

Important Instructions

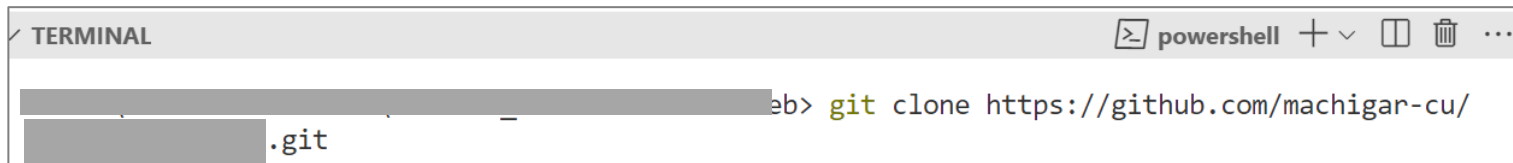
- การส่ง Assignment ผ่าน [Github Classroom](#)
 - งานใน **Assignment 09** เป็นงานต่อจาก **Assignment 08** แต่ใช้ Repository ที่ตั้งต้นที่ให้มา ผ่าน Assignment Invitation Link สำหรับ Assignment 09
 - การส่ง Assignment ภายใต Student Identity ที่กำหนดไว้ตั้งแต่ตอนรับ Invitation ของ Assignment 01
- นิสิตทดสอบ
 - Assignment มาพร้อม Test Script สำหรับทดสอบ project ให้นิสิตทดสอบตาม instruction ในสไลด์ชุดนี้
- นิสิตส่งงานโดย
 - Deploy เว็บไซต์ที่สร้างจาก Assignment ขึ้น Vercel
 - Commit และ push ขึ้น repository ที่ได้จาก Assignment
 - นิสิตต้องใส่ link ไปยัง Vercel website ของนิสิต ใน README.md ใน Repository

Prepare Class Environment



- เปิด Folder สำหรับเก็บหน้า React Web ที่สร้างไว้ใน Assignment 03 ด้วย Visual Studio Code
- ไปที่ Menu View > Terminal

- Clone Repo ของตนเอง พิมพ์คำสั่ง `git clone REPO_URL_FROM_GITHUB`



```
TERMINAL powershell + - [x] [x] ...
C:\Users\user> git clone https://github.com/machigar-cu/
.git
```

Note: ถ้ายังไม่ได้ setup git credential ในเครื่อง อาจขึ้นว่า repository not found ให้เพิ่ม user ของ git account ตามด้วย @ ไปด้านหน้า github

`git clone https://YOUR_USER@github.com/machigar-cu/.....git`

- **YOUR_USER** คือ Github user ของนิสิต

- เปิด Folder ที่ Clone มาได้ (`a09-swdevprac2-xxxxxx`)
ใน Visual Studio Code

- จะพบไฟล์ตั้งต้นของ project

Prepare Class Environment



- ไปที่เมนู View > Terminal

- Configure git username และ email สำหรับ local repository

```
git config --local user.email "example@email.com"
```

```
git config --local user.name "your_userName"
```

```
machigar> git config --local user.email "machigar@gmail.com"
machigar> git config --local user.name "machigar"
```

- ติดตั้ง node packages `npm install`
- Copy ไฟล์เตอร์ `public` และ `src` จาก project ของ Assignment 08 มาทับ ไฟล์เตอร์ `public` และ `src` ใน project ของ Assignment 09 นี้
- ลอง run project จะได้ project ที่ทำไว้ใน Assignment 08 `npm run dev`

Assignment Backend



Backend สำหรับ Assignment นี้ สร้างไว้ให้อยู่แล้ว นิสิตสามารถเข้าใช้งาน API ได้ดังนี้

- API สำหรับ Fetch ข้อมูลรายการโรงพยาบาล

<https://vaccine-app-backend.vercel.app/api/v1/hospitals>

```
← → ↺ 🏠 vaccine-app-backend.vercel.app/api/v1/hospitals
{
  success: true,
  count: 3,
  pagination: { },
  data: [
    - {
      _id: "651fad5f05c6b313f8dfb1e5",
      name: "Chulalongkorn Hospital",
      address: "1873 Rama IV Rd",
      district: "Pathum Wan",
      province: "Bangkok",
      postalcode: "10330",
      tel: "026494000",
      picture: "https://drive.google.com/uc?id=1nckijHnFtGMdbmsYrBao55dGvtYER62P",
      _v: 0,
      id: "651fad5f05c6b313f8dfb1e5"
    },
    - {
      _id: "651fae4b05c6b313f8dfb1e8",
      name: "Rajavithi Hospital",
      address: "2 Phaya Thai Rd, Thung Phaya Thai",
      district: "Ratchathewi",
      province: "Bangkok",
      postalcode: "10400",
      tel: "022062900",
      picture: "https://drive.google.com/uc?id=15kwfVT9wchkh3I9BHKeqftTmj0bFgJtu",
      _v: 0,
      id: "651fae4b05c6b313f8dfb1e8"
    },
    - {
      _id: "651faed405c6b313f8dfb1e9",
      name: "Siriraj Hospital",
      address: "284 Ratchadaphisek Rd, Siriraj",
      district: "Siriraj",
      province: "Bangkok",
      postalcode: "10500",
      tel: "022611111",
      picture: "https://drive.google.com/uc?id=15kwfVT9wchkh3I9BHKeqftTmj0bFgJtu",
      _v: 0,
      id: "651faed405c6b313f8dfb1e9"
    }
  ]
}
```

hid
(hospital ID)



- API สำหรับ Fetch ข้อมูลโรงพยาบาล ตาม Hospital ID (**hid**)

[https://vaccine-app-backend.vercel.app:443/api/v1/hospitals/hid](https://vaccine-app-backend.vercel.app/api/v1/hospitals/hid)

```
← → ↻ 🏠 🌐 vaccine-app-backend.vercel.app/api/v1/hospitals/652fb81d6b64aeec30243507

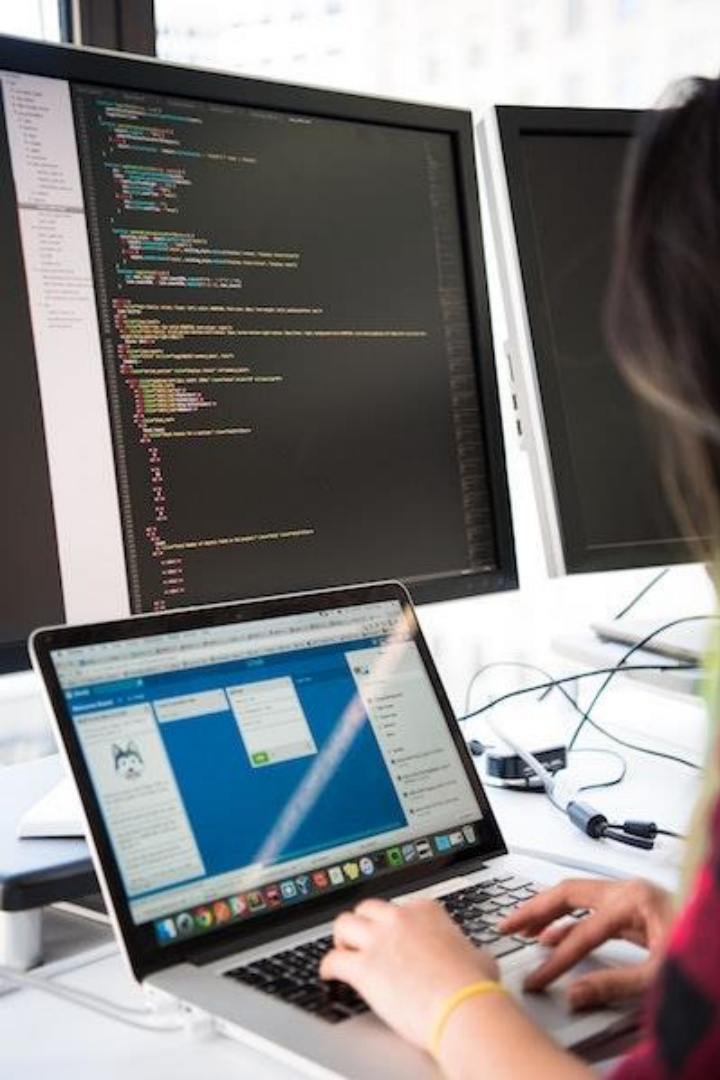
{
  success: true,
  - data: {
    _id: "652fb81d6b64aeec30243507",
    name: "ssss",
    address: "ssss",
    district: "dssf",
    province: "fdf",
    postalcode: "10111",
    tel: "string",
    picture: "https://drive.google.com/uc?id=1nekIjHnFtGMdbmsYrBao55dGvtYER62P",
    __v: 0,
    id: "652fb81d6b64aeec30243507"
  }
}
```

JSON Object Definition



- เพื่อรองรับการ Deploy ขึ้น Vercel ซึ่งบังคับการ program แบบ Strict Type
- ใน Starter Code ที่ให้มา มีการนิยาม `interface` สำหรับ JSON Object ที่ได้รับมาจาก Backend API
- เปิดไฟล์ `interface.ts` ที่ root ของ project directory ซึ่งให้มาใน Starter Code
 - `HospitalJson` นำเสนอ JSON Object ที่ได้จาก <https://vaccine-app-backend.vercel.app/api/v1/hospitals>
 - พิลด์ `data` เป็นอาร์เรย์ ของ `HospitalItem`
- Interface `HospitalItem` และ `HospitalJson` นี้จะถูกนำไปใช้กำหนด type ของ JSON object ใน assignment นี้

```
interface HospitalItem {  
  _id: string,  
  name: string,  
  address: string,  
  district: string,  
  province: string,  
  postalcode: string,  
  tel: string,  
  picture: string,  
  __v: number,  
  id: string  
}  
  
interface HospitalJson {  
  success: boolean,  
  count: number,  
  pagination: Object,  
  data: HospitalItem[]  
}
```



Assignment 9: Data Fetching



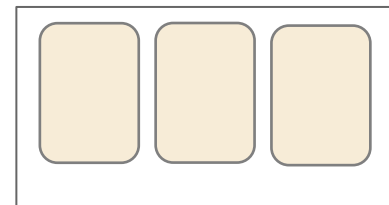
ใน Next.js Project `vaccine-book-app`

- สร้างโฟลเดอร์ `libs` ใน `src` สำหรับเก็บฟังก์ชันต่าง ๆ
- สร้างไฟล์ `libs\getHospitals.tsx`
 - บรรจุฟังก์ชัน `getHospitals` สำหรับ fetch ข้อมูลโรงพยาบาลมาจาก Backend API
 - เพิ่ม `timeout 1000 ms` ก่อนการ `fetch` เพื่อเพิ่ม `delay`



- ปรับปรุง `Card.tsx` ให้รับ props ที่เกี่ยวกับการทำงานของ Rating เป็น optional
 - หากไม่ได้รับ props ที่เกี่ยวกับการทำงานของ Rating จะไม่แสดง Rating บน Card
- สร้าง component `src/components/HospitalCatalog.tsx` เป็น `async functional component` ใช้แสดงรายการโรงพยาบาลที่ดึงมาจาก backend API 3 รายการ
 - มี prop ชื่อ `hospitalsJson` เป็นชนิด `Promise<HospitalJson>` รับข้อมูลที่ fetch มาได้จาก Backend API
 - map ข้อมูลแต่ละชิ้นในฟิลด์ `data` ของ JSON ให้เป็นข้อมูลชนิด `HospitalItem` เพื่อนำมาแสดงบน Card
 - ใช้งาน Component `src/components/Card.tsx` ที่สร้างไว้ใน Assignment ที่ 3 และ 5 เพื่อแสดง Card ในลักษณะดังภาพ
 - ไม่ต้องกำหนด props ที่เกี่ยวกับการทำงานของ Rating ให้กับ Card ที่นำมาวาง

HospitalCatalog

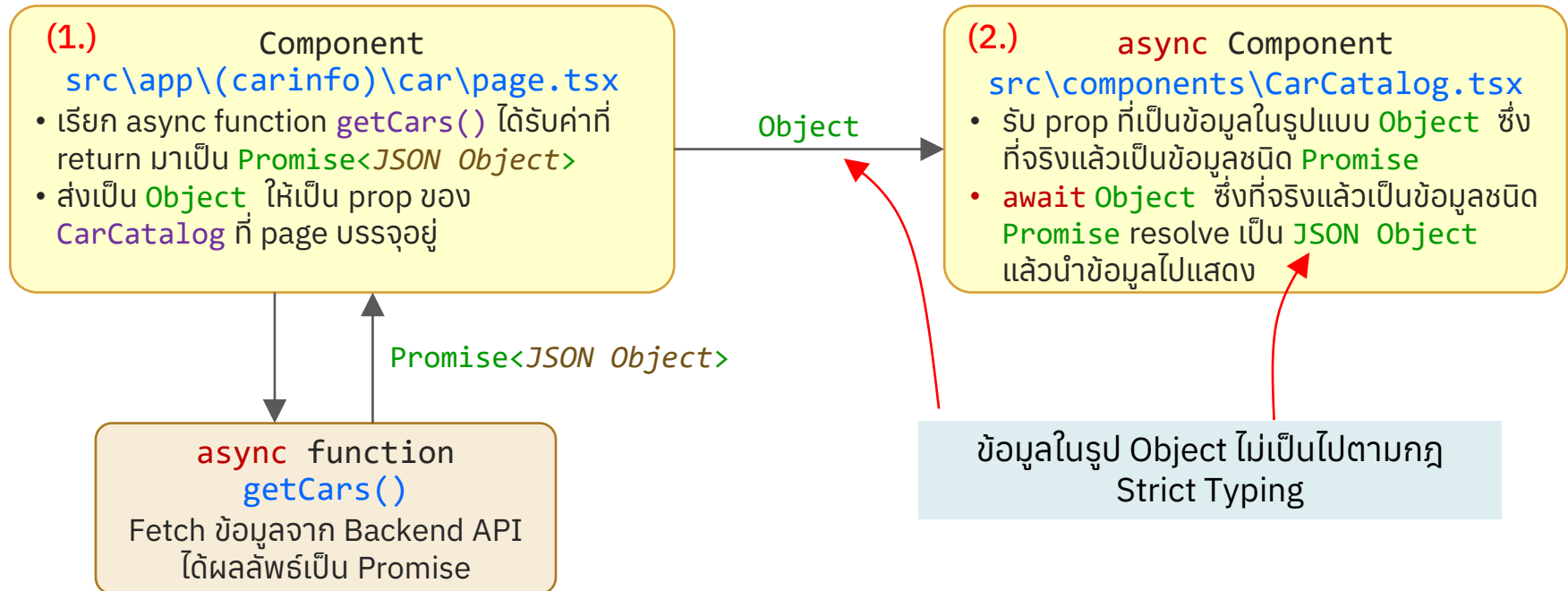




- ที่ `page.tsx` ของ route `\hospital` ใน Route Group (`hospitalinfo`) ที่สร้างใน Assignment 07
 - เรียกฟังก์ชัน `getHospitals` เพื่อ fetch ข้อมูลโรงพยาบาลมาจาก Backend API
 - ลบการใช้งาน `CardPanel` ออกจาก `page`
 - นำ `HospitalCatalog` มาแสดงผลข้อมูลโรงพยาบาลที่ fetch มาจาก Backend API แทน
 - ใช้ `Suspense` เพื่อแสดง loading UI ระหว่างรอการ fetch ข้อมูล โดยใช้ `LinearProgress` จาก Material UI Library
 - เมื่อคลิกบน card ให้ navigate ไปยัง route `\hospital\[hid]` เมื่อ `hid` คือ id ของโรงพยาบาล

Hint: แสดงชุดข้อมูล Hospitals แบบ Strict Typing

- Assignment 09 นี้ มีความแตกต่างจากตัวอย่าง Car Rental App ในคลาส เนื่องจากเรามีการ deploy ขึ้น Vercel ซึ่งต้องการ Strict Typing
- ตัวอย่าง Car Rental App มีการส่งข้อมูลในรูป Object ซึ่งไม่เป็นไปตามกฎ Strict Typing



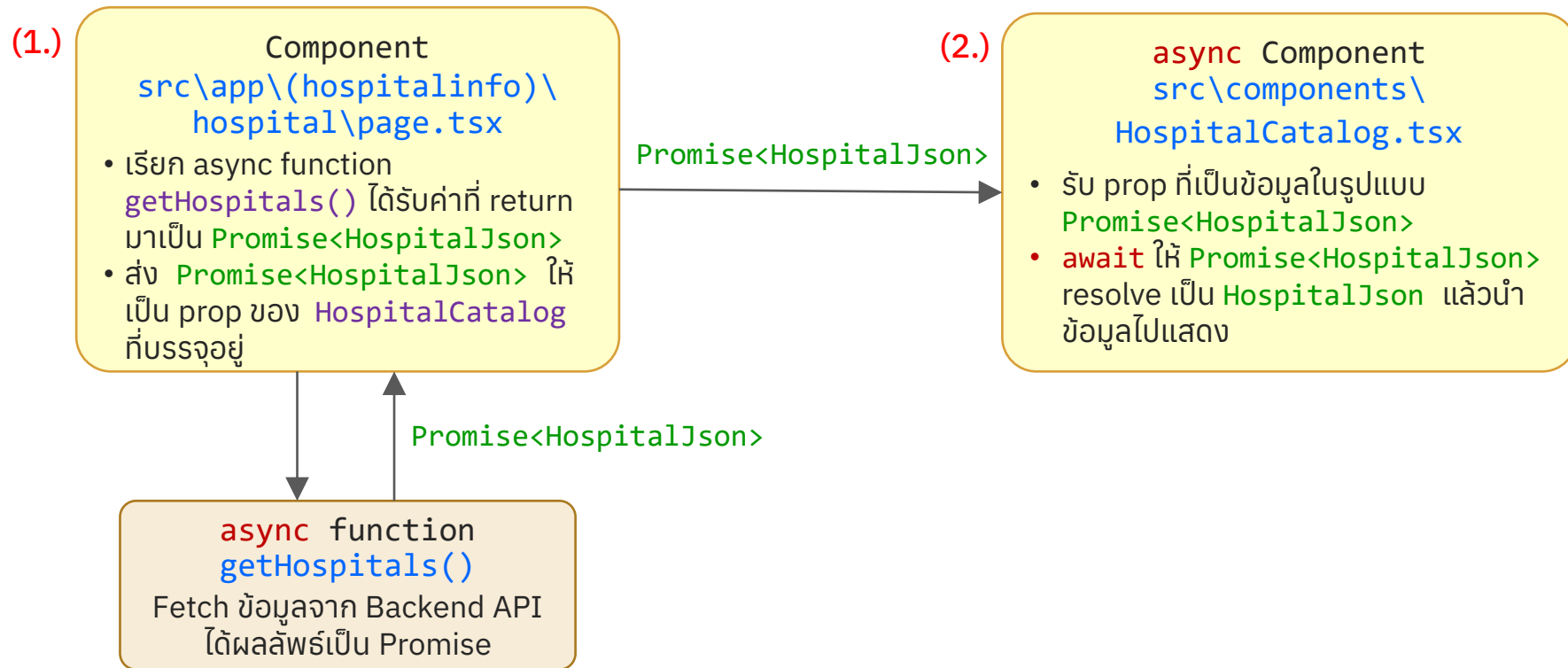
Hint: แสดงชุดข้อมูล Hospitals แบบ Strict Typing

- Assignment 09 นี้ นำข้อมูล hospitals จาก Backend API มาแสดงบน **HospitalCatalog**
- เพื่อให้ deploy ขึ้น Vercel ได้
เราจึงใช้งานข้อมูลที่มีการนิยามโครงสร้าง
(ผ่านการนิยาม interface)
- JSON Object ที่ได้จาก Backend API มีโครงสร้างตาม
Interface ที่นิยามไว้ และ Promise ก็เป็น Promise ที่ Resolve
เป็น JSON Object ที่มีโครงสร้างดังกล่าว
- ใน assignment นี้ให้ implement 2 โดยให้ **HospitalCatalog**
รูปแบบ **Promise<HospitalJson>**

```
interface HospitalItem {  
  _id: string,  
  name: string,  
  address: string,  
  district: string,  
  province: string,  
  postalcode: string,  
  tel: string,  
  picture: string,  
  __v: number,  
  id: string  
}  
  
interface HospitalJson {  
  success: boolean,  
  count: number,  
  pagination: Object,  
  data: HospitalItem[]  
}
```

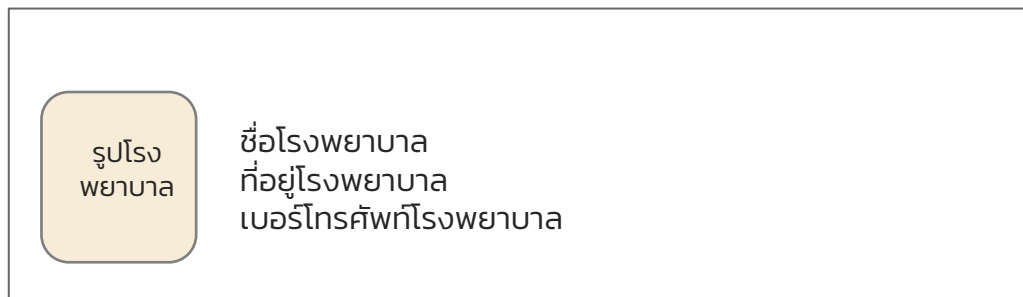
Hint: แสดงชุดข้อมูล Hospitals แบบ Strict Typing

- `HospitalCatalog` รับ prop ที่เป็นข้อมูลในรูปแบบ `Promise<HospitalJson>`





- ในไฟล์เตอร์ `libs` ที่เก็บฟังก์ชันต่าง ๆ สร้างไฟล์ `getHospital.tsx` บรรจุฟังก์ชัน `getHospital` สำหรับ fetch ข้อมูลโรงพยาบาลตาม id ของโรงพยาบาล (`hid`)
- บน `page.tsx` ของ Dynamic Segment `\hospital\[hid]` ที่สร้างใน Assignment 07 ใน Route Group `(hospitalinfo)\hospital` ให้เรียกฟังก์ชัน `getHospital` เพื่อ fetch ข้อมูลโรงพยาบาลที่มี `id = hid` มาจาก Backend API
 - แสดงรายละเอียดของโรงพยาบาล id นั้น ๆ โดยหน้าแสดงรายละเอียดให้แสดงรูป ชื่อของโรงพยาบาล ที่อยู่ จังหวัด รหัสไปรษณีย์ เบอร์โทรศัพท์






- บน `page.tsx` ของ Dynamic Segment `\hospital\[hid]` ที่สร้างใน Assignment 07 ใน Route Group `(hospitalinfo)\hospital`
 - ลบหรือ comment ทิ้งฟังก์ชัน `generateStaticParams` (ถ้ามีการสร้างไว้)
 - ป้องกันไม่ให้เกิดเมื่อ deploy ขึ้น vercel แล้ว ระหว่างการสร้าง static page ที่ server-side นั้น vercel จะพยายาม fetch ข้อมูลของ hospital ที่มี id ตามที่ระบุไว้ใน `generateStaticParams` และจะ fail เนื่องจาก Backend API ไม่มี mock ID เหล่านั้น

Data Fetching




- ทดสอบการทำงานโดยไปที่
<http://localhost:3000/hospital>

Select your hospital
Explore 3 models in our catalog




Chulalongkorn Hospital

☆☆☆☆☆



Rajavithi Hospital

☆☆☆☆☆



Thammasat University Hospital

☆☆☆☆☆

Chulalongkorn Hospital



Name: Chulalongkorn Hospital
Address: 1873 Rama IV Rd
District: Pathum Wan
Postal Code: 10330
Tel: 026494000

Assignment test cases

Assignment 09 มี 4 test cases (รวม autograde 7 คะแนน)

- ลอง run test case ที่เครื่องของตนเอง ไปที่เมนู **View > Terminal**
 - Run **test case 1 (2 คะแนน)** ด้วยคำสั่ง
`npm run test -- case1.test.tsx`
 - Run **test case 2 (2 คะแนน)** ด้วยคำสั่ง
`npm run test -- case2.test.tsx`
 - Run **test case 3 (1 คะแนน)** ด้วยคำสั่ง
`npm run test -- case3.test.tsx`
 - Run **test case 4 (2 คะแนน)** ด้วยคำสั่ง
`npm run test -- case4.test.tsx`

Hint: Vercel Deployment

- ในขณะที่ Deploy ขึ้น Vercel นั้น Code ต่าง ๆ จะได้รับการ Build โดยมีการตรวจสอบ Type Safely ต่าง ๆ ตัวอย่างเช่น
 - Strict Typing
 - Type validity รวมถึง undefined ต่าง ๆ โดยเฉพาะจาก optional props
- ทดลอง build ใน Local Environment ก่อน Deploy ขึ้น Vercel ได้ผ่านคำสั่ง
`npm run build`
- การ build จะทำให้มีการ pre-render หน้าต่าง ๆ ให้เป็น static content และ Optimize เพื่อลด runtime โดยทดลอง run ผลลัพธ์ที่ได้จากการ build ดังนี้
`npm run start`
 - จะเห็นได้ว่า website สามารถโหลดได้รวดเร็ว และมีการ optimize การทำงานเพื่อให้ delay น้อยที่สุด

Hint: Vercel Deployment

- Vercel จำกัดการ fetch ข้อมูลจาก Cross Origin (ใช้ Same-Origin Policy เพื่อความปลอดภัย)
- ใน Assignment นี้เราจำเป็นต้อง Enable การใช้งาน Cross-origin Resource Sharing (CORS) เพื่อดึงข้อมูลจาก API ภายนอก จึงได้เพิ่มการ config ในไฟล์ `next.config.mjs` จาก Starter Code

```
6     async headers() {
7         return [
8             {
9                 // matching all API routes
10                source: "/hospital/:path*",
11                headers: [
12                    { key: "Access-Control-Allow-Credentials", value: "true" },
13                    { key: "Access-Control-Allow-Origin", value: "*" }, // replace this your actual origin
14                    { key: "Access-Control-Allow-Methods", value: "GET,DELETE,PATCH,POST,PUT" },
15                    { key: "Access-Control-Allow-Headers", value: "X-CSRF-Token, X-Requested-With, Accept, Accept-Version, Content-Length, Content-MD5, Content-Type, Date, X-API-Version" },
16                ]
17            }
18        ]
19    }
20 };
```

ปัจจุบันเปิดให้ทุก origin

Development & Deployment

- ใช้แนวทางการ Deploy แบบเดียวกับใน Assignment A03 ถึง A08
 - สร้าง Repository บน Personal GitHub Account สำหรับ Assignment A09
 - ดู Instructions ใน Assignment A01
 - Config ให้ local project มี multiple remote repositories และ Push งานขึ้น repositories ทั้งสอง
 - ดู Instructions ใน Assignment A01
 - Deploy project จาก Personal GitHub Account ขึ้น Vercel
 - ดู Instructions ใน Assignment A01
 - Configure และ Deploy บน Vercel โดยระบบจะตรวจจับได้ว่าเป็น Next.js Framework
- ใส่ link ไปยัง Vercel website ของนิสิต ใน **README.md** ใน Repository