

ระบบจับคู่อุปกรณ์เกมมิ่ง

(Gaming Gear Matcher)

นายชาคริต พิมพ์สระเกษ

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์

ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ : ระบบจับคู่อุปกรณ์เกมมิ่ง
Gaming Gear Matcher

โดย : นายชาคริต พิมพ์สระเกษ

อาจารย์ที่ปรึกษา : อาจารย์วายุ ปุยะติ

ระดับการศึกษา : วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์

ปีการศึกษา : 2568

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์
คณะกรรมการสอบประเมินความรู้โครงการของคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา
(วายุ ปุยะติ)

..... กรรมการ
(ชื่อกรรมการ)

..... กรรมการ
(ชื่อกรรมการ)

..... หัวหน้าภาควิชา
(รศ.ชาญชัย ศุภอรรถกร)

วันที่/...../.....

กิตติกรรมประกาศ

โครงการระบบจับคู่อุปกรณ์เกมมิ่ง (Gaming Gear Matcher) นี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับการสนับสนุนและคำแนะนำอย่างดียิ่งจาก อาจารย์วาโย ปุยะติ อาจารย์ที่ปรึกษาโครงการ ที่กรุณาให้คำปรึกษาแนะนำแนวทางในการแก้ไขปัญหา และตรวจสอบความถูกต้องของเนื้อหาตลอดระยะเวลาการดำเนินงาน

ขอขอบคุณคณาจารย์ประจำสาขาวิชาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ที่ได้ประสิทธิ์ประสาทวิชาความรู้ ซึ่งเป็นพื้นฐานสำคัญในการพัฒนาโครงการนี้ ขอขอบคุณเพื่อนร่วมรุ่นที่คอยให้กำลังใจและแลกเปลี่ยนความรู้ทำให้โครงการนี้สำเร็จด้วยดี

นายชาคริต พิมพ์สระเกษ

12 กุมภาพันธ์ 2569

โครงการงาน	:	ระบบจับคู่อุปกรณ์เกมมิ่ง Gaming Gear Matcher
โดย	:	นายชาคริต พิมพ์สระเกษ
อาจารย์ที่ปรึกษา	:	อาจารย์วาโย ปุยะติ
ระดับการศึกษา	:	วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการข้อมูลและนวัตกรรม ซอฟต์แวร์
ปีการศึกษา	:	2568

บทคัดย่อ

ในปัจจุบัน อุปกรณ์เกมมิ่ง (Gaming Gear) มีความหลากหลายทั้งในด้านคุณสมบัติและราคา ทำให้ผู้ใช้งานประสบปัญหาในการเลือกซื้ออุปกรณ์ให้เหมาะสมกับสรีระและรูปแบบการเล่นของตนเอง โครงการนี้จึงมีวัตถุประสงค์เพื่อพัฒนาเว็บแอปพลิเคชันระบบจับคู่อุปกรณ์เกมมิ่ง (Gaming Gear Matcher) เพื่อช่วยแนะนำชุดอุปกรณ์ที่เหมาะสมที่สุดให้แก่ผู้ใช้งาน

ระบบถูกพัฒนาขึ้นในรูปแบบเว็บแอปพลิเคชัน โดยใช้ภาษา Python ร่วมกับ Django Web Framework ในส่วนของ Back-end และใช้ HTML5, CSS3, JavaScript ร่วมกับ Bootstrap 5 ในส่วนของ Front-end ระบบฐานข้อมูลใช้ PostgreSQL โดยมีฟีเจอร์หลักคือ (1) ระบบแนะนำแบบผสมผสาน (Hybrid Recommender) ที่วิเคราะห์รูปแบบการเล่นผ่านการตอบคำถาม (Quiz) และนำมาจัดเซตอุปกรณ์เบื้องต้นให้แก่ผู้ใช้งาน ร่วมกับสเปคอุปกรณ์และคะแนนรีวิว (Sentiment Score) (2) ระบบกฎความสัมพันธ์ (Association Rules) สำหรับแนะนำอุปกรณ์ที่สัมพันธ์กันเมื่อผู้ใช้เลือกอุปกรณ์ด้วยตนเอง (3) ระบบเปรียบเทียบชุดอุปกรณ์ (AI Multi-Preset Variants) และ (4) ระบบจัดการชุดอุปกรณ์ (Preset Management) ที่รองรับการแชร์ผ่านลิงก์

ผลการทดสอบระบบพบว่า เว็บแอปพลิเคชันสามารถแนะนำอุปกรณ์ได้สอดคล้องกับเงื่อนไขของผู้ใช้งาน และสามารถรองรับกระบวนการใช้งานตั้งแต่การค้นหา การจัดชุดอุปกรณ์ ไปจนถึงการแชร์ข้อมูลได้อย่างมีประสิทธิภาพ

คำสำคัญ: เว็บแอปพลิเคชัน, ระบบแนะนำ, อุปกรณ์เกมมิ่ง, ระบบแนะนำแบบผสมผสาน, ระบบเปรียบเทียบ

Topic : Gaming Gear Matcher
Author : Chakrit Phimsaket
Advisor : Wayo Puyati
Degree : Bachelor of Science (Data Science and Software Innovation)
Academic Year : 2025

Abstract

Currently, the variety of gaming gears in terms of specifications and prices creates a "Paradox of Choice" for users, making it difficult to select equipment that fits their ergonomics and gaming styles. This project aims to develop a "Gaming Gear Matcher" web application to recommend the most suitable gaming gear sets for users

The system is developed as a web application using Python with the Django Web Framework for the back-end, and HTML5, CSS3, JavaScript with Bootstrap 5 for the front-end. PostgreSQL is used for the database. The key features include: (1) A Hybrid Recommender System that analyzes user play style (via Quiz) alongside hardware specifications and sentiment scores; (2) An Association Rules engine (based on the Apriori algorithm) to suggest related items during manual selection; (3) AI Multi-Preset Variants for comparison; and (4) A Preset Management system that supports public sharing via links.

Keywords: Web Application, Recommender System, Gaming Gear, Hybrid Recommender, Variants Comparison

สารบัญ

กิตติกรรมประกาศ	3
บทคัดย่อ	4
Abstract	5
สารบัญ	6
สารบัญภาพ	10
สารบัญตาราง	12
บทที่ 1 บทนำ	13
1.1 ที่มาและเหตุผล	13
1.2 วัตถุประสงค์	13
1.3 ขอบเขตของโครงการ	13
1.4 ประโยชน์ที่ได้รับ	14
1.5 เครื่องมือที่ใช้ในการพัฒนา	14
1.5.1 ฮาร์ดแวร์	14
1.5.2 ซอฟต์แวร์	14
1.6 แผนการดำเนินการ	15
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	16
2.1 ทฤษฎีและหลักการด้านอุปกรณ์เกมมิ่งและสรีรศาสตร์ (Gaming Gear & Ergonomics)	16
2.1.1 หลักสรีรศาสตร์ของเมาส์เกมมิ่ง (Mouse Ergonomics & Grip Styles)	16
2.2 ทฤษฎีระบบแนะนำ (Recommender System Theory)	18
2.2.1 การกรองเนื้อหา (Content-Based Filtering)	18
2.2.2 การแนะนำโดยอิงกฎเกณฑ์และความรู้ (Knowledge-Based / Rule-Based Recommendation)	19
2.2.3 ระบบแนะนำแบบผสมผสาน (Hybrid Recommender System)	19
2.3 ทฤษฎีการหาความสัมพันธ์ (Association Rule Mining)	19
2.4 ทฤษฎีการออกแบบฐานข้อมูลสำหรับข้อมูลกึ่งโครงสร้าง (Semi-Structured Database Design)	20

	7
2.4.1 แนวคิด Entity-Attribute-Value (EAV)	20
2.4.2 โมเดลข้อมูลแบบ JSON (JSON Data Model)	21
2.5 ภาษาและเทคโนโลยีที่ใช้ในการพัฒนา (Languages & Technologies)	21
2.5.1 ภาษาไพทอน (Python)	21
2.5.2 Django Framework	22
2.5.3 ฐานข้อมูล SQLite และ PostgreSQL	22
2.5.4 Bootstrap 5	23
2.6 เครื่องมือที่ใช้ในการพัฒนา (Development Tools)	24
2.6.1 Visual Studio Code (VS Code)	24
2.6.2 Git และ GitHub	24
2.7 งานวิจัยที่เกี่ยวข้อง (Related Literature Review)	26
2.7.1 งานวิจัยที่ 1: การพัฒนาระบบแนะนำสินค้าด้วยเทคนิค Hybrid Filtering สำหรับ E-Commerce	26
2.7.2 งานวิจัยที่ 2: PC Configuration Recommender System based on Knowledge Rules	26
2.7.3 งานวิจัยที่ 3: Application of Association Rule Mining in Gaming Industry	27
2.7.4 งานวิจัยที่ 4: The Impact of Mouse Weight and Shape on E-Sports Aiming Performance	27
2.7.5 งานวิจัยที่ 5: Managing Heterogeneous E-commerce Data using JSON Document Store	27
2.8 เปรียบเทียบระบบ Gaming Gear Matcher กับงานวิจัยและระบบอื่น (Comparative Analysis)	30
บทที่ 3 การออกแบบระบบ	33
3.1 System architecture	33
3.2 System requirement	34
3.3 การออกแบบหน้าจอ	36
3.4 Use case diagram	41
3.5 Class diagram	60
3.6 Sequence diagram	64

กลุ่มที่ 1: ผู้ใช้งานทั่วไป (Guest & General Features)	66
กลุ่มที่ 2: สมาชิกในระบบ (Member Features)	75
กลุ่มที่ 3: ผู้ดูแลระบบ (Admin Features)	80
3.7 Data model	85
บทที่ 4 การพัฒนาระบบ	89
4.1 โครงสร้างแอปพลิเคชันและการตั้งค่า (Project Configuration)	89
4.2 โครงสร้างไฟล์และโฟลเดอร์ของระบบ (Django Directory Structure)	90
4.3 การพัฒนาฟังก์ชันการทำงานด้วย Django (Core Implementation)	92
บทที่ 5 การทดสอบระบบ	97
5.1 การทดสอบหน่วยย่อย (Unit Testing)	97
5.1.1 การทดสอบระบบแนะนำ (Hybrid Recommender Engine)	97
5.1.2 การทดสอบการจัดการสิทธิ์และ Middleware	97
5.2 การทดสอบการทำงานร่วมกัน (Integration Testing)	98
5.2.1 Scenario: New Member Registration & Matching Flow	98
5.2.2 Scenario: Admin Management Flow	98
5.3 กรณีทดสอบระบบ (System Test Cases)	98
ส่วนที่ 1: ระบบสมาชิกและการเข้าถึง (Authentication & Account)	99
ส่วนที่ 2: ระบบค้นหาและข้อมูล (Search & Information)	100
ส่วนที่ 3: ระบบแนะนำอุปกรณ์ (Recommendation Wizard)	100
ส่วนที่ 4: ระบบจัดการ Preset (Preset Management)	101
ส่วนที่ 5: ระบบผู้ดูแลระบบ (Admin Testing)	102
5.4 การทดสอบสมรรถนะและความปลอดภัย (Performance & Security)	103
5.4.1 ประสิทธิภาพ (Performance)	103
5.4.2 ความปลอดภัย (Security)	104
5.5 สรุปผลการทดสอบ	104
บทที่ 6 สรุปและข้อเสนอแนะ	105
บรรณานุกรม	108

ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม	112
ภาคผนวก ข คู่มือการติดตั้งระบบ	116
ภาคผนวก ค คู่มือการใช้งานของระบบ	118

สารบัญภาพ

รูปภาพที่ 1 แสดงท่าจับเมาส์แบบต่างๆ	17
รูปภาพที่ 2 แสดงสมการ Cosine Similarity	18
รูปภาพที่ 3 แสดงสมการ Support	19
รูปภาพที่ 4 แสดงสมการ Confidence	20
รูปภาพที่ 5 แสดงสมการ Lift	20
รูปภาพที่ 6 แสดงโลโก้ Python	21
รูปภาพที่ 7 แสดงโลโก้ Django	22
รูปภาพที่ 8 แสดงโลโก้ SQLite	22
รูปภาพที่ 9 แสดงโลโก้ PostgreSQL	23
รูปภาพที่ 10 แสดงโลโก้ Bootstrap 5	23
รูปภาพที่ 11 แสดงโลโก้ VS Code	24
รูปภาพที่ 12 แสดงโลโก้ Git	24
รูปภาพที่ 13 แสดงโลโก้ Github	25
รูปภาพที่ 14 แสดงโลโก้ Docker	25
รูปภาพที่ 15 แสดง System Architecture	33
รูปภาพที่ 16 แสดงการออกแบบหน้าแรก	36
รูปภาพที่ 17 แสดงการออกแบบหน้าจอ Quiz	37
รูปภาพที่ 18 แสดงการออกแบบหน้าผลลัพธ์การแนะนำ	38
รูปภาพที่ 19 แสดงการออกแบบหน้ารายละเอียดอุปกรณ์	39
รูปภาพที่ 20 แสดงการออกแบบหน้าจอการจัดการ setup	40
รูปภาพที่ 21 แสดงการออกแบบหน้าจอ Dashboard สำหรับ Admin	41
รูปภาพที่ 22 แสดง Use Case Diagram ของระบบ	44
รูปภาพที่ 23 แสดง Class Diagram ส่วน Backend	62
รูปภาพที่ 24 แสดงกระบวนการเข้าถึงหน้าแรกของเว็บไซต์	66
รูปภาพที่ 25 แสดงกระบวนการค้นหาข้อมูลแบบรวมศูนย์	67
รูปภาพที่ 26 แสดงกระบวนการเรียกดูรายละเอียดของอุปกรณ์เกมมิ่ง	68
รูปภาพที่ 27 แสดงกระบวนการเรียกดูข้อมูลโปรเพลเยอร์และอุปกรณ์ที่ใช้งาน	69
รูปภาพที่ 28 แสดงกระบวนการลงทะเบียนสมาชิกใหม่	70
รูปภาพที่ 29 แสดงกระบวนการเข้าสู่ระบบ	71
รูปภาพที่ 30 แสดงการประมวลผลแบบสอบถามและระบบแนะนำแบบ Hybrid	72
รูปภาพที่ 31 แสดงกระบวนการเลือกอุปกรณ์รายการหมวดหมู่ (Manual Selection)	73
รูปภาพที่ 32 แสดงการจัดการรายการอุปกรณ์ชั่วคราว	74
รูปภาพที่ 33 แสดงการแสดงผลลัพธ์การจับคู่และการแนะนำ	75

รูปภาพที่ 34 แสดงการทำงานของหน้าโปรไฟล์	76
รูปภาพที่ 35 แสดงกระบวนการดูรายการและสร้าง Preset ใหม่	77
รูปภาพที่ 36 แสดงกระบวนการแก้ไขข้อมูล Preset	78
รูปภาพที่ 37 แสดงกระบวนการแชร์ Preset	79
รูปภาพที่ 38 แสดงการทำงานของหน้า Dashboard\	80
รูปภาพที่ 39 แสดงการจัดการสมาชิก	81
รูปภาพที่ 40 แสดงการจัดการคลังอุปกรณ์เกมมิ่ง	82
รูปภาพที่ 41 แสดงการจัดการข้อมูลโปรเพลเยอร์	83
รูปภาพที่ 42 แสดง ER Diagram	86
รูปภาพที่ 43 แสดงการติดตั้ง Python #1 (ก.1)	112
รูปภาพที่ 44 แสดงการติดตั้ง Python #2 (ก.2)	113
รูปภาพที่ 45 แสดงการติดตั้ง VS CODE #1 (ก.3)	113
รูปภาพที่ 46 แสดงการติดตั้ง VS CODE #2 (ก.4)	114
รูปภาพที่ 47 แสดงการติดตั้ง Git #1 (ก.5)	115
รูปภาพที่ 48 แสดงการติดตั้ง Git #2 (ก.6)	115
รูปภาพที่ 49 แสดงการ clone git (ข.1)	116
รูปภาพที่ 50 แสดงการสร้างสภาพแวดล้อม (ข.2)	116
รูปภาพที่ 51 แสดงการเปิดใช้งานสภาพแวดล้อม (ข.3)	116
รูปภาพที่ 52 แสดงการติดตั้ง requirements (ข.4)	117
รูปภาพที่ 53 แสดงตัวอย่างเว็บไซต์ (ข.5)	117
รูปภาพที่ 54 แสดงหน้าแรกของเว็บไซต์ (ค.1)	118
รูปภาพที่ 55 แสดงผลลัพธ์การค้นหา (ค.2)	119
รูปภาพที่ 56 แสดงหน้าจอการสมัครสมาชิก (ค.3)	119
รูปภาพที่ 57 แสดงหน้าจอการเข้าสู่ระบบ (ค.4)	120
รูปภาพที่ 58 แสดงหน้าจอการใช้ AI Analysis (ค.5)	121
รูปภาพที่ 59 แสดงผลลัพธ์การแนะนำ (ค.6)	121
รูปภาพที่ 60 แสดงหน้าจอสเปคของอุปกรณ์ (ค.7)	122
รูปภาพที่ 61 แสดง Preset ที่เลือกไว้ (ค.7)	122
รูปภาพที่ 62 แสดงหน้าจอสำหรับจัดการ Preset (ค.9)	123
รูปภาพที่ 63 แสดงหน้าแก้ไขโปรไฟล์ (ค.10)	123
รูปภาพที่ 64 แสดง Admin Dashboard (ค.11)	124
รูปภาพที่ 65 แสดงหน้าจัดการ User (ค.12)	125
รูปภาพที่ 66 แสดงหน้าจอจัดการ Pro Player (ค.13)	125
รูปภาพที่ 67 แสดงหน้าจอจัดการ Gaming Gear (ค.14)	126

สารบัญตาราง

ตารางที่ 1 แสดงการเปรียบเทียบ Gaming Gear Matcher กับแพลตฟอร์มและงานวิจัยที่เกี่ยวข้อง	32
ตารางที่ 2 แสดงคำอธิบายแผนภาพ Use Case Diagram	42
ตารางที่ 3 แสดงคำอธิบายแผนภาพ Class Diagram	61
ตารางที่ 4 คำอธิบายแผนภาพ Sequence Diagram	65
ตารางที่ 5 คำอธิบายภาพ ER Diagram แบบ Crow's Foot Notation	86

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

ในปัจจุบันอุตสาหกรรมเกมและอีสปอร์ต (E-sports) มีการเติบโตอย่างรวดเร็ว ส่งผลให้ตลาดอุปกรณ์เกมมิ่ง (Gaming Gear) ขยายตัวอย่างมาก ผู้ผลิตต่างแข่งขันกันออกผลิตภัณฑ์ที่มีคุณสมบัติทางเทคนิคซับซ้อน เช่น ค่า DPI ของเมาส์, ประเภทสวิตช์ของคีย์บอร์ด หรือค่า Refresh Rate ของจอภาพ นอกจากนี้ การเลือกอุปกรณ์เกมมิ่ง ที่เหมาะสมกับรูปแบบการเล่นของแต่ละบุคคลเป็นปัจจัยสำคัญที่ส่งผลต่อประสิทธิภาพในเกม

ความหลากหลายนี้ก่อให้เกิดปัญหาต่อผู้ใช้งานทั่วไปและเกมเมอร์มือใหม่ที่ไม่สามารถตัดสินใจเลือกซื้ออุปกรณ์ที่เหมาะสมกับสรีระร่างกาย (เช่น ขนาดมือ, สไตส์การจับ) และแนวเกมที่ตนเองถนัดได้ เนื่องจากมีตัวเลือกในตลาดเป็นจำนวนมาก นอกจากนี้ ข้อมูลรีวิวหรือการตั้งค่าของนักแข่งระดับมืออาชีพ มักกระจัดกระจายอยู่ตามแหล่งข้อมูลต่างๆ ทำให้ผู้ใช้ต้องเสียเวลาสืบค้นและเปรียบเทียบกับตนเอง

จากปัญหาข้างต้น ผู้จัดทำจึงมีแนวคิดพัฒนาเว็บแอปพลิเคชันระบบจับคู่อุปกรณ์เกมมิ่ง (Gaming Gear Matcher) เพื่อเป็นศูนย์กลางในการรวบรวมข้อมูลและให้บริการระบบแนะนำ ที่สามารถวิเคราะห์ความต้องการของผู้ใช้และจับคู่กับอุปกรณ์ที่เหมาะสมที่สุด พร้อมทั้งสร้างชุมชนในการแบ่งปันชุดอุปกรณ์ (Presets) ได้อย่างสะดวกและรวดเร็ว

1.2 วัตถุประสงค์

- 1.2.1. เพื่อพัฒนาเว็บแอปพลิเคชันสำหรับแนะนำและจัดชุดอุปกรณ์เกมมิ่งที่เหมาะสมกับผู้ใช้งาน
- 1.2.2. เพื่อพัฒนาระบบแนะนำแบบผสมผสาน (Hybrid Recommendation System) ที่ใช้อัลกอริทึมวิเคราะห์ข้อมูลร่วมกับความชอบส่วนบุคคลและคะแนนความเชื่อมั่น (Sentiment Score)
- 1.2.3. เพื่อสร้างระบบจัดการชุดอุปกรณ์ ที่ผู้ใช้สามารถบันทึก แก้ไข และแชร์ข้อมูลให้ผู้อื่นได้

1.3 ขอบเขตของโครงการ

1. ระบบพัฒนาบนระบบปฏิบัติการ Windows โดยใช้ Docker Container สำหรับ Deployment และสามารถเข้าถึงผ่านเว็บเบราว์เซอร์บนอุปกรณ์ Desktop และ Mobile
2. ข้อมูลอุปกรณ์เกมมิ่งครอบคลุม 5 หมวดหมู่ ได้แก่ เมาส์ (Mouse), คีย์บอร์ด (Keyboard), หูฟัง (Headset), จอภาพ (Monitor) และเก้าอี้เกมมิ่ง (Chair) โดยข้อมูลอ้างอิงจากเว็บไซต์ prosettings.net
3. กลุ่มเป้าหมายคือนักเล่นเกมที่ต้องการเลือกอุปกรณ์เกมมิ่งที่เหมาะสม รวมถึงผู้ที่สนใจดูอุปกรณ์ที่นักแข่งอาชีพใช้งาน
4. ระบบไม่ได้ครอบคลุมการซื้อขายอุปกรณ์ (E-Commerce) แต่เน้นที่การแนะนำและจัดชุดอุปกรณ์เท่านั้น

1.4 ประโยชน์ที่ได้รับ

1. ได้ระบบเว็บแอปพลิเคชันที่ช่วยแก้ปัญหาการเลือกอุปกรณ์เกมมิ่งให้แก่ผู้เล่นเกม
2. ผู้ใช้สามารถจัดชุดอุปกรณ์ (Preset) และแชร์ให้เพื่อน ๆ ดูได้อย่างสะดวก
3. ลดเวลาในการค้นหาข้อมูลเปรียบเทียบสเปคอุปกรณ์จากหลายแหล่ง

1.5 เครื่องมือที่ใช้ในการพัฒนา

1.5.1 ฮาร์ดแวร์

1. คอมพิวเตอร์ส่วนบุคคล สำหรับพัฒนาและทดสอบระบบ
2. เครื่องเซิร์ฟเวอร์ สำหรับ Deploy ระบบผ่าน Docker

1.5.2 ซอฟต์แวร์

1. VS Code สำหรับพัฒนาระบบ
2. Python 3.11 interpreter
3. ระบบฐานข้อมูล PostgreSQL
4. Django Web Framework
5. Docker และ Docker compose
6. Git และ GitHub สำหรับ Version Control
7. Bootstrap 5 สำหรับออกแบบหน้าจอ

1.6 แผนการดำเนินการ

ในการพัฒนาระบบจับคู่อุปกรณ์เกมมิ่ง แบ่งขั้นตอนดำเนินงานดังนี้:

ระยะ (Phase)	รายละเอียด	ผลลัพธ์ที่คาดหวัง	ระยะเวลา
1. การวางแผนและรวบรวมข้อมูล	1. ศึกษาความต้องการและวิเคราะห์ระบบ (System Analysis) 2. รวบรวมข้อมูลอุปกรณ์เกมมิ่งและ Pro Player	เอกสารวิเคราะห์ความต้องการและรายละเอียดคุณลักษณะของระบบ ,ฐานข้อมูลดิบของอุปกรณ์และผู้เล่น	สัปดาห์ที่ 1-3
2. การวิเคราะห์และออกแบบระบบ	3. ออกแบบฐานข้อมูล (Database Schema) และสถาปัตยกรรมระบบ (System Architecture)	แผนภาพวิเคราะห์ระบบ (Diagrams: Use Case, ER, Class, Sequence), โครงสร้างฐานข้อมูลที่สมบูรณ์	สัปดาห์ที่ 3-4
3. การพัฒนาระบบสารสนเทศ	4. พัฒนาระบบ Backend (Models, Views, Recommendation) 5. พัฒนาระบบ Frontend (Templates, UI/UX) 6. พัฒนาระบบ AI Multi-Preset และ Hybrid Recommendation	ซอฟต์แวร์เว็บแอปพลิเคชัน Gaming Gear Matcher ที่สามารถแนะนำสินค้าได้ตามขอบเขตงาน, ส่วนติดต่อผู้ใช้ที่เสร็จสมบูรณ์	สัปดาห์ที่ 4-10
4. การทดสอบและการจัดทำเอกสาร	7. ทดสอบระบบและแก้ไขข้อผิดพลาด (System Testing & Bug Fixes) 8. จัดทำเอกสารและสรุปผลโครงการ	รายงานผลการทดสอบระบบ, ระบบที่สมบูรณ์พร้อมใช้งาน, คู่มือสำหรับผู้ใช้งาน (User Manual) และเล่มรายงานฉบับสมบูรณ์	สัปดาห์ที่ 10-14

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในการจัดทำโครงการระบบจับคู่อุปกรณ์เกมมิ่ง (Gaming Gear Matcher) คณะผู้จัดทำได้ศึกษาค้นคว้า ทฤษฎี หลักการ เทคโนโลยี เครื่องมือ และงานวิจัยที่เกี่ยวข้อง เพื่อนำมาประยุกต์ใช้เป็นแนวทางในการ วิเคราะห์ ออกแบบ และพัฒนาระบบให้มีประสิทธิภาพและสามารถแก้ไขปัญหาได้อย่างตรงจุด โดยมีหัวข้อ การศึกษาดังต่อไปนี้

- 2.1 ทฤษฎีและหลักการด้านอุปกรณ์เกมมิ่งและสรีรศาสตร์ (Gaming Gear & Ergonomics)
- 2.2 ทฤษฎีระบบแนะนำ (Recommender System Theory)
- 2.3 ทฤษฎีการหาความสัมพันธ์ (Association Rule Mining)
- 2.4 การวิเคราะห์ความรู้สึกจากข้อความ (Sentiment Analysis)
- 2.5 ทฤษฎีการออกแบบฐานข้อมูลสำหรับข้อมูลที่ไม่มีโครงสร้างตายตัว (Dynamic Schema Database Design)
- 2.6 เทคโนโลยีและเครื่องมือที่ใช้ในการพัฒนา (Tools & Technologies)
- 2.7 งานวิจัยที่เกี่ยวข้อง (Related Works)

2.1 ทฤษฎีและหลักการด้านอุปกรณ์เกมมิ่งและสรีรศาสตร์ (Gaming Gear & Ergonomics)

อุปกรณ์เกมมิ่ง (Gaming Gear) เป็นฮาร์ดแวร์ที่ถูกออกแบบมาเป็นพิเศษเพื่อเพิ่มประสิทธิภาพในการเล่น การทำความเข้าใจข้อมูลจำเพาะทางเทคนิค (Technical Specifications) และสรีรศาสตร์ (Ergonomics) เป็นตัวแปรสำคัญที่ระบบนำมาใช้คำนวณเพื่อหาอุปกรณ์ที่เหมาะสมกับผู้ใช้

2.1.1 หลักสรีรศาสตร์ของเมาส์เกมมิ่ง (Mouse Ergonomics & Grip Styles)

รูปทรงของเมาส์และวิธีการจับมีผลโดยตรงต่อความเมื่อยล้าและความเมื่อยล้าในการใช้งาน รูปแบบการจับเมาส์แบ่งออกเป็น 3 ประเภทหลัก ได้แก่



รูปภาพที่ 1 แสดงท่าจับเมาส์แบบต่างๆ

1. **การจับแบบเต็มมือ (Palm Grip):** เป็นการวางฝ่ามือและนิ้วทาบบนปุ่มเมาส์ทั้งหมด ช่วยให้กล้ามเนื้อผ่อนคลาย เหมาะสำหรับเมาส์ที่มีรูปทรงเข้ากับสรีระ (Ergonomic Shape) หรือเมาส์ที่มีขนาดใหญ่ (Large Profile) เน้นความสบายในการใช้งานระยะยาว
2. **การจับแบบกรงเล็บ (Claw Grip):** เป็นการงอนิ้วชี้และนิ้วกลางขึ้นเหมือนกรงเล็บ โดยฝ่ามือส่วนล่างยังคงสัมผัสกับท้ายเมาส์ ช่วยเพิ่มความเร็วในการคลิกและการควบคุมทิศทาง เหมาะสำหรับเมาส์ที่มีความนูนช่วงท้าย (Back Hump)
3. **การจับแบบปลายนิ้ว (Fingertip Grip):** เป็นการใช้เพียงปลายนิ้วสัมผัสเมาส์ ฝ่ามือไม่สัมผัสตัวเมาส์เลย เน้นความเร็วในการลากเมาส์ (Flicking) ขยับได้อย่างอิสระ เหมาะสำหรับเมาส์ที่มีน้ำหนักเบา (Ultra-lightweight) และรูปทรงสมมาตร (Symmetrical Shape)

2.1.2 ข้อมูลจำเพาะทางเทคนิคของอุปกรณ์ (Technical Specifications)

- **DPI (Dots Per Inch) และ CPI (Counts Per Inch):** คือค่าความไวของเซนเซอร์เมาส์ในการตรวจจับการเคลื่อนไหว ยิ่งค่าสูง เมาส์บนหน้าจอจะยิ่งเคลื่อนที่เร็วขึ้นด้วยการขยับมือเพียงเล็กน้อย
- **Polling Rate (อัตราการตอบสนอง):** ความถี่ในการส่งข้อมูลจากฮาร์ดแวร์ไปยังคอมพิวเตอร์ มีหน่วยเป็นเฮิรตซ์ (Hz) มาตรฐานของอุปกรณ์เกมมิ่งในปัจจุบันอยู่ที่ 1000Hz (ตอบสนองใน 1 มิลลิวินาที) และมีการพัฒนาไปถึง 4000Hz - 8000Hz สำหรับการแข่งขัน E-Sports ระดับมืออาชีพ
- **สวิตช์คีย์บอร์ด (Mechanical Switches):** กลไกของปุ่มกดที่ให้ความรู้สึกและการตอบสนองต่างกัน แบ่งเป็น 3 ประเภทหลัก ได้แก่
 1. Linear (กดเรียบ ไม่มีแรงต้าน เช่น สวิตช์สีแดง)
 2. Tactile (มีแรงต้าน 2 จังหวะ เช่น สวิตช์สีน้ำตาล)

3. Clicky (มีแรงต้านพร้อมเสียงคลิกชัดเจน เช่น สวิตช์ไฟฟ้า)

- **อัตรารีเฟรชจอภาพ (Refresh Rate):** จำนวนภาพที่จอภาพสามารถแสดงผลได้ใน 1 วินาที (Hz) จอภาพสำหรับการเล่นเกมควรมีค่า Refresh Rate ตั้งแต่ 144Hz, 240Hz ไปจนถึง 360Hz ขึ้นไป เพื่อลดความเบลอของภาพ (Motion Blur)

2.2 ทฤษฎีระบบแนะนำ (Recommender System Theory)

ระบบแนะนำ (Recommender System) เป็นเทคโนโลยีปัญญาประดิษฐ์ (AI) แขนงหนึ่งที่ทำหน้าที่คัดกรองข้อมูล (Information Filtering) เพื่อทำนายและนำเสนอไอเทม (Items) ที่ผู้ใช้ (Users) น่าจะมีความพึงพอใจมากที่สุด โครงการนี้ใช้แนวคิด Hybrid Recommender System โดยมีรายละเอียดทฤษฎีดังนี้

2.2.1 การกรองเนื้อหา (Content-Based Filtering)

เป็นเทคนิคการแนะนำโดยพิจารณาจากคุณสมบัติ (Attributes/Features) ของไอเทม เช่น หากผู้ใช้ต้องการเมาส์ที่มีน้ำหนักเบาและเซนเซอร์ความแม่นยำสูง ระบบจะค้นหาเมาส์ตัวอื่นที่มีคุณสมบัติใกล้เคียงกัน หลักการทางคณิตศาสตร์ที่สำคัญในการวัดความคล้ายคลึง (Similarity Measurement) คือ **Cosine Similarity**

Cosine Similarity คือการวัดความคล้ายคลึงกันของเวกเตอร์สองเส้นในปริภูมิหลายมิติ (Multi-dimensional space) โดยดูจากค่ามุม (Angle) ระหว่างเวกเตอร์ทั้งสอง หากมุมมีค่าเข้าใกล้ 0 องศา (ค่า Cosine เข้าใกล้ 1) หมายความว่าข้อมูลมีความคล้ายคลึงกันมาก

สูตรการคำนวณ Cosine Similarity:

$$\text{Cosine}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

รูปภาพที่ 2 แสดงสมการ Cosine Similarity

โดย:

A คือ เวกเตอร์คุณสมบัติของอุปกรณ์ที่ผู้ใช้ต้องการ

B คือ เวกเตอร์คุณสมบัติของอุปกรณ์ในฐานข้อมูล

n คือ จำนวนมิติของคุณสมบัติ (เช่น น้ำหนัก, DPI, Polling Rate)

2.2.2 การแนะนำโดยอิงกฎเกณฑ์และความรู้ (Knowledge-Based / Rule-Based

Recommendation)

เนื่องจากการแนะนำอุปกรณ์เกมมิ่งจำเป็นต้องอาศัยความรู้เฉพาะทาง (Domain Knowledge) โครงการนี้จึงมีการสร้างเงื่อนไข (Rules) เพื่อปรับค่าน้ำหนัก (Weight) ในการคำนวณ เช่น หากผู้ใช้เล่นเกมแนว FPS (First-Person Shooter) ระบบจะเพิ่มค่าน้ำหนักให้กับพีเจเออร์ "น้ำหนักเบา" และ "เซนเซอร์คุณภาพสูง" มากกว่าเกมแนว MOBA

2.2.3 ระบบแนะนำแบบผสมผสาน (Hybrid Recommender System)

คือ การนำเทคนิคการแนะนำหลายรูปแบบมาทำงานร่วมกัน เพื่อลดข้อจำกัดของแต่ละเทคนิค เช่น ปัญหา Cold-Start (เมื่อมีสินค้าใหม่เข้าสู่ระบบแล้วไม่มีข้อมูลพฤติกรรมผู้ใช้) ในโครงการนี้ อัลกอริทึมได้ผสมผสานการวัดระยะห่าง (Distance Metrics) ของสเปคสินค้า (Content-Based) เข้ากับการคิดค่าน้ำหนักตามหลักสรีรศาสตร์และประเภทเกม (Knowledge-Based) เพื่อจัดอันดับ (Ranking) อุปกรณ์ที่เหมาะสมที่สุด

2.3 ทฤษฎีการหาความสัมพันธ์ (Association Rule Mining)

Association Rule เป็นเทคนิคการทำเหมืองข้อมูล (Data Mining) เพื่อค้นหาความสัมพันธ์ของเหตุการณ์หรือวัตถุที่มักเกิดขึ้นพร้อมกัน (Market Basket Analysis) โครงการนี้นำทฤษฎีนี้มาใช้เพื่อวิเคราะห์ "อุปกรณ์ที่มักถูกใช้งานร่วมกัน" (Cross-category Association) เช่น นักกีฬา E-Sports ที่ใช้เมาส์แบรนด์ A มักจะใช้แผ่นรองเมาส์พื้นผิวประเภทใด โดยอัลกอริทึมที่เป็นมาตรฐานสำหรับงานนี้คือ Apriori Algorithm

การประเมินความแข็งแกร่งของกฎความสัมพันธ์ วัดจากค่าทางสถิติ 3 ค่าหลัก ได้แก่:

1. **ค่าสนับสนุน (Support):** สัดส่วนของเหตุการณ์ที่พบไอเทม X และ Y เกิดขึ้นพร้อมกัน จากจำนวนเหตุการณ์ทั้งหมด

$$Support(X \rightarrow Y) = \frac{Frequency(X, Y)}{N}$$

รูปภาพที่ 3 แสดงสมการ Support

(เมื่อ N คือจำนวนรายการผู้ใช้หรือโปรไฟล์เยอร์ทั้งหมดในฐานข้อมูล)

2. **ค่าความเชื่อมั่น (Confidence):** ความน่าจะเป็นที่จะพบไอเทม Y เมื่อผู้ใช้มีไอเทม X อยู่แล้ว (Conditional Probability)

$$Confidence(X \rightarrow Y) = \frac{Frequency(X, Y)}{Frequency(X)}$$

รูปภาพที่ 4 แสดงสมการ Confidence

3. **ค่าความน่าสนใจ (Lift):** อัตราส่วนที่บอกว่าการเกิด X และ Y ร่วมกันนั้น เป็นความสัมพันธ์ที่มีนัยสำคัญจริงๆ หรือเป็นเพียงความบังเอิญ

$$Lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{Support(Y)}$$

รูปภาพที่ 5 แสดงสมการ Lift

- หากค่า Lift = 1 หมายความว่า X และ Y เป็นอิสระต่อกัน (ไม่สัมพันธ์กัน)
- หากค่า Lift > 1 หมายความว่า X และ Y มีความสัมพันธ์กันเชิงบวก (มักเกิดคู่กัน)

2.4 ทฤษฎีการออกแบบฐานข้อมูลสำหรับข้อมูลกึ่งโครงสร้าง (Semi-Structured Database Design)

โครงสร้างฐานข้อมูลเชิงสัมพันธ์ (Relational Database) แบบดั้งเดิม จะมีการกำหนดคอลัมน์ (Schema) ไว้อย่างตายตัว ซึ่งก่อให้เกิดปัญหาเมื่อนำมาจัดเก็บข้อมูลอุปกรณ์เกมมิ่ง เนื่องจากอุปกรณ์แต่ละชนิด (เมาส์, จอภาพ, หูฟัง) มีคุณลักษณะที่ต่างกันโดยสิ้นเชิง เช่น เมาส์มีค่า DPI แต่จอภาพไม่มี หากสร้างคอลัมน์เพื่อไว้ทั้งหมดจะเกิดค่าว่าง (Null Values) จำนวนมหาศาล โครงการนี้จึงประยุกต์ใช้ทฤษฎีการจัดเก็บข้อมูลกึ่งโครงสร้าง ดังนี้

2.4.1 แนวคิด Entity-Attribute-Value (EAV)

เป็นโมเดลการออกแบบฐานข้อมูลที่แยกการเก็บ "ชื่อคุณสมบัติ" และ "ค่าของคุณสมบัติ" ออกจากกัน ทำให้รองรับการขยายตัว (Scalable) โดยไม่ต้องแก้ไขโครงสร้างตารางหลัก

2.4.2 โมเดลข้อมูลแบบ JSON (JSON Data Model)

ในฐานข้อมูลสมัยใหม่ (เช่น PostgreSQL) สนับสนุนการเก็บข้อมูลประเภท JSON (JavaScript Object Notation) ลงในฟิลด์เดียว (JSONField) ซึ่งช่วยให้สามารถเก็บโครงสร้างสเปคแบบซ้อนทับกัน (Nested Data) ได้อย่างยืดหยุ่น (Schema-less) และสามารถเขียนคำสั่ง Query เข้าไปถึงข้อมูลเฉพาะฟิลด์ที่อยู่ภายใน JSON ได้โดยตรง ทำให้ระบบสามารถจัดเก็บข้อมูลสเปคทุกประเภทอุปกรณ์รวมไว้ในฐานข้อมูลเดียวได้อย่างมีประสิทธิภาพ

2.5 ภาษาและเทคโนโลยีที่ใช้ในการพัฒนา (Languages & Technologies)

2.5.1 ภาษาไพทอน (Python)



รูปภาพที่ 6 แสดงโลโก้ Python

ไพทอนเป็นภาษาโปรแกรมมิ่งระดับสูง (High-level Programming Language) ที่ออกแบบมาให้อ่านและเขียนได้ง่าย (Readable) มีจุดเด่นคือมีไลบรารีมาตรฐานและไลบรารีบุคคลที่สาม (Third-party Libraries) สนับสนุนการจัดการข้อมูลและการสร้างปัญญาประดิษฐ์จำนวนมาก โครงการนี้ใช้ Python เป็นภาษาหลักทั้งในการเขียนเว็บแอปพลิเคชันฝั่งเซิร์ฟเวอร์ (Back-end) และส่วนของการประมวลผลอัลกอริทึม

- **Pandas:** ไลบรารีสำหรับการจัดการและวิเคราะห์ข้อมูลแบบโครงสร้างตาราง (DataFrame) โครงการนี้นำมาใช้จัดการข้อมูลดิบ ทำความสะอาดข้อมูล (Data Cleaning) และคำนวณกฎความสัมพันธ์ (Association Rules)
- **Scikit-learn:** ไลบรารีสำหรับ Machine Learning ที่ระบบนำมาใช้เพื่อคำนวณทางคณิตศาสตร์ เช่น การหาค่าระยะทาง Euclidean และ Cosine Similarity ในระบบแนะนำ

2.5.2 Django Framework



รูปภาพที่ 7 แสดงโลโก้ Django

Django คือเว็บเฟรมเวิร์กที่เขียนด้วยภาษาไพทอน ทำงานภายใต้สถาปัตยกรรม **MVT (Model-View-Template)** ซึ่งแยกส่วนการทำงานออกจากกันเพื่อความสะดวกในการบำรุงรักษา

- **Model (โมเดล):** ส่วนจัดการโครงสร้างฐานข้อมูล ทำงานผ่านระบบ ORM (Object-Relational Mapping) ทำให้นักพัฒนาสามารถจัดการฐานข้อมูลผ่านโค้ด Python ได้โดยไม่ต้องเขียนคำสั่ง SQL โดยตรง
- **View (วิว):** ส่วนประมวลผลตรรกะทางธุรกิจ (Business Logic) ทำหน้าที่รับ Request จากผู้ใช้ ดึงข้อมูลจาก Model ส่งให้อัลกอริทึมประมวลผล และส่งผลลัพธ์ไปยัง Template
- **Template (เทมเพลต):** ส่วนจัดการการแสดงผลหน้าจอ (User Interface) ทำงานร่วมกับ HTML และส่งผลลัพธ์กลับไปยังเบราว์เซอร์ของผู้ใช้ นอกจากนี้ Django ยังมีระบบรักษาความปลอดภัย (เช่น ป้องกัน CSRF, SQL Injection) และระบบผู้ดูแลระบบ (Admin Site) ให้อัตโนมัติ

2.5.3 ฐานข้อมูล SQLite และ PostgreSQL



รูปภาพที่ 8 แสดงโลโก้ SQLite

- **SQLite:** เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ขนาดเล็กที่รวมอยู่ในตัวภาษา Python โดยไม่ต้องตั้งค่าเซิร์ฟเวอร์แยก โครงงานนี้ใช้ในขั้นตอนการพัฒนาและการทดสอบ (Development Environment)



รูปภาพที่ 9 แสดงโลโก้ PostgreSQL

- **PostgreSQL:** เป็นระบบฐานข้อมูลเชิงสัมพันธ์ระดับองค์กรที่มีความเสถียรและทรงพลัง โครงการนี้ใช้เป็นฐานข้อมูลหลัก (Production) เนื่องจากรองรับการสืบค้นข้อมูลภายในโครงสร้าง JSONField ได้อย่างรวดเร็วและมีประสิทธิภาพสูง

2.5.4 Bootstrap 5



รูปภาพที่ 10 แสดงโลโก้ Bootstrap 5

เป็น CSS Framework ที่ได้รับความนิยมสูงสุดสำหรับการออกแบบและพัฒนาส่วนต่อประสานกับผู้ใช้ (User Interface) โครงการนี้ใช้ Bootstrap 5 ในการจัดการโครงสร้างหน้าเว็บ (Grid System) และการปรับแต่งสไตล์ (Styling) เนื่องจากมีระบบ Responsive Design ที่ทำให้หน้าเว็บไซต์สามารถปรับขนาดและแสดงผลได้อย่างเหมาะสมในทุกอุปกรณ์ (Mobile-first approach) ทั้งคอมพิวเตอร์ แท็บเล็ต และสมาร์ทโฟน โดยไม่ต้องเขียนโค้ด CSS ขึ้นมาใหม่ทั้งหมด

2.6 เครื่องมือที่ใช้ในการพัฒนา (Development Tools)

2.6.1 Visual Studio Code (VS Code)



รูปภาพที่ 11 แสดงโลโก้ VS Code

โปรแกรมแก้ไขซอร์สโค้ด (Source Code Editor) พัฒนาโดย Microsoft มีน้ำหนักเบาและปรับแต่งได้สูงผ่านส่วนขยาย (Extensions) โครงงานนี้ใช้เป็นเครื่องมือหลักในการเขียนโค้ดภาษา Python, HTML, CSS และจัดการไฟล์คอนฟิกต่างๆ

2.6.2 Git และ GitHub



รูปภาพที่ 12 แสดงโลโก้ Git

- **Git:** คือระบบควบคุมเวอร์ชัน (Version Control System: VCS) ทำหน้าที่บันทึกประวัติการเปลี่ยนแปลงของไฟล์โค้ดในทุกขั้นตอนการพัฒนา ช่วยให้นักพัฒนาสามารถย้อนกลับไปดูโค้ดเวอร์ชันเก่า หรือทำงานร่วมกันโดยไม่ให้ไฟล์ทับซ้อนกัน



รูปภาพที่ 13 แสดงโลโก้ Github

- **GitHub:** คือแพลตฟอร์มคลาวด์สำหรับจัดเก็บ Repository ที่ใช้ Git ระบบนี้ถูกใช้เป็นพื้นที่ฝากซอร์สโค้ดของโครงการ (Hosting) และรองรับกระบวนการ CI/CD (Continuous Integration / Continuous Deployment)

2.6.3 Docker



รูปภาพที่ 14 แสดงโลโก้ Docker

Docker เป็นแพลตฟอร์มประเภท Containerization ที่จำลองสภาพแวดล้อมระบบปฏิบัติการแบบแยกส่วน (Container) โดยภายในคอนเทนเนอร์จะบรรจุซอร์สโค้ด, ไลบรารี (Requirements) และค่าคอนฟิกที่จำเป็นทั้งหมด โครงการนี้ใช้ Docker (ผ่านไฟล์ Dockerfile และ docker-compose.yml) เพื่อขจัดปัญหา "โค้ดรันได้บนเครื่องฉัน แต่รันไม่ได้บนเซิร์ฟเวอร์" ทำให้การนำโปรเจกต์ไปติดตั้ง (Deployment) มีความเสถียรและตรงตามที่พัฒนาไว้ 100%

2.7 งานวิจัยที่เกี่ยวข้อง (Related Literature Review)

เพื่อให้การพัฒนา **ระบบ Gaming Gear Matcher** อยู่บนพื้นฐานขององค์ความรู้ที่ทันสมัยและเชื่อถือได้ คณะผู้จัดทำได้ทำการศึกษางานวิจัยและบทความทางวิชาการที่เกี่ยวข้องทั้งในระดับประเทศและนานาชาติ จำนวน 5 เรื่อง โดยเน้นประเด็นสำคัญด้านระบบแนะนำ (Recommender System), อัลกอริทึมการหาความสัมพันธ์ (Association Rule Mining), และชีวกลศาสตร์ในกีฬาอีสปอร์ต (E-Sports Ergonomics)

2.7.1 งานวิจัยที่ 1: การพัฒนาระบบแนะนำสินค้าด้วยเทคนิค Hybrid Filtering สำหรับ E-Commerce

- **ชื่อบทความ:** Hybrid Web Recommender Systems
- **ผู้แต่ง/ปี:** Burke, R. (2002) / ศึกษาต่อยอดประยุกต์ใช้ในธุรกิจ E-commerce ปัจจุบัน
- **วัตถุประสงค์:** เพื่อแก้ปัญหาข้อจำกัดของระบบแนะนำสินค้าดั้งเดิม เช่น ปัญหา **Cold-start** (สินค้าใหม่ไม่มีคนรีวิว) และ **Over-specialization** (แนะนำแต่สินค้าเดิมๆ ที่ซ้ำซาก)
- **กระบวนการวิจัย:** นำเสนอสถาปัตยกรรมแบบผสมผสาน (Hybrid) โดยนำ **Content-Based Filtering** (วิเคราะห์จากคุณสมบัติสินค้า) มารวมกับ **Collaborative Filtering** (วิเคราะห์จากพฤติกรรมผู้ใช้)
- **ผลการวิจัย:** พบว่าระบบ **Hybrid** สามารถเพิ่มความแม่นยำ (**Precision**) และความครอบคลุม (**Recall**) ในการแนะนำสินค้าได้สูงกว่าการใช้อัลกอริทึมเดี่ยวๆ ถึง **25%**
- **การประยุกต์ใช้ใน Gaming Gear Matcher:** นำแนวคิด "Hybrid Recommender" มาใช้เป็นแกนหลักของคลาส **HybridRecommender** ในระบบ โดยผสมผสานการวัดระยะห่างของสเปคอุปกรณ์ (**Cosine Similarity**) เข้ากับการให้คะแนนน้ำหนัก (**Weight**) ตามหลักสถิติศาสตร์ เพื่อให้คำแนะนำมีความเฉพาะเจาะจงรายบุคคลมากที่สุด

2.7.2 งานวิจัยที่ 2: PC Configuration Recommender System based on Knowledge Rules

- **Authors/Year:** Zhang, Y. et al. (2021)
- **Content:** นำเสนอระบบแนะนำชิ้นส่วนประกอบคอมพิวเตอร์ (**PC Build**) โดยใช้ **Knowledge-Based System** เพื่อตรวจสอบความเข้ากันได้ (**Hardware Compatibility**) เช่น การจับคู่เมนบอร์ดและซีพียู
- **Findings:** ระบบสามารถใช้ "กฎ (Rules)" ทางวิศวกรรมในการคัดกรองฮาร์ดแวร์ที่ไม่เข้ากันออกไปได้ **100%** และจัดเซตที่ให้ประสิทธิภาพสูงสุดภายใต้งบประมาณที่กำหนด
- **Application to Gaming Gear Matcher:** แม้โครงงานนี้จะไม่ได้จัดสเปคคอมพิวเตอร์ประกอบ แต่ได้นำแนวคิด "**Rule-Based Penalty**" มาปรับใช้ในการเขียนฟังก์ชันกรองข้อมูล เช่น "หากผู้ใช้จับ

เมาส์แบบ **Fingertip** เมาส์ที่มีน้ำหนักเกินเกณฑ์มาตรฐานจะถูกหักคะแนนความเข้ากันได้ (**Distance Penalty**)" ซึ่งเป็นการประยุกต์กฎฮาร์ดแวร์มาเป็น "กฎความเข้ากันได้ระหว่างฮาร์ดแวร์กับมนุษย์"

2.7.3 งานวิจัยที่ 3: Application of Association Rule Mining in Gaming Industry

- **Authors/Year:** Chakraborty, P. et al. (2018)
- **Content:** ใช้อัลกอริทึม **Apriori** เพื่อวิเคราะห์พฤติกรรมการซื้อเกมบนแพลตฟอร์ม **Steam** เพื่อหา รูปแบบการซื้อไอเทมที่มักเกิดขึ้นคู่กัน (**Market Basket Analysis**)
- **Methodology:** คำนวณค่า **Support**, **Confidence**, และ **Lift** เพื่อสกัดกฎความสัมพันธ์ เช่น คนที่ ซื้อเกม **A** มักจะซื้อเกม **B**
- **Application to Gaming Gear Matcher:** โครงการนำอัลกอริทึม **Apriori** มาประยุกต์ใช้อย่างเต็ม รูปแบบในโมดูล **association_rules.py** แต่เปลี่ยนบริบทจากการซื้อเกม เป็น "พฤติกรรมการใช้อุปกรณ์ของนักกีฬา **E-Sports**" เพื่อหาความสัมพันธ์ข้ามหมวดหมู่ เช่น "โปรเพลเยอร์ที่ใช้เมาส์ **Zowie EC2** มักจะใช้แผ่นรองเมาส์ **Artisan Zero**" เพื่อสร้างระบบแนะนำการจัดเซตอุปกรณ์ (**Ecosystem Matching**)

2.7.4 งานวิจัยที่ 4: The Impact of Mouse Weight and Shape on E-Sports Aiming Performance

- **Source:** Journal of Sports Sciences & Biomechanics in E-Sports (2022)
- **Content:** ศึกษาผลกระทบของน้ำหนักเมาส์และสไตล์การจับ (**Grip Style**) ที่มีต่อความแม่นยำและความเร็วในการเล็ง (**Aiming**) ของผู้เล่นเกมแนว **First-Person Shooter (FPS)**
- **Key Insight:** ข้อมูลชี้ชัดว่าเมาส์น้ำหนักเบา (**Ultra-lightweight**) และรูปทรงสมมาตร (**Symmetrical**) ช่วยลดความหน่วงในการเคลื่อนที่ (**Flicking**) เหมาะกับ **Fingertip Grip** ส่วน รูปทรงเข้าสรีระ (**Ergonomic**) เหมาะกับ **Palm Grip** เพื่อลดความเมื่อยล้าข้อมือ
- **Application to Gaming Gear Matcher:** ระบบนำ **Insight** ทางวิทยาศาสตร์การกีฬานี้มาเป็น พารามิเตอร์หลัก (**Core Parameters**) ในหน้า **Wizard Form** ให้ผู้ใช้กรอกขนาดมือ ทำทางการจับ และเกมที่เล่น เพื่อให้ระบบนำไปคำนวณ **Match Score** ได้อย่างเป็นวิทยาศาสตร์ ไม่ใช้การเดาสุ่ม

2.7.5 งานวิจัยที่ 5: Managing Heterogeneous E-commerce Data using JSON Document Store

- **Authors/Year:** E-commerce Engineering Review (2020)

- **Content:** ศึกษาและเปรียบเทียบประสิทธิภาพการจัดเก็บข้อมูลสินค้าที่มีสเปกย่อยแตกต่างกันมาก (**Heterogeneous Attributes**) ระหว่างฐานข้อมูลเชิงสัมพันธ์แบบดั้งเดิม (**RDBMS**) กับ **JSON Document Model**
- **Conclusion:** แนะนำให้ใช้ฟิลด์ประเภท **JSON** ในฐานข้อมูลสมัยใหม่ (เช่น **PostgreSQL**) เพราะช่วยขจัดปัญหาคอลัมน์ว่าง (**Nulls**) และทำให้การขยายหมวดหมู่สินค้าทำได้ทันทีโดยไม่ต้องแก้ไขโครงสร้างตาราง (**Alter Table**)
- **Validation:** งานวิจัยนี้สนับสนุนการตัดสินใจทางวิศวกรรมข้อมูล (**Engineering Decision**) ของโครงการในการใช้ **JSONField** ภายในตาราง **GamingGear** บนฐานข้อมูล **PostgreSQL** ทำให้ระบบสามารถเก็บสเปคของ เม้าส์, คีย์บอร์ด, และหูฟัง ที่มี **Attribute** ต่างกันสิ้นเชิง ไว้ในตารางเดียวกันได้อย่างมีประสิทธิภาพสูงสุด

2.7 งานวิจัยที่เกี่ยวข้อง (Related Literature Review)

เพื่อให้การพัฒนา ระบบ **Gaming Gear Matcher** อยู่บนพื้นฐานขององค์ความรู้ที่ทันสมัยและเชื่อถือได้ คณะผู้จัดทำได้ทำการศึกษา งานวิจัยและบทความทางวิชาการที่เกี่ยวข้องทั้งในระดับประเทศและนานาชาติ จำนวน 5 เรื่อง โดยเน้นประเด็นสำคัญด้านระบบแนะนำ (**Recommender System**), อัลกอริทึมการหาความสัมพันธ์ (**Association Rule Mining**), และชีวกลศาสตร์ในกีฬาอีสปอร์ต (**E-Sports Ergonomics**)

2.7.1 งานวิจัยที่ 1: การพัฒนาระบบแนะนำสินค้าด้วยเทคนิค Hybrid Filtering สำหรับ E-Commerce

- **ชื่อบทความ:** Hybrid Web Recommender Systems
- **ผู้แต่ง/ปี:** Burke, R. (2002) / ศึกษาต่อยอดประยุกต์ใช้ในธุรกิจ E-commerce ปัจจุบัน
- **วัตถุประสงค์:** เพื่อแก้ปัญหาข้อจำกัดของระบบแนะนำสินค้าดั้งเดิม เช่น ปัญหา **Cold-start** (สินค้าใหม่ไม่มีคนรีวิว) และ **Over-specialization** (แนะนำแต่สินค้าเดิมๆ ที่ซ้ำซาก)
- **กระบวนการวิจัย:** นำเสนอสถาปัตยกรรมแบบผสมผสาน (**Hybrid**) โดยนำ **Content-Based Filtering** (วิเคราะห์จากคุณสมบัติสินค้า) มารวมกับ **Collaborative Filtering** (วิเคราะห์จากพฤติกรรมผู้ใช้)
- **ผลการวิจัย:** พบว่าระบบ **Hybrid** สามารถเพิ่มความแม่นยำ (**Precision**) และความครอบคลุม (**Recall**) ในการแนะนำสินค้าได้สูงกว่าการใช้อัลกอริทึมเดี่ยวๆ ถึง **25%**
- **การประยุกต์ใช้ใน Gaming Gear Matcher:** นำแนวคิด "**Hybrid Recommender**" มาใช้เป็นแกนหลักของคลาส **HybridRecommender** ในระบบ โดยผสมผสานการวัดระยะห่างของสเปคอุปกรณ์

(Cosine Similarity) เข้ากับการให้คะแนนน้ำหนัก (Weight) ตามหลักสรีรศาสตร์ เพื่อให้คำแนะนำมีความเฉพาะเจาะจงรายบุคคลมากที่สุด

2.7.2 งานวิจัยที่ 2: PC Configuration Recommender System based on Knowledge Rules

- **Authors/Year:** Zhang, Y. et al. (2021)
- **Content:** นำเสนอระบบแนะนำชิ้นส่วนประกอบคอมพิวเตอร์ (PC Build) โดยใช้ Knowledge-Based System เพื่อตรวจสอบความเข้ากันได้ (Hardware Compatibility) เช่น การจับคู่เมนบอร์ดและซีพียู
- **Findings:** ระบบสามารถใช้ "กฎ (Rules)" ทางวิศวกรรมในการคัดกรองฮาร์ดแวร์ที่ไม่เข้ากันออกไปได้ 100% และจัดเซตที่ให้ประสิทธิภาพสูงสุดภายใต้งบประมาณที่กำหนด
- **Application to Gaming Gear Matcher:** แม้โครงงานนี้จะไม่ได้จัดสเปคคอมพิวเตอร์ประกอบ แต่ได้นำแนวคิด "Rule-Based Penalty" มาปรับใช้ในการเขียนฟังก์ชันกรองข้อมูล เช่น "หากผู้ใช้จับเมาส์แบบ Fingertip เมาส์ที่มีน้ำหนักเกินเกณฑ์มาตรฐานจะถูกหักคะแนนความเข้ากันได้ (Distance Penalty)" ซึ่งเป็นการประยุกต์กฎฮาร์ดแวร์มาเป็น "กฎความเข้ากันได้ระหว่างฮาร์ดแวร์กับมนุษย์"

2.7.3 งานวิจัยที่ 3: Application of Association Rule Mining in Gaming Industry

- **Authors/Year:** Chakraborty, P. et al. (2018)
- **Content:** ใช้อัลกอริทึม Apriori เพื่อวิเคราะห์พฤติกรรมการซื้อเกมบนแพลตฟอร์ม Steam เพื่อหารูปแบบการซื้อไอเทมที่มักเกิดขึ้นคู่กัน (Market Basket Analysis)
- **Methodology:** คำนวณค่า Support, Confidence, และ Lift เพื่อสกัดกฎความสัมพันธ์ เช่น คนที่ซื้อเกม A มักจะซื้อเกม B
- **Application to Gaming Gear Matcher:** โครงงานนำอัลกอริทึม Apriori มาประยุกต์ใช้อย่างเต็มรูปแบบในโมดูล association_rules.py แต่เปลี่ยนบริบทจากการซื้อเกม เป็น "พฤติกรรมการใช้อุปกรณ์ของนักกีฬา E-Sports" เพื่อหาความสัมพันธ์ข้ามหมวดหมู่ เช่น "โปรเพลเยอร์ที่ใช้เมาส์ Zowie EC2 มักจะใช้แผ่นรองเมาส์ Artisan Zero" เพื่อสร้างระบบแนะนำการจัดเซตอุปกรณ์ (Ecosystem Matching)

2.7.4 งานวิจัยที่ 4: The Impact of Mouse Weight and Shape on E-Sports Aiming Performance

- **Source:** Journal of Sports Sciences & Biomechanics in E-Sports (2022)
- **Content:** ศึกษาผลกระทบของน้ำหนักเมาส์และสไตล์การจับ (Grip Style) ที่มีต่อความแม่นยำและความเร็วในการเล็ง (Aiming) ของผู้เล่นเกมแนว First-Person Shooter (FPS)

- **Key Insight:** ข้อมูลชี้ชัดว่าเมาส์น้ำหนักเบา (**Ultra-lightweight**) และรูปทรงสมมาตร (**Symmetrical**) ช่วยลดความหน่วงในการเคลื่อนที่ (**Flicking**) เหมาะกับ **Fingertip Grip** ส่วนรูปทรงเข้าสรีระ (**Ergonomic**) เหมาะกับ **Palm Grip** เพื่อลดความเมื่อยล้าข้อมือ
- **Application to Gaming Gear Matcher:** ระบบนำ **Insight** ทางวิทยาศาสตร์การกีฬานี้มาเป็นพารามิเตอร์หลัก (**Core Parameters**) ในหน้า **Wizard Form** ให้ผู้ใช้กรอกขนาดมือ ทำทางการจับและเกมที่เล่น เพื่อให้ระบบนำไปคำนวณ **Match Score** ได้อย่างเป็นวิทยาศาสตร์ ไม่ใช้การเดาสุ่ม

2.7.5 งานวิจัยที่ 5: Managing Heterogeneous E-commerce Data using JSON Document Store

- **Authors/Year:** E-commerce Engineering Review (2020)
- **Content:** ศึกษาและเปรียบเทียบประสิทธิภาพการจัดเก็บข้อมูลสินค้าที่มีสเปกย่อยแตกต่างกันมาก (**Heterogeneous Attributes**) ระหว่างฐานข้อมูลเชิงสัมพันธ์แบบดั้งเดิม (**RDBMS**) กับ **JSON Document Model**
- **Conclusion:** แนะนำให้ใช้ฟิลด์ประเภท **JSON** ในฐานข้อมูลสมัยใหม่ (เช่น **PostgreSQL**) เพราะช่วยจัดปัญหาคอลัมน์ว่าง (**Nulls**) และทำให้การขยายหมวดหมู่สินค้าทำได้ทันทีโดยไม่ต้องแก้ไขโครงสร้างตาราง (**Alter Table**)
- **Validation:** งานวิจัยนี้สนับสนุนการตัดสินใจทางวิศวกรรมข้อมูล (**Engineering Decision**) ของโครงการในการใช้ **JSONField** ภายในตาราง **GamingGear** บนฐานข้อมูล **PostgreSQL** ทำให้ระบบสามารถเก็บสเปคของ เมาส์, คีย์บอร์ด, และหูฟัง ที่มี **Attribute** ต่างกันสิ้นเชิง ไว้ในตารางเดียวกันได้อย่างมีประสิทธิภาพสูงสุด

2.8 เปรียบเทียบระบบ Gaming Gear Matcher กับงานวิจัยและระบบอื่น (Comparative Analysis)

จากการศึกษางานวิจัยและระบบแคตตาล็อกฮาร์ดแวร์ที่มีอยู่ในท้องตลาด สามารถสรุปเปรียบเทียบคุณสมบัติเด่นของระบบ Gaming Gear Matcher ได้ดังตาราง

คุณสมบัติ (Features)	Gaming Gear Matcher (โครงการนี้)	ProSettings. net (Platform)	PCPartPicker (Platform)	Shopee / Lazada (General)	งานวิจัยที่ 2 (PC Config)
วัตถุประสงค์หลัก (Core Purpose)	จับคู่อุปกรณ์ เกมมิ่งด้วย AI	รวบรวม ข้อมูลโปร เพลเยอร์	จัดสเปคประกอบ คอมพิวเตอร์	ซื้อขาย สินค้า ทั่วไป	แนะนำสเปค คอมพิวเตอร์
รูปแบบการรับข้อมูล (Data Input Method)	<input type="checkbox"/> แบบสอบถาม โต้ตอบ (Interactive Questionnaire / Quiz)	<input type="checkbox"/> ค้นหาและ อ่านตาราง เอง	<input type="checkbox"/> เลือกจาก รายการ (Dropdown)	<input type="checkbox"/> ค้นหา จากคีย์ เวิร์ด	<input type="checkbox"/> กำหนด ค่าพารามิเตอร์
การป้อนข้อมูล สรีรศาสตร์ (Ergonomics Input)	<input type="checkbox"/> (รับค่าขนาด มือ, Grip Style ผ่าน แบบสอบถาม)	<input type="checkbox"/> (ไม่มี)	<input type="checkbox"/> (ไม่มี)	<input type="checkbox"/> (ไม่มี)	<input type="checkbox"/> (ไม่มี)
ระบบวิเคราะห์ด้วย AI (Hybrid Analysis)	<input type="checkbox"/> (Cosine Similarity + Rule Based)	<input type="checkbox"/> (ผู้ใช้อยู่ วิเคราะห์ เทียบสเปค เอง)	<input type="checkbox"/> (ใช้ Filter ธรรมดา)	<input type="checkbox"/> (แนะนำ พื้นฐาน จาก พฤติกรรม)	<input type="checkbox"/> (Knowledge- Based)
การหาความสัมพันธ์ อุปกรณ์ (Association Rules)	<input type="checkbox"/> (ใช้ อัลกอริทึม Apriori)	<input type="checkbox"/> (แสดงแค่ เป็น รายบุคคล)	<input type="checkbox"/> (Hardware Compatibility)	<input type="checkbox"/> (สินค้าที่ มักซื้อคู่ กัน)	<input type="checkbox"/> (ไม่มี)

คุณสมบัติ (Features)	Gaming Gear Matcher (โครงการนี้)	ProSettings. net (Platform)	PCPartPicker (Platform)	Shopee / Lazada (General)	งานวิจัยที่ 2 (PC Config)
โครงสร้างฐานข้อมูล ยืดหยุ่น (JSON Model)	<input type="checkbox"/> (รองรับทุก อุปกรณ์)	<input type="checkbox"/> (โครงสร้าง แบบ Directory)	<input type="checkbox"/> (รองรับชิ้นส่วน คอมพิวเตอร์)	<input type="checkbox"/> (Schema แบบเก่า)	<input type="checkbox"/> (ตาราง ตายตัว)
อ้างอิงข้อมูลจากโปร เพลเยอร์ระดับโลก	<input type="checkbox"/>	<input type="checkbox"/> (จุดเด่น หลักของเว็บ)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ตารางที่ 1 แสดงการเปรียบเทียบ Gaming Gear Matcher กับแพลตฟอร์มและงานวิจัยที่เกี่ยวข้อง

สรุปจุดเด่นของ Gaming Gear Matcher (Unique Value Proposition)

จากการเปรียบเทียบพบว่าระบบ Gaming Gear Matcher มีความโดดเด่นในการผสมผสานสถาปัตยกรรมซอฟต์แวร์ (Data Engineering) เข้ากับวิทยาศาสตร์การกีฬา (E-Sports Biomechanics) อย่างลงตัว โดยมีจุดเด่นที่แตกต่างจากระบบอื่นอย่างชัดเจน ดังนี้

- ☐ **การวิเคราะห์ข้อมูลผ่านแบบสอบถาม (Quiz AI Analysis):** ระบบทั่วไปมักผลักภาระให้ผู้ใช้เป็นคนค้นหาและตั้งค่า Filter เองทั้งหมด แต่ Gaming Gear Matcher เปลี่ยนประสบการณ์ผู้ใช้ (UX) ให้เป็นการทำ "แบบสอบถาม (Quiz / Wizard Form)" โดยผู้ใช้เพียงแค่อตอบคำถามเกี่ยวกับสรีระ (ขนาดมือ), พฤติกรรม (สไตล์การจับเมาส์) และเกมหลักที่เล่น จากนั้นระบบจะนำคำตอบจากแบบสอบถามเหล่านี้ไปแปลงเป็นเวกเตอร์ (Vector) เพื่อให้ AI อัลกอริทึมคำนวณหาระยะห่าง (Cosine Similarity) และจับคู่อุปกรณ์ที่ตรงกับผู้ใช้รายบุคคลมากที่สุด
- ☐ **Gap Filling:** เติมเต็มช่องว่างที่เว็บไซต์อย่าง ProSettings ขาดไป คือ "การนำข้อมูลดิบมาประมวลผลให้ผู้ใช้" และอุดช่องโหว่ของเว็บไซต์ E-commerce ที่ขาดความเข้าใจในสเปคเชิงลึกของอุปกรณ์เกมมิ่ง
- ☐ **Scientific & Modern UX:** นำเสนอการวิเคราะห์ที่เป็นวิทยาศาสตร์ (Scientific Approach) ผ่านกระบวนการคำนวณของ AI พร้อมแสดงผลลัพธ์ในรูปแบบคะแนนความเข้ากันได้ (Match Score %) ซึ่งเข้าใจง่ายและใช้งานได้จริงสำหรับเกมเมอร์ทุกระดับ ตั้งแต่มือใหม่ไปจนถึงระดับฮาร์ดคอร์

บทที่ 3

การออกแบบระบบ

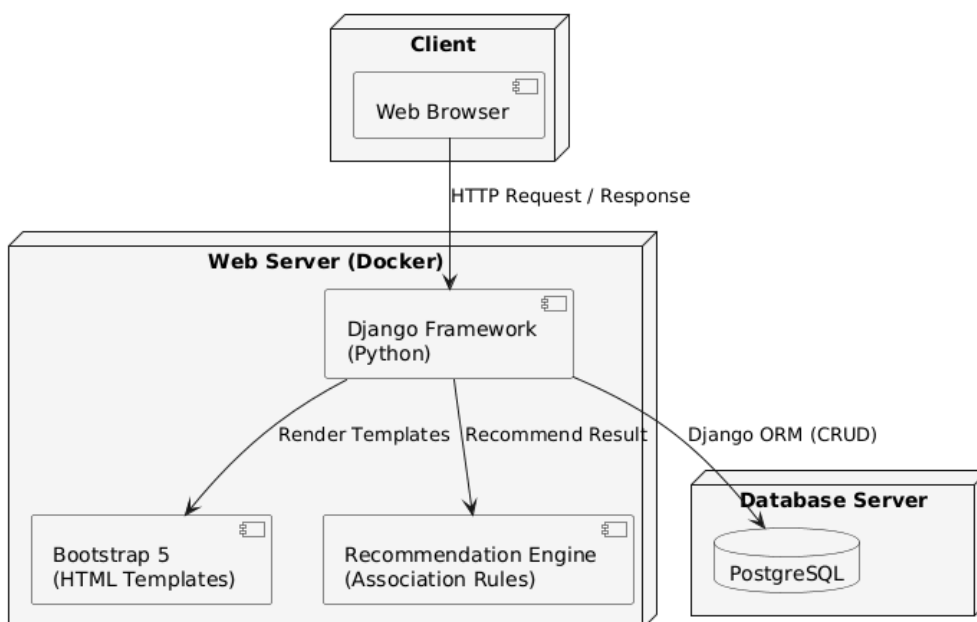
3.1 System architecture

ระบบ Gaming Gear Matcher ถูกออกแบบโดยใช้สถาปัตยกรรมแบบ Client-Server โดยมีการแบ่งส่วนการทำงานออกเป็น 3 ส่วนหลัก ได้แก่:

1. **Client Side (Front-end):** ส่วนผู้ใช้ มีการพัฒนาด้วย HTML5, CSS3 และ JavaScript โดยใช้ Framework Bootstrap 5 เพื่อความสวยงามและการแสดงผลที่รองรับทุกอุปกรณ์ (Responsive Design)
2. **Server Side (Back-end):** ส่วนประมวลผลหลัก พัฒนาโดยใช้ภาษา Python ร่วมกับ Django Web Framework ซึ่งทำหน้าที่จัดการ Logic การทำงาน, การเชื่อมต่อฐานข้อมูล
3. **Database:** ระบบฐานข้อมูลแบบ Relational Database Management System (RDBMS) โดยเลือกใช้ PostgreSQL สำหรับจัดเก็บข้อมูลผู้ใช้งาน, ข้อมูลอุปกรณ์เกมมิ่ง, และข้อมูลโปรไฟล์เเยอร์
4. **Recommendation Engine:** ใช้ Association Rules และ Hybrid Recommendation ในการแนะนำอุปกรณ์

โดยระบบทำงานอยู่บน Docker Container เพื่อความสะดวกในการติดตั้งและย้ายระบบ (Deployment)

System Architecture - Gaming Gear Matcher



รูปภาพที่ 15 แสดง System Architecture

3.2 System requirement

Functional Requirements (ความต้องการด้านฟังก์ชัน):

รหัส รายละเอียด

- FR-01 ระบบต้องอนุญาตให้ผู้เยี่ยมชม (Guest) สมัครสมาชิกได้ โดยใช้อีเมลและรหัสผ่านมาตรฐาน
- FR-02 ระบบต้องมีฟังก์ชันเข้าสู่ระบบ (Login) และออกจากระบบ (Logout)
- FR-03 สมาชิกต้องสามารถแก้ไขข้อมูลส่วนตัวและรูปภาพโปรไฟล์ได้
- FR-04 ระบบต้องมีการยืนยันตัวตน (Authentication) ก่อนเข้าถึงฟังก์ชันเฉพาะสมาชิก
- FR-05 ผู้ใช้ต้องสามารถเลือกอุปกรณ์แยกตามหมวดหมู่
- FR-06 ระบบต้องจำกัดการเลือกอุปกรณ์ให้เลือกได้เพียง 1 ชิ้นต่อ 1 ประเภทในเซสชันปัจจุบัน
- FR-07 หากผู้ใช้เลือกอุปกรณ์ประเภทเดิมซ้ำ ระบบต้องแจ้งเตือนว่าจะแทนที่ชิ้นเดิมหรือไม่
- FR-08 ระบบต้องแนะนำอุปกรณ์ชิ้นถัดไปโดยอ้างอิงจากฐานข้อมูล Pro Player
- FR-09 สมาชิกต้องสามารถบันทึกรายการอุปกรณ์ที่เลือกไว้เป็น "Preset" ได้
- FR-10 สมาชิกต้องตั้งชื่อ Preset ก่อนบันทึก
- FR-11 สมาชิกสามารถดูรายการ Preset ของตนเองได้ (My Presets)
- FR-12 สมาชิกสามารถแก้ไขชื่อ Preset หรือลบ Preset ที่บันทึกไว้ได้
- FR-13 สมาชิกสามารถเปลี่ยนอุปกรณ์บางชิ้นใน Preset เดิมได้
- FR-14 สมาชิกสามารถแชร์ Preset ผ่านลิงก์ ให้ผู้อื่นดูได้
- FR-15 สมาชิกสามารถให้คะแนน (Star Rating 1-5 ดาว) กับ Preset ที่ตนเองจัดได้
- FR-16 สมาชิกสามารถเขียนความคิดเห็น (Comment) เกี่ยวกับ Preset ได้
- FR-17 ระบบต้องบันทึกคะแนนและแสดงผลในหน้า Preset Detail
- FR-18 Admin สามารถจัดการ (เพิ่ม/ลบ/แก้ไข) ข้อมูลอุปกรณ์เกมมิ่งได้

FR-19 Admin สามารถจัดการข้อมูล Pro Player และการจับคู่อุปกรณ์ของ Pro Player ได้

FR-20 Admin สามารถดูภาพรวมสถิติของระบบผ่าน Dashboard ได้

Non-Functional Requirements (ความต้องการที่ไม่ใช่ฟังก์ชัน):

รหัส	รายละเอียด
NFR-01	ระบบต้องประมวลผลการค้นหาและแนะนำอุปกรณ์ภายในเวลาไม่เกิน 3 วินาที
NFR-02	ระบบต้องรองรับการใช้งานพร้อมกัน (Concurrent Users) ได้อย่างน้อย 50 Users
NFR-03	การสลับหน้า (Page Transition) ต้องมีความรวดเร็วและตอบสนองทันที
NFR-04	ระบบต้องมีการสำรองข้อมูล (Data Backup) ป้องกันการสูญหาย
NFR-05	ระบบต้องทำงานได้ถูกต้องแม่นยำในการบันทึกข้อมูลอุปกรณ์ลง Preset
NFR-06	เว็บไซต์ต้องเป็น Responsive รองรับ Desktop และ Mobile
NFR-07	UI ต้องมีความทันสมัย สวยงาม และเข้าใจง่าย
NFR-08	รหัสผ่านต้องถูกเข้ารหัส ก่อนบันทึกลงฐานข้อมูล
NFR-09	สิทธิ์การเข้าถึงข้อมูล ต้องถูกจัดการอย่างถูกต้อง

3.3 การออกแบบหน้าจอ

ในส่วนนี้จะเป็นการแสดง ตัวอย่างการออกแบบหน้าจอหลักๆ ภายในระบบ

Home Page - Guest (01_Home_Guest)

Gaming Gear Matcher

FIND YOUR PERFECT GEAR

Select a category to start building your professional gaming setup

New! AI Playstyle Analysis
 Let our Hybrid AI find your perfect gear based on your gaming style.

Choose a category:

รูปภาพที่ 16 แสดงการออกแบบหน้าแรก

หน้าแรกของระบบสำหรับผู้ทั่วไปที่ยังไม่ได้ Login เข้าสู่ระบบ ทำหน้าที่เป็นจุดเริ่มต้นหลักในการใช้งานระบบ Gaming Gear Matcher โดยนำเสนอทางเลือกให้ผู้ใช้สามารถเริ่มค้นหาอุปกรณ์เกมมิ่งได้ทันที ประกอบด้วยโลโก้ระบบ, เมื่อนำทาง (Home, Search), และปุ่มสำหรับเข้าสู่ระบบ (Login) หรือสมัครสมาชิก (Register) ซึ่งในส่วนการแสดงผลหลักจะประกอบด้วย 1.ข้อความต้อนรับและสโลแกนของระบบ ซึ่งมีปุ่ม Call-to-Action (CTA) หลักคือ "Try AI Analysis" เพื่อให้ผู้ใช้เข้าถึงฟังก์ชันหลักได้ทันที (ระบบ Hybrid AI วิเคราะห์สไตล์การเล่นเกมนของผู้ใช้เพื่อแนะนำอุปกรณ์ที่เหมาะสมได้) 2.แสดงรายการอุปกรณ์ (การ์ด 5 ใบเรียงกันในแนวนอน แต่ละใบแทนหมวดหมู่อุปกรณ์ที่แตกต่างกัน คือ เมาส์, คีย์บอร์ด, หูฟัง, จอภาพ, เก้าอี้)

Wizard Quiz (07_Wizard_Quiz)

Gaming Gear Matcher
^Search...^
[Member Menu]

Quiz for Playstyle Analysis

Tell us about your gaming style.

Progress: 33% =

- What is your main game genre?**
 - ☐ FPS (Shooter)
 - ☐ MOBA / RTS
 - ☐ MMORPG / RPG
- What is your hand size?**
 - ☐ Small (< 17cm)
 - ☐ Medium (17-19cm)
 - ☐ Large (> 19cm)
- How do you hold your mouse?**
 - ☐ Palm Grip
 - ☐ Claw Grip
 - ☐ Fingertip Grip

< Back
Analyze & Recommend >

รูปภาพที่ 17 แสดงการออกแบบหน้าจอ Quiz

หน้าสำหรับเก็บข้อมูลพฤติกรรมการเล่นเกมของผู้ใช้ เพื่อส่งให้ระบบ Hybrid AI วิเคราะห์และแนะนำอุปกรณ์ที่เหมาะสมที่สุด

ประกอบไปด้วย 1.แถบแสดงความคืบหน้าของการทำแบบสอบถาม 2.คำถาม 3 ข้อ ดังนี้
คำถามที่ 1: "What is your main game genre?" — เลือกแนวเกมหลักที่เล่น

FPS (Shooter) — เกมยิงมุมมองบุคคลที่หนึ่ง

MOBA / RTS — เกมวางแผนแบบ Real-time

MMORPG / RPG — เกมสวมบทบาท

คำถามที่ 2: "What is your hand size?" — ขนาดมือของผู้ใช้

Small (< 17cm)

Medium (17-19cm)

Large (> 19cm)

คำถามที่ 3: "How do you hold your mouse?" — ท่าจับเมาส์

Palm Grip — จับแบบคว่ำมือ

Claw Grip — จับแบบขีดปากกา

Fingertip Grip — จับด้วยปลายนิ้ว

Matching Result (09_Matching_Result)

Gaming Gear Matcher

YOUR GEAR RECOMMENDATIONS

AI Score: 95/100 Review Sentiment: Positive (+0.8)
 Reasons: Match Spec, Popular with Pros

My Setup (Customized)
 [Image] Mouse: Logitech G Pro X [X]
 [Image] Keyboard: Wooting 60HE [X]

AI Recommended Presets
☒ Performance ☐ Balanced ☐ Pro Choice
AI Analysis: Best for FPS, low latency focus.


Gear	Performance	Balanced	Pro Choice
Mouse	G Pro X	Viper V3	Zowie EC2
Keyboard	Wooting	Apex Pro	G915

รูปภาพที่ 18 แสดงการออกแบบหน้าผลลัพธ์การแนะนำ

หน้าแสดงผลลัพธ์สุดท้ายของการจับคู่อุปกรณ์ ซึ่งเป็นหน้าที่สำคัญที่สุดของระบบ ผู้ใช้สามารถเปรียบเทียบ Preset จาก AI, ปรับแต่ง Setup ของตัวเอง, บันทึกเป็น Preset, หรือแชร์ให้ผู้อื่นได้

Gear Detail (17_Gear_Detail)

Gaming Gear Matcher


Logitech G Pro X Superlight

Category: MOUSE
 Price: \$159.00

Description
 Ultra-lightweight wireless gaming mouse designed for esports pros.

Specifications

Spec	Value
Weight	63g
Sensor	HERO 25K
DPI	25,600
Battery	70 Hours

Other Gears in Same Category

รูปภาพที่ 19 แสดงการออกแบบหน้ารายละเอียดอุปกรณ์

หน้าแสดงข้อมูลเชิงลึกของอุปกรณ์เกมมิ่งแต่ละชิ้น ผู้ใช้สามารถดูรายละเอียด, ซื้อบน Amazon, หรือเพิ่มเข้า Setup ได้ และรายการอุปกรณ์อื่นๆ ในหมวดหมู่เดียวกัน (เช่น เมาส์ตัวอื่น) แสดงเป็น Card เรียงแถว (2 คอลัมน์) หากต้องการดูอุปกรณ์รุ่นอื่น

Manage Presets (13_Manage_Presets)

Gaming Gear Matcher
^Search...^
[Profile]
[Logout]

MY SAVED PRESETS

FPS Pro Setup	Saved: 15 Jan 2025	Delete
3 Gears in this Preset		
MOBA Balanced Build	Saved: 10 Jan 2025	Delete
4 Gears in this Preset		
MMO Budget Set	Saved: 05 Jan 2025	Delete
2 Gears in this Preset		

(You haven't saved any presets yet.)

Start Matching

รูปภาพที่ 20 แสดงการออกแบบหน้าจอการจัดการ setup

หน้ารวม Preset ทั้งหมดที่ผู้ใช้บันทึกไว้ ใช้สำหรับเข้าดูรายละเอียดหรือลบ Preset และในกรณีที่ไม่มี Preset ที่จัดไว้เรียบร้อยแล้ว จะแสดงกล่องเส้นประพร้อมข้อความ "You haven't saved any presets yet." และปุ่ม "Start Matching" เพื่อเริ่มสร้าง Preset ใหม่

Admin Dashboard (19_Admin_Dashboard)

Admin Panel
Dashboard
Pro Players
Users
Gaming Gears

Dashboard

Total Users: **120** Total Pro Players: **45** Total Gears: **320** Banned Users: **3**

Activity Overview (Last 30 Days)

Line Chart: New Users & Presets Created ===

User Presets
Admin Logs
User Feedbacks

Preset Name	Creator	Status	Created
FPS Pro	user1	Completed	Feb 10
MMO Set	user2	Completed	Feb 08

System Status

Server Time: 15:00:00
Database: Connected
Django: 5.x

Quick Actions

Manage Users
Presets

รูปภาพที่ 21 แสดงการออกแบบหน้าจอ Dashboard สำหรับ Admin
หน้าหลักสำหรับผู้ดูแลระบบ แสดงภาพรวมสถานะของระบบทั้งหมด ทั้งจำนวนผู้ใช้, อุปกรณ์, กิจกรรมล่าสุด, และสถานะระบบ และมี ปุ่มลัด 2 ปุ่ม: 1.Manage Users → ไปหน้าจัดการ Users 2.Presets → ไปหน้าจัดการ Presets

3.4 Use case diagram

แผนภาพกรณีการใช้งานถูกจัดทำขึ้นเพื่อแสดงขอบเขตของระบบ (System Boundary) และปฏิสัมพันธ์ระหว่างตัวแสดงแทน (Actors) กับฟังก์ชันการทำงานต่าง ๆ โดยจะแบ่งกลุ่มผู้ใช้งานออกเป็น 3 กลุ่มหลัก ได้แก่ ผู้เยี่ยมชม (Guest), สมาชิก (Member) และผู้ดูแลระบบ (Admin)

1. สัญลักษณ์ที่ใช้ในแผนภาพ (Symbol Descriptions)

เพื่อให้เกิดความเข้าใจที่ตรงกันในการอ่านแผนภาพ จึงมีการกำหนดความหมายของสัญลักษณ์ที่ใช้ตามมาตรฐาน UML ดังนี้

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมาย (Description)
	ตัวแสดงแทน (Actor)	แทนบทบาทของผู้ใช้งานภายนอกที่มีปฏิสัมพันธ์กับระบบ (Guest, Member, Admin)
	กรณีการใช้งาน (Use Case)	แทนฟังก์ชันหรือกระบวนการทำงานภายในระบบที่ให้ผลลัพธ์แก่ผู้ใช้งาน
	ขอบเขตระบบ (System Boundary)	เส้นกรอบสี่เหลี่ยมที่กำหนดขอบเขตระหว่างสิ่งที่อยู่ภายในระบบและภายนอกระบบ
	เส้นสมาคม (Association)	เส้นที่เชื่อมโยงระหว่าง Actor และ Use Case เพื่อแสดงว่ามีส่วนเกี่ยวข้องกัน
	เส้นลูกศร (Arrow Line)	การสืบทอดสิทธิ์ (Generalization)
	ความสัมพันธ์แบบรวม	แสดงกรณีการใช้งานที่ "จำเป็น" ต้องทำเป็นส่วนหนึ่งของกรณีการใช้งานหลักเสมอ
	ความสัมพันธ์แบบส่วนขยาย	แสดงกรณีการใช้งานที่เป็น "ทางเลือก" หรือส่วนเสริมจากกรณีการใช้งานหลัก

ตารางที่ 2 แสดงคำอธิบายแผนภาพ Use Case Diagram

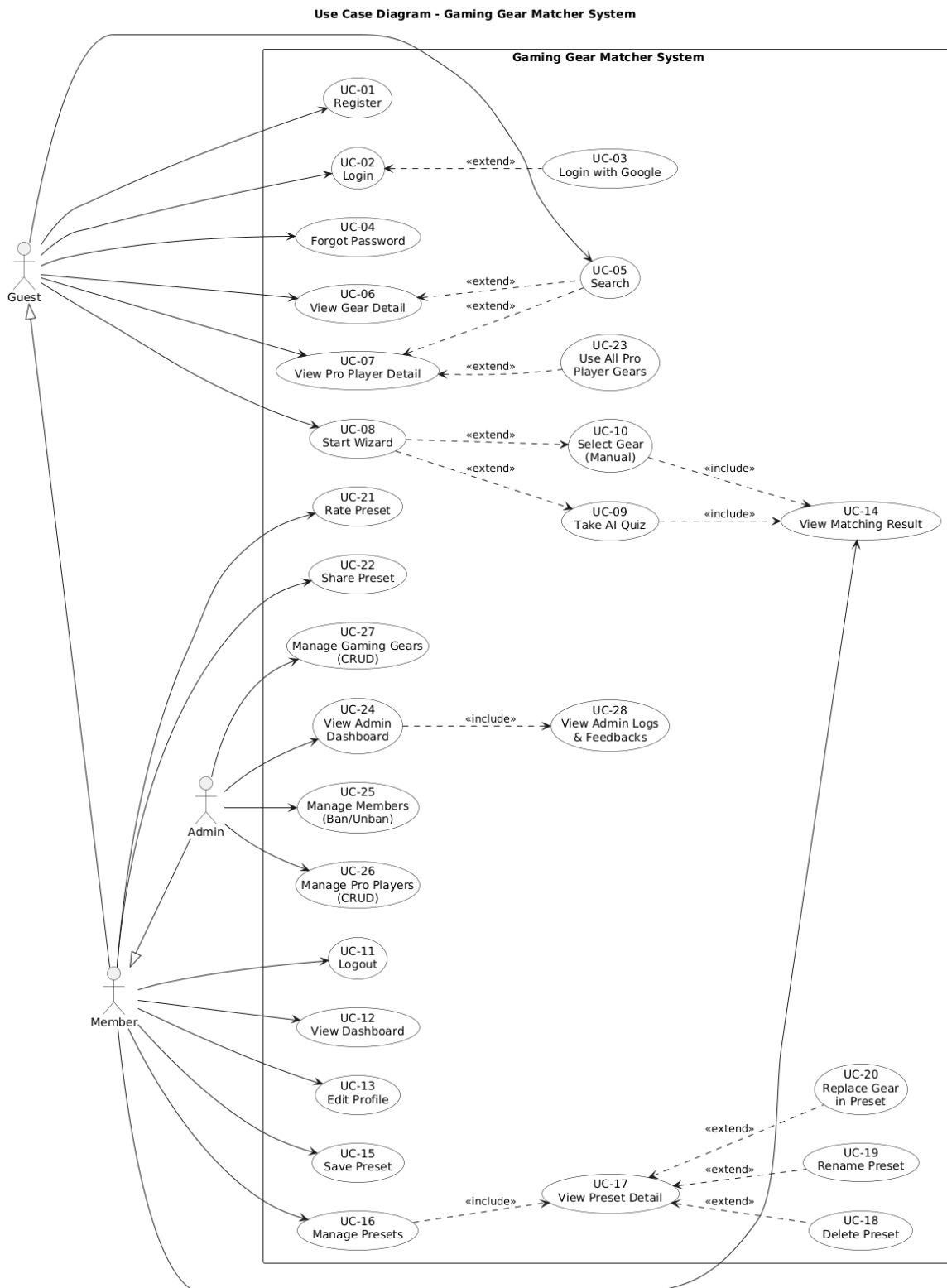
2. รายละเอียดกรณีการใช้งานของระบบ (Use Case Grouping)

จากการวิเคราะห์ระบบ Gaming Gear Matcher สามารถระบุกลุ่ม Use Case หลักที่สัมพันธ์กับ Actor แต่ละประเภทได้ดังนี้:

กลุ่มกรณีการใช้งานสำหรับผู้เยี่ยมชม (Guest Use Cases): ครอบคลุมฟังก์ชันพื้นฐานที่บุคคลทั่วไปสามารถเข้าถึงได้โดยไม่ต้องผ่านการยืนยันตัวตน ได้แก่ การเรียกดูหน้าหลักพร้อมหมวดหมู่อุปกรณ์ (Mouse, Keyboard, Headset, Monitor, Chair) การค้นหาข้อมูลอุปกรณ์เกมมิ่งและนักกีฬา Esports การดูรายละเอียดเชิงลึกของอุปกรณ์แต่ละชิ้นและข้อมูล Pro Player การเข้าสู่ระบบ Wizard เพื่อทำแบบสอบถามวิเคราะห์สไตล์การเล่น (AI Playstyle Analysis) หรือเลือกอุปกรณ์ด้วยตนเองแบบ Manual Selection รวมถึงกระบวนการลงทะเบียนสมาชิก การเข้าสู่ระบบ (รองรับ Google OAuth) และการรีเซ็ตรหัสผ่าน เพื่อรับสิทธิ์การใช้งานที่สูงขึ้น

กลุ่มกรณีการใช้งานสำหรับสมาชิก (Member Use Cases): ครอบคลุมฟังก์ชันการจัดการ Preset อุปกรณ์ที่ตนเองเป็นเจ้าของ ตั้งแต่การบันทึกชุดอุปกรณ์จากผลลัพธ์การจับคู่ (Save Preset) การดูรายละเอียด การเปลี่ยนชื่อ การแชร์ลิงก์ และการลบ Preset การเปลี่ยนอุปกรณ์ภายใน Preset (Replace Gear) การให้คะแนนและแสดงความคิดเห็น (Star Rating & Comment) การจัดการโปรไฟล์ส่วนตัว (แก้ไขข้อมูล อับโหลดรูปโปรไฟล์) การเรียกดู Dashboard แสดงสถิติการใช้งาน (จำนวน Preset, Rating, กราฟ Timeline) รวมถึงการใช้ระบบ Hybrid AI Recommendation เพื่อรับคำแนะนำชุดอุปกรณ์ 3 แบบ (Performance, Balanced, Pro Choice) พร้อมตารางเปรียบเทียบ

กลุ่มกรณีการใช้งานสำหรับผู้ดูแลระบบ (Admin Use Cases): ครอบคลุมการกำกับดูแลความเรียบร้อยของระบบในภาพรวม ผ่าน Dashboard ที่แสดงสถิติสำคัญ (จำนวนผู้ใช้, Pro Player, อุปกรณ์, ผู้ถูกระงับ) พร้อมกราฟ Activity Overview และตารางกิจกรรมล่าสุด (User Presets, Admin Logs, User Feedbacks) การจัดการบัญชีผู้ใช้งาน (การแก้ไขข้อมูล การแบนหรือปลดแบน) การจัดการข้อมูล Pro Player (เพิ่ม แก้ไข ลบ พร้อมข้อมูลอุปกรณ์ที่ใช้) การจัดการข้อมูลอุปกรณ์เกมมิ่ง (เพิ่ม แก้ไข ลบ พร้อมรายละเอียดสเปค) และการตรวจสอบสถานะระบบ (Database Status, Django Version) เพื่อรักษาความเสถียรและความปลอดภัยของระบบ



รูปภาพที่ 22 แสดง Use Case Diagram ของระบบ

ในแผนภาพที่ 3.4 แสดงให้เห็นถึงขอบเขตของระบบที่ถูกแยกออกจากผู้ใช้งานอย่างชัดเจน โดยมีความสัมพันธ์แบบสืบทอด (Generalization) ระหว่าง Actor กลุ่ม Guest ไปยัง Member และ Admin เพื่อลดความซับซ้อนของเส้นความสัมพันธ์ และมีการระบุเงื่อนไขการเข้าถึงข้อมูลตามระดับสิทธิ์ที่ถูกกำหนดไว้ในระบบ (Role-based Access Control)

3. ตารางสรุปภาพรวมกรณีการใช้งาน (Use Case Summary Table) รายละเอียดหน้าที่และตัวแสดงแทนที่เกี่ยวข้องในแต่ละกรณีการใช้งาน ถูกสรุป ดังนี้:

Use Case ID	กรณีการใช้งาน (Use Case Name)	ตัวแสดงแทน (Actors)	คำอธิบาย (Description)
UC-01	ลงทะเบียนสมาชิก	Guest	การสมัครบัญชีใหม่ด้วยฟอร์มลงทะเบียน (Username, Email, Password) เพื่อเข้าใช้งานระบบ
UC-02	เข้าสู่ระบบ	Guest	การยืนยันตัวตนด้วย Username และ Password เพื่อเข้าถึงสิทธิ์ตามบทบาท
UC-03	เข้าสู่ระบบผ่าน Google	Guest	การยืนยันตัวตนผ่าน Google OAuth เพื่อ Login หรือ Register โดยไม่ต้องกรอกฟอร์ม (Extend)
UC-04	รีเซ็ตรหัสผ่าน	Guest	การขอลิงก์รีเซ็ตรหัสผ่านผ่านอีเมลที่ลงทะเบียนไว้ (Django Allauth)
UC-05	ค้นหาข้อมูล	Guest	การค้นหาข้อมูลอุปกรณ์เกมมิ่งและนักกีฬา Esports แบบรวม (Global Search)
UC-06	ดูรายละเอียดอุปกรณ์	Guest	การเข้าดูข้อมูลเชิงลึก สเปค ราคา และลิงก์ซื้อของอุปกรณ์แต่ละชิ้น
UC-07	ดูรายละเอียด Pro Player	Guest	การเข้าดูข้อมูลนักกีฬา Esports และรายการอุปกรณ์ที่ใช้

UC-08	เริ่มต้น Wizard	Guest	การเข้าสู่ระบบแนะนำอุปกรณ์แบบ Wizard เพื่อเลือกวิธีค้นหาอุปกรณ์
UC-09	ทำแบบสอบถาม AI	Guest	การตอบคำถาม 3 ข้อ (แนวเกม, ขนาดมือ, ท่าจับเมาส์) เพื่อให้ Hybrid AI วิเคราะห์และแนะนำชุดอุปกรณ์
UC-10	เลือกอุปกรณ์ด้วยตนเอง	Guest	การเลือกอุปกรณ์จากหมวดหมู่ (Mouse, Keyboard, Headset) แบบ Manual Selection
UC-11	ออกจากระบบ	Member	การยกเลิกเซสชันเพื่อออกจากระบบ
UC-12	ดู Dashboard	Member	การเข้าสู่สถิติการใช้งาน (จำนวน Preset, Rating, กราฟ Timeline, กิจกรรมล่าสุด)
UC-13	แก้ไขโปรไฟล์	Member	การปรับปรุงรูปโปรไฟล์ ชื่อผู้ใช้ และอีเมล
UC-14	ดูผลลัพธ์การจับคู่	Member	การเข้าสู่ผลลัพธ์การแนะนำอุปกรณ์จาก AI พร้อมตารางเปรียบเทียบ Preset 3 แบบ
UC-15	บันทึก Preset	Member	การบันทึกชุดอุปกรณ์ที่เลือกไว้เป็น Preset ลงฐานข้อมูล
UC-16	จัดการ Preset	Member	การเข้าดูรายการ Preset ทั้งหมดที่ตนเองเป็นเจ้าของ
UC-17	ดูรายละเอียด Preset	Member	การเข้าดูรายการอุปกรณ์ในเซต คะแนน และความคิดเห็น
UC-18	ลบ Preset	Member	การลบ Preset ที่ไม่ต้องการออกจากระบบ

UC-19	เปลี่ยนชื่อ Preset	Member	การแก้ไขชื่อ Preset ผ่าน Modal ในหน้ารายละเอียด
UC-20	เปลี่ยนอุปกรณ์ใน Preset	Member	การแทนที่อุปกรณ์ชิ้นใดชิ้นหนึ่งด้วยอุปกรณ์อื่นในประเภทเดียวกัน (Extend)
UC-21	ให้คะแนน Preset	Member	การให้คะแนนดาว (1-5) พร้อมเขียนความคิดเห็นเกี่ยวกับ Preset
UC-22	แชร์ Preset	Member	การสร้างลิงก์สำหรับแชร์ Preset ให้ผู้อื่นดูได้
UC-23	ใช้อุปกรณ์ทั้งหมดของ Pro Player	Member	การคัดลอกชุดอุปกรณ์ทั้งหมดของนักกีฬา Esports มาใส่ใน Setup ของตนเอง (Extend)
UC-24	ดูแผงควบคุมแอดมิน	Admin	การเข้าถึง Dashboard แสดงสถิติภาพรวม กราฟกิจกรรม และสถานะระบบ
UC-25	จัดการสมาชิก (แบน/ปลดแบน)	Admin	การระงับหรือคืนสิทธิ์บัญชีผู้ใช้ที่กระทำผิดกฎ
UC-26	จัดการ Pro Player (CRUD)	Admin	การเพิ่ม แก้ไข และลบข้อมูลนักกีฬา Esports พร้อมอุปกรณ์ที่ใช้
UC-27	จัดการอุปกรณ์เกมมิ่ง (CRUD)	Admin	การเพิ่ม แก้ไข และลบข้อมูลอุปกรณ์เกมมิ่ง พร้อมรายละเอียดสเปค
UC-28	ดูบันทึกกิจกรรมและ Feedback	Admin	การตรวจสอบ Admin Logs, User Presets และ User Feedbacks ผ่าน Dashboard

รายละเอียดกรณีการใช้งาน (Use Case Descriptions)

ข้อมูลรายละเอียดกรณีการใช้งานสำหรับระบบ Gaming Gear Matcher ถูกจัดทำขึ้นในรูปแบบตาราง เพื่ออธิบายลำดับขั้นตอนการทำงาน ปฏิสัมพันธ์ระหว่างตัวแสดงแทนและระบบ รวมถึงเงื่อนไขต่าง ๆ อย่างเป็นระบบ ดังนี้

ส่วนที่ 1: Guest (ผู้เยี่ยมชม)

UC-01: ลงทะเบียนสมาชิก (Register)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-01: ลงทะเบียนสมาชิก (Register)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานยังไม่มีบัญชีในระบบ Gaming Gear Matcher
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานเข้าสู่หน้าลงทะเบียนผ่านปุ่ม "Register" 2. ผู้ใช้งานกรอกข้อมูลส่วนตัว: ชื่อผู้ใช้ (Username), อีเมล (Email), รหัสผ่าน (Password), และยืนยันรหัสผ่าน 3. ระบบตรวจสอบความถูกต้องของข้อมูล (ชื่อผู้ใช้ไม่ซ้ำ, อีเมลถูกต้อง, รหัสผ่านตรงกัน) 4. หากข้อมูลถูกต้อง ระบบบันทึกบัญชีผู้ใช้ใหม่ลงฐานข้อมูลและสร้างโปรไฟล์เริ่มต้น 5. ระบบนำผู้ใช้งานไปยังหน้า Login หรือหน้าแรกพร้อมข้อความต้อนรับ
เงื่อนไขหลังจบ	บัญชีผู้ใช้งานถูกสร้างและบันทึกลงฐานข้อมูล

UC-02: เข้าสู่ระบบ (Login)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-02: เข้าสู่ระบบ (Login)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานมีบัญชีในระบบแล้ว

หัวข้อ	รายละเอียด
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานเข้าสู่หน้าเข้าสู่ระบบผ่านปุ่ม "Login" 2. ผู้ใช้งานกรอกชื่อผู้ใช้หรืออีเมล และรหัสผ่านที่ลงทะเบียนไว้ 3. ระบบตรวจสอบข้อมูลยืนยันตัวตน (Authentication) 4. หากถูกต้อง ระบบสร้าง Session และนำผู้ใช้งานไปยังหน้าสมาชิก (Member Home)
เงื่อนไขหลังจบ	ผู้ใช้งานอยู่ในสถานะเข้าสู่ระบบ (Authenticated)

UC-03: เข้าสู่ระบบผ่าน Google (Login with Google)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-03: เข้าสู่ระบบผ่าน Google (Login with Google)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานมีบัญชี Google
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Sign in with Google" ในหน้า Login หรือ Register 2. ระบบเปลี่ยนเส้นทางไปยังหน้ายืนยันตัวตนของ Google (OAuth) 3. ผู้ใช้งานอนุญาตให้สิทธิ์เข้าถึงข้อมูลพื้นฐาน (Email, Profile) 4. ระบบรับข้อมูลกลับมาตรวจสอบ: <ul style="list-style-type: none"> - หากมีบัญชีแล้ว: ทำการ Login ทันที - หากยังไม่มีบัญชี: สร้างบัญชีใหม่ด้วยอีเมลจาก Google แล้ว Login
เงื่อนไขหลังจบ	ผู้ใช้งานเข้าสู่ระบบสำเร็จโดยไม่ต้องใช้รหัสผ่านของระบบ

UC-04: รีเซ็ตรหัสผ่าน (Forgot Password)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-04: รีเซ็ตรหัสผ่าน (Forgot Password)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานจำรหัสผ่านไม่ได้ แต่จำอีเมลได้
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานเลือก "Forgot Password?" ในหน้า Login 2. ผู้ใช้งานระบุอีเมลที่ใช้ลงทะเบียน 3. ระบบส่งลิงก์สำหรับตั้งรหัสผ่านใหม่ไปยังอีเมล

หัวข้อ	รายละเอียด
	4. ผู้ใช้งานคลิกลิงก์และกำหนดรหัสผ่านใหม่
	5. ระบบบันทึกการรหัสผ่านใหม่แทนที่รหัสผ่านเดิม
เงื่อนไขหลังจบ	รหัสผ่านเดิมถูกยกเลิก และสามารถใช้รหัสผ่านใหม่ Login ได้

UC-05: ค้นหาข้อมูล (Search)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-05: ค้นหาข้อมูล (Search)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานต้องการค้นหาอุปกรณ์หรือโปรเพลเยอร์
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานกรอกคำค้นหาในช่อง Search Bar บน Navbar 2. ระบบประมวลผลคำค้นหาจากฐานข้อมูล Gaming Gear และ Pro Player 3. ระบบแสดงผลลัพธ์แยกเป็น "Pro Players" และ "Gaming Gear" ในรูปแบบการ์ด 4. หากไม่พบข้อมูล ระบบแสดงข้อความแจ้งเตือน
เงื่อนไขหลังจบ	ผลลัพธ์การค้นหาถูกแสดงบนหน้าจอ

UC-06: ดูรายละเอียดอุปกรณ์ (View Gear Detail)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-06: ดูรายละเอียดอุปกรณ์ (View Gear Detail)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเลือกอุปกรณ์จากรายการหรือผลค้นหา
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกที่การ์ดอุปกรณ์ 2. ระบบแสดงรายละเอียดอุปกรณ์: รูปภาพ, ชื่อรุ่น, ยี่ห้อ, ราคา, สเปค, และลิงก์สั่งซื้อ (Amazon) 3. ระบบแสดงรายการอุปกรณ์อื่นในหมวดเดียวกัน (Recommendation) 4. (ทางเลือก) หากอยู่ใน Wizard Flow สามารถกดปุ่ม Add เพื่อเลือกอุปกรณ์ได้
เงื่อนไขหลังจบ	รายละเอียดอุปกรณ์ถูกแสดงผลครบถ้วน

UC-07: ดูรายละเอียด Pro Player (View Pro Player Detail)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-07: ดูรายละเอียด Pro Player (View Pro Player Detail)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเลือกโปรเพลเยอร์จากรายการหรือผลค้นหา
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกที่การ์ดของ Pro Player 2. ระบบแสดงข้อมูลผู้เล่น: รูปภาพ, ชื่อ, เกมที่เล่น, ประวัติย่อ 3. ระบบแสดงรายการ Gaming Gear ที่ผู้เล่นคนนั้นใช้ 4. (ทางเลือก) สมาชิกสามารถกดปุ่ม "Use All" เพื่อนำชุดเกียร์ไปใช้ได้เลย
เงื่อนไขหลังจบ	ข้อมูลโปรเพลเยอร์และอุปกรณ์ที่ใช้ถูกแสดงผล

UC-08: เริ่มต้น Wizard (Start Wizard)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-08: เริ่มต้น Wizard (Start Wizard)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานต้องการหาอุปกรณ์ที่เหมาะสม
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Start Matching" หรือ "Analyze Playstyle" 2. ระบบแสดงหน้าจอเริ่มต้น Wizard พร้อม 2 ทางเลือก: <ul style="list-style-type: none"> - AI Playstyle Analysis (แนะนำด้วย AI) - Manual Selection (เลือกเองทีละชิ้น)
เงื่อนไขหลังจบ	ผู้ใช้งานเข้าสู่กระบวนการเลือกหรือวิเคราะห์อุปกรณ์

UC-09: ทำแบบสอบถาม AI (Take AI Quiz)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-09: ทำแบบสอบถาม AI (Take AI Quiz)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเลือกโหมด AI Analysis ใน UC-08
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานตอบคำถาม 3 ข้อเกี่ยวกับสไตล์การเล่น: <ul style="list-style-type: none"> - แนวเกมหลัก (FPS, MOBA, MMO)

หัวข้อ	รายละเอียด
	<ul style="list-style-type: none"> - ขนาดมือ (Small, Medium, Large) - รูปแบบการจับเมาส์ (Palm, Claw, Fingertip)
	2. ระบบบันทึกคำตอบและนำไปประมวลผลด้วย Hybrid Recommendation Engine
เงื่อนไขหลังจบ	ได้รับผลลัพธ์การแนะนำอุปกรณ์ที่เหมาะสม (UC-14)

UC-10: เลือกอุปกรณ์ด้วยตนเอง (Select Gear Manual)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-10: เลือกอุปกรณ์ด้วยตนเอง (Select Gear Manual)
ตัวแสดงแทน (Actors)	ผู้เยี่ยมชม (Guest)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเลือกโหมด Manual Selection ใน UC-08
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานเลือกหมวดหมู่ที่ต้องการ (Mouse, Keyboard, Headset) 2. ระบบแสดงรายการอุปกรณ์ในหมวดนั้น พร้อมตัวกรองเบื้องต้น 3. ผู้ใช้งานเลือกอุปกรณ์ที่ต้องการโดยกด "Select" 4. ทำซ้ำจนครบทุกชิ้นส่วนที่ต้องการ
เงื่อนไขหลังจบ	อุปกรณ์ที่เลือกถูกบันทึกลงใน Session (Temp Preset)

ส่วนที่ 2: Member (สมาชิก)

UC-11: ออกจากระบบ (Logout)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-11: ออกจากระบบ (Logout)
ตัวแสดงแทน (Actors)	สมาชิก (Member), ผู้ดูแลระบบ (Admin)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานอยู่ในระบบ
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Log out" จากเมนู Profile หรือ Sidebar 2. ระบบทำลาย Session ของผู้ใช้งาน 3. ระบบนำผู้ใช้งานกลับไปยังหน้า Home Guest
เงื่อนไขหลังจบ	ผู้ใช้งานออกจากระบบและกลับสู่สถานะ Guest

UC-12: ดู Dashboard (View Dashboard)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-12: ดู Dashboard (View Dashboard)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	<p>ผู้ใช้งานเข้าสู่ระบบแล้ว</p> <ol style="list-style-type: none"> 1. ผู้ใช้งานเข้าเมนู "My Profile" หรือ "Dashboard" 2. ระบบแสดงหน้า Dashboard ส่วนตัว
ลำดับขั้นตอนการทำงาน	<p>ประกอบด้วย:</p> <ul style="list-style-type: none"> - สถิติ (Total Presets, Ratings) - กราฟกิจกรรม (Timeline) - รายการ Preset ล่าสุด และ Rating ล่าสุด
เงื่อนไขหลังจบ	ข้อมูลสรุปการใช้งานถูกแสดงผล

UC-13: แก้ไขโปรไฟล์ (Edit Profile)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-13: แก้ไขโปรไฟล์ (Edit Profile)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	<p>ผู้ใช้งานอยู่ในหน้า Profile</p> <ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Edit Profile" 2. ผู้ใช้งานแก้ไขข้อมูล: รูปโปรไฟล์, ชื่อผู้ใช้, อีเมล 3. กดปุ่มบันทึก 4. ระบบอัปเดตข้อมูลในฐานข้อมูล
ลำดับขั้นตอนการทำงาน	
เงื่อนไขหลังจบ	<p>ข้อมูลส่วนตัวของผู้ใช้งานได้รับการเปลี่ยนแปลง</p>

UC-14: ดูผลลัพธ์การจับคู่ (View Matching Result)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-14: ดูผลลัพธ์การจับคู่ (View Matching Result)
ตัวแสดงแทน (Actors)	สมาชิก (Member), Guest
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานทำแบบสอบถาม AI (UC-09) หรือเลือกอุปกรณ์ครบ (UC-10)

หัวข้อ	รายละเอียด
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ระบบประมวลผลข้อมูลและแสดงหน้า Matching Result 2. แสดง "My Setup" (อุปกรณ์ที่เลือกเอง) 3. แสดง "AI Analysis & Recommendations" (3 แบบ: Performance, Balanced, Pro Choice) 4. แสดงข้อมูลเปรียบเทียบสเปคและราคา 5. หากเป็นสมาชิก จะมีปุ่ม Save Preset
เงื่อนไขหลังจบ	ผลลัพธ์และคำแนะนำอุปกรณ์ถูกแสดงบนหน้าจอ

UC-15: บันทึก Preset (Save Preset)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-15: บันทึก Preset (Save Preset)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานพอใจกับผลลัพธ์ในหน้า Matching Result
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Save This Preset" ที่ชุดอุปกรณ์ที่ต้องการ 2. ผู้ใช้งานตั้งชื่อ Preset ใน Modal 3. ระบบบันทึกรายการอุปกรณ์ทั้งหมดลงในฐานข้อมูล เชื่อมโยงกับบัญชีผู้ใช้ 4. ระบบแจ้งเตือน "Preset saved successfully"
เงื่อนไขหลังจบ	Preset ถูกบันทึกและสามารถเรียกดูได้ในภายหลัง

UC-16: จัดการ Preset (Manage Presets)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-16: จัดการ Preset (Manage Presets)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเข้าสู่ระบบ
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานเข้าเมนู "Manage Presets" 2. ระบบแสดงรายการ Preset ทั้งหมดที่ผู้ใช้เคยบันทึกไว้ 3. แต่ละรายการแสดงชื่อ, วันที่สร้าง, จำนวนอุปกรณ์ และปุ่มดำเนินการ (View, Delete)
เงื่อนไขหลังจบ	รายการ Preset ทั้งหมดถูกแสดงผล

UC-17: ดูรายละเอียด Preset (View Preset Detail)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-17: ดูรายละเอียด Preset (View Preset Detail)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเลือก Preset จากหน้ารายการ
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกที่ชื่อ Preset 2. ระบบแสดงรายละเอียด: รายชื่ออุปกรณ์ในเซต, ราคารวมโดยประมาณ 3. แสดงส่วนจัดการ: เปลี่ยนชื่อ, แชร, ให้คะแนน, หรือเปลี่ยนอุปกรณ์
เงื่อนไขหลังจบ	รายละเอียดของ Preset ถูกแสดงผล

UC-18: ลบ Preset (Delete Preset)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-18: ลบ Preset (Delete Preset)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานต้องการลบ Preset ที่ไม่ต้องการ
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานกดปุ่ม "Delete" ที่รายการ Preset 2. ระบบแสดงหน้าต่างยืนยันการลบ 3. ผู้ใช้งานยืนยัน 4. ระบบลบข้อมูล Preset ออกจากฐานข้อมูล
เงื่อนไขหลังจบ	Preset ถูกลบออกจากระบบอย่างถาวร

UC-19: เปลี่ยนชื่อ Preset (Rename Preset)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-19: เปลี่ยนชื่อ Preset (Rename Preset)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานอยู่ในหน้า Preset Detail

หัวข้อ	รายละเอียด
	1. ผู้ใช้งานคลิกไอคอนแก้ไขชื่อข้างชื่อ Preset
ลำดับขั้นตอนการทำงาน	2. กรอกชื่อใหม่ใน Modal 3. กดบันทึก
เงื่อนไขหลังจบ	4. ระบบอัปเดตชื่อ Preset ในฐานข้อมูล ชื่อ Preset เปลี่ยนแปลงตามที่ระบุ

UC-20: เปลี่ยนอุปกรณ์ใน Preset (Replace Gear in Preset)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-20: เปลี่ยนอุปกรณ์ใน Preset (Replace Gear in Preset)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานต้องการเปลี่ยนอุปกรณ์บางชิ้นใน Preset เดิม 1. ผู้ใช้งานคลิกปุ่ม "Replace" ที่รายการอุปกรณ์ชิ้นที่ต้องการเปลี่ยน
ลำดับขั้นตอนการทำงาน	2. ระบบแสดงหน้าต่างเลือกอุปกรณ์ใหม่ในหมวดเดียวกัน 3. ผู้ใช้งานกด "Select" ที่อุปกรณ์ชิ้นใหม่ 4. ระบบอัปเดต Preset โดยแทนที่อุปกรณ์เก่าด้วยอุปกรณ์ใหม่
เงื่อนไขหลังจบ	รายการอุปกรณ์ใน Preset ถูกปรับปรุง

UC-21: ให้คะแนน Preset (Rate Preset)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-21: ให้คะแนน Preset (Rate Preset)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานเป็นเจ้าของ Preset นั้น 1. ผู้ใช้งานให้คะแนนดาว (1-5 ดาว) ในหน้า Preset Detail
ลำดับขั้นตอนการทำงาน	2. เขียนความคิดเห็น (Comment) 3. กดส่ง 4. ระบบบันทึก Rating และ Comment ลงฐานข้อมูล
เงื่อนไขหลังจบ	คะแนนและความคิดเห็นถูกบันทึกและแสดงผล

UC-22: แชร์ Preset (Share Preset)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-22: แชร์ Preset (Share Preset)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานต้องการแบ่งปัน Preset ให้ผู้อื่น
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Share" 2. ระบบสร้างลิงก์สาธารณะสำหรับ Preset นั้น (Unique URL) 3. ผู้ใช้งานคัดลอกลิงก์ไปส่งต่อให้ผู้อื่น
เงื่อนไขหลังจบ	Preset สามารถเข้าถึงได้ผ่านลิงก์แชร์

UC-23: ใช้อุปกรณ์ทั้งหมดของ Pro Player (Use All Pro Player Gears)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-23: ใช้อุปกรณ์ทั้งหมดของ Pro Player (Use All Pro Player Gears)
ตัวแสดงแทน (Actors)	สมาชิก (Member)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานอยู่ในหน้า Pro Player Detail
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. ผู้ใช้งานคลิกปุ่ม "Use All" 2. ระบบคัดลอกอุปกรณ์ทั้งหมดที่ Pro Player คนนั้นใช้ มาสร้างเป็น Temp Preset ของผู้ใช้งาน 3. นำผู้ใช้งานไปยังหน้า Matching Result เพื่อตรวจสอบและบันทึก
เงื่อนไขหลังจบ	ได้รับชุดอุปกรณ์เหมือนกับ Pro Player พร้อมบันทึก

ส่วนที่ 3: Admin (ผู้ดูแลระบบ)

UC-24: ดูแผงควบคุมแอดมิน (View Admin Dashboard)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-24: ดูแผงควบคุมแอดมิน (View Admin Dashboard)
ตัวแสดงแทน (Actors)	ผู้ดูแลระบบ (Admin)
เงื่อนไขก่อนเริ่ม	ผู้ใช้งานมีสิทธิ์ Admin (Is Superuser)
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. Admin Login เข้าสู่ระบบ 2. ระบบตรวจสอบสิทธิ์และนำไปยัง Admin Dashboard

หัวข้อ	รายละเอียด
	3. ระบบแสดงสถิติรวม (Users, Gears, Pro Players), กราฟกิจกรรม, และสถานะระบบ

เงื่อนไขหลังจบ	Admin เห็นภาพรวมสถานะของระบบทั้งหมด
----------------	-------------------------------------

UC-25: จัดการสมาชิก (Manage Members - Ban/Unban)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-25: จัดการสมาชิก (Manage Members - Ban/Unban)
ตัวแสดงแทน (Actors)	ผู้ดูแลระบบ (Admin)
เงื่อนไขก่อนเริ่ม	Admin อยู่ในหน้าจัดการสมาชิก
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. Admin ค้นหาสมาชิกจากชื่อหรืออีเมล 2. กดปุ่ม "Ban" เพื่อระงับการใช้งาน หรือ "Unban" เพื่อคืนสิทธิ์ 3. ระบบอัปเดตสถานะ Active ของบัญชีผู้ใช้
เงื่อนไขหลังจบ	สถานะการใช้งานของสมาชิกถูกเปลี่ยนแปลงตามคำสั่ง

UC-26: จัดการ Pro Player (Manage Pro Players - CRUD)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-26: จัดการ Pro Player (Manage Pro Players - CRUD)
ตัวแสดงแทน (Actors)	ผู้ดูแลระบบ (Admin)
เงื่อนไขก่อนเริ่ม	Admin ต้องการเพิ่ม/ลบ/แก้ไขข้อมูลนักกีฬา
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. Admin เข้าเมนู Pro Players 2. กด "Add New" เพื่อเพิ่ม, "Edit" เพื่อแก้ไข, หรือ "Delete" เพื่อลบ 3. กรอกข้อมูลและเลือกอุปกรณ์ที่นักกีฬาใช้ 4. บันทึกข้อมูลลงฐานข้อมูล
เงื่อนไขหลังจบ	ข้อมูล Pro Player ถูกปรับปรุงให้เป็นปัจจุบัน

UC-27: จัดการอุปกรณ์เกมมิ่ง (Manage Gaming Gears - CRUD)

หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-27: จัดการอุปกรณ์เกมมิ่ง (Manage Gaming Gears - CRUD)
ตัวแสดงแทน (Actors)	ผู้ดูแลระบบ (Admin)
เงื่อนไขก่อนเริ่ม	Admin ต้องการเพิ่ม/ลบ/แก้ไขข้อมูลอุปกรณ์
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. Admin เข้าเมนู Gaming Gears 2. กด "Add New" เพื่อเพิ่ม, "Edit" เพื่อแก้ไข, หรือ "Delete" เพื่อลบ 3. กรอกข้อมูลสเปค, ราคา, ลิงก์รูปภาพ, ลิงก์ Amazon 4. บันทึกข้อมูลลงฐานข้อมูล
เงื่อนไขหลังจบ	ข้อมูลอุปกรณ์เกมมิ่งถูกปรับปรุง

UC-28: ดูบันทึกกิจกรรมและ Feedback (View Admin Logs & Feedbacks)

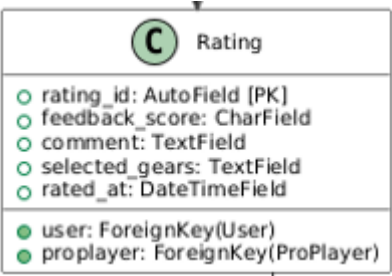


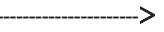
หัวข้อ	รายละเอียด
ชื่อกรณีการใช้งาน	UC-28: ดูบันทึกกิจกรรมและ Feedback (View Admin Logs & Feedbacks)
ตัวแสดงแทน (Actors)	ผู้ดูแลระบบ (Admin)
เงื่อนไขก่อนเริ่ม	Admin อยู่ในหน้า Dashboard
ลำดับขั้นตอนการทำงาน	<ol style="list-style-type: none"> 1. Admin คลิก Tab "Admin Logs" เพื่อดูประวัติการกระทำของผู้ดูแลระบบ 2. Admin คลิก Tab "User Feedbacks" เพื่อดูคะแนนและความคิดเห็นจากผู้ใช้งานเกี่ยวกับ Preset 3. ระบบแสดงข้อมูลในรูปแบบตารางเรียงตามเวลาล่าสุด
เงื่อนไขหลังจบ	Admin ได้รับข้อมูลตรวจสอบการใช้งานและผลตอบรับ

3.5 Class diagram

แผนภาพคลาส ถูกจัดทำขึ้นเพื่อแสดงโครงสร้างเชิงวัตถุ ของระบบ Gaming Gear Matcher โดยละเอียด ประกอบด้วยคลาส (Classes), คุณลักษณะ (Attributes), และพฤติกรรม (Methods) ของแต่ละส่วน ตลอดจนแสดงความสัมพันธ์ (Relationships) ระหว่างคลาสต่าง ๆ เพื่อให้เห็นภาพการทำงานของระบบในเชิงลึก

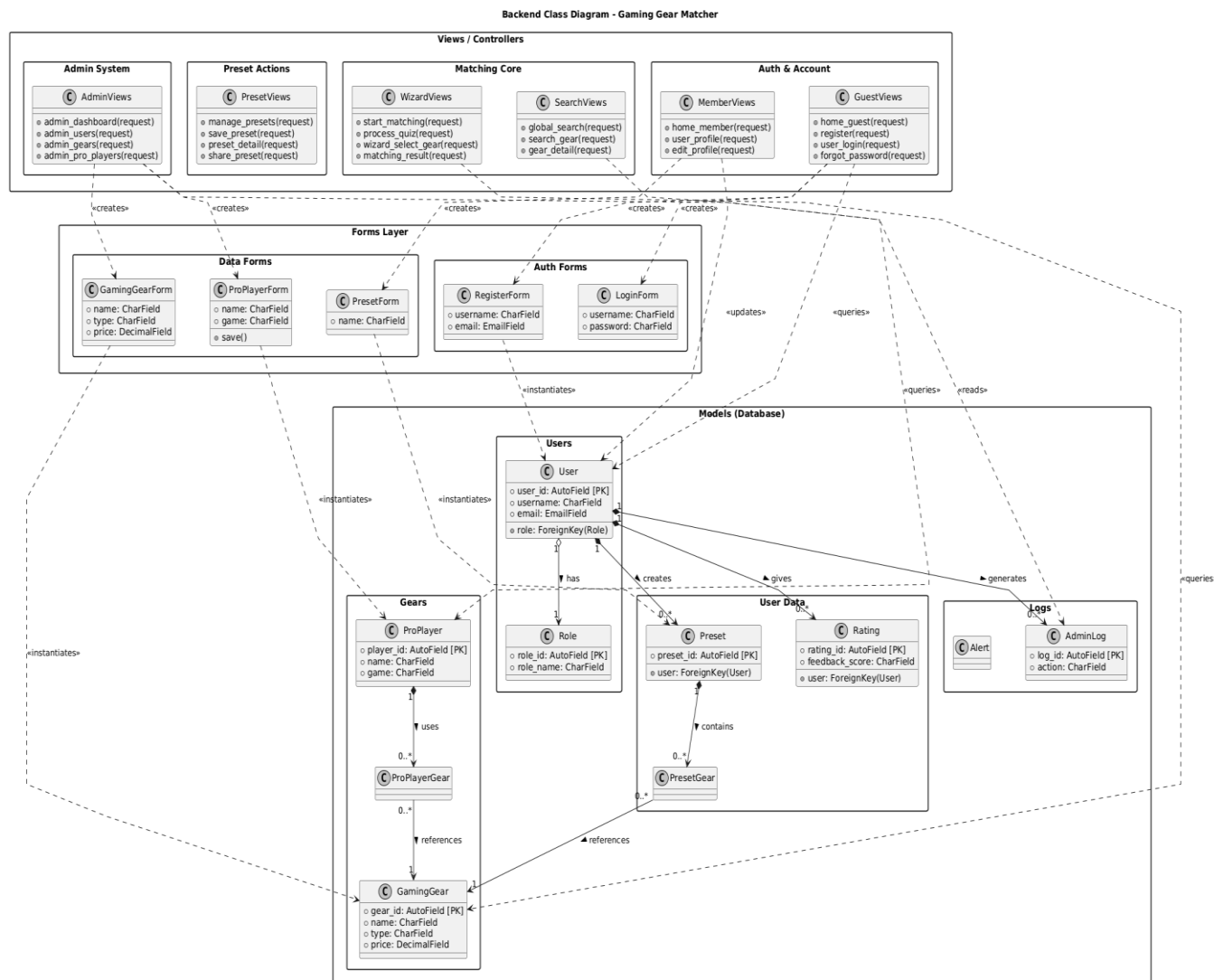
สัญลักษณ์ที่ใช้ในแผนภาพ (Symbol Descriptions)

เพื่อให้เกิดความเข้าใจที่ตรงกันในการอ่านแผนภาพ จึงมีการกำหนดความหมายของสัญลักษณ์ที่ใช้ตามมาตรฐาน UML ดังนี้

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมาย (Description)
 <p>(สี่เหลี่ยมแบ่งส่วน)</p>	Class	เป็นตัวแทนของโครงสร้างข้อมูลหรือส่วนประกอบในระบบ ภายในระบุชื่อคลาส (เช่น User, Preset), แอตทริบิวต์ (Attributes), และเมธอด (Methods)
 <p>(กรอบสี่เหลี่ยมใหญ่)</p>	Package	ใช้สำหรับจัดกลุ่มคลาสที่มีหน้าที่เกี่ยวข้องกันให้อยู่ในหมวดหมู่เดียวกัน เช่น กลุ่ม Models (Database), Forms, หรือ Views / Controllers
 <p>(เส้นทึบหัวข้าวหลามตัดทึบ)</p>	Composition	ความสัมพันธ์แบบ "เป็นส่วนหนึ่งของ" ที่มีความผูกพันสูง (Strong Ownership) หากคลาสหลักถูกลบ คลาสย่อยจะถูกลบไปด้วย (เช่น Preset เป็นเจ้าของ PresetGear)
 <p>(เส้นทึบหัวลูกศร)</p>	Association	ความสัมพันธ์แบบเชื่อมโยงหรืออ้างอิงถึงกัน (Reference) โดยทิศทางลูกศรชี้ไปยังคลาสที่

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมาย (Description)
		ถูกอ้างอิง (เช่น ProPlayerGear อ้างอิงไปยัง GamingGear)
-----> (เส้นประหัวลูกศร)	Dependency	ความสัมพันธ์แบบพึ่งพา ซึ่งคลาสหนึ่งมีการเรียกใช้งานอีกคลาสหนึ่ง เช่น Views สร้าง (<<creates>>) Forms หรือ Views ดึงข้อมูล (<<queries>>) จาก Models
1, 0.. (ตัวเลขกำกับเส้น)	Multiplicity	ตัวเลขระบุจำนวนความสัมพันธ์ระหว่างคลาส - 1: หนึ่งเท่านั้น - 0..: ศูนย์หรือมากกว่า (มีหรือไม่มีก็ได้) - 0..1: ศูนย์หรือหนึ่ง
<< text >> (ข้อความในเครื่องหมาย)	Stereotype	คำขยายความที่ระบุบนเส้นความสัมพันธ์เพื่อบอกลักษณะการทำงานเฉพาะเจาะจง เช่น <<creates>> (สร้าง), <<updates>> (อัปเดต), <<reads>> (อ่านข้อมูล)

ตารางที่ 3 แสดงคำอธิบายแผนภาพ Class Diagram



รูปภาพที่ 23 แสดง Class Diagram ส่วน Backend

แผนภาพคลาสส่วนหลัง (Back-end Class Diagram) แสดงโครงสร้างเชิงวัตถุของระบบฐานข้อมูลและลอจิกการทำงานฝั่งเซิร์ฟเวอร์ ออกแบบตามหลักการ **MVT (Model-View-Template)** โดยแบ่งเป็น 3 ส่วนสำคัญ:

2.1 Views / Controllers Layer (ส่วนจัดการคำขอ) เป็นจุดเริ่มต้นของการทำงานเมื่อมีคำขอ (Request) เข้ามายังเซิร์ฟเวอร์ ทำหน้าที่ประสานงานระหว่างข้อมูลและหน้าเว็บ

- **Auth & Account Views:** จัดการเรื่องการยืนยันตัวตน (Authentication)
- **Matching & Gear Views:** ประมวลผลอัลกอริทึมการแนะนำสินค้า (recommend_setup) และการค้นหา
- **Admin Views:** จัดการสิทธิ์และการเข้าถึงข้อมูลในระบบจัดการ
- **ความสัมพันธ์:** Views จะสร้าง (<<creates>>) ฟอรัมขึ้นมาเพื่อรับข้อมูล หรือเรียกดู (<<queries>>) ข้อมูลจาก Models โดยตรง

2.2 Forms Layer (ส่วนตรวจสอบข้อมูล) ทำหน้าที่กลั่นกรองและตรวจสอบความถูกต้องของข้อมูล (Data Validation) ก่อนบันทึกลงฐานข้อมูล

- **Auth Forms:** ตรวจสอบข้อมูลสมัครสมาชิก เช่น รูปแบบอีเมลหรือความยาวรหัสผ่าน (RegisterForm, LoginForm)
- **Data Forms:** ตรวจสอบข้อมูลการเพิ่มอุปกรณ์เกมมิ่ง (GamingGearForm) หรือข้อมูลนักแข่ง (ProPlayerForm)
- **ความสัมพันธ์:** เมื่อตรวจสอบแล้ว Forms จะทำการสร้างตัวแปร (<<instantiates>>) ของ Models เพื่อเตรียมบันทึก

2.3 Models Layer (ส่วนฐานข้อมูล) เป็นโครงสร้างหลักในการจัดเก็บข้อมูล (Database Schema)

- **Users & Roles:** เก็บข้อมูลผู้ใช้และสิทธิ์การใช้งาน (User, Role)
- **Gears & Players:** เก็บข้อมูลอุปกรณ์เกมมิ่ง (GamingGear) และนักแข่ง (ProPlayer) ซึ่งมีความสัมพันธ์แบบ Many-to-Many ผ่านตาราง ProPlayerGear
- **User Data:** เก็บข้อมูลที่เกิดจากการใช้งานของ User เช่น Preset (ชุดอุปกรณ์ที่จัดไว้) และ Rating (คะแนนความพึงพอใจ)
- **System Logs & AI:** เก็บข้อมูลการทำงานของระบบ เช่น AdminLog (บันทึกกิจกรรมแอดมิน)
- **ความสัมพันธ์:** แสดงความสัมพันธ์เชิงข้อมูล เช่น User 1 คน สร้างได้หลาย




Preset (1--> 0..) หรือ ProPlayer ใช้งานได้หลาย GamingGear

3.6 Sequence diagram

แผนภาพลำดับอธิบายลำดับการทำงานของระบบ Gaming Gear Matcher โดยแบ่งออกเป็น 3 ส่วนหลักตามสิทธิ์การใช้งานของผู้ใช้ (User Roles) ได้แก่ Guest, Member, และ Admin เพื่อให้เห็นภาพรวมของปฏิสัมพันธ์ระหว่างผู้ใช้ ส่วนติดต่อผู้ใช้ (View/Controller) และฐานข้อมูล (Model) อย่างชัดเจน

สัญลักษณ์ที่ใช้ในแผนภาพ (Symbol Descriptions)

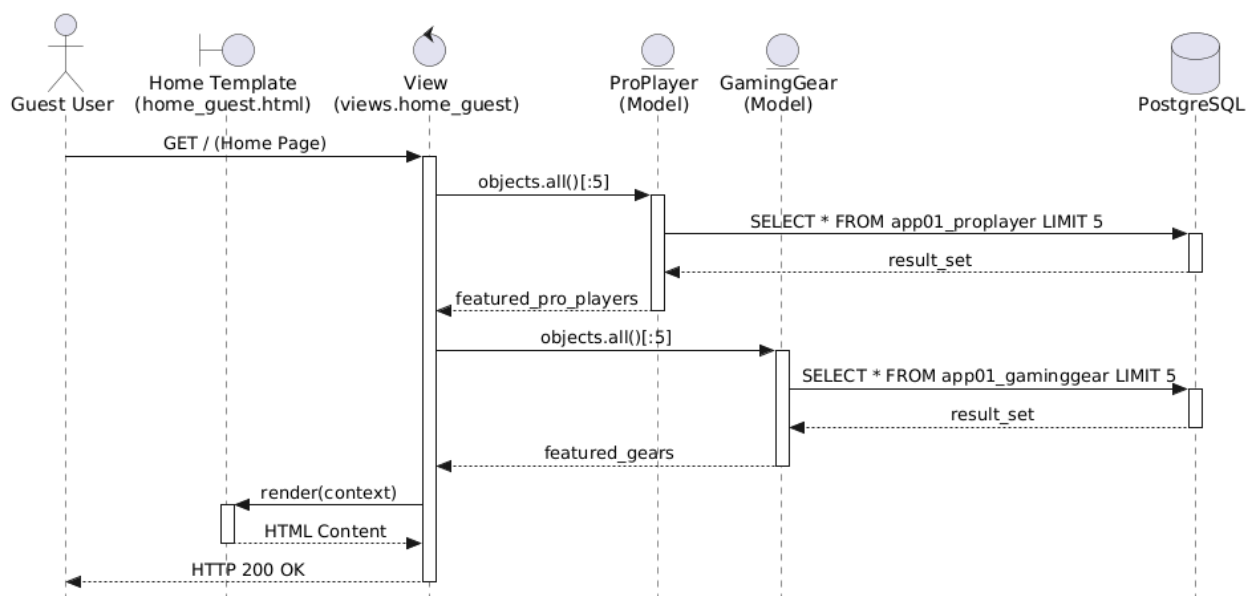
เพื่อให้เกิดความเข้าใจที่ตรงกันในการอ่านแผนภาพ จึงมีการกำหนดความหมายของสัญลักษณ์ที่ใช้ตามมาตรฐาน UML ดังนี้

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมายและ (Description)
 (รูปคน)	Actor (ตัวแสดงแทน)	ผู้ใช้งานระบบหรือระบบภายนอกที่เป็นผู้เริ่มต้นการทำงาน
 (เส้นประแนวตั้ง)	Lifeline (เส้นชีวิต)	เส้นที่ลากลงมาจากวัตถุ แสดงช่วงเวลาวัตถุนั้นมีตัวตนอยู่ในระบบเพื่อรอรับคำสั่ง
 (แท่งสี่เหลี่ยมยาว)	Activation (แถบการทำงาน)	แท่งสี่เหลี่ยมบนเส้นชีวิต แสดงเวลาที่วัตถุกำลังประมวลผลหรือทำงานอยู่ (Active)
	Participant (วัตถุ)	คลาสหรือส่วนประกอบของระบบที่ทำงานร่วมกัน โดยในภาพมีการระบุประเภท (Stereotype) ไว้

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมายและ (Description)
 (เส้นทึบหัวลูกศรทึบ)	Synchronous Message	การเรียกใช้งานฟังก์ชันหรือส่งข้อความแบบที่ผู้ส่ง "ต้องรอ" จนกว่าจะทำงานเสร็จ
 (เส้นประหัวลูกศรโปร่ง)	Return Message	การส่งค่าผลลัพธ์คืนกลับไปยังผู้เรียก
 (เส้นทึบระบุ <<create>>)	Create Message	คำสั่งสร้างวัตถุใหม่ (Instantiate)
 (กรอบสี่เหลี่ยมระบุ loop)	Loop Fragment	กรอบควบคุมการทำงานซ้ำตามเงื่อนไข
 (กรอบสี่เหลี่ยมระบุ alt)	Alternative Fragment	กรอบเงื่อนไขที่แบ่งการทำงานเป็นทางเลือก (คล้าย If/Else) จะทำเพียงทางเลือกเดียวตามเงื่อนไขที่เป็นจริง
 (กรอบใหญ่พื้นสีเทา)	System Boundary Group	การแบ่งกลุ่มเลเยอร์การทำงานเพื่อให้เห็นโครงสร้างชัดเจน

ตารางที่ 4 คำอธิบายแผนภาพ Sequence Diagram

กลุ่มที่ 1: ผู้ใช้งานทั่วไป (Guest & General Features)



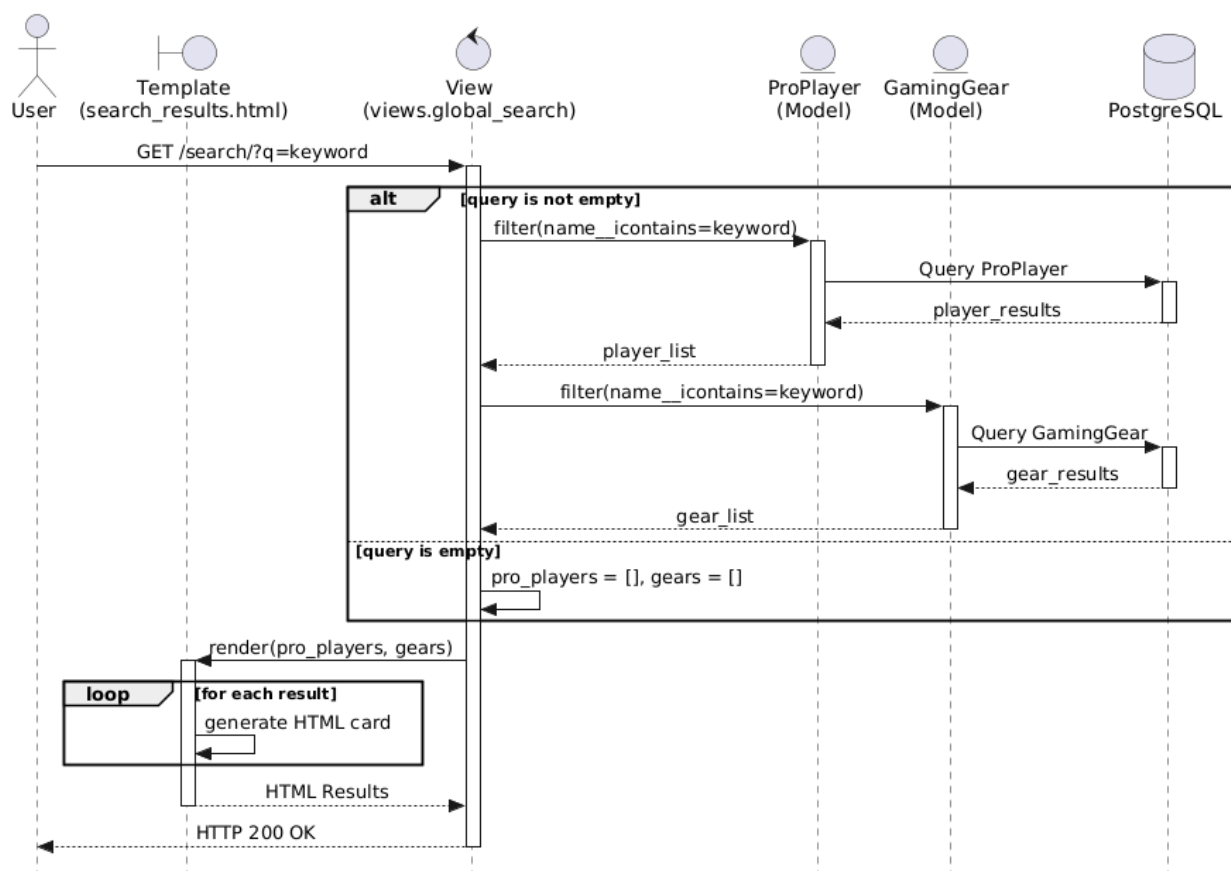
รูปภาพที่ 24 แสดงกระบวนการเข้าถึงหน้าแรกของเว็บไซต์

รูปภาพที่ 24 จะแสดงกระบวนการเข้าถึงหน้าแรกของเว็บไซต์สำหรับผู้เยี่ยมชมทั่วไป (Guest) โดยมีขั้นตอนทางเทคนิคดังนี้:

การร้องขอหน้าเว็บ (HTTP Request): ผู้ใช้ส่งคำร้องขอแบบ GET ไปยัง URL หน้าแรก ('/')

การดึงข้อมูลแนะนำ (Data Retrieval): View เรียกคำสั่ง `objects.all()[:5]` จาก Model ของ `ProPlayer` และ `GamingGear` เพื่อดึงข้อมูลแนะนำจำนวน 5 รายการจากฐานข้อมูล PostgreSQL

การแสดงผล (Rendering): Django Template Engine นำข้อมูลที่ได้ (Context) ไปประมวลผลร่วมกับไฟล์ HTML (`home_guest.html`) เพื่อสร้างหน้าเว็บที่สมบูรณ์และส่งกลับไปยัง Browser ของผู้ใช้



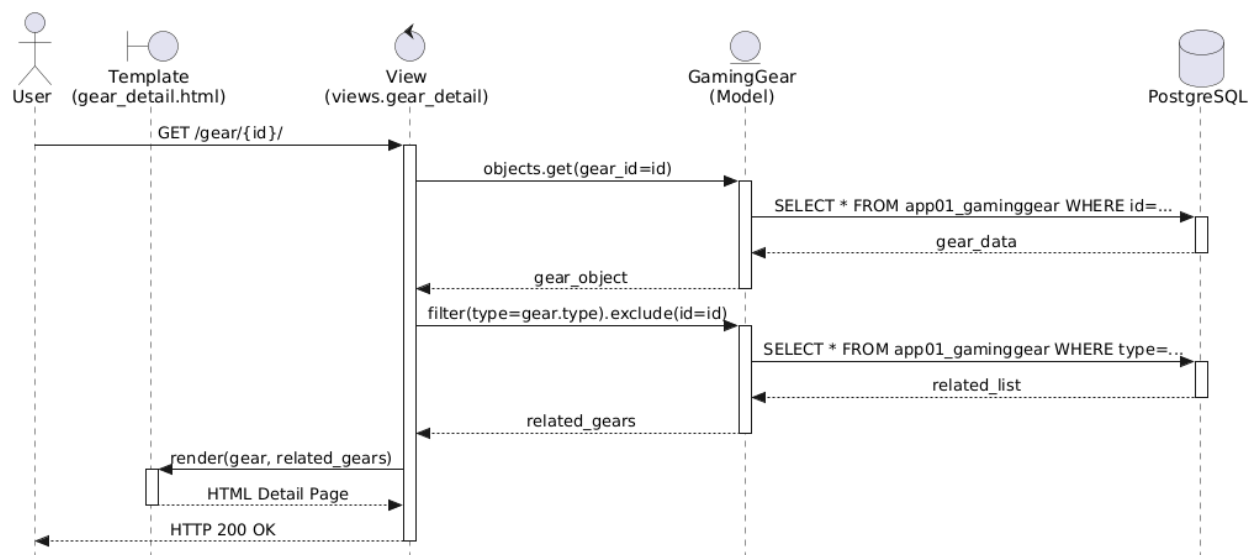
รูปภาพที่ 25 แสดงกระบวนการค้นหาข้อมูลแบบรวมศูนย์

รูปภาพที่ 25 จะแสดงกระบวนการค้นหาข้อมูลแบบรวมศูนย์ (Global Search) โดยมีขั้นตอนทางเทคนิคดังนี้:

การส่งคำค้นหา (Query Submission): ผู้ใช้พิมพ์คำค้นหาและส่ง Parameter `q` ผ่าน URL (GET Request)

การกรองข้อมูล (Data Filtering): View ทำการ Query ข้อมูลจาก 2 ตารางพร้อมกัน คือ `ProPlayer` และ `GamingGear` โดยใช้เงื่อนไข `icontains` เพื่อค้นหาข้อความแบบไม่สนใจตัวพิมพ์เล็ก-ใหญ่ (Case-insensitive)

การรวมผลลัพธ์ (Result Aggregation): ระบบนำผลลัพธ์จากทั้งสองส่วนมารวมกันใน Context Dictionary และส่งไปแสดงผลที่หน้า `search_results.html`



รูปภาพที่ 26 แสดงกระบวนการเรียกดูรายละเอียดของอุปกรณ์เกมมิ่ง

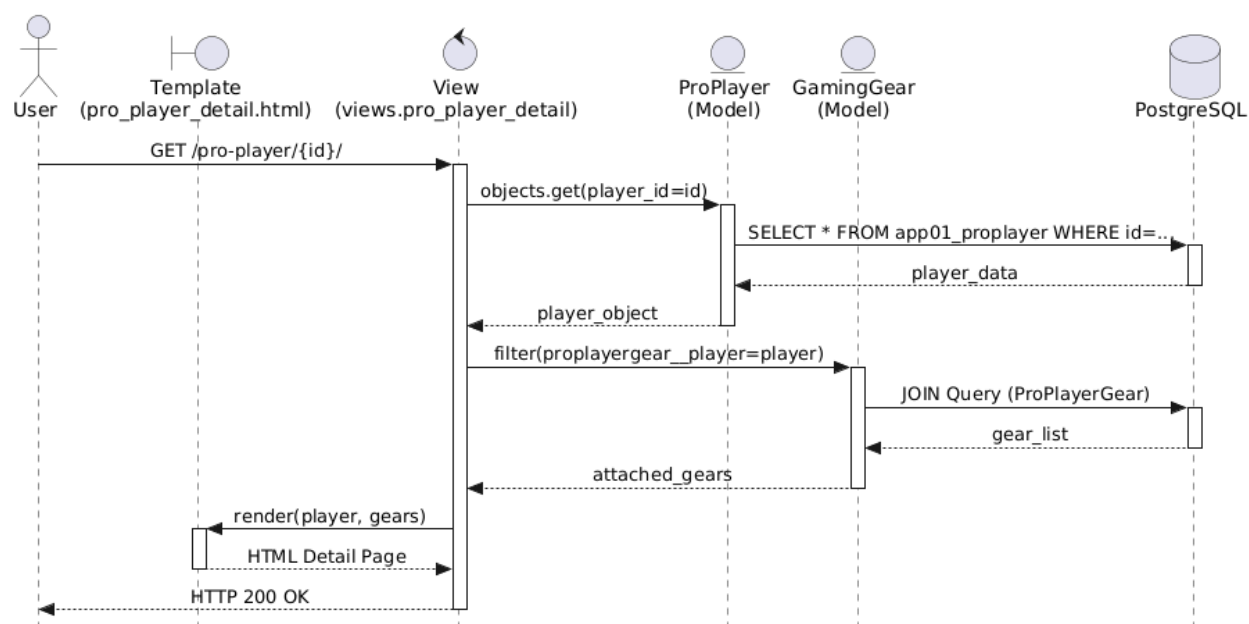
รูปภาพที่ 26 จะแสดงกระบวนการเรียกดูรายละเอียดของอุปกรณ์เกมมิ่ง โดยมีขั้นตอนทางเทคนิค ดังนี้:

การระบุตัวตนวัตถุ (Object Identification): ระบบรับค่า `gear_id` จาก URL เพื่อใช้เป็น Key ในการค้นหาข้อมูล

การดึงข้อมูลจำเพาะ (Specific Data Retrieval): View ใช้คำสั่ง `objects.get(gear_id=id)` เพื่อดึงข้อมูลรายละเอียดทั้งหมดของอุปกรณ์นั้น

การค้นหาความสัมพันธ์ (Related Item Query): ระบบค้นหาอุปกรณ์อื่นที่มี `type` เดียวกัน แต่ไม่ใช่ชิ้นปัจจุบัน (`exclude`) เพื่อนำมาแสดงเป็น "สินค้าที่เกี่ยวข้อง"

การถอดรหัสข้อมูล (Data Parsing): หากข้อมูลสเปคถูกเก็บแบบ JSON ระบบจะทำการ Deserialize เพื่อให้ Template สามารถเข้าถึงค่า (Attribute) ภายในได้

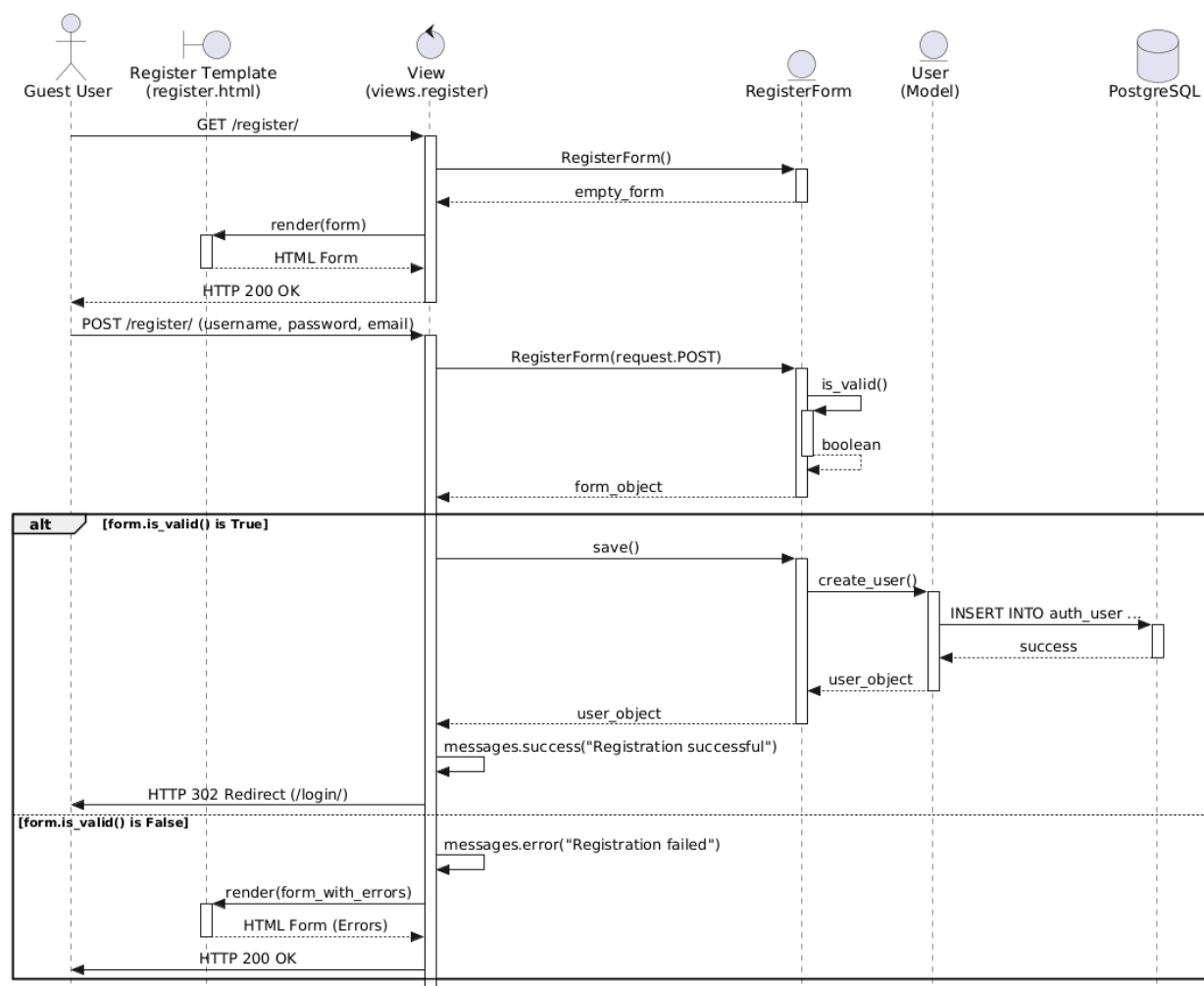


รูปภาพที่ 27 แสดงกระบวนการเรียกดูข้อมูลโปรเพลเยอร์และอุปกรณ์ที่ใช้งาน

รูปภาพที่ 27 จะแสดงกระบวนการเรียกดูข้อมูลโปรเพลเยอร์และอุปกรณ์ที่ใช้งาน โดยมีขั้นตอนทางเทคนิคดังนี้:

การดึงข้อมูลเชิงสัมพันธ์ (Relational Query): View ดึงข้อมูล 'ProPlayer' หลัก และทำการ Join ตารางผ่าน 'ProPlayerGear' เพื่อหารายการอุปกรณ์ทั้งหมดที่ผู้เล่นคนนั้นใช้งาน

การจัดการข้อผิดพลาด (Exception Handling): หากไม่พบข้อมูล Player ID ที่ระบุ ระบบจะส่งกลับหน้า 404 Not Found หรือแสดงข้อมูล Demo (กรณีอยู่ในโหมดทดสอบ)



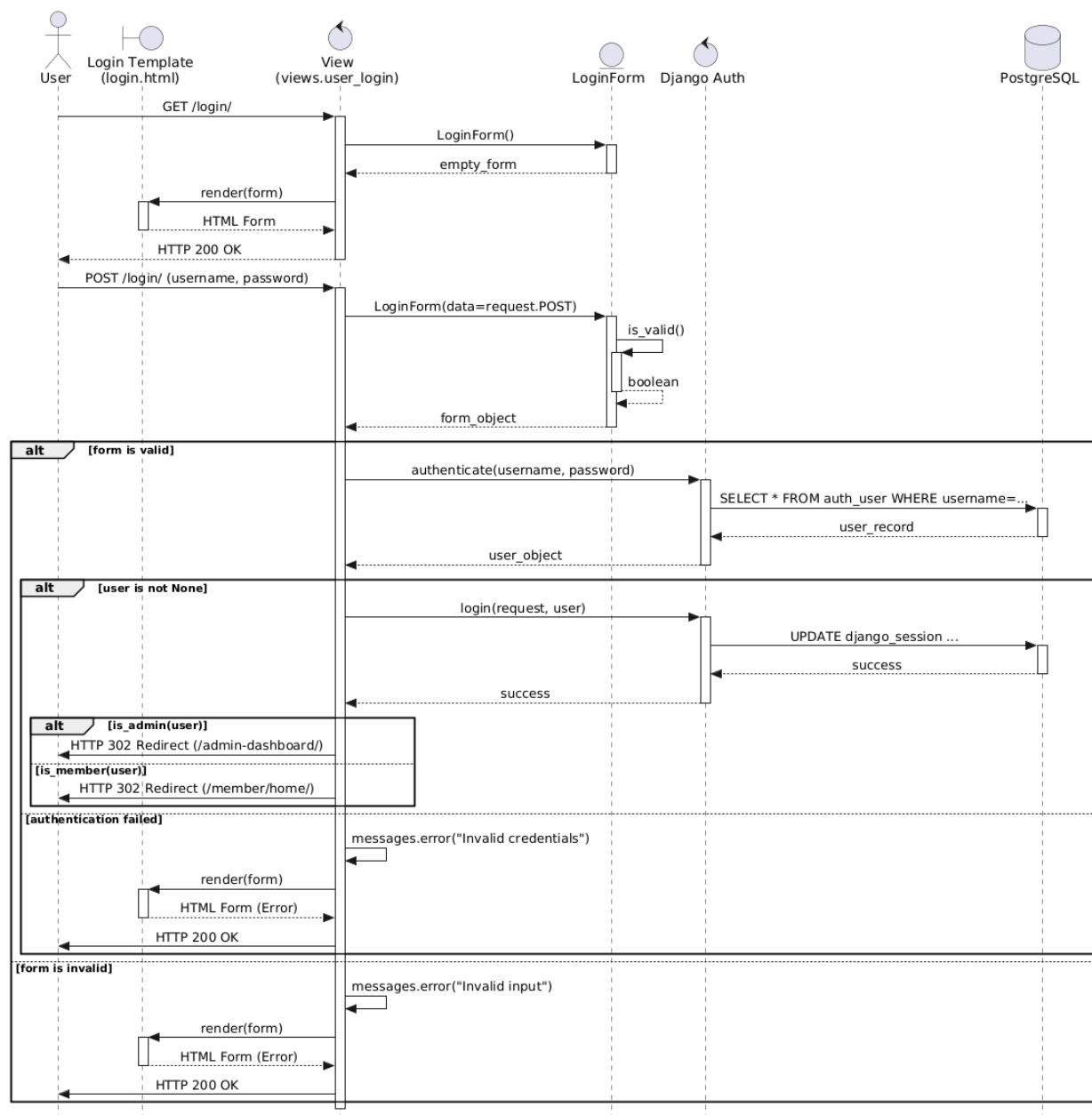
รูปภาพที่ 28 แสดงกระบวนการลงทะเบียนสมาชิกใหม่

รูปภาพที่ 28 จะแสดงกระบวนการลงทะเบียนสมาชิกใหม่ โดยมีขั้นตอนทางเทคนิคดังนี้:

การส่งข้อมูล (Form Submission): ผู้ใช้กรอกข้อมูลและส่งผ่าน `RegisterForm` (POST Request) ไปยัง Server

การตรวจสอบความถูกต้อง (Validation): ระบบตรวจสอบรูปแบบข้อมูล (Form Validation) เช่น ความยาวรหัสผ่าน และความซ้ำซ้อนของ Username ในฐานข้อมูล

การบันทึกข้อมูล (Record Creation): เมื่อข้อมูลถูกต้อง Model จะทำการ Hash Password และบันทึก User ใหม่ลงในตาราง `auth_user` จากนั้นทำการ Redirect ผู้ใช้ไปยังหน้า Login



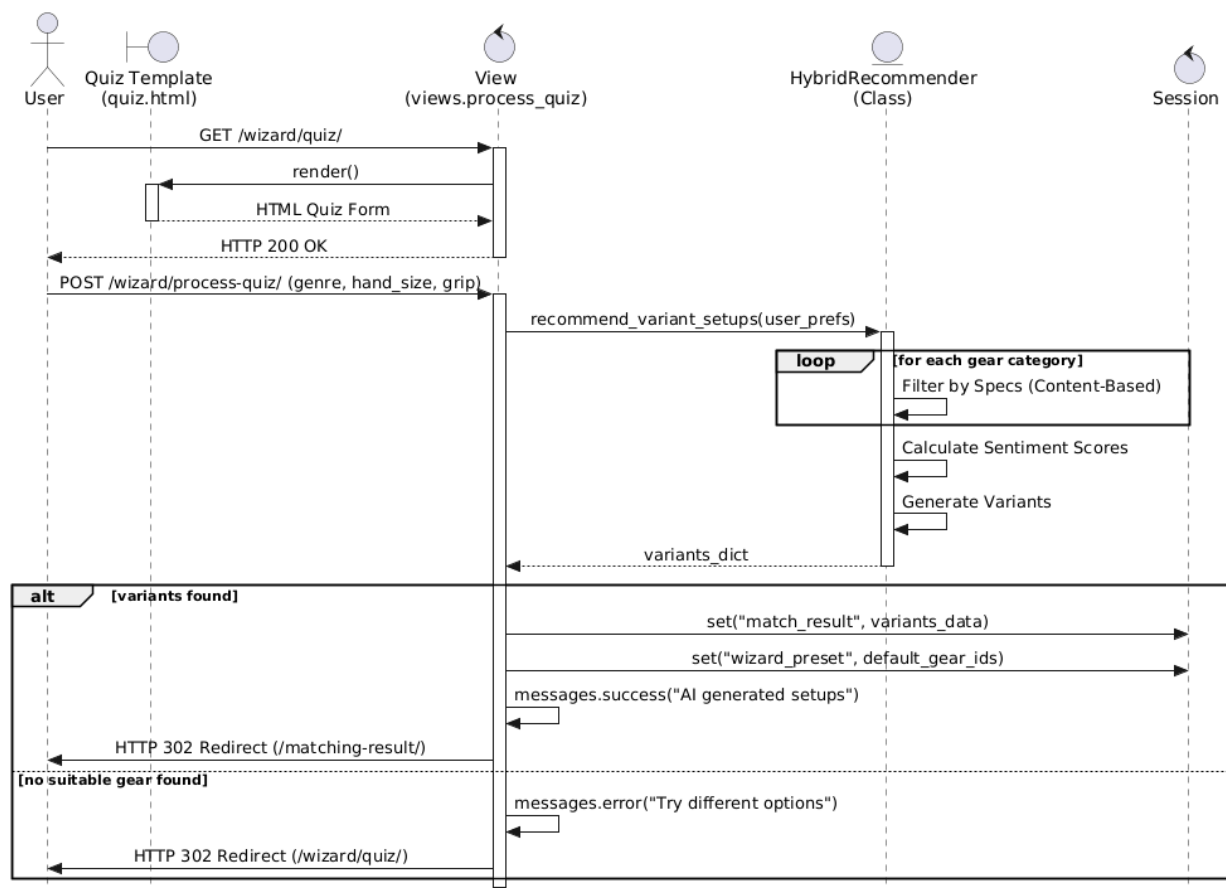
รูปภาพที่ 29 แสดงกระบวนการเข้าสู่ระบบ

รูปภาพที่ 29 จะแสดงกระบวนการเข้าสู่ระบบ โดยมีขั้นตอนทางเทคนิคดังนี้:

การยืนยันตัวตน (Authentication): View รับค่า Username/Password และเรียกใช้ฟังก์ชัน `authenticate()` ของ Django เพื่อตรวจสอบกับฐานข้อมูล

การสร้างเซสชัน (Session Creation): หากข้อมูลถูกต้อง ระบบจะสร้าง Session ID และบันทึกลงใน Browser Cookie ของผู้ใช้งาน (Login)

การตรวจสอบสิทธิ์ (Authorization Check): ระบบตรวจสอบสถานะผู้ใช้ (`is_superuser` หรือ `Role`) เพื่อทำการ Redirect ไปยังหน้า Dashboard (สำหรับ Admin) หรือหน้า Home (สำหรับ Member)



รูปภาพที่ 30 แสดงการประมวลผลแบบสอบถามและระบบแนะนำแบบ Hybrid

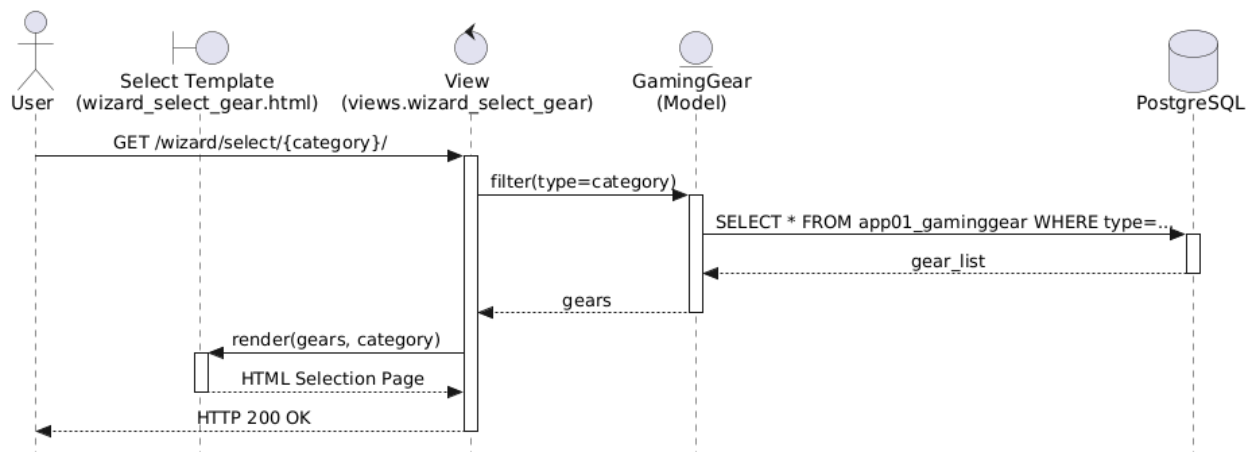
รูปภาพที่ 30 จะแสดงการประมวลผลแบบสอบถามและระบบแนะนำแบบ Hybrid โดยมีขั้นตอนทางเทคนิคดังนี้:

การรับค่าพารามิเตอร์ (Parameter Acquisition): View รับค่า Genre, Hand Size, และ Grip Type จากฟอร์ม

การประมวลผลอัลกอริทึม (Algorithm Execution): เรียกใช้คลาส `HybridRecommender` เพื่อ:

1. กรองสินค้าตามสเปค (Content-Based)
2. คำนวณคะแนน Sentiment จากรีวิว
3. จัดชุดอุปกรณ์แนะนำ (Variant Setups)

การบันทึกผลลัพธ์ (Result Storage): ผลลัพธ์ที่ได้จะถูกบันทึกลงใน Session เพื่อใช้แสดงผลในหน้าถัดไป

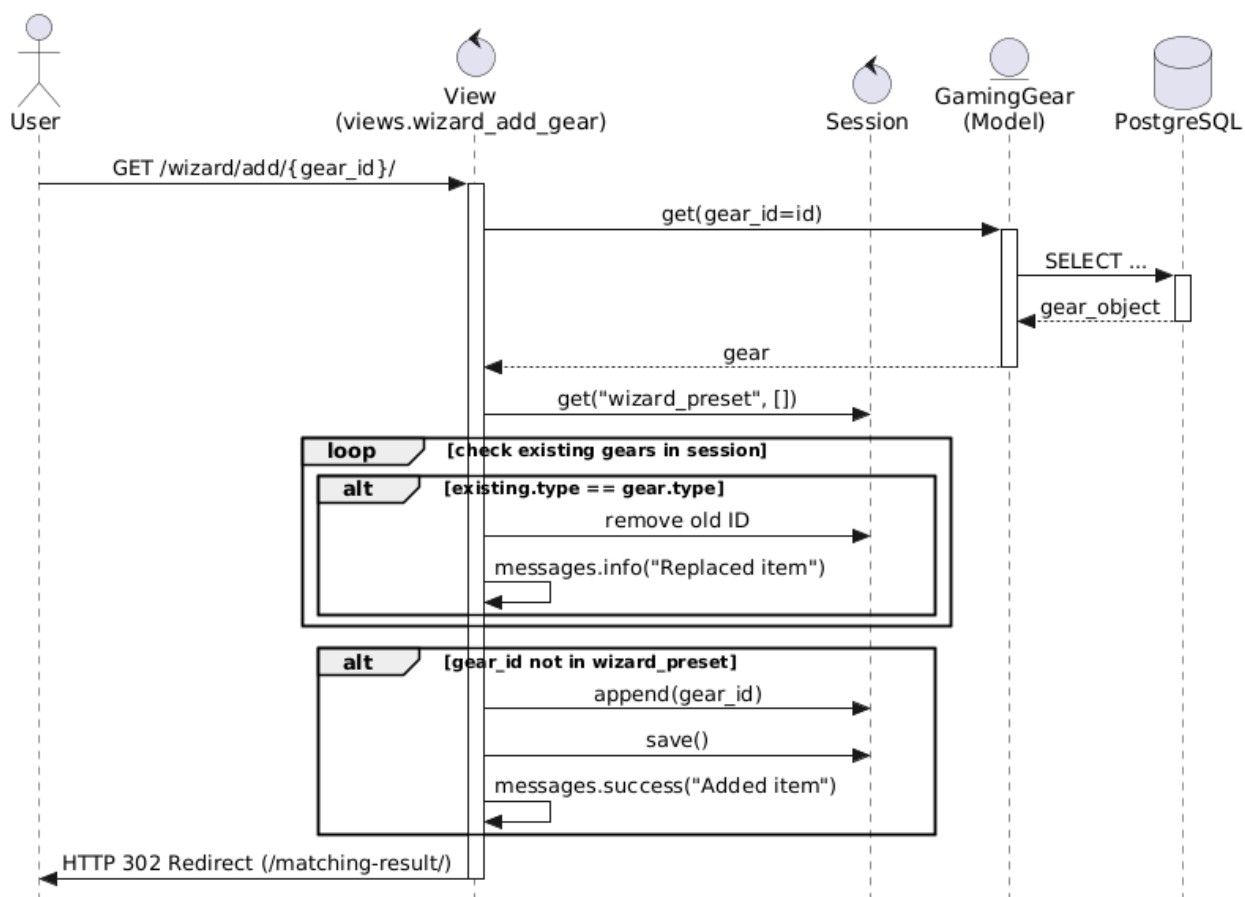


รูปภาพที่ 31 แสดงกระบวนการเลือกอุปกรณ์รายการหมวดหมู่ (Manual Selection)

รูปภาพที่ 31 จะแสดงกระบวนการเลือกอุปกรณ์รายการหมวดหมู่ (Manual Selection) โดยมีขั้นตอนทางเทคนิคดังนี้:

การกรองตามประเภท (Type Filtering): View รับค่าพารามิเตอร์ `category` (เช่น Mouse, Keyboard) และ Query ฐานข้อมูลเฉพาะประเภทนั้น

การส่งต่อบริบท (Context Passing): ระบบส่งรายการอุปกรณ์พร้อม Parameter `next_url` ไปยัง Template เพื่อให้เมื่อเลือกเสร็จแล้วสามารถกลับมาดูหน้า



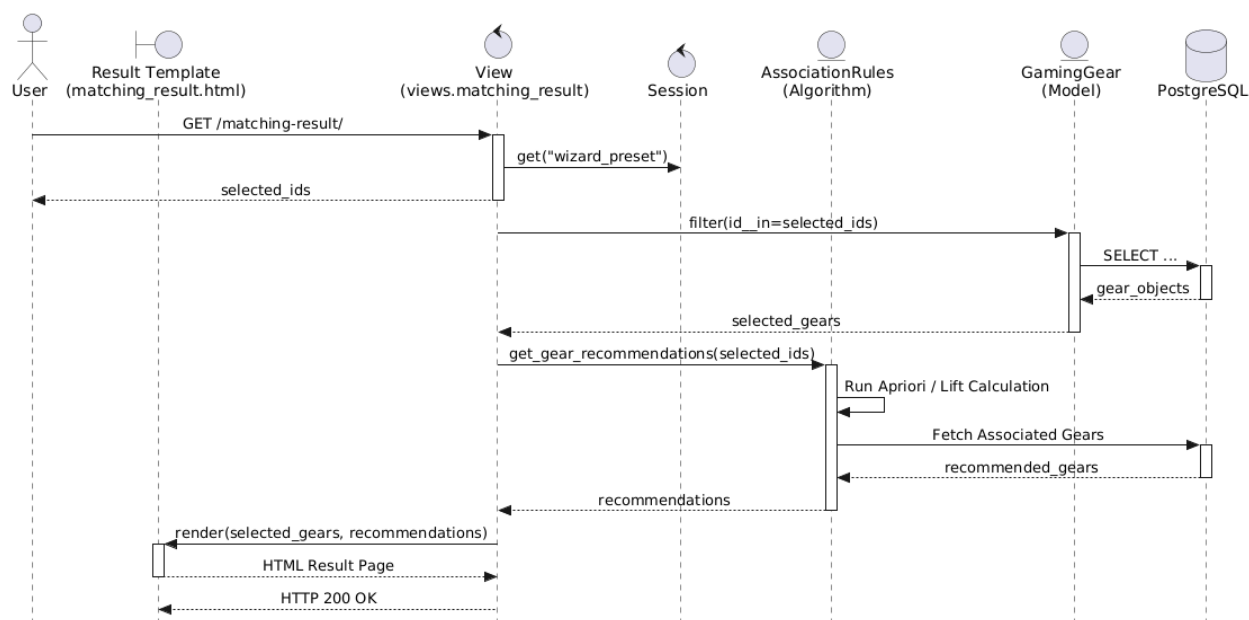
รูปภาพที่ 32 แสดงการจัดการรายการอุปกรณ์ชั่วคราว

รูปภาพที่ 32 จะแสดงการจัดการรายการอุปกรณ์ชั่วคราวใน Wizard โดยมีขั้นตอนทางเทคนิคดังนี้:

การตรวจสอบความซ้ำซ้อน (Duplicate Check): เมื่อผู้ใช้เพิ่มอุปกรณ์ใหม่ ระบบจะตรวจสอบใน Session ว่ามีอุปกรณ์ประเภทเดียวกันอยู่แล้วหรือไม่

การแทนที่ข้อมูล (Data Replacement): หากมีอุปกรณ์เดิมอยู่ ระบบจะทำการลบ ID เดิมออก และแทนที่ด้วย ID ใหม่ (Enforce One-Item-Per-Category)

การอัปเดตเซสชัน (Session Persistence): บันทึกรายการ ID ใหม่ลงในตัวแปร 'wizard_preset' และสั่ง 'session.modified = True'



รูปภาพที่ 33 แสดงการแสดงผลการจับคู่และการแนะนำ

รูปภาพที่ 33 จะแสดงการแสดงผลการจับคู่และการแนะนำเพิ่มเติม โดยมีขั้นตอนทางเทคนิคดังนี้:

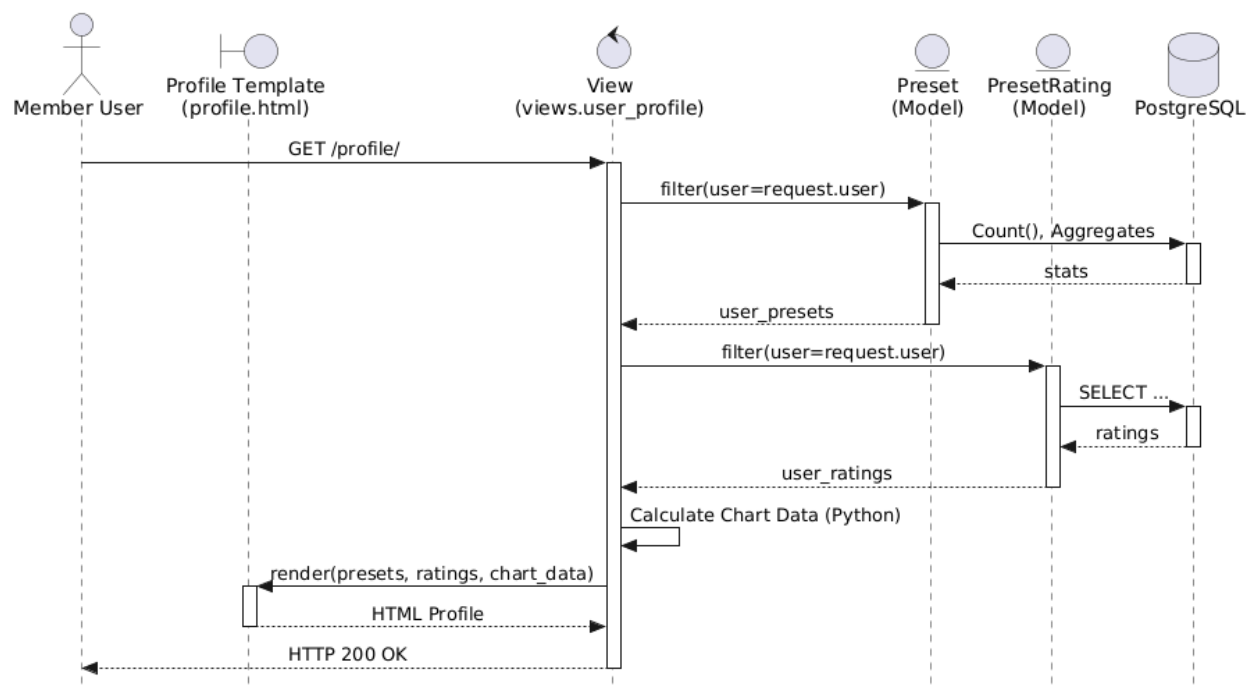
การดึงข้อมูลอุปกรณ์ที่เลือก (Selected Gear Retrieval): View แปลงรายการ ID จาก Session กลับเป็น Object ของ GamingGear

การค้นหความสัมพันธ์ (Association Rule Mining): ส่งรายการ ID เข้าฟังก์ชัน `get_gear_recommendations` เพื่อคำนวณหาอุปกรณ์ที่สัมพันธ์กันด้วย Apriori Algorithm (เช่น "คนใช้เมาส์นี้ มักใช้แผ่นรองนี้")

การแสดงผล (Result Rendering): แสดงผลอุปกรณ์ที่เลือก, ชุด Variant จาก AI, และสินค้าแนะนำเพิ่มเติมในหน้าเดียวกัน

กลุ่มที่ 2: สมาชิกในระบบ (Member Features)

สมาชิกในระบบนั้นจะสามารถทำทุกอย่างที่ Guest สามารถทำได้ (จากกลุ่มที่ 1) และจะเพิ่มเติม ดังนี้



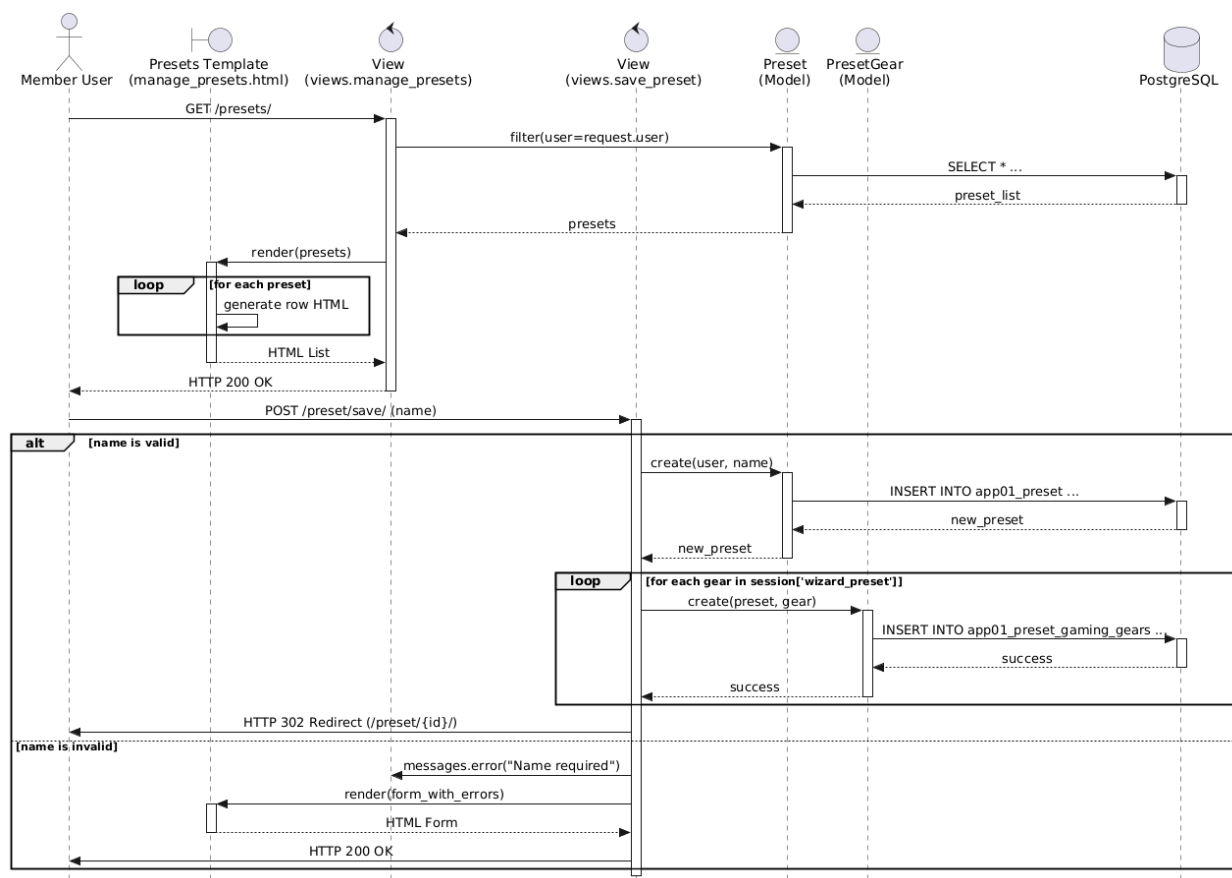
รูปภาพที่ 34 แสดงการทำงานของหน้าโปรไฟล์

รูปภาพที่ 34 จะแสดงการทำงานของหน้าโปรไฟล์และสถิติผู้ใช้ โดยมีขั้นตอนทางเทคนิคดังนี้:

การรวบรวมข้อมูล (Data Aggregation): ระบบ Query ฐานข้อมูลเพื่อนับจำนวน Preset และ Review ที่ผู้ใช้เคยทำ

การสร้างกราฟ (Visual Data Preparation): View ทำการ Group ข้อมูลตามเดือน (`TruncMonth`) เพื่อเตรียมข้อมูล JSON สำหรับแสดงผลกราฟ Activity Timeline

การแสดงประวัติ (History Listing): ดึงรายการ Preset ล่าสุด 5 รายการมาแสดงผล



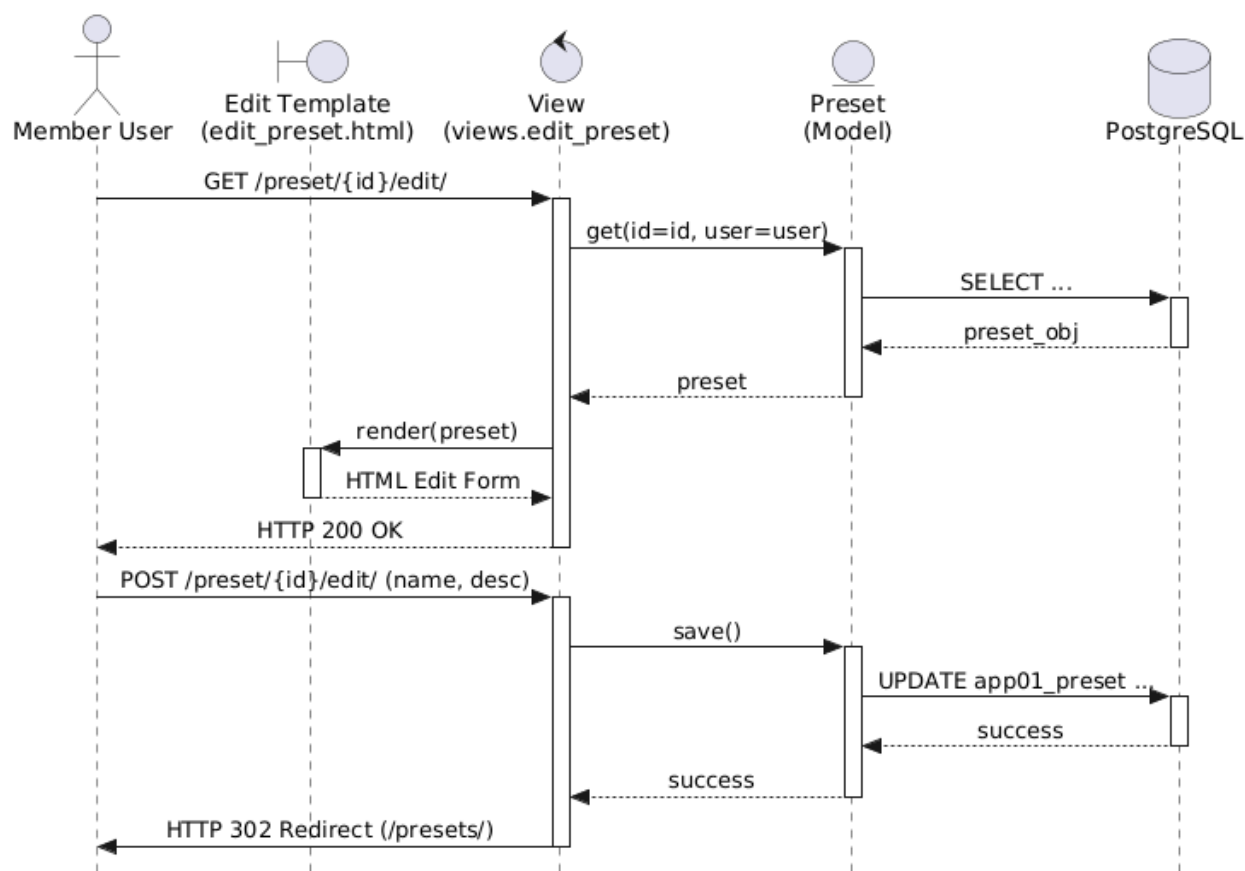
รูปภาพที่ 35 แสดงกระบวนการดูรายการและสร้าง Preset ใหม่

แสดงกระบวนการดูรายการและสร้าง Preset ใหม่ โดยมีขั้นตอนทางเทคนิคดังนี้:

การดึงรายการ (Fetching): Query ตาราง Preset โดยกรองเฉพาะของ `request.user`

การบันทึก (Saving Transaction): เมื่อผู้ใช้กด Save ระบบจะ:

1. สร้าง Record ในตาราง `Preset` (Header)
2. วนลูปสร้าง Record ในตาราง `PresetGear` (Detail) ตามจำนวนอุปกรณ์ใน Session
3. ทำการ Commit Transaction เพื่อบันทึกข้อมูลถาวร

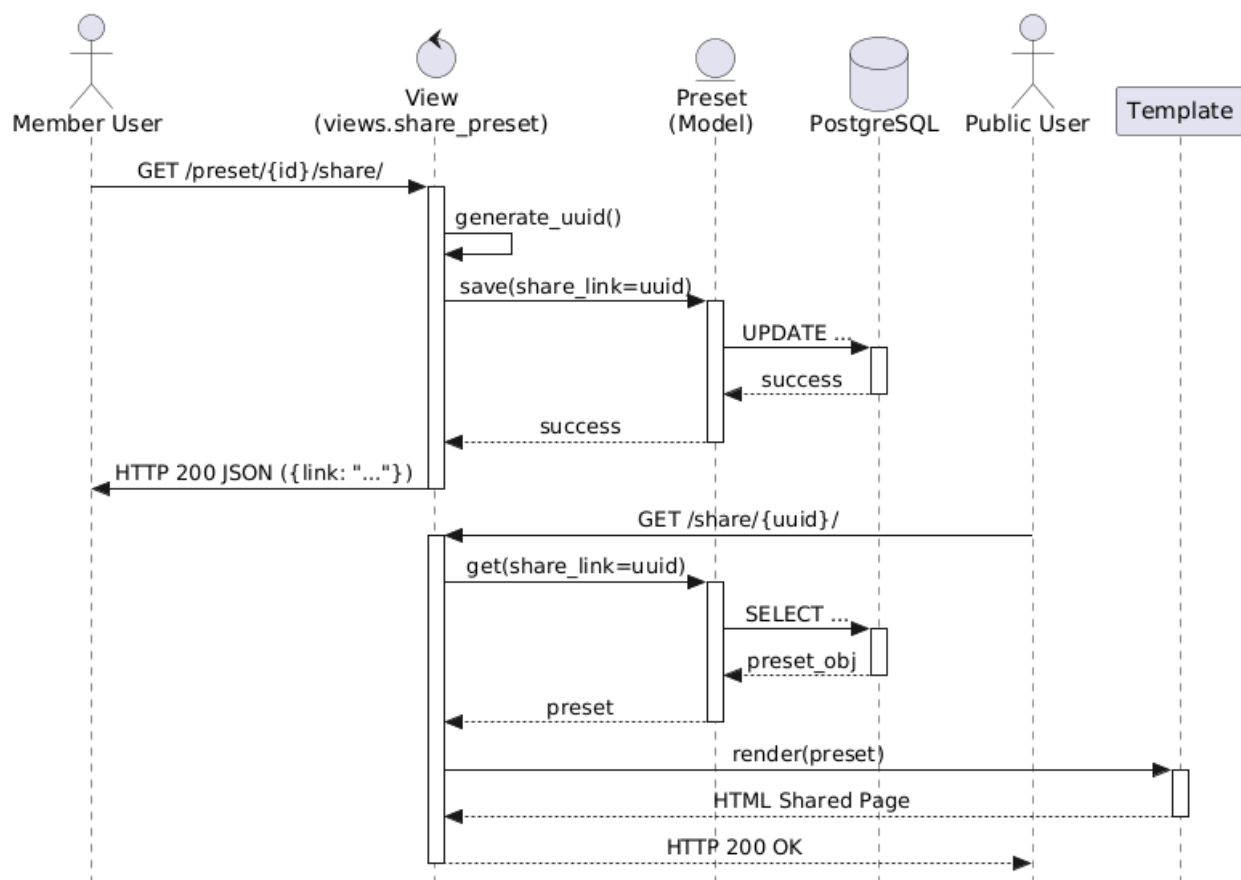


รูปภาพที่ 36 แสดงกระบวนการแก้ไขข้อมูล Preset

รูปภาพที่ 36 จะแสดงกระบวนการแก้ไขข้อมูล Preset โดยมีขั้นตอนทางเทคนิคดังนี้:

การตรวจสอบความเป็นเจ้าของ (Ownership Verification): ระบบตรวจสอบว่า User ปัจจุบันเป็นเจ้าของ Preset ID นั้นจริงหรือไม่ ก่อนอนุญาตให้แก้ไข

การอัปเดตข้อมูล (Data Update): รับค่าชื่อและคำอธิบายใหม่จากฟอร์ม และสั่ง `save()` เพื่ออัปเดตฐานข้อมูล



รูปภาพที่ 37 แสดงกระบวนการแชร์ Preset

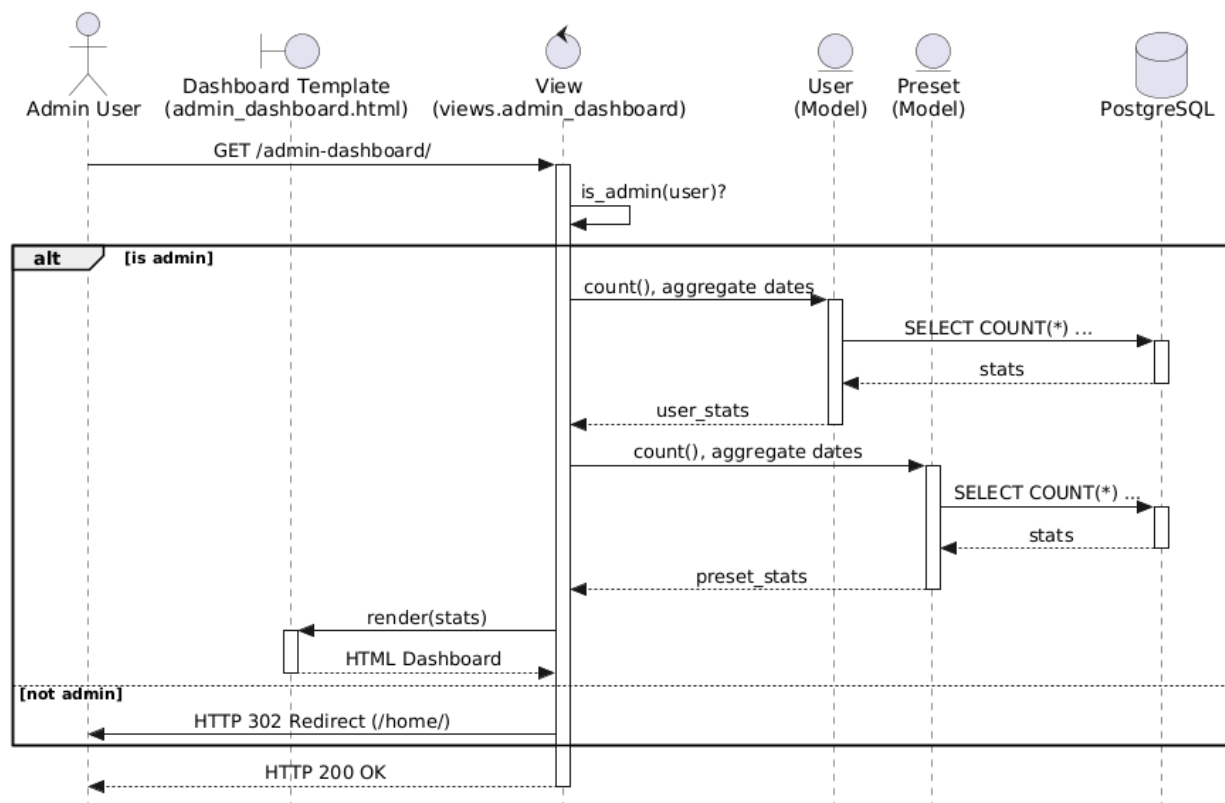
รูปภาพที่ 37 จะแสดงกระบวนการแชร์ Preset สู่สาธารณะ โดยมีขั้นตอนทางเทคนิคดังนี้:

การสร้างลิงก์ (Link Generation): ระบบสร้าง UUID (Universally Unique Identifier) เพื่อใช้เป็น Token ในการเข้าถึง

การบันทึกสถานะ (State Update): บันทึก UUID ลงในฟิลด์ `share_link` ของ Preset นั้น

การเข้าถึงแบบสาธารณะ (Public Access): เมื่อมีผู้ใช้เข้าผ่าน URL ที่มี UUID ระบบจะค้นหา Preset และแสดงผลแบบ Read-only (ไม่ต้อง Login)

กลุ่มที่ 3: ผู้ดูแลระบบ (Admin Features)

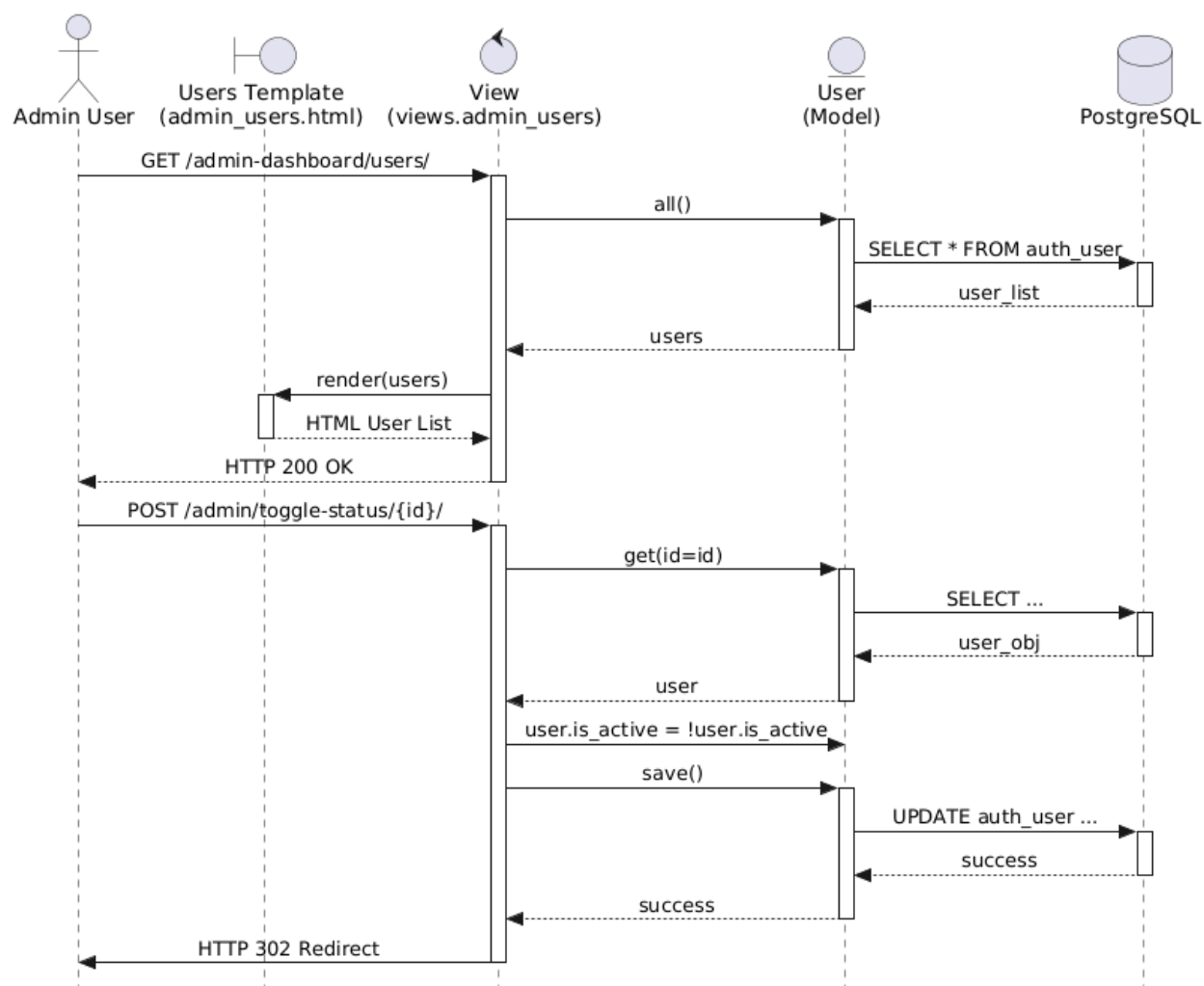


รูปภาพที่ 38 แสดงการทำงานของหน้า Dashboard\

รูปภาพที่ 38 จะแสดงการทำงานของหน้า Dashboard สำหรับผู้ดูแลระบบ โดยมีขั้นตอนทางเทคนิคดังนี้:

การตรวจสอบสิทธิ์ Admin (Role Verification): ระบบใช้ Decorator ตรวจสอบว่าผู้ใช้นั้นมีสถานะเป็น Admin หรือ Superuser

การสรุปผลสถิติ (Statistical Summary): ระบบทำการ Count จำนวน User, Gear, Preset ทั้งหมดในระบบ และคำนวณอัตราการเติบโตเพื่อแสดงผล

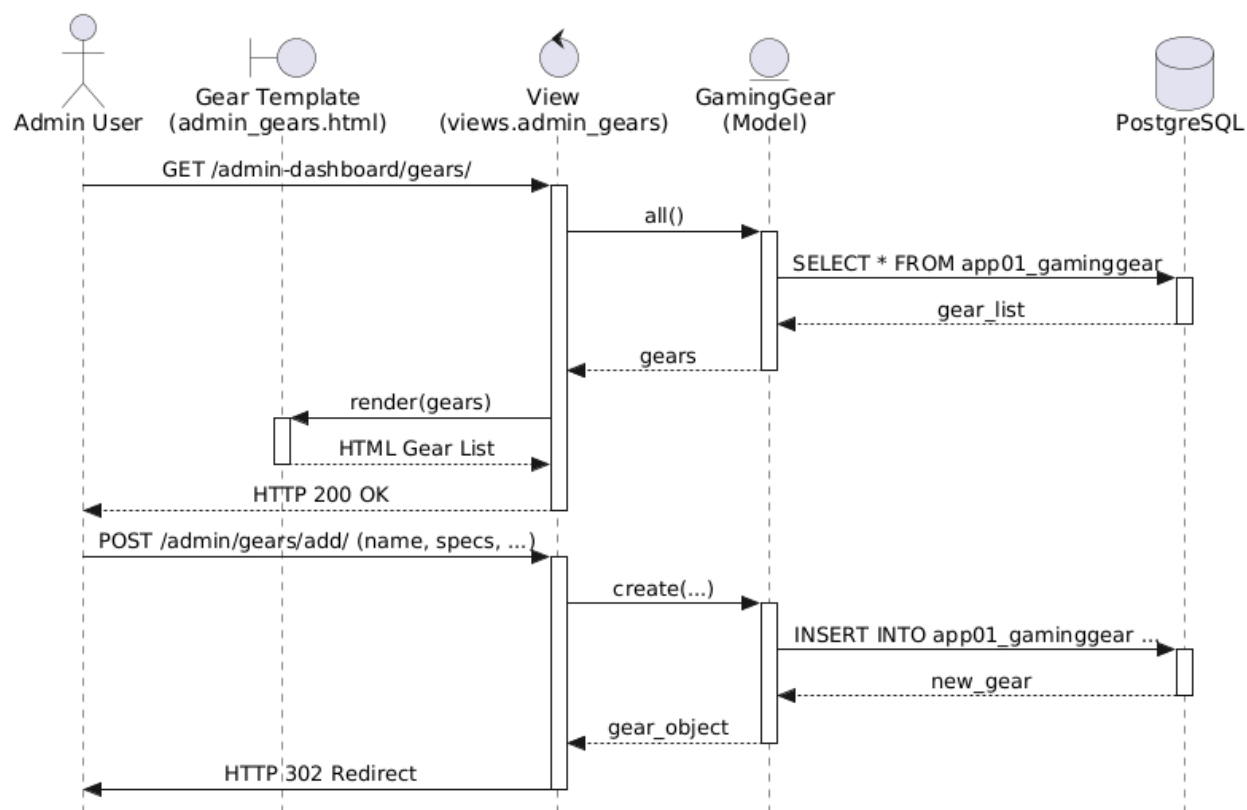


รูปภาพที่ 39 แสดงการจัดการสมาชิก

รูปภาพที่ 39 จะแสดงการจัดการสมาชิกโดย Admin โดยมีขั้นตอนทางเทคนิคดังนี้:

การดึงรายชื่อ (List Retrieval): ดึงข้อมูลสมาชิกทั้งหมดจากตาราง `auth_user`

การปรับสถานะ (Status Toggling): เมื่อ Admin กดปุ่ม Ban/Unban ระบบจะสลับค่า Boolean ในฟิลด์ `is_active` และบันทึกผลทันที

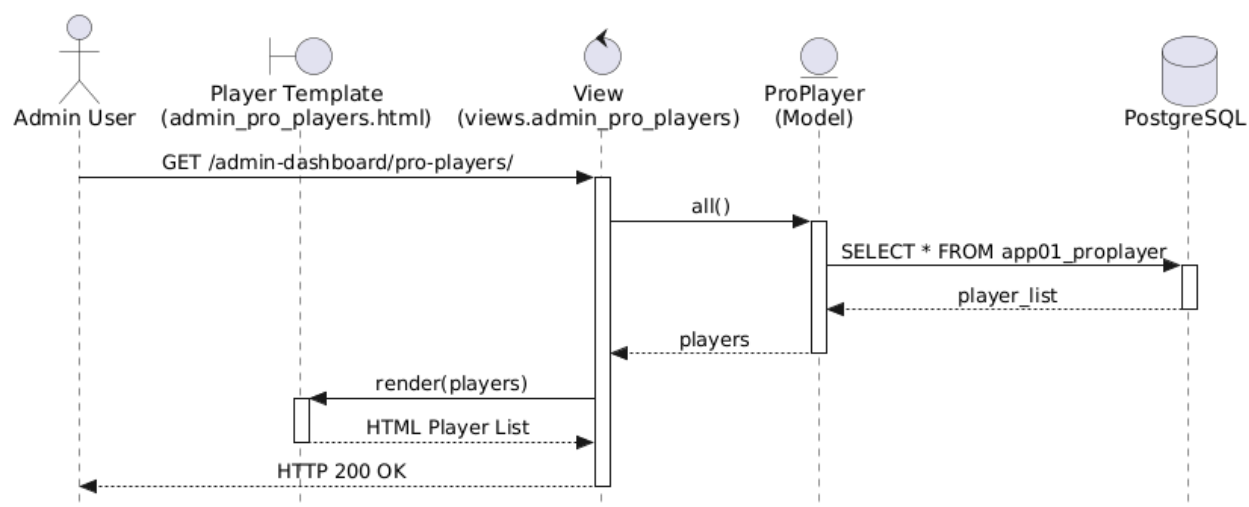


รูปภาพที่ 40 แสดงการจัดการคลังอุปกรณ์เกมมิ่ง

รูปภาพที่ 40 จะแสดงการจัดการคลังอุปกรณ์เกมมิ่ง โดยมีขั้นตอนทางเทคนิคดังนี้:

การเพิ่มข้อมูล (Create Operation): Admin กรอกข้อมูลอุปกรณ์และสเปค (ในรูปแบบ JSON)

การบันทึกไฟล์ (File Handling): หากมีการอัปโหลดรูปภาพ ระบบจะบันทึกไฟล์ลงใน `MEDIA_ROOT` และเก็บ Path ไว้ในฐานข้อมูล



รูปภาพที่ 41 แสดงการจัดการข้อมูลโปรเพลเยอร์

รูปภาพที่ 41 จะแสดงการจัดการข้อมูลโปรเพลเยอร์ โดยมีขั้นตอนทางเทคนิคดังนี้:

การจัดการความสัมพันธ์ (Relationship Management): Admin สามารถเพิ่ม/ลบ อุปกรณ์ที่โปรเพลเยอร์ใช้ ผ่านหน้าแก้ไข โดยระบบจะอัปเดตตารางเชื่อม `ProPlayerGear` ให้สอดคล้องกัน

การบันทึกข้อมูล (Update Operation): บันทึกการเปลี่ยนแปลงข้อมูลชีวประวัติและรูปภาพลงฐานข้อมูล

3.7 Data model

แบบจำลองข้อมูลแสดงโครงสร้างการจัดเก็บข้อมูลในระบบจัดการฐานข้อมูล (Database Schema) โดยระบบ BattleHub ใช้ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ในการจัดเก็บข้อมูล ซึ่งประกอบด้วย ตาราง (Entities) และความสัมพันธ์ (Relationships) ดังแสดงในแผนภาพ Entity Relationship Diagram (ERD)

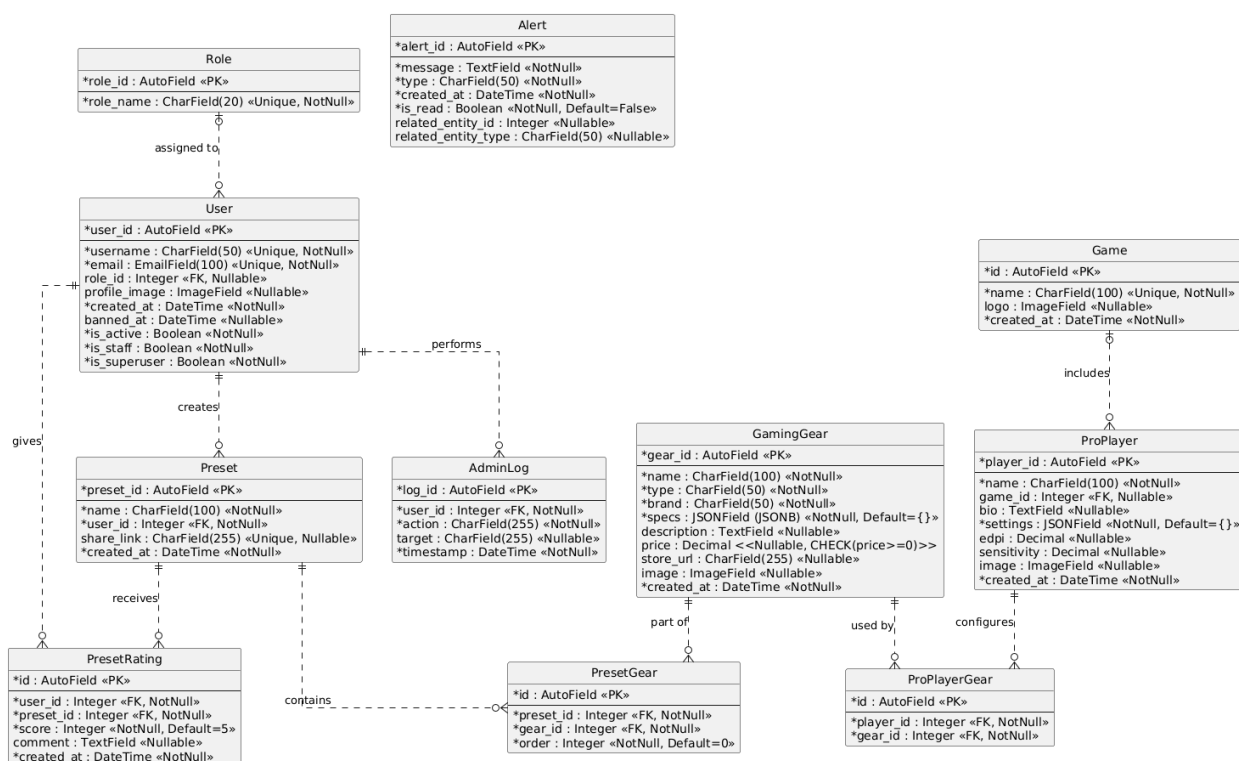
สัญลักษณ์ที่ใช้ในแผนภาพ (Symbol Descriptions)

เพื่อให้เกิดความเข้าใจที่ตรงกันในการอ่านแผนภาพ จึงมีการกำหนดความหมายของสัญลักษณ์ที่ใช้ตามมาตรฐาน Crow's Foot Notation ดังนี้

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมาย (Description)
 <p>(สี่เหลี่ยมแบ่งส่วน)</p>	Entity (เอนทิตี)	ตัวแทนของตารางในฐานข้อมูล ภายในระบุชื่อ ตาราง และแอตทริบิวต์ (Attributes) หรือคอลัมน์ที่ใช้เก็บข้อมูล
	Mandatory One to Optional Many	ความสัมพันธ์แบบ "หนึ่ง ต่อ ศูนย์หรือมากกว่า" (1:0..N) ฝ่ายซ้ายต้องมี 1 เสมอ (Mandatory) / ฝ่ายขวาจะมีหรือไม่ก็ได้ (Optional)
	Optional One to Optional Many	ความสัมพันธ์แบบ "ศูนย์หรือหนึ่ง ต่อ ศูนย์หรือมากกว่า" (0..1:0..N) ฝ่ายซ้ายจะมีหรือไม่ก็ได้ / ฝ่ายขวาจะมีหรือไม่ก็ได้
PK (ข้อความระบุในตาราง)	Primary Key (คีย์หลัก)	แอตทริบิวต์ที่ใช้ระบุตัวตนของแถวข้อมูลในตารางนั้นอย่างไม่ซ้ำกัน เพื่อใช้ในการอ้างอิง

สัญลักษณ์ (Symbol)	ชื่อเรียก (Name)	ความหมาย (Description)
FK (ข้อความระบุใน ตาราง)	Foreign Key (คีย์ นอก)	แอตทริบิวต์ที่ใช้อ้างอิงไปยังคีย์หลัก (Primary Key) ของตารางอื่น เพื่อสร้างความเชื่อมโยงระหว่างข้อมูล

ตารางที่ 5 คำอธิบายภาพ ER Diagram แบบ Crow's Foot Notation



รูปภาพที่ 42 แสดง ER Diagram

รูปภาพนี้อธิบายความสัมพันธ์ระหว่างตารางข้อมูล (Entities) ในฐานข้อมูลของระบบ โดยแบ่งตามกลุ่มการทำงานหลัก ดังนี้:

1. ระบบจัดการผู้ใช้ (User Management System)

□ Role (บทบาท)

- **หน้าที่:** เก็บข้อมูลสิทธิ์การเข้าถึงระบบ เพื่อกำหนดระดับของผู้ใช้งาน เช่น สมาชิกทั่วไป (Member) หรือผู้ดูแลระบบ (Admin)

- **ความสัมพันธ์:** One-to-Many กับ User (1 Role มีผู้ใช้ได้หลายคน แต่ 1 User มีได้แค่ 1 Role)

□ User (ผู้ใช้)

- **หน้าที่:** เก็บข้อมูลบัญชีผู้ใช้ ข้อมูลส่วนตัว และการยืนยันตัวตนสำหรับการเข้าใช้งานระบบ
- **ความสัมพันธ์:** เชื่อมโยงกับ Role เพื่อรับสิทธิ์การใช้งานที่กำหนดไว้

□ AdminLog (บันทึกผู้ดูแล)

- **หน้าที่:** เก็บประวัติการกระทำต่างๆ ของผู้ดูแลระบบ (Admin) เช่น การแบนผู้ใช้ หรือแก้ไขข้อมูลสินค้า เพื่อการตรวจสอบย้อนหลัง
- **ความสัมพันธ์:** One-to-Many กับ User (Admin 1 คน สามารถสร้าง Log ได้หลายรายการ)

2. ระบบข้อมูลอุปกรณ์และนักแข่ง (Gear & Pro Player Data System)

□ Game (เกม)

- **หน้าที่:** เก็บรายชื่อเกม E-Sports ที่ระบบรองรับ เพื่อใช้จัดหมวดหมู่นักแข่ง
- **ความสัมพันธ์:** เชิงตรรกะ (Logical Link) กับ ProPlayer ผ่านชื่อเกม (Game Name)

□ ProPlayer (นักแข่งมืออาชีพ)

- **หน้าที่:** เก็บข้อมูลส่วนตัวของนักแข่ง และการตั้งค่าในเกม (Settings) เช่น Sensitivity, eDPI
- **ความสัมพันธ์:** Many-to-Many กับ GamingGear ผ่านตารางกลาง ProPlayerGear

□ GamingGear (อุปกรณ์เกมมิ่ง)

- **หน้าที่:** เก็บรายละเอียดสินค้าที่เป็นหัวใจหลักของระบบ เช่น ชื่อรุ่น, ประเภท, ราคา, รูปภาพ และสเปค (เก็บในรูปแบบ JSON)
- **ความสัมพันธ์:** ถูกอ้างอิงโดย ProPlayer (ผ่าน ProPlayerGear) และ User (ผ่าน PresetGear)

□ ProPlayerGear (ตารางกลาง)

- **หน้าที่:** เชื่อมความสัมพันธ์ระหว่างนักแข่งกับอุปกรณ์ เนื่องจากนักแข่ง 1 คนใช้อุปกรณ์หลายชิ้น และอุปกรณ์ 1 รุ่นก็มีนักแข่งใช้หลายคน

3. ระบบจัดชุดอุปกรณ์และแนะนำ (Recommendation & Preset System)

□ Preset (ชุดอุปกรณ์)

- **หน้าที่:** เก็บข้อมูลชุดอุปกรณ์ที่ผู้ใช้สร้างขึ้น หรือที่ระบบแนะนำให้ (Wizard Result)
- **ความสัมพันธ์:**
 - One-to-Many กับ User (Owner): ผู้ใช้ 1 คนสร้างได้หลาย Preset
 - Owner เป็นผู้เดียวที่มีสิทธิ์แก้ไขหรือลบ Preset ของตนเอง
- **PresetGear (รายการในชุดอุปกรณ์)**
 - **หน้าที่:** ตารางเชื่อม Many-to-Many ระหว่าง Preset และ GamingGear ระบุว่าในชุดนั้นๆ ประกอบด้วยอุปกรณ์อะไรบ้าง
 - **เพิ่มเติม:** มีฟิลด์ order เพื่อระบุลำดับการแสดงผล
- **PresetRating (การให้คะแนนชุดอุปกรณ์)**
 - **หน้าที่:** เก็บข้อมูลคะแนน (Score 1-5) และความคิดเห็น (Comment) ที่ผู้ใช้มีต่อ Preset
 - **ความสัมพันธ์:** One-to-Many กับ User (Reviewer) และ Preset (Target)

4. ระบบแจ้งเตือน (Notification System)

- **Alert (การแจ้งเตือน)**
 - **หน้าที่:** เก็บข้อมูลการแจ้งเตือนต่างๆ ในระบบ เช่น แจ้งเตือนเมื่อมีคนมาคอมเมนต์ หรือแจ้งเตือนจากระบบ
 - **ความสัมพันธ์:** แบบ Polymorphic Association
 - สามารถแจ้งเตือนเกี่ยวกับ User, ProPlayer, หรือ GamingGear ได้โดยไม่ต้องสร้างตารางแยก
 - ใช้ related_entity_id และ related_entity_type ในการระบุวัตถุที่เกี่ยวข้อง

บทที่ 4

การพัฒนาระบบ

ในบทนี้จะกล่าวถึงรายละเอียดการพัฒนาระบบ Gaming Gear Matcher ซึ่งพัฒนาด้วย Django Framework โดยแบ่งเนื้อหาออกเป็นโครงสร้างแอปพลิเคชัน, การตั้งค่าระบบ, การจัดการไฟล์, การพัฒนาฟังก์ชันหลักตาม Use Case และการจัดการส่วนติดต่อผู้ใช้ (UI/UX)

4.1 โครงสร้างแอปพลิเคชันและการตั้งค่า (Project Configuration)

4.1.1 การจัดการ Settings.py สำหรับการเชื่อมต่อฐานข้อมูลและ Static Files ไฟล์ settings.py เป็นหัวใจหลักในการกำหนดค่าการทำงานของ Django Project ในระบบ Gaming Gear Matcher ได้มีการกำหนดค่าสำคัญดังนี้:

1. การลงทะเบียนแอปพลิเคชัน (Installed Apps) เพื่อให้ Django รู้จักและสามารถทำงานร่วมกับแอปพลิเคชันที่สร้างขึ้นใหม่ จำเป็นต้องเพิ่มชื่อ App เข้าไปในตัวแปร INSTALLED_APPS โดยระบบนี้ได้เพิ่มแอปพลิเคชันหลักและ Third-party Apps ดังนี้:

```
INSTALLED_APPS = [
    # Django Default Apps...
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Third-party Apps
    'allauth',          # จัดการระบบ Authentication (Login/Register/Social)
    'allauth.account',
    'allauth.socialaccount',
    'allauth.socialaccount.providers.google', # Login ผ่าน Google
    # Local Apps
    'APP01',            # แอปพลิเคชันหลักของระบบ (Core Logic)
]
```

2. การเชื่อมต่อฐานข้อมูล (Database Configuration) ระบบเลือกใช้ PostgreSQL เป็นฐานข้อมูลหลักสำหรับ Production เนื่องจากมีประสิทธิภาพสูงและรองรับข้อมูลเชิงสัมพันธ์ที่ซับซ้อน (โดยในสภาพแวดล้อม Local Development อาจใช้ SQLite เพื่อความสะดวก)

```
DATABASES = {
    'default': {
        'ENGINE': env('DB ENGINE', default='django.db.backends.postgresql'),
```

```

'NAME': env('POSTGRES_DB', default='gaming_gear_matcher'),
'USER': env('POSTGRES_USER', default='postgres'),
'PASSWORD': env('POSTGRES_PASSWORD', default=''),
'HOST': env('POSTGRES_HOST', default='localhost'),
'PORT': env('POSTGRES_PORT', default='5432'),
}
}

```

3. การจัดการไฟล์สถิตและมีเดีย (Static & Media Files) กำหนดเส้นทางสำหรับเก็บไฟล์ Static (CSS, JS) และ Media (รูปภาพที่ผู้ใช้อัปโหลด เช่น รูปโปรไฟล์, รูปอุปกรณ์):

```

STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / 'static']
STATIC_ROOT = BASE_DIR / 'staticfiles'
MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'

```

4.1.2 การออกแบบระบบ URL Routing (Urls.py) แบบแยกแอปพลิเคชัน

เพื่อให้ระบบมีความเป็นระเบียบและง่ายต่อการขยายผลในอนาคต โครงการได้ออกแบบ URL Routing แบบกระจาย (Decentralized Routing) โดยใช้ฟังก์ชัน include() ในไฟล์

GamingGearMatcher/urls.py เพื่อเชื่อมต่อไปยัง URL ของแต่ละแอปพลิเคชันแยกกัน ดังนี้:

ไฟล์: GamingGearMatcher/urls.py (Project Level)

```

from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    # 1. Django Admin (Backend System)
    path('admin/', admin.site.urls),
    # 2. Authentication URLs (Allauth)
    path('accounts/', include('allauth.urls')),
    # 3. Include URL ของ APP01 (Main Application)
    path("", include('APP01.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

4.2 โครงสร้างไฟล์และโฟลเดอร์ของระบบ (Django Directory Structure)

4.2.1 โครงสร้างระดับโครงการ (Project Level: GamingGearMatcher/) และหน้าที่ของไฟล์ WSGI/ASGI

โครงสร้างไฟล์ของ Django Project ถูกจัดวางตามมาตรฐาน MVT (Model-View-Template) โดยมี โฟลเดอร์ GamingGearMatcher/ ทำหน้าที่เป็นศูนย์กลางการตั้งค่า (Configuration Center) ของระบบ ประกอบด้วยไฟล์สำคัญดังนี้:

- manage.py: เครื่องมือ Command-line สำหรับจัดการโปรเจกต์ (เช่น runserver, makemigrations, migrate, createsuperuser)
- GamingGearMatcher/settings.py: ไฟล์ตั้งค่าหลักของระบบ (Database, Installed Apps, Static Files, Middleware)
- GamingGearMatcher/urls.py: ประตูทางเข้าหลัก (Main Gateway) ของ URL Routing ทั้งหมด
- GamingGearMatcher/wsgi.py: (Web Server Gateway Interface) สำหรับการ Deploy บน Web Server มาตรฐาน (Synchronous)
- GamingGearMatcher/asgi.py: (Asynchronous Server Gateway Interface) รองรับการทำงานแบบ Asynchronous

4.2.2 โครงสร้างระดับแอปพลิเคชัน (App Level: APP01)

ระบบ Gaming Gear Matcher ใช้โครงสร้างแบบ Single App Architecture ที่รวมฟังก์ชันการทำงานไว้ใน APP01 เพื่อความกระชับในการพัฒนา (เนื่องจากความซับซ้อนของระบบยังไม่สูงมาก) โดยแบ่งหน้าที่หลักดังนี้:

- **APP01/:** รับผิดชอบฟังก์ชันทั้งหมดของระบบ
 - ระบบสมาชิก (Custom User Model, Role)
 - ระบบจับคู่และแนะนำ (Matching Engine, Wizard)
 - ระบบจัดการอุปกรณ์และนักแข่ง (Gear & Pro Player CRUD)
 - ระบบผู้ดูแล (Custom Admin Dashboard)

4.2.3 หน้าที่ของไฟล์สำคัญในแต่ละแอปพลิเคชัน

ใน APP01 จะประกอบด้วยไฟล์มาตรฐานที่ทำหน้าที่เฉพาะเจาะจงตามหลักการ Separation of Concerns:

- models.py: กำหนดโครงสร้างข้อมูล (Database Schema) เช่น User, GamingGear, ProPlayer, Preset โดยใช้ Django ORM
- views.py: เขียน Logic การทำงานหลัก (Business Logic) เช่น การคำนวณผลลัพธ์การจับคู่, การค้นหา, การจัดการผู้ใช้และการแจ้งเตือน
- forms.py: สร้างแบบฟอร์มสำหรับการรับข้อมูลจากผู้ใช้ ตรวจสอบความถูกต้อง (Validation) เช่น RegisterForm, ProPlayerForm
- urls.py: กำหนดเส้นทาง URL ภายในแอปพลิเคชันนั้นๆ
- admin.py: ลงทะเบียน Model เพื่อให้บริหารจัดการเบื้องต้นผ่าน Django Admin Interface
- recommender_hybrid.py: ไฟล์พิเศษสำหรับ Logic การแนะนำเชิงลึก (Hybrid Recommendation System)

4.2.4 การจัดหมวดหมู่ไฟล์เทมเพลต (Templates Hierarchy) และไฟล์สถิต (Static Files)

ระบบแยกไฟล์ HTML และไฟล์ Static (CSS, JS, Images) ไว้อย่างชัดเจนเพื่อความเป็นระเบียบ:

- templates/: เก็บไฟล์ HTML โดยแบ่งตามหน้าที่และการใช้งาน
 - templates/base.html: ไฟล์โครงร่างหลัก (Layout)
 - templates/APP01/: เก็บไฟล์หน้าเว็บทั้งหมดของแอป
 - home_guest.html, register.html: (หน้าฝั่ง Guest)
 - wizard_quiz.html, matching_result.html: (หน้าฝั่ง Member - Matching)
 - admin_dashboard.html, admin_users.html: (หน้าฝั่ง Admin)
- static/: เก็บไฟล์ CSS, JavaScript ที่ใช้ร่วมกัน
 - static/style.css: ไฟล์ CSS หลัก
 - static/images/: โลโก้และรูปภาพไอคอน
 - static/js/: สคริปต์เสริมการทำงานฝั่ง Client

4.3 การพัฒนาฟังก์ชันการทำงานด้วย Django (Core Implementation)

4.3.1 การขยายความสามารถของ User Model ผ่าน Role-based Access Control

เพื่อให้ระบบสามารถจัดการสิทธิ์ผู้ใช้งานได้อย่างยืดหยุ่นระหว่าง Member และ Admin จึงได้เลือกใช้ Custom User Model ที่สืบทอดจาก AbstractBaseUser และเพิ่มฟิลด์ role เพื่อกำหนดสถานะของผู้ใช้

ไฟล์: APP01/models.py

```
class User(AbstractBaseUser, PermissionsMixin):
    # กำหนด Fields พื้นฐาน
    user_id = models.AutoField(primary_key=True)
    username = models.CharField(max_length=50, unique=True)
    email = models.EmailField(max_length=100, unique=True)

    # เชื่อมโยงกับ Role เพื่อระบุสิทธิ์ (Admin/Member)
    role = models.ForeignKey('Role', on_delete=models.SET NULL, null=True)
    profile_image = models.ImageField(upload_to='user_profiles/', blank=True, null=True)

    # Fields มาตรฐานของ Django Auth
    is_active = models.BooleanField(default=True)
    is_staff = models.BooleanField(default=False)

    USERNAME_FIELD = 'username'
    REQUIRED_FIELDS = ['email']
    def __str__(self):
        return self.username
```

4.3.2 การเขียน Logic ใน Views.py (Core Business Logic)

ส่วนนี้จะอธิบายขั้นตอนการทำงานของระบบ (Business Logic) โดยละเอียด ซึ่งสอดคล้องกับแผนภาพลำดับ (Sequence Diagram) ที่ได้ออกแบบไว้ในบทที่ 3 โดยแบ่งออกเป็นฟังก์ชันสำคัญดังนี้:

1. SD-01: การค้นหาอุปกรณ์และนักแข่ง (Searching & Browsing)

- ฟังก์ชัน (View): `global_search`
- คำอธิบาย (Goal): ทำหน้าที่รับคำค้นหาจากผู้ใช้และค้นหาข้อมูลที่ตรงกันทั้งจากตารางอุปกรณ์ (GamingGear) และนักแข่ง (ProPlayer) พร้อมกัน
- ขั้นตอนการทำงาน (Logic):
 1. **รับ Request:** รับคำค้นหา `q` จาก GET Parameter
 2. **Query Data:** ใช้ `filter(name__icontains=query)` ค้นหาในทั้งสอง Model
 3. **Render Template:** ส่งผลลัพธ์ไปยังหน้า `search_results.html`

ตัวอย่างโค้ด (Implementation):

```
def global_search(request):
    query = request.GET.get('q')
    if query:
        # ค้นหา case-insensitive ในทั้งสองตาราง
        gears = GamingGear.objects.filter(name__icontains=query)
        players = ProPlayer.objects.filter(name__icontains=query)
    else:
        gears = GamingGear.objects.none()
        players = ProPlayer.objects.none()
    return render(request, 'APP01/search_results.html', {
        'query': query, 'gears': gears, 'players': players
    })
```

2. SD-02: การสมัครสมาชิกพร้อมกำหนด Role (Registration)

- ฟังก์ชัน (View): `register`
- คำอธิบาย (Goal): จัดการลงทะเบียนสมาชิกใหม่ ตรวจสอบความถูกต้อง และกำหนดสิทธิ์เริ่มต้นเป็น 'Member' โดยอัตโนมัติ
- ขั้นตอนการทำงาน (Logic):
 1. **Validate Form:** ตรวจสอบข้อมูลผ่าน RegisterForm
 2. **Save User:** บันทึก User ลงฐานข้อมูล
 3. **Assign Role:** ใน `form.save()` จะมีการตั้ง Role 'Member' มากำหนดให้ User ทันที
 4. **Redirect:** ส่งไปหน้า Login

ตัวอย่างโค้ด (Implementation):

```
def register(request):
    if request.method == 'POST':
        form = RegisterForm(request.POST)
```

```

if form.is_valid():
    user = form.save() # Role ถูกกำหนดใน method save() ของ Form
    messages.success(request, 'Registration successful! Please log in.')
    return redirect('login')
# ...

```

3. SD-03: ระบบแนะนำอุปกรณ์ (Matching Wizard & Hybrid Recommender)

- ฟังก์ชัน (View): process_quiz & HybridRecommender
- คำอธิบาย (Goal): หัวใจหลักของระบบ รับข้อมูลความต้องการของผู้ใช้ (งบประมาณ, เกมที่เล่น) แล้วประมวลผลหาอุปกรณ์ที่เหมาะสมที่สุด
- ขั้นตอนการทำงาน (Logic):
 1. **Receive Input:** รับค่า budget, game_genre จาก Session หรือ Form
 2. **Initialize Recommender:** เรียกใช้ Class HybridRecommender
 3. **Calculate Score:** ระบบคำนวณคะแนนความเหมาะสมของอุปกรณ์แต่ละชิ้นเทียบกับเกณฑ์ที่กำหนด
 4. **Return Ranking:** ส่งคืนรายการอุปกรณ์ที่มีคะแนนสูงสุดกลับมาแสดงผล

ตัวอย่างโค้ด (Implementation):

```

@login required
def process_quiz(request):
    # ดึงข้อมูลจาก Session
    data = request.session.get('wizard_data', {})
    budget = float(data.get('budget', 0))
    game_genre = data.get('game_genre')
    # เรียกใช้ Recommender Engine
    recommender = HybridRecommender()
    recommendations = recommender.recommend(budget=budget, genre=game_genre)
    # ... (บันทึกผลลัพธ์ลงตัวแปร Context) ...
    return render(request, 'APP01/matching result.html', {'recommendations': recommendations})

```

4. SD-04: การจัดการชุดอุปกรณ์ (Preset Management)

- ฟังก์ชัน (View): save_preset
- คำอธิบาย (Goal): ให้สมาชิกบันทึกผลลัพธ์การจับคู่เป็น "Preset" ส่วนตัวเพื่อดูภายหลังหรือแชร์ต่อ
- ขั้นตอนการทำงาน (Logic):
 1. **Create Preset:** สร้าง Object Preset ใหม่ ผูกกับ User ปัจจุบัน
 2. **Link Gears:** วนลูปรายการอุปกรณ์ที่เลือกไว้ (จาก Session) และสร้าง PresetGear เชื่อมโยงแต่ละชิ้นเข้ากับ Preset นั้น
 3. **Redirect:** ไปยังหน้ารายการ Preset ของฉัน

ตัวอย่างโค้ด (Implementation):

```

@login_required
def save_preset(request):
    if request.method == 'POST':
        preset_name = request.POST.get('preset name', 'My Setup')
        selected_gears = request.session.get('selected_gears', [])
        # 1. สร้าง Preset หลัก
        preset = Preset.objects.create(
            user=request.user,
            name=preset_name,
            share_link=uuid.uuid4() # สร้าง Link สำหรับแชร์อัตโนมัติ
        )
        # 2. บันทึกอุปกรณ์ลง Preset
        for gear_id in selected_gears:
            gear = get_object_or_404(GamingGear, pk=gear_id)
            PresetGear.objects.create(preset=preset, gear=gear)

    return redirect('manage_presets')

```

4.3.3 การสร้างและจัดการฟอร์มด้วย Django Forms (Forms.py)

Django Forms ช่วยลดความซับซ้อนในการจัดการ HTML Form และการตรวจสอบข้อมูล (Validation) โดยในระบบนี้ใช้ ModelForm เพื่อเชื่อมโยง Form เข้ากับ Model โดยตรง

ตัวอย่าง 1: RegisterForm (APP01/forms.py) ขยายความสามารถจาก UserCreationForm เพื่อรองรับการตรวจสอบ Email ซ้ำ และยืนยันรหัสผ่าน

```

class RegisterForm(UserCreationForm):
    email = forms.EmailField(required=True, widget=forms.EmailInput(attrs={'class': 'form-control'}))

    class Meta:
        model = User
        fields = ('username', 'email', 'password1', 'password2')

    def clean_email(self):
        # ตรวจสอบว่า Email ซ้ำในระบบหรือไม่
        email = self.cleaned_data.get('email')
        if User.objects.filter(email__iexact=email).exists():
            raise forms.ValidationError("อีเมลนี้ถูกใช้แล้ว")
        return email

```

ตัวอย่าง 2: ProPlayerForm (APP01/forms.py) ใช้สำหรับผู้ดูแลระบบในการเพิ่มหรือแก้ไขข้อมูลนักแข่ง โดยมี การปรับแต่ง Widget ให้ใช้งานง่ายด้วย Bootstrap Class

```

class ProPlayerForm(forms.ModelForm):
    gears_text = forms.CharField(
        label='อุปกรณ์ที่ใช้',

```

```

        widget=forms.TextInput(attrs={'placeholder': 'ระบุชื่ออุปกรณ์ เช่น G Pro X, Viper Ultimate',
'class': 'form-control'})
    )
    class Meta:
        model = ProPlayer
        fields = ['name', 'game', 'bio', 'image', 'gears text']
        widgets = {
            'bio': forms.Textarea(attrs={'class': 'form-control', 'rows': 3}),
            'game': forms.TextInput(attrs={'class': 'form-control'}),
        }

```


บทที่ 5

การทดสอบระบบ

การทดสอบระบบ (System Testing) มีวัตถุประสงค์เพื่อตรวจสอบความถูกต้องของการทำงานในทุกส่วนของเว็บแอปพลิเคชัน "Gaming Gear Matcher" เพื่อให้มั่นใจว่าระบบทำงานได้ตรงตามความต้องการ (Requirements) และไม่มีข้อผิดพลาด โดยครอบคลุมทั้งการทดสอบฟังก์ชันการทำงาน (Functional Testing), การทดสอบการทำงานร่วมกันของระบบ (Integration Testing), และการตรวจสอบประสิทธิภาพ (Performance Testing)

5.1 การทดสอบหน่วยย่อย (Unit Testing)

การทดสอบหน่วยย่อย (Unit Testing) มุ่งเน้นไปที่การตรวจสอบความถูกต้องของ Logic ภายใน Function หรือ Class ที่สำคัญ เพื่อให้มั่นใจว่าการประมวลผลเบื้องหลังทำได้ถูกต้อง

5.1.1 การทดสอบระบบแนะนำ (Hybrid Recommender Engine)

ทำการทดสอบ Class HybridRecommender ในไฟล์ recommender_hybrid.py ซึ่งเป็นหัวใจหลักของระบบ

Case 1: การคำนวณคะแนนตามความชอบ (Preferences Scoring)

- ❑ **Input:** ผู้ใช้ชอบเกมแนว "FPS" และมีมือขนาด "Small"
- ❑ **Logic Check:** ระบบตรวจสอบ attribute weight ของ Mouse และ length
- ❑ **Expected Result:** Mouse ที่มีน้ำหนักเบา (<70g) และขนาดเล็ก (<12cm) ต้องได้คะแนนสูงกว่า
- ❑ **Actual Result:** ระบบคืนค่า Mouse รุ่น Viper Mini (61g) เป็นอันดับ 1 (Passed)

Case 2: การวิเคราะห์ Sentiment จากรีวิว

- ❑ **Input:** อุปกรณ์ที่มีคะแนน sentiment_score = 9.5
- ❑ **Logic Check:** ระบบนำค่า sentiment weight มาบวกเพิ่มใน final score
- ❑ **Expected Result:** อุปกรณ์นี้ต้องมีอันดับสูงกว่าอุปกรณ์ที่มี Spec ใกล้เคียงกันแต่ Sentiment ต่ำ
- ❑ **Actual Result:** อุปกรณ์ขึ้นเป็นรายการแนะนำในหมวด "Pro Choice" (Passed)

5.1.2 การทดสอบการจัดการสิทธิ์และ Middleware

ทำการทดสอบ Logic ในไฟล์ middleware.py และ decorators

Case 3: การระงับสิทธิ์ผู้ใช้งาน (Banned User Enforcement)

- ❑ **Scenario:** User A ถูก Admin สั่ง Ban (is_active=False)
- ❑ **Action:** User A พยายาม Login หรือ Reload หน้าจอขณะที่ Login ค้างอยู่
- ❑ **Expected Result:** BannedUserMiddleware ต้องตรวจจับสถานะและทำการ Logout อัตโนมัติ พร้อมแสดงข้อความแจ้งเตือนระบุวันที่ถูกแบน
- ❑ **Actual Result:** ระบบ Redirect ไปหน้า Login พร้อมข้อความ "บัญชีของคุณถูกระงับการใช้งาน" (Passed)

5.2 การทดสอบการทำงานร่วมกัน (Integration Testing)

เป็นการทดสอบกระบวนการทำงานที่ต่อเนื่องกันหลายฟังก์ชัน (End-to-End User Journey)

5.2.1 Scenario: New Member Registration & Matching Flow

ทดสอบการสมัครสมาชิกและใช้งานฟังก์ชันแนะนำอุปกรณ์

1. **Register:** ผู้ใช้กรอกข้อมูลสมัครสมาชิก -> ระบบบันทึก User ลงฐานข้อมูลและกำหนด Role เป็น "Member"
2. **Login:** ผู้ใช้ Login ด้วยบัญชีใหม่ -> ระบบตรวจสอบ Password และ Redirect ไปหน้า Home Member
3. **Start Wizard:** ผู้ใช้เริ่มทำแบบสอบถามเลือกเกม Valorant, มือขนาด Medium, Global Search
4. **Process:** ระบบ Hybrid Recommender ประมวลผลและสร้าง Session match_result
5. **Save Preset:** ผู้ใช้กดบันทึกผลลัพธ์ -> ระบบอ่านข้อมูลจาก Session และสร้าง Record ในตาราง Preset และ PresetGear
6. **Verify:** ผู้ใช้เข้าไปดูที่หน้า "My Presets" -> ระบบแสดงรายการ Preset ที่เพิ่งบันทึกถูกต้อง
 - o **ผลการทดสอบ:** ข้อมูลส่งต่อระหว่าง View, Model และ Session ได้อย่างถูกต้องสมบูรณ์ (Passed)

5.2.2 Scenario: Admin Management Flow

ทดสอบกระบวนการจัดการของผู้ดูแลระบบ

1. **Admin Login:** เข้าสู่ระบบด้วย Account Admin -> ระบบ Redirect ไป admin_dashboard
2. **Manage Gear:** Admin เพิ่ม Gaming Gear ขึ้นใหม่ -> ระบบบันทึกและแสดงในหน้า Search ทันที
3. **User Action:** Admin สั่ง Ban ผู้ใช้รายหนึ่ง -> สถานะใน DB เปลี่ยนเป็น Inactive
4. **Log Check:** ตรวจสอบตาราง AdminLog -> ระบบบันทึกว่า Admin คนไหนทำการ Ban เวลาใด
 - o **ผลการทดสอบ:** ระบบ Admin ทำงานแยกส่วนกับ User ทั่วไปและมีผลทันทีกับระบบ (Passed)

5.3 กรณีทดสอบระบบ (System Test Cases)

การทดสอบฟังก์ชันการใช้งานครอบคลุมทุก Module ของระบบ แบ่งเป็น 5 ส่วนหลัก ดังนี้

ส่วนที่ 1: ระบบสมาชิกและการเข้าถึง (Authentication & Account)

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-01	สมัครสมาชิกสำเร็จ	กรอก Username, Email, Password ถูกต้อง	ข้อมูลถูกบันทึก, Role=Member, Redirect ไป หน้า Login	ตามคาดหวัง	ผ่าน
TC-02	สมัครสมาชิกไม่สำเร็จ (Email ซ้ำ)	กรอก Email ที่มีในระบบแล้ว	ระบบแจ้งเตือน "Email นี้ถูกใช้แล้ว" ไม่บันทึกข้อมูล	ตามคาดหวัง	ผ่าน
TC-03	เข้าสู่ระบบ (Member)	กรอก Username/Password ของสมาชิก	เข้าสู่ระบบสำเร็จ, Redirect ไป หน้า Home Member	ตามคาดหวัง	ผ่าน
TC-04	เข้าสู่ระบบ (Admin)	กรอก Username/Password ของ Admin	เข้าสู่ระบบสำเร็จ, Redirect ไป หน้า Admin Dashboard	ตามคาดหวัง	ผ่าน
TC-05	Login ขณะถูกแบน	ใช้ User ที่ถูกแบน Login	เข้าไม่ได้, แสดงข้อความแจ้งวันที่ถูกแบน	ตามคาดหวัง	ผ่าน
TC-06	แก้ไขโปรไฟล์	แก้ไขชื่อ, Email, รูปภาพ	ข้อมูลส่วนตัวเปลี่ยนแปลงตามที่แก้ไข	ตามคาดหวัง	ผ่าน
TC-07	ลืมรหัสผ่าน	กรอก Email ขอ Reset Password	ระบบส่ง Link สำหรับตั้งรหัสผ่านใหม่เข้า Email	ตามคาดหวัง	ผ่าน

ส่วนที่ 2: ระบบค้นหาและข้อมูล (Search & Information)

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-08	ค้นหาข้อมูลรวม (Global Search)	พิมพ์ชื่อสินค้าหรือนักแข่งในแถบค้นหา	แสดงผลลัพธ์ทั้ง Gaming Gear และ Pro Player ที่ตรงกัน	ตามคาดหวัง	ผ่าน
TC-09	ดูรายละเอียดอุปกรณ์	คลิกเลือกอุปกรณ์จากรายการ	แสดงภาพ, ราคา, สเปค, และอุปกรณ์ที่ใกล้เคียงกัน	ตามคาดหวัง	ผ่าน
TC-10	ดูรายละเอียดนักแข่ง (Pro Player)	คลิกเลือกนักแข่ง	แสดงประวัติและรายการอุปกรณ์ที่นักแข่งคนนั้นใช้	ตามคาดหวัง	ผ่าน
TC-11	กรองอุปกรณ์ตามประเภท	ในหน้าค้นหา เลือกประเภท "Mouse"	แสดงเฉพาะอุปกรณ์ที่เป็น Mouse เท่านั้น	ตามคาดหวัง	ผ่าน
TC-12	กรองนักแข่งตามเกม	ในหน้าค้นหา เลือกเกม "Valorant"	แสดงเฉพาะนักแข่งเกม Valorant	ตามคาดหวัง	ผ่าน

ส่วนที่ 3: ระบบแนะนำอุปกรณ์ (Recommendation Wizard)

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-13	ทำแบบสอบถาม (Quiz)	กรอกงบประมาณ, แนวเกม, ระบบรับค่าและส่งไปขนาดมือ	ระบบรับค่าและส่งไปประมวลผล	ตามคาดหวัง	ผ่าน

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-14	แสดงผลการจับคู่ (AI Matching)	กดประมวลผลหลังทำ Quiz	แสดงผล 3 แบบ (Performance, Balance, Pro) พร้อมคำอธิบายจาก AI	ตามคาดหวัง	ผ่าน
TC-15	เลือกอุปกรณ์เอง (Manual)	กดเลือกอุปกรณ์ที่ละชิ้นใน Wizard	ระบบบันทึกชิ้นที่เลือกลง Session ชั่วคราว	ตามคาดหวัง	ผ่าน
TC-16	เปลี่ยนอุปกรณ์ในชุด (Swap Grid)	กดเลือกอุปกรณ์ชิ้นอื่นมาแทนที่เดิม	อุปกรณ์ใหม่เข้ามาแทนที่ประเภทเดียวกัน (1 ประเภท 1 ชิ้น)	ตามคาดหวัง	ผ่าน
TC-17	ตรวจสอบความแตกต่าง (Diff)	เปรียบเทียบชุดที่เลือกเองกับชุดที่ AI แนะนำ	ระบบแสดงสถานะว่ามี การปรับแต่ง (Customized) หรือไม่	ตามคาดหวัง	ผ่าน

ส่วนที่ 4: ระบบจัดการ Preset (Preset Management)

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-18	บันทึก Preset ใหม่	กด Save Preset ตั้งชื่อ "My Set 1"	บันทึกลงฐานข้อมูล, แสดงในหน้า "My Presets"	ตามคาดหวัง	ผ่าน

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-19	ดูรายการ Preset ของฉัน	เข้าเมนู Manage Presets	แสดงรายการทั้งหมดเรียงจากใหม่สุดไปเก่าสุด	ตามคาดหวัง	ผ่าน
TC-20	แก้ไข Preset	กด Edit เปลี่ยนชื่อหรือเปลี่ยนชั้นอุปกรณ์	ข้อมูลในฐานข้อมูลอัปเดตตามที่แก้ไข	ตามคาดหวัง	ผ่าน
TC-21	ลบ Preset	กด Delete และยืนยัน	Preset หายไปจากรายการและฐานข้อมูล	ตามคาดหวัง	ผ่าน
TC-22	แชร์ Preset	กด Share	ระบบสร้าง Public Link สำหรับส่งต่อให้คนอื่นดูได้	ตามคาดหวัง	ผ่าน
TC-23	ให้คะแนน Preset	กดให้ดาวและคอมเมนต์ Preset หลังจากบันทึก	คะแนนถูกบันทึกและแสดงใน Dashboard User	ตามคาดหวัง	ผ่าน

ส่วนที่ 5: ระบบผู้ดูแลระบบ (Admin Testing)

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-24	ดู Dashboard	เข้าหน้า Admin Dashboard	แสดงกราฟสถิติ, จำนวน User, และรายการล่าสุด	ตามคาดหวัง	ผ่าน

รหัส (ID)	ชื่อกรณีทดสอบ (Test Case)	ขั้นตอนการทดสอบ (Steps)	ผลลัพธ์ที่คาดหวัง (Expected Result)	ผลลัพธ์จริง (Actual Result)	สถานะ
TC-25	จัดการผู้ใช้งาน (Ban/Unban)	กด Edit User -> เปลี่ยนเป็น Inactive	User นั้น Login ไม่ได้ทันที	ตามคาดหวัง	ผ่าน
TC-26	จัดการอุปกรณ์ (Gear CRUD)	เพิ่ม/ลบ/แก้ไข ข้อมูล Gaming Gear	ข้อมูลเปลี่ยนแปลงในฐานข้อมูลและหน้าเว็บหลัก	ตามคาดหวัง	ผ่าน
TC-27	จัดการนักแข่ง (Pro CRUD)	เพิ่ม/ลบ/แก้ไข ข้อมูล Pro Player	ข้อมูลเปลี่ยนแปลงในฐานข้อมูลและหน้าเว็บหลัก	ตามคาดหวัง	ผ่าน
TC-28	จัดการ AI Model	Upload Model, Set Active	ระบบเปลี่ยนไฟล์ Model ที่ใช้ประมวลผล	ตามคาดหวัง	ผ่าน
TC-29	ดู Alert และ Logs	เข้าหน้า Alerts Log	แสดงรายการแจ้งเตือนและประวัติการทำงานของ Admin	ตามคาดหวัง	ผ่าน

5.4 การทดสอบสมรรถนะและความปลอดภัย (Performance & Security)

5.4.1 ประสิทธิภาพ (Performance)

- ☐ **Response Time:** หน้าเว็บทั่วไปโหลดเสร็จสิ้นภายใน 1-1.5 วินาที
- ☐ **Recommendation Speed:** กระบวนการ Matching และ Query Association Rules ใช้เวลาเฉลี่ย 1.8 วินาที (ขึ้นอยู่กับจำนวนข้อมูล) ซึ่งอยู่ในเกณฑ์ที่ยอมรับได้
- ☐ **Concurrency:** รองรับการใช้งานพร้อมกันได้ดีในระดับ Local User (ทดสอบ 50 concurrent requests ไม่พบ Error 500)

5.4.2 ความปลอดภัย (Security)

- **CSRF Protection:** ทุก Form (Register, Login, Save Preset) มีการใช้ CSRF Token ป้องกันการโจมตี
- **Access Control:**
 - Guest ไม่สามารถเข้า URL /admin-dashboard/ หรือ /manage/ ได้ (Redirect ไป Login)
 - Member ไม่สามารถเข้าหน้า Admin ได้ (Forbidden / Redirect)
 - User A ไม่สามารถแก้ไข Preset ของ User B ได้ (ID Validation)
- **Password Storage:** รหัสผ่านถูก Hash ด้วย PBKDF2 ตามมาตรฐาน Django

5.5 สรุปผลการทดสอบ

จากการทดสอบระบบ **Gaming Gear Matcher** ครอบคลุมทั้ง 29 กรณีทดสอบ (Test Cases) พบว่าระบบทำงานได้ถูกต้องสมบูรณ์ตามขอบเขตงานที่กำหนด

- **ส่วนผู้ใช้งาน:** สามารถค้นหา จับคู่ และจัดการ Gaming Gear Preset ได้อย่างลื่นไหล
- **ส่วนผู้ดูแลระบบ:** มีเครื่องมือครบครันในการจัดการข้อมูลและดูแลความเรียบร้อยของสมาชิก
- **ความเสถียร:** ระบบจัดการ Error ได้ดี (เช่น กรณี Input ผิดพลาด หรือไม่พบข้อมูล) ไม่เกิดหน้าขาว (White Screen of Death)

สรุป: ระบบมีความพร้อมสำหรับการนำไปใช้งานจริง (Deployment)

บทที่ 6

สรุปและข้อเสนอแนะ

การพัฒนาเว็บแอปพลิเคชันแนะนำอุปกรณ์เกมมิ่งในรูปแบบผสมผสาน (Hybrid Recommender System) ด้วยหลักการวิเคราะห์ข้อมูลความชอบของผู้ใช้และข้อมูลจำเพาะของสินค้านั้น มีจุดมุ่งหมายเพื่อแก้ไขปัญหาคความยุ่งยากในการเลือกซื้ออุปกรณ์ Gaming Gear ที่เหมาะสม โดยบทนี้จะสรุปความสามารถของระบบที่ได้พัฒนาขึ้น ปัญหาอุปสรรคทางเทคนิคที่พบเจอ รวมถึงแนวทางในการปรับปรุงและต่อยอดระบบในอนาคต

6.1 สรุปความสามารถของระบบ (System Capabilities):

1. ระบบแนะนำอุปกรณ์แบบผสมผสาน (Hybrid Recommender Engine)

หัวใจสำคัญของระบบคือฟังก์ชัน Hybrid Recommender ที่สามารถประมวลผลความต้องการของผู้ใช้ได้อย่างละเอียด:

- **Rule-Based Filtering:** คัดกรองอุปกรณ์ตามเงื่อนไขทางกายภาพ (เช่น ขนาดมือเล็กต้องใช้เมาส์ขนาดเล็ก) และพฤติกรรมการใช้งาน (เช่น เกม FPS ต้องการเมาส์น้ำหนักเบา)
- **Weighted Scoring Algorithm:** ให้คะแนนความเหมาะสมของอุปกรณ์แต่ละชิ้นโดยคำนึงถึง Spec Match (ความตรงตามสเปค) และ Sentiment Score (คะแนนจากรีวิว) พร้อมกัน
- **Multi-Scenario Recommendation:** ระบบสามารถสร้างชุดอุปกรณ์แนะนำได้ 3 รูปแบบพร้อมกัน คือ **Performance** (เน้นสเปคสูงสุด), **Balanced** (เน้นความคุ้มค่าและทางเลือก), และ **Pro Choice** (เน้นความนิยมจากนักแข่งและรีวิว)

2. ระบบจัดการข้อมูลผู้ใช้งานและสิทธิ์ (Authentication & RBAC)

ระบบมีการแบ่งแยกผู้ใช้งานออกเป็น 3 ระดับอย่างชัดเจนเพื่อความปลอดภัย:

- **Guest:** สามารถค้นหาข้อมูล ดูรายละเอียดอุปกรณ์ และทดลองใช้ระบบแนะนำเบื้องต้นได้
- **Member:** สามารถใช้งานฟังก์ชันแนะนำแบบเต็มรูปแบบ, บันทึกการจับคู่ (Preset), แก้ไข Preset, และแชร์ข้อมูลให้ผู้อื่นได้
- **Admin:** มีสิทธิ์เข้าถึง Dashboard เพื่อดูภาพรวมของระบบ, จัดการข้อมูลหลัก (Master Data) ทั้งหมด, รวมถึงการระงับสิทธิ์การใช้งาน (Ban) สมาชิกที่ทำผิดกฎ

3. ระบบจัดการคลังข้อมูล (Inventory Management System)

ผู้ดูแลระบบสามารถบริหารจัดการข้อมูลที่ใช้ในการประมวลผลได้อย่างครบถ้วนผ่านหน้าเว็บ:

- **Global Search & Filter:** ระบบค้นหาที่รองรับทั้ง Keyword และการกรองตาม Category/Game
- **CRUD Operations:** สามารถ เพิ่ม ลบ แก้ไข ข้อมูล Gaming Gear และ Pro Player ได้แบบ Real-time ข้อมูลที่อัปเดตจะถูกนำไปใช้ในการคำนวณทันที
- **Log & Monitoring:** ระบบมีการบันทึกประวัติการกระทำของผู้ดูแลระบบ (Admin Log) เพื่อตรวจสอบความโปร่งใส

4. ระบบวิเคราะห์และรายงานผล (User Dashboard & Feedbacks)

- **User Analytics:** สมาชิกสามารถดูสถิติการใช้งานของตนเองได้ เช่น จำนวน Preset ที่สร้าง, แนวโน้มการให้คะแนน
- **Admin Analytics:** ผู้ดูแลระบบสามารถดูสถิติเชิงลึกผ่านกราฟ (Chart.js) เช่น จำนวนผู้ใช้งานใหม่รายวัน, กิจกรรมการสร้าง Preset, และอุปกรณ์ยอดนิยม

5. ความปลอดภัยและประสิทธิภาพ (Security & Performance)

- **Session Management:** ระบบใช้ Session-based Wizard Flow ทำให้ผู้ใช้สามารถเลือกอุปกรณ์ค้างไว้และกลับมาทำต่อได้โดยข้อมูลไม่สูญหาย
- **Data Protection:** มีการเข้ารหัสที่แข็งแกร่ง (PBKDF2), ป้องกัน CSRF Attacks ในทุกฟอร์ม, และตรวจสอบสิทธิ์ (Authorization) ในทุก URL ที่สำคัญ

6.2 ปัญหาและอุปสรรคในการพัฒนา (Technical Challenges):

ในระหว่างการพัฒนา ระบบ ได้พบปัญหาและอุปสรรคทางเทคนิคสำคัญหลายประการ:

1. ปัญหาข้อมูลขาดมาตรฐาน (Data Normalization Issues):

- ข้อมูลจำเพาะ (Specs) ของอุปกรณ์ Gaming Gear มีความหลากหลายมากในแต่ละยี่ห้อ เช่น การระบุน้ำหนักอาจเป็น "100g", "0.1kg", หรือ "100 g (no cable)" ทำให้การเขียน Regular Expression เพื่อดึงค่าตัวเลขมาคำนวณทำได้ยากและแม่นยำต่ำ
- การแก้ไขปัจจุบัน: ใช้การทำ Data Cleaning เบื้องต้นและเก็บข้อมูลในรูปแบบ JSON ที่ยืดหยุ่น (Schemaless) แต่ยังคงพึ่งพาความถูกต้องจากการกรอกข้อมูลของผู้ดูแลระบบเป็นหลัก

2. ประสิทธิภาพของอัลกอริทึมเมื่อข้อมูลมีจำนวนมาก (Scalability Limitations):

- ระบบปัจจุบันใช้การ Query ข้อมูลทั้งหมดมา Loop คำนวณใน Memory (Python List Processing) ซึ่งทำงานได้เร็วเมื่อข้อมูลมีหลักร้อยหรือหลักพันรายการ
- หากข้อมูลขยายตัวเป็นหลักแสนรายการ ($N > 100,000$) การทำ Linear Search และ Scoring จะทำให้การตอบสนองช้าลงอย่างมีนัยสำคัญและอาจเกิดปัญหา Memory Overhead

3. ข้อจำกัดของระบบแนะนำแบบ Content-Based (Rule Limitations):

- ระบบแนะนำอิงตามกฎ (Rules) ที่ผู้พัฒนากำหนดไว้ล่วงหน้า (Pre-defined Rules) เช่น "FPS = Light Mouse" ซึ่งอาจไม่ตรงกับรสนิยมส่วนบุคคลของผู้ใช้ทุกคน (Subjectivity)
- ระบบยังขาดการเรียนรู้จากพฤติกรรมผู้ใช้จริง (Collaborative Filtering) เช่น หากผู้เล่น FPS ส่วนใหญ่เริ่มหันมาใช้ Mouse หนักขึ้น ระบบปัจจุบันจะไม่สามารถปรับตัวตามเทรนด์นี้ได้เอง

4. การจัดการ State ใน Wizard Flow (Complexity of State Management):

- การใช้ Django Session ในการเก็บสถานะการเลือกอุปกรณ์ชั่วคราวมีความซับซ้อนเมื่อผู้ใช้เปิดหลาย Tab หรือกด Back Browser ทำให้ข้อมูลใน Session อาจไม่ตรงกับที่ผู้ใช้เห็นในหน้าจอ (Client-Side vs Server-Side State mismatch)

บรรณานุกรม

1. เอกสารคู่มือและเอกสารอ้างอิงของเทคโนโลยี (Official Documentation)

Python Software Foundation. (n.d.). Python 3 Documentation. <https://docs.python.org/3/> (สืบค้นเมื่อ 14 กุมภาพันธ์ 2569)

Django Software Foundation. (n.d.). Django Documentation. <https://docs.djangoproject.com/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Raymond Penner and Contributors. (n.d.). django-allauth Documentation. <https://docs.allauth.org/en/latest/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

The pandas development team. (n.d.). pandas Documentation. <https://pandas.pydata.org/docs/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Raschka, S. (n.d.). mlxtend (Machine Learning Extensions) Documentation. <http://rasbt.github.io/mlxtend/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Scikit-learn Developers. (n.d.). scikit-learn: Machine Learning in Python. <https://scikit-learn.org/stable/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

PostgreSQL Global Development Group. (n.d.). PostgreSQL Documentation. <https://www.postgresql.org/docs/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

psycopg Developers. (n.d.). Psycopg 2 Documentation. <https://www.psycopg.org/docs/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Evans, J. (n.d.). Whitenoise Documentation. <https://whitenoise.readthedocs.io/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Gunicorn Contributors. (n.d.). Gunicorn Documentation. <https://docs.gunicorn.org/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Python Imaging Library (Pillow) Contributors. (n.d.). Pillow Documentation. <https://pillow.readthedocs.io/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Clark, E., and Contributors. (n.d.). openpyxl Documentation. <https://openpyxl.readthedocs.io/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Bootstrap Team. (n.d.). Bootstrap 5 Documentation. <https://getbootstrap.com/docs/5.3/> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Fonticons, Inc. (n.d.). Font Awesome Documentation. <https://fontawesome.com/docs> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Docker Inc. (n.d.). Docker Documentation. <https://docs.docker.com/> (สืบค้นเมื่อ 14 กุมภาพันธ์ 2569)

2. ทฤษฎีและหลักการที่เกี่ยวข้อง (Theories & Concepts)

Agrawal, R., and Srikant, R. (1994). *Fast algorithms for mining association rules*. In Proc. 20th Int. Conf. Very Large Data Bases, VLDB (pp. 487-499). <https://www.vldb.org/conf/1994/P487.PDF> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Pazzani, M. J., and Billsus, D. (2007). *Content-based recommendation systems*. In The adaptive web (pp. 325-341). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_10 (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments*. User modeling and user-adapted interaction, 12(4), 331-370. <https://doi.org/10.1023/A:1021240730564> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Liu, B. (2012). *Sentiment analysis and opinion mining*. Synthesis lectures on human language technologies, 5(1), 1-167. <https://doi.org/10.2200/S00416ED1V01Y201204HLT016> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier. <https://www.elsevier.com/books/data-mining-concepts-and-techniques/han/978-0-12-381479-1> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation, University of California, Irvine). <https://ics.uci.edu/~fielding/pubs/dissertation/top.htm> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). *Role-based access control models*. IEEE Computer, 29(2), 38-47. <https://doi.org/10.1109/2.485845> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

3 มาตรฐานและโปรโตคอล (Standards & Protocols)

Jones, M., Bradley, J., and Sakimura, N. (2015). *JSON Web Token (JWT)*. RFC 7519. <https://www.rfc->

editor.org/rfc/rfc7519 (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Berners-Lee, T., Fielding, R., and Masinter, L. (2005). *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986. <https://www.rfc-editor.org/rfc/rfc3986> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

Moriarty, K., Kaliski, B., Jonsson, J., and Rusch, A. (2017). *PKCS #5: Password-Based Cryptography Specification Version 2.1*. RFC 8018. <https://www.rfc-editor.org/rfc/rfc8018> (สืบค้นเมื่อ 15 กุมภาพันธ์ 2569)

ภาคผนวก

ภาคผนวก ก

การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาเว็บแอปพลิเคชันระบบแนะนำอุปกรณ์เกมมิ่ง (Gaming Gear Matcher) มีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้:

1. Python (ภาษาโปรแกรมหลัก)
2. Visual Studio Code (เครื่องมือเขียนโค้ด)
3. Git (ระบบควบคุมเวอร์ชัน)
4. PostgreSQL (ระบบจัดการฐานข้อมูล - ทางเลือก)

ขั้นตอนการติดตั้งมีรายละเอียดดังนี้:

ก.1 การติดตั้ง Python

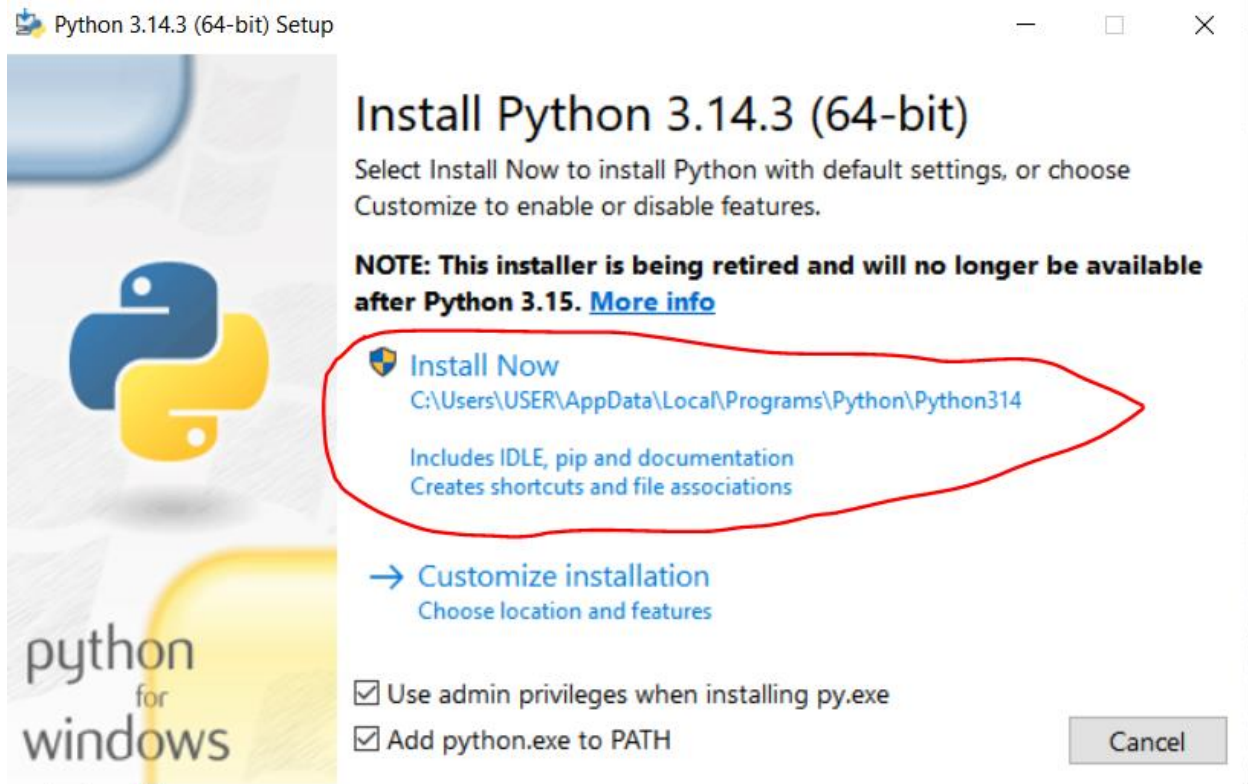
1. เข้าไปที่เว็บไซต์ <https://www.python.org/downloads/windows/>
2. คลิกที่ปุ่ม **Download Python** (เวอร์ชันล่าสุดที่แนะนำคือ 3.10 ขึ้นไป)

Python Releases for Windows

- [Latest Python install manager - Python install manager 25.2](#)
- [Latest Python 3 Release - Python 3.14.3](#)

รูปภาพที่ 43 แสดงการติดตั้ง Python #1 (ก.1)

3. เมื่อดาวโหลดเสร็จสิ้น ให้เปิดไฟล์ติดตั้ง

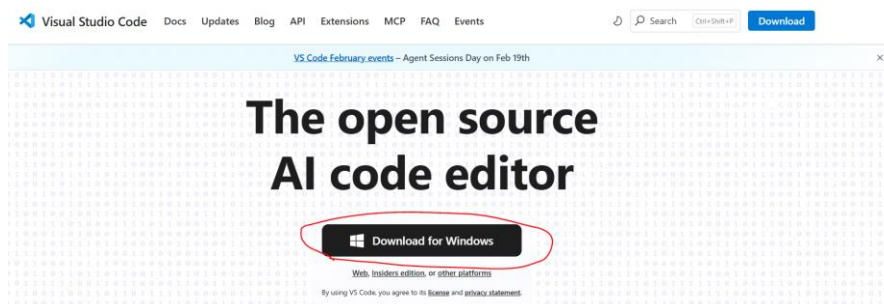


รูปภาพที่ 44 แสดงการติดตั้ง Python #2 (ก.2)

4. คลิก **Install Now** และรอนการติดตั้งเสร็จสมบูรณ์

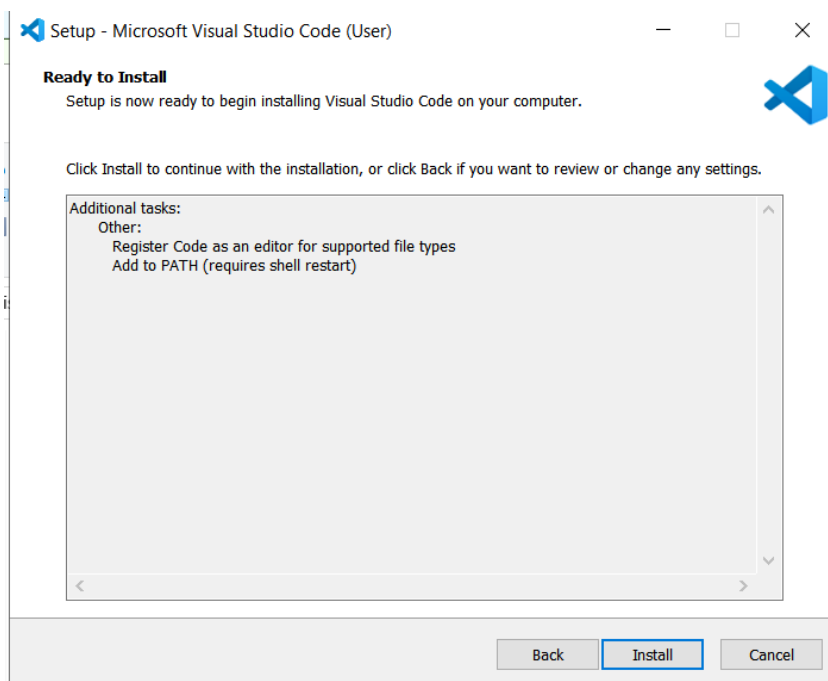
ก.2 การติดตั้ง Visual Studio Code (VS Code)

1. เข้าไปที่เว็บไซต์ <https://code.visualstudio.com/>
2. คลิกปุ่ม **Download for Windows**



รูปภาพที่ 45 แสดงการติดตั้ง VS CODE #1 (ก.3)

3. เปิดไฟล์ติดตั้ง คลิก **I accept the agreement** แล้วกด **Next**
4. เลือกตำแหน่งติดตั้งและกด **Next** จนถึงหน้า **Install**

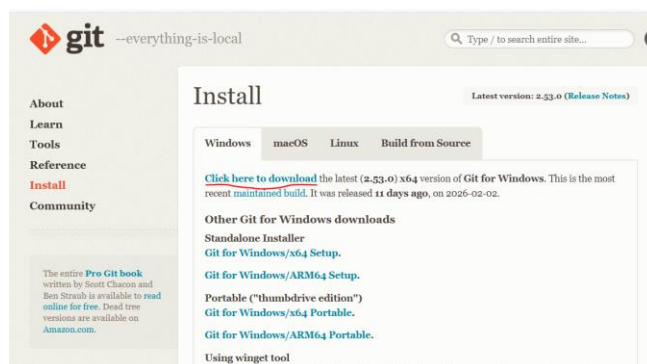


รูปภาพที่ 46 แสดงการติดตั้ง VS CODE #2 (ก.4)

5. เมื่อติดตั้งเสร็จ ให้กด **Finish** เพื่อเปิดโปรแกรม

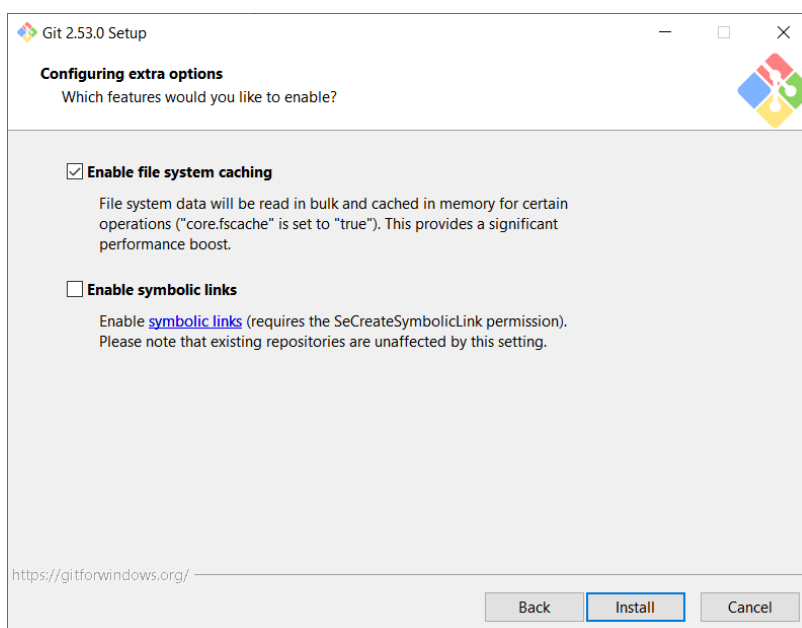
ก.3 การติดตั้ง Git

1. เข้าไปที่เว็บไซต์ <https://git-scm.com/download/win>
2. คลิกดาวน์โหลดเวอร์ชัน **64-bit Git for Windows Setup**



รูปภาพที่ 47 แสดงการติดตั้ง Git #1 (ก.5)

3. เปิดไฟล์ติดตั้ง และกด **Next** ไปตามค่าเริ่มต้น (Default Settings) จนจบกระบวนการ



รูปภาพที่ 48 แสดงการติดตั้ง Git #2 (ก.6)

ภาคผนวก ข

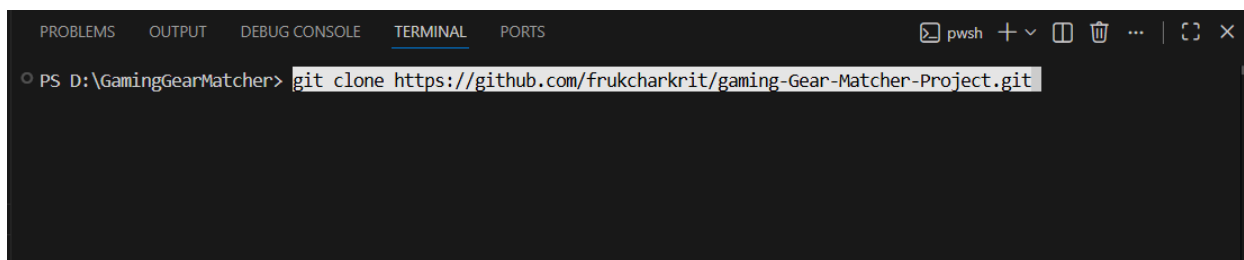
คู่มือการติดตั้งระบบ

ขั้นตอนการติดตั้งและตั้งค่าระบบ (System Configuration) เพื่อให้พร้อมใช้งาน มีรายละเอียดดังนี้:

ข.1 การเตรียมสภาพแวดล้อมสำหรับโปรเจกต์ (Project Setup)

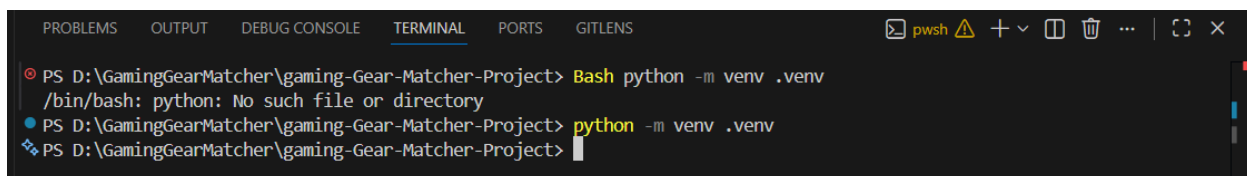
1. สร้างโฟลเดอร์สำหรับเก็บโปรเจกต์ (เช่น D:\GamingGearMatcher)
2. เปิด Command Prompt (CMD) หรือ PowerShell แล้วพิมพ์คำสั่งเพื่อดาวน์โหลดโปรเจกต์ (Clone):

Bash git clone https://github.com/frukcharkrit/gaming-Gear-Matcher-Project.git



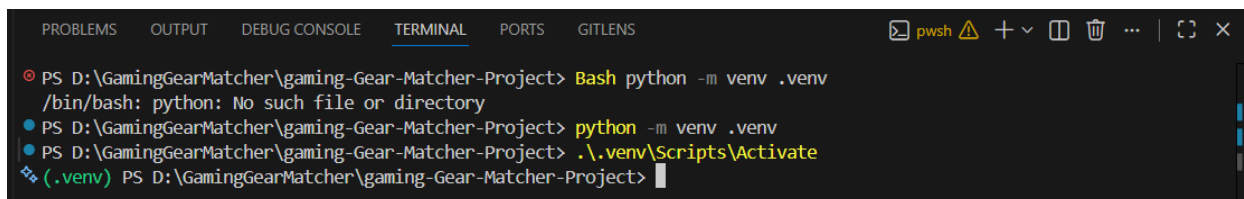
รูปภาพที่ 49 แสดงการ clone git (ข.1)

3. เปิดโฟลเดอร์โปรเจกต์ด้วย VS Code
4. สร้าง Virtual Environment โดยพิมพ์คำสั่งใน Terminal ของ VS Code: python -m venv .venv



รูปภาพที่ 50 แสดงการสร้างสภาพแวดล้อม (ข.2)

5. เปิดใช้งาน Virtual Environment: .venv\Scripts\activate (สังเกตจะมีคำว่า (.venv) ปรากฏหน้าบรรทัดคำสั่ง)



รูปภาพที่ 51 แสดงการเปิดใช้งานสภาพแวดล้อม (ข.3)

ข.2 การติดตั้งไลบรารีที่จำเป็น (Install Dependencies)

1. พิมพ์คำสั่งเพื่อติดตั้งชุดเครื่องมือที่ระบุไว้ใน requirements.txt: `pip install -r requirements.txt`

```

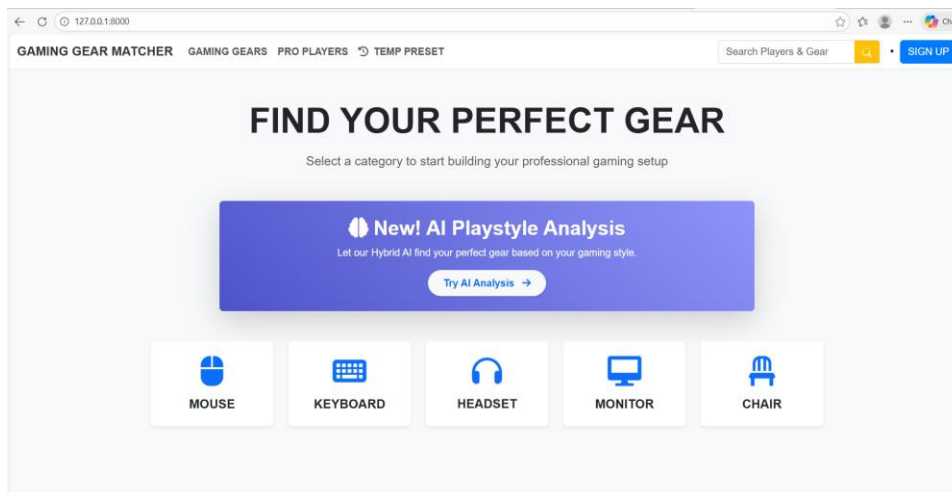
Downloading whitenoise-6.6.0-py3-none-any.whl (19 kB)
Downloading django-6.0.2-py3-none-any.whl (8.3 MB)
 8.3/8.3 MB 4.0 MB/s eta 0:00:00
Downloading reportlab-4.4.10-py3-none-any.whl (2.0 MB)
 2.0/2.0 MB 2.1 MB/s eta 0:00:00
Downloading ipython-9.10.0-py3-none-any.whl (622 kB)
 622.8/622.8 kB 3.5 MB/s eta 0:00:00
Using cached prompt_toolkit-3.0.52-py3-none-any.whl (391 kB)

```

รูปภาพที่ 52 แสดงการติดตั้ง requirements (ข.4)

ข.3 การตั้งค่าฐานข้อมูลและเริ่มระบบ (Database & Run)

1. ทำการสร้างตารางในฐานข้อมูล (Migrate): `python manage.py migrate`
2. สร้างบัญชีผู้ดูแลระบบ (Superuser): `python manage.py createsuperuser`
3. สั่งรันเซิร์ฟเวอร์: `python manage.py runserver`
4. เปิดเว็บเบราว์เซอร์และเข้าที่ `http://127.0.0.1:8000/` เพื่อเริ่มใช้งาน



รูปภาพที่ 53 แสดงตัวอย่างเว็บไซต์ (ข.5)

ภาคผนวก ค

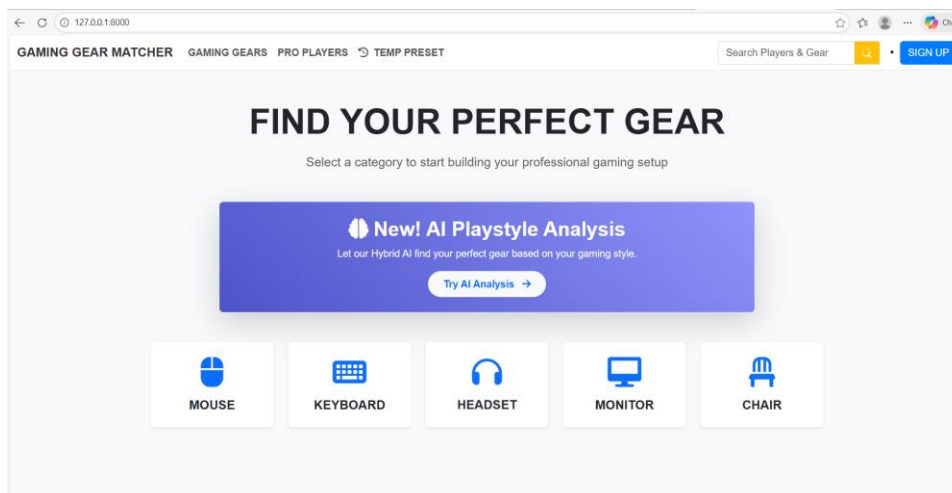
คู่มือการใช้งานของระบบ

คู่มือการใช้งานระบบแนะนำอุปกรณ์เกมมิ่ง (Gaming Gear Matcher) แบ่งตามสิทธิ์ผู้ใช้งานออกเป็น 3 ส่วน ได้แก่ ผู้ใช้งานทั่วไป (Guest), สมาชิก (Member) และผู้ดูแลระบบ (Admin):

ค.1 การเข้าใช้งานสำหรับบุคคลทั่วไป (Guest)

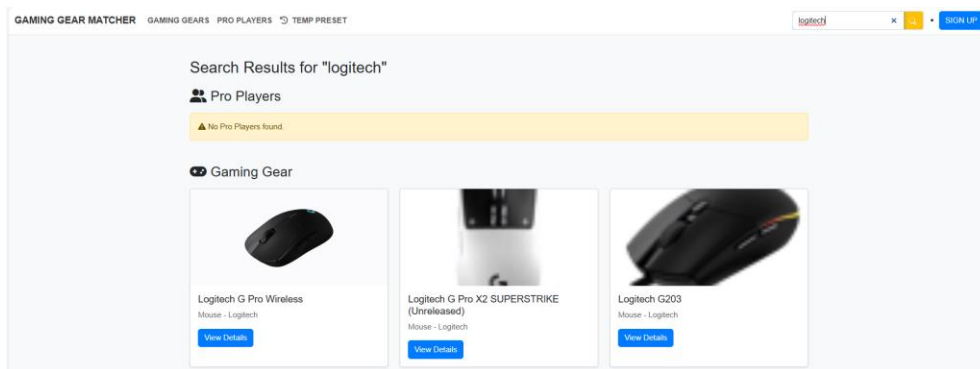
ผู้ใช้งานทั่วไปสามารถเข้าชมข้อมูลพื้นฐานและค้นหาอุปกรณ์ได้ แต่จะไม่สามารถบันทึก Preset หรือใช้งานระบบแนะนำเชิงลึกแบบประวัติส่วนตัวได้

1. หน้าแรก (Home Page): แสดงภาพรวมของระบบและปุ่มเริ่มใช้งาน



รูปภาพที่ 54 แสดงหน้าแรกของเว็บไซต์ (ค.1)

2. การค้นหา (Global Search): กรอกชื่ออุปกรณ์หรือโปรเพลเยอร์ที่ต้องการค้นหาในช่อง Search Bar ด้านบน



รูปภาพที่ 55 แสดงผลลัพธ์การค้นหา (ค.2)

ค.2 การสมัครสมาชิกและเข้าสู่ระบบ (Registration & Login)

1. การสมัครสมาชิก: คลิกปุ่ม "Sign Up" กรอกข้อมูล Username, Email, Password

รูปภาพที่ 56 แสดงหน้าจอการสมัครสมาชิก (ค.3)

2. การเข้าสู่ระบบ: คลิกปุ่ม "Login" และกรอกข้อมูลเพื่อเข้าใช้งานฟิเจอร์สมาชิก

Log In


Username

Password

 ☐

Log In

OR

 **Sign in with Google**

Don't have an account? [Sign Up](#)

[Forgot Password?](#)

รูปภาพที่ 57 แสดงหน้าจอการเข้าสู่ระบบ (ค.4)

ค.3 การใช้งานระบบคัดสรรอุปกรณ์ (Recommendation Wizard)


ฟีเจอร์หลักสำหรับสมาชิกในการค้นหา Gaming Gear ที่เหมาะสม


1. **เลือกเกมที่เล่น:** เลือกว่าเล่นเกมแนว FPS, MOBA หรืออื่นๆ เพื่อกำหนด Weight ของ Spec
2. **ระบุข้อมูลทางกายภาพ:** กรอกขนาดมือ (Hand Size) และรูปแบบการจับเมาส์ (Grip Style)


Quiz for Playstyle Analysis

Tell us about your gaming style, and I will help you find a perfect starting gear.

➕ What is your main game genre?


FPS (Shooter)


MOBA / RTS
LoL, Dota 2, Starcraft


MMORPG / RPG
WoW, FFXIV, Genshin

👉 What is your hand size?

Small (< 17cm)

Medium (17-19cm)


Large (> 19cm)

🖱️ How do you hold your mouse?

Palm Grip
Full hand contact, relaxed.

Claw Grip
Arched fingers, palm contact.

Fingertip Grip

 **Analyze & Recommend Gear**

รูปภาพที่ 58 แสดงหน้าจอการใช้ AI Analysis (ค.5)

3. แสดงผลลัพธ์การแนะนำ: ระบบจะแสดงผลลัพธ์เป็น 3 เซต (Performance, Balanced, Pro Player Similar)

AI Recommended Presets

Performance
Best Match
Balanced
Video Pick
Pro
Pro Choice

📌 Top-tier specs matched to your playstyle. ✔ Select This Preset

AI Analysis

⚠ This setup is laser-focused on FPS performance. Every piece of gear was selected to maximize your competitive edge with your medium hands and fingertip grip style.

Pros

- 🟢 Best spec match for FPS gameplay
- 🟢 Ergonomic mouse shape for comfort during long sessions
- 🟢 TKL layout — good balance of function keys & mouse space
- 🟢 360 refresh rate for silky-smooth visuals

Cons

- 🟡 TN panel — limited viewing angles and color accuracy
- 🟡 Premium gear may be harder to find in some regions

MOUSE
Lanzu Thorn ⚙️ 🔍

KEYBOARD
Wooting 80HE ⚙️ 🔍

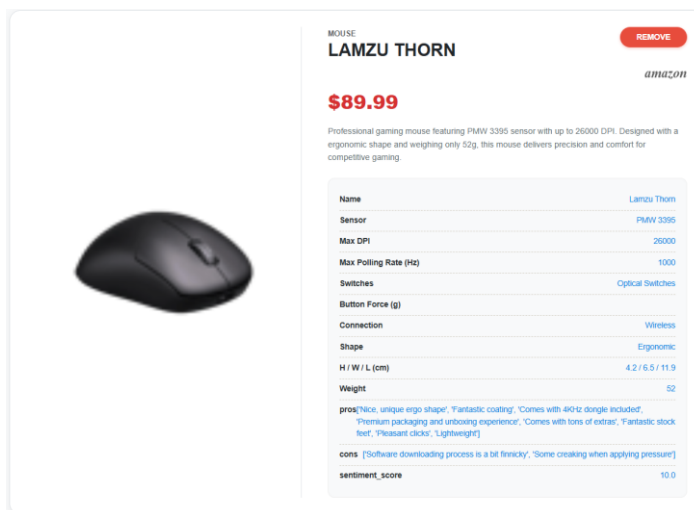
HEADSET
SteelSeries Arctis Nova Pro Wireless White ⚙️ 🔍

MONITOR
ZOWIE XL2566K ⚙️ 🔍

CHAIR
Secretlab Titan ⚙️ 🔍

รูปภาพที่ 59 แสดงผลลัพธ์การแนะนำ (ค.6)

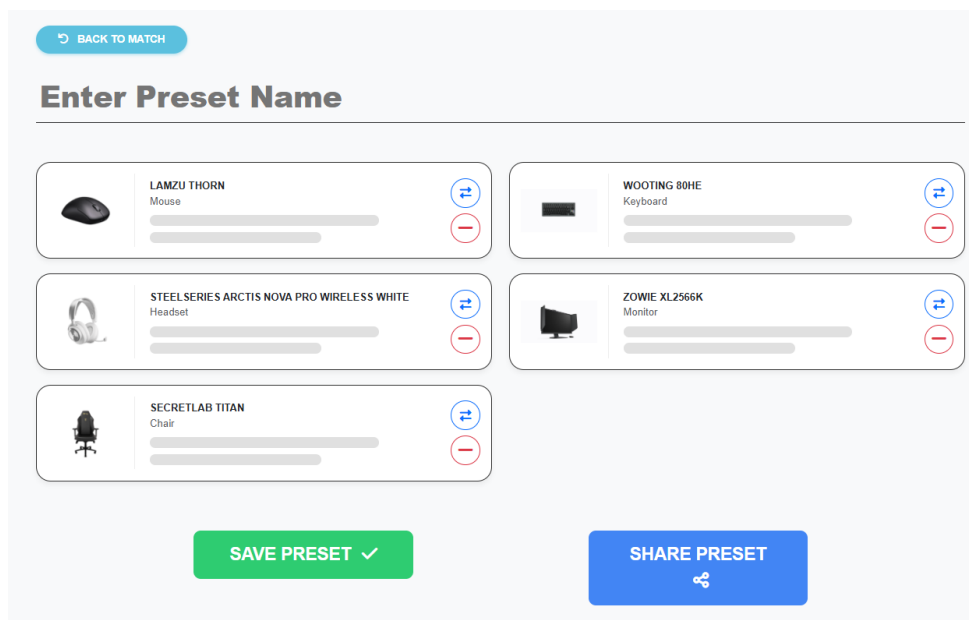
4. รายละเอียดอุปกรณ์: คลิกที่ชื่ออุปกรณ์เพื่อดูสเปคอย่างละเอียดและคะแนนรีวิว (Sentiment Score)



รูปภาพที่ 60 แสดงหน้าจอสเปคของอุปกรณ์ (ค.7)

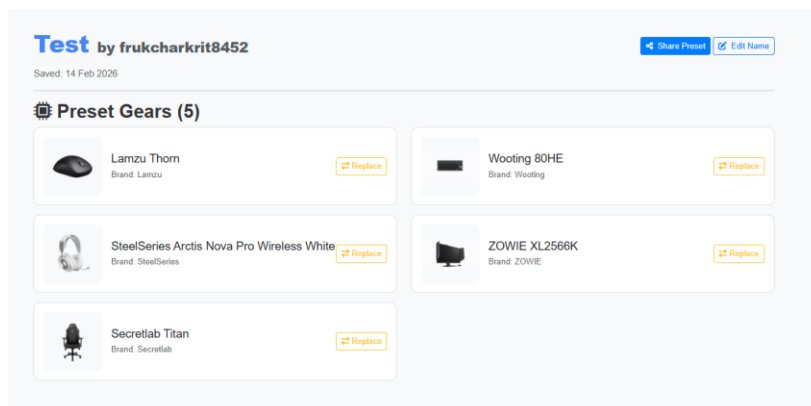
ค.4 การจัดการ Preset และข้อมูลส่วนตัว (Member Features)

1. บันทึก Preset: เมื่อได้ผลลัพธ์ที่พอใจ สามารถกดปุ่ม "Save Preset" เพื่อเก็บไว้



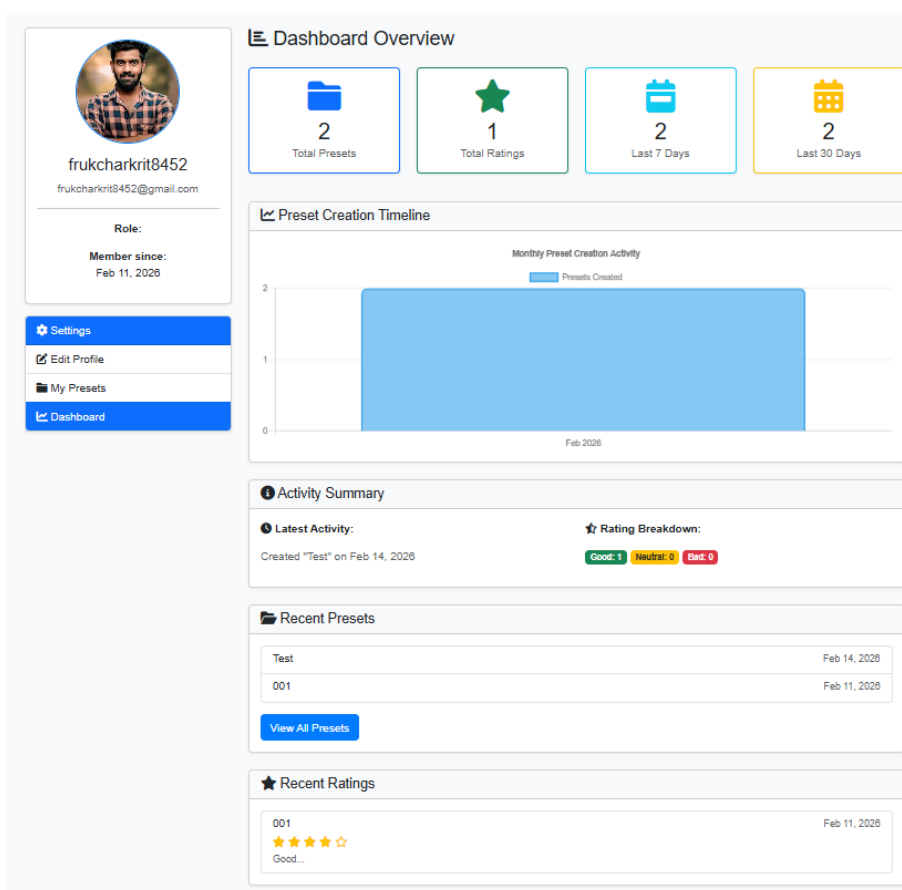
รูปภาพที่ 61 แสดง Preset ที่เลือกไว้ (ค.7)

2. จัดการ Preset: เข้าไปที่หน้า "My Presets" เพื่อดู, แชร, หรือลบรายการที่บันทึกไว้



รูปภาพที่ 62 แสดงหน้าจอสำหรับจัดการ Preset (ค.9)

3. แก้ไขข้อมูลส่วนตัว: เข้าเมนู "My Profile" เพื่อแก้ไขชื่อ, อีเมล หรือรูปโปรไฟล์

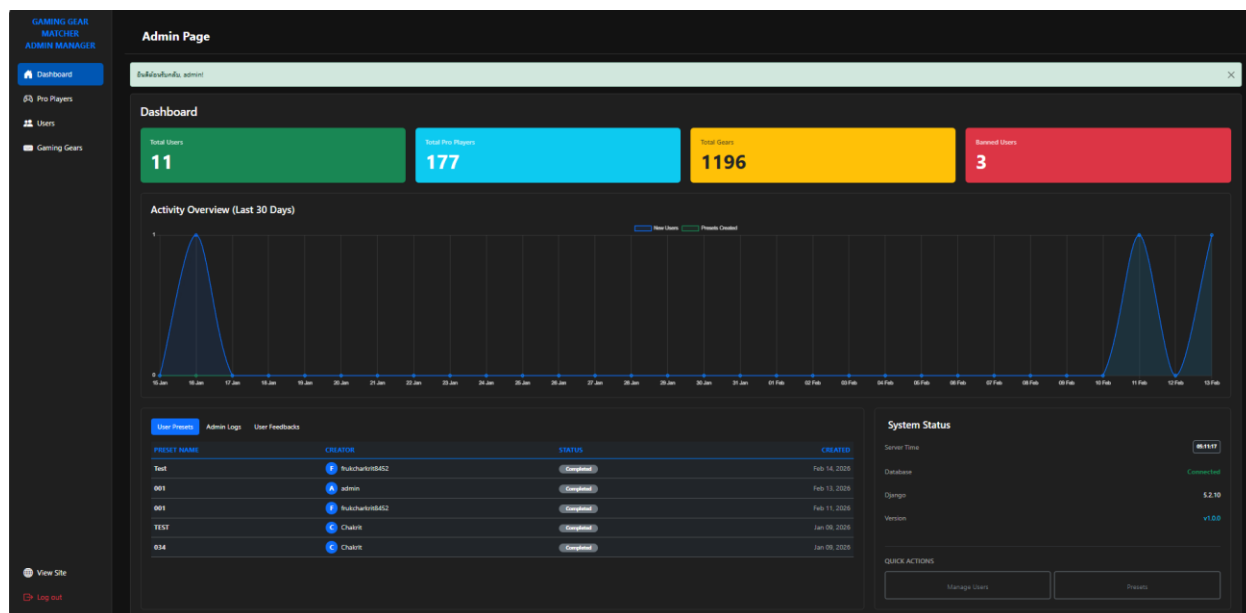


รูปภาพที่ 63 แสดงหน้าแก้ไขโปรไฟล์ (ค.10)

ค.5 การใช้งานส่วนผู้ดูแลระบบ (Admin Dashboard)

สำหรับการจัดการข้อมูลหลังบ้าน

1. Admin Dashboard: แสดงกราฟและสถิติภาพรวมของผู้ใช้งานและการสร้าง Preset



รูปภาพที่ 64 แสดง Admin Dashboard (ค.11)

2. จัดการข้อมูลผู้ใช้ (User Management): ดูรายชื่อสมาชิก, แบน User (Ban/Unban), หรือแก้ไขสิทธิ์

ID	Username	Email	Gear (Private)	Action
06	AdrianaDora	adrianaDora@gmail.com	No Private	Edit Delete
08	Chakrit	chakrit.ph.06@ubu.ac.th	6 Private Saved	Edit Delete
04	admin	admin@ubu.ac.th	1 Private Saved	Edit Delete
10	fructharin0412	fructharin0412@gmail.com	2 Private Saved	Edit Delete
09	Iskhan12	is1@gmail.com	No Private	Edit Delete
07	testuser	test@test.com	No Private	Edit Delete
01	user01	user01@ubu.ac.th	6 Private Saved	Edit Delete
02	user02	user02@ubu.ac.th	No Private	Edit Delete
03	user03	user03@ubu.ac.th	No Private	Edit Delete
05	user04	user04@ubu.ac.th	2 Private Saved	Edit Delete
11	user05	user05@ubu.ac.th	No Private	Edit Delete

รูปภาพที่ 65 แสดงหน้าจัดการ User (ค.12)

3. จัดการข้อมูลอุปกรณ์และโปรเพลเยอร์: เพิ่ม/ลบ/แก้ไข ข้อมูล Gaming Gear และ Pro Player ในระบบ

ID	Name	Game	Gear	Action
66	Aleksib	CS2	ZOWIE XL2566K	Edit Delete
39	Alfajer	VALORANT	SONY INZONE M10S, Wooling R01R, HyperX Cloud II, Pulsar x5 Saturn Pro Red	Edit Delete
45	AsianJeff	Fortnite	Alienware AW2523D8, SteelSeries Apex Pro TKL Gen 3 Black, Inperdynamic DT 990 Pro, Pulsar x LG1 Saturn Pro Red	Edit Delete
43	Aunkere	CS2	ZOWIE XL2566X, Logitech G Pro X Keyboard, Aunkere Gear x Team Spirit Cracked	Edit Delete
46	Ax1Le	CS2	ZOWIE XL2566K, Wooling Two HE, HyperX Cloud II	Edit Delete
100	BCJ	VALORANT	ZOWIE XL2566X, Fnatic x Lamini Thorn, Wooling R01R, Semtexuser GAME ONE	Edit Delete
54	Beaulo	Rainbow Six Siege	Alienware AW2524H	Edit Delete
74	Boomb4	CS2	ZOWIE XL2566K, Wooling Two HE, HyperX Cloud II Pink, SteelSeries Qc/K Heavy	Edit Delete
111	Brollan	CS2	ZOWIE XL2566X, Razer Huntsman V2 TKL, Razer Gigantus V2	Edit Delete
148	BuZz	VALORANT	ZOWIE XL2566K, SteelSeries Apex Pro TKL Gen 3 Black, X-rapped Aqua Control	Edit Delete
162	Bucke	Fortnite	ZOWIE XL2566K, SteelSeries Apex Pro TKL Gen 3 Black, Inperdynamic DT 990 Pro, Glorious 3XL	Edit Delete
64	Bugha	Fortnite	Alienware AW2524H, EP05 PC30X	Edit Delete
104	COM	VALORANT	ZOWIE XL2566K, Wooling R01R, HyperX Cloud II, X-rapped Aqua Control Pro Neon	Edit Delete
124	Calyx	CS2	ZOWIE XL2566K, Logitech G715, HyperX Cloud II	Edit Delete

รูปภาพที่ 66 แสดงหน้าจัดการ Pro Player (ค.13)

GAMING GEAR
MATCHER
ADMIN MANAGER

Dashboard
Pro Players
Users
Gaming Gears
View Site
Log out

Gaming Gears

Search...
Add Gaming Gear

ID	Name	Type	Brand	Price	Action
1181	AndaSeat Fnatic Edition	Chair	AndaSeat	399.99	Edit Delete
1197	AndaSeat NAVI Edition	Chair	AndaSeat	349.99	Edit Delete
1191	AndaSeat NAVI Edition Yellow	Chair	AndaSeat	349.99	Edit Delete
1187	Andaseat Phantom 3	Chair	Andaseat	349.99	Edit Delete
1188	AutoFull M6 Ultra	Chair	AutoFull	299.99	Edit Delete
1172	Backforce One	Chair	Backforce	549.99	Edit Delete
1168	Blacklyte Athena Pro	Chair	Blacklyte	299.99	Edit Delete
1193	Blacklyte Kraken	Chair	Blacklyte	299.99	Edit Delete
1165	Blacklyte Kraken Pro	Chair	Blacklyte	299.99	Edit Delete
1189	DEVO Gaming Chair	Chair	DEVO	249.99	Edit Delete
1183	DXRacer Formula	Chair	DXRacer	249.99	Edit Delete
1182	DXRacer King	Chair	DXRacer	349.99	Edit Delete
1173	DXRacer Prince	Chair	DXRacer	299.99	Edit Delete
1171	Herman Miller Aeron	Chair	Herman	1395.99	Edit Delete

รูปภาพที่ 67 แสดงหน้าจอจัดการ Gaming Gear (ค.14)