

```
'use client';

import Link from 'next/link';

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

import LoadingSpinner from '@shared/components/loading-spinner';

import Input from '@shared/components/ui/Input';

import { useToast } from '@shared/components/ui/Toasts/use-toast';

import useToggle from '@shared/hooks/toggle';

import { cn } from '@shared/utils/cn';

import { trpc } from '@shared/utils/trpc/trpc';

import { useOnClickOutside } from '@shared/hooks/onclick-outside';

import { createQueryString } from '@shared/utils/helpers';

import { useRouter } from 'next/navigation';

import { PLATFORM } from '@shared/constants/links';
```

```
export default function NavbarSearch() {

  const searchRef = useRef(null);

  const toast = useToast();

  const router = useRouter();

  const [search, setSearch] = useState("");

  const [allData, setAllData] = useState<
    Record<string, { title: string; link: string; type: string }[]>
    >({});

  const [filteredData, setFilteredData] = useState<
    Record<string, { title: string; link: string; type: string }[]>
    >({});
```

```
const globalMutation = trpc.main.globalSearch.useMutation();
```

```
const { isOn, setOn, setOff } = useToggle();
```

```
useOnClickOutside(searchRef, setOff);
```

```
useEffect(() => {  
  globalMutation  
    .mutateAsync()  
    .then((res) => {  
      setAllData(res.data);  
      setFilteredData(res.data);  
    })  
    .catch((err: any) => {  
      console.error(err);  
      toast.toast({  
        description: 'Something went wrong',  
      });  
    });  
  });  
  // eslint-disable-next-line react-hooks/exhaustive-deps  
}, []);
```

```
useEffect(() => {  
  if (search) {  
    const filtered = Object.entries(allData).reduce(  
      (acc, [category, items]) => {  
        acc[category] = items.filter((item) =>
```

```
        item.title.toLowerCase().includes(search.toLowerCase()),

    );

    return acc;

  },

  {} as Record<string, { title: string; link: string; type: string }[]>,

);

setFilteredData(filtered);

} else {

  setFilteredData(allData);

}

}, [search, allData]);
```

```
const handleSearchChange = useCallback((value: string) => {

  setSearch(value);

}, []);
```

```
const handleMoreResults = (category: string) => {

  const params = createQueryString({

    category: category.toLowerCase(),

    search,

  });

  router.push(PLATFORM.EXPLORER + '?' + params);

  setOff();

};
```

```

const renderSearchResults = () => {

  return Object.entries(filteredData).map(([category, items]) => {

    const displayItems = items.slice(0, 6);

    const remainingItems = items.length - 6;

    return (

      <div key={category}>

        {displayItems.length > 0 && (

          <h2 className="p-2 py-5 h-7 flex mt-2 mb-3 first:mt-0 items-center text-base text-primary
border-b-slate-900 border-b font-bold bg-black/90 rounded-md shadow-4xl">

            {category}

          </h2>

        )}

        <ul>

          {displayItems.map((item) => (

            <li

              key={item.title}

              className="transition-custom mb-2 text-sm cursor-pointer overflow-hidden shadow-4xl
rounded-md   hover:shadow-5xl   hover:bg-red-500   hover:text-white   flex   items-center
hover:scale-[1.02]"

            >

              <Link href={item.link} className="p-2 py-3 w-full">

                {item.title}

              </Link>

            </li>

          )}}

        </ul>

      </div>
    )
  })
}

```

```
</ul>
```

```
{search && remainingItems > 0 && (
```

```
<p
```

```
onClick={() => handleMoreResults(category)}
```

```
className="text-sm text-gray-400 mt-2 cursor-pointer hover:text-red-300 hover:underline"
```

```
>
```

```
{remainingItems} more results
```

```
</p>
```

```
)}  

```

```
</div>
```

```
);
```

```
});
```

```
};
```

```
return (
```

```
<div ref={searchRef} className="w-full relative ml-10 mt-2 sm:mt-0 lg:ml-0">
```

```
<label hidden htmlFor="search">
```

```
Search
```

```
</label>
```

```
<div className="relative">
```

```
<Input
```

```
placeholder="Search swarms and more..."
```

```
id="search"
```

```
aria-label="Search"
```

```
onFocus={setOn}
```

```
onChange={handleSearchChange}
```

```

value={search}

className="w-full disabled:cursor-not-allowed disabled:opacity-50 text-white max-sm:text-xs
pr-11"

/>

<div

  className={cn(

    'absolute z-50 top-2/4 -translate-y-2/4 right-3 invisible text-white',

    globalMutation.isPending && 'visible',

  )}

>

  <LoadingSpinner />

</div>

</div>

<div

  className={cn(

    'absolute z-40 w-full h-[calc(100vh - 100px)] invisible',

    isOn && 'visible',

    globalMutation.isPending && 'invisible',

  )}

>

  <ul className="py-2 px-3 mt-1 h-[60vh] md:h-[65vh] no-scrollbar w-full overflow-y-auto
bg-secondary text-foreground border dark:bg-black dark:text-white rounded-md shadow-lg">

    {Object.values(filteredData).flat().length > 0 ? (

      renderSearchResults()

    ) : (

```

<p className="text-center py-4">No search result found</p>

)}

</div>

</div>

);

}