

Car Rental System:

Project Summary

We plan to develop a car rental service. Customers around the US will be able to look up available cars to rent based on their location, fuel type, vehicle type, fare per day, model and more. They can also search by preferred model choice and even search for cheaper options available in the area which closely matches this preferred model. Customers will also be able to look at past customers ratings and give their own ratings to the cars that they have rented.

As the car rental admin, we would be able to easily handle people's data including how many days they rent the car for and based on the fact that if the person who is trying to rent the car has a criminal history or is below 25, we may have to increase the insurance on the car. Along with this, if the car is being rented in a district with a high auto-theft rate then the admin could charge a higher fixed deposit on the rental. Admins will also have the ability to change all specifications and details regarding the car.

Application Description

We want to solve the problem of finding rental cars in the USA. We plan to make the process better for both the rental company and the customers

With this car rental application, our main purpose is to make the lookup of cars for customers easier and assist them with booking the rental service. They can easily look up their preferences and even solidify their opinion using past people's reviews. Customers would be easily able to decide on a particular car or even look for more economical options.

On the other side, for the car rental admin, we want to make it easy for them to also access the day-to-day rental data and make decisions based on the past data. We plan to separate the application between customer side and admin side so that both can use it in parallel and make the process smooth.

Creative Component

We will add a map based search feature on our website. Prospective customers can search for cars available by clicking on available cities on the map. This will allow for available cars to be listed, giving the user the ability to view all sorts of information, such as insurance rates, general rental rates, and various car makes/models. Along with this, the major use of our map system would be that users could search their preferred car model and then see which cities that model is available in too. (Interesting for car enthusiasts!!!!) Along with this we could add a feature where customers could list the cities they want to visit and then we could provide them if their preferred model is available in all the locations.

We consider this as an interactive visualization that we plan to add to make it an interactive experience for the user to select the cities that they would want to rent a car in. This is important as it will give users the ability to freely explore the cars and rates that we offer. They can also use this to plan their intercity trips too.

Usefulness:

The usefulness of our application will come from the functionality of vehicle rental. Customers will have the ability to view different models and pricing options for the vehicles that we offer. We'll do this through using real-world datasets that allow us to adjust insurance rates and pricing models according to a variety of factors like crime rates and location information. The basis of the application will be offering users the ability to rent these cars with all of this information on hand, which they can freely view as well in order to make their own judgements and comparisons between features like car make/model, insurance rates, and general rental prices.

A similar application to ours is Zipcar, which also allows for car rentals. However, we intend to differentiate ourselves through a greater use of real-world datasets, such as car theft trends, which will allow us to dynamically calculate the rental price of various cars - a more optimized feature. Additionally, we'll offer a functionality to view a map that lists all of the various car locations throughout the whole of the United States, which will inform users on how car rental prices in one area compare with another.

Realness

We will be using datasets from Kaggle. Specifically, the Car Theft State-by-State (With Makes and Models) dataset (<https://www.kaggle.com/datasets/thedevastator/uncovering-state-by-state-car-theft-trends-in-20>), as well as the Car Rental dataset (<https://www.kaggle.com/datasets/kushleshkumar/cornell-car-rental-dataset>). Both datasets are CSVs.

With each of these datasets, we hope to build out our databases through the information given from the datasets. In the Car Theft dataset, there are 510 entries (cardinality of 510) from varying states about information on car thefts. The information consists of which state it occurred in, the magnitude of the car theft (listed as rank), the make/model of the car, the model year of the car, and the number of thefts of the car in that state, which gives us a degree of 6 when including the entries. For the Car Rental dataset, there are 15.8 million unique owner IDs (cardinality of 15.8 million), of which detail what fuel type the rental car was, the satisfying rating of the customer, the number of renter trips taken with the car, the number of reviews for the owner, the city of where the car was rented, the country of where the car was rented, the latitude of the location of the rental, the longitude of the location of the rental, and the state of where it was rented at, along with the owner id, which gives us a degree of 11.

We aim to use the Car Theft dataset in order to determine rates of insurance by location, as areas with more car theft are more likely to see insurance actually come into play. We'll leverage the locations of where thefts are more prominent, and factor this information accordingly to determine the insurance rates of the area. We will also use the data from the Car Rental dataset in order to determine rates and other general information for our car rental service.

Detailed Functionality:

Our application will provide two primary functionalities for customers and administrators respectively. The following is a breakdown of the key features including how they fulfill Create, Read, Update, and Delete (CRUD) operations, as well as more advanced or “creative” functionalities that enhance the system.

Customer-side Functionality

User Registration and Profile Management

- **Create:** Customers can register a new account by providing basic information (name, email, password, etc.).
- **Read:** Users can view their profile information, reservation history, and saved preferences.
- **Update:** Customers can update personal details (e.g., change their email or password) and manage account settings.
- **Delete:** Users can delete their account if they choose to no longer use the service.

Car Search and Filtering

- Customers can search for available cars by various filters such as location (state, city), vehicle type (SUV, Sedan, etc.), fuel type, fare per day, and model preference.
- A refined search option can suggest cheaper alternatives if a user's preferred model is not available or exceeds their budget.

Booking and Reservation Management

- **Create:** Users can create new reservations by selecting dates, picking the desired car, and confirming payment details.
- **Read:** A user can view a list of all upcoming or past reservations in their account.
- **Update:** If needed, customers can modify their reservation details (dates or selected car) before a cut-off time.
- **Delete:** Users can cancel a reservation (subject to policy), removing it from their bookings.

Ratings and Reviews

- **Create:** After completing a rental, customers can leave a rating and review of the car and overall experience.

- **Read:** Potential renters can read past ratings and reviews to make an informed decision.

Administrator-Side Functionality

Car Inventory Management

- **Create:** Admins can add new car listings to the database, providing details such as model, fuel type, location, and base fare.
- **Read:** Administrators can view the entire car catalog, reservation logs, and usage statistics in an internal dashboard.
- **Update:** Admins can update car details (e.g., rental price, availability status) or adjust information (e.g., if a car is under maintenance).
- **Delete:** Vehicle listings no longer in service can be removed from the system to prevent user bookings.

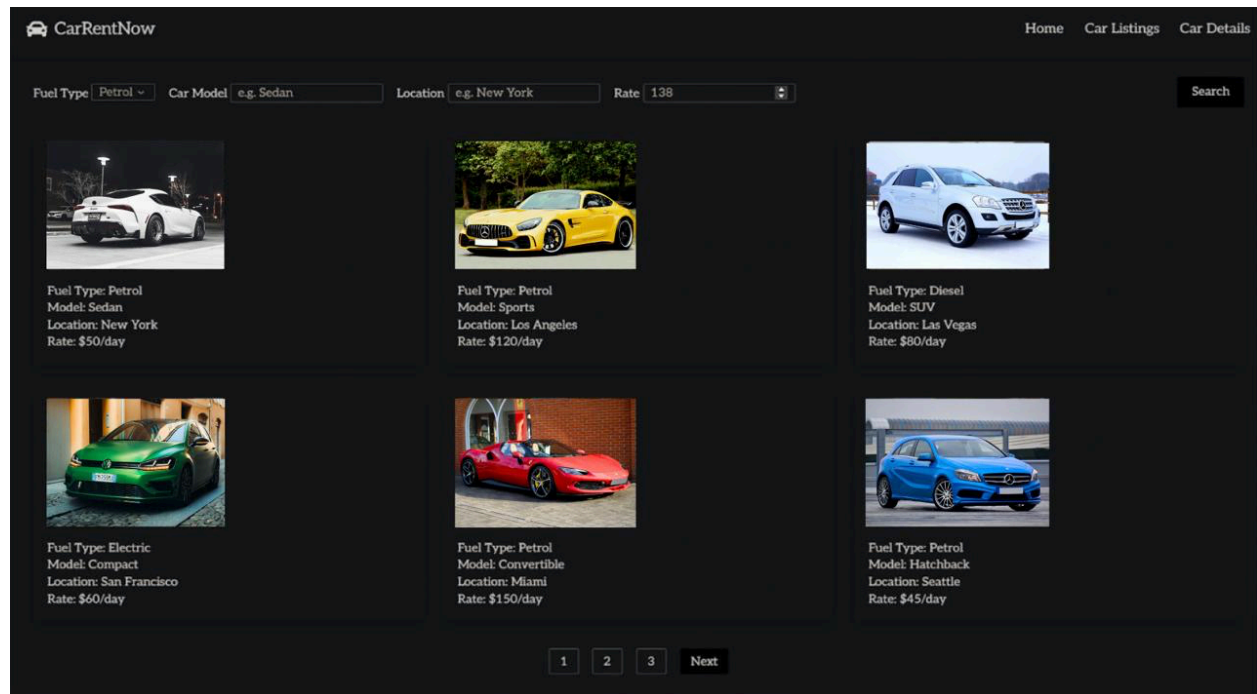
User Management (CRUD)

- **Read:** Admins can look up user profiles, rental histories, and any flags or additional notes (e.g., criminal records or under-25 age).
- **Update:** Admins can impose additional fees or insurance for high-risk renters (under 25 or with past record), ensuring safe and compliant rentals.

Deposit Adjustment

- Drawing on the Car Theft State-by-State dataset, our system calculates the insurance or deposit rates based on theft rates in the area. If a user rents in a region with a high theft rate, the insurance cost or initial deposit is automatically adjusted. This uses real crime/theft data to provide a more accurate rental cost estimate.

UI Mockup



For our broad UI that doesn't include the map, we will have a variety of options for cars. It will list the rates, models of the car, and locations for each car. There will also be a way to specify the search such as preferred car model, rate, location, and fuel type.

Project Work Distribution

Team Members:

- Aryan Chadha- Team Captain, Backend
- Dhruv Malpani - Backend, Database integration
- Joey Zhou - Backend
- Jun Lim - Frontend, frontend-backend integration

Backend Systems Distribution

Jun will be responsible for the majority of the frontend, where we will display our cars and other key information, such as the map of all cars with available prices and insurance rates, allowing for users to make comparisons and judgements. This will

also entail using APIs to access information and ensuring that the frontend-backend workflow is fitting.

Aryan and Joey will work on the database management side. They are responsible to add and manage the tables and make sure that all the stored procedures and transactional queries are working properly. Additionally, as team captain, Aryan will ensure that all members of the team are keeping up to date with the work and the tasks that are assigned to them.

Dhruv will be responsible for creating the API for organizing and querying the database using predefined procedures created by Aryan and Joey. Will also create endpoints for sending information to the frontend.