Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

LLM Application Development with TypeScript and Python

Class Duration

14 hours of live training delivered over 2-3 days to accommodate your scheduling needs.

Student Prerequisites

- Professional software development experience in TypeScript or Python (or both)
- Familiarity with REST APIs and async programming

Target Audience

Software engineers building production LLM-powered features, services, or applications. Equally relevant for full-stack developers adding AI capabilities to existing products and for backend engineers building LLM API wrappers, orchestration layers, or agentic pipelines.

Description

This course teaches end-to-end LLM application development using TypeScript and Python - the two languages most widely used with frontier model APIs. We cover the full development lifecycle: model API integration, prompt pipeline construction, streaming responses, structured output parsing and validation, tool/function calling, conversation state management, error handling and retries, cost tracking, and production deployment patterns. Both TypeScript (using Vercel AI SDK and Anthropic/OpenAI TypeScript clients) and Python (using Pydantic AI, LangChain, and direct SDK) tracks are included, and labs build realistic application components throughout.

Learning Outcomes

- Integrate frontier model APIs (Anthropic, OpenAI, Gemini) in TypeScript and Python applications.
- Build streaming LLM response handlers with proper error and cancellation handling.

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Generate and validate structured outputs using JSON Schema and Zod/Pydantic.
- Implement tool/function calling pipelines with multiple tool invocations.
- Manage conversation history and context window budgets in multi-turn applications.
- Apply retry, timeout, and fallback patterns for production-grade reliability.
- Track token usage and costs per request and per user session.
- Deploy LLM application services with environment-appropriate secrets management.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Node.js 22+ and/or Python 3.12+, API keys for at least one frontier model, and Git.

Training Topics

Model API Integration

- Anthropic, OpenAI, and Gemini SDK setup
- Messages API request/response structure
- Async and streaming response handling
- Error types and retry strategies

Prompt Pipeline Design

- Pipeline architecture: input → prompt → model → output
- Separating prompt templates from application logic
- Parameterized prompts and versioning
- Chain-of-thought and multi-step pipelines

Structured Outputs

- JSON mode and native structured output support
- Validation with Zod (TypeScript) and Pydantic (Python)
- Error recovery for malformed outputs
- Type-safe response handling patterns

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Tool/Function Calling in Applications

- Tool definition and invocation round-trip
- Sequential and parallel tool call patterns
- Tool result aggregation and re-prompting
- Error handling in tool pipelines

Conversation State Management

- Storing and trimming message history
- Context window budget management
- Summarization for long conversations
- Session persistence with a data store

Reliability Patterns

- Retry with exponential backoff
- Timeout and cancellation handling
- Rate limit management
- Primary/fallback model routing

Cost and Token Tracking

- Token counting before and after requests
- Per-request and per-session cost attribution
- Budget limits and soft/hard cost caps
- Cost reporting integration

Deployment

- Environment variable and secrets management
- Containerizing LLM application services
- Health checks and graceful degradation
- Observability: logs, traces, and metrics

