COMP3385 Project 2

Due Date:

April 27, 2024 at 11:59 PM

For this project we will create a fictional Auto Sales site called United Auto Sales that will allow users to add cars that they want to sell, search for available cars and be able to view some more information on those cars. Users can also add a car to their favourites so they can quickly get back to it.

Database Schema

Cars

Column Name	Data Type
`id`	Integer
`description`	String
`make`	String
`model`	String
`colour`	String
`year`	String
`transmission`	String
`car_type`	String
`price`	Decimal/Float
`photo`	String
`user_id`	Integer

Column Name	Data Type
`created_at`	DateTime
`updated_at`	DateTime

Favourites

Column Name	Data Type
`id`	Integer
`car_id`	Integer
`user_id`	Integer

Users

Column Name	Data Type
`id`	Integer
`email`	String
`password`	String
`name`	String
`location`	String
`biography`	String
`photo`	String
`created_at`	DateTime
`updated_at`	DateTime

Note

Ensure that you define the appropriate models and create your database migrations.

Requirements

You should be able to register for an account on our United Auto Sales web application. Once a user has an account, they should be able to login and see available cars. They can also search by the make or model of the cars. A user also has the ability to submit information for a car being sold as well as upload a photo. Users can also view the full details of a car and also "Favourite" cars that they like. And lastly a user can view their own profile and see the cars that they have added to their favourites.

Part 1

The aim of the first part of the project is to build the API for your United Auto Sales application. This includes creating routes (endpoints) for the user registration and login system as well as the ability to add and view cars, search for cars and favourite cars. See Table 1.

The API routes (endpoints)

HTTP Method	Route	Description
`POST`	`/api/v1/auth/register`	Accepts user information and saves it to the database
`POST`	`/api/v1/auth/login`	Accepts login credentials as email and password
`POST`	`/api/v1/auth/logout`	Logout a user
`GET`	`/api/v1/cars`	Return all cars
`POST`	`/api/v1/cars/`	Used for adding new cars
`GET`	`/api/v1/cars/{car_id}`	Get Details of a specific car
`POST`	`/api/v1/cars/{car_id}/favourite`	Add car to Favourites for logged in user
`GET`	`/api/v1/search`	Search for cars by make or model
`POST`	`/api/v1/users/{user_id}`	Get Details of a user
`POST`	`/api/v1/users/{user_id}/favourites`	Get cars that a user has favourited.

Table 1: API Routes

Test to ensure that your API works by using either the Postman REST Client (http://getpostman.com/) or the Curl command line tool to make requests to your API routes (endpoints).

Part 2

You are required to use VueJS to build the front end of your web application that will interact with the API you built in Part 1. You should also ensure the following is in place:

1. You should use the VueRouter library to create routing for your frontend and implement some Vue components to represent the different pages. See Table 2 below for a list of routes.
2. A User should be able to Register for an account and Login to the website. See Figure 2 and 3.
3. You should generate and send the appropriate Authorization header with each request to your API (except the login and register API routes) using a JWT e.g. `Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ`
4. When a user successfully logs in they should see the last 3 cars that have been added from all users. They should also see a search box, where they can search by the make or model of a car. See Figure 4.
5. A user can click on the "View more details" button/link for a car to see the full details on the car. The "Email Owner" button does not need to do anything for this project but should still be included on the page.
6. A user should however but able to click on the "heart" icon to add the car to their favourites. See Figure 5.
7. A user can view their own profile by clicking 'My Profile' from the menu. See Figure 6. When viewing a users' profile, you should be see the details about the user as well as any cars they have favourited.
8. The user should be able to choose to add a new car for sale to system at which point they will supply all the relevant details for the car and upload an image. See Figure 7.
9. You should display a success message if the user successfully adds a new car or failure otherwise.

Frontend Routes

Route	Description
`/`	Display the homepage of the web application

Route	Description
`/register`	Accepts user information and saves it to the database
`/login`	Accepts login credentials as email and password
`/logout`	Logout a user
`/explore`	View/Explore all posts by all users
`/users/{user_id}`	View user profile info as well as all Posts by that user
`/cars/new`	Allow the user to add a new post
`/cars/{car_id}`	View all the details about a car

Table 2: Frontend Routes

Submission

Submit your code via the "Project 2" link on the VLE. You should submit the following links:

- Github repository URL for your Laravel Exercise e.g. https://github.com/{yourusername}/comp3385-project-2

Appendix

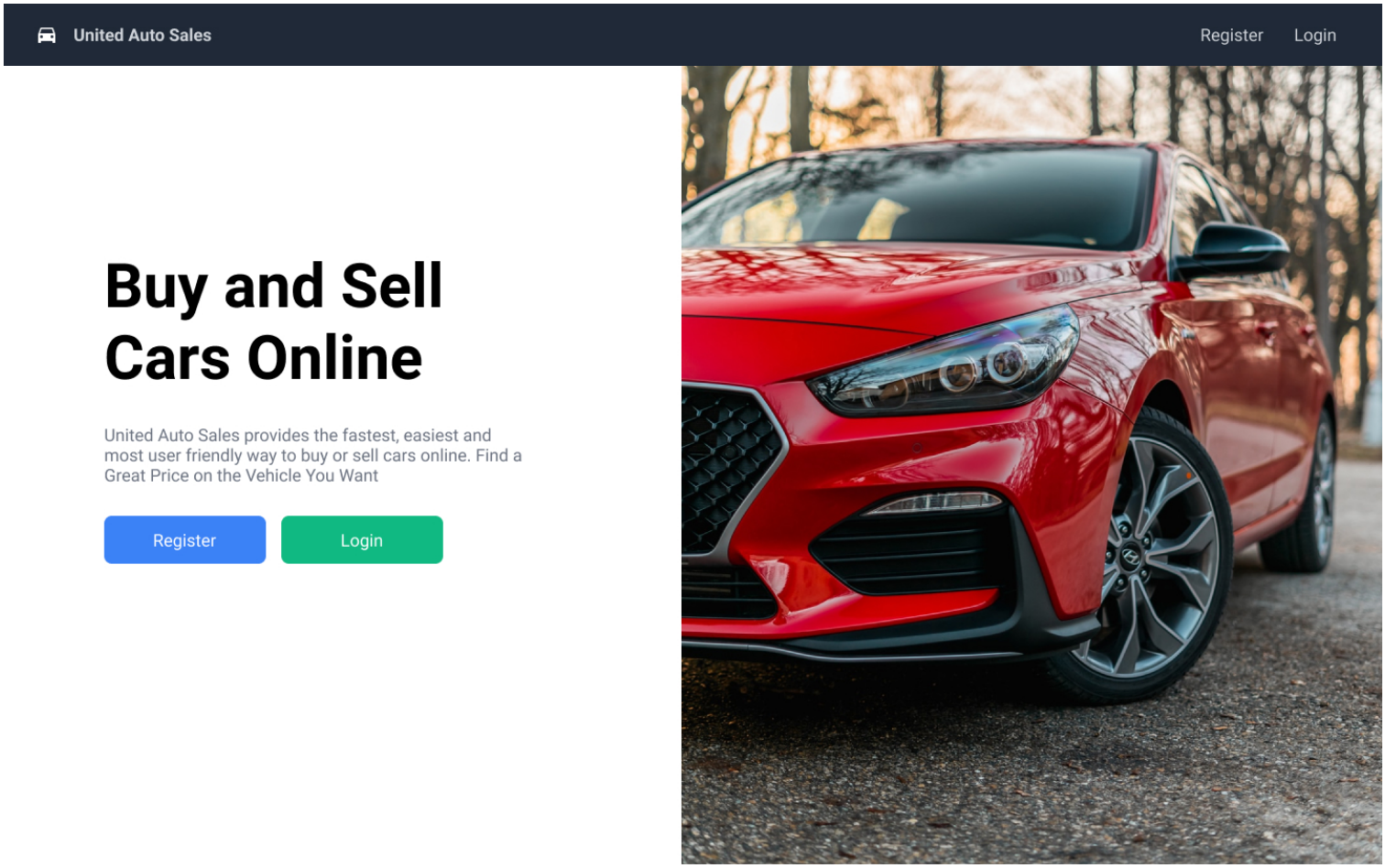

Figure 1. Home page

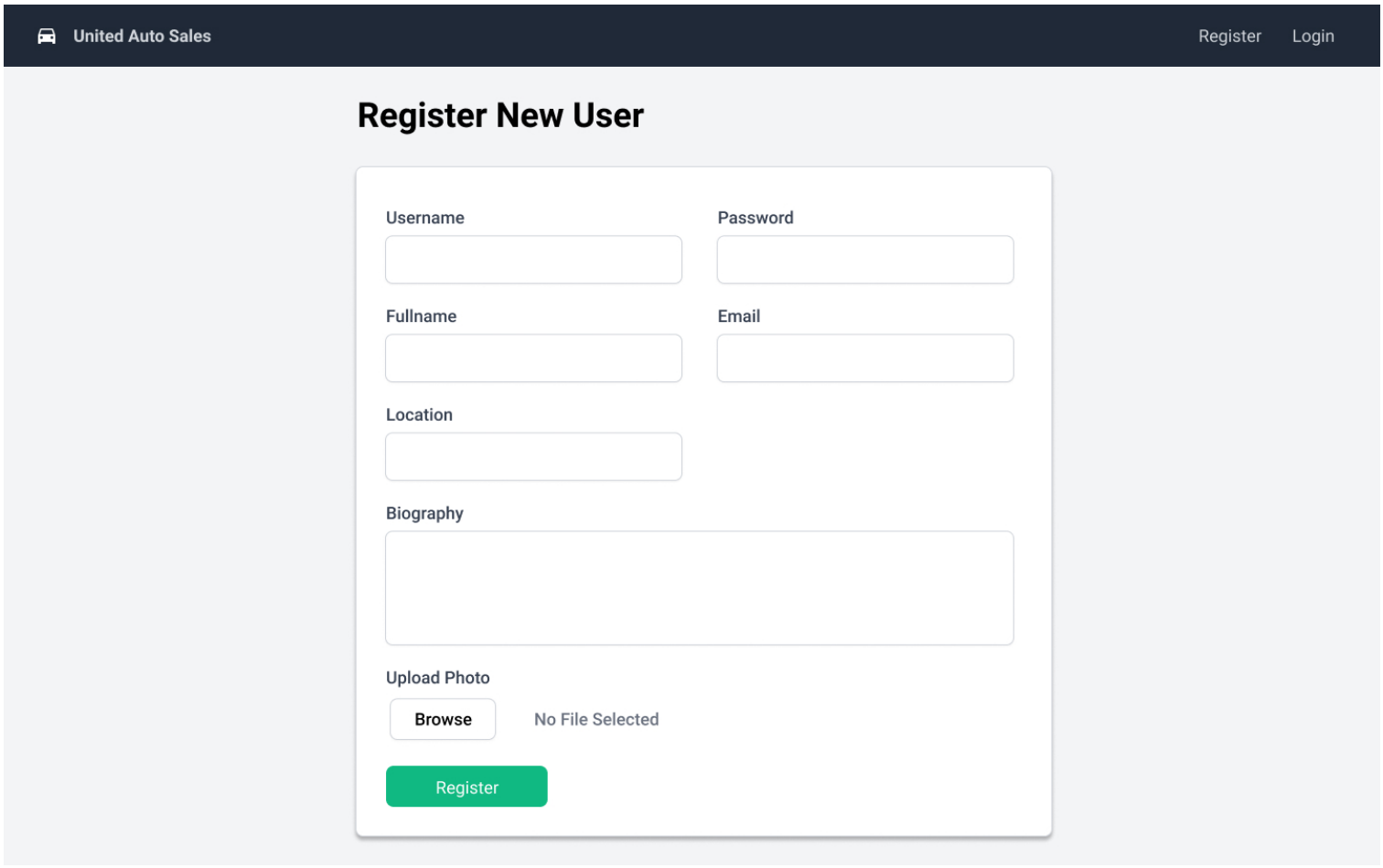

Figure 2. User Registration

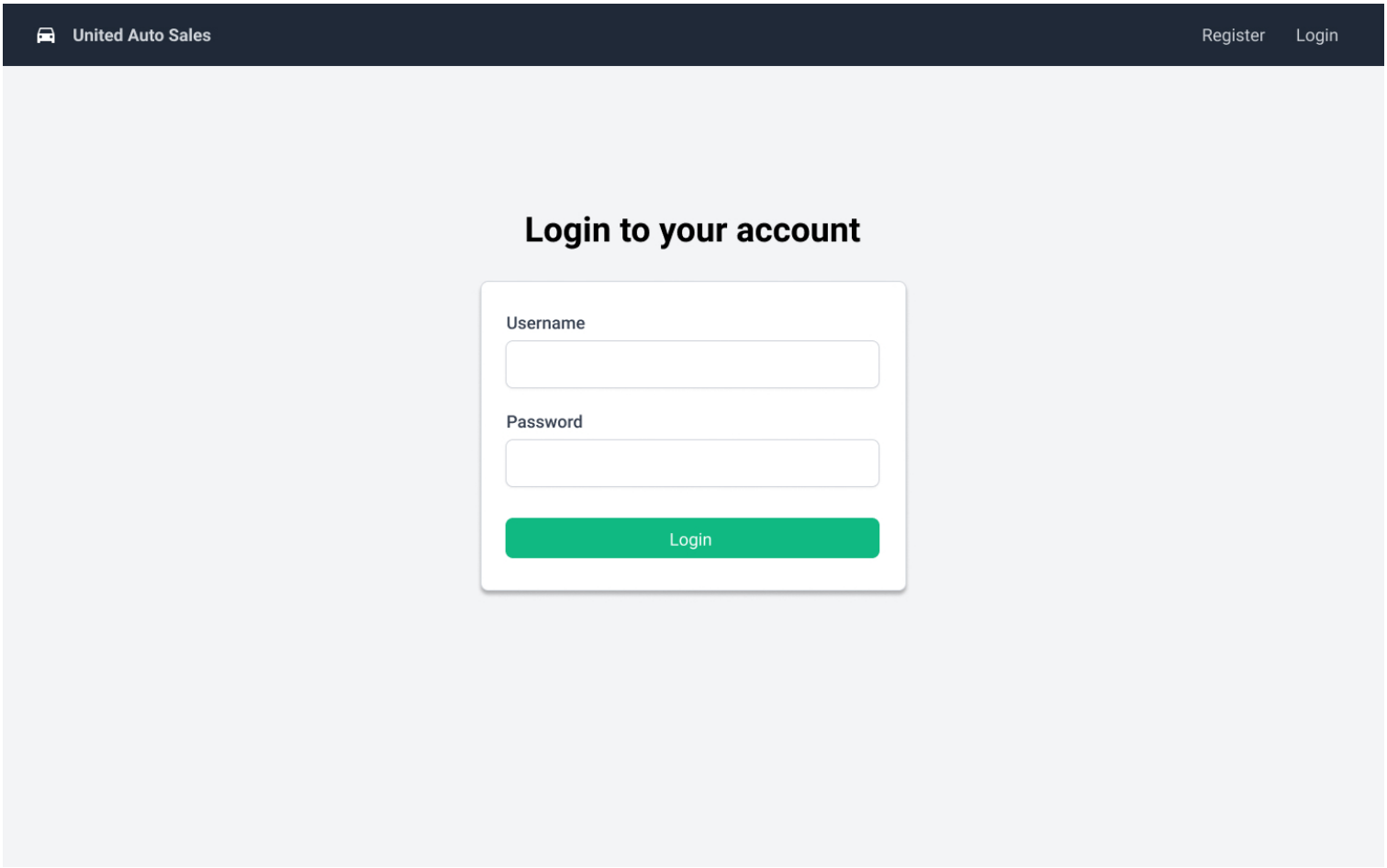

Figure 3. Login

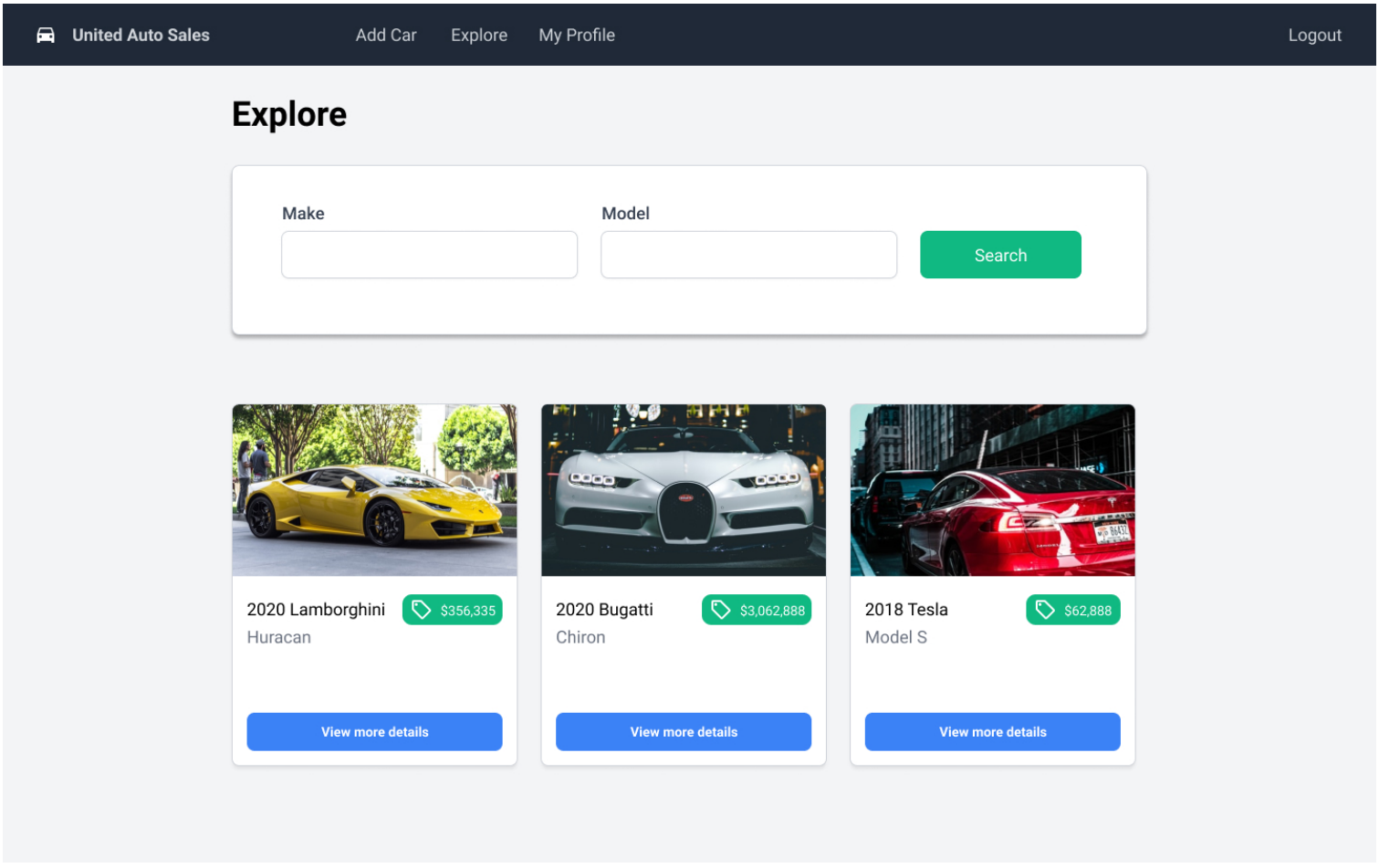

Figure 4: Explore Page. This displays Cars by all users and Search

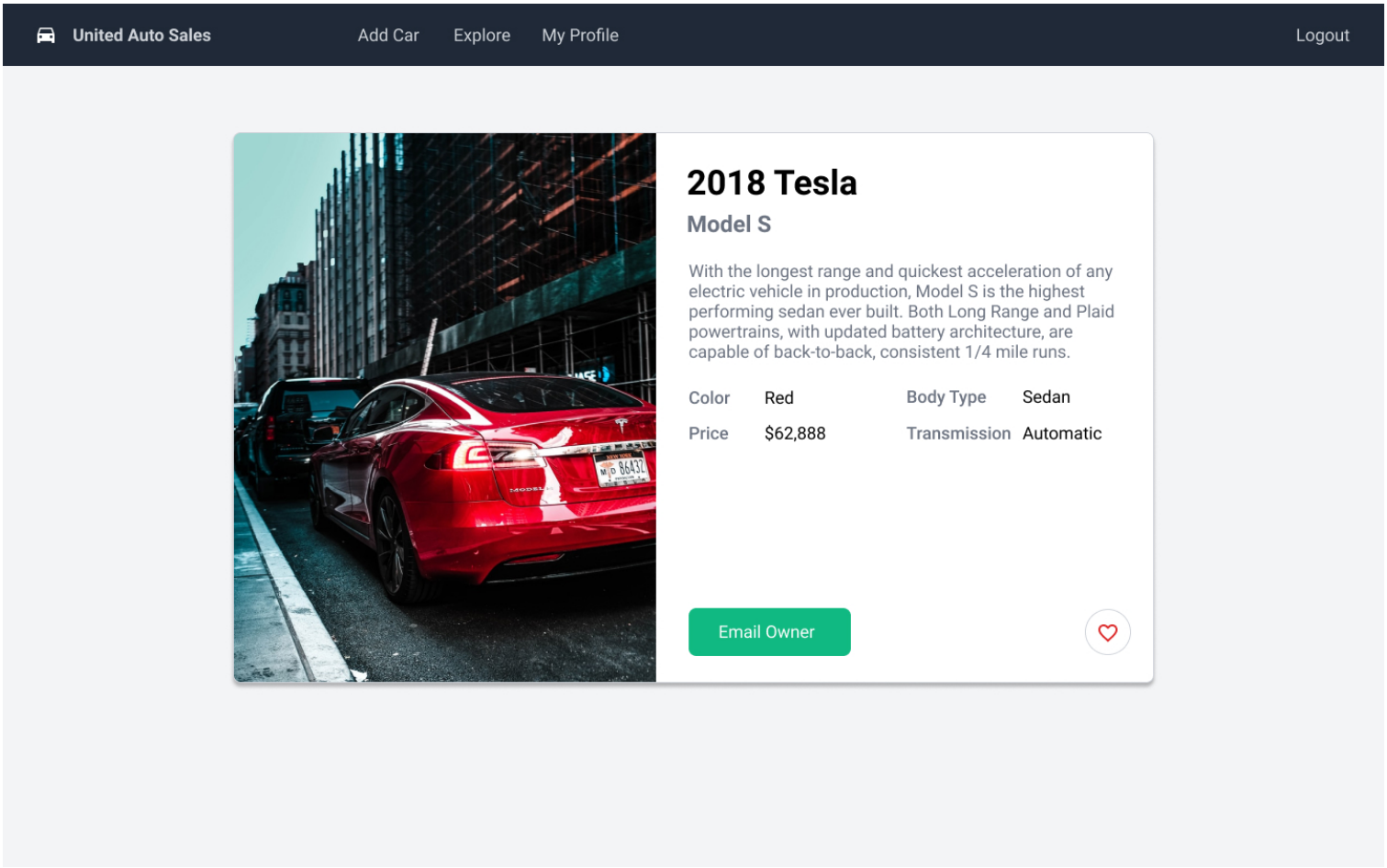

Figure 5: Car details page - when the heart is clicked to favourite a car it should change colour.

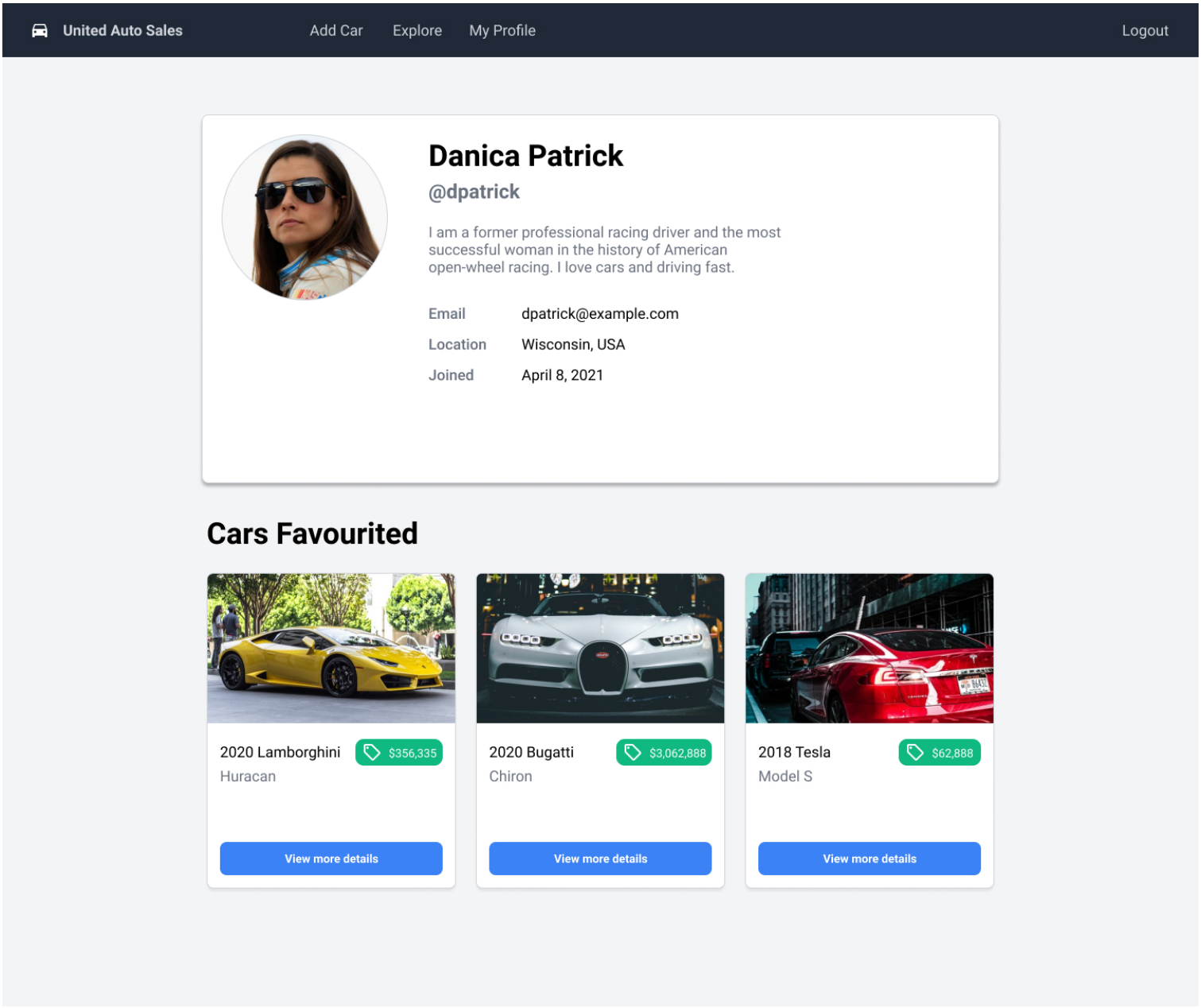

Figure 6: User profile with Favourite cars

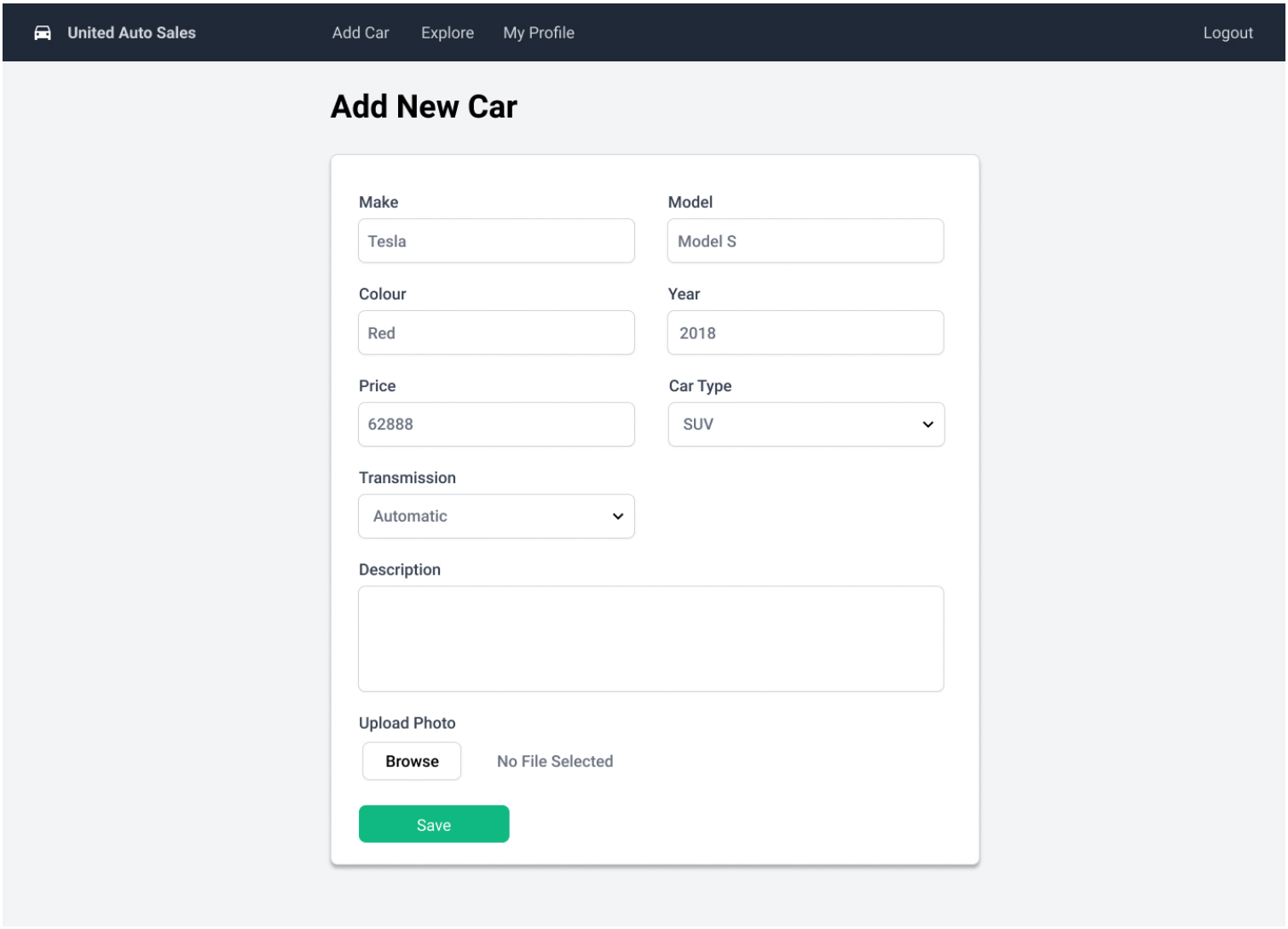

Figure 7: Form for adding a New car