****

**รายงานความคืบหน้าโครงงาน 1**

**การปรับปรุงและพัฒนาความสามารถเว็บแอปพลิเคชันฝึกงานสำหรับภาควิชา 2**

**โดย**

**นายภูชิชย์ กลีบมาลัย**

**นายวงศกร กองกะมุด**

**โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต**

**สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**มหาวิทยาลัยธรรมศาสตร์**

**ปีการศึกษา 2566**

**ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์**

**รายงานความคืบหน้าโครงงาน 1**

**ชื่อโครงงานภาษาไทย** การปรับปรุงและพัฒนาความสามารถเว็บแอปพลิเคชันฝึกงานสำหรับภาควิชา 2

**ชื่อโครงงานภาษาอังกฤษ** Enhancing and Developing the Capabilities of the Department's Internship Web Application

**ชื่อและเลขทะเบียนนักศึกษาคนที่ 1** ภูชิชย์ กลีบมาลัย 6210612716

**ชื่อและเลขทะเบียนนักศึกษาคนที่ 2** วงศกร กองกะมุด 6310611006

**อาจารย์ที่ปรึกษา** อาจารย์ นาวิน สมญาติ

**1. เค้าโครงของโครงงาน**

**1.1 ภูมิหลังทางวิทยาการ**

ในปัจจุบันการประชาสัมพันธ์บริษัทรับนักศึกษาฝึกงานและขั้นตอนในการดำเนินเอกสารการฝึกงานไม่ค่อยมีความสะดวกหรือความเป็นระบบมากนัก ทำให้นักศึกษาหลายคนไม่ทราบการประชาสัมพันธ์และเกิดความสับสนว่าจะต้องดำเนินการเอกสารอย่างไรต่อไป ซึ่งความสับสนของนักศึกษานี้อาจส่งผลถึงการดำเนินเอกสารของเจ้าหน้าที่ที่เกี่ยวข้องและอาจารย์ผู้ประสานงานด้านการฝึกงานของนักศึกษาให้เกิดความล่าช้าหรือผิดพลาดได้

ในขั้นตอนของการดำเนินเอกสารฝึกงานจะต้องผ่านผู้ที่เกี่ยวข้องคือ นักศึกษา เจ้าหน้าที่ที่เกี่ยวข้อง และอาจารย์ผู้ประสานงานรวมถึงบริษัทหรือหน่วยงานที่รับนักศึกษาฝึกงาน ซึ่งผู้จัดทำเห็นว่ามีระบบหรือแพลตฟอร์มจากโครงงานในปีการศึกษา 2565 ที่รวบรวมขั้นตอนต่าง ๆ ในการดำเนินเอกสารการฝึกงานของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์อยู่แล้ว แต่ระบบมีความไม่สมบูรณ์และยังไม่สามารถนำมาใช้งานได้จริง

จากปัญหาขั้นต้นผู้จัดทำจึงเห็นว่าควรพัฒนาระบบต่อจากเดิมให้สมบูรณ์และสามารถใช้งานได้จริง และทำการเพิ่มความสามารถในการใช้งาน เช่น ระบบแจ้งเตือนที่สามารถส่งข้อความหรืออีเมลแจ้งเตือนผู้ใช้งานเกี่ยวกับสถานะและความคืบหน้าของกระบวนการฝึกงาน ระบบยื่นข้อมูลของหน่วยงานหรือบริษัทที่นักศึกษาติดต่อเพื่อไปฝึกงานด้วยตนเอง และระบบสร้างเอกสารอัตโนมัติ และแพลตฟอร์มนี้อาจใช้เป็นพื้นฐานในการพัฒนาและขยายฟังก์ชันเพิ่มเติมในอนาคตเพื่อตอบสนองความต้องการของคณะวิศวกรรมและผู้ประกอบการ

**1.2 วัตถุประสงค์**

1.2.1 เพื่อปรับปรุงและพัฒนาระบบให้สามารถนำไปใช้งานได้จริง

1.2.2 เพื่อปรับปรุงและพัฒนาระบบการดำเนินเอกสารที่เกี่ยวข้องกับการฝึกงาน

1.2.3 เพื่อปรับปรุงและพัฒนาระบบในการประชาสัมพันธ์ข่าวสารเกี่ยวกับการฝึกงาน

1.2.4 เพื่อปรับปรุงและพัฒนาระบบให้สามารถใช้งานได้ภายในเว็บแอปพลิเคชันเดียว

**1.3 ขอบเขตการดำเนินงาน**

1.3.1 พัฒนาระบบด้วยเฟรมเวิร์ก Node.JS และ React

1.3.2 ออกแบบ User Interface (UI) ของระบบใหม่ทั้งหมดให้รองรับการทำงานแบบ Responsive

1.3.3 พัฒนาระบบโพสต์ประกาศหานักศึกษาฝึกงานในระบบให้มีความสามารถและรายละเอียดมากขึ้น

1.3.4 พัฒนาระบบยื่นแจ้งขอเอกสารการฝึกงานสำหรับสถานประกอบการจากภายนอกระบบได้

1.3.5 พัฒนาระบบการสร้างเอกสารอัตโนมัติ

1.3.6 ระบบสามารถบอกสถานะและแจ้งเตือนการดำเนินการเอกสารได้

1.3.7 สนับสนุนเอกสารต่าง ๆ ที่ใช้ในการฝึกงาน และข้อมูลกำหนดการของวิชาฝึกงาน

1.3.8 สนับสนุนการเก็บข้อมูล และการประเมินต่าง ๆ หลังจากเสร็จสิ้นการฝึกงาน

**1.4 เป้าหมายของโครงงาน**

1.4.1 ระบบสามารถนำไปใช้งานได้จริง

1.4.2 ช่วยให้นักศึกษาเข้าถึงประชาสัมพันธ์บริษัทรับนักศึกษาฝึกงานมากขึ้น ลดความซับซ้อนและความสับสนในการจัดการเอกสารสำหรับนักศึกษา อาจารย์ผู้ดูแลและผู้ประกอบการ

1.4.3 การลดการใช้กระดาษและเอกสารที่ไม่จำเป็นสามารถช่วยลดค่าใช้จ่ายในการดำเนินการ

1.4.4 ระบบสามารถช่วยในการติดตามและประเมินผลการฝึกงานของนักศึกษา

1.4.5 แพลตฟอร์มนี้อาจใช้เป็นพื้นฐานในการพัฒนาและขยายฟังก์ชันเพิ่มเติมในอนาคตเพื่อตอบสนองความต้องการของคณะวิศวกรรมศาสตร์และผู้ประกอบการ

**2. ทฤษฎีที่เกี่ยวข้อง**

การปรับปรุงและพัฒนาความสามารถเว็บแอปพลิเคชันฝึกงานสำหรับภาควิชา 2 ทางคณะผู้จัดทำได้ทำการศึกษาค้นคว้าทฤษฎี และหลักการต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาเว็บแอปพลิเคชัน ดังรายละเอียดต่อไปนี้

2.1 เว็บแอปพลิเคชัน (Web Application)

2.2 การพัฒนาเว็บแอปพลิเคชัน (Web Application Development)

2.3 JSON Web Token (JWT)

2.4 Middleware

2.5 MVC (Model-View-Controller)

2.6 Component-Based Architecture

2.7 Asynchronous Programming

2.8 RESTful API

2.9 SQL Injection

**2.1 เว็บแอปพลิเคชัน (Web Application)**

เว็บแอปพลิเคชัน หรือที่เรียกกันทั่วไปว่าเว็บแอป เป็นซอฟต์แวร์เชิงโต้ตอบที่ออกแบบมาเพื่อทำงานภายในเว็บเบราว์เซอร์ต่าง ๆ โดยข้อมูลต่างของเว็บแอปจะถูกจัดเก็บไว้ในเซิร์ฟเวอร์ ณ ที่ใดที่หนึ่ง ทำให้ผู้ใช้สามารถเข้าถึงเว็บแอปพลิเคชันเหล่านั้นผ่านทางอินเทอร์เน็ตโดยผ่านเว็บเบราว์เซอร์บนอุปกรณ์ใดก็ได้ [1]

เว็บแอปพลิเคชันได้รับการพัฒนาเพื่อให้ใช้ในการทำงานต่าง ๆ ได้ง่ายขึ้น โดยเว็บแอปถูกพัฒนาเพื่อใช้งานตามวัตถุประสงค์ที่เฉพาะเจาะจงหรือใช้ในงานที่เหมาะสมเพื่อตอบสนองความต้องการของบุคคลหรือองค์กร โดยมีฟังก์ชันการทำงานที่หลากหลาย แม้ว่าเว็บแอปพลิเคชันส่วนใหญ่จะถูกสร้างขึ้นมาให้สามารถทำงานบนเบราว์เซอร์ใดก็ได้ แต่เว็บแอปพลิเคชันบางตัวอาจมีข้อกำหนดเกี่ยวกับเบราว์เซอร์ที่ใช้งานเพื่อที่จะให้สามารถทำงานได้อย่างมีประสิทธิภาพ [1]

**2.2 การพัฒนาเว็บแอปพลิเคชัน (Web Application Development)**

การพัฒนาเว็บแอปพลิเคชันเป็นกระบวนการสร้างแอปพลิเคชันที่สามารถเข้าถึงและใช้งานได้ผ่านเว็บเบราว์เซอร์ โดยมีการเขียนโปรแกรมทั้งฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ โดยส่วนใหญ่จะเริ่มต้นด้วยการกำหนดปัญหา ตามด้วยขั้นตอนของการออกแบบ จากนั้นนักพัฒนาจะทำการทดสอบก่อนเปิดตัวเว็บแอปพลิเคชัน [2]

เว็บแอปพลิเคชันและเว็บไซต์มีความคล้ายกัน นักพัฒนาเว็บแอปจะใช้เทคโนโลยีในการพัฒนา Front-End เช่น HTML, CSS และ JavaScript ซึ่งจะคล้ายกับการพัฒนาเว็บไซต์ทั่ว ๆ ไป นอกจากนี้ในส่วน Back-End ของเว็บแอปพลิเคชันอาจใช้ภาษาในการเขียนโปรแกรมฝั่งเซิร์ฟเวอร์ เช่น Ruby หรือ Python ซึ่งก็จะคล้ายกับภาษาที่ใช้ในการพัฒนาเว็บไซต์ทั่วไปเช่นกัน [2] โดยเว็บแอปพลิเคชันได้รับการออกแบบให้ทำงานได้อย่างราบรื่นบนอุปกรณ์ต่าง ๆ และมอบประสบการณ์ที่ดีให้กับผู้ใช้ ซึ่งจะมีความแตกต่างกันเมื่อเปรียบเทียบกับเว็บไซต์แบบดั้งเดิม [2]

**2.3 JSON Web Token (JWT)**

JSON Web Token (JWT) เป็น open standard (RFC 7519) ที่กำหนดวิธีการที่เหมาะสมสำหรับการส่งข้อมูลระหว่างกันได้อย่างปลอดภัยในฐานะ JSON Object ที่จะมีขนาดกระทัดรัดและเก็บข้อมูลได้ภายในตัวเอง ซึ่ง JWT ใช้อัลกอริทึม HMAC หรือใช้คู่ของ Public Key และ Private Key โดยใช้อัลกอริทึม RSA หรือ ECDSA ในการเซ็นลายเซ็นดิจิทัล ทำให้ข้อมูลที่ส่งกันนี้มีความน่าเชื่อถือและสามารถตรวจสอบได้ [3]

โดย JWT มักจะถูกใช้สำหรับการ Authorization หรือการตรวจสอบระดับการเข้าถึงหรือสิทธิ์ของผู้ใช้ที่เข้าสู่ระบบอยู่ และใช้ในการส่งข้อมูลระหว่างกัน

**2.4 Middleware**

Middleware คือซอฟต์แวร์ที่เป็นเหมือนตัวกลางที่แอปพลิเคชันต่าง ๆ ใช้ในการสื่อสารระหว่างกัน มีฟังก์ชันในการเชื่อมต่อแอปพลิเคชันได้อย่างมีประสิทธิภาพ โดย Middleware ทำหน้าที่ในการเชื่อมต่อระหว่างเทคโนโลยี เครื่องมือ และฐานข้อมูลต่าง ๆ ให้สามารถรวมเข้าเป็นระบบเดียวได้อย่างราบรื่น โดยจะรวมบริการทั้งหมดให้กับผู้ใช้ เช่น แอปพลิเคชันส่วน Front-End ของ Windows ส่งและรับข้อมูลจากเซิร์ฟเวอร์ Back-End ของ Linux โดยที่ผู้ใช้ไม่ทราบถึงความแตกต่าง [4]

นักพัฒนาจะใช้ Middleware ในการพัฒนาแอปพลิเคชัน ถ้าหากไม่มี Middleware นักพัฒนาจะต้องสร้างโมดูลในการรับส่งข้อมูลสำหรับส่วนประกอบของซอฟต์แวร์แต่ละตัวที่เชื่อมต่อกับแอปพลิเคชัน ซึ่ง Middleware จะช่วยให้นักพัฒนาสามารถประหยัดเวลาในส่วนนี้ได้ [4]

**2.5 MVC (Model-View-Controller)**

MVC หรือ Model-View-Controller เป็นรูปแบบสถาปัตยกรรมที่แยกแอปพลิเคชันออกเป็นสามส่วนหลัก ๆ คือ Model, View และ Controller แต่ละส่วนประกอบถูกสร้างขึ้นเพื่อรองรับการพัฒนาแอปพลิเคชันโดยเฉพาะ โดย MVC เป็นหนึ่งในเฟรมเวิร์กที่ถูกใช้บ่อยที่สุดสำหรับการพัฒนาเว็บ [5]

ส่วน Model เป็นส่วนที่เกี่ยวข้องกับข้อมูลทั้งหมดที่ผู้ใช้จะทำงานด้วย ตัวอย่างเช่น Customer Object จะมีข้อมูลของลูกค้าจากในฐานข้อมูล ซึ่งสามารถนำมาเพื่อแสดงผล หรือนำมาแก้ไขข้อมูล แล้วทำการส่งกลับไปยังฐานข้อมูลได้ โดยจะถูกเรียกใช้จากส่วน Controller [5]

ส่วน View เป็นส่วนที่เกี่ยวข้องกับการแสดงผลทั้งหมดของแอปพลิเคชัน หรือเป็นส่วนที่ติดต่อกับผู้ใช้โดยตรง (User Interface) ตัวอย่างเช่น ผู้ใช้จะเห็นส่วนประกอบต่าง ๆ ของ User Interface ทั้งหมด รวมถึงเห็นข้อมูลที่รับมาแสดงผลจากส่วน Model โดยจะรับคำสั่งจาก Controller และผู้ใช้ [5]

ส่วน Controller ทำหน้าที่เป็นช่องทางในการสื่อสารระหว่างส่วน Model และ View ในการประมวลผลและจัดการกับการร้องขอข้อมูลต่าง ๆ เหมือนกับเป็นศูนย์กลางการทำงานทั้งหมด ตัวอย่างเช่น Customer Controller จะทำหน้าที่ในการจัดการการโต้ตอบและรับข้อมูลจาก Customer View หรือผู้ใช้ และทำการแก้ไขข้อมูลในฐานข้อมูลโดยใช้ Customer Model [5]

**2.6 Component-Based Architecture**

Component-Based Architecture เป็นวิธีการสำหรับการสร้างซอฟต์แวร์หรือแอปพลิเคชันที่จะสร้างส่วนประกอบต่าง ๆ ของแอปพลิเคชันขึ้นมา และนำส่วนประกอบเหล่านั้นมาใช้งานซ้ำ เช่น ปุ่มกดเพื่อยืนยัน, ปุ่มกดเพื่อยกเลิก เป็นต้น โดยที่แต่ละส่วนประกอบจะรวมฟังก์ชันการทำงานต่าง ๆ ที่สามารถจัดเก็บไว้และนำมาใช้ในแอปพลิเคชันโดยที่ไม่ต้องมีการแก้ไขหรือกระทบกับส่วนประกอบอื่น ๆ [6]

โดย Component-Based Architecture ช่วยให้นักพัฒนาสามารถประหยัดเวลาที่ใช้ในการพัฒนาและทดสอบแอปพลิเคชัน มีความน่าเชื่อถือที่เพิ่มขึ้นเนื่องจากมีการทดสอบส่วนประกอบแต่ละชิ้นไปก่อนแล้ว และทำให้มีความยืดหยุ่นในการปรับเปลี่ยนการใช้งานต่าง ๆ ด้วยการเพิ่มหรือเปลี่ยนส่วนประกอบที่ต้องการได้ [6]

**2.7 Asynchronous Programming**

Asynchronous Programming เป็นวิธีในการเขียนโปรแกรมคอมพิวเตอร์แบบขนาน (Parallel) ที่จะทำให้กระบวนการทำงานหนึ่งสามารถทำงานแยกจากฟังก์ชันการทำงานหลักของโปรแกรม โดยเมื่อกระบวนการทำงานนั้นเสร็จสิ้น การส่งหรือใช้ข้อมูลที่ได้มานั้นอาจส่งผลกระทบต่อฟังก์ชันการทำงานหลัก ซึ่ง Asynchronous Programming มักจะช่วยลดหรือป้องกันความล่าช้าจากการทำงานของกระบวนการหนึ่ง ๆ โดยทำให้กระบวนการนั้นสามารถทำงานต่อไปได้ในพื้นหลังของแอปพลิเคชันหลัก หรือทำงานคู่ขนานกันไป [7]

**2.8 RESTful API**

RESTful API ทำหน้าที่เป็นช่องทางที่ช่วยให้เกิดการแลกเปลี่ยนข้อมูลที่ปลอดภัยระหว่างระบบคอมพิวเตอร์สองระบบผ่านทางอินเทอร์เน็ต แอปพลิเคชันที่มีข้อมูลที่มีความละเอียดอ่อนจะต้องมีการสื่อสารที่ราบรื่นกับแอปพลิเคชันของบุคคลที่สามและระบบภายในแอปพลิเคชันเอง ตัวอย่างเช่น ในการสร้างสลิปเงินเดือนและออกใบแจ้งหนี้อัตโนมัติ ระบบบัญชีภายในของแอปพลิเคชันจะต้องแบ่งปันข้อมูลกับระบบธนาคารของผู้ใช้ และสื่อสารกับแอปพลิเคชันใบบันทึกเวลาของระบบภายใน ซึ่ง RESTful API รองรับการแลกเปลี่ยนข้อมูลที่มีความสำคัญนี้ เพราะมีมาตรฐานการสื่อสารระหว่างซอฟต์แวร์ที่มีความปลอดภัย เชื่อถือได้ และมีประสิทธิภาพ [8]

REST ย่อมาจาก Representational State Transfer เป็นสถาปัตยกรรมซอฟต์แวร์ที่กำหนดเงื่อนไขว่า API ควรทำงานอย่างไร โดย REST ถูกสร้างขึ้นครั้งแรกเพื่อเป็นแนวทางในการจัดการการสื่อสารบนเครือข่ายที่ซับซ้อนอย่างอินเทอร์เน็ต ซึ่งสามารถใช้สถาปัตยกรรมที่ใช้ REST เพื่อรองรับการสื่อสารที่มีประสิทธิภาพสูงและเชื่อถือได้ [8]

โดย API ที่ใช้สถาปัตยกรรมที่เป็นไปตามรูปแบบสถาปัตยกรรมแบบ REST จะเรียกว่า REST API และ Web services ที่ใช้สถาปัตยกรรมแบบ REST จะเรียกว่า RESTful Web services ซึ่งคำว่า RESTful API โดยทั่วไปนั้นหมายถึง RESTful Web API [8]

**2.9 SQL Injection**

SQL Injection (SQLi) เป็นช่องโหว่ด้านความปลอดภัยบนเว็บที่ทำให้ผู้โจมตีสามารถแทรกแซงโดยการใช้คำสั่งบางอย่างที่ให้แอปพลิเคชันติดต่อกับฐานข้อมูลได้ เช่น ผ่านทางฟอร์มที่ให้ผู้ใช้กรอกข้อมูล ซึ่งจะทำให้ผู้โจมตีสามารถเข้าถึงข้อมูลที่ไม่ได้รับอนุญาตให้เข้าถึงได้ โดยผู้โจมตีสามารถทำได้มากกว่าการดูข้อมูล อาจทำการแก้ไขหรือลบข้อมูล ซึ่งอาจจะทำให้เกิดการปัญหาต่อแอปพลิเคชันและผู้ใช้ได้ [9]

หากการโจมตีด้วยช่องโหว่นี้ทำได้สำเร็จอาจส่งผลให้ผู้โจมตีสามารถเข้าถึงข้อมูลที่ละเอียดอ่อนได้ เช่น รหัสผ่าน, ข้อมูลบัตรเครดิต หรือข้อมูลส่วนบุคคลของผู้ใช้ เป็นต้น ในกรณีที่รุนแรงมากขึ้น การโจมตีด้วย SQL Injection สามารถยกระดับการโจมตีเพื่อโจมตีเซิร์ฟเวอร์หรือโครงสร้างพื้นฐานต่าง ๆ ของ Back-End ได้ [9]

**3. เทคโนโลยีที่จะใช้ในการทำโครงงาน**

**3.1 Node.js**

Node.js ถูกเผยแพร่ในปี ค.ศ. 2009 โดย Ryan Dahl ซึ่งช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันทั้งส่วน Front-End และส่วน Back-End โดยใช้ JavaScript ซึ่ง Node.js เป็น open-source และสามารถใช้งานได้บนหลากหลายแพลตฟอร์มโดยไม่ได้ขึ้นอยู่กับระบบปฏิบัติการที่ใช้ เช่น Linux, MacOS หรือ Windows [10]

Node.js เป็น JavaScript runtime environment หรือก็คือ เมื่อเขียนโค้ด JavaScript ใน Text editor โค้ดหรือโปรแกรมนั้นจะไม่สามารถทำงานใด ๆ ได้ เว้นแต่ว่าโค้ดนั้นจะถูกเรียกใช้งาน และในการรันโค้ดก็จะต้องมี runtime environment [10]

**3.2 Express.js**

Express.js เป็นเว็บเฟรมเวิร์กสำหรับ Node.js ที่ได้รับความนิยมสูง โดยถูกออกแบบมาเพื่อใช้ในการสร้างเว็บแอปพลิเคชันและ API โดยในการสร้าง Back-End ตั้งแต่การกำหนดค่าพอร์ตไปจนถึงการสร้างตัวจัดการเส้นทาง หากใช้เพียง Node.js อาจจะใช้เวลานาน ซึ่งการใช้เว็บเฟรมเวิร์กอย่าง Express.js จะช่วยให้นักพัฒนาสามารถประหยัดเวลาได้มากขึ้นและสามารถเน้นไปที่งานที่สำคัญอื่น ๆ ได้ โดยจะมีคุณสมบัติหลัก ๆ คือ การจัดการกับ requests, Middleware, Routing เป็นต้น [11]

**3.3 MySQL**

MySQL เป็นระบบจัดการฐานข้อมูลที่เป็น open-source SQL ที่ได้รับความนิยมสูงสุด โดยได้รับการพัฒนา จัดจำหน่าย และสนับสนุนโดย Oracle Corporation [12]

MySQL เป็น Relational Database Management System (RDBMS) [13] ซึ่งเป็นฐานข้อมูลชนิดหนึ่งที่จัดเก็บข้อมูลในรูปแบบของตาราง และแต่ละตารางที่มีความสัมพันธ์กันจะสามารถเข้าถึงข้อมูลกันได้ [14]

**3.4 XAMPP**

XAMPP เป็นเว็บเซิร์ฟเวอร์แบบ open-source ที่ใช้งานได้ฟรี โดยชื่อ XAMPP ย่อมาจาก Cross-Platform, Apache, MySQL, PHP และ Perl ซึ่ง XAMPP ถูกพัฒนาขึ้นโดย Apache Friends [15]

XAMPP เป็นตัวเลือกยอดนิยมสำหรับโปรแกรมเมอร์ที่ต้องการเขียนและทดสอบโค้ดบนเว็บเซิร์ฟเวอร์บน Local Host โดย XAMPP ประกอบไปด้วย MariaDB, Apache HTTP Server และ interpreters สำหรับ PHP และ Perl และยังรองรับภาษาการเขียนโปรแกรมอื่น ๆ มากมาย [15]

**3.5 Postman**

Postman ถูกสร้างขึ้นโดย Abhinav Asthana ซีอีโอและผู้ร่วมก่อตั้งของ Postman ด้วยความตั้งใจที่จะสร้างเครื่องมือที่จะทำให้กระบวนการทดสอบ API ง่ายขึ้น โดย Postman เป็นแพลตฟอร์ม API สำหรับการสร้างและใช้งาน API ทำให้แต่ละขั้นตอนในการใช้งาน API ง่ายขึ้น และเพิ่มความคล่องตัวในการทำงานร่วมกันเพื่อให้สามารถสร้าง API ที่ดีขึ้นได้รวดเร็วยิ่งขึ้น ในปัจจุบัน Postman กลายเป็นแพลตฟอร์ม API ชั้นนำของโลก [16]

**3.6 React**

React.js เป็น JavaScript Library แบบ open-source ซึ่งสร้างขึ้นโดย Facebook โดยมีจุดมุ่งหมายเพื่อลดความซับซ้อนของกระบวนการสร้าง User Interfaces แบบเชิงตอบโต้ [17]

ใน React สามารถพัฒนาเว็บแอปพลิเคชันได้โดยการสร้างส่วนประกอบต่าง ๆ ของเว็บแอปพลิเคชันหรือที่เรียกว่า Components โดยที่สามารถนำมาใช้ซ้ำได้ ส่วนประกอบต่าง ๆ เหล่านี้จะเป็นชิ้นส่วนย่อยของหน้าต่าง ๆ หรือส่วนประกอบขนาดใหญ่กว่า ซึ่งเมื่อนำมาประกอบเข้าด้วยกันก็จะเป็น User Interfaces ที่สมบูรณ์ของเว็บแอปพลิเคชัน [17]

**3.7 Bootstrap**

Bootstrap เป็น open-source ที่ถูกพัฒนาโดย Mark Otto และ Jacob Thornton โดยเป็นชุดเครื่องมือสำหรับการสร้างเว็บไซต์และเว็บแอปพลิเคชัน ซึ่งเป็นเฟรมเวิร์ก HTML, CSS และ JavaScript ที่ได้รับความนิยมมากที่สุดสำหรับการพัฒนาเว็บไซต์หรือการออกแบบที่เน้นมือถือเป็นอันดับแรก [18]

การใช้งาน Bootstrap จะช่วยให้สามารถจัดการการตกแต่งหน้าเว็บได้ง่ายและสะดวกยิ่งขึ้น เช่น รูปแบบของตัวอักษร, สีข้อความ, สีพื้นหลัง และการจัดวางตำแหน่ง เป็นต้น [18]

**3.8 PDFKit**

PDFKit เป็นไลบรารีที่ใช้ในการสร้างเอกสาร PDF สำหรับ Node และเบราว์เซอร์ที่ทำให้การสร้างเอกสารที่ซับซ้อน หลายหน้า และพิมพ์ได้ง่าย มี API ที่ครอบคลุมทั้งฟังก์ชันแบบ Low level และ สูงกว่า โดย PDFKit API ได้รับการออกแบบมาให้มีความเรียบง่าย ดังนั้น ในการสร้างเอกสารที่ซับซ้อนจึงมักทำได้เพียงแค่เรียกใช้ฟังก์ชันเพียงไม่กี่ครั้ง [19]

**3.9 Figma**

Figma เป็นเครื่องมือที่ใช้ในออกแบบโดยให้บริการบนคลาวด์ ซึ่งจะมีความคล้ายคลึงกับ Sketch ในแง่ของฟังก์ชันและฟีเจอร์ แต่มีความแตกต่างที่ทำให้ Figma นั้นเด่นขึ้นมาสำหรับการเป็นตัวเลือกในการทำงานร่วมกันเป็นทีม ซึ่ง Figma นั้นจะช่วยลดความซับซ้อนของกระบวนการออกแบบและมีประสิทธิภาพมากกว่าโปรแกรมอื่น ๆ ในการช่วยให้นักออกแบบและทีมสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ [20]

เนื่องจาก Figma นั้นทำงานบนเบราว์เซอร์ ข้อดีหนึ่งก็คือ Figma สามารถใช้งานได้บนหลากหลายเบราว์เซอร์บนระบบปฏิบัติการที่แตกต่างกัน เช่น macOS, Windows หรือ Linux และทำให้ทีมออกแบบสามารถทำงานร่วมกันได้ง่ายโดยจะมีสถานะบอกว่าใครกำลังทำอะไร [20]

**3.10 Visual Studio Code**

Visual Studio Code หรือ VS Code เป็นโปรแกรมที่ใช้เขียนโค้ดที่สามารถใช้ได้หลากหลายภาษา ถูกพัฒนาโดย Microsoft สามารถใช้งานได้บน Windows, Linux และ macOS โดย VS Code มีคุณสมบัติต่าง ๆ เช่น Debugging, Syntax Highlighting, Intelligent Code Completion, Snippets, Code Refactoring และ Embedded Git และผู้ใช้สามารถเปลี่ยนธีม, คีย์ลัด, การตั้งค่าต่าง ๆ และสามารถติดตั้งส่วนขยายที่เพิ่มฟังก์ชันในการทำงานได้ [21]

**3.11 Docker**

Docker เป็นแพลตฟอร์มซอฟต์แวร์ที่ช่วยให้สามารถสร้าง ทดสอบ และติดตั้งแอปพลิเคชันได้อย่างรวดเร็ว โดย Docker จะใส่ซอฟต์แวร์ต่าง ๆ ลงไปในหน่วยที่เป็นมาตรฐานที่เรียกว่า คอนเทนเนอร์ ซึ่งจะมีทุกอย่างที่ซอฟต์แวร์นั้น ๆ จำเป็นต้องใช้ในการทำงาน รวมทั้งไลบรารี เครื่องมือสำหรับระบบ โค้ด และรันไทม์ การใช้ Docker จะทำให้สามารถติดตั้งและปรับขนาดของแอปพลิเคชันให้เหมาะสมกับสภาพแวดล้อมต่าง ๆ ได้ง่ายขึ้น [22]

**4. โครงงานหรืองานวิจัยทีเกี่ยวข้อง (Literature Review)**

**4.1 โครงงานการพัฒนาเว็บแอปพลิเคชันการฝึกงานของปีการศึกษา 2565**

เว็บแอปพลิเคชันที่ทำหน้าที่ดำเนินเอกสารและรวบรวมทุกขั้นตอนที่เกี่ยวข้องกับการดำเนินการในการฝึกงานของนักศึกเอาไว้ในแพลตฟอร์มเดียว ซึ่งมีตั้งแต่ขั้นตอนการ ลงประกาศรับสมัคร นักศึกษาฝึกงาน การแจ้งความประสงค์ในการฝึกงานการดำเนินเอกสารขอ ความอนุเคราะห์ฝึกงานไปจนถึง ขั้นตอนการประเมินฝึกงาน

**ข้อดี**

1. มีความปลอดภัยค่อนข้างดีเนื่องจากทุกคนต้องล็อกอิน

2. เว็บไซต์เข้าใจง่าย

**ข้อเสีย**

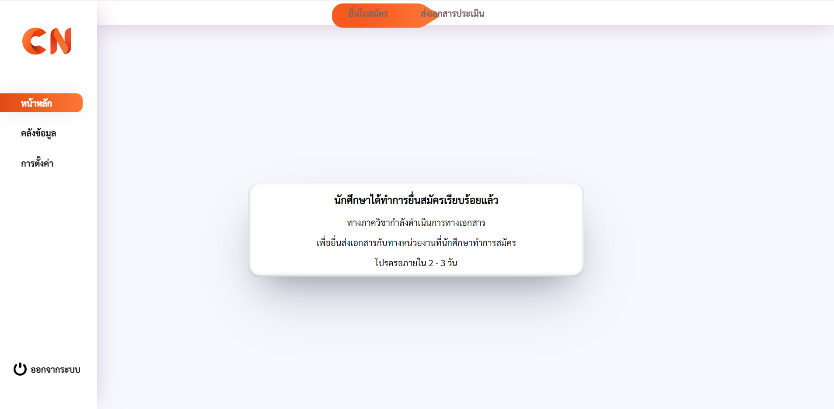
1. ระบบไม่สมบูรณ์มีหลายฟังก์ชันที่ใช้งานไม่ได้

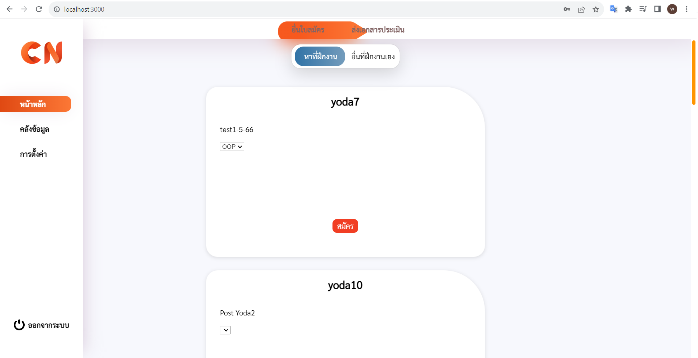
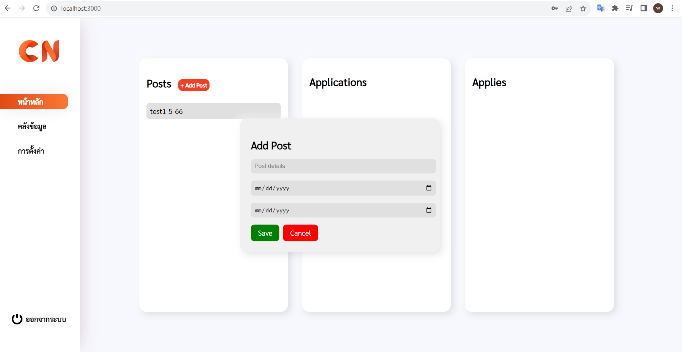
2. หน้าเว็บถึงจะเข้าใจง่ายเเต่ไม่สวยงาม

3. ไม่สามารถเข้าไปดูโพสต์รับสมัครงานได้หากไม่ได้เป็นสมาชิกหรือมีสิทธิ์

4. โพสต์รับสมัครงานมีความละเอียดของเนื้อหาน้อย

5. ขั้นตอนการทำงานของระบบไม่สมบูรณ์





รูปที่ 4.1 ภาพตัวอย่างแอปพลิเคชันการฝึกงาน 2565

**4.2 เว็บเด็กฝึกงาน.com**

เป็นแหล่งรวบรวมข้อมูล ฝึกงาน องค์กร บริษัทที่รับนักศึกษาฝึกงาน ทุกสาขาไว้ทั่วประเทศ และยังมีเครื่องมือช่วยในการสร้างเรซูเม่ อีกทั้งยังรวมประกาศโครงการต่าง ๆ และยังรวบรวมบทความที่เกี่ยวข้องกับการสมัครงาน

**ข้อดี**

1. UI มีรูปแบบที่ใช้งานง่าย

2. มีเครื่องมือช่วยในการสร้างเรซูเม่

3. มีบทความที่มีประโยชน์ช่วยในการสัมภาษณ์และการสมัครงาน

4. เว็บรวบรวมข้อมูลรับสมัครงานหลากหลายสาขาจากบริษัททั่วประเทศ

**ข้อเสีย**

1. บริษัทต้องสร้างโพสต์รับสมัครงานแยกตามแต่ละตำแหน่ง

2. หน้าเว็บถึงจะเข้าใจง่ายเเต่ขาดความน่าสนใจ

A screenshot of a computer

Description automatically generatedรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ใบหน้าของมนุษย์, เว็บไซต์

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 4.2 ภาพประกอบเว็บเด็กฝึกงาน.com

จากการศึกษาเว็บเด็กฝึกงาน.com และโครงงานการพัฒนาเว็บแอปพลิเคชันการฝึกงานของปีการศึกษา 2565 พบว่าโครงงานการพัฒนาเว็บแอปพลิเคชันการฝึกงานของปีการศึกษา 2565 มีการวางระบบ และพัฒนาแอปพลิเคชันด้วย React และ Node.JS ซึ่งเกี่ยวกับกับโครงงานที่ผู้จัดทำได้จัดทำขึ้น

โดยเมื่อเปรียบเทียบ เว็บเด็กฝึกงาน.com กับโครงงานการพัฒนาเว็บแอปพลิเคชันการฝึกงานของปีการศึกษา 2565 ในหลาย ๆ ส่วนของโครงงานมีความไม่สมบูรณ์ ทั้งในเรื่องขั้นตอนการทำงานที่ไม่สมบูรณ์ ระบบโพสต์รับสมัครงานที่ใส่รายละเอียดได้ไม่ครบถ้วน ปัญหาการจัดการตำแหน่งงานที่เปิดรับ นักศึกษาและบริษัทไม่สามารถสร้างโปรไฟล์หรืออัปโหลดเรซูเม่หรือเอกสารได้ ไม่มีระบบจัดการสิทธิ์ในการสมัครงานที่ดี ไม่มีระบบสร้างเอกสารอัตโนมัติ และในส่วนของการเขียนโค้ด จากในโครงงานที่ผู้จัดทำได้ศึกษานั้น โค้ดส่วนใหญ่ทำเสร็จในส่วนของ Back-End ซึ่งก็คือ Node.JS เเต่ Front-End นั้นยังทำไปเเค่บางส่วนซึ่งยังขาดอีกจำนวนหลายหน้า ดังนั้นผู้จัดทำโครงงานจึงได้นำแนวทางในการพัฒนาเว็บแอปพลิเคชันการฝึกงาน และขั้นตอนการทำงานของระบบมาปรับใช้ในการพัฒนาระบบเว็บแอปพลิเคชันการฝึกงาน ให้มีความสมบูรณ์ เพิ่มความละเอียดของแต่ละฟีเจอร์ และสามารถทำงานได้ครบทุกกระบวนการของการฝึกงาน

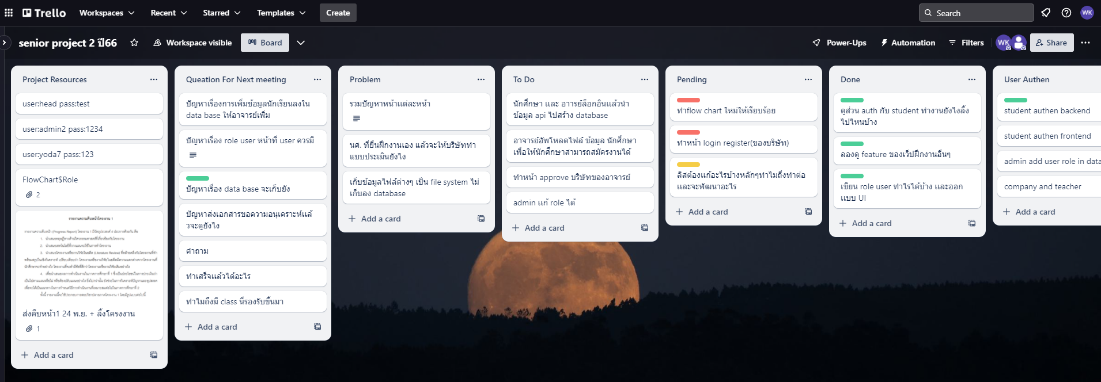
**5. ผลการดำเนินงาน**

เเบ่งหน้าที่การพัฒนาเว็บแอปพลิเคชันในส่วน Front-End และ Back-End ดังนี้

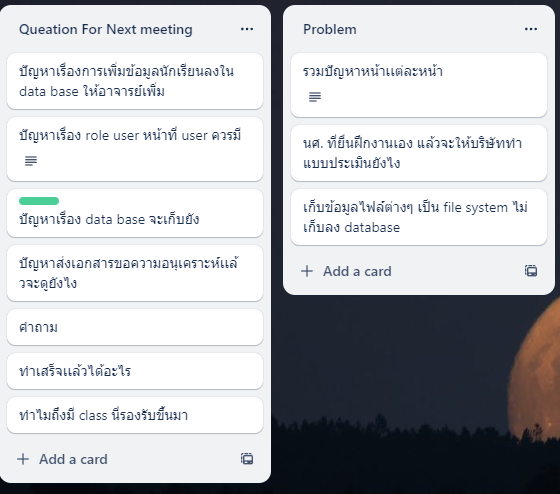
- นายภูชิชย์ กลีบมาลัย มีหน้าที่พัฒนาในส่วน Front-End เป็นหลัก

- นายวงศกร กองกะมุด มีหน้าที่พัฒนาในส่วน Back-End เป็นหลัก

**5.1 วางแผนการทำงานโดยใช้ Trello ในการช่วยในการวางแผนการทำงาน**

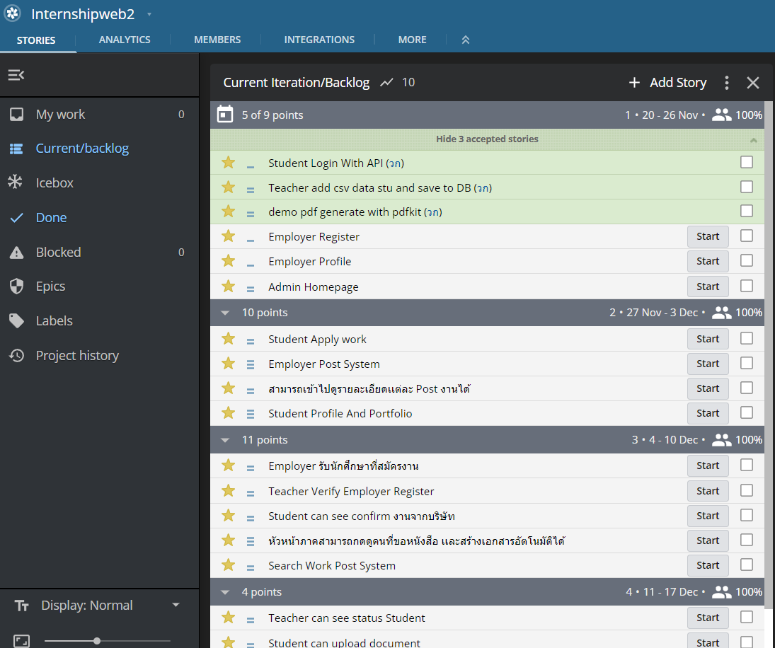


รูปที่ 5.1 ภาพประกอบการใช้งาน Trello

**5.2 ในระหว่างการพัฒนาทางทีมพัฒนาจะเข้าไปพบกับอาจารย์ที่ปรึกษาโครงงานทุกสัปดาห์ซึ่งทางทีมพัฒนาจะมีการคิดปัญหาหรือข้อสงสัยเตรียมไว้ไปถามอาจารย์ที่ปรึกษา และวางแผนต่อไปว่าสัปดาห์ถัดไปจะทำอะไร**

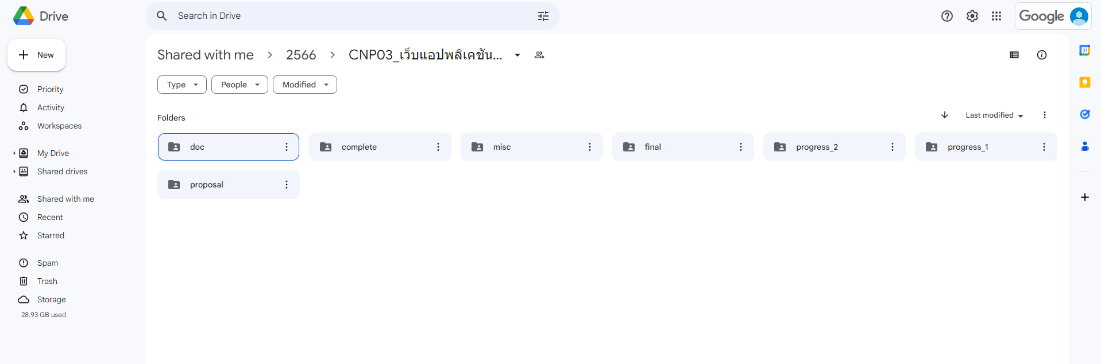
รูปที่ 5.2 ภาพประกอบการใช้งาน Trello

**5.3 ใช้งาน Pivotal Tracker ในการวางแผนพัฒนาระบบตาม Iteration**

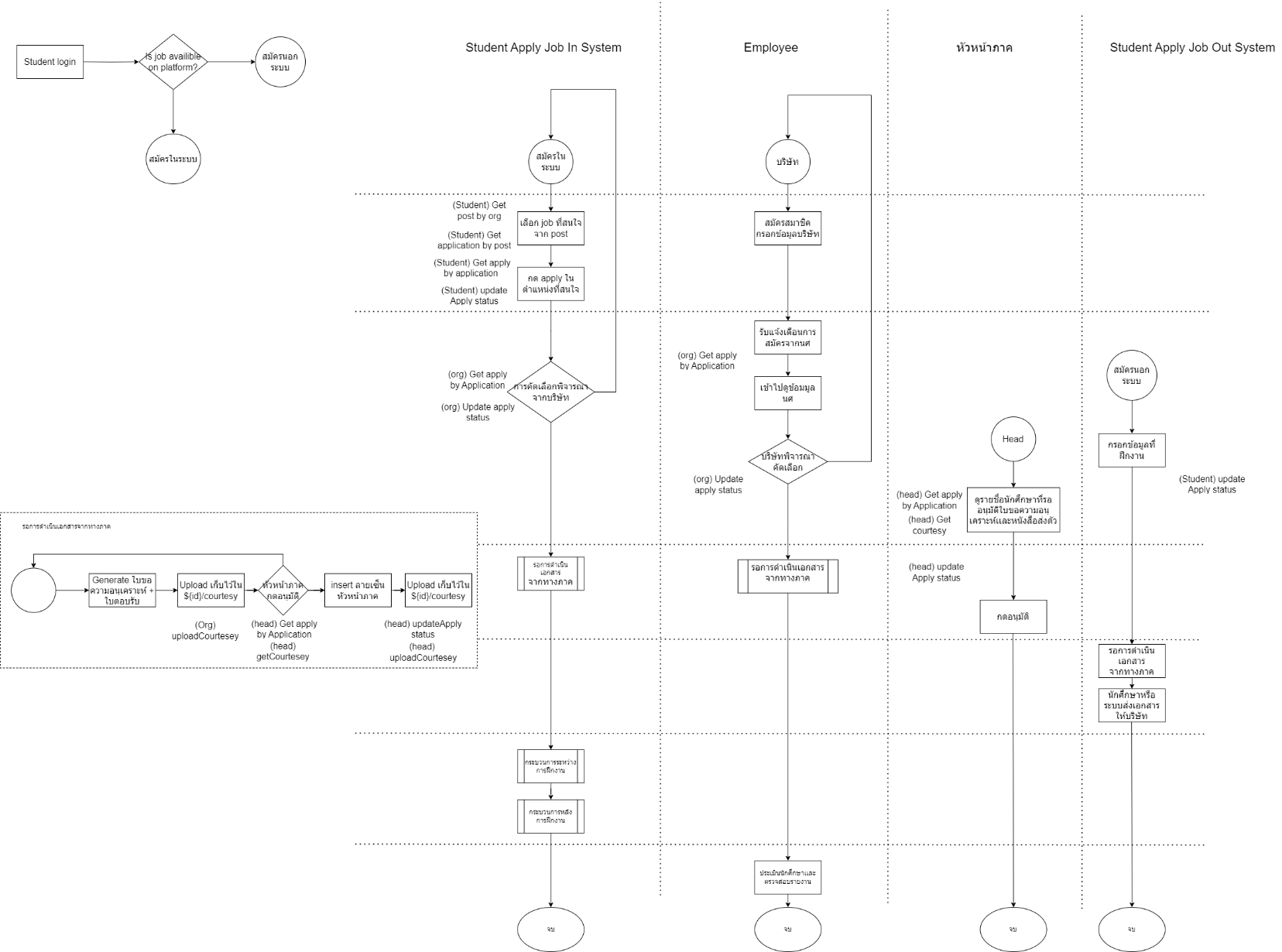


รูปที่ 5.3 ภาพประกอบการใช้งาน Pivotal Tracker

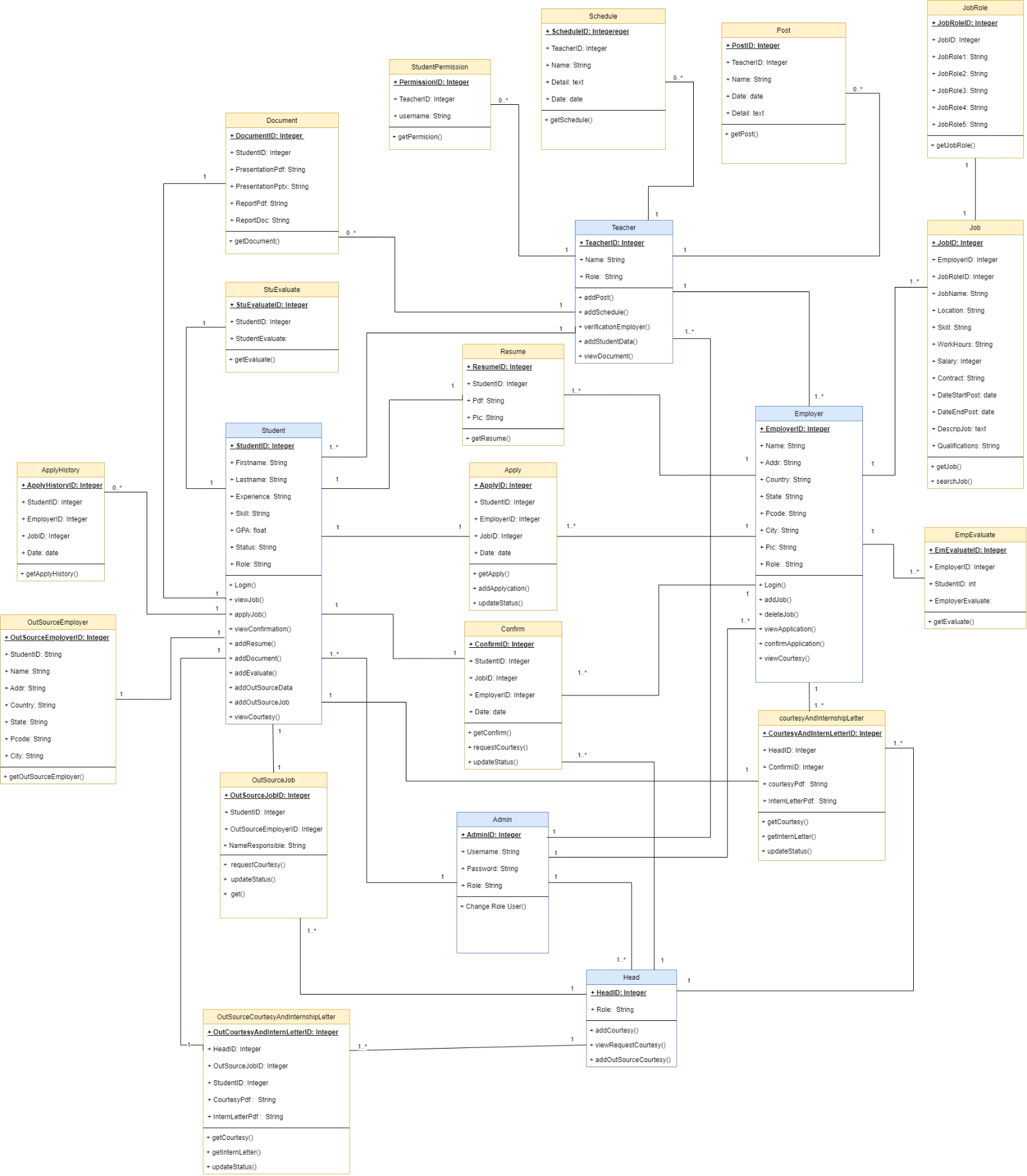
**5.4 ใช้งาน Google Drive ในการรวบรวมเอกสารหรือไฟล์งานที่เกี่ยวข้อง**



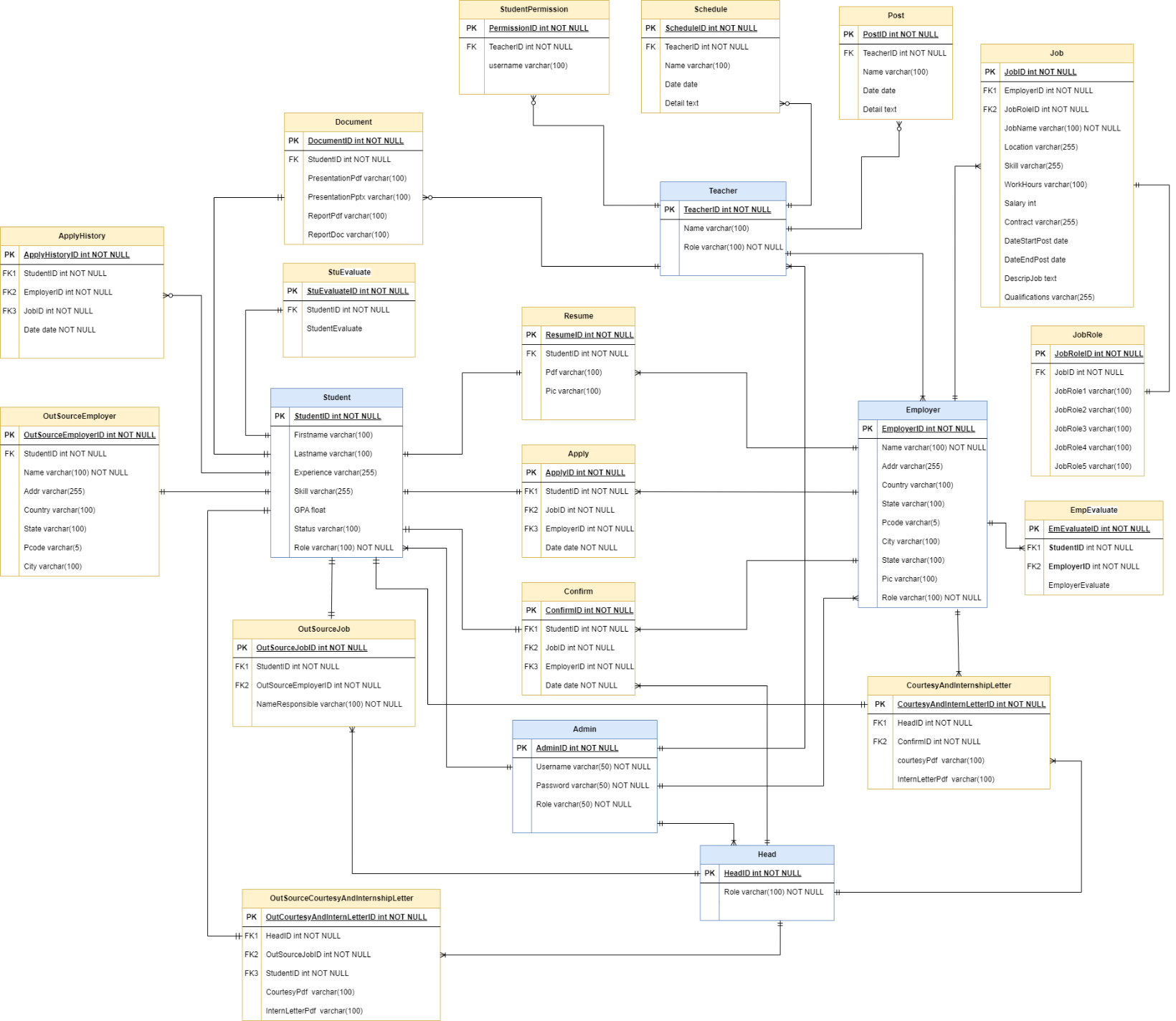
รูปที่ 5.4 ภาพประกอบการใช้งาน Google Drive

**5.5 ออกแบบขั้นตอนการทำงานของระบบ**

รูปที่ 5.5 ภาพตัวอย่าง flow chart

**5.6 ออกแบบ Class Diagram**

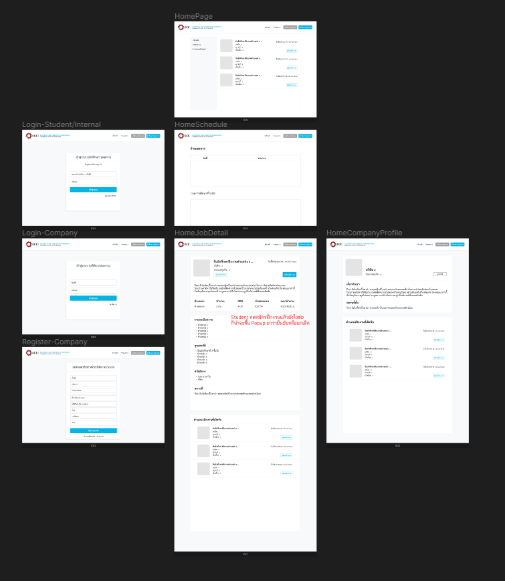
รูปที่ 5.6 ภาพตัวอย่าง Class Diagram

**5.7 ออกแบบ ER Diagram**

รูปที่ 5.7 ภาพตัวอย่าง ER Diagram

**5.8 ทำการออกแบบ User Interface ในโปรแกรม Figma**

รูปภาพประกอบด้วย ข้อความ, ซอฟต์แวร์, ออกแบบ, ภาพหน้าจอ

คำอธิบายที่สร้างโดยอัตโนมัติ

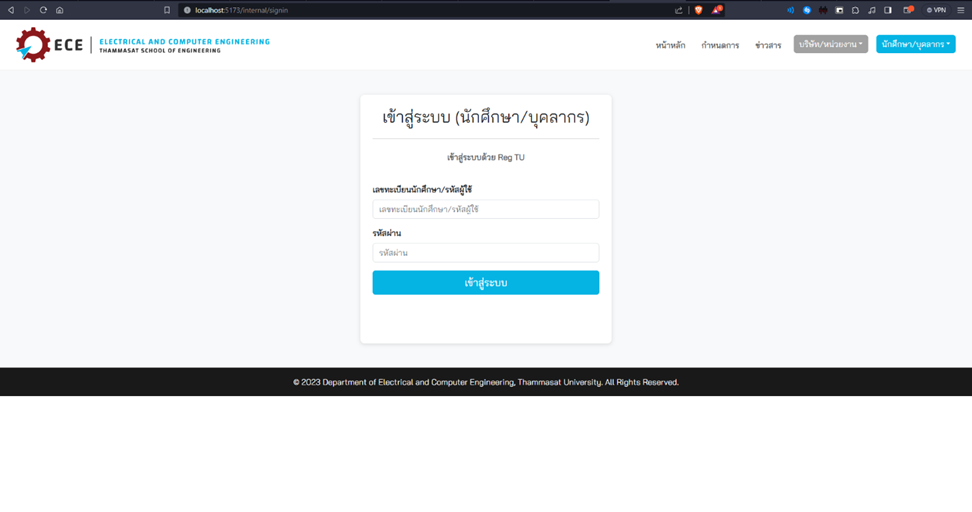
รูปที่ 5.8 ภาพประกอบบางส่วนของการออกแบบ UI ด้วยโปรแกรม Figma

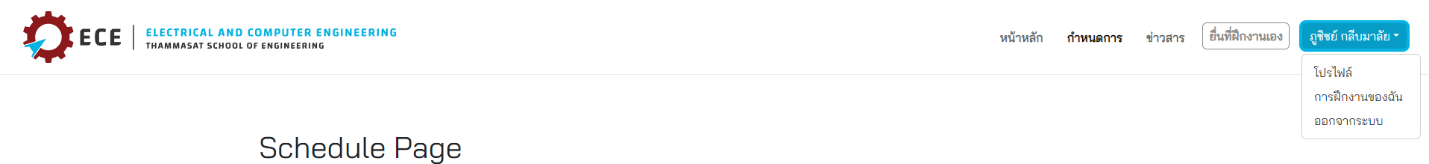
**5.9 ในส่วน Front-End ได้ทำการพัฒนาส่วนของโค้ดด้วย React โดยสร้าง Component พื้นฐานที่ต้องนำไปใช้ซ้ำ และสร้างโครงสร้างของหน้าหลัก ๆ เช่น หน้าหลัก, หน้าลงชื่อเข้าใช้ เป็นต้น**

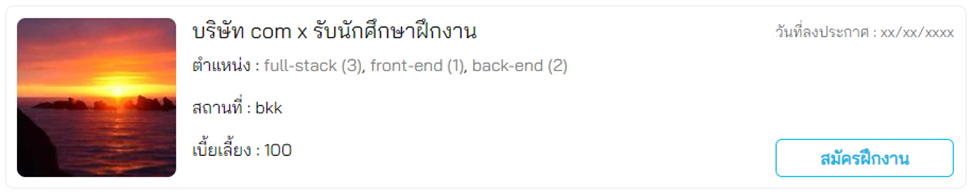
รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, หน้าเว็บ

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 5.9 ภาพหน้าหลักของเว็บไซต์

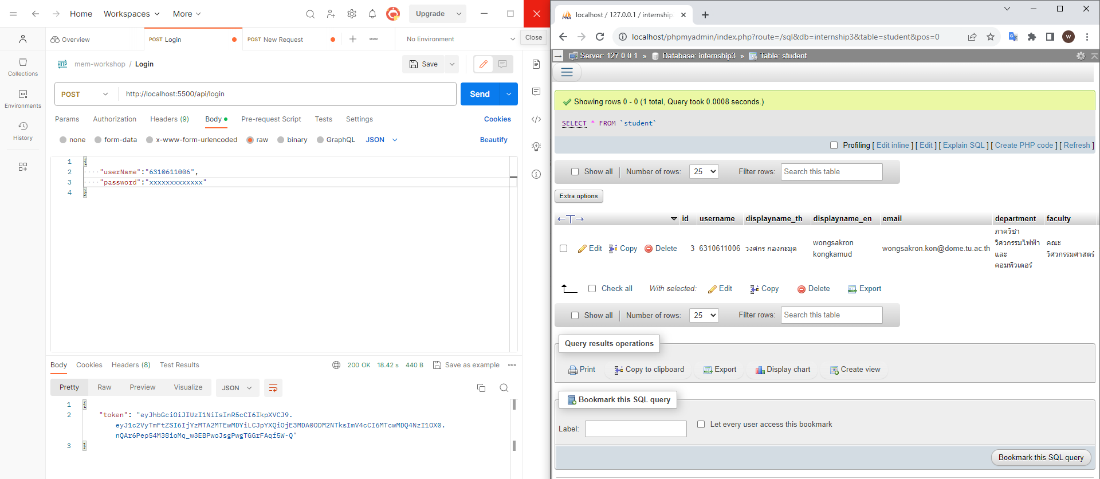


รูปที่ 5.10 ภาพหน้าลงชื่อเข้าใช้ของเว็บไซต์

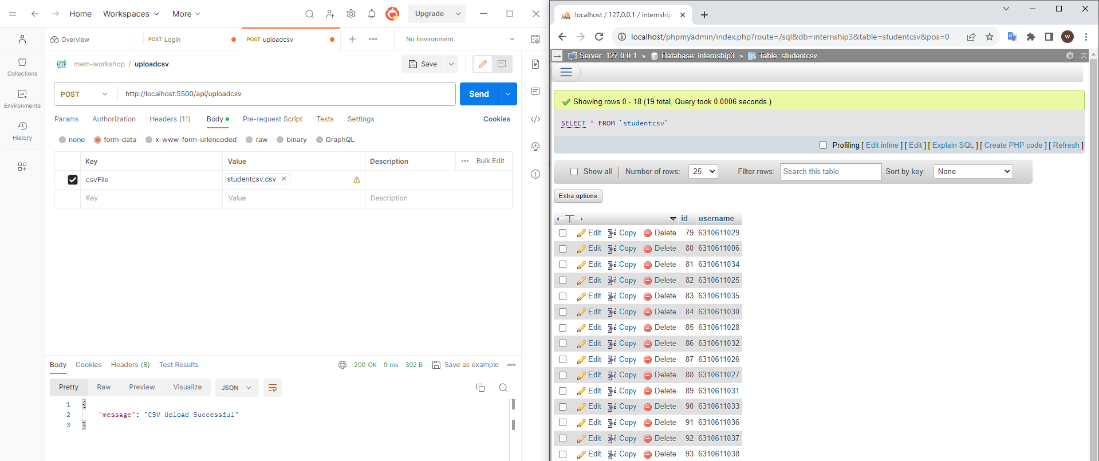


รูปที่ 5.11 ภาพตัวอย่าง Components ที่ได้ทำการสร้าง

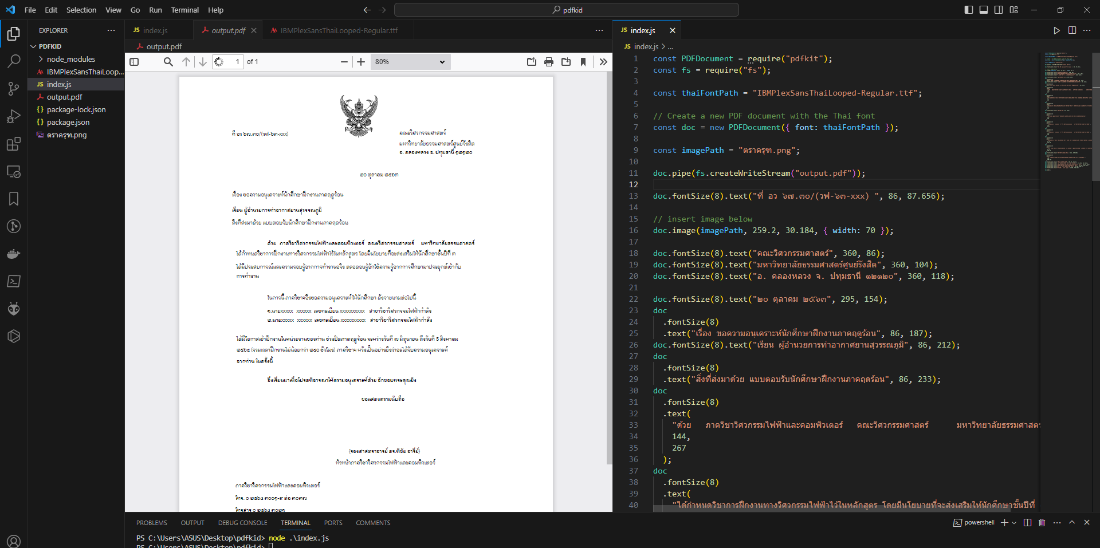
**5.10 ในส่วน Back-End ได้มีการสร้าง Login ด้วย TU API จากนั้นจะนำข้อมูลจาก API ไปเพิ่มในฐานข้อมูลหากยังไม่มี username นั้น**



รูปที่ 5.12 ภาพประกอบการเข้าสู่ระบบด้วย TU API

**5.11 ในส่วน Back-End ได้มีการสร้าง upload csv สำหรับการเพิ่มรายชื่อนักศึกษาที่สามารถลงสมัครงานได้**

รูปที่ 5.13 ภาพประกอบการอัปโหลดไฟล์ csv

**5.12 ในส่วน Back-End ได้มีการทดลองสร้าง Pdf Generate ด้วย PDFkit**

รูปที่ 5.14 ภาพประกอบการทดลองสร้าง Pdf ด้วย PDFKit

**6. ปัญหาและอุปสรรค**

6.1 มีการปรับเปลี่ยน Workflow และ Class Diagram หลายครั้งเพื่อให้ครอบคลุมการทำงาน

6.2 ปัญหา Dependency จากเนื้อหาที่เรียนกับเวอร์ชันที่นำมาใช้ในการพัฒนาแอปพลิเคชัน

6.3 โครงสร้างของฐานข้อมูลมีความซับซ้อนจากการเก็บข้อมูลที่หลากหลาย เช่น ไฟล์เอกสารต่าง ๆ

6.4 ปัญหาการจัดการสิทธิ์ให้กับผู้ใช้ เช่น เลขรหัสนักศึกษาใดบ้างที่สามารถสมัครงานได้

6.5 อุปสรรคจากการเรียนรู้ภาษาใหม่ ทำให้ทำงานได้ช้า เนื่องจากต้องเเบ่งเวลาไปเรียนรู้

**7. แผนดำเนินงานต่อไป**

7.1 พัฒนาแอปพลิเคชันทั้งในส่วนของ Client และ Server ดังต่อไปนี้

7.1.1 เริ่มพัฒนาในส่วนของผู้ใช้งานที่เป็นบริษัท สามารถโพสต์รับสมัครงาน

7.1.2 ผู้ใช้งานที่เป็นหัวหน้าภาคสามารถอนุมัติคำร้องสร้างเอกสารอัตโนมัติได้

7.1.3 ผู้ใช้งานที่เป็นนักศึกษาสามารถสมัครงานได้

7.2 พัฒนาระบบการสร้างเอกสารอัตโนมัติ โดยปรับเปลี่ยนข้อมูลเอกสารตามข้อมูลที่ได้รับมา

7.3 ทดสอบระบบ และนำไปปรับใช้งานบนเซิร์ฟเวอร์

7.4 ดำเนินงานตามตารางแผนการดำเนินงาน

**ตารางที่ 1** แผนการดำเนินงานช่วงเดือนสิงหาคมถึงเดือนธันวาคม

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **แผนการดำเนินงาน** | สิงหาคม | | | | กันยายน | | | | ตุลาคม | | | | พฤศจิกายน | | | | | ธันวาคม | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | | 2 | 3 | 4 |
| ศึกษาการทำงานของระบบเก่า | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| ศึกษาการทำงาน Node JS เเละ React JS |  |  |  | x | x | x | x | x |  |  |  |  |  |  |  |  |  | |  |  |  |
| ออกเเบบ flow การทำงานเพิ่มเติม |  |  |  |  |  |  |  |  | x | x |  |  |  |  |  |  |  | |  |  |  |
| ออกเเบบ User-Interface สำหรับระบบ |  |  |  |  |  |  |  |  |  |  | x | x |  |  |  |  |  | |  |  |  |
| ปรับปรุงฐานข้อมูลสำหรับระบบ |  |  |  |  |  |  |  |  |  |  |  |  | x | x |  |  |  | |  |  |  |
| สร้าง Front-end สำหรับระบบ พร้อม ปรับปรุง Back-end |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | | x | x | x |

**ตารางที่ 2** แผนการดำเนินงานช่วงเดือนมกราคมถึงเดือนพฤษภาคม

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **แผนการดำเนินงาน** | มกราคม | | | | กุมภาพันธ์ | | | | มีนาคม | | | | เมษายน | | | | พฤษภาคม | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| สร้าง Front-end สำหรับระบบ พร้อม ปรับปรุง Back-end | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |
| ทดสอบระบบเเละเเก้ไขข้อผิดพลาด |  |  |  |  |  |  |  |  | x | x | x | x |  |  |  |  |  |  |  |  |
| ปรับใช้ระบบบน server |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x |  |  |  |  |
| จัดทำรายงานโครงงานเเละคู่มือใช้งาน |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x |

**8. บรรณานุกรม**

[1] “Web Application Development in 2023 - A Detailed Guide.” kissflow.com. https://kissflow.com/application-development/web-application-development ‌(accessed Nov. 17, 2023).

[2] “Web Application Development.” atatus.com. https://www.atatus.com/glossary/web-application-development (accessed Nov. 17, 2023).

[3] “Introduction to JSON Web Tokens.” jwt.io. https://jwt.io/introduction (accessed Nov. 18, 2023).

[4] “What is Middleware?.” aws.amazon.com. https://aws.amazon.com/what-is/middleware (accessed Nov. 18, 2023).

[5] “MVC Framework - Introduction.” tutorialspoint.com. https://www.tutorialspoint.com/mvc\_framework/mvc\_framework\_introduction.htm (accessed Nov. 18, 2023).

[6] P. Gillin. “What is Component-Based Architecture?.” mendix.com. https://www.mendix.com/blog/what-is-component-based-architecture (accessed Nov. 18, 2023).

[7] R. Preston. “What Is Asynchronous Programming? (And When To Use It).” indeed.com. https://www.indeed.com/career-advice/career-development/asynchronous-programming (accessed Nov. 18, 2023).

[8] “What is a RESTful API?.” aws.amazon.com. https://aws.amazon.com/what-is/restful-api (accessed Nov. 18, 2023).

[9] “What is SQL injection (SQLi)?.” portswigger.net. https://portswigger.net/web-security/sql-injection (accessed Nov. 18, 2023).

[10] B. Semah. “What Exactly is Node.js? Explained for Beginners.” freecodecamp.org. https://www.freecodecamp.org/news/what-is-node-js (accessed Nov. 17, 2023).

[11] Codecademy Team. “What is Express.js?.” codecademy.com. https://www.codecademy.com/article/what-is-express-js (accessed Nov. 17, 2023).

[12] “What is MySQL?.” dev.mysql.com. https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html (accessed Nov. 17, 2023).

[13] “What is MySQL?.” oracle.com. https://www.oracle.com/mysql/what-is-mysql (accessed Nov. 17, 2023).

[14] “What is a Relational Database (RDBMS)?.” oracle.com. https://www.oracle.com/th/database/what-is-a-relational-database (accessed Nov. 21, 2023).

[15] “What is XAMPP?.” educba.com. https://www.educba.com/what-is-xampp (accessed Nov. 17, 2023).

[16] “About Postman.” postman.com. https://www.postman.com/company/about-postman (accessed Nov. 17, 2023).

[17] D. Herbert. “What is React.js? Uses, Examples, & More.” blog.hubspot.com. https://blog.hubspot.com/website/react-js (accessed Nov. 17, 2023).

[18] “Bootstrap Tutorial.” geeksforgeeks.org. https://www.geeksforgeeks.org/bootstrap (accessed Nov. 18, 2023).

[19] “PDFKit.” pdfkit.org. https://pdfkit.org (accessed Nov. 18, 2023).

[20] B. Kopf. “The Power of Figma as a Design Tool.” toptal.com. https://www.toptal.com/designers/ui/figma-design-tool (accessed Nov. 18, 2023).

[21] “Visual Studio Code.” en.wikipedia.org. https://en.wikipedia.org/wiki/Visual\_Studio\_Code (accessed Nov. 18, 2023).

[22] “What is Docker?.” aws.amazon.com. https://aws.amazon.com/docker (accessed Nov. 18, 2023).