Tittle: GARAGE MANAGEMENT SYSTEM

Team ID: LTVIP2025TMID31613

Team Size: 4

Team Member 1: Pati Ravindra Venkata Saikrishna

Email: ravindrapati004@gmail.com

Team member 2: Mamidi Veera Nagendra

Email: mamidiveeranagendra@gmail.com

Team member 3: Mohammed Alisha

Email:alishamohammed23102005@gmail.com

Team member 4: Kongarapu Mukesh

Email:kongarapumukesh@gmail.com

**PROJECT INDEX:**

1. IDEATION PHASE:

Problem Statement

Empathy Map Canvas

Brainstorming

1. REQUIREMENT ANALYSIS:

Customer Journey map

Solution Requirement

Data Flow Diagram

Technology Stack

1. PROJECT DESIGN: Problem Solution Fit

Proposed Solution Solution Architecture

1. PROJECT PLANNING:

Project Logic

Project Planning

1. FUNCTIONAL AND PERFORMANCE TESTING: Performance Testing
2. ADVANTAGES & DISADVANTAGES
3. CONCLUSION

GitHub & Project Demo Link

**Ideation Phase**

**Brainstorm & Idea Prioritization Template**

|  |  |
| --- | --- |
| Date | 24 June 2025 |
| Team ID | LTVIP2025TMID31613 |
| Project Name | Garage Management System |
| Maximum Marks |  |

🧠 **Brainstorming Matter for Garage Management System**

1. **Stakeholders** 
   * Garage Owners

* + Mechanics/Technicians

* + Customers

* + Suppliers/Vendors

* + Service Advisors

1. **Problems to Solve** 
   * Manual tracking of service records

* + Missed appointments or inefficient booking

* + Lack of inventory control

* + Poor customer communication

* + Unclear billing/invoicing

* + No reminders for recurring services

1. **Core Features to Consider** 
   * **Customer Management**

* + 1. Add, view, update customer details

○ Vehicle history and preferences

* + **Job/Work Order Management**

* + 1. Create new jobs, assign to mechanics

○ Status tracking (Pending, In Progress, Completed)

* + **Appointment & Booking System**

* + 1. Schedule service

○ Reminders (SMS/Email)

* + **Inventory Management**

* + 1. Track spare parts and tools

○ Auto-replenish alerts

* + **Billing & Invoicing**

* + 1. Generate bills with labor + parts

○ Print/export options

○ Payment tracking

* + **Reports & Analytics**

* + 1. Revenue reports

○ Mechanic performance

○ Most serviced vehicle types

* + **Mobile Compatibility**

* + 1. Customer app for booking & tracking

○ Mechanic app for job updates

1. **Nice-to-Have Features** 
   * Loyalty points or discounts

* + GPS integration for pickup/delivery

* + Digital vehicle health card

* + Chat support or helpdesk

* + Feedback system

1. **Technology Stack Ideas** 
   * Backend: Node.js / Django / Laravel

* + Frontend: React / Angular / Vue

* + Database: PostgreSQL / MongoDB / MySQL

* + Mobile: Flutter / React Native

* + Notifications: Twilio / Firebase

* + Hosting: AWS / Azure

**Ideation Phase**

**Empathize & Discover**

|  |  |
| --- | --- |
| Date | 24 June 2025 |
| Team ID | LTVIP2025TMID31613 |
| Project Name | Garage Management System |
| Maximum Marks |  |

# 🔍 Empathy Map – Garage Management System

👤 **User Persona:**

* Garage Owner / Manager

* Technicians / Mechanics

* Customers (vehicle owners)

# 🧠 Think & Feel

* **Garage Owner:**

* 1. Wants better tracking of jobs, payments, and customer details.

○ Worried about missed service deadlines and poor customer retention.

○ Seeks automation to reduce paperwork.

* **Technicians:**

* 1. Thinks about efficient job assignment.

○ Feels stressed by unclear job priorities or tool shortages.

* **Customers:**

* 1. Thinks about getting reliable service on time.

○ Feels anxious about being overcharged or delays.

# 👀 See

* **Garage Owner:**

* 1. Sees messy records and lack of visibility into technician workloads.

○ Notices lost revenue due to inefficient systems.

* **Technicians:**

* 1. Sees outdated job tracking methods.

○ Notices delays due to poor part management.

* **Customers:**

* 1. Sees disorganized garages or lack of transparency in repair process.

# 🗣 Say & Do

* **Garage Owner:**

* 1. Talks about improving business and customer satisfaction.

○ Tries different manual systems or spreadsheets.

* **Technicians:**

* 1. Asks for clearer instructions and part availability.

○ Logs work manually or through phone calls.

* **Customers:**

* 1. Ask for service updates and cost estimates.

○ Often follow up repeatedly for vehicle status.

# 👂 Hear

* **Garage Owner:**

* 1. Hears complaints from customers about delays or communication gaps.

○ Gets feedback from staff on workload issues.

* **Technicians:**

* 1. Hears frustration about part delays or unclear work orders.

○ Gets pressure from owners and customers.

* **Customers:**

* 1. Hears reviews or word-of-mouth about reliable vs. unreliable garages.

○ Influenced by service experiences of friends/family.

# 😟 Pains

* Manual record keeping

* Missed service deadlines

* Miscommunication with customers

* Overloaded technicians

* No service history tracking

# 🌟 Gains

* Digital job cards and reminders

* Transparent billing and service history

* Improved technician assignment

* Higher customer satisfaction and return visits

**Ideation Phase**

**Define the Problem Statements**

|  |  |
| --- | --- |
| Date | 24 June 2025 |
| Team ID | LTVIP2025TMID31613 |
| Project Name | Garage Management System |
| Maximum Marks |  |

# 🔧 Problem Statement: Garage Management System

In today's fast-paced automotive industry, many garages and service centers continue to rely on manual or semi-digital methods to manage their daily operations. These outdated systems often lead to inefficiencies such as:

* Inaccurate or lost customer and vehicle service records

* Poor scheduling and missed service appointments

* Lack of transparency in billing and inventory usage

* Difficulty in tracking mechanic performance and job status

* Limited customer engagement and feedback mechanisms

As garages grow in size or handle more customers, these issues compound and reduce service quality, customer satisfaction, and profitability.

There is a critical need for a centralized, digital **Garage Management System (GMS)** that simplifies and automates key processes like customer and vehicle management, job card handling, inventory tracking, billing, and appointment scheduling. Such a system should enhance operational efficiency, improve customer experience, and provide real-time insights to support informed decision-making.

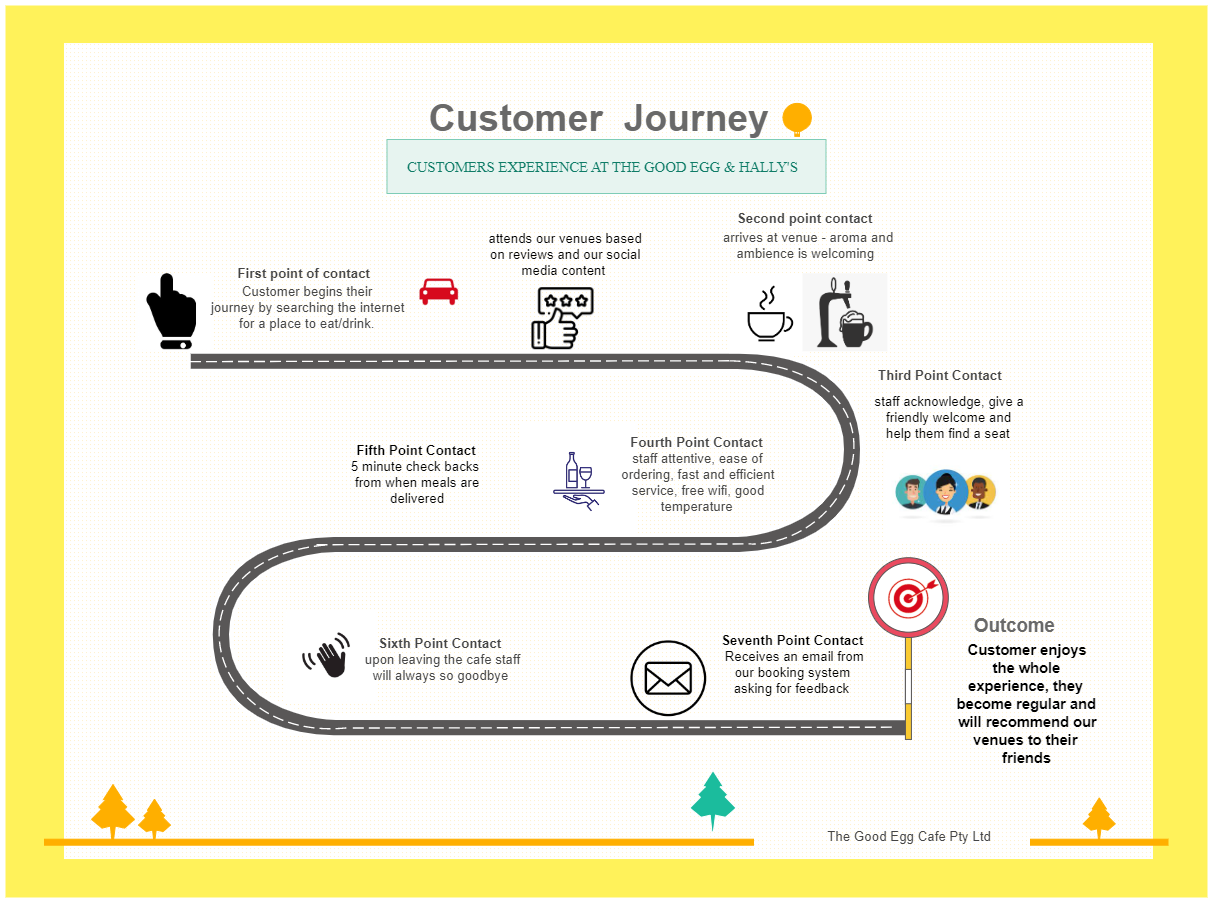
**Project Design Phase -||**

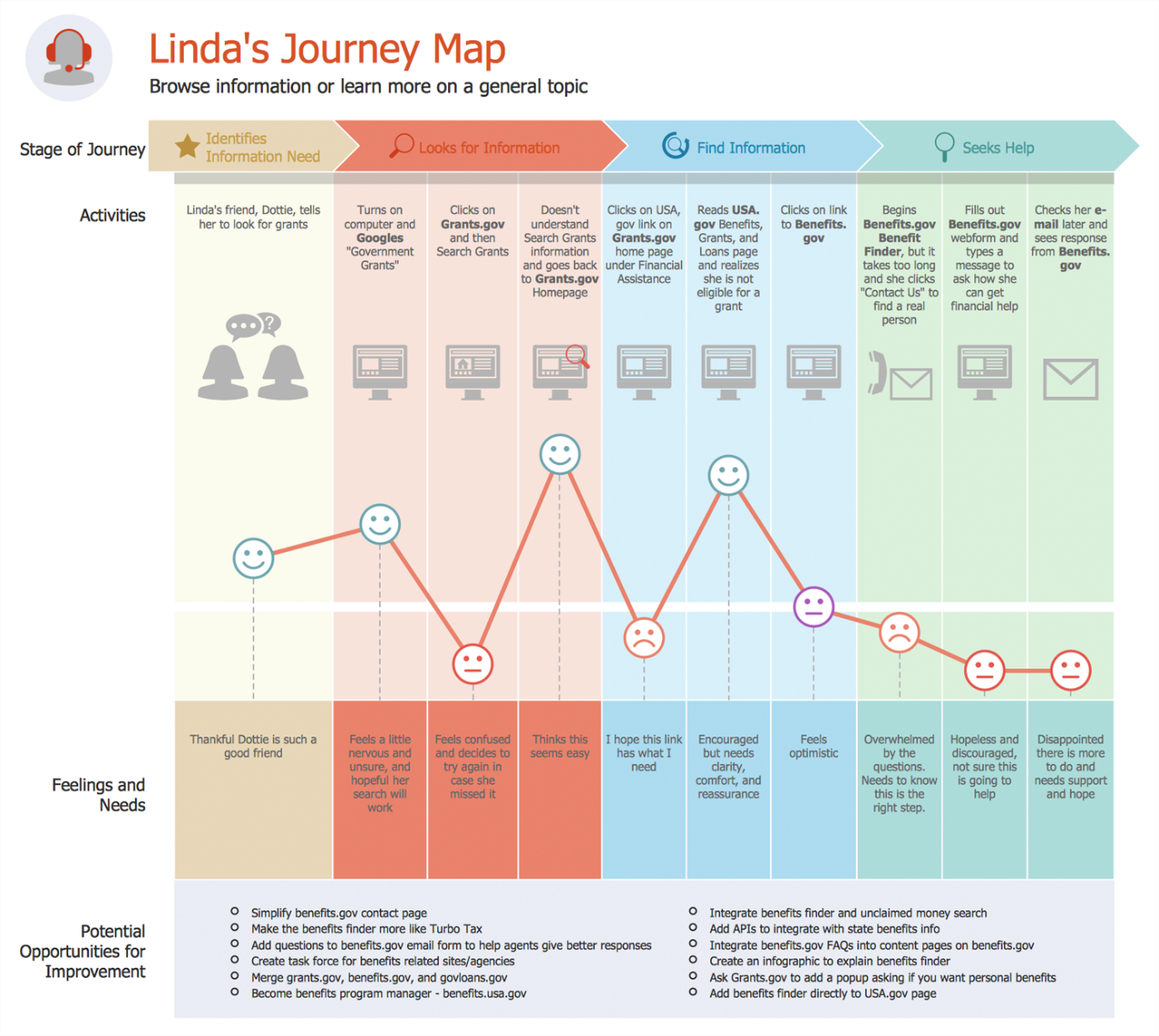
**Customer Journey Map**

|  |  |
| --- | --- |
| **Date** | 24 June2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Customer Journey Map:**

A customer journey map in a garage management system helps visualize the entire experience a customer goes through—from the moment they consider bringing their vehicle in, all the way to post-service follow-up. It's a fantastic tool for spotting pain points, improving customer satisfaction, and streamlining internal operations.





**Essential Elements of a Customer Journey Map:**

1. **Customer Persona**

* A fictional but realistic profile of your ideal customer
* Includes demographics, goals, challenges, and behaviors

2. **Stages of the Journey**

Common stages might include:

* **Awareness**: Customer becomes aware of your product or service
* **Consideration**: Research and comparison phase
* **Purchase**: Final decision and transaction
* **Retention**: Continued use and loyalty-building
* **Advocacy**: Recommending your brand to others

3. **Customer Actions**

* What the customer does at each stage
* Examples: browsing your website, reading reviews, contacting support

4. **Emotions & Pain Points**

* Emotional highs and lows throughout the journey
* Frustrations, confusion, satisfaction, or delight

5. **Touchpoints**

* Interactions between customer and brand
* Can include emails, ads, app usage, in-store visits, social media, etc.

6. **Internal Processes**

* What happens behind the scenes at each stage
* Helpful for aligning internal teams with customer experience

7. **Opportunities for Improvement**

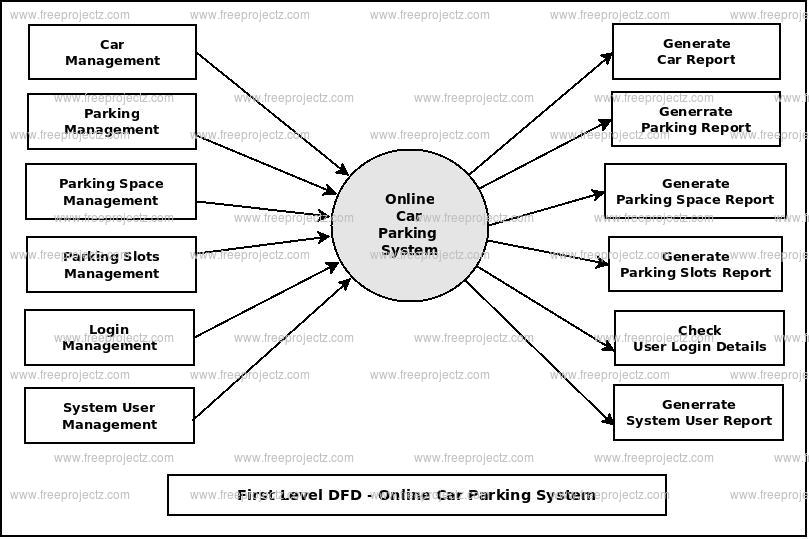
* Gaps or friction points you can enhance
* New strategies to optimize experience and reduce churn

**Project Design Phase -||**

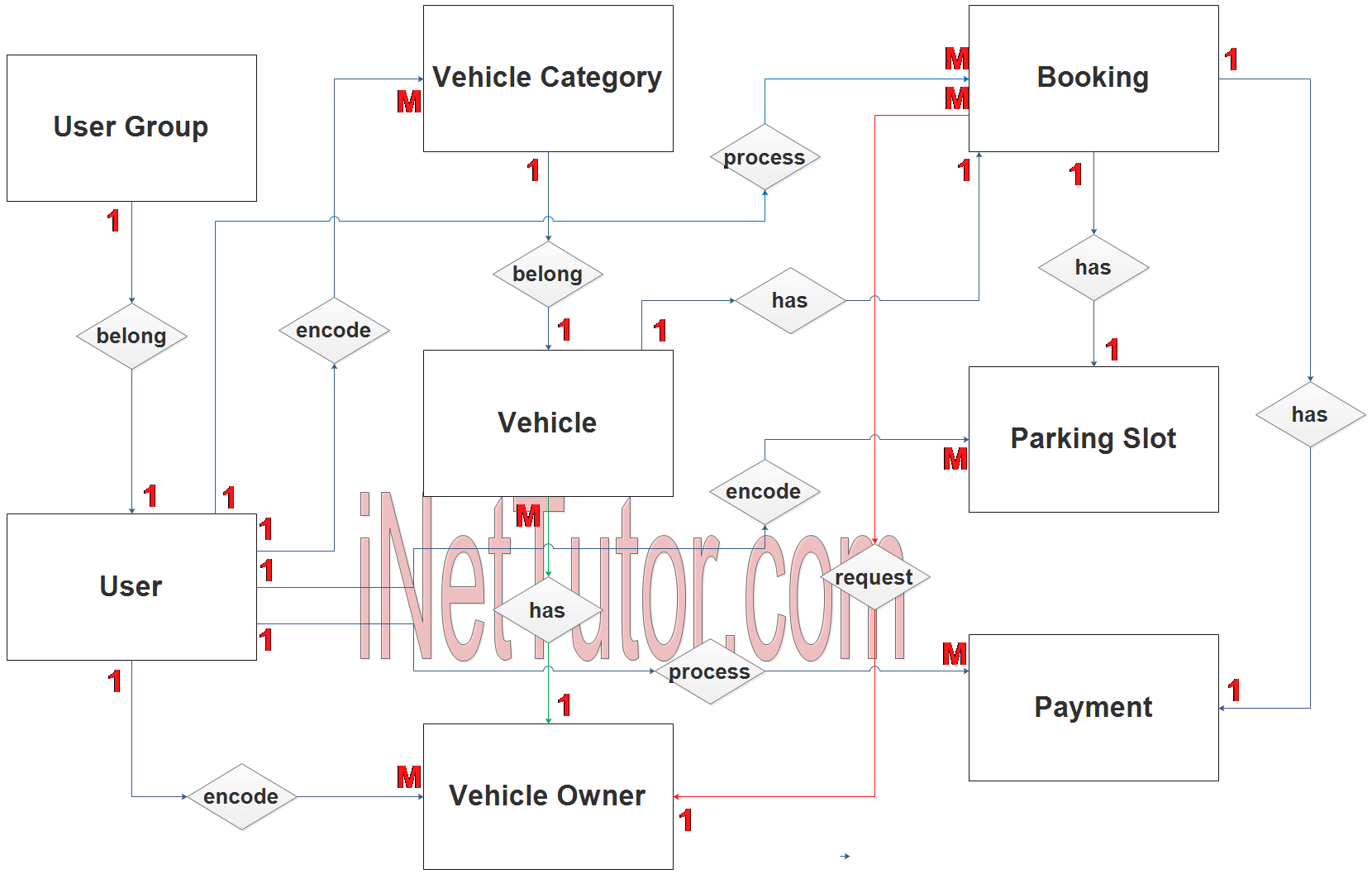
**Data Flow Diagram & User Stories**

|  |  |
| --- | --- |
| **Date** | 24 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Data Flow Diagram:**

****

**Example:**

**flow**

**User Stories:**

**Customer User Stories**

* *As a customer*, I want to book a service appointment online, so that I can avoid waiting at the garage.
* *As a customer*, I want to view my vehicle’s service history, so that I can track past repairs and maintenance.
* *As a customer*, I want to receive notifications when my vehicle is ready, so I can pick it up on time.

**Mechanic/Technician User Stories**

* *As a mechanic*, I want to see a list of assigned jobs, so I can plan my workday efficiently.
* *As a mechanic*, I want to update job status (e.g., In Progress, Completed), so the system reflects real-time progress.
* *As a mechanic*, I want to log parts used, so inventory is automatically updated.

**Admin/Manager User Stories**

* *As an admin*, I want to **add or remove services**, so the system reflects current offerings.
* *As an admin*, I want to **generate invoices**, so I can bill customers accurately.
* *As an admin*, I want to **track inventory levels**, so I can reorder parts before they run out.
* *As an admin*, I want to **view reports on service trends**, so I can make informed business decisions.

**System/User Interface Stories**

* *As a user*, I want to **log in securely**, so my data is protected.
* *As a user*, I want to **navigate the dashboard easily**, so I can find what I need without confusion.

**Project Design Phase -||**

**Solution Requirements(Functional & Non – functional)**

|  |  |
| --- | --- |
| **Date** | 24 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Functional Requirements:**

**1. Customer Management**

* Add, update, and delete customer profiles
* Track customer service history and preferences
* Send appointment reminders or service updates

**2. Vehicle Management**

* Register and manage vehicle details (make, model, VIN, etc.)
* Link vehicles to customer profiles
* Maintain service and repair history

**3. Appointment Scheduling**

* Allow customers or staff to book service appointments
* Display available time slots and technician availability
* Notify technicians and customers of upcoming appointments

**4. Service Management**

* Create and manage service orders and repair jobs
* Track job status and technician assignments
* Define service types and associated costs

**5. Inventory Management**

* Track spare parts and tools
* Manage stock levels and reordering alerts
* Associate parts with specific jobs

**6. Billing and Invoicing**

* Generate service estimates and final bills
* Apply discounts, taxes, and payment methods
* Maintain transaction history and receipts

**7. User Access Management**

* Role-based access for staff (mechanics, managers, etc.)
* Login and authentication for system use
* **Audit logs for activities**

**8. Reports and Analytics**

* Generate reports on daily jobs, revenue, and inventory
* Analyze trends in services and customer visits
* Export data for further analysis

**Non functional Requirements:**

**1. Performance**

* The system should respond to user actions within 2 seconds.
* It should handle multiple users (e.g. up to 100 concurrent logins) without performance degradation.

**2. Scalability**

* Capable of expanding to support additional garages, vehicles, or users as the business grows**.**

**3. Reliability**

* The system must be available 99.9% of the time with minimal downtime.
* It should recover automatically from minor faults without data loss**.**

**4. Usability**

* The interface should be intuitive, with minimal training required for new users.
* Consistent design and terminology throughout the system**.**

**5. Security**

* Role-based access control to restrict sensitive data.
* Data encryption during transmission and storage.
* Regular password updates and login authentication**.**

**6. Maintainability**

* Modular design to allow easy updates or bug fixes.
* Clear documentation for both users and developers**.**

**7. Portability**

* The system should be accessible across desktops, tablets, or mobile devices.
* Cross-platform compatibility (Windows, macOS, etc.).

**8. Compliance**

* Adheres to legal standards for data privacy and record keeping, such as GDPR or local regulations**.**

**Project Design Phase -||**

**Technology Stack(Architecture & Stack)**

|  |  |
| --- | --- |
| **Date** | 24 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Technical Architecture:**

**1. Presentation Layer (Frontend)**

* **Purpose:** Interface where users interact with the system (e.g. booking appointments, tracking jobs**).**
* **Technologies:** HTML, CSS, JavaScript, frameworks like React or Angular
* Devices Supported: Desktop browsers, tablets, possibly mobile apps

**2. Application Layer (Backend)**

* **Purpose:** Core business logic – handles job assignments, billing, customer profiles, etc.
* **Technologies:** Node.js, Python (Django/Flask), Java (Spring Boot), or .NET

** Responsibilities:**

* Processing user requests
* Enforcing business rules
* Managing notifications and workflows

**3. Data Layer (Database)**

* **Purpose:** Store and retrieve structured data for vehicles, customers, parts, services, and transactions

**Technologies:**

* Relational DBMS: PostgreSQL, MySQL, SQL Server
* NoSQL (optional for logs or analytics): MongoDB

**4. Integration Layer**

* **Purpose:** Enables communication with third-party tools or services
* **Examples:**
* SMS/email gateways for reminders
* Payment gateways
* Inventory suppliers
* APIs for vehicle diagnostics (if advanced)

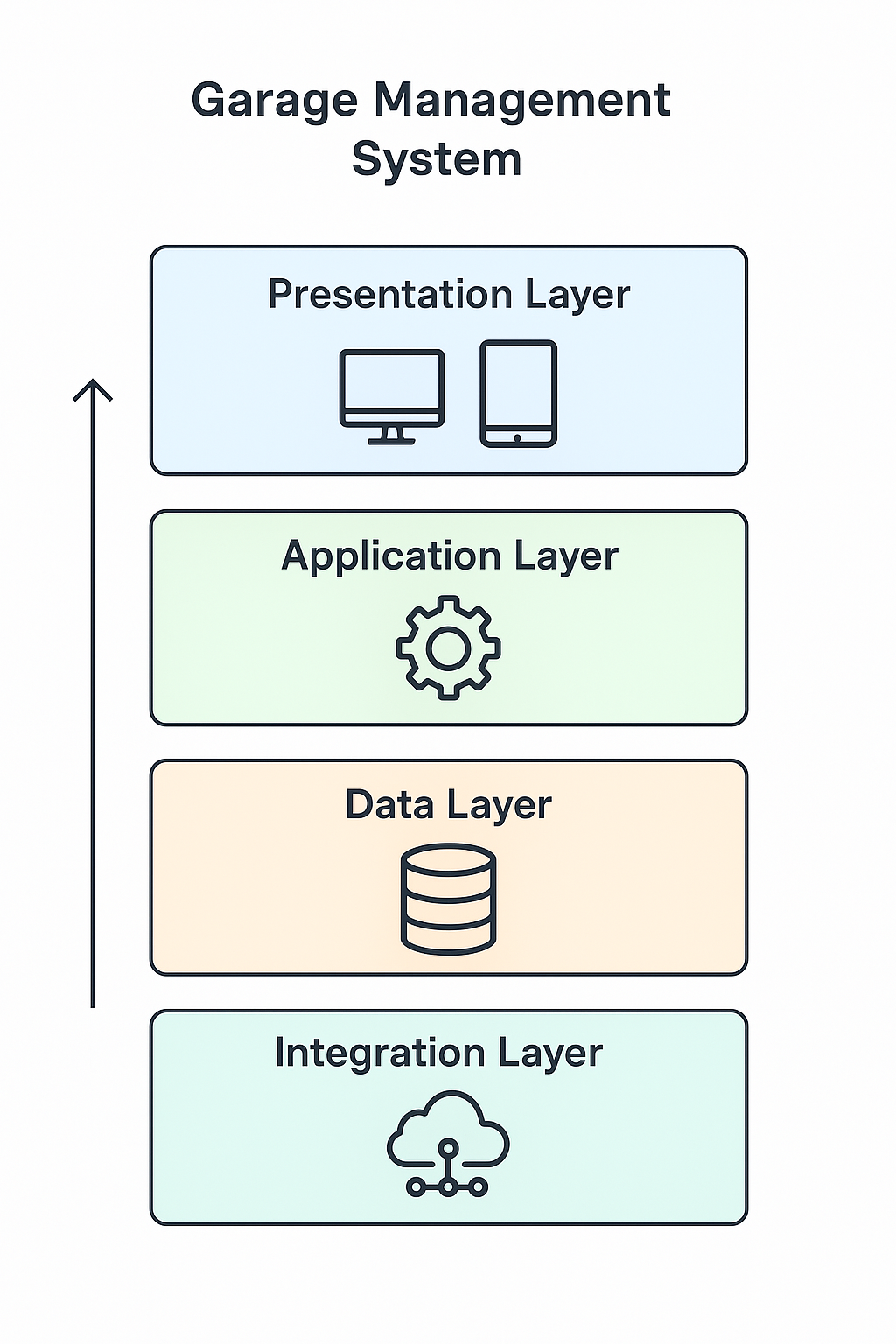
**5. Security Layer (Cross-cutting concern)**

* Role-based access control (RBAC)
* HTTPS/SSL encryption
* Authentication via OAuth2 or JWT

**6. Deployment & Infrastructure**

* **Deployment Options:** Cloud-hosted (e.g. AWS, Azure, GCP) or on-premise
* **Containerization:** Docker, Kubernetes for scalability
* **CI/CD:** Jenkins, GitHub Actions for automated builds and testing
* **Backup & Recovery Tools**

**Diagram:**

****

**Stack:**

**Frontend (Client-Side)**

**Where users interact with the system**

* Languages: HTML5, CSS3, JavaScript
* Frameworks: React.js or Angular
* Styling: Tailwind CSS or Bootstrap
* State Management: Redux (if using React**)**

**Backend (Server-Side Logic)**

**Where all business logic is handled**

* Languages: Node.js (JavaScript), Python (Django/Flask), or Java (Spring Boot)
* Frameworks/Platforms: Express (Node.js), Django (Python)
* API Architecture: REST or GraphQL
* Authentication: JWT (JSON Web Tokens) or OAuth2

**Database Layer**

Stores structured and unstructured data

* **Relational DB:** PostgreSQL or MySQL
* **NoSQL DB (optional):** MongoDB (for logs, analytics, flexible data models)
* **Caching:** Redis (for speeding up access to frequently used data)

**Infrastructure & Deployment**

Where and how the app runs

* **Containerization:** Docker
* **Orchestration:** Kubernetes (for scalability and fault tolerance)
* **Cloud Providers:** AWS (EC2, RDS, S3), Azure, or Google Cloud
* **CI/CD Pipelines:** GitHub Actions, GitLab CI, or Jenkins

**Security & Monitoring**

Protects data and maintains system health

* **Security Protocols:** HTTPS, TLS, CSRF/XSS protection libraries
* **Monitoring Tools:** Prometheus, Grafana, or Datadog
* **Logging:** ELK Stack (Elasticsearch, Logstash, Kibana) or Loki

**Optional Enhancements**

Adds value, especially in customer experience

* **Mobile App:** React Native or Flutter for cross-platform apps
* **Notification Services:** Firebase Cloud Messaging, Twilio (for SMS), or SendGrid (for emails)

**Project Design Phase**

**Problem Solution Fit**

|  |  |
| --- | --- |
| **Date** | 25 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Problem – Solution Fit :**

**Problem Statement:**

* Manual booking and customer record tracking.
* Inefficient inventory management for spare parts and tools.
* Lack of real-time job tracking and service updates.
* Poor communication between mechanics, management, and customers.
* Paper-based billing and service history records.

**Solution Overview**:

* Provide **digital booking and appointment scheduling**.
* Automate **inventory tracking**, alerting when parts are low in stock.
* Offer a **real-time job progress dashboard** for internal use and customer updates.
* Enable **digital invoicing** and maintain a full service history.
* Integrate **customer communication tools** (SMS, email, or app notifications).

**Problem-Solution Fit**

When the system addresses the exact pain points of garage operators, it creates a seamless operational flow. Your project design should clearly map:

* **Each core problem → The feature that solves it**
* **Real-world user need → Usable, intuitive system function**

**Operational Pain Points**

* Manual scheduling leads to double bookings or missed appointments.
* Disorganized work orders cause delays in vehicle servicing.
* Limited visibility into job status for both managers and customers**.**

**Administrative Challenges**

* Inventory mismanagement, especially of fast-moving spare parts.
* Billing errors and lack of streamlined digital invoicing.
* Inconsistent record-keeping for customer history and vehicle data.

**Customer Experience Gaps**

* Poor communication about service progress or delays.
* No real-time updates or easy digital channels for customers to engage.
* Low customer retention due to impersonal or inefficient service.

**Staff Management Struggles**

* Workforce scheduling issues, especially with last-minute absenteeism.
* Lack of performance tracking for technicians or job efficiency.
* Training gaps leading to inconsistent service quality.

**Key Points:**

1. **User Persona Mapping**

* Small independent mechanics
* Multi-bay garage owners
* Fleet maintenance teams

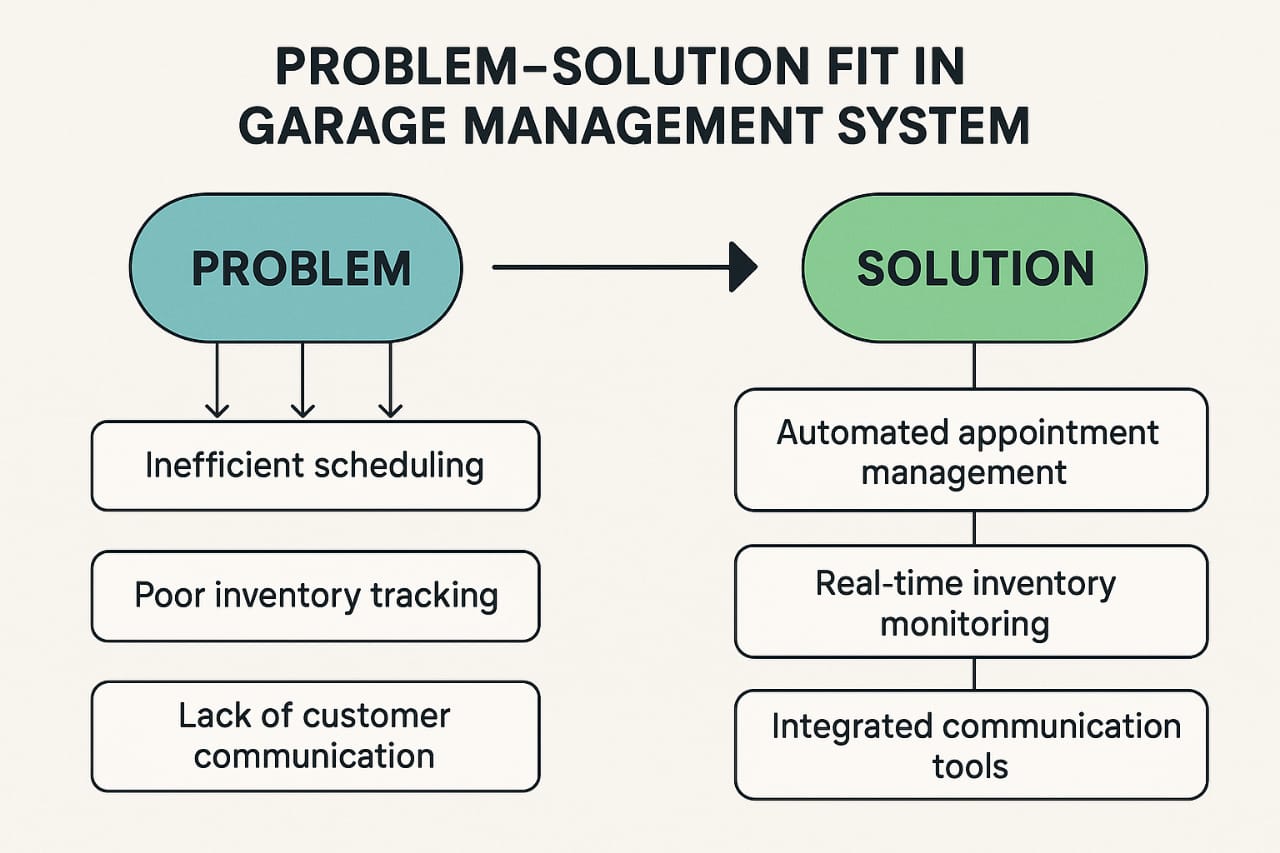
2. **Rapid Feedback Loops**

Once you've built a basic prototype or clickable UI:

* Share it with a few local garages
* Observe how they use it in real time
* Ask: “What do you wish this system could do?”

Their reactions will show you what resonates and what falls flat. This is your fastest path to alignment with real-world need

Diagram:



**Project Design Phase**

**Proposed Solution**

|  |  |
| --- | --- |
| **Date** | 25 January 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Proposed Solution:**

**Proposed Solution for a Garage Management System**

**1. Core Modules:**

**1.Core Modules:**

* Customer Management: Store customer profiles, vehicle history, and contact details.
* Vehicle Intake & Service Scheduling: Digital intake forms, service appointment calendar, and reminders.
* Job Card Management: Auto-generate job cards with repair details, parts used, labor costs, and technician assignments.
* Inventory Management: Track parts and supplies, manage stock levels, reorder alerts.
* Billing & Invoicing: Integrated GST-compliant invoicing, discounts, and multiple payment modes.
* Technician Management: Track technician workload, shift scheduling, and performance analytics.
* Reports & Analytics: Daily income reports, parts usage, service trends, and customer feedback analysis.

**2. Technology Stack (example):**

* Frontend: React.js or Angular
* Backend: Node.js with Express or Django
* Database: PostgreSQL or MongoDB
* Hosting: Cloud-based (AWS, Azure, or Firebase)

**3. Key Features:**

* Mobile-responsive interface
* SMS/Email notifications
* Data backup and recovery
* Role-based access (admin, technician, receptionist)
* Barcode scanning for inventory

**4. Optional Add-ons:**

* Customer app for booking & tracking
* Loyalty points system
* Integration with insurance claim APIs
* Vehicle diagnostics integration (OBD data)

**Project Design Phase**

**Solution Architecure**

|  |  |
| --- | --- |
| **Date** | 25 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Solution Architecture:**

**1.Presentation Layer (Frontend)**

**Handles user interactions.**

* Platforms: Web, Mobile App
* Technologies: React.js / Angular (Web), React Native / Flutter (Mobile)
* Features: UI for booking, service tracking, inventory updates, invoicing

**2. Application Layer (Backend Services)**

**Processes business logic and data flow.**

* API Gateway: Handles routing and security (e.g., API rate limiting, authentication)
* Business Services:
* Customer & Vehicle Management
* Service Request Handling
* Job Card Processing
* Billing & Payments
* Inventory Control

3. **Data Layer**

Stores and retrieves all necessary data.

* **Databases:**
  + PostgreSQL / MySQL (structured data)
  + MongoDB (flexible schemas for logs or service history)
* **File Storage:** AWS S3 or Azure Blob for invoice PDFs, vehicle images, etc.
* **aching:** Redis or Memcached for frequently accessed data

4. **Integration Layer**

Ensures communication with external systems.

* **Payment Gateway:** Razorpay, Stripe, etc.
* **SMS/Email Service:** Twilio, SendGrid
* **Insurance APIs** for claims management (optional)
* **OBD integration** for real-time diagnostics (optional)

5. **Security & Identity Management**

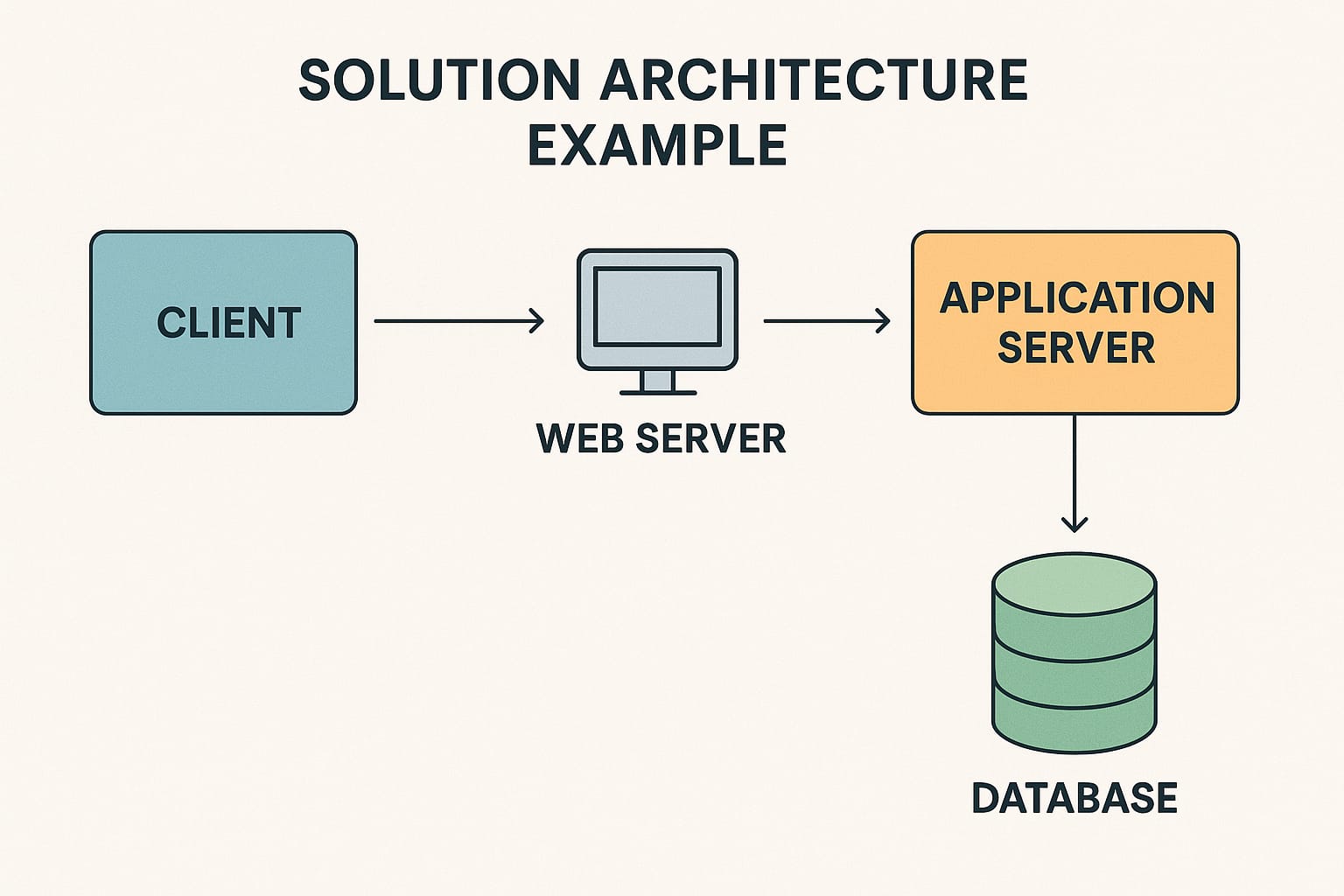
Protects user data and manages access.

* Role-Based Access Control (RBAC)
* OAuth2/JWT for secure login
* Encrypted communication via HTTPS

6. **Deployment & DevOps**

* **Containerization:** Docker
* **Orchestration:** Kubernetes
* **CI/CD Pipelines:** GitHub Actions, Jenkins
* **Monitoring:** Prometheus + Grafana, ELK Stack

**Solution Architecture Diagram** **Example:**

****

**Project Planning Phase**

**Project Planning(Product,Backlog,SpritPlanning,Stories,Story point)**

|  |  |
| --- | --- |
| **Date** | 25 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Project Planning:**

**Product Backlog for Garage Management System**

**Customer & Vehicle Management**

* [High] As a receptionist, I want to add/edit customer details so that I can manage service records.
* [High] As a mechanic, I want to view a vehicle’s service history to diagnose issues quickly.
* [Medium] As a customer, I want to receive reminders for my next service date.

**Appointment & Job Scheduling**

* [High] As a manager, I want to assign jobs to mechanics based on availability.
* [Medium] As a customer, I want to book service slots online**.**

**Inventory Management**

* [High] As a storekeeper, I want to track spare part quantities to avoid stockouts.
* [Medium] As a technician, I want to request spare parts during a job.
* [Low] As a manager, I want low-stock alerts via email or SMS**.**

**Billing & Payments**

* [High] As a cashier, I want to generate invoices automatically after service completion.
* [Medium] As a customer, I want to pay online via UPI or card.
* [Low] As an accountant, I want to export monthly billing reports**.**

**Reporting & Analytics**

* [Medium] As a manager, I want a dashboard that summarizes daily operations.
* [Medium] As an owner, I want to track service trends by vehicle type or season.
* [Low] As a mechanic, I want to view performance stats (jobs completed, ratings).

**Notifications & Communication**

* [Medium] As a customer, I want to receive real-time updates when my vehicle is ready.
* [Low] As a manager, I want to send promotional messages to customers.

**Sprint planning:**

**Sprint 0 – Setup & Planning**

* Finalize product requirements and wireframes
* Define user personas and workflows
* Set up version control, CI/CD pipelines, and collaboration tools
* Assign team roles and prepare backlog

**Sprint 1 – Core Customer & Vehicle Management**

* Add/edit customer and vehicle profiles
* Basic UI for customer lookup and vehicle history
* Database schema setup

**Sprint 2 – Job Scheduling & Mechanic Allocation**

* Create job cards with service types
* Assign jobs to available mechanics
* View schedule dashboard

*Goal:* Garage manager can allocate work and view service pipeline

**Sprint 3 – Inventory & Billing**

* Parts catalog with stock levels
* Use parts in job cards
* Generate basic invoices
* *Goal:* Mechanic can use parts, and cashier can bill the customer

**Sprint 4 – Internal QA & Feedback**

* End-to-end flow testing
* UI polish and bug fixes
* Collect feedback from team or pilot garages

*Goal:* MVP workflow is smooth and usable

**Sprint 5 – Beta Launch & Customer Feedback**

* Launch to 1–2 real garages
* Add feedback loop (e.g., feedback form or survey)
* Tweak features based on real usage

**Stories:**

**Receptionist / Admin**

* *As a receptionist*, I want to register a new customer and their vehicle quickly so I can keep service queues moving.
* *As a receptionist*, I want to schedule appointments and assign them to available mechanics.
* *As an admin*, I want to configure garage hours and holiday calendars to avoid scheduling conflicts.

**Inventory Manager**

* *As an inventory manager*, I want to track incoming and outgoing spare parts to maintain accurate stock records.
* *As an inventory manager*, I want to get low-stock alerts so I can restock parts before they run out.
* *As an inventory manager*, I want to generate monthly inventory reports for procurement planning**.**

**Cashier / Accountant**

* *As a cashier*, I want to generate an invoice once a service is completed so the customer can pay and leave promptly.
* *As a cashier*, I want to accept payments in cash, card, or digital mode.
* *As an accountant*, I want to export daily and monthly billing summaries for auditing purposes.

**Garage Owner / Manager**

* *As a manager*, I want to view daily job status and staff workload to ensure smooth operations.
* *As a manager*, I want access to key metrics like revenue, repeat customers, and high-demand services.
* *As a manager*, I want to send promotional offers to customers to boost repeat business**.**

**Customer (via Web or App)**

* *As a customer*, I want to book a service appointment online so I can avoid calling or waiting in line.
* *As a customer*, I want to get SMS/WhatsApp updates when my vehicle is ready.
* *As a customer*, I want access to my vehicle's service history in case I need it for resale or insurance claims**.**

**Story points:**

**Customer & Vehicle Management:**

|  |  |  |
| --- | --- | --- |
| **story** | **points** | **why** |
| Add/Edit customer profiles | 3 | Simple forms with validations |
| Link vehicle to customer | 3 | Requires relational mapping |

**Job Scheduling& Assignment**

|  |  |  |
| --- | --- | --- |
| **story** | **points** | **why** |
| Create service job card | **3** | Captures issue, service type, mechanic |
| Assign jobs to mechanic | **5** | involves calendar integration, conflict checks |
| View daily job list | **2** | Moderate UI filtering logic |

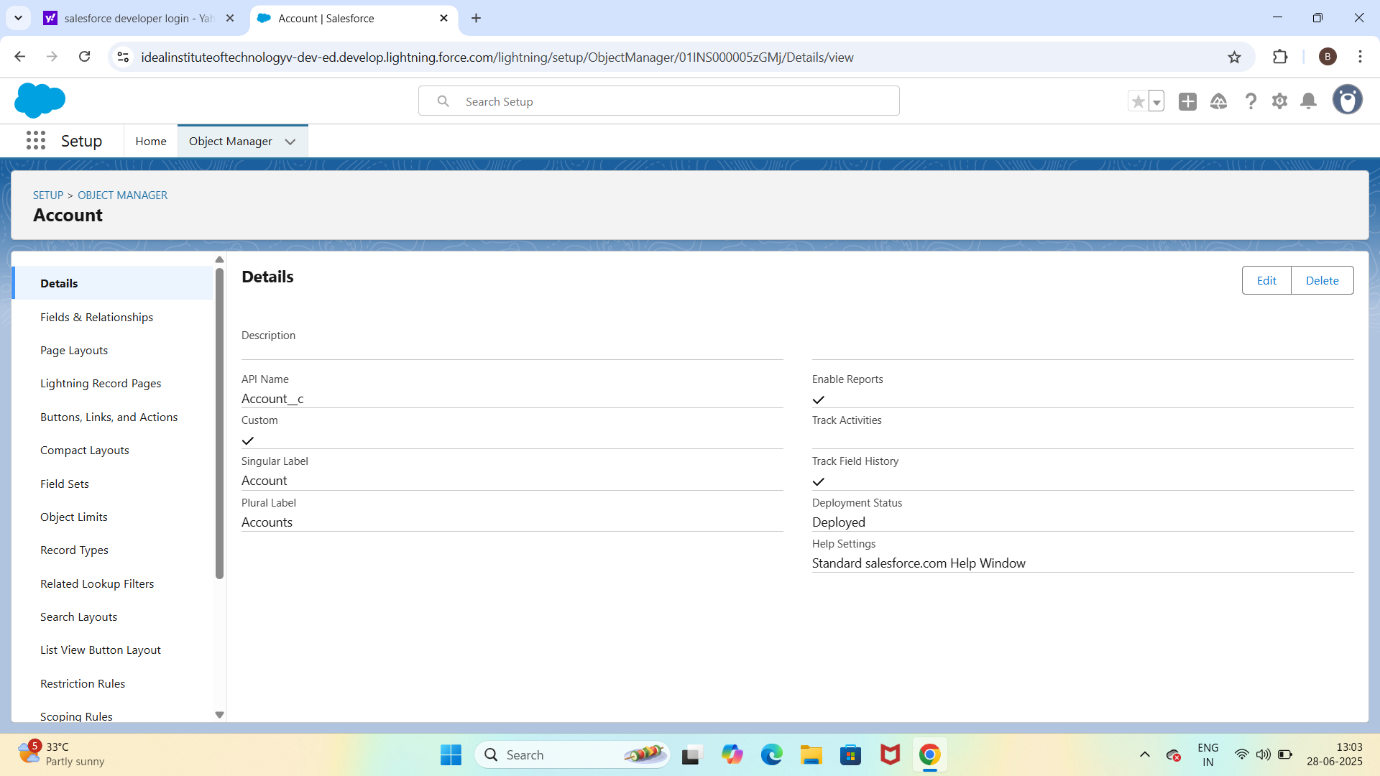
**Project Executable Files (Project Files)**

|  |  |
| --- | --- |
| **Date** | 25 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

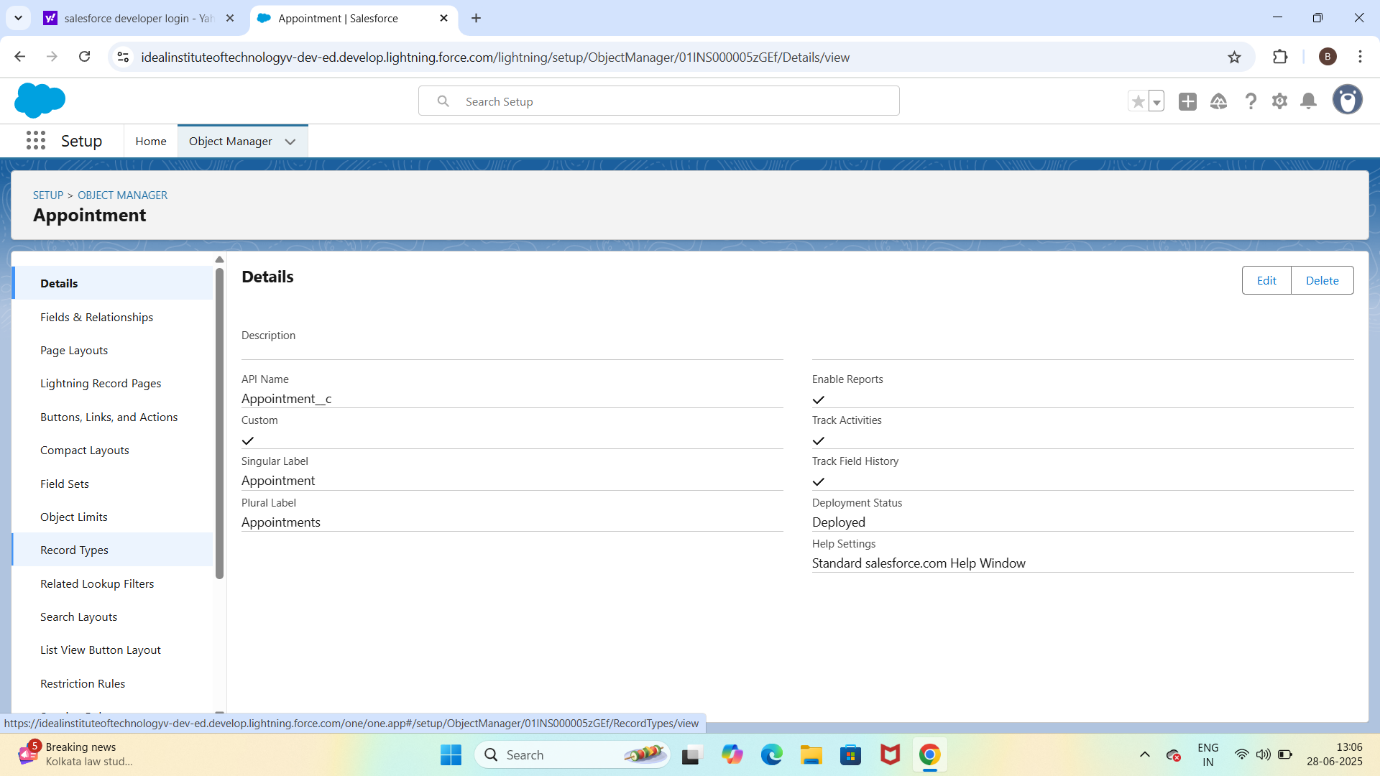
This is a **Salesforce-based implementation**, so the "executables" are not traditional .exe files but configurations and custom code within the Salesforce ecosystem. The project includes:

**A.Custom Salesforce Objects:**

Created via salesforce setup**:**

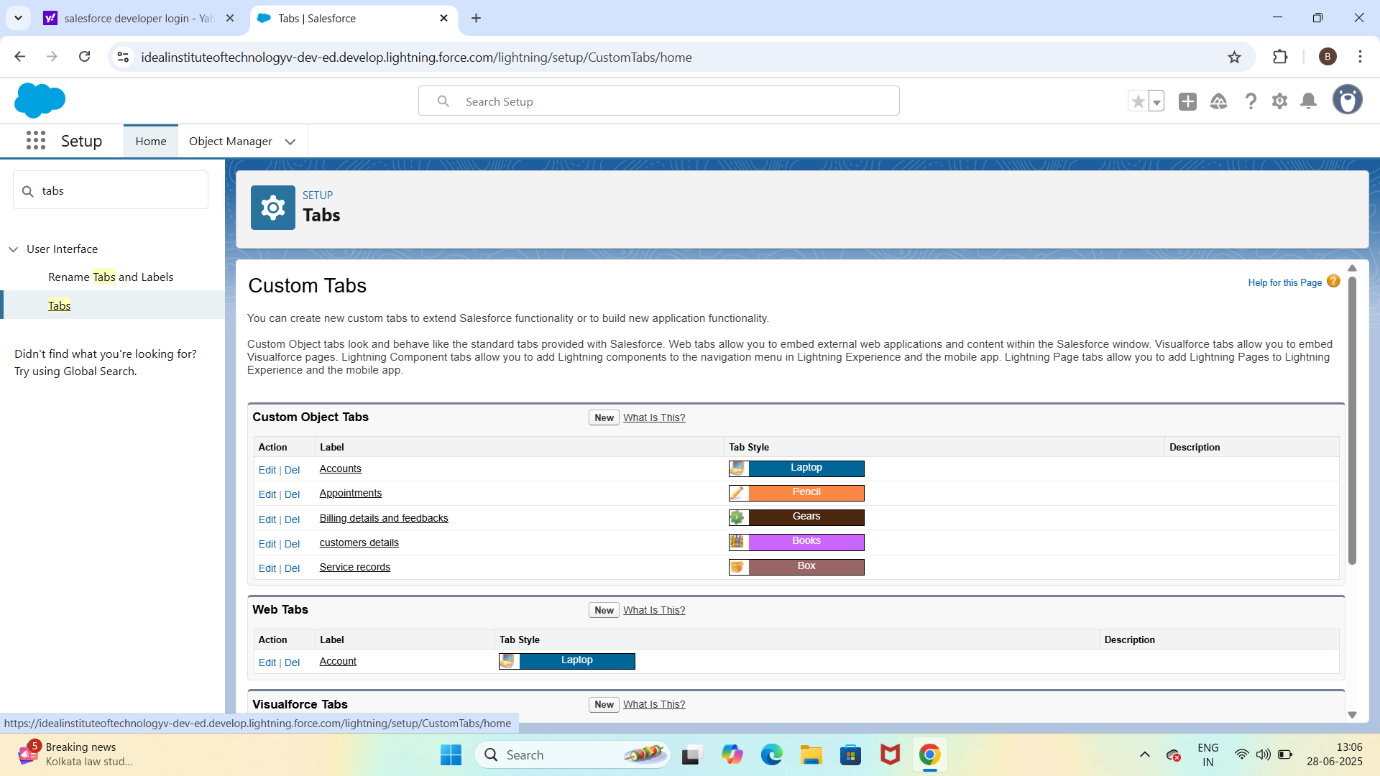
**Account:**

**Appointment:**

****

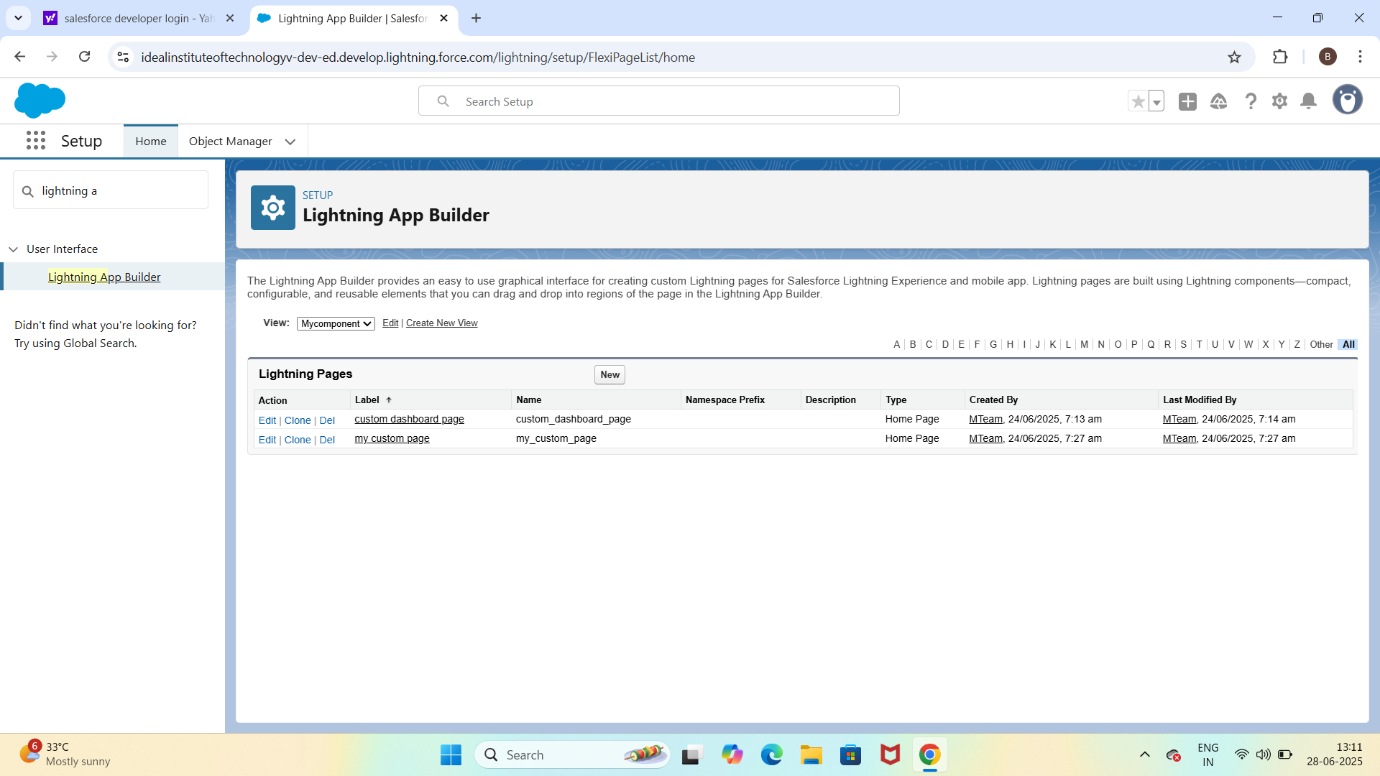
**B.Custom Tabs and APP:**

Tabs for each object

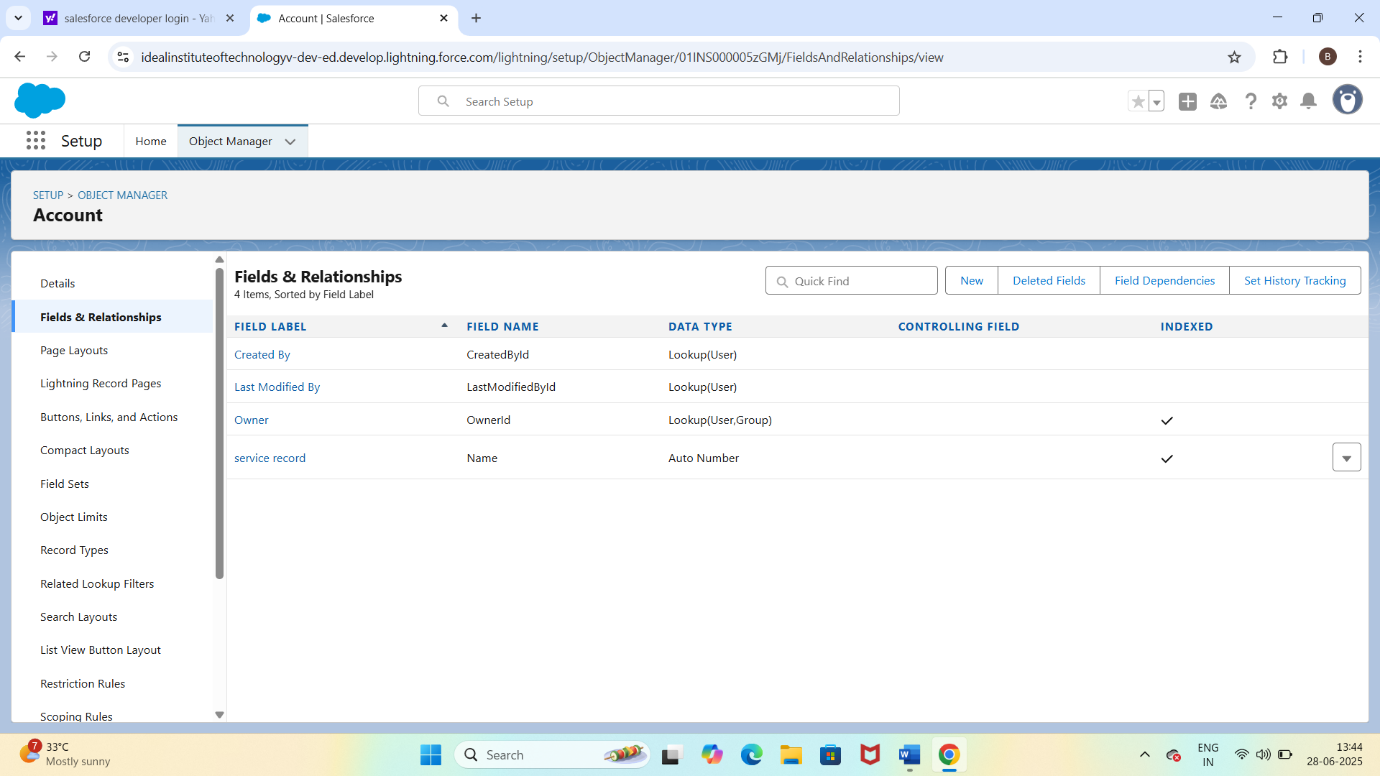
****

**Lightning app:**

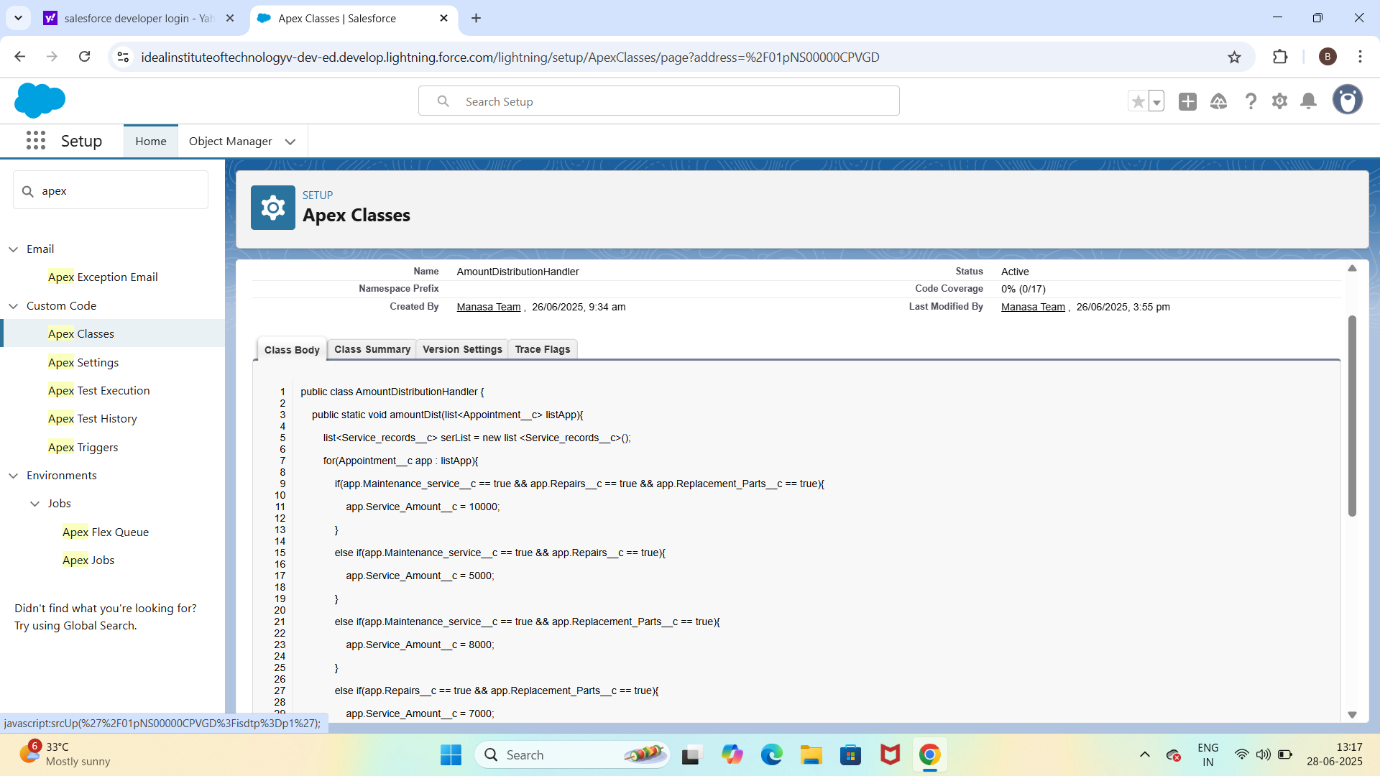
A lightning app called lease management with navigation setup



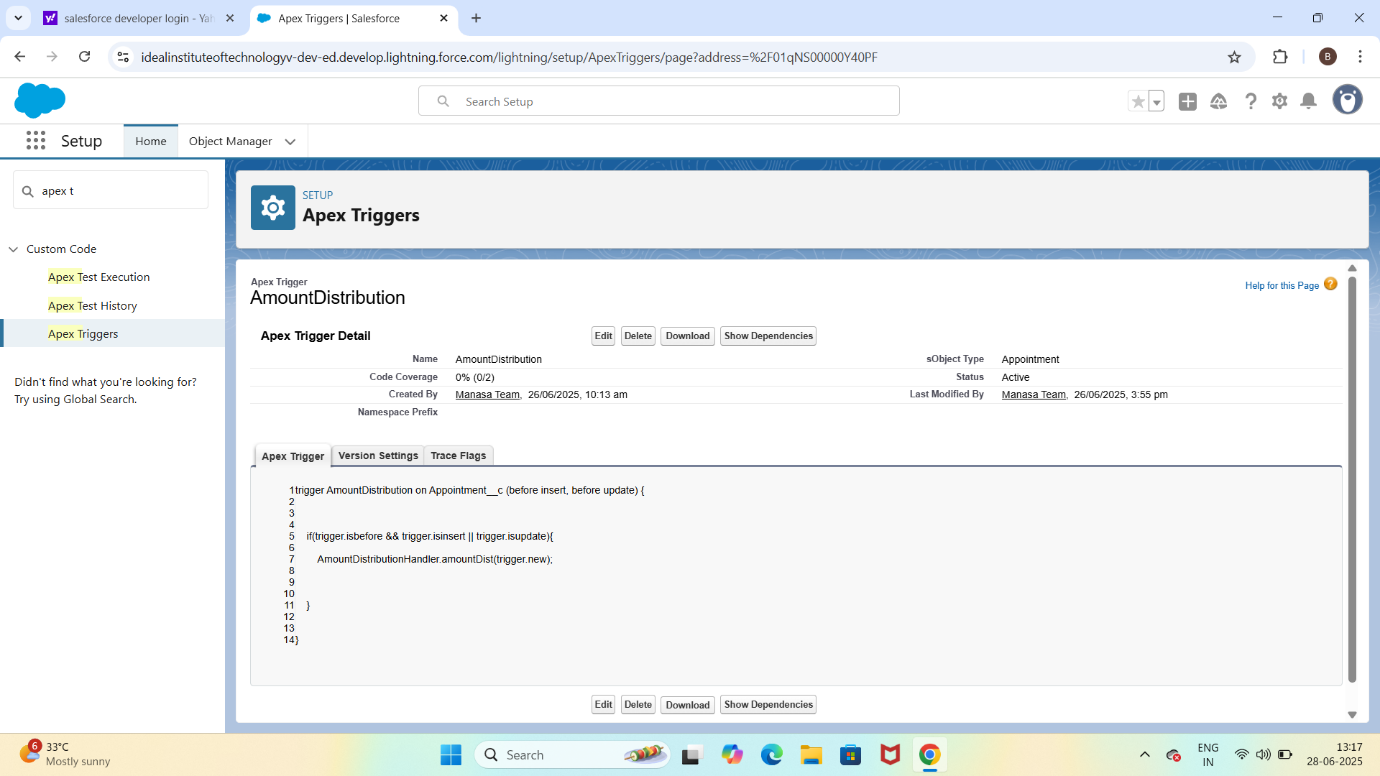
**C.Fields and Relationships**:



**D.Apex Code(Custom Backend logic):**

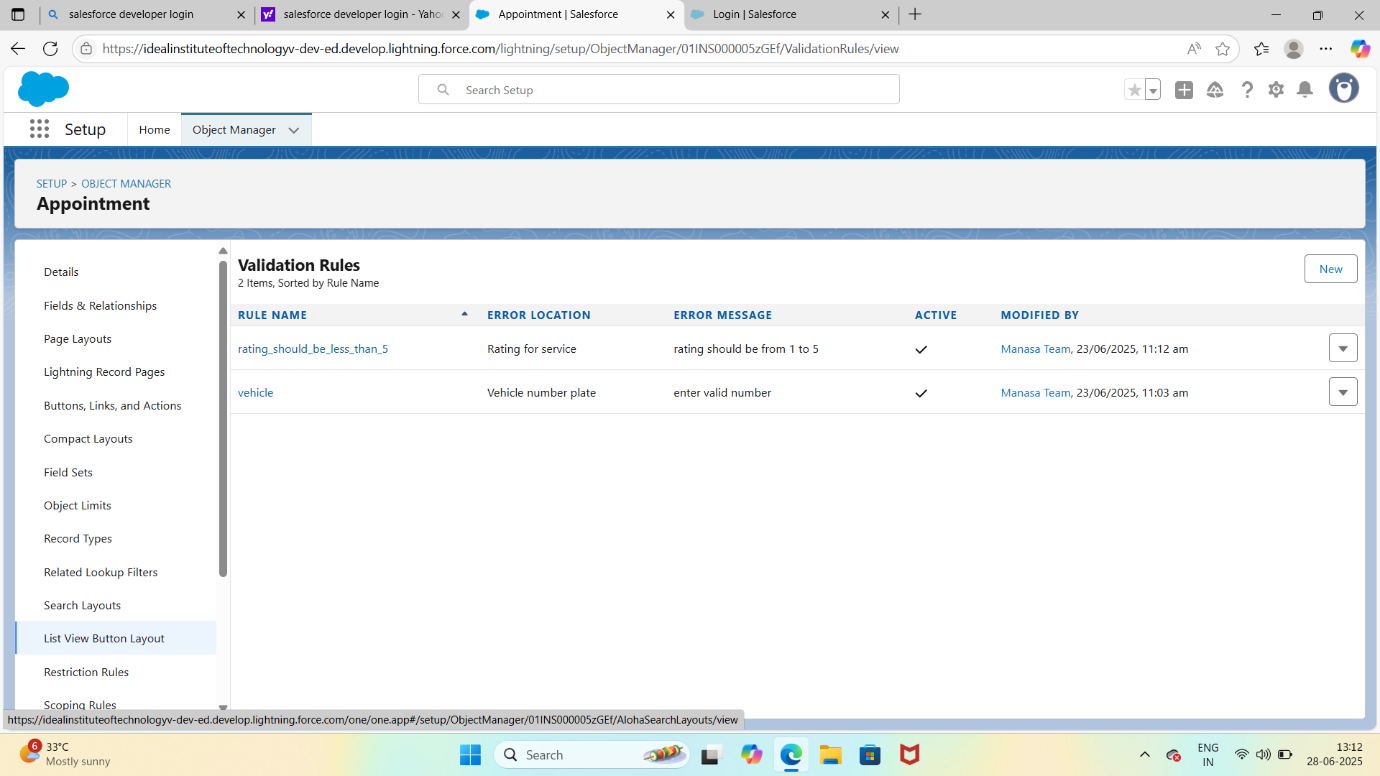
**Apex Class:**

**Apex Trigger:**

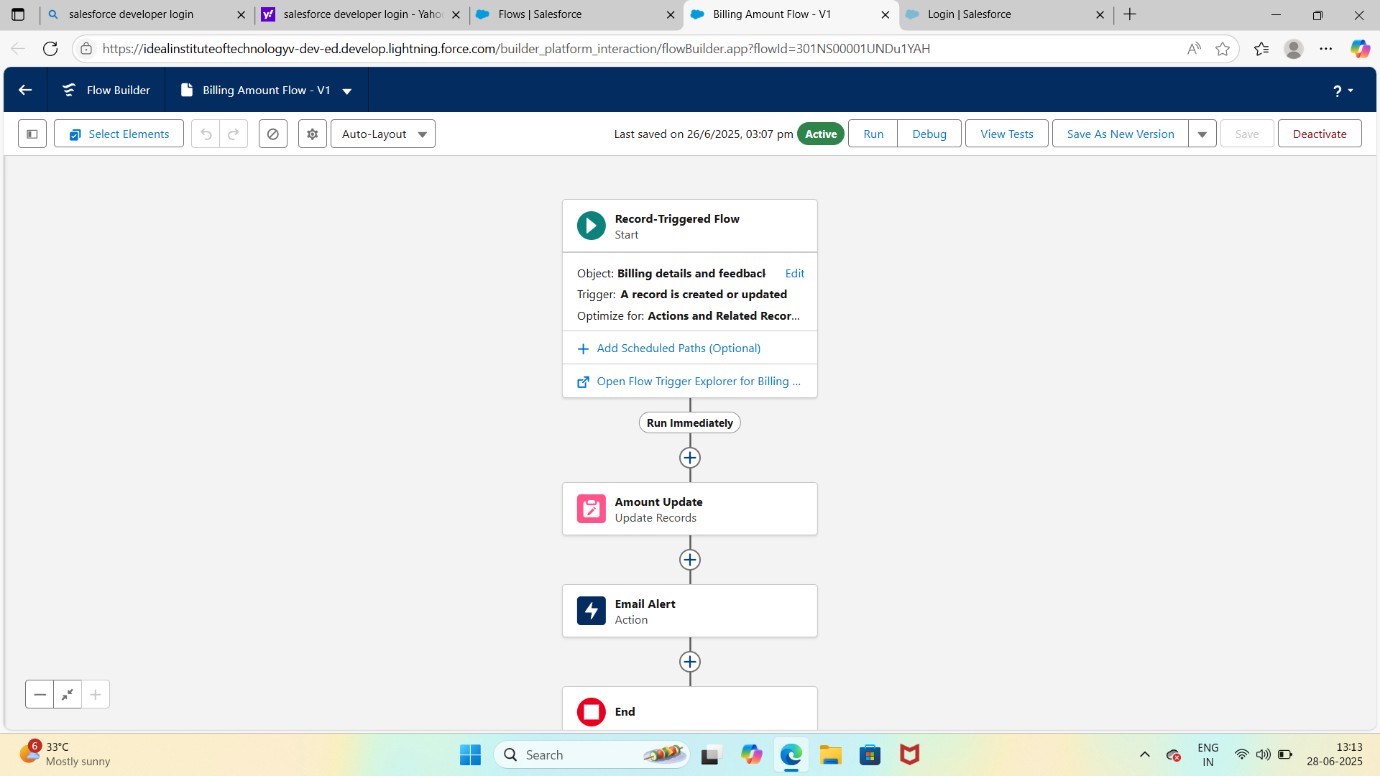
****

**E.Validation Rule:**

Appointment object has a rule: End\_date\_\_c > Start\_date\_\_c

****

**F.Flows:**

****

**Project Development Phase**

**Model Performance Test**

|  |  |
| --- | --- |
| **Date** | 25 June2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Model Performance Testing:**

To test model performance in a Garage Management System (GMS), you'd typically be evaluating how well your system handles tasks like appointment scheduling, inventory tracking, billing, and customer management—especially if you're using machine learning or rule-based automation. Here's how you can approachit**:**

**Key Areas of Performance Testing**

* **Response Time**: Measure how quickly the system responds to user actions (e.g., booking a service or generating an invoice).
* **Throughput**: Test how many transactions or service requests the system can handle per second/minute**.**
* **Scalability**: Simulate increased load (more users, more vehicles) to see how the system scales.
* **Resource Utilization:** Monitor CPU, memory, and database usage under different loads.
* **Error Rate:** Track how often the system fails or returns incorrect results under stress.

**You're Using AI/ML Models:**

If your GMS includes predictive models (e.g., for estimating service time or predicting part failures), you’ll want to test:

* Accuracy: How close are the predictions to actual outcomes?
* Precision/Recall: Especially important if you're flagging potential issues like part failures.
* Latency: How fast does the model return results?
* Model Drift: Is the model still accurate over time as new data comes in?

**Tools You Can Use**

* Apache JMeter or Locust for load and stress testing
* Postman for API performance testing
* TensorBoard or MLflow for tracking ML model metrics
* New Relic or Datadog for real-time system monitoring

**Functional & Performance Testing**

**Model Performance Test**

|  |  |
| --- | --- |
| **Date** | 25 June 2025 |
| **Team ID** | LTVIP2025TMID31613 |
| **Project Name** | Garage Management System |
| **Maximum Marks** |  |

**Test Scenarios &Results:**

**Functional Testing:**

This ensures each feature behaves as intended:

* User Authentication: Login/logout, role-based access (admin, mechanic, customer)
* Appointment Scheduling: Booking, rescheduling, and cancellation flows
* Inventory Management: Adding, updating, and tracking parts
* Billing & Invoicing: Generating accurate bills, applying discounts, tax calculations
* Service History Tracking: Recording and retrieving vehicle service records
* Notifications: Email/SMS alerts for service reminders or updates

**Performance Testing** :

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Scenario** | **What to Test** | **Performance Metric** |
| Booking surge | 100+ users book services simultaneously | Response time < 2s |
| Inventory update | Bulk upload of 500+ parts | CPU/memory usage, DB latency |
| Invoice generation | Generate 1000 invoices in batch | Throughput (invoices/min) |
| Search service history | Query 10 years of records | Query time < 1s |

**Tools:**

* **Selenium:** Automate functional test cases
* **JMeter or Locust:** Simulate concurrent users and load
* **Postman:** Test API endpoints for correctness and speed
* **Grafana + Prometheus:** Monitor system metrics in real time

**\*Advantages of Garage Management System:\***

**1. \*Improved Efficiency\*: Streamlines operations, reducing wait times and increasing productivity.**

**2. \*Enhanced Customer Experience\*: Provides transparent communication, convenient appointment scheduling, and timely updates.**

**3. \*Better Inventory Management\*: Tracks spare parts, reducing stockouts and overstocking.**

**4. \*Data-Driven Insights\*: Generates reports on sales, customer behavior, and operational performance.**

**5. \*Increased Revenue\*: Optimizes pricing, promotions, and services to maximize revenue.**

**6. \*Reduced Errors\*: Automates tasks, minimizing human errors and improving accuracy.**

**7. \*Improved Customer Retention\*: Fosters loyalty through personalized service and timely follow-ups.**

**\*Disadvantages of Garage Management System:\***

**1. \*Implementation Costs\*: Initial investment in software, hardware, and training.**

**2. \*Technical Issues\*: Potential system downtime, glitches, or integration challenges.**

**3. \*Data Security Risks\*: Vulnerability to cyber threats and data breaches.**

**4. \*Dependence on Technology\*: Over-reliance on the system can lead to operational disruptions if it fails.**

**5. \*Training Requirements\*: Staff may require training to effectively use the system.**

**6. \*Customization Limitations\*: Some systems may not fully adapt to specific garage needs.**

**7. \*Maintenance and Upgrades\*: Ongoing costs and efforts to maintain and update the system.**

**By understanding the advantages and disadvantages, garage owners can make informed decisions about implementing a management system that suits their needs.**

**Conclusion:**

**The Garage Management System in Salesforce offers a comprehensive solution for garage owners to streamline operations, enhance customer experience, and drive business growth. By leveraging Salesforce's capabilities, garages can:**

**1. \*Automate workflows\*: Simplify appointment scheduling, job card management, and inventory tracking.**

**2. \*Improve customer engagement\*: Provide personalized service, timely updates, and convenient communication channels.**

**3. \*Gain data-driven insights\*: Analyze sales, customer behavior, and operational performance to inform business decisions.**

**4. Optimizes garage operations: Automates workflows, manages inventory, and tracks customer interactions.**

**5. Improves customer satisfaction: Provides personalized service, timely updates, and convenient communication channels.**

**6. Drives business growth: Offers data-driven insights, enabling informed decisions and strategic planning.**

**Github link:** [**mohammedalisha**](https://github.com/mohammedalisha)**/**[**Garage-Management-System**](https://github.com/mohammedalisha/Garage-Management-System)

**DEMO:**

<https://screenapp.io/app/#/shared/ZTCV-1h-8G>