



ระบบติดตามสถิติผู้เล่นพีบีจี

PUBG Status System

นาย กฤษณะ ดั่งงด้อย 664230007

66/45

โครงการนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 7204903

โครงการด้านเทคโนโลยีสารสนเทศ 2

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

ภาคเรียนที่ 1 ปีการศึกษา 2568

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการแข่งขันเกมพีบีจี ได้รับความนิยมอย่างแพร่หลายทั้งในระดับบุคคลและทัวร์นาเมนต์ อย่างไรก็ตาม การติดตามสถิติผู้เล่นยังขาดระบบที่ รวบรวม ประมวลผล แสดงผล ได้ครบถ้วนและใช้งานสะดวก โดยเฉพาะในส่วนของการดึงข้อมูลมาเปรียบเทียบผลงานรายแมตช์ ต้องรวบรวมข้อมูลจากหลายแหล่งและคำนวณเอง ซึ่งทำให้ล่าช้าและเสี่ยงต่อความผิดพลาด อีกทั้งการเข้าถึงข้อมูลของผู้เล่นระดับโปรเพื่อเทียบกับผู้เล่นทั่วไปก็ทำได้ยาก เนื่องจากยังไม่มีเครื่องมือที่ดึงและแสดงผลเชิงเปรียบเทียบได้โดยตรง ส่งผลให้ กระบวนการคัดตัวผู้เล่นใหม่ ของทีมและสโมสรยังขาดข้อมูลเชิงประจักษ์ที่เพียงพอ และการวิเคราะห์ผลสถิติก็ยังทำได้ไม่สะดวกในทีเดียว

จากปัญหาดังกล่าว โครงการนี้จึงพัฒนาเว็บแอปพลิเคชันสำหรับติดตามสถิติผู้เล่นพีบีจี ที่สามารถรวมข้อมูลจากหลายแหล่งและประมวลผลอัตโนมัติ เพื่อลดความซับซ้อนในการดูสถิติ และเปิดโอกาสให้ผู้เกี่ยวข้องทุกฝ่าย เช่น ผู้เล่น, แมวมอง, ผู้จัดการทีม, และผู้เกี่ยวข้องกับการแข่งขัน สามารถเข้าถึงข้อมูลเพื่อ วิเคราะห์และเปรียบเทียบ ผลงานของผู้เล่นได้สะดวกยิ่งขึ้น ทั้งยังสนับสนุนการ ค้นหาและคัดตัวผู้เล่นใหม่ของแมวมอง ด้วยการดูกราฟสถิติเพื่อใช้ประกอบการตัดสินใจอย่างมีประสิทธิภาพ

1.2 แนวคิดในการแก้ไขปัญหา

แนวคิดของระบบคือทำให้การติดตามและเปรียบเทียบสถิติผู้เล่น รวดเร็ว ถูกต้อง และพร้อมใช้คัตตัว โดยพัฒนาเว็บแอปที่ ดึงข้อมูลอัตโนมัติจากหลายแหล่ง เช่น เอฟพีไออย่างเป็นทางการของเกม และเว็บไซต์สรุปผลการแข่งขัน แล้วทำการแปลงและจัดรูปข้อมูลให้เป็นมาตรฐานเดียวกัน ก่อนคำนวณตัวชี้วัดที่จำเป็นต่อการประเมิน จากนั้นจึงสรุปและเก็บแคชข้อมูล ไฟล์เจสันเพื่อลดการเรียกซ้ำและเพิ่มความรวดเร็วในการแสดงผล

ผลลัพธ์ถูกนำเสนอผ่านกราฟและตารางที่อ่านง่าย ผู้ใช้สามารถ กดดูสถิติผู้เล่นและเปรียบเทียบผู้เล่น ได้ ช่วยให้แมวมองและผู้จัดการทีมใช้ ข้อมูลตัวเลขที่ตรวจสอบได้

ประกอบการตัดสินใจคัตตัว ลดงานรวบรวมข้อมูลด้วยมือ และลดความคลาดเคลื่อนของข้อมูล เมื่ออัปเดตจากแหล่งเดิมอย่างต่อเนื่อง

1.3 วัตถุประสงค์ของระบบ

โครงการนี้มีวัตถุประสงค์หลักในการพัฒนาระบบ เว็บแอปพลิเคชันสำหรับติดตามสถิติผู้เล่นจากรายการแข่งขันพบจี ที่สามารถดึงข้อมูลและแสดงสถิติผู้เล่นได้โดยอัตโนมัติ โดยไม่ต้องกรอกข้อมูลด้วยมือ เพื่อลดความซับซ้อนและเพิ่มความสะดวกในการติดตามผลการแข่งขัน

1.3.1 เพื่อให้แมวมองหรือผู้จัดการทีมสามารถติดตามสถิติผู้เล่นจากรายการแข่งขันพบจีได้

1.3.2 เพื่อให้ผู้เล่นสามารถติดตามคุณสมบัติผู้เล่นและประวัติการเล่นของตนเองได้อย่างรวดเร็วและเข้าใจง่าย

1.3.3 เพื่อให้ผู้ชมสามารถดูสถิติผู้เล่นจากรายการแข่งขันพบจีและเปรียบเทียบสถิติผู้เล่นได้

1.4 ขอบเขตการศึกษา

การศึกษาในโครงการนี้จะเน้นไปที่การพัฒนาระบบ จัดการติดตามสถิติผู้เล่นจากรายการแข่งขันพบจี โดยมีขอบเขตการศึกษาและการพัฒนาในหลายด้านที่เกี่ยวข้อง ดังนี้

1.4.1 ขอบเขตของระบบ

1.4.1.1 แมวมองหรือผู้จัดการทีม

- ก) ติดตามสถิติผู้เล่นจากการแข่งขันพบจี เพื่อใช้ประเมินศักยภาพเบื้องต้น
- ข) เปรียบเทียบสถิติกับผู้เล่นโปร ผ่านกราฟและตารางสรุป
- ค) วิเคราะห์แนวโน้มจากค่าเฉลี่ยต่อเกม เพื่อช่วยการตัดสินใจเบื้องต้น

1.4.1.2 ผู้เล่น

- ก) ติดตามประวัติการเล่นของตนเอง
- ข) คุณสมบัติที่สำคัญของตนเอง
- ค) เปรียบเทียบสถิติระหว่างผู้เล่นในทัวร์นาเมนต์เดียวกัน

1.4.1.3 ผู้ชม

- ก) ติดตามสถิติผู้เล่นจากรายการแข่งขันพบจี
- ข) เปรียบเทียบสถิติผู้เล่นในทัวร์นาเมนต์
- ค) ดูกราฟการเปรียบเทียบผู้เล่น

1.4.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

1.4.2.1 คอมพิวเตอร์ส่วนบุคคล

1.4.3 ซอฟต์แวร์ที่ใช้ในการพัฒนา

1.4.3.1 ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 11 (Microsoft Windows 11)

1.4.3.2 ไพธอน (Python) เวอร์ชัน 3.8 สำหรับการพัฒนาฟลัสเอพีไอ และการเชื่อมต่อกับพับจีเอพีไอ

1.4.3.3 วิวีสตูดิโอโค้ด (Visual Studio Code) หรือไอดีอีอื่นๆ สำหรับการเขียนโค้ดและทดสอบ

1.4.3.4 เพลย์ไรท์ (Playwright) สำหรับการดึงข้อมูลจากทไววีจีจี ผ่านการเรนเดอร์หน้าเว็บ (เฮดเลส โครเมียม)

1.4.3.5 บิวทิฟูลซูป (BeautifulSoup) สำหรับการพาร์สข้อมูล เอชทีเอ็มแอล (HTML) จากทไววีจีจี

1.4.3.6 ฟลัส (Flask) สำหรับการสร้างเว็บแอปพลิเคชัน

1.4.3.7 กิต (Git) สำหรับการควบคุมเวอร์ชันของโค้ด

1.4.4 บริการแบบคลาวด์ที่ใช้ในการพัฒนา

1.4.4.1 เร็นเดอร์ (Render) บริการแบบคลาวด์สำหรับใช้ในการเป็นเว็บเซิร์ฟเวอร์

1.4.4.3 กิตฮับ (GitHub) บริการแบบคลาวด์สำหรับใช้ในการจัดเก็บโค้ดของโปรเจกต์

1.6 ประโยชน์ที่ได้คาดว่าจะได้รับ

1.6.1 แมวมองหรือผู้จัดการทีม สามารถดูสถิติและแนวโน้มผลงานผู้เล่นได้สะดวก ช่วยคัดกรองเบื้องต้นได้

1.6.2 ผู้เล่นสามารถดูสถิติของตนเอง และ ประวัติการเล่นของตนเองได้

1.6.3 ผู้ชมสามารถดูสถิติผู้เล่นจากรายการแข่งขันพับจีและเปรียบเทียบสถิติผู้เล่นได้

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในการพัฒนาระบบติดตามสถิติผู้เล่นพับจี ผู้จัดทำได้ศึกษาข้อมูล แนวคิด และเทคโนโลยีที่เกี่ยวข้องเพื่อใช้เป็นพื้นฐานในการออกแบบและพัฒนาระบบ โดยสามารถสรุป ได้ดังนี้

2.1 ระบบงานเดิม

2.1 ระบบงานเดิม

โดยทั่วไปการติดตามสถิติผู้เล่น พับจี ในปัจจุบันอาศัยทั้งข้อมูลจาก พับจี ออฟฟิเชียล เอพีไอ และเว็บไซต์ภายนอกอย่าง ทไวร์จีจี ซึ่งแม้จะให้ข้อมูลที่น่าเชื่อถือและสะท้อนบริบทการแข่งขันที่เข้มข้นในระบบอีสปอร์ตของพับจีก็ตาม ระบบการแข่งขัน พับจี มีการจัดทัวร์นาเมนต์ระดับโลกอย่างต่อเนื่อง เช่น พับจี โกลบอล ซีรีส์ ซึ่งมีเงินรางวัลรวมสูง 300,000 ดอลลาร์สหรัฐ ต่อรายการ และมีการใช้ระบบเก็บแต้ม พีจีเอสพอย เพื่อให้ทีมมีสิทธิ์เข้าร่วมการแข่งขันชิงแชมป์โลกพับจีโกลบอลแชมเปียนชิป ในช่วงปลายปี โดยการให้คะแนนในการแข่งขันจะคิดจากการนับจำนวนการสังหาร คิลพอย 1 แต้มต่อ 1 การฆ่าและ แต้มอันดับตอนจบเกม อันดับ1ได้ 10 แต้ม อันดับ2ได้ 6แต้ม อันดับ3ได้ 5แต้ม อันดับ4ได้ 4แต้ม อันดับ5ได้ 3แต้ม อันดับ6ได้2แต้ม อันดับ7ถึง8ได้ 1แต้ม อันดับ9ถึง16ได้ 0แต้ม การแข่งขันมีเงินรางวัลสูงมาก ทำให้ผู้ใช้โดยเฉพาะ แมวมอง หรือ ผู้จัดการทีม ต้องการผู้เล่นที่เก่งๆ เพื่อมาแข่ง

2.2 ระบบงานอื่นที่เกี่ยวข้อง

การติดตามและแลกเปลี่ยนข้อมูลสถิติผู้เล่นผ่านเครือข่ายอินเทอร์เน็ตมีพัฒนาการต่อเนื่อง โดยเฉพาะในบริบทของเกมอีสปอร์ตอย่างพับจี ปัจจุบันแหล่งข้อมูลสำคัญที่นิยมใช้งานคือ พับจีออฟฟิเชียลเอพีไอ ซึ่งให้ข้อมูลประวัติแมตช์และสถิติพื้นฐานในรูปแบบเจสัน ที่เชื่อมต่อได้มาตรฐาน และเว็บไซต์ภายนอกอย่าง ทไวร์จีจี ที่จัดทำสรุปสถิติการแข่งขันและรายชื่อผู้เล่นในทัวร์นาเมนต์ต่างๆ ระบบงานดังกล่าวช่วยให้ผู้ใช้เข้าถึงข้อมูลได้สะดวก มีความน่าเชื่อถือ และรองรับการอ้างอิงข้อมูลเชิงการแข่งขัน อย่างไรก็ตาม เมื่อพิจารณาในมุมการใช้งานจริงสำหรับงานแมวมอง ยังพบข้อจำกัดหลายประการ ได้แก่ การต้องดึงข้อมูลจากหลายแหล่งแล้วแปลงข้อมูลให้เป็นมาตรฐานก่อนคำนวณ การคัดกรองตามเงื่อนไขเฉพาะ ที่ผู้ใช้มักต้อง

ดำเนินการด้วยตนเอง ตลอดจนการขาดฟังก์ชันสนับสนุนการทำงานของแมวมองโดยตรง เช่น การทำรายการสั้นๆ การบันทึกข้อสังเกตประกอบตัวเลข และการเปรียบเทียบกับฐานอ้างอิงผู้เล่น อย่างยืดหยุ่น ผู้ใช้งานจำนวนมากจึงยังต้องอาศัยสเปรดชีตหรือการจดบันทึกเพิ่มเติม ซึ่งมีความยุ่งยาก ใช้เวลา และเสี่ยงต่อความคลาดเคลื่อน ส่งผลให้ความต่อเนื่องและความโปร่งใสของกระบวนการคัดเลือกผู้เล่นยังไม่เพียงพอ ระบบติดตามสถิติผู้เล่นพบจึงมีความจำเป็น เพื่อยกระดับ ความพร้อมใช้ของข้อมูลสำหรับการตัดสินใจด้านสรรหานักกีฬาอีสปอร์ตอย่างมีประสิทธิภาพ

2.3 องค์ความรู้ที่เกี่ยวข้อง

ในการพัฒนาโครงการนี้ ผู้จัดทำเลือกใช้ชุดเทคโนโลยีที่เหมาะสมทั้งฝั่งระบบปฏิบัติการ ภาษาพัฒนา เฟรมเวิร์กเว็บ ไลบรารีช่วยคำนวณและแสดงผล และเครื่องมือเชื่อมต่อบริการภายนอก ดังนี้

2.3.1 ไมโครซอฟท์ วินโดวส์ 11 (Microsoft Windows 11)

โครงการนี้พัฒนาบนระบบปฏิบัติการ ไมโครซอฟท์ วินโดวส์ 11 เนื่องจากมีความเสถียรและรองรับซอฟต์แวร์เครื่องมือพัฒนาที่หลากหลาย อีกทั้งมีการจัดการไฟล์และเครือข่ายที่สะดวก รวมถึงคุณสมบัติด้านความปลอดภัยและการอัปเดตสม่ำเสมอเหมาะสำหรับการพัฒนา ทดสอบ และจำลองสภาพแวดล้อมใช้งานของเว็บแอปพลิเคชันภายในเครื่องเดียวกันอย่างมีประสิทธิภาพ



ภาพที่ 2.1 ไมโครซอฟท์ วินโดวส์ 11

ที่มา <https://news.microsoft.com/th-th/2021/06/25/windows11-th/>

2.3.2 ไพธอน (Python)

ภาษาที่ใช้พัฒนาแกนของระบบในโครงการนี้ ทั้งงานให้บริการเอพีไอและงานประมวลผลข้อมูล จุดเด่นคือโค้ดอ่านง่าย มีไลบรารีพร้อมใช้จำนวนมาก และเชื่อมต่อบริการภายนอกได้สะดวก ในโครงการนี้ใช้ไพธอนเวอร์ชัน 3.8 สำหรับพัฒนาเอพีไอด้วย ฟลask, ติดต่อบัซเนสเอพีไอ, ดึงและเรนเดอร์หน้าเว็บด้วย เฟลย์ไรท์, พาร์สข้อมูลเอชทีเอ็มแอลด้วย บิวทิฟูลซูป และจัดรูปข้อมูลเป็น เจสัน รวมทั้งมีการคำนวณค่าสถิติและจัดเก็บผลแบบ แคชไฟล์เจสัน เพื่อลดการเรียกข้อมูลซ้ำ เพิ่มความเร็ว และรองรับการพัฒนาและทดสอบบน วินโดวส์ 11 ได้อย่างมีประสิทธิภาพ



ภาพที่ 2.2 ไพธอน (Python)

ที่มา <https://igate-eg.com/what-is-python/>

2.3.3 วิซวล สตูดิโอ โค้ด (Visual Studio Code)

ใช้ วิซวล สตูดิโอ โค้ด เป็นเครื่องมือพัฒนาหลัก เนื่องจากมีขนาดเล็ก ตั้งค่าไม่ซับซ้อน และรองรับการพัฒนาไพธอนและแฟล็กได้ดีผ่านส่วนขยายที่ติดตั้งเพิ่มเติมได้สะดวก ภายในโปรแกรมมีเทอร์มินัลในตัวเครื่องมือดีบั๊ก ระบบค้นหาไฟล์และจัดการเวิร์กสเปซ รวมถึงการทำงานร่วมกับกิตและกิตฮับ ทำให้การเขียนโค้ด ทดสอบ และแก้ไขโปรแกรมบน วินโดวส์ 11 เป็นไปอย่างรวดเร็วและมีประสิทธิภาพ



ภาพที่ 2.3 วิซวล สตูดิโอ โค้ด (Visual Studio Code)

ที่มา <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/visual-studio-code>

2.3.4 เพลย์ไรท์ (Playwright)

เครื่องมืออัตโนมัติสำหรับเปิดและโต้ตอบกับหน้าเว็บ เพื่อดึงข้อมูลจากเว็บไซต์ที่ต้องเรนเดอร์ด้วยจาวาสคริปต์ โดยสามารถรันแบบ เฮดเลส ผ่านเบราว์เซอร์โครเมียม, ไฟร์ฟอกซ์, เว็บคิต จัดการการรอโหลดองค์ประกอบ, การเลื่อนหน้า, การแบ่งหน้า และการดักจับข้อผิดพลาดของเครือข่ายได้อย่างแม่นยำ ช่วยให้การเก็บข้อมูลมีความเสถียร ลดปัญหาข้อมูลขาดหาย และทำงานร่วมกับกระบวนการพาร์ส เอ็ดทีเอ็มแอล ต่อด้วยบิวกิฟูลซูปได้อย่างมีประสิทธิภาพ



ภาพที่ 2.4 เพลย์ไรท์ (Playwright)

ที่มา <https://www.linkedin.com/pulse/pros-cons-playwright>

2.3.5 บีวทิฟูลซูป (BeautifulSoup)

ไลบรารีสำหรับพาร์สหรือสแครปป์ิงหน้าเว็บและไฟล์เอชทีเอ็มแอล เพื่อดึงข้อมูลออกมาในรูปแบบที่นำไปใช้ได้ เช่น ตารางรายชื่อผู้เล่นหรือค่าสถิติต่าง ๆ โดยสามารถค้นหาแท็กและคุณลักษณะด้วยตัวเลือกที่ยืดหยุ่น (เช่น ชื่อแท็ก คลาส หรือตัวเลือกเชิงโครงสร้าง) จัดการเอชทีเอ็มแอลที่ไม่สมบูรณ์ได้ดี รองรับอักขระยูนิโคด และทำงานรวดเร็วเมื่อติดตั้งตัวเรนเดอร์แบบแอลเอ็กซ์เอ็มแอลร่วมกัน การใช้งานในโครงงานนี้คืออ่านองค์ประกอบของตาราง บนหน้าแข่งขัน ดึงข้อความออกมาทำความสะอาด แปลงเป็นชนิดข้อมูลที่ต้องการ แล้วจัดรูปเป็นข้อมูลเชิงโครงสร้างเพื่อใช้คำนวณและแสดงผลสถิติบนหน้าเว็บต่อไป



ภาพที่ 2.5 บีวทิฟูลซูป (BeautifulSoup)

ที่มา <https://www.matatika.com/data-details/tap-beautifulsoup/>

2.3.6 ฟลาส (Flask)

เฟรมเวิร์กเว็บสำหรับภาษาไพธอนที่มีความเรียบง่ายและยืดหยุ่น เหมาะสำหรับการพัฒนาเอพีไอและเว็บแอปที่ต้องการประมวลผลข้อมูลแล้วส่งผลลัพธ์กลับไปยังหน้าเว็บ จุดเด่นคือมีระบบกำหนดเส้นทาง จัดการคำขอ คำตอบ รองรับเทมเพลต จัดส่งข้อมูลแบบเจสัน ได้สะดวก และสามารถแยกส่วนการทำงานเป็นหมวดหมู่ได้ นอกจากนี้ยังเชื่อมต่อกับไลบรารีอื่นได้ง่าย เช่น เครื่องมือดึงข้อมูลหน้าเว็บและเครื่องมือพาร์สข้อมูล เพื่อนำข้อมูลมาคำนวณก่อนแสดงผล ในโครงงานนี้ ฟลาสถูกใช้เป็นแกนกลางของบริการเอพีไอ สำหรับดึงข้อมูลสถิติผู้เล่น ประมวลผลค่าเฉลี่ย คัดกรองตามเงื่อนไข และส่งข้อมูลไปยังตารางและกราฟบนหน้าเว็บ รวมถึงสามารถปรับใช้บนบริการคลาวด์ได้อย่างสะดวกและรวดเร็ว



ภาพที่ 2.6 ฟลาส (Flask)

ที่มา <https://www.bairesdev.com/blog/what-is-flask/>

2.3.7 กิต (Git)

ระบบควบคุมเวอร์ชันของซอร์สโค้ด ช่วยบันทึกการเปลี่ยนแปลงเป็นลำดับเวลา สามารถย้อนกลับ เทียบความต่าง และทำงานร่วมกันหลายคนได้อย่างเป็นระเบียบ ในโครงการนี้ใช้กิตสำหรับแยกสาขาพัฒนา สร้างคอมมิต และรวมโค้ด ก่อนส่งขึ้นพื้นที่เก็บโค้ดบนบริการคลาวด์ เช่น กิตฮับ เพื่อสำรองโค้ดติดตามงานและตรวจสอบประวัติการแก้ไขได้สะดวก ลดความเสี่ยงโค้ดสูญหายและช่วยให้การพัฒนามีมาตรฐานมากขึ้น



ภาพที่ 2.7 กิต (Git)

ที่มา <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/git-full-logo>

2.3.8 เร็นเดอร์ (Render)

บริการคลาวด์สำหรับนำเว็บแอปและเอพีไอขึ้นใช้งานจริง ช่วยจัดการเซิร์ฟเวอร์ การปรับขนาดทรัพยากรการดีพลอยจากคลังโค้ด เช่น กิตฮับ และการตั้งค่าโดเมน

ใบรับรองความปลอดภัยได้อัตโนมัติ ในโครงการนี้ใช้เรนเดอร์เพื่อเผยแพร่เว็บแอปฟลัส ทำให้ผู้ใช้เข้าถึงระบบติดตามสถิติผู้เล่นผ่านอินเทอร์เน็ตได้สะดวก มีความเสถียร และอัปเดตเวอร์ชันได้รวดเร็วเมื่อมีการแก้ไขโค้ด

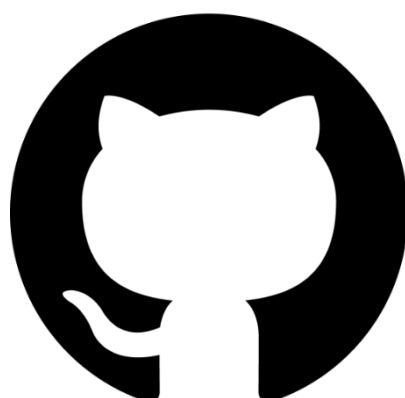


ภาพที่ 2.8 เร็นเดอร์ (Render)

ที่มา <https://seeklogo.com/vector-logo/532232/render>

2.3.9 กิตฮับ (GitHub)

บริการบนคลาวด์สำหรับเก็บซอร์สโค้ดและทำงานร่วมกันบนกิต ช่วยให้ผู้พัฒนาสามารถสำรองโค้ด แยกสาขาพัฒนา (branch) สร้างคอมมิต (commit) เปิดคำขอรวมโค้ด (pull request) และตรวจสอบประวัติการเปลี่ยนแปลงได้อย่างเป็นระบบ ในโครงการนี้ใช้กิตฮับเป็นที่เก็บโค้ดหลัก เชื่อมต่อกับกระบวนการดีพลอยบนเรนเดอร์ อำนวยความสะดวกในการรีวิวโค้ด ติดตามงาน และอัปเดตเวอร์ชันของระบบติดตามสถิติผู้เล่นให้สอดคล้องกันระหว่างสภาพแวดล้อมพัฒนาและใช้งานจริง



ภาพที่ 2.9 กิตฮับ (GitHub)

ที่มา <https://slack.com/marketplace/A01BP7R4KNY-github>

2.3.10 เอชทีเอ็มแอล (HTML)

เอชทีเอ็มแอลเป็นภาษาที่ใช้กำหนดโครงสร้างหน้าเว็บ จัดวางหัวเรื่อง เนื้อหา ตาราง ปุ่ม และส่วนประกอบต่าง ๆ ของระบบ โดยใช้แท็กที่สื่อความหมายเพื่อให้ทั้งผู้ใช้ และเครื่องมือเข้าใจตรงกัน ทำงานร่วมกับซีเอสเอสในการจัดรูปแบบและรองรับหลายขนาดหน้าจอ และทำงานร่วมกับจาวาสคริปต์เพื่อโต้ตอบกับผู้ใช้และแสดงข้อมูลเชิงภาพ เช่น กราฟ และสรุปสถิติ ในโครงงานนี้ เอชทีเอ็มแอลทำหน้าที่เป็นโครงหลักของหน้าแสดงรายชื่อผู้เล่น หน้าแสดงรายละเอียดสถิติ และหน้าสำหรับเปรียบเทียบข้อมูล ให้ข้อมูลอ่านง่าย เป็นระเบียบ และเข้าถึงได้สะดวก



ภาพที่ 2.10 เอชทีเอ็มแอล (HTML)

ที่มา <https://wproom.com/html/>

2.3.11 ซีเอสเอส (CSS)

ซีเอสเอสเป็นภาษาสำหรับกำหนดรูปแบบการแสดงผลของหน้าเว็บ ช่วยจัดการสี ฟอนต์ ระยะห่าง ขนาดเลย์เอาต์ และการตอบสนองต่อหลายขนาดหน้าจอให้ข้อมูลอ่านง่ายและสม่ำเสมอทั้งระบบ ในโครงงานนี้ ซีเอสเอสใช้เพื่อจัดวางตารางสถิติ การ์ดสรุป กริดคอมโพเนนต์ และส่วนหัว/ส่วนท้าย รวมถึงกำหนดธีมสีและลักษณะตัวอักษรให้สอดคล้องกันทั่วทั้งเว็บ นอกจากนี้ยังใช้สื่อสถานะ (เช่น สีเน้นเมื่อโฮเวอร์หรือเลือกผู้เล่น) และปรับรูปแบบให้เหมาะสมกับอุปกรณ์มือถือ แท็บเล็ต และจอเดสก์ท็อป เพื่อประสบการณ์ใช้งานที่ราบรื่น



ภาพที่ 2.11 ซีเอสเอส (CSS)

ที่มา <https://www.dwthai.com/dwarticle/>

2.3.12 จาวาสคริปต์ (JavaScript)

จาวาสคริปต์เป็นภาษาที่ใช้เพิ่มความสามารถเชิงโต้ตอบให้หน้าเว็บ ควบคุมพฤติกรรมขององค์ประกอบ รับ ส่งข้อมูลกับเซิร์ฟเวอร์ และอัปเดตหน้าจอแบบไม่ต้องโหลดใหม่ ในโครงการนี้ จาวาสคริปต์ใช้สำหรับ เรียกเอพีไอ ประมวลผลฝั่งไคลเอนต์เล็กน้อย ควบคุมส่วนติดต่อผู้ใช้ แสดงผลเชิงภาพ เพิ่มประสิทธิภาพการ ผลลัพธ์คือหน้าเว็บที่ตอบสนองไว ผู้ใช้ค้นหา ดูสถิติ เปรียบเทียบผู้เล่นได้ทันที โดยไม่ต้องโหลดหน้าใหม่ทั้งหน้า ทำให้ประสบการณ์ใช้งานราบรื่นและเหมาะกับการคัดตัวผู้เล่นในบริบทจริง



ภาพที่ 2.12 จาวาสคริปต์ (JavaScript)

ที่มา https://logos-world.net/javascript-logo/#google_vignette

บทที่ 3

วิธีการดำเนินงาน

3.1 การศึกษาเบื้องต้น

3.1.1 ระบบงานเดิม

จากการศึกษาวิธีติดตามสถิติผู้เล่นพับบีที่ใช้งานอยู่ในปัจจุบัน พบว่าผู้ใช้ เช่น แมวมอง ผู้จัดการทีม และผู้เล่น มักประสบปัญหาในการ รวบรวม ประมวลผล แสดงผล ข้อมูล ให้ครบถ้วนในทีเดียว ข้อมูลกระจายอยู่หลายแหล่ง ได้แก่ เอฟีไออย่างเป็นทางการของเกม และ เว็บไซต์สรุปผลการแข่งขัน เช่น ทไววีจีจี ที่มีข้อมูลทัวร์นาเมนต์เฉพาะกิจ การใช้งานจริงจึงต้อง เปิดหลายหน้าเว็บบนคอมพิวเตอร์ส่วนบุคคล คัดลอกข้อมูลทีละ แล้วคำนวณค่าเฉลี่ยด้วย ตนเอง ส่งผลให้กระบวนการวิเคราะห์ล่าช้า เสี่ยงต่อความคลาดเคลื่อน และยากต่อการ ตรวจสอบในภายหลัง

นอกจากนี้ ระบบงานเดิมยังมีข้อจำกัดเชิงเทคนิค เช่น ข้อกำหนดอัตราการเรียกใช้งานของเอฟีไอ , การเรนเดอร์หน้าแบบไดนามิกที่ต้องใช้เครื่องมือเฉพาะในการดึงข้อมูล , และความเปลี่ยนแปลงของโครงสร้างหน้าเว็บที่ทำให้ต้องบำรุงรักษาสคริปต์พาร์สอย่างต่อเนื่อง อีกทั้งแพลตฟอร์มสถิติทั่วไปยังไม่ได้ออกแบบให้รองรับการทำงานของงานแมวมองโดยตรง เช่น การเปรียบเทียบผู้เล่นตามเงื่อนไขเดียวกัน การบันทึกข้อสังเกตประกอบตัวเลข หรือการจัดทำ รายชื่อผู้เล่น เพื่อส่งต่อให้ทีมตัดสินใจ

จากปัญหาข้างต้น สามารถสรุปประเด็นสำคัญได้ว่า ระบบงานเดิมยังไม่ตอบโจทย์ ศูนย์รวมข้อมูลเพื่อคัดตัว ที่ผู้ใช้ต้องการกล่าวคือยังขาดเครื่องมือที่สามารถดึงข้อมูลอัตโนมัติ จากหลายแหล่ง แปลงข้อมูลให้เป็นมาตรฐาน คำนวณตัวชี้วัดสำคัญ แสดงผลแบบกราฟและ ตารางเพื่อเปรียบเทียบได้ทันทีในหน้าเดียว จึงเกิดความจำเป็นในการพัฒนาระบบใหม่ที่บูรณาการขั้นตอนทั้งหมดเข้าด้วยกัน ลดงานมือ ลดข้อผิดพลาด และสนับสนุนการคัดตัวผู้เล่นอย่างมี ข้อมูลตัวเลขที่ตรวจสอบได้เป็นหลักฐาน

3.1.2 ระบบงานใหม่

จากข้อจำกัดของระบบงานเดิม จึงออกแบบและพัฒนา ระบบงานใหม่สำหรับ ติดตามสถิติผู้เล่นพับบี โดยรวมขั้นตอน ดึงข้อมูล ทำมาตรฐาน คำนวณ แสดงผล ไว้ในทีเดียว ระบบ

จะเชื่อมต่อกับแหล่งข้อมูลที่ใช้จริง เช่น เอฟพีไออย่างเป็นทางการของเกม และข้อมูลสรุปจากเว็บไซต์ทัวร์นาเมนต์เฉพาะส่วนที่จำเป็น แล้วทำการแปลงรูปแบบข้อมูลเวลา ระยะทาง และค่าตัวเลขให้เป็นมาตรฐานเดียวกันก่อนประมวลผล ลดความผิดพลาดที่เกิดจากการรวบรวมและคำนวณด้วยมือ เมื่อได้ข้อมูลระบบจะคำนวณและแปลงข้อมูล จากนั้นจัดเก็บผลลัพธ์เป็น แคชไฟล์เจสัน เพื่อลดการเรียกข้อมูลซ้ำ เพิ่มความเร็ว และคงความสม่ำเสมอของข้อมูลเมื่อเรียกใช้งานครั้งต่อไป

ผลลัพธ์ถูกนำเสนอผ่านหน้าเว็บที่อ่านง่าย ผู้ใช้สามารถดูรายชื่อผู้เล่นจากรายการแข่งขัน กดดูรายละเอียดรายคนเพื่อเห็นกราฟและตารางสรุป หรือเลือกเปรียบเทียบผู้เล่นได้ภายในหน้าเดียว ช่วยให้ แมวมองและผู้จัดการทีม ใช้ ข้อมูลตัวเลขที่ตรวจสอบได้ ประกอบการตัดสินใจได้ทันที ลดภาระการรวบรวมข้อมูลจากหลายแหล่ง ลดความคลาดเคลื่อน และทำให้กระบวนการตัดสินใจมีประสิทธิภาพ และพร้อมใช้งานจริงในสภาพแวดล้อมการแข่งขัน

3.2 การกำหนดความต้องการของระบบ

การพัฒนาระบบติดตามสถิติผู้เล่นจำเป็นต้องศึกษาความต้องการของเจ้าของระบบและผู้ใช้ระบบ เพื่อให้ได้ข้อกำหนดความต้องการ เพื่อใช้ในการพัฒนาระบบ

3.2.1 ขอบเขตของระบบ

ระบบติดตามสถิติผู้เล่นที่มีขอบเขตการทำงานดังนี้

3.2.1.1 ขอบเขตของระบบที่สามารถทำได้

- ก) ดูประวัติการเล่นได้โดยการใส่ชื่อในเกม
- ข) เปรียบเทียบประวัติการเล่นของตัวเองกับโปรเพลเยอร์
- ค) ดูสถิติของผู้เล่นจากรายการ ฟุตบอล

3.2.1.1 ขอบเขตของระบบที่ไม่สามารถทำได้

- ก) ลองรับการใส่ชื่อแค่แพลตฟอร์มเดียว คือ สตีม แพลตฟอร์มอื่นๆไม่สามารถดูประวัติการเล่นได้โดยการใส่ชื่อในเกมได้
- ข) เปรียบเทียบประวัติการเล่นของตัวเองกับโปรเพลเยอร์ เปรียบเทียบได้แค่กับโปรเพลเยอร์ที่กำหนดไว้เท่านั้น
- ค) ไม่สามารถดูสถิติของผู้เล่นจากรายการอื่นได้

3.2.2 ฮาร์ดแวร์ที่ใช้กับระบบงาน

ฮาร์ดแวร์สำหรับพัฒนาและใช้งานระบบแบ่งเป็น 3 ส่วน เพื่อให้การดึงข้อมูลประมวลผล และเปิดใช้งานผ่านเว็บเป็นไปอย่างต่อเนื่องและเสถียร ดังนี้

3.2.2.1 เครื่องพัฒนาภายใน

ใช้คอมพิวเตอร์ระบบปฏิบัติการ ไมโครซอฟท์ วินโดวส์ 11 สำหรับเขียนโค้ด ทดสอบการดึงข้อมูลด้วย เฟลย์ไรท์ และรันเซิร์ฟเวอร์ระหว่างพัฒนา

3.2.2.2 เครื่องแม่ข่ายบนระบบคลาวด์

เปิดให้บริการเว็บแอปและเอพีไอผ่านผู้ให้บริการคลาวด์ เร็นเดอร์ โดยรันเป็นคอนเทนเนอร์และปรับขนาดทรัพยากรได้ตามการใช้งาน พร้อมพื้นที่ดิสก์เพียงพอสำหรับไฟล์แคชแบบ เจสัน ของระบบ

3.2.2.3 อุปกรณ์ของผู้ใช้ปลายทาง

ผู้ใช้เข้าถึงระบบผ่าน เว็บเบราว์เซอร์ บนคอมพิวเตอร์หรืออุปกรณ์พกพา หน้าเว็บออกแบบให้ตอบสนองหลายขนาดหน้าจอ เพื่อใช้งานได้สะดวกทั้งในคอมพิวเตอร์ตั้งโต๊ะ โน้ตบุ๊ก แท็บเล็ต และโทรศัพท์มือถือ

3.2.3 ซอฟต์แวร์ที่ใช้กับระบบงาน

ซอฟต์แวร์ของระบบแบ่งออกเป็นหลายส่วนที่ทำงานร่วมกัน ดังนี้

3.2.3.1 ส่วนกลาง คือรันไทม์ โหนดเจเอส ให้บริการเว็บและเอพีไอด้วยเฟรมเวิร์กเอ็กซ์เพรสทำหน้าที่รับคำขอจากผู้ใช้งานเว็บเบราว์เซอร์ ประมวลผลข้อมูลสถิติ และส่งผลลัพธ์กลับเป็นหน้าเว็บและข้อมูล เจสัน เพื่อให้หน้าจออัปเดตได้โดยไม่ต้องโหลดทั้งหน้าเพื่อ ดึงข้อมูลจากแหล่งภายนอก ระบบเรียกใช้ เอพีไออย่างเป็นทางการของพับจี ผ่านไลบรารีสำหรับร้องขอเครือข่าย เช่น แอ็กซอสและโหนดเฟตซ์ และใช้เฟลย์ไรท์ โหมดเฮดเลสโครเมียมสำหรับเรนเดอร์หน้าเว็บของแหล่งข้อมูลทัวร์นาเมนต์ที่สร้างเนื้อหาแบบไดนามิก จากนั้นพาร์สส่วนตารางและสถิติด้วยไลบรารี เซียร์โอ แล้วแปลงให้อยู่ในรูปแบบมาตรฐานก่อนนำไปคำนวณตัวชี้วัดสำคัญ เพื่อลดการเรียกข้อมูลซ้ำ และเลี่ยงข้อจำกัดการเรียกใช้ของแหล่งข้อมูลระบบจะจัดเก็บผลดึงข้อมูลและผลคำนวณเป็น ไฟล์แคชแบบเจสัน พร้อมกำหนดอายุข้อมูล (TTL) เมื่อครบกำหนดจะดึงข้อมูลใหม่โดยอัตโนมัติ ค่าตั้งค่าที่แปรผันตามสภาพแวดล้อม (เช่น คีย์เอพีไอและตำแหน่งฐานข้อมูล) จัดการผ่านไฟล์ตัวแปรแวดล้อมด้วยไลบรารีดอตเอ็นวี ทำให้การนำขึ้นใช้งานจริงเป็นมาตรฐาน

3.2.3.2 ส่วนติดต่อหน้าเว็บ พัฒนาด้วย เอชทีเอ็มแอล, ซีเอสเอส, จาวาสคริปต์ และใช้ไลบรารี ชาร์ตเจเอส สำหรับแสดงกราฟแท่ง กราฟเรดาร์ และตารางสรุปสถิติ ผู้ใช้สามารถกดเลือกชื่อผู้เล่นเพื่อแสดง ตารางสรุป ด้านล่างและสลับดูกราฟเปรียบเทียบได้ทันทีผ่านเว็บเบราว์เซอร์ทั้งบนคอมพิวเตอร์และอุปกรณ์พกพา

3.2.3.3 สำหรับการพัฒนาและดูแลโค้ด ใช้ วิวอล สตูดิโอ โค้ด เป็นเครื่องมือหลัก จัดการเวอร์ชันด้วย กิต และเก็บซอร์สโค้ดบน กิตฮับ ส่วนการเปิดให้บริการใช้งานจริงใช้ผู้ให้บริการคลาวด์ เร็นเดอร์ ในรูปแบบคอนเทนเนอร์ที่สามารถปรับขนาดตามปริมาณการใช้

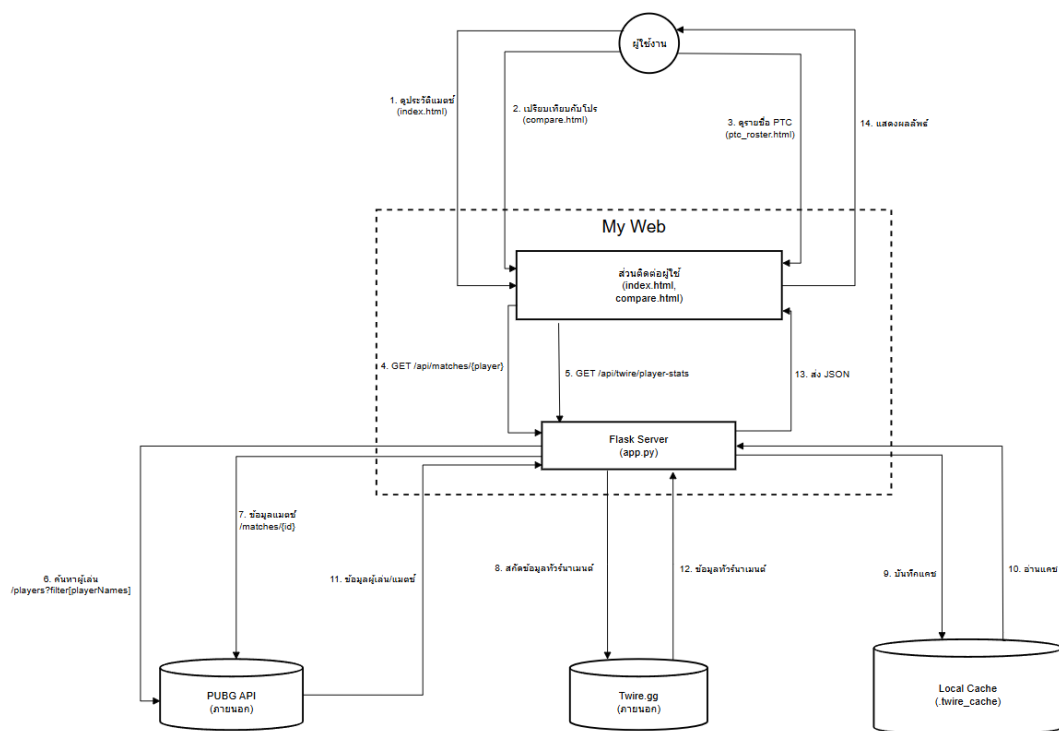
งาน ระบบมีการบันทึกล็อกและจัดการข้อผิดพลาดพื้นฐาน เพื่อให้ตรวจสอบและบำรุงรักษาได้สะดวก พร้อมปรับปรุงประสิทธิภาพและความเสถียรของบริการอย่างต่อเนื่อง

3.3 การออกแบบระบบ

การออกแบบระบบประกอบไปด้วยการออกแบบภาพรวมระบบ และการออกแบบส่วนติดต่อกับผู้ใช้

3.3.1 การออกแบบแผนภาพลำดับการทำงานของระบบ

3.3.1.1 แผนภาพลำดับการทำงานของระบบติดตามสถิติผู้เล่นพีซี (Data Flow Diagram : Context Diagram)



ภาพที่ 3.1 ภาพรวมการทำงานของระบบ

จากภาพประกอบ ภาพรวมระบบของระบบติดตามสถิติผู้เล่นพีซี ทำงานโดยมีผู้ใช้ ระบบประกอบด้วยผู้ใช้งาน

1) ผู้ใช้งาน เข้าถึงเว็บไซต์หลัก หน้าเปรียบเทียบ หรือหน้าดูสถิติผู้เล่นพีซี ซึ่งเป็นส่วนติดต่อผู้ใช้ของ

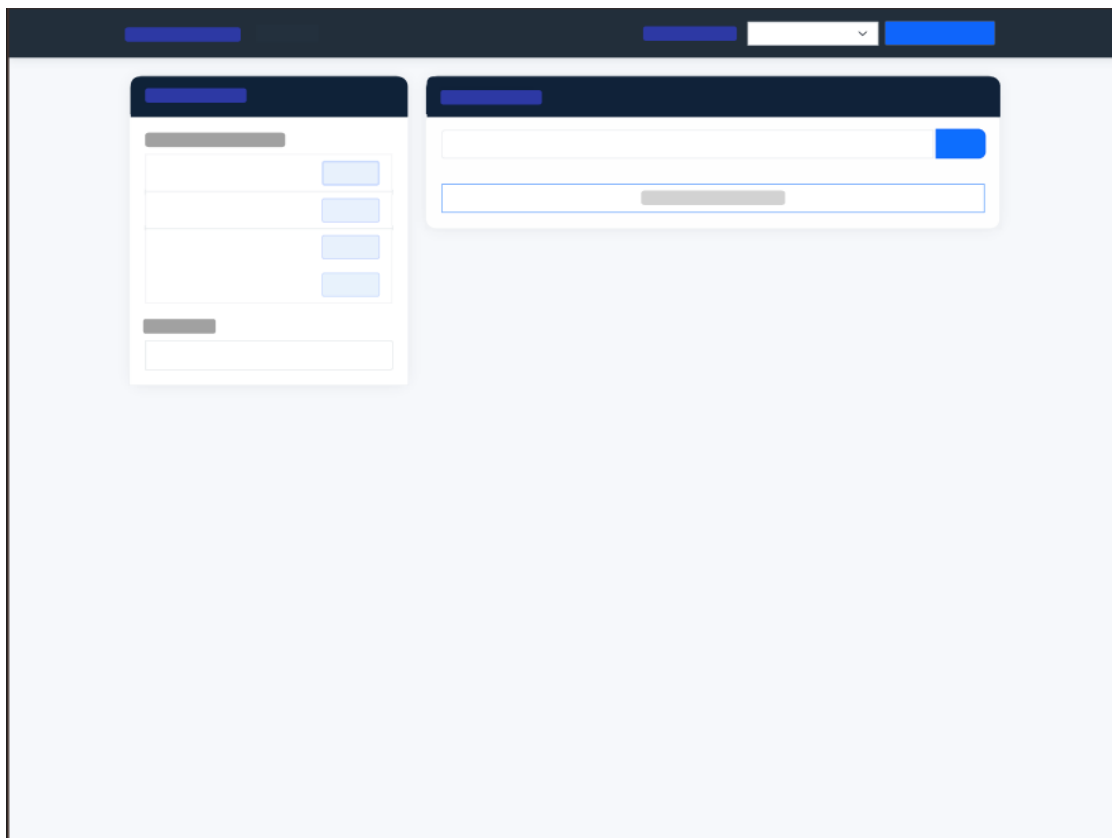
- 2) ส่วนติดต่อผู้ใช้ จะส่งคำขอข้อมูลไปยัง ฟลัสก์เซิร์ฟเวอร์
- 3) ฟลัสก์ เซิร์ฟเวอร์ ทำหน้าที่เป็นศูนย์กลางการประมวลผลข้อมูล สำหรับข้อมูลทัวร์นาเมนต์จะตรวจสอบ โรคอลแคชก่อน เพื่อดึงข้อมูลแคชที่บันทึกไว้ หากไม่มีแคชหรือแคชหมดอายุ จะดึงข้อมูลจาก ทไววีจีจี (ภายนอก) สำหรับข้อมูลผู้เล่นและแมตช์ จะส่งคำขอโดยตรงไปยัง พับจีเอพีไอ (ภายนอก) เพื่อค้นหาผู้เล่น และดึงข้อมูลแมตช์
- 4) ฟลัสก์ เซิร์ฟเวอร์ ประมวลผลและจัดรูปข้อมูลที่ได้ (รวมถึงการบันทึกแคชใหม่) แล้วส่งผลลัพธ์กลับในรูปแบบเจสัน
- 5) ส่วนติดต่อผู้ใช้นำข้อมูลเจสัน ไปแสดงผลเป็นกราฟและตารางให้ผู้ใช้งานเห็น

3.3.2 การออกแบบส่วนติดต่อกับผู้ใช้

การออกแบบโครงร่างหน้าจอของการพัฒนาระบบติดตามสถิติผู้เล่นพับจี ซึ่งเป็นการออกแบบส่วนประกอบของหน้าจอ รวมถึง ตำแหน่งการจัดวาง เพื่อที่จะใช้เป็นแนวทางในการจัดทำเว็บ โดยมีรายละเอียดในการ ออกแบบดังต่อไปนี้

3.3.2.1 ผลการทำงานที่ได้จากการออกแบบหน้าหลัก

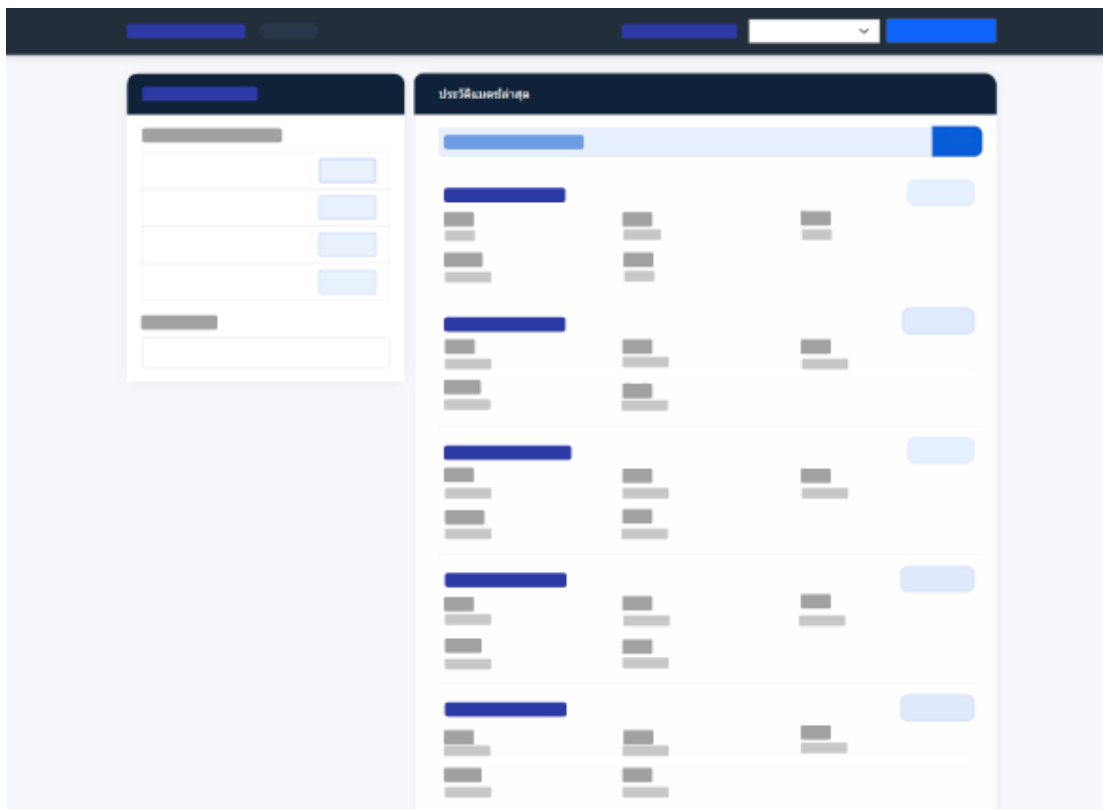
หน้าจอหลัก ส่วนแรกใส่ชื่อในเกมหรือชื่อผู้เล่น แล้วจะแสดงประวัติการเล่นของคนที่เราใส่ชื่อได้ ส่วนสองเปรียบเทียบสเตตัส เมื่อใส่ชื่อแล้วก็สามารถเปรียบเทียบกับรายชื่อโปรเพลเยอร์ได้ ส่วนสาม เมื่อกดปุ่มเปิดดูสเตตัสของผู้เล่นก็จะไปหน้าดูสเตตัสของผู้เล่นจากรายการ ฟีทชี ได้ ดังภาพที่ 3.2



ภาพที่ 3.2 หน้าจอหลัก

3.3.2.2 ผลการทำงานที่ได้จากการออกแบบหน้าหลักส่วนใส่ชื่อแล้วกดค้นหา

เมื่อกดใส่ชื่อแล้วค้นหา ก็จะขึ้น ประวัติการเล่นของคนนั้น ก็จะแสดงชื่อแมพ อันดับที่ยับ รายละเอียดต่างๆ เช่น การฆ่า ตาเมจ น็อค เดินทาง เวลาการอยู่รอด ดังภาพที่ 3.3



ภาพที่ 3.3 หน้าหลักส่วนใส่ชื่อแล้วกดค้นหา

3.3.2.3 ผลการทำงานที่ได้จากการออกแบบหน้าเปรียบเทียบโปรเพลเยอร์

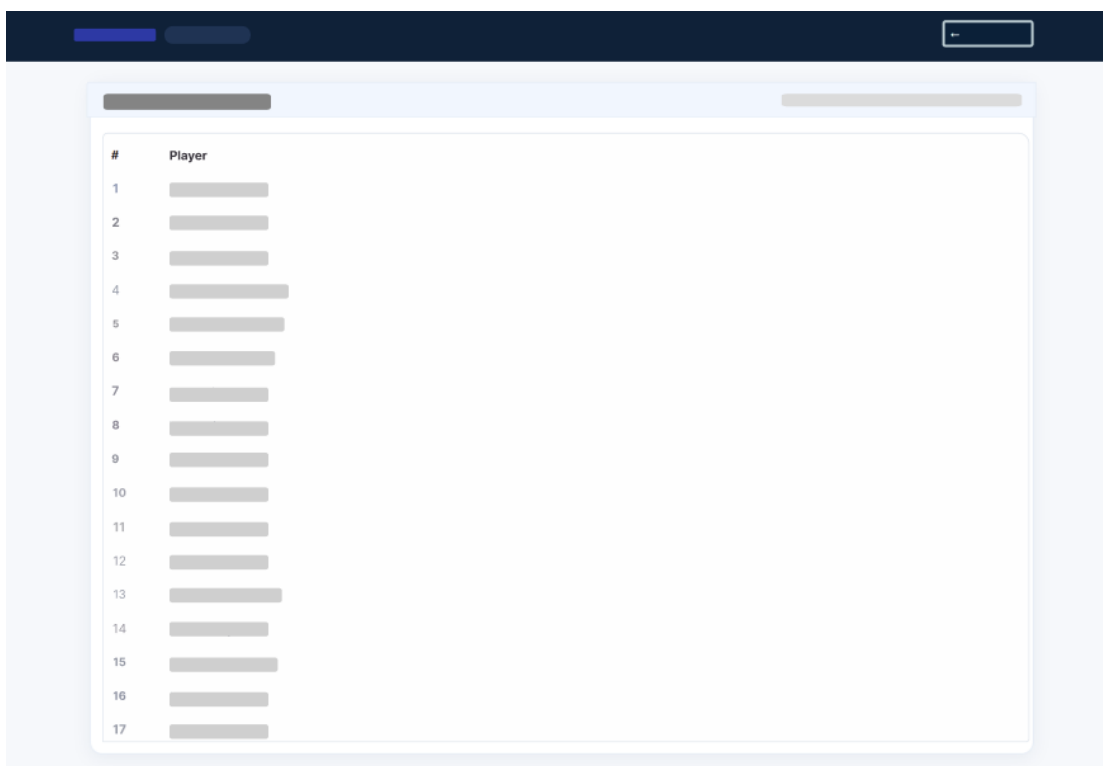
เมื่อกดเปรียบเทียบโปรเพลเยอร์จากหน้าจอหลักแล้วก็จะมาหน้านี้ ก็
จะแสดงสถิติที่เปรียบเทียบโปรเพลเยอร์กับคนที่ใส่ชื่อ ก็จะแสดงเป็นกราฟแท่ง โดยมีค่า ดา
เมจเฉลี่ยต่อเกม, เวลาอยู่รอดเฉลี่ยต่อเกม, อัตราการชนะเฉลี่ยต่อเกม, ฆ่าต่อการตาย และ
เส้นทางเฉลี่ยต่อเกม และมีตารางสรุปให้ด้านล่างอีกที (เว็บจริงสามารถเลื่อนดูได้จะไม่ยาวๆ
แบบนี้) ดังภาพที่ 3.4



ภาพที่ 3.4 หน้าเปรียบเทียบโปรเฟลเยอร์

3.3.2.3 ผลการทำงานที่ได้จากการออกแบบหน้าดูสถิติผู้เล่น

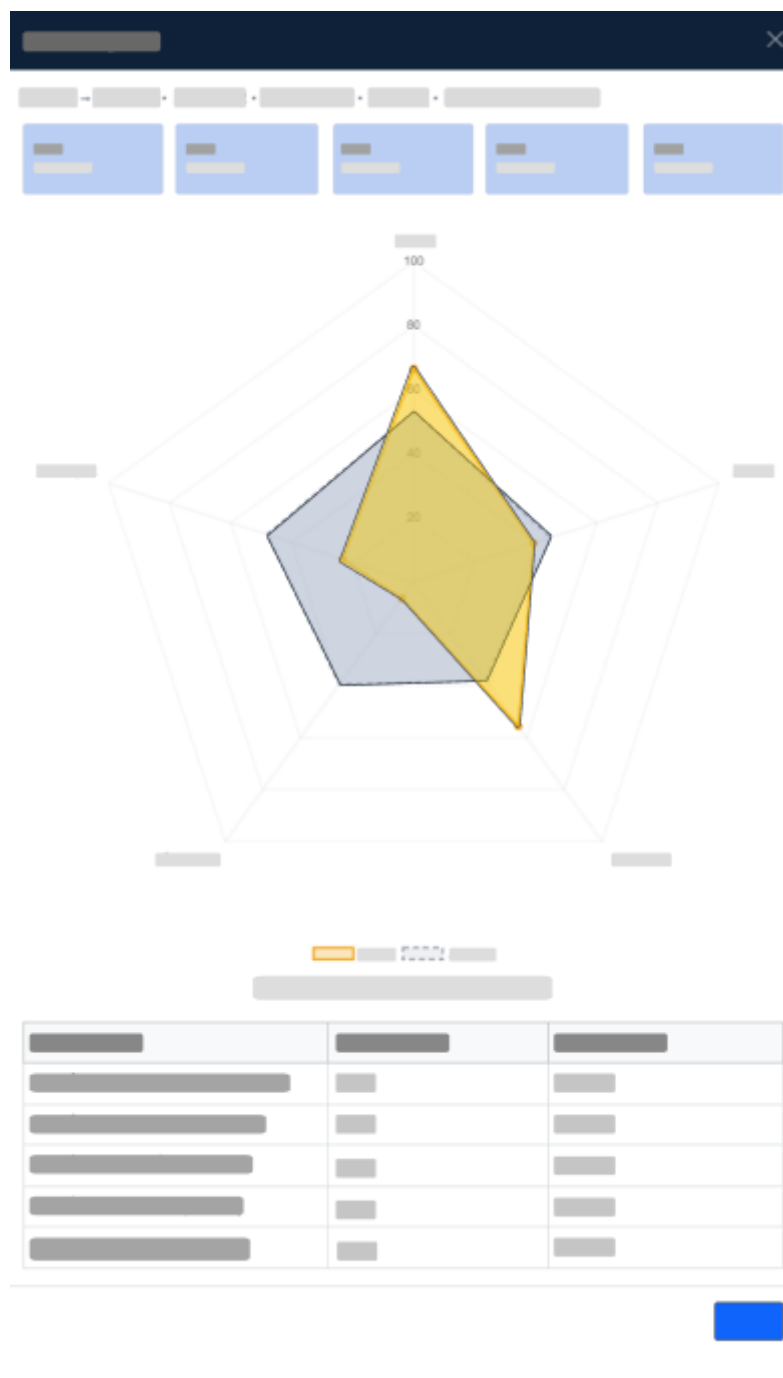
เมื่อกดเปิดดูสถิติของผู้เล่นจากหน้าหลักก็จะมาหน้านี้ จะแสดงรายชื่อผู้เล่นจากรายการ ฟิฟี่ซี สามารถคลิกที่ชื่อเพื่อดูสถิติของผู้เล่นคนนั้น รายชื่อผู้เล่นนี้จะไม่ได้เรียงตามลำดับความเก่งหรือลำดับที่จบ แต่จะเรียงตามรายชื่อทีม ดังภาพที่ 3.5



ภาพที่ 3.5 หน้าดูสถิติผู้เล่น

1) หน้าดูกราฟสถิติของผู้เล่น

เมื่อคลิกที่ชื่อผู้เล่นก็จะเจ้หน้าต่างป๊อบอัปขึ้นมากลางจอโดยจะแสดงค่าเฉลี่ยของทัวร์นาเมนต์ ฟิฟี่ซี ค่าที่ผู้เล่นคนนั้นทำได้ กราฟเรดาร์ชาร์ต เส้นสีเทาคือของค่าเฉลี่ยทัวร์ สีส้มคือของผู้เล่น ด้านล่างสุดจะเป็นตารางสรุปค่าที่ผู้เล่นทำได้เทียบกับค่าเฉลี่ยทัวร์อีกที ดังภาพที่ 3.6



ภาพที่ 3.6 หน้าดูกราฟสถิติของผู้เล่น

3.4 การพัฒนาระบบ

ในการศึกษาและพัฒนาระบบติดตามสถิติผู้เล่นฟุตบอลลีก ผู้พัฒนาได้กำหนดขั้นตอนการพัฒนาที่เน้นการบูรณาการข้อมูลจากแหล่งภายนอกและการแสดงผลเชิงวิเคราะห์ ดังต่อไปนี้

3.4.1 การวิเคราะห์และสำรวจความต้องการ เริ่มต้นจากการสำรวจผู้ใช้งานเป้าหมาย แมวมอง, ผู้จัดการทีม, ผู้เล่น เพื่อระบุตัวชี้วัดสถิติที่สำคัญ และกำหนดรูปแบบการแสดงผลที่ตอบโจทย์การวิเคราะห์ระบุตัวชี้วัดหลักที่ต้องประมวลผล เช่น อัตราการฆ่าต่อการตาย (เคตี), ดาเมจเฉลี่ยต่อเกม, เวลาอยู่รอดเฉลี่ย, และกำหนดขอบเขตของหน้าจอหลัก (โพรไฟล์, คอมแพร์, ฟังก์ชัน รอสเตอร์) และฟังก์ชันการค้นหาที่ต้องมี

3.4.2 การสำรวจและเชื่อมต่อแหล่งข้อมูลภายนอก ระบุแหล่งข้อมูลสำคัญ 2 ส่วน คือ เอพีไออย่างเป็นทางการของพับจี สำหรับประวัติแมตช์ล่าสุด และ ทวีร์จีจี สำหรับสถิติผู้เล่นในทัวร์นาเมนต์ ทดสอบการเชื่อมต่อ เอพีไอ และเขียนโค้ดต้นแบบเพื่อดึงข้อมูลจากเว็บไซต์ทัวร์นาเมนต์ด้วย เบราร์เซอร์ไร้ส่วนติดต่อผู้ใช้ เฮดเลส บราวเซอร์ คือ เฟลย์ไรต์ เพื่อแปลงข้อมูลตารางสถิติให้เป็นโครงสร้าง เจสัน ที่ใช้งานได้อีกจากการขุดข้อมูลเว็บ

3.4.3 การออกแบบสถาปัตยกรรมและการสร้างแบบจำลองข้อมูล วางโครงสร้างระบบแบบโคลเอนต์เซิร์ฟเวอร์ โดยใช้ ไพทอน คือ ฟลาสค์ เป็นส่วน แบ็คเอนต์ ในการจัดการ เอพีไอ และใช้ เอชทีเอ็มแอล จาวาสคริปต์ เป็นส่วน ฟรอนต์เอนต์ ในการแสดงผล ออกแบบผังข้อมูลเจสัน สำหรับการแลกเปลี่ยนข้อมูลระหว่างโคลเอนต์เซิร์ฟเวอร์ และกำหนดโครงสร้างสำหรับระบบแคชไฟล์ ไฟล์เบสแคช โดยระบุอายุแคชไว้ที่ 6 ชั่วโมง เพื่อลดการเรียก เอพีไอ และการขุดข้อมูลเว็บซ้ำซ้อน

3.4.4 การพัฒนาระบบส่วนหลังและการประมวลผลสถิติ การทำงานของ เอพีไอ และพัฒนาจุดให้บริการคำขอ เอพีไอ เอนด์พอยต์เช่น `/api/matches/<ชื่อผู้เล่น>` และ `/api/twirl/player-stats` สร้างโมดูลสำคัญเพื่อ ประมวลผลข้อมูล โดยมีฟังก์ชันเฉพาะทาง เช่น `_to_m_ss` (แปลงหน่วยเวลา), `_distance_str` (แปลงหน่วยระยะทาง), และ `kd_round_to_int` (กฎการปัด เคตี เฉพาะทาง) เพื่อแปลงข้อมูลดิบให้เป็นสถิติที่พร้อมใช้งานในส่วนหน้า

3.4.5 การพัฒนากลไกแคชและป้องกันการเรียกซ้ำ พัฒนากลไกที่ช่วยให้การดึงสถิติจาก ทวีร์จีจี ทำได้เร็วขึ้นและไม่ละเมิดข้อจำกัดการเรียกใช้งานเรท ลิ้มิตเขียนฟังก์ชัน `load_cache` และ `save_cache` เพื่อจัดเก็บข้อมูลสถิติที่ถูกขุดข้อมูลมาในรูปแบบไฟล์ เจสันชั่วคราว โดยใช้ ไลบรารี `pickle` ในการตรวจสอบและเรียกข้อมูลใหม่เมื่อแคชหมดอายุ

3.4.6 การพัฒนาระบบส่วนหน้าและการแสดงผลเชิง สร้างหน้าเว็บด้วย เอชทีเอ็มแอล และ บุตสแตมป์ 5 เพื่อให้ผู้ใช้สามารถค้นหาและเปรียบเทียบข้อมูลได้ใช้ จาวาสคริปต์ และ ไลบรารี `chart.js` ในการแสดงผลลัพท์ หน้าหลักแสดงรายการแมตช์ล่าสุด หน้าเปรียบเทียบแสดงผลสถิติด้วย บาร์ชาร์ต หน้าดูสถิติผู้เล่น แสดง เรตติ้งชาร์ต เพื่อเปรียบเทียบผู้เล่นกับค่าเฉลี่ยในทัวร์นาเมนต์

3.5 การทดสอบระบบ

การทดสอบระบบถูกแบ่งออกเป็น 4 ระดับหลัก เพื่อให้มั่นใจว่าทุกส่วนประกอบของระบบทำงานได้อย่างถูกต้อง แม่นยำ และตอบสนองต่อข้อจำกัดด้านการดึงข้อมูล โดยเฉพาะอย่างยิ่งการตรวจสอบความถูกต้องของสถิติที่ประมวลผลและการแสดงผลวิเคราะห์บนหน้าจอ

3.5.1 การทดสอบหน่วย

การทดสอบนี้มุ่งเน้นการตรวจสอบความถูกต้องของฟังก์ชันแต่ละส่วนบน แบ็คเอนด์ เพื่อยืนยันความแม่นยำในการประมวลผลสถิติที่ซับซ้อน ทดสอบความถูกต้องของฟังก์ชัน `_to_m_ss` (แปลงหน่วยเวลา) และ `_distance_str` (แปลงหน่วยระยะทาง) ว่าสามารถแปลงค่าดิบจาก เอพีไอ ให้เป็นหน่วยที่อ่านได้ถูกต้อง ทดสอบความถูกต้องของโมดูลคำนวณ เช่น การคำนวณค่า เคดี และฟังก์ชัน `kd_round_to_int` (การปัด เคดี เฉพาะทาง) เพื่อให้ผลลัพธ์ที่แสดงมีความแม่นยำตามที่กำหนด

3.5.2 การทดสอบการบูรณาการ

การทดสอบนี้มุ่งเน้นการตรวจสอบการทำงานร่วมกันระหว่างโมดูลต่าง ๆ และการเชื่อมต่อกับแหล่งข้อมูลภายนอกทดสอบการเรียกใช้ เอพีไออย่างเป็นทางการของเกม (ผ่าน `/api/matches/<player_name>`) เพื่อให้แน่ใจว่าสามารถดึงประวัติแมตช์ได้ถูกต้อง ทดสอบการทำงานของ เพลย์ไรท์ และ ทไวร์จีจี (ผ่าน `/api/twire/player-stats`) ว่าสามารถดึงข้อมูลตารางสถิติและแปลงเป็น เจสันได้สำเร็จ

ทดสอบกลไก การอ่านและเขียนแคช และการทำงานของ อายุแคช เพื่อตรวจสอบว่าระบบดึงข้อมูลใหม่เมื่อแคชหมดอายุ และไม่ดึงข้อมูลซ้ำเมื่อแคชยังใช้ได้

3.5.3 การทดสอบระบบตั้งแต่ต้นจนจบ

การทดสอบนี้เป็นการจำลองการใช้งานจริง เพื่อตรวจสอบการทำงานของระบบทั้งหมดตั้งแต่ ฟรอนต์เอนด์จนถึงแบ็คเอนด์ ทดสอบการค้นหาผู้เล่น ผ่านช่องค้นหาบน หน้าหลัก ว่าสามารถแสดงรายการแมตช์ล่าสุดได้อย่างถูกต้อง ทดสอบการแสดงผลกราฟตรวจสอบความถูกต้องของ เรดาร์ชาร์ต ในหน้าดูสถิติผู้เล่น ว่าแสดงค่าสถิติของผู้เล่นเทียบกับค่าเฉลี่ยทัวร์นาเมนต์ได้ตามที่คำนวณไว้จริง ตรวจสอบ บาร์ชาร์ต ในหน้าเปรียบเทียบ ว่าแสดงการเปรียบเทียบสถิติระหว่างผู้เล่น 2 คนได้อย่างถูกต้อง การจัดการการโหลดข้อมูล ทดสอบการแสดงผลสถานะ กำลังโหลด... หรือ เกิดข้อผิดพลาด ในกรณีที่มีการเรียก เอพีไอ ล้มเหลวหรือเกิดความล่าช้า

3.5.4 การทดสอบประสิทธิภาพ

การทดสอบนี้มุ่งเน้นการประเมินความเร็วและความสามารถในการตอบสนองของระบบ ซึ่งเป็นปัจจัยสำคัญเมื่อต้องดึงข้อมูลจากภายนอกทดสอบผลกระทบของการตั้งค่า อายุแคช 6 ชั่วโมง ต่อ เวลาในการตอบสนอง ของทไวร์จีจีเพื่อยืนยันว่าการแคชช่วยลดเวลา

โหลดลงอย่างมีนัยสำคัญ ทดสอบความเร็วในการโหลดหน้าหลักและหน้าคุณสมบัติผู้เล่นบน
เบราว์เซอร์และอุปกรณ์หลัก เดสก์ท็อปและมือถือ เพื่อให้มั่นใจว่ายูไอพร้อมใช้งานอย่างรวดเร็ว

บรรณานุกรม

- ขันติชัย รุจิระการโชติกุล. (2560). ระบบรวบรวมและจัดการมาตรฐานสำหรับการจัดทำเอกสาร Web API (A Standard Collection and Management System for Web API Documentation) [สารนิพนธ์วิทยาศาสตรมหาบัณฑิต, มหาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง]
- สุทิน อุทธรบูรณ์. (2560). กรณีศึกษาการสกัดข้อมูลงานวิจัยบนเว็บเพจด้วยเว็บครอว์เลอร์ [สารนิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต, มหาวิทยาลัยบูรพา]