FastAPI Learning Path (Beginner to Advanced)

This roadmap covers all essential FastAPI topics with step-by-step progression from basic to advanced, including real-world projects.

Beginner Level (FastAPI Basics)

1. Introduction to FastAPI

- Overview and Features
- Why Choose FastAPI? (Speed, Asynchronous Support, Automatic Docs)
- Installation (`pip install fastapi uvicorn`)
- Setting Up a Virtual Environment

2. Creating Your First FastAPI App

- Writing a Basic FastAPI Application
- Running FastAPI with Uvicorn
- Exploring Interactive API Docs (Swagger & ReDoc)

3. Path Operations (API Routes)

- HTTP Methods: `GET`, `POST`, `PUT`, `DELETE`
- Path Parameters (`/{id}`)
- Query Parameters (`?name=John&age=25`)
- Request Body (`POST` with JSON data)

4. Request & Response Handling

- Request Body with Pydantic Models
- Validating Input Data
- Response Models (`response_model`)
- Customizing API Responses (`JSONResponse`, `FileResponse`)

5. Dependency Injection

- Understanding Dependencies
- Using `Depends()` for Code Reusability
- Database Connection as a Dependency

Intermediate Level (Expanding FastAPI Features)

6. Middleware & Background Tasks

- What is Middleware?
- Logging Requests & Responses
- Running Background Tasks (`BackgroundTasks`)

7. Database Integration (SQLAlchemy & Pydantic)

- Setting Up PostgreSQL/MySQL with FastAPI
- SQLAlchemy ORM Basics
- Defining Database Models (`Base`)
- CRUD Operations with SQLAlchemy

8. Authentication & Authorization

- User Authentication with OAuth2 & JWT
- Password Hashing with `bcrypt`
- Role-Based Access Control (RBAC)

9. Handling File Uploads & Downloads

- Uploading Files (`UploadFile`)
- Validating File Types & Sizes
- Serving Static Files (`StaticFiles`)

10. WebSockets for Real-Time Communication

- Setting Up a WebSocket Connection (`websockets`)
- Implementing a Real-Time Chat System
- Handling Multiple Clients

11. Background Task Processing with Celery

- Introduction to Celery
- Running Asynchronous Tasks in the Background
- Using Redis as a Task Queue

Advanced Level (Optimizing & Scaling FastAPI)

12. Security & CORS

- Enabling CORS (`CORSMiddleware`)
- Protecting API Endpoints with JWT
- Security Best Practices

13. Testing FastAPI Applications

- Writing Unit Tests with `pytest`
- Testing API Endpoints with `TestClient`
- Mocking Database Queries

14. Performance Optimization

- Using Redis for Caching
- Optimizing Database Queries
- Improving Response Time with `asyncio`

15. Deploying FastAPI Applications

- Dockerizing a FastAPI App (`Dockerfile`)
- Deploying on AWS/GCP/Render
- CI/CD Pipeline for FastAPI

Medium-Level Projects

Blog API with Authentication

- User Registration & Login (JWT Authentication)
- CRUD Operations for Blogs
- PostgreSQL & SQLAlchemy Integration

Real-Time Chat App with WebSockets

- WebSockets for Bidirectional Communication
- Storing Messages in a Database
- Multi-User Chat Room Functionality

Task Queue System with Celery & Redis

- Implementing Background Task Processing
- Integrating Celery with FastAPI
- Using Redis for Task Queueing

Image Upload & Processing API

- Uploading Images & Storing on Server
- Processing Images with OpenCV/PIL
- Serving Processed Images as API Responses

This roadmap provides a **structured learning approach** for mastering FastAPI, covering everything from **basic API development to real-time WebSocket applications and deployment**.