

รายงาน

พัฒนาเว็บไซต์แผนที่เชิงโต้ตอบเพื่อจัดการข้อมูลเชิงพื้นที่

จัดทำโดย

ชื่อ นางสาวชนกนันท์ ศรีชัยกุล

รหัสนิสิต 66260941

รายงานนี้เป็นส่วนหนึ่งของรายวิชา

web gis Development 104325-3

สาขาภูมิศาสตร์ คณะเกษตรศาสตร์และทรัพยากรธรรมชาติ

มหาวิทยาลัยนเรศวร ภาคเรียนที่ ปีการศึกษา 2568

คำนำ

การจัดการข้อมูลเชิงพื้นที่ถือเป็นองค์ประกอบสำคัญในการวางแผนและตัดสินใจในหลายด้าน เช่น การบริหารจัดการน้ำ การวางผังเมือง และการจัดการสาธารณูปโภค เพื่อให้การเข้าถึงและวิเคราะห์ข้อมูล เป็นไปอย่างสะดวกและแม่นยำ โครงการนี้จึงมุ่งพัฒนา **เว็บไซต์แผนที่เชิงโต้ตอบ (Interactive Web GIS)**

เว็บไซต์นี้สร้างด้วย **Leaflet.js, PostgreSQL/PostGIS, GeoServer และ PHP** โดยสามารถแสดงข้อมูล หลายชั้น เช่น ขอบเขตจังหวัด ถนน ร้านซักผ้า จุด GCP ข้อมูลฝนและระดับน้ำจาก API แบบเรียลไทม์ พร้อม ฟังก์ชัน **เพิ่ม แก้ไข ลบ และค้นหาจุด** บนแผนที่

ในการพัฒนาและจัดการข้อมูล มีการใช้เครื่องมือต่าง ๆ ได้แก่

- **pgAdmin:** จัดการฐานข้อมูล PostgreSQL/PostGIS และตรวจสอบพิกัดเชิงพื้นที่
- **QGIS:** ตรวจสอบและเตรียมข้อมูลเชิงพื้นที่ (GeoJSON, Shapefile)
- **GeoServer:** ให้บริการข้อมูล WMS/WFS เพื่อเชื่อมโยง GIS กับเว็บ
- **Visual Studio Code (VS Code):** เขียนและแก้ไขโค้ด HTML, JavaScript และ PHP

เว็บไซต์ประกอบด้วย 3 หน้าหลัก ได้แก่ หน้าแรกแสดงแผนที่และเลเยอร์ข้อมูล, หน้าเพิ่มจุดจัดการ CRUD ของจุด, และหน้าแนะนำตัวผู้พัฒนา ช่วยให้ผู้ใช้สามารถเข้าถึง วิเคราะห์ และจัดการข้อมูลเชิงพื้นที่ได้สะดวก และมีประสิทธิภาพ

บทที่ 1 วัตถุประสงค์และขอบเขตของโครงการ

1.1 วัตถุประสงค์

1. เพื่อพัฒนาเว็บไซต์แผนที่เชิงโต้ตอบ (Interactive Web GIS) ที่สามารถแสดงข้อมูลเชิงพื้นที่หลายชั้นได้
2. เพื่อให้ผู้ใช้สามารถเพิ่ม แก้ไข ลบ และค้นหาข้อมูลจุดบนแผนที่ได้อย่างสะดวก
3. เพื่อฝึกทักษะการใช้ PostgreSQL/PostGIS และเครื่องมือ GIS เช่น pgAdmin, QGIS และ GeoServer
4. เพื่อสร้างระบบที่ผู้ใช้สามารถเข้าถึง วิเคราะห์ และจัดการข้อมูลเชิงพื้นที่ได้แบบเรียลไทม์
5. เพื่อผสานเทคโนโลยีเว็บกับข้อมูล GIS ให้ทำงานครบวงจร

1.2 ขอบเขตของโครงการ

โครงการนี้ครอบคลุมการพัฒนาเว็บไซต์แผนที่เชิงโต้ตอบ ที่สามารถแสดงและจัดการข้อมูลเชิงพื้นที่หลายประเภท ครอบคลุมพื้นที่ทั่วประเทศ โดยข้อมูลนำเข้าผ่านซอฟต์แวร์ Open Source ได้แก่ QGIS, GeoServer, PostgreSQL/PostGIS และ pgAdmin

ผู้ใช้สามารถเปิด/ปิดเลเยอร์ เพิ่ม แก้ไข ลบ และค้นหาข้อมูลจุดได้จากหน้าเว็บโดยตรง เพื่อให้การบริหารจัดการข้อมูลเชิงพื้นที่สะดวกและรวดเร็ว

บทที่ 2 เทคโนโลยีและเครื่องมือที่ใช้

ในการพัฒนาเว็บไซต์แผนที่เชิงโต้ตอบ (Interactive Web GIS) สำหรับโครงการนี้ ได้มีการใช้เทคโนโลยีและเครื่องมือหลายประเภท เพื่อให้การจัดการข้อมูลเชิงพื้นที่และการพัฒนาเว็บเป็นไปอย่างครบวงจร ดังนี้

1. Leaflet.js

เป็นไลบรารี JavaScript สำหรับสร้างแผนที่แบบโต้ตอบบนเว็บ ช่วยให้ผู้ใช้สามารถซูม เลื่อน และแสดงเลเยอร์ข้อมูล GIS หลายชั้น รวมถึงสามารถเพิ่มจุด (Marker), เส้น (Polyline) และพื้นที่ (Polygon) บนแผนที่ได้

2. PostgreSQL / PostGIS

PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ส่วน PostGIS เป็นส่วนขยายที่ช่วยให้สามารถเก็บและจัดการข้อมูลเชิงพื้นที่ (Spatial Data) เช่น จุด, เส้น, และพอลิโกน ได้ ทำให้สามารถทำงานกับพิกัดเชิงภูมิศาสตร์อย่างแม่นยำ

3. pgAdmin

เครื่องมือสำหรับจัดการฐานข้อมูล PostgreSQL/PostGIS ผ่านอินเทอร์เฟซกราฟิก ใช้สำหรับตรวจสอบโครงสร้างฐานข้อมูล สร้างตาราง จัดการข้อมูลเชิงพื้นที่ และทดสอบคำสั่ง SQL

4. QGIS

เป็นซอฟต์แวร์ GIS แบบ Open Source ใช้สำหรับตรวจสอบ แก้ไข และเตรียมข้อมูลเชิงพื้นที่ เช่น GeoJSON หรือ Shapefile ก่อนนำเข้าสู่ฐานข้อมูลหรือ GeoServer

5. GeoServer

ซอฟต์แวร์ Open Source สำหรับให้บริการข้อมูล GIS ผ่านโพรโตคอลมาตรฐาน เช่น WMS (Web Map Service) และ WFS (Web Feature Service) ช่วยให้เว็บไซต์สามารถดึงข้อมูลแผนที่จากเซิร์ฟเวอร์และแสดงผลบนเว็บแบบเรียลไทม์

6. PHP

ใช้สร้าง Backend ของเว็บไซต์ สำหรับเชื่อมต่อกับฐานข้อมูล, ดึงและบันทึกข้อมูลจุด (CRUD), และทำหน้าที่เป็น Proxy เพื่อเรียกใช้ข้อมูลจาก GeoServer WFS และ API

7. Visual Studio Code (VS Code)

เป็นโปรแกรมแก้ไขโค้ดหลัก ใช้ในการพัฒนา HTML, CSS, JavaScript และ PHP สำหรับเว็บไซต์ ทำให้สามารถจัดการโค้ดและตรวจสอบข้อผิดพลาดได้สะดวก

บทที่ 3 ขั้นตอนการพัฒนาเว็บไซต์

3.1 การเตรียมสภาพแวดล้อม

ติดตั้งโปรแกรมพื้นฐาน ได้แก่

- Java JDK (ใช้รัน GeoServer)
- PostgreSQL + PostGIS (จัดเก็บข้อมูลเชิงพื้นที่)
- GeoServer (ให้บริการข้อมูล WMS/WFS)
- MS4W หรือ Apache Server (โฮสต์เว็บไซต์และไฟล์ JavaScript)
- QGIS (ตรวจสอบและแปลงข้อมูลก่อนนำเข้า GeoServer)

3.2 การสร้างฐานข้อมูล

1. เปิดโปรแกรม pgAdmin
2. สร้าง Database เช่น webgis_db
3. สร้างตารางข้อมูล เช่น laundry, rain_station, province_boundary
4. กำหนด Spatial Column (geom) และพิกัด EPSG:4326

3.3 การตั้งค่า GeoServer

1. สร้าง Workspace และ Store
2. เชื่อมต่อกับฐานข้อมูล PostgreSQL/PostGIS
3. Publish ชั้นข้อมูล (Layer)
4. ตรวจสอบผลใน Layer Preview (ผ่าน OpenLayers)

3.4 การพัฒนาเว็บด้วย Leaflet.js

ใช้ JavaScript เรียกข้อมูลจาก GeoServer (WMS/WFS) และแสดงผลบนแผนที่ เช่น

```
js 🗒 คัดลอกโค้ด  
  
var province = L.tileLayer.wms("http://localhost:8080/geoserver/webgis/wms", {  
  layers: "webgis:province_boundary",  
  format: "image/png",  
  transparent: true  
});  
province.addTo(map);
```

เพิ่มความสามารถในการ CRUD จุดผ่าน PHP API และจัดการเลเยอร์ด้วย L.control.layers()

บทที่ 4 ผลการดำเนินงาน

4.1 ภาพรวมการทำงานของเว็บไซต์

เว็บไซต์แผนที่เชิงโต้ตอบที่พัฒนาในโครงการนี้ประกอบด้วย 3 หน้าเว็บหลัก ได้แก่ index.html, addPoint.html, และ aboutMe.html ซึ่งทำงานเชื่อมโยงกันผ่านระบบฐานข้อมูล PostgreSQL/PostGIS และ GeoServer โดยใช้ PHP เป็นตัวกลางในการเชื่อมต่อข้อมูล

ระบบสามารถดึงข้อมูลจากหลายแหล่ง เช่น GeoServer (WMS/WFS), GeoJSON, และ API ภายนอก เช่น Thaiwater.net เพื่อแสดงข้อมูลฝนและระดับน้ำแบบเรียลไทม์

4.2 หน้าเว็บหลัก (index.html)

หน้าเว็บหลักเป็นส่วนสำคัญที่ใช้ในการแสดงผลข้อมูลเชิงพื้นที่ โดยเชื่อมต่อข้อมูลจากหลายแหล่ง ดังนี้

ข้อมูลที่แสดงในหน้าเว็บหลัก

1. ข้อมูลขอบเขตประเทศไทย — ใช้ไฟล์ GeoJSON แสดงขอบเขตประเทศ
2. ข้อมูลถนนในจังหวัดอุดรดิตถ์ — ดึงจาก GeoServer ผ่านบริการ WMS
3. ข้อมูลขอบเขตอำเภอในจังหวัดอุดรดิตถ์และพิษณุโลก — ดึงจาก GeoServer เช่นกัน
4. ข้อมูลจุดควบคุมภาคพื้นดิน (GCP Points) — แสดงเป็นจุด (Marker) บนแผนที่
5. ข้อมูลฝนและระดับน้ำ — ดึงจาก API ของเว็บไซต์ Thaiwater.net โดยใช้ JavaScript ประมวลผลและอัปเดตแบบเรียลไทม์
6. ข้อมูลร้านซักผ้ารอบมหาวิทยาลัยนเรศวร — เพิ่มเป็น GeoJSON ภายนอก และซ้อนเป็นอีกเลเยอร์ในแผนที่

ฟังก์ชันในหน้า index.html

- Layer Control: เปิด-ปิดเลเยอร์ข้อมูลได้ตามต้องการ
- Base Map Switching: เปลี่ยนพื้นหลังแผนที่ได้ เช่น
 - OpenStreetMap
 - Google Satellite
 - Terrain Map
- Popup ข้อมูล: เมื่อคลิกที่ Marker จะแสดงชื่อ จุด หรือข้อมูลจากฐานข้อมูล
- แสดงพิกัด: แสดงค่าพิกัดละติจูด-ลองจิจูดของตำแหน่งที่คลิก

สรุปการทำงาน

หน้าเว็บหลักจึงเป็นศูนย์กลางของระบบในการรวบรวมข้อมูลจากหลายแหล่ง ทั้ง GeoServer, API, และฐานข้อมูล PostGIS เพื่อให้ผู้ใช้สามารถดูข้อมูลแบบโต้ตอบได้ในหน้าเดียว

4.3 หน้าเพิ่มจุดข้อมูล (addPoint.html)

หน้าเว็บนี้ออกแบบมาเพื่อให้ผู้ใช้สามารถ **เพิ่ม แก้ไข ลบ และค้นหาข้อมูลจุด (CRUD)** ในฐานข้อมูล PostGIS ได้โดยตรง

ขั้นตอนการใช้งาน

1. **เพิ่มจุด:**
 - ผู้ใช้ดับเบิลคลิก (Double-click) บนแผนที่
 - ระบบจะแสดงแบบฟอร์ม Modal สำหรับกรอกชื่อและประเภทข้อมูล
 - เมื่อกด “Submit” ระบบจะบันทึกข้อมูลลงในฐานข้อมูล mini → ตาราง points
 - ฟังก์ชันนี้เขียนด้วย PHP (add_point.php) โดยใช้คำสั่ง ST_MakePoint(lon, lat) ใน PostGIS
2. **แก้ไขจุด (Edit):**
 - เมื่อคลิก Marker จะมีปุ่ม “แก้ไข” ใน Popup
 - สามารถเปลี่ยนชื่อ หรือย้ายตำแหน่งจุดได้
 - เมื่อกดบันทึก ระบบจะอัปเดตฐานข้อมูลผ่าน update_point.php
3. **ลบจุด (Delete):**
 - กดปุ่ม “ลบ” ใน Popup
 - ระบบเรียก delete_point.php เพื่อลบข้อมูลจากฐานข้อมูล
4. **ค้นหาจุด (Search):**
 - มี Dropdown รายชื่อจุดทั้งหมดจากฐานข้อมูล
 - เมื่อเลือกชื่อ ระบบจะซูมไปยังตำแหน่งจุดนั้นโดยอัตโนมัติ
5. **ลากย้ายจุด (Drag Marker):**
 - ผู้ใช้สามารถลาก Marker ไปตำแหน่งใหม่ได้
 - ระบบจะอัปเดตพิกัดใหม่เข้าสู่ฐานข้อมูลอัตโนมัติ

ฐานข้อมูลที่เกี่ยวข้อง

- Database: mini
- Table: points
- Fields: id, name, type_dorm, geom
- SRID: 4326 (WGS84)

ไฟล์โค้ดหลัก

- add_point.php — บันทึกข้อมูลใหม่
- update_point.php — แก้ไขข้อมูล
- delete_point.php — ลบข้อมูล
- get_points.php — ดึงข้อมูลทั้งหมดมาแสดงบนแผนที่

4.4 หน้าแนะนำผู้พัฒนา (aboutMe.html)

- แสดงข้อมูลส่วนตัวของผู้จัดทำโครงการ (ชื่อ, รหัสนิสิต, สาขา)
- ใช้ Leaflet.js แสดงแผนที่ขนาดเล็กประกอบการแนะนำตัว
- เพิ่มข้อมูลข้อความเกี่ยวกับความสนใจ และวัตถุประสงค์ของการพัฒนาเว็บไซต์

4.5 โครงสร้างระบบเว็บไซต์

เว็บไซต์แผนที่เชิงโต้ตอบที่พัฒนาขึ้นมีการจัดโครงสร้างแบบแยกส่วน เพื่อให้เข้าใจง่ายและบำรุงรักษาได้สะดวก โดยแบ่งออกเป็น 3 ส่วนหลัก ดังนี้

1. ส่วนหน้าเว็บ (Frontend)

ประกอบด้วยหน้าเว็บหลัก 3 หน้า ได้แก่

- index.html สำหรับแสดงแผนที่หลักและข้อมูลจาก GeoServer, GeoJSON และ API
- addPoint.html สำหรับเพิ่ม แก้ไข ลบ และค้นหาจุดข้อมูลในฐานข้อมูล
- aboutMe.html หน้าแนะนำผู้พัฒนา ออกแบบด้วยโทนสีพาสเทลและแอนิเมชันอ่อน ๆ

2. ส่วนจัดการข้อมูล (Backend)

ใช้ภาษา PHP เชื่อมโยงเว็บไซต์กับฐานข้อมูล PostgreSQL/PostGIS เพื่อจัดการข้อมูล (CRUD) และติดต่อกับ GeoServer ผ่านบริการ WMS/WFS

3. ส่วนข้อมูลเชิงพื้นที่ (Data Layer)

ประกอบด้วย PostGIS สำหรับจัดเก็บข้อมูล, GeoServer สำหรับให้บริการข้อมูลแผนที่, และ QGIS สำหรับเตรียมและตรวจสอบข้อมูล

โครงสร้างระบบทั้งหมดทำงานร่วมกันในรูปแบบ Client-Server Model ทำให้เว็บไซต์สามารถแสดงและจัดการข้อมูลเชิงพื้นที่ได้แบบโต้ตอบและเรียลไทม์

4.6 การวิเคราะห์โค้ดของระบบเว็บไซต์แผนที่เชิงโต้ตอบ

1. ภาพรวมของระบบ

เว็บไซต์พัฒนาในรูปแบบ Client-Server Model โดยใช้เทคโนโลยี Open Source ทั้งหมดเพื่อแสดงและจัดการข้อมูลเชิงพื้นที่แบบเรียลไทม์

โครงสร้างประกอบด้วย

- Frontend (Leaflet.js + HTML + CSS + JavaScript)
- Backend (PHP + PostgreSQL/PostGIS)
- GeoServer ให้บริการข้อมูลเชิงพื้นที่ผ่าน WMS /WFS
- API ภายนอก จาก Thaiwater.net สำหรับข้อมูลฝนและระดับน้ำ

2. การวิเคราะห์โค้ดฝั่ง Frontend

2.1 หน้าเว็บหลัก index.html

ไฟล์นี้ทำหน้าที่เป็นศูนย์กลางในการแสดงแผนที่ โดยใช้ **Leaflet.js** เป็นเครื่องมือหลักในการสร้างแผนที่แบบโต้ตอบ ตัวอย่างการกำหนดแผนที่เริ่มต้น

js

📄 คัดลอกโค้ด

```
var map = L.map("map").setView([17.63, 100.1], 9);
```

- แสดงข้อมูลหลายเลเยอร์ เช่น ขอบเขตจังหวัดจาก GeoJSON ถนน GCP และร้านซักรีด
- ดึงข้อมูลจาก GeoServer ผ่าน WMS และ WFS
- เชื่อม API จาก Thaiwater.net ด้วย fetch() เพื่ออัปเดตข้อมูลฝนและระดับน้ำแบบเรียลไทม์

ฟังก์ชันโหลดข้อมูลฝนและน้ำ

js

📄 คัดลอกโค้ด

```
function loadRainData(){  
    fetch("https://api2.thaiwater.net/v1/thaiwater_rain")  
    .then(res=>res.json())  
    .then(showRainData);  
}
```

ผลลัพธ์: เมื่อหน้าเว็บโหลด จะได้แผนที่พร้อมข้อมูลหลายชั้น ผู้ใช้สามารถเปิด/ปิดเลเยอร์และดูรายละเอียดได้ใน Popup

2.2 หน้าเพิ่มจุด addPoint.html

ไฟล์นี้ใช้สำหรับเพิ่ม แก้ไข ลบ และค้นหาข้อมูลจุดในฐานข้อมูล PostGIS โดยเชื่อมกับ PHP ผ่าน AJAX การเพิ่มจุดทำได้โดยดับเบิลคลิกบนแผนที่

js

📄 คัดลอกโค้ด

```
map.on("dblclick", function(e){  
    showModal(e.latlng.lat, e.latlng.lng);  
});
```

เมื่อผู้ใช้กด Submit ระบบจะส่งข้อมูลไปยัง add_point.php เพื่อบันทึกลงฐานข้อมูล จุดที่บันทึกจะแสดงบนแผนที่ทันทีด้วย refreshPoints() ซึ่งดึงข้อมูลจาก get_points.php

2.3 หน้าแนะนำตัว aboutMe.html

เป็นหน้าแนะนำผู้พัฒนา ใช้โครงสร้าง HTML และ CSS ตกแต่งแบบพาสเทล มีแอนิเมชัน floating และ gradient เพื่อให้เว็บไซต์ดูมีชีวิตชีวา

เชื่อมโยงกลับไปยัง index.html และ addPoint.html ผ่านปุ่ม nav-btn ภายใน header


2.4 สคริปต์ script.js

เป็นหัวใจของการเชื่อมโยงข้อมูลจุดกับฐานข้อมูล

- refreshPoints() โหลดข้อมูลจุดทั้งหมดจากฐานข้อมูล
- editPoint() แก้ไขชื่อและประเภทข้อมูล
- deletePoint() ลบจุดตาม ID
- dragMarker() อัปเดตพิกัดเมื่อมีการลากจุด

ตัวอย่างบางส่วน

js

 คัดลอกโค้ด

```
function deletePoint(id){
  if(confirm("ต้องการลบข้อมูลนี้หรือไม่?")){
    fetch("delete_point.php?id="+id)
      .then(()=>refreshPoints());
  }
}
```


ผลลัพธ์: ข้อมูลจุดในแผนที่จะอัปเดตแบบเรียลไทม์ โดยไม่ต้องรีเฟรชหน้า

3. การวิเคราะห์โค้ดฝั่ง Backend

3.1 ไฟล์ db.php

ทำหน้าที่เชื่อมต่อฐานข้อมูล PostgreSQL/PostGIS โดยใช้ PDO เพื่อความปลอดภัย

php

 คัดลอกโค้ด

```
$pdo = new PDO("pgsql:host=localhost;dbname=mini", "postgres", "1234");
```

ไฟล์นี้ถูกเรียกใช้โดยทุกไฟล์ PHP ที่เกี่ยวข้อง

3.2 ไฟล์ add_point.php

รับข้อมูลที่ผู้ใช้กรอกจาก หน้า addPoint แล้วบันทึกลงในตาราง points โดยใช้ฟังก์ชัน ST_MakePoint()

php

 คัดลอกโค้ด

```
$sql = "INSERT INTO points (name, type_dorm, geom)
VALUES (:name, :type, ST_SetSRID(ST_MakePoint(:lon, :lat),4326))";
```

3.3 ไฟล์ get_points.php

ดึงข้อมูลทั้งหมดจากฐานข้อมูลแล้วแปลงเป็น GeoJSON เพื่อให้ Leaflet อ่านได้

3.4 ไฟล์ update_point.php และ delete_point.php

- update_point.php ใช้สำหรับอัปเดตชื่อ ประเภท และพิกัดของจุด
- delete_point.php ลบจุดออกจากฐานข้อมูลตาม ID

3.5 ไฟล์ get_wfs.php

เป็น Proxy สำหรับดึงข้อมูลจาก GeoServer ผ่าน WFS เพื่อหลีกเลี่ยงปัญหา CORS และควบคุมการเข้าถึง

4. การเชื่อมโยงข้อมูลเชิงพื้นที่

- ข้อมูลขอบเขตและถนนถูกเก็บในฐานข้อมูล PostGIS และเผยแพร่ผ่าน GeoServer ด้วย WMS/WFS
- ข้อมูลจุดถูกจัดเก็บในตาราง points และสื่อสารกับเว็บผ่าน PHP และ AJAX
- ข้อมูลฝนและระดับน้ำดึงจาก API ภายนอกแบบเรียลไทม์ และแสดงผลพร้อม สัญลักษณ์ (icon) เฉพาะ

5. สรุปผลการวิเคราะห์

จากการวิเคราะห์โค้ดทั้งหมด พบว่า

- ระบบใช้เทคโนโลยี Open Source ครบวงจร ทั้งฝั่งผู้ใช้ เซิร์ฟเวอร์ และฐานข้อมูล
- ออกแบบโครงสร้างไฟล์เป็นระบบ แยกหน้าที่ชัดเจน ง่ายต่อการบำรุงรักษา
- รองรับการทำงานแบบ เรียลไทม์ และ Interactive ได้ดี
- สามารถต่อยอดเพิ่มการวิเคราะห์เชิงพื้นที่ (Spatial Analysis) หรือ Dashboard ในอนาคต

บทที่ 4.7 การทดสอบและประเมินผลการทำงานของระบบ

1. วัตถุประสงค์ของการทดสอบ

เพื่อประเมินความถูกต้อง เสถียรภาพ และประสิทธิภาพของเว็บไซต์ Web GIS ในการแสดงผลข้อมูลและการจัดการข้อมูลจุดเชิงพื้นที่ (เพิ่ม – ลบ – แก้ไข – ค้นหา)

2. กระบวนการทดสอบ

ดำเนินการทดสอบโดยใช้เบราว์เซอร์ Google Chrome และ Mozilla Firefox ภายใต้อุปกรณ์ของมหาวิทยาลัย

นเรศวร

โดยทดสอบฟังก์ชันหลัก 5 รายการ ดังนี้

ลำดับ	รายการทดสอบ	วิธีการทดสอบ	ผลลัพธ์ที่ได้	
1	การโหลดแผนที่หลัก	เปิด index.html ตรวจสอบการเรียก WMS/WFS จาก GeoServer	แผนที่แสดงผลครบ ทุกชั้นข้อมูล ใช้เวลา 2–3 วินาที	
2	การเพิ่มจุด (Add Point)	ดับเบิลคลิกบนแผนที่และกรอกข้อมูลใน Modal	จุดปรากฏบนแผนที่ทันที และบันทึกในฐาน PostGIS	
3	การแก้ไข/ลบจุด	กดปุ่ม Edit หรือ Delete บน Popup	ข้อมูลอัปเดตและลบออกจากแผนที่แบบเรียลไทม์	
4	การเชื่อม API ฝน-ระดับน้ำ	รีเฟรชหน้าเว็บตรวจสอบการโหลดข้อมูลจาก Thaiwater.net	ข้อมูลอัปเดตถูกต้อง มีการแสดง Icon	
5	การตอบสนองบนมือถือ	เปิดเว็บไซต์ผ่านสมาร์ทโฟน	Responsive ครบ แผนที่แสดงสมบูรณ์	

3. ผลการทดสอบและวิเคราะห์

ผลการทดสอบแสดงให้เห็นว่าเว็บไซต์ทำงานได้เสถียร รองรับการแสดงผลหลายชั้นข้อมูลและคำสั่ง CRUD ได้ครบถ้วน ไม่มีข้อผิดพลาดในการเชื่อมต่อฐานข้อมูลหรือ GeoServer การประมวลผลแต่ละคำสั่งใช้เวลาน้อยกว่า 1 วินาที ตอบสนองทันทีโดยไม่ต้องรีเฟรชหน้า (Real-time Update)

บทที่ 5 สรุปผลและข้อเสนอแนะของโครงการ

5.1 สรุปผลการดำเนินงาน

โครงการ “พัฒนาเว็บไซต์แผนที่เชิงโต้ตอบเพื่อจัดการข้อมูลเชิงพื้นที่ (Interactive Web GIS)” มีวัตถุประสงค์เพื่อสร้างเว็บไซต์ที่สามารถแสดงผลและจัดการข้อมูลเชิงพื้นที่ได้แบบเรียลไทม์ โดยผสานเทคโนโลยีด้านเว็บและภูมิสารสนเทศเข้าด้วยกันอย่างเป็นระบบ

จากการพัฒนาและทดสอบระบบ พบว่าเว็บไซต์สามารถ

1. แสดงข้อมูลเชิงพื้นที่หลายชั้น (Layer) เช่น ขอบเขตจังหวัด ถนน ร้านซักผ้า และข้อมูลฝน-ระดับน้ำจาก API
2. ดึงข้อมูลจาก GeoServer ผ่านบริการ WMS/WFS ได้ถูกต้องและต่อเนื่อง
3. รองรับการจัดการข้อมูลจุดแบบครบวงจร (เพิ่ม-แก้ไข-ลบ-ค้นหา) ผ่าน PHP เชื่อมโยงกับฐานข้อมูล PostgreSQL/PostGIS
4. ทำงานแบบเรียลไทม์โดยไม่ต้องรีเฟรชหน้าเว็บ
5. ออกแบบให้มีส่วนติดต่อผู้ใช้ที่เข้าใจง่าย ใช้งานได้ทั้งในคอมพิวเตอร์และโทรศัพท์มือถือ

โดยรวมแล้ว ระบบสามารถตอบโจทย์การใช้งานเชิงภูมิสารสนเทศได้ครบถ้วน เหมาะสำหรับการนำไปต่อยอดใช้งานจริง เช่น การบริหารจัดการจุดตรวจวัดน้ำ จุดบริการ หรือสิ่งอำนวยความสะดวกในพื้นที่ต่าง ๆ

5.2 ปัญหาและอุปสรรคที่พบ

- การเชื่อมต่อ API จากเว็บไซต์ภายนอกบางครั้งมีความล่าช้า ทำให้ข้อมูลแสดงผลช้าชั่วคราว
- การเชื่อมโยงระหว่าง GeoServer กับฐานข้อมูล PostGIS ต้องตั้งค่าพอร์ตและสิทธิ์การเข้าถึงอย่างถูกต้อง ไม่เช่นนั้น จะไม่สามารถ Publish ขึ้นข้อมูลได้
- ข้อมูลจากฐาน PostGIS ที่มีจำนวนมากอาจทำให้เว็บไซต์โหลดช้า หากไม่ได้จัดทำระบบแคช (Cache) หรือ Tile Service

5.3 ข้อเสนอแนะเพื่อการพัฒนาในอนาคต

1. เพิ่มระบบการล็อกอินและกำหนดสิทธิ์ผู้ใช้ (User Authentication & Role Management) เพื่อควบคุมการเข้าถึงข้อมูล เช่น ให้ผู้ดูแลระบบเท่านั้นที่สามารถลบหรือแก้ไขข้อมูลได้
2. พัฒนา Dashboard หรือระบบวิเคราะห์เชิงพื้นที่ (Spatial Analysis Dashboard) เพื่อให้ผู้ใช้สามารถสรุปผลข้อมูล เช่น ปริมาณฝนเฉลี่ย หรือพื้นที่เสี่ยงน้ำท่วม ผ่านกราฟหรือแผนภาพ
3. เชื่อมต่อฐานข้อมูลจากภายนอก (External Database Integration) เช่น ข้อมูลสภาพอากาศจากกรมอุตุนิยมวิทยา หรือข้อมูลจราจร เพื่อเพิ่มความหลากหลายของข้อมูล
4. เพิ่มระบบแคช (Caching System) เพื่อเพิ่มความเร็วในการโหลดข้อมูล โดยเฉพาะเมื่อมีผู้ใช้งานพร้อมกันหลายคน
5. ปรับปรุงการออกแบบส่วนติดต่อผู้ใช้ (User Interface Design) ให้มีความทันสมัย สวยงาม และรองรับอุปกรณ์พกพา (Mobile Responsive) ได้ดียิ่งขึ้น

5.4 บทสรุปโดยรวม

โครงการนี้สามารถพัฒนาเว็บไซต์แผนที่เชิงโต้ตอบที่ทำงานได้จริง และเชื่อมโยงข้อมูลจากหลายแหล่งได้อย่างมีประสิทธิภาพ แสดงให้เห็นถึงการบูรณาการความรู้ด้าน Web Development และ Geographic Information System (GIS) ได้อย่างสมบูรณ์

ระบบนี้จึงเป็นต้นแบบที่ดีในการเรียนรู้และต่อยอดสู่ระบบภูมิสารสนเทศระดับองค์กร หรือหน่วยงานราชการ เช่น การจัดการทรัพยากรน้ำ การวางแผนพื้นที่เกษตร หรือการวิเคราะห์ข้อมูลเชิงพื้นที่อื่น ๆ ในอนาคต

