

(75%) 113-2 JavaScript 期末考 1 -- 斷網考試

2025-06-12, at E201, from 18:20~21:00

Note:

1. 本期末考分兩部分，斷網考試及開放網路考試。
2. 請不要發揮同學愛，作弊雙方除了本次考試 0 分外，平常分數另扣 20 分，情節嚴重者會送校。
3. iClass 上請繳交 md1_xx.pdf, final_md1.zip, final_client_xx.zip, final_server_xx.zip，壓縮前請將 node_modules 刪除
4. 請直接將答案寫在 md1_xx.md 上，老師出題及圖片放在 md1_htc.pdf 上，請依照老師所給的圖片來實作並標註
5. 跟小考相關的檔案及目錄名稱有 xx 時，必須要改成學號後 2 碼，沒有修改時，會視違犯情況扣分。
6. 每一張圖片要有機房左側背景，圖片上要有你的學號(或後兩碼)，圖片標註要跟老師所標註的類似。違者會依情節扣分。
7. 請自評分數，將每一題的 ? 填入分數，沒有填者，不會批改，以 0 分計算。

Your (Name, ID): (?, ?)

Part 1: 斷網考試 (75%)

- P1 (15%): ? 分
- P2 (15%): ? 分
- P3 (15%): ? 分
- P4 (10%): ? 分
- P5 (10%): ? 分
- P6 (10%): ? 分

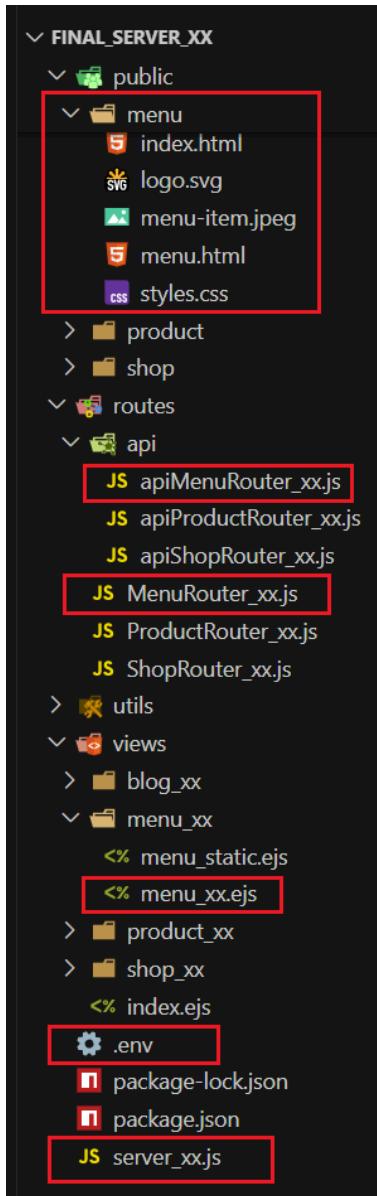
總分: ? 分

Part 2: 開放考試 (25%)

- P7 (15%): ? 分
- P8 (10%): ? 分

Note: 本次考試前端及後端主要以 menu theme 為主來實作，但考題涵蓋了上課 product_xx demo 及小考 2 shop_xx 的部分

檔案及目錄結構如下圖



(15%) P1: 請用本地端的 pgAdmin 來建立 final_js_xx 資料庫及 table menu_xx

請以你的學號最後一個數字 (D)，來決定要實作哪一個分類的頁面。

分類：

breakfast: mid = 1, 4, 7, 10

lunch: mid = 2, 5, 8

dessert: mid = 3, 6, 9

D=1,6 => breakfast(1,4), dessert(6,9)

D=2,7 => lunch(2,5), breakfast(7,10)

D=3,8 => dessert(3,6), lunch(8,2),

D=4,9 => breakfast(4,7), dessert(9,3)

D=5,0 => lunch(5,8), breakfast(10,1)

本次考試，老師題目顯示以

D=4,9 breakfast(4,7), dessert(9,3) 四筆資料

為主來解說。

The screenshot shows the pgAdmin 4 interface. On the left, the object browser displays the database 'final_js_xx' and the table 'menu_xx'. The table has columns: mid [PK] integer, title character varying (255), category character varying (255), price real, local_img text, remote_img text, and description text. The data grid shows four rows:

| mid [PK] | title | category | price | local_img | remote_img | description |
|----------|----------------------|-----------|-------|--------------------------|------------|---------------------------------|
| 1 | 3 godzilla milkshake | dessert | 6.99 | /menu/images/item-3.jpeg | | Artisan pop-up truffaut pug t |
| 2 | 4 country delight | breakfast | 20.99 | /menu/images/item-4.jpeg | | Keffiyeh helvetica normcore |
| 3 | 7 bacon overflow | breakfast | 8.99 | /menu/images/item-7.jpeg | | Carry jianbing viral, poke farr |
| 4 | 9 quarantine buddy | dessert | 16.99 | /menu/images/item-9.jpeg | | Artisan pop-up truffaut pug t |

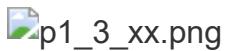
Your Answer

=> SQL code

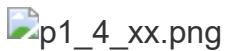
參考上圖，老師定義 SQL 的欄位名稱及資料型別。

將指定的 menu 資訊，寫成 SQL 指令，執行時可以新增資料。

SQL 指令要包含 CREATE TABLE, INSERT 指令



=> pgAdmin 顯示 database final_js_xx, table menu_xx 的四筆資料



(15%) P2: 實作路由 /menu_xx

menu_xx theme 已經放在 /public/menu 目錄下

靜態頁面已經可以透過 /menu_xx/static 顯現，有六筆

顯示頁面 womens_static.ejs 放在 views/shop_xx 下

如果你是實作 hats，靜態頁面請用 hats_static.ejs

=> Chrome 顯示/menu_xx/static 靜態頁面

The screenshot shows a web browser window with the URL `localhost:5000/menu_xx/static` in the address bar. The page title is **Our Menu - Static** located in Htchung, 123456789. Below the title is a navigation bar with five buttons: All, Breakfast, Lunch, Shakes, and Dinner. The main content area displays six menu items in a grid:

- Buttermilk Pancakes \$15.99**
I'm baby woke mlkshk wolf bitters live-edge blue bottle, hammock freegan copper mug whatever cold-pressed
- Diner Double \$13.99**
vaporware iPhone mumblecore selvage raw denim slow-carb leggings gochujang helvetica man braid jianbing. Marfa thundercats
- Godzilla Milkshake \$6.99**
ombucha chillwave fanny pack 3 wolf moon street art photo booth before they sold out organic viral.
- Country Delight \$20.99**
Shabby chic keffiyeh neutra snackwave pork belly shoreditch. Prism austin mlkshk truffaut,
- Egg Attack! \$12.99**
darium vegan pabst bicycle rights kickstarter paleo cento craft beer chia hoodie iPhone everyday
- Oreo Dream \$18.99**
Portland chicharrones ethical edison bulb, paleo cento craft beer chia hoodie iPhone everyday

=> 將 menu_static.ejs 複製一份到 menu_xx.ejs，從資料庫 menu_xx 資料表中取得 4 筆資料顯示

Menu -- From PostgreSQL Server

Htchung, 123456789

All Breakfast Lunch Shakes Dinner

3 - Godzilla Milkshake \$6.99
Artisan pop-up truffaut pug bicycle rights, williamsburg marfa four loko keffiyeh tousled brooklyn.

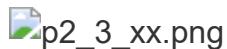
4 - Country Delight \$20.99
Keffiyeh helvetica normcore echo park mlkshk trust fund.

7 - Bacon Overflow \$8.99
Carry jianbing viral, poke farm-to-table banh mi before they sold out hexagon echo park.

9 - Quarantine Buddy \$16.99
Artisan pop-up truffaut pug bicycle rights, williamsburg marfa four loko keffiyeh tousled brooklyn.

Your Answer

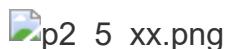
=> 將 menu_static.ejs 複製一份到 menu_xx.ejs，從資料庫 menu_xx 資料表中取得 4 筆資料顯示



=> server_xx.js 有關 code 之重點截圖



=> routes/MenuRouter_xx.js 之重點截圖

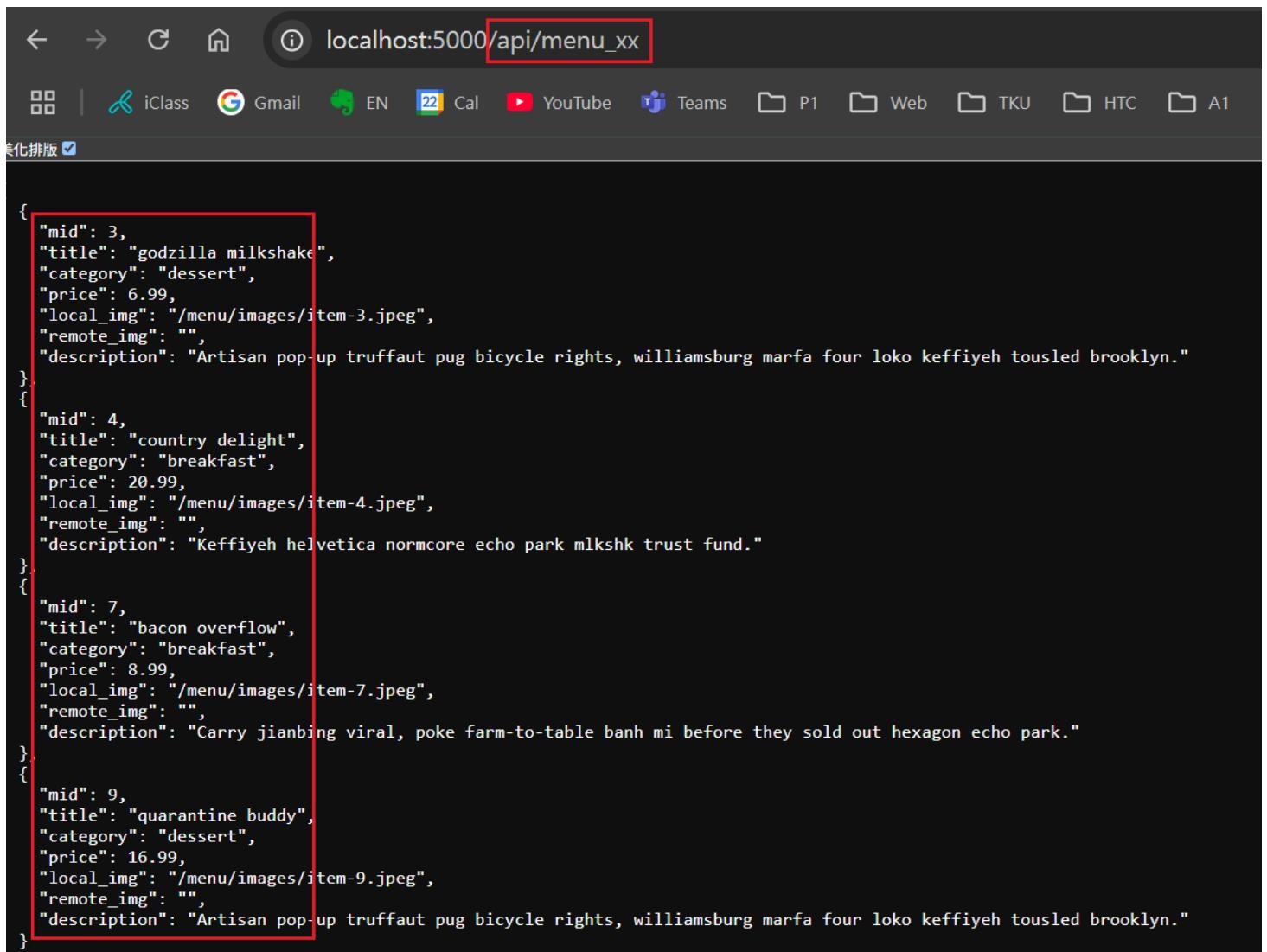


=> menu_xx.ejs 有關 code 之重點截圖



(15%) P3: 實作 路由 /api/menu_xx

=> Chrome 顯示



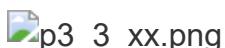
```
[{"mid": 3, "title": "godzilla milkshake", "category": "dessert", "price": 6.99, "local_img": "/menu/images/item-3.jpeg", "remote_img": "", "description": "Artisan pop-up truffaut pug bicycle rights, williamsburg marfa four loko keffiyeh tousled brooklyn."}, {"mid": 4, "title": "country delight", "category": "breakfast", "price": 20.99, "local_img": "/menu/images/item-4.jpeg", "remote_img": "", "description": "Keffiyeh helvetica normcore echo park mlkshk trust fund."}, {"mid": 7, "title": "bacon overflow", "category": "breakfast", "price": 8.99, "local_img": "/menu/images/item-7.jpeg", "remote_img": "", "description": "Carry jianbing viral, poke farm-to-table banh mi before they sold out hexagon echo park."}, {"mid": 9, "title": "quarantine buddy", "category": "dessert", "price": 16.99, "local_img": "/menu/images/item-9.jpeg", "remote_img": "", "description": "Artisan pop-up truffaut pug bicycle rights, williamsburg marfa four loko keffiyeh tousled brooklyn."}]
```

Your Answer

=> Chrome 顯示



=> server_xx.js code 重點



=> apiMenuRouter_xx.js code 重點



(10%) P4: 請以老師提供的框架，寫前端 JavaScript 程式

請以老師提供的 final_xx.html，實作 menu_api_nodejs_xx.js，透過 api 呼叫後端 Node.js server 端程式，可以取得 json 資料，套用在 final_xx.html 頁面中

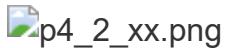
在 menu_api_nodejs_xx.js 中儲存 menu 的變數名稱為 menu_xx，不可以改變

=> Chrome 顯示

The screenshot shows a browser window with the URL 127.0.0.1:5500/menu/final_xx.html. The page title is "Menus -- API From Node.Js" and it displays a menu with items like "3-Godzilla Milkshake", "4-Country Delight", "7-Bacon Overflow", and "9-Quarantine Buddy". Below each item is a small image and a detailed description. At the top of the page, there is a placeholder for "Htchung, 123456789". Below the menu items are buttons for "All", "Breakfast", "Lunch", "Shakes", and "Dessert". The developer tools' Console tab is open, showing the JSON response from the API: "menu_xx.json from Node.js server" containing four menu items with their details (name, price, description). The developer tools also show the file path "menu_api_nodejs_xx.js:1".

Your Answer

=> Chrome 顯示



=> 相關 js code 顯示

相關 js code 顯示

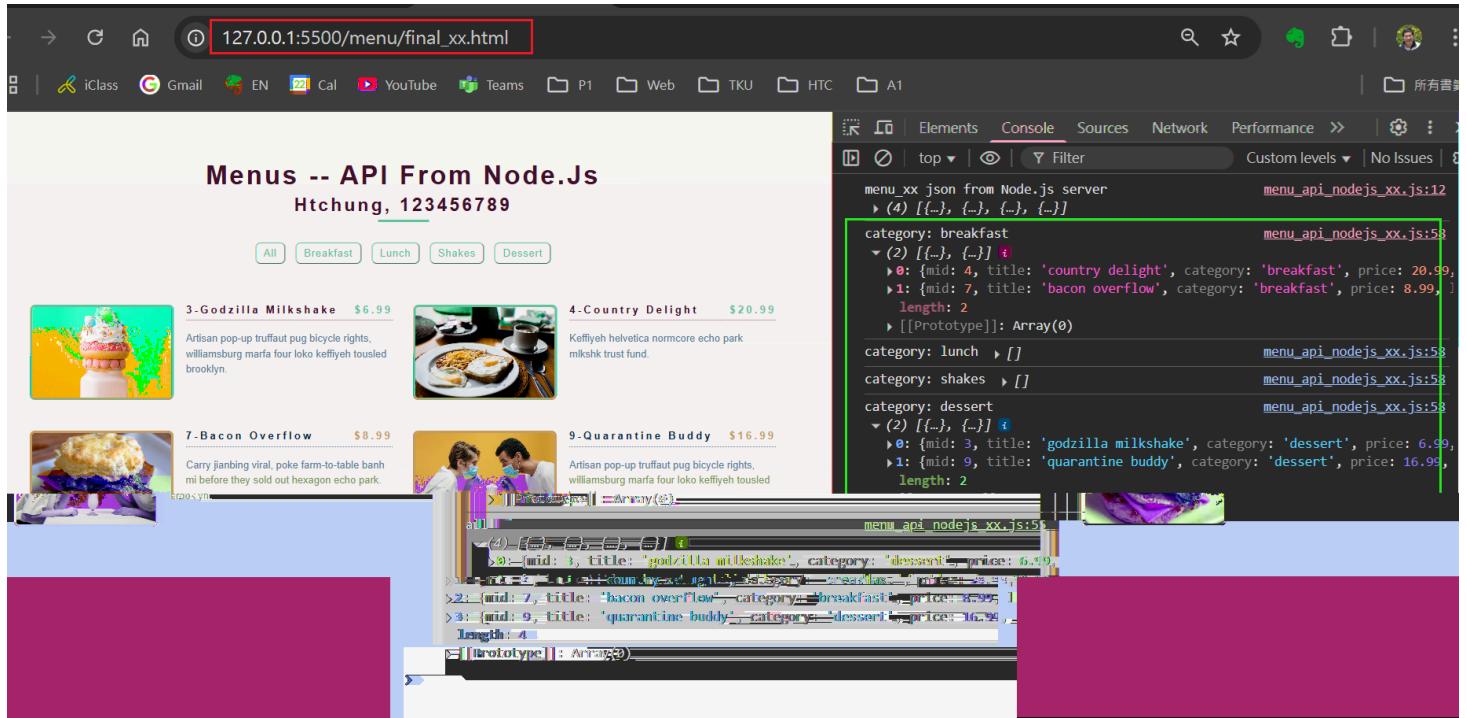
(10%) P5: 請針對 P4 題目，加入 filter 篩選功能

請參考在 server 端的 menu theme，該範例是可以執行 menu 之篩選功能的。其 filter 按鈕會根據所讀入的 menu 之 category 資訊自動建立，但本題是固定的，因此要做調整。

在 menu_api_nodejs_xx.js 中儲存 menu 的變數名稱為 menu_xx，不可以改變

=> Chrome 顯示

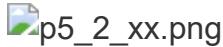
按照順序，點選按鈕 Breakfast, Lunch, Shakes, Dessert，然後用 console.log 顯示筆數，出現的筆數請展開



Your Answer

=> Chrome 顯示

按照順序，點選按鈕 Breakfast, Lunch, Shakes, Dessert，然後用 console.log 顯示筆數，出現的筆數請展開



=> 相關 js code 顯示

相關js code 顯示

(10%) P6: 在前端顯示 product, shop 資訊

請直接以老師給的 code 匯入資料庫，不需要再改學號後兩碼，從資料庫取得 json 資料，顯示在前端，唯一要改的姓名，學號

=> 資料庫顯示 shop_xx 資料表資料

The screenshot shows the pgAdmin interface. On the left, the database structure is visible, including tables like Domains, FTS Configurations, and shop_xx. The shop_xx table is selected, showing its columns: pid [PK] integer, title character varying (255), category_id integer, price real, local_img text, and remote_img text. A red box highlights the first seven rows of data. Below the table, the column definitions are listed.

| pid [PK] integer | title character varying (255) | category_id integer | price real | local_img text | remote_img text |
|------------------|-------------------------------|---------------------|------------|--|----------------------------|
| 1 | Blue Tanktop | 4 | 25 | /shop/img/womens/blue-tank.png | https://i.ibb.co/7CQVJNm |
| 2 | Floral Blouse | 4 | 20 | /shop/img/womens/floral-blouse.png | https://i.ibb.co/4W2DGKn |
| 3 | Floral Dress | 4 | 80 | /shop/img/womens/floral-skirt.png | https://i.ibb.co/KV18Ysr/f |
| 4 | Red Dots Dress | 4 | 80 | /shop/img/womens/red-polka-dot-dress.png | https://i.ibb.co/N3BN1bh |
| 5 | Striped Sweater | 4 | 45 | /shop/img/womens/striped-sweater.png | https://i.ibb.co/KmSkMbH |
| 6 | Yellow Track Suite | 4 | 135 | /shop/img/womens/yellow-track-suit.png | https://i.ibb.co/v1cwwNf/v |
| 7 | White Blouse | 4 | 20 | /shop/img/womens/white-vest.png | https://i.ibb.co/qBcrsJg/w |

=> Chrome 顯示 shop 資訊

The screenshot shows a web browser displaying a shopping page at 127.0.0.1:5500/shop/quiz2_p5_xx.html. The page lists women's clothing items with their details and images. The developer tools' Console tab is open, showing the JSON response from the Node.js server. A red box highlights the JSON data, which matches the data shown in the pgAdmin screenshot.

```

htchung, 123456789
Womens -- API From Node.js
Shop Contact Sign In
htchung, 123456789
Blue Tanktop 25 Floral Blouse 20 Floral Dress 80 Red Dots Dress 80
Stripped Sweater 45 Yellow Track Suite 135 White Blouse 20

```

```

Live reload enabled.
shop_xx json from Node.js server
Array(7)
0: {pid: 1, title: 'Blue Tanktop', category_id: 4, price: 25, local_img: '/shop/img/womens/blue-tank.png', remote_img: 'https://i.ibb.co/7CQVJNm'}
1: {pid: 2, title: 'Floral Blouse', category_id: 4, price: 20, local_img: '/shop/img/womens/floral-blouse.png', remote_img: 'https://i.ibb.co/4W2DGKn'}
2: {pid: 3, title: 'Floral Dress', category_id: 4, price: 80, local_img: '/shop/img/womens/floral-skirt.png', remote_img: 'https://i.ibb.co/KV18Ysr/f'}
3: {pid: 4, title: 'Red Dots Dress', category_id: 4, price: 80, local_img: '/shop/img/womens/red-polka-dot-dress.png', remote_img: 'https://i.ibb.co/N3BN1bh'}
4: {pid: 5, title: 'Striped Sweater', category_id: 4, price: 45, local_img: '/shop/img/womens/striped-sweater.png', remote_img: 'https://i.ibb.co/KmSkMbH'}
5: {pid: 6, title: 'Yellow Track Suite', category_id: 4, price: 135, local_img: '/shop/img/womens/yellow-track-suit.png', remote_img: 'https://i.ibb.co/v1cwwNf/v'}
6: {pid: 7, title: 'White Blouse', category_id: 4, price: 20, local_img: '/shop/img/womens/white-vest.png', remote_img: 'https://i.ibb.co/qBcrsJg/w'}
length: 7
[[Prototype]]: Array(0)

```

=> 資料庫顯示 product_xx 資料表資料

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer tree shows the database structure, including domains, FTS configurations, operators, and tables. A green box highlights the 'product_xx' table under the 'Tables (3)' section. On the right, the main window displays the contents of the 'product_xx' table in a grid format. The columns are: id [PK] integer, title character varying (255), price real, category character varying (255), img text, and remote_img text. The data consists of 12 rows of furniture items.

| | id [PK] integer | title character varying (255) | price real | category character varying (255) | img text | remote_img text |
|----|-----------------|-------------------------------|------------|----------------------------------|--------------------------------|---|
| 1 | 1 | Emperor Bed | 21.99 | Liddy | /product/images/product-1.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 2 | 2 | Accent Chair | 25.99 | Caressa | /product/images/product-2.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 3 | 3 | High-Back Bench | 9.99 | Ikea | /product/images/product-3.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 4 | 4 | Wooden Table | 19.99 | Ikea | /product/images/product-4.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 5 | 5 | Dining Table | 69.99 | Caressa | /product/images/product-5.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 6 | 6 | Entertainmint Center | 25.99 | Liddy | /product/images/product-6.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 7 | 7 | Albany Sectional | 10.99 | Ikea | /product/images/product-7.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 8 | 8 | Sofa Set | 69.99 | Liddy | /product/images/product-8.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 9 | 9 | Utopia Sofa | 69.99 | Liddy | /product/images/product-9.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 10 | 10 | Modern Bookshelf | 8.99 | Marcos | /product/images/product-10.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 11 | 11 | Albany Table | 79.99 | Marcos | /product/images/product-11.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |
| 12 | 12 | Leather Sofa | 39.95 | Liddy | /product/images/product-12.jpg | https://loxxwyirrkjrruqboq.supabase.co/ |

=> Chrome 顯示 product 資訊

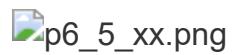
The screenshot shows a Chrome browser window displaying a product listing page at `127.0.0.1:5500/product/product_api_nodejs_xx.html`. The page shows a grid of 12 furniture items with images, titles, and prices. To the right, the developer tools' Elements tab is open, showing the JSON data received from the server. The JSON array contains 12 objects, each representing a product with fields: id, title, price, category, img, and remote_img.

```
product_xx json from Node.js server
product_api_nodejs_xx
product_xx
  ▾ Array(12)
    ▾ 0: {id: 1, title: "Emperor Bed", price: 21.99, category: "Liddy", img: "/product/images/product-1.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 1: {id: 2, title: "Accent Chair", price: 25.99, category: "Caressa", img: "/product/images/product-2.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 2: {id: 3, title: "High-Back Bench", price: 9.99, category: "Ikea", img: "/product/images/product-3.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 3: {id: 4, title: "Wooden Table", price: 19.99, category: "Ikea", img: "/product/images/product-4.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 4: {id: 5, title: "Dining Table", price: 69.99, category: "Caressa", img: "/product/images/product-5.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 5: {id: 6, title: "Entertainmint Center", price: 25.99, category: "Caressa", img: "/product/images/product-6.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 6: {id: 7, title: "Albany Sectional", price: 10.99, category: "Ikea", img: "/product/images/product-7.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 7: {id: 8, title: "Sofa Set", price: 69.99, category: "Liddy", img: "/product/images/product-8.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 8: {id: 9, title: "Utopia Sofa", price: 69.99, category: "Liddy", img: "/product/images/product-9.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 9: {id: 10, title: "Modern Bookshelf", price: 8.99, category: "Marcos", img: "/product/images/product-10.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 10: {id: 11, title: "Albany Table", price: 79.99, category: "Marcos", img: "/product/images/product-11.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ 11: {id: 12, title: "Leather Sofa", price: 39.95, category: "Liddy", img: "/product/images/product-12.jpg", remote_img: "https://loxxwyirrkjrruqboq.supabase.co/"}
    ▾ length: 12
  ▾ [Prototype]: Array(0)
```

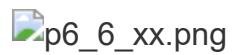
Your Answer

請直接以老師給的 code 匯入資料庫，不需要再改學號後兩碼，從資料庫取得 json 資料，顯示在前端，唯一要改的是姓名，學號

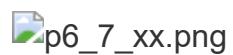
=> 資料庫顯示 shop_xx 資料表資料



=> Chrome 顯示 shop 資訊



=> 資料庫顯示 product_xx 資料表資料



=> Chrome 顯示 product 資訊

