**30-Day FastAPI Challenge Content Kit**  
This includes for **each day**:

* **Topic** (what you’ll learn)
* **LinkedIn Hook** (short, scroll-stopping first line for recruiters/devs)
* **Medium Blog Title** (SEO-friendly)
* **Example Code Idea** (to show in posts)

**FastAPI 30-Day Industry-Oriented Content Plan**

**Phase 1 — Solidify Core API Development (Days 1–10)**

**Day 1 — Project Structure for Scalable FastAPI Apps**

* **LinkedIn Hook:** "Your API’s success starts with its folder structure."
* **Medium Title:** "Production-Ready FastAPI Project Structure for Scalability"
* **Code Idea:** Create app/ folder with main.py, routers/, models/, schemas/, services/, config.py.

**Day 2 — Path & Query Parameters with Validations**

* **LinkedIn Hook:** "Stop writing fragile endpoints. Validate them from day one."
* **Medium Title:** "Validating Path and Query Parameters in FastAPI with Pydantic"
* **Code Idea:** /books/{book\_id}?author=John with constraints like book\_id: int = Path(gt=0).

**Day 3 — Request Body Validation & Nested Models**

* **LinkedIn Hook:** "Garbage in, garbage out — validate your API requests like a pro."
* **Medium Title:** "Mastering Request Body Validation in FastAPI with Nested Models"
* **Code Idea:** Nested schema for Order containing list of OrderItem.

**Day 4 — CRUD Operations with SQLModel/SQLAlchemy**

* **LinkedIn Hook:** "If your API can’t talk to a database, it’s just an echo chamber."
* **Medium Title:** "Building CRUD APIs in FastAPI with SQLModel"
* **Code Idea:** Basic /users CRUD with SQLite.

**Day 5 — Pagination & Filtering for Large Data Sets**

* **LinkedIn Hook:** "Returning 10,000 rows in a single API call is not performance — it’s pain."
* **Medium Title:** "Implementing Pagination and Filtering in FastAPI APIs"
* **Code Idea:** /products?page=2&limit=10&category=electronics.

**Day 6 — Response Models & Custom Response Formats**

* **LinkedIn Hook:** "Inconsistent API responses break clients — fix it once and for all."
* **Medium Title:** "Standardizing API Responses in FastAPI with Response Models"
* **Code Idea:** Success/Error wrapper function with Pydantic.

**Day 7 — Error Handling & Exception Middleware**

* **LinkedIn Hook:** "The difference between a good API and a great API is how it fails."
* **Medium Title:** "Custom Exception Handling in FastAPI for Cleaner Error Responses"
* **Code Idea:** HTTPException, RequestValidationError, custom middleware.

**Day 8 — File Upload & Download API**

* **LinkedIn Hook:** "Yes, FastAPI can handle your 50MB Excel uploads."
* **Medium Title:** "Building File Upload and Download Endpoints with FastAPI"
* **Code Idea:** Upload CSV → parse with Pandas → return summary.

**Day 9 — Dependency Injection in FastAPI**

* **LinkedIn Hook:** "Stop hardcoding your database connections."
* **Medium Title:** "Dependency Injection in FastAPI: Cleaner, Testable Code"
* **Code Idea:** Inject DB session into routes using Depends.

**Day 10 — Environment Variables & Config Management**

* **LinkedIn Hook:** "Your API keys don’t belong in GitHub."
* **Medium Title:** "Managing Environment Variables in FastAPI with Pydantic Settings"
* **Code Idea:** .env file for database credentials, load via BaseSettings.

**Phase 2 — Authentication, Security & Middleware (Days 11–18)**

**Day 11 — JWT Authentication**

* **LinkedIn Hook:** "Secure your FastAPI like your job depends on it — because it does."
* **Medium Title:** "Implementing JWT Authentication in FastAPI"
* **Code Idea:** Login endpoint issuing JWT, verify via middleware.

**Day 12 — Role-Based Access Control (RBAC)**

* **LinkedIn Hook:** "Not all users are created equal — neither should their permissions be."
* **Medium Title:** "Implementing Role-Based Access Control in FastAPI"
* **Code Idea:** role: admin/user enforced at route level.

**Day 13 — OAuth2 with Google/GitHub Login**

* **LinkedIn Hook:** "Your users hate sign-up forms. Let them log in with Google."
* **Medium Title:** "Integrating OAuth2 Login with Google in FastAPI"
* **Code Idea:** Auth via Google OAuth2, store in DB.

**Day 14 — API Rate Limiting with Redis**

* **LinkedIn Hook:** "Protect your API from abuse before it’s too late."
* **Medium Title:** "Rate Limiting in FastAPI using Redis"
* **Code Idea:** Limit requests per IP.

**Day 15 — CORS, HTTPS, and Security Headers**

* **LinkedIn Hook:** "Don’t let a missing header be your API’s downfall."
* **Medium Title:** "Securing FastAPI with CORS, HTTPS, and Headers"
* **Code Idea:** Add CORSMiddleware, Strict-Transport-Security.

**Day 16 — Building Custom Middleware**

* **LinkedIn Hook:** "Middleware is your API’s secret weapon."
* **Medium Title:** "Building Custom Middleware in FastAPI for Logging and Tracking"
* **Code Idea:** Log request/response times.

**Day 17 — Background Tasks**

* **LinkedIn Hook:** "Not everything should block your API response."
* **Medium Title:** "Handling Background Tasks in FastAPI"
* **Code Idea:** Send email after order creation.

**Day 18 — Dependency Overrides for Testing Secure APIs**

* **LinkedIn Hook:** "Test your APIs without leaking your secrets."
* **Medium Title:** "Overriding Dependencies in FastAPI for Testing"
* **Code Idea:** Replace DB or auth dependency with mock.

**Phase 3 — Advanced Features & Real-Time (Days 19–24)**

**Day 19 — Async Programming**

* **LinkedIn Hook:** "If you’re still blocking requests, you’re wasting FastAPI’s potential."
* **Medium Title:** "Async Programming in FastAPI: When and How to Use It"
* **Code Idea:** Async DB calls with async SQLAlchemy.

**Day 20 — WebSockets**

* **LinkedIn Hook:** "Real-time updates in FastAPI are easier than you think."
* **Medium Title:** "Implementing Real-Time WebSockets in FastAPI"
* **Code Idea:** Chat app with multiple clients.

**Day 21 — Event-Driven Architecture**

* **LinkedIn Hook:** "Microservices love events — your FastAPI should too."
* **Medium Title:** "Event-Driven FastAPI with RabbitMQ"
* **Code Idea:** Publish order events to queue.

**Day 22 — Serving ML Models**

* **LinkedIn Hook:** "Your ML model is useless until it’s in production."
* **Medium Title:** "Deploying Machine Learning Models with FastAPI"
* **Code Idea:** Predict endpoint for scikit-learn model.

**Day 23 — File Streaming & Chunked Responses**

* **LinkedIn Hook:** "Send big files without breaking your API."
* **Medium Title:** "File Streaming in FastAPI with Chunked Responses"
* **Code Idea:** Stream 1GB CSV download.

**Day 24 — API Versioning**

* **LinkedIn Hook:** "Version your API or break your clients."
* **Medium Title:** "Implementing API Versioning in FastAPI"
* **Code Idea:** /v1/users and /v2/users with different fields.

**Phase 4 — Testing, Deployment & Optimization (Days 25–30)**

**Day 25 — Unit Testing with pytest**

* **LinkedIn Hook:** "If it’s not tested, it’s broken — you just don’t know it yet."
* **Medium Title:** "Unit Testing FastAPI APIs with pytest"
* **Code Idea:** Test CRUD routes with TestClient.

**Day 26 — API Documentation Best Practices**

* **LinkedIn Hook:** "Your API docs are your first impression — make them count."
* **Medium Title:** "Customizing Swagger and ReDoc in FastAPI"
* **Code Idea:** Add branding/logo to docs.

**Day 27 — Caching with Redis**

* **LinkedIn Hook:** "Speed is a feature — cache it."
* **Medium Title:** "Improving FastAPI Performance with Redis Caching"
* **Code Idea:** Cache /products list.

**Day 28 — CI/CD with GitHub Actions**

* **LinkedIn Hook:** "Deploying manually is so 2010."
* **Medium Title:** "Building a CI/CD Pipeline for FastAPI with GitHub Actions"
* **Code Idea:** Auto-deploy to EC2 on push.

**Day 29 — Deploying on Docker + AWS EC2**

* **LinkedIn Hook:** "Dockerize it before you deploy it."
* **Medium Title:** "Deploying FastAPI on AWS EC2 with Docker"
* **Code Idea:** Dockerfile + docker-compose.

**Day 30 — Deploying on Kubernetes**

* **LinkedIn Hook:** "Your API should scale as fast as your startup."
* **Medium Title:** "Deploying FastAPI on Kubernetes for Scalability"
* **Code Idea:** k8s manifests for API.