import React, { useState, useEffect } from 'react';

const App = () => {

const [searchQuery, setSearchQuery] = useState('');

const [images, setImages] = useState([]);

const [page, setPage] = useState(1);

const [suggestedQueries, setSuggestedQueries] = useState([]);

useEffect(() => {

fetchImages();

loadSuggestedQueries();

}, []);

const fetchImages = async () => {

try {

const response = await fetch(

`https://www.flickr.com/services/api/flickr.photos.getRecent.html?api\_key=YOUR\_API\_KEY&safe\_search=1&page=${page}&per\_page=20&format=json&nojsoncallback=1`

);

const data = await response.json();

setImages((prevImages) => [...prevImages, ...data.photos.photo]);

} catch (error) {

console.error('Error fetching images:', error);

}

};

const searchImages = async () => {

try {

const response = await fetch(

`https://www.flickr.com/services/api/flickr.photos.search.html?api\_key=YOUR\_API\_KEY&safe\_search=1&text=${searchQuery}&page=${page}&per\_page=20&format=json&nojsoncallback=1`

);

const data = await response.json();

setImages(data.photos.photo);

} catch (error) {

console.error('Error searching images:', error);

}

};

const loadSuggestedQueries = () => {

const savedQueries = localStorage.getItem('searchQueries');

if (savedQueries) {

setSuggestedQueries(JSON.parse(savedQueries));

}

};

const saveSearchQuery = () => {

const updatedQueries = [...suggestedQueries, searchQuery];

setSuggestedQueries(updatedQueries);

localStorage.setItem('searchQueries', JSON.stringify(updatedQueries));

};

const handleSearch = () => {

setPage(1);

searchImages();

saveSearchQuery();

};

const handleLoadMore = () => {

setPage((prevPage) => prevPage + 1);

};

return (

<div>

<header>

<input

type="text"

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

placeholder="Search for images"

/>

<button onClick={handleSearch}>Search</button>

</header>

<div>

{images.map((image) => (

<img

key={image.id}

src={`https://live.staticflickr.com/${image.server}/${image.id}\_${image.secret}\_m.jpg`}

alt={image.title}

/>

))}

</div>

<button onClick={handleLoadMore}>Load More</button>

<div>

<h3>Suggested Queries:</h3>

<ul>

{suggestedQueries.map((query, index) => (

<li key={index}>{query}</li>

))}

</ul>

</div>

</div>

);

};

export default App;

continuation of code

import React, { useState, useEffect } from 'react';

const App = () => {

const [searchQuery, setSearchQuery] = useState('');

const [images, setImages] = useState([]);

const [page, setPage] = useState(1);

const [suggestedQueries, setSuggestedQueries] = useState([]);

const [selectedImage, setSelectedImage] = useState(null);

useEffect(() => {

fetchImages();

loadSuggestedQueries();

}, []);

const fetchImages = async () => {

try {

const response = await fetch(

`https://www.flickr.com/services/api/flickr.photos.getRecent.html?api\_key=YOUR\_API\_KEY&safe\_search=1&page=${page}&per\_page=20&format=json&nojsoncallback=1`

);

const data = await response.json();

setImages((prevImages) => [...prevImages, ...data.photos.photo]);

} catch (error) {

console.error('Error fetching images:', error);

}

};

const searchImages = async () => {

try {

const response = await fetch(

`https://www.flickr.com/services/api/flickr.photos.search.html?api\_key=YOUR\_API\_KEY&safe\_search=1&text=${searchQuery}&page=${page}&per\_page=20&format=json&nojsoncallback=1`

);

const data = await response.json();

setImages(data.photos.photo);

} catch (error) {

console.error('Error searching images:', error);

}

};

const loadSuggestedQueries = () => {

const savedQueries = localStorage.getItem('searchQueries');

if (savedQueries) {

setSuggestedQueries(JSON.parse(savedQueries));

}

};

const saveSearchQuery = () => {

const updatedQueries = [...suggestedQueries, searchQuery];

setSuggestedQueries(updatedQueries);

localStorage.setItem('searchQueries', JSON.stringify(updatedQueries));

};

const handleSearch = () => {

setPage(1);

searchImages();

saveSearchQuery();

};

const handleLoadMore = () => {

setPage((prevPage) => prevPage + 1);

};

const handleImageClick = (image) => {

setSelectedImage(image);

};

const handleCloseModal = () => {

setSelectedImage(null);

};

return (

<div>

<header>

<input

type="text"

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

placeholder="Search for images"

/>

<button onClick={handleSearch}>Search</button>

</header>

<div>

{images.map((image) => (

<img

key={image.id}

src={`https://live.staticflickr.com/${image.server}/${image.id}\_${image.secret}\_m.jpg`}

alt={image.title}

onClick={() => handleImageClick(image)}

/>

))}

</div>

<button onClick={handleLoadMore}>Load More</button>

<div>

<h3>Suggested Queries:</h3>

<ul>

{suggestedQueries.map((query, index) => (

<li key={index}>{query}</li>

))}

</ul>

</div>

{selectedImage && (

<div className="modal">

<img

src={`https://live.staticflickr.com/${selectedImage.server}/${selectedImage.id}\_${selectedImage.secret}\_b.jpg`}

alt={selectedImage.title}

/>

<button onClick={handleCloseModal}>Close</button>

</div>

)}

</div>

);

};

export default App;

In this continuation, I've added the functionality to display a modal when an image is clicked. The modal displays the selected image in a larger size and includes a "Close" button to close the modal.

Please note that you'll need to add CSS styles to properly position and style the modal. You can customize the modal's appearance based on your design preferences.

Remember to replace YOUR\_API\_KEY with your actual Flickr API key.

I hope this helps! Let me know if you have any further questions.

import React, { useState, useEffect } from 'react';

const App = () => {

const [searchQuery, setSearchQuery] = useState('');

const [images, setImages] = useState([]);

const [page, setPage] = useState(1);

const [suggestedQueries, setSuggestedQueries] = useState([]);

const [selectedImage, setSelectedImage] = useState(null);

useEffect(() => {

fetchImages();

loadSuggestedQueries();

}, []);

const fetchImages = async () => {

try {

const response = await fetch(

`https://www.flickr.com/services/api/flickr.photos.getRecent.html?api\_key=YOUR\_API\_KEY&safe\_search=1&page=${page}&per\_page=20&format=json&nojsoncallback=1`

);

const data = await response.json();

setImages((prevImages) => [...prevImages, ...data.photos.photo]);

} catch (error) {

console.error('Error fetching images:', error);

}

};

const searchImages = async () => {

try {

const response = await fetch(

`https://www.flickr.com/services/api/flickr.photos.search.html?api\_key=YOUR\_API\_KEY&safe\_search=1&text=${searchQuery}&page=${page}&per\_page=20&format=json&nojsoncallback=1`

);

const data = await response.json();

setImages(data.photos.photo);

} catch (error) {

console.error('Error searching images:', error);

}

};

const loadSuggestedQueries = () => {

const savedQueries = localStorage.getItem('searchQueries');

if (savedQueries) {

setSuggestedQueries(JSON.parse(savedQueries));

}

};

const saveSearchQuery = () => {

const updatedQueries = [...suggestedQueries, searchQuery];

setSuggestedQueries(updatedQueries);

localStorage.setItem('searchQueries', JSON.stringify(updatedQueries));

};

const handleSearch = () => {

setPage(1);

searchImages();

saveSearchQuery();

};

const handleLoadMore = () => {

setPage((prevPage) => prevPage + 1);

};

const handleImageClick = (image) => {

setSelectedImage(image);

};

const handleCloseModal = () => {

setSelectedImage(null);

};

return (

<div>

<header>

<input

type="text"

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

placeholder="Search for images"

/>

<button onClick={handleSearch}>Search</button>

</header>

<div>

{images.map((image) => (

<img

key={image.id}

src={`https://live.staticflickr.com/${image.server}/${image.id}\_${image.secret}\_m.jpg`}

alt={image.title}

onClick={() => handleImageClick(image)}

/>

))}

</div>

<button onClick={handleLoadMore}>Load More</button>

<div>

<h3>Suggested Queries:</h3>

<ul>

{suggestedQueries.map((query, index) => (

<li key={index}>{query}</li>

))}

</ul>

</div>

{selectedImage && (

<div className="modal">

<img

src={`https://live.staticflickr.com/${selectedImage.server}/${selectedImage.id}\_${selectedImage.secret}\_b.jpg`}

alt={selectedImage.title}

/>

<button onClick={handleCloseModal}>Close</button>

</div>

)}

</div>

);

};

export default App;

In this continuation, I've added the functionality to display a modal when an image is clicked. The modal displays the selected image in a larger size and includes a "Close" button to close the modal.

Please note that you'll need to add CSS styles to properly position and style the modal. You can customize the modal's appearance based on your design preferences.

Remember to replace YOUR\_API\_KEY with your actual Flickr API key.

I hope this helps! Let me know if you have any further questions.

Key

**5b993c18c116726e11453061b40fcc56**

**Secret key**

**fa8c6b672b300721**