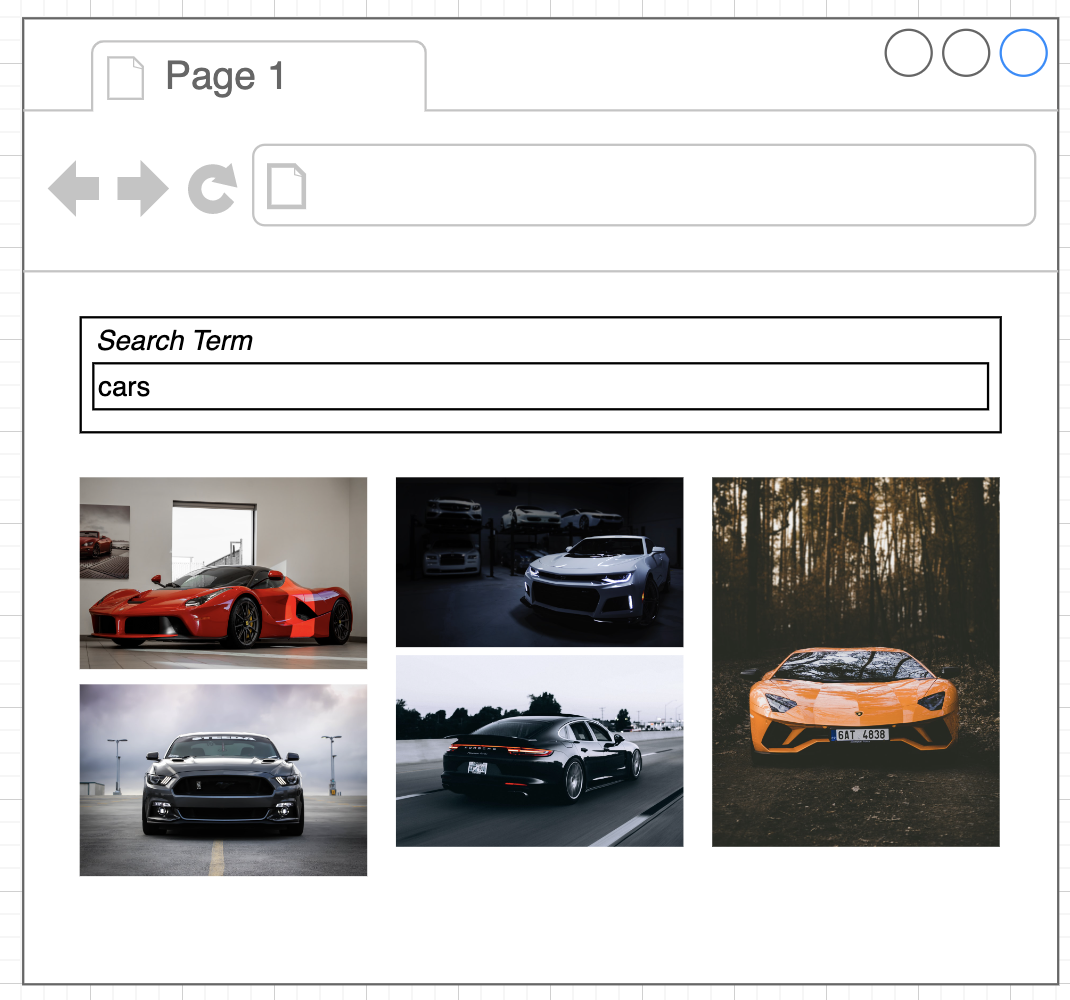
**Assignment 1**



Graphical user interface, application

Description automatically generated with medium confidence

* Create a Search box where user can search for something and corresponding images are visible below the search box. As shown above.
* When individual images are loading, it shows some loading icon in place of that image.
* When no images found corresponding to a search, then a message should display that ‘no images found for the given search’
* Images should be shown in the form of a grid like shown above.
* When user makes the input search field empty, it shows empty.
* Images list and searchTerm needs to be stored and fetched from React Context.
* Use ‘useReducer’ hook to manage states in React Context.
* Search should happen and images should be visible when you press Enter.

API URL: https://api.unsplash.com/search/photos

query: anything like – cars, bikes etc etc.

Authorization header: 'Client-ID 8O50V7bNzfKdVixwS9W9nZVdr0VnrCv9gmeimfdvp6Y'

**documentation**

<https://unsplash.com/documentation#search-photos>

Bonus:

* To limit the number of API calls, use debounce of 1.5 seconds while typing the query in the search box. Search should happen without pressing enter.
* Grid should be responsive.

**Assignment 2 (Priority)**

Design and implement a web application using React and Redux Toolkit that allows users to search for and save their favourite movies. The application should use Redux Toolkit to manage the state of the application, and Redux Toolkit Query to handle API requests for movie data.

The application should have the following features:

1. A search bar that allows users to search for movies by title, and displays the results in a list.
2. The ability to click on a movie from the search results to view more details, including the movie's title, release date, and poster image.
3. A "Favorites" page that displays a list of the user's saved movies.
4. The ability to add and remove movies from the "Favorites" page.
5. A "Watchlist" page that displays a list of movies the user has saved to watch later.
6. The ability to add and remove movies from the "Watchlist" page.

To complete the project, you should use Redux Toolkit to manage the state of the application, including the user's search query, the search results, and the user's saved movies and watchlist. You should use Redux Toolkit Query to handle API requests to retrieve movie data from a third-party API, such as the Open Movie Database (OMDb) API.

You should also use React to build the user interface for the application, and implement the required features using React components and hooks.

<https://www.omdbapi.com/>

http://www.omdbapi.com/?apikey=[yourkey]&