

AI Car Consultant Requirements

1. Product Documentation

1a. Functional Requirements

➤ Photo Upload & Analysis

- Users must be able to upload up to **55 high-resolution photos** of their vehicle.
- The AI agent must process these photos to detect:
 - Visible damage (scratches, dents, missing parts)
 - Modifications (e.g., aftermarket rims, spoilers)
 - General vehicle condition (e.g., cleanliness, tire wear)

➤ Interactive Questionnaire

- Users must answer a structured questionnaire about their car (e.g., year, make, model, modifications, mechanical issues).
- This data will be combined with image and document analysis to improve AI recommendation accuracy.

➤ Document Upload

- Users must be able to upload a **CarFax report or similar PDF file**.
- The AI agent must extract key information such as:
 - Ownership history
 - Accident records
 - Service records
 - Title status

➤ Market Data Integration

- The system must reference current and historical auction data from sources like **Cars & Bids** and **Bring a Trailer**.
- AI should analyze comps to provide tailored pricing, listing, and presentation advice.

➤ Real-Time Listing Advice

- After input is provided, the user must receive a **comprehensive, easy-to-read report** with:
 - Listing score (0–100)
 - Specific action steps (e.g., change title, retake photo, reword description)
 - Recommended price range
 - Market positioning of the vehicle

➤ User Interface Requirements

- The UI must be:
 - Modern and minimalistic
 - Accessible and intuitive for users of all ages and technical levels
 - Fast and responsive, with loading times <2s for main actions

➤ Payment Subscription Plans

- Applications Must Have a Free and Pro subscription
 - Free Subscription will allow the user to create a report On One vehicle.
 - Pro Subscription will allow the user access to all tools and allow to create reports and as many vehicles as the user wants
 - Users must be able to **upgrade** or **downgrade** their subscription at any time via a secure payment interface (e.g., Stripe or PayPal).
 - The system must prevent users on the Free plan from creating more than one report, displaying a prompt to upgrade when limits are reached.

❖ Non-Functional Requirements

➤ Deployment & Accessibility

- The application **must be available as a mobile app** on both the **Apple App Store** and **Google Play Store** for maximum accessibility.
- The application **must also have a responsive web interface**, accessible via major browsers (Chrome, Safari, Firefox, Edge).
- The **mobile experience should be prioritized** in design and performance, ensuring key features are accessible and performant on devices with limited screen size or bandwidth.
- The mobile and web versions must maintain a **consistent UI/UX**, allowing seamless user transitions between platforms

➤ Handle all secure User data in a Secure database

- All user data, including personal information, subscription status, and vehicle reports, **must be stored in a secure database with encryption at rest and in transit**.
- The application must implement **HTTPS** for all network communications to ensure end-to-end encryption.
- Sensitive data (e.g., authentication tokens, payment info) must **never be logged** or exposed in client-side code.
The system should follow **OWASP Top 10 security best practices**, including input validation, secure authentication, and protection against common vulnerabilities (e.g., SQL injection, XSS).
- If using third-party services (e.g., Firebase, Supabase), the app must configure **access rules and roles** to enforce proper data isolation and prevent unauthorized access.

➤ Scalability Requirements

- The application must support at least 10,000 active users with no degradation in performance.
- The system must be horizontally scalable using containerization (e.g., Docker with Kubernetes or Vercel Serverless Functions).

- Maintainability & Logging
 - The backend must be modular and documented for maintainability.
 - Application errors must be logged using a centralized logging solution (e.g., LogRocket, Sentry, or Firebase Analytics).

- User Roles & Permissions
 - Add support for:
 - Anonymous users
 - Logged-in Free users
 - Pro users (paid)
 - Admin users (internal dashboard)

❖ User Flow Diagram

- User opens the Mobile application and gets greeted with the Logo of the application in an Animation.
- If it's the user's first time, then the user will be allowed to sign up Via Google and create their account
- When logged in, the user will be allowed to either upload photos of their car or the application will access the phone camera to take photos of the car, allowing a limit of 55 photos taken or uploaded
- Once photos are taken, the AI Agent will ask a series of questions like VIN, car fax, and other information about the car, like if there are any modifications
- Once all the information is submitted, the AI agent will generate a report with a score, and if the user has a free plan, they can see their score and a summary, and explain where they can do better on their listing based on market research; however, the pro subscription will give out all the information and pro user can also ask the AI agent questions to give it more insight

Set Up / Sprints and Frameworks

❖ FrameWork

- Frontend - Next.js + TailWind CSS
- Mobile Development - Expo + React Native
- BackEnd Database - Google Firebase
- Image Upload - Firebase
- Deployment - Docker / Expo

Renvo.Ai



