เอกสารโครงงานฉบับสมบูรณ์

ไดโน : เว็บแอพลิเคชันศูนย์กลางของการท่องเที่ยวจังหวัดขอนแก่น

Dino: The central web application for tourism in Khon Kaen

โดย
663380203-9 นายคเณศ บัวสด
663380228-3 นายภูริ ชาริชา

อาจารย์ที่ปรึกษา
อ.ธนพล ตั้งชูพงศ์

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษา
วิชา CP354 771 โครงงานทางวิทยาการคอมพิวเตอร์ 1
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
วิทยาลัยการคอมพิวเตอร์
มหาวิทยาลัยขอนแก่น
ปีการศึกษา 2569

คเณศ บัวสด และ ภูริ ชาริชา. 2569. **ไดโน : เว็บแอพลิเคชันศูนย์กลางของการท่องเที่ยวจังหวัดขอนแก่น.** โครงงานคอมพิวเตอร์. ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น.
อาจารย์ที่ปรึกษา: อ.ธนพล ตั้งชูพงศ์

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาเว็บแอปพลิเคชัน “ไดโน” (Dino) ให้เป็นแพลตฟอร์มศูนย์กลางข้อมูลการท่องเที่ยวในจังหวัดขอนแก่น เพื่อแก้ไขปัญหาการเข้าถึงข้อมูลการท่องเที่ยวที่กระจัดกระจายและไม่ครบถ้วน โครงงานนี้ได้บูรณาการเทคโนโลยีปัญญาประดิษฐ์ (AI) ผ่านโมเดลภาษาขนาดใหญ่ (LLMs) ร่วมกับสถาปัตยกรรม Retrieval-Augmented Generation (RAG) ในการพัฒนาระบบหลัก 2 ส่วน ได้แก่ (1) ระบบสร้างแผนการเดินทางแบบปรับตัว (Adaptive Trip Planning) ที่ประเมินจากข้อจำกัดด้านงบประมาณ ระยะเวลา และความสนใจเฉพาะบุคคล และ (2) ระบบแชตบอต (Chatbot) สำหรับโต้ตอบและให้คำแนะนำด้านการท่องเที่ยวแบบเรียลไทม์ โดยใช้ฐานข้อมูลสถานที่และร้านอาหารจาก Google Places API ภายในจังหวัดขอนแก่น

ผลการดำเนินงานในระยะเริ่มต้นพบว่า การพัฒนาโครงสร้างพื้นฐานของระบบทั้งส่วนจัดการหลังบ้าน (Backend) และส่วนแสดงผลหน้าบ้าน (Frontend) มีความคืบหน้าประมาณร้อยละ 60 รวมถึงได้ทำการทดสอบการทำงานของโมเดลภาษาขนาดใหญ่ (Gemini Flash) และการดึงข้อมูลสถานที่จริงได้สำเร็จร้อยละ 100 อย่างไรก็ตาม ระบบยังอยู่ระหว่างการบูรณาการส่วนประกอบต่าง ๆ ให้สามารถทำงานร่วมกันได้อย่างสมบูรณ์ (End-to-End System) ทั้งนี้ เมื่อการพัฒนาและการทดสอบกับผู้ใช้งานจริงเสร็จสมบูรณ์ คาดว่าแอปพลิเคชันนี้จะช่วยลดระยะเวลาในการค้นหาข้อมูล เพิ่มประสิทธิภาพในการวางแผนการเดินทางที่ตอบโจทย์เฉพาะบุคคล ตลอดจนช่วยสนับสนุนผู้ประกอบการท้องถิ่นและส่งเสริมเศรษฐกิจการท่องเที่ยวของจังหวัดขอนแก่นอย่างยั่งยืน

คำสำคัญ: แอปพลิเคชันการท่องเที่ยว, ปัญญาประดิษฐ์, การดึงข้อมูลเสริม (RAG), โมเดลภาษาขนาดใหญ่, ไมโครเซอร์วิส, ขอนแก่น

Khanet Buasod and Phuri Charicha. 2026. **Dino: The central web application for tourism in Khon Kaen.** Computer Project, Bachelor of Science, Program in Computer Science, College of Computing, Khon Kaen University.
Project Advisor: Thanapon Tangchupong

ABSTRACT

This project aims to develop the web application ”Dino” as a centralized tourism information platform in Khon Kaen province to address the problem of fragmented and incomplete tourism data. The project integrates artificial intelligence (AI) technologies via Large Language Models (LLMs) combined with Retrieval-Augmented Generation (RAG) architecture to develop two main systems: (1) an Adaptive Trip Planning system that generates itineraries based on constraints regarding budget, time, and personal interests, and (2) a real-time Chatbot system for interactive communication and tourism recommendations. The system utilizes a database of attractions and restaurants retrieved from the Google Places API within Khon Kaen province.

Preliminary results indicate that the development of the system’s core infrastructure, including both the Backend and Frontend, is approximately 60% complete. Furthermore, the testing of the Large Language Model (Gemini Flash) and real-world location data retrieval has been 100% successful. However, the system is currently undergoing integration of its various components to achieve a fully functional End-to-End System. Once development and User Acceptance Testing (UAT) are fully completed, this application is expected to reduce information search time, enhance the efficiency of personalized trip planning, support local entrepreneurs, and promote the sustainable tourism economy of Khon Kaen province.

Keywords: Tourism Application, Artificial Intelligence, Retrieval-Augmented Generation (RAG), Large Language Models (LLMs), Microservices, Khon Kaen

กิตติกรรมประกาศ

ในการดำเนินโครงงานครั้งนี้ ผู้จัดทำโครงงานได้รับความอนุเคราะห์และความช่วยเหลือจากบุคคลหลายท่านด้วยกัน จึงขอขอบคุณไว้ ณ โอกาสนี้ ขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้และอบรมสั่งสอน ขอขอบพระคุณอาจารย์ธนพล ตั้งชูพงศ์ ที่เป็นที่ปรึกษาโครงงานและได้ให้คำชี้แนะแนวทางในการคิดวิเคราะห์และการพัฒนาระบบของโครงงานนี้ ขอขอบคุณคุณพ่อ คุณแม่ ที่เป็นกำลังใจในการเรียนและการทำโครงงานมาโดยตลอด ขอบคุณเพื่อน ๆ ที่ให้คำปรึกษาและช่วยเหลือในด้านการวิเคราะห์ออกแบบระบบ

ผู้จัดทำ
คเณศ บัวสด
ภูริ ชาริชา

สารบัญ

บทคัดย่อ **ก**

ABSTRACT **ข**

กิตติกรรมประกาศ **ค**

สารบัญ **ง**

สารบัญภาพ **ช**

สารบัญตาราง **ซ**

1 บทนำ **1**

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตและข้อจำกัดของระบบ 2

1.3.1 ขอบเขต 2

1.3.2 ข้อจำกัดของโครงงาน 3

1.4 ประโยชน์ที่คาดว่าจะได้รับ 5

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง **6**

2.1 ทฤษฎีที่เกี่ยวข้อง 6

2.1.1 แนวคิดการท่องเที่ยวอัจฉริยะ (Smart Tourism) 7

2.1.2 ปัญญาประดิษฐ์ (Artificial Intelligence) ในการท่องเที่ยว . 7

2.1.3 การจัดการจุดหมายปลายทางและการมีส่วนร่วมของนักท่องเที่ยว 7

2.1.4 ความยั่งยืน ความเสี่ยง และธรรมาภิบาลข้อมูล/อัลกอริทึม . 8

2.1.5 ทฤษฎีด้าน Chatbot 9

2.1.6 ระบบแนะนำการท่องเที่ยว (Recommendation Systems) 10

2.1.7 การสร้างแผนการเดินทาง (Trip Planning) 11

2.2 งานวิจัยที่เกี่ยวข้อง 15

2.2.1 การใช้ AI เพื่อพัฒนาระบบแนะนำการท่องเที่ยวและ Trip Planning 15

2.2.2 AI และ Chatbot เพื่อยกระดับประสบการณ์ผู้ใช้ 16

2.2.3 การจัดการข้อมูล การวิเคราะห์พฤติกรรม และการมีส่วนร่วมของนักท่องเที่ยว 16

2.2.4 การ ประเมิน ผล ด้วย โมเดล ภาษา ขนาด ใหญ่ ใน ฐานะ ผู้ประเมิน (LLM-as-a-Judge) 17
2.2.5 การ ประเมิน ผล และ วัด ประสิทธิภาพ การ วางแผน การ เดินทางด้วย LLM 18
2.2.6 สรุปผลการเปรียบเทียบงานวิจัยที่เกี่ยวข้อง 20
2.2.7 เทคโนโลยีและเครื่องมือที่เกี่ยวข้อง 21

3 วิธีการดำเนินโครงงาน 23
3.1 ขั้นตอนการดำเนินงาน 23
3.1.1 ศึกษาปัญหาและรวบรวมความต้องการ 23
3.1.2 วิเคราะห์และออกแบบระบบ 25
3.1.3 พัฒนาระบบ 26
3.1.4 ทดสอบและประเมินผล 30
3.1.5 ปรับปรุงและสรุปผล 31
3.2 เครื่องมือที่ใช้ในการดำเนินงาน 32
3.2.1 ฮาร์ดแวร์ 32
3.2.2 ซอฟต์แวร์ 32
3.3 แผนการดำเนินงาน 32

4 การวิเคราะห์ ออกแบบและพัฒนาระบบ 34
4.1 การวิเคราะห์ระบบ 34
4.1.1 การวิเคราะห์ปัญหาและการทำงานเดิม 34
4.1.2 ความต้องการของระบบ 35
4.1.3 Use Case Diagram และรายละเอียด 36
4.1.4 System Sequence Diagram 46
4.1.5 แผนภาพกิจกรรม (Activity Diagram) 48
4.2 การออกแบบระบบ 49
4.2.1 สถาปัตยกรรมของระบบ 49
4.2.2 Sequence Diagram 50
4.2.3 การออกแบบฐานข้อมูล 51
4.2.4 Mockup User Interface 56
4.3 การพัฒนาโปรแกรม 57
4.3.1 ภาษาและเครื่องมือที่ใช้ในการพัฒนา 57
4.3.2 โครงสร้างของโปรแกรม 57
4.3.3 ข้อกำหนดเฉพาะของแต่ละส่วน 57
4.3.4 การติดต่อกับฐานข้อมูลและ Services 57
4.4 การทดสอบระบบ 57
4.4.1 วิธีการทดสอบ 57
4.4.2 ผลการทดสอบ 58

5 สรุปผลการดำเนินงาน 59
5.1 สรุปผลการดำเนินโครงงาน 59
5.2 ข้อจำกัดของระบบ 60
5.2.1 ข้อจำกัดด้านการทำงาน 60

5.2.2 ข้อจำกัดด้านฮาร์ดแวร์และซอฟต์แวร์ 61
5.3 ปัญหาอุปสรรค และแนวทางแก้ไข 61
5.4 ข้อเสนอแนะ ในการพัฒนาต่อไป 62
5.4.1 การปรับปรุงฟังก์ชันการทำงาน 62
5.4.2 แนวทางการวิจัยต่อยอด 62

เอกสารอ้างอิง **63**

ภาคผนวก ก: ตัวอย่าง UI หรือ Prototype **65**

ประวัติผู้เขียน **67**

สารบัญภาพ

ภาพที่ 1 แผนภาพ Heatmap แสดงข้อมูลจากแบบสอบถาม (Ma et al., 2025) . . 8
ภาพที่ 2 แผนภาพ Model Flow (University College of Goizha et al., 2023) . 9
ภาพที่ 3 แผนภาพ Pipeline ของระบบ RAG ที่ได้รับการปรับปรุงตามข้อเสนอโดยกรอบเส้นประสีเขียวแสดง ขั้นตอนการเสริมข้อมูลที่ปรับเปลี่ยน พร้อมการเพิ่มตัวชี้วัดด้านความยั่งยืน (Banerjee et al., 2025) 10
ภาพที่ 4 แผนภาพสถาปัตยกรรมระบบ (Deshmukh et al., 2025) 12
ภาพที่ 5 แผนภาพการไหลของข้อมูลและกระบวนการประสานลำดับงาน (Barua & Kaiser, 2024) 14
ภาพที่ 6 แผนภาพสถาปัตยกรรมระบบ PTS (Personal Travel Solver Framework) (Shao et al., 2025) 15
ภาพที่ 7 แผนภาพ Heatmap แสดงข้อมูลเชิงพฤติกรรม (Ma et al., 2025) 17
ภาพที่ 8 สถาปัตยกรรมกระบวนการประเมินผลและการประมวลผลคำตอบของ LLM-as-a-Judge (Gu et al., 2025) 18
ภาพที่ 9 กรอบแนวคิดการประเมินการวางแผนการเดินทาง (Xie et al., 2024) . . . 19
ภาพที่ 10 เปรียบเทียบแนวโน้มประสิทธิภาพของ LLM เมื่อระดับความซับซ้อนและจำนวนเมืองในแผนการเดินทางเพิ่มขึ้น (Zheng et al., 2024) 20
ภาพที่ 11 แผนภาพสถาปัตยกรรมระบบของงานวิจัย 26
ภาพที่ 12 แผนภาพขั้นตอนการทำงานของระบบวางแผนการเดินทาง 27
ภาพที่ 13 แผนภาพขั้นตอนการทำงานของระบบ Chatbot และระบบแนะนำ 28
ภาพที่ 14 Tourist Use Case Diagram 36
ภาพที่ 15 Admin Use Case Diagram 42
ภาพที่ 16 System Sequence Diagram สำหรับผู้ใช้งานทั่วไป 47
ภาพที่ 17 System Sequence Diagram สำหรับผู้ดูแลระบบ 48
ภาพที่ 18 Activity Diagram ตรรกะการประมวลผลเพื่อสร้างแผนการเดินทาง 48
ภาพที่ 19 สถาปัตยกรรมของระบบ 49
ภาพที่ 20 Detailed Sequence Diagram สำหรับผู้ใช้งานทั่วไป 50
ภาพที่ 21 Detailed Sequence Diagram สำหรับผู้ดูแลระบบ 51
ภาพที่ 22 E-R Diagram แสดงการออกแบบฐานข้อมูลระบบ 52
ภาพที่ 23 Wireframe แสดงการใช้งานระบบ 56
ภาพที่ 24 หน้าจอเข้าสู่ระบบ (Login) 65
ภาพที่ 25 หน้าจอหลัก (Dashboard) 66

สารบัญตาราง

ตารางที่ 1 เปรียบเทียบงานวิจัยที่เกี่ยวข้อง 20
ตารางที่ 2 รายการซอฟต์แวร์ที่ใช้ในการพัฒนา 32
ตารางที่ 3 แผนการดำเนินงาน (Gantt Chart) 33
ตารางที่ 4 ความต้องการเชิงฟังก์ชัน (Functional Requirements) 35
ตารางที่ 5 ความต้องการที่ไม่ใช่เชิงฟังก์ชัน (Non-Functional Requirements) . . 36
ตารางที่ 6 Use Case UC-01: ถามตอบข้อมูลโดยใช้ Chatbot 37
ตารางที่ 7 Use Case UC-02: ค้นหาข้อมูลสถานที่ท่องเที่ยว กิจกรรม และอีเวนท์ . 38
ตารางที่ 8 Use Case UC-04: ดูรายละเอียดข้อมูลสถานที่ (Place Detail) 39
ตารางที่ 9 Use Case UC-05: ดูรายละเอียดข้อมูลอีเวนท์ (View Event Detail) . . 40
ตารางที่ 10 Use Case UC-06: สร้างแผนการเดินทาง (Trip Planning) 41
ตารางที่ 11 Use Case UC-07: การประเมินแผนการเดินทาง 42
ตารางที่ 12 Use Case UC-11: เข้าสู่ระบบ (Login / Authentication) 43
ตารางที่ 13 Use Case UC-06.1/6.2/6.3: จัดการข้อมูลสถานที่ (Create/Update/ Delete Place) 44
ตารางที่ 14 Use Case UC-09.1/9.2/9.3: จัดการข้อมูล Event (Create/Update/ Delete Event) 45
ตารางที่ 15 Use Case UC-10.1/10.2/10.3: จัดการฐานความรู้ (Create/Update/ Delete KnowledgeBase) 46
ตารางที่ 16 ตารางฐานข้อมูล Place (สถานที่ท่องเที่ยว) 52
ตารางที่ 17 ตารางฐานข้อมูล KnowledgeBase 53
ตารางที่ 18 ตารางฐานข้อมูล Event 53
ตารางที่ 19 ตารางฐานข้อมูล Itinerary 54
ตารางที่ 20 ตารางฐานข้อมูล EmbeddingDocument (Vector Store) 55
ตารางที่ 21 ตารางฐานข้อมูล User (Admin) 55
ตารางที่ 22 เกณฑ์การประเมินคุณภาพแผนการเดินทางด้วยเทคนิค LLM-as-a-Judge 58
ตารางที่ 23 ตารางแสดงผลการดำเนินโครงงาน 60
ตารางที่ 24 ตารางแสดงการสรุปปัญหา อุปสรรค และแนวทางแก้ไข 61
ตารางที่ 25 แผนการดำเนินงานในระยะต่อไป 62

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

จังหวัดขอนแก่นนับเป็นจังหวัดที่มีศักยภาพทางการท่องเที่ยวสูง ด้วยความหลากหลายของแหล่งท่องเที่ยว ทั้งทางธรรมชาติ วัฒนธรรม ประวัติศาสตร์ ตลอดจนแหล่งท่องเที่ยวเชิงวิชาการที่โดดเด่นอย่าง “ไดโนเสาร์” ซึ่งถือเป็นอัตลักษณ์ที่สร้างชื่อเสียงให้กับจังหวัด ขอนแก่นจึงเป็นหนึ่งในจุดหมายสำคัญของนักท่องเที่ยวทั้งชาวไทยและต่างชาติที่ต้องการสัมผัสประสบการณ์ที่หลากหลายและแตกต่าง

อย่างไรก็ตาม ปัญหาสำคัญที่ยังคงปรากฏอยู่คือ การเข้าถึงข้อมูลการท่องเที่ยวที่กระจัดกระจายอยู่ในหลายช่องทาง ไม่ว่าจะเป็นเว็บไซต์ หน่วยงานราชการ เพจโซเชียลมีเดีย หรือแอปพลิเคชันทั่วไป นักท่องเที่ยวจำเป็นต้องใช้เวลาค้นหาข้อมูลจากหลายแหล่ง ซึ่งมักพบปัญหาว่าข้อมูลไม่ครบถ้วน ไม่ทันสมัย หรือขาดการรองรับภาษาสำหรับนักท่องเที่ยวต่างชาติ ส่งผลให้การวางแผนท่องเที่ยวไม่มีประสิทธิภาพและอาจทำให้นักท่องเที่ยวพลาดโอกาสในการเข้าถึงประสบการณ์ที่ควรได้รับ

เพื่อแก้ไขข้อจำกัดดังกล่าว จึงเกิดแนวคิดในการพัฒนาโครงงาน “ไดโน: เว็บแอปพลิเคชันศูนย์กลางของการท่องเที่ยวจังหวัดขอนแก่น” ซึ่งเป็นแอปพลิเคชันที่มุ่งเน้นการรวมศูนย์ข้อมูลการท่องเที่ยวอย่างครบวงจร ภายในแอปประกอบด้วยฟังก์ชันการค้นหาแหล่งท่องเที่ยว ร้านอาหาร ที่พัก และกิจกรรมต่าง ๆ ระบบวางแผนการเดินทางที่จัดลำดับสถานที่ให้สอดคล้องกับระยะเวลา ความสนใจ และงบประมาณของผู้ใช้ รวมถึงระบบ Chatbot ที่ตอบคำถามโดยอ้างอิงเฉพาะข้อมูลในฐานความรู้ของระบบ ช่วยให้การค้นหาข้อมูลและการวางแผนทริปเป็นไปอย่างสะดวก รวดเร็ว และทันสมัย

นอกจากการอำนวยความสะดวกแก่นักท่องเที่ยวแล้ว แอปพลิเคชันยังเป็นเครื่องมือสำคัญในการสนับสนุนผู้ประกอบการท้องถิ่นให้สามารถโปรโมทธุรกิจได้ตรงกลุ่มเป้าหมาย อีกทั้งยังสามารถเก็บสถิติการใช้งาน เช่น จุดท่องเที่ยวยอดนิยม ช่วงฤดูกาลท่องเที่ยว หรือกิจกรรมที่ได้รับความสนใจสูง ซึ่งข้อมูลเหล่านี้สามารถนำไปใช้วิเคราะห์แนวโน้มและวางแผนการพัฒนาการท่องเที่ยวของจังหวัดในอนาคต

ดังนั้น การพัฒนาแอปพลิเคชัน “ไดโน” จึงไม่เพียงเป็นการแก้ปัญหาด้านการเข้าถึงข้อมูลการท่องเที่ยวเท่านั้น แต่ยังมีบทบาทสำคัญในการสร้างประสบการณ์การท่องเที่ยวที่ครบถ้วนสำหรับผู้มาเยือน และสนับสนุนเศรษฐกิจท้องถิ่นอย่างยั่งยืน ทั้งยังเป็นการวางรากฐานที่มั่นคงต่อการพัฒนาการท่องเที่ยวของจังหวัดขอนแก่นในระยะยาว

1.2 วัตถุประสงค์

1) เพื่อพัฒนาเว็บแอปพลิเคชันรวมศูนย์ข้อมูลการท่องเที่ยวของจังหวัดขอนแก่น ที่นักท่องเที่ยวสามารถเข้าถึงข้อมูลแหล่งท่องเที่ยว ร้านอาหาร และกิจกรรมต่าง ๆ ที่เกิดขึ้นในจังหวัดขอนแก่น

2) เพื่อพัฒนาและประยุกต์ใช้ระบบตอบโต้ผ่านโปรแกรมอัตโนมัติ (Chatbot) ในการให้บริการข้อมูลการท่องเที่ยว ตอบข้อซักถาม และให้คำแนะนำแก่ผู้ใช้งาน

3) เพื่อ พัฒนา ระบบ วางแผน การ ท่อง เที่ยว ที่ สามารถ จัด ลำดับ แผนการ เดิน ทาง ให้สอดคล้องกับความต้องการของผู้ใช้

1.3 ขอบเขตและข้อจำกัดของระบบ

1.3.1 ขอบเขต

การ วิจัย นี้ มุ่ง เน้น การ พัฒนา ระบบ วางแผน การ เดิน ทาง (Trip Planning System) สำหรับ จังหวัด ขอนแก่น โดย ประยุกต์ ใช้ เทคโนโลยี โมเดล ภาษา ขนาด ใหญ่ (Large Language Models: LLMs) ร่วมกับสถาปัตยกรรม Retrieval-Augmented Generation (RAG) เพื่อสร้างแผนการเดินทางที่สอดคล้องกับความต้องการของผู้ใช้ และสามารถอ้างอิงข้อมูลจากฐานความรู้ที่ได้รับการจัดเก็บและปรับปรุงอย่างต่อเนื่อง ทั้งนี้ ระบบต้นแบบบางส่วนของงานวิจัยนี้ได้รับการพัฒนาและทดสอบการทำงานเบื้องต้นแล้ว ณ ช่วงเวลาที่จัดทำเค้าโครงโครงงานนี้

ระบบที่พัฒนาประกอบด้วยองค์ประกอบหลัก ดังนี้

(1) ระบบ วางแผน การ เดิน ทาง (Trip Planning System) ใช้เทคโนโลยี LLMs ร่วมกับ RAG ในการสร้างแผนการเดินทาง โดยพิจารณาจากข้อมูลที่ผู้ใช้กำหนด ได้แก่ วันที่และระยะเวลาเดินทาง จังหวะการท่องเที่ยว (Pace) ซึ่งส่งผลต่อจำนวนสถานที่ที่บรรจุในแผนแต่ละวัน ความสนใจ (เช่น วัด พิพิธภัณฑ์ ธรรมชาติ หรือร้านอาหาร) งบประมาณในการเดินทาง และสถานที่ที่ผู้ใช้ต้องการไปเป็นพิเศษ (Must-go) เพื่อสร้างแผนการเดินทางที่เหมาะสมและสอดคล้องกับข้อกำหนดของผู้ใช้ ทั้งนี้ ระบบมีกลไกตรวจสอบข้อจำกัดพื้นฐาน (Constraint Verification) ที่ทำงานอิสระจากแบบจำลองภาษา เพื่อยืนยันความสมเหตุสมผลของแผนก่อนนำเสนอผลลัพธ์แก่ผู้ใช้

(2) ระบบฐานความรู้และการสืบค้นข้อมูล (Knowledge Retrieval System) ใช้ฐานข้อมูล PostgreSQL ที่มีส่วนขยายสำหรับจัดเก็บและค้นคืนข้อมูลเชิงเวกเตอร์ (pgvector) เป็นแหล่งจัดเก็บข้อมูลแหล่งท่องเที่ยว ร้านอาหาร ที่พัก กิจกรรม และเหตุการณ์ (Events) ภายในจังหวัดขอนแก่นแบบรวมศูนย์ โดยข้อมูลหลักรวบรวมจาก Google Places API และข้อมูลที่ผู้ดูแลระบบเพิ่มเข้าสู่ระบบผ่านหน้าจอผู้ดูแล ทั้งนี้ ในระยะถัดไปของงานวิจัยมีแผนขยายแหล่งข้อมูลให้ครอบคลุมแหล่งข้อมูลสาธารณะอื่นเพิ่มเติม เช่น สื่อสังคมออนไลน์ของผู้ประกอบการในพื้นที่ เพื่อเพิ่มความครบถ้วนและความทันสมัยของฐานความรู้ ข้อมูลที่จัดเก็บนี้สนับสนุนการค้นคืนข้อมูลสำหรับระบบ RAG และระบบตอบคำถามอัตโนมัติ

(3) ระบบ Chatbot (Chatbot System) ใช้เทคโนโลยี LLMs และ RAG ในการตอบคำถามเกี่ยวกับการท่องเที่ยวจังหวัดขอนแก่น โดยอ้างอิง

เฉพาะข้อมูลที่ค้นคืนได้จากฐานความรู้ของระบบ พร้อมกลไกตรวจจับกรณีที่ไม่พบข้อมูลตรงกับคำถาม เพื่อลดโอกาสการตอบคำถามที่คลาดเคลื่อนจากข้อมูลจริง (Hallucination) เพื่อให้ข้อมูลที่เกี่ยวข้องกับสถานที่ท่องเที่ยว ร้านอาหาร กิจกรรม การเดินทาง และข้อมูลที่ผู้ดูแลระบบเผยแพร่

เว็บแอปพลิเคชัน (Web-based Application) รองรับการใช้งานของผู้ใช้ 2 กลุ่ม ได้แก่

(1) ผู้ใช้งานทั่วไป สามารถค้นหาข้อมูลแหล่งท่องเที่ยว กิจกรรม และสร้างแผนการเดินทางผ่านเว็บไซต์ รวมถึงสมัครสมาชิกและเข้าสู่ระบบเพื่อใช้งานระบบสะสมคะแนนจากการสแกน QR Code ณ สถานที่ที่เข้าร่วมโครงการ ซึ่งเป็นความสามารถเพิ่มเติม (Optional Feature) ของระบบ

(2) ผู้ดูแลระบบ (Administrator) สามารถจัดการข้อมูลแหล่งท่องเที่ยว ร้านอาหาร กิจกรรมหรืออีเวนต์ในจังหวัดขอนแก่น รวมถึงจัดการฐานความรู้ที่ใช้สำหรับระบบ Chatbot ผ่านแดชบอร์ดผู้ดูแลระบบ

ในด้านสถาปัตยกรรม ระบบได้รับการออกแบบเป็นบริการอิสระ 3 ส่วน (เว็บแอปพลิเคชัน บริการข้อมูลหลัก และบริการปัญญาประดิษฐ์) ที่สื่อสารระหว่างกันผ่าน RESTful API เพื่อให้สามารถพัฒนาและทดสอบแต่ละส่วนแยกจากกันได้ โดยองค์ประกอบเสริมสำหรับรองรับการขยายระบบในระดับใช้งานจริง เช่น API Gateway และ Message Broker จะได้รับการพัฒนาเพิ่มเติมในระยะถัดไปของโครงการ

ขอบเขตของข้อมูลในการวิจัยครอบคลุมจังหวัดขอนแก่น โดยในระยะต้นแบบเน้นการรวบรวมข้อมูลสถานที่ท่องเที่ยว ร้านอาหาร ที่พัก และกิจกรรมในเขตอำเภอเมืองขอนแก่นเป็นหลัก ร่วมกับอำเภอที่มีศักยภาพด้านการท่องเที่ยวที่คัดเลือกไว้เบื้องต้น (เช่น ภูเวียง อุบลรัตน์ ภูผาม่าน ชุมแพ น้ำพอง และหนองเรือ) และมีแผนขยายการเก็บข้อมูลให้ครอบคลุมอำเภออื่นภายในจังหวัดเพิ่มเติมในระยะถัดไปของงานวิจัย

การประเมินผลของระบบมุ่งเน้นด้านความถูกต้องและความเหมาะสมของแผนการเดินทาง ความถูกต้องของการตอบคำถามของระบบ Chatbot ประสิทธิภาพของการค้นคืนข้อมูลผ่าน RAG ระยะเวลาในการสร้างแผนการเดินทาง และความพึงพอใจของผู้ใช้งาน โดยประเมินผ่านการทดสอบเชิงประสิทธิภาพและแบบสอบถามจากผู้ใช้งานจริง

1.3.2 ข้อจำกัดของโครงงาน

(1) ข้อจำกัดด้านพื้นที่และข้อมูล การวิจัยจำกัดขอบเขตข้อมูลเฉพาะจังหวัดขอนแก่น โดยในระยะต้นแบบข้อมูลยังกระจุกตัวอยู่ในเขตอำเภอเมืองและอำเภอท่องเที่ยวสำคัญบางส่วนเท่านั้น ยังไม่ครอบคลุมทุกอำเภอของจังหวัด ส่งผลให้การนำระบบไปประยุกต์ใช้กับพื้นที่อื่นหรือขยายความครอบคลุมภายในจังหวัดจำเป็นต้องรวบรวมและจัดเตรียมฐานข้อมูลเพิ่มเติม นอกจากนี้ ความครบถ้วน

และความถูกต้องของข้อมูลยังขึ้นอยู่กับแหล่งข้อมูลภายนอก (Google Places API) และข้อมูลที่ผู้ดูแลระบบนำเข้าสู่ระบบ

(2) ข้อจำกัดด้านการอัปเดตข้อมูล ข้อมูลสถานที่ท่องเที่ยว ร้านอาหาร และกิจกรรมอาจมีการเปลี่ยนแปลงอยู่ตลอดเวลา เช่น เวลาเปิด–ปิด การย้ายสถานที่ หรือการยกเลิกกิจกรรม การนำเข้าข้อมูลในระยะต้นแบบดำเนินการเป็นครั้งคราวผ่านกระบวนการดึงและนำเข้าข้อมูล มิใช่การซิงค์ข้อมูลแบบเรียลไทม์ หากไม่มีการปรับปรุงข้อมูลอย่างสม่ำเสมอ อาจส่งผลให้คำแนะนำของระบบไม่สอดคล้องกับสถานการณ์จริง

(3) ข้อจำกัดด้านเทคนิคของโมเดลปัญญาประดิษฐ์ แม้ระบบจะใช้สถาปัตยกรรม RAG ร่วมกับกลไกตรวจจับคำตอบที่ไม่มีข้อมูลรองรับ (Grounded Generation และ Hallucination Guard) เพื่อลดปัญหาการสร้างข้อมูลที่คลาดเคลื่อน (Hallucination) ของ LLMs แต่ยังคงมีโอกาสที่โมเดลจะสร้างคำตอบที่ไม่ถูกต้องหรือไม่สอดคล้องกับบริบทของข้อมูลได้ โดยเฉพาะในกรณีที่ฐานความรู้ไม่มีข้อมูลรองรับหรือข้อมูลไม่เพียงพอ

(4) ข้อจำกัดด้านค่าใช้จ่ายในการดำเนินงาน ระบบมีต้นทุนในการให้บริการ เช่น ค่าใช้จ่ายของเซิร์ฟเวอร์ ค่าใช้บริการ API ของแบบจำลองภาษา และค่าใช้จ่ายในการดูแลฐานข้อมูลที่จัดเก็บทั้งข้อมูลทั่วไปและข้อมูลเวกเตอร์ ซึ่งอาจส่งผลต่อความสามารถในการขยายระบบหรือรองรับผู้ใช้งานจำนวนมากในอนาคต

(5) ข้อจำกัดด้านสถาปัตยกรรมระบบ ระบบในระยะต้นแบบสื่อสารระหว่างบริการย่อยผ่าน RESTful API แบบ Synchronous เท่านั้น ยังไม่มีองค์ประกอบเสริม เช่น API Gateway หรือ Message Broker สำหรับการสื่อสารแบบ Asynchronous ซึ่งจำเป็นต่อการรองรับปริมาณผู้ใช้งานจำนวนมากในระดับใช้งานจริง องค์ประกอบดังกล่าวเป็นแนวทางการพัฒนาต่อยอดในระยะถัดไปของโครงการ

(6) ข้อจำกัดด้านระบบยืนยันตัวตนผู้ใช้ ระบบสมัครสมาชิกและเข้าสู่ระบบในระยะต้นแบบยังอยู่ในขั้นตอนการพัฒนา โดยกลไกการยืนยันตัวตนและการจัดการสิทธิ์ผู้ใช้ (Authentication & Authorization) ที่สมบูรณ์และมีความปลอดภัยในระดับใช้งานจริงจะได้รับการพัฒนาต่อในระยะถัดไปของโครงการ

(7) ข้อจำกัดของระบบสะสมคะแนน (Point System) ระบบสะสมคะแนนจากการสแกน QR Code เป็นความสามารถเพิ่มเติมของระบบ ซึ่งประสิทธิภาพในการสร้างการมีส่วนร่วมของผู้ใช้งานขึ้นอยู่กับจำนวนสถานที่ที่เข้าร่วมโครงการ ความร่วมมือจากหน่วยงานที่เกี่ยวข้อง และ

การดำเนินกิจกรรมส่งเสริมการท่องเที่ยวของจังหวัดขอนแก่น ซึ่งอยู่นอกเหนือการควบคุมโดยตรงของทีมวิจัย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) นักท่องเที่ยวสามารถเข้าถึงข้อมูลแหล่งท่องเที่ยว ร้านอาหาร และกิจกรรมในจังหวัดขอนแก่นได้อย่างครบถ้วนและสะดวกในที่เดียว

2) ช่วยให้ผู้ใช้สามารถวางแผนทริปที่เหมาะสมกับความสนใจ เวลา และงบประมาณได้อย่างมีประสิทธิภาพมากขึ้น โดยระบบจะแนะนำสถานที่และกิจกรรมที่สอดคล้องกับความสนใจของผู้ใช้งาน

3) ส่งเสริมการใช้ AI และ Chatbot ในการให้ข้อมูลเชิงโต้ตอบ ทำให้ประสบการณ์การท่องเที่ยวของผู้ใช้มีความเป็นส่วนตัวและทันสมัย

4) หน่วยงานที่เกี่ยวข้องสามารถนำข้อมูลจากระบบไปใช้ประโยชน์ในการวิเคราะห์เชิงนโยบาย เช่น การจัดสรรทรัพยากรด้านการท่องเที่ยวหรือการวางแผนกลยุทธ์การตลาด

5) เป็นกรณีศึกษาและต้นแบบในการพัฒนา Smart Tourism Platform ที่สามารถขยายผลไปยังจังหวัดอื่น ๆ หรือพัฒนาต่อยอดในระดับภูมิภาคได้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ในยุคดิจิทัล การพัฒนาการท่องเที่ยวอัจฉริยะ (Smart Tourism) ได้กลายเป็นประเด็นที่ได้รับความสนใจจากนักวิจัยและผู้พัฒนาแพลตฟอร์มทั่วโลก อันเป็นผลมาจากพฤติกรรมนักท่องเที่ยวที่เปลี่ยนแปลงอย่างรวดเร็ว และความคาดหวังที่สูงขึ้นต่อข้อมูลที่ถูกต้อง การเข้าถึงบริการที่สะดวก รวมถึงประสบการณ์การท่องเที่ยวที่สามารถตอบสนองความต้องการเฉพาะบุคคล (Sousa et al., 2024) ความท้าทายที่สำคัญจึงอยู่ที่การสร้างระบบที่สามารถนำเสนอข้อมูลแบบเรียลไทม์ ให้คำแนะนำที่แม่นยำ และสามารถปรับตามบริบทของผู้ใช้ได้อย่างมีประสิทธิภาพ

สำหรับจังหวัดขอนแก่น ซึ่งนับเป็นศูนย์กลางด้านการท่องเที่ยวของภาคตะวันออกเฉียงเหนือ ยังคงเผชิญกับปัญหาการกระจัดกระจายของข้อมูลการท่องเที่ยว แหล่งข้อมูลที่มีอยู่ในปัจจุบันมิได้ถูกรวบรวมอย่างเป็นระบบ ทำให้นักท่องเที่ยวที่ต้องการค้นหาแหล่งท่องเที่ยว ร้านอาหาร ที่พัก หรือกิจกรรมต่าง ๆ ต้องประสบความยากลำบากในการเข้าถึง อีกทั้งยังขาดเครื่องมือที่ช่วยในการวางแผนการเดินทางหรือแนะนำเส้นทางที่สอดคล้องกับความสนใจและงบประมาณของผู้ใช้ ส่งผลให้ประสบการณ์การท่องเที่ยวของจังหวัดขอนแก่นยังไม่สามารถสะท้อนศักยภาพที่มีอยู่ได้อย่างเต็มที่

เพื่อแก้ไขปัญหาดังกล่าว จึงได้มีการพัฒนาโครงงาน “Dino: The central web application for tourism in Khon Kaen” โดยมีเป้าหมายในการสร้างแพลตฟอร์มศูนย์กลางที่สามารถรวบรวมและจัดการข้อมูลด้านการท่องเที่ยวได้อย่างครบวงจร พร้อมทั้งบูรณาการเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence: AI) โดยเฉพาะโมเดลภาษาขนาดใหญ่ (LLMs) ร่วมกับสถาปัตยกรรม RAG เพื่อสนับสนุนการดำเนินงานใน 3 องค์ประกอบหลัก ได้แก่ ระบบฐานความรู้และการสืบค้นข้อมูล (Knowledge Retrieval System) ระบบวางแผนการเดินทาง (Trip Planning System) และระบบ Chatbot (Chatbot System)

งานวิจัยที่เกี่ยวข้องได้ชี้ให้เห็นว่า การประยุกต์ใช้ AI โดยเฉพาะการผสาน RAG สามารถเพิ่มความแม่นยำในการแนะนำข้อมูลและสร้างระบบที่ปรับตามบริบทของผู้ใช้ได้อย่างยั่งยืน (Banerjee et al., 2025) ขณะเดียวกัน งานวิจัยอีกด้านหนึ่งเน้นไปที่การพัฒนาระบบวางแผนการเดินทางแบบเรียลไทม์ซึ่งช่วยเพิ่มประสิทธิภาพในการสร้างประสบการณ์ที่เหมาะสมกับความสนใจและงบประมาณของผู้ใช้ (Deshmukh et al., 2025)

ดังนั้น โครงงาน Dino จึงมิใช่เพียงการสร้างเครื่องมือดิจิทัลเพื่ออำนวยความสะดวกแก่นักท่องเที่ยวเท่านั้น หากแต่ยังเป็นการบูรณาการองค์ความรู้จากงานวิจัยและเทคโนโลยีสมัยใหม่เพื่อต่อยอดสู่การพัฒนากลไกการจัดการท่องเที่ยวที่ชาญฉลาด เพิ่มศักยภาพในการแข่งขัน และยกระดับจังหวัดขอนแก่นสู่การเป็นเมืองท่องเที่ยวอัจฉริยะอย่างยั่งยืน

2.1.1 แนวคิดการท่องเที่ยวอัจฉริยะ (Smart Tourism)

Smart Tourism คือการใช้เทคโนโลยีในการสร้างระบบนิเวศท่องเที่ยวที่เชื่อมโยงข้อมูล บริการ และประสบการณ์ของผู้ใช้ งานของ Ma et al. (2025) แสดงให้เห็นว่า AI สามารถช่วยวิเคราะห์พฤติกรรมการเดินทางของผู้ใช้และปรับปรุงการจัดการจุดหมายปลายทางแบบเรียลไทม์ ในขณะที่งานของ Milton and Philosophers (2023) ได้สรุปแนวโน้มความท้าทายและโอกาสของการใช้ AI ใน Smart Tourism เช่น ปัญหาความเป็นส่วนตัวและธรรมาภิบาลข้อมูล

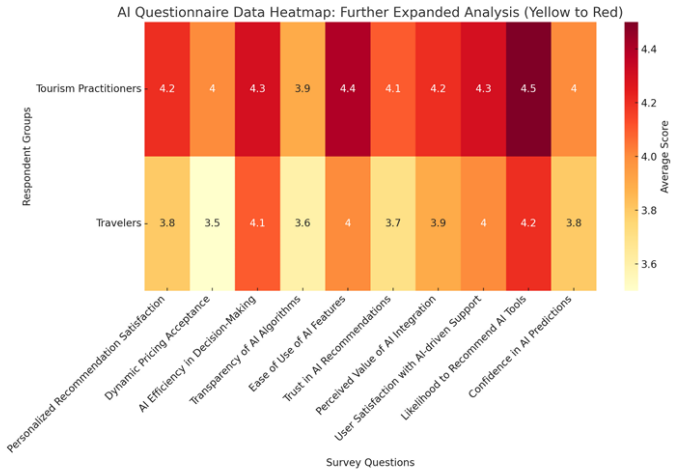
2.1.2 ปัญญาประดิษฐ์ (Artificial Intelligence) ในการท่องเที่ยว

ปัญญาประดิษฐ์ (AI) มีบทบาทสำคัญอย่างยิ่งในการปฏิวัติประสบการณ์ท่องเที่ยวยุคใหม่ เพราะช่วยให้แพลตฟอร์มสามารถวิเคราะห์พฤติกรรมและความต้องการของผู้ใช้แบบเรียลไทม์ งานของ Sousa et al. (2024) ศึกษามุมมองของนักท่องเที่ยวที่มีต่อการนำ AI มาใช้ในอุตสาหกรรมท่องเที่ยวและการบริการ พบว่านักท่องเที่ยวคาดหวังให้ระบบ AI สามารถให้คำแนะนำที่ “แม่นยำ” และ “เหมาะสมกับบริบท” แต่ในขณะเดียวกันยังมีความกังวลเรื่อง ความถูกต้องของข้อมูล และ ความเป็นส่วนตัว

ในขณะที่งานของ Milton and Philosophers (2023) วิเคราะห์เชิงลึกเกี่ยวกับโอกาสและความท้าทายในการนำ AI มาใช้ โดยชี้ให้เห็นว่า AI ช่วยเพิ่มศักยภาพในด้าน การวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data), การทำ personalization, และ การสร้างโมเดลทำนายพฤติกรรมนักท่องเที่ยว แต่ยังมีปัญหาเชิงเทคนิค เช่น การจัดการข้อมูลที่ซับซ้อนและการบูรณาการหลายแหล่งข้อมูลเข้าด้วยกัน

2.1.3 การจัดการจุดหมายปลายทางและการมีส่วนร่วมของนักท่องเที่ยว

แนวคิดการจัดการจุดหมายปลายทาง (Destination Management) และการสร้างประสบการณ์ที่ดึงดูดนักท่องเที่ยวมีความสำคัญอย่างมากในยุคที่ผู้ใช้ต้องการ ข้อมูลแบบเฉพาะบุคคล งานวิจัยของ Ma et al. (2025) นำเสนอการใช้ AI เพื่อวิเคราะห์ พฤติกรรมการเดินทางของผู้ใช้ ผ่านเทคนิค heatmap analytics และ data visualization ซึ่งช่วยให้ผู้พัฒนาระบบเข้าใจลักษณะการใช้งานของผู้ใช้และสามารถปรับกลยุทธ์การนำเสนอข้อมูลให้ตรงตามความสนใจของแต่ละคน



ภาพที่ 1 แผนภาพ Heatmap แสดงข้อมูลจากแบบสอบถาม (Ma et al., 2025)

จากภาพข้างต้น แสดง Heatmap แสดงการประเมิน AI ของผู้ประกอบวิชาชีพและนักท่องเที่ยวใน 6 มิติ โดยใช้สีเหลือง–แดงแทนคะแนนต่ำ–สูง ผู้ประกอบวิชาชีพให้คะแนนสูงกว่าทุกมิติ โดยเฉพาะความง่ายในการใช้งานและประสิทธิภาพการตัดสินใจ ขณะที่นักท่องเที่ยวต่ำในด้านการยอมรับราคาไดนามิกและความโปร่งใสของอัลกอริทึม จึงควรปรับปรุงความโปร่งใสราคา กลยุทธ์ราคาไดนามิก และการควบคุมโดยผู้ใช้เพื่อเสริมสร้างความเชื่อมั่น

ผลการประเมินพบว่า ผู้ประกอบวิชาชีพให้คะแนนสูงกว่านักท่องเที่ยวในทุกมิติ โดยเฉพาะด้านความง่ายในการใช้งานคุณลักษณะ AI (4.4 เทียบกับ 4.0) และประสิทธิภาพการตัดสินใจที่เสริมด้วย AI (4.3 เทียบกับ 4.1) สะท้อนระดับความคุ้นเคยและความมั่นใจกับเทคโนโลยี AI ที่สูงกว่า ขณะที่นักท่องเที่ยวให้คะแนนต่ำอย่างมีนัยในมิติการยอมรับราคาแบบไดนามิก (3.5) และความโปร่งใสของอัลกอริทึม (3.6) บ่งชี้ถึงความไม่ไว้วางใจต่อระบบ AI ความกังวลต่อราคาที่ไม่โปร่งใส กลไกคำแนะนำที่เข้าใจยาก และความสงสัยต่อความเป็นธรรมของ AI ทั้งในด้านการกำหนดราคาและการตัดสินใจ ซึ่งอาจทำให้ผู้ใช้บางส่วนจ่ายแพงกว่า หรือได้รับคำแนะนำที่ด้อยกว่า

ในทางกลับกัน Banerjee et al. (2025) เสนอว่าการสร้าง ระบบแนะนำที่ยั่งยืน (sustainable recommendation) จะช่วยเพิ่ม การมีส่วนร่วม (engagement) ของนักท่องเที่ยวในระยะยาว โดยใช้ RAG ผสานข้อมูลเรียลไทม์จากหลายแหล่ง เพื่อสร้างการแนะนำที่แม่นยำและสอดคล้องกับบริบท

2.1.4 ความยั่งยืน ความเสี่ยง และธรรมาภิบาลข้อมูล/อัลกอริทึม

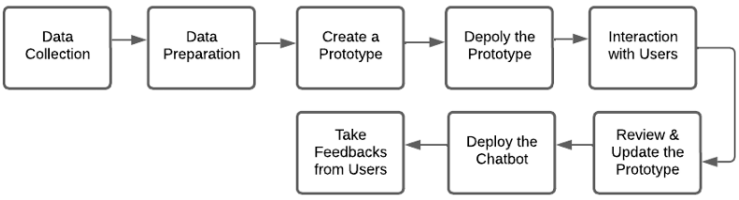
หนึ่งในประเด็นที่สำคัญสำหรับการพัฒนาแพลตฟอร์ม Smart Tourism คือการสร้างความสมดุลระหว่าง ประสบการณ์ผู้ใช้, ความยั่งยืน, และ การปกป้องข้อมูลส่วนบุคคล งานของ Banerjee et al. (2025) ชี้ให้เห็นว่าระบบแนะนำการท่องเที่ยวควรคำนึงถึง การใช้ทรัพยากรอย่างมีประสิทธิภาพ เพื่อลดผลกระทบทางสิ่งแวดล้อม เช่น การแนะนำเส้นทางที่ปล่อยคาร์บอนต่ำ

ในอีกมุมหนึ่ง งานของ Sousa et al. (2024) พบว่าผู้ใช้นักท่องเที่ยวให้ความสำคัญอย่างมากต่อ ความปลอดภัยของข้อมูล และความโปร่งใสของระบบ AI การละเลยประเด็น

นี้อาจทำให้ผู้ใช้เกิดความไม่เชื่อมั่นและลดการมีส่วนร่วมในระยะยาว

2.1.5 ทฤษฎีด้าน Chatbot

การใช้ Chatbot ในอุตสาหกรรมการท่องเที่ยวได้พัฒนาอย่างก้าวกระโดดในช่วงไม่กี่ปีที่ผ่านมา โดยมีการบูรณาการ Artificial Intelligence Markup Language (AIML) และ Natural Language Processing (NLP) เพื่อสร้างระบบสนทนาที่สามารถตอบคำถามของผู้ใช้ได้แบบเรียลไทม์ (Ali et al., 2023) งานวิจัยพบว่า Chatbot สำหรับการท่องเที่ยวสามารถช่วยลดเวลาในการค้นหาข้อมูลได้เฉลี่ย 45% และเพิ่มความพึงพอใจของนักท่องเที่ยวมากกว่า 70% เมื่อเทียบกับการค้นหาข้อมูลแบบเดิม



ภาพที่ 2 แผนภาพ Model Flow (University College of Goizha et al., 2023)

จากภาพข้างต้น แสดงขั้นตอนและเทคนิคที่ใช้ในการเก็บรวบรวม วิเคราะห์และตีความข้อมูลสำหรับการวิจัย โดยระเบียบวิธีประกอบด้วย 8 ขั้นตอน

(1) การเก็บรวบรวมข้อมูล: รวบรวมข้อมูลเกี่ยวกับเมือง Sulaimani ครอบคลุมโรงแรม ภัตตาคาร ศูนย์การค้า พิพิธภัณฑ์ และแหล่งท่องเที่ยวยอดนิยม

(2) การเตรียมข้อมูล: จัดโครงสร้างข้อมูลให้อยู่ในรูปแบบชุดคำถาม–คำตอบ และบันทึกเป็นไฟล์ AIML

(3) การสร้างต้นแบบ: สร้างแบบจำลองเริ่มต้นโดยใช้ไฟล์ AIML ที่จัดเตรียมไว้

(4) การเผยแพร่ต้นแบบ: นำต้นแบบไปใช้งานบนเว็บไซต์ที่โฮสต์ภายในเครื่อง (local)

(5) การโต้ตอบของผู้ใช้: ผู้ใช้ทำการโต้ตอบกับบอทที่ได้เผยแพร่

(6) การปรับปรุงต้นแบบ: ทบทวนและแก้ไขต้นแบบตามข้อเสนอแนะและพฤติกรรมการใช้งานของผู้ใช้

(7) การเผยแพร่โมเดลฉบับสมบูรณ์: เผยแพร่รุ่นสุดท้ายให้ผู้ใช้ใช้งานอีกครั้ง

(8) ข้อเสนอแนะจากผู้ใช้: หลังการเผยแพร่และการใช้งานบอท รวบรวมข้อเสนอแนะเพื่อประเมินความพึงพอใจ ความสามารถในการใช้งาน (usability) และความมีส่วนร่วมของผู้ใช้

ผลสำรวจของ Ali et al. (2023) ยังชี้ว่า Chatbot แบบ AI ที่มีระบบเรียนรู้พฤติกรรมผู้ใช้ (Adaptive-Chatbot) สามารถแนะนำสถานที่ท่องเที่ยว ร้านอาหาร และกิจกรรมที่เหมาะสมได้อย่างแม่นยำถึง 82% ทำให้ผู้ใช้รู้สึกว่าข้อมูลที่ได้รับมีความน่าเชื่อถือและเป็นประโยชน์มากกว่า Chatbot แบบ rule-based เดิม ๆ โดยมีนักท่องเที่ยวกว่า 64% ระบุว่าพร้อมใช้ระบบนี้หากข้อมูลถูกปรับให้เหมาะสมกับตนเอง

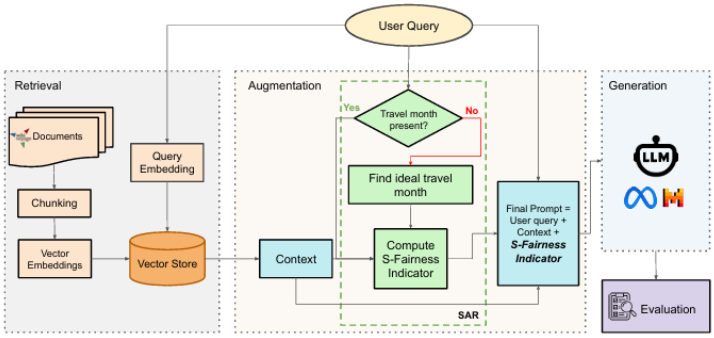
นอกจากนี้ แนวโน้มในงานวิจัยล่าสุดได้ขยายขีดความสามารถของ Chatbot ให้รองรับการแปลภาษาอัตโนมัติ การแนะนำกิจกรรมตามฤดูกาล และการโต้ตอบแบบมัลติมีเดีย เช่น การส่งแผนที่ รูปภาพ และลิงก์ข้อมูลท่องเที่ยว (Sousa et al., 2024) ซึ่งเป็นแนวคิดสำคัญสำหรับการพัฒนา Smart Tourism Applications ในอนาคต

2.1.6 ระบบแนะนำการท่องเที่ยว (Recommendation Systems)

การสร้างระบบแนะนำที่ตอบโจทย์ผู้ใช้เป็นหัวใจสำคัญสำหรับการท่องเที่ยวที่ตอบโจทย์ผู้ใช้แบบเฉพาะบุคคล (Personalized Experience) งานของ Banerjee et al. (2025) เสนอเทคนิค RAG ในการพัฒนาระบบแนะนำที่ใช้การเรียนรู้จากฐานข้อมูลจำนวนมหาศาลเพื่อสร้างข้อเสนอที่เหมาะสมที่สุดสำหรับผู้ใช้

ผลการทดลองของ Banerjee et al. (2025) พบว่าระบบ RAG-Based Tourism Recommender สามารถปรับปรุงความแม่นยำของคำแนะนำได้สูงถึง 91% เมื่อเทียบกับวิธี Collaborative Filtering แบบดั้งเดิม ซึ่งช่วยลดเวลาในการค้นหาแผนการเดินทางจากเฉลี่ย 15 นาที เหลือเพียง 4 นาที นอกจากนี้ยังมีการประเมินความพึงพอใจของผู้ใช้ (User Satisfaction) พบว่าผู้ใช้กว่า 88% พึงพอใจกับแผนการเดินทางที่ระบบแนะนำ

ในทำนองเดียวกัน งานของ Jyothi et al. (2025) ที่พัฒนา Travel Finder ซึ่งเป็นระบบแนะนำและจัดการทริปท่องเที่ยว พบว่าการใช้ Hybrid Recommendation (ผสม Content-Based + Collaborative Filtering) สามารถเพิ่มการมีส่วนร่วมของผู้ใช้มากกว่า 76% และช่วยสร้างชุมชนของผู้เดินทางที่มีความสนใจคล้ายกัน ซึ่งเป็นแนวโน้มสำคัญของ Smart Tourism Ecosystem



ภาพที่ 3 แผนภาพ Pipeline ของระบบ RAG ที่ได้รับการปรับปรุงตามข้อเสนอโดยกรอบเส้นประสีเขียวแสดง ขั้นตอนการเสริมข้อมูลที่ปรับเปลี่ยน พร้อมการเพิ่มตัวชี้วัดด้านความยั่งยืน (Banerjee et al., 2025)

จากภาพข้างต้น อธิบายองค์ประกอบในการสร้างคำแนะนำการท่องเที่ยว

Retrieval (การสืบค้นข้อมูล): เริ่มจากการนำเอกสารข้อมูลแหล่งท่องเที่ยวมาแบ่งส่วน (Chunking) สร้างเวกเตอร์ฝังตัว (Vector Embeddings) และจัดเก็บในฐานข้อมูลเวกเตอร์ เมื่อผู้ใช้ป้อนคำถาม ระบบจะสร้างเวกเตอร์คำค้น (Query Embedding) เพื่อดึงข้อมูลหรือบริบท (Context) ที่มีความสอดคล้องและเกี่ยวข้องที่สุดออกมา

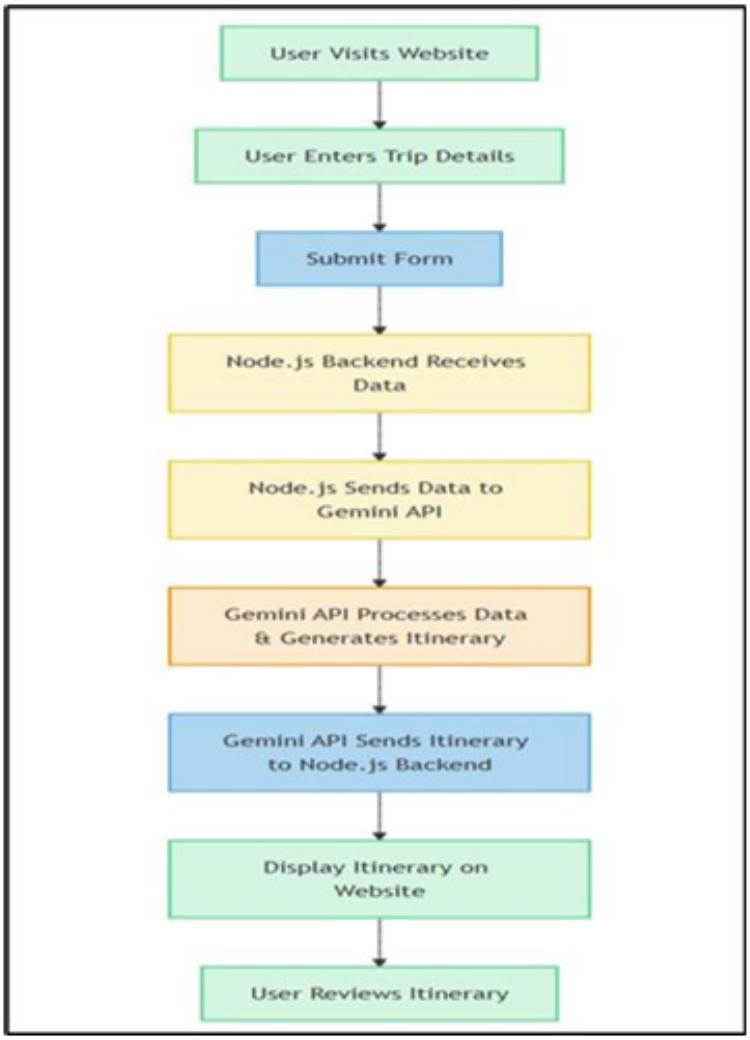
Augmentation (การเสริมข้อมูล): ระบบจะนำคำถามของผู้ใช้มารวมกับข้อมูลบริบทที่ดึงมาได้ เพื่อเสริมสร้างเป็น Prompt สุดท้ายที่มีความสมบูรณ์และมีข้อมูลอ้างอิงที่ถูกต้อง

Generation (การสร้างคำตอบ): ส่ง Prompt สุดท้ายที่ได้รับการเสริมข้อมูลแล้วไปให้โมเดลภาษาขนาดใหญ่ (LLM) ประมวลผลเพื่อสร้างคำตอบหรือแผนการเดินทางที่แม่นยำ พร้อมด้วยขั้นตอนการประเมินผล (Evaluation) ภายหลังการสร้างคำตอบ

2.1.7 การสร้างแผนการเดินทาง (Trip Planning)

การวางแผนการเดินทาง (Trip Planning) กำลังถูกประยุกต์ใช้ AI, Machine Learning และ Real-Time Data งานวิจัยของ Deshmukh et al. (2025) เสนอการพัฒนาระบบ AI-powered Trip Planner ที่ใช้ข้อมูลเรียลไทม์จากหลายแหล่ง เช่น สภาพอากาศ การจราจร ราคาที่พัก และความหนาแน่นของนักท่องเที่ยว เพื่อนำมาวิเคราะห์และสร้างแผนการเดินทางที่เหมาะสมที่สุด

ผลการทดลองพบว่า AI-powered Trip Planning สามารถปรับแผนการเดินทางให้เหมาะกับผู้ใช้แบบเรียลไทม์ และเพิ่มความแม่นยำในการแนะนำกิจกรรมได้สูงถึง 89% ขณะที่ช่วยลดค่าใช้จ่ายเฉลี่ยต่อทริปได้มากกว่า 12% เมื่อเทียบกับการวางแผนแบบ manual



ภาพที่ 4 แผนภาพสถาปัตยกรรมระบบ (Deshmukh et al., 2025)

จากภาพข้างต้น อธิบายองค์ประกอบและหน้าที่ของระบบวางแผนการเดินทางด้วย AI

(1) ผู้ใช้เข้าเว็บไซต์: กระบวนการเริ่มเมื่อผู้ใช้เปิดใช้งานเว็บแอปพลิเคชันผ่านอุปกรณ์ โดยแพลตฟอร์มจะแสดงอินเทอร์เฟซที่ใช้งานง่ายและสวยงามเพื่อแนะนำขั้นตอนการวางแผนทริป

(2) ผู้ใช้กรอกรายละเอียดทริป: ผู้ใช้ป้อนข้อมูลที่เกี่ยวข้องกับการเดินทาง ได้แก่ จุดหมายปลายทาง วันที่เดินทาง ระยะเวลา งบประมาณ ความสนใจ/ความชอบ

(3) ส่งแบบฟอร์ม: เมื่อกรอกข้อมูลครบถ้วน ผู้ใช้กดส่งแบบฟอร์ม เพื่อกระตุ้นให้มีการส่งคำขอไปยังระบบฝั่งเซิร์ฟเวอร์เพื่อประมวลผลต่อ

(4) ระบบฝั่งเซิร์ฟเวอร์รับข้อมูล: จะรับข้อมูลแบบฟอร์ม พร้อมทั้งดูแล

ความปลอดภัย ตรวจสอบความถูกต้อง และจัดรูปแบบข้อมูลก่อนส่งต่อไปยังกระบวนการ AI

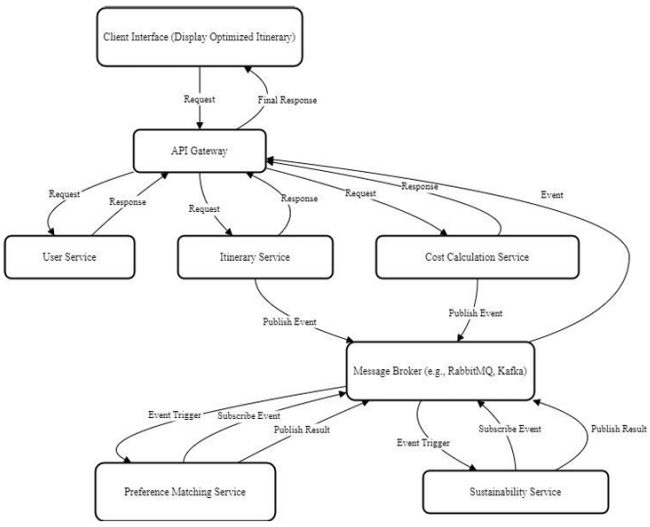
(5) ระบบฝั่งเซิร์ฟเวอร์ส่งข้อมูลไปยังโมเดล AI: ระบบฝั่งเซิร์ฟเวอร์ส่งข้อมูลที่จัดโครงสร้างแล้วไปยังโมเดลภาษาขนาดใหญ่ ซึ่งทำหน้าที่ทำความเข้าใจเจตนา ความชอบของผู้ใช้ และปัจจัยเชิงบริบท เช่น สภาพอากาศปัจจุบันหรือแนวโน้มการท่องเที่ยว

(6) โมเดล AI ประมวลผลและสร้างแผนการเดินทาง ด้วย NLP และอัลกอริทึม AI: ระบบจะตีความความต้องการของผู้ใช้และจัดทำแผนการเดินทางที่ปรับให้เหมาะสม โดยครอบคลุม กิจกรรมรายวัน คำแนะนำสถานที่ท่องเที่ยว ประสบการณ์ท้องถิ่นและร้านอาหาร วิธีการเดินทางและช่วงเวลา การผสานข้อมูลสภาพอากาศและคำแนะนำการเดินทางแบบเรียลไทม์

(7) โมเดล AI ส่งแผนการเดินทางกลับมายังฝั่งเซิร์ฟเวอร์: เมื่อสร้างแผนเสร็จสิ้น ระบบจะส่งผลลัพธ์กลับมาที่ฝั่งเซิร์ฟเวอร์เพื่อจัดรูปแบบให้อยู่ในโครงสร้างที่เหมาะสมต่อการแสดงผลบนหน้าเว็บ

(8) แสดงแผนการเดินทางบนเว็บไซต์: ส่วนติดต่อผู้ใช้จะแสดงแผนการเดินทางแบบไดนามิกด้วยโครงสร้างที่อ่านง่าย มีการแบ่งตามวัน แผนที่ ข้อเสนอแนะกิจกรรม และตัวเลือกการจองเพิ่มเติมตามความเหมาะสม

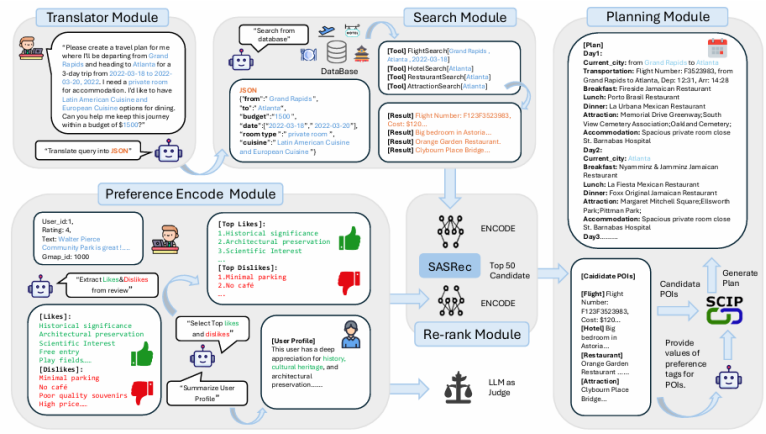
ในอีกด้านหนึ่ง สำหรับการออกแบบสถาปัตยกรรมเพื่อรองรับการประมวลผลข้อมูลที่ซับซ้อน งานวิจัยของ Barua and Kaiser (2024) ได้นำเสนอการใช้ สถาปัตยกรรมไมโครเซอร์วิส (Microservices Architecture) ซึ่งแบ่งการทำงานของระบบออกเป็นบริการย่อย ๆ (เช่น User Service, Itinerary Service และ Preference Matching Service) ที่ทำงานแยกจากกันแต่สามารถสื่อสารกันผ่าน API Gateway และ Message Broker สถาปัตยกรรมนี้ช่วยให้แพลตฟอร์มมีความยืดหยุ่น รองรับผู้ใช้งานจำนวนมากได้พร้อมกัน และสามารถประมวลผลแผนการเดินทางได้อย่างรวดเร็ว



ภาพที่ 5 แผนภาพการไหลของข้อมูลและกระบวนการประสานลำดับงาน (Barua & Kaiser, 2024)

จากภาพข้างต้น อธิบายองค์ประกอบและหน้าที่ของกระบวนการไหลข้อมูลและการประสานงาน: คำขอจากไคลเอนต์ทั้งหมดถูกส่งไปยัง API Gateway ซึ่งทำหน้าที่เป็นจุดทางเข้าและกำหนดเส้นทางคำขอไปยังไมโครเซอร์วิสที่เกี่ยวข้อง (เช่น User Service, Itinerary Service, Cost Calculation Service) ผ่านทั้งการสื่อสารแบบซิงโครนัส (RESTful APIs) และแบบอะซิงโครนัสผ่านตัวกลางส่งข้อความ (Message Broker เช่น RabbitMQ หรือ Apache Kafka) สำหรับบริการที่ทำงานแบบขับเคลื่อนด้วยอีเวนต์ (เช่น Sustainability Service, Preference Matching Service) ก่อนที่ API Gateway จะรวบรวมผลลัพธ์และส่งต่อแผนการเดินทางที่ปรับให้เหมาะสมไปยังส่วนติดต่อผู้ใช้

อย่างไรก็ตาม ในส่วนของกลไกการคำนวณเพื่อสร้างแผนการเดินทางที่ต้องตอบสนองทั้งข้อจำกัดและความชอบส่วนบุคคลที่ซับซ้อน งานวิจัยล่าสุดของ Shao et al. (2025) ได้นำเสนอระบบ Personal Travel Solver (PTS) ซึ่งเปลี่ยนจากการใช้อัลกอริทึมค้นหาแบบดั้งเดิม (Heuristic Algorithms) มาเป็นการบูรณาการความสามารถของ โมเดลภาษาขนาดใหญ่ (LLMs) เข้ากับ ตัวแก้ปัญหาเชิงตัวเลข (Numerical Solvers) ระบบ PTS ใช้ LLM ในการตีความคำสั่งภาษาธรรมชาติของผู้ใช้และวิเคราะห์ประวัติรีวิวเพื่อสกัดเป็นคะแนนความพึงพอใจ (Preference Score) จากนั้นจะแปลงข้อมูลเหล่านี้ให้อยู่ในรูปแบบเชิงสัญลักษณ์ (Symbolic Representation) เพื่อส่งต่อให้ Mathematical Solver (เช่น SCIP Solver) ทำการคำนวณหาเส้นทางและตารางเวลาที่ดีที่สุดภายใต้ข้อจำกัด



ภาพที่ 6 แผนภาพสถาปัตยกรรมระบบ PTS (Personal Travel Solver Framework) (Shao et al., 2025)

จากภาพข้างต้น อธิบายองค์ประกอบและหน้าที่ของแผนภาพสถาปัตยกรรมระบบ PTS: โมดูลการแปลความหมาย (Translator Module) รับคำสั่งภาษาธรรมชาติและใช้ LLM แปลงเป็นรูปแบบเชิงสัญลักษณ์เพื่อสกัดข้อจำกัดที่ชัดเจน; โมดูลการค้นหา (Search Module) นำเงื่อนไขไปสืบค้นข้อมูลเบื้องต้นจากฐานข้อมูล; โมดูลเข้ารหัสความชอบ (Preference Encode Module) ใช้ LLM วิเคราะห์ประวัติการรีวิวเพื่อสกัดคะแนนความพึงพอใจ; โมดูลจัดอันดับใหม่ (Re-rank Module) จัดเรียงลำดับผลลัพธ์ตามคะแนนความพึงพอใจ; และโมดูลการวางแผน (Planning Module) ส่งข้อมูลที่คัดกรองแล้วเข้าสู่ตัวแก้ปัญหาทางคณิตศาสตร์เพื่อคำนวณและจัดตารางเวลาที่สมบูรณ์แบบ

2.2 งานวิจัยที่เกี่ยวข้อง

การศึกษางานวิจัยที่เกี่ยวข้องถือเป็นขั้นตอนสำคัญสำหรับการสร้างกรอบแนวคิดในการพัฒนา Dino: The central web application for tourism in Khon Kaen ซึ่งมุ่งพัฒนาแพลตฟอร์มศูนย์กลางเพื่อสนับสนุนการท่องเที่ยวในจังหวัดขอนแก่น โดยรวบรวมข้อมูลแหล่งท่องเที่ยว ร้านอาหาร ที่พัก กิจกรรม และบริการต่าง ๆ พร้อมระบบวางแผนการเดินทางที่ใช้ปัญญาประดิษฐ์ (AI) เพื่อช่วยสร้างประสบการณ์การท่องเที่ยวที่เหมาะสมกับความต้องการของผู้ใช้งาน

เพื่อตอบโจทย์นี้ มีการทบทวนงานวิจัย 11 ชิ้น ที่เกี่ยวข้องกับ AI, Smart Tourism, Recommendation Systems, Chatbots, Sustainable Tourism, Traveler Engagement, Microservices Architecture และการประเมินผลระบบวางแผนการเดินทางด้วย LLM ซึ่งช่วยให้เห็นแนวโน้มหลักในการออกแบบระบบท่องเที่ยวอัจฉริยะ รวมถึงช่องว่างขององค์ความรู้ (Research Gaps) และแนวทางการต่อยอด (Implications) ที่ Dino สามารถพัฒนาได้

2.2.1 การใช้ AI เพื่อพัฒนาระบบแนะนำการท่องเที่ยวและ Trip Planning

งานของ Banerjee et al. (2025) นำเสนอการใช้ RAG เพื่อพัฒนาระบบแนะนำการท่องเที่ยวแบบยั่งยืน โดยระบบจะดึงข้อมูลจากหลายแหล่ง เช่น บทความรีวิว สถิติ

การท่องเที่ยว และโซเชียลมีเดีย จากนั้นสร้างคำแนะนำที่เหมาะสมกับความสนใจเฉพาะบุคคลมากขึ้น ซึ่งช่วยปรับปรุงคุณภาพของคำแนะนำให้ตรงกับโปรไฟล์ของผู้ใช้งาน

ผลการศึกษานี้สอดคล้องกับงานของ Jyothi et al. (2025) ที่พัฒนา Travel Finder ซึ่งใช้ Collaborative Filtering และ Clustering Algorithms เพื่อสร้างกลุ่มนักท่องเที่ยวตามความสนใจและลักษณะการเดินทาง ผลการทดลองพบว่าระบบสามารถลดเวลาในการค้นหาแหล่งท่องเที่ยวลง 34% และเพิ่มความพึงพอใจของผู้ใช้งานได้อย่างมีนัยสำคัญ ซึ่งสนับสนุนข้อค้นพบของ Banerjee et al. (2025) ว่าการใช้ AI เพื่อปรับคำแนะนำให้ตรงกับความต้องการส่วนบุคคล (Personalization) เป็นปัจจัยสำคัญต่อประสบการณ์ของนักท่องเที่ยว

อย่างไรก็ตาม ทั้งสองงานมีความแตกต่างกันในด้านแนวทาง Banerjee et al. (2025) มุ่งเน้นการสร้างระบบแนะนำเชิงข้อมูลเชิงลึก (Data-driven Recommendations) ผ่านการใช้โมเดลภาษาขนาดใหญ่ (LLMs) ขณะที่ Travel Finder ของ Jyothi et al. (2025) มุ่งเน้นการจับกลุ่มผู้ใช้ตามพฤติกรรมจริงเพื่อต่อยอดการแนะนำแบบ Social-based Recommendation ความแตกต่างนี้เปิดโอกาสให้ Dino สามารถผสมผสานข้อดีของทั้งสองวิธีเพื่อสร้างระบบที่ทั้ง แม่นยำ และ ยืดหยุ่น ในการแนะนำเส้นทางและกิจกรรมการท่องเที่ยว

2.2.2 AI และ Chatbot เพื่อยกระดับประสบการณ์ผู้ใช้

Ali et al. (2023) พัฒนาระบบ Chatbot Tourist Guide โดยใช้ AIML เพื่อสร้างการสนทนาที่เป็นธรรมชาติกับนักท่องเที่ยว ระบบนี้สามารถตอบคำถามพื้นฐาน เช่น สถานที่ท่องเที่ยว ร้านอาหาร และข้อมูลตั๋วเข้าชม ผลการทดลองกับนักท่องเที่ยว 150 คน พบว่า 78% ของผู้ใช้พึงพอใจกับความเร็วและความถูกต้องของคำตอบ นอกจากนี้ Chatbot ยังช่วยลดภาระงานของเจ้าหน้าที่ศูนย์บริการนักท่องเที่ยวลงถึง 40%

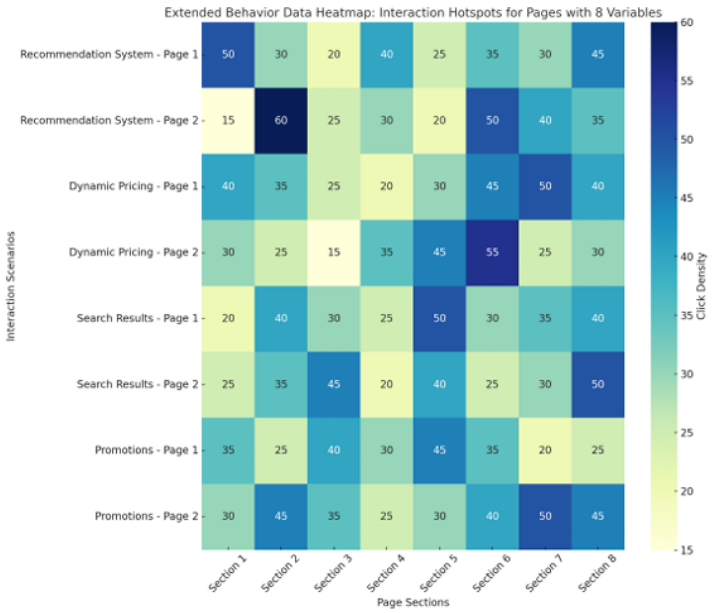
ผลนี้สอดคล้องกับงานของ Deshmukh et al. (2025) ที่พัฒนาแพลตฟอร์ม AI-powered Personalized Itinerary Planning ซึ่งมี Chatbot ทำงานร่วมกับ Recommendation Engine เพื่อสร้างแผนการท่องเที่ยวอัตโนมัติ โดย Chatbot สามารถดึงข้อมูลแบบเรียลไทม์ เช่น สภาพอากาศ การจราจร และราคาที่พัก เพื่อปรับคำแนะนำให้สอดคล้องกับสถานการณ์ปัจจุบัน ผลการทดสอบกับกลุ่มนักท่องเที่ยว 300 คน พบว่า 85% ของผู้ใช้ยืนยันว่าแผนการเดินทางที่ระบบสร้างให้มีความเหมาะสมและตรงตามความต้องการสูง

จากการสังเคราะห์ พบว่าทั้งสองงานสนับสนุนข้อสรุปสำคัญว่า Chatbot ไม่ได้เป็นเพียงเครื่องมืออำนวยความสะดวก แต่สามารถทำหน้าที่เป็น ตัวกลางในการวิเคราะห์ข้อมูลแบบเรียลไทม์ ซึ่งช่วยปรับแผนการท่องเที่ยวให้เหมาะสมมากขึ้น อย่างไรก็ตาม ความแตกต่างระหว่าง AIML-based Chatbot ของ Ali et al. (2023) และระบบ AI-integrated Chatbot ของ Deshmukh et al. (2025) แสดงให้เห็นโอกาสในการพัฒนา Dino ให้รองรับ Natural Language Processing (NLP) ที่มีความฉลาดและยืดหยุ่นมากยิ่งขึ้น

2.2.3 การจัดการข้อมูล การวิเคราะห์พฤติกรรม และการมีส่วนร่วมของนักท่องเที่ยว

Ma et al. (2025) ศึกษาการใช้ AI เพื่อเพิ่มการมีส่วนร่วมของนักท่องเที่ยว (Traveler Engagement) และการจัดการจุดหมายปลายทาง (Destination Management) โดยใช้เทคนิค Heatmap Analytics และ Real-time Behavioral Tracking เพื่อระบุพื้นที่ที่นักท่อง

เที่ยวให้ความสนใจสูง ผลการทดลองแสดงให้เห็นว่าข้อมูลเชิงพฤติกรรมช่วยให้หน่วยงานท้องถิ่นปรับปรุงการจัดสรรทรัพยากรและออกแบบแผนการตลาดที่ตอบโจทย์ได้มากขึ้น



ภาพที่ 7 แผนภาพ Heatmap แสดงข้อมูลเชิงพฤติกรรม (Ma et al., 2025)

การวิเคราะห์ Heatmap ของข้อมูลพฤติกรรมผู้ใช้จากสามแพลตฟอร์มท่องเที่ยวออนไลน์แสดงในภาพข้างต้น สื่อถึงการกระจาย “ความเข้มข้นของการปฏิสัมพันธ์” ของผู้ใช้ในส่วนต่าง ๆ ได้แก่ ระบบแนะนำ การกำหนดราคาแบบไดนามิก ผลการค้นหา และหน้าโปรโมชัน

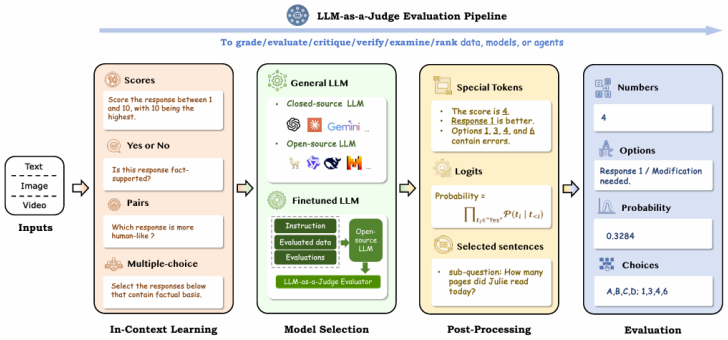
ขณะเดียวกัน Sousa et al. (2024) ทำการสำรวจความคิดเห็นนักท่องเที่ยวจำนวน 500 คน เกี่ยวกับการใช้ AI ในการวางแผนการเดินทางและการเข้าถึงข้อมูล ผลการศึกษาพบว่า 72% ของนักท่องเที่ยวเลือกใช้แพลตฟอร์มที่มี AI ในการแนะนำสถานที่ท่องเที่ยว และ 68% ยืนยันว่าพวกเขามีแนวโน้มที่จะกลับมาใช้บริการซ้ำ หากระบบสามารถปรับคำแนะนำให้เหมาะกับพฤติกรรมการท่องเที่ยวส่วนบุคคล

การเปรียบเทียบสองงานนี้พบว่าทั้ง Ma et al. (2025) และ Sousa et al. (2024) ชี้ให้เห็นถึงความสำคัญของ Data-driven Personalization และ Traveler Engagement แต่ต่างกันที่ Ma et al. (2025) มุ่งเน้นการวิเคราะห์เชิงเทคนิคเพื่อช่วยองค์กรปรับปรุงโครงสร้างการจัดการ ขณะที่ Sousa et al. (2024) มุ่งสำรวจพฤติกรรมและความต้องการของผู้บริโภคโดยตรง ซึ่งแสดงให้เห็นว่าการพัฒนา Dino ควรผสมผสานข้อมูลทั้งสองฝั่งเข้าด้วยกันเพื่อให้เกิดแพลตฟอร์มที่ ตอบโจทย์ทั้งนักท่องเที่ยวและผู้กำหนดนโยบาย

2.2.4 การประเมินผลด้วยโมเดลภาษาขนาดใหญ่ในฐานะผู้ประเมิน (LLM-as-a-Judge)

งานวิจัยสำรวจเชิงลึกโดย Gu et al. (2025) ได้วิเคราะห์ขีดความสามารถและความน่าเชื่อถือของกรอบการทำงานแบบ LLM-as-a-Judge อย่างเป็นระบบ โดยระบุว่าโมเดลภาษาขนาดใหญ่มีความสามารถในการตีความภาษาธรรมชาติและจำลองกระบวนการตัดสินใจได้

ใกล้เคียงกับมนุษย์ ส่งผลให้มีความยืดหยุ่นในการประเมินคำตอบเชิงคุณภาพที่มีลักษณะปลายเปิด (Open-ended Evaluation) ได้อย่างมีประสิทธิภาพ เหนือกว่าตัววัดผลเชิงสถิติแบบดั้งเดิม อีกทั้งยังช่วยลดต้นทุนและระยะเวลาในการทดสอบระบบลงอย่างมีนัยสำคัญ อย่างไรก็ตาม งานวิจัยดังกล่าวยังชี้ให้เห็นถึงข้อจำกัดและอคติแฝงของโมเดลภาษาในฐานะผู้ประเมิน เช่น อคติจากความยาวของคำตอบ (Verbosity Bias) อคติจากลำดับตัวเลือก (Position Bias) และความผันผวนของการให้คะแนน จึงเสนอกลไกเพิ่มความน่าเชื่อถือของการประเมินผ่านการกำหนดเกณฑ์การให้คะแนนที่ชัดเจน (Rubrics) การใช้คำตอบอ้างอิงประกอบการตัดสิน (Reference-based Evaluation) และการประเมินซ้ำเพื่อหาค่าเฉลี่ยสถิติ ซึ่งช่วยเพิ่มระดับความสอดคล้องระหว่างการประเมินของโมเดลและมนุษย์ (Human Alignment) ได้อย่างมีประสิทธิภาพ



ภาพที่ 8 สถาปัตยกรรมกระบวนการประเมินผลและการประมวลผลคำตอบของ LLM-as-a-Judge (Gu et al., 2025)

จากภาพข้างต้น แสดงกรอบแนวคิดกระบวนการประเมินแบบ LLM-as-a-Judge ซึ่งเริ่มต้นจากการรับโจทย์/คำสั่ง (Prompt/Query) และคำตอบที่ต้องการประเมิน (Candidate Response) ป้อนเข้าสู่โมเดลภาษาขนาดใหญ่ที่ทำหน้าที่เป็นผู้ประเมิน (Judge Model) โดยผ่านกลไกการกำหนดเกณฑ์ (Rubrics) และคำตอบอ้างอิง (Reference Answer) จากนั้นโมเดลจะสร้างกระบวนการวิเคราะห์เชิงตรรกะแบบ Chain-of-Thought (CoT) เพื่อให้เหตุผลรองรับ ก่อนส่งออกผลการประเมินในรูปแบบคะแนนเชิงปริมาณหรือการจัดอันดับคุณภาพ ซึ่งช่วยลดความผันผวนและเพิ่มความน่าเชื่อถือของการประเมินผลระบบตอบโต้

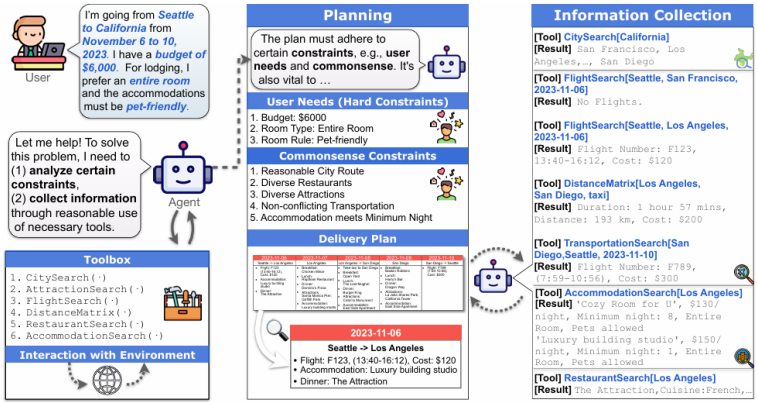
จากการวิเคราะห์งานวิจัยของ Gu et al. (2025) โครงงาน Dino จึงได้นำแนวคิด LLM-as-a-Judge มาประยุกต์ใช้ในส่วนการประเมินผลระบบตอบโต้และคำแนะนำสถานที่ท่องเที่ยว โดยตระหนักถึงทั้งจุดเด่นด้านความรวดเร็วและข้อควรระวังเรื่องอคติของโมเดล ระบบจึงออกแบบให้มีการกำหนดเกณฑ์การประเมินความถูกต้องของเนื้อหาเชิงประวัติศาสตร์และวัฒนธรรมของจังหวัดขอนแก่นอย่างรัดกุม ร่วมกับการใช้กลไก Chain-of-Thought เพื่อบังคับให้โมเดลผู้ประเมินแสดงเหตุผลประกอบก่อนสรุปคะแนน ส่งผลให้กระบวนการวัดผลของโครงงานมีความน่าเชื่อถือ โปร่งใส และรองรับการทดสอบในปริมาณมากได้อย่างมีประสิทธิภาพ

2.2.5 การประเมินผลและวัดประสิทธิภาพการวางแผนการเดินทางด้วย LLM

การประยุกต์ใช้โมเดลภาษาขนาดใหญ่ในระบบวางแผนการเดินทาง (Trip Planning) และระบบแนะนำสถานที่ท่องเที่ยว ได้รับความสนใจอย่างแพร่หลายในปัจจุบัน

อย่างไรก็ตาม ความท้าทายสำคัญของโมเดลภาษาขนาดใหญ่คือการประมวลผลข้อจำกัดเชิงตรรกะที่ซับซ้อน การคำนวณตารางเวลาและงบประมาณผิดพลาด ตลอดจนการเกิดปัญหาการสร้างข้อมูลเท็จ ด้วยเหตุนี้ จึงเกิดแนวทางการพัฒนากรอบการประเมินเพื่อวัดประสิทธิภาพของระบบวางแผนการเดินทางด้วยปัญญาประดิษฐ์ในบริบทการใช้งานจริงอย่างเป็นระบบ

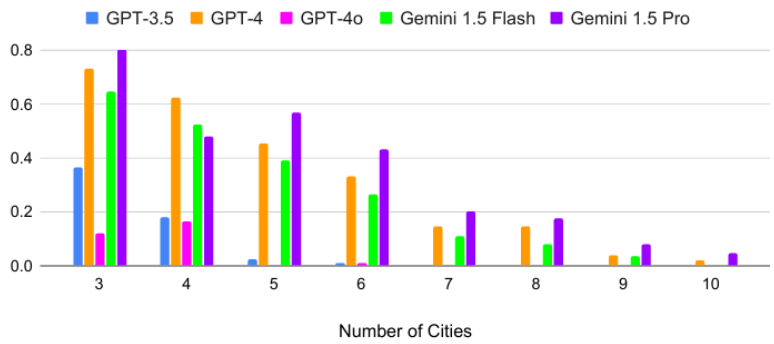
งานวิจัยของ Xie et al. (2024) ได้เสนอกรอบการประเมิน TravelPlanner ซึ่งเป็น Benchmark มาตรฐานสำหรับทดสอบขีดความสามารถของ AI Agents ในการวางแผนการเดินทางในโลกจริง งานวิจัยดังกล่าวเกิดขึ้นจากข้อจำกัดของการประเมินแบบดั้งเดิมที่มักวัดผลเฉพาะภารกิจการตอบคำถามสั้น ๆ ซึ่งไม่ครอบคลุมกระบวนการตัดสินใจหลายขั้นตอน (Complex Multi-step Tasks) โดยกำหนดให้ตัวแทนปัญญาประดิษฐ์ทำหน้าที่ดึงข้อมูลผ่านการเชื่อมต่อ API กับฐานข้อมูลจำลอง แล้วนำมาประเมินความถูกต้องของแผนการเดินทาง (Pass Rate) ใน 3 มิติหลัก ได้แก่ ข้อจำกัดเชิงสภาพแวดล้อม (Environment Constraints) เพื่อตรวจสอบว่าสถานที่ เที่ยวบิน หรือโรงแรม มีอยู่จริงในฐานข้อมูลโดยไม่เกิดการหลอนข้อมูล ข้อจำกัดเชิงสามัญสำนึก (Commonsense Constraints) เพื่อประเมินความสมเหตุสมผลของตารางเวลาและข้อจำกัดตามเงื่อนไขผู้ใช้ (Hard Constraints) เพื่อตรวจสอบความถูกต้องด้านงบประมาณและเงื่อนไขเฉพาะบุคคล



ภาพที่ 9 กรอบแนวคิดการประเมินการวางแผนการเดินทาง (Xie et al., 2024)

ผลการทดลองของ Xie et al. (2024) พบว่าโมเดลภาษาขนาดใหญ่ระดับนำส่วนใหญ่มีอัตราการผ่านเกณฑ์ซับซ้อนสมบูรณ์ต่ำกว่าร้อยละ 10 เนื่องจากมักพบปัญหาการคำนวณงบประมาณสะสมผิดพลาดและการจัดตารางเวลาที่ขัดกับสามัญสำนึก

ขณะเดียวกัน Zheng et al. (2024) จาก Google DeepMind ได้เสนอแนวทางการประเมิน NATURAL PLAN เพื่อทดสอบขีดความสามารถเชิงตรรกะและการวางแผน (Natural Language Planning) ของโมเดลภาษาขนาดใหญ่เมื่อต้องรับคำสั่งภาษาธรรมชาติที่มีข้อจำกัดเชื่อมโยงกันเป็นห่วงโซ่ซับซ้อน โดยไม่พึ่งพาการเรียกใช้เครื่องมือภายนอก แต่มุ่งเน้นการป้อนข้อมูลบริบทครบถ้วน (Full-Context) เพื่อประเมินความสามารถในการวางแผนการเดินทางเชื่อมโยงหลายเมือง (Multi-city Trips)



ภาพที่ 10 เปรียบเทียบแนวโน้มประสิทธิภาพของ LLM เมื่อระดับความซับซ้อนและจำนวนเมืองในแผนการเดินทางเพิ่มขึ้น (Zheng et al., 2024)

ผลการศึกษาของ Zheng et al. (2024) ชี้ให้เห็นว่าเมื่อจำนวนเมืองและข้อจำกัดเพิ่มขึ้นจาก 2 เมืองเป็น 5 ถึง 10 เมือง อัตราความถูกต้องในการสร้างแผนงานของโมเดลภาษาชั้นนำอย่าง GPT-4o และ Gemini 1.5 Pro จะลดลงอย่างรุนแรงจนเหลือต่ำกว่าร้อยละ 5 ซึ่งสะท้อนถึงข้อจำกัดของโมเดลภาษาในการประมวลผลข้อจำกัดเชิงตัวเลข เวลา และตรรกะซับซ้อนด้วยตนเองเพียงอย่างเดียว

จากการสังเคราะห์ผลงานวิจัยของ Xie et al. (2024) และ Zheng et al. (2024) โครงงาน Dino จึงได้นำแนวคิดการประเมินข้อจำกัด 3 ระดับ ได้แก่ ข้อจำกัดเชิงสภาพแวดล้อม ข้อจำกัดเชิงสามัญสำนึก และข้อจำกัดตามเงื่อนไขผู้ใช้ มาประยุกต์ใช้เป็นเกณฑ์ประเมินเชิงปริมาณในส่วนการทดสอบประสิทธิภาพของระบบวางแผนการเดินทาง ควบคู่กับการนำแนวคิดการประเมินแบบ LLM-as-a-Judge และกลไก RAG มาใช้วัดผลความถูกต้องเชิงบริบทของระบบตอบโต้ เพื่อป้องกันการหลอนข้อมูล และส่งมอบแผนการท่องเที่ยวจังหวัดขอนแก่นที่มีความถูกต้อง สมเหตุสมผล และตรงตามความต้องการของผู้ใช้งานอย่างมีประสิทธิภาพ

2.2.6 สรุปผลการเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

จากการทบทวนวรรณกรรมข้างต้น สามารถสรุปผลการเปรียบเทียบได้ดังตารางที่ 1

ตารางที่ 1 เปรียบเทียบงานวิจัยที่เกี่ยวข้อง

งานวิจัย	RAG / LLM	Recomm.	Chatbot	Trip Planning
Banerjee et al. (2025)	✓	✓	–	–
Jyothi et al. (2025)	–	✓	–	✓
Ali et al. (2023)	–	–	✓	–
Deshmukh et al. (2025)	–	✓	✓	✓
Shao et al. (2025)	✓	–	–	✓
Barua & Kaiser (2024)	–	–	–	✓
โครงงานนี้ (Dino)	✓	✓	✓	✓

จากการวิเคราะห์งานวิจัยทั้ง 11 ชิ้น พบว่าแนวโน้มหลักของการวิจัยด้าน AI ในการท่องเที่ยว มุ่งเน้นไปที่การสร้างประสบการณ์การท่องเที่ยวที่เป็นส่วนบุคคลมากขึ้น และการพัฒนาระบบสนับสนุนการตัดสินใจของผู้ใช้ อย่างไรก็ตาม เมื่อสังเคราะห์เนื้อหาของงานวิจัยพบว่า ช่องว่างองค์ความรู้ที่สำคัญยังคงอยู่ในประเด็น การบูรณาการข้อมูลหลายระบบเข้าด้วยกัน กล่าวคือแต่ละงานมักมุ่งเน้นความสามารถเพียงด้านใดด้านหนึ่ง (การแนะนำ, Chatbot, การวางแผนทริป, การประเมินผล หรือสถาปัตยกรรมระบบ) โดยยังไม่มีงานใดผสานทั้ง RAG-based Recommendation, Chatbot แบบเรียลไทม์, ระบบวางแผนทริป และกรอบการประเมินผลแบบ LLM-as-a-Judge เข้าไว้ในแพลตฟอร์มเดียวกัน นอกจากนี้ แม้ Xie et al. (2024) และ Zheng et al. (2024) จะได้สร้าง Benchmark ประเมินผลระดับสากลไว้แล้ว แต่กรอบการประเมินส่วนใหญ่ยังจำกัดอยู่ในข้อมูลระดับโลกหรือข้อมูลจำลอง ยังขาดการประยุกต์ใช้กับระบบท่องเที่ยวระดับพื้นที่ (Local Tourism) เช่น จังหวัดขอนแก่น ซึ่งเป็นช่องว่างที่โครงงาน Dino มุ่งเข้าไปแก้ไข นอกจากนี้ Sousa et al. (2024) และ Milton and Philosophers (2023) ยังชี้ให้เห็นถึงความท้าทายด้าน ความเป็นส่วนตัวของข้อมูลและธรรมาภิบาล ซึ่งยังไม่ได้รับการแก้ไขอย่างสมบูรณ์ในงานวิจัยส่วนใหญ่

2.2.7 เทคโนโลยีและเครื่องมือที่เกี่ยวข้อง

(1) Frontend: React

ใช้ React ในการพัฒนาส่วนติดต่อผู้ใช้ ทั้งฝั่งเว็บและฝั่งมือถือ (Web/Mobile) เพื่อสร้างส่วนติดต่อผู้ใช้แบบ component-based ที่นำกลับมาใช้ซ้ำได้และอัปเดตข้อมูลแบบเรียลไทม์

(2) Backend: Node.js (Express) และ Python (FastAPI)

ระบบฝั่งเซิร์ฟเวอร์แบ่งเป็น 2 บริการอิสระต่อกัน (Independent Services) คือ Backend API พัฒนาด้วย Node.js ร่วมกับเฟรมเวิร์ก Express สำหรับให้บริการ RESTful API ทั่วไปของระบบ และ AI Service พัฒนาด้วย Python ร่วมกับเฟรมเวิร์ก FastAPI แยกต่างหากสำหรับส่วนที่เกี่ยวข้องกับปัญญาประดิษฐ์และ RAG Pipeline โดยทั้งสองบริการสื่อสารระหว่างกันผ่าน RESTful API เท่านั้น ยังไม่มี API Gateway หรือ Message Broker ในระยะต้นแบบนี้

(3) ฐานข้อมูล: PostgreSQL พร้อมส่วนขยาย pgvector

ใช้ฐานข้อมูล PostgreSQL (ผ่านบริการ Supabase) ที่มีส่วนขยายสำหรับจัดเก็บและค้นคืนข้อมูลเชิงเวกเตอร์ (pgvector) เป็นแหล่งจัดเก็บข้อมูลแบบรวมศูนย์ ทั้งข้อมูลทั่วไปของระบบ (เช่น ข้อมูลสถานที่ ผู้ใช้ กิจกรรม) และ Embeddings ที่ใช้ในการสืบค้นข้อมูลของระบบ RAG ไว้ในฐานข้อมูลเดียวกัน โดยเวกเตอร์ฝังตัวสร้างขึ้นด้วยโมเดล paraphrase-multilingual-MiniLM-L12-v2 (384 มิติ) ผ่านไลบรารี sentence-transformers ซึ่งประมวลผลบนเครื่องคอมพิวเตอร์ในเครื่อง (Local Inference)

(4) AI & Libraries: LLM ผ่าน OpenAI-compatible API

ใช้โมเดลภาษาขนาดใหญ่ (LLM) gemini-2.5-flash ผ่านช่องทาง API ที่รองรับมาตรฐาน OpenAI-compatible ร่วมกับไลบรารีสำหรับสร้าง RAG Pipeline เพื่อสนับสนุนระบบแชตบอตและระบบวางแผนการเดินทาง

(5) API ภายนอก: Google Places API

ใช้ Google Places API สำหรับดึงข้อมูลตั้งต้น (Data Seeding) ของสถานที่ท่องเที่ยว ร้านอาหาร และกิจกรรมในจังหวัดขอนแก่น

บทที่ 3
วิธีการดำเนินโครงงาน

3.1 ขั้นตอนการดำเนินงาน

การดำเนินงานวิจัยนี้แบ่งออกเป็น 5 ส่วนหลัก เพื่อให้การพัฒนาแอปพลิเคชันเป็นไปอย่างมีระบบและครอบคลุมทั้งด้านเทคโนโลยี การจัดการข้อมูล การออกแบบสถาปัตยกรรมระบบ และการประเมินผลลัพธ์ โดยเนื้อหาในส่วนนี้อ้างอิงจากการออกแบบและพัฒนาต้นแบบระบบ (Prototype) ที่ดำเนินการจริงในงานวิจัย ได้แก่ เว็บแอปพลิเคชันแนะนำการท่องเที่ยวจังหวัดขอนแก่นที่ประกอบด้วยระบบ Chatbot ให้คำแนะนำและระบบวางแผนการเดินทางที่ขับเคลื่อนด้วยโมเดลภาษาขนาดใหญ่ (LLM) ร่วมกับเทคนิค RAG

3.1.1 ศึกษาปัญหาและรวบรวมความต้องการ

(1) การศึกษาวรรณกรรมและทฤษฎีที่เกี่ยวข้อง

ขั้นตอนนี้มีวัตถุประสงค์เพื่อรวบรวมและวิเคราะห์แนวคิด เทคนิค และเครื่องมือที่ทันสมัยในการพัฒนาระบบแนะนำการท่องเที่ยว (Tourism Recommender Systems: TRS) และระบบอัจฉริยะ เพื่อใช้เป็นกรอบแนวคิดในการออกแบบระบบ

(1) **การศึกษาระบบแนะนำการท่องเที่ยวและการวางแผนการเดินทาง:** ศึกษาแนวคิดพื้นฐานของระบบ TRS และวิวัฒนาการไปสู่ระบบที่พิจารณาหลายภาคส่วน (Multistakeholder Recommendation) โดยเฉพาะแนวโน้มการบูรณาการปัจจัยด้านความยั่งยืน (Sustainability) เข้าไปในการวางแผนการเดินทาง ซึ่งครอบคลุมทั้งมิติเศรษฐกิจ สังคม และสิ่งแวดล้อม แนวคิดนี้ถูกนำมาใช้เป็นกรอบอ้างอิงในการออกแบบเกณฑ์การให้คำแนะนำของระบบ กล่าวคือ การจัดลำดับและคัดเลือกสถานที่ในแผนการเดินทางของงานวิจัยนี้พิจารณาปัจจัยด้านงบประมาณ ระยะเวลา ความสนใจของผู้ใช้ และระยะทาง/เวลาการเดินทางระหว่างสถานที่ร่วมกัน โดยยังไม่ได้นำปัจจัยด้านผลกระทบสิ่งแวดล้อมเชิงปริมาณ (เช่น Carbon Footprint) มาใช้ในการคำนวณโดยตรงในต้นแบบปัจจุบัน ซึ่งจะกล่าวถึงในฐานะแนวทางการต่อยอดงานวิจัยในอนาคต

(2) **การศึกษาเทคโนโลยีปัญญาประดิษฐ์สำหรับการประมวลผลภาษาธรรมชาติ:** ศึกษานวัตกรรมที่ใช้โมเดลภาษาขนาดใหญ่ (LLMs) และการประมวลผลภาษาธรรมชาติ (NLP) สำหรับการสร้างคำแนะนำและการตอบโต้กับผู้ใช้งาน โดยเน้นที่สถาปัตยกรรม RAG ซึ่งเป็นแนวทางที่มีประสิทธิภาพในการ

จัดการข้อมูลการท่องเที่ยวที่มีการเปลี่ยนแปลงอยู่เสมอ (Dynamic Data) โดยไม่จำเป็นต้องปรับจูนโมเดล LLM (Fine-tuning) ซ้ำทุกครั้งที่ข้อมูลเปลี่ยนแปลง แนวคิดนี้ถูกนำไปประยุกต์ใช้จริงในทั้งส่วนของระบบ Chatbot และระบบวางแผนการเดินทางของงานวิจัยนี้

(3) **การศึกษาสถาปัตยกรรมระบบสำหรับแอปพลิเคชัน AI:** ศึกษาแนวทางการออกแบบระบบแบบแยกส่วนบริการ (Service-Oriented / Microservices Architecture) ซึ่งช่วยให้แต่ละองค์ประกอบของระบบสามารถพัฒนา ทดสอบ และปรับขนาด (Scale) ได้อย่างเป็นอิสระต่อกัน อันเหมาะสมกับแอปพลิเคชัน AI ที่มีภาระการประมวลผลแตกต่างกันในแต่ละส่วน (เช่น การประมวลผลภาษาธรรมชาติที่ใช้ทรัพยากรสูงกว่าการให้บริการข้อมูลทั่วไป) แนวคิดนี้ถูกนำมาใช้เป็นหลักในการออกแบบระบบจริงของงานวิจัย

(2) การเก็บรวบรวมและเตรียมข้อมูล

การเตรียมข้อมูลถือเป็นรากฐานสำคัญในการสร้างระบบอัจฉริยะที่แม่นยำและน่าเชื่อถือ

(1) **การรวบรวมข้อมูลแหล่งท่องเที่ยวและกิจกรรม:** ข้อมูลหลักของระบบรวบรวมจาก Google Places API (New) ครอบคลุมข้อมูลสถานที่ท่องเที่ยว โรงแรม/ที่พัก และร้านอาหารในพื้นที่จังหวัดขอนแก่น โดยข้อมูลที่ดึงมาประกอบด้วยชื่อสถานที่ ประเภท/แท็ก พิกัดภูมิศาสตร์ คะแนนรีวิว ระดับราคา เวลาทำการ (รวมช่วงเวลาเปิด–ปิดแบบละเอียดรายวัน) และสถานะทางธุรกิจ (Business Status) การนำเข้าข้อมูลในระยะต้นแบบนี้ดำเนินการผ่านสคริปต์ดึงและนำเข้าข้อมูลแบบครั้งเดียว มิใช่การซิงค์ข้อมูลแบบเรียลไทม์ โดยผู้ดูแลระบบสามารถเพิ่มหรือแก้ไขข้อมูลสถานที่ กิจกรรม และฐานความรู้เพิ่มเติมภายหลังได้ผ่านหน้าจอผู้ดูแลระบบ (Admin Dashboard) ที่พัฒนาขึ้นเป็นส่วนหนึ่งของระบบ

(2) **การเตรียมและจัดเก็บข้อมูล:**

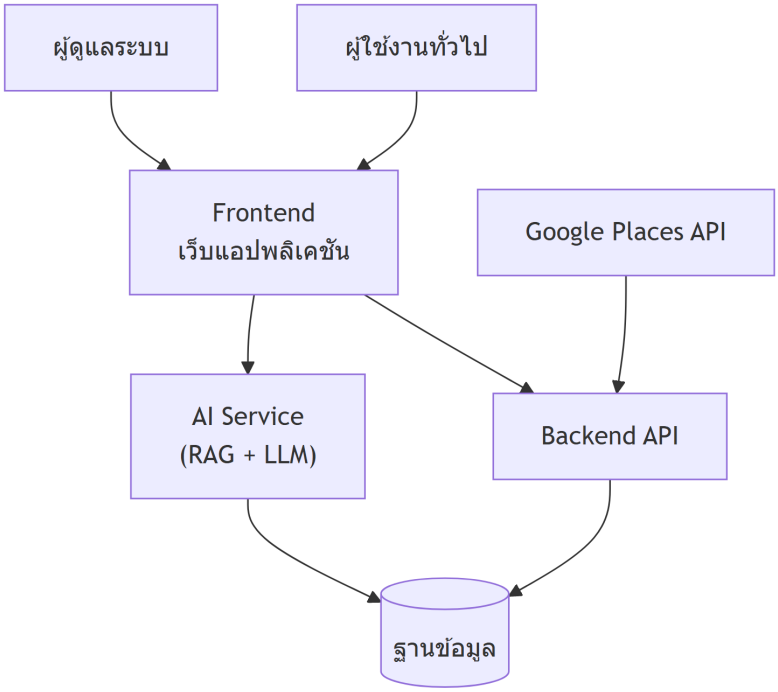
- การทำความสะอาดและจัดรูปแบบข้อมูล (Data Preprocessing): ข้อมูลดิบที่รวบรวมมาจะถูกนำมาแปลงให้อยู่ในรูปแบบมาตรฐานของระบบ (Normalizing) เช่น การรวมประเภทสถานที่ย่อยจาก Google ให้อยู่ในหมวดหมู่ภาษาไทยที่ระบบใช้งาน (เช่น ร้านอาหาร คาเฟ่ วัด พิพิธภัณฑ์ สวนสาธารณะ ตลาด ที่พัก) และการตรวจสอบความสมบูรณ์ของข้อมูลก่อนบันทึกลงฐานข้อมูล

- การแบ่งส่วนข้อมูลและการสร้างเวกเตอร์ (Chunking and Vector Embeddings): สำหรับการใช้งานร่วมกับ RAG ข้อมูลแต่ละรายการ (สถานที่ กิจกรรม และเนื้อหาในฐานความรู้) จะถูกแปลงเป็นเวกเตอร์ฝังตัว (Vector Embeddings) ด้วยโมเดล paraphrase-multilingual-MiniLM-L12-v2 (384 มิติ) ซึ่งประมวลผลบนเครื่องคอมพิวเตอร์ในเครื่อง (Local Inference) ผ่านไลบรารี sentence-transformers โดยจะมีการคำนวณเวกเตอร์ใหม่ทุกครั้งที่มีการนำเข้าหรือแก้ไขข้อมูล
- การจัดเก็บในฐานข้อมูลเวกเตอร์: เวกเตอร์ที่สร้างขึ้นจะถูกจัดเก็บไว้ในฐานข้อมูล PostgreSQL ตัวเดียวกับที่จัดเก็บข้อมูลหลักของระบบ (Supabase) โดยอาศัยส่วนขยาย pgvector ในการจัดเก็บและคำนวณค่าความคล้ายคลึงเชิงมุม (Cosine Similarity) การเลือกใช้ฐานข้อมูลเดียวกันแทนการติดตั้งฐานข้อมูลเวกเตอร์แยกต่างหาก ช่วยลดความซับซ้อนของระบบในระยะต้นแบบ และยังคงประสิทธิภาพการค้นคืนที่เพียงพอสำหรับขนาดข้อมูลของกรณีศึกษานี้

3.1.2 วิเคราะห์และออกแบบระบบ

การพัฒนาระบบแบ่งออกเป็น 3 องค์ประกอบที่พัฒนาและรันแยกจากกัน (Independent Services) ได้แก่ ส่วนเว็บแอปพลิเคชัน (Frontend) ส่วนบริการข้อมูลหลัก (Backend API) และส่วนบริการปัญญาประดิษฐ์ (AI Service) ซึ่งครอบคลุมทั้งระบบ Chatbot และระบบวางแผนการเดินทาง

(1) **การออกแบบสถาปัตยกรรมระบบ:** ระบบที่พัฒนาในงานวิจัยนี้ใช้แนวทางสถาปัตยกรรมแบบแยกส่วนบริการ (Service-Oriented Architecture) ซึ่งประกอบด้วย 3 บริการหลักที่เป็นอิสระต่อกัน คือ Frontend Service พัฒนาด้วย React 18 ร่วมกับ Vite และ react-router-dom ทำหน้าที่เป็นส่วนติดต่อผู้ใช้งานทั้งหมด ทั้งฝั่งผู้ใช้ทั่วไปและฝั่งผู้ดูแลระบบ; Backend API Service พัฒนาด้วย Node.js และ Express ทำหน้าที่ให้บริการ RESTful API สำหรับข้อมูลสถานที่ กิจกรรม ฐานความรู้ และระบบสะสมคะแนน/รางวัล รวมถึงเป็นผู้จัดการโครงสร้างฐานข้อมูล (Schema/Migrations) และกระบวนการนำเข้าข้อมูลจาก Google Places API; และ AI Service พัฒนาด้วยภาษา Python ผ่านเฟรมเวิร์ก FastAPI ทำหน้าที่ให้บริการระบบ Chatbot และระบบวางแผนการเดินทางที่ใช้กลไก RAG และ LLM โดยเชื่อมต่อกับฐานข้อมูลเดียวกันกับ Backend API Service ทั้งสามบริการสื่อสารระหว่างกันผ่าน RESTful API เท่านั้น ในระยะต้นแบบนี้ยังไม่มีองค์ประกอบเสริมสำหรับรองรับการขยายตัวในระดับใช้งานจริง เช่น API Gateway และ Message Broker

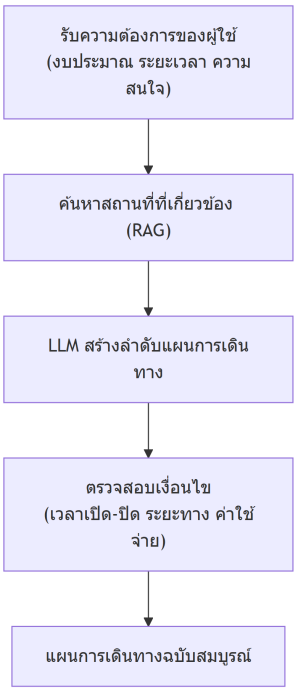


ภาพที่ 11 แผนภาพสถาปัตยกรรมระบบของงานวิจัย

(2) **การออกแบบ UX/UI ของเว็บแอปพลิเคชัน:** กำหนดโฟลว์การใช้งาน (User Flow) ตั้งแต่การเข้าสู่ระบบ การป้อนความต้องการท่องเที่ยว ไปจนถึงการรับแผนการเดินทางที่ระบบสร้างให้ ออกแบบหน้าจอสำคัญ เช่น หน้าแรก หน้ารายละเอียดสถานที่/กิจกรรม หน้าแบบฟอร์มวางแผนทริป และหน้าแสดงผลแผนการเดินทาง โดยยึดหลักการออกแบบที่มีผู้ใช้เป็นศูนย์กลาง (User-Centered Design: UCD)

3.1.3 พัฒนาระบบ

(1) **การพัฒนาระบบวางแผนการเดินทาง (Trip Planning):** ระบบวางแผนการเดินทางใช้กลไก LLM ร่วมกับ RAG และชั้นการคำนวณเชิงกำหนดแน่นอน (Deterministic Computation Layer) ทำงานร่วมกันเป็นกลไกหลักในการสร้างแผนการเดินทาง โดยมีขั้นตอนดังนี้



ภาพที่ 12 แผนภาพขั้นตอนการทำงานของระบบวางแผนการเดินทาง

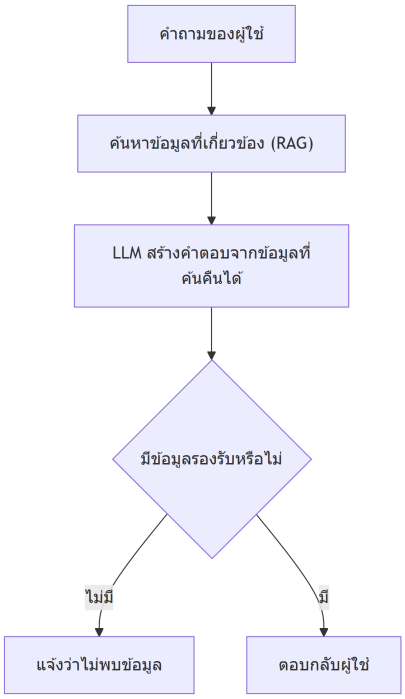
(1) Query Understanding: ระบบรับข้อมูลจากผู้ใช้ ได้แก่ ระยะเวลาการเดินทาง งบประมาณ จังหวะการท่องเที่ยว (ผ่อนคลาย/มาตรฐาน/แน่น) ความสนใจ สถานที่ที่ต้องไปแน่นอน (Must-go) และช่วงเวลาเดินทางในแต่ละวัน

(2) Candidate Retrieval (RAG): ระบบสร้าง Query Embedding จากความสนใจของผู้ใช้ แล้วค้นหาความคล้ายคลึงกับข้อมูลสถานที่ที่จัดเก็บในฐานข้อมูลเวกเตอร์ด้วยเทคนิคการค้นคืนแบบผสม (Hybrid Search) ซึ่งรวมผลจากความคล้ายคลึงเชิงเวกเตอร์ (Cosine Similarity) กับการจับคู่คำสำคัญเชิงตัวอักษร (Trigram/Keyword Matching) ผ่านฟังก์ชันของฐานข้อมูล เพื่อให้ได้รายชื่อสถานที่ผู้สมัคร (Candidate List) ที่สอดคล้องกับความสนใจของผู้ใช้ จากนั้นจึงคัดกรองผู้สมัครตามช่วงราคาที่สอดคล้องกับงบประมาณที่ผู้ใช้ระบุ และจำกัดจำนวนตามความจุของแต่ละวันที่คำนวณจากจังหวะการเดินทาง

(3) Prompt Construction & Generation: ระบบส่งรายชื่อสถานที่ผู้สมัครพร้อมเงื่อนไขของผู้ใช้ให้ LLM สังเคราะห์เป็นลำดับแผนการเดินทางรายวันในรูปแบบ JSON ที่มีโครงสร้างกำหนดไว้ล่วงหน้า โดยกำหนดเป็นเงื่อนไขบังคับว่า LLM ต้องเลือกสถานที่จากรายชื่อผู้สมัครที่ให้ไปเท่านั้น ห้ามสร้างสถานที่ขึ้นใหม่ (Grounded Generation)

(4) Guardrails / Constraint Verification: นอกเหนือจากเงื่อนไขที่กำหนดให้ LLM ปฏิบัติตามแล้ว ระบบมีชั้นตรวจสอบและคำนวณซ้ำแบบกำหนดแน่นอน (Deterministic Backstop) ทำงานอิสระจาก LLM เพื่อรับประกันความถูกต้องของแผนการเดินทาง ได้แก่ การคำนวณระยะทางและเวลาเดินทางระหว่างสถานที่จากพิกัดจริง การตรวจสอบเวลาทำการจริงของแต่ละสถานที่เทียบกับเวลาที่คาดว่าจะไปถึง (และตัดสถานที่ที่ปิดทำการออกจากแผนโดยอัตโนมัติ) การควบคุมมิให้กิจกรรมเกินช่วงเวลาสิ้นสุดที่ผู้ใช้กำหนด การประมาณค่าใช้จ่ายรวม (ค่าน้ำมันตามระยะทางและค่าใช้จ่ายโดยประมาณตามประเภทสถานที่) และการแทรกช่วงเวลาว่างในกรณีที่มีช่องว่างระหว่างกิจกรรมนานเกินเกณฑ์ที่กำหนด กลไกนี้ทำหน้าที่เป็นตัวตรวจทานข้อจำกัดพื้นฐานของแผนการเดินทางก่อนนำเสนอผลลัพธ์แก่ผู้ใช้ และในกรณีที่ LLM ให้ผลลัพธ์ที่ไม่สามารถประมวลผลได้ (เช่น รูปแบบ JSON ผิดพลาด) ระบบจะไม่ใช้ค่าคาดเดาทดแทน แต่จะแจ้งข้อผิดพลาดกลับไปยังผู้ใช้โดยตรง เพื่อป้องกันมิให้ผู้ใช้ได้รับแผนการเดินทางที่ไม่น่าเชื่อถือ

(2) **การพัฒนาระบบ Chatbot และระบบแนะนำ (Chatbot and Recommendation System):**



ภาพที่ 13 แผนภาพขั้นตอนการทำงานของระบบ Chatbot และระบบแนะนำ

(1) Information Retrieval: ระบบรับคำสั่งที่เป็นภาษาธรรมชาติ

(Natural Language Query) จากผู้ใช้ ก่อนการค้นคืนจะมีขั้นตอนเขียนคำถามใหม่ (Query Rewriting) โดยอาศัย LLM แปลงคำถามต่อเนื่อง (Follow-up Question) ที่อ้างอิงบริบทการสนทนาก่อนหน้าให้กลายเป็นประโยคค้นหาที่สมบูรณ์ในตัวเอง จากนั้นแปลงเป็น Query Embedding ด้วยโมเดล paraphrase-multilingual-MiniLM-L12-v2 เพื่อค้นหาความคล้ายคลึงกันแบบผสม (Hybrid Search) ทั้งจากตารางข้อมูลสถานที่และตารางฐานความรู้ทั่วไป (ประวัติศาสตร์ วัฒนธรรม การเดินทาง) แยกจากกัน

(2) Response Generation: ระบบใช้ LLM (gemini-2.5-flash ผ่านช่องทาง API ที่รองรับมาตรฐาน OpenAI-compatible) สังเคราะห์คำตอบที่เป็นข้อความพร้อมคำอธิบายที่สอดคล้องกับบริบทที่ค้นคืนได้เท่านั้น โดยกำหนดให้ LLM ระบุแท็กกำกับประเภทคำตอบไว้ต้นข้อความทุกครั้ง (อ้างอิงจากข้อมูลสถานที่เพียงอย่างเดียว อ้างอิงจากฐานความรู้เพียงอย่างเดียว อ้างอิงทั้งสองส่วน หรือไม่พบข้อมูลที่เกี่ยวข้องเลย) เพื่อให้ระบบตรวจจับกรณีที่ไม่พบข้อมูลตรงกับคำถามได้อย่างแน่นอน (Deterministic Hallucination Guard) และแสดงข้อความแจ้งเตือนมาตรฐานแทนคำตอบที่ไม่มีหลักฐานรองรับ คำตอบของระบบถูกส่งกลับแบบทยอยแสดงผลทีละหน่วยคำ (Token-by-token Streaming) ผ่าน Server-Sent Events (SSE) เพื่อให้ผู้ใช้เห็นความคืบหน้าของคำตอบระหว่างที่ LLM กำลังประมวลผล

(3) การจัดอันดับสถานที่ (Recommendation): รายชื่อสถานที่ที่ได้จากการค้นคืนแบบผสมจะถูกจัดอันดับตามคะแนนความสอดคล้องที่คำนวณได้ ก่อนนำไปใช้เป็นลำดับการนำเสนอสถานที่ในคำตอบของ Chatbot และเป็นรายชื่อสถานที่ผู้สมัครที่ส่งต่อให้ระบบวางแผนการเดินทางคัดกรองต่อตามงบประมาณและความจุของแต่ละวัน

(3) **การพัฒนา Frontend:** พัฒนาด้วยเฟรมเวิร์ก React ร่วมกับเครื่องมือ Vite เพื่อสร้างส่วนติดต่อผู้ใช้ที่โต้ตอบได้ (Interactive UI) และรองรับการใช้งานข้ามอุปกรณ์ (Responsive Web Design) ทั้งบน Desktop, Tablet และ Mobile

(4) **การพัฒนา Backend และการเชื่อมต่อกับบริการต่าง ๆ:** ออกแบบ RESTful API สำหรับดึงข้อมูลจากบริการวางแผนการเดินทาง ระบบแนะนำ และข้อมูลกิจกรรม/สถานที่ พัฒนาหน้าจอสำหรับผู้ดูแลระบบ (Admin Dashboard) เพื่อจัดการข้อมูลสถานที่ กิจกรรม ฐานความรู้ และรหัส QR สำหรับระบบสะสมคะแนน

(5) **การทดสอบด้านการใช้งาน (Usability Testing):** ทดสอบกับกลุ่มผู้ใช้เป้าหมาย (นักศึกษา/นักท่องเที่ยวในจังหวัดขอนแก่น) โดยใช้เกณฑ์ประเมินด้านความง่ายในการใช้งาน (Ease of Use) ความพึงพอใจ (Satisfaction) และความเข้าใจในการนำเสนอข้อมูล

3.1.4 ทดสอบและประเมินผล

(1) **การทดสอบประสิทธิภาพของระบบวางแผนการเดินทาง:** วัดผลลัพธ์ของระบบวางแผนการเดินทางด้วยตัวชี้วัดหลายด้าน (Performance Metrics) ได้แก่ เวลาตอบสนอง (Response Time) คือระยะเวลารวมที่ระบบใช้ตั้งแต่รับคำขอจนถึงส่งแผนการเดินทางสำเร็จให้ผู้ใช้ แยกพิจารณาเป็นเวลาในขั้นตอนการค้นคืนข้อมูล (Retrieval) และขั้นตอนการสังเคราะห์แผนของ LLM; ความแม่นยำในการตอบสนองความต้องการของผู้ใช้ (Accuracy in Meeting User Preferences) คือสัดส่วนของความต้องการที่ผู้ใช้ระบุ (เช่น งบประมาณ สถานที่ที่ต้องไปแน่นอน ความสนใจ) ที่ปรากฏอยู่ในแผนการเดินทางที่ระบบสร้างจริง; และตัวชี้วัด Pass Rate ที่ปรับจากกรอบแนวคิด TravelPlanner ให้สอดคล้องกับขอบเขตของระบบ แบ่งเป็น 3 ตัวชี้วัดเชิงปริมาณ (%) คือ Environment Constraint Pass Rate (อัตราส่วนสถานที่/ร้านอาหารในแผนการเดินทางที่มีอยู่จริงในฐานข้อมูลของระบบ ซึ่งเป็นผลจากกลไก Grounded Generation โดยตรง จึงใช้ยืนยันความถูกต้องของกลไกป้องกัน Hallucination), Commonsense Constraint Pass Rate (อัตราส่วนแผนงานที่มีความสมเหตุสมผลเชิงเวลาและการเดินทาง เช่น ไม่จัดกิจกรรมซ้อนเวลา ไม่ไปสถานที่ที่ปิดทำการ ระยะเวลาเดินทางระหว่างจุดสอดคล้องกับระยะทางจริง) และ Hard Constraint Pass Rate (อัตราส่วนแผนงานที่บรรลุเงื่อนไขที่ผู้ใช้กำหนดครบถ้วนทั้งงบประมาณ ระยะเวลา และสถานที่ที่ต้องไปแน่นอน)

(2) **การประเมินผลโมเดล RAG/Chatbot:** ประเมินความเกี่ยวข้องของคำตอบ (Answer Relevance) ว่าคำตอบที่สร้างโดย LLM ตอบคำถามของผู้ใช้ได้ดีเพียงใด โดยใช้ LLM อีกตัวหนึ่งที่แยกอิสระจากโมเดลที่ใช้งานจริงในระบบเป็นผู้ประเมิน (LLM-as-a-Judge) ให้คะแนนในระดับ 0–100 เพื่อลดความลำเอียงจากการใช้โมเดลเดียวกันทั้งสร้างคำตอบและประเมินผล; ประเมินความน่าเชื่อถือ (Faithfulness) โดยนับจำนวนคำตอบที่หลุดออกนอกบริบทที่ค้นคืนได้ (Out-of-Context) ซึ่งบ่งชี้ถึงการเกิด Hallucination ของโมเดล โดยเปรียบเทียบผลก่อนและหลังการเปิดใช้งานกลไกแท็กกำกับคำตอบเพื่อยืนยันประสิทธิภาพของกลไกดังกล่าว; และประเมิน Context Recall / Precision ของขั้นตอนการค้นคืนข้อมูล (Retrieval) ว่าสามารถดึงข้อมูลสถานที่/ฐานความรู้ที่เกี่ยวข้องกับคำถามจริงออกมาได้ครบถ้วนและแม่นยำเพียงใด โดยเปรียบเทียบผลระหว่างการค้นคืนด้วยความคล้ายคลึงเชิงเวกเตอร์อย่างเดียวกับการค้นคืนแบบผสม (Hybrid Search) ที่ระบบใช้งานจริง

เพื่อยืนยันความคุ้มค่าของการออกแบบดังกล่าว

(3) **การประเมินความพึงพอใจของผู้ใช้:** ใช้แบบสอบถามเพื่อประเมินความรู้สึกและทัศนคติของผู้ใช้เมื่อมีปฏิสัมพันธ์กับระบบ AI โดยวัดความพึงพอใจโดยรวม ความรู้สึกด้านบวกหรือด้านลบ และการยอมรับทางเทคโนโลยี สำหรับ Chatbot จะประเมินความพึงพอใจของผู้ใช้ (User Satisfaction) ร่วมกับค่า Precision, Recall และ F-score ของความสามารถในการตอบคำถามให้ตรงประเด็นเพิ่มเติม

3.1.5 ปรับปรุงและสรุปผล

(1) **การวิเคราะห์ผลลัพธ์ของโมเดล:** เปรียบเทียบผลการประเมินที่ได้จากหัวข้อการทดสอบและประเมินผล โดยเน้นการวิเคราะห์เชิงลึก 2 ส่วน คือ Trip Planning วิเคราะห์ว่าการผสานกลไก LLM เข้ากับชั้นการคำนวณเชิงกำหนดแน่นอน (Deterministic Backstop) ภายใต้สถาปัตยกรรมแบบแยกส่วนบริการสามารถยกระดับคุณภาพของแผนการเดินทาง (Itinerary Quality) ได้อย่างมีประสิทธิภาพตามเป้าหมายที่ตั้งไว้หรือไม่ เมื่อเทียบกับแนวทางที่ใช้ LLM สร้างแผนโดยไม่มีชั้นตรวจสอบเพิ่มเติม; และ Chatbot วิเคราะห์ว่ากลไก RAG ร่วมกับการค้นคืนแบบผสม (Hybrid Search) ช่วยให้ระบบให้คำแนะนำที่ถูกต้องและมีความเกี่ยวข้องของคำตอบสูงขึ้นโดยไม่เพิ่มอัตราการเกิด Hallucination หรือไม่ เมื่อเทียบกับแนวทางค้นคืนด้วยความคล้ายคลึงเชิงเวกเตอร์เพียงอย่างเดียว

(2) **การวิเคราะห์เชิงพฤติกรรมและการยอมรับของผู้ใช้:** วิเคราะห์ข้อมูลความพึงพอใจของผู้ใช้ โดยพิจารณาว่าผู้ใช้รับรู้ถึงประโยชน์หลักของการใช้ AI อย่างไร (เช่น การเข้าถึงข้อมูลที่รวดเร็วขึ้น การได้รับแผนการเดินทางที่ตรงความต้องการโดยไม่ต้องค้นหาข้อมูลเอง เวลาที่ใช้ในการวางแผนสั้นลง) และระบุข้อกังวลหลักของผู้ใช้ (เช่น ความถูกต้องของข้อมูลที่ AI ให้ ความเป็นส่วนตัวและความปลอดภัยของข้อมูล การพึ่งพาเทคโนโลยีสูง) ซึ่งข้อมูลเหล่านี้สามารถนำไปสู่การปรับปรุงความโปร่งใสของระบบ (เช่น การแสดงแหล่งที่มาของคำแนะนำ) และการสร้างความเชื่อมั่นในระบบ

(3) **การสรุปผลและข้อเสนอแนะ:** สรุปผลลัพธ์หลักที่แสดงให้เห็นถึงความสำเร็จของการบูรณาการ LLM และเทคนิค RAG เข้ากับสถาปัตยกรรมแบบแยกส่วนบริการในการวางแผนการเดินทางและการแนะนำการท่องเที่ยว พร้อมระบุข้อจำกัดที่พบในงานวิจัย ได้แก่ (1) ปัญหา Cold-start สำหรับผู้ใช้ใหม่ที่ยังไม่มีประวัติความสนใจ (2) ขนาดชุดข้อมูลที่จำกัดเฉพาะพื้นที่จังหวัดขอนแก่น (3) ระบบยืนยันตัวตนที่ยังอยู่ในรูปแบบจำลองและยังไม่รองรับการใช้งานจริงเต็มรูปแบบ และ (4) สถาปัตยกรรมระบบในระยะต้นแบบยังไม่มี API Gateway

และ Message Broker สำหรับรองรับการขยายตัวในระดับการใช้งานจริง จากข้อจำกัดดังกล่าว งานวิจัยนี้เสนอแนวทางสำหรับการพัฒนาต่อยอดในอนาคต ได้แก่ การขยายขอบเขตให้ครอบคลุมตัวชี้วัดด้านความยั่งยืนเชิงปริมาณที่ยังไม่ได้นำมาใช้ในต้นแบบปัจจุบัน (เช่น Carbon Footprint จากการเดินทาง และผลกระทบทางเศรษฐกิจต่อผู้ประกอบการท้องถิ่น) การเสริมสถาปัตยกรรมด้วย API Gateway และ Message Broker เพื่อรองรับการขยายตัวของระบบ การพัฒนาระบบยืนยันตัวตนให้สมบูรณ์ และการพัฒนาไปสู่ระบบแนะนำแบบสนทนาที่ซับซ้อนยิ่งขึ้น (Conversational Recommender System) ที่สามารถปรับแผนการเดินทางแบบโต้ตอบกับผู้ใช้ได้แบบต่อเนื่อง

3.2 เครื่องมือที่ใช้ในการดำเนินงาน

3.2.1 ฮาร์ดแวร์

3.2.2 ซอฟต์แวร์

เครื่องมือด้านซอฟต์แวร์ที่ใช้ในการพัฒนาโครงงานแสดงดังตารางที่ 2

ตารางที่ 2 รายการซอฟต์แวร์ที่ใช้ในการพัฒนา

ลำดับ	ซอฟต์แวร์	วัตถุประสงค์
1	React (Vite)	พัฒนา Frontend Service (Web)
2	Node.js (Express)	พัฒนา Backend API Service หลัก
3	Python (FastAPI)	พัฒนา AI Service สำหรับระบบ Chatbot และระบบวางแผนการเดินทาง (RAG)
4	PostgreSQL (Supabase) + pgvector	จัดเก็บข้อมูลหลักของระบบและเวกเตอร์ฝังตัว (Vector Embeddings) สำหรับ RAG ในฐานข้อมูลเดียวกัน
5	sentence-transformers (paraphrase-multilingual-MiniLM-L12-v2)	สร้างเวกเตอร์ฝังตัวแบบประมวลผลในเครื่อง (Local Inference)
6	gemini-2.5-flash (ผ่าน OpenAI-compatible API)	โมเดลภาษาขนาดใหญ่ (LLM) สำหรับแชตบอตและการวางแผนทริป
7	Google Places API	ดึงข้อมูลตั้งต้น (Data Seeding) ของสถานที่ท่องเที่ยว

3.3 แผนการดำเนินงาน

แผนการดำเนินงานตลอดระยะเวลาโครงงานแสดงดังตารางที่ 3

ตารางที่ 3 แผนการดำเนินงาน (Gantt Chart)

	ปี 2569				
การดำเนินงาน	6	7	8	9	10
1. การศึกษาวรรณกรรมและทฤษฎีที่เกี่ยวข้อง					
2. การ เก็บ รวบรวม และ เตรียมข้อมูล					
3. การออกแบบและพัฒนาระบบ					
4. การ ทดสอบ และ ประเมิน ผลระบบ					
5. การวิเคราะห์และสรุปผล					

บทที่ 4
การวิเคราะห์ ออกแบบและพัฒนาระบบ

การพัฒนาแอปพลิเคชันส่วนกลางสำหรับการท่องเที่ยวในจังหวัดขอนแก่น (Dino) มุ่งเน้นการใช้เทคโนโลยีปัญญาประดิษฐ์เชิงสร้างสรรค์ (Generative AI) เพื่อแก้ปัญหาความซับซ้อนในการวางแผนการเดินทาง โดยเปลี่ยนจากการใช้อัลกอริทึมแบบดั้งเดิมมาเป็นการใช้ LLM ร่วมกับฐานข้อมูลเวกเตอร์ในการประมวลผล ผลการวิเคราะห์และพัฒนาระบบมีรายละเอียดดังต่อไปนี้

4.1 การวิเคราะห์ระบบ

4.1.1 การวิเคราะห์ปัญหาและการทำงานเดิม

(1) ปัญหาการวางแผนการท่องเที่ยว

ปัญหา สำคัญ ใน ปัจจุบัน คือ การ เข้า ถึง ข้อมูล การ ท่อง เที่ยว ที่กระจัดกระจายอยู่ในหลายช่องทาง เช่น เว็บไซต์ หน่วยงานราชการ เพจโซเชียลมีเดีย หรือแอปพลิเคชันทั่วไป นักท่องเที่ยวจำเป็นต้องใช้เวลาค้นหาข้อมูลจากหลายแหล่ง ซึ่งมักพบปัญหาว่าข้อมูลไม่ครบถ้วน ไม่ทันสมัย นอกจากนี้ แหล่งข้อมูลที่มีอยู่ในปัจจุบันมิได้ถูกรวบรวมอย่างเป็นระบบ ทำให้นักท่องเที่ยวประสบความยากลำบากในการเข้าถึงข้อมูลเพื่อใช้ในการวางแผนการเดินทางให้สอดคล้องกับความสนใจและงบประมาณของตนเอง

(2) การทำงานเดิม

กระบวนการทำงานแบบดั้งเดิม (Manual Process) กำหนดให้นักท่องเที่ยวต้องดำเนินการสืบค้นและประมวลผลข้อมูลด้วยตนเองผ่านขั้นตอนที่มีความซับซ้อนและใช้เวลามาก ดังนี้

(1) การสืบค้นข้อมูล นักท่องเที่ยวทำการค้นหาข้อมูลสถานที่ท่องเที่ยว ที่พัก และร้านอาหาร ผ่านแพลตฟอร์มต่าง ๆ ที่แยกส่วนกันและไม่มีการเชื่อมโยงข้อมูลอย่างเป็นระบบ

(2) การตรวจสอบเส้นทางและระยะเวลา นักท่องเที่ยวต้องนำรายการสถานที่ที่สนใจไปสืบค้นต่อในระบบภูมิสารสนเทศหรือแอปพลิเคชันแผนที่ เพื่อคำนวณระยะทางและประเมินระยะเวลาในการเดินทาง

(3) การจัดตารางเวลาและงบประมาณ นักท่องเที่ยวต้องทำการคำนวณค่าใช้จ่ายและจัดลำดับแผนการเดินทางด้วยตนเอง (Manual Scheduling) ซึ่งมีโอกาสเกิดข้อผิดพลาดและสร้างความยุ่งยากในเชิงปฏิบัติ

4.1.2 ความต้องการของระบบ

การวิเคราะห์ความต้องการของระบบถูกแบ่งออกเป็น 2 ส่วนหลัก ได้แก่ ความต้องการเชิงหน้าที่ที่ระบบต้องตอบสนองต่อผู้ใช้งาน และความต้องการที่ไม่ใช่เชิงหน้าที่ซึ่งเป็นตัวกำหนดมาตรฐานและคุณภาพของระบบ

(1) ความต้องการเชิงฟังก์ชัน

ความ ต้องการ ใน ส่วน นี้ ระบุ ถึง ฟังก์ชัน การ ทำงาน หลัก ที่ ระบบแอปพลิเคชันศูนย์กลางการท่องเที่ยวต้องสามารถดำเนินการได้ เพื่อรองรับการใช้งานของนักท่องเที่ยวและผู้ดูแลระบบ แสดงดังตารางที่ 4

ตารางที่ 4 ความต้องการเชิงฟังก์ชัน (Functional Requirements)

รหัส	ชื่อความต้องการ	รายละเอียดความต้องการ
FR-01	ระบบจัดการและแสดงผลข้อมูลการท่องเที่ยว	ระบบต้องสามารถแสดงรายละเอียดของสถานที่ท่องเที่ยวและกิจกรรม และอนุญาตให้นักท่องเที่ยวค้นหาข้อมูลตามหมวดหมู่ได้
FR-02	ระบบ สร้าง แผนการ เดินทางอัตโนมัติ	ระบบ ต้อง รับ ข้อมูล เงื่อนไข จาก ผู้ใช้ ได้แก่ วัน เวลา สถานที่ ความสนใจ งบประมาณ ความเร็วของทริป และประมวลผลเพื่อสร้างแผนการเดินทาง (Trip Planning) ที่จัดเรียงลำดับสถานที่และเวลาได้อย่างเหมาะสมแบบอัตโนมัติ
FR-03	ระบบ ผู้ ช่วย เสมือนอัจฉริยะ (Chatbot)	ระบบต้องมีแชตบอตที่ขับเคลื่อนด้วย LLM เพื่อรับคำสั่ง วิเคราะห์บริบท และโต้ตอบให้คำแนะนำด้านการท่องเที่ยวกับผู้ใช้งานผ่านทางเว็บไซต์
FR-04	ระบบจัดการส่วนหลังบ้าน (Back-office)	ระบบต้องมีส่วนให้ผู้ดูแลระบบเข้าสู่ระบบเพื่อดำเนินการจัดการฐานข้อมูล (CRUD) สำหรับเพิ่ม ลบ แก้ไข และอัปเดตข้อมูลสถานที่ท่องเที่ยวและกิจกรรมได้

(2) ความต้องการที่ไม่ใช่เชิงฟังก์ชัน

ความ ต้องการ ส่วน นี้ ครอบคลุม ถึง ข้อ จำกัด ทาง สถาปัตยกรรม และคุณสมบัติด้านประสิทธิภาพของระบบ เพื่อให้ระบบปัญญาประดิษฐ์และซอฟต์แวร์ทำงานได้อย่างมีเสถียรภาพ แสดงดังตารางที่ 5

ตารางที่ 5 ความต้องการที่ไม่ใช่เชิงฟังก์ชัน (Non-Functional Requirements)

รหัส	ชื่อความต้องการ	รายละเอียดความต้องการ
NFR-01	สถาปัตยกรรม และ การรองรับการขยายตัว	ระบบ ต้อง ได้ รับ การ ออกแบบ ด้วย สถาปัตยกรรมไมโครเซอร์วิส เพื่อให้แต่ละส่วนทำงาน แยก กัน อย่าง อิสระ และ รองรับ การขยายตัว (Scalability) เมื่อมีผู้ใช้งานเพิ่มขึ้น
NFR-02	ความถูกต้องและความน่าเชื่อถือของ AI	ระบบ LLM ต้องมีกลไกลดการบิดเบือนข้อมูล (Hallucination) โดย ทำงาน ร่วม กับ ระบบสืบค้นข้อมูล เพื่อบังคับให้อ้างอิงคำตอบจากฐานข้อมูลของระบบเป็นหลัก
NFR-03	ประสิทธิภาพ ใน การ ตอบสนอง (Performance)	ระบบ ต้อง มี ความหน่วง เวลา (Latency) ในการประมวลผลและส่งกลับคำตอบจากแชตบอตในระดับที่ต่ำ เพื่อให้การสนทนากับผู้ใช้เป็นไปอย่างราบรื่น
NFR-04	ความ ปลอดภัย ของ ข้อมูล (Security)	ระบบ จัดการ หลัง บ้าน ต้อง มี การ ยืนยัน ตัว ตน (Authentication) และการกำหนดสิทธิ์ (Authorization) ที่รัดกุม เพื่อป้องกันการเข้าถึงหรือแก้ไขฐานข้อมูลโดยไม่ได้รับอนุญาต

4.1.3 Use Case Diagram และรายละเอียด

แทรกภาพ Tourist Use Case Diagram ที่นี่ (ต้นฉบับใน Research Me (1).docx)
ใช้คำสั่ง `\includegraphics`

ภาพที่ 14 Tourist Use Case Diagram

นัก ท่อง เที่ยว (Tourist) เป็น ผู้ใช้ งาน ระดับ ทั่วไป (End-User) ของ ระบบสารสนเทศ ด้าน การ ท่อง เที่ยว ซึ่ง มี บทบาท สำคัญ ใน การ เข้า มา ปฏิสัมพันธ์ กับ ระบบ เพื่อ สืบค้นข้อมูลและวางแผนการเดินทางตามความต้องการของตนเอง ผู้ใช้งานสามารถเรียกดูรายละเอียด

ข้อมูลกิจกรรมและสถานที่ท่องเที่ยวในเชิงลึก ค้นหาข้อมูลตามเงื่อนไขหรือความสนใจ สร้างแผนการเดินทาง (Trip Planning) ด้วยตนเอง และโต้ตอบกับระบบแชทบอทเพื่อสอบถามข้อมูลหรือขอคำแนะนำเพิ่มเติมได้แบบเรียลไทม์ รายละเอียดเชิงลึกของแต่ละฟังก์ชันแสดงในตารางที่ 6–11

ตารางที่ 6 Use Case UC-01: ถามตอบข้อมูลโดยใช้ Chatbot

Use Case ID	UC-01
Actor	นักท่องเที่ยว (Tourist), Google Gemini API
คำอธิบาย	นักท่องเที่ยวสามารถโต้ตอบเพื่อสอบถามข้อมูลเกี่ยวกับการท่องเที่ยว สถานที่ กิจกรรมต่าง ๆ หรือฐานความรู้ที่ระบบมี ผ่านอินเทอร์เฟซแชทบอท โดยระบบจะส่งข้อความไปประมวลผลที่ Google Gemini API เพื่อสร้างคำตอบที่ถูกต้องและเป็นธรรมชาติ
เงื่อนไขเบื้องต้น	ระบบเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต และ Google Gemini API อยู่ในสถานะพร้อมใช้งาน
กระแสหลัก (Main Flow)	1. นักท่องเที่ยวพิมพ์คำถามหรือข้อความลงในช่องแชทและกดส่ง 2. ระบบรับข้อความและส่งต่อ (API Request) ไปยัง Google Gemini API โดยผ่าน RAG service เพื่อให้ได้ข้อมูลที่ตรงกับฐานข้อมูลที่มีอยู่ 3. Google Gemini API ประมวลผลภาษาธรรมชาติและส่งคำตอบกลับมา (API Response) 4. ระบบแสดงผลคำตอบบนหน้าจอแชทบอทให้นักท่องเที่ยวรับทราบ
กระแสทางเลือก	3a. หากไม่สามารถเชื่อมต่อ Google Gemini API ได้ (เช่น Timeout) ระบบจะแสดงข้อความขออภัยและให้ผู้ใช้งานลองใหม่อีกครั้งในภายหลัง
เงื่อนไขภายหลัง	ข้อมูลการโต้ตอบถูกแสดงผลบนหน้าจอ แสดงประวัติการสนทนา และระบบพร้อมรับคำถามถัดไป

ตารางที่ 7 Use Case UC-02: ค้นหาข้อมูลสถานที่ท่องเที่ยว กิจกรรม และอีเวนท์

Use Case ID	UC-02
Actor	นักท่องเที่ยว (Tourist)
คำอธิบาย	นักท่องเที่ยวสามารถสืบค้นข้อมูลด้านการท่องเที่ยวภายในระบบ โดยระบุคำค้นหาหรือตัวกรองที่ต้องการ เพื่อดูรายการสถานที่ กิจกรรม หรืออีเวนท์ที่สอดคล้องกับความสนใจ
เงื่อนไขเบื้องต้น	นักท่องเที่ยวอยู่ในหน้าแรก หรือหน้าสำหรับค้นหาข้อมูลของระบบ
กระแสหลัก (Main Flow)	1. นักท่องเที่ยวระบุคำค้นหา หรือเลือกหมวดหมู่ที่ต้องการ (สถานที่, อีเวนท์) 2. ระบบตรวจสอบเงื่อนไขและดึงข้อมูลจากฐานข้อมูล 3. ระบบแสดงรายการผลลัพธ์การค้นหาที่ตรงตามเงื่อนไข
กระแสทางเลือก	2a. หากไม่มีข้อมูลที่ตรงกับเงื่อนไขการค้นหา ระบบจะแสดงข้อความแจ้งเตือนว่า “ไม่พบข้อมูลที่ค้นหา”
เงื่อนไขภายหลัง	ระบบแสดงรายการผลการค้นหาบนหน้าจอ
Extend	UC-04 ดูรายละเอียดข้อมูลสถานที่, UC-05 ดูรายละเอียดข้อมูลอีเวนท์

ตารางที่ 8 Use Case UC-04: ดูรายละเอียดข้อมูลสถานที่ (Place Detail)

Use Case ID	UC-04
Actor	นักท่องเที่ยว (Tourist)
คำอธิบาย	เป็นฟังก์ชันส่วนขยาย (Extension) เมื่อนักท่องเที่ยวต้องการดูข้อมูลเชิงลึกของสถานที่ที่สนใจ ระบบจะแสดงข้อมูลครบถ้วน ได้แก่ ชื่อสถานที่, รูปภาพ (Gallery), คะแนนและรีวิวจากผู้เยี่ยมชม, ที่อยู่, เวลาทำการ, ข้อมูลติดต่อ, สิ่งอำนวยความสะดวก, ช่วงราคา, แผนที่, วิธีชำระเงิน, ที่จอดรถ, การเข้าถึงสำหรับผู้พิการ และสถานีชาร์จรถยนต์ไฟฟ้า (EV)
เงื่อนไขเบื้องต้น	ระบบกำลังแสดงรายการผลการค้นหาสถานที่อยู่ และนักท่องเที่ยวได้เลือกสถานที่ที่สนใจซึ่งมี Place ID พร้อมส่งต่อมาใน URL Parameter
กระแสหลัก (Main Flow)	1. นักท่องเที่ยวคลิกเลือกรายการสถานที่ที่สนใจจากผลการค้นหา 2. ระบบแสดงหน้า Loading เพื่อรอแสดงผล 3. เมื่อได้รับข้อมูลสำเร็จ ระบบแสดงหน้ารายละเอียดอย่างครบถ้วน (Hero Section, Gallery, ชื่อสถานที่, คะแนนดาว, ประเภท, ที่อยู่, สถานะเปิด/ปิด, คำอธิบาย, สิ่งอำนวยความสะดวก, รีวิว, ข้อมูลติดต่อ, แผนที่, เวลาทำการ, วิธีชำระเงิน, ที่จอดรถ, การเข้าถึงสำหรับผู้พิการ, สถานีชาร์จ EV และปุ่ม Quick Actions)
กระแสทางเลือก	[A1] เกิดข้อผิดพลาดในการเรียก API: ระบบแสดงหน้าข้อผิดพลาดพร้อมปุ่ม “กลับ” [A2] ไม่พบข้อมูลสถานที่: ระบบแสดงข้อความ “ไม่พบข้อมูลสถานที่” กลางหน้าจอ
เงื่อนไขภายหลัง	นักท่องเที่ยวได้รับทราบข้อมูลรายละเอียดของสถานที่ที่เลือกอย่างครบถ้วน และสามารถเปิดเส้นทางใน Google Maps, โทรติดต่อสถานที่, เขียนรีวิว หรือแชร์สถานที่นี้
Extend	ขยายการทำงานของ UC-02 ค้นหาข้อมูลสถานที่ท่องเที่ยว กิจกรรม และอีเวนท์

ตารางที่ 9 Use Case UC-05: ดูรายละเอียดข้อมูลอีเวนท์ (View Event Detail)

Use Case ID	UC-05
Actor	นักท่องเที่ยว (Tourist)
คำอธิบาย	ฟังก์ชันส่วนขยายของการค้นหาข้อมูล เมื่อนักท่องเที่ยวต้องการดูข้อมูลเชิงลึกของอีเวนท์ (เช่น เทศกาล งานคอนเสิร์ต) ระบบจะแสดงรายละเอียดครบถ้วน ได้แก่ ชื่องาน ประเภท วันเวลาและตารางกิจกรรม สถานที่จัดงานพร้อมแผนที่ ค่าเข้างานและช่องทางซื้อบัตร ข้อมูลผู้จัดงาน รวมถึงความเหมาะสมสำหรับกลุ่มผู้เข้าร่วมประเภทต่าง ๆ
เงื่อนไขเบื้องต้น	ระบบกำลังแสดงรายการผลการค้นหาซึ่งมีรายการประเภท “อีเวนท์” ปรากฏอยู่ และอีเวนท์นั้นมีสถานะเป็น published และไม่ถูกยกเลิก
กระแสหลัก (Main Flow)	1. นักท่องเที่ยวคลิกเลือกรายการอีเวนท์ที่สนใจจากผลการค้นหา 2. ระบบดึงข้อมูลทั้งหมดจากฐานข้อมูล 3. ระบบแสดงหน้าจอรายละเอียดของอีเวนท์ (ชื่องานและภาพปก, ประเภทงานและแท็ก, วันที่จัดงานและตาราง, สถานที่จัดงานพร้อมพิกัดแผนที่, ข้อมูลค่าเข้างานและลิงก์ซื้อบัตร, ข้อมูลผู้จัดงานและช่องทางติดต่อ, กลุ่มที่เหมาะสม) 4. นักท่องเที่ยวอ่านและรับทราบข้อมูลที่ต้องการ
กระแสทางเลือก	[A1] ไม่พบข้อมูลอีเวนท์: ระบบแจ้งว่า “ไม่พบข้อมูลอีเวนท์นี้” และแนะนำให้กลับไปยังหน้าผลการค้นหา [A2] อีเวนท์ถูกยกเลิก: ระบบแสดงรายละเอียดตามปกติ แต่แจ้งเตือนอย่างชัดเจนว่างานนี้ได้ถูกยกเลิกแล้ว
เงื่อนไขภายหลัง	นักท่องเที่ยวได้รับทราบข้อมูลรายละเอียดของอีเวนท์ที่เลือกอย่างครบถ้วน และสามารถกดลิงก์ซื้อบัตรหรือดูแผนที่สถานที่จัดงานได้ทันที
Extend	ขยาย การ ทำงาน ของ UC-02 ค้นหา ข้อมูล สถาน ที่ ท่อง เที่ยว กิจกรรม และอีเวนท์

ตารางที่ 10 Use Case UC-06: สร้างแผนการเดินทาง (Trip Planning)

Use Case ID	UC-06
Actor	นักท่องเที่ยว (Tourist)
คำอธิบาย	นักท่องเที่ยวสามารถจัดการและออกแบบแผนการเดินทางส่วนตัว โดยระบุปัจจัยสำคัญ ได้แก่ วันเดินทาง ที่พัก ความสนใจ งบประมาณ และช่วงเวลาที่ต้องการเดินทางในแต่ละวัน จากนั้นระบบจะใช้ AI ในการวิเคราะห์และสร้างแผนการเดินทางแบบวัน-ต่อ-วัน พร้อมประมาณการระยะทาง เวลา และค่าใช้จ่าย
เงื่อนไขเบื้องต้น	นักท่องเที่ยวเข้าสู่เมนูการสร้างแผนการเดินทาง และระบบพร้อมใช้งาน
กระแสหลัก (Main Flow)	1. นักท่องเที่ยวระบุข้อมูลเบื้องต้นสำหรับทริป (วันเริ่มต้น-สิ้นสุด, ชื่อที่พัก, ความสนใจ, ระดับงบประมาณ, สถานที่ที่ต้องการไป, ช่วงเวลาเดินทางในแต่ละวัน) 2. ระบบแนะนำสถานที่ที่สอดคล้องกับความสนใจของผู้ใช้ 3. ระบบประมวลผลข้อมูลที่ได้รับทั้งหมดผ่าน AI 4. ระบบแสดงแผนการเดินทางแบบวัน-ต่อ-วัน (ตารางเวลา ระยะทาง เวลาเดินทาง ค่าใช้จ่ายโดยประมาณ) 5. นักท่องเที่ยวสามารถโต้ตอบกับแผน เช่น ขยาย/ยุบรายละเอียดแต่ละวัน 6. ระบบบันทึกแผนการเดินทางลงในฐานข้อมูลเพื่อเป็นประวัติการสร้าง
กระแสทางเลือก	2a. หากกรอกข้อมูลไม่ครบถ้วน ระบบแจ้งเตือนรายการที่ขาดหาย 2b. หากวันสิ้นสุดน้อยกว่าวันเริ่มต้น ระบบแจ้งเตือนให้แก้ไข 2c. หากระยะเวลาทริปเกิน 7 วัน ระบบแจ้งเตือนให้ปรับช่วงเวลา 2d. หากระบบ AI ขัดข้อง ระบบแจ้งว่า “ระบบไม่สามารถใช้งานได้ชั่วคราว”
เงื่อนไขภายหลัง	แผนการเดินทางถูกสร้างและแสดงผลสำเร็จ และถูกบันทึกลงในฐานข้อมูลเพื่อให้ผู้ดูแลระบบสามารถดูและจัดการได้
Extend	UC-07 การประเมินแผนการเดินทาง (Trip Evaluation)

ตารางที่ 11 Use Case UC-07: การประเมินแผนการเดินทาง

Use Case ID	UC-07
Actor	นักท่องเที่ยว (Tourist)
คำอธิบาย	ฟังก์ชันส่วนขยายที่ช่วยนักท่องเที่ยวตรวจสอบความสมเหตุสมผลของแผนการเดินทางที่สร้างขึ้น โดยให้ข้อมูล Feedback กลับมาเพื่อให้ระบบสร้างแผนการเดินทางใหม่ที่เหมาะสมกว่าเดิม
เงื่อนไขเบื้องต้น	นักท่องเที่ยวมีข้อมูลแผนการเดินทางอยู่ในระบบจาก UC-06 สร้างแผนการเดินทาง
กระแสหลัก (Main Flow)	1. เมื่อแผนการเดินทางถูกสร้างเสร็จสิ้น ระบบแสดงปุ่ม “ถูกใจ”/“ไม่ถูกใจ” ให้กับแต่ละสถานที่ พร้อมช่องกรอกสิ่งที่ต้องการปรับปรุง 2. เมื่อกรอกเรียบร้อย นักท่องเที่ยวกดปุ่ม “ปรับแก้แผนการเดินทางอีกครั้ง” 3. ระบบแสดงแผนการเดินทางที่ถูกสร้างขึ้นใหม่
กระแสทางเลือก	2a. หากผู้ใช้ไม่กรอก ถือว่าสถานที่นั้นเหมาะสมแล้ว ไม่เปลี่ยนแปลง 2b. หากกรอกข้อความที่ไม่เกี่ยวข้อง ระบบตอบกลับว่า “กรุณากรอกข้อมูลที่เป็นประโยชน์ต่อการแก้ไขแผนการเดินทางเท่านั้น”
เงื่อนไขภายหลัง	นักท่องเที่ยวได้รับแผนการเดินทางที่ปรับปรุงเรียบร้อยตามความต้องการแล้ว
Extend	ขยายการทำงานของ UC-06 สร้างแผนการเดินทาง (Trip Planning)

แทรกภาพ Admin Use Case Diagram ที่นี่ (ต้นฉบับใน Research Me (1).docx)
ใช้คำสั่ง `\includegraphics`

ภาพที่ 15 Admin Use Case Diagram

ผู้ดูแลระบบ (Administrator) เป็นผู้ใช้งานที่มีสิทธิ์ในระดับสูงของระบบ มีบทบาทสำคัญในการบริหารจัดการโครงสร้างข้อมูลและการทำงานเบื้องหลังของระบบ (Back-end Management) การเข้าถึงระบบต้องผ่านกระบวนการยืนยันตัวตน (Authentication) ก่อนดำเนินการในฟังก์ชันต่าง ๆ ซึ่งครอบคลุมการจัดการข้อมูลสถานที่ท่องเที่ยว ข้อมูลกิจกรรม (Event) และฐานความรู้สำหรับระบบแชทบอทแบบ CRUD (Create, Read, Update, Delete) รายละเอียดแสดงในตารางที่ 12–15

ตารางที่ 12 Use Case UC-11: เข้าสู่ระบบ (Login / Authentication)

Use Case ID	UC-11
Actor	ผู้ดูแลระบบ (Administrator)
คำอธิบาย	ผู้ดูแลระบบต้องยืนยันตัวตนก่อนเข้าใช้งานระบบจัดการหลังบ้าน (Admin Portal) ระบบตรวจสอบ Email และ Password และกำหนดสิทธิ์ตาม Role (admin)
เงื่อนไขเบื้องต้น	ผู้ดูแลระบบมีบัญชีผู้ใช้ (role: admin) ในฐานข้อมูล (Collection: User)
กระแสหลัก (Main Flow)	1. ผู้ดูแลระบบเข้าสู่หน้าล็อกอินของระบบ Admin Portal 2. กรอก Email และ Password 3. ระบบตรวจสอบข้อมูลประจำตัวกับฐานข้อมูล 4. ระบบตรวจสอบว่า role ของบัญชีเป็น ’admin’ 5. ระบบสร้าง Session หรือ Token และอนุญาตการเข้าถึง 6. ระบบนำทางผู้ดูแลระบบไปยังหน้า Dashboard
กระแสทางเลือก	3a. หาก Email/Password ไม่ถูกต้อง ระบบแสดงข้อความแจ้งเตือน 4a. หาก role ไม่ใช่ ’admin’ ระบบปฏิเสธการเข้าถึง
เงื่อนไขภายหลัง	ผู้ดูแลระบบเข้าสู่ระบบสำเร็จและสามารถเข้าถึงฟังก์ชันจัดการได้

ตารางที่ 13 Use Case UC-06.1/6.2/6.3: จัดการข้อมูลสถานที่ (Create/Update/Delete Place)

Use Case ID	UC-06.1 สร้าง / UC-06.2 แก้ไข / UC-06.3 ลบ ข้อมูลสถานที่
Actor	ผู้ดูแลระบบ (Administrator)
คำอธิบาย	ผู้ดูแลระบบเพิ่ม/แก้ไข/ลบข้อมูลสถานที่ท่องเที่ยวใน Collection: Place
เงื่อนไขเบื้องต้น	ผู้ดูแลระบบเข้าสู่ระบบสำเร็จแล้ว (UC-11)
กระแสหลัก (Main Flow)	*สร้าง:* กรอกข้อมูลหลัก (core.name, location, address) → ระบบตรวจสอบความซ้ำซ้อนของ google_place_id (Unique) → ระบุข้อมูลเพิ่มเติม (features, extra, media, openingHours) → บันทึกข้อมูล *แก้ไข:* ระบบแสดงข้อมูลเดิมในฟอร์ม → แก้ไขข้อมูล → บันทึกการเปลี่ยนแปลง → ระบบอัปเดตข้อมูลและปรับ Timestamp *ลบ:* เลือกสถานที่และกด “ลบ” → ยืนยันการลบ → ระบบลบข้อมูลออกจาก Collection: Place
กระแสทางเลือก	หาก google_place_id ซ้ำ ระบบแจ้งเตือน “สถานที่นี้มีในระบบแล้ว” หากลบข้อมูลในช่องบังคับขณะแก้ไข ระบบแจ้งเตือนให้กรอกข้อมูล หากสถานที่ถูกเชื่อมโยงอยู่ใน Event หรือ Activity ระบบอาจแสดงคำเตือนก่อนให้ลบ
เงื่อนไขภายหลัง	ข้อมูลสถานที่ถูกเพิ่ม/อัปเดต/ลบในฐานข้อมูลตามคำสั่ง

ตารางที่ 14 Use Case UC-09.1/9.2/9.3: จัดการข้อมูล Event (Create/Update/Delete Event)

Use Case ID	UC-09.1 สร้าง / UC-09.2 แก้ไข / UC-09.3 ลบ ข้อมูล Event
Actor	ผู้ดูแลระบบ (Administrator)
คำอธิบาย	ผู้ดูแลระบบสร้าง/แก้ไข/ลบข้อมูลกิจกรรมที่มีเงื่อนไขทางเวลา (เช่น งานคอนเสิร์ต, เทศกาล) ใน Collection: Event
เงื่อนไขเบื้องต้น	ผู้ดูแลระบบเข้าสู่ระบบสำเร็จแล้ว (UC-11)
กระแสหลัก (Main Flow)	*สร้าง:* กรอกข้อมูลทั่วไป (name, slug, category, description) → กำหนดช่วงเวลาจัดงาน (schedule.startDate, endDate, sessions) → เลือกสถานที่จัดงาน (venues เชื่อมกับ PlaceId) → กำหนดข้อมูลบัตร (admission) และผู้จัดงาน (organizer) → ตั้งสถานะเป็น upcoming และบันทึก *แก้ไข:* แก้ไขรายละเอียด หรือ อัปเดตสถานะ (status/isPublished) → บันทึก *ลบ:* เลือก Event และยืนยันการลบ → ระบบลบ Document ออกจาก Collection: Event
กระแสทางเลือก	หาก startDate ช้ากว่า endDate ระบบแจ้งเตือนให้แก้ไข หาก slug ซ้ำในระบบ ระบบแจ้งเตือนให้ตั้งค่า slug ใหม่
เงื่อนไขภายหลัง	ข้อมูล Event ใหม่/อัปเดต/ถูกลบ พร้อมแสดงผลในแอปพลิเคชันตามคำสั่ง

ตารางที่ 15 Use Case UC-10.1/10.2/10.3: จัดการฐานความรู้ (Create/Update/Delete KnowledgeBase)

Use Case ID	UC-10.1 สร้าง / UC-10.2 แก้ไข / UC-10.3 ลบ ฐานความรู้
Actor	ผู้ดูแลระบบ (Administrator)
คำอธิบาย	ผู้ดูแลระบบเพิ่ม/แก้ไข/ลบชุดข้อมูลความรู้ที่ใช้เป็น Context ให้แชทบอท
เงื่อนไขเบื้องต้น	ผู้ดูแลระบบเข้าสู่ระบบสำเร็จแล้ว (UC-11)
กระแสหลัก (Main Flow)	*สร้าง:* กรอกหัวข้อ (title) และเนื้อหา (content) → เลือกหมวดหมู่ (category เช่น transport, food-culture) → กำหนดสถานะ isActive และ isPinned → บันทึกลง Collection: Knowledge-Base *แก้ไข:* แก้ไข content หรือเปลี่ยนสถานะ isPinned/isActive → ระบบอัปเดตข้อมูล หากมีการเปลี่ยน isPinned ระบบจะจัดการ System Prompt ใหม่ *ลบ:* เลือกฐานความรู้และยืนยันการลบ → ระบบลบข้อมูลออกและอัปเดต System Prompt หากความรู้นั้นเคยถูก Pin ไว้
กระแสทางเลือก	หากเนื้อหาหรือหัวข้อซ้ำกับรายการที่มีอยู่ ระบบแจ้งเตือนให้ตรวจสอบ หากกรอกข้อมูล Required ไม่ครบ ระบบแจ้งเตือน หาก KnowledgeBase มี isPinned=true อยู่ ระบบอาจแจ้งเตือนเป็นพิเศษก่อนลบ
เงื่อนไขภายหลัง	แชทบอทมีชุดข้อมูลใหม่/อัปเดต/ถูกลบสำหรับใช้งานตามคำสั่ง

4.1.4 System Sequence Diagram

(1) สำหรับผู้ใช้งานทั่วไป (นักท่องเที่ยว)

ระบบในส่วนของผู้ใช้งานทั่วไปประกอบด้วยฟังก์ชันหลักที่มีตรรกะการทำงานแยกจากกันอย่างชัดเจน จึงแยกแผนภาพออกตามฟังก์ชันหลัก ดังนี้

แทรกภาพ System Sequence Diagram: Chatbot / ค้นหาข้อมูล / สร้างแผนการเดินทาง

ภาพที่ 16 System Sequence Diagram สำหรับผู้ใช้งานทั่วไป

ฟังก์ชันถามตอบข้อมูลโดยใช้ Chatbot: กระบวนการเริ่มต้นเมื่อนักท่องเที่ยวส่งคำถามหรือข้อความเข้าสู่ระบบ ภายใต้เงื่อนไขทางเลือก 2 กรณี คือ (1) ประมวลผลสำเร็จ ระบบส่งคำตอบกลับไปแสดงผลบนแชทบอท หรือ (2) เกิดข้อผิดพลาด/หมดเวลาตอบสนอง (Timeout) ระบบแสดงข้อความขออภัยและเสนอให้ลองใหม่

ฟังก์ชันค้นหาข้อมูลสถานที่และกิจกรรม: เริ่มต้นเมื่อนักท่องเที่ยวส่งคำสั่งค้นหาพร้อมพารามิเตอร์ (คำค้นหา, หมวดหมู่) หากพบข้อมูลตรงเงื่อนไข ระบบแสดงผลลัพธ์การค้นหา หากไม่พบระบบแจ้งเตือน สำหรับการดูรายละเอียดเชิงลึก (เงื่อนไขทางเลือกเสริม) หากดึงข้อมูลสำเร็จระบบแสดงรายละเอียดครบถ้วน หากผิดพลาดหรือข้อมูลสูญหายระบบแจ้งเตือน

ฟังก์ชันสร้างแผนการเดินทาง: เริ่มต้นเมื่อนักท่องเที่ยวส่งเงื่อนไขทริป (วันที่เดินทาง รูปแบบที่พัก ความสนใจ งบประมาณ) ระบบตรวจสอบความถูกต้องของข้อมูลนำเข้าก่อน (เช่น ระยะเวลาเกิน 7 วัน) จากนั้นตรวจสอบความพร้อมของบริการ AI หากพร้อมใช้งานระบบแสดงผลแผนการเดินทางแบบรายวัน หากไม่พร้อมระบบแจ้งงดให้บริการชั่วคราว นอกจากนี้ยังรองรับกระบวนการประเมินและปรับปรุงแผน (Feedback) โดยตรวจสอบคุณภาพของข้อมูลตอบรับก่อนนำไปประมวลผลร่วมกับ AI เพื่อสร้างแผนฉบับปรับปรุงใหม่

(2) สำหรับผู้ดูแลระบบ (Admin Portal)

ระบบในส่วนของผู้ดูแลระบบประกอบด้วยฟังก์ชันการจัดการข้อมูลพื้นฐาน (CRUD) ซึ่งมีรูปแบบมาตรฐานคล้ายคลึงกัน คือผู้ดูแลระบบต้องผ่านการตรวจสอบสิทธิ์ (Authentication) ก่อน จากนั้นจึงส่งคำสั่งจัดการข้อมูลไปยังระบบและระบบตอบกลับสถานะการดำเนินการ โดยแยกส่วนการจัดการฐานความรู้ (Knowledge Base) ออกมาต่างหากเนื่องจากมีกระบวนการทำงานที่แตกต่างกัน

แทรกภาพ System Sequence Diagram: การเข้าสู่ระบบ/จัดการข้อมูลมาตรฐาน และการจัดการฐานความรู้

ภาพที่ 17 System Sequence Diagram สำหรับผู้ดูแลระบบ

การจัดการข้อมูลแบบมาตรฐาน: ผู้ใช้งานส่งคำขอเข้าสู่ระบบ ระบบตรวจสอบความถูกต้องของข้อมูลบัญชี หากถูกต้องระบบอนุญาตให้เข้าถึงฟังก์ชัน CRUD โดยแต่ละคำสั่งระบบจะติดต่อกับฐานข้อมูลและส่งผลลัพธ์กลับ

การจัดการฐานความรู้: ผู้ดูแลระบบเลือกคำสั่ง (Create/Update/Delete) ระบบแสดงฟอร์มหรือกล่องยืนยัน เมื่อส่งข้อมูลแล้วระบบตรวจสอบภายใต้เงื่อนไขทางเลือก – หากไม่ผ่านการตรวจสอบหรือหัวข้อซ้ำซ้อน ระบบแจ้งข้อผิดพลาด หากผ่านเกณฑ์ ระบบยืนยันความสำเร็จและปรับปรุงบริบทข้อมูล (System Prompt) ของ AI โดยอัตโนมัติ

4.1.5 แผนภาพกิจกรรม (Activity Diagram)

ก่อนอธิบายการรับส่งข้อมูลระหว่าง Component ย่อย โครงงานนี้ได้จำลองตรรกะการตัดสินใจของอัลกอริทึมในการสร้างแผนการเดินทาง (Trip Planning Algorithm) เพื่อแสดงกระบวนการทำงานแบบเป็นลำดับขั้นตอน ตั้งแต่การรับเงื่อนไขจากผู้ใช้ การประมวลผล RAG การส่งคำสั่งให้ LLM ไปจนถึงการจัดการข้อมูลหลังการประมวลผล (Post-processing)

แทรกภาพ Activity Diagram: ตรรกะการประมวลผลเพื่อสร้างแผนการเดินทาง

ภาพที่ 18 Activity Diagram ตรรกะการประมวลผลเพื่อสร้างแผนการเดินทาง

แผนภาพใช้รูปแบบการแบ่งช่อง (Swimlane) แสดงขอบเขตความรับผิดชอบของ 4 องค์ประกอบหลัก ได้แก่ ผู้ใช้งาน (User/Client), ระบบหลังบ้าน (Backend: TripBuilder Service), ฐานข้อมูลและการดึงข้อมูล (Database & RAG), และโมเดลภาษาขนาดใหญ่ (LLM) โดยมีขั้นตอนดังนี้ (1) ผู้ใช้งานส่งเงื่อนไขการเดินทาง (TripInput) เข้าสู่ระบบ (2) ระบบหลังบ้านคำนวณช่องว่าง (Slot) ของเวลาในทริป ค้นหาโรงแรม สถานที่บังคับ (Must-Go) และใช้ RAG ค้นหาสถานที่เพิ่มเติมตามความสนใจ พร้อมกรองตามระดับงบประมาณ (3) ระบบสร้าง Dynamic Prompt จากความเร็วทริป (Pace) และงบประมาณ ส่งให้ LLM ประมวลผลและจัดเรียงตารางเวลาเบื้องต้น คืนผลลัพธ์เป็น JSON (4) ระบบหลังบ้านแปลงข้อมูล JSON และวนลูปตรวจสอบทีละสถานที่ในแต่ละวัน คำนวณระยะทางและเวลาเดินทางจริง ตรวจสอบเวลาเปิด-ปิด (หากต้องรอเปิดนานกว่า 45 นาที แทรกกิจกรรม “พักผ่อนตามอัธยาศัย” หากรอน้อยกว่ากำหนด สถานะ “Waiting”) คำนวณเวลาแวะพักตามลักษณะทริปและค่าน้ำมัน (5) ระบบวนลูปจนครบทุกสถานที่ สรุปภาพรวมและค่าใช้จ่ายทั้งหมด พร้อมส่งผลลัพธ์ (TripResponse) กลับไปแสดงผลให้ผู้ใช้งานทราบ

4.2 การออกแบบระบบ

4.2.1 สถาปัตยกรรมของระบบ

แนวคิดการออกแบบระบบถูกพัฒนาภายใต้สถาปัตยกรรมไมโครเซอร์วิส เพื่อแยกส่วนการทำงานออกจากกันอย่างชัดเจน ทั้งในเชิงตรรกะ (Logical) และเชิงกายภาพ (Physical) ทำให้แอปพลิเคชันสามารถรองรับการขยายตัว (Scalability) ได้ดี

แทรกภาพ System Architecture Diagram

ภาพที่ 19 สถาปัตยกรรมของระบบ

4.2.1. **ส่วนติดต่อผู้ใช้งาน (Frontend Component):** พัฒนาด้วย React แยกเป็น 2 แพลตฟอร์ม คือ Mobile Application สำหรับผู้ใช้งานทั่วไป (USER) และ Web Application สำหรับผู้ดูแลระบบ (ADMIN)

4.2.2. **ส่วนการจัดการระบบหลัก (Core Backend Component):** ใช้ Node.js (Express) ทำหน้าที่ควบคุมลอจิกพื้นฐาน บริหารจัดการฐานข้อมูลหลัก (MongoDB/Firebase) และเป็นตัวกลางเชื่อมต่อ

(API Gateway ภายใน) เพื่อส่งต่อคำสั่งที่ซับซ้อนไปยังฝั่ง AI รวมถึงจัดการ Data Pipeline ดึงข้อมูลจาก Google Places API

4.2.3. **ส่วนประมวลผลปัญญาประดิษฐ์ (AI & Algorithm Component):** พัฒนาด้วย Python (FastAPI) ทำหน้าที่รับคำสั่งจาก Node.js เพื่อกระจายงานไปยังโมเดล LLM (ผ่าน Ollama) และโต้ตอบกับ Vector Database (Chroma) สำหรับระบบ RAG และการวางแผนการเดินทาง

4.2.2 Sequence Diagram

แทรกภาพ Detailed Sequence Diagram: การสร้างแผนการเดินทางด้วย RAG/LLM และ Chatbot

ภาพที่ 20 Detailed Sequence Diagram สำหรับผู้ใช้งานทั่วไป

การสร้างแผนการเดินทางด้วย RAG และ LLM ประกอบด้วย 5 ส่วนหลัก ได้แก่ นักท่องเที่ยว (User), API Gateway, RAG Service, Vector Database และ LLM มีลำดับขั้นตอน 8 ขั้นตอน: (1) ผู้ใช้ป้อนเงื่อนไขการเดินทางผ่าน API Gateway (2) API Gateway ส่งต่อคำขอไปยัง RAG Service (3) RAG Service สืบค้นข้อมูลสถานที่ที่สอดคล้องใน Vector Database (4) Vector Database คืนค่าข้อมูลบริบท (Context) (5) RAG Service ประกอบ Context และ Prompt ส่งให้ LLM (6) LLM สร้างผลลัพธ์แผนการเดินทางส่งคืน RAG Service (7) RAG Service ส่งแผนที่สมบูรณ์กลับไปยัง API Gateway (8) API Gateway แสดงผลบนหน้าจอผู้ใช้

การทำงานของระบบผู้ช่วยเสมือน (Chatbot) มีลำดับเหตุการณ์ 9 ขั้นตอน: ผู้ใช้ส่งคำถามผ่าน API Gateway → ส่งต่อไปยัง Chatbot Service → ค้นหาบริบทใน Vector Database → ดึงข้อมูลสถานที่ที่เกี่ยวข้อง 3 อันดับแรก → ประกอบคำสั่ง (System Prompt + Context Prompt + User Message) → ส่งให้ LLM ประมวลผล → LLM สร้างคำตอบส่งคืน Chatbot Service → ส่งคืนคำตอบไปยัง API Gateway → แสดงผลบนหน้าจอผู้ใช้

แทรกภาพ Detailed Sequence Diagram: การบันทึกและแก้ไขข้อมูลผ่าน API (ฝั่งผู้ดูแลระบบ)

ภาพที่ 21 Detailed Sequence Diagram สำหรับผู้ดูแลระบบ

การจัดการข้อมูล (สถานที่/กิจกรรม/อีเวนต์) ผ่านสถาปัตยกรรมแบบบริการ (Service-Oriented Architecture) ประกอบด้วย Frontend, API Gateway, Data Service, Embedding Service, ฐานข้อมูลเชิงเอกสาร (Document Database) และฐานข้อมูลเวกเตอร์ กระบวนการเริ่มจากผู้ดูแลระบบส่งคำขอผ่าน API Gateway ซึ่งตรวจสอบสิทธิ์ (Authentication/Authorization) ก่อนส่งต่อไปยัง Data Service เพื่อตรวจสอบความถูกต้องของข้อมูล (Data Validation) และบันทึก/ปรับปรุงข้อมูลลงฐานข้อมูลเชิงเอกสาร จากนั้นส่งข้อความไปยัง Embedding Service เพื่อแบ่งข้อความ (Chunking) และแปลงเป็นเวกเตอร์ (Embedding) ก่อนจัดเก็บในฐานข้อมูลเวกเตอร์ สุดท้ายส่งผลลัพธ์กลับไปยังส่วนติดต่อผู้ใช้ในรูปแบบสถานะสำเร็จ

4.2.3 การออกแบบฐานข้อมูล

ระบบใช้ฐานข้อมูลเชิงเอกสาร (Document-Oriented Database) ด้วย MongoDB เป็นฐานข้อมูลหลัก และใช้ฐานข้อมูลเวกเตอร์สำหรับจัดเก็บเวกเตอร์ฝังตัว (Embedding Vectors) ที่ใช้ในกระบวนการค้นคืนข้อมูลเชิงความหมาย (Semantic Retrieval) การออกแบบฐานข้อมูลมุ่งเน้นความยืดหยุ่น รองรับข้อมูลกึ่งโครงสร้าง (Semi-Structured Data) ที่ได้จาก Google Places API และข้อมูลที่ผู้ดูแลระบบเพิ่มเข้าสู่ระบบ

แทรกภาพ ER Diagram แสดงการออกแบบฐานข้อมูลระบบ

ภาพที่ 22 E-R Diagram แสดงการออกแบบฐานข้อมูลระบบ

โครงสร้างของแต่ละตารางแสดงดังตารางที่ 16–21

ตารางที่ 16 ตารางฐานข้อมูล Place (สถานที่ท่องเที่ยว)

Field	Type	Description
_id	ObjectId	Primary Key
google_place_id	String (unique)	รหัสสถานที่จาก Google
core.name	String	ชื่อสถานที่
core.primaryType	String	ประเภทหลัก
core.types	Array	ประเภททั้งหมด
core.location	Object	lat, lng
core.rating	Number	คะแนนรีวิว
core.userRatingCount	Number	จำนวนรีวิว
core.priceLevel	Number	ระดับราคา
contact.phone	String	เบอร์โทร
contact.website	String	เว็บไซต์
address.formatted	String	ที่อยู่
openingHours	Object	เวลาทำการ
media.photos	Array	รูปภาพ
reviews	Array	รีวิว
features	Object	คุณสมบัติสถานที่
extra	Object	ข้อมูลเพิ่มเติม
maps	Object	ลิงก์ Google Maps
metadata	Object	ข้อมูลการดึงล่าสุด
createdAt / updatedAt	Date	วันที่สร้าง / แก้ไข

ตารางที่ 17 ตารางฐานข้อมูล KnowledgeBase

Field	Type	Description
_id	ObjectId	Primary Key
title	String	หัวข้อข้อมูล
content	String	เนื้อหา
category	Enum	transport, weather, local-customs ฯลฯ
isActive	Boolean	เปิดใช้งาน
isPinned	Boolean	ใส่ใน system prompt
tags	Array	คีย์เวิร์ด
metadata.sourceUrl	String	แหล่งอ้างอิง
createdAt	Date	วันที่สร้าง

ตารางที่ 18 ตารางฐานข้อมูล Event

Field	Type	Description
_id	ObjectId	Primary Key
name	String	ชื่องาน
slug	String	URL
category	Enum	festival, concert ฯลฯ
description	String	รายละเอียด
venues	Array	สถานที่จัด
schedule.startDate / endDate	Date	วันเริ่ม / วันสิ้นสุด
admission	Object	ข้อมูลบัตร
organizer	Object	ผู้จัด
status	Enum	upcoming, ongoing
metadata	Object	สถานะเผยแพร่

ตารางที่ 19 ตารางฐานข้อมูล Itinerary

Field	Type	Description
_id	ObjectId	Primary Key
title	String	ชื่อทริป
theme	String	ธีมการเดินทาง
total_days	Number	จำนวนวันทั้งหมด
total_distance_km	Number	ระยะทางรวม (กม.)
total_cost_estimate	Number	ค่าใช้จ่ายรวมประเมิน (บาท)
generatedBy	String	แหล่ง ที่มา (Default: “llm”/“system”)
isPublic	Boolean	สถานะ การ เผย แพร่ (Default: false)
userId	ObjectId	อ้างอิงเจ้าของทริป (User)
note	String	หมายเหตุจากระบบ
itin[].day / date	Number / String	วันที่ของทริป
itin[].day_cost_estimate	Number	ค่าใช้จ่ายประเมินในแต่ละวัน
itin[].day_travel_time_total	Number	เวลา เดิน ทาง รวม ใน แต่ละ วัน (นาที)
itin[].schedule[].place.*	Mixed	Snapshot ของ สถาน ที่ (id, name, lat/lng, types, rating, price_level, address)
itin[].schedule[].arrival_time / departure_time	String	เวลาที่ไปถึง / ออกจากสถานที่
itin[].schedule[].travel_time_min / distance_km	Number	เวลา / ระยะทางจากสถานที่ก่อนหน้า
itin[].schedule[].status	String	สถานะ (เช่น “Open”, “End of Day”)
itin[].schedule[].wait_time_min	Number	เวลารอ (ถ้ามี)
summary	Object	ข้อมูลสรุปทริป
createdAt	Date	วันที่สร้างแผนการเดินทาง

หมายเหตุ: `itin[]` *ย่อมาจาก* `itinerary[]` *(ชื่อเต็มของฟิลด์อาเรย์ในเอกสาร)*

ตารางที่ 20 ตารางฐานข้อมูล EmbeddingDocument (Vector Store)

Field	Type	Description
_id	ObjectId	Primary Key
sourceType	Enum	place, activity, event
sourceId	ObjectId	อ้างอิงต้นทาง
google_place_id	String	(ถ้ามี)
chunkText	String	ข้อความที่ใช้ embed
embedding	Array(Number)	เวกเตอร์
createdAt	Date	วันที่สร้าง

ตารางที่ 21 ตารางฐานข้อมูล User (Admin)

Field	Type	Description
_id	ObjectId	Primary Key
name	String	ชื่อ
email	String	อีเมล (ใช้ล็อกอิน)
password	String (hashed)	รหัสผ่านที่ผ่านการเข้ารหัส
role	String	สิทธิ์การใช้งาน (เช่น admin)
createdAt	Date	วันที่สร้าง

ตาราง Place ถูกออกแบบมาเพื่อจัดเก็บข้อมูลเชิงลึกของสถานที่ท่องเที่ยว ร้านอาหาร และจุดสนใจต่าง ๆ แบบรวมศูนย์ รองรับการบูรณาการร่วมกับระบบภายนอก (เช่น google_place_id) ตาราง KnowledgeBase ทำหน้าที่เป็นคลังความรู้ส่วนกลาง โดยแอตทริบิวต์ isPinned กำหนดเงื่อนไขในการดึงข้อมูลไปใช้เป็น System Prompt ตาราง Event มุ่งเน้นบริหารจัดการข้อมูลกิจกรรมที่มีข้อจำกัดทางเวลา ตาราง Itinerary จัดเก็บรายละเอียดแผนการเดินทางที่สร้างโดยระบบ (AI-generated) แบบเบ็ดเสร็จ ตาราง EmbeddingDocument รองรับสถาปัตยกรรม RAG โดยแปลงข้อความให้อยู่ในรูปแบบเวกเตอร์ และตาราง User ถูกออกแบบมาเพื่อการบริหารจัดการสิทธิและการเข้าถึงระบบของผู้ดูแลระบบเป็นหลัก โดยจัดเก็บรหัสผ่านในรูปแบบที่ผ่านการเข้ารหัส (Hashed Password)

4.2.4 Mockup User Interface

แทรกภาพ Wireframe แสดงการใช้งานระบบ

ภาพที่ 23 Wireframe แสดงการใช้งานระบบ

(1) ส่วนของผู้ใช้งานทั่วไป (Client-Side)

ได้รับการออกแบบในรูปแบบแอปพลิเคชันสำหรับสมาร์ทโฟน (Mobile Application) เพื่อเน้นความสะดวกในการเข้าถึงข้อมูลระหว่างการเดินทาง ประกอบด้วยหน้าจอและฟังก์ชันการทำงานที่สำคัญ ได้แก่

- หน้าหลัก (Home) และหน้ารายละเอียดสถานที่: แสดงผลข้อมูล รูปภาพ และรายละเอียดของสถานที่ท่องเที่ยวต่าง ๆ พร้อมระบบนำทาง (Navigation)
- ระบบวางแผนการเดินทางและแนะนำสถานที่ (Trip Planning & Recommend): ช่วยให้นักท่องเที่ยวออกแบบเส้นทาง หรือรับการแนะนำสถานที่ท่องเที่ยวที่เหมาะสมตามความต้องการ
- ระบบถาม-ตอบอัตโนมัติ (Chatbot): ทำหน้าที่เป็นผู้ช่วยเสมือนจริง ครอบคลุมข้อมูลทั่วไปเกี่ยวกับจังหวัดขอนแก่น การช่วยเหลือนักท่องเที่ยว และข้อมูลเชิงลึกเกี่ยวกับสถานที่ท่องเที่ยวที่จัดเก็บไว้ในฐานข้อมูล

(2) ส่วนของผู้ดูแลระบบ (Admin-Side)

ได้รับการออกแบบในรูปแบบเว็บแอปพลิเคชัน (Web Application) สำหรับการใช้งานบนคอมพิวเตอร์ เพื่อประสิทธิภาพในการบริหารจัดการฐานข้อมูลขนาดใหญ่ ประกอบด้วยส่วนการทำงานหลัก ได้แก่

- หน้ากระดานสรุปข้อมูล (Dashboard): แสดงผลภาพรวมและสถิติที่สำคัญของระบบ เช่น ตัวเลขสถิติการใช้งาน ข้อมูลสถานที่ยอดนิยม หรือกิจกรรมที่ได้รับความสนใจ
- ระบบจัดการฐานข้อมูล (Data Management): แบ่งออกเป็นส่วนการจัดการ “ข้อมูลสถานที่ท่องเที่ยว” และ “ข้อมูลกิจกรรม”

ช่วยให้ผู้ดูแลระบบสามารถเพิ่ม ลบ ตรวจสอบสถานะ และแก้ไขปรับปรุงข้อมูลในระบบให้เป็นปัจจุบันอยู่เสมอ

4.3 การพัฒนาโปรแกรม

4.3.1 ภาษาและเครื่องมือที่ใช้ในการพัฒนา

4.3.1. **Frontend:** React (Web/Mobile)

4.3.2. **Backend:** Node.js (Express), Python (FastAPI สำหรับ AI Gateway)

4.3.3. **Database:** MongoDB (ข้อมูลกึ่งโครงสร้าง), Chroma (ฐานข้อมูลเวกเตอร์)

4.3.4. **AI & Libraries:** โมเดลภาษาขนาดใหญ่ (LLM) ผ่าน OpenAI API, ไลบรารีสำหรับสร้าง RAG Pipeline

4.3.5. **API ภายนอก:** Google Places API สำหรับดึงข้อมูลตั้งต้น (Data Seeding)

4.3.2 โครงสร้างของโปรแกรม

4.3.3 ข้อกำหนดเฉพาะของแต่ละส่วน

4.3.4 การติดต่อกับฐานข้อมูลและ Services

4.4 การทดสอบระบบ

4.4.1 วิธีการทดสอบ

(1) การทดสอบยอมรับจากผู้ใช้ (User Acceptance Testing: UAT)

ทดสอบกับกลุ่มผู้ใช้เป้าหมาย (นักศึกษา/นักท่องเที่ยวในจังหวัดขอนแก่น) โดยใช้เกณฑ์ประเมินด้านความง่ายในการใช้งาน (Ease of Use) ความพึงพอใจ (Satisfaction) และความเข้าใจในการนำเสนอข้อมูล

(2) วิธีการประเมินผลของการวิจัย (LLM as a Judge)

เพื่อให้มั่นใจว่าแผนการเดินทางและคำแนะนำที่ระบบสร้างขึ้นด้วย AI มีความถูกต้องและใช้งานได้จริง งานวิจัยนี้ได้พัฒนากลไกการตรวจสอบผลลัพธ์โดยใช้ LLM อีกตัวหนึ่งทำหน้าที่เป็น “ผู้ประเมิน” (Judge) แทนการเขียนฟังก์ชันตรวจสอบแบบตายตัว โดยใช้เกณฑ์การประเมินดังตารางที่ 22

ตารางที่ 22 เกณฑ์การประเมินคุณภาพแผนการเดินทางด้วยเทคนิค LLM-as-a-Judge

มิติการประเมิน	**คะแนนเต็ม**	**ปัจจัยที่ใช้พิจารณา**
1. ความสอดคล้องกับข้อจำกัดและเงื่อนไข (Constraint Compliance)	40	สถานที่บังคับ (Must-go places), จังหวะการเดินทาง (Trip pace), กรอบงบประมาณ (Budget level), กรอบเวลา (Time bounds)
2. ความสมเหตุสมผลด้านพื้นที่และเวลา (Spatial-Temporal Logic)	30	การเดินทางและเส้นทาง (Travel feasibility), การจัดตารางมื้ออาหาร (Meal scheduling), ความเหมาะสมของเวลา (Time appropriateness)
3. การตอบสนองความชอบส่วนบุคคลและประสบการณ์ (Personalization & Experience)	30	ความตรงใจ (Interest alignment), ความลื่นไหลของทริป (Natural flow), ความหลากหลาย (Diversity of POIs)
รวม	**100**	

มิติที่ 1 (Constraint Compliance) จะหักคะแนนเมื่อ: แผนการเดินทางไม่บรรจุสถานที่ที่ผู้ใช้งานระบุว่า “ต้องไป”; จัดสรรเวลาไม่สอดคล้องกับรูปแบบความเร็วของทริปที่กำหนด (เช่น ขาดการเว้นระยะพักในรูปแบบผ่อนคลาย หรือมีช่วงเวลาว่างมากเกินไปในรูปแบบอัดแน่น); เลือกสถานที่ที่มีระดับราคาขัดแย้งกับงบประมาณ; หรือจัดตารางกิจกรรมล่วงเลยเวลาเริ่มต้น/สิ้นสุดของวัน

มิติที่ 2 (Spatial-Temporal Logic) จะหักคะแนนเมื่อ: กำหนดเวลาเดินทางน้อยกว่าความเป็นจริงหรือจัดเส้นทางวกวน (Zigzagging); จัดมื้ออาหารหลักติดต่อกันโดยไม่มีกิจกรรมอื่นคั่น; หรือจัดตารางกิจกรรมขัดกับบริบทของสถานที่ (เช่น จัดตารางเยี่ยมชมตลาดกลางคืนในเวลากลางวัน)

มิติที่ 3 (Personalization & Experience) จะเพิ่มคะแนนเมื่อสถานที่ถูกคัดเลือกตรงกับความสนใจของผู้ใช้อย่างแม่นยำ และลำดับกิจกรรมมีความต่อเนื่องสลับอารมณ์การท่องเที่ยวได้ดี แต่จะหักคะแนนหากแผนการเดินทางมีความจำเจ ซ้ำซ้อน หรือขาดจุดเด่น (เช่น จัดกำหนดการเยี่ยมชมวัด 4 แห่งติดต่อกัน)

4.4.2 ผลการทดสอบ

ผลการทดสอบและความคืบหน้าล่าสุดของแต่ละส่วนของระบบ ณ ช่วงเวลาจัดทำรายงานฉบับนี้ แสดงไว้ในบทที่ 5 หัวข้อ “สรุปผลการดำเนินโครงงาน” เนื่องจากระบบยังอยู่ระหว่างการพัฒนาและยังไม่ได้เชื่อมต่อกันแบบครบวงจร (End-to-End) การทดสอบแบบ UAT เต็มรูปแบบจึงยังไม่ได้ดำเนินการ

บทที่ 5
สรุปผลการดำเนินงาน

การจัดทำโครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาระบบแนะนำการท่องเที่ยว โดยผสานการทำงานของปัญญาประดิษฐ์ โมเดลภาษาขนาดใหญ่ (LLM) และข้อมูลสถานที่จริง ซึ่งจากการทดลองและพัฒนาระบบ สามารถสรุปผลการดำเนินงาน ข้อจำกัด ปัญหาและอุปสรรค ตลอดจนข้อเสนอแนะในการพัฒนาต่อไปได้ดังนี้

5.1 สรุปผลการดำเนินโครงงาน

การดำเนินโครงงานในระยะนี้ เน้นไปที่การทดสอบเทคโนโลยีแกนหลัก (Core Technologies) และการขึ้นโครงสร้างของเว็บแอปพลิเคชัน โดยคณะผู้จัดทำได้ดำเนินการพัฒนาส่วนของเว็บแอปพลิเคชัน (Frontend และ Backend พื้นฐาน) ไปแล้วร้อยละ 60 ของระบบทั้งหมด นอกจากนี้ยังได้ทำการทดสอบการทำงานของโมเดลภาษาขนาดใหญ่ (Gemini Flash) ในการทำความเข้าใจบริบทและสร้างคำแนะนำ รวมถึงการทดสอบเชื่อมต่อและดึงข้อมูลจาก Google Places API เพื่อนำข้อมูลสถานที่จริงมาเป็นฐานข้อมูลเริ่มต้น

ความคืบหน้าของแต่ละส่วนแสดงดังตารางที่ 23

ตารางที่ 23 ตารางแสดงผลการดำเนินโครงงาน

ส่วน การ ทำงาน (Module)	รายละเอียดผลการดำเนินงาน	%	สถานะความคืบหน้า
1. ระบบจัดการส่วนหลัง และ ฐาน ข้อมูล (Backend)	พัฒนา โครงสร้าง ฐาน ข้อมูล (Database Models) 7 โมเดล และ API สำหรับ จัดการ ข้อมูล สถาน ที่ (Place Management) ตลอดจนตรวจเช็คสถานะระบบ	70%	สำเร็จ: Models 100%, Place APIs 100% / คงค้าง: API กิจกรรม, อีเวนต์ และ โปรแกรมทริป
2. ระบบ ส่วน หน้า สำหรับผู้ใช้งาน (User Portal)	พัฒนา ส่วน แสดง ผล (UI) หน้า หลัก หน้า ราย ละเอียด สถาน ที่ หน้าแนะนำ และคอมโพเนนต์ย่อย พร้อมเชื่อมต่อ API เพื่อแสดงผล ข้อมูลสถานที่	75%	สำเร็จ: UI/UX หลัก 90%, เชื่อมต่อ API 80% / คงค้าง: ลอจิกของแชทบอทและ ระบบวางแผนทริป
3. ระบบ จัดการ สำหรับ ผู้ ดูแล ระบบ (Admin Portal)	พัฒนา หน้า กระดาน ข้อมูล (Dashboard) หน้า จัดการ สถาน ที่ กิจกรรม และ อีเวนต์ รวม ถึง โครงสร้าง UI Components พื้น ฐานครบถ้วน	20%	คงค้าง: การยืนยันตัว ตน และการเชื่อมต่อ ข้อมูลจริงแทน Mock Data
4. กา รบู รณา การ ปัญญาประดิษฐ์ (AI & LLM Integration)	ออกแบบ โครงสร้าง ข้อมูล Embedding และ ทดสอบ การ ทำงานของ Gemini Flash ในการ สร้างแผนการเดินทาง	–	ทดสอบสำเร็จ รอการ ปรับปรุง และ ทดลอง ใช้ จริง (พร้อม บูรณา การ เข้า กับ ระบบ แช ทบ อท และ วาง แผน ทริป)
5. การเชื่อมต่อข้อมูล ภายนอก (Data Re-trieval)	บูรณา การ Google Places API ครบทั้ง 4 บริการ ได้แก่ การค้นหา, ดึง ราย ละเอียด, รูปภาพ และ ข้อมูล แผนที่ พร้อม ระบบ Map-per	100%	สำเร็จ ครบ ทั้ง 4/4 โมดูลย่อยที่กำหนดไว้

5.2 ข้อจำกัดของระบบ

5.2.1 ข้อจำกัดด้านการทำงาน

จากการพัฒนาระบบในเฟสปัจจุบัน พบว่าระบบยังมีข้อจำกัดสำคัญ คือ การทำงานของระบบยังอยู่ในลักษณะแยกส่วน (System Components ยังไม่ได้เชื่อมต่อกันอย่างสมบูรณ์) โดยส่วนของเว็บแอปพลิเคชันที่พัฒนาแล้วประมาณร้อยละ 60 ยังไม่ได้ทำการเชื่อมต่อเข้ากับระบบประมวลผล AI (Gemini Flash) และ API สำหรับดึงข้อมูลสถานที่ ทำให้ยังไม่สามารถทดสอบการทำงานของระบบแบบครบวงจร (End-to-End System) ได้

นอกจาก นี้ ระบบ ยัง ขาด ข้อมูล ความ หนา แน่น ของ ผู้คน ใน แต่ละ ช่วง เวลา (Popular Times หรือ Crowd Density) ซึ่งเป็นตัวแปรสำคัญที่ช่วยให้โมเดล LLM สามารถประมวล ผล และ แนะนำ แผนการ เดิน ทาง ที่ หลีก เลี่ยง ช่วง เวลา ที่ มี ผู้คน หนา แน่น ได้ อย่าง มี

ประสิทธิภาพ

5.2.2 ข้อจำกัดด้านฮาร์ดแวร์และซอฟต์แวร์

ระบบพึ่งพาการประมวลผลผ่าน Gemini Flash API ซึ่งมีการกำหนดเกณฑ์การใช้งาน (Rate Limits) ทั้งในแง่ของจำนวนครั้งต่อนาที (RPM) และจำนวน Token ต่อวัน ทำให้อาจเกิดการหยุดชะงักของบริการเมื่อโควต้าหมดลงหากใช้งานระยะยาว

5.3 ปัญหาอุปสรรค และแนวทางแก้ไข

ในระหว่างการพัฒนาและทดสอบระบบ คณะผู้จัดทำพบปัญหาทางเทคนิคหลายประการ โดยเฉพาะในส่วนของการรวบรวมข้อมูลเชิงลึกจากแหล่งข้อมูลภายนอก ซึ่งได้มีการวิเคราะห์สาเหตุและกำหนดแนวทางแก้ไข ดังแสดงในตารางที่ 24

ตารางที่ 24 ตารางแสดงการสรุปปัญหา อุปสรรค และแนวทางแก้ไข

ปัญหาและอุปสรรค	สาเหตุหลัก	แนวทางแก้ไข
ไม่ สามารถ ดึง ข้อมูล Popular Times จาก Google Maps ได้อย่างสม่ำเสมอ	การ เขียน สคริปต์ ดึง ข้อมูล (Web Scraping) ถูก บล็อก โดย ระบบ ป้องกัน การ ดึง ข้อมูล อัตโนมัติ (Anti-Scraping / IP Blocking) ของ Google	1. ศึกษาและทดลองใช้ Third-Party APIs ที่ให้บริการข้อมูลแผนที่อย่างถูกต้องตามข้อกำหนด (เช่น Serp-Api) 2. หากไม่สามารถเข้าถึงข้อมูลได้ อาจใช้วิธีจำลองข้อมูลทางสถิติ (Mock Data) เพื่อทดสอบตรรกะของระบบ LLM ในเบื้องต้น 3. ปรับใช้ข้อมูลอื่นที่สามารถดึงได้จาก Google Places API เช่น รีวิวผู้ใช้ หรือเวลาเปิด-ปิดสถานที่ มาเป็นตัวแปรในการประเมินแทน
โค วต้า ของ LLM อาจจะหมดลงได้หากใช้งานระยะยาว	ระบบพึ่งพาการประมวลผลผ่าน Gemini Flash API ซึ่งมีการกำหนดเกณฑ์การใช้งาน (Rate Limits) ทั้งจำนวนครั้งต่อนาที (RPM) และจำนวน Token ต่อวัน	1. ศึกษาและติดตั้ง Local LLM (เช่น Llama 3 หรือ Mistral) บนเครื่องเซิร์ฟเวอร์ส่วนตัว เพื่อลดการพึ่งพา API ภายนอก 2. ระบบ Hybrid Processing: ใช้ Local LLM จัดการงานพื้นฐาน และใช้ Gemini Flash เฉพาะงานวิเคราะห์เชิงลึกที่ซับซ้อน 3. Semantic Caching: จัดเก็บคำตอบที่พบบ่อยไว้ใน Vector Database เพื่อลดการเรียกใช้งาน LLM ซ้ำซ้อน

5.4 ข้อเสนอแนะ ในการพัฒนาต่อไป

5.4.1 การปรับปรุงฟังก์ชันการทำงาน

เพื่อให้ระบบมีความสมบูรณ์และสามารถใช้งานได้จริง ควรเร่งดำเนินการพัฒนาส่วน API Endpoint สำหรับเชื่อมต่อระหว่าง Web Application กับระบบประมวลผล AI เพื่อให้ระบบสามารถรับข้อมูลจากผู้ใช้งาน (Input) และส่งผลลัพธ์กลับไปแสดงผลแก่ผู้ใช้ (Output) ได้อย่างสมบูรณ์ นอกจากนี้ ควรศึกษาวิธีการดึงข้อมูลเกี่ยวกับความหนาแน่นของผู้คน (Crowd Metrics) จากแหล่งข้อมูลเปิดอื่น ๆ เพื่อเพิ่มประสิทธิภาพให้ระบบสามารถแนะนำแผนการเดินทางได้อย่างยืดหยุ่นและสอดคล้องกับสถานการณ์จริง (Dynamic Recommendation) เมื่อระบบพัฒนาเสร็จสมบูรณ์ ควรนำไปทดสอบกับผู้ใช้งานจริงด้วย UAT เพื่อประเมินประสิทธิภาพของระบบ รวมถึงระดับความพึงพอใจของผู้ใช้งาน

5.4.2 แนวทางการวิจัยต่อยอด

จากผลการดำเนินงานและปัญหาอุปสรรคที่พบ คณะผู้จัดทำได้กำหนดแผนการดำเนินงานในช่วง 4 เดือนถัดไป เพื่อให้โครงงานสามารถพัฒนาเสร็จสมบูรณ์และนำไปใช้งานได้จริง โดยมีรายละเอียดดังแสดงในตารางที่ 25

ตารางที่ 25 แผนการดำเนินงานในระยะต่อไป

ระยะเวลา	แผนการดำเนินงาน	รายละเอียด
เดือนที่ 1	การ จัดการ ข้อมูล และ การ พัฒนา เว็บ ส่วน ที่ เหลือ	ศึกษาและทดลองวิธีการทางเลือกในการดึงข้อมูล Popular Times (เช่น Third-Party API หรือ Mock Data); พัฒนาส่วนของ Web Application ที่เหลืออีกประมาณ 40% ให้เสร็จสมบูรณ์; เพิ่มฟังก์ชันสำหรับรับข้อมูลจากผู้ใช้งาน
เดือนที่ 2	การเชื่อมต่อระบบ (System Integration)	พัฒนา API Endpoint เพื่อเชื่อมต่อ Web Application กับ Gemini Flash และ Google Places API; ทดสอบการส่งข้อมูลระหว่าง Frontend และ Backend
เดือนที่ 3	การปรับแต่งโมเดลและการทดสอบระบบ	ปรับแต่ง Prompt Engineering ของ Gemini Flash เพื่อวิเคราะห์ข้อมูลสถานที่และสร้างแผนการเดินทางได้แม่นยำ; ทดสอบระบบแบบ System Integration Testing; แก้ไขข้อผิดพลาด (Bug Fixing)
เดือนที่ 4	การประเมินผลและการจัดทำเอกสาร	ทดสอบระบบกับผู้ใช้งานจริงด้วย UAT; วิเคราะห์ผลการใช้งานและรวบรวมข้อเสนอแนะ; จัดทำรายงานโครงงานฉบับสมบูรณ์และเตรียมนำเสนอ

เอกสารอ้างอิง

Ali, A. A., Kaka-khan, K. M., & Al-chalabi, I. A. (2023). Chatbot-based tourist guide using artificial intelligence markup language. *The Journal of Duhok University*, *26*(2), 578–587. https://doi.org/10.26682/csjuod.2023.26.2.52

Banerjee, A., Satish, A., & Wörndl, W. (2025). Enhancing tourism recommender systems for sustainable city trips using retrieval-augmented generation. https://doi.org/10.1007/978-3-031-87654-7_3

Barua, B., & Kaiser, M. S. (2024). Optimizing travel itineraries with ai algorithms in a microservices architecture: Balancing cost, time, preferences, and sustainability. https://doi.org/10.48550/arXiv.2410.17943

Deshmukh, C., Khan, A., Upadhyay, P., & Savalkar, S. (2025). Ai-powered personalized travel itinerary planning using real-time data. *IJSAT - International Journal on Science and Technology*, *16*(2). https://doi.org/10.71097/IJSAT.v16.i2.3867

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., & Guo, J. (2025). A survey on llm-as-a-judge. https://doi.org/10.48550/arXiv.2411.15594

Jyothi, A. N., Adimulam, G. R., Neelam, C., Gowud Vnl, J. N., Unnamatla, R. K., & Tumpati, K. (2025). Travel finder – an ai-powered travel planning and recommendation system. *International Journal of Innovative Science and Research Technology*, 697–706. https://doi.org/10.38124/ijisrt/25apr515

Ma, X., Chen, X., & Yuan, C. (2025). Crossroads of ai and tourism: Enhancing destination management and traveler engagement. *Proceedings of the 2025 2nd International Conference on Generative Artificial Intelligence and Information Security*, 31–36. https://doi.org/10.1145/3728725.3728731

Milton, T., & Philosophers, C. (2023). Artificial intelligence in tourism – a review of trends opportunities and challenges. *1*, 01–11.

Shao, Z., Wu, J., Chen, W., & Wang, X. (2025). Personal travel solver: A preference-driven llm-solver system for travel planning. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 27622–27642. https://doi.org/10.18653/v1/2025.acl-long.1339

Sousa, A. E., Cardoso, P., & Dias, F. (2024). The use of artificial intelligence systems in tourism and hospitality: The tourists' perspective. *Administrative Sciences*, *14*(8), 165. https://doi.org/10.3390/admsci14080165

Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., & Su, Y. (2024). Travelplanner: A benchmark for real-world planning with language agents. *Proceedings of the 41st International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2402.01622

Zheng, H. S., Mishra, S., Zhang, H., Chen, X., Chen, M., Nova, A., Hou, L., Cheng, H., Le, Q. V., Chi, E. H., & Zhou, D. (2024). Natural plan: Benchmarking llms on natural language planning. https://doi.org/10.48550/arXiv.2406.04520

ภาคผนวก ก
ตัวอย่าง UI หรือ Prototype

ภาพหน้าจอระบบ

ภาคผนวกนี้แสดงตัวอย่างหน้าจอของระบบที่พัฒนาขึ้น เพื่อให้เห็นภาพรวมของการออกแบบส่วนติดต่อกับผู้ใช้

หน้าจอเข้าสู่ระบบ (Login) แสดงดังภาพที่ 24

แทรกภาพหน้าจอ Login ที่นี่
ใช้คำสั่ง `\includegraphics`

ภาพที่ 24 หน้าจอเข้าสู่ระบบ (Login)

หน้าจอหลัก (Dashboard) แสดงดังภาพที่ 25

แทรกภาพหน้าจอหลักที่นี่

ภาพที่ 25 หน้าจอหลัก (Dashboard)

คู่มือการใช้งาน

อธิบายขั้นตอนการใช้งานระบบโดยสังเขป...

ประวัติผู้เขียน

นักศึกษาคนที่ 1

นาย/นางสาว... เกิดเมื่อวันที่... ที่... สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย จากโรงเรียน... เมื่อปีการศึกษา... ปัจจุบันกำลังศึกษาระดับปริญญาตรี หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น