© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722297

Leveraging Volunteered Geographic Information and High-Resolution Satellite Imagery in a Web-GIS Framework for Smart Campus Navigation

WAREKUROMOR TUBOLAYEFA (PHD)[1], ADEPOJU OLUMIDE MATHEW (PHD)[2], ALAGA ABAYOMI TAOFEEK[3], JEGA IDRIS MOHAMMMED[4], ASHIBUOGWU KEVIN[5], UMAR TANIMU YAKUBU[6], OKONKWO-BRUNO CHUKWUNWEIKE[7], BULUS EMMANUEL DUWONGS[8], MOHAMMED JIBRIL TURAKI[9], ALFRED JOY OCHOJONA[10], ABAH CATHERINE JUMMAI[11], EBUH ALEXANDRA CHUKWUYEM[12], ISA BILYAMINU SHAMAKI[13], OSAYI VICTOR NNABUIKE[14], YAHAYA AHMED[15], MUSTAPHA ABBAS YAHUZA[16], SUNMOLA KUDIRAT OLADAYO[17]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17] *National Space Research and Development Agency (NASRDA), Abuja, Nigeria.*

Abstract- Micro-scale navigation within academic campuses, government complexes, and scientific facilities represents a significant research and application gap for standard Global Navigation Satellite System (GNSS) platforms. This gap is primarily driven by the absence of granular internal routing networks and facility-level spatial metadata. Commercial web-mapping applications routinely depict large institutional campuses as featureless geometric boundary polygons, creating geographic voids that lack internal pedestrian walkways and building entry points. This paper presents the technical design, system architecture, and practical deployment of NasrdaNavi, a web-based campus navigation system developed for the National Space Research and Development Agency (NASRDA) headquarters at the Obasanjo Space Centre in Abuja, Nigeria. Utilizing a hybrid geospatial pipeline, the system integrates localized drone imagery from the NASRDA data pipeline, crowdsourced vectors from OpenStreetMap (OSM), and high-resolution ESRI World Imagery satellite basemaps. These multi-source spatial layers were geometrically cleaned, topologically corrected in QGIS, and served via a lightweight Flask backend and MapLibre GL JS client engine. The single-page application (SPA) executes real-time hardware geolocation tracking, attribute-based point-of-interest (POI) queries, dynamic reachable-area (isochrone) polygon generation, and client-side pathfinding across private institutional road and footpath networks without heavy server-side processing. This study provides a cost-effective, replicable framework for overcoming last-mile campus navigation bottlenecks while operationalizing practical space-to-earth geospatial solutions within Spatial Data Infrastructure (SDI) guidelines and addressing critical constraints of national spatial data sovereignty.

Keywords: Web-GIS, Campus Micro-Navigation, OpenStreetMap, MapLibre GL JS, Satellite Imagery, Data Sovereignty, Spatial Data Infrastructure

I. INTRODUCTION

Commercial web-mapping navigation platforms such as Google Maps, Apple Maps, and Bing Maps have transformed macro-scale vehicular routing and urban navigation globally (Netek et al., 2020). Over the past decade, cloud-based mapping platforms like Esri ArcGIS Online, CARTO, and Mapbox Studio have democratized web cartography by enabling organizations to author, style, and host interactive web maps via accessible user interfaces (Roth et al., 2021; Zejdlik & Vozenelik 2025).

However, commercial web-mapping services exhibit structural limitations at the micro-scale. Private institutional, academic, research, and military complexes are frequently depicted as coarse geometric boundary polygons, termed "geographic voids" that lack internal pedestrian paths, driveway connectivity, parking structures, building entry thresholds, or departmental metadata (Neelakantappa et al., 2025). Commercial map providers lack economic incentive to map closed or semi-private facilities, leaving visitors, emergency services, and facility managers without reliable turn-by-turn guidance once inside campus perimeters.

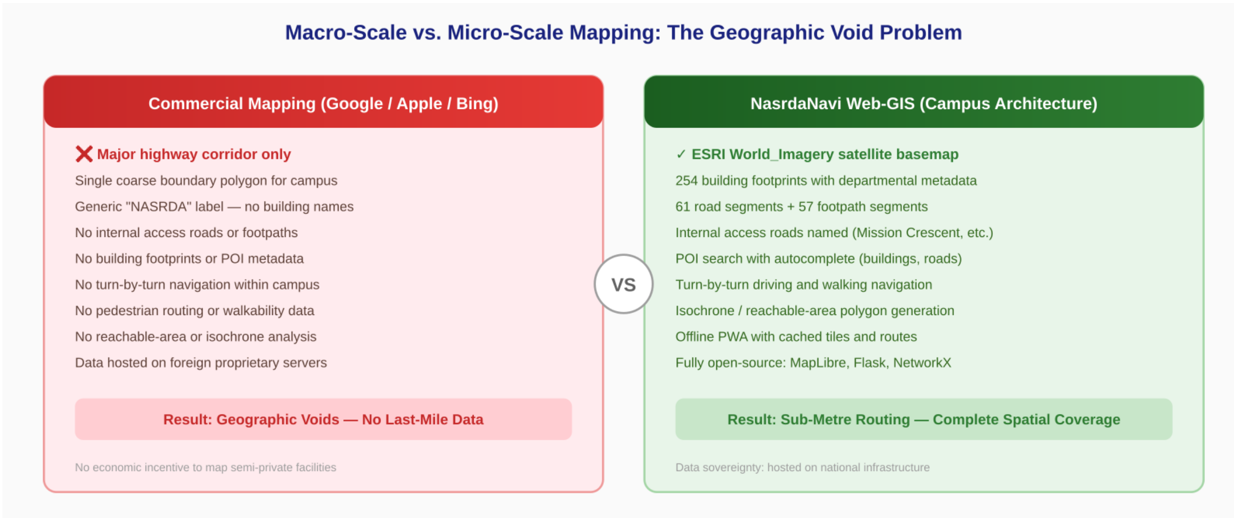

Figure 1: The geographic void: Macro-scale vs Micro-scale mapping

The National Space Research and Development Agency (NASRDA), like many national research institutions, operates from an expansive campus: the Obasanjo Space Centre in Abuja, Nigeria (8°59'22.8"N, 7°23'13.6"E). The absence of a localized, granular navigation application complicates logistics, hinders effective inter-departmental coordination, and impairs wayfinding for external visitors across the site. Building an in-house micro-scale campus Web-GIS navigation platform addresses a critical operational need by digitizing internal pathways, building footprints, and key points of interest. Furthermore, it serves as an active live laboratory that operationalizes NASRDA's space-to-earth mandate under the NASRDA Act, converting spatial data assets into an accessible, high-performance web application that demonstrates national technological capability and spatial data sovereignty.

This paper presents the development and deployment of *NasrdaNavi*, a Web-GIS navigation application for the NASRDA headquarters campus. It outlines the workflow from crowdsourced vector data extraction from OpenStreetMap (OSM), integration of NASRDA drone imagery with high-resolution ESRI World Imagery basemaps, and topological processing in QGIS. The study details the front-end rendering architecture, NetworkX routing engine, and operational migration from proprietary API dependencies to an open-source software stack. Together, the implementation demonstrates a scalable approach to campus navigation using open and hybrid geospatial technologies.

II. WEB MAPPING AND WEB MAP APPLICATIONS

A. Evolution and Definitions

The evolution of web mapping from static displays to dynamic, interactive geographic information systems represents a major development in cartography and geospatial technology. Veenendaal et al. (2017) categorized this development into distinct eras: early static web maps (1993–1998), interactive maps (1999–2004), Web 2.0 (2005–2009), and mobile and cloud-based web mapping (2010–present). This progression has changed how individuals, organizations, and governments access, visualize, and interact with geographic information.

A web map is a digital map accessible through a web browser or web-based application, allowing users to view, interact with, and analyse geographic data without desktop GIS software (Neumann, 2008). Sack (2017) defined web mapping as "the process of designing, implementing, generating, and delivering maps on the World Wide Web," emphasizing that web maps function as interactive tools for dynamic data exploration rather than digital copies of paper maps.

Web map applications are interactive platforms that enable users to analyze geographic information through standard browser software (Netek et al., 2020). Neumann (2008) categorized web maps into five primary types: static web maps, interactive web maps, animated web maps, collaborative web maps, and mobile web maps. Web map applications serve varied domains, including navigation, urban planning, disaster response, environmental monitoring, and public policy participation (Roth et al., 2021). These platforms have broadened access to spatial data, enabling non-specialists to interact with complex geospatial datasets (Longley et al., 2015).

B. Technical Architecture: Raster Pixels vs Client-Side Vector Tiling

The architecture of web mapping has undergone a fundamental transformation, shifting from server-side raster rendering to client-side vector tiling. This evolution has redefined performance, interactivity, and cartographic flexibility in Web-GIS applications (Antoniou et al., 2020; Netek et al., 2020).

The Raster Tile Era

Early Web-GIS architectures relied on server-centric raster tile pyramids or Web Map Services (WMS) where map images were pre-rendered on the server and delivered as static .png or .jpg tiles (OGC, 2021; Roth et al., 2014). While straightforward to implement, this approach imposed high network bandwidth consumption, fixed step zoom levels (e.g., $z=11$, $z=12$), limited styling flexibility, and pixelation or symbol overlap at non-native zoom scales (Netek et al., 2020). These constraints made raster-based systems inefficient for dynamic, interactive applications operating in resource-constrained environments.

The Vector Tiling Revolution

Modern web mapping has transitioned to client-side vector tiling, where raw geometric data and attributes are transmitted in compressed binary format (e.g., pbf) and rendered dynamically in the browser using WebGL hardware acceleration. This shift moves the rendering workload to the client device, enabling smooth fractional zooming (e.g., $z=11.25$), resolution-independent scaling, runtime cartographic styling, direct attribute interactivity (hover effects, popups, spatial queries), and a 70–80% reduction in bandwidth usage (Antoniou et al., 2020; Netek et al., 2020).

Table 1: Comparison of Web Map Delivery Architectures (adapted from Antoniou et al., 2020; Netek et al., 2020).

Feature	Server-Side Raster Tiles	Client-Side Vector Tiles
Data Format	Server-side rendered image grids (.png/.jpg)	Client-side rendered geometry (.pbf)
Zoom Levels	Fixed step zoom levels (z=11, z=12)	Continuous fractional zoom (z=11.25)
Bandwidth Usage	High bandwidth usage (re-downloads image grids per pan/zoom)	70-80% lower bandwidth consumption
Styling Capabilities	Static visual styling (requires full tile re-rendering)	Dynamic runtime cartographic styling
Examples	Legacy WMS, Mapy.cz, default OSM tiles	MapLibre GL JS, Mapbox GL JS, Google Maps

While OpenStreetMap (OSM) is often associated with raster rendering in its default browser interface, its underlying database consists of raw vector geometries that serve as the primary source for modern vector tile architectures (Haklay, 2010).

C. Components of Modern Web Maps

This architectural shift from raster to vector style rendering supports modular, interoperable Web-GIS applications. Modern web mapping stacks incorporate tile servers for spatial grid distribution, JavaScript client libraries (e.g., Leaflet, OpenLayers, MapLibre GL JS) for map rendering, geocoding APIs for address-coordinate conversion, routing engines for network pathfinding, and data APIs for feature attributes (Coetzee et al., 2020; Netek et al., 2020).

The Mapbox Vector Tile Specification

The Mapbox Vector Tile (MVT) specification provides an open, compact standard for vector tiling. MVT encodes geometries and key-value metadata attributes into Protocol Buffers (.pbf) indexed by $z/x/y$ tile coordinates (Mapbox, 2026). This standard provides significant operational advantages for institutional campus navigation systems, including reduced network payloads over limited connectivity networks, dynamic script-driven styling, and instant attribute interactivity on client devices (Netek et al., 2020).

D. Web Mapping Ecosystems: Tools, Platforms, and Design Philosophies

Selecting appropriate software components requires evaluating trade-offs between system control, licensing costs, rendering performance, and long-term maintainability, factors that directly influence institutional Web-GIS design (Longley et al., 2015; Sack, 2017).

Open-Source Client Libraries

Open-source JavaScript libraries form the foundation of custom Web-GIS applications, providing programmatic flexibility without vendor licensing restrictions (Roth et al., 2014). Leaflet and OpenLayers are established 2D mapping engines supporting standard OGC protocols (WMS, WFS) and raster tile delivery (Veenendaal et al., 2017). MapLibre GL JS, an open-source fork of Mapbox GL JS, uses WebGL hardware acceleration to render vector layers client-side (Netek et al., 2020). This enables dynamic runtime styling, feature-level interactivity, and smooth zoom scaling, making open-source libraries effective tools for institutional platforms where data sovereignty is required (Antoniou et al., 2020).

Enterprise Platforms

Enterprise platforms manage data storage, user access control, and service interoperability across organizations (Longley et al., 2015). Commercial systems like Esri ArcGIS Online supply turnkey cloud hosting and spatial analysis tools, but their proprietary licensing models can introduce recurring operational costs and limit source-code customization. Conversely, open-source platforms like GeoNode integrate OGC and INSPIRE metadata standards with spatial databases (such as PostGIS), allowing institutions to maintain direct operational control over data governance and service deployment (Coetzee et al., 2020).

3D and Specialized Visualization Engines

Specialized rendering engines extend web mapping beyond conventional 2D displays. CesiumJS provides globe-based 3D visualization, supporting terrain elevation models, satellite orbital trajectories, and spatial time-series data (CesiumJS, 2026). Libraries like Deck.gl use WebGL to process high-density spatial datasets (e.g., point clouds, heatmaps), while D3.js enables custom data-driven vector graphics (Roth et al., 2021). While these tools support advanced analytical workflows, they demand greater client-side compute resources and developer implementation effort.

Classification of Web Mapping Tools

Table 2: Functional Categorization of Web Mapping Software and Service Frameworks.

Category	Major Frameworks / Services	Operational Characteristics	System Trade-offs
Global Mapping Services (Rana & Joliveau 2009; Veenendaal et al. 2017)	Google Maps, HERE Maps	Hosted APIs providing basemaps, geocoding, and routing services.	High ease of use; restricted by usage rate limits, API costs, and closed data access.

Map Embedding Tools (Veenendaal et al. 2017)	Google Maps Embed API	Lightweight iframe/script frames for simple map displays.	Fast implementation; lacks programmatic query controls and raw feature access.
Developer APIs & SDKs (Netek et al. 2020; Roth et al. 2014)	Mapbox GL JS, CesiumJS	Programmable client-side engines for vector styling and 3D scenes.	High performance and design control; requires specialized software engineering.
Collaborative Data Platforms (Haklay 2010; Haklay & Weber 2008)	OpenStreetMap (OSM)	Crowdsourced spatial data engine with open database licensing.	Unrestricted feature vector access; requires automated geometry validation.
Language-Specific Libraries (Coetzee et al. 2020)	Folium (Python), Leaflet (R)	Programmatic bindings integrating web maps into data science pipelines.	Direct workflow integration; requires web server wrappers for broad public deployment.

E. Design Philosophies in Web Mapping

Beyond technical capabilities, web mapping platforms reflect contrasting approaches to data ownership, cartographic authority, and user interaction:

Commercial Cartography

Platforms such as Google Maps and Bing Maps emphasize interface usability, automated routing, and search indexing (Rana & Joliveau, 2009). However, their closed data models restrict style customization, obscure raw feature geometries, and centralize data control within private infrastructure (Neumann, 2008).

Community-Driven Mapping

Volunteered Geographic Information (VGI) from crowdsourced frameworks like OpenStreetMap permits direct editing by local users (Haklay & Weber, 2008). This decentralized approach captures fine-scale local details, such as informal walkways, rural road alignments, and local place names, that commercial providers frequently omit (Haklay, 2010). Maintaining topological consistency, however, requires systematic validation.

The Hybrid Architectural Model

For institutional systems like *NasrdaNavi,* combining these models offers a functional middle ground. Leveraging open-source rendering engines (MapLibre GL JS) alongside community-sourced base vectors (OSM) and validated agency datasets ensures data sovereignty, visual customization, and system longevity without reliance on proprietary cloud APIs (Antoniou et al., 2020; Netek et al., 2020).

F. Volunteered Geographic Information (VGI) and Data Quality Validation

The emergence of Volunteered Geographic Information (VGI), posited by Goodchild (2007), fundamentally altered spatial data production. OpenStreetMap (OSM) represents the primary crowdsourced geospatial database globally, populated by millions of contributors (Haklay, 2010; OpenStreetMap Contributors, n.d.-a).

While OSM provides detailed road coverage in major urban centres, research documents positional and completeness discrepancies in developing nations (Haklay, 2010; Sanni & Ahmed, 2023). In regions with smaller contributor communities, OSM datasets often present missing building footprints, unmapped secondary pathways, and disconnected line topologies that break automated routing algorithms (Haklay, 2010). To overcome VGI quality limitations, researchers recommend fusing crowdsourced vectors with authoritative high-resolution remote sensing assets (Neelakantappa et al., 2025; Sanni & Ahmed, 2023). High-resolution satellite imagery from the ESRI World Imagery basemap provides an accessible ground-truth layer against which crowdsourced vectors can be positionally validated and topologically snapped (Aziz & Omar, 2025). Furthermore, when public institutions contribute their corrected campus maps back to OSM, it generates mutual benefits for local facility management and open mapping communities (Jean-Daubias et al., 2024).

G. Spatial Data Infrastructure (SDI) and Data Sovereignty

Spatial Data Infrastructure (SDI) policies establish technical standards, administrative agreements, and data-sharing protocols necessary to leverage location data for sustainable development (Rajabifard & Williamson, 2002; Shah et al., 2025; United Nations ECA, 2023). In Nigeria, the National Geoinformation Policy mandates the development of an integrated NGDI to harmonize spatial datasets across government ministries, departments, and agencies (Kufoniyi & Akinyede, 2004; Warekuromor et al., 2017). Developing localized Web-GIS portals supports national spatial data infrastructure goals by establishing distributed, provenance-enabled data access protocols (Warekuromor et al., 2017). This bottom-up approach enables public institutions in space-faring nations to exercise spatial data sovereignty, ensuring critical institutional infrastructure maps remain under national administrative control rather than hosted exclusively on foreign proprietary platforms (Brovelli et al., 2021; Soulé, 2024; Warekuromor et al., 2017).

H. User-Centred Design and Cartographic Evaluation

Cartographic design choices directly influence user navigation efficiency and spatial understanding. Modern web architecture enables open data publishing, but accessible tools also mean maps can be published without adherence to cartographic design principles (Neumann, 2008). Common usability issues in web applications include unclear symbol hierarchies, poorly placed UI widgets, missing scale references, and inconsistent regional place-name labeling. In campus environments, these visual design flaws cause spatial disorientation and reduce system adoption rates.

Systematic evaluation methodologies assess these design variables. Skopeliti and Stamou (2019) benchmarked online map services across six fundamental cartographic dimensions: Map Key, Map Scale, Map Layout, Navigation Controls, Labelling, and Analytical Tools. Nivala et al. (2008) combined heuristic evaluations with usability testing to isolate interface logic errors and visual hierarchy gaps. Barvíř and Voženílek (2020) evaluated platforms based on graphic density, while Wang (2014) used empirical laboratory testing with 42 participants to identify layout design factors that optimize search speeds and map interaction efficiency. Applying a standardized evaluation framework provides a benchmark for testing custom Web-GIS installations before public deployment.

III. STUDY AREA AND METHODOLOGY

The study area covers the headquarters campus of the National Space Research and Development Agency (NASRDA), located at the Obasanjo Space Centre along the Umaru Musa Yar'Adua Expressway in Abuja, Nigeria (8°59'22.8"N, 7°23'13.6"E). As the primary hub of Nigeria's space program, the campus integrates administrative, research, technical, and public engagement functions within a multi-use environment. The site features a network of internal roads and pedestrian pathways connecting secure research zones, administrative offices, training facilities, research farms, and public spaces. This functional complexity, combined with restricted access zones and regular movement of staff, trainees, visitors, and international collaborators, presents a navigation environment representative of scientific and institutional campuses.

Several key facilities comprise the operational layout of the campus. The NASRDA Administrative Building houses the Office of the Director-General alongside core administrative units, including Finance, Legal Services, and Space Enterprise. Nearby, the Prof. Robert Ajayi Boroffice Multipurpose Conference Centre serves as the primary venue for national and international space policy dialogues, technical workshops, and institutional meetings. High-security technical installations include the Assembly, Integration, Testing and Design Centre (AITDC) and the Unmanned Aerial Vehicle (UAV) laboratory

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722297

which are dedicated to satellite and UAV engineering under controlled access protocols, and the Ground Receiving Station, equipped with large satellite dish arrays for orbital communication, telemetry tracking, and satellite data reception. Technical research is further supported by the Advanced UAV Laboratory, which conducts localized drone development and sensor integration for agricultural, environmental, and infrastructure monitoring.

Academic, public engagement, and data infrastructure functions are distributed across dedicated campus complexes. The Institute of Space Science and Engineering (ISSE) supports postgraduate training and advanced academic research in aerospace engineering and space sciences, while the National Space Museum and Planetarium provides public-facing educational outreach programs, school visits, and astronomy exhibits that require intuitive navigation for external visitors. The NGDI Building serves as the central repository hosting satellite data acquired over Nigeria in fulfilment of NASRDA's statutory mandate under the NASRDA Act. Notably, it houses the National Clearinghouse for the National Geospatial Data Infrastructure (NGDI) alongside the Mission Planning and Satellite Data Management Department, the Centre for Satellite Technology Development (CSTD), and the agency's ICT Department. Additionally, the Strategic Space Applications Building hosts the Strategic Space Applications Department together with specialized analytical units, including the ACRESAL Lab for remote sensing in land degradation and natural resource conservation, and the GRID3 Lab for high-resolution population distribution and health infrastructure spatial analytics. The coexistence of high-security technical facilities, analytical laboratories, policy centres, and public engagement spaces makes the NASRDA campus a representative testbed for *NasrdaNavi*, a custom Web-GIS navigation platform built using open-source libraries and client-side vector tiling architectures. By focusing on this site, the study evaluates how a lightweight, locally adaptable spatial navigation solution can address the operational wayfinding requirements of a modern scientific institution.

A. Data Collection

To construct a topologically connected campus basemap, a multi-source data pipeline was established integrating three primary data streams:

NASRDA Drone Imagery

High-resolution aerial imagery of the National Space Research and Development Agency (NASRDA) campus was acquired via an Unmanned Aerial Vehicle (UAV) survey. This dataset captured fine-scale spatial planimetric details, including distinct building footprints, internal pedestrian walkways, parking areas, green spaces and vegetative land cover, providing necessary visual coverage for detailed digitizing and feature extraction across the campus (See Figure 2).

Figure 1: Aerial image from NASRDA UAV Survey

ESRI World_Imagery Satellite Base Layer

To complement the drone survey, high-resolution satellite imagery covering the campus was sourced from the ESRI World_Imagery basemap, a publicly accessible tiled map service hosted on ArcGIS Online (server.arcgisonline.com). The World_Imagery layer provides sub-meter multispectral satellite coverage compiled from Maxar, Earthstar Geographics, and the GIS User Community, refreshed periodically. This served as an uncompromised ground-truth reference for validating vector geometries, resolving positional displacements, and digitizing missing campus infrastructure (See Figure 3).

Figure 3: ESRI World_Imagery satellite basemap covering the NASRDA Obasanjo Space Centre campus, serving as ground truth. Building footprints (254), road segments (61), and footpath networks (57) are rendered as vector overlays on sub-meter Maxar satellite image

OpenStreetMap (OSM) Vector Extraction

Baseline vector layers representing regional roads, major building outlines, and surrounding transportation corridors were extracted from OpenStreetMap using Overpass API queries. The raw vectors were saved as shapefiles for topological processing in desktop GIS software.

B. Methodology and System Architecture

Adapting the smart campus navigation methodology outlined by Neelakantappa et al. (2025), our technical architecture processes raw spatial inputs through a four-stage transformation pipeline (See Figure 4): Data Extraction, Geometric Cleansing/Vectorization, GeoJSON Serving and Graph Construction, and Client-Side Rendering.

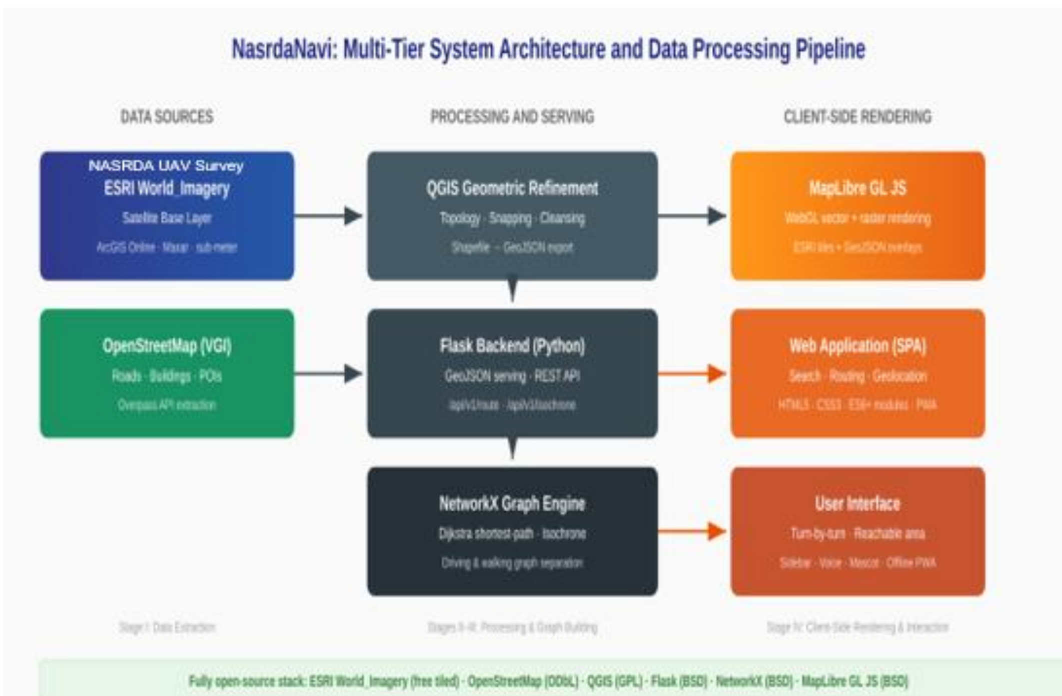

Figure 4: NasrdaNavi: Schematic Diagram of Multi-Tier System Architechture

Stage I: Geometric Refinement and Topology Construction

Raw VGI data derived from crowdsourced inputs often contains topological errors, such as dangling nodes, unclosed polygons, self-intersecting lines, and missing attribute key-values, that prevent proper spatial queries and routing (Haklay, 2010).

Vector layers were isolated in QGIS and projected to WGS 84 / UTM Zone 32N (EPSG:32632) to enable metric snapping and distance calculations. Feature layers were structured into Building Footprints (P_bld), Transit Corridors (E_road), and Points of Interest (N_poi). Using the ESRI World Imagery satellite basemap as a ground-truth backdrop, manual and automated snapping operations were conducted in QGIS. Disconnected road segments were snapped to common intersection vertices using a 0.1m tolerance threshold to construct a continuous topological graph.

To enable client-side searching, filtering, and dynamic popups, features were assigned a standardized property schema:

T = f_id, f_name, f_class, f_attr

Where each spatial feature T maps to explicit attribute fields:

- Buildings (P_bld): Indexed via unique identifiers *(f_id),* official facility names *(f_name),* structural classifications (*f_class*, e.g., administrative, research, educational), and parent facility codes (*fac_ref*) to support relational search queries.
- Road Networks (*E_road*): Segmented into directed edge segments containing street typologies (highway, service, footway), surface characteristics (paved, unpaved), and access rules (public, restricted, pedestrian).
- Points of Interest (*N_poi*): Represented as point nodes tagged with searchable metadata strings, department affiliations, and building entry associations (Delling & Werneck, 2013).

Stage II: GeoJSON Serving and Graph Construction

Once cleaned and verified in QGIS, the spatial layers were exported as GeoJSON files and served directly by a Flask REST API backend. Rather than requiring a proprietary cloud tile compilation step, the GeoJSON files are loaded as static vector sources by the MapLibre GL JS client, and the ESRI World_Imagery basemap is fetched as a remote raster tile layer from ArcGIS Online. The legacy vector tile specification (.pbf) was evaluated but ultimately replaced with direct GeoJSON serving, eliminating the need for a proprietary tile compilation step.

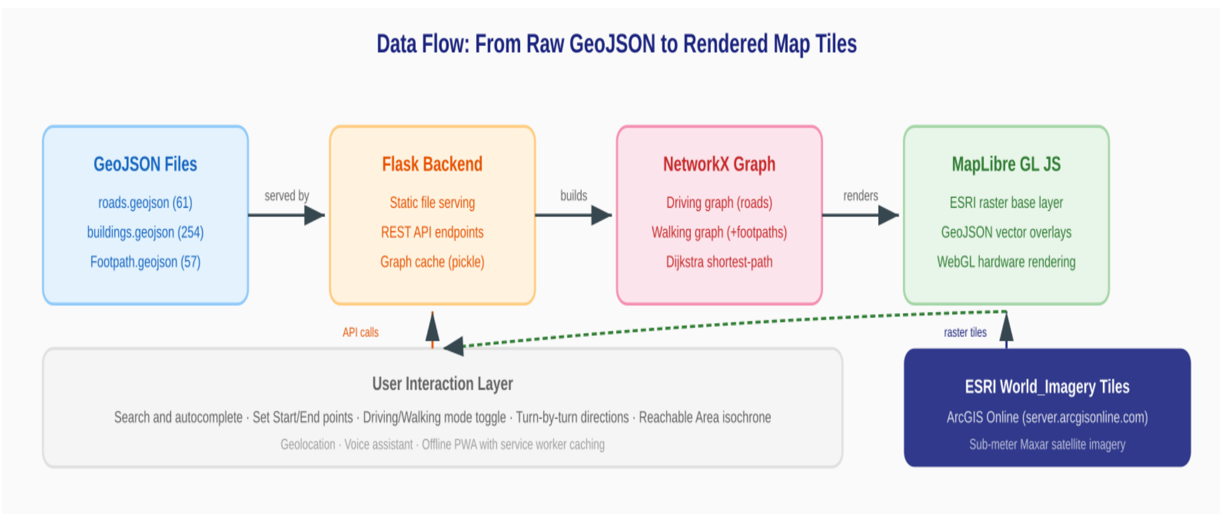

Figure 5: Schematic Diagram of the Data Flow

Cartographic stylesheets were defined client-side using MapLibre GL JS style specifications (see Figure 5), applying institutional colour schemes (NASRDA blue #002FA7, orange #FF8C00 and green #008751), visibility zoom thresholds (z=14 to z=18), and typographic hierarchies. The Flask backend concurrently builds a directed routing NetworkX graph *(G = V,E)* from the road and footpath GeoJSON data, caching it as a pickled graph object for fast subsequent route calculations.

Stage III: Front-End Web Client Architecture

The front-end user interface was built as a responsive Single Page Application (SPA) using HTML5, CSS3, JavaScript (ES6+), and MapLibre GL JS rendering library.

The system execution flow operates as follows:

i. Asynchronous Initial Load: The browser fetches the lightweight HTML/CSS layout and initializes the MapLibre GL JS map canvas pointing to the remote raster tile and GeoJSON layer endpoints.
ii. Real-Time Geolocation Tracking: Device positioning is managed by wrapping browser hardware requests inside a *MapLibre GL JS*

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722297

GeolocateControl framework, displaying user coordinates as an animated location marker relative to campus roads.

iii. Interactive Search & Autocomplete: An asynchronous search panel parses input strings against the local feature attribute index (f_name, f_class), providing real-time suggestions and zooming directly to selected POI coordinates.

iv. Pathfinding & Reachability Analysis: Server-side Python routing via NetworkX computes Dijkstra shortest-path routes across the campus road/footpath graph, while convex-hull isochrone analysis generates reachable-area polygons from origin points. The results are returned as GeoJSON payload feature collections to the client for rendering.

IV. SYSTEM IMPLEMENTATION AND RESULTS

A. System Implementation and Interface Deployment (NasrdaNavi)

The integrated Web-GIS platform, named *NasrdaNavi,* was successfully deployed in the agency's testing environment. The client interface renders the full campus layout (see Figure 6), showing building geometries, primary access roads, secondary footpaths, and points of interest superimposed over the ESRI World_Imagery satellite base layer.

Figure 6: The active NasrdaNavi Web-GIS application interface, showing the full campus layout with building footprints, road networks, green areas, and the ESRI World_Imagery satellite basemap over the Obasanjo Space Centre

The single-page application interface incorporates an intuitive navigation panel on the left, providing:

- Search bar with autocomplete functionality for buildings and roads.
- Navigation point selection controls ("Set Start" and "Set End").
- Mode selection toggles for "Driving" and "Walking" directions.
- Quick-action buttons to "Clear Route", "Find Nearest", and generate "Reachable Area" polygons.
- Map view navigation controls (zoom, tilt, compass orientation, and full-screen toggles).

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722297

Figure 7: NasrdaNavi driving route calculation showing the orange route line rendered over the ESRI satellite basemap, with turn-by-turn directions panel displaying road names and distances.

B. Cartographic Evaluation

Following the empirical evaluation framework established by Skopeliti and Stamou (2019), the *NasrdaNavi* campus Web-GIS application was systematically evaluated across six primary cartographic dimensions:

- Map Key (Map Legend): Evaluated based on the visual design and representational accuracy of campus feature symbology. Point features (Points of Interest), line strings (road and footpath networks), and polygon layers (building footprints) were rendered using high-contrast visual variables. Colour selection incorporated official institutional accents (NASRDA blue #002FA7, green #008751 for administrative and green areas; orange #FF8C00 for primary transit nodes and active routes) to ensure visual legibility against the satellite imagery backdrop.
- Map Scale: Assessed on the clarity and responsiveness of scale indicators across zoom interactions. The interface provides a dynamic graphic scale bar that updates continuously during fractional zooming (z=14 to z=18). Scale representation maintains legibility without visually cluttering the viewport during close-range campus navigation.
- Map Layout: Evaluated on structural balance, visual clutter, and screen space allocation. The single-page application layout reserves the primary viewport for the interactive map canvas, placing search, route planning, and layer control panels in collapsible side overlays to minimize distraction and maximize spatial context.
- Navigation Elements: Evaluated based on the responsiveness, graphic design, and placement of interactive UI widgets. On-screen controls include explicit zoom/pan buttons, a compass orientation indicator, an asynchronous search panel with autocomplete suggestions, and a *GeolocateControl* button for real-time device positioning relative to campus road networks.
- Labels: Evaluated across font selection, colour contrast, placement logic, and label collision management. MapLibre GL JS symbol layers were configured with zoom-dependent text visibility thresholds and collision avoidance parameters, ensuring building names and department labels scale cleanly without overlapping adjacent vector features.
- Analytical Tools: Assessed on the functional execution and performance of interactive spatial utilities. Key evaluated tools include server-side

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722297

Dijkstra shortest-path calculations (NetworkX), metric distance measurement, attribute query popups (T = f_id, f_name, f_class, f_attr), reachability isochrone polygon rendering, and basemap/layer visibility toggles.

In alignment with web usability criteria (Nivala et al., 2008; Skopeliti & Stamou, 2019), the evaluation verified a clear operational distinction between visual interface quality (aesthetic harmony, appropriate symbology contrast, and freedom from visual distractions) and functional performance (intuitive UI layout, low query latency, and reliable routing output).

C. Dynamic Routing and Reachable-Area Polygon Generation

The application executes client-side spatial operations without requiring server round-trips. Users can click on any building polygon or input origin/destination query strings to render turn-by-turn paths across internal network edges.

Figure 8: NasrdaNavi reachable-area analysis showing a semi-transparent polygon overlay delineating the geographic zone reachable within a specified walking or driving radius from a selected origin point.

The system provides two core analytical capabilities:

i. Dynamic Path Calculation: Shortest-path trajectories are calculated along valid graph edges, displaying a high-contrast orange route line connecting origin and destination markers accompanied by turn-by-turn directions.
ii. Reachable Area Overlays: Prompting the "Reachable Area" tool generates a semi-transparent directional polygon overlay, visually bounding the geographic zone reachable within a defined walking or driving time radius from a selected origin node.

D. Vector Performance and Rendering Responsiveness

Live user interface testing confirmed that client-side GeoJSON rendering via MapLibre GL JS eliminates the buffering delays common to legacy Web-GIS platforms operating on institutional networks. Because campus features are stored as vector GeoJSON and the ESRI base layer uses cached raster tiles, both rendered via browser WebGL hardware acceleration, users can pan, rotate, tilt, and zoom across zoom levels (z=14 to z=18) without resolution degradation or map image pixelation.

E. System Performance Evaluation

To validate the system's operational viability, key performance metrics were measured across routing and reachability operations (Table 1). All computations were performed server-side via the Flask/NetworkX backend, with results returned as GeoJSON to the MapLibre GL JS client.

Table 3: Performance metrics for routing and isochrone operations. Computation times include graph loading from pickle cache and Dijkstra traversal.

Operation	Distance	ETA	Steps	Compute Time	Nodes
Driving route	1,156 m	139 s	7	~8 ms	-
Walking route	1,058 m	755 s	15	~10 ms	-
Isochrone (5 min driving)	-	5 min	-	~14 ms	396
Isochrone (10 min walking)	-	10 min	-	~14 ms	418

The routing engine constructs separate driving and walking graphs from the campus GeoJSON data. The driving graph incorporates only road segments, while the walking graph additionally includes footpath segments with footpath-to-road bridging within a 50-meter threshold to resolve pedestrian network gaps. Shortest-path computation uses Dijkstra's algorithm via NetworkX, with edge weights derived from the Haversine distance formula. The isochrone (reachable-area) function computes all graph nodes reachable within a specified travel time using single-source Dijkstra path lengths, then generates a convex hull polygon from the reachable node coordinates using *Shapely*. All routing API responses are returned as *GeoJSON FeatureCollections*, enabling seamless client-side rendering via MapLibre GL JS.

F. Platform Migration: Transitioning from Proprietary to Open-Source

The initial architectural prototype relied extensively on the proprietary Mapbox cloud ecosystem for vector tile hosting and style management. During development, however, the implementation team encountered operational constraints, including functional paywalls and aggressive subscription rate limits imposed on commercial API access keys. These fiscal and structural barriers prompted a strategic migration to a fully open-source software stack. Specifically, MapLibre GL JS replaced Mapbox GL JS for client-side rendering, Esri World_Imagery replaced Mapbox satellite tiles for basemap visualization, and a self-hosted Flask framework integrated with NetworkX replaced Mapbox Studio for backend graph processing and routing-tile compilation.

For public institutions and research agencies operating within developing economies, transitioning to an open-source stack effectively mitigates subscription-based financial risks and prevents restrictive vendor lock-in. The current NasrdaNavi implementation is entirely self-hosted using MapLibre GL JS, Flask, and NetworkX, eliminating external proprietary API dependencies. Future iterations may scale this architecture further by adopting PostGIS spatial databases and pg_tileserv for high-volume enterprise deployments (Coetzee et al., 2020; Brovelli et al., 2021; Çardak et al., 2026).

V. CONCLUSION AND FUTURE WORK

The *NasrdaNavi* campus navigation project demonstrates that high-precision, localized Web-GIS applications can be successfully engineered by integrating bespoke, locally generated spatial data (NASRDA drone survey) with existing Volunteered Geographic Information (VGI) frameworks, specifically OpenStreetMap (OSM) vectors and the Esri World_Imagery satellite basemap, leveraging modern open-source web-mapping technologies. The deployed system effectively bridges a critical operational navigation gap that commercial web-mapping providers routinely overlook, delivering granular turn-by-turn routing, point-of-interest (POI) queries, and localized reachability analysis across the NASRDA Obasanjo Space Centre headquarters campus.

To advance this web-based proof of concept into an enterprise-grade smart campus management framework, the project roadmap targets four sequential development phases (See Figure 9)

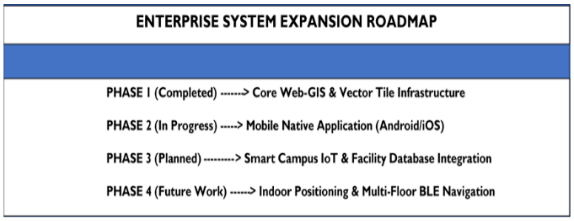

Figure 9: Phased Roadmap for NasrdaNavi Scale-up

Phase 1 (Completed) - Core Infrastructure & Web Deployment

This foundational phase achieved the validation of web-based client interface components, cartographic style specifications, satellite base layers, and localized vector dataset schemas to establish a stable deployment environment.

Phase 2 (In Progress): Native Mobile Application Development

This phase focuses on compiling native Android and iOS mobile application builds using open-source mobile software development kits (SDKs). These builds leverage low-level hardware sensors, specifically the device magnetometer, gyroscope, and fused GNSS telemetry, to execute robust location smoothing and facilitate offline vector tile caching.

Phase 3 (Planned): Smart Campus IoT & Facility Integration

This phase will integrate planimetric building polygons with internal agency databases, enabling real-time room scheduling queries, automated facility maintenance tracking, mobility analytics, and environmental sensor monitoring (Mokbel et al., 2024; Haque et al., 2024).

Phase 4 (Future Work): Indoor Navigation Infrastructure

Long-term objectives seek to expand system capabilities into vertical, multi-floor positioning by deploying Bluetooth Low Energy (BLE) beacons or Wi-Fi fingerprinting arrays to guide visitors through internal building corridors (Philippopoulos et al., 2024)

Ultimately, this project serves as a replicable model for how national space programs in developing nations can successfully deploy practical, user-centric geospatial applications. By doing so, public institutions can systematically enhance operational efficiency, exercise absolute spatial data sovereignty, and demonstrate applied technological capability on a global scale.

REFERENCES

[1] Antoniou, V., Ragia, L., & Nomikou, P. (2020). Vector tiles vs. raster tiles for dynamic web mapping: Performance analysis. *Geo-spatial Information Science*, 23(4), 312–326. https://doi.org/10.1080/10095020.2020.1798361

[2] Aziz, F. A. A., & Omar, N. (2025). Development of a web-based GIS application for Politeknik Ungku Omar using QGIS and Leaflet. *Jurnal Kejuruteraan, Teknologi dan Sains Sosial*, 11(2), 160–169.

[3] Barvíř, R., & Voženílek, V. (2020). Graphic density of web-mapping platforms: Evaluation of cartographic content. *ISPRS International Journal of Geo-Information*, 9(11), Article 665. https://doi.org/10.3390/ijgi9110665

[4] Brovelli, M. A., Li, K. J., & Eom, K. S. (2021). Moving toward open geospatial systems: The UN Open GIS initiative. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2021, 183–188. https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-183-2021

[5] Çardak, G., Topcu, İ., & Anbaroğlu, B. (2026). An open source framework for routing and event management in university campuses. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XI-5-2026, 25–33.

[6] CesiumJS. (2026). *CesiumJS 3D geospatial visualization platform*. Cesium. https://cesium.com/platform/cesiumjs/

[7] Coetzee, S., Ivánová, I., Sititham, P., & Rautenbach, V. (2020). Open source software for spatial data infrastructures: A systematic review. *ISPRS International Journal of Geo-*

Information, 9(11), Article 680. https://doi.org/10.3390/ijgi9110680

[8] Delling, D., & Werneck, R. F. (2013). *Customizable point-of-interest queries in road networks* (Technical Report No. MSR-TR-2013-90). Microsoft Research.

[9] Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geographic information. *GeoJournal*, 69(4), 211–221. https://doi.org/10.1007/s10708-007-9111-y

[10] Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design*, 37(4), 682–703. https://doi.org/10.1068/b35145

[11] Haklay, M., & Weber, P. (2008). OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4), 12–18. https://doi.org/10.1109/MPRV.2008.80

[12] Haque, R., Razzak, M. A., & Zeleti, F. A. (2024). *Advancing sustainable transport in Ireland* (MaaS4IRL Project Report). Sustainable Energy Authority of Ireland.

[13] Jean-Daubias, S., et al. (2024). Use of OSM for the OPALE campus application: Mutual benefits and future developments. *HAL Open Science*, Article 04512984.

[14] Kufoniyi, O., & Akinyede, J. O. (2004). Mainstreaming geospatial information for sustainable national development in Nigeria. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(B4), 7–13.

[15] Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic information science and systems* (4th ed.). John Wiley & Sons.

[16] Mapbox. (2026). *Vector tile specification*. Mapbox Documentation. https://docs.mapbox.com/vector-tiles/specification/

[17] Mokbel, M. F., Sakr, M., Xiong, L., Züfle, A., Yuan, J., Jensen, C. S., Magdy, A., Shekhar, S., & Zeitouni, K. (2024). Mobility data science: Perspectives and challenges. *ACM Transactions on Spatial Algorithms and Systems*, 10(2), Article 8. https://doi.org/10.1145/3652158

[18] Neelakantappa, B. B., Madhura, S., Meghana, C. L., & Mandara, H. P. (2025). Smart campus navigation system. *International Journal for Multidisciplinary Research*, 7(1), Article 34970.

[19] Netek, R., Masopust, J., Pawinger, C., & Voženílek, V. (2020). Performance testing on vector tile map rendering in web browsers. *ISPRS International Journal of Geo-Information*, 9(10), Article 620. https://doi.org/10.3390/ijgi9100620

[20] Neumann, B. (2008). Web mapping and WebCartography. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of database systems* (pp. 3521–3526). Springer.

[21] Nivala, A.-M., Brewster, S., & Sarjakoski, L. T. (2008). Usability evaluation of web mapping sites. *The Cartographic Journal*, 45(2), 129–138. https://doi.org/10.1179/174327708X307585

[22] Open Geospatial Consortium. (2021). *OpenGIS Web Map Service (WMS) implementation specification* (Version 1.3.0). OGC. https://www.ogc.org/standards/wms

[23] OpenStreetMap Contributors. (n.d.-a). *Copyright and license*. OpenStreetMap. Retrieved August 9, 2026, from https://www.openstreetmap.org/copyright

[24] Philippopoulos, P. I., Drivas, I. C., Tselikas, N. D., Koutrakis, K. N., Melidi, E., & Kous, D. (2024). A holistic approach for enhancing museum performance and visitor experience. *Sensors*, 24(3), Article 966. https://doi.org/10.3390/s24030966

[25] Rajabifard, A., & Williamson, I. P. (2002). Spatial Data Infrastructures: An initiative to facilitate spatial data sharing. *Global Environmental Databases*, 2, 107–115.

[26] Rana, S., & Joliveau, T. (2009). NeoGeography: An extension of mainstream Web 2.0 or a new scientific paradigm for Geographic Information Science? *Journal of Location Based Services*, 3(2), 75–81. https://doi.org/10.1080/17489720903130006

[27] Roth, R. E., Donohue, R. G., Sack, C. M., Wallace, T. R., & Buckingham, O. A. (2014).

Process design for web mapping: Applications and lessons from the Penn State Online Master's in GIS program. *Cartographic Perspectives*, (78), 31–47. https://doi.org/10.14714/CP78.1274

[28] Roth, R. E., Sack, C. M., Baldrica-Franklin, G., Underwood, N., & Donohue, R. G. (2021). Web mapping: A workbook for interactive cartography and visualization on the Open Web. *Abstracts of the ICA*, 3, Article 248. https://doi.org/10.5194/ica-abs-3-248-2021

[29] Sack, C. M. (2017). Web mapping. *The Geographic Information Science & Technology Body of Knowledge* (3rd Qtr 2017 ed.). University Consortium for Geographic Information Science. https://doi.org/10.22224/gistbok/2017.3.11

[30] Sanni, O. J., & Ahmed, M. B. (2023). Geo-spatial modeling and campus web-GIS development: A paradigm shift for smart infrastructure in Nigerian universities. *African Journal of Geo-Informatics*, 7(2), 114–128.

[31] Shah, P., Patel, J., Adhvaryu, J., & Prajapati, J. (2025). Spatial Data Infrastructure for Sustainable Development: Evaluating PM Gati Shakti as India's geospatial framework. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 571–576.

[32] Skopeliti, A., & Stamou, L. (2019). Quality assessment of web map basemaps: A cartographic evaluation. *Proceedings of the International Cartographic Association*, 2, Article 120. https://doi.org/10.5194/ica-proc-2-120-2019

[33] Soulé, F. (2024). *Negotiating Africa's digital partnerships amid geopolitical competition* (Special Report). Centre for International Governance Innovation.

[34] United Nations Economic Commission for Africa. (2023). *Global geospatial information management in Africa: Financial appropriation framework and the economic benefits of geospatial information in Africa*. United Nations Digital Repository. https://hdl.handle.net/10855/50655

[35] Veenendaal, B., Brovelli, M. A., & Li, S. (2017). Review of web mapping: Eras, trends and directions. *ISPRS International Journal of Geo-Information*, 6(10), Article 317. https://doi.org/10.3390/ijgi6100317

[36] Wang, L. (2014). Usability and user experience testing of interactive web mapping interfaces. *Computers, Environment and Urban Systems*, 45, 112–123. https://doi.org/10.1016/j.compenvurbsys.2014.03.001

[37] Warekuromor, T., James, A., Anifowose, B., & Trodd, N. (2017). A distributed, scalable and provenance-enabled data access protocol for spatial data infrastructure. In *Proceedings of the 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 180–185). IEEE. https://doi.org/10.1109/CSCWD.2017.8066691