

Skoolie Quest

Overview

- A. Skoolie refers to a converted bus or recreational vehicle (RV). Skoolie Quest is a social platform for Skoolie enthusiasts. Creating a profile on the website allows users to post their blogs, share their favorite campsite locations, and engage with the Skoolie community.
- B. Skoolie Quest Features:
 - a. A community page for sharing blogs, interesting articles, etc.
 - b. A map section for locating campsites you can park your Skoolie.
 - c. Build section for resources to convert your Skoolie.
- C. Tools and Technologies:
 - a. HTML & CSS
 - b. JavaScript
 - c. Google Maps JavaScript API
 - d. iOverlander API
 - e. Recreation.Gov API
 - f. Cloud Firestore
- D. Goals:
 - a. Craft a user-friendly experience. Ensure the platform looks great and works seamlessly on all devices, from smartphones to desktops.
 - b. Incorporate essential web features. Integrate a variety of web development elements to enhance functionality and user experience.
 - c. Leverage third-party APIs and backend cloud services. Utilize external APIs and backend cloud services to bring in valuable data and features.
 - d. Create an engaging interface. Design a visually appealing, intuitive, and user-friendly application that keeps users engaged and excited.

Responsive Design

- A. Media Queries
 - 1. Use CSS media queries to adjust styles based on screen size and orientation.

B. Reusable Components

1. Create reusable HTML components for consistency

C. CSS Grid & CSS Flexbox

1. Implement CSS grid for complex layouts that require precise positioning and alignment.
2. Utilize CSS flexbox for creating flexible and responsive layout structures.

Features

A. JavaScript Fetch API

1. Use JavaScript Fetch API to parse through a JSON object.
2. There are icons on the map to denote campsites. When interacting with these icons an information card is compiled. This will display campsite information: location, description, and other information.

B. Third Party API

1. Embed a dynamic map using the google maps JavaScript API. Utilize JavaScript functions and methods provided by the API to display and interact with the map.
2. Retrieve data using the iOverlander API and Recreation.Gov API to fetch data on camping spots. Create HTTP request to API to pull existing camping location data and display it on the map.

C. Database

1. Store submitted data from maps and user information in Cloud Firestore under a collection database.

D. Word Counter

1. Ensure blog posts adhere to a maximum character limit. Using an HTML structure for text and an element to display the character count.
2. Display a real-time character count to the user as they type and update the character count dynamically to provide immediate feedback. Using JavaScript real time count.

Development

A. Structure

1. Utilize specific IDs and classes to structure the HTML framework effectively.

2. Design the layout using Balsamiq, and then implement it using CSS grid and Flexbox for a responsive and flexible design

B. Interactivity

1. Leverage various JavaScript frameworks from multiple sources to enhance user interface.
2. Implement advanced interactivity and efficient state management to provide a seamless user experience.

C. Database

1. Utilize Cloud Firestore to store and manage data collected from users.
2. Implement fetch features to retrieve data and integrate it within the map functionality.

Review

- A. Create and test prototypes multiple times to ensure functionality and user experience.
- B. Utilize Visual Studio Codes debugging features to thoroughly test the webpage and review the code for any issues.
- C. Comment and organize code for better readability
- D. Seek feedback from peers and mentors throughout the development process to improve the website.

Documentation and Submission

- A. Create and maintain GitHub repository. Regularly commit changes for better version control and review process.
- B. Comprehensive Readme file to demonstrate a detailed project outline.
- C. Ensure the project meets all requirements and is ready for final submission and presentation.