

[Github URL](#)

[My Vercel URL](#)

## S1. 前端 theme (html, css)

The screenshot shows a web application interface with a header bar containing tabs like "Blog Demo" and "tku\_211417075 | Supabase". Below the header, there's a search bar and a sidebar with various icons. The main content area displays two sections: "Food" and "Drinks".

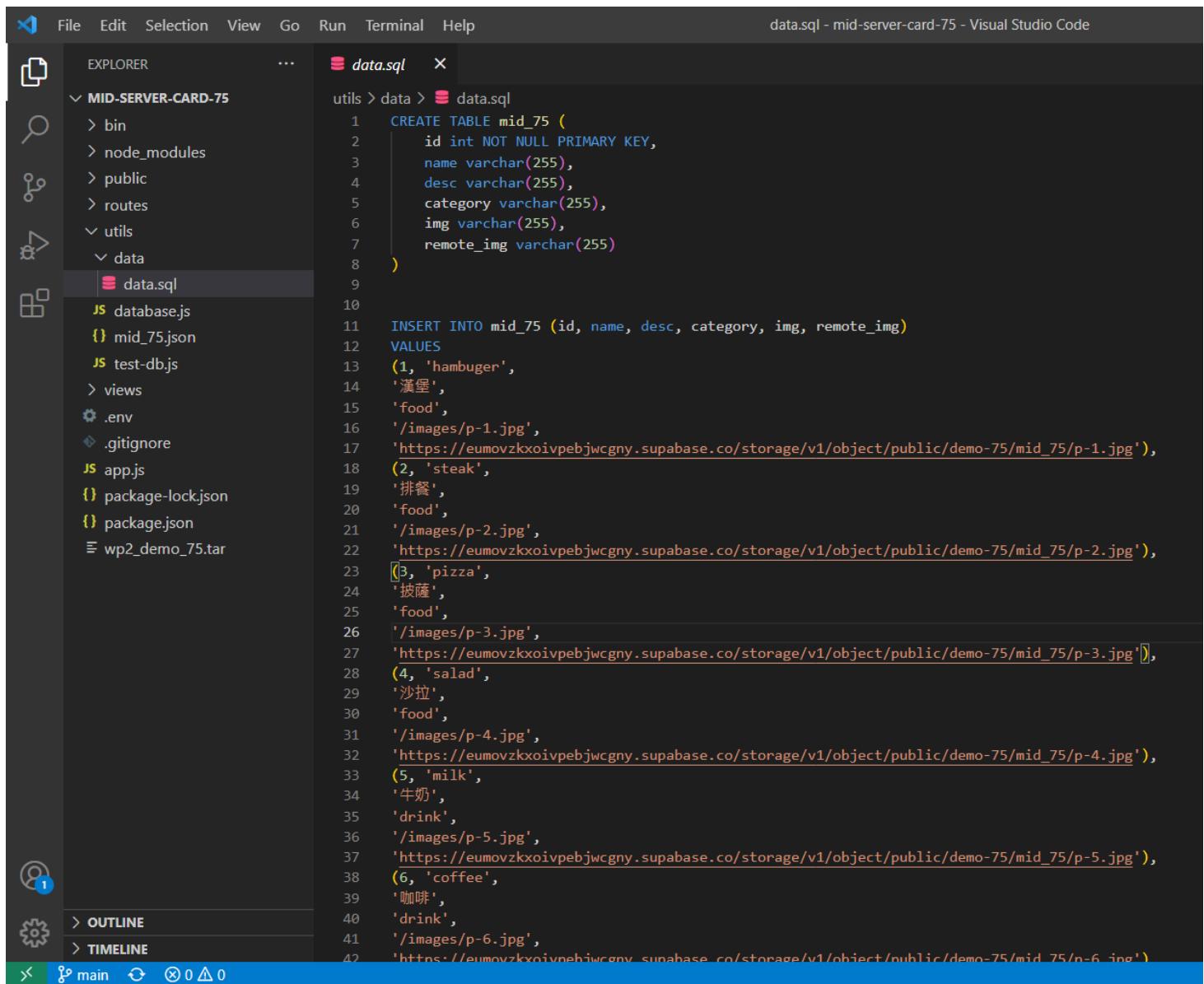
**Food**

- FOOD Hamburger 汉堡 [更多資訊](#)
- FOOD Steak 牛排 [更多資訊](#)
- FOOD Pizza 披萨 [更多資訊](#)
- FOOD Salad 沙拉 [更多資訊](#)

**Drinks**

- DRINK Milk 牛奶 [更多資訊](#)
- DRINK Coffee 咖啡 [更多資訊](#)
- DRINK Cola 可乐 [更多資訊](#)
- DRINK Juice 果汁 [更多資訊](#)

## S2. 提供相關的SQL指令，顯示如何將theme 內data放入資料庫中



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** data.sql - mid-server-card-75 - Visual Studio Code.
- Explorer:** Shows a tree view of the project structure:
  - MID-SERVER-CARD-75
    - bin
    - node\_modules
    - public
    - routes
    - utils
      - data
        - data.sql
  - JS database.js
  - {} mid\_75.json
  - JS test-db.js
  - > views
  - .env
  - .gitignore
  - JS app.js
  - {} package-lock.json
  - {} package.json
  - ≡ wp2\_demo\_75.tar
- Search:** Shows 1 result.
- Outline:** Shows > OUTLINE.
- Timeline:** Shows > TIMELINE.
- Status Bar:** Shows main, a refresh icon, and a terminal icon.

The content of the 'data.sql' file is as follows:

```
utils > data > data.sql
1 CREATE TABLE mid_75 (
2     id int NOT NULL PRIMARY KEY,
3     name varchar(255),
4     desc varchar(255),
5     category varchar(255),
6     img varchar(255),
7     remote_img varchar(255)
8 )
9
10
11 INSERT INTO mid_75 (id, name, desc, category, img, remote_img)
12 VALUES
13 (1, 'hamburger',
14 '漢堡',
15 'food',
16 '/images/p-1.jpg',
17 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-1.jpg'),
18 (2, 'steak',
19 '排餐',
20 'food',
21 '/images/p-2.jpg',
22 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-2.jpg'),
23 (3, 'pizza',
24 '披薩',
25 'food',
26 '/images/p-3.jpg',
27 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-3.jpg'),
28 (4, 'salad',
29 '沙拉',
30 'food',
31 '/images/p-4.jpg',
32 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-4.jpg'),
33 (5, 'milk',
34 '牛奶',
35 'drink',
36 '/images/p-5.jpg',
37 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-5.jpg'),
38 (6, 'coffee',
39 '咖啡',
40 'drink',
41 '/images/p-6.jpg',
42 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-6.jpg')
```

pgAdmin 4

File Object Tools Help

postgres/postgres@supabase-tku-211417075

No limit

Query History

```

1 CREATE TABLE mid_75 (
2     id int NOT NULL PRIMARY KEY,
3     name varchar(255),
4     desc varchar(255),
5     category varchar(255),
6     img varchar (255),
7     remote_img varchar(255)
8 )
9
10
11 INSERT INTO mid_75 (id, name, desc, category, img, remote_img)
12 VALUES
13 (1, 'hamburger',
14 '漢堡',
15 'food',
16 '/images/p-1.jpg',
17 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-1.jpg'),
18 (2, 'steak',
19 '排餐',
20 'food',
21 '/images/p-2.jpg',
22 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-2.jpg'),
23 (3, 'pizza',
24 '披薩',
25 'food',
26 '/images/p-3.jpg',
27 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-3.jpg'),
28 (4, 'salad',
29 '沙拉',
30 'food',
31 '/images/p-4.jpg',
32 'https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-4.jpg'),
Total rows: 0 of 0

```

Data Output    Messages    Notifications

No data output. Execute a query to get output.

Ln 29, Col 6

pgAdmin 4

File Object Tools Help

Browser

	id	name	category	desc	img	remote_img
1	1	hamburger	food	漢堡	/images/p-1.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-1
2	2	steak	food	排餐	/images/p-2.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-2
3	3	pizza	food	披薩	/images/p-3.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-3
4	4	salad	food	沙拉	/images/p-4.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-4
5	5	milk	drink	牛奶	/images/p-5.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-5
6	6	coffee	drink	咖啡	/images/p-6.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-6
7	7	cola	drink	可樂	/images/p-7.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-7
8	8	juice	drink	果汁	/images/p-8.jpg	https://eumovzxoivpebjwcgny.supabase.co/storage/v1/object/public/demo-75/mid_75/p-8

Total rows: 8 of 8    Query complete 00:00:01.121

Ln 1, Col 1

## S3. node.js server

The screenshot shows the Supabase Table editor interface. On the left, there's a sidebar with 'Tables' and three entries: 'card\_75', 'cart\_75', and 'mid\_75'. The main area displays a table with columns: id, name, category, desc, img, and remote\_img. The data consists of eight rows:

	id	name	category	desc	img	remote_img
1	1	hamburger	food	漢堡	/images/p-1.jpg	https://eumovzkoivpebjwcn.y.supabase.
2	2	steak	food	排餐	/images/p-2.jpg	https://eumovzkoivpebjwcn.y.supabase.
3	3	pizza	food	披薩	/images/p-3.jpg	https://eumovzkoivpebjwcn.y.supabase.
4	4	salad	food	沙拉	/images/p-4.jpg	https://eumovzkoivpebjwcn.y.supabase.
5	5	milk	drink	牛奶	/images/p-5.jpg	https://eumovzkoivpebjwcn.y.supabase.
6	6	coffee	drink	咖啡	/images/p-6.jpg	https://eumovzkoivpebjwcn.y.supabase.
7	7	cola	drink	可樂	/images/p-7.jpg	https://eumovzkoivpebjwcn.y.supabase.
8	8	juice	drink	果汁	/images/p-8.jpg	https://eumovzkoivpebjwcn.y.supabase.

## S4. 提供 api 路由，可以透過 api 路由取得 json data

The screenshot shows a browser window with the URL `localhost:5000/api/mid2_75`. The page displays a JSON array representing the food items from the previous table:

```
// 20230415220455
// http://localhost:5000/api/mid2_75
[
  {
    "id": "1",
    "name": "hamburger",
    "category": "food",
    "desc": "漢堡",
    "img": "/images/p-1.jpg",
    "remote_img": "https://eumovzkoivpebjwcn.y.supabase.co/storage/v1/object/public/demo-75/mid_75/p-1.jpg"
  },
  {
    "id": "2",
    "name": "steak",
    "category": "food",
    "desc": "排餐",
    "img": "/images/p-2.jpg",
    "remote_img": "https://eumovzkoivpebjwcn.y.supabase.co/storage/v1/object/public/demo-75/mid_75/p-2.jpg"
  },
  {
    "id": "3",
    "name": "pizza",
    "category": "food",
    "desc": "披薩",
    "img": "/images/p-3.jpg",
    "remote_img": "https://eumovzkoivpebjwcn.y.supabase.co/storage/v1/object/public/demo-75/mid_75/p-3.jpg"
  },
  {
    "id": "4",
    "name": "salad",
    "category": "food",
    "desc": "沙拉",
    "img": "/images/p-4.jpg",
    "remote_img": "https://eumovzkoivpebjwcn.y.supabase.co/storage/v1/object/public/demo-75/mid_75/p-4.jpg"
  },
  {
    "id": "5",
    "name": "milk",
    "category": "drink",
    "desc": "牛奶",
    "img": "/images/p-5.jpg",
    "remote_img": "https://eumovzkoivpebjwcn.y.supabase.co/storage/v1/object/public/demo-75/mid_75/p-5.jpg"
  }
]
```

# F1. 前端靜態頁面及local端資料(加入選單)

Screenshot of a web browser showing a static page with dynamic data fetched from a local JSON file.

The browser address bar shows: `localhost:3000/mid_75`. A red box highlights this URL.

The page title is "Fetch Midtren From Static Json -Ray 211417075".

The navigation menu includes: HOME, BLOGS DEMO +, MEGA MENU +, BLOG, and CONTACT.

**Food**

-  **FOOD**  
Hambuger  
漢堡  
[更多資訊](#)
-  **FOOD**  
Steak  
排餐  
[更多資訊](#)
-  **FOOD**  
Pizza  
披薩  
[更多資訊](#)
-  **FOOD**  
Salad  
沙拉  
[更多資訊](#)

**Drink**

-  **DRINK**  
Milk  
牛奶  
[更多資訊](#)
-  **DRINK**  
Coffee  
咖啡  
[更多資訊](#)
-  **DRINK**  
Cola  
可樂  
[更多資訊](#)
-  **DRINK**  
Juice  
果汁  
[更多資訊](#)

JS App\_75.js 5 JS BlogsLocalJsonPage\_75.js 9+ JS BlogsStaticPage\_75.js 9+ X JS BlogsNodeServerPage\_75.js 9+ JS BlogsSupabasePage\_75.js 9+ JS NavbarResp\_75.js

```

src > pages > JS BlogsStaticPage_75.js > [x] BlogsStaticPage_75
6   const [id, setId] = useState('211417075');
7   const { pName, pId, blogs } = useDemoContext_75();
8   return [
9     <Wrapper>
10       <section className="blogs">
11         <div className="section-title">
12           <h2>Fetch Midtren From Static Json--{pName} {pId}</h2>
13         </div>
14         <div className="blogs-center">
15           <h1>Food</h1>
16         </div>
17         <br />
18         <div className="blogs-center2">
19           <article className="blog">
20             
25             <div className="blog-content">
26               <span>food<i className="fa-solid fa-mug-saucer"></i></span>
27               <h3>hamburger</h3>
28               <p>漢堡</p>
29               <a href="#">更多資訊</a>
30             </div>
31           </article>
32           <article className="blog">
33             

```

PROBLEMS 61 OUTPUT DEBUG CONSOLE TERMINAL

?[90m at Module.\_load (node:internal/modules/cjs/loader:922:12)?[39m

Ray@MSI MINGW64 /c/TkuSchool/React/mid-client-card-75/src/context (main)

localhost:3001/local\_75

Fetch Midtren From Local Json -- R

HOME BLOGS DEMO + MEGA MENU + BLOG CONTACT

**Food**

-   
FOOD  
**Hambuger**  
漢堡  
[更多資訊](#)
-   
FOOD  
**Steak**  
牛排  
[更多資訊](#)
-   
FOOD  
**Pizza**  
披薩  
[更多資訊](#)
-   
FOOD  
**Salad**  
沙拉  
[更多資訊](#)

**Drink**

-   
DRINK  
**Milk**  
牛奶  
[更多資訊](#)
-   
DRINK  
**Coffee**  
咖啡  
[更多資訊](#)
-   
DRINK  
**Cola**  
可樂  
[更多資訊](#)
-   
DRINK  
**Juice**  
果汁  
[更多資訊](#)

localhost:3000/local\_75

DevTools is now available in Chinese!

Always match Chrome's language Switch DevTools to Chinese

Don't show again

Console

1 Issue: 1

blogs BlogsLocalJsonPage\_75.js:10

category: "food"  
desc: "漢堡"  
id: 1  
img: "/images/p-1.jpg"  
name: "hambuger"  
remote\_img: "https://eumovzxoivpebjwcgny.supabase

[[Prototype]]: Object

► 1: {id: 2, name: 'steak', category: 'food', desc: '#'}  
► 2: {id: 3, name: 'pizza', category: 'food', desc: '#'}  
► 3: {id: 4, name: 'salad', category: 'food', desc: '#'}  
► 4: {id: 5, name: 'milk', category: 'drink', desc: '#'}  
► 5: {id: 6, name: 'coffee', category: 'drink', desc: '#'}  
► 6: {id: 7, name: 'cola', category: 'drink', desc: '#'}  
► 7: {id: 8, name: 'juice', category: 'drink', desc: '#'}  
length: 8

[[Prototype]]: Array(0)

pName Ray BlogsLocalJsonPage\_75.js:11  
blog\_data BlogsLocalJsonPage\_75.js:13  
blogs react\_devtools\_backend.js:2709  
pName Ray react\_devtools\_backend.js:2709  
Console What's New

Highlights from the Chrome 112 update

JS App\_75.js 5

JS BlogsLocalJsonPage\_75.js 9+ X

JS BlogsStaticPage\_75.js 9+

JS BlogsNodeServerPage\_75.js 9+

JS BlogsSupabase

src &gt; pages &gt; JS BlogsLocalJsonPage\_75.js &gt; [o] BlogsLocalJsonPage\_75

```
12  const [data, setData] = useState(blogsData);
13  console.log('blog data', data);
14  return (
15      <Wrapper>
16          <section className="blogs">
17              <div className="section-title">
18                  <h2>Fetch Midtren From Local Json --{pName} {pId}</h2>
19              </div>
20              <div className="blogs-center">
21                  <h1>Food</h1>
22                  <br />
23                  <div className="blogs-center2">
24                      {blogs.map(item => {
25                          const { id, img, remoteImg, category, name, desc } = item;
26                          if (category === 'food') {
27                              return (
28                                  <article key={id} className="blog">
29                                      <img
30                                          src={img}
31                                          alt={name}
32                                          className="img blog-img"
33                                      />
34                                      <div className="blog-content">
35                                          <span>{category}<i className="fa-solid fa-mug-saucer"></i></span>
36                                          <h3>{name}</h3>
37                                          <p>{desc}</p>
38                                          <a href="#">更多資訊</a>
39                                      </div>
40                                  </article>
41                              )
42                          )
43                      })
44                  </div>
45                  <div className="blogs-center">
46                      <h1>Drink</h1>
47                  </div>
48                  <br />
49                  <div className="blogs-center2">
50                      {blogs.map(item => {
51                          const { id, img, remoteImg, category, name, desc } = item;
52                          if (category === 'drink') {
53                              return (
54                                  <article key={id} className="blog">
55                                      <img
56                                          src={img}
```

&gt; supabase

Aa al

Ln 22, Col 23 Spaces:

## F2. 可以透過 server api 取得json資料，放入 React components(加入選單)

The screenshot shows a web application interface. At the top, there's a navigation bar with links like HOME, BLOGS DEMO +, MEGA MENU +, BLOG, and CONTACT. Below this is a header with the text "Fetch Midtern From Node" and "Ray 211417075". The main content area is divided into two sections: Food and Drink. The Food section contains four items: Hamburger, Steak, Pizza, and Salad. The Drink section contains four items: Milk, Coffee, Cola, and Juice. Each item has a small image, a category name (FOOD or DRINK), a desc, and a "更多資訊" link. On the right side, the DevTools sidebar is open, showing the component tree and a detailed view of the "Food" category data under the "Context.Provider" component.

```
category: "food"
desc: "漢堡"
id: 1
img: "/images/p-1.jpg"
name: "hamburger"
remote_img: "https://eumovzkoivpebjwcgny.supabase.co/storage/v1/o/ot.../p-1.jpg"
new_entry: ""
x 1: {category: "food", desc: "排餐", id: 2, img: "/images..."}
x 2: {category: "food", desc: "披薩", id: 3, img: "/images..."}
x 3: {category: "food", desc: "沙拉", id: 4, img: "/images..."}
x 4: {category: "drink", desc: "牛奶", id: 5, img: "/image..."}
x 5: {category: "drink", desc: "咖啡", id: 6, img: "/image..."}
x 6: {category: "drink", desc: "可樂", id: 7, img: "/image..."}
x 7: {category: "drink", desc: "果汁", id: 8, img: "/image..."}
```

## F3. 可以直接透過 Supabase API 取得React components 所要的 json資料(加入選單)

This screenshot is similar to the previous one, showing the same food and drink menu. The URL has changed to localhost:3000/supabase\_75. The DevTools sidebar shows the same component tree and expanded "Food" category data as in the previous screenshot.

```
category: "food"
desc: "漢堡"
id: 1
img: "/images/p-1.jpg"
name: "hamburger"
remote_img: "https://eumovzkoivpebjwcgny.supabase.co/storage/v1/o/ot.../p-1.jpg"
new_entry: ""
x 1: {category: "food", desc: "排餐", id: 2, img: "/images..."}
x 2: {category: "food", desc: "披薩", id: 3, img: "/images..."}
x 3: {category: "food", desc: "沙拉", id: 4, img: "/images..."}
x 4: {category: "drink", desc: "牛奶", id: 5, img: "/image..."}
x 5: {category: "drink", desc: "咖啡", id: 6, img: "/image..."}
x 6: {category: "drink", desc: "可樂", id: 7, img: "/image..."}
x 7: {category: "drink", desc: "果汁", id: 8, img: "/image..."}
```

## F4. 請使用 Context API 及 useReducer 來儲存資料

The screenshot shows two code editors in Visual Studio Code. The left editor contains `DemoContext_75.js` and the right editor contains `DemoReducer_75.js`. Both files are highlighted with a pink border.

**DemoContext\_75.js:**

```
File Edit Selection View Go Run Terminal Help
JS DemoContext_75.js 3
src > context > JS DemoContext_75.js > DemoProvider_75 > fetchBlogDataFromSupabase
4
5 import { supabase } from '../db/clientSupabase'
6
7 const initialState = {
8   pName: 'Ray',
9   pid: '211417075',
10  blogs: []
11 }
12
13 const DemoContext_75 = React.createContext();
14
15 const DemoProvider_75 = ({children}) => {
16   const [state, dispatch] = useReducer(DemoReducer_75, initialState);
17   const fetchBlogDataFromSupabase = async () => {
18     try {
19       let { data, error } = await supabase.from('mid_75').select('*')
20
21       console.log('data123123', data);
22       dispatch({type: 'GET_BLOGS_SUPABASE_SUCCESS', payload: data});
23       // setdata(data);
24     } catch (error) {
25       console.log(error);
26     }
27   }
28
29
30   useEffect(() => {
31     fetchBlogDataFromSupabase();
32   }, []);
33
34   return (
35     <DemoContext_75.Provider value={...state}>
36       {children}
37     </DemoContext_75.Provider>
38   )
39 }
40
```

**DemoReducer\_75.js:**

```
JS DemoReducer_75.js X
src > context > JS DemoReducer_75.js > ...
1 const DemoReducer_75 = (state,action) => {
2   if(action.type === 'GET_BLOGS_SUPABASE_SUCCESS') {
3     return {...state, blogs: action.payload}
4   }
5
6 }
7
8 export default DemoReducer_75;
```

Bottom status bar: Ln 20, Col 59 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live