

รายงาน

เรื่อง ความคืบหน้าโครงการ 2

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 จนทำให้อาจจะไม่น่าซื้อหรือน่าสนใจของผู้ซื้อ 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) สรุปผลการดำเนินงานโครงการครั้งที่ 1

จากการดำเนินงานในครั้งที่ 1 มีการดำเนินงานไปตามแผนที่วางไว้ แบ่งส่วนของ Front-end และ Back-end ivo อย่างชัดเจน แต่ก็มีบางระบบที่ยังไม่สามารถทำให้เสร็จตามเป้าหมายได้ เช่น ระบบพูดคุยสื่อสารระหว่างผู้ใช้งาน และ ระบบการเข้าใช้งานระบบโดยใช้ TU API Developers

คำแนะนำของอาจารย์ที่ปรึกษากับกรรมการจากการสอบโครงการ 1 สามารถนำมาสรุปเป็นหัวข้อได้ดังนี้

- 2.1) User interface นั้นยังทำได้ค่อนข้างไม่ดี ต้องมีการตกแต่งเพิ่มเติม
- 2.2) แนะนำเปลี่ยนชื่อหัวข้อโครงการให้สอดคล้องกับระบบที่จะทำ
- 2.3) ควรเริ่มทำที่ Front-end ก่อนแล้วค่อยไปทำ Back-end ที่หลัง

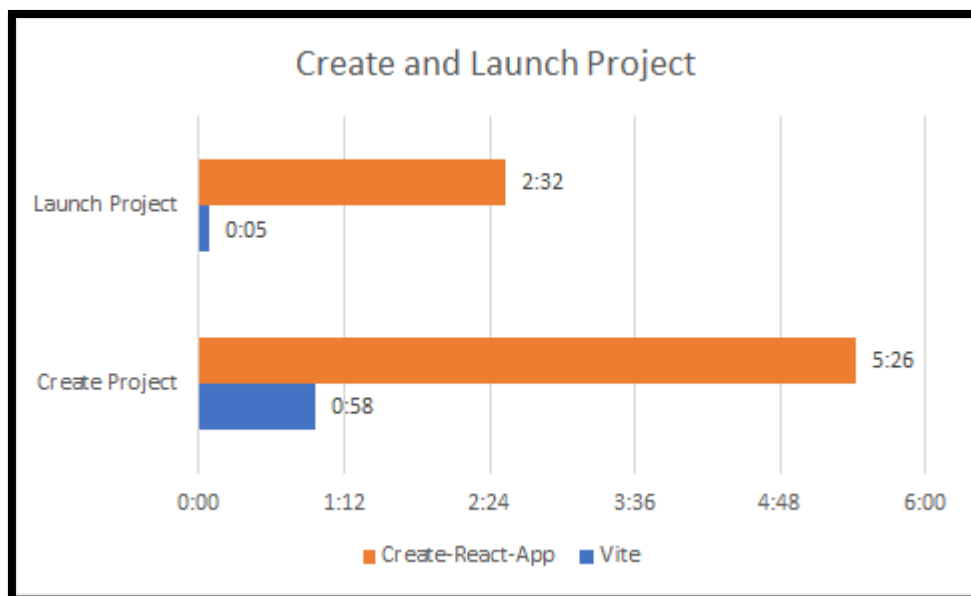
3) ผลการดำเนินงาน

หลังจากการดำเนินงานมาเป็นระยะเวลาหนึ่ง ได้มีการเพิ่มเติมและแก้ไขในส่วน Front-end และ Back-end ดังนี้

3.1) Front-end

3.1.1) เปลี่ยนแปลงการสร้างโปรเจก

จากที่แต่ก่อนใช้ Create-React-App เปลี่ยนการสร้างโปรเจก ReactJS โดยใช้ ViteJS ในการสร้างโปรเจกแทน เนื่องจาก ViteJS เป็น Framework library ที่มีความโดดเด่นในเรื่องของการติดตั้งความเร็วในการทำงาน และ Fast Refresh ซึ่งหาก Code มีการเปลี่ยนแปลง หน้าเว็บไซต์ก็จะมีการเปลี่ยนแปลงทันที ทำให้สะดวกต่อการพัฒนา Front-end เป็นอย่างมาก อีกอย่างคือหากใช้ ViteJS ในการสร้างโปรเจก ReactJS ก็จะมีไฟล์ Template เตรียมมาพร้อม ทำให้เพิ่มง่ายและสะดวกต่อการพัฒนาต่อไป

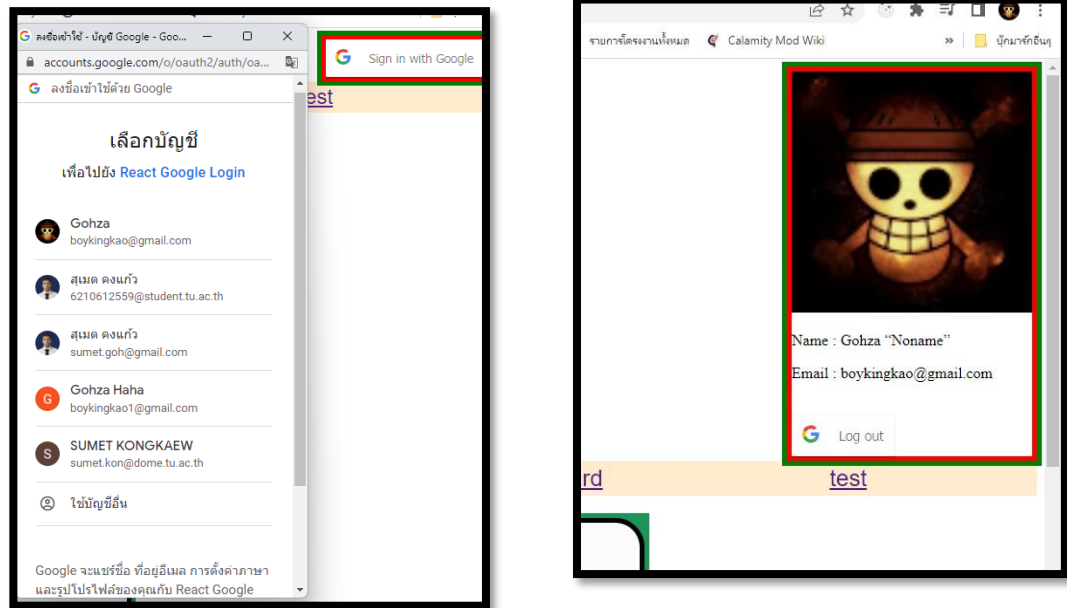


รูปภาพที่ 1 เปรียบเทียบความระยะเวลาในการรันโปรเจกระหว่าง ViteJS และ Create-React-App

ที่มา: <https://dev.to/frsfuzn/get-faster-creating-react-project-with-vite-4gmf>

3.1.2) ระบบการเข้าใช้งานระบบ (Authentication)

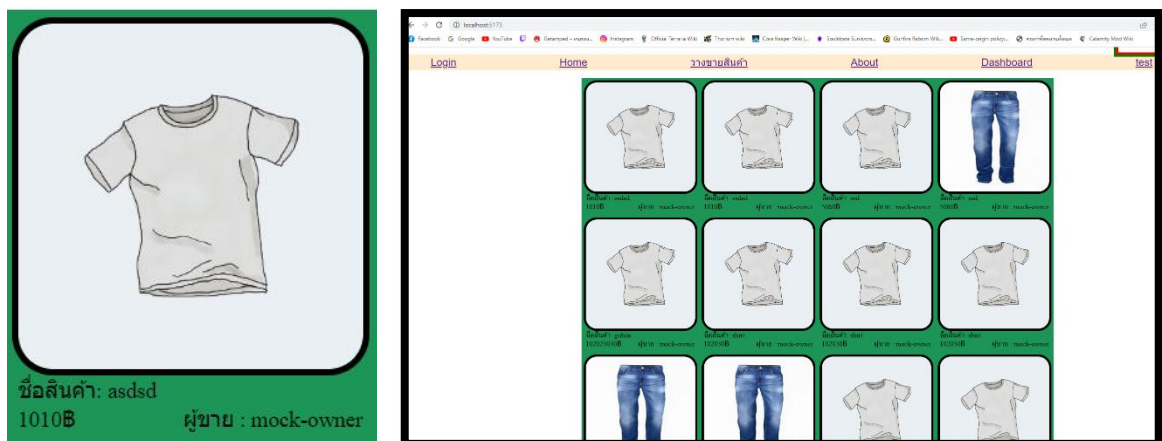
ใช้ Google API ชื่อว่า OAuth 2.0 เป็นการ Login โดยใช้ Account ของ Google ที่ทำการ Login ไว้ใน Browser หรือ Application อยู่แล้ว เมื่อ Login ก็จะมีการเก็บข้อมูลผู้ Login เข้าสู่ Localstorage เพื่อนำไปใช้กับระบบต่าง ๆ ที่จะเพิ่มเข้ามาในอนาคต



รูปภาพที่ 2 , 3 ระบบ Authentication ผ่าน OAuth 2.0

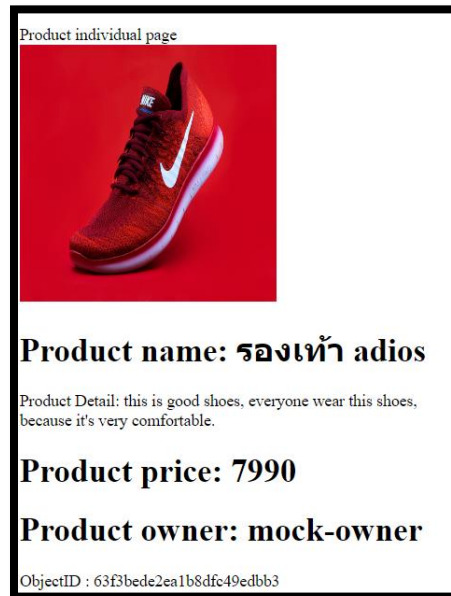
3.1.3) หน้าแสดงข้อมูลสินค้า

สินค้าทั้งหมดจะแสดงในหน้าเว็บไซต์ในรูปแบบของ Grid ซึ่งจะมีสินค้า 4 รายการต่อแถว แล้วต่อลงมาเรื่อย ๆ ข้อมูลสินค้าจะมีการรับข้อมูลสินค้าในรูปแบบไฟล์ JSON มาจากทางฝั่ง Back-end ที่จะคอยเชื่อมต่อกับฐานข้อมูล MongoDB Atlas



รูปภาพที่ 4 , 5 หน้าเว็บไซต์แสดงสินค้า

สินค้าแต่ละรายการจะสามารถคลิกไปที่รูปภาพ เพื่อ Redirect ไปยังหน้าที่แสดงข้อมูลสินค้า รายการนั้น ๆ อย่างเดียว ซึ่งมีคำอธิบายสินค้าในขณะที่หน้าเว็บไซต์สินค้ารวม ไม่มี เนื่องจากต้องการประหยัดพื้นที่หน้าเว็บไซต์ให้แสดงสินค้าได้เยอะ ๆ และสามารถคลิกที่ชื่อผู้ขาย เพื่อ Redirect ไปยังหน้าแสดงข้อมูลของผู้ขายสินค้านั้น ๆ



รูปภาพที่ 6 หน้าเว็บไซต์แสดงสินค้านั้น ๆ

3.1.3) ระบบเพิ่มข้อมูลสินค้า

ผู้ใช้งานสามารถเพิ่มข้อมูลสินค้าได้ โดยหลังจากกรอกข้อมูลในแบบฟอร์มแล้วจะทำการส่งข้อมูลไปให้ทางฝั่ง Back-end ผ่าน HTTP POST Request หลังจากนั้นเพิ่มข้อมูลสินค้าเข้าฐานข้อมูล โดยจะมี field ให้กรอกทั้งหมด 4 อย่าง คือ ชื่อสินค้า รูปภาพสินค้า รายละเอียดสินค้า และ ราคาสินค้า การส่งรูปภาพเข้า database จะต้องการทำการแปลงรูปภาพเป็น Base64 ก่อนส่งให้ Back-end



รูปภาพที่ 7 หน้าเว็บไซต์เพิ่มข้อมูลสินค้า

3.1.4) Deploy

ได้มีการทดสอบการ Deploy ในส่วนของ Front-end โดยผ่านเว็บไซต์ผู้ให้บริการ Hosting แบบ Cloud ชื่อว่า Netlify ซึ่งสามารถ Hosting Static Website ได้ฟรี ไม่มีค่าใช้จ่าย

การ Deploy มีไว้ทำให้ User คนอื่นใน Internet สามารถเข้าถึงเว็บไซต์ที่เราสร้างไว้ได้ และเพื่อเป็นการทดสอบหาจุดผิดพลาดจากการ Deploy แบบ Cloud เมื่อเทียบกับการ Deploy แบบ Localhost

3.2) Back-end

3.2.1) เชื่อมต่อฐานข้อมูล and สร้าง schema

3.2.2) สร้าง API

มีการสร้าง API เพื่อรองรับการเรียกใช้ HTTP Request จากทางฝั่ง Front-end โดยมีดังนี้

3.2.2.1) “/products” เป็น HTTP GET Request เมื่อเรียกใช้ API นี้ จะทำการค้นหาข้อมูลสินค้าทั้งหมดในฐานข้อมูล MongoDB แล้วจะมีการ Response ข้อมูลสินค้าทั้งหมดไปในรูปของไฟล์ JSON

```
// /product
router.get('/', async function (req, res, next) {
  // find data from mongodb
  let products = await Product.find();
  console.log(products)
  res.json(products);
});
```

รูปภาพที่ 8 API /product

3.2.2.1) “/products/add” เป็น HTTP POST Request รับ Request พร้อมข้อมูลสินค้าจากหน้าระบบเพิ่มข้อมูลสินค้า เมื่อเรียกใช้ API นี้ จะทำการเพิ่มข้อมูลสินค้าเข้าในฐานข้อมูล MongoDB

```
// /product/add
router.post('/add', function (req, res, next) {
  var product = req.body;
  const product1 = new Product({
    name: product.name,
    image: product.image,
    detail: product.detail,
    price: product.price,
    owner: "mock-owner"
  });
  product1.save();
  console.log("add product")
});
```

รูปภาพที่ 9 API /product/add

3.2.2.1) “/products/addTest” เป็น HTTP GET Request ที่เอาไว้ทดสอบเรียกใช้ API

ผ่าน Postman

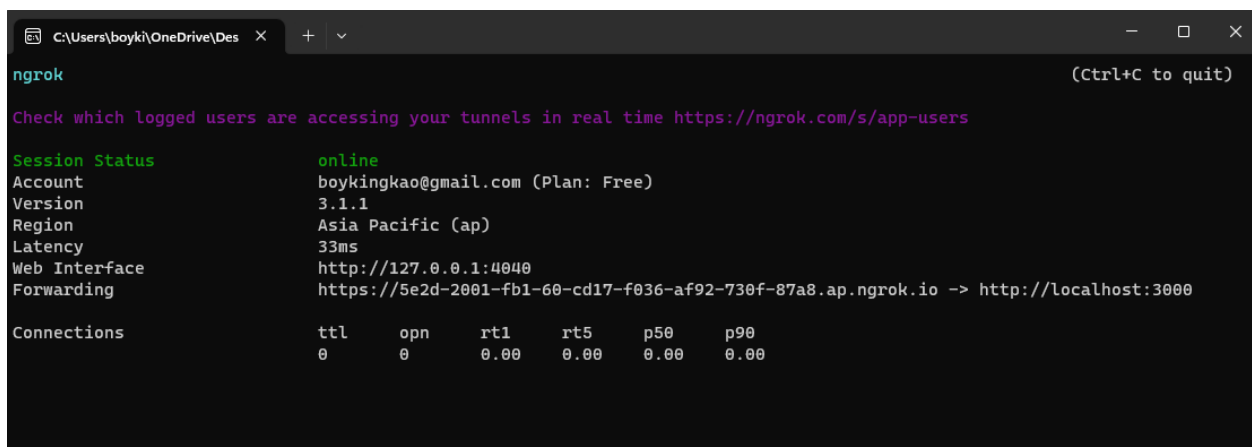
```
// /product/addTest
router.get('/addTest', function (req, res, next) {
  console.log("test api")

  const product = new Product({
    name: "Frontend name",
    image: "Frontend img",
    detail: "brabrahbra",
    price: Math.floor(Math.random() * 1000) + 1,
    owner: "owner"
  });
  product.save();
  console.log("add product")
  // res ok
  res.status(200).send("ok");
});
```

รูปภาพที่ 10 API /product/addTest

3.2.3) Ngrok

Ngrok เป็น Tool ที่สร้างโดย Github ซึ่งสามารถทำให้คนอื่นสามารถเข้าถึง Web Application ที่เปิดไว้ใน Localhost ได้ผ่านทางอินเทอร์เน็ต เหตุผลที่ต้องใช้เพราะเมื่อเวลา Hosting Front-end ขึ้น cloud ก็จะทำให้สามารถทำการ HTTP Request มายัง Backend ที่เปิดไว้ใน Localhost ได้ และ Ngrok สามารถใช้ได้ฟรี โดยแบบฟรี Subdomain ของ Ngrok จะสุ่มทุกครั้งตอนทำการ HTTP Tunnel Forwarding



The screenshot shows a terminal window for the Ngrok application. At the top, it says 'ngrok' and '(Ctrl+C to quit)'. Below that, it provides a link to check logged users: 'https://ngrok.com/s/app-users'. The 'Session Status' section shows the user is 'online', with account 'boykingkao@gmail.com (Plan: Free)', version '3.1.1', region 'Asia Pacific (ap)', latency '33ms', web interface 'http://127.0.0.1:4040', and forwarding 'https://5e2d-2001-fb1-60-cd17-f036-af92-730f-87a8.ap.ngrok.io -> http://localhost:3000'. A 'Connections' table is also displayed with columns: ttl, opn, rt1, rt5, p50, p90. The table shows a single row with values: 0, 0, 0.00, 0.00, 0.00, 0.00.

ttl	opn	rt1	rt5	p50	p90
0	0	0.00	0.00	0.00	0.00

รูปภาพที่ 11 การ HTTP Tunnel Forwarding โดยใช้ Ngrok

4) ปัญหาและอุปสรรค

ระหว่างการทำงานได้พบเจอปัญหา ทั้งในส่วน Front-end และ Back-end ดังนี้

4.1) Front-end

4.1.1) เกิด Error code 413 ที่ทำให้ไม่สามารถ Fetch API ไปทางฝั่ง Back-end ได้ เหตุผลมาจากการที่ Data รูปภาพที่แปลงเป็น BASE64 แล้วมีขนาดใหญ่เกินไป เมื่อส่งไปกับการ Fetch API ทำให้ทางฝั่ง Server ไม่รับ Request นั้น เนื่องจากมีขนาดใหญ่เกินไปจากขีดจำกัดของ HTTP Request

4.1.2) ทักษะในด้านการตกแต่งเว็บไซต์ด้วย CSS ค่อนข้างต่ำ

4.2) Back-end

4.2.1) รับ HTTP Request ไม่ได้เนื่องจาก Request นั้นมีขนาดใหญ่เกินไป ดังนั้นสิ่งที่แก้คือ การปรับเพิ่มขีดจำกัดของ HTTP Request ให้สูงขึ้น เพื่อที่จะทำให้รับ HTTP Request ที่มี Data ขนาดใหญ่ส่งมาได้

4.2.2) การใช้ Ngrok กับทาง Back-end ไม่สามารถใช้ได้ทันที ต้องมีการ Coding โดยเพิ่ม Library Ngrok เข้ามาใช้ก่อนเพื่อเป็นการทำให้เซิร์ฟเวอร์ที่อยู่ใน Localhost เข้าสู่อินเทอร์เน็ต

```
23
24 const ngrok = require('ngrok');
25 (async function() {
26   const url = await ngrok.connect(port, {
27     subdomain: "69b1-2001-fb1-61-7ead-6c88-a0b3-b168-4887",
28     authToken: "2EyKLw7wSJxe0yGdizndswCtEML_3z2QpCq9MhAi5Hw7rKc6R" })
29   console.log(`ngrok tunnel started at ${url}`);
30 })();
31
```

รูปภาพที่ 12 ใช้ Ngrok Library กับ NodeJS

5) แผนดำเนินงานต่อไป

- 5.1) ระบบข้อความ (Text Chat) เพื่อการสื่อสารระหว่างผู้ใช้งาน
- 5.2) ระบบการซื้อขายสินค้าแบบ P2P คือ ผู้ใช้ทั้ง 2 ฝ่ายติดต่อซื้อขายกันได้
- 5.3) ตกแต่งเว็บไซต์เพิ่มเติม
- 5.4) ระบบการยืนยันตัวตนสำหรับบุคลากรในมหาวิทยาลัยธรรมศาสตร์

6) การแสดงผลงานขั้นสุดท้าย

- 6.1) มีระบบเพียงพอที่จะเรียกเว็บไซต์นี้เป็นประเภทเว็บไซต์อีคอมเมิร์ซ
- 6.2) เว็บไซต์มีการออกแบบที่ดี สามารถดึงดูดผู้เข้าชมใหม่ๆ ได้