



เว็บแอปพลิเคชันสำหรับการวางแผนการเดินทาง

โดย

นางสาว สิริธร คำกุล

นางสาวณัฐนิชา บรรลือทรัพย์

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

เว็บแอปพลิเคชันสำหรับการวางแผนการเดินทาง

โดย

นางสาว สิริธร คำกุล

นางสาว ญัฐนิชา บรรลือทรัพย์

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

TRIP PLANNING WEB APPLICATION

BY

MISS SIREETHORN DAMKUL

MISS NATNICHIA BUNLUESAB

A FINAL-YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2024

COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ


นางสาว สิริธร คำกุล
นางสาว ญัฐนิชา บรรลือทรัพย์

เรื่อง


เว็บแอปพลิเคชันสำหรับการวางแผนการเดินทาง

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

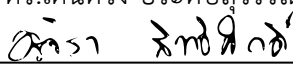
อาจารย์ที่ปรึกษา


(ผศ. ดร.ปกรณ์ ลิ้มสุทธิพรชัย)

กรรมการสอบโครงการพิเศษ


(ผศ. ดร.เด่นดวง ประดับสุวรรณ)

กรรมการสอบโครงการพิเศษ


(ผศ. ดร.อรจิรา สิทธิศักดิ์)

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ


ของ

นางสาว สิริธร คำกุล
นางสาว ญัฐนิชา บรรลือทรัพย์
เรื่อง


เว็บแอปพลิเคชันสำหรับการวางแผนการเดินทาง

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

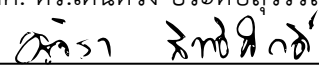
อาจารย์ที่ปรึกษา


(ผศ. ดร.ปกรณ ลีสุทธิพรชัย)

กรรมการสอบโครงการพิเศษ


(ผศ. ดร.เด่นดวง ประดับสุวรรณ)

กรรมการสอบโครงการพิเศษ


(ผศ. ดร.อรจิรา สิทธิศักดิ์)

หัวข้อโครงการพิเศษ	เว็บแอปพลิเคชันสำหรับการวางแผนการเดินทาง
ชื่อผู้เขียน	นางสาว สิริธร คำกุล
ชื่อผู้เขียน	นางสาวณัฐนิชา บรรลือทรัพย์
ชื่อปริญญา	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
สาขาวิชา/คณะ/มหาวิทยาลัย	สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
อาจารย์ที่ปรึกษาโครงการพิเศษ	ผศ. ดร.ปกรณ์ ลีสุทธิพรชัย
ปีการศึกษา	2567

บทคัดย่อ

เว็บแอปพลิเคชัน PlanMyTrip เป็นระบบช่วยวางแผนการเดินทางที่ออกแบบมาเพื่อค้นหาเส้นทางที่ใช้เวลาเดินทางรวมน้อยที่สุดระหว่างสถานที่หลายแห่ง โดยผู้ใช้งานสามารถกำหนดเงื่อนไขต่าง ๆ ได้ เช่น จำนวนจุดแวะ เวลาที่ต้องการเริ่มเดินทาง วิธีการเดินทาง และตัวเลือกการหลีกเลี่ยงค่าผ่านทาง เพื่อให้แผนการเดินทางสอดคล้องกับความต้องการและสถานการณ์จริง

ระบบรองรับการกรอกสถานที่ได้สูงสุด 10 จุด พร้อมระบบแนะนำสถานที่อัตโนมัติ (Autocomplete) ผ่าน Google Places API เพื่อความสะดวกและลดข้อผิดพลาด ข้อมูลที่ป้อนจะถูกนำไปคำนวณผ่าน Google Maps Directions API เพื่อประเมินระยะทางและเวลาเดินทาง จากนั้นประมวลผลด้วยอัลกอริทึม A* (A-Star) เพื่อจัดลำดับเส้นทางที่เหมาะสมที่สุด นอกจากนี้ยังมีการตรวจสอบเวลาเปิด-ปิดของแต่ละสถานที่ เพื่อคัดกรองเส้นทางที่ไม่สามารถไปถึงได้ในเวลานั้นโดยอัตโนมัติ

เมื่อคำนวณเสร็จแล้ว ระบบจะแสดงแผนการเดินทางเรียงตามเวลาที่ใช้จากน้อยไปมาก พร้อมรายละเอียดแต่ละช่วง ได้แก่ ชื่อสถานที่ ระยะทาง และระยะเวลาโดยประมาณ ผู้ใช้งานสามารถคลิกปุ่มเพื่อเปิดเส้นทางใน Google Maps ได้ทันทีโดยไม่ต้องกรอกข้อมูลใหม่ ทำให้สามารถใช้งานจริงได้อย่างสะดวกรวดเร็ว เว็บแอปนี้เหมาะสำหรับผู้ที่ต้องการวางแผนการเดินทางหลายจุดในวัน เช่น นักท่องเที่ยวหรือผู้ที่มีธุระหลายแห่ง ช่วยลดความซับซ้อนในการวางแผน และใช้งานได้ทุกอุปกรณ์

โครงการนี้มีเป้าหมายเพื่อนำเสนอแนวทางในการประยุกต์ใช้เว็บแอปพลิเคชัน (Web Application) กับการวางแผนการเดินทาง โดยผลการดำเนินงานของโครงการได้บรรลุตามวัตถุประสงค์ที่กำหนดไว้ได้แก่ การช่วยประหยัดเวลาในการวางแผนการเดินทาง และการจัดลำดับเส้นทางได้ตรงตามเงื่อนไขของผู้ใช้งาน อีกทั้งยังสามารถพัฒนาได้ภายใต้ขอบเขตที่กำหนดอย่างครบถ้วน ไม่ว่าจะเป็นรูปแบบการใช้งานแบบ One Day Trip การรองรับทุกอุปกรณ์ด้วย Responsive Design หรือการใช้ข้อมูลและบริการจาก Google Maps API อย่างมีประสิทธิภาพ แสดงให้เห็นถึงความเป็นไปได้ในการนำเทคโนโลยีเว็บแอปพลิเคชันมาประยุกต์ใช้ในการช่วยวางแผนการเดินทางได้อย่างมีประสิทธิภาพ

คำสำคัญ: เว็บแอปพลิเคชัน, วางแผนการเดินทาง, Google Map API

Thesis Title	TRIP PLANNING WEB APPLICATION
Author	Miss Sireethorn Damkul
Author	Miss Natnicha Bunluesab
Degree	Bachelor of Science
Major Field/Faculty/University	Computer Science Faculty of Science and Technology Thammasat University
Project Advisor	Asst. Prof. Dr. Pakorn Leesutthipornchai
Academic Years	2024

ABSTRACT

PlanMyTrip is a web application developed to assist users in efficiently planning travel routes across multiple locations by determining the sequence that results in the shortest total travel time. Users can input up to 10 destinations and customize key parameters such as departure time, mode of transportation (e.g., driving or walking), and preferences like avoiding toll roads. The system aims to offer flexible, real-world planning by taking user-defined constraints into account.

The application features an autocomplete input powered by the Google Places API to ensure fast and accurate location entry. Once the user provides the necessary information, the system retrieves travel time and route data using the Google Maps Directions API. It then processes this data using the A* (A-Star) algorithm to determine the optimal visiting order. Furthermore, it checks the opening and closing hours of each location using the Google Places Details API to eliminate infeasible routes based on time constraints.

After processing, the application presents the most time-efficient travel plan first, along with route details such as place names, distances, and estimated travel durations. A key feature allows users to open the entire optimized route directly in Google Maps via a generated link eliminating the need to re-enter destinations manually. PlanMyTrip is ideal for tourists, sales representatives, or anyone needing to visit multiple places in a single day, providing a streamlined, accurate planning experience accessible from any device.

This project aimed to propose a practical approach to applying web applications in travel planning. The results demonstrate that the project successfully achieved its objectives: reducing the time required for travel planning and generating route sequences based on user-defined conditions. Moreover, the system was developed within the defined scope, including one-day trip support, responsive interface design, and efficient integration of Google Maps APIs. These outcomes confirm the feasibility and effectiveness of using web-based technologies for optimized travel planning.

Keywords: Web application, trip planning, Google Maps API

กิตติกรรมประกาศ

ข้าพเจ้าขอแสดงความขอบคุณอย่างสูงต่อ อาจารย์ ผศ. ดร. ปกรณ์ ลีสุทธิพรชัย อาจารย์ที่ปรึกษาหลัก ซึ่งได้กรุณาสละเวลาอันมีค่าให้คำแนะนำ คำปรึกษา และชี้แนะแนวทางในการทำวิทยานิพนธ์ฉบับนี้ด้วยความตั้งใจและใส่ใจในทุกรายละเอียด ทำให้ข้าพเจ้าสามารถดำเนินงานจนสำเร็จลุล่วงได้ด้วยดี

นอกจากนี้ ข้าพเจ้ายังขอขอบคุณ เพื่อนนักศึกษา และ บุคคลผู้เกี่ยวข้องทุกท่าน ที่ได้ร่วมแรงร่วมใจกันสนับสนุนและแบ่งปันความรู้ ความคิด และกำลังใจตลอดระยะเวลาที่ทำงานวิจัยชิ้นนี้

สุดท้ายนี้ ข้าพเจ้าขอขอบคุณ ครอบครัว ที่คอยเป็นกำลังใจสำคัญ และสนับสนุนทั้งด้านจิตใจและทรัพยากรตลอดการศึกษา

หากวิทยานิพนธ์ฉบับนี้มีข้อบกพร่องประการใด ข้าพเจ้าน้อมรับคำแนะนำเพื่อเป็นแนวทางในการพัฒนาต่อไป

นางสาว สิริธร คำกุล

นางสาว ณิชฌิชา บรรลือทรัพย์

สารบัญ

	หน้า
ABSTRACT	4
กิตติกรรมประกาศ	5
สารบัญ	6
สารบัญภาพ	10
สารบัญตาราง	12
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.1.1 ความเปลี่ยนแปลงในยุคดิจิทัลและความสำคัญของการเดินทาง	1
1.1.2 ข้อจำกัดและปัญหาของการวางแผนเส้นทางแบบดั้งเดิม	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ของโครงการ	3
บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	4
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง	4
2.1.1 การเดินทาง	4
2.1.2 การวางแผนการเดินทาง	6
2.1.3 เว็บแอปพลิเคชัน (Web Application)	6
2.1.4 แนวคิด Path Optimization Problem	7
2.1.5 แนวคิด Shortest Path with Constraints	8

2.1.6	ทฤษฎีการเลือกเส้นทาง (Route Choice Theory)	9
2.1.7	ทฤษฎีอัลกอริทึม A* (A-Star Algorithm Theory)	11
2.2	เทคโนโลยีที่เกี่ยวข้อง	13
2.2.1	Figma	13
2.2.2	HTML5	14
2.2.3	CSS3	16
2.2.4	Bootstrap	17
2.2.5	JavaScript	18
2.2.6	React	19
2.2.7	Node.js	20
2.2.8	Express.js	22
2.2.9	Google Maps Platform	22
2.3	แอปพลิเคชันที่เกี่ยวข้อง	24
2.3.1	Google Maps	24
2.3.2	Visit a city	26
บทที่ 3	วิธีการวิจัย	31
3.1	ภาพรวมของโครงการและสภาพแวดล้อมของระบบ	31
3.2	การวิเคราะห์ขอบเขตและความต้องการของระบบ	32
3.2.1	คำอธิบายแผนภาพกรณีใช้งาน (Use Case Description)	33
3.3	ออกแบบการทำงานของระบบ	34
3.3.1	คำอธิบาย use case ของระบบ (Use case Specification)	34
3.4	ขั้นตอนการเรียกใช้ API ในเว็บแอปพลิเคชัน Plan My Trip	40
3.5	วิธีการคำนวณการหาเส้นทางที่ดีที่สุดโดยใช้ A* Algorithm	41
3.5.1	องค์ประกอบในการคำนวณ	41

3.5.2	ลักษณะการคำนวณ	41
3.5.3	ตัวอย่างเหตุการณ์	42
3.6	ขั้นตอนการทำงานเว็บแอปพลิเคชัน	46
3.7	ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย	48
3.8	ผลลัพธ์ที่คาดหวัง	49
3.9	ภาพระบบ	50
3.9.1	หน้าหลักเว็บแอปพลิเคชัน Plan my trip	50
3.9.2	หน้าวิธีการใช้งานเว็บแอปพลิเคชัน Plan my trip	51
3.9.3	หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip	52
3.9.4	Responsive design	56
3.9.5	การเปลี่ยนภาษา	59
บทที่ 4	ผลการดำเนินงาน	62
4.1	การจัดเตรียมฮาร์ดแวร์และซอฟต์แวร์	62
4.1.1	เครื่องมือที่ใช้ในการพัฒนา (Software)	62
4.2	แผนการดำเนินงาน	63
4.3	การทดสอบระบบ	65
4.3.1	การทดสอบกรณีระหว่างที่ผู้ใช้กรอกข้อมูลสถานที่และเลือกเงื่อนไข	65
4.3.2	การทดสอบการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ	66
4.3.3	การทดสอบการแสดงผลเส้นทาง	68
4.3.4	การทดสอบความแม่นยำของเส้นทาง (Accuracy)	70
4.3.5	การทดสอบความเร็วในการประมวลผล (Performance)	71
4.3.6	การทดสอบการเชื่อมต่อข้อมูลสถานที่ไปยัง google maps	72
4.4	สรุปการประเมินประสิทธิภาพของโครงการ	73

4.4.1	การประเมินการทำงานของฟังก์ชันต่าง ๆ ในเว็บแอปพลิเคชัน (Functional Test) ด้วยกรณีทดสอบ	73
บทที่ 5	สรุปผลการดำเนินการและข้อเสนอแนะ	74
5.1	สรุปผลการดำเนินงาน	74
5.2	ปัญหาและอุปสรรคที่พบ	76
5.2.1	ข้อจำกัดของข้อมูลจาก Google Maps API	76
5.2.2	ความซับซ้อนในการประมวลผลเงื่อนไขเวลาเปิด-ปิด	76
5.2.3	การคำนวณลำดับเส้นทางที่เหมาะสมภายใต้เงื่อนไขเฉพาะผู้ใช้	76
5.2.4	การจัดการข้อผิดพลาดและแจ้งเตือนผู้ใช้	76
5.2.5	การประเมินเส้นทางที่ดีที่สุดในเชิงคุณภาพ	76
5.3	ข้อเสนอแนะและสิ่งที่สามารถพัฒนาต่อยอด	77
5.3.1	การประยุกต์ใช้ปัญญาประดิษฐ์ (AI) หรือ Machine Learning	77
5.3.2	การเชื่อมต่อกับแหล่งข้อมูลภายนอกเพิ่มเติมการประยุกต์ใช้ปัญญาประดิษฐ์ (AI) หรือ Machine Learning	77
5.3.3	การเพิ่มระบบบันทึกประวัติการเดินทาง	77
5.3.4	การออกแบบระบบให้รองรับหลายภาษา (Multilingual Interface)	77
5.3.5	เพิ่มระบบแนะนำเส้นทางอัตโนมัติตามสถานการณ์ (Smart Suggestion)	78
5.3.6	เพิ่มการแสดงผลทางเลือกในการเดินทางที่หลากหลาย	78
5.3.7	การขยายระบบเพื่อรองรับการวางแผนเดินทางแบบหลายวัน (Multi-day Trip Planning)	78
5.3.8	เงื่อนไขการเดินทางด้วย รถยนต์ หรือรถสาธารณะ	79
5.3.9	การคำนวณหาลำดับเส้นทางที่ดีที่สุด	79
5.3.10	ในกรณีที่ไม่สามารถหาเส้นทางที่ดีที่สุดได้ตรงกับเงื่อนไขที่ผู้ใช้กำหนด	79
	รายการอ้างอิง	80

สารบัญภาพ

ภาพที่ 2.1 แสดงสัญลักษณ์ของภาษา Figma (Figma LOGO)	13
ภาพที่ 2.2 แสดงสัญลักษณ์ของภาษา HTML5 (HTML5 LOGO)	14
ภาพที่ 2.3 แสดงสัญลักษณ์ของภาษา HTML5 (HTML5 LOGO)	15
ภาพที่ 2.4 แสดงสัญลักษณ์ของภาษา CSS3 (CSS3 LOGO)	16
ภาพที่ 2.5 แสดงสัญลักษณ์ Bootstrap (Bootstrap LOGO)	17
ภาพที่ 2.6 แสดงสัญลักษณ์ของภาษา Javascript5 (Javascript5 LOGO)	19
ภาพที่ 2.7 แสดงสัญลักษณ์ของ React (React LOGO)	20
ภาพที่ 2.8 แสดงสัญลักษณ์ของ Express JS (Express LOGO)	22
ภาพที่ 2.9 แสดงสัญลักษณ์ของ Google Maps Platform	23
ภาพที่ 2.10 แสดงสัญลักษณ์ของ Visit a city (Visit a city LOGO)	26
ภาพที่ 3.1 แสดงสถาปัตยกรรมระบบ (Architecture Diagram)	31
ภาพที่ 3.2 แผนภาพกรณีศึกษา (Use Case Diagram)	32
ภาพที่ 3.3 แสดงภาพตัวอย่างกราฟเวลาในการเดินทาง	42
ภาพที่ 3.4 แผนภาพกิจกรรมของเว็บแอปพลิเคชัน (1)	46
ภาพที่ 3.5 แผนภาพกิจกรรมของเว็บแอปพลิเคชัน (2)	47
ภาพที่ 3.6 หน้าหลักเว็บแอปพลิเคชัน Plan my trip	50
ภาพที่ 3.7 หน้าวิธีการใช้งานเว็บแอปพลิเคชัน Plan my trip	51
ภาพที่ 3.8 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (1)	52
ภาพที่ 3.9 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (2)	53
ภาพที่ 3.10 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (3)	53
ภาพที่ 3.11 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (4)	54
ภาพที่ 3.12 แสดงภาพผลลัพธ์การวางแผนการเดินทาง	54

ภาพที่ 3.13 ภาพแสดงหน้า Google maps	55
ภาพที่ 3.14 ภาพแสดงหน้าหลักแบบ Responsive	56
ภาพที่ 3.15 ภาพแสดงหน้าวิธีการใช้งานแบบ Responsive	57
ภาพที่ 3.16 ภาพแสดงหน้าวางแผนการเดินทางแบบ Responsive (1)	58
ภาพที่ 3.17 ภาพแสดงหน้าวางแผนการเดินทางแบบ Responsive (2)	58
ภาพที่ 3.18 ภาพแสดงปุ่มเปลี่ยนภาษา	59
ภาพที่ 3.19 ภาพแสดงหน้าหลักเมื่อเป็นภาษาอังกฤษ	59
ภาพที่ 3.20 ภาพแสดงหน้าวิธีการใช้งานเมื่อเป็นภาษาอังกฤษ	60
ภาพที่ 3.21 ภาพแสดงหน้าวางแผนการเดินทางเมื่อเป็นภาษาอังกฤษ	61
ภาพที่ 3.22 ภาพแสดงผลลัพธ์เส้นทางเมื่อเป็นภาษาอังกฤษ	61
ภาพที่ 4.1 ภาพแสดงในการประมวลผลและแสดงผลข้อมูล (1)	71
ภาพที่ 4.2 ภาพแสดงในการประมวลผลและแสดงผลข้อมูล (2)	72

สารบัญตาราง

ตารางที่ 2.1 การเปรียบเทียบแอปพลิเคชันที่คล้ายคลึงกัน	29
ตารางที่ 3.1 รายละเอียดกรณีการใช้งานระบบทั้งหมด	33
ตารางที่ 3.2 รายละเอียดกรณีผู้ใช้งานกรอกข้อมูลการเดินทางลงในระบบ	34
ตารางที่ 3.3 รายละเอียดกรณีระบบคำนวณเส้นทางที่เหมาะสม	36
ตารางที่ 3.4 รายละเอียดกรณีระบบแจ้งเตือนข้อจำกัด	37
ตารางที่ 3.5 รายละเอียดกรณีระบบแสดงผลเส้นทาง	38
ตารางที่ 3.6 ส่งข้อมูลไปยัง Google Maps	39
ตารางที่ 4.1 การดำเนินงานที่ผ่านมาจนถึงปัจจุบัน	63
ตารางที่ 4.2 การทดสอบกรณีระหว่างที่ผู้ใช้กรอกข้อมูลสถานที่และเลือกเงื่อนไข	65
ตารางที่ 4.3 การทดสอบการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ	66
ตารางที่ 4.4 การทดสอบการแสดงผลเส้นทาง	68
ตารางที่ 4.5 การทดสอบความแม่นยำของเส้นทาง (Accuracy)	70
ตารางที่ 4.6 การทดสอบ ความเร็วในการประมวลผล (Performance)	71
ตารางที่ 4.7 การทดสอบการเชื่อมต่อข้อมูลสถานที่ไปยัง google maps	72
ตารางที่ 4.8 กรณีทดสอบการทำงานของฟังก์ชัน	73

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

1.1.1 ความเปลี่ยนแปลงในยุคดิจิทัลและความสำคัญของการเดินทาง

ในยุคปัจจุบันที่เทคโนโลยีดิจิทัลเข้ามามีบทบาทอย่างกว้างขวางในชีวิตประจำวันของผู้คน การเดินทางกลายเป็นกิจกรรมที่จำเป็นและมีความสำคัญมากขึ้น ไม่ว่าจะเป็นการเดินทางเพื่อทำงาน ท่องเที่ยว หรือการจัดการธุระส่วนตัวต่าง ๆ โดยเฉพาะในเมืองใหญ่หรือพื้นที่ที่มีการพัฒนาอย่างรวดเร็ว ความสามารถในการวางแผนการเดินทางที่มีประสิทธิภาพกลายเป็นปัจจัยสำคัญที่ช่วยให้ผู้คนสามารถใช้ชีวิตได้อย่างสะดวกและคล่องตัวมากยิ่งขึ้น ความต้องการพื้นฐานของผู้เดินทางในยุคดิจิทัลจึงไม่ได้มีเพียงแค่การไปถึงจุดหมายปลายทางเท่านั้น แต่ยังรวมถึงความสะดวกสบาย ความรวดเร็ว และการใช้เวลาอย่างคุ้มค่าในแต่ละช่วงของการเดินทางด้วย

นอกจากนี้ ผู้คนจำนวนไม่น้อยยังมีความจำเป็นในการเดินทางไปยังหลายจุดหมายปลายทางภายในระยะเวลาอันจำกัด เช่น การจัดแผนท่องเที่ยวแบบ one-day trip หรือการทำธุระหลายแห่งภายในวันเดียว ซึ่งทำให้การวางแผนเส้นทางมีความซับซ้อนและต้องพิจารณาปัจจัยหลายด้านพร้อมกัน ไม่ว่าจะเป็นเวลาเปิด-ปิดของสถานที่แต่ละแห่ง รูปแบบการเดินทางที่ใช้ เช่น การขับรถ การเดินเท้า หรือการใช้ขนส่งสาธารณะ รวมถึงการประเมินเวลาที่เหมาะสมในการอยู่แต่ละจุด และลำดับการเดินทางที่คุ้มค่าที่สุด ทั้งหมดนี้สะท้อนให้เห็นถึงความจำเป็นในการใช้เทคโนโลยีและระบบอัจฉริยะเพื่อช่วยจัดการการวางแผนการเดินทางให้เป็นไปอย่างแม่นยำและมีประสิทธิภาพ

1.1.2 ข้อจำกัดและปัญหาของการวางแผนเส้นทางแบบดั้งเดิม

แม้ว่าในปัจจุบันจะมีแอปพลิเคชันช่วยในการเดินทางหลากหลายรูปแบบ แต่หลายระบบยังคงมีข้อจำกัดด้านฟังก์ชันการใช้งาน เช่น การค้นหาสถานที่ที่สะดวกโดยไม่สามารถวางแผนแบบรวมหลายจุดหมายปลายทางพร้อมกันได้ ทำให้ผู้ใช้งานต้องใช้เวลาอันยาวนานในการรวบรวมข้อมูลด้วยตนเอง และวางแผนลำดับการเดินทางแยกจากกัน นอกจากนี้ยังขาดความสามารถในการปรับเปลี่ยนแผนอย่างยืดหยุ่นตามสถานการณ์จริง ซึ่งอาจส่งผลให้เกิดความไม่สะดวกในการเดินทางในแต่ละวัน

เครื่องมือที่มีอยู่จำนวนมากยังไม่สามารถวิเคราะห์ข้อมูลเชิงลึกในมิติต่าง ๆ พร้อมกันได้ เช่น การนำเวลาเปิด-ปิดของสถานที่ ระยะเวลาที่ใช้ในแต่ละจุด และรูปแบบการ

เดินทางเฉพาะบุคคลมาประมวลผลร่วมกันอย่างเป็นระบบ ความไม่แน่นอนของข้อมูล เช่น เวลาเปิด-ปิดที่อาจมีการเปลี่ยนแปลง หรือความคลาดเคลื่อนของการประเมินเวลาในการเดินทาง ส่งผลให้ผู้ใช้งานไม่สามารถวางแผนเส้นทางที่เหมาะสมและตอบโจทย์ความต้องการเฉพาะตัวได้อย่างเต็มที่ จึงเกิดความจำเป็นในการพัฒนาระบบที่สามารถจัดการข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพและแม่นยำ

1.2 วัตถุประสงค์

โครงการนี้มีเป้าหมาย เพื่อนำเสนอแนวทางในการประยุกต์ใช้เว็บแอปพลิเคชัน (Web Application) กับการวางแผนการเดินทาง มาใช้ในการพัฒนาระบบที่ส่งผลให้สามารถหาเส้นทางที่ดีที่สุดสำหรับแผนการเดินทางของบุคคล เพื่อให้บรรลุเป้าหมาย ดังกล่าว จึงกำหนดวัตถุประสงค์ของโครงการดังต่อไปนี้

- 1.2.1 เพื่อศึกษาและพัฒนาเว็บแอปพลิเคชันที่ช่วยประหยัดเวลาในการวางแผนการเดินทาง
- 1.2.2 เพื่อศึกษาและพัฒนาเว็บแอปพลิเคชันที่ช่วยจัดลำดับเส้นทางการเดินทางได้สอดคล้องกับเงื่อนไขที่ผู้ใช้งานกำหนด

1.3 ขอบเขตของโครงการ

โครงการนี้พัฒนาเป็นเว็บแอปพลิเคชัน (Web Application) เพื่อใช้สำหรับการวางแผนการเดินทางโดยมีขอบเขตดังนี้

- 1.3.1 เพื่อศึกษาและพัฒนาเว็บแอปพลิเคชันที่ช่วยประหยัดเวลาในการวางแผนการเดินทาง
- 1.3.2 เพื่อศึกษาและพัฒนาเว็บแอปพลิเคชันที่ช่วยจัดลำดับเส้นทางการเดินทางได้สอดคล้องกับเงื่อนไขที่ผู้ใช้งานกำหนด
- 1.3.3 เว็บแอปพลิเคชันนี้รองรับการแสดงผลแบบ Responsive Design เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพทั้งบนคอมพิวเตอร์ สมาร์ทโฟน และแท็บเล็ต
- 1.3.4 เว็บแอปพลิเคชันรองรับวิธีการเดินทางในรูปแบบ 1 วัน (one day trip) เท่านั้น

- 1.3.5 กลุ่มเป้าหมาย คือกลุ่มผู้ใช้งานทั่วไปที่ต้องการวางแผนการเดินทางแบบส่วนตัวหรือเพื่อการท่องเที่ยว และ ผู้ที่ต้องการระบบช่วยจัดการเวลาและเส้นทางที่เหมาะสมที่สุดในการเดินทาง
- 1.3.6 การประมวลผลขึ้นอยู่กับความเสถียรและความเร็วของการเชื่อมต่ออินเทอร์เน็ต รวมถึงความถูกต้องของข้อมูลจาก Google Maps API

1.4 ประโยชน์ของโครงการ

- 1.4.1 เว็บแอปพลิเคชันนี้ เป็นเครื่องมือที่ช่วยให้ผู้ที่ต้องการวางแผนการเดินทางสามารถประเมินระยะเวลาในการเดินทางได้ ส่งผลให้ลดเวลาที่ผู้ใช้งานต้องเสียไปในการคำนวณเส้นทางด้วยตัวเอง โดยประมวลผลข้อมูลทั้งหมดและนำเสนอแผนการเดินทางที่ชัดเจนและพร้อมใช้งาน
- 1.4.2 เว็บแอปพลิเคชันนี้ มีระบบช่วยจัดลำดับเส้นทางการเดินทางที่เหมาะสมที่สุด โดยคำนึงถึงเวลาเปิด-ปิดของสถานที่ ระยะเวลาเดินทาง และระยะเวลาที่ผู้ใช้งานต้องการใช้ในแต่ละสถานที่ ทำให้ผู้ใช้งานสามารถไปถึงจุดหมายปลายทางครบทุกจุดภายในเวลาที่กำหนด
- 1.4.3 เว็บแอปพลิเคชันนี้ เป็นเครื่องมือที่มีแจ้งเตือนข้อผิดพลาดของการวางแผนระบบสามารถแจ้งเตือนเมื่อมีสถานที่ที่ปิดทำการในวันที่เลือก หรือเมื่อแผนการเดินทางไม่สามารถสำเร็จได้ตามเวลาที่กำหนด ช่วยให้ผู้ใช้งานสามารถปรับแผนได้ล่วงหน้า

บทที่ 2

วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

2.1.1 การเดินทาง

2.1.1.1 แนวคิดพื้นฐานเกี่ยวกับการเดินทาง

การเดินทาง (Travel or Trip) หมายถึงการเคลื่อนย้ายจากจุดหนึ่งไปยังอีกจุดหนึ่งเพื่อวัตถุประสงค์ต่าง ๆ เช่น การทำงาน การพักผ่อน หรือการติดต่อธุรกิจ แนวคิดเรื่องการเดินทางครอบคลุมทั้งระยะทาง เวลา และประสบการณ์ที่เกิดขึ้นระหว่างการเคลื่อนที่ ซึ่งถือเป็นส่วนหนึ่งของกิจกรรมทางสังคมและเศรษฐกิจที่สำคัญของมนุษย์

จากมุมมองทางสังคมและเศรษฐกิจ การเดินทางช่วยสร้างความเชื่อมโยงระหว่างพื้นที่และผู้คน สนับสนุนการพัฒนาทางเศรษฐกิจผ่านการแลกเปลี่ยนสินค้า บริการ และวัฒนธรรม นอกจากนี้ยังเป็นเครื่องมือสำคัญในการกระจายทรัพยากรและลดความเหลื่อมล้ำทางภูมิศาสตร์ ทำให้เกิดแนวคิดที่ว่า การเดินทางเป็น “ฟังก์ชันของกิจกรรม” ที่สังคมต้องการ และการวางแผนเส้นทางหรือระบบการขนส่งจึงต้องตอบสนองต่อความต้องการนี้อย่างมีประสิทธิภาพ

2.1.1.2 ปัจจัยที่มีผลต่อการเดินทาง

(1) ระยะทางและเวลาเดินทาง: เป็นปัจจัยพื้นฐานที่มีผลต่อการตัดสินใจเดินทาง ผู้เดินทางมักจะเลือกเส้นทางที่ใช้เวลาน้อยหรือมีระยะทางสั้นที่สุดเพื่อลดต้นทุนและความเหนื่อยล้า แต่ในบางกรณีอาจมีข้อจำกัดที่ทำให้เลือกเส้นทางยาวขึ้น เช่น การหลีกเลี่ยงค่าผ่านทางหรือถนนที่แออัด

(2) ค่าใช้จ่าย: ค่าใช้จ่ายที่เกี่ยวข้องกับการเดินทาง เช่น ค่าน้ำมัน ค่าตัวโดยสาร หรือค่าผ่านทาง เป็นปัจจัยที่ผู้เดินทางต้องพิจารณา โดยเฉพาะกลุ่มที่มีงบประมาณจำกัด การเลือกวิธีเดินทางที่ประหยัดจึงมีความสำคัญมาก

(3) วัตถุประสงค์ของการเดินทาง: เหตุผลในการเดินทาง เช่น การไปทำงาน การไปโรงเรียน หรือการท่องเที่ยว จะมีผลต่อรูปแบบและความถี่ของการเดินทาง ตัวอย่างเช่น การเดินทางเพื่อทำงานมักมีความถี่สูงและเวลาแน่นอน ขณะที่การเดินทางเพื่อพักผ่อนอาจยืดหยุ่นและไม่ประจำ

(4) ลักษณะทางภูมิศาสตร์และสิ่งแวดล้อม: สภาพภูมิประเทศ ความพร้อมของโครงสร้างพื้นฐาน เช่น ถนน สถานีขนส่ง ระบบรถไฟ และปัจจัยด้านสภาพอากาศ ก็ส่งผลต่อรูปแบบการเดินทางและการเลือกเส้นทาง

(5) ส่วนบุคคลและสังคม: เพศ อายุ รายได้ และความชอบส่วนบุคคล เช่น การเดินทางแบบกลุ่มหรือคนเดียว รวมถึงปัจจัยทางวัฒนธรรม ก็มีอิทธิพลต่อพฤติกรรมการเดินทาง

2.1.1.3 ความสำคัญของการเดินทางในยุคปัจจุบัน

การเดินทางเป็นกิจกรรมพื้นฐานที่มีบทบาทสำคัญอย่างยิ่งในชีวิตประจำวันของมนุษย์ ไม่ว่าจะเป็นการเดินทางเพื่อทำงาน การเรียน การพักผ่อน หรือการดำเนินธุรกรรมต่าง ๆ ในสังคมสมัยใหม่ การเดินทางไม่เพียงแต่ช่วยเชื่อมโยงผู้คนกับสถานที่ต่าง ๆ เท่านั้น แต่ยังเป็นตัวกลางสำคัญที่ขับเคลื่อนเศรษฐกิจ สังคม และวัฒนธรรมของแต่ละประเทศ

(1) การเดินทางกับการพัฒนาเศรษฐกิจและสังคม

ในเชิงเศรษฐกิจ การเดินทางช่วยส่งเสริมการแลกเปลี่ยนสินค้าและบริการระหว่างพื้นที่ ช่วยเพิ่มโอกาสทางธุรกิจและการจ้างงาน รวมถึงสนับสนุนภาคการท่องเที่ยวซึ่งเป็นแหล่งรายได้หลักของหลายประเทศ การมีระบบการเดินทางที่ดีและมีประสิทธิภาพจะช่วยลดต้นทุนเวลาและค่าใช้จ่ายในการเคลื่อนย้าย ส่งผลให้ธุรกิจสามารถดำเนินงานได้รวดเร็วและมีประสิทธิภาพมากขึ้น

ในด้านสังคม การเดินทางเปิดโอกาสให้ผู้คนได้เข้าถึงโอกาสต่าง ๆ ไม่ว่าจะเป็นการศึกษา การรักษาพยาบาล หรือกิจกรรมทางสังคมอื่น ๆ ทำให้เกิดการพัฒนาคุณภาพชีวิตและสร้างความสัมพันธ์ระหว่างชุมชน การเดินทางที่สะดวกและปลอดภัยยังช่วยลดความเหลื่อมล้ำทางสังคม โดยเฉพาะในพื้นที่ชนบทหรือพื้นที่ที่ขาดแคลนบริการพื้นฐาน

(2) การเดินทางและการใช้ชีวิตในยุคปัจจุบัน

ในยุคดิจิทัลและโลกที่เชื่อมโยงกันอย่างรวดเร็ว การเดินทางยังคงเป็นปัจจัยสำคัญที่ช่วยเสริมสร้างความสะดวกสบายและประสิทธิภาพในการใช้ชีวิตประจำวันของผู้คน การเดินทางที่มีการวางแผนที่ดีช่วยให้ผู้ใช้สามารถบริหารเวลาได้อย่างเหมาะสม ลดความเครียดจากปัญหาการจราจรติดขัด และเพิ่มประสิทธิภาพในการทำงานหรือการเรียน

นอกจากนี้ การเดินทางยังมีบทบาทสำคัญในการส่งเสริมสุขภาพจิตและร่างกาย โดยเฉพาะอย่างยิ่งการเดินทางที่ผสมผสานการใช้เทคโนโลยีและการออกกำลังกาย เช่น การเดินหรือปั่นจักรยานในเมือง ซึ่งช่วยกระตุ้นให้เกิดกิจกรรมทางกาย ลดภาวะความเครียด และเพิ่มคุณภาพชีวิตโดยรวม

2.1.2 การวางแผนการเดินทาง

2.1.2.1 ความหมายของการวางแผนการเดินทาง

การวางแผนเส้นทางการเดินทางหมายถึงการกำหนดและจัดระเบียบเส้นทางที่เหมาะสมที่สุดในการเดินทางจากจุดเริ่มต้นไปยังจุดหมายปลายทาง โดยคำนึงถึงปัจจัยต่างๆ เช่น เวลา, ระยะทาง, วิธีการเดินทาง, สภาพการจราจร, ความสะดวกสบาย, และความปลอดภัย

การวางแผนเส้นทางการเดินทางอาจรวมถึงการเลือกใช้เส้นทางที่มีสถานที่สำคัญหรือจุดแวะพักที่น่าสนใจ รวมไปถึงการคำนวณเวลาที่จะใช้ในการเดินทาง และการเตรียมพร้อมสำหรับอุปสรรคที่อาจเกิดขึ้นระหว่างทาง

2.1.2.2 ประโยชน์สำหรับการวางแผนการเดินทาง

การวางแผนการเดินทางมีประโยชน์หลายประการ ซึ่งช่วยให้การเดินทางมีความสะดวกสบายและประหยัดเวลา รวมถึงลดความเสี่ยงในการเผชิญปัญหาหรืออุปสรรคที่อาจเกิดขึ้นระหว่างทาง ประโยชน์หลัก ๆ ได้แก่ :

- (1) ช่วยประหยัดเวลา : สามารถเลือกเส้นทางที่รวดเร็วที่สุด หรือหลีกเลี่ยงเส้นทางที่มีการจราจรหนาแน่น ลดเวลาที่สูญเสียไปกับการหลงทางหรือหยุดพักโดยไม่จำเป็น
- (2) ลดความเครียดและความกังวล: เมื่อมีแผนการเดินทางที่ชัดเจนและเตรียมตัวพร้อมแล้ว จะทำให้รู้สึกมั่นใจและไม่ต้องวิตกกังวลเกี่ยวกับสิ่งที่ต้องทำในระหว่างการเดินทาง
- (3) ประหยัดค่าใช้จ่าย : ช่วยให้สามารถคำนวณงบประมาณที่จำเป็นได้อย่างแม่นยำ
- (4) ช่วยให้มีเวลามากขึ้นสำหรับกิจกรรมที่สำคัญ : เมื่อการเดินทางมีการวางแผนที่ดี คุณจะมีเวลาเพียงพอในการทำกิจกรรมที่ต้องการ เช่น การเที่ยวชมสถานที่ท่องเที่ยว หรือการพักผ่อน

2.1.3 เว็บไซต์แอปพลิเคชัน (Web Application)

2.1.3.1 ความหมายของเว็บไซต์แอปพลิเคชัน (Web Application)

Web Application คือโปรแกรมหรือซอฟต์แวร์ที่ทำงานผ่าน เว็บเบราว์เซอร์ โดยไม่จำเป็นต้องติดตั้งบนเครื่องผู้ใช้งาน ถูกพัฒนาด้วยเทคโนโลยีที่ใช้บนเว็บ เช่น HTML, CSS, JavaScript รวมถึงใช้ Backend และ Database เพื่อจัดการข้อมูลและ

ประมวลผล โดยสามารถเข้าถึงได้ผ่านเครือข่ายอินเทอร์เน็ตหรือเครือข่ายในองค์กร (Intranet) ซึ่งก็คือเว็บไซต์ที่มีลักษณะเหมือนหน้าเว็บไซต์ทั่วไป แต่มีฟังก์ชันการทำงานที่หลากหลายโดยเป้าหมายคือการใช้งานให้ผู้ใช้งานได้ ‘ใช้งาน’ เว็บไซต์จริง ๆ เช่น Web Application สำหรับคิดเลข จับเวลา แปลภาษา หรือใช้แลกเปลี่ยนข้อมูลกับคนทั้งโลก เช่น เว็บไซต์ Social Media

2.1.3.2 ข้อดีของเว็บแอปพลิเคชัน (Web Application)

- (1) เข้าถึงได้จากทุกที่ทุกเวลา (ตราบดีที่มีอินเทอร์เน็ต)
- (2) ไม่ต้องกังวลเรื่องความเข้ากันได้กับระบบปฏิบัติการ
- (3) ลดภาระในการติดตั้งและบำรุงรักษา

2.1.4 แนวคิด Path Optimization Problem

2.1.4.1 ความหมายของ Path Optimization Problem

ปัญหาการค้นหาเส้นทางที่เหมาะสมที่สุดหรือเส้นทางที่มีประสิทธิภาพสูงสุดภายใต้ข้อกำหนดหรือข้อจำกัดต่างๆ โดยปัญหานี้มักปรากฏในหลายบริบท เช่น การขนส่ง, การวางแผนการเดินทาง, การกระจายสินค้า, และการวางแผนเครือข่ายคอมพิวเตอร์

2.1.4.2 แนวคิดของ Path Optimization Problem

Path Optimization Problem มีเป้าหมายในการหาเส้นทางที่ดีที่สุดตามเกณฑ์ที่กำหนด เช่น:

- (1) เส้นทางที่สั้นที่สุด (Shortest Path): ลดระยะทางหรือเวลาการเดินทางให้น้อยที่สุด
- (2) เส้นทางที่มีต้นทุนต่ำที่สุด (Cost Optimization): ลดต้นทุน เช่น ค่าน้ำมัน, ค่าผ่านทาง, หรือค่าใช้จ่ายอื่นๆ
- (3) เส้นทางที่มีประสิทธิภาพที่สุด: อาจพิจารณาทั้งเวลา, ระยะทาง, และต้นทุนรวมกัน
- (4) เส้นทางที่เหมาะสมตามข้อจำกัด (Constraints): เช่น เวลาทำการของสถานที่, การหลีกเลี่ยงจุดติดขัด, หรือข้อจำกัดทางทรัพยากร

2.1.4.3 ประเภทของ Path Optimization Problem

- (1) Shortest Path Problem
 - หาทางเดินที่มีระยะทางสั้นที่สุดจากจุดเริ่มต้นไปยังจุดปลายทาง
 - ตัวอย่าง: การหาทางเดินในแผนที่

(2) Traveling Salesman Problem (TSP)

- หาลำดับเส้นทางที่สั้นที่สุดเพื่อเยี่ยมชมจุดทั้งหมดและกลับมาที่จุดเริ่มต้น
- ใช้ในโลจิสติกส์และการขนส่ง

(3) Constrained Shortest Path Problem

- หาทางเดินที่เหมาะสมที่สุดโดยมีข้อจำกัดเพิ่มเติม เช่น เวลาเปิด-ปิดสถานที่ ระยะเวลาที่ต้องใช้ในจุดแวะพัก

(4) Multi-Criteria Optimization

- หาทางเดินที่เหมาะสมที่สุดภายใต้เป้าหมายหลายอย่างพร้อมกัน เช่น ค่าใช้จ่าย ระยะทาง และความเร็ว

2.1.4.4 ตัวอย่างการแก้ปัญหาในโรงงาน

ในโรงงานวางแผนการเดินทางที่กำลังพัฒนา Path Optimization Problem สามารถใช้ในการ:

- (1) จัดลำดับเส้นทาง: กำหนดว่าควรไปสถานที่ใดก่อน โดยคำนึงถึงข้อจำกัด เช่น
 - เวลาเปิด-ปิดของสถานที่
 - ระยะเวลาที่ผู้ใช้งานต้องการใช้ในแต่ละจุด
 - ระยะทางหรือเวลาที่ใช้ในการเดินทางระหว่างสถานที่
- (2) เลือกเส้นทางที่ดีที่สุด: เปรียบเทียบเส้นทางที่เป็นไปได้หลายเส้นทาง เพื่อหาทางเลือกที่ใช้เวลาหรือระยะทางน้อยที่สุด
- (3) คำนึงถึงข้อจำกัดเวลา: ต้องสามารถไปถึงสถานที่ทันเวลาเปิดทำการ

2.1.5 แนวคิด Shortest Path with Constraints**2.1.5.1 ความหมายของ Shortest Path with Constraints**

เป็นปัญหาที่ต้องการค้นหาเส้นทางที่ใช้ทรัพยากรน้อยที่สุด (เช่น ระยะทาง, เวลาเดินทาง, ค่าใช้จ่าย) จากจุดเริ่มต้นไปยังจุดหมาย โดยมีเงื่อนไขหรือข้อจำกัดเพิ่มเติมที่ต้องปฏิบัติตามระหว่างการค้นหาเส้นทาง เช่น:

- (1) เวลาเปิด-ปิดของสถานที่: สถานที่บางแห่งอาจมีเวลาให้บริการที่จำกัด
- (2) ลำดับการเยี่ยมชม: สถานที่บางแห่งอาจต้องไปก่อนหรือหลังอีกสถานที่หนึ่ง
- (3) ทรัพยากรจำกัด: เช่น น้ำมัน, เวลาเดินทางสูงสุด, หรือจำนวนสถานที่ที่สามารถเยี่ยมชมได้
- (4) เงื่อนไขเฉพาะอื่นๆ: เช่น หลีกเลี่ยงถนนที่มีค่าทางด่วน หรือเส้นทางที่ไม่ปลอดภัย

2.1.5.2 ข้อจำกัดของ Shortest Path with Constraints ในโครงงานนี้

- (1) สถานที่แต่ละแห่งมีเวลาเปิด-ปิดที่ต้องปฏิบัติตาม
- (2) ผู้ใช้งานกำหนดเวลาที่ต้องการอยู่ในแต่ละสถานที่
- (3) เส้นทางต้องเหมาะสมกับเวลาที่เริ่มต้นและลำดับการเดินทาง
- (4) ต้องประมวลผลเพื่อหาวิธีที่ทำให้ผู้ใช้งานสามารถเยี่ยมชมสถานที่ทั้งหมดได้ภายในระยะเวลาที่กำหนด

2.1.6 ทฤษฎีการเลือกเส้นทาง (Route Choice Theory)

ทฤษฎีการเลือกเส้นทาง เป็นทฤษฎีที่อธิบายถึงกระบวนการและปัจจัยที่มีผลต่อการตัดสินใจเลือกเส้นทางในการเดินทางของผู้คน ซึ่งเป็นหัวใจสำคัญในการวางแผนเส้นทาง ไม่ว่าจะเป็นการเดินทางโดยยานพาหนะส่วนตัว ระบบขนส่งสาธารณะ หรือการเดินทางเท้า โดยทฤษฎีนี้มีพื้นฐานจากการวิเคราะห์พฤติกรรมของผู้เดินทาง (travelers) ในการตัดสินใจเลือกเส้นทางที่เหมาะสมที่สุดตามเป้าหมายของตนเอง

ทฤษฎีการเลือกเส้นทางเป็นพื้นฐานสำคัญในวิชาการด้านวิศวกรรมจราจรและภูมิสารสนเทศ (GIS) โดยอธิบายพฤติกรรมของผู้เดินทางในการตัดสินใจเลือกเส้นทางหนึ่ง ๆ จากจุดเริ่มต้นไปยังจุดหมายปลายทาง โดยมีปัจจัยหลายประการที่มีอิทธิพลต่อการตัดสินใจ เช่น ความยาวของเส้นทาง เวลาที่ใช้ในการเดินทาง ความสะดวกสบาย ความปลอดภัย และต้นทุนที่เกี่ยวข้อง เช่น ค่าผ่านทางหรือค่าน้ำมัน

สำหรับระบบ PlanMyTrip ทฤษฎีนี้ถูกนำมาใช้ผ่านการประมวลผลเส้นทางโดยอิงกับข้อมูลจริงจาก Google Maps Directions API รวมถึงการพิจารณา “เวลารวม” ที่ใช้ในการเดินทางเป็นตัวชี้วัดหลักในการจัดลำดับเส้นทาง ระบบยังรองรับการตัดสินใจของผู้ใช้ในมิติต่าง ๆ เช่น การเลือกหลีกเลี่ยงค่าผ่านทาง หรือการกำหนดวิธีการเดินทางที่ต้องการ เช่น เดินเท้าหรือรถยนต์ ซึ่งเป็นการเปิดโอกาสให้ผู้ใช้งานสามารถเลือกเส้นทางที่เหมาะสมกับบริบทของตนเองมากที่สุด ถือเป็นการประยุกต์ใช้ทฤษฎีนี้ในบริบทของระบบอัจฉริยะ

2.1.6.1 แนวคิดหลักของทฤษฎีการเลือกเส้นทาง (Route Choice Theory)

โดยทั่วไป การเลือกเส้นทางเป็นการแก้ไขปัญหาคับข้องเกี่ยวกับการหาค่า “ความเหมาะสม” หรือ “ประสิทธิภาพ” ของเส้นทางหนึ่ง ๆ ซึ่งขึ้นอยู่กับปัจจัยหลายด้าน เช่น ระยะทาง เวลาใช้เดินทาง ค่าใช้จ่าย ความปลอดภัย ความสะดวกสบาย และปัจจัยเฉพาะอื่น ๆ เช่น การเลี่ยงค่าผ่านทางหรือถนนที่มีการจราจรหนาแน่น ทฤษฎีการเลือกเส้นทางจึงเป็นส่วนสำคัญที่ช่วยอธิบายว่าทำไมผู้เดินทางจึงเลือกเส้นทางใดเส้นทางหนึ่งแทนที่จะเป็นเส้นทางอื่น

นอกจากนี้ ผู้เดินทางแต่ละคนมีความชอบและข้อจำกัดต่างกัน เช่น บางคนอาจเน้นการเดินทางที่เร็วที่สุด ขณะที่บางคนอาจเลือกเส้นทางที่สบายงามหรือปลอดภัยกว่า ทำให้ระบบวางแผนเส้นทางต้องสามารถรองรับรูปแบบการเลือกเส้นทางที่หลากหลาย และปรับเปลี่ยนตามลักษณะของผู้ใช้ได้

2.1.6.2 การนำทฤษฎีไปใช้ในระบบวางแผนเส้นทาง

ทฤษฎีการเลือกเส้นทางถูกนำมาใช้เพื่อสร้างแบบจำลองและอัลกอริทึมที่สามารถประมวลผลข้อมูลเส้นทางที่เป็นไปได้ทั้งหมด และเลือกเส้นทางที่เหมาะสมที่สุดตามเงื่อนไขที่ตั้งไว้ เช่น ระยะทางน้อยที่สุด เวลาเดินทางรวดเร็วที่สุด หลีกเลี่ยงถนนที่มีค่าผ่านทาง หรือแม้กระทั่งการจัดลำดับสถานที่ที่ต้องแวะในกรณีเดินทางหลายจุด

อัลกอริทึมยอดนิยมที่มักใช้ในกระบวนการเลือกเส้นทาง เช่น Dijkstra's Algorithm, A* Algorithm, หรือ Genetic Algorithm จะประมวลผลข้อมูลเส้นทางโดยอิงกับข้อมูลพารามิเตอร์ต่าง ๆ ที่ผู้ใช้กำหนด และสภาพแวดล้อมในปัจจุบัน เช่น เวลาทำการของสถานที่ ปริมาณจราจรที่คาดการณ์ไว้ หรือข้อจำกัดอื่น ๆ ที่เกี่ยวข้อง

2.1.6.3 ปัจจัยที่มีผลต่อการเลือกเส้นทาง

(1) ระยะทางและเวลาเดินทาง: ผู้เดินทางส่วนใหญ่จะเลือกเส้นทางที่มีระยะทางสั้นที่สุดหรือใช้เวลาน้อยที่สุด เพื่อประหยัดเวลาและต้นทุนเชื้อเพลิง แต่ในบางกรณี อาจเลือกเส้นทางที่ใช้เวลานานกว่าเพื่อหลีกเลี่ยงสภาพการจราจรติดขัด หรือถนนที่มีอุบัติเหตุบ่อยครั้งผู้ใช้งานกำหนดเวลาที่ต้องการอยู่ในแต่ละสถานที่

(2) ค่าใช้จ่าย: การเลือกเส้นทางที่หลีกเลี่ยงค่าผ่านทาง หรือเส้นทางที่ลดค่าใช้จ่ายด้านน้ำมันหรือค่ารถโดยสาร เป็นปัจจัยที่มีผลโดยตรงต่อต้นทุนในการเดินทาง

(3) เงื่อนไขเฉพาะของผู้เดินทาง: เช่น เวลาทำการของสถานที่ที่ต้องแวะ เวลาที่ต้องการถึงจุดหมาย รวมถึงข้อจำกัดเฉพาะ เช่น ต้องแวะรับ-ส่งบุคคลบางคน หรือขนส่งสินค้าบางประเภท

2.1.7 ทฤษฎีอัลกอริทึม A* (A-Star Algorithm Theory)

2.1.7.1 ความหมายและที่มาของอัลกอริทึม A* (A-Star Algorithm Theory)

อัลกอริทึม A* หรือ A-Star เป็นหนึ่งในอัลกอริทึมค้นหาเส้นทาง (Pathfinding Algorithm) ที่ได้รับความนิยมอย่างแพร่หลายในด้านวิทยาการคอมพิวเตอร์ ปัญญาประดิษฐ์ และการคำนวณระยะทางในการเดินทาง โดยถูกพัฒนาขึ้นในปี ค.ศ. 1968 โดย Peter Hart, Nils Nilsson และ Bertram Raphael จากสถาบัน Stanford Research Institute

จุดเด่นของอัลกอริทึม A* คือการผสมผสานจุดแข็งของ Dijkstra's Algorithm (ที่เน้นการค้นหาเส้นทางที่มีต้นทุนน้อยที่สุด) กับแนวคิดของ Greedy Best-First Search (ที่เน้นค้นหาเป้าหมายเร็วที่สุด) ด้วยการใช้ฟังก์ชันประมาณค่า (Heuristic Function) เพื่อคาดการณ์ค่าระยะทางหรือค่าใช้จ่ายจากจุดปัจจุบันไปถึงจุดเป้าหมาย

2.1.7.2 หลักการพื้นฐานของอัลกอริทึม A*

A* พยายามหาเส้นทางจากจุดเริ่มต้นไปยังจุดหมายที่มี “ต้นทุนรวมน้อยที่สุด” โดยใช้ฟังก์ชันประเมินดังนี้:

$$f(n) = g(n) + h(n)$$

โดยที่:

$f(n)$: ต้นทุนรวมของโหนด n

$g(n)$: ต้นทุนจริงจากจุดเริ่มต้นมาถึงโหนด n

$h(n)$: ฟังก์ชันประมาณค่า (Heuristic) จากโหนด n ไปยังเป้าหมาย

จุดสำคัญ คือ $h(n)$ ต้อง “ไม่ประเมินเกินจริง” (Admissible) เพื่อให้ A^* ทำงานได้อย่างถูกต้อง กล่าวคือ:

$$h(n) \leq h^*(n)$$

โดย $h^*(n)$ คือค่าจริงจาก n ไปยังเป้าหมาย

2.1.7.3 ขั้นตอนการทำงานของอัลกอริทึม A^*

- (1) กำหนดระยะทางจากจุดเริ่มต้นไปยังตัวเองเป็น 0 และจุดอื่นเป็นอนันต์
- (2) ทำเครื่องหมายจุดเริ่มต้นเป็น “จุดที่เยี่ยมชมแล้ว” (visited)
- (3) สำหรับทุกจุดที่เชื่อมกับจุดที่เยี่ยมชมแล้ว ให้คำนวณระยะทางใหม่ หากระยะทางใหม่นั้นสั้นกว่าค่าปัจจุบัน ให้แทนที่ด้วยค่าที่ต่ำกว่า
- (4) เลือกจุดที่ยังไม่เยี่ยมชมซึ่งมีระยะทางน้อยที่สุดเพื่อเป็นจุดพิจารณาต่อไป
- (5) ทำซ้ำขั้นตอนนี้จนกว่าจะเยี่ยมชมจุดหมายปลายทางหรือเยี่ยมชมครบทุกจุด

2.2 เทคโนโลยีที่เกี่ยวข้อง

2.2.1 Figma

2.2.1.1 ความหมายของ Figma

Figma คือเครื่องมือออกแบบ UI/UX ที่ใช้สำหรับการออกแบบส่วนติดต่อผู้ใช้ (User Interface) และประสบการณ์ผู้ใช้ (User Experience) แบบออนไลน์ ซึ่งช่วยให้ทีมออกแบบสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพในเวลาเดียวกัน โดยไม่ต้องติดตั้งซอฟต์แวร์ใด ๆ บนเครื่องคอมพิวเตอร์ เนื่องจากเป็นเครื่องมือที่ทำงานบนเว็บเบราว์เซอร์



ภาพที่ 2.1 แสดงสัญลักษณ์ของภาษา Figma (Figma LOGO)

2.2.1.2 คุณสมบัติของ Figma

- (1) Figma ช่วยในการออกแบบอินเตอร์เฟซของเว็บไซต์หรือแอปพลิเคชัน โดยให้ผู้ใช้สามารถสร้างโปรโตไทป์และ wireframes (โครงร่างของหน้าจอ) ได้
- (2) ทีมออกแบบสามารถทำงานร่วมกันในโปรเจกต์เดียวกันได้แบบเรียลไทม์ ซึ่งเป็นการช่วยให้ทีมออกแบบมีความคล่องตัวและสามารถแก้ไขงานหรือเพิ่มความเห็นได้ทันที
- (3) Figma มีฟีเจอร์การแชร์งานและคอมเมนต์ โดยผู้ที่ได้รับอนุญาตสามารถดูหรือคอมเมนต์การออกแบบได้
- (4) สามารถสร้างโปรโตไทป์ (การจำลองการใช้งานจริง) เพื่อทดสอบการทำงานของฟีเจอร์ที่ออกแบบในรูปแบบที่เหมือนแอปพลิเคชันหรือเว็บไซต์จริง

2.2.2 HTML5

2.2.2.1 ความหมายของ HTML

HTML5 คือเวอร์ชันล่าสุดของ Hypertext Markup Language (HTML) ซึ่งเป็นภาษาที่ใช้ในการสร้างโครงสร้างและเนื้อหาของหน้าเว็บ เช่น การสร้างฟอร์มกรอกข้อมูลต่าง ๆ ที่ผู้ใช้จะใช้งานโดย HTML5 ได้รับการพัฒนาให้สามารถรองรับการทำงานที่ซับซ้อนมากขึ้นรวมถึงการรองรับมัลติมีเดีย, การจัดการข้อมูล, และการโต้ตอบกับผู้ใช้ได้ดีขึ้นมากกว่าเวอร์ชันก่อนหน้า



ภาพที่ 2.2 แสดงสัญลักษณ์ของภาษา HTML5 (HTML5 LOGO)

2.2.2.2 คุณสมบัติหลักของ HTML5

- (1) รองรับมัลติมีเดีย:
 - HTML5 รองรับการฝังสื่อเช่น วิดีโอ (<video>) และเสียง (<audio>) โดยไม่ต้องใช้ปลั๊กอินเสริม (เช่น Flash)
- (2) การจัดการข้อมูล:
 - HTML5 รองรับการเก็บข้อมูลใน Web Storage เช่น localStorage และ sessionStorage ซึ่งช่วยให้สามารถเก็บข้อมูลชั่วคราวในเบราว์เซอร์ได้
- (3) รองรับการพัฒนาเว็บแอปพลิเคชันที่มีการโต้ตอบ:
 - HTML5 รองรับการใช้งาน canvas และ SVG (Scalable Vector Graphics) เพื่อวาดกราฟิกและแสดงผลข้อมูลต่าง ๆ บนหน้าเว็บ

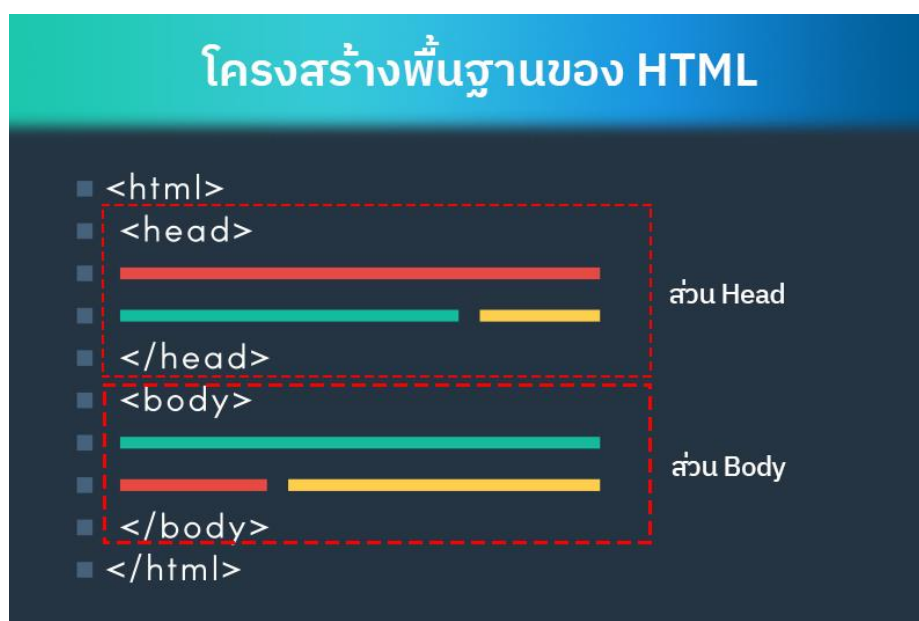
- รองรับ Geolocation API ซึ่งใช้ในการระบุตำแหน่งของผู้ใช้ (ใช้ในกรณีที่ต้องการให้ผู้ใช้เลือกจุดเริ่มต้นในการเดินทาง)

(4) ปรับปรุงการทำงานกับฟอร์ม:

- HTML5 เพิ่มการควบคุมฟอร์มต่าง ๆ ที่ดีกว่า เช่น การตรวจสอบข้อมูลด้วยประเภทฟอร์ม (email, number, date ฯลฯ) และการจัดการฟอร์มที่ง่ายขึ้น

(5) รองรับการทำงานแบบออฟไลน์:

- HTML5 รองรับ Application Cache และ Service Workers ซึ่งช่วยให้เว็บแอปพลิเคชันสามารถทำงานแบบออฟไลน์ได้



ภาพที่ 2.3 แสดงสัญลักษณ์ของภาษา HTML5 (HTML5 LOGO)

2.2.3 CSS3

2.2.3.1 ความหมายของ CSS3

CSS3 คือเวอร์ชันล่าสุดของ Cascading Style Sheets (CSS) ซึ่งเป็นภาษาที่ใช้สำหรับการจัดการลักษณะการแสดงผล (Presentation) ของเอกสาร HTML หรือ XML โดยเน้นการออกแบบรูปลักษณ์ เช่น สี ตัวอักษร ขนาด ระยะห่าง การจัดวาง และการเคลื่อนไหวต่าง ๆ เพื่อให้หน้าเว็บหรือแอปพลิเคชันดูสวยงามและตอบสนองต่อผู้ใช้ได้ดี



ภาพที่ 2.4 แสดงสัญลักษณ์ของภาษา CSS3 (CSS3 LOGO)

2.2.3.2 คุณสมบัติของ CSS3

- (1) โมดูล (Modules) การแยกเป็นโมดูลช่วยให้นักพัฒนาเลือกใช้เฉพาะส่วนที่ต้องการได้สะดวก:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Animations and Transitions
- (2) การเพิ่มความสามารถในการออกแบบ:
 - Border Radius: สร้างมุมโค้งให้กับกล่องหรือปุ่ม

- Gradients: ใส่สีแบบไล่ระดับให้พื้นหลัง
- Shadows: สร้างเงากับข้อความหรือกล่อง
- Flexbox และ Grid: สำหรับการจัดวางองค์ประกอบบนหน้าเว็บให้ยืดหยุ่น

(3) การเพิ่ม Animation และ Transition:

- ใช้สำหรับสร้างความเคลื่อนไหว เช่น การเปลี่ยนสี การแสดง/ซ่อนองค์ประกอบ หรือสร้างเอฟเฟกต์ที่น่าสนใจ

(4) Media Queries:

- ช่วยสร้างหน้าเว็บแบบ Responsive Design ซึ่งสามารถปรับเปลี่ยนการแสดงผลตามขนาดหน้าจอหรืออุปกรณ์ที่ใช้

2.2.4 Bootstrap

2.2.4.1 ความหมายของ Bootstrap

Bootstrap คือ Frontend Framework ที่เป็นเครื่องมือช่วยในการออกแบบและพัฒนาเว็บไซต์หรือเว็บแอปพลิเคชันให้มีความสวยงาม และรองรับการแสดงผลแบบ Responsive Design โดยไม่ต้องเขียน CSS หรือ JavaScript ขึ้นมาใหม่ทั้งหมด เป็น Framework ที่ได้รับความนิยมอย่างมากในวงการพัฒนาเว็บ เนื่องจากใช้งานง่ายและมีฟีเจอร์ที่ครอบคลุมการออกแบบพื้นฐาน



ภาพที่ 2.5 แสดงสัญลักษณ์ Bootstrap (Bootstrap LOGO)

2.2.4.2 คุณสมบัติของ Bootstrap

- (1) Responsive Design: ใช้ระบบ Grid System ที่ช่วยในการจัดวางเลย์เอาต์ให้เหมาะสมกับทุกขนาดหน้าจอ เช่น มือถือ แท็บเล็ต หรือคอมพิวเตอร์
- (2) Component ที่ใช้งานง่าย: มีองค์ประกอบที่เตรียมไว้ให้ใช้งาน เช่น Navbar, Buttons, Forms, Modals, Cards, Dropdowns เป็นต้น
- (3) ปรับแต่งได้ง่าย: สามารถปรับแต่ง CSS หรือใช้ Bootstrap Utilities เพื่อแก้ไขรูปลักษณะได้อย่างรวดเร็ว
- (4) รวม JavaScript Components: รองรับการใช้งาน JavaScript Components ที่พัฒนาไว้ล่วงหน้า เช่น Carousel, Modal, Tabs, และ Alerts
- (5) ความเข้ากันได้กับเบราว์เซอร์: ออกแบบมาให้รองรับเบราว์เซอร์ส่วนใหญ่ เช่น Chrome, Safari
- (6) ซิมและไอคอน: มีซิมสำเร็จรูปและชุดไอคอนที่ช่วยให้การออกแบบง่ายขึ้น

2.2.5 JavaScript

2.2.5.1 ความหมายของ JavaScript

คือ ภาษาการเขียนโปรแกรมแบบไดนามิก ที่ใช้ในการพัฒนาเว็บเพื่อสร้างฟังก์ชันการทำงานบนฝั่งผู้ใช้ (Client-side) และฝั่งเซิร์ฟเวอร์ (Server-side) โดยเฉพาะในส่วน Frontend ที่ทำให้เว็บไซต์สามารถตอบสนองต่อผู้ใช้งานได้แบบเรียลไทม์ เช่น การกดปุ่ม การกรอกแบบฟอร์ม หรือการแสดงผลข้อมูลโดยไม่ต้องโหลดหน้าใหม่ทั้งหมด เพิ่มความสามารถและการโต้ตอบในหน้าเว็บ (Interactive Web Applications) เช่น การตอบสนองกับผู้ใช้ในทันที (Real-time interactions) การจัดการ DOM (Document Object Model) และการสื่อสารกับ API เพื่อดึงข้อมูลจากเซิร์ฟเวอร์ โดย JavaScript เป็นส่วนสำคัญใน **Frontend** ในการทำเว็บแอปพลิเคชันสำหรับวางแผนการเดินทาง เนื่องจากต้องการให้เว็บแอปพลิเคชันมีฟังก์ชันที่โต้ตอบกับผู้ใช้งานในทันที เช่น การแนะนำสถานที่ (Autocomplete) การตรวจสอบเงื่อนไข เช่น เวลาเปิด-ปิด และการแสดงผลข้อมูลเส้นทางที่เหมาะสมที่สุด



ภาพที่ 2.6 แสดงสัญลักษณ์ของภาษา Javascript5 (Javascript5 LOGO)

2.2.5.2 คุณสมบัติของ JavaScript

- (1) Dynamic and Lightweight: เขียนโค้ดได้ยืดหยุ่นและสามารถทำงานร่วมกับ HTML และ CSS ได้ง่าย
- (2) Event-driven: รองรับการทำงานที่ตอบสนองกับการกระทำของผู้ใช้งาน เช่น การคลิกหรือการเลื่อนหน้า
- (3) Asynchronous Processing: จัดการงานหลายงานพร้อมกัน เช่น การดึงข้อมูลจาก API โดยไม่ทำให้หน้าเว็บหยุดทำงาน
- (4) Cross-platform: ทำงานได้บนทุกเบราว์เซอร์และระบบปฏิบัติการ
- (5) Extensible: ใช้งานร่วมกับ Framework หรือ Library เช่น React, Vue, และ Angular

2.2.6 React

2.2.6.1 ความหมายของ React

React เป็นไลบรารี (Library) สำหรับการพัฒนา Frontend User Interface (UI) ที่สร้างโดย Facebook เน้นการสร้าง Component-Based Architecture ซึ่งทำให้การออกแบบและพัฒนา UI เป็นระบบระเบียบ มีประสิทธิภาพ และสามารถนำกลับมาใช้ซ้ำได้ง่าย โดย React เหมาะสำหรับการสร้าง Single Page Applications (SPAs) ที่ต้องการความเร็วในการตอบสนองและการแสดงผลข้อมูลแบบไดนามิก



ภาพที่ 2.7 แสดงสัญลักษณ์ของ React (React LOGO)

2.2.6.2 คุณสมบัติเด่นของ React

- (1) Component-Based Architecture โดย React แบ่งส่วน UI ออกเป็นชิ้นส่วนเล็กๆ (Components) ที่สามารถพัฒนา ทดสอบ และใช้งานแยกกันได้
- (2) Virtual DOM โดย React ใช้ Virtual DOM ในการอัปเดตเฉพาะส่วนที่เปลี่ยนแปลงในหน้าเว็บ ทำให้การแสดงผลเร็วขึ้นเมื่อเทียบกับการอัปเดต DOM จริง
- (3) Declarative Syntax โดย React ใช้แนวทางเขียนโค้ดแบบประกาศ (Declarative) ซึ่งช่วยให้นักพัฒนากำหนดว่า UI ควรเป็นอย่างไร แทนที่จะบอกวิธีการทำ
- (4) One-Way Data Binding โดย React มีการจัดการข้อมูลแบบทิศทางเดียว ทำให้ง่ายต่อการตรวจสอบข้อผิดพลาดและการควบคุมข้อมูล
- (5) Rich Ecosystem โดย React มีไลบรารีเสริม เช่น React Router, Redux, และเครื่องมือที่ช่วยในการพัฒนาอย่าง React Developer Tools

2.2.7 Node.js

2.2.7.1 ความหมายของ Node.js

Node.js คือแพลตฟอร์มโอเพ่นซอร์สที่ใช้รันโค้ด JavaScript ฝั่ง Server-side ซึ่งพัฒนาขึ้นโดย Ryan Dahl ในปี 2009 และสร้างขึ้นบน V8 JavaScript Engine (เอนจินที่ Google ใช้ใน Chrome) โดย Node.js ถูกออกแบบมาให้สามารถจัดการงาน I/O (Input/Output) ได้อย่างมีประสิทธิภาพผ่าน event-driven architecture และ

non-blocking I/O model ทำให้เหมาะกับงานที่ต้องประมวลผลพร้อมกันจำนวนมาก เช่น เว็บเซิร์ฟเวอร์, API, หรือแอปพลิเคชันเรียลไทม์

2.2.7.2 คุณสมบัติหลักของ Node.js

(1) Event-Driven Architecture

- Node.js ใช้แนวคิด **Event Loop** เพื่อจัดการคำสั่งต่างๆ แบบไม่ต้องรอคำสั่งก่อนหน้าทำงานเสร็จ
- เหมาะสำหรับงานที่มีการประมวลผลพร้อมกันจำนวนมาก เช่น การเรียก API, การอ่าน/เขียนไฟล์

(2) Non-Blocking I/O

- งาน I/O เช่น อ่านไฟล์หรือเรียกฐานข้อมูลจะถูกดำเนินการในแบ็กกราวด์ โดยไม่รบกวนการทำงานของคำสั่งอื่นๆ

(3) Single Programming Language

- ใช้ JavaScript ได้ทั้งฝั่ง Frontend และ Backend

(4) Cross-Platform

- สามารถใช้งานได้หลายระบบปฏิบัติการ เช่น Windows, macOS, และ Linux

(5) มี Package Manager (npm)

- Node.js มาพร้อมกับ npm (Node Package Manager) ซึ่งช่วยให้พัฒนาแอปได้เร็วขึ้น ด้วยไลบรารีและแพ็คเกจที่พร้อมใช้งานมากมาย

2.2.7.3 การทำงานของ Node.js

Node.js ใช้ Single-Threaded Event Loop ในการจัดการคำสั่ง โดยไม่สร้างหลาย Thread ขึ้นมาเหมือนภาษาอื่นๆ เช่น Java หรือ PHP กระบวนการทำงาน:

- (1) รับคำขอ (Request) จาก Client
- (2) ตรวจสอบคำขอ และส่งไปยัง Event Queue
- (3) ใช้ Event Loop จัดการคำสั่งแบบ Asynchronous (ไม่รอให้คำสั่งเสร็จทีละตัว)
- (4) ตอบกลับ Client หลังจากคำขอนั้นเสร็จสิ้น

2.2.8 Express.js

2.2.8.1 ความหมายของ Express.js

Express.js เป็น เว็บเฟรมเวิร์ค (Web Framework) สำหรับ Node.js ที่ใช้ในการสร้าง เว็บแอปพลิเคชัน และ API อย่างรวดเร็วและง่ายดาย โดยเป็นเฟรมเวิร์ค ระดับเบา (lightweight) ที่ทำให้ Node.js สามารถจัดการกับการรับส่งข้อมูล (HTTP requests/responses) ได้ง่ายขึ้น



ภาพที่ 2.8 แสดงสัญลักษณ์ของ Express JS (Express LOGO)

2.2.8.2 คุณสมบัติเด่นของ Express.js

- (1) มีโครงสร้างเรียบง่าย ใช้งานสะดวกสำหรับการสร้างแอปพลิเคชัน และ API
- (2) สามารถใช้ middleware เพื่อตรวจสอบหรือแก้ไขข้อมูลก่อนส่งถึงฟังก์ชันที่เกี่ยวข้อง
- (3) มีระบบการจัดการเส้นทาง (Routing) ที่ยืดหยุ่นและทรงพลัง
- (4) รองรับ RESTful APIs ช่วยให้การสร้างและจัดการ API เป็นเรื่องง่าย
- (5) รองรับการติดตั้งแพ็คเกจเสริมจาก npm เพื่อเพิ่มฟังก์ชันการทำงาน

2.2.9 Google Maps Platform

2.2.9.1 ความหมายของ Google Maps Platform

Google Maps Platform คือชุดของบริการและเครื่องมือที่พัฒนาโดย Google สำหรับนักพัฒนา (Developers) เพื่อผสมผสานความสามารถของ Google Maps เข้ากับแอปพลิเคชันหรือเว็บไซต์ของตนเอง โดย API เหล่านี้ช่วยให้ผู้พัฒนาสามารถเพิ่มแผนที่ แสดงตำแหน่ง หรือจัดการข้อมูลแผนที่ได้หลากหลายรูปแบบ เช่น ค้นหาเส้นทาง , แสดงสถานที่ใกล้เคียง, และอื่นๆ



Google Maps Platform

ภาพที่ 2.9 แสดงสัญลักษณ์ของ Google Maps Platform

2.2.9.2 ประเภทของ Google Maps Platform

- (1) Maps ใช้สำหรับแสดงแผนที่ในแอปหรือเว็บ เช่น แผนที่ถนน, แผนที่ภาพถ่ายดาวเทียม, Streetview เป็นต้น
- (2) Places ใช้สำหรับค้นหาสถานที่ เช่น รับรายละเอียดสถานที่, แสดงผล autocomplete, รีวิวจากผู้ใช้งาน เป็นต้น
- (3) Routes สำหรับการคำนวณเส้นทางต่าง ๆ เช่น คำนวณระยะทาง, การเดินทางตามประเภทการเดินทาง, พิจารณาการจราจร เป็นต้น

2.3 แอปพลิเคชันที่เกี่ยวข้อง

2.3.1 Google Maps

2.3.1.1 ฟังก์ชันการทำงาน:

- (1) การค้นหาสถานที่ (Place Search)
 - ค้นหาสถานที่สำคัญ ร้านค้า หรือจุดสนใจ (POI - Point of Interest)
 - มีระบบแนะนำสถานที่แบบอัตโนมัติ (Autocomplete)
- (2) การแสดงแผนที่ (Map Display)
 - แสดงแผนที่ในรูปแบบต่าง ๆ เช่น แผนที่ปกติ (Normal Map), แผนที่ภาพถ่ายดาวเทียม (Satellite Map), และแผนที่การจราจร (Traffic Map)
- (3) การคำนวณเส้นทาง (Directions)
 - ให้ข้อมูลเส้นทางการเดินทางระหว่างจุดเริ่มต้นและจุดหมาย
 - รองรับหลายรูปแบบการเดินทาง เช่น เดิน, ขับรถ, ขี่จักรยาน, หรือใช้ระบบขนส่งสาธารณะ
- (4) การวางแผนเส้นทางแบบหลายจุด (Multiple Waypoints)
 - รองรับการเพิ่มหลายจุดแวะพักในเส้นทางเดียว และคำนวณเส้นทางที่เหมาะสมที่สุด
- (5) การแสดงเวลาการเดินทาง (Travel Time)
 - คำนวณเวลาที่ใช้ในการเดินทางระหว่างแต่ละจุด โดยพิจารณาสภาพจราจรแบบเรียลไทม์
- (6) การแสดงข้อมูลสถานที่ (Place Details)
 - แสดงข้อมูลสำคัญของสถานที่ เช่น ชื่อ, ที่อยู่, เวลาเปิด-ปิด, หมายเลขโทรศัพท์ และรีวิว
- (7) การนำทางแบบเรียลไทม์ (Real-time Navigation)
 - มีฟังก์ชันนำทางแบบเรียลไทม์ พร้อมการอัปเดตข้อมูลจราจร
- (8) การแสดงข้อมูลสภาพจราจรและเวลาเดินทาง (Traffic and ETA)
 - แสดงข้อมูลการจราจรในพื้นที่ และเวลาเดินทางที่คาดการณ์ (ETA - Estimated Time of Arrival)
- (9) การฝังแผนที่ในเว็บหรือแอป (Embedding Maps)
 - รองรับการฝังแผนที่แบบโต้ตอบในเว็บไซต์หรือแอปพลิเคชัน

(10) การแปลงพิกัด GPS (Geocoding/Reverse Geocoding)

- แปลงที่อยู่เป็นพิกัด GPS และในทางกลับกัน

2.3.1.2 ความเกี่ยวข้องกับโครงการนี้

- (1) การค้นหาและแนะนำสถานที่: ระบบในโครงการนี้ Place Search และ Autocomplete เพื่อให้ผู้ใช้สามารถกรอกสถานที่ได้สะดวก และแนะนำสถานที่ที่ตรงกับความต้องการ
- (2) การคำนวณเส้นทาง: Directions API ช่วยในการคำนวณเส้นทางระหว่างแต่ละสถานที่ในโครงการ รวมถึงการพิจารณาเวลาเดินทางและสภาพจราจร
- (3) การวางแผนเส้นทางแบบหลายจุด: Multiple Waypoints ช่วยวางแผนเส้นทางสำหรับการเดินทางที่มีหลายจุดแวะพัก และค้นหาเส้นทางที่เหมาะสมที่สุด
- (4) การแสดงข้อมูลสถานที่: Place Details API ใช้เพื่อดึงข้อมูลเวลาเปิด-ปิดของสถานที่ ซึ่งเป็นส่วนสำคัญของเงื่อนไขในโครงการ
- (5) การแสดงแผนที่และเส้นทาง: การแสดงแผนที่ผ่าน Google Maps ช่วยให้ผู้ใช้สามารถมองเห็นเส้นทางชัดเจน และสามารถนำทางไปยังจุดหมายปลายทางได้
- (6) การคำนวณเวลาเดินทาง: Travel Time และ ETA เป็นส่วนสำคัญในการคำนวณและแสดงเวลารวมของการเดินทาง รวมถึงการอยู่ในแต่ละสถานที่
- (7) การแปลงพิกัด GPS: หากโครงการต้องการแปลงพิกัดสถานที่หรือแสดงข้อมูลบนแผนที่ การใช้ Geocoding API จะช่วยให้ทำงานได้ง่ายขึ้น

2.3.2 Visit a city



ภาพที่ 2.10 แสดงสัญลักษณ์ของ Visit a city (Visit a city LOGO)

2.3.2.1 ฟังก์ชันที่การทำงาน:

- (1) การแนะนำแผนการเดินทางที่เหมาะสม
 - ผู้ใช้สามารถเลือกแผนการเดินทางสำเร็จรูปตามจำนวนวันที่ต้องการ เช่น 1 วัน, 3 วัน
 - แสดงลำดับการท่องเที่ยวตามความเหมาะสม (Optimized Route) เช่น ระยะทางที่ใกล้ที่สุด และเวลาที่ใช้ในการเดินทาง
- (2) การคำนวณเวลาเปิด-ปิดของสถานที่
 - ใช้ข้อมูลเวลาเปิด-ปิดของสถานที่ในการกำหนดลำดับการท่องเที่ยว
 - หากสถานที่ไม่สามารถเข้าชมได้ในเวลาที่กำหนด ระบบจะแนะนำให้เปลี่ยนแปลง
- (3) การแสดงข้อมูลการเดินทางระหว่างจุดหมาย
 - คำนวณระยะทางและเวลาในการเดินทางระหว่างแต่ละสถานที่

- เสนอแนะการเดินทางในรูปแบบต่าง ๆ เช่น การเดิน, การใช้รถสาธารณะ
- (4) การปรับแต่งแผนการเดินทาง (Customizable Itinerary)
 - ผู้ใช้สามารถเพิ่มหรือลบสถานที่ที่ต้องการไปเยี่ยมชมในแผนการเดินทางได้
 - ระบบอัปเดตแผนการเดินทางอัตโนมัติเมื่อมีการเปลี่ยนแปลง
- (5) แสดงระยะเวลาที่ต้องใช้ต่อสถานที่
 - ระบบกำหนดเวลาขั้นต่ำที่ควรใช้ในแต่ละสถานที่ และรวมเวลานี้ไว้ในแผนการ
- (6) การแสดงแผนที่และการนำทาง (Map and Navigation)
 - มีแผนที่แบบโต้ตอบที่แสดงเส้นทางระหว่างสถานที่
 - เชื่อมต่อกับ Google Maps หรือแอปนำทางอื่น ๆ เพื่อการเดินทาง

2.3.2.2 ความเกี่ยวข้องกับโครงการนี้

- (1) การคำนวณเส้นทางที่เหมาะสม: Visit a City ใช้การคำนวณเส้นทางเพื่อจัดลำดับการเดินทางตามเงื่อนไข เช่น ระยะทางที่สั้นที่สุด หรือเวลาเปิด-ปิดสถานที่ ซึ่งสอดคล้องกับแนวคิด "Path Optimization" ในโครงการของคุณ
- (2) การจัดการเวลาในแต่ละสถานที่: ฟังก์ชันที่แสดงเวลาที่ควรใช้ในแต่ละสถานที่และการคำนวณรวมเหมือนกับโครงการที่คุณพัฒนา ซึ่งให้ผู้ใช้กำหนดเวลาพักในแต่ละจุด
- (3) ความยืดหยุ่นในการปรับแผน: การปรับแผนการเดินทางแบบเรียลไทม์ (Real-time Adjustment) ของ Visit a City มีความคล้ายคลึงกับโครงการของคุณที่ให้ผู้ปรับข้อมูล เช่น เวลาที่ใช้ในแต่ละสถานที่
- (4) การเชื่อมต่อกับระบบแผนที่: Visit a City ใช้การเชื่อมต่อกับ Google Maps เพื่อแสดงเส้นทางและการนำทาง ซึ่งตรงกับการเชื่อมต่อ API ที่โครงการของคุณวางแผนไว้
- (5) การประมวลผลเวลาเปิด-ปิด: การพิจารณาเวลาเปิด-ปิดของสถานที่ที่เป็นข้อจำกัดที่สำคัญใน Visit a City และโครงการของคุณใช้แนวคิดนี้เพื่อกำหนดว่าเส้นทางใดสามารถดำเนินการได้

Function	Plan My Trip	Google Maps	Visit a city
1. การค้นหาสถานที่ (Place Search)	มีระบบแนะนำสถานที่แบบ Autocomplete	มีระบบค้นหาสถานที่และ Autocomplete	มีระบบค้นหาสถานที่ในแผนการเดินทาง
2. การแสดงเส้นทาง (Route Display)	แสดงเฉพาะเส้นทางที่สอดคล้องกับเงื่อนไข	แสดงเส้นทางแบบครบทุกเส้นทาง	แสดงเส้นทางของทริปที่จัดวางไว้
3. การวางแผนเส้นทางแบบหลายจุด (Multi-Stop)	รองรับการเพิ่มจุดแวะพักหลายจุดได้สูงสุด 5 สถานที่	รองรับการเพิ่มหลายจุดแวะพัก	จัดการเส้นทางที่แนะนำพร้อมแผนที่ในวันเที่ยวในแต่ละวัน
4. เวลาเปิด-ปิดสถานที่ (Opening Hours)	คำนวณโดยใช้ API ของสถานที่ Place details	มีข้อมูลเวลาเปิด-ปิดจาก Place Details API	แสดงเวลาเปิด-ปิดในสถานที่ที่แนะนำ
5. การคำนวณเวลาเดินทาง (Travel Time)	คำนวณเวลาเดินทางรวมและเวลาที่ใช้ในแต่ละสถานที่	คำนวณเวลาเดินทางระหว่างจุดพร้อมสภาพจราจร	แสดงเวลาเดินทางในแผนที่จัดให้
6. การคำนวณเส้นทางที่เหมาะสม (Optimal Route)	ใช้อัลกอริทึมเฉพาะ เช่น A* หรือ Constraint-Based	ใช้การคำนวณเส้นทางที่เร็วที่สุดโดยอัตโนมัติ	จัดวางเส้นทางที่เหมาะสมในแผนที่เที่ยว
7. การปรับแต่งเงื่อนไขการเดินทาง (Customization)	ผู้ใช้เลือกสถานที่และระยะเวลาที่ใช้ในแต่ละจุด	-	ผู้ใช้สามารถจัดการกิจกรรมและระยะเวลาที่ใช้
8. การแสดงแผนที่ (Map Display)	-	รองรับการแสดงแผนที่ทั้งปกติและภาพถ่ายดาวเทียม	แสดงแผนที่พร้อมเส้นทางของแผนที่เที่ยว
9. การนำทาง (Navigation)	มีลิงก์นำทางไปยัง Google Maps	มีระบบนำทางแบบเรียลไทม์	-
10. การจัดเก็บข้อมูล (Data Storage)	-	มีการบันทึกสถานที่โปรดหรือประวัติการค้นหา	จัดเก็บแผนการเดินทางสำหรับผู้ใช้
11. การแสดงสถานที่แนะนำ	-	มีฟีเจอร์สถานที่ยอดนิยมในพื้นที่	มีระบบแนะนำสถานที่พร้อมแผนที่เที่ยว

Function	Plan My Trip	Google Maps	Visit a city
(Recommended POIs)			
12. การแสดงข้อมูลสถานที่ (Place Details)	ดึงข้อมูลเวลาเปิด-ปิด โดย API	แสดงข้อมูลพื้นฐาน เช่น เวลาเปิด-ปิด รีวิว	แสดงรายละเอียดของสถานที่ในแผนที่
13. ผู้จัดการทริป (Trip Management)	ไม่มีการจัดการทริปแบบเต็มรูปแบบ	-	จัดการแผนการเดินทางทั้งหมดได้

ตารางที่ 2.1 การเปรียบเทียบแอปพลิเคชันที่คล้ายคลึงกัน

จากตารางที่ 2.1 เป็นการเปรียบเทียบแอปพลิเคชันที่มีลักษณะบางส่วนคล้ายคลึงกันเว็บแอปพลิเคชัน Plan My Trip Google Maps และ Visit a City ต่างก็เป็นเครื่องมือที่มีเป้าหมายช่วยอำนวยความสะดวกในการเดินทาง โดยแต่ละระบบมีจุดเด่นและวัตถุประสงค์การใช้งานที่แตกต่างกันออกไป เว็บแอปพลิเคชันนี้เน้นการสร้างเส้นทางการเดินทางที่ปรับแต่งได้ตามความต้องการของผู้ใช้งาน เช่น การกำหนดเวลาที่จะใช้ในแต่ละสถานที่ และข้อจำกัดด้านเวลาเปิด-ปิดของสถานที่เป้าหมาย โดยให้ผลลัพธ์เป็นเส้นทางที่เหมาะสมกับเงื่อนไขที่สุด ทั้งในด้านเวลาและระยะทาง พร้อมรายละเอียดเกี่ยวกับแต่ละเส้นทาง ซึ่งช่วยตอบโจทย์การวางแผนแบบเฉพาะบุคคลได้อย่างมีประสิทธิภาพ

ในขณะที่ Google Maps นั้นโดดเด่นในเรื่องการนำทางแบบเรียลไทม์ การค้นหาสถานที่ยอดนิยม และการแสดงเส้นทางที่หลากหลาย โดยเน้นความสะดวกสบายสำหรับการเดินทางในชีวิตประจำวัน อย่างไรก็ตาม ระบบนี้ไม่ได้รองรับการวางแผนที่สามารถระบุเวลาที่ใช้ในแต่ละจุดได้ ทำให้เหมาะสำหรับผู้ใช้งานที่ต้องการคำแนะนำเส้นทางที่รวดเร็วและต้องการทราบแค่ระยะเวลาในการเดินทางเท่านั้น

ด้าน Visit a City มีความสามารถในการช่วยวางแผนการเดินทางในเชิงกำหนดการ โดยจัดลำดับสถานที่ตามความนิยมและเวลาเปิด-ปิดแบบอัตโนมัติ เหมาะสำหรับนักท่องเที่ยวที่ต้องการแผนการเดินทางที่ครอบคลุมสถานที่ท่องเที่ยวยอดนิยม แต่ไม่สามารถปรับแต่งเวลาที่ใช้ในแต่ละสถานที่ได้เหมือนกับเว็บแอปในโครงการนี้

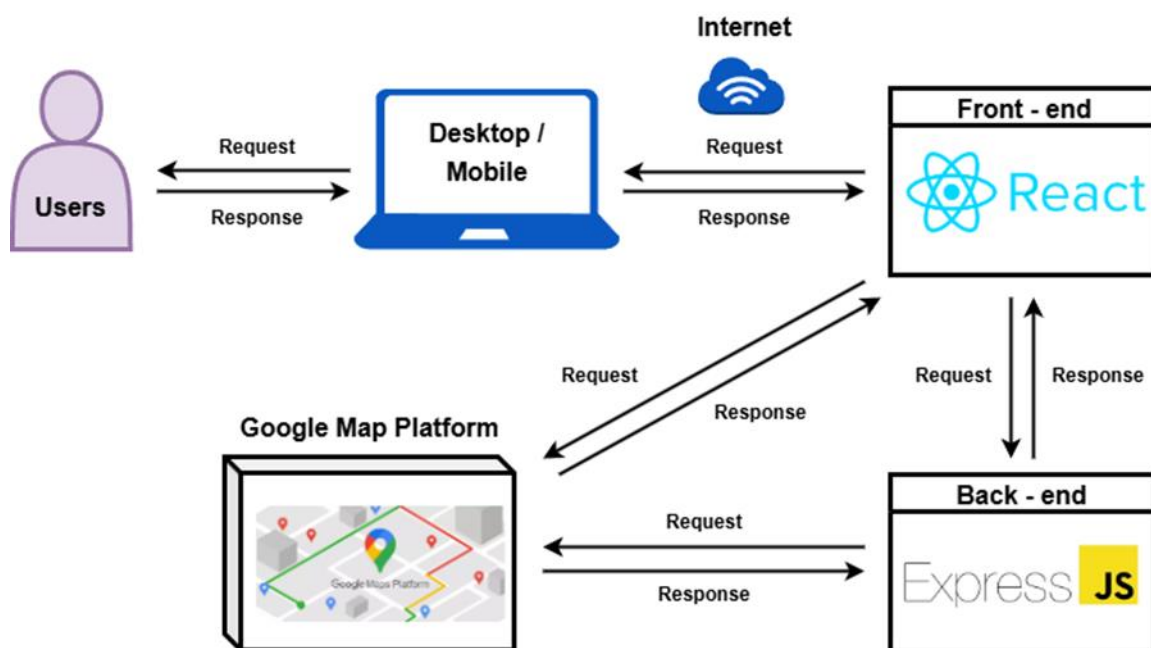
สรุปได้ว่า เว็บแอปพลิเคชันนี้มีความแตกต่างและโดดเด่นในด้านการปรับแต่งเส้นทางและเวลาที่ใช้ในแต่ละสถานที่อย่างละเอียด ซึ่งเป็นสิ่งที่ Google Maps และ Visit a City ไม่

สามารถตอบสนองได้ในระดับเดียวกัน การพัฒนาเว็บแอปนี้จึงเหมาะสมสำหรับผู้ใช้งานที่ต้องการ
แผนการเดินทางที่ยืดหยุ่นและตรงกับความต้องการเฉพาะตัวมากที่สุด

บทที่ 3

วิธีการวิจัย

3.1 ภาพรวมของโครงงานและสภาพแวดล้อมของระบบ

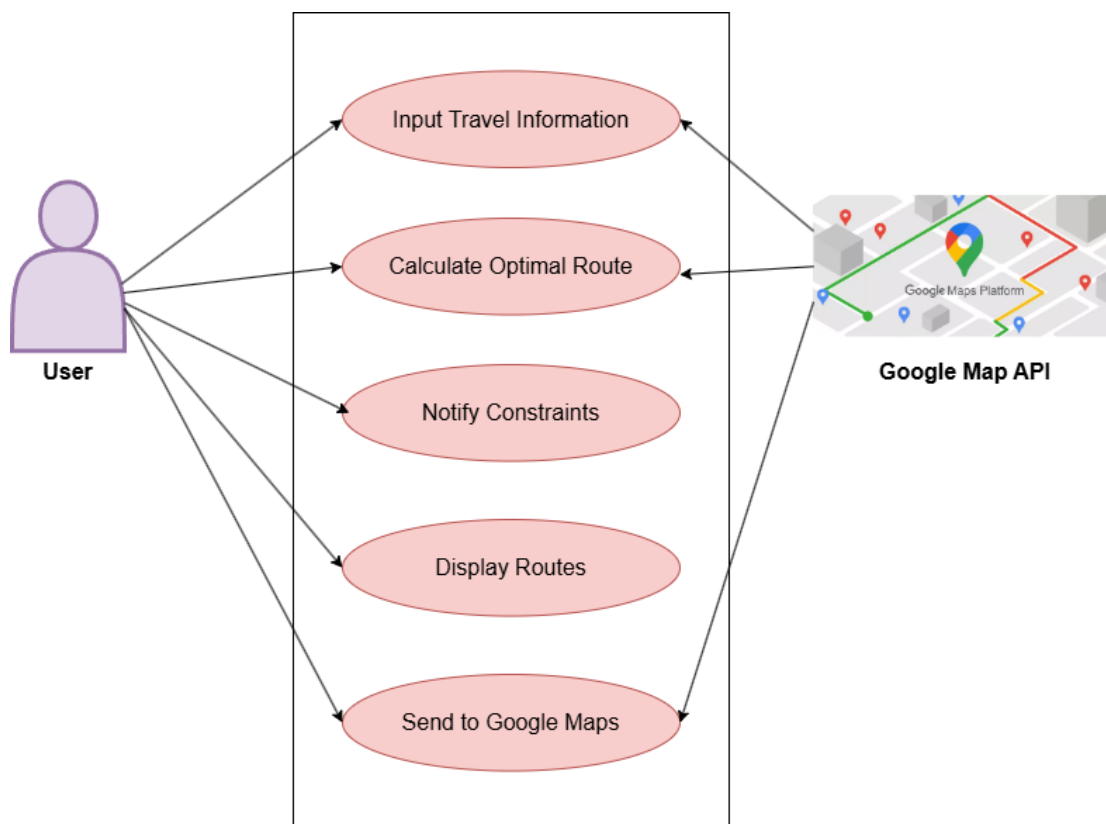


ภาพที่ 3.1 แสดงสถาปัตยกรรมระบบ (Architecture Diagram)

จากภาพที่ 3.1 แสดงสถาปัตยกรรมของระบบที่ผู้ใช้ใช้งานเว็บแอปพลิเคชัน Plan My Trip ผ่านการทำงานบนอุปกรณ์สมาร์ทโฟน (Mobile device) เพื่อแสดงแผนการเดินทางที่ดีที่สุดจากสถานที่ที่ผู้ใช้ โดยมีรายละเอียดดังนี้

- (1) Mobile Client เป็นส่วนที่ผู้ใช้ใช้งานเข้าถึงระบบ คืออุปกรณ์สมาร์ทโฟน (Mobile device) โดยจะส่ง Request เข้าไปใน Front-end และรอรับ Response เพื่อแสดงผลให้ผู้ใช้ โดยมีการทำงานผ่านเครือข่ายอินเทอร์เน็ต
- (2) Front-end เป็นส่วนของหน้าเว็บแอปพลิเคชัน โดยจะส่ง Request เข้าไปยัง Backend จากนั้นจะรอรับ Response เพื่อขอข้อมูล ต่าง ๆ นำไปแสดงบนหน้าเว็บแอปพลิเคชัน

3.2 การวิเคราะห์ขอบเขตและความต้องการของระบบ



ภาพที่ 3.2 แผนภาพกรณีศึกษา (Use Case Diagram)

จากภาพที่ 3.2 เป็นภาพที่แสดงกรณีใช้งานของเว็บแอปพลิเคชัน Plan My Trip โดยผู้ใช้งานระบบจะเป็นผู้วางแผนการเดินทาง ผู้ใช้งานจะต้องกรอกข้อมูลสถานที่และเงื่อนไขต่างๆ ซึ่งสถานที่ที่ผู้ใช้เลือกจะจำกัดเพียงสถานที่ ที่มีอยู่ใน Google Map เท่านั้น โดยเมื่อผู้ใช้กรอกข้อมูลทั้งหมดและกดปุ่มประมวลผลแล้ว ระบบจะประมวลผลเพื่อแสดงแนวทางการเดินทางที่ดีที่สุดออกมาทางหน้าจอ โดยผู้ใช้งานสามารถส่งข้อมูลเส้นทางการเดินทางที่ได้ไปยัง google maps เพื่อเริ่มต้นการเดินทางได้อย่าง Real-time

3.2.1 คำอธิบายแผนภาพกรณีใช้งาน (Use Case Description)

ตารางที่ 3.1 รายละเอียดกรณีการใช้งานระบบทั้งหมด

Use case ID	Use case name	Description
UC-01	กรอกข้อมูลการเดินทาง (สถานที่, เงื่อนไข)	ผู้ระบุข้อมูล 1. สถานที่ (ไม่เกิน 10 สถานที่) 2. วันที่เริ่มต้นเดินทาง 3. เวลาที่เริ่มต้นเดินทาง 4. เวลาพักในแต่ละสถานที่ 5. หลีกเลียงค่าผ่านทาง (optional)
UC-02	คำนวณเส้นทางที่เหมาะสม	ระบบประมวลผลเลือกเส้นทาง ที่ดีที่สุดภายใต้เงื่อนไขทั้งหมด
UC-03	แจ้งเตือนข้อจำกัด	ระบบทำการแจ้งเตือนข้อจำกัด ของเงื่อนไข ผู้ใช้สามารถ เปลี่ยนแปลงเงื่อนไขการ เดินทาง หรือเลือกให้ระบบ คำนวณเส้นทางโดยไม่สน เงื่อนไขนั้น
UC-04	แสดงผลเส้นทาง	ระบบแสดงลำดับเส้นทางที่ เหมาะสม พร้อมข้อมูลสำคัญ เช่น ระยะเวลารวม และลำดับ การเดินทาง
UC-05	ส่งข้อมูลไปยัง Google Maps	ระบบเชื่อมต่อกับ Google map ผู้ใช้งานสามารถส่ง ข้อมูลเส้นทางที่ผู้ใช้งานเลือก ไปยัง Google Maps เพื่อเปิด การนำทางได้อย่าง Real-time

3.3 ออกแบบการทำงานของระบบ

3.3.1 คำอธิบาย use case ของระบบ (Use case Specification)

ตารางที่ 3.2 รายละเอียดกรณีผู้ใช้งานกรอกข้อมูลการเดินทางลงในระบบ

รหัสยูสเคส (Use Case ID)	UC-01
ชื่อยูสเคส (User Case Name)	กรอกข้อมูลการเดินทาง
ผู้ใช้งาน (Actor)	ผู้ใช้งาน, Google Maps API
คำอธิบาย (Description)	<ol style="list-style-type: none"> ผู้ใช้งานระบุข้อมูล <ul style="list-style-type: none"> สถานที่ (ไม่เกิน 10 สถานที่) วันที่เริ่มต้นเดินทาง เวลาที่เริ่มต้นเดินทาง เวลาพักในแต่ละสถานที่ หลักเลี้ยวค่าผ่านทาง (optional) Google Maps API ขึ้นข้อความแนะนำสถานที่อัตโนมัติ (Autocomplete) ให้ผู้ใช้งานกดเลือกสถานที่ได้
เงื่อนไขก่อนหน้า (Pre-condition)	ผู้ใช้เข้าสู่หน้าเว็บแอปพลิเคชันเรียบร้อยแล้ว
เงื่อนไขภายหลัง (Post-condition)	ระบบได้รับข้อมูลครบถ้วนเพื่อใช้ในการคำนวณเส้นทาง
กระแสหลัก (Basic Flow)	<ol style="list-style-type: none"> ผู้ใช้งานเปิดหน้าเว็บ ผู้ใช้งานกรอกข้อมูล: <ul style="list-style-type: none"> วันที่เดินทาง เวลาเริ่มต้น รูปแบบการเดินทาง หลักเลี้ยวค่าผ่านทาง ผู้ใช้งานเพิ่มสถานที่พร้อมระยะเวลาที่ต้องการอยู่ในแต่ละสถานที่

รหัสยูสเคส (Use Case ID)	UC-01
กระแสรอง (Alternative Flow)	<ul style="list-style-type: none">• กรณีข้อมูลไม่ครบถ้วน: ระบบแจ้งเตือนให้กรอกข้อมูลที่ขาด• กรณีรูปแบบข้อมูลไม่ถูกต้อง: ระบบแจ้งให้แก้ไข

ตารางที่ 3.3 รายละเอียดกรณีระบบคำนวณเส้นทางที่เหมาะสม

รหัสยูสเคส (Use Case ID)	UC-02
ชื่อยูสเคส (User Case Name)	คำนวณเส้นทางที่เหมาะสม
ผู้ใช้งาน (Actor)	ผู้ใช้, Google Maps API
คำอธิบาย (Description)	ระบบประมวลผลเลือกเส้นทางที่ดีที่สุดภายใต้เงื่อนไขทั้งหมด
เงื่อนไขก่อนหน้า (Pre-condition)	ระบบได้รับข้อมูลจาก UC-01
เงื่อนไขภายหลัง (Post-condition)	ระบบสร้างลำดับเส้นทางที่เหมาะสมและพร้อมแสดงผล
กระแสหลัก (Basic Flow)	<ol style="list-style-type: none"> 1. ผู้ใช้งานกดปุ่ม “เริ่มวางแผนการเดินทาง” 2. ระบบดึงข้อมูลเส้นทางและเวลาจาก Google Maps API 3. ระบบคำนวณเส้นทางที่เหมาะสมที่สุดตามข้อจำกัด 4. ระบบตรวจสอบความเป็นไปได้และจัดลำดับเส้นทาง 5. ระบบบันทึกผลเส้นทางที่คำนวณได้
กระแสรอง (Alternative Flow)	กรณีไม่สามารถหาลำดับเส้นทางที่เหมาะสม: แจ้งเตือนผู้ใช้งานเพื่อแก้ไขข้อมูลหรือ ยืนยันการคำนวณเส้นทางที่ดีที่สุดโดยไม่สนใจเงื่อนไข

ตารางที่ 3.4 รายละเอียดกรณีระบบแจ้งเตือนข้อจำกัด

รหัสยูสเคส (Use Case ID)	UC-03
ชื่อยูสเคส (User Case Name)	แจ้งเตือนข้อจำกัด
ผู้ใช้งาน (Actor)	ผู้ใช้งาน
คำอธิบาย (Description)	ระบบทำการแจ้งเตือนข้อจำกัดของเงื่อนไข ผู้ใช้สามารถเปลี่ยนแปลงเงื่อนไขการเดินทาง หรือเลือกให้ระบบคำนวณเส้นทางโดยไม่สนเงื่อนไขนั้น
เงื่อนไขก่อนหน้า (Pre-condition)	ระบบพยายามคำนวณเส้นทางจาก UC-02
เงื่อนไขภายหลัง (Post-condition)	ผู้ใช้งานได้รับคำแนะนำเพื่อปรับปรุงข้อมูล
กระแสหลัก (Basic Flow)	<ol style="list-style-type: none"> 1. ระบบตรวจสอบข้อจำกัดจากข้อมูล 2. ระบบแจ้งเตือนผู้ใช้งานเกี่ยวกับข้อจำกัด 3. ผู้ใช้งานปรับข้อมูลตามคำแนะนำ 4. ระบบเริ่มคำนวณใหม่
กระแสรอง (Alternative Flow)	กรณีผู้ใช้งานไม่แก้ไขข้อมูล: ระบบแสดงคำแนะนำเพิ่มเติม

ตารางที่ 3.5 รายละเอียดกรณีระบบแสดงผลเส้นทาง

รหัสยูสเคส (Use Case ID)	UC-04
ชื่อยูสเคส (User Case Name)	แสดงผลเส้นทาง
ผู้ใช้งาน (Actor)	ผู้ใช้
คำอธิบาย (Description)	ระบบแสดงลำดับเส้นทางที่เหมาะสม พร้อมข้อมูลสำคัญ เช่น ระยะเวลารวม และลำดับการเดินทาง
เงื่อนไขก่อนหน้า (Pre-condition)	ระบบคำนวณเส้นทางเรียบร้อยแล้วใน UC-02
เงื่อนไขภายหลัง (Post-condition)	ผู้ใช้งานสามารถเลือกเส้นทางที่ต้องการ
กระแสหลัก (Basic Flow)	<ol style="list-style-type: none"> 1. ระบบแสดงข้อมูลเส้นทางที่ดีที่สุดตามลำดับ 2. ระบบแสดงตัวเลือกเส้นทางเพิ่มเติม 3. ผู้ใช้งานเลือกเส้นทางที่ต้องการ
กระแสรอง (Alternative Flow)	กรณีผู้ใช้งานเปลี่ยนใจ: ระบบย้อนกลับไปยังการแก้ไขข้อมูล

ตารางที่ 3.6 ส่งข้อมูลไปยัง Google Maps

รหัสยูสเคส (Use Case ID)	UC-05
ชื่อยูสเคส (User Case Name)	ส่งข้อมูลไปยัง Google Maps
ผู้ใช้งาน (Actor)	ผู้ใช้, Google Maps API
คำอธิบาย (Description)	ระบบเชื่อมต่อกับ Google map ผู้ใช้งานสามารถกดส่งข้อมูลเส้นทางที่ผู้ใช้งานเลือกไปยัง Google Maps เพื่อเปิดการนำทางได้อย่าง Real-time
เงื่อนไขก่อนหน้า (Pre-condition)	ผู้ใช้งานเลือกเส้นทางใน UC-04
เงื่อนไขภายหลัง (Post-condition)	ระบบเปิด Google Maps พร้อมข้อมูลเส้นทางที่ตั้งค่าไว้
กระแสหลัก (Basic Flow)	<ol style="list-style-type: none"> 1. ผู้ใช้งานกดเลือกเส้นทาง 2. ระบบส่งข้อมูลเส้นทางไปยัง Google Maps API 3. ระบบเปิด Google Maps พร้อมเส้นทางที่เลือก
กระแสรอง (Alternative Flow)	กรณี API มีปัญหา: ระบบแจ้งเตือนและแนะนำผู้ใช้งานใช้วิธีอื่น

3.4 ขั้นตอนการเรียกใช้ API ในเว็บแอปพลิเคชัน Plan My Trip

การเรียกใช้ API เป็นส่วนสำคัญในการดึงข้อมูลที่เกี่ยวข้อง เช่น เส้นทาง การเดินทาง สถานที่เปิด-ปิด และการประเมินเวลาเดินทาง

(1) เลือก API ที่เกี่ยวข้อง

- Google Maps API: ใช้สำหรับดึงข้อมูลสถานที่ เส้นทาง การคำนวณระยะทาง และเวลาการเดินทาง
- Google Places API: ใช้สำหรับให้ผู้ใช้ค้นหาสถานที่ด้วย autocomplete

(2) ตั้งค่าและส่งคำขอ (Request)

- กำหนด API Key เพื่อเข้าถึง API (ต้องตั้งค่าในไฟล์ .env เพื่อความปลอดภัย)
- ส่งคำขอ HTTP (GET/POST) ไปยัง API โดยระบุพารามิเตอร์ เช่น พิกัดของสถานที่ เวลาที่เริ่มเดินทาง หรือโหมดการเดินทาง

ข้อมูลที่ได้จาก API จะถูกนำไปใช้ในแต่ละส่วนของระบบดังนี้:

- Google Maps Directions API: ดึงข้อมูลเส้นทางที่สั้นที่สุดระหว่างสถานที่ พร้อมระยะทางและเวลาที่ใช้เดินทาง
- Google Places API: ให้ผู้ใช้สามารถกรอกชื่อสถานที่แบบอัตโนมัติ

ข้อมูลทั้งหมดจะถูกนำมาใช้ในฟังก์ชันที่จัดการการคำนวณ เช่น การเรียงลำดับเส้นทางหรือการตรวจสอบเงื่อนไขเวลาเปิดปิดของสถานที่ การเก็บข้อมูลเพื่อนำมาแสดงผล

(1) ข้อมูลจาก API

- เก็บข้อมูลที่ได้จาก API ไว้ในตัวแปรแบบชั่วคราวในหน่วยความจำ (Memory)
- ใช้โครงสร้างข้อมูล เช่น JSON สำหรับเก็บรายละเอียดของสถานที่ เส้นทาง และเวลาที่ใช้

(2) Frontend State Management

- ใช้ React State Management เช่น useState หรือ Redux เพื่อเก็บข้อมูลที่รับจาก Backend

3.5 วิธีการคำนวณการหาเส้นทางที่ดีที่สุดโดยใช้ A* Algorithm

3.5.1 องค์ประกอบในการคำนวณ

A* (อ่านว่า “A star”) เป็น อัลกอริทึมค้นหาเส้นทางที่เร็วที่สุด จากจุดเริ่มต้นไปยังจุดเป้าหมาย โดย A* จะพิจารณา “คะแนนรวม” ของแต่ละเส้นทางที่กำลังพิจารณาอยู่ โดยใช้สูตร:

$$F(n) = g(n) + h(n)$$

- n คือ โหนดปัจจุบัน
- $g(n)$ = ต้นทุนจริงจากจุดเริ่มต้นถึง n
- $h(n)$ = ค่าประมาณต้นทุนจาก n ไปยังเป้าหมาย (heuristic)
- $f(n)$ = ต้นทุนรวมโดยประมาณของเส้นทางผ่าน n

โดยมีการนิยามสถานะเริ่มต้น (Initial State):

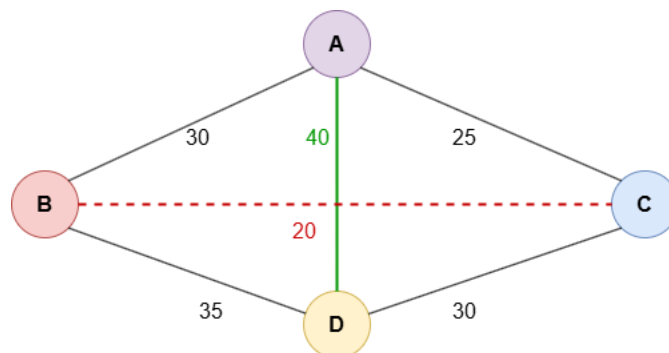
- เริ่มจากจุดแรก (index 0) สร้าง sequence เริ่มต้น เช่น [0]
- สร้าง mask (bitmask) เพื่อเก็บว่าผ่าน node ไหนแล้ว
- ถ้ามีจุด locked (ล็อกลำดับไว้), จะสร้าง sequence ที่ "บังคับไปตามนั้นก่อน"

3.5.2 ลักษณะการคำนวณ

- g คือ ค่าที่ใช้สะสมระยะทางจริง (distance) ที่เดินทางมาแล้ว
- h คือ ค่าประมาณระยะทางที่เหลือ ในอนาคต ใช้ Minimum Spanning Tree (MST) จากจุดที่เหลือ (ยังไม่ได้ไป) โดยอิงจาก straight-line (หรือระยะทางจริง)
- รวมเป็น Priority = $g + h$ แล้ว ใช้ Priority นี้จัดลำดับใน MinHeap (คิวลำดับ)
- วนลูป A Search* โดยการดึงสถานะที่ priority น้อยที่สุดออกจาก heap
- ถ้า mask == ALL \rightarrow หมายถึงผ่านทุกจุดแล้ว \rightarrow บันทึกเส้นทางนี้เป็นคำตอบ

- ถ้าไม่ : ลองขยาย (expand) ไปยังจุดอื่น ๆ ที่ยังไม่ไป ข้ามจุดที่เวลาปิดหรือไม่ผ่านเงื่อนไข locked positions จากนั้นสร้างสถานะใหม่ และเพิ่มเข้า heap ด้วย priority ใหม่
- ใช้ Priority Queue จัดลำดับ node ที่ควรขยายต่อ (ตาม $f = g + h$)
- แต่ละ node จะเก็บ:
 - path: ลำดับจุดที่ไปมาแล้ว เช่น ["A", "B", "C"]
 - visitedSet: ชุดของจุดที่แวะไปแล้ว
 - g: ค่าใช้จ่ายที่เดินมาถึงตอนนี้ (เวลา)
 - h: ประมาณเวลาที่ต้องใช้เพื่อไปยังจุดที่เหลือ
 - $f = g + h$: ใช้จัดลำดับใน priority queue
- เก็บผลลัพธ์ที่ "ไปครบทุกจุด" และจบที่ node terminal (เช่น $A \rightarrow B \rightarrow C \rightarrow D$)

3.5.3 ตัวอย่างเหตุการณ์



ภาพที่ 3.3 แสดงภาพตัวอย่างกราฟเวลาในการเดินทาง

การคำนวณรอบที่ 1 : จาก A

เปิดทางไป B, C, D:

- $A \rightarrow B$:
 - $g = 30$

- เหลือ $\{C, D\}$, $h = \min(20, 35) = 20$
- $f = 30 + 20 = 50$
- $A \rightarrow C$:
 - $g = 25$
 - เหลือ $\{B, D\}$, $h = \min(20, 30) = 20$
 - $f = 25 + 20 = 45$
- $A \rightarrow D$:
 - $g = 40$
 - เหลือ $\{B, C\}$, $h = \min(35, 30) = 30$
 - $f = 40 + 30 = 70$

เลือก $A \rightarrow C$ ต่อ ($f = 45$ ต่ำสุด)

การคำนวณรอบที่ 2: จาก $A \rightarrow C$

เปิดทางต่อไป B, D:

- $A \rightarrow C \rightarrow B$:
 - $g = 25 + 20 = 45$
 - เหลือ D, $h = 35$
 - $f = 45 + 35 = 80$
- $A \rightarrow C \rightarrow D$:
 - $g = 25 + 30 = 55$
 - เหลือ B, $h = 35$
 - $f = 55 + 35 = 90$

เลือก $A \rightarrow C \rightarrow B$ ต่อ ($f = 80$)

การคำนวณรอบที่ 3: จาก $A \rightarrow C \rightarrow B$

เหลือ D:

- $A \rightarrow C \rightarrow B \rightarrow D$:
 - $g = 25 + 20 + 35 = 80$
 - ครบทุกจุดเก็บเป็นอันดับที่ 1

กลับมารอบก่อนหน้า: $A \rightarrow B$

เปิดไป C, D:

- $A \rightarrow B \rightarrow C$:
 - $g = 30 + 20 = 50$
 - เหลือ D, $h = 30$
 - $f = 80$
- $A \rightarrow B \rightarrow D$:
 - $g = 30 + 35 = 65$
 - $h = 30$
 - $f = 95$

เลือก $A \rightarrow B \rightarrow C$ ต่อ ($f = 80$)

รอบต่อมา: $A \rightarrow B \rightarrow C \rightarrow D$

- $g = 30 + 20 + 30 = 80$
- ครบทุกจุดเก็บเป็นอันดับที่ 2

ถัดไป: $A \rightarrow D$

เปิดทาง B, C:

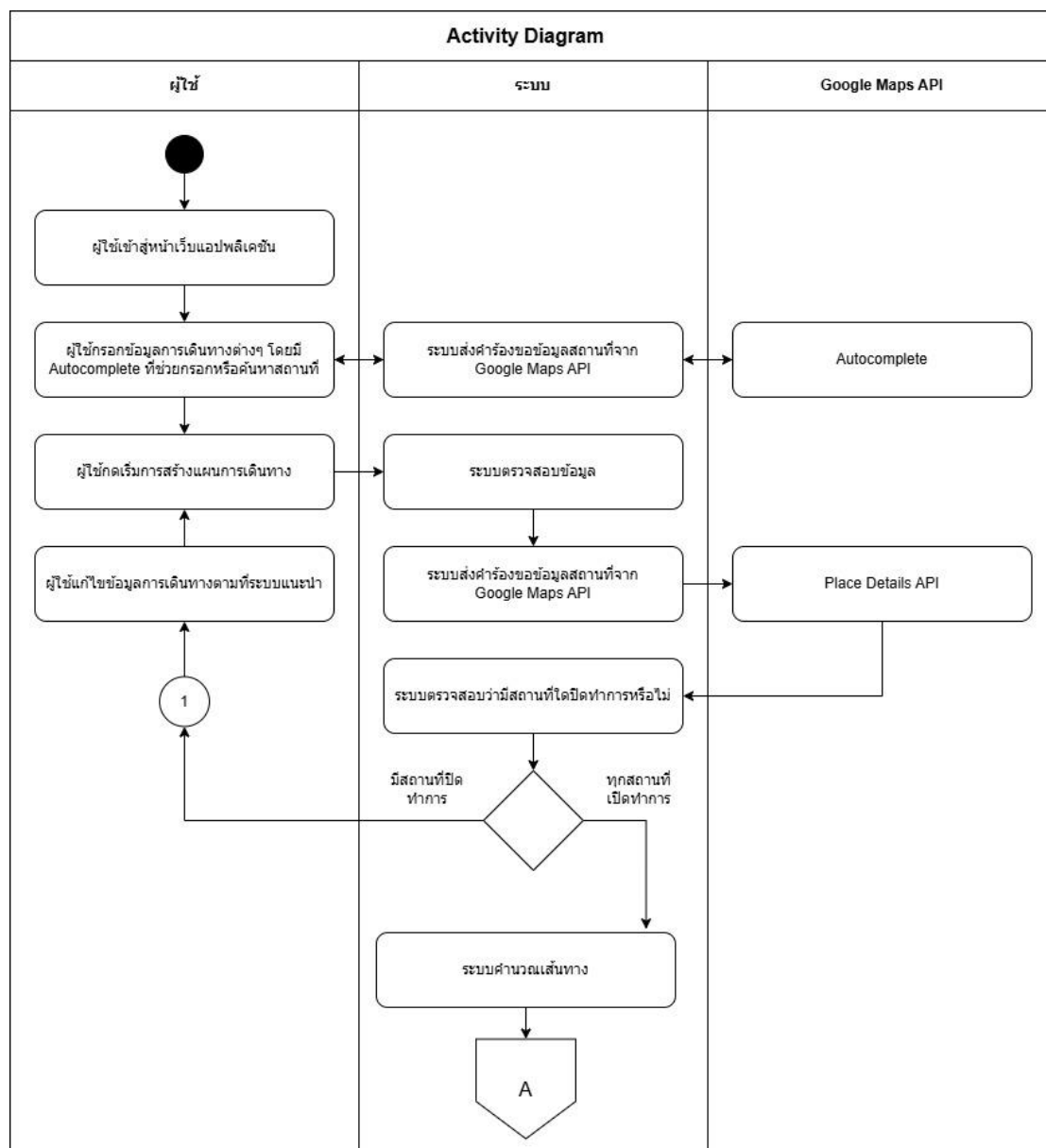
- $A \rightarrow D \rightarrow C$:
 - $g = 40 + 30 = 70$
 - เหลือ B, $h = 20$
 - $f = 90$
- $A \rightarrow D \rightarrow B$:
 - $g = 40 + 35 = 75$
 - เหลือ C, $h = 20$
 - $f = 95$

เลือก $A \rightarrow D \rightarrow C$

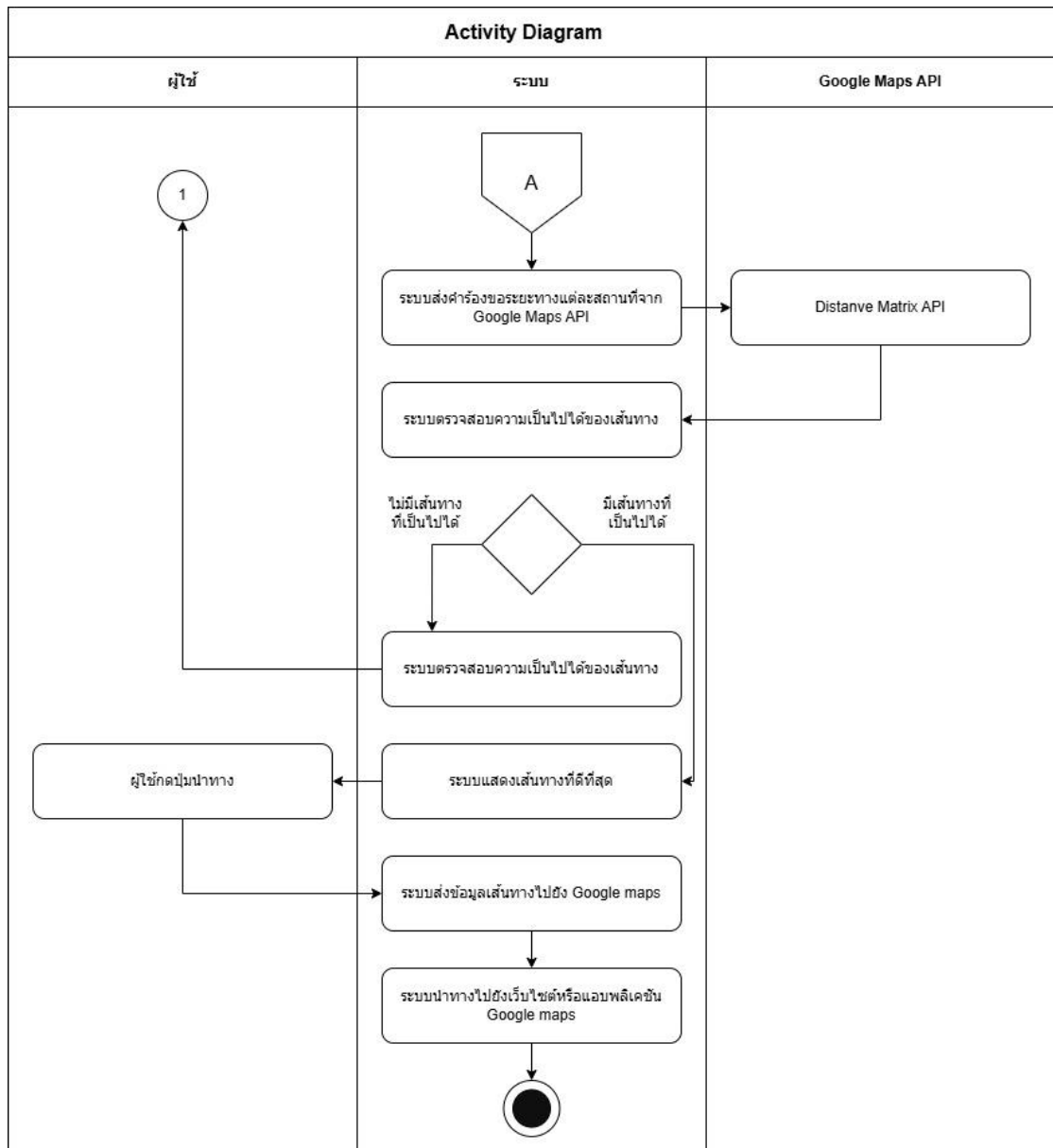
$A \rightarrow D \rightarrow C \rightarrow B$

- $g = 40 + 30 + 20 = 90$
- ครบเก็บเป็นอันดับที่ 3

3.6 ขั้นตอนการทำงานเว็บแอปพลิเคชัน



ภาพที่ 3.4 แผนภาพกิจกรรมของเว็บแอปพลิเคชัน (1)



ภาพที่ 3.5 แผนภาพกิจกรรมของเว็บแอปพลิเคชัน (2)

3.7 ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย

เว็บแอปพลิเคชัน Plan My Trip สำหรับการวางแผนการเดินทาง มีประเด็นที่น่าสนใจและสิ่งที่ท้าทาย ดังนี้

- (1) การคำนวณเส้นทางที่ดีที่สุดภายใต้ข้อจำกัด (Path Optimization with Constraints)

การรวมปัจจัยที่หลากหลาย เช่น เวลาเปิด-ปิดของสถานที่, เวลาที่ต้องใช้ในแต่ละสถานที่, และรูปแบบการเดินทาง (รถยนต์, เดิน, ขนส่งสาธารณะ) ในการคำนวณเส้นทาง และการเลือกใช้อัลกอริทึมวิธีการแก้ปัญหาที่เหมาะสม เพื่อให้ประมวลผลได้รวดเร็วและแม่นยำ โดยความท้าทายในส่วนนี้คือ การทำให้การคำนวณเส้นทางทำงานได้ภายในเวลาที่เหมาะสม แม้มีหลายจุดหมายและข้อจำกัดซับซ้อน รวมไปถึงการจัดลำดับสถานที่ที่ต้องไปโดยคำนึงถึงปัจจัยเวลาของผู้ใช้งาน

- (2) การเชื่อมต่อและใช้งาน API (API Integration)

การศึกษา Google Map Platform เพื่อที่จะได้มีความเข้าใจเกี่ยวกับ Google Map API แต่ละตัวว่าสามารถใช้งานได้อย่างไรบ้างการใช้ Google Maps API และ Place API เพื่อดึงข้อมูลเส้นทาง, เวลาเดินทาง, และสถานที่ที่เกี่ยวข้อง และการประยุกต์ใช้งาน API อย่างมีประสิทธิภาพเพื่อลดความซับซ้อนของการพัฒนาระบบ ความท้าทายในส่วนนี้คือ การจัดการโควตาการใช้งาน API (เช่น การจำกัดจำนวนการเรียก API ฟรีในแต่ละเดือน) รวมไปถึง การรวมข้อมูลจากหลาย API และประมวลผลให้อยู่ในรูปแบบที่เหมาะสมสำหรับผู้ใช้งาน

- (3) การจัดลำดับสถานที่ตามเงื่อนไขที่ผู้ใช้กำหนด (Time-Windowed Routing)

การคำนึงถึงเวลาเปิด-ปิดและเวลาที่จะใช้ในแต่ละสถานที่ ทำให้เกิดเส้นทางที่สมจริงและปฏิบัติได้ มีความท้าทายในส่วนการจัดลำดับการเดินทางให้ตรงกับเวลาที่กำหนดในแต่ละสถานที่อย่างมีประสิทธิภาพ และการประมวลผลเส้นทางที่หลากหลายเพื่อเลือกเส้นทางที่ตามเงื่อนไขที่ผู้ใช้กำหนด

3.8 ผลลัพธ์ที่คาดหวัง

(1) ระบบวางแผนการเดินทางที่มีประสิทธิภาพ

- สามารถคำนวณเส้นทางเรียงตามลำดับที่ใช้เวลาน้อยที่สุดไปเวลามากที่สุดจากแนวทางการเดินทางที่เป็นไปได้
- รองรับการคำนวณเส้นทางภายใต้ข้อจำกัด เช่น:
 - เวลาเปิด-ปิดของสถานที่
 - ระยะเวลาที่ใช้ในแต่ละสถานที่
 - รูปแบบการเดินทาง (เช่น รถยนต์, การเดิน)

(2) การแสดงผลเส้นทางและรายละเอียดที่ชัดเจน

- เส้นทางแนะนำ จะแสดงเฉพาะตัวเลือกที่ตรงตามข้อกำหนดของผู้ใช้งาน
- แสดงข้อมูลประกอบที่สำคัญ เช่น:
 - เส้นทางที่ดีที่สุด
 - ระยะทางรวมที่ต้องเดินทาง
 - ระยะเวลาทั้งหมดที่รวมการเดินทางและการใช้เวลาที่สถานที่ต่าง ๆ

(3) อินเทอร์เฟซที่ใช้งานง่าย

- มี หน้าจอผู้ใช้งานที่เข้าใจง่าย และเป็นมิตรต่อผู้ใช้ (User-Friendly)
- รองรับการป้อนข้อมูล เช่น:
 - จุดเริ่มต้นและจุดหมายปลายทาง (สูงสุด 10 จุด)
 - เวลาที่จะใช้ในแต่ละสถานที่

(4) รองรับการเชื่อมต่อ Google Maps

- สามารถ ส่งเส้นทางที่ผู้ใช้เลือก ไปยัง Google Maps เพื่อให้ผู้ใช้งานสามารถติดตามเส้นทางได้ทันที
- รองรับการกำหนดรูปแบบเส้นทาง เช่น หลีกเลี่ยงค่าทางด่วน

(5) การจัดการข้อผิดพลาดและข้อจำกัด

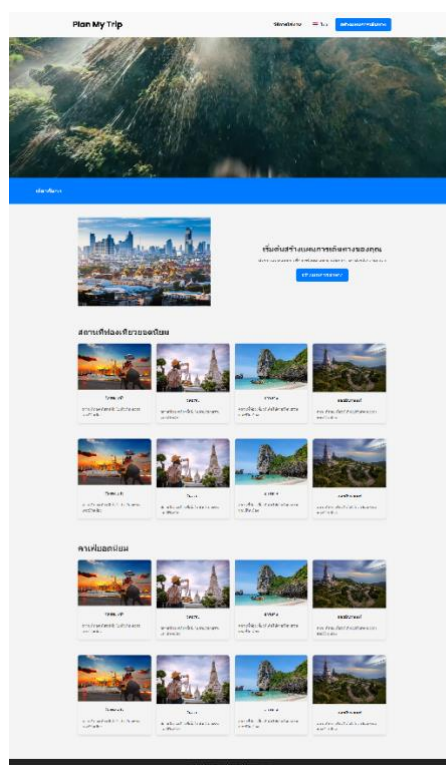
- ระบบสามารถ แจ้งเตือนผู้ใช้งาน หากไม่สามารถคำนวณเส้นทางที่เป็นไปได้ภายใต้ข้อจำกัด เช่น:
 - เวลาที่ไม่เพียงพอสำหรับการเดินทางไปยังทุกสถานที่
 - สถานที่ปิดในวันทีระบุ
- มีการเสนอทางเลือก เช่น:
 - ปรับลำดับเส้นทาง

- ขยายเวลาเดินทาง หรือเปลี่ยนสถานที่
 - (6) รองรับการเดินทางแบบหลากหลายรูปแบบ
- รองรับการคำนวณเส้นทางสำหรับรูปแบบการเดินทางที่ต่างกัน เช่น: รถยนต์ (เลือกได้ว่าจะหลีกเลี่ยงค่าทางด่วนหรือไม่), การเดิน, รถมอเตอร์ไซด์

3.9 ภาพระบบ

ภาพของระบบเว็บแอปพลิเคชัน (Web Application) Plan My Trip มีองค์ประกอบ และวิธีการใช้งานดังต่อไปนี้

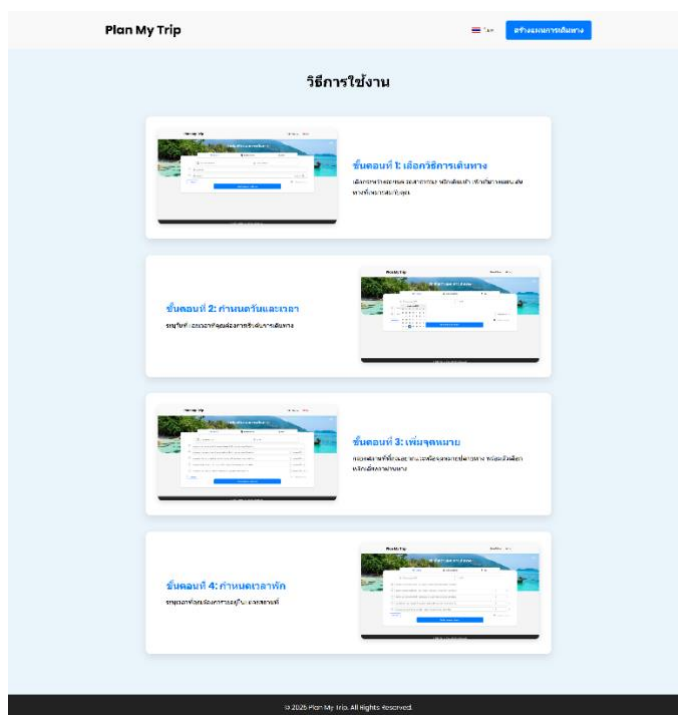
3.9.1 หน้าหลักเว็บแอปพลิเคชัน Plan my trip



ภาพที่ 3.6 หน้าหลักเว็บแอปพลิเคชัน Plan my trip

จากภาพที่ 3.6 แสดงภาพหน้าหลักเมื่อผู้ใช้เข้าสู่เว็บแอปพลิเคชัน (Web Application) Plan My Trip ซึ่งออกแบบมาให้รองรับได้หลายอุปกรณ์ (Responsive Design) โดยจะมีฟีเจอร์ในการใช้งาน ได้แก่ การเปลี่ยนภาษา การสร้างแผนการเดินทาง และแนะนำวิธีการใช้งานเว็บแอปพลิเคชัน (Web Application)

3.9.2 หน้าวิธีการใช้งานเว็บแอปพลิเคชัน Plan my trip



ภาพที่ 3.7 หน้าวิธีการใช้งานเว็บแอปพลิเคชัน Plan my trip

จากภาพที่ 3.7 แสดงภาพการแสดงผลของเว็บแอปพลิเคชันหน้าวิธีการใช้งานของเว็บแอปพลิเคชัน Plan my trip โดยหน้าวิธีการใช้งานจะเป็นหน้าที่จะอธิบายวิธีการสร้างแผนการเดินทางทุกขั้นตอน เพื่อให้ผู้ใช้สามารถสร้างแผนการเดินทางได้

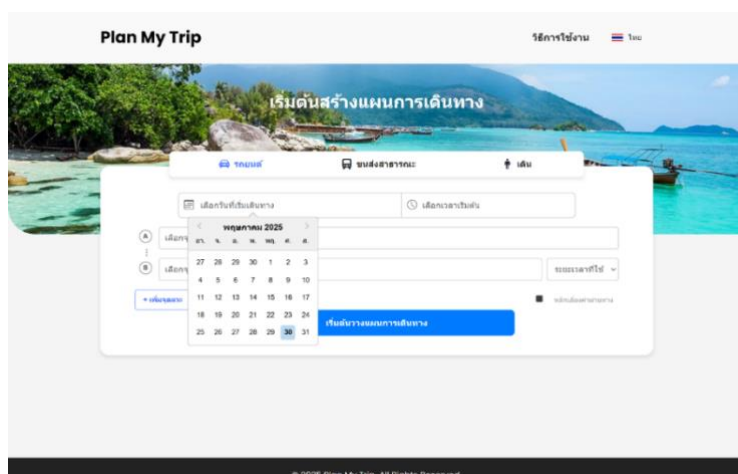
3.9.3 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip

The screenshot shows the 'Plan My Trip' web application interface. At the top, there's a header with the logo 'Plan My Trip' on the left and 'บริการใช้งาน ไทย' on the right. Below the header is a large banner image of a tropical beach. In the center, there's a white form titled 'เริ่มต้นสร้างแผนการเดินทาง' (Start creating a travel plan). The form has three tabs: 'รถยนต์' (Car), 'ขนส่งสาธารณะ' (Public Transport), and 'เดิน' (Walk). Under the 'รถยนต์' tab, there are two input fields: 'เลือกวันที่เริ่มเดินทาง' (Select start date) and 'เลือกเวลาเริ่มต้น' (Select start time). Below these are two dropdown menus: 'เลือกจุดเริ่มต้น' (Select start point) and 'เลือกจุดหมาย' (Select destination). To the right of the destination dropdown is a 'ระยะเวลาที่ใช้' (Travel time) dropdown. At the bottom of the form is a blue button labeled 'เริ่มวางแผนการเดินทาง' (Start planning the trip). There is also a small checkbox labeled 'หลีกเลี่ยงทางด่วน' (Avoid toll roads) and a '+ เพิ่มจุดหมาย' (Add more destinations) button. The footer of the page says '© 2025 Plan My Trip. All Rights Reserved.'

ภาพที่ 3.8 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (1)

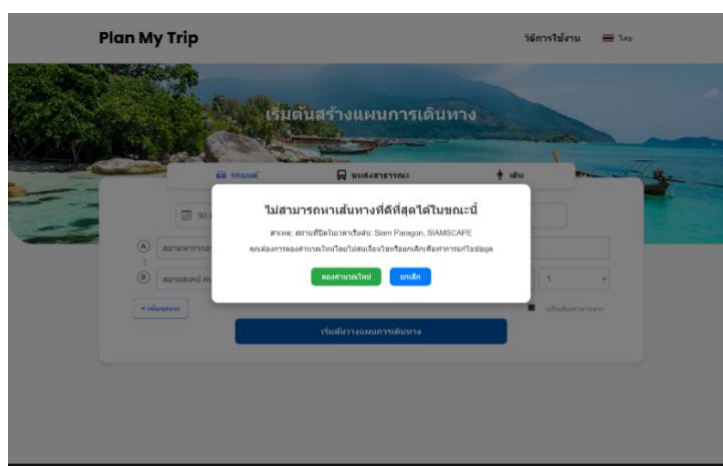
ภาพที่ 3.8 แสดงภาพหน้าการสร้างแผนการเดินทางของเว็บแอปพลิเคชัน (Web Application) Plan My Trip มีฟีเจอร์ในการทำงานได้แก่ การกรอกข้อมูลการเดินทาง การสร้างแผนการเดินทาง และการแจ้งเตือนในกรณีที่สถานที่ใดสถานที่หนึ่งปิดทำการหรือระยะเวลาในแต่ละสถานที่มากเกินไปจนไม่สามารถวางแผนการเดินทางได้ โดยผู้ใช้จะต้องกรอกข้อมูลเพื่อสร้างแผนการเดินทาง ดังนี้

- (1) ผู้ใช้จะต้องเลือกประเภทการเดินทาง โดยเลือกระหว่างการเดินทางโดยรถยนต์และเดินทางเท้า
- (2) ผู้ใช้จะต้องกรอกวันที่ต้องการวางแผนการเดินทางและเวลาที่จะเริ่มเดินทาง
- (3) ผู้ใช้จะต้องกรอกข้อมูลสถานที่ที่ต้องการจะไป อย่างน้อย 2 สถานที่ และสามารถกรอกได้มากที่สุด 10 สถานที่ และจะต้องระบุระยะเวลาที่ผู้ใช้ต้องการจะอยู่ที่สถานที่นั้นๆด้วย โดยสามารถระบุได้ตั้งแต่ 1-4 ชั่วโมง
- (4) หากผู้ใช้เลือกประเภทการเดินทางโดยรถยนต์ ผู้ใช้สามารถกำหนดได้ว่าจะหลีกเลี่ยงค่าผ่านทางหรือไม่
- (5) ผู้ใช้สามารถเลือกได้ว่าจะลือสถานที่หรือไม่ ในกรณีที่ต้องการไปสถานที่นั้นเป็นลำดับที่ผู้ใช้ต้องการ



ภาพที่ 3.9 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (2)

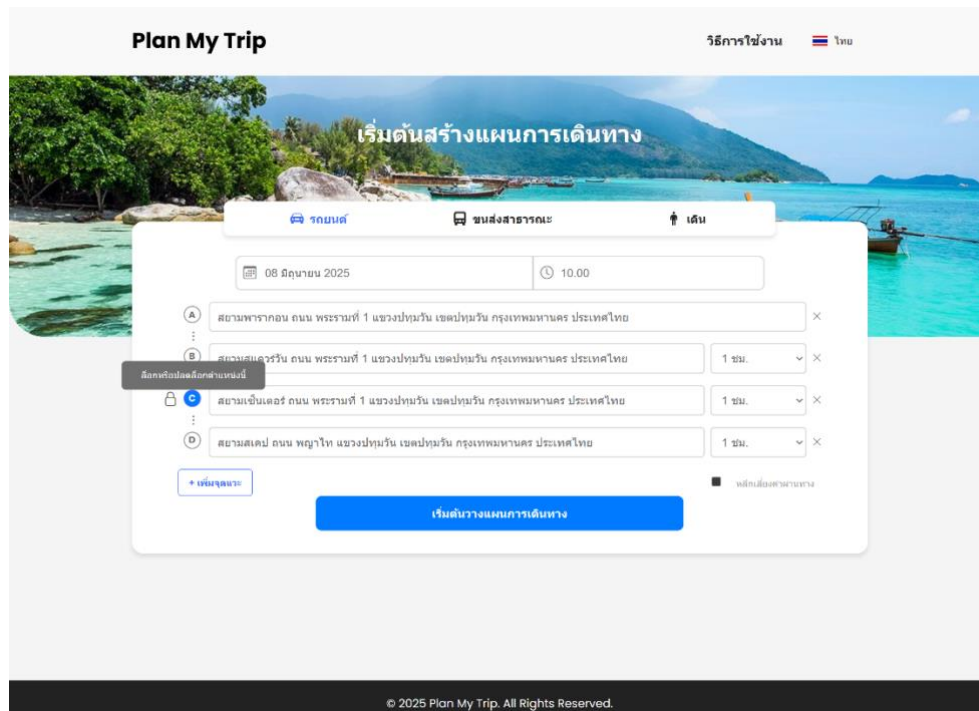
ภาพที่ 3.9 แสดงภาพขั้นตอนการกรอกข้อมูลวันที่ที่เริ่มเดินทางและเวลาที่เริ่มเดินทางของหน้าการสร้างแผนการเดินทางของเว็บแอปพลิเคชัน (Web Application) Plan My Trip



ภาพที่ 3.10 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (3)

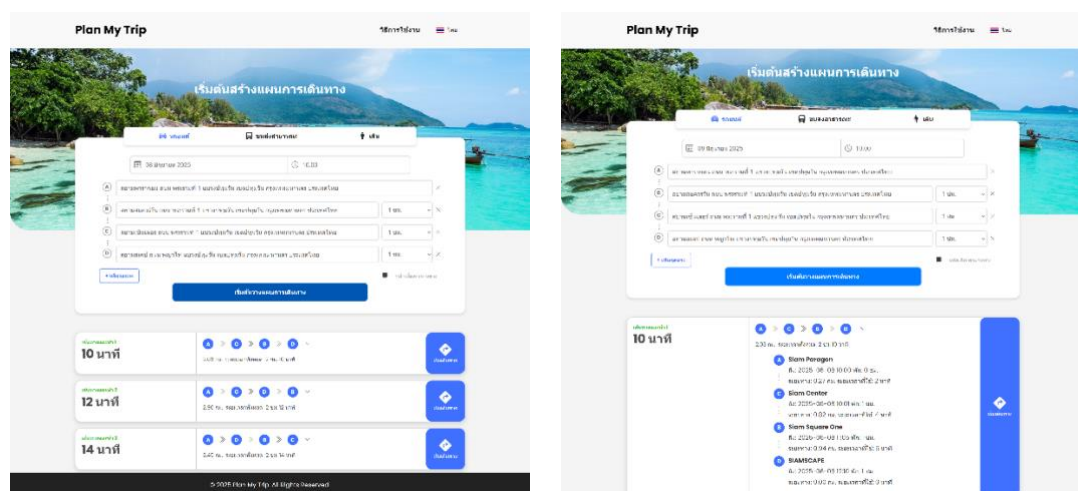
ภาพที่ 3.10 แสดงภาพระบบแจ้งเตือนเมื่อระบบไม่สามารถคำนวณเส้นทางได้ ตามเงื่อนไขที่ผู้ใช้งานกำหนด ในหน้าการสร้างแผนการเดินทางของเว็บแอปพลิเคชัน (Web Application) Plan My Trip โดยแจ้งเตือนจะมีดังนี้

- (1) เมื่อมีสถานที่ปิดทำการในวันที่เดินทาง
- (2) เมื่อไม่สามารถไปได้ครบทุกสถานที่ในวันที่เดินทาง



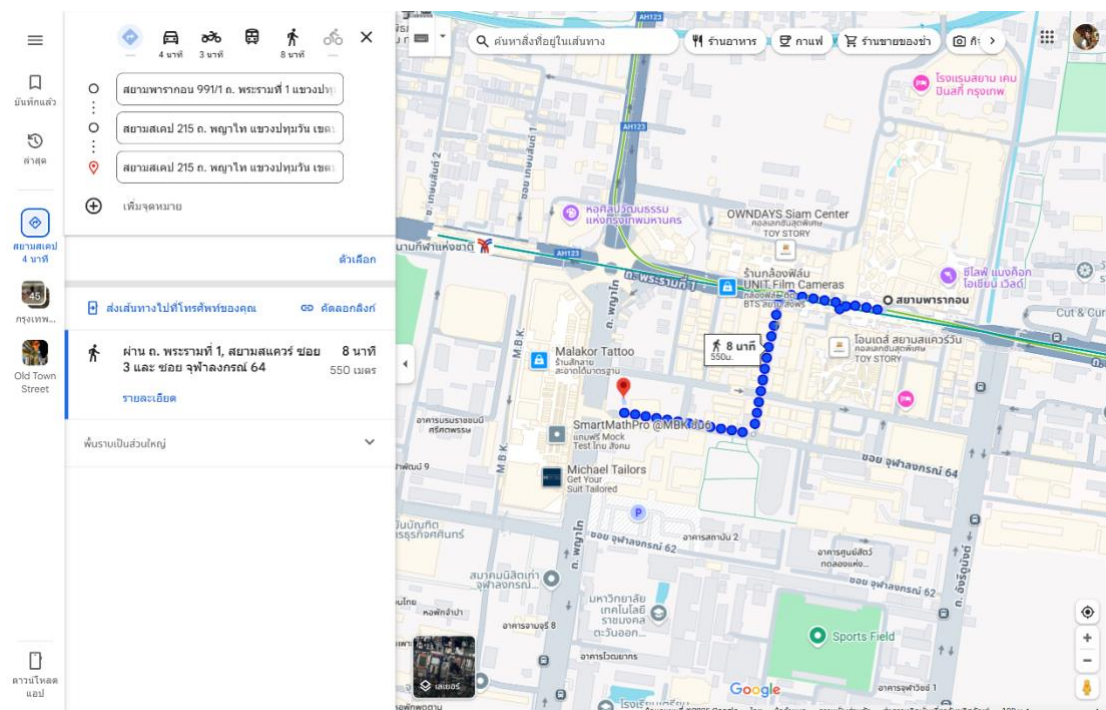
ภาพที่ 3.11 หน้าวางแผนการเดินทางเว็บแอปพลิเคชัน Plan my trip (4)

ภาพที่ 3.10 แสดงภาพการล็อกสถานที่ในกรณีที่ต้องการไปสถานที่นั้นเป็นลำดับที่ผู้ใช้ต้องการ ในหน้าการสร้างแผนการเดินทางของเว็บแอปพลิเคชัน (Web Application) Plan My Trip เช่น จากภาพผู้ใช้ต้องการให้สถานที่ C คือจุดสุดท้ายที่ต้องการจะไป ผู้ใช้จะต้องล็อกสถานที่ C ก่อนกดสร้างแผนการเดินทาง



ภาพที่ 3.12 แสดงภาพผลลัพธ์การวางแผนการเดินทาง

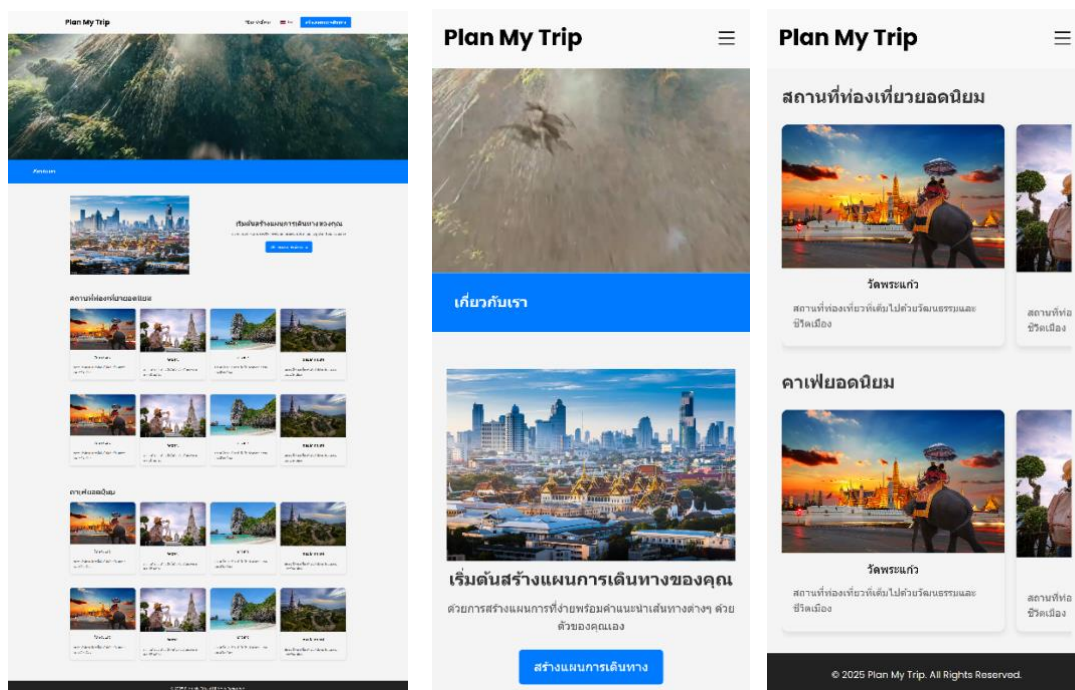
จากภาพที่ 3.12 แสดงภาพหน้าผลลัพธ์การวางแผนการเดินทางของเว็บแอปพลิเคชัน (Web Application) Plan My Trip มีฟีเจอร์ในการทำงานได้แก่ แสดงแผนการเดินทางพร้อมบอกรายละเอียดการเดินทาง การนำแผนการเดินทางไปสู่ Google Map ตามแผนที่ผู้ใช้เลือก อีกทั้งยังสามารถแก้ไขรายละเอียดต่างๆ เพื่อวางแผนการเดินทางอีกครั้ง



ภาพที่ 3.13 ภาพแสดงหน้า Google maps

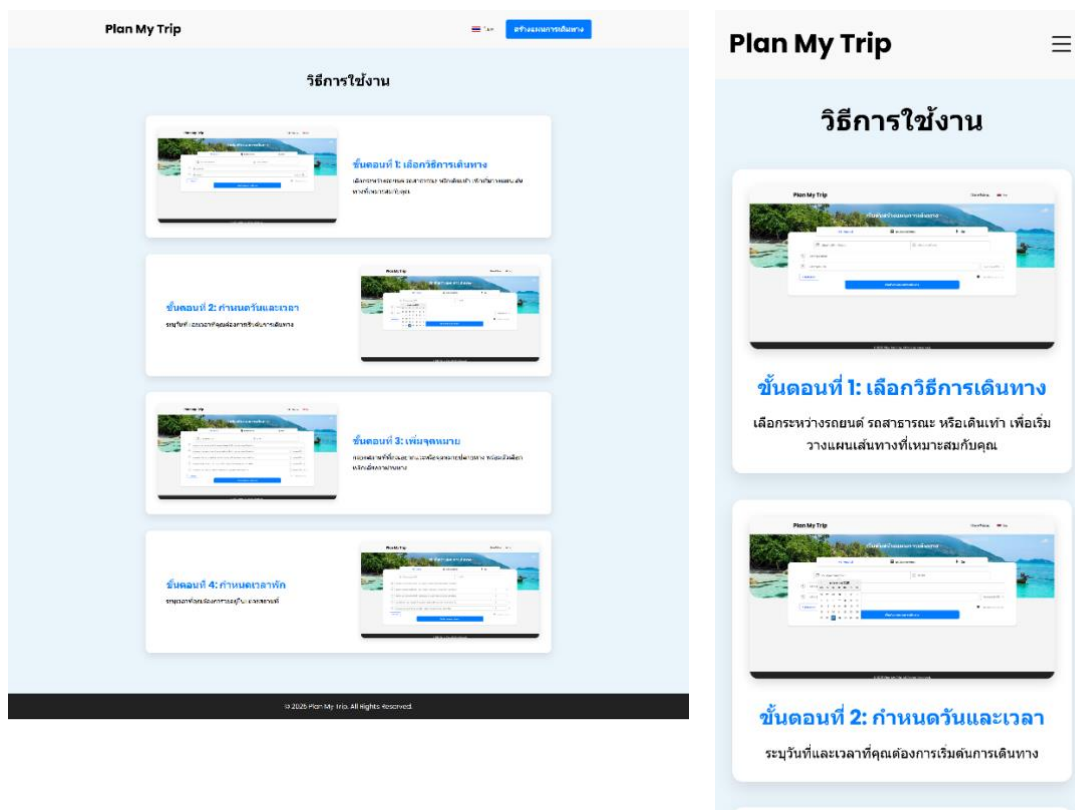
จากภาพที่ 3.13 แสดงภาพเมื่อผู้ใช้กดปุ่มการเดินทางของเส้นทางที่ผู้ต้องการจะเดินทาง ระบบจะนำทางเข้าสู่ Google maps พร้อมข้อมูลเส้นทางที่ผู้ใช้เลือกโดยเรียงลำดับแล้ว

3.9.4 Responsive design



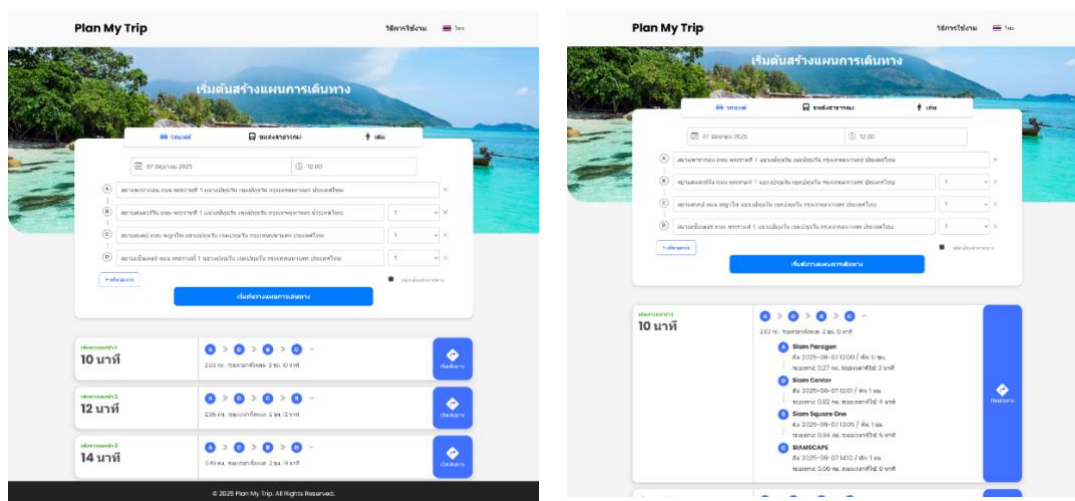
ภาพที่ 3.14 ภาพแสดงหน้าหลักแบบ Responsive

จากภาพที่ 3.14 แสดงภาพการแสดงผลของเว็บแอปพลิเคชันหน้าหลักของเว็บแอปพลิเคชัน Plan my trip เมื่อผู้ใช้งานใช้เว็บแอปพลิเคชันผ่านคอมพิวเตอร์ แท็บเล็ต และ โทรศัพท์มือถือ โดยหน้าหลักจะเป็นหน้าแรกเมื่อผู้ใช้เข้าสู่เว็บแอปพลิเคชัน



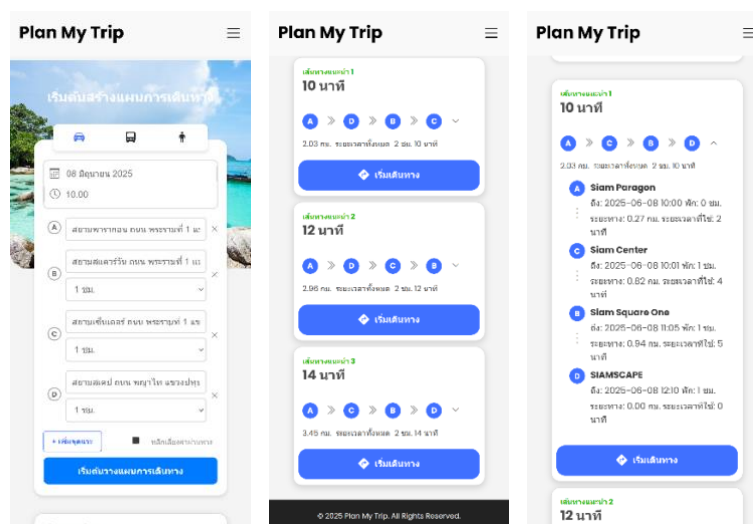
ภาพที่ 3.15 ภาพแสดงหน้าวิธีการใช้งานแบบ Responsive

จากภาพที่ 3.15 แสดงภาพการแสดงผลของเว็บแอปพลิเคชันหน้าวิธีการใช้งาน เมื่อผู้ใช้งานใช้เว็บแอปพลิเคชันผ่านคอมพิวเตอร์ แท็บเล็ต และโทรศัพท์มือถือ โดยหน้าวิธีการใช้งานจะเป็นหน้าที่จะอธิบายวิธีการสร้างแผนการเดินทาง



ภาพที่ 3.16 ภาพแสดงหน้าวางแผนการเดินทางแบบ Responsive (1)

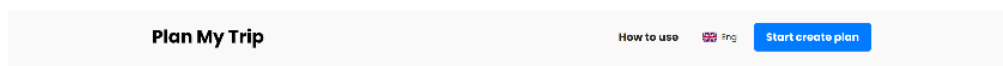
จากภาพที่ 3.16 แสดงภาพการแสดงผลของเว็บแอปพลิเคชันหน้าสร้างแผนการเดินทาง เมื่อผู้ใช้งานใช้เว็บแอปพลิเคชันผ่านคอมพิวเตอร์ แล็บท็อป แท็บเล็ต โดยหน้าสร้างแผนการเดินทางจะเป็นหน้าที่ที่ผู้ใช้จะสร้างแผนเดินทาง และเป็นหน้าที่แสดงผลลัพท์เมื่อระบบประมวลผล



ภาพที่ 3.17 ภาพแสดงหน้าวางแผนการเดินทางแบบ Responsive (2)

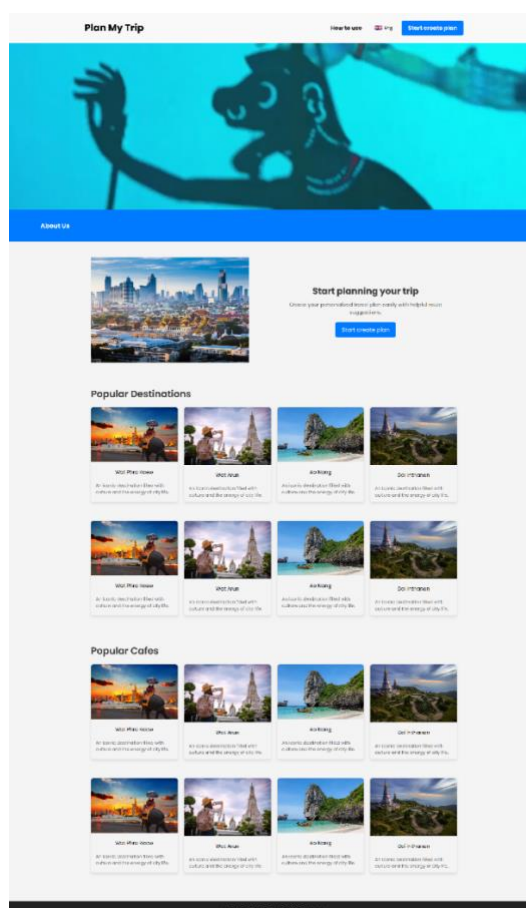
จากภาพที่ 3.17 แสดงภาพการแสดงผลของเว็บแอปพลิเคชันหน้าสร้างแผนการเดินทาง เมื่อผู้ใช้งานใช้เว็บแอปพลิเคชันผ่านโทรศัพท์มือถือ

3.9.5 การเปลี่ยนภาษา



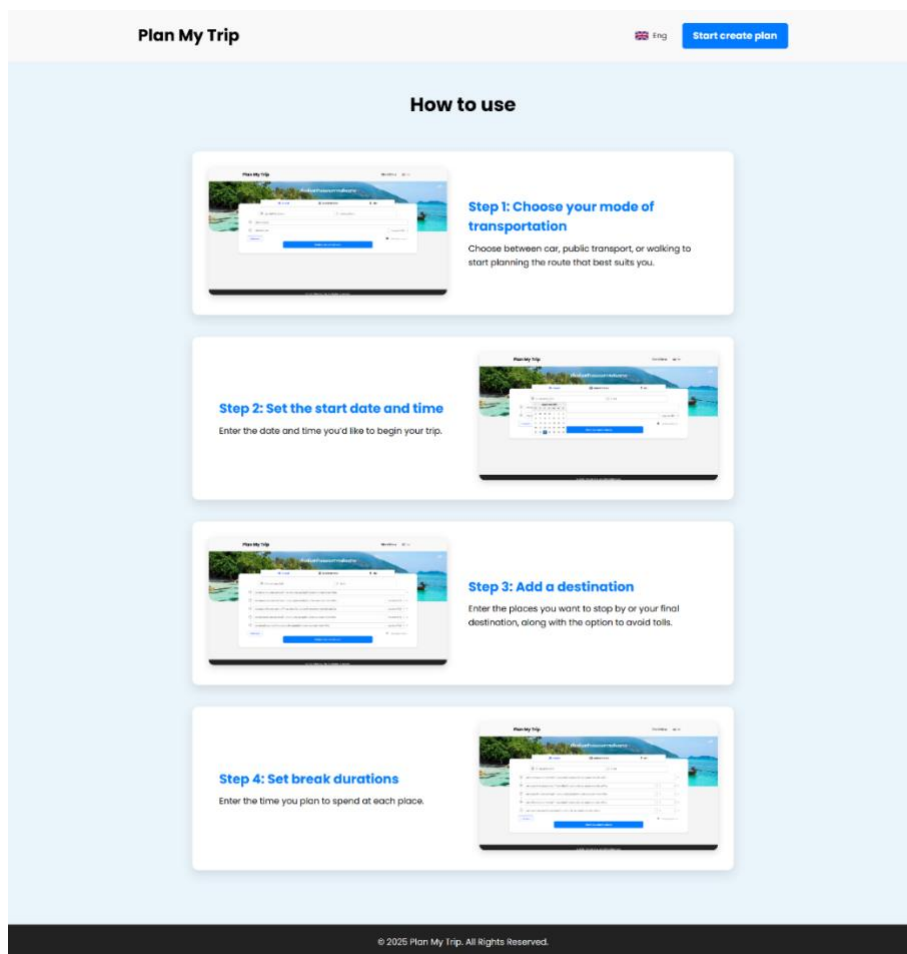
ภาพที่ 3.18 ภาพแสดงปุ่มเปลี่ยนภาษา

จากภาพที่ 3.18 แสดงปุ่มเปลี่ยนภาษาที่อยู่ใน Header ของเว็บแอปพลิเคชัน Plan my trip โดยเว็บแอปพลิเคชัน Plan my trip รองรับภาษาไทยและภาษาอังกฤษเท่านั้น



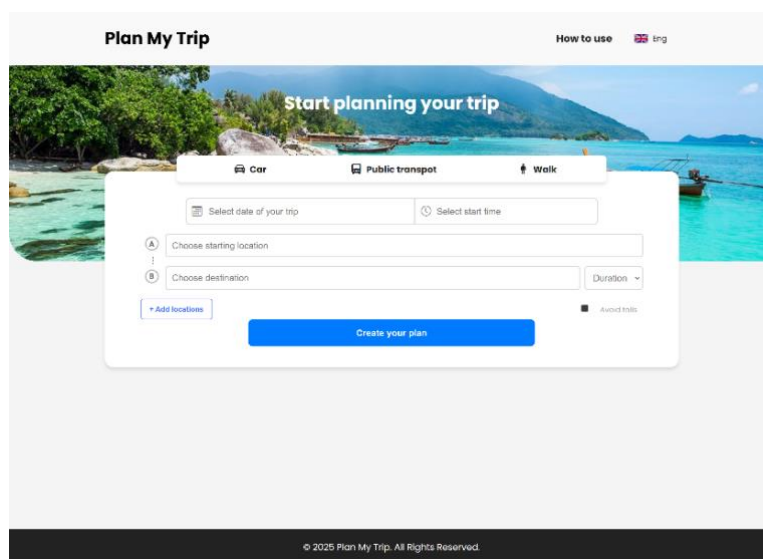
ภาพที่ 3.19 ภาพแสดงหน้าหลักเมื่อเป็นภาษาอังกฤษ

จากภาพที่ 3.19 แสดงภาพหน้าหลักของเว็บแอปพลิเคชัน Plan my trip เมื่อเปลี่ยนภาษาเป็นภาษาอังกฤษ



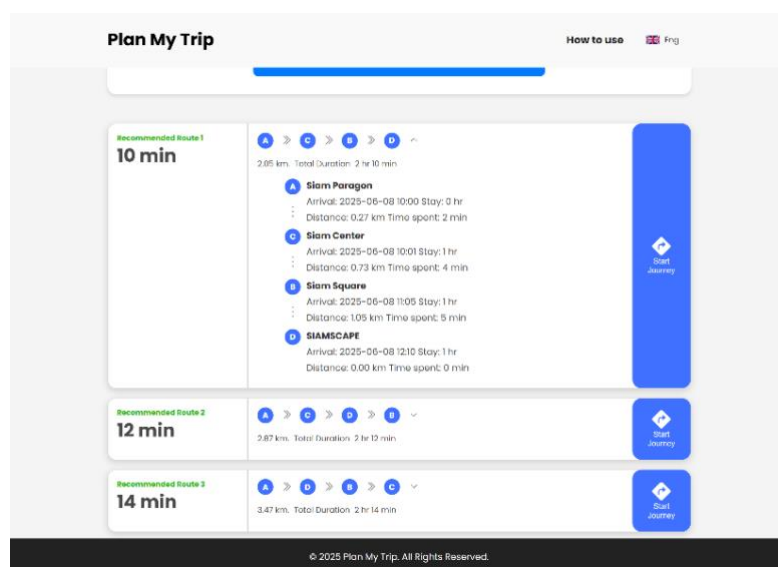
ภาพที่ 3.20 ภาพแสดงหน้าวิธีการใช้งานเมื่อเป็นภาษาอังกฤษ

จากภาพที่ 3.20 แสดงภาพหน้าวิธีการใช้งานของเว็บแอปพลิเคชัน Plan my trip เมื่อเปลี่ยนภาษาเป็นภาษาอังกฤษ



ภาพที่ 3.21 ภาพแสดงหน้าวางแผนการเดินทางเมื่อเป็นภาษาอังกฤษ

จากภาพที่ 3.21 แสดงภาพหน้าวางแผนการเดินทางของเว็บแอปพลิเคชัน Plan my trip เมื่อเปลี่ยนภาษาเป็นภาษาอังกฤษ



ภาพที่ 3.22 ภาพแสดงผลลัพธ์เส้นทางเมื่อเป็นภาษาอังกฤษ

จากภาพที่ 3.22 แสดงภาพผลลัพธ์ 3 เส้นทางที่ดีที่สุดที่สุ่มหน้าวางแผนการเดินทางของเว็บแอปพลิเคชัน Plan my trip เมื่อเปลี่ยนภาษาเป็นภาษาอังกฤษ

บทที่ 4

ผลการดำเนินงาน

4.1 การจัดเตรียมฮาร์ดแวร์และซอฟต์แวร์

4.1.1 เครื่องมือที่ใช้ในการพัฒนา (Software)

- (1) ภาษาที่ใช้ในการพัฒนา: HTML, CSS, JavaScript
- (2) โปรแกรมที่ใช้ในการพัฒนา: VScode, Figma
- (3) Framework ที่ใช้ในการพัฒนา: Bootstrap, React, Express.js

ขั้นตอน	มค.68				กพ.68				มีค.68				เมย.68				พค.68			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.ศึกษาความเป็นไปได้																				
2.ศึกษาเครื่องมือที่ใช้และวางแผนการดำเนินงาน																				
3.พัฒนาระบบ (Frontend – Backend)																				
4.ทดสอบและปรับปรุง																				
5.นำเสนอโครงการ																				

4.3 การทดสอบระบบ

4.3.1 การทดสอบกรณีระหว่างที่ผู้ใช้กรอกข้อมูลสถานที่และเลือกเงื่อนไข

ตารางที่ 4.2 การทดสอบกรณีระหว่างที่ผู้ใช้กรอกข้อมูลสถานที่และเลือกเงื่อนไข

Test Case 01: การทดสอบกรณีผู้ใช้สามารถกรอกข้อมูลสถานที่และเลือกเงื่อนไข			
Test Description		ผู้ใช้กรอกข้อมูลสถานที่ที่ต้องการวางแผน และเงื่อนไขต่างๆ	
Pre-Condition		-	
Post-Condition		ผู้ใช้กรอกข้อมูลสถานที่ที่ต้องการวางแผน และเงื่อนไขต่างๆได้สำเร็จ	
Result		ผ่านการทดสอบ (Pass)	
Case	Action	Expect test	Result (Pass / Fail)
1	ผู้ใช้กำลังกรอกสถานที่เริ่มต้น จุดสิ้นสุด และ จุดแวะ	ระบบแสดงการเติมคำ อัตโนมัติ (Autocomplete)	ผ่านการทดสอบ (Pass)
2	ผู้ใช้กดเลือกวันที่เริ่มเดินทาง	ระบบแสดง Drop- down ของปฏิทินให้ ผู้ใช้เลือก	ผ่านการทดสอบ (Pass)
3	ผู้ใช้กดเลือกเวลาเริ่มเดินทาง	ระบบแสดง Drop- down ของเวลาให้ ผู้ใช้งานเลือก	ผ่านการทดสอบ (Pass)
4	ผู้ใช้กำลังกรอกระยะเวลาที่ใช้	ระบบแสดง Drop- down ของเวลาหลัก ชั่วโมงให้ผู้ใช้งานเลือก โดยมีตั้งแต่ 1-4	ผ่านการทดสอบ (Pass)

4.3.2 การทดสอบการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ

ตารางที่ 4.3 การทดสอบการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ

Test Case 02: การทดสอบการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ			
Test Description	ระบบแจ้งเตือนข้อผิดพลาด เพื่อให้ผู้ใช้แก้ไข หรือคำนวณต่อโดยไม่สนใจ		
Pre-Condition	ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง หลังจากกรอกข้อมูลทั้งหมดแล้ว		
Post-Condition	ผู้ใช้กดยกเลิกเพื่อทำการแก้ไขหรือ กดเลือกคำนวณโดยไม่สนใจ		
Result	ผ่านการทดสอบ (Pass)		
Case	Action	Expect test	Result (Pass / Fail)
1	ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง (อินเทอร์เน็ตขัดข้อง)	มี pop-up แจ้งเตือน ว่าโปรดตรวจสอบ อินเทอร์เน็ต	ผ่านการทดสอบ (Pass)
2	ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง (ผู้ใช้งานกรอกข้อมูลไม่ครบ)	มี pop-up แจ้งเตือน กรุณากรอกข้อมูลให้ ครบก่อนเริ่มวางแผน!	ผ่านการทดสอบ (Pass)
3	ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง (สถานที่ปิดทำการในวันดังกล่าว)	มี pop-up แจ้งเตือน สถานที่ X ปิดทำการ ในวันดังกล่าว คุณ ต้องการลองคำนวณ ใหม่โดยไม่สนใจ หรือไม่ มีปุ่มให้ ผู้ใช้งานเลือก ได้แก่ ลองคำนวณใหม่ และ ยกเลิก	ผ่านการทดสอบ (Pass)

Test Case 02: การทดสอบการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ			
Test Description		ระบบแจ้งเตือนข้อผิดพลาด เพื่อให้ผู้ใช้แก้ไข หรือคำนวณต่อโดยไม่สนใจ	
Pre-Condition		ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง หลังจากที่ยกรอกข้อมูลทั้งหมดแล้ว	
Post-Condition		ผู้ใช้กดยกเลิกเพื่อทำการแก้ไขหรือ กดเลือกคำนวณโดยไม่สนใจ	
Result		ผ่านการทดสอบ (Pass)	
Case	Action	Expect test	Result (Pass / Fail)
4	ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง (ไม่สามารถหาเส้นทางที่ไปได้ภายใต้เงื่อนไขทั้งหมด)	มี pop-up แจ้งเตือน ไม่สามารถหาเส้นทางที่ไปได้ภายใต้เงื่อนไขทั้งหมด คุณต้องการลองคำนวณใหม่โดยไม่สนใจหรือไม่ มีปุ่มให้ผู้ใช้เลือกได้แก่ ลองคำนวณใหม่ และ ยกเลิก	ผ่านการทดสอบ (Pass)

4.3.3 การทดสอบการแสดงผลเส้นทาง

ตารางที่ 4.4 การทดสอบการแสดงผลเส้นทาง

Test Case 03: การทดสอบการแสดงผลเส้นทาง			
Test Description		ระบบแสดงผลเส้นทางที่ดีที่สุด เรียงตามลำดับ	
Pre-Condition		ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง หรือยืนยันลองคำนวณใหม่ หลังจากกรอกข้อมูลทั้งหมดแล้ว	
Post-Condition		ผู้ใช้งานสามารถกด เพื่อดูรายละเอียดเส้นทางได้ หรือทำการ link ไปที่ google map	
Result		ผ่านการทดสอบ (Pass)	
Case	Action	Expect test	Result (Pass / Fail)
1	ระบบแสดงผลเส้นทาง ให้ผู้ใช้งานทราบ (สามารถคำนวณได้ตาม เงื่อนไขทั้งหมด)	แสดงผลเส้นทางที่ดีที่สุด ที่ ตรงตามทุกเงื่อนไข เรียงลำดับจากเส้นทางที่ใช้ เวลาน้อยที่สุด	ผ่านการทดสอบ (Pass)
2	ระบบแสดงผลเส้นทาง ให้ผู้ใช้งานทราบ (ไม่สามารถคำนวณได้ตาม เงื่อนไขทั้งหมด)	แสดงผลเส้นทางที่ดีที่สุด โดย ไม่สนใจเงื่อนไขเรียงลำดับจาก เส้นทางที่ใช้เวลา น้อยที่สุด	ผ่านการทดสอบ (Pass)

Test Case 03: การทดสอบการแสดงผลเส้นทาง			
Test Description		ระบบแสดงผลเส้นทางที่ดีที่สุด เรียงตามลำดับ	
Pre-Condition		ผู้ใช้กดเริ่มต้นวางแผนการเดินทาง หรือยืนยันลองคำนวณใหม่ หลังจากกรอกข้อมูลทั้งหมดแล้ว	
Post-Condition		ผู้ใช้สามารถกด เพื่อดูรายละเอียดเส้นทางได้ หรือทำการ link ไปที่ google map	
Result		ผ่านการทดสอบ (Pass)	
3	<p>ระบบแสดงผลเส้นทาง</p> <p>ให้ใช้งานทราบ</p> <p>(กรณีที่ทำการล๊อคตำแหน่งสถานที่โดยสถานที่ที่ถูกล๊อคจะถูกจัดอยู่ในลำดับที่ล๊อคไว้เสมอ)</p>	<p>การแสดงผลเป็นไปตามการล๊อคสถานที่โดยสถานที่ที่ถูกล๊อคจะจัดอยู่ในตำแหน่งที่ล๊อคเสมอ</p>	<p>ผ่านการทดสอบ</p> <p>(Pass)</p>

4.3.4 การทดสอบความแม่นยำของเส้นทาง (Accuracy)

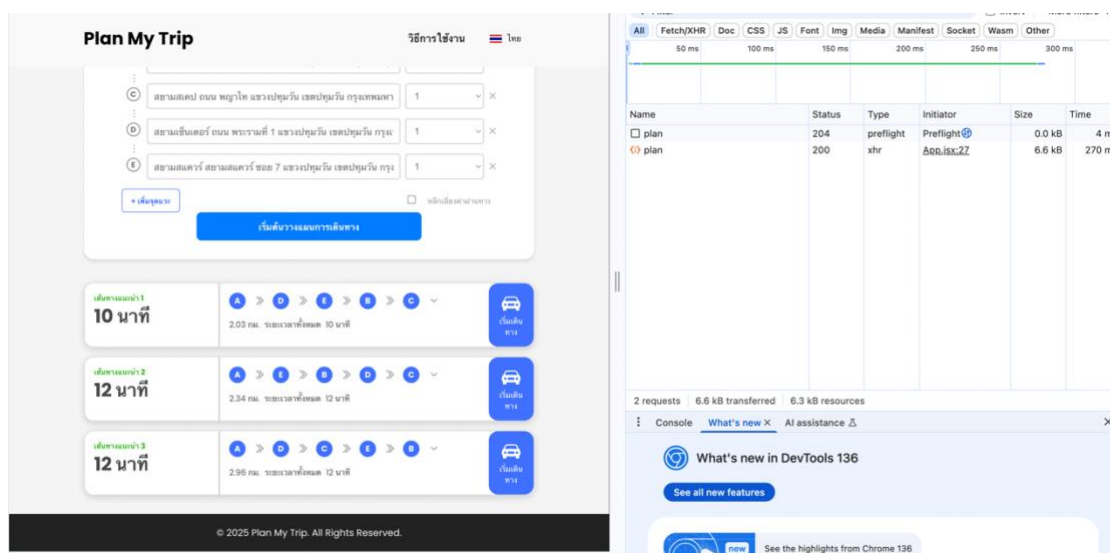
ตารางที่ 4.5 การทดสอบความแม่นยำของเส้นทาง (Accuracy)

Test Case 04: การทดสอบความแม่นยำของเส้นทาง (Accuracy)			
Test Description	ระบบแสดงผลข้อมูลได้อย่างแม่นยำตรงตามเงื่อนไข		
Pre-Condition	ผู้ใช้งานเริ่มต้นวางแผนการเดินทาง หรือยืนยันลองคำนวณใหม่ หลังจากกรอกข้อมูลทั้งหมดแล้ว		
Post-Condition	-		
Result	ผ่านการทดสอบ (Pass)		
Case	Action	Expect test	Result (Pass / Fail)
1	กดเริ่มวางแผนการเดินทาง (ครั้งที่ 1)	แสดงผลลัพธ์เส้นทาง ที่ดีที่สุดจากทั้งหมด เทียบกับ google maps หรือ ตรงตาม การคำนวณแบบทุก เส้นทาง	ผ่านการทดสอบ (Pass)
2	กดเริ่มวางแผนการเดินทาง (ครั้งที่ 2 ทำการสลับตำแหน่งการ กรอกข้อมูลสถานที่ หรือเงื่อนไขต่างๆ ที่ต่างกันอย่างเห็นได้ชัด)	การแสดงผลเปลี่ยนไป จากครั้งแรก เช่น ข้อมูลรายละเอียดการ เดินทาง	ผ่านการทดสอบ (Pass)

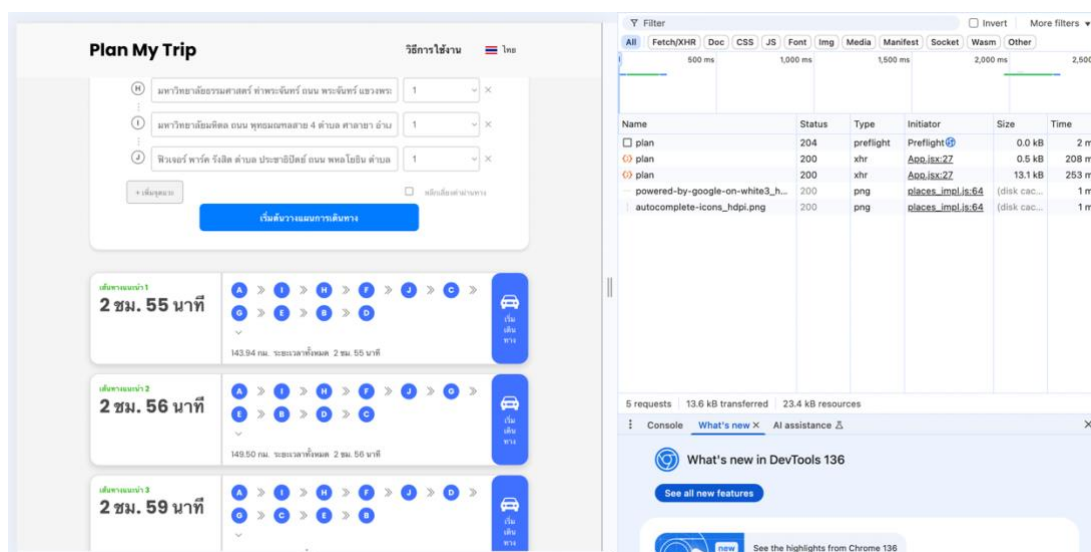
4.3.5 การทดสอบความเร็วในการประมวลผล (Performance)

ตารางที่ 4.6 การทดสอบ ความเร็วในการประมวลผล (Performance)

Test Case 05: ความเร็วในการประมวลผล (Performance)			
Test Description	ระบบแสดงผลข้อมูลได้อย่างรวดเร็วในระยะเวลาที่เหมาะสม		
Pre-Condition	ผู้ใช้งานเริ่มต้นวางแผนการเดินทาง หรือยืนยันการจองค่านวนใหม่ หลังจากกรอกข้อมูลทั้งหมดแล้ว		
Post-Condition	-		
Result	ผ่านการทดสอบ (Pass)		
Case	Action	Expect test	Result (Pass / Fail)
1	กดเริ่มวางแผนการเดินทาง (สำหรับ 5 สถานที่)	ระบบแสดงผลข้อมูล ภายใน 3 วินาที	ผ่านการทดสอบ (Pass)
2	กดเริ่มวางแผนการเดินทาง (สำหรับ 10 สถานที่)	ระบบแสดงผลข้อมูล ภายใน 3 วินาที	ผ่านการทดสอบ (Pass)



ภาพที่ 4.1 ภาพแสดงในการประมวลผลและแสดงผลข้อมูล (1)



ภาพที่ 4.2 ภาพแสดงในการประมวลผลและแสดงผลข้อมูล (2)

4.3.6 การทดสอบการเชื่อมต่อข้อมูลสถานที่ไปยัง google maps

ตารางที่ 4.7 การทดสอบการเชื่อมต่อข้อมูลสถานที่ไปยัง google maps

Test Case 06: การทดสอบการเชื่อมต่อข้อมูลสถานที่ไปยัง google maps			
Test Description	ระบบสามารถเชื่อมต่อข้อมูลสถานที่ไปยัง google maps		
Pre-Condition	ผู้ใช้กดเดินทาง		
Post-Condition	ระบบ link ไปยัง google maps		
Result	ผ่านการทดสอบ (Pass)		
Case	Action	Expect test	Result (Pass / Fail)
1	กดเดินทาง	ระบบ link ข้อมูลสถานที่ไปยัง google maps สามารถเริ่มนำทางได้เลยโดยไม่ต้องกรอกข้อมูลสถานที่อีกครั้ง	ผ่านการทดสอบ (Pass)

4.4 สรุปการประเมินประสิทธิภาพของโครงการ

4.4.1 การประเมินการทำงานของฟังก์ชันต่าง ๆ ในเว็บแอปพลิเคชัน (Functional Test) ด้วยกรณีทดสอบ

ตารางที่ 4.8 กรณีทดสอบการทำงานของฟังก์ชัน

กรณีทดสอบ	ผ่าน	ไม่ผ่าน
ผู้ใช้งานกรอกข้อมูลสถานที่และเลือกเงื่อนไข	✓	
ระบบแจ้งเตือนข้อผิดพลาดให้ผู้ใช้งานทราบ	✓	
ระบบแสดงผลเส้นทางที่ดีที่สุด เรียงตามลำดับ	✓	
ระบบมีความแม่นยำของเส้นทาง (Accuracy)	✓	
ระบบมีความเร็วในการประมวลผล (Performance)	✓	
เชื่อมต่อข้อมูลสถานที่ไปยัง google maps	✓	

บทที่ 5

สรุปผลการดำเนินการและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินการพัฒนาเว็บแอปพลิเคชัน Plan My Trip ซึ่งมีวัตถุประสงค์หลักเพื่อเป็นเครื่องมืออำนวยความสะดวกในการวางแผนการเดินทางสำหรับผู้ใช้งานทั่วไป โดยมุ่งเน้นให้สามารถกำหนดจุดเริ่มต้น จุดหมายปลายทาง จุดแวะพัก รวมถึงระบุวัน เวลา และเงื่อนไขต่าง ๆ ที่เกี่ยวข้องกับการเดินทางได้อย่างยืดหยุ่น เช่น การเลือกวิธีการเดินทางระหว่างรถยนต์หรือเดินเท้า การระบุระยะเวลาที่จะใช้ในแต่ละสถานที่ ตลอดจนการตั้งค่าหลักเลี่ยงค่าผ่านทาง ทั้งนี้เพื่อให้สอดคล้องกับข้อจำกัดและความต้องการเฉพาะตัวของผู้ใช้แต่ละราย

ระบบได้รับการออกแบบโดยคำนึงถึงหลักการใช้งานที่ง่าย สะดวก และมีประสิทธิภาพ โดยมีการจัดทำส่วนติดต่อผู้ใช้ (User Interface) ที่เป็นมิตรต่อผู้ใช้ พร้อมทั้งสามารถรองรับการกรอกข้อมูลได้อย่างหลากหลาย และรองรับกรณีผู้ใช้ต้องการปรับแต่งเงื่อนไขระหว่างการวางแผนได้โดยสะดวก ในกรณีที่ผู้ใช้ป้อนข้อมูลซึ่งไม่สามารถคำนวณเส้นทางที่เป็นไปได้ ระบบจะมีการแจ้งเตือนข้อผิดพลาดให้ผู้ใช้ทราบอย่างชัดเจน เพื่อให้สามารถทำการแก้ไขและดำเนินการต่อได้อย่างราบรื่น

ในด้านเทคนิค ระบบได้นำเทคโนโลยี Google Maps API มาใช้ในการดึงข้อมูลระยะทางและระยะเวลาเดินทางจริง พร้อมประยุกต์ใช้อัลกอริทึม A* (A-Star Search Algorithm) เพื่อใช้ในการค้นหาเส้นทางที่เหมาะสมที่สุดภายใต้เงื่อนไขที่ผู้ใช้กำหนด โดยระบบสามารถพิจารณาเวลาเปิด-ปิดของสถานที่ต่าง ๆ และตรวจสอบความเป็นไปได้ของการเดินทางตามลำดับจุดแวะที่ระบุ ทั้งนี้ยังสามารถจัดเรียงลำดับจุดแวะใหม่ให้เหมาะสมที่สุดโดยอัตโนมัติ เพื่อให้สอดคล้องกับข้อจำกัดด้านเวลาและความเป็นไปได้ในเชิงพื้นที่

จากการทดสอบการใช้งานในกลุ่มเป้าหมาย พบว่าผู้ใช้สามารถทำความเข้าใจและใช้งานระบบได้อย่างมีประสิทธิภาพ โดยไม่จำเป็นต้องมีคำแนะนำเพิ่มเติม อีกทั้งระบบสามารถประมวลผลข้อมูลและแสดงผลลัพธ์ได้อย่างรวดเร็ว มีความแม่นยำของเส้นทางใกล้เคียงกับบริการชั้นนำ เช่น Google Maps และสามารถรองรับการปรับแต่งแผนการเดินทางได้อย่างหลากหลาย ไม่ว่าจะเป็นการเพิ่มหรือลดจุดแวะ การเปลี่ยนแปลงระยะเวลาที่ใช้ในแต่ละสถานที่ ตลอดจนการเลือกใช้วิธีการเดินทางที่แตกต่างกันในแต่ละช่วง

สรุปได้ว่า เว็บแอปพลิเคชัน Plan My Trip ที่พัฒนาขึ้นสามารถตอบสนองต่อความต้องการของผู้ใช้งานได้อย่างครอบคลุม มีจุดเด่นในด้านความสามารถในการจัดการเงื่อนไขที่ซับซ้อน ความแม่นยำในการคำนวณเส้นทาง ความรวดเร็วในการประมวลผล และการใช้งานที่เข้าใจง่าย โดยสามารถนำไปใช้เป็นเครื่องมือในการวางแผนการเดินทางได้จริง ช่วยลดระยะเวลาและความยุ่งยากในการจัดการเส้นทาง อีกทั้งยังมีศักยภาพในการพัฒนาต่อยอดเพื่อรองรับความต้องการที่ซับซ้อนมากยิ่งขึ้นในอนาคต

5.2 ปัญหาและอุปสรรคที่พบ

ในการพัฒนาเว็บแอปพลิเคชัน PlanMyTrip มีการพบปัญหาและอุปสรรคที่สำคัญหลายประการทั้งในด้านเทคนิค และด้านการออกแบบระบบ เพื่อให้ระบบสามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างครอบคลุม ดังนี้

5.2.1 ข้อจำกัดของข้อมูลจาก Google Maps API

ระบบต้องพึ่งพาข้อมูลจาก Google Maps API ในการประมวลผลระยะทางและเวลาเดินทาง รวมถึงข้อมูลเวลาเปิด-ปิดของสถานที่ ซึ่งข้อมูลบางแห่งไม่ครอบคลุมหรือไม่มีข้อมูลเวลาเปิด-ปิดอย่างชัดเจน ส่งผลให้การตรวจสอบความเป็นไปได้ของเส้นทางในบางกรณีไม่สามารถดำเนินการได้อย่างสมบูรณ์

5.2.2 ความซับซ้อนในการประมวลผลเงื่อนไขเวลาเปิด-ปิด

การจัดลำดับเส้นทางตามเวลาเปิด-ปิดของแต่ละสถานที่ที่มีความซับซ้อน โดยเฉพาะในกรณีที่เวลาปิดของสถานที่ข้ามวัน หรือเปิดตลอด 24 ชั่วโมง ระบบจำเป็นต้องออกแบบเงื่อนไขเฉพาะในการคำนวณเพื่อรองรับรูปแบบเวลาเหล่านี้ ซึ่งใช้เวลาในการพัฒนาและทดสอบมากกว่าที่คาดการณ์

5.2.3 การคำนวณลำดับเส้นทางที่เหมาะสมภายใต้เงื่อนไขเฉพาะผู้ใช้

แม้ว่าจะมีการประยุกต์ใช้อัลกอริทึม A* เพื่อคำนวณเส้นทาง แต่การรองรับเงื่อนไขเฉพาะ เช่น การระบุจุดแวะที่ต้องไปก่อนหรือหลัง (fixed sequence) รวมถึงระยะเวลาใช้ในแต่ละสถานที่ ทำให้กระบวนการวางแผนต้องมีการประมวลผลแบบ iterative และตรวจสอบความเป็นไปได้ของแต่ละเส้นทางอย่างละเอียด ซึ่งเพิ่มความซับซ้อนของระบบ

5.2.4 การจัดการข้อผิดพลาดและแจ้งเตือนผู้ใช้

การออกแบบระบบให้สามารถตรวจสอบและแจ้งเตือนผู้ใช้กรณีข้อมูลที่ป้อนไม่สามารถนำไปคำนวณได้ตามเงื่อนไขที่ตั้งไว้ ต้องอาศัยการตรวจสอบที่รัดกุมและการจัดการข้อความแจ้งเตือนอย่างชัดเจน เพื่อไม่ให้ผู้ใช้เกิดความสับสนหรือเข้าใจผิด

5.2.5 การประเมินเส้นทางที่ดีที่สุดเชิงคุณภาพ

แม้ระบบจะสามารถแสดงเส้นทางที่เป็นไปได้ แต่การประเมินว่าเส้นทางใด “ดีที่สุด” สำหรับผู้ใช้บางราย อาจต้องอาศัยความรู้สึกส่วนตัว เช่น ต้องการหลีกเลี่ยงการเดินมากเกินไป หรือต้องการแวะพักในช่วงเวลาใดเวลาหนึ่ง ซึ่งระบบยังไม่สามารถประมวลผลในเชิงพฤติกรรมได้ทั้งหมด

5.3 ข้อเสนอแนะและสิ่งที่สามารถพัฒนาต่อยอด

เนื่องจากการพัฒนาเว็บแอปพลิเคชัน Plan My Trip ซึ่งเป็นระบบช่วยวางแผนการเดินทางที่เน้นการคำนวณเส้นทางอย่างเหมาะสมตามข้อจำกัดของผู้ใช้งาน เช่น เวลาเปิด-ปิดของสถานที่ ระยะเวลาที่ใช้ในแต่ละจุด และเงื่อนไขการเดินทางต่าง ๆ คณะผู้จัดทำเห็นว่าระบบดังกล่าวยังสามารถต่อยอดและพัฒนาให้เป็นระบบที่มีความสมบูรณ์ยิ่งขึ้น และสามารถตอบโจทย์ความต้องการของผู้ใช้งานได้อย่างรอบด้านมากขึ้นในอนาคต โดยมีข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาระบบต่อไป ดังนี้

5.3.1 การประยุกต์ใช้ปัญญาประดิษฐ์ (AI) หรือ Machine Learning

ควรนำระบบ AI มาวิเคราะห์ข้อมูลพฤติกรรมของผู้ใช้งาน รวมถึงปรับปรุงการประเมินเส้นทางที่เหมาะสมในระดับรายบุคคล เพื่อให้คำแนะนำที่สอดคล้องกับความต้องการของผู้ใช้ในแต่ละกรณีมากยิ่งขึ้น เช่น การเลือกเส้นทางที่หลีกเลี่ยงรถติดหรือเลือกช่วงเวลาที่ใช้เดินทางน้อย

5.3.2 การเชื่อมต่อกับแหล่งข้อมูลภายนอกเพิ่มเติมการประยุกต์ใช้ปัญญาประดิษฐ์ (AI) หรือ Machine Learning

ควรขยายความสามารถของระบบในการดึงข้อมูลจากบริการภายนอก เช่น ระบบขนส่งสาธารณะ ระบบจราจร หรือข้อมูลสถานที่จากแอปพลิเคชันท่องเที่ยวต่าง ๆ เพื่อเพิ่มความแม่นยำของเวลาเดินทาง และอัปเดตข้อมูลสถานที่ให้เป็นปัจจุบันอยู่เสมอ

5.3.3 การเพิ่มระบบบันทึกประวัติการเดินทาง

เสนอให้พัฒนาฟีเจอร์ที่สามารถบันทึกการเดินทางย้อนหลัง เพื่อให้ผู้ใช้งานสามารถเรียกดูเส้นทางที่เคยวางแผนไว้ รวมถึงสามารถใช้ข้อมูลดังกล่าวในการปรับแผนการเดินทางในอนาคตได้สะดวกยิ่งขึ้น

5.3.4 การออกแบบระบบให้รองรับหลายภาษา (Multilingual Interface)

เพื่อให้ระบบสามารถเข้าถึงกลุ่มผู้ใช้งานที่หลากหลายยิ่งขึ้น โดยเฉพาะนักท่องเที่ยวจากต่างประเทศ ควรพัฒนาระบบให้สามารถแสดงผลและใช้งานได้หลายภาษา พร้อมคำอธิบายการใช้งานที่ชัดเจน

5.3.5 เพิ่มระบบแนะนำเส้นทางอัตโนมัติตามสถานการณ์ (Smart Suggestion)

ควรมีระบบที่สามารถแนะนำเส้นทางอัตโนมัติในกรณีที่ผู้ใช้ใส่ข้อมูลไม่ครบถ้วน หรือหากเส้นทางที่เลือกไม่สามารถดำเนินการได้ตามเงื่อนไข ระบบควรเสนอทางเลือกที่ใกล้เคียง เพื่อให้ผู้ใช้สามารถปรับเปลี่ยนแผนได้โดยสะดวก

5.3.6 เพิ่มการแสดงผลทางเลือกในการเดินทางที่หลากหลาย

ควรมีระบบแสดงทางเลือกในการเดินทางหลายรูปแบบ เช่น ทางเลือกที่เร็วที่สุด ทางเลือกที่ประหยัดค่าใช้จ่ายที่สุด หรือทางเลือกที่เหมาะสมกับการพักผ่อน เพื่อให้ผู้ใช้สามารถเปรียบเทียบและเลือกเส้นทางได้อย่างตรงตามวัตถุประสงค์

5.3.7 การขยายระบบเพื่อรองรับการวางแผนเดินทางแบบหลายวัน (Multi-day Trip Planning)

ในเวอร์ชันปัจจุบัน ระบบ Plan My Trip ออกแบบมาเพื่อรองรับการเดินทางแบบหนึ่งวัน (One Day Trip) เท่านั้น อย่างไรก็ตาม ผู้ใช้งานบางกลุ่ม เช่น นักท่องเที่ยวหรือนักเดินทางที่มีแผนการท่องเที่ยวระยะยาว อาจมีความต้องการวางแผนการเดินทางในช่วงเวลาหลายวันติดต่อกัน คณะผู้จัดทำจึงเห็นควรเสนอให้พัฒนาระบบเพิ่มเติม เพื่อรองรับแผนการเดินทางแบบหลายวัน

ระบบควรสามารถแยกการวางแผนตามวันที่แตกต่างกัน โดยผู้ใช้งานสามารถระบุสถานที่ จุดแวะพัก และระยะเวลาที่ใช้ในแต่ละวันได้อย่างอิสระ พร้อมทั้งคำนวณเส้นทางให้เหมาะสมตามข้อจำกัดรายวัน เช่น เวลาเปิด-ปิดของสถานที่ ความเหมาะสมของระยะทางระหว่างแต่ละจุดในวันนั้น ๆ และเงื่อนไขของผู้ใช้ เช่น ความต้องการที่พักในแต่ละคืน

การรองรับการเดินทางหลายวันจะช่วยเพิ่มความยืดหยุ่นในการใช้งาน และทำให้ระบบสามารถตอบสนองความต้องการในบริบทที่หลากหลายขึ้น เช่น การท่องเที่ยวเชิงวัฒนธรรม การเดินทางเพื่อธุรกิจ หรือการเดินทางแบบครอบครัวที่มีแผนซับซ้อน โดยไม่จำกัดเพียงแค่การเดินทางในหนึ่งวัน

5.3.8 เงื่อนไขการเดินทางด้วย รถยนต์ หรือรถสาธารณะ

ปัจจุบันแสดงการคำนวณเป็นการเลือกรูปแบบการเดินทางเดียวตลอดทั้งการเดินทาง ในอนาคตสามารถพัฒนาให้ระบุแต่ละจุดแวะของการเดินทางว่าเป็นการเดินทางประเภทใดด้วย เช่น หากสองสถานที่มีระยะห่างไม่มาก อาจจะเสนอให้ผู้ใช้งานเลือกเดินทางด้วยการเดินเท้าไปที่จุดแวะนั้น

5.3.9 การคำนวณหาลำดับเส้นทางที่ดีที่สุด

การคำนวณหาลำดับเส้นทางที่ดีที่สุดนี้ ไม่ได้นำเวลาในการรอคอยมาคำนวณด้วย เช่น เวลาประมาณในการรอคอยรถสาธารณะ หรือ เวลาประมาณในการหาที่จอดรถ ในอนาคตสามารถพัฒนาให้ระบบนำมาใช้ในการคำนวณได้

5.3.10 ในกรณีที่ไม่สามารถหาเส้นทางที่ดีที่สุดได้ตรงกับเงื่อนไขที่ผู้ใช้กำหนด

ปัจจุบันระบบจะเสนอทางเลือกกว่าผู้ใช้งานต้องการแก้ไขเงื่อนไขหรือไม่ ถ้าไม่ต้องการทำการแก้ไขระบบจะคำนวณเส้นทางที่ดีที่สุดโดยไม่สนใจเงื่อนไข ในอนาคตสามารถพัฒนาให้ระบบยังสามารถเลือกเส้นทางที่ดีที่สุดซึ่งอาจไม่ตรงกับเงื่อนไขทั้งหมด แต่ใกล้เคียงกับเงื่อนไขที่ผู้ใช้งานกำหนดให้มากที่สุด

รายการอ้างอิง

- [1] @FordAntiTrust. “สุดมันส์กับ A* Search Algorithm - _-”. จาก <https://www.thaicyberpoint.com/ford/blog/id/128/> [สืบค้นเมื่อ 9 พฤศจิกายน 2567]
- [2] BornToDev. “ความรู้เบื้องต้นเกี่ยวกับ Express.js และการใช้งาน”. จาก <https://www.borntodev.com/2023/11/04/express-js-คืออะไร/> [สืบค้นเมื่อ 1 พฤศจิกายน 2567]
- [3] BornToDev. “ทำความรู้จักกับ React และการใช้งานเบื้องต้น”. จาก https://expert-programming-tutor.com/tutorial/article/KC0030046002_Astar_Algorithm_in_Java.php [สืบค้นเมื่อ 1 พฤศจิกายน 2567]
- [4] expert-programming-tutor. “A* Algorithm การค้นหาทางลัดไปยังจุดหมายในโลกการเขียนโปรแกรม”. จาก https://expert-programming-tutor.com/tutorial/article/KC0030046002_Astar_Algorithm_in_Java.php [สืบค้นเมื่อ 10 พฤศจิกายน 2567]
- [5] Google Maps Platform. “Custom Map Tool & Product”. จาก <https://mapsplatform.google.com/maps-products/#maps-section> [สืบค้นเมื่อ 5 พฤศจิกายน 2567]
- [6] Jedsada Saengow. (26 พฤษภาคม 2561). “[React Native] คืออะไร ทำความรู้จักและเริ่มต้นสร้าง Project”. จาก <https://medium.com/jed-ng/react-native-ทำความรู้จัก-และ-เริ่มต้นสร้าง-project-91788ef6cac3> [สืบค้นเมื่อ 12 พฤศจิกายน 2567]
- [7] MarcusCode. “การใช้งาน Express.js บน Node.js”. จาก <https://marcuscode.com/tutorials/nodejs/using-expressjs> [สืบค้นเมื่อ 1 พฤศจิกายน 2567]
- [8] Supanus Poolthaveetham. “Web Application คืออะไร? แตกต่างจากเว็บไซต์ธรรมดาอย่างไร?”. จาก <https://www.criclabs.co/post/what-is-web-application> [สืบค้นเมื่อ 1 พฤศจิกายน 2567]

[9] Swiftlet. (20 กรกฎาคม 2558). “Google Map API คืออะไร?”. จาก <https://swiftlet.co.th/google-api-คืออะไร/> [สืบค้นเมื่อ 5 พฤศจิกายน 2567]

[10] TechUp. (26 มกราคม 2566). “รู้จักกับ React เครื่องมือจำเป็นสำหรับ Front-end Developer”. จาก <https://www.techupth.com/articles/react> [สืบค้นเมื่อ 4 พฤศจิกายน 2567]