CEN 308 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

CarDiaries

Prepared by:
Ahmed Becirevic
Kerim Celjo

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

24.06.2022.

TABLE OF CONTENTS

1. Introduction

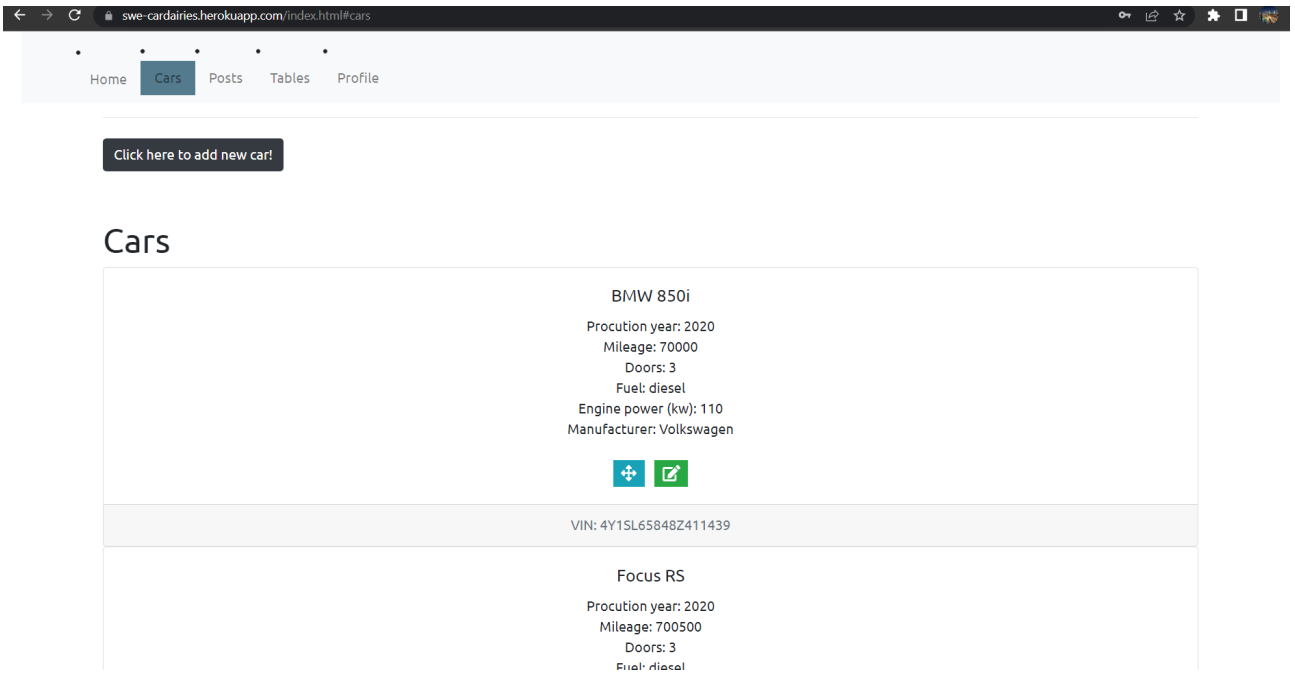
1.1. About the Project

CarDiaries is a web based application that offers users a platform to post about their personal cars. Other not signed in users can also view the posts for a particular car by inputin the VIN. The application can be access through [HERE.]

1.2. Project Functionalities and Screenshots

Main features of CarDiaries include:

- login and register with validation and confirmation through email
- forgot password
- create and update car
- create post belonging to a car and upload an image
- edit post
- server-side processed datatable of accounts
- logout



Click here to add new car!

Cars

Update Car

BMW 850i | 2020

4Y1SL65848Z411439 | 70000

3 | 110

Volkswagen | diesel

Clear | Update Car

VIN: 4Y1SL65848Z411439

Focus RS

Home | Cars | Posts | Tables | Profile

Click here to add new car!

Name of the Car | Prodcution Year

VIN | Mileage

Number of Doors | Engine Power (kw)

Manufacturer | Fuel Type

Clear | Add Car

Cars

Home | Cars | Posts | Tables | Profile

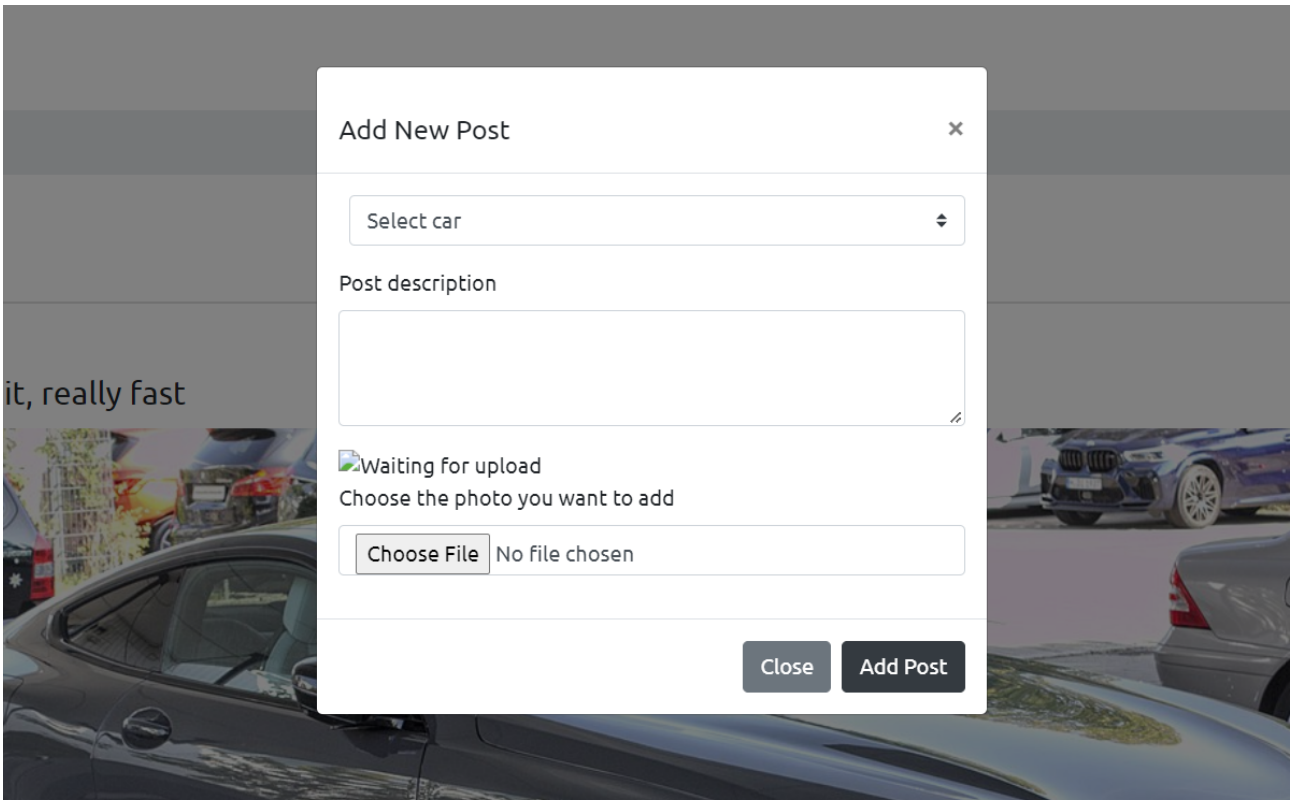
Home

Collection of posts.

Add Post

Post description:

Just bought it, really fast



Home Cars Posts Tables Profile

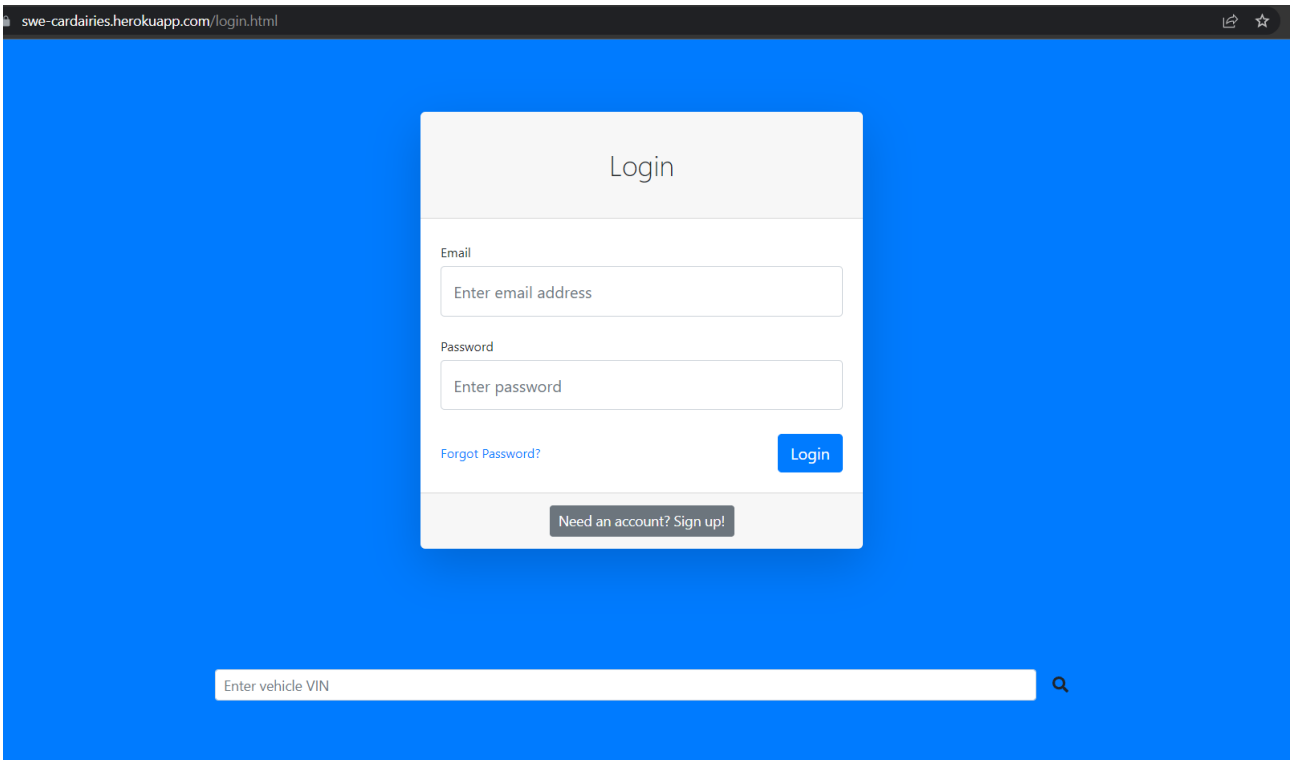
Your Profile Information

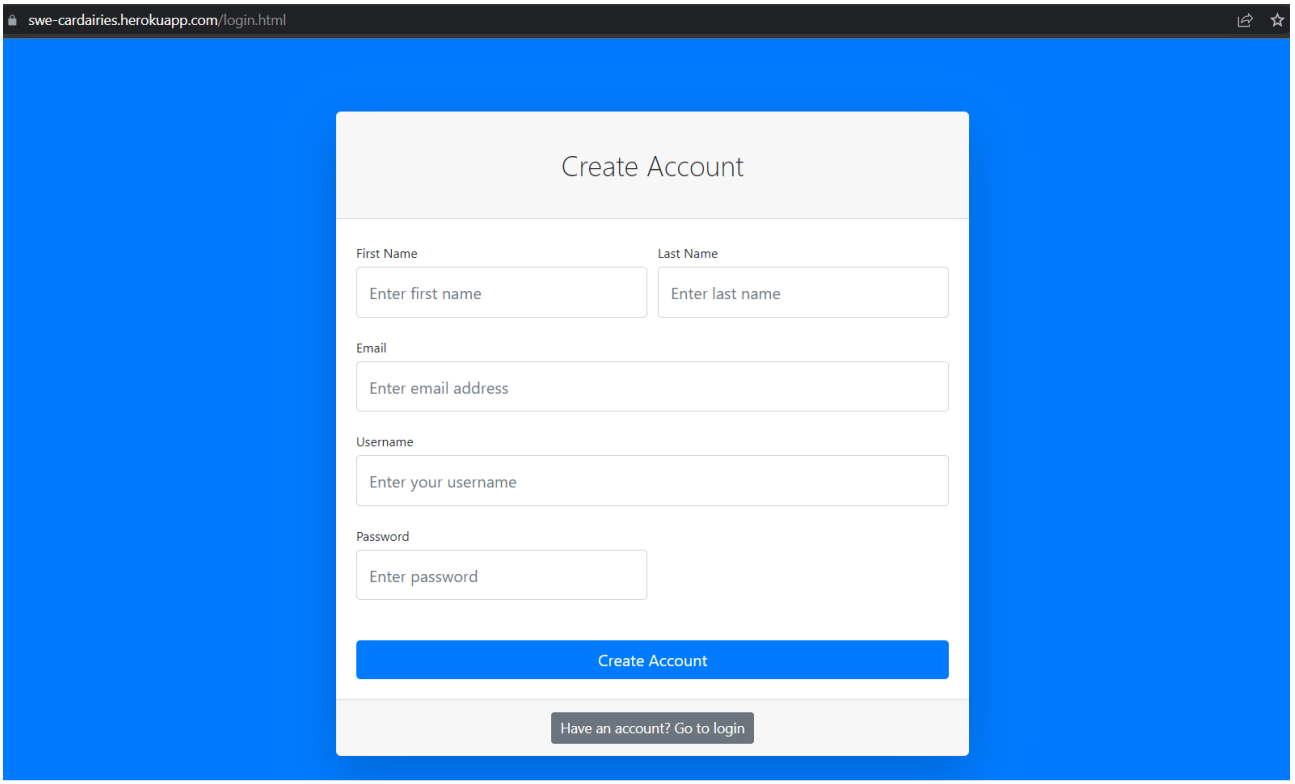
ahmadis

First Name: **Ahmed**
Last Name: **Becirevic**
Email: **becirevicahmed@gmail.com**
Account creation time: **2021-04-03 18:09:53**

ID: 61

Log Out


swe-cardairies.herokuapp.com/login.html
Login
Email
Enter email address
Password
Enter password
Forgot Password?
Login
Need an account? Sign up!
Enter vehicle VIN


swe-cardairies.herokuapp.com/login.html
Create Account
First Name
Enter first name
Last Name
Enter last name
Email
Enter email address
Username
Enter your username
Password
Enter password
Create Account
Have an account? Go to login

2. Project Structure

2.1. Technologies

Technologies used for the backend development of the application are: PHP, Flight PHP, composer for the package management, AWS S3 client to connect to DigitalOcean spaces for image upload, Gmail SMTP for sending emails. Database for the project is MySQL using InnoDB engine for tables.

Frontend part of the project was made using HTML, CSS and jQuery. Bootstrap was used to help with styling and functionality. Frontend router was implemented to make it a single page application.

Coding standard for PHP was PEAR which was forced through the PHPStorm IDE. Frontend formatting was done through the built in IDE formatter.

2.2. Database Entities

Database entities used for the application include:

- users
- cars
- posts
- images
- accounts

2.3. Architectural Pattern

Architectural pattern used fro this project is the **layered architecture pattern** that was implemented in the backend with three layers. The first (most low-level) layer is the data access layer where we access the database entities, the layer above it is the services layer that contains all of the business logic and the top layer is the application layer that contains the routes or the endpints of the API. This pattern was chosen because it allows a really good decoupling of different logic of the RESTful API and a clear flow of the data. This pattern also allows for a good extensibility and project growth.

2.4. Design Patterns

For this project we implemented two design patterns: Singleton and DAO design pattern.\

- **Singleton -** can be seen in the class DatabaseConnection
  (path: api/DatabaseConnection.class.php)
  It was used to ensure a single instance of the database connection. Since we need to query the database very frequently it also allowed a global access to the DB connection.
- **DAO** (data access object) - this pattern was chosen to abstract the logic of the database querying and manipulation from the services. It was implemented by making one BaseDao class that had the general methods for manipulating the data

(create, update, delete, fetch) and every other class (that represented one database entity) extended this class and its methods. This way we provided separation of logic regarding the database.
(path: api/dao/BaseDao.class.php, api/dao/UserDao.class.php etc.)

3. Conclusion

This project was a huge learning curve regarding design patterns. We are satisfied with the implementation and the features. We wish that we wrote tests for the backend but we did not manage due to the lack of time. UI tests were written in Selenium and cover all of the big features. They were all successful and all of the functionalities of the web app are stable and functional.