

# Project Design Phase-II

## Technology Stack

Date	7 Nov 2025
Team ID	NM2025TMID06700
Project Name	Medical inventory management
Marks	4 Marks

## ✓ Technology Stack – Medical Inventory Management System

### 1. Overview

The Medical Inventory Management System is a multi-tier application with **frontend, backend, database, integration, and deployment layers**. The chosen technologies ensure real-time inventory management, scalability, security, and ease of use.

### 2. Technology Stack by Layer

Layer	Purpose	Example Technology / Tools	Notes
<b>Frontend (Presentation Layer)</b>	User interface for Admin, Pharmacist, Medical Staff	React.js, Angular, Vue.js, or Flutter (for mobile)	Provides responsive

			web/mobile access
<b>Backend (Application Layer)</b>	Business logic, inventory management, request handling	Node.js (Express), Django (Python), Spring Boot (Java)	REST / GraphQL API for frontend communication
<b>Database (Data Layer)</b>	Stores inventory, user info, vendors, purchase orders, transactions	MySQL, PostgreSQL, or MongoDB	Supports real-time querying and reporting
<b>Integration Layer</b>	External systems and hardware	Barcode / RFID scanners, Email / SMS notification APIs, Vendor APIs	For real-time stock updates and alerts
<b>Security Layer</b>	Data encryption and access control	JWT / OAuth 2.0, SSL/TLS, Role-Based Access Control	Ensures data privacy and regulatory compliance
<b>Reporting &amp; Analytics</b>	Generate dashboards, trends, and reports	Tableau, Power BI, or Chart.js / Recharts	Helps in decision making and audits
<b>Deployment / Hosting</b>	Server hosting and environment management	AWS (EC2, RDS), Azure, Google Cloud, On-Premise	Cloud hosting enables scalability and remote access
<b>Version Control</b>	Code collaboration and management	Git + GitHub / GitLab / Bitbucket	Maintains codebase history and team collaboration
<b>Testing / QA Tools</b>	Automated and manual testing	Selenium, Jest, Postman, JMeter	Ensures system reliability and performance
<b>Backup / Recovery</b>	Data safety and disaster recovery	AWS S3 Backup, Cron Jobs, Database Replication	Protects against data loss

### 3. Recommended Architecture Pattern

- **3-Tier Architecture**
  - Presentation Layer (Frontend)
  - Business Logic Layer (Backend)
  - Data Layer (Database)
- **Optional:** Microservices for scalability in larger hospitals

## 4. Hardware Requirements

Hardware	Purpose
Barcode / RFID Scanners	Fast item identification and inventory updates
Computers / Tablets / Mobile Devices	Access for Admin, Staff, and Pharmacist
Server / Cloud Environment	Hosting backend, database, and notifications
Network Infrastructure	Reliable LAN / Wi-Fi for real-time communication

## 5. Key Considerations

- Ensure **cross-platform support** for web and mobile apps.
- Implement **secure and encrypted communication** (HTTPS, SSL).
- Ensure **scalability** to support multiple departments or hospitals.
- Plan for **hardware integration** (barcode / RFID) in workflows.