Source code

INdex.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { BrowserRouter } from 'react-router-dom';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
```

App.js

```
import {Heading} from './components/Heading';
import './App.css';
import Main from './components/main';
import './components/mainstyle.css';
import videobg from './img/livewallPokemon.mp4'

function App() {
  //destrucutirng the array
  // const[pokemon, setPokemon] = useState(["bulbasaur", "charmander"])
  //the two above names are default
```

```
 // setpokemon will set the values
  return (
   <div className='backdrop'>
    <video src={videobg} autoPlay loop muted className='video-bg' type='video/mp4' />
    <div className="pokemonFinder">
     <Heading />
     <div>
      <Main/>
     </div>
     <div className="background-cover"></div>
    </div>
   </div>
  );
 }

export default App
```

App.cs

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;300;400;700&display=swap');
*
{
  box-sizing: border-box;
}
body
{
  /*background-image: linear-gradient( 0deg, var( --bg-color ), var( --bg-color ) ), url( './img/pokepika2.png' ); */
  margin: 0;
  padding: 0;
  font-family: Poppins, sans-serif;
```

```
    display: flex;

    align-items: center;

    justify-content: center;

    height: 100vh;

    overflow: hidden;

}


.backdrop{

    position: absolute;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    overflow: hidden;

}

.video-bg{

    position: absolute;

    top: 40%;

    left: 50%;

    transform: translate(-50%, -50%) scale(0.8);

    min-width: 100%;

    min-height: 100%;

    object-fit: cover;

    z-index: -1;

}


.pokemonFinder

{

    padding: 1rem;

    border-radius: 1rem;

    position: relative;
    z-index: 1;
    background: rgba(255, 255, 255, 0.2);
    max-height: 100vh;
    overflow-y: auto;
}

/*
    Background Pokeball decoration
*/
.bg-pokeball
{
    position: absolute;
    display: flex;
    align-items: center;
    justify-content: center;
    right: -10px;
    bottom: -10px;
    font-size: 10px;
    color: rgba( 255,255,255,.2 );
}

.bg-pokeball::before, .bg-pokeball::after
{
    display: block;
    border-radius: 100%;
}

.bg-pokeball::before
{
    content: '';
    position: relative;
    width: 10em;

    padding-bottom: 10em;

    border: 4em solid currentColor;

    clip-path: polygon( 0 0, 0 40%, 50% 40%, 50% 60%, 0 60%, 0 100%, 100% 100%, 100% 60%, 50% 60%, 50% 40%, 100% 40%, 100% 0 );

}


.bg-pokeball::after

{

    content: '';

    position: absolute;

    width: 5em;

    padding-bottom: 5em;

    background-color: currentColor;

}


.background-cover {

    height: 100vh;

}
```

Components

Card

card.jsx

```
import React, { Component ,useState } from 'react';

import './card.css';

import Overlay from '../PokeDetails/pokeinfo';
```

```
export default function Card({ pokemon, loading}) {

    const [selectedPokemon, setSelectedPokemon] = useState(null);

    const handleCardClick = (pokemon) => {
      setSelectedPokemon(pokemon);
    };

    const handleCloseOverlay = () => {
      setSelectedPokemon(null);
    };


    console.log(pokemon);
    return (
        <div className="card-container">
        {loading ? (
          <h1>Loading...</h1>
        ) : (
            <div className="card-wrapper">{
                pokemon.map((item) => (

                    <div className={`card ${item.name} type-${item.types[0].type.name}`} key={item.id} onClick={() => handleCardClick(item)}>
                        <div className="bg-pokeball"></div>
                        <span className="pokemon-id">{`# ${item.id}`}</span>

                        <div className="card-title">
                            <h2>{item.name}</h2>
                        </div>


                        <div className="pokemon-image">
                            <img alt={item.name} src={item.sprites.other["official-artwork"].front_default} />
                        </div>


                    </div>
                ))}
            </div>
        )}
        <Overlay pokemon={selectedPokemon} onClose={handleCloseOverlay} />
     </div>
    );
 }
```

Card.css

```
@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=swap');


*{


   font-family: 'Roboto', sans-serif;
   justify-content: center; /* Horizontally center content */
   align-items: center; /* Vertically center content */
   margin: 0;
}
```

```
.card-container
{
    padding: 3rem;
    height: 100%;
    cursor: pointer;

}

.card-wrapper {
    display: flex;
    flex-wrap: wrap;
    gap: 2.2rem; /* Adjust the gap between cards */
    padding-right: 1.2rem;
    padding-left: 1.2rem;
    margin: 0 auto;

    max-width: 1200px; /* Set a maximum width for the cards */
    width: 100%; /* Ensure it takes full available width */
    padding: 0 1.2rem;
  }

.card
{
    width: calc(50% - 3rem);
    background-color: var( --color );
    color: #FFF;
    box-shadow: 0 0 20px 0 var( --color );
    border-radius: 3rem;
    overflow: hidden;
    height: 28rem;
    padding: 3rem 1rem 3rem 3rem;
    position: relative;

    display: flex;

    align-items: center;

    justify-content: space-between;

    will-change: box-shadow, transform;

    transition: .3s all;

}


.card .card-title

{

    margin: 0;

    font-size: 1.5rem;

}


.card .card-title h2

{

    text-transform: capitalize;

}





.card .pokemon-image

{

    align-self:flex-end;

    display: flex;

    align-items: center;

    justify-content: flex-end;

    max-width: 100%;

    max-height: 24rem;

}
```

```
.pokemon-image img {
     /* Adjust this value to make the image larger */
    height: auto;
    width: auto;
  }



.card img
{
    max-width: 100%;
    max-height: 100%;
    filter: drop-shadow( 0 0 5px rgba( 0,0,0,.2 ) );
}

.pokemon-id
{
    position: absolute;
    top: 20px;
    right: 40px;
    font-size: 2.5rem;
    color: rgba( 255,255,255,.2 );
}

@media screen and ( max-width: 767px )
{
    :not( .details-view ) .card
    {
        align-items: flex-start;
        justify-content: space-between;
        height: 40vh;
        padding: 2rem;

        width: 100%;

    }


    .card .card-title h2

    {

        font-size: 1.2rem;

    }


    .pokemon-id

    {

        font-size: 2rem;

        top: 20px;

        right: 20px;

    }

}


@media screen and ( min-width: 1024px )

{

    .card:hover

    {

        transform: translateY( -5px );

        box-shadow: 0 0 20px 5px var( --color );

    }

}





/* Pokemon type card background colors */

.type-normal { --color: #C2C2A1; }
```

```
.type-flying { --color: #BAB0D5; }

.type-ghost { --color: #735797; }

.type-dark { --color: #333; }

.type-steel { --color: #CCCCDE; }

.type-ground { --color: #B1736C; }

.type-poison { --color: #7C538C; }

.type-grass { --color: #48D0B0; }

.type-fire { --color: #FB6C6C; }

.type-electric { --color: #FFD86F; }

.type-fairy { --color: #f469a9; }

.type-bug { --color: #C3CE75; }

.type-fighting { --color: #d6b591; }

.type-water { --color: #609FB5; }

.type-psychic { --color: #9B7FA6; }

.type-ice { --color: #7FCCEC; }
```

```
.type-rock { --color: #a6aab6; }


.type-dragon { --color: #F9BE00; }
```

Components- Heading

index.js

```
export { default as Heading } from './Heading';
```

Headings

```
import './Heading.css';
import Pokelogo from '../../img/pokemon_logo.png';
import React from 'react';
import Lottie from 'react-lottie';
import pokelogoAnimation from '../../img/pokeball_animation.json'; // Adjust the path



export default function Heading({ pokeData }) {


    const defaultOptions = {
        loop: true,
        autoplay: true,
        animationData: pokelogoAnimation,
      };
    return (
  <div className="heading-container"id="html">
        <div className="pokelogo-container">
            <img src={Pokelogo} alt="Pokemon Logo" className="pokelogo" />
        </div>
        <div className="lottie-animation">
            <Lottie options={defaultOptions} height={100} width={100} />
        </div>

    </div>
    );
}
```

heading.css

```
.heading-container {
    display: flex;
    align-items: center;
    justify-content: center;

  }

  .pokelogo-container {
    display: flex;
    justify-content: center;
    align-items: center;
  }

  .pokelogo {
    /* max-width: 90; */
    height: auto;
    width: 70%;

  }

  .lottie-animation {
    display: flex;
    align-items: center;
```

```
 }


 .lottie-animation .lottie-container {
  max-width: 80%;
 }
```

Components-Pokeinfo

```
import React from "react";
import './pokeinfo.css';

const Overlay = ({ pokemon, onClose }) =>{
  if (!pokemon) return null;

 return (
  <div>
    <div className="overlay">
      <div className={`overlay-content type-${pokemon.types[0].type.name}`}>
        <div className="overlay-header">
            <button onClick={onClose} className="close-button">
            X
            </button>
        </div>
        <div className="overlay-body">

          <img src={pokemon.sprites.other.dream_world.front_default} alt="pokemon"></img>
          <h1>{pokemon.name}</h1>
          <div className="info">
            <table>
              <tr>
  <th>
    <h2>Abilities</h2>
  </th>
  <td>
    <h3>
      {pokemon.abilities[0].ability.name}
      {pokemon.abilities[1] && `, ${pokemon.abilities[1].ability.name}`}
    </h3>
  </td>
</tr>
<tr>
  <th>
    <h2>Height</h2>
  </th>
  <td>
    <h3>{pokemon.height}</h3>
  </td>
</tr>
<tr>
  <th>
    <h2>Weight</h2>
  </th>
  <td>
    <h3>{pokemon.weight}</h3>
  </td>
</tr>
<tr>
  <th>
    <h2>Species</h2>
  </th>
  <td>
```

```
                  <h3>{pokemon.species.name}</h3>
                </td>
              </tr>
            </table>
          </div>



        </div>
      </div>
    </div>
  </div>

 )
}
export { Overlay as default }; // Named export with alias 'default'
```

Pokeinfo.css

```
.overlay {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(92, 92, 92, 0.8);
  display: flex;
  align-items: center;
  justify-content: center;
  z-index: 1000;
 }

 .overlay-content {
  background-color: var( --color );

  padding: 30px;

  border-radius: 3rem;

  max-width: 80%;

  overflow: auto;

  /* background: rgba(47, 47, 47, 1);  */

  will-change:  transform;

  transition: .3s all;

}


.overlay-header {

  display: flex;

  justify-content: space-between;

}




.close-button {

  background: #f64a4a;

  border:white;

  padding: 5px 20px;

  color: white;

  border-radius: 5px;

  cursor: pointer;

  margin-left: auto;

}


.close-button:hover {

  background: #e74c3c;

}
```

```
.overlay-body {

  margin-top: 5px;

  color: white;

  text-transform: capitalize;

  text-align: center;

}


.overlay-body img {

  height: 300px;

  width: 300px;

  max-width: 100%;

  display: block;

  margin:0px 40px;


}


.overlay-body h1 {

  font-size: 30px;

  display: block;

  margin: 0;

  font-weight: 1000;

}


.info {

  margin-top: 20px;

  margin-bottom: 10px;

}


.info table{

  border-collapse: collapse;

  width: 70%;
  margin: 0 auto;
}

.info th, .info td {
  padding: 10px; /* Add padding for space between columns */
  text-align: left; /* Align text to the left within cells */
}

.info h2 {
  margin: 0;
  font-size: 18px;
}

.info h3 {
  margin: 0;
  font-size: 16px;
}



@media screen and ( min-width: 1024px )
{
    .overlay-content:hover
    {
        transform: translateY( -4px );
        box-shadow: 0 0 15px 5px var( --color );
    }
}


/*
```

```
   Pokemon type colors

*/

.type-normal { --color: #C2C2A1; }

.type-flying { --color: #BAB0D5; }

.type-ghost { --color: #735797; }

.type-dark { --color: #333; }

.type-steel { --color: #CCCCDE; }

.type-ground { --color: #B1736C; }

.type-poison { --color: #7C538C; }

.type-grass { --color: #48D0B0; }

.type-fire { --color: #FB6C6C; }

.type-electric { --color: #FFD86F; }

.type-fairy { --color: #f469a9; }

.type-bug { --color: #C3CE75; }

.type-fighting { --color: #d6b591; }

.type-water { --color: #609FB5; }

.type-psychic { --color: #9B7FA6; }
```

```
.type-ice { --color: #7FCCEC; }

.type-rock { --color: #a6aab6; }

.type-dragon { --color: #F9BE00; }
```

Components-Searchar.js

```
import React, { useState } from 'react';
import './searchbar.css';

const SearchBar = ({ onSearch }) => {
  const [searchQuery, setSearchQuery] = useState('');

  const handleInputChange = (e) => {
    const query = e.target.value;
    setSearchQuery(query);
    // Call the onSearch function to pass the search query to the parent component
    onSearch(query);
  };

  return (
    <input
      type="text"
      placeholder="Catch Pokemon"
      value={searchQuery}
      onChange={handleInputChange}
    />
  );
};
```

```
export default SearchBar;
```

Components-main.jsx

```
import React, { useEffect, useState } from "react";
import Card from './Cards/card';
import SearchBar from "./Searchbar/searchbar";

const Main=()=>{

  const [pokeData, setPokeData]=useState([]);
  const [loading,setLoading]=useState(true);
  const [url,setUrl]=useState("https://pokeapi.co/api/v2/pokemon/")
  const [nextUrl,setNextUrl]=useState();
  const [prevUrl,setPrevUrl] =useState();
  //search result
  const [searchResults, setSearchResults] = useState([]);

  //function of fetching pokemon via api

  const fetchData = async () => {
    setLoading(true);
    try {
     const response = await fetch(url);
     if (!response.ok) {
      throw new Error('Network response was not ok');
     }
     const data = await response.json();
```

```
      //console.log(data.results);

      setNextUrl(data.next);
      setPrevUrl(data.previous);
    //   const pokemonUrls = await getPokemon(data.results)
    //   console.log(pokemonUrls)
      getPokemon(data.results)
      setLoading(false);
      //console.log(pokeData)


    } catch (error) {
      console.error('Error fetching data:', error);
    }
    setLoading(false);
  };


  //function to get pokemon
//   const getPokemon = async(response)=>{
//     response.map(async(item)=>{
//         const resultResponse = await fetch(item.url);
//         //item.url will only return a array of urls we need an array of arrays
//         const results = await resultResponse.json().url;
//         console.log(results);
//     })
//   }
const getPokemon = async (results) => {
    try {
      const pokemonDataArray = await Promise.all(results.map(async (item) => {
        const response = await fetch(item.url);
    if (!response.ok) {
     throw new Error('Network response was not ok');
    }
    const pokemonData = await response.json();
    return pokemonData;
   }));
   //console.log('Pokemon Data:', pokemonDataArray);

   setPokeData(prevState => [...prevState, ...pokemonDataArray]);

  }

  catch (error) {
   console.error('Error fetching data:', error);
  }
 };

const handleSearch = async (query) => {
 // Perform search based on the query
 if (query.trim() === '') {
  setSearchResults([]);
  return;
 }


 const searchUrl = `https://pokeapi.co/api/v2/pokemon/${query.toLowerCase()}`;
  setLoading(true);
  try {
   const response = await fetch(searchUrl);
   if (!response.ok) {
    throw new Error('Network response was not ok');
   }
   const data = await response.json();
   setSearchResults([data]);
   setLoading(false);
  } catch (error) {
   console.error('Error fetching data:', error);
   setLoading(false);
  }
};

useEffect(() => {
 const fetchDataAndHandleWarning = async () => {
  await fetchData();
 };
 fetchDataAndHandleWarning();
 // eslint-disable-next-line react-hooks/exhaustive-deps
}, [url]);

//runs whenever the application renders
//if we run empty array runs when the page is laoded, when value changes the page is reloaded

return(
 <div>
 <SearchBar onSearch={handleSearch} />
 <div>
  {loading ? (
   <div>Loading...</div>
  ) : (

   <Card pokemon={searchResults.length > 0 ? searchResults : pokeData} loading={loading} />
  )}
```

```
    {/* <Card pokemon={pokeData} loading={loading} /> */}
    <div className="btn-group">
     <button className={`buttonControl ${prevUrl ? '' : 'disabled'}`} onClick={()=>{setPokeData([])
              setUrl(prevUrl)}} >Previous</button>

     <button className={`buttonControl ${nextUrl ? '' : 'disabled'}`} onClick={()=>{setPokeData([])
       setUrl(nextUrl)}} >Next</button>
    </div>
   </div>
  </div>
  )
}

export default Main;
```