Automate: Service Meets Simplicity

Rakshith P, Assistant Professor, Dept. of CSE, SJCE-JSS STU, Mysuru.

Hemanth D L, Dept. of CSE, SJCE-JSS STU, Mysuru.
M Suhas, Dept. of CSE, SJCE-JSS STU, Mysuru.
Ganesh S Pandith, Dept. of CSE, SJCE-JSS STU, Mysuru.
Janapriya Gandhi M, Dept. of CSE, SJCE-JSS STU, Mysuru.

---***---

Abstract - In the quick and technologically changing world of today, the motor vehicle maintenance and repair sector is dependent greatly on traditional manual practices, which produce inefficiencies, transparency, and customer dissatisfaction. Small and local garages, especially, struggle to innovate towards contemporary customer demands like online reservation, price quotation, and payment through digital means. This discrepancy points towards a single digital solution to digitize service provision while taking care of both customers and mechanics.

Automate, our suggested intelligent, web-based system, offers vehicle service management with streamlined roles for customers, mechanics, and administrators. The system features a garage suggestion tool based on geolocation information and user feedback. Automate also has a sentiment analysis system to make respectful communication to and from the users and ensures safe online payments with Razorpay integration. Through the provision of digital access to booking of services, tracking, payment, and feedback, Automate ensures users are empowered with an ease of experience and supports local garages to succeed in a competitive business environment.

***Key Words*:** Vehicle Maintenance, Online Booking, Geolocation, Web Platform, Garage Recommendation, and Sentiment Analysis.

1. INTRODUCTION

The automobile service and repair industry has been dependent for a long time on conventional, human-based methods of scheduling services, estimating costs, and customer communication. This usually results in inefficiencies, opaqueness, and inconvenience to car owners, particularly when dealing with local or small scale automobile repair shops. At a time when digital transformation is redefining sectors, the provision of intelligent systems to the automotive service industry. To fill this shortfall, we offer a platform aimed at optimizing the vehicle service process. Automate allows users to schedule services online and pay securely—all from one easy-to-use interface. The platform features individual modules for customers, mechanics, and administrators, each of which is customized to their respective needs and roles. Apart from the main features, Automate combines sophisticated technologies such as geolocation for finding garages near one in case of emergencies, and Gen AI-powered sentiment analysis to maintain a polite communication platform.

By accommodating both 2-wheeler and 4-wheeler services, and providing digital assistance to small and independent mechanics, Automate wants to close the technology gap in the industry. This research paper describes the inspiration behind the project, system architecture, important features, and the contribution of AI towards increasing functionality. Through this platform, we seek to enhance operational effectiveness, enhance customer satisfaction, and enable local providers with usable digital solutions.

2. LITERATURE SURVEY

In the automobile service industry, digital transformation is increasingly gaining ground with many different solutions created to support workshop management, payment integration, and online booking of services. Most of these solutions are aimed at larger dealerships and service chains and, as a result, leave smaller and local garages without the digital support they need. To better understand current trends, barriers and opportunities, we will look at studies and systems that currently exist in relation to online car servicing platforms, AI and geographic services.

A well-known example is Trinco Automobile in Sri Lanka; providing online vehicle booking and payment options. Users can book their services and pay for them through online banking or mobile app. The platform is very basic, provides little AI integration, and does not feature cost estimation, predictive maintenance, or real-time service tracking. The MAS Motors LLC system, another example, was developed for a Toyota dealership in Libya, and has a web-based management system to manage functions such as scheduling appointments to tracking stock levels. Although Mas Motors LLC is quite holistic for the internal operations of the dealership, it does not include any features for customers to have an interaction nor apply any AI integrations or insights. Geolocation is also an area of research, and some of the work down by Japara et al. (2023) show a quality and accurate way to estimate distance and location validation using their models such as Haversine and Vincenty. These methods are highly useful in modeling use cases such as ride-hailing, logistics, and service discovery, but also for determining the proximity of garages in the event of a vehicle emergency.

Although current systems serve singular aspects of digital servicing, there is a glaring lack of a joined-up solution that pulls AI, geolocation, user engagement, and payment systems together – especially regarding support of local garages. This gap identifies the perfect opportunity for a platform like Automate because it can more cohesively deliver those aspects and offer modern technology incursions for the small service provider that has traditionally been limited to the immediate market (brick and mortar business providers) and has been out of the sphere of influence of the digital ecosystem.

3. TOOLS AND TECHNOLOGIES

The creation of Automate employs state-of-the-art web technologies and AI built in to offer a seamless, intelligent, and scalable platform for vehicle maintenance and repair services. The front-end component uses HTML, Tailwind CSS, JavaScript, and React.js which have provided a modular and functional user experience across customers, mechanics, and administrators. Tailwind CSS will simplify styling overall and fasten the UI development. React.js provides the ability to render dynamic components when data updates are detected. The libraries, such as Axios and React Query, allow us to manage API calls and support us in an efficient asynchronous manner with all data requests from the client side.

The back-end uses Node.js and Express.js combined to compose API routes. Node.js handles business logic; Express.js manages API endpoints and AI functionality.

The system uses the MongoDB database as the primary database, chosen for its capacity and performance to address structured service data, structured user profiles, structured invoices, and service histories. Role-Based Access Control (RBAC) is incorporated to restrict parts of the application to the valid access of a user type (i.e. customer, mechanic, or admin role). Authentication is achieved using JWT (JSON Web Tokens) to manage secure and stateless sessions across the application. The platform is fully hosted on a cloud platform Render that provides seamless development and deployment of continuous integration, auto-scaling capabilities, and the ability to deliver the application performance globally.

4. SYSTEM ARCHITECTURE

Automate development is based on a new, modular architecture that allows scalability, responsiveness, and hassle-free interaction between customers, mechanics, and administrators.

HTML, TailwindCSS, JavaScript, and React.js are used to build the front-end of the platform, providing a dynamic and intuitive interface to all user types. TailwindCSS streamlines developing UIs by enforcing consistent design, while React.js enables reusable architecture to manage handling service bookings, mechanic profiles, and service tracking. Utilizing Axios or React Query allows developers to organize data fetching and the state of the application to support to real-time service updates and create invoices.

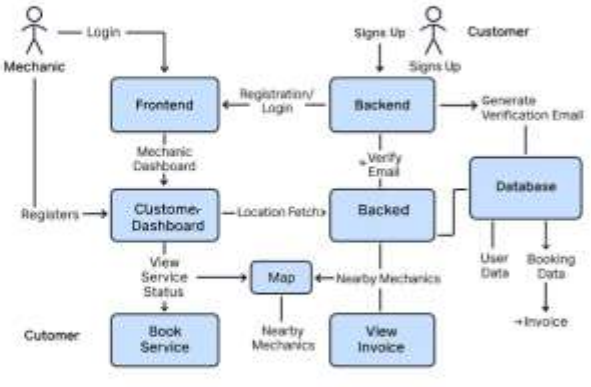

Fig. 1

The Customer-Mechanic Interaction Module is the communication center of Automate. Through this module, customers can view garage profiles, order services, view live updates of their job progress, and give feedback for service. Mechanic users are able to manage appointments, modify service status, and charge customers to generate invoices. It is possible to design user role-based dashboards to ensure users can access only the features that are relevant to them. Customers can search for mechanic shops by location, search through customer ratings, see a real estimate of the cost of the service, and receive predictive alerts for scheduled maintenance.

The recommendation engine supports the customer by recommending mechanic shops nearby (by distance), by ratings and previous service history of the user. These recommendations are produced using clustering and ranking algorithms to ensure accuracy and personalization.

With its capability, the Geolocation Module uses mapping services, such as the Leaflet API, to identify garages near the user during emergencies. The Geolocation Module uses distance-calculating algorithms, such as the Haversine formula, to ascertain accurate geolocation. For example, if multiple services are called in relation to the emergency event, the system should provide the fastest response time possible given its location. The Invoice and Payment Module enables the mechanics to create detailed service invoices that the customer can digitally receive. Within the system, the customer can access and view the invoice and choose to pay for the service using Razorpay, an online payment system that is integrated into the system. The entire invoice and payment process is tracked for review by the customer and by the admin. Additionally, Respect is an important consideration and the Sentiment analysis Module provide a layer of respect monitoring through monitoring the user interactions using the Gen AI model. In certain instances the user may use harmful and/or toxic language messages and the system is publicly observable in real-time with resonated active, passive or rejectivity actions.

5. IMPLEMENTATION STRATEGY

Automate will grow through phases to ensure structured development, modular deployment, and functional integration of AI and geolocations features. This phased approach allows for agile iteration, subsequent testing, and user-driven evolution, resulting in a reliable, scalable, and user friendly vehicle maintenance platform.

A. Phase 1: UI/UX Design and Database Framework: This stage creates an intuitive, easily accessible interface and sets up the underlying database structure. The front-end's visual structure is based on wireframes and commercially accepted UI/UX design standards, enabling customers, mechanics, and administrators to navigate the interface cleanly. The back-end will require organization of a MongoDB database with collections for different user profile types, vehicle information, service history, complaints, payments and feedback. We require user role access by discriminatory access type needed to plan prototype serving specific letter types of users (eg. access to administrative collection metadata only).

B. Phase 2: Core Function Development: In the development phase two, we will build the platform core functionality, e.g., service booking, scheduling mechanisms, and job status update. Customers can search for close-by garages, book a service, as well as monitor the progress in real time. Mechanics can view jobs, change service statuses, and generate digital invoices. The Leaflet API will be integrated so that garages will be geolocation-enabled and hence will be found easily. Secure online payments will also be facilitated by the use of Razorpay on the platform.

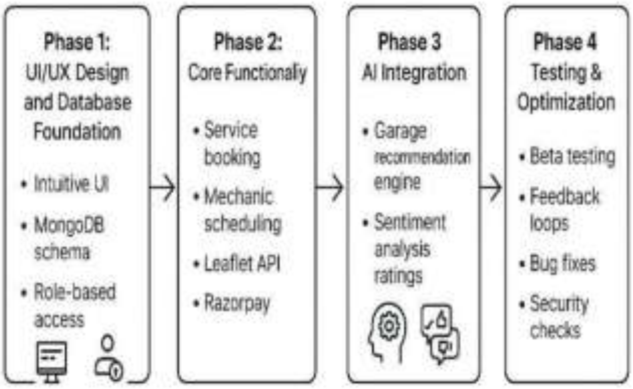

Fig. 2

C. Phase 3: AI Integration and Smart Features: This phase includes AI-driven upgrades to enhance the user interface and operational efficiency. The garage recommendation engine ranks nearby service providers by user rating, proximity, and availability. A Gen AI-based sentiment analysis system is included for ensuring safe communication between the users. It assists in delivering a numeric-based rating value between 0-5, where 0 shows that the mechanic shop is least recommended, whereas 5 shows that the mechanic shop is good and is highly recommended.

D. Phase 4: Testing, Feedback, and Optimization: The final phase is the iterative testing and improvements phase. The platform is launched in beta mode so it can be tested in the real world setting with selected customers, mechanics, and admins. Input is gathered through surveys, session tracking, and bug reports. Based on what is learned, refinements are done to enhance system performance, user experience, and feature reliability. Usability problems, loading times, and edge-case bugs are solved, and security checks are done prior to the full scale deployment of the platform. We test our application in all depth from the customer as well as mechanic perspective. We ensure proper booking is being done or not.

6. EVALUATION AND RESULTS

To assess the impact of Automate on performance, usability, and real-world use a pilot test was conducted with a diverse user group which included vehicle owners, local technicians and administrative testers. Analysis focused on four areas: the experience of booking a service, the efficiency of garage recommendations, and satisfaction with use. Both quantitative and qualitative information was collected in the form of structured questionnaires, feedback sheets, and direct observation of user conduct under platform use. The outcomes were promising: 85% of customers noted enhanced convenience in service reservation vis-à-vis the conventional telephone-based or walk-in setup. Users specifically pointed to real-time status information and the feature to choose garages by rating and distance as significant enhancements.

In fact, the garage recommendation functionality based on geolocation and service ratings received a high score from 80% of customers, especially in emergency situations, where time and distance mattered.

Mechanics found the platform useful in completing job requests, with many commenting how easy it was to recap job status and prepare invoices electronically. Sentiment analysis feedback was also positive, with the platform effectively flagging inappropriate content and ensure a polite environment.

7. CONCLUSION

The Automate platform has been architected to transform the vehicle maintenance and repair environment by introducing digital technology to each phase of the service process. From scheduling appointments and getting estimates for service costs to monitoring job status and paying securely online, Automate eliminates the key inefficiencies of conventional servicing practices. With its modular design, developed using contemporary web technologies, it facilitates seamless role-based interactions between customers, mechanics, and administrators, to make the system user-friendly, scalable, and efficient.

The garage recommendation platform puts a recommendation list together by having users answer key questions about the problems they have with their vehicle - then using geolocation work with the user's report to recommend vetted garages in their area. The sentiment analysis also adds to the dignity of the environment by watching over user interaction, and keeps the professional tone of communications across the platform.

The levels of satisfaction with the evaluations were very high, confirming that Automate met its users' needs while giving the local mechanics the tools to access digital. So not only was the local garage making it easier for its customers' access, but allowing small garages the ability to compete for business in an ever-growing digital marketplace. In the future, Automate has the ability to innovate even further with IoT vehicle diagnosis, advanced analytics and service reminders for vehicle servicing giving it the opportunity to be the beginning of smart vehicle servicing.

8. REFERENCES

1) Node.js Documentation – Official documentation for the server-side JavaScript runtime used for backend development. https://nodejs.org/docs/latest/api

2) MongoDB Documentation – Comprehensive guide for working with the NoSQL database used in scalable applications. https://www.mongodb.com/docs

3) ReactJS Documentation – Reference for building interactive and component-based user interfaces. https://legacy.reactjs.org/docs/getting-started.html

4) Axios API – Documentation for Axios, a promise-based HTTP client used for API calls. https://axios-http.com/docs/intro

5) Generative AI for Hate Speech Detection: Evaluation and findings - https://arxiv.org/abs/2311.09993

6) Geolocation System Module Creation to Validate User Location Coordinates - https://thescipub.com/abstract/jcssp.2023.1050.1064