



MERN STACK-BASED CAR RENTAL WEBSITE DEVELOPMENT

Archana Monish Naware
Computer Engineering Department,
Lokmanya Tilak College of Engineering
Navi Mumbai, India

Sneh Vilas Parab
Computer Engineering Department,
Lokmanya Tilak College of Engineering
Navi Mumbai, India

Swastik Chandrashekhar Bakale
Computer Engineering Department,
Lokmanya Tilak College of Engineering
Navi Mumbai, India

Shrishty Anil Saxena
Computer Engineering Department,
Lokmanya Tilak College of Engineering
Navi Mumbai, India

Sameep Atul Anjaria
Computer Engineering Department,
Lokmanya Tilak College of Engineering
Navi Mumbai, India

Abstract: The Car Rental System is a web-based platform designed to provide a user-friendly interface and a seamless rental experience. It is built using the MERN stack, which ensures fast and responsive user interfaces and efficient processing of customer requests and data. The platform includes a payment gateway for secure and swift payment processing, enabling efficient rental transactions. With the increasing demand for car rental services due to the rise of tourism and ride-sharing services, the platform offers a convenient and efficient solution. The system's advanced functionality enables fast retrieval and management of customer data, along with filtering capabilities to facilitate easy navigation and car listing. By integrating a geo-location API, the platform allows customers to find rental cars quickly and easily. MongoDB serves as the platform's database to store and manage customer information, vehicle information, and rental data. The system's flexibility allows renters and rentees to add and delete cars as per their requirements, reducing the time and effort required to rent a car. Overall, the Car Rental System provides a reliable and efficient solution for car rental companies to offer customizable rental experiences to their customers.

Keywords: MERN stack, geo-location API, MongoDB

I. INTRODUCTION

The Car Rental system is an innovative and sophisticated web-based application designed to provide customers with a seamless and personalized car rental experience. The system was developed to meet the increasing demand for a more convenient and efficient way of renting a car, without the need for in-person visits, long queues, or complicated paperwork.

The Car Rental system leverages the power of the Google Geolocation API to locate available vehicles and display them on a map, enabling customers to select the nearest and most suitable vehicle based on their budget and preference. Additionally, the system incorporates advanced filtering options to refine search results, allowing customers to tailor their search according to specific criteria, such as car type, fuel type, and price range. The Car Rental system is built using the MERN stack, which integrates MongoDB, ExpressJS, ReactJS, and NodeJS, providing a high degree of flexibility, scalability, and ease of development. The web-based platform of the system allows customers to rent cars, compare vehicles, make reservations, and complete the rental process, all in one location. The system is also equipped with robust data management capabilities, enabling it to handle vast amounts of customer data, vehicle

information, rental details, and payment transactions securely and efficiently. One of the system's significant advantages is its payment gateway integration, which ensures the secure and seamless processing of payments, offering customers a range of payment options to enhance their convenience. Furthermore, the Car Rental system provides car rental businesses with an effective and efficient way of managing their customer base, including customer registration, account management, and feedback. The system's data analytics capabilities also provide businesses with valuable insights into customer behavior, rental trends, and system usage, enabling them to make data-driven decisions and improve their overall service quality.

The proposed Car Rental system is a comprehensive, efficient, and customer-centric platform that provides a seamless and personalized car rental experience. Its advanced features, user-friendly interface, and robust data management capabilities make it an ideal solution for car rental businesses of all sizes and individuals looking to rent a car for personal or professional purposes.

II. LITERATURE REVIEW

A. Limitations of Current System

The current system has certain limitations that need to be addressed in order to improve its performance and

efficiency. The limitations of the current system include inadequate data storage capacity, limited processing power, and potential security vulnerabilities. Firstly, the system's data storage capacity may be insufficient to handle the increasing volume of data generated and processed by modern businesses and organizations, leading to delays and inefficiencies. Secondly, the system's processing power may be limited, resulting in slower performance and longer processing times for complex tasks. Additionally, the system may have potential security vulnerabilities, such as weak authentication mechanisms or outdated encryption protocols, which could expose sensitive data to unauthorized access or cyber-attacks. Addressing these limitations through technological advancements and robust security measures is critical to enhance the system's capabilities and ensure its effectiveness in meeting the evolving demands of today's data-driven world.

B. Scope of Proposed System

The proposed system is a car recommendation system with an intuitive and user-friendly interface that allows customers to search, select, and rent vehicles, as well as manage their rental reservations. The system will be designed as an evolving website for customer-to-customer (C2C) interactions, with separate login pages for both companies and users. The system will have features that enable the car rental business to manage their customer base, including customer registration, account management, and customer feedback. Additionally, the system will implement reporting and analytics features that provide insights into customer behavior, rental trends, and system usage, to support data-driven decision-making. The database design and management will store customer and vehicle information, rental reservations, and payment transactions. The system will also integrate a payment gateway with the ability to generate and print invoices for each successful transaction.

III. METHADODOLOGY

Modern software development methodologies have evolved significantly over time, with the traditional Waterfall approach being largely replaced by Agile development. Waterfall, which involves a linear progression through each stage of the software development life cycle, has been shown to be inflexible and less responsive to changing requirements.

In contrast, Agile development is a more iterative and flexible approach that involves revisiting each stage of the development cycle multiple times. This allows for more dynamic requirement gathering, design, and development processes, which can lead to faster and more effective software development. Agile development contains several methodologies, including Extreme Programming (XP), Kanban, and Scrum. For this project, we adopted the Scrum method of Agile development, which is a widely used framework in team settings, but we found it to be an effective methodology for individual development as well. The Scrum process involves breaking down the project into smaller, manageable pieces known as sprints, with each sprint aiming to achieve the full development of a set of user stories. To begin, we generated a product backlog via user stories in the requirements gathering stage, which served as the foundation for the project. The overall development of

the application was then split into sprints, with each user story assigned to a sprint in a logical order. We utilized Jira, a free online software tool, for managing the tasks involved within sprints and tracking issues. This allowed us to organize the development process effectively and ensure that each sprint was completed before moving on to the next. One of the key benefits of the Scrum method is its flexibility. It allows developers to easily adapt to changing requirements, as the project progresses. The agile nature of Scrum enables developers to make adjustments and improvements in real-time. This results in the delivery of a high-quality product that meets the changing needs of the user. Furthermore, Scrum encourages collaboration and communication between team members, which results in a more efficient and productive development process. By working closely together and continuously providing feedback, team members can identify and address issues early on, reducing the likelihood of mistakes or delays. In conclusion, the Scrum methodology is an effective way to develop software, whether as an individual or as part of a team. Its iterative and flexible approach allows for quick adaptation to changing requirements, while also encouraging collaboration and communication among team members. By utilizing Scrum, we were able to successfully deliver a high-quality product that met the project's objectives within the allotted time frame.

A. Operational Framework

To carry out our study, we developed a comprehensive operational framework that was tailored to meet the specific needs and objectives of our research. This framework involved a range of processes and procedures that were meticulously planned and executed, allowing us to collect, analyse, and interpret data efficiently and effectively. Some of the key components of our operational framework included defining the project's scope and objectives, identifying the target population and strategy, selecting appropriate tools and techniques, developing a systematic and standardized system to collect and organise data, conducting statistical analyses of the survey data, and presenting the findings in a clear and meaningful manner. It is important to note that all of these components were developed with our project's unique requirements in mind.

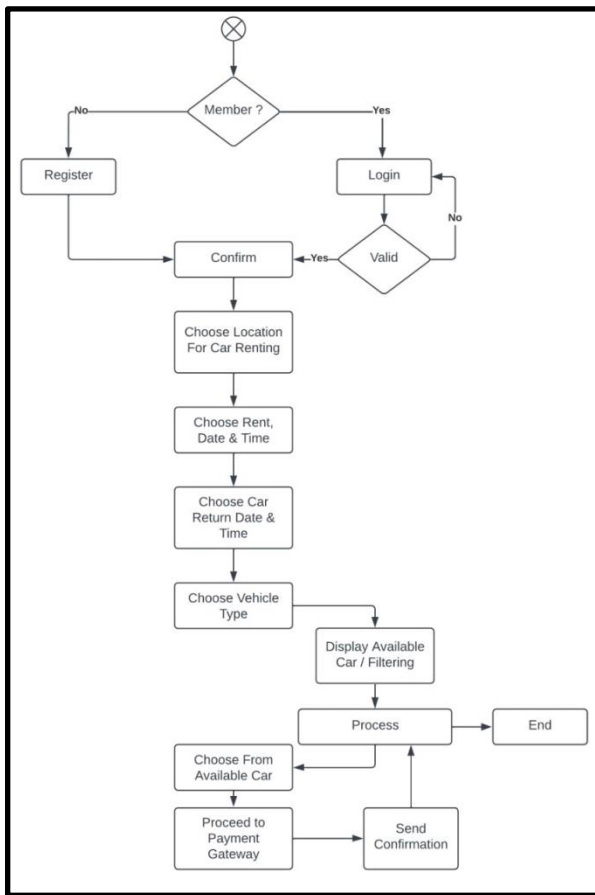


Figure 1. Flowchart of operational framework

B. Data Sets

The success of any vehicle rental system heavily relies on the efficient management of data sets. In the case of this project, there are three core data sets that are essential to consider, namely users, vehicle owners, and vehicles. First and foremost, the user data set is pivotal, as they are the primary audience of the system. This data set comprises a diverse range of individuals from varying age groups and income classes, each with their unique preferences and characteristics. Additionally, users may hold different types of vehicle licenses, with varying degrees of experience in vehicle rentals, and may or may not possess insurance coverage. On the other hand, the vehicle owner data set is equally important, as they provide the vehicles to be rented by the clients. This data set is similar to that of the user data set, with some additional information, including the type of vehicles they own, their respective mileage, brand, and overall condition. Lastly, the vehicle data set comprises information such as the type of vehicle, size, number of seats available, and brand, all of which must be accurately managed to ensure the system runs efficiently. Overall, managing these data sets effectively will be crucial to the success of the vehicle rental system, and as such, appropriate measures must be put in place to ensure that these data sets are accurately captured, stored, and utilized throughout the system.

In addition to these primary datasets, we also used a range of secondary datasets, such as government reports, academic studies, and industry statistics, to supplement and contextualize our findings.

C. UML Diagrams

To improve the visualization and communication of our system's design and functionality, we utilized Unified Modeling Language (UML) diagrams. These diagrams, including Use Case Diagrams, Class Diagrams, and Activity Diagrams, allowed us to showcase the relationships between users and the software system, the structure and hierarchy of classes and objects, and the different steps involved in completing a specific task or function. Through the use of UML diagrams, we were able to present a clear and concise overview of the system to users. Furthermore, the diagrams aided us in identifying and addressing potential design or implementation issues early on, which helped us to mitigate any significant problems.

Overall, the implementation of UML diagrams was an essential component of our design process, as it enabled us to create a comprehensive and detailed plan for our system's functionality, ultimately leading to a more efficient and user-friendly product.

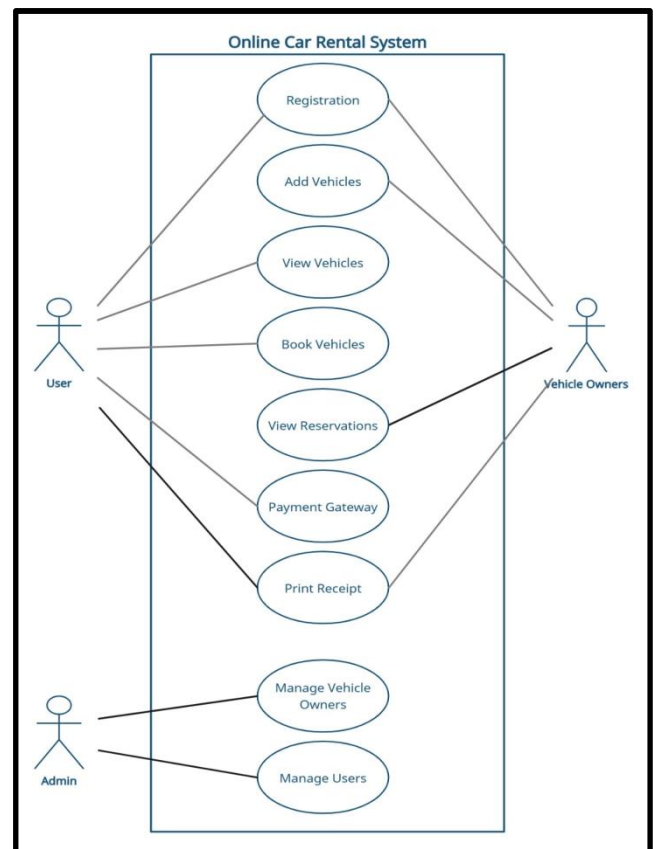


Figure 2. Use Case Diagram of System

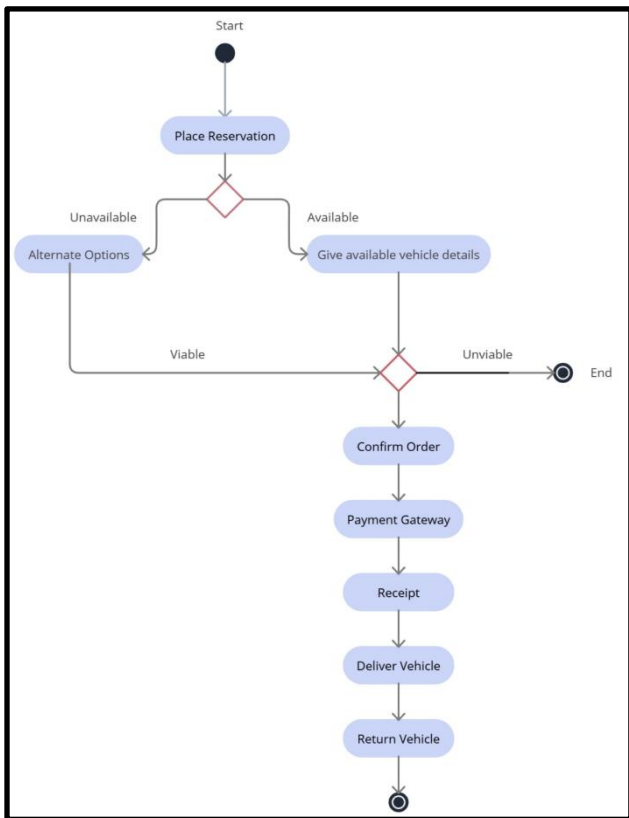


Figure 3. Activity diagram of system

IV. RESULTS

The results of our project are presented here, showcasing the data collected from the evaluations and reviews of our Online Car Rental System, aligned with the objectives highlighted in the introduction section of Chapter 1. Furthermore, a total of 35 participants took part in our study, providing feedback through questionnaires after thoroughly testing the application.

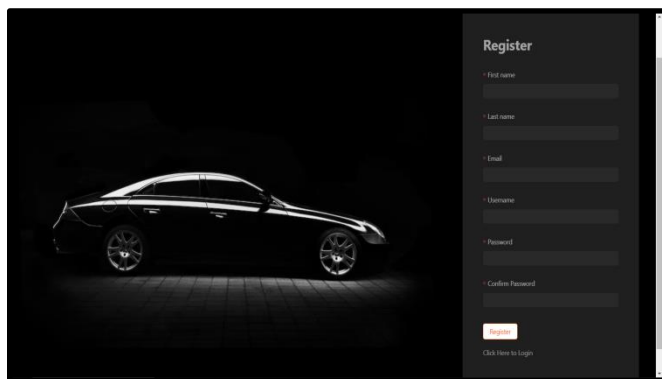


Figure 4. Registration page of the system

Figure 4 exhibits the registration page of the system, comprising fields for the user's name, last name, username, and password to input their personal details. The page also includes a register button for submission. Moving on to Figure 5, it presents the main menu page of the system, where users can select from a list of available vehicles for rent. Once a user has chosen their desired vehicle, the

application will automatically redirect them to the respective vehicle page.

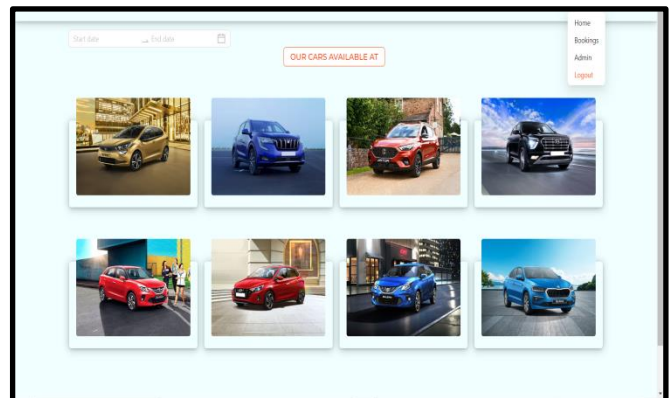


Figure 5. Main Page after Login

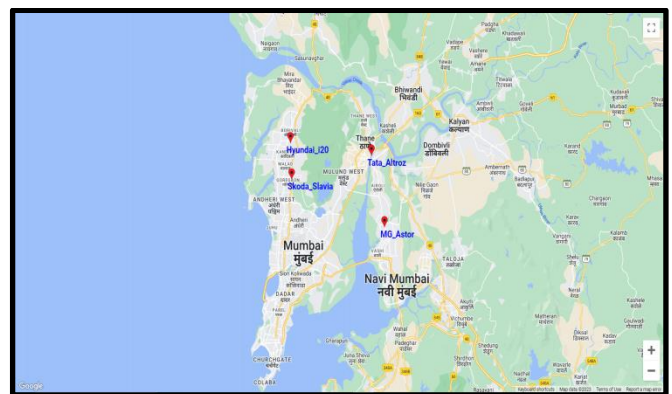


Figure 6. Locations of available cars

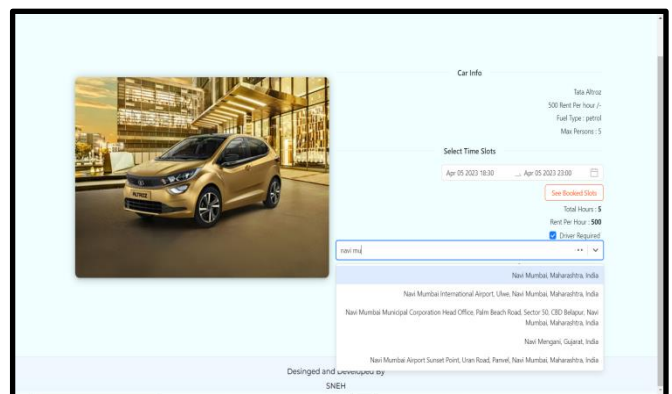


Figure 7. Adding details & fare calculation after car selection

Figure 6 presents an interactive map that displays the current locations of all available cars for rent. Users can easily identify the available vehicles based on their geographical locations. On the other hand, Figure 7 illustrates the chosen vehicle page, where users can input various trip details, including the date, time, and location for pickup. The system then automatically calculates the fare for the trip based on the provided information, providing a seamless user experience.

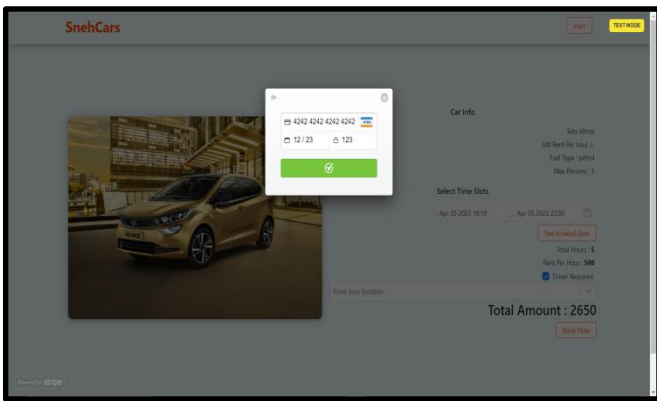


Figure 8. Payment page

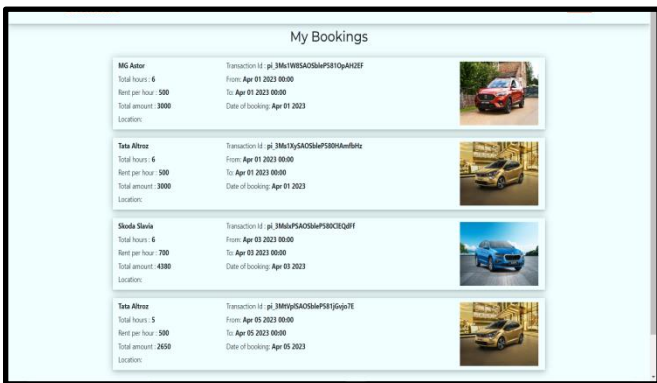


Figure. 9 User’s bookings page

Figure 8 exhibits the payment gateway integrated with the Stripe API, providing users with a secure and convenient option to make payments using their debit/credit cards. This feature ensures a seamless and hassle-free payment process for users. While, Figure 9 showcases the bookings history page, displaying all the bookings ever made by the user. This page includes comprehensive details about each trip, such as the car make and model, rent for the specified time period, final cost of the trip, exact time and date of the booking, as well as the transaction ID associated with each transaction. This information provides users with a complete overview of their booking history, facilitating easy tracking and management of past trips.

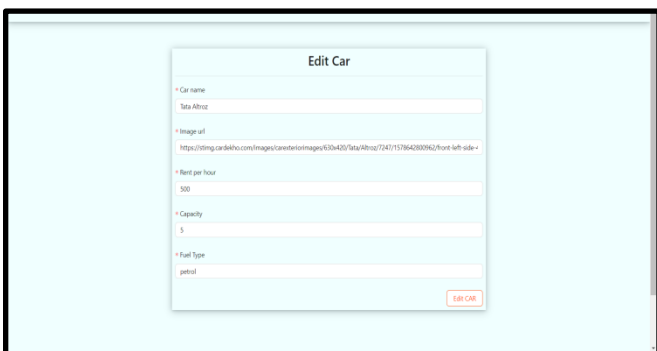


Figure 10. Input of car details

Figure 10 showcases the car rental posting editing page, intended for the providers of the rental cars rather than the

customers. This page allows car rental providers to modify various details related to their posted vehicles. These include editing the name of the vehicle, updating pictures, changing the rental price, adjusting passenger capacity, and modifying the fuel type of the vehicle. This feature empowers car rental providers with the flexibility to manage and update their car listings according to their preferences and requirements.

V. CONCLUSION

In conclusion, the MERN stack-based Car Rental System provides a modern and efficient solution for car rental companies and individual renters. With advanced functionality, including geo-location integration, filtering capabilities, and payment gateway integration, the platform offers a convenient and personalized rental experience. The use of the Scrum methodology allowed for efficient and flexible development, resulting in a high-quality product that meets user requirements. The system's robust data management capabilities enable secure and efficient handling of vast amounts of customer data, vehicle information, rental details, and payment transactions. Overall, the Car Rental System is a comprehensive and reliable platform that provides an ideal solution for car rental businesses and individuals looking to rent a car for personal or professional purposes.

VI. ACKNOWLEDGMENT

"We would like to express our sincere gratitude to the following individuals and organizations who supported us throughout this project:

Prof. Archana Naware, our project guide, for her guidance, expertise, and encouragement. Our colleagues and friends for their valuable feedback, ideas, and support throughout the project. We appreciate the learning experience and the opportunity to work on this project."

VII. REFERENCES

- [1] Kesrarat, D., Songcharoenkit, S., Nanthapornpisut, P., & Thawonthammarat, L. (2017, February). Smart Matching for Car Rental. In Proceedings of the 9th International Conference on Machine Learning and Computing (pp. 529-533). ACM. .
- [2] Khaled, M. S. M., Arefin, S., Kumar, D. S. R., & Tuhin, A. H. (2015). Software Requirements Specification for Online Car Rental System.
- [3] S. L. Fong, D. W. Y. Chin, R. A. Abbas, A. Jamal, and F. Y. Ahmed, "Smart City Bus Application With QR Code: A Review," in 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2019: IEEE, pp. 34-39.
- [4] S. R. Manalu, A. Wibisurya, N. Chandra, and A. P. Oedijanto, "Development and evaluation of mobile application for room rental information with chat and push notification," in 2016 International Conference on Information Management and Technology (ICIMTech), 2016: IEEE, pp. 7-11
- [5] Saufi, N. N. C., Razak, N. S. M., & Mansor, H. (2019, January). FoRent: vehicle forensics for car rental system. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy (pp. 153-157). ACM.

- [6] Li, Z. (2013). Design and realization of car rental management system based on AJAX+ SSH. Information Technology Journal, 12(14), 2756-2761.
- [7] Sapuan, M. K. M. (2012). Rental Car Online System (Doctoral dissertation, UMP).