



CS 2567/CSC64

รายงานความก้าวหน้าโครงการ ครั้งที่ 2

การออกแบบและการพัฒนาแชทบอท เพื่อการท่องเที่ยว

ในจังหวัดขอนแก่น

Chatbot design and development

for tourism in Khon Kaen

โดย

643020603-8 นางสาวชลธิชา บุญสันต์

643021345-9 นางสาวอารีรักษ์ กลางอินทร์เดช

อาจารย์ที่ปรึกษาโครงการ : ผศ.ดร.พุทธิดี ศิริแสงตระกูล

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา SC314775 โครงการวิทยาการคอมพิวเตอร์ 2

ภาคการเรียนที่ 2 ปีการศึกษา 2567

สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์

มหาวิทยาลัยขอนแก่น

(เดือน มกราคม พ.ศ. 2568)



CS 2567/CSC64

รายงานความก้าวหน้าโครงการ ครั้งที่ 2

การออกแบบและการพัฒนาแชทบอท เพื่อการท่องเที่ยว

ในจังหวัดขอนแก่น

Chatbot design and development

for tourism in Khon Kaen

โดย

643020603-8 นางสาวชลธิชา บุญสันต์

643021345-9 นางสาวอารีรักษ์ กลางอินทร์เดช

อาจารย์ที่ปรึกษาโครงการ : ผศ.ดร.พุทธิดี ศิริแสงตระกูล

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา SC314775 โครงการวิทยาการคอมพิวเตอร์ 2

ภาคการเรียนที่ 2 ปีการศึกษา 2567

สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์

มหาวิทยาลัยขอนแก่น

(เดือน มกราคม พ.ศ. 2568)

นางสาวชลธิชา บุญสนัด และนางสาวอารีรักษ์ กลางอินทร์เดช. 2567. การออกแบบและการพัฒนาแชทบอท เพื่อ  
การท่องเที่ยวในจังหวัดขอนแก่น. โครงการงานคอมพิวเตอร์ วิทยาลัยการศึกษาศาสตร์บัณฑิต สาขาวิชาวิทยาการ  
คอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น.

อาจารย์ที่ปรึกษา: ผศ.ดร.พุทธิศักดิ์ ศิริแสงตระกูล

## บทคัดย่อ

งานวิจัยนี้นำเสนอการออกแบบและพัฒนาแชทบอทการท่องเที่ยวในจังหวัดขอนแก่น โดยมีวัตถุประสงค์ เพื่อรวบรวมคำถามที่เกี่ยวข้องกับการท่องเที่ยวในจังหวัดขอนแก่น และเพื่อออกแบบและพัฒนาแชทบอทให้สามารถตอบคำถาม แล้วให้ข้อมูลของสถานที่ท่องเที่ยวในจังหวัดขอนแก่นผ่านแอปพลิเคชันไลน์ โดยจะทำการรวบรวมความคิดเห็นและความต้องการของนักท่องเที่ยวผ่านแบบสอบถามออนไลน์ การสำรวจจากเว็บไซต์รีวิวและโซเชียลมีเดีย ข้อมูลที่ได้ถูกนำมาวิเคราะห์และออกแบบโครงสร้างของแชทบอท โดยจะทำการจำแนกประเภทตามสถานที่ท่องเที่ยวและอำเภอของจังหวัดขอนแก่น แชทบอทนี้ใช้เทคโนโลยีการประมวลผลภาษาธรรมชาติ เพื่อให้สามารถตอบสนองคำถามเกี่ยวกับสถานที่ท่องเที่ยว และข้อมูลสำคัญอื่น ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพ การพัฒนาแชทบอทจะใช้แพลตฟอร์มไดอะล็อกฟลอร์สำหรับการสร้างและฝึกสอนอินเทรนด์ที่เกี่ยวข้องกับการท่องเที่ยวในขอนแก่น โดยข้อมูลจะถูกเก็บไว้ในฐานข้อมูลโพสต์เกรสคิวเอล เพื่อให้สามารถเข้าถึงได้ง่ายและรวดเร็ว นอกจากนี้ยังมีการสร้างเซิร์ฟเวอร์โหนดเจเอสเพื่อเชื่อมต่อระหว่างไลน์และไดอะล็อกฟลอร์ ฐานข้อมูล ทำให้แชทบอทยังสามารถค้นหาข้อมูลเพิ่มเติมจากเว็บไซต์เมื่อไม่มีคำตอบในฐานข้อมูลได้ เพื่อให้ผู้ใช้ได้รับข้อมูลที่หลากหลายและครบถ้วน หากคำถามที่แชทบอทไม่สามารถตอบได้ ก็จะถูกบันทึกลงฐานข้อมูล เพื่อใช้ในการพัฒนาแชทบอทให้มีความครบถ้วนและครอบคลุมมากยิ่งขึ้น นอกจากนี้ยังพัฒนาได้จัดทำแบบสำรวจความพึงพอใจของผู้ใช้งานที่มีต่อแชทบอททุกแพลตฟอร์มเพื่อนำผลการสำรวจมาพัฒนาและปรับปรุงประสิทธิภาพแชทบอทให้มีคุณภาพต่อไป

**คำสำคัญ:** แชทบอท, การท่องเที่ยว, ขอนแก่น, การประมวลผลภาษาธรรมชาติ

Miss Cholthicha Boonsanud and Miss Arreerak Klangindet. 2024. **Chatbot design and development for tourism in Khon Kaen**. Bachelor of Science Project in Computer Science, College of Computer, Khon Kaen University.

**Thesis Advisors:** Asst. Prof. Pusadee Seresangtakul, Ph.D.

## ABSTRACT

This research presents the design and development of a travel chatbot for Khon Kaen province. The objective is to gather questions related to tourism in Khon Kaen and to design and develop a chatbot that can answer questions and provide information about tourist attractions in Khon Kaen through the LINE application. Feedback and needs from tourists will be collected through online surveys, reviews from websites, and social media. The collected data will be analyzed to design the chatbot's structure, categorizing it by tourist attractions and districts in Khon Kaen. This chatbot uses natural language processing technology to respond to questions about tourist attractions and other important information quickly and efficiently. The chatbot development will use the Dialogflow platform to create and train intents related to tourism in Khon Kaen. The data will be stored in a PostgreSQL database for easy and quick access. Additionally, a Node.js server will be created to connect LINE, Dialogflow, and the database, allowing the chatbot to search for additional information from websites when there are no answers in the database. This ensures users receive diverse and comprehensive information. If the chatbot cannot answer a question, it will be recorded in the database to improve the chatbot's completeness and coverage. Furthermore, the developers have created a user satisfaction survey for the chatbot using Google Forms to gather survey results for further development and improvement of the chatbot's quality.

**Keywords:** Chatbot, Tourism, Khon Kaen, Natural Language Processing

## กิตติกรรมประกาศ

ในการดำเนินโครงการครั้งนี้ ผู้จัดทำโครงการได้รับความอนุเคราะห์ และความช่วยเหลือจากบุคคลหลายท่านด้วยกัน จึงขอขอบคุณไว้ ณ โอกาสนี้

ขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้

ขอขอบพระคุณอาจารย์ ผศ.ดร.พฤษดี ศิริแสงตระกูล ที่เป็นพี่ปรึกษาโครงการและได้ให้คำชี้แนะนำ แนวทางในการคิดวิเคราะห์ และการพัฒนาขอบเขตของโครงการนี้

ขอขอบคุณเพื่อนๆ ที่ให้คำปรึกษา

ผู้จัดทำ

ชลธิชา บุญสันต์

อารีรักษ์ กลางอินทร์เดช

## สารบัญ

	หน้า
บทคัดย่อ	ก
ABSTRACT	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญภาพ	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1. ความเป็นมาและความสำคัญ	1
2. วัตถุประสงค์ของโครงการ	1
3. เป้าหมายและขอบเขต	1
4. ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง	4
1. งานวิจัยที่เกี่ยวข้อง	4
2. ทฤษฎีที่เกี่ยวข้อง	5
บทที่ 3 วิธีการดำเนินงาน	8
1. การกำหนดขอบเขตและเป้าหมาย	8
2. ศึกษาข้อมูลและวิธีใช้งานเครื่องมือต่างๆ ที่ใช้งานวิจัย	8
3. การเก็บรวบรวมข้อมูล	10
บทที่ 4 การวิเคราะห์ระบบ และการพัฒนา	11
1. การวิเคราะห์ระบบ	11
2. การออกแบบแซทบอท	31
3. การพัฒนาแซทบอท	35
4. การทดสอบแซทบอท	76
5. การประเมินความพึงพอใจของผู้ใช้งาน	99
บทที่ 5 บทสรุป	102
1. สรุปผลการดำเนินโครงการ	102
2. ข้อจำกัด	102
3. ปัญหาอุปสรรค	102

4. ข้อเสนอแนะ	102
เอกสารอ้างอิง	103
ภาคผนวก	105
ภาคผนวก ก แบบสอบถามความคิดเห็นของผู้ที่สนใจท่องเที่ยว	106
ภาคผนวก ข แบบสอบถามความพึงพอใจการใช้งานงานเซทบอท	107

## สารบัญภาพ

	หน้า
ภาพที่ 1 แผนภาพสถาปัตยกรรมของระบบ	12
ภาพที่ 2 ยูสเคสไดอะแกรมของการออกแบบและพัฒนาแชทบอท การท่องเที่ยวจังหวัดขอนแก่น	20
ภาพที่ 3 แผนภาพอี-อาร์ไดอะแกรม (ER Diagram)	26
ภาพที่ 4 หลักการทำงานโดยรวมของแชทบอทการท่องเที่ยวจังหวัดขอนแก่น	31
ภาพที่ 5 ขั้นตอนการส่งข้อมูลเพื่อใช้ในการโต้ตอบการสนทนา	32
ภาพที่ 6 โครงสร้างโดยรวมของแชทบอท (Chatbot Conversational Flow)	33
ภาพที่ 7 โครงสร้างการสนทนาของระบบแชทบอท (Chatbot Conversational Flow)	34
ภาพที่ 8 การดึงข้อมูลโปรไฟล์ผู้ใช้	39
ภาพที่ 9 บันทึกหรืออัปเดตข้อมูลโปรไฟล์ผู้ใช้งานในฐานข้อมูล	41
ภาพที่ 10 การบันทึกการสนทนา	43
ภาพที่ 11 การดึงไลน์ไอดีจากฐานข้อมูล	44
ภาพที่ 12 บันทึกคำตอบที่ดึงมาจากเว็บไซต์ลงในฐานข้อมูล	45
ภาพที่ 13 การดึงข้อมูลมาใช้ในการตอบคำถามจากฐานข้อมูล	47
ภาพที่ 14 การดึงข้อมูลมาใช้ในการตอบคำถามจากฐานข้อมูล (ต่อ)	48
ภาพที่ 15 จัดการ Intent หลักที่มาจาก Dialogflow	52
ภาพที่ 16 จัดการ Intent หลักที่มาจาก Dialogflow (ต่อ)	53
ภาพที่ 17 การดึงข้อมูลจากเว็บ	58
ภาพที่ 18 การจัดการคำตอบของอินเทนท์กรณีดึงจากเว็บไซต์	59
ภาพที่ 19 การจัดการคำตอบของอินเทนท์กรณีดึงจากเว็บไซต์ (ต่อ)	62
ภาพที่ 20 การดึงแผนที่เส้นทางการเดินทาง	65
ภาพที่ 21 การดึงแผนที่เส้นทางการเดินทาง (ต่อ)	66
ภาพที่ 22 การจัดการพยากรณ์อากาศ	68
ภาพที่ 23 สร้างเวิร์กโฟลว์โหนดเจเอส และใช้เอ็กซ์เพรสเจเอส เพื่อสร้างเว็บฮุก (webhook)	70
ภาพที่ 24 สร้างเวิร์กโฟลว์โหนดเจเอส และใช้เอ็กซ์เพรสเจเอส เพื่อสร้างเว็บฮุก (webhook) (ต่อ)	71
ภาพที่ 25 การสร้างตารางในฐานข้อมูล	74
ภาพที่ 26 การกำหนดค่าและการตั้งค่าของเวิร์กโฟลว์	75



## สารบัญตาราง

	หน้า
ตารางที่ 1 ตารางแสดงข้อมูล Use Case Description สอบถามข้อมูลรายละเอียดสถานที่ท่องเที่ยว	21
ตารางที่ 2 ตารางแสดงข้อมูล Use Case Description การจัดการข้อมูลคำตอบ	22
ตารางที่ 3 ตารางแสดงข้อมูล Use Case Description การจัดการอินเทอร์เน็ต	23
ตารางที่ 4 ตารางแสดงข้อมูล Use Case Description ดูประวัติผู้ใช้งาน	24
ตารางที่ 5 ตารางแสดงข้อมูล Use Case Description ดูประวัติคำถามและคำตอบ	25
ตารางที่ 6 users	27
ตารางที่ 7 conversations	28
ตารางที่ 8 web_answers	29
ตารางที่ 9 place	30
ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น	76
ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น	92
ตารางที่ 12 สรุปผลประเมินความพึงพอใจผู้ใช้งาน	100

## บทที่ 1

### บทนำ

#### 1. ความเป็นมาและความสำคัญ

ปัจจุบันผู้คนให้ความสำคัญกับการท่องเที่ยวกันมากขึ้น และได้รับความนิยมเป็นอย่างมาก ไม่ว่าจะเป็นเพื่อพักผ่อนหย่อนใจ เพื่อสร้างแรงบันดาลใจ หรือเพื่อเปิดโลกทัศน์ให้กว้างขึ้น ซึ่งการท่องเที่ยวก็มีหลายรูปแบบ ไม่ว่าจะเป็นรูปแบบการท่องเที่ยวในแหล่งธรรมชาติ รูปแบบการท่องเที่ยวในแหล่งวัฒนธรรม เป็นต้น การท่องเที่ยวเป็นอุตสาหกรรมที่มีความสำคัญ และในจังหวัดขอนแก่นเองก็มีสถานที่ท่องเที่ยวที่น่าสนใจหลายแห่ง เช่น สถานที่ทางประวัติศาสตร์ วัฒนธรรม และธรรมชาติที่สวยงาม อย่างไรก็ตามในการค้นหาข้อมูล ผู้ใช้จำเป็นต้องเข้าเว็บไซต์จำนวนมากและติดตั้งแอปพลิเคชันต่างๆ ลงในโทรศัพท์มือถือของตน เพื่อสำรวจสถานที่และวางแผนการเดินทาง ซึ่งเป็นกระบวนการที่ใช้เวลานาน ด้วยเหตุผลนี้ผู้วิจัยจึงมีแนวคิดในการพัฒนาระบบแชทบอทสำหรับการท่องเที่ยวในจังหวัดขอนแก่น โดยจะรวบรวมคำถามที่เกี่ยวข้องกับการท่องเที่ยว นำมาพัฒนาเป็นแชทบอทเพื่อตอบคำถาม หรือให้ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่น และสามารถสื่อสารกับผู้ใช้งานได้ตลอด 24 ชั่วโมง มีการเรียนรู้อย่างต่อเนื่องจากการโต้ตอบของผู้ใช้

ดังนั้นงานวิจัยนี้จึงได้นำเสนอการออกแบบเว็บไซต์และการพัฒนาแชทบอทผ่านแอปพลิเคชันไลน์ เพื่อบริการข้อมูลของสถานที่ท่องเที่ยวในจังหวัดขอนแก่นให้กับนักท่องเที่ยวและผู้ที่สนใจ โดยสามารถตอบคำถามเพื่อแสดงข้อมูลรายละเอียดเกี่ยวกับสถานที่นั้นๆ และจำแนกประเภทของสถานที่ท่องเที่ยวให้สามารถค้นหาข้อมูลได้ง่ายมากขึ้น และเป็นอีกหนึ่งช่องทางในการส่งเสริมการท่องเที่ยวในจังหวัดขอนแก่นอีกด้วย

#### 2. วัตถุประสงค์ของโครงการ

- 2.1 เพื่อรวบรวมคำถามที่เกี่ยวข้องกับการท่องเที่ยวในจังหวัดขอนแก่น
- 2.2 เพื่อออกแบบและพัฒนาแชทบอทให้สามารถตอบคำถาม แล้วให้ข้อมูลของสถานที่ท่องเที่ยวในจังหวัดขอนแก่นผ่านแอปพลิเคชันไลน์

#### 3. เป้าหมายและขอบเขต

- 3.1 ขอบเขตของข้อมูลที่ทำให้การเก็บข้อมูลเพื่อนำมาพัฒนาแชทบอท
  - 3.1.1 คำถามเกี่ยวกับสถานที่ท่องเที่ยว
  - 3.1.2 ข้อมูลของสถานที่ท่องเที่ยว เพื่อนำมาตอบคำถาม
- 3.2 ขอบเขตของเนื้อหาที่แชทบอทสามารถตอบได้
  - 3.1.3 ข้อมูลทั่วไป

- (1) ชื่อสถานที่
- (2) ที่อยู่
- (3) เวลาเปิด-ปิด
- (4) ค่าเข้าชม
- (5) รูปภาพสถานที่
- (6) รายละเอียดเพิ่มเติมของสถานที่

#### 3.1.4 ข้อมูลแยกตามประเภท

- (1) ภูเขา
- (2) สวนสัตว์
- (3) อุทยานแห่งชาติ
- (4) วัด
- (5) พิพิธภัณฑ์
- (6) สวนน้ำ
- (7) สวนสาธารณะ
- (8) แหล่งซื้อปิ้ง
- (9) ร้านอาหารมิชลินไกด์

#### 3.1.5 ข้อมูลการจำแนกประเภทอำเภอ

- (1) อำเภอเมืองขอนแก่น
- (2) อำเภอน้ำพอง
- (3) อำเภอสีชมพูนุ
- (4) อำเภออุบลรัตน์
- (5) อำเภอภูเวียง
- (6) อำเภอหนองเรือ
- (7) อำเภอชุมแพ
- (8) อำเภอเวียงเก่า
- (9) อำเภอบ้านฝาง
- (10) อำเภอเขาสนกวาง
- (11) อำเภอเปือยน้อย
- (12) อำเภอกระนวน
- (13) อำเภอภูผาม่าน

#### 3.1.6 ข้อจำกัดของการวิจัย

- (1) แห่บอทรองรับเฉพาะภาษาไทย

(2) แหบทบอสามารถใช้งานได้นบนแอปพลิเคชันไลน์

(3) แหบทบอให้เฉพาะข้อมูลที่เกี่ยวข้องกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่น ได้แก่ ที่อยู่ เวลาเปิด-ปิด ค่าเข้าชม รูปภาพสถานที่ รายละเอียดของสถานที่ ร้านอาหารมิชลินไกด์

#### 4. ประโยชน์ที่คาดว่าจะได้รับ

4.1 ได้ฐานข้อมูลคำถามที่เกี่ยวข้องกับการท่องเที่ยวในจังหวัดขอนแก่น

4.2 ได้แหบทบอที่สามารถตอบคำถาม หรือให้ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่น และสามารถสื่อสารกับผู้ใช้งานได้ตลอด 24 ชั่วโมงผ่านทางแอปพลิเคชันไลน์

## บทที่ 2

### งานวิจัยและทฤษฎีที่เกี่ยวข้อง

#### 1. งานวิจัยที่เกี่ยวข้อง

1.1 งานวิจัยของ Reem Alotaibi , Ahlam Ali, Haya Alharthi เรื่อง AI Chatbot for Tourism Recommendations A Case Study in the City of Jeddah, Saudi Arabia [1]

ได้สร้างแชทบอทโดยใช้การประมวลผลภาษาธรรมชาติ และการเรียนรู้ของเครื่องจักร ซึ่งเป็นแชทบอทแบบข้อความปัญญาประดิษฐ์ที่พัฒนามาเป็นแอปพลิเคชันมือถือ จำลองการแชทกับผู้ใช้ในการประมวลผลภาษาธรรมชาติ นอกจากนี้ยังมีการโต้ตอบแบบสองทาง และเป็นจุดติดต่อเดียวสำหรับการสื่อสารของผู้ใช้ทั้งหมด และให้บริการกรณีการใช้งานในเมืองเจดดาห์ ประเทศซาอุดีอาระเบีย ทำให้พบว่าแชทบอทสามารถเข้าใจความหมายและคำขอของผู้ใช้ได้ นอกจากนี้ยังพบว่าการโต้ตอบมีประสิทธิภาพ การตอบสนองจะได้รับทันที ผู้ใช้มีความพึงพอใจและชอบแนวคิดที่พวกเขาสามารถแสดงความต้องการในภาษาธรรมชาติได้ และผู้ใช้อย่างสนุกสนานกับการโต้ตอบและรู้สึกมีส่วนร่วม และวิจัยนี้ไม่ได้กล่าวถึงข้อจำกัดใด ๆ อย่างชัดเจน สิ่งสำคัญคือต้องทราบว่าแชทบอทได้รับการประเมินในบริบทเฉพาะ เช่น เมืองเจดดาห์ ประเทศซาอุดีอาระเบีย ประสิทธิภาพของแชทบอทอาจแตกต่างกันไปตามบริบทหรือสถานที่อื่น ๆ

1.2 งานวิจัยของ Mario Casillo, Fabio Clarizia, Giuseppe D’Aniello b, Massimo DeSanto, Marco Lombardi, Domenico Santaniello เรื่อง CHAT-Bot: A cultural heritage aware teller-bot for supporting touristic experiences [2]

เสนอระบบผู้แนะนำที่สามารถพัฒนาเส้นทางท่องเที่ยวที่ปรับเปลี่ยนได้ โดยใช้อินเทอร์เฟซของแชทบอท ระบบแนะนำจุดสนใจและบริการที่เกี่ยวข้องตามโปรไฟล์ของนักท่องเที่ยวและบริบท สถาปัตยกรรมที่นำเสนอช่วยให้สามารถวิเคราะห์ข้อความเพื่อรับรู้บริบทและแนะนำบริการและเนื้อหาได้ ผลการทดลองครั้งแรกเป็นที่น่าพอใจและแสดงถึงศักยภาพของแนวทางที่เสนอ การพัฒนาในอนาคตอาจรวมถึงปฏิสัมพันธ์ที่มากขึ้นของระบบกับแหล่งข้อมูลและบริการใหม่ที่แตกต่างกัน การประยุกต์ใช้วิธีการที่เสนอกับสภาพแวดล้อมที่ซับซ้อนมากขึ้น และการปรับปรุงดังกล่าวตามความคิดเห็นที่ได้รับ สรุปได้ว่าแนวทางที่นำเสนอสามารถอำนวยความสะดวกแก่นักท่องเที่ยวในทุกช่วงเวลาของประสบการณ์การเดินทางของผู้ใช้

1.3 งานวิจัยของ Ana Paula Chaves, Jesse Egbert, Toby Hocking, Eck Doerry, Marco Aurelio Gerosa เรื่อง Chatbots Language Design: The Influence of Language Variation on User Experience with Tourist Assistant Chatbots [3]

นำเสนอการรวบรวมการสนทนาของผู้เชี่ยวชาญด้านโดเมนมนุษย์ (ผู้ช่วยนักท่องเที่ยว) ที่มีปฏิสัมพันธ์กับนักท่องเที่ยวในสถานการณ์การค้นหาข้อมูลนักท่องเที่ยว คลังสนทนาอีกคลังหนึ่งในโดเมนการท่องเที่ยวที่มีอยู่ทางออนไลน์และมักใช้ในการวิจัยภาษาธรรมชาติ การสนทนาที่รวบรวมไว้ได้รับการจัดการตามทฤษฎีการลงทะเบียนเพื่อสร้างการสนทนาที่มีเนื้อหาที่เทียบเท่ากัน แต่อยู่ในริจิสเตอร์ที่แตกต่างกัน การรับรู้ของผู้ใช้ได้รับการวิเคราะห์เกี่ยวกับ

รูปแบบต่างๆ ในการลงทะเบียนเพื่อแสดงความสัมพันธ์ระหว่างรูปแบบภาษาและการรับรู้ของผู้ใช้ สรุปได้ว่าการให้ความสำคัญกับการลงทะเบียนการสนทนาที่เหมาะสม เป็นปัจจัยสำคัญสำหรับคุณภาพของการสนทนาของแชทบอทที่รับรู้ และ ดังนั้นจึงมีความสำคัญต่อความสำเร็จของแชทบอทในอนาคต การศึกษาแสดงให้เห็นว่าพื้นฐานทางทฤษฎีของการวิเคราะห์การลงทะเบียนที่แนะนำในบทความนี้สามารถเป็นเครื่องมือที่มีประสิทธิภาพในการระบุลักษณะของการลงทะเบียนการสนทนาที่ใช้ในโดเมนเป้าหมายอื่น ๆ และสามารถเปิดเผยคุณลักษณะทางภาษาเฉพาะอย่างเป็น

1.4 งานวิจัยของ Lamya Benaddi, Charaf Ouaddi, Abdeslam Jakimi, Brahim Ouchao เรื่อง A Systematic Review of Chatbots: Classification, Development, and Their Impact on Tourism [4]

นำเสนอแชทบอทที่มีการพัฒนาอย่างรวดเร็วและแพร่กระจายไปทั่วทุกสาขาของการท่องเที่ยว แสดงให้เห็นถึงวิวัฒนาการที่เห็นได้ชัดและการใช้งานที่เพิ่มขึ้นในเว็บไซต์ แอปมือถือ และแอปโซเชียลเน็ตเวิร์ก วิจัยนี้ระบุคำถามการวิจัยเกี่ยวกับแง่มุมต่างๆ ของความคืบหน้าของแชทบอท รวมถึงการจำแนกประเภท สถาปัตยกรรม เครื่องมือการพัฒนา การใช้หลัก และผลกระทบต่อภาคการท่องเที่ยว และนำเสนอภาพรวมที่ครอบคลุมของแชทบอทแนะนำการจำแนกประเภทใหม่ตามเกณฑ์เฉพาะ สำรวจสถาปัตยกรรม แนวคิด และองค์ประกอบสำคัญ ประเมินเครื่องมือการพัฒนาที่มีอยู่ และตรวจสอบการรวมของแชทบอทในภาคการท่องเที่ยวในช่วงทศวรรษที่ผ่านมา จากการวิเคราะห์ผลกระทบของแชทบอทต่อฟังก์ชันต่างๆ ความสำคัญของแชทบอทในการปฏิวัติอุตสาหกรรมการท่องเที่ยว

## 2. ทฤษฎีที่เกี่ยวข้อง

### 2.1 ความหมายและความสำคัญของแชทบอท

แชทบอท คือ เป็นการโต้ตอบกลับโดยอัตโนมัติ โดยจำลองการสนทนาที่เป็นลายลักษณ์อักษรหรือคำพูดของมนุษย์ แชทบอทบางตัวนั้นเป็นโปรแกรมพื้นฐาน โดยเสนอตัวเลือกเมนูง่ายๆ ให้ผู้ใช้คลิกอย่างไรก็ตามแชทบอทขั้นสูงสามารถใช้ประโยชน์จากปัญญาประดิษฐ์ (Artificial Intelligence : AI) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing : NLP) เพื่อทำความเข้าใจข้อมูลของผู้ใช้และนำทางการสนทนาที่ซับซ้อนของมนุษย์ได้อย่างง่ายดาย

### 2.2 ประเภทของแชทบอท

(1) เมนูหรือแชทบอทตามปุ่ม (Menu or button-based chatbots) แชทบอทชนิดนี้เลือกคำถามจากเมนูที่มีให้ โดยการคลิกที่ปุ่มตัวเลือกจากเมนูที่มีสคริปต์ซึ่งตรงกับความต้องการได้ดีที่สุด ขึ้นอยู่กับสิ่งที่ผู้ใช้คลิก และตอบกลับตามข้อความตามรูปแบบที่ถูกกำหนดไว้

(2) แชทบอทตามกฎ (Rules-based chatbots) แชทบอทประเภทนี้ดำเนินการโดยใช้การตรวจจับคำหลักพื้นฐาน แชทบอทตามกฎทำหน้าที่เป็นคำถามที่พบบ่อยซึ่งได้ตอบโดยพื้นฐานแล้วโปรแกรมออกแบบการสนทนาจะกำหนดชุดตัวเลือกคำถามและคำตอบไว้ล่วงหน้าเพื่อให้แชทบอทสามารถเข้าใจข้อมูลของผู้ใช้และตอบสนองได้อย่างถูกต้อง

(3) แชทบอทตามคีย์เวิร์ด (Keyword recognition-based chatbots) แชทบอทประเภทนี้จะยืดหยุ่นมากขึ้น เนื่องจากมีทั้งปุ่มให้กด และสามารถพิมพ์แชทเข้ามาได้ด้วย โดยแชทบอทจะมีความพิเศษขึ้นคือจะตอบสนองตามคำ

หรือคีย์เวิร์ดที่ตั้งไว้ เน้นการใช้ภาษาธรรมชาติจึงทำให้แชทบอทนี้เหมาะกับการใช้ตอบคำถามทั่วไป โดยแชทบอทจะตอบสนองตามคำที่เจอ แต่ถ้าเจอคำถามแบบซับซ้อนก็อาจจะทำให้แชทบอทสับสนได้

(4) แชทบอทที่ขับเคลื่อนด้วยปัญญาประดิษฐ์ (AI-powered chatbots) แชทบอทที่ขับเคลื่อนด้วยปัญญาประดิษฐ์สามารถเข้าใจคำถามและตอบสนองได้หลากหลาย ด้วยการใช้การประมวลผลภาษาธรรมชาติและอัลกอริทึมการเรียนรู้เชิงลึกแชทบอทสามารถจดจำข้อมูลการสนทนา และปรับปรุงประสิทธิภาพของตนเองตลอดเวลา

(5) แชทบอทเสียง (Voice chatbots) แชทบอทเสียงที่ใช้ปัญญาประดิษฐ์ สามารถทำให้การสื่อสารกับผู้ใช้เป็นไปอย่างรวดเร็วและสะดวก เพิ่มประสิทธิภาพในการแก้ไขปัญหาและลดเวลารอ การใช้เทคโนโลยีการประมวลผลภาษาธรรมชาติและการบูรณาการกับเทคโนโลยีคำพูดช่วยให้แชทบอทเสียงเข้าใจและตอบสนองตามความต้องการของผู้ใช้ได้ดีขึ้น

(6) แชทบอทปัญญาประดิษฐ์เจนเนอเรชัน (Generative AI chatbots) แชทบอทที่ใช้การเรียนรู้ขั้นสูง มีความซับซ้อนมากขึ้นกว่าประเภทอื่นโดยมีการใช้การเรียนรู้เครื่องจักรและเอไอ มาพัฒนาแชทบอทให้สามารถจดจำการสนทนาและการเรียนรู้จากคำถามที่เจอเพื่อนำมาพัฒนาเพิ่มประสิทธิภาพในการตอบคำถามครั้งต่อไป [5]

## 2.3 หลักการทำงานของแชทบอท

แชทบอท เป็นโปรแกรมคอมพิวเตอร์ ที่สามารถสื่อสารกับมนุษย์ได้ผ่านทางอินเทอร์เน็ตเพชการแชท หรือจากการส่งข้อความเสียง โดยสามารถตั้งโปรแกรมให้ทำงานตามทริกเกอร์และอัลกอริทึมเฉพาะ ทำให้มีความสามารถในการโต้ตอบและปรับตัวตามบทสนทนาของมนุษย์ แชทบอทสามารถให้คำตอบตามกฎที่ได้รับการโปรแกรมล่วงหน้า และสามารถใช้ฐานความรู้ การจับคู่รูปแบบ การเรียนรู้เชิงลึก และประมวลผลภาษาธรรมชาติ เพื่อปรับปรุงประสิทธิภาพของการตอบสนอง ในขณะที่แชทบอทที่มีศักยภาพมากขึ้นอาจมีการเรียนรู้เชิงลึกเพื่อปรับปรุงการตอบสนองในระยะยาว บางบอทยังสามารถทำงานตามคำสั่งหรือคำขอที่ไม่ได้เป็นคำถามเท่านั้น เช่น การสั่งซื้อสินค้า การตั้งค่า หรือการให้ข้อมูล

โปรแกรมแชทบอทที่ทำงานด้วยปัญญาประดิษฐ์ มักใช้เทคนิคการเรียนรู้เชิงลึก เพื่อปรับปรุงความสามารถในการตอบสนองต่อบทสนทนาที่ซับซ้อน และยังสามารถเรียนรู้จากประสบการณ์และข้อมูลใหม่ ทำให้มีความเข้าใจและสามารถตอบสนองกับสถานการณ์หลากหลายได้ การพัฒนาแชทบอทด้วยปัญญาประดิษฐ์ ต้องใช้การวิเคราะห์ข้อมูลและการประมวลผลภาษาธรรมชาติที่ดีเพื่อให้บอทสามารถเข้าใจความหมายและบทบาทของคำศัพท์ในบทสนทนา นอกจากนี้ การทดสอบและปรับปรุงต่อโมเดลปัญญาประดิษฐ์ ในขณะที่ทำงานจะเป็นส่วนสำคัญเพื่อให้แชทบอทมีประสิทธิภาพและความถูกต้อง [6]

## 2.4 ไลน์บอท (LINE Bot) และ เอพีไอการส่งข้อความของไลน์ (Messaging API)

ไลน์บอท เป็นเครื่องมือสำหรับนักวางแผนหรือนักออกแบบในการจำลองสถานการณ์แชทบอท โดยไม่ต้องมีการสนับสนุนจากนักพัฒนา ข้อความทุกประเภทที่มีอยู่ในแอปไลน์พร้อมใช้งานเมื่อสร้างสถานการณ์เหล่านี้ไลน์บอทจะสร้างโค้ดโดยอัตโนมัติตามสถานการณ์ที่คุณออกแบบ ซึ่งสามารถนำมาใช้ในการพัฒนาแชทบอทได้ในภายหลัง [7]

ไลน์เอพีไอ เปิดโอกาสให้นักพัฒนาสร้างบริการที่ตอบโต้กับทุกรูปแบบธุรกิจ เนื่องจากไลน์เป็นช่องทางสื่อสารหลักของคนไทยและมีการปฏิสัมพันธ์กับธุรกิจและแบรนด์ การใช้ไลน์เอพีไอ ไม่ต้องให้ผู้ใช้งานโหลดแอปเพิ่มเติม เพียง

กตเพิ่มเพื่อนก็สามารถเข้าถึงบริการต่างๆ ที่ใช้แชทบอทได้ นี่ทำให้มีการเรียนรู้ที่น้อยลงและผู้ใช้งานสามารถสื่อสารและทำธุรกิจได้โดยสะดวก และทำให้ไลน์เป็นส่วนสำคัญของการเชื่อมต่อทางธุรกิจและการสื่อสารของคนไทยในปัจจุบัน [8]

## 2.5 การประมวลผลภาษาธรรมชาติ

การประมวลผลภาษาธรรมชาติ หมายถึงสาขาหนึ่งของปัญญาประดิษฐ์ที่ช่วยให้เครื่องจักรสามารถอ่าน ทำความเข้าใจ และสืบความหมายจากภาษาของมนุษย์ได้ การประมวลผลภาษาธรรมชาติ ผสมผสานสาขาภาษาศาสตร์และวิทยาการคอมพิวเตอร์เพื่อถอดรหัสโครงสร้างและแนวปฏิบัติของภาษา และเพื่อสร้างแบบจำลองที่สามารถเข้าใจ แจกแจง และแยกรายละเอียดที่สำคัญออกจากข้อความและคำพูด [9]

## 2.6 เว็สแคปปิง (Web Scraping)

เว็บสแคปปิง การดึงข้อมูลเว็บไซต์เพื่อดึงข้อมูลจากหน้าเว็บโดยอัตโนมัติ การดึงข้อมูลเว็บไซต์ได้เพื่อแปลงข้อมูลในโครงสร้างเอชทีเอ็มแอลที่ซับซ้อนให้เป็นรูปแบบที่มีโครงสร้าง เช่น สเปนซ์ซีตหรือฐานข้อมูล และเว็บสแคปปิงมีประโยชน์อย่างมากสำหรับการรวบรวมข้อมูลจากเว็บนั้น ๆ เช่น การดึงข้อมูลราคาสินค้าจากเว็บไซต์ ดึงข้อมูลผู้ติดต่อจากเว็บไซต์ต่าง ๆ [10]

## 2.7 ฐานข้อมูล (Database)

ฐานข้อมูล คือการรวบรวมข้อมูลที่มีโครงสร้างหรือข้อมูลที่จัดระเบียบโดยทั่วไปจะถูกเก็บไว้ในระบบอิเล็กทรอนิกส์ภายในคอมพิวเตอร์ การจัดเก็บและการเข้าถึงข้อมูลนั้นทำได้ด้วยการใช้ระบบจัดการฐานข้อมูล (Database Management System : DBMS) ซึ่งมีหน้าที่ทำให้การจัดการข้อมูลเป็นไปอย่างระบบและมีประสิทธิภาพ ระบบจัดการฐานข้อมูลช่วยให้ผู้ใช้สามารถทำงานกับข้อมูล ดึงข้อมูล แก้ไข ลบ และจัดการความปลอดภัยของข้อมูลได้ง่ายขึ้น นอกจากนี้ระบบจัดการฐานข้อมูล ยังมีบทบาทในการควบคุมการเข้าถึงข้อมูล การจัดการทรัพยากรร่วมกัน และการปรับปรุงประสิทธิภาพของระบบฐานข้อมูล [11]



## บทที่ 3

### วิธีการดำเนินงาน

#### 1. การกำหนดขอบเขตและเป้าหมาย

1.1 กำหนดจุดประสงค์ในการทำโครงการสร้างแชทบอทเพื่อให้สามารถตอบคำถามและให้ข้อมูลของสถานที่ท่องเที่ยวในจังหวัดขอนแก่นผ่านแอปพลิเคชันไลน์ และรวบรวมคำถามที่เกี่ยวข้องกับการท่องเที่ยวในจังหวัดขอนแก่น เพื่อให้ข้อมูลที่เป็นประโยชน์และตรงตามความต้องการของผู้ใช้งาน

1.2 กำหนดขอบเขตและเป้าหมายในการพัฒนาแชทบอทที่สามารถใช้งานได้บนแอปพลิเคชันไลน์ และตอบคำถามเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่นอย่างถูกต้องและรวดเร็ว จะช่วยให้การทำงานของบอทเป็นไปตามความต้องการ โดยการให้ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยว เวลาเปิด-ปิด ค่าเข้าชม รูปภาพ แผนที่ และร้านอาหาร มิชลินไกด์ เป็นต้น

#### 2. ศึกษาข้อมูลและวิธีใช้งานเครื่องมือต่างๆ ที่ใช้งานวิจัย

##### 2.1 ไดอะล็อกโฟลว์ (Dialog Flow)

ไดอะล็อกโฟลว์ คือ ตัวแทนเสมือนที่จัดการการสนทนาที่เกิดขึ้นพร้อมกันกับผู้ใช้ปลายทาง เป็นโมดูลความเข้าใจภาษาธรรมชาติที่เข้าใจถึงความแตกต่างของภาษามนุษย์ ทำหน้าที่แปลงข้อความหรือเสียงที่ผู้ใช้ป้อนเข้ามาระหว่างการสนทนาเป็นข้อมูลที่มีโครงสร้าง ซึ่งแอปและบริการที่สามารถเข้าใจมนุษย์ได้ และสามารถออกแบบและสร้างตัวแทน เพื่อจัดการกับประเภทการสนทนาที่เกี่ยวข้องกับระบบ

ซึ่งองค์ประกอบของไดอะล็อกโฟลว์ มีดังนี้

2.1.1 เอเจนต์ (Agents) อเจนต์เสมือนที่จัดการการสนทนาที่เกิดขึ้นพร้อมกันกับผู้ใช้ปลายทาง คือโมดูลความเข้าใจภาษาธรรมชาติที่ถูกนำมาใช้ในการแปลงข้อความหรือเสียงที่ผู้ใช้ป้อนเข้ามาระหว่างการสนทนาเป็นข้อมูลที่มีโครงสร้างไดอะล็อกโฟลว์ ทำหน้าที่แปลงข้อความหรือเสียงจากผู้ใช้เพื่อให้ข้อมูลที่สามารถเข้าใจได้ตามโครงสร้างที่กำหนดไว้

2.1.2 อินเทนท (Intents) คือส่วนสำคัญที่ช่วยในการระบุความต้องการหรือเจตนาที่ผู้ใช้มีเมื่อทำการสื่อสารกับบอทหรือแชทบอท นั่นคืออินเทนท ในไดอะล็อกโฟลว์ทำหน้าที่เป็นตัวกลางที่ช่วยแปลงคำถามหรือข้อความที่ผู้ใช้ป้อนเข้ามาเป็นคำสั่งหรือการทำงานที่บอทควรทำต่อไป

2.1.3 พารามิเตอร์ (Parameters) ใช้ในการบันทึกและอ้างอิงค่าที่ผู้ใช้ปลายทางระบุไว้ในระหว่างเซสชัน พารามิเตอร์แต่ละตัวมีชื่อและ ประเภทเอนทิตี พารามิเตอร์เป็นข้อมูลที่มีโครงสร้างซึ่งต่างจากอินพุตของผู้ใช้ปลายทางแบบดิบตรงที่สามารถใช้เพื่อดำเนินการตรรกะบางอย่างหรือสร้างการตอบสนองได้อย่างง่ายดาย

2.1.4 เอนทิตี (Entities) ในส่วนของพารามิเตอร์จะมีชนิดของข้อมูล หรือ เอนทิตีที่ไต่ปัวด้วย ซึ่งจะทำให้ตัวไดอะล็อกโฟลว์ รู้ว่าจะต้องดึงข้อมูลมาจากผู้ใช้อย่างไร ในส่วนของข้อความที่ส่งมาให้ สำหรับประเภทของข้อมูลนั้นก็จะมีทั้งของระบบ หรือเราจะกำหนดขึ้นมาเองก็ได้

2.1.5 การปฏิบัติตาม (Fulfillments) มีหน้าที่ในการตอบสนองกับผู้ใช้ในรอบของการสนทนา โดยต้องมีการตอบกับผู้ใช้ด้วยคำตอบที่เป็นการตอบคำถาม ขอข้อมูล หรือการสิ้นสุดเซสชัน แอปเจนต์อาจต้องติดต่อบริการเพื่อสร้างคำตอบแบบไดนามิกหรือดำเนินการต่างๆในรอบของการสนทนา [12]

## 2.2 ซีเอสเอส (CascadingStyleSheet : CSS)

ซีเอสเอส คือ ภาษาใช้สำหรับการจัดรูปแบบการแสดงผลขององค์ประกอบของภาษาเอชทีเอ็มแอลบนหน้าเว็บ โดยกำหนดการแสดงผลการจัดวางอักษร พื้นหลัง สี แบบอักษร และขนาดตัวอักษร ทำให้เพิ่มความสวยงาม และง่ายต่อการควบคุมรูปแบบการแสดงผลของภาษาเอชทีเอ็มแอล [13]

## 2.3 จาวาสคริปต์ (JavaScript)

จาวาสคริปต์ คือ ภาษาโปรแกรมเพื่อเพิ่มการโต้ตอบ พฤติกรรมแบบไดนามิกให้กับหน้าเว็บ ให้ทำงานในเว็บเบราว์เซอร์ ช่วยให้นักพัฒนาสามารถสร้างคุณสมบัติต่างๆ เช่น การโต้ตอบของผู้ใช้ ภาพเคลื่อนไหว การตรวจสอบแบบฟอร์ม และการอัปเดตแบบเรียลไทม์ โดยไม่ต้องมีการโหลดซ้ำ จาวาสคริปต์มีการโต้ตอบกับโมเดลวัตถุเอกสาร (Document Object Model : DOM) เพื่อจัดการองค์ประกอบบนหน้าเว็บ ให้ตอบสนองต่อการกระทำของผู้ใช้งาน และสามารถเข้าถึงฟังก์ชันการทำงานของเบราว์เซอร์ [14]

## 2.4 เอ็กซ์เพรสเจส (Express.js)

เอ็กซ์เพรสเจส คือ เฟรมเวิร์กสำหรับโหนดเจส (Node.js) ที่ช่วยให้การพัฒนาเว็บแอปพลิเคชันและเอพีไอ (API) ง่ายและรวดเร็วขึ้น โดยมีคุณสมบัติในการกำหนดเส้นทางของระบบ (Routing) การจัดการคำขอและการตอบสนอง (Request handling and Response) การรับส่งข้อมูลของระบบ (Middleware) ไปจนถึงการจัดการสิ่งต่างๆ ไม่ว่าจะเป็นการเข้าถึงฐานข้อมูล การจัดการกับข้อผิดพลาด เอ็กซ์เพรสเจสมีการออกแบบที่ยืดหยุ่นและสามารถปรับแต่งได้ ทำให้นักพัฒนาสามารถสร้างแอปพลิเคชันได้ตามความต้องการ [15]

## 2.5 วิว (Vue)

วิว คือ เป็นเฟรมเวิร์คจาวาสคริปต์ที่ได้รับความนิยมอย่างมากในปัจจุบัน มีจุดเด่นคือใช้งานง่าย เรียนรู้ได้เร็ว และมีความยืดหยุ่นสูง สามารถนำไปใช้พัฒนาส่วนติดต่อผู้ใช้ (User Interface) ของเว็บแอปพลิเคชันได้หลากหลายรูปแบบ ตั้งแต่เว็บไซต์ขนาดเล็กไปจนถึงแอปพลิเคชันขนาดใหญ่ที่มีความซับซ้อน และยังสามารถทำงานร่วมกับไลบรารีและเฟรมเวิร์กอื่น ๆ [16]

## 2.6 เทลวินด์ ซีเอสเอส (Tailwind CSS)

เทลวินด์ ซีเอสเอส คือ เป็นเฟรมเวิร์คซีเอสเอสที่มีคลาสสำเร็จรูปที่ถูกกำหนดไว้ในการตกแต่งองค์ประกอบต่างๆ ซึ่งแต่ละคลาสจะทำหน้าที่ในการปรับแต่งสไตล์ซีเอสเอสอย่างใดอย่างหนึ่ง เช่น สี ขนาดตัวอักษร ระยะขอบ หรือเงา โดยไม่จำเป็นต้องเขียนซีเอสเอสเอง สามารถปรับแต่งแก้ไขคลาสได้ตามต้องการ ซึ่งช่วยลดเวลาและความซับซ้อนในการออกแบบและพัฒนาเว็บไซต์ [17]

### 3. การเก็บรวบรวมข้อมูล

ผู้จัดทำการศึกษาและรวบรวมความต้องการของผู้ใช้งาน โดยเก็บรวบรวมข้อมูลดังนี้

#### 3.1 การเก็บข้อมูลปฐมภูมิ (Primary Data Collection)

3.1.1 แบบสอบถามออนไลน์ สร้างแบบสอบถามออนไลน์เพื่อเก็บรวบรวมความคิดเห็นและข้อมูลจากนักท่องเที่ยวและผู้สนใจท่องเที่ยวในจังหวัดขอนแก่น เพื่อทราบถึงความต้องการและคำถามที่พบบ่อย เช่น สถานที่ท่องเที่ยวที่น่าสนใจ กิจกรรมที่ต้องการทำ ข้อมูลที่ต้องการทราบเพิ่มเติม

3.1.2 สืบหาข้อมูลจากเว็บไซต์พันทิป (Pantip) สืบหาข้อมูลจากกระทู้ต่าง ๆ ในเว็บไซต์พันทิปเพื่อหาความต้องการของนักท่องเที่ยวว่าต้องการทราบเรื่องใดบ้างและคำถามเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่น

#### 3.1.3 โซเชียลมีเดีย (Social Media)

(1) Facebook สืบหาความคิดเห็นและคำถามในกลุ่มหรือเพจที่เกี่ยวข้องกับการท่องเที่ยวในจังหวัดขอนแก่น

(2) Twitter ค้นหาคำถามและความคิดเห็นโดยใช้แฮชแท็กที่เกี่ยวข้องกับการท่องเที่ยวในขอนแก่น

(3) Instagram อ่านความคิดเห็นและคำถามที่อยู่ในคอมเมนต์ของโพสต์ที่เกี่ยวข้องกับสถานที่ท่องเที่ยวในขอนแก่น

3.1.4 เว็บไซต์รีวิว (Review Sites) : เว็บไซต์รีวิวได้แก่ TripAdvisor Wongnai Lemon8 True ID Foursquare สามารถอ่านรีวิวและคำถามจากนักท่องเที่ยวเกี่ยวกับสถานที่ท่องเที่ยวในขอนแก่น

#### 3.2 การเก็บข้อมูลทุติยภูมิ (Secondary Data Collection)

3.2.1 ศึกษาบทความวิจัย รายงานวิจัย และเอกสารที่เกี่ยวข้องกับเซทบอทและการท่องเที่ยวในจังหวัดขอนแก่น เพื่อทำความเข้าใจเกี่ยวกับทฤษฎีและแนวทางการพัฒนาเซทบอท

3.2.2 ค้นคว้าและศึกษาทฤษฎีที่เกี่ยวข้องศึกษาหลักการและทฤษฎีต่าง ๆ เพื่อนำมาพัฒนาเซทบอท และนำเทคโนโลยีที่ศึกษามาประยุกต์ใช้ เช่น การประมวลผลภาษาธรรมชาติ ไลน์บอท ไดอะล็อกโฟลว์ เอ็กชัฟเฟอเรนซ์

3.2.3 รวบรวมข้อมูลสถานที่ท่องเที่ยวต่าง ๆ ในจังหวัดขอนแก่น เช่น รายชื่อสถานที่ท่องเที่ยวที่สำคัญ ที่ตั้ง ค่าเข้าชม เวลาเปิด-ปิด และทำการจำแนกประเภทของสถานที่ท่องเที่ยว เช่น ภูเขา สวนสัตว์ อุทยานแห่งชาติ วัด พิพิธภัณฑ์ สวนน้ำสวนสาธารณะ แหล่งช้อปปิ้ง ร้านอาหารมิชลินไกด์ และจำแนกประเภทตามอำเภอข้อมูลการจำแนกประเภทอำเภอ ได้แก่ อำเภอเมือง อำเภอน้ำพอง อำเภออุบลรัตน์ อำเภอสหัสขันธ์ อำเภอภูเวียง อำเภอหนองเรือ อำเภอชุมแพ อำเภอเวียงเก่า อำเภอบ้านฝาง อำเภอเขาสนวนกวาง อำเภอเปือยน้อย อำเภอกระนวน เป็นต้น

3.2.4 รวบรวมข้อมูลจากเว็บไซต์ท่องเที่ยวต่าง ๆ เช่น รีวิวจากนักท่องเที่ยว บทความเกี่ยวกับสถานที่ท่องเที่ยว เพื่อเข้าใจถึงประสบการณ์และความพึงพอใจของนักท่องเที่ยว

## บทที่ 4

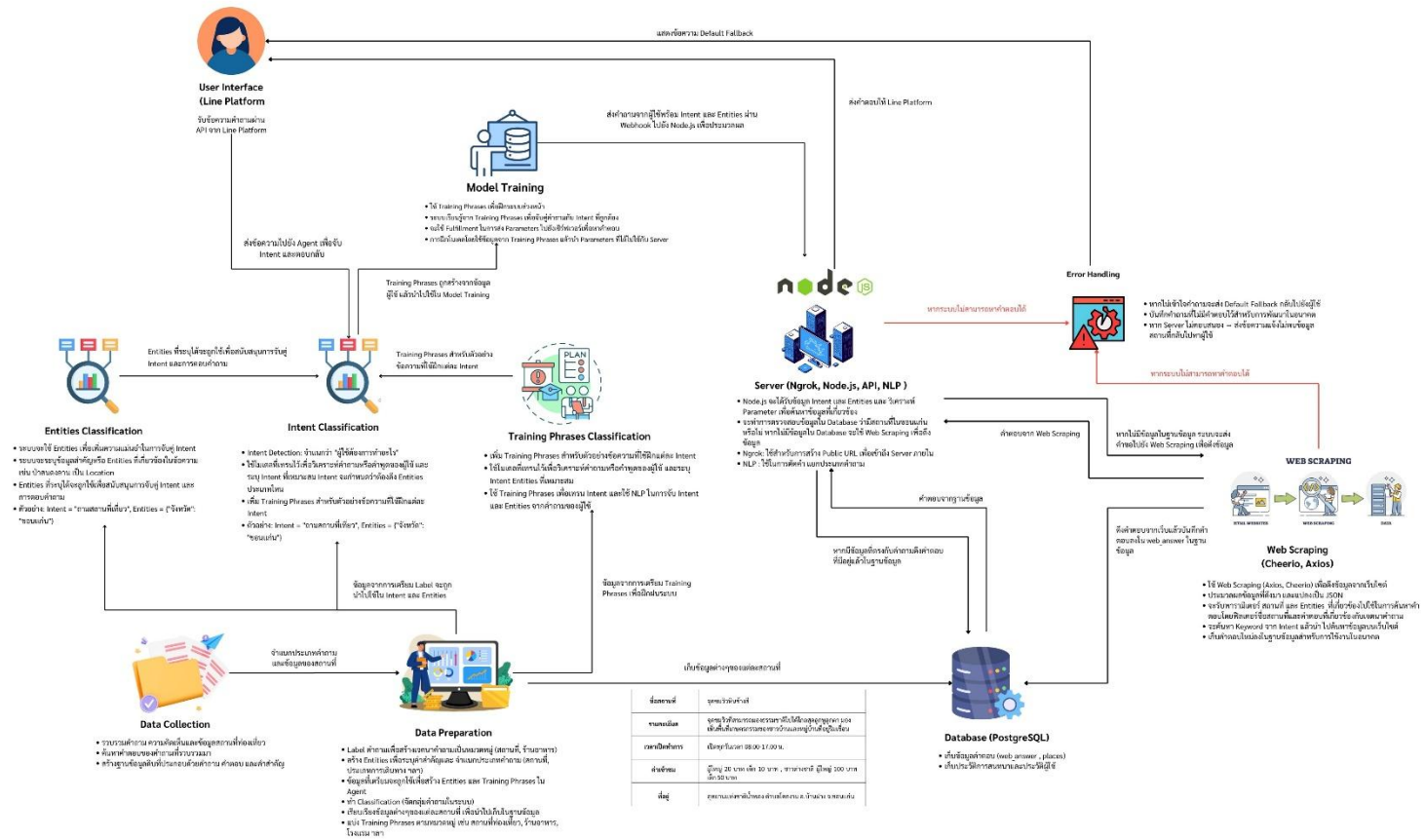
### การวิเคราะห์ระบบ และการพัฒนา

งานวิจัยในครั้งนี้ผู้จัดทำได้ดำเนินการวิเคราะห์และออกแบบการใช้แพลตฟอร์มแอปพลิเคชันไลน์มาประยุกต์ใช้ และพัฒนาแชทบอท เพื่อการท่องเที่ยวในจังหวัดขอนแก่น โดยผู้วิจัยได้วางแผนการดำเนินการ ดังนี้

1. การวิเคราะห์ระบบ
2. การออกแบบแชทบอท
3. การพัฒนาแชทบอท
4. การทดสอบแชทบอท
5. การประเมินความพึงพอใจของผู้ใช้

#### 1. การวิเคราะห์ระบบ

- 1.1 สถาปัตยกรรมของระบบ (System architecture) ของแชทบอทการท่องเที่ยวในจังหวัดขอนแก่น



ภาพที่ 1 แผนภาพสถาปัตยกรรม

จากภาพที่ 1 แชนบอทมีกระบวนการทำงาน ดังนี้

1.1.1 กระบวนการเก็บรวบรวมข้อมูล (Data Collection) รวบรวมข้อมูลที่เกี่ยวข้องกับคำถามหรือคำขอของผู้ใช้ เช่น คำถามเกี่ยวกับสถานที่ท่องเที่ยวขอนแก่น รวบรวมข้อมูลคำตอบเกี่ยวกับสถานที่ท่องเที่ยว ได้แก่ รายละเอียดสถานที่ท่องเที่ยว ค่าเข้าชม เวลาทำการ เป็นต้น เพื่อให้ระบบสามารถตอบคำถามได้อย่างถูกต้องและแม่นยำ

1.1.2 การเตรียมข้อมูล (Data Preparation) การเตรียมข้อมูลให้พร้อมใช้งาน (Preprocessing) ข้อมูลที่ได้จาก Data Collection มักยังไม่พร้อมใช้งาน จำเป็นต้องจัดการ เช่น ล้างข้อมูล (Cleaning) ติดป้ายกำกับ (Labeling) และจัดโครงสร้างข้อมูล สร้างชุดข้อมูลตัวอย่าง (Training Data) ที่เหมาะสมสำหรับการฝึกโมเดล เพื่อให้ระบบเข้าใจคำถามของผู้ใช้ แยกประเภทคำถามแล้วระบุเจตนาของคำถามให้ครอบคลุม ได้แก่ คำถามเกี่ยวกับค่าเข้าชม จะถูกสร้างไว้ที่เจตนาของค่าเข้าชม และแยกประเภทข้อมูล เช่น ชื่อสถานที่ เวลา หรือสถานที่ เพื่อให้ระบบเข้าใจเนื้อหาและบริบทของข้อความ

#### 1.1.3 ผู้ใช้งาน (User)

ผู้ใช้งานที่ต้องการสอบถามข้อมูลเกี่ยวกับการท่องเที่ยวในจังหวัดขอนแก่นผ่านแอปพลิเคชันไลน์ โดยส่งข้อความคำถามไปยังแชทบอท

#### 1.1.4 แอปพลิเคชันไลน์ (LINE)

เป็นแพลตฟอร์มการสื่อสารหลักที่ผู้ใช้จะพิมพ์ข้อความคำถามไปยังแชทบอท การสื่อสารจะถูกส่งไปยังเซิร์ฟเวอร์ผ่านทาง API ที่เชื่อมต่อระหว่างไลน์ และระบบ

LINE Messaging API จะรับข้อความจากผู้ใช้และส่งข้อมูลนั้นต่อไปยังเซิร์ฟเวอร์ (ในที่นี้ใช้โหนดเจเอส) ซึ่งเป็นตัวกลางในการประมวลผลคำถาม

LINE Developer เป็นแพลตฟอร์มที่ใช้สำหรับพัฒนาบอทในไลน์เมื่อได้รับข้อความจากผู้ใช้ ระบบจะส่งข้อความไปยัง Webhook (เชื่อมต่อกับ Node.js) เพื่อทำการประมวลผลต่อไป

#### 1.1.5 โหนดเจเอส (Node.js)

เป็นเทคโนโลยีเซิร์ฟเวอร์ฝั่งหลังบ้าน (Backend Server) ที่ทำหน้าที่รับข้อความจาก LINE API ผ่าน Webhook (เป็นจุดเชื่อมต่อระหว่างไลน์และระบบ) และเชื่อมต่อไปยัง ไดอะล็อกโพล์ เพื่อประมวลผลการวิเคราะห์ความหมายของข้อความ

จะทำการสร้าง Webhook เพื่อรับข้อมูลจากไลน์และส่งต่อข้อมูลไปยัง ไดอะล็อกโพล์ เพื่อวิเคราะห์เจตนาของคำถามที่ได้รับ ระบบจะรับพารามิเตอร์จากไดอะล็อกโพล์มาเพื่อประมวลผลคำตอบ ใช้ Ngrok สำหรับการสร้าง Public URL เพื่อเข้าถึง Server ภายใน

ใช้หลักการของการประมวลผลภาษาธรรมชาติ Natural Language Processing (NLP) เป็นเทคโนโลยีที่ใช้ในการแยกคำ วิเคราะห์ความหมาย และตรวจสอบข้อมูลที่เกี่ยวข้องในข้อความของผู้ใช้ NLP จะช่วยในการตัดคำแยกคำ และจัดหมวดหมู่เพื่อส่งต่อไปยังระบบต่าง ๆ ในระบบโดยใช้กระบวนการดังนี้

Tokenization (การแบ่งคำ) TF-IDF (Term Frequency-Inverse Document Frequency) การหาค่า TF-IDF ของคำในเอกสาร Named Entity Recognition (NER) การตรวจจับและจำแนกชื่อสิ่งต่าง ๆ ในข้อความ และ Regular Expressions (RegEx) ใช้สำหรับการกรองข้อมูลจากคำตอบ

การประมวลผล NLP (Natural Language Processing) เพื่อดึงคำสำคัญจากคำถามที่ผู้ใช้ส่งมาและจับคู่กับคำตอบที่เหมาะสมในฐานข้อมูลหรือเรียกใช้งานเว็บ scraping หากข้อมูลไม่อยู่ในฐานข้อมูล ในกรณีที่ไม่สามารถระบุ Intent ได้ ระบบอาจเลือกใช้ Intent เริ่มต้น (Fallback Intent)

#### 1.1.6 ไดอะล็อกโฟลว์ (Dialogflow)

เป็นแพลตฟอร์มการประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP) ที่ทำหน้าที่วิเคราะห์ข้อความที่ผู้ใช้ส่งมา และตรวจสอบว่าเจตนาของผู้ใช้คืออะไร (Intent Matching)

เอเจนต์ (Agent) : ตัวแทนเสมือนของระบบที่ทำหน้าที่ประมวลผลการสื่อสารที่เกิดขึ้นระหว่างผู้ใช้และเซิร์ฟเวอร์ โดย Agent จะประกอบด้วย อินเทนส์ (เจตนา) ที่มีการกำหนดไว้ล่วงหน้า เช่น คำถามเกี่ยวกับสถานที่ท่องเที่ยว พยากรณ์อากาศจังหวัดขอนแก่น อีเว้นประจำวัน โปรแกรมการท่องเที่ยว ฯลฯ และทำหน้าที่จัดการ Intent และ Entity เป็นเหมือนสมองของระบบที่ใช้วิเคราะห์คำถามจากผู้ใช้

อินเทนส์: เป็นหมวดหมู่ของคำถามที่ถูกจัดทำได้ โดยแต่ละ Intent จะมีตัวอย่างคำถาม (Training Phrases) ที่เป็นไปได้ เช่น “มีสถานที่ท่องเที่ยวไหนแนะนำบ้าง” โดย ไดอะล็อกโฟลว์ จะทำการจับคู่ข้อความที่ได้รับจากผู้ใช้กับ อินเทนส์ ที่ตรงกัน โดยระบบจะทำการสร้างอินเทนส์ไว้ทั้งหมด 30 อินเทนส์ จะแบ่งออกตามประเภทสถานที่ อาทิ ร้านอาหาร พยากรณ์อากาศ รายละเอียดสถานที่ ค่าธรรมเนียมเข้าชม เวลาเปิดทำการ โปรแกรมการท่องเที่ยว และแบ่งตามชื่อร้านอาหาร เป็นต้น

เอนทิตี ทำหน้าที่ดึงข้อมูลเฉพาะ เช่น ชื่อสถานที่หรือเวลา จากข้อความที่ผู้ใช้ป้อน กำหนด Entity เช่น สถานที่ (location), เวลา (time) เพื่อช่วยให้ระบบเข้าใจรายละเอียด

Model Training กระบวนการ Training Phrases ที่ผู้พัฒนาป้อนจะถูกนำไปใช้เทรนโมเดล Machine Learning เพื่อให้ระบบเข้าใจภาษาและสามารถจับคู่ Intent ได้แม่นยำขึ้น

Intent Matching ระบบจะวิเคราะห์ข้อความของผู้ใช้และจับคู่กับ Intent ที่เหมาะสมที่สุด หากไม่พบ Intent ที่ตรง ระบบจะใช้ Fallback Intent เพื่อตอบกลับ เช่น "ขอโทษค่ะ ฉันไม่เข้าใจคำถามนี้"

Fulfillment จัดการคำตอบแบบไดนามิก โดยเชื่อมต่อกับระบบภายนอก เช่น API หรือฐานข้อมูล

1.1.7 ดึงคำตอบจากฐานข้อมูล ใช้ Intent Mapping เป็นกระบวนการที่ใช้ในการจับคู่คำถามของผู้ใช้กับอินเทนส์ที่เหมาะสมในระบบ โดยเมื่อผู้ใช้ส่งคำถามผ่านแอปพลิเคชันไลน์ คำถามจะถูกส่งไปยังเซิร์ฟเวอร์ที่ทำงานด้วย Node.js ก่อนที่เซิร์ฟเวอร์จะส่งต่อไปยัง Dialogflow ซึ่งใช้เทคโนโลยี NLP ในการวิเคราะห์ข้อความและจับคู่กับอินเทนส์ที่กำหนดไว้ล่วงหน้า

ทำการจัดกลุ่มคำถามที่มีเจตนาเดียวกัน เช่น คำถามเกี่ยวกับที่พักหรือร้านอาหาร เมื่อตรวจพบอินเทนส์ที่สอดคล้อง ระบบจะส่งพารามิเตอร์ เช่น เอนทิตีและอินเทนส์ ไปค้นหาคำตอบในฐานข้อมูล โดยจะเริ่มจากฐานข้อมูลที่เพิ่มเองในระบบ หากไม่พบคำตอบ จะค้นหาต่อในฐานข้อมูลที่ดึงข้อมูลจากเว็บไซต์ หลังจากได้คำตอบ ระบบจะจัด

หมวดหมู่ข้อมูล เช่น ค่าเข้า เบอร์โทร หรือเวลาเปิดทำการคำตอบที่ฟิลเตอร์และเตรียมไว้จะถูกส่งกลับไปยัง Node.js เพื่อตอบกลับผู้ใช้ในรูปแบบข้อความหรือรูปภาพที่ปรากฏในแอปพลิเคชันไลน์ ให้ผู้ใช้เข้าถึงข้อมูลได้อย่างสะดวกและรวดเร็ว

1.1.8 เว็บภายนอก (Web Scraping ) เป็นแหล่งข้อมูลเสริมที่เกี่ยวข้องกับสถานที่ท่องเที่ยวในขอนแก่น ซึ่งจะดึงข้อมูลนี้ผ่านการเก็บข้อมูลจากเว็บ (Web Scraping) จะทำหน้าที่ในการดึงและส่งข้อมูลนี้กลับมาเมื่อจำเป็น

เป็นกระบวนการที่ใช้เพื่อดึงข้อมูลจากเว็บไซต์ภายนอก เมื่อไม่มีคำตอบที่ต้องการในฐานข้อมูลของระบบเอง ระบบจะทำการไปหาข้อมูลจากเว็บท่องเที่ยวหรือเว็บอื่น ๆ ที่เกี่ยวข้องโดยใช้เทคนิค Web Scraping ซึ่งจะทำการดึงข้อมูลจากหน้าเว็บและฟิลเตอร์คำตอบเพื่อให้ได้คำตอบที่ตรงกับคำถามของผู้ใช้

ถ้าในกรณีที่ระบบไม่พบคำตอบในฐานข้อมูลหรืออินเทอร์เน็ตที่ฝึกไว้โดนดเจเอส จะทำการเรียกใช้ เทคนิคการดึงข้อมูลจากเว็บไซต์ (Web Scraping) โดยใช้ไลบรารีเช่น Cheerio และ Axios เพื่อตรวจสอบและดึงข้อมูลจากเว็บไซต์ภายนอกที่เกี่ยวข้อง

การประมวลผล: ข้อมูลที่ดึงมาได้จะถูกฟิลเตอร์ผ่านการประมวลผลโดยธรรมชาติและแปลงเป็นข้อมูลที่สามารถตอบกลับไปยังผู้ใช้ได้

1.1.9 ฐานข้อมูล (Database): ใช้ฐานข้อมูล PostgreSQL เก็บข้อมูลคำถาม-คำตอบ ข้อมูลสถานที่ท่องเที่ยว และการโต้ตอบของผู้ใช้งาน ซึ่งฐานข้อมูลนี้เป็นแหล่งข้อมูลหลักในการตอบคำถามของผู้ใช้งาน และบริหารจัดการโดยเซิร์ฟเวอร์ โหนดเจเอส

ฐานข้อมูล: ทำหน้าที่เก็บข้อมูลที่เกี่ยวข้องกับการท่องเที่ยว เช่น ข้อมูลสถานที่ ร้านอาหาร กิจกรรมต่าง ๆ ซึ่งข้อมูลนี้จะถูกใช้เพื่อเป็นคำตอบให้กับผู้ใช้เมื่อมีการสอบถามข้อมูลที่เกี่ยวข้อง

การดึงข้อมูล: เมื่อ ไดอะล็อกโพล์ ตรวจสอบอินเทอร์เน็ตที่ตรงกันกับคำถามผู้ใช้ข้อมูลจะถูกดึงจากฐานข้อมูลและส่งคำตอบกลับไปยังผู้ใช้โดยผ่านโหนดเจเอส

1.1.10 การจัดการข้อผิดพลาด (Error Handling) หาก Intent ไม่ตรง ระบบจะใช้ Default Fallback หาก Server ไม่ตอบสนอง อาจเกิดจากความล้มเหลวทางเทคนิค เช่น API ล่ม

1.1.11 แอดมิน (Admin) จัดการข้อมูลสถานที่และข้อมูลต่าง ๆ ที่จำเป็นในระบบและดูแลและตรวจสอบข้อมูลคำถาม-คำตอบที่ถูกบันทึกในดาต้าเบส

การตอบกลับไปยังผู้ใช้: ข้อมูลคำตอบที่ได้จากฐานข้อมูลหรือจาก เทคนิคการดึงข้อมูลจากเว็บไซต์ (Web Scraping) จะถูกส่งกลับไปยัง โหนดเจเอส เพื่อทำการฟิลเตอร์และจัดรูปแบบคำตอบให้เหมาะสม จากนั้นคำตอบจะถูกส่งกลับไปยังผู้ใช้ผ่านทาง LINE API เพื่อให้ผู้ใช้ได้รับคำตอบที่ต้องการ

สรุปได้ว่า ระบบนี้ออกแบบมาเพื่อให้ผู้ใช้งานสามารถสอบถามเกี่ยวกับสถานที่ท่องเที่ยวในขอนแก่นผ่านไลน์โดยระบบจะใช้ไดอะล็อกโพล์เพื่อวิเคราะห์คำถามและดึงคำตอบจากฐานข้อมูลหรือแหล่งข้อมูลภายนอกเมื่อจำเป็น ขณะเดียวกันยังสามารถเก็บข้อมูลและจัดการโดยแอดมินผ่านโหนดเจเอส



## 1.2 กระบวนการทำงานของแชทบอทสถานที่ท่องเที่ยวขอนแก่น

1.2.1 กระบวนการเก็บรวบรวมข้อมูล (Data Collection) การเก็บรวบรวมข้อมูลเป็นขั้นตอนสำคัญสำหรับการพัฒนาระบบแชทบอท เพื่อให้ระบบสามารถตอบคำถามผู้ใช้ได้อย่างถูกต้องและครอบคลุม ซึ่งในกรณีนี้เป็นข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวในขอนแก่น กำหนดประเภทข้อมูลที่ต้องการรวบรวม กำหนดประเภทข้อมูลที่ต้องการรวบรวม กำหนดแหล่งข้อมูลหลังจากรวบรวมข้อมูล จะต้องจัดเรียงข้อมูลให้อยู่ในรูปแบบที่พร้อมใช้งานแยกประเภทของข้อมูล และ การตรวจสอบความถูกต้อง (Data Validation)

1.2.2 การเตรียมข้อมูล (Data Preparation) เป็นขั้นตอนสำคัญในการพัฒนาระบบแชทบอท เพื่อให้ข้อมูลที่ได้มีคุณภาพสูงสุดและพร้อมสำหรับการนำไปใช้ในการฝึก (Training) ตัว Dialogflow Agent กระบวนการเตรียมข้อมูลในระบบของแชทบอทสถานที่ท่องเที่ยวในขอนแก่น สามารถแบ่งออกเป็นขั้นตอนย่อย ๆ

การรวบรวมข้อมูลพื้นฐาน ข้อมูลที่เกี่ยวข้องกับสถานที่ เช่น ชื่อ, ประเภท, ที่อยู่, เวลาเปิด-ปิด, คำอธิบาย ฯลฯ

การทำ Label ข้อมูล การระบุประเภทของข้อมูล (Labeling) กำหนด "หมวดหมู่" หรือ "ประเภท" ให้กับข้อมูล เพื่อให้ระบบเข้าใจได้ง่าย เช่น: สถานที่ประเภทธรรมชาติ เช่น บึงแก่นนคร, อุทยานแห่งชาติภูเวียง การ Label จะช่วยให้ระบบสามารถแยกประเภทของสถานที่ได้เมื่อผู้ใช้ระบุคำค้นหา เช่น "สถานที่ธรรมชาติ" หรือ "วัด"

การสร้าง Entities บทบาทของ Entities ใช้ระบุคำสำคัญในคำถามของผู้ใช้ เช่น ชื่อสถานที่ ประเภทสถานที่ หรือ คำที่อธิบายลักษณะสถานที่ และ การเตรียม Entities จัดกลุ่มข้อมูลเพื่อสร้าง Entity เช่น Location: ชื่อพื้นที่ เช่น "อำเภอเมือง", "ภูผาม่าน"

การสร้าง Training Phrases ข้อความตัวอย่างที่ผู้ใช้จะถาม เพื่อสอน Dialogflow ให้เข้าใจ Intent เขียนตัวอย่างข้อความที่ครอบคลุมคำถามที่ผู้ใช้ถาม

การสร้าง Intents ใช้เพื่อจับคู่คำถามของผู้ใช้กับการตอบสนองที่เหมาะสม แยก Intents ตามวัตถุประสงค์

1.2.3 ผู้ใช้ส่งข้อความผ่านไลน์ โดยส่งคำถามเกี่ยวกับสถานที่ท่องเที่ยวผ่านแอปพลิเคชันไลน์ ตัวอย่างเช่น ผู้ใช้ถามว่า สถานที่ท่องเที่ยวในขอนแก่นที่น่าสนใจมีอะไรบ้าง ระบบจะทำหน้าที่รับคำถาม หลังจากผู้ใช้ส่งข้อความ ข้อมูลจะถูกส่งไปยัง LINE Developer ผ่าน Messaging API ของไลน์ซึ่งเป็นแพลตฟอร์มที่ใช้สำหรับพัฒนาและสื่อสารกับบอท ในการส่งข้อมูลนี้ ข้อความจะถูกจัดเป็น Event Object ทำให้ โหนดเจเอส รู้ว่าข้อมูลใดต้องการการตอบสนองและข้อมูลใดต้องการส่งต่อไปยังไดอะล็อกโฟลว์หรือฐานข้อมูล ซึ่งประกอบด้วยข้อมูลหลายอย่าง เช่น อดีของผู้ใช้ที่ส่งข้อความ ข้อความที่ผู้ใช้ส่งมา เวลาที่ข้อความถูกส่ง

1.2.4 ผู้ใช้ส่งข้อความผ่านไลน์และข้อมูลจะถูกส่งยัง LINE Developer เป็นแพลตฟอร์มที่ใช้สำหรับพัฒนาบอทในไลน์เมื่อ LINE Developer ได้รับข้อความหรืออีเว้นท์จากผู้ใช้ ข้อมูลนั้นจะถูกส่งไปยัง Webhook URL ซึ่งเป็น URL ที่เซิร์ฟเวอร์ (Node.js) ของเรากำหนดไว้ Webhook URL เป็นจุดเชื่อมต่อที่ทำให้เซิร์ฟเวอร์ของเราสามารถรับและประมวลผลข้อมูลที่ส่งมาจากไลน์ เมื่อได้รับข้อความจากผู้ใช้ ระบบจะส่งข้อความไปยัง Webhook (เชื่อมต่อกับโหนดเจเอส) เพื่อทำการประมวลผลต่อไป

1.2.5 การประมวลผลข้อความด้วยโหนดเจเอส เซิร์ฟเวอร์กลางที่เชื่อมต่อข้อมูล เมื่อคำถามจากผู้ใช้ถูกส่งมายังแอปพลิเคชันไลน์ข้อมูลเหล่านี้จะถูกส่งต่อไปยังเซิร์ฟเวอร์ของระบบแชทบอทที่ทำงานด้วย โหนดเจเอส ซึ่งเป็น

เทคโนโลยีฝั่งเซิร์ฟเวอร์ที่รับคำถามผ่านการตั้งค่า Webhook ที่เชื่อมต่อกับไลน์และจะรับคำถามจากไลน์และส่งต่อคำถามนั้นไปยังระบบไดอะล็อกโพล์เพื่อประมวลผลเพิ่มเติม โดยหน้าที่หลักของ โหนดเจเอส คือเป็นตัวกลางในการเชื่อมต่อและรับส่งข้อมูลระหว่างผู้ใช้และระบบต่างๆ ในแชทบอท

#### 1.2.6 การประมวลผลคำถามด้วยไดอะล็อกโพล์ (Natural Language Processing - NLP)

หลังจากที่ส่งคำถามไปยัง ไดอะล็อกโพล์ระบบนี้จะทำหน้าที่ประมวลผลคำถามโดยใช้ Natural Language Processing (NLP) เพื่อจับคู่เจตนาของคำถามที่ผู้ใช้ส่งมาไดอะล็อกโพล์จะทำการแยกและวิเคราะห์ข้อความเพื่อตรวจสอบว่าเจตนาของคำถามคืออะไร เช่น การถามเกี่ยวกับที่เที่ยว หรือร้านอาหาร

ไดอะล็อกโพล์ ทำงานโดยใช้การสร้างเอนทิตีเพื่อจำแนกประเภทคำถามไว้ และสร้างอินเทนต์ทั้งหมด 30 อินเทนต์ จะทำการกำหนดเอนทิตีแต่ละคำถามเพื่อให้ประมวลผลคำตอบได้แม่นยำมากคำ และอินเทนต์จะถูกฝึกฝนไว้ล่วงหน้าด้วยข้อมูลตัวอย่าง (Training Data) หากคำถามตรงกับอินเทนต์ที่ถูกฝึกไว้ ระบบจะสามารถจับคู่คำถามกับเจตนาที่ตรงกันได้ และจะดำเนินการค้นหาคำตอบจากแหล่งข้อมูลที่เตรียมไว้

ใน ไดอะล็อกโพล์ จะมี Training Phrases ที่เป็นตัวอย่างของคำถามหลาย ๆ รูปแบบที่ผู้ใช้สามารถถามได้ ระบบจะใช้ข้อมูลเหล่านี้ในการฝึกฝนให้รู้จักและเข้าใจความหมายที่ซ่อนอยู่ในคำถามที่มีการใช้คำแตกต่างกัน การจับคู่ อินเทนต์ นี้เรียกว่า Intent Matching ซึ่งเป็นกระบวนการสำคัญในการทำให้แชทบอทสามารถตอบคำถามได้อย่างแม่นยำ

#### 1.2.7 การค้นหาคำตอบจากฐานข้อมูล

Intent Mapping เป็นกระบวนการที่ใช้ในการจับคู่คำถามของผู้ใช้กับอินเทนต์ที่เหมาะสมในระบบ โดยเมื่อผู้ใช้ส่งคำถามผ่านแอปพลิเคชันไลน์ คำถามจะถูกส่งไปยังเซิร์ฟเวอร์ที่ทำงานด้วย Node.js ก่อนที่เซิร์ฟเวอร์จะส่งต่อไปยัง Dialogflow ซึ่งใช้เทคโนโลยี NLP ในการวิเคราะห์ข้อความและจับคู่กับอินเทนต์ที่กำหนดไว้ล่วงหน้า

จะทำการจัดกลุ่มคำถามที่มีเจตนาเดียวกัน เช่น คำถามเกี่ยวกับที่พักหรือร้านอาหาร เมื่อตรวจพบอินเทนต์ที่สอดคล้อง ระบบจะส่งพารามิเตอร์ เช่น ชื่อสถานที่และอินเทนต์ ไปค้นหาคำตอบในฐานข้อมูล โดยจะเริ่มจากฐานข้อมูลที่เพิ่มเองในระบบ หากไม่พบคำตอบ จะค้นหาต่อในฐานข้อมูลที่ดึงข้อมูลจากเว็บไซต์ หลังจากได้คำตอบ ระบบจะจัดหมวดหมู่ข้อมูล เช่น ค่าเช่า เบอร์โทร หรือเวลาเปิดทำการคำตอบที่ฟิลเตอร์และเตรียมไว้จะถูกส่งกลับไปยัง Node.js เพื่อตอบกลับผู้ใช้ในรูปแบบข้อความหรือรูปภาพที่ปรากฏในแอปพลิเคชันไลน์ ให้ผู้ใช้เข้าถึงข้อมูลได้อย่างสะดวกและรวดเร็ว

#### 1.2.8 การค้นหาข้อมูลจากเว็บหากไม่มีในฐานข้อมูล

ในกรณีที่ระบบไม่สามารถหาคำตอบที่เกี่ยวข้องจากฐานข้อมูลได้ ระบบโหนดเจเอส จะทำการดึงข้อมูลมาใช้ในการตอบคำถามจากเว็บไซต์ภายนอกโดยใช้เทคนิค Web Scraping ถูกนำมาใช้ในกรณีที่ไม่มีข้อมูลในฐานข้อมูลภายในที่ตรงกับคำถามของผู้ใช้ กระบวนการเริ่มต้นด้วยการระบุเจตนาของผู้ใช้ (Intent Recognition) เพื่อตรวจสอบว่าผู้ใช้งานต้องการข้อมูลประเภทใด เช่น เวลาเปิดทำการ หรือค่าธรรมเนียมการเช่า โดยระบบจะทำการ

Tokenization (การตัดคำ) TF-IDF (Term Frequency-Inverse Document Frequency) การหาค่า TF-IDF ของคำ ในเอกสาร Named Entity Recognition (NER) การตรวจจับและจำแนกชื่อสิ่งต่าง ๆ

เมื่อได้คำสำคัญแล้ว ระบบจะทำการกรองข้อมูลจากเว็บไซต์ภายนอกโดยใช้เทคนิค Regular Expressions เพื่อตรวจสอบรูปแบบของข้อมูล เช่น ค่าเข้า เวลาเปิดทำการ หรือเบอร์โทรศัพท์ ข้อมูลเหล่านี้จะถูกจัดเก็บในรูปแบบ JSON และรวมเข้ากับข้อมูลจากแหล่งอื่นๆ โดยใช้เทคนิค Fuzzy Matching เพื่อให้ได้คำตอบที่แม่นยำที่สุด

หากข้อมูลมีความยาวเกินไป ระบบจะสรุปคำตอบโดยการตัดข้อความที่ไม่จำเป็นออกและเพิ่ม "..." เพื่อให้ อ่านง่าย ข้อมูลที่ผ่านการประมวลผลทั้งหมดนี้จะถูกส่งกลับไปยังผู้ใช้อย่างสะดวกและครบถ้วนในรูปแบบที่เข้าใจง่าย เช่น ข้อความหรือรูปภาพ ผ่านแพลตฟอร์มที่ผู้ใช้ส่งคำถามเข้ามา

การฟิลเตอร์และส่งคำตอบกลับไปยังผู้ใช้ หลังจากทีระบบดึงข้อมูลที่ตรงกับคำถามจากฐานข้อมูลหรือ เว็บไซต์แล้ว โหนดเจเอส จะทำหน้าที่ ฟิลเตอร์คำตอบ เพื่อให้เนื้อหาที่ตอบกลับมีความเหมาะสมและตรงกับเจตนาของ คำถามมากที่สุด โดยการฟิลเตอร์นี้อาจรวมถึงการตรวจสอบความถูกต้องของข้อมูลหรือการกรองเนื้อหาที่ไม่จำเป็นออก เมื่อได้คำตอบที่เหมาะสมแล้ว ระบบจะส่งข้อมูลกลับไปยังแอปพลิเคชันเพื่อให้ผู้ใช้ได้รับคำตอบแบบเรียลไทม์ กระบวนการทั้งหมดนี้เกิดขึ้นในระยะเวลาเพียงไม่กี่วินาที ทำให้ผู้ใช้สามารถได้รับคำตอบอย่างรวดเร็วและทันที

1.2.9 ระบบค้นหาตำแหน่งสถานที่ตามคำถามของผู้ใช้ได้รับการออกแบบมาเพื่อตรวจสอบและส่งข้อมูล ตำแหน่งกลับไปยังผู้ใช้ผ่านแพลตฟอร์ม เช่น LINE Chatbot โดยใช้ข้อมูลจากฐานข้อมูลภายในและไฟล์ JSON สำหรับ กระบวนการเริ่มต้นด้วยการวิเคราะห์ข้อความ (Text Analysis) เพื่อตรวจจับชื่อสถานที่ที่ผู้ใช้ป้อนเข้ามา พร้อมจัดการ ข้อมูลให้สะอาด เช่น การลบตัวเลขหรือสัญลักษณ์ที่ไม่จำเป็น

จากนั้น ระบบจะเปรียบเทียบข้อความชื่อสถานที่กับข้อมูลในระบบโดยใช้เทคนิค String Matching หาก ชื่อไม่ตรงกันอย่างสมบูรณ์ จะใช้ Fuzzy Matching เพื่อค้นหาชื่อสถานที่ที่ใกล้เคียงที่สุด การรวมข้อมูลจากฐานข้อมูล และไฟล์ JSON สำหรับจะช่วยให้ได้ข้อมูลที่ครอบคลุมและสมบูรณ์มากขึ้น

ระบบจะสร้างข้อความที่มีข้อมูลตำแหน่ง เช่น ชื่อสถานที่, ที่อยู่, ละติจูด และลองจิจูด พร้อมจัดส่งข้อมูล ในรูปแบบที่เหมาะสมกับแพลตฟอร์ม เช่น การสร้างข้อความแผนที่หรือการส่งลิงก์ตำแหน่งกลับไปยังผู้ใช้ผ่าน LINE Chatbot เพื่อให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายและรวดเร็ว

1.2.10 ระบบจัดการพยากรณ์อากาศออกแบบมาเพื่อให้ผู้ใช้สามารถขอข้อมูลพยากรณ์อากาศของเมืองที่ต้องการได้อย่างง่ายดาย โดยทำงานร่วมกับ OpenWeatherMap API เพื่อดึงข้อมูลสภาพอากาศที่แม่นยำ ผู้ใช้สามารถ ระบุชื่อเมืองในคำถามหรือหากไม่ได้ระบุ ระบบจะใช้ค่าเริ่มต้น เช่น "ขอนแก่น" เพื่อทำการค้นหา หลังจากได้รับชื่อเมือง ระบบจะเชื่อมต่อกับ OpenWeatherMap API เพื่อดึงข้อมูลพยากรณ์อากาศของเมืองนั้น ข้อมูลที่ได้รับจะถูกประมวลผล และจัดระเบียบให้เข้าใจง่าย เช่น แสดงอุณหภูมิ ความชื้น และความเร็วลมในรูปแบบที่ชัดเจน จากนั้นจะสร้างข้อความ ตอบกลับที่มีรายละเอียดสภาพอากาศครบถ้วน เช่น "อุณหภูมิ 30°C ความชื้น 60% และความเร็วลม 15 กม./ชม." พร้อมส่งข้อมูลกลับไปยังผู้ใช้ในรูปแบบที่รองรับแพลตฟอร์ม เช่น LINE Chatbot เพื่อให้การเข้าถึงข้อมูลพยากรณ์ อากาศเป็นเรื่องง่ายและสะดวกสำหรับผู้ใช้

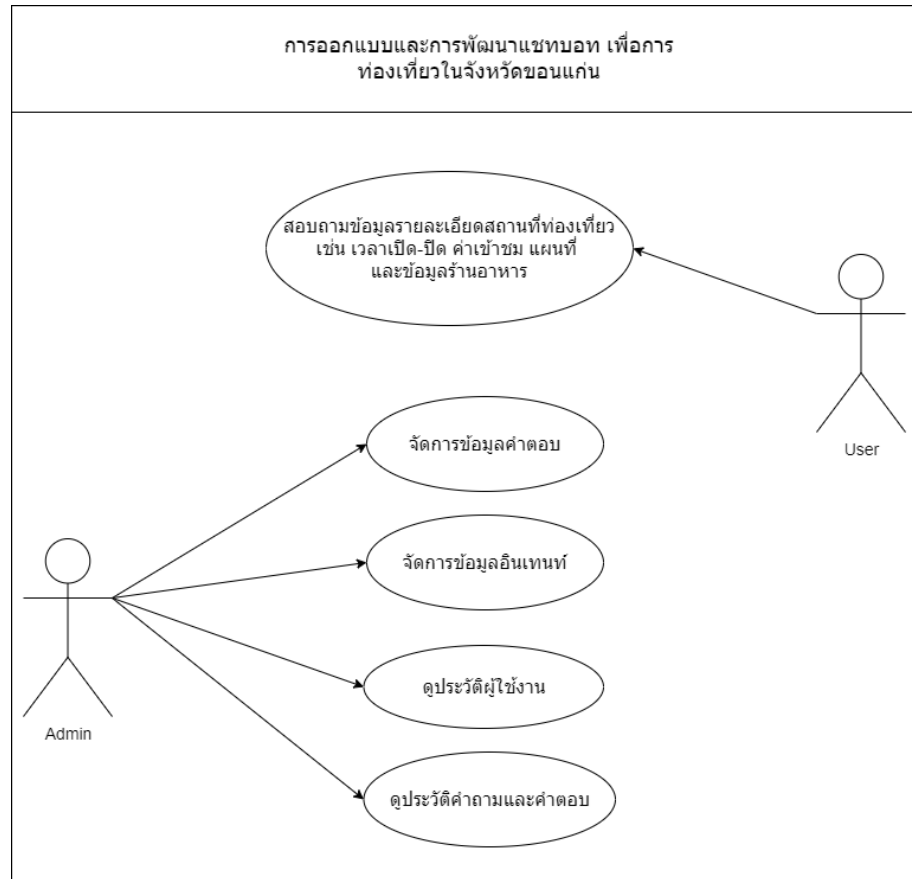
1.2.11 การส่งคำตอบกลับไปยังผู้ใช้ หลังจากที่ได้รับคำตอบจากไดอะล็อกโฟลว์หรือจากฐานข้อมูลแล้ว โหนดเจเอส จะทำหน้าที่ส่งคำตอบกลับไปยังผู้ใช้ผ่าน LINE Messaging API ผู้ใช้จะเห็นคำตอบในรูปแบบของข้อความที่ถูกส่งกลับในแอปพลิเคชันไลน์ข้อมูลที่ผู้ใช้ได้รับจะเป็นข้อมูลที่มีความเกี่ยวข้องกับคำถามที่ถูกลถาม เช่น รายละเอียดของสถานที่ท่องเที่ยวที่น่าสนใจในขอนแก่น

1.2.12 การเก็บข้อมูลการโต้ตอบเพื่อพัฒนาระบบ แอปพลิเคชันยังมีการเก็บข้อมูลการโต้ตอบระหว่างผู้ใช้และระบบ เพื่อใช้ในการปรับปรุงระบบในอนาคต ข้อมูลเช่นคำถามที่ผู้ใช้ถามบ่อย หรือคำถามที่ยังไม่มีคำตอบในระบบ จะถูกบันทึกไว้เพื่อนำไปใช้ในการปรับปรุงการฝึกฝนอินเทนท์ และเพิ่มข้อมูลในฐานข้อมูลให้มีความครอบคลุมมากขึ้น นอกจากนี้ ผู้ดูแลระบบยังสามารถเข้ามาปรับปรุงและเพิ่มอินเทนท์ใหม่ ๆ ให้ครอบคลุมกับคำถามที่ผู้ใช้ถามได้มากขึ้น

1.2.13 การดูแลและอัปเดตข้อมูลโดยผู้ดูแลระบบ ผู้ดูแลระบบสามารถเข้าถึงข้อมูลที่บันทึกไว้ในระบบและทำการปรับปรุงข้อมูลหรือคำตอบใหม่ ๆ รวมถึงการเพิ่ม อินเทนท์ ใหม่ที่เกี่ยวข้องกับคำถามเฉพาะหรือสถานที่ท่องเที่ยวใหม่ ๆ ในขอนแก่น สิ่งนี้จะช่วยให้แอปพลิเคชันสามารถให้ข้อมูลที่เป็นปัจจุบันและตรงตามความต้องการของผู้ใช้ได้มากขึ้น

กระบวนการทำงานของแอปพลิเคชันสถานที่ท่องเที่ยวขอนแก่นเป็นการทำงานร่วมกันระหว่างผู้ใช้ ระบบไลน์ โหนดเจเอส ไดอะล็อกโฟลว์ ฐานข้อมูล และกระบวนการ เทคนิคการดึงข้อมูลจากเว็บไซต์ (Web Scraping) เพื่อให้ผู้ใช้ได้รับข้อมูลที่ถูกต้องและเป็นปัจจุบัน ระบบยังสามารถปรับปรุงและขยายฟังก์ชันการทำงานได้อย่างต่อเนื่องเพื่อให้บริการได้อย่างมีประสิทธิภาพมากขึ้นในอนาคต

1.3 การออกแบบและการพัฒนาแอปพลิเคชัน เพื่อการท่องเที่ยวในจังหวัดขอนแก่น ได้นำเสนอการวิเคราะห์ระบบด้วยยูสเคสไดอะแกรม (User case diagram) โดยแอดมิน (Admin) ทำหน้าที่เพิ่ม ลบ แก้ไข ข้อมูลคำตอบและอินเทนท์ สามารถดูประวัติผู้มาใช้งาน ดูประวัติคำถามและคำตอบ ส่วนผู้ใช้สามารถสอบถามข้อมูลรายละเอียดสถานที่ท่องเที่ยว เวลาเปิด-ปิด เส้นทางเดินทาง ข้อมูลค่าเข้าชม และข้อมูลร้านอาหาร ดังรูปภาพที่ 2 ดังนี้



ภาพที่ 1 ยูสเคสไดอะแกรมของการออกแบบและพัฒนาแอปพลิเคชัน การท่องเที่ยวจังหวัดขอนแก่น

ตารางที่ 1 ตารางแสดงข้อมูล Use Case Description สอบถามข้อมูลรายละเอียดสถานที่ท่องเที่ยว

Use case name:	สอบถามข้อมูลรายละเอียดสถานที่ท่องเที่ยว เช่น เวลาเปิด-ปิด ค่าเข้าชม แผนที่ และ ข้อมูลร้านอาหาร	
Scenario	ผู้ใช้บริการต้องการสอบถามข้อมูลสถานที่ท่องเที่ยวในจังหวัดขอนแก่นผ่านแชทบอท ระบบจะให้ข้อมูลรายละเอียดของสถานที่ท่องเที่ยว รายละเอียด เวลาเปิด-ปิด ค่าเข้าชม แผนที่ และ ข้อมูลร้านอาหาร	
Brief description:	ผู้ใช้ต้องการสอบถามข้อมูลรายละเอียดสถานที่ท่องเที่ยว	
Actors:	ผู้ใช้	
Preconditions:	แชทบอทจะต้องทำงานได้และข้อมูลรายละเอียดสถานที่ท่องเที่ยว	
Postconditions:	ผู้ใช้จะได้รับข้อมูลรายละเอียดเกี่ยวกับแหล่งท่องเที่ยวที่ร้องขอ	
Flow of activities:	User	System
	<ol style="list-style-type: none"> <li>1. เพิ่มเพื่อนในแอปพลิเคชันไลน์</li> <li>2. เริ่มการสนทนากับแชทบอทและถามเกี่ยวกับสถานที่ท่องเที่ยวที่เฉพาะเจาะจง</li> <li>3. รับคำตอบจากแชทบอท</li> <li>4. ขอรายละเอียดเพิ่มเติมหรือสถานที่ท่องเที่ยวอื่นๆ</li> </ol>	<ol style="list-style-type: none"> <li>1. แสดงหน้าไลน์แชทบอทสถานที่ท่องเที่ยว</li> <li>2. รับคำถามจากผู้ใช้งาน</li> <li>3. ประมวลผลคำถามเพื่อระบุแหล่งท่องเที่ยวที่ต้องการสอบถาม</li> <li>4. ดึงข้อมูลรายละเอียดที่เกี่ยวข้องจากฐานข้อมูลหรือจากเว็บไซต์</li> <li>5. สร้างการตอบสนองด้วยข้อมูลที่ร้องขอ</li> <li>6. ส่งข้อมูลรายละเอียดให้ผู้ใช้งาน</li> </ol>
Exception conditions	<ol style="list-style-type: none"> <li>1. ผู้ใช้ถามเกี่ยวกับสถานที่ท่องเที่ยวที่ไม่อยู่ในขอบเขตของโครงการ</li> <li>2. คำถามของผู้ใช้ไม่สมบูรณ์หรือไม่ชัดเจน</li> <li>3. ผู้ใช้ทำการสอบถามจำนวนมากเกินไปในช่วงเวลาสั้นๆ</li> </ol>	

ตารางที่ 2 ตารางแสดงข้อมูล Use Case Description การจัดการข้อมูลคำตอบ

Use case name:	การจัดการข้อมูลคำตอบ	
Scenario	ผู้ดูแลระบบต้องการเพิ่ม ลบ แก้ไขคำตอบในระบบ	
Brief description:	ผู้ดูแลระบบจัดการข้อมูลคำตอบ	
Actors:	ผู้ดูแลระบบ	
Preconditions:	ผู้ดูแลระบบจะต้องได้รับอนุญาตในการจัดการข้อมูลคำตอบ	
Postconditions:	การจัดการข้อมูลคำตอบจะถูกบันทึกลงในระบบ	
Flow of activities:	User	System
	1. เข้าหน้าสำหรับแอดมิน	1. แสดงหน้าจอสำหรับแอดมิน
	2. การจัดการข้อมูลคำตอบ	2. แสดงหน้าจอการจัดการข้อมูลคำตอบ
	3. เลือกการดำเนินการเพิ่ม ลบ หรือ แก้ไขคำตอบ	3. แสดงหน้าจอการเพิ่ม ลบ หรือแก้ไขคำตอบ
	4. เพิ่ม ลบ หรือแก้ไขคำตอบ	4. ตรวจสอบข้อมูลที่ทำกรจัดการข้อมูล
	5. ยืนยันการดำเนินการ	5. บันทึกข้อมูลลงในระบบ
Exception conditions	-	

ตารางที่ 3 ตารางแสดงข้อมูล Use Case Description การจัดการอินเทอร์เน็ต

Use case name:	การจัดการอินเทอร์เน็ต	
Scenario	ผู้ดูแลระบบต้องการเพิ่ม ลบ หรือแก้ไขอินเทอร์เน็ตในระบบ	
Brief description:	ผู้ดูแลระบบจัดการอินเทอร์เน็ตในระบบ	
Actors:	ผู้ดูแลระบบ	
Preconditions:	ผู้ดูแลระบบจะต้องได้รับอนุญาตในการจัดการอินเทอร์เน็ต	
Postconditions:	การจัดการอินเทอร์เน็ตจะถูกบันทึกลงในระบบ	
Flow of activities:	User	System
	1. เข้าหน้าสำหรับแอดมิน	1. แสดงหน้าจอสำหรับแอดมิน
	2. การจัดการข้อมูลอินเทอร์เน็ต	2. แสดงหน้าจอการจัดการข้อมูลอินเทอร์เน็ต
	3. เลือกการดำเนินการเพิ่ม ลบ หรือแก้ไขอินเทอร์เน็ต	3. แสดงหน้าจอการเพิ่ม ลบ หรือแก้ไขอินเทอร์เน็ต
	5. เพิ่ม ลบ หรือแก้ไขอินเทอร์เน็ต	4. ตรวจสอบข้อมูลที่ทำกรจัดการข้อมูล
	5. ยืนยันการดำเนินการ	5. บันทึกข้อมูลลงในระบบ
Exception conditions	-	



ตารางที่ 4 ตารางแสดงข้อมูล Use Case Description ดูประวัติผู้ใช้งาน

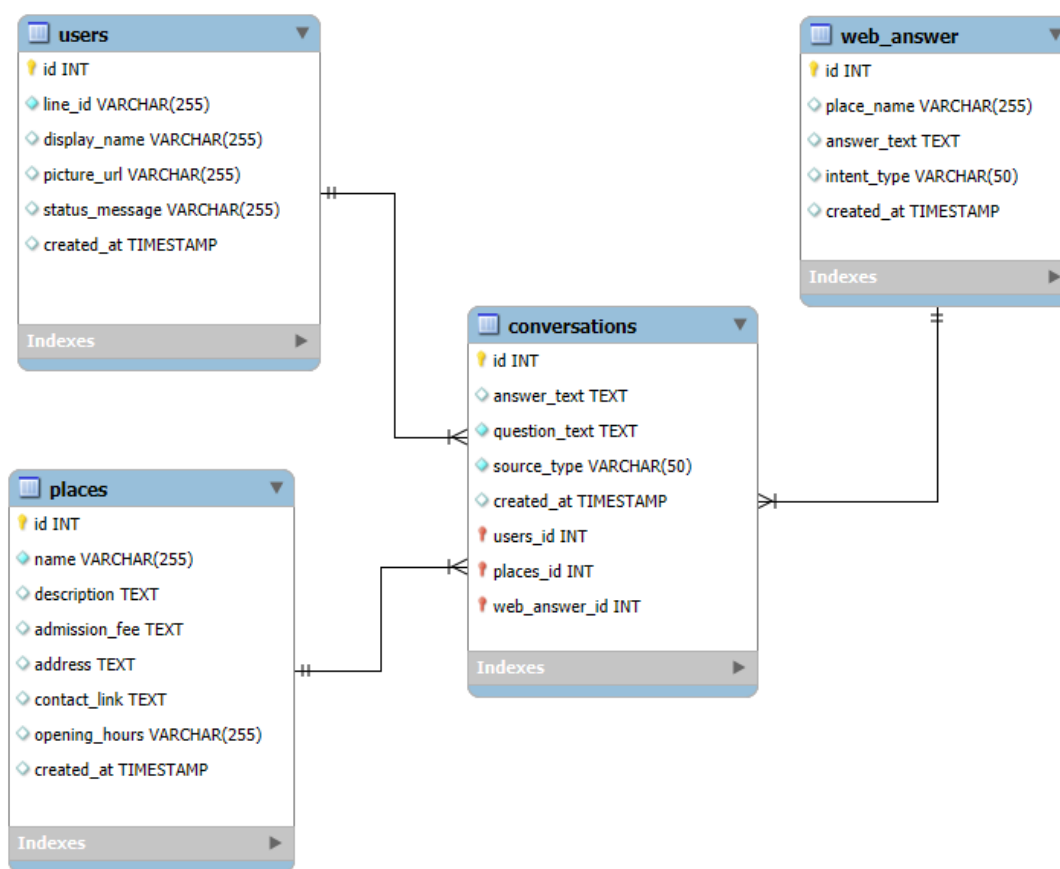
Use case name:	ดูประวัติผู้ใช้งาน	
Scenario	ผู้ดูแลระบบต้องการดูข้อมูลผู้ที่มาสอบถามข้อมูลจากแชทบอท เพื่อนำไปวิเคราะห์ต่อไป	
Brief description:	ผู้ดูแลระบบต้องการดูข้อมูลของผู้ที่มาสอบถามข้อมูลจากแชทบอท	
Actors:	ผู้ดูแลระบบ	
Preconditions:	ผู้ดูแลระบบจะต้องได้รับอนุญาตในการเข้าดูฐานข้อมูล	
Postconditions:	-	
Flow of activities:	User	System
	1. เข้าหน้าสำหรับแอดมิน  2. ข้อมูลผู้ใช้งาน	1. แสดงหน้าจอสำหรับแอดมิน  2. แสดงหน้าจอข้อมูลของผู้ใช้งาน
Exception conditions	มีข้อมูลของผู้ใช้งานในระบบ	

ตารางที่ 5 ตารางแสดงข้อมูล Use Case Description ดูประวัติคำถามและคำตอบ

Use case name:	ดูประวัติคำถามและคำตอบ	
Scenario	ผู้ดูแลระบบต้องการดูประวัติคำถามและคำตอบ ข้อมูลนี้จะใช้เพื่อวิเคราะห์ ปรับปรุงการตอบสนองของแชทบอท	
Brief description:	ผู้ดูแลระบบต้องการดูข้อมูลประวัติคำถามและคำตอบจากแชทบอท	
Actors:	ผู้ดูแลระบบ	
Preconditions:	ผู้ดูแลระบบจะต้องได้รับอนุญาตในการเข้าดูฐานข้อมูล	
Postconditions:	-	
Flow of activities:	User	System
	1. เข้าหน้าสำหรับแอดมิน  2. ข้อมูลประวัติคำถามและคำตอบ	1. แสดงหน้าจอสำหรับแอดมิน  2. แสดงหน้าจอข้อมูลประวัติคำถามและคำตอบ
Exception conditions	มีข้อมูลประวัติคำถามและคำตอบในระบบ	

#### 1.4 แผนภาพอี-อาร์ไดอะแกรม (ER Diagram)

การออกแบบฐานข้อมูล โดยฐานข้อมูลจะประกอบไปด้วย 4 ตาราง ได้แก่ ตารางการสนทนา ใช้เก็บข้อมูลการสนทนาระหว่างผู้ใช้กับแชทบอท ตารางผู้ใช้ ใช้เก็บข้อมูลในส่วนของผู้ใช้งาน ตารางสถานที่ ใช้เก็บข้อมูลต่างๆของแต่ละสถานที่ที่จะนำมาตอบคำถาม ในกรณีค้นหาคำตอบจากฐานข้อมูล และตารางคำตอบจากเว็บไซต์ ใช้เก็บคำตอบ ในกรณีที่ดึงมาจากเว็บไซต์ลงในฐานข้อมูล ดังภาพที่ 3



ภาพที่ 2 แผนภาพอี-อาร์ไดอะแกรม (ER Diagram)

1.5 พจนานุกรมข้อมูล (Data Dictionary)

ตารางที่ 6 users

Data name	Description	Key	Constraints	Type	Length	Data Value
user_id	ไอดี	PK	Not null	INT		1
line_id	ไอดีไลน์ผู้ใช้			VARCHAR	255	mileee_
display_name	ชื่อผู้ใช้			VARCHAR	255	Mile
picture_url				VARCHAR	255	https://sprofile.line-scdn.net/0hyliGLRfJ
status_message				VARCHAR	255	ข้อความสถานะ
created_at	เวลาที่เพิ่ม ข้อมูลลง ฐานข้อมูล			TIMESTAMP		2024-07-25 16:32:05

ตารางที่ 7 conversations

Data name	Description	Key	Constraints	Type	Length	Data Value
conversations_id	ไอดี	PK	Not null	INT		1
question_text	คำถาม		Not null	TEXT		ต้นตาลเปิดกี่โมง
answer_text	คำตอบ			TEXT		เปิดทุกวัน 16.00-23.00 น.
source_type	ประเภทของการ ตั้งคำตอบ		VARCHAR	50		web_database
created_at	เวลาที่เพิ่มข้อมูล ลงฐานข้อมูล			TIMESTAMP		2024-07-25 16:32:05
user_id	ไอดีของตาราง users	FK	Not null	INT		1
place_id	ไอดีของตาราง place	FK	Not null	INT		1
web_answer_id	ไอดีของตาราง web_answer	FK	Not null	INT		1

ตารางที่ 8 web\_answers

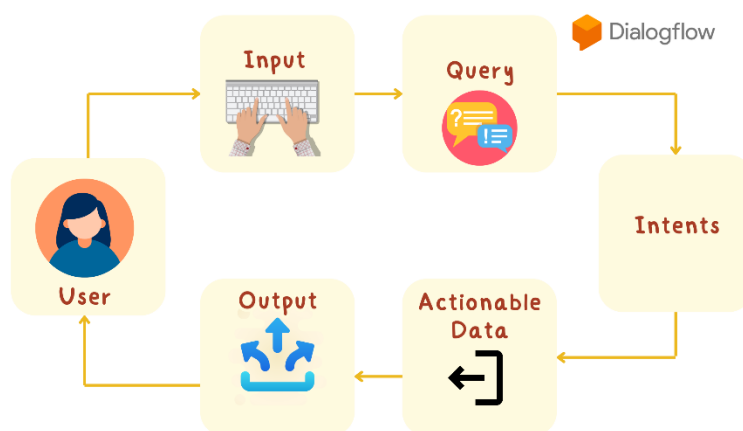
Data name	Description	Key	Constraints	Type	Length	Data Value
id	ไอดี	PK	Not null	INT		1
place_name	ชื่อสถานที่			VARCHAR	255	ผาชมตะวัน
answer_text	คำตอบ			TEXT		เปิดทุกวัน 16.00-23.00 น.
Intent_type	ชื่ออินเทอร์เน็ต			VARCHAR	50	ค่าธรรมเนียมการเข้า
created_at	เวลาที่เพิ่มข้อมูลลงฐานข้อมูล			TIMESTAMP		2024-07-25 16:32:05

ตารางที่ 9 place

Data name	Description	Key	Constraints	Type	Length	Data Value
id	ไอดี	PK	Not null	INT		1
name	ชื่อสถานที่			VARCHAR	255	ผาชมตะวัน
description	รายละเอียด			TEXT		ผาชมตะวัน เป็นความสวยงามอลังการของทะเลหมอกยามเช้า โดยเฉพาะในช่วงปลายฤดูฝน
admission_ fee	ค่าเข้าชม			TEXT		ผู้ใหญ่ 40 บาท เด็ก 20 บาท , ชาวต่างชาติ ผู้ใหญ่ 200 บาท เด็ก 100 บาท
address	ที่อยู่			TEXT		อุทยานแห่งชาติภูเวียง ตำบลในเมือง อำเภอ เวียงเก่า จังหวัด ขอนแก่น
contact_ link	ลิงค์ติดต่อ			TEXT		<a href="https://www.facebook.com/Phuwiangnp71/">https://www.facebook.com/Phuwiangnp71/</a>
opening_ hours	เวลาเปิดทำการ			VARCHAR	255	เปิดให้เข้าชม ทุกวัน : 08.30-16.30 น.
created_at	เวลาที่เพิ่ม ข้อมูลลง ฐานข้อมูล			TIMESTAMP		2024-07-25 16:32:05

## 2. การออกแบบแชทบอท

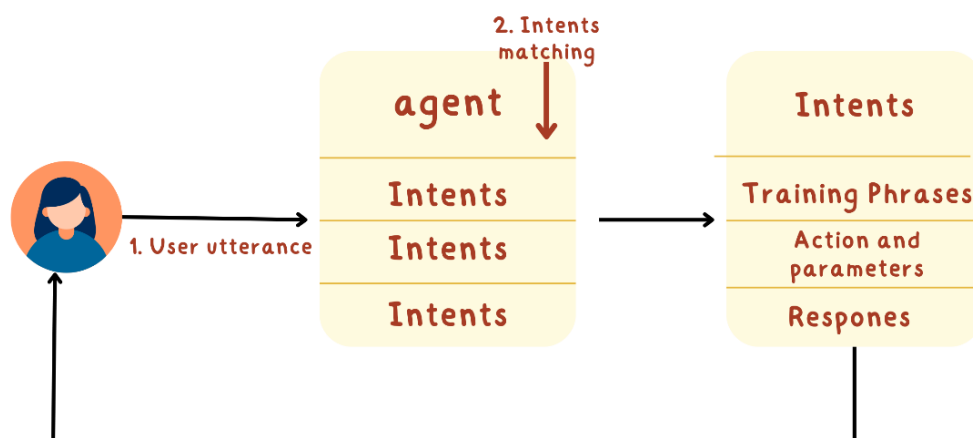
2.1 การออกแบบหลักการทำงานโดยรวมของแชทบอทการท่องเที่ยวจังหวัดขอนแก่น การพัฒนาระบบแชทบอท บนแพลตฟอร์มไลน์ สามารถอธิบาย หลักการทำงานตามภาพที่ 4 เมื่อผู้ใช้พิมพ์ข้อความเข้ามา ข้อความนั้นจะถูกส่งไปที่ ไดอะล็อกโฟลว์ ซึ่งจะใช้การประมวลผลภาษาธรรมชาติ เพื่อแปลงข้อความ จากนั้นจะค้นหาอินเทนท์ที่ตรงกับข้อความที่ผู้ใช้พิมพ์เข้ามา เมื่อเจออินเทนท์ที่ตรงแล้วไดอะล็อกโฟลว์ จะตอบกลับข้อความไปยังผู้ใช้งานผ่านไลน์



ภาพที่ 3 หลักการทำงานโดยรวมของแชทบอทการท่องเที่ยวจังหวัดขอนแก่น

2.2 ออกแบบขั้นตอนการส่งข้อมูลเพื่อใช้ในการโต้ตอบการสนทนา เมื่อผู้ใช้งานได้ สนทนากับไลน์บอทระบบจะตรวจสอบแพลตฟอร์มที่ผู้ใช้งานใช้งานอยู่และเชื่อมต่อกับเอเจนต์ในไดอะล็อกโฟลว์ ไดอะล็อกโฟลว์ จะเปรียบเทียบข้อความที่ผู้ใช้งานส่งเข้ามา กับ Training Phrases ในแต่ละอินเทนท์จะให้น้ำหนักคำที่มีความใกล้เคียงหรือคล้ายกับ Training Phrases มากที่สุด ไดอะล็อกโฟลว์ จะเลือกอินเทนท์ที่มีข้อมูลคล้ายกับข้อความผู้ใช้งานมากที่สุดไดอะล็อกโฟลว์ จะดำเนินการเลือกใช้ข้อมูลที่ได้กำหนดไว้ในส่วนของ การตอบกลับ (Response) ของอินเทนท์ที่เลือกมาตอบกลับผู้ใช้งาน ดังภาพที่ 5





ภาพที่ 4 ขั้นตอนการส่งข้อมูลเพื่อใช้ในการโต้ตอบการสนทนา

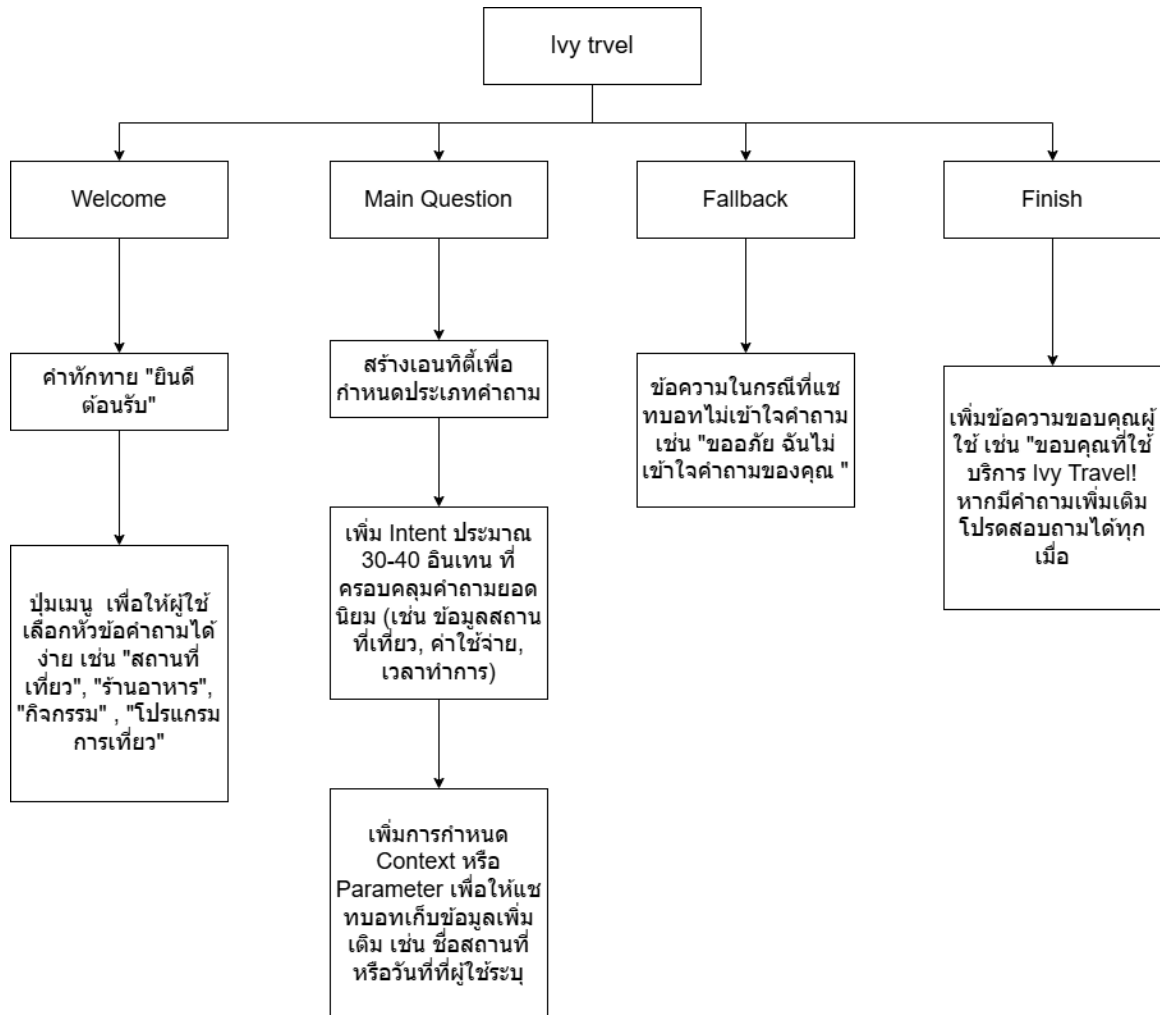
## 2.3 ออกแบบและวางแผนโครงสร้างของแชทบอท (Chatbot Conversational Flow) ประกอบด้วย

### 2.3.1 Welcome เป็นการทักทายก่อนสนทนา

2.3.2 Main Question เป็นอินเทนท์คำตอบหลักที่จะบอกมีข้อมูลอะไรบ้าง บอทสามารถทำอะไรได้บ้าง เช่น ข้อมูลสถานที่ท่องเที่ยวขอนแก่น ร้านอาหารมิชลินไกด์ ค่าเข้าชม เวลาเปิด-ปิด แผนที่ เป็นต้น

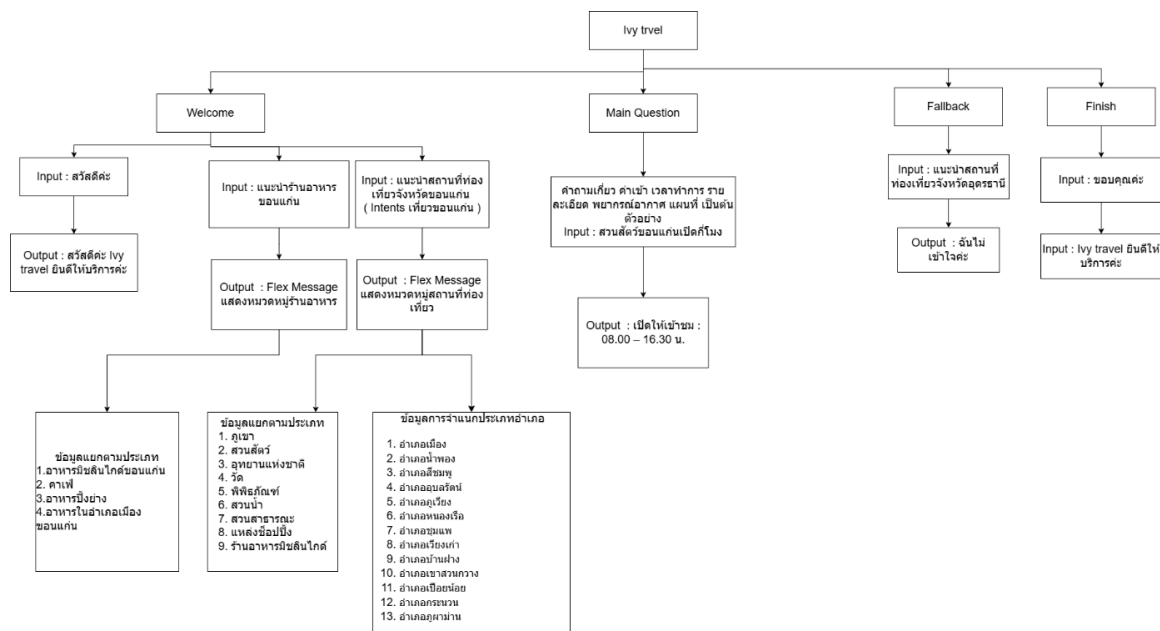
2.3.3 Fallback เป็นอินเทนท์สำหรับ ส่งข้อความให้ผู้ใช้ในกรณีที่ผู้ใช้ส่งคำตอบไม่ถูกต้องหรือไม่เกี่ยวข้องกับสถานที่ท่องเที่ยวและร้านอาหารมิชลินไกด์ขอนแก่น

2.3.4 Finish เป็นอินเทนท์สำหรับ ผู้ใช้จบบทสนทนากับบอทเรียบร้อยแล้ว บอทจะแสดงข้อความขอบคุณ ดังแสดงในภาพที่ 6



ภาพที่ 5 โครงสร้างโดยรวมของแชทบอท (Chatbot Conversational Flow)

- 2.4 การกำหนดกลุ่มข้อมูลคำถามและกลุ่มข้อมูลคำตอบเพื่อใช้ในการโต้ตอบการสนทนา โดยมีองค์ประกอบ
- Welcome เป็นการทักทายก่อนสนทนา เช่น ผู้ใช้ส่งข้อความ สวัสดีค่ะ บอทจะตอบกลับสวัสดีค่ะ Ivy travel ยินดีให้บริการค่ะ
  - Main Question เป็นอินเทนท์คำตอบหลักที่จะบอกมีข้อมูลอะไรบ้าง เช่น ผู้ใช้ส่งข้อความ ข้อมูลสถานที่ท่องเที่ยวขอนแก่น บอทจะตอบ Flex Message แสดงหมวดหมู่ร้านอาหาร
  - Fallback เป็นอินเทนท์สำหรับ ส่งข้อความให้ผู้ใช้ในกรณีที่ผู้ใช้ส่งคำตอบไม่ถูกต้องหรือไม่เกี่ยวข้องกับสถานที่ท่องเที่ยวและร้านอาหารมีขลิ้นโค้ดขอนแก่น เช่น ผู้ใช้ส่งข้อความแนะนำสถานที่ท่องเที่ยวจังหวัดอุดรธานี บอทจะตอบกลับ ฉันไม่เข้าใจค่ะ
  - Finish เป็นอินเทนท์สำหรับ ผู้ใช้จบบทสนทนากับบอทเรียบร้อยแล้ว บอทจะแสดงข้อความขอบคุณ ดังตัวอย่างภาพที่ 7



ภาพที่ 6 โครงสร้างการสนทนาของระบบแชทบอท (Chatbot Conversational Flow)

### 3. การพัฒนาแชทบอท

3.1 การพัฒนาแอปพลิเคชันแชทบอทในการแนะนำสถานที่ท่องเที่ยวในจังหวัดขอนแก่น ผู้จัดทำใช้เครื่องมือในการพัฒนาระบบประกอบไปด้วย

3.1.1 ไลน์ ดีเวลลอปเปอร์ (LINE Developers) เป็นเครื่องมือสร้างบัญชีเริ่มต้นในการพัฒนาระบบแชทบอทขึ้น

3.1.2 ไลน์ออฟฟิเชียลแอคเคาท์ (LINE Official Account) เป็นเครื่องมือหลักในการออกแบบฟังก์ชันต่างๆ ของระบบแชทบอท

3.1.3 ไลน์บอทดีไซน์ เป็นเครื่องมือออกแบบรูปแบบต่างๆ และแปลงให้อยู่ในรูปแบบเจสัน

3.1.4 ไดอะล็อกโฟลว์ เป็นเครื่องมือสร้างแชทบอท เจตนา บอทตอบกลับ

3.1.5 ฐานข้อมูล เป็นเครื่องมือที่ใช้ในการรวบรวมคำถามต่างๆ โดยผู้จัดทำได้ใช้เครื่องมือข้างต้นมาพัฒนาแชทบอทให้ใช้งานง่ายขึ้น และแสดงการพัฒนาแชทบอท ในแอปพลิเคชันไลน์

3.1.6 เว็สแคปปิง (Web Scraping) เป็นการดึงข้อมูลจากเว็บไซต์

3.1.7 โหนดเจเอส (Node.js) ใช้ในการพัฒนาเซิร์ฟเวอร์ สำหรับการเชื่อมต่อกับไลน์เมสเสจเอพีไอและไดอะล็อกโฟลว์

3.1.8 เอ็กซ์เพรสเจเอส (Express.js) เป็นเว็บเฟรมเวิร์กสำหรับ โหนดเจเอส ใช้ในการสร้าง API และจัดการเส้นทางต่างๆ

#### 3.2 วิธีการพัฒนาแชทบอท

3.2.1 สร้างบัญชีเริ่มต้นที่จะใช้ในการพัฒนาระบบแชทบอทโดยสมัครจากเว็บไซต์ ไลน์ดีเวลลอปเปอร์ และทำการสร้างการบริหาร (Administration) ชื่อว่า Ivy travel จากนั้นสร้าง ชาแนล (Channels) ชื่อ Khonkaen Travel

3.2.2 ไลน์ออฟฟิเชียลแอคเคาท์ เป็นเครื่องมือหลักที่ผู้จัดทำใช้ในการออกแบบฟังก์ชันต่างๆ ของระบบแชทบอท ซึ่งสร้างหลังจากสร้างบัญชีของไลน์ดีเวลลอปเปอร์ซึ่งจะเชื่อมต่อกันโดยอัตโนมัติใช้ในการสร้างข้อความตอบกลับและสร้างริชเมนู

3.2.3 ไลน์บอทดีไซน์ เป็นเครื่องมือออกแบบรูปแบบต่างๆ และแปลงให้อยู่ในรูปแบบเจสัน แล้วนำไปใช้ในการในการคำตอบแบบ Flex Message ในไดอะล็อกโฟลว์

3.2.4 เชื่อมต่อไลน์บอทกับไดอะล็อกโฟลว์ ให้สามารถทำงานร่วมกันและตอบคำถามในไลน์ได้

(1) เชื่อมต่อไดอะล็อกโฟลว์กับไลน์เข้าด้วยกันโดยเลือก Integrations แล้วเลือกที่แอปพลิเคชันไลน์

(2) นำ Channel ID Channel Secret และ Channel Access Token ในไลน์ดีเวลลอปเปอร์ของเรา มาใส่ในช่องที่ให้กรอก และนำลิงก์ของไดอะล็อกโฟลว์ ไปวางในไลน์ดีเวลลอปเปอร์

3.2.5 การใช้ข้อมูลที่เก็บรวบรวมมา การออกแบบและปรับปรุงแชทบอท

(1) การสร้าง อินเทนท์ และ Entities: ใช้ข้อมูลที่เก็บรวบรวมเพื่อลงรายละเอียดใน อินเทนท์ และ Entities ของ ไดอะล็อกโฟลว์

(2) เพิ่ม Training Phrases: ข้อมูลที่เก็บมา เช่น คำถามและคำตอบ สามารถนำมาสร้าง Training Phrases ใน อินเทนท์ ของ ไดอะล็อกโพล์ เพื่อให้โมเดลเรียนรู้และจับคู่คำถามจากผู้ใช้ได้ดีขึ้น

(3) การสร้างฐานข้อมูล: ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยว เวลาทำการ ค่าเข้าชม และประเภทของสถานที่ สามารถใช้ในการสร้างฐานข้อมูลที่แชทบอทจะใช้ในการตอบคำถามของผู้ใช้

(4) การอัปเดตข้อมูล: ข้อมูลจากแหล่งต่าง ๆ ช่วยในการอัปเดตและจัดการข้อมูลที่แชทบอทใช้ เช่น การเปลี่ยนแปลงเวลาเปิด-ปิด หรือการเพิ่มสถานที่ใหม่

3.2.6 สร้างโปรเจกต์ในไดอะล็อกโพล์ และสร้างอินเทนท์ต่าง ๆ ตามคำถามที่ได้อบรมมา เช่น สร้างอินเทนท์ที่ตามที่ต้องการทั้งหมด ผู้จัดทำ ได้สร้างโครงสร้างประโยคที่เป็นคำถามของบอทไว้ทั้งหมด โดยแยกอินเทนท์ออกเป็นทั้งหมด 30 อินเทนท์ มีองค์ประกอบดังนี้

(1) แนะนำสถานที่ท่องเที่ยวขอนแก่น

(2) แนะนำร้านอาหาร ได้แก่ ร้านอาหารมิชลินไกด์ขอนแก่น คาเฟ่ ร้านปิ้งย่าง ร้านอาหารในเมืองขอนแก่น

(3) จำแนกประเภทตามชื่อสถานที่ท่องเที่ยวแต่ละประเภท เช่น ภูเขา สวนสัตว์ พิพิธภัณฑ์ อุทยานแห่งชาติ สวนน้ำ วัด สวนสาธารณะ แหล่งช้อปปิ้ง

(4) จำแนกประเภทตามอำเภอข้อมูลการจำแนกประเภทอำเภอ ได้แก่ อำเภอเมือง อำเภอน้ำพอง อำเภออุบลรัตน์ อำเภอสหัสขันธ์ อำเภอภูเวียง อำเภอหนองเรือ อำเภอชุมแพ อำเภอเวียงเก่า อำเภอบ้านฝาง อำเภอเขาสมบวง อำเภอเปือยน้อย อำเภอกระนวน อำเภอภูผาม่าน

(5) จำแนกประเภทตาม เวลาเปิด-ปิด รายละเอียดสถานที่ ค่าเข้าชม รูปภาพที่แสดง แผนที่ ของแต่ละสถานที่ท่องเที่ยวและพยากรณ์อากาศ ในแต่ละอินเทนท์ให้เพิ่มตัวอย่างการถาม (Training Phrases) ที่เกี่ยวข้องกับหัวข้อนั้น ๆ และเพิ่มข้อมูลตอบกลับเกี่ยวกับข้อมูลสถานที่ท่องเที่ยว เวลาเข้าชม ค่าเข้าชม รูปภาพ แผนที่ ร้านอาหาร ช่องทางการติดต่อ เป็นต้น

3.2.7 การเทรนข้อมูลไดอะล็อกโพล์ จะทำการเทรนข้อมูลโดยการ เพิ่ม Training Phrases ใน ไดอะล็อกโพล์เป็นกระบวนการที่สำคัญในการฝึกฝนแชทบอทให้สามารถเข้าใจคำถามหรือข้อความที่ผู้ใช้ส่งมาได้อย่างถูกต้อง โดย Training Phrases คือประโยคตัวอย่างที่ใช้ในการฝึกให้ระบบจับคู่ข้อความจากผู้ใช้อินเทนท์ที่ถูกต้องกระบวนการทำ Training Phrases สามารถอธิบายได้ตามขั้นตอนต่อไปนี้

(1) เริ่มจากการสร้าง อินเทนท์ ใหม่ใน ไดอะล็อกโพล์โดย อินเทนท์ คือการระบุจุดมุ่งหมายหรือหัวข้อหลักที่ผู้ใช้ถามถึง เช่น อินเทนท์ สำหรับถามเกี่ยวกับ "สถานที่ท่องเที่ยวในขอนแก่น" หรือ "เวลาเปิดปิดของสวนสัตว์ขอนแก่น"

(2) กำหนด Training Phrases ในแต่ละ อินเทนท์ จำเป็นต้องมีตัวอย่างประโยค (Training Phrases) เพื่อให้ระบบเรียนรู้รูปแบบต่าง ๆ ของการถามคำถามในหัวข้อนั้น ๆ Training Phrases คือชุดของประโยคหรือข้อความที่เป็นตัวอย่างคำถามที่ผู้ใช้จะถาม หากมีข้อมูลคำถามเยอะมาก จะใช้ Bulk Import ช่วยให้คุณสามารถเพิ่มข้อมูล

จำนวนมากได้อย่างรวดเร็ว โดยใช้ไฟล์ JSON หรือ CSV ไดอะล็อกโพล์ จะฝึกฝนโมเดลอัตโนมัติทุกครั้งที่มีการเปลี่ยนแปลงใน อินเทนท์ หรือ Training Phrases

(3) การสร้าง Training Phrases คำถามที่ผู้ใช้สามารถถามในหัวข้อเดียวกันอาจมีรูปแบบหรือการใช้คำที่หลากหลาย ตัวอย่างเช่น สถานที่ท่องเที่ยวในขอนแก่นมีอะไรบ้าง? แนะนำที่เที่ยวในขอนแก่นหน่อย มีที่เที่ยวในขอนแก่นที่แนะนำสนใจบ้าง?

(4) นำประโยคตัวอย่างที่คิดไว้มาป้อนเข้าไปในส่วน Training Phrases ของ อินเทนท์ นั้น ๆ ในไดอะล็อกโพล์ การใส่หลายประโยคที่ใช้คำต่างกันจะช่วยให้ระบบมีความเข้าใจที่หลากหลายมากขึ้น

(5) ระบุ Entity (หากจำเป็น): หากในประโยคมีข้อมูลที่เฉพาะเจาะจง เช่น ชื่อสถานที่ ชื่อจังหวัด หรือ เวลาเปิด-ปิด สามารถระบุ Entity เพื่อให้ระบบเรียนรู้ว่า ข้อมูลนั้นเป็นข้อมูลเฉพาะที่ต้องดึงออกมาตอบคำถามในอนาคต ตัวอย่างเช่น หากมีคำถาม "ขอนแก่นมีอะไรน่าเที่ยวบ้าง" ระบุว่า "ขอนแก่น" เป็น Entity ของจังหวัด

(6) การฝึกฝนระบบ (Training) หลังจากใส่ Training Phrases เรียบร้อยแล้ว ระบบ ไดอะล็อกโพล์ จะทำการเรียนรู้จากประโยคตัวอย่างที่ใส่เข้าไป เมื่อผู้ใช้งานคำถามในรูปแบบต่าง ๆ ที่คล้ายคลึงกับ Training Phrases ระบบจะสามารถจับคู่กับ อินเทนท์ ที่ถูกต้องได้

การทำ Training Phrases ใน ไดอะล็อกโพล์ คือการใส่ประโยคตัวอย่างที่ผู้ใช้สามารถถามในหัวข้อหรือ อินเทนท์ ต่าง ๆ การระบุและใส่ประโยคตัวอย่างที่หลากหลายจะช่วยให้เซพบอทสามารถจับคู่คำถามกับ อินเทนท์ ที่ถูกต้องได้แม่นยำมากขึ้น การฝึกฝนนี้เป็นกระบวนการที่ต้องทำอย่างต่อเนื่องและควรมีการเพิ่มประโยคใหม่ ๆ เมื่อพบว่าผู้ใช้งานถามคำถามในรูปแบบที่ระบบยังไม่รู้จัก

3.2.8 การตอบกลับ (Responses) เมื่อสร้างโครงสร้างประโยคคำถามเสร็จผู้จัดทำใช้กำหนดคำตอบโดยใช้ฟังก์ชันไลน์ในรูปแบบต่างๆ ดังนี้

(1) การตอบกลับแบบ Custom Payload ซึ่งจะใช้โค้ดในรูปแบบของเจสัน ในการตอบกลับแบบ Flex Message แสดงเมนูให้เลือกถามคำถาม เช่น รายละเอียด ค่าเข้าชม เวลาเปิด-ปิด แผนที่ และ ร้านอาหารแต่ละประเภท

(2) การตอบกลับแบบ Custom Payload ซึ่งจะใช้โค้ดในรูปแบบของเจสัน โดยการส่ง Location

(3) การตอบกลับแบบ Custom Payload ในรูปแบบของเจสันเป็นแบบการส่งรูปภาพ

(4) การตอบกลับแบบข้อความจะดึงคำตอบมาจากฐานข้อมูลและเว็บไซต์ภายนอก ในการสร้างคำตอบของบอทไว้ทั้งหมด ดังนี้ สถานที่ท่องเที่ยวขอนแก่น จำแนกตามประเภท ได้แก่ ภูเขา สวนสัตว์ พิพิธภัณฑ์ อุทยานแห่งชาติ สวนน้ำ วัด สวนสาธารณะ แหล่งช้อปปิ้ง ร้านอาหารมิชลินไกด์ จำแนกประเภทตามอำเภอข้อมูลการจำแนกประเภทอำเภอ ได้แก่ อำเภอเมือง อำเภอน้ำพอง อำเภออุบลรัตน์ อำเภอสหัสขันธ์ อำเภอภูเวียง อำเภอหนองเรือ อำเภอชุมแพ อำเภอเวียงเก่า อำเภอบ้านฝาง อำเภอเขาสนกวาง อำเภอเปือยน้อย อำเภอกระนวน เวลาเปิด-ปิด รายละเอียด

สถานที่ ค่าเข้าชม รูปภาพที่แสดง และ เปิด Fulfillment ในไดอะล็อกบ็อกซ์ แต่ละอินเทนท์เพื่อเชื่อมต่อกับเว็บฮุกและส่งคำตอบที่ดึงมาจากเว็บไซต์

### 3.2.9 การเก็บข้อมูลผู้ใช้และข้อมูลคำถามคำตอบ

(1) การจัดการผู้ใช้ (User Management) ดึงข้อมูลโปรไฟล์ของผู้ใช้จากแอปพลิเคชันไลน์ (ผ่านไลน์เอพีไอ) บันทึกหรืออัปเดตข้อมูลโปรไฟล์ผู้ใช้งานในฐานข้อมูล ใช้ไลน์ไอดีเพื่อเชื่อมโยงผู้ใช้กับข้อมูลในฐานข้อมูล

(2) การดึงข้อมูลโปรไฟล์ผู้ใช้ (User Profile Fetching) ดึงข้อมูลโปรไฟล์ของผู้ใช้จากไลน์เอพีไอ

(3) การบันทึกการสนทนา (Conversation Logging) เก็บข้อมูลการสนทนาระหว่างผู้ใช้และระบบ เช่น คำถาม คำตอบ และแหล่งข้อมูล

(4) การจัดการคำตอบจากเว็บไซต์ (Web Answer Management) บันทึกคำตอบที่ดึงมาจากเว็บไซต์ลงในฐานข้อมูล (เฉพาะคำตอบที่มาจากเว็บไซต์)

เก็บข้อมูลโปรไฟล์ผู้ใช้และการสนทนาเพื่อการใช้งานในอนาคต ปรับปรุงการตอบสนองข้อมูลที่เก็บไว้สามารถนำมาปรับปรุงระบบในอนาคต เช่น การเรียนรู้เจตนาผู้ใช้เพิ่มเติม

```

const getUserProfile = async (lineId) => {
  try {
    const userLineId = String(lineId);

    if (!userLineId || typeof userLineId !== "string") {
      console.error("Invalid lineId: It should be a non-empty string.");
      return null;
    }

    const response = await axios.get(
      `https://api.line.me/v2/bot/profile/${userLineId}`,
      {
        headers: {
          Authorization: `Bearer ${process.env.LINE_CHANNEL_ACCESS_TOKEN}`,
        },
      },
    );

    if (!response.data.userId) {
      console.error("No userId found in the profile response.");
      return null;
    }

    return {
      userId: response.data.userId,
      displayName: response.data.displayName,
      pictureUrl: response.data.pictureUrl,
      statusMessage: response.data.statusMessage,
    };
  } catch (error) {
    if (error.response) {
      console.error("Error fetching user profile:", error.response.data);
    } else {
      console.error("Error fetching user profile:", error.message);
    }
    return null;
  }
};

```

ภาพที่ 7 การดึงข้อมูลโปรไฟล์ผู้ใช้

จากภาพที่ 8 อธิบายได้ว่า ฟังก์ชัน getUserProfile มีการทำงานแบบ asynchronous (ทำงานแบบไม่รอจนเสร็จ) เพื่อดึงข้อมูลโปรไฟล์ของผู้ใช้จาก LINE Messaging API ผ่านการส่งคำขอ HTTP แบบ GET ไปยัง API Endpoint



- ฟังก์ชันนี้จะรับพารามิเตอร์ `lineId` ซึ่งเป็น ID ของผู้ใช้ LINE ที่ต้องการดึงข้อมูลโปรไฟล์
- พารามิเตอร์ `lineId` ถูกแปลงเป็นสตริงด้วย `String(lineId)` เพื่อให้แน่ใจว่าเป็นประเภทข้อมูล string และตรวจสอบว่า `lineId` ไม่ว่าง (non-empty) และเป็นสตริง ถ้าไม่ตรงเงื่อนไขจะพิมพ์ข้อผิดพลาดและคืนค่า `null`
- ใช้ไลบรารี `axios` เพื่อทำ HTTP GET Request ไปที่ LINE API ด้วย URL ที่รวม `userLineId` และใช้ Authorization Header พร้อม Bearer Token (ดึงมาจาก `process.env.LINE_CHANNEL_ACCESS_TOKEN`) เพื่อยืนยันสิทธิ์ในการเข้าถึง API
- เมื่อได้รับคำตอบจาก API (`response.data`) จะตรวจสอบว่ามี `userId` ในข้อมูลหรือไม่ ถ้าไม่มี `userId` แสดงข้อผิดพลาดและคืนค่า `null`
- ถ้าผลลัพธ์ถูกต้อง จะคืนค่าวัตถุที่มีข้อมูลโปรไฟล์ของผู้ใช้ เช่น `userId`, `displayName`, `pictureUrl`, และ `statusMessage`
- ถ้าเป็นข้อผิดพลาดจากเซิร์ฟเวอร์ (มี `response` ใน `error`) จะแสดงข้อมูลจากเซิร์ฟเวอร์ (`error.response.data`) ถ้าเป็นข้อผิดพลาดทั่วไป จะแสดงข้อความข้อผิดพลาด (`error.message`) คืนค่า `null` เพื่อบ่งบอกว่าการดึงข้อมูลล้มเหลว

```

const saveUser = async (userProfile, dbClient) => {
  if (!userProfile || !userProfile.userId) {
    console.error("User profile or userId is missing.");
    return;
  }

  const query = `
INSERT INTO users (line_id, display_name, picture_url, status_message)
VALUES ($1, $2, $3, $4)
ON CONFLICT (line_id) DO UPDATE
SET display_name = $2, picture_url = $3, status_message = $4;
`;

  const values = [
    userProfile.userId,
    userProfile.displayName,
    userProfile.pictureUrl,
    userProfile.statusMessage,
  ];

  try {
    await dbClient.query(query, values);
    console.log(`User with line_id ${userProfile.userId} saved successfully.`);
  } catch (err) {
    console.error(
      `Error saving user with line_id ${userProfile.userId}:`,
      err.stack
    );
  }
};

```

ภาพที่ 8 บันทึกหรืออัปเดตข้อมูลโปรไฟล์ผู้ใช้งานในฐานข้อมูล

จากภาพที่ 9 อธิบายได้ว่า ฟังก์ชัน saveUser ทำหน้าที่บันทึกข้อมูลโปรไฟล์ผู้ใช้ไปยังฐานข้อมูล PostgreSQL โดยใช้คำสั่ง SQL INSERT หรือ UPDATE ขึ้นอยู่กับว่าข้อมูลผู้ใช้นั้นมีอยู่ในฐานข้อมูลแล้วหรือไม่

– ฟังก์ชันมีการรับพารามิเตอร์ 2 ตัว

userProfile – เก็บข้อมูลโปรไฟล์ของผู้ใช้ เช่น userId, displayName, pictureUrl, และ statusMessage

dbClient – ตัวเชื่อมต่อฐานข้อมูล (Database Client) ที่ใช้สำหรับการส่งคำสั่ง SQL ไปยังฐานข้อมูล

- ฟังก์ชันตรวจสอบว่า userProfile และ userId มีค่าหรือไม่ ถ้าค่าใดขาดหายไป จะพิมพ์ข้อผิดพลาดและยุติการทำงาน (ไม่ส่งคำสั่ง SQL)
- คำสั่ง SQL INSERT เพิ่มข้อมูลใหม่ไปยังตาราง users โดยมีคอลัมน์ line\_id, display\_name, picture\_url, และ status\_message ใช้ค่าพารามิเตอร์ \$1, \$2, \$3, \$4 เพื่อแทนค่าที่จะใส่ในแต่ละคอลัมน์ ON CONFLICT ถ้า line\_id มีอยู่ในฐานข้อมูลแล้ว จะอัปเดตค่าของ display\_name, picture\_url, และ status\_message
- สร้างอาร์เรย์ values ที่เก็บค่าจาก userProfile เพื่อใช้แทนตัวแปร \$1, \$2, \$3, \$4 ในคำสั่ง SQL
- ใช้ dbClient.query(query, values) เพื่อส่งคำสั่ง SQL และค่าพารามิเตอร์ไปยังฐานข้อมูล และใช้ await เพราะการทำงานกับฐานข้อมูลเป็น asynchronous (ใช้เวลา)
- ถ้าเกิดข้อผิดพลาดระหว่างการบันทึก จะแสดงข้อผิดพลาดพร้อมกับ line\_id และรายละเอียด (err.stack)

```

const saveConversation = async (
  questionText,
  answer,
  lineId,
  placeId,
  sourceType,
  webAnswerId,
  dbClient
) => {
  try {
    if (!dbClient) {
      console.error("Database client is not defined.");
      return;
    }

    const userId = await getUserIdFromLineId(lineId, dbClient);

    const query = `
      INSERT INTO conversations (question_text, answer_text, user_id, place_id, source_type, web_answer_id)
      VALUES ($1, $2, $3, $4, $5, $6)
      ON CONFLICT (user_id, place_id) DO UPDATE SET
        question_text = $1,
        answer_text = $2,
        source_type = $5,
        web_answer_id = $6;
    `;

    await dbClient.query(query, [
      questionText,
      answer,
      userId,
      placeId,
      sourceType,
      webAnswerId,
    ]);
  } catch (err) {
    console.error("Error saving conversation:", err);
    throw err;
  }
};

```

ภาพที่ 9 การบันทึกการสนทนา

ในภาพที่ 10 ฟังก์ชัน saveConversation ซึ่งทำหน้าที่บันทึกข้อมูลบทสนทนา (conversation) ลงในฐานข้อมูล โดย

- ฟังก์ชันรับพารามิเตอร์
  - questionText ข้อความคำถามจากผู้ใช้
  - answer ข้อความคำตอบที่สร้างขึ้น
  - lineId LINE ID ของผู้ใช้

placeId รหัสสถานที่ที่เกี่ยวข้องกับคำถาม

sourceType ประเภทของแหล่งข้อมูลที่ใช้ในการสร้างคำตอบ

webAnswerId รหัสคำตอบจากแหล่งข้อมูลบนเว็บ

dbClient - ตัวเชื่อมต่อฐานข้อมูล

- ฟังก์ชันตรวจสอบว่า dbClient ถูกส่งมาและใช้งานได้ ถ้าไม่มี dbClient จะพิมพ์ข้อผิดพลาดและยุติการทำงานทันที

- ใช้ฟังก์ชัน getUserIdFromLineId เพื่อตรวจสอบและดึง userId ที่สัมพันธ์กับ lineId จากฐานข้อมูล การดึง userId มีความจำเป็น เพราะตาราง conversations ใช้ user\_id เป็นข้อมูลอ้างอิงแทน lineId

- ใช้คำสั่ง SQL INSERT INTO เพิ่มข้อมูลใหม่ลงในตาราง conversations ใช้คำสั่ง dbClient.query เพื่อส่งคำสั่ง SQL ไปยังฐานข้อมูล พร้อมค่าพารามิเตอร์ที่ระบุในอาร์เรย์

- ใช้ await เนื่องจากการทำงานกับฐานข้อมูลเป็น asynchronous

- ถ้าการส่งคำสั่ง SQL ล้มเหลว จะแสดงข้อความข้อผิดพลาดพร้อมรายละเอียด (err) และใช้ throw err เพื่อโยนข้อผิดพลาดกลับไปให้ฟังก์ชันที่เรียกใช้สามารถจัดการเพิ่มเติมได้

```
const getUserIdFromLineId = async (lineId, dbClient) => {
  const query = "SELECT id FROM users WHERE line_id = $1";
  const result = await dbClient.query(query, [lineId]);
  console.log("Fetched user ID from database:", result.rows);

  if (result.rows.length > 0) {
    return result.rows[0].id;
  } else {
    throw new Error("User not found in database");
  }
};
```

ภาพที่ 10 การดึงไลน์ไอดีจากฐานข้อมูล

จากภาพที่ 11 อธิบายว่าฟังก์ชัน getUserIdFromLineId ใช้สำหรับดึง User ID จากฐานข้อมูลโดยอ้างอิงจาก LINE ID

- โดยฟังก์ชันรับพารามิเตอร์ 2 ตัว คือ lineId - LINE ID ของผู้ใช้ที่ต้องการค้นหาในฐานข้อมูล,
- dbClient - ตัวเชื่อมต่อฐานข้อมูล (Database Client) ที่ใช้สำหรับส่งคำสั่ง SQL
- สร้างคำสั่ง SQL เพื่อค้นหา id (User ID) จากตาราง users โดยเงื่อนไขคือ line\_id ตรงกับ \$1 ซึ่ง \$1 เป็น placeholder สำหรับค่าที่จะใส่ในคำสั่ง SQL เพื่อป้องกัน SQL Injection
  - ใช้คำสั่ง SQL dbClient.query เพื่อส่งคำสั่ง SQL ไปยังฐานข้อมูล ใส่ค่า lineId ในอาร์เรย์ [lineId] เพื่อแทน \$1 ในคำสั่ง SQL
  - ผลลัพธ์จากฐานข้อมูล (result.rows) เป็นอาร์เรย์ที่แต่ละแถวเก็บผลลัพธ์ของคำสั่ง SQL
  - ถ้ามีแถว (rows.length > 0) จะส่งคืน id ของแถวแรก (result.rows[0].id)
  - ถ้าไม่มีข้อมูลในฐานข้อมูล (rows.length === 0) จะแสดงข้อผิดพลาด "User not found in database" เพื่อให้ฟังก์ชันที่เรียกใช้รู้ว่าข้อมูลไม่ถูกพบ

```
const saveWebAnswer = async (
  answerText,
  placeName,
  intentType,
  isFromWeb,
  dbClient
) => {
  try {
    if (isFromWeb && answerText && answerText.trim() !== "") {
      const query = `
        INSERT INTO web_answer (answer_text, place_name, intent_type)
        VALUES ($1, $2, $3)
      `;
      const values = [answerText, placeName, intentType];
      await dbClient.query(query, values);
      console.log("Saved answer from website to the database.");
    } else {
      console.log("This answer was not from a website, not saving.");
    }
  } catch (err) {
    console.error("Error saving web answer:", err.stack);
  }
};
```

ภาพที่ 11 บันทึกคำตอบที่ดึงมาจากเว็บไซต์ลงในฐานข้อมูล

จากภาพที่ 12 อธิบายได้ว่า saveWebAnswer เป็นฟังก์ชันสำหรับบันทึกคำตอบที่มาจากเว็บไซต์ลงในฐานข้อมูล โดยตรวจสอบข้อมูลก่อนว่าคำตอบนั้นมาจากเว็บไซต์และมีเนื้อหาไม่ว่างเปล่า โดยฟังก์ชันรับพารามิเตอร์ 5 ตัว

- มีการตรวจสอบเงื่อนไขการบันทึก ตรวจสอบว่าคำตอบมาจากเว็บไซต์หรือไม่ (true) ตรวจสอบว่าคำตอบ (answerText) มีค่า (ไม่ใช่ null หรือ undefined) ตรวจสอบว่า answerText ไม่ใช่สตริงว่างเปล่า (หลังตัดช่องว่างหน้าหลังด้วย .trim())

- ถ้าเงื่อนไขผ่าน ฟังก์ชันจะดำเนินการบันทึกข้อมูลลงฐานข้อมูล
- ถ้าเงื่อนไขไม่ผ่าน จะแสดงข้อความ: "This answer was not from a website, not saving." และหยุดการทำงานโดยไม่บันทึกข้อมูล

- ใช้คำสั่ง SQL เพิ่มข้อมูลใหม่ลงในตาราง web\_answer

### 3.2.10 การดึงข้อมูลมาใช้ในการตอบคำถามจากฐานข้อมูล

(1) Intent Mapping คือการจับคู่คำถามของผู้ใช้กับ Intent และข้อมูลในฐานข้อมูล ตรวจสอบคำถามของผู้ใช้และดูว่าอินเทนต์ใด ชื่อสถานที่ที่เป็นที่ใดในไดอะล็อกโฟลว์ที่สอดคล้องกับคำถามนั้นใช้ชื่อสถานที่และชื่ออินเทนต์เพื่อดึงคำตอบที่เหมาะสมจากฐานข้อมูล

(2) เมื่อผู้ใช้ส่งคำถามเข้ามาผ่านแอปพลิเคชันไลน์ข้อมูลจะถูกส่งไปยังเซิร์ฟเวอร์ที่ทำงานด้วย โหนดเจเอส จากนั้น โหนดเจเอส จะส่งต่อคำถามนี้ไปยัง ไดอะล็อกโฟลว์ ซึ่งมีการตั้งค่าการจับคู่เจตนาของคำถาม (Intent Matching) อยู่ โดย ไดอะล็อกโฟลว์ จะใช้กระบวนการ Natural Language Processing (NLP) ในการแยกและวิเคราะห์ข้อความเพื่อจับคู่กับ อินเทนต์ (Intent) ที่ได้ถูกสร้างไว้ล่วงหน้า

(3) อินเทนต์ คือการจัดกลุ่มคำถามที่มีเจตนาเดียวกัน เช่น คำถามเกี่ยวกับที่พัก สถานที่ท่องเที่ยว หรือร้านอาหาร คำถามที่ถูกตรวจพบว่าตรงกับอินเทนต์จะถูกนำไปประมวลผลเพื่อค้นหาคำตอบต่อไป

(4) หลังจากที่ ไดอะล็อกโฟลว์ จับคู่เจตนาของคำถามได้แล้ว ระบบจะทำการส่งพารามิเตอร์ไปยังเซิร์ฟเวอร์ จะนำชื่อสถานที่และอินเทนต์ไปค้นหาคำตอบในฐานข้อมูลโดยที่จะค้นหาจากฐานข้อมูลแรกก่อนคือฐานข้อมูลที่ทำเองที่เพิ่มไว้ในระบบ ถ้าไม่พบคำตอบระบบจะทำการไปค้นหาในฐานข้อมูลที่ทำจากเว็บไซต์ หลังจากดึงข้อมูลคำตอบได้แล้ว ระบบจะแยกคำตอบเป็นหมวดหมู่ เช่น ค่าเช่า, เบอร์โทร, เวลาเปิดทำการ หรืออื่น ๆ หากคำถามที่ถูกส่งมาเกี่ยวข้องกับข้อมูลที่มีอยู่ใน ฐานข้อมูล (Database) ที่ถูกเตรียมไว้ ระบบจะดึงข้อมูลเหล่านั้นออกมาเพื่อเป็นคำตอบสำหรับคำถามของผู้ใช้

(5) หลังจากที่ข้อมูลถูกฟิลเตอร์และเตรียมพร้อมแล้ว คำตอบจะถูกส่งกลับไปยัง โหนดเจเอส ซึ่งจะทำหน้าที่เชื่อมต่อกับแอปพลิเคชันไลน์อีกครั้งเพื่อตอบกลับผู้ใช้ ข้อมูลที่ได้จะปรากฏในรูปแบบข้อความหรือรูปภาพที่ถูกส่งไปยังแอปพลิเคชันไลน์โดยผู้ใช้จะสามารถอ่านและใช้งานข้อมูลได้อย่างสะดวก

การดึงข้อมูลจากฐานข้อมูลทำงานโดยเริ่มจากการจับคู่คำถามกับอินเทนต์ที่ตั้งไว้ จากนั้นจะนำชื่อสถานที่และอินเทนต์ไปทำการค้นหาคำตอบในฐานข้อมูล และฟิลเตอร์ข้อมูลเพื่อให้เหมาะสมกับคำถามที่ถูกส่งมา กระบวนการทั้งหมดนี้เกิดขึ้นในระยะเวลาอันรวดเร็ว ทำให้ผู้ใช้สามารถได้รับข้อมูลที่ต้องการดังนี้

```
const getAnswerForIntent = async (intentName, placeName, dbClient) => {
  if (!placeName) {
    console.log("No placeName provided");
    return { answer: null, placeId: null };
  }

  console.log(`Intent Name: ${intentName}, Place Name: ${placeName}`);

  const queries = {
    ค่าธรรมเนียมการเข้า:
      "SELECT admission_fee AS answer, id AS place_id FROM places WHERE name LIKE $1",
    เส้นทางไปยังสถานที่:
      "SELECT address AS answer, id AS place_id FROM places WHERE name LIKE $1",
    เบอร์โทร:
      "SELECT contact AS answer, id AS place_id FROM places WHERE name LIKE $1",
    รายละเอียด:
      "SELECT description AS answer, id AS place_id FROM places WHERE name LIKE $1",
  };

  try {
    const query = queries[intentName] || null;
    if (!query) {
      console.log(`No query found for intent: ${intentName}`);
      return { answer: null, placeId: null };
    }
  }

  console.log(`Running query: ${query}`);
}
```

ภาพที่ 12 การดึงข้อมูลจากตาราง place



```

const result = await dbClient.query(query, [`%${placeName}%`]);
console.log(`Database query result:`, result.rows);

if (result.rows.length > 0) {
  let answerText = result.rows[0].answer;
  const placeId = result.rows[0].place_id;

  const filteredAnswer = {
    fee: null,
    contact: null,
    openingHours: null,
    other: null,
  };

  if (intentName === "ค่าธรรมเนียมการเข้า") {
    filteredAnswer.fee = answerText
      ? answerText.trim()
      : "ไม่พบข้อมูลค่าธรรมเนียมการเข้า";
  } else if (intentName === "เส้นทางไปยังสถานที่") {
    filteredAnswer.path = answerText
      ? answerText.trim()
      : "ไม่พบข้อมูลเส้นทางไปยังสถานที่";
  } else if (intentName === "เบอร์โทร") {
    filteredAnswer.contact = answerText
      ? answerText.trim()
      : "ไม่พบข้อมูลเบอร์โทร";
  } else if (intentName === "รายละเอียด") {
    filteredAnswer.detail = answerText
      ? answerText.trim()
      : "ไม่พบข้อมูลรายละเอียด";
  }

  console.log("Filtered answer:", filteredAnswer);

  return {
    answer: filteredAnswer,
    placeId,
  };
}

console.log("No data found for the place");
return { answer: null, placeId: null };
} catch (error) {
  console.error("Error fetching data from places table:", error.stack);
  throw new Error("Database query error");
}
};

```

ภาพที่ 13 การดึงข้อมูลจากตาราง place (ต่อ)

ในภาพที่ 13 และ 14 ฟังก์ชัน `getAnswerForIntent` ทำหน้าที่ดึงข้อมูลจากฐานข้อมูล PostgreSQL ตาม Intent Name (ความตั้งใจของคำถาม) และชื่อสถานที่ที่ผู้ใช้ระบุ ฟังก์ชันรับพารามิเตอร์ 3 มาตัว

- ตรวจสอบว่าพารามิเตอร์ `placeName` มีค่า ถ้าไม่มีค่า (null หรือ undefined)
- ใช้คำสั่ง SQL queries เป็นออบเจกต์ที่เก็บคำสั่ง SQL สำหรับ Intent ต่าง ๆ ใช้คอลัมน์ name ของตาราง `places` ร่วมกับเงื่อนไข `LIKE` เพื่อค้นหาสถานที่ที่ชื่อคล้ายกับ `placeName` โดยใช้ `%` เป็น Wildcard
- ตรวจสอบว่ามีคำสั่ง SQL สำหรับ `intentName` หรือไม่ ถ้าไม่มีคำสั่ง SQL ที่ตรงกับ `intentName` แสดงข้อความแจ้งว่าไม่มีคำสั่งสำหรับ Intent นั้น และคืนค่า { `answer: null, placeId: null` }
- ใช้ `dbClient.query` ส่งคำสั่ง SQL (query) พร้อมพารามิเตอร์ `placeName` โดยเพิ่ม `%` เพื่อให้รองรับการค้นหาที่มีส่วนเหมือน (Partial Match) ผลลัพธ์จะถูกเก็บใน `result.rows` ซึ่งเป็นอาร์เรย์ของแถวข้อมูลที่ได้มาจากการผลลัพธ์จากฐานข้อมูล กรณีพบข้อมูล จะดึงค่าจากคอลัมน์ `answer` และ `place_id` จากแถวแรก (`result.rows[0]`) กรณีแปลงผลลัพธ์ตาม Intent ลบช่องว่างหน้าหลังด้วย `.trim()` ถ้าไม่มีข้อมูล (null หรือว่างเปล่า) แสดงข้อความ "ไม่พบข้อมูล..." และกรณีไม่พบข้อมูล ข้อความว่าไม่พบข้อมูลสำหรับสถานที่ และคืนค่า { `answer: null, placeId: null` }

```

const getAnswerFromWebAnswerTable = async (intentType, placeName, dbClient) => {
  if (!dbClient) {
    throw new Error("Database client is not initialized.");
  }

  if (!placeName || !intentType) {
    return { answer: null, placeId: null };
  }

  const query = `
    SELECT answer_text AS answer, id AS place_id
    FROM web_answer
    WHERE place_name = $1 AND intent_type = $2
  `;

  try {
    const result = await dbClient.query(query, [placeName, intentType]);

    if (result.rows.length > 0) {
      const answerText = result.rows[0].answer;

      const answer = {
        fee: null,
        contact: null,
        openingHours: null,
        other: null,
      };

      if (answerText.includes("ค่าเช่า")) {
        answer.fee = answerText.match(/ค่าเช่า(?:[^:]*:)?(?:[^\s]*)/)[0].trim();
      }
      if (answerText.includes("เบอร์โทร")) {
        answer.contact = answerText.match(/เบอร์โทร(?:[^:]*:)?(?:[^\s]*)/)[0].trim();
      }
      if (answerText.includes("เวลาเปิดทำการ")) {
        answer.openingHours = answerText
          .match(/เวลาเปิดทำการ(?:[^:]*:)?(?:[^\s]*)/)[0]
          .trim();
      }

      if (!answer.fee && !answer.contact && !answer.openingHours) {
        answer.other = answerText;
      }

      return {
        answer,
        placeId: result.rows[0].place_id,
      };
    } else {
      return { answer: null, placeId: null };
    }
  } catch (error) {
    console.error("Error fetching answer from web_answer table:", error.stack);
    throw new Error("Database query error");
  }
};

```

ภาพที่ 14 การดึงข้อมูลจากตาราง web\_answer

จากภาพที่ 15 ฟังก์ชัน `getAnswerFromWebAnswerTable` ทำหน้าที่ดึงข้อมูลคำตอบจากฐานข้อมูล โดยมีการตรวจสอบค่าที่เกี่ยวข้องและประมวลผลคำตอบให้อยู่ในรูปแบบที่ใช้งานได้ง่ายขึ้น

- ตรวจสอบค่าเริ่มต้นโดย
  - if (!dbClient): ตรวจสอบว่ามีการส่ง dbClient (Client ของฐานข้อมูล) มาหรือไม่ ถ้าไม่มีจะโยนข้อผิดพลาด (throw) ว่า "Database client is not initialized."
  - if (!placeName || !intentType): ตรวจสอบว่ามีค่าทั้ง placeName (ชื่อสถานที่) และ intentType (ประเภทข้อมูลที่ต้องการ เช่น เวลาทำการ, ค่าเช่า) หรือไม่ ถ้าไม่มีจะคืนค่าที่ไม่มีคำตอบ (answer: null, placeld: null)
  - สร้างคำสั่ง SQL เพื่อดึงข้อมูลจากราย web\_answer
  - ประมวลผลคำตอบจากฐานข้อมูล await dbClient.query(query, [placeName, intentType]): เรียกใช้คำสั่ง SQL และส่งค่าพารามิเตอร์ placeName และ intentType
  - ถ้าพบผลลัพธ์ (จำนวนแถวมากกว่า 0) ให้ดึงข้อความคำตอบ (answer\_text) จากแถวแรก (result.rows[0])
    - Answer สร้างออบเจกต์เปล่าเพื่อเก็บคำตอบในแต่ละประเภท
    - ตรวจสอบว่า answerText มีค่าที่ระบุข้อมูล โดยใช้ Regular Expression
    - ถ้าทั้งหมดไม่มีข้อมูล จะเก็บข้อความทั้งหมดใน other
    - คืนค่าคำตอบ (answer) และ placeld ของสถานที่

```

const handleIntent = async (
  agent,
  dbClient,
  questionText,
  location = "",
  displayName = ""
) => {
  const intentName = agent.request_.body.queryResult.intent.displayName;
  const lineId = agent.originalRequest.payload.data.source.userId;

  let answer = "";
  let placeId = null;
  let sourceType = "";
  let answerText = "";
  let isFromWeb = false;
  let webAnswerId = null;
  let responseMessage = "";

  try {
    console.log("Handling intent with location filtering...");

    const userProfile = await getUserProfile(lineId);
    console.log("User Profile:", userProfile);

    if (userProfile) {
      await saveUser(userProfile, dbClient);
    }

    let placeName = location;
    if (!location) {
      console.log("No location provided, extracting from text...");
      const apiKey = "AIzaSyDhNISSRJNm2AEVQLFbTnze3I8BuABUqGw";
      placeName = await extractPlaceFromText(questionText, apiKey);

      if (!placeName) {
        console.log("No location found from text. Sending default response.");
        agent.add("ไม่พบข้อมูลสถานที่ที่เกี่ยวข้องกับคำถามของคุณ.");
        return;
      }

      console.log("Place extracted successfully:", placeName);
    }

    console.log(`Intent Name: ${intentName}, Place Name: ${placeName}`);
  }
}

```

ภาพที่ 15 จัดการ Intent หลักที่มาจาก Dialogflow

```

const dbResult = await getAnswerForIntent(intentName, placeName, dbClient);

if (dbResult && dbResult.answer) {
  answer = dbResult.answer;
  placeId = dbResult.placeId;
  console.log("Found answer from database:", answer);
  sourceType = "database";
  isFromWeb = false;

  if (intentName === "ค่าธรรมเนียมการเข้า") {
    responseMessage = answer.fee || "ไม่พบข้อมูลค่าธรรมเนียมการเข้า";
  } else if (intentName === "เส้นทางไปยังสถานที่") {
    responseMessage = answer.path || "ไม่พบข้อมูลเส้นทางไปยังสถานที่";
  } else if (intentName === "เบอร์โทร") {
    responseMessage = answer.contact || "ไม่พบข้อมูลเบอร์โทร";
  } else if (intentName === "รายละเอียด") {
    await PlaceDetailsIntent(
      agent,
      dbClient,
      questionText,
      location,
      lineId
    );
    return;
  } else if (intentName === "เวลาเปิดทำการ") {
    responseMessage = answer.openingHours || "ไม่พบข้อมูลเวลาเปิดทำการ";
  }
} else {
  // Query the secondary database
  const webResult = await getAnswerFromWebAnswerTable(
    intentName,
    placeName,
    dbClient
  );
};

```

ภาพที่ 16 จัดการ Intent หลักที่มาจาก Dialogflow (ต่อ)

```

if (webResult.answer) {
  const filteredAnswer = webResult.answer;
  console.log("Filtered Answer from DB:", filteredAnswer);

  if (filteredAnswer.fee) {
    responseMessage = `${filteredAnswer.fee}`;
  } else if (filteredAnswer.contact) {
    responseMessage = `${filteredAnswer.contact}`;
  } else if (filteredAnswer.openingHours) {
    responseMessage = `${filteredAnswer.openingHours}`;
  } else if (filteredAnswer.other) {
    responseMessage = filteredAnswer.other;
  } else {
    responseMessage = "ข้อมูลไม่ครบถ้วน";
  }

  placeId = null;
  webAnswerId = webResult.placeId;
  sourceType = "web_database";
  isFromWeb = true;
}

```

ภาพที่ 17 จัดการ Intent หลักที่มาจาก Dialogflow (ต่อ)

จากภาพที่ 16, 17 และ 18 ฟังก์ชัน `handleIntent` ซึ่งทำหน้าที่ตอบคำถามของผู้ใช้ โดยอ้างอิงจาก intent (วัตถุประสงค์ของคำถาม) และ สถานที่ ที่ระบุในคำถาม โดยมีการเชื่อมต่อกับฐานข้อมูลและการประมวลผลข้อมูลจากหลายแหล่ง รวมถึงการบันทึกประวัติการสนทนาและคำตอบลงในฐานข้อมูล

- ดึงข้อมูล Intent และ LINE ID โดย `intentName` ดึงชื่อ intent ของคำถามจากคำขอที่ส่งมา `lineId` ดึง LINE ID ของผู้ใช้จากคำขอที่ส่งมาผ่าน LINE
- ดึงข้อมูลโปรไฟล์ผู้ใช้จาก LINE API ด้วยฟังก์ชัน `getUserProfile` ถ้าได้โปรไฟล์ผู้ใช้ จะบันทึกข้อมูลลงในฐานข้อมูลด้วย `saveUser`
- ตรวจสอบและดึงชื่อสถานที่ ถ้าไม่มีสถานที่ (location) ที่ระบุ จะใช้ฟังก์ชัน `extractPlaceFromText` เพื่อดึงชื่อสถานที่จากข้อความคำถาม ถ้าไม่พบชื่อสถานที่ จะส่งข้อความแจ้งผู้ใช้และยุติการทำงาน
- เรียกใช้ฟังก์ชัน `getAnswerForIntent` เพื่อดึงคำตอบและข้อมูลจากฐานข้อมูล ถ้าพบคำตอบ เก็บข้อมูลคำตอบในตัวแปร ระบุ `sourceType` เป็น `database` ตั้งค่า `isFromWeb` เป็น `false`
- ถ้าคำตอบไม่มีในฐานข้อมูลหลัก จะดึงข้อมูลจากตาราง `web_answer` มาตอบ

### 3.2.11 การดึงข้อมูลมาใช้ในการตอบคำถามจากเว็บไซต์ภายนอก

การดึงข้อมูลมาใช้ในการตอบคำถามจากเว็บไซต์ภายนอกโดยใช้ เทคนิคการดึงข้อมูลจากเว็บไซต์ (Web Scraping) ในกรณีที่ระบบพบว่าไม่มีข้อมูลที่ตรงกับคำถามของผู้ใช้ในระบบจะดึงข้อมูลภายใน ระบบจะใช้การ เทคนิคการดึงข้อมูลจากเว็บไซต์ (Web Scraping) เพื่อดึงข้อมูลจากเว็บไซต์ภายนอก โดยผ่านกระบวนการดังนี้

#### หลักการทำงานหลัก

(1) การระบุเจตนา (Intent Recognition): ระบบจะระบุเจตนาของผู้ใช้ (Intent) เช่น การถามเกี่ยวกับ "เวลาเปิดทำการ" หรือ "ค่าธรรมเนียมการเข้า" ระบบเชื่อมโยงเจตนาเข้ากับข้อมูลที่เก็บไว้ในฐานข้อมูลหรือแหล่งข้อมูลภายนอก

(2) การระบุคำสำคัญ (Keyword Extraction): ระบบจะดึงคำสำคัญจากคำถามของผู้ใช้ เช่น "ค่าธรรมเนียมเข้าชม" หรือ "เวลาเปิดทำการ" ใช้เทคนิค TF-IDF (Term Frequency-Inverse Document Frequency) เพื่อจัดลำดับความสำคัญของคำสำคัญในคำถาม

(3) การกรองข้อมูล (Data Filtering): ข้อมูลที่ค้นพบจะถูกกรองและจัดหมวดหมู่ เช่น ค่าเข้า, เวลาเปิดทำการ, หรือเบอร์โทรศัพท์ ใช้การจับคู่ข้อความแบบละเอียด เช่น Regular Expressions เพื่อระบุรูปแบบของคำตอบ

(4) การรวมและสรุปคำตอบ (Answer Summarization): หากคำตอบมีความยาวเกินไป ระบบจะสรุปข้อมูลโดยตัดข้อความที่เกินความจำเป็น หากคำตอบมาจากหลายแหล่ง ระบบจะรวมคำตอบและลำดับความสำคัญ

#### เทคนิคที่ใช้ในแต่ละขั้นตอน

(1) การดึงคำสำคัญ (Keyword Extraction) ใช้เทคนิค TF-IDF: ใช้ในการวิเคราะห์ความสำคัญของคำในเอกสาร โดยคำที่ปรากฏบ่อยในคำถามแต่ไม่บ่อยในข้อมูลทั้งหมด จะมีคะแนน TF-IDF สูง ตัวอย่าง: คำว่า "เวลาเปิดทำการ" อาจมีคะแนน TF-IDF สูงในคำถามเกี่ยวกับเวลา แต่ไม่เกี่ยวข้องกับคำถามอื่น

(2) การประมวลผลคำ: แยกคำในประโยคด้วย Wordcut หรือ Tokenization ทำความสะอาดคำ เช่น การลบช่องว่างและแปลงคำให้เป็นตัวพิมพ์เล็ก (Normalization) ตัวอย่างผลลัพธ์: ["เวลา", "เปิด", "ทำการ"]

(3) การกรองและจัดหมวดหมู่ข้อมูล (Data Filtering) การกรองด้วยคีย์เวิร์ด: ใช้คำสำคัญที่ดึงมา เช่น "ค่าเข้า", "เวลาเปิด" เพื่อกรองข้อมูลจากคำตอบทั้งหมด ใช้ชุดคำหลักที่เกี่ยวข้อง (Keyword Groups) เพื่อเพิ่มความแม่นยำ คำที่เกี่ยวกับ "เวลา": ["เวลาเปิด", "ปิด", "เวลาทำการ"] คำที่เกี่ยวกับ "ค่าเข้า": ["ค่าเข้า", "ค่าธรรมเนียม", "ราคา ค่าเข้าชม"] หรือใช้ การใช้ Regular Expressions ตรวจสอบรูปแบบข้อความ

(4) การรวมข้อมูลจากหลายแหล่ง (Data Merging) ดึงข้อมูลด้วย Web Scraping (เช่น Cheerio) จัดเก็บข้อมูลในรูปแบบ JSON สำหรับค้นหาในอนาคตหากพบข้อมูลในหลายแหล่ง รวมข้อมูลจากทุกแหล่ง และเรียงลำดับตามความสำคัญ ใช้เทคนิค Fuzzy Matching เพื่อตรวจสอบความใกล้เคียงของชื่อสถานที่

(5) การสรุปคำตอบ (Answer Summarization) หากข้อความยาวเกินไป (เช่น เกิน 200 อักขระ) ระบบจะตัดข้อความและเพิ่ม "..." เพื่อให้เข้าใจง่าย

#### วิธีการทำงาน



(1) ส่งคำขอ (Request) ไปยังเว็บไซต์เมื่อ Intent ที่จับคู่มาไม่พบข้อมูลในฐานข้อมูล Node.js server จะเริ่มกระบวนการ การดึง HTML จากเว็บไซต์ระบบจะส่งคำขอ (HTTP Request) ไปยัง URL ของหน้าเว็บที่ต้องการดึงข้อมูลเมื่อได้รับ HTML ของหน้าเว็บ ระบบจะดึงโครงสร้างทั้งหมดของหน้าเว็บมาเพื่อการประมวลผล

(2) การวิเคราะห์โครงสร้าง HTML ระบบใช้เครื่องมือวิเคราะห์โครงสร้าง HTML (เช่น Cheerio) เพื่อเลือกข้อมูลที่ต้องการ เช่น ชื่อสถานที่ รายละเอียด รูปภาพ ฯลฯ ข้อมูลที่ต้องการจะถูกค้นหาโดยการระบุตำแหน่ง (Selectors) เช่น ชื่อสถานที่ที่จะถูกค้นจาก <h2> หรือรูปภาพจาก <img> ที่เกี่ยวข้อง

(3) การดึงข้อมูลตามหมวดหมู่ ระบบจะแบ่งข้อมูลออกเป็นหมวดหมู่ เช่น: ชื่อสถานที่: ดึงจากหัวข้อ หรือแท็กที่เกี่ยวข้อง เช่น <h2> หรือ <h3> รายละเอียด: ดึงข้อความจากแท็กที่อยู่ถัดจากชื่อ เช่น <p> รูปภาพ: ดึง URL ของรูปภาพจากแท็ก <img> ข้อมูลเพิ่มเติม: เช่น รายการจุดเด่นหรือคำแนะนำ ดึงจาก <ul> หรือ <li>

(4) การจัดรูปแบบข้อมูล ข้อมูลที่ดึงมาจะถูกจัดให้อยู่ในรูปแบบที่เหมาะสม เช่น: รวมรูปภาพในรูปแบบของลิสต์ (Array) สร้างโครงสร้างข้อมูลที่ชัดเจน เช่น "ชื่อสถานที่", "รายละเอียด", "รูปภาพ", "ข้อมูลเพิ่มเติม" ข้อมูลจะถูกตรวจสอบเพื่อจัดการกับกรณีที่ไม่มีข้อมูล เช่น การเพิ่มข้อความว่า "ไม่มีรูปภาพ" หรือ "ไม่มีรายละเอียด"

(5) การจัดเก็บข้อมูล ข้อมูลที่ดึงมาและจัดหมวดหมู่แล้วจะถูกจัดเก็บในไฟล์ JSON เพื่อให้พร้อมใช้งานในกระบวนการค้นหา การจัดเก็บในรูปแบบ JSON ช่วยให้ข้อมูลสามารถนำไปใช้ต่อในระบบอื่น ๆ ได้อย่างง่ายดาย เช่น การค้นหาคำตอบในอนาคต

(6) การรวมข้อมูลจากหลายเว็บไซต์ หากต้องการดึงข้อมูลจากหลายแหล่ง ระบบจะประมวลผลแต่ละ URL และรวมข้อมูลทั้งหมดเข้าด้วยกัน กระบวนการรวมข้อมูลจะจัดการกับข้อมูลซ้ำซ้อน เพื่อให้ได้คำตอบที่ครบถ้วนและชัดเจนที่สุด

(7) การเตรียมข้อมูลสำหรับค้นหา หลังจากข้อมูลถูกดึงและจัดเก็บ ระบบจะใช้ข้อมูลนี้ในกระบวนการค้นหาคำตอบเมื่อผู้ใช้ส่งคำถามที่เกี่ยวข้อง ระบบจะเชื่อมโยงข้อมูลที่ดึงมาจากเว็บกับคำถามของผู้ใช้ เช่น ชื่อสถานที่หรือรายละเอียดคำถาม เพื่อให้คำตอบตรงประเด็น

(8) ข้อมูลในไฟล์ JSON จะถูกใช้ในกระบวนการค้นหาคำตอบ ระบบจะเชื่อมโยงคำถามของผู้ใช้กับข้อมูลในไฟล์ JSON หากคำถามเกี่ยวข้องกับข้อมูลที่มีอยู่ใน JSON เช่น ชื่อสถานที่หรือรายละเอียด ระบบจะใช้ข้อมูลนั้นเพื่อตอบคำถามทันที

#### การดึงคำตอบ

(1) ระบุสถานที่และเจตนา (Intent) โดยจะทำการตรวจสอบสถานที่ (Location) : หากผู้ใช้ระบุสถานที่มาในคำถาม ระบบจะใช้ชื่อสถานที่นั้นในการค้นหาข้อมูล หากไม่มีสถานที่ในคำถาม ระบบจะพยายามดึงชื่อสถานที่จากคำถามโดยใช้ API สำหรับวิเคราะห์ข้อความ เช่น Google Places API หรือฟังก์ชันวิเคราะห์ข้อความ ตรวจสอบ Intent : Intent คือจุดประสงค์ของคำถาม เช่น "เวลาเปิดทำการ", "ค่าธรรมเนียมการเข้า", หรือ "รายละเอียด" ระบบจะใช้ Intent นี้เพื่อระบุประเภทข้อมูลที่ต้องการดึงจากเว็บไซต์

(2) ค้นหาข้อมูลในแหล่งข้อมูลเว็บ ค้นหาในไฟล์ JSON ที่มีข้อมูลที่เก็บมาล่วงหน้า: ระบบจะโหลดข้อมูลจากไฟล์ JSON (ที่ดึงมาจากการ Web Scraping ล่วงหน้า) เช่น data1.json และ data2.json ข้อมูลในไฟล์ JSON จะประกอบด้วยสถานที่, รายละเอียด, และข้อมูลอื่น ๆ ที่จัดเก็บในโครงสร้างชัดเจน และทำการกรองข้อมูลที่เกี่ยวข้อง หากพบชื่อสถานที่ในข้อมูล ระบบจะกรองเฉพาะข้อมูลที่เกี่ยวข้องกับสถานที่นั้น หากไม่พบชื่อสถานที่ ระบบจะพยายามค้นหาสถานที่ที่ใกล้เคียงที่สุดโดยใช้ Fuzzy Matching

(3) ฟิวเตอร์ข้อมูลตามอินเทอร์เน็ต เมื่อพบข้อมูลที่เกี่ยวข้องกับสถานที่ ระบบจะกรองข้อมูลตามประเภทของอินเทอร์เน็ตและใช้ Keyword Groups เพื่อระบุคำสำคัญสำหรับการกรองข้อมูลที่เกี่ยวข้องในแต่ละหมวดหมู่

(4) จัดการกรณีไม่พบข้อมูล หากยังไม่พบข้อมูล ระบบจะแจ้งผู้ใช้ว่า "ไม่พบข้อมูลที่เกี่ยวข้อง"

(5) สรุปคำตอบและส่งกลับ ข้อมูลที่ผ่านการฟิวเตอร์จะถูกจัดรูปแบบให้เหมาะสม เช่น การรวมคำตอบในรูปแบบข้อความ หรือการแนบข้อมูลเพิ่มเติม เช่น รูปภาพหรือลิงก์ คำตอบที่ได้จะถูกส่งกลับไปยังผู้ใช้ผ่านไลน์แพลตฟอร์ม

การดึงข้อมูลจากเว็บไซต์ภายนอกโดยใช้ เทคนิคการดึงข้อมูลจากเว็บไซต์ (Web Scraping) ในระบบแชทบอทที่เกี่ยวข้องกับสถานที่ท่องเที่ยวในขอนแก่นนั้นทำงานตามกระบวนการที่อธิบายไว้ข้างต้น เมื่อระบบไม่พบข้อมูลในฐานข้อมูลภายใน ระบบจะส่งคำขอไปยังเว็บไซต์ภายนอกเพื่อดึงข้อมูลที่เกี่ยวข้องและส่งกลับไปยังผู้ใช้ ซึ่งช่วยให้แชทบอทสามารถตอบคำถามผู้ใช้ได้อย่างครอบคลุมแม้ว่าข้อมูลนั้นจะไม่อยู่ในฐานข้อมูลของระบบเอง ตามวิธีทำดังนี้

```

const fetchHTMLAndSaveToJSON1 = async (url, outputFilePath) => {
  try {
    console.log(`Fetching HTML from: ${url}`);
    const { data: html } = await axios.get(url);
    console.log("Fetched HTML successfully.");

    const $ = cheerio.load(html);
    let results = [];

    $("h2").each((i, el) => {
      const locationName = $(el).text().trim();
      const locationDetailImg = $(el).nextAll("p").eq(1).text().trim();
      const listImg = $(el)
        .nextUntil("h2", "p")
        .find("img")
        .map((i, img) => $(img).attr("src").trim())
        .get();

      const locationDetail = $(el).next("p").text().trim();
      const listItems = $(el)
        .nextUntil("h2", "ul")
        .find("li")
        .map((i, li) => $(li).text().trim())
        .get();

      results.push({
        สถานที่: locationName,
        รูปภาพ: listImg,
        รายละเอียดรูปภาพ: locationDetailImg,
        รายละเอียด: locationDetail,
        ข้อมูลที่ค้นพบ: listItems,
      });
    });

    if (results.length === 0) {
      console.log("No data found. Please check the website structure.");
      return;
    }

    fs.writeFileSync(outputFilePath, JSON.stringify(results, null, 2), "utf8");
    console.log(`Data saved to ${outputFilePath}`);
  } catch (error) {
    console.error("Error fetching and saving data:", error);
  }
};

```

ภาพที่ 18 การดึงข้อมูลจากเว็บ

จากรูปที่ 19 ฟังก์ชัน fetchHTMLAndSaveToJSON1 ซึ่งมีหน้าที่หลักคือดึงข้อมูลจากหน้าเว็บ (Web Scraping) และจัดเก็บข้อมูลในรูปแบบไฟล์ JSON โดยใช้ไลบรารี axios, cheerio, และ fs สำหรับการประมวลผล โค้ดนี้สามารถแบ่งการทำงานได้เป็นขั้นตอนดังนี้

- รับ Input และเตรียมการ รับพารามิเตอร์ url: URL ของเว็บไซต์ที่ต้องการดึงข้อมูล
- outputFilePath: ที่อยู่ของไฟล์ JSON ที่จะบันทึกข้อมูล
- ใช้ **axios** ดึงเนื้อหา HTML ของเว็บไซต์จาก URL ที่ระบุ และเก็บผลลัพธ์ในตัวแปร html
- โหลด HTML ที่ได้มาด้วยไลบรารี **cheerio** เพื่อให้สามารถค้นหาและจัดการกับข้อมูลใน HTML ได้ง่ายขึ้น (คล้ายกับ jQuery)
- ค้นหาและประมวลผลข้อมูล ค้นหาแท็ก <h2> ซึ่งคาดว่าเป็นหัวข้อของสถานที่ใน HTML
- ตรวจสอบผลลัพธ์ หาก results ไม่มีข้อมูล (length = 0) จะแสดงข้อความเตือนและจบการทำงาน
- ใช้ fs.writeFileSync เพื่อเขียนข้อมูลใน results เป็นไฟล์ JSON ที่ระบุใน outputFilePath
- ใช้ JSON.stringify แปลงข้อมูลเป็น JSON แบบจัดรูปแบบ (indent = 2)
- จับข้อผิดพลาด (เช่น URL ไม่ถูกต้อง หรือเว็บไซต์เปลี่ยนโครงสร้าง) และแสดงข้อความผิดพลาดใน

คอนโซล

```

} else {
  const webData1 = loadDataFromFile("./data1.json");
  const webData2 = loadDataFromFile("./data2.json");

  let combinedWebData = [];
  if (webData1 && webData1.length > 0) {
    combinedWebData = [...webData1];
  }

  if (webData2 && webData2.length > 0) {
    combinedWebData = [...combinedWebData, ...webData2];
  }

  if (combinedWebData.length === 0) {
    responseMessage = "ไม่พบข้อมูลในทั้งสองแหล่งข้อมูล";
    agent.add(responseMessage);
    return;
  }
}

```

ภาพที่ 19 การจัดการคำตอบของอินเทอร์เน็ตดึงจากเว็บไซต์

จากภาพที่ 20 ทำหน้าที่จัดการข้อมูลที่เกี่ยวข้องกับการตอบคำถามจากผู้ใช้งาน โดยจะทำการค้นหาข้อมูลจากฐานข้อมูล (database) หรือแหล่งข้อมูลที่เก็บในไฟล์ JSON เพื่อสร้างข้อความตอบกลับให้เหมาะสมที่สุด

### 1. ตรวจสอบข้อมูลจากฐานข้อมูล (Database)

- เช็คว่า webResult.answer มีข้อมูลหรือไม่ ถ้ามี จะกรองข้อมูลที่สำคัญจาก webResult.answer และเก็บไว้ในตัวแปร filteredAnswer

- ตรวจสอบว่า filteredAnswer มีข้อมูลในฟิลด์ใดบ้าง เช่น:

fee: ถ้ามีข้อมูลเกี่ยวกับค่าธรรมเนียม จะตั้งค่า responseMessage เป็นค่าของ fee

contact: ถ้ามีข้อมูลการติดต่อ จะตั้งค่า responseMessage เป็นค่าของ contact

openingHours: ถ้ามีข้อมูลเวลาเปิดทำการ จะตั้งค่า responseMessage เป็นค่าของ openingHours

other: ถ้ามีข้อมูลอื่น จะใช้ค่าในฟิลด์นี้

- ถ้าไม่มีข้อมูลที่ครบถ้วนในฟิลด์เหล่านี้ จะตอบว่า "ข้อมูลไม่ครบถ้วน"

- ตั้งค่าตัวแปร placeId, webAnswerId, และ sourceType เพื่อระบุแหล่งที่มาของข้อมูล

### 2. กรณีไม่มีข้อมูลจากฐานข้อมูล

- โหลดข้อมูลจากไฟล์ JSON เรียกฟังก์ชัน loadDataFromFile() เพื่อโหลดข้อมูลจากไฟล์ data1.json และ data2.json รวมข้อมูลจากทั้งสองไฟล์เข้าด้วยกันในตัวแปร combinedWebData

- ตรวจสอบว่ามีข้อมูลใน combinedWebData หรือไม่ ถ้าไม่มีข้อมูลเลย จะตอบกลับว่า "ไม่พบข้อมูลในทั้งสองแหล่งข้อมูล"

### 3. ค้นหาสถานที่ที่เกี่ยวข้อง

- ใช้ฟังก์ชัน getCorrectLocation() เพื่อค้นหาสถานที่ที่ใกล้เคียงกับคำค้นหาของผู้ใช้ ถ้าไม่พบสถานที่ที่ตรงกับคำค้นหา จะตอบว่า "ไม่พบสถานที่ที่ใกล้เคียงกับคำที่พิมพ์"

- ถ้าพบสถานที่ ใช้ filter() กรองข้อมูลใน combinedWebData เพื่อเลือกเฉพาะสถานที่ที่มีชื่อที่เกี่ยวข้องกับสถานที่ที่แก้ไขมา (correctedLocation.สถานที่)

### 4. ค้นหาข้อมูลเพิ่มเติมด้วยคำสำคัญ

- ใช้ฟังก์ชัน extractKeywords() เพื่อแยกคำสำคัญ (keywords) จากคำถามของผู้ใช้ ถ้าไม่สามารถแยกคำสำคัญได้ จะตอบว่า "ไม่พบคำสำคัญสำหรับการค้นหาในเว็บไซต์"

- ใช้ฟังก์ชัน filterByKeyword() เพื่อกรองข้อมูลที่เกี่ยวข้องกับคำสำคัญ ตั้งค่าข้อความตอบกลับ (responseMessage) ด้วยผลลัพธ์ที่ได้จากการกรอง หรือข้อความ "ไม่พบข้อมูลที่ตรงกับคำสำคัญ" ถ้าไม่มีข้อมูลตรงกัน

#### 5. ผลลัพธ์และแหล่งที่มา

- ตั้งค่าตัวแปร sourceType เป็น "website" และ isFromWeb เป็น true เพื่อบอกว่าข้อมูลมาจาก

เว็บไซต์

```

const correctedLocation = getCorrectLocation(
  placeName,
  combinedWebData
);
if (!correctedLocation || !correctedLocation.สถานที่) {
  console.log("Corrected location not found:", correctedLocation);
  responseMessage = "ไม่พบสถานที่ที่ใกล้เคียงกับคำที่พิมพ์";
  agent.add(responseMessage);
  return;
}

console.log("Corrected Location:", correctedLocation);

const filteredData = combinedWebData.filter(
  (item) =>
    item.สถานที่ &&
    typeof item.สถานที่ === "string" &&
    item.สถานที่
      .toLowerCase()
      .includes(correctedLocation.สถานที่.toLowerCase())
);

if (filteredData.length === 0) {
  responseMessage = "ไม่พบข้อมูลสถานที่ที่ตรงกับคำถามในเว็บไซต์";
  agent.add(responseMessage);
  return;
}

const keywords = await extractKeywords(questionText, dbClient);
if (keywords.length === 0) {
  responseMessage = "ไม่พบคำสำคัญสำหรับการค้นหาในเว็บไซต์";
  agent.add(responseMessage);
  return;
}

answerText = filterByKeyword(
  filteredData,
  keywords,
  questionText,
  displayName
);
responseMessage = answerText || "ไม่พบข้อมูลที่ตรงกับคำสำคัญ";
sourceType = "website";
isFromWeb = true;
}
}

```

ภาพที่ 20 การจัดการคำตอบของอินเทอร์เฟซนี้ดึงจากเว็บไซต์ (ต่อ)

ในภาพที่ 21 ทำหน้าที่ตรวจสอบและกรองข้อมูลที่เกี่ยวข้องกับสถานที่จากแหล่งข้อมูลที่มี เพื่อสร้างคำตอบที่เหมาะสมต่อคำถามของผู้ใช้ โดยมีการทำงานดังนี้

#### 1. ค้นหาสถานที่ที่ต้องการ

- เรียกฟังก์ชัน `getCorrectLocation(placeName, combinedWebData)` ฟังก์ชันนี้พยายามค้นหาสถานที่ที่ใกล้เคียงที่สุดกับชื่อสถานที่ (`placeName`) ใน `combinedWebData` ซึ่งเป็นข้อมูลที่รวมมาจากหลายแหล่ง
- ตรวจสอบผลลัพธ์จาก `getCorrectLocation` ถ้าผลลัพธ์ไม่มีข้อมูล (`!correctedLocation`) หรือไม่ มีฟิลด์ `สถานที่` แสดงข้อความว่า "ไม่พบสถานที่ที่ใกล้เคียงกับคำที่พิมพ์" หยุดการทำงานโดยใช้ `return`

#### 2. กรองข้อมูลสถานที่ที่เกี่ยวข้อง

- ใช้ `filter()` กรอง `combinedWebData` เพื่อค้นหาข้อมูลชื่อสถานที่ (`item.สถานที่`) ตรงกับสถานที่ที่ได้รับการแก้ไข (`correctedLocation.สถานที่` มีเงื่อนไข คือ ฟิลด์ `item.สถานที่` ต้องมีค่าและเป็นประเภท `string` ชื่อสถานที่ (`item.สถานที่`) จะถูกแปลงเป็นตัวพิมพ์เล็ก (`toLowerCase()`) และตรวจสอบว่ามีค่าที่เหมือนกับ `correctedLocation.สถานที่` อยู่ในนั้น
- ตรวจสอบผลลัพธ์ ถ้า `filteredData` ไม่มีข้อมูล ตอบกลับด้วยข้อความว่า "ไม่พบข้อมูลสถานที่ที่ตรงคำถามในเว็บไซต์" หยุดการทำงานโดยใช้ `return`

#### 3. ดึงคำสำคัญจากคำถาม

- ใช้ฟังก์ชัน `extractKeywords(questionText, dbClient)` เพื่อแยกคำสำคัญ (`keywords`) จากคำถาม (`questionText`) คำสำคัญจะช่วยให้การกรองข้อมูลที่เกี่ยวข้อง
- ตรวจสอบผลลัพธ์ ถ้าคำสำคัญ (`keywords`) ไม่มีเลย ตอบกลับด้วยข้อความว่า "ไม่พบคำสำคัญสำหรับการค้นหาในเว็บไซต์" หยุดการทำงานโดยใช้ `return`

#### 4. กรองข้อมูลด้วยคำสำคัญ

- เรียกฟังก์ชัน `filterByKeyword(filteredData, keywords, questionText, displayName)` ฟังก์ชันนี้กรองข้อมูลใน `filteredData` โดยใช้คำสำคัญ (`keywords`) และข้อมูลคำถาม (`questionText`) เพื่อค้นหาคำตอบที่ตรงที่สุด
- ตรวจสอบผลลัพธ์ ถ้าผลลัพธ์จาก `filterByKeyword` ไม่มีค่า ตอบกลับด้วยข้อความว่า "ไม่พบข้อมูลที่ตรงกับคำสำคัญ"
- ถ้ามีค่า ตั้งค่า `responseMessage` เป็นคำตอบที่ได้ (`answerText`)
- 5. ตั้งค่าตัวแปรแหล่งที่มา
  - ระบุแหล่งที่มาของข้อมูล ตั้งค่า `sourceType` เป็น "website" และตั้งค่า `isFromWeb` เป็น `true`



### 3.2.12 การดึงแผนที่เส้นทางการเดินทาง

ออกแบบมาเพื่อค้นหาตำแหน่งสถานที่ (Location) ตามคำถามของผู้ใช้ จากนั้นส่งข้อมูลตำแหน่งกลับไปยังผู้ใช้งานแพลตฟอร์ม เช่น LINE Chatbot โดยใช้ข้อมูลจากฐานข้อมูลในระบบและไฟล์ JSON สำรอง

(1) การวิเคราะห์ข้อความ (Text Analysis): ตรวจสอบชื่อสถานที่ (Place Name) ที่ผู้ใช้ป้อน และจับคู่กับข้อมูลในฐานข้อมูลหรือไฟล์ JSON ใช้เทคนิคการจัดการข้อความ เช่น การลบตัวเลขนำหน้า (removeLeadingNumbers) เพื่อจัดรูปแบบข้อมูลให้สะอาด

(2) การค้นหาข้อมูล (Data Matching): ใช้การเปรียบเทียบข้อความ (String Matching) ระหว่างชื่อสถานที่ที่ผู้ใช้ระบุและข้อมูลในระบบ ใช้ Fuzzy Matching สำหรับกรณีที่ชื่อสถานที่ไม่ตรงกันเป๊ะ เพื่อค้นหาสถานที่ที่ใกล้เคียงที่สุด

(3) การรวมข้อมูล (Data Merging): รวมข้อมูลจากหลายแหล่ง เช่น ฐานข้อมูลภายในและไฟล์ JSON สำรอง เพื่อให้ได้ข้อมูลที่สมบูรณ์ที่สุด

(4) การส่งข้อความสถานที่ (Location Message): สร้างข้อความที่มีข้อมูลตำแหน่ง เช่น ชื่อสถานที่, ที่อยู่, ละติจูด, ลองจิจูด ส่งข้อความกลับไปยังผู้ใช้งานในรูปแบบที่รองรับโดยแพลตฟอร์ม (เช่น LINE)

ขั้นตอนการทำงาน

(1) รับข้อมูลคำถามและวิเคราะห์สถานที่ รับคำถามจากผู้ใช้งานพร้อมกับชื่อสถานที่ (placeName) ที่ระบุในคำถาม หากผู้ใช้ไม่ได้ระบุชื่อสถานที่อย่างชัดเจน ระบบจะพยายามจับคู่ชื่อสถานที่จากข้อความคำถาม

(2) ค้นหาข้อมูลสถานที่ในฐานข้อมูล ตรวจสอบฐานข้อมูลในระบบ (ข้อมูลปัจจุบันที่เก็บอยู่ในตัวแปร locations.locations.locations) ว่ามีข้อมูลเกี่ยวกับสถานที่หรือไม่ หากพบสถานที่ ดึงข้อมูล เช่น ชื่อ, ที่อยู่, ละติจูด, ลองจิจูด ลบตัวเลขนำหน้าชื่อสถานที่ (หากมี) เพื่อจัดรูปแบบข้อมูลให้สะอาด สร้างข้อความตำแหน่ง (Location Message) และส่งกลับให้ผู้ใช้งาน

(3) ใช้ไฟล์ JSON สำรอง หากไม่พบข้อมูลในฐานข้อมูล ระบบจะค้นหาข้อมูลในไฟล์ JSON (data1.json และ data2.json) รวมข้อมูลจากไฟล์ JSON ทั้งหมดเข้าด้วยกันเพื่อลดการค้นหาซ้ำซ้อน

(4) ค้นหาสถานที่ที่ใกล้เคียง (Fuzzy Matching) ใช้ Fuzzy Matching เพื่อค้นหาสถานที่ที่ใกล้เคียงกับชื่อที่ผู้ใช้ระบุ หากพบสถานที่ที่ใกล้เคียง ดึงข้อมูลสถานที่ เช่น ชื่อ, ที่อยู่, ละติจูด, ลองจิจูด จาก JSON สร้างข้อความตำแหน่งและส่งกลับให้ผู้ใช้งาน

(5) จัดการกรณีไม่พบข้อมูล หากไม่พบข้อมูลในทั้งฐานข้อมูลและ JSON แจ้งผู้ใช้งานว่าไม่พบสถานที่ระบบจะหยุดทำงานและไม่ส่งข้อความตำแหน่ง

(6) ส่งข้อความตำแหน่งกลับผู้ใช้งาน ข้อมูลตำแหน่งที่ได้จากฐานข้อมูลหรือ JSON จะถูกจัดรูปแบบให้เหมาะสมส่งข้อมูลกลับในรูปแบบข้อความที่ผู้ใช้เข้าใจได้ง่าย พร้อมกับตำแหน่งในรูปแบบแผนที่

```

async function sendLocationBasedOnQuestion(agent) {
  const userId = agent.originalRequest.payload.data.source.userId;
  const placeName = agent.parameters.Location?.[0];

  try {
    let locationMessage = null;

    if (placeName && Array.isArray(locations.locations.locations)) {
      locations.locations.locations.forEach((loc) => {
        if (placeName.toLowerCase().includes(loc.title.toLowerCase())) {
          locationMessage = {
            type: "location",
            title: removeLeadingNumbers(loc.title),
            address: loc.address,
            latitude: loc.latitude,
            longitude: loc.longitude,
          };
        }
      });
    }

    if (locationMessage) {
      console.log("Found location in current database:", locationMessage);
      await client.pushMessage(userId, locationMessage);
      agent.add(`ได้เลยค่ะ นี่คือเส้นทางไป ${locationMessage.title} ค่ะ`);
      return;
    }
  }

  console.log(
    "Location not found in current database, searching in JSON files..."
  );

  const webData1 = await loadDataFromFile("./data1.json");
  const webData2 = await loadDataFromFile("./data2.json");

  let combinedWebData = [...webData1, ...webData2];

  if (combinedWebData.length === 0) {
    agent.add("ไม่พบข้อมูลในไฟล์แหล่งข้อมูลเพิ่มเติม");
    return;
  }
}

```

ภาพที่ 21 การดึงแผนที่เส้นทางการเดินทาง

```

const correctedLocation = getCorrectLocation(placeName, combinedWebData);
if (!correctedLocation || !correctedLocation.สถานที่) {
  console.log(
    "Corrected location not found in JSON data:",
    correctedLocation
  );
  agent.add("ไม่พบสถานที่ที่ใกล้เคียงกับคำที่พิมพ์");
  return;
}

console.log("Corrected Location from JSON:", correctedLocation);

locationMessage = {
  type: "location",
  title: removeLeadingNumbers(correctedLocation.สถานที่),
  address: correctedLocation.ที่อยู่ || "ไม่มีข้อมูลที่อยู่",
  latitude: correctedLocation.latitude || 0,
  longitude: correctedLocation.longitude || 0,
};

console.log("Sending location message from JSON data:", locationMessage);

await client.pushMessage(userId, locationMessage);
agent.add(`นี่คือเส้นทางไป ${locationMessage.title} จากข้อมูลเพิ่มเติมค่ะ`);
} catch (error) {
  console.error("Error processing location request:", error);
  agent.add("ขออภัย, ไม่สามารถส่งข้อมูลตำแหน่งได้ในขณะนี้.");
}
}

```

ภาพที่ 22 การดึงแผนที่เส้นทางการเดินทาง (ต่อ)

ในภาพที่ 22 และ 23 อธิบายได้ว่า ฟังก์ชัน sendLocationBasedOnQuestion ซึ่งมีวัตถุประสงค์ในการค้นหาและส่งข้อมูลตำแหน่ง (location) ให้กับผู้ใช้งาน

- เริ่มต้นฟังก์ชัน sendLocationBasedOnQuestion รับค่าจากผู้ใช้งาน
  - userId: ระบุ ID ของผู้ใช้งานจาก agent.originalRequest.payload.data.source.userId
  - placeName: ชื่อสถานที่ที่ผู้ใช้งานระบุจาก agent.parameters.Location?.[0]

- ค้นหาสถานที่ในฐานข้อมูลปัจจุบัน ตรวจสอบว่า placeName และข้อมูลใน locations.locations.locations มีอยู่หรือไม่ ใช้ forEach วนลูปข้อมูลในฐานข้อมูล เช็คว่า placeName มีส่วนที่ตรงกับชื่อสถานที่ (loc.title) หรือไม่ โดยเปรียบเทียบข้อความในรูปแบบตัวพิมพ์เล็ก (toLowerCase())

- ถ้าพบ สร้างข้อความตำแหน่ง (locationMessage) ที่มี ส่งข้อความตำแหน่งไปยังผู้ใช้งานด้วย client.pushMessage ส่งข้อความยืนยันผ่าน agent.add หยุดการทำงานของฟังก์ชัน (return)

- ถ้าไม่พบข้อมูลในฐานข้อมูล โหลดข้อมูลจากไฟล์ data1.json และ data2.json โดยใช้ loadDataFromFile รวมข้อมูลจากทั้งสองไฟล์ในตัวแปร combinedWebData
- ถ้าไม่มีข้อมูลในไฟล์ ตอบกลับว่า "ไม่พบข้อมูลในไฟล์แหล่งข้อมูลเพิ่มเติม" และหยุดการทำงาน (return)
- ใช้ฟังก์ชัน getCorrectLocation เพื่อค้นหาสถานที่ที่ใกล้เคียงที่สุดใน combinedWebData
- ถ้าพบสถานที่ที่ใกล้เคียง สร้างข้อความตำแหน่ง (locationMessage) โดยใช้ข้อมูลจาก correctedLocation ส่งข้อความตำแหน่งไปยังผู้ใช้งาน client.pushMessage ตอบกลับข้อความยืนยันด้วย agent.add

### 3.2.13 การจัดการพยากรณ์อากาศ

ออกแบบมาเพื่อให้ผู้ใช้สามารถขอพยากรณ์อากาศของเมืองที่ระบุ โดยการเชื่อมต่อกับ OpenWeatherMap API เพื่อดึงข้อมูลสภาพอากาศ และส่งข้อมูลที่จัดรูปแบบอย่างชัดเจนกลับไปยังผู้ใช้

(1) การระบุเมือง ผู้ใช้ระบุชื่อเมืองในคำถาม (Intent) หรือใช้ค่าเริ่มต้น (Default) เช่น "ขอนแก่น" หากไม่ได้ระบุเมือง

(2) การดึงข้อมูลพยากรณ์อากาศ ใช้ API จาก OpenWeatherMap เพื่อดึงข้อมูลพยากรณ์อากาศตามชื่อเมืองที่กำหนด

(3) การประมวลผลข้อมูล ข้อมูลที่ได้รับจะถูกแยกออกเป็นส่วนสำคัญ เช่น อุณหภูมิ ความชื้น ความเร็วลม

(4) การตอบกลับ สร้างข้อความที่มีข้อมูลสภาพอากาศอย่างชัดเจนและส่งกลับไปยังผู้ใช้  
ขั้นตอนการทำงาน

(1) การรับคำถามจากผู้ใช้งาน ระบบตรวจจับ Intent "พยากรณ์อากาศ" เมื่อผู้ใช้งานคำถาม เช่น "อากาศวันนี้ที่ขอนแก่นเป็นอย่างไร?" ดึงชื่อเมืองจากพารามิเตอร์ geo-city หรือใช้ค่าเริ่มต้น เช่น "ขอนแก่น" หากไม่มีการระบุ

(2) การเรียก OpenWeatherMap API สร้าง URL สำหรับคำขอ HTTP โดยใส่: ชื่อเมือง API Key การตั้งค่าหน่วยวัด

(3) การประมวลผลข้อมูลที่ได้รับ ดึงข้อมูลสำคัญ สภาพอากาศ (เช่น "มีเมฆบางส่วน") อุณหภูมิ (เช่น 28°C) ความชื้น (เช่น 65%) ความเร็วลม (เช่น 2 m/s) หาก API ไม่สามารถส่งข้อมูลได้ ระบบจะแจ้งข้อผิดพลาด เช่น "ไม่สามารถดึงข้อมูลพยากรณ์อากาศได้ในขณะนี้."

(4) การตอบกลับผู้ใช้ ส่งข้อความที่จัดรูปแบบแล้วกลับไปยังผู้ใช้ผ่านแพลตฟอร์ม เช่น LINE หากเกิดข้อผิดพลาด ระบบจะแจ้งข้อความอธิบายสถานการณ์ เช่น "เกิดข้อผิดพลาดในการเชื่อมต่อกับบริการพยากรณ์อากาศ."

```

const getWeatherForecast = async (city) => {
  const API_KEY = "673273db4167b63e98b0cc10eb00af13";
  const url = `https://api.openweathermap.org/data/2.5/weather?q=${city}&units=metric&appid=${API_KEY}`;

  try {
    const response = await axios.get(url);

    if (response.data) {
      const weatherData = response.data;
      const description = weatherData.weather[0].description;
      const temp = weatherData.main.temp;
      const feelsLike = weatherData.main.feels_like;
      const humidity = weatherData.main.humidity;
      const windSpeed = weatherData.wind.speed;

      return (
        `
        📍 พยากรณ์อากาศใน ${city}

        🌤️ สภาพอากาศ: ${
          description.charAt(0).toUpperCase() + description.slice(1)
        }

        🌡️ อุณหภูมิ: ${temp}°C (รู้สึกเหมือน ${feelsLike}°C)

        💧 ความชื้น: ${humidity}%

        🌬️ ความเร็วลม: ${windSpeed} m/s`
      );
    } else {
      return "ไม่สามารถดึงข้อมูลพยากรณ์อากาศได้ในขณะนี้.";
    }
  } catch (error) {
    console.error("Error fetching weather data:", error);
    return "เกิดข้อผิดพลาดในการเชื่อมต่อกับบริการพยากรณ์อากาศ.";
  }
};

```

### ภาพที่ 23 การจัดการพยากรณ์อากาศ

จากภาพที่ 24 อธิบายได้ว่า ฟังก์ชัน getWeatherForecast ที่ใช้สำหรับดึงข้อมูลพยากรณ์อากาศจาก OpenWeatherMap API ตามชื่อเมืองที่ระบุ และคืนผลเป็นข้อความที่มีรายละเอียดเกี่ยวกับสภาพอากาศ

API\_KEY เป็นคีย์สำหรับเข้าถึง API ของ OpenWeatherMap

url เป็น URL ที่ใช้เรียก API ของ OpenWeatherMap

– ใช้ axios.get(url) เพื่อเรียก API และดึงข้อมูลสภาพอากาศ ทำงานแบบ asynchronous โดยใช้ await เพื่อรอผลลัพธ์จาก API

– ถ้าการตอบกลับ (response.data) มีข้อมูล weatherData: เก็บข้อมูล JSON ที่ได้จาก API และดึงข้อมูลสำคัญออกมา

- ใช้ฟังก์ชัน `charAt(0).toUpperCase() + description.slice(1)` เพื่อให้ตัวอักษรแรกของ `description` เป็นตัวพิมพ์ใหญ่
- จัดการข้อผิดพลาด กรณีไม่มีข้อมูล (else) คืนข้อความว่า "ไม่สามารถดึงข้อมูลพยากรณ์อากาศได้ในขณะนี้."
- กรณีเกิดข้อผิดพลาด (catch) จะจับข้อผิดพลาดที่เกิดจากการเชื่อมต่อ API หรือข้อผิดพลาดอื่นๆ แสดงข้อความว่า "เกิดข้อผิดพลาดในการเชื่อมต่อกับบริการพยากรณ์อากาศ." และแสดงข้อผิดพลาดลงใน console เพื่อการดีบั๊ก

3.2.14 สร้างเซิร์ฟเวอร์โหนดเจส และใช้เอ็กซ์เพรสเจส เพื่อสร้างเว็บฮุก (webhook) เพื่อนำข้อมูลมาใช้ในการตอบคำถามของผู้ใช้ และเรียกเรียกใช้งาน API ภายนอก เพื่อให้สามารถรับและตอบกลับคำถามได้ การดึงข้อมูลจากฐานข้อมูลและทำการเชื่อมต่อ ใช้เว็บฮุกเพื่อเก็บบันทึกการสนทนา การโต้ตอบ และข้อมูลที่เกี่ยวข้องกับผู้ใช้

(1) การเชื่อมต่อเว็บฮุก เป็นการการตั้งค่าระบบให้สามารถส่งข้อมูลไปยังแอปพลิเคชันไลน์และไดอะล็อกโฟลว์ ช่วยให้สามารถสร้างบอทที่ตอบสนองต่อผู้ใช้ในไลน์ และสามารถเก็บบันทึกการสนทนายาระหว่างบอทและผู้ใช้ เพื่อปรับปรุงและพัฒนาในอนาคต โดยมีกระบวนการดังนี้

- ผู้ใช้งานส่งข้อความใน ไลน์: ผู้ใช้งาน ส่งข้อความผ่านแอป ไลน์ซึ่งข้อความนี้จะถูกส่งไปยัง LINE Server
- LINE Server จะส่ง Webhook Event (JSON payload) ไปยัง Webhook URL ที่ตั้งค่าไว้ใน LINE Developers Console
- Node.js Server รับ Webhook Event และส่งข้อมูลไปยัง ไดอะล็อกโฟลว์: โดยโหนดเจส Server จะมี Endpoint ที่รองรับ Webhook Event นี้ เมื่อได้รับ Webhook Event จะตรวจสอบประเภทของข้อความและประมวลผลข้อมูล จากนั้นจะส่งข้อความที่ได้รับจากผู้ใช้งานไปยัง ไดอะล็อกโฟลว์ เพื่อให้ตรวจสอบ Intent และตอบกลับ
- ไดอะล็อกโฟลว์ จะตรวจสอบอินเท้นท์ของข้อความที่ได้รับและประมวลผลเพื่อตอบกลับข้อความ เมื่อ ไดอะล็อกโฟลว์ ประมวลผลเสร็จ จะส่ง Fulfillment Response กลับมายัง Node.js Server
- Node.js Server จะได้รับข้อความตอบกลับจาก ไดอะล็อกโฟลว์ และส่งข้อความนี้กลับไปยัง LINE Server โดยใช้ Reply Token ที่ได้รับมาจาก Webhook Event LINE Server จะส่งข้อความตอบกลับไปยังผู้ใช้งานผ่านแอปไลน์

```

const handleWebhookRequest = async (req, res, dbClient) => {
  try {
    const lineId =
      req.body.originalDetectIntentRequest.payload?.data?.source?.userId;
    const questionText = req.body.queryResult.queryText;
    const location = Array.isArray(req.body.queryResult.parameters.location)
      ? req.body.queryResult.parameters.location[0]
      : req.body.queryResult.parameters.location;
    const displayName = req.body.queryResult.intent.displayName;
    const parameters = req.body.queryResult.parameters;

    console.log("Request Body:", req.body);
    console.log("Received Parameters:", parameters);
    console.log("line_id:", lineId);
    console.log("questionText:", questionText);
    console.log("intent displayName:", displayName);
    console.log("location:", location);

    if (!lineId || !questionText || !displayName) {
      console.error("Missing parameters:", {
        lineId,
        questionText,
        displayName,
      });
      return res.status(400).send("Missing required parameters.");
    }

    const agent = new WebhookClient({ request: req, response: res });

    const getDistrictData = () => {
      try {
        const data = fs.readFileSync("./data3.json", "utf8");
        const json = JSON.parse(data);

        if (!json.districts || !Array.isArray(json.districts)) {
          throw new Error("โครงสร้างข้อมูลผิดพลาดหรือไม่มีข้อมูลอำเภอ");
        }

        const districts = json.districts
          .filter(
            (d) =>
              d.name !== "พื้นที่รับผิดชอบจังหวัดขอนแก่นรวมทั้งหมด 26 อำเภอ"
          )
          .map((d, index) => `${index + 1}. ${d.name}`)
          .join("\n");

        return `**จังหวัดขอนแก่นมี ${
          json.districts.length - 1
        } อำเภอ:**\n\n${districts}`;
      } catch (error) {
        console.error("Error reading JSON file:", error.message);
        return "เกิดข้อผิดพลาดในการดึงข้อมูลอำเภอ กรุณาตรวจสอบไฟล์ข้อมูล";
      }
    };

    if (lineId) {
      console.log(`Saving line_id to database...`);
      await saveUser(lineId, dbClient);
    } else {
      console.log("Missing line_id, skipping user save.");
    }
  }
}

```

ภาพที่ 24 สร้างเซิร์ฟเวอร์โหนดเจเอส และใช้เอ็กซ์เพรสเจเอส เพื่อสร้างเว็บฮุก (webhook)

```

const intentMap = new Map();
intentMap.set("คำสรรณนิมิตการเข้า", (agent) =>
  handleIntent(agent, dbClient, questionText, location, displayName)
);

intentMap.set("ข้อมูลทั่วไปจังหวัดขอนแก่น", (agent) => {
  const districtList = getDistrictData();
  agent.add(districtList);
});

intentMap.set("ประเภทสถานที่ท่องเที่ยว", (agent) => {
  const questionText = req.body.queryResult.queryText;
  console.log("Question Text in Intent:", questionText);
  sendTouristFlexMessage(agent, questionText);
});

intentMap.set("ประเภทอำเภอ", (agent) => {
  const questionText = req.body.queryResult.queryText;
  console.log("Question Text in Intent:", questionText);
  sendDistrictFlexMessage(agent, questionText);
});

intentMap.set("พยากรณ์อากาศ", async (agent) => {
  const city = agent.parameters["geo-city"] || "ขอนแก่น";
  const weatherResponse = await getWeatherForecast(city);
  agent.add(weatherResponse);
});

intentMap.set("Default Welcome Intent", (agent) =>
  handleIntent(agent, dbClient, questionText, location, displayName)
);
intentMap.set("Default Fallback Intent", (agent) =>
  handleIntent(agent, dbClient, questionText, location, displayName)
);
intentMap.set("รายละเอียด", (agent) =>
  handleIntent(agent, dbClient, questionText, location, displayName)
);
intentMap.set("เวลาเปิดทำการ", (agent) =>
  handleIntent(agent, dbClient, questionText, location, displayName)
);

intentMap.set("เที่ยวขอนแก่น", (agent) => sendFlexMessage(agent, "kkc"));
intentMap.set("เลือกอำเภอ", (agent) => sendFlexMessage(agent, "district"));
intentMap.set("สถานที่ขอซื้อ", (agent) => {
  const questionText = req.body.queryResult.queryText;
  console.log("Question Text in Intent:", questionText);
  sendOneDayTripFlexMessage(agent, questionText);
});
intentMap.set("ร้านอาหารดังขอซื้อ", (agent) =>
  sendFlexMessage(agent, "restaurant")
);
intentMap.set("อาหารระดับมิชลินไกด์", (agent) =>
  sendFlexMessage(agent, "michelin")
);
intentMap.set("คาเฟ่ขอซื้อ", (agent) => sendFlexMessage(agent, "Cafe"));
intentMap.set("ร้านอาหารในเมืองขอนแก่น", (agent) =>
  sendFlexMessage(agent, "restaurantkkc")
);

intentMap.set("ร้านอาหารบุฟเฟ่", (agent) =>
  sendFlexMessage(agent, "buffet")
);
intentMap.set("เลือกประเภทสถานที่", (agent) =>
  sendFlexMessage(agent, "kkctype")
);
intentMap.set("เส้นทางไปยังสถานที่", sendLocationBasedOnQuestion);

agent.handleRequest(intentMap);

```

ภาพที่ 25 สร้างเซิร์ฟเวอร์โหนดเจเอส และใช้เอ็กซ์เพรสเจเอส เพื่อสร้างเว็บฮุก (webhook) (ต่อ)



จากภาพที่ 25 และ 26 อธิบายได้ว่าฟังก์ชัน `handleWebhookRequest` เป็นตัวจัดการคำขอ (Webhook Handler) ที่เชื่อมต่อกับ Dialogflow เพื่อประมวลผลคำถามและเจตนาของผู้ใช้

- ดึงข้อมูลที่เกี่ยวข้องจาก `req.body` ตรวจสอบว่าค่าที่จำเป็น (`lineId`, `questionText`, `displayName`) มีครบหรือไม่ ถ้าขาดค่าใดค่าหนึ่ง ให้ส่งสถานะ HTTP 400 พร้อมข้อความว่า "Missing required parameters."

- ฟังก์ชันเพิ่มเติม `getDistrictData` ฟังก์ชันนี้โหลดข้อมูลอำเภอจากไฟล์ `data3.json` ตรวจสอบว่าโครงสร้างข้อมูลถูกต้องหรือไม่ กรองข้อมูลอำเภอที่ไม่เกี่ยวข้อง คืนรายการอำเภอในรูปแบบข้อความที่เข้าใจง่าย

- และบันทึก `lineId` ลงฐานข้อมูล

- ใช้ `intentMap` เพื่อจับคู่ Intent (เจตนา) กับฟังก์ชันที่จัดการเจตนา นั้น หรือ Intent ที่เกี่ยวข้องกับข้อมูลอื่นๆ

- เรียก `handleRequest` เพื่อจับคู่ Intent ของผู้ใช้กับฟังก์ชันใน `intentMap`

3.2.15 การพัฒนา API สำหรับการเชื่อมต่อกับฐานข้อมูลและระบบการจัดการข้อมูลผ่าน Backend (Node.js และ Express.js) และการพัฒนาหน้าเว็บ (Frontend) เพื่อให้ผู้ใช้สามารถโต้ตอบกับข้อมูลผ่านระบบได้ ทั้งการเพิ่ม ลบ อัปเดต และการแสดงผลข้อมูล และเพื่อเก็บข้อมูลคำถาม-คำตอบที่ได้จากผู้ใช้และทำการเชื่อมต่อกับฐานข้อมูลกับเว็บฮุก เพื่อให้สามารถบันทึกและดึงข้อมูลคำถาม-คำตอบได้ดังนี้

(1) การรับคำขอจากผู้ใช้งาน (User Interaction) ผู้ใช้งานโต้ตอบกับหน้าเว็บ (Frontend) เช่น การเพิ่ม ลบ หรือแก้ไขข้อมูล โดยการกดปุ่มหรือกรอกข้อมูลที่หน้าเว็บ ซึ่งเป็นส่วนที่ผู้ใช้งานมองเห็นและโต้ตอบได้โดยตรง

(2) การส่งคำขอไปยังระบบหลังบ้าน (Backend Request): เมื่อผู้ใช้งานทำการโต้ตอบกับหน้าเว็บ ระบบจะส่งคำขอ (Request) ผ่าน API ไปยังระบบหลังบ้าน (Backend) โดยคำขออาจประกอบด้วยข้อมูลที่ผู้ใช้ป้อนเข้ามา หรือข้อมูลที่ต้องการประมวลผล

(3) การประมวลผลคำขอในระบบหลังบ้าน (Processing Request) ระบบหลังบ้านจะทำหน้าที่ประมวลผลคำขอที่ได้รับ เช่น การดึงข้อมูลจากฐานข้อมูล การเพิ่ม ลบ หรืออัปเดตข้อมูลในฐานข้อมูลตามคำขอที่ส่งมา

(4) การดึงข้อมูลจากฐานข้อมูล (Database Interaction) หากมีการร้องขอข้อมูลหรือการจัดการข้อมูล ระบบหลังบ้านจะทำการเชื่อมต่อกับฐานข้อมูลเพื่อดึงข้อมูลที่เกี่ยวข้อง เช่น การแสดงรายการข้อมูลที่บันทึกไว้ หรือทำการบันทึกข้อมูลใหม่ลงในฐานข้อมูล

(5) การส่งผลลัพธ์กลับไปยังระบบหน้าบ้าน (Response to Frontend) หลังจากคำขอถูกประมวลผลสำเร็จ ระบบหลังบ้านจะส่งผลลัพธ์กลับไปยังระบบหน้าบ้าน เช่น การยืนยันว่าข้อมูลถูกเพิ่มสำเร็จ หรือข้อมูลที่ดึงมาจากฐานข้อมูล

(6) การอัปเดตการแสดงผลบนหน้าเว็บ (UI Update) ระบบหน้าบ้านจะรับผลลัพธ์ที่ได้รับจากระบบหลังบ้านและอัปเดตการแสดงผลให้ผู้ใช้งานเห็นข้อมูลที่เป็นปัจจุบัน เช่น แสดงรายการข้อมูลใหม่ที่ถูกเพิ่ม หรือการแสดงผลข้อมูลที่ถูกอัปเดตหรือลบออกไป

โดยมีขั้นตอนการสร้างฐานข้อมูลดังนี้

- เชื่อมต่อ Docker เพื่อการสร้างบริการฐานข้อมูล PostgreSQL ภายในคอนเทนเนอร์ Docker โดยใช้ Docker Compose เพื่อลดความซับซ้อนในการตั้งค่าและการจัดการฐานข้อมูล
- สร้างตารางในฐานข้อมูลโดยการเขียนโค้ดสร้างตารางและ โดยใช้โปรแกรม DBeaver แล้วทำการสร้างฐานข้อมูล เลือกสร้างฐานข้อมูล และเลือก PostgreSQL กรอกข้อมูลที่ทำารตั้งค่าไว้ในไฟล์ docker-compose.local.yml
- ทำการต่อ API สร้างฟังก์ชัน เพิ่ม ลบ อัปเดต แต่ละหน้าให้ครบ โดยมีข้อมูลหน้าเว็บดังนี้หน้าเว็บสำหรับเก็บข้อมูลผู้ใช้งาน หน้าสำหรับเก็บคำถาม-คำตอบ หน้าสำหรับเก็บข้อมูลคำถาม อินเทนท์ คำตอบ
- เชื่อมต่อระหว่างหน้าบ้านและหลังบ้านให้สามารถใช้งานร่วมกันได้ โดยให้ผู้ใช้งานสามารถใช้งานหน้าเว็บได้เลย หน้าเว็บสามารถทำการเพิ่มข้อมูลคำตอบ คำถามถาม อินเทนท์จากไดอะล็อกโพล์ โดยมีขั้นตอนการทำดังนี้

```

const createTables = async () => {
  const createUsersTable = `
  CREATE TABLE IF NOT EXISTS users (
    id SERIAL PRIMARY KEY,
    line_id VARCHAR(255) UNIQUE NOT NULL,
    display_name VARCHAR(255),
    picture_url VARCHAR(255),
    status_message VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
  );
  `;

  const createPlacesTable = `
  CREATE TABLE IF NOT EXISTS places (
    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    admission_fee TEXT,
    address TEXT,
    contact_link TEXT,
    opening_hours VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
  );
  `;

  const createConversationsTable = `
  CREATE TABLE IF NOT EXISTS conversations (
    id SERIAL PRIMARY KEY,
    answer_text TEXT,
    question_text TEXT NOT NULL,
    user_id INT NOT NULL,
    web_answer_id INT,
    place_id INT,
    source_type VARCHAR(50) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (place_id) REFERENCES places(id),
    FOREIGN KEY (web_answer_id) REFERENCES web_answer(id),
    CONSTRAINT unique_user_place UNIQUE (user_id, place_id)
  );
  `;

  const createWebAnswerTable = `
  CREATE TABLE IF NOT EXISTS web_answer (
    id SERIAL PRIMARY KEY,
    place_name VARCHAR(255),
    answer_text TEXT NOT NULL UNIQUE,
    intent_type VARCHAR(50) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
  );
  `;

  try {
    await client.query(createUsersTable);
    await client.query(createWebAnswerTable);
    await client.query(createPlacesTable);
    await client.query(createConversationsTable);
    console.log("Tables are created or already exist.");
  } catch (err) {
    console.error("Error creating tables", err.stack);
  }
};

```

ภาพที่ 26 การสร้างตารางในฐานข้อมูล

จากภาพที่ 27 เป็นการสร้างตารางในฐานข้อมูล โดยมีทั้งหมด 4 ตารางได้แก่ users, places, conversations, web\_answer

- ใช้คำสั่ง CREATE TABLE IF NOT EXISTS ตรวจสอบว่าตารางมีอยู่แล้วหรือไม่ (ถ้ามี จะไม่สร้างใหม่)

เรียก client.query:

- ส่งคำสั่ง SQL ไปยังฐานข้อมูล PostgreSQL เพื่อสร้างแต่ละตาราง
- ใช้ try...catch จัดการข้อผิดพลาด

```
app.use((req, res, next) => {
  req.client = client;
  next();
});

app.post("/webhook", (req, res) => {
  handleWebhookRequest(req, res, client);
});

app.get("/api/data", (req, res) => {
  res.json({ message: "Hello from Node.js backend!" });
});
app.use("/places", placesRouter);
app.use("/users", usersRouter);
app.use("/conversations", conversationsRouter);

module.exports = app;
```

ภาพที่ 27 การกำหนดค่าและการตั้งค่าของเซิร์ฟเวอร์

จากภาพที่ 28 เป็นการกำหนดค่าและการตั้งค่าของเซิร์ฟเวอร์ ทำหน้าที่ตั้งค่าการทำงานสำหรับเซิร์ฟเวอร์ Node.js โดยใช้ Express.js ในการจัดการเส้นทาง (routes) และการทำงานต่างๆ ของแอปพลิเคชัน

- เพิ่ม Middleware ทุกคำขอจะมี req.client ซึ่งช่วยให้เข้าถึงฐานข้อมูล (หรือทรัพยากรอื่นๆ) ได้ง่ายขึ้น

- ตั้งค่า Webhook Endpoint สำหรับรับคำขอจากไดอะล็อกโฟลว์
- สร้าง API Test Endpoint เพื่อส่งข้อความ JSON เพื่อทดสอบว่าเซิร์ฟเวอร์ทำงานได้ปกติ
- โมดูลเส้นทาง (Routers)แยกการจัดการ API ออกเป็นโมดูลสำหรับสถานที่ (places), ผู้ใช้ (users), และบันทึกการสนทนา (conversations) เพื่อการพัฒนาที่มีประสิทธิภาพ

#### 4. การทดสอบเซทบอท

ผลการทดสอบ การออกแบบและการพัฒนาเซทบอท เพื่อการท่องเที่ยวในจังหวัดขอนแก่น เพื่อตรวจสอบประสิทธิภาพของระบบโดยรวม ตามตารางที่ 10-11

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_1	Flow Main Question	1. แอดไลน์ 2.กดที่เลือกเมนูที่ริชเมนู 3.กดเลือกเมนู "แนะนำสถานที่ท่องเที่ยวขอนแก่น"	1	Type "แนะนำสถานที่ท่องเที่ยวขอนแก่น"	สวัสดีค่ะ ไอรี่ยินดีให้บริการ สนใจท่องเที่ยวแบบไหนดีคะ แสดง Flex Message ประเภทสถานที่ท่องเที่ยว 1. เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว 2.เที่ยวแบบอำเภอ	สวัสดีค่ะ ไอรี่ยินดีให้บริการ สนใจท่องเที่ยวแบบไหนดีคะ แสดง Flex Message ประเภทสถานที่ท่องเที่ยว 1. เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว 2.เที่ยวแบบอำเภอ	
TC_chat_bot_2	Flow เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว	1.กดที่เลือกเมนู 2.กดเลือกเมนู "เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว"	2	Type "เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว"	แสดง Flex Message แยกตามสถานที่ท่องเที่ยว 1.ภูเขา 2.สวนสัตว์ 3.อุทยานแห่งชาติ 4.วัด 5.พิพิธภัณฑ์ 6.สวนน้ำ	แสดง Flex Message แยกตามสถานที่ท่องเที่ยว 1.ภูเขา 2.สวนสัตว์ 3.อุทยานแห่งชาติ 4.วัด 5.พิพิธภัณฑ์ 6.สวนน้ำ	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
					7.สวนสาธารณะ 8.แหล่งซื้อปิ้ง	7.สวนสาธารณะ 8.แหล่งซื้อปิ้ง	
TC_chat_bot_3	Flow เที่ยว แบบ อำเภอ	1.กดที่เลือกเมนู 2.กดเลือกเมนู "เที่ยวแบบ อำเภอ"	2	Type "เที่ยว แบบ อำเภอ"	แสดง Flex Message แยก ตามอำเภอ สถานที่ท่องเที่ยว 1.อำเภอเมือง 2.อำเภอน้ำพอง 3.อำเภอสีชมพู 4.อำเภอภูเวียง 5.อำเภอหนองเรือ 6.อำเภอชุมแพ 7.อำเภอเวียงเก่า 8.อำเภอบ้านฝาง 9.อำเภอเขาสวน กวาง 10.อำเภอเปือย น้อย 11.อำเภออุบล รัตน์ 12.อำเภอ กระนวน	แสดง Flex Message แยก ตามอำเภอ สถานที่ท่องเที่ยว 1.อำเภอเมือง 2.อำเภอน้ำพอง 3.อำเภอสีชมพู 4.อำเภอภูเวียง 5.อำเภอหนองเรือ 6.อำเภอชุมแพ 7.อำเภอเวียงเก่า 8.อำเภอบ้านฝาง 9.อำเภอเขาสวน กวาง 10.อำเภอเปือย น้อย 11.อำเภออุบล รัตน์ 12.อำเภอ กระนวน	
TC_chat_bot_4	Flow เที่ยว แยก ตาม ประเภท	1.กดที่เลือกเมนู 2.กดเลือกเมนู "ไปเที่ยวภูเขา ไหนดี"	2	Type "ไป เที่ยวภูเขา ไหนดี"	แสดง Flex Message เที่ยว ภูเขาไหนดี 1. จุดชมวิวหิน ช้างสี 2. ผาชมตะวัน	แสดง Flex Message เที่ยว ภูเขาไหนดี 1. จุดชมวิวหิน ช้างสี 2. ผาชมตะวัน	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
	สถานที่ท่องเที่ยว				3. ป่าสนดงลาน 4. น้ำตกบ่าหลวง 5. น้ำตกตาดฟ้า	3. ป่าสนดงลาน 4. น้ำตกบ่าหลวง 5. น้ำตกตาดฟ้า	
TC_chat_bot_5	Flow ที่เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว	กดเลือกเมนู "ไปเที่ยวสวนสัตว์ที่ไหนดี"	2	Type "ไปเที่ยวสวนสัตว์ที่ไหนดี"	แสดง Flex Message ไปเที่ยวสวนสัตว์ที่ไหนดี 1.สวนสัตว์ขอนแก่น 2.Khonkaen Exotic Pets	แสดง Flex Message ไปเที่ยวสวนสัตว์ที่ไหนดี 1.สวนสัตว์ขอนแก่น 2.Khonkaen Exotic Pets	
TC_chat_bot_6	Flow ที่เที่ยวแยกตามประเภทสถานที่ท่องเที่ยว	กดเลือกเมนู "ไปเที่ยวอุทยานแห่งชาติ ที่ไหนดี"	2	Type "ไปเที่ยวอุทยานแห่งชาติ ที่ไหนดี"	แสดง Flex Message เที่ยวอุทยานแห่งชาติ ที่ไหนดี 1.อุทยานแห่งชาติภูผาม่าน 2. อุทยานแห่งชาติภูเวียง 3.อุทยานแห่งชาติน้ำพอง	แสดง Flex Message เที่ยวอุทยานแห่งชาติ ที่ไหนดี 1.อุทยานแห่งชาติภูผาม่าน 2. อุทยานแห่งชาติภูเวียง 3.อุทยานแห่งชาติน้ำพอง	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_7	Flow เที่ยว แยก ตาม ประเภท สถานที่ ท่องเที่ยว	กดเลือกเมนู "ไปเที่ยวพิพิธภัณฑ์ที่ไหนดี"	2	Type "ไปเที่ยวพิพิธภัณฑ์ที่ไหนดี"	แสดง Flex Message เที่ยวพิพิธภัณฑ์ที่ไหนดี 1. พิพิธภัณฑ์ไดโนเสาร์ภูเวียง 2. ปราสาทเปือยน้อย	แสดง Flex Message เที่ยวพิพิธภัณฑ์ที่ไหนดี 1. พิพิธภัณฑ์ไดโนเสาร์ภูเวียง 2. ปราสาทเปือยน้อย	
TC_chat_bot_8	Flow เที่ยว แยก ตาม ประเภท สถานที่ ท่องเที่ยว	กดเลือกเมนู "ไปเที่ยวสวนน้ำที่ไหนดี"	2	Type "ไปเที่ยวสวนน้ำที่ไหนดี"	แสดง Flex Message เที่ยวสวนน้ำที่ไหนดี 1. สวนน้ำสวนสัตว์ขอนแก่น 2. สวนน้ำไดโนวอเตอร์ปาร์ค 3. บางแสน 2	แสดง Flex Message เที่ยวสวนน้ำที่ไหนดี 1. สวนน้ำสวนสัตว์ขอนแก่น 2. สวนน้ำไดโนวอเตอร์ปาร์ค 3. บางแสน 2	
TC_chat_bot_9	Flow เที่ยว แยก ตาม ประเภท สถานที่ ท่องเที่ยว	กดเลือกเมนู "ไปเที่ยวสวนสาธารณะที่ไหนดี"	2	Type "ไปเที่ยวสวนสาธารณะที่ไหนดี"	แสดง Flex Message ไปเที่ยวสวนสาธารณะที่ไหนดี 1. บึงแก่นนคร 2. บึงทุ่งสร้าง 3. บึงแก่นนคร 4. บึงศรีฐาน	แสดง Flex Message ไปเที่ยวสวนสาธารณะที่ไหนดี 1. บึงแก่นนคร 2. บึงทุ่งสร้าง 3. บึงแก่นนคร 4. บึงศรีฐาน	



ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_10	Flow เที่ยว แยก ตาม ประเภท สถานที่ ท่องเที่ยว	กดเลือกเมนู "ไปเที่ยววัดที่ไหนดี"	2	Type "ไปเที่ยววัดที่ไหนดี"	แสดง Flex Message ไปเที่ยววัดที่ไหนดี 1. พระมหาธาตุแก่นนคร (วัดหนองแวง) 2. ศาลหลักเมือง 3. พระธาตุขามแก่น 4. วัดทุ่งเศรษฐี 5. วัดถ้ำแสงธรรม 6. วัดถ้ำผาเกิ้ง 7. วัดป่าแสงอรุณ 8. วัดแก้วจักรพรรดิศิริสุทธารวาส 9. เทวาลัยศิวะมหาเทพ 10. วัดป่าธรรมอุทยาน	แสดง Flex Message ไปเที่ยววัดที่ไหนดี 1. พระมหาธาตุแก่นนคร (วัดหนองแวง) 2. ศาลหลักเมือง 3. พระธาตุขามแก่น 4. วัดทุ่งเศรษฐี 5. วัดถ้ำแสงธรรม 6. วัดถ้ำผาเกิ้ง 7. วัดป่าแสงอรุณ 8. วัดแก้วจักรพรรดิศิริสุทธารวาส 9. เทวาลัยศิวะมหาเทพ 10. วัดป่าธรรมอุทยาน	
TC_chat_bot_11	Flow เที่ยว แยก ตาม	กดเลือกเมนู "ไปช้อปปิ้งที่ไหนดี"	2	Type "ไปช้อปปิ้งที่ไหนดี"	แสดง Flex Message ไปช้อปปิ้งที่ไหนดี	แสดง Flex Message ไปช้อปปิ้งที่ไหนดี	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
	ประเภทสถานที่ท่องเที่ยว				1.ถนนคนเดินขอนแก่น 2.เปิดท้ายรถกัญญา 3.ตลาดต้นตาล	1.ถนนคนเดินขอนแก่น 2.เปิดท้ายรถกัญญา 3.ตลาดต้นตาล	
TC_chat_bot_12	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "ป่าสนดงลาน"	3	Type "ป่าสนดงลาน"	แสดง Flex Message ป่าสนดงลาน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message ป่าสนดงลาน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_13	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "น้ำตกบ่าหลวง"	3	Type "น้ำตกบ่าหลวง"	แสดง Flex Message น้ำตกบ่าหลวง 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message น้ำตกบ่าหลวง 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_14	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "จุดชมวิวหินช้างสี"	3	Type "จุดชมวิวหินช้างสี"	แสดง Flex Message จุดชมวิวหินช้างสี 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message จุดชมวิวหินช้างสี 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_15	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "พาชมตะวัน"	3	Type "พาชมตะวัน"	แสดง Flex Message พาชมตะวัน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message พาชมตะวัน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_16	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "น้ำตกตาดฟ้า"	3	Type "น้ำตกตาดฟ้า"	แสดง Flex Message น้ำตกตาดฟ้า 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message น้ำตกตาดฟ้า 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_17	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "อุทยานแห่งชาติภูผาม่าน"	3	Type "อุทยานแห่งชาติภูผาม่าน"	แสดง Flex Message อุทยานแห่งชาติภูผาม่าน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message อุทยานแห่งชาติภูผาม่าน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_18	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "อุทยานแห่งชาติภูเวียง"	3	Type "อุทยานแห่งชาติภูเวียง"	แสดง Flex Message อุทยานแห่งชาติภูเวียง 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message อุทยานแห่งชาติภูเวียง 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_19	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "อุทยานแห่งชาติน้ำพอง"	3	Type "อุทยานแห่งชาติน้ำพอง"	แสดง Flex Message อุทยานแห่งชาติน้ำพอง 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message อุทยานแห่งชาติน้ำพอง 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_20	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "สวนสัตว์ขอนแก่น"	3	Type "สวนสัตว์ขอนแก่น"	แสดง Flex Message สวนสัตว์ขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message สวนสัตว์ขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_21	Flow แยกตามชื่อสถานที่ท่องเที่ยว	1.กดที่เลือกเมนู Exotic Pets & The Fountain Show	3	Type "Khonkaen Exotic Pets & The Fountain Show"	แสดง Flex Message Khonkaen Exotic Pets & The Fountain Show 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message Khonkaen Exotic Pets & The Fountain Show 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_22	Flow แยกตามชื่อสถานที่	กดเลือกเมนู "พิพิธภัณฑ์ไดโนเสาร์ภูเวียง"	3	Type "พิพิธภัณฑ์ไดโนเสาร์ภูเวียง"	แสดง Flex Message พิพิธภัณฑ์ไดโนเสาร์ภูเวียง	แสดง Flex Message พิพิธภัณฑ์ไดโนเสาร์ภูเวียง	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
	ท่องเที่ยว				1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_23	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "ปราสาทเปือยน้อย"	3	Type "ปราสาทเปือยน้อย"	แสดง Flex Message ปราสาทเปือยน้อย 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message ปราสาทเปือยน้อย 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_24	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "สวนน้ำสวนสัตว์ขอนแก่น"	3	Type "สวนน้ำสวนสัตว์ขอนแก่น"	แสดง Flex Message สวนน้ำสวนสัตว์ขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message สวนน้ำสวนสัตว์ขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_25	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "สวนน้ำไดโนวอเตอร์ปาร์ค"	3	Type "สวนน้ำไดโนวอเตอร์ปาร์ค"	แสดง Flex Message สวนน้ำไดโนวอเตอร์ปาร์ค 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message สวนน้ำไดโนวอเตอร์ปาร์ค 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_26	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "บางแสน 2"	3	Type "บางแสน 2"	แสดง Flex Message บางแสน 2 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message บางแสน 2 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_27	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "ถนนคนเดินขอนแก่น"	3	Type "ถนนคนเดินขอนแก่น"	แสดง Flex Message ถนนคนเดินขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message ถนนคนเดินขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_28	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "เปิดท้ายรถจักรยาน"	3	Type "เปิดท้ายรถจักรยาน"	แสดง Flex Message เปิดท้ายรถจักรยาน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เปิดท้ายรถจักรยาน 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_29	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "ตลาดต้นตาล"	3	Type "ตลาดต้นตาล"	แสดง Flex Message ตลาดต้นตาล 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message ตลาดต้นตาล 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_30	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "เซ็นทรัลพลาซ่าขอนแก่น"	3	Type "เซ็นทรัลพลาซ่าขอนแก่น"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_31	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "พระมหาธาตุแก่นนคร (วัดหนองแวง)"	3	Type "พระมหาธาตุแก่นนคร (วัดหนองแวง)"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_32	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "ศาลหลักเมือง"	3	Type "ศาลหลักเมือง"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_33	Flow แยกตามชื่อสถานที่	กดเลือกเมนู "วัดถ้ำแสงธรรม"	3	Type "วัดถ้ำแสงธรรม"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
	ท่องเที่ยว				1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_34	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "วัดถ้ำผาเกิ้ง"	3	Type "วัดถ้ำผาเกิ้ง"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_35	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "วัดป่าแสงอรุณ"	3	Type "วัดป่าแสงอรุณ"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_36	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "วัดแก้วจักรพรรดิสิริสุทธาวาส"	3	Type "วัดแก้วจักรพรรดิสิริสุทธาวาส"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	



ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_37	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "เทวาลัยศิวิล มหาเทพ"	3	Type "เทวาลัยศิวิล มหาเทพ"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_38	Flow แยกตามชื่อสถานที่ท่องเที่ยว	กดเลือกเมนู "วัดป่าธรรมอุทยาน"	3	Type "วัดป่าธรรมอุทยาน"	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	แสดง Flex Message เซ็นทรัลพลาซ่าขอนแก่น 1.รายละเอียด 2.แผนที่ 3.ค่าเข้าชม 4.เวลาเปิด-ปิด	
TC_chat_bot_39	Flow แยกตามเวลาเปิด-ปิด	กดเลือกเมนู เวลาเปิด-ปิดสวนสัตว์ขอนแก่น	4	Type "สวนสัตว์ขอนแก่น เปิดกี่โมง"	เวลาเปิดทำการ : 08.00 - 16.30 น.	เวลาเปิดทำการ : 08.00 - 16.30 น.	
TC_chat_bot_40	Flow แยกแผนที่สวนสัตว์	กดเลือกเมนู แผนที่สวนสัตว์ขอนแก่น	4	Type "เส้นทางไปสวนสัตว์ขอนแก่น"	แสดง Location สวนสัตว์ขอนแก่น	แสดง Location สวนสัตว์ขอนแก่น	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_41	Flow แยกค่า เข้าชม	กดเลือกเมนูค่า เข้าชมสวนสัตว์ ขอนแก่น	4	Type " ค่าเข้า สวนสัตว์ ขอนแก่น "	ค่าเข้า : ผู้ใหญ่ 100 บาท เด็ก 20 บาท ผู้สูงอายุ 60 ปีขึ้นไปเข้าฟรี อัตราค่าเข้าชม น้ำ ผู้ใหญ่ 30 บาท เด็ก 20 บาท	ค่าเข้า : ผู้ใหญ่ 100 บาท เด็ก 20 บาท ผู้สูงอายุ 60 ปีขึ้นไปเข้าฟรี อัตราค่าเข้าชม น้ำ ผู้ใหญ่ 30 บาท เด็ก 20 บาท	
TC_chat_bot_42	Flow แยก ตาม รายละเอียด	กดเลือกเมนู รายละเอียดสวน สัตว์ขอนแก่น	4	Type " รายละเอียดสวน สัตว์ ขอนแก่น "	ที่เที่ยวขอนแก่น เด็กๆ ชอบ ผู้ใหญ่ ก็เที่ยวได้ สวน สัตว์ขอนแก่น ภายในมีสัตว์ให้ ชมมากมาย เช่น นกฟลามิงโก เสือ กวาง และยีราฟ เป็นต้น ไฮไลท์คือ สะพานเดินชม สัตว์ นักท่องเที่ยว สามารถเดินชม สัตว์ได้จากมุมสูง และสามารถ ใกล้ชิดกับเหล่า ยีราฟได้ ครอบครัวไหนมา เที่ยวขอนแก่นมี เด็กมาด้วยแนะนำ เลย	ที่เที่ยวขอนแก่น เด็กๆ ชอบ ผู้ใหญ่ ก็เที่ยวได้ สวน สัตว์ขอนแก่น ภายในมีสัตว์ให้ ชมมากมาย เช่น นกฟลามิงโก เสือ กวาง และยีราฟ เป็นต้น ไฮไลท์คือ สะพานเดินชม สัตว์ นักท่องเที่ยว สามารถเดินชม สัตว์ได้จากมุมสูง และสามารถ ใกล้ชิดกับเหล่า ยีราฟได้ ครอบครัวไหนมา เที่ยวขอนแก่นมี เด็กมาด้วยแนะนำ เลย	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat_bot_43	Flow ค่าเข้าชม	สอบถาม “ค่าเข้าชมสวนสัตว์ขอนแก่นกี่บาทคะ”	1	Type "ค่าเข้าชมสวนสัตว์ขอนแก่น"	ค่าเข้า : ผู้ใหญ่ 100 บาท เด็ก 20 บาท ผู้สูงอายุ 60 ปีขึ้นไปเข้าฟรี อัตราค่าเข้าชม น้ำ ผู้ใหญ่ 30 บาท เด็ก 20 บาท	ค่าเข้า : ผู้ใหญ่ 100 บาท เด็ก 20 บาท ผู้สูงอายุ 60 ปีขึ้นไปเข้าฟรี อัตราค่าเข้าชม น้ำ ผู้ใหญ่ 30 บาท เด็ก 20 บาท	
TC_chat_bot_44	Flow รายละเอียด	สอบถาม” รายละเอียดสวนสัตว์ขอนแก่น”	1	Type "รายละเอียดสวนสัตว์ขอนแก่น"	ที่เที่ยวขอนแก่น เด็กๆ ชอบ ผู้ใหญ่ก็เที่ยวได้ สวนสัตว์ขอนแก่น ภายในมีสัตว์ให้ชมมากมาย เช่น นกฟลามิงโก เสือ กวาง และยีราฟ เป็นต้น ไฮไลท์คือสะพานเดินชม สัตว์ นักท่องเที่ยวสามารถเดินชม สัตว์ได้จากมุมสูง และสามารถใกล้ชิดกับเหล่ายีราฟได้ ครอบครัวยุคใหม่เที่ยวขอนแก่นมีเด็กมาด้วยแนะนำเลย	ที่เที่ยวขอนแก่น เด็กๆ ชอบ ผู้ใหญ่ก็เที่ยวได้ สวนสัตว์ขอนแก่น ภายในมีสัตว์ให้ชมมากมาย เช่น นกฟลามิงโก เสือ กวาง และยีราฟ เป็นต้น ไฮไลท์คือสะพานเดินชม สัตว์ นักท่องเที่ยวสามารถเดินชม สัตว์ได้จากมุมสูง และสามารถใกล้ชิดกับเหล่ายีราฟได้ ครอบครัวยุคใหม่เที่ยวขอนแก่นมีเด็กมาด้วยแนะนำเลย	

ตารางที่ 10 ผลการทดสอบสถานที่ท่องเที่ยวขอนแก่น (ต่อ)

Test Case ID	Scenar io	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/ Fail
TC_cha t bot_45	Flow แผนที่	สอบถาม เส้นทางไปสวน สัตว์ขอนแก่น	1	Type " แผนที่ สวนสัตว์ ขอนแก่น "	แสดง Location สวนสัตว์ขอนแก่น	แสดง Location สวนสัตว์ขอนแก่น	
TC_cha t bot_46	Flow เวลา เปิด-ปิด	สอบถามเวลา เปิด-ปิดสวน สัตว์ขอนแก่น	1	Type " เวลาเปิด- ปิดสวน สัตว์ ขอนแก่น "	เวลาเปิดทำการ : 08.00 - 16.30 น.	เวลาเปิดทำการ : 08.00 - 16.30 น.	
TC_cha t bot_47	Flow Fallba ck	รีวิวนสวนสัตว์ โคราช	1	Type "รีวิวนสวน สัตว์ โคราช"	ฉันไม่เข้าใจคะ พุด ใหม่ได้ไหมคะ	ฉันไม่เข้าใจคะ พุด ใหม่ได้ไหมคะ	
TC_cha t bot_48	Flow Fallba ck	เปิดกี่โมง	1	Type "เปิดกี่ โมง"	อะไรนะคะ พุด ใหม่ได้ไหมคะ	อะไรนะคะ พุด ใหม่ได้ไหมคะ	
TC_cha t bot_49	Flow Fallba ck	ไปขอนแก่นยังไ งคะ	1	Type "ไป ขอนแก่น ยังไง"	อีกที่ไ้ไหมคะ	อีกที่ไ้ไหมคะ	

ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_01	Flow Main Question	1. แอดไลน์ 2.กดที่เลือกเมนูที่ริชเมนู 3.กดเลือกเมนู "แนะนำร้านอาหารมิชลินไกด์ขอนแก่น"	1	Type "แนะนำร้านอาหารมิชลินไกด์ขอนแก่น"	แสดง Flex Message ร้านอาหารมิชลินไกด์ขอนแก่น 1.อาหารอีสานหมวด 1 2. อาหารอีสานหมวด 2 3.อาหารไทย 4.อาหารไทย-จีน-อิตาลี 5.อาหารริมทาง 6.อาหารประเภทเส้น	แสดง Flex Message ร้านอาหารมิชลินไกด์ขอนแก่น 1.อาหารอีสานหมวด 1 2. อาหารอีสานหมวด 2 3.อาหารไทย 4.อาหารไทย-จีน-อิตาลี 5.อาหารริมทาง 6.อาหารประเภทเส้น	
TC_chat bot_02	Flow แยกตามประเภทร้านอาหาร	กดเลือกเมนู "อาหารอีสานหมวด 1"	2	Type "อาหารอีสานหมวด 1"	แสดง Flex Message ร้านอาหารอีสานหมวด 1 1. ตำกระเทียมสะเกต 2.มีกินฟาร์ม MEKIN FARM 3. ประไพ 4. สุขใจแลนด์	แสดง Flex Message ร้านอาหารอีสานหมวด 1 1. ตำกระเทียมสะเกต 2.มีกินฟาร์ม MEKIN FARM 3. ประไพ 4. สุขใจแลนด์	

ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_03	Flow แยกตามประเภทร้านอาหาร	กดเลือกเมนู "อาหารอีสานหมวด 2"	2	Type "อาหารอีสานหมวด 2"	แสดง Flex Message ร้านอาหารอีสาน หมวด 2 1. โซเจ้งโภชนา (บ้านไผ่) 2. ประสิทธิ์โภชนา 3. ไก่ย่างวรรณา 4. ไก่ย่างระเบียบ เขาสวนกวาง	แสดง Flex Message ร้านอาหารอีสานหมวด 2 1. โซเจ้งโภชนา (บ้านไผ่) 2. ประสิทธิ์โภชนา 3. ไก่ย่างวรรณา 4. ไก่ย่างระเบียบ เขาสวนกวาง	
TC_chat bot_04	Flow แยกตามประเภทร้านอาหาร	กดเลือกเมนู "อาหารไทย"	2	Type "อาหารไทย"	แสดง Flex Message ร้านอาหารอาหารไทย 1. ครั้วสุพรรณิการ์บาย คุณยายสมศรี 2. สีนานวล คาเฟ่ (กั๋งสตาล) 3. แก่น 4. คุณแจ่ง ก้วยเตี่ยวปากหม้อเข้าวัง	แสดง Flex Message ร้านอาหารอาหารไทย 1. ครั้วสุพรรณิการ์บาย คุณยายสมศรี 2. สีนานวล คาเฟ่ (กั๋งสตาล) 3. แก่น 4. คุณแจ่ง ก้วยเตี่ยวปากหม้อเข้าวัง	

ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_05	Flow แยกตามประเภทร้านอาหาร	กดเลือกเมนู "อาหารไทย-จีน-อิตาลี"	2	Type "อาหารไทย-จีน-อิตาลี"	แสดง Flex Message อาหารไทย-จีน-อิตาลี 1. มานะ โภชนาการ 2. บ้านเฮง 3. แวก้า อิตาเลียน บาย ฟาปิโอ	แสดง Flex Message อาหารไทย-จีน-อิตาลี 1. มานะ โภชนาการ 2. บ้านเฮง 3. แวก้า อิตาเลียน บาย ฟาปิโอ	
TC_chat bot_06	Flow แยกตามประเภทร้านอาหาร	กดเลือก "อาหารประเภทเส้น"	2	Type "อาหารประเภทเส้น"	แสดง Flex Message ร้านอาหารประเภทเส้น 1. ศรีเรือน ผัดไทย (สาขา ถนนรื่นจิตร) 2. บะหมี่กวางตั้ง 3. ก๋วยเตี๋ยวเนื้อฮေးจ่อย 4. เล้ง เย็นตาโฟ	แสดง Flex Message ร้านอาหารประเภทเส้น 1. ศรีเรือน ผัดไทย (สาขา ถนนรื่นจิตร) 2. บะหมี่กวางตั้ง 3. ก๋วยเตี๋ยวเนื้อฮေးจ่อย 4. เล้ง เย็นตาโฟ	

ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_07	Flow แยกตามประเภทร้านอาหาร	กดเลือกเมนู "อาหารริมทาง"	2	Type "อาหารริมทาง"	แสดง Flex Message อาหารริมทาง 1. โจ๊ก ก๋วยจั๊บน้ำร้อน ต้มเส้น บัตร์คิว 2. เกาเหลา เซี่ยงจี้- เลือดหมู เจ้าเก่า 3. ขอนแก่น คอกหมูย่าง 4. ข้าวต้มซัง 24 น. 5. เฟอทำบ่อ	แสดง Flex Message อาหารริมทาง 1. โจ๊ก ก๋วยจั๊บน้ำร้อน ต้มเส้น บัตร์คิว 2. เกาเหลา เซี่ยงจี้- เลือดหมู เจ้าเก่า 3. ขอนแก่น คอกหมู ย่าง 4. ข้าวต้มซัง 24 น. 5. เฟอทำบ่อ	
TC_chat bot_08	Flow แยกตามเวลาเปิด-ปิด	กดเลือกเมนู "เวลาเปิด-ปิดสี นานวล คาเฟ่ (กังสดาล)"	3	Type "สี นานวล คาเฟ่ (กังสดาล) เปิดกี่โมง"	เปิดทุกวัน 10.00-22.00 น.	เปิดทุกวัน 10.00-22.00 น.	
TC_chat bot_09	Flow แยกตามแผนที่	กดเลือกเมนู "แผนที่ สี นานวล คาเฟ่ (กังสดาล)"	3	Type "เส้นทางไปสี นานวล คาเฟ่ (กังสดาล)"	แสดง Location ร้าน	แสดง Location ร้าน	



ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_10	Flow แยกค่าเข้าชม	กดเลือกเมนู "ช่องทางการติดต่อ สีนานวล คาเฟ่ (กังสดาล)"	3	Type "ช่องทางการติดต่อร้าน สีนานวล คาเฟ่ (กังสดาล)"	เพจ See-na-nuan cafe' I สีนานวล คาเฟ่	เพจ See-na-nuan cafe' I สีนานวล คาเฟ่	
TC_chat bot_11	Flow แยกตามรายละเอียดรายละเอียด	สอบถามรายละเอียด สีนานวล คาเฟ่ (กังสดาล)	1	Type "รายละเอียด สีนานวล คาเฟ่ (กังสดาล)"	ร้านสีนานวล คาเฟ่ "สุดยอด ร้านอาหารไทยโบราณ ขอนแก่น" ร้านค่อนช้างใหญ่ จอดรถสบาย มีหลากหลายโซนให้เลือกนั่ง ทั้งชั้นบนชั้นล่าง outdoor ดนตรีสด ร้านตกแต่งสวยงาม	ร้านสีนานวล คาเฟ่ "สุดยอด ร้านอาหารไทยโบราณ ขอนแก่น" ร้านค่อนช้างใหญ่ จอดรถสบาย มีหลากหลายโซนให้เลือกนั่ง ทั้งชั้นบนชั้นล่าง outdoor ดนตรีสด ร้านตกแต่งสวยงาม	
TC_chat bot_12	Flow แยกตามเวลาเปิด-ปิด	สอบถาม "เวลาเปิด-ปิด สีนานวล คาเฟ่ (กังสดาล)"	1	Type "สีนานวล คาเฟ่ (กังสดาล) เปิดกี่โมง"	เปิดทุกวัน 10.00-22.00 น.	เปิดทุกวัน 10.00-22.00 น.	

ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_13	Flow แยกตามแผนที่	สอบถาม "แผนที่ที่ สีนานวล คาเฟ่ (กั๋งสตาล)"	1	Type "เส้นทางไป สีนานวล คาเฟ่ (กั๋งสตาล)"	แสดง Location ร้าน	แสดง Location ร้าน	
TC_chat bot_14	Flow แยกค่าเข้าชม	สอบถาม "ช่องทางการติดต่อ สีนานวล คาเฟ่ (กั๋งสตาล)"	1	Type "ช่องทางการติดต่อร้าน สีนานวล คาเฟ่ (กั๋งสตาล)"	เพจ See-nuan cafe'   สีนานวล คาเฟ่	เพจ See-nuan cafe'   สีนานวล คาเฟ่	
TC_chat bot_15	Flow แยกตามรายละเอียด	สอบถาม "รายละเอียด สีนานวล คาเฟ่ (กั๋งสตาล)"	1	Type "รายละเอียด สีนานวล คาเฟ่ (กั๋งสตาล)"	ร้าน สีนานวล คาเฟ่ "สุดยอด ร้านอาหารไทยโบราณ ขอนแก่น" ร้านค่อนช้างใหญ่ จอดรถสบาย มีหลากหลายโซนให้เลือกนั่ง ทั้งชั้นบนชั้นล่าง outdoor ดนตรีสด ร้านตกแต่งสวยงาม	ร้าน สีนานวล คาเฟ่ "สุดยอด ร้านอาหารไทยโบราณ ขอนแก่น" ร้านค่อนช้างใหญ่ จอดรถสบาย มีหลากหลายโซนให้เลือกนั่ง ทั้งชั้นบนชั้นล่าง outdoor ดนตรีสด ร้านตกแต่งสวยงาม	

ตารางที่ 11 ผลการทดสอบร้านอาหารมิชลินไกด์ขอนแก่น (ต่อ)

Test Case ID	Scenario	Test Action & Test Case	Step	Input	Expected Result	Actual Result	Pass/Fail
TC_chat bot_16	Flow Fallback	สอบถาม คาเฟ่ 11 AM อยู่ที่ ไหน	1	Type "11 AM อยู่ที่ ไหน"	ฉันฟังไม่ทันค่ะ	ฉันฟังไม่ทันค่ะ	
TC_chat bot_17	Flow Fallback	สอบถาม ร้านเบเกอรี่	1	Type "ร้านเบเกอรี่"	ฉันไม่เข้าใจค่ะ	ฉันไม่เข้าใจค่ะ	
TC_chat bot_18	Flow Fallback	สอบถามรีวิว อาหารเหนือ	1	Type "รีวิว อาหารเหนือ"	ช่วยพูดอีกทีได้ไหมคะ	ช่วยพูดอีกทีได้ไหมคะ	

## 5. การประเมินความพึงพอใจของผู้ใช้งาน

การประเมินความพึงพอใจของผู้ใช้งานการออกแบบและการพัฒนาแชทบอท เพื่อการท่องเที่ยวในจังหวัดขอนแก่นมีรูปแบบการประเมินดังนี้

### 5.1 กลุ่มเป้าหมาย

กลุ่มเป้าหมายที่ใช้ในการประเมินครั้งนี้เป็นบุคคลที่สนใจเกี่ยวกับการท่องเที่ยวจังหวัดขอนแก่น และบุคคลที่ต้องการทราบรายละเอียดข้อมูลการท่องเที่ยวจังหวัดขอนแก่นในการใช้งานการออกแบบและการพัฒนาแชทบอท เพื่อการท่องเที่ยวในจังหวัดขอนแก่น

### 5.2 แบบประเมิน

รายละเอียดแบบการประเมินมีหลักๆ 3 หัวข้อ คือ ข้อมูลทั่วไป แบบสอบถามความพึงพอใจผู้ใช้งาน และข้อเสนอแนะ

#### 5.2.1 ข้อมูลทั่วไปประกอบไปด้วย

(1) เพศ ได้แก่ ชาย หญิง ไม่ระบุ

(2) อายุ

#### 5.2.2 แบบสอบถามความพึงพอใจผู้ใช้งานประกอบไปด้วย

(1) ด้านการใช้งาน ได้แก่ แชทบอทสามารถโต้ตอบได้อย่างรวดเร็ว แชทบอทสามารถให้ข้อมูลได้ครบถ้วน แชทบอทสามารถใช้งานได้สะดวกและง่ายต่อการใช้งาน ความพึงพอใจกับการใช้งานแชทบอทนี้ ข้อมูลที่แชทบอทให้มา มีประโยชน์ต่อการวางแผนการท่องเที่ยว แชทบอทมีความสามารถในการตอบคำถามที่หลากหลาย

(2) ด้านการความถูกต้องของเนื้อหา ได้แก่ แชทบอทแสดงเนื้อหาถูกต้อง ทันสมัย และมีความน่าเชื่อถือ ตัวอักษร การเว้นวรรค ความชัดเจนของรูปภาพ และข้อความที่แสดง แชทบอทสามารถโต้ตอบกับผู้ใช้งานได้อย่างเหมาะสม แชทบอทสามารถตอบคำถามได้ครอบคลุมตามที่สนใจเกี่ยวกับการท่องเที่ยวในจังหวัดขอนแก่น ข้อมูลที่ให้นั้นมีความหลากหลายและน่าสนใจ ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวแต่ละแห่งมีความละเอียดเพียงพอ รูปภาพและสื่อต่างๆ ที่แชทบอทนำเสนอมีความน่าสนใจและเป็นประโยชน์

(3) ภาพรวมของระบบ ได้แก่ แชทบอทตรงต่อความต้องการของผู้ใช้งาน แชทบอทช่วยอำนวยความสะดวกให้แก่ผู้ใช้งาน แชทบอทสามารถใช้งานได้จริง ความพึงพอใจกับการใช้งานแชทบอทโดยรวม แชทบอทนี้มีประโยชน์ต่อการท่องเที่ยวในจังหวัดขอนแก่น

#### 5.2.3 ข้อเสนอแนะ ให้ผู้ใช้งานกรอกรายละเอียดข้อเสนอแนะในการปรับปรุงเพิ่มเติม

ในการประเมินแชทบอทนั้นผู้พัฒนาได้ทำการสร้างแบบสอบถามโดยผ่านการใช้งาน Google Forms เพื่อความสะดวกในการประเมินและง่ายต่อการสรุปผลการประเมิน

### 5.3 สรุปผลประเมินความพึงพอใจผู้ใช้งาน

สรุปผลประเมินความพึงพอใจผู้ใช้งานของการออกแบบและการพัฒนาแชทบอท เพื่อการท่องเที่ยวในจังหวัดขอนแก่น โดยมีผู้เข้าร่วมประเมินจำนวน คน เป็นผู้ชาย คน ผู้หญิง คน และไม่ระบุ คน โดยสามารถสรุปการประเมินผลได้ ดังต่อไปนี้

ตารางที่ 12 สรุปผลประเมินความพึงพอใจผู้ใช้งาน

รายการสอบถามการใช้งานระบบ	ระดับความพึงพอใจ					ผลคะแนน
	5	4	3	2	1	
1. ด้านการใช้งาน						
1.1 แอปพลิเคชันสามารถโต้ตอบได้อย่างรวดเร็ว						
1.2 แอปพลิเคชันสามารถให้ข้อมูลได้ครบถ้วน						
1.3 แอปพลิเคชันใช้งานได้ง่ายสะดวกและง่ายต่อการใช้งาน						
1.4 ความพึงพอใจกับการใช้งานแอปพลิเคชันนี้						
1.5 ข้อมูลที่แอปพลิเคชันให้มา มีประโยชน์ต่อการวางแผนการท่องเที่ยว						
1.6 แอปพลิเคชันมีความสามารถในการตอบคำถามที่หลากหลาย						
2. ด้านการความถูกต้องของเนื้อหา						
2.1 แอปพลิเคชันแสดงเนื้อหาถูกต้อง ทันสมัย และมีความน่าเชื่อถือ						
2.2 ตัวอักษร การเว้นวรรค ความชัดเจนของรูปภาพ และข้อความที่แสดง						
2.3 แอปพลิเคชันสามารถโต้ตอบกับผู้ใช้ได้อย่างเหมาะสม						
2.4 แอปพลิเคชันสามารถตอบคำถามได้ครอบคลุมตามที่สนใจเกี่ยวกับการท่องเที่ยวในจังหวัดขอนแก่น						
2.5 ข้อมูลที่แอปพลิเคชันให้มานั้นมีความหลากหลายและน่าสนใจ						
2.6 ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวแต่ละแห่งมีความละเอียดเพียงพอ						
2.7 รูปภาพและสื่อต่างๆ ที่แอปพลิเคชันนำเสนอมีความน่าสนใจและเป็นประโยชน์						
3. ภาพรวมของระบบ						
1.1 แอปพลิเคชันตรงต่อความต้องการของผู้ใช้งาน						
1.2 แอปพลิเคชันช่วยอำนวยความสะดวกให้แก่ผู้ใช้งาน						
1.3 แอปพลิเคชันใช้งานได้จริง						
1.4 ความพึงพอใจกับการใช้งานแอปพลิเคชันโดยรวม						
1.5 แอปพลิเคชันมีประโยชน์ต่อการท่องเที่ยวในจังหวัดขอนแก่น						

จากตารางที่ 12 สรุปได้ว่า (รอการดำเนินการในลำดับต่อไป)

## บทที่ 5

### บทสรุป

#### 1. สรุปผลการดำเนินโครงการ

งานวิจัยนี้ได้นำเสนอการออกแบบและพัฒนาแชทบอทเพื่อตอบคำถามและให้ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่นผ่านแอปพลิเคชันไลน์ โดยมีการรวบรวมข้อมูลจากแหล่งต่างๆ เพื่อนำมาพัฒนาแชทบอทที่ผสานการทำงานหลายเทคโนโลยี เช่น LINE, Node.js, Dialogflow, และ Web Scraping เพื่อให้ผู้ใช้สามารถรับข้อมูลการท่องเที่ยวได้อย่างรวดเร็วและแม่นยำ มีการวิเคราะห์คำถามผ่าน NLP และทำงานร่วมกับฐานข้อมูลรวมถึงการค้นหาข้อมูลจากเว็บภายนอกเพื่อให้สามารถตอบคำถามได้อย่าง และมีการสำรวจความพึงพอใจของผู้ใช้งานที่มีต่อแชทบอทผ่าน Google Forms เพื่อนำผลการสำรวจมาพัฒนาและปรับปรุงประสิทธิภาพแชทบอทให้มีคุณภาพต่อไป

จากการประเมินประสิทธิภาพของแชทบอท ผ่าน 3 ประเด็นหลัก คือ 1) ข้อมูลทั่วไปของผู้ประเมิน 2) ความพึงพอใจของผู้ใช้งาน ทั้งหมด 3 ด้าน และ 3) ข้อเสนอแนะอื่นๆ สรุปผลได้ดังนี้ (รอการดำเนินการในลำดับต่อไป)

#### 2. ข้อจำกัด

- 2.1 ปริมาณข้อมูลคำถามที่รวบรวมไว้ในปัจจุบันยังมีน้อย
- 2.2 ข้อมูลคำถามที่ใช้ในการฝึกอบรมโมเดลมีปริมาณจำกัด

#### 3. ปัญหาอุปสรรค

- 3.1 การพัฒนาแชทบอทให้สามารถเข้าใจและตอบคำถามของผู้ใช้ได้อย่างถูกต้อง อาจมีความซับซ้อนและต้องใช้เวลาในการเทรนโมเดล
- 3.2 ความหน่วงหรือการตอบกลับช้าของแชทบอทนั้น อาจเกิดการประมวลผลข้อมูลที่ใช้เวลานาน หรือการเชื่อมต่ออินเทอร์เน็ตที่ไม่เสถียร
- 3.3 การดึงข้อมูลจากเว็บไซต์ภายนอกมาใช้ในการตอบคำถาม เว็บไซต์บางแหล่งไม่สามารถดึงข้อมูลมาได้ เนื่องจากโครงสร้างของเว็บไซต์มีการเปลี่ยนแปลง หรือมีการป้องกันการดึงข้อมูลอัตโนมัติ
- 3.4 URL ของรูปภาพมีการเปลี่ยนแปลงบ่อย ทำให้แชทบอทไม่สามารถแสดงรูปภาพได้

#### 4. ข้อเสนอแนะ

- 4.1 เพิ่มจำนวนและความหลากหลายของคำถามที่ใช้ในการเทรนโมเดล เพื่อให้แชทบอทสามารถตอบคำถามได้หลากหลายและครอบคลุมมากขึ้น
- 4.2 พัฒนาความสามารถของแชทบอทในการให้คำแนะนำเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่น เช่น การจองที่พัก หรือการซื้อบัตรเข้าชมสถานที่ท่องเที่ยว
- 4.3 เพิ่มฟังก์ชันการทำงานอื่นๆ ให้กับแชทบอท เช่น การแปลภาษา

## เอกสารอ้างอิง

1. Alotaibi, R., Ali, A., Alharthi, H., & Almehamadi, R. (2020). AI Chatbot for Tourism Recommendations: A Case Study in the City of Jeddah, Saudi Arabia. **International Journal of Interactive Mobile Technologies**, 14(19), 18–30.
2. Casillo, M., Clarizia, F., D’Aniello, G., De Santo, M., Lombardi, M., & Santaniello, D. (2020). CHAT-Bot: A cultural heritage aware teller-bot for supporting touristic experiences. **Pattern Recognition Letters**, (131), 234–243.
3. Chaves, A. P., Egbert, J., Hocking, T., Doerry, E., & Gerosa, M. A. (2022). Chatbots Language Design: The Influence of Language Variation on User Experience with Tourist Assistant Chatbots. **ACM Transactions on Computer-Human Interaction**, 29(2), 1-38.
4. Benaddi L., Ouaddi C., Jakimi A., & Ouchao B. (2024). A Systematic Review of Chatbots: Classification, Development, and Their Impact on Tourism. **IEEE Access**. (12), 78799- 78810.
5. IBM. (2023). **5 types of chatbot and how to choose the right one for your business**. Retrieved August 18, 2023, from <https://www.ibm.com/blog/chatbot-types/>
6. Luchanka, A. (2022). **How do chatbots work? Often with a little help from AI**. Retrieved August 18, 2023, from <https://ventionteams.com/blog/how-do-chatbots-really-work>
7. Line developer. (2564). **LINE Bot Designer**. Retrieved August 18, 2023, from <https://developers.line.biz/media/messaging-api/using-bot-designer/bot-designer-user-manual.pdf>
8. LINE Developers TH. (n.d.). **Messaging API**. Retrieved August 18, 2023, from <https://linedevth.line.me/th/messaging-api>
9. Hardeniya, N., Perkins, J., Chopra, D., Joshi, N., & Mathur, I. (2016). **Natural Language Processing: Python and NLTK**. Birmingham, UK : Packt Publishing.
10. Dhanashree. (2023). **Web Scraping with Node.js: A Comprehensive Guide for 2024**. Retrieved March 20, 2023, from <https://nanonets.com/blog/web-scraping-with-node-js/>
11. Oracle. (n.d.). **What Is a Database?**. Retrieved December 9, 2023, from <https://www.oracle.com/in/database/what-is-database/?source=rh-rail>
12. Google cloud. (n.d.). **Dialogflow CX basics**. Retrieved December 13, 2023, from <https://cloud.google.com/dialogflow/cx/docs/basics#:~:text=Dialogflow%20translates%20end%20Duser%20text,a%20human%20call%20center%20agent.>



13. กังวาน อศวไกรวสิน. (2556). คู่มือสร้างเว็บไซต์ด้วย HTML5 CSS3&JavaScript ฉบับสมบูรณ์. กรุงเทพฯ: โปรวิชั่น.
14. บัญชา ปะสีละเตสัง. (2564). พัฒนา Web application ด้วย JavaScript และ Node.js. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.
15. จีราวุธ วารินทร์. (2562). พัฒนาเว็บแอปพลิเคชันด้วย Node.js Express+MongoDB. กรุงเทพฯ : รีไวย์.
16. จีราวุธ วารินทร์. (2563). พัฒนาเว็บแอปพลิเคชันด้วย Vue.js Vuex + Bootstrap. กรุงเทพฯ : รีไวย์.
17. Geeksforgeeks. (2024). Introduction to Tailwind CSS. Retrieved June 12, 2024, from <https://www.geeksforgeeks.org/introduction-to-tailwind-css/>

# ภาคผนวก

## ภาคผนวก ก

## แบบสอบถามความคิดเห็นของผู้ที่สนใจท่องเที่ยว

**คำชี้แจง** แบบสำรวจนี้มีวัตถุประสงค์เพื่อศึกษาความคิดเห็นของผู้ที่สนใจท่องเที่ยวในจังหวัดขอนแก่น ให้ทราบถึงความต้องการและคำถามที่พบบ่อย เพื่อนำมาพัฒนาแซทบอท ให้สามารถตอบคำถาม และให้ข้อมูลของสถานที่ได้ตามความต้องการ โดยมีรายละเอียดของคำถามดังต่อไปนี้

ตอนที่ 1 ข้อมูลทั่วไปของผู้ประเมิน จำนวน 2 ข้อ

ตอนที่ 2 ความคิดเห็นของผู้ที่สนใจท่องเที่ยวในจังหวัดขอนแก่น จำนวน 2 ข้อ

-----

## ส่วนที่ 1 ข้อมูลทั่วไปของตอบแบบสอบถาม

เพศ ☐ ชาย ☐ หญิง ☐ ไม่ระบุ

อายุ ☐ ต่ำกว่า 20 ปี ☐ 20-30 ปี ☐ 31-40 ปี ☐ 41-50 ปี ☐ 51-60 ปี ☐ 61 ปีขึ้นไป

## ส่วนที่ 2 ความคิดเห็นของผู้ที่สนใจท่องเที่ยวในจังหวัดขอนแก่น

1. มีความสนใจอยากมาท่องเที่ยวที่ไหนบ้างในจังหวัดขอนแก่น
2. อยากทราบอะไรบ้างเกี่ยวกับสถานที่ท่องเที่ยวในจังหวัดขอนแก่น

## ภาคผนวก ข

## แบบสอบถามความพึงพอใจการใช้งานงานเซทบอท

**คำชี้แจง**

1. แบบสำรวจนี้มีวัตถุประสงค์เพื่อศึกษาความพึงพอใจของผู้ใช้งานที่มีต่อเซทบอท เพื่อนำผลการสำรวจไปพัฒนาและปรับปรุงประสิทธิภาพเซทบอทให้มีคุณภาพต่อไป โดยมีรายละเอียดของคำถามดังต่อไปนี้

ตอนที่ 1 ข้อมูลทั่วไปของผู้ประเมิน จำนวน 2 ข้อ

ตอนที่ 2 ความพึงพอใจของผู้ตอบแบบสอบถาม จำนวน 18 ข้อ

ตอนที่ 3 ข้อเสนอแนะ จำนวน 1 ข้อ

2. โปรดตอบคำถามทุกข้อตามความเป็นจริง เพื่อให้การสำรวจในครั้งนี้เกิดประโยชน์สูงสุด

**ส่วนที่ 1 ข้อมูลทั่วไปของตอบแบบสอบถาม**

เพศ ☐ ชาย ☐ หญิง ☐ ไม่ระบุ

อายุ ☐ ต่ำกว่า 20 ปี ☐ 20-30 ปี ☐ 31-40 ปี ☐ 41-50 ปี ☐ 51-60 ปี ☐ 61 ปีขึ้นไป

**ส่วนที่ 2 ความพึงพอใจของผู้ตอบแบบสอบถาม**

**คำชี้แจง** กรุณาเลือกระดับความพึงพอใจของการใช้งานเซทบอท โดย 5 หมายถึง มีความพึงพอใจมากที่สุด 4 หมายถึง มีความพึงพอใจมาก 3 หมายถึง มีความพึงพอใจปานกลาง 2 หมายถึง มีความพึงพอใจน้อย และ 1 หมายถึง มีความพึงพอใจน้อยที่สุด

รายการสอบถามการใช้งานระบบ	ระดับความพึงพอใจ				
	5	4	3	2	1
1.1 เซทบอทสามารถโต้ตอบได้อย่างรวดเร็ว					
1.2 เซทบอทสามารถให้ข้อมูลได้ครบถ้วน					
1.3 เซทบอทสามารถใช้งานได้สะดวกและง่ายต่อการใช้งาน					
1.4 ความพึงพอใจกับการใช้งานเซทบอทนี้					
1.5 ข้อมูลที่เซทบอทให้มา มีประโยชน์ต่อการวางแผนการท่องเที่ยว					
1.6 เซทบอทมีความสามารถในการตอบคำถามที่หลากหลาย					

รายการสอบถามการใช้งานระบบ	ระดับความพึงพอใจ				
	5	4	3	2	1
1.7 แชทบอทแสดงเนื้อหาถูกต้อง ทันสมัย และมีความน่าเชื่อถือ					
1.8 ตัวอักษร การเว้นวรรค ความชัดเจนของรูปภาพ และข้อความที่แสดง					
1.9 แชทบอทสามารถโต้ตอบกับผู้ใช้งานได้อย่างเหมาะสม					
1.10 แชทบอทสามารถตอบคำถามได้ครอบคลุมตามที่สนใจเกี่ยวกับการท่องเที่ยวในจังหวัดขอนแก่น					
1.11 ข้อมูลที่แชทบอทให้มานั้นมีความหลากหลายและน่าสนใจ					
1.12 ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวแต่ละแห่งมีความละเอียดเพียงพอ					
1.13 รูปภาพและสื่อต่างๆ ที่แชทบอทนำเสนอมีความน่าสนใจและเป็นประโยชน์					
1.14 แชทบอทตรงต่อความต้องการของผู้ใช้งาน					
1.15 แชทบอทช่วยอำนวยความสะดวกให้แก่ผู้ใช้งาน					
1.16 แชทบอทสามารถใช้งานได้จริง					
1.17 ความพึงพอใจกับการใช้งานแชทบอทโดยรวม					
1.18 แชทบอทนี้มีประโยชน์ต่อการท่องเที่ยวในจังหวัดขอนแก่น					

### ส่วนที่ 3 ข้อเสนอแนะ

.....

.....

.....

.....

.....

ลงชื่อผู้ทำโครงการงาน .....

(นางสาวชลธิชา บุญสนัด)

ลงชื่อผู้ทำโครงการงาน .....

(นางสาวอารีรักษ์ กลางอินทร์เดช)

วันที่ .....

การตรวจสอบจากอาจารย์ที่ปรึกษาโครงการงาน

.....  
 .....  
 .....

(ลงชื่อ) .....

(ผศ. ดร.พฤษดี ศิริแสงตระกูล)

วันที่...../...../.....