รายงานการพัฒนาแผนที่บนเว็บ

จัดทำโดย

นางสาวจารุนันต์ เย็นใจ

รหัสนิสิต66160500

เสนอ

รศ.ดร.สิทธิชัย ชูสำโรง

รายงานนี้เป็นส่วนหนึ่งของรายวิชาWeb GIS Development

การพัฒนาแผนที่บนเว็บ

ภาคเรียนที่1 ปีการศึกษา2568

มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาเรื่อง **“การพัฒนาเว็บไซต์แผนที่เชิงโต้ตอบเพื่อจัดการข้อมูลเชิงพื้นที่”** มีวัตถุประสงค์เพื่อศึกษา ออกแบบ และพัฒนาเว็บไซต์ที่สามารถแสดงผลและบริหารจัดการข้อมูลเชิงพื้นที่ได้อย่างมีประสิทธิภาพ โดยใช้เทคโนโลยีเว็บและระบบสารสนเทศภูมิศาสตร์ (Geographic Information System: GIS) มาประยุกต์ร่วมกัน เพื่อให้ผู้ใช้งานสามารถสืบค้น ดูข้อมูล และโต้ตอบกับแผนที่ได้แบบเรียลไทม์การพัฒนาเว็บไซต์ลักษณะนี้มีความสำคัญอย่างยิ่งในยุคปัจจุบัน เนื่องจากข้อมูลเชิงพื้นที่ถูกนำมาใช้ในการตัดสินใจด้านต่าง ๆ เช่น การวางผังเมือง การบริหารทรัพยากรธรรมชาติ การขนส่ง และการจัดการภัยพิบัติ การจัดทำระบบแผนที่เชิงโต้ตอบจึงช่วยให้หน่วยงานหรือผู้ใช้งานทั่วไปสามารถเข้าถึงและใช้ประโยชน์จากข้อมูลได้อย่างสะดวก รวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้นผู้จัดทำหวังว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อผู้ที่สนใจในการพัฒนาเว็บไซต์ด้าน GIS และสามารถนำความรู้ที่ได้ไปประยุกต์ใช้ในการสร้างสรรค์ระบบสารสนเทศเชิงพื้นที่ในอนาคตต่อไปได้อย่างเหมาะสม

จัดทำโดย

นางสาวจารุนันต์ เย็นใจ

บทนำ

ที่มาและความสำคัญของปัญหา

ในยุคปัจจุบัน เทคโนโลยีสารสนเทศมีบทบาทสำคัญต่อการจัดการข้อมูลในหลายด้าน โดยเฉพาะข้อมูลเชิงพื้นที่ (Spatial Data) ซึ่งเป็นข้อมูลที่มีความสัมพันธ์กับตำแหน่งทางภูมิศาสตร์ เช่น พิกัดที่ตั้งของสถานที่ ถนน อาคาร หรือทรัพยากรธรรมชาติ ข้อมูลประเภทนี้มีปริมาณมากและซับซ้อน การนำเสนอข้อมูลเชิงพื้นที่ในรูปแบบแผนที่เชิงโต้ตอบ (Interactive Map) จึงเป็นแนวทางที่ช่วยให้ผู้ใช้สามารถเข้าถึง วิเคราะห์ และทำความเข้าใจกับข้อมูลได้อย่างสะดวกและมีประสิทธิภาพมากยิ่งขึ้น

อย่างไรก็ตาม การจัดการข้อมูลเชิงพื้นที่ผ่านเว็บไซต์ยังคงมีข้อจำกัดในด้านความสามารถในการโต้ตอบ การอัปเดตข้อมูล และการนำเสนอผลลัพธ์ที่เข้าใจง่าย โครงการ **“การพัฒนาเว็บไซต์แผนที่เชิงโต้ตอบเพื่อจัดการข้อมูลเชิงพื้นที่”** จึงถูกจัดทำขึ้นเพื่อพัฒนาแพลตฟอร์มที่สามารถแสดงผลและบริหารข้อมูลเชิงพื้นที่ได้แบบเรียลไทม์ โดยผสานเทคโนโลยีเว็บและระบบสารสนเทศภูมิศาสตร์ (Geographic Information System: GIS) เข้าด้วยกัน เพื่อให้ผู้ใช้สามารถเพิ่ม แก้ไข หรือตรวจสอบข้อมูลได้ผ่านระบบออนไลน์

แอปพลิเคชั่นเกี่ยวกับอะไร

แอปพลิเคชันแผนที่แบบ Interactive สำหรับแสดงข้อมูล GeoJSON และข้อมูลภูมิศาสตร์ต่างๆ ของประเทศไทย รวมถึงข้อมูลสภาพอากาศ, คุณภาพอากาศ, UV Index และแผ่นดินไหวทั่วโลก

แอปพลิเคชันนี้เปรียบเสมือน Google Maps แบบพิเศษ ที่สามารถแสดงข้อมูลต่างๆ บนแผนที่ ตำแหน่งโรงพยาบาล ตำแหน่งนักเรียน เส้นแบ่งจังหวัดและอำเภอ สภาพอากาศแต่ละจังหวัด ดัชนีรังสียูวี (UV Index) คุณภาพอากาศ (PM2.5, PM10) แผ่นดินไหวทั่วโลก

การทำงานโดยรวม

1. เปิดเว็บไซต์ ⟶ หน้า login.html

↓

2. เข้าสู่ระบบด้วย username/password ⟶ ตรวจสอบกับ GeoServer

↓

3. เข้าสู่หน้าแผนที่หลัก ⟶ index.html

↓

4. กดปุ่มเปิด layer ต่างๆ (โรงพยาบาล, สภาพอากาศ ฯลฯ)

↓

5. ระบบดึงข้อมูล ⟶ จาก GeoServer หรือ API ภายนอก

↓

6. แสดงผลบนแผนที่ ⟶ เป็นจุด, วงกลม, หรือเส้นแบ่งเขต

↓

7. คลิกที่จุดบนแผนที่ ⟶ แสดงข้อมูลรายละเอียด

โครงสร้างไฟล์และคำอธิบาย

ไฟล์ HTML เป็นหลัก Index.html . ใช้ทำแผนที่หลัก ทำหน้าที่เป็นหน้าเว็บหลักที่แสดงแผนที่และเมนูต่างๆ

แบ่งหน้าจอเป็น 3 ส่วน คำอธิบาย (ซ้าย) แผนที่(กลาง) เมนูควบคุม(ขวา) มีปุ่มเปิด/ปิด layer ต่างๆ เช่น โรงพยาบาล, สภาพอากาศมีฟอร์มค้นหาโรงพยาบาลและนักเรียน โหลดไฟล์ JavaScript ทั้งหมดมาใช้งาน

Login.html หน้าเข้าสู่ระบบ

ทำหน้าที่ หน้าสำหรับกรอก username และ password

ตรวจสอบสิทธิ์กับ GeoServer ก่อนเข้าใช้งานระบบ ถ้าเข้าสู่ระบบสำเร็จ ⟶ ไปหน้า index.html

ถ้าล้มเหลว ⟶ แสดงข้อความแจ้งเตือน

ไฟล์CSS

Styles.css ไฟล์ตกแต่งหน้าตา ทำหน้าที่กำหนดสี, ขนาด, รูปแบบของปุ่ม, popup, tooltip ต่างๆ ตกแต่ง popup ที่แสดงข้อมูลเมื่อคลิกบนแผนที่ตกแต่ง tooltip ที่แสดงเมื่อชี้เมาส์กำหนดสีธีมมืด (Dark Mode) ปรับแต่ง scrollbar, ปุ่ม, และ icon

Config.js ไฟล์กำหนดค่าคงที่ ทำหน้าที่ เก็บค่าตั้งต้นที่ใช้ทั้งระบบ กำหนดเวลา cache (เก็บข้อมูลไว้ 5 นาที)

กำหนดภูมิภาคเริ่มต้น (ภาคกลาง) กำหนด URL ของ GeoServer กำหนดรายชื่ออำเภอในพิษณุโลก

Regions.js ข้อมูลภูมิภาคและจังหวัด ทำหน้าที่ เก็บรายชื่อจังหวัดทั้งหมดแบ่งตามภูมิภาค ภาคเหนือ: 9 จังหวัด ภาคกลาง: 22 จังหวัดภาคตะวันออกเฉียงเหนือ: 19 จังหวัดภาคตะวันออก: 7 จังหวัดภาคตะวันตก: 5 จังหวัด ภาคใต้: 14 จังหวัด ตัวอย่างเช่น เมื่อเลือกภาคเหนือ → ระบบจะแสดงข้อมูลเฉพาะ 9 จังหวัดในภาคเหนือ

Utils.js ฟังก์ชันช่วยเหลือทั่วไป ทำหน้าที่ เก็บฟังก์ชันเล็กๆ ที่ใช้บ่อยๆ

getWeatherDesc() แปลงรหัสสภาพอากาศเป็นภาษาไทย (เช่น 0 = แจ่มใส)

getUVRiskLevvel() คำนวณระดับความเสี่ยงจาก UV (ต่ำ/ปานกลาง/สูง/อันตราย)

escapeXml() ป้องกันข้อมูลที่ไม่ปลอดภัยเมื่อส่งไป GeoServer

formatThaiDateTime() แปลงวันที่เวลาเป็นรูปแบบไทย

ไฟล์จัดการหลักระบบ

Script.js ไฟล์หลักของทั้งระบบ (Entry Point) ทำหน้าที่เป็นศูนย์กลางควบคุมทุกอย่าง เหมือนหัวหน้าทีม

1.สร้างแผนที่ สร้างแผนที่ Leaflet ขึ้นมา

2.สร้าง Base Layers เตรียมแผนที่พื้นฐาน 4 แบบ (ถนน/ภูมิศาสตร์/ดาวเทียม/มืด)

3.สร้าง Layer Groups แยกชั้นข้อมูลต่างๆ (โรงพยาบาล, สภาพอากาศ, UV, คุณภาพอากาศ, แผ่นดินไหว, เส้นแบ่งจังหวัด/อำเภอ)

4. เริ่มต้น Managers สั่งให้ตัวจัดการแต่ละประเภทพร้อมทำงาน

5.ผูก Event Handlers เชื่อมปุ่มและฟอร์มกับฟังก์ชันต่างๆ

6. จัดการการเปิด/ปิด เมื่อกดปุ่ม → เปิด/ปิดข้อมูลบนแผนที่

เปรียบเทียบ เหมือนหัวหน้าโครงการที่คอยสั่งงานทีมงานแต่ละคน (Manager) ให้ทำงานของตัวเอง

Ui-manager.js ตัวจัดการส่วนแสดงผล ทำหน่าที่ดูแลส่วนที่แสดงบนหน้าเว็บ

แสดงรายการ layer ที่กำลังเปิดอยู่ (Active Layers) แสดง/ซ่อน ตัวเลือกภูมิภาค (Region Selector) เมื่อจำเป็น อัปเดตสถานะปุ่มต่างๆ ติดตามว่า layer ไหนเปิดอยู่บ้าง

cache-manager.js ตัวจัดการแคช ทำหน้าที่ เก็บข้อมูลที่ดึงมาแล้วไว้ชั่วคราว เพื่อไม่ต้องดึงใหม่ทุกครั้ง
เก็บข้อมูลไว้ 5 นาที ถ้ายังไม่หมดเวลา → ใช้ข้อมูลเก่า (เร็วกว่า) ถ้าหมดเวลาแล้ว → ดึงข้อมูลใหม่
ประโยชน์ ช่วยให้เว็บไซต์เร็วขึ้น และประหยัดการใช้อินเทอร์เน็ต

ไฟล์จัดการระบบเข้าสู่ระบบ

ไฟล์จัดการระบบเข้าสู่ระบบ auth.js ตัวจัดการการเข้าสู่ระบบ ทำหน้าที่ตรวจสอบว่าผู้ใช้มีสิทธิ์เข้าใช้ระบบหรือไม่รับ username และ password จากฟอร์ม ส่งไปตรวจสอบกับ GeoServer ถ้าถูกต้อง → บันทึกว่าเข้าสู่ระบบแล้ว ถ้าผิด → แสดงข้อความแจ้งเตือน
เปรียบเทียบ เหมือนเจ้าหน้าที่รักษาความปลอดภัยที่คอยตรวจบัตรประชาชนก่อนเข้าอาคาร
Geoserver-api.js ตัวติดต่อกับ GeoServer ทำหน้าที่เป็นคนกลางในการส่ง/รับข้อมูลกับ GeoServer
ดึงข้อมูลดึงรายการโรงพยาบาล/นักเรียนจาก GeoServer เพิ่มข้อมูลเพิ่มโรงพยาบาล/นักเรียนใหม่
แก้ไขข้อมูล แก้ไขข้อมูลที่มีอยู่ ลบข้อมูล ลบโรงพยาบาล/นักเรียน
สร้างคำสั่ง XML พิเศษที่ GeoServer เข้าใจได้ กรองข้อมูลตามที่ต้องการ (ค้นหาตามชื่อ/อำเภอ)
เปรียบเทียบ เหมือนนักแปลที่แปลภาษาระหว่างเว็บไซต์กับฐานข้อมูล GeoServer

ไฟล์จัดการข้อมูลแต่ละประเภท (Managers)

Hospital-manager.js ตัวจัดการข้อมูลโรงพยาบาล ทำหน้าที่ ดูแลทุกอย่างเกี่ยวกับโรงพยาบาลบนแผนที่
1.โหลดข้อมูล ดึงข้อมูลโรงพยาบาลจาก GeoServer
2.แสดงจุดบนแผนที่ สร้างจุดสีน้ำเงิน/เขียวแทนโรงพยาบาล
3.แสดง เมื่อคลิกจุด → แสดงชื่อ, ที่อยู่, อำเภอ
4.ค้นหา กรองโรงพยาบาลตามชื่อหรืออำเภอ
5.เพิ่ม แก้ไข ลบ จัดการข้อมูลผ่านฟอร์ม
Student-manager.js ตัวจัดการข้อมูลนักเรียน ทำหน้าที่ ดูแลข้อมูลนักเรียนบนแผนที่
1.โหลดข้อมูลดึงข้อมูลนักเรียนจาก GeoServer
2.แสดงไอคอนสร้างไอคอนคนสีน้ำเงิน (👤) แทนนักเรียน
3.แสดง Popup เมื่อคลิก → แสดงชื่อ, รหัสนักเรียน, ระดับชั้น, โรงเรียน
4.ค้นหากรองนักเรียนตามชื่อหรือรหัส
5.เพิ่ม แก้ไข ลบจัดการข้อมูลผ่านฟอร์ม
Province-manager.js ตัวจัดการเส้นแบ่งจังหวัด ทำหน้าที่แสดงเส้นแบ่งเขตจังหวัดบนแผนที่
1.โหลดข้อมูลอ่านไฟล์ provinces.geojson
2.วาดเส้นแบ่ง วาดเส้นสีส้มทอง (#f59e0b) รอบจังหวัด
3.แสดง Tooltip เมื่อชี้เมาส์ → แสดงชื่อจังหวัด (ไทย/อังกฤษ) และภูมิภาค
4.ไฮไลต์ เมื่อชี้เมาส์ → เส้นหนาขึ้น, สีชัดขึ้น

รูปแบบเส้นสีทอง น้ำหนัก 3px แบบโปร่งใส

District-manager.js ตัวจัดการเส้นแบ่งอำเภอ ทำหน้าที่แสดงเส้นแบ่งเขตอำเภอบนแผนที่

1.โหลดข้อมูลอ่านไฟล์ districts.geojson

2.วาดเส้นแบ่งวาดเส้นสีม่วง (#6366f1) รอบอำเภอ

3.แสดง Tooltip เมื่อชี้เมาส์ → แสดงชื่ออำเภอและจังหวัด

4.ไฮไลต์ เมื่อชี้เมาส์ → เส้นหนาขึ้น, สีชัดขึ้น

รูปแบบ เส้นสีม่วง น้ำหนัก 2px แบบโปร่งใส

ไฟล์จัดการข้อมูลจากAPI ภายนอก

Weather-manager.js-ตัวจัดการสภาพอากาศ ทำหน้าที่แสดงสภาพอากาศปัจจุบันของแต่ละจังหวัด

1.ดึงข้อมูลเรียก API จาก Open-Meteo (ฟรี)

2.แสดงผลสร้างไอคอนแสดงอุณหภูมิและสภาพอากาศ

3.Popupรายละเอียด คลิกแล้วเห็น

อุณหภูมิ

ความชื้น

ความเร็วลม

โอกาสฝนตก

กราฟพยากรณ์อากาศ 24 ชั่วโมง

4.เลือกภูมิภาค สามารถเลือกดูเฉพาะภาคที่สนใจ

ข้อมูลที่แสดง อุณหภูมิ, ความชื้น, ลม, ฝน, สภาพอากาศ (แจ่มใส/มีเมฆ/ฝนตก ฯลฯ)

uv-manager.js ตัวจัดการดัชนีรังสี ทำหน้าที่ แสดงระดับรังสี UV ของแต่ละจังหวัด

1. ดึงข้อมูล เรียก API จาก Open-Meteo

2. คำนวณระดับ แบ่งเป็น 5 ระดับ

สีเขียว ต่ำ (0-2.9): ปลอดภัย

สีแหลือง ปานกลาง (3-5.9): ควรใช้ครีมกันแดด

สีส้ม สูง (6-7.9): ใช้ครีมกันแดดและหมวก

สีแดง สูงมาก (8-10.9): หลีกเลี่ยงแดดเที่ยง

🔴🔴 อันตราย (11+): อันตราย! หลีกเลี่ยงแดดทั้งวัน

3. แสดงวงกลม สีและขนาดตามระดับความเสี่ยง

4. คำแนะนำ ให้คำแนะนำวิธีป้องกัน

air-quality-manager.js ตัวจัดการคุณภาพอากาศ ทำหน้าที่แสดงคุณภาพอากาศของแต่ละจังหวัด

1. ดึงข้อมูล เรียก API จาก Open-Meteo

2.แสดงค่า PM2.5, PM10, US AQI (Air Quality Index)

3. แบ่งระดับ ตามมาตรฐาน US AQI

สีเขียว ดี (0-50): อากาศดี

สีเหลือง ปานกลาง (51-100): ยอมรับได้

สีส้ม ไม่ดีต่อกลุ่มเสี่ยง (101-150)

สีแดง ไม่ดี (151-200)

สีม่วง อันตราย (201+): ควรอยู่ในบ้าน

4. แสดงวงกลม สีและขนาดตามความรุนแรง

earthquake-manager.js ตัวจัดการแผ่นดินไหว ทำหน้าที่แสดงแผ่นดินไหวทั่วโลก 7 วันย้อนหลัง

1. ดึงข้อมูลเรียก API จาก USGS (สหรัฐอเมริกา)

2.แสดงผลสร้างวงกลมแสดงจุดที่เกิดแผ่นดินไหว

3.ข้อมูล ขนาด (Magnitude), ความลึก, เวลา, สถานที่

4. สีตามความลึก

น้ำเงิน: ตื้น (0-70 กม.)

ส้ม: ปานกลาง (70-300 กม.)

แดง: ลึก (300+ กม.)

5. ขนาดวงกลม ใหญ่ขึ้นตามความรุนแรง

6. แจ้งเตือน มีสัญลักษณ์พิเศษถ้ามี Tsunami Warning

ไฟล์ข้อมูล GeoJSON

provinces.geojson ข้อมูลเส้นแบ่งจังหวัด เก็บพิกัดเส้นแบ่งเขตจังหวัดทั้ง 77 จังหวัด

districts.geojson ข้อมูลเส้นแบ่งอำเภอเก็บพิกัดเส้นแบ่งเขตอำเภอทั่วประเทศ

hospitals.geojson ข้อมูลโรงพยาบาลเก็บตำแหน่งและรายละเอียดโรงพยาบาล

thailand-provinces.js ข้อมูลพิกัดศูนย์กลางจังหวัดเก็บพิกัด latitude/longitude ของศูนย์กลางจังหวัดทั้ง 77 จังหวัด

Flow การทำงานโดยละเอียด

script.jsไฟล์หลักของแอปพลิเคชัน (Main Entry Point)

สร้างและตั้งค่าแผนที่ Leaflet จัดการ base layers (แผนที่ถนน, ภูมิศาสตร์, ดาวเทียม) สร้าง layer groups สำหรับข้อมูลต่างๆ (โรงพยาบาล, สภาพอากาศ, UV, คุณภาพอากาศ, แผ่นดินไหว) เริ่มต้น managers ทั้งหมด และเชื่อมโยงกับ UI จัดการ event handlers สำหรับฟอร์มและปุ่มต่างๆ จัดการการสลับ layer และ region selector

js/config.jsไฟล์กำหนดค่าคงที่ (Configuration)

CACHE_DURATION`: ระยะเวลาเก็บ cache (15 นาที)

DEFAULT_REGION`: ภูมิภาคเริ่มต้น (ภาคกลาง)

js/auth.js จัดการระบบ Authentication กับ GeoServer
AuthManager class จัดการการเข้าสู่ระบบและตรวจสอบสิทธิ์
login(username, password) เข้าสู่ระบบผ่าน Spring Security logout(): ออกจากระบบ
showLoginRequired() แสดงแจ้งเตือนให้เข้าสู่ระบบใช้ sessionStorage เก็บข้อมูลผู้ใช้
รองรับ Spring Security form authentication
js/geoserver-api.js API สำหรับติดต่อกับ GeoServer (WFS/WFS-T) GeoServerAPI class จัดการคำ
ขอ WFS และ WFS-T
fetchHospitals(filter) ดึงข้อมูลโรงพยาบาลจาก WFS พร้อมกรองข้อมูล
insertHospital(properties, coordinates) เพิ่มโรงพยาบาลใหม่ผ่าน WFS-T
updateHospital(fid, properties, coordinates) แก้ไขข้อมูลโรงพยาบาล
deleteHospital(fid) ลบโรงพยาบาล
สร้าง XML สำหรับ WFS-T transactions (Insert, Update, Delete)
จัดการ CQL filter สำหรับค้นหาข้อมูล Escape XML และ CQL อย่างปลอดภัย
js/hospital-manager.js จัดการข้อมูลและการแสดงผลโรงพยาบาล
HospitalManager class: จัดการ layer โรงพยาบาล
loadHospitals(filter) โหลดและแสดงโรงพยาบาลจาก GeoServer
addHospital(properties, coordinates) เพิ่มโรงพยาบาลใหม่
updateHospital(fid, properties, coordinates) แก้ไขข้อมูล
deleteHospital(fid) ลบโรงพยาบาล
สร้าง marker และ popup สำหรับแต่ละโรงพยาบาล จัดการฟอร์มเพิ่ม/แก้ไขข้อมูล รองรับการกรองตามชื่อและ
อำเภอ
js/weather-manager.jsจัดการข้อมูลสภาพอากาศปัจจุบัน
WeatherManager class: ดึงและแสดงข้อมูลสภาพอากาศ
loadWeatherForecast() โหลดข้อมูลสภาพอากาศจาก Open-Meteo API
getWeatherDesc(code) แปลง weather code เป็นคำอธิบาย
setRegion(region เปลี่ยนภูมิภาคที่แสดง
แสดงอุณหภูมิ, ความชื้น, ความเร็วลม, โอกาสฝนตก รองรับการพยากรณ์อากาศรายชั่วโมง ใช้ cache เพื่อลดการ
เรียก API
js/uv-manager.js
จัดการข้อมูล UV Index (ดัชนีรังสียูวี) UVManager class จัดการข้อมูล UV Index
loadUVIndex() โหลดข้อมูล UV Index จาก Open-Meteo API
getUVRiskLevel(uv) คำนวณระดับความเสี่ยงจาก UV Index
setRegion(region)เปลี่ยนภูมิภาค แบ่งระดับความเสี่ยง: ต่ำ, ปานกลาง, สูง, สูงมาก, อันตราย
แสดงคำแนะนำการป้องกันแสงแดด ใช้สีแสดงระดับความเสี่ยงที่ต่างกัน

js/air-quality-manager.js จัดการข้อมูลคุณภาพอากาศ
AirQualityManager class จัดการข้อมูลคุณภาพอากาศ
loadAirQuality() โหลดข้อมูล PM2.5, PM10, AQI จาก Open-Meteo API
setRegion(region) เปลี่ยนภูมิภาค
แสดงค่า US AQI (Air Quality Index) แสดงค่า PM2.5 และ PM10 จัดระดับตามมาตรฐาน US AQI: ดี, ปานกลาง, ไม่ดี, อันตราย ใช้สีและขนาดวงกลมแสดงระดับมลพิษ
js/earthquake-manager.js จัดการข้อมูลแผ่นดินไหว
EarthquakeManager class จัดการข้อมูลแผ่นดินไหวทั่วโลก
loadEarthquakes() โหลดข้อมูลแผ่นดินไหว 7 วันย้อนหลังจาก USGS
renderEarthquakes(data) แสดงผลข้อมูลแผ่นดินไหวบนแผนที่
แสดงขนาด (magnitude) และความลึก (depth) แสดงเวลาและสถานที่เกิดแผ่นดินไหว แจ้งเตือนหากมี tsunami warningใช้สีแยกตามความลึก และขนาดวงกลมตามความรุนแรง
js/regions.js กำหนดข้อมูลภูมิภาคและจังหวัดREGIONS object เก็บรายชื่อจังหวัดแยกตามภูมิภาค
north ภาคเหนือ (9 จังหวัด) centralภาคกลาง (22 จังหวัด)
northeas ภาคตะวันออกเฉียงเหนือ (19 จังหวัด)
east ภาคตะวันออก (7 จังหวัด)
west ภาคตะวันตก (5 จังหวัด)
south ภาคใต้ (14 จังหวัด)
getProvincesByRegion(region) ฟังก์ชันดึงรายชื่อจังหวัดตามภูมิภาค
js/ui-manager.js จัดการส่วน UI ต่างๆ ของแอปพลิเคชันUIManager` classจัดการ UI elements
setLayerState(layerName, isActive) อัปเดตสถานะ layer ที่เปิดใช้งาน
updateActiveLayersDisplay() แสดงรายการ layer ที่กำลังเปิดอยู่
แสดง/ซ่อน region selector ตามความจำเป็น จัดการการแสดงผล active layers panel ติดตามสถานะของ layers ทั้งหมด
js/cache-manager.js จัดการ cache ข้อมูล API
CacheManager`class จัดการ cache เพื่อลดการเรียก API isValid(cacheType, region ตรวจสอบว่า cache ยังใช้ได้หรือไม่ get(cacheType, region) ดึงข้อมูลจาก cache set(cacheType, region, data) บันทึกข้อมูลลง cache clear(cacheType) ล้าง cache getStatus() ดูสถานะ cache กำหนดเวลา cache 15 นาที (CACHE_DURATION) รองรับ cache แยกตามภูมิภาคช่วยเพิ่มประสิทธิภาพและลดการใช้ bandwidth
js/utils.js ฟังก์ชันช่วยเหลือทั่วไป (Utility Functions)
escapeXml(unsafe) Escape อักขระพิเศษใน XML
escapeCQL(s) Escape string สำหรับ CQL query
getWeatherDesc(code) แปลง WMO weather code เป็นคำอธิบาย

getUVRiskLevel(uv) คำนวณระดับความเสี่ยง UV

formatThaiDateTime(dateString) แปลงวันที่เป็นรูปแบบไทย

formatNumber(num) Format ตัวเลขให้มีจุลภาค

API ที่ใช้งาน

1. GeoServer WFS/WFS-T - จัดการข้อมูลโรงพยาบาล
2. Open-Meteo API - ข้อมูลสภาพอากาศ, UV Index, คุณภาพอากาศ
3.USGS Earthquake API - ข้อมูลแผ่นดินไหวทั่วโลก

Features

แสดงแผนที่โรงพยาบาลจาก GeoServer

เพิ่ม/แก้ไข/ลบข้อมูลโรงพยาบาลผ่าน WFS-T

แสดงสภาพอากาศปัจจุบันและพยากรณ์รายชั่วโมง

แสดงค่า UV Index พร้อมคำแนะนำ

แสดงคุณภาพอากาศ (PM2.5, PM10, AQI)

แสดงแผ่นดินไหว 7 วันย้อนหลังทั่วโลก

ระบบ cache ลดการเรียก API

ลือกดูข้อมูลตามภูมิภาค

ระบบ authentication กับ GeoServer

การใช้งาน

1. เปิดไฟล์ `index.html` ในเบราว์เซอร์
2. เลือก layer ที่ต้องการแสดงจากปุ่มทางขวาบนของแผนที่
3. เลือกภูมิภาคที่ต้องการดูข้อมูล (สำหรับ layer ที่รองรับ)
4. คลิกที่ marker เพื่อดูรายละเอียด
5. ใช้ฟอร์มทางขวาเพื่อเพิ่มโรงพยาบาลใหม่ (ต้อง login ก่อน)

หมายเหตุ

ข้อมูลจาก API ถูก cache ไว้ 15 นาที

การเพิ่ม/แก้ไข/ลบโรงพยาบาลต้อง login เข้าระบบ GeoServer ก่อน

แผ่นดินไหวแสดงข้อมูล 7 วันย้อนหลังจาก USGS