

Haidar Ramadhan

Membuat UI untuk website pada page.tsx, yang akan mengakses local api.

```
website > src > app > page.tsx > HomePage > data.map() callback
1  "use client"; // Tell Next.js this is a client-side component
2
3  import { useEffect, useState } from "react"; // Import React hooks
4
5  // Type definition for estimate data
6  type Estimate = {
7    id: number; // Unique ID
8    houseSize: number; // Size of the house
9    price: number; // Predicted price
10 };
11
12 export default function HomePage() {
13   const [houseSize, setHouseSize] = useState(""); // Input state
14   const [data, setData] = useState<Estimate[]>([]); // List of estimates
15   const [loading, setLoading] = useState(false); // Loading state
16
17   // GET: fetch all estimates
18   const fetchData = async () => {
19     const res = await fetch("/api/estimate"); // Call API (proxy)
20     const json = await res.json(); // Parse response
21     setData(json.rumah); // Save data to state
22   };
23
24   // POST: send house size for prediction
25   const handlePredict = async () => {
26     if (!houseSize) return; // Prevent empty input
27
28     setLoading(true); // Start loading
29
30     await fetch("/api/estimate", {
31       method: "POST", // HTTP POST
32       headers: { "Content-Type": "application/json" }, // JSON body
33       body: JSON.stringify({ houseSize: Number(houseSize) }), // Send data
34     });
35
36     setHouseSize(""); // Reset input
37     setLoading(false); // Stop loading
38     fetchData(); // Refresh list
39   };
40 }
```

0 Git Graph Ln 73, Col 32 Spaces: 2

website > src > app > page.tsx > HomePage > data.map() callback

```
12 export default function HomePage() {
41 // DELETE: remove estimate by ID
42 const handleDelete = async (id: number) => {
43   await fetch(`/api/estimate/${id}`, { method: "DELETE" }); // Delete API
44   fetchData(); // Refresh list
45 };
46
47 // Run once when component mounts
48 useEffect(() => {
49   fetchData(); // Load initial data
50 }, []);
51
52 return (
53   <main style={styles.container}> {/* Main container */}
54     <h1>House Price Prediction</h1> {/* Title */}
55
56     <div style={styles.form}> {/* Input section */}
57       <input
58         type="number" // Numeric input
59         placeholder="House size" // Hint text
60         value={houseSize} // Controlled value
61         onChange={(e) => setHouseSize(e.target.value)} // Update state
62         style={styles.input} // Styling
63       />
64       <button onClick={handlePredict} style={styles.button}>
65         {loading ? "Loading..." : "Predict"} {/* Button text */}
66       </button>
67     </div>
68
69     <div style={styles.list}> {/* List container */}
70       {data.map((item) => (
71         <div key={item.id} style={styles.card}> {/* Card */}
72           <span>
73             Size: {item.houseSize} | Price: {item.price} {/* Display data */}
74           </span>
75           <button
76             style={styles.delete} // Delete button style
77             onClick={() => handleDelete(item.id)} // Delete handler
78           >
```