Car Finder Web Portal

**Team Members**

John Hayde: [jhayde@clemson.edu](mailto:jhayde@clemson.edu)

**Problem Description**

Due to the COVIC-19 Pandemic, many small car dealerships have been unable to

operate. Most have been permanently closed, while some have been able to see a few

customers a day. However, most people do not want to leave their house out of fear of the

pandemic. Additionally, these small dealerships are in to financial standing to afford a largescale database system for their car sales lot. They do not need all the bells and whistles of a large-scale company database – they just need to sell cars. That is where my portal will come in – to help small businesses sell more cars.

**Project Motivation**

This problem is interesting because it covers all of the database features. I will get to use

multiple tables to keep track of cars, users, and admin. All data will be able to be edited by the

appropriate user (user editing their info, admin adding/removing cars). The potential users for

this program are people wishing to buy a car without leaving their home, a number that is

exponentially growing day by day.

**Description of Approach Taken and System Architecture**

To solve this problem, I used a number of different systems and services. I used HTML to help design the basic framework and layout for the front facing pages. HTML is also used on all pages that do not need any dynamic content. I used a combination of HTML, PHP, and MySQL to develop the pages that needed to support dynamic content. HTML was used to provide the basic layout and framework, PHP was used to display data on the front end and process data on the backend, and MySQL was invoked through PHP to connect to the database. The database is hosted through the Clemson Buffet service. CSS is used to style the tables that are displayed upon a query.

The overall system architecture is very simple. There are main pages written in HTML and PHP to support dynamic content. MySQL is used through PHP to connect to the database for data verification, data queries, and data updating. There are a number of backend services performed by PHP which the user does not have access to. These services are located in the PHP folder. Three tables are used to store data: admin, user, and vehicle. The admin table simply holds the username and password for and administrator. The user table holds all pertinent information for each user. The vehicle table holds all of the vehicles for sale and contains all pertinent information for each vehicle (VIN, make, model, color, price, mileage, year).

**Function and Feature Implementations**

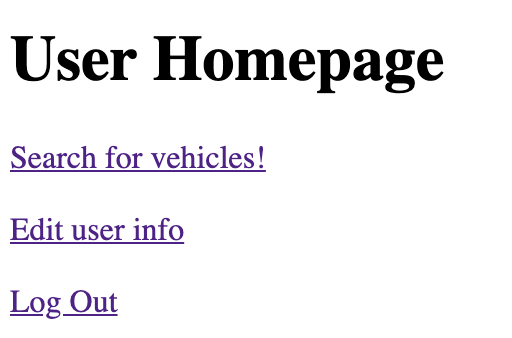
All of the basic features have been implemented in my portal.

User Account Management

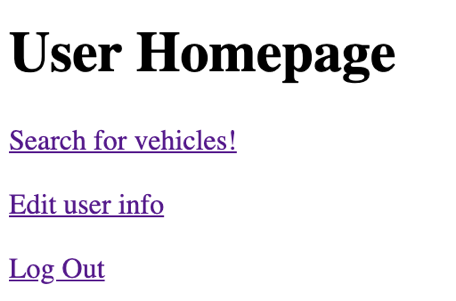
* A picture containing table

  Description automatically generated**A picture containing table

  Description automatically generated**New user registration module: this feature is implemented by selecting “New User? Register Here” from the main page. This takes the user to the registration page. If they fill out all of the boxes, and their passwords match, then they will be sent to the user homepage and their information will be added to the database. If they leave a box empty or the passwords do not match, they will get an error message and be prompted to try again. This feature can be tested by navigating to the page and entering information for a new user. First, leave a box blank and try to submit. Then have mismatched passwords. Finally, enter everything correct and get sent to the user homepage.

****

* Sign in module: if a user already has an account, they can sign in by selecting “Already Registered? Login Here” from the main page. This takes them to a new page where they are prompted for their username and password. If they are correct with what is stored in the database, the user will be sent to the user homepage. If not, they will get an error message saying that their username or password is wrong, and to try again. This feature can be tested by first successfully making an account, then logging out and using the username and password to log back into the system. First enter an incorrect username or password to see the error handling. Then enter the right ones and get logged in.

**Graphical user interface

Description automatically generatedA picture containing diagram

Description automatically generated**

* Profile update module: if a user wishes to change some of their information, they are welcome to do so after they have already logged in. They can login, then select “Edit user info”. From there, they are prompted to re-enter all of their information. Users are not able to change their username, but they are asked to verify it on this page. They have the option to change all of their personal information, including their password. If the passwords do not match, the data will not be saved, and they will be prompted to re-enter the new information. If all boxes are filled and the passwords match, they will be sent back to the user homepage. This can be tested by first creating a user account, then navigating to this page. Try to submit the form with blank lines or mismatched passwords, then have all fields correct and hit submit. You can change your password, log out, and log back in with your new password to show the update is working.

**Text

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated**

Data Management

* Allow users to view category level data, users are able to query data sets over a specified category: once logged in, users have the option to select “Search for vehicles!” which takes them to the vehicle search page. Here, there are detailed instructions on the different types of searches they can make. Users have the option to search over all fields, or only by make or make and model. If a search fails, it will default to searching for just the make and model to give the user a better idea of what is available. Once a search is made, the results will populate in a table below the Search button. To test this feature, first search for just a make. Then, search for make and model. Then, try searching with every field filled. You will see varying results populate, and different messages show up depending on the success of your search.

Text, letter

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

* Users are able to update existing record: while users cannot post vehicles for sale, as this portal is for small businesses to sell vehicles, users do have the options to edit their personal information. They can do this by selecting “Edit user info” from the user home page. From there, they are prompted to enter their username for verification, and they can re-enter all of their information, including their password. This can be tested by entering your username and new information, including a new password. Then, log out and log back in. since you updated the password, your old one will not work, and your new one will. This will not work if fields are blank or the passwords do not match.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Database Administration

* View/query/update user accounts: an admin has 2 options here, they can view and search for user accounts, and they can update a specific user account. To view and query the user accounts, select “View/Query User Accounts” from the admin home page. This prompts you to search by username, first and last name, or just last name. a search of username or first and last name will only yield oner user, while a search of just last name could yield multiple users. To test this, you can create a new user using the. New user registration, then log into the admin portal and search for that user by any 3 of the options available.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

To edit the user account, an admin will select “Update User Account” from the admin home page. This takes them to a new page which prompts them to enter the user’s username and then new information in the fields below. All fields must be filled out and the passwords must match in order for it to be successful. If not, the admin will be prompted to try again. You can test this by editing a user’s information, then searching for that user on the admin user query page. If it is successful, the admin will be routed back to the admin home page.

Graphical user interface

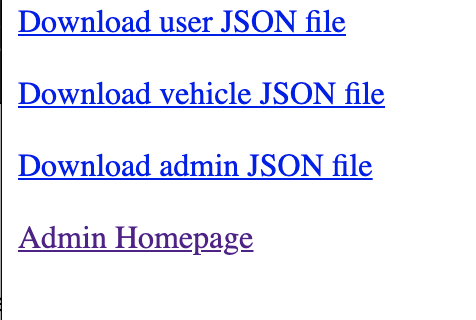
Description automatically generatedA picture containing graphical user interface

Description automatically generated

* Backup selected tables related to web portal: To use this feature, log into the admin page, then select “Backup Database Tables”. This takes you to a new page which prompts you to click to create a backup of the current tables and download them. When you click this link, a copy of each table of the database is made and converted into a JSON file. You will be sent to a new page which prompts you to download each file. This can be tested by downloading a copy of the files, then adding more users to the database and again downloading the files. Each time you download the files, a new copy is made based on the current state of the tables.

Graphical user interface, text, application

Description automatically generated



**Limitations of Current Implementation and Thoughts on How to Fix**

My current implementation has a few limitations. The first is that there is no handling for if a user tries to create a new account for someone with the same username or email. This is a large constraint that needs to be fixed going forward, as it is impossible for a user to know what the taken usernames are. It would also be useful to implement account recovery if a user already has an account but forgot their login information.

The second is that the portal is not extremely appealing in terms of design. Should I move forward with this and want actually business to use it, I would want to implement more CSS/JavaScript to make it look fancier.

The third limitation is that the vehicle search is not perfect. Ideally, I would like for the user to be able to fill out any of the fields and search based on those factors. I could not figure out a good way to implement this, so I had to control the search a little bit with my instructions on the page. Going forward, I will have to find a better way to figure out how to do a search based on any field that is filled out, not just specific fields. I am still unsure of how to exactly implement this.

**Link to Web Portal**

<http://webapp.cs.clemson.edu/~jhayde/4620/project/>

**Admin Username and Password**

Username: root

Password: root\_pass

**Link to Source Code**

<http://webapp.cs.clemson.edu/~jhayde/4620/project/download_tarball.php>