Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	25 June 2025
Team ID	LTVIP2025TMID31434
Project Name	CRM Application for Jewel Management - (Developer)

Technical Architecture:

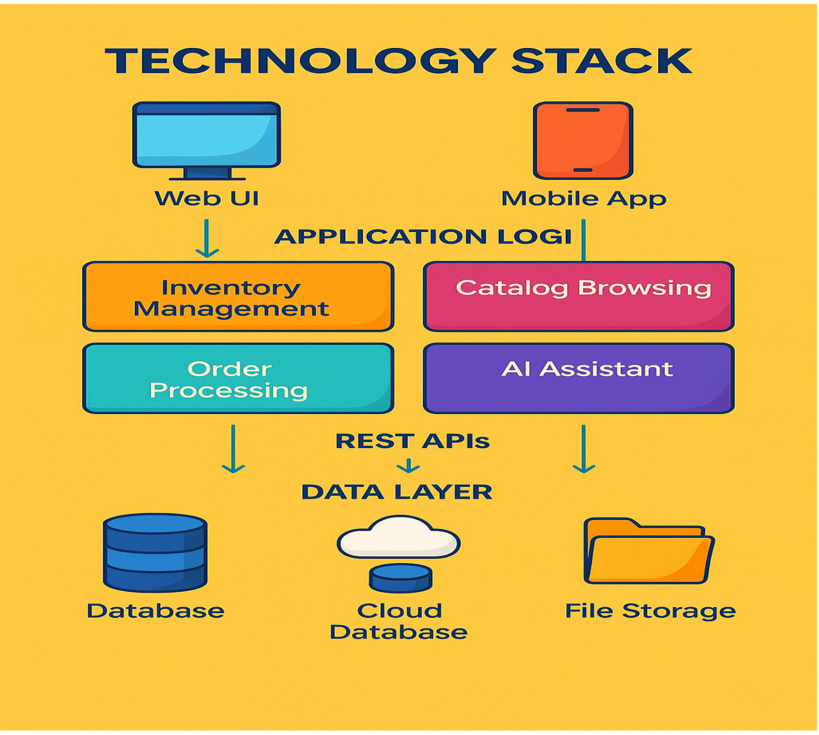
The architecture is designed to digitally streamline and optimize jewelry store operations like inventory tracking, catalog browsing, customer engagement, and order processing, using scalable, secure, and cloud-supported technologies.

Imagine a **3-tier architecture**:

1. Frontend (User Interface)
→ Users interact through **Web UI** or **Mobile App**

2. Backend (Application Logic Layer)
→ Handles inventory, customer catalog browsing, order requests, etc.

3. Data Layer (Database + Storage)
→ Stores product info, customer data, and logs



Example: "Where elegance meets efficiency!"

1️⃣USER INTERFACE (Frontend Layer)

This layer allows users to interact with the system either via a Web UI (for store owners/staff) or a Mobile App (for customers).

👨‍💼 Components:

- Web UI: Dashboard for store staff to manage inventory, orders
- Mobile App: Allows customers to browse catalogs, place orders

🌐 Technologies:

- HTML, CSS, JavaScript, ReactJS (Web)
- Flutter / React Native (Mobile App)

2️⃣APPLICATION LOGIC (Business Layer)

This middle layer contains all the core logic that powers features of the system.

⚙️ Modules:

- Inventory Management: Tracks stock levels and updates
- Catalog Browsing: Displays categorized jewelry to customers
- Order Processing: Handles orders, payments, and invoices
- AI Assistant: Uses AI (e.g., IBM Watson Assistant) to help users search and resolve queries

🔧 Technologies:

- Python / Node.js for backend services
- RESTful APIs to communicate between frontend and backend
- IBM Watson for chatbot/AI support

3️⃣DATA LAYER (Storage & Backend Layer)

This layer stores all the data required for the application to function.

🗂️ Components:

- Database: Stores product, user, and transaction data (e.g., MySQL)
- Cloud Database: Secure and scalable data storage on cloud (e.g., IBM Cloudant)
- File Storage: Stores images of jewelry, documents, receipts (e.g., AWS S3 / IBM Block Storage)

Reference:
https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

Table-1 : Components & Technologies:

S.No	**Component**	**Description**	**Technology**
1.	User Interface	Web UI & Mobile App for customers and store owners	HTML, CSS, ReactJS / Flutter
2.	Application Logic-1	Inventory management, catalog upload & edit logic	Python / Node.js
3.	Application Logic-2	Voice-based search for customers	IBM Watson Speech-to-Text (STT)
4.	Application Logic-3	Chatbot for customer queries	IBM Watson Assistant
5.	Database	Store info, product details, user data	MySQL (for structured data)
6.	Cloud Database	Cloud-stored inventory backups	IBM Cloudant
7.	File Storage	Store images, invoices, jewelry designs	IBM Block Storage / AWS S3
8.	External API-1	Fetch weather info for delivery planning (optional)	IBM Weather API
9.	External API-2	Aadhar-based customer KYC validation	Aadhar API
10.	Machine Learning Model	Recommend jewelry based on past purchases & search patterns	Product Recommendation ML model (Python)
11.	Infrastructure (Server / Cloud)	Cloud deployment with scalable container setup	IBM Cloud Foundry / Kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	ReactJS, Flask, TensorFlow, Node.js, Express	ReactJS, Python, TensorFlow, Flask
2.	Security Implementations	Access control, encryption, secure login	SHA-256, OAuth 2.0, IAM, SSL, OWASP
3.	Scalable Architecture	3-tier with microservices for catalog, inventory, chat	Docker, Kubernetes
4.	Availability	Distributed servers with load balancers	IBM Load Balancer, Multi-zone nodes
5	Performance	CDN for UI assets, caching for frequently accessed items, async APIs	Redis, Cloudflare CDN, Async Python

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d