



ระบบแชทบอทอัจฉริยะเพื่อให้ข้อมูล สำหรับ คณะวิศวกรรมศาสตร์

Smart Information Chatbot System of Engineering

นายกรภัทร เจริญสุข

นายกิตติพงศ์ คavecเพงน้อย

นายอนุวัฒน์ กลีน์สกาน

นายไชยพัฒน์ สมวงศ์

ปริญญาอินพอร์ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาชีวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

พ.ศ.2568

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ระบบແຈຕບອທອຈະຣີຍະເພື່ອໃຫ້ຂໍ້ມູນ ສໍາຮັບ ຄณະວິສວກຮມສາສົກ

นายกรກັກທຣ ເຈີນສຸຂ

ນາຍກິດຕິພງສ ດາແພນນ້ອຍ

ນາຍອນວັນນ ກລິນໂສກຄນ

ນາຍໄຊຍພັນນ ສນແວງ

ປະຈຸບັນພົນຮັບເປັນສ່ວນໜຶ່ງຂອງການສຶກສາຕາມຫລັກສູດປະຈຸບັນພົນຮັບເປັນ

ສາຂາວິຊາວິສວກຮມຄອມພິວເຕອນ

ຄະນະວິສວກຮມສາສົກ ມາວິທານາລັບເຕັກໂນໂລຢີຮາຊມງຄລພຣະນຄຣ

ພ.ສ.2568

ລົບສິທິຂອງຄະນະວິສວກຮມສາສົກ ມາວິທານາລັບເຕັກໂນໂລຢີຮາຊມງຄລພຣະນຄຣ

Smart Information Chatbot System of Engineering

Mr. Kornrapat Charoensuk

Mr. Gidtipong Capangnoi

Mr. Anuwat Klinsopon

Mr. Chaiyapat Somwang

THIS PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE BACHELOR OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
RAJAMANGALA UNIVERSITY OF TECHNOLOGY PHRA NAKHON
2025



ชื่อปริญญาบัตร
ชื่อนักศึกษา ระบบแข่งขันชิงรางวัล สำหรับ คณะวิศวกรรมศาสตร์
นายอนุวัฒน์ กลินโภณ รหัส 056550405137-8
นายไชยพัฒน์ สมวงศ์ รหัส 056550405140-2
นายกรภัทร เจริญสุข รหัส 056550405145-1
นายกิตติพงศ์ คานะน้อย รหัส 056550405147-7
ที่ปรึกษาปริญญาบัตร อาจารย์ ผศ.ดร.เกรียงไกร เหลืองคำพล
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2568

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร
อนุมัติให้ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญาบัตร

..... ประธานกรรมการ

()

..... กรรมการ

()

..... กรรมการ

()

..... หัวหน้าสาขา

()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร



Project Title Smart Information Chatbot System of Engineering

By Mr. Anuwat Klinsopon 056550405137-8

Mr. Chaiyapat Somwang 056550405140-2

Mr. Kornrapat Charoensuk 056550405145-1

Mr. Gidtipong Capangnoi 056550405147-7

Project Advisor Asst. Prof. Dr. Kriengkri Luangampol

Major program Computer Engineering

Academic Year 2025

Accepted by Department of Computer Engineering, the Faculty of Engineering Rajamangala University of Technology Phra Nakhon in Partial Fulfillment of the Requirements for the Bachelor Degree of Engineering.

Project Committee

..... Chairperson

()

..... Member

()

..... Member

()

..... Head of Department

()

Copyright of the Faculty of Engineering Rajamangala University of Technology Phra Nakhon.



ชื่อปริญญาบัตร	ระบบแข่งขันทักษะเพื่อให้ข้อมูล สำหรับ คณะวิศวกรรมศาสตร์		
ชื่อนักศึกษา	นายอนุวัฒน์ กลินสกุณ	รหัส 056550405137-8	
	นายไชยพัฒน์ สมแวง	รหัส 056550405140-2	
	นายกรภัทร เจริญสุข	รหัส 056550405145-1	
	นายกิตติพงศ์ คำแพงน้อย	รหัส 056550405147-7	
ที่ปรึกษาปริญญาบัตร	อาจารย์ ผศ.ดร.เกรียงไกร เหลืองคำพล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ปีการศึกษา	2568		

บทคัดย่อ

ระบบผู้ช่วยอัจฉริยะสำหรับคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร พัฒนาเพื่อแก้ปัญหาการเข้าถึงข้อมูลและลดภาระงานเจ้าหน้าที่ในการตอบคำถามซ้ำๆ โดยใช้เทคโนโลยี Hybrid Chatbot ที่ผสมผสาน Rule-based System และ Machine Learning (TF-IDF + Logistic Regression) ระบบประกอบด้วย Frontend (HTML5, CSS3, JavaScript), Backend (PHP 8.0+), AI Engine (Python Flask), และฐานข้อมูล MySQL จัดเก็บข้อมูล 4 ตาราง (staff, faq, news, chat_logs) ใช้สถาปัตยกรรม 3-Tier และรองรับ RESTful API ระบบสามารถตอบคำถามอัตโนมัติ ค้นหาข้อมูลบุคลากร 118 ท่าน แสดงข่าวสารผ่าน Web Scraping และบันทึกประวัติการสนทนาร้อม Confidence Score ผลการพัฒนาพบว่า ระบบให้บริการตลอด 24 ชั่วโมง ลดภาระงานกว่า 80% และตอบสนองภายใน 500 มิลลิวินาที สอดคล้องกับแนวทางมหาวิทยาลัยดิจิทัล

(ปริญญาบัตรนี้จำนวน - หน้า)

คำสำคัญ: ระบบแข่งขัน, ปัญญาประดิษฐ์, การประมวลผลภาษาธรรมชาติ, TF-IDF, Logistic Regression, Web Scraping, คณะวิศวกรรมศาสตร์



Project Title	Smart Information Chatbot System of Engineering	
By	Mr. Anuwat Klinsopon	056550405137-8
	Mr. Chaiyapat Somwang	056550405140-2
	Mr. Kornrapat Charoensuk	056550405145-1
	Mr. Gidtipong Capangnoi	056550405147-7
Project Advisor	Asst. Prof. Dr. Kriengkri Luangampol	
Major program	Computer Engineering	
Academic Year	2025	

Abstract

Smart Information Chatbot System for the Faculty of Engineering, Rajamangala University of Technology Phra Nakhon Developed to solve data accessibility problems and reduce staff workload in answering repetitive questions utilizing Hybrid Chatbot technology that combines a Rule-based System and Machine Learning (TF-IDF + Logistic Regression). The system consists of a Frontend (HTML5, CSS3, JavaScript), Backend (PHP 8.0+), AI Engine (Python Flask), and a MySQL database storing data in 4 tables (staff, faq, news, chat_logs). It uses a 3-Tier Architecture and supports RESTful API. The system is capable of automatic question answering, searching information on 118 staff members, displaying news via Web Scraping, and recording chat logs with Confidence Scores. Development results show that the system provides service 24 hours a day, reduces workload by over 80%, and responds within 500 milliseconds, consistent with the Digital University guidelines.

(Total - Pages)

Keywords: Chatbot System, Artificial Intelligence, Natural Language Processing, TF-IDF, Logistic Regression, Web Scraping, Faculty of Engineering

กิตติกรรมประกาศ

การจัดทำปริญญาในเรื่อง ระบบแซตบอทอัจฉริยะเพื่อให้ข้อมูล สำหรับ คณะวิศวกรรมศาสตร์ ในครั้งนี้ สำเร็จลุล่วงตามวัตถุประสงค์ได้ด้วยดี เนื่องด้วยได้รับความอนุเคราะห์และความกรุณาอย่างสูงจาก ผู้ช่วยศาสตราจารย์ ดร.เกรียงไกร เหลืองอมาพ อาจารย์ที่ปรึกษาโครงงาน ที่ได้กรุณاسلับเวลาถ่ายทอดความรู้ ให้แนวคิด และคำแนะนำอันทรงคุณค่าในการพัฒนาระบบแซตบอทให้มีความถูกต้องสมบูรณ์ ตลอดจนเอาใจ ใส่ติดตามความคืบหน้าและเป็นกำลังใจสำคัญให้แก่คณาจัดทำเสมอมา ผู้จัดทำรู้สึกซาบซึ้งในความกรุณาและ ขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณคณาจารย์ภาควิชาชีวกรรมคอมพิวเตอร์ทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ และให้คำแนะนำอันเป็นประโยชน์ในการจัดทำโครงงาน ตลอดจนอำนวยความสะดวกในการใช้งานสถานที่ และอุปกรณ์ต่าง ๆ ที่จำเป็นต่อการดำเนินงาน

ท้ายที่สุดนี้ คณาจัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่ให้การสนับสนุนทั้งในด้าน ทุนทรัพย์และเป็นกำลังใจสำคัญที่ผลักดันให้ผู้จัดทำมีความมุ่งมั่นและอดทน จนสามารถพัฒนาอุปสรรคต่าง ๆ และทำโครงงานนี้ได้สำเร็จ รวมถึงขอขอบคุณเพื่อน ๆ ทุกคนที่เคยให้ความช่วยเหลือ และเปลี่ยนความคิดเห็น และเป็นกำลังใจให้แก่กันเสมอมา

คณาจัดทำ

สารบัญ

สารบัญ(ต่อ)

สารบัญตาราง

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบัน เทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence: AI) และระบบประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) ได้รับการพัฒนาและนำมาใช้ในหลายด้าน เพื่อช่วยให้การสื่อสารระหว่างมนุษย์และเครื่องคอมพิวเตอร์เป็นไปอย่างมีประสิทธิภาพ หนึ่งในเทคโนโลยีที่ได้รับความนิยมคือ "ระบบแชットบอต (Chatbot)" ซึ่งสามารถตอบกับผู้ใช้งานได้อัตโนมัติผ่านช่องทาง โดยมีจุดประสงค์เพื่อให้ข้อมูลและบริการต่าง ๆ ได้อย่างรวดเร็วและสะดวก

ภายในสถาบันการศึกษา โดยเฉพาะคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร มีกิจกรรมที่นักศึกษา บุคลากร และบุคคลทั่วไปต้องการทราบอยู่เสมอ เช่น ข้อมูลหลักสูตรและสาขาวิชาต่างๆ (10 สาขา) ข้อมูลอาจารย์และบุคลากรกว่า 118 ท่าน ข่าวสารและกิจกรรมของคณะที่มีการอัปเดตอย่างต่อเนื่อง คำถามที่พบบ่อย (FAQ) เกี่ยวกับการสมัครเรียน ทุนการศึกษา การกู้ยืมเงินนักศึกษา และข้อมูลการติดต่อน่วยงานต่างๆ ซึ่งข้อมูลเหล่านี้มีการเปลี่ยนแปลงอยู่ตลอดเวลา การให้ข้อมูลด้วยวิธีการเดิม เช่น การโทรศัพท์สอบถามหรือการค้นหาผ่านเว็บไซต์ มักทำให้ผู้ใช้เสียเวลาในการค้นหาข้อมูลที่กระจายอยู่ในหลายแหล่ง และเพิ่มภาระให้กับเจ้าหน้าที่ในการตอบคำถามซ้ำๆ ที่คล้ายกัน

ดังนั้นจึงมีแนวคิดในการพัฒนา "ระบบผู้ช่วยอัจฉริยะ (Smart Information Chatbot System) สำหรับคณะวิศวกรรมศาสตร์" โดยระบบใช้เทคโนโลยี Hybrid Chatbot ที่ผสมผสานระหว่าง Rule-based System และ Machine Learning (TF-IDF + Logistic Regression) เพื่อตอบคำถามได้อย่างแม่นยำ ระบบสามารถเข้ามาร่วมตัวกับฐานข้อมูล MySQL และดึงข้อมูลข่าวสารจากเว็บไซต์มหาวิทยาลัยแบบอัตโนมัติผ่าน Web Scraping เพื่อให้ข้อมูลที่ถูกต้องและเป็นปัจจุบันแก่ผู้ใช้งานได้ตลอดเวลา ระบบดังกล่าวจะช่วยอำนวยความสะดวกในการเข้าถึงข้อมูล ลดภาระงานเจ้าหน้าที่ เพิ่มประสิทธิภาพในการสื่อสาร และสร้างภาพลักษณ์ที่ทันสมัยให้กับคณะ นอกจากนี้ ระบบยังเก็บบันทึกประวัติการสนทนา (Chat Logs) เพื่อนำไปวิเคราะห์พฤติกรรมผู้ใช้งาน วัดค่าความมั่นใจ (Confidence Score) ในการตอบคำถาม และเวลาในการตอบสนอง (Response Time) เพื่อพัฒนาการให้บริการอย่างต่อเนื่อง รวมถึงมีระบบผู้ดูแล (Admin Dashboard) สำหรับบริหารจัดการข้อมูล FAQ ข่าวสาร บุคลากร และสถิติการใช้งานได้อย่างเป็นระบบ

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อพัฒนาระบบผู้ช่วยอัจฉริยะ (Chatbot) ที่สามารถตอบคำถามอัตโนมัติ (FAQ) เกี่ยวกับข้อมูลของคณะวิศวกรรมศาสตร์ เช่น ข้อมูลหลักสูตร ทุนการศึกษา ข่าวสาร และช่องทางการติดต่อ ได้อย่างถูกต้องและรวดเร็ว

1.2.2 เพื่อเชื่อมต่อและดึงข้อมูลจากเว็บไซต์และฐานข้อมูล MySQL ของคณะแบบอัตโนมัติ ทำให้ข้อมูลที่นำเสนอ มีความถูกต้องและเป็นปัจจุบันอยู่เสมอ

1.2.3 เพื่อให้ผู้ใช้งานสามารถค้นหาข้อมูลบุคลากรหรืออาจารย์ทั้ง 118 ท่าน ได้อย่างสะดวก เช่น ชื่อ ตำแหน่ง สาขาวิชา อีเมล หมายเลขโทรศัพท์ และความเชี่ยวชาญ

1.2.4 เพื่อพัฒนาระบบ AI Engine ด้วย Machine Learning (TF-IDF + Logistic Regression) สำหรับการจำแนกเจตนา (Intent Classification) และตอบคำถามได้อย่างแม่นยำ

1.2.5 เพื่อสร้างระบบจัดเก็บและวิเคราะห์ข้อมูลการสนทนາ (Chat Logs) พร้อมวัดค่า Confidence Score และ Response Time สำหรับใช้ปรับปรุงระบบอย่างต่อเนื่อง

1.2.6 เพื่อพัฒนาระบบผู้ดูแล (Admin Dashboard) ที่สามารถจัดการข้อมูล FAQ ข่าวสาร บุคลากร และตรวจสอบสถิติการใช้งานได้อย่างเป็นระบบ

1.3 ขอบเขตของโครงการ

1.3.1 ลักษณะของระบบ

โครงการนี้เป็นการพัฒนาระบบผู้ช่วยอัจฉริยะ (Smart Information Chatbot System) สำหรับคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร โดยใช้เทคโนโลยีเว็บแอปพลิเคชัน (Web Application) แบบ Hybrid Chatbot ที่สามารถทำงานได้ทั้งบนคอมพิวเตอร์และอุปกรณ์พกพา

1.3.2 ขอบเขตการทำงานของระบบ

1.3.2.1 ระบบสามารถตอบคำถามอัตโนมัติ (FAQ Chatbot) โดยใช้ Machine Learning (TF-IDF + Logistic Regression) ร่วมกับ Keyword Matching เพื่อดึงคำตอบจากฐานข้อมูล พร้อมแสดงค่า Confidence Score

1.3.2.2 ระบบสามารถดึงข่าวสารจากเว็บไซต์ของคณะวิศวกรรมศาสตร์แบบอัตโนมัติผ่าน Web Scraping และอัปเดตเข้าสู่ฐานข้อมูลตาราง news

1.3.2.3 ระบบสามารถค้นหาข้อมูลบุคลากรทั้ง 118 ท่าน จากฐานข้อมูลตาราง staff โดยแสดงข้อมูล ชื่อ (ไทย/อังกฤษ) ตำแหน่ง สาขา อีเมล เบอร์โทรศัพท์ ความเชี่ยวชาญ และรูปภาพ

1.3.2.4 ระบบสามารถให้ข้อมูลการติดต่อกัน เช่น หมายเลขโทรศัพท์ อีเมล เฟซบุ๊ก และเว็บไซต์ พร้อมแสดงลิงก์ที่คลิกได้

1.3.2.5 ระบบสามารถบันทึกประวัติการสนทนา (Chat Logs) ลงในตาราง chat_logs พร้อมข้อมูล session_id, user_message, bot_response, sources, confidence, response_time_ms, user_ip, และ user_agent

1.3.2.6 มีส่วนติดต่อผู้ใช้ (User Interface) ที่ใช้งานง่าย ทันสมัย รองรับ Responsive Design สำหรับทั้งคอมพิวเตอร์และสมาร์ตโฟน

1.3.3 ขอบเขตด้านผู้ใช้งาน

1.3.3.1 ผู้ใช้งานทั่วไป (นักศึกษา บุคลากร และผู้สนใจทั่วไป) สามารถใช้งานแชットบอตเพื่อสอบถามข้อมูลต่างๆ ได้ผ่านเว็บเบราว์เซอร์

1.3.3.2 ผู้ดูแลระบบ (Administrator) สามารถจัดการข้อมูลภายใน เช่น เพิ่ม/แก้ไข/ลบ FAQ ข่าวสาร บุคลากร และดูสถิติการใช้งานได้ผ่าน Admin Dashboard

1.3.4 ขอบเขตด้านเทคนิคและการพัฒนา

1.3.4.1 ระบบพัฒนาด้วย Frontend (HTML5, CSS3, JavaScript) และ Backend (PHP 8.0+ พร้อม PDO สำหรับจัดการฐานข้อมูล)

1.3.4.2 ใช้ฐานข้อมูล MySQL (utf8mb4) สำหรับจัดเก็บข้อมูล 4 ตารางหลัก คือ staff (บุคลากร), faq (คำถาม-คำตอบ), news (ข่าวสาร) และ chat_logs (ประวัติการสนทนา)

1.3.4.3 พัฒนาระบบ AI Engine ด้วย Python 3.x + Flask Framework ใช้เทคนิค TF-IDF (Term Frequency-Inverse Document Frequency) ร่วมกับ Logistic Regression จาก Scikit-learn สำหรับการจำแนกเจตนา (Intent Classification)

1.3.4.4 มีการเชื่อมต่อกับเว็บไซต์มหาวิทยาลัยเพื่อดึงข้อมูลข่าวสารแบบอัตโนมัติผ่าน Web Scraping Script (Python + BeautifulSoup)

1.3.4.5 ระบบใช้สถาปัตยกรรมแบบ 3-Tier Architecture (Presentation Layer, Application Layer, Data Layer) และรองรับ RESTful API สำหรับการสื่อสารระหว่าง Frontend และ Backend

1.3.4.6 ระบบทำงานบน Apache Web Server (XAMPP Stack) และสามารถทดสอบผ่านเว็บเบราว์เซอร์มาตรฐาน เช่น Google Chrome, Microsoft Edge, Safari รองรับ Responsive Design สำหรับอุปกรณ์ทุกชนิด

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 อำนวยความสะดวกให้กับนักศึกษา บุคลากร และผู้สนใจทั่วไป ในการค้นหาข้อมูลข่าวสาร บุคลากร และช่องทางการติดต่อของคณะวิศวกรรมศาสตร์ได้อย่างรวดเร็วและถูกต้องตลอด 24 ชั่วโมง

1.4.2 ลดภาระงานของเจ้าหน้าที่ในการตอบคำถามซ้ำๆ และเพิ่มประสิทธิภาพในการให้บริการข้อมูลแก่ผู้ใช้งานได้มากกว่า 80% จากเดิม

1.4.3 ช่วยให้คณะวิศวกรรมศาสตร์มีระบบบริการข้อมูลที่ทันสมัย สอดคล้องกับแนวทางของมหาวิทยาลัยดิจิทัล (Digital University) และเพิ่มภาพลักษณ์ความเป็นองค์กรอัจฉริยะ (Smart Organization)

1.4.4 สามารถนำข้อมูลจากประวัติการสนทนา (Chat Logs) พร้อมค่า Confidence Score และ Response Time มาวิเคราะห์ เพื่อพัฒนาระบบทอบคำถามให้มีความแม่นยำและครอบคลุมมากยิ่งขึ้น

1.4.5 เป็นแหล่งเรียนรู้และตัวอย่างการประยุกต์ใช้เทคโนโลยีปัญญาประดิษฐ์ (AI) Machine Learning และระบบประมวลผลภาษาธรรมชาติ (NLP) ในการพัฒนาแอปพลิเคชันที่สามารถนำไปต่อยอดในงานอื่นๆ ได้

1.4.6 สนับสนุนให้นักศึกษาผู้จัดทำโครงการได้พัฒนาทักษะด้านการวิเคราะห์ ออกแบบ และพัฒนาระบบจริง รวมถึงการทำงานร่วมกันเป็นทีมในรูปแบบของโครงการเทคโนโลยีสารสนเทศครบวงจร

1.5 ขั้นตอนการศึกษา

การดำเนินโครงการ "ระบบผู้ช่วยอัจฉริยะ (Smart Information Chatbot System) สำหรับคณะวิศวกรรมศาสตร์" มีขั้นตอนการศึกษาดำเนินงานดังนี้

1.5.1 ศึกษาและวิเคราะห์ข้อมูลเบื้องต้น

1.5.1.1 ศึกษาข้อมูลพื้นฐานเกี่ยวกับเทคโนโลยีปัญญาประดิษฐ์ (AI), ระบบประมวลผลภาษาธรรมชาติ (NLP), Machine Learning (TF-IDF, Logistic Regression) และการทำงานของ Hybrid Chatbot

1.5.1.2 วิเคราะห์ระบบการให้ข้อมูลของคณะวิศวกรรมศาสตร์ในปัจจุบัน เพื่อรับปัญหาและข้อจำกัดของวิธีการให้บริการเดิม

1.5.1.3 ศึกษาความต้องการของผู้ใช้งาน (นักศึกษา บุคลากร และผู้สนใจทั่วไป) เพื่อกำหนดขอบเขตและฟังก์ชันของระบบ

1.5.2 ออกแบบระบบ

1.5.2.1 ออกแบบโครงสร้างระบบ (System Architecture) แบบ 3-Tier ให้เหมาะสมกับการเชื่อมต่อฐานข้อมูล MySQL และเว็บไซต์ของมหาวิทยาลัย

1.5.2.2 ออกแบบฐานข้อมูล (Database Design) สำหรับจัดเก็บ 4 ตารางหลัก: staff, faq, news, chat_logs พร้อมกำหนด Primary Key, Foreign Key และ Index

1.5.2.3 ออกแบบ AI Engine Pipeline ตั้งแต่ Text Preprocessing, TF-IDF Vectorization, Model Training ไปจนถึง Prediction & Response Generation

1.5.2.4 ออกแบบส่วนติดต่อผู้ใช้ (User Interface Design) ให้ใช้งานง่าย ทันสมัย รองรับ Responsive Design สำหรับทั้งคอมพิวเตอร์และอุปกรณ์พกพา

1.5.3 พัฒนาระบบ

1.5.3.1 พัฒนา Backend API ด้วย PHP 8.0+ พร้อม PDO สำหรับจัดการฐานข้อมูล MySQL และเชื่อมต่อกับ AI Engine

1.5.3.2 พัฒนา AI Engine ด้วย Python Flask ใช้ Scikit-learn สำหรับ TF-IDF + Logistic Regression และ pythainlp สำหรับ Thai Tokenization

1.5.3.3 พัฒนามódูล Web Scraping (Python + BeautifulSoup) เพื่อดึงข้อมูลข่าวสารจากเว็บไซต์คณวิศวกรรมศาสตร์แบบอัตโนมัติ

1.5.3.4 พัฒนา Frontend Interface (HTML5, CSS3, JavaScript) ให้มีลักษณะคล้ายระบบสนทนาระบบทาม (Chat Interface) ที่ทันสมัยและใช้งานง่าย

1.5.3.5 พัฒนาระบบผู้ดูแล (Admin Dashboard) สำหรับจัดการข้อมูล FAQ ข่าวสาร บุคลากร และดูสถิติการใช้งาน Chat Logs พร้อม Confidence Score และ Response Time

1.5.4 ทดสอบระบบ (System Testing)

1.5.4.1 ทดสอบการทำงานของแต่ละส่วน เช่น การตอบคำถาม (Accuracy Testing) การค้นหาข้อมูลบุคลากร และการดึงข่าวสาร (Web Scraping)

1.5.4.2 ทดสอบ AI Model โดยวัด Accuracy, Precision, Recall, F1-Score จากชุดข้อมูลทดสอบ (Test Dataset)

1.5.4.3 ทดสอบการเชื่อมต่อกับฐานข้อมูล MySQL และเว็บไซต์ของมหาวิทยาลัย รวมถึงทดสอบ Response Time ของระบบ

1.5.4.4 ทดสอบการใช้งานบนอุปกรณ์และเบราว์เซอร์ต่างๆ (Cross-browser & Cross-device Testing) เพื่อประเมินความถูกต้องและเสถียรของระบบ

1.5.5 ประเมินผลและปรับปรุงระบบ

1.5.5.1 รวบรวมความคิดเห็นจากผู้ใช้งานจริง (User Acceptance Testing) เพื่อประเมินความพึงพอใจและประสิทธิภาพของระบบ

1.5.5.2 ปรับปรุงระบบให้ตอบสนองต่อความต้องการของผู้ใช้งานยิ่งขึ้น โดยอิงจากข้อเสนอแนะและผลการทดสอบ

1.5.5.3 วิเคราะห์ข้อมูลจาก Chat Logs พร้อมค่า Confidence Score เพื่อรับคำแนะนำที่ระบบตอบได้ไม่ดี และเพิ่ม FAQ ใหม่เข้าระบบ

1.5.5.4 ปรับปรุง AI Model ด้วยการ Retrain ด้วยข้อมูล FAQ ที่เพิ่มเติม เพื่อเพิ่มความแม่นยำของระบบ

1.5.6 จัดทำรายงานและสรุปผลโครงการ

1.5.6.1 สรุปผลการพัฒนาและทดสอบระบบ

1.5.6.2 วัดผลประสิทธิภาพของระบบ เช่น ความแม่นยำในการตอบคำถาม (Accuracy) เวลาตอบสนอง (Response Time) และความพึงพอใจของผู้ใช้งาน

1.5.6.3 จัดทำรายงานโครงการฉบับสมบูรณ์ พร้อมนำเสนอผลการดำเนินงาน

1.6 นิยามศัพท์และคำย่อ

1.6.1 AI (Artificial Intelligence) - ปัญญาประดิษฐ์ เทคโนโลยีที่ให้คอมพิวเตอร์สามารถเลียนแบบความสามารถของมนุษย์ในการคิด เรียนรู้ และตัดสินใจ

1.6.2 NLP (Natural Language Processing) - การประมวลผลภาษาธรรมชาติ เทคนิคที่ช่วยให้คอมพิวเตอร์เข้าใจ ตีความ และตอบสนองต่อภาษาของมนุษย์

1.6.3 Chatbot - ระบบแชทบอท โปรแกรมที่ออกแบบมาเพื่อสนทนาระหว่างผู้ใช้งานผ่านข้อความอัตโนมัติ

1.6.4 FAQ (Frequently Asked Questions) - คำถามที่พบบ่อย ชุดคำถามและคำตอบที่ผู้ใช้งานมักสอบถามเป็นประจำ

1.6.5 Machine Learning - การเรียนรู้ของเครื่อง เทคนิคที่ทำให้คอมพิวเตอร์สามารถเรียนรู้จากข้อมูลและปรับปรุงประสิทธิภาพโดยไม่ต้องถูกโปรแกรมโดยตรง

1.6.6 TF-IDF (Term Frequency-Inverse Document Frequency) - เทคนิคการวัดความสำคัญของคำในเอกสาร โดยพิจารณาความถี่ของคำและความหายากของคำนั้นในชุดเอกสารทั้งหมด

1.6.7 Logistic Regression - อัลกอริทึมการเรียนรู้แบบมีผู้สอน (Supervised Learning) ที่ใช้สำหรับการจำแนกประเภท (Classification)

1.6.8 Web Scraping - เทคนิคการดึงข้อมูลจากเว็บไซต์โดยอัตโนมัติ

1.6.9 RESTful API (Representational State Transfer API) - รูปแบบสถาปัตยกรรมการสื่อสารระหว่างระบบผ่าน HTTP Protocol

1.6.10 Admin Dashboard - ระบบหลังบ้านสำหรับผู้ดูแลระบบในการจัดการข้อมูลและตรวจสอบ
สถิติการใช้งาน

1.6.11 Chat Logs - บันทึกประวัติการสนทนาระหว่างผู้ใช้งานและระบบแชทบอท

1.6.12 Confidence Score - ค่าความมั่นใจที่ระบบมีต่อคำตอบที่ให้ โดยมีค่าระหว่าง 0.00 ถึง 1.00

1.6.13 Response Time - ระยะเวลาที่ระบบใช้ในการประมวลผลและตอบกลับคำถาม วัดเป็น
มิลลิวินาที (ms)

1.6.14 RMUTP - มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร (Rajamangala University of
Technology Phra Nakhon)

1.6.15 Hybrid Chatbot - แชทบอทแบบผสมผสานที่ใช้ทั้ง Rule-based และ AI-powered เพื่อ^{เพิ่มประสิทธิภาพการตอบคำถาม}

1.6.16 Intent Classification - การจำแนกเจตนาของผู้ใช้งาน เพื่อรับรู้ผู้ใช้ท้องการถามหรือขอ^{ข้อมูลเรื่องใด}

1.6.17 Responsive Design - การออกแบบเว็บไซต์ให้สามารถแสดงผลได้เหมาะสมกับทุกขนาด
หน้าจอ

1.6.18 3-Tier Architecture - สถาปัตยกรรมระบบ 3 ชั้น ประกอบด้วย Presentation Layer,
Application Layer และ Data Layer

1.7 วางแผนการศึกษา

	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
	2568		2569		
ศึกษาข้อมูลและวิเคราะห์ระบบ					
ออกแบบระบบ (System Design)					
พัฒนาโมดูล Chatbot					
พัฒนาโมดูลดึงข่าว (Fetch News)					
พัฒนาโมดูลค้นหาบุคลากร (Staff Info)					
พัฒนา UI ของแชทบอท					
พัฒนาระบบผู้ดูแล (Admin Dashboard)					
ทดสอบระบบ (Testing)					
ประเมินผลและปรับปรุง					
จัดทำรายงานและสรุปผล					

ตารางที่ 1.1 แผนการดำเนินงาน

บทที่ 2

ทฤษฎีและเนื้อหาที่เกี่ยวข้อง

2.1 บทนำ

ในการพัฒนาระบบ [Smart Information Chatbot System of Engineering] คณะผู้จัดทำได้ศึกษาค้นคว้าหลักการ ทฤษฎี และเทคโนโลยีที่เกี่ยวข้อง เพื่อนำมาประยุกต์ใช้ในการออกแบบและพัฒนาระบบให้มีประสิทธิภาพ สามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างถูกต้องแม่นยำ โดยเนื้อหาในบทนี้จะครอบคลุมองค์ความรู้ตั้งแต่พื้นฐานกระบวนการทำงานของ แซทบอท, การประมวลผลภาษาธรรมชาติ (NLP) เพื่อให้ระบบเข้าใจภาษาไทย, เทคนิคการสืบค้นและจัดเก็บข้อมูลอย่างเป็นระบบ, สถาปัตยกรรมของเว็บแอปพลิเคชันและการเชื่อมต่อผ่าน API, ตลอดจนหลักการออกแบบส่วนติดต่อผู้ใช้ (UI/UX) เพื่อสร้างประสบการณ์การใช้งานที่ดี ซึ่งมีรายละเอียดดังต่อไปนี้

2.2. ทฤษฎีเกี่ยวกับระบบแซทบอท (Chatbot System)

2.2.1 ประเภทของแซทบอท: อธิบายความแตกต่างระหว่าง Rule-based (ทำงานตามกฎ), AI-powered (ใช้ Machine Learning) และ Hybrid (ลูกผสม) ซึ่งโปรเจคนี้อาจเน้นแบบ Hybrid หรือ Rule-based ที่มีความซับซ้อน

2.2.1.1 Rule-based Chatbot (แซทบอทแบบอิงกฎ) แซทบอทประเภทนี้ทำงานโดยอาศัยชุดคำสั่งหรือกฎเกณฑ์ที่ถูกกำหนดไว้อย่างตายตัว (Predefined Rules) ภายใต้โครงสร้างแบบผังต้นไม้มีการตัดสินใจ (Decision Tree) ระบบจะทำหน้าที่ตัดสินใจโดยอิงจากผู้ใช้งานตามเส้นทางที่ถูกออกแบบไว้ล่วงหน้าเท่านั้น

2.2.1.2 AI-powered Chatbot (แซทบอทปัญญาประดิษฐ์) หรือเรียกว่า Conversational AI คือแซทบอทที่ขับเคลื่อนด้วยเทคโนโลยีปัญญาประดิษฐ์ โดยอาศัยการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) และการเรียนรู้ของเครื่อง (Machine Learning) เพื่อทำความเข้าใจเจตนา (Intent) ของผู้ใช้งาน

2.2.1.3 Hybrid Chatbot (แซทบอทแบบลูกผสม) คือการบูรณาการเทคโนโลยีระหว่างแซทบอทแบบอิงกฎ (Rule-based) และแซทบอทปัญญาประดิษฐ์ (AI-powered) เข้าด้วยกัน เพื่อตีจุดเด่นของทั้งสองระบบมาใช้ในการเพิ่มประสิทธิภาพการให้บริการสูงสุด

2.2.2 องค์ประกอบหลัก (Architecture)

โครงสร้างการทำงานของแซทบอทโดยทั่วไปประกอบด้วย 4 ส่วนสำคัญที่ทำงานประสานกัน เพื่อแปลงข้อมูลนำเข้า (Input) จากมนุษย์ ให้เป็นการตอบสนอง (Response) ที่เหมาะสม ดังนี้

2.2.2.1 ส่วนติดต่อผู้ใช้ (User Interface - UI) ส่วนติดต่อผู้ใช้ หรือ UI คือช่องทาง (Channel) ที่ทำหน้าที่เป็นสื่อกลางในการรับส่งข้อมูลระหว่างผู้ใช้งานกับระบบคอมพิวเตอร์ ทำหน้าที่รับข้อมูลนำเข้า (Input) ซึ่งอาจอยู่ในรูปแบบข้อความ (Text) หรือเสียง (Voice) และแสดงผลลัพธ์ (Output) กลับไปยังผู้ใช้

1. ทฤษฎีที่เกี่ยวข้อง: หลักการปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ (Human-Computer Interaction: HCI) และการออกแบบประสบการณ์ผู้ใช้ (User Experience: UX)

2. รูปแบบการทำงาน: ในปัจจุบัน UI มักจะเชื่อมต่อผ่าน Platform ต่างๆ เช่น Web Widget, Mobile Application หรือ Social Media Platform (Line, Facebook Messenger) ผ่านทาง API (Application Programming Interface)

2.2.2.2 ส่วนประมวลผลภาษาธรรมชาติ (NLP Engine) NLP (Natural Language Processing) คือกลไกหลักที่เปรียบเสมือนสมองของแซทบอท ทำหน้าที่วิเคราะห์และทำความเข้าใจภาษา มนุษย์ที่ไม่มีรูปแบบตายตัว (Unstructured Data) ให้เป็นข้อมูลที่คอมพิวเตอร์สามารถนำไปประมวลผลต่อได้ โดยแบ่งกระบวนการย่อยเป็น

1. Natural Language Understanding (NLU): การทำความเข้าใจภาษา ประกอบด้วย

- Intent Classification: การจำแนกเจตนาของผู้ใช้ว่าต้องการสื่อสารเรื่องอะไร

- Entity Extraction: การสกัดคำสำคัญหรือข้อมูลจำเพาะจากประโยค เช่น วันที่, สถานที่, ชื่อสินค้า

2. Natural Language Generation (NLG): การสร้างประโยคตอบกลับให้เป็นภาษาธรรมชาติที่มนุษย์เข้าใจได้

2.2.2.3 ส่วนจัดการบทสนทนาและสถานะ (Dialog Manager) Dialog Manager (DM) ทำหน้าที่เป็นหน่วยควบคุมตรรกะ (Logic) ของการสนทนา เพื่อกำหนดว่าระบบควรตอบโต้ผู้ใช้งานอย่างไร โดยอาศัยข้อมูลที่ได้จาก NLP

1. หน้าที่หลัก

- State Tracking: การติดตามสถานะของการสนทนาว่าปัจจุบันคุยกันขั้นตอนใด

- Context Management: การจดจำริบบทและข้อมูลก่อนหน้า (History) เพื่อให้การสนทนาต่อเนื่อง

2. ประเภทของระบบ: อาจใช้ระบบแบบกฎเกณฑ์ (Rule-based) สำหรับคำนวณง่ายๆ หรือใช้ระบบ Machine Learning สำหรับสนทนาระบบที่มีความซับซ้อน (Contextual Conversation)

2.2.2.4 คลังความรู้และฐานข้อมูล (Knowledge Base/Database) สำหรับเก็บข้อมูลสารสนเทศที่แข็ง固定 ต้องนำมาใช้ในการประมวลผลหรือตอบคำถาม แบ่งเป็น 2 ลักษณะ คือ

1. Knowledge Base (KB): คลังความรู้ที่มักเก็บข้อมูลแบบไม่มีโครงสร้าง หรือกึ่งโครงสร้าง เช่น ไฟล์เอกสาร, คู่มือการใช้งาน, หรือชุดคำถาม-คำตอบ (FAQ) ที่เตรียมไว้

2. Database: ฐานข้อมูลที่มีโครงสร้างชัดเจน (Structured Data) ใช้เก็บข้อมูลธุรกรรม (Transaction) เช่น ข้อมูลสมาชิก, ประวัติการสั่งซื้อ หรือสต็อกสินค้า เพื่อให้ Dialog Manager ดึงไปใช้งานแบบ Real-time

2.2.3 ประโยชน์ของระบบแข็ง固定 ต้องการ การทำงานโดยอัตโนมัติ ใช้ในกระบวนการทางธุรกิจ ก่อให้เกิดประโยชน์ต่อองค์กรในหลายมิติ ทั้งด้านการบริหารจัดการต้นทุน การยกระดับการให้บริการ และการวางแผนกลยุทธ์ โดยสามารถจำแนกประเด็นสำคัญได้ดังนี้

2.2.3.1 การลดต้นทุนและการใช้ทรัพยากรบุคคล (Cost and Resource Optimization) แข็ง固定 ต้องการลดภาระค่าใช้จ่ายในการจ้างพนักงานสำหรับตอบคำถามพื้นฐานช้าๆ (Routine Inquiries) ทำให้องค์กรสามารถบริหารจัดการทรัพยากรบุคคลให้ไปฟอกสักบงานที่มีความซับซ้อนหรืองานเชิงวิเคราะห์ที่ต้องใช้ทักษะมนุษย์เด้มากขึ้น ส่งผลให้ต้นทุนต่อหน่วยในการให้บริการลูกค้า (Cost per Contact) ลดลงในระยะยาว

2.2.3.2 การให้บริการที่ต่อเนื่องตลอด 24 ชั่วโมง (24/7 Availability) ข้อจำกัดเรื่องเวลาทำงานของมนุษย์ไม่ส่งผลกระทบต่อแข็ง固定 ต้องการ ทำให้องค์กรสามารถให้บริการลูกค้าได้ตลอด 24 ชั่วโมง โดยไม่มีวันหยุด ซึ่งเป็นปัจจัยสำคัญในการสร้างความพึงพอใจและตอบโจทย์พฤติกรรมผู้บริโภคในยุคดิจิทัลที่ต้องการความช่วยเหลือแบบทันท่วงที (Real-time)

2.2.3.3 การรองรับผู้ใช้งานจำนวนมากพร้อมกัน (High Scalability) ระบบแข็ง固定 ต้องสามารถรองรับการสนทนา กับผู้ใช้งานจำนวนมากได้ในเวลาเดียวกัน (Concurrent Users) โดยไม่ส่งผลกระทบต่อความเร็วในการตอบกลับ แตกต่างจากพนักงาน (Human Agent) ที่สามารถดูแลลูกค้าได้ทีละ 1-2 รายเท่านั้น ช่วยแก้ปัญหาความชัดในช่วงเวลาที่มีปริมาณการติดต่อสูง (Peak Hours)

2.2.3.4 ความรวดเร็วและความถูกต้องของข้อมูล (Speed and Accuracy) แข็ง固定 ต้องสามารถสืบค้นข้อมูลจากฐานข้อมูลและตอบกลับได้ในเวลาเรียบร้อย (Wait Time) ของลูกค้า นอกจากนี้ ข้อมูลที่ตอบกลับยังมีความถูกต้องและเป็นมาตรฐานเดียวกัน (Standardization) ตามที่โปรแกรมไว้ ช่วยลดความผิดพลาดที่เกิดจากมนุษย์ (Human Error)

2.2.3.5 การเก็บรวบรวมและวิเคราะห์ข้อมูลลูกค้า (Data Collection and Analytics) การสนทนาผ่านแข็ง固定 ต้องเป็นแหล่งข้อมูล Big Data ที่สำคัญ องค์กรสามารถนำประวัติการสนทนา (Chat Logs)

ไปวิเคราะห์เพื่อทำความเข้าใจพฤติกรรม ความสนใจ และปัญหาของลูกค้า (Pain Points) นำไปสู่การปรับปรุงสินค้า บริการ หรือการทำแคมเปญการตลาดแบบเฉพาะเจาะจง (Personalization) ได้อย่างแม่นยำ

2.2.3.6 การสร้างภาพลักษณ์ที่ทันสมัย (Modern Brand Image) การนำเทคโนโลยีปัญญาประดิษฐ์มาใช้ในการบริการลูกค้า ช่วยส่งเสริมภาพลักษณ์ขององค์กรให้ดูทันสมัย มีนวัตกรรม และแสดงถึงความใส่ใจในการพัฒนาช่องทางการสื่อสารเพื่ออำนวยความสะดวกให้แก่ผู้ใช้บริการ

2.3 ทฤษฎีเกี่ยวกับการเรียนรู้ของแชทบอท (Machine Learning for Chatbot)

2.3.1 ทฤษฎีการเรียนรู้ของแชทบอท (Chatbot Learning Theories) การที่แชทบอ�能够เข้าใจและตอบโต้ภาษาบนภาษาเดียวกัน อาศัยทฤษฎีและกระบวนการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นการทำให้คอมพิวเตอร์เรียนรู้จากข้อมูล (Data-driven) โดยมีทฤษฎีและโมเดลที่เกี่ยวข้องหลักๆ

2.3.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นทฤษฎีพื้นฐานที่สุดที่ใช้ในการสร้างแชทบอทประเภท NLU (Natural Language Understanding)

1. หลักการ: การป้อนข้อมูลเข้าไปในระบบประกอบด้วย "ข้อมูลนำเข้า" (Input) และ "คำตอบที่ต้องการ" (Label/Target) เพื่อให้โมเดลเรียนรู้ความสัมพันธ์

2. การประยุกต์ใช้ในแชทบอท

- Intent Classification: สอนให้บทจำแนกเจตนา เช่น ป้อนประโยค "อยากกินข้าว" พร้อมติดป้ายกำกับว่าเป็น Intent "สั่งอาหาร" เมื่อมีข้อมูลมากพอ บทจะแยกแยกประโยคใหม่ๆ

- Entity Recognition: การสอนให้บทระบุคำเฉพาะ เช่น ชื่อคน สถานที่ หรือเวลาจากประโยคตัวอย่าง

2.3.1.2 การเรียนรู้เชิงลึก (Deep Learning) และโครงข่ายประสาทเทียม สำหรับการประมวลผลภาษาที่ซับซ้อน ปัจจุบันนิยมใช้โครงข่ายประสาทเทียม (Neural Networks) ที่จำลองการทำงานของสมองมนุษย์

- Recurrent Neural Networks (RNN) และ LSTM (Long Short-Term Memory): เป็นโมเดลยุคแรกๆ ที่ออกแบบมาเพื่อจัดการข้อมูลแบบลำดับ (Sequence Data) เช่น ประโยคสนทนา โดยมีความสามารถในการ "จำจำ" คำก่อนหน้าเพื่อทำความเข้าใจรูปแบบคำถัดไป

- Transformer Architecture: (ทฤษฎีเบื้องหลัง ChatGPT/Gemini) เป็นสถาปัตยกรรมใหม่ที่ใช้กลไก Self-Attention ช่วยให้โมเดลสามารถเข้าใจความสัมพันธ์ของคำทุกคำในประโยคได้พร้อมกัน ไม่ว่าจะอยู่ห่างกันแค่ไหน ทำให้เข้าใจบริบทได้ลึกซึ้งกว่า RNN

2.3.1.3 การเรียนรู้แบบถ่ายโอน (Transfer Learning) ทฤษฎีนี้ช่วยลดข้อจำกัดเรื่องข้อมูลสอน (Training Data) ที่มีจำนวนน้อย

- หลักการ: การนำโมเดลภาษาขนาดใหญ่ (Pre-trained Language Models) ที่ผ่านการเรียนรู้จากข้อมูลมหาศาลในอินเทอร์เน็ตมาแล้ว (เช่น BERT, GPT) มาทำการปรับจูน (Fine-tuning) ด้วยข้อมูลเฉพาะทางขององค์กร

- ประโยชน์: ช่วยให้สร้างแซทบทอทที่คล้ายได้เร็วขึ้น โดยไม่ต้องเริ่มสอนภาษาตั้งแต่ศูนย์

2.3.1.4 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) มักใช้ในขั้นตอนการปรับปรุงความลื่นไหลของการสนทนากับ AI หรือในแซทบทอทขั้นสูง (Generative AI)

- หลักการ: การเรียนรู้จากการลองผิดลองถูก (Trial and Error) โดยระบบจะได้รับ "รางวัล" (Reward) เมื่อตอบถูกหรือทำให้สนทนาระบุนได้ด้วยดี และได้รับ "บทลงโทษ" (Penalty) เมื่อตอบผิด

- RLHF (Reinforcement Learning from Human Feedback): การให้มนุษย์เข้ามาช่วยให้คะแนนคำตอบของบอท เพื่อจูนให้บทตอบได้ตรงใจและเป็นธรรมชาติที่สุด

2.3.1.5 การวัดผลโมเดลการเรียนรู้ (Model Evaluation) ในเชิงทฤษฎี การเรียนรู้ต้องมีการวัดผลเพื่อยืนยันประสิทธิภาพ ค่าทางสถิติที่นิยมใช้ได้แก่

- Confusion Matrix: ตารางแสดงความถูกต้องและความผิดพลาดในการทำนาย

- Accuracy: ความแม่นยำโดยรวม

- F1-Score: ค่าเฉลี่ย Harmonic ระหว่าง Precision และ Recall ซึ่งใช้วัดประสิทธิภาพได้ดีที่สุดในกรณีที่ข้อมูลแต่ละคลาสไม่เท่ากัน

2.4 การประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP)

การประมวลผลภาษาธรรมชาติ หรือ NLP เป็นสาขานึงของปัญญาประดิษฐ์ (Artificial Intelligence) ที่มุ่งเน้นให้คอมพิวเตอร์สามารถเข้าใจ ตีความ และตอบสนองต่อภาษามนุษย์ได้ สำหรับระบบแซทบทอท กระบวนการ NLP ถือเป็นหัวใจสำคัญในการเปลี่ยนข้อความดิบ (Raw Text) ให้เป็นข้อมูลที่มีความหมาย เพื่อนำไปสู่การประมวลผลคำตอบที่ถูกต้อง โดยมีกระบวนการย่อยดังนี้

2.4.1 Text Preprocessing การเตรียมข้อมูลข้อความ (Text Preprocessing) คือกระบวนการเตรียมข้อมูลดิบให้พร้อมสำหรับการวิเคราะห์ โดยการลดความซ้ำซ้อนและกำจัดข้อมูลขยะ (Noise) เพื่อให้ระบบสามารถประมวลผลได้อย่างมีประสิทธิภาพสูงสุด

2.4.1.1 Tokenization: การตัดคำ Tokenization คือกระบวนการแบ่งข้อความหรือประโยคภาษาๆ ออกเป็นหน่วยคำย่อยที่เล็กที่สุดที่ยังมีความหมาย เรียกว่า "Token" สำหรับภาษาที่มีการเว้นวรรคคำ เช่น เนื้อเพลงภาษาอังกฤษ (Word Segmentation) จะใช้ช่องว่างเป็นตัวกำหนด แต่สำหรับภาษาไทย ซึ่งเป็น

ภาษาที่เขียนติดกัน (Script Continua) กระบวนการนี้จึงมีความซับซ้อนกว่า จำเป็นต้องใช้อัลกอริทึมที่อ้างอิง พจนานุกรม (Dictionary-based) หรือใช้ Machine Learning ใน การวิเคราะห์ขอบเขตของคำ เพื่อให้แยกคำได้อย่างถูกต้อง เช่น แปลงประโยค "ตารางเรียนวันนี้" ให้กลายเป็น ["ตาราง", "เรียน", "วัน", "นี้"]

2.4.1.2 Normalization: การทำความสะอาดข้อความ Normalization คือกระบวนการแปลงข้อความให้อยู่ในรูปแบบมาตรฐานเดียวกัน เพื่อลดความผันแปรของข้อมูล (Data Variance) ซึ่งประกอบด้วย

- การกำจัดอักขระพิเศษ (Special Character Removal): การลบสัญลักษณ์ที่ไม่สื่อความหมาย เช่น Emoji เครื่องหมายวรรคตอนที่ไม่จำเป็น หรือโค้ด HTML
- การจัดการคำย่อและคำสlang: การแปลงภาษาพูดหรือคำย่อให้เป็นคำเต็มที่ระบบเข้าใจ เช่น แปลง "พน." เป็น "พรุ่น" หรือ "มหาลัย" เป็น "มหาวิทยาลัย"
- Stop Word Removal: การตัดคำฟู่เมื่ออยู่ที่พบบ่อยแต่ไม่มีนัยสำคัญต่อความหมายหลัก เช่น "ครับ" "ค่ะ" "ที่" "ซึ่ง" "อัน" ออกไปเพื่อลดขนาดข้อมูลและเพิ่มประสิทธิภาพการประมวลผล

2.4.2 Matching Algorithms หลังจากผ่านกระบวนการเตรียมข้อมูล ระบบจะใช้อัลกอริทึมในการจับคู่ข้อความที่ได้รับ (Input) เข้ากับฐานข้อมูลความรู้ (Knowledge Base) หรือเจตนา (Intent) ที่กำหนดไว้ โดยมีเทคนิคสำคัญดังนี้

2.4.2.1 Keyword Matching: การจับคู่คำหลักแบบตรงตัว เป็นเทคนิคพื้นฐานที่สุด โดยระบบจะตรวจสอบว่าในประโยคที่ผู้ใช้พิมพ์เข้ามา มี "คำหลัก" (Keyword) ที่กำหนดไว้หรือไม่ หากพบคำดังกล่าว ระบบจะดึงคำตอบที่เชื่อมโยงไว้ออกมาทันที วิธีนี้มีความรวดเร็วและแม่นยำสูงในกรณีที่คำไม่ซับซ้อน แต่เมื่อจำกัดคือหากผู้ใช้รูปประโยคอื่นที่ไม่มี Keyword ระบบจะไม่สามารถตอบได้

2.4.2.2 Synonym Expansion: การขยายความหมายด้วยคำพ้อง เพื่อแก้ปัญหาความแข็งที่อของ Keyword Matching ระบบจะมีการใช้พจนานุกรมคำพ้องความหมาย (Thesaurus) เข้ามาช่วย โดยกำหนดกลุ่มคำที่มีความหมายเหมือนกันให้ซึ่งกันและกัน (Intent) เดียวกัน

- ตัวอย่าง: การกำหนดให้คำว่า "อาจารย์", "ครู", "ผู้สอน", "Lecturer" ทั้งหมดมีค่าเท่ากับ Intent: INSTRUCTOR_INFO ทำให้ไม่ว่าผู้ใช้จะพิมพ์คำไหนมา ระบบก็สามารถเข้าใจได้ว่าผู้ใช้ ต้องการสอบถามข้อมูลเกี่ยวกับอาจารย์

2.4.2.3 Fuzzy Matching: การจับคู่คำที่สะกดผิดหรือใกล้เคียง เป็นเทคนิคที่ใช้จัดการกับความผิดพลาดจากการพิมพ์ (Typo) ของผู้ใช้งาน โดยใช้อัลกอริทึมวัดระยะห่างระหว่างคำ (Edit Distance) เช่น Levenshtein Distance ซึ่งคำนวณจำนวนครั้งของการลบ แทรก หรือเปลี่ยนตัวอักษร เพื่อให้คำหนึ่งกล้ายเป็นอีกคำหนึ่ง

- การทำงาน: หากผู้ใช้พิมพ์ว่า "สวัสดี" (พิมพ์ผิด - ตกล ส.สื่อ) ระบบจะคำนวณความคล้ายคลึง และพบว่าใกล้เคียงกับคำว่า "สวัสดี"มากกว่าเกณฑ์ที่กำหนด (Threshold) จึงอนุมานว่าผู้ใช้ต้องการทักทาย

2.4.2.4 Regular Expression (Regex): การจับรูปแบบแพทเทิร์นข้อมูลความเฉพาะทาง Regular Expression คือชุดรหัสรูปแบบอักษรที่ใช้กำหนดแพทเทิร์นการค้นหาข้อมูลที่มีโครงสร้างแน่นอน (Structured Data) ไม่ใช่การจับคู่คำต่อคำ แต่เป็นการจับคู่ตาม "รูปแบบ"

- การประยุกต์ใช้: ใช้ในการดึงข้อมูลสำคัญ (Entity Extraction) ออกจากประโยคสนทนา เช่น

- รหัสนักศึกษา: รูปแบบตัวเลข 10 หลักขึ้นไป
- เบอร์โทรศัพท์: รูปแบบ 0xx-xxx-xxxx
- อีเมล: รูปแบบ text + @ + domain.com
- ประจำชื่อ: วิธีนี้ช่วยให้บทสามารถดึงข้อมูลที่ถูกต้องไปประมวลผลต่อได้ทันทีโดยไม่ต้องพึ่งพา AI ที่ซับซ้อน

2.5 ระบบจัดการฐานข้อมูล (Database Management System)

ระบบจัดการฐานข้อมูล (DBMS) ทำหน้าที่เป็นซอฟต์แวร์ตระกูลที่ช่วยในการจัดเก็บ ค้นหา และจัดการข้อมูลสำหรับแขบทวาระ เพื่อให้สามารถดึงข้อมูลมาตอบโต้กับผู้ใช้งานได้อย่างถูกต้องและรวดเร็ว โดยมีรายละเอียดการออกแบบดังนี้

2.5.1 Relational Database: การใช้ฐานข้อมูลเชิงสัมพันธ์ (MySQL) ฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System - RDBMS) คือรูปแบบการจัดเก็บข้อมูลในลักษณะของ "ตาราง" (Table) ที่มีความสัมพันธ์เชื่อมโยงกันผ่านคีย์ (Key)

2.5.1.1 เหตุผลที่เลือกใช้ MySQL: ในการพัฒนาระบบนี้ เลือกใช้ MySQL เนื่องจากเป็นระบบจัดการฐานข้อมูลแบบ Open Source ที่มีความเสถียรสูง รองรับคำสั่ง SQL (Structured Query Language) ที่เป็นมาตรฐานสากล และมีประสิทธิภาพในการประมวลผลข้อมูลจำนวนมาก (Scalability) ซึ่งเหมาะสมกับการจัดเก็บข้อมูลของแขบทวาระที่มีการเรียกใช้ข้อมูลตลอดเวลา (High Availability)

2.5.1.2 หลักการทำงาน: ข้อมูลจะถูกแยกเก็บเป็นตารางย่อยๆ ตามประเภทของข้อมูล และเชื่อมโยงกันด้วยความสัมพันธ์แบบต่างๆ เช่น One-to-One, One-to-Many เพื่อลดความซ้ำซ้อนและรักษาความถูกต้องของข้อมูล (Data Integrity)

2.5.2 Database Design การออกแบบฐานข้อมูล (Database Design) เป็นขั้นตอนสำคัญในการวางแผนสร้างการจัดเก็บข้อมูลเพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพสูงสุด

2.5.2.1 โครงสร้างตารางที่ใช้ (Tables) การออกแบบ Schema ของฐานข้อมูลประกอบด้วย ตารางหลัก 4 ตาราง ที่จำเป็นต่อการทำ้งานของแพทบอท ดังนี้

1. Table: staff (ข้อมูลบุคลากร): เก็บข้อมูลอาจารย์และบุคลากรของคณะวิศวกรรมศาสตร์

- Columns: id (PK), name_th, name_en, position_th, position_en, department, email, phone, expertise, photo_url, is_active, created_at, updated_at

2. Table: faq (คำถาม-คำตอบ): ตารางหลักสำหรับเก็บคำถามที่พบบ่อยและคำตอบ (Knowledge Base)

- Columns: id (PK), question, answer, keywords, category, department, is_active

3. Table: news (ข่าวสาร): เก็บข่าวสารและกิจกรรมของคณะ มีการอัปเดตอัตโนมัติผ่าน

Web Scraping

- Columns: id (PK), title, summary, category, link_url, published_date, is_active, created_at

4. Table: chat_logs (ประวัติการสนทนา): บันทึกการสนทนาเพื่อนำไปวิเคราะห์และปรับปรุงระบบ

- Columns: id (PK), session_id, user_message, bot_response, sources, confidence, response_time_ms, user_ip, user_agent, created_at

2.5.2.2 Normalization: การจัดระเบียบข้อมูลเพื่อลดความซ้ำซ้อน Normalization คือ กระบวนการออกแบบตารางให้เป็นไปตามกฎมาตรฐาน (Normal Forms) เพื่อลดความซ้ำซ้อนของข้อมูล (Data Redundancy) และป้องกันความผิดปกติในการจัดการข้อมูล (Anomalies)

- การออกแบบในระบบนี้: ระบบใช้หลักการ Normalization ถึงระดับ 3NF (Third Normal Form) โดยแต่ละตารางมี Primary Key ที่ไม่ซ้ำกัน และคอลัมน์ทั้งหมดขึ้นอยู่กับ Primary Key โดยตรง อย่างไรก็ตาม สำหรับข้อมูลสาขาวิชา (department) ระบบเลือกไม่แยกเป็นตารางแยก แต่เก็บเป็น VARCHAR ในตาราง staff และ faq โดยตรง เนื่องจากข้อมูลสาขาวิชามีจำนวนจำกัด (10 สาขา) และไม่มีการเปลี่ยนแปลงบ่อย ซึ่งเป็น Trade-off ระหว่างความเรียบง่ายในการ Query กับการลดความซ้ำซ้อน

2.5.2.3 Indexing: การทำดัชนีเพื่อเพิ่มประสิทธิภาพการค้นหา Indexing คือเทคนิคการสร้างโครงสร้างข้อมูลพิเศษ (เช่น B-Tree) เพื่อเพิ่มความเร็วในการค้นหาข้อมูล

- ความสำคัญต่อแพทบอท: แพทบอทต้องตอบสนองแบบ Real-time ดังนั้นการค้นหาข้อมูลโดยไม่มี Index (Full Table Scan) จะทำให้ระบบช้ามาก

- ประเภท Index ที่ใช้:

1. B-Tree Index: ใช้กับคอลัมน์ที่ค้นหาบ่อย เช่น department, is_active, session_id ช่วยให้การค้นหาเร็วขึ้นจาก $O(n)$ เป็น $O(\log n)$

2. FULLTEXT Index: ใช้กับตาราง faq สำหรับคอลัมน์ question, keywords, answer เพื่อรับการค้นหาข้อความแบบ Natural Language ด้วยคำสั่ง MATCH() AGAINST()

2.5.2.4 Transaction & ACID: คุณสมบัติการรับประกันความถูกต้องของข้อมูล ACID คือ คุณสมบัติ 4 ประการที่รับประกันความถูกต้องและความสมบูรณ์ของข้อมูลเมื่อมีการทำธุรกรรม (Transaction)

- Atomicity (ความเป็นหน่วยเดียว): การทำงานต้องสำเร็จทั้งหมดหรือไม่ทำเลย (All or Nothing) เช่น เมื่อบันทึก chat log พร้อมอัปเดตสถิติ ต้องสำเร็จพร้อมกัน หากส่วนใดส่วนหนึ่งล้มเหลวระบบจะยกเลิก (Rollback) ทั้งหมด

- Consistency (ความถูกต้อง): ข้อมูลหลังการทำธุรกรรมต้องถูกต้องตามกฎที่กำหนด เช่น ค่า confidence ต้องอยู่ระหว่าง 0.00-1.00 หรือ is_active ต้องเป็น 0 หรือ 1 เท่านั้น

- Isolation (ความแยกตัว): เมื่อมีผู้ใช้หลายคนเข้าใช้งานพร้อมกัน การทำธุรกรรมของแต่ละคนต้องไม่กระทบกัน ระบบใช้ระดับ Isolation แบบ READ COMMITTED เพื่อป้องกันการอ่านข้อมูลที่ยังไม่ได้ยืนยัน

- Durability (ความคงทน): เมื่อข้อมูลถูกบันทึกเสร็จสิ้นแล้ว จะถูกเก็บอย่างถาวรแม้ระบบจะไฟดับหรือเกิดข้อผิดพลาด MySQL มีระบบ Transaction Log และ Crash Recovery อัตโนมัติเพื่อรับประกันคุณสมบัตินี้

2.6 สถาปัตยกรรมเว็บแอปพลิเคชัน (Web Application Architecture)

สถาปัตยกรรมเว็บแอปพลิเคชัน คือ พิมพ์เขียวหรือโครงสร้างที่กำหนดรูปแบบการปฏิสัมพันธ์ระหว่างแอปพลิเคชัน (Application), มิดเดิลแวร์ (Middleware) และฐานข้อมูล (Database) เพื่อให้ระบบสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ รองรับการขยายตัว และง่ายต่อการบำรุงรักษา

2.6.1 3-Tier Architecture สถาปัตยกรรมแบบ 3 ชั้น (Three-Tier Architecture) เป็นรูปแบบโครงสร้างพื้นฐานที่นิยมที่สุดในการพัฒนาเว็บแอปพลิเคชัน โดยแบ่งระบบออกเป็น 3 ส่วนอิสระที่ทำงานประสานกัน ได้แก่

2.6.1.1 Presentation Tier (ส่วนแสดงผล)

- หน้าที่เป็นส่วนติดต่อกับผู้ใช้งาน (User Interface - UI) ผ่านเว็บเบราว์เซอร์หรือแอปพลิเคชัน

- รับผิดชอบการรับข้อมูลนำเข้า (Input) จากผู้ใช้ และแสดงผลลัพธ์ (Output)
- เทคโนโลยีที่ใช้มักเป็น HTML, CSS, และ JavaScript (หรือ Framework เช่น React, Vue.js)

2.6.1.2 Application Tier (ส่วนประมวลผล)

- เปรียบเสมือนสมองของระบบ หรือที่เรียกว่า "Logic Tier" ทำหน้าที่ประมวลผลตรรกะทางธุรกิจ (Business Logic)
 - รับคำขอ (Request) จากส่วนแสดงผล มาวิเคราะห์ คำนวณ และตัดสินใจ ก่อนจะส่งคำสั่งไปยังฐานข้อมูล
 - เป็นที่อยู่ของโค้ดโปรแกรมหลัก (Source Code) เช่น ภาษา Python, PHP, Node.js หรือ Java

2.6.1.3 Data Tier (ส่วนข้อมูล)

- ทำหน้าที่จัดเก็บและบริหารจัดการข้อมูล (Database Management)
- รับคำสั่งจาก Application Tier เพื่อทำการเพิ่ม ลบ แก้ไข หรือค้นหาข้อมูล (CRUD Operations) และส่งข้อมูลกลับไป
 - ประกอบด้วยระบบฐานข้อมูล เช่น MySQL, PostgreSQL หรือ MongoDB

2.6.2 MVC Pattern (Model-View-Controller) MVC เป็นรูปแบบสถาปัตยกรรมซอฟต์แวร์ (Software Architectural Pattern) ที่ใช้ในการจัดระเบียบโค้ดภายใน Application Tier เพื่อแยกหน้าที่การทำงานให้ชัดเจน ลดความซับซ้อน และช่วยให้นักพัฒนาหลายคนทำงานร่วมกันได้ง่ายขึ้น ประกอบด้วย 3 ส่วนคือ

2.6.2.1 Model (ส่วนข้อมูลและตรรกะ)

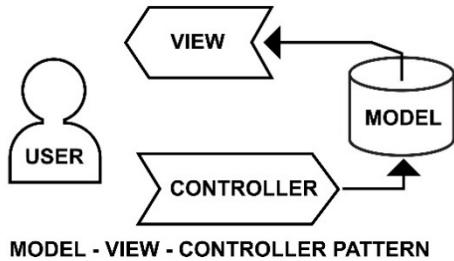
- รับผิดชอบจัดการข้อมูลและกฎทางธุรกิจ (Business Logic) เชื่อมต่อกับฐานข้อมูล โดยตรง
- เมื่อข้อมูลมีการเปลี่ยนแปลง Model จะแจ้งเตือนไปยัง View เพื่ออัปเดตการแสดงผล

2.6.2.2 View (ส่วนแสดงผล)

- รับผิดชอบการแสดงผลข้อมูลที่ได้รับจาก Model ให้ผู้ใช้งานเห็น (User Interface)
- ไม่มีหน้าที่ในการคำนวณหรือประมวลผลตรรกะที่ซับซ้อน

2.6.2.3 Controller (ส่วนควบคุม)

- ทำหน้าที่เป็นตัวกลาง (Intermediary) ระหว่าง Model และ View
- รับ Input จากผู้ใช้ (ผ่าน View) และประมวลผลเพื่อส่งให้ Model อัปเดตข้อมูล หรือส่งให้ View เปลี่ยนหน้าจากการแสดงผล



รูปที่ 2.6.2.3 Model-View-Controller

2.6.3 Client-Server Model Client-Server Model คือรูปแบบสถาปัตยกรรมเครือข่ายที่อธิบายกระบวนการสื่อสารและแบ่งหน้าที่ระหว่างคอมพิวเตอร์ 2 ฝั่ง

2.6.3.1 Client (ผู้ใช้งาน)

- คือเครื่องคอมพิวเตอร์หรืออุปกรณ์ของผู้ใช้งาน (เช่น สมาร์ตโฟนที่เปิดหน้าแดชบอร์ด)
- หน้าที่เป็น "ผู้ร้องขอ" (Requester) โดยส่งคำขอข้อมูล (Request) ผ่านเครือข่ายอินเทอร์เน็ตไปยังเครื่องแม่ข่าย

2.6.3.2 Server (ผู้ให้บริการ)

- คือเครื่องคอมพิวเตอร์ประสิทธิภาพสูงที่เปิดใช้งานตลอดเวลา
- หน้าที่เป็น "ผู้ให้บริการ" (Provider) โดยรับคำขอ (Listening) ประมวลผลตามคำสั่ง และส่งข้อมูลตอบกลับ (Response) ไปยัง Client

2.6.3.3 กระบวนการทำงาน (Request-Response Cycle)

- การสื่อสารระหว่าง Client และ Server มักทำผ่านโปรโตคอล HTTP/HTTPS โดย Client ส่ง HTTP Request และ Server ตอบกลับด้วย HTTP Response (พร้อม Status Code เช่น 200 OK, 404 Not Found)

2.7 การเขียนต่อผ่าน API (RESTful Services)

การเขียนต่อระหว่างส่วนหน้าบ้าน (Frontend) และหลังบ้าน (Backend) ในระบบซอฟต์แวร์ นิยมใช้รูปแบบ API (Application Programming Interface) เพื่อเป็นสะพานเชื่อมโยงข้อมูล โดยสถาปัตยกรรมที่เป็นมาตรฐานหลักคือ RESTful Services ซึ่งมีความยืดหยุ่นสูงและเป็นสากล

2.7.1 REST Principles: หลักการออกแบบ API แบบ Stateless REST (Representational State Transfer) เป็นรูปแบบสถาปัตยกรรมซอฟต์แวร์สำหรับการส่งผ่านข้อมูลบนเครือข่ายอินเทอร์เน็ต โดยมีหัวใจสำคัญคือหลักการ Statelessness (ความไร้สถานะ)

2.7.1.1 ความหมายของ Stateless: ในการสื่อสารแต่ละครั้ง (Request) จาก Client ไปยัง Server จะต้องบรรจุข้อมูลที่จำเป็นทั้งหมดสำหรับการประมวลผลไปด้วยเสมอ Server จะไม่ทำการบันทึกสถานะการเชื่อมต่อ (Session) ของผู้ใช้งานที่ผ่านมา

2.7.1.2 ประโยชน์: ทำให้ Server ไม่ต้องแบกรับภาระหน่วยความจำในการจัดจำนำใช้งานทุกคน ช่วยให้ระบบสามารถรองรับผู้ใช้งานจำนวนมากได้ (Scalability) และง่ายต่อการขยายระบบ

2.7.2 HTTP Methods: การใช้คำสั่ง GET, POST, PUT, DELETE ในการสั่งการให้ API ทำงาน RESTful API จะใช้ "คำกริยา" (Verbs) ตามมาตรฐาน HTTP Protocol เพื่อระบุการกระทำที่ต้องการ (Action) ต่อทรัพยากรข้อมูล ดังนี้

2.7.2.1 GET (ดึงข้อมูล): ใช้สำหรับร้องขอข้อมูลจาก Server โดยไม่มีการเปลี่ยนแปลงแก้ไขข้อมูลใดๆ เช่น การดึงประวัติการสนทนาระหว่างผู้ใช้งาน

2.7.2.2 POST (ส่งข้อมูล/สร้างใหม่): ใช้สำหรับส่งข้อมูลใหม่ไปบันทึกที่ Server เช่น การส่งข้อความแรกที่ผู้ใช้เพิ่งพิมพ์ (Input Message) ไปประมวลผล

2.7.2.3 PUT (แก้ไข/แทนที่): ใช้สำหรับการอัปเดตข้อมูลที่มีอยู่แล้วให้เป็นค่าใหม่ เช่น การแก้ไขข้อมูลส่วนตัวสมาชิก หรือการเปลี่ยนสถานะงาน

2.7.2.4 DELETE (ลบ): ใช้สำหรับลบข้อมูลออกจากฐานข้อมูล เช่น การลบบัญชีผู้ใช้ หรือลบรายการสินค้า

2.7.3 JSON Format: รูปแบบข้อมูลมาตรฐานที่ใช้รับส่ง JSON (JavaScript Object Notation) คือรูปแบบมาตรฐานในการจัดเก็บและแลกเปลี่ยนข้อมูลคอมพิวเตอร์ ซึ่งเข้ามาแทนที่ XML เนื่องจากมีโครงสร้างที่กระชับและประมวลผลได้เร็วกว่า

2.7.3.1 โครงสร้าง: จัดเก็บข้อมูลในรูปแบบคู่ของ Key: Value (Key-Value Pair) ภายใต้วงล้อ { } ซึ่งมนุษย์สามารถอ่านเข้าใจได้ง่าย (Human-readable)

2.7.4 Security: การรักษาความปลอดภัย เนื่องจาก API เป็นช่องทางเข้าออกของข้อมูลสำคัญ ระบบจึงต้องมีมาตรการรักษาความปลอดภัยที่เข้มงวด

2.7.4.1 HTTPS (Hypertext Transfer Protocol Secure): การเข้ารหัสข้อมูลในระหว่างการส่งผ่านเครือข่าย (Data in Transit) ด้วยโปรโตคอล SSL/TLS เพื่อป้องกันการตักจับหรือแอบอ่านข้อมูล (Eavesdropping) โดยผู้ไม่หวังดี

2.7.4.2 API Key / Authentication: การระบุตัวตนผู้ใช้งาน API โดยผู้ใช้หรือแอปพลิเคชันต้องแนบรหัสผ่านพิเศษ (API Key หรือ Token) มา กับทุก Request เพื่อให้ Server ตรวจสอบสิทธิ์ว่าได้รับอนุญาตให้เข้าถึงข้อมูลหรือไม่ และยังช่วยป้องกันการโจมตีแบบ Spam Request ได้อีกด้วย

2.8 การออกแบบส่วนติดต่อผู้ใช้ (UI/UX Design)

การออกแบบส่วนติดต่อผู้ใช้ (User Interface - UI) และประสบการณ์ผู้ใช้ (User Experience - UX) เป็นปัจจัยสำคัญที่จะกำหนดความสำเร็จของระบบซอฟต์แวร์ การออกแบบที่ดีต้องคำนึงถึงความสะดวกในการใช้งาน การเข้าถึงข้อมูลที่ง่าย และความรู้สึกพึงพอใจของผู้ใช้งาน โดยมีหลักการสำคัญดังนี้

2.8.1 Usability Principles: ความสม่ำเสมอ (Consistency) และความเรียบง่าย (Simplicity) หลักการ "Usability" หรือความสามารถในการใช้งาน มุ่งเน้นการลดภาระทางความคิดของผู้ใช้ (Cognitive Load) เพื่อให้ผู้ใช้สามารถบรรลุเป้าหมายได้อย่างรวดเร็ว

2.8.1.1 ความสม่ำเสมอ (Consistency): การออกแบบต้องเป็นไปในทิศทางเดียวกันทั่วระบบ

- Visual Consistency: การใช้โทนสี รูปแบบตัวอักษร และไอคอน ให้เหมือนกันทุกหน้า เพื่อสร้างความจำจำ (Brand Identity)

- Functional Consistency: ตำแหน่งของปุ่มสั่งการหรือเมนูต้องวางอยู่ในตำแหน่งเดิม ไม่ย้ายไปมา เพื่อให้ผู้ใช้เกิดความคุ้นเคยและเรียนรู้การใช้งานได้เร็ว (Learnability)

2.8.1.2 ความเรียบง่าย (Simplicity): การตัดสิ่งที่ไม่จำเป็นออกตามหลักการ "Less is More"

- ลดจำนวนขั้นตอนการคลิก (Clicks) ให้เหลือเท่าที่จำเป็น

- ใช้พื้นที่ว่าง (White Space) เพื่อให้อ่านง่ายและสบายตา

- หลีกเลี่ยงการแสดงข้อมูลที่หนาแน่นเกินไปในหน้าจอส่วนหน้า เพื่อให้ผู้ใช้โฟกัสกับเนื้อหาหลักได้ทันที

2.8.2 Responsive Design: การแสดงผลที่รองรับทุกหน้าจอ (Mobile-first) โดยเฉพาะบนมือถือ ในยุคปัจจุบัน ผู้ใช้งานส่วนใหญ่นิยมเข้าถึงอินเทอร์เน็ตผ่านสมาร์ตโฟน การออกแบบจึงต้องยึดหลัก Mobile-First Strategy คือการให้ความสำคัญกับการออกแบบบนหน้าจอมือถือเป็นลำดับแรก ก่อนจะขยายไปยังหน้าจอขนาดใหญ่

2.8.2.1 Liquid Layout & Flexible Grids: การใช้โครงสร้างหน้าเว็บที่ยืดหยุ่น สามารถปรับขนาดและความกว้างขององค์ประกอบต่างๆ (เช่น กล่องข้อความ รูปภาพ ปุ่มกด) ให้พอดีกับความกว้างของหน้าจออุปกรณ์ (Viewport) โดยอัตโนมัติ

2.8.2.2 Touch-Friendly Interface: การออกแบบจุดสัมผัส (Touch Targets) เช่น ปุ่มเมนู หรือตัวเลือก Quick Reply ให้มีขนาดใหญ่พอที่นิ้วมือจะสัมผัสได้แม่นยำ (โดยทั่วไปควรมีขนาดไม่ต่ำกว่า 44x44 pixels) และมีการเว้นระยะห่างเพื่อป้องกันการกดผิด

2.8.2.3 Readable Content: ขนาดตัวอักษรต้องอ่านง่ายโดยไม่ต้องซูมหน้าจอ และจัดลำดับความสำคัญของเนื้อหาให้เหมาะสมกับพื้นที่จำกัด

2.8.3 Conversational Design: หลักการออกแบบบทสนทนาให้ดูเป็นธรรมชาติ ไม่เหมือนหุ่นยนต์ จนเกินไป Conversational Design คือศาสตร์ของการออกแบบ "บทพูด" ให้กับระบบ เพื่อให้ผู้ใช้งานรู้สึกเหมือนกำลังสนทนากับมนุษย์ (Human-like Interaction) มากกว่าการป้อนคำสั่งคอมพิวเตอร์

2.8.3.1 Persona Development: การกำหนดบุคลิกภาพของแพทบทอทให้ชัดเจน เช่น การใช้สรพนาม (ผม/ดิฉัน/น้อง) น้ำเสียง (ทางการ/เป็นกันเอง) เพื่อสร้างอารมณ์ร่วมและความน่าเชื่อถือ

2.8.3.2 Cooperative Principle: การออกแบบบทสนทนาตามหลักความร่วมมือ โดยอบทต้องให้ข้อมูลที่ "กระชับ ตรงประเด็น และชัดเจน" ไม่ตอบยาวเหยียดจนผู้ใช้จับใจความไม่ได้

2.8.3.3 Turn-taking & Feedback: ระบบต้องมีการโต้ตอบที่แสดงสถานะชัดเจน เช่น การแสดงสัญลักษณ์ "กำลังพิมพ์..." (Typing Indicator) เพื่อให้ผู้ใช้รู้ว่าระบบกำลังประมวลผลอยู่ และไม่รู้สึกว่าถูกทิ้งไว้

2.8.3.4 Error Handling: เมื่อบอทไม่เข้าใจคำสั่ง ต้องมีการออกแบบประโยชน์ตอบรับที่สุภาพ และแนะนำทางเลือก (Fallback Message) เช่น "ขอโทษครับ ผมไม่แน่ใจว่าเข้าใจถูกไหม คุณหมายถึง... ใช่หรือไม่?" แทนที่จะตอบแค่ว่า "Error" หรือไม่ตอบเลย

บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

บทที่ 4

ผลการดำเนินงาน

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

បររណោនុករម

ประวัติผู้จัดทำ

ชื่อ-สกุล

นายกรภัทร เจริญสุข

วัน/เดือน/ปีเกิด

29 ตุลาคม 2546

ที่อยู่ปัจจุบัน

21 ซอยงามวงศ์วาน 18 ถนนงามวงศ์วาน ตำบลบางเขน อำเภอเมือง
จังหวัดนนทบุรี 11000

เบอร์โทรศัพท์

092-4598535

ประวัติการศึกษา

มัธยมศึกษาตอนต้น (ม.ต้น)

โรงเรียนเบญจมราชนุสรณ์ พ.ศ.2559-2561

มัธยมศึกษาตอนปลาย (ม.ปลาย) แผนการเรียน คณิตศาสตร์คอมพิวเตอร์

โรงเรียนเบญจมราชนุสรณ์ พ.ศ.2562-2564

ระดับปริญญาตรี (ป.ตรี) วศ.บ.(วิศวกรรมคอมพิวเตอร์)

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร พ.ศ.2565-2568

ประวัติผู้จัดทำ

ชื่อ-สกุล

นายกิตติพงษ์ คานะเงน้อย

วัน/เดือน/ปีเกิด

ที่อยู่ปัจจุบัน

เบอร์โทรศัพท์

ประวัติการศึกษา

มัธยมศึกษาตอนต้น (ม.ต้น)

โรงเรียน พ.ศ.2559-2561

มัธยมศึกษาตอนปลาย (ม.ปลาย)

โรงเรียน พ.ศ.2562-2564

ระดับปริญญาตรี (ป.ตรี) วศ.บ.(วิศวกรรมคอมพิวเตอร์)

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร พ.ศ.2565-2568

ประวัติผู้จัดทำ

ชื่อ-สกุล

นายอุรุวัฒน์ กลินโสมณ

วัน/เดือน/ปีเกิด

ที่อยู่ปัจจุบัน

เบอร์โทรศัพท์

ประวัติการศึกษา

มัธยมศึกษาตอนต้น (ม.ต้น)

โรงเรียน พ.ศ.2559-2561

มัธยมศึกษาตอนปลาย (ม.ปลาย) แผนการเรียน

โรงเรียน พ.ศ.2562-2564

ระดับปริญญาตรี (ป.ตรี) วศ.บ.(วิศวกรรมคอมพิวเตอร์)

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร พ.ศ.2565-2568

ประวัติผู้จัดทำ

ชื่อ-สกุล

นายไชยพัฒน์ สมวงศ์

วัน/เดือน/ปีเกิด

ที่อยู่ปัจจุบัน

เบอร์โทรศัพท์

ประวัติการศึกษา

มัธยมศึกษาตอนต้น (ม.ต้น)

โรงเรียนเบญจมราชนุสรณ์ พ.ศ.2559-2561

มัธยมศึกษาตอนปลาย (ม.ปลาย) แผนการเรียน คณิตศาสตร์คอมพิวเตอร์

โรงเรียนเบญจมราชนุสรณ์ พ.ศ.2562-2564

ระดับปริญญาตรี (ป.ตรี) วศ.บ.(วิศวกรรมคอมพิวเตอร์)

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร พ.ศ.2565-2568