Python Full Stack – Advanced (Expert / Enterprise Level)

📘 Module 1: Advanced Python & Modern Architecture

Advanced Concepts

- Asyncio for asynchronous architecture
- Multithreading vs Multiprocessing
- Design Patterns (Factory, Singleton, Repository, MVC…)
- Clean Code Principles
- SOLID Principles
- Logging & Monitoring (Logging module, structlog)
- Profiling Python Code (cProfile, timeit)

Architecture Patterns

- Monolithic vs Microservices
- API Gateway Pattern
- Event-driven architecture

📘 Module 2: Advanced Django / Django REST Framework

Advanced Django

- Custom user models
- Custom middleware
- Signals (pre-save, post-save)
- Caching (site, template, per-view)
- Class-based views deep dive
- Complex model relationships
- Query optimization
  - select_related
  - prefetch_related
  - Raw queries

Advanced DRF

- Advanced Serializers (Nested, Polymorphic)
- Permissions & Throttling
- Versioning APIs
- DRF Filters, Pagination, Ordering
- API Exception Handling
- Rate Limiting

📘 Module 3: FastAPI (Modern High-Performance Python Backend)

- Async API development
- Path operations
- Pydantic Models (V2)
- Background tasks
- Advanced dependency injection
- JWT + OAuth2 auth
- WebSockets in FastAPI
- Integrating SQLAlchemy 2.0 with FastAPI
- Deploying FastAPI with Gunicorn/Uvicorn
- Comparison to Django REST Framework

📘 Module 4: Advanced Databases (SQL + NoSQL + Redis)

PostgreSQL (Advanced)

- Stored procedures
- Triggers
- Partitioning
- Indexing strategies
- Query optimization
- Normalization vs Denormalization

MongoDB (Advanced)

- Aggregation pipelines
- MapReduce
- Sharding & Replicas
- Mongo Atlas performance tuning

Redis (Enterprise Level)

- Caching API responses
- Redis Pub/Sub
- Redis for session storage
- Redis for rate limiting
- Redis Stream (Real-time pipelines)

📘 Module 5: Real-Time Applications

- Django Channels (WebSockets)
- FastAPI WebSockets
- Live dashboard
- Real-time notifications
- Chat application architecture
- Using Redis Pub/Sub for real-time systems

📘 Module 6: Microservices Architecture

- Breaking monolith into microservices
- API Gateway design
- Authentication between services
- Communication patterns:
  - REST
  - gRPC
  - Message queues
- Service discovery
- Circuit breakers

Message Queue Technologies

- RabbitMQ
- Kafka (Basic → Intermediate)
- Pub/Sub patterns

📘 Module 7: Background Jobs & Task Processing

- Celery + Redis
- Scheduling tasks
- Email sending
- Bulk operations
- Cron jobs
- Worker queues
- Monitoring Celery Tasks

📘 Module 8: Advanced Authentication & Security

- Token rotation
- Refresh token strategies
- HttpOnly cookies
- Rate limiting (Redis)
- XSS, CSRF, Clickjacking
- Helmet-like security for Django
- Password hashing algorithms (bcrypt, Argon2)
- Secure API design principles

📘 Module 9: Advanced Frontend (React or Angular)

React (Advanced)

- Redux Toolkit + RTK Query
- React Query for API caching
- Performance optimization
- Code splitting + Suspense
- React Optimizer (React 19+)
- Building admin dashboards

Angular (Advanced)

- NgRx State management
- Angular Signals
- Standalone Components
- Interceptors
- Lazy loading optimization
- Angular Material admin panels

📘 Module 10: File Storage & Cloud Integrations

- AWS S3 (Uploading files)
- Cloudinary (Image optimization)
- Presigned URLs
- File compression
- Handling large file uploads

📘 Module 11: DevOps for Full-Stack Developers

Docker

- Dockerfile optimization
- Docker Compose multi-container
- Docker networking
- Environment variables, .env management

CI/CD Pipelines

- GitHub Actions (multi-step pipeline)
- Automatic deployment
- Testing pipeline integration

Cloud Deployment

- AWS EC2
- AWS Elastic Beanstalk
- DigitalOcean apps
- Railway / Render advanced deployment
- Nginx reverse proxy configuration

📘 Module 12: Performance Optimization

- Django caching layers
- Redis optimization
- CDN usage
- Database indexing
- Query profiling
- Optimizing React/Angular builds
- Edge functions (Cloudflare Workers, Vercel Edge)

📘 Module 13: AI Integration (Optional)

- Using OpenAI API
- Building AI-powered features
- Text generation API
- Image generation API
- Chatbot integration into Django/FastAPI
- Sentiment analysis

📘 Module 14: Enterprise-Level Capstone Project

Students build a **production-grade full-stack application** using:

✓ Django/DRF or FastAPI

✓ React or Angular

✓ PostgreSQL + Redis

✓ WebSockets

✓ Celery + RabbitMQ

✓ Admin dashboard

✓ JWT auth

✓ Docker deployment

✓ CI/CD pipeline
✓ Cloud hosting

Example Advanced Projects:

- Real-time Trading Dashboard
- Chat + Notification System
- Multi-restaurant Food Delivery App
- SaaS Multi-Tenant CRM
- Hospital Automation System
- E-Commerce with Microservices