

Github URL

Github URL for Vercel

Vercel URL

Video: W12-P1: Refine code of /demo/shop_xx/node

The screenshot shows a Windows desktop environment. On the left is a Microsoft Visual Studio Code (VS Code) interface. The Explorer sidebar shows a project structure for 'W12_NEXT_FULLSTACK_02' with several folders like 'client_02', 'public', 'src', and 'app'. A red box highlights the 'category' folder under 'app'. Inside 'category', there are files: 'page.jsx' (which is the active editor), 'layout.js', and 'page.js'. The code in 'page.jsx' is focused on fetching shop data from a Node.js API based on the category parameter. On the right side of the screen is a web browser window displaying a shopping website. The URL bar shows '/demo/shop_02/node/womens'. The page content includes a header with 'TKUdemo', 'Shop', 'Contact', and 'Sig' links, and a section titled 'Womens' showing three items: 'Floral Blouse', 'Floral Dress', and 'Red Dots Dre:'. The status bar at the bottom of the screen shows the date and time: '2025/12/7 17:31:22 +0800'.

```
client_02 > src > app > demo > shop_02 > node > [category] > [page.jsx] > FetchShopByCategory_02 > [fe]
  1 'use client'
  2
  3 import { useState, useEffect } from 'react'
  4 import { useRouter } from 'next/navigation'
  5 import Wrapper from '../../../../../wrapper/Shop_02'
  6 import Product_02 from '../../../../../_components/Product_02'
  7
  8 const FetchShopByCategory_02 = () => {
  9   const [shop_02, setShop_02] = useState([])
 10   const params = useParams()
 11   const category = params.category
 12   console.log('category', category)
 13
 14   const fetchShopFromNode = async () => {
 15     try {
 16       const response = await fetch(
 17         `http://localhost:5001/api/shop2_02/${category}`
 18       )
 19       const data = await response.json()
 20       console.log('shop_02 data', data)
 21       if (data.length !== 0) {
 22         setShop_02(data)
 23       }
 24     } catch (err) {
 25       console.log(err)
 26     }
 27   }
 28
 29   useEffect(() => {
 30     if (category) {
 31       fetchShopFromNode()
 32     }
 33   }, [category])
 34 }
```

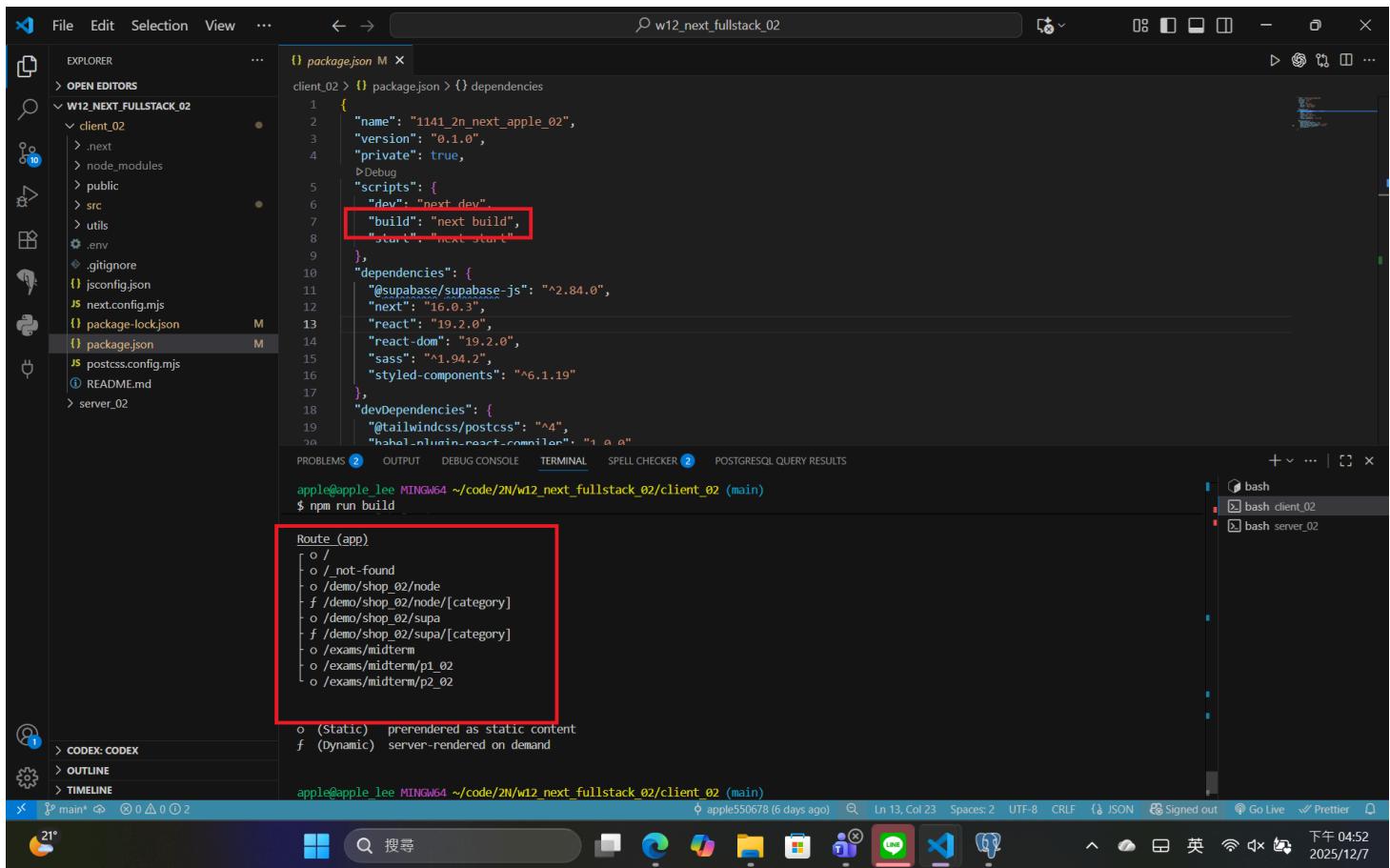
42a2d41 apple550678

2025-12-07 17:31:22 +0800

Video: W12-P1: Refine code of /demo/shop

W12-P2: Deploy code to Vercel

=> npm run build



The screenshot shows a Windows desktop environment with VS Code open. The file structure in the Explorer sidebar includes .next, node_modules, public, src, utils, .env, .gitignore, jsonconfig.json, next.config.mjs, package-lock.json, package.json, postcss.config.mjs, README.md, and server_02. The package.json file is selected in the Explorer.

In the package.json file, the "scripts" section is expanded, showing the "dev", "build", and "start" scripts. The "build" script is highlighted with a red box.

The terminal window at the bottom shows the command \$ npm run build being run, followed by the output:

```
Route (app)
o /
o ./not-found
o /demo/shop_02/node
f /demo/shop_02/node/[category]
o /demo/shop_02/supa
f /demo/shop_02/supa/[category]
o /exams/midterm
o /exams/midterm/p1_02
o /exams/midterm/p2_02
```

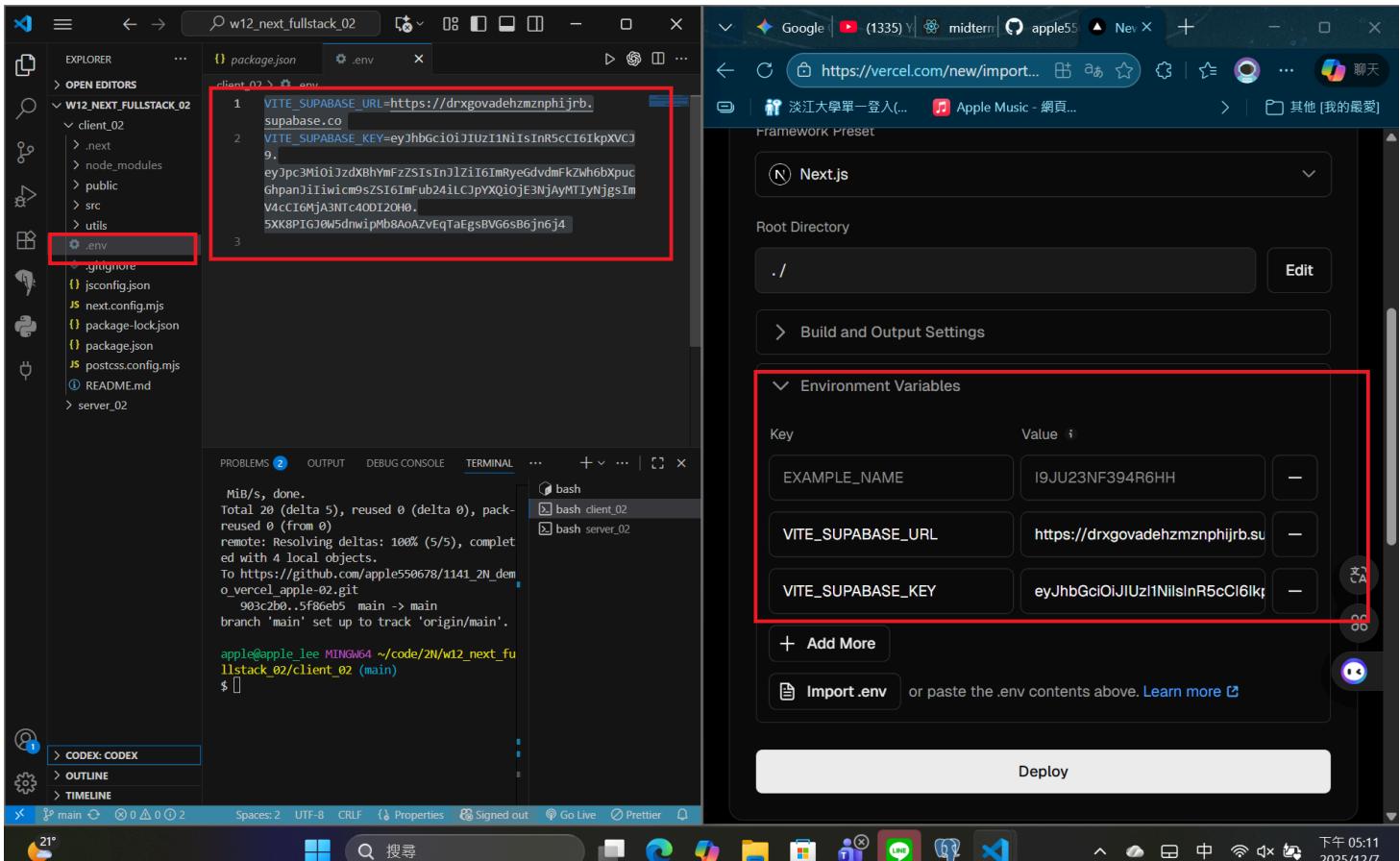
A red box highlights the route tree output in the terminal. The status bar at the bottom indicates the user is on a Mac (apple) with a MINGW64 shell, and the date and time are shown as 2025/12/7 04:52.

=> npm start

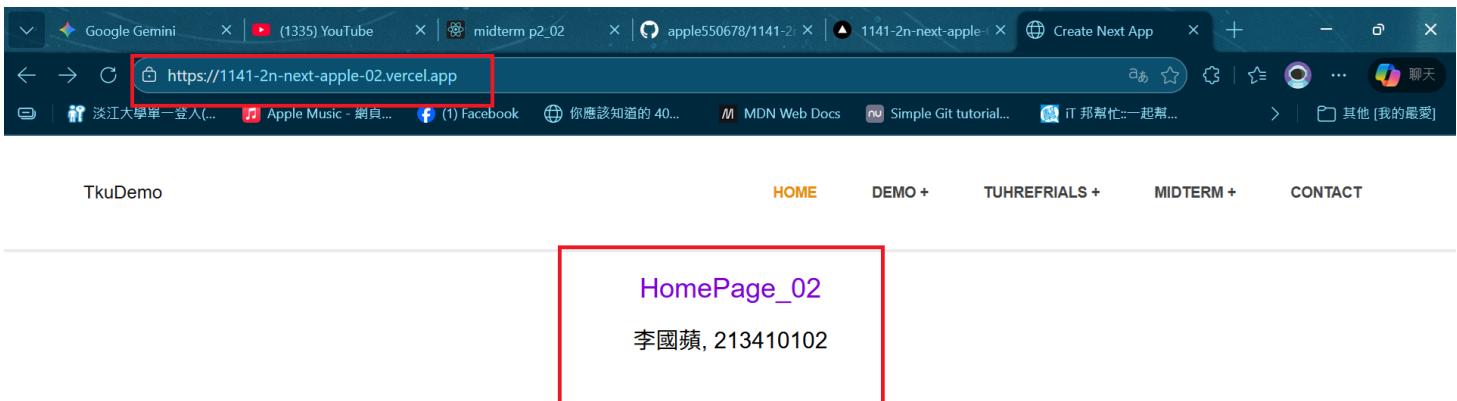
The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure for "W12_NEXT_FULLSTACK_02". A red box highlights the "client_02" folder, which contains several JSON files like ".next", "BUILD_ID", "app-path-routes-manifest.json", etc., and other files like "public", "src", "utils", ".env", ".gitignore", "jsconfig.json", "next.config.mjs", "package-lock.json", "package.json", "postcss.config.mjs", "README.md", and "server_02".
- Code Editor:** The "package.json" file is open. A red box highlights the "scripts" section, specifically the "start" key which is set to "next start".
- Terminal:** The terminal window shows the command \$ npm start being run, followed by the output: "1141_2n_next_apple_02@0.1.0 start > next start". Below this, it shows the Next.js version (16.0.3) and the URLs for local and network access. It also includes a warning about inferred workspace root and lockfiles.
- Bottom Status Bar:** Shows the current user (apple), the date (6 days ago), file status (main*), and system information including battery level, signal strength, and the date (2025/12/7).

=> Vercel, add environment variables



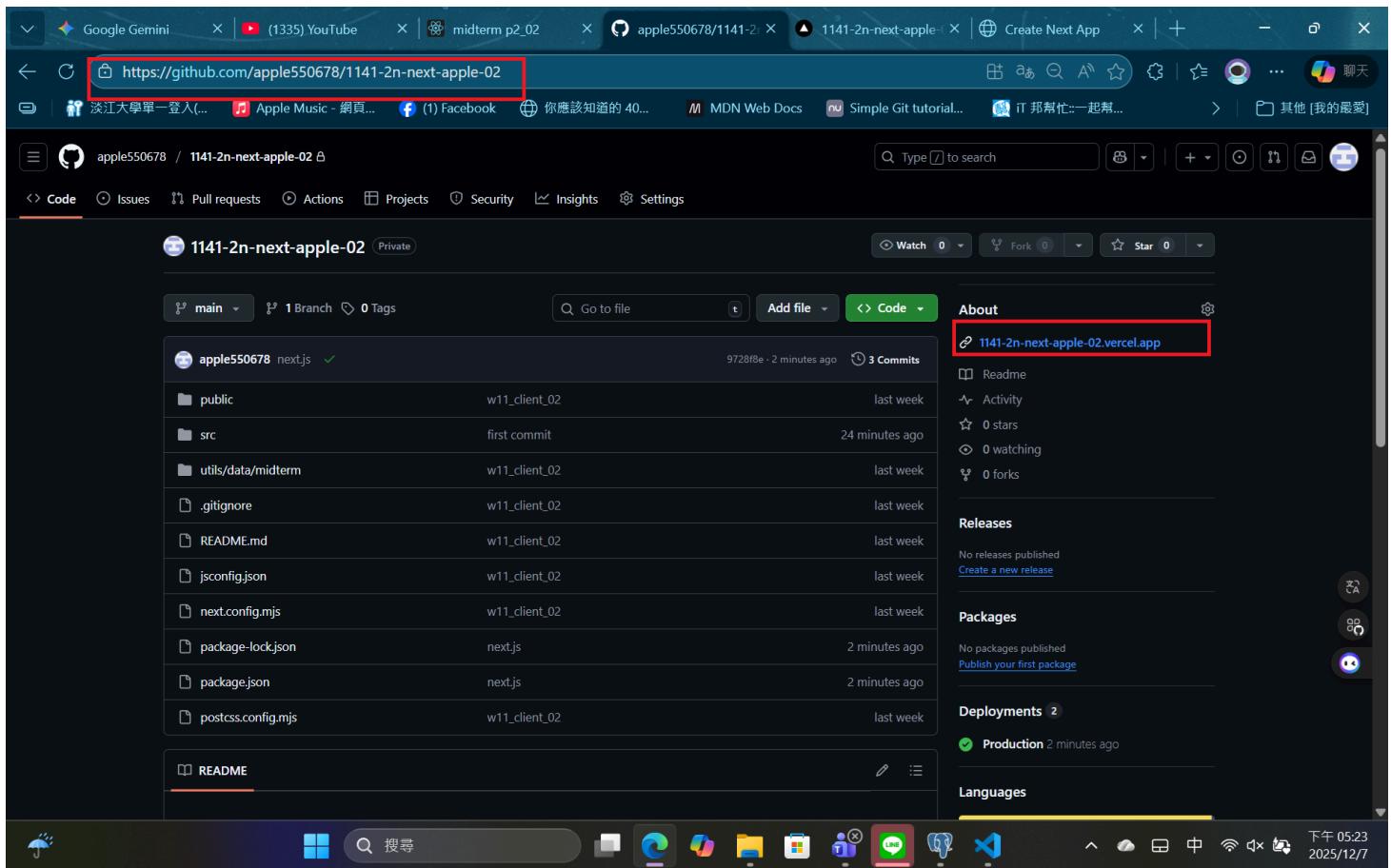
=> Vercel homepage success



=> Github repo and Vercel URL

Gitbu Next URL

Vercel Next URL



=> Share to teacher and TA

htchung@gms.tku.edu.tw

sian-0018

The screenshot shows a GitHub repository settings page for a private repository. The left sidebar is titled 'Access' and includes sections for 'Code and automation', 'Security', and 'Integrations'. The 'Collaborators' section is highlighted with a red box. The main content area shows a 'Direct access' section indicating 2 entities have access, and a 'Manage access' section where two pending invitees are listed: 'Hsingtai Chung' and 'sian-0018', both awaiting responses. A red box highlights the list of pending invitees.

Collaborators

Private repository
Only those with access to this repository can view it

Manage visibility

Code and automation

- Branches
- Tags
- Rules
- Actions
- Models
- Webhooks
- Copilot
- Environments
- Codespaces
- Pages

Security

- Advanced Security
- Deploy keys
- Secrets and variables

Integrations

- Github Apps
- Email notifications

Manage access

Add people

Select all Type

Find a collaborator...

Hsingtai Chung Awaiting htchung's response Pending Invite

sian-0018 Awaiting sian-0018's response Pending Invite

Previous Next

© 2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information

bac54c9 apple550678

2025-12-07 17:33:55 +0800

W12-P2: Deploy code to Vercel

Video: W12-P3: Implement /demo/shop2_xx/supabase to fetch products from Supabase

=> use SQL to get category2_xx data

The screenshot shows the Supabase dashboard with an SQL Editor window open. The query being run is:

```
category2_02, shop2_02
select * from category2_02;
```

The results table displays the following data:

cid	cname	size	image_url	remote_image_url	link_u
1	hats	NULL	/images/midterm/homepage/hats.png	https://i.ibb.co/cvpnL1/hats.png	/demo/
2	jackets	NULL	https://i.ibb.co/px2tCc3/jackets.png	/images/midterm/homepage/jackets.png	/demo/
3	sneakers	NULL	https://i.ibb.co/0jqHpn/sneakers.png	/images/midterm/homepage/sneakers.png	/demo/
4	womens	large	https://i.ibb.co/GCCdy8t/womens.png	/images/midterm/homepage/womens.png	/demo/
5	mens	large	https://i.ibb.co/R70vBrQ/men.png	/images/midterm/homepage/mens.png	/demo/

Below the table, it says "5 rows (Limited to only 100 rows)".

=> use SQL to get shop2_xx data

The screenshot shows the Supabase dashboard with the SQL Editor open. The query being run is:

```
category2_02, shop2_02
select * from category2_02;
select * from shop2_02;
```

The results table displays 35 rows of data from the shop2_02 table, with columns: pid, pname, cat_id, price, img_url. The results table is highlighted with a red border.

pid	pname	cat_id	price	img_url
29	White Blouse	4	20	/images/midterm/womens/white-vest.png
30	Camo Down Vest	5	325	/images/midterm/mens/camo-vest.png
31	Floral T-shirt	5	20	/images/midterm/mens/floral-shirt.png
32	Black & White Longsleeve	5	25	/images/midterm/mens/long-sleeve.png
33	Pink T-shirt	5	25	/images/midterm/mens/pink-shirt.png
34	Jean Long Sleeve	5	40	/images/midterm/mens/roll-up-jean-shirt.png
35	Burgundy T-shirt	5	25	/images/midterm/mens/polka-dot-shirt.png

=> set foreign key

The screenshot shows the Supabase Table Editor interface. On the left, there's a sidebar with 'Table Editor' and a list of tables: schema public, New table, blog_02, category2_02, and shop2_02. The shop2_02 table is selected, displaying 12 rows of data with columns pid, pname, cat_id, price, img_url, and remote_img_u.

A modal window titled "Add foreign key relationship to shop2_02" is open. It shows a mapping from the shop2_02 table's "cat_id" column to the category2_02 table's "cid" column. A red box highlights this mapping. Below it, a question asks "Which action is most appropriate?", with "Cascade" selected. Another red box highlights "Cascade". Further down, another red box highlights "Set NULL" under "Action if referenced row is removed".

At the bottom right of the modal, there are "Cancel" and "Save" buttons.

The browser address bar shows the URL: <https://supabase.com/dashboard/project/drxdgovadehzmznphijrb/editor/60282?schema=public>.

=> set RLS policies for public access for all CRUD

The screenshot shows the Supabase dashboard with a red box highlighting the 'Policy Name' field containing 'public access for all CRUD operations'. Below it, the 'Table on clause' dropdown is set to 'public.shop2_02'. A second red box highlights the 'Policy Behavior as clause' dropdown set to 'Permissive' and the 'Policy Command for clause' dropdown with 'ALL' selected. On the right, a sidebar titled 'Templates' lists several RLS policy options, with the 'Enable read access for all users' template selected. The bottom of the screen shows a Windows taskbar with various icons.

Policy Name: public access for all CRUD operations

Table on clause: public.shop2_02

Policy Behavior as clause: Permissive

Policy Command for clause: ALL

Target Roles to clause: Defaults to all (public) roles if none selected

```
1 create policy "public access for all CRUD operations"
2 on "public"."shop2_02"
3 as PERMISSIVE
4 for ALL
5 to public
6 using (
7   true
8 );
```

Use check expression

Templates

- SELECT** Enable read access for all users
- INSERT** Enable insert for authenticated users only
- DELETE** Enable delete for users based on user_id
- UPDATE** Enable update for users based on user_id

=> Chrome _ local

Screenshot of a web browser showing a shopping page for women's clothing. The URL in the address bar is `localhost:3000/demo/shop_02/supa/womens`. The page displays several items with images and descriptions. A red box highlights the user information at the top: "李國蘋, 213410102". Another red box highlights the "Womens" category title. The developer tools' "Elements" tab is open, showing the "data" section of the "page.jsx:22" file, which contains an array of product objects. A third red box highlights this data array.

The data array from the developer tools:

```
data
  ▼ (7) [ { }, { }, { }, { }, { }, { }, { } ]
    ▶ 0: { pid: 23, pname: 'Blue Tanktop', price: 80 }
    ▶ 1: { pid: 24, pname: 'Floral Blouse', price: 20 }
    ▶ 2: { pid: 25, pname: 'Floral Dress', price: 25 }
    ▶ 3: { pid: 26, pname: 'Red Dots Dress', price: 80 }
    ▶ 4: { pid: 27, pname: 'Striped Sweater', price: 25 }
    ▶ 5: { pid: 28, pname: 'Yellow Track Suit', price: 80 }
    ▶ 6: { pid: 29, pname: 'White Blouse', price: 25 }
    length: 7
    [[Prototype]]: Array(0)
```

=> Chrome _ Vercel

The screenshot shows a web browser window with multiple tabs open. The active tab displays a shopping website for women's apparel. The URL in the address bar is https://1141-2n-next-apple-02.vercel.app/demo/shop_02/supa/womens. The page header includes a user profile section with the name "李國蘋, 213410102". Below this, the category "Womens" is centered. Four product cards are displayed in a row: "Blue Tanktop" (a woman in a green tank top), "Floral Blouse" (a woman in a white floral off-the-shoulder blouse), "Floral Dress" (a woman in a white floral dress), and "Red Dots Dress" (a woman in a red polka-dot dress). At the bottom of the page, there are two more product cards: "20°" (a woman in a brown and white patterned top) and "20" (a woman in a yellow hoodie and pants). The browser interface includes a search bar, a taskbar with various icons, and a system tray showing the date and time (下午 08:44, 2025/12/7).

85488c4 apple550678

2025-12-07 20:47:27 +0800

Video: W12-P3: Implement /demo/shop2_x