HIS on Website

**Requirement**

สร้างหน้า Frontend ที่สามารถทำงานบนเว็บ Browser ได้ โดยที่แบบเดิมทำงานบน Windows

**Technical Requirement**

OS ที่ใช้ในการพัฒนา

* Windows 10 Pro

Editor

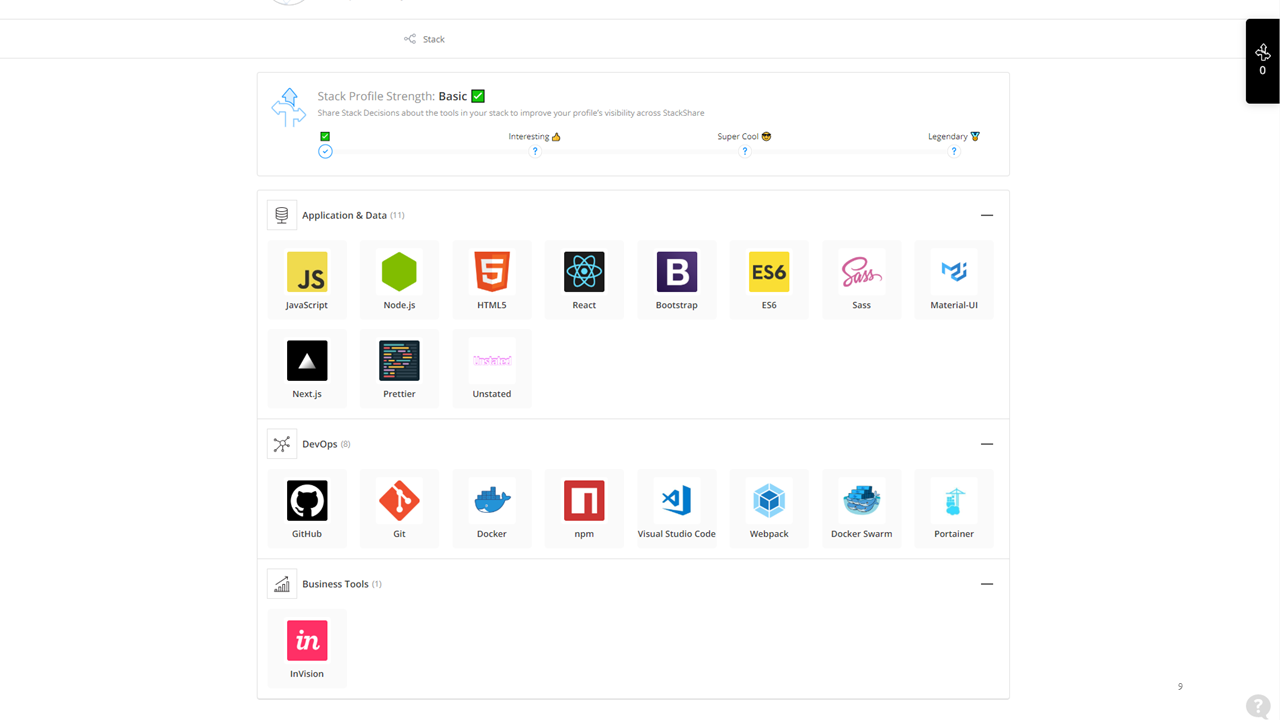
* Visual Studio Code

Development Language

* HTML
* JavaScript

**เครื่องที่ใช้ในโปรเจค**

(เครื่องมือบางอย่างอาจจะไม่ได้ใช้ในโปรเจค เนื่องจากถอดออกกระทันหัน)

****

1.JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น

2. NodeJS คือ JavaScript มีการ Complied เป็น Byte Code ด้วย

3. HTML คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ Tag ในการกำหนดการแสดงผล HTML

4. React เป็น JavaScript Library ที่ถูกสร้างโดย Facebook React เป็นแค่ UI โดยสร้างมาจากพื้นฐานแนวความคิดแบบ MVC(Model View Controller) ซึ่งหมายถึงว่า React มีหน้าที่จัดการกับ Model หรือ View แต่ส่วนใหญ่จะเป็น View

5. Bootstrap คือ Frontend Framework ที่รวม HTML, CSS และ JS เข้าด้วยกันสำหรับพัฒนา Web ที่รองรับทุก Smart Device

6. SASS คือ CSS Preprocessor ซึ่งทำงานเหมือนกับ LESS CSS หลักการเขียนจะคล้าย ๆ กับ CSS ทั่วไป แต่มีฟังก์ชั่นเพิ่มขึ้นมาช่วยให้เขียนได้ง่ายขึ้น โดยหลังจากเขียนเสร็จแล้วก็ Compile เป็นไฟล์ CSS ปกติเพื่อเอาไปใช้งานจริงได้

7. Next.js เปรียบเสมือนอีก Version นึงของ create-react-app คือ Config อะไรให้หมด

8. Prettier เป็น code formatter โดยตัว Prettier จะดูโค้ดว่า ส่วนนี้ยาวเกินไปหรือเปล่า ดูเรื่อง space/tab หรือ spacing ก่อน comma หรือ keyword และตาม Styleguide ด้วย Plugin มันก็จะ format code ให้ตามที่เราตั้งค่าไว้ ซึ่งรองรับภาษาต่างๆ JavaScript (ES6+) ,React (JSX),Flow,TypeScript,CSS, CSS,JSON,GraphQL,Markdown

9. GitHub คือ website Git (version control repository) ที่อยู่บน internet มีการทำงานแบบเดียวกับ Git เลย แต่สามารถเข้าถึงข้อมูลและจัดการไปผ่าน web

10. Git คือ Version Control แบบ Distributed ตัวหนึ่ง เป็นระบบที่ใช้จัดเก็บและควบคุมการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ชนิดใดก็ได้ ไม่ว่าจะเป็น Text File หรือ Binary File

11. Docker คือ engine ตัวหนึ่งที่มีการทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง server เพื่อใช้ในการ run service ที่ต้องการ

12. VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด

13. Webpack คือ เครื่องมือที่ใช้ในการรวม module ที่เราเขียน ให้เป็นไฟล์ที่เราจะนำไปใช้งานจริงๆ ความสามารถหลักๆของมันมีดังนี้ครับ รวม module หลายๆ module ให้กลายเป็นไฟล์ๆเดียว

14. Docker swarm เป็น Native Clustering ของ Docker คือเป็นเครื่องมือช่วยจัดการเครื่อง server ที่รัน Docker หลายๆเครื่องให้อยู่ในสภาพแวดล้อมเดียวกัน

**วิธี Run project**

**```**

**npm i**

**npm run dev**

**```**

เปิดบราวเซอร์ : <http://localhost:3000/>

**หากต้องการใช้ Docker run project**

**```**

docker build -t react-docker-app .

docker run -d -p 3000:3000 react-docker-app

```

เปิดบราวเซอร์ : <http://localhost:3000/>

**Stop and remove all containers**

```

docker stop $(docker ps -a -q)

docker rm $(docker ps -a -q)

```

**Remove all images**

```

docker rmi $(docker images -a -q)

```

**หากผู้ที่ไม่เคยเขียน React เลย**

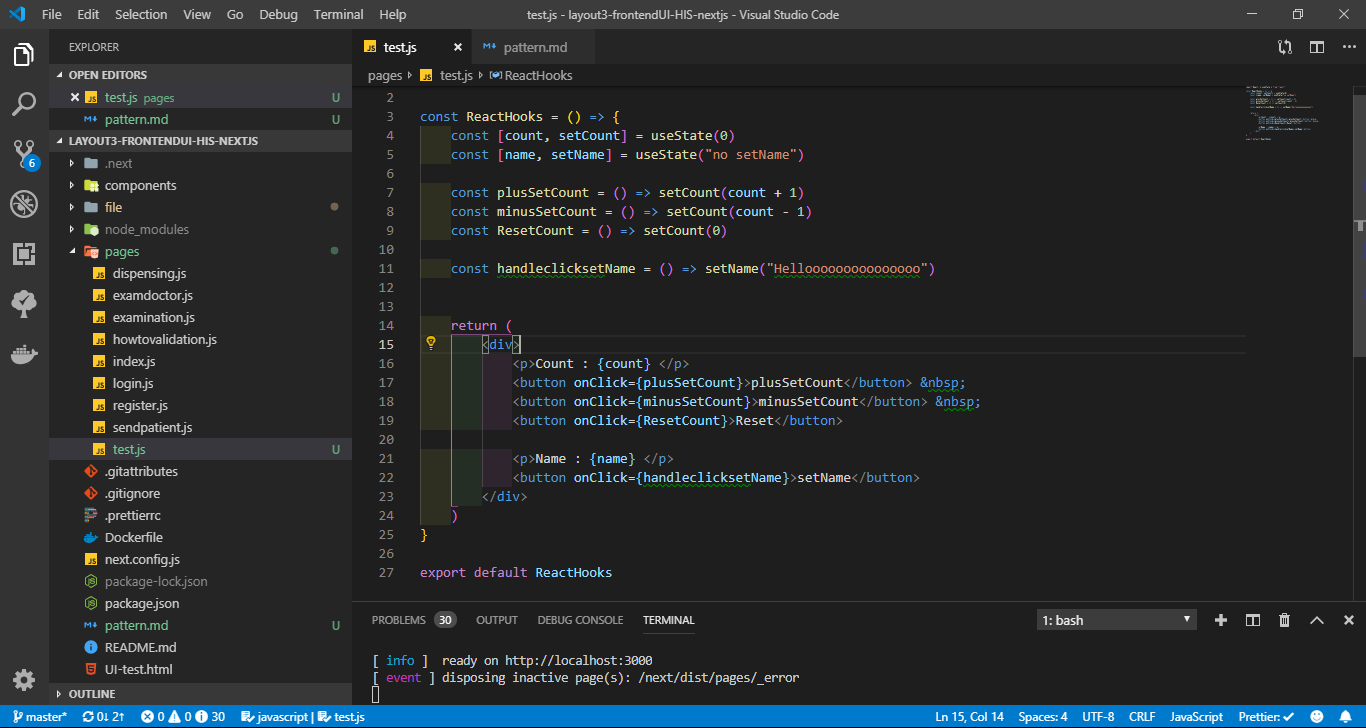
ไฟล์ pattern.md จะเป็น Pattern ในการเขียน React ซึ่งมี 3 Pattern ด้วยกัน

1. class components
2. functional components
3. react hooks

**ตัวอย่าง เช่น**

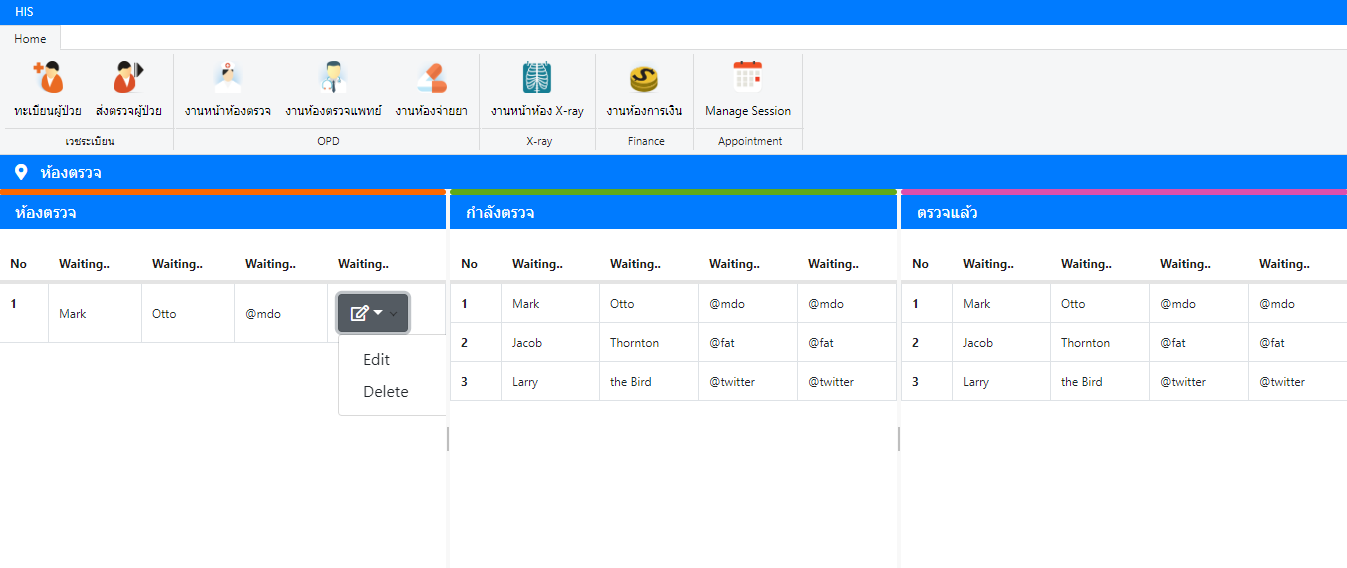
ลอง Copy code ในไฟล์ pattern.md และ ทำการสร้างไฟล์ test.js ภายใต้โฟลเดอร์ pages

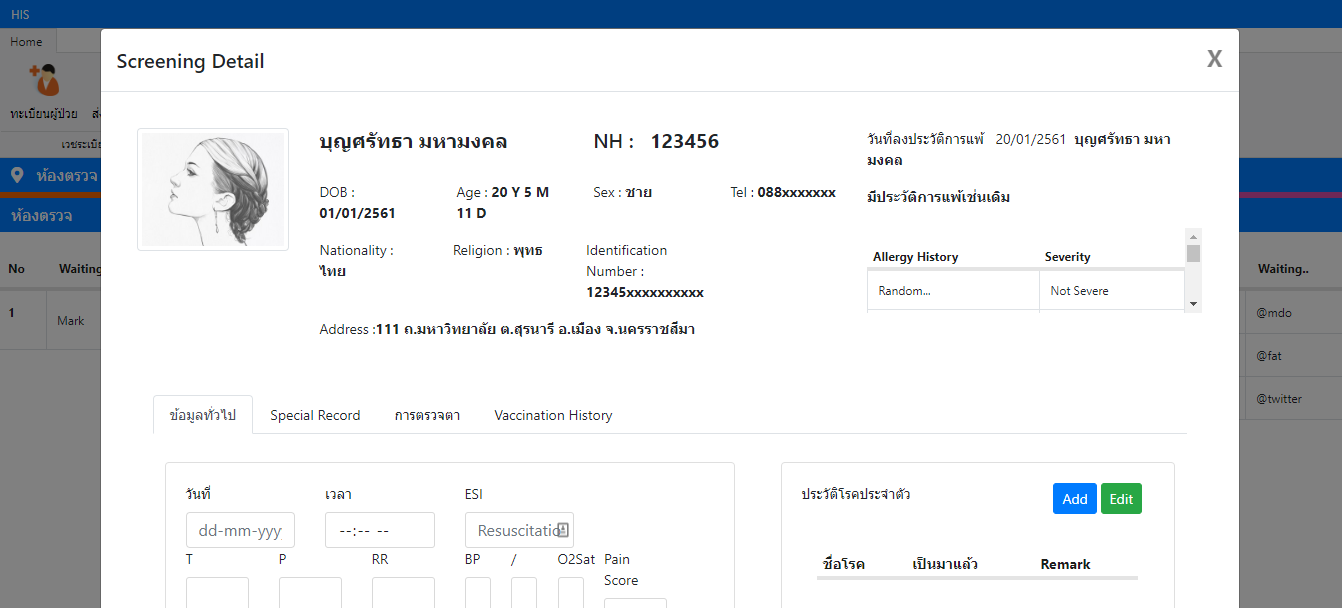
เปิดบราวเซอร์ : <http://localhost:3000/test>



**โฟลเดอร์ components/screening-detail-feature จะเป็นโฟลเดอร์ ที่รวม components ของ feature screening-detail**

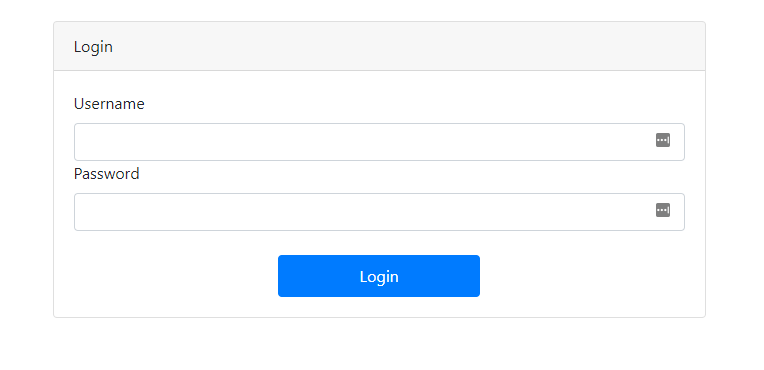
เปิดบราวเซอร์ : <http://localhost:3000/examination> จากนั้นคลิกที่ Edit ก็จะพบ



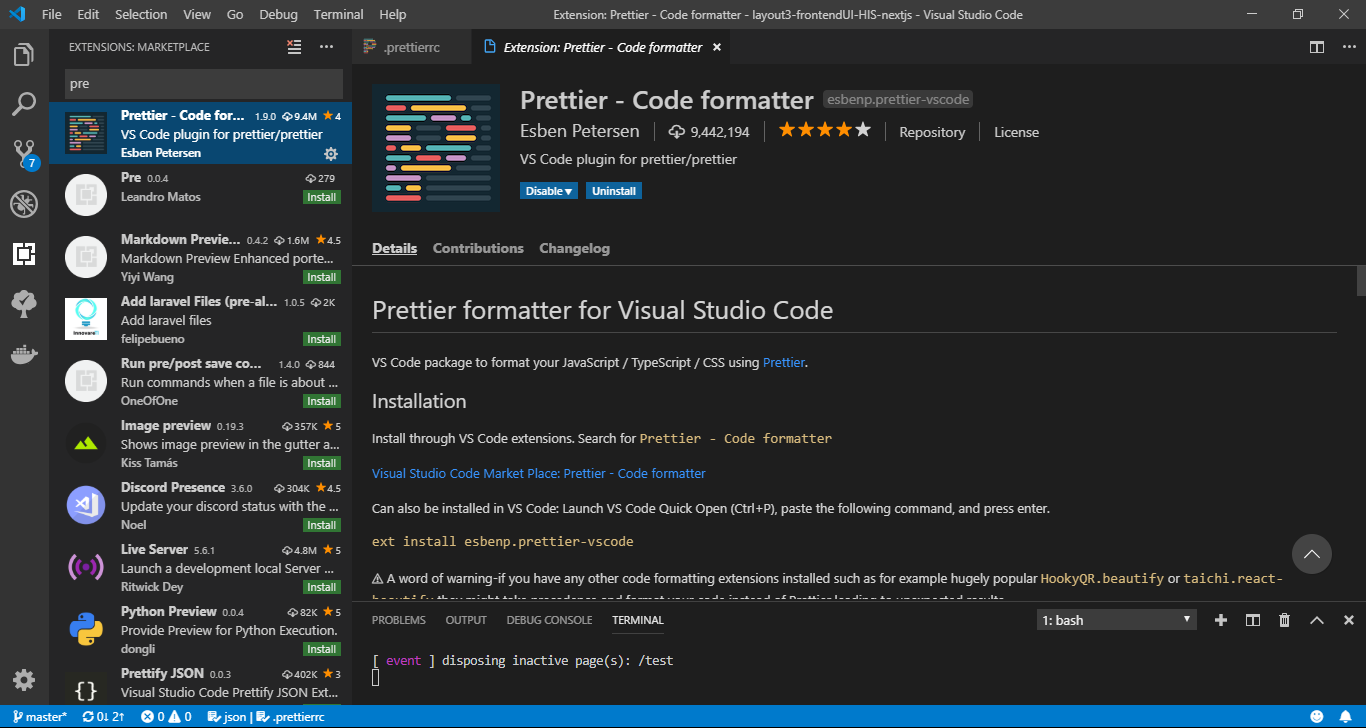


**โฟลเดอร์ components/login**

เปิดบราวเซอร์ : <http://localhost:3000/login>



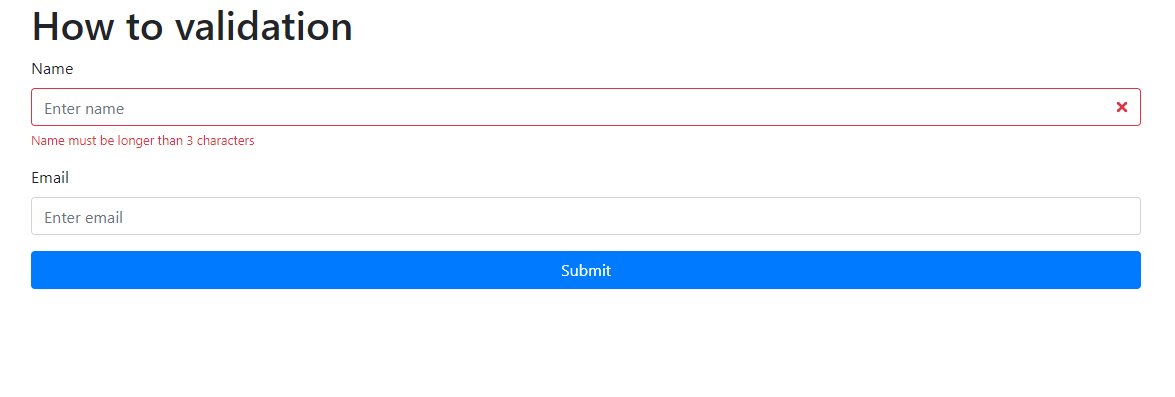
**ไฟล์ .prettierrc จะเป็นมาตราฐานของการกำหนด style การเขียน code ให้ไปในรูปแบบเดียวกัน**



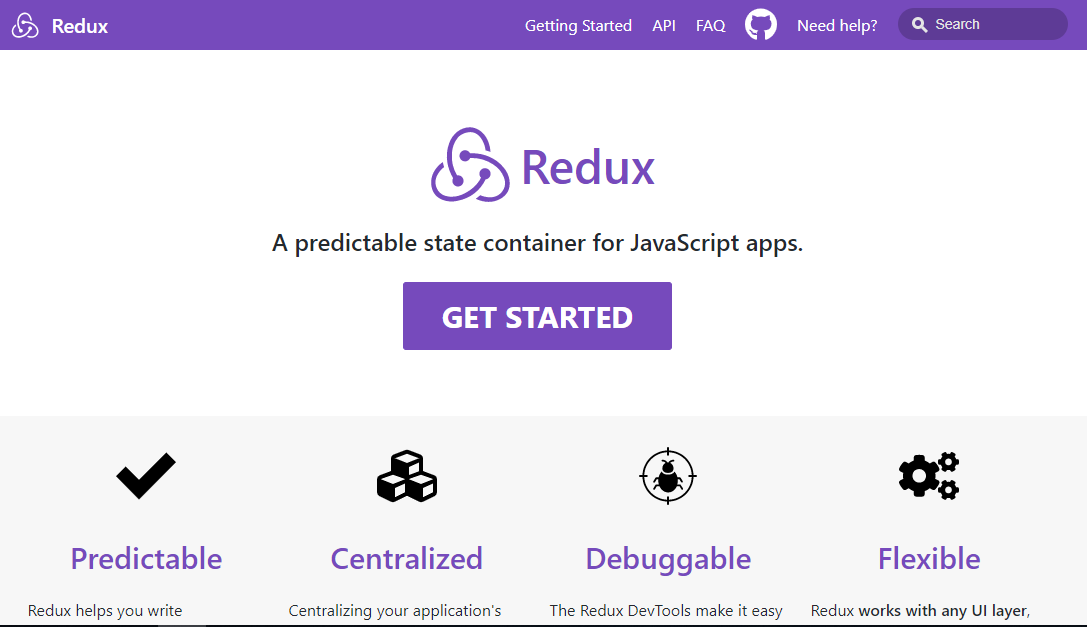
**กด Alt + Shift + F เพื่อใช้งาน**

**ไฟล์ pages/howtovalidation.js เป็น ไฟล์ตัวอย่างสำหรับการดักข้อมูลที่ผู้ใช้งานกรอกข้อมูลผิดพลาด**

เปิดบราวเซอร์ : <http://localhost:3000/howtovalidation>



**เครื่องที่จะต้องนำเข้า เพื่อทำโปรเจคต่อไป คือ Redux**



**เป็นตัวเก็บ state ของ react**

**ประโยชน์ของมันคือ เก็บ state เอาไว้ที่เดียว**

**State = ตัวแปร**

**-------------------------------------------------------------**

**สมมุติ ระบบ Login**

**หากจะเข้า หน้า** <http://localhost:3000/homepage> **ได้**

**ก็จะต้องเข้าหน้า** <http://localhost:3000/login> **ก่อนเท่านั้น**

**หากอยู่ดีๆ เข้า** <http://localhost:3000/homepage> ทันที

ตัวโปรแกรมก็จะทำการ เด้งกลับมาหน้า <http://localhost:3000/login> ทันที