

# NEW\_WORLD\_MAP v14.1 기능 점검 및 개선 보고서

## 목차

1. 개요 - HTML v14.1 기능 검증 개요 및 분석 대상
2. 주요 오류 및 비동작 기능 식별 - 동작하지 않는 UI 기능 목록과 원인 분석
3. 기능 구현 비교 (TXT vs HTML) - 기능별 구현 여부 매핑표
4. 기능별 수정/추가 방안 - 누락된 기능에 대한 개선 전략 및 코드 예시
5. 4.1 모바일 사이드바 토글 기능 추가
6. 4.2 엑셀 시트 전환 기능 구현
7. 4.3 위/경도 열 선택 값 반영
8. 4.4 지도 배경 스타일 선택 기능 구현
9. 4.5 마커 스타일 변경 즉시 반영 및 선택 상태 유지
10. AI 개선 명령어 프롬프트 예시 - Claude/GPT에 기능 개선을 요청하는 효율적 프롬프트 사례

## 1. 개요

청주 금연구역 점검 지도의 통합 버전(v14.1) HTML 파일과 기능 기준으로 제공된 이전 버전(v12.1) 텍스트를 비교 분석하였습니다. v14.1 HTML UI 상에서 버튼이나 입력 필드는 보이지만 동작하지 않거나, TXT에 명시된 기능이 HTML에서 누락된 경우를 중점적으로 확인했습니다. 본 보고서에서는 이러한 비동작 기능과 누락 기능을 식별하고, 코드 상의 원인을 분석하였으며, 각 기능을 복구하거나 재구현하기 위한 방안을 제시합니다. 또한, Claude AI (Code Interpreter)나 GPT-4 같은 AI 도구를 활용하여 이러한 수정 작업을 수행할 수 있도록 효율적인 프롬프트 설계 예시를 제공합니다. 기술 수준은 중급 이상의 개발자 및 AI 프롬프트 엔지니어를 대상으로 하며, 핵심 내용은 표와 코드 스니펫으로 구조화하였습니다.

## 2. 주요 오류 및 비동작 기능 식별

v14.1 HTML에서 UI는 존재하지만 사용자 조작에 반응하지 않는 주요 기능은 다음과 같습니다. 각 항목에 대해 증상과 원인을 분석한 결과, 대부분 이벤트 리스너 누락 또는 로직 미구현으로 확인되었습니다.

### 2.1 비동작 기능 목록 및 분석표

아래 표는 v14.1에서 발견된 동작하지 않는 UI 기능들과 그 문제 원인을 정리한 것입니다.

비동작 UI 기능	문제 상황 및 원인 분석
모바일 메뉴 토글 버튼 (≡ 사이드바 열기)	화면 상단에 메뉴 버튼( #toggleSidebar )이 보이나, 클릭 시 좌측 사이드바가 열리거나 닫히지 않음. 원인: 해당 버튼에 대한 이벤트 리스너 미구현 (사이드바 표시/숨김 동작이 연결되지 않음).
엑셀 시트 선택 드롭다운 (시트 전환)	파일 업로드 후 다중 시트 Excel의 시트를 선택하는 드롭다운( #sheetSelect )이 표시되지만, 사용자가 값을 변경해도 데이터가 그 새로운 시트로 로드되지 않음. 원인: 시트 선택 변경 시 processWorksheet를 호출하는 로직 없음 (이벤트 리스너 누락).

비동작 UI 기능	문제 상황 및 원인 분석
위도/경도 열 선택 옵션 (CSV/Excel 열 매핑)	고급 설정 내 위도 열/경도 열 선택 창( <code>#latColumn</code> , <code>#lngColumn</code> )이 제공되나, 실제 데이터 파싱에 이 선택값이 반영되지 않음. <b>원인:</b> <code>processWorksheet</code> 에서 위경도 컬럼 인덱스를 고정 값으로 사용하고 있으며 선택된 옵션을 참조하지 않음.
배경지도 스타일 선택 (VWorld/없음)	설정 섹션의 지도 배경 선택 메뉴( <code>#mapStyle</code> : VWorld 지도/배경지도 없음)가 존재하나, 선택을 변경해도 지도 배경에 변화가 없음. <b>원인:</b> 해당 <code>&lt;select&gt;</code> 변경 시 타일 레이어를 추가/제거하는 처리가 없음 (이벤트 리스너 및 관련 로직 미구현).
마커 스타일 변경 즉시 반영 (원형/사각/말풍선)	설정 섹션의 마커 스타일 옵션( <code>#markerStyle</code> )을 변경해도 지도 위 마커 모양이 바로 갱신되지 않음. <b>원인:</b> 마커 스타일 변경 이벤트 리스너가 없어, 사용자가 값을 바꿔도 <code>updateMarkers()</code> 등이 호출되지 않음. (필터 재적용이나 새로고침 시에만 반영되는 상태) 또한, 수동으로 <code>updateMarkers()</code> 를 호출하면 선택된 마커 강조 표시가 초기화되는 부수 문제가 있음.

위 문제들은 UI에서 사용자 경험을 저해하는 요소로, 아래 3. 기능 구현 비교표에서 전체 기능 중 누락된 부분을 맥락 속에서 더 살펴보고, 4. 기능별 수정 방안에서 각각의 해결책을 상세히 제시합니다.

### 3. 기능 구현 비교 (TXT vs HTML)

v12.1 기능 기준 TXT에 명시된 주요 기능들이 v14.1 HTML에 어떻게 구현되었는지 비교한 매핑표입니다. 기능의 구현 여부를 O (구현됨), X (누락/미구현), △ (부분 구현/부분 오류)로 표시했습니다.

기능명	v14.1 구현 여부	설명
다크 모드 토글 (화면 테마)	O (구현됨)	좌측 메뉴에서 다크 모드 스위치( <code>#darkMode</code> )로 body 클래스 토글 - 정상 동작
파일 업로드 (Excel/CSV)	O (구현됨)	엑셀/CSV 선택 버튼 및 숨겨진 <code>&lt;input type="file"&gt;</code> 로 파일 로드 - 정상 동작
다중 시트 선택 (Sheet 선택)	X (누락)	시트 드롭다운 UI는 생성되나, 선택 변경에 따른 데이터 재로드 미구현 (항상 첫 시트만 로드)
위도/경도 열 선택 (컬럼 매핑)	X (누락)	UI 제공되나, 실제 <code>processWorksheet</code> 에서 고정된 열 인덱스(H/G)만 사용 (선택값 반영 안 됨)
좌표 없는 데이터 포함 필터	O (구현됨)	<code>#includeNoCoords</code> 체크박스 구현 - 정상 동작 (체크시 좌표 null도 표시)
미점검 시설만 표시 필터	O (구현됨)	<code>#onlyUninspected</code> 체크박스 구현 - 정상 동작 (inspectionDate 유무로 필터)
검색 및 필터 칩 (군집/지역/업종)	O (구현됨)	검색창 및 군집/읍면동/카테고리별 필터 - 정상 동작 (칩 토글 및 토글 All 포함)
필터 결과 일괄 선택 추가	O (구현됨)	“ 필터결과 모두 선택 ” 버튼 - 정상 동작 (일일 한도 고려하여 선택목록 추가)
선택목록 패널 표시/갱신	O (구현됨)	🔍 선택 목록 (0) 버튼 - 정상 동작 (토글로 패널 열기/닫기 및 목록 갱신)

기능명	v14.1 구현 여부	설명
모바일 사이드바 토글 (메뉴 열기)	△ (부분 구현)	≡ 메뉴 버튼 UI만 존재 - 이벤트 미구현으로 클릭 시 반응 없음 (모바일에서 사이드바 열 수 없음)
마커 클러스터링 토글	O (구현됨)	<code>#clusterMarkers</code> 체크박스 - 정상 동작 (클러스터 그룹 토글)
라벨 표시 토글	O (구현됨)	<code>#showLabels</code> 체크박스 - 정상 동작 (마커 tooltip 토글)
마커 스타일 선택 (원형/사각 등)	△ (부분 구현)	스타일 <code>&lt;select&gt;</code> UI는 있음 - 변경 즉시 반영되지 않음 (이벤트 누락), 추후 새로고침 시 적용됨
마커 크기 조절 (슬라이더)	O (구현됨)	<code>#markerSize</code> range 슬라이더 - 정상 동작 (input시 CSS 변수 및 표시값 업데이트)
지도 영역 전체보기 (Fit bounds)	O (구현됨)	🔍 전체 보기 버튼 - 정상 동작 ( <code>fitToMarkers()</code> 로 모든 마커 영역으로 지도 이동)
지도 새로고침 (Refresh)	O (구현됨)	새로고침 버튼 - 정상 동작 (필터/마커 초기화 후 재적용)
경로 계산 (선택목록)	O (구현됨)	경로 계산 버튼 - 정상 동작 (최근접/클러스터/TSP 알고리즘, fixEnds 옵션 처리)
시작/끝 지점 고정 옵션	O (구현됨)	<code>#fixEnds</code> 체크박스 - 정상 동작 (경로 알고리즘에 반영됨)
경로 외 마커 숨기기 옵션	O (구현됨)	<code>#toggleHideNonRoute</code> 체크박스 - 정상 동작 (경로 계산 후 비경로 마커 <code>.hidden-marker</code> 처리)
A/B 거리 측정 (두 점 간 거리)	O (구현됨)	지도에서 A/B 선택 기능 - 정상 동작 (A,B 선택 후 직선거리 표시 및 폴리라인)
경로 초기화 (Clear Route)	O (구현됨)	A/B 경로 초기화 버튼 - 정상 동작 (A/B 모드 해제 및 마커/라인 제거)
외부 내비게이션 연동 (길찾기)	O (구현됨)	마커 팝업 내 길찾기 버튼 - 정상 동작 (카카오맵, 네이버지도, 구글지도 경로 새 창으로 모두 열기)
선택목록 Excel 내보내기	O (구현됨)	선택항목 내보내기 버튼 - 정상 동작 (선택된 항목만 추출하여 Excel 다운로드)
전체 데이터 Excel 내보내기	O (구현됨)	Excel 내보내기 버튼 - 정상 동작 (필터 적용된 전체 데이터를 Excel로 저장)
PDF 내보내기 (지도 +메타정보)	O (구현됨)	PDF 내보내기 버튼 - 정상 동작 (지도 캡처 이미지 및 통계 정보로 PDF 생성)
이미지 내보내기 (지도 캡처 PNG)	O (구현됨)	🖨 이미지 내보내기 버튼 - 정상 동작 (지도 캔버스 캡처를 PNG로 다운로드)
선택목록 비우기 (모두 삭제)	O (구현됨)	🗑 모두 삭제 버튼 - 정상 동작 (선택 마커 초기화 및 선택목록/경로 리셋)

(주: O=정상 구현, X=미구현, △=부분 구현/오류)

위 표에서 굵게 표시된 기능들 중 X/△로 분류된 부분이 곧 v14.1에서 개선이 필요한 사항입니다. 다음 장에서는 이러한 누락/오류 기능들에 대한 구체적인 수정 방안과 코드를 제시합니다.

## 4. 기능별 수정/추가 방안

앞서 확인한 누락된 기능들에 대해, HTML 구조와 JS 코드 상 어디를 수정/추가해야 하는지 구체적으로 설명합니다. 각 섹션에서는 필요한 변경을 간략히 요약하고, 예시 코드를 포함하여 명확한 적용 방안을 제시합니다. 제시된 코드 스니펫은 v14.1 코드베이스에 직접 적용할 수 있는 형태로 작성되었습니다.

### 4.1 모바일 사이드바 토글 기능 추가

**문제점:** 모바일 화면에서 사이드바(`#sidebar`)가 기본적으로 숨겨진 채(`transform: translateX(-100%)`)이고, 상단 메뉴 버튼(`#toggleSidebar`)을 눌러도 사이드바를 표시할 방법이 없습니다. 이는 `#toggleSidebar` 버튼 클릭 시 사이드바에 `open` 클래스를 부여/제거하여 열고 닫는 로직이 누락되었기 때문입니다.

**해결 방안:** 자바스크립트 초기화 코드에서 해당 버튼의 이벤트 리스너를 추가해야 합니다. v14.1에서는 이벤트 핸들러들을 모아 `setupEventListeners()` 메서드에서 정의하고 있으므로, 여기에 사이드바 토글 기능을 구현합니다. 버튼 클릭 시 사이드바 요소에 `.open` 클래스를 토글하면, CSS의 미디어쿼리에서 정의된 `#sidebar.open { transform: translateX(0); }`가 적용되어 사이드바가 화면에 슬라이드인 됩니다.

```
// (CheongJuInspectionMap.setupEventListeners 내부)
document.getElementById('toggleSidebar').addEventListener('click', () => {
  const sidebar = document.getElementById('sidebar');
  sidebar.classList.toggle('open');
});
```

위 코드를 `setupEventListeners()` 함수 내 적절한 위치(예: 다른 이벤트 리스너 등록 구문들 마지막)에 추가합니다. 이렇게 하면 모바일 환경 등에서 ≡ 메뉴 버튼을 클릭하여 사이드바를 열고 닫을 수 있습니다. 추가로, 사이드바가 열릴 때 화면 폭에 따라 필요 시 `#toggleSidebar` 버튼을 숨기거나 (`display: none`), 반대로 사이드바 닫힘 시 다시 보이게 하는 세밀한 UX 조정은 상황에 따라 고려할 수 있습니다. 기본적으로는 위 한 줄 추가로 핵심 기능은 복구됩니다.

### 4.2 엑셀 시트 전환 기능 구현

**문제점:** 파일을 로드할 때 첫 번째 시트만 처리하고, 사용자가 시트 드롭다운(`#sheetSelect`)을 변경해도 아무 일도 일어나지 않습니다. v14.1 `loadFile()` 메서드에서는 업로드한 Excel의 모든 시트 이름을 드롭다운에 채우지만, 시트 선택 변경에 대한 이벤트 처리가 없습니다.

**해결 방안:** `loadFile()` 함수 내에서 시트 목록을 채운 직후, 드롭다운의 `change` 이벤트 리스너를 추가합니다. 선택된 시트 이름에 대응하는 워크시트를 찾아 `processWorksheet(...)` 로 불러오도록 구현합니다. 이때 `loadFile`의 지역 변수인 `workbook`을 이벤트에서 참조할 수 있도록 클로저를 이용합니다.

```
// (CheongJuInspectionMap.loadFile 내부, sheetSelect 옵션 채운 직후)
sheetSelect.addEventListener('change', () => {
  const sheetName = sheetSelect.value;
  const newWorksheet = workbook.Sheets[sheetName];
```

```

if (newWorksheet) {
  this.processWorksheet(newWorksheet);
}
});

```

위 처럼 `sheetSelect`의 `change` 핸들러를 등록하면, 사용자가 다른 시트를 선택할 때 즉시 해당 시트의 데이터를 불러와서 지도와 표를 갱신할 수 있습니다. 이 변경으로 다중 시트 Excel 파일도 자유롭게 전환하여 확인할 수 있게 됩니다. (참고: `workbook` 객체는 `XLSX.read` 결과로 얻은 통합 문서이며, `Sheets` 프로퍼티로 각 시트의 데이터를 접근합니다.)

### 4.3 위/경도 열 선택 값 반영

**문제점:** 고급 설정에 **위도 열** 및 **경도 열**을 사용자가 지정할 수 있게 UI를 제공하지만, 정작 코드에서는 컬럼 인덱스를 고정된 값으로 사용하고 있습니다. 그 결과, 사용자가 다른 열을 선택해도 데이터 파싱에 전혀 영향을 주지 않습니다.

v14.1 `processWorksheet()` 코드를 살펴보면, 예를 들어 위도(`lat`)는 `row[7]` (즉, H열)로, 경도(`lng`)는 `row[6]` (G열)로 고정되어 있습니다. 이 인덱스들은 기본값으로 UI에서 선택되어 있지만, 만약 사용 데이터 형식이 달라 사용자가 드롭다운에서 다른 열 문자를 고르면 현재는 무시되는 것입니다.

**해결 방안:** `processWorksheet()` 에서 데이터 매핑 시, 고정 인덱스 대신 **현재 설정된 열 선택값**을 참조하도록 변경합니다. 방법은 다음과 같습니다: 1. 함수 진입 시 `document.getElementById('latColumn').value` 와 `...('lngColumn').value` 를 읽어 현재 선택된 열 문자를 가져옵니다 (예: "H", "G"). 2. 해당 문자를 0-index 숫자 인덱스로 변환합니다. ('A' -> 0, 'B' -> 1, ... 'H' -> 7 등). 간단하게는 `letter.charCodeAt(0) - 'A'.charCodeAt(0)` 공식을 사용할 수 있습니다. 3. 그 인덱스를 이용해 `row[...]` 에서 값을 가져오도록 수정합니다.

```

// (CheongJuInspectionMap.processWorksheet 내부 데이터 매핑 부분 개선)
const latCol = document.getElementById('latColumn').value; // 예: "H"
const lngCol = document.getElementById('lngColumn').value; // 예: "G"
const latIndex = latCol.charCodeAt(0) - 'A'.charCodeAt(0); // "H"->7
const lngIndex = lngCol.charCodeAt(0) - 'A'.charCodeAt(0); // "G"->6

this.data = jsonData.slice(1).map((row, index) => {
  return {
    // ... 기타 필드 매핑 ...
    lat: parseFloat(row[latIndex]) || null,
    lng: parseFloat(row[lngIndex]) || null,
  };
}).filter(item => item.name);

```

위와 같이 수정하면, `latColumn`/`lngColumn` 드롭다운의 설정에 따라 해당 열의 데이터를 읽어와 `this.data`를 구성하게 됩니다. **주의:** 이 로직이 적용되려면 `processWorksheet`가 호출되는 타이밍에 사용자가 원하는 열을 선택한 상태여야 합니다. 파일을 불러온 직후 시점을 기준으로 하면, 고급 설정에서 사용자가 열을 변경한 뒤 다시 파일을 불러오는 워크플로우를 염두에 두어야 합니다. 필요하다면 `latColumn` 또는 `lngColumn`의 `change` 이벤트에 대해 `processWorksheet`를 재호출하는 기능도 고려할 수 있습니다 (예: 사용자가 열을 바꿀 때 즉시 현재 데이터에 재적용). 기본적으로는 파일 불러오기 전에 올바른 열이 선택되어 있다는 가정하에 위 처리만으로도 정상 동작할 것입니다.

## 4.4 지도 배경 스타일 선택 기능 구현

**문제점:** 설정 섹션의 **지도 배경** 옵션 (`#mapStyle`)은 VWorld 지도와 배경지도 없음 두 가지를 선택할 수 있게 되어 있지만, 선택을 바꿔도 지도 타일 레이어에 아무런 변화가 없습니다. 초기 `setupMap()`에서는 VWorld 지도를 기본 타일로 추가하지만, 이후 사용자의 입력을 반영하는 코드가 없기 때문에 **지도 배경 전환 기능이 사실상 죽어있는 상태**입니다.

**해결 방안:** 두 부분의 구현이 필요합니다: - **(A) 기본 타일 레이어 참조 보관:** `setupMap()`에서 지도 초기화 시 `L.tileLayer(...)`의 반환 값을 클래스의 프로퍼티(e.g. `this.baseLayer`)로 저장합니다. 이렇게 해야 이후에 해당 레이어를 제거하거나 다시 추가할 수 있습니다. (현재 코드에서는 `L.tileLayer(...).addTo(this.map)`로 추가만 하고 참조를 보관하지 않음) - **(B) #mapStyle 변경 이벤트 처리:** `setupEventListeners()` 등에 `mapStyle` `<select>`의 `change` 이벤트 리스너를 추가합니다. 선택 값이 "none" 일 경우 `this.baseLayer`를 `this.map`에서 제거(`removeLayer`)하고, "vworld" 일 경우 `this.baseLayer`를 다시 지도에 추가(`addLayer`)하는 식으로 분기합니다.

아래는 구현 예시입니다:

```
// (CheongJuInspectionMap.setupMap 수정)
this.baseLayer = L.tileLayer(vworldURL, { attribution: '© VWorld', maxZoom: 18 });
this.baseLayer.addTo(this.map);

// (CheongJuInspectionMap.setupEventListeners 추가)
document.getElementById('mapStyle').addEventListener('change', (e) => {
  if (e.target.value === 'none') {
    // 배경지도 제거
    if (this.baseLayer) this.map.removeLayer(this.baseLayer);
  } else {
    // VWorld 지도 표시
    if (this.baseLayer && !this.map.hasLayer(this.baseLayer)) {
      this.map.addLayer(this.baseLayer);
    }
  }
});
```

위에서 `vworldURL`은 VWorld WMTS API URL 문자열이고, `this.baseLayer`는 초기화 시 저장해둔 Leaflet 타일 레이어 객체입니다. 이렇게 하면 사용자가 배경지도 없음을 선택할 경우 기존 베이스 타일을 제거하여 지도 배경이 없어지고, 다시 VWorld 지도로 변경하면 해당 타일 레이어를 복구할 수 있습니다.

**추가 고려:** 만약 배경지도를 제거한 상태에서 사용자가 지도를 팬/줌 하면 회색 배경만 나오게 되는데, UX 향상을 위해 이 상태에서의 배경 표현(예: 격자 무늬 또는 안내 텍스트)을 추가로 고려할 수 있지만, 이는 범위를 벗어난 개선이므로 기본 동작 구현에 집중합니다.

## 4.5 마커 스타일 변경 즉시 반영 및 선택 상태 유지

**문제점:** 현재 `#markerStyle` (마커 모양 선택) 변경 시 즉각적인 지도 업데이트가 없습니다. 사용자가 원형/사각형/말풍선으로 스타일을 바꿔도 새 데이터 로드나 필터 동작이 일어나야만 변화가 적용됩니다. 이는 마커를 다시 그려주는 `updateMarkers()` 나 이에 준하는 함수 호출이 필요함을 의미합니다. 한편, 단순히 `updateMarkers()`를 호출하면 모든 마커를 새로 생성하기 때문에 **이미 선택된 마커들의 강조 표시(클래스 `selected` 및 크기 확대 스타일)**가 초기화되는 부작용이 있습니다.

**해결 방안:** 1. **스타일 변경 이벤트 추가:** `setupEventListeners()` 에서 `#markerStyle` `<select>` 의 `change` 이벤트를 감지하여 `this.updateMarkers()` 를 호출합니다. 이렇게 하면 사용자가 스타일을 바꾸는 즉시 모든 마커 아이콘 HTML을 새 스타일로 다시 생성하여 지도에 반영합니다. 2. **선택 상태 보존:** `updateMarkers()` 는 내부적으로 `clearAllMarkers()` 후 `createMarker()` 로 마커들을 새로 추가합니다. 이 과정에서 기존에 선택되었던 마커들의 **DOM 요소가 새로 만들어지므로**, 선택 표시가 사라집니다. 이를 해결하려면, `updateMarkers()` 호출 후 현재 `this.selectedMarkers` 집합을 순회하여 해당 마커 요소에 다시 `.selected` 클래스를 부여하고 스타일을 조정해 줍니다. (`addToSelection()` 메서드에서 처음 선택 시 하는 작업을 재수행)

구현 예시는 다음과 같습니다:

```
// (CheongJuInspectionMap.setupEventListeners 내부)
document.getElementById('markerStyle').addEventListener('change', () => {
  this.updateMarkers();
  // 선택된 마커들의 강조 스타일 유지 처리
  this.selectedMarkers.forEach(id => {
    const marker = this.markers.get(id);
    if (marker) {
      const el = marker.getElement();
      el.classList.add('selected');
      el.style.transform = 'scale(1.2)';
      el.style.zIndex = '1000';
    }
  });
});
```

위 코드에서, `updateMarkers()` 호출 직후 `this.markers` 맵에는 새로 생성된 모든 마커가 담겨 있습니다. `this.selectedMarkers` (선택된 항목들의 ID 집합)를 이용하여 각 선택된 마커를 찾아가 `.selected` 클래스를 다시 달고, CSS 변환과 z-index를 원래 강조된 상태로 재적용합니다. 이렇게 하면 사용자가 마커 스타일을 변경해도 선택목록에 있던 마커들은 계속 강조 표시가 유지되어 사용자 혼란을 줄일 수 있습니다.

**대안:** 보다 근본적으로는 `updateMarkers()` 내에서 선택 상태를 고려하여 마커 생성 시 `.selected` 클래스를 붙이는 로직을 포함하는 것이 바람직합니다. 다만 수정 범위를 최소화하기 위해 위와 같은 보조 처리를 이벤트 쪽에서 수행하였습니다. 이 변경 후에는 사용자가 마커 스타일을 변경하는 즉시 새 스타일이 지도에 반영되고, 선택 표시도 유지됩니다.

## 5. AI 개선 명령어 프롬프트 예시

마지막으로, 위에 제시한 수정 작업들을 **AI(Code Interpreter 또는 GPT)**에게 효과적으로 지시하는 예시 프롬프트를 제공합니다. 토큰 절약을 위해 핵심만 간결하게 지시하며, AI가 이해하기 쉽게 **파일명, 함수명, 추가할 코드** 등을 구체적으로 언급합니다. 각 프롬프트는 현재 코드베이스를 수정/보완하는 형태로 작성되었습니다.

### 5.1 사이드바 토글 이벤트 추가

파일 "NEW\_WORLD\_MAP\_CLUSTER\_NAV\_v14.1.html"을 열고, `CheongJuInspectionMap` 클래스의 `setupEventListeners` 함수에 사이드바 토글 기능을 추가해주세요.  
 구체적으로: `setupEventListeners` 마지막에 다음 코드를 삽입합니다.

```
document.getElementById('toggleSidebar').addEventListener('click', () => {
  document.getElementById('sidebar').classList.toggle('open');
});
```

## 5.2 엑셀 시트 전환 로직 구현

"NEW\_WORLD\_MAP\_CLUSTRO\_NAV\_v14.1.html" 내 loadFile 함수에서, sheetSelect 드롭다운의 onchange 이벤트를 처리해주세요.

sheetSelect 옵션들을 채운 바로 다음에 아래 코드를 추가합니다:

```
sheetSelect.addEventListener('change', () => {
  const name = sheetSelect.value;
  const ws = workbook.Sheets[name];
  if (ws) this.processWorksheet(ws);
});
```

이렇게 하면 사용자가 시트를 변경할 때 해당 시트 데이터를 불러옵니다.

## 5.3 위도/경도 열 선택값 반영

파일 "NEW\_WORLD\_MAP\_CLUSTRO\_NAV\_v14.1.html"의 processWorksheet 함수에서, 위도/경도 컬럼 인덱스를 동적으로 설정하도록 코드를 수정해주세요.

row[7], row[6]처럼 고정된 인덱스 대신, latColumn과 lngColumn의 선택값을 읽어 인덱스를 계산한 뒤 사용합니다. 예를 들어:

```
const latCol = document.getElementById('latColumn').value;
const lngCol = document.getElementById('lngColumn').value;
const latIndex = latCol.charCodeAt(0) - 65;
const lngIndex = lngCol.charCodeAt(0) - 65;

... 이후 item.lat = parseFloat(row[latIndex]) || null, item.lng =
parseFloat(row[lngIndex]) || null 로 적용.
```

이렇게 변경하여 사용자 지정 열이 반영되도록 해주세요.

## 5.4 지도 배경 선택 기능 개선

"NEW\_WORLD\_MAP\_CLUSTRO\_NAV\_v14.1.html"에서:

- 1) setupMap 함수에서 기본 지도 타일 레이어를 this.baseLayer 변수에 할당하도록 수정.
- 2) setupEventListeners 함수에 mapStyle 선택(<select id="mapStyle">)의 변경 이벤트를 추가. 선택값이 "none"이면 this.map.removeLayer(this.baseLayer), 값이 "vworld"이면 this.map.addLayer(this.baseLayer)를 호출.

위 두 가지를 구현해서 배경지도 선택이 동작하도록 만드세요.



## 5.5 마커 스타일 변경 즉시 적용

파일 "NEW\_WORLD\_MAP\_CLUSTRO\_NAV\_v14.1.html"의 `setupEventListeners` 함수에 `markerStyle <select>` 변경 시 마커를 다시 그리는 기능을 추가해주세요.

구체적으로:

`markerStyle`의 'change' 이벤트에서 `this.updateMarkers()`를 호출하고, 이후 `this.selectedMarkers`를 순회하면서 선택된 마커들 (`this.markers`에서 해당 id로 가져온)의 DOM 요소에 `.selected` 클래스를 다시 추가하고 `transform/zIndex` 스타일도 복구해주세요.

이로써 마커 스타일 변경 즉시 지도 아이콘이 바뀌고, 선택되어 있던 마커는 하이라이트 유지되도록 합니다.

以上の프롬프트들은 각 기능별로 압축적으로 작성되었으며, 실제 Claude나 GPT에게 입력 시 해당 지시에 맞춰 코드를 수정해줄 것으로 기대됩니다. 필요에 따라 파일 경로나 함수명을 정확히 지정함으로써 AI가 혼동 없이 수정을 수행하도록 유도했습니다.

## 결론

본 보고서에서는 **NEW\_WORLD\_MAP\_CLUSTRO\_NAV\_v14.1** HTML의 기능 구현 상태를 점검하고, **누락되었거나 정상 동작하지 않는 기능들을 식별하여 그 원인을 분석**하였습니다. 이어서, 해당 기능들의 복구 및 개선을 위한 구체적인 코드 수정 방안을 제시하고, AI를 활용한 자동 수정을 위한 모범 프롬프트 예시도 함께 제공하였습니다.

요약하면, v14.1에서 **모바일 사이드바 토글, 엑셀 시트 전환, 위도/경도 열 매핑, 지도 배경 선택, 마커 스타일 변경** 등의 기능이 보완이 필요했고, 각 경우별로 **이벤트 리스너 추가나 로직 개선**을 통해 손쉽게 기능 복구가 가능함을 확인했습니다. 제시된 수정 방안들을 적용하면, 사용자 인터페이스의 일관성과 편의성이 향상되고, v12.1에서 의도했던 모든 기능들이 v14.1에서도 온전히 동작하게 될 것입니다.

향후 이와 같은 통합 버전 업데이트 시에는, 새로운 UI 요소를 추가할 때 대응하는 JS 로직이 누락되지 않도록 유의하고, 크로스체크를 거치는 것이 바람직합니다. 또한, AI 코드 지원 도구를 사용할 경우 본 보고서의 프롬프트 예시처럼 **명확한 지시**를 주는 것이 원하는 결과를 얻는 핵심임을 마지막으로 강조드립니다.