

# (50%) Web 程式設計一 期中考 1 -- 斷網考試

2024-11-6, 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 mid1\_xx.pdf，還有 md\_mid1.zip, client\_xx (src.zip), server\_xx.zip 三個壓縮檔，壓縮前請將 node\_modules 全部砍掉
3. 請直接將答案寫在 md\_mid1\_xx/mid1\_xx.md 上，老師出題及圖片放在 mid1\_htc.pdf 上，請依照老師所給的圖片來實作並標註
4. 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。
5. 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。
6. 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 (25%) -- server 後端實作

- P1 (15%): ? 分
- P2 (10%): ? 分

## Part 2 (25%) -- client 前端實作

- P4 (10%): ? 分
- P5 (15%): ? 分

**總分: ? 分**

# Part 1 (25%) -- server 後端實作

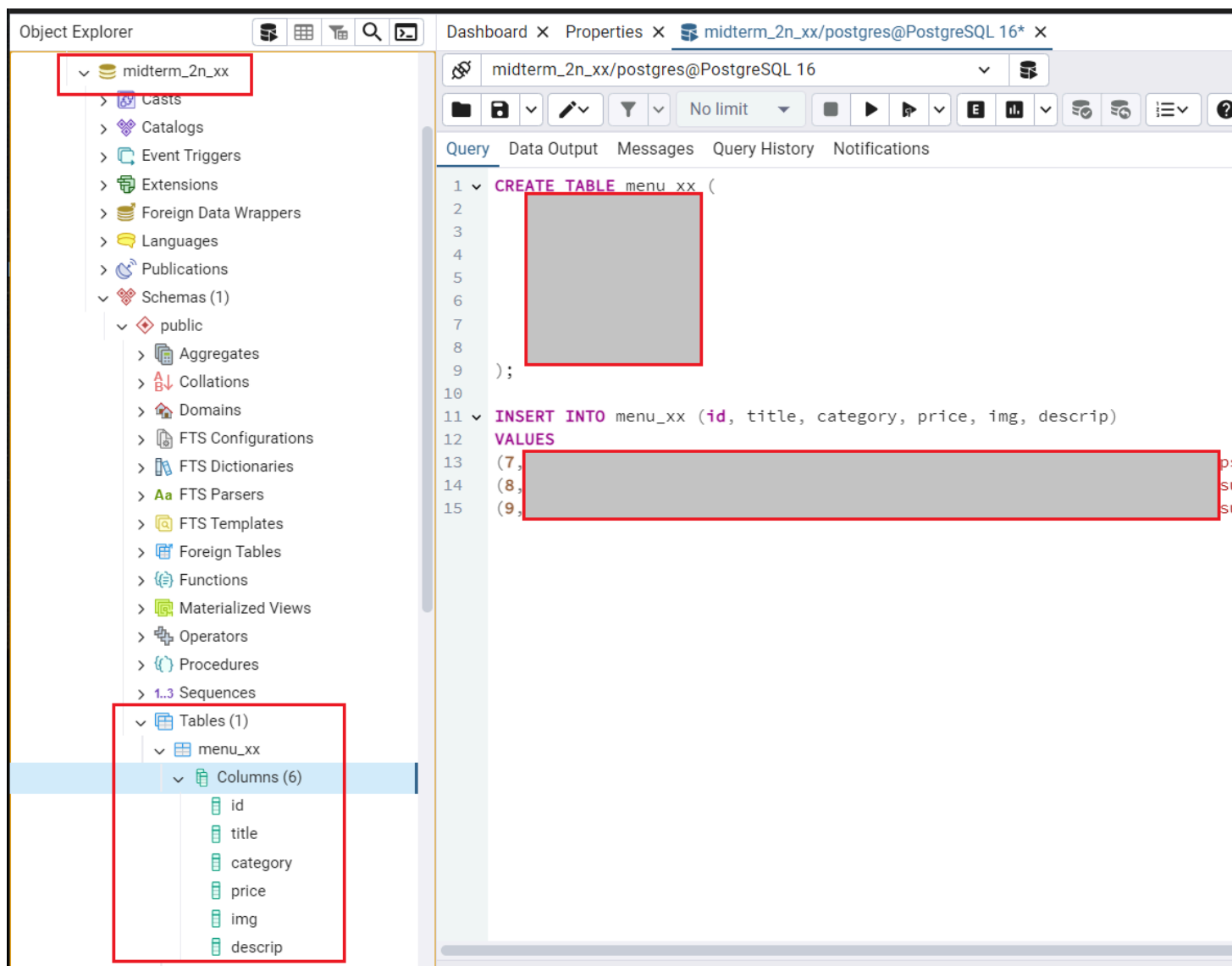
(15%) P1: 在 pgAdmin 產生的一個 database，名稱為 **midterm\_2n\_xx**，用 SQL 指令在 pgAdmin 建立一個 **product\_xx** 資料表，然後按照下列規則，用 SQL 指令匯入 4 筆資料

請用老師提供的 `utils/midterm_xx/menu_data.js` 中的 json 陣列，根據你的學號最後一碼，比對 `id`，往下取 3 筆，如果到底，則循環回到第 1 筆繼續往下取。學號最後一碼如為 0，則從第 1 筆往下取 3 筆。

下圖是以學號最後一碼為 7 當成範例。

=> 顯示 database, table and SQL

記得圖片 7 資料要存入 `/mid_xx/item-7.jpeg` 中，以方面 client 前端存取圖片時的需要，其餘類推



=> 顯示 database.js 設定，及顯示 3 筆資料

The screenshot shows a VS Code editor with a project named 'MIDTERM\_CLOSED\_XX'. The file explorer on the left shows a directory structure with files like 'client\_xx', 'md\_mid1\_xx', 'md\_quiz1\_xx', 'server\_xx', 'node\_modules', 'public', 'routes', 'utils', 'data', 'menu\_data.js', 'menu\_xx.sql', 'database-demo.js', 'database-local.js', 'database-supabase.js', 'views', '.env', '.gitignore', 'package-lock.json', and 'package.json'. The 'database.js' file is highlighted in the file explorer and its content is visible in the editor. The code defines a PostgreSQL pool and a query to fetch menu items. The terminal shows the command 'connecting local database midterm\_2n\_xx' and the server running on port 3000. The PostgreSQL query result is displayed in the 'Data Output' tab, showing 3 rows of menu items.

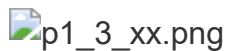
```
server_xx > utils > JS database.js > ...
1 import pg from 'pg';
2 const { Pool } = pg;
3
4 let pool;
5
6 if (process.env.DATABASE === 'LOCAL') {
7   pool = new Pool({
8     // ...
9   });
10 }
11
12 console.log('connecting local database', pool);
13
14 options.database);
```

connecting local database midterm\_2n\_xx  
Server running on port 3000  
[nodeemon] restarting due to changes...  
[nodeemon] restarting due to changes...  
[nodeemon] starting node -env file -env server.js  
connecting local database midterm\_2n\_xx  
Server running on port 3000

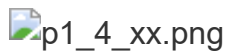
id	title	category	price	img
1	Bacon Overflow	breakfast	8.99	/menu/images
2	American Classic	lunch	12.99	/menu/images
3	Quarantine Buddy	lunch	13.99	/menu/images

## Your Answer

=> 顯示 database, table and SQL

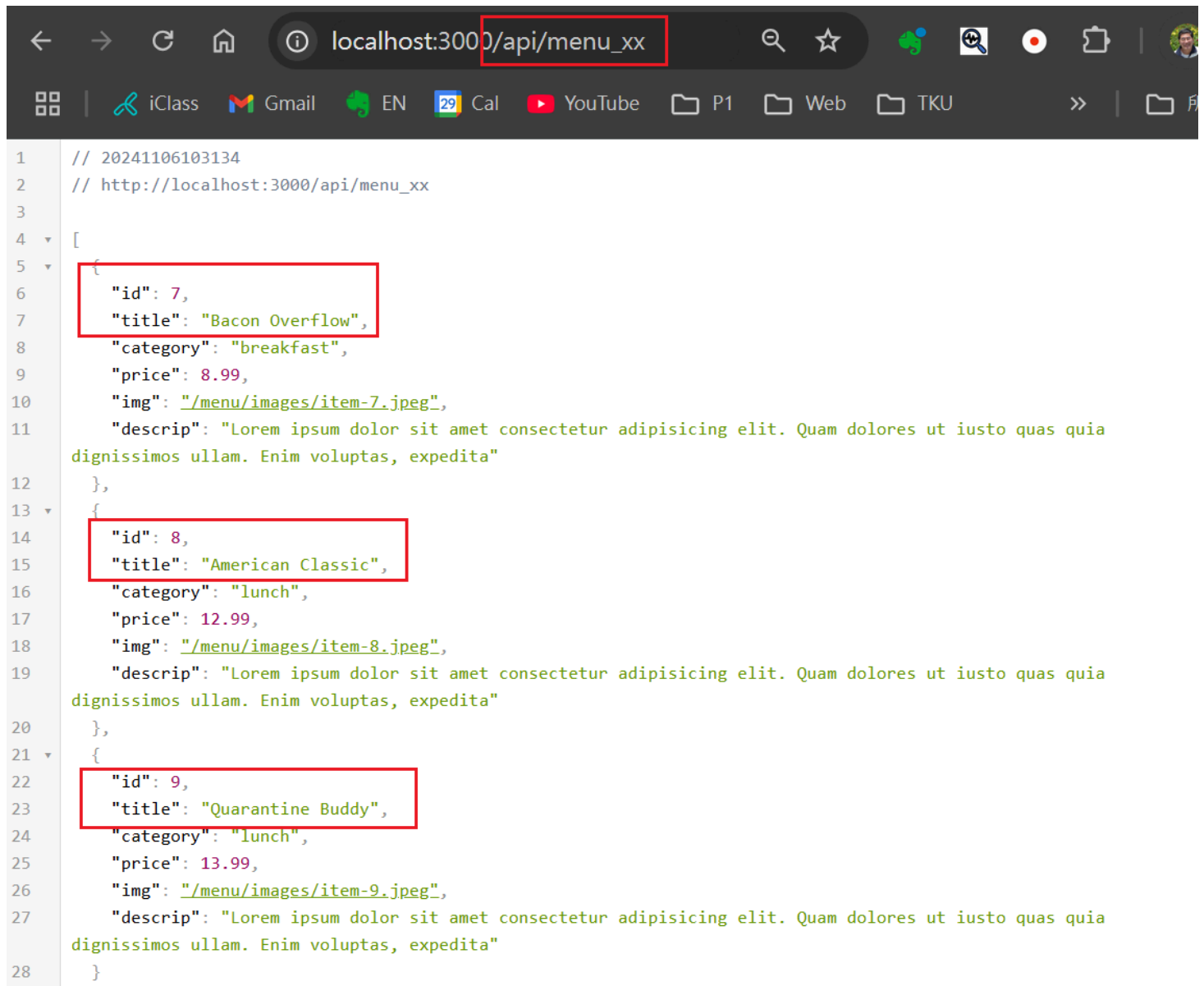


=> 顯示 database.js 設定，及顯示 3 筆資料



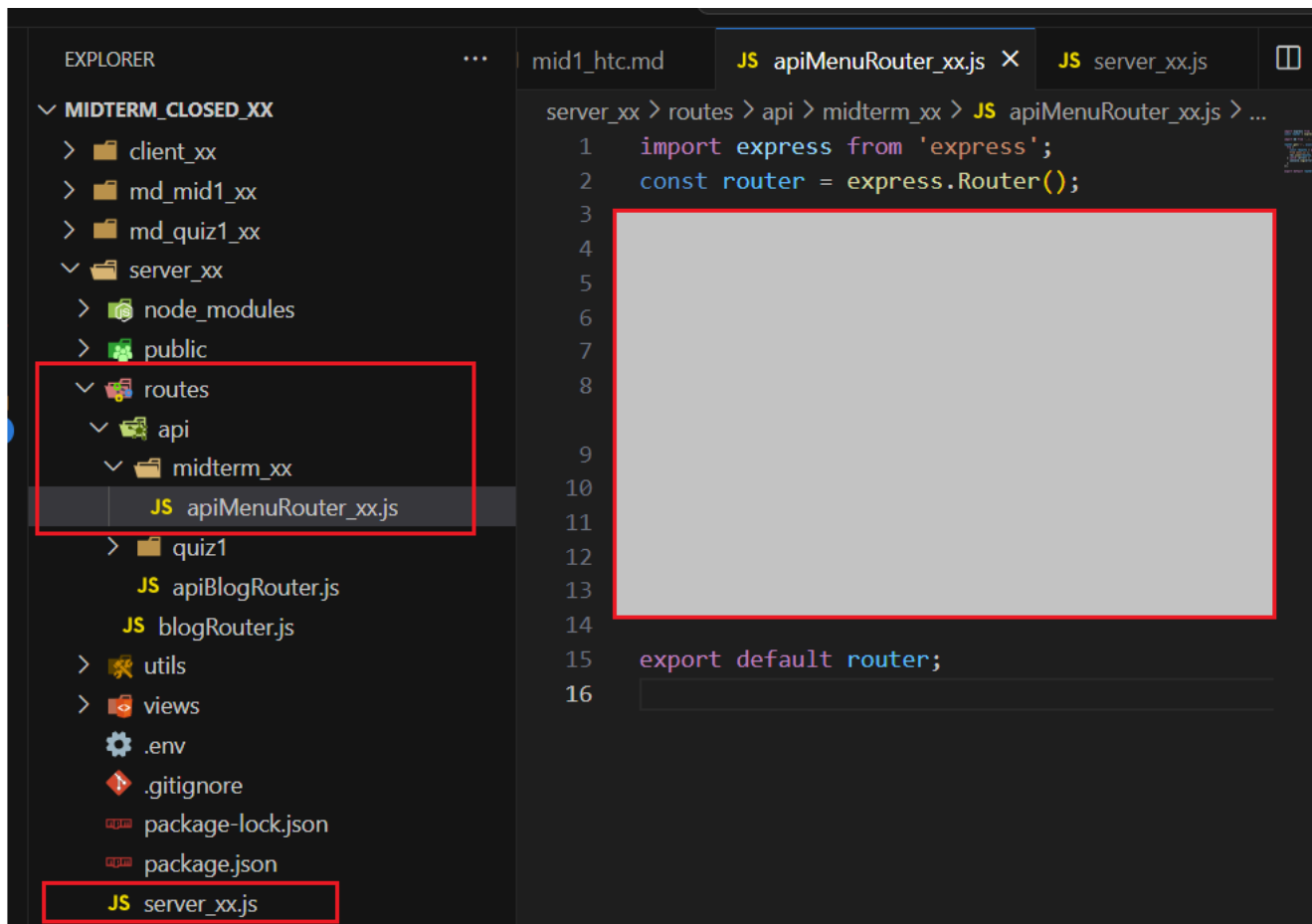
(10%) P2: 透過路由 /api/menu\_xx，取得 menu\_xx 資料表內的 3 筆資料，並顯示在 Chrome，處理的程式 apiMenuRouter\_xx.js 放在 routes/api/midterm\_xx 目錄下

=> Chrome 顯示



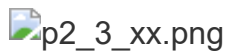
```
1 // 20241106103134
2 // http://localhost:3000/api/menu_xx
3
4 [
5   {
6     "id": 7,
7     "title": "Bacon Overflow",
8     "category": "breakfast",
9     "price": 8.99,
10    "img": "/menu/images/item-7.jpeg",
11    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
12  },
13  {
14    "id": 8,
15    "title": "American Classic",
16    "category": "lunch",
17    "price": 12.99,
18    "img": "/menu/images/item-8.jpeg",
19    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
20  },
21  {
22    "id": 9,
23    "title": "Quarantine Buddy",
24    "category": "lunch",
25    "price": 13.99,
26    "img": "/menu/images/item-9.jpeg",
27    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
28  }
29 ]
```

=> 處理程式 **apiMenuRouter\_xx.js** 相關 code

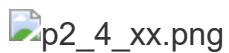


## Your Answer

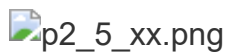
=> Chrome 顯示



=> 處理程式 **server\_xx.js** 相關 code



=> 處理程式 **apiMenuRouter\_xx.js** 相關 code



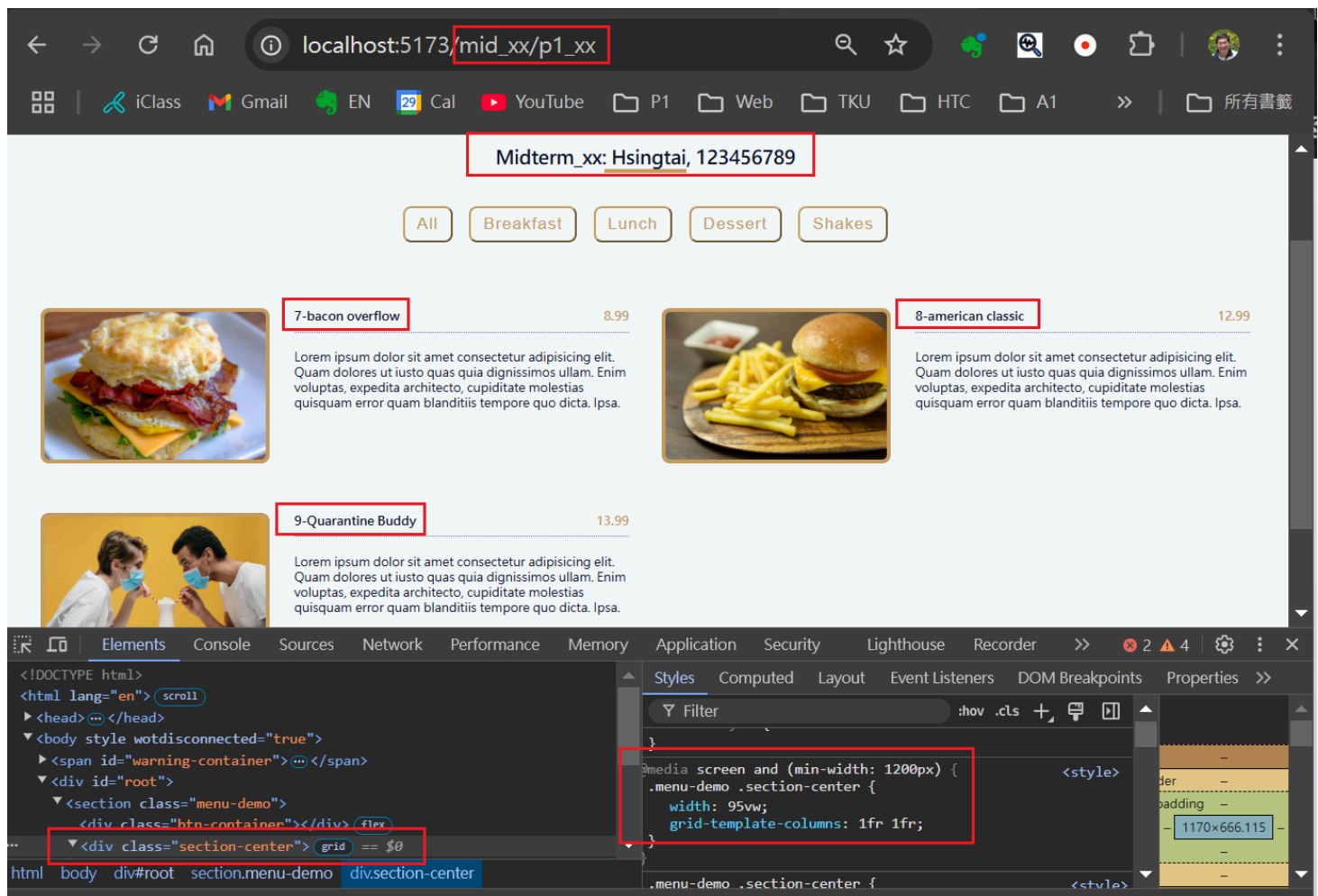
## Part 2 (25%) -- client 前端實作

### (10%) P3: 從老師給的 menu theme 中取得 Part 1 所做的 3 筆資料，以 Static 方式來實作

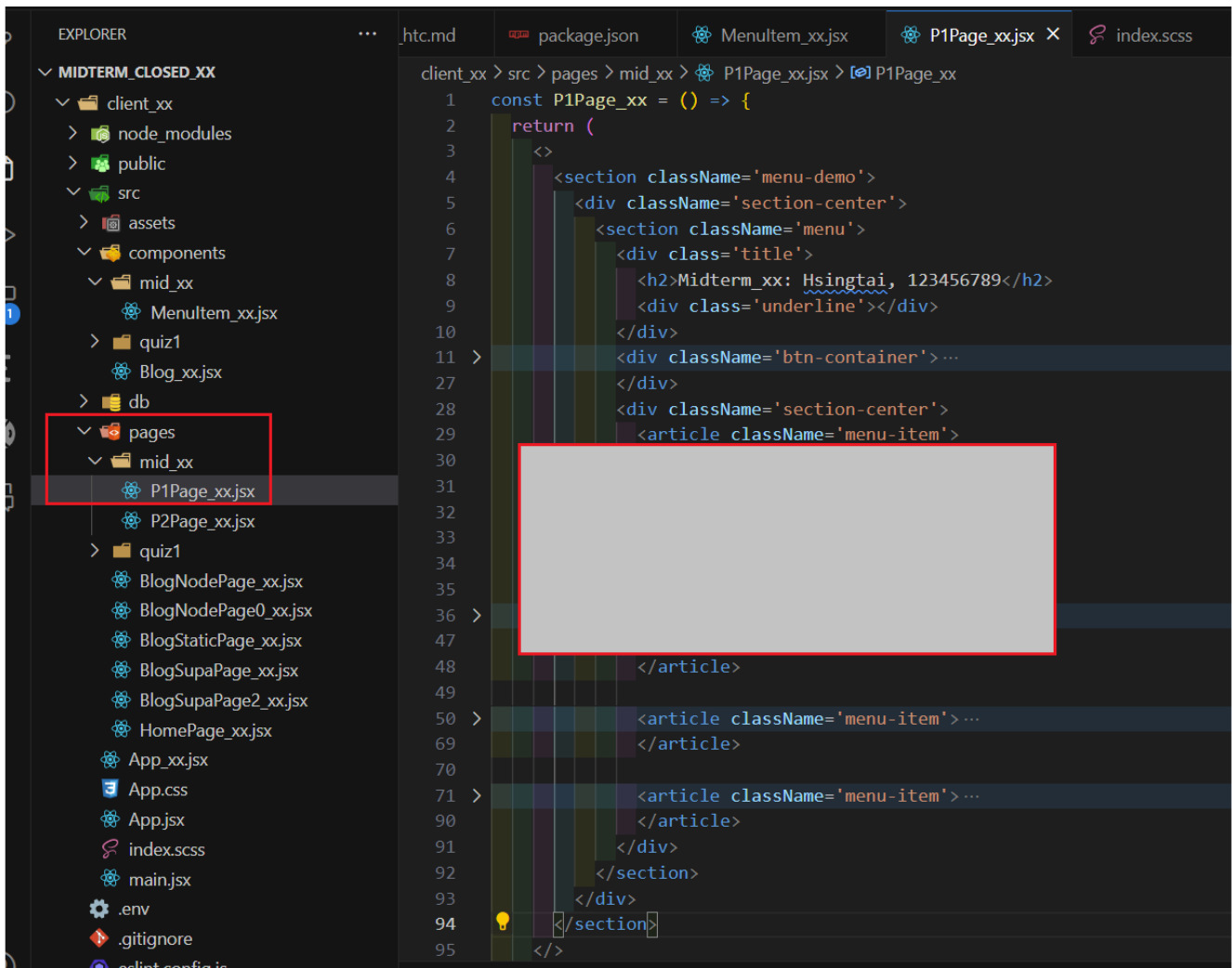
透過路由 /mid\_xx/p1\_xx 所指定的 pages/mid\_xx/P1Page\_xx.jsx 來顯示，每一個產品名稱前要加入 id，如下圖。

圖片已經放入 /public/mid\_xx 中，請注意圖片檔名為 jpeg

=> Chrome 顯示

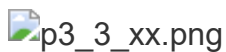


=> P1Page\_xx.jsx 重點截圖，最上方請顯示姓名及學號，展開第一個產品內容，其餘產品內容內縮

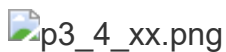


## Your Answer

=> Chrome 顯示如上圖



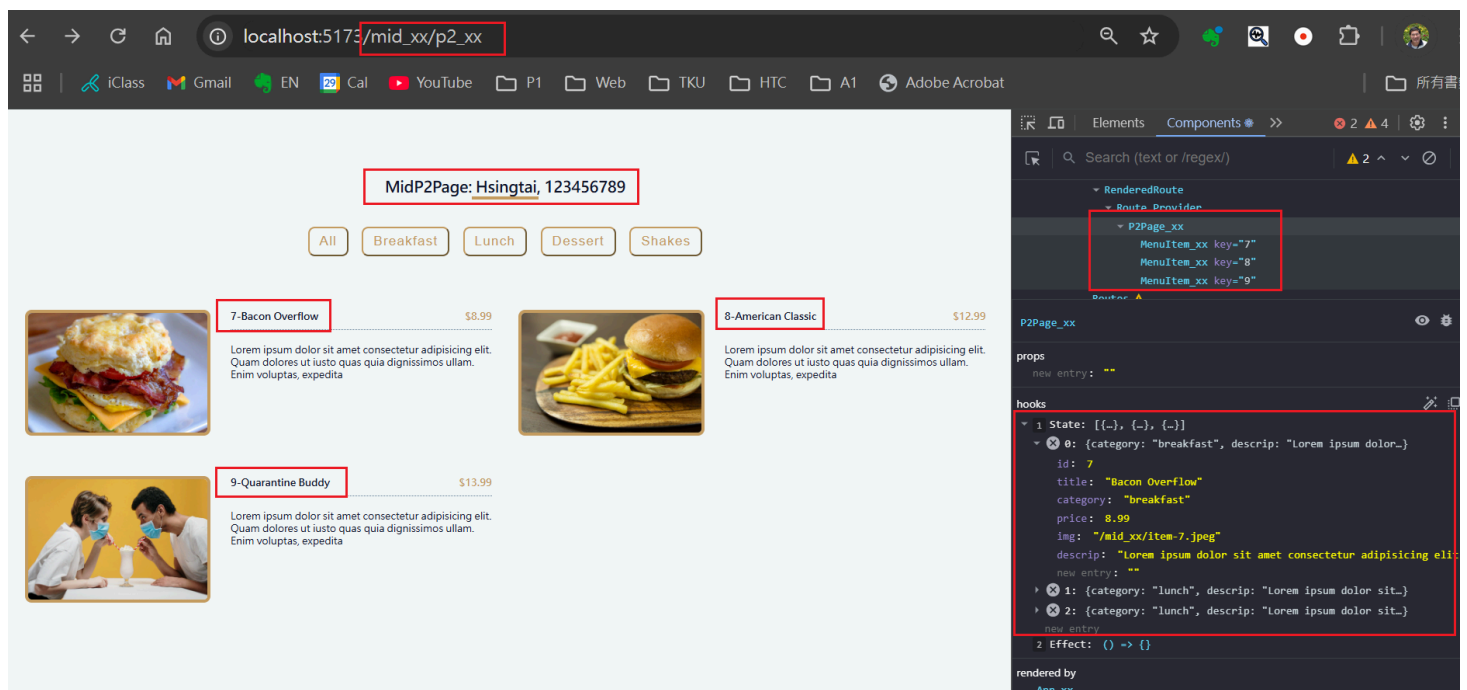
=> P1Page\_xx.jsx 重點截圖，最上方請顯示姓名及學號，展開第一個產品內容，其餘產品內容內縮



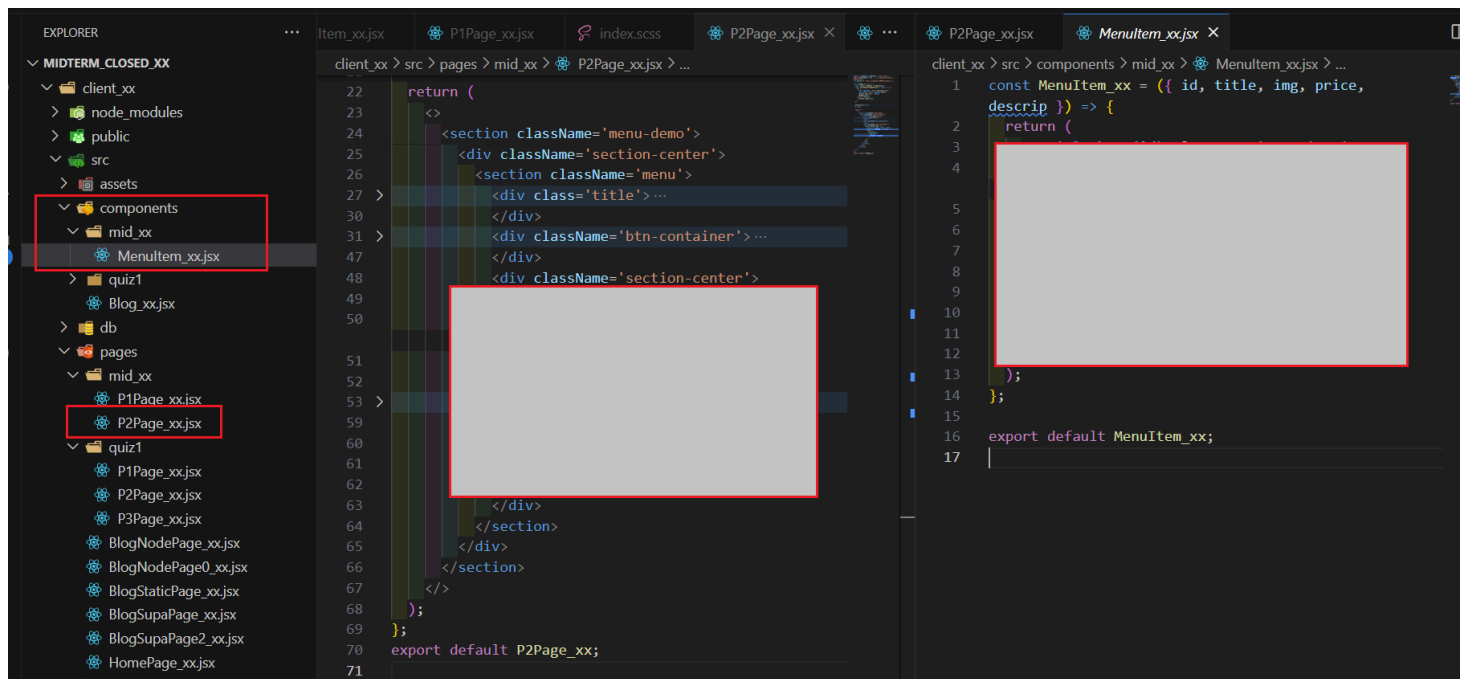
## (15%) P4: 從 server 取得 menu\_xx json 資料，並套用在前端

透過路由 /mid\_xx/p2\_xx 所指定的 pages/mid\_xx/P2Page\_xx.jsx 來顯示，每一個產品必須透過 <MenuItem\_xx /> 元件來顯示，產品名稱前要加入 id，如下圖

## => Chrome 顯示

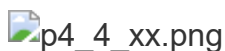


## => 顯示 P2Page\_xx.jsx & MenuItem\_xx.jsx 程式重點

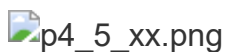


## Your Answer

### => Chrome 顯示如上




### => 要顯示是如何從 Node server 抓取資料之程式重點





=> 要顯示 **P2Page\_xx.jsx & MenuItem\_xx.jsx** 程式重點

p4\_6\_xx.png