RTPlan Road Trip Planning Program:

Managing and visualizing your itinerary

Department of Computer Science and Engineering

California State University, Fullerton

Project Advisor: Dr. Lidia Morrison

Submitted By: Zhihuang Chen

CPSC 491

May 20, 2023

Table of Contents

Table of Figures

Abstract

According to a 2017 survey by the online statistics portal Statista of 965 U.S. residents using transportation data on vacation, 73 percent of respondents chose to drive themselves. This finding is even higher than traveling by air (69%). The main reason for the popularity of car travel is its flexible itinerary. Travelers can change their accommodation at any time. Self-driving travel is more convenient than planes, trains, and other travel methods that require people to choose stations in advance. However, its disadvantage is that the cost is uncertain. People cannot know whether a stop they decide on a whim will exceed their travel budget. So, a tool that helps people figure out their road trip budget would go a long way. In this article, we discuss the construction of a navigation website based on a static website and Google Map API to provide users with a practical travel planning tool. Static websites are typically used for simple online brochures, portfolios, documentation sites, or personal blogs that don't require frequent content updates. Individual users must obtain data in network clouds such as Google Maps through APIs. The projects discussed in this article were developed on a budget and for non-profit purposes and will be based on GitHub Pages and a low-productivity API. Since the static website application data is based on the data of other servers, a function that can output data, such as printing, is required to facilitate offline use by users. Ensure users can view content offline and explore it throughout development.

1.0 Introduction

RTPlan is a comprehensive road trip planning tool designed to provide reliable assistance to road travelers. With the help of RTPlan, users can know the travel expenses, time, distance, and the number of refueling times of the trip before departure. At the same time, by utilizing these features, users can effectively plan their road trips while visualizing their budgets.

Additionally, the website will be accessible on any device as a static website application through a single code base. Since static websites involve no server-side processing, they tend to load quickly because there is no need to retrieve data from a database or dynamically generate content. In terms of hosting, static websites can be hosted on simple web servers, and they don't require specialized server configuration or databases. Static websites are generally more secure in terms of security because there are no server-side scripts or database interactions that attackers can exploit. However, the downside of static websites is limited interaction. Static websites interact differently than dynamic websites. First, they cannot process user input or perform complex calculations on the server side. The second is that the content of a static website remains the same unless manually updated. It often includes text, images, and other media. They consider that the accounting system will be added to develop subsequent versions. Therefore, it is necessary to change the framework of the website to a dynamic website.

2.0 Problem Statement

While various online tools are available for road trip planning, many fall short of providing a comprehensive, intuitive, and user-friendly experience. Many current tools do not provide information about travel conditions, the range a vehicle can drive on a full tank, marked gas stations on a map, and budgets. It is a major flaw in road trip planning, as the lack of this information greatly reduces the accuracy and timeliness of planning a trip. Some tools can be complex and non-intuitive, making the planning process tedious and overwhelming. A user-friendly interface and easy-to-understand features are essential to encourage more users to use the tool effectively. In order to solve these problems, an accurate and easy-to-use tool is needed which can provide users with convenient and accurate road trip planning through an integrated service platform. It will improve the planning process, making it more efficient and enjoyable.

2.1 Problem Definition

The road trip planning process is fragmented and lacks a user-centric approach. Current solutions often offer a fragmented approach, requiring travelers to utilize multiple resources and tools to plan a single trip. This disjointed approach increases the time and effort required to plan, making the process less enjoyable. Many of these planning tools do not provide accurate and personalized data or interactive elements; they only judge trip budgets based on distance. Therefore, the problem defined here is the lack of a comprehensive, user-friendly, interactive tool that integrates the distance,

time and budget data required for road trip planning and provides a personalized plan according to the User's vehicle.

2.2 Problem Rationale

RTPlan is a comprehensive road trip planning tool designed to help users gain useful data by providing accurate pre-departure travel information including cost, time, distance and refueling points. With RTPlan, users can create effective road trip plans while visualizing their budget. The initial approach was to develop RTPlan as a static website application using a single code base. Static websites that don't involve server-side processing load quickly because there's no need to retrieve data from a database or dynamically generate content. They can be hosted on simple web servers, requiring no specialized server configuration or databases, providing a security advantage since there are no server-side scripts or database interactions that attackers can exploit. However, considering the limitations of static websites, such as the inability to process user input or perform complex calculations on the server side, manual updates are required, and RTPlan will be transitioned to a dynamic website framework in subsequent versions. This approach will allow us to introduce an accounting system, enhanced user interaction, and customization.

3.0 What is Known About the Problem

The Importance of Road Trip Planning: Planning a road trip itinerary can be complex and tedious. In the article "Work and financial planning," author Jason Butler wrote:

"Planning your finances can seem daunting and overwhelming. If you aren't comfortable with numbers and feel bamboozled by financial jargon, it's easy to switch off, particularly given. There are any number of other, more enjoyable, activities on which you can spend your time and money." (Butler, 2019) Planners need to collect a lot of data and information to make an effective and perfect plan. Tools are sometimes needed to help plan even with solid data and information. It is possible to improve the economy of the travel plan effectively.

Static website & GitHub Pages: A static website is a collection of HTML, CSS, and sometimes JavaScript files, where each page is a separate file stored on the server. In the article "Building a static website with Jekyll and GitHub Pages," the Amanda Visconti said: "Static websites, on the other hand, do not use a database to store information; instead, all information to be displayed on each webpage is already contained in an HTML file for that webpage." (Visconti, 2016) Thus, this type of website has advantages over a dynamic website because it requires fewer server resources, loads faster, and is simpler to develop and maintain. However, static websites cannot serve dynamic functionality or content without external APIs or serverless functions. Also, the web server used by this project is GitHub Pages. It is a feature offered by GitHub that allows users to host websites directly from repositories on GitHub. Amanda Visconti wrote in the article: "GitHub Pages is a free place to store the files that run a website and host that website for people to visit" (Visconti, 2016). GitHub Pages is a free static website server. Essentially, GitHub Pages is a static site hosting service that takes HTML, CSS, and JavaScript files directly from a

repository on GitHub, optionally runs them through a build process, and then publishes the website.

Google Maps API: The Google Maps API is a service that allows developers to embed Google Maps into web pages using JavaScript or HTML. Adding dynamic map content, location services, and other features can revamp static web pages. However, it does require some coding ability to implement and can slow down your webpage if not done properly. In terms of cost implications, while using the Google Maps API is free up to a point, for sites with high traffic or extensive use of the API, the cost can add up quickly. Therefore, developers must consider these potential costs when integrating APIs into static websites. Figure 1 is the charging details of Google Maps API.

SKU: Static Maps

A request to the Maps Static API.

MONTHLY VOLUME RANGE (Price per MAP LOAD)		
0–100,000	100,001–500,000	500,000+
0.002 USD per each (2.00 USD per 1000)	0.0016 USD per each (1.60 USD per 1000)	Contact Sales for volume pricing

Figure 1: Price of Map API

4.0 Proposed Method of Solving the Problem

The proposed solution involves the creation of a comprehensive road trip planning application called RTPlan, which seamlessly integrates various elements of road trip planning into one easy-to-use platform. Here is how the solution is expected to work:

unified platform, integration with Google Maps API, personalization, cost and time estimation, and user-friendly interface.

RTPlan will contain all the necessary features for planning a road trip, including route maps, stopover suggestions, and fuel cost estimates. Users will no longer need to switch between multiple apps or websites to gather the necessary information, reducing the time and effort spent on travel planning. At the same time, users can get personalized data information according to their vehicles, and it is easy to use.

Overall, RTPlan will provide a holistic solution to the fragmented and often frustrating road trip planning process by bringing together all the necessary information and tools into one efficient, user-friendly, interactive platform.

4.1 Project

This project involves integrating the Google Maps API into a static website to provide users with a navigation interface while outputting trip budgets and information based on the vehicle information entered by the User. This feature will enable users to mark a location from the website and then get the corresponding information from the marked point of departure and destination.

4.2 Benefits, Significance, and Future

The website helps users plan their travel more effectively by bringing all the necessary information and tools into one place. At the same time, the app helps users estimate and manage their travel expenses, potentially saving them money. The ability to customize an itinerary ensures a more enjoyable and personalized travel experience.

The significance of RTPlan is that it can revolutionize how people plan road trips. It addresses a common pain point for travelers and fills a gap in the market for a comprehensive, easy-to-use road trip planning tool. By simplifying the trip planning process, RTPlan has the potential to make road trips easier and more enjoyable for travelers at large. In addition, integration with Google Maps API increases the convenience and affordability of road trips.

Going forward, RTPlan can be expanded and improved in a number of ways, for example, incorporating features that allow users to share their itineraries and experiences with others, creating a community of road trip enthusiasts, and secondly, building with service providers such as hotels, restaurants, and tourist attractions Partnerships to provide users with special offers or recommendations. Alternatively, implement machine learning algorithms to provide users with personalized destination recommendations based on their past travel, preferences, and behavior. RTPlan has the potential to be a helpful road trip planning tool, and its future extensions and improvements could further increase the user experience and its functionality..

5.0 Objectives

The goal of this project is to integrate Google Maps API into a static website to enable navigation in the static website based on the location selected by the User. Ensure compatibility across different devices and browsers while maintaining optimal page load times. The result of this project will be a static website with the dynamic capabilities of Google Maps. Users of the site can easily access route and travel budget information, providing them with a better and more interactive experience.

6.0 Activities

To develop a static website using Google Maps API integration, the first step is to create the front end of the static website using HTML, CSS, and JavaScript. It will lay the foundation for the design and interactivity of the site. In the second step, integrate Google Maps into the site using the API key obtained from the Google Cloud Platform. Customize the map to show specific locations and other features as needed. The third step is to create a web application manifest file that provides information about the application (such as name, author, icon, and description) in a JSON text file. The fourth step is to test the site across different devices and browsers, including Google Maps and Navigation, to ensure full functionality. The fifth step is to write settings files and documentation to facilitate the deployment of the website to the hosting server and ensure that the website is optimized for performance, including fast load times and efficient use of resources. Then, deploy the code base to a hosting server or the GitHub Pages platform, ensuring that the site is accessible to users and works as expected on the live server. Finally, make the codebase available in an open-source repository to invite contributions from the developer community or to allow others to learn from the project and contribute new ideas.

7.0 Development Environment

The site will be developed using Visual Studio Code and tested using Jekyll. The codebase will consist of HTML, CSS, and JavaScript.

Hardware:	A PC or laptop
Programming Languages:	JavaScript
Font-end Technologies:	HTML, CSS
IDE Platform:	Visual Studio Code, GitHub Desktop, Google Maps Platform, Jekyll
Database:	None
Operating System:	macOS Ventura 13.3.1 (a)

Browser: Chrome (Recommended), Firefox, Safari

8.0 Operational Environment

The operating environment for developing a static website integrated with the Google Maps API includes a combination of hardware, software, and networking considerations.

Server:	GitHub Page
Google Cloud Platform account:	Need a Google Cloud Platform account to obtain an API key for Google Maps.
Networking:	Accessing the Google Maps API, testing the website, and deploying it to a server requires a reliable internet connection.

9.0 Overall Description

9.1 General Overview

Static website with Google Maps API integration designed to provide an interactive and accessible online user experience. It combines the simplicity and speed of static web pages with the powerful mapping and navigation capabilities of the Google Maps API. The site can be easily accessed through any web browser and requires no separate development for the Android or iOS platforms. By integrating the Google Maps API, the site allows users to view a specific location or get itinerary information and a street scene of the location. It is especially beneficial for businesses that want to help customers find their location or learn about their itinerary. The project provides a cost-effective solution for businesses or individuals wanting to provide their users with an interactive, reliable, easy-to-use web presence without requiring extensive development resources or platform-specific maintenance.

9.2 Application Overview

Incorporating the Google Maps API into a static website turns it into a highly interactive and powerful application that takes the best of both web and native applications. Users can get directions, travel costs, and itinerary information based on the selected location and view the surrounding area using a detailed interactive map. Integrating the Google Maps API into a static website can provide a powerful application that provides a seamless, interactive, and user-friendly user experience, benefiting the business and its customers.

10.0 System Architecture

10.1 Architecture Design

The static website for the Google Maps API integration will follow a modular design, with each component representing a specific application functionality.

1. Home Page: The home page is the central controller of the application, from where users can navigate to different components according to their needs.
2. Maps component: This component integrates the Google Maps API into the website. It displays the business location and provides an interface for users to interact with the map, see directions, and explore the surrounding area.
3. Vehicle Information Component: This component will fetch information from the User.
4. Match Vehicle Information Component: This component uses JavaScript to match user input with pre-stored vehicle information on the website.
5. *(In plan) Account system component: This component will allow customers to store vehicle information and historical data of trips by registering an account, but this involves dynamic website content.
6. *(In plan) Machine learning prediction module: This module will predict the MPG of the vehicle based on the vehicle data of the past three years.

The current project version only includes 1 to 4 components. Each component is designed to be self-contained and independent for easy maintenance, updates, and extensibility. Interactions between components are handled seamlessly to ensure a smooth user experience.

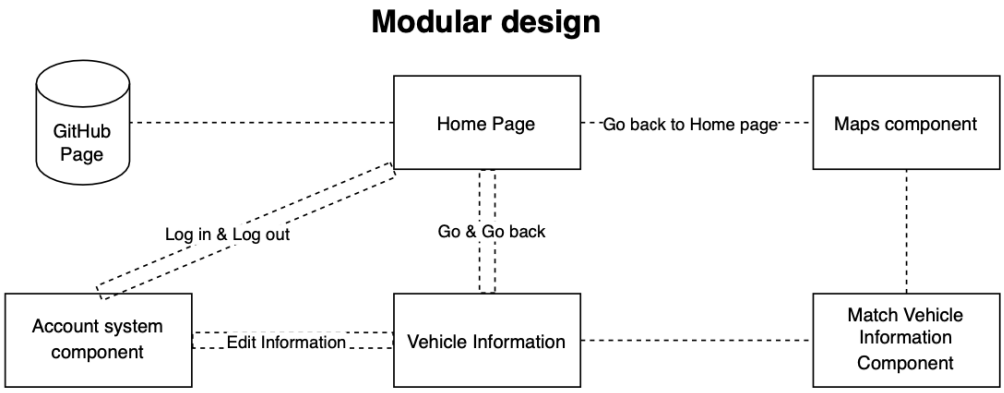

Figure 2 Modula design

11.0 UML Diagrams

11.1 Use Case Diagram

When the User opens the website, the User will see the login screen. From here, users can sign up for a new account, log into an account, or use the program as a guest. If the User chooses to use the program as a guest, the program will ask for vehicle information; otherwise, if the User has an account, he will directly enter the map viewing page.

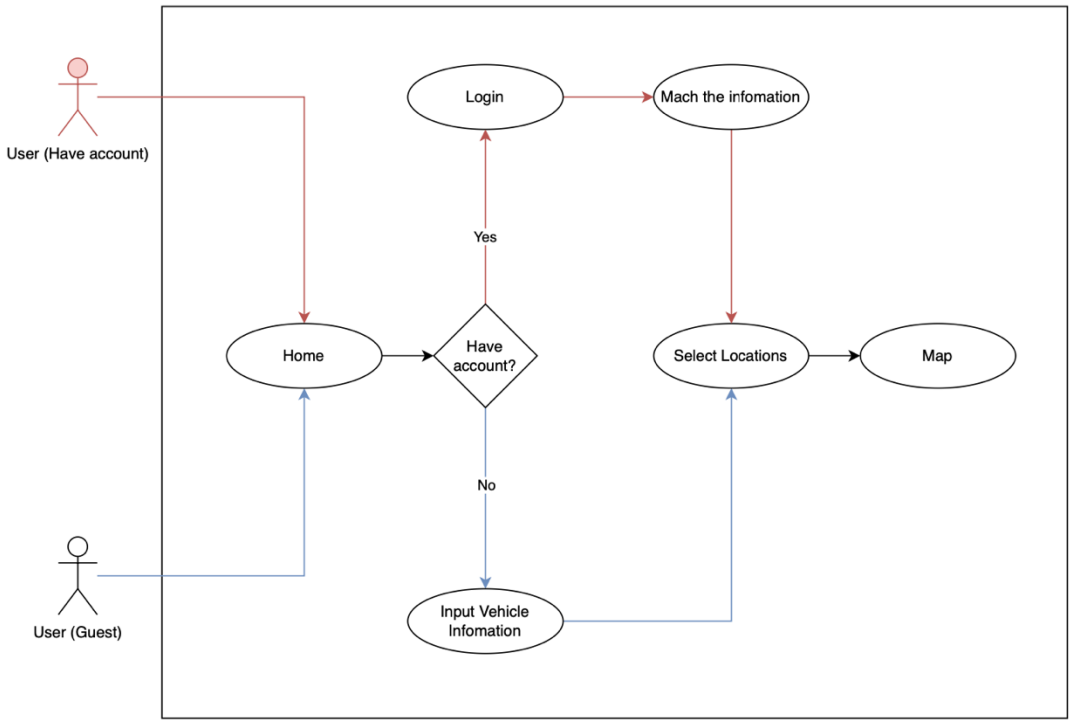

Figure 3: Use Case Diagram

11.2 Sequence Diagram

Sequence Diagram

Users enter vehicle and journey information, and the System organizes the information. Then, find matching information in the server and return the data to the User.

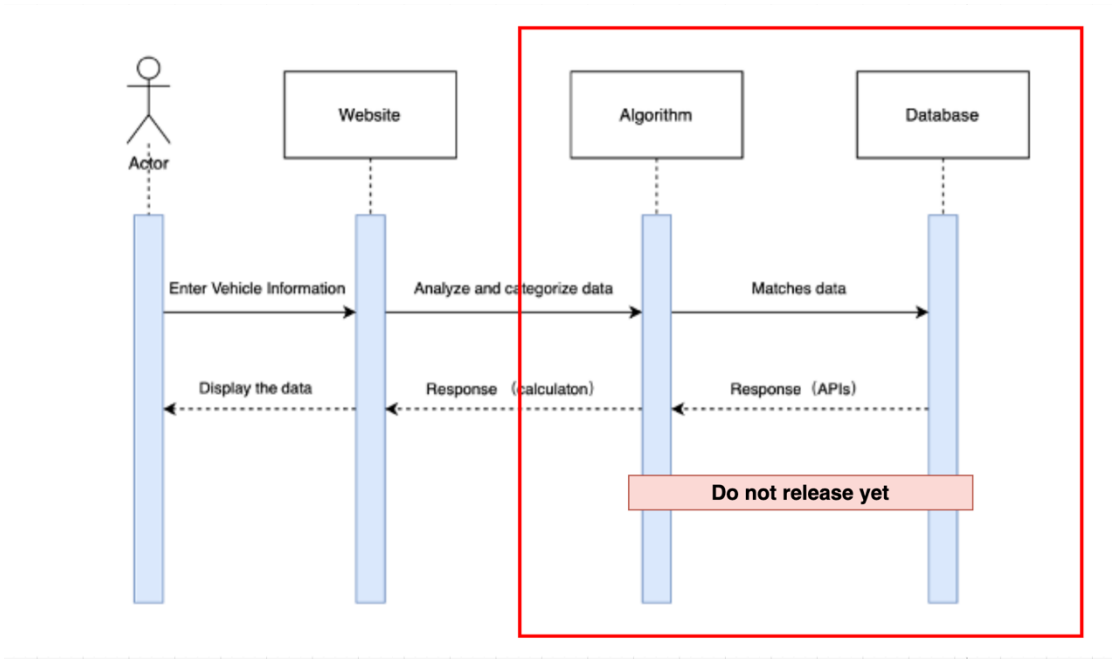

Figure 4: Sequence Diagram

11.3 Activity Diagram

Find the corresponding data in the database through the User's input, such as the official data of the vehicle manufacturer and the power consumption data that comes from the U.S. Department of Energy.

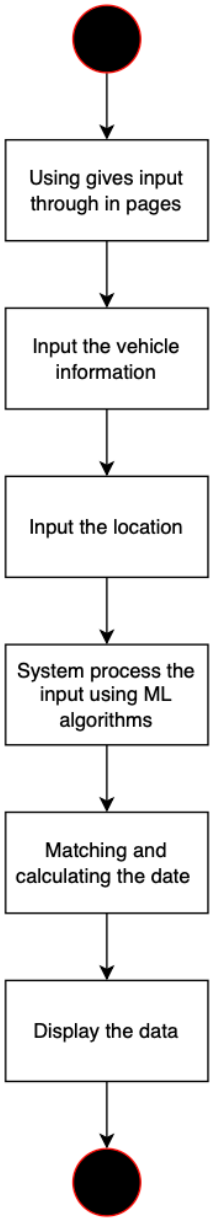

Figure 5: Activity Diagram20

12.0 Implementation

The website is primarily available to users in the form of web pages. Users can access web pages through specific URLs. There are two input boxes and three buttons on this page. The two input windows can be used to input username and password, respectively. At the same time, if the username or password is empty, you cannot log in. In addition, the three buttons are login, registration, and visitor entrance. The following is a display of the Home page.

Figure 6-a: Home page

Figure 6-b: Home page when Username or Password is empty.

Profile Page:

There are three buttons on this page. Of these, one goes to the map page. The second is to go to the vehicle information page. The third is to log out. At the same time, the User's username and email information will be displayed on this page. The following is the display of the Profile page.

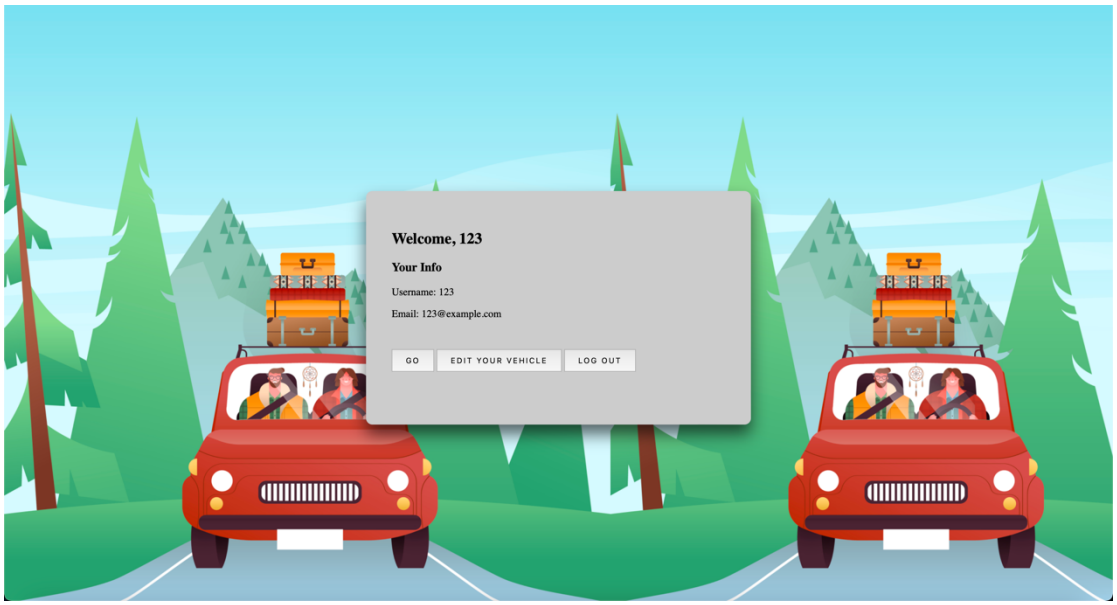

Figure 7: Profile page

Create Account Page:

There are three input boxes and two buttons on this page. The three input boxes are username, email address, and password. Like the Home page, the registration button will not be valid if the input box is empty. In addition, the two buttons are registration and return to the Home page. The following is a display of the Create Account page.

Figure 8: Create an account page.

Vehicle Information Page:

There are three selection boxes and a button to go to the next page on this page. The three selection boxes are year, make, and model. The following is a display of the Vehicle Information Page.

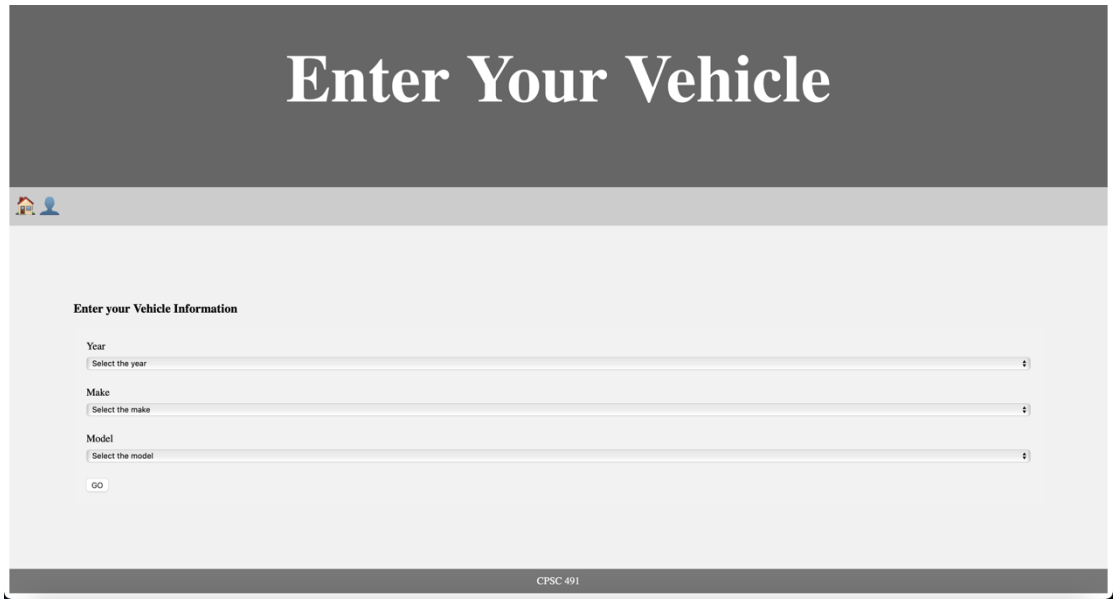

Figure 9a: Vehicle Information Page

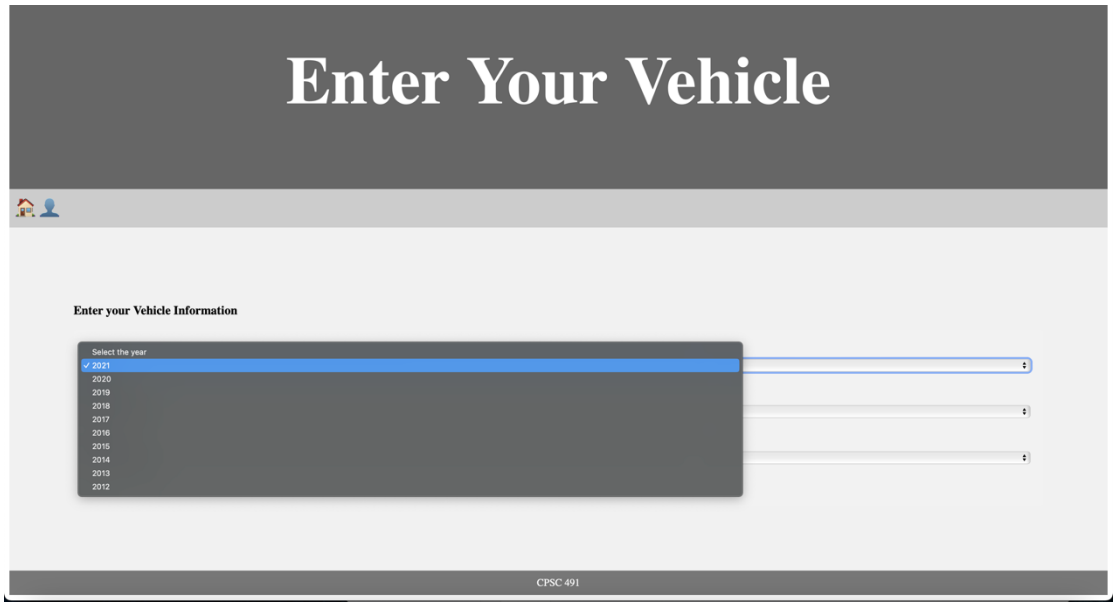

Figure 9b: Vehicle Information Page (Select Year)

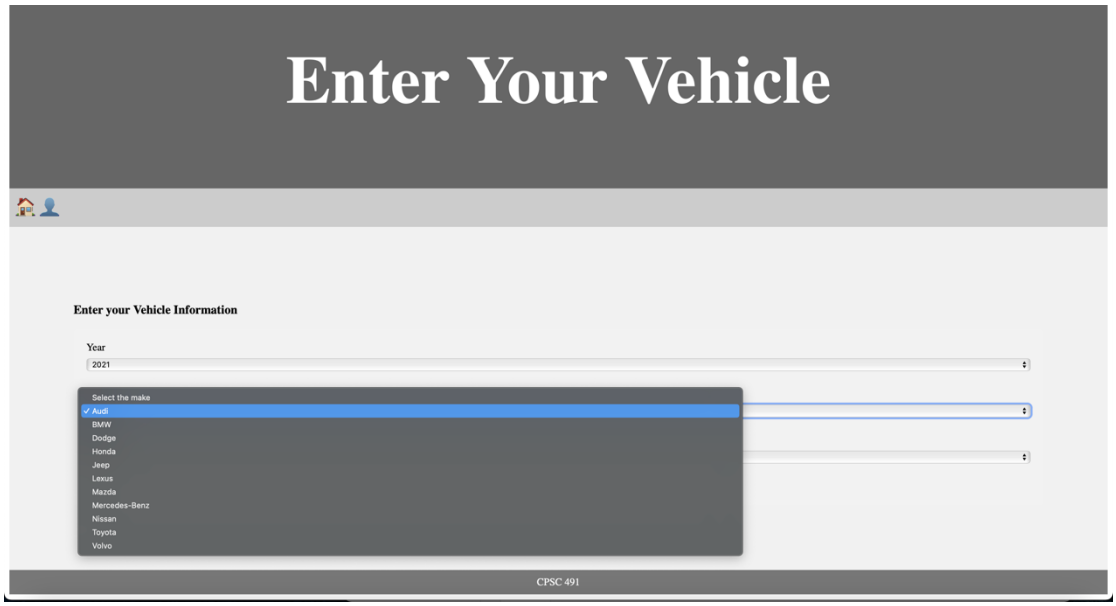

Figure 9c: Vehicle Information Page (Select Make)25

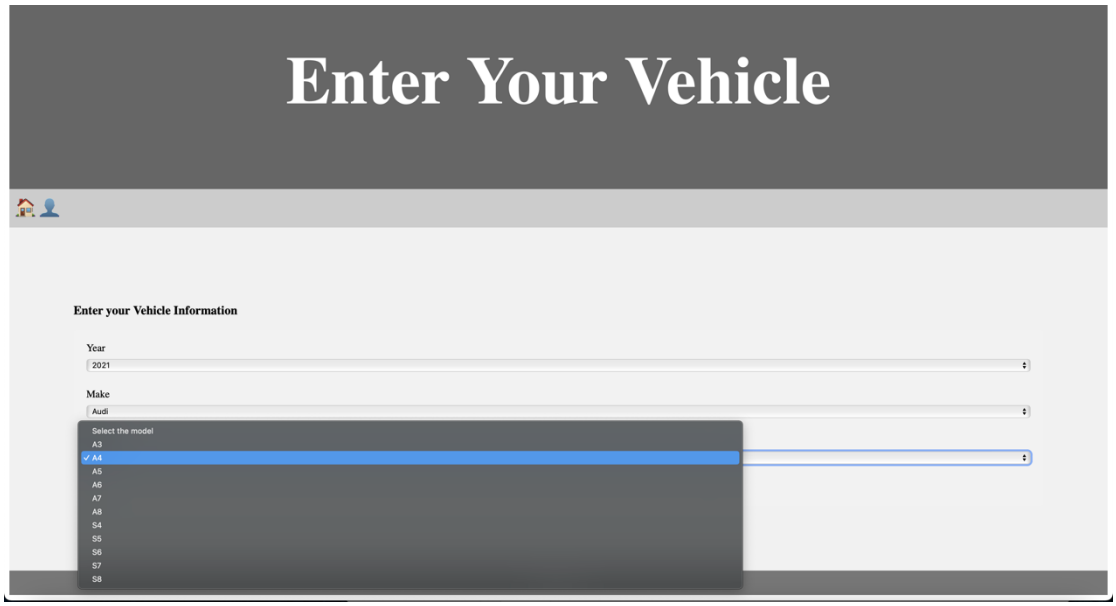

Figure 9d: Vehicle Information Page (Select Model)

Map View page:

There are four buttons in total on this page. They are the first button on the navigation bar in the upper left corner to return to the Home Page and the second button to log out of the account. The other two are at the bottom of the information output window, restarting calculation, and print button. Regarding the choice of location, users only need to click on the place they want to go on the map. The website will then show the User the budget, distance, time, number of times they need to refuel, suggested stops, and gas stations near the suggested stops. The following is a display of the Map View page.

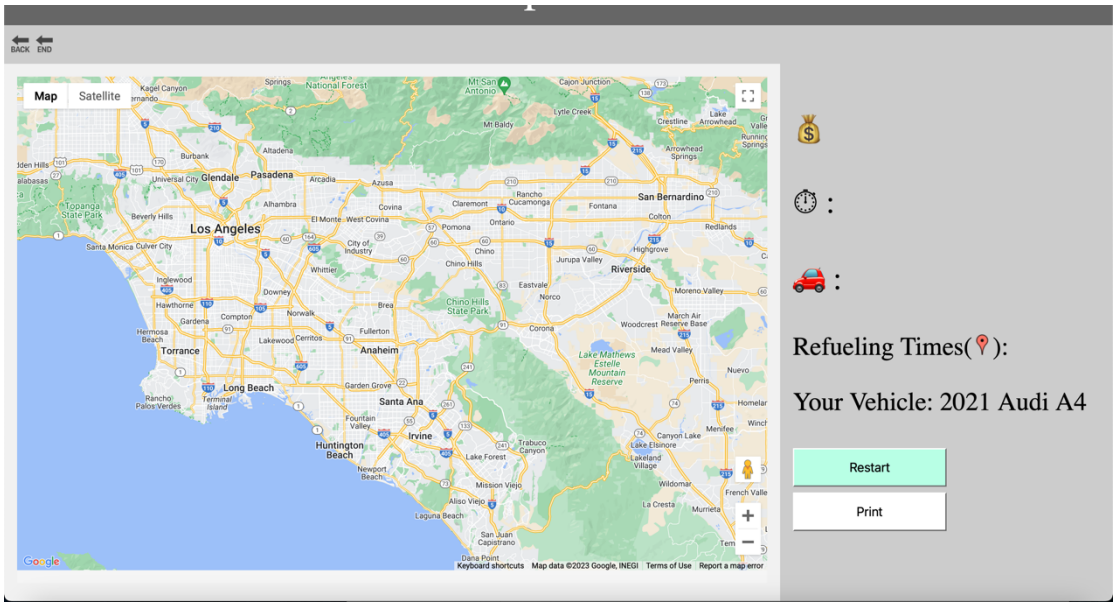

Figure 10: Map Page

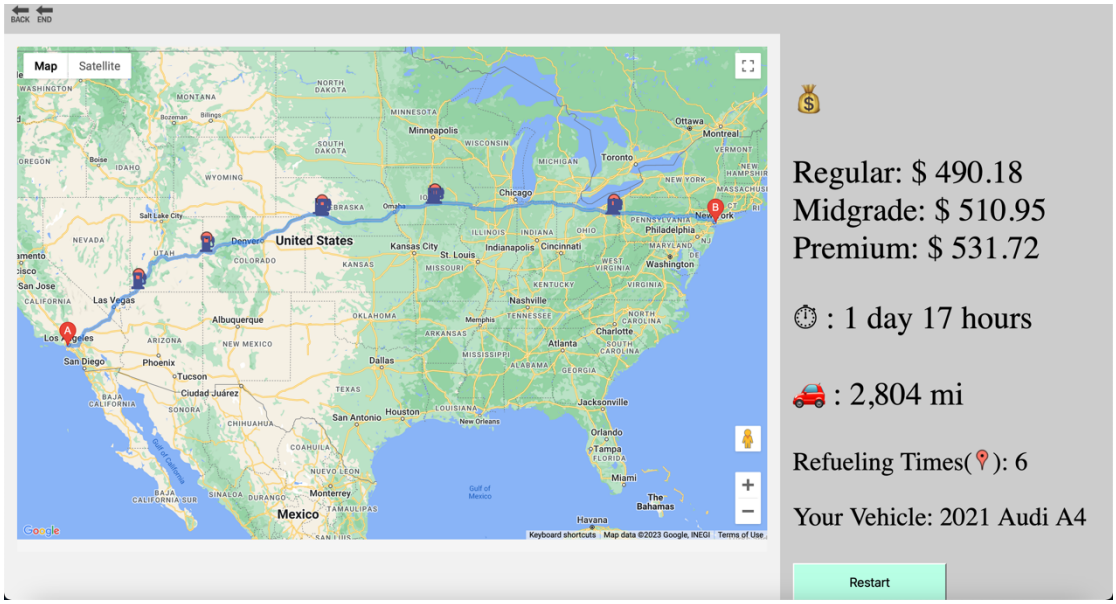

Figure 11a: Map Information Page (Information)

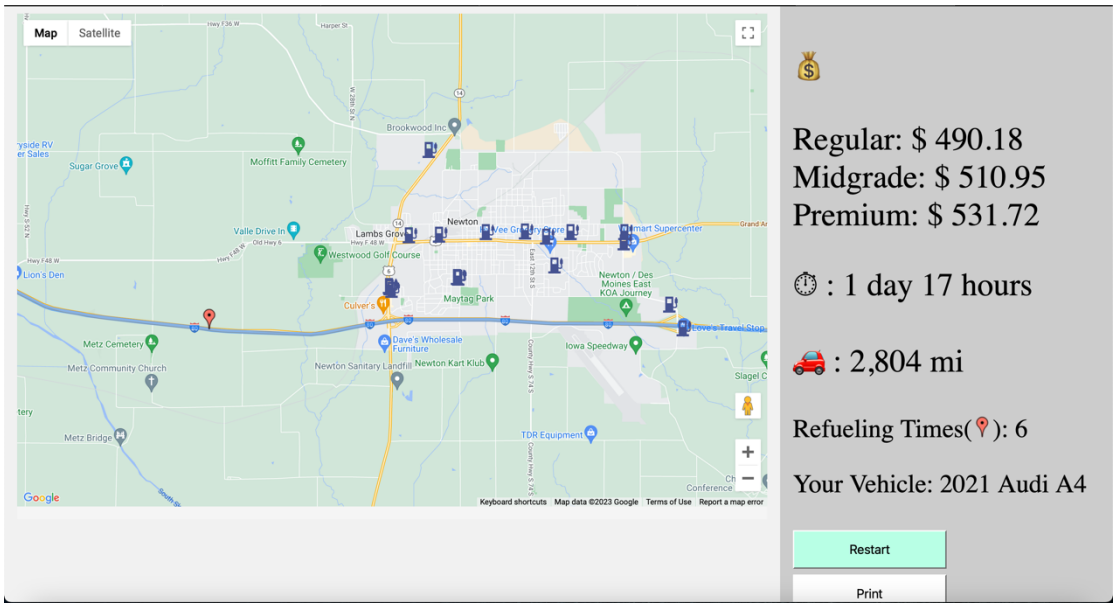

Figure 11b: Map Information Page (Gas station near Stop point)

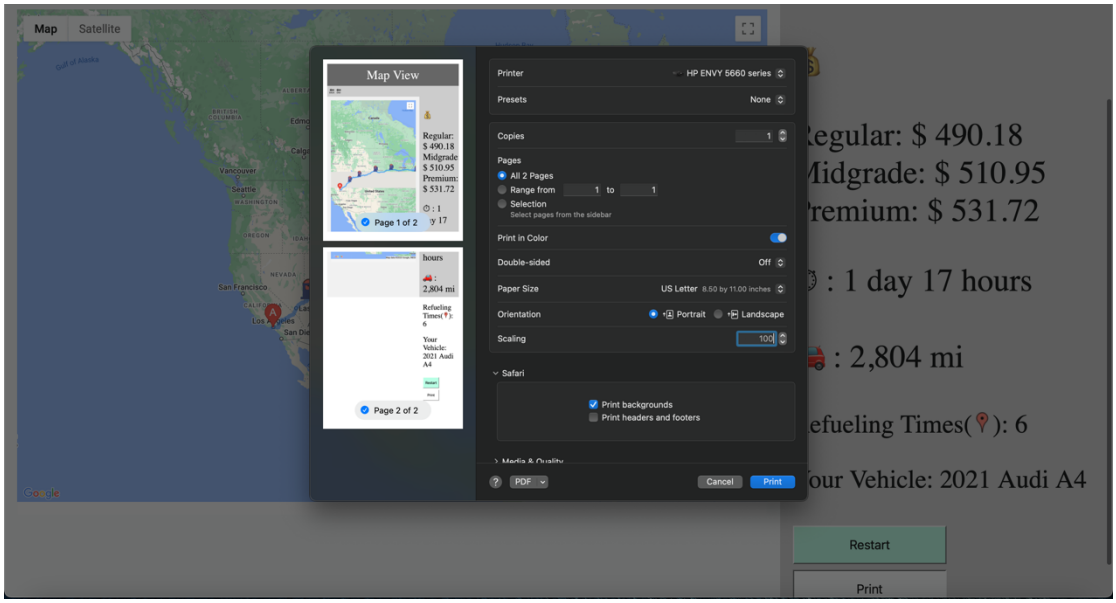

Figure 11c: Map Information Page (Print page)

13.0 Test and Integration

The test environment of the website is to test the runnability of the website files through the built-in function of GitHub Page. Then use any browser to visit the website and verify and proofread the website output.

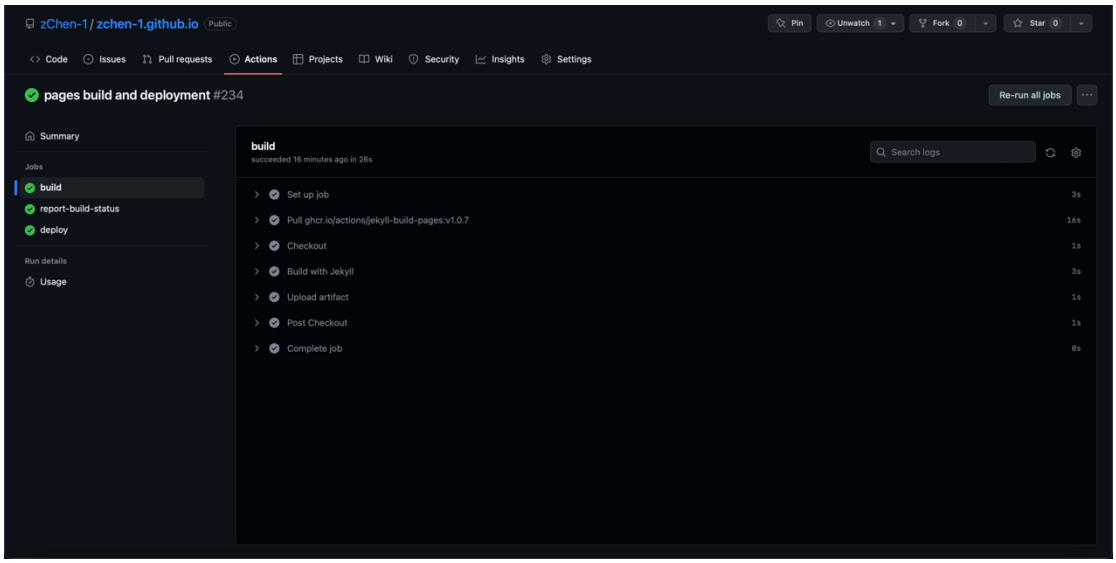

Figure 12a: GitHub window

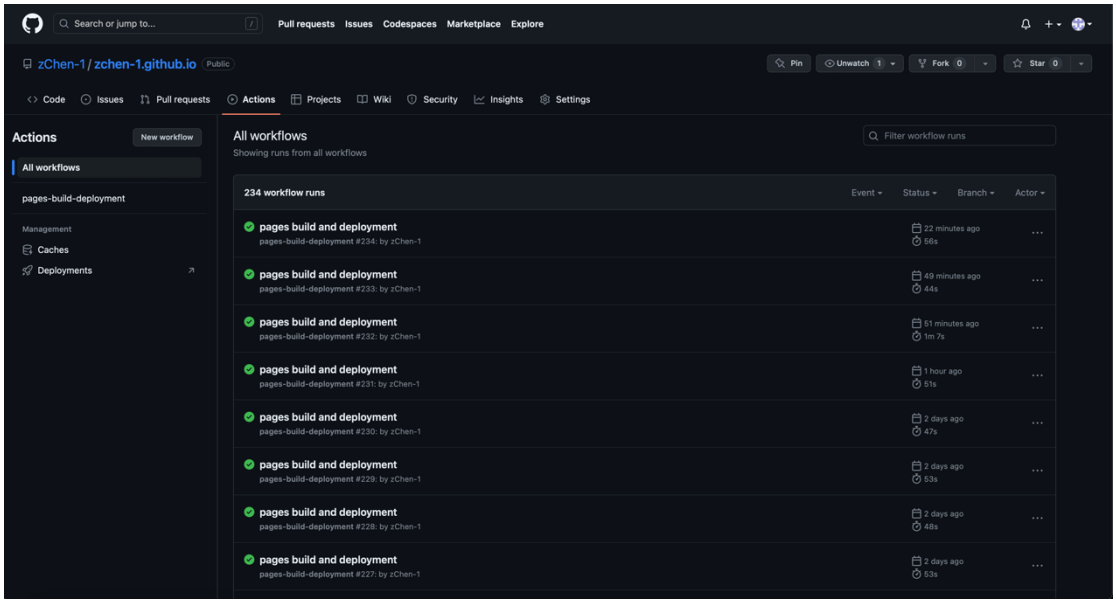

Figure 12b: GitHub window28

14. Enhancement suggestions

Future improvements will include more features, e.g., Complete the accounting system, database and import data for machine learning. Data collection and cleaning have been done in terms of predicting MPG using machine learning algorithms. Algorithmic models and codes have also been completed. However, the next step will be changing the static to a dynamic website. Thereby implementing a web page that can access the database. In general, the project's original intention is to provide users with a tool to help plan travel.

15.0 References

Lowe, J., Butler, J., & Luu, L. (2019). Essential Personal Finance: A Practical Guide for Employees (1st ed.). Routledge. https://doi-org.lib-proxy.fullerton.edu/10.4324/9781351041669

Napolitano, C. M., & Freund, A. M. (2016). On the Use and Usefulness of Backup Plans. Perspectives on Psychological Science, 11(1), 56–73. http://www.jstor.org/stable/26358545

Visconti, A. (2016, April 18). Building a static website with Jekyll and GitHub Pages. Programming Historian. https://programminghistorian.org/en/lessons/building-static-sites-with-jekyll-github-pages