

## 1. ชื่อโครงการ

ระบบจัดตารางอ่านหนังสืออัตโนมัติ

## 2. ที่มาและเหตุผล

นักศึกษาจำนวนมากในปัจจุบันประสบปัญหาในการวางแผนการอ่านหนังสืออย่างมีประสิทธิภาพ โดยเฉพาะในช่วงสอบที่มีเนื้อหาจำนวนมากและเวลาจำกัด การอ่านแบบไม่มีแผน หรือการวางแผนด้วยตนเองโดยไม่เข้าใจลำดับความสำคัญของเนื้อหา อาจนำไปสู่การเตรียมตัวสอบที่ไม่พร้อม ทำให้ผลการเรียนอาจไม่เป็นไปตามที่คาดหวัง และเกิดความเครียดจากการจัดการเวลาที่ไม่ดี

ระบบการจัดตารางอ่านหนังสือส่วนใหญ่ในปัจจุบัน เช่น การใช้กระดาษ ปฏิทิน หรือแอปพลิเคชันปฏิทินทั่วไป ยังคงเป็นแบบพื้นฐาน ไม่สามารถปรับเปลี่ยนตามพฤติกรรมผู้ใช้ หรือแนะนำตามความยากง่ายของวิชาได้ ทำให้ผู้ใช้ต้องใช้เวลาและแรงในการวางแผนด้วยตนเอง ซึ่งอาจไม่เหมาะสมกับแต่ละบุคคล

ด้วยเหตุนี้ เราจึงมีแนวคิดที่จะพัฒนาระบบ Smart Study Planner ที่สามารถจัดตารางอ่านหนังสือให้เหมาะสมกับเป้าหมายของผู้ใช้ โดยอิงจากข้อมูลต่าง ๆ เช่น ระยะเวลาในการเตรียมตัว วิชาที่ต้องสอบ ความยากของเนื้อหา และพฤติกรรม การอ่านของผู้ใช้ ระบบนี้จะช่วยให้ผู้ใช้สามารถเตรียมตัวสอบได้อย่างมีประสิทธิภาพและตรงจุดมากยิ่งขึ้น ลดความเครียดและเพิ่มโอกาสในการประสบความสำเร็จในการเรียน

## 3. วัตถุประสงค์

3.1 เพื่อพัฒนาและออกแบบระบบ Smart Study Planner ที่สามารถจัดตารางอ่านหนังสือส่วนบุคคลได้อย่างมีประสิทธิภาพ โดยพิจารณาจากข้อมูล เช่น ระยะเวลาในการเตรียมตัว วิชาที่ต้องสอบ ความยากของเนื้อหา และพฤติกรรม การอ่านของผู้ใช้

3.2 เพื่อสร้างกลไกการปรับตารางอ่านหนังสืออัตโนมัติ ที่สามารถปรับเปลี่ยนแผนการอ่านได้ตามความคืบหน้า หรือเมื่อมีการเปลี่ยนแปลงปัจจัยภายนอก เช่น วันสอบเลื่อน เพื่อให้ผู้ใช้สามารถปรับตัวได้ทันเวลาที่

3.3 เพื่อประเมินประสิทธิภาพและความพึงพอใจของผู้ใช้งาน ต่อระบบ Smart Study Planner ในด้านความสะดวกในการใช้งาน การช่วยเพิ่มประสิทธิภาพในการจัดตารางอ่านหนังสือ และลดความกังวลในการเตรียมตัวสอบ

## 4. ประโยชน์ที่คาดว่าจะได้รับ

#### 4.1 มีระบบช่วยวางแผนการอ่านหนังสืออย่างมีระบบ

ระบบสามารถวิเคราะห์ข้อมูลเวลาว่างและวิชาที่ต้องสอบ เพื่อสร้างตารางอ่านหนังสือที่เหมาะสม ลดปัญหาการอ่านไม่ทันหรืออ่านซ้ำโดยไม่จำเป็น

#### 4.2 มีระบบช่วยเพิ่มประสิทธิภาพในการเตรียมสอบ

ระบบสามารถจัดลำดับความสำคัญของเนื้อหาและแนะนำตารางอ่านหนังสือเฉพาะบุคคล ทำให้การเตรียมสอบมีความชัดเจนและมีประสิทธิภาพมากขึ้น

#### 4.3 มีระบบช่วยลดภาระในการวางแผนด้วยตนเอง

ผู้ใช้งานไม่จำเป็นต้องจัดตารางเองด้วยกระดาษหรือแอปทั่วไป เพราะระบบสามารถประมวลผลและสร้างตารางให้อัตโนมัติ

#### 4.4 มีระบบส่งเสริมการบริหารเวลาอย่างมีประสิทธิภาพ

ระบบช่วยให้ผู้ใช้ตระหนักถึงการจัดสรรเวลาอ่านหนังสืออย่างเหมาะสม และสามารถบาลานซ์เวลาใน ชีวิตประจำวันได้ดีขึ้น

#### 4.5 มีระบบที่สามารถนำไปพัฒนาและต่อยอดได้ในอนาคต

ระบบมีโครงสร้างที่รองรับการเพิ่มฟีเจอร์ใหม่ เช่น การเชื่อมต่อกับ Google Calendar, การสร้างแบบทดสอบหลังการอ่าน, การใช้ AI วิเคราะห์พฤติกรรมการอ่าน เพื่อเพิ่มความชาญฉลาดของระบบ

### 5. เครื่องมือที่ใช้ในการพัฒนา

#### 5.1 ฮาร์ดแวร์

- คอมพิวเตอร์ส่วนบุคคล

ใช้สำหรับเขียนโปรแกรม พัฒนาเว็บแอปพลิเคชัน และทดสอบระบบ ทั้งในส่วน Frontend และ Backend รวมถึงการจัดการฐานข้อมูล

- อุปกรณ์เชื่อมต่ออินเทอร์เน็ต

ใช้สำหรับเชื่อมต่อระบบกับอินเทอร์เน็ตเพื่อทดสอบการใช้งานในสภาพแวดล้อมจริง และทดสอบการเข้าถึงระบบจากผู้ใช้ปลายทาง

#### 5.2 ซอฟต์แวร์ (Software)

- Visual Studio Code  
ใช้เป็นเครื่องมือหลักในการพัฒนาเว็บแอปพลิเคชัน Smart Study Planner ทั้งในส่วน Frontend และ Backend
- Python Interpreter  
ใช้สำหรับรันภาษา Python และรองรับการทำงานของเฟรมเวิร์ก Django ซึ่งเป็นเทคโนโลยีหลักของระบบ
- Django Framework  
ใช้สำหรับพัฒนาเว็บแอปพลิเคชันแบบ Full-Stack รองรับการจัดการฐานข้อมูล การจัดการผู้ใช้ และการสร้าง API สำหรับระบบแนะนำตารางอ่านหนังสืออัจฉริยะ
- PostgreSQL  
ใช้เป็นฐานข้อมูล สำหรับจัดเก็บข้อมูลผู้ใช้ ตารางอ่านหนังสือ ข้อมูลวิชา วันสอบ และสถิติการอ่าน
- Supabase  
ใช้สำหรับจัดเก็บไฟล์ประกอบการเรียน เช่น เอกสารการเรียนการสอน หรือไฟล์แนบอื่น ๆ ที่ผู้ใช้นำมาอัปโหลดเข้าสู่ระบบ
- Gemini (Google AI Platform)  
ใช้เป็นระบบปัญญาประดิษฐ์สำหรับให้คำแนะนำเกี่ยวกับการจัดตารางอ่านหนังสือ และแสดงคำแนะนำเมื่อตารางเวลาของผู้ใช้อาจไม่เพียงพอ
- Git และ GitHub  
ใช้สำหรับควบคุมเวอร์ชันของซอร์สโค้ด (Version Control) และเป็นพื้นที่จัดเก็บโค้ดกลางสำหรับการพัฒนาร่วมกัน
- Google Chrome เวอร์ชันล่าสุด  
ใช้สำหรับทดสอบการทำงานของเว็บแอปพลิเคชัน และตรวจสอบการแสดงผลของระบบจากมุมมองของผู้ใช้จริง

### 5.3 ภาษาโปรแกรมที่ใช้

- Python เวอร์ชัน 3.12
- HTML5
- CSS3
- JavaScript

- SQL

#### 5.4 ฐานข้อมูลที่ใช้

- PostgreSQL

### 6. Functional Requirements

#### 6.1 ผู้ใช้ทั่วไป/นักศึกษา(User/Student)

- สมัครสมาชิก
- เข้าสู่ระบบ
- กรอกข้อมูลวิชา (ชื่อวิชา, ความยาก, วันที่สอบ)
- เพิ่มไฟล์การเรียนการสอน(.pdf .txt)
- กำหนดเวลาว่างในแต่ละวัน/สัปดาห์
- สร้างตารางอ่านหนังสืออัตโนมัติ
- ดูแผนการอ่าน (รายวัน / ปฏิทิน)
- ติดตามว่าอ่านจบแล้วในแต่ละวิชา
- ทำแบบทดสอบหลังเรียน
- สรุปเนื้อหาหลังอ่าน
- ดูความคืบหน้าของแผน
- แก้ไขวิชา และ สร้างแผนใหม่
- รับการแจ้งเตือน (เวลาอ่านใกล้ถึง, มีเวลาเพิ่ม, ใกล้ถึงวันสอบ)
- เชื่อมต่อ Google Calendar
- แสดงความคิดเห็นหรือให้คำแนะนำเพิ่มเติมเกี่ยวกับระบบ

#### 6.2 ระบบ (System)

- ประมวลผลข้อมูลวิชา + เวลาว่าง
- คำนวณและจัดตารางอ่านหนังสือให้อัตโนมัติ
- ปรับแผนอ่านใหม่เมื่อมีการเปลี่ยนแปลง
- คำนวณเปอร์เซ็นต์ความคืบหน้าตามวิชา ที่ติ๊กแล้ว
- สร้างแบบทดสอบหลังเรียน
- สรุปเนื้อหาหลังผู้ใช้ติ๊กว่าอ่านแล้ว

- แจ้งเตือนความคืบหน้าในการอ่านแต่ละวิชา
- แจ้งเตือนเมื่อใกล้ถึงวันสอบ

## 7. Non-Functional Requirements

### 7.1 ประสิทธิภาพ (Performance)

- ระบบต้องสามารถประมวลผลและสร้างตารางอ่านหนังสืออัตโนมัติได้เร็วและไม่หน่วงเกินไป หลังจากผู้ใช้กดยืนยันข้อมูล
- ระบบต้องรองรับการประมวลผลคำสั่งจาก AI (Gemini) เพื่อสรุปเนื้อหาหรือสร้างแบบทดสอบ โดยมีการแสดงสถานะการโหลด (Loading State) หากใช้เวลาเกิน 3 วินาที เพื่อประสบการณ์ที่ดีของผู้ใช้
- ระบบฐานข้อมูล (PostgreSQL) ต้องรองรับการเรียกดูข้อมูลตารางเรียนของผู้ใช้งานพร้อมกันได้โดยไม่มีคความล่าช้า

### 7.2 ความปลอดภัย (Security)

- รหัสผ่านของผู้ใช้งานต้องถูกเข้ารหัส (Encryption) ก่อนบันทึกลงฐานข้อมูล (เช่น ใช้มาตรฐาน pbkdf2\_sha256 ของ Django)
- ไฟล์เอกสารประกอบการเรียน (.pdf, .txt) ที่อัปโหลดขึ้น Supabase ต้องมีการกำหนดสิทธิ์การเข้าถึง ให้เฉพาะเจ้าของบัญชีเท่านั้นที่เปิดดูได้
- ระบบต้องมีการยืนยันตัวตน (Authentication) ก่อนเข้าถึงข้อมูลส่วนตัวและตารางอ่านหนังสือ

### 7.3 ความน่าเชื่อถือ (Reliability & Availability)

- ระบบต้องสามารถทำงานได้ตลอด 24 ชั่วโมง เพื่อให้นักศึกษาสามารถเข้ามาดูตารางอ่านหนังสือได้ตลอดเวลา
- ข้อมูลตารางอ่านหนังสือและความคืบหน้า (Progress) ต้องมีการบันทึกทันทีที่มีการเปลี่ยนแปลง เพื่อป้องกันข้อมูลสูญหายหากระบบขัดข้อง

### 7.4 การใช้งาน (Usability)

- ระบบต้องมีการออกแบบหน้าจอแบบ Responsive Design เพื่อให้สามารถแสดงผลและใช้งานได้ดี
- User Interface (UI) ต้องมีความเรียบง่าย เข้าใจได้ง่าย มีการแจ้งเตือนที่ชัดเจนเมื่อใกล้ถึงเวลาอ่านหนังสือ

### 7.5 การขยายระบบ (Scalability & Maintainability)

- Source Code ต้องมีการจัดเก็บเวอร์ชันผ่าน Git/GitHub เพื่อให้ง่ายต่อการพัฒนาต่อยอดและแก้ไขบั๊กในอนาคต

- โครงสร้างโค้ดต้องเขียนตามมาตรฐานของ Django Framework เพื่อรองรับการเพิ่มฟีเจอร์ในอนาคต เช่น การเชื่อมต่อกับ Google Calendar API