

Python to AI Systems — Full Syllabus

Module 2: Version Control Basics (Git)

- Git init, add, commit
- .gitignore
- GitHub: creating a repo, push/pull
- Writing good commit messages

Module 3: Testing Basics

- Why test code
- pytest basics: writing and running tests
- Assertions
- Testing simple functions
- Habit: write at least one test file per future project

Module 4: Advanced Python

- Virtual Environments
- Decorators
- Iterators
- Generators
- Context Managers
- Regular Expressions
- Type Hinting
- Dataclasses

Module 5: SQL Fundamentals

- Database Basics
- Tables
- CRUD
- Filtering
- Sorting
- Aggregate Functions
- Joins
- Group By

Module 6: PostgreSQL

- Installation
- Database Design
- Relationships
- Constraints
- Indexes

Module 7: Python + Database

- sqlite3
- PostgreSQL
- SQLAlchemy ORM
- Alembic Migrations

Project: Employee Management Database

Module 8: Automation

- File Automation
- Excel Automation
- PDF Processing
- Email Automation
- Folder Automation

- Web Scraping
- Browser Automation
- Scheduling Tasks

Projects: File Organizer

Module 9: Git Collaboration

- Branching
- Pull Requests
- Collaboration Workflows

Module 10: HTTP & REST APIs

- Client-Server Architecture
- HTTP / HTTPS
- JSON
- REST APIs
- GET, POST, PUT, PATCH, DELETE
- Status Codes
- Headers
- Authentication
- JWT

Module 11: Flask

- Flask Basics
- Routing
- Templates (Jinja2)
- Forms
- Sessions
- Cookies
- CRUD Applications
- REST APIs with Flask

- Authentication
- File Uploads

Project: Blog Application

Module 12: FastAPI — Part 1 (Core)

- FastAPI Basics
- Uvicorn
- Routing
- Path Parameters
- Query Parameters
- Request Body
- Response Models
- Pydantic

Module 13: FastAPI — Part 2 (Production Concepts)

- Dependency Injection
- Middleware
- Background Tasks
- WebSockets
- Authentication
- Database Integration
- Async Programming

Project: Production REST API

Module 14: Software Engineering & Deployment

(Positioned right after the Flask/FastAPI projects so students deploy something real, instead of learning Docker in the abstract)

- Linux Basics
- Environment Variables
- Configuration Files

- Docker
- Docker Compose
- Testing (integration-level, building on Module 3's pytest foundation)
- Logging
- Deployment — deploy the Blog Application and/or Production REST API

Module 15: AI Foundations

- What is AI?
- Machine Learning
- Deep Learning
- Neural Networks
- Large Language Models
- Transformers
- Tokens
- Context Windows

Module 16: Prompt Engineering

- Prompt Engineering
- Zero-shot
- Few-shot
- Chain of Thought
- Role Prompting
- Structured Outputs

Module 17: LLM API Integration

- OpenAI API
- Gemini API
- Claude API
- API Integration
- Streaming Responses
- Function Calling
- API Cost & Rate-Limit Awareness (*added — matters once 3 providers are in play*)

Module 18: RAG (Retrieval-Augmented Generation)

- What is RAG?
- Embeddings
- Chunking
- Vector Databases
- FAISS
- ChromaDB
- Pinecone
- Similarity Search

Module 19: AI Frameworks

- LangChain
- LlamaIndex
- Prompt Templates
- Chains
- Retrieval
- Memory
- Output Parsers

Module 20: AI Agents

- Tool Calling
- Function Calling
- MCP
- LangGraph
- CrewAI
- Multi-Agent Systems

Module 21: AI System Design

- Architecture
- Authentication
- Memory
- Chat History
- Vector Database
- Monitoring
- Caching
- Security
- Scalability
- Cost Optimization

Module 22: Production AI Projects

Choose 2–3 of the following:

- AI Customer Support Chatbot
- AI Resume Analyzer
- AI Document Chat (RAG)

Final Capstone: Build an AI SaaS Platform

Features:

- User Authentication
- PostgreSQL Database
- Flask/FastAPI Backend
- REST APIs
- File Upload
- OpenAI/Gemini Integration
- RAG Pipeline
- AI Agents
- Chat History
- Docker

- Cloud Deployment