## Technical documentation for developers section

Comprehensive guide for developers involved in the software project. It provides detailed information about the internal workings of the system, including code documentation, APIs, and system architecture.

1. Code Documentation: This involves detailed comments within the source code to explain its functionality, usage, and any important considerations for developers who might need to modify or maintain the code in the future. Code documentation typically includes:
    - Descriptions of classes, methods, and functions.
    - Explanation of input parameters and return values.
    - Details of any algorithms or complex logic implemented.
    - Notes on error handling and edge cases.

2. API Documentation: If the software includes APIs (Application Programming Interfaces) for interaction with other systems or modules, this documentation provides clear specifications for developers on how to use these APIs. It includes:
    - Endpoint URLs and methods (GET, POST, PUT, DELETE, etc.).
    - Request and response data formats (e.g., JSON, XML).
    - Authentication and authorization mechanisms.
    - Sample requests and responses.

3. System Architecture Documentation: This gives developers a high-level overview of the system's architecture, including its components, interactions, and dependencies. It typically includes:
    - Diagrams illustrating the overall system architecture (e.g., layered architecture, microservices architecture).
    - Description of each component/module and its responsibilities.
    - Details of communication protocols and data flow between components.
    - Dependencies on external services or libraries.

4. Database Schema Documentation: If the system involves a database, this documentation outlines the database schema design, including:
    - Tables, columns, and data types.
    - Relationships between tables (foreign key constraints).
    - Indexes and constraints.
    - Data migration or seeding scripts.