🐍 Python Backend Development - Full Roadmap

🎯 Goal

Become a **strong backend developer using Python** within 1 year with full-stack project contribution potential, internship-readiness, and hackathon utility.

📅 1-Year Python Backend Developer Course Roadmap

Phase 1: Core Python (Weeks 1–4)

- Syntax, datatypes, loops, conditionals
- Functions, recursion, modules, file handling
- Lists, Tuples, Dictionaries, Sets
- String manipulation
- Error handling, exceptions

Phase 2: Intermediate Python + OOPs (Weeks 5–8)

- Object-Oriented Programming (classes, objects, inheritance, polymorphism)
- Functional programming basics (lambda, map, filter)
- List comprehensions
- Decorators, Generators
- Virtual environments (venv, pipenv)

Phase 3: Data Structures & Algorithms (Weeks 9–12)

- Arrays, Stacks, Queues, Linked Lists
- Trees, Graphs, Hashmaps
- Searching & Sorting
- Time & Space Complexity
- Problem solving on Leetcode (50+ problems minimum)

Phase 4: Web Development with Flask (Weeks 13–16)

- Flask basics: routes, templates, static files
- Jinja2 templating engine
- Forms and CRUD operations
- SQLite / SQLAlchemy
- Blueprints & app factory structure

Phase 5: Django Framework (Weeks 17–24)

- Project structure, MVT architecture
- Admin panel, authentication system
- Models, views, templates
- Forms, middleware, signals
- Django ORM deep dive
- Static/media files
- Deployment basics with PythonAnywhere/Render

Phase 6: MySQL + Database Mastery (Weeks 25–28)

- SQL queries (SELECT, INSERT, JOIN, etc.)
- Table design, keys, constraints
- ER Diagrams
- Connecting Django to MySQL
- DB schema design for real apps

Phase 7: REST APIs & Django REST Framework (Weeks 29–32)

- What is an API?
- HTTP methods, status codes
- JSON request/response
- Building APIs with DRF
- API Authentication: JWT, Token-based
- Testing APIs with Postman

Phase 8: Asynchronous Python & Background Tasks (Weeks 33–36)

- Basics of async, `asyncio`, `aiohttp`
- Celery + Redis for background tasks (emails, cron jobs)

Phase 9: Version Control + CI/CD + Docker (Weeks 37–40)

- Git, GitHub, branching
- Pull requests, code reviews
- Docker basics, Dockerizing Django apps
- GitHub Actions for CI/CD

Phase 10: Deployment & Security (Weeks 41–44)

- Hosting on Render, Railway, or Vercel
- NGINX, Gunicorn
- HTTPS, CSRF, CORS, Authentication
- Env files and secret management

Phase 11: System Design Basics (Weeks 45–48)

- Client-server model
- Load balancing, scaling, database sharding
- Caching, queues, rate limiting

Phase 12: Capstone Project + Resume Prep (Weeks 49–52)

- 2 major backend projects (with full auth, APIs, database, deploy)
- Document with README, video demo, GitHub commits
- Resume creation
- Mock interviews
- Open-source or hackathon participation

▯ Course Outcomes

- Able to create complete backend applications using Django & DRF
- Mastery over Python + MySQL-based projects
- Confident in real-world project contribution
- Build deployable REST APIs for any use case
- Competent in debugging, testing, documenting projects
- Intern/hackathon-ready

▯ Relevant Keywords

`Python`, `Backend`, `Django`, `Flask`, `MySQL`, `ORM`, `CRUD`, `Authentication`, `REST API`, `DRF`, `Git`, `Docker`, `CI/CD`, `Postman`, `Async`, `Redis`, `Celery`, `System Design`, `MVC`, `MVT`, `Version Control`, `Security`, `Testing`, `Leetcode`, `JWT`

▯ Recommended YouTube Channels & Web Resources

▯ YouTube Channels:

- Telusko
- CodeWithHarry (Python + Django)
- Amigoscode
- Tech With Tim
- Traversy Media
- freeCodeCamp
- The Net Ninja
- Corey Schafer (Python, OOP, Flask, Django)
- Bro Code (Django REST API)
- Akshay Saini (DSA clarity)

▯ Web Resources:

- W3Schools
- GeeksForGeeks
- RealPython
- Django Documentation
- DRF Documentation
- FullStackOpen
- Leetcode (practice)
- HackerRank
- Roadmap.sh

Suggested Projects:

1. Expense Tracker
2. Blog with user login and CRUD
3. Job board with APIs
4. Student Management System
5. Notes app with JWT auth
6. Chat app (Django + Websockets)
7. Portfolio backend (with DB and REST endpoints)

Start consistent. Don’t skip phases. Avoid tutorials after Phase 4. Build more. Learn by doing.