

(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-supa-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。

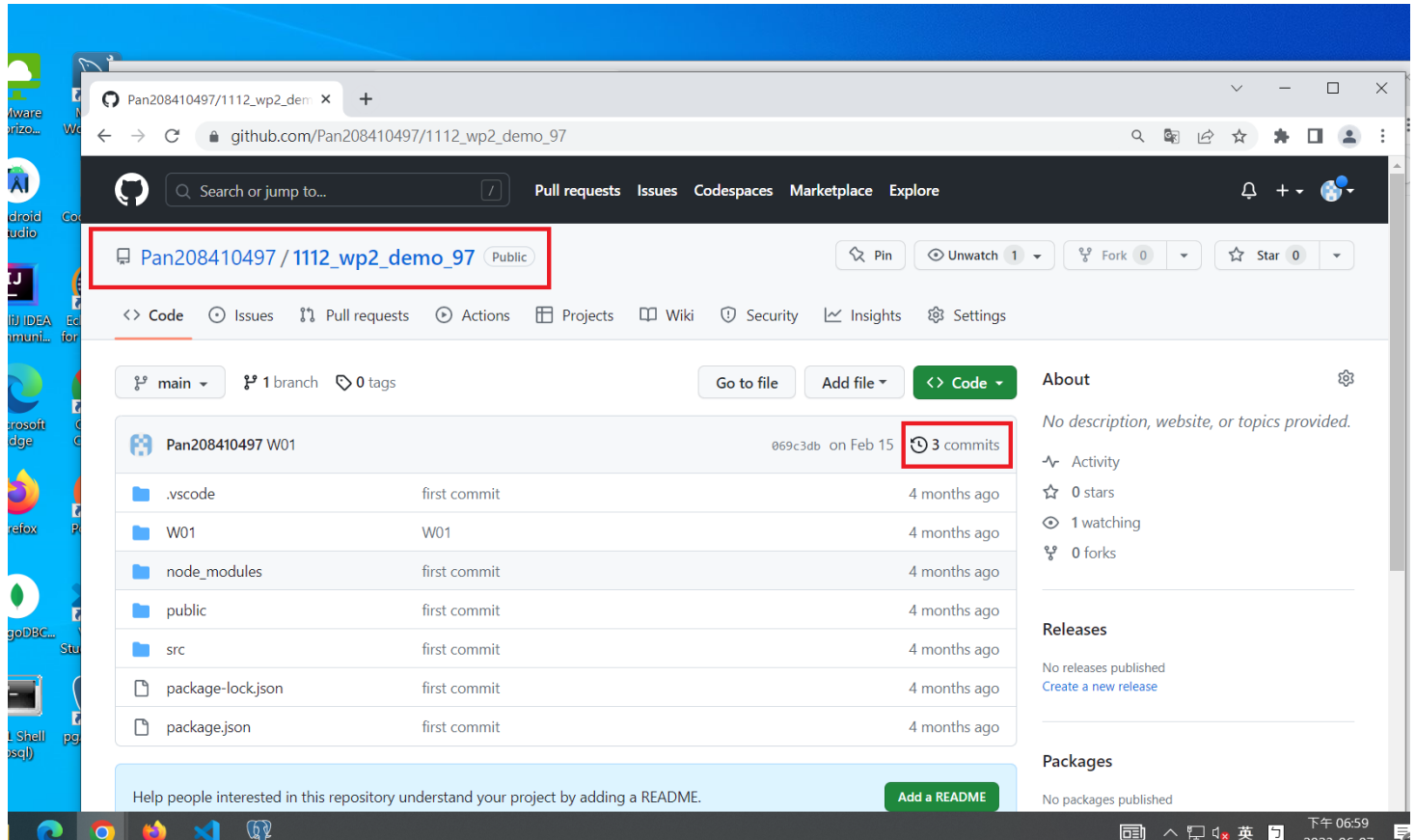
Your (Name, ID): (name, id)

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

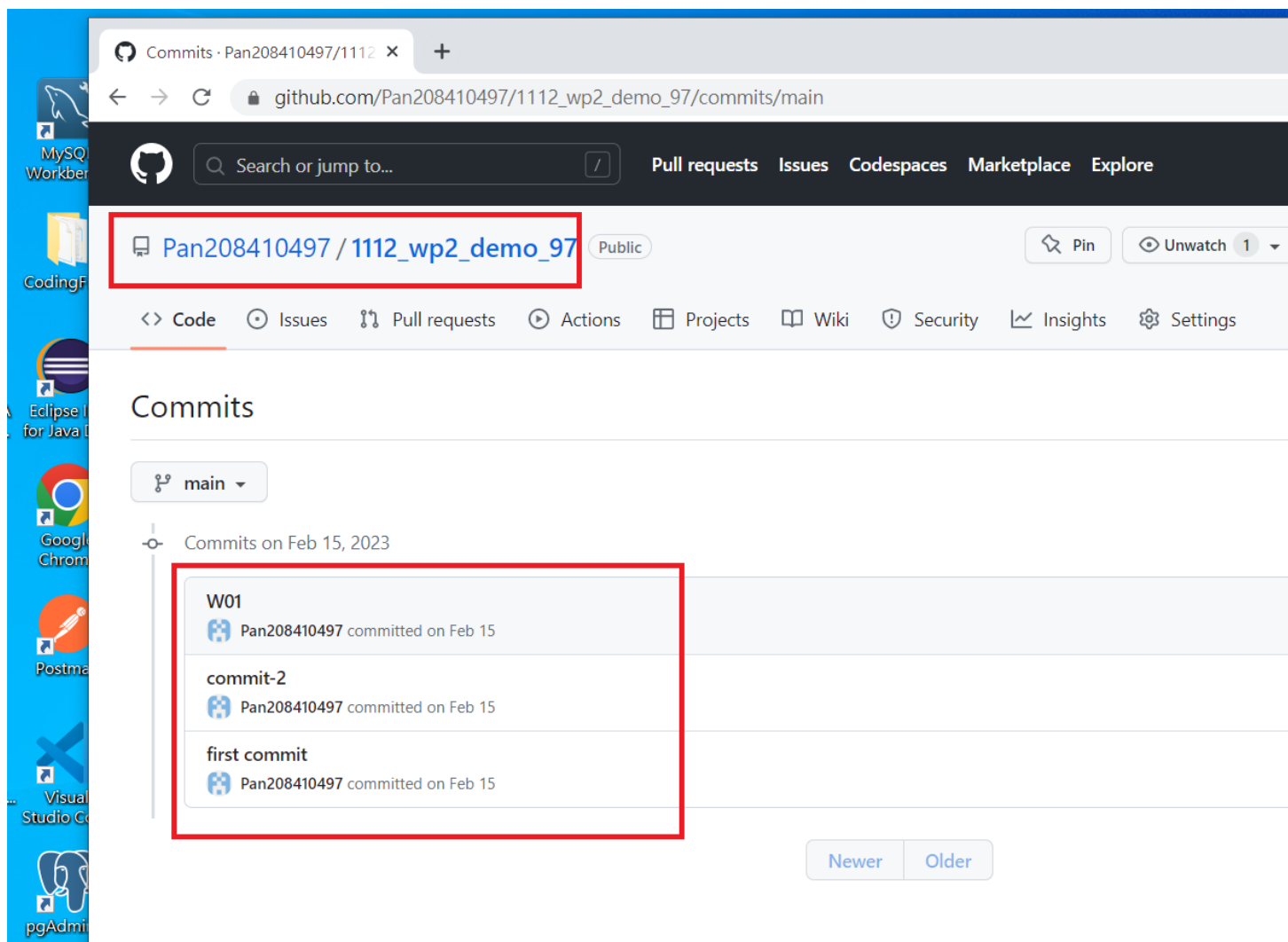
(30%) P1: Show Info of Your Github Repo, Client Supa Github, Vercel as follows

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

My Github Repo URL

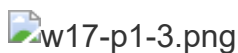


(6%) P1-2: Show Your Github commits for week 15 and 16



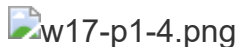
(6%) P1-3: Show Your Github SupaDemo URL

[My Github SupaDemo URL](#)

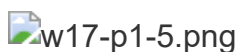


(6%) P1-4: Show Your Vercel page for SupaDemo, with route /supa_menu_xx

[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

The screenshot shows the jwt.io website interface. The browser's address bar displays 'jwt.io'. The website's header includes the JWT logo, navigation links (Debugger, Libraries, Introduction, Ask), and a 'Crafted by auth0 by Okta' badge. A pink banner at the top reads 'Learn what's new with Auth0 at Devday23 on May 16 Register now →'. The main content area displays a JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImJleXhncGZvcnR0dHVnc21hbXp0Iiwicm9sZSI6ImFub24iLCJpYXZ5L290YyMzZG4NjUsImV4cCI6MjAwMTcxNDg2NX0.FZDaPqk3HpYZmSmBWswpa9a3iEn1I8at7kn9i2Nm8ZI`. The token is broken down into three parts: Header (Algorithm & Token Type), Payload (Data), and Signature (Verify Signature). The Header shows `{ "alg": "HS256", "typ": "JWT" }`. The Payload shows `{ "iss": "supabase", "ref": "reyxgpfortttugsmamzt", "role": "anon", "iat": 1686138865, "exp": 2001714865 }`. The Signature section shows the HMACSHA256 calculation: `HMACSHA256(base64UrlEncode(header) + ".", base64UrlEncode(payload), secret base64 encoded)`. The secret is `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImJleXhncGZvcnR0dHVnc21hbXp0Iiwicm9sZSI6ImFub24iLCJpYXZ5L290YyMzZG4NjUsImV4cCI6MjAwMTcxNDg2NX0.FZDaPqk3HpYZmSmBWswpa9a3iEn1I8at7kn9i2Nm8ZI`. A blue button labeled 'SHARE JWT' is visible. At the bottom, a red box highlights the 'Signature Verified' status.

new anon key by generating a new JWT secret

The screenshot shows the JWT.io website interface. The browser's address bar displays 'jwt.io'. The website has a dark header with the JWT logo and navigation links: 'Debugger', 'Libraries', 'Introduction', and 'Ask'. A pink banner at the top reads 'Learn what's new with Auth0 at Devday23 on May 16 Register now →'. The main content area is divided into two panels. The left panel contains a large text box with a generated JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InJleXhncGZvenR0dHVnc21hbXp0Iiwicm9sZSI6ImFub24iLCJpYXQiOiJlZ20DYxMzg4NjUsImV4cCI6MjAwMTcxNDg2NX0.04FPSQTaFXld3q_uzLTGFZYTJgkQRjzQVQKtrymAUSU`. The right panel shows the token's structure: 'Type of token' is 'JWT', 'HEADER: ALGORITHM & TOKEN TYPE' is '{ "alg": "HS256", "typ": "JWT" }', and 'PAYLOAD: DATA' is '{ "iss": "supabase", "ref": "reyxgpfortttugsmamzt", "role": "anon", "iat": 1686138865, "exp": 2001714865 }'. Below the payload, the 'VERIFY SIGNATURE' section shows the HMACSHA256 algorithm and the secret key 'K4GbCIes38SLIVoFxJdBv:'. A blue button labeled 'SHARE JWT' is at the bottom right. A red box highlights the 'Signature Verified' status at the bottom left.

Learn what's new with Auth0 at Devday23 on May 16 Register now →

JWT

Debugger Libraries Introduction Ask

Crafted by auth0 by Okta

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InJleXhncGZvenR0dHVnc21hbXp0Iiwicm9sZSI6ImFub24iLCJpYXQiOiJlZ20DYxMzg4NjUsImV4cCI6MjAwMTcxNDg2NX0.04FPSQTaFXld3q_uzLTGFZYTJgkQRjzQVQKtrymAUSU

Type of token

HEADER: ALGORITHM & TOKEN TYPE

PAYLOAD: DATA

VERIFY SIGNATURE

HMACSHA256(

base64UrlEncode(header) + "." +

base64UrlEncode(payload),

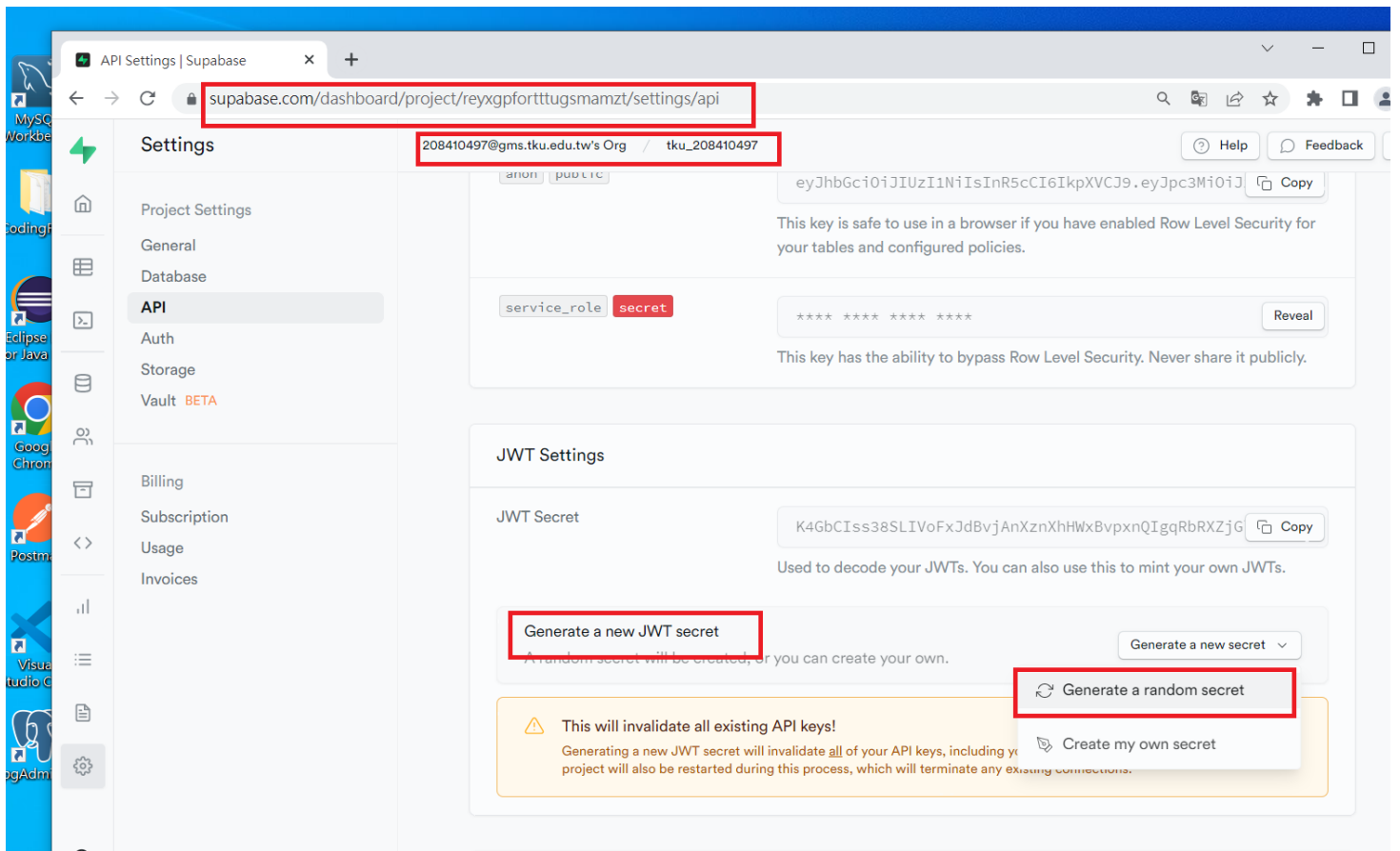
K4GbCIes38SLIVoFxJdBv:

) ☐ secret base64 encoded

Signature Verified

SHARE JWT

下午 07:5 2023-06-



(10%) P2-2: Use Postman to get all menu_xx products

Postman, old anon key, fail to get all menu_xx products

The screenshot displays the Postman interface with a GET request to the URL `https://reyxgpfortttugsmamzt.supabase.co/rest/v1/cart_97?select=*`. The request is configured with the following headers:

Key	Value	Description
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.30.1	
<input checked="" type="checkbox"/> Accept	*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> apikey	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...	

The response is shown in the Body tab, indicating a 401 Unauthorized status with the following JSON message:

```
1 {
2   "message": "Invalid API key",
3   "hint": "Double check your Supabase 'anon' or 'service_role' API key."
4 }
```

Postman, new anon key, successfully get all menu_xx products

HomeWorkspacesAPI NetworkExplore

Search Postman

Invite

Upgrade

My Workspace

NewImport

Overview

GET https://reyxgpfortttugsmamzt.supabase.co/rest/v1/cart_97?select=*

No Environment

Collections

My first collection

First folder inside collection

Second folder inside collection

Create a collection for your requests

Create Collection

GET https://reyxgpfortttugsmamzt.supabase.co/rest/v1/cart_97?select=*

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

Postman-Token<calculated when request is sent>

Host<calculated when request is sent>

User-AgentPostmanRuntime/7.30.1

Accept*/*

Accept-Encodinggzip, deflate, br

Connectionkeep-alive

apikeyeyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Body

200 OK641 ms1.91 KB

Save Response

PrettyRawPreviewVisualizeJSON

```
49  "amount": 1
50  {
51    "id": 8,
52    "title": "Samsung Galaxy Flip 4",
53    "price": 999.99,
54    "img": "https://reyxgpfortttugsmamzt.supabase.co/storage/v1/object/public/demo-97/cart-97/samsungs7.png",
55    "amount": 1
56  }
57  }
58
```


(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 4 interface. The left sidebar displays the database structure, with 'category2_97' and 'shop2_97' tables highlighted under the 'public' schema. The main pane shows a query result for 'category2_97'.

Query: `select * from category2_97`

id	[PK] integer	name	character varying	size	character varying	remote_url	character varying	local_url	character varying
1	1	hats	[null]			https://i.ibb.co/cvpntL1/hats.png	/img/homepage/		
2	2	jackets	[null]			https://i.ibb.co/px2tCc3/jackets.png	/img/homepage/		
3	3	sneakers	[null]			https://i.ibb.co/0jqHnpn/sneakers.png	/img/homepage/		
4	4	womens	large			https://i.ibb.co/GCCdy8t/womens.png	/img/homepage/		
5	5	mens	large			https://i.ibb.co/R70vBrQ/men.png	/img/homepage/		

Total rows: 5 of 5 Query complete 00:00:00.141 Ln 1, Col 27

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

The screenshot shows the Postman interface with a new GET request created. The URL is `https://reyxgpfortttugsmamzt.supabase.co/rest/v1/category2_97?select=*`. The Headers tab is active, showing the following headers:

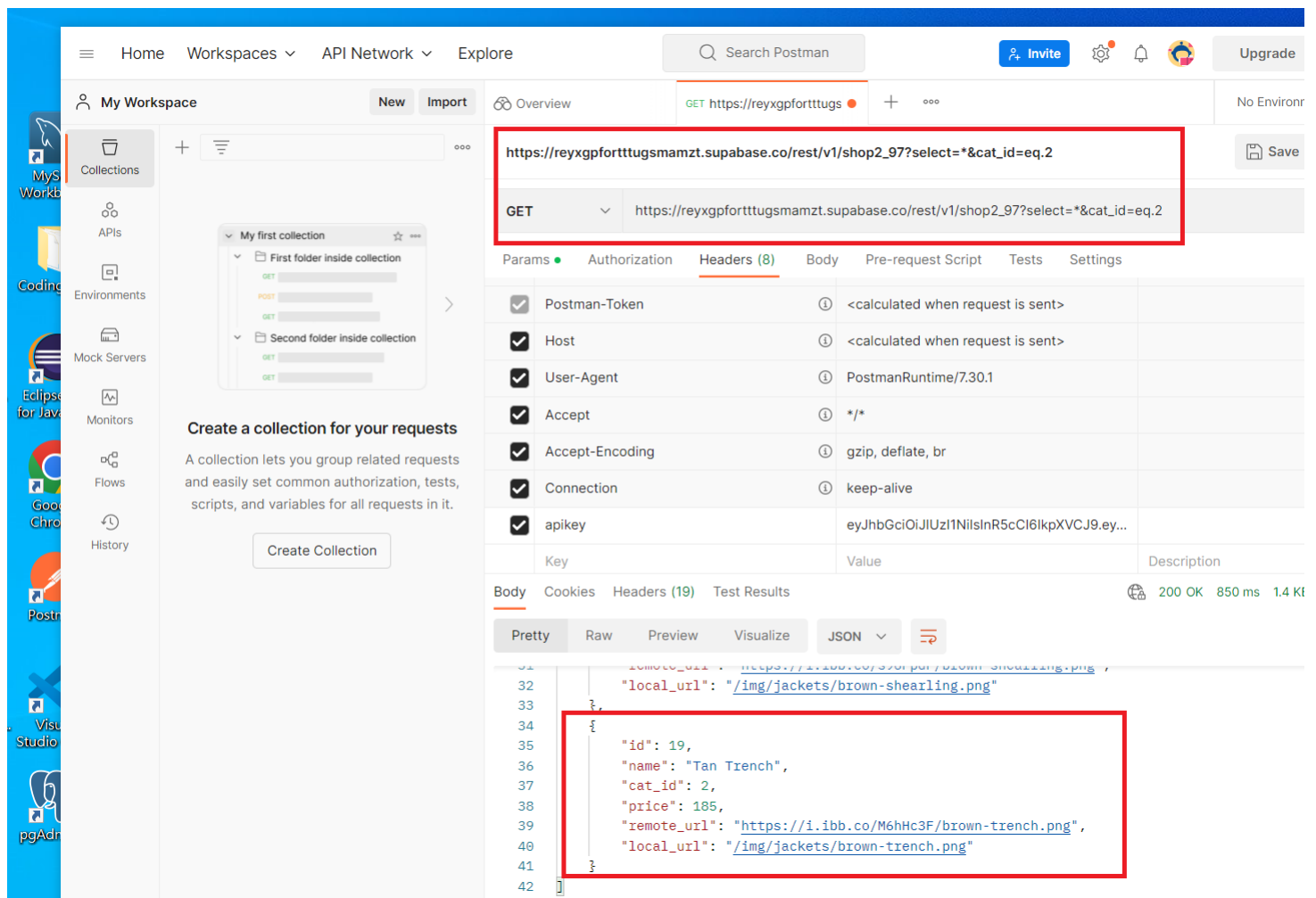
Key	Value	Description
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.30.1	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
apikey	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...	

The Body tab is active, showing a JSON response:

```
31  {
32    "local_url": "/img/homepage/womens.png",
33    "link_url": "/crown2_xx/shop_xx/womens"
34  },
35  {
36    "id": 5,
37    "name": "mens",
38    "size": "large",
39    "remote_url": "https://i.ibb.co/R70vBrQ/men.png",
40    "local_url": "/img/homepage/mens.png",
41    "link_url": "/crown2_xx/shop_xx/mens"
42  }
43 }
```

(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)



(5%) Git log for P2

remember to git log for P2


4f4951b htchung Wed Jun 7 16:57:57 2023 +0800 P2: Supabase & Postman API testing

(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

w17-p3-11.png


Wrapper code

w17-p3-12.png


(10%) P3-2: Show Crown2 homepage with route /supa_crown2_xx, by fetching category data from Supabase and put them into Crown2SupaPage_xx.js

w17-p3-21.png


code for fetching category data

w17-p3-22.png

(10%) P3-3: Show shop2 products with route /supa_shop2_xx, by fetching shop2 products from Supabase and put them into Shop2SupaPage_xx.js


w17-p3-31.png

code for fetching category data

w17-p3-32.png

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

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2023-6-6"
```

w17-p3-4.png