Automated parcel-based reports for different environmental conditions based on open-source technologies

Edah ŠAHINOVIĆ

[1]Z_GIS, University of Salzburg, Austria · edah.sahinovic@stud.plus.ac.at

Abstract

Technological advancements catalyzed a new way of working and unlocked the potential of new process automation. More specifically, the advancement of GIS technologies supported the automated workflow automation, which, in turn, reduced time-consuming operations regarding spatial domain processes. Combining technologies with geoprocessing and geovisualization tools produces powerful platforms to accommodate specific needs. This project aims to provide such a combination of various tools to develop a fully open-source web-based platform that allows the report generation based on a user-requested parcel's environmental situational data. The information about the parcels is intended to be communicated through narrative, and maps, as well as complementary tabular forms generated per concrete user requests. With such a solution at their disposal to facilitate the need for quicker exchange of the parcel-based information, providers and requesters can enjoy the benefits of reducing the workload, costs and time taken by conventional approach. The solution proposed in the project uses the region of Upper Austria as a playing field for testing, and it is based on the Python packages and algorithms for queries, data extraction and processing. The report generation is based on the data hosted on Open Geospatial Consortium Web Services. The end users of the study are both GIS and non-GIS professionals associated with service-providing institution, as well as report requesters. The results are intended to be delivered through a system which generates PDF reports as output based on geographical snapshots and land data descriptions of the relevant parcel. The research is intended to contribute to the growing field of automation that facilitates the decision-making processes, as well as to provide a blueprint for potential projects aiming to deliver environmental parcel reports for land valuation based solely on open-source technology.

Keywords: Land parcel reports, automation, open geospatial consortium web services, open-source libraries

1. Introduction

The role that information and communication technologies (ICTs) play in organisation of the business and leisure domains increases rapidly, and these tools play a key role in the execution of everyday professional and private planning and decision-making processes. This is evident in the governmental sectors, for instance, where ICT applications and e-services are used to accommodate the need for the quicker service delivery to citizens, on the one hand, and to improve government management and operations, on the other[1]. Technology aids governmental endeavours to utilize delivery and accessibility of information, namely decentralizing the workload of public administration of the services provided to the businesses, citizens and government employees[2].

These endeavors are carried out through various ICT system solutions, including the provision of automated reports to stakeholders, allowing for timely and accurate information delivery without the need for manual intervention. Through e-services, automated reports are easily accessible as they can be accessed by stakeholders from anywhere at any point of time, going hand in hand with enhancing the transparency in government operations and stakeholder engagement and satisfaction. Reports in fields such as land usage plans, landowner boundaries plans and construction permission plans include spatial information to provide comprehensive and detailed insights, rendering GIS to play a critical role in collecting, processing and visualizing such spatial information.

Web-based GIS platforms, compared to traditional methods, offer governments more efficient and effective means to better serve their citizens, with two main categories of them existing: first type involving interactive online maps, enabling the users to perform spatial queries, address searches, route analyses, buffering and overlaying analysis; while second type involves the transmission of spatial documents from various agencies to clients through the internet[1].

The main objective of *ParcelInsights* is to create an open-source web-based platform that generates a PDF report with detailed environmental information, presented in narrative, map and tabular forms. The user can interact and request a report by clicking on a map or by entering a parcel unit IDs such as administrative unit and parcel number in two simple input boxes provided. The report contains information regarding the land's topography, geology and natural hazards as well as information regarding the parcel itself and its neighboring units. The web application provides an easy-to-use interface for users to quickly access and download the desired report.

2. Material and Methods

ParcelInsights is a website designed for users that intend to request summary reports on environmental conditions for specific land parcels in Upper Austria region. The application overcomes the traditional methods of two-sided communication in requesting, on the one hand, and creating and delivering reports in physical format, on the other hand. The application consists of front-end component that allows interaction between the user and the tool and the back-end component to deliver the request by extracting the spatial information and converting it into appropriate formats to be transmitted into a PDF-based report.

The complete project utilizes open-source technologies such as OGC Web Services, web-based technologies, and Python libraries, namely the fundamental ones for the project being *OWSlib, Rasterio*, *Geopandas, ReportLab* and *Flask* as a web framework. By using these technologies, the project avoids reliance on proprietary software solutions and eliminates dependencies, which is also one of key goals of this paper. These technologies will be discussed in more detail in further subheadings.

Due to high deployment cost and potential cyber-attacks with short time to implement the countermeasures, the website associated with the project is maintained as a repository on the GitHub platform. The source code, documentation and project management can be found on: https://git.sbg.ac.at/st22_512323/i3-project-st23-sahinovic.

2.1. OGC Web Services

To enable the accessibility of the spatial datasets relevant for this project, they were published using the Open Geospatial Consortium (OGC) Web Services. OGC Web Services enable the sharing and accessing of geospatial data over the internet. In this project, the spatial datasets were made available through two of the University's geospatial hosting servers: GeoServer and ArcGIS REST Service. The GeoServer server used for hosting the datasets can be accessed via the following link: https://geoserver22s.zgis.at/geoserver/web/?1 and the ArcGIS REST Service server can be accessed via the following link: https://zgis188.geo.sbg.ac.at/arcgis/rest/services. After publishing the datasets as services, they are made available for querying either by HTTP request or by utilizing the OWSLib, a Python library that will be later discussed in further detail.

The datasets that were acquired and published as OGC Web Services play a crucial role in reflecting the environmental conditions of the parcels, which are essential to the project's objectives. These datasets hold a wide range of geospatial information related to the topographic, geological and flood risk characteristics of the parcel's area of concern. Additionally, one of the datasets represents the parcel data itself, providing detailed information about its cadastral boundary, its neighboring parcels, the administrative unit the parcel belongs to and its unique parcel number. The complete list of datasets found within the study area to play a significant role reflecting the environmental conditions of the parcel unit and their metrics used can are presented in Table 1.

Table 1. Datasets relevant for the project, their source, servers hosted on and metrics used

Dataset	Dataset Source and URL	Hosted on/as	Metrics (extracted and derived)
Cadastral boundaries	Sourced from a third party on GeoServer	GeoServer/WFS	Municipality, district, province, country, size, perimeter, latitude and longitude
Geology 1:20,000	Source: data.gv.at URL: https://www.data.gv.at/katalog/en/dataset/land-ooe_geologie-1-20-000	GeoServer/WFS	Code, formation, lithology main, lithology nearby, short title, coverage
Flood risk zones	Source: Land Oberösterreich URL: https://www.land-oberoesterreich.gv.at/122181.htm	GeoServer/WFS	Land ID, zone category, coverage
Digital Elevation Model 10m	Source: data.gv.at URL: https://www.data.gv.at/katalog/de/dataset/land-ooe_digitales-gelandemodell-dgm-als-10m-raster-oberosterreich	ArcGIS REST Service/WMS and WCS	Minimum, average and maximum elevation
Aspect	Derived from the Digital Elevation Model 10m	ArcGIS REST Service /WMS and WCS	Orientation categories
Slope	Derived from the Digital Elevation Model 10m	ArcGIS REST Service/WMS and WCS	Slope categories

2.2. Front-end component

Front-end component designs a web page (as seen in Figure 1.) dedicated to enable user interaction with its elements such as a web map and input box in order to provide parameters necessary for the back-end to handle the request. This small, simple web application is built based on the three fundamental building blocks of the web: HTML, CSS, and JavaScript. The web map is based on OpenLayers [3] JavaScript library and constitutes a main element of the web page. The initial setup of the web map represents a simple base map with a sample parcel units' layer for the region of Upper

Austria for which a single parcel feature can be selected and for which a report is to be generated. If the user is not confident in navigating to the corresponding parcel unit via the map element, simple input boxes coexist and require two parcel inputs, namely parcel administrative unit and parcel number, as parameters to distinguish from other parcels. If the user chooses to interact with the map, after clicking on the parcel the status section exists to prompt the user if he is certain whether the current selection is the one the user has intended to process. Additionally, the web page consists of a text-based description of the project as well as of sets of instructions how to use the page. The last element of the web site is a button input widget which acts as a trigger to perform the action of generating and downloading a PDF after the correct input has been previously given. This widget only becomes responsive after one of the two input methods are initiated, ensuring the user he has interacted with the elements in a correct way by providing sufficient information to be processed.

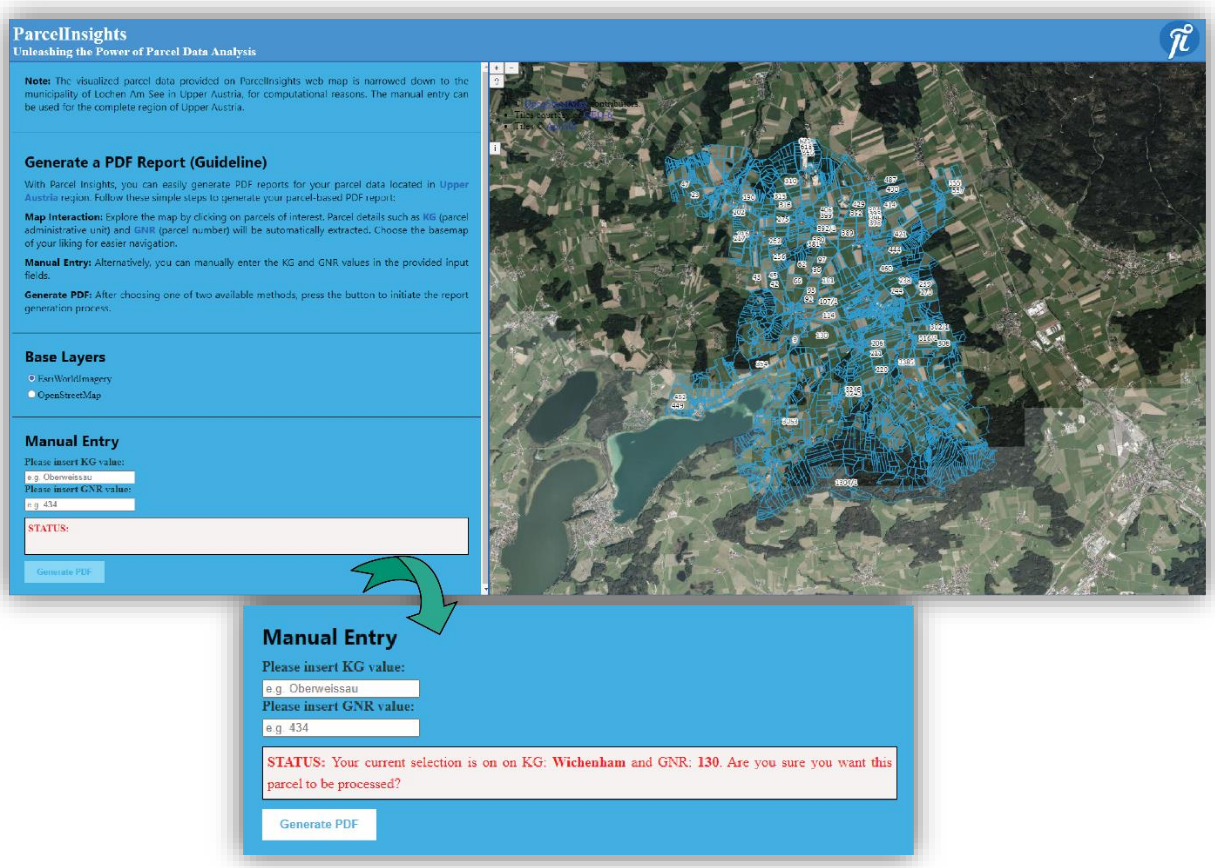

Figure 1. The ParcelInsights web page. Top: the initial web page view. Bottom: the 'status' section after the user has clicked on one of the parcels highlighted on the map.

2.3. Back-end component

As previously highlighted, one of the key objectives of the project is to steer clear of proprietary software and dependencies and to exclusively build upon open-source software. This approach fosters knowledge sharing, encourages collaboration, and enhances improvements and updates by other contributors. Therefore, Python programming language was chosen as the primary tool to handle the project. Python serves an adequate environment for data querying, extracting, processing, analysis as well as their visualization in different forms, providing consistent integration and compatibility through the project workflow. In the project, Python is utilized to send requests and retrieve relevant geospatial data from the OGC Web Services. Additionally, to establish a connection between the processing back-end and the front-end interface, Python's web framework Flask was used.

2.3.1. Python libraries

Further Python open-source libraries make up the main pillars of this project:

1) Flask [4]*:* the project relies on a *Flask* web framework that provides a minimalist and modular approach for building straightforward websites. Its capabilities enable a seamless two-way communication with front-end components, making it perfect fit for the project, as it transfers provided inputs for the analysis and returns the desired outcomes to the web interface.
2) OWSLib [5]: OWSLib is a tool for establishing interaction with Open Geospatial Consortium (OGC) Web Services (OWS), such as Web Feature Services (WFS), Web Coverage Services (WCS) and Web Map Services (WMS) as a base library to perform requests towards datasets that will be processed as outputs for the final report. Furthermore, it ideally serves as a filter to request a portion of the dataset, restricting the download of a full dataset to avoid congestion of the data transfer. The intention of using this library is spatial filtering, that is, a retrieval of features and maps from general datasets (topography, geology and natural hazards) to the extent of the parcel in question to mitigate the request overload.
3) Geopandas [6]: tool and format for enabling vector spatial data format transformations and spatial operations, including geospatial information extraction, coordinate system transformations and spatial joins.
4) Rasterio [7]: a tool for reading and processing raster spatial data formats. Rasterio supports geospatial operations such as resampling, reprojection and data translation. In this project, Rasterio will be utilized for efficient handling and manipulation of raster datasets to meet the project’s specific requirements.
5) ReportLab [8]: a tool for generating customized PDF documents programmatically, rendering it a perfect fit the on-the-fly report generation. For this project, *ReportLab* will be employed to generate PDF reports containing and presenting spatial information in structured and easily comprehendible formats such as text paragraphs, images and tables.

Other libraries used within the project aside the aforementioned ones and are complementing their work in certain aspect are: PIL, Contextily, Matplotlib, Datetime, Geopy, OS, Pandas, Pyproj, Numpy and Shapely.

2.3.2. OGC Web Services querying, information extraction and analysis

Python’s OWSLib library established the connection with OGC Web Services by querying and retrieving the subsets of published WFS, WCS and WMS datasets. To accomplish this, the “getCoverage” operation was issued by specifying the parameters such as the web service version, bounding box, spatial reference system, output format, and size, depending on the type of the web service. The initial retrieval refers to the parcel in question which is extracted by two parameters specified through the web page, serving as a bounding box for further dataset extraction.

The WFS features were stored in geopandas format, subsequently allowing for both information extraction and plotting on maps. It further eliminates the need for local storing as during the script process the information is stored within a defined variable. For WMS request, local saves in this case are unavoidable as further steps are undertaken with the retrieved data to receive the desired output. The WMS initiates two requests, one retrieval being the raster representation of the given area as the second one retrieves its legend. These two elements are subsequently merged into a single image representation. Similarly, for WCS requests, local saves were performed as the initial raster retrieval conforms not to the extend of the parcel feature itself, but rather by specifying a bounding box consisting of four points. After cropping to the true extent, Python’s Numpy histogram was used to categorize the single band cell values into bins based on predefined categories.

2.3.3. Fitting to the PDF document

Upon obtaining the relevant datasets through the requests made to the OGC Web Services and transforming them to the desired formats, the subsequent step involves the creation of a comprehensive PDF document. The process of programmatically generating the PDF document beings by setting up the document's structure and styles, such as defining even spacings between the elements, their position, size, and visual appearance as well as structuring its header and footer.

Populating of the document with content starts by adding a first page dedicated to the front page, presenting the project title and subtitle. Next pages are dedicated for each layer of interest by entering a loop procedure. Figure 2. presents the structure of each page for these layers of interest having the identical structure: the title, introductory statement sentences, a map image, a table, and finally a textual description. The title identifies the specific layer being discussed. The introductory statements provide a brief overview of the layer by providing background information that prepares the reader for what they are about to see. Map image as a visual layer representation helps readers interpret the spatial distribution of the present features within the parcel. Table provides a structured summary of the key metrics that are either extracted or derived from the layer of concern. Finally, a rule-based text description at the end is generated by converting the relevant information into textual form.

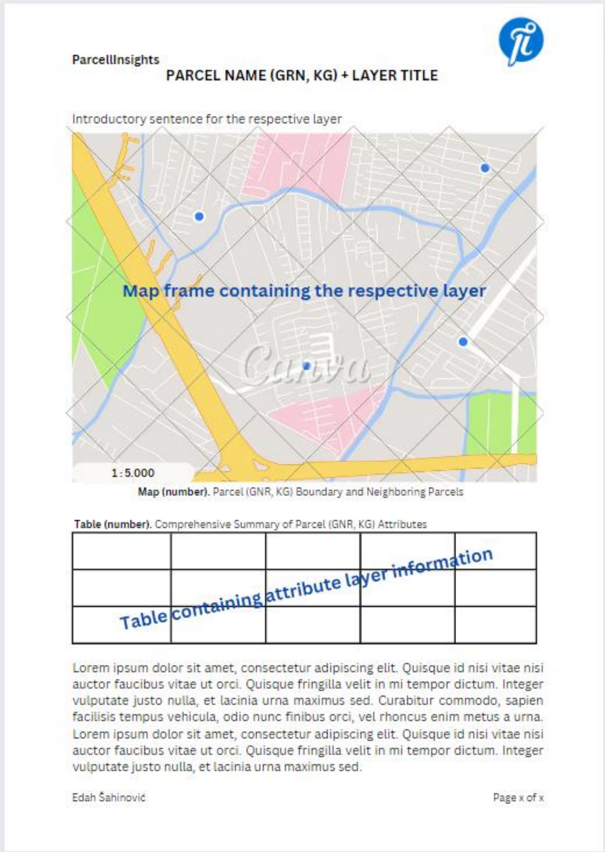

Figure 2. Mockup of page layout for parcel-based reports

3. Results

3.1. Generated PDF reports

The output of the tool is a final PDF report with the description and the status on topographical features, flood risk zones and geological characteristics for the designated parcel unit as well as information regarding the parcel itself. The results of the report are communicated through various forms which include maps, tabular representation, and textual information. Along with generic graphical output, textual representation provides their contextualization, allowing the reader to understand the significance of the results in a broader context.

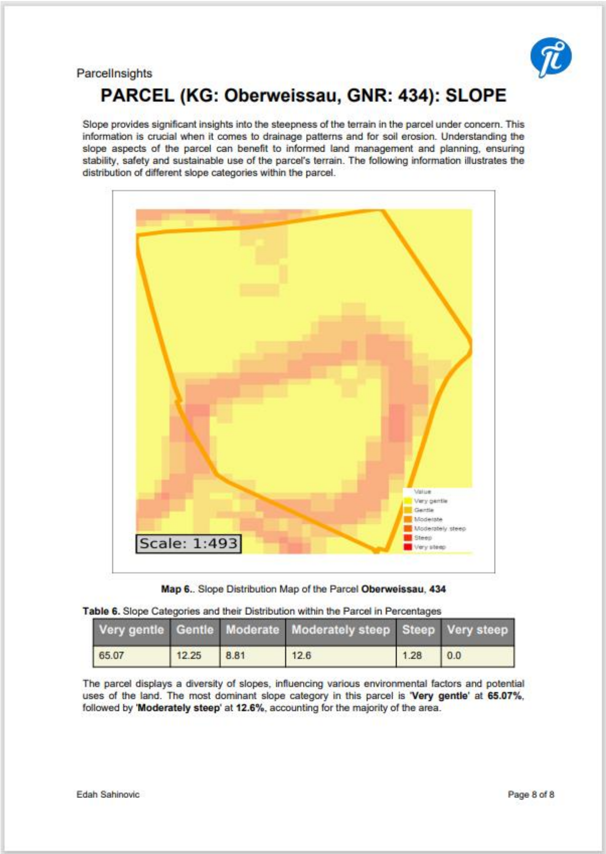

Figure 3. Example slope characteristic output of a parcel-based report

A final parcel-based report, depending on if the parcel of concern is also affected by flood risk zones, generally contains about eight pages. The report encompasses six distinct environmental factors which are described, including the overview of the parcel itself. parcel layer provides information about where is situated by including its municipality, district, province, and country as well as its metrics such as perimeter, latitude, and longitude. Furthermore, neighbouring parcels that share borders with the parcel of concern are also included in the map overview. This allows for better understanding of the parcel's surroundings and neighbouring land units. geology layer represents geological information on a scale of 1:20,000 and includes information as code, formation, main and nearby lithology. It provides insights into the geological characteristics, rock types and formations that influence the land's geological context. Flood risk zones layer (if present) represents parts or complete extent of the parcel affected by flood prone areas. It helps identify areas that are due to flood risks either a no construction zone (red zone), priority zone for retention, runoff, and water management (red-yellow zone), mandatory and precautionary zone (yellow zone) and zone for water management needs (blue zone). Digital elevation model exhibits the topography and elevation characteristics, including minimum, average, and maximum elevation values within the area. It aids in analysing patterns such as drainage and landscape features. Aspect layer provides and overview of the direction of slopes as well as distribution of flat ground. This can help in better understanding of the solar exposure and wind patterns of the area. Lastly, slope layer provides information about the steepness of the terrain, presented as slope categories from very gentle to very steep. It can aid in land suitability assessment for construction or interventions needed to be undertaken for construction.

3.2. Result validation

This section is dedicated to identifying inconsistencies in the output that may have occurred during the undergoing processing steps. Validation was performed on the parcel unit that can be found on project's GitHub repository as a parcel-based PDF document named as "parcel_report_sample.pdf". To assess the accuracy of the data which has undergone various steps from source data to subsequently obtained processed data, a cross-check was performed. The comparison was carried out between the (final) processed data and the source data. Additionally, the processed data was also run through the same procedure within the ArcGIS Pro software as the source data. Here, the aspect and slope layer have been taken under consideration as they cover the entire area as well as their values exhibit a wide range of variability and scattering.

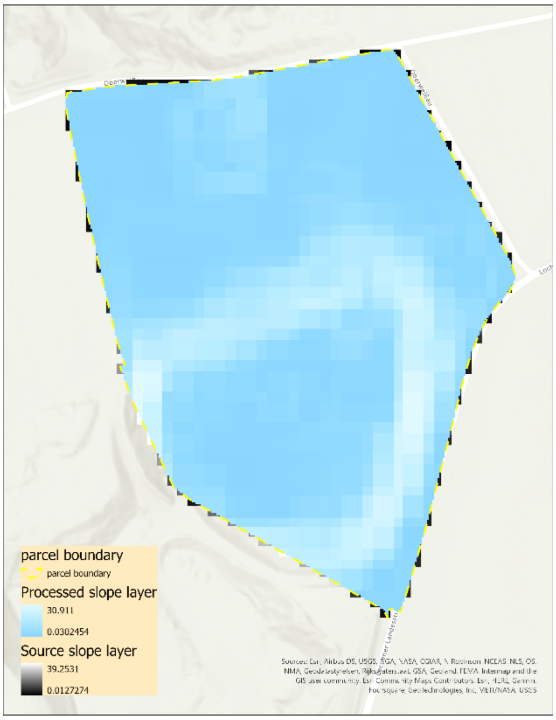

Figure 4. Slope overlay comparison between ArcGIS Pro and Rasterio clipping method outputs

By examining the results in Table 2. it is conclusive that the processed aspect layer exhibits unacceptable degree of inconsistencies. In contrast, the slope layer shows a high degree of consistency. With regards to slope layer, the validation process indicated a close match between the two processed data, most probably indicating that higher variations between the source data and processed data with Rasterio arise due to differing raster clipping methods. Visual inspection of these two outputs in Figure 4. proves that the layers don't overlay perfectly as well as the existence of mild grid cell displacements between them. ArcGIS Pro's "clip raster" tool preserves cells that are partially outside of the parcel's extent while Rasterio's clipping method slices them, not maintaining any cell parts outside the parcel's extent. This is also evident by comparing the two processed data results from different environments as their shape is preserved regardless of the environment they were categorized and quantified in.

Regarding the aspect layer, the situation differs. High degree value inconsistency is present among all three results. Moreover, both results from ArcGIS Pro show that flat category does not exist within the study area, which is also evident from visual inspection in Figure 5. Note that the flat category within the visualization does not exist since in ArcGIS Pro it is not possible to categorize a non-existent value. This discrepancy suggests that during the categorization of orientations in Rasterio, the value “-1” representing the flat surface was not properly assigned to a category. When the processed layer is imported into ArcGIS Pro, the band displays values ranging from 0.132 to 359.9.

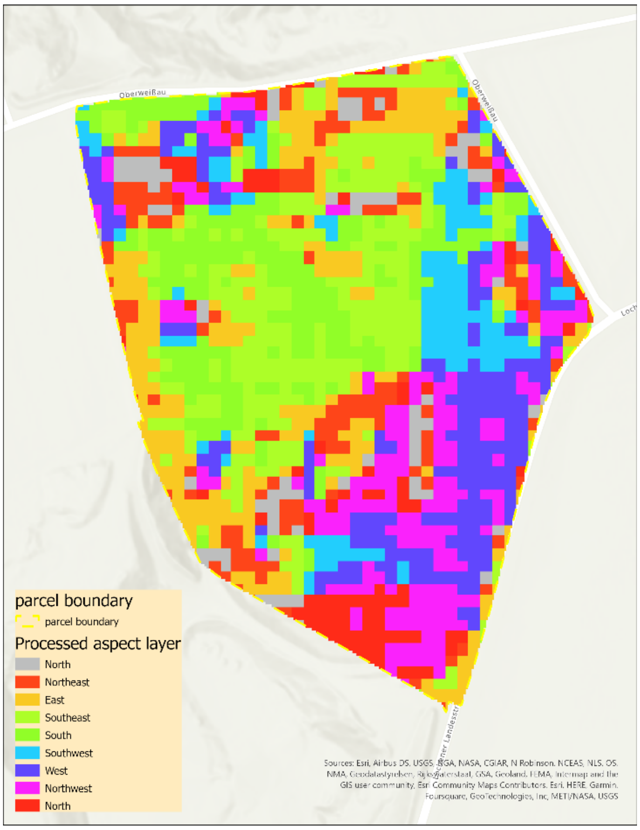

Figure 5. Processed aspect layer orientation distribution

Table 2. Comparison of aspect and slope values in % derived from source data and processed data within Python environment and ArcGIS Pro

Layer	Category	Source data (ArcGIS Pro)	Processed data (ArcGIS Pro)	Processed data (Rasterio)
Slope	0 – 5°	63.48	64.87	65.07
	5 - 10°	12.8	12.27	12.25
	10 - 15°	9.30	8.81	8.81
	15 - 25°	13.48	12.62	12.6
	25 - 35°	0.94	1.25	1.28
	> 35°	0.00	0.0	0.0
Aspect	Flat	0.00	0.0	4.67
	North	11.19	10.3	8.57
	Northeast	7.35	8.61	14.71
	East	13.91	14.68	18.11
	Southeast	18.42	18.09	14.29
	South	14.71	14.3	8.25

	Southwest	7.73	8.21	13.41
	West	13.1	13.47	12.3
	Northwest	13.6	12.35	5.68

4. Discussion

The results presented in the previous section demonstrate the successful achievement of the project’s objective and overall development of the workflow based solely on open-source technologies. Additionally, the web-based interface serving as a graphical user interface for GIS-based back-end processing showcases the potential of web-based applications in replacing desktop GIS applications. This endeavour does not only enhance the accessibility to the GIS-related platforms via World Wide Web but also eliminates procedures such as download, installation and storing of the packages on a local machine.

However, the project did encounter several shortcomings that need to be addressed. Such an issue is accessing certain datasets such as WMS related to hillshade layer which was intended to serve as a visual representation of the digital elevation model. These issues of unknown nature can seriously harm the complete project as it presents an essential component of the complete workflow. Furthermore, a ParcelInsights’ web map provides a locally stored GEOJSON file instead of utilizing the WFS retrieval which resulted in slow performance. For the purposes of this project as a temporary solution a subset is provided since OpenLayers did not prove to hold computational power to handle complex shape datasets such as parcel units. One of encompassing ideas of the project is to host and access all the geospatial information ‘on cloud’ as requested, keeping the overall system solution lightweight. This issue could be mitigated in future versions of the project by implementing features such as user input to narrow down the search area by a desired district, municipality, or a community. Lastly, controlling the zoom extent by providing a map scale for the image representations brought complex challenges, rendering the maps not being accurately set to the intended extent.

5. Outlook

The ParcelInsights platform, in its current development stage, represents a minimalistic framework aimed at achieving its primary objective – to deliver a fully open-source web-based platform that allows the report generation based on a user-requested parcel’s environmental situational data. With its core functionalities set up, such as retrieval and transformation of geospatial datasets hosted on servers into meaningful graphical and textual representations, as well as generation of comprehensive PDF reports, it surely stands as a starting point for future advancements in parcel-based insights.

One opportunity for expansion lies in the obtainance of similar datasets for the rest of the regions of Austria, enabling the plaform to deliver parcel-based reports beyond its initial focus on Upper Austria, scaling from regional to national level. If the dataset tables were standardized, the scope of the analysis capabilities in the current state would easily gain significance even on multi-national and global scale. In addition, considering other environmental factors, such as vegetation cover, would offer a more detailed insights into the current state of the parcel units. Furthermore, taking into account the analysis of incorporated datasets could derive new comprehensions, such as assessing the photovoltaic potential based on topographical characteristics.

Another powerful asset lies in the potential of introducing the machine learning alghorithms which would enable the compilation and analysis of interconnected environmental conditions, offering deeper comprehension and predictive capabilities. Moreover, replacing a rule-based approach would provide contextual information in dynamic, more adaptable and truthful manner regarding the parcel's state.

6. Conclusion

Distinguishing it from conventional methods, ParcelInsights platform offers a swift and intuitive procedure to reach a report for the parcel of user's interest, avoiding physical interaction, costs and time consumptions for the ones requesting and providing reports. By integrating various open-source technologies it is evident that one does not need to rely on desktop-based proprietary software, making the platform ready for deployment to be widely accessible to anyone at any time. As a widely accessible tool, it has the potential to revolutionize the communication regarding the environmental information for informed decision-making

Acknowledgement

I would like to express my appretiation to Manfred Mittlböck, Senior Lecturer, for his support throughout the project by providing expertise and guidance, particularly in the aspect concerned with OGC Web Services. I would also like to extend my appretiation for the continous support provided by ChatGPT (OpenAI) in solving programming tasks throughout the development of this project. The guidance ChatGPT has provided has significantly alleviated the burdens and reduced the time that would be greatly consumed by using conventional search engines.

References

[1] A. Geymen and T. Yomralioglu, "Spatial data-based e-municipality applications," *Proceedings of The Institution of Civil Engineers-municipal Engineer - PROC INST CIVIL ENG MUNIC ENG,* vol. 163, pp. 77-88, 01/06 2010, doi: 10.1680/muen.2010.163.2.77.

[2] V. R. Prybutok, X. Zhang, and S. Ryan, "Evaluating leadership, IT quality, and net benefits in an e-government environment," *Information & Management,* vol. 45, pp. 143-152, 04/01 2008, doi: 10.1016/j.im.2007.12.004.

[3] OpenLayers. "OpenLayers: v6.4.3." OpenLayers. https://openlayers.org/ (accessed 27.3.2023.)

[4] M. Grinberg, "Flask web development: developing web applications with python," *O'Reilly Media, Inc.,* 2018.

[5] T. Kralidis and S. Gillies. "OWSLib: v0.28.1." OWSLib. https://github.com/geopython/OWSLib (accessed 27.3.2023.)

[6] K. Jordahl *et al.* "Geopandas: v0.12.2." Zenodo. https://geopandas.org/en/stable/ (accessed 27.3.2022.)

[7] S. Gillies. "Rasterio: v1.3.6." Mapbox. https://rasterio.readthedocs.io/en/stable/ (accessed 27.3.2023).

[8] I. ReportLab. "ReportLab: v4.0.0." https://www.reportlab.com/ (accessed 26.3.2023.)