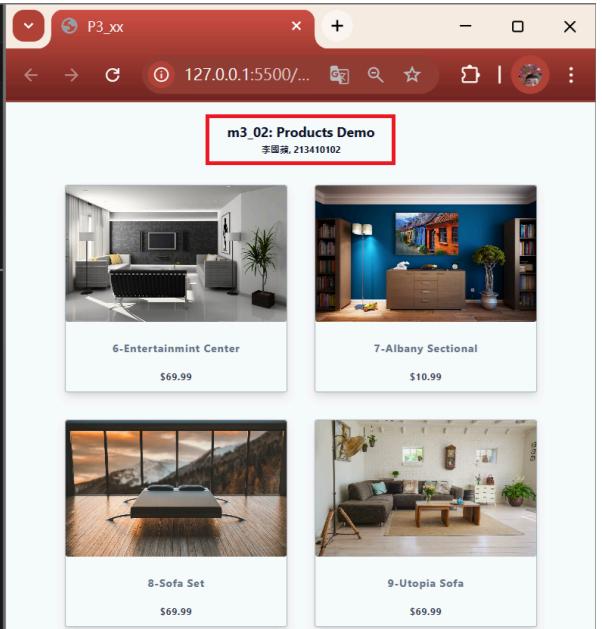


```
git config --global user.email "apple0974199326@gmail.com"
```

```
git config --global user.name "apple550678"
```

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



The screenshot shows a Microsoft Edge browser window with the title 'P3_xx'. The page content is a product catalog with four items:

- 6-Entertainment Center \$69.99
- 7-Albany Sectional \$10.99
- 8-Sofa Set \$69.99
- 9-Utopia Sofa \$69.99

The browser's address bar shows the URL '127.0.0.1:5500/...'. Below the browser is a code editor window for 'product_sup_02.html'. A red box highlights the following code in the body section:

```
<div style="text-align: center; margin-top: 2rem">
  <h2>m3_02: Products Demo</h2>
  <h4 style="margin-top: -1.5rem">李國蘋, 213410102</h4>
</div>
```

1a8f36d apple550678

Thu Apr 24 18:46:18 2025 +0800 Video: W10-P1: Copy mid1 m3_xx answer ar

Video: W10-P2: Setup Supabase with project yourcode_1n_xx

=> show product_xx table with 3 data

The screenshot shows the Supabase Table Editor interface. At the top, there are two project dropdowns: 'apple_lee' (Free) and 'apple_lee_1n_02'. Below them is a 'Table Editor' sidebar with icons for schema, table creation, search, and refresh. A red box highlights the 'product_02' table entry in the sidebar. The main area displays the 'product_02' table data:

	id	title	price	category	img
1	Emperor Bed	21.99	Liddy	./images/product-1.jpg	
2	Accent Chair	25.99	Caressa	./images/product-2.jpg	
3	High-Back Bench	9.99	Ikea	./images/product-3.jpg	

Below the table, the status bar shows 'Page 1 of 1', '100 rows', '3 records', and system icons for refresh, data, and definition.

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

The screenshot shows the Supabase SQL Editor interface. On the left, there's a sidebar with navigation links like Home, Templates, Quickstarts, and a list of databases and tables. The main area is the SQL Editor, which contains the following SQL code:

```
1 CREATE TABLE product_02 (
2     id INT NOT NULL PRIMARY KEY,
3     title varchar(255),
4     price real,
5     category varchar(255),
6     img text,
7     remote_img text
8 );
9
10 INSERT INTO product_02(id, title, price, category, img, remote_img)
11 VALUES
12 (1, 'Emperor Bed', 21.99, 'Liddy', './images/product-1.jpg', ''),
13 (2, 'Accent Chair', 25.99, 'Caressa', './images/product-2.jpg', ''),
14 (3, 'High-Back Bench', 9.99, 'Ikea', './images/product-3.jpg', '')
15
16 DROP TABLE product_02
```

The code is highlighted with syntax coloring. A red box highlights the entire SQL statement. Another red box highlights the table name 'product_02' in the sidebar and the results panel.

In the results panel, it says "Success. No rows returned" and shows "0 row". The status bar at the bottom right indicates the time is 07:33 and the date is 2025/4/24.

=> show Project URL and ANON KEY for this project

The screenshot shows the Supabase API Settings page. On the left, a sidebar lists 'PROJECT SETTINGS' (General, Compute and Disk, Infrastructure, Integrations, Add Ons, Vault), 'CONFIGURATION' (Database, Data API, Authentication, Storage, Edge Functions, Log Drains), and 'BILLING'. A red box highlights the 'Data API' section under Configuration. A red circle highlights the gear icon at the bottom of the sidebar. The main area is titled 'API Settings' and contains two sections: 'Project URL' and 'Project API Keys'. The 'Project URL' section shows the URL <https://szkizgkwbthcdfrjhba.supabase.co> with a 'Copy' button. The 'Project API Keys' section shows two keys: 'anon' (public) and 'service_role' (secret). The 'secret' key is described as safe to use in a browser if Row Level Security is enabled. Both keys have 'Copy' buttons. A red box highlights the 'Project API Keys' section.

8aa10e6 apple550678

Thu Apr 24 19:36:32 2025 +0800 Video: W10-P2: Setup Supabase with project

Video: W10-P3: Get 12 products from Supabase

=> show how to fetch products from Supabase

The screenshot shows a development environment with two main panes: a code editor and a browser window.

Code Editor (VS Code):

- File Structure:** Explorer sidebar shows files like `w02_dom_02`, `demo`, `w03_tictactoe_02`, etc., under `1132-1N-DEMO-02`.
- Current File:** `product_supabase_02.js` (marked as modified).
- Code Content:** The file contains a function `fetchProducts` which uses `supabase` to select data from the `'product_02'` table. It also includes a `displayProducts` function to handle the fetched data.

Browser Output:

- Title:** Supabase Product
- URL:** 127.0.0.1:5500/demo/w10_p...
- Content:** A product listing page showing three items:
 - 1-Emperor Bed**: \$21.99
 - 2-Accent Chair**: \$5.99
 - 3-High-Back Bench**
- Console Output (DevTools):** Shows the JSON data returned by the `fetchProducts` function, listing 12 products with details like title, price, and ID.

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

The screenshot shows the VS Code interface with two open files:

- product_supabase_02.js**: Contains the logic to fetch products from Supabase. It includes a script tag for importing supabase-js and a call to `supabase.createClient`.
- clientSupabase_02.js**: Contains a long, obfuscated API key for Supabase.

Both files are highlighted with red boxes, indicating they are the focus of the demonstration.

```
const SUPABASE_URL = 'https://szkizgkwbwhcfrjhba.supabase.co'
const SUPABASE_ANON_KEY =
  'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
  eyJpc3MiOiJzdXBhYmFzZSIsInJlZii6InN6a216Zt3YnR3
  aGNKZnjqagJhIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDU0
  OTE0MTNsImV4cC16MjA2MTA2NzQzM30.
  9oToaZc3_BTZdLVx_jNDAZVE5nxoUyUdC4t6y75ngU'

export const supabase = supabase.createClient(
  SUPABASE_URL, SUPABASE_ANON_KEY)
```

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Supabase Products Demo</title>
    <link rel="stylesheet" href="./css/product_02.css" />
  </head>

  <body>
    <div style="text-align: center; margin-top: 2rem">
      <h2>Get Products from Supabase</h2>
      <h4 style="margin-top: -1.5rem">李國蘋, 213410102</h4>
    </div>
    </div>
    <div class="products-container"></div>
    <script type="module" src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js"></script>
    <script type="module" src="./js/product_supabase_02.js"></script>
  </body>
</html>
```

6364ed1 apple550678

Sun Apr 27 21:32:53 2025 +0800 Video: W10-P3: Get 12 products from Supabase

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

=> show the first two images are from Supabase storage

The screenshot shows a web browser window with three tabs, all titled "Supabase Products Demo". The active tab displays a product catalog page at the URL 127.0.0.1:5500/demo/w10_product_supabase_02/product_supabase_02.html. The page title is "Get Products from Supabase" and includes a reference number: 213410102.

The catalog lists twelve products in a grid:

- 1-Emperor Bed (\$21.99)
- 2-Accent Chair (\$5.99)
- 3-High-Back Bench (\$9.99)
- 4-Wooden Table (\$19.99)
- 5-Dining Table (\$6.99)
- 6-Entertainment Center (\$69.99)
- 7-Albany Sectional (\$10.99)
- 8-Sofa Set (\$69.99)
- 9-Utopia Sofa (\$69.99)

The first two products, "Emperor Bed" and "Accent Chair", have their image URLs highlighted with red boxes in the browser's developer tools. The DevTools panel shows the HTML code for the page, with the image URLs for these two items explicitly pointing to Supabase storage:

```
<!DOCTYPE html>
<html lang="en"> <!-- scroll -->
  <head> ... </head>
  <body>
    <div style="text-align: center; margin-top: 2rem">
      <h2>Get Products from Supabase</h2>
      <h4 style="margin-top: -1.5rem">李國楓, 213410102</h4>
    </div>
    <div class="products-container" style="grid">
      <div class="single-product">
        
        <div> ...
      </div>
      <div class="single-product" style="grid">
        
        <div> ...
      </div>
      ...
    </div>
    <div> ...
    </div>
  </body>
</html>
```

The DevTools also shows the "element.style" section, which is currently empty.

=> show remote_img in product_xx table

The screenshot shows the Supabase Storage interface. On the left, the sidebar has 'Storage' selected. Under 'ALL BUCKETS', the bucket 'apple_lee_1n_02' is listed, with its dropdown menu open and 'Public' selected. The main area shows a folder named 'product_02' which contains twelve files: 'product-10.jpg', 'product-11.jpg', 'product-12.jpg', 'product-1.jpg' (which is highlighted with a red box), 'product-2.jpg' (also highlighted with a red box), 'product-3.jpg', 'product-4.jpg', 'product-5.jpg', 'product-6.jpg', 'product-7.jpg', 'product-8.jpg', and 'product-9.jpg'. To the right of the list, there is a preview image of two chairs with patterned seats and wooden legs. Below the preview, details for 'product-1.jpg' are shown: 'product-1.jpg', 'image/jpeg - 61.39 KB', 'Added on 2025/4/27 下午9:37:50', and 'Last modified 2025/4/27 下午9:37:50'. At the bottom right of the preview area, there are 'Download' and 'Get URL' buttons, with 'Get URL' also highlighted with a red box.

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

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** pane on the left:
 - Open Editors: 1132-1N-DEMO-02
 - demo
 - md
 - w01_dom_02
 - w02_dom_02
 - w02_tictactoe_02
 - w03_basics_02
 - w03_tictactoe_02
 - w04_basics_02
 - w05_basics_02
 - w06_prep_tests_02
 - w10_mid1_p3_02
 - w10_product_supabase_02
 - css
 - # product_02.css
 - images
 - js
 - clientSupabase_02.js
 - product_02.sql
 - product_supabase_02.js** (selected)
 - products_data_02.js
 - product_supabase_02.html
 - js
 - scss
- Editor** pane in the center:

```
const fetchProducts = async () => {
  try {
    const response = await fetch('https://api.supabase.com/v1/products');
    const products = await response.json();
    return products;
  } catch (err) {
    console.log(err);
  }
}

const displayProducts = (products) => {
  let productsContent = products
    .map((product) => {
      const { id, title, price, remote_img } = product
      return (
        <div class="single-product">
          <img
            src={`${remote_img}`}
            class="single-product-img img"
            alt={`${title}`}
          />
          <h3 class="name">${id}-${title}</h3>
          <span class="price">${price}</span>
        </div>
      );
    })
    .join('');
  return productsContent;
}
```

The code in the editor is a JavaScript function `displayProducts` that takes an array of products and returns an HTML string. It uses template literals to construct the HTML. Two specific parts of the code are highlighted with red boxes: the `src` attribute of the `img` tag and the template literal `\${remote_img}`.
- TERMINAL** pane at the bottom:

```
apple@apple-lee:~/Desktop/1132-1N-demo-02 (main)
```

f12d88c apple550678 Sun Apr 27 21:58:21 2025 +0800 Video: W10-P4: Upload 12 images to Supal