

When we click on the note card, it redirects to above page to update. Or we can delete them.

Code in NoteDetailPage.jsx:

```
import {useState} from "react"
import { useNavigate } from "react-router"

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)
  const [saving, setSaving] = useState(false)

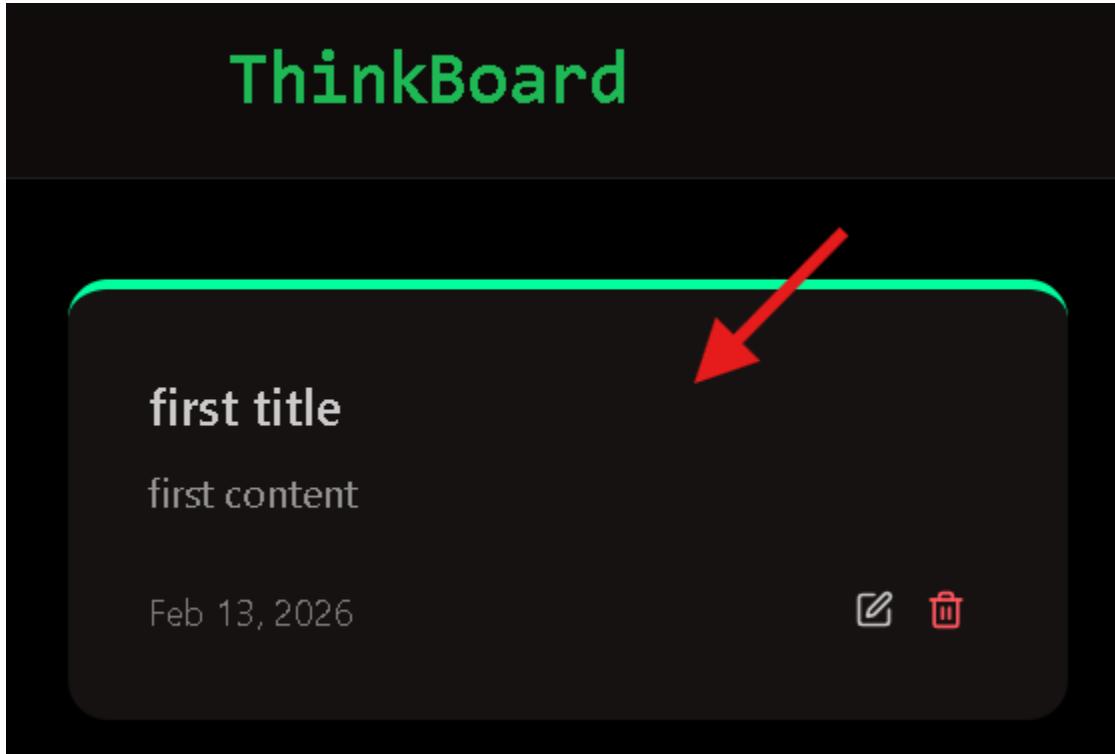
  const navigate = useNavigate()

  return (
    <div>
      NoteDetailPage
    </div>
```

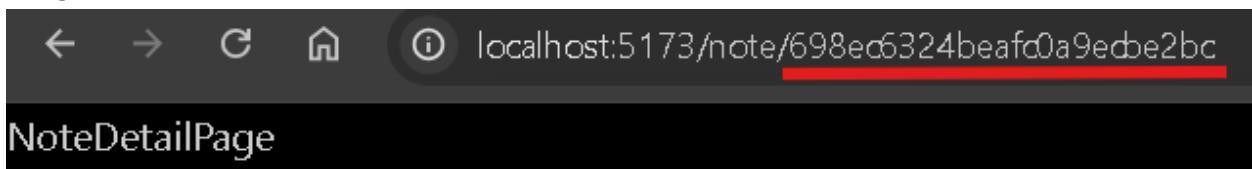
```
)  
}
```

```
export default NoteDetailPage
```

Now to get id:



We got an id:



To get the ID:

We can use the hook, **useParams**

In App.jsx:

```
const App = () => {  
  /* :id <- is dynamic */  
  <Route path="/note/:id" element={<NoteDetailPage />} />  
};
```

A screenshot of a code editor showing the 'App.jsx' file. The code defines a functional component 'App' that returns a single route component. The route is defined with a dynamic parameter ':id' in the path '/note/:id'. The component to be rendered is 'NoteDetailPage'.

Make sure **id** word must be same in both **App.jsx** and **NoteDetailPage.jsx**.

```
HomePage.jsx NotesNotFound.jsx NoteDetailPage.jsx X
> NoteDetailPage.jsx > NoteDetailPage > id
const NoteDetailPage = () =>
  const {id} = useParams()
```

Code in NoteDetailPage.jsx:

```
import {useState} from "react"
import { useNavigate, useParams } from "react-router"

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)
  const [saving, setSaving] = useState(false)

  const navigate = useNavigate()

  const {id} = useParams()
  console.log({id})

  return <div>NoteDetailPage</div>

}

export default NoteDetailPage
```

In browser:

ThinkBoard

first title

first content

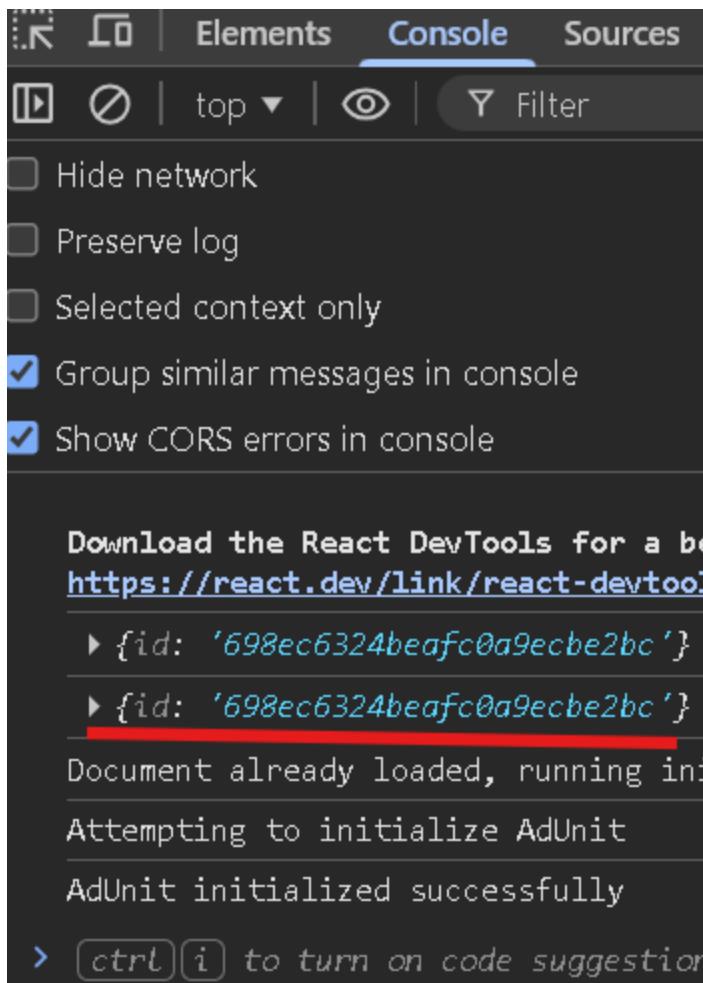
Feb 13, 2026



localhost:5173/note/698ec6324beafc0a9edbe2bc

NoteDetailPage

In console:



Code in NoteDetailPage.jsx: Fetching notes

```
import {useState, useEffect} from "react"
import { useNavigate, useParams } from "react-router"
import toast from "react-hot-toast";
import api from "../lib/axios"

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)
  const [saving, setSaving] = useState(false)

  const navigate = useNavigate()

  const {id} = useParams()
  // console.log({id})

  // Now let's try to fetch the actual note
```

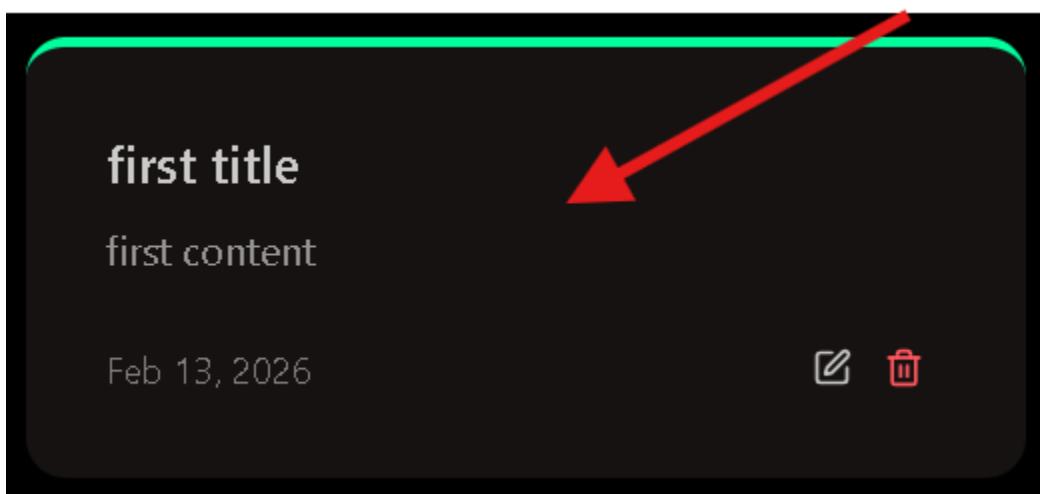
```
useEffect(() => {
  const fetchNote = async() => {
    try {
      const res = await api.get(`/notes/${id}`)
      setNote(res.data)
    } catch (error) {
      console.log("Error in fetching note:", error)
      toast.error("Failed to fetch the note")
    } finally{
      setLoading(false)
    }
  }
  fetchNote()
}, [id])

console.log({note})

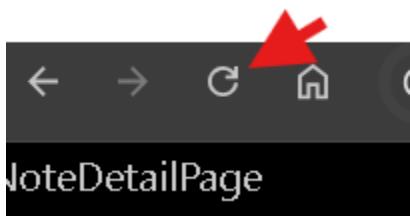
return <div>NoteDetailPage</div>
}

export default NoteDetailPage
```

In browser:



Refresh page:



In console:

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. At the top, there are several icons: a play button, a stop button, a dropdown menu set to 'top', a refresh button, and a 'Filter' input field. Below the toolbar, there is a list of configuration options with checkboxes:

- Hide network
- Log XMLHttpRequests
- Preserve log
- Eager evaluation
- Selected context only
- Autocomplete suggestions
- Group similar messages in console
- Treat code errors as warnings
- Show CORS errors in console

Below these settings, the console output area displays the following text:

```
react-devtools
Download the React DevTools for a better development experience
https://react.dev/link/react-devtools
▶ {note: null}
▼ {note: null} ⓘ
  note: null
  ▶ [[Prototype]]: Object
Document already loaded, running initialization immediately
Attempting to initialize AdUnit
AdUnit initialized successfully
▼ {note: {...}} ⓘ
  ▶ note: {_id: '698ec6324beafc0a9ecbe2bc', title: 'first title'}
  ▶ [[Prototype]]: Object
◀ ━━━━━━
▼ {note: {...}} ⓘ
  ▶ note: {_id: '698ec6324beafc0a9ecbe2bc', title: 'first title'}
  ▶ [[Prototype]]: Object
```

Code in NoteDetailPage.jsx:

```
import {useState, useEffect} from "react"
import { useNavigate, useParams } from "react-router"
import toast from "react-hot-toast";
import { LoaderIcon } from "lucide-react";
import api from "../lib/axios"
```

```

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)
  const [saving, setSaving] = useState(false)

  const navigate = useNavigate()

  const {id} = useParams()
  // console.log({id})

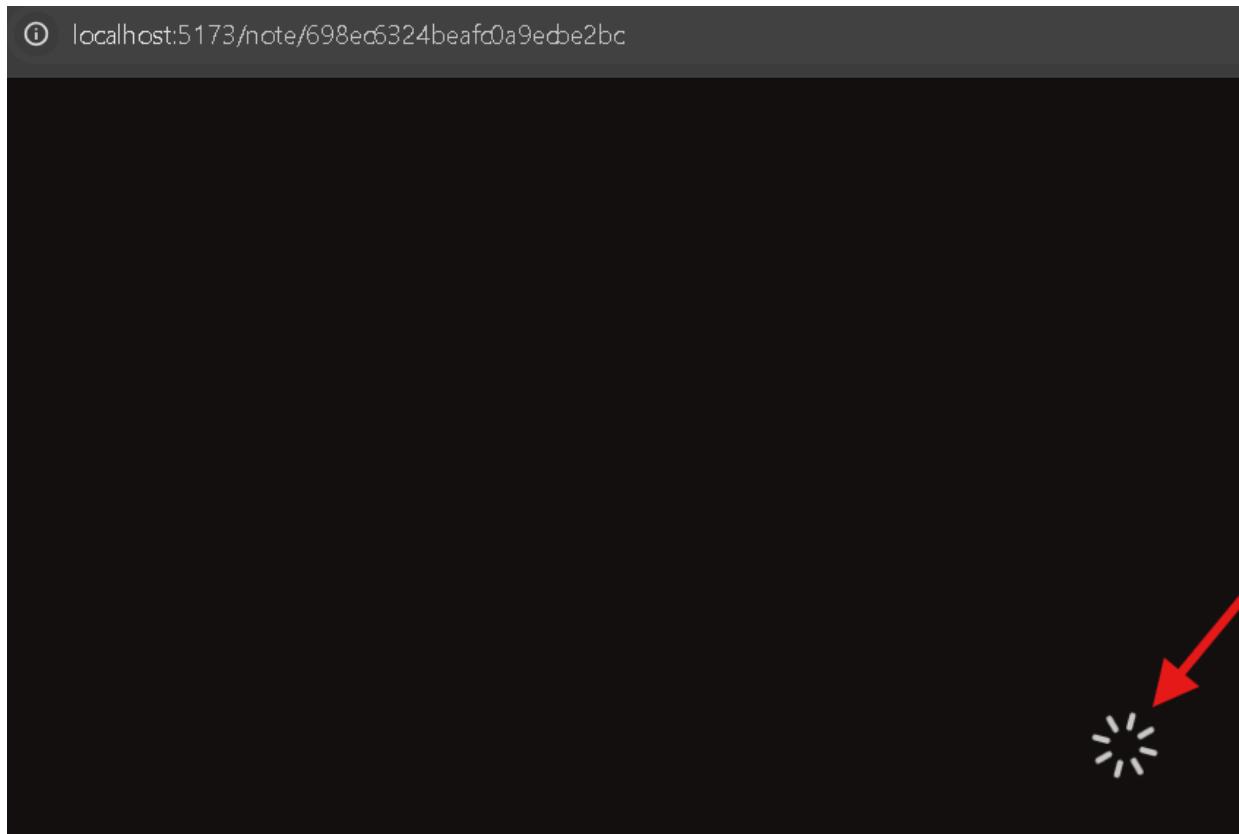
  // Now let's try to fetch the actual note
  useEffect(() => {
    const fetchNote = async() => {
      try {
        const res = await api.get(`/notes/${id}`)
        setNote(res.data)
      } catch (error) {
        console.log("Error in fetching note:", error)
        toast.error("Failed to fetch the note")
      } finally{
        setLoading(false)
      }
    }
    fetchNote()
  }, [id])

  console.log({note})
  // if(loading) {
  if(true) {
    return (
      <div className="min-h-screen bg-base-200 flex items-center justify-center">
        <LoaderIcon className="animate-spin size-10" />
      </div>
    )
  }

  return <div>NoteDetailPage</div>
}

export default NoteDetailPage
In browser: Continuously spins, because of -> if(true) {

```



Code in NoteDetailPage.jsx:

```
import {useState, useEffect} from "react"
import { useNavigate, useParams } from "react-router"
import toast from "react-hot-toast";
import { LoaderIcon, ArrowLeftIcon, Trash2Icon } from "lucide-react";
import api from "../lib/axios"
import { Link } from "react-router";

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)
  const [saving, setSaving] = useState(false)

  const navigate = useNavigate()

  const {id} = useParams()
  // console.log({id})

  // Now let's try to fetch the actual note
  useEffect(() => {
```

```

const fetchNote = async() => {
  try {
    const res = await api.get(`/notes/${id}`)
    setNote(res.data)
  } catch (error) {
    console.log("Error in fetching note:", error)
    toast.error("Failed to fetch the note")
  } finally{
    setLoading(false)
  }
}
fetchNote()
}, [id])

console.log({note})

const handleDelete = () => {

}

if(loading) {
// if(true) {
return (
<div className="min-h-screen bg-base-200 flex items-center justify-center">
  <LoaderIcon className="animate-spin size-10" />
</div>
)
}

return (
<div className="min-h-screen bg-base-200">
  <div className="container mx-auto px-4 py-8">
    <div className="flex items-center justify-between mb-6">
      <Link to="/" className="btn btn-ghost">
        <ArrowLeftIcon className="h-5 w-5" />
        Back to Notes
      </Link>
      <button onClick={handleDelete} className="btn btn-error btn-outline">
        <Trash2Icon className="h-5 w-5" />
        Delete Note
      </button>
    </div>
  </div>
</div>

```

```
)  
}  
  
export default NoteDetailPage
```

In browser:



Takes too much space.

Code in NoteDetailPage.jsx:

```
import {useState, useEffect} from "react"  
import { useNavigate, useParams } from "react-router"  
import toast from "react-hot-toast";  
import { LoaderIcon, ArrowLeftIcon, Trash2Icon } from "lucide-react";  
import api from "../lib/axios"  
import { Link } from "react-router";  
  
const NoteDetailPage = () => {  
  const [note, setNote] = useState(null)  
  const [loading, setLoading] = useState(true)  
  const [saving, setSaving] = useState(false)  
  
  const navigate = useNavigate()  
  
  const {id} = useParams()  
  // console.log({id})  
  
  // Now let's try to fetch the actual note  
  useEffect(() => {  
    const fetchNote = async() => {  
      try {  
        const res = await api.get(`/notes/${id}`)  
        setNote(res.data)  
      } catch (error) {  
        console.log("Error in fetching note:", error)  
        toast.error("Failed to fetch the note")  
      } finally{  
        setLoading(false)  
      }  
    }  
    fetchNote()  
  })
```

```
}, [id])

console.log({note})

const handleDelete = () => {

}

if/loading) {
// if(true) {
return (
<div className="min-h-screen bg-base-200 flex items-center justify-center">
  <LoaderIcon className="animate-spin size-10" />
</div>
)
}

return (
<div className="min-h-screen bg-base-200">
  <div className="container mx-auto px-4 py-8">
    <div className="max-w-2xl mx-auto">

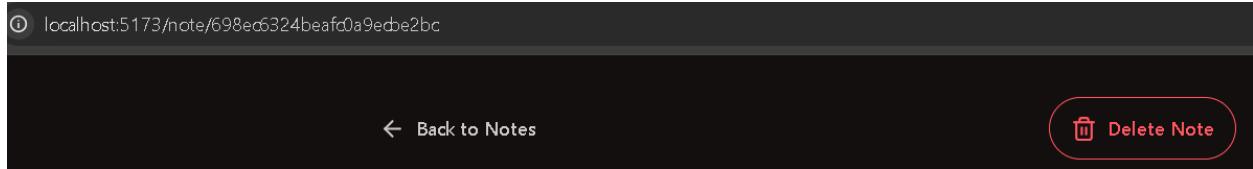
      <div className="flex items-center justify-between mb-6">
        <Link to="/" className="btn btn-ghost">
          <ArrowLeftIcon className="h-5 w-5" />
          Back to Notes
        </Link>
        <button onClick={handleDelete} className="btn btn-error btn-outline">
          <Trash2Icon className="h-5 w-5" />
          Delete Note
        </button>
      </div>

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

}

export default NoteDetailPage
```

In browser:



Code in NoteDetailPage.jsx:

```
import {useState, useEffect} from "react"
import { useNavigate, useParams } from "react-router"
import toast from "react-hot-toast";
import { LoaderIcon, ArrowLeftIcon, Trash2Icon } from "lucide-react";
import api from "../lib/axios"
import { Link } from "react-router";

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)
  const [saving, setSaving] = useState(false)

  const navigate = useNavigate()

  const {id} = useParams()
  // console.log({id})

  // Now let's try to fetch the actual note
  useEffect(() => {
    const fetchNote = async() => {
      try {
        const res = await api.get(`/notes/${id}`)
        setNote(res.data)
      } catch (error) {
        console.log("Error in fetching note:", error)
        toast.error("Failed to fetch the note")
      } finally{
        setLoading(false)
      }
    }
    fetchNote()
  }, [id])

  console.log({note})
}
```

```
const handleDelete = () => {  
}  
const handleSave = () => {  
}  
  
}  
  
if/loading) {  
// if(true) {  
return (  
  <div className="min-h-screen bg-base-200 flex items-center justify-center">  
    <LoaderIcon className="animate-spin size-10" />  
  </div>  
)  
}  
  
return (  
  <div className="min-h-screen bg-base-200">  
    <div className="container mx-auto px-4 py-8">  
      <div className="max-w-2xl mx-auto">  
        <div className="flex items-center justify-between mb-6">  
          <Link to="/" className="btn btn-ghost">  
            <ArrowLeftIcon className="h-5 w-5" />  
            Back to Notes  
          </Link>  
          <button onClick={handleDelete} className="btn btn-error btn-outline">  
            <Trash2Icon className="h-5 w-5" />  
            Delete Note  
          </button>  
        </div>  
  
        <div className="card bg-base-100">  
          <div className="card-body">  
            <div className="form-control mb-4">  
              <label className="label">  
                <span className="label-text">Title</span>  
              </label>  
              <input  
                type="text"  
                placeholder="Note title"  
                className="input input-bordered"  
                value={note.title}  
                onChange={(e) => setNote({ ...note, title: e.target.value })}  
              />  
            </div>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>  
)
```

```
</div>

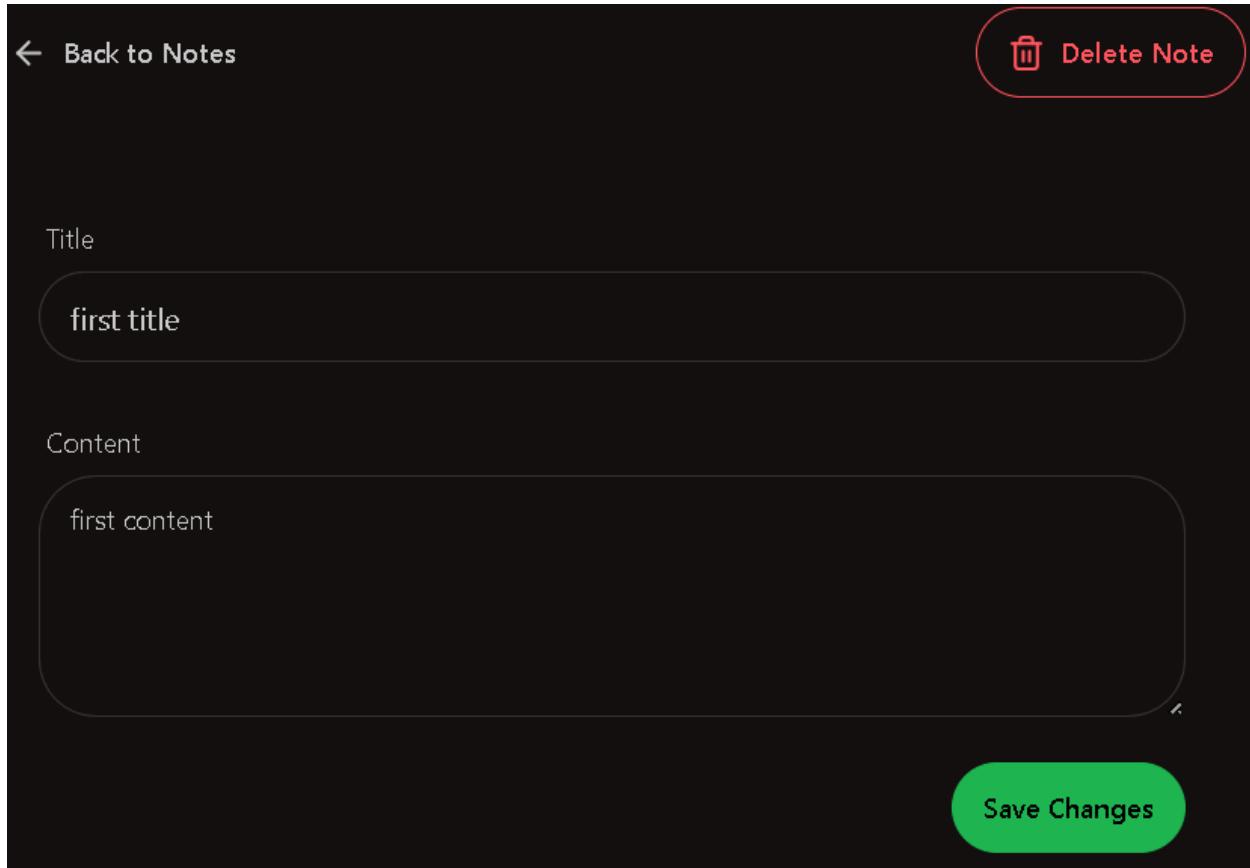
<div className="form-control mb-4">
  <label className="label">
    <span className="label-text">Content</span>
  </label>
  <textarea
    placeholder="Write your note here..." 
    className="textarea textarea-bordered h-32"
    value={note.content}
    onChange={(e) => setNote({ ...note, content: e.target.value })}>
  />
</div>

<div className="card-actions justify-end">
  <button className="btn btn-primary" disabled={saving} onClick={handleSave}>
    {saving ? "Saving..." : "Save Changes"}
  </button>
</div>
</div>
</div>
</div>
</div>
</div>
);

}

export default NoteDetailPage
```

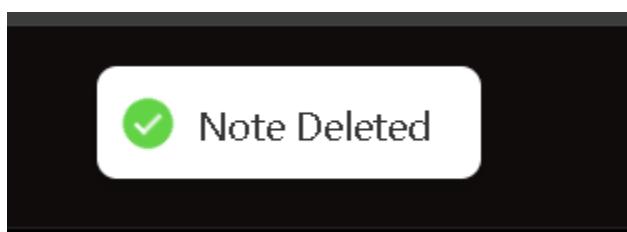
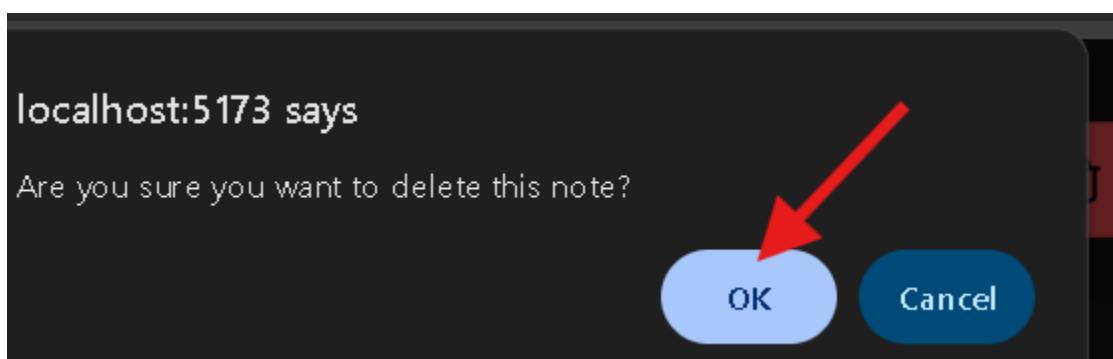
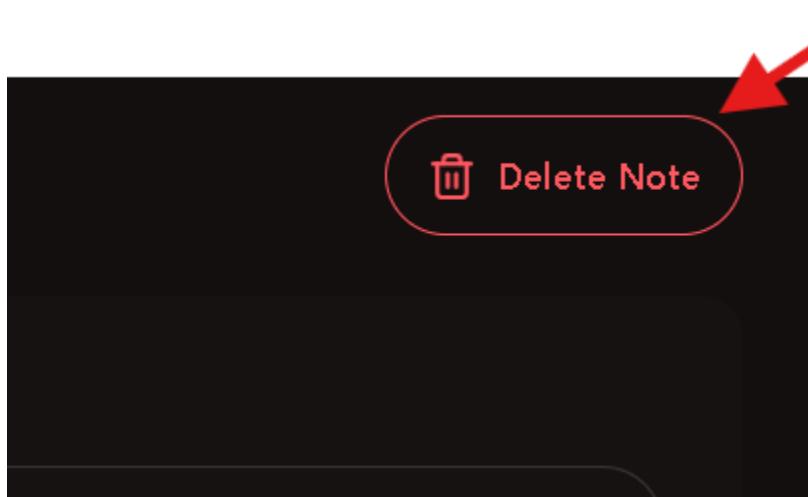
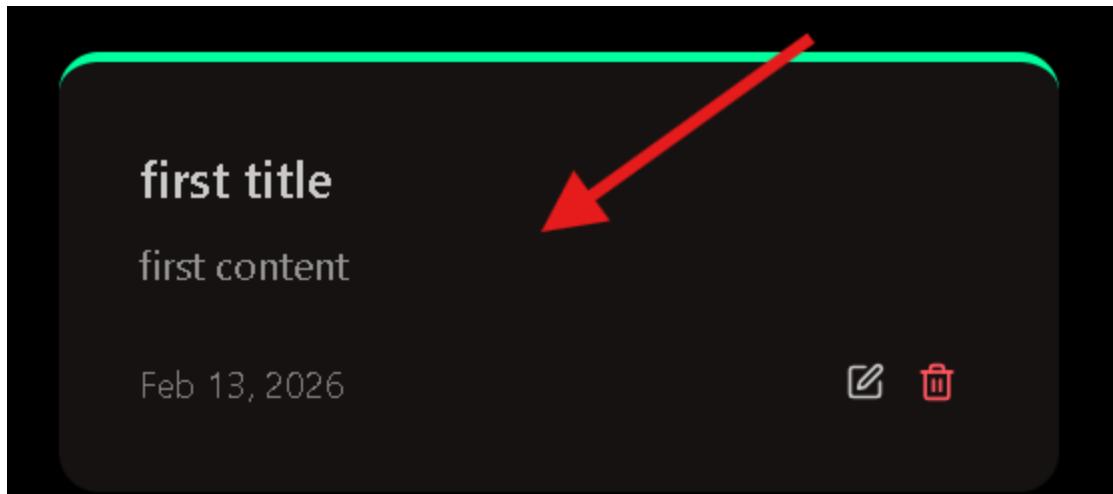
In browser:



Code in NoteDetailPage.jsx:

```
const handleDelete = async () => {
  if(!window.confirm("Are you sure you want to delete this note?")) return
  // if user wants to delete
  try {
    await api.delete(`/notes/${id}`)
    toast.success("Note Deleted")
    // Navigate to the HomePage
    navigate("/")
  } catch (error) {
    console.log("Error deleting the note:", error)
    toast.error("Failed to delete note")
  }
}
```

In browser:

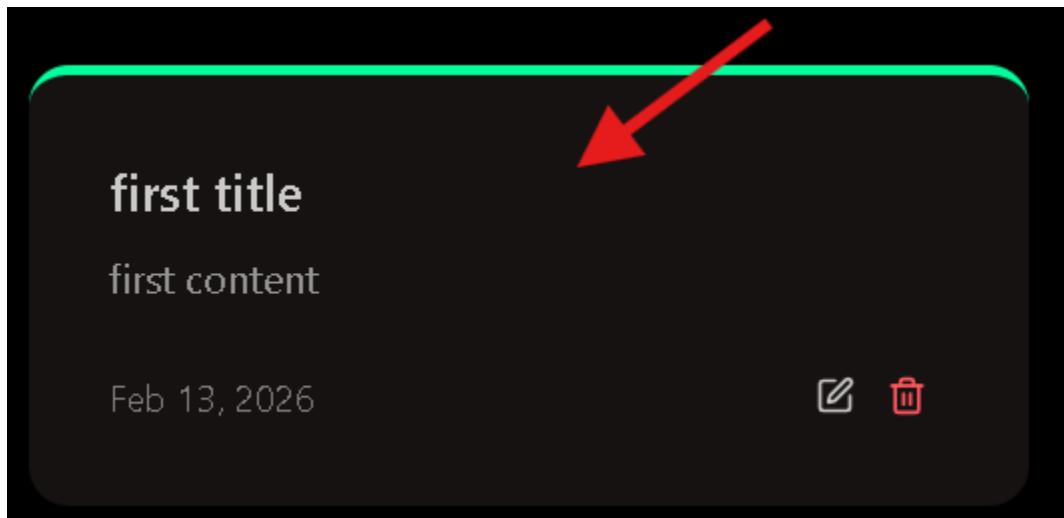


Code in NoteDetailPage.jsx:

```
const handleSave = async () => {
  // if title or content not provided
  if(!note.title.trim() || !note.content.trim()){
    toast.error("Please add a title or content")
    return
  }
  setSaving(true)

  try {
    await api.put(`/notes/${id}`, note)
    toast.success("Note Updated Successfully")
  } catch (error) {
    console.log("Error saving the note:", error)
    toast.error("Failed to Update Note!")
  } finally{
    setSaving(false)
  }
}
```

In Browser:



[← Back to Notes](#)

 Delete Note

Title

updated first title

Content

updated first content

 Save Changes

[← Back to Notes](#)

 Note Updated Successfully

 Delete Note

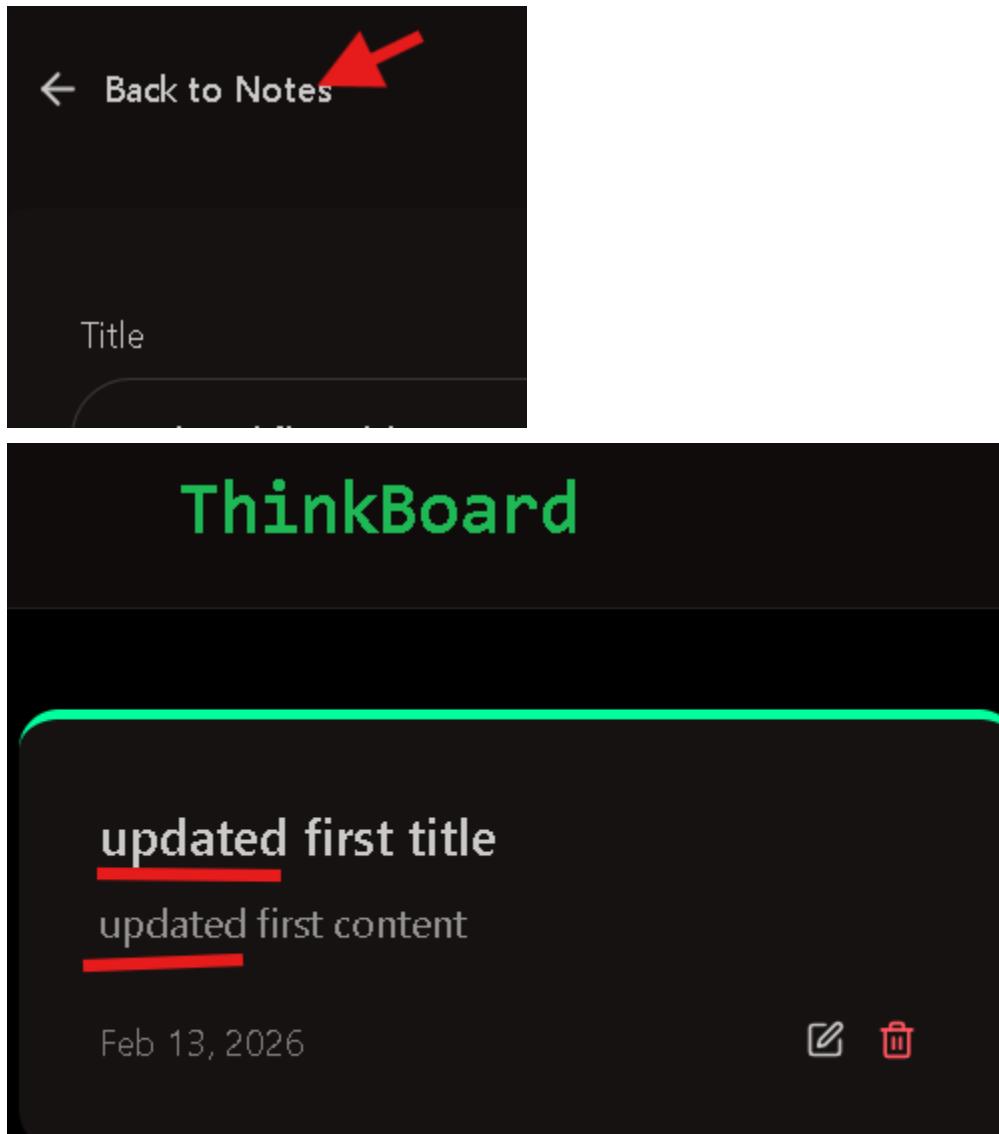
Title

updated first title

Content

updated first content

Save Changes



After updating, it must automatically navigate to the home page.

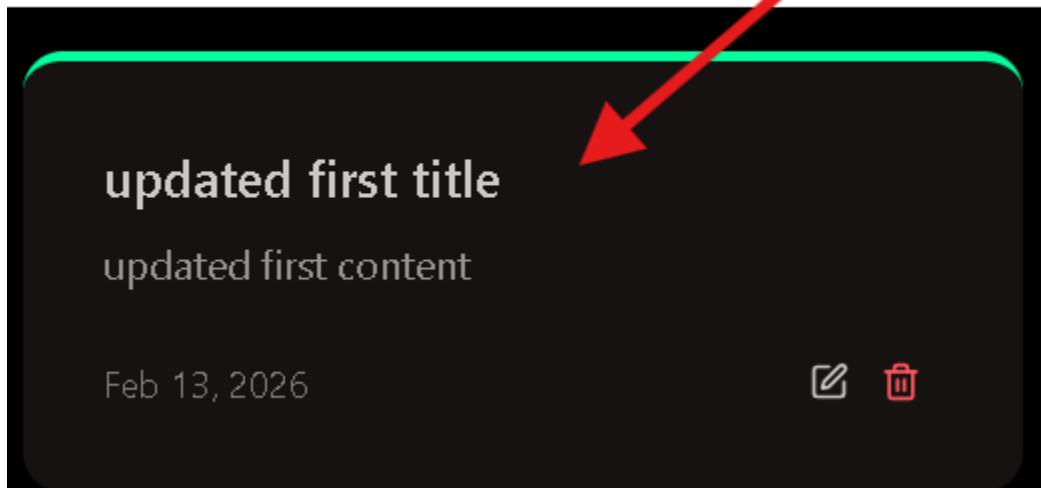
Code in NoteDetailPage.jsx:

```
const handleSave = async () => {
  // if title or content not provided
  if(!note.title.trim() || !note.content.trim()){
    toast.error("Please add a title or content")
    return
  }
  setSaving(true)

  try {
    await api.put(`/notes/${id}` , note)
```

```
toast.success("Note Updated Successfully")
navigate("/")
} catch (error) {
  console.log("Error saving the note:", error)
  toast.error("Failed to Update Note!")
} finally{
  setSaving(false)
}
}
```

In browser:



[← Back to Notes](#)

 Delete Note

Title

first title

Content

first content

 Save Changes

ThinkBoard

 Note Updated Successfully

 Failed to Update Note!

Loading notes...

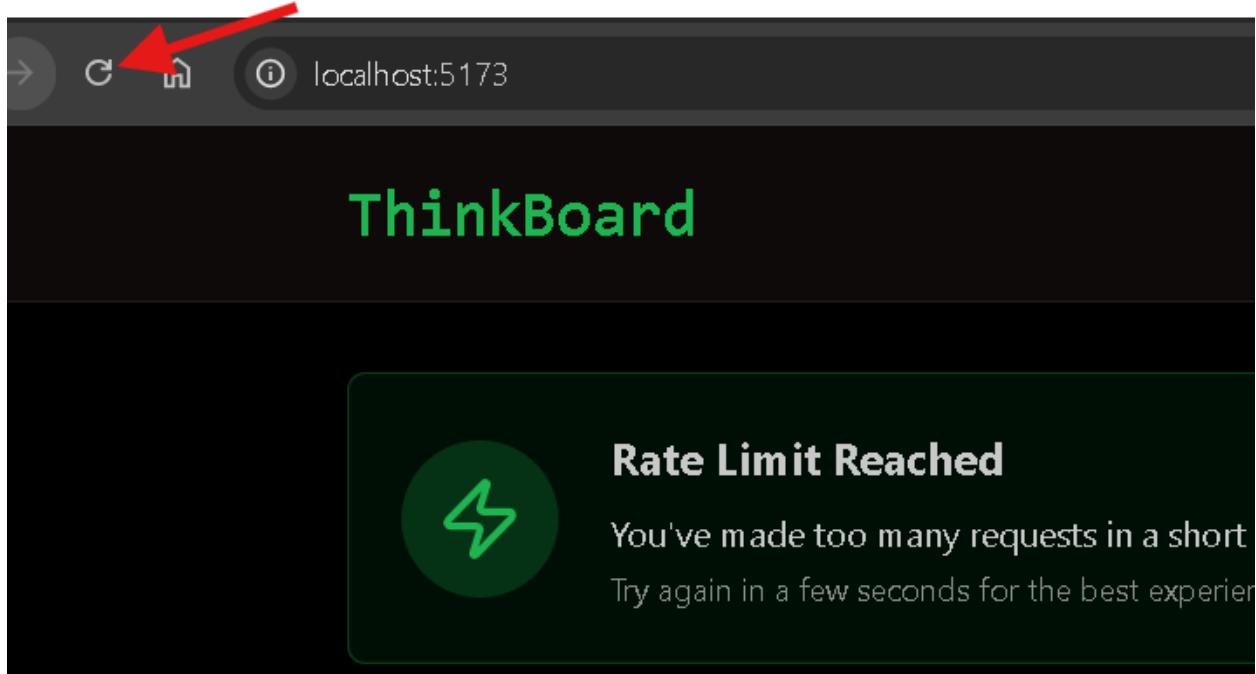


No notes yet

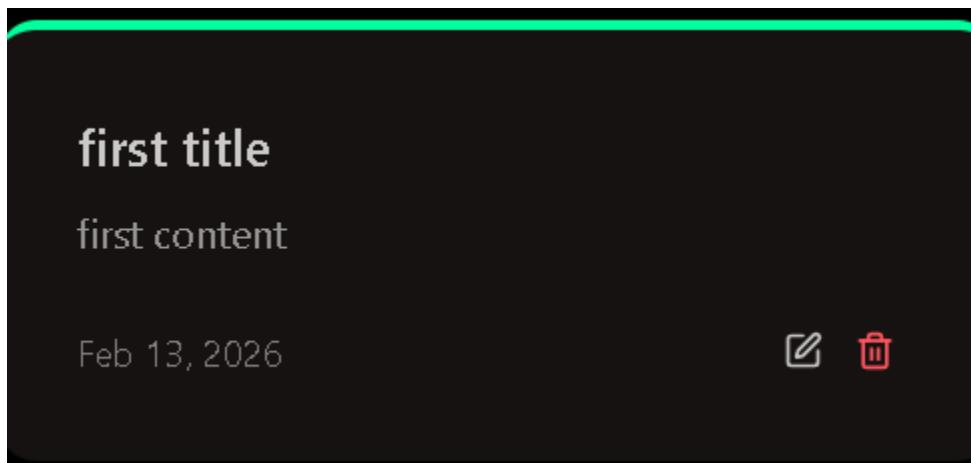
Ready to organize your thoughts? Create your first note
started on your journey.

[Create Your First Note](#)

After some time: refresh



Updated Successfully!



Code in NoteDetailPage.jsx:

```
import {useState, useEffect} from "react"
import { useNavigate, useParams } from "react-router"
import toast from "react-hot-toast";
import { LoaderIcon, ArrowLeftIcon, Trash2Icon } from "lucide-react";
import api from "../lib/axios"
import { Link } from "react-router";

const NoteDetailPage = () => {
  const [note, setNote] = useState(null)
  const [loading, setLoading] = useState(true)

  useEffect(() => {
    const fetchNote = async () => {
      setLoading(true)
      const response = await api.get(`notes/${params.id}`)
      setNote(response.data)
      setLoading(false)
    }
    fetchNote()
  }, [params.id])
```

```
const [saving, setSaving] = useState(false)

const navigate = useNavigate()

const {id} = useParams()
// console.log({id})

// Now let's try to fetch the actual note
useEffect(() => {
  const fetchNote = async() => {
    try {
      const res = await api.get(`/notes/${id}`)
      setNote(res.data)
    } catch (error) {
      console.log("Error in fetching note:", error)
      toast.error("Failed to fetch the note")
    } finally{
      setLoading(false)
    }
  }
  fetchNote()
}, [id])

console.log({note})

const handleDelete = async () => {
  if(!window.confirm("Are you sure you want to delete this note?")) return
  // if user wants to delete
  try {
    await api.delete(`/notes/${id}`)
    toast.success("Note Deleted")
    // Navigate to the homePage
    navigate("/")
  } catch (error) {
    console.log("Error deleting the note:", error)
    toast.error("Failed to delete note")
  }
}

const handleSave = async () => {
  // if title or content not provided
  if(!note.title.trim() || !note.content.trim()){
    toast.error("Please add a title or content")
    return
  }
}
```

```

setSaving(true)

try {
  await api.put(`/notes/${id}`, note)
  toast.success("Note Updated Successfully")
  navigate("/")
} catch (error) {
  console.log("Error saving the note:", error)
  toast.error("Failed to Update Note!")
} finally{
  setSaving(false)
}
}

if/loading) {
// if(true) {
return (
  <div className="min-h-screen bg-base-200 flex items-center justify-center">
    <LoaderIcon className="animate-spin size-10" />
  </div>
)
}

return (
<div className="min-h-screen bg-base-200">
  <div className="container mx-auto px-4 py-8">
    <div className="max-w-2xl mx-auto">
      <div className="flex items-center justify-between mb-6">
        <Link to="/" className="btn btn-ghost">
          <ArrowLeftIcon className="h-5 w-5" />
          Back to Notes
        </Link>
        <button onClick={handleDelete} className="btn btn-error btn-outline">
          <Trash2Icon className="h-5 w-5" />
          Delete Note
        </button>
      </div>
      <div className="card bg-base-100">
        <div className="card-body">
          <div className="form-control mb-4">
            <label className="label">
              <span className="label-text">Title</span>
            </label>

```

```

<input
  type="text"
  placeholder="Note title"
  className="input input-bordered"
  value={note.title}
  onChange={(e) => setNote({ ...note, title: e.target.value })}
/>
</div>

<div className="form-control mb-4">
  <label className="label">
    <span className="label-text">Content</span>
  </label>
  <textarea
    placeholder="Write your note here..."
    className="textarea textarea-bordered h-32"
    value={note.content}
    onChange={(e) => setNote({ ...note, content: e.target.value })}
  />
</div>

<div className="card-actions justify-end">
  <button className="btn btn-primary" disabled={saving} onClick={handleSave}>
    {saving ? "Saving..." : "Save Changes"}
  </button>
</div>
</div>
</div>
</div>
</div>
);

}

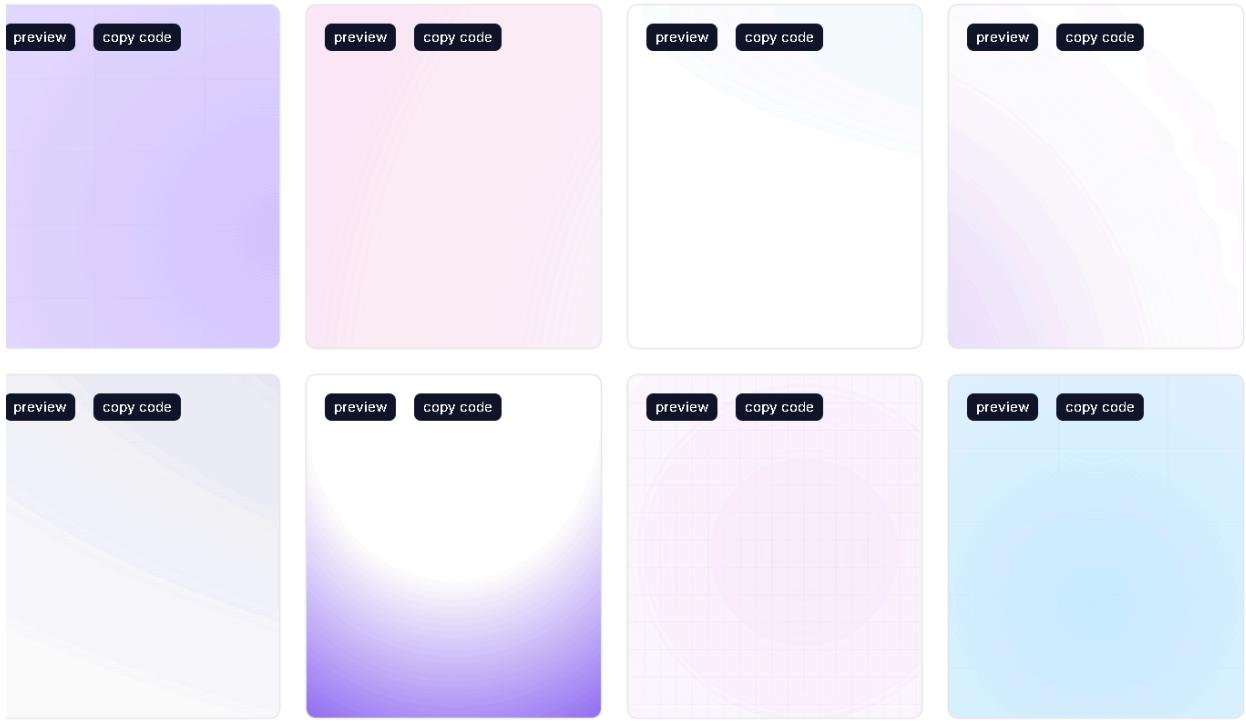
export default NoteDetailPage

```

To get the background:

Visit: <https://bg.ibelick.com/>

Which has more background:



Available in `App.jsx` file:

```
<div className="absolute inset-0
-z-10 h-full w-full items-center
px-5 py-24
[background:radial-gradient(125%_125%
_at_50%_10%,#000_60%,#00FF9D40_100%
)]" />
```