

# (10%) Web Programming 2N 期末上機考 1 - - 開放網路

2023-06-07, 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 w17\_xx.md, w17\_xx.pdf, 還有 1112-client-supabase-1234567.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 w17\_xx.md 檔案上，要實作的部分放在 w17\_xx.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。

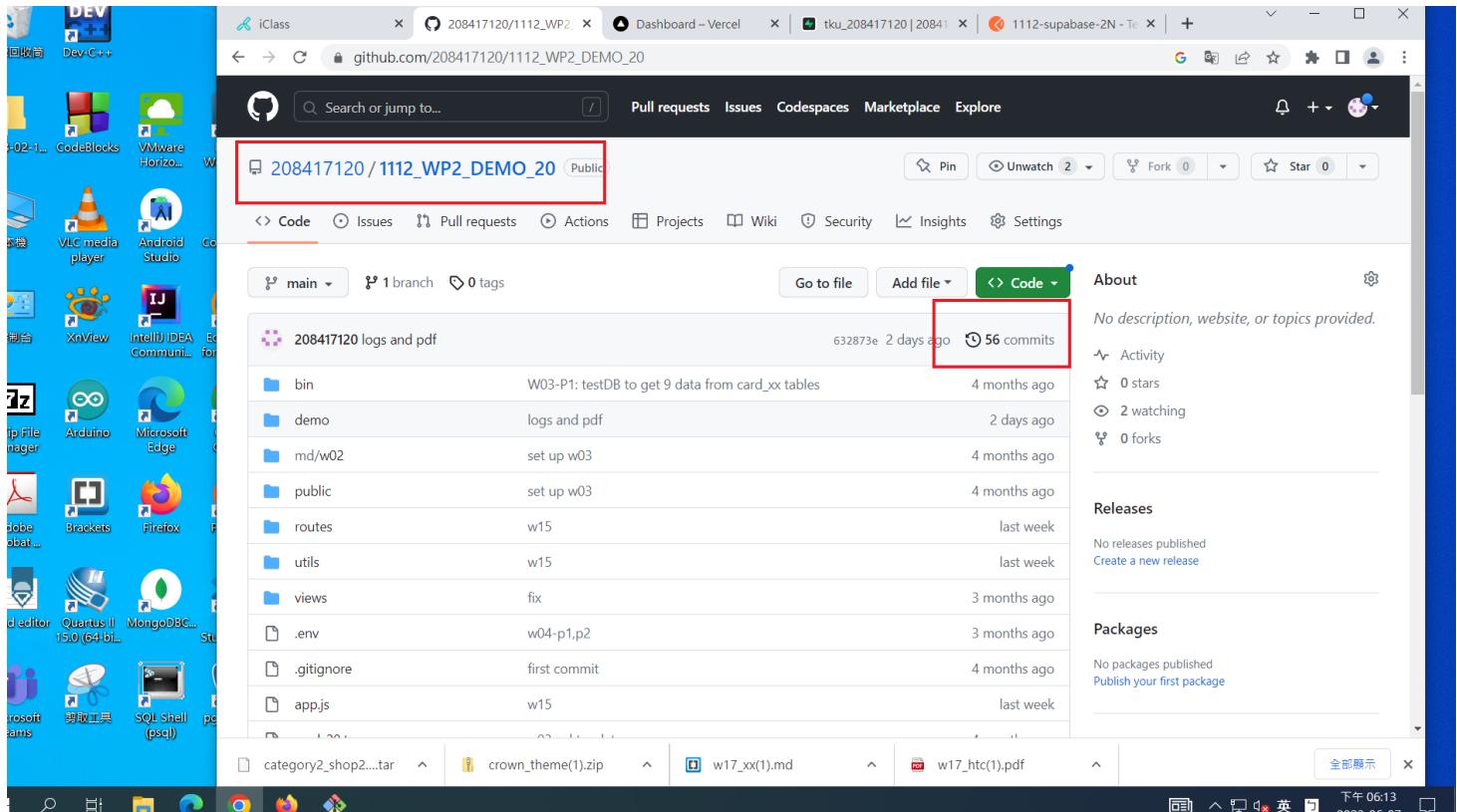
Your (Name, ID): (name, id)

- P1-1 (6%): 6 分
- P1-2 (6%): 6 分
- P1-3 (6%): 6 分
- P1-4 (6%): 6 分
- P1-5 (6%): 6 分
- P2-1 (10%): 10 分
- P2-2 (10%): 10 分
- P2-3 (10%): 10 分
- P2-4 (5%): 5 分
- P3-1 (10%): 10 分
- P3-2 (10%): 10 分
- P3-3 (10%): ? 分
- P3-4 (5%): 5 分
- 總分: 90 分

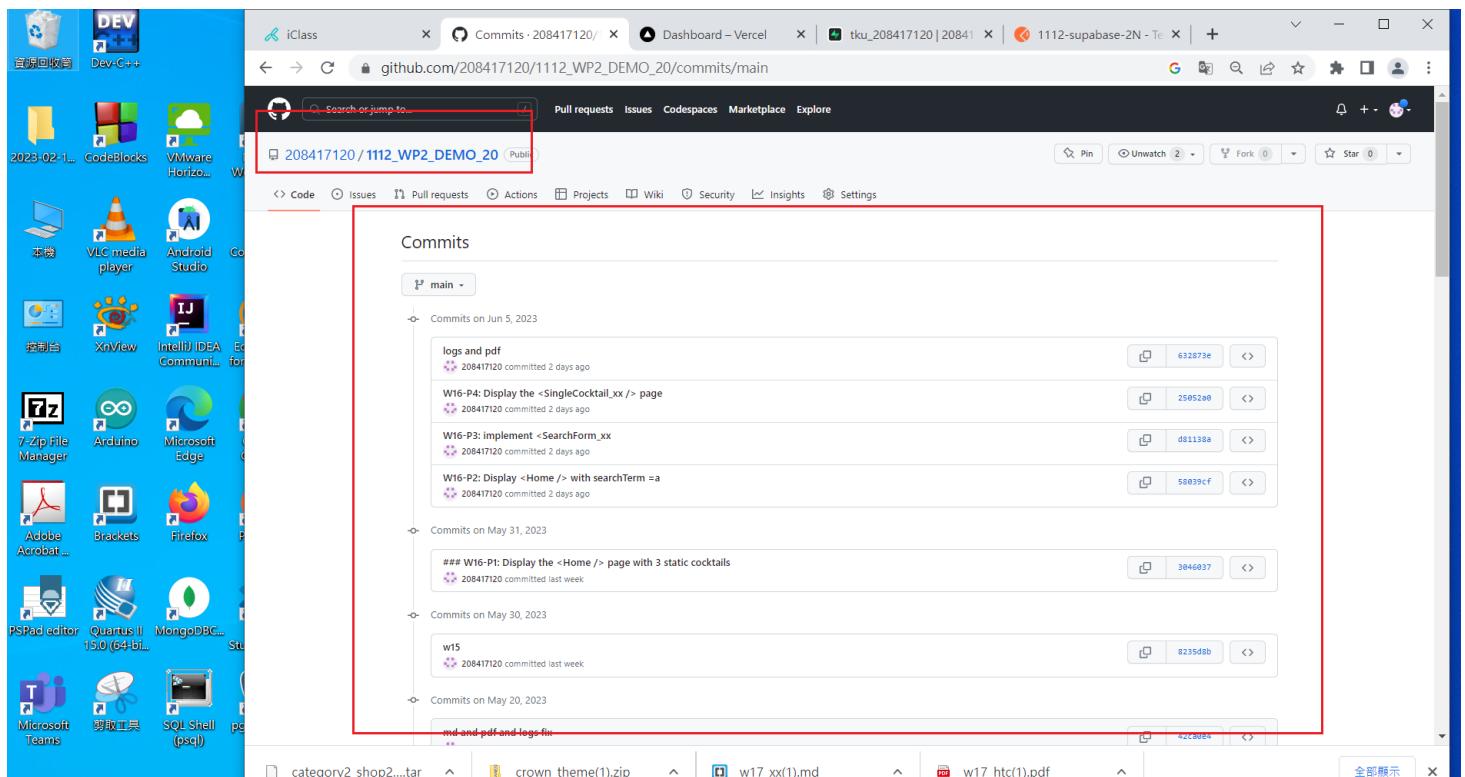
# (30%) P1: Show Info of Your Github Repo, Client Supa Github, Vercel as follows

## (6%) P1-1: Show Your Github Repo

My Github Repo URL

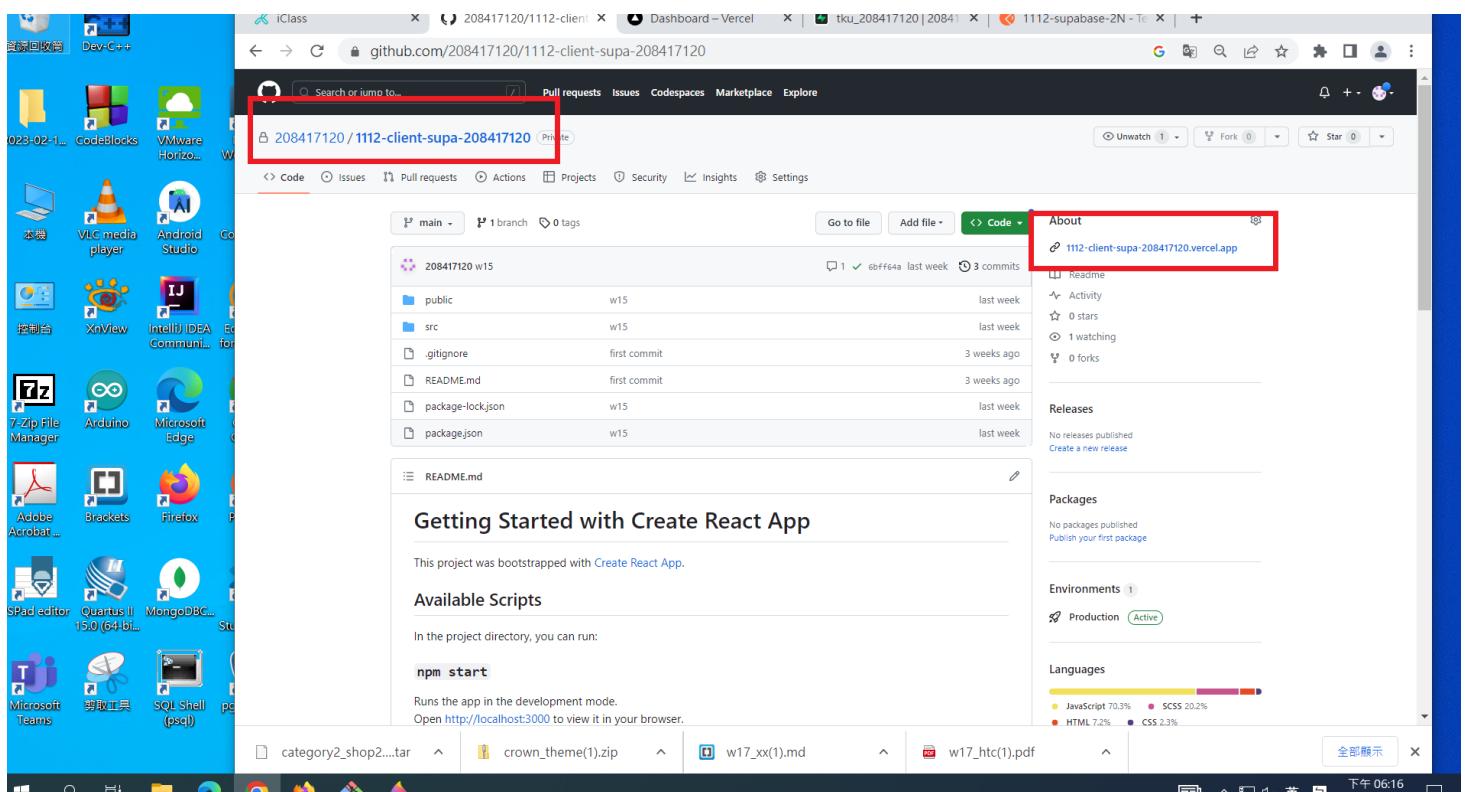


## (6%) P1-2: Show Your Github commits for week 15 and 16



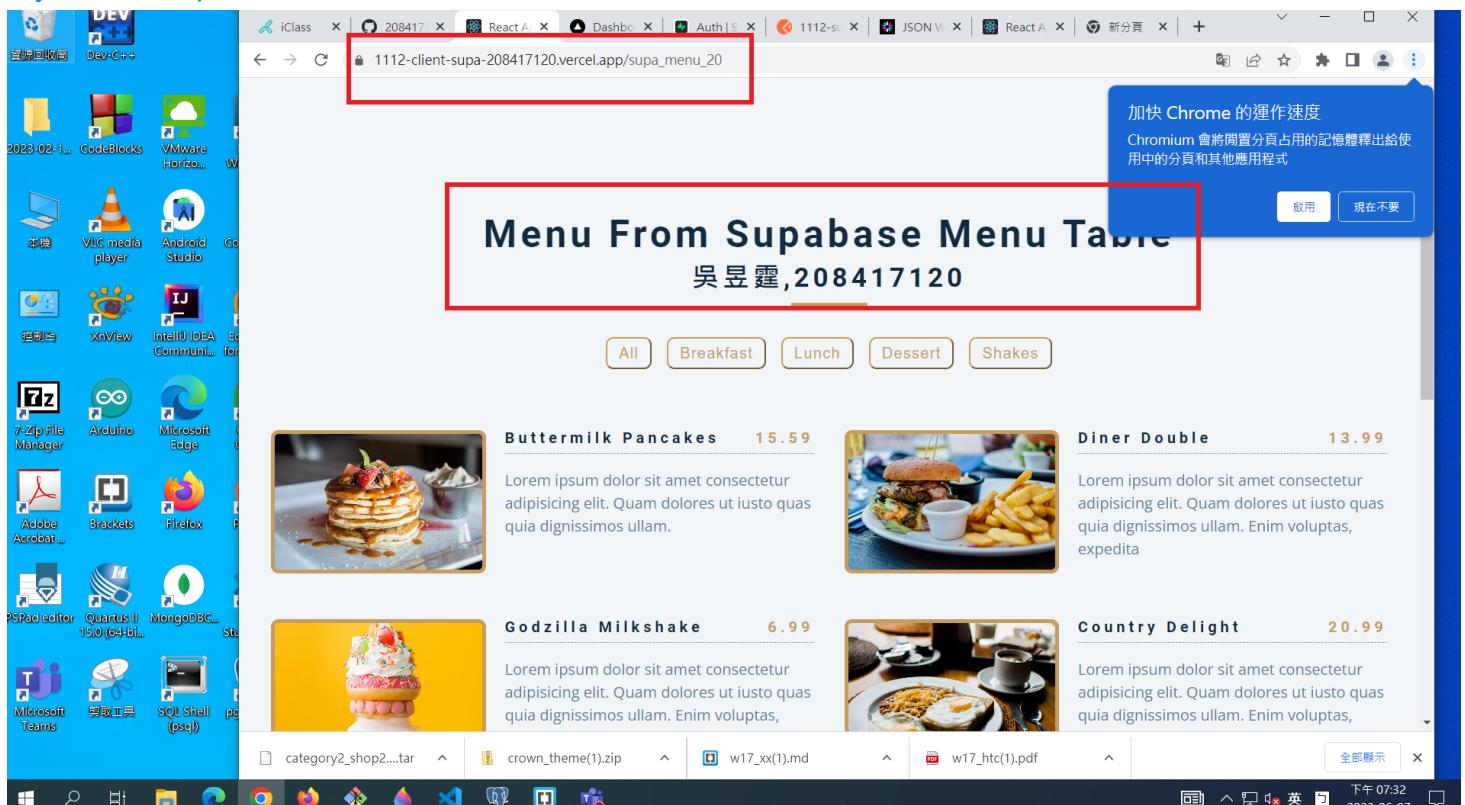
## (6%) P1-3: Show Your Github SupaDemo URL

My Github SupaDemo URL

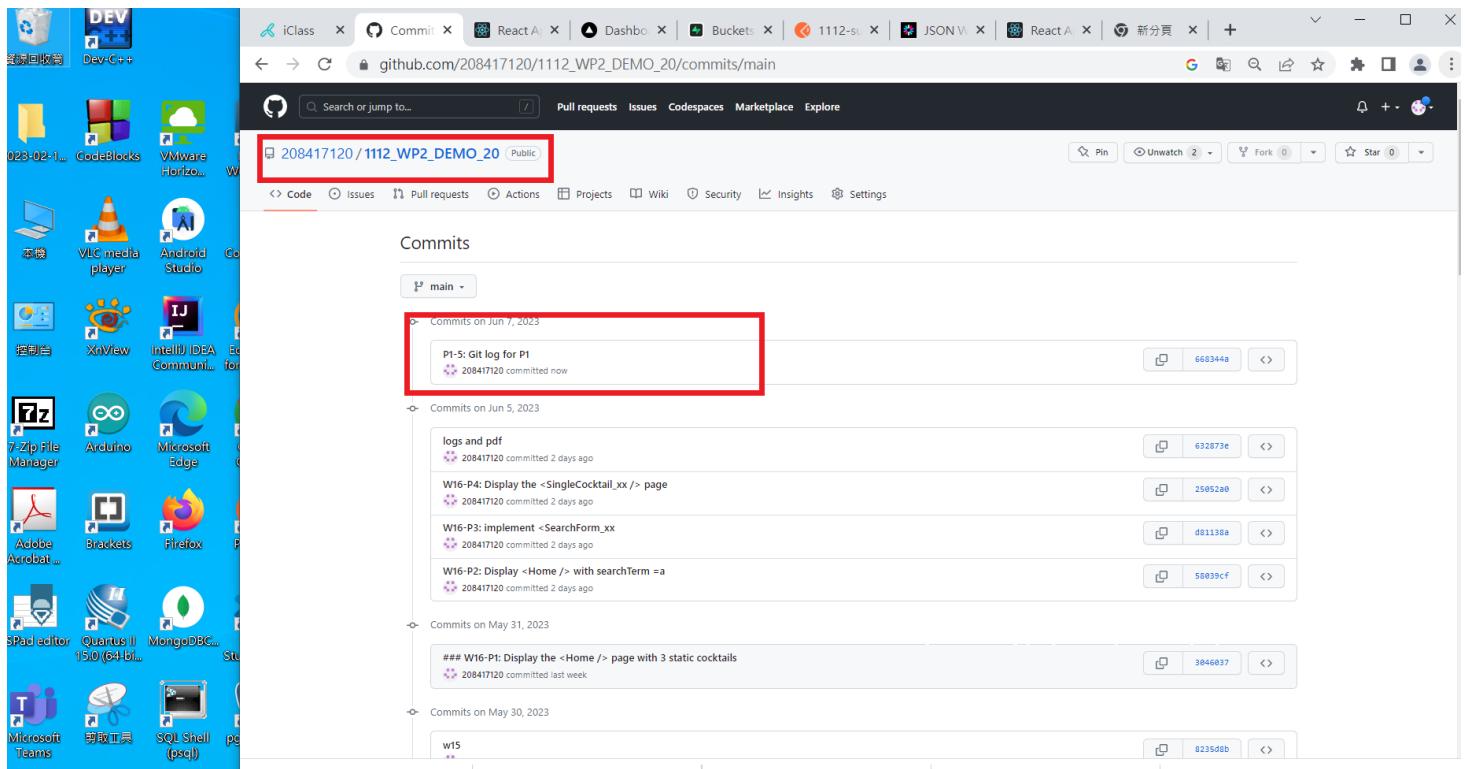


## (6%) P1-4: Show Your Vercel page for SupaDemo, with route /supa\_menu\_xx

My Vercel SupaDemo URL



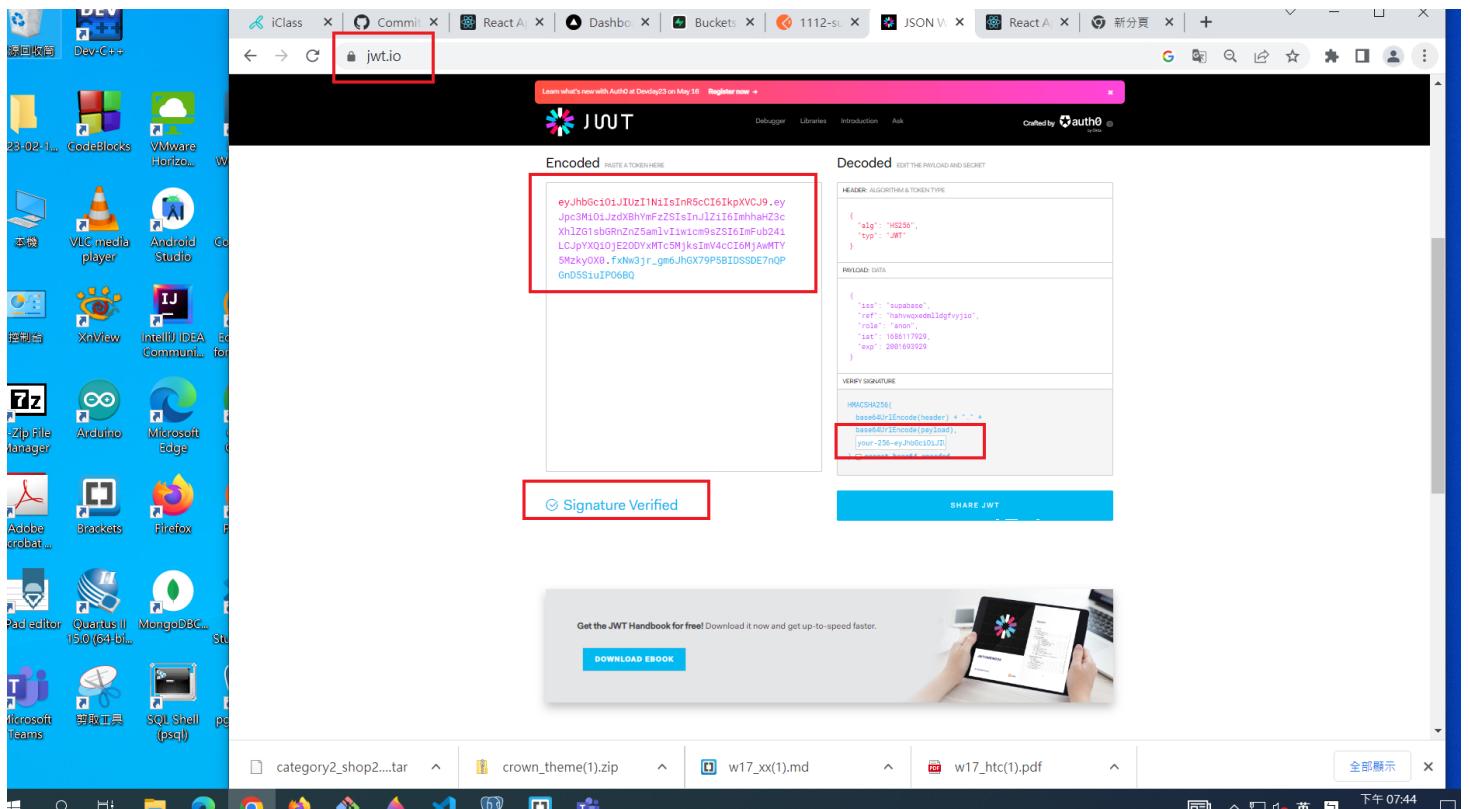
## (6%) P1-5: Git log for P1



## (35%) P2: Supabase & Postman API testing

### (10%) P2-1: show old anon key, and new anon key

old anon key



## new anon key by generating a new JWT secret

Encoded (PASTE A TOKEN HERE)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZ1IGIiXhahZ3cXH12G1sbGrNzN5aMlvIiwiemzS1EiIjFub241LCJpYXQiOjE200Y4MzgzbzUsInV4C16HjAwMTcxND3Nx0.YQ4GU74Inxxa_wZzAOJxZqeNAT_xX22yvMB16rJoc
```

Decoded (EDIT THE PAYLOAD AND SECRET)

HEADER ALGORITHM & TOKEN TYPE

```
{ "alg": "HS256", "typ": "JWT" }
```

Payload DATA

```
{ "iss": "supabase", "ref": "https://supabase.firebaseio.com", "role": "anon", "iat": 1691714375, "exp": 2091714375 }
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + '.' +  
  base64UrlEncode(payload),  
  your-128eyJhbGciOiI"
```

Secret base64 encoded

Signature Verified

SHARE JWT

Get the JWT Handbook for free! Download it now and get up-to-speed faster.

DOWNLOAD EBOOK

下午 07:48

Settings

208417120's Org / tku\_208417120

Your API is secured behind an API gateway which requires an API Key for every request. You can use the keys below to use Supabase client libraries.

anon public

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZ1IGIiXhahZ3cXH12G1sbGrNzN5aMlvIiwiemzS1EiIjFub241LCJpYXQiOjE200Y4MzgzbzUsInV4C16HjAwMTcxND3Nx0.YQ4GU74Inxxa\_wZzAOJxZqeNAT\_xX22yvMB16rJoc

This key is safe to use in a browser if you have enabled Row Level Security for your tables and configured policies.

service\_role secret

\*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\*

This key has the ability to bypass Row Level Security. Never share it publicly.

JWT Settings

JWT Secret

Generate a new JWT secret

A random secret will be created, or you can create your own.

This will invalidate all existing API keys! Generating a new JWT secret will invalidate all of your API keys, including your service role, also be restarted during this process, which will terminate any existing connections.

Generate a new secret

Generate a random secret

Create my own secret

category2\_shop2....tar crown\_theme(1).zip w17\_xx(1).md w17\_htc(1).pdf 全部顯示

下午 07:46

## (10%) P2-2: Use Postman to get all menu\_xx products

Postman, old anon key, fail to get all menu\_xx products

The screenshot shows a Windows desktop with a blue taskbar at the bottom. On the taskbar, there are icons for various applications like Dev-C++, VLC media player, and Microsoft Edge. The main window is Postman, which is displaying an API request. The URL is `https://hahvwqxedmldgfyvji.supabase.co/rest/v1/menu_20?select=*`. The Headers tab is selected, showing two entries: `apikey` with value `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...` and `Authorization` with value `Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...`. The Body tab shows a JSON response with a message: `"message": "Invalid API key", "hint": "Double check your supabase 'anon' or 'service_role' API key."`. The status bar at the bottom right of the Postman window shows the time as 07:49.

Postman, new anon key, successfully get all menu\_xx products

The screenshot shows a Windows desktop with a blue taskbar at the bottom. The taskbar icons are identical to the previous screenshot. The main window is Postman, displaying the same API request as before. However, the Headers tab now shows a single entry: `Authorization` with a long, complex value. The Body tab shows a successful JSON response with an item: `{ "id": 10, "title": "dani-rendina", "category": "breakfast", "price": 9.99, "img": "https://hahvwqxedmldgfyvji.supabase.co/storage/v1/object/public/demo-20/md_img/dani-rendina.jpg", "descrip": "I love ceramic table ware (check out that squeezer!) so i combined these elements" }`. The status bar at the bottom right shows the time as 07:54.

## (10%) P2-3: Use pgAdmin to connect to Supabase, and restore category2\_xx, shop2\_xx table (no RLS)

show category2\_xx, shop2\_xx in pgAdmin

The screenshot shows the pgAdmin interface connected to a Supabase database. The left sidebar lists various objects like Materialized Views, Operators, Procedures, Sequences, and Tables. The 'Tables' section is expanded, showing categories such as card\_20, card\_xx, category2\_20, product\_20, and shop2\_20. The 'category2\_20' and 'shop2\_20' tables are selected and highlighted with red boxes. The main query editor window contains the following SQL query:

```
1 SELECT * FROM public.category2_20
2 ORDER BY id ASC
```

The results grid displays five rows of data from the category2\_20 table:

id	name	size	remote_url	local_url
1	hats	[null]	https://i.ibb.co/cvpnL1/hats.png	/img/homepage/
2	jackets	[null]	https://i.ibb.co/px2Cc3/jackets.png	/img/homepage/
3	sneakers	[null]	https://i.ibb.co/0jqHpn/p/sneakers.png	/img/homepage/
4	womens	large	https://i.ibb.co/GCCdy8t/womens.png	/img/homepage/
5	mens	large	https://i.ibb.co/R70vBrQ/mens.png	/img/homepage/

Total rows: 5 of 5 Query complete 00:00:01.061 Ln 1, Col 1

In Postman, create a new request, and get five categories successfully

The screenshot shows the Postman interface with a collection named '1112-supabase-2N'. A specific request titled 'GET Get category data from category2\_20' is highlighted with a red box. The request URL is https://hahvwqxedmildgfvjio.supabase.co/rest/v1/category2\_20?select=\*. The 'Headers' tab shows two authorization headers: 'apikey' and 'Authorization'. The 'Authorization' header value is a Bearer token. The response body is displayed in JSON format, showing the same five categories as the pgAdmin query:

```
"id": 5,
"name": "mens",
"size": "large",
"remote_url": "https://i.ibb.co/R70vBrQ/mens.png",
"local_url": "/img/homepage/mens.png",
"link_url": "/crown2_xx/shop_xx/mens"
```

**(5%) P2-4: In Postman, get shop2\_xx table (no RLS) with cat\_id=x (last digit of your ID % 5)**

show all shop2\_xx products with cat\_id = 4 (in my case)

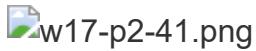
The screenshot shows a Windows desktop with a blue taskbar at the bottom. On the taskbar, there are icons for various applications including iClass, Commi, React A, Dashboard, shop2\_xx, get sho, JSON, React A, and a new tab. The desktop background is light blue with several icons for software like Dev-C++, CodeBlocks, VMware Horizon, VLC media player, Android Studio, XnView, IntelliJ IDEA, Microsoft Edge, 7-Zip File Manager, Arduino, Microsoft Edge, Adobe Acrobat, Brackets, Firefox, PSPad editor, Quartus II, MongoDB, Microsoft Teams, and SQL Shell (psql).  
The main window is a browser displaying the Postman application. The left sidebar shows 'Team Workspace' with two collections: '1112-supabase-2A' and '1112-supabase-2N'. The '1112-supabase-2N' collection contains a single item: 'GET get shop2\_20 table (no RLS) with cat\_id=x (last digit of your ...')'. This item has a red box around it.  
The right side of the Postman interface shows the request details. The method is 'GET' and the URL is 'https://hahvwqxedmildgfyjio.supabase.co/rest/v1/shop2\_20?select=\*&cat\_id=eq.4'. Below the URL, under 'Headers', there are two checked items: 'apikey' with value 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...' and 'Authorization' with value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6I...'.  
Under 'Body', the response is displayed in JSON format. A red box highlights the entire JSON object:

```
48:     "local_url": "/img/womens/yellow-track-suit.png"
49:
50:
51:   {
52:     "id": 34,
53:     "name": "White Blouse",
54:     "cat_id": 4,
55:     "price": 20,
56:     "remote_url": "https://i.ibb.co/qBcrsJg/white-vest.png",
57:     "local_url": "/img/womens/white-vest.png"
58:   }
```

**(5%) Git log for P2**

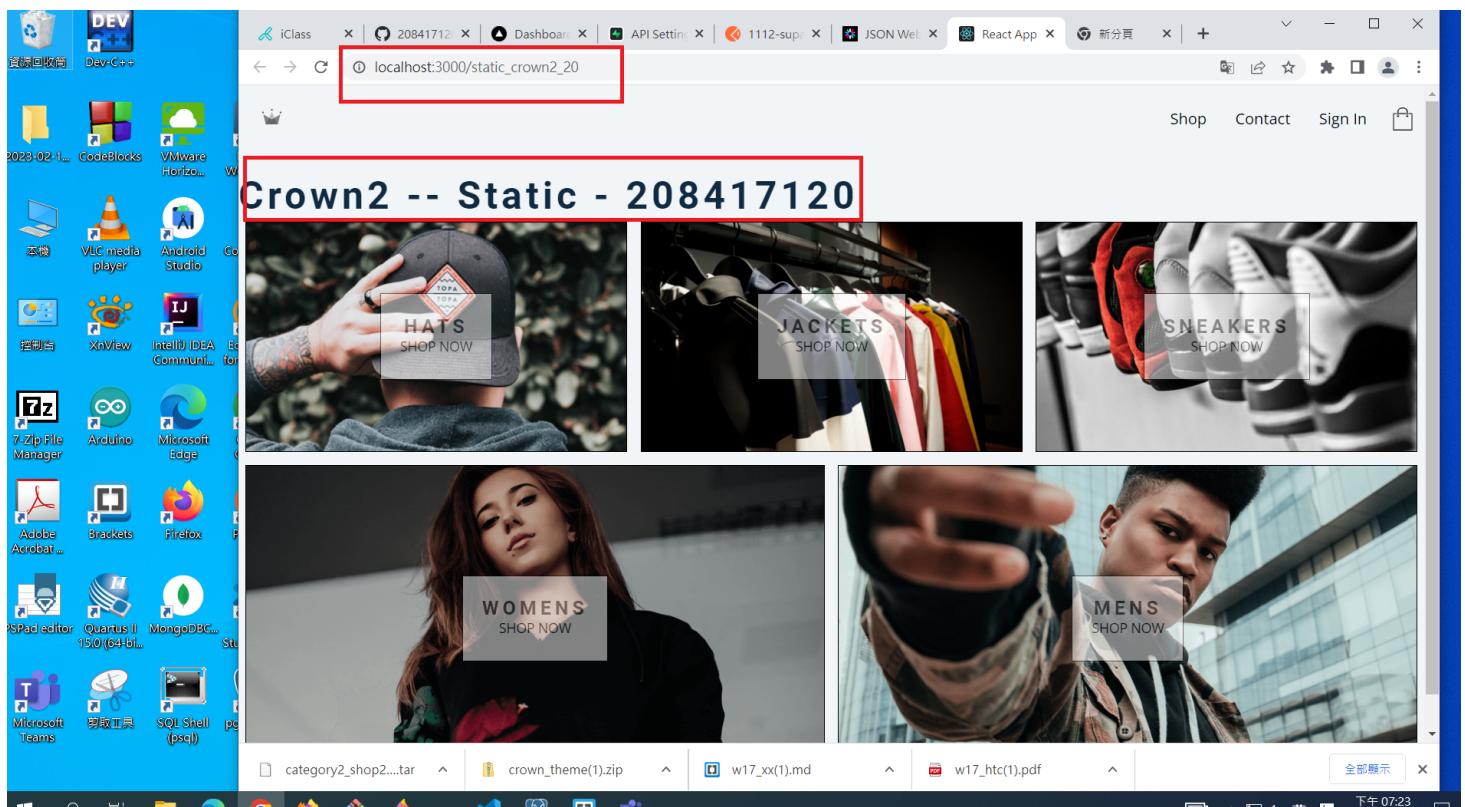
**remember to git log for P2**

9f7aa94 208417120      Wed Jun 7 20:16:57 2023 +0800      Git log for P2

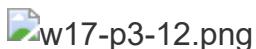


# (35%) P3: React Using Supabase

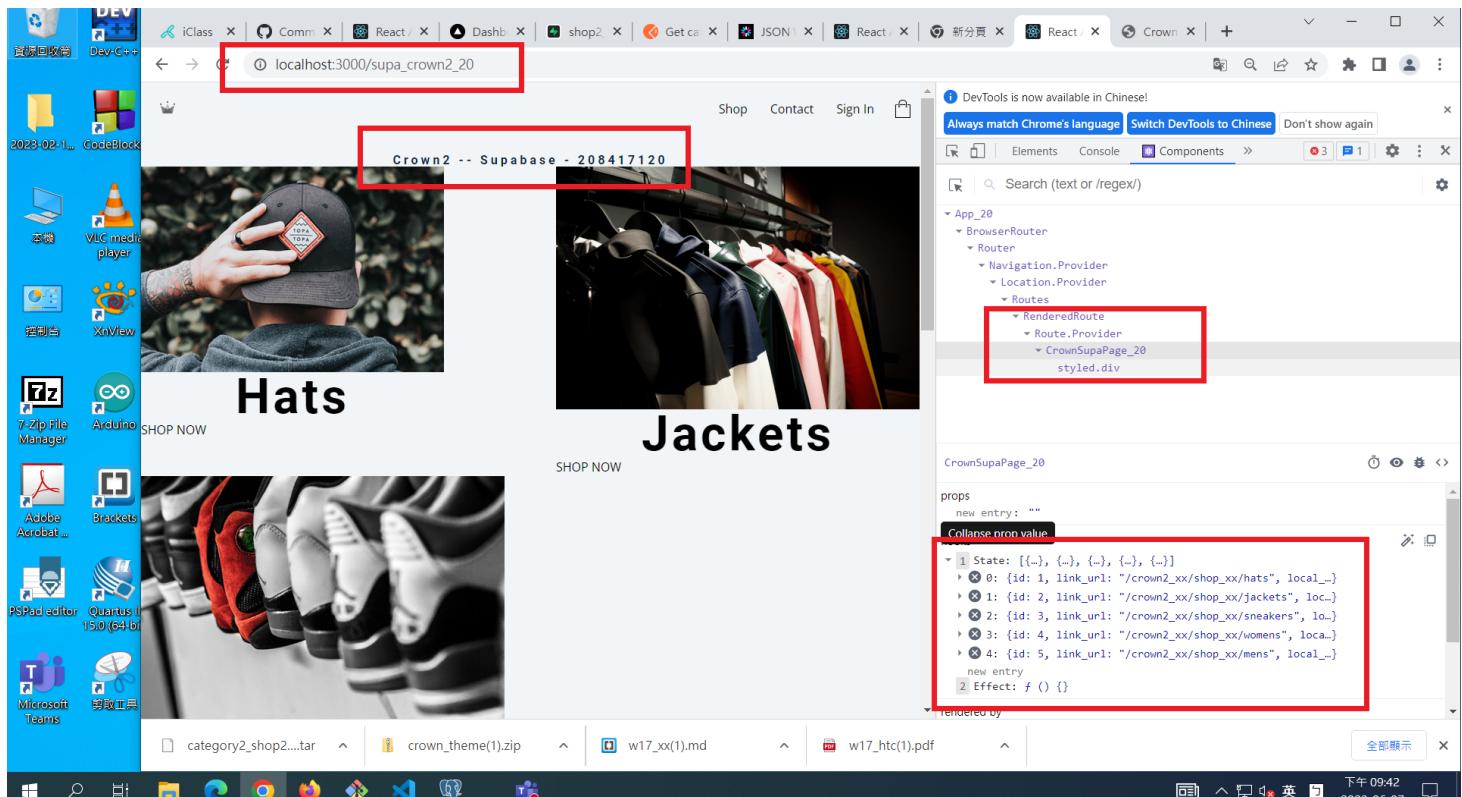
(10%) P3-1: Show Crown2 homepage with route /static\_crown2\_xx, by converting index.html directly to CrownStaticPage\_xx.js



## Wrapper code



(10%) P3-2: Show Crown2 homepage with route /supa\_crown2\_xx, by fetching category data from Supabase and put them into Crown2SupaPage\_xx.js



## code for fetching category data

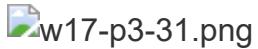
The screenshot shows a code editor with the file `CrownSupaPage_20.js` open. The code uses the `useEffect` hook to fetch data from Supabase. It then maps over the fetched categories to render a menu. The menu items are wrapped in `div` elements with a class of `menu-item`, which includes an image and a link to the category page. The code editor's sidebar shows the project structure, including files like `index.scss`, `App_20.js`, and other components and pages.

```
const { id, name, size, remote_url, local_url, link_url } = category;
return (
  <div className={`${size}menu-item`}>
    <img className='background-image' src={local_url} alt='' />
    <a href={link_url} className='content'>
      <h1>{name}</h1>
      <span>SHOP NOW</span>
    </a>
  </div>
);
};

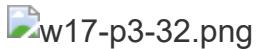
</div>
</div>
</Wrapper>
);

export default CrownSupaPage_20;
```

**(10%) P3-3: Show shop2 products with route /supa\_shop2\_xx, by fetching shop2 products from Supabase and put them into Shop2SupaPage\_xx.js**



### code for fetching category data



**(5%) P3-4: All logs for P1, P2, P3**

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2023-6-6"
c598f5b 208417120      Wed Jun 7 21:45:34 2023 +0800    p3
9f7aa94 208417120      Wed Jun 7 20:16:57 2023 +0800    Git log for P2
668344a 208417120      Wed Jun 7 19:35:28 2023 +0800    P1-5: Git log for P1
```

```
User@E201-04 MINGW64 /d/1112_WP2_DEMO_20 (main)
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2023-6-6"
c598f5b 208417120      Wed Jun 7 21:45:34 2023 +0800    p3
9f7aa94 208417120      Wed Jun 7 20:16:57 2023 +0800    Git log for P2
668344a 208417120      Wed Jun 7 19:35:28 2023 +0800    P1-5: Git log for P1

User@E201-04 MINGW64 /d/1112_WP2_DEMO_20 (main)
```