CTR-912 Cursor AI

Overview

Course Duration: 5 Days

About This Course

This course offers a hands-on journey into AI-assisted full-stack development using Cursor AI. Participants learn how to write effective prompts, generate and refine production-quality code, and deploy full-stack applications through AI-supported workflows. Covering database modeling, authentication, UI/UX design, backend logic, code review, and ethical AI use, the course equips graduates with the skills and confidence to build, test, and launch real-world applications enhanced by artificial intelligence.

Audience Profile

This course is designed for developers, data engineers, and tech enthusiasts who want to accelerate their coding workflow using AI tools like Cursor. It's ideal for both beginners who want to understand how AI can assist in full-stack development, and experienced developers who wish to integrate AI-powered code generation, automation, and project management into their existing workflows.

At Course Completion

- Use Cursor AI effectively for code generation, refactoring, and documentation.
- Apply prompt engineering techniques to guide AI toward producing accurate, maintainable code.
- Build and deploy full-stack web applications (frontend + backend + database) using AI assistance.
- Design and manage databases, APIs, and authentication systems through AI-driven development.
- Review, test, and secure AI-generated code to ensure quality and reliability.
- Implement custom AI workflows, rules, and agents to automate repetitive development tasks.

Course Outline

Module 1: Introduction to AI-Assisted Development & Cursor AI

- What "vibe coding" means and why it's emerging (with context from tools like Cursor).
- Overview of Cursor AI: what it is, how it works, modes (chat vs composer vs agent)
- Advantages, limitations, and best practices for AI-powered code generation
- Setup: installing Cursor AI, linking to GitHub / code repository, creating first workspace
- Project vision: selecting the target full-stack application you'll build through the course

Module 2: Prompt Engineering for Code Generation

- Fundamentals of prompt engineering: how to write effective prompts for code generation (clear intent, scope, examples)
- Structured prompting: chaining prompts, refining, iterating results
- Using Cursor's "Rules" or "Agent Mode" to steer generation
- Contextualising for your project: specifying stack, architecture, style guidelines
- Hands-on: generate components/modules via prompts and review/refine output
- **Module 3: Setting Up the Tech Stack**
  Choosing a technology stack (for example: Next.js with TypeScript, Tailwind CSS, Drizzle ORM, PostgreSQL, etc.)
- Configuring project scaffolding via Cursor AI: prompts to generate folder structure, config files, README
- Version control practices and branching strategy when using AI-generated code
- CI/CD basics (brief) as preparation for later deployment

Module 4: Database Modeling & Data Layer with AI

- Defining relational data models: entities, relationships, constraints
- Using Cursor (and prompts) to create schema migrations, ORM models via Drizzle or similar
- Seeding and managing data with AI assistance
- Ensuring data integrity, handling versioning and migrations with AI-generated code

Module 5: Authentication, Authorization & Billing

- User authentication flows: sign-up, login, password reset using AI code generation
- Role-based access control (RBAC): prompts to generate protected routes, middleware
- Subscription/billing model: integrating e.g. Stripe (or similar) via AI-generated modules
- Security considerations: verifying generated code, securing secrets and environment variables

Module 6: UI/UX & Frontend Components via AI

- Designing accessible, responsive UIs: using Tailwind CSS, UI libraries (e.g., shadcn)
- Prompting Cursor to generate reusable UI components, layout system, theme switching
- Connecting frontend to backend via API endpoints (AI-generated stubs + manual refinement)
- Optimizing performance, accessibility, user experience

Module 7: UI/UX & Frontend Components via AI

- Defining API endpoints: contract first vs prompt first
- Using Cursor to generate server logic, controllers, services
- Integrating ORM queries, business rules, validations
- Testing generated code: unit tests, integration tests, and how to prompt AI to generate tests
- Error handling, logging, monitoring (basic)

Module 8: Full-Stack Integration, Deployment & Scaling

- Connecting frontend and backend, end-to-end flows
- Environment management: dev vs staging vs production using prompts
- Deployment pipelines: e.g., Vercel, AWS, or other hosting, using AI to generate deployment configs
- Scaling considerations: database scaling, caching, static assets, AI-generated optimization
- Monitoring & alerting setup (basic introduction)

Module 9: Quality, Maintainability & AI-Generated Code Review

- How to review and audit AI-generated code: quality, readability, security
- Refactoring generated code: prompts for optimization, code quality improvements
- Documentation generation via prompts, API docs, README updates
- Handling technical debt in AI-assisted development workflows

Module 10: Advanced AI Workflows and Customisation

- Custom rules for Cursor: creating project-specific “Cursor Rules” to standardize output
- Agent workflows: chaining AI agents, multi-agent prompts, automating workflows
- Advanced prompt patterns: templates, context memory, dynamic prompt generation
- Integrating additional AI tools: LLMs, vector search, generative assets (optional)

Module 11: Security, Ethics & Responsible AI in Development

- Security risks of generated code: injection, improper auth, dependencies
- Ethical considerations: ownership, reliability, transparency of AI-generated code
- Governance in AI-assisted development: versioning, audits, responsibility
- Using prompts to generate secure code—and to validate security aspects

Module 12: Project Capstone: Build and Launch Your App

- Final project: build a production-ready full-stack app using Cursor AI as the primary driver
- Define project requirements, milestones, deliverables
- Use learned workflows: prompt engineering → code generation → review → deployment • Demo, publish, and gather feedback
- Retrospective: what worked, what didn't, how you'll apply this in real-world workflows

Module 13: Future Trends & Next Steps

- Review of emerging trends: "vibe coding", AI-native development workflows
- Continuous learning: how to keep your AI workflows sharp
- Team workflows: how to scale AI-assisted dev in a team, codebase, organization
- Resources, communities, and how to stay ahead in the AI-development space