

FindACarFor.Me

Phase II Technical Report

Trent Ho, Varun Jawarani, Jatin Kulkarni, James Stuedemann, Taqi Hossain

October 24th, 2022

1 Motivation

While listing aggregators are already a consistent element of purchasing a vehicle—and while they provide thorough specifications on each make and model in the listing—there exists an inherent bias in a for-profit aggregator that FindACarFor.Me attempts to remove, especially with its dual focus on frugality and safety. Our product will be that comprehensive source for car buyers through its: location-specific listings and fuel stations, in-depth safety comparison for vehicles, and composite cost-to-purchase overview. Our intended (but not entire) audience are users with questions like the following:

1. What are listings near me that fulfill my criteria (make, model, safety, etc.)
 - How much will I spend on fuel if I drive x miles weekly?
2. Is my surrounding area properly equipped for my purchase of an electric vehicle?
 - What types of fuel are less expensive in my area?
3. What is the safest and most budget-friendly car for my son?
 - Where can I purchase affordable vehicles nearby?

2 User Stories

Phase I User Stories:

1. Safety Information
 - *Story: I am the parent of a 16-year-old girl who is about to get her driver's license. I am very nervous about her driving, as car accidents are one of the leading causes of death in teenagers. I would love if your website had information about safety ratings for different cars, and if I could sort the list to see the safest cars.*

- We have hard-coded the Safety Data into our model page, but we will need to add data sourced directly from the API and add functions to sort.
- Estimated: 20 minutes
- Actual: Our sort function is incomplete, but the safety ratings are visible.

2. Pricing

- *Story: Hi, I am a full time student who works a minimum wage job. I need to buy a car to drive to work, but I can barely afford my tuition as it is. I would like to be able to see a list of the cheapest cars. Gas is also expensive, so I would like to be able to see the gas stations around me and be able to compare their prices.*
- As of now, we are pulling information on price and in the process of paginating, so while filtering isn't available, price comparison can still be done.
- Estimated: 2 hours
- Actual: 3 hours

3. Charging Stations

- *Story: Hi, I recently bought a Tesla Model X. I love this car, but I have been nervous to drive it because I am worried I will run out of charge and not be able to find a charging station. It would be very handy to be able to search charging stations by their location, so I can plan my trips and make sure I'm never stranded without a way to charge my car.*
- We are currently in the process of implementing the Google Places API to add fuel station locations by location, but as of now we have hard-coded the information to be based around Austin, TX.
- Estimated: 2.5 hours
- Actual: 4-5 hours due to Google Cloud Console.

4. Car Listings

- *Story: Hello, I have been trying to buy a Honda Civic for some time now, but I can't seem to find any dealership that has them in stock. It would be great if your website had a list of car listings I could search through to find a Civic for sale. Thanks!*

- We have hard-coded the makes and models for the vehicles for now, but we will be introducing a search function in the next phase.
- Estimated: 1 hour
- Actual: Our APIs are static as of now, so we aren't searching through the API—results will be limited to the three hard-coded cards.

5. Electric vs Gas

- *Story: Hi, I want to buy a car, but I'm torn on if an electric or gas-powered car is better for my situation. I would love to be able to compare the attributes of different electric and gas-powered cars, so I can make an informed decision.*
- With regard to electric and gas-powered cars, our website will give users the ability to compare local gas prices with local charging prices, and the amount and distance of gas/charging stations relative to the user's chosen location. These features are currently out of the scope of this phase, but they will be implemented in future phases.
- Estimated: 20 minutes
- Actual: We have partially implemented this feature, but not at a level where it can be compared side-by-side. We foresee that being much later.

Phase II User Stories:

6. Work Car Concerns

- *Story: Hi, I am buying a car for daily life. I mainly use it to work, and I do not travel often. I wonder what kind of cars would suit me better. Based on my daily usage, I would prefer a vehicle with lower fuel costs and more convenience to fix if there's something wrong. Would your website provide information on that?*
- We will have fuel costs in your area for each fuel station, and a miles-per-gallon specification for each listing, so you'll be able to find a car that's affordable for commuting to and from work - especially with our future commute cost calculator. As for reliability, we'll provide recalls and complaints for you to browse on your new car.
- Estimated: 4 hours

- Actual: The commute calculator will take longer, but fuel and reliability information is already there.

7. ADAS

- *Story: Hi, I am a new driver and this will be my first car. I've learned that ADAS is pretty important. I wonder if your website is going to be providing any information on that?*
- We are in the process of displaying our stored information on whether a model has Stability Control, Front-end Collision Warnings, and other standard ADAS features, all of which we have pulled from NHTSA
- Estimated: 15 minutes
- Actual: 5 minutes - it was already present in our database.

8. New Customer Service

- *Story: Hi, I am completely new to the car-buying process. Is your website going to provide customer service or any crash course on car buying?*
- This will not happen. Our website doesn't support this functionality, and leading people on their decisions to purchase would only exaggerate the bias in car purchasing that we are trying to remove.
- Estimated: N/A
- Actual: N/A

9. Historical Car Prices

- *Story: Due to the pandemic, I figure car price has been fluctuating a lot. Would your website be providing the car price during the past based on different dealers as well? Also, I think a model to predict the future car price will help the customer to make an easier decision.*
- Price will be one of the main attributes we display for cars, but providing past and future prices are out of the scope of our website. We will eventually provide features to allow sorting by prices, which will help users compare cars for their current needs in the present.
- Estimated: 3-4 hours for filtering and sorting

- Actual: N/A - filtering and sorting will take precedence over historical car prices, which we do not have an API to even access.

10. Ranking by Attributes

- *Story: Hi, I wonder if your website is going to be providing a ranking based on horsepower, capacity, and swept volume?*
- Hi. Unfortunately, sorting and ranking features are currently out of the scope of this phase. We will be implementing sorting features in a future phase based on the attributes we currently display on our site.
- Estimated: 2 hours
- Actual: We have partially implemented this feature, because we have filterable attributes, but we haven't added the ability to filter or sort based on attribute.

3 Models

There are presently three models to compare vehicles with: fuel stations (EV charging and gasoline); car specifications by make, model, and year; and car listings from nearby dealerships. All the models are strongly related, with attributes to filter and sort on. Fuel stations are relevant to a consumer's decision between an EV and a gasoline-powered car, and they also expand on the cost-per-vehicle with fuel prices. Yearly car models are useful for general specifications and price averages, and to compare potential vehicles on a broad scale before seeking individual listings. The listings are often the final barrier for a car buyer, and we aim to clarify the process by supplementing the listings with fuel and vehicle information on the consumer's location.

1. Fuel Stations

- *Filtered Specifically On:*
 - Distance
 - Price
 - Fuels Available: (EV / Gas / Diesel)
 - Rating
 - Gas Station Company (for rewards purposes)
- Phone Number
- Address (<city>, <state>)
- Google Maps Link
- Website
- Opening Hours

2. Car Model Specifications

- *Filtered Specifically On:*
 - Safety Ratings
 - Average MSRP
 - Miles Per Gallon
 - Make / Model / Year
 - Fuel Type
- Vehicle type (Sedan, truck, SUV, crossover, station wagon)
- VIN Number
- FWD/RWD/AWD/4×4
- Car Manufacturer Website

- Model Recall History

3. Car Listings

- *Filtered Specifically On:*
 - Price
 - Mileage
 - Make / Model / Year
 - Fuel Type
 - Location
- Vehicle type (Sedan, truck, SUV, crossover, station wagon)
- Car features
- VIN link to decoder
- General Version of Car
- Dealership information

4 RESTful API

We combed the available free or student-oriented API options for vehicle data, and we decided to pull data from [openchargemap](#), [Google Places](#), [NHTSA](#), [CIS](#), and now we will also be scraping [GasBuddy](#). Using Postman, we created a client-side API for our own project, documenting the paths, variables, and query options for each schema that we implemented—which can be found [here](#). The documentation shows that we have listed endpoints for all data that we are permitted to share, but we have nonetheless created endpoints for the data that we have received through

education licenses. For listings, we offer a multitude of nested details, divided into various sets of information (the parameters: `sort_by`, `features`, `model`, etc.) for both our clarity and the consumers we convert to API users.

5 Tools

For the front-end side of our web application, we employed react-bootstrap for the UI, react-router-dom for the URL subpaths and model pages, and Node.js for general website programming. When testing the front-end, we used two major programs: Selenium (in combination with Python modules *pytest* and *webdriver_manager*) for testing headless instances of the GUI on Chrome; and Jest, a JavaScript test framework that meshes well with npm. As for the back-end, we used Postman to handle our API documentation, and wrote Python tests to verify the database for their innate JSON compatibility. With a special focus on GitLab Boards, our workflow was dependent on the stage that our issues were in. This development strategy allowed us to streamline and prioritize as needed—especially in coordination with Gitflow, a system that kept us safe from issues in merge requests and branching.

For the backend side, we decided to use flask and the related flask-sqlalchemy. This allowed us to create a backend server which can process incoming restful api calls, and route them accordingly. We then used mySQL as the database for our tables. This has easy integration with flask-sqlalchemy, where we can easily load data into the tables, and also query the data out. In order to obtain the data from our outside api's, we created a couple scripts which will make the api requests, and save the resulting concatenated data into json files.

6 Hosting

Our hosting process began with Namecheap, a domain name vendor that provides free .me URLs to students. To deploy our website, we used AWS Amplify. The full-stack solution service comes with a pre-deployed HTTPS service, so it was simple to link our domain securely and ensure we had a TLS/SSL certificate. In order to host both a development branch and a main branch as accessible websites, we created a build configuration in Amplify that deployed our websites from our GitLab commits on the aforementioned branches. In order to host our backend, we used Elastic Beanstalk to setup a dynamic backend server. This include many useful auto scaling features, which given our circumstances may not be the most useful, however seems like is good practice in general. The Elastic Beanstalk includes one EC2 instance running Docker on our backend flask server, and also an AWS RDS instance to hold the mySQL database. We also used AWS Route 53 in order to route our URL to both AWS Amplify and AWS EC2 while maintaining the unique SSL certificates for each respective service.