

# Web App + Web API Final Project Report 261207

## Members

นาย จิรายุ เกษมสุข	600612149
นาย สพล โกษาวัง	610610619
นาย เขม มหามาตร	610610696

## Book Web App

**App** สำหรับการเก็บข้อมูลหนังสือที่ผู้สนใจโดยมีระบบลงชื่อเข้าใช้และระบบเพิ่มผู้ใช้ใหม่

การแบ่ง component และ endpoint ในส่วนของการแสดงผล

**Component** จะแบ่งออกเป็น 6 หน้าหลักคือ

1.forms จะแบ่งออกเป็น LoginForm.js สำหรับการลงชื่อเข้าใช้ และ SignupForm.js สำหรับการเพิ่มผู้ใช้ใหม่โดยทั้งสองจะทำงานภายใต้ LoginPage.js และ SignupPage.js

2.messages สำหรับการแจ้งเตือนเมื่อ การลงชื่อเข้าใช้ หรือ การลงทะเบียนเพิ่มผู้ใช้ใหม่เกิดข้อผิดพลาด โดยจะ แสดงข้อความออกมาจาก InlineError.js

3.navigation เป็น navigation bar สำหรับที่ผู้ใช้ที่ลงทะเบียนแล้วโดยจะแสดงอยู่ด้านบนของ pages สามารถ logout และ กลับไปที่ Dashboard page ได้

4.pages รวมหน้าเว็บต่างๆ ของ web app จะประกอบไปด้วย DashboardPage.js Homepage.js LoginPage.js NewBookPage.js และ SignupPage.js

5.routes จะเป็นส่วนกำหนด endpoint ของ web app ว่า ใครจะสามารถเข้าส่วนไหนของ webpage ดังเช่น guestRoute.js จะเป็นส่วนกำหนดว่าผู้ที่ยังไม่ได้ลงทะเบียนจะเข้าถึงได้เฉพาะในส่วนของ homepage และ signup page ส่วน UserRoute.js จะกำหนดให้กับผู้ที่มีบัญชีการใช้งาน เข้าถึง dashboard ข้างใน และหน้าเว็บสำหรับเพิ่มข้อมูลหนังสือ

6.ctas จะมี menu ปุ่มของการ add book เป็นส่วนประกอบในหน้าของ "Dashboard Page"

## อื่นๆนอกเหนือจาก Component คือ

actions จะทำหน้าที่นำคำสั่งที่สามารถติดต่อกับ api จาก api.js มาใช้งาน โดยแบ่ง เป็น auth.js และ users.js

auth.js จะเป็นตัวควบคุมการ login ของผู้ใช้งานโดยจะ ส่ง req ด้วย credentials ให้กับ api เพื่อไปตรวจสอบ user ใน mongodb จากนั้น api จะ res user data กลับมา ถ้าการ login สำเร็จ จึงสามารถเก็บ user token มาไว้ใน local storage ได้ และ ให้ dispatch user data ไปใน userLoggedIn เพื่อให้ web app ทราบว่ามีผู้ใช้งานในระบบแล้ว(userLoggedIn และ LoggedOut เป็นฟังก์ชัน เพื่อเปลี่ยน type ให้เป็นสถานะต่างๆ) และนำไปทำงานร่วมกับ reducer เพื่อ เปลี่ยน stage ของ web app ต่อไป ทำนองเดียวกันกับการ logout จะนำ user token ใน local storage ออก และ ให้ เปลี่ยน type ใน user logged out เพื่อไม่ให้มี ผู้ใช้ในระบบ

users.js จะนำคำสั่งจาก api.js มาใช้ในส่วนของการ signup โดยจะให้ api ส่ง req data เพื่อไปบันทึกผู้ใช้งาน ใน database ถ้าสำเร็จแล้ว api จะส่ง response กลับมาเพื่อให้ app เก็บ user token ลงใน local storage และ dispatch userLoggedIn เพื่อเปลี่ยน stage ของ app ต่อไป

reducers จะแบ่ง books.js และ user.js ในที่นี้อธิบายในส่วนของการ user.js เนื่องจาก books.js ยังไม่สามารถทำงานได้สมบูรณ์ user.js จะทำหน้าที่เปลี่ยน stageต่างๆ ของ app โดยทำงาน ร่วมกับ types.js ที่ต้องทำงานร่วมกันกับ auth.js และ users.js

วิธีการจัดเก็บ Token ที่ได้รับจาก API : เก็บไว้ใน Local Storage และนำไปใช้งาน ใน index.js เพื่อให้ app ทราบว่ามีผู้ใช้งานในระบบ

**For Testing registered email: cpe@gmail.com password: 12345**

## สิ่งที่ได้เรียนรู้และต้องนำไปศึกษาเพิ่มเติม

การจัดการ stage ของ web app ด้วย redux ประกอบกับการใช้ reducer รวมไปถึงการใช้ prop-types เพื่อมาช่วยเป็นเงื่อนไขที่จะเข้าถึง stage ต่างๆ และการใช้งาน redux-thunk

\*หมายเหตุ เนื่องจากผู้จัดทำยังไม่สามารถทำให้ระบบทำงานได้อย่างราบรื่นและสามารถใช้งานได้จริงได้อย่างสมบูรณ์ จึงขอเก็บไว้เป็นการเรียนรู้เพื่อนำไปพัฒนาในครั้งต่อไป อีกทั้งในท้ายรายงานจะมีสรุปปัญหาที่เกิดขึ้นจากการทำโปรเจกต์ครั้งนี้

## Book API

### รายการ Endpoint ทั้งหมด

(From react app)	(From API)
/login - ระบุที่อยู่ของหน้า loginpage เพื่อแสดง loginform	/api/auth – ใช้สำหรับการ login และเช็คบัญชีผู้ใช้งานว่าตรงกับในdatabase หรือไม่
/signup – ระบุที่อยู่ของหน้า signuppage เพื่อแสดง signupform	/api/users – ใช้สำหรับการ signup และ บันทึกข้อมูลบัญชีผู้ใหม่ใน database
/about – ที่อยู่ของหน้า About	/api/books/ - ใช้เพื่อเข้าไปยังส่วนฐานข้อมูลของหนังสือ
../books/new – ระบุที่อยู่ของหน้า newbookpage เพื่อแสดงเมนูการเพิ่มหนังสือ	/api/books/search – ใช้เพื่อค้นหาหนังสือที่สนใจใน database
../dashboard – ระบุที่อยู่ของหน้า dashboardpage เพื่อแสดงลิงก์ไปยังหน้าเพื่อหนังสือ	

### Endpoint ที่ต้องผ่าน authentication จึงจะสามารถใช้งานได้

(From react app)	(From API)
../books/new ../dashboard	/api/auth /api/books/ /api/books/search

### สิ่งที่ทำไม่สำเร็จ/ปัญหาที่เกิดขึ้น

- 1.Web App ไม่สามารถเก็บข้อมูลหนังสือได้ เนื่องจากยังไม่สามารถเชื่อมและดึงข้อมูล จากเว็บข้อมูลหนังสือภายนอกได้ และ ไม่ได้ทำ ส่วนของเก็บข้อมูลผู้ใช้ ที่สามารถ แก่ ลบ เพิ่ม ได้
- 2.ในบางส่วนของเว็บมีปัญหาที่เกิดจากการใช้ **package** ที่อาจทำงานซ้อนทับกัน อย่างเช่น **react-router** กับ การทำงานของ **redux** อาจทำให้บาง **component** ทำงานไม่ได้อย่างสมบูรณ์

### Reference

Rem Zolotych (2017/8/31). Build Real Web App with React. Retrieved from <https://www.youtube.com/watch?v=NO2DaxhoWHk&t=3s>