One-Month Training Roadmap

Python • Django • FastAPI • AI Foundations

Self-learning program designed for new interns to become productive contributors in backend and AI-driven projects.

WEEK 1 — Python & Engineering Foundations

Weekly Goals

- Build strong foundational Python knowledge
- Write clean, readable, modular code
- Understand Git workflow and code collaboration
- Work with JSON, CSV, files, exceptions, and modules
- Develop habit of documenting and testing code

Daily Breakdown

Day 1 — Setup, Python Basics, Git

- Install Python, IDE, Git, virtual environment tools
- Review variables, basic data types, input/output
- Write basic scripts: unit converter, calculator
- Create GitHub repo and push first script
- Submit end-of-day reflection and progress update

Day 2 — Data Structures & Functions

- Practice lists, sets, tuples, dictionaries
- Write functions with parameters, return types, type hints
- Implement word-frequency counter, list duplicate remover
- Solve 5 beginner problems from HackerRank or LeetCode
- Push work via feature branch and open PR

Day 3 — Modules, Packages, File Handling

- Learn imports, package structure, reusable modules
- Read/write JSON and CSV datasets
- Script: analyze student marks CSV and generate summary
- CLI Expense Tracker — add, delete, update, categorize
- Log errors to file using exception handling

Day 4 — Object-Oriented Programming & PEP-8

- Create multiple classes with relationships and inheritance
- Refactor previous project into OOP structure
- Follow PEP-8, naming conventions, docstrings
- Generate requirements.txt, organize folders properly
- Push work and request review from peers

Day 5 — Testing, Debugging & Final Week Project

- Learn pytest — assertions, fixtures, organizing tests
- Debug using print, breakpoints, and stack traces
- Final Project: Contact Book CLI with CRUD and JSON storage
- Write at least 10 tests including edge cases
- Prepare README with setup instructions and usage examples

WEEK 2 — Django Web Development

Weekly Goals

- Understand MVC/MVT architecture
- Build CRUD web applications
- Use Django ORM for relational databases
- Implement authentication, forms, templates, admin usage
- Document endpoints and deployment basics

Day 1 — Django Setup & Intro

- Install Django and start a new project
- Understand project structure, apps, URLs, views
- Create simple homepage, template rendering
- Push initial Django project to GitHub

Day 2 — Models, ORM & Migrations

- Create models with relationships
- Perform CRUD operations using ORM
- Use Django admin panel to manage data
- Practice filtering, sorting, pagination

Day 3 — Templates, Forms & Validations

- Create reusable templates and layouts
- Implement Django forms and CSRF protection
- Add create/update/delete UI for objects
- Display validation errors and success messages

Day 4 — Authentication & User Management

- Implement registration, login, logout
- Restrict views using login_required
- Customize User model and profiles
- Add role-based permissions

Day 5 — Django Week Project

- Build Task Management Web App
- Features: user login, CRUD tasks, categories, due dates
- Use Django admin for management
- Write 6 API views using Django REST Framework
- Deploy locally and share demo video link

WEEK 3 — FastAPI, APIs & Microservices

Weekly Goals

- Understand API-first backend development
- Design and document REST API endpoints
- Use Pydantic models for validation
- Implement CRUD operations with SQLAlchemy
- Write unit/integration tests for APIs
- Containerize application using Docker

Day 1 — FastAPI Basics & Routing

- Create FastAPI project and run local server
- Build GET, POST, PUT, DELETE endpoints
- Explore Swagger UI and ReDoc documentation
- Return JSON responses with Pydantic schemas

Day 2 — Database Integration

- Configure SQLite/PostgreSQL connection
- Implement SQLAlchemy models and migrations
- Write CRUD services for core entities
- Practice dependency injection patterns

Day 3 — Authentication, Security & Middleware

- Implement JWT authentication
- Protect routes using authentication dependencies
- Enable CORS, write custom middleware
- Discuss API rate limiting and security basics

Day 4 — Testing, Clients & Pagination

- Write API unit tests using pytest
- Use TestClient to simulate requests
- Build Python consumer script using httpx/requests
- Add filtering, sorting and pagination to endpoints

Day 5 — FastAPI Week Project

- Build Book Review REST API service
- Users login and post reviews with ratings
- Admin can approve or remove reviews
- Generate OpenAPI documentation
- Optional: Containerize with Docker

WEEK 4 — Intro to AI, LLMs & Capstone Project

Weekly Goals

- Get comfortable with Python-based AI workflow
- Use external APIs like OpenAI or HuggingFace
- Perform text summarization, classification, embeddings
- Understand vector databases and basic RAG concepts
- Build end-to-end full stack or multi-service project

Day 1 — AI Fundamentals in Python

- Review NumPy, Pandas, Matplotlib
- Load dataset and perform data cleaning
- Train simple ML model using scikit-learn
- Interpret accuracy and validation metrics

Day 2 — Working with LLM APIs

- Explore prompt engineering basics
- Call OpenAI or similar LLM APIs from FastAPI script
- Summarization, rewriting, classification examples
- Store prompts and responses for debugging

Day 3 — Mini AI Utilities

- Build keyword extractor API
- Build sentiment analysis CLI tool
- Convert text to bullet-point notes
- Add logging, exception handling, retry logic

Day 4 — Capstone Project Development

- Choose a project integrating Django + FastAPI + AI
- Define requirements, architecture, API contract
- Split responsibilities if working in teams
- Implement core features and track progress

Day 5 — Final Capstone Submission

- Complete documentation and README
- Record short demo video or presentation
- Submit GitHub repo with project board/tasks
- Mentor code review, scoring, improvement plan

Daily Deliverables Checklist

- Write 3–5 sentence progress update
- Push work to GitHub
- Submit weekly mini-project
- Ask doubts with clear context and attempts made
- Document learnings in README or notes file

Final Evaluation Criteria

- Consistency of daily learning and commitment
- Code quality, readability, maintainability
- Testing discipline and edge-case handling
- Understanding of Python, Django, FastAPI fundamentals
- Ability to build and document real projects
- Communication, autonomy, improvement mindset

End of Training Roadmap