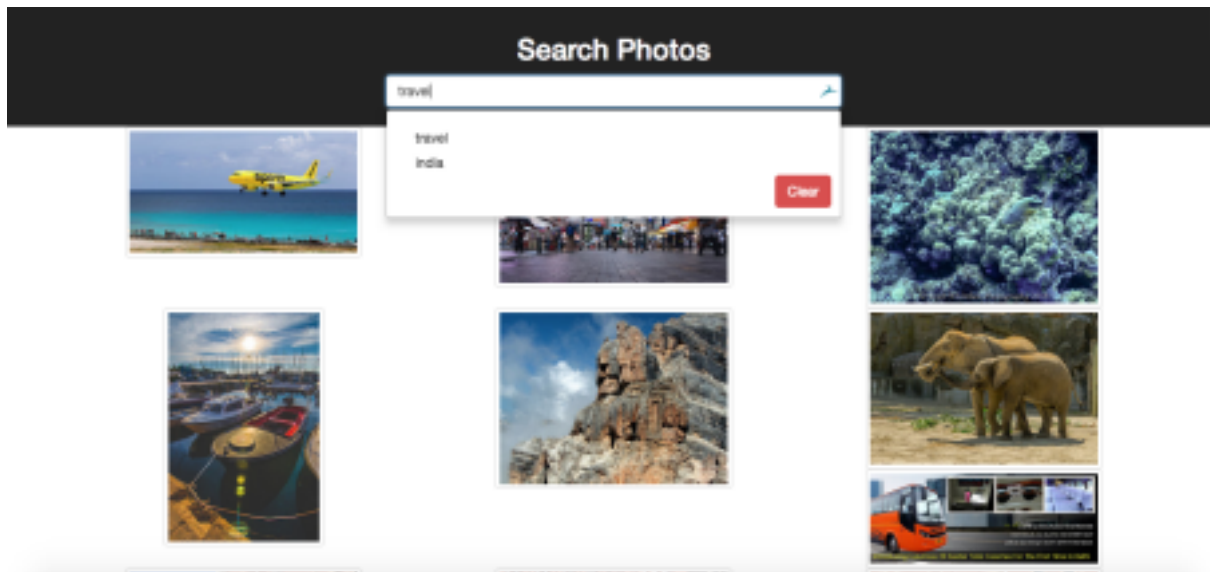




# ReactJS Assignment



Create a minimal react application as per the description and use the above interface as a reference:

## Requirements

1. **Default** - By default, show the images from <https://www.flickr.com/services/api/flickr.photos.getRecent.html>  
(NOTE - add `safe_search` value in query params (read reference) to avoid restricted images)
2. **Search** - A search bar should be placed in the header. When a user is **typing**, display results from <https://www.flickr.com/services/api/flickr.photos.search.html>
3. **Infinite Scroll** - As you scroll down to the bottom of the page, display more results if there are more results.
4. **Suggestions** - Save their search queries (in the browser itself) so that the next time they come back, you can suggest search queries (like as a list/tags near the search bar).
5. **Preview** - Clicking on a photo in the results will bring the photo up in a modal.

*Please ensure that you complete (almost) all the must-have requirements before submitting your code.*

## Bonus

- Make the search as efficient as possible with edge cases(0 results, More than x results )
- Use any UI framework e.g. bootstrap or material or antd(ant-design)



- Showing proper loaders and placeholders
- Page should be responsive
- Feel free to use any open-source libraries
- The search bar section should be fixed on top of the page results as you scroll down on the page
- Deploy your project on Heroku or Vercel (or any other) and share the deployed link.

## Resources

- We recommend using <https://github.com/facebook/create-react-app> but not mandatory if you're comfortable doing it any other way.
- You are free to use any open-source library or framework with react-js. You will also be judged based on your choice of libraries and patterns.

## What We're Looking for

- Code quality and style, good code should be self-documenting with comments when necessary
- Architectural decisions, keep it simple but extensible, not too much over-engineering - UX considerations, styles don't have to be polished, but they should be reasonable and usable
- Reasonable type checking and error handling
- Performance, minimizing network requests.

## Output

- Please commit to Github/Bitbucket public repo and deploy the application on a public link using any provider like heroku.com or netlify.com or Vercel or any other platform of your choice.