Pre-Implementation Technical Questionnaire for E-commerce Inventory Management Service

Architecture & Technology Stack Questions

Q1. Backend Technology Stack

Choice: Python with FastAPI

Rationale: FastAPI offers high performance, asynchronous capabilities, and excellent support for RESTful APIs. It allows rapid development and integrates smoothly with modern Python tools like Pydantic, SQLAlchemy, and async DB drivers.

Priority: High

Timeline: Week 1

Q2. Database Selection

Choice: PostgreSQL (Relational DB)

Why: Offers strong consistency, supports complex queries, full-text search, and is ACID-compliant. Excellent for structured data like SKUs, categories, and product relationships.

Priority: High

Timeline: Week 1

Q3. API Architecture Pattern

Choice: Modular monolith

Justification: Provides a clean separation of concerns without the complexity of microservices. Easier to deploy and manage early on while still allowing future scaling.

Priority: High

Timeline: Week 1

Data Model & Schema Questions

Q4. SKU Attributes Specification

Standard: Size, Color, Weight, Price, Inventory Count, Barcode/UPC, Dimensions

Dynamic: Support custom attributes per product category, Use flexible key-value pairs (JSON column in PostgreSQL)

Custom Attributes: Based on category-specific schema (e.g., "material" for apparel, "battery life" for electronics)

Priority: High

Timeline: Week 2

Q5. Category Hierarchy

Choice: Hierarchical categories

Max depth levels: 5

Priority: Medium

Timeline: Week 2

Q6. Data Types & Constraints

Category Name: Max Length 100, Alphanumeric & spaces

Category Description: Max Length 500, Plain text or Markdown

Product Name: Max Length 150, Alphanumeric & symbols

Product Description: Max Length 2000, Markdown supported

SKU Code: Max Length 64, Uppercase, hyphenated

Priority: High

Timeline: Week 2

Performance & Scalability Questions

Q7. Expected Scale

Categories: 500

Products: 50,000

SKUs: 500,000+

Concurrent Users: 1,000-5,000

Read/Write Ratio: 85% reads / 15% writes

Priority: High

Timeline: Ongoing

Q8. Pagination Strategy

Default: 20

Max: 100

Type: Cursor-based pagination

Priority: High

Timeline: Week 2

Search & Filtering Questions

Q9. Search Requirements

Search in names AND descriptions, Full-text search, Fuzzy search, Highlighting

Search Engine: Elasticsearch

Priority: High

Timeline: Week 4

Q10. Advanced Filtering

Multi-category, Price range, Date-based, Inventory status, Custom attribute

Priority: Medium

Timeline: Week 4

Security & Authentication Questions

Q11. Future Authentication

Design with future auth, Placeholder middleware

Planned: JWT tokens

Priority: Medium

Timeline: Week 5

Q12. Rate Limiting & Security

Rate limiting: 100 RPM, CORS, Input sanitization, SQL injection prevention

Different limits by endpoint: Yes

Priority: High

Timeline: Week 5

API Design Questions

Q13. Response Format Standard

Option B (Envelope Format)

Priority: Medium

Timeline: Week 2

Q14. Error Response Format

Use as-is, add trace/request ID

Priority: Medium

Timeline: Week 2

Q15. API Versioning

URL versioning (/v1/api/categories)

Priority: Medium

Timeline: Week 2

Testing & Quality Questions

Q16. Testing Strategy

Unit + Integration + E2E, Contract, Performance

Framework: PyTest

Priority: High

Timeline: Ongoing, starting Week 2

Q17. Code Quality Tools

Linting: Flake8

Formatting: Black

Static Analysis: SonarQube

Priority: Medium

Timeline: Week 3

Deployment & Environment Questions

Q18. Deployment Target

AWS (ECS Fargate, RDS, S3), Docker containers, Docker Compose (local)

Priority: High

Timeline: Week 5

Q19. Environment Configuration

Environments: 3 (dev, staging, prod)

Config: Env vars

Secrets: AWS Secrets Manager

Priority: High

Timeline: Week 5

Monitoring & Logging Questions

Q20. Observability Requirements

Track: Latency, Errors, DB performance, CPU/Memory, Business metrics

Logging: JSON

Integration: ELK Stack, Sentry

Priority: High

Timeline: Week 5-6

Q21. Health Checks

Liveness, DB check, Dependency check, System status

Priority: Medium

Timeline: Week 3

Business Logic Edge Cases

Q22. Deletion Strategy

Soft deletes, Audit trail

Priority: Medium

Timeline: Week 3

Q23. Concurrent Updates

Optimistic locking (versioning)

Priority: High

Timeline: Week 3

Q24. Bulk Operations

Bulk create/update/delete, CSV import/export

Priority: Medium

Timeline: Week 4

Q25. Data Consistency

Strict enforcement, Archive instead of delete

Priority: High

Timeline: Week 3

Additional Considerations

Q26. Caching

Redis, Application-level, DB query cache

Priority: Medium

Timeline: Week 4-5

Q27. File Upload

Product/category images, CSVs, S3

Priority: Medium

Timeline: Week 4

Q28. Integrations

External APIs, Inventory systems, Analytics, Notifications

Priority: Medium

Timeline: Week 5+

Completion Details

Estimated Timeline: 6-8 weeks (MVP)

Additional Notes:

- Multilingual support: future scope

- Admin dashboard to reuse same API

- IaC for infra (Terraform/CloudFormation)