

(12.5%) Web Programming 2N 期中上機考

1 -- 開放網路

2023-11-08, at E201

Note:

1. 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。
2. iClass 上請繳交 mid1_xx.md, mid1_xx.pdf，還有 client_mid1_xx.zip, server_mid1_xx.zip, w09_mid1_xx.zip (node_module 請移除，如有需要，移除 /public/img 下所有圖片)
3. 老師將題目提供在 w09_mid1_xx.md 檔上，要實作的部分放在 mid1_htc.pdf 上，請依照老師實作的圖片在你自己的 Github repo, Supabase, Vercel 上實作
4. 每一張圖片要有機房左側背景，要有你的學號，圖片標註要跟老師所標註的類似。違者會依情節扣分。
5. 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): (?, ?)

P1 (30%) -- 基本題

- P1-1 (6%): ? 分
- P1-2 (6%): ? 分
- P1-3 (6%): ? 分
- P1-4 (6%): ? 分
- P1-5 (6%): ? 分

P2(40%) -- Backend: Supabase, pgAdmin

- P2-1 (6%): ? 分
- P2-2 (6%): ? 分
- P2-3 (10%): ? 分
- P2-4 (6%): ? 分
- P2-5 (6%): ? 分
- P2-6 (6%): ? 分

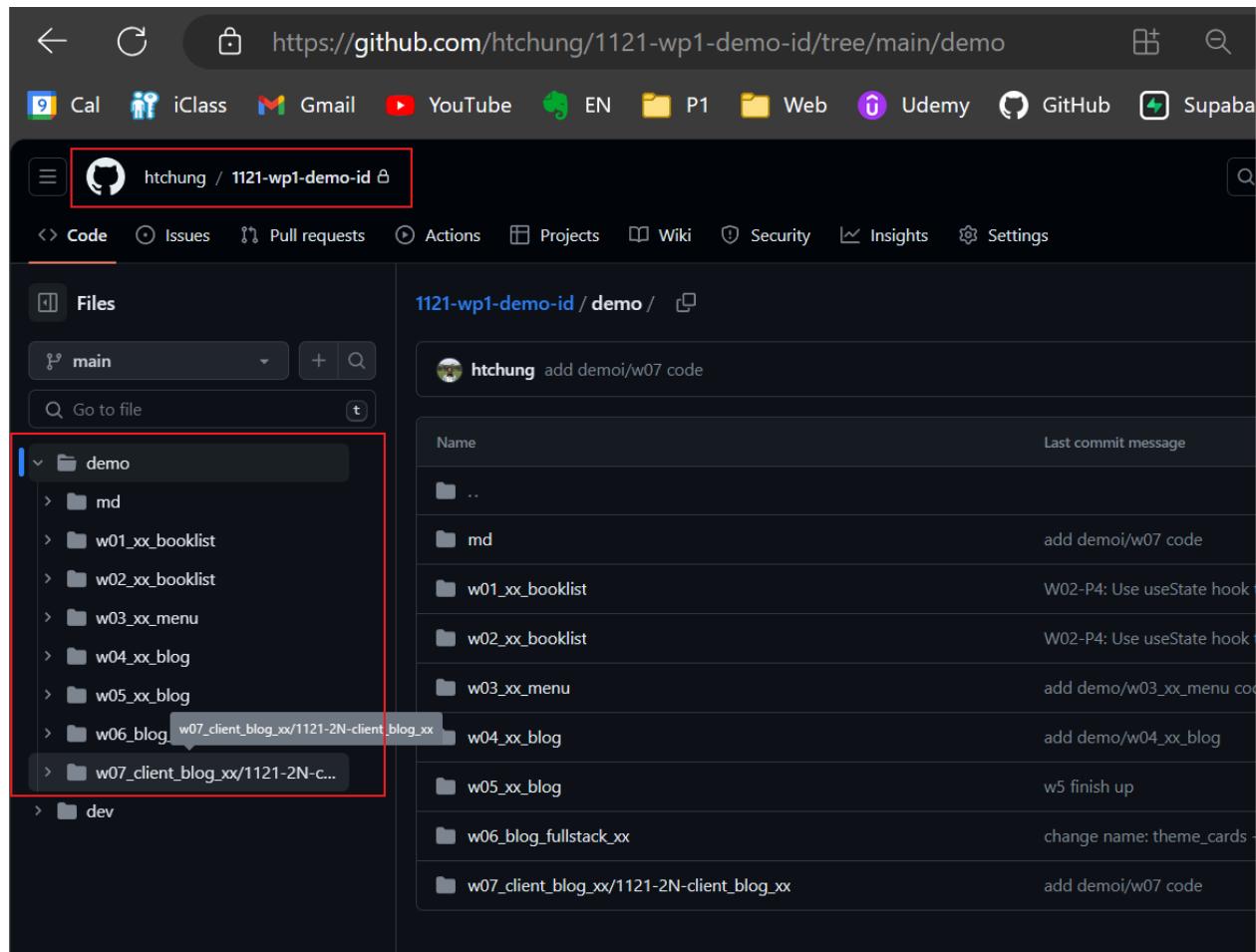
P3 (30%) -- Frontend: React

- P3-1 (10%): ? 分
- P3-2 (10%): ? 分
- P3-3 (10%): ? 分

(30%) P1: Show Info of Your Github Repo, Client Supa Github, Vercel as follows

(6%) P1-1: Show Your Github Repo page with all class demos

My Github Repo URL



The screenshot shows a GitHub repository page for 'htchung / 1121-wp1-demo-id'. The 'Code' tab is selected. On the left, there's a sidebar with 'Files' and a dropdown set to 'main'. A red box highlights the repository name 'htchung / 1121-wp1-demo-id' in the top navigation bar. Another red box highlights the 'demo' folder in the file tree on the left. The main area displays a list of files with their commit history:

Name	Last commit message
..	
md	add demo/w07 code
w01_xx_booklist	W02-P4: Use useState hook
w02_xx_booklist	W02-P4: Use useState hook
w03_xx_menu	add demo/w03_xx_menu co
w04_xx_blog	add demo/w04_xx_blog
w05_xx_blog	
w06_blog	w5 finish up
w07_client_blog_xx/1121-2N-client/blog_xx	change name: theme_cards -
w07_client_blog_xx/1121-2N-client/blog_xx	add demo/w07 code

(6%) P1-2: Show Your Github commits for week 7

The screenshot shows a GitHub repository page for `htchung / 1121-wp1-demo-id`. The main heading is "Commits". A dropdown menu shows "main" is selected. Below it, a section titled "Commits on Oct 25, 2023" lists several commits:

- `add demo1/w07 code` by `htchung` committed 2 weeks ago
- `W07-P4: Implement BlogLocalJson_xx.js to fetch blogs from json data` by `htchung` committed 2 weeks ago
- `W07-P3: Deploy to vercel, using /supa_xx to get blogs from Supabase` by `htchung` committed 2 weeks ago
- `W07-P2: Fetch blog data from Supabase` by `htchung` committed 2 weeks ago
- `### W07-P1: jwt introduction using anon key in Supabase` by `htchung` committed 2 weeks ago
- `add dev/w07 files` by `htchung` committed 2 weeks ago
- `first commit` by `htchung` committed 2 weeks ago

A red box highlights the first five commits listed above.

(6%) P1-3: Show Your Github URL, deploy to Vercel for Supabase blog demo

My Github URL for Supabase blog demo

The screenshot shows a GitHub repository page for `htchung / 1121-2N-client-blog_xx`. The main heading is "1121-2N-client-blog_xx". The "About" section on the right shows the URL `1121-2-n-client-blog-xx.vercel.app` highlighted with a red box. The repository structure on the left includes a folder named "public" highlighted with a red box, and files like `src`, `.gitignore`, `README.md`, `package-lock.json`, and `package.json`.

(6%) P1-4: Show Your Vercel page for Supabase blog demo, with route /supa_xx

My Vercel URL for Supabase blog demo

The screenshot shows a browser window with the URL https://1121-2-n-client-blog-xx.vercel.app/supa_xx. The page content is a blog with two posts: "LIFESTYLE" and "TRAVEL". The "LIFESTYLE" post is titled "Seven Reasons Why Coffee Is Awesome" and has a placeholder text. The "TRAVEL" post is titled "Travel To Paris" and also has a placeholder text. Below each post is a "Read More" link. On the right side of the browser, there is a DevTools sidebar showing the state structure of the application. The state tree includes nodes like "yr", "Pe", "xe", "Context.Provider", "Context.Provider", "e", "me", and "vr". The "vr" node is expanded, showing an array of 8 items under "State". Each item has a category ("lifestyle" or "travel") and a placeholder description.

```
vr
  State: [{}]
    0: {category: "lifestyle", descrip: "Lorem ipsum d..."}
    1: {category: "travel", descrip: "Lorem ipsum dolo..."}
    2: {category: "lifestyle", descrip: "Lorem ipsum d..."}
    3: {category: "lifestyle", descrip: "Lorem ipsum d..."}
    4: {category: "lifestyle", descrip: "Lorem ipsum d..."}
    5: {category: "travel", descrip: "Lorem ipsum dolo..."}
    6: {category: "travel", descrip: "Lorem ipsum dolo..."}
    7: {category: "travel", descrip: "Lorem ipsum dolo..."}
    8: {category: "travel", descrip: "Lorem ipsum dolo..."}
```

(6%) P1-5: Git push P1 with git log

The screenshot shows a GitHub repository page for "htchung / 1121-wp1-demo-id". The "Code" tab is selected. The main area shows a commit log with one entry: "P1-5: Git push P1 with git log" by "htchung" committed 3 minutes ago. This commit is highlighted with a red box.

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2023-11-07"
```

d68c61e htchung Wed Nov 8 04:31:54 2023 +0800 P1-5: Git push P1 with git log

(40%) P2: Backend: Node using Supabase & pgAdmin

(6%) P2-1: Show connection info in Supabase, and use pgAdmin to connect to Supabase

=> show connection info from Supabase

The screenshot shows the Supabase dashboard under the 'Settings' tab. A red box highlights the 'Database Settings' section. Inside this section, the 'Connection info' sub-section is displayed with the following details:

Setting	Value	Action
Host	db.sldvcsgcmhhgtrjjccxd.supabase.co	<input type="button" value="Copy"/>
Database name	postgres	<input type="button" value="Copy"/>
Port	5432	<input type="button" value="Copy"/>
User	postgres	<input type="button" value="Copy"/>
Password	[The password you provided when you created this project]	

At the bottom of the 'Database Settings' section, there is a note: "We only collect analytics essential to ensuring smooth operation of our service". Below this note are three buttons: "Accept", "Opt out", and "Learn more".

=> show how to connect to Supabase in pgAdmin

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** On the left, it shows the database structure under "tku-htctku".
 - Databases:** postgres (selected), containing Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, and Schemas (12).
 - Schemas:** auth, extensions, graphql, graphql_public, pgbouncer, pgodium, pgodium_masks, pgtle, public, Aggregates.
- Dashboard:** Shows a query result table with two rows:

4	4	Coffee Make You Feel Good	Lore ipsum dolor sit amet consectetur adipisicing el...
5	5	Coffee Make You Calm	Lore ipsum dolor sit amet consectetur adipisicing el...
- Properties:** Shows the connection settings for the selected "postgres" database.
 - Connection Tab:** Host name/address: db.sldvcsgcmhhgtrjjccxd.supabase.co, Port: 5432, Maintenance database: postgres, Username: postgres.
 - General Tab:** Kerberos authentication? (disabled).
 - Buttons:** Close, Reset, Save.

(6%) P2-2: 在 pgAdmin 中下 SQL 指令建立 product_xx 資料

=> 在 local pgAdmin 檢視

The screenshot shows the pgAdmin interface. On the left, the Object Explorer pane displays a tree structure of database objects. A red box highlights the 'product_xx' table under the 'Tables (4)' section. Below it, the 'Columns (6)' section is also highlighted with a red box, showing columns: pid, name, category_id, price, local_img_url, and remote_img_url. On the right, the main pane shows the SQL query window with the following code:

```
1 CREATE TABLE product_xx (
2     pId int NOT NULL PRIMARY KEY,
3     name varchar(255),
4     category_id int,
5     price real,
6     local_img_url varchar(255),
7     remote_img_url varchar(255)
8 );
```

The 'Messages' tab in the bottom right shows the result of the query: 'CREATE TABLE'. It also notes that the query was successful and returned in 251 msec.

=> 在 Supabase 檢視

The screenshot shows the Supabase Table Editor. At the top, the URL is https://supabase.com/dashboard/project/sldvcs... and the schema is set to 'public'. The table 'tku_xx' is selected. A red box highlights the 'product_xx' table in the list of tables. The table structure is shown in the main editor area, with columns: pid (int4), name (varchar), category_id (int4), and price (real). A warning message at the top states: 'WARNING: You are allowing anonymous access to your table.' with links to 'Enable Row Level Security' and 'Dismiss'. Below the table structure, it says 'This table is empty' and 'Add rows to your table to get started.' A 'Import data via CSV' button is at the bottom.

(10%) P2-3: 在 Supabase 下用 SQL 指令建立 product_xx 資料

請以學號最後一碼除以 5 取餘數取得一個 category，對應到 category id 如下：

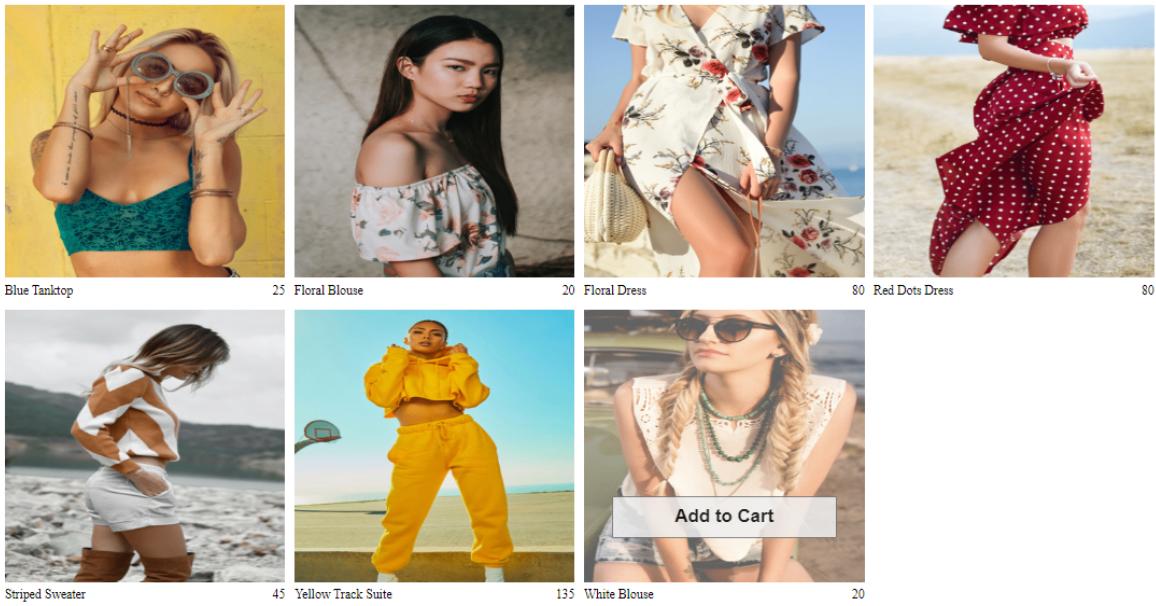
Category: 1(hats), 2(jackets), 3(sneakers), 4(womens), 5(mens)

請實作對應 html theme 中該類的產品。舉例，如果是 4(womens)，要實作 womens.html，如果是 0，要實作 mens.html。

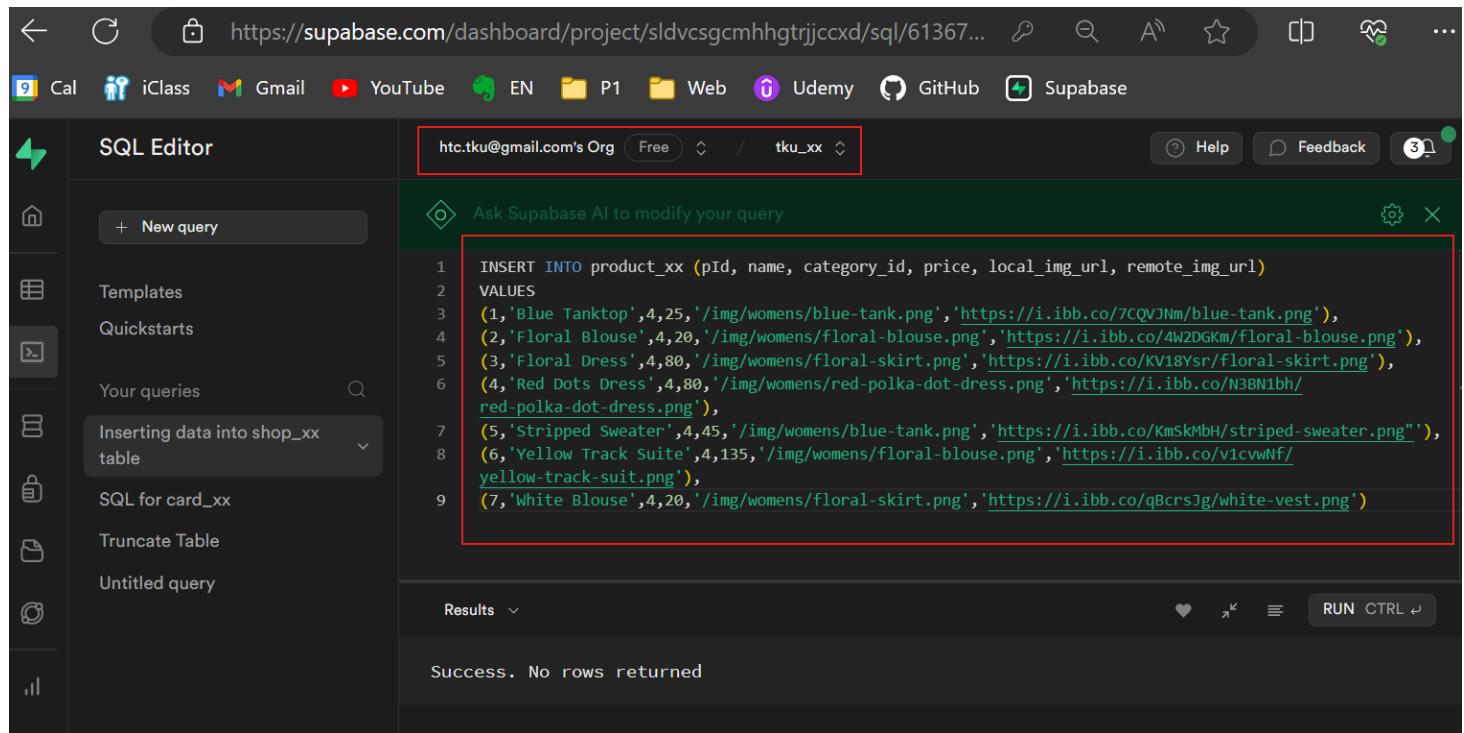
以下以 womens.html 來做說明。

=> womens.html

Womens



=> 在 Supabase 下用 SQL Editor 下 SQL 指令

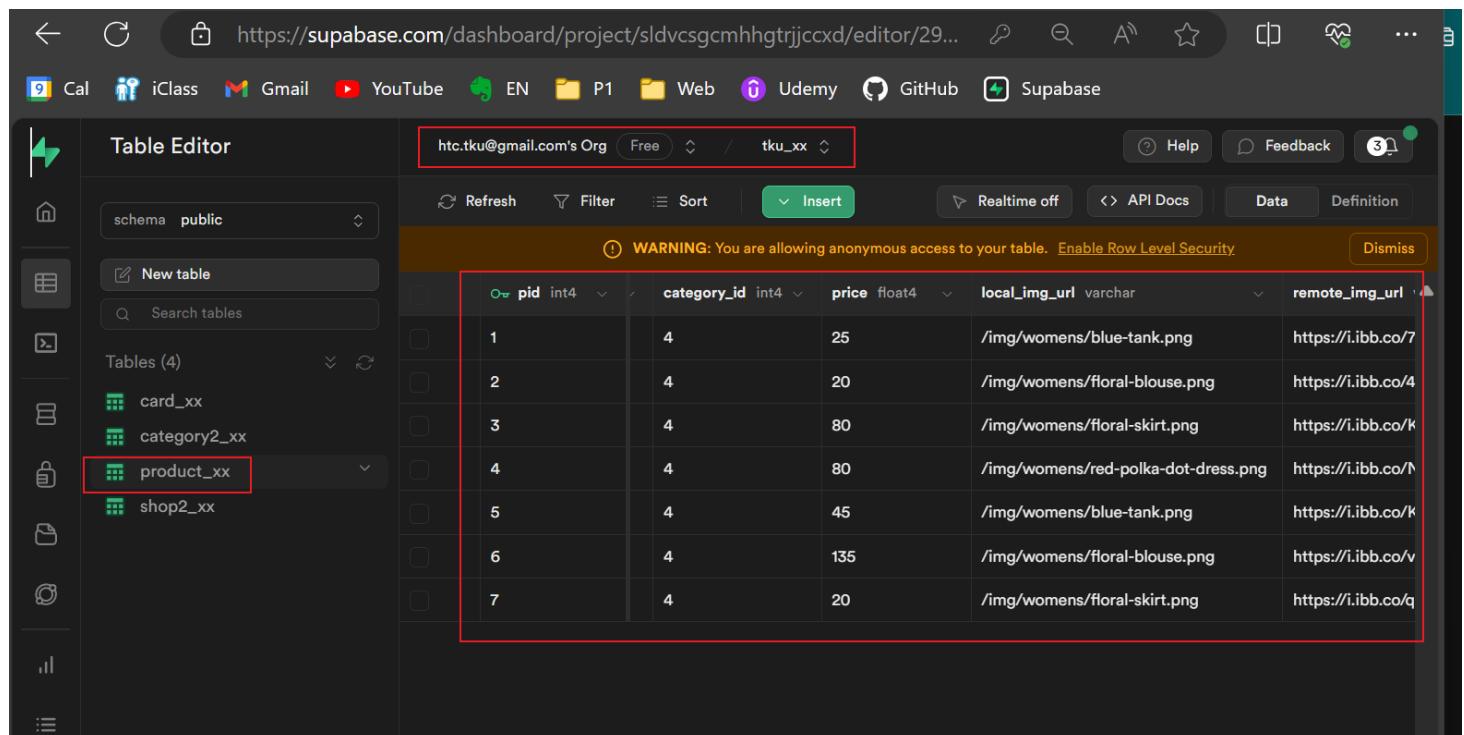


The screenshot shows the Supabase SQL Editor interface. On the left sidebar, there are various navigation options like Templates, Quickstarts, Your queries, Inserting data into shop_xx table, SQL for card_xx, Truncate Table, and Untitled query. The main area displays an SQL query:

```
1 INSERT INTO product_xx (pId, name, category_id, price, local_img_url, remote_img_url)
2 VALUES
3 (1,'Blue Tanktop',4,25,'/img/womens/blue-tank.png','https://i.ibb.co/7CQVJNm/blue-tank.png'),
4 (2,'Floral Blouse',4,20,'/img/womens/floral-blouse.png','https://i.ibb.co/4W2DGKm/floral-blouse.png'),
5 (3,'Floral Dress',4,80,'/img/womens/floral-skirt.png','https://i.ibb.co/KV18Ysr/floral-skirt.png'),
6 (4,'Red Dots Dress',4,80,'/img/womens/red-polka-dot-dress.png','https://i.ibb.co/N3BN1bh/
red-polka-dot-dress.png'),
7 (5,'Striped Sweater',4,45,'/img/womens/blue-tank.png','https://i.ibb.co/KmSKMbH/striped-sweater.png'),
8 (6,'Yellow Track Suite',4,135,'/img/womens/floral-blouse.png','https://i.ibb.co/v1cvwNf/
yellow-track-suit.png'),
9 (7,'White Blouse',4,20,'/img/womens/floral-skirt.png','https://i.ibb.co/qBcrsJg/white-vest.png')
```

The results section below shows "Success. No rows returned".

=> 要能顯示 pid, category_id, price, local_img_url, remote_img_url



The screenshot shows the Supabase Table Editor interface. On the left sidebar, there are options like schema public, New table, and a list of tables including card_xx, category2_xx, product_xx, and shop2_xx. The product_xx table is selected. The main area shows the table structure and data:

	pid	category_id	price	local_img_url	remote_img_url
1	4	25	/img/womens/blue-tank.png	https://i.ibb.co/7CQVJNm/blue-tank.png	
2	4	20	/img/womens/floral-blouse.png	https://i.ibb.co/4W2DGKm/floral-blouse.png	
3	4	80	/img/womens/floral-skirt.png	https://i.ibb.co/KV18Ysr/floral-skirt.png	
4	4	80	/img/womens/red-polka-dot-dress.png	https://i.ibb.co/N3BN1bh/red-polka-dot-dress.png	
5	4	45	/img/womens/blue-tank.png	https://i.ibb.co/KmSKMbH/striped-sweater.png	
6	4	135	/img/womens/floral-blouse.png	https://i.ibb.co/v1cvwNf/yellow-track-suit.png	
7	4	20	/img/womens/floral-skirt.png	https://i.ibb.co/qBcrsJg/white-vest.png	

A warning message at the top says: "WARNING: You are allowing anonymous access to your table. Enable Row Level Security".

(6%) P2-4: 透過 Supabase · 取得 Project URL 和 anon key

The screenshot shows the Supabase dashboard under the 'API' settings. The 'Project URL' is displayed as <https://sldvcsgcmhhgtrjjccxd.supabase.co>. Below it, the 'Project API keys' section shows an 'anon' key: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InNsZHJc2dj...1NzcyNH0.uk-`. A red box highlights the 'anon' key area.

(6%) P2-5: 解析 anon key · 用 JWT secret 來確認 anon key 是正確的

The screenshot shows the jwt.io debugger interface. An encoded JWT token is pasted into the 'Encoded' field: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InNsZHJc2dj...1NzcyNH0.uk-`. The 'Decoded' section shows the token structure:

```
HEADER: ALGORITHM & TOKEN TYPE
{
  "alg": "HS256",
  "typ": "JWT"
}

PAYLOAD: DATA
{
  "iss": "supabase",
  "ref": "sldvcsgcmhhgtrjjccxd",
  "role": "anon",
  "iat": 1683081724,
  "exp": 1698657724
}

VERIFY SIGNATURE
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-key
) □ secret base64 encoded
```

A red box highlights the 'Signature Verified' status at the bottom left. Another red box highlights the 'your-256-bit-key' placeholder in the verify signature section.

(6%) P2-6: 透過 Supabase · 取得如何呼叫讀取 product_xx 所有資料的方法

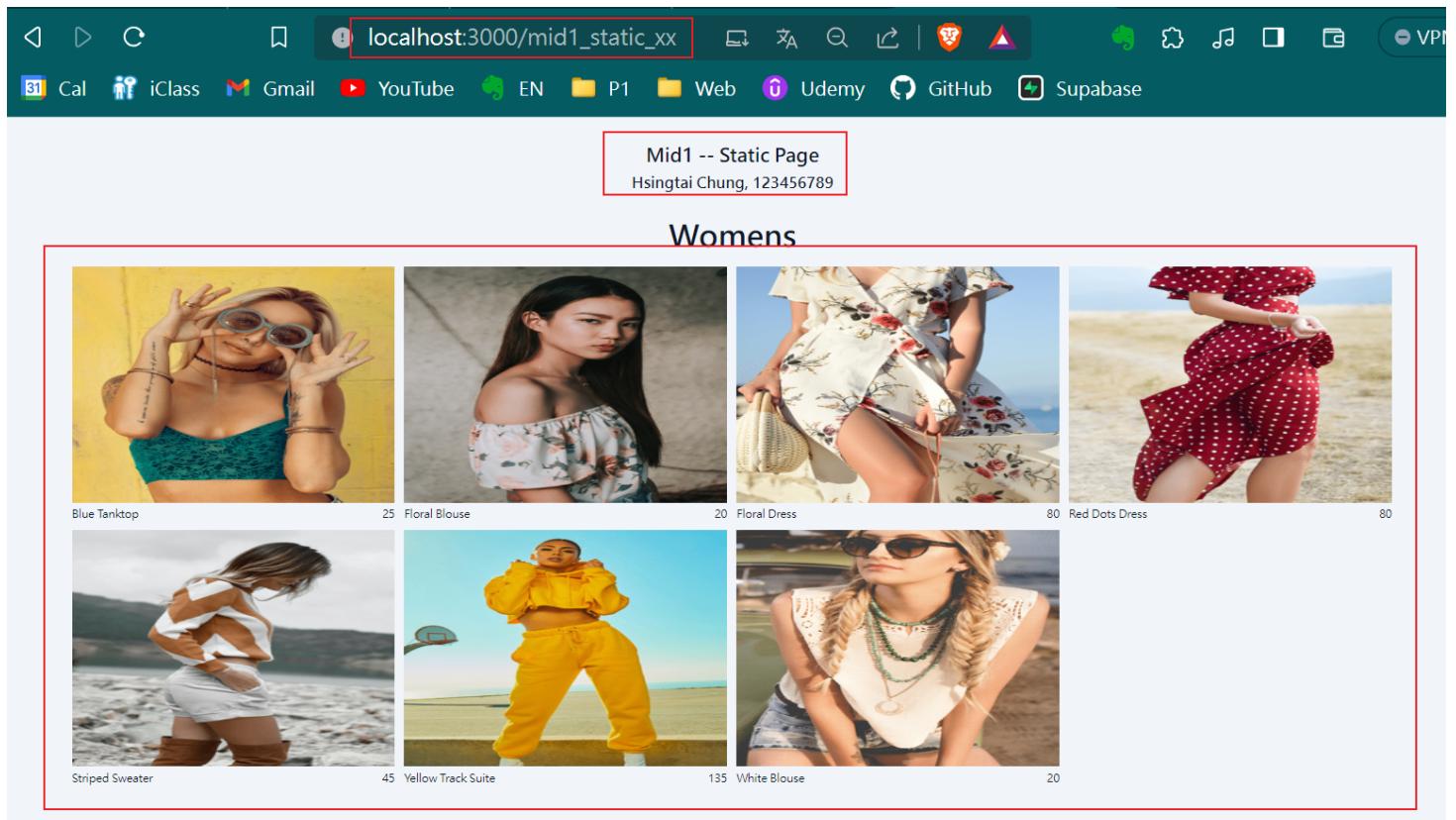
The screenshot shows the Supabase Table Editor interface. On the left, there's a sidebar with various icons and a list of tables: card_xx, category2_xx, product_xx (which is selected and highlighted with a red box), and shop2_xx. In the center, under 'Tables & Views', the 'product_xx' table is selected. A red box highlights the 'Read rows' button. To the right, the 'API Docs' section is displayed, also with a red box highlighting the 'Read rows' heading. The documentation provides examples for reading rows:

```
let { data: product_xx, error } = await supabase
  .from('product_xx')
  .select('*')
```

Below this, there are sections for 'Read specific columns' and 'Read foreign tables', each with its own code example.

(30%) P3: React Using Supabase

(10%) P3-1: 將 P2-3 靜態頁面轉成 React 頁面，放在 `pages/mid1_xx/Mid1StaticPage_xx.js`，透過路由 route `/mid1_static_xx` 可以顯示。



=> 顯示程式碼，可以看到有多少筆 product 資料 (某些 code 要內收)

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

- File Explorer:** Shows the project structure under "CLIENT_MID1_XX". Key folders include "node_modules", "public", "src", "data", and "pages". Inside "pages", there are "mid1_xx" and "midp_xx" folders. "mid1_xx" contains "Mid1StaticPa... 7" (highlighted with a red box). Other files in "pages" include "ShopLocalIso... 7", "ShopNodePage_x...", "ShopStaticPage_x...", "BlogLocalJson_xx.js", "BlogNodeServer_xx.js", "BlogStaticPage_xx.js", and "HomePage_xx.js".
- Search Bar:** Contains the text "client_mid1_xx".
- Code Editor:** Displays the content of "Mid1StaticPage_xx.js".

```
const Mid1StaticPage_xx = () => [
  <>
    <div className='shop-page'>
      <div className='section-title'>
        <h2> Mid1 -- Static Page </h2>
        <h3> Hsingtai Chung, 123456789</h3>
      </div>
      <div className='collection-page'>
        <h1 className='title'>Womens</h1>
        <div className='items'>
          <div className='collection-item'>..</div>
          <div className='collection-item'>..</div>
        </div>
      </div>
    </div>
  );
};

export default Mid1StaticPage_xx;
```

Two specific sections of the code are highlighted with red boxes:

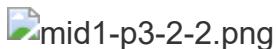
- The header section: `<div className='shop-page'>` to `<div className='collection-page'>`.
- The product listing section: `<div className='items'>` to `</div>`.

(10%) P3-2: 藉由 P2-6 Supabase 之存取方式，取得 product_xx 中所有資料，並透過路由 /mid1_supapage_xx 顯示在 pages/mid1_xx/Mid1SupaPage_xx.js 中。

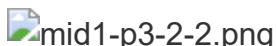
The screenshot shows a web browser window with the URL `localhost:3000/mid1_supapage_xx`. The main content area displays a grid of women's clothing items with images, names, and prices. A red box highlights the header "Mid1 -- Supa Page" and the user information "Hsingtai Chung, 123456789". The DevTools Components tab is open, showing the component tree for the application. A red box highlights the component `Mid1SupaPage_xx` under the `App_xx` tree. The state of this component is expanded, showing an array of objects representing products. The first few objects are:

```
1 State: "Hsingtai Chung"
2 State: 123456789
3 State: [{"category_id": 4, "local_img_url": "/img/womens/blue-t..."}, {"category_id": 4, "local_img_url": "/img/womens/floral..."}, {"category_id": 4, "local_img_url": "/img/womens/floral..."}, {"category_id": 4, "local_img_url": "/img/womens/red-po..."}, {"category_id": 4, "local_img_url": "/img/womens/blue-t..."}, {"category_id": 4, "local_img_url": "/img/womens/floral..."}]
```

=> code for fetching products data



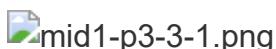
=> code for rendering products data



(10%) P3-3: 透過路由 /mid1_supapage_xx 到 Supabase 讀取 card_xx 中某個 category 的資料，以你的學號最後一碼對應到 card_xx 之 id，取得該筆資料的 category，如果是 0，則選 id = 1。顯示頁面是 pages/mid1_xx/Mid1Supa2Page_xx.js 。

Hint: 請自行查 Supabase API doc，要如何存取符合某些欄位的資料

=> Supabase API doc



=> Supabase 上該 category 資料

The screenshot shows the Supabase Table Editor interface. On the left, there's a sidebar with icons for Home, Tables, and a search bar. Below it, a list of tables: 'card_xx' (highlighted with a red box), 'category2_xx', 'product_xx', and 'shop2_xx'. The main area is titled 'Table Editor' and shows a table with columns: id (int4), title (varchar), descrip (varchar), and category (varchar). A warning message at the top says 'WARNING: You are allowing anonymous access to your table. Enable Row Level Security.' Below the table, there are buttons for Refresh, Filter, Sort, Insert, Realtime off, API Docs, Data, and Definition.

	id	title	descrip	category
1	1	Seven Reasons Why Coffee Is Awesome	Lorem ipsum dolor sit amet consectetur a	lifestyle
2	2	Travel To Paris	Lorem ipsum dolor sit amet consectetur a	travel
3	3	Coffee Brings Friendship	Lorem ipsum dolor sit amet consectetur a	lifestyle
4	4	Coffee Make You Feel Good	Lorem ipsum dolor sit amet consectetur a	lifestyle
5	5	Coffee Make You Calm	Lorem ipsum dolor sit amet consectetur a	lifestyle
6	6	101 Tower In Taipei	Lorem ipsum dolor sit amet consectetur a	travel
7	7	Sun Rise From The Mountain	Lorem ipsum dolor sit amet consectetur a	travel
8	8	Serene Lake With Trees	Lorem ipsum dolor sit amet consectetur a	travel
9	9	Rocks Of Queen Head In Yehliu Taiwan	Lorem ipsum dolor sit amet consectetur a	travel

=> 瀏覽器顯示 category 資料，要用 React DevTools Components 來顯示讀到的資料

The screenshot shows a browser window displaying a travel blog page. The URL is 'localhost:3000/mid1_supapage_xx'. The page content includes a header 'Mid1 -- Supabase Blog Page' and 'Category -- Travel', followed by four travel-related cards. Each card has a thumbnail, a category tag ('TRAVEL'), a title, a short description, and a 'Read More' link. To the right of the page, the React DevTools Components panel is open, showing the component tree and state. The state 'Mid1Supa2Page_xx' is expanded, showing the array of travel items. Each item has a key (0, 1, 2, 3, 4) and properties: 'category' (travel), 'descrip' (Lorem ipsum dolor si...), and 'State' (Hsingtai Chung, 123456789).

```
state: {  
  Mid1Supa2Page_xx: [Object, Object, Object, Object, Object]  
}  
  
Object {  
  category: "travel",  
  descrip: "Lorem ipsum dolor si...",  
  State: "Hsingtai Chung"  
},  
Object {  
  category: "travel",  
  descrip: "Lorem ipsum dolor si...",  
  State: "123456789"  
},  
Object {  
  category: "travel",  
  descrip: "Lorem ipsum dolor si...",  
  State: "[...]"  
},  
Object {  
  category: "travel",  
  descrip: "Lorem ipsum dolor si...",  
  State: "[...]"  
},  
Object {  
  category: "travel",  
  descrip: "Lorem ipsum dolor si...",  
  State: "[...]"  
}
```

=> code for fetching category data

