**ความรู้ที่ใช้ในโปรแกรมนี้**

1. Python : โค้ดนี้เขียนด้วยภาษา Python

2. tkinter : สร้าง GUI ทั้งหมดด้วยไลบรารี่ tkinter

3. SQLite : ใช้ SQLite ในการสร้างและจัดการกับฐานข้อมูลของโปรแกรม และจัดเก็บไฟล์ลงในฐานข้อมูล

4. Object-Oriented Programming (OOP) : ออกแบบโปรแกรมในรูปแบบ OOP ด้วยการสร้างคลาส LibraryManager สำหรับจัดการฟังก์ชันหลักของโปรแกรม

5. Error Handling : ใช้ try และ except เพื่อจัดการกับข้อผิดพลาด เช่น การกรอกข้อมูล

6. CRUD : จัดการกับฐานข้อมูลด้วยการ Create (เพิ่ม), Read (อ่าน), Update (แก้ไข), และ Delete (ลบ) ข้อมูล

**การทำงานของโปรแกรม**

1. นำเข้าโมดูล:

- `tkinter`: ใช้สำหรับการสร้างและจัดการ GUI.

- `sqlite3`: ใช้สำหรับการจัดการฐานข้อมูล SQLite.

2. คลาส LibraryApp:

- \_\_init\_\_: ในเมธอดนี้จะทำการสร้างหน้าต่าง GUI, ติดต่อกับฐานข้อมูล, และเรียกฟังก์ชันอื่น ๆ ภายในคลาสเพื่อเริ่มต้นแอปพลิเคชัน.

- style\_widgets: ปรับแต่งรูปแบบของวิดเจ็ตที่อยู่ภายในหน้าต่าง GUI.

- create\_widgets: สร้างวิดเจ็ตต่าง ๆ ที่จะปรากฏในหน้าต่าง GUI ได้แก่ ปุ่ม, ช่องข้อความ, และรายการหนังสือ.

- add\_book: เพิ่มหนังสือใหม่ลงในฐานข้อมูล

- borrow\_book: ยืมหนังสือจากฐานข้อมูลและเปลี่ยนสถานะของหนังสือนั้น ๆ เป็น "Borrowed".

- return\_book: คืนหนังสือเข้าสู่ฐานข้อมูลและเปลี่ยนสถานะของหนังสือกลับเป็น "Available".

- edit\_book: แก้ไขรายละเอียดของหนังสือ.

- display\_books: แสดงรายการหนังสือทั้งหมดที่มีในฐานข้อมูลบนหน้าต่าง GUI.

- display\_borrowers: แสดงรายการผู้ยืมหนังสือทั้งหมดที่มีในฐานข้อมูลบนหน้าต่าง GUI.

3. if \_\_name\_\_ == "\_\_main\_\_": เริ่มต้นการทำงานของโปรแกรม.

ภาพรวม: เมื่อเริ่มต้นแอปพลิเคชัน, ผู้ใช้จะเห็นหน้าต่าง GUI ที่มีช่องกรอกข้อมูลหนังสือ, รายการหนังสือ, ช่องกรอกข้อมูลผู้ยืม, รายการผู้ยืม, และปุ่มสำหรับการจัดการหนังสือและการยืม-คืน. การทำงานทั้งหมดจะเก็บข้อมูลลงในฐานข้อมูล SQLite ที่ชื่อว่า "library.db"