# Features to Add

## 1. Seller Features

## a. Add a Car to the Database

* Feature Idea: Allow sellers to list their cars for sale by filling out a form.
* Implementation:

Create a new route (`/sell`) with a form to input car details (e.g., brand, model, price, mileage, etc.).

Save the input data into the existing dataset (`data2`) or a new database table for seller listings.

Validate inputs to ensure no invalid or incomplete data is added.

## b. Edit or Remove Listed Cars

* Feature Idea: Provide sellers the ability to update or delete their car listings.
* Implementation:

Use a unique identifier (like a car ID or seller email) to allow sellers to manage their listings.

Create routes like `/edit/<car\_id>` and `/delete/<car\_id>` to handle these operations.

## c. Seller Dashboard

* Feature Idea: A personalized dashboard where sellers can view their listed cars and the number of buyer inquiries for each car.
* Implementation:

Store seller details in a separate table (e.g., seller name, contact info, car ID, etc.).

Use a query to fetch listings associated with a specific seller.

## 2. Buyer Features

## a. View All Available Cars

* Feature Idea: Create a dedicated page for buyers to browse all available cars (similar to an online catalog).
* Implementation:

Create a route (`/browse`) that displays all cars in the database.

Allow sorting (e.g., by price, mileage) and filtering (e.g., by location, brand, fuel type).

## b. Inquiry or Contact Seller

* Feature Idea: Allow buyers to contact sellers directly for more details or negotiations.
* Implementation:

Add a "Contact Seller" button on each car listing.

Use a form to send a message to the seller, storing the message in a database or sending it via email.

## c. Wishlist or Save Cars

* Feature Idea: Allow buyers to save cars they’re interested in for later reference.
* Implementation:

Use a session or database to store buyerselected cars.

Display the wishlist in the buyer's dashboard.

## 3. User Accounts and Authentication

* Feature Idea: Introduce accounts for both sellers and buyers.
* Implementation:

Use a library like FlaskLogin or FlaskSecurity for user authentication.

Create separate dashboards for sellers and buyers.

Use user roles (`seller`, `buyer`) to restrict access to certain features (e.g., only sellers can list cars).

## 4. Enhanced Recommendation System

## a. Buyer Preferences

* Allow buyers to save their preferences (e.g., favorite brand, price range) for automatic filtering of recommendations.
* Provide notifications when a new car matches their saved preferences.

## b. Seller Insights

* Show sellers data like average prices for similar cars to help them set competitive prices.

## 5. Advanced Filtering Options

## a. Detailed Search Filters for Buyers

* Add filters for more granular options, like:

Car color

Model year

Number of previous owners

## b. LocationBased Recommendations

* Integrate locationbased filtering using Google Maps API or similar services.
* Show nearby cars to the buyer based on their current location.

## 6. Analytics and Insights

## a. Price Trends

* Create visualizations to show price trends for specific brands or models.
* Let sellers and buyers see the market trends.

## b. Popular Cars

* Display the most viewed or inquired cars on the homepage.

## 7. Payment Integration

* Allow sellers to mark cars as sold or enable a direct payment feature using platforms like Stripe or PayPal.
* Enable buyers to pay a small reservation fee online to hold a car.

## 8. Notifications and Alerts

## a. Email Notifications

* Notify buyers when a car matching their preferences is listed.
* Notify sellers when a buyer inquires about their car.

## b. RealTime Alerts

* Use WebSocket or similar technologies to provide realtime alerts to buyers and sellers.

## 9. Reviews and Ratings

* Allow buyers to leave reviews for sellers.
* Display average ratings for sellers to build trust.

## 10. Integration with External APIs

* Car Valuation API: Provide price estimates for cars based on features.
* Car History API: Show buyers a report of the car’s accident history or maintenance records.

## Database Considerations

To support these new features:

1. 1. User Table: To store buyer and seller details (name, email, password, etc.).
2. 2. Car Table: For listing details (car features, seller ID, status).

3. Inquiry Table: To store buyerseller interactions.

## UI Enhancements

* Improve the homepage with clear navigation for "Buy a Car" and "Sell a Car."
* Use frameworks like Bootstrap for a responsive and modern design.

## Scalability

To handle more users and cars:

1. 1. Use a proper database (e.g., MySQL, PostgreSQL) instead of a flat CSV file.
2. 2. Consider deploying the app on a cloud service like AWS, Azure, or Heroku.