

Python Full Stack AI Engineer: 22 Courses in 1

Build AI apps, agents, chatbots, and full stack web projects using Python, APIs, databases, automation, and modern AI tools. For developers ready to build real products.

Cybersecurity Instructor: Derek Holloway • 5 sections • 57 lessons • 27.00h total

[View this course online](#) [Browse all courses](#)

About this course

Most developers learn Python basics, then hit a wall when they need to ship a complete AI-powered application. They know the syntax but struggle to connect databases, build APIs, integrate LLMs, and deploy anything that actually works in production. The gap between "I can write Python" and "I can build full stack AI products" is where careers stall.

This course closes that gap. Over 50+ hours of video, you'll work through 22 focused modules covering Python foundations, full stack web development with Flask/FastAPI and React, data pipelines and automation, machine learning and LLM integration with RAG and agents, and production engineering for AI systems. Every section includes hands-on projects, from a CLI task manager and full stack web app to an autonomous AI agent with tool access. You'll build real applications that solve real problems, not toy examples.

Course curriculum

Python Foundations and Programming Essentials

- **Setting Up Your Python Development Environment (12m)**
Install Python 3.12+, configure VS Code with extensions, set up virtual environments, and create your first project structure. We'll also cover pip, pyproject.toml, and dependency management basics.
- **Core Python Syntax: Variables, Data Types, and Operators (22m)**
Master strings, numbers, booleans, lists, dictionaries, sets, and tuples. Understand type hints, f-strings, and when to use each data type for clean, readable code.
- **Control Flow: Conditionals, Loops, and Pattern Matching (18m)**
Write decision-making logic with if/elif/else, iterate with for and while loops, and use Python 3.10+ match-case statements. Includes practical exercises with nested conditions.
- **Functions, Closures, and Decorators (25m)**
Define reusable functions with arguments, default values, and *args/**kwargs. Learn closures, decorators, and lambda functions. You'll write your own logging decorator from scratch.
- **Object-Oriented Python: Classes, Inheritance, and Dunder Methods (30m)**
Build classes with attributes and methods, use inheritance and composition, implement special methods like __init__, __str__, __repr__, and __eq__. Covers abstract base classes and dataclasses.
- **Modules, File Handling, and Error Management (20m)**
Organize code into modules and packages. Read/write files in text, JSON, CSV, and binary formats. Handle exceptions with try/except/finally, custom exceptions, and context managers.
- **Scripting Essentials: Regex, Command-Line Tools, and Automation (25m)**
Use regular expressions for text parsing, build CLI tools with argparse and click, and write automation scripts for file management and system tasks.
- **Data Structures and Algorithms with Python (32m)**
Implement stacks, queues, linked lists, trees, and hash maps. Solve common algorithm patterns: searching, sorting, recursion, and dynamic programming. Focuses on problem-solving skills for technical interviews and real-world optimization.
- **Project: Build a CLI Task Manager and File Automation Script (45m)**
Apply everything from this section by building two projects: a command-line task manager with persistent storage and a file automation script that organizes, renames, and processes files in bulk.

Full Stack Web Development with Python

- **HTML and CSS Essentials for Backend Developers (20m)**
Learn semantic HTML, CSS selectors, Flexbox, Grid layout, and responsive design. Focus on what you need to build functional UIs without becoming a designer.
- **JavaScript Fundamentals: DOM Manipulation and Async Code (25m)**
Understand variables, functions, DOM manipulation, event handling, fetch API, and async/await. Enough JS to build interactive frontends and understand React.
- **React Fundamentals: Components, State, and Props (30m)**
Build single-page applications with React. Create functional components, manage state with hooks, pass data through props, and handle user events. Set up a project with Vite.

- **React Intermediate: Routing, Context, and Custom Hooks (25m)**
Add client-side routing with React Router, manage global state with Context API, and create reusable custom hooks. Build a multi-page application structure.
- **Backend with Flask: Routing, Templates, and Blueprints (28m)**
Set up a Flask application with routing, Jinja2 templates, request handling, and blueprints for modular code organization. Build a REST API from scratch.
- **FastAPI for High-Performance APIs (32m)**
Create async APIs with FastAPI. Use Pydantic models for validation, automatic OpenAPI documentation, dependency injection, and background tasks. Compare approaches with Flask.
- **PostgreSQL: Relational Database Design and Queries (28m)**
Design relational schemas, write SQL queries, use joins and indexes, and connect PostgreSQL to Python with SQLAlchemy ORM. Covers migrations with Alembic.
- **MongoDB and NoSQL for Flexible Data Storage (22m)**
Work with document-based storage using MongoDB and PyMongo. Learn when to choose NoSQL over relational databases, schema design patterns, and aggregation pipelines.
- **RESTful API Design, Documentation, and GraphQL Basics (25m)**
Design clean REST APIs following best practices. Document endpoints with Swagger/OpenAPI. Introduction to GraphQL with Strawberry for Python. Includes versioning and pagination strategies.
- **Authentication and Authorization: JWT, OAuth2, and Security (30m)**
Implement user registration, login, and password hashing. Secure APIs with JWT tokens, OAuth2 flows (Google, GitHub), role-based access control, and CSRF/XSS protection.
- **Docker, CI/CD Pipelines, and Cloud Deployment (35m)**
Containerize applications with Docker and Docker Compose. Set up GitHub Actions for automated testing and deployment. Deploy to Render and AWS EC2 with proper environment configuration.
- **Project: Build and Deploy a Full Stack Web Application (55m)**
Combine all skills from this section: a React frontend, Flask/FastAPI backend, PostgreSQL database, user authentication, and Docker deployment. You'll have a production-ready web application by the end.

Data, APIs, and Automation

- **Working with Third-Party APIs: OpenAI, Google, Stripe (25m)**
Make authenticated API calls, handle pagination and rate limits, manage API keys securely, and process responses. Real examples with OpenAI, Google Maps, and Stripe payment APIs.
- **Social Media and Utility APIs: Twitter, Telegram, and Email (22m)**
Integrate with Twitter/X API, Telegram Bot API, and SMTP email services. Build reusable API client classes and handle webhooks for real-time notifications.
- **Data Processing with Pandas and NumPy (30m)**
Load, clean, transform, and analyze datasets using Pandas DataFrames and NumPy arrays. Covers groupby operations, merging datasets, handling missing data, and performance optimization.
- **Building Data Cleaning and Processing Pipelines (22m)**
Create reusable data pipelines that extract, transform, and load data from multiple sources. Use Python generators, logging, and error handling for production-grade data workflows.

- **Web Scraping with BeautifulSoup and Scrapy (28m)**
Extract data from websites using BeautifulSoup for simple pages and Scrapy for large-scale crawling. Handle pagination, dynamic content, and respect robots.txt and rate limits.
- **Browser Automation with Selenium and Playwright (25m)**
Automate browser interactions for testing and scraping dynamic websites. Use Selenium WebDriver and Playwright for headless browsing, form submission, and screenshot capture.
- **Async Python and Background Task Processing with Celery (30m)**
Write asynchronous code with asyncio and aiohttp. Set up Celery with Redis as a message broker for background task processing. Schedule recurring jobs and monitor task queues.
- **Building Telegram and Discord Bots (28m)**
Create functional chatbots for Telegram and Discord. Handle commands, inline keyboards, message formatting, and integrate with external APIs for dynamic responses.
- **Project: Multi-Source Data Dashboard and Social Media Automation Bot (50m)**
Build a dashboard that aggregates data from APIs, web scraping, and databases. Also create a social media bot that posts scheduled content and responds to user interactions automatically.

AI and Machine Learning Integration

- **Machine Learning Fundamentals with scikit-learn (35m)**
Understand supervised and unsupervised learning, train classification and regression models, evaluate performance with metrics, and prevent overfitting with cross-validation.
- **TensorFlow and PyTorch Basics for Deep Learning (30m)**
Build neural networks for image and text tasks. Understand tensors, layers, loss functions, and training loops. Compare TensorFlow and PyTorch approaches for common use cases.
- **Natural Language Processing: Tokenization and Embeddings (28m)**
Process text data with tokenization, stemming, lemmatization, and word embeddings (Word2Vec, GloVe). Build a sentiment analysis model and text classification pipeline.
- **Working with the OpenAI API for LLM Applications (25m)**
Use the OpenAI API for chat completions, function calling, vision, and structured output. Manage token usage, implement retry logic, and handle streaming responses.
- **Open-Source LLMs: Hugging Face Transformers and Ollama (30m)**
Run models locally with Ollama and use Hugging Face Transformers for inference and fine-tuning. Compare open-source models (Llama, Mistral, Phi) and choose the right model for your task.
- **Prompt Engineering: Techniques, Patterns, and Evaluation (22m)**
Write effective prompts using few-shot, chain-of-thought, and role-based patterns. Evaluate prompt quality systematically and build prompt templates for production applications.
- **Fine-Tuning LLMs and Model Evaluation (28m)**
Fine-tune open-source models on custom datasets using LoRA and QLoRA. Evaluate model performance with automated benchmarks and human feedback loops.
- **Vector Databases: Pinecone, ChromaDB, and Embeddings Storage (25m)**
Store and query vector embeddings with Pinecone and ChromaDB. Understand similarity search, indexing strategies, and metadata filtering for retrieval applications.
- **Retrieval-Augmented Generation (RAG) Pipelines (35m)**
Build a complete RAG system: document ingestion, chunking strategies, embedding generation,

vector storage, retrieval, and answer generation. Handle multi-document queries and source attribution.

- **Building AI Agents with LangChain and Tool Use (32m)**
Create AI agents that use tools, maintain memory, and make multi-step decisions. Implement tool calling, agent executors, and conversation history management with LangChain.
- **Multi-Agent Orchestration and Complex Workflows (28m)**
Design systems where multiple AI agents collaborate on tasks. Implement supervisor patterns, agent-to-agent communication, and workflow graphs for complex problem solving.
- **Computer Vision Basics: Image Classification and Object Detection (30m)**
Process images with OpenCV, build image classifiers with transfer learning, and implement object detection using YOLO. Integrate vision capabilities into web applications.
- **Project: RAG Knowledge Assistant and Autonomous AI Agent (55m)**
Build two AI projects: a RAG-powered knowledge assistant that answers questions from your documents, and an autonomous agent that uses tools (web search, code execution, APIs) to complete tasks independently.

Advanced Full Stack AI Applications and Production Engineering

- **System Design for AI Applications: Architecture Patterns (30m)**
Design scalable AI systems using microservices, event-driven architecture, and service decomposition. Plan for horizontal scaling, fault tolerance, and data consistency.
- **Caching Strategies with Redis for AI Applications (20m)**
Implement caching for LLM responses, API results, and session data using Redis. Reduce latency and API costs with intelligent cache invalidation and TTL strategies.
- **Chatbot Engineering: Multi-Turn Conversations and Context (32m)**
Build production chatbots that handle multi-turn conversations, maintain context across sessions, manage conversation state, and implement graceful error handling and fallback responses.
- **AI Safety: Guardrails, Content Filtering, and Output Validation (25m)**
Implement guardrails for AI applications: input validation, output filtering, prompt injection prevention, content moderation, and rate limiting. Use tools like Guardrails AI and NeMo Guardrails.
- **Streaming Responses with WebSockets and Server-Sent Events (28m)**
Build real-time AI interfaces that stream LLM responses to users. Implement WebSockets for bidirectional communication and Server-Sent Events for one-way streaming with FastAPI.
- **Frontend Integration: Building AI-Powered UI Components (30m)**
Create React components for AI interfaces: chat windows, streaming text displays, file upload with progress, and real-time status indicators. Handle loading states and error recovery.
- **Observability: Logging, Monitoring, and Tracing for AI Systems (25m)**
Set up structured logging, distributed tracing with OpenTelemetry, and monitoring dashboards. Track LLM usage, latency, error rates, and user behavior in production.
- **Testing AI Applications: Unit, Integration, and Evaluation Tests (28m)**
Write tests for AI applications: mock LLM calls, test agent tool usage, evaluate RAG quality, and create regression test suites. Use pytest fixtures and parametrize for thorough coverage.

- **Cost Optimization: Managing LLM and Infrastructure Expenses (22m)**
Track and reduce costs from LLM API calls, compute resources, and database usage. Implement token counting, model cascading, caching, and auto-scaling strategies.
- **Project Management with Linear and Agile for AI Teams (18m)**
Manage AI projects with Agile practices. Use Linear for task tracking, write technical specifications, estimate work for AI features, and handle the uncertainty inherent in ML projects.
- **Building a Complete AI SaaS Product: From Idea to Deployment (45m)**
Combine everything you've learned to build a multi-feature AI SaaS application. Includes user management, billing integration, an AI-powered core feature, admin dashboard, and production deployment.
- **Portfolio Projects: Showcasing Your Full Stack AI Skills (30m)**
Structure your GitHub portfolio, write compelling README files, create demo videos, and deploy project showcases. Prepare for technical interviews with system design and coding questions specific to AI engineering.
- **The AI Engineering Landscape in 2026: Tools, Trends, and Career Paths (22m)**
Review the current state of AI engineering as of mid-2026. Understand emerging tools, career paths (ML engineer, AI application developer, platform engineer), and how to keep learning as the field evolves.
- **Course Wrap-Up: Your Next Steps as a Full Stack AI Engineer (15m)**
Review the skills you've built across all 22 course modules. Get a structured learning path for continued growth, recommended open-source projects to contribute to, and communities to join.

Discover our blog

Learn more with free articles written by our professors and guest experts.

Latest article: **[Flight Attendant Preparation: What to Study Before You Apply](#)**

[Visit the blog](#) [All courses](#) [This course](#)