

บริษัท สเปซแทร็กซ์ จำกัด (SPACETRAX CO., LTD)

เลขทะเบียนนิติบุคคล: 0505564016926

ที่อยู่: 87/9, สตาร์เวิร์ค, ชั้น 2, ห้อง 209, พุ่งโฮเต็ล, วัดเกต, เมือง, เชียงใหม่ 50000

## ประสบการณ์จากการฝึกงาน

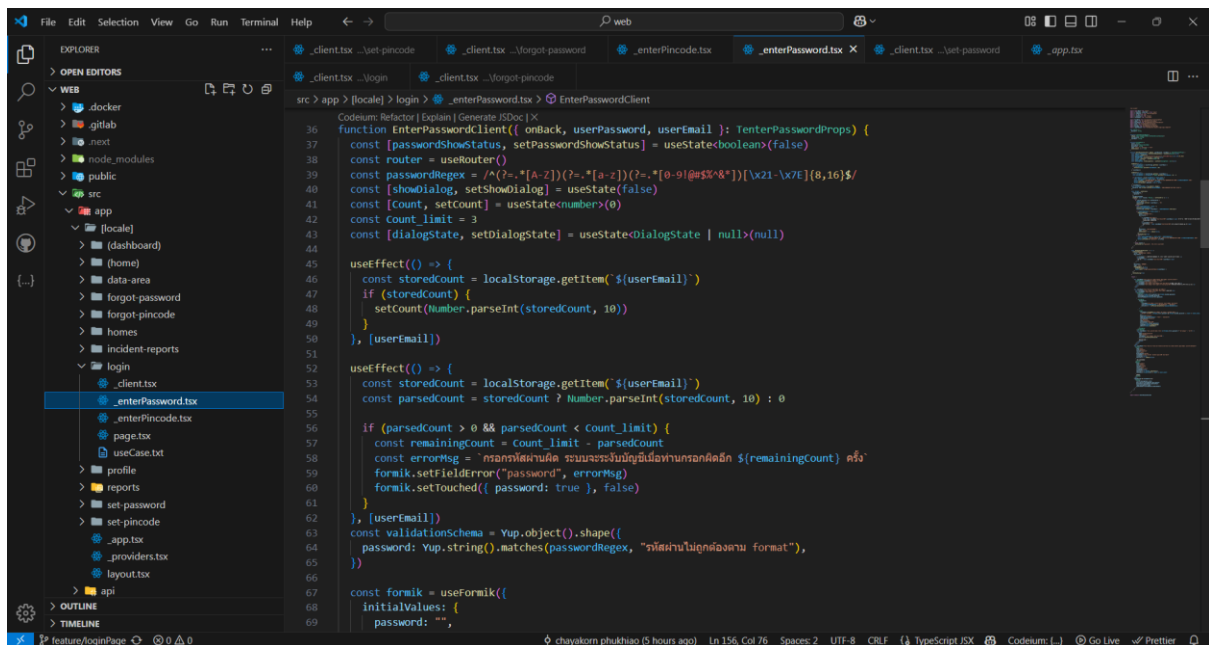
เริ่มต้นฝึกงานในตำแหน่ง Frontend Developer และ UX/UI Designer และปัจจุบันได้ร่วมทำงานกับทีมพัฒนาในโปรเจกต์จริงของบริษัทในตำแหน่ง Software Developer (Frontend Developer) โดยมีหน้าที่หลักคือ พัฒนาโครงสร้างของระบบให้เหมือนตามทีออกแบบใน Figma เชื่อมต่อ API และเขียน Flow การทำงานให้เป็นไปตามที่บริษัทกำหนด เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพและสอดคล้องกับความต้องการขององค์กร

## 1. ภาษาที่ใช้ในการพัฒนา

HTML, JavaScript, TypeScript – ใช้ในการพัฒนาโครงสร้างและการทำงานของเว็บไซต์

## 2. เทคโนโลยี Framework / Library ที่ใช้ในการพัฒนา

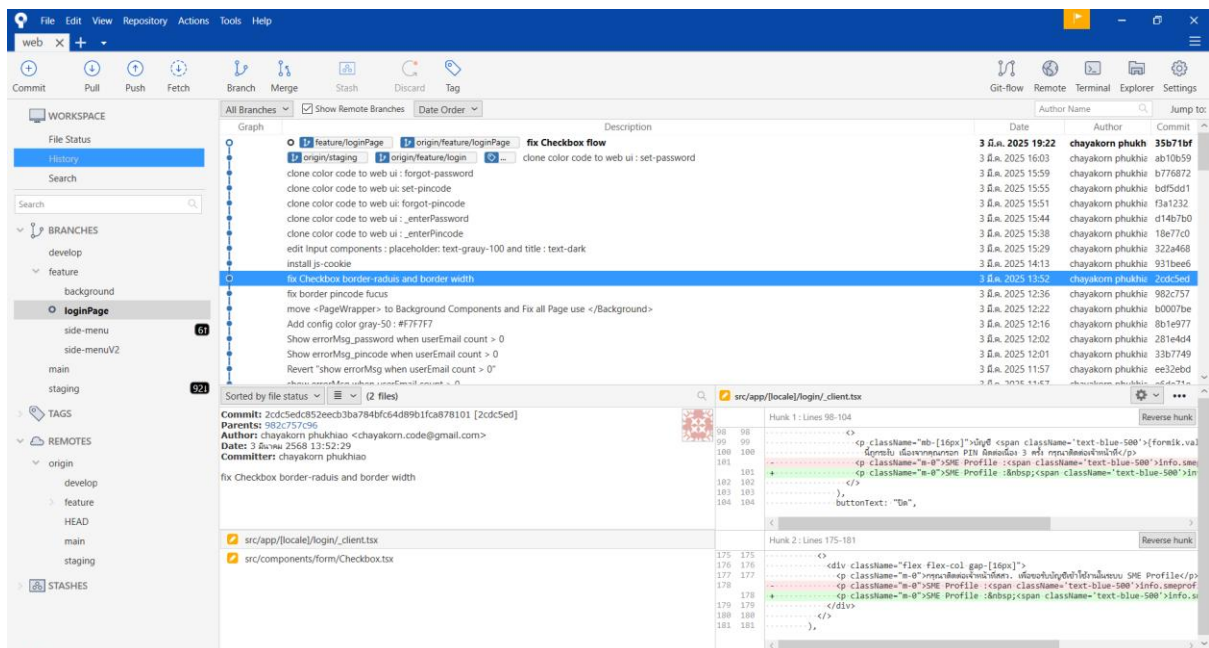
2.1 Visual Studio Code เครื่องมือหลักที่ใช้สำหรับการพัฒนา Software โดยรองรับส่วนเสริม (Extensions) ต่าง ๆ ที่ช่วยเพิ่มประสิทธิภาพในการทำงาน เช่น Git Integration, Debugging และ Auto-Completion ทำให้สามารถพัฒนาโค้ดได้สะดวกและรวดเร็วขึ้น



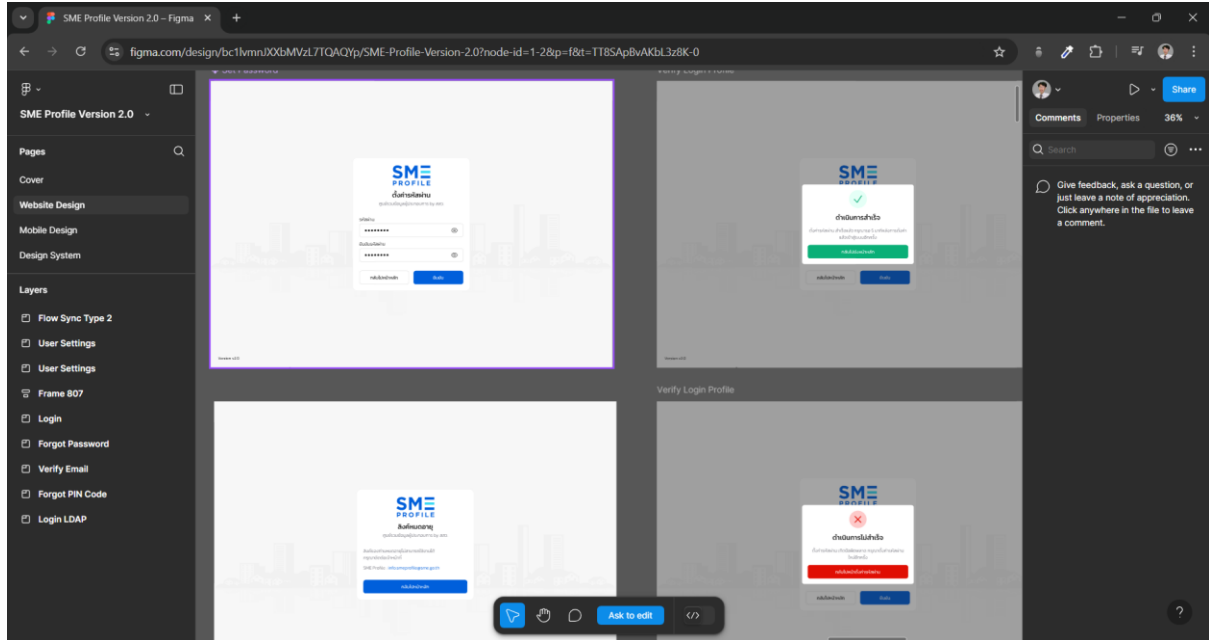
2.2 Next.js เป็น React Framework ใช้ในการพัฒนาเว็บแอปพลิเคชัน โดยเน้นการศึกษาด้าน State Management เพื่อควบคุมข้อมูลในแอปพลิเคชันให้เป็นระเบียบ รวมถึงการใช้ Routing และ Parameter Handling เพื่อจัดการเส้นทางของหน้าเว็บ ทำให้สามารถพัฒนาแอปพลิเคชันที่มีประสิทธิภาพ และโหลดเร็วขึ้น

2.3 GitLab ใช้การทำงานร่วมกับทีมพัฒนาใช้ เป็นเครื่องมือสำหรับจัดการเวอร์ชันของโค้ด โดยรองรับการทำงานร่วมกันแบบเป็นระบบ เช่น การสร้าง Branch สำหรับพัฒนาแต่ละฟีเจอร์ การ Merge Request เพื่อรีวิวโค้ด และระบบ CI/CD (Continuous Integration / Continuous Deployment) ที่ช่วยให้สามารถอัปเดตโค้ดได้อย่างต่อเนื่องและปลอดภัย

นอกจากนี้ยังใช้ SourceTree เป็นเครื่องมือ GUI สำหรับบริหารจัดการโค้ด ทำให้การดึง (Pull), ส่ง (Push) และรวมโค้ด (Merge) เป็นไปอย่างสะดวกและลดข้อผิดพลาดในการทำงาน

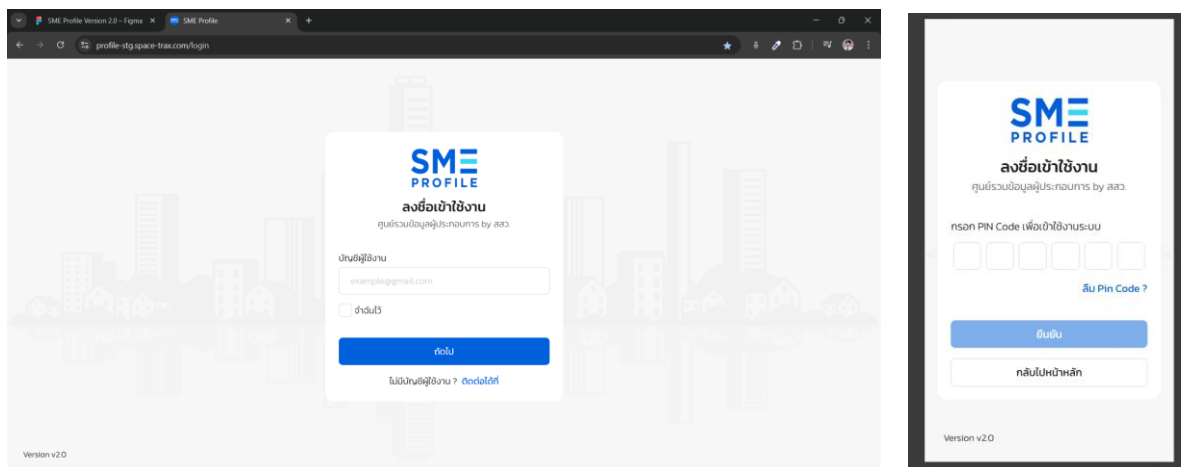


**2.5 Figma** ใช้สำหรับออกแบบ UI และสร้าง Prototype เพื่อให้การออกแบบตรงกับความต้องการของระบบเป็นเครื่องมือหลักในการสร้าง Wireframe, Prototype และออกแบบส่วนติดต่อผู้ใช้ (User Interface) ซึ่งช่วยให้สามารถออกแบบระบบได้ตรงกับความต้องการ และสามารถทำงานร่วมกับทีมพัฒนาได้อย่างมีประสิทธิภาพ



**2.6 Formik** ใช้ในส่วนของการจัดการฟอร์ม เป็นไลบรารีที่ช่วยให้สามารถสร้างฟอร์มที่รองรับการตรวจสอบข้อมูล (Validation) ได้ง่ายขึ้น โดยสามารถกำหนดรูปแบบข้อมูลที่ได้รับเข้ามาและส่งไปยังเซิร์ฟเวอร์ได้อย่างสะดวกและปลอดภัย

**2.7 Tailwind CSS** ใช้ในการการตกแต่งและออกแบบส่วนแสดงผลของเว็บไซต์ใช้ Tailwind CSS ซึ่งเป็น CSS Framework ที่ใช้ Utility Classes ทำให้สามารถออกแบบ UI ได้อย่างรวดเร็ว โดยไม่ต้องเขียน CSS เองมากนัก และยังช่วยให้เว็บไซต์สามารถรองรับ Responsive Design ได้ง่าย



**2.8 Redux.js** ใช้การจัดการข้อมูลและสถานะของแอปพลิเคชันใช้ Redux.js เพื่อช่วยควบคุมข้อมูลให้อยู่ในรูปแบบที่เป็นระบบ ลดปัญหา Prop Drilling และทำให้สามารถแชร์ข้อมูลระหว่าง Components ได้อย่างมีประสิทธิภาพ นอกจากนี้ยังสามารถใช้ Middleware เช่น Redux Thunk หรือ Redux Saga เพื่อจัดการข้อมูลแบบ Asynchronous ได้อย่างมีประสิทธิภาพ

### 3. แนวคิดและหลักการในการพัฒนา

**3.1 Clean Architecture** เป็นแนวคิดในการออกแบบโครงสร้างซอฟต์แวร์ที่ช่วยให้ระบบมีความยืดหยุ่น ง่ายต่อการดูแลรักษา และรองรับการขยายตัวของโค้ดในอนาคต โดยแนวคิดหลักของ Clean Architecture คือการแยกส่วนต่าง ๆ ของระบบออกจากกันอย่างชัดเจน และลดการพึ่งพาของแต่ละเลเยอร์ ซึ่งช่วยให้โค้ดมีความเป็นโมดูล และสามารถทดสอบหรือเปลี่ยนแปลงบางส่วนของระบบได้โดยไม่กระทบกับส่วนอื่น ๆ

#### Clean Architecture มาใช้ใน Frontend Development

- แยก UI Components ออกจาก Business Logic
- ใช้ State Management (เช่น Redux) ในการจัดการข้อมูล
- ใช้ Hooks หรือ Custom Hooks แทนการเรียก API ตรงใน Component
- ใช้ Service Layer สำหรับการดึงข้อมูลจาก API

