Image Search App

Retrieves Images from API on users queries

# Objective:

The task is to create a native mobile app for searching images using third-party APIs. The app should allow users to:

1. **Search Images**: Users can enter a search term in a search box to find images.
2. **View Image Details**: Users can click on an image to view it in full, using a shared element transition for a seamless experience. They should also be able to view metadata such as:

* User who uploaded the image
* Title of the image
* Number of downloads
* Number of views
* Number of likes

## Technical Specifications:

1. **Frontend**: The app should be built using React Native CLI.
2. **API Key**: A free account must be created with a third-party service to obtain an API key for making API calls to fetch images.

**REACT NATIVE CLI:**

React Native CLI (Command Line Interface) is a tool that provides a command-line interface for creating and managing React Native projects. It is an alternative to Expo CLI and offers more flexibility and customization options, making it suitable for developers who need to integrate native modules, access deeper configuration, or work with existing native code.

**Key Features and Capabilities of React Native CLI**

1. **Project Initialization**: You can create a new React Native project using the CLI with specific configurations, such as the project name and template.
2. **Development and Debugging**: The CLI provides commands to start the development server, build the app, and run it on Android and iOS simulators or physical devices. It also offers debugging options, including reloading the app and accessing the developer menu.
3. **Linking Native Modules**: The CLI allows you to link native modules (native code libraries) to your project, making it possible to use functionalities written in Java, Swift, or Objective-C.
4. **Platform-Specific Configuration**: It offers more control over platform-specific settings, such as modifying Android and iOS build configurations and integrating with native codebases.
5. **Custom Commands and Plugins**: You can create custom commands and extend the functionality of the CLI with plugins.

**Basic Commands**

* **npx react-native init ProjectName**: Initializes a new React Native project with the specified name.
* **npx react-native run-android**: Builds and runs the app on an Android emulator or device.
* **npx react-native run-ios**: Builds and runs the app on an iOS simulator or device.
* **npx react-native start**: Starts the Metro bundler, which serves the JavaScript code to the app.

#### PRE-REQUIRED TOOLS/DEVICES/TECHNOLOGY VERSIONS:

1. Laptop
2. CODE EDITOR (eg: VSCode)
3. Android Studio/Deivce
4. Java Development Kit(jdk) version 17
5. Third party API user account

### ROADMAP TO COMPLETE PROJECT

**1. Initial Setup**

**a. Project Initialization**

* **Install Node.js**: Ensure Node.js is installed on your machine.
* **React Native CLI**: Install the React Native CLI if not already installed.
  + - npm install -g react-native-cli
* **Create a New Project**
  + - npx react-native init YOURPROJECTNAME
* **Navigate to the Project Directory**
  + - cd YOURPROJECTNAME

**b. API Key Setup**

* **Choose an Image API**: Select a third-party service like Unsplash, Pixabay, or Pexels.
* **Create an Account**: Sign up for a free account and obtain your API key.

**2. Basic App Structure**

**a. Set Up Dependencies**

* **Install Required Packages**
  + - npm install axios react-navigation react-navigation-stack
    - npm install @react-navigation/native
    - npm install @react-navigation/stack
    - npm install react-native-gesture-handler react-native-reanimated
* **Setup Navigation**

Create a simple navigation stack with screens for Search and Image Details.

**b. UI Components**

* **Search Screen**:
  + A TextInput for entering search terms.
  + A Button for initiating the search.
  + A FlatList for displaying images.
* **Image Details Screen**:
  + Display the selected image in full view.
  + Show metadata like the Title, user, downloads, and views.

**3. API Integration**

**a. Axios Setup**

* **Create an API Service**:
  + Set up Axios for making API requests.
  + Include the API key in the headers or parameters.

**b. Fetch and Display Images**

* **Search Functionality**:
  + Fetch images based on the search term and selected filters.
  + Update state with the retrieved images and render them in the FlatList.
* **Image Details**:
  + Pass the selected image data to the Image Details screen and display it.

**4. Testing and Debugging (30 mins)**

**a. Testing**

* **Run the App**: Test the app on an Android emulator/device.
  + - npx react-native run-android #For Android
* **Check for Errors**: Ensure all features work correctly, and handle any errors or crashes.

**b. Refinement**

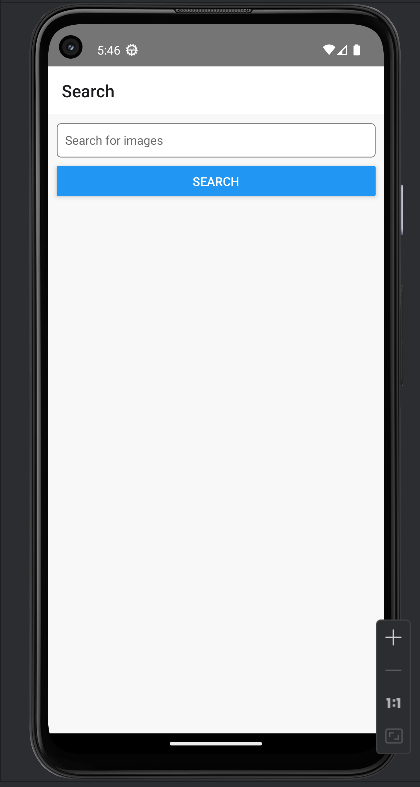
* **UI/UX Enhancements**: Make any necessary adjustments to improve the user experience.
* **Code Cleanup**: Remove any unnecessary code and add comments for clarity.

**5. Running the Project on Android**

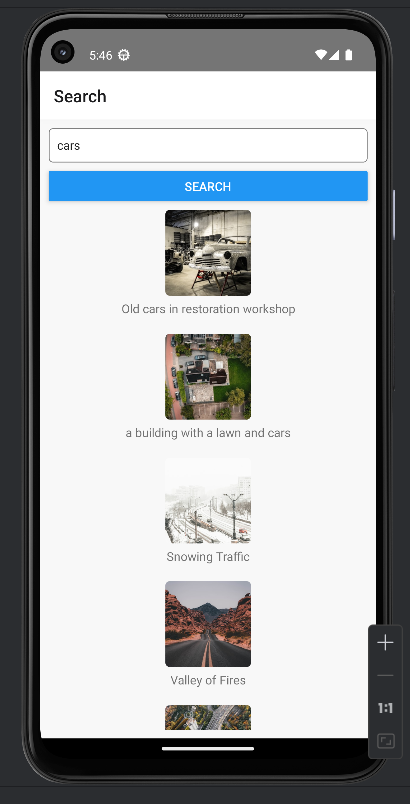
* **Start the Metro Bundler:** First, start the Metro bundler, which serves your JavaScript code:
  + - npx react-native start
* **Run the App:** In another terminal window, run the Android app:
  + - npx react-native run-android

This command compiles your React Native project for Android and installs the app on the emulator or connected device.

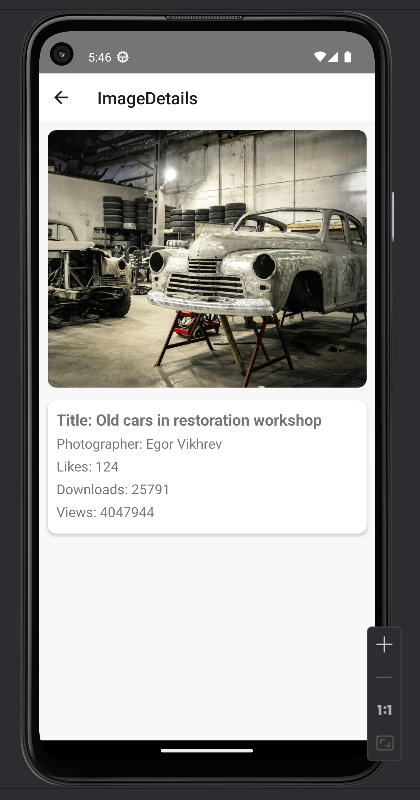
##### RESULTS:



**HOME SCREEN** of the App



**SEARCH SCREEN** of the APP



**IMAGE DETAILS SCREEN** of the APP