รายงานโครงงาน Life Health

บทที่ 1: บทนำและแนวคิดโครงงาน

1.1 ที่มา

ในปัจจุบัน ผู้คนเริ่มให้ความสำคัญกับการดูแลสุขภาพมากขึ้น ทั้งในด้านร่างกายและจิตใจ โดยมีเทคโนโลยีเข้ามามีบทบาทสำคัญในการช่วยติดตามพฤติกรรมสุขภาพ เช่น การนอนหลับ การดื่มน้ำ และอัตราการเต้นของหัวใจ อย่างไรก็ตาม แอปพลิเคชันติดตามสุขภาพที่มีอยู่ในปัจจุบันบางส่วนยังคงซับซ้อน ใช้งานยาก หรือไม่ตอบโจทย์ผู้ใช้งานบางกลุ่ม โดยเฉพาะกลุ่มผู้สูงอายุที่ไม่คุ้นชินกับเทคโนโลยี

จากปัญหาดังกล่าวจึงเกิดแนวคิดในการพัฒนาโครงการ Life Health Tracker (LHT) ซึ่งเป็นระบบติดตามสุขภาพที่เน้นความเรียบง่าย ใช้งานง่าย เหมาะสำหรับผู้ใช้ทุกวัย โดยสามารถบันทึกข้อมูลสุขภาพประจำวันและดูผลย้อนหลังได้อย่างสะดวก

1.2 วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาระบบติดตามสุขภาพที่ใช้งานง่ายและเข้าถึงได้ทุกเพศทุกวัย
- เพื่อนำเสนอข้อมูลสุขภาพในรูปแบบที่เข้าใจง่าย เช่น กราฟ ตัวเลข และสี
- เพื่อให้ผู้ใช้สามารถบันทึก ตรวจสอบ และประเมินสุขภาพของตนเองได้ทุกวัน

1.3 กลุ่มเป้าหมาย

- ผู้ใช้ทั่วไป
- ผู้ที่ใส่ใจสุขภาพ
- ผู้สูงอายุที่ไม่ถนัดเทคโนโลยี

บทที่ 2: การออกแบบระบบ

2.1 แผนภาพแสดงสถาปัตยกรรมของระบบ (System Architecture Diagram)

ระบบ Life Health Tracker (LHT) ถูกออกแบบโดยใช้โครงสร้างแบบสถาปัตยกรรม 4 ชั้น (Four-layer Architecture) ซึ่งแต่ละชั้นมีหน้าที่เฉพาะและเชื่อมโยงการทำงานกันอย่างเป็นลำดับขั้น โดยมีรายละเอียดดังนี้:

ชั้นที่ 1: ชั้นแสดงผลต่อผู้ใช้ (User Interface Layer)

หน้าที่: เป็นจุดเชื่อมต่อหลักระหว่างผู้ใช้งานกับระบบ

- Mobile Application: แอปพลิเคชันที่รองรับระบบปฏิบัติการ iOS และ Android
- Web Browser: เว็บไซต์ที่สามารถเข้าถึงผ่านเบราว์เซอร์ทั่วไป

กลุ่มผู้ใช้งาน ได้แก่ เจ้าหน้าที่โรงพยาบาล ผู้ดูแลระบบ และผู้ป่วย

ชั้นที่ 2: ส่วนติดต่อผู้ใช้ด้านหน้า (Frontend Layer)

หน้าที่: รับผิดชอบในการจัดแสดงผลข้อมูล การประมวลผลเบื้องต้น และการโต้ตอบกับผู้ใช้

- HTML / CSS / JavaScript: เทคโนโลยีพื้นฐานสำหรับโครงสร้าง การตกแต่ง
- State Management: การจัดการสถานะข้อมูลในแต่ละหน้า
- Routing: การกำหนดเส้นทางระหว่างหน้าเว็บไซต์ต่าง ๆ

ชั้นที่ 3: ส่วนประมวลผลเบื้องหลัง (Backend Layer)

หน้าที่: ทำหน้าที่เป็นตัวกลางในการประมวลผลและเชื่อมต่อข้อมูลระหว่าง Frontend และฐานข้อมูล

- JSON Server: ระบบจำลอง API สำหรับพัฒนาและทดสอบ
- RESTful API: รูปแบบการเชื่อมต่อแบบมาตรฐาน

- HTTP Methods:
    - GET: ดึงข้อมูลจากฐานข้อมูล
    - POST: เพิ่มข้อมูลใหม่เข้าสู่ระบบ
    - PUT: แก้ไขข้อมูลที่มีอยู่
    - DELETE: ลบข้อมูล

หมายเหตุ: Backend ทำงานที่พอร์ต 3001

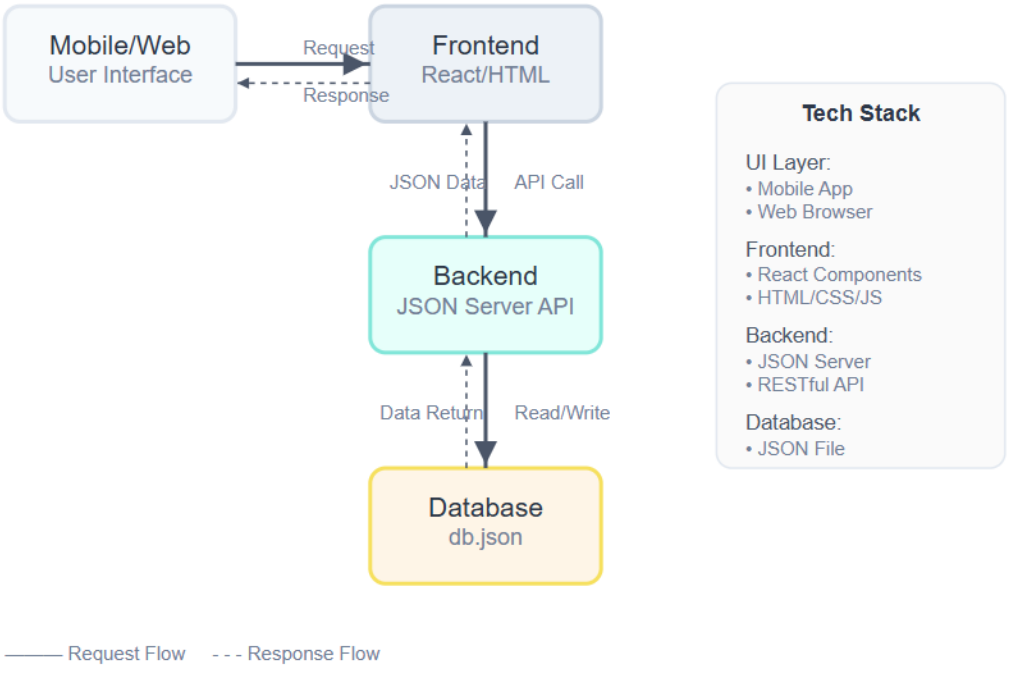
ชั้นที่ 4: ฐานข้อมูล (Database Layer)

หน้าที่: เป็นแหล่งจัดเก็บข้อมูลถาวรของระบบ

- JSON File Storage: การจัดเก็บข้อมูลในรูปแบบไฟล์ .json
- ประเภทข้อมูลที่จัดเก็บ: รายการกิจกรรมโรงพยาบาล วันที่และรายละเอียดกิจกรรม รูปภาพประกอบ และลิงก์เอกสาร

2.2 การไหลของข้อมูลภายในระบบ (Data Flow)

การไหลของข้อมูลในระบบสามารถแบ่งออกเป็น 2 ส่วนหลัก ได้แก่:



- Request Flow (เส้นทึบ ———)
    - ผู้ใช้กรอกข้อมูลผ่านแบบฟอร์มในหน้าเว็บ
    - ข้อมูลถูกส่งไปยัง Frontend เพื่อประมวลผลและจัดเตรียม API Request
    - API Request ถูกส่งไปยัง Backend ผ่าน HTTP Protocol
    - Backend ประมวลผลและจัดการคำสั่งกับฐานข้อมูล (Database)
- Response Flow (เส้นประ - - -)
    - ฐานข้อมูลตอบกลับข้อมูลให้กับ Backend
    - Backend ส่ง JSON Response กลับไปยัง Frontend
    - Frontend อัปเดตหน้าจอให้ผู้ใช้เห็นข้อมูลที่เปลี่ยนแปลง

2.3 ตัวอย่างการทำงานของระบบ

1. ผู้ใช้กรอกข้อมูลกิจกรรมผ่านแบบฟอร์มในหน้าเว็บไซต์ (UI Layer)
2. ระบบ Frontend (React) ประมวลผลข้อมูลและจัดส่ง HTTP POST Request
3. JSON Server (Backend) รับข้อมูลและทำการบันทึกลงไฟล์ db.json

ตัวอย่างโครงสร้างข้อมูลใน db.json


```
{
  "dailyRecords": [
    {
      "id": "6c8e",
      "date": "2025-06-05",
```

```
    "waterIntake": 2600,
    "sleepHours": 7,
    "exercise": "โยคะ",
    "stressLevel": "ปานกลาง"
  }
]
```


4. ระบบส่งผลลัพธ์การทำงานกลับผ่านทุกชั้น
5. หน้าเว็บแสดงข้อความแจ้งเตือนผู้ใช้ว่า "บันทึกสำเร็จ"

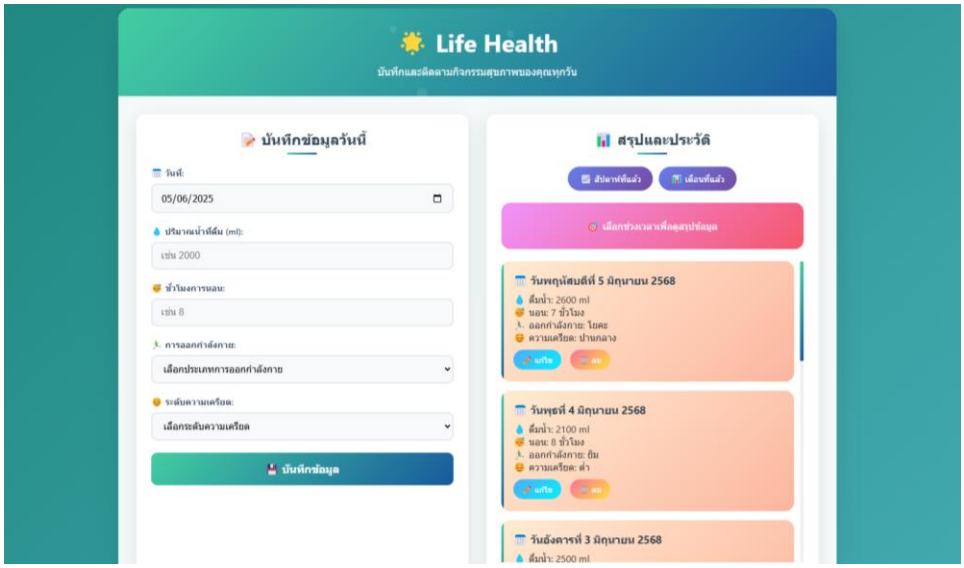
2.4 ข้อดีของสถาปัตยกรรมระบบ

- การแยกหน้าที่ชัดเจน: แต่ละชั้นมีความรับผิดชอบเฉพาะด้าน ช่วยให้ดูแลและพัฒนาระบบได้ง่าย
- ความยืดหยุ่นในการพัฒนา: สามารถปรับเปลี่ยนหรือพัฒนา Frontend และ Backend ได้อย่างอิสระ
- ต้นทุนต่ำ: ใช้ JSON Server และ JSON File แทนฐานข้อมูลจริง ลดต้นทุนในการจัดการระบบ
- เหมาะสำหรับโครงการขนาดเล็กถึงปานกลาง: หรือใช้สำหรับต้นแบบ (Prototype) ก่อนนำไปสู่ระบบขนาดใหญ่

บทที่ 3 การออกแบบ UX/UI

3.1 หน้าเว็บ

หน้าเว็บของแอปพลิเคชันแบ่งออกเป็นสองส่วนหลัก ได้แก่ ฝั่งซ้ายสำหรับการกรอกข้อมูล และ ฝั่งขวาสำหรับแสดงผลย้อนหลัง โดยมีรายละเอียดดังนี้:



ส่วนหัว (Header)

- ชื่อแอป: Life Health
- คำอธิบายใต้ชื่อแอป: “บันทึกและติดตามพฤติกรรมสุขภาพของคุณทุกวัน”
- สีพื้นหลัง: ไล่เฉดสีฟ้า-เขียว เพื่อสร้างความรู้สึกสดชื่นและผ่อนคลาย

ฝั่งซ้าย: ฟอร์มกรอกข้อมูลสุขภาพประจำวัน

ประกอบ ดังนี้:

- วันที่: ผู้ใช้สามารถเลือกวันที่ต้องการบันทึก
- ปริมาณน้ำที่ดื่ม (ml): กรอกเป็นตัวเลข

- จำนวนชั่วโมงการนอน: กรอกจำนวนชั่วโมง
- การออกกำลังกาย: เลือกจากเมนูแบบดรอปดาวน์ (เช่น เดิน, วิ่ง, ปั่นจักรยาน)
- ระดับความเครียด: เลือกจากเมนูแบบดรอปดาวน์ (เช่น ต่ำ, ปานกลาง, สูง)
- ปุ่ม “บันทึกข้อมูล”: ออกแบบให้มีขนาดใหญ่ ใช้สีเขียวฟ้า มองเห็นได้ชัดเจน

ฝั่งขวา: แสดงผลย้อนหลัง

- มี ปุ่มให้ผู้ใช้เลือกดูข้อมูลย้อนหลัง ได้ตามช่วงเวลา เช่น เป็นสัปดาห์ หรือ เป็นเดือน
- การแสดงผลใช้ รูปแบบ Card UI แต่ละรายการมี:
    - สีพื้นหลังของการ์ด เพื่อสื่อถึงระดับสุขภาพหรือความรู้สึก
    - ข้อความสรุปแบบสั้น เช่น "ดีมาก" หรือ "ควรพักผ่อนเพิ่ม"

บทที่ 4 การพัฒนาและปัญหา

4.1 เทคโนโลยีที่ใช้ในการพัฒนา

ในการพัฒนาแอปพลิเคชัน Life Health ได้เลือกใช้เทคโนโลยีที่เหมาะสมกับงานพัฒนาเว็บแอปพลิเคชันขนาดเล็ก โดยมีรายละเอียดดังนี้:

- Frontend: ใช้ภาษา HTML, CSS, JavaScript เพื่อช่วยในการเพิ่มประสิทธิภาพของระบบ
- Backend: ใช้ JSON Server ในการจำลอง API สำหรับจัดเก็บข้อมูลและดึงข้อมูล
- Design Tool: ใช้ Figma ในการออกแบบหน้าตา (UI) และวางโครงสร้างการใช้งาน (UX)
- ฐานข้อมูล: ใช้ไฟล์ db.json เป็นฐานข้อมูลจำลอง ซึ่งสามารถเก็บข้อมูลสุขภาพในรูปแบบ JSON

4.2 ปัญหาและการแก้ไข

ปัญหา: ข้อมูลไม่แสดงผลบนหน้าจอ

สาเหตุ: API ไม่สามารถเชื่อมต่อกับ JSON Server ได้

แนวทางแก้ไข:

- ตรวจสอบว่า JSON Server กำลังทำงานอยู่ที่พอร์ต 3000
- ตรวจสอบว่าไฟล์ db.json มีข้อมูลที่ถูกต้องและอยู่ในรูปแบบที่เหมาะสม
- ทดสอบเรียก API ด้วยเครื่องมือ เบราว์เซอร์

บทที่ 5 การทดสอบและประเมินผล

5.1 วิธีการทดสอบ : เพื่อประเมินความสามารถในการใช้งานของระบบ ได้มีการ ทดสอบจริงกับกลุ่มผู้ใช้งาน คือ นักศึกษาโดยใช้วิธีการสังเกตการใช้งานจริง และสอบถามความคิดเห็นหลังการใช้งาน เพื่อประเมินความเข้าใจ ความสะดวก และความพึงพอใจของผู้ใช้

5.2 ผลที่ได้จากการทดสอบ

ผลการทดสอบพบว่า:

- ระบบสามารถใช้งานได้จริง ทั้งในส่วนของการกรอกข้อมูล และการดูประวัติย้อนหลัง
- UI ได้รับคำชมว่าเข้าใจง่าย มีความน่าสนใจ สีสันไม่ฉูดฉาดจนเกินไป และจัดวางองค์ประกอบเหมาะสม

5.2 ข้อเสนอแนะจากผู้ใช้

ผู้ใช้งานให้คำแนะนำ ดังนี้: ควรเพิ่ม ระบบแจ้งเตือน (Notification) หากผู้ใช้ยังไม่ได้กรอกข้อมูลประจำวัน และ อยากให้ วิเคราะห์ข้อมูลสุขภาพ เพื่อให้คำแนะนำในแต่ละวันได้

บทที่ 6 สรุปผลการดำเนินงาน

โครงการ Life Health Tracker (LHT) เป็นระบบติดตามสุขภาพรายวันที่ใช้งานง่าย เหมาะสำหรับผู้ใช้ทุกวัย โดยสามารถบันทึกข้อมูลสุขภาพ ตรวจสอบย้อนหลัง และดูภาพรวมสุขภาพในแต่ละวัน ระบบถูก

ออกแบบโดยเน้นความเรียบง่าย มีความน่าใช้ และสามารถพัฒนาเพิ่มเติมได้ในอนาคต เช่น การแจ้งเตือน หรือการวิเคราะห์ข้อมูลสุขภาพอย่างละเอียดขึ้น

โครงงานนี้ช่วยให้ผู้พัฒนาได้เรียนรู้กระบวนการออกแบบ พัฒนา และทดสอบระบบ พร้อมทั้งรับฟังความคิดเห็นจากผู้ใช้จริง เพื่อนำไปปรับปรุงระบบให้ดีขึ้นต่อไป