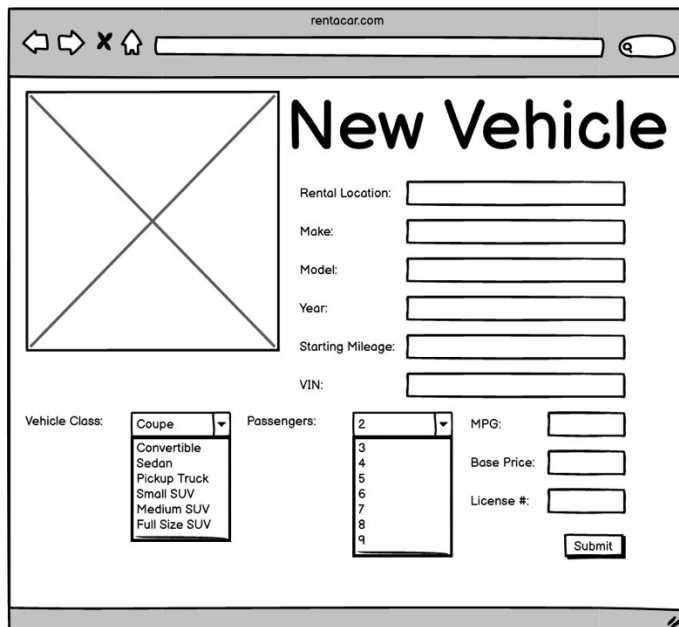
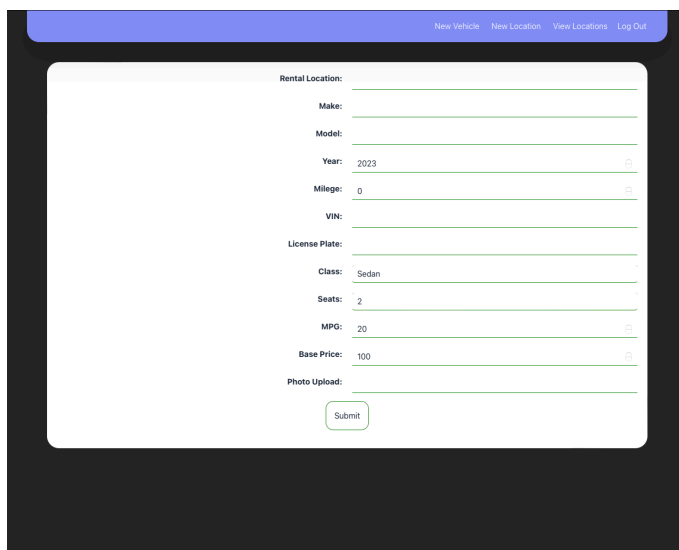


UI UPDATES



The original UI for the 'New Vehicle' form is shown in a browser window with the URL 'rentacar.com'. It features a large placeholder image on the left and a series of input fields on the right. The fields include 'Rental Location', 'Make', 'Model', 'Year', 'Starting Mileage', 'VIN', 'Vehicle Class' (a dropdown menu with options like Coupe, Convertible, Sedan, etc.), 'Passengers' (a dropdown menu with options 2 through 9), 'MPG', 'Base Price', and 'License #'. A 'Submit' button is located at the bottom right.



The updated UI for the 'New Vehicle' form is shown in a browser window. It features a dark blue header with navigation links: 'New Vehicle', 'New Location', 'View Locations', and 'Log Out'. The form itself is a white card with a dark border. It contains the same fields as the original UI, but the 'Rental Location' field is now a text input instead of a dropdown. The 'Year' field is set to '2023', 'Mileage' to '0', 'VIN' to an empty field, 'License Plate' to an empty field, 'Class' to 'Sedan', 'Seats' to '2', 'MPG' to '20', and 'Base Price' to '100'. A 'Photo Upload' field is also present. A 'Submit' button is located at the bottom left of the form.

The New Vehicle screen is very close to the original idea. All fields are still present. I just changed the image field to a textbox, because storing images on the DB is not a great idea.

rentacar.com

New Location

Address 1:

Address 2:

City:

State:

ZIP-Code:

Phone Number:

Vehicle Capacity:

☐ Airport location? ☐ Enter current vehicles?

New Vehicle New Location View Locations Log Out

Street Address:

City:

State:

ZIP Code:

Phone Number:

Capacity:


Store Number:


Airport Location?: ☐

The New location page is very similar to the new vehicle page. The same things had to be changed regarding images.

rentacar.com

Make a Reservation

When will your trip start? 

When will your trip end? 

Where will you pick up the vehicle?

Dropping off at the same place? ☒

Drop off location:

[New Reservation](#) [View Reservations](#) [Log Out](#)

Make a new Reservation

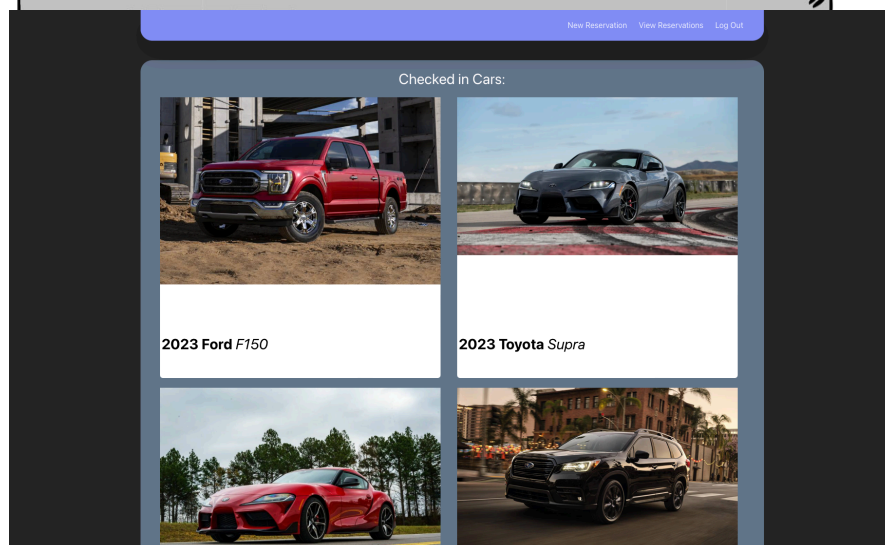
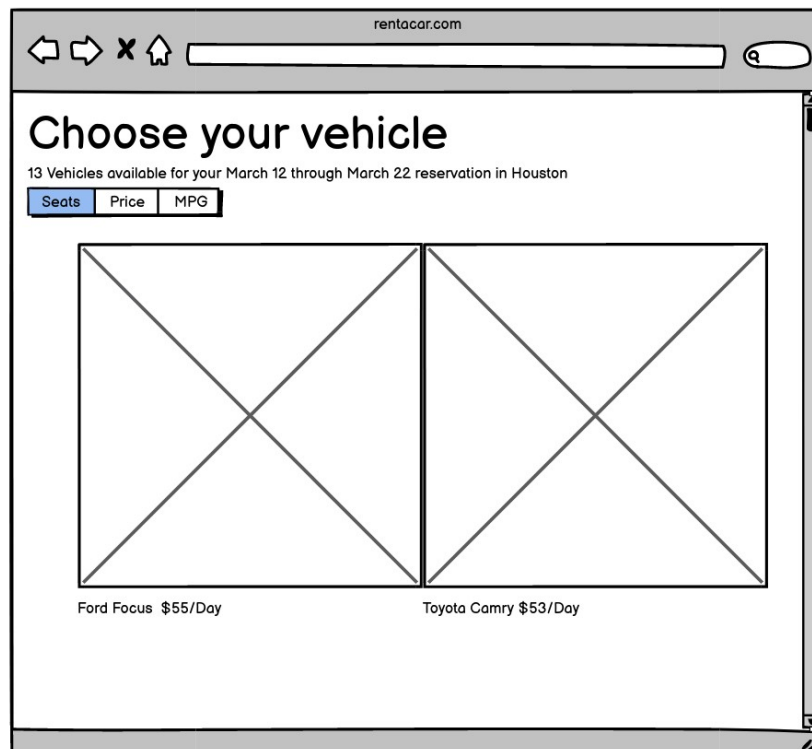
Rental starts on:

Rental ends on: Same as pickup date? ☐

Rental starts at:

Rental ends at: Same as pickup location? ☐

The new Reservation page is very much the same as the initial concept. All the same fields are present. That being said, there are fields for copying both the date and location because maybe there is a one day rental within a single city.



The vehicle selection screen retains much of the same functionality as the wireframe. It has not gotten the filters yet, prices are also hidden within the click through on the vehicle in order to hopefully retain user attention longer.

rentacar.com

License # ABC 123 History

History

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

Add new History logs here if anything of note has happened...

Damage
Service
Police

Submit

New Vehicle New Location View Locations Log Out

License # CRAVE Facts and History:

Make:	Honda
Model:	CR-V
Year:	2017
Base Price:	100
Current Location:	in1

History:

Purchased from user
Lost under lake for 5 years

New history:

Submit

The History update page now shows the vehicle's specs as well as past history items. History is stored as an array of strings on the DB, so types are not necessary

TESTS

Most testing for my application will be done via Puppeteer or some similar front-end manipulation tool. This is because this application does not implement any significantly complex algorithms.

- Test login
 - Should be able to log in to front end as
 - Admin
 - Employee

- Customer
 - Each type will be verified by checking the contents of the navbar
- Test Create and delete items
 - Each type of collection
 - Vehicle
 - User
 - Location
 - Reservation
 - Should get a success message or redirect upon completion
 - This will test the integration with backend via front end
- Test Lifecycle of rental
 - Steps involved
 - New rental (same day)
 - View reservation
 - Add specific vehicle
 - Checkout (no payment)
 - Check back in
 - Update history (possibly)
- Test Authentication
 - If not logged in should redirect to home page
 - If logged in with wrong permissions, should redirect to valid page

Coverage of tests is nearly complete. Being able to run all of these processes should test the front end design, the back end connecting to the front end and modifying data as necessary, and the integration between the two.

In terms of testing nonfunctional requirements and UI requirements, a good strategy would be user testing with friends and family to critique usability as well as functionality.