วารสารบัณฑิตศึกษา มหาวิทยาลัยธุรกิจบัณฑิตย์
ปีที่ 13 ฉบับที่ 1 (สิงหาคม 2567-พฤศจิกายน 2567)

การพัฒนาเว็บแอพพลิเคชันแหล่งรวมพระเครื่อง (LANGPRA)

DEVELOPMENT OF AMULET COLLECTION WEB APPLICATION (LANGPRA)

เหนือชล แก้วจิตรตาวงษ์[1]
รศ.ดร.วราพร จิระพันธุ์ทอง[2]

บทคัดย่อ

การศึกษาเพื่อพัฒนาเว็บแอปพลิเคชันแหล่งรวมพระเครื่อง (Langpra) โดยใช้ Next.js และทดสอบประสิทธิภาพของแอปพลิเคชันดังกล่าว เนื่องจากในปัจจุบันการเข้าถึงข้อมูลเกี่ยวกับพระเครื่องเป็นไปได้อย่างลำบากและยากต่อการเข้าถึงโดยเฉพาะในกลุ่มคนรุ่นใหม่ เนื่องจากข้อมูลส่วนใหญ่อยู่ในหนังสือสมัยเก่า ขณะเดียวกันเทคโนโลยีและการใช้อินเทอร์เน็ตมีบทบาทสำคัญในชีวิตประจำวันมากขึ้น ผู้คนจึงหันมาใช้อินเทอร์เน็ตเป็นช่องทางหลักในการค้นหาข้อมูล การศึกษาขั้นตอนและกระบวนการพัฒนาเว็บแอปพลิเคชัน การออกแบบและพัฒนาระบบโดยใช้เทคโนโลยี Next.js และการทดสอบประสิทธิภาพของระบบเพื่อประเมินความสามารถในการใช้งานจริง

ผลการวิจัยพบว่าเว็บแอปพลิเคชัน Langpra ที่พัฒนาขึ้นสามารถรวบรวมและจัดการข้อมูลพระเครื่องได้อย่างมีประสิทธิภาพ ผู้ใช้สามารถค้นหาข้อมูลพระเครื่องได้สะดวกและรวดเร็ว แอปพลิเคชันนี้ยังมีความเสถียรและสามารถรองรับผู้ใช้จำนวนมากได้ ผู้ใช้ส่วนใหญ่พึงพอใจในเว็บแอพพลิเคชัน Langpra โดยมองว่ามีความง่ายต่อการใช้งานและเป็นประโยชน์ในการใช้งานจริง ข้อเสนอแนะที่ได้รับส่วนใหญ่เป็นบวก ผลการทดสอบแสดงให้เห็นว่า Langpra มีประสิทธิภาพในการโหลดข้อมูลและมีการตอบสนองที่ดีในการใช้งานจริง ประโยชน์ที่คาดหวังจากการพัฒนาเว็บแอปพลิเคชันนี้คือ การเพิ่มความสะดวกสบายในการเข้าถึงข้อมูลพระเครื่องสำหรับผู้ใช้ทั่วไป และการส่งเสริมการใช้งานเทคโนโลยีในด้านการศึกษาพระเครื่อง นอกจากนี้ แอปพลิเคชัน Langpra ยังเป็นเครื่องมือที่ช่วยเสริมสร้างทักษะการใช้งานคอมพิวเตอร์และเทคโนโลยีให้กับผู้ใช้ ซึ่งเป็นประโยชน์ในด้านการเรียนรู้และการศึกษาต่อไปในอนาคต

คำสำคัญ: แหล่งรวมพระเครื่อง, เว็บแอพพลิเคชัน

---

[1] นักศึกษาหลักสูตรวิทยาศาสตรมหาบัณฑิตสาขาวิชาวิสวกรรมเว็บและการพัฒนาแอปพลิเคชันบนอุปกรณ์พกพา

[2] อาจารย์ที่ปรึกษา วิทยาลัยครีเอทีฟดีไซน์ แอนด์ เอ็นเตอร์เทนเมนต์เทคโนโลยี

1.บทนำ

1.1 ที่และความสำคัญ

ปัจจุบันพระเครื่องได้มีความเสื่อมถอยความนิยมลงและเข้าถึงกลุ่มคนรุ่นใหม่ได้ยากขึ้นจากการที่เทคโนโลยีมีการพัฒนาไปอย่างรวดเร็ว ทำให้คนสนใจหนังสือน้อยลงและใช้อินเตอร์เน็ตมากขึ้น ช่วยอำนวยความสะดวกจนกลายเป็นสิ่งที่เราต้องใช้ในชีวิตประจำวันและคอมพิวเตอร์ได้มีบทบาทในการดำรงชีวิตของเรามากในปัจจุบัน ทั้งในโรงเรียน มหาวิทยาลัย รวมไปถึงองค์กรใหญ่ ๆ ที่ใช้คอมพิวเตอร์เข้ามาในการเก็บข้อมูล จัดการเอกสารต่าง ๆ การคำนวน และการค้นหาข้อมูล และอื่น ๆ การสะสมและศึกษาพระเครื่องถือเป็นวัฒนธรรมและความสนใจที่สำคัญของคนไทย ซึ่งมีคุณค่าทั้งในด้านประวัติศาสตร์ ศิลปะ และศาสนา พระเครื่องไม่เพียงแต่เป็นวัตถุมงคลที่มีความหมายเชิงจิตวิญญาณ แต่ยังเป็นสัญลักษณ์ของศิลปะและวัฒนธรรมไทยที่มีประวัติยาวนาน อย่างไรก็ตาม การเข้าถึงข้อมูลและการศึกษาเกี่ยวกับพระเครื่องนั้นยังคงมีข้อจำกัด ข้อมูลเกี่ยวกับพระเครื่องนั้นจะอยู่ในหนังสือสมัยเก่าเป็นส่วนใหญ่ ทำให้การจัดหาข้อมูลเป็นไปได้อย่างลำบาก แต่ทั้งนี้เว็บแอปพลิเคชันต่าง ๆ ที่มีหลากหลาย มักจะมีปัญหาอยู่บ้างให้พบเจอ หากอยากนำมาใช้โดยมีคนใช้งานนั้น ยังคงไม่ตอบโจทย์ในเรื่องของการใช้งาน เช่น ระบุไม่ได้ว่า ใครเป็นคนส่งข้อความ ไม่มีการบอกอะไรเลยเรื่องช่องทางที่ทุกคนต้องรู้ การดูย้อนหลังไม่เก็บข้อมูล ไม่ค่อยส่วนตัว ทำให้ข้อมูลสำคัญอยู่คนละที่ ก็เพราะข้อมูลทุกอย่างนั้นถูกเก็บไว้ในเซิร์ฟเวอร์ของผู้ให้บริการ Platform อย่างที่กล่าว จึงยากมาก ๆ ที่จะต้องนำข้อมูลนั้นมาดูแลเก็บดูแลรักษาและวิเคราะห์ เพื่อให้ไว้ซึ่งประโยชน์ด้านอื่น ๆ อีกมากมาย และเสี่ยงต่อการถูกนำไปใช้โดยไม่ได้รับขอก่อน เพราะอาจเป็นข้อมูลที่เอาไว้ใช้ลับ ๆ ในองค์กรหรือส่วนตัว

ในยุคนี้ธุรกิจในวงการพระเครื่องในประเทศไทยเป็นอุตสาหกรรมที่มีมูลค่ามหาศาลและเติบโต อย่างต่อเนื่อง โดยในปี 2562 มูลค่าธุรกิจพระเครื่องถูกประเมินไว้อยู่ระหว่าง 17,000 - 23,000 ล้านบาท [1] ซึ่งมีการเติบโตทุกปีเฉลี่ยประมาณ 10-20% โดยความนิยมในพระเครื่องไม่ได้จำกัดเฉพาะในประเทศเท่านั้น แต่ยังขยายไปสู่ตลาดต่างประเทศ เช่น จีน ซึ่งนักสะสมจากจีนให้ความสนใจอย่างมากกับพระเครื่องที่มีพุทธคุณด้านโชคลาภและการป้องกันภัย รวมถึงเทคโนโลยีและอินเทอร์เน็ตมีบทบาทสำคัญในชีวิตประจำวัน แค่ขาดการเข้าถึงได้ง่ายเพียงเท่านั้น จึงต้องมีการพัฒนาแหล่งข้อมูลออนไลน์ที่สามารถเข้าถึงได้ง่ายและสะดวกในการเก็บข้อมูลต่าง ๆ เช่น ประวัติ ความเป็นมา และรายละเอียดของพระเครื่องที่มีคุณค่า ไม่ควรถูกจำกัดอยู่ในหนังสือหรือแหล่งข้อมูลที่ยากต่อการเข้าถึงเว็บแอปพลิเคชันแหล่งรวมพระเครื่อง (Langpra) โดยใช้ Next.js ได้ถูกพัฒนาขึ้นมาเพื่อตอบโจทย์ความต้องการนี้ โดยมีวัตถุประสงค์หลักในการรวบรวมข้อมูลพระเครื่องจากแหล่งต่าง ๆ มาไว้ในที่เดียว ทำให้ผู้ใช้สามารถค้นหาและศึกษาได้อย่างสะดวกและรวดเร็ว นอกจากนี้ การใช้ Next.js ในการพัฒนายังช่วยให้แอปพลิเคชันมีความเสถียรและมีประสิทธิภาพสูง สามารถรองรับผู้ใช้จำนวนมากได้โดยไม่เกิดปัญหา การพัฒนาเว็บแอปพลิเคชัน Langpra ยังช่วยส่งเสริมการใช้งานเทคโนโลยีในกลุ่มคนรุ่นใหม่ ทำให้พวกเขามีความสนใจในการศึกษาพระเครื่องมากขึ้น การนำเทคโนโลยีมาใช้ในการเก็บรักษาและ

เผยแพร่ข้อมูลเกี่ยวกับพระเครื่องจะช่วยให้วัฒนธรรมและความรู้เกี่ยวกับพระเครื่องสามารถสืบทอดไปสู่คนรุ่นหลังได้อย่างมีประสิทธิภาพ การสร้างเว็บแอปพลิเคชัน Langpra นี้นอกจากจะช่วยแก้ไขปัญหาการเข้าถึงข้อมูลพระเครื่องแล้ว ยังเป็นการสร้างสรรค์เครื่องมือที่สามารถใช้ในการศึกษาและวิจัยเกี่ยวกับพระเครื่องได้ในอนาคต ทำให้ผู้ที่สนใจสามารถเข้าถึงข้อมูลได้อย่างง่ายดายและมีประสิทธิภาพโดยไม่ต้องพึ่งพาแหล่งข้อมูลที่กระจัดกระจายและไม่เป็นระบบ

จากที่มาข้างต้นเป็นเหตุผลให้ทางผู้จัดทำสร้าง “เว็บแอปพลิเคชันแหล่งรวมพระเครื่อง จาก Next.js” เพื่อสร้างและทดสอบประสิทธิภาพเว็บแอปพลิเคชัน ทั้งนี้เพื่อแก้ปัญหาในปัจจุบันที่การหาข้อมูลพระเครื่องนั้นยากลำบากและเข้าถึงได้ยาก อีกทั้งในปัจจุบันคนรุ่นใหม่ให้ความสนใจในการศึกษาทางหนังสือลดลงเป็นอย่างมาก จึงได้ศึกษาถึงการนำเทคโนโลยีมาใช้ในการหาข้อมูลให้กับคนรุ่นใหม่ได้ใช้งานได้ง่ายขึ้น เพื่อช่วยกระตุ้นให้ผู้ใช้มีความสนใจมากขึ้น และตอบสนองในเรื่องของการใช้งานง่าย สะดวก รวดเร็ว โดยมุ่งเน้นไปที่การใช้งานเป็นอันดับแรก

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาการสร้างการพัฒนาเว็บแอปพลิเคชันแหล่งรวมพระเครื่อง

2. แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดการพัฒนาเว็บแอปพลิเคชัน (Web Application Development)

การพัฒนาเว็บแอปพลิเคชันเป็นกระบวนการที่ครอบคลุมการออกแบบ การสร้าง และการบำรุงรักษาโปรแกรมที่ทำงานผ่านเว็บเบราว์เซอร์ การพัฒนาเว็บแอปพลิเคชันมีความสำคัญในยุคปัจจุบัน เนื่องจากการใช้งานอินเทอร์เน็ตและเว็บแอปพลิเคชันได้เข้ามามีบทบาทสำคัญในชีวิตประจำวันของผู้คนมากขึ้น ตั้งแต่การทำงาน การศึกษา การสื่อสาร ไปจนถึงการบันเทิง ความหมายของเว็บแอปพลิเคชัน คือโปรแกรมประยุกต์ที่สามารถทำงานได้บนเว็บเบราว์เซอร์ ผู้ใช้สามารถเข้าถึงและใช้งานได้ผ่านอินเทอร์เน็ตโดยไม่ต้องติดตั้งโปรแกรมบนเครื่องคอมพิวเตอร์ของตนเอง ตัวอย่างของเว็บแอปพลิเคชันที่รู้จักกันดี ได้แก่ Gmail, Google Docs, Facebook, และ Twitter

2.2 การพัฒนาซอฟต์แวร์แบบ Agile (Agile Software Development)

ทฤษฎีการพัฒนาซอฟต์แวร์แบบ Agile (Agile Software Development) เป็นแนวคิดที่ได้รับความนิยมอย่างแพร่หลายในวงการพัฒนาซอฟต์แวร์และเทคโนโลยีสารสนเทศ เนื่องจากมีความยืดหยุ่นและสามารถปรับตัวได้ดีในสภาพแวดล้อมที่เปลี่ยนแปลงอย่างรวดเร็ว Agile มุ่งเน้นการพัฒนาแบบเป็นขั้นตอนที่สามารถตรวจสอบและปรับปรุงได้ตลอดเวลา โดยให้ความสำคัญกับการทำงานร่วมกันระหว่างทีมพัฒนาและลูกค้า เพื่อให้ได้ผลลัพธ์ที่ตรงตามความต้องการและมีคุณภาพสูง

หลักการของ Agile มีหลักการสำคัญ 4 ข้อที่เป็นพื้นฐานในการพัฒนาซอฟต์แวร์ ได้แก่

(1) Individuals and Interactions over Processes and Tools: ให้ความสำคัญกับคนและการสื่อสารระหว่างทีมมากกว่ากระบวนการและเครื่องมือ เพราะทีมงานที่มีความสามารถและการสื่อสารที่ดีจะทำให้การพัฒนาเป็นไปอย่างราบรื่น

(2) Working Software over Comprehensive Documentation: มุ่งเน้นการส่งมอบซอฟต์แวร์ที่ทำงานได้จริงมากกว่าการมีเอกสารที่ครบถ้วน แต่ยังไม่ได้ใช้จริง

(3) Customer Collaboration over Contract Negotiation: การทำงานร่วมกับลูกค้าอย่างใกล้ชิดและต่อเนื่องมีความสำคัญมากกว่าการเจรจาตามสัญญา เพราะความต้องการของลูกค้าอาจเปลี่ยนแปลงได้

(4) Responding to Change over Following a Plan: ความสามารถในการตอบสนองต่อการเปลี่ยนแปลงมีความสำคัญมากกว่าการทำตามแผนที่วางไว้ เนื่องจากสภาพแวดล้อมและความต้องการของผู้ใช้สามารถเปลี่ยนแปลงได้ตลอดเวลา

กระบวนการของ Agile ใช้การพัฒนาแบบเป็นขั้นตอนที่เรียกว่า "สปรินท์" (Sprint) ซึ่งเป็นช่วงเวลาสั้น ๆ ประมาณ 1-4 สัปดาห์ ในแต่ละสปรินท์ ทีมพัฒนาจะทำงานในส่วนที่กำหนดและสามารถตรวจสอบผลลัพธ์ได้ทันที การทำงานในสปรินท์จะเริ่มต้นด้วยการวางแผน (Sprint Planning) ตามด้วยการพัฒนา (Development) การทดสอบ (Testing) และการประชุมตรวจสอบผล (Sprint Review) และสุดท้ายคือการประเมินผลและวางแผนการปรับปรุง (Sprint Retrospective)

2.3 เทคโนโลยีของ Next.js

Next.js เป็นเฟรมเวิร์กที่ถูกพัฒนาขึ้นบนพื้นฐานของ React ซึ่งเป็นไลบรารีสำหรับการสร้างส่วนต่อประสานกับผู้ใช้ (User Interface) ที่ได้รับความนิยมอย่างแพร่หลาย Next.js ถูกออกแบบมาเพื่อให้การพัฒนาเว็บแอปพลิเคชันมีประสิทธิภาพมากขึ้น ด้วยการรวมความสามารถที่หลากหลาย ทั้งการเรนเดอร์ฝั่งเซิร์ฟเวอร์ (Server-side Rendering) และฝั่งไคลเอนต์ (Client-side Rendering) การสร้างเว็บไซต์แบบสถิต (Static Site Generation) และการปรับปรุงประสิทธิภาพในหลายๆด้าน [2] ทั้งนี้ยังมี Gatsby เป็นเฟรมเวิร์กสำหรับการพัฒนาเว็บไซต์แบบ Static Site Generator (SSG) ซึ่งสร้างขึ้นบนพื้นฐานของ React เหมือน Next.js โดยมีการผสมผสานเทคโนโลยีหลาย ๆ ตัวเพื่อสร้างเว็บไซต์ที่มีประสิทธิภาพ การเลือกใช้ Gatsby หรือ Next.js ขึ้นอยู่กับลักษณะและความต้องการของโปรเจกต์ หากเว็บไซต์ของคุณมีเนื้อหาคงที่และต้องการความเร็วในการโหลดสูง Gatsby จะเป็นตัวเลือกที่ดีเกี่ยวกับ static หากคุณต้องการความยืดหยุ่นในการเรนเดอร์และเนื้อหาของเว็บแอปพลิเคชันมีการเปลี่ยนแปลงบ่อย Next.js จะเหมาะสมกว่า ทั้งสองเฟรมเวิร์กนี้มีเครื่องมือและฟีเจอร์ที่ช่วยให้นักพัฒนาสามารถสร้างเว็บแอปพลิเคชันได้อย่างมีประสิทธิภาพ แต่การเลือกใช้งานขึ้นอยู่กับลักษณะเฉพาะของแต่ละโปรเจกต์ ซึ่งในที่นี้ Next.jsจะเหมาะสมกับโปรเจ็กที่กำลังทำมากกว่า

ตารางที่ 2.1 ตารางแสดงการเปรียบเทียบ **Next.js** กับ Framework อื่น ๆ

Framework	ข้อดี	ข้อเสีย	จุดเด่นของ Next.js
Next.js	- รองรับการทำ Server-Side Rendering (SSR) และ Static Site Generation (SSG) ทำให้เว็บโหลดเร็วขึ้นและเป็นมิตรกับ SEO - มีฟีเจอร์พร้อมใช้งานทันที เช่น Routing แบบอัตโนมัติ - สามารถใช้ API routes ในตัว ทำให้ไม่จำเป็นต้องสร้าง backend แยกต่างหาก - รองรับการปรับแต่งการเรนเดอร์ได้อย่างยืดหยุ่น เช่น ISR (Incremental Static Regeneration)	- ต้องทำความคุ้นเคยกับการตั้งค่า SSR และ SSG - ขนาดโปรเจกต์อาจใหญ่ขึ้นเนื่องจากฟีเจอร์ต่าง ๆ	- เป็น fullstack framework ที่รวมทั้ง Frontend และ Backend ในตัวเดียว ทำให้สะดวกในการพัฒนา - เหมาะสำหรับ SEO มากกว่าหลาย ๆ Framework ที่เป็น client-side เท่านั้น - การจัดการ Routing ที่ง่าย โดยไม่ต้องใช้ third-party libraries เช่นใน React
React	- มีชุมชนผู้ใช้งานขนาดใหญ่และมี Libraries ที่เกี่ยวข้องมากมาย - เป็น framework ที่เบาและสามารถขยายได้ง่าย	- ต้องการการตั้งค่าเองสำหรับการทำ SSR หรือ SSG ผ่านเครื่องมืออื่น ๆ เช่น Next.js - ไม่มี routing ในตัว ต้องใช้ libraries เพิ่มเติม	- Next.js เป็น extension ของ React ที่มาพร้อมฟีเจอร์ SSR และ SSG ทำให้ React ที่ใช้ Next.js มีความสามารถมากกว่าในการทำ SEO และการโหลดหน้าเว็บเร็ว
Vue.js	- เรียนรู้และเริ่มต้นง่าย - มีเครื่องมือและ Libraries ในตัวมากมาย เช่น Vue Router - ระบบ two-way data binding ที่สะดวก	- มีชุมชนขนาดเล็กกว่า React - ขาดการสนับสนุน SSR และ SSG ที่แข็งแกร่งในตัว	- Next.js มีการรองรับ SSR/SSG ที่ดีกว่า Vue.js และเป็นมิตรกับ SEO มากกว่า
Angular	- รองรับการทำงานแบบ full-featured (Frontend, State Management, Routing) ในตัว	- ขนาดใหญ่และใช้ทรัพยากรเยอะ - ต้องเรียนรู้เยอะกว่า Framework อื่น	- Next.js มีความยืดหยุ่นมากกว่าและเบากว่า Angular ทำให้พัฒนาเว็บและระบบที่

	- เหมาะสำหรับแอพพลิเคชันขนาดใหญ่		ต้องการประสิทธิภาพสูงและโหลดเร็วได้ดีกว่า

2.1.4 ภาษาของ TypeScript

TypeScript เป็นภาษาการเขียนโปรแกรมที่ถูกพัฒนาขึ้นโดย Microsoft โดยมีเป้าหมายเพื่อเพิ่มความสามารถและความยืดหยุ่นในการพัฒนา JavaScript TypeScript ถูกออกแบบมาเพื่อขยายความสามารถของ JavaScript ด้วยการเพิ่มการตรวจสอบชนิดข้อมูล (Static Typing) และฟีเจอร์ที่ทันสมัยที่ช่วยให้นักพัฒนาสามารถเขียนโค้ดที่มีความชัดเจน อ่านง่าย และลดข้อผิดพลาดในระหว่างการพัฒนา TypeScript มุ่งเน้นการเพิ่มความสามารถและการควบคุมในการเขียนโปรแกรมด้วยการนำแนวคิดสำคัญหลายประการมาใช้ เช่น**การตรวจสอบชนิดข้อมูล (Static Typing)**: TypeScript เพิ่มการตรวจสอบชนิดข้อมูลเข้าไปใน JavaScript ทำให้นักพัฒนาสามารถระบุชนิดของตัวแปร ฟังก์ชัน และคลาสได้อย่างชัดเจน ซึ่งช่วยลดข้อผิดพลาดที่เกิดจากการใช้ตัวแปรผิดชนิดและเพิ่มความน่าเชื่อถือของโค้ด หรือ**การคอมไพล์ (Compilation)**: TypeScript ต้องผ่านกระบวนการคอมไพล์เพื่อแปลงโค้ด TypeScript ให้เป็น JavaScript ก่อนที่จะนำไปใช้งาน กระบวนการนี้ช่วยตรวจสอบข้อผิดพลาดที่เกิดขึ้นในระหว่างการเขียนโค้ดและแจ้งเตือนนักพัฒนาให้แก้ไขก่อนที่จะรันโปรแกรมจริง**และความเข้ากันได้กับ JavaScript (Compatibility with JavaScript)**: TypeScript ถูกออกแบบให้สามารถทำงานร่วมกับโค้ด JavaScript ได้อย่างราบรื่น นักพัฒนาสามารถเพิ่ม TypeScript เข้าไปในโปรเจกต์ JavaScript ที่มีอยู่เดิมได้โดยไม่ต้องเปลี่ยนแปลงโค้ดทั้งหมด รวมถึง**การใช้ฟีเจอร์ของ ECMAScript (ECMAScript Features)**: TypeScript รองรับฟีเจอร์ที่ทันสมัยของ ECMAScript ซึ่งเป็นมาตรฐานของ JavaScript ทำให้นักพัฒนาสามารถใช้ฟีเจอร์ใหม่ ๆ เช่น คลาส โมดูล และฟังก์ชันแอสิงโครนัสได้ทันที

2.5 เทคโนโลยีของ Firebase

Firebase เป็นแพลตฟอร์มการพัฒนาแอปพลิเคชันบนระบบคลาวด์ที่ถูกพัฒนาขึ้นโดย Google โดยมีวัตถุประสงค์เพื่อช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น Firebase มุ่งเน้นการให้บริการในหลายด้าน ตั้งแต่การจัดการฐานข้อมูล การตรวจสอบผู้ใช้ การส่งการแจ้งเตือน และการเก็บรวบรวมข้อมูลการใช้งานแอปพลิเคชัน ทำให้นักพัฒนาสามารถมุ่งเน้นไปที่การสร้างฟังก์ชันหลักของแอปพลิเคชันได้โดยไม่ต้องกังวลเกี่ยวกับการตั้งค่าและการจัดการโครงสร้างพื้นฐาน Firebase มีแนวคิดหลักในการให้บริการเครื่องมือและบริการที่ครอบคลุมทุกด้านของการพัฒนาแอปพลิเคชัน ตั้งแต่การพัฒนา การทดสอบ การเผยแพร่ และการดูแลรักษา โดยแนวคิดหลักของ Firebase ประกอบด้วยการให้บริการแบบบูรณาการFirebase รวมบริการต่าง ๆ ที่จำเป็นในการพัฒนาแอปพลิเคชันเข้าด้วยกันในแพลตฟอร์มเดียว ทำให้นักพัฒนาสามารถใช้บริการต่าง ๆ ได้อย่างสะดวกและรวดเร็ว โดยไม่ต้องตั้งค่าและจัดการโครงสร้างพื้นฐานเอง และต่อมาการทำงานแบบเรียลไทม์ Firebase มีบริการฐานข้อมูลแบบเรียลไทม์ที่ช่วยให้นักพัฒนาสามารถจัดเก็บและดึงข้อมูลได้อย่างรวดเร็วและทันที ทำให้การพัฒนาแอปพลิเคชันที่ต้องการ

การตอบสนองแบบเรียลไทม์เป็นไปอย่างราบรื่น การรองรับหลายแพลตฟอร์ม: Firebase รองรับการพัฒนาแอปพลิเคชันทั้งบนระบบปฏิบัติการ Android, iOS และเว็บ ทำให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่สามารถใช้งานได้บนหลายแพลตฟอร์มโดยใช้โค้ดเบสเดียวกัน

ตารางที่ 2.2 ตารางเปรียบเทียบฐานข้อมูล Firebase กับฐานข้อมูลยอดนิยม

Database	ข้อดี	ข้อเสีย	จุดเด่นของ Firebase
Firebase Firestore	- Real-time Database: รองรับการซิงโครไนซ์ข้อมูลแบบเรียลไทม์ ทำให้เหมาะสำหรับแอพพลิเคชันที่ต้องการอัปเดตข้อมูลทันที - Backend-as-a-Service (BaaS): ไม่ต้องดูแล	- Query Limited: Query ที่ใช้ใน Firestore อาจไม่ยืดหยุ่นเท่า SQL databases - Cost: ค่าบริการแบบ Pay-as-you-go อาจเพิ่มขึ้นตามปริมาณการใช้งานที่สูงขึ้น	- Firebase ให้บริการแบบครบวงจรทั้งฐานข้อมูลแบบ NoSQL, Real-time, และ Authentication ทำให้การพัฒนาแอพเป็นไปอย่างรวดเร็ว - สามารถรองรับผู้ใช้จำนวนมากได้ทันทีโดยไม่ต้องตั้งค่าเซิร์ฟเวอร์หรือสเกลเอง - มี Real-time database ซึ่ง

Database	ข้อดี	ข้อเสีย	จุดเด่นของ Firebase
Firebase Firestore	- Real-time Database: รองรับการซิงโครไนซ์ข้อมูลแบบเรียลไทม์ ทำให้เหมาะสำหรับแอพพลิเคชันที่ต้องการอัปเดตข้อมูลทันที - Backend-as-a-Service (BaaS): ไม่ต้องดูแลเซิร์ฟเวอร์เอง ให้บริการ Backend สำหรับการจัดเก็บข้อมูล, Authentication และ Hosting - Scalability: รองรับการขยายระบบอย่างอัตโนมัติ ไม่	- Query Limited: Query ที่ใช้ใน Firestore อาจไม่ยืดหยุ่นเท่า SQL databases - Cost: ค่าบริการแบบ Pay-as-you-go อาจเพิ่มขึ้นตามปริมาณการใช้งานที่สูงขึ้น	- Firebase ให้บริการแบบครบวงจรทั้งฐานข้อมูลแบบ NoSQL, Real-time, และ Authentication ทำให้การพัฒนาแอพเป็นไปอย่างรวดเร็ว - สามารถรองรับผู้ใช้จำนวนมากได้ทันทีโดยไม่ต้องตั้งค่าเซิร์ฟเวอร์หรือสเกลเอง - มี Real-time database ซึ่งเป็นจุดแข็งเมื่อเทียบกับฐานข้อมูลอื่นๆ ที่มักต้องเพิ่มฟังก์ชันการซิงค์แบบเรียลไทม์เพิ่มเติม

	ต้องจัดการการเพิ่มขนาดระบบด้วยตนเอง - Cross-platform Support: รองรับการเชื่อมต่อหลายแพลตฟอร์ม เช่น Android, iOS และเว็บ ได้โดยตรง		
MySQL	- Structured Data: เหมาะสำหรับข้อมูลที่มีโครงสร้างชัดเจน - Transaction Management: รองรับการทำธุรกรรมที่ต้องการความถูกต้องของข้อมูลอย่างสูง เช่น ระบบการเงิน - SQL Queries: มี Query ที่ยืดหยุ่นและทรงพลัง	- Scaling Limitations: การสเกลระบบใน MySQL อาจซับซ้อนกว่าเมื่อเทียบกับฐานข้อมูลแบบ NoSQL เช่น Firebase - ไม่มีการซิงค์ข้อมูลแบบเรียลไทม์ในตัว	- Firebase เหมาะกับการพัฒนาแอพพลิเคชันที่ไม่ต้องการโครงสร้างข้อมูลที่ซับซ้อน แต่เน้นการซิงโครไนซ์ข้อมูลแบบเรียลไทม์และความง่ายในการใช้งาน

Database	**ข้อดี**	**ข้อเสีย**	**จุดเด่นของ Firebase**
PostgreSQL	- Advanced SQL Features: รองรับการทำงานเชิงซับซ้อนและรองรับ JSON, XML ได้ในตัว - ACID Compliance: เหมาะกับการจัดการข้อมูลที่ต้องการความน่าเชื่อถือ เช่น ธุรกรรมทางการเงิน - Extensibility: สามารถขยายและปรับแต่งตามความต้องการได้ง่าย	- Setup and Maintenance: ต้องมีการดูแลเซิร์ฟเวอร์และการติดตั้งให้เหมาะสมกับการใช้งาน - ไม่รองรับการซิงค์ข้อมูลแบบเรียลไทม์ในตัว	- Firebase ไม่ต้องการการดูแลเซิร์ฟเวอร์ และมีฟังก์ชันซิงค์ข้อมูลแบบเรียลไทม์ที่เหมาะสำหรับแอพพลิเคชันที่ต้องการการอัปเดตข้อมูลอย่างต่อเนื่อง
MongoDB	- Schema-less: รองรับโครงสร้างข้อมูลที่ยืดหยุ่น ทำ	- Lack of ACID Compliance: ไม่	- Firebase ให้บริการด้านการซิงค์ข้อมูลแบบเรียลไทม์ในตัว

	ให้เหมาะกับข้อมูลที่ไม่มีโครงสร้างชัดเจน - Scaling: รองรับการสเกลระบบได้ง่าย เนื่องจากเป็น NoSQL - High Write Throughput: รองรับปริมาณการเขียนข้อมูลที่สูง	รองรับการทำธุรกรรมที่มีความซับซ้อนสูง (เวอร์ชันใหม่มีการรองรับบางส่วน แต่ไม่สมบูรณ์แบบ) - ไม่รองรับการซิงค์ข้อมูลแบบเรียลไทม์โดยตรง	ทำให้การพัฒนาแอพพลิเคชันที่ต้องการการอัปเดตแบบ Real-time เป็นไปได้ง่ายและเร็วกว่า MongoDB

2.6 องค์ความรู้เกี่ยวกับพระเครื่อง

พระเครื่องเป็นวัตถุที่มีความสำคัญทางศาสนาและวัฒนธรรมในประเทศไทย มีความเชื่อว่าพระเครื่องสามารถนำพลังงานบวกมาให้ผู้ครอบครอง ช่วยป้องกันภัยอันตราย และนำความโชคดีมาให้ การศึกษาและทำความเข้าใจเกี่ยวกับพระเครื่องนั้นมีหลายแง่มุม ซึ่งสามารถแบ่งออกเป็นหลายด้าน ในด้านประวัติศาสตร์และความเป็นมา พระเครื่องมีต้นกำเนิดมาตั้งแต่สมัยโบราณ เริ่มแรกถูกสร้างขึ้นมาเพื่อใช้ในพิธีกรรมทางศาสนา และเป็นเครื่องหมายของความศรัทธา พัฒนาการของพระเครื่องได้รับอิทธิพลจากวัฒนธรรมและศาสนาต่าง ๆ ที่เข้ามาในประเทศไทย พระเครื่องได้รับการพัฒนาและเปลี่ยนแปลงรูปลักษณ์และรูปแบบไปตามยุคสมัยและวัฒนธรรมต่าง ๆ มีการสร้างพระเครื่องในหลายรูปแบบ เช่น พระบูชา พระผง พระกริ่ง และอื่น ๆ วัตถุประสงค์และการใช้งาน พระเครื่องถูกใช้ในการบูชาเพื่อสักการะและแสดงถึงความศรัทธาในพระพุทธศาสนา มีความเชื่อว่าพระเครื่องสามารถป้องกันภัยอันตรายและเสริมดวงให้กับผู้ครอบครอง และยังถูกใช้เป็นเครื่องรางของขลังเพื่อเสริมสร้างพลังงานบวกและความมั่นใจให้กับผู้สวมใส่ การสร้างพระเครื่องใช้วัสดุต่าง ๆ เช่น ดินเผา โลหะ ผงพระพุทธคุณ และอื่น ๆ วัสดุที่ใช้มีผลต่อคุณภาพและความศักดิ์สิทธิ์ของพระเครื่อง กระบวนการสร้างพระเครื่องมักเกี่ยวข้องกับการทำพิธีกรรมศักดิ์สิทธิ์และการอธิษฐานให้พระเครื่องมีพลังงานบวก

(1) ประเภทของพระเครื่อง

(1.1) พระบูชา: พระพุทธรูปที่ใช้สำหรับการบูชาที่บ้านหรือสถานที่ศักดิ์สิทธิ์

(1.2) พระกริ่ง: พระพุทธรูปที่มีการใส่เม็ดโลหะภายในเพื่อให้เกิดเสียงกริ่งเมื่อเขย่า

(1.3) พระผง: พระเครื่องที่ทำจากผงศักดิ์สิทธิ์ มักมีขนาดเล็กและพกพาง่าย

(1.4) พระปิดตา: พระพุทธรูปที่มีลักษณะนั่งคู้เข่าและเอามือปิดตา เชื่อกันว่ามีพลังในการป้องกันอันตราย

2.7 งานวิจัยที่เกี่ยวข้อง

2.7.1 การพัฒนาแอปพลิเคชันแหล่งท่องเที่ยวในจังหวัดจันทบุรี

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ ออกแบบ พัฒนา และประเมินประสิทธิภาพแอปพลิเคชันแหล่งท่องเที่ยวในจังหวัดจันทบุรี ประชากรที่ใช้ในการวิจัยครั้งนี้ ได้แก่ ผู้เชี่ยวชาญด้านเทคโนโลยีสารสนเทศที่มีส่วนในการทดสอบ และใช้งานแอปพลิเคชันที่พัฒนาขึ้น จำนวน 15 คน การเลือกกลุ่มตัวอย่างใช้แบบเจาะจง เครื่องมือที่ใช้ ได้แก่ แอปพลิเคชันแหล่งท่องเที่ยวในจังหวัดจันทบุรี และแบบประเมินประสิทธิภาพ ซึ่งการพัฒนาแอปพลิเคชันใช้ React Native เป็นเฟรมเวิร์ก และ Golang พัฒนา API สำหรับใช้เป็นส่วนต่อประสานงานการจัดการข้อมูลในแอปพลิเคชัน กับฐานข้อมูล Mysql วิเคราะห์ข้อมูลโดยการแจกแจงความถี่ ร้อยละ ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน ผลการวิจัยพบว่า แอปพลิเคชันที่พัฒนาขึ้นสามารถแสดงผลการทำงานได้ทั้งระบบ Android และ iOS และการประเมินประสิทธิภาพของแอปพลิเคชันใน 5 ด้านอยู่ในระดับดี ($\bar{x}$ = 4.09) โดยเรียงลำดับค่าเฉลี่ยจากมากไปหาน้อย ดังนี้ ด้านความยืดหยุ่น ง่ายในการใช้งาน ($\bar{x}$ = 4.36) ด้านความสามารถ ($\bar{x}$ = 4.25) ด้านความถูกต้อง ($\bar{x}$ = 4.08) ด้านประสิทธิภาพ ($\bar{x}$ = 4.00) และ ด้านการรักษาความปลอดภัย ($\bar{x}$ = 3.78) [5]

2.7.2 การพัฒนาเว็บไซต์ นำเสนอวัดประดู่ลาย จ.กำแพงเพชร ด้วยภาพ 3 มิติ

การวิจัยนี้พัฒนาเว็บไซต์โดยใช้ภาษา PHP ร่วมกับ MySQL มีวัตถุประสงค์เพื่อ 1) พัฒนาเว็บไซต์ประชาสัมพันธ์และนำเสนอวัดประดู่ลาย จ.กำแพงเพชร ใน รูปแบบภาพและวิดีโอ 3 มิติในส่วนของอาคารและสิ่งก่อสร้างภายในวัด และ 2) ประเมินผลความพึงพอใจที่มี ต่อเว็บไซต์เว็บไซต์นี้ได้พัฒนาตามหลักการของวงจรการพัฒนาระบบ ที่มีการออกแบบแผนภาพบริบทของ ระบบ แผนผังความสัมพันธ์ของข้อมูล ฐานข้อมูล และส่วนติดต่อผู้ใช้งาน จากผลการประเมิน พบว่า ผู้ใช้งานมี ความพึงพอใจในด้านรูปแบบที่นำเสนอมีความน่าสนใจมากที่สุด (ค่าเฉลี่ย 4.75) และด้านสามารถใช้ในการ ประชาสัมพันธ์วัดประดู่ลาย (ค่าเฉลี่ย 4.70) แต่มีความพึงพอใจในด้านเสียงพากย์และดนตรีประกอบมีความ เหมาะสม ได้รับการประเมินน้อยที่สุด (ค่าเฉลี่ย 4.35) โดยภาพรวมมีระดับความพึงพอใจมากที่สุด (ค่าเฉลี่ย 4.50) [6]

2.7.3 Next.js Framework for Modern Web Applications" โดย John Doe, 2020

การวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาและประเมินความสามารถของเฟรมเวิร์ก Next.js ในการพัฒนาเว็บแอปพลิเคชันสมัยใหม่ โดยเน้นที่ประสิทธิภาพในการเรนเดอร์ การจัดการเส้นทาง การดึงข้อมูล และการเพิ่มประสิทธิภาพเว็บไซต์ งานวิจัยนี้รวมถึงการสร้างแอปพลิเคชันตัวอย่างเพื่อทดสอบฟีเจอร์หลักของ Next.js และการเปรียบเทียบกับเฟรมเวิร์กอื่น ๆ เช่น Gatsby และ Create React App ผลการวิจัยแสดงให้เห็นว่า Next.js มีประสิทธิภาพสูงในการพัฒนาเว็บแอปพลิเคชันที่มีความยืดหยุ่นและสามารถปรับตัวได้ตามความต้องการของผู้ใช้ ซึ่งได้บทสรุปว่าNext.js เป็นเฟรมเวิร์กที่มีประสิทธิภาพสูงและมีความยืดหยุ่นในการพัฒนาเว็บแอปพลิเคชันสมัยใหม่ [8] การวิจัยนี้แสดงให้เห็นถึงความสามารถของ Next.js ในการพัฒนาแอปพลิเคชันที่มีประสิทธิภาพและสามารถปรับตัวได้ตามความต้องการของผู้ใช้ ข้อเสนอแนะสำหรับงานวิจัยใน

อนาคตคือการศึกษาและพัฒนาฟีเจอร์เพิ่มเติมของ Next.js เพื่อเพิ่มประสิทธิภาพและความสามารถในการใช้งานของเว็บแอปพลิเคชัน

2.2.4 Web Performance Testing: Methodologies, Tools and Challenges โดย Hegde and S. PM [7]

การวิจัยนี้มุ่งเน้นไปที่การทดสอบประสิทธิภาพของเว็บแอปพลิเคชัน โดยวิเคราะห์และประเมินเครื่องมือและเทคนิคต่าง ๆ ที่ใช้ในการทดสอบประสิทธิภาพ การศึกษาครั้งนี้รวมถึงการสร้างแอปพลิเคชันตัวอย่างและการใช้เครื่องมือทดสอบต่าง ๆ เช่น JMeter, LoadRunner และ Lighthouse ผลการวิจัยแสดงให้เห็นถึงความสำคัญของการทดสอบประสิทธิภาพและเสนอแนะแนวทางปฏิบัติที่ดีที่สุดสำหรับนักพัฒนาในการปรับปรุงประสิทธิภาพของเว็บแอปพลิเคชัน เพื่อให้มั่นใจว่าแอปพลิเคชันสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองต่อความต้องการของผู้ใช้ได้อย่างดี

3. ระเบียบวิธีวิจัย

การพัฒนาเว็บแอพพลิเคชันแหล่งรวมพระเครื่อง จาก Next.js จะแบ่งขั้นตอนการทำงานออกเป็นส่วน ๆ แล้วทำการเรียกใช้ เพื่อแก้ปัญหาในปัจจุบันที่การหาข้อมูลพระเครื่องนั้นยากลำบากและเข้าถึงได้ยาก อีกทั้งในปัจจุบันคนรุ่นใหม่ให้ความสนใจในการศึกษาทางหนังสือลดลงเป็นอย่างมาก จึงได้ศึกษาถึงการนำเทคโนโลยีมาใช้ในการหาข้อมูลให้กับคนรุ่นใหม่ได้ใช้งานได้ง่ายขึ้น เพื่อช่วยกระตุ้นให้ผู้ใช้มีความสนใจมากขึ้นและตอบสนองในเรื่องของการใช้งานง่าย สะดวก รวดเร็ว โดยมุ่งเน้นไปที่การใช้งานเป็นอันดับแรก

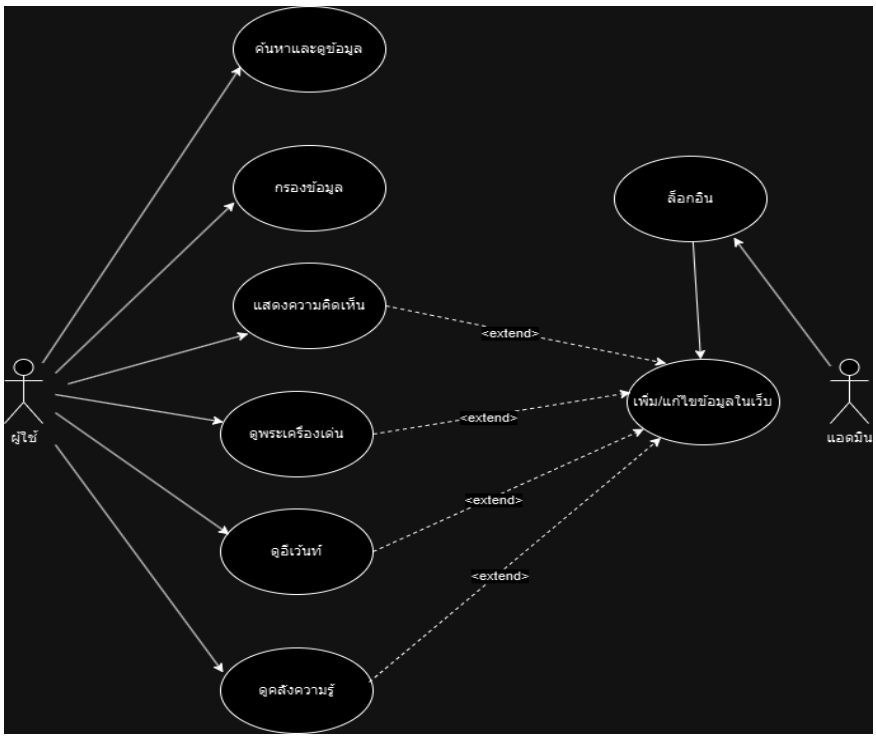
3.1 การวิเคราะห์ปัญหา และศึกษาค้นคว้าข้อมูล

3.1.1 ศึกษาปัญหา และความต้องการของผู้ใช้งานเป้าหมาย เช่น ผู้ที่สนใจในพระเครื่อง นักสะสมพระเครื่อง หรือผู้ซื้อขายพระเครื่องออนไลน์ เพื่อมานำออกแบบฟังก์ชันและวิเคราะห์ปัญหาเพื่อให้ตรงความต้องการของผู้ใช้งาน

3.1.2 ศึกษาเครื่องมือในการพัฒนาและข้อจำกัดในด้านต่าง ๆ ที่มีผลกระทบต่อการพัฒนาระบบ เพื่อนำแนวคิดในการศึกษาครั้งนี้ไปทำการต่อยอดในการพัฒนาแอปพลิเคชันแหล่งรวมพระเครื่อง รวมไปถึงทฤษฎีต่าง ๆ ที่สามารถนำมาใช้ในการพัฒนา ศึกษาเว็บแอพพลิเคชันโดยเรียนรู้ JavaScript, TypeScript, React, Next.js, SSR, SSG, Dynamic Routing, API Routes รวมถึงการใช้งาน Firebase เช่น Cloud Firestore, Realtime Database, Cloud Functions, Hosting พร้อมทั้งปรับปรุงประสิทธิภาพและการใช้งาน Google Form.

3.2 การวิเคราะห์ และการออกแบบระบบ

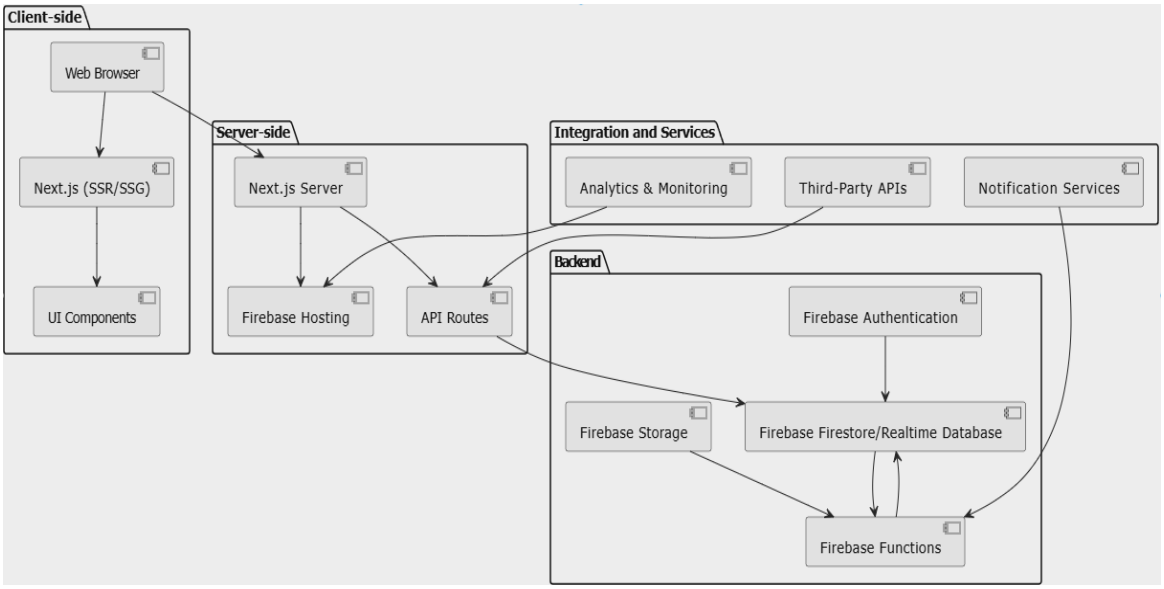
3.2.1 Use Case diagram ซึ่งจะเห็นการทำงานได้จาก UML Diagram ในการแสดงถึงโครงสร้าง และความสัมพันธ์ของข้อมูล ดังนี้



ภาพที่ 3.1 Use Case diagram

3.2.2 สถาปัตยกรรมของระบบ (System Architecture)

การสร้างสถาปัตยกรรมระบบ สำหรับเว็บแอพพลิเคชัน Langpra ที่ใช้ Next.js และ Firebase สามารถแบ่งออกเป็นหลายส่วน ซึ่งจะประกอบไปด้วยส่วนต่าง ๆ ของระบบที่ทำงานร่วมกันอย่างไร้รอยต่อ โดยจะอธิบายเป็นภาพดังนี้



ภาพที่ 3.2 System Architecture

3.2.3 Use Case Description

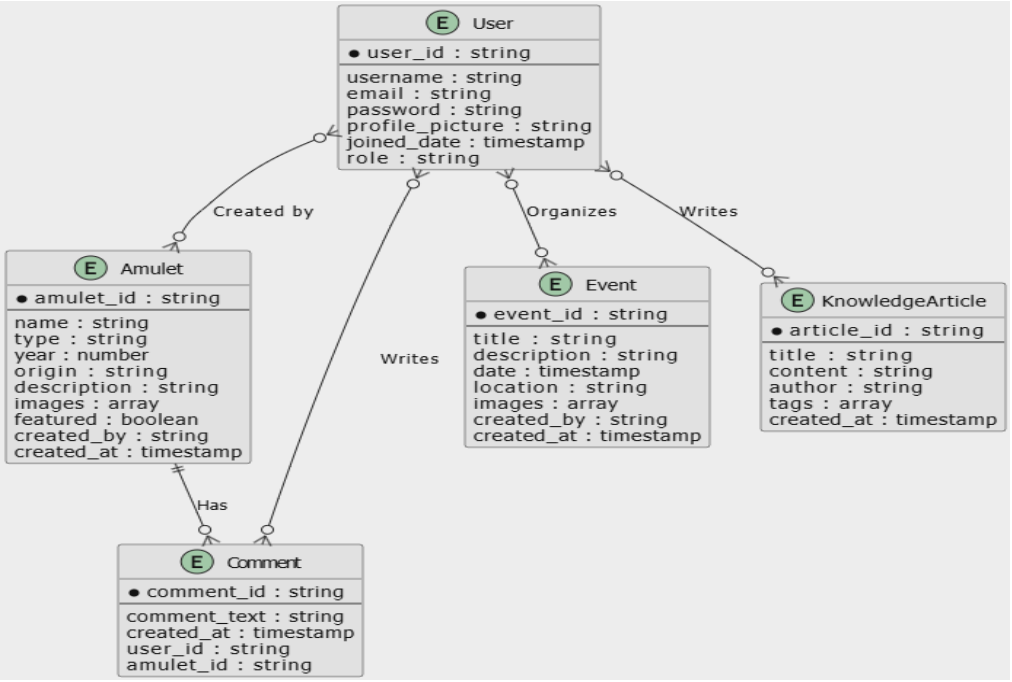
Use Case Description จะอธิบายรายละเอียดการทำงานของแต่ละ Use Case ในระบบได้ชัดเจนขึ้น สำหรับเว็บแอพพลิเคชัน Langpra ที่กำลังพัฒนา โดยสามารถอธิบายรายละเอียดของแต่ละ Use Case ได้ดังนี้:

Use Case: ค้นหาและการเรียกดูข้อมูล (Search and View Information)

1. Use Case Name: ค้นหาและการเรียกดูข้อมูล
2. Actor: ผู้ใช้
3. Description: ผู้ใช้สามารถค้นหาและเรียกดูข้อมูลเกี่ยวกับพระเครื่องที่ต้องการได้ โดยการค้นหาจะทำผ่านกล่องค้นหาหรือระบบกรองข้อมูล เพื่อให้สามารถเข้าถึงข้อมูลที่ต้องการได้อย่างรวดเร็วและสะดวก
4. Precondition: ผู้ใช้เข้าสู่ระบบเรียบร้อยแล้ว (หากมีระบบล็อกอิน)
5. Basic Flow:
   (1) ผู้ใช้ป้อนคำค้นหาในกล่องค้นหา หรือเลือกตัวกรองข้อมูล
   (2) ระบบจะแสดงผลลัพธ์ที่ตรงกับคำค้นหาหรือตัวกรองข้อมูลที่เลือก
   (3) ผู้ใช้เลือกพระเครื่องจากผลลัพธ์เพื่อดูรายละเอียด
   (4) ระบบแสดงข้อมูลรายละเอียดของพระเครื่องที่เลือก
6. Alternative Flow**:**
   (1) หากไม่มีข้อมูลที่ตรงกับคำค้นหา ระบบจะแจ้งผู้ใช้และแนะนำให้ลองค้นหาด้วยคำอื่น
7. Postcondition**:** ผู้ใช้ได้รับข้อมูลพระเครื่องที่ต้องการดูเรียบร้อยแล้ว
8. Exceptions:
   (1) ระบบไม่สามารถแสดงผลลัพธ์ได้เนื่องจากปัญหาทางเทคนิค หรือการเชื่อมต่ออินเทอร์เน็ตไม่เสถียร

3.2.4 ออกแบบฐานข้อมูล (ER Diagram)

การออกแบบฐานข้อมูลสำหรับเว็บแอพพลิเคชัน Langpra โดยใช้ Firebase จำเป็นต้องพิจารณาโครงสร้างข้อมูลที่สามารถรองรับการใช้งานที่หลากหลาย เช่น การจัดเก็บข้อมูลพระเครื่อง ความคิดเห็น อีเว้นท์ และคลังความรู้ ต่อไปนี้เป็นการออกแบบฐานข้อมูลในระดับโครงสร้าง (Schema Design) สำหรับ Firebase Firestore:



ภาพที่ 3.3 ER Diagram

3.2.5 การออกแบบระบบ (System Design Documentation)

(1) วัตถุประสงค์ของเอกสาร

เอกสารฉบับนี้จัดทำขึ้นเพื่ออธิบายรายละเอียดของการออกแบบระบบแอพพลิเคชัน Langpra ซึ่งเป็นแอพพลิเคชันสำหรับการรวบรวมและแบ่งปันข้อมูลเกี่ยวกับพระเครื่อง เอกสารนี้จะช่วยให้ทีมพัฒนาเข้าใจการออกแบบในเชิงสถาปัตยกรรม ข้อมูล และการทำงานร่วมกันของระบบ

(2) ขอบเขตของระบบ

ระบบนี้ประกอบด้วยฟังก์ชันการค้นหาข้อมูลพระเครื่อง การแสดงความคิดเห็น การแสดงข้อมูลเกี่ยวกับอีเว้นท์ และคลังความรู้เกี่ยวกับพระเครื่อง ผู้ใช้สามารถเข้าสู่ระบบเพื่อเข้าถึงข้อมูลและใช้ฟีเจอร์ต่าง ๆ และผู้ดูแลระบบสามารถจัดการเนื้อหาภายในระบบได้ข

(3) สถาปัตยกรรมระบบ (System Architecture)

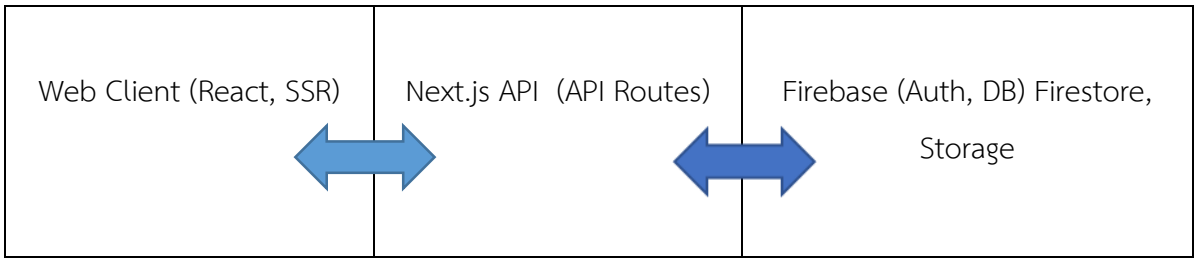
ภาพรวมสถาปัตยกรรม ระบบนี้ใช้เทคโนโลยีหลักคือ Next.js ในการจัดการส่วน Frontend และ Firebase ในการจัดการ Backend รวมถึงการจัดเก็บข้อมูล ระบบจะแบ่งการทำงานออกเป็นสองส่วนใหญ่คือ:

(1) Client-side (Frontend): พัฒนาโดยใช้ Next.js ที่ช่วยให้สามารถทำงานแบบ Server-Side Rendering (SSR) และ Static Site Generation (SSG) ได้

(2) Server-side (Backend): ใช้ Firebase เพื่อจัดการ Authentication, Firestore สำหรับจัดเก็บข้อมูล และ Firebase Functions สำหรับจัดการการทำงานเชิงตรรกะที่ต้องการบนฝั่งเซิร์ฟเวอร์

ตารางภาพด้านล่างนี้แสดงถึงการเชื่อมต่อระหว่าง Client-side, API, และ Backend (Firebase) ที่ใช้ในระบบ

ตารางที่ 3.1 ตารางแสดงถึงการเชื่อมต่อระหว่าง Client-side, API, และ Backend (Firebase) ที่ใช้ในระบบ



(4) การออกแบบระบบผู้ใช้ (User Interface Design)

(1) หน้าจอการค้นหาพระเครื่อง (Amulet Search Screen)

(1.1) หน้าจอหลัก: ผู้ใช้สามารถค้นหาข้อมูลพระเครื่องได้ผ่านทางช่องค้นหา

(1.2) ส่วนประกอบ:

(1.2.1) ช่องค้นหา: ผู้ใช้สามารถป้อนข้อมูลที่ต้องการค้นหา เช่น ชื่อพระเครื่อง

(1.2.2) ผลการค้นหา: ระบบจะแสดงผลลัพธ์ตามข้อมูลที่ค้นหา

(2) หน้าจอแสดงความคิดเห็น (Comment Screen)

(2.1) หน้าจอแสดงรายละเอียดพระเครื่อง: ผู้ใช้สามารถดูข้อมูลพระเครื่องและแสดงความคิดเห็นได้

(2.2) ส่วนประกอบ:

(2.2.1) ข้อมูลพระเครื่อง: แสดงข้อมูลต่าง ๆ เช่น ชื่อ, ปีที่สร้าง, ประเภท

(2.2.1) ฟอร์มแสดงความคิดเห็น: ผู้ใช้สามารถกรอกความคิดเห็นเกี่ยวกับพระเครื่อง

(5) การออกแบบระบบ Backend (Backend Design)

(1) Firebase Authentication

(1.1) ระบบใช้ Firebase Authentication ในการจัดการการเข้าสู่ระบบของผู้ใช้ รวมถึงการรองรับการลงทะเบียนผ่าน Google หรืออีเมล

(1.2) ระบบนี้จัดการการยืนยันตัวตนแบบ OAuth ซึ่งทำให้การจัดการผู้ใช้มีความปลอดภัยและง่ายต่อการขยายระบบ

(2) Firebase Firestore

(2.1) การจัดเก็บข้อมูลพระเครื่อง (Amulets Collection): ข้อมูลพระเครื่องจะถูกจัดเก็บในรูปแบบเอกสาร (Documents) ภายใน Collection "Amulets"

(2.2) การจัดเก็บความคิดเห็น (Comments Sub-Collection): ความคิดเห็นจะถูกจัดเก็บใน Sub-Collection ของแต่ละเอกสารพระเครื่อง ซึ่งทำให้การจัดการข้อมูลเชื่อมโยงได้ง่ายและรวดเร็ว

(3) Firebase Functions

(3.1) Firebase Functions จะใช้สำหรับการประมวลผลเชิงตรรกะที่ต้องการการทำงานในฝั่งเซิร์ฟเวอร์ เช่น การแจ้งเตือนผู้ใช้เมื่อมีการอัปเดตข้อมูลพระเครื่อง หรือการตรวจสอบความปลอดภัยในการทำงานบางส่วน

(6) การทดสอบระบบ (System Testing)

(1) การทดสอบหน่วย (Unit Testing)

(1.1) ระบบจะทำการทดสอบการทำงานของฟังก์ชันแต่ละส่วนแยกกัน เช่น การค้นหา การแสดงความคิดเห็น เพื่อให้มั่นใจว่าแต่ละฟังก์ชันทำงานได้อย่างถูกต้อง

(1.2) การทดสอบรวม (Integration Testing)

(1) ทดสอบการทำงานร่วมกันระหว่างส่วนต่าง ๆ ของระบบ เช่น การเชื่อมต่อระหว่าง API ของ Next.js และ Firebase Firestore เพื่อให้แน่ใจว่าข้อมูลถูกดึงและแสดงผลอย่างถูกต้อง

(7) สรุปผลการออกแบบ (Conclusion)

เอกสารการออกแบบนี้ได้อธิบายรายละเอียดเกี่ยวกับสถาปัตยกรรมของระบบ โครงสร้างฐานข้อมูล การออกแบบส่วนติดต่อผู้ใช้ และการจัดการการทำงานในฝั่ง Backend ซึ่งจะเป็นแนวทางในการพัฒนาและบำรุงรักษาระบบแอพพลิเคชัน Langpra ให้สามารถทำงานได้อย่างมีประสิทธิภาพและรองรับการขยายตัวในอนาคต

3.3 การพัฒนาระบบ

การพัฒนาระบบสำหรับเว็บแอพพลิเคชัน Langpra ที่ใช้ Next.js และ Firebase สามารถแบ่งออกเป็นหลายขั้นตอนหลัก ๆ ซึ่งครอบคลุมกระบวนการตั้งแต่การวางแผน การออกแบบ ไปจนถึงการทดสอบและเปิดใช้งานระบบ ต่อไปนี้คือขั้นตอนและแนวทางที่สามารถใช้ในการพัฒนาระบบ:

3.3.1 การวางแผนระบบ (Planning)

กำหนดวัตถุประสงค์และเป้าหมาย:ระบุวัตถุประสงค์หลักของระบบ เช่น การให้ข้อมูลที่ถูกต้องและรวดเร็วเกี่ยวกับพระเครื่อง การสร้างชุมชนออนไลน์สำหรับนักสะสมพระเครื่อง การอำนวยความสะดวกในการจัดการและแสดงผลข้อมูลที่เกี่ยวข้อง

การวิเคราะห์ความต้องการ (Requirements Analysis):รวบรวมความต้องการจากผู้ใช้และผู้มีส่วนได้ส่วนเสียอื่น ๆ เช่น ฟังก์ชันการค้นหา ฟังก์ชันการกรองข้อมูล การแสดงความคิดเห็น เป็นต้น จัดทำเอกสารความต้องการ (Requirements Document) เพื่อสรุปและยืนยันข้อกำหนดก่อนการพัฒนา

3.3.2 การออกแบบระบบ (System Design)

การออกแบบสถาปัตยกรรมระบบ (System Architecture Design):วางโครงสร้างสถาปัตยกรรมของระบบ เช่น การใช้ Next.js สำหรับ Frontend และการจัดการเซิร์ฟเวอร์, การใช้ Firebase สำหรับ Backend ในการจัดเก็บข้อมูลและการจัดการ Authentication

การออกแบบฐานข้อมูล (Database Design):ออกแบบ ERD เพื่อกำหนดโครงสร้างและความสัมพันธ์ของข้อมูลใน Firebase Firestore รวมถึงการจัดทำเอกสารที่เกี่ยวข้อง

การออกแบบ UI/UX (User Interface and User Experience Design):ออกแบบหน้าตาของแอพพลิเคชันให้ใช้งานง่ายและมีความสวยงาม รวมถึงการทำ Wireframe และ Prototype เพื่อทดสอบก่อนการพัฒนาจริง

3.3.3 การพัฒนาระบบ (Development)

การตั้งค่าโปรเจค (Project Setup):ตั้งค่าโปรเจค Next.js รวมถึงการเชื่อมต่อกับ Firebase โดยการตั้งค่า Firebase SDK และบริการต่าง ๆ เช่น Authentication, Firestore, Hosting การพัฒนา Frontend:พัฒนาส่วนของผู้ใช้ เช่น หน้าแรก หน้าแสดงรายละเอียดพระเครื่อง หน้าค้นหา หน้าคลังความรู้ โดยใช้ React Components และ Next.js Routing

การพัฒนา Backend:พัฒนา API Routes ใน Next.js เพื่อเชื่อมต่อกับ Firebase และจัดการคำขอที่มาจาก Frontend เช่น การดึงข้อมูลจาก Firestore การอัปเดตข้อมูล

การผสานรวม Firebase Services:ผสานรวม Firebase Authentication สำหรับการจัดการผู้ใช้และการยืนยันตัวตน ใช้ Firebase Firestore สำหรับการจัดเก็บข้อมูลพระเครื่อง อีเว้นท์ คลังความรู้ และความคิดเห็น ใช้ Firebase Storage สำหรับจัดเก็บไฟล์มีเดีย เช่น รูปภาพของพระเครื่อง

3.3.4 การทดสอบระบบ (Testing)

การทดสอบหน่วย (Unit Testing):ทดสอบการทำงานของแต่ละ Component และแต่ละฟังก์ชันการทำงานของระบบแยกกัน เพื่อให้แน่ใจว่าไม่มีข้อผิดพลาด

การทดสอบรวม (Integration Testing):ทดสอบการทำงานร่วมกันของทุกส่วน เช่น การค้นหาข้อมูลและแสดงผลในหน้าผู้ใช้ การเชื่อมต่อกับฐานข้อมูล และการจัดการคำขอจาก API

การทดสอบประสิทธิภาพ (Performance Testing):ทดสอบการโหลดหน้าเว็บ การดึงข้อมูลจาก Firebase และการจัดการผู้ใช้พร้อมกันหลายคน เพื่อประเมินประสิทธิภาพและปรับปรุงความเร็ว

การทดสอบการใช้งาน (User Acceptance Testing - UAT):ให้กลุ่มผู้ใช้ทดลองใช้งานระบบเพื่อเก็บข้อมูลเกี่ยวกับความพึงพอใจ ปัญหาที่พบ และข้อเสนอแนะสำหรับการปรับปรุง

3.3.5 การปรับปรุงและการเปิดใช้งาน (Deployment and Maintenance)

การปรับปรุงตามผลการทดสอบ:ปรับปรุงและแก้ไขข้อผิดพลาดหรือปัญหาที่พบในขั้นตอนการทดสอบ รวมถึงการปรับปรุง UI/UX ตามคำแนะนำของผู้ใช้

การเปิดใช้งานระบบ (Deployment):Deploy ระบบไปยัง Firebase Hosting และตั้งค่าคอนฟิกที่จำเป็น เช่น การตั้งค่า SSL, การตั้งค่า Domain Name

การบำรุงรักษาและอัปเดต (Maintenance and Updates):จัดการบำรุงรักษาระบบอย่างต่อเนื่อง เพื่อให้ระบบทำงานได้อย่างราบรื่น รวมถึงการอัปเดตฟีเจอร์ใหม่ๆ ตามความต้องการของผู้ใช้

3.3.6 การวิเคราะห์หลังการเปิดใช้งาน (Post-Deployment Analysis)

การเก็บข้อมูลการใช้งาน (Monitoring):ใช้เครื่องมือ Analytics เช่น Google Analytics หรือ Firebase Analytics เพื่อติดตามการใช้งานและวิเคราะห์พฤติกรรมผู้ใช้

การปรับปรุงและพัฒนาต่อเนื่อง:นำข้อมูลที่ได้จากการวิเคราะห์มาใช้ในการปรับปรุงระบบและเพิ่มฟีเจอร์ใหม่ๆ เพื่อให้ระบบตอบสนองความต้องการของผู้ใช้ได้ดียิ่งขึ้น

4. ผลการดำเนินงาน

จากการดำเนินการพัฒนาเว็บแอพพลิเคชัน Langpra สามารถแบ่งออกเป็นหลายด้าน โดยผลการดำเนินงานจะช่วยให้เห็นภาพรวมของโครงการและความสำเร็จที่เกิดขึ้น

4.1 ผลการออกแบบและพัฒนาระบบ

4.1.1 การพัฒนาฟีเจอร์ต่าง ๆ ของระบบ

(1) พัฒนาฟีเจอร์การค้นหาและการเรียกดูข้อมูลพระเครื่อง ซึ่งสามารถค้นหาได้รวดเร็วและมีการกรองข้อมูลที่ตรงกับความต้องการของผู้ใช้

(2) เพิ่มระบบแสดงความคิดเห็นที่อนุญาตให้ผู้ใช้สามารถแสดงความคิดเห็นในแต่ละพระเครื่องได้ และความคิดเห็นเหล่านั้นจะถูกแสดงในทันที

(3) การพัฒนา UI/UX ที่ใช้งานง่ายและมีความสวยงาม ทำให้ผู้ใช้สามารถใช้งานเว็บแอพพลิเคชันได้อย่างราบรื่นและเป็นมิตรต่อผู้ใช้

4.1.2 การผสานรวม Firebase Services

(1) Firebase Authentication ถูกผสานรวมเพื่อจัดการการเข้าสู่ระบบและการลงทะเบียนของผู้ใช้ ทำให้ระบบมีความปลอดภัยและจัดการผู้ใช้ได้อย่างมีประสิทธิภาพ

(2) Firestore ถูกใช้เป็นฐานข้อมูลสำหรับการจัดเก็บข้อมูลพระเครื่อง อีเว้นท์ และคลังความรู้ ซึ่งทำให้ข้อมูลถูกจัดเก็บและดึงข้อมูลได้อย่างรวดเร็ว

(3) Firebase Storage ถูกใช้ในการจัดเก็บรูปภาพและไฟล์ต่าง ๆ ที่เกี่ยวข้องกับพระเครื่อง ทำให้การจัดการไฟล์มีประสิทธิภาพและสามารถขยายตัวได้ง่าย

4.1.3 การทดสอบและการแก้ไขข้อผิดพลาด:

(1) ระบบผ่านการทดสอบหน่วย (Unit Testing) และการทดสอบรวม (Integration Testing) เพื่อให้แน่ใจว่าแต่ละส่วนของระบบทำงานได้อย่างถูกต้องเมื่อรวมเข้าด้วยกัน

(2) การทดสอบประสิทธิภาพทำให้สามารถปรับปรุงความเร็วในการโหลดและการตอบสนองของระบบได้

(3) ข้อผิดพลาดที่พบในขั้นตอนการทดสอบถูกแก้ไขเรียบร้อยแล้ว เพื่อให้ระบบทำงานได้อย่างเสถียร

4.1.4 ผลการดำเนินงานด้านการเปิดใช้งานและการบำรุงรักษา

(1) การเปิดใช้งานระบบ : ระบบได้ถูก Deploy ไปยัง Firebase Hosting เรียบร้อยแล้ว ทำให้สามารถเข้าถึงได้ผ่านโดเมนที่กำหนดและมีการตั้งค่า SSL เพื่อความปลอดภัยในการใช้งาน ระบบมีการตั้งค่าการแจ้งเตือนผ่าน Firebase Cloud Messaging เพื่อแจ้งเตือนผู้ใช้เกี่ยวกับอีเว้นท์ใหม่ ๆ หรือพระเครื่องที่น่าสนใจ

(2) การบำรุงรักษาและการอัปเดต : ระบบมีการบำรุงรักษาอย่างต่อเนื่องเพื่อให้มั่นใจว่าทำงานได้อย่างมีประสิทธิภาพและปลอดภัย มีการอัปเดตระบบเพื่อตอบสนองต่อความต้องการของผู้ใช้ที่เพิ่มขึ้น เช่น การเพิ่มฟีเจอร์ใหม่ ๆ หรือการปรับปรุง UI/UX ตามคำแนะนำของผู้ใช้

4.1.5 ผลการดำเนินงานด้านผู้ใช้และการตอบรับ

(1) การใช้งานของผู้ใช้ : จำนวนผู้ใช้ที่ลงทะเบียนและใช้งานระบบเพิ่มขึ้นอย่างต่อเนื่อง เนื่องจากระบบมีฟังก์ชันที่ครบครันและใช้งานง่าย ผู้ใช้มีความพึงพอใจในประสิทธิภาพของระบบ โดยเฉพาะในส่วนของการค้นหาและการกรองข้อมูลพระเครื่อง

(2) การตอบรับจากผู้ใช้ : ผลการสำรวจความคิดเห็นแสดงให้เห็นว่าผู้ใช้มีความพึงพอใจต่อการใช้งานระบบในระดับสูง คำแนะนำจากผู้ใช้ถูกนำมาปรับปรุงระบบให้ดียิ่งขึ้น ทำให้ระบบมีความเป็นมิตรต่อผู้ใช้และตอบสนองต่อความต้องการที่แท้จริง

4.1.6 การวิเคราะห์หลังการเปิดใช้งาน

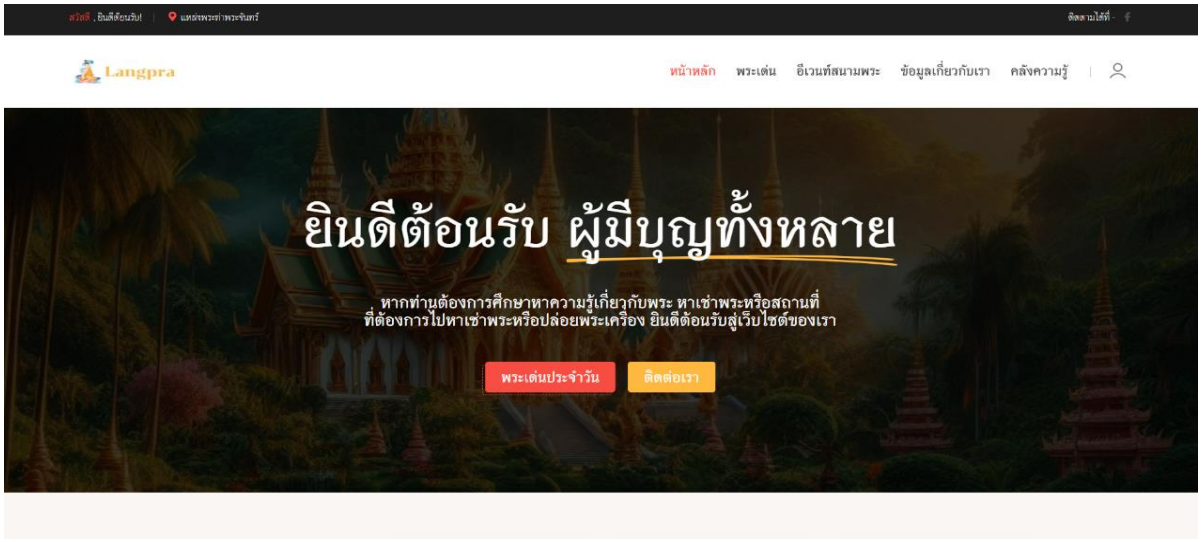
(1) การวิเคราะห์ข้อมูลการใช้งาน: การติดตามการใช้งานของผู้ใช้ผ่าน Google Analytics และ Firebase Analytics ช่วยให้เข้าใจพฤติกรรมของผู้ใช้ และทำให้สามารถปรับปรุงระบบให้ตรงกับความต้องการของผู้ใช้ได้มากขึ้น ข้อมูลที่ได้จากการวิเคราะห์นำมาใช้ในการวางแผนสำหรับการพัฒนาฟีเจอร์ใหม่ ๆ ในอนาคต

(2) การปรับปรุงต่อเนื่อง : ระบบถูกปรับปรุงอย่างต่อเนื่องตามผลการวิเคราะห์ ทำให้ระบบสามารถตอบสนองต่อความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพมีการวางแผนสำหรับการขยายฟีเจอร์และการเพิ่มประสิทธิภาพของระบบในระยะยาว

การสรุปผลการดำเนินงานนี้จะช่วยให้เห็นภาพรวมของความสำเร็จและการพัฒนาที่เกิดขึ้นในโครงการพัฒนาเว็บแอพพลิเคชัน Langpra และสามารถใช้เป็นแนวทางในการปรับปรุงและพัฒนาต่อไปในอนาคต

4.2 การใช้งานแอพพลิเคชั่น

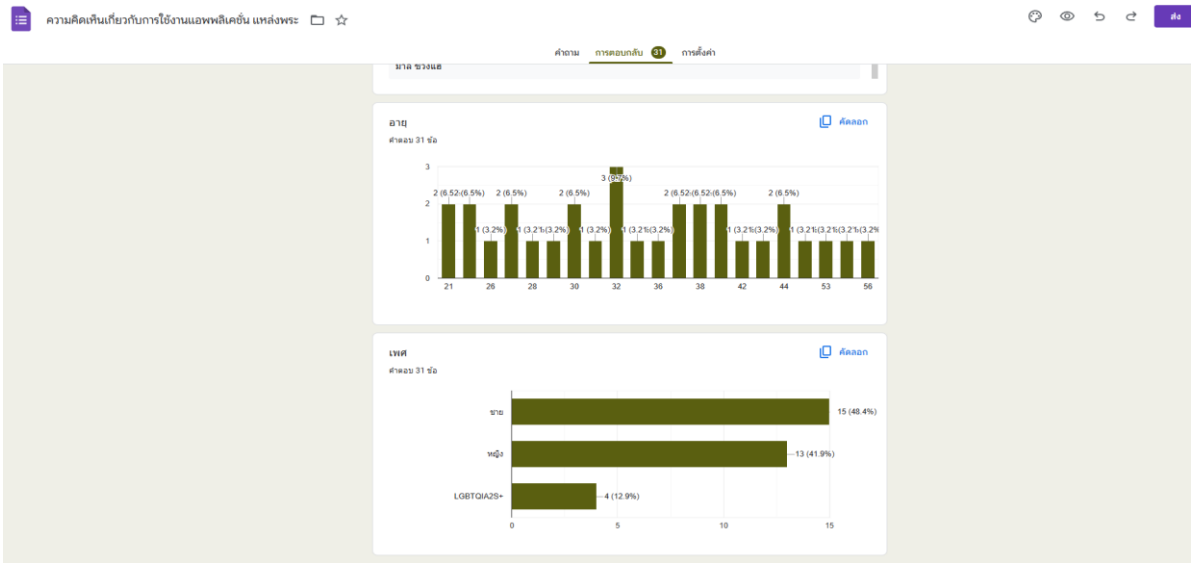
หน้าหลักของเว็บแอพพลิเคชั่นแหล่งรวมพระเครื่อง ซึ่งในหน้าหลักจะมีข้อมูลของอีเว้นท์สนามพระโชว์เป็นตัวอย่าง และมีคลิปให้ความรู้ และสามารถกดเลือกเพื่อต้องการไปหน้าต่าง ๆ ได้ โดยจะมีหัวข้อในการเลือกเพื่อไปหน้าที่ต้องการดังนี้ หน้าหลัก, พระเด่น ,อีเว้นท์สนามพระ, ข้อมูลเกี่ยวกับเรา, คลังความรู้



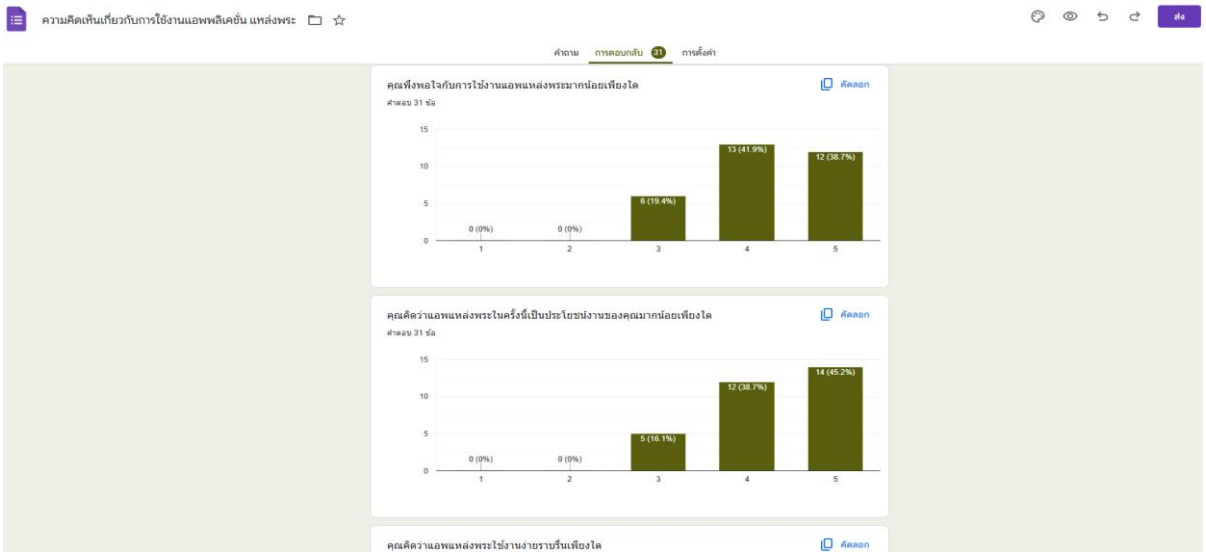
ภาพที่ 4.1 หน้าจอหลัก

4.3 ผลสำรวจในการใช้งานแอพพลิเคชั่น

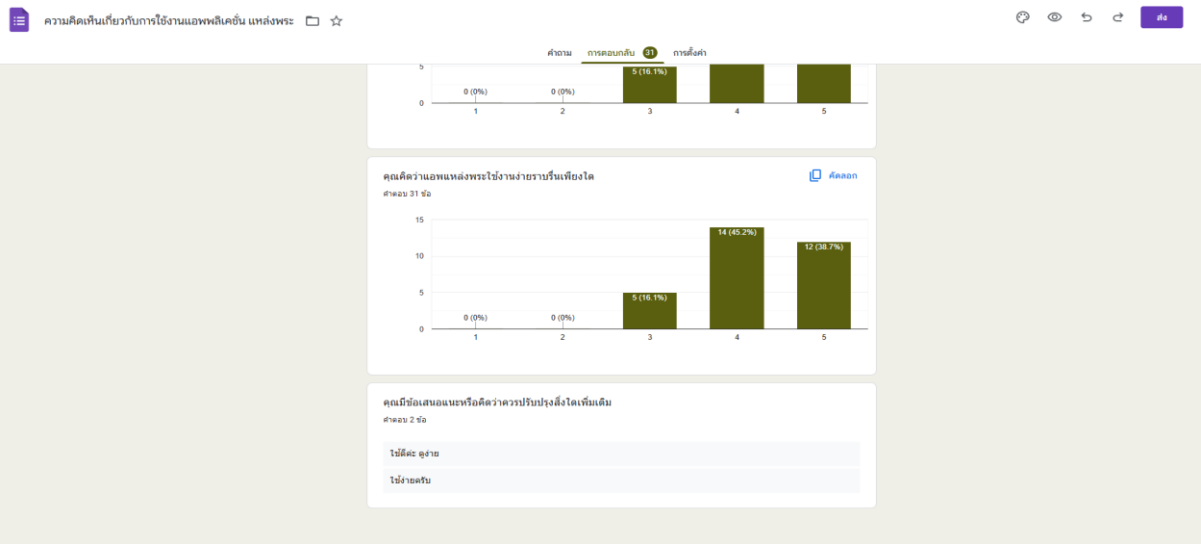
ความคิดเห็นเกี่ยวกับการใช้งานแอพพลิเคชั่นแหล่งพระ ข้อมูลพื้นฐานของผู้ตอบแบบสอบถาม จากการสร้างกลุ่มทดลองเก็บข้อมูลของผู้ใช้งาน 31 คน ซึ่งเป็นผู้เชี่ยวชาญทางด้านวงการพระเครื่องในสนามพระท่าพระจันทร์ ซึ่งที่เลือกกลุ่มคนเหล่านี้เพราะเป็นผู้ที่มีจุดประสงค์ตรงกับความต้องการใช้แอพพลิเคชันและง่ายต่อการเก็บข้อมูลมาทำการทดลอง



ภาพที่ 4.26 ความคิดเห็นเกี่ยวกับการใช้งานแอพพลิเคชั่นแหล่งพระ



ภาพที่ 4.27 ความคิดเห็นเกี่ยวกับการใช้งานแอพพลิเคชั่นแหล่งพระส่วนความพึงพอใจกับการใช้งานแอพพลิเคชั่น



ภาพที่ 4.28 ความคิดเห็นเกี่ยวกับการใช้งานแอพพลิเคชั่นแหล่งพระส่วนประโยชน์ต่องาน

1. ช่วงอายุ: ผู้ตอบแบบสอบถามส่วนใหญ่มีอายุอยู่ในช่วง 32 ปี (9.7%) และช่วงอายุอื่น ๆ มีการกระจายตัวไม่มาก เช่น อายุ 21, 26, 30, 34, 36 ปี เป็นต้น โดยช่วงอายุของผู้ตอบแบบสอบถามครอบคลุมตั้งแต่ 21 ถึง 56 ปี

2. เพศ: ผู้ตอบแบบสอบถามส่วนใหญ่เป็นเพศชาย (48.4%) รองลงมาคือเพศหญิง (41.9%) และมีเพศ LGBTQIA2S+ บางส่วน (12.9%) การตอบรับเกี่ยวกับการใช้งานแอพพลิเคชัน

3. ความพึงพอใจในการใช้งาน: ผู้ตอบแบบสอบถามมีความพึงพอใจในระดับสูง โดย 41.9% ให้คะแนน 4 และ 38.7% ให้คะแนน 5 (ซึ่งเป็นคะแนนสูงสุด)

4. การเป็นประโยชน์ในการใช้งาน: มี 45.2% ของผู้ตอบแบบสอบถามที่เห็นว่าแอพพลิเคชันเป็นประโยชน์ในการใช้งานมาก (ให้คะแนน 5) และอีก 38.7% ให้คะแนน 4

5. ความง่ายในการใช้งาน: ผู้ตอบแบบสอบถามเห็นว่าแอพพลิเคชันใช้งานง่าย โดย 45.2% ให้คะแนน 4 และ 38.7% ให้คะแนน 5

6. ข้อเสนอแนะเพิ่มเติม: มีความคิดเห็นบางส่วนที่เสนอว่าแอพพลิเคชันใช้งานง่าย และน่าจะมีการปรับปรุงต่อไป

จากข้อมูลที่ได้ ผู้ใช้ส่วนใหญ่พึงพอใจในเว็บแอพพลิเคชัน Langpra โดยมองว่ามีความง่ายต่อการใช้งานและเป็นประโยชน์ในการใช้งานจริง ข้อเสนอแนะที่ได้รับส่วนใหญ่เป็นบวก อย่างไรก็ตาม ควรมีการพิจารณาปรับปรุงแอพพลิเคชันให้ดีขึ้นอย่างต่อเนื่องตามข้อเสนอแนะที่ได้รับจากผู้ใช้

4.4 การทดสอบระบบ

การทดสอบระบบการใช้งานแอพพลิเคชั่นแหล่งพระ ได้ทำการสร้างกลุ่มทดลองเก็บการทดสอบของผู้ใช้งาน 31 คน ซึ่งเป็นผู้เชี่ยวชาญทางด้านวงการพระเครื่องในสนามพระท่าพระจันทร์ ซึ่งผลทดสอบโดยรวมจะแสดงในตารางการทดสอบระบบว่าฟังก์ชันการทำงานของระบบทุกส่วนทำงานถูกต้องหรือไม่ โดยเราจะแบ่งเป็นส่วนต่าง ๆ ของระบบและแสดงผลการทดสอบ ทั้งนี้ได้มีการทดสอบทั้งในระ MacOS และ Window โดยการทดสอบทุกระบบนั้นปกติและไม่มีปัญหาใดๆ

ตารางที่ 4.1 ตารางแสดงการทดสอบระบบเว็บแอพพลิเคชั่นแหล่งพระ

ฟังก์ชันการทำงาน	รายละเอียดการทดสอบ	ผลการทดสอบ
การล็อกอิน (Login)	ทดสอบการเข้าสู่ระบบด้วยข้อมูลผู้ใช้ที่ถูกต้องและข้อมูลที่ผิดพลาด รวมถึงการลงทะเบียนผู้ใช้ใหม่	ผ่าน
การค้นหาข้อมูล (Search Function)	ทดสอบการค้นหาข้อมูลพระเครื่องด้วยคีย์เวิร์ดต่าง ๆ และแสดงผลการค้นหา	ผ่าน
การกรองข้อมูล (Filter Data)	ทดสอบการกรองข้อมูลพระเครื่องตามประเภท, ปีที่สร้าง และแหล่งที่มา	ผ่าน
การแสดงความคิดเห็น (Comments)	ทดสอบการเพิ่มความคิดเห็นในหน้ารายละเอียดพระเครื่อง โดยผู้ใช้ที่ล็อกอินแล้ว	ผ่าน
การแสดงพระเครื่องเด่น (Featured Amulets)	ทดสอบการแสดงพระเครื่องที่ถูกทำเครื่องหมายว่าเป็นพระเด่นในหน้าหลัก	ผ่าน
การจัดการอีเว้นท์ (Events Management)	ทดสอบการเพิ่ม, แก้ไข, และลบข้อมูลอีเว้นท์ที่แสดงในระบบ	ผ่าน
การจัดการคลังความรู้ (Knowledge Management)	ทดสอบการเพิ่มบทความ, แก้ไขเนื้อหา, และลบบทความในคลังความรู้	ผ่าน
ระบบการจัดการของแอดมิน (Admin Panel)	ทดสอบการเพิ่ม/แก้ไขข้อมูลในระบบ เช่น การเพิ่มพระเครื่อง, แก้ไขข้อมูล, และลบข้อมูล	ผ่าน
การจัดการไฟล์ (File Management)	ทดสอบการอัปโหลดรูปภาพพระเครื่องและแสดงผลในระบบ	ผ่าน
การซิงค์ข้อมูลแบบเรียลไทม์ (Real-Time Data Sync)	ทดสอบการอัปเดตข้อมูลในเวลาจริง โดยที่ผู้ใช้จะเห็นข้อมูลที่เปลี่ยนแปลงได้ทันที	ผ่าน

การแจ้งเตือน (Notifications)	ทดสอบการส่งการแจ้งเตือนให้ผู้ใช้เมื่อมีการอัปเดตอีเว้นท์หรือพระเครื่องใหม่ ๆ	ผ่าน

ฟังก์ชันการทำงานทั้งหมดผ่านการทดสอบ ระบบสามารถทำงานได้อย่างถูกต้องและสมบูรณ์ตามที่ออกแบบไว้

5. บทสรุป และข้อเสนอแนะ

ในบทนี้เป็นการนำเสนอการใช้งานระบบของผู้ใช้จริงและนำเสนอข้อเสนอแนะในการพัฒนาระบบโดยมีรายละเอียดดังนี้

5.1 สรุปผล

การพัฒนาเว็บแอพพลิเคชัน Langpra ได้บรรลุวัตถุประสงค์ที่ตั้งไว้ในการสร้างแพลตฟอร์มที่รวบรวมข้อมูลเกี่ยวกับพระเครื่องได้อย่างมีประสิทธิภาพ โดยการใช้เทคโนโลยี Next.js และ Firebase ช่วยให้การพัฒนาระบบเป็นไปอย่างรวดเร็วและสามารถขยายตัวได้ในอนาคต ระบบได้รับการทดสอบและปรับปรุงเพื่อให้มั่นใจว่ามีประสิทธิภาพในการทำงานและเป็นมิตรต่อผู้ใช้ การเปิดใช้งานระบบประสบความสำเร็จ ผู้ใช้มีความพึงพอใจในประสบการณ์การใช้งาน และระบบได้รับการตอบรับที่ดีจากผู้ใช้ในด้านความสะดวกในการเข้าถึงข้อมูลและการใช้งานฟังก์ชันต่าง ๆ

5.2 ข้อเสนอแนะ

5.2.1 การพัฒนาฟีเจอร์เพิ่มเติม: ในอนาคต ควรพิจารณาการเพิ่มฟีเจอร์ที่ช่วยให้ผู้ใช้สามารถมีส่วนร่วมมากขึ้น เช่น ระบบการให้คะแนนพระเครื่อง ระบบการซื้อขายผ่านแพลตฟอร์ม และการแนะนำพระเครื่องตามความสนใจของผู้ใช้

5.2.1 การปรับปรุงความปลอดภัย: ควรมีการปรับปรุงมาตรการความปลอดภัยให้เข้มงวดขึ้น โดยเฉพาะในส่วนของการยืนยันตัวตนผู้ใช้และการจัดการข้อมูลส่วนบุคคล เพื่อป้องกันการเข้าถึงข้อมูลที่ไม่พึงประสงค์

5.2.2 การปรับปรุงประสิทธิภาพของระบบ: แม้ว่าระบบจะทำงานได้ดีในปัจจุบัน แต่ควรมีการวางแผนในการปรับปรุงประสิทธิภาพอย่างต่อเนื่อง เช่น การเพิ่มประสิทธิภาพในการดึงข้อมูลจากฐานข้อมูล การจัดการกับผู้ใช้จำนวนมากพร้อมกัน และการลดเวลาการโหลดหน้าเว็บ

5.2.3 การส่งเสริมการใช้งานและการตลาด: ควรมีการวางแผนการตลาดและการส่งเสริมการใช้งานเพื่อเพิ่มจำนวนผู้ใช้ในระบบ เช่น การสร้างแคมเปญการตลาดออนไลน์ การจัดกิจกรรมพิเศษที่เกี่ยวข้องกับพระเครื่อง และการสร้างเนื้อหาที่น่าสนใจเพื่อดึงดูดผู้ใช้ใหม่ ๆ

5.2.4 การรับฟังความคิดเห็นและปรับปรุงระบบ: ควรมีการรับฟังความคิดเห็นจากผู้ใช้อย่างต่อเนื่อง เพื่อนำข้อมูลเหล่านี้มาปรับปรุงและพัฒนาระบบให้ตอบสนองต่อความต้องการของผู้ใช้ได้ดียิ่งขึ้น

5.2.5 การเตรียมพร้อมสำหรับการขยายตัวในอนาคต: ควรวางแผนสำหรับการขยายตัวในอนาคต ไม่ว่าจะเป็นการเพิ่มเซิร์ฟเวอร์ การปรับปรุงฐานข้อมูลให้รองรับข้อมูลที่เพิ่มขึ้น หรือการพัฒนาฟีเจอร์ใหม่ ๆ ที่สอดคล้องกับการเติบโตของระบบ

การดำเนินโครงการนี้เป็นพื้นฐานที่ดีสำหรับการพัฒนาและขยายแพลตฟอร์มต่อไปในอนาคต การนำข้อเสนอแนะเหล่านี้ไปปรับใช้จะช่วยให้ระบบสามารถตอบสนองต่อความต้องการที่เปลี่ยนแปลงไปและยังคงเป็นที่นิยมในหมู่ผู้ใช้ได้อย่างยั่งยืน

บรรณานุกรม

ภาษาไทย

กรุงเทพธุรกิจ. (2566). ตลาดพระเครื่องเฟื่องฟู คาดปีนี้เงินสะพัด 22,000 ล้านบาท. bangkokbiznews [ออนไลน์]. https://www.bangkokbiznews.com/world/1123076. เข้าถึงเมื่อ: 20 กรกฎาคม 2566.

ศุภชัย สุมพานิช. (2563). *พัฒนา Web Apps ด้วย React Bootstrap + Redux*, กรุงเทพมหานคร: สำนักพิมพ์โปรวิชั่น.

สุวิทย์ สุวรรณโณ และ ณัฐธิดา สุวรรณโณ. (2555). แอพพลิเคชั่นระบบสืบค้นแหล่งท่องเที่ยวเชิงนิเวศ: อุทยานนกน้ำทะเลน้อยจังหวัดพัทลุง. มหาวิทยาลัยสงขลานครินทร์, คณะการจัดการสิ่งแวดล้อม.

วุฒิภัทร หนูยอด, เจษฎา สิงห์ทองชัย และ มานิตย์ สิงห์ทองชัย. (2565). การพัฒนาแอปพลิเคชันแหล่งท่องเที่ยวในจังหวัดจันทบุรี. *วารสารวิชาการวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครสวรรค์*, ปีที่ 7, ฉบับที่ 1, หน้า 96-105.

ธีรวุธ พุ่มเอี่ยม, พระครูโกศลวชิรกิจ และ ฆัมภิชา ตันติสันติสม. (2562). การพัฒนาเว็บไซต์นำเสนอวัดประดู่ลาย จ.กำแพงเพชร ด้วยภาพ 3 มิติ," *สักทอง : วารสารวิทยาศาสตร์และเทคโนโลยี (สทวท.)*, ปีที่ 6, ฉบับที่ 1, หน้า 95-103.

ภาษาต่างประเทศ

Davis, F. D. (1989). Perceived usefulness: perceived ease of use, and user acceptance of information technology. *MIS quarterly*, vol. 13, no. 3, pp. 319-340, September 1989.

Hegde, V. and PM, S. (2014). Web Performance Testing: Methodologies, Tools and Challenges. *International Journal of Scientific Engineering and Research (IJSER)*, vol. 2, no 1, pp. 67-73.