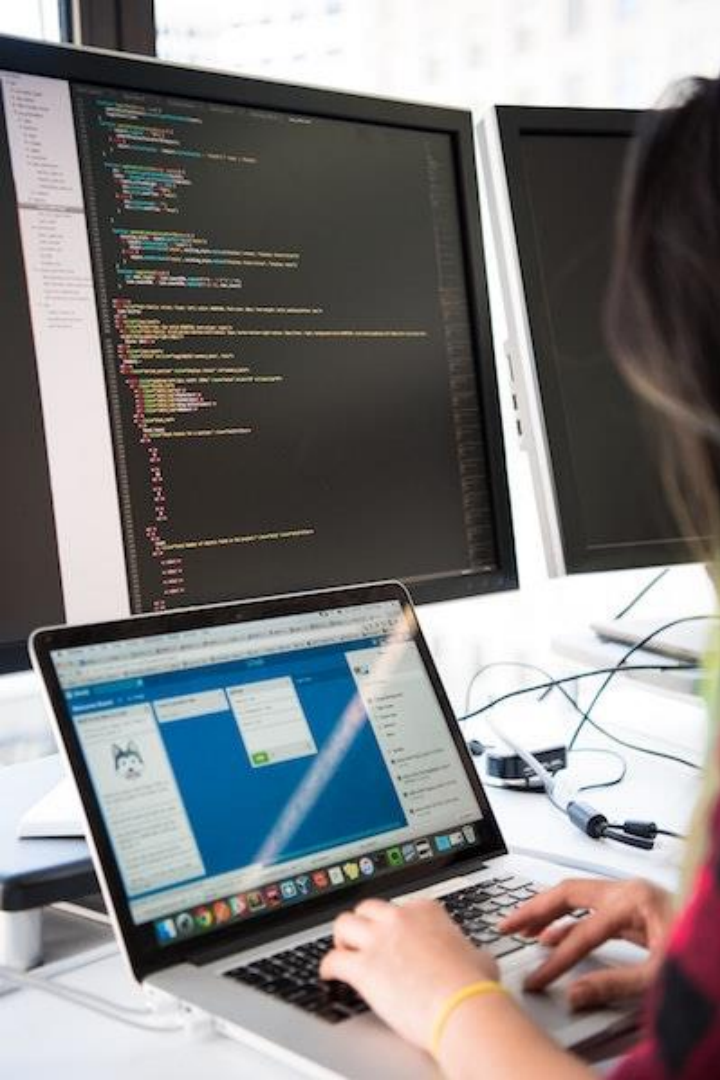


# Important Instructions

- การส่ง Assignment ผ่าน [Github Classroom](#)
  - งานใน **Assignment 11** เป็นงานต่อจาก **Assignment 10** แต่ใช้ Repository เริ่มต้นที่ให้มา ผ่าน Assignment Invitation Link สำหรับ Assignment 11
  - การส่ง Assignment ภายใต้ 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



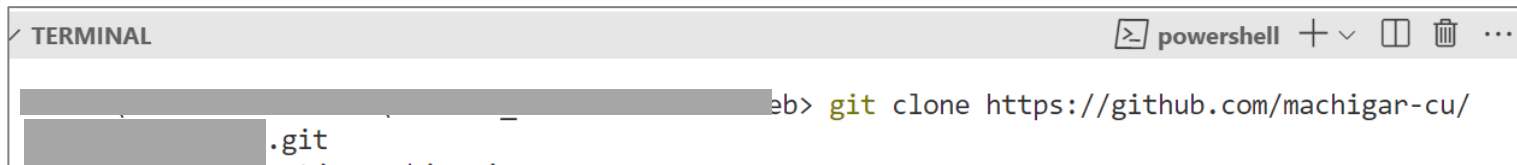
## Assignment 11: Redux

# 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
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 (`a11-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`

# Prepare Class Environment



- **Copy** โฟลเดอร์และไฟล์ต่อไปนี้จาก project ของ Assignment 10
  - โฟลเดอร์ `public` มาทับ โฟลเดอร์ `public` ใน project ของ Assignment 11
  - โฟลเดอร์ `src` มาทับ โฟลเดอร์ `src` ใน project ของ Assignment 11
  - ไฟล์ `.env.local` มาที่ root ของ project ของ Assignment 11
- ลอง run project จะได้ project ที่ทำไว้ใน Assignment 10 `npm run dev`

# Prepare Class Environment



- ติดตั้ง Redux Toolkit ใน [vaccine-booking-app](#) เพื่อใช้ในการเก็บข้อมูลการจองวัคซีน
  - ใช้ [ReduxJS Toolkit V2.2.8](#) และ [React Redux V.9.1.2](#) ผ่าน console ด้วยคำสั่ง

```
npm i @reduxjs/toolkit@2.2.8 react-redux@9.1.2
```

- แก้ไขไฟล์ `interfaces.ts` ที่ Root Directory ของ project สำหรับระบุโครงสร้าง booking item โดยตั้งชื่อ interface เป็น `BookingItem` โดยประกอบด้วย
  - `name` เก็บ string ที่เป็น ชื่อ ของผู้จองวัคซีน
  - `surname` เก็บ string นามสกุล ของผู้จองวัคซีน
  - `id` เก็บ string รหัสประจำตัวประชาชน ของผู้จองวัคซีน
  - `hospital` เก็บ string โรงพยาบาล ที่ต้องการรับวัคซีน
  - `bookDate` เก็บ string วันที่ต้องการรับวัคซีน

```
interface HospitalItem {  
  success: boolean,  
  count: number,  
  pagination: Object,  
  data: HospitalItem[]  
}
```

```
interface BookingItem {  
  name: string;  
  surname: string;  
  id: string;  
  hospital: string;  
  bookDate: string;  
}
```



- สร้างไฟล์ `src/redux/features/bookSlice.ts` สำหรับเก็บ State Slice
  - นิยาม type ชื่อ `BookState` เป็น Object ที่มี attribute `bookItems` เก็บ array ของ `BookingItem`
  - ให้ Initial State เป็น Object ชนิด `BookState` ที่ `bookItems` เป็น empty array
  - สร้าง Reducer Function ให้กับ `bookSlice` โดย
    - ฟังก์ชัน `addBooking` รับ Payload action เป็นชนิด `BookingItem` สำหรับเพิ่มข้อมูล booking การจองวัคซีน โดยแต่ละรหัสบัตรประชาชน สามารถทำการจองได้เพียงการจองเดียว หากมีการจองวัคซีนใหม่ที่คำรหัสประชาชนดังกล่าวเคยถูกใช้ทำการจองมาก่อนหน้าแล้ว ข้อมูลการจองใหม่จะไปแทนที่ข้อมูลการจองเดิม
    - ฟังก์ชัน `removeBooking` รับ Payload action ชนิด `string` เป็น `id` หรือรหัสบัตรประชาชน สำหรับลบข้อมูล booking การจองวัคซีนของผู้จองที่มีรหัสบัตรประชาชนดังกล่าว
- สร้างไฟล์ `src/redux/store.ts` สำหรับเก็บ Redux Store โดยมี `bookSlice` เป็น Reducer State Slice



- สร้างไฟล์ `src/redux/ReduxProvider.tsx` สำหรับใช้เข้าถึง Redux Store State และนำไปใช้งาน
- ที่ `page.tsx` ของ route สำหรับการนัดรับวัคซีน ที่ <http://localhost:3000/booking>
  - ลบส่วนของการใช้งาน Server Session, การ `getUserProfile` และส่วนแสดงข้อมูล User Profile ออก
  - ปรับ `page.tsx` ให้เป็น **client component** และไม่เป็น async component
  - ใช้ Form การจองรับวัคซีน ที่สร้างไว้ใน Assignment 05 โดยปรับให้ฟอร์มประกอบด้วย
    - `TextField` ที่มีแอททริบิวต์ `name` เป็น Name รับชื่อ
    - `TextField` ที่มีแอททริบิวต์ `name` เป็น Lastname รับนามสกุล
    - `TextField` ที่มีแอททริบิวต์ `name` เป็น Citizen ID รับรหัสประจำตัวประชาชน
    - `Select` ที่ตั้ง `id` เป็น hospital รับโรงพยาบาล โดยเลือกจาก 3 ตัวเลือก Chulalongkorn Hospital, Rajavithi Hospital, Thammasat University Hospital
    - `DatePicker` ที่รับวันที่ต้องการรับวัคซีน
    - `Button` ปุ่มจองวัคซีนที่มีแอททริบิวต์ `name` เป็น Book Vaccine เพื่อส่งฟอร์ม
  - ปรับปรุงให้เมื่อกดปุ่ม `Button` ที่มีแอททริบิวต์ `name` เป็น Book Vaccine แล้วให้สร้างการจองวัคซีนและเก็บข้อมูลไว้ใน Redux Store

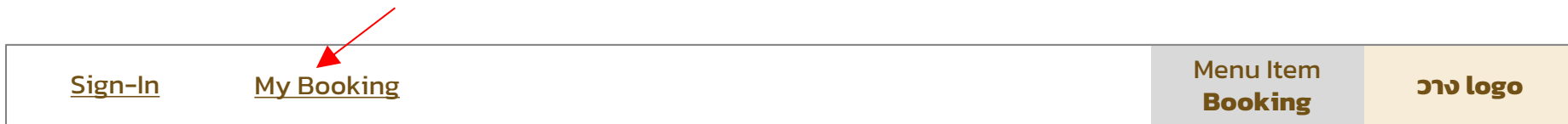




- สร้าง `components\BookingList.tsx` เพื่อแสดงข้อมูลในการจองวัคซีนที่ user จองไว้
  - อ่านข้อมูลการจองวัคซีนจาก Redux Store มาแสดงบนหน้านี้
  - สำหรับแต่ละข้อมูลการจอง ให้แสดง
    - ชื่อ ของผู้จองวัคซีน
    - นามสกุล ของผู้จองวัคซีน
    - รหัสประจำตัวประชาชน ของผู้จองวัคซีน
    - โรงพยาบาล ที่ต้องการรับวัคซีน
    - วันที่ต้องการรับวัคซีน
  - หากไม่มีข้อมูลการจองวัคซีน ให้แสดงข้อความว่า “No Vaccine Booking”
  - เพิ่มปุ่มยกเลิกการจองวัคซีน โดยเมื่อกดปุ่มให้ลบการจองวัคซีนจาก Redux Store



- สร้าง `page.tsx` สำหรับ route ไปยังหน้าแสดงข้อมูลการนัดรับวัคซีน ที่ <http://localhost:3000/mybooking> แล้วนำ Component `BookingList` มาวางไว้บนหน้านี้
- ปรับ `TopMenu.tsx` ให้เพิ่มเมนูสำหรับเข้าถึง route <http://localhost:3000/mybooking>



# Assignment test cases

## Assignment 11 มี 2 test cases

- 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
```

# Development & Deployment

- ใช้แนวทางการ Deploy แบบเดียวกับใน Assignment A03 ถึง A10
    - สร้าง Repository บน Personal GitHub Account สำหรับ Assignment A11
      - 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

# Deploy to Vercel

- เพื่อให้ Site ที่ Deploy ขึ้น Vercel สามารถทำงานได้แบบ Site ที่เราทดสอบที่ localhost เราต้องตั้ง **environment variables** ซึ่งประกาศไว้ที่ไฟล์ `.env.local` ให้กับ Vercel ด้วย
- ในการ Configure Project ช่วง deploy
  - เลือกส่วน **Environment Variables** กำหนด Environment Variables ต่อไปนี้
  - **NEXTAUTH\_SECRET** กำหนดค่าเป็นค่าเดียวกับที่กำหนดไว้ใน `.env.local`
  - เราจะมาใส่ **NEXTAUTH\_URL** หลังจาก Deploy สำเร็จ เพราะต้องใช้ URL ของ Vercel ที่ได้มาจากการ Deploy

```
$ .env.local X
$ .env.local
1  NEXTAUTH_URL=http://localhost:3000
2  NEXTAUTH_SECRET='Ey7nTKnggBc0bRN8WUjyShw2qzOZ6KW4fUyqcKBePxY='
```

**Configure Project**

Project Name  
[ ] one

Framework Preset  
Next.js

Root Directory  
./

> Build and Output Settings

Environment Variables

Key	Value (Will Be Encrypted)
NEXTAUTH_SECRET	Bc0bRN8WUjyShw2qzOZ6KW4fUyqcKBePxY=

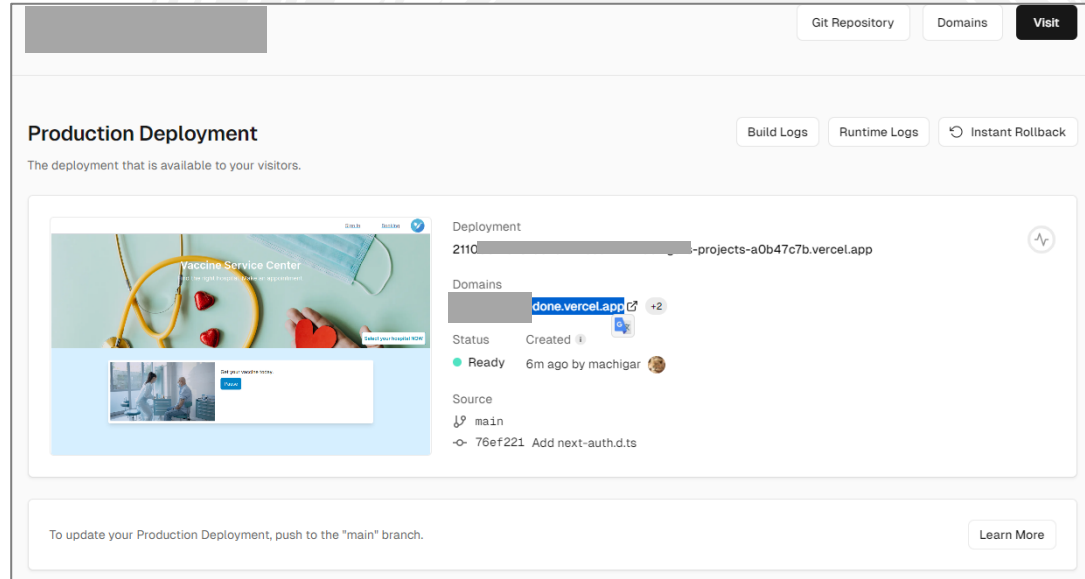
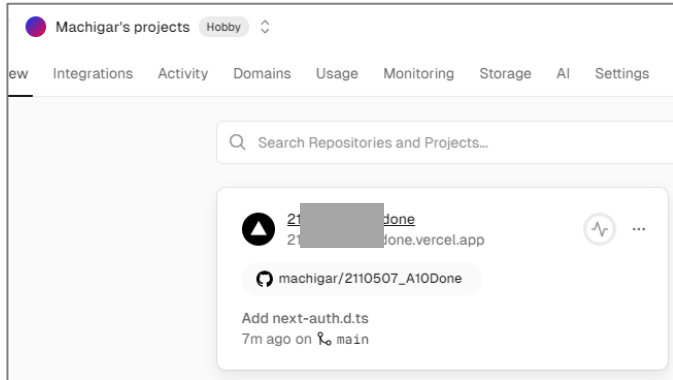
TIP: Paste a .env above to populate the form

Learn more about [Environment Variables](#)

Deploy

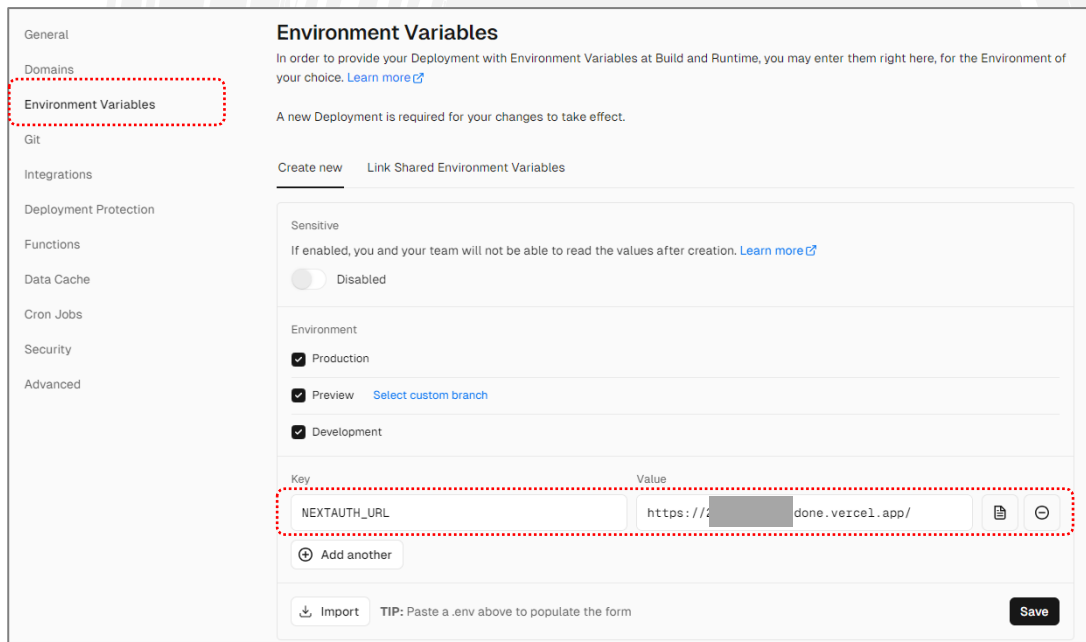
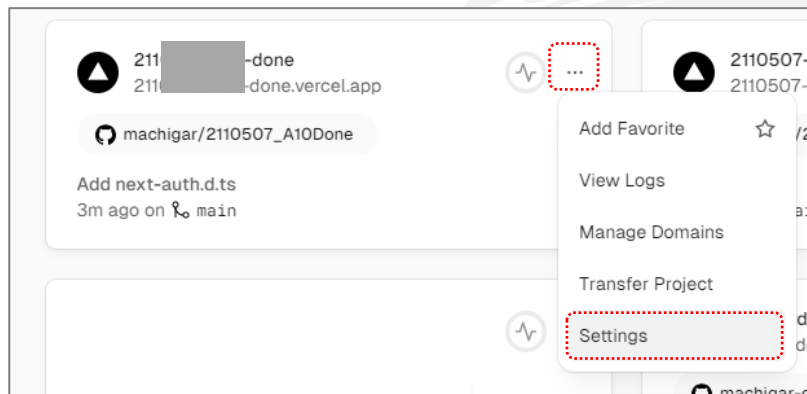
# Deploy to Vercel

- Deploy เว็บไซต์ขึ้น Vercel จนเสร็จ และ add domain
- Copy domain URL ที่ได้



# Deploy to Vercel

- ไปที่ Deployment ของเว็บไซต์
- เลือกปุ่มเมนู ... > Settings
- เลือก Section Environment Variables
  - เพิ่ม NEXTAUTH\_URL กำหนดค่าเป็นค่า domain ที่ copy มา



# Deploy to Vercel

## ■ Redeploy Project

- ไปที่ Deployments เลือก Deployment ล่าสุด
- เลือก Redeploy และยืนยันการ Redeploy to Production

The screenshot shows the Vercel interface for a project named 'Machigar's projects'. The 'Deployments' tab is active. Below the 'Deployments' header, there's a status bar indicating 'Continuously generated from machig... Done'. A search bar and filters for 'All Branches...', 'Select Date Range', 'All Environments', and 'Status (4/5)' are present. The deployment list shows two entries: 'nxleoenzz' (Ready, 1m 25s (3m ago), main branch) and 'e3q0xf1gs' (Error, 58s (20m ago), main branch). The 'nxleoenzz' deployment is highlighted with a red dashed box. A context menu is open for this deployment, showing options like 'Instant Rollback', 'Promote', 'Redeploy', and 'Inspect Deployment'. The 'Redeploy' option is highlighted with a red dashed box.

## Redeploy to Production

You are about to create a new Deployment with the same source code as your current Deployment, but with the newest configuration from your Project Settings.

The following Domains will point to your new Deployment:

21105...ne.vercel.app

☐ Use existing Build Cache

[Learn more about caching](#)

Cancel

Redeploy