Developing an open-source web GIS application to map the completeness and positional accuracy of the OSM road data at the county level in the U.S.

Madusha Sammani

The University of Alabama

Course Number: GY 570

Developing an open-source web GIS application to map the completeness and positional accuracy of the OSM road data at the county level in the U.S.

Introduction

Geographic Information Science (GIS) is a transformative technology that facilitates the storage, manipulation, analysis, and visualization of spatial data in digital form (Alesheikh et al., 2002). Its advanced capabilities enable sophisticated location-based analyses such as geostatistical analysis, geographic modeling, and network analysis, revolutionizing traditional map-making by seamlessly integrating point, line, and polygon features through digital technologies (Mitchell, 2005; Rowland et al., 2020). Despite those advantages, widespread access to GIS and proficiency in GIS techniques are not universal. The initial setup of GIS requires heavy investment, and the cost of GIS software has previously been a barrier for many (Agrawal & Gupta, 2017). However, recent years have seen the emergence of powerful open-source alternatives such as QGIS, making GIS technology more accessible to a broader audience (Lv et al., 2016). Because of the widespread use of GIS technologies, most researchers, including geographers and environmental analysts, are interested in integrating GIS, which increases the number of GIS specialists. As a result, there has been a significant increase in the integration of GIS across various research fields, including geography and environmental science. This trend has led to a growing demand for GIS specialists who can effectively leverage its capabilities. Consequently, the proliferation of GIS technologies has contributed to expanding expertise in spatial analysis and data visualization among researchers worldwide.

While GIS technologies continue to expand, their widespread efficiency hinges on the availability, accuracy, and completeness of data. Many web GIS portals now offer access to a range of data sources, both commercial and freely available, spanning spatial and temporal dimensions (Mooney et al., 2010). For instance, the United States Geological Survey (USGS) provides access to commercial data like QuickBird high-resolution satellite imagery alongside freely available options such as moderate-resolution Landsat imagery. Equity, collaboration, and accessibility can improve significantly when spatial data are available on the Internet (Dragićević & Balram, 2004). Volunteered Geographical Information (VGI) has emerged as a crucial source within GIS, centralizing spatial data accessibility for all users. Notably, platforms like OpenStreetMap (OSM) contribute versatile road data, freely editable and frequently updated by a global volunteer base (Kounadi, 2009). This data not only includes road layouts but also crucial information like road limits, offering distinct advantages over other datasets like the TIGER (Topologically Integrated Geographic Encoding and Referencing) dataset from the US Census Bureau, which tends to be less frequently updated and less comprehensive in terms of road link information. Despite its advantages, the practical application of OSM data necessitates an evaluation of its completeness to ensure its suitability for specific use cases (Girres & Touya, 2010).

GIS serves as a powerful tool for creating maps that visualize the completeness of OSM road data. However, traditional GIS outputs are static, lacking the dynamic capabilities necessary for real-time data dissemination and interaction. The advent of the Internet has revolutionized this landscape, enabling the seamless transfer of information across interconnected computer networks through various protocols (Hardie, 1998). Internet GIS encompasses a broad spectrum of technologies, including web GIS and GIS services such as REST APIs, leveraging the Internet for communication (Kuria et al., 2019). Web GIS, in particular, emerges as an accessible and cost-effective solution for disseminating geospatial information and providing necessary data in decision-making (Ajwaliya et al., 2017). By harnessing the interactive nature of web maps, which allow users to engage with and manipulate spatial data, web GIS facilitates enhanced communication and cognitive processes, enabling users to draw conclusions more readily. The popularity of Web GIS is increasing due to the widespread accessibility of the internet and the decreasing cost of computers and mobile devices. For example, more than 50,000 new web applications were developed by mid-2007 using Google Maps (Haklay et al., 2008).

Various web GIS architectures have been proposed to meet the evolving needs (Agrawal & Gupta, 2017). One of the positive developments is the integration of modern web frameworks with database systems, enabling automatic updates to web maps as underlying data changes (Kraak, 2004). Free and Open-Source Software (FOSS) has played a pivotal role in democratizing access to web GIS tools. Platforms such as OpenLayers and MapServer, developed under the auspices of the Open Source Geospatial (OSGeo) Foundation, empower developers to build interactive mapping applications that seamlessly integrate with web browsers, databases, and map services. These tools facilitate dynamic interactions with spatial data, allowing users to zoom in, pan, and explore map features in real time (Marquesuzaà, 2009). Open-source web mapping software is cost-effective but requires serious coding skills while having unlimited ability to utilize spatial data (Fu, 2018). However, it can slow the application performance based on the high volume of requests from the client (Gkatzoflias et al., 2013).

Therefore, the main objective of this study is to map the results of a detailed evaluation of the OSM dataset's coverage, aiding in identifying areas where the OSM data is either sufficiently detailed or requires further refinement to match the TIGER dataset through the internet.

Data

This web GIS application is based on a map out of the thematic map layers to compare road data from OpenStreetMap (OSM) with TIGER data. Initial data for mapping the completeness and positional accuracy at the county level; 11 fields were considered, which were stored in ESRI shapefile. ESRI shapefile is a vector data format. ESRI shapefiles can be used by other software and applications without any license restriction. Many third-party Python programming modules such as Shapely and GDAL exist for reading and writing shapefiles (Toms et al., 2018). The fields that are used in mapping:

1. The ratio of the total length of OSM roads in a county to the total length of TIGER roads in the county (one field).
2. Buffer analysis: TIGER data was used to create individual buffers around the road in 1-meter increments starting from 1 meter up to 5 meters. The total length of OSM roads and TIGER roads within buffers was derived, and the ratio between the total length of OSM roads and TIGER roads within each county was calculated.
    - Dissolved Buffer Analysis: When evaluating alignment within specific distances, if the buffers around TIGER roads overlapped, they were treated as a single, unified region. This approach helps to understand the collective coverage of OSM data in relation to the TIGER dataset (a total of five fields).
    - Individual Buffer Analysis: In contrast, when the buffers around TIGER roads overlapped, each was considered a distinct region. This method offers insight into the detailed, one-to-one correspondence of OSM data to the TIGER roads (a total of five fields).

Web GIS design

Figure 1 shows the web-GIS architecture design. Web GIS application consists of three parts. They are the client browser, application server, and database server. The User Interface (UI) Tier is the front-end layer of a system architecture. It provides an interface for users to interact with the system. This interaction occurs through different platforms such as web browsers, mobile devices, or desktop applications (Ghimire et al.). This layer receives the maps to the web interface as a Web Map Service (WMS), which is a PNG format. The UI Tier is crucial for ensuring that users can easily navigate and use the features of the system, making it a vital part of any software architecture.

The Application Logic Tier is the middle layer between the UI and Database Tiers. It can process the data, manage database connections, handle mapping interactions (GIS server), and manage user requests, such as "How close is this to..." (Hardwick, 2012). In my application, I used Application Logic Tier to store data, manage database connections, and handle mapping interactions (GIS server). For that, I used the GeoDjango framework. It is connected to the database and GIS servers to process and respond to mapping requests. UI uses HTML, CSS, JavaScript, and frameworks such as Leaflet and Bootstrap from the middle tier to provide functionality. These technologies and frameworks help create a responsive and interactive interface that can adapt to different device screens and enhance user experience.

The Database Tier stores the spatial data with location in the form of the database table and provides the query facilities for the application (Adnan et al., 2010). In addition, the database provides the ability to update and delete facilities. It is denoted as Create-Retrieve-Update-Delete (CRUD) feature capabilities (Tsega, 2010). This Tier uses PostgreSQL with the PostGIS extension to store, manage, and retrieve spatial data. PostgreSQL supports large datasets and can transfer data securely (Baranovskiy & Zharikova, 2016).

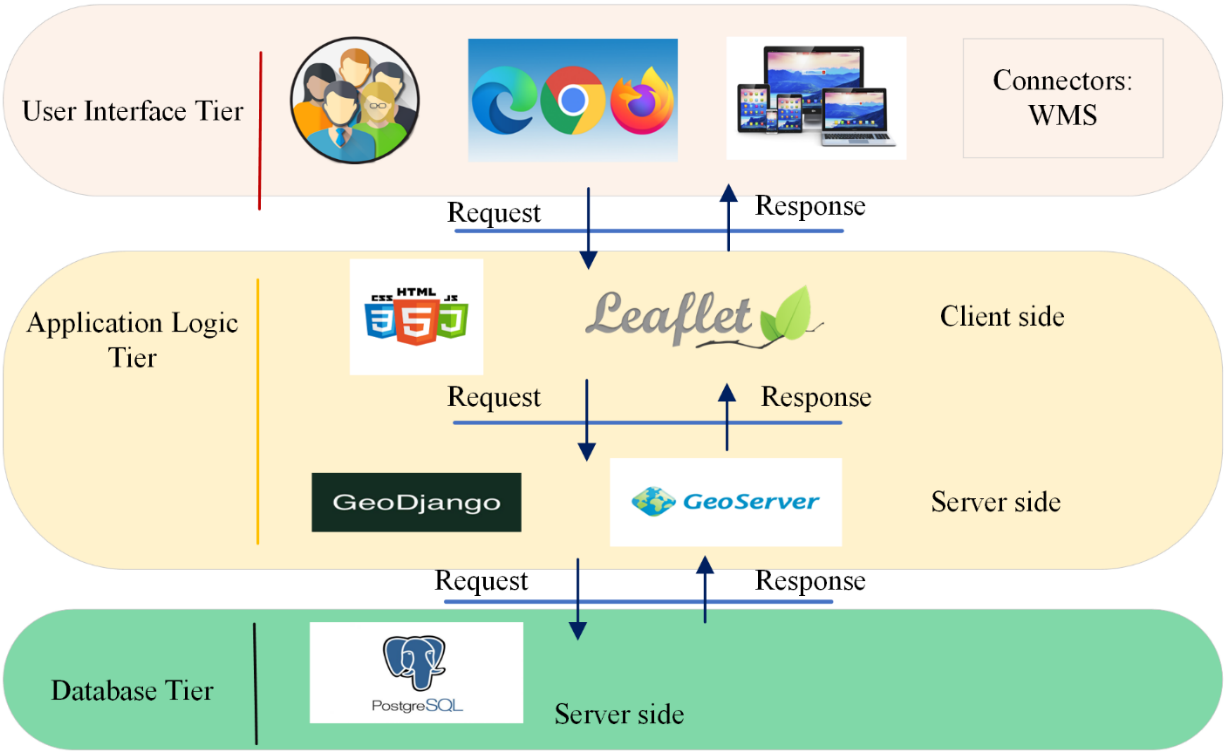

Figure 1. Web-GIS design architecture

Web GIS implementation

Several web application frameworks are available, such as TurboGears, Struts, and Web2Py (Kipkemei, 2010). The proposed Web GIS architecture relies on GeoDjango as the fundamental framework for its geospatial data handling capabilities within the Django web framework. GeoDjango is a Python-based web framework. It allows it to handle increasing amounts of spatial data and the number of users (scalability). In addition, GeoDjango provides security, such as data validation and user permissions for database accessibility (Moyon et al., 2021). GeoDjango integrates seamlessly with PostGIS, an extension for PostgreSQL that adds support for geographic objects, allowing location queries to be run in SQL. This project manages and publishes spatial data within web GIS applications using GeoDjango, which allows the free assembly of PostgreSQL with PostGIS spatial extension and GeoServer (Contreras et al., 2010).

Using GeoDjango, an initial model named "**Shp**" was developed to facilitate file uploads. This model is structured to capture essential details such as the file name, description, the file itself, the upload date, and an automatically generated file ID. These files are integrated into the PostgreSQL database via the Django admin panel. For effective management within the Django administrative interface, the **Shp** model is registered within **admin.py**.

Once the model is registered, the **post_save** and **post_delete** Django signals are utilized to automate the data handling. When a shapefile is saved or modified, the **post_save** signal triggers the conversion of the updated file into a **GeoDataFrame** using **GeoPandas**. This conversion facilitates uploading of the **GeoDataFrame** to the PostgreSQL database table. Moreover, the **post_delete** signal ensures that any deletion of records is cleanly handled by removing corresponding entries from the database. Additionally, the integration extends to publishing the shapefile on **GeoServer**. Geopandas is a powerful Python library that is used to work with geospatial data. Pandas library only can work with numerical and categorical data (Toms et al., 2018).

Following the integration with GeoServer, QGIS was employed for layer styling, leading to the creation of eleven distinct styles. These styles were subsequently uploaded to GeoServer and connected with the corresponding map layers. In the final stage of development, HTML, CSS, and JavaScript (JS) were utilized within the GeoDjango web framework alongside Leaflet to craft the application's user interface and interactive components. HTML is a platform-independent markup language that served as the foundational structure to be displayed in a web browser, while CSS was utilized to ensure cohesive HTML styling for an enhanced user experience (Soomro et al., 1999). JavaScript was crucial in enabling dynamic interactions and functionality within the application interface. Ultimately, the map was published on GeoServer, completing the development process.

In the web application, mapping functions are implemented using Leaflet libraries, which provide functions for zooming in/out, searching locations, controlling layers, printing the map, viewing in full screen, zooming to a layer, displaying the map scale, the map legend, and showing map coordinates. The leaflet is open-source, and it is a lightweight JavaScript library. Bootstrap, which is a production-ready front-end toolkit that provides CSS and JavaScript frameworks, was used to develop responsive and efficient web interfaces (Figure 2). Some of the Python libraries used in web apps are listed below (Table 1).

Table 1. Some python libraries

Library	**Purpose**
Fiona	Open shapefile and access attribute data
Shapely	To handle the geometry (pass the coordinates) of GeoDataFrame
GDAL	Reading, writing, and manipulating raster and vector data, such as coordinate system transformation and feature selection.
pyproj	For cartographic transformation
pyshp	Reading and writing shapefiles
Geoserver-rest	Uploading and managing geospatial map layers
SQLAlchemy	To interact with the database

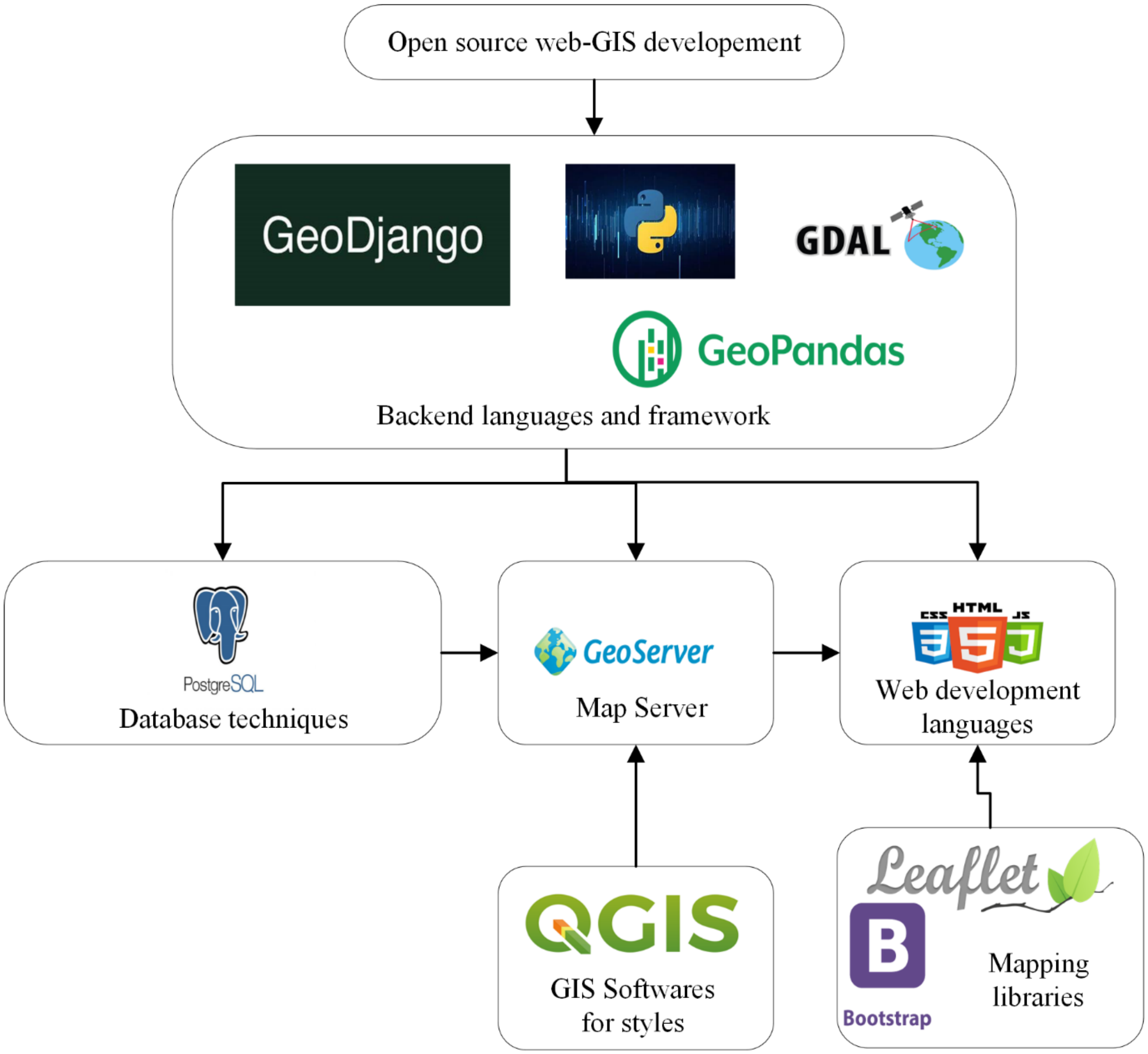

Figure 2. Flowchart to summarize the implemented method of web-GIS application

Results and Discussion

Developed web-GIS application has a base map, eleven operational layers, and tools. Base maps provide a reference map. In my application, without creating a custom base map, I have integrated the OpenStreetMap tile layer to serve as the base map. This is facilitated by utilizing the L.tileLayer function from the Leaflet.js library, which loads and displays the tiles sourced from OpenStreetMap's servers. I have incorporated a total of 11 different layers as operational layers using the WMS (Web Map Service) protocol, each configured using different SLD styles. These layers are hosted on GeoServer and accessed via the Leaflet.js library. Common tasks such as geocoding, printing, and zoom in/zoom out are available. All of these components are web services (Figure 3). The advantage of web services is they can reuse and remix with many web apps.

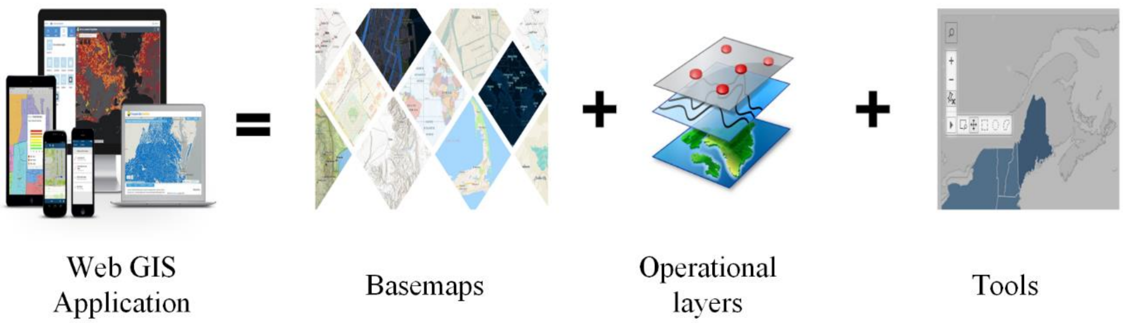

Figure 3. The basic components of a web GIS application

The developed web application is responsive. The web-GIS platform graphical interface includes a legend, zoom in/out, print map, scale, layer control, home, search by address, and full-screen view to aid end-user interaction with the visualized data. Web mapping applications can be static or interactive. Static mapping displays a map on the web, while interactive map viewers can somehow interact with the map, such as selecting map data layers to view or zoom in on a particular part of the map or data query. The limitation of static maps is it does not provide the ability to analyze the data (Zheng et al., 2000). This application allows viewers to select the data layer to visualize. But it is mostly static. Creating an interactive map requires more knowledge.

This web application is a light server and hybrid approach that provides client- and server-side services. In the service-side applications, maps from the GeoServer are transferred as .png images (Chang & Park, 2006). However, this map format minimizes the requirements of client users, such as mouse movements and clicks (Brisaboa et al., 2005). However, the HTML5 map tag solves this issue to some extent. Maps in my web application are not active. They do not provide dynamic information when I click on the map, such as a pop-up. On the client side, this web application provides an opportunity for light interaction with the map interface. The full code of the designed web application is available on GitHub. The URL for the GitHub repository is https://github.com/MadushaSammani/django.git.

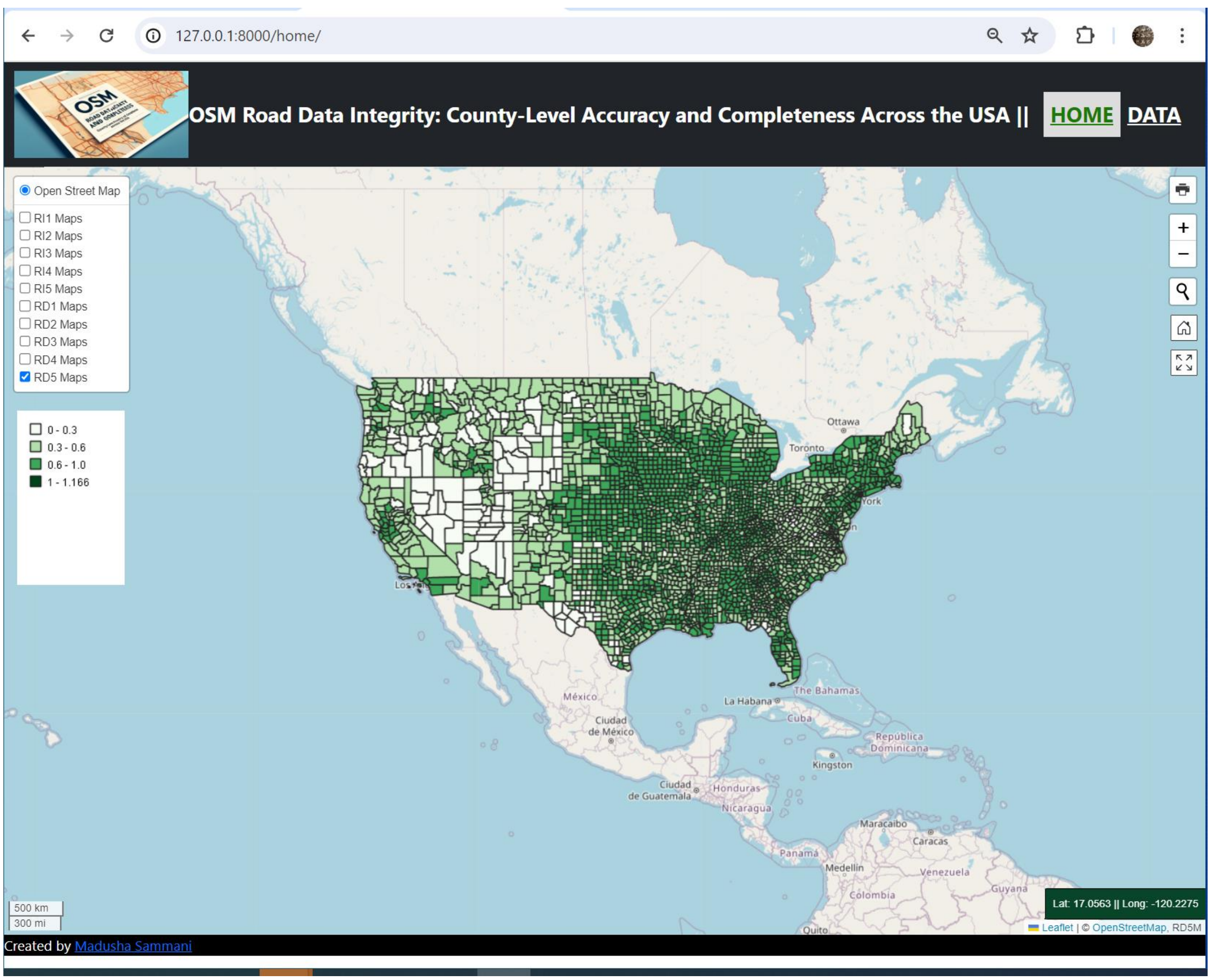

Figure 4. Interface of web application

Conclusion

The primary objective of this web application is to visually map the accuracy and completeness of OpenStreetMap (OSM) road data at the county level across the USA. I used open-source software for web mapping. Unlike open-source web mapping, subscription web mapping such as ESRI ArcGIS Online is very easy and provides different templates. However, they are not affordable for many. This tool is designed without any cost to aid users who rely on OSM data, enabling them to determine whether the data meets the specific requirements for their projects. In addition to that, volunteers who contributed to the OSM can identify areas that they need to contribute more. With its user-friendly interface, the application ensures that even individuals with limited technical expertise can easily navigate and utilize its features effectively. This accessibility makes it an invaluable resource for anyone leveraging OSM road data for detailed and accurate geographic analysis and planning.

Initially, developing the GeoDjango framework posed several challenges. However, after practicing on multiple GeoDjango projects, I have become more comfortable publishing maps using GeoDjango, PostgreSQL with PostGIS spatial extension, and GeoServer. Currently, I am in the process of deploying a GeoDjango application on a Ubuntu server. I'm exploring two setups: one involving Nginx, Tomcat, and PostGIS, and another using Docker Compose, Gunicorn, and Nginx. Despite my progress, I am still on a learning curve and expect it will take some time to master these deployments. One of the significant hurdles I've encountered involves issues with GDAL libraries, which has led me to pause deployment considerations for this project.

References

Adnan, M., Singleton, A., & Longley, P. (2010). Developing efficient web-based GIS applications.

Agrawal, S., & Gupta, R. (2017). Web GIS and its architecture: a review. *Arabian Journal of Geosciences*, *10*, 1-13.

Ajwaliya, R. J., Patel, S., & Sharma, S. A. (2017). Web-GIS based application for utility management system. *Journal of Geomatics, 11*(1), 86-97.

Alesheikh, A. A., Helali, H., & Behroz, H. (2002). Web GIS: technologies and its applications. Symposium on geospatial theory, processing and applications,

Baranovskiy, N., & Zharikova, M. (2016). Web-GIS platform for forest fire danger prediction in Ukraine: Prospects of RS technologies. Remote Sensing of Clouds and the Atmosphere XXI,

Brisaboa, N. R., Luaces, M. R., Paramá, J. R., Trillo, D., & Viqueira, J. R. (2005). Improving accessibility of web-based GIS applications. 16th International Workshop on Database and Expert Systems Applications (DEXA'05),

Chang, Y. S., & Park, H. D. (2006). XML Web Service-based development model for Internet GIS applications. *International Journal of Geographical Information Science, 20*(4), 371-399.

Contreras, F., Farjas, M., & Melero, F. (2010). Free and open source software in archaeological research processes: an application to the study of African red slip ware in northern Italy.

Dragićević, S., & Balram, S. (2004). A Web GIS collaborative framework to structure and manage distributed planning processes. *Journal of geographical systems*, *6*(2), 133-153.

Fu, P. (2018). Getting to know Web GIS. *Photogrammetric engineering & remote SenSing*, *84*(2), 59-60.

Ghimire, C., Karki, D., Yadav, M., & Aryal, S. WEB-GIS BASED DISASTER INFORMATION MANAGEMENT SYSTEM OF LALITPUR METROPOLITIAN CITY.

Girres, J. F., & Touya, G. (2010). Quality assessment of the French OpenStreetMap dataset. *Transactions in GIS*, *14*(4), 435-459.

Gkatzoflias, D., Mellios, G., & Samaras, Z. (2013). Development of a web GIS application for emissions inventory spatial allocation based on open source software tools. *Computers & Geosciences*, *52*, 21-33. https://doi.org/https://doi.org/10.1016/j.cageo.2012.10.011

Haklay, M., Singleton, A., & Parker, C. (2008). Web Mapping 2.0: The Neogeography of the GeoWeb. *Geography Compass*, *2*(6), 2011-2039. https://doi.org/https://doi.org/10.1111/j.1749-8198.2008.00167.x

Hardie, A. (1998). The development and present state of web-GIS. *Cartography*, *27*(2), 11-26.

Hardwick, C. (2012). GeoSocial: Shifting the Paradigm with PlaceSpeak.

Kounadi, O. (2009). Assessing the quality of OpenStreetMap data. *Msc geographical information science, University College of London Department of Civil, Environmental And Geomatic Engineering*, 19.

Kraak, M.-J. (2004). The role of the map in a Web-GIS environment. *Journal of geographical systems*, *6*(2), 83-93.

Kuria, E., Kimani, S., & Mindila, A. (2019). A framework for web GIS development: a review. *International Journal of Computer Applications*, *178*(16), 6-10.

Lv, D., Ying, X., Gao, X., Tao, W., Cui, Y., & Hua, T. (2016). A WebGIS platform design and implementation based on open source GIS middleware. 2016 24th International Conference on Geoinformatics,

Marquesuzaà, C. (2009). WIND: an Interaction Lightweight Programming Model for Geographical Web Applications.

Mitchell, T. (2005). *Web mapping illustrated: using open source GIS toolkits*. " O'Reilly Media, Inc.".

Mooney, P., Corcoran, P., & Winstanley, A. C. (2010). Towards quality metrics for OpenStreetMap. Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems,

Moyon, E. S., Morales, E. M. O., & Jayoma, J. M. (2021, 28-30 Nov. 2021). Design and development of a web GIS-based visualization and analytical platform for Farm-to-Market Road projects of the Philippines' Department of Agriculture. 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM),

Rowland, A., Folmer, E., & Beek, W. (2020). Towards self-service gis—combining the best of the semantic web and web gis. *ISPRS international journal of geo-information*, *9*(12), 753.

Soomro, T. R., Zheng, K., & Pan, Y. (1999). HTML and multimedia Web GIS. Proceedings Third International Conference on Computational Intelligence and Multimedia Applications. ICCIMA'99 (Cat. No. PR00300), Toms, S., Crickard, P., & Van Rees, E. (2018). *Mastering geospatial analysis with Python: explore GIS processing and learn to work with GeoDjango, CARTOframes and MapboxGL-Jupyter*. Packt Publishing Ltd.

Tsega, H. B. (2010). *Spatial Database Modeling and Consistency in Web Frameworks Engineering: the case of python-powered web2py framework* University of Twente].

Zheng, K.-g., Rahim, S. T., & Pan, Y.-h. (2000). Web GIS: implementation issues. *Chinese Geographical Science*, *10*, 74-79.