Technical vs Functional Documentation

Difference Between Technical and Functional Documentation

1. Definition

- **Technical Documentation**: Provides detailed information about the system's architecture, code, APIs, deployment, and troubleshooting. It is mainly intended for developers, system administrators, and IT teams.
- **Functional Documentation**: Describes how the system works from a business perspective, detailing user requirements, workflows, and system functionalities. It is intended for business analysts, end-users, and stakeholders.

---

2. Purpose

- **Technical Documentation**: Helps developers build, maintain, and troubleshoot the system efficiently.
- **Functional Documentation**: Ensures that business requirements are met and guides users on how to use the system.

---

3. Audience

- **Technical Documentation**: Software developers, system architects, DevOps engineers, IT teams.
- **Functional Documentation**: Business analysts, project managers, product owners, end-users.

---

4. Content

- **Technical Documentation Includes:**
    - System architecture diagrams
    - API documentation
    - Database schema and entity relationships
    - Code structure and development guidelines
    - Troubleshooting and error handling
- **Functional Documentation Includes:**
    - Business requirements and user stories
    - System features and functionalities
    - User interface (UI) design and workflows
    - Use cases and scenarios
    - Acceptance criteria

---

5. Format & Examples

- **Technical Documentation Formats:**
    - API Reference (e.g., Swagger, Postman)
    - System Architecture Documents
    - Code Comments & Inline Documentation
    - Infrastructure & Deployment Guides
- **Functional Documentation Formats:**
    - Business Requirement Document (BRD)
    - Functional Specification Document (FSD)
    - User Manuals & Help Guides
    - Flowcharts & Use Case Diagrams

Key Differences at a Glance

Aspect	**Technical Documentation**	**Functional Documentation**
Purpose	Explains system architecture, code, and APIs	Defines system functionality and business processes
Audience	Developers, architects, IT teams	Business analysts, product managers, end-users
Focus	System design, coding, APIs, deployment	User requirements, workflows, system behavior
Formats	API docs, system diagrams, code structure	BRD, FSD, use cases, UI design
Examples	API reference, database schema	User manual, business rules document