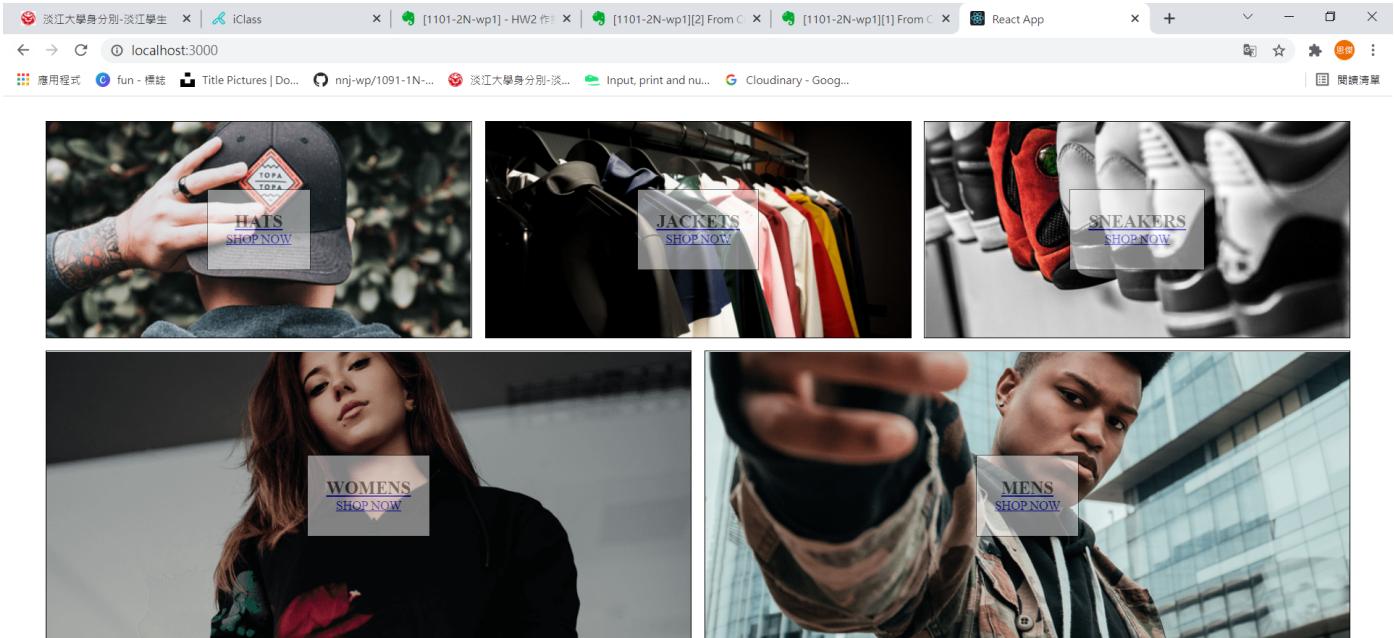


1. (local) 請以 App_xx.js 顯示下圖片，繳交

. Chrome 截圖

. 目錄結構及 App_xx 部分內容截圖 (以學號後 2 碼來展開 category 內容，其他收起)

. App_xx.js code (程式碼，非圖片)



The screenshot shows the Visual Studio Code interface. The left sidebar displays the project structure for 'REACT_CROWN_51', including files like index.html, jackets.html, mens.html, overview.html, sneakers.html, w4_markdown..., womens.html, and node_modules. The main editor pane shows the code for 'App_51.js'. The code defines a function 'App()' that returns a component structure. It includes imports for './App_51.scss' and './App_51.js'. The component structure uses 'menu-item' divs with background images and links to other pages like '/jackets.html' and '/womens.html'. The bottom status bar indicates 'Compiled successfully!'. The bottom navigation bar includes icons for bash, Live Share, Watch Sass, Go Live, Spell, Prettier, and a terminal.

```
function App() {
  return (
    <div className="homepage">
      <div className="directory-menu">
        <div className="menu-item">...</div>
        <div className="menu-item">
          
          <a href="/jackets.html" className="content">
            <h1 className="title">JACKETS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
        <div className="menu-item">
          
          <a href="/sneakers.html" className="content">
            <h1 className="title">SNEAKERS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
        <div className="large menu-item">
          
          <a href="/womens.html" className="content">
            <h1 className="title">WOMENS</h1>
          </a>
        </div>
      </div>
    </div>
  );
}
```

```
import './App_51.js';

function App() {
  return (
    <div className="homepage">
      <div className="directory-menu">
        <div className="menu-item">
          
          <a href="./hats.html" className="content">
            <h1 className="title">HATS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
        <div className="menu-item">
          
          <a href="./jackets.html" className="content">
            <h1 className="title">JACKETS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
        <div className="menu-item">
          
          <a href="./sneakers.html" className="content">
            <h1 className="title">SNEAKERS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
        <div className="large menu-item">
          
          <a href="./womens.html" className="content">
            <h1 className="title">WOMENS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
        <div className="large menu-item">
          
          <a href="./mens.html" className="content">
            <h1 className="title">MENS</h1>
            <span className="subtitle">SHOP NOW</span>
          </a>
        </div>
      </div>
    </div>
  );
}

export default App;
```

2. (local) 將 homepage 分成如下圖的 components · Homepage_xx.js, Directory_xx.js,

MenuItem_xx.js · 並且能夠正常顯示如第 1 題。繳交

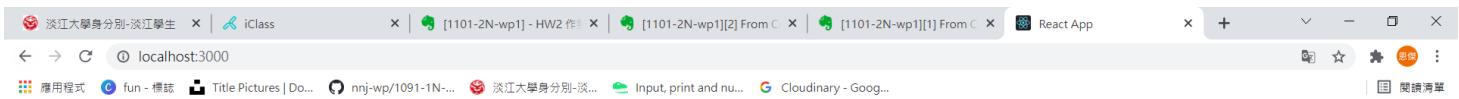
. Chrome 截圖

. 目錄結構 · 要含所有用到的 components

. Homepage_xx.js code (程式碼 · 非圖片)

. Directory_xx.js code

. MenuItem_xx.js code



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

- File Explorer:** Shows the project structure under "REACT_CROWN_51".
- Code Editor:** Displays the content of `Directory_51.js`. The code defines a `Directory_51` component that returns a `div` containing four `MenuItem_51` components with names "HATS", "JACKETS", "SNEAKERS", and "WOMENS". Each item has a `remoteUrl` pointing to an image URL.
- Terminal:** Shows the command `./src/App_51.js` being run.
- Bottom Bar:** Includes icons for file operations, search, and various extensions like Live Share, Watch Sass, Go Live, Spell, and Prettier.

```
Homepage_51.js =>
import React from 'react';

import Directory_51 from '../components/Directory_51';

const Homepage_51 = () => {
  return (
    <div className="homepage">
      <Directory_51 />
    </div>
  );
};

export default Homepage_51;
```

```
Directory_51.js =>
import React from 'react';
import './Directory_51.scss';

import MenuItem_51 from './MenuItem_51';

const Directory_51 = () => {
  return (
    <div>
      <div className="directory-menu">
        <MenuItem_51
          name="HATS"
          remoteUrl="https://i.ibb.co/cvpntL1/hats.png"
          size=" "
        />
        <MenuItem_51
          name="JACKETS"
          remoteUrl="https://i.ibb.co/px2tCc3/jackets.png"
          size=" "
        />
        <MenuItem_51
          name="SNEAKERS"
          remoteUrl="https://i.ibb.co/0jqHpnP/sneakers.png"
          size=" "
        />
        <MenuItem_51
          name="WOMENS"
          remoteUrl="https://i.ibb.co/GCCdy8t/womens.png"
          size="large"
        />
        <MenuItem_51
          name="MENS"
          remoteUrl="https://i.ibb.co/R70vBrQ/men.png"
          size="large"
        />
      </div>
    </div>
  );
};

export default Directory_51;
```

```

MenuItem =>
import React from 'react';
import './MenuItem_51.scss';

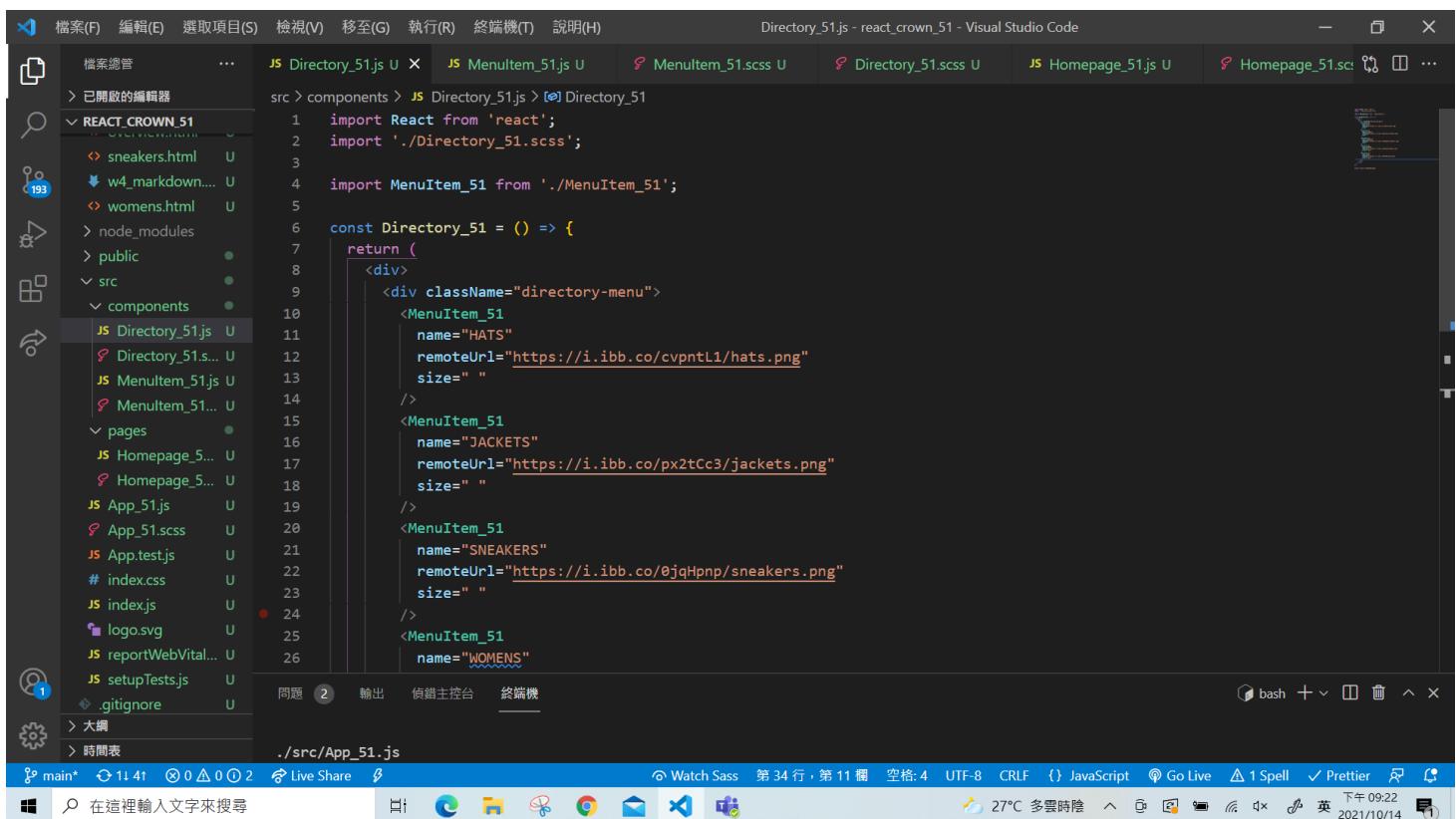
const MenuItem_51 = ({ name, remoteUrl, size }) => {
  return (
    <div className={`${size} menu-item`}>
      <img className="background-image" src={remoteUrl} alt="" />
      <a href="./hats.html" className="content">
        <h1 className="title">{name}</h1>
        <span className="subtitle">SHOP NOW</span>
      </a>
    </div>
  );
};

export default MenuItem_51;

```

3. (local) 將 menu items 五筆資料放入 JSON 陣列中，並能正常顯示如第 1 題。繳交

- . Chrome 截圖
- . JSON 陣列 (程式碼，非圖片)
- . Directory_xx.js code (程式碼，非圖片)



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

- File Explorer:** Shows the project structure under "REACT_CROWN_51". The "src" folder contains "components" which has "Directory_51.js". Other files like "sneakers.html", "womens.html", etc., are also listed.
- Code Editor:** Displays the content of "Directory_51.js". The code defines a "Directory_51" component that returns a list of five "MenuItem_51" components. Each item has a name, remote URL, and size.
- Bottom Status Bar:** Shows the current file is "main*", line 1141, and the terminal output "bash +". It also includes icons for live share, watch sass, go live, spell check, prettier, and other development tools.

```

const Directory_51 = () => {
  return (
    <div>
      <div className="directory-menu">
        <MenuItem_51
          name="HATS"
          remoteUrl="https://i.ibb.co/cvpntL1/hats.png"
          size=" "
        />
        <MenuItem_51
          name="JACKETS"
          remoteUrl="https://i.ibb.co/px2tCc3/jackets.png"
          size=" "
        />
        <MenuItem_51
          name="SNEAKERS"
          remoteUrl="https://i.ibb.co/0jqHppn/sneakers.png"
          size=" "
        />
        <MenuItem_51
          name="WOMENS"
        />
      </div>
    </div>
  );
};

```

```
import React, { useState } from 'react';
import './Directory_51.scss';
import items from './menu-items-data';
import MenuItem_51 from './MenuItem_51';

const Directory_51 = () => {
  const [menuItems, setMenuItems] = useState(items);
  console.log('menuItems', menuItems);
  return (
    <div>
      <div className="directory-menu">
        {menuItems.map((item) => {
          const { id, name, remoteUrl, size } = item;
          return (
            <MenuItem_51
              key={id}
              name={name}
              remoteUrl={remoteUrl}
              size={size}
            />
          );
        })}
      </div>
    </div>
  );
};

export default Directory_51;
```

```

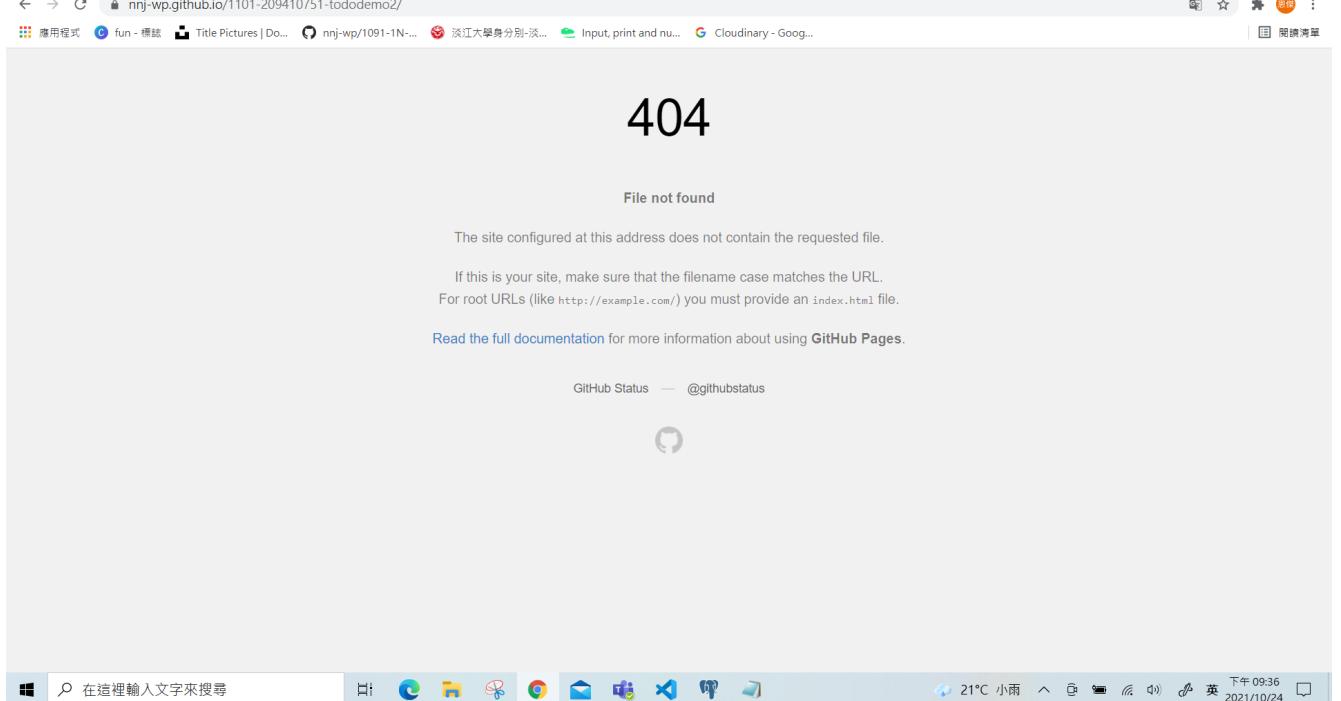
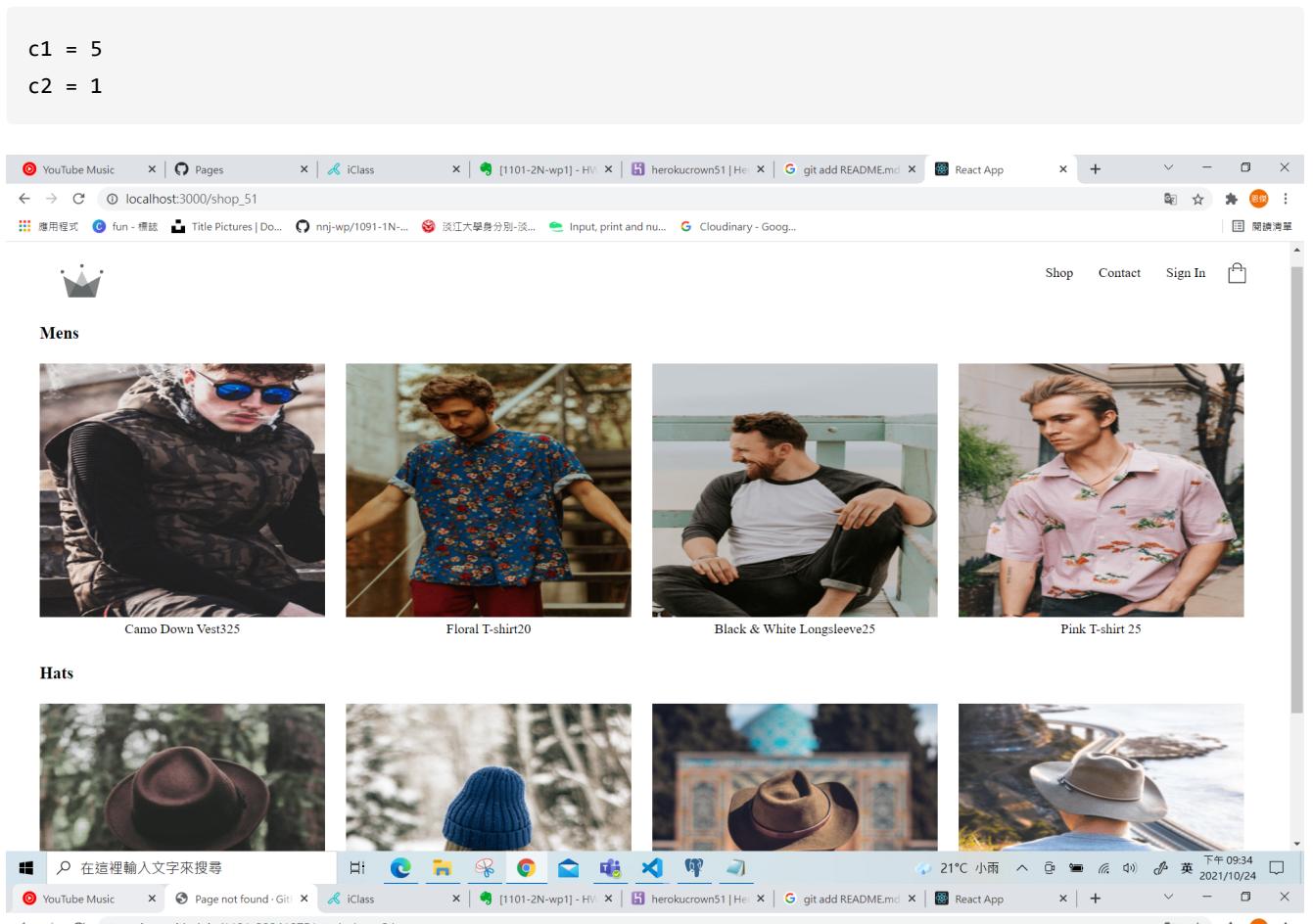
const items = [
  {
    id: 1,
    name: 'HATS',
    remoteUrl: 'https://i.ibb.co/cvpntL1/hats.png',
    size: ''
  },
  {
    id: 2,
    name: 'JACKETS',
    remoteUrl: 'https://i.ibb.co/px2tCc3/jackets.png',
    size: ''
  },
  {
    id: 3,
    name: 'SNEAKERS',
    remoteUrl: 'https://i.ibb.co/0jqHppnp/sneakers.png',
    size: ''
  },
  {
    id: 4,
    name: 'WOMENS',
    remoteUrl: 'https://i.ibb.co/GCCdy8t/womens.png',
    size: 'large'
  },
  {
    id: 5,
    name: 'MENS',
    remoteUrl: 'https://i.ibb.co/R70vBrQ/men.png',
    size: 'large'
  },
];
export default items;

```

- 4. (local, github) 實作路由 /shop_xx，可以顯示 overview.html 內兩種 category 的資訊 (c1, c2)。每一 category 要顯示 4 筆資料。c1, c2 由學號後兩碼來決定，顯示順序先 c1 再 c2，說明如下：
 1, 6 -- hats
 2, 7 -- jackets
 3, 8 -- sneakers
 4, 9 -- womens
 5, 0 -- mens
 如果 c1 = c2 兩個相同，那 c2 就取下一個。要繳交

- . 說明 c1, c2 那是坐那兩個
- . local 端 Chrome 執行結果截圖
- . Github page 執行結果截圖
- . 重要相關的原始程式碼

```
c1 = 5
c2 = 1
```



重要程式碼 => CollectionOverview

```
const CollectionOverview_51 = ({ title1, title2 }) => {
  const [collectionItems, setCollectionItems] = useState(items);
  console.log('collectionItems', collectionItems);
  const hatsItem = collectionItems.slice(0, 4);
  console.log('hatsItem', hatsItem);
  const mensItem = collectionItems.slice(4);
  console.log('mensItem', mensItem);
  return (
    <div className="collection-overview">
      <div className="collection-preview">
        <h2 className="title">{title1}</h2>
        <div className="preview">
          {mensItem.map((item) => {
            const { id, remoteUrl, name, price } = item;
            return (
              <CollectionItem_51
                key={id}
                remoteUrl={remoteUrl}
                name={name}
                price={price}
              />
            );
          })}
        </div>
      </div>
      <div className="collection-preview">
        <h2 className="title">{title2}</h2>
        <div className="preview">
          {hatsItem.map((item) => {
            const { id, remoteUrl, name, price } = item;
            return (
              <CollectionItem_51
                key={id}
                remoteUrl={remoteUrl}
                name={name}
                price={price}
              />
            );
          })}
        </div>
      </div>
    </div>
  );
};
```

重要程式碼 => CollectionItem

```
const CollectionPreview_51 = ({ remoteUrl, name, price }) => {
  return (
    <div className="collection-item">
      <img className="image" src={remoteUrl} />
      <div className="collection-footer">
        <span className="name">{name}</span>
        <span className="price">{price}</span>
      </div>
      <button className="custom-button">Add to Cart</button>
    </div>
  );
};
```

重要程式碼 => Collection-item-data

```
const items =[  
  {  
    id: 1,  
    remoteUrl: 'https://i.ibb.co/ZYW3VTp/brown-brim.png',  
    name: 'Brown Brim',  
    price: '25'  
  },  
  {  
    id: 2,  
    remoteUrl: 'https://i.ibb.co/ypkgK0X/blue-beanie.png',  
    name: 'Blue Beanie',  
    price: '18'  
  },  
  {  
    id: 3,  
    remoteUrl: 'https://i.ibb.co/QdJwgmp/brown-cowboy.png',  
    name: 'Brown Cowboy',  
    price: '35'  
  },  
  {  
    id: 4,  
    remoteUrl: 'https://i.ibb.co/RjBLWxB/grey-brim.png',  
    name: 'Grey Brim',  
    price: '25'  
  },  
  {  
    id: 5,  
    remoteUrl: 'https://i.ibb.co/xJS0T3Y/camo-vest.png',  
    name: 'Camo Down Vest',  
    price: '325'  
  },  
  {  
    id: 6,  
    remoteUrl: 'https://i.ibb.co/qMQ75QZ/floral-shirt.png',  
    name: 'Floral T-shirt',  
    price: '20'  
  },  
  {  
    id: 7,  
    remoteUrl: 'https://i.ibb.co/55z32tw/long-sleeve.png',  
    name: 'Black & White Longsleeve',  
    price: '25'  
  },  
  {  
    id: 8,  
    remoteUrl: 'https://i.ibb.co/RvwnBL8/pink-shirt.png',  
    name: 'Pink T-shirt ',  
    price: '25'  
},
```

```
];
export default items;
```

- 5. 將目前所完成的 React code 放入 Github 中，建立的 repository 要顯示 id 資訊。要繳交
 - . repository URL 及截圖，可以看到你的原始程式碼，
 - . Github page 設定之截圖
 - . Github page URL 及截圖，可以執行 React 程式
 - . 以上截圖，都必須有網址列，如 HW1，上面會顯示 github page 的資訊。

<https://github.com/nnj-wp/1101-209410751-tododemo2.git>

The screenshot shows a browser window with multiple tabs open. The active tab is titled "github.com/nnj-wp/1101-209410751-tododemo2/settings". The main content area displays the "GitHub Pages" settings. A yellow banner at the top states: "Pages settings now has its own dedicated tab! Check it out here!". Below this, there is a "Danger Zone" section containing four buttons: "Change repository visibility" (with "Change visibility" button), "Transfer ownership" (with "Transfer" button), "Archive this repository" (with "Archive this repository" button), and "Delete this repository" (with "Delete this repository" button). At the bottom of the page, there is a navigation bar with links to "Contact GitHub", "Pricing", "API", "Training", "Blog", and "About". The Windows taskbar at the bottom shows various pinned icons and the system tray.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is published at <https://nnj-wp.github.io/1101-209410751-tododemo2/>

Branch: main / (root) Save

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more](#).

Choose a theme

Custom domain

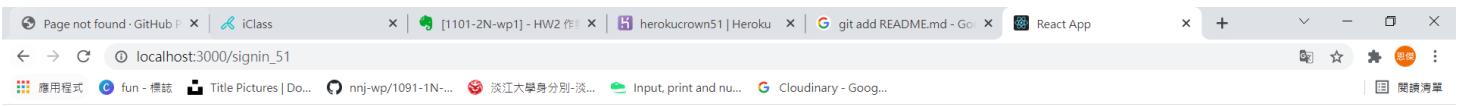
Custom domains allow you to serve your site from a domain other than nnj-wp.github.io. [Learn more](#).

Save Remove

Enforce HTTPS

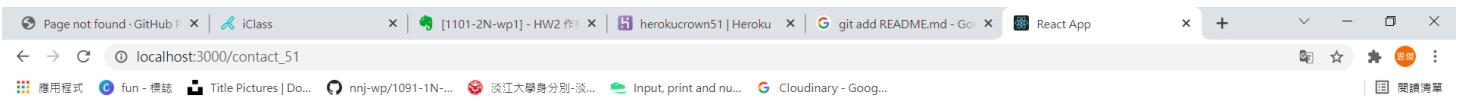
<https://nnj-wp.github.io/1101-209410751-tododemo2/>

- 6. (local) 加入 header，能顯示 logo 及選單，
 - 點選 logo，會回到主頁 (路由 /)
 - 點選 Shop，會到 Shop Overview Page (路由 /shop_xx)，要能顯示上面第 4 題的內容
 - 點選 Contact，會到 Contact Page (路由 /contact_xx)
 - 點選 Contact，會到 Signin Page (路由 /signin_xx)要繳交
 - . local 端 Chrome /shop_xx 執行結果截圖
 - . local 端 Chrome /contact 執行結果截圖
 - . 跟路由 /shop_xx 有關的原始程式碼



Shop Contact Sign In

SignInPage



Shop Contact Sign In

ContactPage

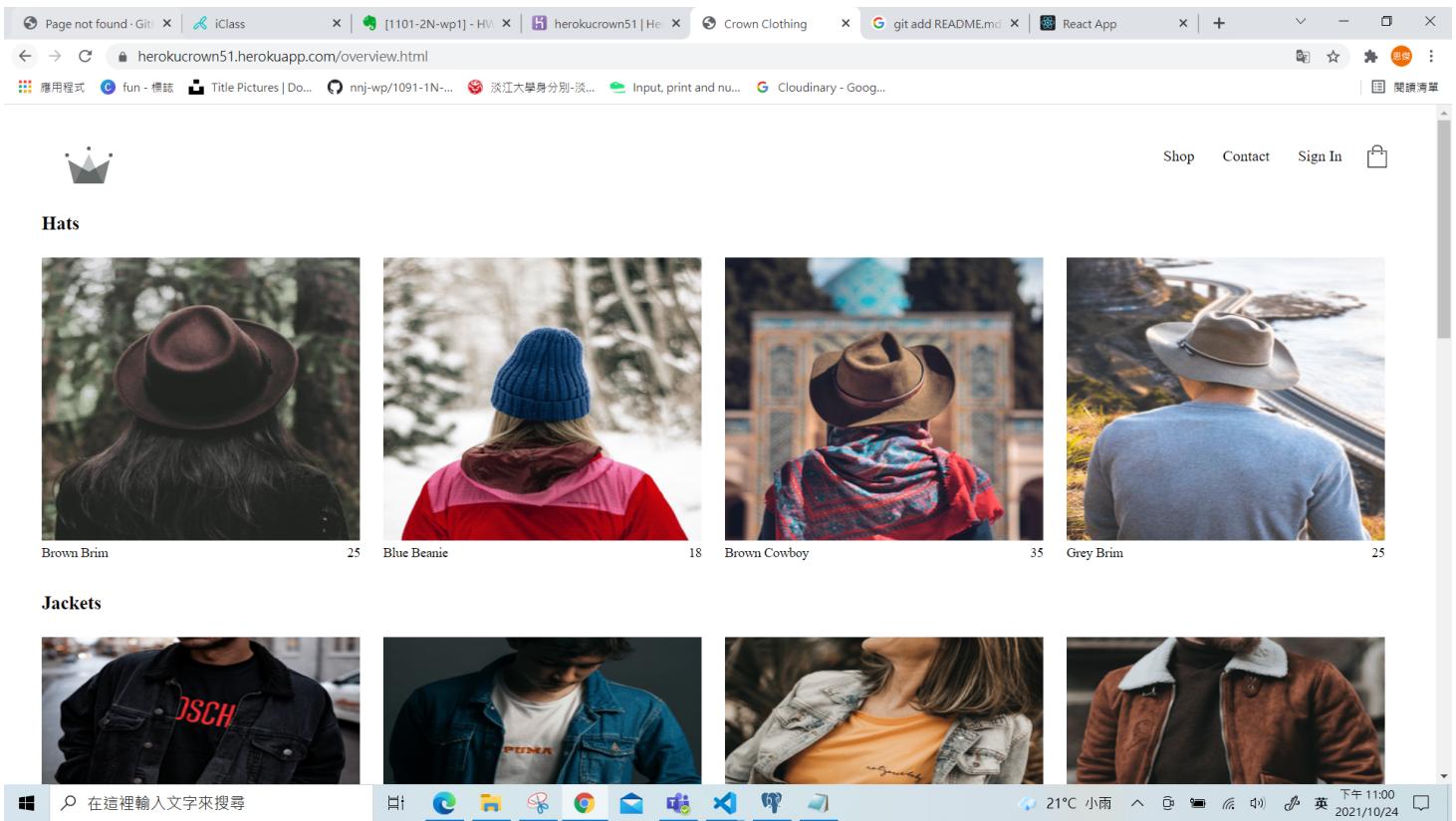


```

function App_51() {
  return (
    <div>
      <Header_51 />
      <Switch>
        <Route exact path="/" component={Homepage_51} />
        <Route exact path="/shop_51" component={ShopOverviewPage_51} />
        <Route exact path="/contact_51" component={ContactPage_51} />
        <Route exact path="/signin_51" component={SiginPage_51} />
      </Switch>
    </div>
  );
}

```

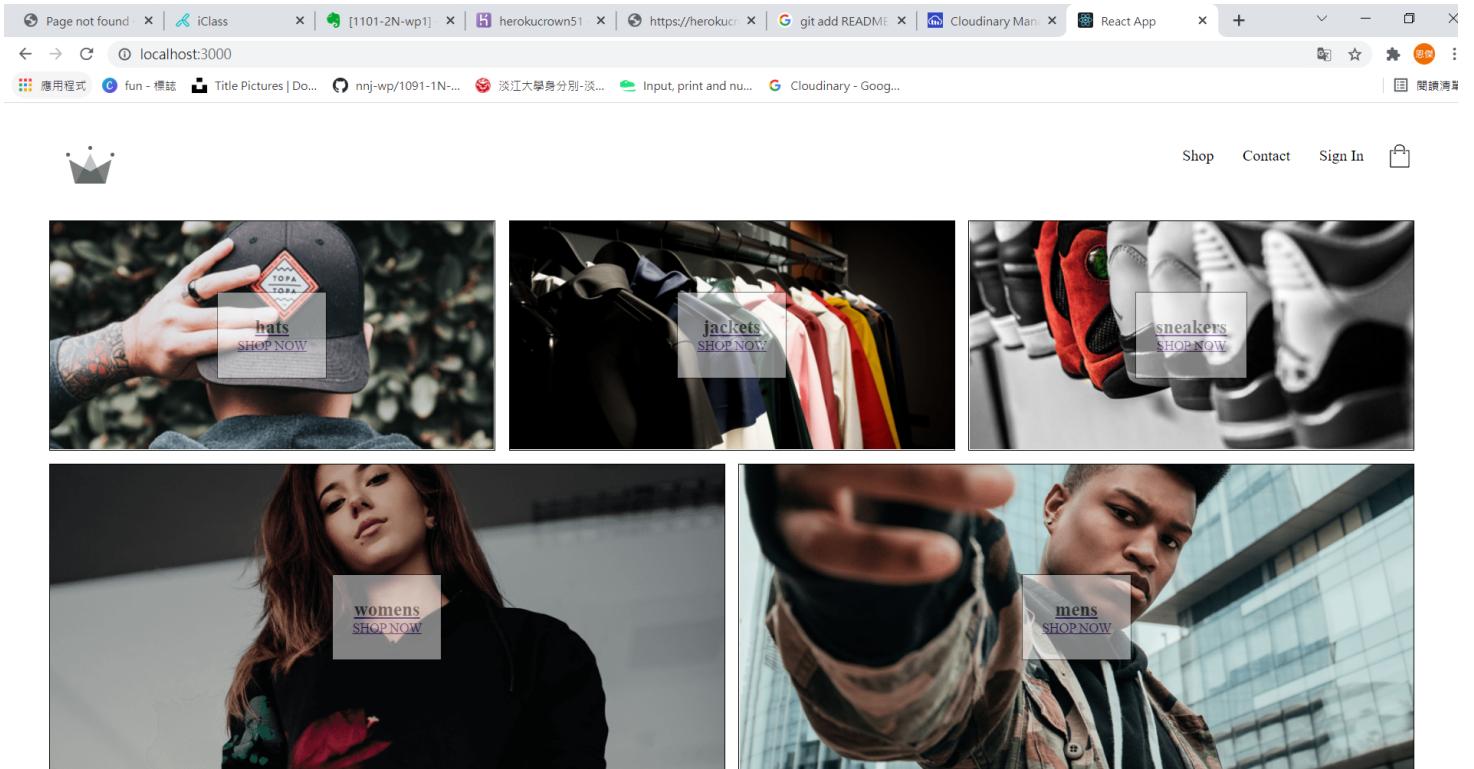
- 7. (heroku) 在 Heroku 上要能顯示第 6 題的 header。要繳交
 - . Heroku 上 Chrome /shop_xx 執行結果截圖



- 8. (local) 首頁 category 五筆資料，能夠透過你個人放在 Heroku /api_xx/category_xx 取得。如果無法成功，可以先用老師的測試。
 - . local 執行結果
 - . Directory2_xx 相關程式截圖

. Heroku app URL

. Heroku Database URL



A screenshot of Visual Studio Code showing the file `Directory2_51.js`. The code uses React hooks `useState` and `useEffect` to fetch menu items from a Heroku API. The `useEffect` hook sends a GET request to `'https://herokucrown51.herokuapp.com/api_51/Category_51'` and logs the response to the console. The fetched data is then mapped to create menu items.

```
import React, { useState, useEffect } from 'react';
import './Directory_51.scss';
// import items from './menu-items-data';
import axios from 'axios';
import MenuItem_51 from './MenuItem_51';

const Directory2_51 = () => {
  const [menuItems, setMenuItems] = useState([]);

  useEffect(() => {
    const fetchMenuItems = async () => {
      const HEROKU_URL =
        'https://herokucrown51.herokuapp.com/api_51/Category_51';
      const { data } = await axios.get(HEROKU_URL);
      console.log('/api/category_51', data);
      setMenuItems(data);
    };
    fetchMenuItems();
  }, []);
  console.log('menuItems', menuItems);
  return (
    <div>
      <div className="directory-menu">
        {menuItems.map(item => {
          const { id, name, remote_url, size } = item;
          return (
            <MenuItem_51
              key={id}
              name={name}
            </MenuItem_51>
          );
        })}
      </div>
    </div>
  );
}

export default Directory2_51;
```

heroku =>

<https://herokucrown51.herokuapp.com/>

Database =>

https://herokucrown51.herokuapp.com/api_51/Category_51

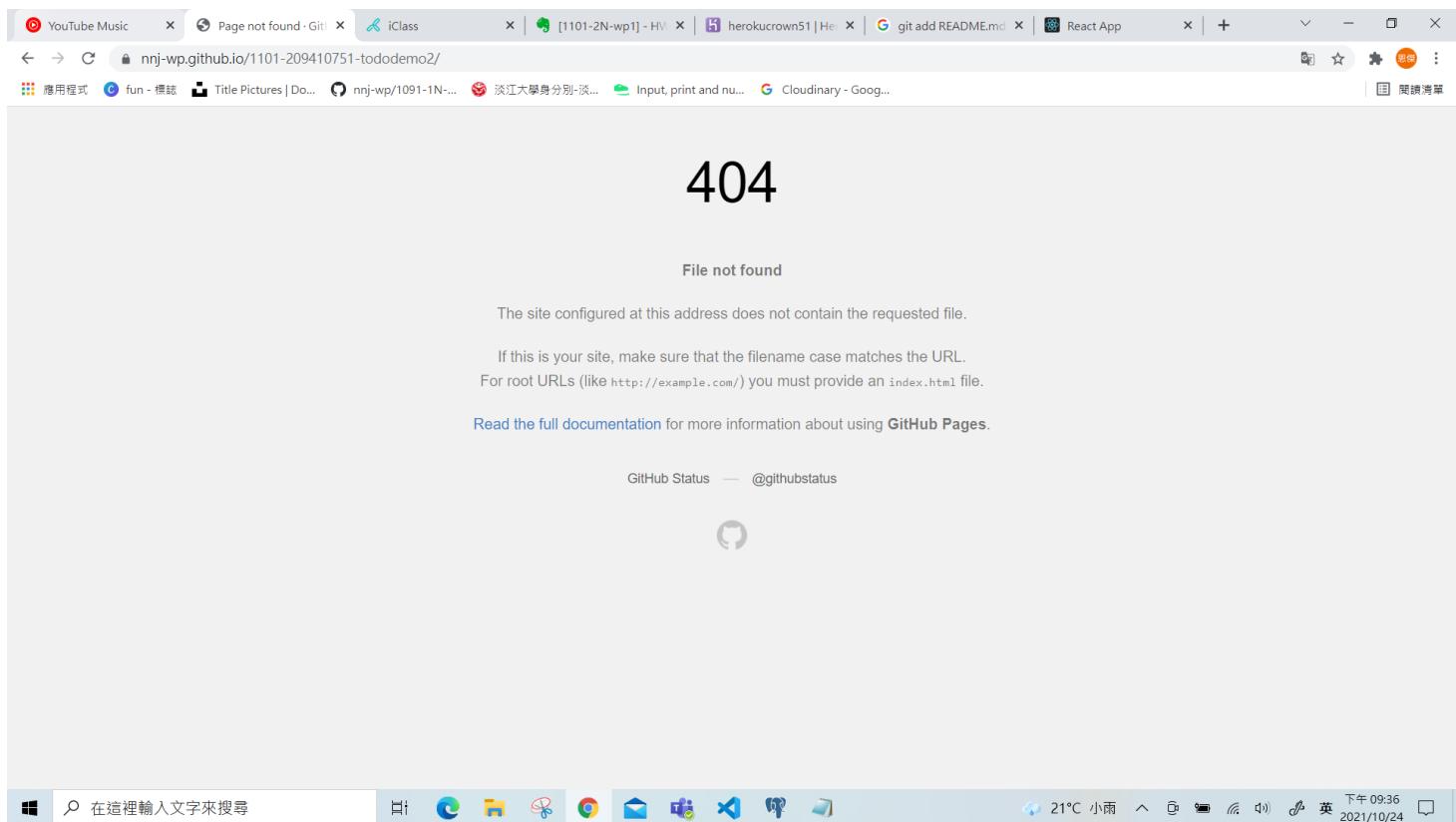
- 9. (Github page) 上第 8 題，請將程式碼發佈到 github page，並能直接從 Github page 執行，自 Heroku /api_xx/category_xx 取得 category 五筆資料，要繳交
 - . Github repo URL
 - . Github page URL
 - . Chrome 圖片有關 Github page 執行路由 / 結果
 - . Chrome 圖片有關 Github page 執行路由 /shop_xx 結果

Github repo URL =>

<https://github.com/nnj-wp/1101-209410751-tododemo2.git>

Github page =>

<https://nnj-wp.github.io/1101-209410751-tododemo2/>



vercel

404

File not found

The site configured at this address does not contain the requested file.

If this is your site, make sure that the filename case matches the URL.
For root URLs (like <http://example.com/>) you must provide an `index.html` file.

[Read the full documentation](#) for more information about using GitHub Pages.

[GitHub Status](#) — [@githubstatus](#)

