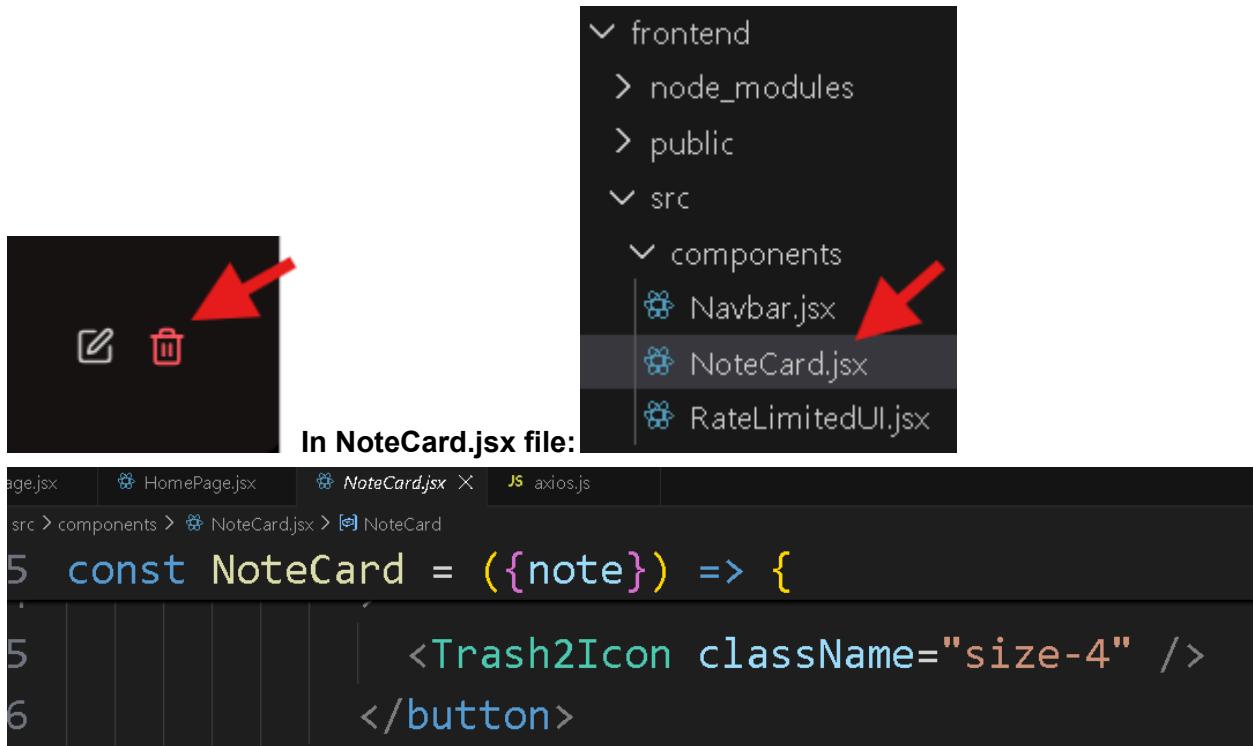


Add Delete functionality:



Code in NoteCard.jsx:

```
import { PenSquareIcon, Trash2Icon } from "lucide-react";
import { Link } from "react-router";
import { formatDate } from "../lib/utils";

const NoteCard = ({note}) => {

  const handleDelete = async(e, id) =>{
    // get rid of navigation behavior
    // When you click on trash icon, it will not navigate
    e.preventDefault()
  }

  return (
    <Link
      to={`/note/${note._id}`}
      className="card bg-base-100 hover:shadow-lg transition-all duration-200 border-t-4 border-solid border-[#00FF9D]"
    >
```

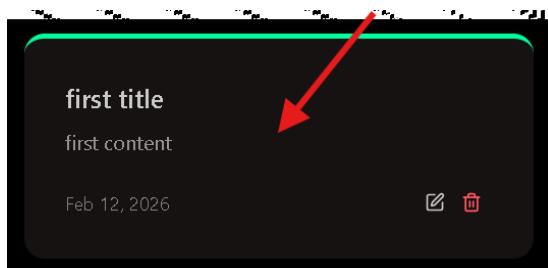
```

<div className="card-body">
  <h3 className="card-title text-base-content">{note.title}</h3>
  <p className="text-base-content/70 line-clamp-3">{note.content}</p>
  <div className="card-actions justify-between items-center mt-4">
    <span className="text-sm text-base-content/60">
      {/* {note.createdAt} */}
      {formatDate(new Date(note.createdAt))}
    </span>
    <div className="flex items-center gap-1">
      <PenSquareIcon className="size-4" />
      <button
        // trying to know which note is deleting by _id
        className="btn btn-ghost btn-xs text-error" onClick={(e) => handleDelete(e,
        note._id)}>
        <Trash2Icon className="size-4" />
      </button>
    </div>
  </div>
</div>
</div>
</Link>
);
}

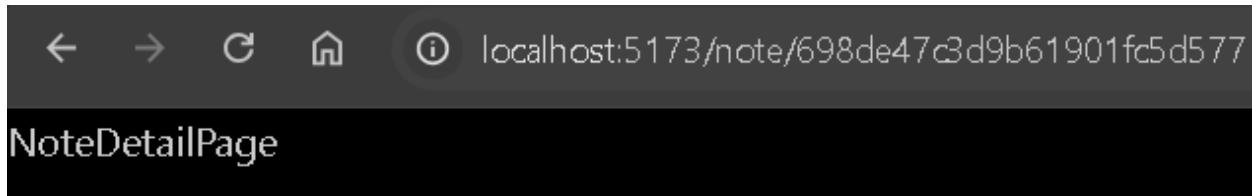
```

export default NoteCard

Then in browser:



Takes to the NoteDetail Page



When we click on Delete icon:



does not navigate to any page.

Code in NoteCard.jsx:

```
import { PenSquareIcon, Trash2Icon } from "lucide-react";
import { Link } from "react-router";
import { formatDate } from "../lib/utils";
import api from "../lib/axios"
import toast from "react-hot-toast"

const NoteCard = ({note}) => {

  const handleDelete = async(e, id) =>{
    // get rid of navigation behavior
    // When you click on trash icon, it will not navigate
    e.preventDefault()
    // If they say no to delete after confirm
    if(!window.confirm("Are You sure, You want to Delete then Note?")) return;

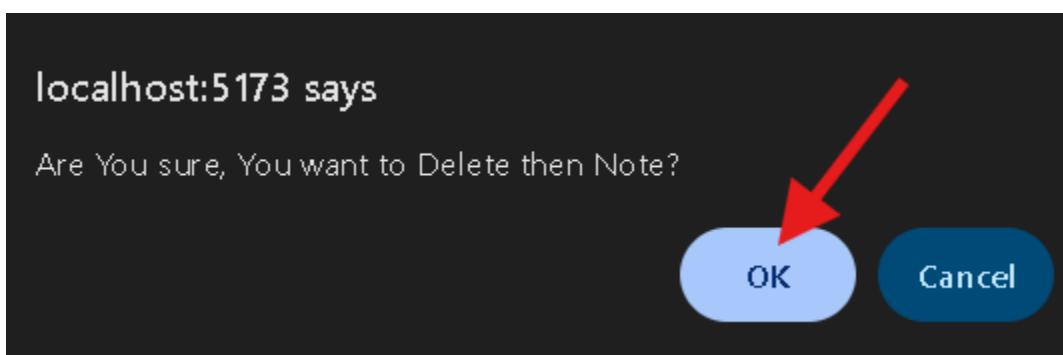
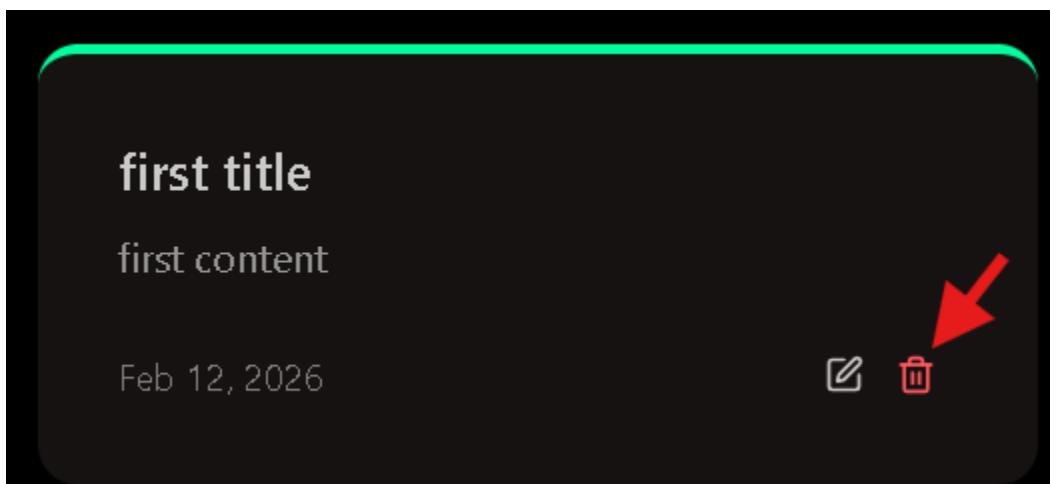
    try{
      await api.delete(`/notes/${id}`)
      toast.success("Note deleted successfully")
    }catch{
      console.log("Error in handleDelete", error)
      toast.error("Failed to Delete Note!")
    }
  }

  return (
    <Link
      to={`/note/${note._id}`}
      className="card bg-base-100 hover:shadow-lg transition-all duration-200 border-t-4 border-solid border-[#00FF9D]"
    >
      <div className="card-body">
        <h3 className="card-title text-base-content">{note.title}</h3>
        <p className="text-base-content/70 line-clamp-3">{note.content}</p>
        <div className="card-actions justify-between items-center mt-4">
          <span className="text-sm text-base-content/60">
            {/* {note.createdAt} */}
            {formatDate(new Date(note.createdAt))}
          </span>
          <div className="flex items-center gap-1">
            <PenSquareIcon className="size-4" />
            <button
              // trying to know which note is deleting by _id
```

```
        className="btn btn-ghost btn-xs text-error" onClick={(e) => handleDelete(e, note._id)}>
          <Trash2Icon className="size-4" />
        </button>
      </div>
    </div>
  </div>
</Link>
);
}
```

export default NoteCard

In browser:



Before refresh:

ThinkBoard

first title

first content

Feb 12, 2026



my s
some

Feb 11

Refresh the browser:



After refresh: its removed

ThinkBoard

my second note

some content 2

Feb 11, 2026



my first note

some content 1

Feb 11, 2026



But we want to remove the UI, without refreshing it.

In database:

Cluster0 > notes_db > notes

Documents 2 Aggregations Schema Indexes

Type a query: { field: 'value' } or [General](#)

[ADD DATA](#) [UPDATE](#) [DELETE](#) [EXPORT](#)

```
_id: ObjectId('698c0ab3f540242a5abb0379')
title : "my first note"
content : "some content 1"
createdAt : 2026-02-11T04:50:59.240+00:00
updatedAt : 2026-02-11T04:50:59.240+00:00
__v : 0
```

```
_id: ObjectId('698c0c048123dba253ad879c')
title : "my second note"
content : "some content 2"
createdAt : 2026-02-11T04:56:36.167+00:00
updatedAt : 2026-02-11T04:56:36.167+00:00
__v : 0
```

We have NoteCard component:

```
const NoteCard = ({note}) => {
  const handleDelete = async(e, id) {
    // get rid of navigation behavior
```

Which is under the HomePage:

```
const HomePage = () => {
  // fetching the notes
  const [notes, setNotes] = useState([])
```

All of our notes are stored in this state. Once it's deleted we want to update and filter it out.

In HomePage.jsx:

```
= () => {
  [notes.map(note => (
    <NoteCard key={note._id} note={note} setNotes={setNotes} />
  ))]
```

In NoteCard.jsx:

```
const NoteCard = ({note, setNotes}) => {
```

Code in HomePage.jsx:

```
import { useEffect } from 'react'
import { useState } from 'react';
import Navbar from '../components/Navbar'
import RateLimitedUI from '../components/RateLimitedUI';
import toast from "react-hot-toast";
import NoteCard from "../components>NoteCard"
import api from "../lib/axios"
```

```
const HomePage = () => {
  const [isRateLimited, setIsRateLimited] = useState(false);
  // fetching the notes
  const [notes, setNotes] = useState([])
  // to keep track of loading state, set to true, soon
  // as we load the page, we need to fetch the notes
  const [loading, setLoading] = useState(true)
  // To able to fetch that
```

```

useEffect(() => {
  const fetchNotes = async () => {
    try {
      // For get method
      // const res = await axios.get("http://localhost:5001/api/notes")
      // For post method
      // const res = await axios.post("http://localhost:5001/api/notes")
      const res = await api.get("/notes")
      // const data = await res.json()
      // Which is lot more comfortable then fetch API
      console.log(res.data)// instead of console logging the data
      // We want to update the notes state
      setNotes(res.data)
      // If we can get the data, ratelimited to false
      setIsRateLimited(false)
    } catch (erro) {
      console.log("Error fetching notes")
      console.log(erro.response)
      // We want to check for the state in:
      // const [isRateLimited, setIsRateLimited] = useState(false);
      // 429 for rateLimited
      if(erro.response?.status === 429){
        setIsRateLimited(true)
      }else{
        toast.error("Failed to load Notes")
      }
    } finally{
      setLoading(false)
    }
  }

  fetchNotes()
}, [])
return (
  // min-h-screen = minimum height will be screen
  // takes entire screen
  <div className="min-h-screen">
    {/* First it will have navbar component */}
    <Navbar />
    {isRateLimited && <RateLimitedUI />}

    <div className="max-w-7xl mx-auto p-4 mt-6">
      {loading && <div className="text-center text-primary py-10">Loading notes...</div>}
      {notes.length > 0 && !isRateLimited && (

```

```

        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
          {notes.map(note => (
            <NoteCard key={note._id} note={note} setNotes={setNotes} />
          )))
        </div>
      </div>
    );
};

export default HomePage;

```

Code in NoteCard.jsx:

```

import { PenSquareIcon, TrashIcon } from "lucide-react";
import { Link } from "react-router";
import { formatDate } from "../lib/utils";
import api from "../lib/axios"
import toast from "react-hot-toast"

const NoteCard = ({note, setNotes}) => {

  const handleDelete = async(e, id) =>{
    // get rid of navigation behavior
    // When you click on trash icon, it will not navigate
    e.preventDefault()
    // If they say no to delete after confirm
    if(!window.confirm("Are You sure, You want to Delete then Note?")) return;

    try{
      await api.delete(`/notes/${id}`)
      // removing note that deleted without refresh
      // keeping notes that not same as deleted note id
      setNotes((prev) => prev.filter((note) => note._id !== id))
      toast.success("Note deleted successfully")
    }catch{
      console.log("Error in handleDelete", error)
      toast.error("Failed to Delete Note!")
    }
  }

  return (
    <Link

```

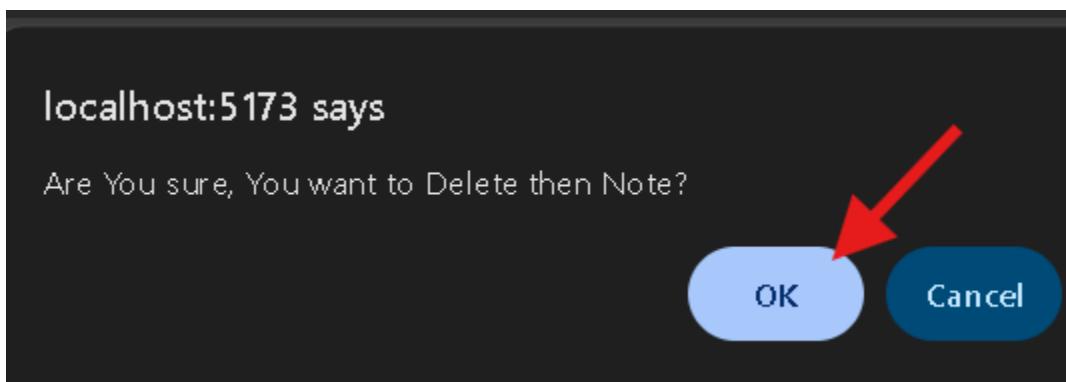
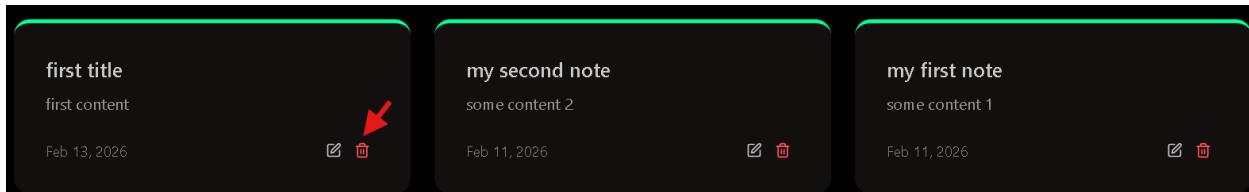
```

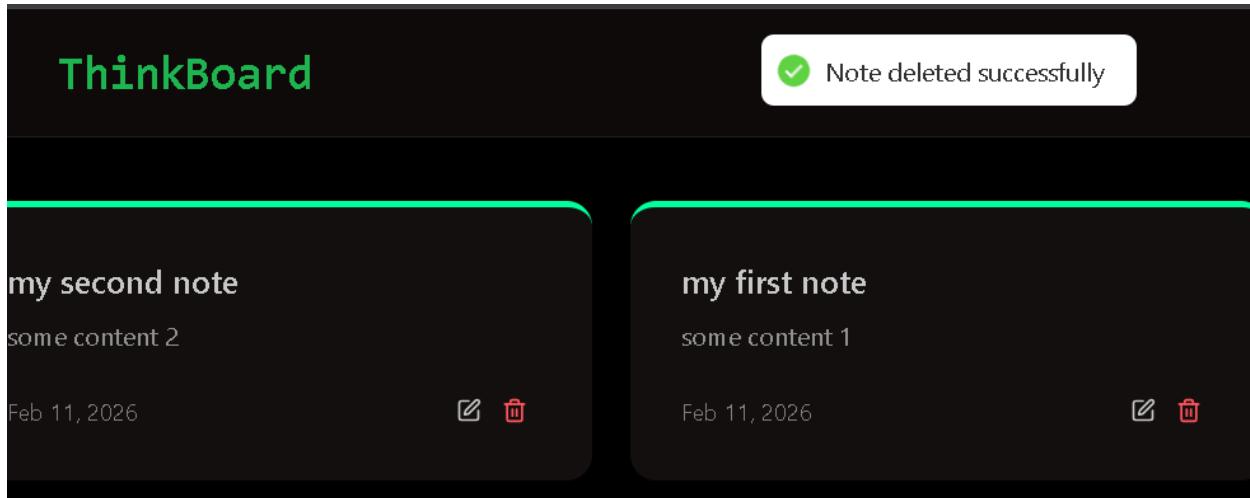
        to={`/note/${note._id}`}
        className="card bg-base-100 hover:shadow-lg transition-all duration-200
        border-t-4 border-solid border-[#00FF9D]"
      >
      <div className="card-body">
        <h3 className="card-title text-base-content">{note.title}</h3>
        <p className="text-base-content/70 line-clamp-3">{note.content}</p>
        <div className="card-actions justify-between items-center mt-4">
          <span className="text-sm text-base-content/60">
            {/* {note.createdAt} */}
            {formatDate(new Date(note.createdAt))}
          </span>
          <div className="flex items-center gap-1">
            <PenSquareIcon className="size-4" />
            <button
              // trying to know which note is deleting by _id
              className="btn btn-ghost btn-xs text-error" onClick={(e) => handleDelete(e,
              note._id)}>
              <Trash2Icon className="size-4" />
            </button>
          </div>
        </div>
      </div>
    </div>
  </Link>
);
}

```

export default NoteCard

In browser:





Removed the Note Card ui without refresh.

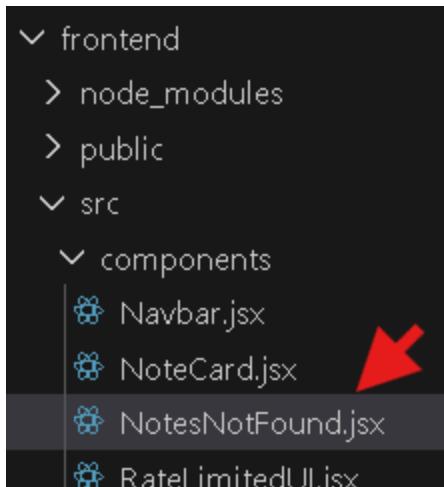
It's not good to have empty space or blank:



In HomePage.jsx:

```
HomePage.jsx X NoteCard.jsx JS axios.js
ges > HomePage.jsx > HomePage
mePage = () => {
  {notes.length === 0 && !isRateLimited && <NotesNotFound />}
```

Let's create NotesNotFound component,



type **rafce** in NotesNotFound.jsx and then **click enter**.

Then code within it:

```
import React from 'react'
```

```
const NotesNotFound = () => {
  return (
    <div>
      </div>
  )
}
```

export default NotesNotFound

Import it in HomePage.jsx:

A screenshot of a code editor showing the 'HomePage.jsx' file. The cursor is at the beginning of the 'import' statement for 'NotesNotFound'. The code in the editor is:

```
import { NoteCard } from '../components/NoteCard'
import NotesNotFound from '../components/NotesNotFound';
```

Code in NotesNotFound.jsx:

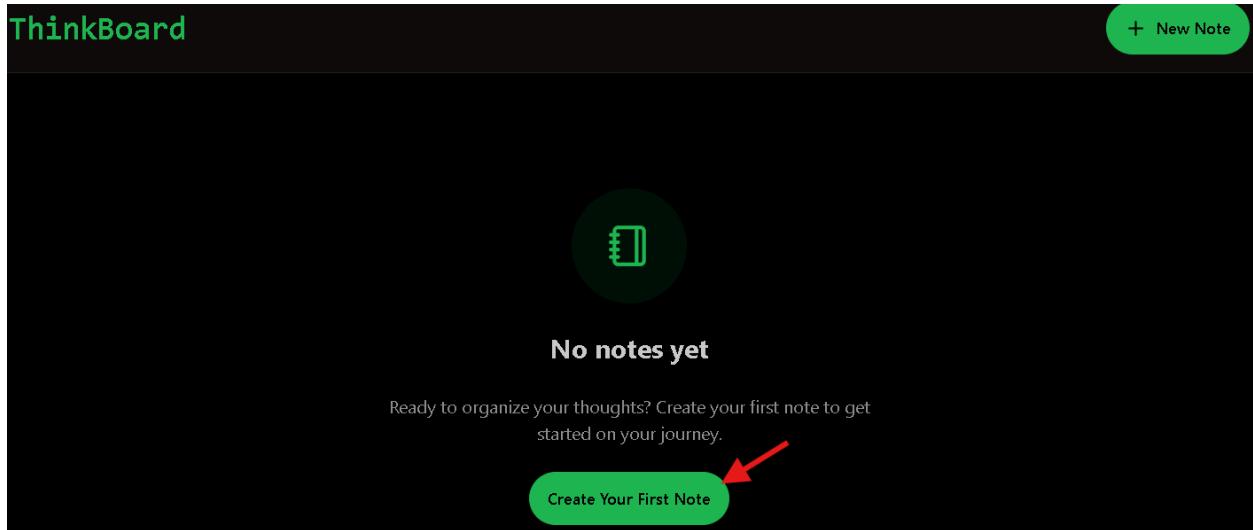
```
import { NotebookIcon } from "lucide-react";
import { Link } from "react-router";
```

```
const NotesNotFound = () => {
  return (
    <div className="flex flex-col items-center justify-center py-16 space-y-6 max-w-md mx-auto text-center">
      <div className="bg-primary/10 rounded-full p-8">
        <NotebookIcon className="size-10 text-primary" />
      </div>
      <h3 className="text-2xl font-bold">No notes yet</h3>
    </div>
  )
}
```

```
<p className="text-base-content/70">  
  Ready to organize your thoughts? Create your first note to get started on your journey.  
</p>  
<Link to="/create" className="btn btn-primary">  
  Create Your First Note  
</Link>  
</div>  
);  
};
```

export default NotesNotFound;

In Browser:



After click:

[← Back to Notes](#)

Create New Note

Title

Note Title

Content

Write your notes here...

Create Note