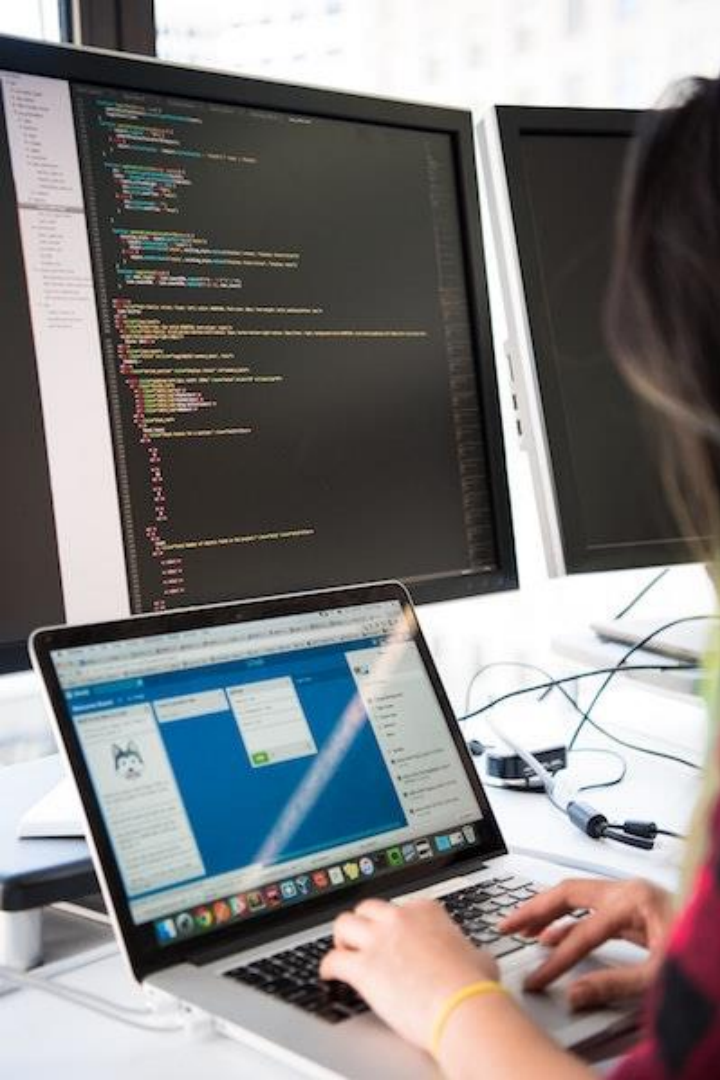


Important Instructions

- การส่ง Assignment ผ่าน Github Classroom
 - งานใน Assignment 08 เป็นงานต่อจาก Assignment 07 แต่ใช้ Repository เริ่มต้นที่ให้มา ผ่าน Assignment Invitation Link สำหรับ Assignment 08
 - 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



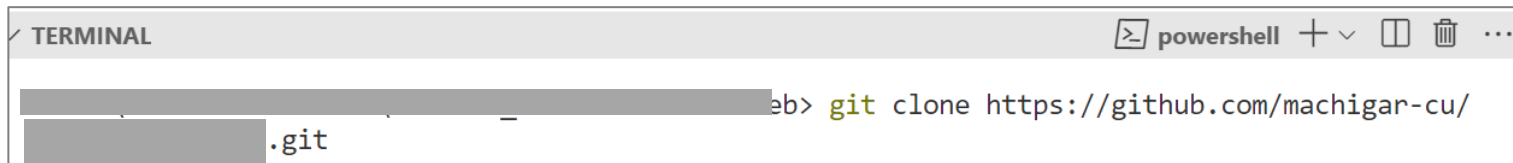
Assignment 8: React Hooks

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 (`a08-swdevprac2-xxxxxx`)
ใน Visual Studio Code

- จะพบไฟล์ตั้งต้นของ project

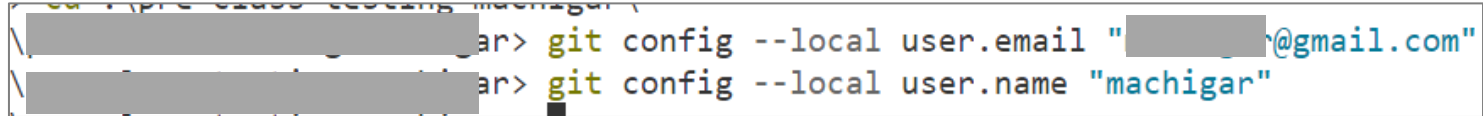
Prepare Class Environment



- 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 "[redacted]@gmail.com"  
machigar> git config --local user.name "machigar"
```

- ติดตั้ง node packages `npm install`
- Copy ไฟล์เตอร์ `public` และ `src` จาก project ของ Assignment 07 มาทับ ไฟล์เตอร์ `public` และ `src` ใน project ของ Assignment 08 นี้
- ลอง run project จะได้ project ที่ทำไว้ใน Assignment 07 `npm run dev`

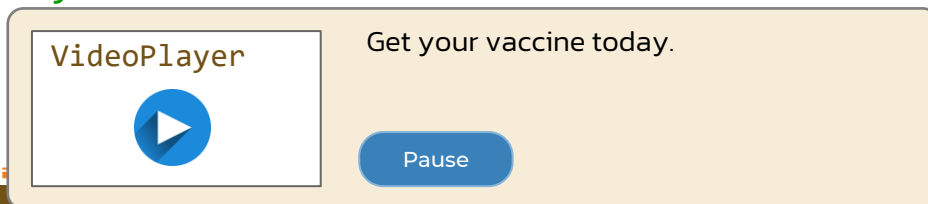


เพิ่มการใช้งาน `useRef` และ `useEffect` เพื่อวาง `video` ไว้บนหน้าแรกของ `vaccine-book-app`

- สร้าง component `src\components\VideoPlayer.tsx` บรรจุ `video` element สำหรับแสดง `video` โดยรับ Props ดังต่อไปนี้
 - `videoSrc` เป็น string เพื่อใช้รับ path ไปยังไฟล์ `video` ที่จะแสดง
 - `isPlaying` เป็น boolean เพื่อใช้รับ สถานะ `play/pause`
 - ให้ export module นี้เป็น default export
- `VideoPlayer` แสดง `video` จาก path ที่รับเข้ามา พร้อม `play/pause` `video` ตามสถานะที่รับเข้ามา โดยใช้ `useEffect` ในการกำหนดการ `play/pause` โดยให้ effect run เมื่อ render ครั้งแรกและเมื่อค่าสถานะ `play/pause` เปลี่ยน
- สร้าง component `src\components\PromoteCard.tsx` เพื่อจะนำไปวางไว้ที่หน้าแรก
 - ให้ component `PromoteCard` บรรจุ `VideoPlayer`
 - ให้ export module นี้เป็น default export

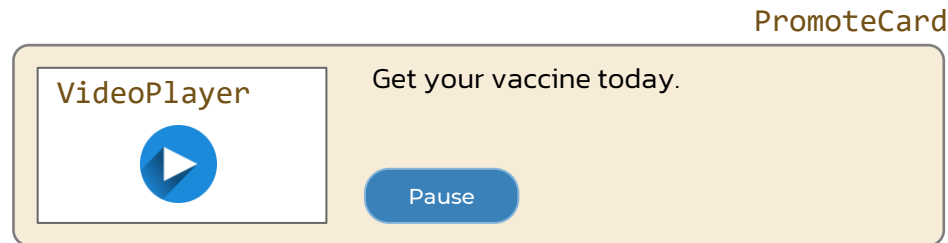
```
export default function VideoPlayer({
```

```
export default function PromoteCard() {
```





- นำไฟล์ `getvaccine.mp4` ไปวางที่ `public\vdo\getvaccine.mp4`
- สำหรับ component `src\components\PromoteCard.tsx`
 - ให้ component `PromoteCard` บรรจุ `VideoPlayer`
 - `PromoteCard` ส่ง path ไปยังไฟล์ video `getvaccine.mp4` ไปให้เป็น props `vdoSrc` ของ `VideoPlayer` และส่ง props `isPlaying` ซึ่งสถานะตั้งต้นของการเล่น video เป็น true
 - `PromoteCard` บรรจุปุ่ม 1 ปุ่ม สำหรับกำหนดสถานะ play/pause ให้กับ `VideoPlayer`
 - เก็บสถานะการ play/pause ไว้ใน state variable
 - หาก vdo เล่นอยู่ให้แสดงข้อความบนปุ่มว่า Pause และเมื่อกดปุ่มจะเป็นการหยุด vdo
 - หาก vdo หยุดอยู่ให้แสดงข้อความบนปุ่มว่า Play และเมื่อกดปุ่มจะเป็นการเล่น vdo





- สร้าง Custom Hook ชื่อ `src\hooks\useWindowListener.tsx` สำหรับฟัง window event ซึ่งรับ props
 - `eventType` เป็น string เพื่อใช้รับชนิดของ event ที่จะฟัง
 - `listener` เป็น EventListener ที่จะทำงานเมื่อมี event ดังกล่าวเกิดขึ้น
- ใช้ `useEffect` ในการ add listener โดยให้ทำงานเฉพาะเมื่อ render ครั้งแรก และมี clean up function สำหรับ remove listener
- ใน component `PromoteCard` เรียกใช้ `useWindowListener` เพื่อระงับการคลิกขวาบน window ที่บรรจุ `PromoteCard` เพื่อไม่ให้แสดง context menu เมื่อคลิกขวา โดย
 - ฟัง event `contextmenu`
 - ให้ listener เรียก `preventDefault()` บน event object เพื่อระงับการทำ default action ซึ่งก็คือการแสดง context menu
- **Note:** ใน project นี้ ไม่มี window listener อื่น นอกเหนือจากที่กำหนดใน `useWindowListener.tsx`

Assignment test cases

Assignment 08 มี 4 test cases (รวม autograde 5 คะแนน)

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

Development & Deployment

- ใช้แนวทางการ Deploy แบบเดียวกับใน Assignment A03 ถึง A07
 - สร้าง Repository บน Personal GitHub Account สำหรับ Assignment A08
 - 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