interactive application interfaces.

The content also incorporates the MCP specification released on 28 July 2026, including its stateless architecture, alongside pull-request code scanning and merge-protection practices. The course is led by an instructor with over 30 years of industry experience and concentrates on real, currently demanded development practices rather than academic theory. The scope is deliberately limited to foundations and an integrated working approach that can reasonably be covered in approximately six to seven instructional hours.

Target audience: Senior and junior software developers, front-end developers and QA automation engineers.

Learning outcomes

Participants will be able to:

- Select ChatGPT, Claude Code and Gemini capabilities according to development task, risk and integration requirements.
- Establish controlled AI-assisted workflows for coding, debugging, testing and documentation.

- Identify the architectural changes required to move from a static interface to a tool-using AI application.
- Explain and configure the core components of an MCP-connected application.
- Structure generative and adaptive user interfaces around validated model outputs.
- Apply automated code-quality and security checks before deployment.

Mandatory Prerequisites

This is beginner-level training in **applied AI development**, not beginner-level software development. All prerequisites are mandatory.

- Ability to independently write, debug and test applications using JavaScript/TypeScript or Python.
- Ability to build a basic web interface and consume a REST API.
- Working knowledge of JSON, asynchronous programming, environment variables and API authentication.
- Practical experience with Git, branches, commits, pull requests and merge-conflict resolution.
- Working knowledge of unit testing, integration testing and basic CI/CD concepts.
- A fully configured laptop with administrative access, Git, VS Code, a current supported programming runtime and package-management tools.
- Authorised access to ChatGPT/OpenAI, Claude/Anthropic, Gemini/Google and a GitHub repository.
- A non-confidential sample application or repository suitable for AI-assisted modification.
- Completion of the trainer's pre-course environment and technical-readiness check.

Participants who cannot meet these requirements should complete prerequisite programming, Git and API training before attending.

Training outline

1. Current AI Tools for Software Delivery

Tool Capabilities and Appropriate Usage

1. ChatGPT plugins, Apps SDK and MCP-backed tools
2. Claude Code development workflows
3. Gemini multimodal and image-generation capabilities
4. Task-to-tool selection
5. Model limitations and output verification

Responsible Operating Boundaries

1. Sensitive code and confidential data
2. Secret and credential protection
3. Human approval requirements
4. Permission scoping
5. Untrusted prompts and external context

2. AI-Assisted Development and Automation

Effective Codebase Context

1. Repository instructions
2. Requirements and acceptance criteria
3. Relevant-file selection
4. Structured development requests
5. Change planning before implementation

Development and QA Workflows

1. Codebase navigation
2. Controlled code generation
3. Refactoring assistance
4. Debugging and failure analysis
5. Unit and integration test generation
6. Pull-request review support

Repeatable AI Automation

1. Structured outputs
2. Tool and function invocation
3. Claude Code skills, hooks and subagents
4. Approval checkpoints
5. Logging and change traceability

3. From Static Interfaces to Agentic Applications

Agentic Application Architecture

1. Model, application and tool boundaries
2. Function schemas and argument validation
3. Multi-step task orchestration
4. Application state and conversation state
5. Retries, fallbacks and error handling

Generative and Adaptive User Interfaces

1. Model-generated content
2. Model-selected interface components
3. Structured UI payloads
4. Streaming and progressive rendering
5. User confirmation for consequential actions
6. Accessibility and predictable interaction

AI-Generated Visual Assets

1. Gemini native image generation
2. Conversational image editing
3. Visual constraints and brand consistency
4. Asset review and approval
5. Generated-content provenance

4. Model Context Protocol Integration

Current MCP Foundations

1. Host, client and server architecture
2. Stateless request architecture
3. Protocol and capability negotiation
4. Tools, resources and prompts
5. Input and output schemas

Connecting Applications and Models

1. Local and remote MCP servers
2. Tool discovery and invocation
3. Claude Code MCP connectivity
4. ChatGPT MCP-backed integration
5. Authentication and authorisation
6. Version compatibility

MCP Security and Reliability

1. Least-privilege tool design
2. Read-only and destructive operations
3. User approval controls
4. Prompt-injection boundaries
5. Validation, timeouts and error contracts
6. Audit and ownership requirements

5. Pre-Deployment Scanning and Release Controls

Automated Quality Gates

1. Formatting, linting and type checking
2. Unit and integration test gates
3. Static application security testing
4. Dependency and licence scanning
5. Secret scanning

Pull-Request Protection

1. Code scanning on pull requests
2. Severity thresholds
3. Merge-protection rules
4. AI-assisted findings triage
5. False-positive management
6. Human review of AI-generated changes

Production Readiness

1. Security and privacy review
2. Output evaluation criteria
3. Failure and fallback paths
4. Monitoring and operational ownership
5. Controlled release approval

Disclaimer

This course outline is provided as an indicative training framework and does not constitute a fixed or binding syllabus. The trainer reserves the right to amend, reorder, replace or omit content where reasonably necessary to accommodate participant readiness, technical changes, operational constraints or learning priorities, without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur