

Github URL

Github URL for Vercel

Vercel URL

Video: W07-P1: Use tailwind to style react icons and add a delete button

The screenshot displays a development environment with a code editor on the left and a browser window on the right.

Code Editor (VS Code):

```
const Blog2_02 = ({ id, img, category, title, descrip }) => {
  return (
    <article key={id} className='blog'>
      <img src={img} alt='Coffee photo' className='img blog-img' />
      <div className='blog-content'>
        <span className='flex'>
          {category}
          {category === 'lifestyle' ? (
            <FaMugSaucer className='ml-1' />
          ) : (
            <FaGlobe className='ml-1' />
          )}
        </span>
        <h3>{title}</h3>
        <p>{descrip}</p>
        <div className='flex justify-between items-center'>
          <Link to='#'>read more</Link>
          <button type='button' className='text-red-700 bg-red-100 capitalize px-2 py-1 hover:bg-red-300 rounded'>
            delete
          </button>
        </div>
      </div>
    </article>
  )
}
```

Browser:

The browser shows a website titled "TkuDemo" with a navigation bar including "HOME", "MEGA MENU +", "DEMO +", "TUTORIALS +", and "CONTACT". The main content area is titled "Blogs From Local Json --". It lists three blog posts:

- W7-BlogLocalJson_02**: LIFESTYLE Seven Reasons Why Coffee Is Awesome. Includes a "Read More" button and a "Delete" button.
- W4-BlogNode_02**: TRAVEL Travel To Paris. Includes a "Read More" button and a "Delete" button.
- W6-BlogSupa_02**: LIFESTYLE Coffee Brings Friendship. Includes a "Read More" button and a "Delete" button.

ab2e534 apple550678

2025-10-31 16:06:47 +0800

Video: W07-P1: Use tailwind to style re

Video: W07-P2: Implement delete a blog, clear all blogs, load all blogs

=> delete first blog

The screenshot shows a browser window with multiple tabs open. The active tab is 'localhost:5173/localjson2_02'. The page displays a list of three blog posts:

- Travel To Paris**: Category: TRAVEL, Description: "Lorem ipsum dolor sit amet consectetur adipisicing elit.", Read More, Delete button.
- Coffee Brings Friendship**: Category: LIFESTYLE, Description: "Lorem ipsum dolor sit amet consectetur adipisicing elit.", Read More, Delete button.
- Coffee Make You Feel Good**: Category: LIFESTYLE, Description: "Lorem ipsum dolor sit amet consectetur adipisicing elit.", Read More, Delete button.

A red box highlights the first blog post. The right side of the screen shows the React DevTools component tree and state hooks. The component tree shows the structure of the application, and the state hook for 'BlogLocalJsonPage2_02' is expanded, showing the data for each blog post. A red box highlights the data for the first blog post, which matches the one highlighted on the left.

```
hooks
  1 State: "李國蘋"
  2 State: 213410102
  3 State: [ { }, { }, { }, { }, { }, { }, { }, { } ]
    0: { category: "travel", descrip: "Lorem ipsum dolor si..." }
      id: 2
      title: "Travel To Paris"
      descrip: "Lorem ipsum dolor sit amet consectetur adipisicing e..."
      category: "travel"
      img: "/images/photo-2.jpg"
      remote_url: "https://erogcveccbzsyhbgrtf.supabase.co/storage/...
      new_entry: ""
```

=> clear all blogs

The screenshot shows a browser window with multiple tabs open. The active tab is 'localhost:5173/localjson2_02'. The page content displays 'Blogs From Local Json 2 -- 李國蘋, 213410102' with two buttons: 'Clear All Blogs' (highlighted with a red box) and 'Load All Blogs'.

On the right side of the browser, the developer tools component tree is visible. It shows the following structure:

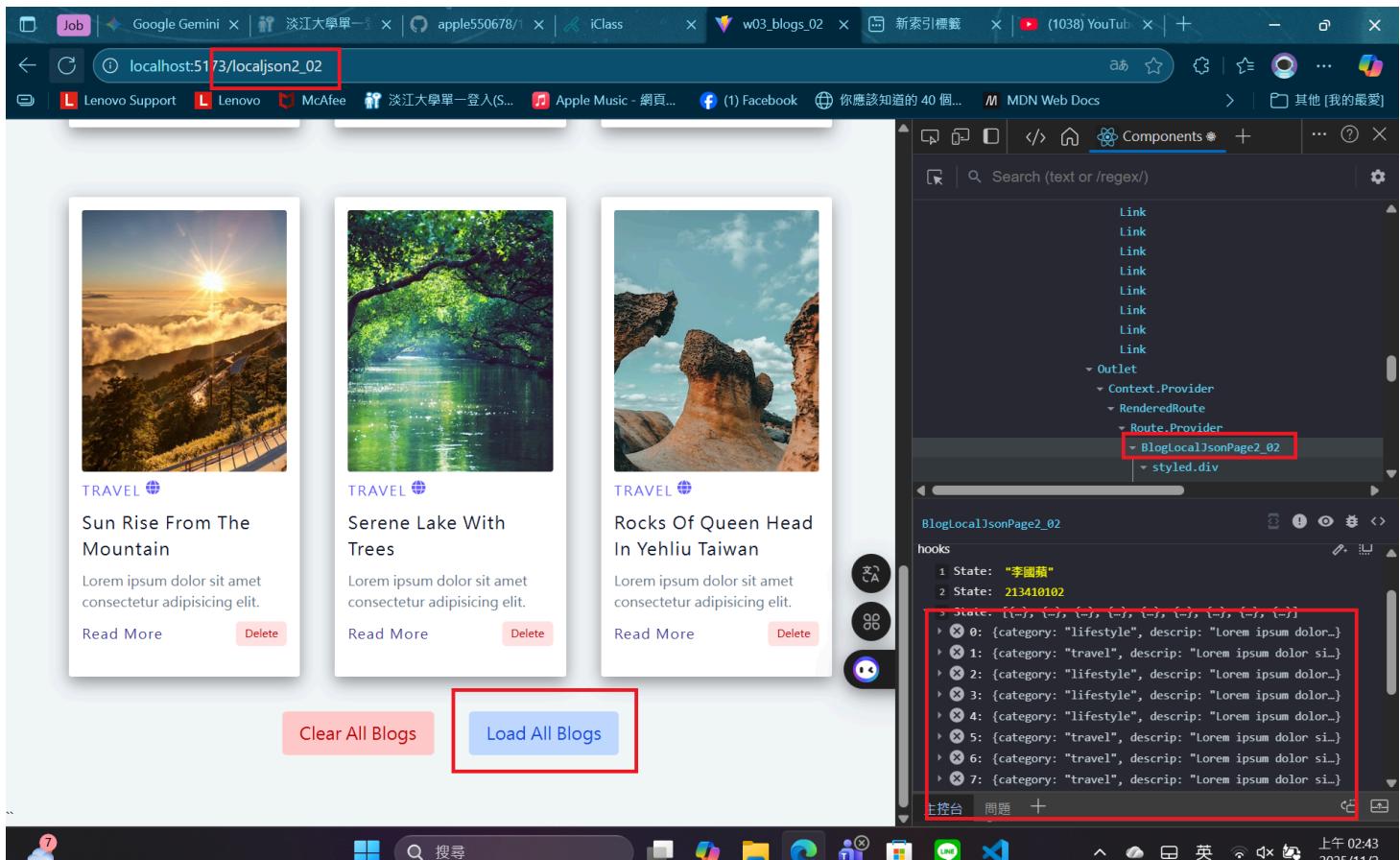
- Link
- Outlet
- Context.Provider
- RenderedRoute
- Route.Provider
- BlogLocalJsonPage2_02
- styled.div

In the 'hooks' section of the developer tools, there is a list of state variables:

- 1 State: "李國蘋"
- 2 State: 213410102
- 3 State: [] (highlighted with a red box)
- 4 State: {msg: "", show: false, type: ""}

The 'source' section shows the file 'BlogLocalJsonPage2_02.jsx?t=1762022126795:8'.

=> load all blogs



=> code for deleting a blog

```
src > pages > BlogLocalJsonPage2_02.jsx M
5  const BlogLocalJsonPage2_02 = () => {
8    const [blogs_02, setBlogs_02] = useState(blogData_02)
10   const removeItem = (id) => {
11     setBlogs_02(blogs_02.filter((blog) => blog.id !== id))
12   }
13
14   const clearAllBlogs = () => {
15   }
16
17   const loadAllBlogs = () => [
18     ...
19   ]
20
21   return (
22     ...
23     <section className='blogs'>
24       <div className='section-title'>
25         <h2>
26           blogs from local json -- {name}, {id}
27           {''}
28         </h2>
29       </div>
30       <div className='blogs-center'>
31         {blogs_02.map((item) => {
32           const { id, title, descrip, category, img } = item
33           return (
34             <Blog2_02
35               key={id}
36               id={id}
37               title={title}
38               descrip={descrip}
39               category={category}
40               img={img}
41               removeItem={removeItem}
42             >
43           )
44         })
45       </div>
46     </section>
47   )
48
49   <button type='button' onClick={() => removeItem(id)}>
50     delete
51   </button>
52
53   <button type='button' onClick={clearAllBlogs}>
54     clear all blogs
55   </button>
56
57   <button type='button' onClick={loadAllBlogs}>
58     load all blogs
59   </button>
60
61   </div>
62 </div>
63
64   ...
65 </div>
66 )
```

=> code for clear and load all blogs

```
src > pages > BlogLocalJsonPage2_02.jsx U
1  import { useState } from 'react'
2  import blogData_02 from '../assets/data/blogData'
3  import Blog2_02 from '../components/Blog2_02'
4
5  const BlogLocalJsonPage2_02 = () => {
6    const [name, setName] = useState('李國蘋')
7    const [id, setId] = useState(213410102)
8    const [blogs_02, setBlogs_02] = useState(blogData_02)
9
10   const removeItem = (id) => {
11     setBlogs_02(blogs_02.filter((blog) => blog.id !== id))
12   }
13
14   const clearAllBlogs = () => {
15     setBlogs_02([])
16   }
17
18   const loadAllBlogs = () => {
19     setBlogs_02(blogData_02)
20   }
21
22   return (
23     ...
24     <section className='blogs'>
25       <div className='section-title'>
26         <h2>
27           blogs from local json -- {name}, {id}
28           {''}
29         </h2>
30       </div>
31       <div className='blogs-center'>
32         {blogs_02.map((item) => {
33           ...
34         })
35       </div>
36     </section>
37   )
38
39   <button type='button' onClick={clearAllBlogs}>
40     clear all blogs
41   </button>
42
43   <button type='button' onClick={loadAllBlogs}>
44     load all blogs
45   </button>
46
47   </div>
48 </div>
49
50   ...
51 </div>
52 )
```

Video: W07-P3: Implement Alert_xx

=> code for Alert_xx

The screenshot shows two open files in VS Code:

- BlogLocalJsonPage2_02.jsx**: This file contains a component that uses the `useState` hook to manage an alert state. It defines a function `showAlert` that updates the state. The component then conditionally renders the `Alert_02` component based on the `alert.show` value.
- Alert_02.jsx**: This file defines a functional component `Alert_02`. It takes `alert` and `showAlert` as props. It uses `useEffect` to set a timeout that calls `showAlert` after 3000ms. It then returns a styled `p` element with the alert message.

```
const [blogData_02] = useState([])

const [alert, setAlert] = useState({
  show: false,
  msg: '',
  type: ''
})

const showAlert = (show = false, msg = '') => {
  setAlert({ show, msg, type })
}

const removeItem = (id) => {
  showAlert(true, 'blog removed', 'danger')
  setBlogs_02(blogs_02.filter((blog) => blog.id !== id))
}

const clearAllBlogs = () => {
  showAlert(true, 'clear all blogs', 'danger')
  setBlogs_02([[]])
}

const loadAllBlogs = () => {
  showAlert(true, 'load all blogs', 'success')
  setBlogs_02([])
  setBlogs_02(blogData_02)
}

return (
  <div>
    {alert.show && <Alert_02 alert={alert}>
      <showAlert={showAlert}>/</showAlert>
    }
  </div>
)
```

```
import { useEffect } from 'react'

const Alert_02 = ({ alert, showAlert }) => {
  useEffect(() => {
    const timeout = setTimeout(() => {
      showAlert()
    }, 3000)
    return () => clearTimeout(timeout)
  }, [alert])
}

if (alert.type === 'success') {
  return (
    <p className='grid items-center text-center capitalize bg-green-200'>
      {alert.msg}
    </p>
  )
} else if (alert.type === 'danger') {
  return (
    <p className='grid items-center text-center capitalize bg-red-200'>
      {alert.msg}
    </p>
  )
}

export default Alert_02
```