

Developing Cloud Based Forms Builder for GIS Data Collection using Open-Source Technologies

Mohammed Mahmoud Ibrahim Musa

Faculty of Computer Science and Information Technology, Alzaiem Alazhari University, Khartoum North, Sudan

Dr. Ashraf Osman Ibrahim

Arab Open University, Khartoum, Sudan

Dr. Eltaib Saeed Mohamed Ganawa

Faculty of Geographical and Environmental Sciences, University of Khartoum, Khartoum, Sudan

Abstract:

Location information has become an important part of information systems due to the fact that more than 80% of information is linked to a location on the earth's surface. Collecting geospatial information requires special tools, technologies, and methods. Most data collection platforms are designed without major consideration for GIS data, since GIS data requires a special type of database management system, and most of these platforms use non-spatial databases, so spatial objects are not stored properly, especially for polygon and line features. The objective of this study is to develop Form Builder platform for GIS data collection by using open-source technologies. The study highlights the power of open-source technologies in GIS applications as well as importance of GIS database for Form Builder applications. The cloud-based form builder application designed to be simple and easy to use for non-GIS users to design GIS data collection forms, visualize collected data and export collected data as shapefiles to be used in other GIS software for further processing or analysis. The cloud-based Form builder is a web-based GIS application that enables people to design their GIS data collection forms, visualize collected data, and export collected data to different GIS formats. Open-source technologies used in all aspects of the development of the cloud-based form builder application, Django and Geodjango as a python based web framework used to develop the form builder application, PostgreSQL and PostGIS database systems used as geospatial database management systems hosted on the cloud, GDAL as a geospatial data engine used for reading and writing GIS data from and to the geospatial database, Geopandas with leaflet API used for visualization of GIS data on web pages.

Keywords: Cloud Computing, Open-Source GIS, Form Builder, GIS Data Collectio

تطوير منشئ نماذج قائم على الحوسبة السحابية لجمع بيانات نظم المعلومات

الجغرافية باستخدام التقنيات المفتوحة المصدر

أ.محمد محمود إبراهيم موسى - جامعة الزعيم الأزهري

د. أشرف عثمان إبراهيم - الجامعة العربية المفتوحة

د. الطيب سعيد محمد غناوه - جامعة الخرطوم

المستخلص:

الموقع الجغرافي أصبح جزء مهم في نظم المعلومات، حيث ان أكثر من 80 % من المعلومات مرتبطة بشكل مباشر او غير مباشر بموقع معين على سطح الارض. جمع المعلومات الجغرافية او المكانية يتطلب مجموعة معينة من الادوات والتقنيات والاساليب. معظم المنصات الخاصة بجمع البيانات تم تصميمها دون مراعاة للاشكال المختلفة للمعلومات الجغرافية، حيث يتم استخدام قواعد بيانات غير مكانية في الغالب، وما ان تخزين البيانات الجغرافية يتطلب قواعد بيانات معينة تدعم تخزين وإدارة البيانات الجغرافية فان عملية تخزين البيانات في هذه المنصات تتم بصورة غير فعالة خاصة عند تخزين المعالم الخطية والمساحية. تهدف هذه الدراسة الى تطوير منصة لانشاء نماذج لجمع بيانات نظم المعلومات الجغرافية باستخدام التقنيات المفتوحة المصدر، حيث تسلط هذه الدراسة الضوء على قوة التقنيات المفتوحة المصدر في تطبيقات نظم المعلومات الجغرافية وكذلك اهمية استخدام قواعد البيانات المكانية في منصات انشاء نماذج جمع البيانات. تم تصميم منشئ النماذج القائم على الحوسبة السحابية ليكون بسيطاً وسهل الاستخدام بالنسبة للمستخدمين الغير محترفين في التعامل مع نظم المعلومات الجغرافية، حيث يمكن التطبيق للمستخدمين من تصميم نماذج جمع بيانات نظم المعلومات الجغرافية، عرض بيانات التي تم تجميعها، وتصدير البيانات التي تم تجميعها في شكل ملفات shapefile حتى يسهل استخدامها في برامج نظم المعلومات الجغرافية الاخرى لاجراء عمليات المعالجة والتحليل عليها. منشئ نماذج جمع بيانات نظم المعلومات الجغرافية هو تطبيق ويب يسمح للأفراد بتصميم نماذج جمع البيانات الخاصة بهم، حيث تم استخدام التقنيات المفتوحة المصدر في جميع جوانب تطوير منشئ النماذج، تم استخدام Django و Geodjango كإطار لتطوير تطبيقات الويب قائم على لغة البايثون Python في تطوير تطبيق منشئ النماذج، كما استخدمت انظمة قواعد البيانات PostgreSQL مع PostGIS لتخزين بيانات نظم المعلومات الجغرافية، وتم استخدام محرك البيانات الجغرافية DGAL لقراءة وكتابة البيانات الجغرافية من والى قواعد البيانات المكانية، وتم استخدام المكتبة البرمجية Leaflet مع Geopandas لعرض بيانات نظم المعلومات الجغرافية على صفحات الويب.

الكلمات المفتاحية: الحوسبة السحابية، نظم المعلومات الجغرافية المفتوحة المصدر، منشئ النماذج، جميع بيانات نظم المعلومات الجغرافية.

Introduction:

Cloud computing technology increasingly become an important solution for organizations that looking to drive their business forward. The NIST defined Cloud computing a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (1). In Cloud Computing, computing resources are provided as on-demand services over the internet, these services delivered in three models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), Google Doc is an example of SaaS, Firebase Database is an example of PaaS while the Amazon EC2 is an example of IaaS. Organizations using cloud computing technology to consume cloud-computing services according to their needs. According to the NIST, there are four-deployment model for cloud computing private cloud, community cloud, public cloud and hybrid cloud. Recently, Geospatial Technology are increasingly provided on cloud computing, which provides dynamically scalable GIS technology, GIS Data and GIS functions as a web service. GIS Data collection considered as one of the most expensive activities in GIS applications (2). Geospatial data collection may include remote sensing data, field data and other in-house GIS data conversion processes. Traditional field data collection (i.e. pen-and-paper based) is a time consuming and bulky task. For example, we need to prepare base maps, collect an ancillary dataset, and other paperwork. This is not practical to use in real-time disaster information collection, which occurs in unpredictable places and requires a quick emergency response (3). The use of Open-Source Web Mapping tools is still today quite limited due to their higher required skills with respect to APIs and only occurred in few literatures' studies (4) (5) (6). In this paper aims to develop cloud-based form builder for GIS data collection using open-source technologies to enable people to

design their GIS data collection forms on the cloud using Open-source technologies, through Form Builder application user can visualize collected data as tables, maps as well as export collected data as shapefiles.

Related Work:

Survey Builder tool, initiated by the Career Resource Center (CRC) and College of Engineering in order to develop a dynamic web-based survey tool, The Survey Builder application designed for the University of Florida is a tool that lets users create surveys, administer them, and analyze survey results all through the web. It allows any department to create a survey and administer it over the web. Questions can be Likert scale, yes/no, short answer or open response. Each question has a unique ID and can be reused in subsequent surveys. Colleges have the ability to develop standard questions that are included in all departmental surveys. All results are stored in a database accessible by the department that creates the survey (7). Survey builder provides a lot of features, including Create Survey, Administer Survey, conduct survey, view survey results and link survey but all these features do not consider the location information.

Harvard Humanitarian Initiative in association with Brigham and Women's Hospital, USAID have designed Kobo toolbox an open-source data collection tool, researcher can build a questionnaire choosing various patterns in the Kobo Toolbox website. Once the questionnaire is organized, the survey can run through the android application 'Kobo collect' for data collection by downloading it without any expenditure. Kobo toolbox provides Form Builder, Question library, Data collection via android or Web browser, manage, download and analysis of the data, Access control (8). Although Kobo toolbox and Kobo collect are widely used in humanitarian crisis like natural disasters, natural calamities, manmade disasters etc. Kobo toolbox has some limitation, the database system used in

Kobo toolbox is not spatial database which led to some limitations when designing data collection forms for GIS data.

Methodology:

Cloud Based Form Builder for GIS data collection will be developed using Open-source technologies, Figure 1: shows the general architecture of Cloud based form builder for GIS data collection as well as software and tools used in the development. PostgreSQL alongside with PostGIS extension will be used as Cloud GIS database, GDAL as Spatial Data engine will be used for reading and writing to GIS database, Django and Geodjango as python web framework following (Model, View, Template) MVT pattern will used to develop the Cloud based web GIS application of the Form Builder, Geopandas and Leaflet API will used to visualize and export collected GIS data.

Cloud based form builder is a web-based application that enable users to create their own data collection form for GIS, in form builder users can specify type of GIS data will be collected points, lines or polygons. To create form user must be registered, and registered users need to add login details to login to the cloud form builder, once user logged in user can create data collection form, all form created by user will be linked to that user, once user created a form the form details and related fields will be stored on cloud GIS database and a new spatial table will be created on cloud GIS database based on form details. Through form builder users can add enumerators by adding enumerators details these details will be stored on cloud database in enumerators table, enumerator can use his username and password to login and conduct data collection.

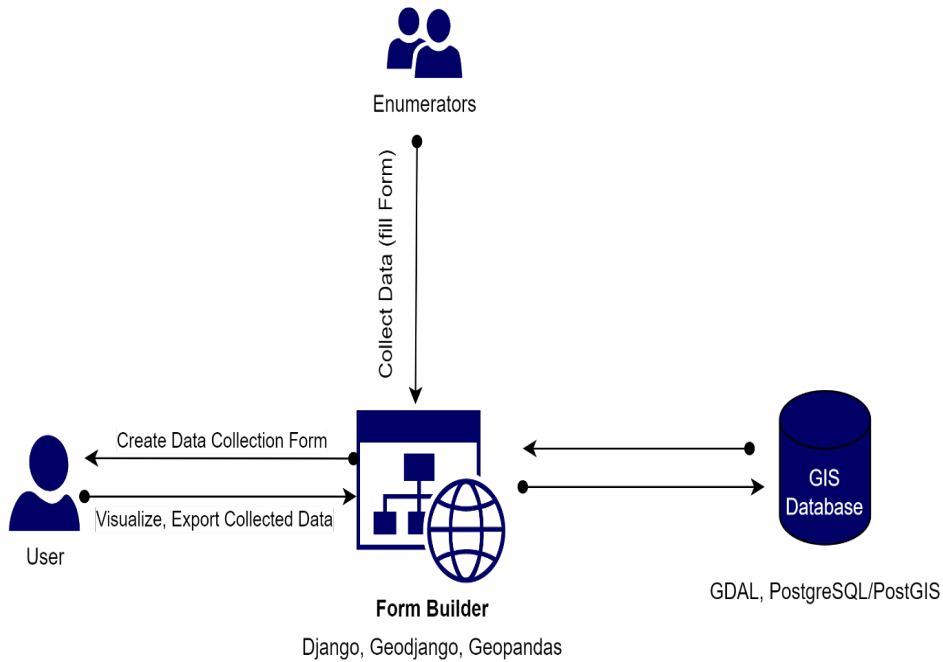


Figure 1: General Architecture of Cloud Based Form Builder

Since the idea for Cloud based form builder based on create a table for each data collection form and each data collection form composed of a set of entry element each entry element is considered as a field (column) in the table. Two related tables created one for forms information and the other for the fields' information. The forms table include the following fields (FORM_ID, FORM_NAME, FORM_DESCRIPTION, GEOMETRY_TYPE, USER_ID). FORM_ID field is unique and auto incremented field. The FORM_NAME field must follow the naming rules, it must start with a letter (cannot start with number) and can only contain alpha-numeric characters and underscores. Since the FORM_NAME field is constrained field, the FORM_DESCRIPTION field allow users to add a brief description to the form. The GEOMETRY_TYPE field allow the user to specify the type of geometry (Point, Polyline, and Polygon) for the Geospatial Data, if there is no need for spatial data, the user can leave this field blank. The

USER_ID field used to specify the ID of the user who have created the form. Name of data table related to the form composed of FORM_NAME and FORM_ID to avoid the conflict. Since the platform is cloud based and used by different users from different areas and sometimes users might use the same name for their form, and it is not allowed to have two table with the same name in database system. The fields table include the following fields (FIELD_ID, FORM_ID, FIELD_NAME, FIELD_LABEL, FIELD_TYPE), FIELD_ID field is unique and auto incremented field. The FORM_ID used to specify the ID of the Form related to the field. The FIELD_NAME field must follow the naming rules. Since the FIELD_NAME field is constrained field, the FIELD_LABEL field allow users to add a brief description to the field. The FIELD_TYPE field allow the user to specify the type of data (integer, float, text), field name used as column name in the related form's table and the field type to specify the data type for the column.

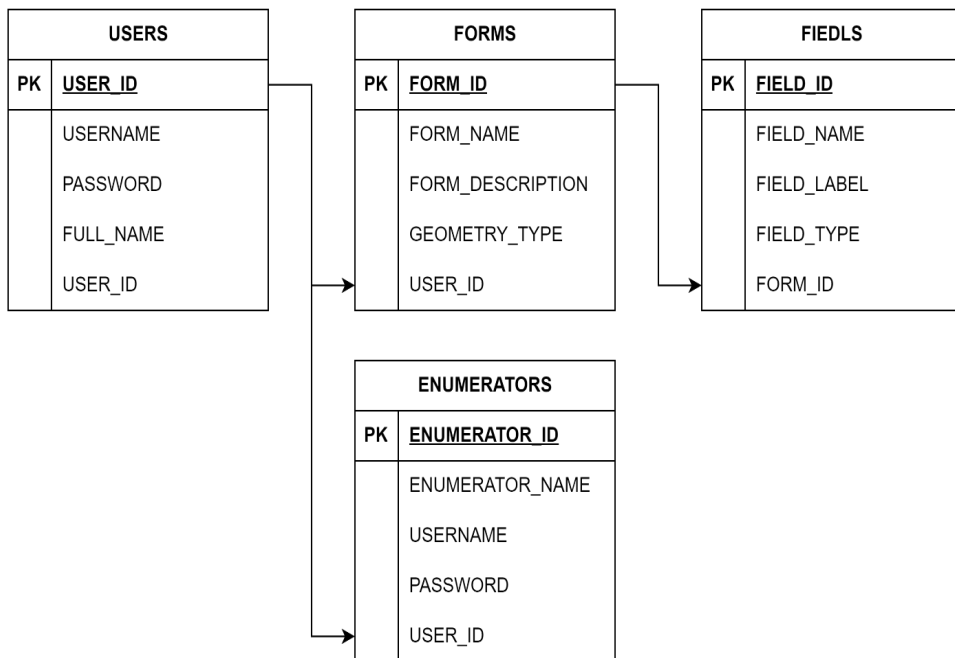


Figure 2: General Schema of Cloud GIS database

Once a user creates a form, form details will be stored on forms table and related fields details will be stored on field tables. Afterwards, a data table related to the form will be created. The data table name will be composed of form name and form id to avoid duplication and conflicts. Field names will be used as column names for the data tables while field type will be used as data type for each data type for each column. Figure 3: shows the structure of the database.

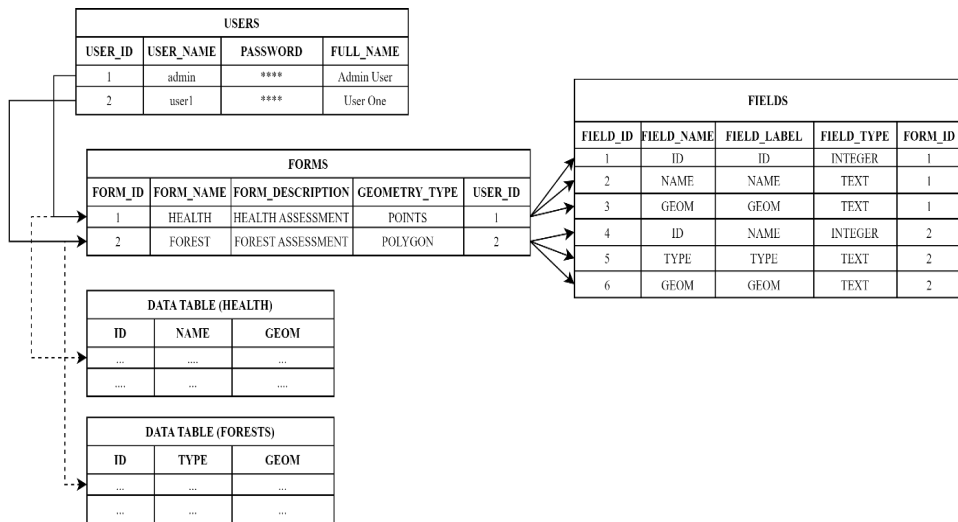


Figure 3: Database Structure

Results

The cloud-based form builder for GIS data collection platform developed to test the proposed synchronization algorithm, as presented in Figure 1, PostgreSQL/PostGIS is used as GIS database for cloud platform, and GDAL used as engine for reading and writing data in GIS database, most of data collection platform using non-spatial database which make storage of spatial object is difficult, for example in Kobotool box one of the most common data collection platform, spatial information for points data stored by adding two columns (longitude and latitude) to store the coordinate of points, when it comes to line and polygon features it is difficult to represent such spatial data in longitude and latitude column, be-

cause line and polygon features composed of multiple points, and it does not make sense to create many longitude and latitude columns to store the coordinates of these points, using PostGIS as a database for the platform enable users to create form with various spatial data types such points, lines and polygons. Form Builder web application built by using Django a python-based web framework, Geodjango as add-on for Django used to create GIS data model, Geopandas used to extract, transform and export GIS data to various format, and Django rest framework used to create rest web services. the main page of Form Builder web application enables users to create a new account or login if the user already registered, User can create an account on the platform by using sign up button, sign up button will open a new page which enable users to create a new account by adding their username, email address, and password. Users' details will be stored on user table within cloud database. Figure 4: shows the main page of the platform.

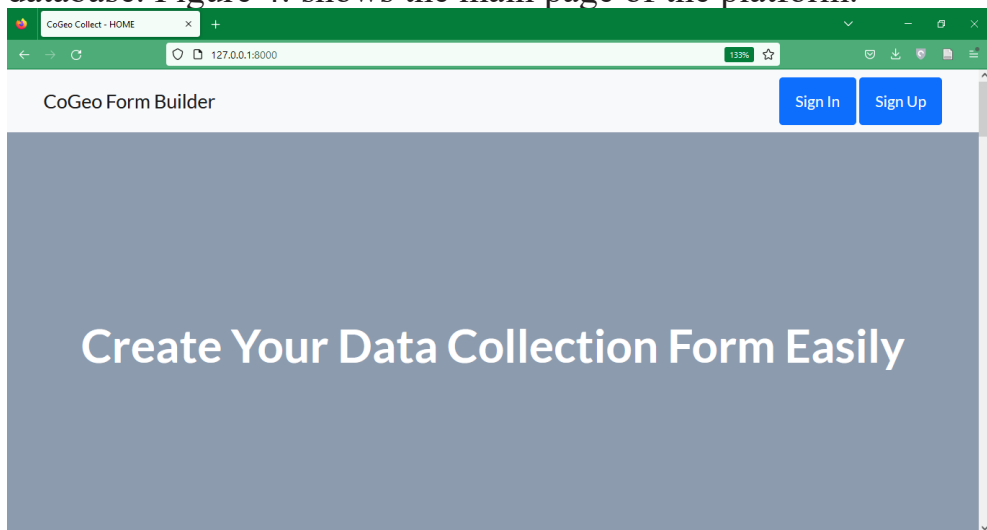


Figure 4: Home Page of the Form Builder platform

Sign up page enable users to create a new account, the sign-up page designed to be as simple as possible, user only need to add the username, email and password. Figure 5: shows the sign-up page.

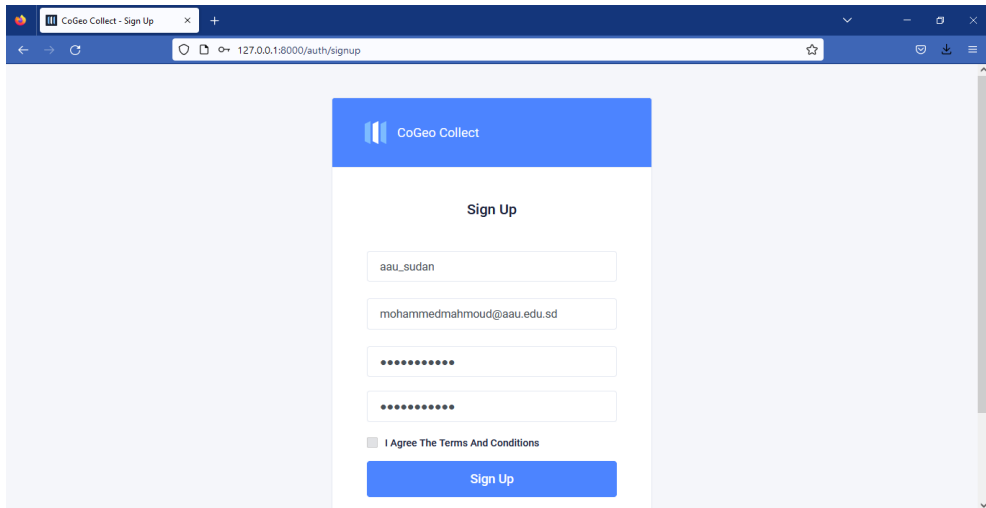


Figure 5: Sign Up Page

If the user already registered, user can use sign in button to sign in to the platform, in sign in page user should add the username and password to sign in, entered username and password will be checked across user table if the user is existing with the same details the user will be signed in. Figure 6: shows the sign-in page.

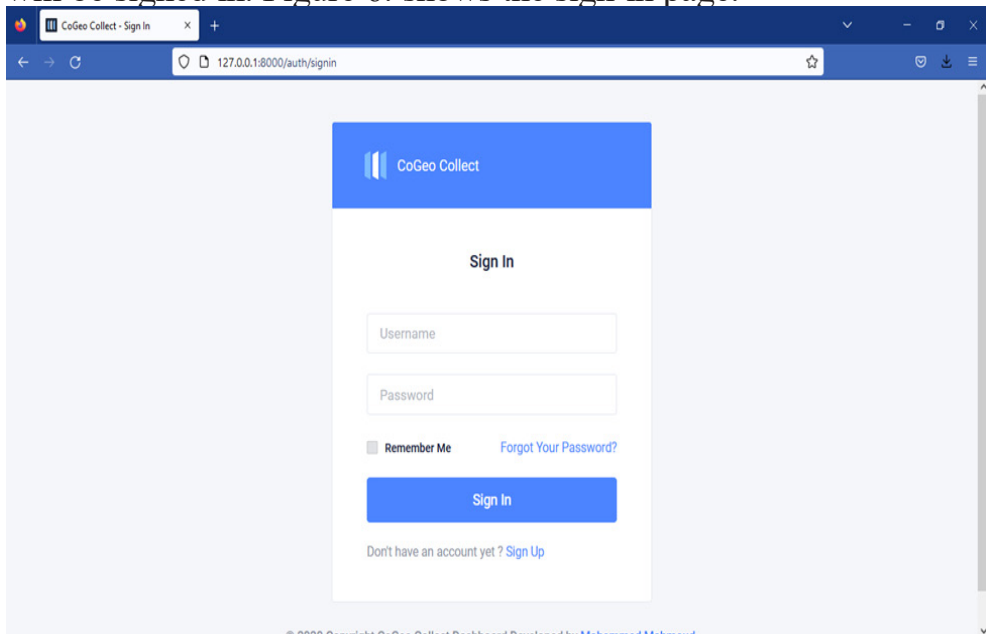


Figure 6: Sign in Page

Once user signed in, form list page will appear to the user, the form list page shows the list of forms created by the user, in addition to a button to enable user to create a new form, form list will be retrieved from the forms table on cloud database. Figure 7: shows the form list page. In Form List Page, users can see created from in addition to buttons to manage the forms, these buttons include button to delete the form, button to view the form and button to edit the form, once the user click on delete button the form details and related fields will be deleted from forms and fields tables on cloud database as well as the data table related to the form will be deleted from the database. Figure 44: shows the Form list with options to view, edit or delete form.

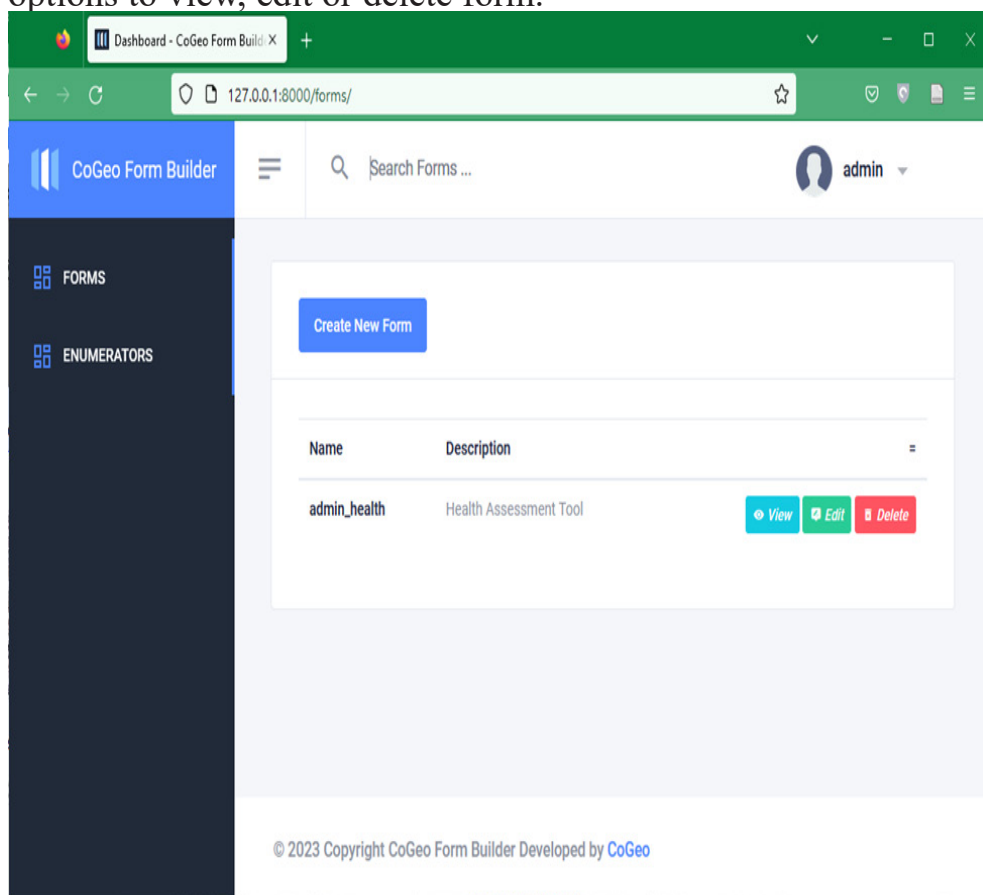


Figure 7: Form List page

Create a new form page will enable users to create their data collection form, user should add form details which include the form name, form description, and specify the geometry type. And addition to that user can add fields to the form, field details include, field name, field label and field type. The form details would be stored on the forms table in cloud database, while field details will be stored in fields table in cloud database. Once the user clicks on create form button a new data table will be created for the form, the data table name composed of username of the form creator and the form name while the columns of the table will be the fields of the form. Figure 8: shows the create new form page.

The screenshot shows the 'Create Data Collection Form' interface. The sidebar on the left has a 'FORMS' section. The main form area includes:

- Form Name:** A text input field containing 'forest'.
- Geometry Type:** A dropdown menu currently showing 'POINT'.
- Form Title:** A text area containing 'Forest In Sudan'.
- Field Table:** A table with three columns: 'Field Name', 'Field Label', and 'Field Type'. It contains one row: 'forest_name', 'Forest Name', and 'INTEGER'.
- Buttons:** 'Add Field' (green), 'Create Form' (blue), and 'Remove' (red).

Figure 8: Create Form Page

View Form details page enable user to collect data through web pages, user can click on Open Form button to open data collection form and fill the form with data, in addition to that view

form details enable users to view collected data in a form of table or map and export collected data to shapefiles, which standard format of GIS data supported by various GIS software. Figure 9: shows Form details page.

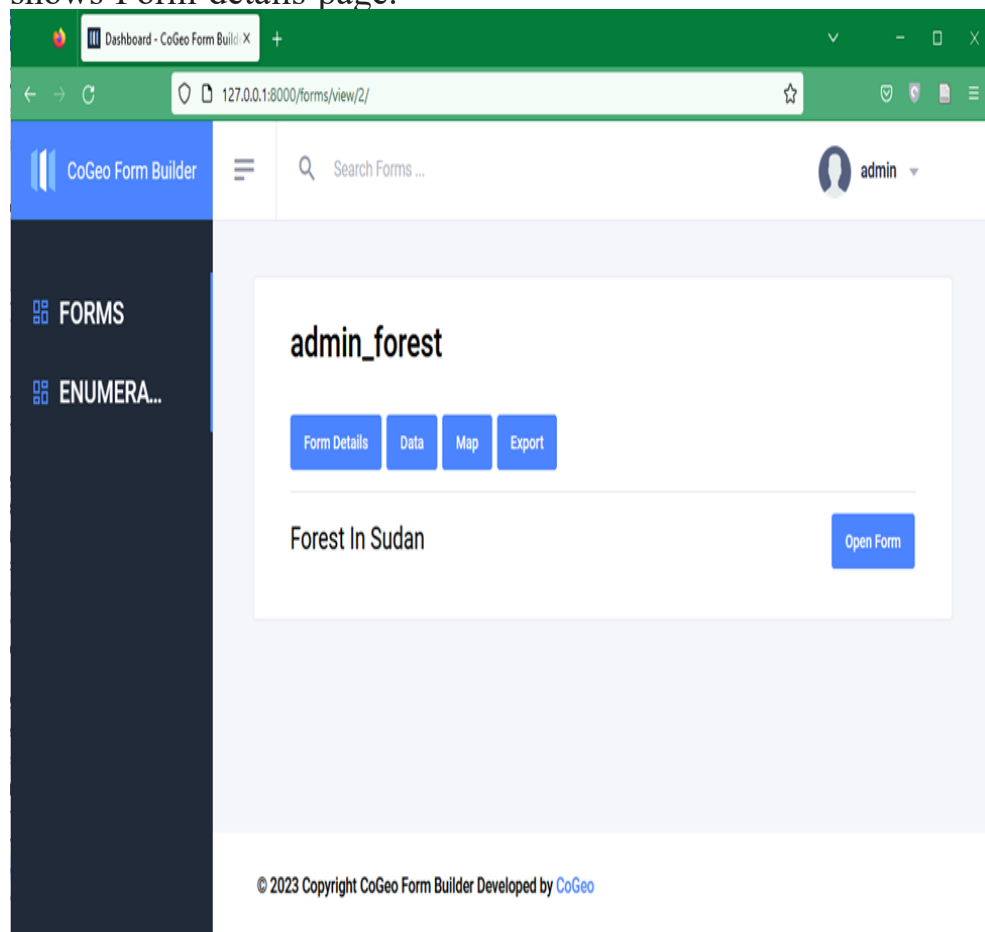


Figure 9: View Form Details Page

Enumerators can open Collect Data page using web browser on their smartphones to collect data enumerators can fill forms' fields and draw the geometry related to the form, collected data will be stored on data table related to the form. Figure 10: shows Data collection web page.

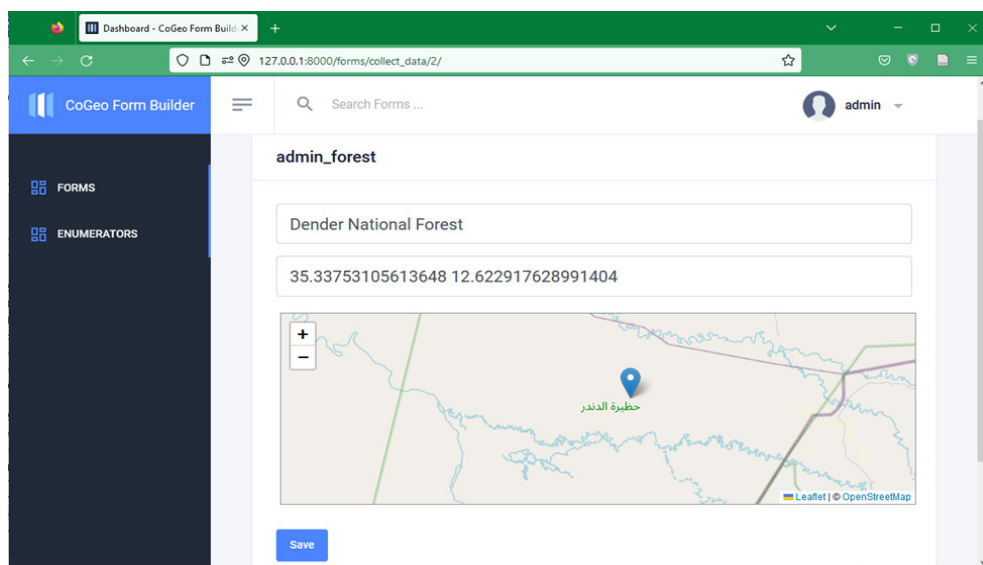


Figure 10: Collect Data Web Page

Data view page shows the collected data in form of table, user can open data view page by using data button on view form details page. The data retrieved from data table related to the form. Figure 11: shows data view page.

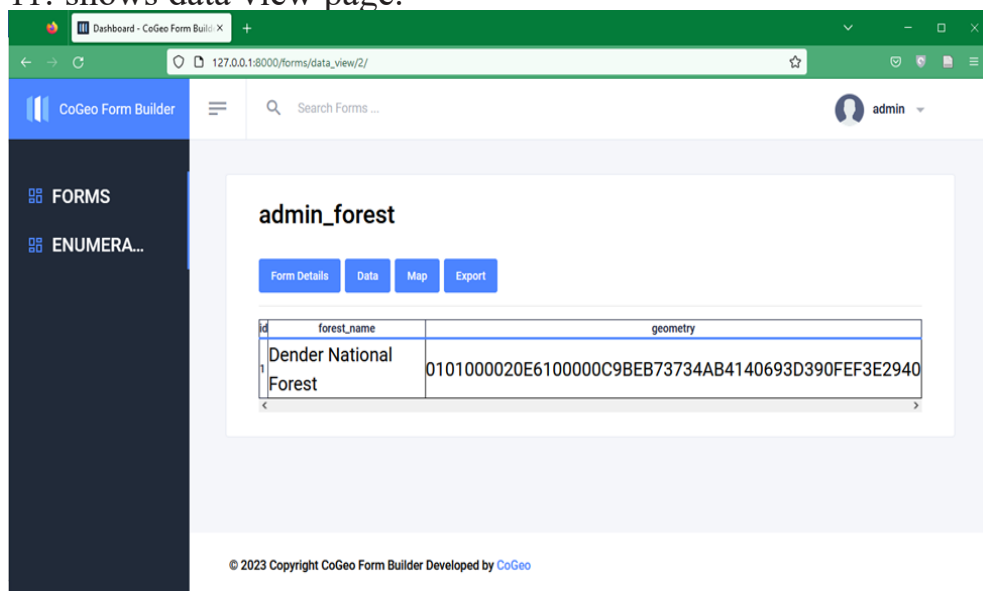


Figure 11: View Data as Table Page

Map view page shows collected data on a map, user can open Map view page by using Map button on form details page, the data retrieved from data table related to the form based on geometry value type and value the map displayed, Leaflet API used as an API for Web Mapping and Open Streets Map used as base map for map view. Figure 12: shows Map View page.

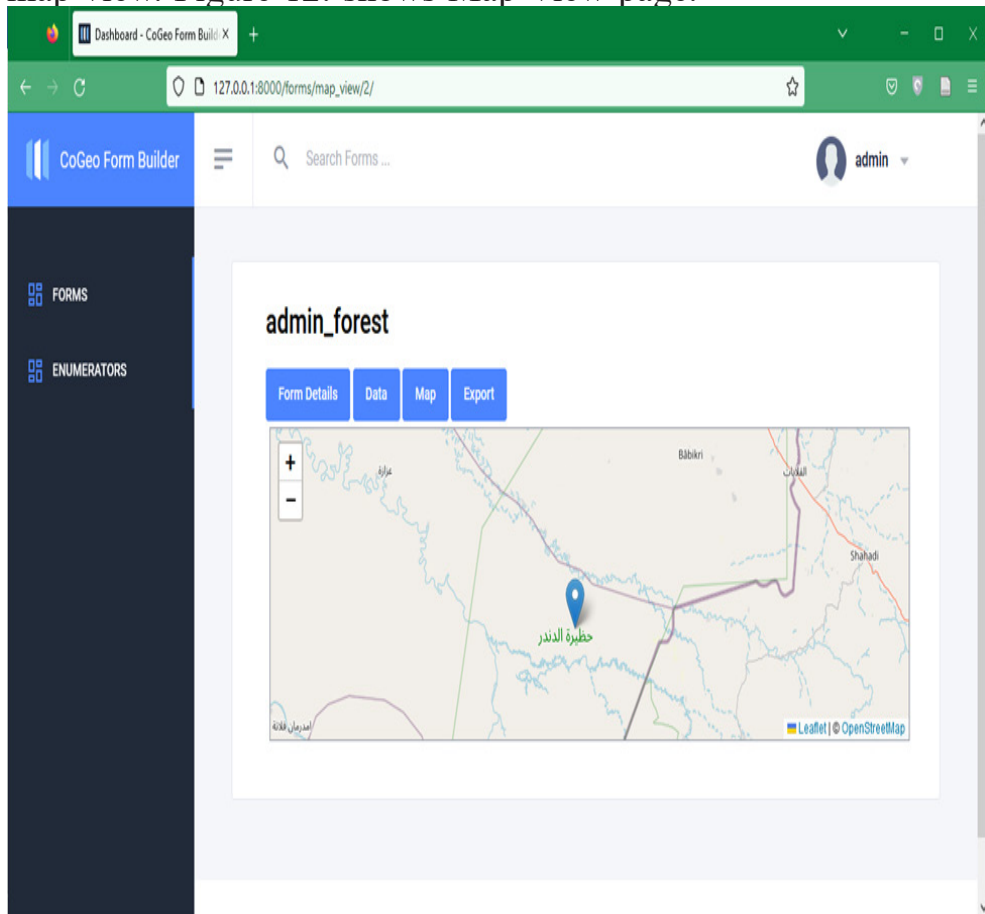


Figure 12: View Data as Map Page

Export data page enable users to export collected data as shape-file, collected data retrieved from data table related to the form and converted to shapefile by using Geopandas. Figure 13: shows export page.

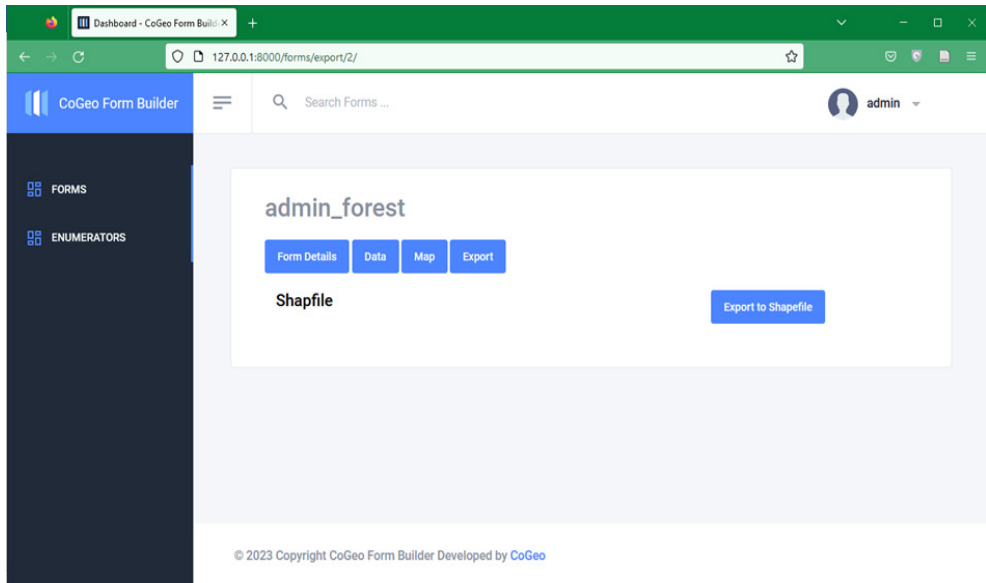


Figure 13: Export data as Shapefile

Enumerators list page shows the list of enumerators added by user, from the enumerators list page user can delete existing enumerators or add new enumerators, user can delete existing enumerator by using delete button, and can add a new enumerator by using Create Enumerator button. Figure 14: shows enumerators list.

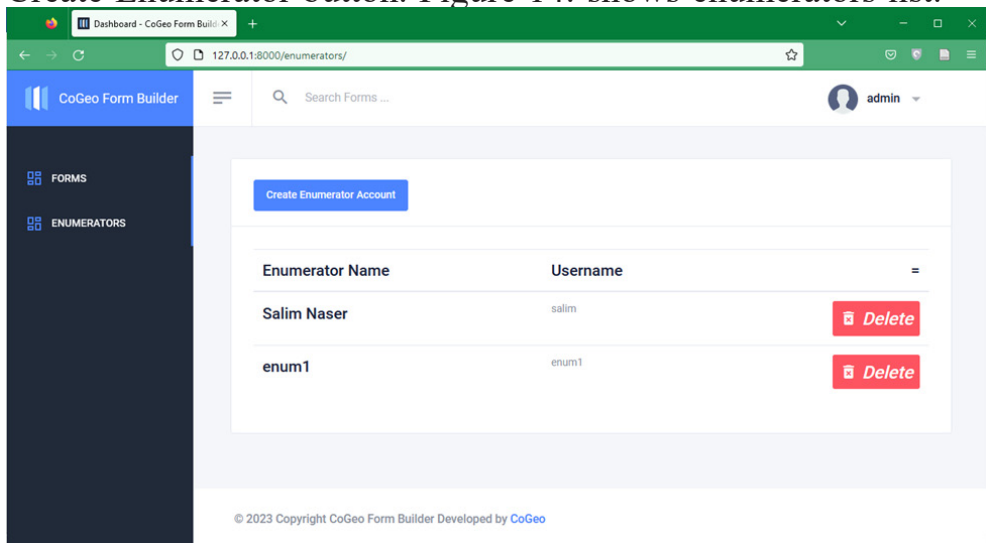


Figure 14: Enumerators List Page

Create Enumerator page enable user to add and account for enumerator by adding Enumerator name, username and password. Enumerator details stored on Enumerators table on cloud database. Figure 15: shows create new enumerator page.

The screenshot shows a web application interface for 'CoGeo Form Builder'. The browser's address bar shows the URL '127.0.0.1:8000/enumerators/new'. The application has a dark blue sidebar on the left with a menu icon and two visible options: 'FORMS' and 'ENUMERA...'. The main content area is white and titled 'Create Enumerator'. It contains three text input fields, each with a label above it: 'Enumerator Name', 'Username', and 'Password'. Below these fields is a blue button labeled 'Create Enumerator'. The top of the application has a green header bar with the 'CoGeo Form Builder' logo, a search bar, and a user profile icon labeled 'admin'.

Figure 15: Add Enumerator Account Page

Conclusion:

Since location become an important element in Information systems, an efficient tool to collect location information is highly needed, invention of Web technologies and cloud computing technology enable people do conduct collaborative collection of GIS data. Cloud Based Form Builder of GIS data collection is an integrated platform to enable people to create forms for GIS data collection. Form builder application is web-based application developed by Django/Geodjango framework to enable people to design their GIS data collection form, the form builder enables people to create a form for points, polylines and polygons, as well as visualizing and exporting collected data. PostgreSQL with PostGIS extension were used as cloud GIS database, using PostgreSQL/PostGIS as spatial database enable us to create different types of spatial tables which make the platform unique, whereas most of existing data collection platform use non-spatial databases which is lead to limitation on integration of collected data with GIS software. Interfaces of Cloud form builder application designed to be simple and easy to use, which enable non-GIS users to develop GIS data collection forms easily. Cloud Based form builder for GIS data collection developed by using open-source technologies which highlight the power of open-source technologies in GIS applications and how open-source technologies can mitigate the cost of GIS solution.

0.1.Recommendations:

To make Cloud GIS data collection platform more efficient, additional GIS functions must be added to enable users to conduct different GIS functions on collected data, it is recommended to integrate Cloud GIS Data collection platform with one of open sources GIS server such as Geoserver, MapServer. Integration of Cloud GIS data collection platform with GIS server enable data tables of collected GIS data to be shared as Web GIS services using different GIS service standards such Web Map Services (WMS) and Web Feature Services (WFS), which enable users to consume

collected GIS data in different applications and platforms which provide a real-time consumption of collected GIS data. It is also recommended to provide some query operations to enable users to conduct spatial and attribute-based queries on collected GIS data, these query operations also enable users to export collected GIS data according to the user needs. It is recommended to enable users to upload GIS data to Cloud GIS data collection platform such as Shapefiles, GeoJSON and KML, which allow users to integrate collected GIS data with other GIS data. It is recommended to develop a Mobile Application to enable enumerators to conduct the data collection activity through their smartphones.

References:

- (1) Mell P, Grance T. The NIST definition of cloud computing. 2011;
- (2) Longley PA, Goodchild MF, Maguire DJ, Rhind DW. GIS data collection. *Geographic Information Systems and Science*. 2001;32–7.
- (3) Lwin KK, Murayama Y. Web-based GIS system for real-time field data collection using a personal mobile phone. *Journal of Geographic Information System*. 2011;3(4):382.
- (4) Brovelli M, Minghini M, Zamboni G. Web-based Participatory GIS with data collection on the field—A prototype architecture. In: *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*. 2013. p. 5.
- (5) Brovelli MA, Minghini M, Zamboni G. Public Participation GIS: a FOSS architecture enabling field-data collection. *International Journal of Digital Earth*. 2015;8(5):345–63.
- (6) Xia D, Xie X, Xu Y. Web GIS server solutions using open-source software. In: *2009 IEEE International Workshop on Open-source Software for Scientific Computation (OSSC)*. IEEE; 2009. p. 135–8.
- (7) Hoit M. Survey Builder: A Tool to Support Assessment. In: *2004 Annual Conference*. 2004. p. 9–1142.
- (8) Lakshminarasimhappa MC. Web-based and smart mobile app for data collection: Kobo Toolbox/Kobo collect. *Journal of Indian Library Association*. 2022;57(2):72–9.