High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for the **ShopList - Inventory Management API**. The API provides backend inventory management for small e-commerce shops, supporting CRUD operations, pagination, and search, built with Node.js, Express, and PostgreSQL.

1. System Architecture Overview

Architecture Description:
The system follows a layered architecture with a RESTful API server interfacing between client applications and a PostgreSQL database. The API handles business logic, validation, and data persistence.

Component	Role
API Server	Handles HTTP requests, routes, validation, and business logic
Database Layer	Manages data storage, retrieval, and schema (PostgreSQL)
Client Interface	External clients (e.g., web/mobile apps) consuming the API

2. Component Interactions

Sequence Step	Description
1. Client → API Server	Sends HTTP requests (CRUD, search, pagination)
2. API Server → Database	Executes SQL queries for data operations
3. Database → API Server	Returns query results or status
4. API Server → Client	Responds with data or operation status (JSON)

3. Data Flow Overview

Data Flow	Description
Product CRUD Operations	Create, read, update, delete product records
Pagination & Search Requests	Query products with filters and pagination parameters
Validation & Error Handling	API validates input and returns errors as needed

4. Technology Stack

Layer	Technology/Framework
API Server	Node.js, Express
Database	PostgreSQL
ORM/Query Layer	(Optional) Knex.js or Sequelize
API Format	REST, JSON

5. Scalability & Reliability

- **Scalability:**
  Stateless API design enables horizontal scaling. Database can be optimized with indexing and connection pooling.
- **Reliability:**
  Input validation, error handling, and use of transactions ensure data integrity.
- **Security:**
  API endpoints should implement authentication and authorization (e.g., JWT), and sanitize inputs to prevent SQL injection.

End of Document