

Your Github URL

Your Vercel URL

W10-P1:

=>W10-P1: Copy mid1 m3_xx answer and rearrange it for W10 demo

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the file `product_supabase_14.html` with the following content:

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" c
6     <meta name="viewport" content="width=
7     <title>P3_14</title>
8     <link rel="stylesheet" href="./css/p
9   </head>
10  <body>
11    <div style="text-align: center; margin
12      <h2>m3_14: Products Demo</h2>
13      <h4 style="margin-top: -1.5rem">Li
14    </div>
15  </div>
16  <div class="products-container"></di
17  <script type="module" src="./js/prod
18  </body>
19 </html>
20

```

The browser window shows a products demo page titled "m3_14: Products Demo" by Liang-Ting Lin, 913410014. It features two products: "6-Entertainmint Center" and "7-Albany Sectional".

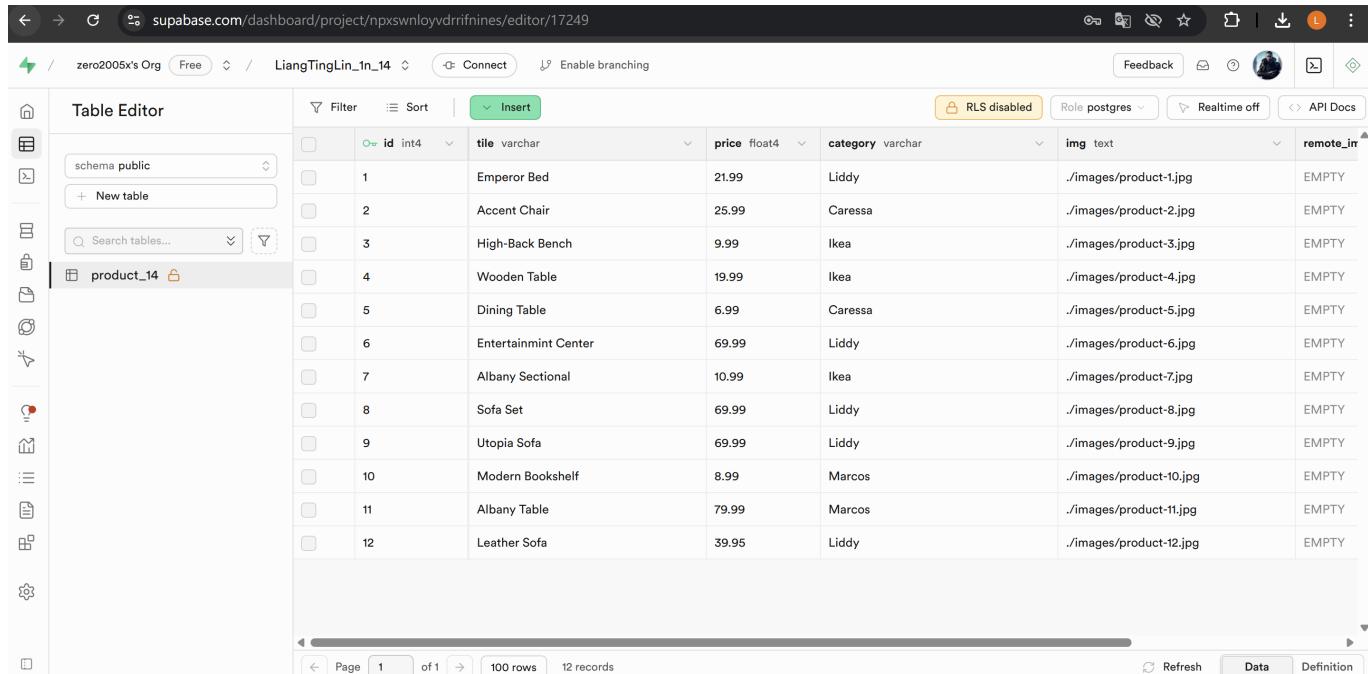
346347b%09zero2005x%09Thu Apr 24 18:42:17 2025 +0800

Copy mid1 m3_xx answer

and rearrange it for W10 demo

W10-P2: Setup Supabase with project LiangTingLin_1n_14

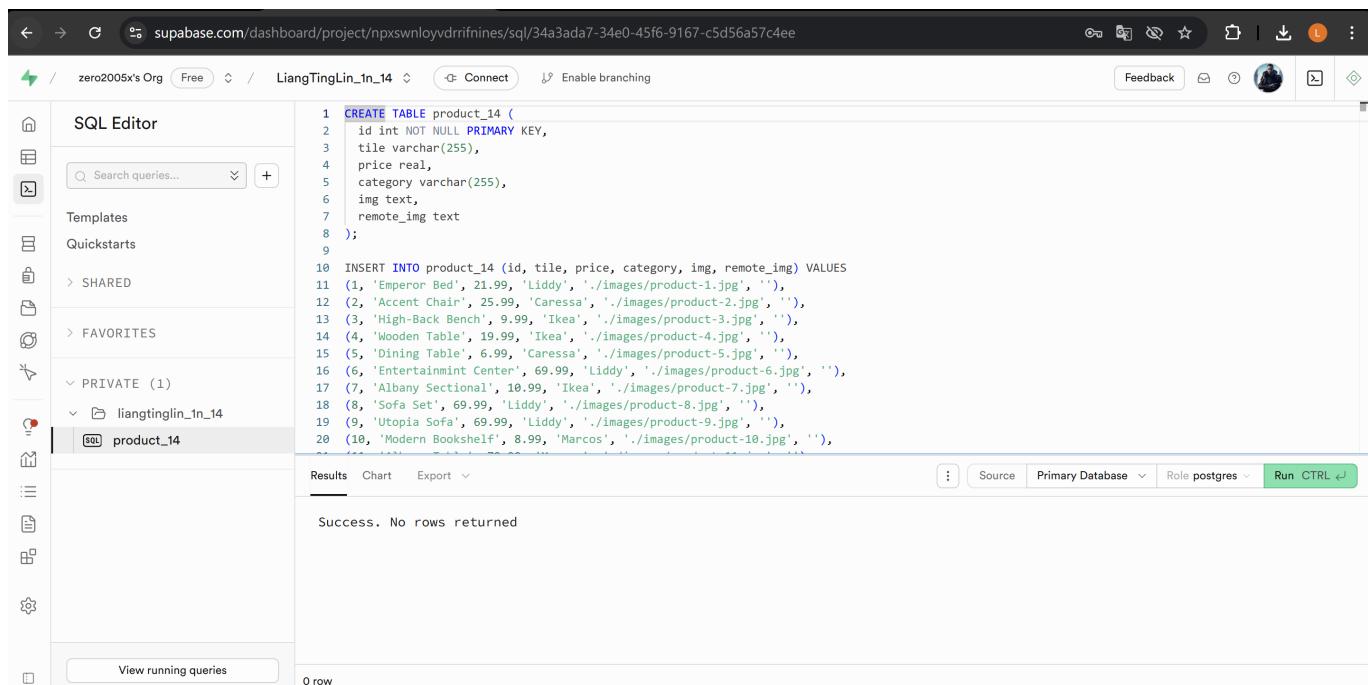
=> show product_14 table with 3 data



The screenshot shows the Supabase Table Editor interface. On the left is a sidebar with icons for home, database, table, search, and more. The main area is titled "Table Editor" and shows a table named "product_14". The table has columns: id (int4), tile (varchar), price (float4), category (varchar), img (text), and remote_img (text). The data consists of 12 rows of furniture items, each with a unique ID, name, price, category, and image URL.

	id	tile	price	category	img	remote_img
1	1	Emperor Bed	21.99	Liddy	./images/product-1.jpg	EMPTY
2	2	Accent Chair	25.99	Carella	./images/product-2.jpg	EMPTY
3	3	High-Back Bench	9.99	Ikea	./images/product-3.jpg	EMPTY
4	4	Wooden Table	19.99	Ikea	./images/product-4.jpg	EMPTY
5	5	Dining Table	6.99	Carella	./images/product-5.jpg	EMPTY
6	6	Entertainmint Center	69.99	Liddy	./images/product-6.jpg	EMPTY
7	7	Albany Sectional	10.99	Ikea	./images/product-7.jpg	EMPTY
8	8	Sofa Set	69.99	Liddy	./images/product-8.jpg	EMPTY
9	9	Utopia Sofa	69.99	Liddy	./images/product-9.jpg	EMPTY
10	10	Modern Bookshelf	8.99	Marcos	./images/product-10.jpg	EMPTY
11	11	Albany Table	79.99	Marcos	./images/product-11.jpg	EMPTY
12	12	Leather Sofa	39.95	Liddy	./images/product-12.jpg	EMPTY

=> show SQL to create product_14 table and insert three data



The screenshot shows the Supabase SQL Editor interface. On the left is a sidebar with icons for home, database, table, search, and more. The main area is titled "SQL Editor" and shows a query window with the following SQL code:

```

CREATE TABLE product_14 (
  id int NOT NULL PRIMARY KEY,
  tile varchar(255),
  price real,
  category varchar(255),
  img text,
  remote_img text
);

INSERT INTO product_14 (id, tile, price, category, img, remote_img) VALUES
(1, 'Emperor Bed', 21.99, 'Liddy', './images/product-1.jpg', ''),
(2, 'Accent Chair', 25.99, 'Carella', './images/product-2.jpg', ''),
(3, 'High-Back Bench', 9.99, 'Ikea', './images/product-3.jpg', ''),
(4, 'Wooden Table', 19.99, 'Ikea', './images/product-4.jpg', ''),
(5, 'Dining Table', 6.99, 'Carella', './images/product-5.jpg', ''),
(6, 'Entertainmint Center', 69.99, 'Liddy', './images/product-6.jpg', ''),
(7, 'Albany Sectional', 10.99, 'Ikea', './images/product-7.jpg', ''),
(8, 'Sofa Set', 69.99, 'Liddy', './images/product-8.jpg', ''),
(9, 'Utopia Sofa', 69.99, 'Liddy', './images/product-9.jpg', ''),
(10, 'Modern Bookshelf', 8.99, 'Marcos', './images/product-10.jpg', ''),
(11, 'Albany Table', 79.99, 'Marcos', './images/product-11.jpg', ''),
(12, 'Leather Sofa', 39.95, 'Liddy', './images/product-12.jpg', '');

```

The results section shows "Success. No rows returned" and "0 row".

=> show Project URL and ANON KEY for this project

The screenshot shows the Supabase API Settings page. On the left sidebar, under 'PROJECT SETTINGS', 'General', 'Compute and Disk', 'Infrastructure', 'Integrations', 'Add Ons', and 'Vault (Alpha)' are listed. Under 'CONFIGURATION', 'Database', 'Data API' (which is selected), 'Authentication', 'Storage', 'Edge Functions', and 'Log Drains' are listed. Under 'BILLING', 'Subscription' is listed. The main content area is titled 'API Settings' and contains a 'Project URL' section with a URL field containing 'https://npxswnloyvdrrifnines.supabase.co' and a 'Copy' button. Below it is a 'Project API Keys' section with two key types: 'anon' (public) and 'service_role' (secret). The 'anon' key is described as safe for use in a browser if Row Level Security is enabled. The 'service_role' key is described as having the ability to bypass Row Level Security and should never be shared publicly.

00fe629%09zero2005x%09Thu Apr 24 19:27:01 2025 +0800 Setup Supabase with project LiangTingLin_1n_14

W10-P3: Get 12 products from Supabase

=> show how to fetch products from Supabase

```

product_supabase_14 > js > JS products_supabase_14.js M | JS products_data_14.js M | product_supabase_14.html M | ...
8 const fetchProducts = async () => {
11   console.log('data', data);
12   if (error) throw error;
13   return data;
14 } catch (error) {
15   console.error("Error fetching products:", error);
16   return [];
17 }
18 }

20 > const displayProducts = (products) => {
40   };

41

42 document.addEventListener("DOMContentLoaded", async () => {
43   fetchProducts()
44     .then((products) => {
45       if (products.length > 0) {
46         displayProducts(products);
47       } else {
48         console.log("No products found in the database.");
49         displayProducts(products_14);
50       }
51     })
52     .catch((error) => {
53       console.error("Error fetching products:", error);
54       // Fallback to local data if there's an error
55       displayProducts(products_14);
56     });
57   displayProducts([products_14]);
58 });
59

```

The screenshot shows a browser window displaying a product listing page titled 'Get Products Demo From Supabase' by 'Liang-Ting Lin, 913410014'. The page features two chairs with colorful, patterned covers. Below the chairs, the text '1-Undefined' and '\$21.99' is displayed. To the right of the page is a developer tools panel showing the JavaScript code for 'products_14' and the resulting JSON data 'data' which contains 12 product objects. The developer tools also show the network tab with a successful request to '127.0.0.1:54869/demo/w10_product_supabase_14.html'.

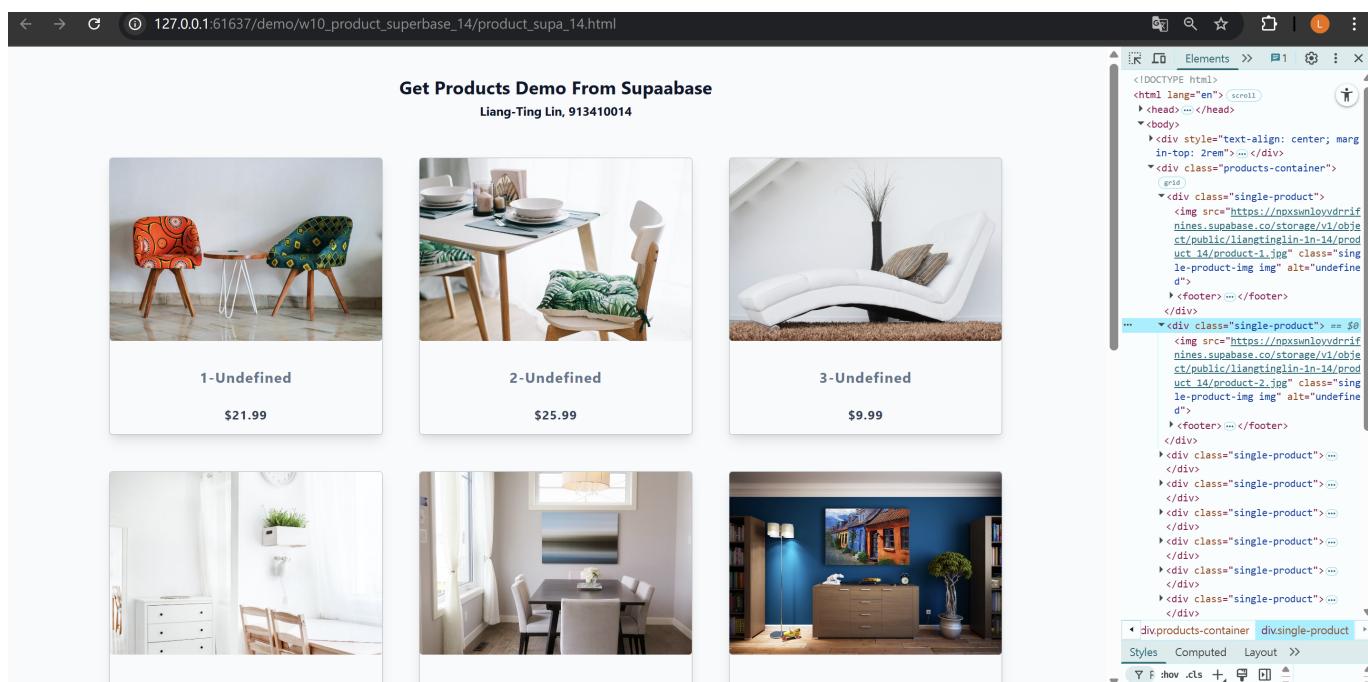
=> Show how to get supabase-js to create supabase client

```
File Edit Selection View Go Run ... 1132-1N-DEMO-14 08 □ □ - JS clientSupabase_14.js U ... product_supa_14.html M ... demo > w10_product_superbase_14 > JS clientSupabase_14.js > supabaseAuthListener > _supabase.auth.onAuthStateChange demo > w10_product_superbase_14 > product_supa_14.html > html > head > title 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <meta charset="UTF-8" /> 5 <meta http-equiv="X-UA-Compatible" content="IE=edge" /> 6 <meta name="viewport" content="width=device-width, initial-scale=1" /> 7 <title>Supabase Product Demo</title> 8 <link rel="stylesheet" href=".css/product_14.css" /> 9 </head> 10 <body> 11 <div style="text-align: center; margin-top: 2rem"> 12 <h2>Get Products Demo From Supabase</h2> 13 <h4 style="margin-top: -1.5rem">Liang-Ting Lin, 913410014</h4> 14 </div> 15 <div class="products-container"></div> 16 <script type="module" 17 src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js" 18 ></script> 19 <script type="module" src=".js/products_supa_14.js"></script> 20 </body> 21 </html> 22 23 24 25 26
```

File Edit Selection View Go Run ... 1132-1N-DEMO-14 08 □ □ - JS clientSupabase_14.js U ... product_supa_14.html M ... demo > w10_product_superbase_14 > JS clientSupabase_14.js > supabaseAuthListener > _supabase.auth.onAuthStateChange demo > w10_product_superbase_14 > product_supa_14.html > html > head > title 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <meta charset="UTF-8" /> 5 <meta http-equiv="X-UA-Compatible" content="IE=edge" /> 6 <meta name="viewport" content="width=device-width, initial-scale=1" /> 7 <title>Supabase Product Demo</title> 8 <link rel="stylesheet" href=".css/product_14.css" /> 9 </head> 10 <body> 11 <div style="text-align: center; margin-top: 2rem"> 12 <h2>Get Products Demo From Supabase</h2> 13 <h4 style="margin-top: -1.5rem">Liang-Ting Lin, 913410014</h4> 14 </div> 15 <div class="products-container"></div> 16 <script type="module" 17 src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js" 18 ></script> 19 <script type="module" src=".js/products_supa_14.js"></script> 20 </body> 21 </html> 22 23 24 25 26

W10-P4: Upload 12 images to Supabase storage and show these remote images

=> show the first two images are from Supabase storage



=> show remote_img in product_xx table

```

File Edit Selection View Go Run ...
demo > w10_product_supabase_14 > js > JS products_supabase_14.js > displayProducts > productsContent > products.map() callback
8 const fetchProducts = async () => {
17 }
18 };
19
20 const displayProducts = (products) => {
21   let productsContent = products
22   .map((product) => [
23     const { id, title, price, img, remote_img } = product;
24     return (
25       <div class="single-product">
26         <img
27           src=${remote_img ? remote_img : img}
28           alt=${title}
29         />
30         <footer>
31           <h3 class="name">${id}- ${title}</h3>
32           <span class="price">${price}</span>
33         </footer>
34       </div>
35     );
36   );
37   .join("");
38   productContainer.innerHTML = productsContent;
39 };
40 };
41
42 document.addEventListener("DOMContentLoaded", async () => {
43   fetchProducts()
44   .then((products) => {
45     if (products.length > 0) {
46       displayProducts(products);
47     } else {
48       console.log("No products found in the database.");
49       displayProducts(products_14);
50     }
51   });
52 });

```

=> show the js code to use remote_img instead of img

File
product-1.jpg
product-2.jpg
product-3.jpg
product-4.jpg
product-5.jpg
product-6.jpg
product-7.jpg
product-8.jpg
product-9.jpg
product-10.jpg
product-11.jpg
product-12.jpg

0bb8034%09zero2005x%09Thu Apr 24 20:45:09 2025 +0800 W10-P4: Upload 12 images to Supabase storage and show these remote images

W10-log: git logs for W10

Commits

main

All users All time

Commits on Apr 24, 2025

- W10-P4: Upload 12 images to Supabase storage and show these remote images
zero2005x committed now · ✓ 1 / 1 0bb8034
- Get 12 products from Supabase
zero2005x committed 32 minutes ago · ✓ 1 / 1 8a0f3fa
- Setup Supabase with project LiangTingLin_1n_14
zero2005x committed 1 hour ago · ✓ 1 / 1 00fe629
- Resolve merge conflict on exams/rename.sh
zero2005x committed 2 hours ago · ✓ 1 / 1 8040752
- Copy mid1 m3_xx answer and rearrange it for W10 demo
zero2005x committed 2 hours ago 346347b

Commits on Apr 17, 2025

- fix m1. m2. m3