



รายงานการปฏิบัติงานสหกิจศึกษา

เรื่อง

รายงานสหกิจศึกษาตำแหน่ง Fullstack Deverloper ที่ G-ABLE

ณ บริษัท จีเอเบิล จำกัด (มหาชน)
127/30 อาคารปัญจานี ชั้น 25
ถนนนนทรี แขวงช่องนนทรี เขตยานนาวา กรุงเทพมหานคร 10120

โดย

นายสิปปกร คำมีสว่าง รหัส 650610813

รายงานนี้เป็นส่วนหนึ่งของกระบวนวิชา 261495 สหกิจศึกษา
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2568



Cooperative Education Report

**Cooperative Education Report for the Position of Fullstack Developer at
G-ABLE**

**127/30 Panjathani Tower, 25th Floor,
Nonsi Road, Chong Nonsi Subdistrict, Yan Nawa District, Bangkok 10120**

Sippakon Khammisawang 650610813

**This report is part of the course 261495 Cooperative Education
Department of Computer Engineering
Faculty of Engineering, Chiang Mai University
Academic Year 2025**

หัวข้อรายงาน : รายงานสหกิจศึกษาตำแหน่ง Fullstack Developer ที่ G-ABLE
: Cooperative Education Report for the Position of Fullstack Developer at
G-ABLE

โดย : นายสิปกร คำมีสว่าง รหัส 650610813

ภาควิชา : วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษา : อ.ดร. ชินวัตร อิศราดิสัยกุล

ปริญญา : วิศวกรรมศาสตรบัณฑิต

สาขา : วิศวกรรมคอมพิวเตอร์

ปีการศึกษา : 2568

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์.ดร.ธงชัย ฟองสมุทร)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ผศ.ดร. นวดนัย คุณเดิศกิจ)

คณะกรรมการสอบรายงาน

..... อาจารย์ที่ปรึกษาสหกิจศึกษา
(รศ.ดร.ปฏิเวช วุฒิสารวัฒนา)

..... พนักงานที่ปรึกษา
(เชีย แสงจันทร์)
(System Analyst)

หัวข้อรายงาน : รายงานสหกิจศึกษาตำแหน่ง Fullstack Developer ที่ G-ABLE
: Cooperative Education Report for the Position of Fullstack Developer at
G-ABLE

โดย : นายสิปกร คำมีสว่าง รหัส 650610813

ภาควิชา : วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษา : อ.ดร. ชินวัตร อิศราดิสัยกุล

ปริญญา : วิศวกรรมศาสตรบัณฑิต

สาขา : วิศวกรรมคอมพิวเตอร์

ปีการศึกษา : 2568

บทคัดย่อ

รายงานฉบับนี้เป็นการนำเสนอผลการปฏิบัติงานสหกิจศึกษาของนักศึกษาวิศวกรรมศาสตรบัณฑิต สาขา วิศวกรรมคอมพิวเตอร์ ในตำแหน่ง Fullstack Developer ที่บริษัท จีเอเบิล จำกัด (G-ABLE Company Limited) ในระหว่างระยะเวลาการปฏิบัติงาน นักศึกษาได้มีส่วนร่วมในกระบวนการพัฒนาซอฟต์แวร์ทั้งใน ส่วนของ Frontend และ Backend โดยมีหน้าที่หลักในการพัฒนาและดูแลระบบเว็บแอปพลิเคชัน ออกแบบและพัฒนา API สำหรับการสื่อสารข้อมูลระหว่างระบบ รวมถึงการเชื่อมต่อและจัดการฐานข้อมูลเพื่อ ให้การทำงานของระบบเป็นไปอย่างมีประสิทธิภาพ ซึ่งประสบการณ์ทั้งหมดนี้มีส่วนสำคัญในการเพิ่มประสิทธิภาพและความยืดหยุ่นของกระบวนการพัฒนาซอฟต์แวร์ เพื่อให้บริษัทสามารถส่งมอบโซลูชันดิจิทัลที่มีคุณภาพและตอบสนองความต้องการของลูกค้าได้อย่างรวดเร็วและมีประสิทธิผล

Report Title : Cooperative Education Report for the Position of Fullstack Developer at
G-ABLE

Name : Sippakon Khammisawang 650610813

Department : Computer Engineering

Report Advisor : Chinawat Isradisaikul, Ph.D.

Degree : Bachelor of Engineering

Program : Computer Engineering

Academic Year : 2025

ABSTRACT

This report presents the results of the cooperative education experience of a Bachelor of Engineering student in Computer Engineering, who served as a Fullstack Developer at G-ABLE Company Limited. During the internship period, the student participated in the software development process for both frontend and backend systems. The main responsibilities included developing and maintaining web applications, designing and implementing APIs for data communication between systems, as well as connecting and managing databases to ensure efficient system performance. All of these experiences played an important role in enhancing the efficiency and flexibility of the software development process, enabling the company to deliver high-quality digital solutions that effectively and promptly meet customer needs.

กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัท จีเอเบล จำกัด (มหาชน) ตั้งแต่วันที่ 16 เมษายน 2568 ถึง วันที่ 15 ตุลาคม 2568 ส่งผลให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่างๆ ที่มีค่ามากmany สำหรับรายงานวิชาสหกิจศึกษาฉบับนี้ สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนจากหลายฝ่าย ดังนี้

1. ปรีชา ปิติจรุณพงศ์ — Lending Business Solution Manager
2. ไชยา แสงจันทร์ — System Analyst

และบุคคลท่านอื่นๆ ที่ไม่ได้กล่าวนามทุกท่านที่ได้ให้คำแนะนำช่วยเหลือในการจัดทำรายงาน ข้าพเจ้าicr'ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่มีส่วนร่วมในการให้ข้อมูล เป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับชีวิตของการทำงานจริง ข้าพเจ้าขอขอบคุณ ไว้ ณ ที่นี่

นายสิบปกร คำมีสว่าง

15 ตุลาคม 2568

สารบัญ

บทคัดย่อ	๑
Abstract	๒
กิตติกรรมประกาศ	๓
สารบัญ	๔
สารบัญรูป	๕
สารบัญตาราง	๖
1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 ขอบเขตของรายงาน	1
1.3 ประโยชน์ที่ได้รับ	1
1.4 ประวัติความเป็นมาของบริษัท	1
1.5 บริการและผลิตภัณฑ์ของบริษัท	2
1.6 ผู้บริหารของบริษัท	2
1.7 งบการเงินและงบกำไรขาดทุน	4
1.7.1 งบการเงิน	4
1.7.2 งบกำไรขาดทุน	5
1.8 หน้าที่ของหน่วยงานที่ได้มาสหกิจ	6
2 รายละเอียดเกี่ยวกับการทำงาน / ผลการปฏิบัติงาน	7
2.1 ปรับความรู้พื้นฐานของการเป็น Fullstack	7
2.1.1 Angular	8
2.1.2 Java Spring Boot	9
2.1.3 SQL Server	10
2.2 TOR	11
2.3 งานที่ได้รับมอบหมาย	11
2.3.1 Other Charge Module	11
2.3.2 Client/Partner/Setting/General Module	21
2.3.3 FA-AR Module	24
2.4 สวัสดิการที่ได้รับ	25
2.5 วัฒนธรรมองค์กร	25
3 สรุปผลการปฏิบัติงาน	26
3.1 สรุปผลการศึกษา/การฝึกปฏิบัติงาน	26
3.2 ประโยชน์ที่ได้รับจากการศึกษา/การฝึกปฏิบัติงาน	26
3.3 ข้อเสนอจากบริษัท	26
บรรณานุกรม	27
ก เอกสารสำคัญที่เกี่ยวข้อง	29

สารบัญรูป

1.1	ผู้บริหารและตำแหน่งของบริษัท	3
1.2	งบการเงินย้อนหลังตั้งแต่ก่อตั้งบริษัท	4
1.3	งบกำไรขาดทุนย้อนหลังตั้งแต่ก่อตั้งบริษัท	5

สารบัญตาราง

1.1 ตารางแสดงตำแหน่งผู้บริหารของบริษัท 2

บทที่ 1

บทนำ

1.1 วัตถุประสงค์

- เพื่อนำความรู้ทั้ง Frontend (การพัฒนาหน้าบ้าน) และ Backend (การพัฒนาหลังบ้าน/ฐานข้อมูล) มาพัฒนาโครงการหรือระบบจริงขององค์กร
- เพื่อฝึกฝนการใช้งานเครื่องมือ (Tools) เฟรมเวิร์ก (Frameworks) และเทคโนโลยีที่ทันสมัยและเป็นที่ยอมรับในภาคอุตสาหกรรม
- เพื่อฝึกฝนการทำงานร่วมกับบุคลากรในสายงานที่เกี่ยวข้อง การสื่อสารในสภาพแวดล้อมการทำงานจริง การวางแผนงานและการจัดการความรับผิดชอบภายใต้กรอบเวลาที่กำหนด
- เพื่อให้นักศึกษาสามารถถวิเคราะห์ วินิจฉัย และหาแนวทางแก้ไขปัญหาทางเทคนิคที่เกิดขึ้นกับระบบจริงได้อย่างเป็นระบบและมีประสิทธิภาพ

1.2 ขอบเขตของรายงาน

รายงานฉบับนี้ครอบคลุมการปฏิบัติงานสหกิจศึกษาในตำแหน่ง Fullstack Developer ณ บริษัท จีเอเบิล จำกัด (G-ABLE Company Limited) โดยเนื้อหาครอบคลุมงานพัฒนาเว็บแอปพลิเคชันทั้งส่วน Frontend และ Backend การออกแบบและพัฒนา API การเชื่อมต่อฐานข้อมูล และการทำงานร่วมกับทีมพัฒนา เพื่อสะท้อนทักษะและประสบการณ์ที่ได้รับจากการฝึกปฏิบัติงานจริง

1.3 ประโยชน์ที่ได้รับ

- ได้พัฒนาความรู้และทักษะด้านการพัฒนาเว็บแอปพลิเคชันทั้งส่วน Frontend และ Backend
- ได้เรียนรู้การทำงานจริงในสภาพแวดล้อมขององค์กรด้านเทคโนโลยีสารสนเทศ
- ได้เพิ่มพูนประสบการณ์ในการทำงานร่วมกับทีมพัฒนาและการใช้เครื่องมือในการพัฒนาซอฟต์แวร์
- ได้นำความรู้ทางวิศวกรรมคอมพิวเตอร์มาประยุกต์ใช้ในการทำงานจริงอย่างมีประสิทธิภาพ

1.4 ประวัติความเป็นมาของบริษัท

G-Able [1] ก่อตั้งขึ้นในปี 2532 ในชื่อ บริษัท โลจิก จำกัด โดยเริ่มจากการเป็นตัวแทนจำหน่ายฮาร์ดแวร์ และโซลูชันไอที ก่อนจะพัฒนาสู่การเป็นผู้นำด้าน Digital Enabler หรือผู้ให้บริการเทคโนโลยีดิจิทัลแบบครบวงจร. บริษัทได้ขยายธุรกิจไปสู่โซลูชันที่หลากหลาย เช่น Cloud, Cybersecurity, Data Analytics, AI และ Software Development.

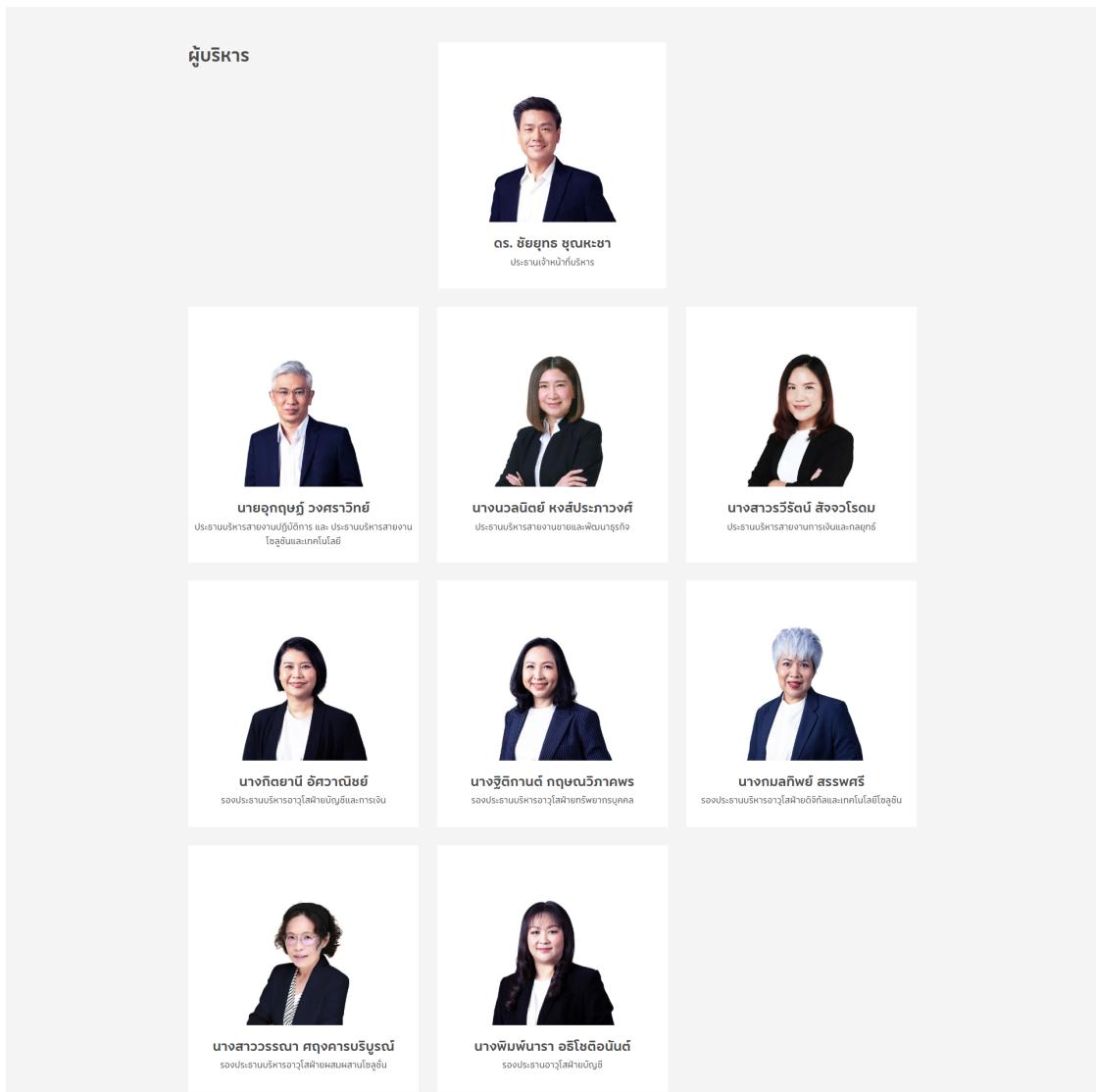
1.5 บริการและผลิตภัณฑ์ของบริษัท

จีเอเปิล ให้บริการด้านดิจิทัลโซลูชั่นแบบครบวงจร [2] เพื่อช่วยให้องค์กรประสบความสำเร็จในการทำ ดิจิทัล ทรานส์ฟอร์เมชั่น โซลูชั่นถูกออกแบบและคิดมาให้ใช้งานได้อย่างราบรื่นตั้งแต่การวางแผนเชิงกลยุทธ์ การออกแบบระบบ และการใช้งานระบบ ไปจนถึงโครงสร้างพื้นฐานเครือข่าย การจัดการพื้นที่เก็บข้อมูล ด้วยบริการระดับมืออาชีพ

1.6 ผู้บริหารของบริษัท

Name	Position
ดร.ชัยยุทธ ชุมพาณิช นายอุกฤษฎ์ วงศารวิทย์	ประธานเจ้าหน้าที่บริหาร ประธานบริหารสายงานปฏิบัติการ และ ประธาน บริหารสายงานโซลูชั่นและเทคโนโลยี
นางนวลนิตย์ วงศ์ประภาวงศ์ นางสาวรรษัต์ สัจจาวโรдум นางกิตยาณี อัศวานิชย์ นางธิติกานต์ กฤษณ์วิภาวด นางกมลทิพย์ สรรพศรี	ประธานบริหารสายงานขายและพัฒนาธุรกิจ ประธานบริหารสายงานการเงินและกลยุทธ์ รองประธานบริหารอาวุโสฝ่ายบัญชีและการเงิน รองประธานบริหารอาวุโสฝ่ายทรัพยากรบุคคล รองประธานบริหารอาวุโสฝ่ายดิจิทัลและเทคโนโลยี โซลูชั่น
นางสาววรรณษา ศุตุรงค์บริบูรณ์ นางพิมพ์นรา อธิชาติอนันต์	รองประธานบริหารอาวุโสฝ่ายผู้ผลิตและโซลูชั่น รองประธานอาวุโสฝ่ายบัญชี

ตารางที่ 1.1: ตารางแสดงตำแหน่งผู้บริหารของบริษัท



รูปที่ 1.1: ผู้บริหารและตำแหน่งของบริษัท

1.7 งบการเงินและงบกำไรขาดทุน

1.7.1 งบการเงิน

Balance Sheet 2022 - 2024

Unit : Baht	2022	2023	2024
Total Asset	3,572,822,237	5,053,859,748 +41.45%	5,331,095,889 +5.48%
Total Liability	2,486,148,224	2,984,174,480 +20.03%	3,301,669,797 +10.63%
Equity	1,086,674,013	2,069,685,268 +90.46%	2,029,426,092 -1.94%

รูปที่ 1.2: งบการเงินย้อนหลังตั้งแต่ก่อตั้งบริษัท

จากข้อมูลใน รูปที่ 1.2 จะเห็นได้ว่าบริษัท **G-Able** มีสินทรัพย์รวมอยู่ที่ 3,573 ล้านบาท ในปี 2022 (ค.ศ. 2022) และเพิ่มขึ้นอย่างรวดเร็วเป็น 5,054 ล้านบาท ในปี 2023 หรือเติบโตประมาณร้อยละ 41.5 ก่อนจะลดลงเป็น 5,331 ล้านบาท ในปี 2024 (+5.5%) ซึ่งสะท้อนถึงการขยายธุรกิจและการบริหารสินทรัพย์ที่มีประสิทธิภาพ ในด้านโครงสร้างทุน บริษัทมี หนี้สินรวม เพิ่มจาก 2,486 ล้านบาท เป็น 3,302 ล้านบาท ขณะที่ ส่วนผู้ถือหุ้น เพิ่มจาก 1,087 ล้านบาท เป็น 2,070 ล้านบาท ในปี 2023 ก่อนจะลดเหลือ น้อยเหลือ 2,029 ล้านบาท ในปี 2024 แสดงว่าบริษัทสามารถควบคุมหนี้สินได้ดีและยังมีทุนเพียงพอต่อการเติบโต

สรุป: G-Able มีสินทรัพย์สูง โครงสร้างทุนแข็งแรง หนี้สินอยู่ในระดับเหมาะสม และยังคงเติบโตอย่างมั่นคงในระยะยาว

1.7.2 งบกำไรขาดทุน

Income Statement 2022 - 2024

Unit : Baht	2022	2023	2024
Total Income	4,308,556,708.00	4,516,553,595.00 +4.82%	5,275,081,560.00 +16.79%
Net Profit	424,586,473.00	188,995,959 -55.48%	175,500,060 -7.14%

รูปที่ 1.3: งบกำไรขาดทุนย้อนหลังตั้งแต่ก่อตั้งบริษัท

รายได้รวมของบริษัท **G-Able** ในช่วงปี 2022–2024 เติบโตต่อเนื่องจาก 4,308 ล้านบาท เป็น 5,275 ล้านบาท หรือเฉลี่ยประมาณร้อยละ 10 ต่อปี แสดงถึงการขยายธุรกิจที่แข็งแกร่ง อย่างไรก็ตาม กำไรสุทธิกลับลดลงจาก 425 ล้านบาท ในปี 2022 เหลือ 176 ล้านบาท ในปี 2024 เนื่องจากต้นทุนและค่าใช้จ่ายที่เพิ่มขึ้น

แนวโน้มในอีก 2–3 ปีข้างหน้า คาดว่ารายได้จะยังเติบโตได้ราวร้อยละ 8–12 ต่อปี โดยอาจแตะระดับ 6,500–7,500 ล้านบาท ขณะที่กำไรสุทธิอาจทรงตัวหรือลดลงเล็กน้อย หากยังไม่สามารถควบคุมต้นทุนได้อย่างมีประสิทธิภาพ ทั้งนี้ การเพิ่มสัดส่วนรายได้จากการซื้อฟร์ェนช์แลนด์ลูชั่นที่มีกำไรสูง จะเป็นกุญแจสำคัญต่อการเติบโตในอนาคต

1.8 หน้าที่ของหน่วยงานที่ได้มาสหกิจ

ในตำแหน่ง Full Stack Developer ที่บริษัท G-ABLE หน้าที่หลักของคือการพัฒนาและปรับปรุงระบบทั้งส่วนหน้า (Front-end) และส่วนหลัง (Back-end) ของเว็บแอปพลิเคชัน โดยมุ่งเน้นให้ระบบทำงานได้อย่างมีประสิทธิภาพและตอบสนองต่อความต้องการของผู้ใช้

นอกจากนี้ ยังได้ทำงานร่วมกับทีม UX/UI เพื่อปรับปรุงประสบการณ์การใช้งานของผู้ใช้ (User Experience) และออกแบบส่วนติดต่อผู้ใช้ (User Interface) ให้ใช้งานได้สะดวกและสวยงาม รวมถึงทำงานร่วมกับทีม QA (Quality Assurance) เพื่อทดสอบ ตรวจสอบ และแก้ไขข้อผิดพลาดของระบบก่อนนำเข้าใช้งานจริง เพื่อให้มั่นใจว่าระบบมีความถูกต้อง เสถียร และพร้อมใช้งานตามมาตรฐานของบริษัท

บทที่ 2

รายละเอียดเกี่ยวกับการทำงาน / ผลการปฏิบัติงาน

2.1 ปรับความรู้พื้นฐานของการเป็น Fullstack

แนวทางในการเริ่มต้นทำงานในสายงาน Fullstack จำเป็นต้องมีการศึกษาและปรับพื้นฐานความรู้ที่สำคัญ เพื่อให้แน่ใจว่าพร้อมสำหรับการทำงานจริง เนื่องจาก Fullstack เป็นสายงานที่มีความใหม่และมีการพัฒนาอย่างต่อเนื่องในวงการซอฟต์แวร์ โดยหัวข้อที่ได้รับมอบหมายให้ศึกษาเพื่อเตรียมความพร้อมจะประกอบด้วย

- **Angular:** Framework สำหรับพัฒนา Front-end Web Application แบบ Single Page Application (SPA) ช่วยให้การจัดการ Component, Template, และ Data Binding มีประสิทธิภาพ
- **Java Spring Boot:** Framework สำหรับพัฒนา Back-end Application ช่วยให้สร้าง REST API และจัดการ Business Logic ได้อย่างเป็นระบบ
- **SQL Server:** ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) สำหรับจัดเก็บและจัดการข้อมูลของแอปพลิเคชัน

ทั้งนี้การศึกษาหัวข้อเหล่านี้มีระยะเวลาประมาณ 2-4 สัปดาห์ และในท้ายที่สุดจะต้องมีการนำเสนอสิ่งที่ได้เรียนรู้ ให้กับพี่ ๆ ในทีมได้ฟังและประเมินว่าพร้อมที่จะทำงานจริงหรือไม่ อย่างไรก็ตามรายละเอียดในหัวข้อย่อยต่าง ๆ หลังจากนี้จะเป็นการนำเสนอสิ่งที่ได้เรียนรู้และได้นำมาประยุกต์ใช้ในการทำงานจริง ส่วนหัวข้อนอกเหนือจากที่จะกล่าวถึงก็สำคัญไม่น้อยเช่นกันแต่จะขึ้นนำเสนอ Documentation ที่ได้ทำสรุปการเรียนรู้มาแล้วนั้นในส่วนภาคผนวก

2.1.1 Angular

Angular คือเฟรมเวิร์กสำหรับฝั่ง Front-end ที่ใช้ในการพัฒนา Web Application แบบ Single Page Application (SPA) ซึ่งช่วยให้นักพัฒนาสามารถสร้างส่วนติดต่อผู้ใช้ (User Interface) ที่มีประสิทธิภาพ ตอบสนองเร็ว และสามารถดูแลรักษา (Maintainable) ได้ง่าย โดย Angular ออกแบบมาให้ทำงานแบบ Component-Based และมีระบบจัดการ Data Binding และ Dependency Injection ที่แข็งแกร่ง

คุณสมบัติเด่นของ Angular

- Component-Based Architecture: แบ่งส่วนของหน้าเว็บออกเป็น Component ย่อย ๆ ทำให้เกิดเป็นระเบียบและน่ากลับมาใช้ซ้ำได้
- Two-Way Data Binding: ช่วยให้ข้อมูลระหว่าง Model และ View เชื่อมโยงกันแบบอัตโนมัติ ลดความซับซ้อนในการอัปเดต UI
- Dependency Injection: จัดการ Service และ Object ที่จำเป็นได้อย่างมีประสิทธิภาพ เพิ่มความยืดหยุ่นในการพัฒนา
- Angular CLI: เครื่องมือ Command Line ที่ช่วยสร้าง Project, Component, Service และ Build แอปพลิเคชันได้รวดเร็วและเป็นมาตรฐาน

ปรับทุกอย่างของ Angular ใน Fullstack Development

- หน้าที่เป็น Front-end ที่รับผิดชอบการแสดงผลและการโต้ตอบกับผู้ใช้
- เรียกใช้งาน RESTful APIs จากฝั่ง Backend (เช่น Spring Boot) เพื่อรับและส่งข้อมูลในรูปแบบ JSON
- จัดการ State ของแอปพลิเคชัน และอัปเดตข้อมูลบนหน้าเว็บโดยไม่ต้อง Reload ทั้งหน้า (SPA)
- ใช้ร่วมกับระบบ Routing ภายในเพื่อสลับหน้าแบบ Dynamic โดยไม่ต้องโหลดจากเซิร์ฟเวอร์ใหม่

2.1.2 Java Spring Boot

Spring Boot คือเฟรมเวิร์กที่ได้รับความนิยมสูงสำหรับภาษา Java (ภาษาโปรแกรมมิ่ง) ซึ่งออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันบนผู้เซิร์ฟเวอร์ (Backend) ได้อย่างรวดเร็ว และง่ายดาย โดยเฉพาะอย่างยิ่งการสร้าง RESTful APIs และ Microservices

คุณสมบัติเด่นของ Spring Boot

- Auto-Configuration: Spring Boot จะตั้งค่าเริ่มต้นที่จำเป็นให้กับโปรเจกต์โดยอัตโนมัติ ทำให้ลดการเขียนโค้ดและไฟล์ตั้งค่า (Configuration) ที่ซ้ำซ้อนลงไปมาก
- Stand-alone Applications: สามารถรันแอปพลิเคชันได้โดยตรงด้วยตัวมันเอง เนื่องจากมี Embedded Server (เช่น Tomcat หรือ Jetty) ติดมาให้ด้วย ไม่จำเป็นต้องติดตั้งหรือตั้งค่าเซิร์ฟเวอร์ภายนอก
- Starter Dependencies: มีชุด Dependencies สำเร็จรูป (Starters) ที่ช่วยในการรวมไลบรารีที่เกี่ยวข้องกันเข้าด้วยกัน (เช่น spring-boot-starter-web สำหรับการสร้างเว็บแอปพลิเคชัน) ทำให้จัดการ Dependency ได้ง่ายขึ้น

บริบทของ Spring Boot ใน Fullstack Development

- สร้าง RESTful APIs ซึ่งเป็นช่องทางสื่อสารมาตรฐาน โดยส่งและรับข้อมูลในรูปแบบ JSON
- Spring Boot ทำหน้าที่จัดการ Business Logic ทั้งหมด เช่น การคำนวณ การตรวจสอบสิทธิ์ (Security) และการประมวลผลข้อมูล
- จัดการการเชื่อมต่อและโต้ตอบกับ Database (เช่น MySQL, PostgreSQL, MongoDB) โดยมักใช้ร่วมกับ Spring Data JPA/Hibernate

2.1.3 SQL Server

Microsoft SQL Server คือระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ที่พัฒนาโดยบริษัท Microsoft ใช้สำหรับจัดเก็บ จัดการ และเรียกใช้ข้อมูลอย่างมีประสิทธิภาพ โดยรองรับการทำงานร่วมกับภาษา SQL (Structured Query Language) ซึ่งเป็นภาษามาตรฐานในการจัดการฐานข้อมูล

คุณสมบัติเด่นของ SQL Server

- Relational Database: จัดเก็บข้อมูลในรูปแบบตารางที่มีความสัมพันธ์กัน (Relation)
- Security and Authentication: รองรับการเข้ารหัสข้อมูล การจัดการสิทธิ์ของผู้ใช้ และการเชื่อมต่อแบบ Windows Authentication
- Backup and Recovery: มีระบบสำรองและกู้คืนข้อมูลที่เชื่อถือได้ ป้องกันข้อมูลสูญหาย
- Performance Optimization: มีเครื่องมือช่วยวิเคราะห์และปรับปรุงประสิทธิภาพ เช่น Query Optimizer และ Indexing

บริบทของ SQL Server ใน Fullstack Development

- ทำหน้าที่เป็นระบบจัดเก็บข้อมูลหลักของแอปพลิเคชัน เช่น ข้อมูลผู้ใช้ สินค้า หรือรายการคำสั่งซื้อ
- ใช้ร่วมกับ Backend Framework เช่น Spring Boot เพื่อเชื่อมต่อและเรียกใช้ข้อมูลผ่าน ORM (เช่น Hibernate หรือ JPA)
- รองรับการเขียน Stored Procedures และ Views เพื่อช่วยประมวลผลข้อมูลภายในฐานข้อมูลโดยตรง

2.2 TOR

เนื่องจากการมาสหกิจศึกษาจำเป็นต้องมีการประเมินที่เข้มงวด ดังนั้นจึงได้จัดทำ TOR ขึ้นเพื่อเป็นมาตรฐาน และข้อตกลงในการทำงาน และประเมินผลร่วมกับบริษัทและอาจารย์ เพื่อให้มั่นใจว่างานสามารถทำได้ตาม เป้าหมายของ Fullstack Project

- ทำงานในลักษณะของ Task-based โดยกำหนดจำนวนขั้นต่ำของ Task ไว้ที่ 60 tasks เนื่องจากทีม Fullstack ทำงานตามแนวทาง Agile โดยแบ่งงานเป็น Sprint ทุก Sprint จะมีการ Planning เพื่อกำหนดลำดับความสำคัญของ Task และมีการประชุม Weekly-Update ทุกวันพุธเพื่อติดตามความคืบหน้า ทำให้การทำงานมีความรอบคอบและสามารถปรับเปลี่ยนได้ตามความต้องการของ Product Owner และลูกค้า

ดังนั้นหัวข้อถัดไปจะเป็นการนำเสนอ Task งานที่ได้รับมอบหมายทั้งหมด เพื่อแสดงให้เห็นว่าสามารถบรรลุเป้าหมายของ TOR ได้ตามแผนงาน Agile

2.3 งานที่ได้รับมอบหมาย

สิบเนื้องจากงานที่ได้รับมอบหมายจะเป็นในลักษณะ Task งาน ดังนี้ใน Section นี้จะนำเสนอ Task งาน ทั้งหมด พร้อมรายละเอียดของแต่ละ Task งาน โดยจะแบ่งออกเป็น 3 กลุ่มหลัก ดังนี้:

- TASK:** งานพัฒนาไฟอร์ใหม่หรือทำงานตาม requirement ที่กำหนด
- CR (Change Request):** งานปรับปรุงหรือปรับแก้ตามคำขอเพิ่มเติมจากผู้ใช้งานหรือ Product Owner
- DEFECT:** งานแก้ไขบคหรือปัญหาที่เกิดขึ้นในระบบ

การแบ่งกลุ่ม Task ดังกล่าวช่วยให้สามารถติดตามความคืบหน้าในลักษณะ Agile ได้อย่างชัดเจน และสะท้อนถึงความสามารถในการจัดการงานทั้งการพัฒนา การปรับปรุง และการแก้ไขปัญหา

2.3.1 Other Charge Module

2.3.1.1 Task

- GISLOCK-5943 [FA-Other charge request]** จัดทำหน้าจอ **Other charge request** เป็นการเพิ่มโปรแกรมใหม่ในแพนกของการเงิน โดยการเป็นสร้างโปรแกรมที่ชื่อว่า Other Charge Request และ Other Charge Request List
Other Change Request : เป็นโปรแกรมเกี่ยวกับการเรียกค่าใช้จ่ายเพิ่มเติมนอกเหนือจากเบี้ยประกัน
Other Change Request List : เป็นหน้าที่เอาไว้รายการที่เราสร้างไว้
- GISLOCK-6755 [Set up Transaction Code]** เพิ่มหน้าจอ **Set up Transaction Code** เป็นการเพิ่มโปรแกรมใหม่ที่เราจะนำไปตั้งค่ารายการ Transaction Code ที่จะใช้ในโปรแกรม Other

Charge Request กับ Other Expense Request

โดยจะแบ่ง 2 โปรแกรมคือ Set up Transaction Code และ Set up Transaction Code List

- **GISLOCK-7019 [Other Charge] เพิ่มปุ่ม Update (สำหรับ Refresh) ข้อมูล Installment และ ShareDept**

เป็นการเพิ่มปุ่ม Update ในหน้าจอ Other Charge Request เพื่อให้ผู้ใช้สามารถกดปุ่มนี้เพื่อรีเฟรชข้อมูลในตาราง Installment และ Share Department ได้ แบบอัตโนมัติ

- **GISLOCK-7106 [Other Charge] แก้ไขโปรแกรม Other charge Request List เพิ่มเงื่อนไข**

ในการ query เพื่อค้นหาข้อมูลโดยกรองเฉพาะ Entry_by ที่ตรงกับ User Login

เป็นการแก้ไขโปรแกรม Other Charge Request List โดยเพิ่มเงื่อนไขในการค้นหาข้อมูลให้สามารถกรองเฉพาะ Entry_by ที่ตรงกับ User Login นั้น ๆ ได้

อนาคตจะแบ่งให้แสดงแค่แผนกที่ตัวเองรับผิดชอบ

- **GISLOCK-7200 [Other Charge] แก้ไขโปรแกรม Charge Request ส่วนของการ Share dept Installment**

ปรับการแสดงข้อความเตือน เมื่อมีการกรอกข้อมูลไม่ครบถ้วนในส่วนของ Share Department และ Installment

โดยจะมีการเช็คให้ข้อมูลของ Installment และ Share Department ต้องมีค่าเบอร์เซ็นต์รวมกันเท่ากับ 100% ถึงจะสามารถบันทึกข้อมูลได้

และจะต้องมีจำนวนเงินที่สัมพันธ์กับจำนวนเงินใน Transaction ด้วย

- **GISLOCK-7289 [Other Charge] Set up Transaction Code เพิ่ม DropdownFlag**

เป็นการเพิ่มตัวเลือก DropdownFlag ในหน้าจอ Set up Transaction Code

เพื่อให้ผู้ใช้สามารถเลือกได้ว่ารายการ Transaction Code ที่สร้างขึ้นมานั้น จะนำไปแสดงในหน้าจอ Other Charge Request หรือไม่

- **GISLOCK-7688 [Other Charge] Other Charge Request Booking**

เป็นการเพิ่มโปรแกรมในส่วนของการแผนกการเงิน เป็นโปรแกรมที่แผนกการเงินจะเอาไว้ตรวจสอบหรือแก้ไขรายการที่ถูกอนุมัติแล้ว

โดยจะมีการแยกเป็น 2 โปรแกรมคือ Other Charge Booking และ Other Charge Booking List

- **GISLOCK-7689 [Other Charge] Other Charge Request บันทึกขาคู่**

เป็นส่วนหนึ่งของโปรแกรม Other Charge Request ที่เพิ่มขึ้นมา

โดยจะเป็นการบันทึกข้อมูลแบบขาคู่ คือการที่เราจะต้องบันทึกข้อมูลทั้งในฝั่งของ Other Charge Request และ Other Expense Request พร้อมกัน

โดยจะมี 2 แบบ คือ :

1. แบบ auto ที่จะสร้าง Other Expense Request ให้อัตโนมัติเมื่อมีการบันทึก Other Charge Request

- แบบ manual ที่จะให้ผู้ใช้เป็นคนเลือกว่าจะสร้าง Other Expense Request ด้วยตัวเอง

2.3.1.2 Defect

- **GISLOCK-5457 [Other Charge]** ปรับการแสดงผลข้อมูลในส่วน Bill to ให้ไปในทางเดียว กับ AP
เพิ่มการ Clear โดยการกดปุ่ม X ที่ Bill To
- **GISLOCK-6189 [Other Charge/Expense] Standard Search Search like เวลาเก็บ Data ใน DB ให้ค้นด้วย Space Bar**
 1. ปรับตรง placeholder ของการ Search เราจะເອການເວັ້ນວຽກໂຄກ ຈະແສດງໃຫຍ່ໄດ້ / คົ້ນ
 2. ตรง Database ຈະປັບຈາກການເກີບທີ່ຕົ້ນໂດຍ | ມາໃຊ້ເປັນ " " (Spacebar ອີ່ວິ້າ ການເຄາະຂ່ອງວ່າງແທນ)
- **GISLOCK-6518 [FA- Other charge request]** ปรับ feature ທຳນ້າຈອ Other charge request
ເປັນການແກ້ທາງໆ ຈຸດໃນໂປຣແກຣມ Other Charge Request ທີ່ໄດ້ມີການພັນນາຫຼືນມາ ໂດຍມີການປັບປຸງດັ່ງນີ້
 1. ປັບປຸງການແສດງ ປຸ່ມ Add/Clear ຂອງສ່ວນ Share Department ໃຫ້ຕຳແໜ່ງປັບຕິພາດທຳນ້າຈອ
 2. ປັບປຸງທຳນ້າ Approve ເມື່ອກົດ view mode ມາຕ້ອງໄຟມີປຸ່ມ Add/Clear ຂອງແຕ່ລະສ່ວນແສດງ
 3. view mode ທີ່ກົດມາຈາກທຳນ້າ List ສ່ວນຂອງ Transaction ຈະຕ້ອງເອົາຂໍ້ມູນຂອງຊຸດແຮກມາແສດງ
 4. Hq/Branch ແສດງເປັນຂໍ້ມູນ ໄນຕ້ອງແສດງໂຄດ
 5. Exchange Rate ໃນ Transaction ໃຫ້ແສດງ 8 ຕຳແໜ່ງ
 6. ເມື່ອເຮັດວຽກ Transaction ແລ້ວເຮົາຈະກຳນົດ Currency/Exchange Rate ໄນໃຫ້ແກ້ໄຂໄດ້ ເພື່ອ Transaction ຕ່ອໄປຈະໄດ້ Currency/Exchange Rate ເໝືອນກັນ
 7. ກຣີນີ້ເຮົາມີ Installment ອີ່ວິ້າ Share Department ແລ້ວເຮົາໄປແກ້ໄຂ Transaction ເມື່ອຈະເຫັນ ເຮົາຈະຕ້ອງເຕືອນບອກວ່າ Transaction ມີເປົ້າຢັ້ງແປງ ແລະ ເຫັນໄມ້ໄດ້
- **GISLOCK-6561 [Other Charge] Pop up Bill To ແສດງຂໍ້ມູນເດືອກກັນ ຊ້າງ**

ສາເຫຼຸດທີ່ຂໍ້ມູນແສດງຊ້າງ ເນື່ອງຈາກມີການເຊື່ອມໂຍງ (JOIN) ກັບຕາງໆຂໍ້ມູນອື່ນ ເຊັ່ນ ຕາງໆທີ່ອູ່ (Address) ຊຶ່ງມີຫລາຍໄຮຍການຕ່ອຮ້າສຸກຄ້າຫຼືພາർທນອ້າ ຮຳໃຫ້ຂໍ້ມູນທັກນີ້ຮ່າງກວດສະບັບຕາມ ຈຳນວນຂໍ້ມູນທີ່ອູ່ທີ່ສັນພົບກັນໃນຕາງໆດັ່ງກ່າວ

ວິທີແກ້ໄຂຄື່ອງ ການປັບຄຳສັ່ງ SQL ທີ່ໃຊ້ໃນການດຶງຂໍ້ມູນ ໂດຍການໃໝ່ຄຳສັ່ງ DISTINCT ເພື່ອໃຫ້ແສດງ ເນິພາຮ່າງກວດທີ່ໄໝຊ້າກັນ ອີ່ວິ້າການປັບໂຄງສ່າງການ JOIN ໃຫ້ເໝາະສົມກັບຄວາມຕ້ອງການຂອງຂໍ້ມູນ ທີ່ຈະແສດງ
- **GISLOCK-6563 [Other Charge] Contact Information** ຈະຕ້ອງດຶງຂໍ້ມູນມາຈາກ Contact Person ຂອງ Client ອີ່ວິ້າ ອີ່ວິ້າ Partner ມາແສດງໂດຍ link up ກັບ Bill to ທີ່ໄດ້ເລືອກ

ปรับปรุงตรง Contact Information :

1. ดึงข้อมูลมาแสดงอัตโนมัติตาม Main Contact
2. กรณีมีหลาย Contact เปลี่ยนได้กดเปลี่ยนตรง modal จะมี Contact ที่สามารถเปลี่ยนได้

- **GISLOCK-6566 [Other Charge]** ปรับ wording ของชื่อฟิลด์ ทั้งหน้า Request และ List เป็นการแก้ไขคำขอของหน้า Request และ List
- **GISLOCK-6568 [Other Charge]** ตรวจสอบการ Save Draft ในงานและการแสดงค่าเริ่มต้นฟิลด์ Statusเพิ่มการเช็คตอนจะ Save Draft ใบคำขอ ให้เช็คว่าเราเลือกตัวที่ต้อง require หรือยัง ถ้ายังให้แจ้งเตือน และปรับการแสดงค่าเริ่มต้นของฟิลด์ Status ให้แสดงเป็น "Save Draft" เมื่อสร้างใบคำขอใหม่
- **GISLOCK-6571 [Other Charge]** ส่วน Share Department ปรับการแสดงปุ่ม Add Clear และเมื่อเลือกข้อมูล Department รูปแบบฟอร์มต้องไม่ขึ้นไปมาปรับปรุงการแสดงปุ่ม Add และ Clear ในส่วนของ Share Department ให้มีตำแหน่งที่คงที่ ไม่ขึ้นไปมาเมื่อมีการเลือกข้อมูล
- **GISLOCK-6575 [Other Charge]** การใช้งานปุ่ม View ในส่วน Other Charge Transactionเพิ่มปุ่ม Back ในหน้าจอ View ของส่วน Other Charge Transaction เพื่อให้ผู้ใช้สามารถกลับไปยังหน้าจอ ก่อนหน้าได้อย่างสะดวก
- **GISLOCK-6576 [Other Charge]** กรณี Total Amount ใน Transaction และ Installment มียอดไม่เท่ากัน เมื่อ Save Submit จะต้องมี Pop up ตาม เมื่อ Total Amount ในส่วนของ Transaction และ Installment มียอดไม่เท่ากัน ระบบจะต้องแสดง Pop-up แจ้งเตือนผู้ใช้ว่า ยอดรวมไม่ตรงกัน และให้ผู้ใช้ทำการตรวจสอบและแก้ไขข้อมูลก่อนที่จะทำการ Save หรือ Submit ใบคำขอ
- **GISLOCK-6577 [Other Charge List]** ปรับการใช้งาน Request Date From / To
 1. เพิ่มตระ Coondition Search โดยการเพิ่มการค้นหาด้วย Status 2. ไม่ Default วันที่ให้ต้อนเริ่ม เพราะให้มาเลือกเอง
- **GISLOCK-6578 [Other Charge List]** ฟิลด์ Bill To กดแล้วไม่แสดง Pop up และขอความหมายของ Request No. Running No. C2568XXXXX สาเหตุที่กดแล้วไม่แสดง Pop-up เกิดจากการที่เราไปเรียก id ของ Pop-up นั้นไม่ถูกต้อง
- **GISLOCK-6583 [Other Charge List]** ไม่แสดงข้อมูล Bill to Name เกิดจากการที่ก่อนหน้านี้มีการเก็บข้อมูล Bill to Name ในตารางที่ไม่ถูกต้อง

วิธีแก้ไขคือ ตอนที่เราจะเชฟใบงานเราวางเลือกให้ถูกฟิล์ดพอตันจะเรียกมาหน้า List ก็จะได้ข้อมูลที่ถูกต้อง

- **GISLOCK-6589 [Other Charge List]** การใช้งานและความหมายของ **Submit Date** เพิ่มคำอธิบายให้ฟิล์ด Submit Date ในหน้า Other Charge List
- **GISLOCK-6591 [Other Charge]** ปรับชื่อฟิล์ด **Entry By** และ **Entry Date** ให้กับหน้าจอ **Request** และ **List**
เปลี่ยนมาใช้ชื่อฟิล์ดเป็น Entry By และ Entry Date ในหน้าจอ Request และ List
- **GISLOCK-6592 [Other Charge List]** เพิ่มการแสดงข้อมูลในคอลัมน์ **Invoice Date**
เพิ่มการแสดงข้อมูลในคอลัมน์ Invoice Date ในหน้า Other Charge List
- **GISLOCK-6649 [Other Charge][Request Other Charge]** ปรับการทำงานส่วน **Contact Information** ให้มีฟังก์ชันการทำงานเหมือนกับ AP
ปรับการทำงานตามนี้
 1. เมื่อเลือก Bill to Code แล้วให้ดึงข้อมูล Contact Information มาแสดงอัตโนมัติ โดยเลือกจาก All Purpose เป็นลำดับแรก
 2. หากมีหลาย contact สามารถกดปุ่มแวนขยายเพื่อแสดง modal ของ contact ที่มีหลายอันแล้วเลือกได้
- **GISLOCK-6675 [Other Charge][Request Other Charge]** ส่วน **Request Information** ทำการ **Edit** ข้อมูลในตารางแต่ระบบไม่ **Update** ตาม
เกิดจากการที่เราส่งข้อมูลไปแต่่ว่าไอดีในการอัปเดตข้อมูลนั้นไม่ถูกส่งไป เลยไม่รู้ว่าจะแก้ไขที่ข้อมูลไหน ทำการแก้ไขโดยการส่งไอดีเพิ่มไปด้วย
- **GISLOCK-6773 [Other Charge][Request Other Charge] Submit** ใบงาน ระบบแสดง **The sum of all tables is not equal.**
เพิ่มการแสดง Modal เมื่อผลรวมของ Total Amount ไม่เท่ากันใน Installment และ Share Department
- **GISLOCK-6877 [OTC][Request Other Charge]** กรณีเลือก Edit รายการเพื่อตรวจสอบข้อมูล แต่ **Entry Date** แสดงเป็นวันที่ปัจจุบัน ซึ่งไม่ถูกต้องให้แก้ไขเป็นวันที่บันทึกใบงานตั้งต้น จากเดิมก่อนหน้านี้เป็นการ mock up ข้อมูลมาเลยทำให้การแสดงเป็นวันปัจจุบันเสมอ เราจะแก้ไขใน mode edit ให้ไปดึงข้อมูลจากฐานข้อมูลมาแสดงแทน
- **GISLOCK-6878 [OTC][Request Other Charge]** กรณีเลือก **Invoice Name** ฟิล์ด **Invoice Address** จะต้องแสดงข้อมูลที่สัมพันธ์กับ **Name** ที่เลือก
ปรับปรุงการทำงานของฟิล์ด **Invoice Address** ให้แสดงข้อมูลที่สัมพันธ์กับ **Invoice Name** ที่ผู้ใช้

เลือก

โดยการเขื่อมโยงข้อมูลจากฐานข้อมูลเพื่อดึงที่อยู่ที่ตรงกับชื่อที่เลือกมาแสดง

- **GISLOCK-6945 [Other Charge List]** ปรับการแสดง Location และ Nationality กรณี Bill To เป็นข้อมูลเก่า โดยระบบต้องแสดงข้อมูลที่ได้เลือกจากตอนบันทึกเดิมที่เรามีการเก็บ เป็นค่า location ประมาณนี้แต่เรามีการปรับรูปแบบใหม่ให้เป็นคีย์ L แบบนี้แทน เราเลยมีเขียน sql เพื่อดึงข้อมูลเดิมมาแล้วดักแคสที่ส่งออกไป ตอนเราจะดึงข้อมูลมาแสดงก็จะได้ข้อมูลใหม่ที่เป็นคีย์ L แทน แล้วตอนเชฟเราก็จะเชฟเป็นคีย์ L ไปเลย ทำให้เป็นการอัปเดตข้อมูลให้เป็นรูปแบบใหม่ทั้งหมด
- **GISLOCK-6953 [Other Charge]** 1.แก้ไขการแสดง Request Information 2.แก้ไขการแสดง Currency/Exchange Rate 3.แก้ไขการแสดง Bill to Code
 - 1. เมื่อทำการกด Icon Edit ข้อมูล Transaction Code ในตารางและไปกดปุ่ม Request Information ข้อมูลเดิมจะต้องไม่หาย
 - 2. เมื่อ Edit เข้ามาจากหน้า List ถ้ามี Transaction อู้เราะจะ Disable Currency และ Exchange Rate เพราะจะใช้ตามอันที่มี 3. เมื่อ x กากบาท Code หรือ x กากบาท Pop up จะต้องเรียก Service ใหม่ทุกครั้ง เป็นการ refresh ทุกครั้งที่เรากดมาใหม่
- **GISLOCK-7031 [Other Charge]** ปรับการทำงานปุ่ม Submit และ Save Draft จากเดิมเมื่อเป็น view mode ปุ่ม Submit กับ Save Draft มันจะ Disable แต่ยังกดและทำงานได้ เราเลยปรับให้ปุ่มมันกดไม่ได้เลยเมื่อเป็น view mode
- **GISLOCK-7064 [Other Charge]** เมื่อ Submit and Save Draft ข้อมูลที่บันทึกไว้ในหน้า จึงต้องไม่เคลียร์และ Redirect to List
 - จากอันเดิมก่อนหน้านี้พอเรากด Submit หรือ Save Draft ข้อมูลนั้นจะเคลียร์ออกจากหน้าจอและนำทางไปสู่หน้าลิสต์
 - เราเลยปรับให้มีกด Submit หรือ Save Draft ข้อมูลที่บันทึกไว้ยังคงอยู่ในหน้าจอ และจะ Redirect to List เช่นเดิมเมื่อผู้ใช้เลือกที่จะกลับไปยังหน้าลิสต์ด้วยตัวเอง
- **GISLOCK-7144 [Other Charge List]** ยอด Amount แสดงไม่ตรงกับการบันทึกในหน้า Request
 - เปลี่ยนจากเดิมเราจะใช้ผลรวมของ Amount ในตาราง Transaction มาแสดง เป็นการไปดึงยอดผลรวมของ Total Amount มาแสดงแทน
- **GISLOCK-7215 [Other Charge]** ตรวจสอบการคำนวณ Payment Due Date ให้ตรงตามเงื่อนไข
 - ปรับปรุงการคำนวณ Payment Due Date ในหน้าจอ Other Charge Request โดยจะมีเงื่อนไขดังนี้
 1. การคิด Payment Due Date จะเป็น วันที่สร้างใบคำขอ + payment term ของคนนั้นๆ แต่ถ้า

ไม่มีจะเป็นค่าเริ่มต้นให้ 30 วัน

2. ถ้า Payment Due Date ตรงกับสารอาทิตย์ และ วันหยุดนักขัตฤกษ์ จะเลื่อนไปเป็นวันทำการถัดไปโดยอัตโนมัติ

- **GISLOCK-7219 [Other Charge]** เพิ่มการแจ้งเตือน Share Department และ Installment กรณีที่ Share (%) มีจำนวน 0.00 และมากกว่า 100.00
ปรับการแจ้งเตือนจากเดิมไม่มีจุดทวนนิยม เป็นการเพิ่มจุดทวนนิยม 2 ตำแหน่งในการแจ้งเตือนเมื่อ Share (%) มีจำนวน 0.00 และมากกว่า 100.00 เพิ่มการตักเคสที่กรอกค่าเกิน 100.00 และน้อยกว่า 0.00 หรือเท่ากับ 0.00
- **GISLOCK-7230 [Other Charge]** แก้ไขการกดปุ่ม Clear และ เพิ่ม Popup has been saved ให้กับการเพิ่มข้อมูลลงตาราง เพิ่มการแสดง modal เมื่อ add update และ delete เพื่อบอกว่าทำเสร็จแล้วของ Transaction , Request Information , Share Department , Installment
- **GISLOCK-7241 [Other Charge]** ลบรายการ Transaction ออกไปแล้ว ข้อมูลส่วน Installment และ ปุ่ม Update ยังแสดงอยู่ในหน้าจอ เพิ่มการเช็คตอนที่เราจะเช็คค่าใหม่ให้อัปเดตตามเงื่อนไข
- **GISLOCK-7243 [Other Charge]** กดปุ่ม Update และไม่ Update ข้อมูล กรณี Installment และ Share Dept มีข้อมูลทั้งคู่ เกิดจากการที่เราส่ง payload ไปให้หลังบ้านไม่ถูกเลยทำให้การอัปเดตทำงานไม่ถูกต้อง แก้ไขโดยการปรับ payload ก็จะสามารถอัปเดตได้ปกติ
- **GISLOCK-7258 [Other Charge]** เพิ่มการปรับ % ของ Share Department และ Installment เพิ่มการคำนวนเบอร์เซ็นต์ให้อัตโนมัติ เช่น เริ่มต้นเรามี 100 แล้วเราเพิ่มอั่นแรกไป 20 แล้วมันก็จะเหลือ 80 ให้ใช้ครั้งต่อไป รวมถึงการคำนวน total amount ด้วย
- **GISLOCK-7395 [Other Charge]** เมื่อเลือก Vat Calculation = Non Vat ให้ Disable Field 1.Vat Type 2.Vat Rate % 3.Vat Amount จะทำการปิดฟิล์ดตั้งกล่าวเมื่อเลือก Vat Calculation เป็น Non Vat
- **GISLOCK-7419 [Other Charge]** ส่วน Contact Information เพิ่มเงื่อนไขเมื่อไม่มีการเลือก Code ให้ Disable ส่วน Contact ไว้ก่อนเสมอ ปรับปรุงให้เมื่อเราเข้ามาครั้งแรกตอนที่เรายังไม่เลือก ลูกค้าหรือพาร์ทเนอร์ เราจะ Disable ส่วนของ Contact ไว้ก่อน

- **GISLOCK-7439 [Other Charge] Vat Rate % = 0 ให้ Disable Vat Amount**
ใน Transaction เราเลือก Vat Rate เป็น 0 ก็จะ Disable Vat Amount ไปเลย เพื่อจะได้ไม่สามารถแก้ไขได้ เพราะเราเลือก 0 แล้ว
- **GISLOCK-7487 [Other Charge] ตรวจสอบการค้นหาข้อมูลแบบ ค้นหาเมื่อง่าวงระหว่างคำปรับรูปแบบการ search ใหม่ จากเดิมเราจะ search ได้ทีละตัว แต่เราจะทำการเพิ่มไป เป็นการ search แบบหลายตัว เช่น "AA BA"
เราจะแสดงผลลัพธ์ที่ตรงกับ AA มาทั้งหมดก่อน ตามด้วย BA ต่อด้วย**
- **GISLOCK-7494 [Other Charge] ปรับตำแหน่งการแสดงปุ่ม Search และ Clear
ปรับตำแหน่งการแสดงปุ่ม**
- **GISLOCK-7498 [Other Charge] Set up Transaction Code แก้ไขตามรายการ**
 1. Account Code ทำเป็น * พิล์ดเพื่อป้องบันการ Add ลงตารางโดยที่ไม่เลือกข้อมูล
 2. เพิ่มการทำงานของปุ่ม Clear หลัก
 3. ปรับ Icon : edit, delete ให้ตรงกับโปรแกรม Standard
 4. Account Code จะต้องแสดงแต่ Code และ Account Name สามารถดูได้อย่างเดียวไม่สามารถแก้ไขได้
 7. ปรับพิล์ด Type มาเป็น Transaction Type ตามหน้า List 8. เพิ่มการ Validate WHT Rate * และเป็น require field
- **GISLOCK-7529 [Other Charge] โปรแกรม Set up Transaction Code ทำการ Delete รายการในตารางแล้ว รายการไม่ถูกลบไป**
ปัญหาเกิดจาก Race Condition คือคำสั่ง GET ทำงานก่อนที่คำสั่ง DELETE จะเสร็จสมบูรณ์ ทำให้เห็นข้อมูลที่ยังไม่ถูกลบจริง รอให้การลบเสร็จสิ้นก่อนดึงข้อมูล หรือใช้ Transaction/Locking เพื่อควบคุมลำดับการทำงานไม่ให้คำสั่งชนกัน.
- **GISLOCK-7554 [Other Charge] ลบรายการที่มี Vat Amount ที่ส่วน Transaction ออกไปแล้ว แต่ในส่วน Installment ยังคงยอด Vat Amount เก่าไว้**
เกิดจากการที่เราทำการแก้ไขใน Transaction และในส่วนของ Installment ไม่ได้รู้ว่ามันเปลี่ยน เลยแก้ไขโดยการสร้างฟังก์ชันที่จะเช็คว่าถ้ามีการเปลี่ยนแปลงและเข้าเงื่อนไขที่เราตั้งก็จะไปทำฟังก์ชันนี้ก็สามารถทำงานได้ปกติ
- **GISLOCK-7587 [Other Charge] Set up Transaction Code เปลี่ยนจากปุ่ม Submit/Clear เป็น Save/Clear
เปลี่ยนหน้า UI ของปุ่ม**
- **GISLOCK-7626 [Other Charge] เปลี่ยนจากการส่งข้อมูลแบบ queryParams เป็นแบบ**

getState

ปรับจากเดิมที่เป็นการรับมาจาก Params เราปรับเปลี่ยนมาใช้ในรูปแบบ State ซึ่งปลอดภัยและใช้งานได้ดีกว่า

- **GISLOCK-7628 [Other Charge]** ปรับ status เป็นตัวกลางเพื่อให้ sort จะได้ เนื่องจากตอนนี้เป็นการใช้เงื่อนไขในการแปลงจาก code เป็น word และนำมาระดับ ไม่ใช่การ Join เพิ่ม status ของ Other Charge ให้เป็น Workflow ของตัวกลางเพื่อที่จะได้เข้ากระบวนการ approval ของส่วนกลาง และ จะสามารถนำมาระดับ Sort ได้
- **GISLOCK-7629 [Other Charge]** ลบข้อมูลในตาราง **Installment** แล้วในตาราง **Transaction** มีแต่ค่า **VATNO** ระบบไม่ **Disable Field Vat** ในส่วน **Installment** ให้เกิดจากตอนที่เรา แก้ไขข้อมูลใน Installment แล้วเราไม่ได้เช็คของ Transaction เลยทำให้ตอนเราลบแล้วจะเช็ตใหม่เลยกลับมาเปิด Vat Amount ทั้งๆที่ใน Transaction ยังมี VATNO เราแก้ไขโดยการเพิ่มการเช็คไปตอนเราเคลียร์เพื่อจะเช็ตใหม่ ก็จะทำให้ฟิลเตอร์เปิดปิดตามข้อมูล Transaction ที่รามี
- **GISLOCK-7631 [Other Charge][Set up Transaction Code] Mode Edit** ข้อมูลที่เคยบันทึกไว้ เช่น **WHT Rate** กับ **Show Trans code** ไม่ตรงกับที่บันทึก ปัญหานี้เกิดจาก type ของ ฟิลเตอร์ WHT Rate ไม่ตรงกับ Database เราเลยแก้ไขโดยการปรับ type ของฟิลเตอร์ WHT Rate ให้ตรงกับ Database ก็จะสามารถดึงข้อมูลมาแสดงในโหมด Edit ได้ถูกต้อง
- **GISLOCK-7694 [Other Charge]** แก้ไขการค้นหาช่วงวันที่ **Date From** และ **Date To** ให้ตรงกับช่วงที่ค้นหา จากเดิมตอนเราเลือกใน format เป็น ISO 8601 อาจมีปัจจัยเรื่อง Timezone เข้ามายกเวื่องทำให้การค้นหาไม่ตรงกับช่วงวันที่ที่เลือก เราเลยแก้ไขโดยปรับเป็นรูปแบบ dd/mm/yyyy แทน ที่นี่ก็จะไม่มีปัญหาระบطةเรื่อง Timezone แล้ว ก็จะทำให้การค้นหาตรงกับช่วงวันที่ที่เลือก
- **GISLOCK-7737 [Other Charge] Set up Transaction Code List กดปุ่ม Add แล้วไม่แสดงหน้าจอ Set up Transaction Code** เราปรับรับจากเดิมที่มีการใช้ **getParam** มาเป็นการใช้ **getState** พอเรากด Add มาจากหน้า List อาจทำให้ State ที่ส่งมาไม่ครบ เราเลยแก้ไขให้ตอนกด Add มันจะไม่เอา State อะไรมาส่งเลย ทำให้มัน Redirect มาที่หน้า Set up Transaction Code List ปกติ แก้ไขโดยการ set ค่า default ให้กับ State ที่ขาดหายไป
- **GISLOCK-7793 [Other Charge]** ส่วน **Share Department** ขาดการแสดงคอลัมน์ **Department Name** ที่ตารางแสดงผล เพิ่มส่วนของ Department Name ในตารางแสดงผลของ Share Department

โดยการดึงข้อมูลชื่อแผนกจากฐานข้อมูลมาแสดงในตาราง

- **GISLOCK-7807 [Other Charge] OTC List กด link condition search และกดปุ่ม search ระบบแจ้งเตือน Invalid column name 'OTC_BILLTO_CD'.**
เกิดจากการที่เราปรับ Database ใหม่ ทำให้ชื่อคอลัมน์เปลี่ยนไป
เราเลยแก้ไข SQL query ให้ตรงกับชื่อคอลัมน์ใหม่ที่ถูกต้อง
- **GISLOCK-7847 [Other Charge] ฟิลเตอร์ Vat Amount ส่วนของ Installment ไม่ Disable Field กรณี Transaction Code เป็น Non Vat ทั้งหมด (Edit จากหน้าจอ List เข้ามา)**
เพิ่มการเช็คข้อมูลตอนกด edit ว่าในตาราง Transaction มีแต่ Non Vat หรือไม่
ถ้ามีแต่ Non Vat ก็จะทำการ Disable Field Vat Amount ในตาราง Installment ให้อัตโนมัติ
- **GISLOCK-7860 [Other Charge List] Action Recall ไม่สามารถใช้งานได้**
จากเดิมเราจะเป็นการเขียน mock up ขึ้นมา ทำให้ Action Recall ไม่สามารถใช้งานได้
เราเลยปรับให้ Action Recall สามารถใช้งานได้จริง โดยการเชื่อมต่อกับฐานข้อมูลและดึงข้อมูลที่ถูกต้องมาแสดง

2.3.1.3 CR

- **GISLOCK-6697 [OTC][Request OTC] ปรับการแสดงส่วน Invoice 1.Invoice Name 2.Invoice Address 3.Invoice Date**
ปรับการแสดง Invoice จากเดิมที่แสดงข้อมูลในรูปแบบ modal
เป็นการแสดงข้อมูลในรูปแบบของ field ที่แสดงบนหน้าจอปกติแทน
- **GISLOCK-6715[OTC][OTC List] แก้ไขชื่อคอลัมน์ และ ดึงข้อมูลมาแสดง หลังมีการ Demo 1 - 22/7/68**
ปรับจากเดิมหลังจากที่ได้มีการ Demo กับทางผู้ใช้งานไปเมื่อวันที่ 22 กรกฎาคม 2568
โดยมีการเปลี่ยนแปลงดังนี้
 1. เปลี่ยนชื่อคอลัมน์จาก Invoice Date เป็น Payment Due Date
- **GISLOCK-6696 [OTC][OTE][Request OTC] ปรับการแสดง HQ/Branch และ Branch No.**
 - 1.ปรับ HQ/Branch มาเป็นแสดงเป็นชื่อแทนโค้ด
 - 2.Branch No. เป็นค่าว่างเมื่อ HQ/Branch เป็นสำนักงานใหญ่
 - 3.การแสดง Headquarter หรือ สำนักงานใหญ่อ้างอิงตาม Main Language
- **GISLOCK-6698 [FA - Other Charge/Other Expense] หน้าจอ Request Other Charge ปรับ Contact Information เพิ่มคอลัมน์ Purpose For**
เพิ่มคอลัมน์ Purpose For ในส่วนของ Contact Information
เพื่อให้ผู้ใช้สามารถระบุวัตถุประสงค์ในการติดต่อได้อย่างชัดเจนมากขึ้น

- **GISLOCK-6699 [OTC][OTE][Request OTC]** ส่วน Other Charge Transaction

- 1.Currency กรณีเป็น Thai หรือ Local default ค่าเป็น THB
- 2.Exchange Rate ดึงค่ามาจากค่ากลาง
- 3.ปรับ UI ของ Description ของ Transaction
- 4.ที่โปรแกรม Request Information :
 - สลับฟิลด์ Insured มาแสดงก่อน Type of Insurance
 - Type of Insurance ปรับมาเป็น Dropdown List
 - Brokerage Amount คำนวน Auto เมื่อเรา ระบุ Brokerage %

- **GISLOCK-7196 [All Screen] แก้ไข Icon ในปุ่ม Save Draft ให้เหมือนกัน**

ปรับไอคอนในปุ่ม Save Draft ให้มีความสอดคล้องและเหมือนกันในทุกหน้าจอ
เพื่อให้ผู้ใช้งานสามารถจดจำและใช้งานได้ง่ายขึ้น

- **GISLOCK-7390 [Date] เพิ่มเรื่องการ validate วันที่ From To**
GISLOCK-7390 [Other Charge] เพิ่มเรื่องการ validate วันที่ From To

ปรับไปดึงฟังก์ชันการตรวจสอบวันที่ From และ To จากฟังก์ชันกลาง
เพื่อให้การตรวจสอบมีความถูกต้องและสอดคล้องกับมาตรฐานที่กำหนดไว้

- **GISLOCK-7617 [Other Charge] ปรับชื่อฟิลด์ ในหน้าจอ และ DB**

ปรับชื่อฟิลด์ในหน้าจอและฐานข้อมูลให้มีความสอดคล้องและเข้าใจง่ายขึ้น
เพื่อให้ผู้ใช้งานและเข้าใจข้อมูลได้อย่างถูกต้อง หลังจากที่ได้รับข้อเสนอแนะจากผู้ใช้งาน

2.3.2 Client/Partner/Setting/General Module

2.3.2.1 Defect

- **GISLOCK-4938 [Transfer Prospect to Client][Client Amendment]** กด Edit > Update ระบบ Clear ข้อมูล Attachments

เกิดจากการที่ตอนแรกมันเซ็ต setAtm_puid_ref_file เป็น null เลยส่งค่า invalidData ไปเลยทำให้แสดงค่านั้น แก้ไขโดยการเซ็ตค่าก่อนจะส่งไปทำให้หายเจอ แล้วก็สามารถแสดงได้ปกติ

- **GISLOCK-5128 [New Prospect/Client]** กรณี Duplicate งานต่าง Department แล้วมีการแนบ Attachment ระบบไม่สามารถ Preview File ได้

เกิดจากการที่ตอนแรกไม่มีการ set department เลยทำให้การส่ง atm_can_preview ไปเป็น N ซึ่งเปิดคูมีได้ แต่พอเรา setDept ใหม่ คือเราที่เข้ามา เลย พอเรา setDept เป็นคนที่เราแนบเข้ามา มันเลยไปเช็ค dept ใน database มันเลยตรงกันทำให้ส่ง atm_can_preview มาให้หน้าบ้านเลยสามารถ preview ได้

- **GISLOCK-5157 [Partner Amendment]** เปลี่ยนชื่อ Field จาก Thai Name, English Name เป็น Partner Name

เปลี่ยนชื่อ Field จาก Thai Name, English Name เป็น Partner Name (TH) , Partner Name (EN)

- **GISLOCK-5170 [Partner Amendment][Partner Previous Details]** แก้ไขชื่อ Attachments และ Address Location

เปลี่ยนชื่อ Field จาก Address , Attachment เป็น Address Location , Attachments

- **GISLOCK-5230 [Partner Amendment]** ระบบไม่ Disable Field Payment Method Detail

เพิ่มการ Disable Field Payment Method Detail

- **GISLOCK-5231 [Partner]** ระบบไม่แสดงข้อมูล Withholding Tax Rate

งานนี้เป็นงานแก้ไขโดยการเปลี่ยนไปใช้ฟังก์ชันกลางที่ทำไว้แล้ว ลั่นเดิมเป็นเรียกฟังก์ชันเก่าเลยไม่แสดงข้อมูล Withholding Tax Rate พอเราปรับไปใช้ฟังก์ชันใหม่ก็จะแสดงมาปกติ

- **GISLOCK-5265 [Approve Client][Approve Partner]** ระบบแสดง Attached By 'ไม่ถูกต้อง (ติดการ์ด 5621)

งานนี้เป็นงานแก้ไขเงื่อนไขการเช็ค typeof partnerTypeValueList === 'object' เพิ่มไปด้วย ที่เหลือจะเป็นงานที่ไปปรับตรงของ GISLOCK-5261

- **GISLOCK-5302 [Partner Amendment]** แก้ไขการแสดงข้อมูล Contact Person ของปุ่ม Partner Previous Detail

เปลี่ยนชื่อ Field จาก Relation to Client เป็น Relation to Partner

- **GISLOCK-5699 [Setting][Program on Module] Edit** รายการแล้ว Error

มี 2 กรณี คือ

1. Save แล้ว Error เพราะมันเอาชื่อที่เราแก้ไปเช็คให้ตรง menu และมันเอาชื่อนั้นมาเช็คแล้วเจอกับชื่อในส่วนของ module แก้ไขโดยการเปลี่ยนเงื่อนไข จากเดิมใช้ชื่อ ตอนนี้ปรับมาใช้ prog_id และ folder_id มาเช็คแทน เพราะมันคนละส่วนกัน

2. เพิ่มกรณีเมื่อเรากด Edit เข้ามาแล้วกดปุ่ม Clear ก็จะนำข้อมูลที่แก้ไขไปก่อนหน้านี้กลับมาได้

- **GISLOCK-5752 [Partner][Partner Amendment] Accordion General Info** ระบุข้อมูลที่ Company Name (EN) เป็น Lockton ระบบไม่ Auto เลือกข้อมูล Inter Company (LWT Group) เป็น Yes

เพิ่มเงื่อนไขในการเช็ค Company Name (EN) ว่าเป็น Lockton หรือเปล่า ถ้าใช่ก็จะ set ค่า Inter Company (LWT Group) เป็น Yes ให้อัตโนมัติในหน้า Partner Amendment ด้วย

- **GISLOCK-5753 [Partner Amendment]** ไม่สามารถกด Sort Description ที่ Popup Copy Address From Address Location ได้

เกิดจากการที่เราส่งตัว formcontrol ไปผิด เดิมส่ง descriptionEn มาเป็น description แทน

- **GISLOCK-5761 [General Setup][Document Running] Icon Warning** ต้องเป็นเครื่องหมายตกใจ สีเหลือง
งานนี้เป็นงานแก้ไขโดยการเปลี่ยนไอค่อนของ modal แจ้งเตือนใหม่
- **GISLOCK-7013 [Parent/Group] Add Parent และไม่มีตัวเลือก Auto Create Client Group**
ระบบแสดงไม่ถูกต้อง
งานนี้เป็นงานแก้ไขหน้า UI ตรงการแสดงผล ตอนแรกมันจะแสดงภายในกรอบทำให้แสดงไม่ครบ เราปรับมาใช้ appendTo: 'body' เพื่อให้แสดงข้อมูลได้ถูกต้อง
- **GISLOCK-7052 [New Prospect/Client] ระบบไม่แสดง Effective Date ที่ Introducer Info**
เกิดจากการที่ type ของ Effective Date ตรงหน้าบ้านประการไม่ตรงกับ type ที่หลังบ้านส่งมาให้
- **GISLOCK-7056 [Document Set] เมื่อกด Add Document Set ระบบไม่สามารถกดปุ่ม Save ได้**
เกิดจากการที่เราทำ action อื่นก่อนมา add เช่น update มันจำข้อมูลเดิม ทำให้ Save ไม่ได้ แก้ไขโดยการ reset ข้อมูลก่อนหน้า ก่อนจะ add เพื่อให้ข้อมูลเป็นอันปัจจุบัน
- **GISLOCK-7096 [Prospect/Client][Contact Person] กดเลือก All Purpose ให้ระบบปิด Dropdown**
เมื่อเรากดเลือก All Purpose ให้ระบบปิด Dropdown
- **GISLOCK-7154 [Country] ระบบไม่สามารถ Search ได้ และไม่สามารถ Delete ได้**
มี 2 กรณี คือ
 1. ไม่สามารถ Search ได้ สาเหตุมาจากการที่หลังบ้านเราใช้ SQL ที่ผิด เดิมใช้ LIKE ที่ไม่มี %% ครอบทำให้ค้นหาไม่เจอ เราเลยเพิ่ม %% ครอบเข้าไป
 2. ไม่สามารถ Delete ได้ เพราะตอน get ข้อมูลมาไม่ได้เลขไอเดียที่ตรงกับตัวที่จะลบ พอเรากด delete หน้าบ้านมันเลยไม่รู้ว่าต้องลบตัวไหน แก้ไขโดยการ get ไอเดียตัวเดียว และสามารถ delete ได้ปกติ

2.3.2.2 CR

- **GISLOCK-5176 [Partner Amendment] เพิ่ม Field Start Date**
งานนี้เป็นงานปรับปรุงจากอันเดิมคือในส่วนของหน้า Partner Amendment เราจะเพิ่มฟิลด์ Start Date เข้าไป
- **GISLOCK-5769 [General Setup][Standard Code] แก้ไขหน้าจอตาม New Requirement**

งานนี้เป็นงานปรับปรุงจากอันเดิมคือในส่วนของหน้า Standard Code เราจะเพิ่มฟิลด์ Create By , Create Date , Update By , Update Date เข้าไป

- **GISLOCK-5773 [General Setup][Document Running Format] แก้ไขหน้าจอตาม New Requirement**

งานนี้เป็นงานปรับปรุงจากอันเดิมคือในส่วนของหน้า Document Running Format เราจะเพิ่มฟิลด์ Create By , Create Date , Update By , Update Date เข้าไป

- **GISLOCK-5774 [General Setup][Document Running Sequence] แก้ไขหน้าจอตาม New Requirement**

งานนี้เป็นงานปรับปรุงจากอันเดิมคือในส่วนของหน้า Document Running Sequence เราจะเพิ่มฟิลด์ Create By , Create Date , Update By , Update Date เข้าไป

- **GISLOCK-5832 [Client][Partner] ระบบไม่สามารถ Preview File กรณีเป็น View Mode ได้**

งานนี้เป็นงานปรับปรุงโดยการเพิ่มการ preview file ใน view mode เข้าไป

- **GISLOCK-7065 [LWT Bank Account] แก้ไขโปรแกรมตาม New Requirement**

ปรับจากเดิมที่เป็น Modal มาเป็นหน้า page งานนี้เป็นการทำหน้า UI ของ LWT Bank Account

- **GISLOCK-7098 [New Prospect/Client][New Partner] กรณีเลือก All Purpose ให้ Disable ตัวเลือกอื่นๆ**

งานนี้เป็นงานปรับปรุงจากอันเดิมคือในส่วนของ Contact จะมี Dropdown ที่เลือก Purpose กรณีที่เราเลือก "All Purpose" เราจะ Disable ฟิลด์อื่นแล้วก็ปิด Dropdown

- **GISLOCK-7223 [Prospect/Client] เพิ่มข้อมูลสถานะเกี่ยวกับการบันทึกข้อมูล Prospective/Client**

เพิ่ม Field ของ Entry By , Entry Date , Modification By , Modification Date ที่หน้าของ New Client , Client Amendment , Transfer Prospect to Client

- **GISLOCK-7224 [Partner] เพิ่มข้อมูลสถานะเกี่ยวกับการบันทึกข้อมูล Partner & Market Security**

เพิ่ม Field ของ Entry By , Entry Date , Modification By , Modification Date ที่หน้าของ New Partner , Partner Amendment , Market Security

2.3.3 FA-AR Module

2.3.3.1 Defect

- **GISLOCK-5808 [FA-AR][ใบนัดรับเงิน] Appointment Information กรณีเปลี่ยน Page ที่ไม่ใช่ 1 ระบบเรียงลำดับข้อมูลไม่ถูกต้อง**

เกิดจากการที่หลังบ้านไม่ได้เขียน SQL เพื่อเรียงข้อมูล เลยทำให้ตอนที่เรากดไปหน้าใหม่ข้อมูลมันเลย
เรียงไม่ถูกต้อง

- **GISLOCK-5432 [FA-AR][ใบนัดรับเงิน] Appointment No wording ผิด**
งานนี้เราแก้แค่ wording เป็น Appointment No.
- **GISLOCK-5424 [FA-AR][Appointment List] placeholder wording ผิด**
งานนี้เป็นการปรับ placeholder เป็นคำใหม่ 'Client Code/Client Group/Bill Statement No'
- **GISLOCK-5622 [FA-AR][New ใบนัดรับเงิน] ชื่อคอลัมน์ PaymentDueDate ไม่เว้นวรรค**
งานนี้เป็นงานแก้การเว้นวรรคจาก 'PaymentDue Date' เป็น 'Payment Due Date'

2.4 สวัสดิการที่ได้รับ

บริษัท G-Able ให้ความสำคัญกับเรื่องของสวัสดิการที่ดีให้กับพนักงาน โดยเฉพาะนักศึกษาสหกิจศึกษาที่เข้ามาทำงานในบริษัท ถึงแม้จะไม่ได้เป็นพนักงานประจำ แต่ก็ได้รับสวัสดิการที่ดีจากบริษัทอย่างเช่น

- ทำงาน 5 วัน / สัปดาห์ ทำงานแบบ Work from home
(09:00 - 18:00)
- โน๊ตบุ๊คสำหรับใช้ทำงานระหว่างฝึกงาน
- เป้ยเลี้ยงในการทำงาน 500 บาท / วันทำงาน (ไม่นับวันลา หรือวันหยุด)
- วันหยุดประจำปีตามประเทศไทย และวันหยุดพิเศษตามประเทศไทย

ทั้งนี้ทั้งหมดที่กล่าวมาเป็นเพียงส่วนหนึ่งของสวัสดิการที่ได้รับจากบริษัท G-Able และยังมีสวัสดิการอื่นๆ ที่ยังไม่ได้กล่าวถึง

2.5 วัฒนธรรมองค์กร

บริษัท จีเอเบิล จำกัด (มหาชน) ตระหนักรถึงความสำคัญของการมีระบบการกำกับดูแลกิจการที่ดี ซึ่งเป็นสิ่งสำคัญที่จะช่วยส่งเสริมการดำเนินงานของบริษัทให้มีประสิทธิภาพ และมีการเจริญเติบโตอย่างยั่งยืน

บทที่ 3

สรุปผลการปฏิบัติงาน

3.1 สรุปผลการศึกษา/การฝึกปฏิบัติงาน

ตามข้อตกลงในเอกสารเงื่อนไขการประเมินผลการปฏิบัติงาน (TOR) ซึ่งได้กำหนดให้ต้องดำเนินงานให้แล้วเสร็จไม่น้อยกว่า 60 งาน ในช่วงระยะเวลาของการฝึกปฏิบัติงาน ข้าพเจ้าได้ดำเนินการแล้วเสร็จจำนวน 104 งาน ทั้งนี้ รายละเอียดของแต่ละงานได้ถูกสรุปไว้ในส่วนของรายงานก่อนหน้านี้อย่างครบถ้วน

จากการดำเนินงานที่กล่าวมาข้างต้น ข้าพเจ้าสามารถบรรลุเป้าหมายของการฝึกปฏิบัติงานตามที่ได้กำหนดไว้ ซึ่งถือเป็นการบรรลุผลสำเร็จตามเกณฑ์ที่วางไว้

3.2 ประโยชน์ที่ได้รับจากการศึกษา/การฝึกปฏิบัติงาน

การฝึกปฏิบัติศักยภาพศึกษาที่บริษัท G-ABLE ในตำแหน่ง *Trainee Fullstack Developer* ตลอดระยะเวลา 6 เดือนนั้น ทำให้ผมได้รับประสบการณ์ที่มีค่ามากมาย โดยเฉพาะการทำงานร่วมกับทีมและการเรียนรู้ทักษะใหม่ ๆ ที่ไม่เคยได้ศึกษามาก่อน การได้รับโอกาสจากเพื่อนทีมและหัวหน้างานที่คอยช่วยเหลือและให้คำแนะนำเป็นสิ่งสำคัญมาก โดยเฉพาะพี่เลี้ยงของผมที่ให้ความไว้วางใจและเปิดโอกาสให้ผมได้ลองผิดลองถูกในการทำงานจริง ซึ่งช่วยพัฒนาทักษะและความมั่นใจในการทำงานอย่างมาก ทำให้ประสบการณ์ศักยภาพศึกษาครั้งนี้ประสบความสำเร็จ

ในส่วนของ *Hard Skills* ที่ได้เรียนรู้ หนึ่งในสิ่งที่สำคัญคือการได้ลงมือทำงานในตำแหน่ง *Fullstack Developer* โดยได้ใช้เทคโนโลยี Angular, Java Spring Boot และ SQLServer ซึ่งเป็นความรู้ที่ไม่ได้เรียนในห้องเรียน การได้เห็นและทดลองใช้เทคโนโลยีเหล่านี้กับระบบจริงของระบบประกันภัย ซึ่งเป็นระบบที่มีความซับซ้อนและสำคัญมาก ทำให้ผมเข้าใจถึงการออกแบบและจัดการระบบที่ซับซ้อนมากขึ้น พี่เลี้ยงของผมได้มอบหมายงานเกี่ยวกับ *Fullstack Development* ให้ผมรับผิดชอบส่วนใหญ่ ทำให้ผมได้เรียนรู้การทำงานทั้ง Frontend, Backend และ Database อย่างลึกซึ้ง และได้นำความรู้นี้ไปปรับใช้ใน โครงการ ของผม ทั้งในด้านการออกแบบสถาปัตยกรรมระบบ (*System Architecture*) และการประยุกต์ใช้เทคโนโลยีให้เหมาะสมกับ-project

นอกจากนี้ *Soft Skills* ก็เป็นสิ่งที่ผมได้พัฒนาระหว่างการทำงานในศักยภาพศึกษา บริษัท G-ABLE มีกิจกรรมให้เข้าร่วมเป็นประจำเพื่อส่งเสริม *Soft Skills* เช่น การทำงานร่วมกับคนต่าง *Generation*, ทักษะการสื่อสาร, การเข้าสังคม และการสร้างโปรดไฟล์สำหรับการสมัครงานในอนาคต กิจกรรมเหล่านี้ช่วยให้ผมพัฒนาความสามารถในการทำงานร่วมกับผู้อื่นและสร้างเครือข่ายที่ดีสำหรับอนาคต

3.3 ข้อเสนอจากบริษัท

ขณะนี้ทางบริษัท ยังไม่มีตำแหน่งพนักงานประจำสำหรับตำแหน่ง *Fullstack Developer* แต่บริษัทได้เสนอให้เข้าทำงานในรูปแบบ สัญญาจ้างรายปี แทน โดยมีเงินเดือนเริ่มต้นที่ 30,000 บาทต่อเดือน และสามารถปรับตามประสบการณ์และความสามารถของผู้สมัคร

បរទេសាន្តកម្ម

- [1] About g-able. G-Able Public Company Limited. [Online]. Available: <https://www.g-able.com/th/about/g-able/>
- [2] G-able's solutions. G-Able Public Company Limited. [Online]. Available: <https://www.g-able.com/th/solution/cybersecurity>

ภาคผนวก

ภาคผนวก ก
เอกสารสำคัญที่เกี่ยวข้อง

• เอกสารสำคัญของคณะ

- หนังสือยินยอมให้เผยแพร่รายงานปฏิบัติงานสหกิจศึกษา (หน้า 30)
- วศ.สก.-03/04 รายงานตัวเข้าสหกิจศึกษา (หน้า 31)
- วศ.สก-06 แบบแจ้งโครงร่างรายงานการปฏิบัติงานสหกิจศึกษา (หน้า 34)
- วศ.สก-10 แบบสรุปการปฏิบัติงานสหกิจศึกษาประจำสัปดาห์ (หน้า 44)
- วศ.สก.-11 แบบสรุปผลการปฏิบัติงานสหกิจศึกษา (หน้า 45)

• เอกสารสำคัญที่เกี่ยวข้องกับงาน

- เอกสารสรุปโปรแกรมที่สร้าง (หน้า 51)
- เอกสารสรุปสิ่งที่ได้เรียนรู้และลองทำ Angular (หน้า 62)
- เอกสารสรุปสิ่งที่ได้เรียนรู้และลองทำ Java Spring Boot (หน้า 68)
- เอกสารสรุปสิ่งที่ได้เรียนรู้และลองทำ SQL Server (หน้า 73)



ศูนย์ Entaneer Academy

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

หนังสือยินยอมให้เผยแพร่รายงานปฏิบัติงานสหกิจศึกษา

เพื่อเป็นการส่งเสริมการพัฒนาการศึกษาของประเทศไทย ข้าพเจ้าในฐานะตัวแทนหน่วยงานหรือ
บริษัท**G-A BLE**..... มีความยินดีให้คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเชียงใหม่ เผยแพร่เนื้อหาในรายงานสรุปผลโครงการของนักศึกษา ภาคการศึกษาที่**1**.....ประจำปี
การศึกษา**2568**..... ณ สถานประกอบการของข้าพเจ้าในส่วนของ “กิจกรรมที่นักศึกษาสหกิจศึกษาทำ
รายงาน/โครงการ” “บหคดีย่อ” และ “ข้อเสนอแนะในรายงาน/โครงการ” โดย

- อนุญาตให้ระบุชื่อหน่วยงานหรือบริษัท
- ไม่อนุญาต
- อื่นๆ โปรดระบุ.....

ลงชื่อ ดร. วันรัตน์ วงศ์สวัสดิ์ ผู้มีอำนาจกระทำการแทน
(ดร. วันรัตน์ วงศ์สวัสดิ์) นิติบุคคล/ผู้ประกอบการ
วันที่ 23 / ๐๗ / ๒๕๖๘

ประทับตรา บริษัท / โรงงาน (ถ้ามี)

ศูนย์ Entaneer Academy (สหกิจศึกษาและฝึกงาน)

239 ต.สุเทพ อ.เมือง จ.เชียงใหม่ 50200 โทรศัพท์ (053)944179 ต่อ 108 โทรสาร (053)944113 E-mail:interncoop@eng.cmu.ac.th

ใบรายงานตัวเข้ารับการฝึกศักยิจ

[2342]

ชื่อนักศึกษา สิปกร คำเมสว่าง

รหัสนักศึกษา 650610813

นักศึกษาสาขาวิชาศิวกรรมศาสตร์ คอมพิวเตอร์

คณะศิวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

เบอร์โทรศัพท์นักศึกษา 0955093818

เบอร์โทรศัพท์ผู้ปกครอง (กรณีฉุกเฉิน) 0957716532

ได้รายงานตัวเพื่อเข้ารับการฝึกศักยิจ ณ G-ABLE PUBLIC COMPANY LIMITED

ที่อยู่ 127/30 Panjathani Tower ถ.Nonsi Road แขวงช่องนนทรี เขตดอนเมือง กรุงเทพมหานคร 10120

ผู้ควบคุมการฝึกศักยิจ Mr. Chaiya Saengchan ตำแหน่ง Application Developer

โทรศัพท์ 0644276042 โทรสาร -

พี่เลี้ยง Mr. Chaiya Saengchan โทรศัพท์ 0644276042 อีเมล chaiya.s@g-able.com

ผู้ประสานงานการนิเทศ Khanison Kwaosibsam โทรศัพท์ 065-719-4168 อีเมล khanison.k@g-able.com

ลักษณะงานที่นักศึกษารับผิดชอบ พัฒนาและแก้ไขโปรแกรม: - พัฒนาโปรแกรมตาม Requirement ให้ถูกต้องและมีประสิทธิ

ภาพ - ใช้ Angular (Front-end), Java Spring Boot (Back-end), SQL Server (Database) -

ทำงานตาม Agile Methodology - วิเคราะห์ปัญหาซึ่งลึก ป้องกันปัญหาล่วงหน้าอย่างมีเหตุ

ผล

ระยะเวลาการฝึกศักยิจ 16 เม.ย. 2568 ถึง 15 ต.ค. 2568 เวลาเข้าฝึกศักยิจ 09:00 น. เวลาออกฝึกศักยิจ 18:00 น.

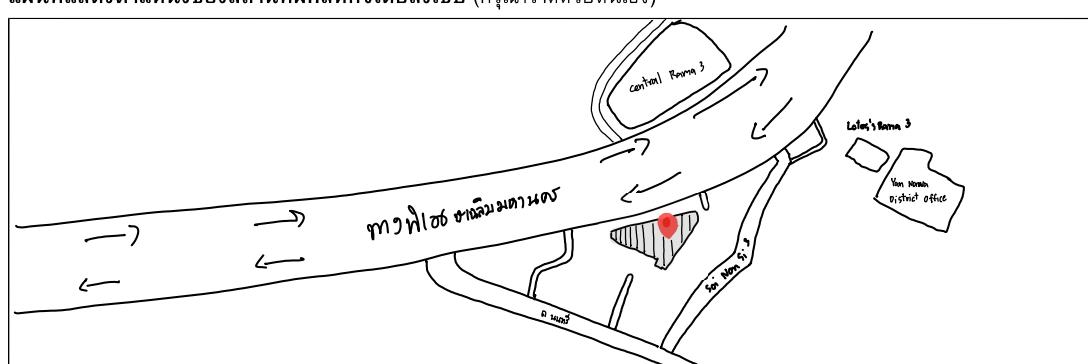
วันหยุดบริษัท ทุกวันเสาร์ ทุกวันอาทิตย์

วันหยุดอื่น ๆ -

รวมเป็นเวลา 131 วัน (นับเฉพาะวันที่เข้ารับการฝึกศักยิจ)

ที่พักนักศึกษาขณะเข้ารับการฝึกศักยิจ 239 ตรอกหอพักชาย 5 ถ.ห้วยแก้ว ต.สุเทพ อ.เมือง จ. เชียงใหม่ 50200

แผนที่แสดงตำแหน่งของสถานที่ฝึกศักยิจโดยสังเขป (กรุณาวาดด้วยตนเอง)



ในการฝึกศักยิจครั้งนี้ ข้าพเจ้าจะปฏิบัติตามกฎระเบียบและข้อบังคับ ตลอดจนรับผิดชอบงานที่ได้รับมอบหมายจากทางสถานที่รับฝึกศักยิจดังกล่าวข้างต้นอย่างเต็มความสามารถ และในการฝึกศักยิจจะไม่นำความลับ ข้อมูล รวมทั้งรูปถ่ายและคลิปวิดีโอ ภายในบริษัทออกมายกเว้นแต่ก่อนได้รับการอนุญาตจากบริษัท

ลงชื่อ..... สิปกร คำเมสว่าง

ลงชื่อ..... ชาญ แสงจันทร์

(สิปกร คำเมสว่าง)

(Mr. Chaiya Saengchan)

นักศึกษาผู้เข้ารับการฝึกศักยิจ

ผู้ควบคุมการฝึกศักยิจ

วันที่ 16 / พฤษภาคม / 2568

วันที่ 16 / พฤษภาคม / 2568

ประทับตรา บริษัทโรงงาน (ถ้ามี)

สแกนเอกสารและแนบไฟล์ที่มีข้อมูลและลายเซ็นของผู้ควบคุมการฝึกศักยิจในเว็บไซต์ <https://intern.eng.cmu.ac.th> ภายในสัปดาห์แรกของการเข้าฝึกศักยิจ

แบบแจ้งโครงการร่างรายงานการปฏิบัติงานสหกิจศึกษา
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

วิชา สอนภาษา รหัสวิชา 261495
ประจำภาคการศึกษาที่ 1 ปีการศึกษา 2564

คำชี้แจง

รายงานนี้ถือเป็นส่วนหนึ่งของการปฏิบัติงานสหกิจศึกษา มีวัตถุประสงค์เพื่อฝึกฝนทักษะการสื่อสาร (Communication Skill) ของนักศึกษา และจัดทำข้อมูลที่เป็นประโยชน์ของสถานประกอบการ นักศึกษาจะต้องขอรับคำปรึกษาจากพนักงานที่ปรึกษา (Job Supervisor) เพื่อกำหนดหัวข้อรายงานที่เหมาะสม โดยคำนึงถึงความต้องการของสถานประกอบการเป็นหลัก ตัวอย่างของรายงานได้แก่ ผลงานวิจัยที่นักศึกษาปฏิบัติ รายงานวิชาการที่นำเสนอ การสรุปข้อมูลหรือสถิติบางประการ การวิเคราะห์ และประเมินผลข้อมูล เป็นต้น ทั้งนี้ รายงานอาจจะจัดทำเป็นส่วนของนักศึกษาสหกิจศึกษามากกว่า 1 คนได้

ในการนี้ที่สถานประกอบการมิได้มีข้อกำหนดกลุ่มนักศึกษาให้ทำรายงานในหัวข้อดังนี้ นักศึกษาจะต้องพิจารณาเรื่องที่ตนสนใจและหันไปทำการสำรวจ โดยปรึกษากับพนักงานที่ปรึกษาเสียก่อน ตัวอย่างหัวข้อที่จะใช้เขียนรายงาน ได้แก่ รายงานวิชาการที่นักศึกษาสนใจ รายงานการปฏิบัติงานที่ได้รับมอบหมาย หรือแผนและวิธีการปฏิบัติงานที่จะทำให้บรรลุถึงวัตถุประสงค์ของการเรียนรู้ที่นักศึกษาวางแผนเป้าหมายไว้จากการปฏิบัติงานสหกิจศึกษาครั้งนี้ (Learning Objectives) เมื่อกำหนดหัวข้อได้แล้ว ให้นักศึกษาจัดทำโครงร่างของเนื้อหารายงานพ้องสังเขป ตามแบบฟอร์มที่กำหนด ทั้งนี้ ให้ปรึกษาภัณฑ์งานที่ปรึกษาเสียก่อนแล้วจึงส่งกลับมาจังหวะ **ภายในเดือน มกราคม** ของแต่ละปี

ชื่อ – นามสกุล (นักศึกษา) นาย วิมลกร คำสุรุ่ง รหัสประจำตัว 650610413
สาขาวิชา เทคโนโลยี คณะ วิศวกรรมศาสตร์ ภาคปกติ ภาค.....
ปฏิบัติงานสหกิจศึกษาน (ชื่อสถานประกอบการ) บริษัท จีโนเทค จำกัด (มหาชน)
เลขที่ 127/30 ถนน หมู่ที่ ตำบล/แขวง ห้องน้ำ
อำเภอ/เขต ขอนแก่น จังหวัด กรุงเทพมหานคร รหัสไปรษณีย์ 10110
โทรศัพท์ 02 741 9333 โทรสาร -

ขอแจ้งรายละเอียดเกี่ยวกับโครงร่างรายงานการปฏิบัติงานสหกิจศึกษา ดังนี้

1. หัวข้อรายงาน (Report Title) (อาจจะขอเปลี่ยนแปลงหรือแก้ไขเพิ่มเติมได้ในภายหลัง)
ภาษาไทย <u>รายงานปฏิบัติงานสหกิจศึกษา ตำแหน่ง Fullstack Developer</u>
ภาษาอังกฤษ <u>Cooperative Education Report : Fullstack Developer</u>

2. รายละเอียดเนื้อหาของรายงาน (อาจจะขอเปลี่ยนแปลงหรือแก้ไขเพิ่มเติมได้ในภายหลัง)
2.1 (รายละเอียดบุคคล) เหตุผลและความเป็นมาของการศึกษา
<u>ประสงค์ให้ได้รับความรู้ทั่วไปในด้านเทคโนโลยีและสามารถนำไปใช้ในการทำงานจริงๆ ได้ เช่น การเขียนโปรแกรมภาษา C และภาษา Python รวมถึงการเขียนโปรแกรมเว็บไซต์ด้วยภาษา HTML, CSS และ JavaScript ซึ่งจะช่วยให้สามารถทำงานในอาชีวศึกษาได้ดีขึ้น</u>

2.2 วัตถุประสงค์ของการศึกษา

1. ពិនិត្យរបាយការណ៍ នូវការបង្កើតអនុវត្តន៍យុទ្ធសាស្ត្រ ដើម្បីស្វែងរកចំណាំរបស់ក្រុមហ៊ុន
 2. ពិនិត្យរបាយការណ៍របស់ក្រុមហ៊ុន ជាអ្នករូបការ Fullstack Developer និងនៅ Frontend នាម Backend
 3. ពិនិត្យរបាយការណ៍របស់ក្រុមហ៊ុន និងនៅក្នុងក្រុមហ៊ុន ដើម្បីស្វែងរកចំណាំរបស់ក្រុមហ៊ុន
 4. ពិនិត្យរបាយការណ៍របស់ក្រុមហ៊ុន និងនៅក្នុងក្រុមហ៊ុន ដើម្បីស្វែងរកចំណាំរបស់ក្រុមហ៊ុន

2.3 ขอบเขตของการศึกษา

მართვის და მოწყვეტილებების განვითარება Fullstack Developer

2.4 ผลที่คาดว่าจะได้รับ

3. ขั้นตอนการดำเนินงาน (Methodology) (อาจขอเปลี่ยนแปลงหรือแก้ไขเพิ่มเติมได้ในภายหลัง)

ગુજરાત યૂનિવર્સિટી | BA (Business Analytics) | MA (Quality Assurance)

- ցանցը BA ինչպես վրայութեան համար Requirement կամ
 - ցանցը BA ինչպես պահանջ և դաշտ առաջարկ կամ համար

၁၃၁ နေဂတ်ပြုမှုနှင့် အာရာရှင်များ နေဂတ်ပြုမှုနှင့် အာရာရှင်များ နေဂတ်ပြုမှုနှင့် အာရာရှင်များ

4. TOR (Terms of Reference)

4.1 ນັກຂາ ໂດຍຮ່ວມປັບປຸດ

- ໃຫ້ໄດ້ໃຫ້ມີຄວາມສູງຂອງການສ່ວນຕົວທີ່ສຳເນົາຢູ່ແປປພັດທະນຸ້ນ Front nor Back ມັນຫຼື່ນັ້ນການວຽກງານ

ແນວວຽກຕົວແນວ (reference code) ພັນຍະນຸ້ນການງານ Jira

4.2 ອະນຸຍາວາ

16 ລາວມາດ ຖ. 2562 – 15 ລາວມາດ 2568

4.3 ແນ້ນທີ່ນັ້ນມີຄວາມເລັກ

- ໂຄສະໜາກ Fullstack ດ້ວຍຕົວຢ່າງ Jira

- ຕັ້ງປົກຕະຕາມການຝຶກ ທີ່ມີການຕົ້ນຕົ້ນ

- ມີຄວາມອານຸຍາຍ ດັ່ງນີ້ ເຊັ່ນ ຫຼື່ນການທີ່ມີຄວາມອານຸຍາຍ

- ມີມີຮຽນ weekly update ຖຸກຕົ້ນຕົ້ນ

4.4 ໄກສອນການປັບປຸດຕິຈິນພາ

- ສິ່ງອານ Task ຖື່ນຕົ້ນຕົ້ນ ອັນດີ ແລ້ວ Task ເວັບຕົ້ນຕົ້ນ

- ຮັບອານວາງໃນວັນທີ 2022 Release ຢົດສອນວາງຈາກ ແລ້ວ ດັ່ງກ່າວກ່າວກ່າວ

ລັງຊື່ອ ສັນຕະກິດ ສັນຕະກິດ
(ນາງ ສິນຕະກິດ ສັນຕະກິດ)
ນັກສຶກຂາສະກິດສຶກຂາ

ວັນທີ 1 ທຸກາຄນ 2568

ລັງຊື່ອ ນາງ ແສງຈົນທົງ ແສງຈົນທົງ
(ນາງ ປິບ ແສງຈົນທົງ)
ພັນການທີ່ປັບປຸດ
ຕຳແໜ່ງ System Analyst
ວັນທີ 2 ທຸກາຄນ 2568



งานบริการการศึกษาและพัฒนาคุณภาพนักศึกษา
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

แบบสรุปการปฏิบัติงานสหกิจศึกษาประจำปี

ชื่อ-สกุล (นาย/นางสาว) รัตน์กร ศิริร่วง รหัสนักศึกษา 650610413

นักศึกษาภาควิชาชีวกรรม คณิตศาสตร์

ชื่อสถานที่ฝึกสหกิจศึกษา G-ABLE PUBLIC COMPANY LIMITED

ชื่อ-สกุลพนักงานที่ปรึกษา นายไชยา แสงจันทร์

ระยะเวลาฝึกสหกิจศึกษา ระหว่างวันที่ 16 พฤษภาคม 2564 ถึง 15 ตุลาคม 2564

วัน เดือน ปี	สัปดาห์ที่ 1	
รายละเอียด	นักศึกษา <u>รัตน์กร ศิริร่วง</u>	พนักงานที่ปรึกษา <u>ไชยา แสงจันทร์</u>
วัน เดือน ปี		สัปดาห์ที่ 2
รายละเอียด		
วัน เดือน ปี	นักศึกษา <u>รัตน์กร ศิริร่วง</u>	พนักงานที่ปรึกษา <u>ไชยา แสงจันทร์</u>

วัน เดือน ปี	สัปดาห์ที่ 3		
๒๕/๐๔/๖๘ -๐๑/๐๕/๖๘	- ไม่ใช้งาน defect (ไม่มีไฟบัด) หากสนใจ ว่าจะซื้อ เนื่อง เนื่องจากในรูป - ประชุม ประจำ รุ่น ๓		
ลายมือชื่อ	นักศึกษา ภัณฑ์ ภานุ วงศ์ ผู้สอน ดร. มงคล พันธุ์		
วัน เดือน ปี	สัปดาห์ที่ 4		
๐๕/๐๕/๖๘ -๐๙/๐๕/๖๘	- ไม่ใช้งาน defect (ไม่มีไฟบัด) หากสนใจ ว่าจะซื้อ เนื่อง เนื่องจากในรูป - ประชุม ประจำ รุ่น ๓		
ลายมือชื่อ	นักศึกษา ภัณฑ์ ภานุ วงศ์ ผู้สอน ดร. มงคล พันธุ์		
วัน เดือน ปี	สัปดาห์ที่ 5		
๑๒/๐๕/๖๘ -๑๖/๐๕/๖๘	- ซ่อมต่อ ทบทวนงาน ที่ว่าง กรณี เนื่องจาก งาน เอกงาน ก็ได้ QA (Branch ฐานข้อมูล)	- ซ่อมต่อ defect ที่ยังคงอยู่	- ประชุม ประจำ รุ่น ๓
ลายมือชื่อ	นักศึกษา ภัณฑ์ ภานุ วงศ์ ผู้สอน ดร. มงคล พันธุ์		

วัน เดือน ปี	สัปดาห์ที่ 6
19/09/69 -	- ซ้อมทักษะ ภาษาอังกฤษ ในการเขียน รายงานงาน สำหรับ QA (Branch โขเข็งทองคำ) - ตรวจสอบ defect ที่มากที่สุด - ประชุม ประจำสัปดาห์
ลายมือชื่อ	นักศึกษา รุ่งปอง ลักษณ์วงศ์ พนักงานที่ปรึกษา นรุสุล แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 7
26/09/69 -	- ซ้อมทักษะที่ได้รับ defect ในงาน CR (งานประเมินปัญหาต้องแก้ไขใน requirement) - ประชุม ประจำสัปดาห์
ลายมือชื่อ	นักศึกษา รุ่งปอง ลักษณ์วงศ์ พนักงานที่ปรึกษา นรุสุล แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 8
03/10/69 -	- ฝึกฝนทักษะ Module Client / Partner ระหว่าง FA-PR - แก้ defect ของ Module ที่มี - ประชุม ประจำสัปดาห์
ลายมือชื่อ	นักศึกษา รุ่งปอง ลักษณ์วงศ์ พนักงานที่ปรึกษา นรุสุล แสงจันทร์

วัน เดือน ปี	สัปดาห์ที่ 9	
09/10/64	- นำเอกสารมา SIT (brand ขอกำ)	
-	- แก้ defect กอง Module นี้	
13/10/64	- ประชุมนักวิชาชีพตาม	
รายมือชื่อ	นักศึกษา รุ่นปี ภาคเรียน	พนักงานที่ปรึกษา ผู้ร่วมกิจกรรม
วัน เดือน ปี	สัปดาห์ที่ 10	
16/10/64	- นำเอกสารมา SIT (brand ขอกำ)	
-	- ถ่ายรูป Module Other Charge	
20/10/64	- ประชุมนักวิชาชีพตาม	
รายมือชื่อ	นักศึกษา รุ่นปี ภาคเรียน	พนักงานที่ปรึกษา ผู้ร่วมกิจกรรม
วัน เดือน ปี	สัปดาห์ที่ 11	
27/10/64	- add Other charge Request page	
-	- add Other charge request list page	
27/10/64	- ประชุมนักวิชาชีพตาม	
รายมือชื่อ	นักศึกษา รุ่นปี ภาคเรียน	พนักงานที่ปรึกษา ผู้ร่วมกิจกรรม

วัน เดือน ปี	สัปดาห์ที่ 12
๒๘ ๐๖ ๖๔ - ๐๔ ๐๗ ๖๔	<ul style="list-style-type: none"> - add Installment - edit Request Info - ประชุม ประจำ 月 ตุน
ลายมือชื่อ	นักศึกษา รุ่งปอง ภานุช่อ ผู้รับผิดชอบ
วัน เดือน ปี	สัปดาห์ที่ 13
๐๗ ๐๗ ๖๔ - ๑๑ ๐๗ ๖๔	<ul style="list-style-type: none"> - Edit share Department - Add other charg Approve - ประชุม ประจำ 月 ตุน
ลายมือชื่อ	นักศึกษา รุ่งปอง ภานุช่อ ผู้รับผิดชอบ
วัน เดือน ปี	สัปดาห์ที่ 14
๑๔ ๐๗ ๖๔ - ๑๖ ๐๗ ๖๔	<ul style="list-style-type: none"> - Edit search Bar in OTC List - Add Invoice Date In OTC List - Add Model Installment and share Department - ประชุม ประจำ 月 ตุน
ลายมือชื่อ	นักศึกษา รุ่งปอง ภานุช่อ ผู้รับผิดชอบ

วัน เดือน ปี	สัปดาห์ที่ 15
21 107164 -	- add Back button in view mode - Update request Info - ฝรั่งเศสภาษาไทย
วัน เดือน ปี	สัปดาห์ที่ 16
24 107164 -	- Add set up Transaction Code page - Add set up Transaction Code List - ฝรั่งเศสภาษาไทย
วัน เดือน ปี	สัปดาห์ที่ 17
04 109164 -	- Update Invoice section - Add payment due day - Edit currency and Exchange rate In Transaction - ฝรั่งเศสภาษาไทย
ลายมือชื่อ	นักศึกษา รุ่งปอง คำนึงด้วง พนักงานที่ปรึกษา นฤทธิ์ แสงจันทร์

วัน เดือน ปี	สัปดาห์ที่ 18	
11/04/64 -19/04/64	<ul style="list-style-type: none"> - add Auto Update In Installment and share Department - Edit submit and save draft in view mode - ฝึกหัดปรับตั้งค่า 	
ลายมือชื่อ	นักศึกษา รุ่งนภา ภานุวงศ์	พนักงานที่ปรึกษา น.ส. แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 19	
16/04/64 -22/04/64	<ul style="list-style-type: none"> - Edit auto attr, automit and save draft - Edit LNT bank account page - ฝึกหัดปรับตั้งค่า 	
ลายมือชื่อ	นักศึกษา รุ่งนภา ภานุวงศ์	พนักงานที่ปรึกษา น.ส. แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 20	
23/04/64 -29/04/64	<ul style="list-style-type: none"> - Edit error message for installment and share Department - Edit scroll in form - Edit search and delete in Country page - ฝึกหัดปรับตั้งค่า 	
ลายมือชื่อ	นักศึกษา รุ่งนภา ภานุวงศ์	พนักงานที่ปรึกษา น.ส. แสงจันทร์

วัน เดือน ปี	สัปดาห์ที่ 21
01/09/64 -	- add share introduces and relate OTE - Update Amount calculation - Edit Refresh Installment - ฝึกอบรมผู้รับผิดชอบ
ลายมือชื่อ	นักศึกษา รุ่งปอง คำสุขวงศ์ พนักงานที่ปรึกษา บุญเรือง แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 22
05/09/64 -	- Disable VAT Amount when VAT% = 0 - Edit currency - add submit form in Client /pastinny - ฝึกอบรมผู้รับผิดชอบ
ลายมือชื่อ	นักศึกษา รุ่งปอง คำสุขวงศ์ พนักงานที่ปรึกษา บุญเรือง แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 23
19/09/64 -	- Edit search In OTC List - Edit Dropdown - add Line No. In transactions - ฝึกอบรมผู้รับผิดชอบ
ลายมือชื่อ	นักศึกษา รุ่งปอง คำสุขวงศ์ พนักงานที่ปรึกษา บุญเรือง แสงจันทร์

วัน เดือน ปี	สัปดาห์ที่ 24
22/10/64 -	- Edit Condition Search in OTC List - Edit Set up transcode
26/10/64	- change getparams to state - ฝึกอบรมฟังรับฟังใจ
ลายมือชื่อ	นักศึกษา รุ่งปอง ภานุวงศ์ พนักงานที่ปรึกษา วิชัย แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 25
27/10/64 -	- validate date form , date to - rename database and display - Edit path - ฝึกอบรมฟังรับฟังใจ
ลายมือชื่อ	นักศึกษา รุ่งปอง ภานุวงศ์ พนักงานที่ปรึกษา วิชัย แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ 26
03/11/64 -	- Add FF Booking - update share Introducer and relate Other Expense - ฝึกอบรมฟังรับฟังใจ
ลายมือชื่อ	นักศึกษา รุ่งปอง ภานุวงศ์ พนักงานที่ปรึกษา วิชัย แสงจันทร์

วัน เดือน ปี	สัปดาห์ที่ ๒๗	
๑๓/๑๐/๖๖ - ๑๕/๑๐/๖๖	<ul style="list-style-type: none"> - Add Approve OTC Request in Approval List - ฝึกซ้อมภาษาไทยโดย 	
ลายมือชื่อ	นักศึกษา กมลกร ภิรมย์สุขวงศ์	พนักงานที่ปรึกษา วิชัย แสงจันทร์
วัน เดือน ปี	สัปดาห์ที่ ๒๘	
ลายมือชื่อ	นักศึกษา	พนักงานที่ปรึกษา
วัน เดือน ปี	สัปดาห์ที่ ๒๙	
ลายมือชื่อ	นักศึกษา	พนักงานที่ปรึกษา



งานบริการการศึกษาและพัฒนาคุณภาพนักศึกษา
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

แบบสรุปผลการปฏิบัติงานสหกิจศึกษา

ชื่อ-สกุล (นาย/นางสาว) นาย สมปกร คำสี่รุ่ง รหัสนักศึกษา 770610413
ภาควิชา/วิศวกรรม ระบบฯ เทคฯ ชื่อ-สกุล พนักงานที่ปรึกษา นาย เชีย แกล้งกุญช์
ตำแหน่ง System Analyst โทรศัพท์ 0854726001 E-Mail: chaia.s@gable.com
ชื่อสถานที่ฝึกสหกิจศึกษา G-ABLE PUBLIC COMPANY LIMITED
ระยะเวลาการฝึกสหกิจศึกษา ระหว่างวันที่ 16 เมษายน 2563 ถึง 15 พฤษภาคม 2563

ขอสรุปผลการปฏิบัติงานตลอดระยะเวลาการฝึกสหกิจศึกษาของข้าพเจ้า ดังนี้

- | | | | |
|--------------|-------|----------|-------------------------------|
| 1. วันลาป่วย | จำนวน | <u>0</u> | วัน (แนบใบลาที่ได้รับอนุมัติ) |
| 2. วันลาภิจ | จำนวน | <u>0</u> | วัน (แนบใบลาที่ได้รับอนุมัติ) |
| 3. วันขาดงาน | จำนวน | <u>0</u> | วัน |

คงเหลือจำนวนวันเข้าฝึกสหกิจศึกษา 131 วัน

ลงชื่อ


(นาย เชีย แกล้งกุญช์)

พนักงานที่ปรึกษาสหกิจศึกษา
17 / ก.พ. / 2568

ประทับตรา บริษัท/โรงงาน (ถ้ามี)

ลงชื่อ สมปกร คำสี่รุ่ง
(นาย สมปกร คำสี่รุ่ง)

นักศึกษาผู้เข้ารับการฝึกสหกิจศึกษา
15 / พ.ค. / 2568

โปรแกรมที่ได้ทำ

Other Charge Request

The screenshot displays the 'Other Charge Request' application interface. The main window is titled 'Other Charge Request' and includes the following sections:

- Request Information:** Contains fields for Request No., Charge to Code, Charge to Name (TH), Charge to Name (EN), ID No., Nationality (Thai or Foreigner), Invoice Name, Payment Due Date, Period Start, Period End, Brokerage Amount, and Remark.
- Billing Information:** Includes fields for Billing Name, Email Address, Billing Address, Mobile Number, Office Phone, Ext., Home Phone, and Fax.
- Other Charge Details:** Shows a table for 'Other Charge Transaction' with columns: Action, Line No., Transaction, Description, Currency, Exchange Rate, Amount, VAT Calculation, VAT Type, VAT Rate %, VAT Amount, Total Amount, and VAT Rate %.
- Request Information (Red Circle 1):** A detailed view of the request information section, showing a table with columns: Action, Template, Item, Quantity, and Amount.
- Attachments (Red Circle 2):** A section for attaching files, showing a table with columns: Action, Document Type, Expiry Date, Description, Attached Files, Attached By, and Attached Date.
- Share Department:** A section for sharing income, showing a table with columns: Action, Department, Department Name, Share (%), Share Income Amount, and Share Income Amount.

หน้าสำหรับใช้เก็บการสร้างใบคำขอเพื่อเรียกค่าใช้จ่ายเพิ่มเติมนอกเหนือจากเบี้ยประกัน มีการระบุว่าจะเก็บที่ไหน ที่อยู่ที่ไหน ค่าใช้จ่ายนั้นมี Görบัง การผ่อนชำระ การแนบไฟล์ และ การเรียกเก็บจากแผนกไหน

Other Charge Request List

The screenshot shows a web-based application interface for managing other charge requests. At the top, there are search filters for 'Charge to Code', 'Charge to Name', and 'Charge to Abbreviation', along with fields for 'Request No.', 'Request Date From', 'Request Date To', and 'Status'. Below the filters is a 'Standard Search' button and a 'Search' button. The main area displays a grid titled 'Other Charge List' with columns: Action, Request No., Charge to Code, Charge to Name, Charge to Abbreviation, Amount, Entry By, Entry Date, and Submit. There are 8 items listed in the grid. At the bottom, there are buttons for 'Page Size' (set to 10) and 'Page' (set to 1/1).

ເປັນໜ້າທີ່ເອົາໄວ້ດູ/ແກ້ໄຂ/ລບ/ເຮືອກກຳລັບ ຮາຍການ Other Charge Request ທີ່ເຮົາສ້າງໄວ້

Other Charge Approve List

The screenshot shows a web-based application interface for approving other charge requests. At the top, there are search filters for 'Charge to Code/Charge To Name (ID)/Charge To Name (EN)/Request No.', along with a 'Search' and 'Clear' button. The main area displays a grid titled 'Other Charge Request Approve' with columns: Action, Request No., Charge to Code, Charge to Name, Amount, Entry By, Entry Date, Submit Date, Invoice Date, and Status. There are 6 items listed in the grid. At the bottom, there are buttons for 'Page Size' (set to 10) and 'Page' (set to 1/1).

ເປັນໜ້າທີ່ເອົາໄວ້ດູ/ແກ້ໄຂ/ ຮາຍການ ທີ່ຖຸກ Submit ມາຈາກ Other Charge Request

Other Charge Approve

The screenshot displays the 'Other Charge Request' form. Key sections include:

- Request Details:** Fields for Request No., Charge to Code, Charge to Name (TH), Charge to Name (EN), Location (Local/Overseas), Tax ID, HQ/Branch, Branch No., Invoice Name, Payment Due Date, and Invoice Address.
- Billing Information:** Fields for Billing Name, Email Address, Mobile Number, Office Phone, Ext., Home Phone, and Billing Address.
- Other Charge Transaction:** A table showing currency, exchange rate, amount, VAT calculation (VAT Type, VAT Rate %, WHT Rate %, WHT Amount, Net Amount), and total amounts.
- Request Information:** A table listing items with columns for Action, Line No., Transaction, Description, Currency, Exchange Rate, Amount, VAT Calculation, VAT Type, VAT Rate %, VAT Amount, WHT Rate %, and WHT Amount.
- Attachments:** A section for document type, expiry date, and description, with an upload file button.
- Share Department:** A table showing department, share percentage, share income amount, and department name.
- Footer:** Fields for Entry By (G_Dararat), Entry Date (21/08/2025), Modification By (G_Dararat), Modification Date (15/09/2025), Status (Wait for BU Approve), and buttons for Return, Report, and Approve.

ເປັນຫຼາກທີ່ຈະຮັບການອນນຸມດີຈາກທົວໜ້າພໍ່ອຈະເປັນທາງຜ່ານໄປສູ່ກະບຽນການຮອງແພນກບັນຊີຕ່ອໄປ

Set up Transaction Code List

The screenshot shows the 'Set up Transaction Code List' page. At the top, there is a search bar with placeholder text 'Transaction Code/Transaction Type/transaction Description'. Below the search bar is a toolbar with a magnifying glass icon for 'Search' and a circular icon for 'Clear'. A blue header bar labeled 'Set up Transaction Code List' spans across the page. Below the header is a table with columns: Action, Transaction Code T1, Transaction Type T1, Description T1, and WHT Rate T1. The table contains 14 items, each represented by a small icon and a row number. At the bottom of the table, there are buttons for 'Page Size' (set to 10), '(14 items)', and navigation arrows. The status bar at the bottom right shows 'Page: 1 / 2'.

ເປັນໜ້າທີ່ເອົາໄວ້ແກ້ໄຂ/ລົບ ລາຍກາຮ Transaction Code ທີ່ເຮົາສ້າງໄວ້

Set up Transaction Code

The screenshot shows the 'Set up Transaction Code' page. At the top, there is a search bar with placeholder text 'Transaction Type' and a dropdown menu. Below the search bar is a toolbar with a magnifying glass icon for 'Search' and a circular icon for 'Clear'. A blue header bar labeled 'Set up Transaction Code' spans across the page. The main area is divided into two sections: 'Transaction Code' and 'Account Information'. The 'Transaction Code' section contains fields for 'Transaction Type' (dropdown), 'Transaction Code' (text input), and 'Description' (text input). The 'Account Information' section contains fields for 'Account Code' (dropdown), 'Account Name' (text input), and 'Dr/Cr' (dropdown). Below these sections is a table with columns: Action, Account Code T1, Account Name T1, and Dr/Cr T1. The table has one row with the message 'No Data Found'. At the bottom of the table, there are buttons for 'Page Size' (set to 10), '(0 Items)', and navigation arrows. The status bar at the bottom right shows 'Page: 1 / 1'.

ເປັນໜ້າກາຮຕັ້ງຄ່າເພີ່ມຄ່າໃຊ້ຈ່າຍອື່ນໆ ທີ່ເຮົາຈະນຳໄປໃຫ້ກັບ Other Charge Request ແລະ Other Expense Request

Other Charge Request Booking

Sippakorn Khammaswong ▾

Other Charge Request Booking

Request No.	Charge to Code *	Charge to Name (TH)	Charge to Name (EN)
Nationality • Thai - Foreigner	ID No.		
Invoice Name Select an option	Payment Due Date	01/11/2025	
Invoice Address			

Billing Information

Billing Name	Email Address	Billing Address
Mobile Number	Ex. xx@gmail.com	Office Phone Ext. 1001 Home Phone Ext. 022345678
Ex. 0999999999	Ex. 1001	Ex. 022345678

Other Charge Details

Other Charge Transaction

Transaction Code *Select transaction code

Description

Currency	Exchange Rate	Amount(Currency)	Amount(THB)		
THB	1.000000	0.00	0.00		
VAT Calculation	VAT Type %	VAT Rate %	VAT Amount(THB)	Total Amount(Currency)	Total Amount(THB)
• VAT - Non VAT	VAT Type %	VAT Rate %	0.00	0.00	0.00
WHT Rate %	WHT Type %	WHT Rate %	WHT Amount(THB)	Net Amount(Currency)	Net Amount(THB)
WHT Rate %	WHT Type %	WHT Rate %	0.00	0.00	0.00

Request Information

Action	Line No. T1	Transaction T1	Description T1	Currency T1	Exchange Rate T1	Amount(Currency) T1	Amount(THB) T1	VAT Calculation T1	VAT Type T1	VAT Rate
✓	1									
✓	2									

125,000.00 125,000.00

Page Size: 10 (2 items) ▾ Page: 1 / 1 ▾ Update

Installment

Attachments

Document Type * Select an option Expiry Date Description Upload File Click to upload or drag & drop (5MB) + Add Clear

Action	Document Type T1	Expiry Date T1	Description T1	Attached Files	Attached By T1	Attached Date T1
No Data Found						

Page Size: 10 (0 items) ▾ Page: 1 / 1 ▾

Share Department

Department Share (%) Share Income Amount Upfront

Action	Department T1	Share (%) T1	Share Income Amount Upfront T1
✓		100.00	

123,364.49

Page Size: 10 (0 items) ▾ Page: 1 / 1 ▾

Account Booking

Action	Account Code	Account Name	Voucher Line	Voucher Distrib	Dept ID	Dr	Cr
✓							
Total							

Page Size: 10 (0 items) ▾ Page: 1 / 1 ▾

Entry By G_Touch Entry Date 08/10/2025 Modification By G_Touch Modification Date 09/10/2025 Status Account Booking Submit Save Draft Clear

เป็นหน้าสำหรับแพนกบัญชีที่ผ่านการ approve มาจาก head ของแผนกแล้ว มาทำการตรวจสอบเช็คและแก้ไขเพื่อจะทำต่อไปในกระบวนการของบัญชี

Other Charge Request Booking List

เมื่อนหน้าที่อาชีวศึกษา/เรียนต์รายการ Other Charge Request ที่ถูกอนุมัติมาจากขั้นตอนของการ Approve จากค่อนหน้านี้

LWT Bank Account

,

Angular Tutorial

What is Angular?

This topic can help you understand Angular: what Angular is, what advantages it provides, and what you might expect as you start to build your applications.

Angular is a development platform, built on [TypeScript](#). As a platform, Angular includes:

- A component-based framework for building scalable web applications
- A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more
- A suite of developer tools to help you develop, build, test, and update your code

With Angular, you're taking advantage of a platform that can scale from single-developer projects to enterprise-level applications. Angular is designed to make updating as straightforward as possible, so take advantage of the latest developments with minimal effort. Best of all, the Angular ecosystem consists of a diverse group of over 1.7 million developers, library authors, and content creators.

Angular applications: The essentials

This section explains the core ideas behind Angular. Understanding these ideas can help you design and build your applications more effectively.

Components

Components are the building blocks that compose an application. A component includes a TypeScript class with a [@Component\(\)](#) decorator, an HTML template, and styles. The [@Component\(\)](#) decorator specifies the following Angular-specific information:

- A CSS selector that defines how the component is used in a template. HTML elements in your template that match this selector become instances of the component.
- An HTML template that instructs Angular how to render the component
- An optional set of CSS styles that define the appearance of the template's HTML elements

The following is a minimal Angular component.

The following is a minimal Angular component.

```
import { Component } from '@angular/core';

@Component({
  selector: 'hello-world',
  template: `
    <h2>Hello World</h2>
    <p>This is my first component!</p>
  `
})
export class HelloWorldComponent {
  // The code in this class drives the component's behavior.
}
```

To use this component, you write the following in a template:

```
<hello-world></hello-world>
```

When Angular renders this component, the resulting DOM looks like this:

```
<hello-world>
  <h2>Hello World</h2>
  <p>This is my first component!</p>
</hello-world>
```

Angular's component model offers strong encapsulation and an intuitive application structure. Components also make your application painless to unit test and can improve the general readability of your code.

For more information on what to do with components, see the [Components](#) section.

,

Templates

Every component has an HTML template that declares how that component renders. You define this template either inline or by file path.

Angular adds syntax elements that extend HTML so you can insert dynamic values from your component.

Angular automatically updates the rendered DOM when your component's state changes. One application of this feature is inserting dynamic text, as shown in the following example.

```
<p>{{ message }}</p>
```



The value for message comes from the component class:

```
import { Component } from '@angular/core';

@Component ({
  selector: 'hello-world-interpolation',
  templateUrl: './hello-world-interpolation.component.html'
})
export class HelloWorldInterpolationComponent {
  message = 'Hello, World!';
}
```



When the application loads the component and its template, the user sees the following:

```
<p>Hello, World!</p>
```



Notice the use of double curly braces—they instruct Angular to interpolate the contents within them.

Angular also supports property bindings, to help you set values for properties and attributes of HTML elements and pass values to your application's presentation logic.

,

```
<p  
  [id]="sayHelloId"  
  [style.color]="fontColor">  
  You can set my color in the component!  
</p>
```

Notice the use of the square brackets—that syntax indicates that you're binding the property or attribute to a value in the component class.

Declare event listeners to listen for and respond to user actions such as keystrokes, mouse movements, clicks, and touches. You declare an event listener by specifying the event name in parentheses:

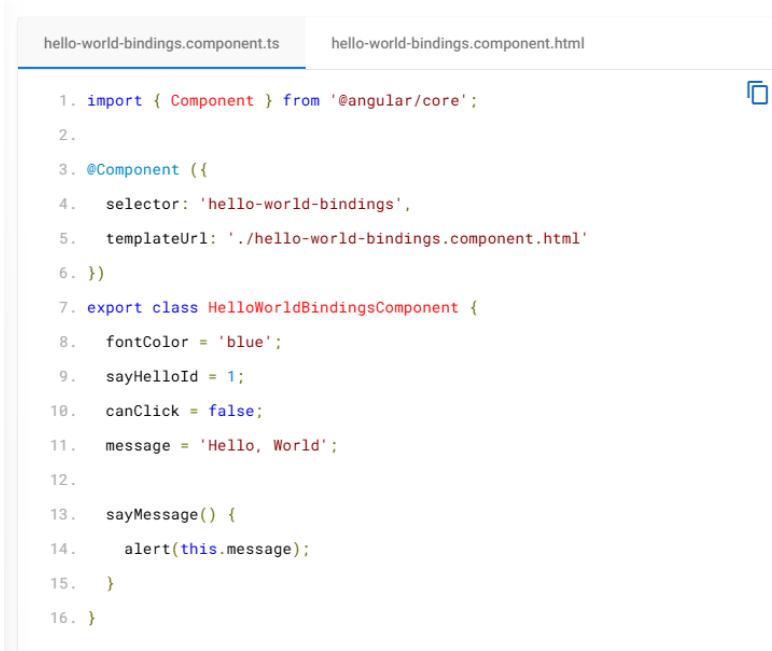
```
<button  
  type="button"  
  [disabled]="canClick"  
  (click)="sayMessage()">  
  Trigger alert message  
</button>
```

The preceding example calls a method, which is defined in the component class:

```
sayMessage() {  
  alert(this.message);  
}
```

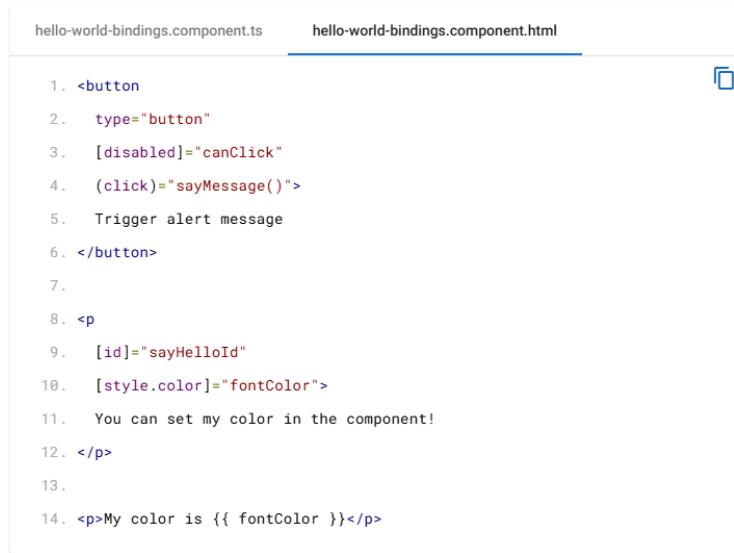
The following is a combined example of Interpolation, Property Binding, and Event Binding within an Angular template:

,



The screenshot shows a code editor with two tabs: "hello-world-bindings.component.ts" and "hello-world-bindings.component.html". The "hello-world-bindings.component.ts" tab is active, displaying the following TypeScript code:

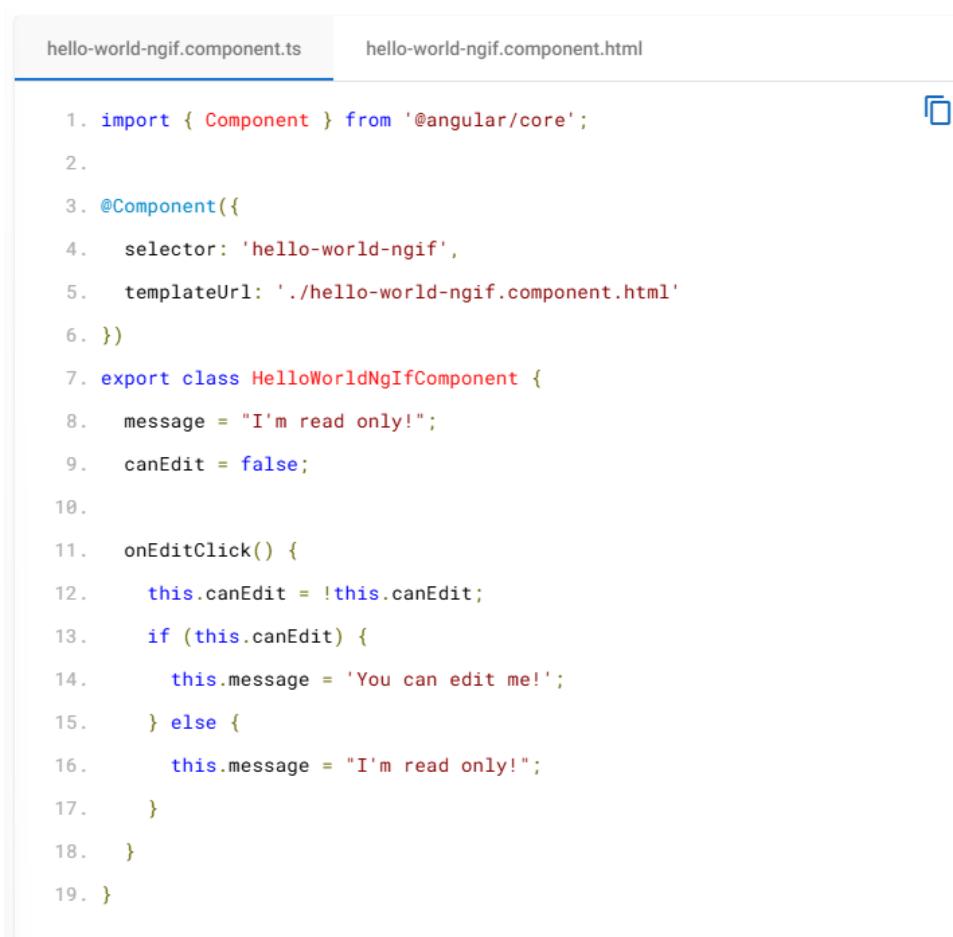
```
1. import { Component } from '@angular/core';
2.
3. @Component ({
4.   selector: 'hello-world-bindings',
5.   templateUrl: './hello-world-bindings.component.html'
6. })
7. export class HelloWorldBindingsComponent {
8.   fontColor = 'blue';
9.   sayHelloId = 1;
10.  canClick = false;
11.  message = 'Hello, World';
12.
13.  sayMessage() {
14.    alert(this.message);
15.  }
16. }
```



The screenshot shows a code editor with two tabs: "hello-world-bindings.component.ts" and "hello-world-bindings.component.html". The "hello-world-bindings.component.html" tab is active, displaying the following HTML code:

```
1. <button
2.   type="button"
3.   [disabled]="canClick"
4.   (click)="sayMessage()"
5.   Trigger alert message
6. </button>
7.
8. <p
9.   [id]="sayHelloId"
10.  [style.color]="fontColor">
11.  You can set my color in the component!
12. </p>
13.
14. <p>My color is {{ fontColor }}</p>
```

,



The screenshot shows a code editor with two tabs: "hello-world-ngif.component.ts" and "hello-world-ngif.component.html". The "hello-world-ngif.component.ts" tab is active, displaying the following TypeScript code:

```
1. import { Component } from '@angular/core';
2.
3. @Component({
4.   selector: 'hello-world-ngif',
5.   templateUrl: './hello-world-ngif.component.html'
6. })
7. export class HelloWorldNgIfComponent {
8.   message = "I'm read only!";
9.   canEdit = false;
10.
11.   onEditClick() {
12.     this.canEdit = !this.canEdit;
13.     if (this.canEdit) {
14.       this.message = 'You can edit me!';
15.     } else {
16.       this.message = "I'm read only!";
17.     }
18.   }
19. }
```

Add features to your templates by using [directives](#). The most popular directives in Angular are `*ngIf` and `*ngFor`. Use directives to perform a variety of tasks, such as dynamically modifying the DOM structure. And create your own custom directives to create great user experiences.

The following code is an example of the `*ngIf` directive.

,

Angular's declarative templates let you cleanly separate your application's logic from its presentation.

Templates are based on standard HTML, for ease in building, maintaining, and updating.

For more information on templates, see the [Templates](#) section.

Dependency injection

Dependency injection lets you declare the dependencies of your TypeScript classes without taking care of their instantiation. Instead, Angular handles the instantiation for you. This design pattern lets you write more testable and flexible code. Understanding dependency injection is not critical to start using Angular, but it is strongly recommended as a best practice. Many aspects of Angular take advantage of it to some degree.

To illustrate how dependency injection works, consider the following example. The first file, logger.service.ts, defines a Logger class. This class contains a writeCount function that logs a number to the console.

```
import { Injectable } from '@angular/core';

@Injectable({providedIn: 'root'})
export class Logger {
  writeCount(count: number) {
    console.warn(count);
  }
}
```

Next, the hello-world-di.component.ts file defines an Angular component. This component contains a button that uses the writeCount function of the Logger class. To access that function, the Logger service is injected into the HelloWorldDI class by adding private logger: Logger to the constructor.

,

```
import { Component } from '@angular/core';
import { Logger } from '../logger.service';

@Component({
  selector: 'hello-world-di',
  templateUrl: './hello-world-di.component.html'
})
export class HelloWorldDependencyInjectionComponent {
  count = 0;

  constructor(private logger: Logger) { }

  onLogMe() {
    this.logger.writeCount(this.count);
    this.count++;
  }
}
```

For more information about dependency injection and Angular, see the [Dependency injection in Angular](#) section.

First-party libraries

The section, [Angular applications: the essentials](#), provides a brief overview of a couple of the key architectural elements that are used when building Angular applications. The many benefits of Angular really become clear when your application grows and you want to add functions such as site navigation or user input. Use the Angular platform to incorporate one of the many first-party libraries that Angular provides.

Some of the libraries available to you include:

,

LIBRARY	DETAILS
Angular Router	Advanced client-side navigation and routing based on Angular components. Supports lazy-loading, nested routes, custom path matching, and more.
Angular Forms	Uniform system for form participation and validation.
Angular HttpClient	Robust HTTP client that can power more advanced client-server communication.
Angular Animations	Rich system for driving animations based on application state.
Angular PWA	Tools for building Progressive Web Applications (PWA) including a service worker and Web application manifest.
Angular Schematics	Automated scaffolding, refactoring, and update tools that simplify development at large scale.

These libraries expand your application's capabilities while also letting you focus more on the features that make your application unique. Add these libraries knowing that they're designed to integrate flawlessly into and update simultaneously with the Angular framework.

These libraries are only required when they can help you add features to your applications or solve a particular problem.

Modules

Angular *NgModules* differ from and complement JavaScript (ES2015) modules. An NgModule declares a compilation context for a set of components that is dedicated to an application domain, a workflow, or a

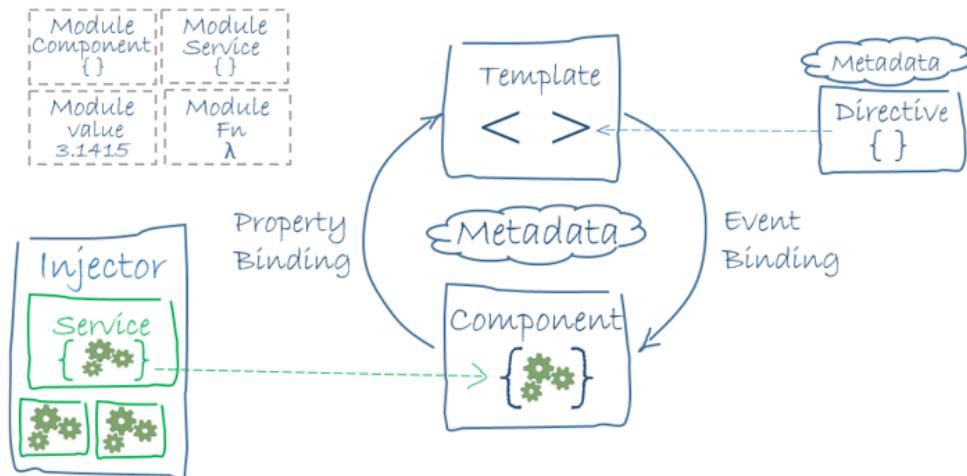
,

closely related set of capabilities. An NgModule can associate its components with related code, such as services, to form functional units.

Every Angular application has a *root module*, conventionally named AppModule, which provides the bootstrap mechanism that launches the application. An application typically contains many functional modules.

Like JavaScript modules, NgModules can import functionality from other NgModules, and allow their own functionality to be exported and used by other NgModules. For example, to use the router service in your app, you import the [Router](#) NgModule.

Organizing your code into distinct functional modules helps in managing development of complex applications, and in designing for reusability. In addition, this technique lets you take advantage of *lazy-loading* —that is, loading modules on demand— to minimize the amount of code that needs to be loaded at startup.



,

,

Spring Boot

Spring Beans and Dependency Injection

You are free to use any of the standard Spring Framework techniques to define your beans and their injected dependencies. We often find that using `@ComponentScan` (to find your beans) and using `@Autowired` (to do constructor injection) works well.

If you structure your code as suggested above (locating your application class in a root package), you can add `@ComponentScan` without any arguments. All of your application components (`@Component`, `@Service`, `@Repository`, `@Controller` etc.) are automatically registered as Spring Beans.

The following example shows a `@Service` Bean that uses constructor injection to obtain a required `RiskAssessor` bean:

```
package com.example.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class DatabaseAccountService implements AccountService {

    private final RiskAssessor riskAssessor;

    @Autowired
    public DatabaseAccountService(RiskAssessor riskAssessor) {
        this.riskAssessor = riskAssessor;
    }

    // ...
}
```

,

If a bean has one constructor, you can omit the `@Autowired`, as shown in the following example.

```
@Service
public class DatabaseAccountService implements AccountService {

    private final RiskAssessor riskAssessor;

    public DatabaseAccountService(RiskAssessor riskAssessor) {
        this.riskAssessor = riskAssessor;
    }

    // ...
}
```

Using the `@SpringBootApplication` Annotation

Many Spring Boot developers like their apps to use auto-configuration, component scan and be able to define extra configuration on their "application class". A single `@SpringBootApplication` annotation can be used to enable those three features, that is:

- `@EnableAutoConfiguration`: enable Spring Boot's auto-configuration mechanism
- `@ComponentScan`: enable `@Component` scan on the package where the application is located (see the best practices)
- `@Configuration`: allow to register extra beans in the context or import additional configuration classes

```
package com.example.myapplication;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication // same as @Configuration @EnableAutoConfiguration
@ComponentScan
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

,

To finish our application, we need to create a single Java file. By default, Maven compiles sources from `src/main/java`, so you need to create that directory structure and then add a file named `src/main/java/Example.java` to contain the following code:

```
import org.springframework.boot.*;
import org.springframework.boot.autoconfigure.*;
import org.springframework.web.bind.annotation.*;

@RestController
@EnableAutoConfiguration
public class Example {

    @RequestMapping("/")
    String home() {
        return "Hello World!";
    }

    public static void main(String[] args) {
        SpringApplication.run(Example.class, args);
    }
}
```

Although there is not much code here, quite a lot is going on. We step through the important parts in the next few sections.

The `@RestController` and `@RequestMapping` Annotations

The first annotation on our `Example` class is `@RestController`. This is known as a stereotype annotation. It provides hints for people reading the code and for Spring that the class plays a specific role. In this case, our class is a web `@Controller`, so Spring considers it when handling incoming web requests.

The `@RequestMapping` annotation provides “routing” information. It tells Spring that any HTTP request with the `/` path should be mapped to the `home` method. The `@RestController` annotation tells Spring to render the resulting string directly back to the caller.

,

The `@EnableAutoConfiguration` Annotation

The second class-level annotation is `@EnableAutoConfiguration`. This annotation tells Spring Boot to “guess” how you want to configure Spring, based on the jar dependencies that you have added. Since `spring-boot-starter-web` added Tomcat and Spring MVC, the auto-configuration assumes that you are developing a web application and sets up Spring accordingly.

Developing Web Application

Spring Boot is well suited for web application development. You can create a self-contained HTTP server by using embedded Tomcat, Jetty, Undertow, or Netty. Most web applications use the `spring-boot-starter-web` module to get up and running quickly. You can also choose to build reactive web applications by using the `spring-boot-starter-webflux` module. If you have not yet developed a Spring Boot web application, you can follow the “Hello World!” example in the Getting started section.

The “Spring Web MVC Framework”

The Spring Web MVC framework (often referred to as “Spring MVC”) is a rich “model view controller” web framework. Spring MVC lets you create special `@Controller` or `@RestController` beans to handle incoming HTTP requests. Methods in your controller are mapped to HTTP by using `@RequestMapping` annotations.

The following code shows a typical `@RestController` that serves JSON data:

```
@RestController
@RequestMapping(value="/users")
public class MyRestController {

    @RequestMapping(value="/{user}", method=RequestMethod.GET)
    public User getUser(@PathVariable Long user) {
        // ...
    }

    @RequestMapping(value="/{user}/customers", method=RequestMethod.GET)
    List<Customer> getUserCustomers(@PathVariable Long user) {
        // ...
    }

    @RequestMapping(value="/{user}", method=RequestMethod.DELETE)
    public User deleteUser(@PathVariable Long user) {
        // ...
    }
}
```

,

Spring MVC is part of the core Spring Framework, and detailed information is available in the reference documentation. There are also several guides that cover Spring MVC available at spring.io/guides. Spring MVC Auto-configuration Spring Boot provides auto-configuration for Spring MVC that works well with most applications. The auto-configuration adds the following features on top of Spring's defaults:

- Inclusion of ContentNegotiatingViewResolver and BeanNameViewResolver beans.
- Support for serving static resources, including support for WebJars (covered later in this document)).
- Automatic registration of Converter, GenericConverter, and Formatter beans.
- Support for HttpMessageConverters (covered later in this document).
- Automatic registration of MessageCodesResolver (covered later in this document).
- Static index.html support.
- Automatic use of a ConfigurableWebBindingInitializer bean (covered later in this document)

If you want to keep those Spring Boot MVC customizations and make more MVC customizations (interceptors, formatters, view controllers, and other features), you can add your own `@Configuration` class of type `WebMvcConfigurer` but without `@EnableWebMvc`.

If you want to provide custom instances of `RequestMappingHandlerMapping`, `RequestMappingHandlerAdapter`, or `ExceptionHandlerExceptionResolver`, and still keep the Spring Boot MVC customizations, you can declare a bean of type `WebMvcRegistrations` and use it to provide custom instances of those components.

If you want to take complete control of Spring MVC, you can add your own `@Configuration` annotated with `@EnableWebMvc`, or alternatively add your own `@Configuration`-annotated `DelegatingWebMvcConfiguration` as described in the Javadoc of `@EnableWebMvc`

,

HttpMessageConverters

Spring MVC uses the `HttpMessageConverter` interface to convert HTTP requests and responses. Sensible defaults are included out of the box. For example, objects can be automatically converted to JSON (by using the Jackson library) or XML (by using the Jackson XML extension, if available, or by using JAXB if the Jackson XML extension is not available). By default, strings are encoded in UTF 8.

If you need to add or customize converters, you can use Spring Boot's `HttpMessageConverters` class, as shown in the following listing:

```
import org.springframework.boot.autoconfigure.http.HttpMessageConverters;
import org.springframework.context.annotation.*;
import org.springframework.http.converter.*;

@Configuration(proxyBeanMethods = false)
public class MyConfiguration {

    @Bean
    public HttpMessageConverters customConverters() {
        HttpMessageConverter<?> additional = ...
        HttpMessageConverter<?> another = ...
        return new HttpMessageConverters(additional, another);
    }
}
```

Any `HttpMessageConverter` bean that is present in the context is added to the list of converters. You can also override default converters in the same way.

,

SQL Server

1. What is Microsoft SQL Server?

Microsoft SQL Server is a **Relational Database Management System (RDBMS)** developed by Microsoft. Its primary function is to store and retrieve data as requested by other software applications. It can run on a single machine or across a network.

The system uses **Transact-SQL (T-SQL)** as its primary query language, which is Microsoft's proprietary extension of the standard SQL (Structured Query Language).

2. Core Components

SQL Server is not just a database engine; it's a comprehensive data platform that includes several key services:

- **Database Engine:** The core service responsible for storing, processing, and securing data. It handles transactions, locking, and concurrency.
- **SQL Server Integration Services (SSIS):** A platform for building high-performance data integration and Extract, Transform, Load (ETL) solutions.
- **SQL Server Analysis Services (SSAS):** Provides data analysis and business intelligence capabilities, allowing users to create and manage multidimensional data cubes (OLAP) and tabular models.
- **SQL Server Reporting Services (SSRS):** A server-based reporting platform used to create, manage, and deliver a wide variety of interactive and printed reports.

3. Common SQL Server Editions

SQL Server comes in several editions tailored for different needs:

- **Enterprise:** The premium, full-featured edition designed for large-scale, mission-critical applications. It includes advanced features for high availability, security, and performance.

,

- **Standard:** Provides core database capabilities, reporting, and analytics for mid-tier applications and data marts. It has fewer features and lower capacity limits than Enterprise.
- **Express:** A free, entry-level edition ideal for learning, development, and small-scale applications. It has significant limitations on database size, memory, and CPU usage.
- **Developer:** A free, full-featured edition (identical to Enterprise) licensed only for non-production development and testing environments.

4. Key Concepts in SQL Server

- **Database:** A structured collection of data. It contains objects like tables, views, stored procedures, and functions.
- **Table:** The primary object for storing data, organized into rows (records) and columns (attributes).
- **Schema:** A container for database objects. It helps in organizing objects and managing permissions. The default schema is dbo (database owner).
- **T-SQL (Transact-SQL):** The language used to interact with the database. It includes:
 - **DDL (Data Definition Language):** Commands like CREATE, ALTER, DROP (to define database structure).
 - **DML (Data Manipulation Language):** Commands like SELECT, INSERT, UPDATE, DELETE (to query and modify data).
 - **DCL (Data Control Language):** Commands like GRANT, REVOKE (to manage permissions).
- **Stored Procedure:** A pre-compiled set of one or more T-SQL statements saved in the database. It improves performance, security, and reusability.
- **Index:** A data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space.

,

5. Common T-SQL Command Examples

Here are practical examples of the most frequently used Transact-SQL commands for daily database operations.

A. Data Manipulation Language (DML) - Working with Data

Assume we have a table named Employees with columns: EmployeeID, FirstName, LastName, and Department.

1. SELECT (Querying Data) *To retrieve all employees from the 'Sales' department:*

SQL



```
SELECT EmployeeID, FirstName, LastName  
FROM Employees  
WHERE Department = 'Sales';
```

2. INSERT (Adding New Data) *To add a new employee record to the table:*

SQL



```
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department)  
VALUES (101, 'John', 'Doe', 'Marketing');
```

3. UPDATE (Modifying Existing Data) *To change an existing employee's department:*

SQL



```
UPDATE Employees  
SET Department = 'Human Resources'  
WHERE EmployeeID = 101;
```

,

- 4. DELETE (Removing Data)** To remove an employee's record from the table:

SQL

```
DELETE FROM Employees  
WHERE EmployeeID = 101;
```



B. Data Definition Language (DDL) - Defining Database Structure

- 1. CREATE TABLE** To create a new table to store product information:

SQL

```
CREATE TABLE Products (  
    ProductID INT PRIMARY KEY,  
    ProductName VARCHAR(100) NOT NULL,  
    Price DECIMAL(10, 2),  
    DateAdded DATETIME DEFAULT GETDATE()  
);
```



- 2. ALTER TABLE** To add a new column for 'StockQuantity' to the Products table:

SQL

```
ALTER TABLE Products  
ADD StockQuantity INT;
```



- 3. DROP TABLE** To completely delete the Products table and all its data (use with caution):

SQL

```
DROP TABLE Products;
```



,

C. Stored Procedures and Joins

1. CREATE STORED PROCEDURE To create a reusable command that gets employees by department:

```
SQL

CREATE PROCEDURE sp_GetEmployeesByDept
    @DeptName VARCHAR(50)
AS
BEGIN
    SELECT EmployeeID, FirstName, LastName
    FROM Employees
    WHERE Department = @DeptName
    ORDER BY LastName;
END;
```

To use (execute) this procedure:

```
SQL

EXEC sp_GetEmployeesByDept @DeptName = 'Sales';
```

2. JOIN (Combining Data from Multiple Tables) Assume we have a second table Departments (DeptName, Manager). To get a list of employees and their managers:

```
SQL

SELECT
    e.FirstName,
    e.LastName,
    d.Manager
FROM
    Employees AS e
INNER JOIN
    Departments AS d ON e.Department = d.DeptName;
```