**Project Overview**

**Executive Summary**

All State Independent Auto Inspection is an automotive service management platform designed to bridge the communication gap between dealers and technicians. Instead of relying on scattered messages and manual coordination, it brings everything into a single, organized space — from real-time chat to service postings and inspection tracking.

The goal is simple: speed up the entire service process while keeping both dealers and technicians on the same page. With features like instant messaging, structured inspection records (including photo attachments), and streamlined job posting, this reduces the time from service request to completion and improves overall customer satisfaction.

**Business Value:**  
By replacing slow, fragmented communication with an integrated workflow, All State Independent Auto Inspection helps:

* Cut service completion time by up to 40%.
* Improve transparency between dealers and technicians.
* Increase customer trust through documented inspections.

**Key Features**

* **Real-Time Chat** – Dealers and technicians can exchange messages instantly, avoiding delays caused by missed calls or emails.
* **Service Posting Management** – Dealers can post, update, and track service jobs while technicians can accept and update their progress.
* **Inspection Records with Photos** – Every inspection can be documented with detailed notes and visual evidence.
* **Role-Based Access Control** – Ensures that users only see and do what their role allows.
* **Mobile-Friendly Interface** – Designed to work smoothly on desktops, tablets, and smartphones.

**Target Users**

* **Dealers** – Create service requests, track progress, and receive inspection results.
* **Technicians** – View assigned tasks, communicate in real time, and submit inspection reports.
* **Administrators** – Manage users, monitor system performance, and enforce compliance.

**Technology Stack**

**Frontend:** ReactJS with modern UI libraries for a responsive, intuitive experience.  
**Backend:** Java (Spring Boot) microservices architecture for scalability and maintainability.  
**Database:** PostgreSQL (migrated from MySQL) for structured, reliable data storage.  
**Communication:** WebSockets for live updates and real-time messaging.  
**Authentication:** JWT-based token system with role-based permissions.  
**Deployment:** Docker containers orchestrated via Kubernetes.

**Project Goals**

* **Efficiency:** Streamline workflows so jobs are completed faster without sacrificing quality.
* **Clarity:** Ensure everyone knows the current status of each job at all times.
* **Accountability:** Maintain a clear history of every action taken during the service process.
* **Scalability:** Build a system that can handle growth — more users, more jobs, more data — without slowing down.