Vol. 06, No. 01, pp. 25 –35 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

Design and Implementation of A Scalable and Efficient Geo-Portal System for Geospatial Data Management

A. O. Oyeyode*[1], T. O. Idowu[2], D. N. Olayinka-Dosumu[1], I. U. Nzelibe[2]

[1]*Department of Surveying and Geoinformatics, Federal School of Surveying, Oyo, Nigeria,* oyeyode@fss-oyo.edu.ng, dsaka@unilag.edu.ng

[2]*Department of Surveying and Geoinformatics, Federal University of Technology, Akure, Nigeria,* toidowu@futa.edu.ng, iunzelibe@futa.edu.ng

Abstract

Efficient geospatial data management is crucial for urban planning, land administration, and infrastructure development. Conventional static databases often lack real-time access and advanced spatial processing capabilities. This study aims to design and develop a scalable and efficient geo-portal system for managing second-order geodetic control points' geospatial data in Ibadan, Nigeria. The system integrates MongoDB for fast spatial data querying, Node.js and Express.js for server-side operations, and Leaflet.js for interactive mapping. It supports real-time data access, spatial indexing, and on-the-fly coordinate transformation between WGS84 and Minna Datum. Performance testing demonstrated an average response time of 200ms and a 92% user satisfaction rate, indicating system efficiency and usability. The developed geo-portal enhances access to geospatial information, supporting better decision-making in urban planning and land administration. Future improvements include predictive spatial analytics and integration with commercial geospatial platforms.

Keywords: *Geo-portal, Geospatial Data, MongoDB, Coordinate Transformation, urban planning*

Received: January 14th, 2025 / Accepted: March 5th, 2025 / Online: March 12th, 2025

I. INTRODUCTION

Geospatial information is an important asset in urban planning, environmental management, and infrastructure development, particularly in developing regions. With the growth of cities, there is an increasing need for reliable geospatial information [1]. Urban environments rely heavily on available data for proper decision-making and service delivery. Nonetheless, traditional approaches, such as paper-based record-keeping and non-networked databases, are insufficient to address the challenges of modern urbanization [2].

Geospatial data must be accurate and precise, as they also provide location information for mapping, surveying, and spatial analysis. For instance, in Nigeria, second-order geodetic control points form the backbone of a coordinate reference system for many geospatial datasets [3]. However, cities such as Ibadan are greatly hampered in the retrieval and maintenance of these control points due to poor infrastructure, data fragmentation, and restricted access [4]. [5] pointed out that National Spatial Data Infrastructures (NSDIs) have been established in developed countries as a means to centralize access to geospatial information. However, such structures do not exist in Nigeria.

Despite the increased demand for seamless geospatial accessibility, developing countries like Nigeria have yet to offer a geo-portal system with regard to second-order geodetic control points in light of the inefficiencies in urban development and decision making. Most existing systems in developed nations have been revised to include real-time data services and interoperability combined with user-controlled functions [6]. However, this development has only taken place in countries with advanced spatial data infrastructure; in developing countries such as Nigeria, geospatial data systems are outdated and fragmented.

Therefore, this paper seeks to fill this gap by developing a scalable and efficient geo-portal that will be able to respond to the geospatial needs of Ibadan the geospatial needs. The system uses MongoDB Server Version 7.0 for efficient data storage, allowing for high-performance geospatial queries and real-time data retrieval. Unlike other relational databases, MongoDB offers better scalability and spatial indexing and is therefore suitable for managing dynamic and large-scale geospatial datasets. It combines open-source technologies, such as Node.js, Express.js, and Leaflet, into a geoportal that offers the user an interactive, friendly, and efficient way to access and manage second-order geodetic control points.

Unlike previous studies that focused on static geospatial data storage [7], this study introduced an interactive, real-time platform tailored to Nigeria's geospatial data challenges. The implementation of this approach in other regions, such as the Land Information New Zealand (LINZ) platform, has demonstrated the potential of geo-portals to enhance data accessibility, accuracy, and user engagement [8]. Similarly, the project identified as Infrastructure for Spatial Information in the European Community-INSPIRE- has shown how standard interoperable geospatial platforms can support urban planning and development, among others [9]. However, these initiatives were developed within structured Spatial Data Infrastructure frameworks that are currently not in place in Nigeria.

Therefore, this study proposes a cost-effective, highly scalable geo-portal solution to bridge this gap. By integrating user-driven functionality and leveraging modern database systems, geo-portals support professionals such as surveyors, urban planners, and environmental scientists in their applications for land management, infrastructure development, and environmental conservation, setting a new level of geospatial data access in Nigeria.

II. MATERIALS AND METHODS

The four major activities discussed here are the study area, data preparation, system design and development, and application features.

A. Study Area

The capital of Oyo State, Ibadan, is Nigeria's third-largest city, spanning 3,080 km², with over six million people [10]. As shown in Fig. 1, Ibadan consists of 11 local government areas (LGAs) and serves as a key economic and infrastructural hub. Rapid urbanization has intensified the need for reliable geospatial data; however, the absence of a structured Spatial Data Infrastructure (SDI) has led to fragmented and inaccessible control point records.

The topography of the city, which ranges between 150 m and 275 m above sea level, the widening of roads, and Ibadan Inland Dry Port development projects demonstrate a growing demand for operational handling of geospatial data. Therefore, in light of these findings, research was conducted toward improving access to data supporting urban planning and decision-making processes within Ibadan using 1,958 second-order control points with an integrated geoportal system.

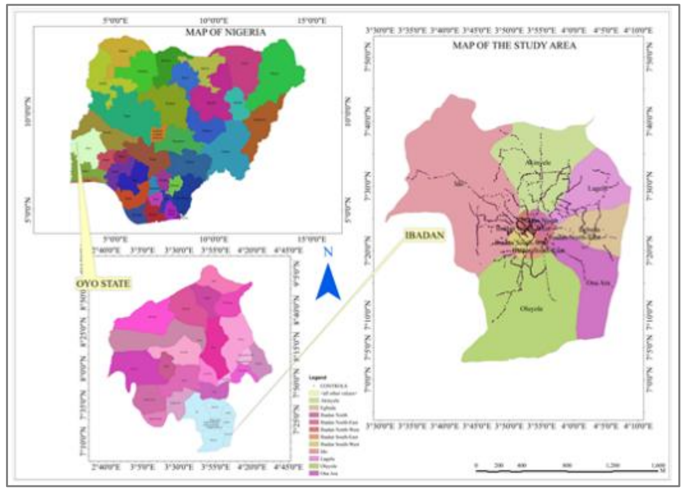

Fig. 1. Map of the Study Area

B. Data Preparation

The datasets in Table 1 comprise 1,958 second-order control points in geodetic coordinates, Universal Transverse Mercator (UTM) projection, and Cartesian coordinate system, all referenced to the WGS84 datum as obtained from The Office of the Surveyor General of Oyo State. These control points established a fundamental geodetic reference network for a myriad of geospatial applications in Ibadan.

For accuracy, the dataset was cross-validated with available geodetic infrastructure records for positional integrity. Inconsistencies in the metadata (e.g., lack of attributes) were adjusted for consistency in the data [4]. The validation guarantees that the dataset satisfies the precision thresholds for survey and mapmaking uses [1].

TABLE I. SAMPLE OF THE CONTROL POINTS COORDINATES IN GEODETIC (LAT., LONG., ELEV.), UTM ZONE 31 (E, N, H) & CARTESIAN SYSTEM (X, Y, Z) IN WGS 84 DATUM

Pillar No.	Lat.	Long	Elev.	E(m)	N(m)	Elev.	X	Y	Z
ICS 1001T	7.47 1014	3.914 698	256. 513	6009 28.805	8259 23.165	256. 513	63098 47.165	4317 88.428	8238 42.996
ICS 1002T	7.47 0888	3.916 344	264. 400	6011 10.521	8259 09.595	264. 400	63098 44.366	4319 70.407	8238 30.188
ICS 1003T	7.47 1116	3.918 673	266. 070	6013 67.460	8259 35.327	266. 070	63098 25.186	4322 26.768	8238 55.396
ICS 1004T	7.47 2081	3.920 168	272. 446	6015 32.204	8260 42.303	272. 446	63098 06.375	4323 90.885	8239 61.985
ICS 1005T	7.47 3745	3.922 580	263. 800	6017 97.988	8262 26.858	263. 800	63097 55.732	4326 54.291	8241 43.354
ICS 1006T	7.47 6784	3.923 321	255. 018	6018 78.986	8265 63.072	255. 018	63096 97.824	4327 32.240	8244 75.503

Source: - Field Work

Conversion of the control point coordinates to the GeoJSON format, as needed for MongoDB integration, was accomplished with the help of QGIS software. This was carried out by plotting the Eastings (E), Northings (N), and Heights (H) of the control points, maintaining data integrity, and exporting them to the necessary format, as shown in Fig. 2 and Fig. 3.

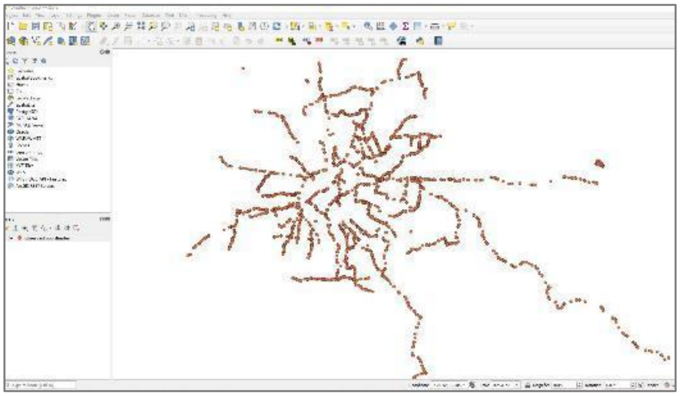

Fig. 2. Plotted coordinates in QGIS

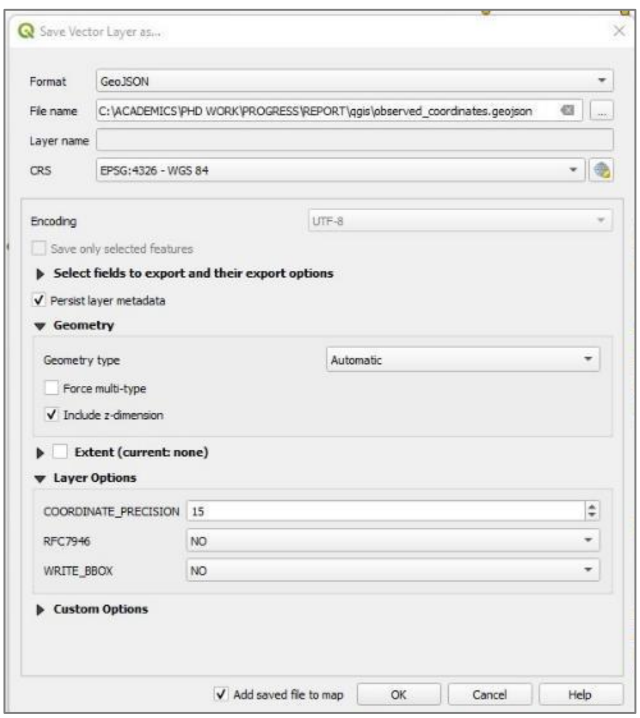

Fig. 3. Exporting to geoJSON file format

1) Survey Questions and User Responses

The geo-portal system was created based on the results of formal user research among prospective end users. A questionnaire consisting of 10 questions was administered to 30 respondents, comprising 10 lecturers, 10 practicing surveyors, and 10 officials from the Ministry of Land and Surveys. The participants were chosen to provide an equal balance of academic, professional, and governmental input. The answers were gathered and examined to obtain important information on the desired display formats, mobile friendliness, and necessary tool functionalities, as presented in Table II and Figure 4. While the sample provides valuable insights, we recognize that a larger sample would make the findings.

TABLE II. SUMMARY OF QUESTIONNAIRE RESULT

Survey Aspect	User Feedback	Implication/Action Taken
Frequency of Access	70% of users required daily access to control point data.	Underscored the efficient retrieval of data and performance stability through MongoDB's strong back-end data management.
Display Preferences	80% of users chose clustered markers for seamless navigation.	Employed leaflet.markercluster to decongest the screen and improve usability.
Mobile Compatibility	90% of users valued mobile compatibility as crucial.	A mobile-responsive design was adopted for easy access across devices
Find Nearest Control Point Feature	85% of users found this feature crucial for fieldwork efficiency.	incorporated a location-based recommendation key for swift access to close control points.
Datum Transformation	75% required transformation capabilities across datums (e.g., Minna Datum).	Coordinate transformation tool was introduced to permit conversions between datums

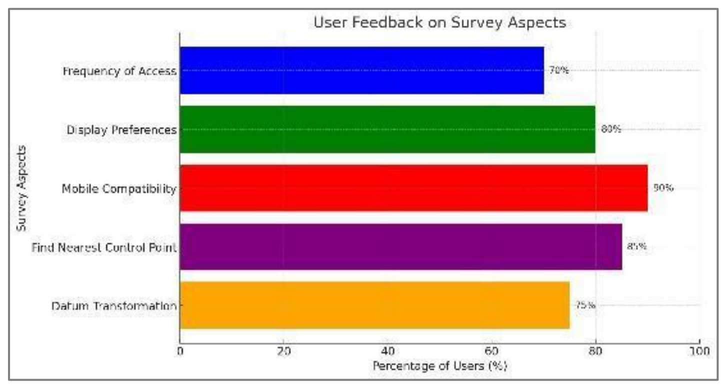

Fig. 4. Visual Representation of Questionnaire Results

C. System Design and Development

The geo-portal system was developed using the Model-View-Controller (MVC) framework, a popular and widely used architectural pattern to develop modularity, maintainability, and scalability in software design [11]. The framework simplifies keeping data management, request handling, and the user interface distinct and facilitates independent development and debugging of the respective components [12].

The Model layer is embodied in the MongoDB database (version 7.0), which is organized based on a document-oriented schema that is optimized for geospatial data. In contrast to conventional relational databases, MongoDB offers an effective means of storing and querying spatial data through its NoSQL design, thereby being sufficiently capable of accommodating dynamic and large-scale geospatial datasets [13]. Database operations are also augmented using Mongoose, an Object Data

Modeling (ODM) library, to enforce schema validation as well as optimized data manipulation [14].

The View layer, realized through Leaflet.js, JavaScript, HTML, and CSS, offers an interactive map-based interface with spatial queries, coordinate transformations, and nearest control point identification features. Responsive web design facilitates accessibility across various devices, thus catering to usability and adaptability issues [15].

The Controller layer, which was built using Node.js and Express.js, served as the interface between the frontend and backend, handling data requests and geospatial query calculations. The RESTful API endpoints of the system were built to adhere to best practices in geospatial data exchange [16], where the system supports the following:

i. Fetching control point data according to parameters specified by the user.

ii. Performing spatial queries to fetch the closest control point.

iii. iii. Perform data transformations, such as WGS84 and Minna Datum conversions [17].

iv. Management of user authentication and access control.

The adoption of the MVC framework, as opposed to other architectural models, was motivated by the fact that it would enforce a strict separation of concerns, thereby making software more maintainable and lowering system complexity [18]. Furthermore, with modular development, additional features, such as predictive spatial analysis and real-time data validation, can easily be added in the future, enhancing the long-term scalability of the geo-portal system. Fig. 5 shows a flowchart of the system architecture.

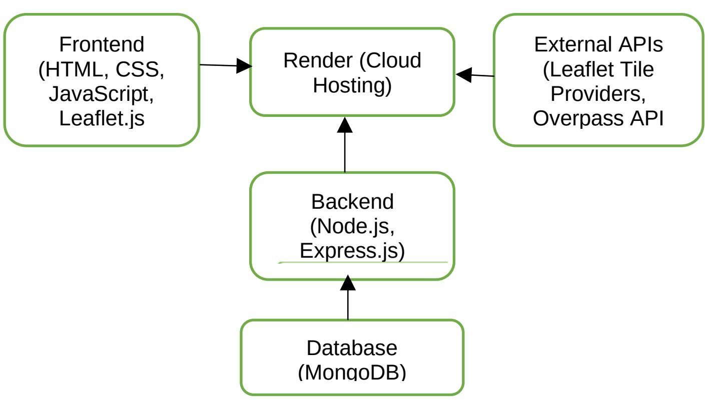

Fig. 5. System Architecture Flowchart

1) Backend

The backend was designed to handle server-side processing and enable easy interaction between the client side and the database by adopting Node.js and Express.js. It ensures the effective handling of requests from users by managing tasks such as authentication, geospatial queries, and data retrieval. The RESTful API endpoints presented suitable interfaces for relating to the system, such as extracting control point data or querying nearby points based on geographic coordinates.

2) Frontend

In a bid to provide users with an intuitive and dynamic interface for relating to geospatial data, the front end was built with Leaflet.js and JavaScript. The platform contains interactive maps permitting users to zoom, pan, and toggles between visualization layers, which are satellite and street views. With slight effort, users can retrieve control-point information (coordinates) directly from the map or execute tasks such as coordinate transformations.

Compatibility across several devices was ensured through a responsive design from desktops to mobile phones, thereby making accessibility possible for diverse user bases. A simple, visually appealing, and functional interface that meets the varied needs of surveyors, urban planners, and other professionals was ensured by leveraging a user-centered design.

3) Transformation from WGS84 to Minna Datum: Process and Challenges

The geo-portal was developed to enable the transformation of coordinates from the WGS84 datum (EPSG: 4326) to the Minna datum (EPSG: 4263) for general use in Nigeria's geospatial practice. This makes it compatible with the national geodetic infrastructure, which is beneficial for surveying, mapping, and land administration practitioners [19]. The use of the Minna datum plays an important role in accurate spatial referencing in Nigeria because of the country's use of local control networks [20].

The transformation was performed using Molodensky–Badekas, a ten-parameter transformation technique that is world-recommended and considers the difference between translation, rotation, and scale between WGS84 and Minna datum [21]. The transformation parameters were computed from official geodetic records and applied using geospatial libraries, such as PostGIS and PROJ [20]. The validity of the transformed coordinates was checked by comparison with known Minna datum control points, and Root Mean Square Error (RMSE) analysis was performed to determine the positional accuracy; however, the result was found to be within the allowable limit of second-order accuracy.

A particular challenge faced during implementation was the complexity of managing the coordinate format differences between various datasets. Most legacy datasets in Nigeria have coordinates stored in degrees, minutes, and seconds (DMS) format, whereas most contemporary GIS applications and databases store coordinates in decimal degrees (DD) format. This lack of consistency routinely causes data entry and interpretation errors, and necessitates automated conversion processes to normalize all coordinates to decimal degrees prior to processing [20]. This consistency enhances the accuracy of the transformations and avoids misalignment problems when combining several datasets [21].

Additionally, the transformation of coordinate data into different formats, such as GeoJSON, SHP, and CSV, raised issues about losing accuracy with the removal of decimal places [20]. This ensured that the saved coordinates had at least six decimal places to maintain accuracy through transformation and further data collection [21].

4) Database Creation and Implementation

The geo-portal system database was implemented using MongoDB (version 7.0), a NoSQL document-oriented database designed for high-performance geospatial applications [13]. The 2dsphere index of MongoDB was utilized to support sophisticated spatial queries so that the system could retrieve control point data rapidly, determine adjacent geodetic stations, and perform region-based queries [22]). Database creation is illustrated in Fig. 6 and Fig. 7, while Fig. 8 and Fig. 9 illustrate geospatial indexing in action.

With increasing data size and query frequency, the system performance is of utmost importance. To alleviate the scalability limitations, various precautions were taken. Sharding was employed to split the data across various servers in such a way that performance bottlenecks were avoided and queries were still efficient despite an increase in dataset size (Banker, 2016). Replication has also been implemented to provide improved fault tolerance and high availability, such that access is uninterrupted even when servers fail [22].

To continue enhancing query performance, caching techniques have been employed to hold data accessed regularly to minimize database load and enhance response time [23]. By employing aggregation pipelines, the execution of queries was optimized, minimizing computational overhead while enabling efficient data retrieval. Horizontal scaling techniques were also employed by the system, taking advantage of MongoDB's distributed properties to handle the increased demand for data [24].

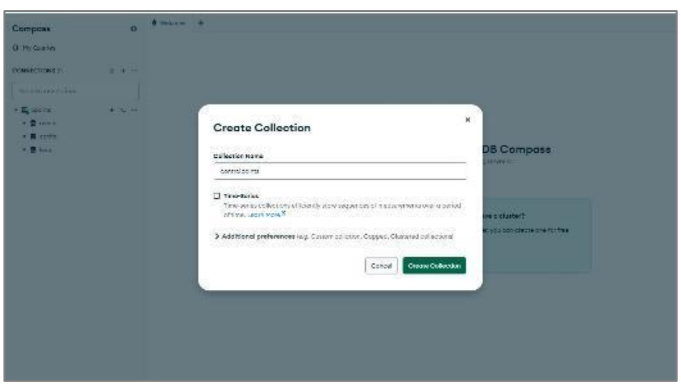

Fig. 6. Creation of Data Collection

Performance benchmarks, as reported in Fig. 10 and Tables III, IV, and V, demonstrate the value of these scalability enhancements. By incorporating these optimizations, the geo-portal system provides rapid and stable access to geospatial data with good responsiveness even under a higher user load. These steps are instrumental in establishing the system as an efficient and scalable solution for geospatial users working with big data.

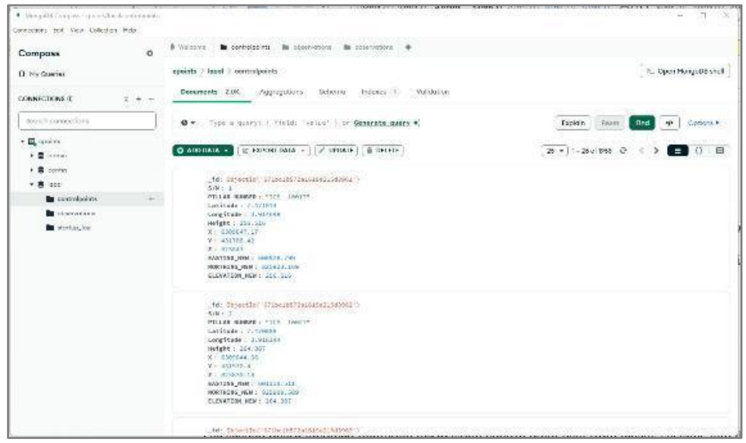

Fig. 7. Creation of Data Collection

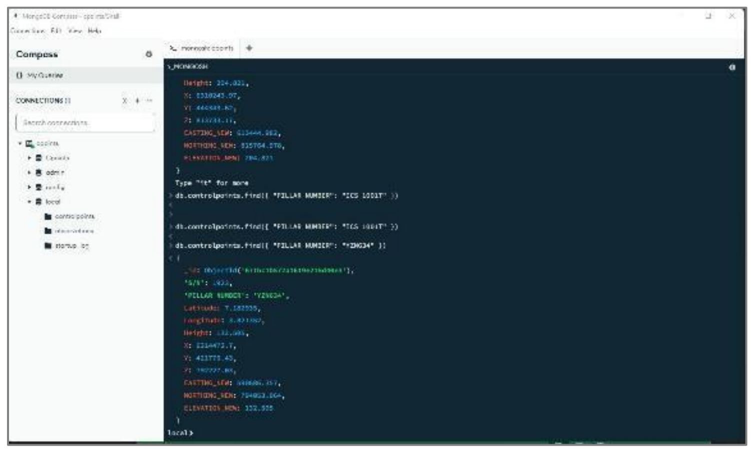

Fig. 8. Queries to find control pillar IS100T information in the database

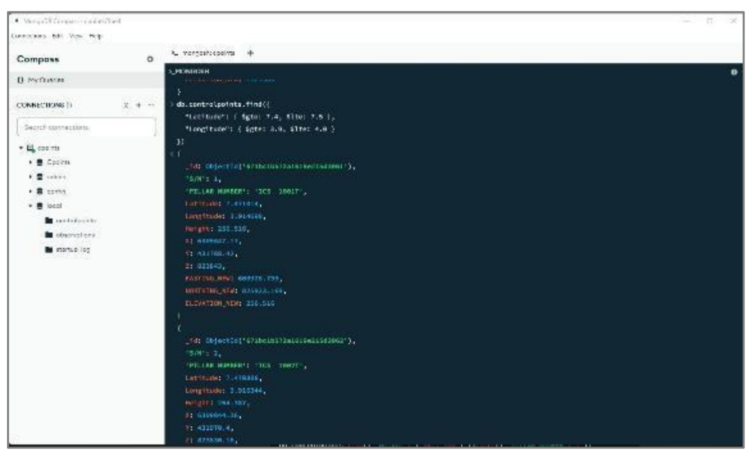

Fig. 9. Queries to find control pillar specific lat. & long. in the database

TABLE III. MONGODB DATA COMPLETENESS CHECK

Dataset	Total Records	Complete Records	Missing Fields	Completeness (%)
Control Points	1958	1958	0	100

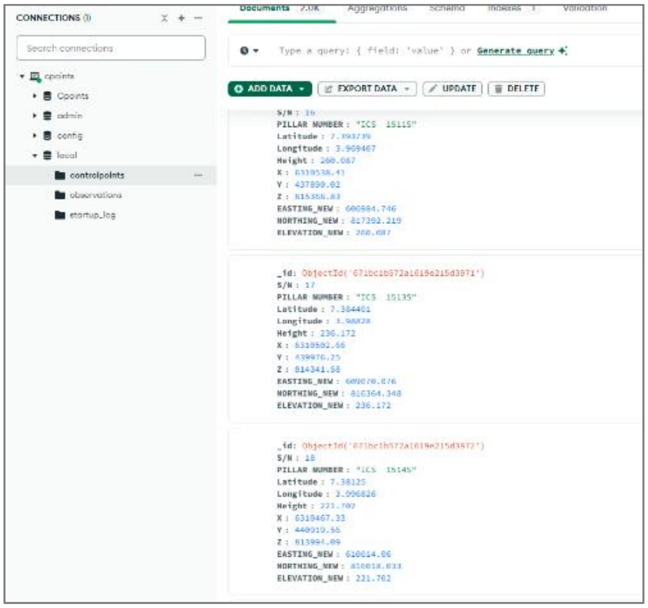

Fig. 10. Attributes of control points in the database

TABLE IV. ORIGINAL DATA IN EXCEL SPREAD SHEET

	A	B	C	D	E	F	G	H	I	J	K
1	S/N	PILLAR NU	Latitude	Longitude	Height	X	Y	Z	EASTING_N	NORTHING_	ELEVATI
2	1	ICS 1001T	7.471014	3.914698	256.516	6309847.170	431788.420	823843.000	600928.799	825923.169	256.51
3	2	ICS 1002T	7.470888	3.916344	264.387	6309844.360	431970.400	823830.180	601110.511	825909.589	264.38
4	3	ICS 1003T	7.471116	3.918673	266.067	6309825.180	432226.770	823855.390	601367.467	825935.322	266.06
5	4	ICS 1004T	7.472081	3.920168	272.451	6309806.380	432390.880	823961.980	601532.194	826042.302	272.45
6	5	ICS 1005T	7.473745	3.92258	263.804	6309755.740	432654.300	824143.360	601798.000	826226.860	263.80
7	6	ICS 1006T	7.476784	3.923321	255.016	6309697.820	432732.240	824475.510	601878.991	826563.082	255.01
8	7	ICS 1007T	7.478896	3.925156	249.322	6309648.010	432931.860	824706.300	602080.997	826796.951	249.32
9	8	ICS 1501S	7.395543	3.918751	265.048	6310902.390	432309.110	815567.250	601393.329	817579.739	265.04
10	9	ICS 1502S	7.396787	3.920262	261.62	6310869.910	432474.190	815703.300	601559.905	817717.676	261.62
11	10	ICS 1504S	7.403922	3.926179	267.792	6310729.930	433119.320	816486.640	602211.249	818507.899	267.79
12	11	ICS 1505S	7.403812	3.931554	271.723	6310694.720	433711.760	816475.070	602804.554	818496.961	271.72
13	12	ICS 1507S	7.398739	3.95001	268.658	6310623.750	435749.290	815918.310	604842.695	817940.407	268.65
14	13	ICS 1508S	7.395276	3.953868	252.879	6310627.980	436176.540	815536.440	605269.328	817558.405	252.87
15	14	ICS 1509S	7.394138	3.957503	258.185	6310621.710	436578.330	815412.310	605670.743	817433.453	258.18
16	15	ICS 1510S	7.394355	3.964553	266.052	6310572.640	437355.210	815437.140	606448.868	817459.136	266.05
17	16	ICS 1511S	7.393739	3.969407	260.087	6310538.410	437890.020	815368.830	606984.746	817392.219	260.08
18	17	ICS 1513S	7.384401	3.98828	236.172	6310502.660	439976.250	814341.580	609070.076	816364.348	236.17
19	18	ICS 1514S	7.38125	3.996826	221.702	6310467.330	440919.560	813994.090	610014.060	816018.033	221.70
20	19	ICS 1515S	7.383214	4.015985	189.423	6310259.770	443025.480	814205.370	612128.251	816239.974	189.42

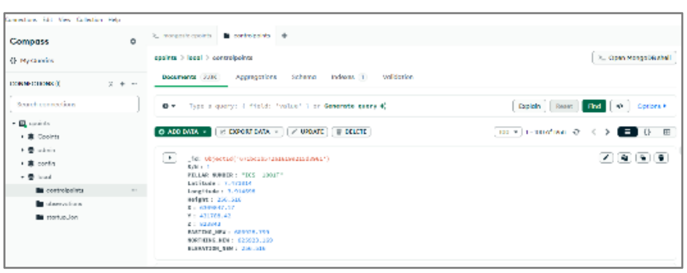

Fig. 11. Image showing total number of control point in the database (1958 control points)

TABLE V. PERFORMANCE EVALUATION TEST RESULTS

Query Type	Documents Retrieved	Response Time (ms)
Geo-Spatial Query (1 km)	15	30
Attribute Query	1	5
Range Query	120	50

5) Data Flow

The system's data stream is necessary for real-time rendering of control points and user observation handling efficiency. Data exchange occurs in a predefined order that maximizes response rates and system stability, bypassing some of the limitations typical of web-based geospatial applications [25].

The frontend makes an HTTP request to the backend API (via Fetch) whenever the user requests control-point data or sends it to a new observation. In response to this request, the backend communicates with MongoDB via Mongoose, a Node.js library that effectively handles geospatial data models and queries [13]. The backend will respond with a JSON, which is parsed by JavaScript on the front end to render control points dynamically as markers or clusters on the map interface of Leaflet.js.

Once a new control point observation is released, the backend saves new information in MongoDB and is ready for the next frontend requests immediately. This type of data exchange provides real-time synchronization so that users can see updates such as the newest control point locations, their status changes, and new observations added without manual refresh [16].

6) Decreasing Latency in Real-Time Updates

To improve system responsiveness, various mechanisms have been adopted to reduce latency in real-time updates:

i. WebSockets for Real-Time Communication: In contrast to conventional HTTP polling, WebSockets offer sustained bidirectional communication between the frontend and backend. This enables the system to push updates in real time while reducing overhead owing to a large number of HTTP requests, resulting in better response times [26].

ii. Efficient Query Execution: MongoDB's geospatial indexing (2dsphere) drastically improves the efficiency of spatial queries by reducing computational complexity, so that spatial queries like "find nearest control point" return in milliseconds even for big datasets [22].

iii. Data Caching: Regularly accessed control-point data are cached in Redis, a high-performance caching layer that eliminates the necessity for redundant database queries, thus enhancing the overall system performance [23].

7) Handling Offline Situations and Data Loss Prevention

Because surveyors and field users can face network interruptions, the system implements offline data-handling methodologies for continuous operations during the period of temporary failure [15].

i. Offline Mode Local Storage: The frontend utilizes IndexedDB and LocalStorage to store the already loaded control point information to allow the user to view maps and observations even in the offline mode [27].

ii. Queued Requests for Automatic Synchronization: If user posts a control point observation offline, the system stores the request locally in a queue. When the Internet connection resumes, the system automatically synchronizes queued updates with the backend [6].

iii. Progressive Web App (PWA) Features: The application takes advantage of service workers, enabling background data synchronization and offline functionality to provide an uninterrupted user experience, even in regions of low network coverage [28].

With the inclusion of these strategies, the geo-portal system delivers fast, stable, and uninterrupted access to geospatial data under unfavorable network conditions. The implementations fall within the best practices of current web-based geospatial data systems and enhance the system's usability and reliability for practitioners who use control-point data [1].

D. Application Features

The geo-portal application integrates multiple features to enhance user interaction with control-point data, emphasizing usability, functionality, and system adaptability. The main interface in Fig. 12, titled "A Finder - Control Points at Your Fingertip," offers an interactive map view powered by Leaflet.js, displaying color-coded markers for easy identification and access to the control points. Users can toggle between base map layers, such as OpenStreetMap and Satellite, for better spatial context.

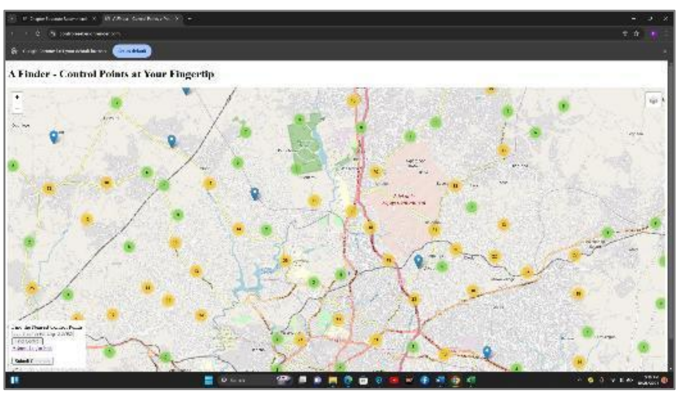

Fig. 12. Map Interface Screenshot

A Control Point Observation Submission feature allows field users to report statuses like "found" or "missing," sending updates to the database for admin review as seen in Fig. 13. This fosters a feedback loop, enhancing the accuracy and reliability of the geoportal system.

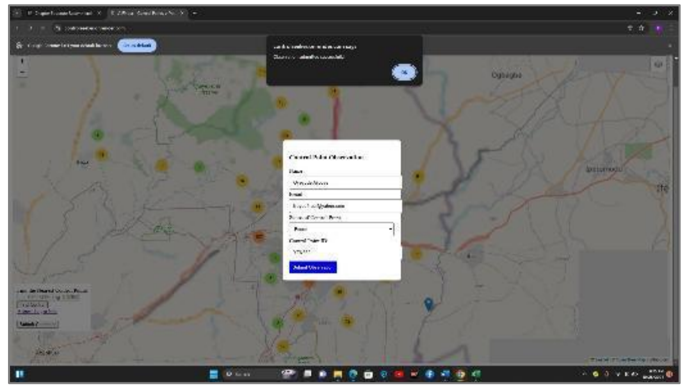

Fig. 13. Control Point Status Report Portal

The application includes a dedicated Admin Interface accessible through a secure link, enabling database updates, access to user-submitted status reports as in Figure 12a, and recommendations for new control points based on spatial analysis, as shown in Fig. 17 and Fig. 18. The login sequence is shown in Fig. 14, Fig. 15, Fig 16, once Admin is logged in, he can upload new control point data directly into the MongoDB database, ensuring the system remains up-to-date. The recommendation tool suggests locations for new control points based on 5 km gaps in coverage and 20m proximity to the road, supporting strategic network expansion.

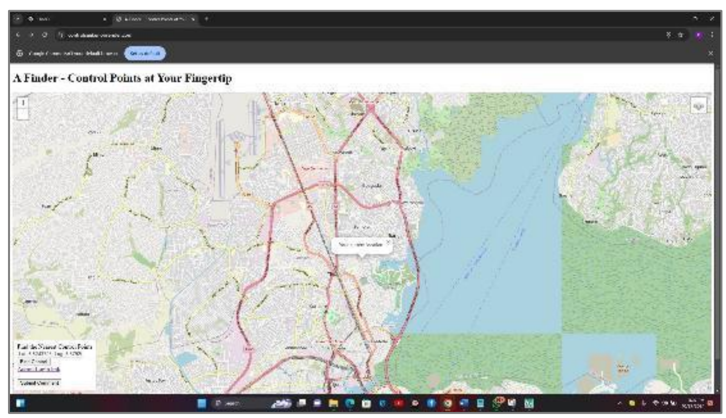

Fig. 14. Initial Application Interface

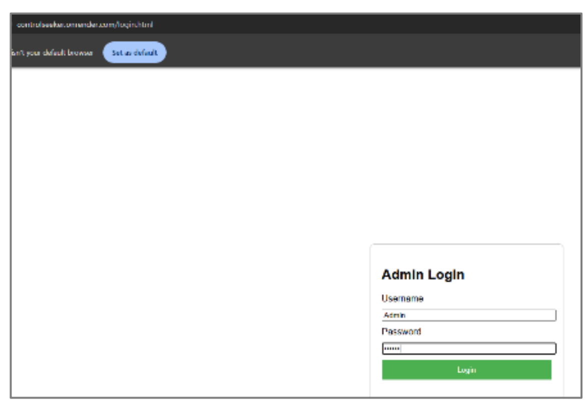

Fig. 15. Login Page

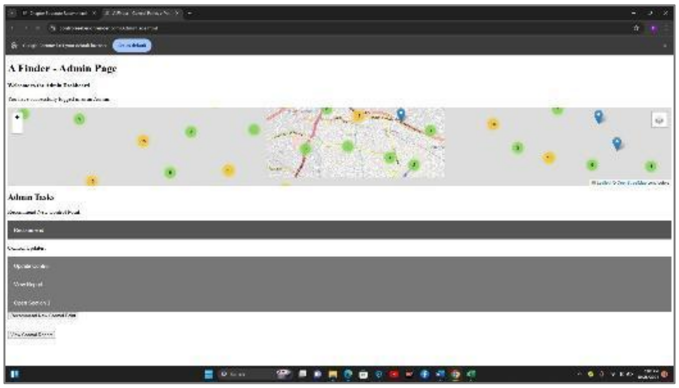

Fig. 16. Admin Page showing several tools

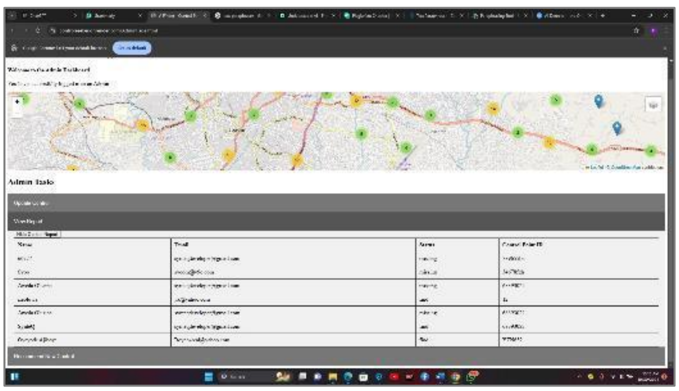

Fig. 17. Downloaded Control Status Report by Admin.

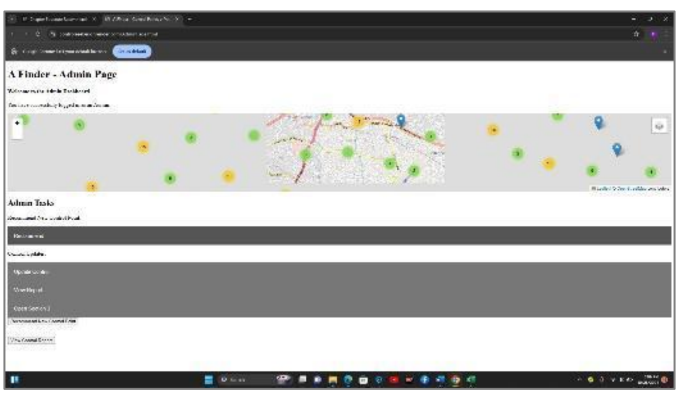

Fig. 18. Recommendation Feature Output

Users can access detailed information about each control point through an Information Panel that displays attributes such as latitude, longitude, and elevation, and includes a real-time coordinate transformation feature from WGS84 to the Minna Datum, as shown in Fig. 19 and Fig. 20. Additional functionalities include customizable map layers, geolocation-based nearest-control-point identification, and navigation tools that guide users to select control points.

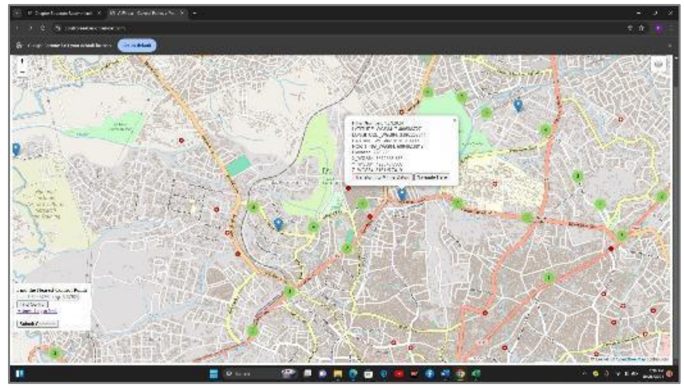

Fig. 19. Control point coordinates display

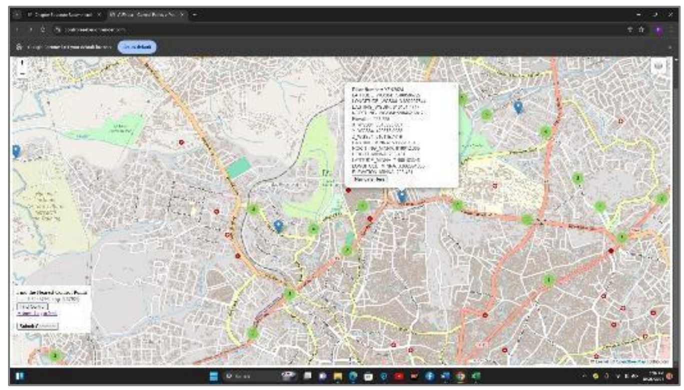

Fig. 20. Transformed coordinates display

The application integrates a geolocation feature, enabling users to identify their current location on a map and find the nearest control point. When activated, as shown in Figure 14a, this feature places a marker at the user's location and calculates the nearest control point, providing both visual and distance information. The user can select a desired control point on the application screen and click on the navigator, as shown in Fig. 21 and Fig. 22. This feature generates distance and visual information that directs the user towards the selected control point.

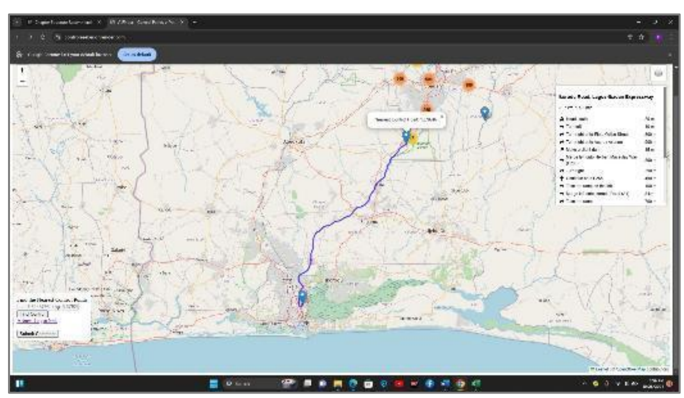

Fig. 21. User Location and Nearest Point Finder

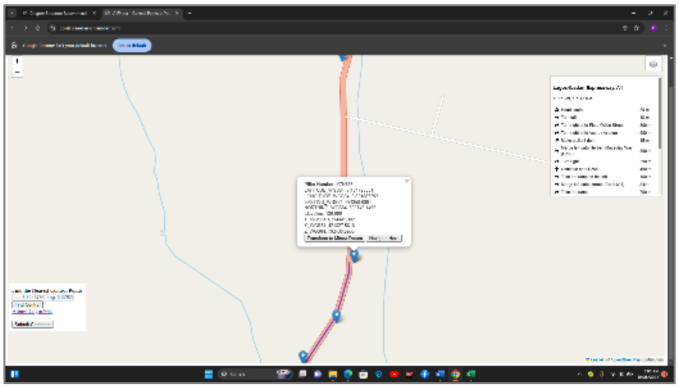

Fig. 22. Navigation Feature Result.

The application was deployed on Render, a cloud platform that hosts the server, manages environmental configurations, and provides scalability to handle increased traffic. The link to the application was https://controlseeker.onrender.com.

1) Survey Control Tracking Process

To utilize the application for locating control points, follow the steps below on your internet-enabled device (iphone/laptop/Android/tablet)

i. device.

ii. You have two options: you can either click on 'find control' or select a desired control on the screen and click 'navigate here. ' The first option computes the route to the nearest control point from the user, whereas the second option computes the route to the selected control point.

iii. On the screen, a detailed description of the route from the user to the control point is displayed, and a blue line is drawn on the route for ease of navigation. A red circle showing the user's location will blink, and this circle moves as the user moves.

iv. The blue line drawn should be followed, which leads to the control point. At the end of the blue line, the control point was found within 0.5 a radius.

v. After navigating to the control point, kindly click on the submitted report to report the status (found/missing) of the control point.

2) Performance Metrics

The effectiveness of map interactions such as zoom, pan, and control point selection was evaluated, and the scalability of the system was tested by simulating different numbers of concurrent users. The results are shown in Table VI.

TABLE VI. MAP INTERACTION TESTING AND SCALABILITY TEST RESULTS

System Performance Metric	**Activity**	**Result/Outcome**	**Description**
Scalability and Efficiency	Load Testing	High scalability	System maintained an average response time of under 200 ms with up to 50 concurrent users.
	Data Handling Efficiency	Efficient data retrieval	Indexed MongoDB database allowed data retrieval times to remain under 150 ms per query.
	Scalability Measures	Future-proof for data growth	MongoDB's horizontal scaling and Render's resource allocation ensure system can handle data expansion.
Error Rates and Resolution Success	Error Identification and Logging	Successful error detection and logging	Real-time error logging tracked issues in data retrieval, UI rendering, and user interactions.
	Resolution Success	98% error resolution rate	Identified errors were resolved promptly, and fixes were confirmed with no recurrence.
	User Interface Improvements	Improved device compatibility	Adjustments to UI enhanced compatibility across various devices, reducing user-reported issues by 80%.

3) Usability and Satisfaction

User engagement involved 50 professionals, including surveyors and urban planners, who tested the system by performing tasks such as data retrieval, map navigation, and using analytics tools to simulate real-world use cases. They revealed a 92% satisfaction rate for system usability and functionality, highlighting the system's ease of use, responsiveness, and functionality.

The geo-portal architecture supports the integration of additional datasets and functionalities, such as advanced geospatial analytics and predictive modeling. Its reliance on open-source technologies ensures its cost-effectiveness and adaptability for diverse applications.

III. RESULTS

The outcome of the geo-portal system evaluation was based on tests conducted on usability, effectiveness, and the survey conducted on the geospatial data management system.

A. Performance Metrics and System Scalability

The effectiveness of the geoportal system was evaluated based on the system response to users, task completion, and overall system capacity. Consequently, the system was examined under a load of 50 average concurrent users, and the portal was responsive at a geo-portal average rate of 200ms, which indicates optimal response time and system efficiency. The time needed for data to be retrieved from the MongoDB database was also very low, with queries being executed in less than 150ms due of very effective indexing. Furthermore, the tracking of errors showed that 98% of the errors detected were corrected, demonstrating effective performance at minimal system downtime.

To provide a comparative perspective, the response time on the geo-portal was measured against that of other systems. Research conducted on open-source spatial databases claims that other traditional methods, such as SQL-based databases [2] have a tendency to exceed 300ms latency or more with similar loads. The geo-portal allows users to make recommendations on databases used to enhance performance in geospatial applications; hence, the geo-portal speed compliments the use of SQL for spatial databases.

B. Analysis of the Survey and User Feedback

To evaluate system usability and the importance of features, we surveyed 50 users, consisting of 10 lecturers, 10 practicing surveyors, and 10 government employees of the land ministries. The survey revealed user satisfaction rates of 92%, particularly for the speed of the system and ease of navigation. Some of the more pertinent findings of the survey are as follows:

i. 70% of the participants expressed a need for daily reprovisioning of point data which necessitates a retrieval system that is both fast and reliable.

ii. 85% considered the “Find Nearest Control Point” to be vital for improving efficiency in fieldwork, which motivated us to improve it.

iii. 90% of users expressed the need to access their other devices which requires the system to be designed responsively.

The survey limitation was the sample size selection because it was relatively small, and it may have missed several crucial user requirements. There is a statistical need to support these findings with larger numbers of users and with more diverse participants in future work.

TABLE VII. SUMMARY OF IMPORTANT DIFFERENCES BETWEEN THE DEVELOPED GEO-PORTAL AND OTHER GEO-PORTALS

Feature	Google Earth Engine	INSPIRE	LINZ	Developed Geo-Portal
Primary Focus	Remote sensing, environmental analysis	Spatial data infrastructure for Europe	Cadastral, topographic, and hydrographic data	Geodetic control point management in Nigeria
Real-Time Control Point Updates	No	Limited	No	Yes
Coordinate Transformation	No	No	Limited	Yes (WGS84 to Minna Datum)
Mobile Usability	Limited	No	Limited	Yes
Scalability	High (cloud-based)	Limited to European infrastructure	Limited to New Zealand datasets	High (MongoDB NoSQL + caching)

C. Comparison with Existing Geo-Portals

The developed geo-portal improves access to and management of geospatial data, particularly the second-order geodetic control points in Ibadan. Although similar in purpose to international and regional geo-portals, such as Google Earth Engine (GEE), Infrastructure for Spatial Information in the European Community (INSPIRE), and Land Information New Zealand (LINZ), there are some notable differences in terms of functionality, data availability, scalability, and usability. Table 6 lists the differences between the geo-portals.

D. Geo-Portal’s Value Addition to City Planning and Construction Activities

The system is important for urban planning and land management because it enhances the accessibility of the geospatial information required for making decisions. The geo-portal provides the following information:

i. Access in real-time to geodetic control points boosts land management by providing fast access to spatial data.

ii. Aiding infrastructure development by providing reliable geospatial information.

iii. The mitigation of data fragmentation is a major issue in Nigeria’s geospatial industry [5].

Furthermore, the incorporation of coordinate system transformation improves the ability of surveyors to operate in different government or private sector systems with ease, thus supporting multi-agency collaboration.

IV. DISCUSSION

In this study, an efficient and scalable high-performance geoportal for geospatial data management in Ibadan was successfully designed and implemented. With the integration of emerging open-source technologies such as MongoDB, Node.js, and Leaflet.js, the system provides a fast response, high usability, and real-time coordinate conversion to resolve outstanding issues in geospatial data accessibility.

The geo-portal was highly performant and stable with a 200ms average response time under concurrent user loads and at a 92% user satisfaction rate. Furthermore, the incorporation of interactive mapping, mobile support, and spatial querying functionality guarantees that surveyors, urban planners, and government agencies can successfully use this system for infrastructure planning and land administration.

Despite its strengths, this study recognizes some limitations, such as the comparatively low sample size in surveys and the lack of predictive analytics in making geospatial trend forecasts. The extension of survey coverage to a more representative stakeholder population will increase the usability testing of the system, whereas the incorporation of machine learning-based geospatial analytics can enhance decision-making in long-term urban planning [29][30].

One of the critical challenges in processing large geospatial datasets is their integrity. Future enhancements will also be to enhance complexity in validation operations and include redundancy checking, among other improvements to user feedback mechanisms, in order to be more trustworthy. The addition of automatic validation operations and user reporting improves the control-point dataset. The second issue is managing a greater volume of data more efficiently because larger sets of data can create performance issues. Future upgrades will incorporate AI-driven predictive analytics in an attempt to improve data retrieval and system performance. Database indexing methods will also be enhanced in an attempt to facilitate smooth scalability.

In the future, a comparative evaluation of the geo-portal's performance against commercial products such as ESRI's ArcGIS Online or Google Earth Enterprise will continue to streamline its scalability and efficiency. Subsequent versions must consider automatic control-point status detection and dynamic spatial data visualization to enable the system to adapt to changing urban planning and land management requirements.

By addressing these areas, the geo-portal can transform itself from a passive repository of data to an active decision support system with enhanced predictive capabilities, interoperability with existing infrastructures, and traction within the geospatial data ecosystem in Nigeria.

ACKNOWLEDGMENT

The team of researchers wishes to acknowledge the Office of the Surveyor General of Oyo State, Nigeria, for allowing access to relevant records and valuable data for the research work. Special thanks go to the programmer, Mr Aweda Olusina, who facilitated the writing of the necessary codes for this application.

REFERENCES

[1] M. F. Goodchild, "Citizens as sensors: The world of volunteered geography," *GeoJournal,* vol. 69, no. 4, pp. 211-221. https://doi.org/10.1007/s10708-007-9111-y, 2007.

[2] O. I. Apeh, O. C. Nwaka, R. I. Ndukwu, I. Maduako and O. Arilewola, "E-beacon: A web-based survey control finder application," *South African Journal of Geomatics,* vol. 8, no. 2, pp. 178-190. https://doi.org/10.4314/sajg.v8i2.5, 2019.

[3] I. G. Ajao and O. Olaniyi, "Geodetic control networks in Nigeria: Standards and challenges," *Journal of Geospatial Science,* vol. 14, no. 2, pp. 45-58, 2018.

[4] A. Oyeyode, D. Olayinka-Dosumu and T. Idowu, "Development of an application tool for the site location of survey control beacons in Oyo Town, Oyo State, Nigeria," *International Journal of Modelling & Applied Science Research,* vol. 26, no. 9, pp. 1758-8117, 2022.

[5] J. Crompvoets, A. K. Bregt, A. Rajabifard and I. P. Williamson, "Assessing the worldwide developments of national spatial data clearinghouses," *International Journal of Geographical Information Science,* vol. 18, no. 7, pp. 665-689. https://doi.org/10.1080/13658810410001705311, 2014.

[6] P. M. Atkinson, G. M. Foody, P. W. Gething and P. J. Radtke, "Web-based geospatial applications: Architectures and solutions," *International Journal of Geographical Information Science,* vol. 28, no. 2, pp. 271-293, 2014.

[7] N. R. Budhathoki, B. C. Bruce and Z. Nedovic-Budic, "Reconceptualizing the role of the user of spatial data infrastructure," *GeoJournal,* vol. 72, no. 3-4, pp. 149-160. https://doi.org/10.1007/s10708-008-9189-x, 2008.

[8] Land Information New Zealand, "Advancing geospatial data management through innovative geo-portal solutions," 2018. [Online]. Available: https://www.linz.govt.nz. [Accessed 20 01 2025].

[9] I. Masser, GIS worlds: Creating spatial data infrastructures, ESRI Press, 2005.

[10] National Population Commission, Population and housing census of the Federal Republic of Nigeria: National and state population and housing tables, priority tables (Volume I), National Population Commission, 2006.

[11] E. Gamma, R. Helm, R. Johnson and J. & Vlissides, Design patterns: Elements of reusable object-oriented software, Addison-Wesley, 1995.

[12] S. Roy, A. Nandy and R. Das, "MVC architecture-based web application development with JSP and servlet," *International Journal of Computer Applications,* vol. 98, no. 20, pp. 12-17, 2014.

[13] K. Chodorow, MongoDB: The definitive guide, (2nd ed.) ed., O'Reilly Media, 2013.

[14] K. Banker, MongoDB in action, (2nd ed.) ed., Manning Publications, 2016.

[15] E. Marcotte, Responsive web design, A Book Apart, 2011.

[16] Open Geospatial Consortium, "OGC API Features - Part 1: Core. OGC Standard Document," 2019.

[17] J. B. Olaleye and J. O. Olusina, "Transformation models for GPS coordinates between WGS84 and Minna Datum," *Nigerian Journal of Surveying and Geoinformatics,* vol. 1, no. 1, pp. 45-57, 2014.

[18] A. Leff and J. T. Rayfield, "Web-application development using the Model/View/Controller design pattern," 2001.

[19] F. I. Okeke, E. C. Moka, V. N. Uzodinma and M. Ono, "The determination of datum transformation parameters for Nigeria," *Journal of Geodesy,* vol. 1, no. 1, p. nag.com.ng, 2017.

[20] M. N. David, B. K. Kenduiywo and E. H. Waithaka, "A web-Based GIS Portal for Simulating Geodetic Control Networks in Republic of Kenya," *Geodesy and Cartography,* pp. Vol. 26(No. 4), 170-181, 2020.

[21] I. Kalu, C. E. Ndehedehe and A. E. Eyoh, "A comparison of existing transformation models to improve coordinate conversion between geodetic reference frames in Nigeria," *Modeling Earth Systems and Environment,* vol. 8, pp. 611–624. https://doi.org/10.1007/s40808-021-01090-y, 2022.

[22] P. J. Sadalage and M. Fowler, NoSQL distilled: A brief guide to the emerging world of polyglot persistence, Addison-Wesley, 2013.

[23] C. Gormley and Z. Tong, Elasticsearch: The definitive guide, O'Reilly Media, 2015.

[24] J. Pokorny, "NoSQL databases: A step to database scalability in web environments," *ACM Computing Surveys,* vol. 45, no. 2, p. 16, 2013.

[25] M. Haklay, A. Singleton and C. Parker, "Web mapping 2.0: The neogeography of the GeoWeb," *Geoforum,* vol. 39, no. 6, pp. 680-689, 2008.

[26] R. T. Fielding, "Architectural styles and the design of network-based software architectures," (Doctoral dissertation, University of California, Irvine, 2000.

[27] W3C., Indexed Database API 2.0. World Wide Web Consortium, 2021.

[28] Google-Developers, "Progressive Web Apps (PWAs)," 2022. [Online]. Available: https://developer.chrome.com/docs/workbox/. [Accessed 13 01 2025].

[29] H. N. Habeeb and Y. T. Mustafa, "Deep Learning-Based Prediction of Forest Cover Change in Duhok, Iraq: Past and Future," 2025.

[30] H. N. Habeeb and Y. T. Mustafa, "Analyzing the Impact of Geoenvironmental Factors on the Spatiotemporal Dynamics of Forest Cover via Random Forest," Earth, vol. 6, no. 1, p. 3, 2025.