



การพัฒนาเว็บไซต์บริษัททูล แมเนจเมนท์
Tool Management Website

ชัยวัฒน์ โจอทตะคุ รหัส 6340505211

ปฏิบัติงาน ณ
บริษัท เรเวล ซอฟต์แวร์ จำกัด
เลขที่ 271/55 ตรอกวัดท่าตะโก ตำบลในเมือง อำเภอเมืองนครราชสีมา
จังหวัดนครราชสีมา 30000

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชาสหกิจศึกษา
คณะวิทยาการจัดการ สาขาคอมพิวเตอร์ธุรกิจ
มหาวิทยาลัยราชภัฏนครราชสีมา
ปีการศึกษา 2566

การพัฒนาเว็บไซต์บริษัททุล แมเนจเม้นต์
Tool Management Website

ชัยวัฒน์ โจอิตะคุ รหัส 6340505211

ปฏิบัติงาน ณ
บริษัท เรเวล ซอฟต์ จำกัด
เลขที่ 271/55 ตรอกวัดท่าตะโก ตำบลในเมือง อำเภอเมืองนครราชสีมา
จังหวัดนครราชสีมา 30000

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชาสหกิจศึกษา
คณะวิทยาการจัดการ สาขาคอมพิวเตอร์ธุรกิจ
มหาวิทยาลัยราชภัฏนครราชสีมา
ปีการศึกษา 2566

หัวข้อโครงการ	การพัฒนาเว็บไซต์บริษัททูล แมเนจเม้นต์
นักศึกษา	นายชัยวัฒน์ โจงตะคุ รหัส 6340505211
ที่ปรึกษาโครงการ	นายจักรพงษ์ สุขทรัพย์เจริญ
สาขา	คอมพิวเตอร์ธุรกิจ
ปีการศึกษา	2566

บทคัดย่อ

ในส่วนของซอฟต์แวร์ที่เป็นเว็บไซต์ของบริษัท มีเป้าหมายที่ชัดเจนในการแจกจ่ายข่าวสาร และส่งเสริมการตลาดสำหรับบริการการบริหารจัดการเครื่องมือ (Tool Management) ของบริษัท ทำให้ผู้ใช้ทั่วไปสามารถเข้าชมและเรียนรู้เพิ่มเติมเกี่ยวกับผลิตภัณฑ์ และโรงงานอุตสาหกรรมการผลิตต่างๆ ที่เกี่ยวข้อง

การสร้างเว็บไซต์ที่สามารถทำงานได้อย่างมีประสิทธิภาพนั้น เรียกใช้ความรู้และความเข้าใจทางด้านภาษาคอมพิวเตอร์เพื่อการพัฒนาส่วนต่างๆของเว็บไซต์ เพื่อให้สามารถดึงข้อมูลจากเซิร์ฟเวอร์ และฐานข้อมูลไปแสดงผลบนหน้าจอได้อย่างถูกต้อง

โดยในการสร้างและพัฒนาเว็บไซต์ Tool Management ใช้เฟรมเวิร์ก Nuxt JS และ Vue JS เพื่อให้การพัฒนาเว็บไซต์เป็นไปอย่างรวดเร็วและมีประสิทธิภาพ เพื่อให้โครงการนี้สามารถนำไปใช้พัฒนาต่อได้ในอนาคตเพื่อใช้ในการสร้างเว็บไซต์ในสายอาชีพต่อไปได้โดยง่าย และมีประสิทธิภาพเช่นเดียวกัน

การออกแบบโครงสร้างเว็บไซต์ได้ถูกวางแผนอย่างเหมาะสมและมีการแบ่งส่วนการทำงานอย่างชัดเจน เช่น Header, Banner, Info และ Footer เพื่อให้ผู้ใช้ทั่วไปสามารถเข้าถึงข้อมูลและบริการได้อย่างสะดวกสบาย การนำเสนอผลงานของบริษัทที่มีความคล่องตัวและปรับได้ตามความต้องการของลูกค้า โดยใช้รูปแบบที่ทันสมัย ดูสบายตา เข้าถึงง่าย และมีความสวยงาม

คำสำคัญ : ซอฟต์แวร์, เว็บไซต์, Nuxt JS, Vue JS

กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัทเรเวล ซอฟต์แวร์ จำกัด Revel Soft Co., LTD ตั้งแต่วันที่ 12 ธันวาคม พ.ศ.2566 ถึงวันที่ 5 เมษายน พ.ศ.2567 ส่งผลให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีคุณค่ามากมาย สำหรับรายงานสหกิจศึกษานี้สำเร็จลงได้ด้วยดีจากความร่วมมือและการสนับสนุนจากหลายฝ่าย ดังนี้

1. ผศ.อภิสิทธิ์ เดิมสันเทียะ (อาจารย์นิเทศ)
2. นายสันติสุข สิริจินดาวิวัฒน์ (ผู้บริหาร)
3. นายจักรพงษ์ สุขทรัพย์เจริญ (ที่ปรึกษาสหกิจ)

นอกจากนี้ขอขอบคุณอาจารย์ และบุคคลท่านอื่น ๆ ที่ไม่ได้กล่าว ณ ที่นี้ซึ่งท่านเหล่านั้นได้กรุณาช่วยเหลือให้คำแนะนำในการจัดทำโครงการ และรายงานฉบับนี้ จึงขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ข้อมูลคำแนะนำ และวิธีการปฏิบัติงาน รวมถึงเป็นที่ปรึกษาในการจัดทำโครงการ และรายงานฉบับนี้จนเสร็จสมบูรณ์

ณ ที่นี้สุดท้ายนี้ หากมีสิ่งผิดพลาดหรือขาดตกบกพร่องประการใด ผู้จัดทำต้องขออภัยเป็นอย่างสูงในสิ่งผิดพลาดและข้อบกพร่องที่เกิดขึ้น และผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงการสหกิจครั้งนี้จะมีประโยชน์ไม่มากนักน้อยสำหรับหน่วยงานที่เกี่ยวข้อง ตลอดจนผู้ที่สนใจที่จะศึกษาการพัฒนาการจินตทัศน์ข้อมูลต่อไป

ชัยวัฒน์ โจอทตะคุ

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญรูป	ฉ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 แผนการดำเนินงาน	2
บทที่ 2 ข้อมูลสถานประกอบการและเอกสารที่เกี่ยวข้อง	3
2.1 รายละเอียดเกี่ยวกับสถานประกอบการ	4
2.2 ภาษาเอชทีเอ็มแอล (Hypertext Markup Language : HTML)	5
2.3 ภาษาซีเอสเอส (Cascading Style Sheets : CSS)	6
2.4 ภาษาเอสคิวแอล (Structured Query Language : SQL)	6
2.5 โปรแกรมมายเอสคิวแอล เวิร์คเบ็นช์ (MySQL Workbench)	8
2.6 ภาษาจาวา สคริปต์ (JavaScript)	9
2.7 ภาษาไทป์ สคริปต์ (TypeScript)	11
2.8 นักซ์ เจเอส เฟรมเวิร์ค (Nuxt JS Framework)	12
2.9 วีว เจเอส (Vue JS)	13
2.10 โหนด เจเอส (Node JS)	14
2.11 โปรแกรมมิชวล สตูดิโอ โค้ด (Visual Studio Code)	15
2.12 โปรแกรมกิตฮับ เดสก์ท็อป (GitHub Desktop)	15
2.13 ทฤษฎีการใช้งานตัวเครื่องมือฟิกม่า (Figma)	16

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินโครงการการปฏิบัติงาน	18
3.1 ออกแบบโครงสร้างข้อมูลในฐานข้อมูล	18
3.2 ออกแบบ UX/UI ในส่วนหน้าบ้าน ผ่าน Figma	25
3.3 สร้างโปรเจกต์ด้วยเฟรมเวิร์คของ Nuxt JS	26
3.4 เขียน API สำหรับส่งข้อมูลระหว่างหน้าบ้าน หลังบ้าน และตัวเซิร์ฟเวอร์	27
3.5 สร้างส่วนของเว็บไซต์หน้าบ้านตามทีออกแบบ	33
บทที่ 4 ผลการดำเนินโครงการการปฏิบัติงาน	36
4.1 ผลการออกแบบโครงสร้าง	36
4.2 ผลการออกแบบที่นำการออกแบบมารวมกัน	37
4.3 ผลการทดสอบการทำงานของระบบงานในเบราว์เซอร์ต่าง ๆ	38
4.4 ผลการเปรียบเทียบโครงงานที่พัฒนา	39
บทที่ 5 สรุปการดำเนินงานโครงการการปฏิบัติงาน	41
5.1 สรุปผลการดำเนินงาน	41
5.2 ปัญหาและอุปสรรค	41
5.3 แนวทางในการพัฒนาและข้อเสนอแนะ	41
บรรณานุกรม	42
ภาคผนวก	44
ภาคผนวก ก	45
ประวัติผู้จัดทำโครงงาน	46

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการปฏิบัติงานของนักศึกษา	2
4.1 การทดสอบความสามารถตามขอบเขตที่กำหนด	38

สารบัญรูป

รูปที่	หน้า
2.1	4
2.2	5
2.3	6
2.4	8
2.5	9
2.6	10
2.7	11
2.8	12
2.9	13
2.10	14
2.11	15
2.12	16
2.13	17
3.1	19
3.2	19
3.3	26
3.4	26
3.5	27
3.6	27
3.7	28
3.8	29
3.9	30
3.10	30
3.11	30
3.12	31
3.13	31
3.14.	32
3.15	32

สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.16	การเรียกใช้ข้อมูลจากฐานข้อมูล	33
3.17	หน้าตาเว็บไซต์ที่ออกแบบใน Figma	34
3.18	หน้าตาของเว็บไซต์หน้าบ้าน(1)	34
3.19	หน้าตาของเว็บไซต์หน้าบ้าน(2)	35
4.1	ภาพของตัวเว็บไซต์ที่ออกแบบเสร็จสิ้น	37
4.2	หน้าตาเว็บไซต์ Tool Management(เก่า)	39
4.3	หน้าตาเว็บไซต์ Tool Management(ใหม่)	39

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัทที่มุ่งเน้นการพัฒนาซอฟต์แวร์ที่มีคุณภาพและให้บริการที่ประทับใจสำหรับลูกค้าทั้งในตลาดภายในประเทศและต่างประเทศด้วยความมุ่งมั่นในการนำเทคโนโลยีที่ทันสมัยมาใช้ในการสร้างผลิตภัณฑ์และบริการที่ประทับใจลูกค้า มุ่งหวังที่จะเป็นผู้นำในการสร้างนวัตกรรมทางด้านเทคโนโลยีและซอฟต์แวร์ในอนาคต

คณะผู้จัดทำโครงการได้รับมอบหมายในการสร้างเว็บไซต์ใหม่สำหรับบริษัท ทูล แมเนจเม้นต์ เพื่อเสนอโปรไฟล์และบริการของบริษัทอย่างมีประสิทธิภาพและน่าสนใจต่อลูกค้าทั้งภายในและภายนอก การโปรโมทบริษัทผ่านเว็บไซต์เป็นวิธีที่สำคัญในการสร้างความรู้สึกลงและการเชื่อมั่นในบริษัท ดังนั้น เว็บไซต์ใหม่จะถูกออกแบบให้สื่อถึงค่านิยมและวิสัยทัศน์ของบริษัทอย่างชัดเจน และมุ่งเน้นไปที่ประสบการณ์ผู้ใช้ที่ดีและการให้บริการที่เต็มที่ ความสามารถในการเชื่อมต่อและสร้างความสัมพันธ์ที่ยั่งยืนกับลูกค้าจะเป็นปัจจัยสำคัญในการสร้างเว็บไซต์ใหม่นี้ โดยใช้เทคโนโลยีที่ทันสมัยและการออกแบบที่สวยงามและใช้งานง่าย ส่วนข้าพเจ้าอยู่ฝั่งทีมพัฒนาระบบของเว็บไซต์

1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาข้อมูลเกี่ยวกับการพัฒนาเว็บไซต์ บริษัท ทูล แมเนจเม้นต์
- 1.2.2 เพื่อออกแบบและพัฒนาพัฒนาเว็บไซต์ บริษัท ทูล แมเนจเม้นต์

1.3 ขอบเขต

- 1.3.1 สามารถเข้าสู่ระบบเพื่อจัดการกับข้อมูลหลังบ้านได้
- 1.3.2 สามารถเพิ่ม ลบ และแก้ไขข้อมูลหน้าเว็บไซต์จากระบบหลังบ้านได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 เพื่อใช้ในการนำเสนอสำหรับผู้เข้าชมข้อมูลของบริษัท
- 1.4.2 ได้ทราบหลักการและความรู้จากการทำเว็บไซต์ของบริษัท
- 1.4.3 ได้ทราบความรู้เรื่องการใช้ API และ Framework ในการสร้างเว็บไซต์ และสามารถนำไปปรับประยุกต์ใช้ และบูรณาการให้เข้ากับการศึกษา และการทำงานอื่นๆ ต่อไป

1.5 ขั้นตอนการดำเนินงาน

- 1.5.1 ศึกษาเครื่องมือ
- 1.5.2 รวบรวมข้อมูล
- 1.5.3 ออกแบบโครงสร้าง
- 1.5.4 ออกแบบงาน
- 1.5.5 ส่งมอบงาน

1.6 แผนการดำเนินงาน

การปฏิบัติงานสหกิจศึกษามีแผนการดำเนินงานตั้งแต่ 12 ธันวาคม พ.ศ.2566 ถึง

5 เมษายน พ.ศ.2567 โดยมีรายละเอียดดังนี้

ตารางที่ 1.1 ตารางแผนการปฏิบัติงานของนักศึกษา

งาน	ธ.ค. 2566				ม.ค. 2567				ก.พ. 2567				มี.ค. - เม.ย. 2567				
1.ศึกษาเครื่องมือ	■																
2.รวบรวมข้อมูล					■												
3.ออกแบบโครงสร้าง									■								
4.ออกแบบงาน										■							
5.ส่งมอบงาน													■				

บทที่ 2

ข้อมูลสถานประกอบการและเอกสารที่เกี่ยวข้อง

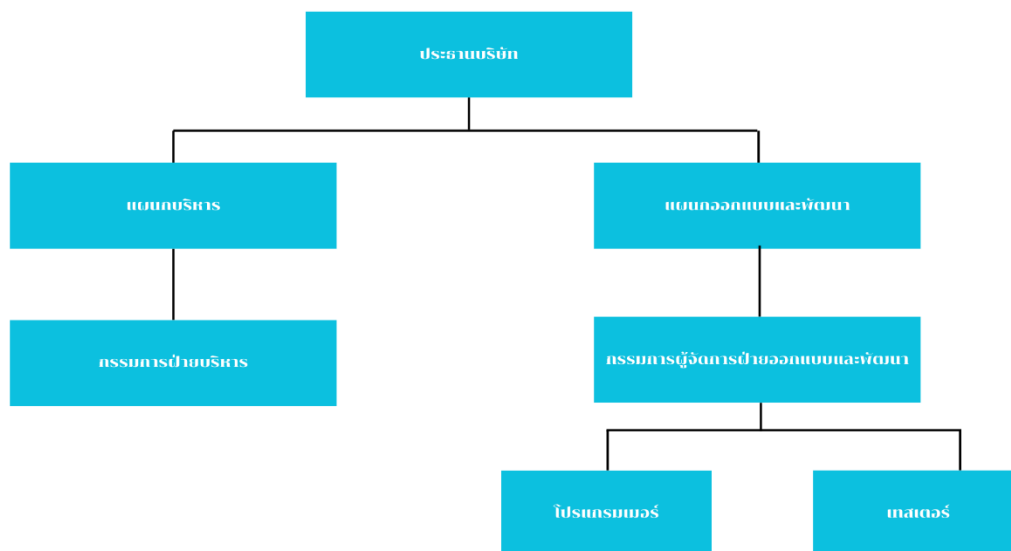
ในบทนี้จะกล่าวถึงหลักการ และทฤษฎีพื้นฐานที่ใช้ในการพัฒนาเว็บไซต์บริษัท ทูลแมนเจมেন্ট เพื่อใช้เป็นแนวทางการพัฒนาโครงการนี้ต่อไป โดยสามารถแบ่งออกมาเป็นหัวข้อต่างๆ ดังต่อไปนี้

- 2.1 รายละเอียดเกี่ยวกับสถานประกอบการ
- 2.2 ภาษาเอชทีเอ็มแอล (Hypertext Markup Language : HTML)
- 2.3 ภาษาซีเอสเอส (Cascading Style Sheets : CSS)
- 2.4 ภาษาเอสคิวแอล (Structured Query Language : SQL)
- 2.5 โปรแกรมมายเอสคิวแอล เวอร์คเบ็นช์ (MySQL Workbench)
- 2.6 ภาษาจาวา สคริปต์ (JavaScript)
- 2.7 ภาษาไทป์ สคริปต์ (TypeScript)
- 2.8 นัคซ์ เจเอส เฟรมเวิร์ค (Nuxt JS Framework)
- 2.9 วิว เจเอส (Vue JS)
- 2.10 โหนด เจเอส (Node JS)
- 2.11 โปรแกรมวิซวล สตูดิโอ โค้ด (Visual Studio Code)
- 2.12 โปรแกรมกิตฮับ เดสก์ท็อป (GitHub Desktop)
- 2.13 ทฤษฎีการใช้งานตัวเครื่องมือฟิกมา (Figma)

2.1 รายละเอียดเกี่ยวกับสถานประกอบการ

บริษัท เรเวล ซอฟต์แวร์ จำกัด ก่อตั้งเมื่อ 8 สิงหาคม 2559 โดย นายธนะ เทพชูลีพรศิลป์ และ นายสันติสุข สิริจินดาธิวัฒน์ ซึ่งมีวัตถุประสงค์ที่จะพัฒนาระบบบริหารจัดการองค์กร (ERP) ให้กับผู้ประกอบการธุรกิจระดับเล็กและระดับกลาง โดยบริษัท เรเวล ซอฟต์แวร์ จำกัด ให้บริการด้านคำปรึกษา, ออกแบบและพัฒนาเว็บไซต์, แอปพลิเคชันบนมือถือ, โปรแกรมระบบปฏิบัติการต่างๆ พัฒนาโปรแกรมให้เหมาะสมกับธุรกิจของลูกค้า ตลอดจนการดูแลบำรุงรักษาระบบ หรือโปรแกรมต่างๆ ที่ช่วยอำนวยความสะดวกในการดำเนินงานและบริการที่ลูกค้าต้องการนำไปใช้ในธุรกิจ โดยได้รับความไว้วางใจจากกลุ่มลูกค้าจากหลายกลุ่มธุรกิจ อาทิ อุตสาหกรรม โรงแรม โรงพยาบาล สถาบันการศึกษา และหน่วยงานภาคเอกชนชั้นนำของประเทศไทย

2.1.1 โครงสร้างองค์กร



รูปที่ 2.1 โครงสร้างองค์กร

2.2 ภาษาเอชทีเอ็มแอล (Hypertext Markup Language : HTML)

คือภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบน website หรือที่เราเรียกกันว่าเว็บเพจ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) และจากการพัฒนาทางด้าน Software ของ Microsoft ทำให้ภาษา HTML เป็นอีกภาษาหนึ่งที่ใช้เขียนโปรแกรมได้ หรือที่เรียกว่า HTML Application HTML เป็นภาษาประเภท Markup สำหรับการสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำได้โดยใช้โปรแกรม Text Editor ต่างๆ เช่น Notepad, Editplus หรือจะอาศัยโปรแกรมที่เป็นเครื่องมือช่วยสร้างเว็บเพจ เช่น Microsoft FrontPage, Dream Weaver ซึ่งอำนวยความสะดวกในการสร้างหน้า HTML ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรม web browser เช่น IE Microsoft Internet Explorer (IE), Mozilla Firefox, Safari, Opera, และ Netscape Navigator เป็นต้น [1]

โปรแกรมที่ 2.2 ตัวอย่างการใช้งานโค้ดภาษาเอชทีเอ็มแอล

```

1 <html>
2 <head>
3 </head>
4 <body>
5   Hello World
6   <h1>Hello World</h1>
7   <p>Hello World</p>
8 </body>
9 </html>

```

จากโปรแกรมที่ 2.2 สามารถอธิบายการทำงานดังนี้

บรรทัดที่ 5 ประกาศข้อความ Hello World

บรรทัดที่ 6 ประกาศข้อความ Hello World ใน tag <h1>

บรรทัดที่ 7 ประกาศข้อความ Hello World ใน tag <p>



รูปที่ 2.2 ผลลัพธ์การรันจากการเขียนโค้ดในโปรแกรมที่ 2.2

จากรูปที่ 2.2 สามารถอธิบายได้ว่าผลการรันจากโปรแกรมที่ 2.2 เป็นการแสดงผล Hello World ที่แตกต่างกันตาม tag ที่ใช้ผ่านทางหน้าจอ

2.3 ภาษาซีเอสเอส (Cascading Style Sheets : CSS)

คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลล์์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดย องค์กร World Wide Web Consortium หรือ W3C [2]

2.4 ภาษาเอสคิวแอล (Structured Query Language : SQL)



รูปที่ 2.3 รูปโลโก้ของตัวภาษา SQL

โปรแกรมระบบจัดการฐานข้อมูลที่พัฒนาโดยบริษัท MySQL AB ซึ่งมีหน้าที่เก็บข้อมูลอย่างเป็นระบบ และรองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูลที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่น ๆ เพื่อสร้างระบบงานที่รองรับความต้องการของผู้ใช้ เช่นการทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side

Script) เช่น PHP, ASP.NET, หรือ JSP รวมถึงการทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น Visual Basic .NET, Java, หรือ C# โดย MySQL สามารถทำงานได้บนระบบปฏิบัติการหลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์สที่ถูกนำไปใช้งานมากที่สุด

MySQL ถือเป็นระบบจัดการฐานข้อมูล (Database Management System - DBMS) แบบ relational ซึ่งมีความสามารถในการเก็บข้อมูลทั้งหมดในรูปแบบของตาราง และสามารถเชื่อมโยงข้อมูลระหว่างตารางได้อย่างยืดหยุ่น โดยใช้ภาษา SQL เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล

MySQL เป็นโปรแกรมที่แจกจ่ายให้ใช้งานแบบ Open Source ซึ่งหมายความว่าผู้ใช้งานทุกคนสามารถใช้งานและปรับแต่งได้ตามต้องการ โดยสามารถดาวน์โหลดโปรแกรม MySQL ได้ฟรีจากอินเทอร์เน็ตและใช้งานโดยไม่มีค่าใช้จ่ายใด ๆ

ข้อดีของ MySQL ได้แก่การเพิ่มประสิทธิภาพให้กับแอปพลิเคชัน เนื่องจาก stored procedure สามารถคอมไพล์และเก็บไว้ในฐานข้อมูลเพื่อเรียกใช้งานได้เร็วขึ้น และลดการส่ง traffic ระหว่างแอปพลิเคชันและเซิร์ฟเวอร์ฐานข้อมูล อีกทั้งยังช่วยในการลดงานของนักพัฒนาโดยการซ่อนการทำงานในรูปแบบของ interface ของฐานข้อมูล และช่วยเพิ่มความปลอดภัยของข้อมูลด้วยการจำกัดสิทธิ์การเข้าถึง stored procedure ของผู้ดูแลระบบ

อย่างไรก็ตาม MySQL ก็มีข้อเสียบางประการ เช่น การพัฒนาและดูแล stored procedure อาจจะซับซ้อนและใช้ความสามารถทางเทคนิคที่สูง รวมถึงยากต่อการ debug และหากมีการใช้งาน stored procedure จำนวนมากมากเกินไปอาจทำให้เกิดการใช้ทรัพยากรของระบบมากขึ้นได้ และโครงสร้างของ stored procedure ก็มักจะซับซ้อนในกรณีที่การทำงานมีความซับซ้อนมาก

ดังนั้นการใช้ MySQL ในโครงการควรพิจารณาความสมเหตุสมผลและความเหมาะสมของการนำมาใช้งานให้ดีที่สุดเพื่อให้ได้ระบบที่มีประสิทธิภาพและปลอดภัยอย่างเหมาะสม โดยคำนึงถึงความต้องการและความเหมาะสมของโครงการในทุกขั้นตอนของการพัฒนาและดูแลระบบ [3]

โปรแกรมที่ 2.3 ตัวอย่างการใช้งานโค้ดภาษาเอสควิแอล

```

45 Task.insertProduct = (connection, data = {}) => new Promise((resolve, reject) => {
46   let sql = `INSERT INTO tb_product (
47     product_id,
48     product_name,
49     product_img,
50     product_type,
51     addby,
52     adddate
53   ) VALUES (
54     ${connection.escape(data.product_id)},
55     ${connection.escape(data.product_name)},
56     ${connection.escape(data.product_img)},
57     ${connection.escape(data.product_type)},
58     ${connection.session.actionby},
59     NOW()
60   )`
61

```


รูปที่ 2.5 ผลลัพธ์จากการเก็บชุดข้อมูลเข้าไปในฐานข้อมูลจากโปรแกรมที่ 2.3

จากรูปที่ 2.5 จะเห็นได้ว่าหากมีการเรียกใช้ข้อมูลจากโปรแกรมที่ 2.3 โปรแกรมจัดการฐานข้อมูลจะทำการเก็บค่าตามคำสั่งในตัวโค้ด

2.6 ภาษาจาวา สคริปต์ (JavaScript)

เป็นภาษาโปรแกรมที่นักพัฒนาใช้ในการสร้างหน้าเว็บแบบอินเทอร์แอคทีฟ ตั้งแต่การอัปเดตข้อมูลล่าสุดของสื่อโซเชียลไปจนถึงการแสดงภาพเคลื่อนไหวและแผนที่แบบอินเทอร์แอคทีฟ ฟังก์ชันของ JavaScript สามารถปรับปรุงประสบการณ์ที่ผู้ใช้จะได้รับจากการใช้งานเว็บไซต์ และในฐานะที่เป็นภาษาในการเขียนสคริปต์ฝั่งไคลเอนต์ จึงเป็นหนึ่งในเทคโนโลยีหลักของ World Wide Web ยกตัวอย่างเช่น เมื่อคุณท่องเว็บแล้วเห็นภาพสไลด์ เมนูหรือป๊อปอัพแบบคลิกให้แสดงผล หรือสื่อบรรยากาศที่เปลี่ยนแบบไดนามิกบนหน้าเว็บ นั่นคือคุณเห็นเอฟเฟกต์ของ JavaScript

แต่ก่อนเว็บเพจเป็นแบบคงที่คล้ายกับหน้าหนังสือ โดยหลักแล้วหน้าเพจคงที่แสดงข้อมูลในเค้าโครงที่ตายตัว และไม่ได้ทำทุกอย่างที่ตอนนี้เราคาดหวังจากเว็บไซต์สมัยใหม่ JavaScript เกิดขึ้นในฐานะเทคโนโลยีฝั่งเบราว์เซอร์เพื่อทำให้เว็บแอปพลิเคชันมีความเป็นไดนามิกมากขึ้น เมื่อใช้ JavaScript เบราวเซอร์จะสามารถตอบสนองต่อการโต้ตอบของผู้ใช้และเปลี่ยนแปลงเค้าโครงเนื้อหาบนเว็บเพจได้

เมื่อภาษาผ่านการพัฒนาอย่างเต็มที่ นักพัฒนา JavaScript ก็สร้างไลบรารี เฟรมเวิร์ก และแนวทางปฏิบัติในการเขียนโปรแกรม แล้วเริ่มนำ JavaScript ไปใช้นอกเว็บเบราว์เซอร์ วันนี้ คุณสามารถใช้ JavaScript สำหรับทั้งการพัฒนาฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ เราให้ตัวอย่างกรณีการใช้งานที่พบบ่อยบางส่วนในหัวข้อย่อยต่อไปนี้:

แต่ก่อนเว็บเพจเป็นแบบคงที่คล้ายกับหน้าหนังสือ โดยหลักแล้วหน้าเพจคงที่แสดงข้อมูลในเค้าโครงที่ตายตัว และไม่ได้ทำทุกอย่างที่ตอนนี้เราคาดหวังจากเว็บไซต์สมัยใหม่ JavaScript เกิดขึ้นในฐานะเทคโนโลยีฝั่งเบราว์เซอร์เพื่อทำให้เว็บแอปพลิเคชันมีความเป็นไดนามิกมากขึ้น เมื่อใช้ JavaScript เบราวเซอร์จะสามารถตอบสนองต่อการโต้ตอบของผู้ใช้และเปลี่ยนแปลงเค้าโครงเนื้อหาบนเว็บเพจได้

เมื่อภาษาผ่านการพัฒนาอย่างเต็มที่ นักพัฒนา JavaScript ก็สร้างไลบรารี เฟรมเวิร์ก และแนวทางปฏิบัติในการเขียนโปรแกรม แล้วเริ่มนำ JavaScript ไปใช้นอกเว็บเบราว์เซอร์ วันนี้ คุณสามารถใช้ JavaScript สำหรับทั้งการพัฒนาฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ เราให้ตัวอย่างกรณีการใช้งานที่พบที่บ่อยบางส่วนในหัวข้อย่อต่อไปนี้:

ภาษาโปรแกรมทั้งหมดทำงานด้วยการแปลไวยากรณ์ที่คล้ายภาษาอังกฤษเป็นโค้ดสำหรับเครื่อง จากนั้นระบบปฏิบัติการจะเรียกใช้โค้ดนั้น JavaScript ได้รับการจัดประเภทอย่างกว้าง ๆ ว่าเป็นภาษาเขียนสคริปต์ หรือภาษาที่แปลผลแล้ว โค้ด JavaScript ได้รับการแปลผล นั่นคือ แปลโดยตรงเป็นโค้ดภาษาสำหรับเครื่อง ด้วยกลไก JavaScript ในขณะที่ในภาษาโปรแกรมอื่น ๆ คอมไพเลอร์จะคอมไพล์โค้ดทั้งหมดเป็นโค้ดสำหรับเครื่องในขั้นตอนที่แยกต่างหาก ดังนั้น ภาษาเขียนสคริปต์ทั้งหมดจึงเป็นภาษาโปรแกรม แต่ไม่ใช่ว่าภาษาโปรแกรมทั้งหมดจะเป็นภาษาเขียนสคริปต์เสมอไป [4] ซึ่งในตัวเว็บไซต์ส่วนมากจะมีการเรียกใช้ฟังก์ชันที่รับค่าต่างๆ มาแสดงผลบนตัวเว็บไซต์ โดยที่สามารถเขียนตัวฟังก์ชันในการเรียกค่าจากตัวฐานข้อมูลได้ตามรูปที่ 2.6

```

10 Task.getProductBy = async (connection, data) => {
11   const { options = [] } = data
12   const products = await ProductModel.getProductBy(connection, data)
13
14   if (options.includes('product_detail_lists')){
15     for (const product of products.docs){
16       product.product_detail_lists = await ProductDetailListModel.getProductDetailListBy(connection, {
17         product_id: product.product_id,
18       }).then(res => res.docs)
19     }
20   }
21   if (options.includes('product_feature')){
22     for (const product of products.docs){
23       product.product_feature = await ProductFeatureModel.getProductFeatureBy(connection, {
24         product_id: product.product_id,
25       }).then(res => res.docs)
26     }
27   }
28   return products
29 }

```

รูปที่ 2.6 ส่วนโปรแกรมที่ถูกเขียนด้วยภาษา JavaScript เพื่อใช้ในการเรียกค่า

บรรทัดที่ 10 เป็นการประกาศชื่อฟังก์ชันโดยกำหนดชื่อเป็น getProduct และให้รับค่าที่ต้องการ (Attribute) เป็นค่า data

บรรทัดที่ 11 กำหนดให้ตัว options เก็บค่าทั้งหมดของ data

บรรทัดที่ 12 กำหนดให้ตัวแปร products เก็บค่าจาก data โดยเรียกใช้ตัวโมเดลที่ใช้ในการเรียกค่าของ product

บรรทัดที่ 14 เป็นส่วนของการเช็คค่า `product_detail_lists` ใน `options` นั้นมีข้อมูลอยู่หรือไม่

บรรทัดที่ 15-18 คือลูปในการเรียกใช้โมเดลเพื่อเก็บค่าของ `product_detail_lists` โดยมีเงื่อนไขว่า `product_id` ต้องตรงกันกับ `product_id` ของ `products`

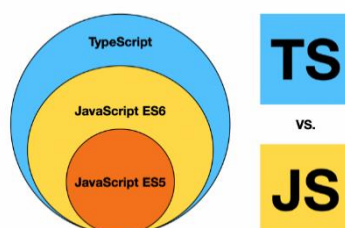
บรรทัดที่ 21 เป็นส่วนของการเช็คค่า `product_feature` ใน `options` นั้นมีข้อมูลอยู่หรือไม่

บรรทัดที่ 22-25 คือลูปในการเรียกใช้โมเดลเพื่อเก็บค่าของ `product_feature` โดยมีเงื่อนไขว่า `product_id` ต้องตรงกันกับ `product_id` ของ `products`

บรรทัดที่ 28 เป็นการส่งค่า `products` กลับไปที่หน้า `page` ที่เรียกใช้งาน

2.7 ภาษาไทป์ สคริปต์ (TypeScript)

โดยพื้นฐานแล้ว JavaScript นั้นเป็นภาษา Script ที่ก่อนหน้านี้เป็น Client-Side Programming เท่านั้น แต่ตอนนี้ (หลังจากที่มี Node.js) มันสามารถใช้สำหรับ Server-Side Programming ได้ด้วย ทำงานได้ทั้งบน Web Browser และ Web Server ซึ่ง JavaScript จะช่วยให้ Code ของเราสามารถโต้ตอบ (Interact) กับ Browser และยังสามารถเปลี่ยนหรือ Update ทั้ง HTML และ CSS แม้ว่าคุณสามารถใช้ JavaScript กับ REST APIs, XML รวมทั้งอื่น ๆ แต่มันก็ไม่ได้ถูกออกแบบมาเพื่อพัฒนา Applications, Applets หรือระบบขนาดใหญ่ที่มีความซับซ้อน แต่ในปัจจุบันมันกำลังจะสามารถทำแบบนั้นได้ เมื่อ JavaScript เริ่มเติบโตมากขึ้น Code ของมันก็เริ่มหนักและซับซ้อนมากขึ้นจนไม่สามารถตอบสนองต่อ Requirement ของ Object-Oriented Programming ได้ มันจึงเป็นต้นเหตุที่ไป “จำกัด” การเติบโตของ JavaScript ซึ่งเป็นภาษา Server-Side และ Multitasking และด้วยเหตุนี้เอง TypeScript เกิดขึ้นเพื่อเอาชนะปัญหาดังกล่าว



รูปที่ 2.7 ความแตกต่างระหว่าง TypeScript กับ JavaScript

TypeScript ก็คือภาษา JavaScript ใน Version ที่ได้รับการ Upgrade นั้นเอง มันสามารถทำงานบน Node.js Environment หรือ Web Browser ต่าง ๆ ที่มีการรองรับ ECMAScript 3 ขึ้นไป TypeScript เป็น Statically Compiled Language ที่ได้จัดเตรียมทั้ง Static Typing, Classes และ Interface ไว้ให้แล้ว ช่วยให้คุณสามารถเขียน Code ของ JavaScript ที่เรียบง่ายและ Clean ได้อย่างสะดวกขึ้น ดังนั้น การใช้ TypeScript จะช่วยให้คุณสามารถสร้าง Software ที่ปรับใช้งานได้ง่าย และมีประสิทธิภาพมากยิ่งขึ้น [5]

2.8 นักซ์ เจเอส เฟรมเวิร์ค (Nuxt JS Framework)



รูปที่ 2.8 ภาพโลโก้ของตัว Nuxt JS Framework

เป็นเฟรมเวิร์ค JavaScript ที่ใช้สำหรับการพัฒนาแอปพลิเคชัน และเว็บแอปพลิเคชัน โดยมีพื้นฐานมาจาก Vue.js ซึ่งเป็นเฟรมเวิร์ค JavaScript ที่ได้รับความนิยมในการพัฒนาเว็บแอปพลิเคชันแบบเอสเอส (SSR) และแอปพลิเคชันเว็บแบบสแป โดย Nuxt.js มาพร้อมกับคุณสมบัติที่มากมายเพื่อช่วยให้การพัฒนาเว็บแอปพลิเคชันเป็นเรื่องที่สะดวกและรวดเร็วขึ้น นอกจากนี้ยังมีความยืดหยุ่นสูงในการตั้งค่าและใช้งาน โดยที่ตัว Nuxt JS เป็นตัวจัดการระบบต่างๆ ไม่ว่าจะเป็นระบบการกำหนดเส้นทางของไหลของข้อมูล (Routing) การดึงข้อมูลจากไฟล์ต่างๆ (Data Fetching) การแคชข้อมูล (Caching) หรือการเรียกใช้งานตัวหน้าเว็บไซต์แบบออนไลน์ (Deployment) และอื่นๆ นักซ์ เจเอส เฟรมเวิร์ค เลยเป็นระบบที่นิยมใช้กันในปัจจุบันนี้อย่างมาก โดยสามารถอธิบายโครงสร้าง ที่ตัวนักซ์ เจเอส จัดการได้

2.9 วิว เจเอส (Vue JS)



รูปที่ 2.9 ภาพโลโก้ของตัว Vue JS

เป็นเฟรมเวิร์ก JavaScript แบบไดนามิกที่มีชื่อเสียงในการสร้างส่วนต่อประสานผู้ใช้ มันเป็นที่รู้จักดีในชุดทักษะการเรียนรู้ที่สูง แม้จะเป็นเฟรมเวิร์กที่เข้าถึงง่ายและตรงไปตรงมา แต่ก็มีคุณสมบัติที่ยืดหยุ่นและเหมาะสำหรับการใช้งานในการพัฒนาเว็บแอป ด้วยความเข้าใจพื้นฐานเกี่ยวกับ HTML, CSS และ JavaScript เพียงเท่านั้น เราสามารถเริ่มต้นสร้างเว็บแอปใน Vue.js ได้ทันที เส้นโค้งในการเรียนรู้ย่อมเป็นที่ตั้งแต่เริ่มต้น และมีความยืดหยุ่นที่สูงสุดในการพัฒนาเว็บแอป

ในชุดเฟรมเวิร์ก JavaScript ทุกชุดมีคุณสมบัติและความสามารถเฉพาะตัว แต่ Vue.js ได้รับความนิยมนสูงสุดในช่วง 5 ปีที่ผ่านมาบน Github.com แม้ว่ามันจะมีอุตสาหกรรมที่น้อยกว่า React JS แต่ Vue.js ยังคงได้รับการจัดอันดับที่สูงและมีการพูดถึงอย่างดี

ไลบรารีหลักของ Vue ใช้คอมโพเนนต์แบบโมเดลมุมมองเพื่อสร้างการประสานงานแบบไดนามิกระหว่างโมเดลและคอมโพเนนต์มุมมอง ทำให้การสร้างโปรแกรมเว็บที่ใช้แนวคิดแอปหน้าเดียวกันเป็นเรื่องง่าย รากฐานของไลบรารี Vue คือ HTML, CSS และ JavaScript ด้วยการเชื่อมโยงข้อมูลแบบสองทาง ทำให้นักออกแบบสามารถสร้างเว็บแอปที่สามารถทำงานบนคอมพิวเตอร์ของผู้ใช้และส่งมอบส่วนประกอบที่สมจริงโดยไม่ต้องอัปเดตหน้าจอใหม่

Vue.js มีเอกสารที่เป็นที่ยอมรับมากที่สุดในวงการ มันเสนอกระบวนการที่ง่ายและขั้นตอน โดยเฉพาะ ซึ่งเหมาะสำหรับผู้เริ่มต้น แม้ว่าความเข้าใจใน Vue.js จะง่ายกว่า React.js และ Angular แต่ Vue.js ยังมีคุณสมบัติที่โดดเด่น เช่น แอตทริบิวต์ HTML ที่ช่วยให้โปรแกรมเมอร์สามารถขยาย HTML ได้ตามต้องการในขั้นตอนการพัฒนา.

2.10 โหนด เจเอส (Node JS)



รูปที่ 2.10 ภาพโลโก้ของตัว Node JS

คือ สภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ที่ทำงานด้วย V8 engine นั้นหมายความว่าเราสามารถใช้งาน Node.js ในการพัฒนาแอปพลิเคชันแบบ Command line แอปพลิเคชัน Desktop หรือแม้แต่เว็บเซิร์ฟเวอร์ได้ โดยที่ Node.js จะมี APIs ที่เราสามารถใช้งานได้สำหรับการทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ค เป็นต้น

นอกจากนี้ถูกพัฒนาและทำงานด้วยใช้ Chrome V8 engine สำหรับคอมไพล์ภาษาจาวาสคริปต์ (JavaScript) ให้เป็นภาษาเครื่องด้วยการคอมไพล์แบบทันทีทันใด (Just-in-time : JIT) เพื่อเพิ่มประสิทธิภาพการทำงานของภาษาจาวาสคริปต์ (JavaScript) จากที่แต่เดิมมันเป็นภาษาที่มีการทำงานแบบ Interpreted อีกทั้งเป็นโปรแกรมที่สามารถใช้ได้ทั้งบน Windows, Linux และ Mac OS X ซึ่งหมายความว่าคุณสามารถเขียนโปรแกรมในภาษาจาวาสคริปต์ (JavaScript) และนำไปรันได้ทุกระบบปฏิบัติการที่สนับสนุนโดยโหนด เจเอส (Node.js) นี้เป็นแนวคิดของการเขียนครั้งเดียวแต่ทำงานได้ทุกที่ (Write once, run anywhere)

2.11 โปรแกรมวิซวล สตูดิโอ โค้ด (Visual Studio Code)



รูปที่ 2.11 ภาพไอคอนของตัวโปรแกรม Visual Studio Code

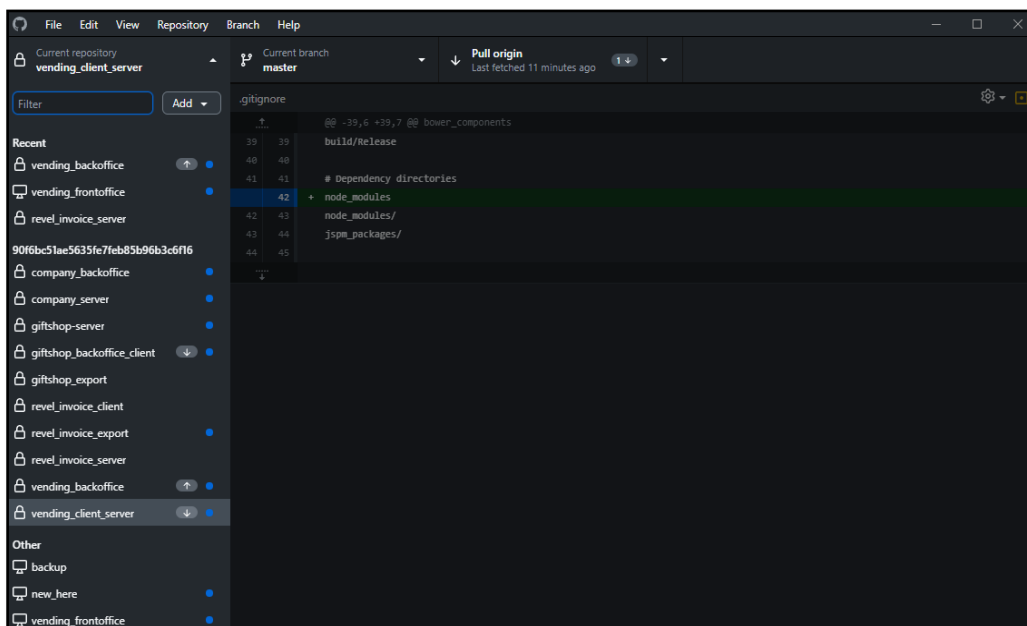
เป็นโปรแกรมแก้ไขซอร์สโค้ดที่พัฒนาโดยไมโครซอฟท์สำหรับ Windows, Linux และ macOS มีการสนับสนุนสำหรับการดีบั๊ก การควบคุม Git ในตัวและ GitHub การเน้นไวยากรณ์ การเติมโค้ดอัจฉริยะ ตัวอย่าง และ code refactoring มันสามารถปรับแต่งได้หลายอย่าง ให้ผู้ใช้สามารถเปลี่ยนธีม แป้นพิมพ์ลัด การตั้งค่า และติดตั้งส่วนขยายที่เพิ่มฟังก์ชันการทำงานเพิ่มเติม ซอร์สโค้ดนั้นฟรีและโอเพนซอร์สและเผยแพร่ภายใต้สิทธิ์การใช้งานของสถาบัน (Massachusetts Institute of Technology : MIT) โบนารีที่คอมไพล์แล้วเป็นโปรแกรมฟรีและสามารถใช้ส่วนตัวหรือเพื่อการค้าได้

วิซวล สตูดิโอ โค้ดใช้โอเล็กตรอนเป็นเฟรมเวิร์กที่ใช้ในการปรับใช้แอปพลิเคชัน Node.js สำหรับเดสก์ท็อปที่รันบนเอ็นจิน Blink แม้ว่าจะใช้เฟรมเวิร์กอเล็กตรอน ซอฟต์แวร์นี้ไม่ได้ใช้อะตอม และใช้คอมโพเนนต์ตัวแก้ไขเดียวกัน (ชื่อรหัส "Monaco") กับที่ใช้ใน Azure DevOps (เดิมชื่อ Visual Studio Online และ Visual Studio Team Services) [6]

2.12 โปรแกรมกิตฮับ เดสก์ท็อป (GitHub Desktop)

คือแอปพลิเคชันโอเพนซอร์สที่ใช้ Electron ซึ่งเขียนด้วย TypeScript และใช้ React GitHub ทำหน้าที่เป็นแหล่งเก็บข้อมูลสำหรับซอร์สโค้ดและสามารถติดตั้งและใช้กับระบบปฏิบัติการประเภทต่างๆได้ Brendan Forster พัฒนา GitHub Desktop เพื่อให้ผู้ใช้ Gnu / Linux สามารถทำได้ ทำงานกับ GitHub จากเดสก์ท็อป

GitHub Desktop พยายามลดความยุ่งยากและ ทำให้เวิร์กโฟลว์ Git และ GitHub สามารถเข้าถึงได้มากขึ้น เป้าหมายคือการทำให้เวิร์กโฟลว์ทั่วไปเป็นเรื่องง่ายดังนั้นนักพัฒนาที่ระดับเริ่มต้นและผู้มีประสบการณ์จึงมีประสิทธิผลเมื่อทำงานกับ Git และ GitHub ดังรูปที่ 2.10



รูปที่ 2.12 ภาพหน้าต่างของโปรแกรม GitHub Desktop

GitHub Desktop ไม่ได้แทนที่ฟังก์ชันการทำงานของ Git แต่เป็นเพียงเครื่องมือที่ช่วยให้ผู้ใช้และทีมของเขาทำงานได้มากขึ้น แม้ว่าโปรแกรมนี้จะสามารถใช้ได้กับคนหลากหลาย แต่ส่วนใหญ่เป็นนักพัฒนาซอฟต์แวร์ที่พบว่าโปรแกรมนี้มีประโยชน์มากที่สุด โดยจุดประสงค์หลักของแอปพลิเคชันนี้คือการขยายคุณสมบัติของ GitHub ไม่ใช่เพื่อจำลองชุดคุณสมบัติของตัวเว็บไซต์หลักของ GitHub GitHub Desktop เป็นเครื่องมือที่มีประโยชน์สำหรับผู้เริ่มต้นพัฒนา แต่ไม่ใช่เครื่องมือการสอนอย่างชัดเจน โดยหลักแล้วผู้ใช้งานต้องการความช่วยเหลือในการทำงานให้เสร็จอย่างรวดเร็วและมีประสิทธิภาพมากขึ้นด้วยวิธีที่สอดคล้องกับแนวทางปฏิบัติที่ดีที่สุด [7]

2.13 ทฤษฎีการใช้งานตัวเครื่องมือฟิกมา (Figma)

เป็นเครื่องมือออกแบบอินเทอร์เฟซแบบทำงานร่วมกัน (the collaborative interface design tool) ซึ่งความสามารถในการทำงานร่วมกัน (collaborative) กลายเป็นจุดเด่นที่ทำให้ Figma ได้รับความนิยมในกลุ่มของนักออกแบบ (User Experience/User Interface :UX/UI) เพราะในโลกของการทำงานจริงเหล่านักออกแบบจำเป็นจะต้องทำงานร่วมกับนักออกแบบคนอื่นๆ ไปจนถึงผู้ที่เกี่ยวข้อง ตัวอย่างเช่น Productor Manager, Developer, Marketing และอื่นๆ เป็นต้น

นักออกแบบจึงไม่ได้ใช้ Figma สำหรับแค่การจัดวางโครงสร้างหน้าต่าง (Layout Interface) แต่ยังใช้สำหรับการสร้างแบบจำลอง (Prototype) และพิธีเซ้นท์งานเพื่อให้เกิดความเข้าใจที่ตรงกัน และยังสามารถแบ่งปันความคิดของตัวเองได้ ในส่วน Design System นั้นสามารถหยิบยืมงานกันได้ และทำให้งานออกแบบในภาพรวมมีความสอดคล้องกันโดยที่เราได้ใช้ตัวฟิกมาในการออกแบบใส่ส่วนของหน้าตาเว็บไซต์ Tool Management ดังรูปที่ 2.13



รูปที่ 2.13 ภาพตัวอย่างของหน้าต่างการออกแบบเว็บไซต์

บทที่ 3

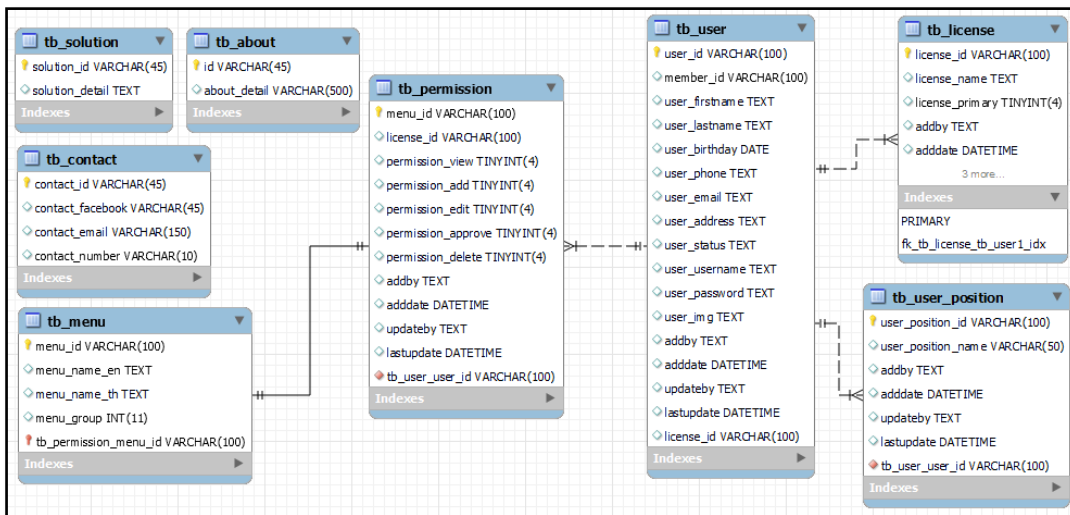
วิธีการดำเนินงานโครงการการปฏิบัติงาน

ในการจัดทำโครงการขึ้นนี้หลังจากผู้จัดการศึกษารายละเอียด รวบรวมเนื้อหาข้อมูลต่างๆ และทฤษฎีที่เกี่ยวข้องแล้ว ได้ดำเนินการออกแบบ และพัฒนาตัวระบบเว็บไซต์ซึ่งมีขั้นตอนในการดำเนินงานดังต่อไปนี้

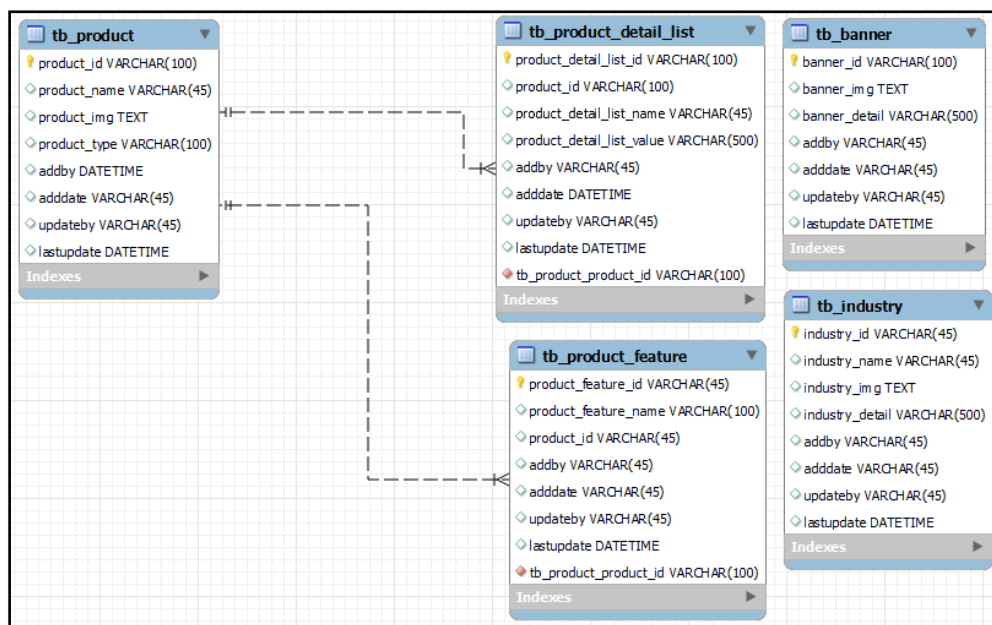
- 3.1 ออกแบบโครงสร้างข้อมูลในฐานข้อมูล
- 3.2 ออกแบบ UX/UI ในส่วนหน้าบ้าน ผ่าน Figma
- 3.3 สร้างโปรเจ็คด้วยเฟรมเวิร์คของ Nuxt JS
- 3.4 เขียน API สำหรับส่งข้อมูลระหว่างหน้าบ้าน หลังบ้าน และตัวเซิร์ฟเวอร์
- 3.5 สร้างส่วนของเว็บไซต์หน้าบ้านตามทีออกแบบ

3.1 ออกแบบโครงสร้างข้อมูลในฐานข้อมูล

ขั้นตอนการดำเนินงานในการออกแบบระบบจัดเก็บข้อมูลของฐานข้อมูลนั้นสำคัญมาก ซึ่งเราจำเป็นที่จะให้ระบบนั้นเก็บข้อมูลได้สะดวกและทำงานได้อย่างมีประสิทธิภาพ จะเริ่มจากการกำหนดชื่อตัวแปรต่างๆ เพื่อใช้เป็นคีย์เวิร์ด (Keyword) ในการค้นหาและจัดการกับข้อมูลต่างๆ โดยสามารถใช้ภาษา SQL เพื่อสร้างตารางและกำหนดประเภทของข้อมูลที่ต้องการดังรูปที่ 3.1 และรูปที่ 3.2



รูปที่ 3.1 ตารางความสัมพันธ์ของข้อมูลที่ใช้



รูปที่ 3.2 ตารางความสัมพันธ์ของข้อมูลที่ใช้(2)

โดยจากรูปที่ 3.1 และรูปที่ 3.2 มีการเก็บข้อมูลอยู่ทั้งหมด 13 ตาราง โดยมีรายละเอียดการเก็บข้อมูลของแต่ละตารางดังนี้

1. tb_license

- Columns:
 - license_id: VARCHAR, Primary Key – เก็บรหัส ID ของตาราง license

- license_name: TEXT – เก็บชื่อของผู้ที่มีสิทธิ์ในการเข้าถึง
- license_primary: TINYINT – เก็บค่าความสำคัญของสิทธิ์ในการใช้งาน
- addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
- adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
- updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
- lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

2. tb_user_position

- Columns:
 - user_position_id: VARCHAR, Primary Key – เก็บรหัส ID ของตำแหน่งผู้ใช้งานระบบ
 - user_position_name VARCHAR – เก็บชื่อของตำแหน่งผู้ใช้งานระบบ
 - addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
 - lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

3. tb_user

- Columns:
 - user_id: VARCHAR, Primary Key - เก็บรหัส ID ของผู้ใช้งานระบบ
 - member_id: VARCHAR, Primary Key - เก็บรหัส ID ของสมาชิก
 - user_firstname: TEXT - เก็บข้อมูลชื่อจริงของผู้ใช้งานระบบ
 - user_lastname: TEXT – เก็บข้อมูลนามสกุลของผู้ใช้งานระบบ
 - user_birthday: DATE - เก็บข้อมูลวันเกิดของผู้ใช้งานระบบ
 - user_phone: TEXT - เก็บข้อมูลเบอร์โทรศัพท์ของผู้ใช้งานระบบ

- user_email: TEXT - เก็บข้อมูลอีเมลล์ของผู้ใช้งานระบบ
- user_address: TEXT - เก็บข้อมูลที่อยู่ของผู้ใช้งานระบบ
- user_status: TEXT - เก็บข้อมูลสถานะของผู้ใช้งานระบบ
- user_username: TEXT - เก็บข้อมูลชื่อผู้เข้าใช้ของผู้ใช้งานระบบ
- user_password: TEXT - เก็บข้อมูลรหัสผ่านของผู้ใช้งานระบบ
- user_img: TEXT - เก็บข้อมูลรูปภาพของผู้ใช้งานระบบในรูปแบบข้อความ
- addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
- adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
- updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
- lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล
- license_id: VARCHAR - เก็บค่ารหัส ID ของสิทธิ์การเข้าถึง

4. tb_permission

- Columns:
 - menu_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของเมนูที่สามารถเข้าถึงได้
 - license_id: VARCHAR - เก็บค่า ID ของสิทธิ์การเข้าถึง
 - permission_view: TINYINT - เก็บค่าสิทธิ์ในการมองเห็นข้อมูล
 - permission_add: TINYINT - เก็บค่าสิทธิ์ในการเพิ่มข้อมูล
 - permission_edit: TINYINT - เก็บค่าสิทธิ์ในการแก้ไขข้อมูล
 - permission_approve: TINYINT - เก็บค่าสิทธิ์ในการยืนยันข้อมูล
 - permission_delete: TINYINT - เก็บค่าสิทธิ์ในการลบข้อมูล.
 - addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล

- updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
- lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

5. tb_solution

- Columns:
 - solution_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของตาราง solution
 - solution_detail: TEXT - เก็บข้อมูลรายละเอียดของตาราง solution

6. tb_about

- Columns:
 - Id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของตาราง about
 - about_detail: TEXT - เก็บข้อมูลรายละเอียดของตาราง about

7. tb_contact

- Columns:
 - contact_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของตาราง contact
 - contact_facebook: VARCHAR - เก็บข้อมูลชื่อ facebook
 - contact_email: VARCHAR - เก็บข้อมูลชื่อ email
 - contact_number: VARCHAR - เก็บข้อมูลเบอร์โทรศัพท์

8. tb_menu

- Columns:
 - menu_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของเมนู
 - menu_name_en: TEXT - เก็บค่าชื่อภาษาอังกฤษของเมนู
 - menu_name_th: TEXT - เก็บค่าชื่อภาษาไทยของเมนู
 - menu_group: INT - เก็บค่าตัวเลขประจำกลุ่ม

9. tb_banner

- Columns:

- banner_id: VARCHAR, Primary Key – เก็บค่ารหัส ID ของตาราง banner
- banner_img: TEXT – เก็บข้อมูลรูปภาพของ banner ในรูปแบบข้อความ
- banner_detail: VARCHAR – เก็บข้อมูลรายละเอียดของรูปภาพในตาราง banner
- addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
- adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
- updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
- lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล
-

10. tb_industry

- Columns:
 - industry_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของ industry
 - industry_name: VARCHAR – เก็บค่าข้อมูลชื่อของ industry
 - industry_img: TEXT – เก็บข้อมูลรูปภาพของ industry ในรูปแบบข้อความ
 - industry_detail: VARCHAR – เก็บข้อมูลรายละเอียดของรูปภาพในตาราง industry
 - addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
 - lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

11. tb_product

- Columns:
 - product_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของ product
 - product_name: VARCHAR - เก็บค่าชื่อของ product
 - product_img: TEXT - เก็บข้อมูลรูปภาพของ product ในรูปแบบข้อความ
 - product_type: VARCHAR - เก็บข้อมูลประเภทของรูปภาพในตาราง product
 - addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
 - lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

12. tb_product_detail_list

- Columns:
 - product_detail_list_id: VARCHAR, Primary Key - เก็บค่ารหัส ID ของ product_detail_list
 - product_id: VARCHAR - เก็บค่ารหัส ID ของ product
 - product_detail_list_name: VARCHAR - เก็บค่าชื่อของ product_detail_list
 - product_detail_list_value: VARCHAR - เก็บค่าข้อมูลหัวข้อย่อยที่เป็นรายละเอียดของ product_detail_list
 - addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล

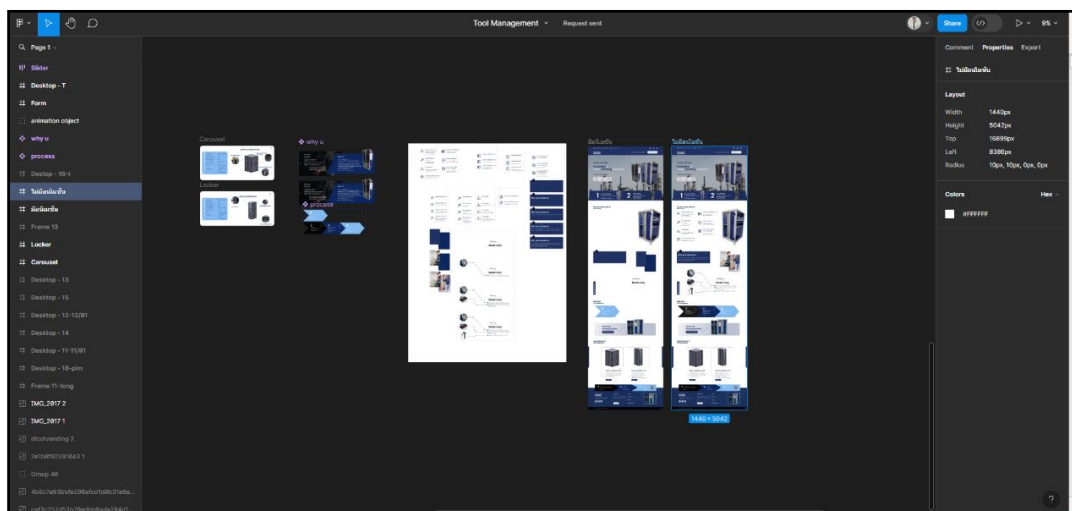
- lastupdate: DATETIME - เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

13. tb_product_feature

- Columns;
 - product_feature_id: VARCHAR, Primary Key – เก็บค่ารหัส ID ของ product_feature
 - product_feature_name: VARCHAR – เก็บค่าข้อมูลชื่อของ product_feature
 - product_id: VARCHAR - เก็บค่ารหัส ID ของ product
 - addby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - adddate: DATETIME - เก็บค่าวันและเวลาของผู้ใช้งานระบบที่เพิ่มข้อมูล
 - updateby: VARCHAR - เก็บค่าชื่อของผู้ใช้งานระบบที่แก้ไขข้อมูล
 - lastupdate: DATETIME – เก็บค่าวันและเวลาผู้ใช้งานระบบที่แก้ไขข้อมูล

3.2 ออกแบบ UX/UI ในส่วนหน้าบ้าน ผ่าน Figma

การออกแบบหน้าเว็บไซต์ให้ผู้ใช้งานทั่วไปสามารถใช้งานได้อย่างสะดวกสบาย และง่ายต่อการเข้าใจ โดยการออกแบบ UX/UI ผ่านเว็บไซต์ฟิกมา (Figma) ให้ความสำคัญในเรื่องของการออกแบบให้มีความเหมาะสมและดูทันสมัย รวมไปถึงทำให้การนำ UX/UI ที่ออกแบบไปแปลงเป็นตัวเว็บไซต์นั้นก็จะมีความง่ายที่เพิ่มขึ้นด้วย ไม่ว่าจะเป็นการสร้างตาม UX/UI ที่ออกแบบไว้แล้ว หรือการนำไปใช้กับเครื่องมือตัวอื่นๆ ในการแปลงไฟล์จาก Figma ไปเป็นโค้ดที่ถูกออกแบบและจัดองค์ประกอบตามที่เรากำหนดไว้แล้ว ก็สามารถทำได้ และตัวเว็บไซต์ Figma นั้นใช้งานในรูปแบบ browser-based ที่ทุกคนสามารถทำงานพร้อมกันได้ทุกที่ ทุกเวลา และยังมี Features อีกมากมายให้เลือกใช้ที่สามารถช่วยให้การส่งต่องานระหว่างทีมทำได้ง่ายขึ้นดังรูปที่ 3.3



รูปที่ 3.3 การออกแบบหน้าตาเว็บไซต์ทูล แมเนจเมนต์

จากรูปที่ 3.2 เป็นการออกแบบหน้าตาเว็บไซต์ทูล แมเนจเมนต์ ที่เป็นส่วนที่มีความสำคัญของเว็บไซต์บริษัท โดยการออกแบบให้ดูเหมาะสมกับผู้ใช้งานทั่วไป ซึ่งสามารถดูรายละเอียดการออกแบบต่างๆ ได้อย่างชัดเจนผ่าน Figma

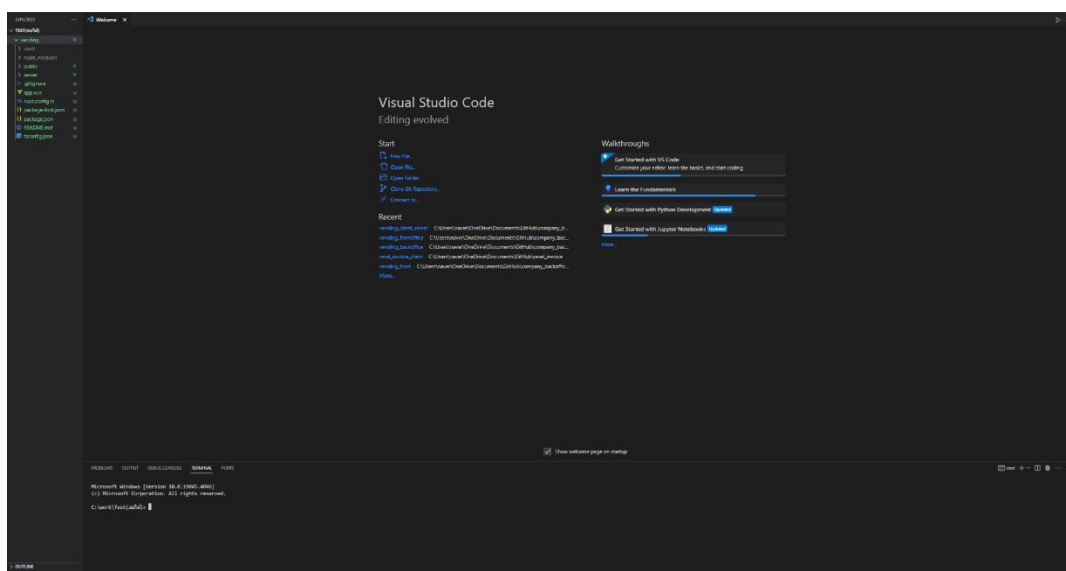
3.3 สร้างโปรเจ็คด้วยเฟรมเวิร์คของ Nuxt JS

การพัฒนาตัวเว็บไซต์ให้มีประสิทธิภาพและมีการจัดการที่สะดวกสบายต่อการใช้งานก็เป็นส่วนหนึ่งที่จำเป็น เพราะฉะนั้นการเลือกภาษาและเฟรมเวิร์ค (framework) ให้เหมาะสมกับประเภทงานที่ทำก็มีผลมากเช่นกัน โดยที่เราต้องทำการติดตั้งเราจำเป็นที่จะต้องติดตั้งตัว Node JS ก่อน จากนั้นให้เข้าไปที่ command prompt แล้วสร้างโพลเดอร์โปรเจ็คขึ้นมาด้วยการพิมพ์คำสั่งใน command prompt ตามรูปที่ 3.4

```
npx nuxt@latest init <project-name>
```

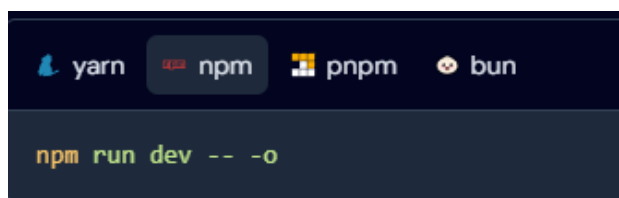
รูปที่ 3.4 ภาพตัวอย่างคำสั่งสำหรับสร้างโปรเจ็ค Nuxt JS

จากรูปที่ 3.4 เป็นการเรียกใช้คำสั่งบน command prompt สำหรับการสร้างโปรเจ็ค Nuxt JS ขึ้นมาใหม่ จากนั้นให้เปิดตัวโพลเดอร์ที่ทำการติดตั้งแล้วขึ้นมาบนโปรแกรม Visual Studio Code จากนั้นเปิดหน้าต่างเทอร์มินัลสำหรับป้อนคำสั่งต่อไปดังรูปที่ 3.5



รูปที่ 3.5 หน้าตาหลังจากติดตั้ง Nuxt JS เรียบร้อย

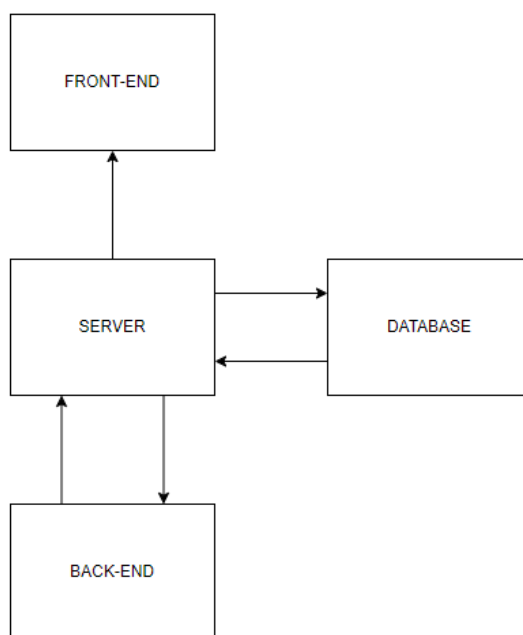
ต่อไปให้ทำการป้อนคำสั่ง `npm install` ลงไปในช่องเทอร์มินัล (Terminal) เพื่อทำการติดตั้ง `node_module` ที่ใช้งานสำหรับโปรเจ็คนี้เป็นอันเรียบร้อย ขั้นตอนสุดท้ายคือการเรียกใช้คำสั่งเพื่อรันตัวเฟรมเวิร์คให้ทำงานบน browser แบบ real-time ด้วยคำสั่ง `npm run dev` ดังรูปที่ 3.6



รูปที่ 3.6 คำสั่งในการรันตัวเว็บไซต์ให้แสดงผลบน browser

3.4 เขียน API สำหรับส่งข้อมูลระหว่างหน้าบ้าน หลังบ้าน และตัวเซิร์ฟเวอร์

ในส่วนนี้เราจำเป็นที่จะต้องสร้างโพลเดอร์ในการจัดการแต่ละส่วนอย่างเป็นระเบียบ โดยสามารถเชื่อมโยงหากันได้อย่างมีประสิทธิภาพที่สุด ซึ่งจะกำหนดให้เป็นตามรูปที่ 3.7



รูปที่ 3.7 การไหลของข้อมูลเว็บไซต์ทั่วไป

ขั้นตอนต่อไปนี้จะเป็นการสร้างโพลเดอร์ทั้ง 3 โพลเดอร์ซึ่งมี frond-end, server และ back-end โดยแต่ละส่วนจะแบ่งการทำงานได้ดังนี้

- 3.4.1 Front-end คือ ส่วนที่ทำหน้าที่เป็นหน้าตาของเว็บไซต์ที่ใช้งานจริง
- 3.4.2 Server คือ ส่วนที่เป็นส่วนกลางในการจัดการกับการไหลของข้อมูลทั้งหมด โดยที่ตัว server จะทำหน้าที่ทั้งรับและส่งผ่านข้อมูลจากฐานข้อมูลเพื่อไปอัปเดตส่วนต่างๆ หรือจะรับข้อมูลจากส่วนต่างๆ เพื่อไปอัปเดตตัวฐานข้อมูลได้เช่นกัน เพราะฉะนั้นตัว server นั้นเป็นส่วนที่สำคัญมากๆ ในการพัฒนาเว็บไซต์
- 3.4.3 Back-end คือ ส่วนที่รับหน้าที่การจัดการข้อมูลที่เราต้องการ รวมไปถึงสิทธิ์ในการเข้าถึงและการเข้าใช้งานด้วย ส่วนมากจะเป็นส่วนที่ใช้จัดการกับข้อมูลผู้ใช้งานเว็บไซต์ จนไปถึงการเพิ่ม ลด และแก้ไขข้อมูลทั้งหมดของเว็บไซต์

หลังจากที่สร้างทุกส่วนครบแล้ว ต่อไปคือการเขียนโค้ดลงไป โดยเริ่มต้นจากตัว server ก่อน เพราะว่าเป็นส่วนหัวใจหลักในการสื่อสารและจัดการข้อมูลทั้งหมด เริ่มต้นดังนี้

- 1) สร้างโพลเดอร์สำหรับการทำงานหลักก็คือ Model, Controller และ Routing โดยแต่ละโพลเดอร์จะทำหน้าที่ที่ช่วยเหลือกันและกัน อธิบายได้คร่าวๆ ดังนี้

- Model: ทำหน้าที่เรียกใช้ เพิ่ม ลบ และแก้ไขข้อมูลจากฐานข้อมูลของเว็บไซต์โดยมีการใช้ภาษา SQL ในการสั่งการในคำสั่งต่างๆ
- Controller: ทำหน้าที่ส่งออกคำสั่งที่อยู่ในส่วนของ model ไปยัง back-end และ front-end โดยมี route ในการชี้ทางให้
- Route: ทำหน้าที่เป็นเหมือนต่อลำเลียงการไหลของข้อมูลจาก server ไปยัง front-end และ back-end โดยจะเป็นตัวที่ใช้เชื่อมกับส่วนอื่นๆ ที่จำเป็นจะต้องใช้งานในส่วน server นี้ ดังรูปที่ 3.8

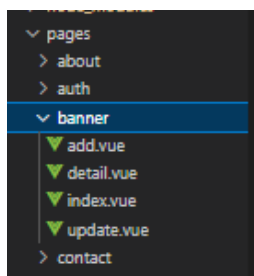
```

1
2 module.exports = (app) => {
3   app.use(require("../utils/req-handler/connect-provider"))
4   app.use(require("../utils/req-handler/res-format"))
5
6   app.get("/", (req, res) => {
7     res.json({ message: `Welcome to ${process.env.APP_NAMESPACE}` });
8   });
9   app.use("/about", require("./about.route"))
10  app.use("/auth", require("./auth.route"))
11  app.use("/license", require("./license.route"))
12  app.use("/user", require("./user.route"))
13  app.use("/user-position", require("./user-position.route"))
14  app.use("/banner", require("./banner.route"))
15  app.use("/product", require("./product.route"))
16  app.use("/product-detail-list", require("./product-detail-list.route"))
17  app.use("/solution", require("./solution.route"))
18  app.use("/industry", require("./industry.route"))
19  app.use("/contact", require("./contact.route"))
20  app.use("/product-feature", require("./product-feature.route"))
21
22 }

```

รูปที่ 3.8 การสร้าง route เพื่อเชื่อมต่อกับส่วนหลังบ้าน




- 2) หลังจากสร้างส่วนของการส่งคำสั่งและส่วนของการเชื่อมต่อกับหลังบ้านแล้ว ต่อไปก็จะเป็นส่วนของหลังบ้านที่จำเป็นจะต้องออกแบบระบบการจัดการข้อมูลที่ต้องการจะนำไปใช้กับส่วนองเว็บไซต์หน้าบ้าน โดยที่มีตัวอย่างในส่วนของ page หน้าต่างตัวอย่างของ banner ดังรูปที่ 3.9



รูปที่ 3.9 ส่วนของ page ที่เก็บไฟล์หน้าเว็บไซต์

แบนเนอร์

เพิ่มรายการใหม่

...	#	รหัสแบนเนอร์	รูปภาพแบนเนอร์	รายละเอียดแบนเนอร์
⋮	1	UP20240214001		banner1
⋮	2	UP20240217001		why_you_should_us_1
⋮	3	UP20240217002		why_you_should_us_2


รูปที่ 3.10 หน้าตา index.vue


แบนเนอร์ / เพิ่มรายการใหม่

เพิ่มรายการใหม่

รหัสแบนเนอร์ *

UP20240228001

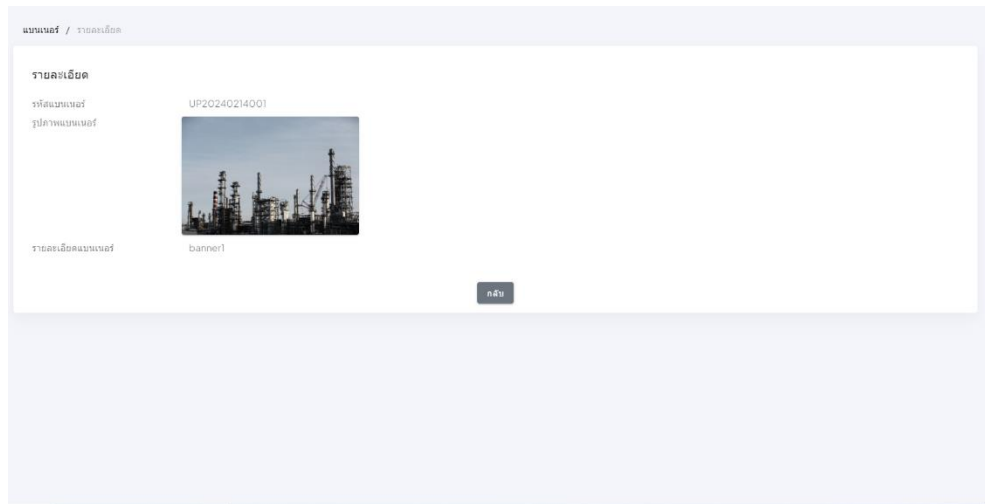


 รูป (จำกัด)

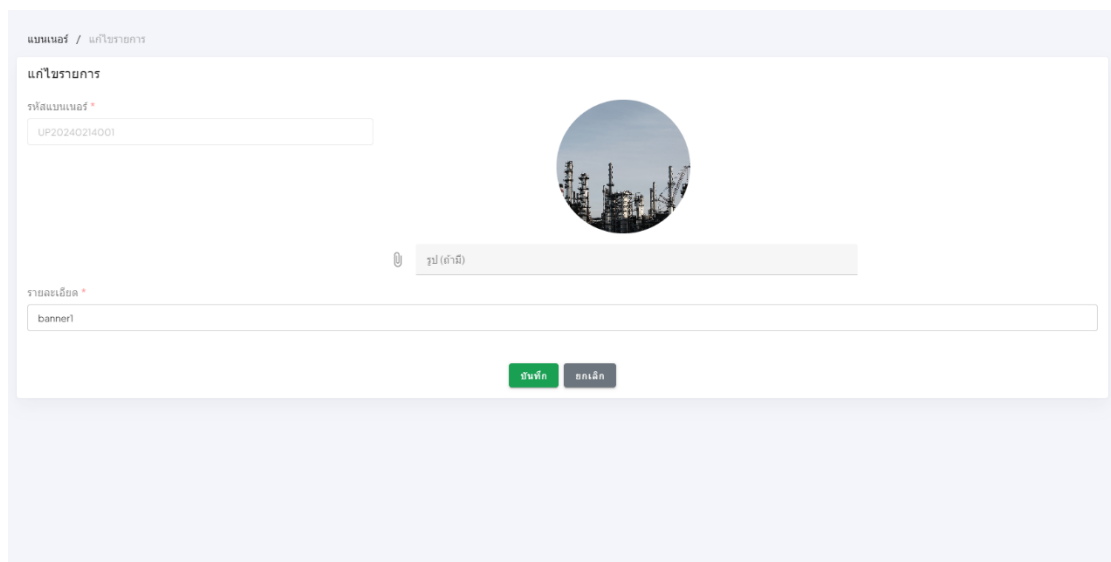
รายละเอียด *

บันทึก ยกเลิก

รูปที่ 3.11 หน้าตา add.vue



รูปที่ 3.12 หน้าตา detail.vue



รูปที่ 3.13 หน้าตา update.vue

จากรูปที่ 3.10 -3.13 เป็นหน้าตาของส่วน page ของ banner ที่สร้างขึ้นมาใช้งานสำหรับ ส่วนของหลังบ้าน ส่วนอื่นๆ เนื่องจาก Vue นั้นจัดการไฟล์ต่างๆ อย่างเป็นระเบียบ ทำให้การเขียน หน้าต่างและการเรียกใช้องค์ประกอบนั้นง่ายและมีประสิทธิภาพมาก ในส่วนต่างๆ ของหลังบ้านจะมี ส่วนที่ใช้ในการติดต่อกับตัว server นั่นก็คือส่วนของ typescript ที่จะรับและส่งคำสั่งจากฝั่ง server ในการเก็บและเรียกใช้ข้อมูลทั้งหมด


```
import type { Banner } from "~/misc/types"
const prefix = 'banner'

const generateBannerID = () => Promise<string> => secureFetch(`${useRuntimeConfig().public.
apiBaseUrl}/${prefix}/generateBannerID`, { method: "POST", })

const getBannerBy = (data?: any): Promise<Banner[]> => secureFetch(
`${useRuntimeConfig().public.apiBaseUrl}/${prefix}/getBannerBy`, {
method: "POST",
body: JSON.stringify(data),
})

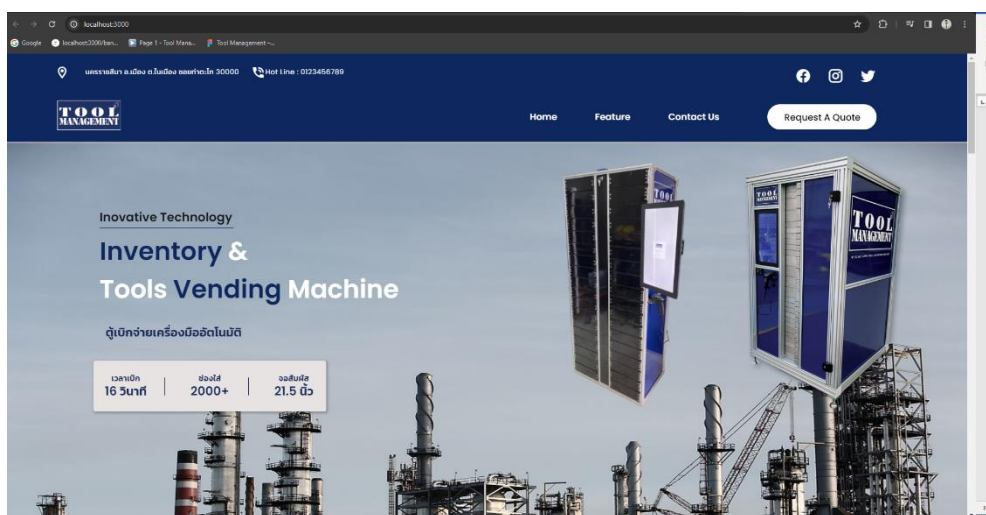
const getBannerByID = (data: { banner_id: string }): Promise<Banner> => secureFetch(
`${useRuntimeConfig().public.apiBaseUrl}/${prefix}/getBannerByID`, {
method: "POST",
body: JSON.stringify(data),
})

const insertBanner = async (data: { banner: Banner, banner_img?: File[] }): Promise<Banner> => {
const formData = new FormData();
formData.append("banner", JSON.stringify(data.banner));
if (data.banner_img?.length) formData.append("banner_img", data.banner_img[0]);
return await secureFetch(
`${useRuntimeConfig().public.apiBaseUrl}/${prefix}/insertBanner`, {
method: "POST",
body: formData,
})
}
```

รูปที่ 3.14 ส่วนของ composables ที่ใช้ typescript

จากรูปที่ 3.14 เป็นส่วนที่ใช้ติดต่อกับตัวของเซิร์ฟเวอร์เพื่อลำเลียงข้อมูลทุกประเภทและคำสั่งต่างๆ

- 3) ต่อมาเป็นส่วนของหน้าบ้านซึ่งมีหลักการสร้างขึ้นมาเดียวกันกับฝั่งหลังบ้าน เปลี่ยนแค่เพียงดีไซน์ของหน้าเว็บไซต์ตามแบบที่ทำในเว็บ Figma และมีแค่การรับค่ามาจากส่วนฐานข้อมูล ซึ่งส่วนของเว็บไซต์หน้าบ้านจะมีหน้าตาแบบรูปที่ 3.15



รูปที่ 3.15 หน้าตาหน้าบ้านของเว็บไซต์ Tool Management

```

98 <hr>
99
100 <!-- Banner Content -->
101
102 <div class="banner" style="height: 85%; position: relative;">
103 
106 <!-- Left banner -->
107 <v-col cols="6">
108 <v-container style=" position: relative; top: 15%; padding-left: 20%">
109 <v-row class="d-flex flex-column">
110 <v-col class="text-h3 text-capitalize">innovative technology</v-col>
111 <div class="ms-3 mt-1 mb-2" style="width: 265px; height: 1px; background:
112 <span style="color: #0e275f"></span></div>
113
114 <v-col class="text-capitalize" style="font-weight: 600; font-size: 50px;">
115 <span style="color: #0e275f">inventory </span>
116 <span style="color: #fff">&</span></span>
117 <div>
118 <span style="color: #fff">tools </span>
119 <span style="color: #0e275f">vending </span>
120 <span style="color: #fff">machine</span></div>
121 </v-col>
122 </v-row>
123 </v-container>

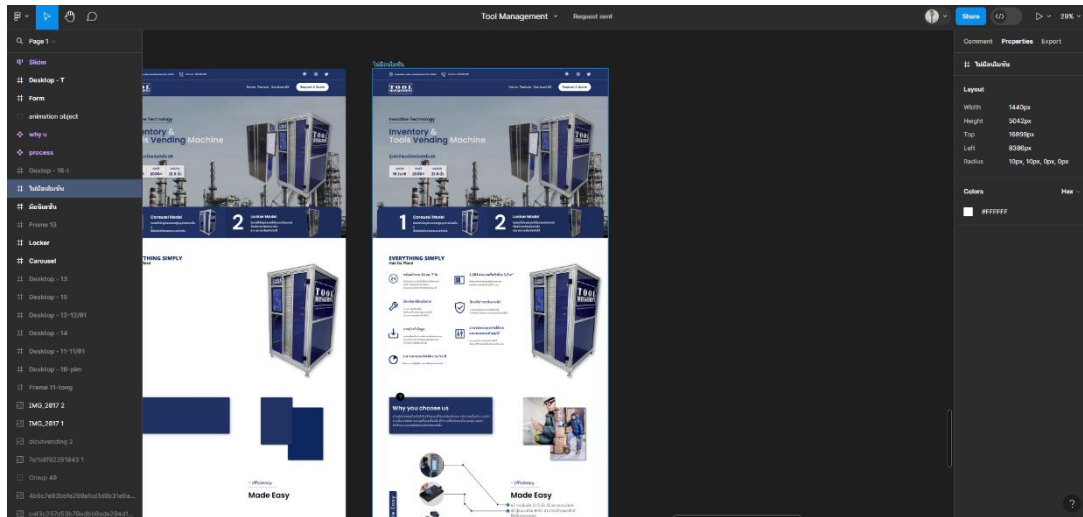
```

รูปที่ 3.16 การเรียกใช้ข้อมูลจากฐานข้อมูล

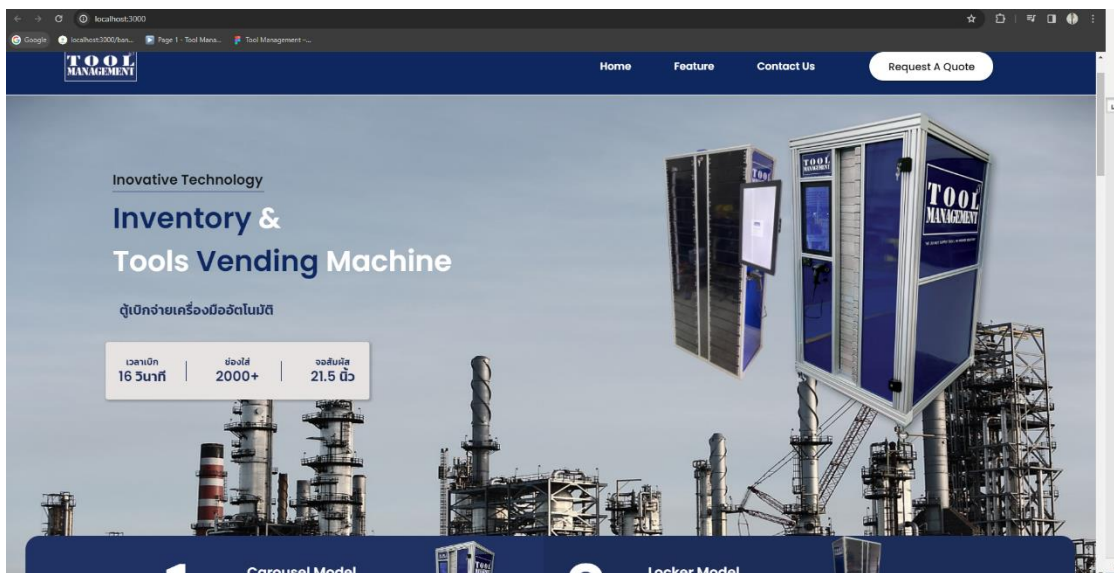
จากรูปที่ 3.16 เป็นใช้คำสั่งดึงข้อมูลโดยใช้ api เรียกไฟล์จากโฟลเดอร์ public ในส่วนของ server โดยใช้ชื่อที่เก็บอยู่ใน banner_img เป็นตัวตรวจสอบกับไฟล์ภาพ หากตรงกันจะนำไฟล์ภาพนั้นมาแสดงขึ้นบนเว็บไซต์ได้

3.5 สร้างส่วนของเว็บไซต์หน้าบ้านตามทีออกแบบ

เมื่อสร้างส่วนของการจัดการข้อมูลทั้งหมดเสร็จแล้ว ในส่วนที่เหลือส่วนสุดท้ายนั้นก็คือหน้าตาของเว็บไซต์ที่ต้องการให้ผู้งานทั่วไปเข้ามาในหน้าเว็บไซต์และได้ประสบการณ์ใช้งานที่ดี ดังนั้นการทำในส่วนหน้าบ้านให้ใช้งานง่ายนั้นก็สำคัญเช่นกัน เราจึงได้ทำการออกแบบส่วนนั้นเว็บ Figma เรียบร้อยแล้ว ดังรูปที่ 3.17



รูปที่ 3.17 หน้าตาเว็บไซต์ที่ออกแบบใน Figma



รูปที่ 3.18 หน้าตาของเว็บไซต์หน้าบ้าน(1)



รูปที่ 3.19 หน้าตาของเว็บไซต์หน้าบ้าน(2)

จากรูปที่ 3.18 และรูปที่ 3.19 เป็นส่วนหน้าตาเว็บไซต์ที่จัด layout ให้ตรงกับดีไซน์ใน Figma มากที่สุดตามต้องการ เพื่อให้การพัฒนาเว็บไซต์มีประสิทธิภาพและมีการใช้งานที่ง่าย

บทที่ 4

ผลการดำเนินโครงการการปฏิบัติงาน

การออกแบบเว็บไซต์บริษัท เรเวลซอฟต์แวร์ จำกัด จะทำการออกแบบในรูปแบบที่ทันสมัย ดูสบายตา และเข้าถึงง่าย ซึ่งมีขั้นตอนและผลการดำเนินงาน ดังต่อไปนี้

- 4.1 ผลการออกแบบโครงสร้าง
- 4.2 ผลการออกแบบที่นำการออกแบบมารวมกัน
- 4.3 ผลการทดสอบการทำงานของระบบงานในเบราว์เซอร์ต่าง ๆ
- 4.4 ผลการเปรียบเทียบโครงการที่พัฒนา

4.1 ผลการออกแบบโครงสร้าง

การออกแบบโครงสร้างในขั้นแรกจะออกแบบที่มีโครงสร้าง 4 ส่วน คือ 1.ส่วน Header 2. ส่วน Banner 3.ส่วน Info 4.ส่วน Footer ดังนี้

4.1.1 ส่วนของ Header

เป็นส่วนที่เป็นเมนูที่มีข้อมูลติดต่อ และสามารถเลือกเมนูต่างๆได้ โดยสามารถกดเข้าไปดูยังส่วนที่ต้องการได้ทันที

4.1.2 ส่วนของ Banner

เป็นส่วนที่แสดงสิ่งแรกๆที่ผู้เข้าชมสามารถมองเห็นและเป็นจุดเด่นของเว็บไซต์ที่บ่งบอกจุดเด่นของเว็บไซต์อย่างชัดเจน

4.1.3 ส่วนของ Info

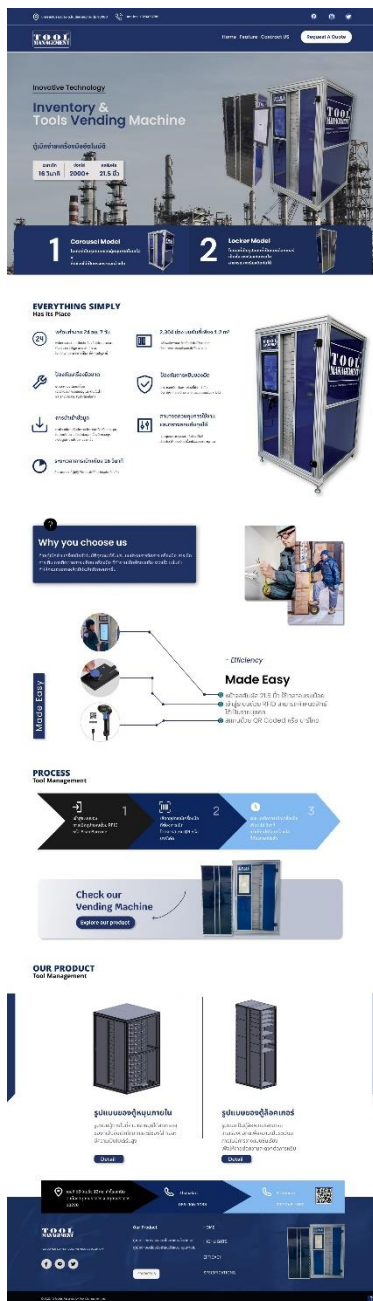
เป็นส่วนที่แสดงและบอกเกี่ยวกับข้อมูลของเว็บไซต์หรือบริการของเว็บไซต์ และผลิตภัณฑ์ของบริษัทว่ามีผลิตภัณฑ์รุ่นอะไรบ้างที่สามารถให้ผู้สนใจรับรู้ได้

4.1.4 ส่วนของ Footer

ส่วนปิดท้ายเว็บไซต์ที่มีช่องทางติดต่อและรายละเอียดสรุปสุดท้าย

4.2 ผลการออกแบบที่นำการออกแบบมารวมกัน

ทางบริษัทได้ทำการเลือกการออกแบบที่ทันสมัย ดูสบายตา และเข้าถึงง่าย จึงต้องการนำผลงานของนักศึกษาที่มาสหกิจมารวมกันให้เป็นเว็บไซต์ ซึ่งการออกแบบสุดท้ายจะนำผลงานที่ทำมานำมารวมกันและปรับปรุงให้ทุกอย่างสามารถเข้ากันได้ตามภาพที่ 4.1



รูปที่ 4.1 ภาพของตัวเว็บไซต์ที่ออกแบบเสร็จสิ้น

4.3 ผลการทดสอบการทำงานของระบบงานในเบราว์เซอร์ต่าง ๆ

จากการทดสอบผ่านการใช้งานบนเว็บเบราว์เซอร์ต่าง ๆ ระบบนี้ทำงานผ่านระบบอินเทอร์เน็ตสามารถทดสอบโดยใช้เครื่องคอมพิวเตอร์ที่มีระบบปฏิบัติการ ตั้งแต่ Windows 10 ขึ้นไป โปรแกรมเบราว์เซอร์ที่ทำการทดสอบ มีรายละเอียดดังนี้

- มอซิลลา ไฟร์ฟอกซ์93 (Mozilla Firefoxversion93)
- กูเกิลโครม7.6.381.20.90.1 (Google Chromeversionversion7.6.381.20.90.1)
- ไมโครซอฟท์ เอจ 10 (Microsoft Edgeversion10)
- โอเปรา (Opera)

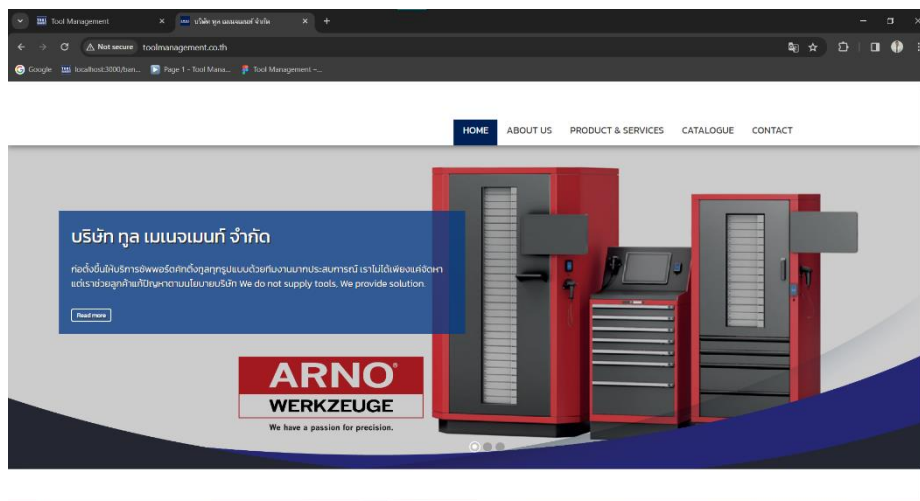
สรุปผลการทดสอบการทำงานของระบบงานในเบราว์เซอร์ต่าง ๆ ได้ดังตารางที่ 4.1

ตารางที่ 4.1 การทดสอบความสามารถตามขอบเขตที่กำหนด

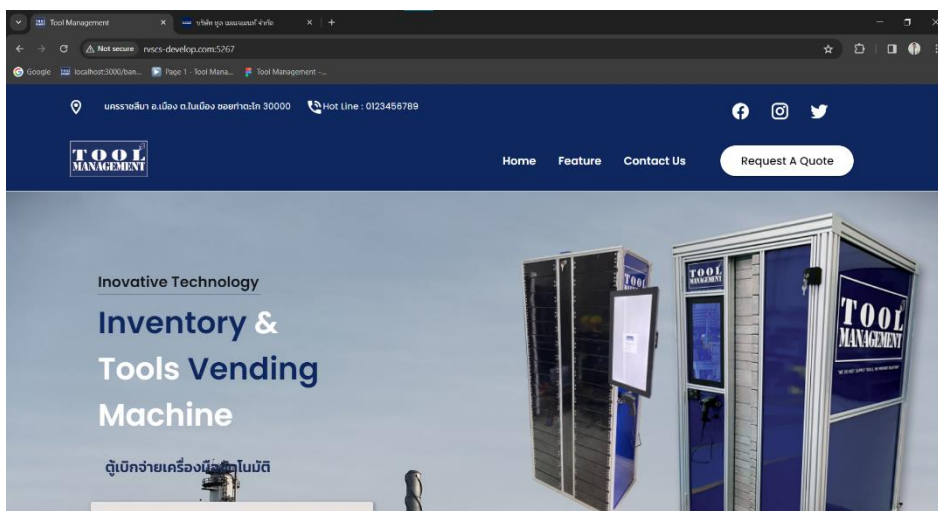
เบราว์เซอร์	ผลการทดสอบ
มอซิลลา ไฟร์ฟอกซ์	สามารถทำงานได้เป็นอย่างดี ไม่พบปัญหา
กูเกิลโครม	สามารถทำงานได้เป็นอย่างดี ไม่พบปัญหา
ไมโครซอฟท์ เอจ	สามารถทำงานได้เป็นอย่างดี ไม่พบปัญหา
โอเปรา	สามารถทำงานได้เป็นอย่างดี ไม่พบปัญหา

จากตารางที่ 4.1 สรุปผลการทดสอบผ่านการใช้งานบนเว็บเบราว์เซอร์ต่าง ๆ ได้ว่าสามารถทำงานได้ดีในทุกเบราว์เซอร์ที่ทำการทดสอบแต่แนะนำให้ใช้งานเว็บแอปพลิเคชันผ่านเบราว์เซอร์กูเกิลโครมเป็นหลัก เนื่องจากการพัฒนาเว็บแอปพลิเคชันทำการพัฒนาบน เบราว์เซอร์ทุกเบราว์เซอร์ ไม่มีปัญหาในการทำงาน

4.4 ผลการเปรียบเทียบโครงการที่พัฒนา



รูปที่ 4.2 หน้าตาเว็บไซต์ Tool Management(เก่า)



รูปที่ 4.3 หน้าตาเว็บไซต์ Tool Management(ใหม่)

เนื่องจากการพัฒนาตัวโครงการนั้น มีการพัฒนามาจากเว็บไซต์ตัวเก่าของตัวบริษัททูลแมเนจเม้นท์ซึ่งพัฒนาต่อยอดด้วยภาษา Vue JS และใช้ Nuxt JS Framework แทนจากเว็บไซต์เก่าที่พัฒนาด้วยภาษา React เนื่องจากทางบริษัทมีความสนใจในการเปลี่ยนการออกแบบหน้าเว็บไซต์เพื่อให้มีความทันสมัย และมีประสิทธิภาพในการทำงานบนเบราว์เซอร์มากขึ้นไปอีก โดยความแตกต่างจากเว็บไซต์เดิมตามรูปที่ กับเว็บไซต์ที่ออกแบบใหม่ตามรูปที่ 4.2 และรูปที่ 4.3 มีดังนี้

- 4.4.1 ทำให้การดูแลรักษา และพัฒนาเว็บไซต์ง่ายขึ้นเนื่องจากโครงสร้างภาษามีความซับซ้อนน้อยกว่า
- 4.4.2 มีส่วน Back-office ในการจัดการข้อมูลของเว็บไซต์
- 4.4.3 ทำให้เว็บไซต์ของบริษัทมีประสิทธิภาพและความเสถียรภาพมากขึ้นกับเบราว์เซอร์
- 4.4.4 ในส่วนของหน้าต่างที่แสดงบนเว็บไซต์มีความสวยงามมากขึ้น

บทที่ 5

สรุปผลการดำเนินงานโครงการการปฏิบัติงาน

จากการออกแบบเว็บไซต์ให้กับบริษัทที่เข้าสหกิจ บริษัท เรเวลซอฟต์แวร์ จำกัด ทางบริษัทได้ประเมินและทำการสรุปผลการทำงานดังต่อไปนี้

5.1 สรุปผลการดำเนินงาน

การออกแบบโครงสร้างเว็บไซต์ได้ถูกวางแผนอย่างเหมาะสมและมีการแบ่งส่วนการทำงานอย่างชัดเจน เช่น Header, Banner, Info และ Footer เพื่อให้ผู้ใช้ทั่วไปสามารถเข้าถึงข้อมูลและบริการได้อย่างสะดวกสบาย การนำเสนอผลงานของบริษัทที่มีความคล่องตัวและปรับได้ตามความต้องการของลูกค้า โดยใช้รูปแบบที่ทันสมัย ดูสบายตา เข้าถึงง่าย และมีความสวยงาม

การทดสอบความสามารถของเว็บไซต์ในเบราว์เซอร์ต่าง ๆ ได้ผลลัพธ์ที่ดี และมีการแนะนำให้ใช้งานเว็บไซต์ผ่าน Google Chrome เป็นหลักเนื่องจากมีประสิทธิภาพและความเสถียรสูงที่สุด

5.2 ปัญหาและอุปสรรค

อุปสรรคในการทำระบบนี้คือเป็นเรื่องของความรู้ การศึกษาวิธีการทำเว็บไซต์ต่างๆ ประสบการณ์ ที่ผู้จัดทำมีน้อยเกินไป และได้รับงานจากโปรเจกต์อื่นๆ ในบริษัทมากเกินไป และมีคนในกลุ่มจบการฝึกงานไปก่อน ในส่วนงานนั้นก็ต้องนำมาทำต่อคนเดียว จึงทำให้เวลาในการทำโครงงานนั้นมีไม่พอ แนวทางการแก้ไขปัญหาคือ ต้องจัดเวลาการทำงานให้ดีกว่านี้ และปฏิเสธงานที่ได้รับมากเกินไป

5.3 ข้อเสนอแนะ

5.3.1 ตัวเว็บไซต์ที่ดีไซน์บางส่วนไม่ได้คำนึงถึงการพัฒนาเว็บสำหรับมือใหม่ จึงใช้เวลามากเกินไปในการเรียนรู้

5.3.2 ควรพัฒนาตัวเว็บไซต์ให้รองรับกับอุปกรณ์อื่นโดยใช้ Responsive

บรรณานุกรม

- [1] mindphp (2565 , พฤศจิกายน. 7) HTML คืออะไร [ออนไลน์], แหล่งที่มา :
<https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2026-html-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html>
- [2] mindphp (2560 , มีนาคม . 14) CSS คืออะไร [ออนไลน์], แหล่งที่มา :
<https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2193-css-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html>
- [3] สีสื่อไอทีเสาะห์ (2563). ระบบจัดการ GYM , แหล่งที่มา :
https://www.commsci.psu.ac.th/uploads/spd_files/1022_ICTMP08_2020.pdf
- [4] Amazon Web Services. (2560 , กุมภาพันธ์. 15). JavaScript คืออะไร [ออนไลน์], แหล่งที่มา : <https://aws.amazon.com/th/what-is/javascript/>
- [5] nopparat.jap (2565 , กรกฎาคม. 15). TypeScript คืออะไร [ออนไลน์], แหล่งที่มา :
<https://il.mahidol.ac.th/th/i-learning-clinic/computer-articles/%E0%B8%97%E0%B8%B3%E0%B9%84%E0%B8%A1%E0%B8%96%E0%B8%B6%E0%B8%87%E0%B8%95%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B9%83%E0%B8%8A%E0%B9%89-typescript-%E0%B8%97%E0%B8%B1%E0%B9%89%E0%B8%87%E0%B8%97%E0%B8%B5/#:~:text=Typescript%20%E0%B8%81%E0%B9%87%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%20JavaScript>

[.E0%B8%84%E0%B8%B8%E0%B8%93%E0%B8%AA%E0%B8%B2%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%96%E0%B9%80%E0%B8%82%E0%B8%B5%E0%B8%A2%E0%B8%99%20Code%20%E0%B8%82%E0%B8%AD%E0%B8%87](#)

- [6] Wikipedia (2560 , กรกฎาคม 17). วิชาลสตูดิโอโค้ด [ออนไลน์], แหล่งที่มา :
<https://th.wikipedia.org/wiki/E0%B8%A7%E0%B8%B4%E0%B8%8A%E0%B8%A7%E0%B8%A5%E0%B8%AA%E0%B8%95%E0%B8%B9%E0%B8%94%E0%B8%B4%E0%B9%82%E0%B8%AD%E0%B9%82%E0%B8%84%E0%B9%89%E0%B8%94>
- [7] Damien A (2564 , พฤษภาคม. 20). GitHub Desktop ทำงานร่วมกับ GitHub [ออนไลน์], แหล่งที่มา :
<https://ubunlog.com/th/E0%B9%80%E0%B8%94%E0%B8%AA%E0%B8%81%E0%B9%8C%E0%B8%97%E0%B9%87%E0%B8%AD%E0%B8%9B-github-E0%B8%97%E0%B8%B3%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A3%E0%B9%88%E0%B8%A7%E0%B8%A1%E0%B8%81%E0%B8%B1%E0%B8%9A-github-E0%B8%88%E0%B8%B2%E0%B8%81%E0%B9%80%E0%B8%94%E0%B8%AA%E0%B8%81%E0%B9%8C%E0%B8%97%E0%B9%87%E0%B8%AD%E0%B8%9B/>
- [8] True Digital Academy (2566 , มกราคม. 26). รู้จัก 'Figma' Tool ออกแบบ Web/App ที่ดีไซน์เนอร์ทั่วโลกเลือกใช้ [ออนไลน์], แหล่งที่มา :
<https://www.truedigitalacademy.com/blog/get-to-know-figma>

ภาคผนวก

ภาคผนวก ก.

ประวัติผู้จัดทำ

ประวัติผู้จัดทำ

ชื่อ – นามสกุล	นายชัยวัฒน์ โจอิตะคุ
ภูมิลำเนา	76/1 หมู่ 14 ต.ตุม อำเภอบึงระจิง จังหวัดนครราชสีมา 30150
เกิดวันที่	1 มีนาคม พ.ศ.2545
สถานที่เกิด	โรงพยาบาลมหาราชนครราชสีมา
Email.	6340505211@nrru.ac.th