**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

**Date:** 27 June 2025  
**Team ID:**   
**Project Name:** Medical Inventory Management  
**Maximum Marks:** 4 Marks

**Technical Architecture:**

The deliverable includes the architecture diagram and the information below as per Table 1 and Table 2.

**Example: Medical Inventory Management System for Hospitals and Clinics**

**Guidelines followed:**

* Included all core processes such as inventory tracking, stock updates, order placement, and alerting as separate application logic blocks.
* Infrastructure is designed with clear demarcation between cloud services and local hosting for flexibility.
* Integrated third-party APIs for medicine verification and user authentication.
* Used cloud database and storage services for scalable data management.
* ML model interface is used for demand prediction based on historical stock usage.

**Table-1: Components & Technologies**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | User Interface | Web interface for inventory and orders management | HTML, CSS, JavaScript, React.js |
| 2 | Application Logic-1 | Inventory update and reorder logic | Python (Flask) |
| 3 | Application Logic-2 | Automated stock alert generation | Python + Scheduler (Celery/Crontab) |
| 4 | Application Logic-3 | Chatbot to assist medical staff | IBM Watson Assistant / Dialogflow |
| 5 | Database | Stores product, supplier, stock, and order details | MySQL / PostgreSQL |
| 6 | Cloud Database | Hosted DB for scalable cloud access | Amazon RDS / Google Cloud SQL |
| 7 | File Storage | Storage for invoices and reports | AWS S3 / Google Cloud Storage |
| 8 | External API-1 | User authentication | Firebase Auth / OAuth 2.0 |
| 9 | External API-2 | Medicine info verification | OpenFDA API |
| 10 | Machine Learning Model | Predict demand for medical supplies | Time Series Forecasting (scikit-learn) |
| 11 | Infrastructure | Deployed on cloud with backup on local | AWS EC2, Kubernetes, Local VM |

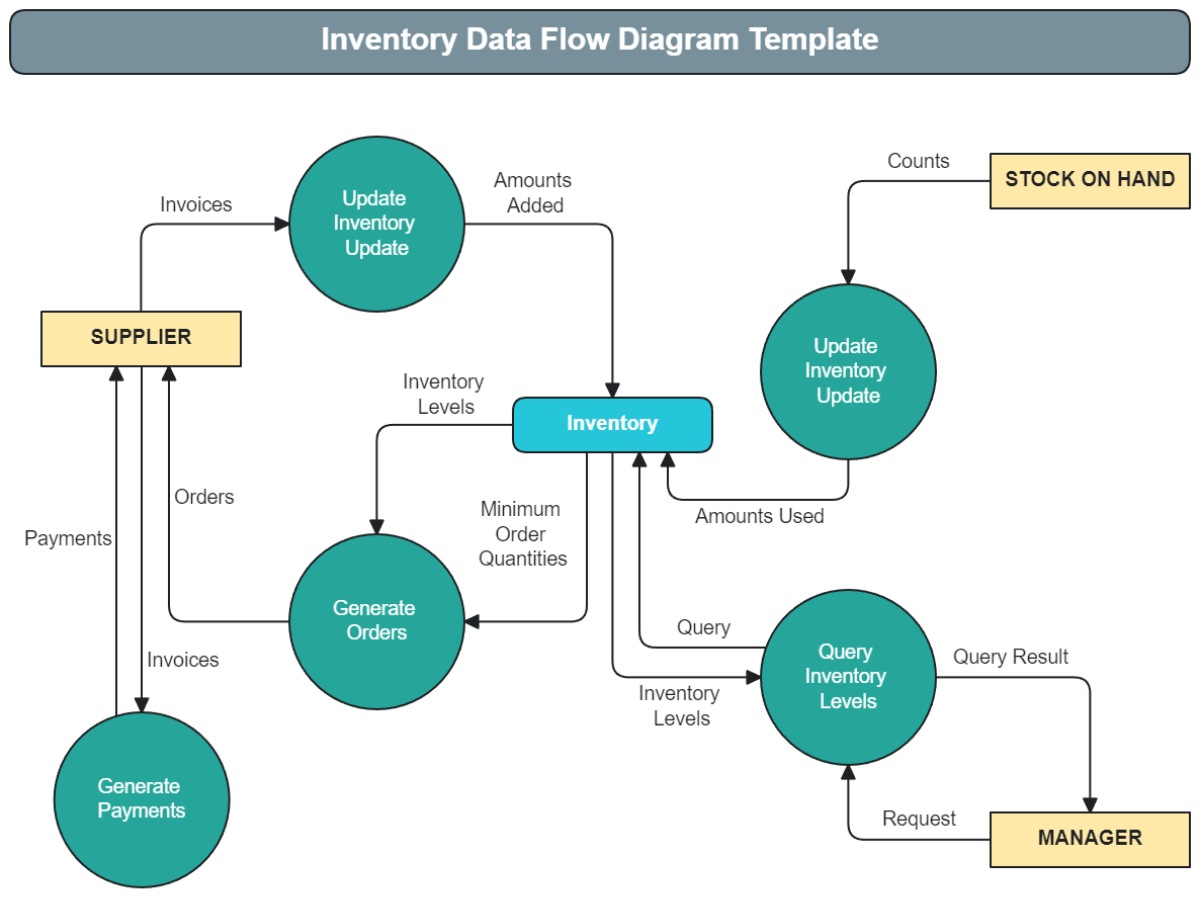
**Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Used for backend and frontend development | Flask, React.js |
| 2 | Security Implementations | Encrypted connections and secure API access | SSL/TLS, OAuth 2.0, IAM Roles |
| 3 | Scalable Architecture | Built using modular microservices and containers | Docker, Kubernetes |
| 4 | Availability | Load balancer and multi-zone cloud servers used for uptime | AWS Load Balancer, GCP Zones |
| 5 | Performance | Caching, async jobs, and CDN used for efficient access | Redis Cache, Cloudflare CDN, Async APIs |

**References:**

* <https://developer.ibm.com/patterns/smart-medical-inventory-system-using-ai-and-iot/>
* <https://c4model.com/>
* <https://www.ibm.com/cloud/architecture>
* <https://aws.amazon.com/architecture/>
* <https://firebase.google.com/docs/auth>
* <https://medium.com/@thirumagal2020/healthcare-inventory-management-system-using-iot-and-cloud-1c37f3c8a278>

**Example:**



* This data flow diagram outlines the data flow in an inventory management system. Administrators handle inventory and create reports. Employees, on the other hand, access stock data and update records. The cloud server syncs inventory data, processes shipments, and ensures accurate stock levels. The system sends invoices and tracks barcode requests, enhancing efficiency. This easy process keeps inventory levels just right. It cuts down on mistakes and boosts overall efficiency. Cloud services provide real-time updates and smooth communication. This DFD makes inventory management more effective and reliable.