รายงาน

คู่มือการพัฒนาเว็บไซต์

ผู้จัดทำ

นางสาวปารวี ยิ่งใหญ่ รหัสนิสิต 66164119

เสนอ

รศ.ดร. สิทธิชัย ชูสำโรง

รายงานประกอบวิชา 104325-3 การพัฒนาแผนที่บนเว็บ

(Web GIS Development) ภาคการศึกษาที่ 1 ปีการศึกษา 2568

สาขาภูมิศาสตร์ คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม

มหาวิทยาลัยนเรศวร

คำนำ

การพัฒนาเว็บไซต์เป็นหนึ่งในทักษะที่สำคัญในยุคดิจิทัล ซึ่งไม่เพียงแต่จะช่วยในการแสดงข้อมูล แต่ยังเป็นเครื่องมือที่สามารถใช้ในการเผยแพร่ความรู้และการศึกษาได้อย่างมีประสิทธิภาพ สำหรับสาขาภูมิศาสตร์ การใช้เว็บไซต์ในการนำเสนอข้อมูลทางภูมิศาสตร์สามารถช่วยให้นักเรียน นักศึกษา และบุคคลทั่วไปเข้าถึงข้อมูลและแหล่งเรียนรู้ได้สะดวกยิ่งขึ้น

ในรายงานฉบับนี้ จะนำเสนอคู่มือการพัฒนาเว็บไซต์ในรูปแบบ Mini Project ซึ่งมุ่งเน้นไปที่การสร้างเว็บไซต์ที่สามารถใช้ในการนำเสนอข้อมูลที่เกี่ยวข้องกับภูมิศาสตร์ เช่น แผนที่แสดงข้อมูลนิสิต แผนที่แสดงข้อมูลคาเฟ่ของจังหวัดแพร่

หวังว่าคู่มือนี้จะเป็นประโยชน์ในการพัฒนาเว็บไซต์สำหรับการเรียนการสอนและการศึกษาในสาขาภูมิศาสตร์ รวมถึงสามารถนำไปปรับใช้ในโปรเจ็กต์จริงได้ในอนาคต

บทนำ

บทที่1

1.1) ที่มาและความสำคัญ

ในปัจจุบัน เว็บไซต์มีบทบาทสำคัญในชีวิตประจำวัน โดยเฉพาะในด้านการศึกษาและการแสดงข้อมูลภูมิศาสตร์ การพัฒนาเว็บไซต์ในสาขาภูมิศาสตร์เกิดขึ้นจากความต้องการในการนำเสนอข้อมูลภูมิศาสตร์อย่างมีประสิทธิภาพและเข้าถึงง่าย โดยใช้เทคโนโลยีที่ทันสมัย เช่น ระบบ Geographic Information System (GIS) ซึ่งช่วยให้ข้อมูลทางภูมิศาสตร์สามารถแสดงผลในรูปแบบดิจิทัลผ่านเว็บไซต์ได้ โดยไม่ต้องใช้ซอฟต์แวร์ที่ซับซ้อน

เว็บไซต์ที่พัฒนาขึ้นในสาขาภูมิศาสตร์ทำให้การเข้าถึงข้อมูลและการวิเคราะห์เชิงพื้นที่ง่ายขึ้น ไม่ว่าจะเป็นแผนที่ออนไลน์หรือข้อมูลเชิงพื้นที่แบบเรียลไทม์ ซึ่งช่วยให้นักเรียน นักศึกษา หรือผู้เชี่ยวชาญสามารถเข้าถึงข้อมูลได้สะดวกและรวดเร็ว

การพัฒนาเว็บไซต์นี้ยังมีความสำคัญในการสนับสนุนการศึกษาและการวิจัยทางภูมิศาสตร์ โดยสามารถนำเสนอข้อมูลที่ซับซ้อนได้ในรูปแบบที่เข้าใจง่าย อีกทั้งยังสามารถใช้เป็นเครื่องมือในการตัดสินใจในด้านการวางแผนและการจัดการทรัพยากรธรรมชาติ เช่น การจัดการภัยพิบัติ การวิเคราะห์สภาพภูมิอากาศ หรือการวางแผนการใช้ที่ดิน

เว็บไซต์ยังช่วยเสริมสร้างทักษะดิจิทัลของนักศึกษาในสาขาภูมิศาสตร์ โดยการออกแบบและพัฒนาเว็บไซต์ที่เกี่ยวข้องกับข้อมูลภูมิศาสตร์จะช่วยให้พวกเขามีทักษะการใช้เทคโนโลยีที่จำเป็นในโลกปัจจุบัน

สรุปแล้ว การพัฒนาเว็บไซต์ในสาขาภูมิศาสตร์ไม่เพียงแต่ทำให้การศึกษาและการเข้าถึงข้อมูลสะดวกขึ้น แต่ยังช่วยส่งเสริมการวิจัย การศึกษาด้านภูมิศาสตร์ และการตัดสินใจในภาคธุรกิจและการบริหารจัดการทรัพยากรธรรมชาติให้มีประสิทธิภาพมากยิ่งขึ้น

1.2) วัตถุประสงค์

1.เพื่อศึกษาและพัฒนาเว็บไซต์ที่ใช้ในการนำเสนอข้อมูลภูมิศาสตร์

2.เพื่อเสริมสร้างทักษะในการใช้เทคโนโลยีดิจิทัล

3.เพื่อส่งเสริมการใช้เว็บไซต์เป็นเครื่องมือในการวิเคราะห์และวิจัยทางภูมิศาสตร์

4.เพื่อพัฒนาเครื่องมือออนไลน์ที่ช่วยในการวิเคราะห์ข้อมูลภูมิศาสตร์

5.เพื่อฝึกฝนทักษะการออกแบบและพัฒนาเว็บไซต์

1.3) ประโยชน์ที่คาดว่าจะได้รับ

การเข้าใจพื้นฐานการพัฒนาเว็บไซต์:

- เรียนรู้พื้นฐานต่างๆ เช่น HTML, CSS, JavaScript ที่ใช้ในการสร้างเว็บไซต์ รวมถึงวิธีการเชื่อมต่อกับฐานข้อมูล (Database) และการใช้งานฟังก์ชันต่างๆ ของเว็บไซต์

การเรียนรู้เกี่ยวกับกระบวนการพัฒนาเว็บไซต์:

- เข้าใจขั้นตอนต่างๆ ในการพัฒนาเว็บไซต์ ตั้งแต่การวางแผน, ออกแบบ, พัฒนา, ทดสอบ และปรับปรุงเว็บไซต์

การพัฒนาเว็บไซต์ที่เหมาะสมกับผู้ใช้:

- สร้างเว็บไซต์ที่สามารถตอบสนองความต้องการของผู้ใช้ได้เป็นอย่างดี โดยใช้ข้อมูลและข้อคิดเห็นจากผู้ใช้ในการปรับปรุงเว็บไซต์

บทที่ 2

1.หน้าแรก หน้า login logout และ เมนู

index.html > ...

```html
<!DOCTYPE html>
<html lang="th">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Website</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body class="homepage">

<div id="content" class="homepage-container">
    <div class="icon-container">
        <a href="profile.html" class="icon-link">About Me</a>
        <a href="agi.html" class="icon-link">Agi 64-67</a>
        <a href="test.html" class="icon-link">Web คามคู่มือ</a>
        <a href="indexcafe.html" class="icon-link">Phrae</a>
    </div>
    <button id="logoutBtn">Logout</button>
</div>

<script>
    // ตรวจสอบว่าเคยล็อกอินหรือไม่
    if (!localStorage.getItem("isLoggedIn")) {
        window.location.href = "login.html";
    }

    // Logout
    document.getElementById("logoutBtn").addEventListener("click", function() {
        localStorage.removeItem("isLoggedIn");
        window.location.href = "logout.html";
    });
</script>

</body>
</html>
```

ส่วนที่ 1: <head> - ส่วนข้อมูลเกี่ยวกับหน้าเว็บ

ส่วนที่ 1: `<head>` - ส่วนข้อมูลเกี่ยวกับหน้าเว็บ

html Copy code

```html
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Website</title>
    <link rel="stylesheet" href="styles.css">
</head>
```

คำอธิบาย

<meta charset="UTF-8">:

- กำหนดการเข้ารหัสตัวอักษรเป็น UTF-8 ซึ่งรองรับการแสดงผลตัวอักษรภาษาไทยและตัวอักษรพิเศษอื่นๆ ได้ถูกต้อง

<meta name="viewport" content="width=device-width, initial-scale=1.0">:

- กำหนดการแสดงผลเว็บไซต์ให้เหมาะสมกับขนาดหน้าจอของอุปกรณ์มือถือ (Responsive Web Design)

<title>My Website</title>:

- กำหนดชื่อเว็บไซต์ที่จะแสดงในแท็บของเบราว์เซอร์

<link rel="stylesheet" href="styles.css">:

- เชื่อมโยงไฟล์ **CSS** (Cascading Style Sheets) ที่ชื่อว่า styles.css ซึ่งใช้ในการตกแต่งและออกแบบหน้าตาเว็บไซต์

ส่วนที่ 2: <body> - เนื้อหาหลักของเว็บไซต์

```
<body class="homepage">
  <div id="content" class="homepage-container">
    <div class="icon-container">
      <a href="profile.html" class="icon-link">About Me</a>
      <a href="agi.html" class="icon-link">Agi 64-67</a>
      <a href="test.html" class="icon-link">Web ตามคู่มือ</a>
      <a href="indexcafe.html" class="icon-link">Phrae</a>
    </div>
    <button id="logoutBtn">Logout</button>
  </div>
</body>
```

คำอธิบาย

<body class="homepage">:

- ตั้งชื่อคลาสให้กับ <body> ว่า homepage ซึ่งสามารถใช้ในไฟล์ CSS หรือ JavaScript เพื่อจัดการกับสไตล์หรือฟังก์ชันต่างๆ

** **:

- สร้าง **div** เพื่อเก็บเนื้อหาหลักของเว็บไซต์ไว้ โดยใช้ id="content" และ class="homepage-container" สำหรับการจัดการผ่าน CSS หรือ JavaScript

** **:

- สร้าง **div** อีกตัวเพื่อเก็บลิงก์ต่างๆ ที่จะแสดงผลในหน้าเว็บ ซึ่งมีการใช้คลาส icon-container เพื่อสามารถควบคุมการแสดงผลผ่าน CSS

<a href="profile.html" class="icon-link">About Me</a>:

- เป็นลิงก์ที่เชื่อมโยงไปยังหน้าต่างๆ ของเว็บไซต์ (เช่น profile.html, agi.html, test.html, และ indexcafe.html) โดยแต่ละลิงก์มีข้อความที่แสดงในหน้าเว็บ (เช่น "About Me", "Agi 64-67")

<button id="logoutBtn">Logout</button>:

- ปุ่มที่เมื่อคลิกแล้วจะทำการออกจากระบบ (Logout)

ส่วนที่ 3: <script> - การทำงานด้วย JavaScript

```
<script>

  // ตรวจสอบว่าเคยล็อกอินหรือไม่

  if (!localStorage.getItem("isLoggedIn")) {

    window.location.href = "login.html";

  }
```

```
  // Logout

   document.getElementById("logoutBtn").addEventListener("click", function() {

      localStorage.removeItem("isLoggedIn");

      window.location.href = "logout.html";

   });

</script>
```

คำอธิบาย

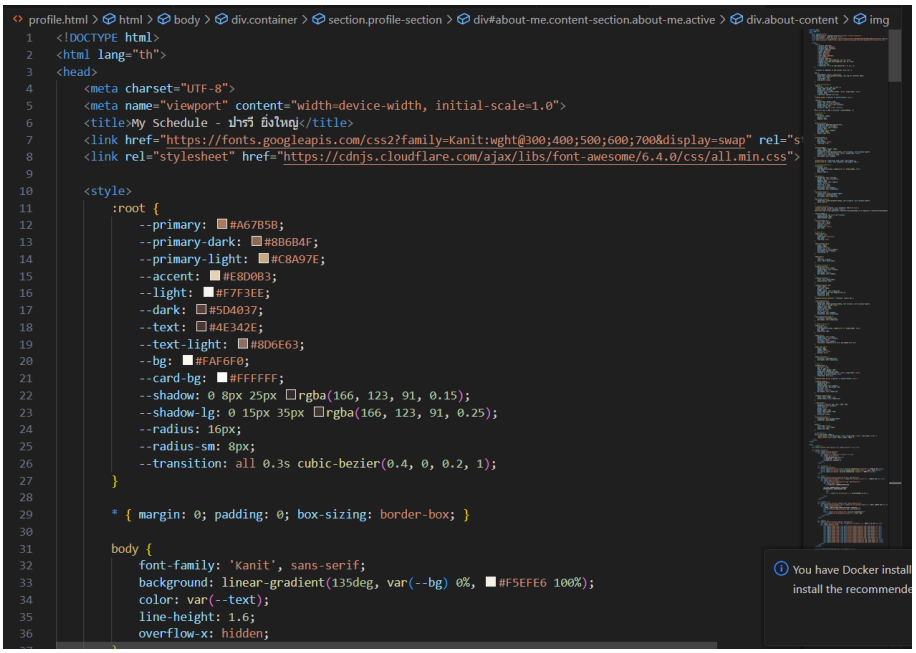
ตรวจสอบการล็อกอิน:

- localStorage.getItem("isLoggedIn") จะเช็คค่าที่เก็บใน localStorage ว่ามีการเก็บข้อมูลการล็อกอินหรือไม่ ถ้าไม่มีค่าหมายความว่ายังไม่ได้ล็อกอิน
- ถ้ายังไม่ได้ล็อกอิน (if (!localStorage.getItem("isLoggedIn"))), ระบบจะพาผู้ใช้ไปยังหน้า login.html เพื่อทำการล็อกอินใหม่

ฟังก์ชันการล็อกเอาท์ (Logout):

- document.getElementById("logoutBtn"): หาองค์ประกอบของปุ่มที่มี id="logoutBtn"
- .addEventListener("click", function() {...}): เมื่อมีการคลิกปุ่มนี้จะทำการดำเนินการในฟังก์ชันที่กำหนด
- localStorage.removeItem("isLoggedIn"): ลบข้อมูลการล็อกอินออกจาก localStorage
- window.location.href = "logout.html": เปลี่ยนหน้าไปยัง logout.html เพื่อแสดงหน้าที่ผู้ใช้ถูกออกจากระบบแล้ว

2.หน้าสอง About Me



```
profile.html > html > body > div.container > section.profile-section > div#about-me.content-section.about-me.active > div.about-content > img
<!DOCTYPE html>
<html lang="th">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Schedule - ปารวี ยิ่งใหญ่</title>
    <link href="https://fonts.googleapis.com/css2?family=Kanit:wght@300;400;500;600;700&display=swap" rel="s
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <style>
        :root {
            --primary: #A67B5B;
            --primary-dark: #8B6B4F;
            --primary-light: #C8A97E;
            --accent: #E8D0B3;
            --light: #F7F3EE;
            --dark: #5D4037;
            --text: #4E342E;
            --text-light: #8D6E63;
            --bg: #FAF6F0;
            --card-bg: #FFFFFF;
            --shadow: 0 8px 25px rgba(166, 123, 91, 0.15);
            --shadow-lg: 0 15px 35px rgba(166, 123, 91, 0.25);
            --radius: 16px;
            --radius-sm: 8px;
            --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
        }

        * { margin: 0; padding: 0; box-sizing: border-box; }

        body {
            font-family: 'Kanit', sans-serif;
            background: linear-gradient(135deg, var(--bg) 0%, #F5EFE6 100%);
            color: var(--text);
            line-height: 1.6;
            overflow-x: hidden;
```

1. HTML Structure (โครงสร้าง HTML)

- <head>: กำหนดข้อมูลเมตา เช่น การตั้งค่ารหัสตัวอักษร, ชื่อเว็บไซต์, การเชื่อมโยงฟอนต์, และการเชื่อมโยงไอคอนฟอนต์จาก FontAwesome
- <body>: ประกอบไปด้วยเนื้อหาหลักที่มีส่วนต่าง ๆ เช่น:

  โปรไฟล์ส่วนตัว: แสดงชื่อและข้อมูลเกี่ยวกับผู้ใช้

  Navigation: ปุ่มเพื่อเปลี่ยนส่วนที่แสดง (เกี่ยวกับฉัน, ตารางเรียน, งานอดิเรก)

  ข้อมูลส่วนตัว: แสดงข้อมูลเกี่ยวกับผู้ใช้และข้อมูลการศึกษา

  ตารางเรียน: แสดงภาพตัวอย่างตารางเรียนพร้อมปุ่ม "ดูแบบเต็ม"

  งานอดิเรก: แสดงงานอดิเรกด้วยภาพหลายภาพในรูปแบบกริด

2. CSS Styling (สไตล์ CSS)

- การตั้งค่าผลลัพธ์ของตัวแปรใน :root เช่น สีพื้นหลัง, สีตัวอักษร, และการตั้งค่ารูปร่าง เช่น --primary, --bg
- การใช้กริดและ Flexbox เพื่อจัดการการแสดงผลและตำแหน่งขององค์ประกอบในหน้า
- การกำหนดสไตล์ให้กับปุ่ม, ภาพ, และโครงสร้างต่าง ๆ เช่น การทำให้ปุ่มมีเงา, การทำให้ปุ่มเมื่อhover แล้วมีการเคลื่อนไหว

3. JavaScript Functions (ฟังก์ชัน JavaScript)

- **Loading Animation**: เมื่อโหลดหน้าเว็บเสร็จ ฟังก์ชัน window.addEventListener('load') จะซ่อนสปินเนอร์ที่แสดงขณะโหลด
- **showSection(id)**: ฟังก์ชันนี้จะเปลี่ยนหน้าแสดงข้อมูลตามปุ่มที่คลิก โดยจะซ่อนทุกๆ content-section และแสดงแค่ส่วนที่ผู้ใช้เลือก
- **viewSchedule()**: เปิด modal เพื่อแสดงตารางเรียนแบบเต็ม
- **closeSchedule()**: ปิด modal ที่แสดงตารางเรียนแบบเต็ม

4. Modal (หน้าต่างป๊อปอัพ)

- **schedule-modal**: เมื่อต้องการดูตารางเรียนแบบเต็ม จะมีหน้าต่างป๊อปอัพปรากฏขึ้น พร้อมปุ่มปิด

3.หน้าสาม Agi 64-67

```
agi.html > html > head > style > :root
<!DOCTYPE html>
<html lang="th">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Faculty og Agriculture - มหาวิทยาลัยนเรศวร</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />
    <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>
    <link href="https://fonts.googleapis.com/css2?family=Kanit:wght@300;400;500;600;700&display=swap" rel="s
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <style>
        :root {
            --primary: #2E8B57;
            --primary-dark: #1F6B45;
            --primary-light: #4CAF7D;
            --secondary: #FF6B35;
            --accent: #FFD166;
            --light: #F8F9FA;
            --dark: #2D3748;
            --text: #4A5568;
            --text-light: #718096;
            --bg: #F7FAFC;
            --card-bg: #FFFFFF;
            --success: #48BB78;
            --warning: #ED8936;
            --danger: #F56565;
            --gradient-primary: linear-gradient(135deg, #2E8B57 0%, #4CAF7D 100%);
            --gradient-secondary: linear-gradient(135deg, #FF6B35 0%, #FF8E53 100%);
            --gradient-accent: linear-gradient(135deg, #FFD166 0%, #FFE08A 100%);
            --shadow: 0 10px 25px rgba(0,0,0,0.08);
            --shadow-lg: 0 20px 40px rgba(0,0,0,0.12);
            --radius: 16px;
            --radius-sm: 8px;
            --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
        }
```

1. ส่วนติดต่อผู้ใช้ (UI)

- **Header** - แสดงชื่อคณะและข้อมูลผู้ใช้
- **Navigation** - ปุ่มเลือกชั้นปี (AGI64-67)
- **Statistics Cards** - สถิติรวมจำนวนนิสิต, หลักสูตร, ภาควิชา
- **Chart** - กราฟแสดงข้อมูลแบบ doughnut
- **Map** - แผนที่แสดงการกระจายตัว
- **Table** – ตารางรายชื่อนิสิต

2. ฟังก์ชันการทำงาน

- loadStudents() - ดึงข้อมูลจาก API
- updateChart() - อัพเดทกราฟ
- updateMap() - อัพเดทแผนที่
- searchTable() - ค้นหาในตาราง
- selectTable() - สลับระหว่างชั้นปี

3. ฟีเจอร์สำคัญ

- **Responsive Design** - ปรับตามหน้าจอ
- **Real-time Data** - โหลดข้อมูลจาก API
- **Interactive** - ค้นหา, เรียงข้อมูล, Pagination
- **Visualization** - กราฟและแผนที่

4. เทคโนโลยี

- **Frontend**: HTML, CSS, JavaScript
- **Libraries**: Chart.js, Leaflet (แผนที่), Font Awesome
- **Styling**: Glassmorphism, Modern Design

4.หน้าสี่ Phrae

```
indexcafe.html > html > body > script > overlays
<!DOCTYPE html>
<html lang="th">
<head>
<meta charset="utf-8">
<title>My Web GIS - Cafe Phrae</title>

<!-- Leaflet -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
 integrity="sha256-p4NxAoJBhIIN+hmNHrzRCf9tD/miZyoHS5obTRR9BMY="
 crossorigin=""/>
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
 integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1lvTlZBo="
 crossorigin=""></script>

<!-- jQuery สำหรับเรียก API -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<!-- Font Awesome สำหรับไอคอน -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>
/* เต็มหน้าจอ */
html, body {
  height: 100%;
  margin: 0;
  padding: 0;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background-color: #f5f1eb;
}

/* Header */
.header {
  height: 80px;
  display: flex;
  align-items: center;
  justify-content: center;
  padding: 0 30px;
```

การตั้งค่าแผนที่ (Leaflet):

- แผนที่ถูกสร้างขึ้นด้วย **Leaflet** ที่มีการตั้งค่าการแสดงผลเริ่มต้น (ตำแหน่ง [18.145, 100.141] และระดับซูม 13)
- เพิ่มแผนที่พื้นฐาน (Base Layers) 4 แบบ ได้แก่ Google Maps, OpenStreetMap, Esri Street, และ Esri Imagery

Layer ที่เป็น WMS:

- มีการเชื่อมต่อกับ **GeoServer** เพื่อดึงข้อมูล WMS เช่น ขอบเขตอำเภอ (phrae_amphoe), ถนนหลัก (phrae_road), และแหล่งท่องเที่ยว (phrae_point)

การแสดงข้อมูลคาเฟ่ (GeoJSON):

- ใช้ **fetch API** เพื่อดึงข้อมูล GeoJSON จากไฟล์ json_sql.php ที่ประกอบด้วยข้อมูลคาเฟ่
- แต่ละจุด (marker) ของคาเฟ่จะใช้ไอคอนเป็นอิโมจิ ☕ และเมื่อคลิกจะแสดงข้อมูลเช่น ชื่อ, ที่อยู่, เวลาเปิดทำการ

การเพิ่ม Layer บนแผนที่:

- สถานที่คาเฟ่จะถูกเพิ่มใน **cafeLayer** ซึ่งสามารถเปิด/ปิดได้ผ่าน control panel ของแผนที่
- มี **legend** (คำอธิบาย) และ **control panel** เพื่อควบคุมการแสดงผลของแต่ละ Layer

5.check_auth.php

```
api > check_auth.php
<?php
header('Content-Type: application/json');
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Methods: GET, POST, OPTIONS');
header('Access-Control-Allow-Headers: Content-Type');

// ไฟล์ config
require_once './config.php';

class AuthAPI {
    private $pdo;

    public function __construct() {
        $this->pdo = DatabaseConfig::getConnection();
    }

    public function checkAuth() {
        try {
            // ในเวอร์ชันจริงควรใช้ session หรือ JWT
            // ที่นี่เราจะใช้การตรวจสอบแบบง่ายๆ ด้วย cookie หรือ parameter

            $user_id = $_COOKIE['user_id'] ?? $_GET['user_id'] ?? null;

            if ($user_id) {
                // ตรวจสอบว่ามีผู้ใช้ในระบบหรือไม่
                $stmt = $this->pdo->prepare("
                    SELECT s_id as id, s_name as name, หลักสูตร as program, ภาควิชา as department
                    FROM agi64
                    WHERE s_id = :user_id
                    LIMIT 1
                ");
                $stmt->execute([':user_id' => $user_id]);
                $user = $stmt->fetch();

                if ($user) {
                    return [
                        'loggedin' => true
```

Header การตั้งค่า API:

- header('Content-Type: application/json'): กำหนดว่าเนื้อหาที่ส่งกลับจะเป็น JSON
- header('Access-Control-Allow-Origin: *'): เปิดให้เรียก API จากทุกโดเมน
- header('Access-Control-Allow-Methods: GET, POST, OPTIONS'): กำหนดวิธีการที่อนุญาตให้เรียก API
- header('Access-Control-Allow-Headers: Content-Type'): อนุญาตให้ตั้งค่าหัวข้อ Content-Type

การเชื่อมต่อฐานข้อมูล:

- require_once './config.php': โหลดการตั้งค่าการเชื่อมต่อฐานข้อมูลจากไฟล์ config.php
- DatabaseConfig::getConnection(): ใช้ฟังก์ชันเพื่อเชื่อมต่อกับฐานข้อมูล

คลาส AuthAPI:

- **__construct()**: สร้างการเชื่อมต่อกับฐานข้อมูล
- **checkAuth()**: ฟังก์ชันนี้จะตรวจสอบว่า user ได้ล็อกอินหรือไม่
    - ถ้ามี user_id จาก **cookie** หรือ **parameter** ใน URL ($_GET['user_id']), จะใช้ user_id นี้เพื่อตรวจสอบในฐานข้อมูล
    - ถ้าพบข้อมูลผู้ใช้ในฐานข้อมูล (ตาม s_id), จะส่งข้อมูลผู้ใช้กลับ (ID, ชื่อ, หลักสูตร, ภาควิชา)
    - ถ้าไม่พบข้อมูล หรือไม่มี user_id จะตอบกลับว่า "ไม่ได้ล็อกอิน"

ผลลัพธ์:

- ถ้าผู้ใช้ล็อกอินสำเร็จ: ส่งผลลัพธ์ในรูปแบบ loggedin: true พร้อมข้อมูลผู้ใช้
- ถ้าไม่ล็อกอิน: ส่งผลลัพธ์ loggedin: false พร้อมข้อความ "ไม่ได้ล็อกอิน"
- ถ้ามีข้อผิดพลาดจากฐานข้อมูล: ส่งผลลัพธ์ loggedin: false พร้อมข้อความ "Database error"

7.insert.php

insert.php

```php
<?php
$host = "localhost";
$port = "5432";
$dbname = "plk_backup";
$user = "postgres";
$password = "postgres";

$db = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$db) { die("ไม่สามารถเชื่อมต่อฐานข้อมูลได้"); }

$lat  = $_POST['lat']  ?? null;
$lng  = $_POST['lng']  ?? null;
$name = $_POST['name'] ?? null;

if ($lat && $lng && $name) {
    $insert_sql = "
    INSERT INTO points(geom, name)
    VALUES (ST_SetSRID(ST_Point($lng, $lat), 4326), '$name');
    ";
    $res = pg_query($db, $insert_sql);

    if (!$res) { die("Insert failed: " . pg_last_error($db)); }
}

header('Content-Type: application/json');
echo json_encode(['status'=>'success']);
?>
```

การเชื่อมต่อฐานข้อมูล: เชื่อมต่อกับฐานข้อมูล PostgreSQL โดยใช้ข้อมูลที่ระบุ เช่น host, port, dbname, user, password ถ้าไม่สามารถเชื่อมต่อจะหยุดการทำงานและแสดงข้อความ "ไม่สามารถเชื่อมต่อฐานข้อมูลได้"

รับข้อมูลจาก $_POST: รับพิกัดละติจูด (lat), ลองจิจูด (lng) และชื่อ (name) จากฟอร์มที่ส่งมา (ถ้าไม่มีข้อมูลจะตั้งเป็น null)

ตรวจสอบข้อมูล: ถ้ามีค่าทั้ง lat, lng, และ name จะทำการแทรกข้อมูลลงในตาราง points โดยใช้คำสั่ง SQL พร้อมสร้าง Geometry Point ด้วย ST_SetSRID และ ST_Point ที่ใช้ SRID 4326 (ระบบพิกัด WGS 84)

แทรกข้อมูล: ใช้ pg_query() เพื่อรันคำสั่ง SQL และแทรกข้อมูลเข้าไปในฐานข้อมูล ถ้าไม่สำเร็จจะแสดงข้อผิดพลาด

ส่งผลลัพธ์: ส่งคำตอบกลับในรูปแบบ JSON โดยระบุว่า "status": "success"

8.json_sql.php

```php
<?php
include('conn.php');

// ==========================================
// json_cafe_phrae.php : ดึงข้อมูลทั้งหมดเป็น GeoJSON
// ==========================================

$sql = "
    SELECT
        id,
        Name,
        Address,
        Latitude,
        Longitude,
        Opening_hours,
        Open_Daily,
        ST_AsGeoJSON(geom) AS geojson
    FROM cafe_phrae
    ORDER BY id DESC;
";

$result = pg_query($conn, $sql);
if (!$result) {
    die(json_encode([
        'status' => 'error',
        'message' => pg_last_error($conn)
    ]));
}

// 🔹 สร้าง FeatureCollection
$geojson = [
    'type' => 'FeatureCollection',
    'features' => []
];

// 🔹 วนลูปเพิ่มข้อมูลแต่ละแถวเป็น Feature
```

เชื่อมต่อฐานข้อมูล: รวมไฟล์ conn.php ซึ่งคาดว่าเป็นการเชื่อมต่อฐานข้อมูล PostgreSQL.

คำสั่ง SQL: ดึงข้อมูลจากตาราง cafe_phrae โดยเลือกคอลัมน์ต่างๆ เช่น id, name, address, latitude, longitude, opening_hours, และแปลงข้อมูล geometry (geom) ให้เป็น GeoJSON ด้วยฟังก์ชัน ST_AsGeoJSON.

ตรวจสอบผลลัพธ์: ถ้าไม่สามารถดึงข้อมูลได้ จะแสดงข้อความแสดงข้อผิดพลาดในรูปแบบ JSON.

สร้าง GeoJSON:

- สร้าง array ที่เก็บข้อมูลในรูปแบบ **GeoJSON** โดยใช้ FeatureCollection.
- วนลูปผลลัพธ์จากฐานข้อมูล และแปลงแต่ละแถวให้เป็น Feature โดยใช้ข้อมูลในคอลัมน์ต่างๆ

ส่งข้อมูลกลับ: ส่งข้อมูล GeoJSON ในรูปแบบ JSON พร้อมกับการตั้งค่า Content-Type เป็น application/json เพื่อให้ผู้ใช้สามารถรับข้อมูลในรูปแบบ JSON ได้

ปิดการเชื่อมต่อ: ปิดการเชื่อมต่อฐานข้อมูลหลังจากส่งข้อมูลเสร็จ