**Project Design Phase-II**

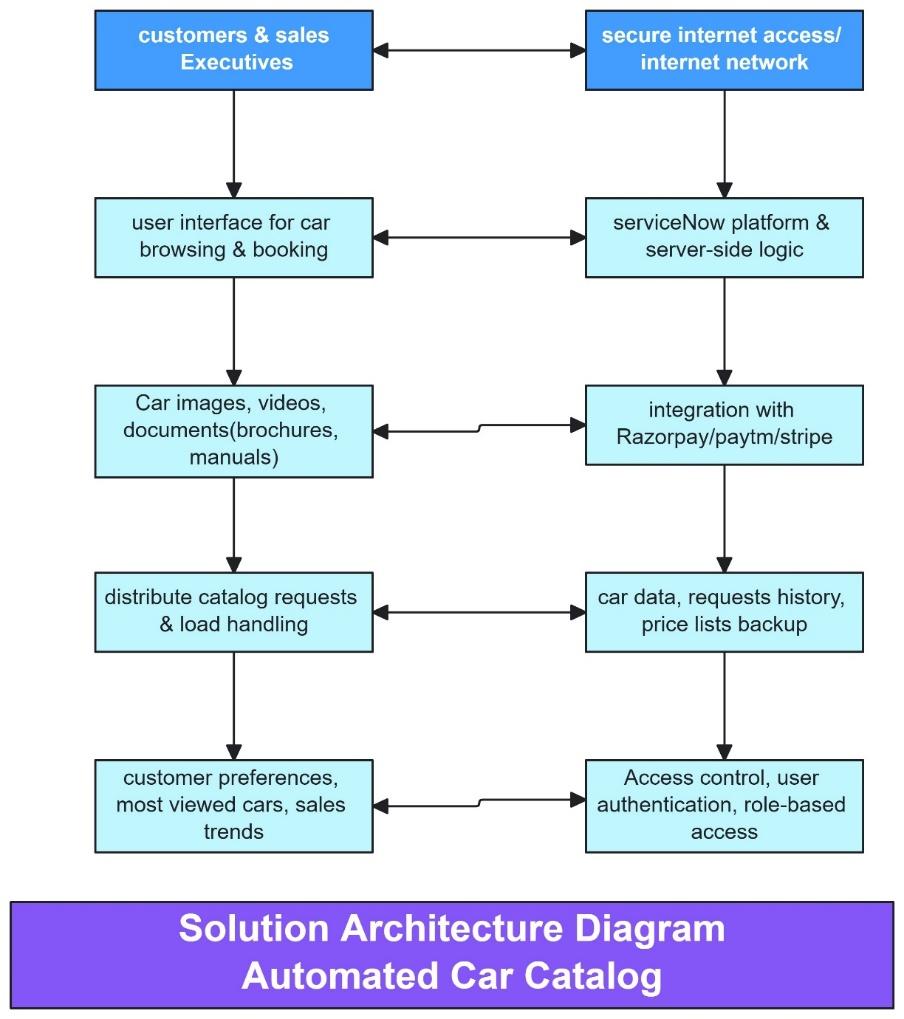
**Technology Stack (Architecture & Stack)Technical Architecture:**

| Date | 31 june 2025 |
| --- | --- |
| Team ID | LTVIP2025TMID28960 |
| Project Name | Automated Car Catalog System For Enhanced Showroom Management |
| Maximum Marks | 4 Marks |

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example:

Automated car catalog system for enhanced showroom management (e.g., ServiceNow)



**Table 1: Component Details**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | User interacts with car catalog via service portal to browse and request car models | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | Handles catalog item form logic, car model details, field validation and submission control | JavaScript (Client Scripts, UI Policies) |
|  | Application Logic-2 | Business logic for booking, approval, status updates, notifications | Flow Designer, Script Includes, UI Actions |
|  | Application Logic-3 | Integrates car inventory and showroom database for updates | Business Rules, Integration Scripts (MID Server) |
|  | Database | Stores car specifications, availability, user preferences, booking history | MySQL / NoSQL / CMDB (ServiceNow) |
|  | Cloud Database | Dynamic data like inventory, bookings, user details across multiple branches | ServiceNow Cloud CMDB |
|  | File Storage | Stores documents(e.g, id oroof, payment receipts) uploaded by customers | ServiceNow File Attachment System / Block Storage |
|  | External API-1 | Integration with payment systems(e.g., razorpay, paytm) | Razorpay API, UPI API |
|  | External API-2 | Integrates with RTO or government vehicle registration systems | Vahan API, Digilocker API |
|  | Machine Learning Model | Suggests suitable car models based on preference, budget and usage | Python ML Model served via Flask / TensorFlow API |
|  | Infrastructure | Hosted on ServiceNow cloud with optional local showroom network integrations | Cloud: ServiceNow SaaS; Local: MID Server, VPN |

**Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1. | Open-Source Frameworks | Frontend frameworks and backend logic for catalog | AngularJS / ReactJS, JavaScript, Python |
|  | Security Implementations | Role-based access, secure payment, audit trail | | SHA-256, HTTPS, OAuth 2.0, IAM Policies (ServiceNow) | | --- |  |  | | --- | |
|  | Scalable Architecture | Modular catalog with reusable scripts, and workflows | 3-tier architecture with loosely coupled logic |
|  | Availability | Highly availability across branches and cloud regions | Load Balancer, Distributed Cloud Architecture |
|  | Performance | Optimized forms, real-time data sync, asynchronous workflows | Client-side validation, Business Rule optimization |