**PROJECT REPORT ON**

**CAR BUYING SITE  
(SUTO AUTOMOBILES)**

Submitted in partial fulfilment of the requirement for the

Course BEE (22CS026) of

# COMPUTER SCIENCE AND ENGINEERING

**B.E. BATCH – 2022**

**IN**

**JAN – 2025**



|  |  |
| --- | --- |
| **Under the Guidance of** | **Submitted By: -** |
| Shaffu | Suyash Dubey |
|  | 2210990880,  Tanish Dalal  2210990888 |
|  |  |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CHITKARA UNIVERSITY

# PUNJAB

# CERTIFICATE

This is to certify that the project entitled ‘**Car Buying Website**’ has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab, during the academic semester July 2024 - December 2024. It is a bona fide piece of project work carried out by **Suyash Dubey(2210990878), Tanish Dalal(2210990888)** and towards the partial fulfilment for the award of the course **Integrated Project (CS 203)** under the guidance and supervision of **Shaffu**.

**Signature of Project Guide:**

Shaffu

[DCSE]

# CANDIDATE DECLARATION

I**, Suraj Dubey-2210990880 and Tanish Dalal-2210990888,** from B.E.-2022 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled ‘**Car Rental Website’** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

|  |  |
| --- | --- |
| **Signature** |  |
| Suyash Dubey Tanish Dalal |  |
| 2210990880 2210990880 |  |
| **Place: Punjab** |  |
| **Date: 2/10/2024** |  |

# ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour and acts during the course of study.

We express our sincere gratitude to all for providing us an opportunity to undergo Integrated Project as a part of the curriculum.

We are thankful to **Shaffu** for her support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to **Mr. Vimal Jawla** who provided his valuable suggestions and precious time in accomplishing our integrated project report.

Lastly, we would like to thank the almighty and my parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve my quality of work.

|  |  |
| --- | --- |
| **Suyash Dalal** | **Tanish Dalal** |
| **2210990878** | **2210990888** |

**TABLE OF CONTENTS**

[1. ABSTRACT 5](#_Toc184973535)

[2. INTRODUCTION 6](#_Toc184973536)

[2.1 Background 6](#_Toc184973537)

[2.2 Problem Statement 7](#_Toc184973538)

[3. SOFTWARE AND HARDWARE REQUIREMENTS 8](#_Toc184973539)

[3.1 Methodology 8](#_Toc184973540)

[3.2 Programming/Working Environment 8](#_Toc184973541)

[3.3 Requirements to run application 9](#_Toc184973542)

[4. DATABASE ANALYSING, DESIGN AND IMPLEMENTATION 10](#_Toc184973543)

[4.1 Database Design 10](#_Toc184973544)

[4.2 Implementation 11](#_Toc184973545)

[5. PROGRAM’S STRUCTURE ANALYZING AND GUI CONSTRUCTING 12](#_Toc184973546)

[6.LIMITATIONS 18](#_Toc184973547)

[7.CONCLUSION 18](#_Toc184973548)

[8.FUTURE SCOPE 19](#_Toc184973549)

[9.REFERENCES 19](#_Toc184973550)

## 1. ABSTRACT

The car buying eCommerce platform is designed to revolutionize the vehicle purchasing process by providing a comprehensive, user-friendly digital marketplace that seamlessly connects buyers and sellers. Utilizing a sophisticated backend developed with Express.js and MongoDB, the platform ensures robust scalability and reliability, while the frontend, powered by EJS templates, delivers dynamic and responsive web interfaces. Customers can explore an extensive vehicle inventory, compare specifications, access detailed pricing information, and complete purchases through an intuitive, feature-rich platform.

The system offers advanced features including secure user authentication, personalized vehicle recommendations, comprehensive vehicle history reports, interactive comparison tools, and integrated financing options. For sellers and administrators, the platform provides powerful management tools to list vehicles, track sales, monitor transactions, and generate detailed analytics reports. Strategic integration with modern APIs, including payment gateways, vehicle verification services, and financial institutions, creates a streamlined and interconnected automotive marketplace ecosystem.

Built with rigorous security protocols, the platform incorporates advanced data encryption, secure authentication mechanisms, and comprehensive transaction protection to safeguard user information and financial data. By addressing critical challenges in the automotive sales landscape, the platform reduces transactional complexity, enhances transparency, and improves overall user experience. Leveraging cutting-edge web technologies and adhering to best development practices, this eCommerce solution establishes a robust foundation for digital transformation in vehicle sales, effectively meeting the sophisticated needs of both individual buyers and automotive businesses.

## 2. INTRODUCTION

### 2.1 Background

The car buying industry has undergone significant transformation driven by increasing digital adoption, evolving consumer behavior, and technological advancements. Traditional car dealerships often rely on manual and in-person processes that can be time-consuming and less convenient, leading to inefficiencies and potential customer dissatisfaction. To address these challenges, there has been a shift toward online platforms that offer streamlined car purchasing experiences and enhanced customer engagement.

Modern car buying platforms aim to bridge the gap between buyers and sellers by providing features such as real-time inventory listings, detailed vehicle information, and secure payment options. These platforms leverage advancements in web development, cloud computing, and database management to ensure reliability, scalability, and security. The integration of APIs for vehicle data, geolocation, and payment gateways further enriches the functionality, offering convenience and flexibility to users.

As the demand for digital car purchasing solutions grows, businesses require robust platforms that cater to their operational needs while delivering a seamless user experience. This project aims to develop a comprehensive car buying eCommerce website for Suto Automobiles using Express.js, MongoDB, and EJS. By incorporating features such as user authentication, inventory management, and data analytics, this platform seeks to modernize car buying operations and provide a competitive edge in the evolving automotive market landscape.

### 2.2 Problem Statement

The modern car buying industry faces several key challenges, particularly in terms of user-centric features, efficient purchasing processes, and seamless inventory management. While many platforms offer basic functionalities for browsing and purchasing vehicles, users and businesses often encounter limitations in features that cater to their specific needs and expectations. The primary issues in this space include:

**I. Limited Vehicle Search and Comparison Tools**

* **Problem**: Existing car buying platforms often lack advanced filtering, real-time inventory updates, and comparison features to help users find and evaluate suitable vehicles.
* **Consequence**: Customers experience difficulty locating vehicles that match their preferences and requirements, leading to missed opportunities and reduced satisfaction.

**II. Inefficient Buying and Payment Processes**

* **Problem**: Many platforms do not offer streamlined workflows for purchasing or financing vehicles, nor do they support multiple secure payment options.
* **Consequence**: Users face frustration during the buying process, leading to abandoned transactions and loss of potential sales.

**III. Poor Customer and Seller Communication**

* **Problem**: Communication between buyers and sellers is often limited, with no real-time support or updates on inquiries or purchase statuses.
* **Consequence**: Customers experience uncertainty during the purchasing journey, while sellers struggle with managing inquiries effectively and ensuring a smooth experience.

**IV. Lack of Comprehensive Inventory Management Tools for Sellers**

* **Problem**: Sellers often rely on manual processes or limited tools to manage vehicle listings, pricing, availability, and customer feedback.
* **Consequence**: Businesses face operational inefficiencies, slower turnover of vehicles, and difficulty in meeting customer expectations.

## 3. SOFTWARE AND HARDWARE REQUIREMENTS

### 3.1 Methodology

The development of the car buying eCommerce website follows a structured methodology to ensure functionality, user-friendliness, and scalability. The process begins with a comprehensive requirement analysis, gathering insights from customers and businesses to identify essential features such as vehicle search, purchase workflows, and secure payment integration. A modular system design is created using Express.js for the backend and EJS for server-side rendering, supported by a MongoDB database to manage vehicle listings, user accounts, and transactions.

The implementation involves developing API endpoints and integrating third-party services such as payment gateways and geolocation tools. The frontend is enhanced for seamless interactivity using SCSS for styling, ensuring a responsive and visually appealing design. Finally, the platform is deployed on a scalable cloud infrastructure, with ongoing maintenance and updates guided by user feedback to continuously enhance features and the overall user experience.

**3.2 Programming/Working Environment**

**Frontend:**

* **React.js**: For creating dynamic user interfaces and enabling real-time interactivity.
* **SCSS**: For styling, enabling modular and maintainable CSS with features like variables and nested rules, while ensuring a responsive design across devices.

**Backend:**

* **Node.js**: For server-side scripting and handling API requests.
* **Express.js**: For routing and middleware handling, ensuring smooth communication between the frontend and backend.
* **MongoDB**: As the primary NoSQL database for storing user data, vehicle listings, transactions, and admin-related information.

**Development Tools:**

* **Visual Studio Code**: For code editing and debugging.
* **Git/GitHub**: For version control and collaboration.
* **Mongosh**: For managing MongoDB databases and querying collections effectively.

### 3.3 Requirements to run application

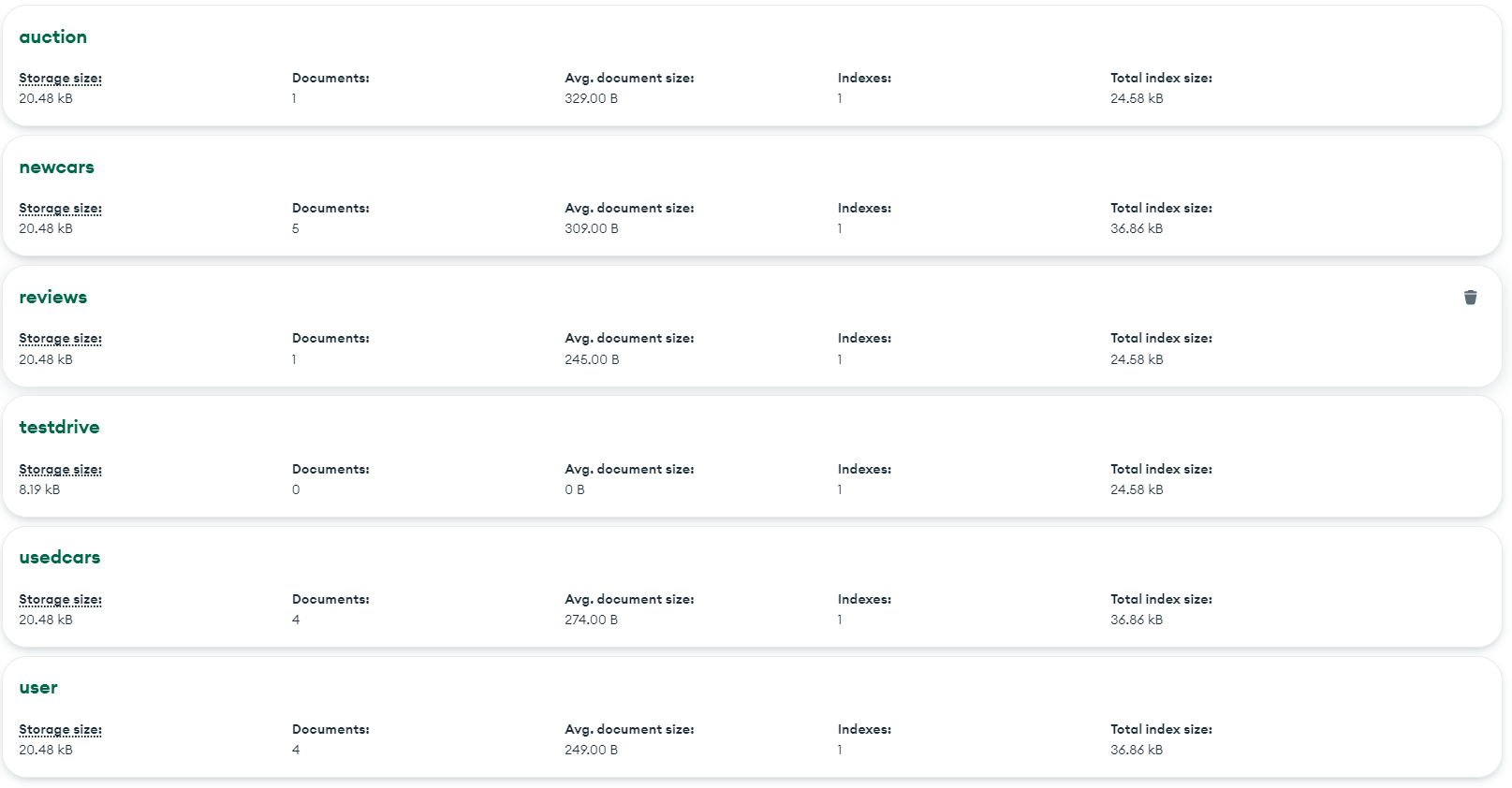
**Software Requirements:**

1. **Operating System:** 
   * Windows, MacOS, or Linux, with support for Node.js.
2. **Required Software:** 
   * **Node.js** (v14.0 or higher) to run the backend and manage dependencies.
   * **MongoDB** (v4.4 or higher) for database management.
3. **Dependencies:** 
   * **React.js** for frontend development.
   * **Express.js** for server-side routing.
   * **Mongoose** for MongoDB integration.

**Hardware Requirements:**

1. **Processor:** 
   * Minimum: Intel Core i3 or equivalent.
2. **RAM:** 
   * Minimum: 2 GB.
3. **Storage:** 
   * Minimum: 20 GB free space and will grow further as the user adds podcasts and music.

## 4. DATABASE ANALYSING, DESIGN AND IMPLEMENTATION

****

**Objective:**

The platform aims to provide a seamless car-buying experience by allowing users to explore available cars, compare options, book test drives, and complete purchases efficiently. Simultaneously, it offers businesses tools to manage car listings, track customer interactions, and analyze sales through a well-structured system.

**Key Entities:**

1. **New Cars**: Stores details about new vehicles, including car IDs, models, specifications, availability, and pricing.
2. **Used Cars**: Records information about pre-owned vehicles, including condition, mileage, pricing, and availability.
3. **Auction**: Manages auction-specific vehicle data, including auction IDs, bidding details, and availability for competitive sales.
4. **Reviews**: Stores customer reviews and ratings for vehicles, providing insights into user satisfaction and vehicle performance.
5. **Test Drive**: Tracks test drive bookings, including customer details, preferred vehicle, and scheduled times.
6. **User**: Maintains user profiles, including personal details, preferences, and purchase history.

**User Interaction**

Users can explore available cars, book test drives, apply for loans or financing, and track their purchase progress seamlessly.

### 

**4.1 Database Design**

**The database consists of the following collections:**

* **NewCars:** Contains details about new vehicles, including make, model, specifications, pricing, and availability.
* **UsedCars:** Stores information about pre-owned vehicles, such as condition, mileage, pricing, and seller details.
* **Auction:** Manages auction-related data, including auction IDs, bidding details, starting prices, and availability for competitive vehicle sales.
* **Reviews:** Maintains user-generated reviews and ratings for vehicles, helping buyers make informed decisions.
* **TestDrive:** Tracks test drive booking details, such as customer information, vehicle ID, and preferred dates/times**.**
* **User:** Stores user registration data, including login credentials, profile information, and purchase history.

**Schema Design**:

* **NewCars Collection:** {car\_id, model, make, year, price, specifications, availability, location }
* **UsedCars Collection:** {car\_id, model, make, year, price, mileage, condition, availability, location }
* **Auction Collection:** {auction\_id, car\_id, starting\_price, current\_bid, bid\_end\_date, status }
* **Reviews Collection:** {review\_id, user\_id, car\_id, rating, review\_text, timestamp}
* **TestDrive Collection:** {test\_drive\_id, user\_id, car\_id, scheduled\_date, status, comments }
* **User Collection:** {user\_id, name, email, phone, registered\_at, purchase\_history }

**Relationships**:

1. **User -TestDrive:**

Type: One-to-Many : A user can schedule multiple test drives, but each test drive isassociated with one user.

1. **User - Reviews:**

Type: One-to-Many : A user can write multiple reviews, but each review is associated with one user.

1. **Cars - TestDrive:**

Type: One-to-Many : A car can have multiple test drive bookings, but each test drive is associated with one car.

1. **Cars - Reviews:**

Type: One-to-Many : A car can have multiple reviews, but each review is associated with one car.

1. **Cars - Transactions:**

Type: One-to-One : A car can only be sold once, and each transaction is linked to one car.

1. **Cars - Auction:**

Type: One-to-One or One-to-Many : A car can either be listed in one auction or not listed at all.

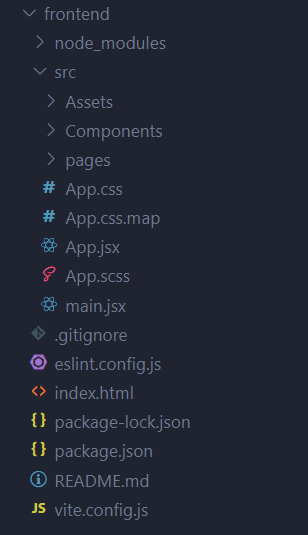
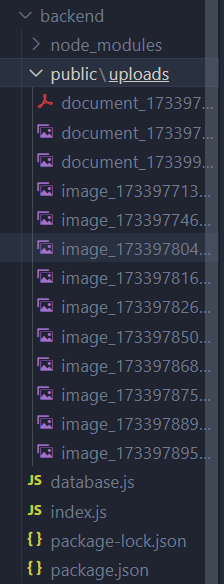
### 4.2 Implementation

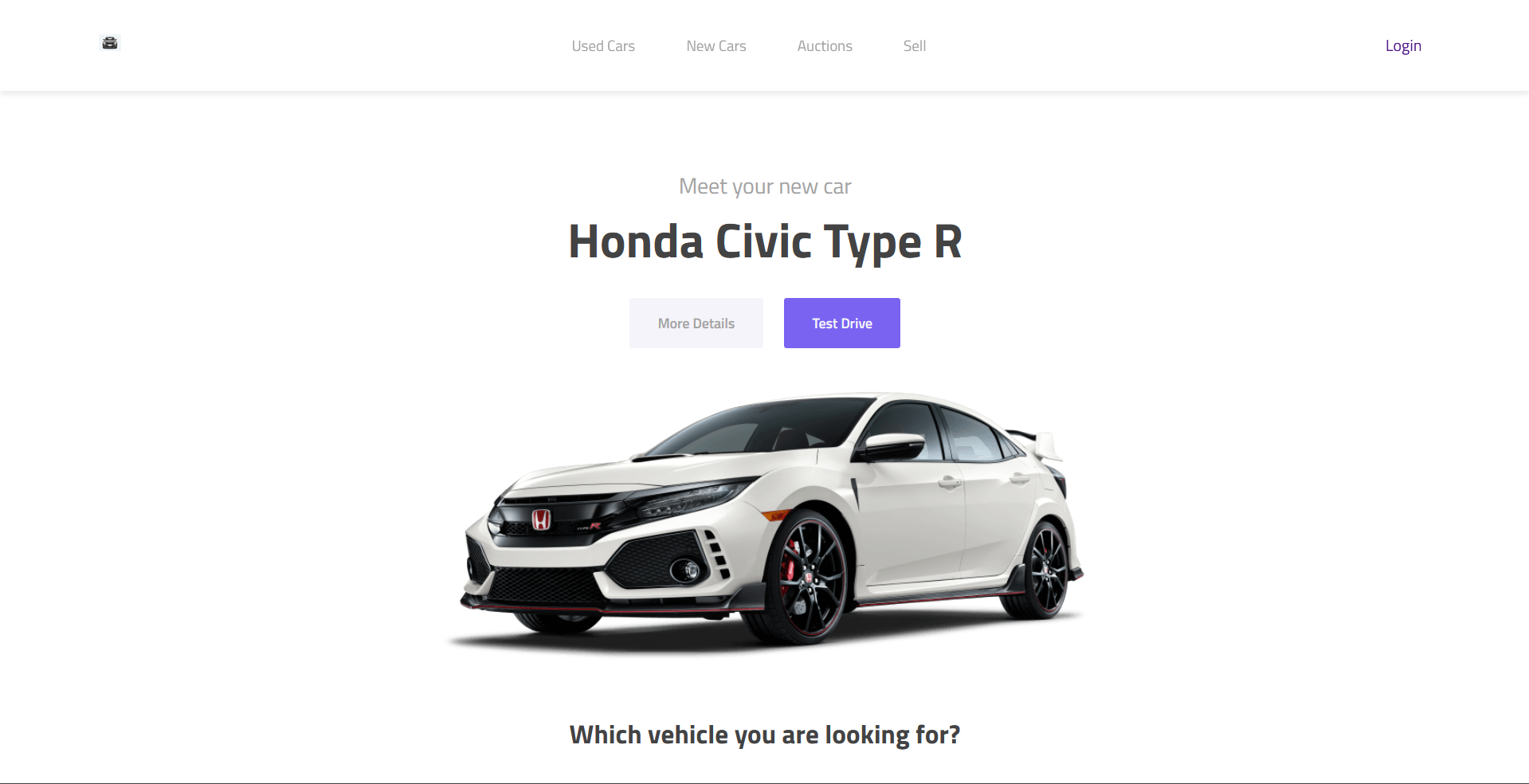
* **Database Setup**:
  1. Created using MongoDB, with collections for each entity.

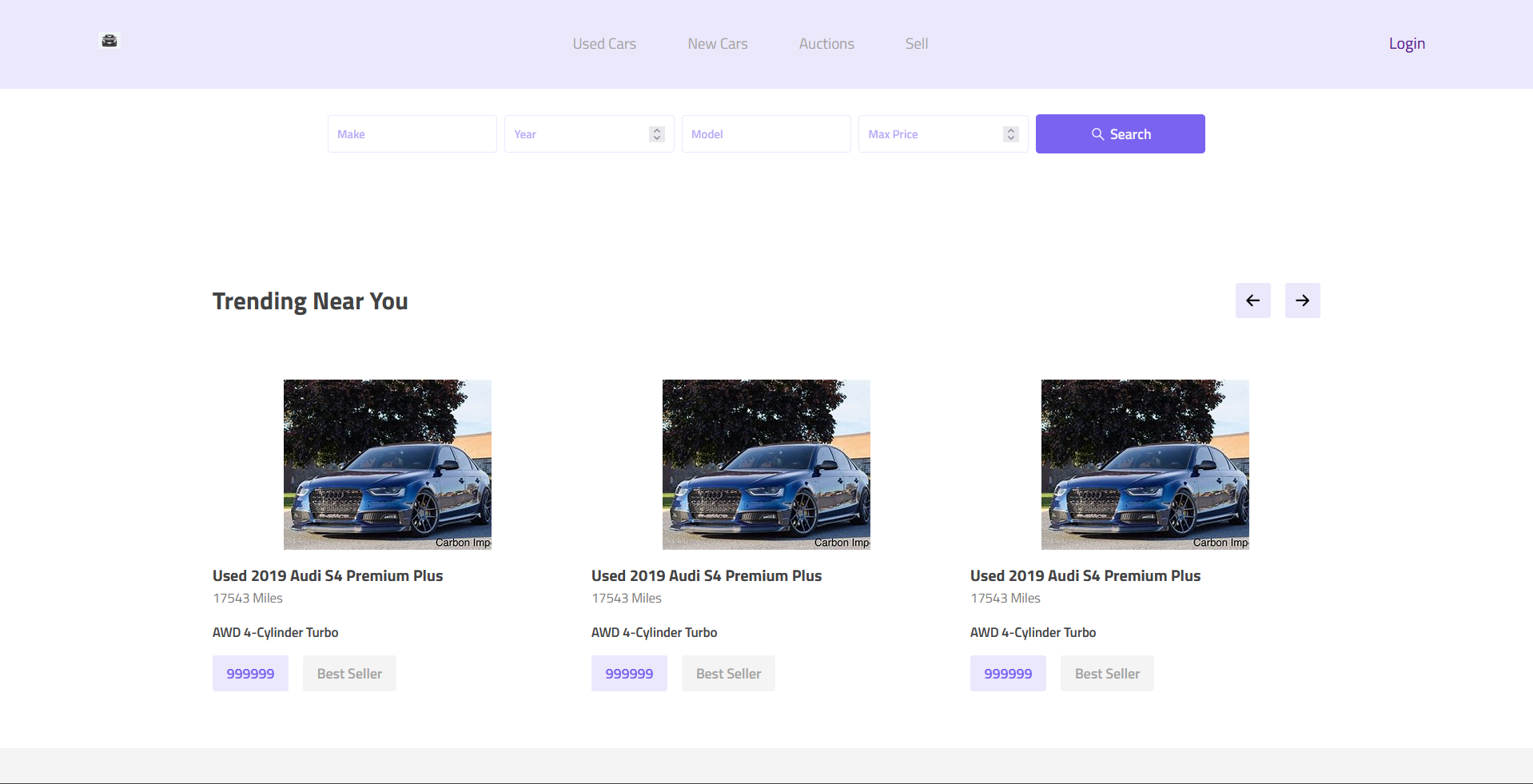
○ Indexes implemented for efficient querying

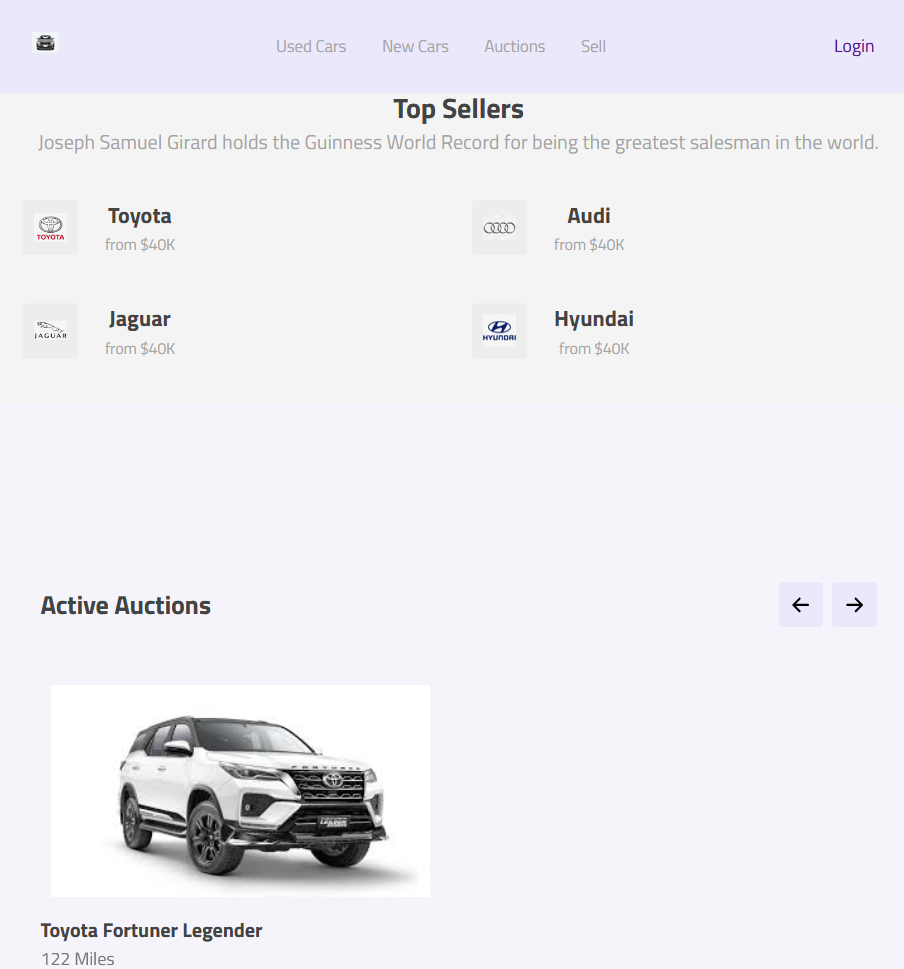
* **CRUD Operations**:
* **Create:**Add a new auction, car (new or used), review, test drive booking, or user account.
* **Read:** Retrieve details of an auction, car (new or used), review, test drive booking, or user account (e.g., auction ID, car ID, user ID, etc.).
* **Update:** Modify the details of an existing auction, car (new or used), review, test drive booking, or user account.
* **Delete:** Remove an auction, car (new or used), review, test drive booking, or user account.

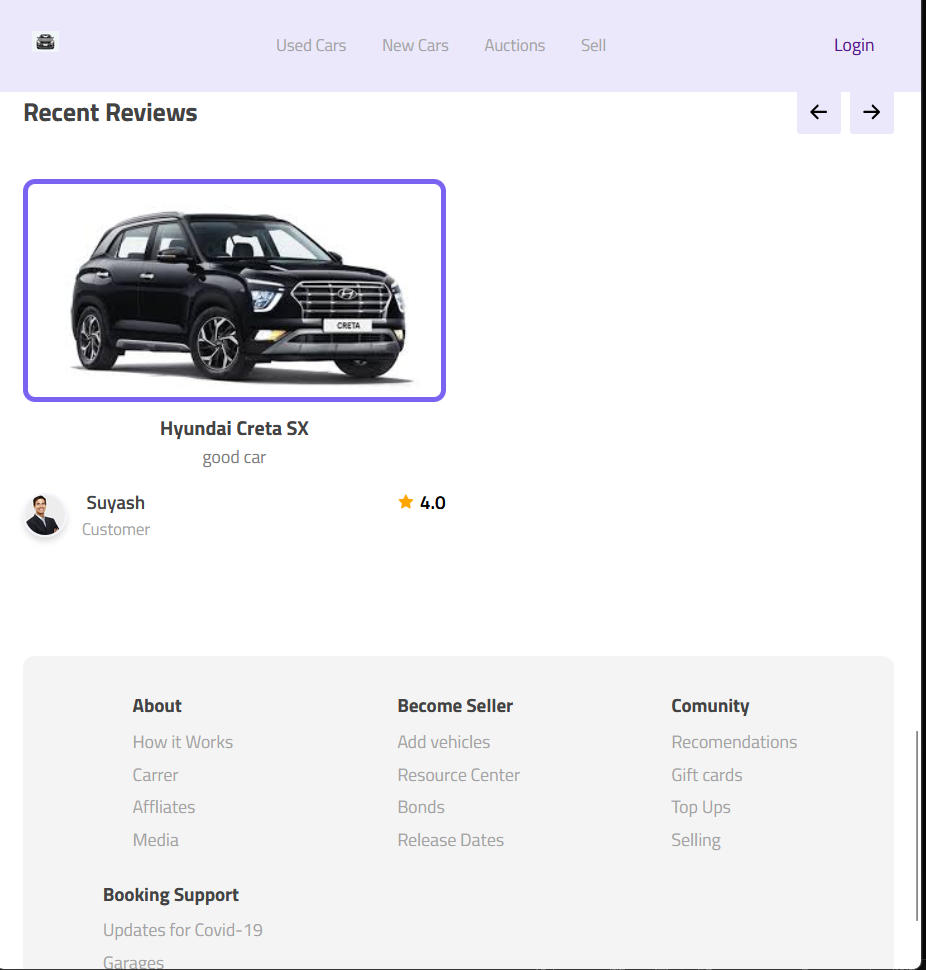
## 5. PROGRAM’S STRUCTURE ANALYZING AND GUI CONSTRUCTING

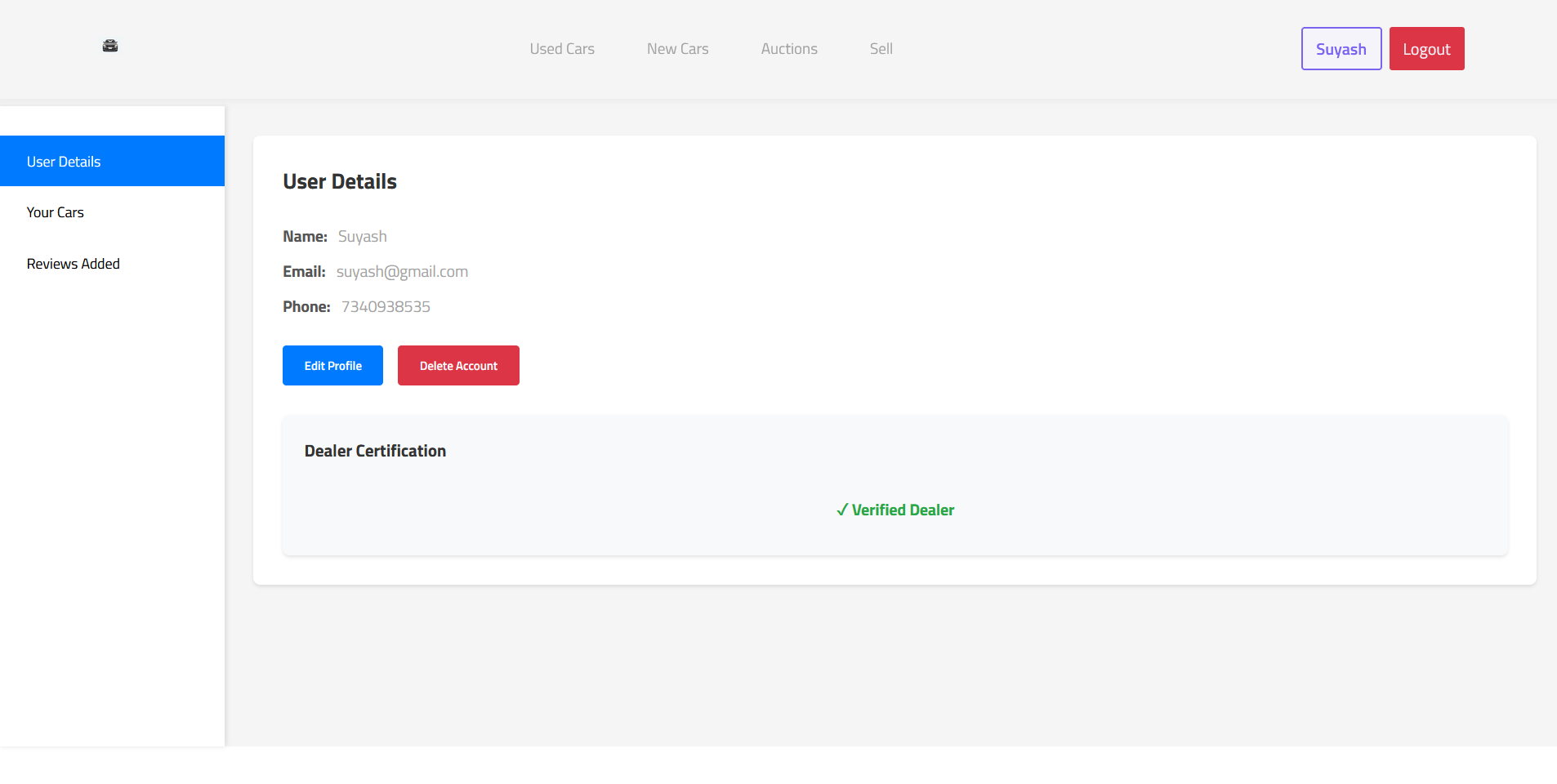
****

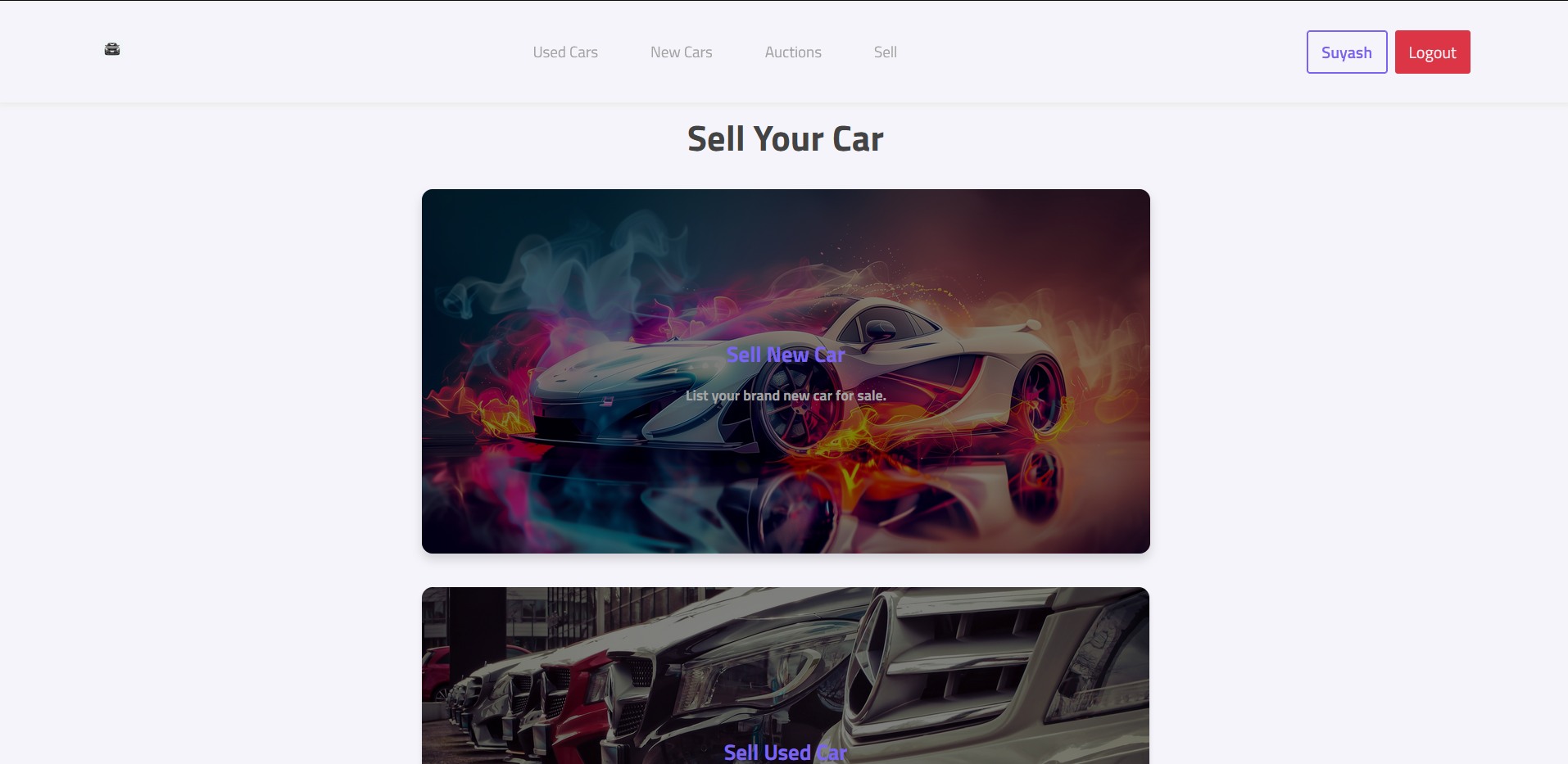


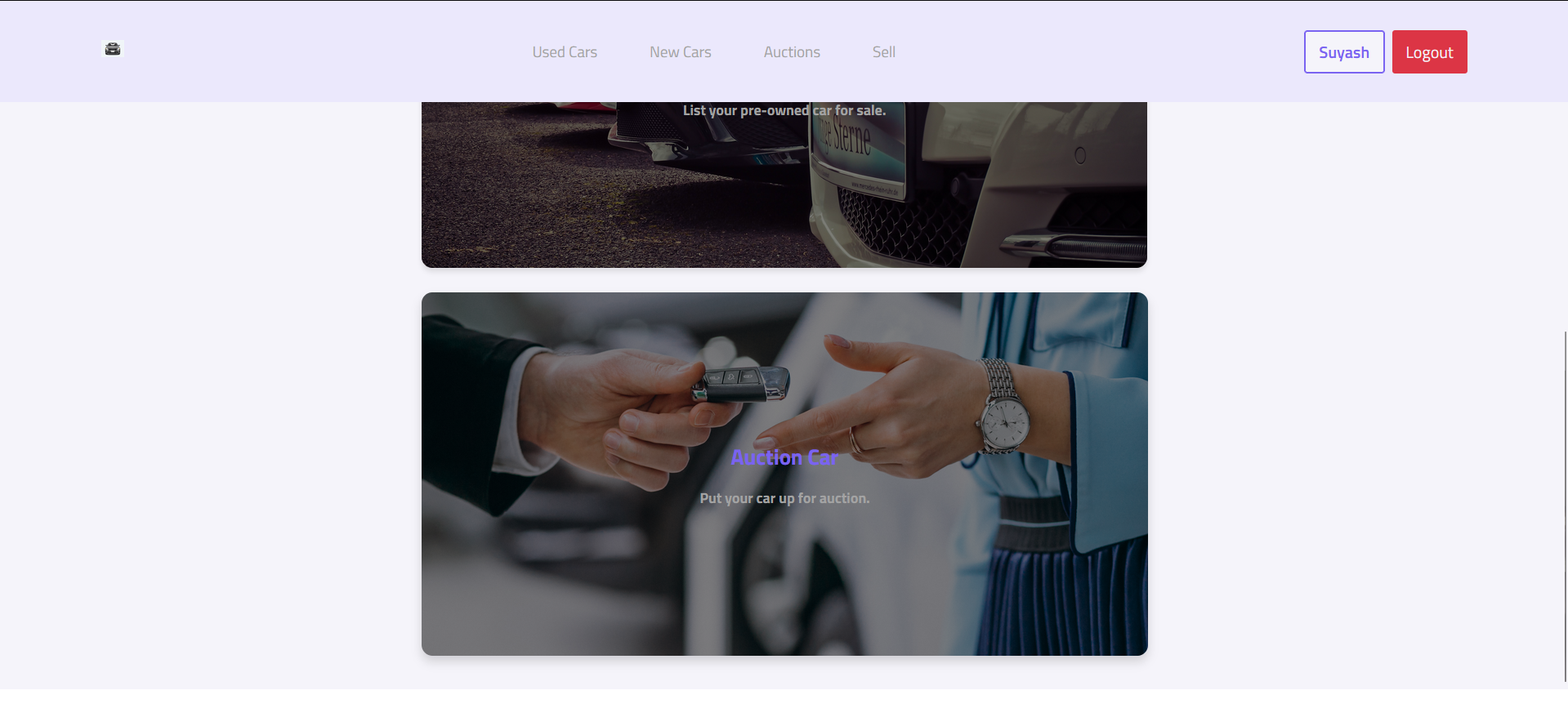


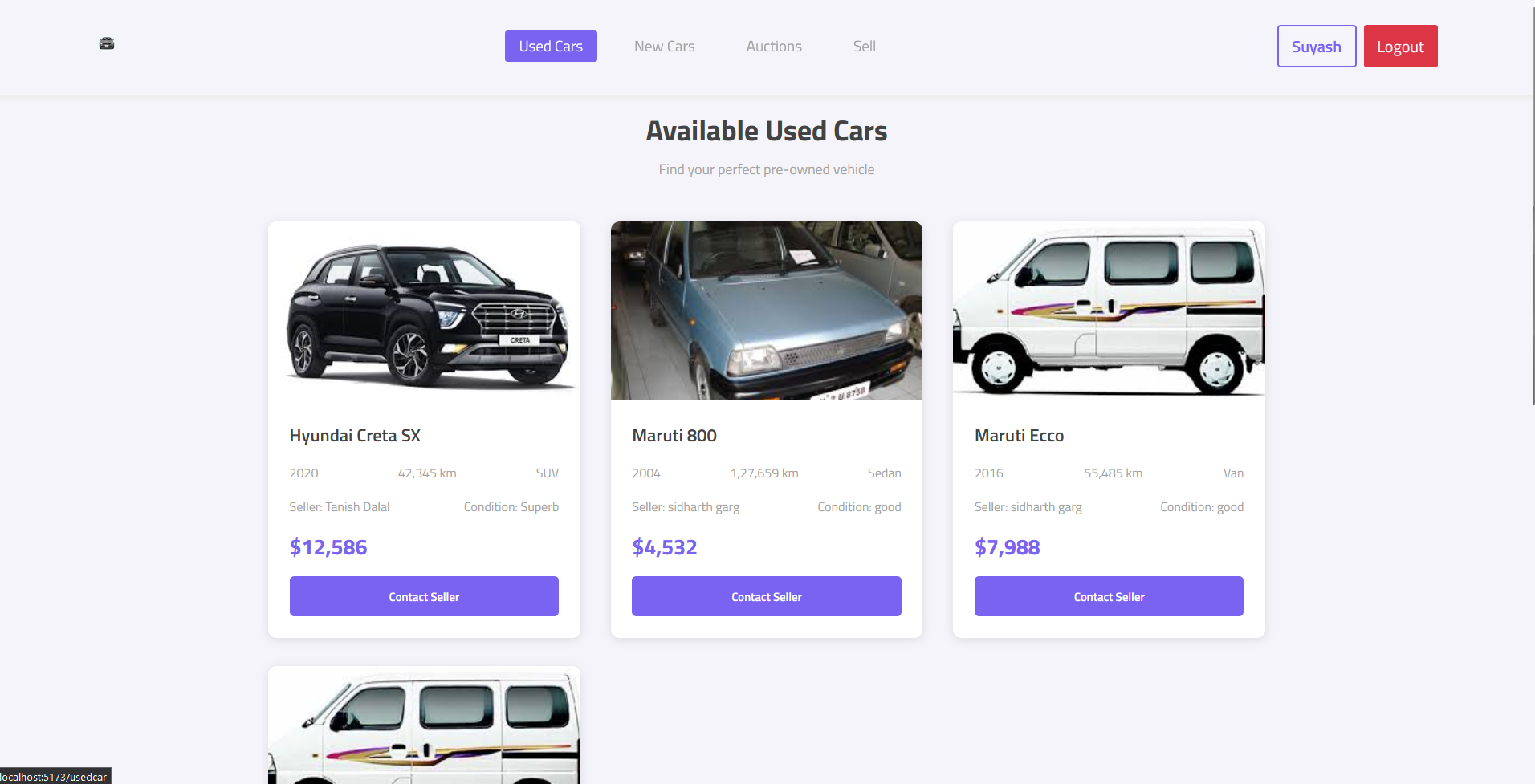


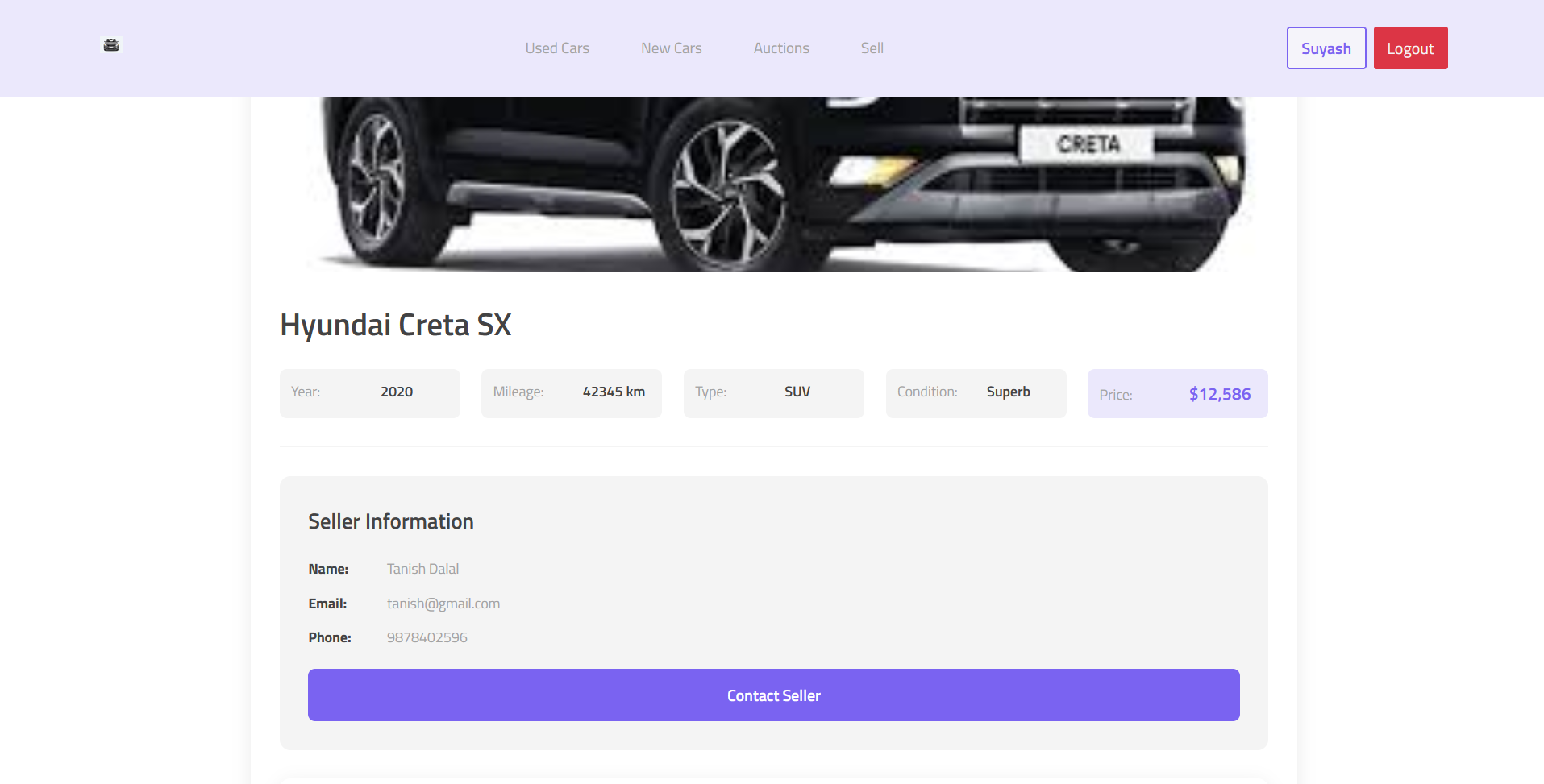


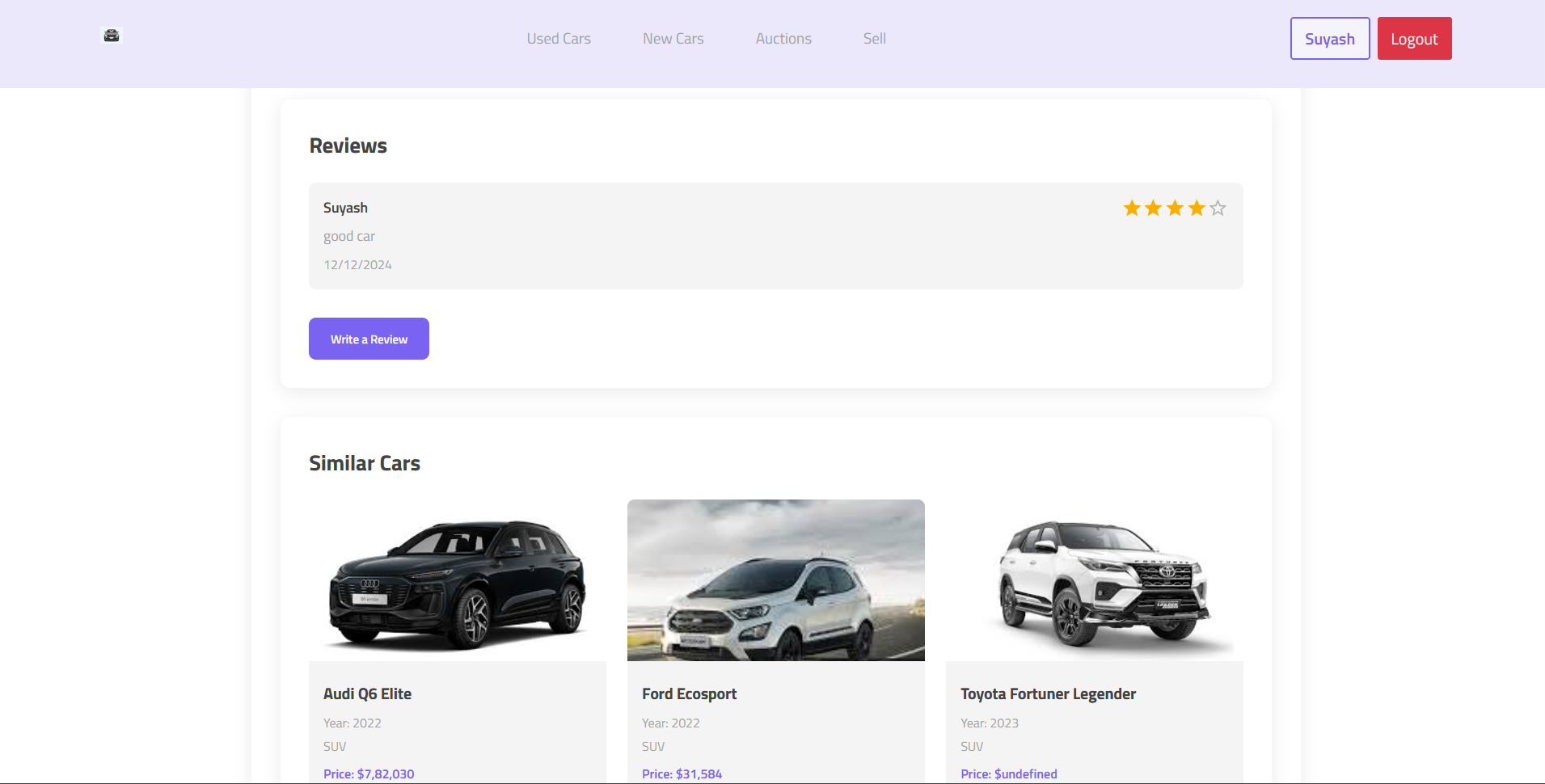












## 

## 6.LIMITATIONS

* **Limited Payment Options:** The platform does not support any payment methods, the user can only contact the dealer directly to confirm his booking and about the car.
* **Vehicle Availability Issues:** Real-time inventory updates might not always be accurate, leading to situations where cars may be unavailable despite appearing to be available online.
* **Geographical Constraints:** The service may only be available in specific locations, limiting the reach to customers in other areas or countries.
* **Dependence on Internet Connectivity:** As an online platform, the website’s functionality is dependent on stable internet connectivity, making it inaccessible during outages or poor network conditions.
* **Scalability Concerns**: If the platform grows rapidly, there might be challenges with handling high traffic volumes, requiring continuous optimization of database and server infrastructure.

## 7.CONCLUSION

The car buying website aims to provide a comprehensive and user-friendly platform for exploring, comparing, and purchasing vehicles. By integrating an advanced database, interactive search features, and a streamlined purchasing process, the platform creates a transparent and efficient marketplace for both buyers and sellers. The website offers detailed vehicle information, comprehensive comparison tools, and multiple communication channels to support customers throughout their car buying journey.

While facing challenges such as varying inventory availability and potential regional limitations, the platform establishes a robust framework for automotive transactions. The intuitive design enables users to navigate through extensive vehicle listings, access detailed specifications, and connect with sellers seamlessly. Advanced filtering options, financing calculators, and transparent pricing mechanisms empower customers to make informed purchasing decisions.

Looking forward, the website is positioned to continuously enhance its capabilities through technological innovations, expanded vehicle databases, and improved user experience features. By adapting to emerging market trends and customer preferences, the platform can solidify its competitive edge in the dynamic automotive e-commerce landscape. The ongoing commitment to user-centric design and technological advancement ensures the website remains a reliable and attractive destination for potential car buyers.

## 8.FUTURE SCOPE

* **Expanded Payment Options**: Integrating more payment gateways, including digital wallets and international payment systems, to cater to a wider customer base.
* **AI-Driven Recommendations:** Implementing machine learning algorithms to offer personalized vehicle suggestions based on user preferences, past bookings, and behaviour.
* **Mobile Application Development:** Creating a mobile app for iOS and Android platforms to provide users with a more convenient, on-the-go experience for booking and managing rentals.
* **Global Expansion:** Extending the service to more regions and countries, allowing users worldwide to access the platform and book cars from various locations.
* **Advanced Analytics and Reporting:** Incorporating advanced analytics to provide businesses with deeper insights into booking trends, customer behaviour, and fleet management, aiding in decision-making and operational optimization.

## 9.REFERENCES

1. MongoDB Documentation: MongoDB, Inc.<https://www.mongodb.com/docs/>
2. Express.js Documentation: Express.js.<https://expressjs.com/>
3. React Documentation: React.<https://reactjs.org/docs/getting-started.html>
4. Node.js Documentation: Node.js.<https://nodejs.org/en/docs/>