

(100%) Web 程式設計 2N 期末上機考-2 -- 開放考試

2024-4-12 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，情節嚴重者會送校。
2. iClass 上請繳交 md_xx.zip，還有 f2_xx.zip 2 個檔 (node_modules 請刪除)
3. 請直接將答案寫在 f2_xx.md 上，老師實作的圖片放在 f2_htc.pdf 上，請依照老師所給的圖片來實作並標註
4. 請用老師提供的 f2_xx code 來實作 P2, P3, P4 等 3 題
5. 跟期中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。跟期中考題無關的檔案及目錄名稱有 xx，則不必修改。
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. 有標註要 git log 的題目，必須要做，這是我們上課練習的一部分。
8. 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): (林泓君, 207410290)

f2 開放網路題目 (50%)

- P1-1 (10%): 10 分
- P1-2 (10%): 10 分
- P1-3 (10%): 10 分
- P1-4 (10%): 10 分
- P2 (15%): 15 分
- P3 (20%): 0 分
- P4-1 (5%): 5 分
- P4-2 (10%): 0 分
- P4-3 (10%): 0 分

總分: 60 分

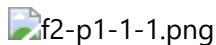
(40%) P1: Show Info of Your Github Repo, Client Supa Github, Vercel as follows

(10%) P1-1: Show Your Github Repo for all md and demo

=> My Github repo URL

https://github.com/htchung/1122-wp2-2N_xx

=> 下圖左邊框顯示所有的 demo，右邊框顯示所有的 md



Your Answer

=> My Github repo URL

My Github repo URL for demo

=> 下圖左邊框顯示所有的 demo · 右邊框顯示所有的 md

The screenshot shows a GitHub repository interface. On the left, there is a sidebar with a 'Files' section. A red box highlights the 'demo' folder under 'main'. On the right, a main content area shows a table of commits. A red box highlights the table header and the first few rows. The table has columns for 'Name', 'Last commit message', and 'Last commit date'. The commits listed are:

Name	Last commit message	Last commit date
..		
md	upload final answer	last week
w01_crud_90	upload files	now
w03_crud_90	upload w03-crud_90	3 months ago
w04	upload w04 folder	3 months ago
w05_taskSupa_90/clientSupa_90	upload w05.pdf	3 months ago
w06/1122-2N-tasksVercel-90	W6-P2: Delete a task in Supabase	3 months ago
w10_mid21_90	rename	last month
w11_mid21	upload w13	last week
w12_mid21_90	upload w13	last week
w13_wildOasis_starter_90	upload w13	last week
w14_final_90	W11-P2: Use Postman to test GET /api/blogs, and DELETE /api/blogs/:id	last week
w15_wildOasis_90	upload final answer	last week

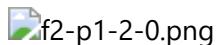
=> git log

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-11"
f1ddd91 sky001 Wed Jun 12 18:53:53 2024 +0800 upload f2-p1-1
```

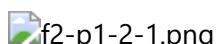
(10%) P1-2: Show all your git logs from W10 to W15

=> My Github URL for git logs · 標註本學期 git log 次數

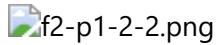
My Github URL for git logs



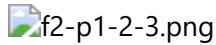
=> git log for w15 and w14



=> git log for w13 and w12

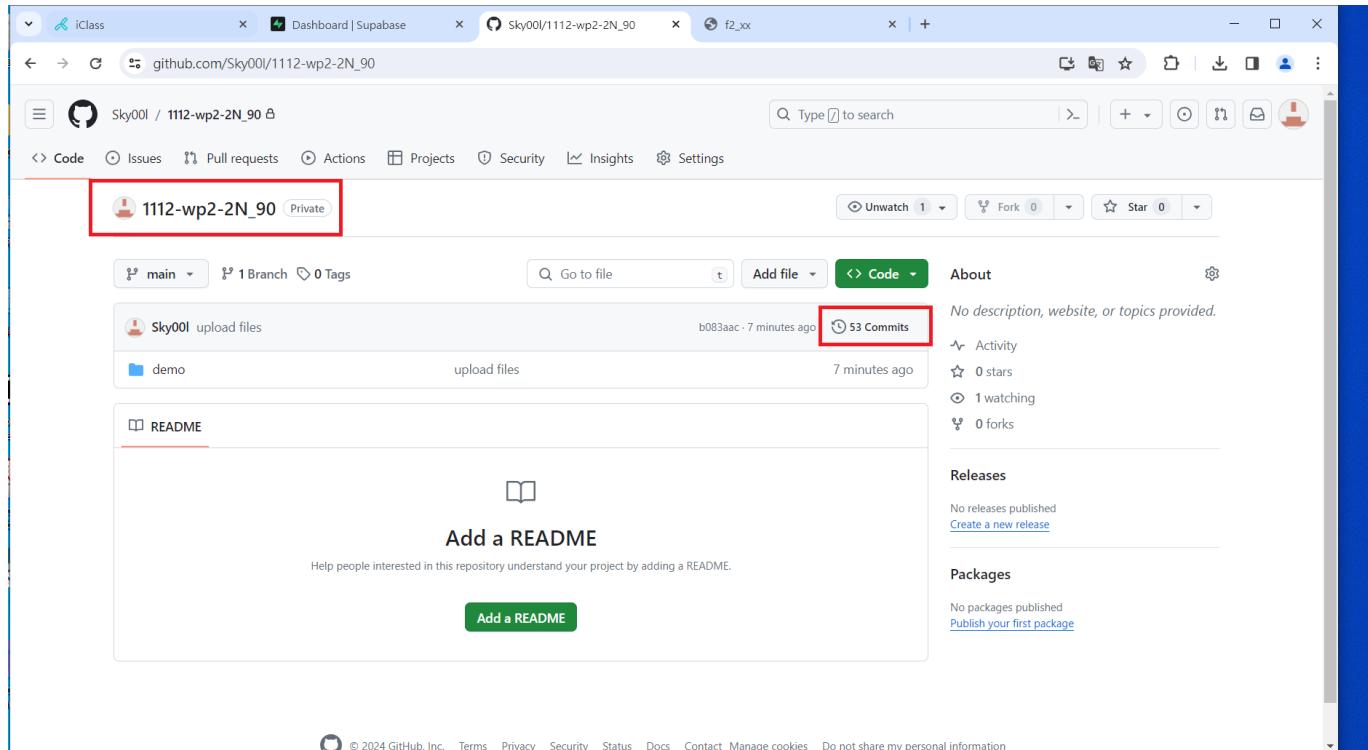


=> git log for w11 and w10



Your Answer

=> My Github URL for git logs · 標註本學期 git log 次數



The screenshot shows a GitHub repository page for '1112-wp2-2N_90'. The repository is private. It has 1 branch and 0 tags. There is 1 commit from 'Sky001' uploaded 7 minutes ago. A red box highlights the repository name '1112-wp2-2N_90' and the '53 Commits' link. The repository has 0 stars, 1 watching, and 0 forks. There are no releases or packages published.

=> git log for w15 and w14

upload w15 files

- Sky001 committed last week
- upload w14.pdf,w15.pdf**
- Sky001 committed last week
- W15-P3: Implement login using LoginForm, useLogin, login (apiAuth.jsx), navigate to /dashboard when successful**
- Sky001 committed last week
- W15-P2: For landing page, get 8 cabins data from Supabase**
- Sky001 committed last week
- W15-P1: Show static page <LandingStatic_xx /> using route /landing**
- Sky001 committed last week

Commits on Jun 4, 2024

upload w11.pdf

- Sky001 committed last week
- W11-P3: Use Postman to test DELETE /api/blogs/clearAll, and POST /api/blogs/loadAll**
- Sky001 committed last week
- W11-P2: Use Postman to test GET /api/blogs, and DELETE /api/blogs/:id**
- Sky001 committed last week

upload w14.pdf

- Sky001 committed last week
- W14-P2: Make jonas code work, and show cabins and bookings info**
- Sky001 committed last week
- W14-P1: Use sql to create schemas of cabins_xx, guests_xx, bookings_xx, settings_xx, and use Uploader to upload all data**
- Sky001 committed last week

upload w13

- Sky001 committed last week

upload w13.pdf

- Sky001 committed last week

=> git log for w13 and w12

Sky001 committed last week

upload w13

- Sky001 committed last week
- upload w13.pdf**
- Sky001 committed last week
- W13-P2: Get first booking_xx data, with corresponding cabin_xx and guest_xx data**
- Sky001 committed last week

Commits on Jun 3, 2024

W13-P1: Get first Cabins_xx data from Supabase

Sky001 committed last week

Commits on May 11, 2024

upload w12.pdf

- Sky001 committed last month
- W12-P4: Insert a blog using React Query**
- Sky001 committed last month
- W12-P2: Update a blog with id =20**
- Sky001 committed last month

Commits on May 8, 2024

W12-P3: Use React Query to read all blogs from Node server

Sky001 committed last month

W12-P1: Insert a blog using local json array

Sky001 committed last month

Commits on May 7, 2024

W11-P1: Make mobileBtn work correctly

Sky001 committed last month

=> git log for w11 and w10

The screenshot shows two separate GitHub repository pages for 'f2_xx' and 'f2_xx'.

Top Repository (W12):

- Commit: W12-P2: Update a blog with id = 20 (by Sky001 last month)
- Commit: W12-P3: Use React Query to read all blogs from Node server (by Sky001 last month)
- Commit: W12-P1: Insert a blog using local json array (by Sky001 last month)
- Commit: W11-P1: Make mobileBtn work correctly (by Sky001 last month)
- Commit: rename (by Sky001 last month)
 - add w10_mid21.90 (by Sky001 last month)
 - W10-P2: Add Navbar_xx with 3 menu items (by Sky001 last month)
 - W10-P1: Create HomeLayout_xx for all routes (by Sky001 last month)
- Commit: W6-P2: Delete a task in Supabase (by Sky001 3 months ago)
- Commit: W6-P3: Get all tasks from node server (by Sky001 3 months ago)
- Commit: MIG-D1: Update a task in Supabase (by Sky001 3 months ago)

Bottom Repository (W15):

- Commit: W15-P3: Implement login using LoginForm, useLogin, login (apiAuth.jsx), navigate to /dashboard when successful (by Sky001 last week)
- Commit: W15-P2: For landing page, get 8 cabins data from Supabase (by Sky001 last week)
- Commit: W15-P1: Show static page <LandingStatic_xx /> using route /landing (by Sky001 last week)
- Commit: upload w11.pdf (by Sky001 last week)
- Commit: W11-P3: Use Postman to test DELETE /api/blogs/clearAll, and POST /api/blogs/loadAll (by Sky001 last week)
- Commit: W11-P2: Use Postman to test GET /api/blogs, and DELETE /api/blogs/id (by Sky001 last week)
- Commit: upload w14.pdf (by Sky001 last week)
- Commit: W14-P2: Make jonas code work, and show cabins and bookings info (by Sky001 last week)
- Commit: W14-P1: Use sql to create schemas of cabins_xx, guests_xx, bookings_xx, settings_xx, and use Uploader to upload all data (by Sky001 last week)
- Commit: upload w13.pdf (by Sky001 last week)
- Commit: W13-P2: Get first booking_xx data, with corresponding cabin_xx and guest_xx data (by Sky001 last week)
- Commit: upload w12.pdf (by Sky001 last week)
- Commit: W12-P1: Get first Cabin_xx data from Supabase (by Sky001 last week)

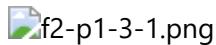
=> git log

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-11"
f11f7b6 sky001 Wed Jun 12 19:02:05 2024 +0800 upload f2-p1-2 answer
```

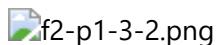
(10%) P1-3: 佈署 w15 demo 到 Vercel · 並顯示如下資訊

w15 demo deploy to Vercel

=> 顯示在 Vercel 上的 landing page



=> 透過 Vercel 頁面，以含有你學號的 email login，登入後在 Dashboard 頁面顯示你個人及登入之 user 資訊



Your Answer

=> 顯示在 Vercel 上的 landing page

A screenshot of a web browser displaying the landing page for 'The Wild Oasis'. The URL '1122-2-n-tasks-vercel-90.vercel.app/landing' is highlighted in a red box. The page features a header with a logo and the text 'The Wild Oasis (207410290)'. Below the header, there is a section titled 'Our Luxury Cabins' with a descriptive paragraph. Six cabin options are listed in a grid: Cabin 001 (\$250 / night), Cabin 002 (\$350 / night), Cabin 003 (\$300 / night), Cabin 004 (\$500 / night), Cabin 005 (\$350 / night), and Cabin 006 (\$800 / night). Each cabin listing includes a thumbnail image, the cabin number, the price per night, the guest capacity (e.g., 'For up to 2 guests'), and a 'Details & reservation →' link.

=> 透過 Vercel 頁面，以含有你學號的 email login，登入後在 Dashboard 頁面顯示你個人及登入之 user 資訊

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'f2_xx' and displays a dashboard for 'The Wild Oasis'. The dashboard includes a logo, navigation links for Home, Bookings, Cabins, Users, and Settings, and a 'SAMPLE DATA' section with 'Upload ALL' and 'Upload bookings ONLY' buttons. A red box highlights the title 'Final 2 Exam of Web Prog2, 2024' and the guest information: Guest name: 林泓君, Guest student ID: 207410290, Supabase email: 207410290@gmail.com, and Supabase user ID: 9e30df4d-6968-46cf-a745-810d8719319b.

The right side of the screen shows the Chrome DevTools Application tab. It lists various storage types and their contents. A red box highlights the 'Local storage' entry for the URL 'https://1122-2-n-tasks-velcel-90.vercel.app'. The storage item contains a key-value pair where the key is 'sb-ophlhndtmgrbqpyaatag-auth-token' and the value is '{"access_token": "eyJhbGciOiJIUzI1NiIsImtpZCI6Ik1qbldxeFNNd0xzRk5McG...'.

```
access_token: "eyJhbGciOiJIUzI1NiIsImtpZCI6Ik1qbldxeFNNd0xzRk5McG..."  
expires_at: 1718195851  
expires_in: 3600  
refresh_token: "PhyE6R8Y_2vbSngIMZSehw"  
token_type: "bearer"  
user: {id: "9e30df4d-6968-46cf-a745-810d8719319b", aud: "authenticated", app_metadata: {provider: "email", providers: ["email"]}, confirmed_at: "2024-06-04T14:59:57.471657Z", created_at: "2024-06-04T14:59:57.467067Z", email: "207410290@gmail.com", email_verified_at: "2024-06-04T14:59:57.471657Z", id: "9e30df4d-6968-46cf-a745-810d8719319b", identities: [{"identity_id": "1d9637ff-1400-4294-b37e-d4a50e0aa885", is_anonymous: false}, last_sign_in_at: "2024-06-12T11:37:31.882574861Z", phone: "", role: "authenticated", updated_at: "2024-06-12T11:37:31.884369Z", user_metadata: {}}
```

=> Dashboard 之程式碼 (code)

```
src > pages > Dashboard_xx.jsx > ...
1 import styled from 'styled-components';
2
3 const Layout = styled.main`
4   display: grid;
5   grid-template-columns: 48rem;
6   align-content: center;
7   justify-content: center;
8   gap: 3.2rem;
9   background-color: var(--color-grey-50);
10  list-style-type: none;
11  h1 {
12    font-size: 18px;
13    text-align: center;
14    font-weight: bolder;
15  }
16
17 function Dashboard_xx() {
18  return (
19    <Layout>
20      <div>
21        <h1>Final 2 Exam of Web Prog2, 2024</h1>
22      </div>
23      <div>
24        <h2>
25          <li>Guest name:林泓君</li>
26          <li>Guest student ID:207410290</li>
27          <li>Supabase email:207410290@gmail.com</li>
28          <li>Supabase user ID:9e30df4d-6968-46cf-a745-810d8719319b</li>
29        </h2>
30      </div>
31    </Layout>
32  );
33
```

=> git log

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-11"  
b501498 sky001 Wed Jun 12 19:42:24 2024 +0800 upload f2-p1-3 answer
```

(10%) P1-4: 透過 P1-3 之程式登入後取得 Supabase access token, api keys

=> 登入後，取得 access token



=> 從 jwt.io 檢視 access token



=> Project URL & anon key



Your Answer

=> 登入後 · 取得 access token

The screenshot shows a browser window with several tabs open. The active tab is '1122-2-n-tasks-vercel-90.vercel.app/dashboard'. The page displays a guest's information: name (林泓君), student ID (207410290), Supabase email (207410290@gmail.com), and Supabase user ID (9e30df4d-6968-46cf-a745-810d8719319b). On the left, there's a sidebar with links for Home, Bookings, Cabins, Users, and Settings. Below that is a 'SAMPLE DATA' section with 'Upload ALL' and 'Upload bookings ONLY' buttons.

In the bottom right corner of the browser window, the Chrome DevTools Network tab is visible. It shows a request to 'https://1122-2-n-tasks-vercel-90.vercel.app'. In the 'Application' panel, under 'Storage', there is a key-value pair for 'access_token'. The value is a JSON object containing various fields like 'access_token', 'refresh_token', 'user', and 'app_metadata'. A red box highlights the 'access_token' field.

```

{
  "access_token": "eyJhbGciOiJIUzI1NiIsImtpZCI6Ik1qbWdxefNId0xzRk5hcc...",
  "refresh_token": "PhyE6R8Y_2vbSngIMZSehw",
  "token_type": "bearer",
  "user": {
    "id": "9e30df4d-6968-46cf-a745-810d8719319b",
    "aud": "authenticated"
  },
  "app_metadata": {
    "provider": "email",
    "providers": ["email"]
  },
  "aud": "authenticated",
  "confirmed_at": "2024-06-04T14:59:57.471657Z",
  "created_at": "2024-06-04T14:59:57.467067Z",
  "email": "207410290@gmail.com",
  "email_verified": true,
  "exp": "2024-06-05T14:59:57.471657Z",
  "id": "9e30df4d-6968-46cf-a745-810d8719319b",
  "is_anonymous": false,
  "last_sign_in_at": "2024-06-12T11:37:31.882574861Z",
  "phone": null,
  "role": "authenticated",
  "updated_at": "2024-06-12T11:37:31.884369Z",
  "user_metadata": {}
}

```

=> 從 jwt.io 檢視 access token

eyJhbGciOiJIUzI1NiIsImtpZCI6Ik1qbWdxeFN
Nd0xzRk5McGs1lCJ0eXAiO1JKV1QifQeyJhdQ
i0iJhdXRoZWS0aWhdGVkIiwiZXhwIjoxNzE4MT
k10DUxlCJpYXQ10je3MTgx0TiyNTEsInlzcyl6I
mh0dHBz018vb3BobGhuZHRTz3JicXB5YWFOYWcu
c3VwYUhc2Uy28vYXVBaC92MSIsInN1Yi16ij1
1MzBkZjRkLTY5NjgtNDZjZi1hNzQ1LTgxMQ4Nz
ESMzE5ViIsImVtWlsIjoIMjA3NDEwMjkwGGdtY
WlsLmNvbSIsInBob25lIjoiIiwiYXBwX21ldGFK
YXRhIj07InByb3ZpZGVyIjoiZW1haWw1CJwcm9
2aWRlcmlsIzsZW1haWw1XX0sInVzZxJ [Issuer who created and signed this token]
RhdGEiOn9LCJyb2x1IjoiYXV0aGVudGljYXR1Z
CisImFhbCI6ImFhbDeliCJhbXIi0l7Im1ldGhv
ZCI6InBh3cN3b3JkTiwidGltZXN0YW1wi0xNzE
4MTkyMjUxfV0sInNlc3Npb25faWQi0iITM2ViY2
YyoC03NjQSLTQ5NzMtYTE3Yi1hODBiNjK4MjhM
zUiLCJpc19hb9ueW1vdXMi0mZhbnHnfQ.XbrYT
zk9Lr7wIKAcF961980PEhutspQq9TFMfkC0Wi|

HEADER: ALGORITHM & TOKEN TYPE	
{	"alg": "HS256", "typ": "JWT"
PAYLOAD: DATA	
{	"aud": "authenticated", "exp": 1718195851, "iat": 1718192251, "iss": "https://ophlndtmgrbqpyaatag.supabase.co/auth/v1", "sub": "20740290@gmail.com", "email": "20740290@gmail.com", "phone": "", "app_metadata": { "provider": "email", "providers": ["email"]}, "user_metadata": {}, "role": "authenticated", "aa1": "aa1", "amr": [("method": "password", "timestamp": 1718192251)], "session_id": "53ebcf28-7649-4973-a17c-a80b69828f35", "is_anonymous": false }
VERIFY SIGNATURE	

=> Project URL & anon key

supabase.com/dashboard/project/ophlndtmgrbqpyaatag/settings/api

API Settings

Project URL: https://ophlndtmgrbqpyaatag.supabase.co

Project API keys

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

anon public

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJpc3Mi0jJzdXBhYml

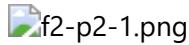
service_role secret

=> git log

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-11"
5844df7 sky001 Wed Jun 12 19:50:57 2024 +0800 upload f2-p1-4 answer
```

以下 P2, P3, P4 題目請透過老師提供的 f2_xx 來實作，跟 P2, P3, P3 有關的檔案及目錄，如有 xx 需改成學號後兩碼，跟期中考無關的檔案及目錄出現 xx，則不需要修改。

(15%) P2: 實作路由 /about_xx，點選上方選單 About_xx，可以顯示 About 訊息如下圖



Your Answer

=> Chrome 顯示圖片如上

```

<!DOCTYPE html>
<html lang="en">
  <head> ...
    </head>
    <body>
      <div id="root">
        <header class="bg-slate-100 border-b border-primary-900 px-8 py-5"> ...
          </header>
          <div class="flex-1 px-8 py-12 grid"> ...
            <main class="max-w-7xl mx-auto w-full">
              <div class="grid grid-cols-5 gap-x-24 gap-y-32 text-lg items-center">
                ...
                <div class="col-span-3"> ...
                  <div class="col-span-2">
                    ...
                    
                  </div>
                </div>
              </div>
            </main>
          </div>
        </div>
      </div>
    </body>
  </html>

```

=> 相關重點 code 說明

The screenshot shows a code editor with two files open:

- App_xx.jsx**: Contains a component named `About_xx` which imports `Link` from `react-router-dom` and `Header_xx` from `../components/Header_xx'`. It returns a component structure with `<Header_xx />` and a `<div class='flex-1 px-8 py-12 grid'>...`.
- about.html**: A template file with a `<header>` element containing `<div class="flex-1 px-8 py-12 grid">...`.

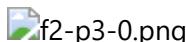
=> git log

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-11"
ac62932 sky001 Wed Jun 12 20:48:18 2024 +0800 upload f2-p2 all answer
```

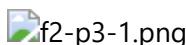
(20%) P3: 請實作 /cabins_xx/:id，當點選 Details and reservation 時，顯示該 cabin id 的詳細資訊，如下圖。

如果直接從全部 `cabins` 中篩選 `id`，顯示所要的畫面，而沒有實作 `/cabins_xx/:id` 路由者，得 10 分。

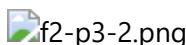
以下老師顯示 Cabin 001 及 008 兩筆。同學們顯示請以你的學號後兩個數字分別除以 8 取餘數來決定。舉例，如果學號最後兩個數字為 85，則先顯示 Cabin 005，後顯示 Cabin 008。



=> 點選 Cabin 001 之 Details 按鈕，顯示 Cabin 001 詳細資訊

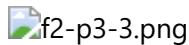


=> 點選 Cabin 008 之 Details 按鈕，顯示 Cabin 001 詳細資訊

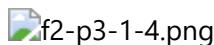


Your Answer

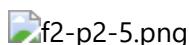
=> 以你的學號最後 1 個數字，依老師上述規則，點選對應的 **Cabin** 之 **Details** 按鈕，顯示該 **Cabin** 詳細資訊



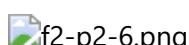
=> 以你的學號最後第 2 個數字，依老師上述規則，點選對應的 **Cabin** 之 **Details** 按鈕，顯示該 **Cabin** 詳細資訊



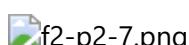
=> **CabinById_xx.jsx** 相關重點 **code** 說明



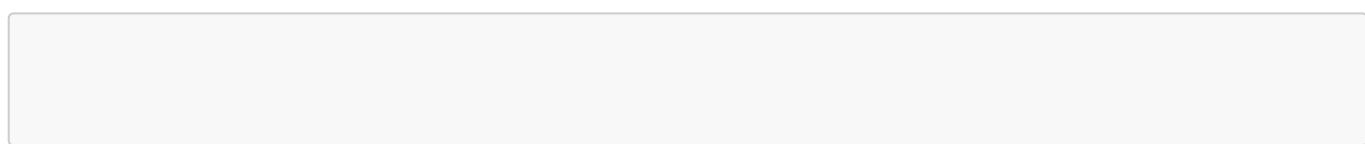
=> **useCabinById_xx.jsx** 相關重點 **code** 說明



=> **apiCabins.jsx** 相關重點 **code** 說明

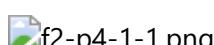


=> **git log**

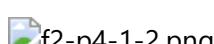


(5%) P4-1: 請在 Supabase 加入一筆有你

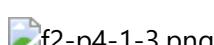
=> 用 **SQL** 指令新增一筆有你學號的 **email**



=> 在 **guests_xx** 中檢查剛新增、有你學號的 **guest id**



=> 列出在 **Supabase** 所有 **cabin** 的 **id** 及 **name**



Your Answer

=> 用 **SQL** 指令新增一筆有你學號的 **email**

The screenshot shows the Supabase SQL Editor interface. On the left, there's a sidebar with various database objects like 'YOUR QUERIES' and tables such as 'blog_in_90_table', 'Bookings Table', 'company&store', etc. The main area contains a SQL query editor with the following code:

```

1 -- Always verify the output before executing
2
3 insert into
4   guests_90 (
5     "fullName",
6     "email",
7     "nationality",
8     "nationalID",
9     "countryFlag"
10    )
11   values
12   (
13     '林泓君',
14     '207410290@gms.tku.edu.tw',
15     'Taiwan',
16     '207410290',
17     'https://falcdn.com/tw.svg'
18   );
19

```

The results section shows a message: "Success. No rows returned". To the right, there's an AI Assistant panel with a message: "Hi, how can I help you? I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve." There are also buttons for "Feedback", "Close Assistant", "Exclude database metadata in queries", and "Clear history".

=> 在 **guests_xx** 中檢查剛新增、有你學號的 guest id

The screenshot shows the Supabase Table Editor for the 'guests_90' table. The table has columns: id, created_at, fullName, email, and nationality. The data includes rows for various users, including one with id 31 and email 207410290@gms.tku.edu.tw. The 'guests_90' table is highlighted with a red box.

	id	created_at	fullName	email	nationality
	16	2024-06-03 18:51:08.148727	Fatima Ahmed	fatima@example.com	Pakistan
	17	2024-06-03 18:51:08.148727	David Smith	david@gmail.com	Australia
	18	2024-06-03 18:51:08.148727	Marie Dupont	marie@gmail.com	France
	19	2024-06-03 18:51:08.148727	Ramesh Patel	ramesh@gmail.com	India
	20	2024-06-03 18:51:08.148727	Fatimah Al-Sayed	fatimah@gmail.com	Kuwait
	21	2024-06-03 18:51:08.148727	Nina Williams	nina@hotmail.com	South Africa
	22	2024-06-03 18:51:08.148727	Taro Tanaka	taro@gmail.com	Japan
	23	2024-06-03 18:51:08.148727	Abdul Rahman	abdul@gmail.com	Saudi Arabia
	24	2024-06-03 18:51:08.148727	Julie Nguyen	julie@gmail.com	Vietnam
	25	2024-06-03 18:51:08.148727	Sara Lee	sara@gmail.com	South Korea
	26	2024-06-03 18:51:08.148727	Carlos Gomez	carlos@yahoo.com	Colombia
	27	2024-06-03 18:51:08.148727	Emma Brown	emma@gmail.com	Canada
	28	2024-06-03 18:51:08.148727	Juan Hernandez	juan@yahoo.com	Argentina
	29	2024-06-03 18:51:08.148727	Ibrahim Ahmed	ibrahim@yahoo.com	Nigeria
	30	2024-06-03 18:51:08.148727	Mei Chen	mei@gmail.com	Taiwan
	31	2024-06-12 13:15:24.866844	林泓君	207410290@gms.tku.edu.tw	Taiwan

=> 列出在 Supabase 所有 cabin 的 id 及 name

	id	created_at	name	maxCapacity	regularPrice	discount	image
	1	2024-06-03 18:51:08.430315	001	2	250	0	https://ophlhndtn
	2	2024-06-03 18:51:08.430315	002	2	350	25	https://ophlhndtn
	3	2024-06-03 18:51:08.430315	003	4	300	0	https://ophlhndtn
	4	2024-06-03 18:51:08.430315	004	4	500	50	https://ophlhndtn
	5	2024-06-03 18:51:08.430315	005	6	350	0	https://ophlhndtn
	6	2024-06-03 18:51:08.430315	006	6	800	100	https://ophlhndtn
	7	2024-06-03 18:51:08.430315	007	8	600	100	https://ophlhndtn
	8	2024-06-03 18:51:08.430315	008	10	1400	0	https://ophlhndtn

=> git log

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-11"
2cc9390 sky001 Wed Jun 12 21:20:27 2024 +0800 upload f2-p4-1 answer
```

(10%) P4-2: 在 P3 題下面加入一個按鈕 Add Reservation，可以新增一筆預訂，請用 P4-1 的 guest id 來預訂，訂購資訊會寫入 bookings_xx 中

考試時，為了節省時間，不需要產生 form，可以直接參考下圖右方老師資料來進行輸入，但是 guest id 必須符合 P4-1 你所建立的 guest id。

為了節省時間，本題不須使用 React Query。

=> Chrome 成功寫入並正確回傳資訊



=> 在 Supabase bookings_xx 檢查剛新增的資料，並全部列出

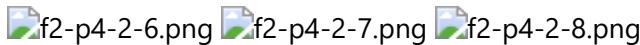


Your Answer

=> Chrome 成功寫入並正確回傳資訊



=> 在 Supabase bookings_xx 檢查剛新增的資料，並全部列出

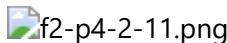


=> CabinById.jsx 相關的 code

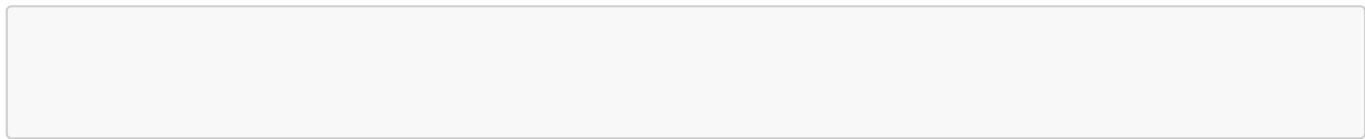
本題可以直接呼叫 apiBooking.jsx，不需要透過 React Query



=> apiBooking.jsx 相關的 code



=> git log

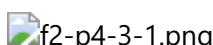


(5%) P4-3: 在 P3 題下面加入一個按鈕 Update Reservation，可以修改 P4-2 所建立的預訂資料

本題要修改的資料有 totalPrice，因為計算錯誤，計算公式為 $(250+50)*5 = 1500$ ，另外 observations 改成 'We will be late check-in after 10pm, please help pick us up at the airport.'

為了節省時間，本題不須使用 React Query。

=> Chrome 成功寫入並正確回傳資訊

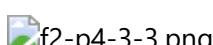


=> 在 Supabase bookings_xx 檢查更新的資料



Your Answer

=> Chrome 成功更新並正確回傳資訊



=> 在 Supabase bookings_xx 檢查剛更新的資料，並全部列出

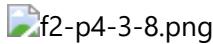


=> CabinById.jsx 相關的 code

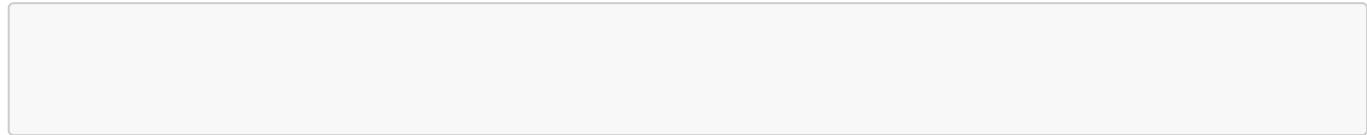
本題可以直接呼叫 apiBooking.jsx，不需要透過 React Query



=> **apiBooking.jsx** 相關的 **code**



=> **git log**



(5%) P4-4: 在 P3 題下面加入一個按鈕 **Delete Reservation**，可以刪除 P4-2 所建立的預訂資料

bookingId 必須是 P4-2 所建立的 id

為了節省時間，本題不須使用 React Query。

=> **Chrome** 成功刪除



=> 在 **Supabase** 下 **SQL** 指令，顯示已經找不到刪除的資料了



Your Answer

=> **Chrome** 成功寫入並正確回傳資訊

=> **Chrome** 成功刪除



=> 在 **Supabase** 下 **SQL** 指令，顯示已經找不到刪除的資料了



=> **CabinById.jsx** 相關的 **code**

本題可以直接呼叫 **apiBooking.jsx**，不需要透過 React Query



=> **apiBooking.jsx** 相關的 **code**



=> **git log**

