

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

Date	01 NOV 2025
Team ID	NM2025TMID04024
Project Name	Garage Management System
Maximum Marks	4 Marks

Technical Architecture:

This deliverable describes the technical architecture for a Garage Management System (GMS) — a web + mobile-enabled application to manage vehicle servicing, appointments, mechanics, inventory (spare parts), billing, and customer records.

Architectural highlights:

- Clear separation of UI, business logic, persistent storage, and integrations.
- Hybrid deployment: frontend and mobile clients delivered via CDN; backend hosted on cloud (managed services) with optional on-premise components for sensitive data.
- Asynchronous job processing for long-running tasks (invoice generation, bulk imports, notification delivery).

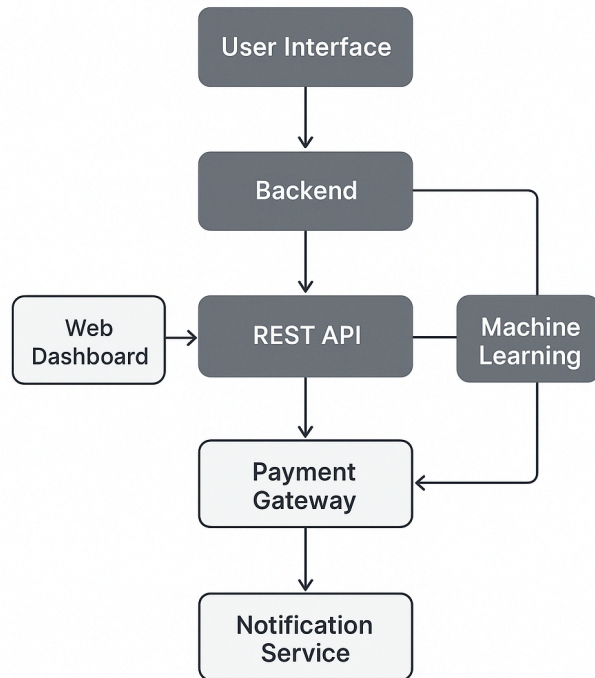
Example: Order processing during pandemics for offline mode

Reference: “GARAGE MANAGEMENT SYSTEM” (IRJMETS, March 2025) —

https://www.irjmets.com/uploadedfiles/paper//issue_3_march_2025/69215/final/fin_irjmets1742019066.pdf

Garage Management System

Technology Stack (Architecture & Stack)



Guidelines:

- Include all major processes as application logic blocks (booking, job management, inventory, billing, reporting).
- Provide infrastructure demarcation (Client / Edge / Cloud / On-prem as applicable).
- Indicate external interfaces (payment gateway, SMS/Email provider, vehicle data APIs, supplier APIs).
- Indicate data storage components and backup strategy.
- Indicate any ML/analytics interfaces (e.g., predictive maintenance or parts demand forecasting) if applicable.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web and mobile UI for customers to book services, view status and pay online.	React (Web), React Native or Flutter (Mobile), Hosted on CDN (Netlify/Vercel)
2.	User Interface - Admin/Mechanic	Web dashboard for admin & mechanics to manage jobs, inventory, and reporting.	React + Material/Tailwind; Admin panel (React + AntD)
3.	Application Logic - Booking & Scheduling	Handles appointment slots, conflicts, reminders and ETA calculations.	Django
4.	Application Logic - Job Management	Create/assign/track job cards, time logs, and status transitions.	Backend service (REST/
5.	Application Logic - Inventory Management	Track spare parts, stock levels, reorder thresholds, supplier orders.	Service with DB-backed transactions (Postgres),
6.	Application Logic - Billing & Payments	Generate invoices, apply taxes/discounts, integrate payment gateway..	Integrated with Stripe/Paytm/PayU
7.	Database	Stores users, jobs, inventory, invoices, payments, and audit logs.	PostgreSQL
8.	Cloud Database / Managed Services	Managed DB, backups, and scaling.	SQL / Azure Database
9.	File Storage	Store invoice PDFs, images (vehicle photos), and logs.	Google Cloud Storage
10.	External API - Payment Gateway	Payment processing and refunds.	PayPal / Razorpay / PayU

11.	External API - SMS/Email	Notifications for appointments and job updates.	AWS SNS + SES / SendGrid
12.	Infrastructure	Hosting & orchestration, CI/CD.	Kubernetes, CI: GitHub Actions / GitLab CI
13.	Monitoring & Logging	Observability, alerts, and crash reporting.	Prometheus + Grafana, ELK stack or CloudWatch
14.	Security	Auth, RBAC, encryption, audit trails.	OAuth2 / OpenID Connect

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Many components use OSS frameworks; core platform may use both OSS and managed services.	React, Node.js, PostgreSQL, Kubernetes (open-source)
2.	Security Implementations	Role-based access controls, input validation, CSRF/XSS protections, secure storage of credentials,	OAuth2, HTTPS, Vault/Secrets Manager, WAF, periodic security audits
3.	Scalable Architecture	Design for peak-day load (multiple concurrent bookings and job updates). Use horizontal scaling and stateless services.	Containerized services, auto-scaling groups, managed DB replicas
4.	Availability	High availability for the scheduling and billing services; planning for failover and backups.	Multi-AZ deployments, read-replicas, automated backups and point-in-time recovery
5.	Performance	Responsive UI, low-latency job updates, optimized DB queries for reporting.	Caching (Redis), indexed DB schema, async processing for heavy tasks

6.	Extensibility	Ability to add features: loyalty programs, fleet management, supplier portals.	Well-versioned APIs
7.	Data Retention & Privacy	Retain customer and vehicle data per regulations; provide data export and deletion workflows.	Data retention policies, GDPR/Local compliance workflows, audit logs
8.	Backup & Disaster Recovery	Regular backups and defined RTO/RPO for critical data.	Automated DB backups, snapshot retention policies
9.	Observability & Alerting	Track key metrics: job throughput, average repair time, stockouts, payment failures.	Prometheus metrics, Grafana dashboards, alert rules, incident runbooks