

# รายงาน

เรื่อง ความคืบหน้าโครงการ 1

สาขาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยธรรมศาสตร์



## เรื่อง

ร้านค้ามหาวิทยาลัยธรรมศาสตร์ ออนไลน์

TU Shop Online

อาจารย์ที่ปรึกษา

อ.ดร. ชุมพล บุญมี

## ผู้จัดทำ

นายสุเมต คงแก้ว 6210612559

## 1) คำโครงของโครงการ

### 1.1) ที่มา

ที่มาของโครงการนี้คือมาจากการที่นักศึกษาภายในมหาวิทยาลัยธรรมศาสตร์ที่ความคิดอยากจะเริ่มธุรกิจขายสินค้าและบริการเล็ก ๆ แต่ไม่รู้ว่าจะเริ่มที่ไหนดี หากเริ่มที่เว็บไซต์อีคอมเมิร์ซใหญ่ ๆ เช่น Shopee หรือ Lazada เป็นต้น อาจจะสู้ร้านค้าที่ใหญ่ ๆ ที่มีต้นทุนเยอะกว่าไม่ได้ จนทำให้อาจจะไม่เป็นที่น่าซื้อหรือน่าสนใจของผู้ซื้อ และมหาวิทยาลัยธรรมศาสตร์ขณะนี้ยังไม่มีเว็บไซต์แบบอีคอมเมิร์ซสำหรับนักศึกษาโดยเฉพาะ

### 1.2) เป้าหมาย

เป้าหมายของโครงการนี้ ผู้จัดทำต้องการสร้างเว็บไซต์ที่เป็นรูปแบบเว็บอีคอมเมิร์ซเพื่อต้องการให้นักศึกษาภายในมหาวิทยาลัยธรรมศาสตร์ที่อยากจะเริ่มขายสินค้าหรือบริการได้มีพื้นที่ไปทบทธุรกิจของตนเองให้เป็นที่รู้จักได้ โดยให้เป็นที่รู้จักภายในมหาวิทยาลัยก่อน โดยวางขายสินค้าและบริการ ที่เป็นรูปแบบเว็บไซต์อีคอมเมิร์ซของโครงการนี้

### 1.3) ขอบเขต

ขอบเขตการดำเนินงานในการพัฒนาและออกแบบเว็บไซต์ซื้อขายสินค้า จะเป็นการทำงานในลักษณะ Web Application โดยมีขอบเขตการทำงานดังนี้

#### 1.3.1) ระบบ Authentication

#### 1.3.2) ระบบการซื้อและวางขายสินค้าในเว็บไซต์โดยใช้ฐานข้อมูลในการเก็บข้อมูล

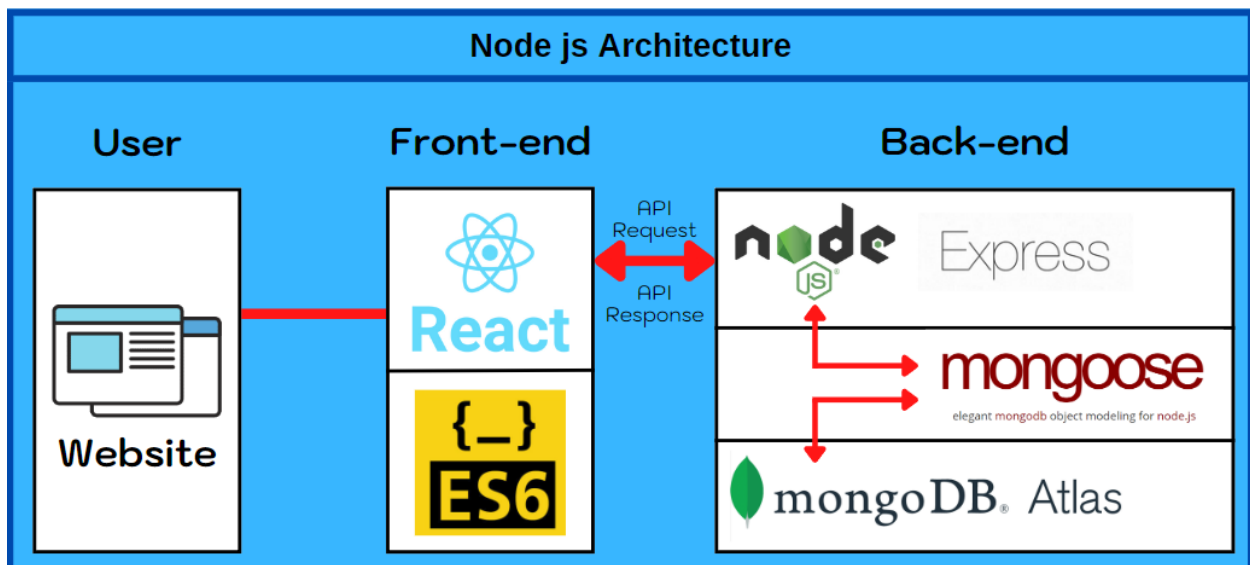
#### 1.3.3) ระบบ Chatting

## 2) ผลการดำเนินงาน

ก่อนดำเนินงานจะมีการวางแผนก่อนที่โดยจะมีการสำรวจเลือกโปรแกรมที่ใช้ในการสร้างส่วน Front-end, เลือก Framework ที่จะใช้ในการทำส่วน Back-end และเลือกฐานข้อมูลใช้สำหรับการเก็บข้อมูลทั้งหมด

โปรแกรมและทรัพยากรที่พบเจอว่าสามารถใช้ในการทำโครงการได้ ในส่วนของ Front-end จะมี ReactJS, VueJS, NextJS และ AngularJS ส่วนของ Back-end จะมี NodeJS Django และในส่วนของฐานข้อมูล จะมีเป็นแบบ SQL เช่น MySQL, PostgreSQL, MariDB และฐานข้อมูลแบบ NoSQL เช่น MongoDB, Redis, Firebase

หลังจากวิเคราะห์เสร็จสิ้น จะมาสู่การตัดสินใจเลือกโปรแกรมหรือทรัพยากรที่จะใช้ในการทำโครงการ โดยทาง Front-end เลือกใช้ ReactJS เพราะจะทำให้หน้าเว็บไซต์ประมวลผลเร็วขึ้น และจะมีการแบ่งส่วนภายในเว็บไซต์เป็นส่วนย่อย ๆ ได้อย่างชัดเจน เรียกว่า Component ส่วน Back-end เลือกใช้ NodeJS เป็น Environment และ ExpressJS เป็น Framework เพราะ NodeJS ใช้ภาษาเดียวในการพัฒนาก็คือ Javascript ซึ่ง Javascript ก็ใช้กับ React เช่นกัน ส่วนของฐานข้อมูลจะเลือกเป็นฐานข้อมูลแบบ NoSQL MongoDB เนื่องจากฐานข้อมูลแบบ NoSQL มีความยืดหยุ่นที่สูงกว่า SQL และ MongoDB เก็บข้อมูลรูปแบบไฟล์ JSON



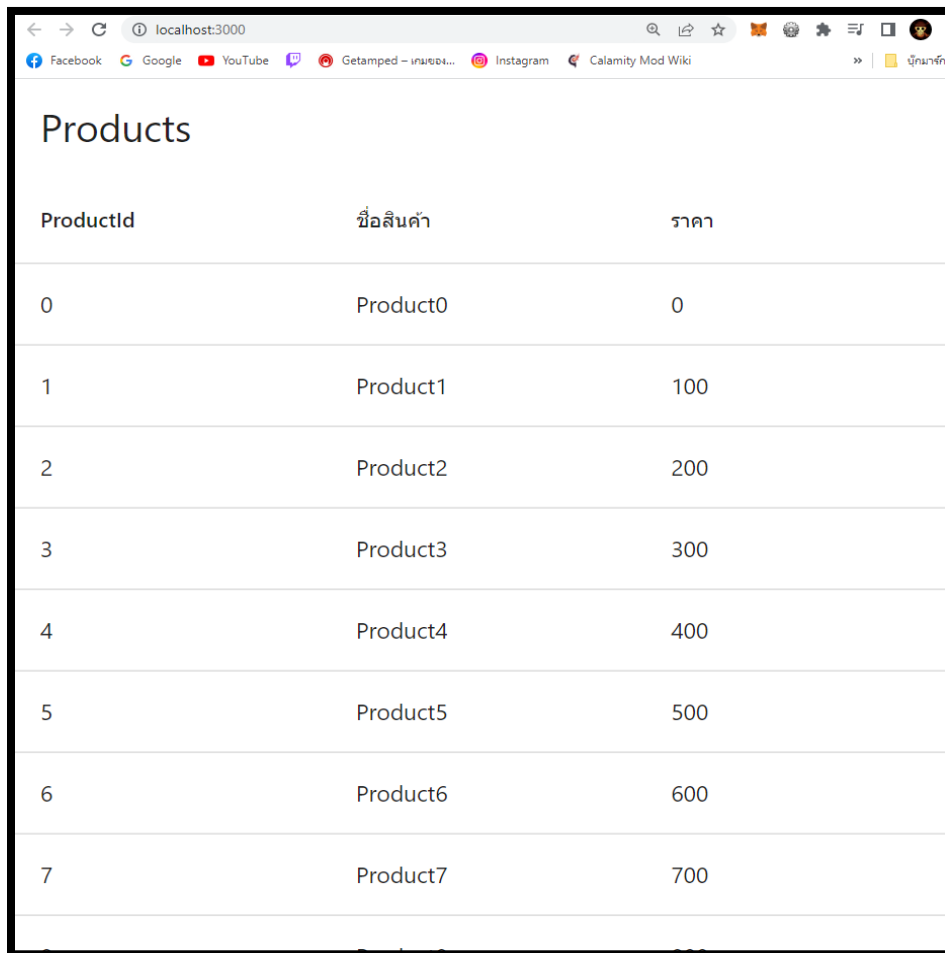
รูปภาพที่ 1 แสดงแผนผังโครงสร้างสำหรับการทำโครงการ

จากการดำเนินงานที่กำหนดไว้ จะแบ่งการดำเนินงานออกเป็น 2 ส่วน ดังนี้

2.1) Front-end โดยจะใช้ ReactJS ซึ่งเป็น Javascript Library มีไว้สำหรับแสดงหน้าเว็บไซต์ ในส่วนนี้มีการดำเนินการที่ทำได้แล้ว ดังนี้

2.1.1) มีการใช้ HTTP Request อย่าง Axios ซึ่งเป็น Library นำมาใช้ในการ Fetch API จากฝั่ง Back-end ทำให้สามารถดึงหรือจัดการข้อมูลจากฐานข้อมูลได้

2.1.2) ดึงข้อมูลจาก API ฝั่ง Back-end แล้วนำมาแสดงผลบนหน้าเว็บไซต์



The screenshot shows a web browser window at localhost:3000. The page title is 'Products'. Below the title is a table with three columns: 'ProductId', 'ชื่อสินค้า', and 'ราคา'. The table contains 8 rows of data, with ProductId ranging from 0 to 7 and prices ranging from 0 to 700.

ProductId	ชื่อสินค้า	ราคา
0	Product0	0
1	Product1	100
2	Product2	200
3	Product3	300
4	Product4	400
5	Product5	500
6	Product6	600
7	Product7	700

รูปภาพที่ 2 แสดงการขอข้อมูลจาก MongoDB atlas แล้วแสดงที่หน้าเว็บไซต์

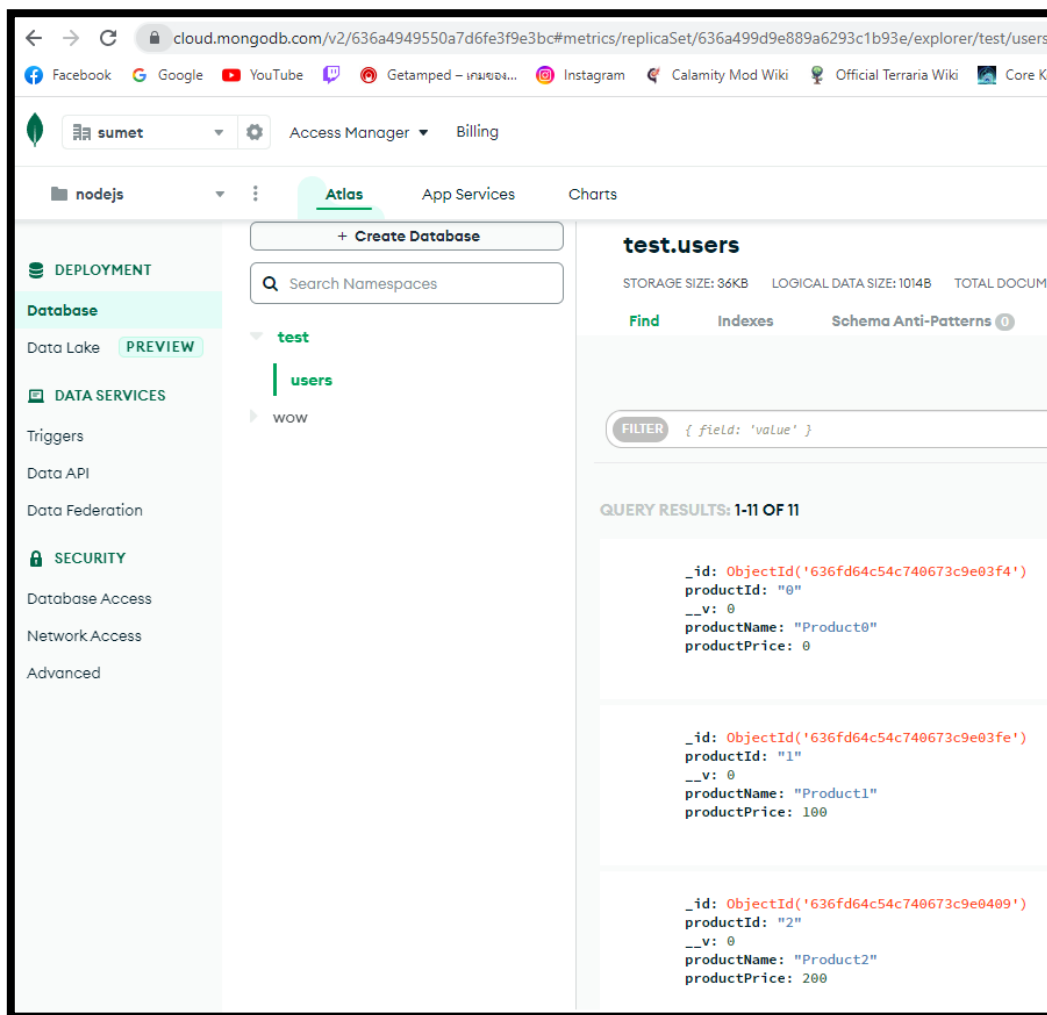
2.2) Back-end โดยจะใช้ NodeJS ซึ่งเป็น Environment ที่ใช้ภาษา Javascript เช่นเดียวกับ ReactJS โดยจะให้ Back-end ทำหน้าที่เป็น API สำหรับส่งข้อมูลให้กับทางฝั่ง Front-end

ในส่วนนี้มีการดำเนินการที่ทำไปแล้ว ดังนี้

2.2.1) เปิดเซิร์ฟเวอร์ ได้มีการติดตั้งและทดลองเปิดเซิร์ฟเวอร์โดยใช้คอมพิวเตอร์ตัวเองเป็นเซิร์ฟเวอร์เพื่อทดสอบฟังก์ชันที่ทำไว้

2.2.2) มีการเชื่อมต่อกับฐานข้อมูลแบบ NoSQL โดยใช้เป็น MongoDB Atlas ซึ่งเป็นฐานข้อมูลแบบ Cloud

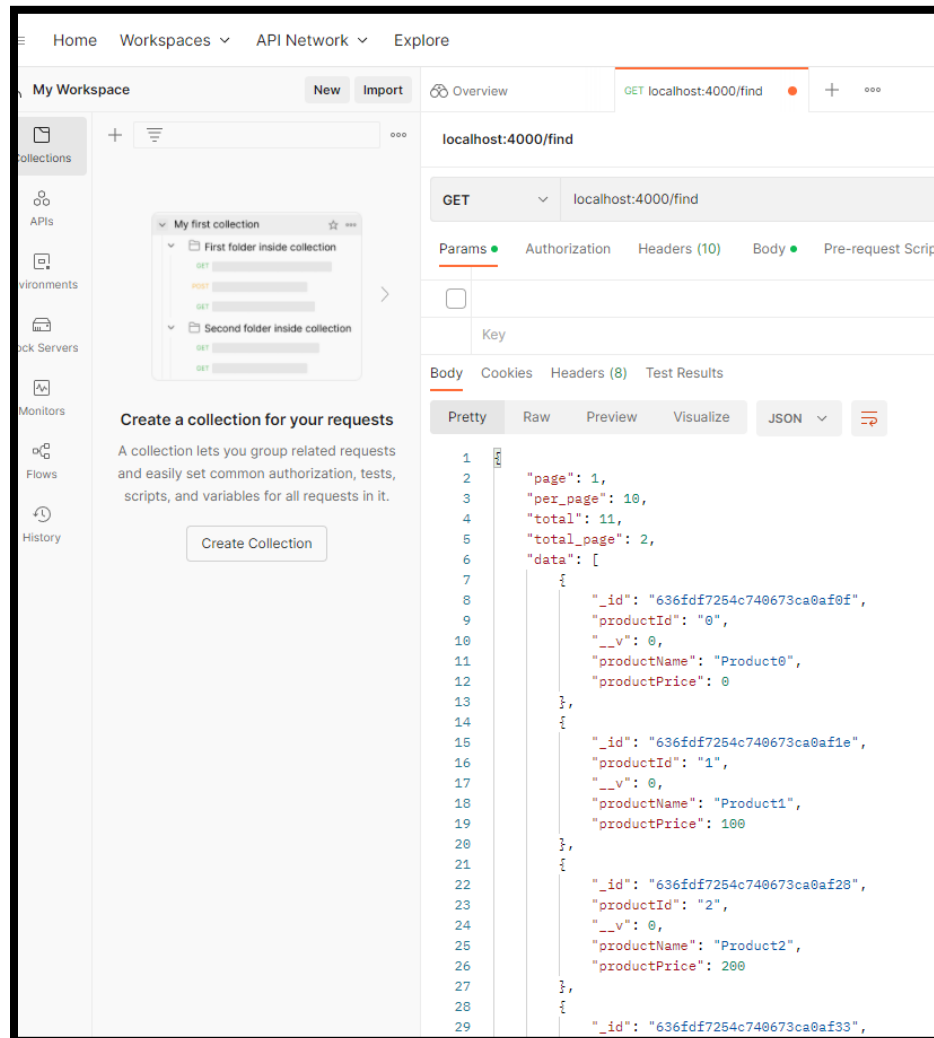
2.2.3) ทดสอบการจัดการฐานข้อมูลที่ได้ทำการเชื่อมไว้ โดยทำการลองเพิ่ม, ลบ, และเรียกดูข้อมูลทั้งหมดในฐานข้อมูล



รูปภาพที่ 3 แสดงข้อมูลในฐานข้อมูล MongoDB Atlas

2.4) ใช้ CORS(Cross-Origin Resource Sharing) เนื่องจากการที่ ReactJS กับ NodeJS เมื่อเปิดเซิร์ฟเวอร์แล้วจะมีโดเมนต่างกันทำให้ทางฝั่ง Front-end หรือ ReactJS ไม่สามารถทำการ Fetch API จากทางฝั่ง Back-end ได้เนื่องจากไม่ได้รับอนุญาต จึงต้องมี CORS มาช่วยในการทำให้ส่งข้อมูลคนละโดเมนกันได้

2.5) สร้าง API สำหรับส่งข้อมูลให้ทาง Front-end โดย API จะไปทำการเชื่อมต่อกับฐานข้อมูล MongoDB Atlas เมื่อเชื่อมต่อแล้วจะมีการตอนกลับเป็นข้อมูลที่อยู่ในรูป JSON



รูปภาพที่ 4 แสดงการทดสอบเรียกใช้ API ผ่าน Postman

โดยรวมส่วนใหญ่ที่ทำจะเป็นทางฝั่ง Back-end โดยทำเป็น API คอยรับ Request และ Respond ข้อมูลกลับไปฝั่งหน้า เนื่องจากว่าเป็นส่วนที่สำคัญกว่าเพราะหากทราบว่า Back-end ที่ทำไว้มีการทำงานอย่างไรจะทำให้มองเห็นภาพรวมได้มากขึ้น และไปทำฝั่ง Front-end ต่อได้

### 3) ปัญหาและอุปสรรค

ระหว่างการทำงานได้พบเจอปัญหา ทั้งในด้านของความรู้และทักษะในการสร้างเว็บไซต์ เนื่องจากโครงการที่ทำมีเนื้อหาที่ไม่ได้มีเนื้อหาสอนอยู่ในหลักสูตรที่เรียนอยู่ด้วย ต้องทำการค้นหาคำความรู้เองทั้งบทความจากเว็บไซต์ต่าง ๆ และวิดีโอบน Youtube ปัญหาที่พบเจอ มีทางด้าน Front-end และ Back-end ดังนี้

3.1) Front-end เนื่องจากการดำเนินงานของโครงการมีการใช้ ReactJS สำหรับทางฝั่ง Front-end ซึ่ง ReactJS เป็นสิ่งใหม่และไม่มีสอนระหว่างเรียน ทำให้ต้องหาคำความรู้ผ่านอินเทอร์เน็ต ฟังก์ชันที่มีอยู่ใน React มีอยู่มาก แต่ยังไม่ประยุกต์ใช้ยังไม่มาก ทำให้แค่เบื้องต้นเท่านั้น และ React ใช้ JSX เป็นรูปแบบภาษา Javascript เพื่อให้ Render ผลลัพธ์ออกมาเป็น User Interface ในส่วนนี้ก็ต้องเรียนรู้ Syntax ของ JSX เพื่อที่จะใช้งานได้คล่องแคล่วยิ่งขึ้น

3.2) Back-end ปัญหาที่เคยพบเจอระหว่างการทำงาน มีดังนี้

3.2.1) API ที่ทำไว้ไม่มีการ Return ค่ากลับมา

3.2.2) ไม่สามารถเชื่อมต่อกับฐานข้อมูล MongoDB Atlas ได้

### 4) แผนดำเนินงานต่อไป

4.1) ระบบการเข้าใช้งานระบบ (Authentication) โดยใช้ TU API

4.2) ระบบวางขายสินค้าในเว็บไซต์หรือ CRUD (Create-Read-Update-Delete) ที่ทำการจัดการฐานข้อมูลสินค้าผ่าน User Interface หรือฝั่ง Front-end

4.3) ระบบข้อความ (Text Chat) เพื่อการสื่อสารระหว่างผู้ใช้งาน

4.4) การ Deploy Server ขึ้นบน Cloud