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

| Date | 31 January 3035 |
|---|---|
| Team ID | PNT2025TMID04160 |
| Project Name | Vehicle Management System |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

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

**Example: Order processing during pandemics for offline mode**

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

| Sr.no | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Web-based and mobile interface for vehicle management | HTML, CSS, JavaScript, Salesforce Lightning, React.js |
| 2. | Application Logic-1 | Core business logic for vehicle management (e.g., user registration, sales tracking) | Salesforce Apex |
| 3. | Application Logic-2 | AI-based recommendation system for vehicle suggestions | Salesforce Einstein AI |
| 4. | Application Logic-3 | Customer Support Chatbot | Salesforce Einstein Bot |
| 5. | Database | Stores vehicle, sales, and customer data | Salesforce Object Database |

| | | | |
|---|---|---|---|
| 6. | Cloud Database | Cloud-based database for scalable data storage | Salesforce Database (PostgreSQL or NoSQL) |
| 7. | File Storage | Stores invoices, documents, and images of vehicles | Salesforce Content Library, AWS S3 |
| 8. | External API-1 | Vehicle history check API (to get previous records of vehicles) | National Vehicle Registry API |
| 9. | External API-2 | Maps and Location API for tracking vehicle deliveries | Google Maps API. |
| 10. | Machine Learning Model | AI-based pricing predictions and vehicle demand forecasting | Salesforce Einstein AI, TensorFlow |
| 11. | Infrastructure (Server / Cloud) | Cloud-based deployment of Salesforce application | Salesforce Cloud, AWS, Kubernetes |

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Frameworks used for frontend, backend, and AI services | React.js, TensorFlow, Node.js |
| 2. | Security Implementations | Role-based access, data encryption, OAuth authentication, firewall protection | SHA-256 encryption, OAuth, Salesforce IAM, OWASP standards |
| 3. | Scalable Architecture | Supports multi-tier architecture and microservices for better performance | Salesforce Microservices, Kubernetes |

| | | Ensures high availability with auto-scaling and load balancing | AWS Load Balancer, Multi-zone Deployment |
|---|---|---|---|
| 4. | Availability | | |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 5. | Performance | Optimized response time using caching and CDNs for fast data retrieval | Redis Cache, Cloud CDN |

**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**