DIGITAL KNOW YOUR AUDIENCE RESEARCH APPLICATION

USER'S MANUAL

Version 1.0

04/15/2022

Revision Sheet

Release No.	Date	Revision Description
Rev. 0	4/15/22	User’s Manual Created

Table of Contents

List of Acronyms Used

1.	ODK	Open Data Kit
2.	DIGIKYAR	Digital Know Your Audience Research Application
3.	DB	Database
4.	HTML	HyperText Markup Language
5.	SCSS	Sassy Cascading Style Sheets
6.	CSS	Cascading Style Sheets
7.	XLS	Excel Spreadsheet
8.	XML	Extensible Markup Language
9.	ID	Identifier
10.	SPSS	Statistical Package for the Social Sciences
11.	CSV	Comma-Separated Values
12.	BSD	Berkeley Source Distribution
SQL		Structured Query Language

1 GENERAL INFORMATION

1.1 System Overview

The project seeks to enhance the capacity of Small and Medium Media Houses in Uganda in Audience Research by developing and deploying a Digital Know Your Audience Research Application (DIGIKYAR) to improve data collection and programming for better community sensitive programming that will cater for most community information needs, greater popularity for the community radio stations in the market, improved internal operations and revenue generation, for their sustainability. This will be a success because only one trained staff with a hand device can move on foot or motor cycle to the required sample size with custom question and carry out the audience research using the application and provide reports as and when needed by the media house.

The deployment of the Digital Know Your Audience Research Application (DIGIKYAR) will be accompanied by building the capacity of staff (journalists and management) from selected small and medium radio stations in Uganda on this Audience Research Application and basic principles of audience research as well as improving knowledge and skills on marketing and fundraising for sustainability of both the application and the radio stations. More still, Radio Apac will take lead in creating awareness about the relevance of audience research and adoption for community development sensitive programming.

This initiative will therefore solve the challenges of lack of in-depth investigative research and analysis of local issues among small and medium radio stations and lack of formal education among radio reporters on local developmental disciplines as well as lack of access to local institutions that can train reporters which often make the communities to remain ignorant about the issues affecting them amidst existence of many small and medium radio stations that can play a key role in educating and informing the communities about some of the local issues they face in the communities. In addition, enhancing the capacity of small and medium radio stations in Uganda will address the challenge of not being able to attract the service of qualified and professional journalists who have been trained in audience research principles at university degree or post graduate levels.

2 SYSTEM SUMMARY

2.1 System Overview

DIGIKYAR is a simple, robust and powerful tool data collection tool based on Open Data Kit (ODK) that seeks to enhance the capacity of Small and Medium Media Houses in Uganda to improve data collection and programming for better community sensitive programming that will cater for most community information needs, greater popularity for the community radio stations in the market, improved internal operations and revenue generation, for their sustainability. DIGIKYAR allows you to easily develop digital data collection forms that work on both mobile devices and web browsers.

2.2 The Digikyar Application Architecture

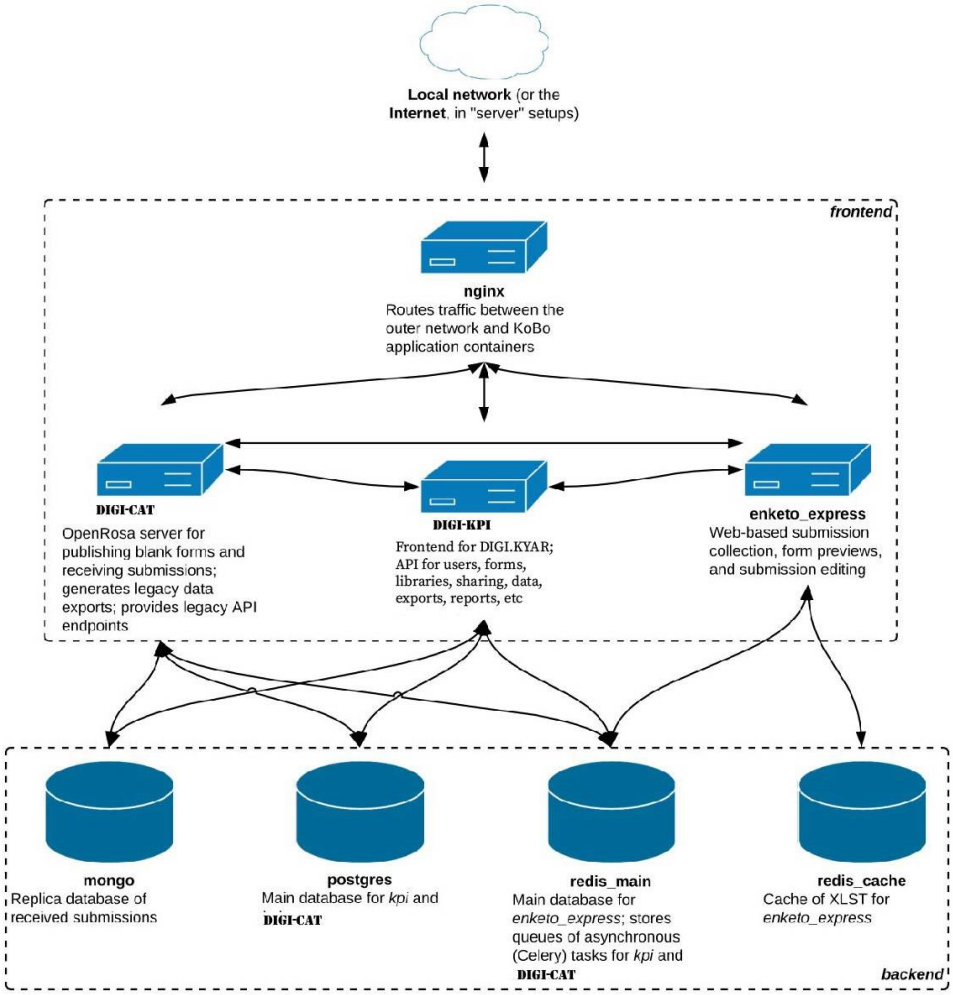


Figure 1:The DIGIKYAR Architecture.

2.3 Explaining the Digikyar Development Architecture

2.3.1 The Front-End

The front-end is what a user sees and interacts with (user interface). DIGI-KPI is the (frontend) server for DIGIKYAR.

- It includes an API for users to access data and manage their forms, question library, sharing settings, create reports, and export data on phones, tablets or any browser
- The tools used in building it are: Python, JavaScript, CoffeeScript, SCSS, TypeScript, and HTML.

2.3.2 The Back-End

The back-end is part of the application that is hidden from the user (what some would call, under the hood). This part is responsible for data processing, storing the data, and mathematical operations. The specialists responsible for the development of front-end and back-end parts of the application are called front-end developers and back-end developers accordingly. DIGI-CAT is our backend-server.

- It provides blank forms to Collect and Enketo and for receiving and storing submissions.
- The tools used in building are: Python, JavaScript, Cascading Style Sheets (CSS), HyperText Markup Language (HTML) Docker and the Shell

2.3.3 The Data Base

- This project is using 3 different databases
- Postgres, Mongo and Redis
- Postgres is the main data base for front-end and back-end

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need, it stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

PostgreSQL DB is a powerful, open-source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the POSTGRES project at the University of California at Berkeley and has more than 30 years of active development on the core platform.

Redis DB is an open source (BSD licensed), in-memory data structure store used as a database, cache, message broker, and streaming engine. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams.

2.4 DIGIKYAR Application Links

- Website: https://research.digikyar.org
- Accessing and sending data to the Digikyar Server via the ODK Android Application: https://dr.digikyar.org
- To register and create an account, click https://da.digikyar.org/accounts/register/#/
- To sign in click here https://da.digikyar.org/accounts/login/?next=%2F%23%2F#/

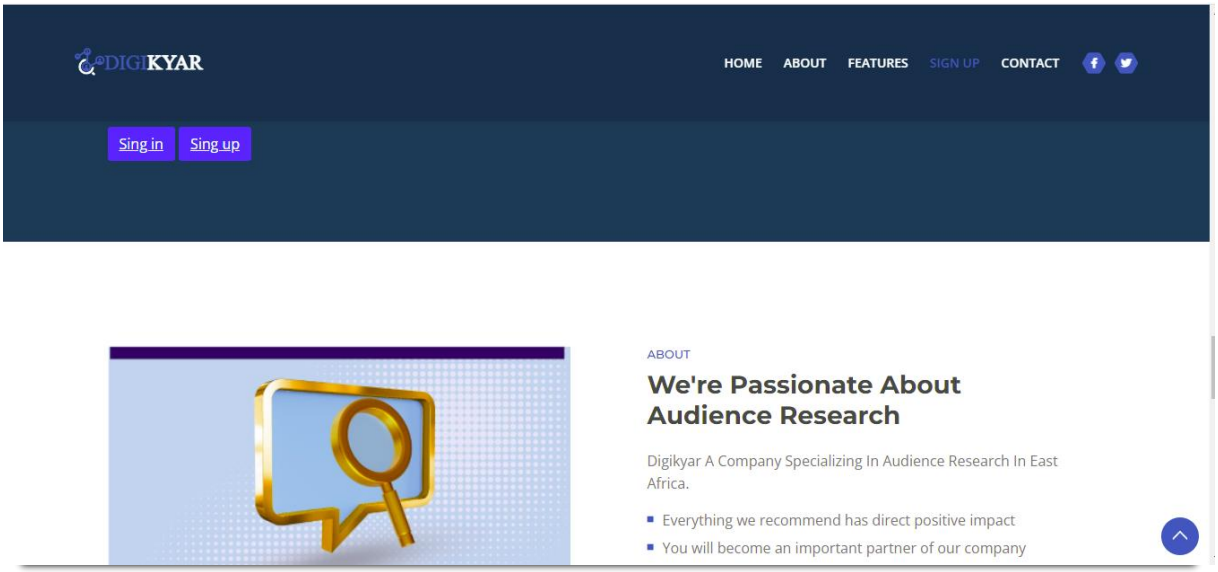


Figure 5: Sign Up, sign in and Create an Account on Digikyar.

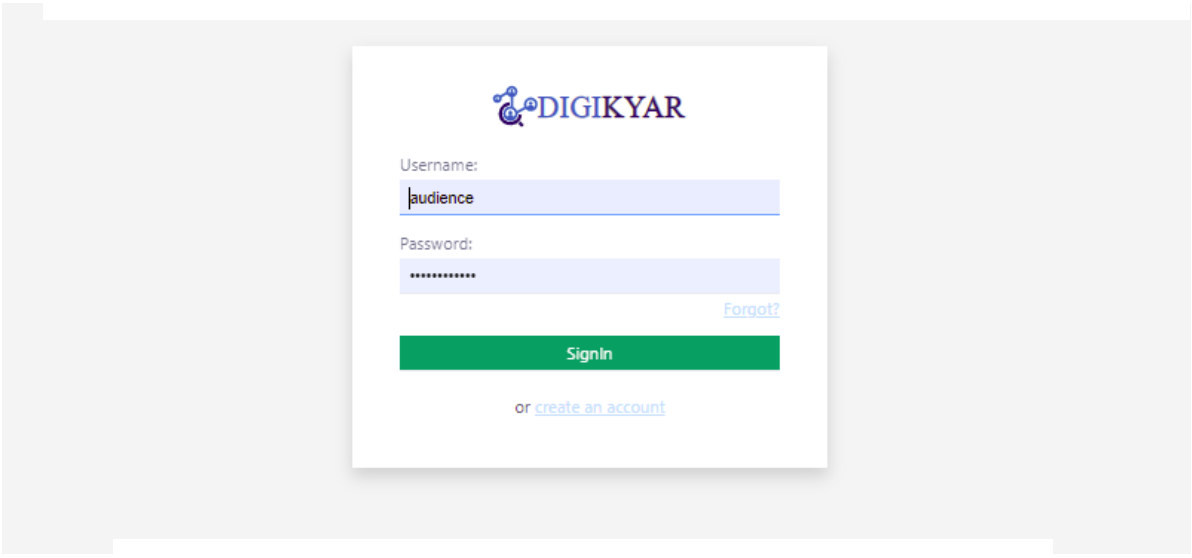


Figure 6: Login interface to the DIGIKYAR server.

2.5 DIGIKYAR Information and Support

The following DIGIKYAR private emails have been set up to provide updates/information and technical assistance to users. Various project teams have been put in copy to provide the necessary support.

a) Email: info@digikyar.org This has been setup to provide information, updates and feedback to DIGIKYAR users' inquiries. Radio Apac Team and netLabs!UG are all able to receive emails under this domain.

b) Email: support@digikyar.org This has been setup specifically to offer technical support to DIGIKYAR users. netLabs!UG team has been assigned under this domain.

c) Email: sales@digikyar.org This has been setup up for Radio Apac Limited sales and client support team.

2.6 Digikyar Website page

The major goal of the website page is to give users with a user-friendly interface for registering, creating accounts, and logging in to access the Digikyar server. It also includes explicit instructions on how to use Digikyar, as well as objectives, achievements, and updates concerning the Digikyar Application.

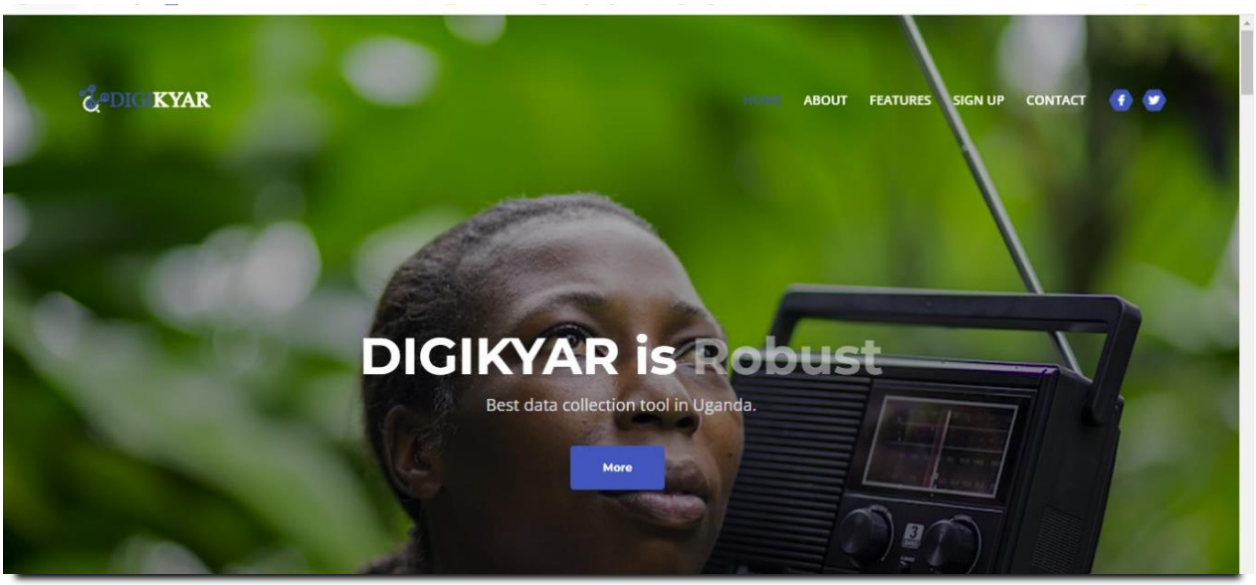


Figure 7:Digikyar website page.

3 GETTING STARTED

3.1 Creating an Account on Digikyar

The first step for all users when getting started with DIGIKYAR is to create an account. Below, we guide you through the steps on creating and accessing your account on the Digikyar servers.

To create a new account on Digikyar, go to the Digikyar homepage and click SIGN UP on the menu. This will take you to the GET STARTED section of the website. On the Create Your Account page, enter the details on the form and click Create Account.

Make sure to keep your password in a safe place to avoid losing it. The username must be all lowercase letters, without any spaces or symbols.

DIGIKYAR will send you an email with a link to activate your account. If you did not get the activation email, check your spam folder. For more help, you can also refer to *2.5.1 Didn't Receive Activation Link After Signing Up*.

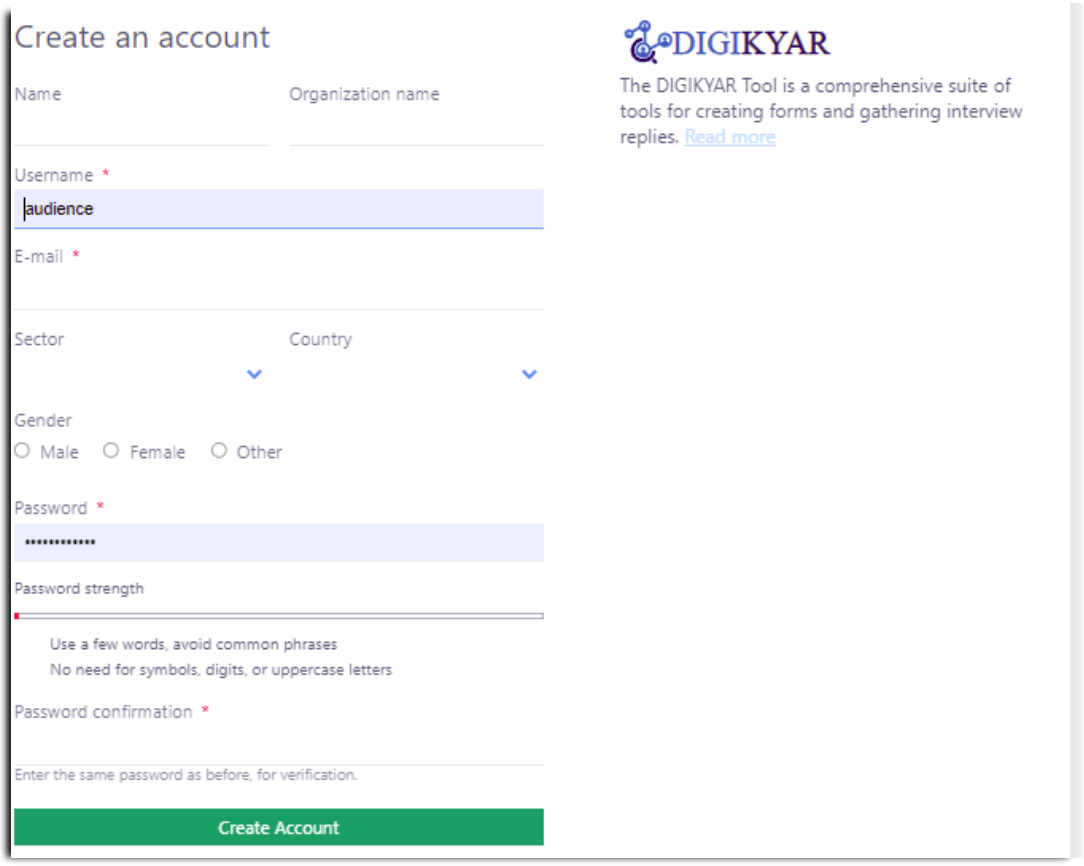


Figure 8: Digikyar get started page for registering a new account.

Log in to the email address you used when creating your account and click the activation link sent to you. This will automatically log you into **DIGIKYAR.**

3.2 Didn’t Receive Activation Link After Signing Up

After you create your account, you will need to click on the link that’s sent to the email address you provided within 72 hours of creating the account. If you don’t click on the link within three days, you’ll have to sign up again for a new account with a different username.

It’s possible you didn’t receive the email because you had a typo in your email address. Also, it’s recommended to check if the activation email landed in your spam or junk mail folder, as it might have been placed there by your email provider.
If neither of those work and you’re still not seeing an activation link email, please email support@digikyar.org

3.3 Troubleshooting your account

- Make sure to follow the guidelines listed below to create a valid username:
- The length of a username can range from 2 to 30 characters.
- A username could be made up of a combination of lower-case characters, numbers, and underscores. Use of special characters (such as ~, !, @, #, $, %, ^, &, *, ?) are not valid.
- The username must start with a lower-case character.
- You may be required to choose another username (despite following all the points listed above) if the username has already been registered to another user.
- Choose a strong password so that your account and sensitive data is less vulnerable to attack or abuse.
- Store your username and password in a safe place, such as a password manager, so that you don’t lose access to your account.
- Ensure the entered email address is correct otherwise you will not receive the activation link.

3.4 Creating a New Form or Questionnaire in DIGIKYAR

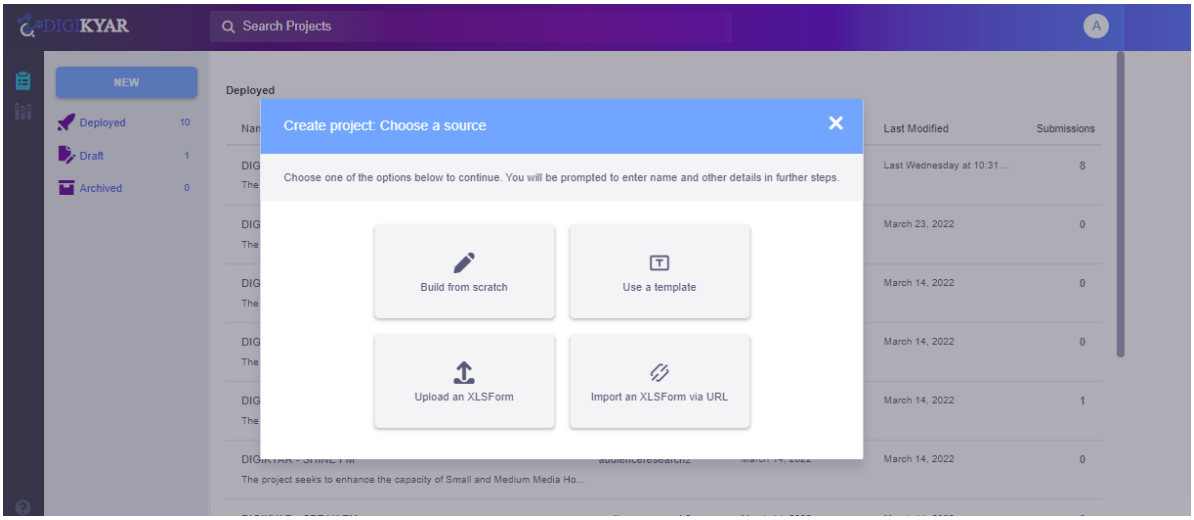


Figure 9: Options for creating a new questionnaire.

You can create a new form in four different ways. When you click on the New button, your options are:

1. **Build from scratch:** To create a new form from scratch, you can input project details and click Create Project.
2. **Use a template:** To use a template, you need to first go to Library and create one.
3. **Upload an XLSForm**: Import an existing form from an external file in the XLSForm format, for example when sharing form drafts between colleagues.
4. **Import an XLSForm:** via URL Enter a valid XLSForm URL to import an existing form. For instructions, visit Importing an XLSForm via URL.

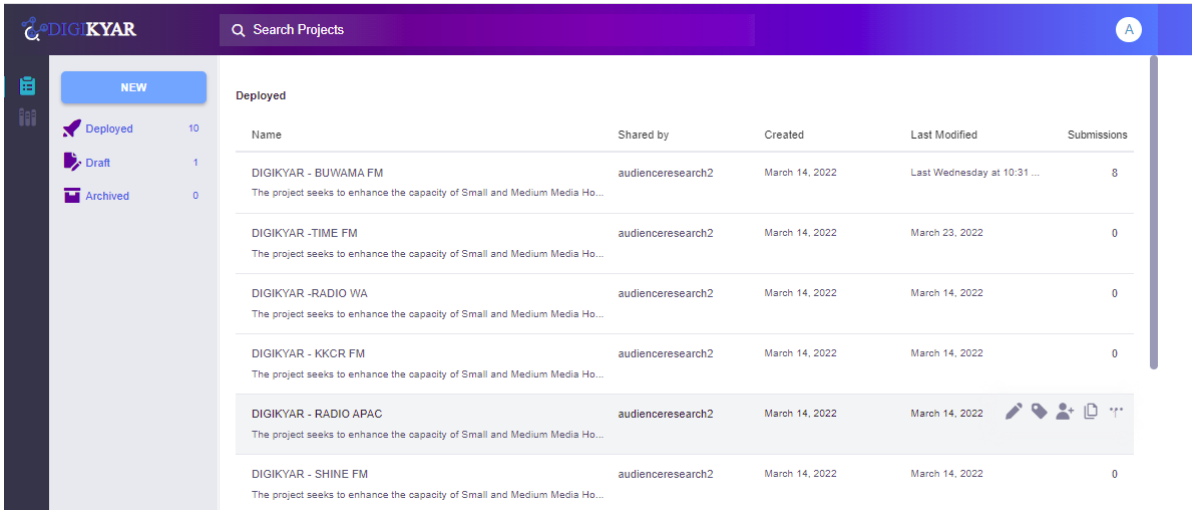


Table 1: List of forms for the project's created.

The Project List View shows all of the projects that have been created in, and shared with your account. To open a project, just click on its title. In this view, you can also organize your projects (add project tags, archive, or delete), download a project's form, or clone the form to use in a new project.

When creating very large forms (1000+ questions and high complexity), there is the possibility of the page hanging due to several reasons that are not limited to the browser, CPU, Internet connectivity, among others. In these cases, we recommend that you use XLSForm which is not constrained by this limitation.

4 BUILDING A QUESTIONAIRE FROM SCRATCH

4.1 Overview of All Form builder Functions

4.1.1 Adding a Question

Questions in your form can be added at any point by clicking on the button that is below every question. Type in the name of your question, click 'Add Question'. Then, choose the type that's applicable to your question.

4.1.2 List of Question Types

The below table provides a high-level summary of each of the response types available to use in your XLSForm and form builder:

Table 2: List of question types.

Question type	Answer input
integer	Integer (i.e., whole number) input.
decimal	Decimal input.
range	Range input (including rating).
text	Free text response.
select_one [options]	Multiple choice question; only one answer can be selected.
select_multiple [options]	Multiple choice question; multiple answers can be selected.
select_one_from_file [file]	Multiple choice from file; only one answer can be selected.
select_multiple_from_file [file]	Multiple choice from file; multiple answers can be selected.
rank [options]	Rank question; order a list.
note	Display a note on the screen, takes no input. Shorthand for type=text with readonly=true.
geopoint	Collect a single GPS coordinate.
geotrace	Record a line of two or more GPS coordinates.
geoshape	Record a polygon of multiple GPS coordinates; the last point is the same as the first point.
date	Date input.
time	Time input.
dateTime	Accepts a date and a time input.
image	Take a picture or upload an image file.

Question type	Answer input
audio	Take an audio recording or upload an audio file.
background-audio	Audio is recorded in the background while filling the form.
video	Take a video recording or upload a video file.
file	Generic file input (txt, pdf, xls, xlsx, doc, docx, rtf, zip)
barcode	Scan a barcode, requires the barcode scanner app to be installed.
calculate	Perform a calculation;
acknowledge	Acknowledge prompt that sets value to "OK" if selected.
hidden	A field with no associated UI element which can be used to store a constant.
xml-external	Adds a reference to an external XML data file.

Additionally, DIGIKYAR -specific types can also be used from within the form builder.

Formbuilder Question type	**Answer input**
Rating	Compare different items using a common scale.
Ranking	Compare a list of different objects to one another.
Question Matrix	Create a group of questions that display in a matrix format.

NB: Calculate Questions: are not displayed in your form, but are executed automatically as your form is being answered.
The Question Matrix Type: is only supported in Enketo and with the Grid Theme set.

4.1.3 Using the Question Options

After adding a question, there are many different customizations you can make to it using the question options. To get to the question options screen of a question, click its Settings button.

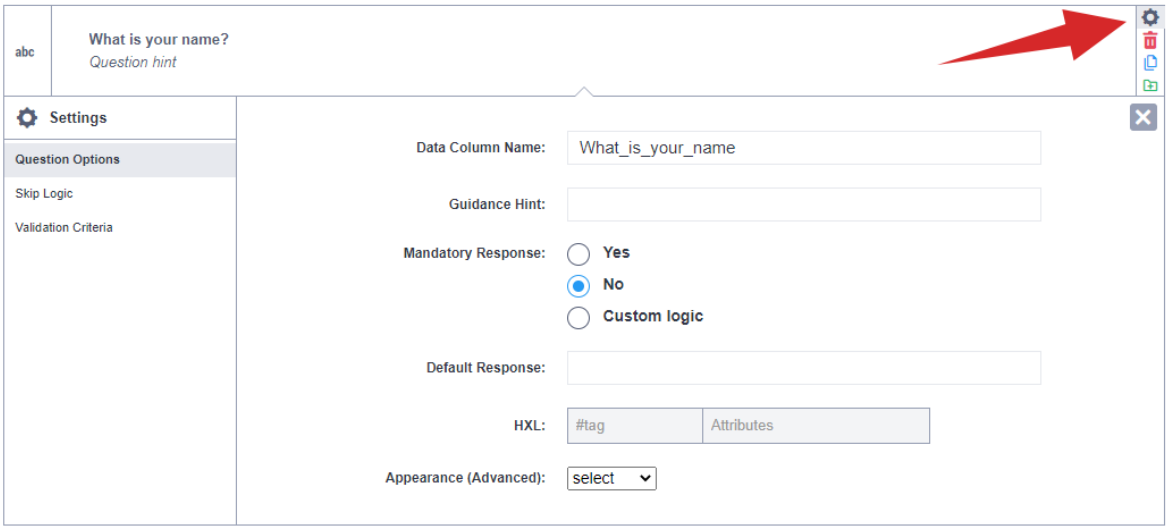


Figure 10:Using the question options.

4.1.4 Data Column Name

The Data Column Name is the unique identifier (ID) of your question.
This field is mandatory for every question. Only letters, numbers, and underscores are allowed in this field, and the field must start with a letter or an underscore. You can input anything you like, such as what_is_your_name or age.

The Data Column Name is important because it is used in the column headers of tables and spreadsheets after your data has been collected. If you want your spreadsheet to follow a specific naming convention, you should specify the name for each of your questions before deploying the form as a data collection project.

4.1.5 Guidance Hint (optional)

Guidance Hints are extra instructions that you can add to your questions as notes. By default, in Enketo web forms, the guidance hints are displayed under an accordion which can be expanded and collapsed as shown below.

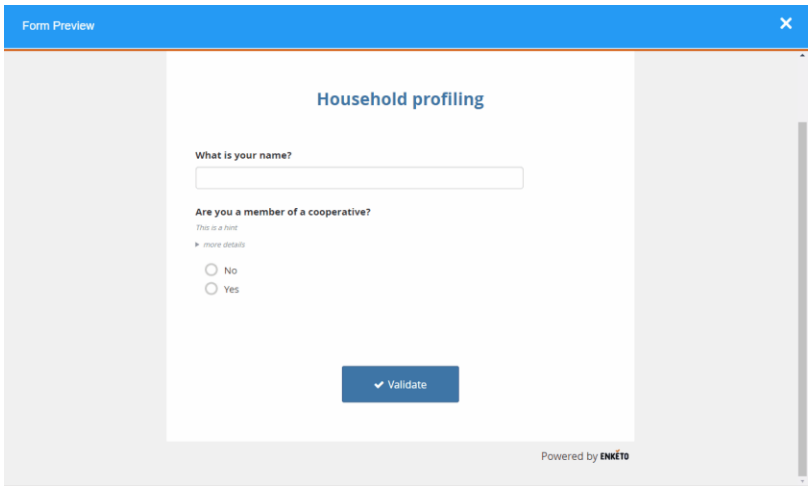


Figure 11: Guidance hint as displayed in Enketo web forms.

4.1.6 Mandatory Response

This setting allows you to specify whether the question must be responded to at all times or not. In XLSForm, this is called required.

In DIGIKYAR, there are three options for mandatory response:

1. **Yes** - The question must be answered at all times. If a response is not provided, the user will not be able to move to the next question or save the form.
2. **No** - The question is not mandatory and hence can be manually skipped.
3. **Custom logic** - You can define logic using XLSForm code which will define when the question will be mandatory. For example, if you set the following custom logic ${age} > 18, the question will be mandatory when a preceding question with the data column name age is greater than 18.

4.1.7 Default Response (optional)

This allows specifying a default response that the interviewer can accept or change. In most studies this would not be recommended as it might create an accidental bias, but it may be useful for date or time questions where the responses tend to be around a certain known point.

For **Date questions**, the default response needs to be written in the format YYYY-MM-DD e.g., 1974-12-31).

For **Select One** or **Select Many** questions the response needs to be written using the unique Value (xml value) - not the label (e.g., first grade rather than First grade).

4.1.8 Adding Skip Logic

Skip logic is also sometimes referred to as 'branching' or 'relevant conditions.' By default, all questions are always visible. Skip logic controls which question should be displayed only if a certain condition (or conditions) is fulfilled. Conditions are always applied to the question or group that should be sometimes hidden, sometimes visible. (This is important, as many paper surveys approach the problem from the other direction, writing things like "If yes, go to question 35".)

Conditions can be added to each question by clicking on Settings inside the question card, then Skip Logic. There are two ways to add a skip logic condition;

1. Using the skip logic wizard to help you build your conditions
2. Manually enter your skip logic in XLSForm code

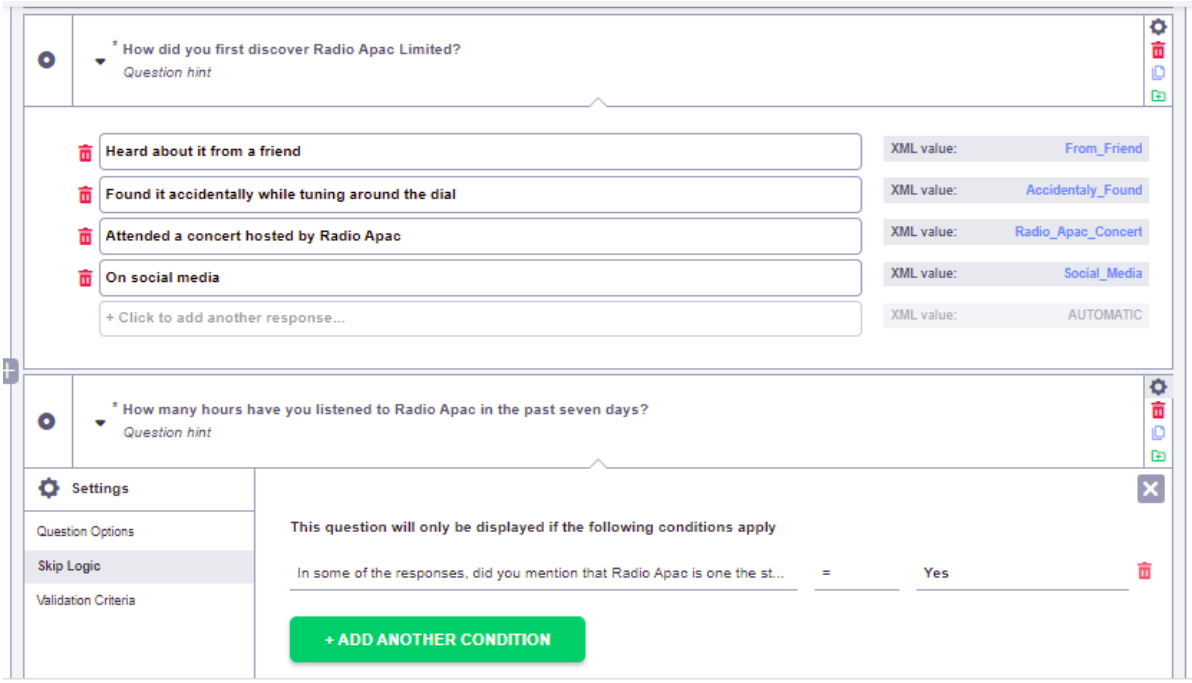


Figure 12: Adding skip logic to your questions.

4.1.9 Skip logic example – Formbuilder

Q1: Are you currently in school?

Q2: What's your school level? Q3: Which grade are you in?

You would want to display the third question only if the respondent answers 'Yes' to the first question. The correct skip logic condition should display:

Q1: Are you currently in school? = Yes

You can delete skip logic conditions by clicking on the trash can symbol.

To add multiple conditions, add your first one, then click on the 'Add a condition' button. When using two or more conditions, be sure to choose between the two options whether the question should match any (at least one) of these criteria, or all of them.

4.1.10 Manually enter your skip logic in XLSForm code

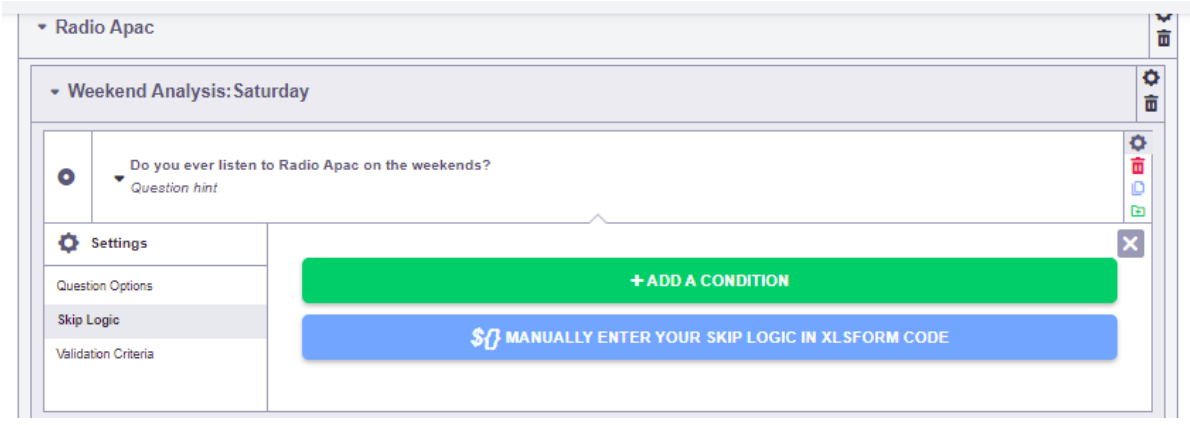


Figure 13:2.7.9 Manually enter your skip logic in XLSForm code

4.1.11 Skip logic example -XLSForm code

Q1. Do you rear animals?

Q2. Which of these animals do you own?

Q3. Kindly type the main breed of dog you own?

You would want to display Q2 if Q1 is Yes, so you would need to add the following condition on the relevant column

${Q1} = '1' or ${Q1} = 'Yes'

Note: This depends on whether the response item is labelled as Yes for Yes or 1 for Yes as in our example.

Q3 will only be displayed if Q2 has Dogs as one of the response items. Since Q2 is a multiple response then you would add this condition in the relevant columns for Q3.

selected(${Q2}, 'Dogs')

5 CREATING DIGIKYAR QUESTIONNAIRES WITH XLS

When building forms for DIGIKYAR, there are two main methods you can use. One is by using the DIGIKYAR formbuilder, and the other is using XLSForm. XLSForm is a standard for creating forms in Microsoft Excel and makes it possible to create basic and advanced forms in a user-friendly format.

NB: There are a number of XLSForm features that are not yet supported by the formbuilder and therefore a knowledge of XLSForm can be important for more complex forms.

5.1 What is XLSForm

An XLSForm is simply an Excel file structured in a standardized format that can be uploaded to DIGIKYAR to generate a data collection form.

Not only does using XLSForm offer enhanced functionalities, but it can also make collaborating on form building easier in some case. For example, members of your team can work on building the form in Google Sheets, allowing for real time collaboration.

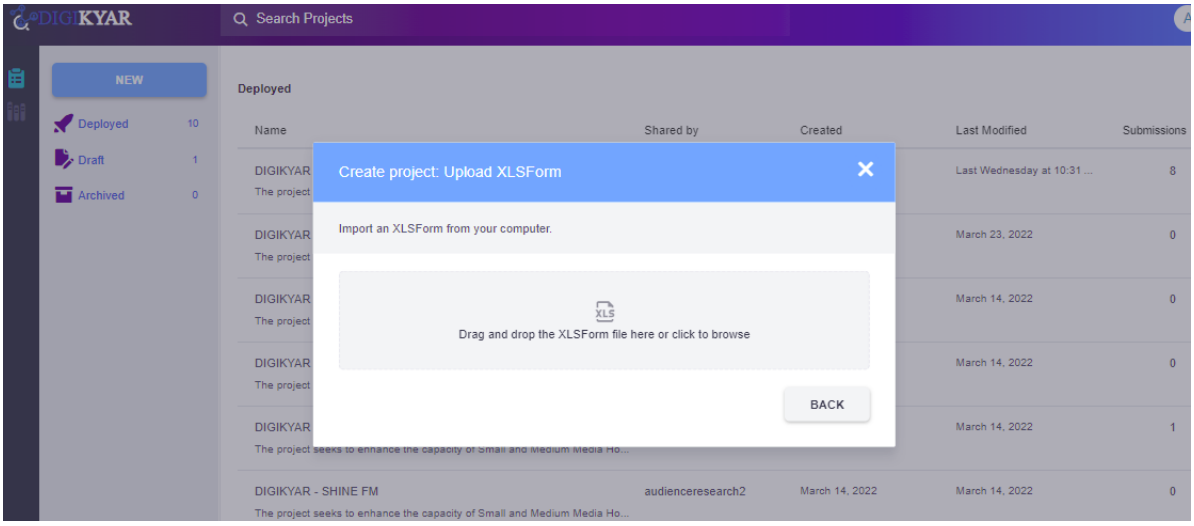


Figure 14: Creating DIGIKYAR Questionnaires using XLS.

5.2 How to set up XLSForm

To get started with XLSForm, do the following:

1. Create a workbook (either in Microsoft Excel or Google Sheets).
2. Create three worksheets: survey, choices, and settings respectively (sheet names must all be in lowercase).
3. In the survey worksheet, create three columns with headings: type, name, and label.
4. In the choice's worksheet, create three columns with headings: list_name, name, label
5. In the settings worksheet, create a column named form_title (the settings sheet is optional)

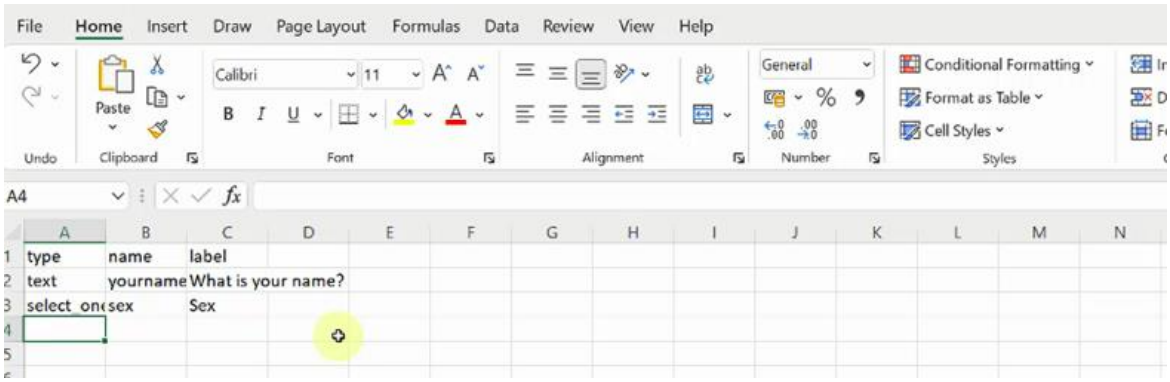


Figure 15:Creating DIGIKYAR questionnaires using Excell XLS

5.3 Adding XLS questions

Questions in XLSForm go in the survey sheet. We will practice adding a few questions: What is your name, Sex and How old are you?

1. In the survey sheet, under the type column, type "text". This is the question type for our first question.
2. Under the name column, type "yourname". This will be the variable name of our first question. The variable name uniquely identifies each question in the form. It's also the name of the question in the database when we start collecting data. Each question must have a unique name and cannot contain spaces or symbols (except the underscore).
3. Under the label column, type "What is your name". This label will be shown as the question text on the form when we collect data.

For our second question:

1. Still in the survey sheet, type select_one gender in the type column below the previous question (make sure to put a space between the 2 words). select_one is the question type that allows us to specify a list of choices where a user will only be allowed to pick one choice. (If a user can select several choices, this would be specified by the select_multiple question type.) "sex" is the name of the list of choices which we have to define in the choices sheet.
2. Type the name and label of the question as "sex" and "Sex" respectively.

Add the final question as follows:

type	name	label
integer	age	How old are you?

5.4 Adding choices

Regardless of the type of multiple-choice question (select_one or select_multiple), the next step will be to define the list of **choices** in the choices sheet. Each list of choices must have the same list_name.

Since we defined one question that has a list of choices (" sex") in the previous step, we need to add this list in the **choices** sheet as follows:

1. Switch to the choices sheet so that you can add your list of choices for the "Sex" question.
2. In the cell below list_name, type "sex". This is the list name we defined for the "Sex" question in the survey sheet. In the cell below name, type "male". This is the value that will be stored when the user chooses the option "Male". Under label, type "Male". This is what will be shown for this option in the survey.
3. For the second choice, type "sex" as list_name, "female" as the name, and "Female" as the label.

5.5 Adding settings

It is not mandatory to include the settings sheet in the XLSForm - any form will work just fine without it. However, at minimum, you can define the form_title.

NB: Without the form_title in the settings sheet, Digikyar, will, by default, use the file name as the name of the project when you import the XLSForm.
Below the form_title column, type "DIGIKYAR Practice Questionnaire" as the title of the form we are creating.

5.6 Uploading and previewing the XLSForm in DIGIKYAR

Once you have finished creating the XLSForm you must upload it to DIGIKYAR in order to preview it.

1. While you are on the Project List view (the first screen after you have just logged in to DIGIKYAR), click NEW.
2. Click Upload an XLSForm (if you were creating the form using Google Sheets, you can first download the file as an Excel file and then upload it.
3. Choose the Excel file you just created and wait until it uploads.
4. Confirm the project details and click CREATE PROJECT.
5. You can then click the Preview form button to preview your form.

5.7 Downloading an XLSForm from DIGIKYAR

DIGIKYAR allows you to download a form you have been creating using the formbuilder as an XLSForm file. This might be useful for several reasons such as:

- You need to add some advanced features that are not yet supported in the formbuilder.
- You would like to make changes to the form that might be easier through XLSForm (such as quickly duplicating a large number of questions).
- Your computer resources such as your browser or internet connection.
- You would like to collaborate on the form with members of your team and you prefer sharing the XLSForm.
- Manage versioning of your form outside of the DIGIKYAR platform.

- You are needing assistance from the support team and need to share your form with us.

5.8 Download the XLSForm while you are on the Project List view:

1. Hover your mouse over the name of the project.
2. Click the More actions icon.
3. Click Download XLS.
4. Save the file.

5.9 Replacing a form with an XLSForm file

You can replace an existing form with a new version using an XLSForm (for example, after having edited the form in Excel). To do this while you are on the Project List view:

1. Hover your mouse over the name of the project.
2. Click the More actions icon.
3. Click Replace form.
4. Choose the file.

5.10 XLSForm Basic Format

Each Excel workbook usually has two worksheets: **survey** and **choices**. A third optional worksheet called **settings** can add additional specifications to your form and is described below.

5.11 The survey worksheet

This worksheet gives your form its overall structure and contains most of the content of the form. It contains the full list of questions and information about how they should appear in the form. Each row usually represents one question; however, there are certain other features described below that you can add to the form to improve the user experience.

5.12 The choices worksheet

This worksheet is used to specify the answer choices for multiple choice questions. Each row represents an answer choice. Answer choices with the same **list name** are considered part of a related set of choices and will appear together for a question. This also allows a set of choices to be reused for multiple questions (for example, yes/no questions).

Both of these worksheets have a set of mandatory columns that must be present for the form to work. Additionally, each worksheet has a set of optional columns that allow further control over the behavior of each entry in the form, but are not essential to have. Every entry must have values for each of the mandatory columns, but the optional columns may be left blank.

The **survey** worksheet has 3 mandatory columns: **type**, **name**, and **label**.

- The **type** column specifies the type of entry you are adding.

- The **name** column specifies the unique variable name for that entry. No two entries can have the same name. Names have to start with a letter or an underscore. Names can only contain letters, digits, hyphens, underscores, and periods. Names are case-sensitive.
- The **label** column contains the actual text you see in the form. Alternatively, label translation columns can be used.

type	name	label
today	today	
select_one gender	gender	Respondent's gender?
integer	age	Respondent's age?

Figure 16: Survey choices settings

The columns you add to your Excel workbook, whether they are mandatory or optional, may appear in any order. Optional columns may be left out completely. Any number of rows may be left blank. All .xls file formatting is ignored, so you can use dividing lines, shading, and other font formatting to make the form more readable.

One thing to keep in mind when authoring forms in Excel is that the syntax you use must be precise. For example, if you write **Choices** or **choice** instead of **choices**, the form won't work.

5.13 Question types

XLSForm supports a number of question types. These are just some of the options you can enter in the **type** column in the **survey** worksheet in your XLSForm:

Question type	Answer input
integer	Integer (i.e., whole number) input.
decimal	Decimal input.
range	Range input (including rating)
text	Free text response.
select_one [options]	Multiple choice question; only one answer can be selected.
select_multiple [options]	Multiple choice question; multiple answers can be selected.
select_one_from_file [file]	Multiple choice from file; only one answer can be selected.
select_multiple_from_file [file]	Multiple choice from file; multiple answers can be selected.
rank [options]	Rank question; order a list.
note	Display a note on the screen, takes no input. Shorthand for type=text with readonly=true.
geopoint	Collect a single GPS coordinate.

Question type	Answer input
geotrace	Record a line of two or more GPS coordinates.
geoshape	Record a polygon of multiple GPS coordinates; the last point is the same as the first point.
date	Date input.
time	Time input.
dateTime	Accepts a date and a time input.
image	Take a picture or upload an image file.
audio	Take an audio recording or upload an audio file.
background-audio	Audio is recorded in the background while filling the form.
video	Take a video recording or upload a video file.
file	Generic file input (txt, pdf, xls, xlsx, doc, docx, rtf, zip)
barcode	Scan a barcode, requires the barcode scanner app to be installed.
calculate	Perform a calculation; see the Calculation section below.
acknowledge	Acknowledge prompt that sets value to "OK" if selected.
hidden	A field with no associated UI element which can be used to store a constant
xml-external	Adds a reference to an external XML data file

Figure 17: XLS Question types

5.14 GPS

For example, to collect the name and GPS coordinates of a store, you would write the following:

type	name	label
text	store_name	What is the name of this store?
geopoint	store_gps	Collect the GPS coordinates of this store.

To collect a line or shape of GPS coordinates, you can use one of the following:

type	name	label	hint
geotrace	pipe	Pipeline	Please walk along the pipeline and record the coordinates of each corner point
geoshape	border	Border	Please walk along the border and record the coordinates of each corner point

5.15 GPS with accuracy Threshold

When recording GPS coordinates in ODK Collect, it automatically collects the gps when an accuracy level of 5 meters or less is reached. You can change this default behavior by specifying an accuracy Threshold;

this could be less than 5m or more than 5m. You will need to add a column with heading **body::accuracy Threshold** on the survey sheet of your XLSForm. Then specify your preferred accuracy threshold value for this column on your Geopoint question, as in the example shown below:

type	name	label	body::accuracyThreshold
geopoint	store_gps	Collect the GPS coordinates of this store.	1.5

5.16 Multiple choice

XLSForm supports both **select_one** (select only one answer) and **select_multiple** (select multiple answers) questions. Writing a multiple choice question requires adding a **choices** worksheet to your Excel workbook. Here is an example of a **select_one** question:

type	name	label
select_one yes_no	likes_pizza	Do you like pizza?

list name	name	label
yes_no	yes	Yes
yes_no	no	No

Note that the **yes_no** in the **survey** worksheet must match the **yes_no** in the **list name** column in the **choice's** worksheet. This ensures that the form displays the correct list of answer choices for a particular question.

We can also add multiple choice questions that allow multiple answers to be selected, like so:

type	name	label
select_multiple pizza_toppings	favorite_toppings	What are your favorite pizza toppings?

list name	name	label
pizza_toppings	cheese	Cheese
pizza_toppings	pepperoni	Pepperoni
pizza_toppings	sausage	Sausage

survey**choices**settings

5.17 Choice names

The `name` column of the choices sheet defines the values that will be saved when each choice is selected during data collection. Choice names for **select_multiple** must not contain spaces because spaces are used as a separator when an answer with multiple selected choices is saved. Choice names for **select_one** questions may contain spaces. However, we recommend avoiding them to make analysis

easier. Additionally, this makes it possible to convert the question to a **select_multiple** in a future form version.

In general, choice names should be unique within a single choice list. If two choices from the same list have the same name, they will be impossible to tell apart in analysis. If you have duplicate choice names, you will get an error and your form will not be converted. However, it may sometimes be appropriate to have duplicate choice names. An example would be if you use a cascading select and the choices with the same name are differentiated by a preceding question. If you do need to use duplicate choice names, you can suppress the error by using the `allow_choice_duplicates` setting:

allow_choice_duplicates
yes

5.18 Specify other

For multiple choice questions, surveys often include an option of marking **other** when their answer choice is not listed. Then they are usually asked to specify the other option. This is possible through XLSForm by including **or_other** after the answer choice list name in the survey worksheet. The choices worksheet stays the same. See below:

type	name	label
select_multiple pizza_toppings or_other	favorite_topping	What are your favorite pizza toppings?

list name	name	label
list name	name	label
pizza_toppings	cheese	Cheese
pizza_toppings	pepperoni	Pepperoni
pizza_toppings	sausage	Sausage

5.19 Caveat

When you export data using this **or_other** option, in the **favorite_topping** column, you will see a value **other**. A separate column will have the answer for the questions in which the user selected **other**. This makes data analysis more cumbersome, so we do not recommend the **or_other** construct for large scale data collection efforts. See the **Relevant** section below for an alternative method more appropriate for large scale projects.

5.20 Multiple choice from file

The options in a multiple-choice question can also be taken from a separate file instead of the choices sheet. This is particularly useful if the options are dynamic or if the list of options is used in multiple surveys. Two types of files are supported: CSV and XML files. See usage examples below:

type	name	label	choice_filter
select_multiple_from_file country.csv	liv	In which countries did you live?	
select_one_from_file countries.xml	cou	In which country do you live now?	
select_one_from_file countries.xml	cit	What is the closest city?	name=${cou}
select_one_from_file households.csv	hh	Select household number	

The files require a specific format. A CSV file requires a `name` and `label` column which represent the value and label of the options. An XML file requires a structure as shown below:

```
<root>
  <item>
    <name/>
    <label/>
    ...
  </item>
</root>
```

Both CSV and XML files may have additional columns and XML nodes as long as the above-mentioned basic requirements are met.

If the CSV or XML files use different names for the choice `name` and `label`, add a column to the survey sheet named `parameters`, and specify the custom names with the `value` and `label` parameters. See usage examples below:

type	name	label	parameters
select_multiple_from_file country.csv	liv	In which countries did you live?	value=ccode
select_one_from_file countries.xml	cou	In which country do you live now?	label=cname
select_one_from_file households.csv	hh	Select household number	value=housenum, label=housename

Note that this question type is generally the preferred way of building select questions from external data as it is the most versatile and works across applications. However, if your external data file consists of many thousands of lines, please test carefully whether the performance is satisfactory on the lowest spec device you intend to use. If it is too slow, consider using External Selects or Dynamic selects from preloaded data if your data collection application supports it.

5.21 Rank

The rank widget can be used to let respondents order a list of options. The answer is saved as an ordered, space-separated list of option values where all options are always included. The syntax is very similar to multiple-choice questions.

type	name	label
rank pizza_toppings	toppings	Order pizza toppings with your favorite on top

list name	name	label
pizza_toppings	cheese	Cheese
pizza_toppings	pepperoni	Pepperoni
pizza_toppings	sausage	Sausage

To prevent bias, it is often recommended to use the randomize feature in conjunction with this widget.

5.22 Range

To restrict integer or decimal inputs to a specific range, you can use the **range** question. This question can be used with 3 optional space-separated parameters: **start**, **end**, and **step** in a **parameter's** column. The default values are 0, 10, and 1 respectively. The example below will create a question that allows input from 0 until 17 with a step of 1. Using a decimal step will result in decimal values being collected.

type	name	label	parameters
range	amount	What is the age of the child?	start=0 end=17 step=1

To display a range question as a **rating widget** using stars, you can add the rating appearance as shown below:

type	name	label	appearance	parameters
range	rated	What rating do you give?	rating	start=1 end=5 step=1

5.23 Image

To upload an image file the **image** question type can be used. To ensure the images are not too large, you can optionally set the **max-pixels** parameter which will automatically downsize the uploaded image to match the largest side of the image with the pixel value provided.

type	name	label	parameters
image	img	Upload an image	max-pixels=1000

5.24 Audio recording quality

Certain clients use a value for **quality** in the **parameters** column to configure audio recording quality for question types **audio** or **background-audio**. Both question types accept **quality** values `voice-only`, `low` and `normal`. **audio** additionally accepts a **quality** of `external` to specify that an external application should be used for recording.

type	name	parameters
audio	animal_sound	quality=normal

5.25 Metadata

XLSForm has a number of data type options available for meta data collection:

Metadata type	Meaning
start	Start date and time of the survey.
end	End date and time of the survey.
today	Day of the survey.
deviceid	Unique client identifier. Can be user-reset.
phonenumber	Phone number (if available).
username	Username configured (if available).
email	Email address configured (if available).
audit	Log enumerator behavior during data entry

Note that some metadata fields only apply for mobile phone-based forms. For example, if you wanted to collect all of these types of metadata, put the following in your form (typically at the beginning, but can be at any point of your form):

type	name	label	parameters
start	start		
end	end		
today	today		
deviceid	deviceid		
phonenumber	phonenumber		
username	username		
email	email		
audit	audit		[optional, see below]

type	name	label	parameters

Notice that there are no labels associated with the metadata question types. This is because the phone captures these variables automatically. These questions will not appear on the screen of the phone, but you will see them when viewing your submitted survey data.

5.26 Audit enumerator behavior and location tracking

The **audit** meta question will enable ODK Collect to log how people navigate through a form during data entry. For example, this can be used to measure how much time an enumerator took to fill in a question, responses that were edited later on, or when the form was saved.

Optionally, the **audit** metaquestion can be configured to also record the location of the enumerator throughout the interview. This may be useful for quality control or to record exact paths taken between each respondent. To do this, add a column called **parameters** to your form and enter three required parameters: **location-priority**, **location-min-interval**, and **location-max-age**.

This example below would collect the precise GPS location every 180 seconds and will discard coordinates collected more than 300 seconds ago.

Note: For now, this feature is only available in Collect, but not in Enketo webforms.

type	name	label	parameters
audit	audit		location-priority=high-accuracy location-min-interval=180 location-max-age=300

5.27 External XML data

For advanced users, who need to perform complex queries on external data without restrictions, an external XML data file can be added with question type **xml-external**. The value in the **name** column can be used to refer to this data in any formula (e.g., for a calculation, constraint, relevant, or choice_filter) using the **instance('name')** function. A file with the same name and the **.xml** extension should be uploaded with the form. See below for an example that requires uploading a file called houses.xml with the form.

If your external data file consists of many thousands of lines, please test carefully whether the performance is satisfactory on the lowest spec device you intend to use. If it is too slow, consider using External Selects instead if your data collection application supports this.

type	name	label	calculation
xml-external	houses		
integer	rooms	How many rooms?	
calculate	count		count(instance('houses')/house[rooms = current()/../rooms])

type	name	label	calculation

5.28 Question Hints

5.28.1 Regular hints

Sometimes you want to add a small hint to a question on your form, instructing the user how to answer the question, but you don't want the hint to be part of the question itself. It's easy to add hints to questions in XLSForms. Simply add a **hint** column and add your hint message. See below for an example.

type	name	label	hint
text	name	What is the name of this store?	Look on the signboard if the store has a signboard.
geopoint	geopoint	Collect the GPS coordinates of this store.	

5.28.2 Guidance hints

There is a special kind of hint that is normally not shown in the form. It is only shown in special views. An example would to show these hints on print-outs or during a training for enumerators. These hints are called **guidance hints** and can be added in the **guidance_hint** column. See below for an example.

type	name	label	guidance_hint	relevant
integer	age	Age?		
text	name	Name?	This will only be shown for age > 18.	${age} > 18

5.29 Formulas

Formulas are used in the constraint, relevant, calculation, and trigger columns and optionally also in the default, and required columns. Formulas allow you to add additional functionality and data quality measures to your forms.

Formulas are composed of functions and operators (+,*,div,etc.). A well-documented full list of operators and functions can be found in the ODK documentation. For the technically inclined, the underlying XForms specification is the actual source document for the supported functions.

5.30 Constraints

One way to ensure data quality is to add constraints to the data fields in your form. For example, when asking for a person's age, you want to avoid impossible answers, like -22 or 200. Adding data constraints

in your form is easy to do. You simply add a new column, called **constraint**, and type in the formula specifying the limits on the answer. In the example below, the answer for the person's age must be less than or equal to 150. Note how the. in the formula refers back to the question variable.

type	name	label	constraint
integer	age	How old are you?	. <= 150

In this example, the formula. `<= 150` is saying that the value entered. for the question must be less than or equal to 150. If the user puts 151 or above as the answer, s/he will not be allowed to move on to the next question or submit the form.

5.31 Constraint message

If you want to include a message with your constraint, telling the user why the answer is not accepted, you can add a constraint_message column to your form. See the example below.

type	name	label	constraint	constraint_message
integer	respondent_age	Respondent's age	. >=18	Respondent must be 18 or older to complete the survey.

In this example, if the user enters an age less than 18, then the error message in the constraint_message column appears.

5.32 Relevant

One great feature of XLSForm is the ability to skip a question or make an additional question appear based on the response to a previous question. Below is an example of how to do this by adding a **relevant** column for a **select_one** question, using our pizza topping example from before:

type	name	label	relevant
select_one yes_no	likes_pizza	Do you like pizza?	
select_multiple pizza_toppings or_other	favorite_topping	Favorite toppings	${likes_pizza} = 'yes'

In this example, the respondent is asked, “Do you like pizza?” If the answer is **yes**, then the pizza topping question appears below. Note the `${ }` around the variable **likes_pizza.** These are required in order for the form to reference the variable from the previous question.

In the next example, below, we use relevant syntax for a **select_multiple** question, which is slightly different from the **select_one** question example above.

type	name	label	relevant
select_one yes_no	likes_pizza	Do you like pizza?	
select_multiple pizza_toppings or_other	favorite_topping	Favorite toppings	${likes_pizza} = 'yes'
text	favorite_cheese	What is your favorite type of cheese?	selected(${favorite_topping}, 'cheese')

list name	name	label
pizza_toppings	cheese	Cheese
pizza_toppings	pepperoni	Pepperoni
pizza_toppings	sausage	Sausage

Since the pizza topping question allows multiple responses, we have to use the `selected(${favorite_topping}, 'cheese')` expression, because we want the cheese question to appear every time the user selects **cheese** as one of the answers (regardless of whether additional answers are selected).

Earlier we mentioned there was an alternative method for specifying other for multiple choice questions which is more appropriate for large scale surveys. This can be done using the same relevant syntax from the example above:

type	name	label	relevant
select_multiple pizza_toppings	favorite_toppings	What are your favorite pizza toppings?	
text	favorite_toppings_other	Specify other:	selected(${favorite_toppings}, 'other')

list name	name	label
pizza_toppings	cheese	Cheese
pizza_toppings	pepperoni	Pepperoni
pizza_toppings	sausage	Sausage
pizza_toppings	other	Other

Note that you must include **other** as an answer choice in the **choices** worksheet.

5.33 Calculation

Your survey can perform calculations using the values of preceding questions. In most cases using a **calculate** type question is appropriate. For example, in the survey below, we have calculated the tip for a meal and displayed it to the user:

type	name	label	calculation
decimal	amount	What was the price of the meal?	
calculate	tip		${amount} * 0.18
note	display	18% tip for your meal is: ${tip}	

Note that the **${tip}** in the last line will be replaced with the actual tip amount when viewing and filling out the form.

The calculate type calculates **text** but calculations can also be added to any other question types. Non-text types can be useful for data analysis, e.g if a date or date-time is calculated. **If no label and no hint is included, the calculation will be hidden.** See example below which is the equivalent of the previous form:

type	name	label	hint	calculation
decimal	amount	What was the price of the meal?		
text	tip			${amount} * 0.18
note	display	18% tip for your meal is: ${tip}		

And this is an example when a non-text type is needed because of data analysis requirements:

type	name	label	hint	calculation
date	day			today()

If a label or hint is included, the question will be visible on the form and the calculated value will be shown in the input field or widget. This is generally only recommended for **read-only** questions to avoid re-calculating (erasing) a user-entered value. See example below:

Note: Using non-text calculation types has no effect on using the calculation result within the form itself.

type	name	label	readonly	calculation
decimal	amount	What was the price of the meal?		
note	display	18% tip for your meal is:		${amount} * 0.18
date	today	Today's date is:	true	today()

Note the difference with the first form in this section is how the calculated tip value is displayed. In the first example it was shown in the label and in the last example it is shown inside a readonly input field.

5.34 Trigger

A trigger column can be used to run a calculation only when another visible question in the form changes. This means that the question that is serving as the trigger has to have a label or a hint (otherwise it will be hidden). See a simple but very useful example below:

type	name	label	calculation	trigger
integer	temp	Enter the current temperature		
dateTime	temp_ts		now()	${temp}

This will calculate a timestamp immediately after a respondent enters a temperature. If the user goes back and changes the temperate, the timestamp will be re-calculated.
All the regular calculation features apply to these special value-change-triggered calculations as well. So, you can e.g. use a label or hint to display the calculation question on the form to the user.
Multiple questions may have the same trigger. See this example, where two calculations are triggered by the temperature question (one is hidden, and the other is shown):

type	name	label	calculation	trigger	readonly
integer	temp	Enter temperature in Celsius			
dateTime	temp_ts		now()	${temp}	
text	temp_F	Temperature in Fahrenheit	32 + 1.8 * ${temp}	${temp}	true
calculate	temp_K		273.15 + ${temp}	${temp}	

In the form above the temp_F question is shown to the user and the temp_K question is hidden, just as they would be if trigger was not used.
An important and powerful difference with regular calculations is that **the calculation value with a trigger may also be empty**, which serves to clear a value from the form. See example below:

type	name	label	calculation	trigger
text	name	What is the name of the oldest person here?		
integer	age	How old is this person?		${name}

If the respondent using this form has entered the name and age of person A and subsequently finds out there is an older person B, the age field will be cleared as soon as the name of person B has been entered.

5.35 Required

It's simple to mark certain questions as required in your form. Marking them as required means the user will not be able to move on to the next question or submit the form without entering an answer for that question.
To make questions required, add a **required** column to your survey worksheet. Under that column, mark questions as required by writing **yes.** See the example below:

type	name	label	constraint	required
integer	age	How old are you?	. <= 150	yes

5.36 Required message

If you want to customize the message displayed to users when they leave a required question blank, you can add a **required_message** column to your form. See the example below.

type	name	label	required	required_message
integer	respondent_age	Respondent's age	yes	Sorry, this answer is required.

5.37 Randomize Choices

For any question type that shows a **list of choices** the shown order of the choices displayed to the user can be randomized with the **parameters** column. See below:

type	parameters	name	label
select_one toppings	randomize=true	top	Favorite?

For reproducible randomization, a **seed** can be explicitly provided as shown below.

type	parameters	name	label	calculation
calculate		sd		once(decimal-date-time(now()))
select_one toppings	randomize=true, seed=${sd}	top	Favorite?	

Note that `once()` is used to prevent re-randomizing for example when a draft record is loaded for editing.

5.38 Grouping questions

To create a group of questions in your form try the following:

type	name	label
begin group	respondent	Respondent
text	name	Enter the respondent’s name
text	position	Enter the respondent’s position within the school.
end group		

This is a good way to group related questions for data export and analysis. Notice how **end group** doesn't require a name or label, because it is hidden in the form.

5.39 Nesting groups within groups

Groups of questions can be nested within one another:

type	name	label
begin group	hospital	Hospital
text	name	What is the name of this hospital?
begin group	hiv_medication	HIV Medication
select_one yes_no	have_hiv_medication	Does this hospital have HIV medication?
end group		
end group		

You always have to end the most recent group that was created first. For instance, the first **end group** you see closes the HIV medication group, and the second one closes the beginning hospital group. When working with groups and you keep getting error messages when trying to upload your form, double-check that for each **begin group** you have one **end group.**

5.40 Skipping

One neat feature of XLSForm is the ability to skip a group of questions by combining the group feature with relevant syntax. If you want to skip a group of questions all at once, put the relevant attribute at the beginning of a group like follows:

type	name	label	relevant
integer	age	How old are you?	
begin group	child	Child	${age} <= 5
integer	muac	Record this child's mid-upper arm circumference.	
select_one yes_no	mrdt	Is the child's rapid diagnostic test positive?	
end group			

In this example, the two child group questions (**muac** and **mrdt**) will only appear if the child's **age** from the first question is less than or equal to five.

5.41 Repeats

A user can repeat questions by using the **begin repeat** and **end repeat** construct:

type	name	label
begin repeat	child_repeat	
text	name	Child's name
decimal	birthweight	Child's birthweight
select_one male_female	sex	Child's sex
end repeat		

list name	name	label
male_female	male	Male
male_female	female	Female

In this example, the **name, birthweight,** and **sex** fields are grouped together in a repeat, and the user can collect the same information about multiple children by selecting the option in the form to add another repeat.

The **label** column is optional for **begin repeat**. Assigning a label to a repeat will add the label as a title to the block of repeat questions in the form.

When a repeat is shown in a table of contents, the label used to represent each repeat is the label of the first group inside that repeat. In the example below, if a repeat is filled out with values `Preity` for `first_name`, `Zinta` for `last_name` and `71` for `age`, that repeat will be summarized as "Preity Zinta - 71":

type	name	label
begin repeat	person_repeat	
begin group	person	${first_name} ${last_name} - ${age}
text	first_name	First name
text	last_name	Last name
integer	age	Age
end group		
end repeat		

5.42 Fixed repeat counts

Instead of allowing an infinite number of repeats, the form designer can specify an exact number of repeats by using the **repeat_count** column:

type	name	label	repeat_count
begin repeat	child_repeat		3
text	name	Child's name	
decimal	birthweight	Child's birthweight	
end repeat			

In the above example, exactly **3** child repeats will be created.

5.43 Dynamic repeat counts

The repeat count can be set to an expression that refers to other fields in the form. In the example below, the number that the user inputs for the **num_hh_members** field dictates the number of **hh_member** repeats added:

type	name	label	repeat_count
integer	num_hh_members	Number of household members?	
begin repeat	hh_member		${num_hh_members}
text	name	Name	
integer	age	Age	
end repeat			

5.44 Only add repeats in certain conditions

Like with groups, all of the questions in a repeat can be skipped based on some condition. In the example below, the person filling out the form will only be given the opportunity to add children if they first indicate that there are children to add:

type	name	label	relevant
select_one yes_no	has_child	Do any children live here?	
begin repeat	child_repeat		${has_child} = 'yes'
text	name	Child's name	
decimal	birthweight	Child's birthweight	
end repeat			

list_name	name	label
yes_no	yes	Yes
yes_no	no	No

5.45 Representing zero repeats

By default, the person filling the form will see the questions corresponding to one repeat before getting the option to add more. To represent 0 repeats, there are three options:

- Teach the people filling out the form to delete the first repeat added
- If the exact number of repeats is known ahead of time, **use a dynamic repeat count**
- If the exact number of repeats is not known ahead of time, **use relevant** to only prompt the user for repeats if there are some to add

5.46 Multiple language support

It's easy to add multiple languages to a form. You simply have to name your **label::language1 (code), label::language2 (code),** etc., and your forms will be available in multiple languages. See the example below. Select a different form language from the pulldown menu of data collection application (this may be located under the **Menu** key). For the form below, English and Español will show up as the possible options.

type	name	label::English (en)	label::Español (es)	constraint
integer	age	How old are you?	¿Cuántos años tienes?	. <= 150

You can also add different language columns for hints and media files by using the same `::language (code)` construct, as shown in the example below. See also the XLSForm reference table, which includes a list of all column headers that can accept a language modification.

hint::English (en)	hint::Dutch (nl)	image::English (en)	image::Dutch (nl)
a hint	een hint	old_person_cartoon.png	ouwe_strip.png

Form language and user interface language may be the determined separately by the application and may not match. To facilitate matching both (in the future), it is recommended, though optional, to add a 2-character language code after the language name. The official 2-character language codes, called ***subtags*.**

5.47 Media

You can include questions in your form that display images or that play video or audio files. If using the ODK mobile client for form submission, you need to put the media files that you want to include in the **/odk/forms/formname-media** folder on your phone, and then reference the exact file name in the **media** column in your form. See below for an example of how to do this.

type	name	label	image	video
note	media_example	Media example	example.jpg	example.mp4

5.48 Pre-loading CSV data

Pre-loading data is done when one wants to reference pre-existing data in a survey form. You can be able to reference data in your survey form (the survey you are now authoring), from a pre- existing data in a specific survey form or any other source. For example, if you have pre-existing data from a household survey and you want to collect follow-up data about the household occupants. You can be able to reference the household survey data in your survey form. To reference pre-existing data in a survey form:
Upload one or more .csv files as support files when you upload your form definition (the same way you upload media support files as explained in the Media section above). The first row of each .csv file should be a header that includes short:

- Unique names for each column

- Subsequent rows which should contain the data itself

Each csv file should contain at least one column that can be used to uniquely identify each row. Such columns will be used, at survey time, to look up which row's data to pull into the survey. For the columns that will be used for looking up rows add **_key** to the end of the column name in the first row. Any columns with names ending in **_key** will be indexed for faster look-ups on your survey devices. See below an example of the columns on a .csv file:

name_key	name
mango	Mango
orange	Orange

5.49 How to pull data from CSV

You can be able to pull data from .csv file by including one or more .csv files in your form during the survey time. For each data field that you want to pull into your survey:

- Add a **calculate field** to your survey.
- Give that field **a name**
- Then in its **calculation** column, call the **pulldata()** function, indicating which field to pull from which row of which .csv file.

See below for an example:

type	name	label	calculation
calculate	fruit		pulldata('fruits', 'name', 'name_key', 'mango')
note	note_fruit	The fruit ${fruit} pulled from csv.	

Once you have loaded .csv data into a survey field using the **pulldata()** function, you can reference that field in later relevance conditions, constraints, and labels, just as you would reference any other field that was filled in by the user.

5.50 Important notes on usage of pre-loaded data

Compress a large .csv file into a **.zip archive** before uploading it.

Save .csv file in **UTF-8 format** if pre-loaded data contains non-English fonts or special characters this enables your Android device to render the text correctly.

Data fields pulled from a .csv file are considered to be text strings therefore use the **int() or number()** functions to convert a pre-loaded field into numeric form.

If the .csv file contains sensitive data that you may not want to upload to the server, upload a blank .csv file as part of your form, then replace it with the real .csv file by hand-copying the file onto each of your devices.

5.51 Dynamic selects from pre-loaded data.

Once your form has one or more pre-loaded .csv files, you can dynamically pull the choice lists for **select_one** and **select_multiple** fields from those .csv files. Multiple-choice fields with dynamic choice lists follow the same general syntax as regular, static select_one and select_multiple fields.
The following should be done:

- Specify **select_one listname** or **select_multiple listname** in the type column (where **listname** is the name of your choice list)
- Specify any special **appearance styles** in the appearance column
- Include one or more rows for your **listname** on the **choice's** worksheet.

Below is an example of the **survey worksheet**:

type	name	label	appearance
select_one fruits	fruits	Select a fruit	search('fruits')

There are three differences when the choice list should be pulled from one of your pre-loaded .csv files:
In the appearance column:

Include a **search () expression** that specifies which .csv rows to include in the choice list.
If the field should use a non-default appearance style. The non-default appearance style goes into the column first, followed by a space, then the **search () expression**. [e.g., **quick search ()]**

On the **choices worksheet** row should indicate which .csv columns to use for the label and selected value. As follows:
list_name column: specify the name of your choice list as you normally would.
name column: include the name of the .csv column to use for uniquely identifying selected choices.
label column: include the name of the .csv column to use for labeling the choices.

Note:
If you wish to include multiple columns in the labels, include a comma-separated list of all columns to include. The name column will be dynamically populated based on the column name you put there, and the label column will be dynamically populated based on the column name(s) you put there.
In your choice's worksheet row, you may also include a .csv column name in the image column. If you do, the image filename to use will be pulled from the specified .csv column.

Note:
If you refer to image files in this way, you must always upload those image files as media file attachments when you upload your form to the server.
See below an example of the choice's worksheet:

list name	name	label
fruits	name_key	name

5.52 Cascading selects

A lot of forms start out by asking the location of the respondent, with each location selection specifying what the subsequent location choices will be (e.g., state » district » village). Instead of adding a **select_one** field for each location option, you can use cascade select. In order to use cascade selects, you will need to create a **choice_filter** column in your survey worksheet and add the location attribute columns in your choice's worksheet.

5.53 External selects

If a form has selects with a large number of choices (e.g., hundreds or thousands), that form can slow down form loading and navigation if Multiple Choice from File is used. The best workaround to this issue is to use external selects in those data collection applications (such as ODK Collect) that support it.
Enabling external selects is straightforward.
Instead of **select_one** for the prompt type, use **select_one_external.**
Instead of the **choices** sheet, put external choices in the **external_choices** sheet.

When an XLSForm with external choices is converted to an XForm, two files will be produced, the **XForm** (e.g., form-filename.xml) with all the normal choices, and an **itemsets.csv** with the external choices.
The **itemsets.csv** file can be uploaded to any ODK-compatible server (e.g., ODK Aggregate) as a media file. It will be downloaded to any ODK-compatible (e.g., ODK Collect) like any other media file and saved to the [form-filename]-media folder. Clients like ODK Collect load media files from the SD card and so your form with a large number of choices will now load very quickly.

5.54 Default

Adding a default field means that a question will be pre-populated with an answer when the user first sees the question. This can help save time if the answer is one that is commonly selected or it can serve to show the user what type of answer choice is expected. See the example below.

type	name	label	default
date	survey_date	Survey date?	2010-06-15
decimal	weight	Respondent's weight? (in kgs)	51.3

The respondent can simply change the answer by tapping in the answer field and entering another answer. You can also add a default calculation, which will only be calculated only once when the form loads or - if the question is inside a repeat - when the repeat is added.

type	name	label	default
date	d	Enter the date the event occurred?	today()

5.55 Read only

Adding a read only field means that a question cannot be edited. Read only fields can be combined with default fields to deliver information back to a user.

type	name	label	read_only	default
integer	num	Please patient is:	yes	5

5.56 Appearance

The **appearance** column allows you to change the appearance of questions in your form. The following table lists the possible appearance attributes and how the question appears in the form.

Appearance attribute	Question type	Description
multiline	text	Best if used with web clients, makes the text box multiple lines long.
minimal	select_one, select_multiple	Answer choices appear in a pull-down menu.
quick	select_one	Relevant for mobile clients only, this attribute auto-advances the form to the next question after an answer is selected.
no-calendar	date	For mobile devices only, used to suppress the calendar.
month-year	date	Select a month and year only for the date.
year	date	Select only a year for the date.
horizontal-compact	select_one, select_multiple	For web clients only, this displays the answer choices horizontally.
horizontal	select_one, select_multiple	For web clients only, this displays the answer choices horizontally, but in columns.
likert	select_one	Best if used with web clients, makes the answer choices appear as a Likert scale.
compact	select_one, select_multiple	Displays answer choices side by side with minimal padding and without radio buttons or checkboxes. Particularly useful with image choices.
quickcompact	select_one	Same as previous, but auto-advances to the next question (in mobile clients only).
field-list	groups	Entire group of questions appear on one screen (for mobile clients only).
label	select_one, select_multiple	Displays answer choice labels (and not inputs).

Appearance attribute	Question type	Description
list-nolabel	select_one, select_multiple	Used in conjunction with **label** attribute above, displays the answer inputs without the labels (make sure to put **label** and **list-nolabel** fields inside a group with **field-list** attribute if using mobile client).
table-list	groups	An easier way to achieve the same appearance as above, apply this attribute to the entire group of questions (might slow down the form a bit).
signature	image	Allows you to trace your signature into your form (mobile clients only).
draw	image	Allows you to sketch a drawing with your finger on the mobile device screen.

5.57 Settings worksheet

The **settings** worksheet is optional, but it is highly recommended to specify **form_title, form_id and version** at a minimum. Other settings allow you to further customize your form, including setting an overall style theme or encrypting your records.
An example **settings** worksheet is below:

form_title	form_id	version	instance_name	default_language	public_key
Example	ex_id	2017021501	concat(${firstname}, ' ', ${lastname})	English (en)	IIBIjANBg…

The settings column headings available are:

- **form_title:** The title of the form that is shown to users. The form title is pulled from **form_id** if **form_title** is blank or missing.
- **form_id**: The name used to uniquely identify the form on the server. The form id is pulled from the XLS file name if **form_id** is blank or missing.
- **version:** String of up to 10 numbers that describes this revision. Revised form definitions must have numerically greater versions than previous ones. A common convention is to use strings of the form 'yyyymmddrr'.
- **instance_name:** Expression using form fields to identify for each form submission.
- **default_language:** In localized forms, this sets which language should be used as the default. The same format as described for adding translations should be used, including the language code.
- **public_key:** For encryption-enabled forms, this is where the public key is copied and pasted.

- **submission_url:** This url can be used to override the default server where finalized records are submitted to.
- **style**: For web forms, specify the form style.
- **name:** XForms root node name. This is rarely needed,

5.58 Encrypted forms

Encryption-enabled forms provide a mechanism to keep *finalized* records private at all times. This includes the time *after a record is marked as final* that it is stored on the device and server as well as during transport, even when http is used for communication. Encrypted records including their uploaded files, such as photos, are completely inaccessible to anyone not possessing the private key.
To encrypt XLS forms, add the **public_key** column to the **settings** worksheet and paste the base64-encoded public RSA key as its value.

form_id	public_key
mysurvey	IIBIjANBgklawWEserewrwesgdreewrwe32serfserfewrwerewtwer23sgfrqjwerk3423432…

5.59 Specify alternative server

It is possible to specify an alternative server to send your submissions to in the **submission_url** column on the **settings** worksheet. Make sure to use the full URL that submissions should be sent to *including the path*.
If this column is left out or kept empty, submissions will go the default destination for the provider you are using for your surveys.

5.60 Specify form submission name

In the **settings** worksheet, you can specify a unique name for each form submission using fields filled in by the user during the survey. On the settings worksheet, add a column **called instance_name**. Write in the expression that defines the unique form instance name using fields from the survey worksheet.

5.61 Specify XForms root node name

In some rare cases, it may be helpful to explicitly specify a **root node name** for the generated XForm. For example, this may be necessary if updating a form that was converted with an older form converter that used a root node name other than `data`. In the **settings** worksheet, you can specify an identifier to use for the XForms root node name by adding a collumn called **name**. By default, the XForms root node name is `data`.

5.62 Multiple webpage forms

Web forms can be split into multiple pages using the style theme **pages.**
In the **settings** tab, create a column called **style** and set it to **pages**, as follows:

form_title	form_id	style
example title	example_id	pages

In your **survey** tab, group together the questions you would like to appear on each page and then set the appearance for the group to **field-list**. See the example below.

type	name	label	appearance
type	name	label	appearance
begin group	group1		field-list
text	name	Respondent's name	
integer	age	Respondent's age	
text	address	Respondent's address	
end group			

5.63 Grid theme forms

The **theme-grid** style allows your form to mimic the look of traditional paper surveys by compacting multiple questions into one row. This style is best used with larger screens (e.g., computers or tablets). It also makes a nice print out!

To create a Grid form, in the **settings** tab, under the **style** column, write **theme-grid**, as follows:

form_title	form_id	style
example title	example_id	theme-grid

In your **survey** tab, group together the questions you would like to appear in each section and then set the appearance for each field according to the desired width (the default width is 4). See the example below.

type	name	label	appearance
begin group	group1		
text	name	Respondent's name	w3
integer	age	Respondent's age	w1
text	address	Respondent's address	w4
end group			

5.64 Styling prompts

Markdown support in XLSForm allows for increased emphasis through bold and italics, different sized headers, various fonts and colors, and clickable web links in ODK Collect 1.4.9 and Enketo.

emphasize words by wrapping them inside _ or `*`

strongly emphasize words by wrapping them inside __ or `**`

add a link by using `[name of link](url)`

add various sized headers by prepending # (biggest) to ###### (smallest) to header text
style text for color or font with span tags (e.g., `<span style="color:#f58a1f">orange</span>`, `<span style="color:red; font-family:cursive">red and cursive</span>`)
add a line break where you want it with Ctrl-Enter or Ctrl-Alt-Enter (may be different key combination for some spreadsheet software)
add your favorite emojis 😍📋😍!
use superscript with the `<sup>` tag (e.g. `100 m<sup>2</sup>` turns into 100 m^2)
use subscript with the `<sub>` tag (e.g. `H<sub>2</sub>O` turns into H_2O)
use the `\` character before `#`, `*`, `_`, and `\` to prevent special styling effects to be triggered by these characters

5.65 Advanced use and extensibility

It is possible to use XLSForm to create XForms with custom or experimental features. This is great for custom applications with a specific feature that is not suitable for the larger community.
The **survey** sheet has support for 3 column prefixes **(instance::, bind::, body::)** that add attributes to the XForm output, either in the *primary instance*, *bind*, or *form control*.

type	name	label	instance::hxl
integer	population	How many people present?	#population

The **settings** sheet has support for defining (multiple space-separated) additional custom namespaces and namespace prefixes using the **namespaces** column. You'll then be able to use those namespaces in the survey sheet, for example to properly define a custom attribute with your organisation's own namespace. See example below that adds 2 additional namespaces and uses them to add custom attributes:

title	namespaces
My Form	esri="http://esri.com/xforms" enk="http://enketo.org/xforms

surveychoices**settings**

type	name	label	bind::esri:fieldLength	bind::enk:for
text	desc	Describe	50	
text	desc_comment	Comments		${a}

surveychoicessettings

5.66 More resources

The XLSform standard document can guide you through the specific input types, column headers, and so on that are legitimate syntax in XLSForms. If you want to dig in deeper to understand XForms and go beyond XLSForms, here are some resources to understand them:

- XForms as supported by the ODK ecosytem
- ODK Form design guidelines

- Ona Form design overview

5.67 History

The XLSForm was originally developed by Andrew Marder and Alex Dorey of the Sustainable Engineering Lab at Columbia University. As XLSForms became adopted by the ODK Community, SEL worked with the ODK Team to develop the current specification. PyXForm, the library used to convert XLSForms to XForms, is an open-source project supported by members of ODK, SEL, Ona, and SurveyCTO.

6 OVERVIEW ON DATA COLLECTION TOOLS

DIGIKYAR allows data collection in multiple ways. Because DIGIKYAR is built on the Xform/ODK standards, our forms are compatible with a number of different tools that can be used for data collection.
For Android devices, we recommend using ODK Collect Andoid app which can be downloaded from the Google Play Store and installed on any standard Android phone or tablet.
For any other devices (including iPhones, iPads, or any laptop or computer), we recommend using the webform for collecting data.

6.1 The DIGIKYAR Operation Process

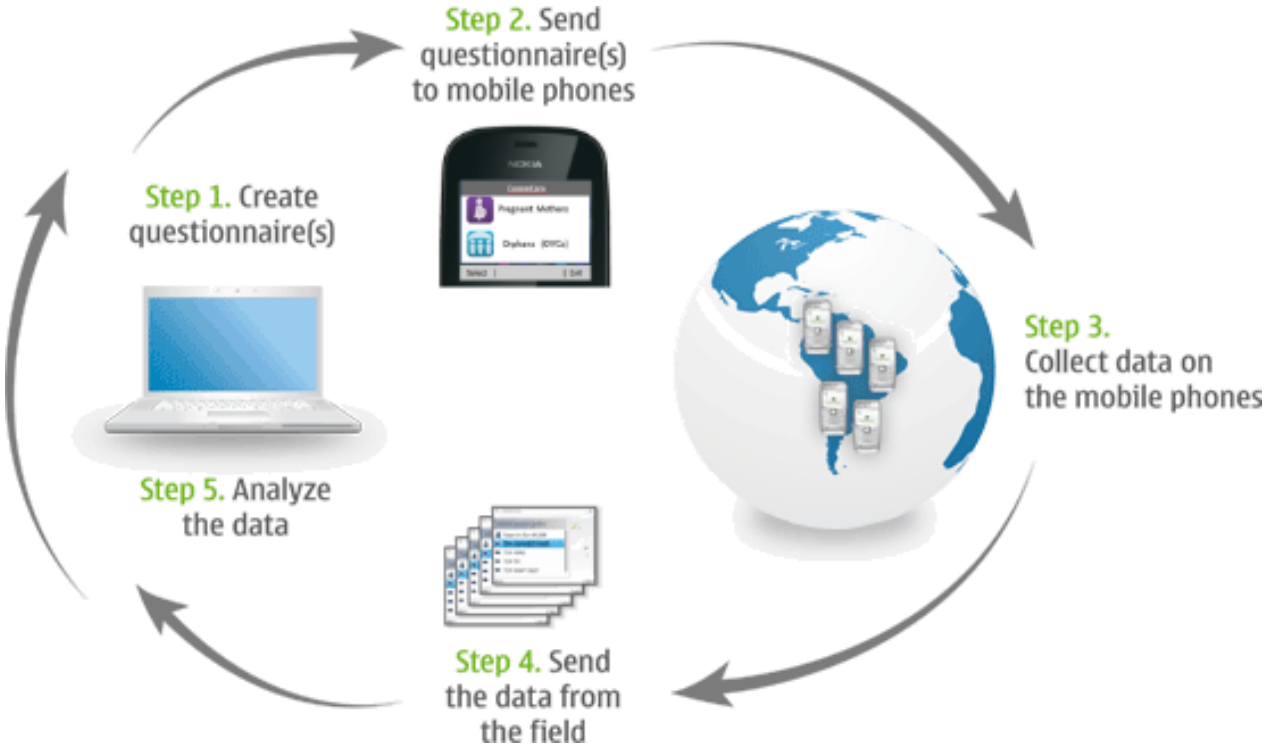


Figure 18: DIGIKYAR operation process

6.2 Simple Data Entry & User-Friendly Interface

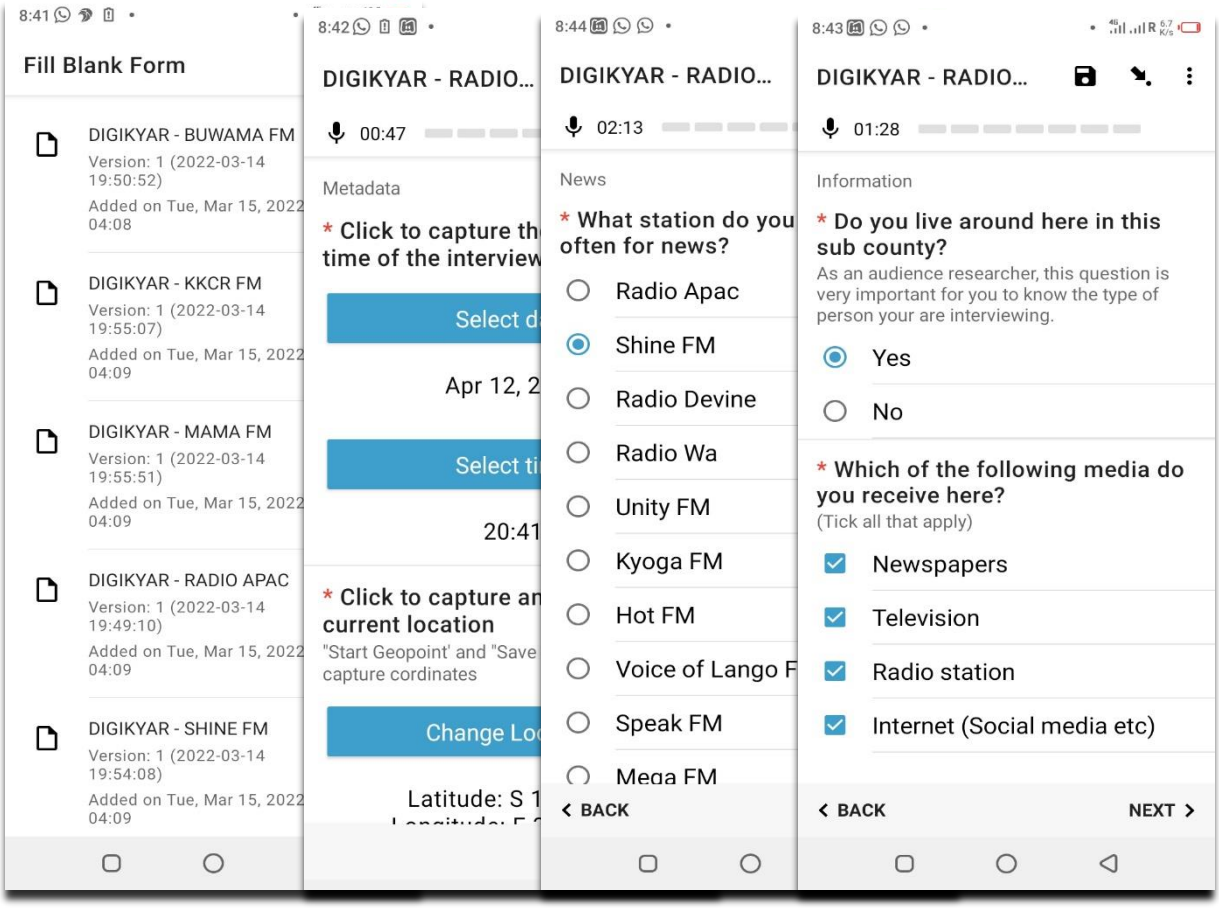


Figure 19:DIGIKYAR simple and user-friendly interface

6.3 Steps on using DIGIKYAR – ODK Application.

Step 1 - Download and install ODK Collect, an android-based toolbox from Google play store.

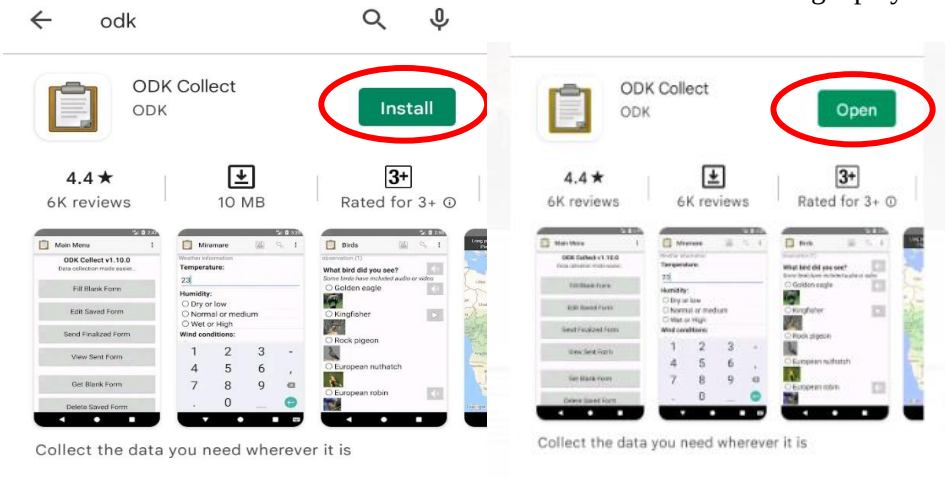


Figure 20:Downloading and installing ODK Collect

Step 2 – After Installation, open the application and click on Manually enter project details

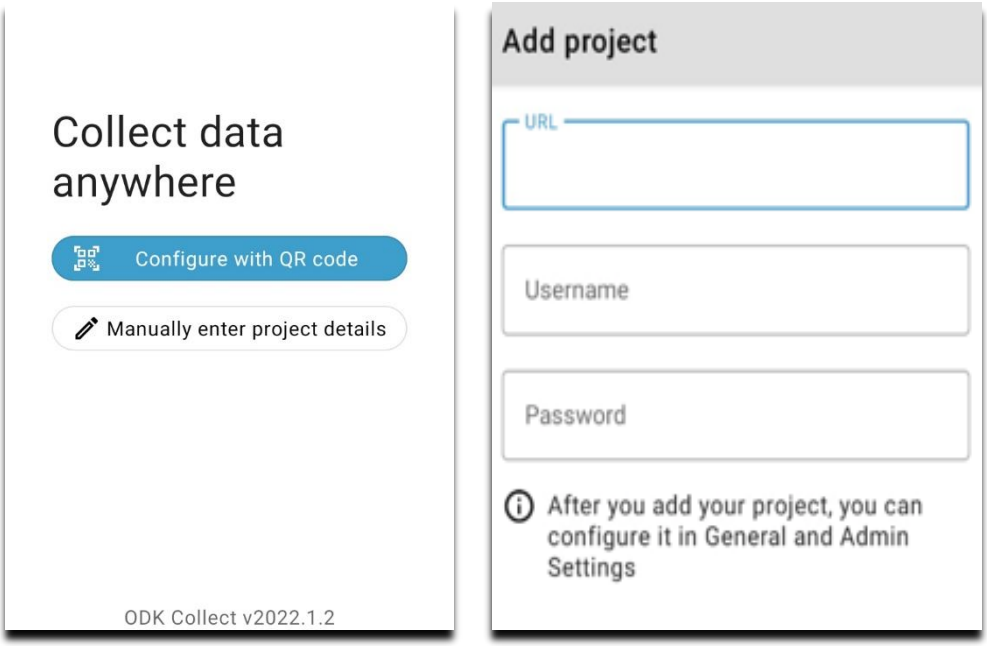
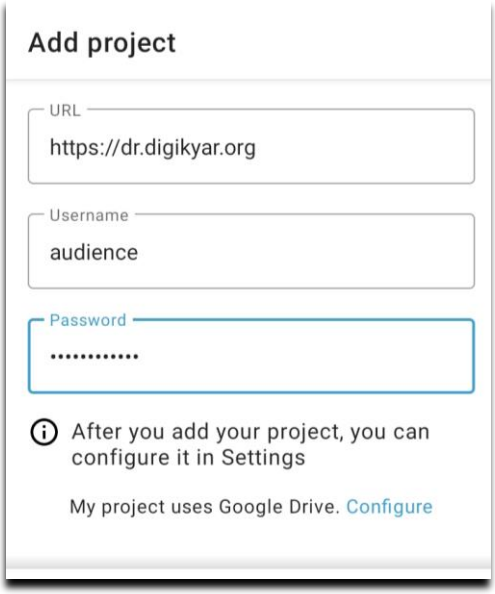


Figure 21: Accessing DIGIKYAR via ODK Android Application

Step 3 – Enter the DIGIKYAR server URL, username and password to login and access all the deployed content.



URL: https://dr.digikyar.org

USERNAME: audience

PASSWORD: xxxxxxx

Figure 22: Logging into the DIGIKYAR Application Server

Step 4 – Login, choose Get blank form and select the DIGIKYAR App form to get started.

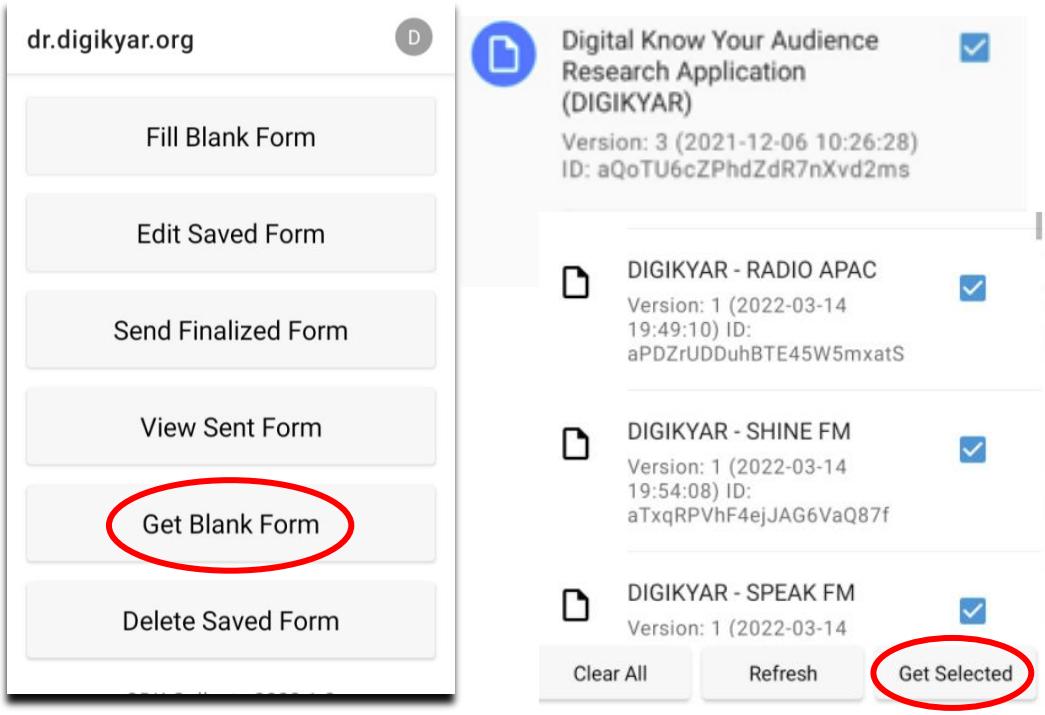


Figure 23:Get blank form and select the DIGIKYAR App form to get started.

Step 5 – Choose to fill blank form and get started with the audience survey while pressing next for other questions.

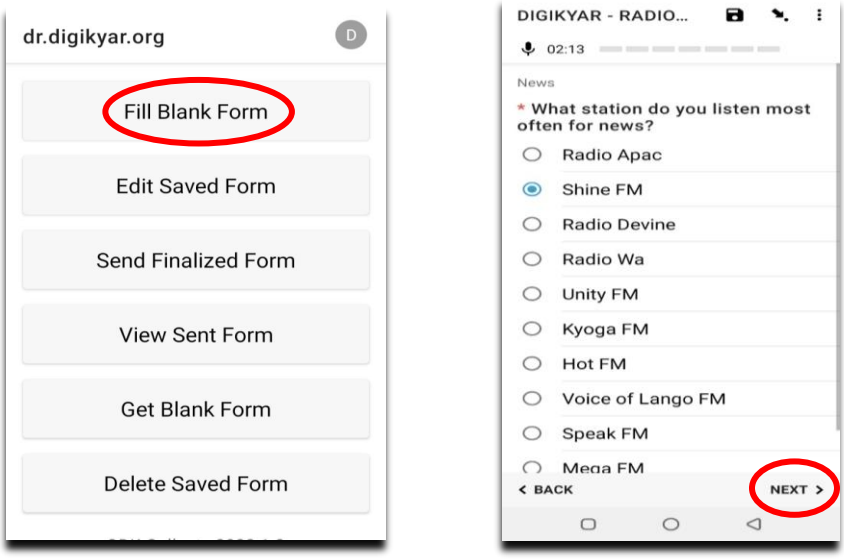


Figure 24:Select fill blank form and get started

Step 6 – After the survey, save the form and exit and this will be viewed or edited in the saved forms. Note that you do not need internet to conduct and save the survey/interview.

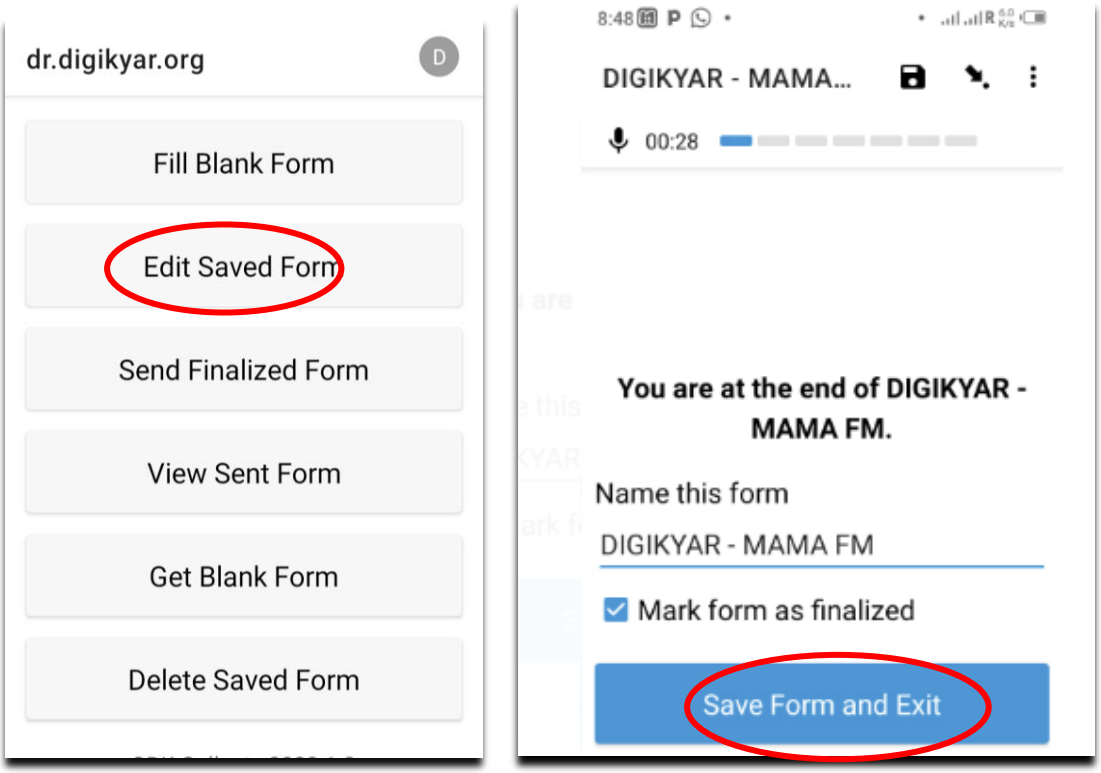
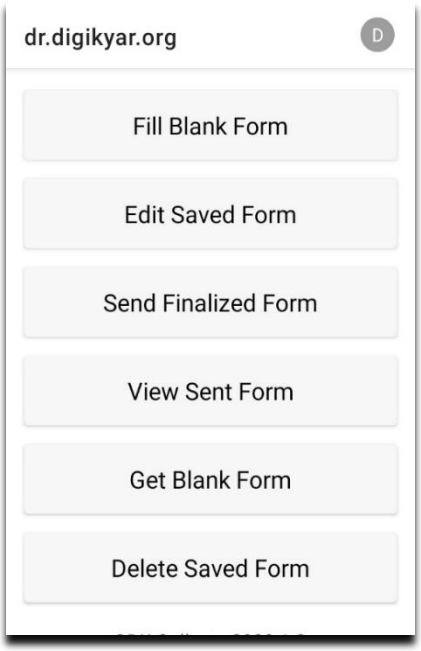


Figure 25: Save and view the edited form

Step 7- Later when you are able to connect to stable internet, you can then send finalized forms.



]

Figure 26: Sending finalised forms to Digikyar server

7 MANAGING PROJECTS AND DATA

7.1 Project Summary

The Project Summary tab gives you a visual display of all the high-level information related to your project. Depending on whether you're the project owner or a user with restricted permissions, different information will be shown to you. The Project Summary includes:

- **Description:** Includes all project metadata.
- **Submissions:** A tally and graph of the submissions (up to the past 31 days).
- **Form details**: Includes last modified, last submission, and number of questions.
- **Quick links:** Useful links related to data collection for the project.
- **Data:** Useful links to information related to submitted data.
- **Team members:** List of users that have access to the project.

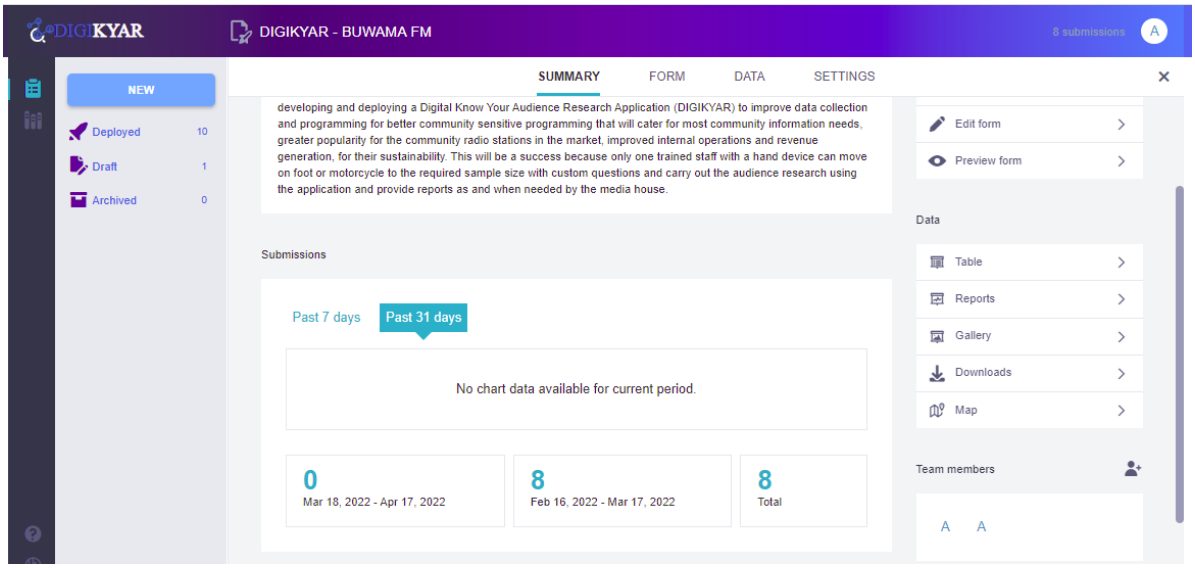


Figure 27: Project summary

7.2 Exporting and Downloading Your Data

You can download your data in multiple formats. As soon as any data has been collected, the following formats are available for download:

- XLS (formatted spreadsheet)
- XLS (legacy version)
- CSV (comma separated values)
- CSV (legacy version)
- Media Attachments (a zip folder of your photos, videos, or sound recordings - if applicable)

- KML (a file containing all the GPS points collected - if applicable)
- Excel Analyser
- SPSS Labels

XLS and CSV (non-legacy) export types have additional, one-time options that allow you to choose the value and header format, decide whether or not to include groups in headers, and customize the group separator.

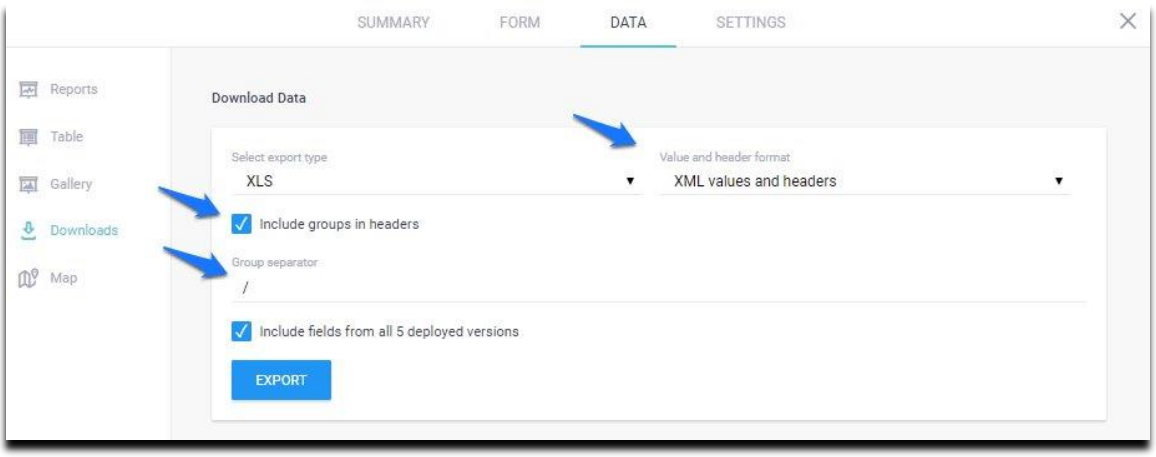


Figure 28: Exporting and Downloading Data

7.3 Process of downloading data

- After setting up the format and type of data, click the Export button.
- Each new export can take a few seconds to be created and will be shown below in the Exports section of the page.
- Each export is retained as a unique snapshot in the system, so it is possible to go back to previous non-deleted exports.

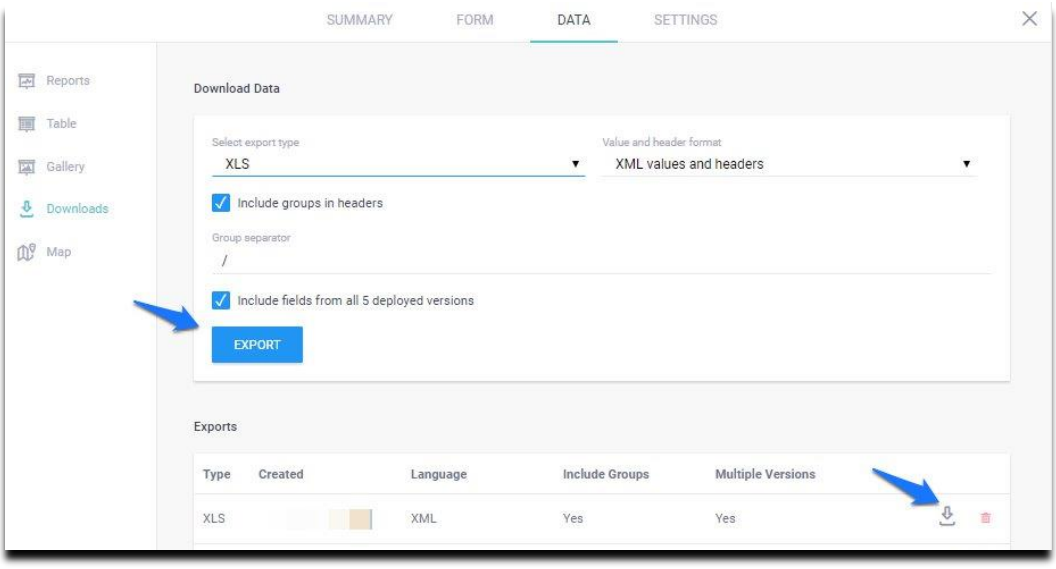


Figure 29:5.7 Process of downloading data

7.4 Viewing and Creating Custom Reports

On the reports page in the Data section of your project, you can create and configure multiple summary reports with custom graphs, tables, and color schemes. The custom reports are shareable by either printing, saving as a PDF, or storing them on an external cloud service.

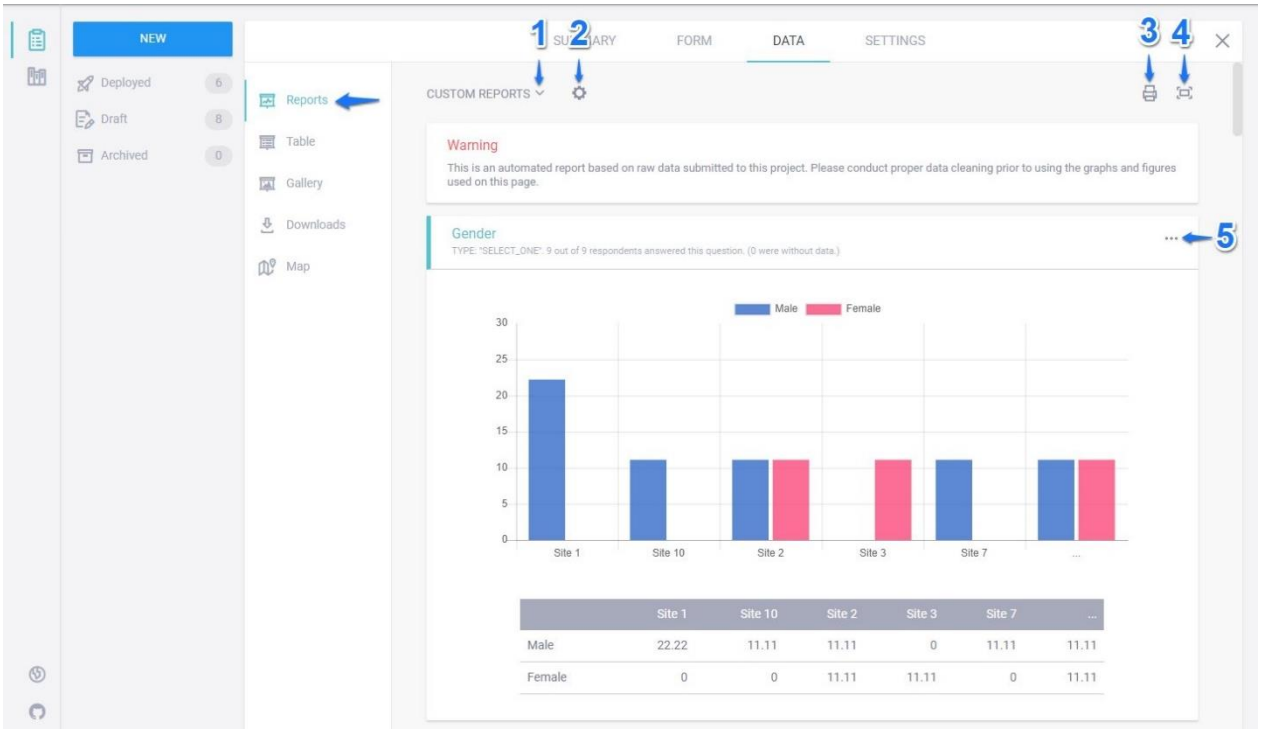


Figure 30:Viewing and Creating Custom Reports

The CUSTOM REPORTS allow you to view the Default Report based on the dataset available in your survey project. It also allows you to Create New Report (by giving it a unique title and selecting which questions to include or exclude) or search through previously created custom reports.

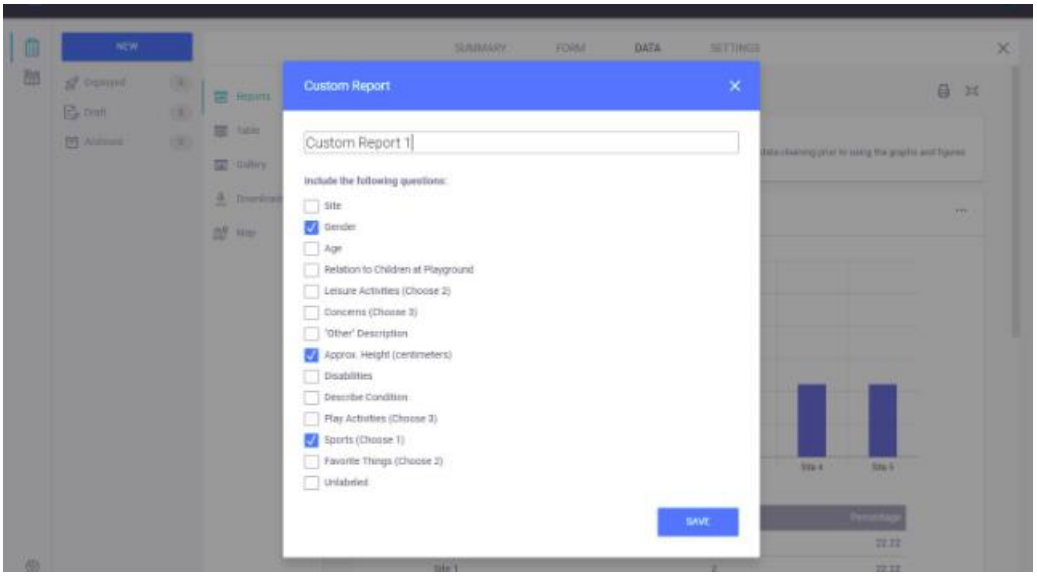


Figure 31: Creation of custom reports

7.5 DIGIKYAR Edit Report Style

The Edit Report Style button allows you to change the CHART TYPE to Vertical, Donut, Area, Horizontal, Pie and Line. It also allows you to choose different COLORS for your charts.

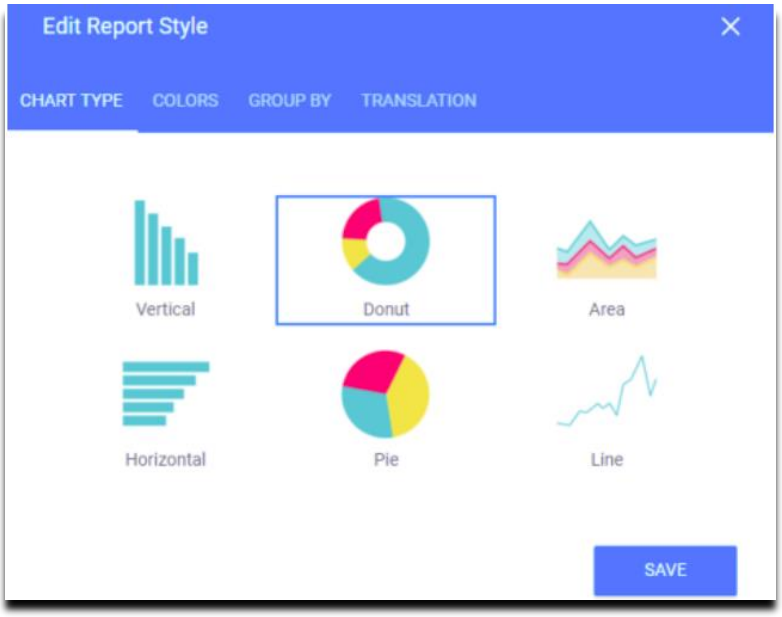


Figure 32:DIGIKYAR report style

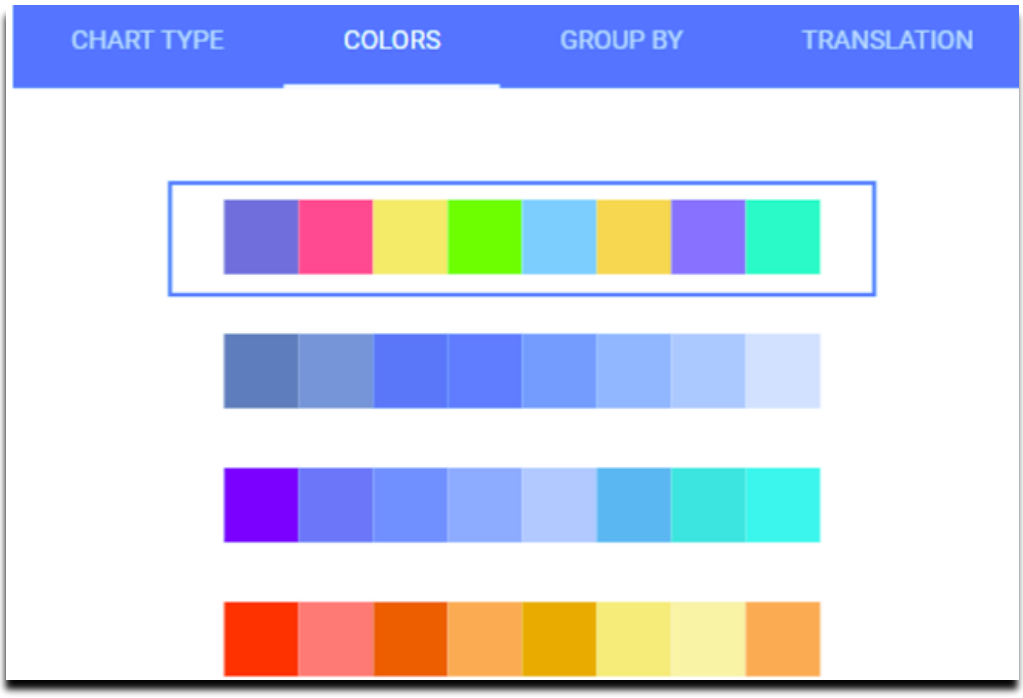


Figure 33: Working with color styles

7.6 Printing DIGIKYAR reports

You could also print the custom report by clicking the Print icon or save it as PDF by selecting Save as PDF under Destination.

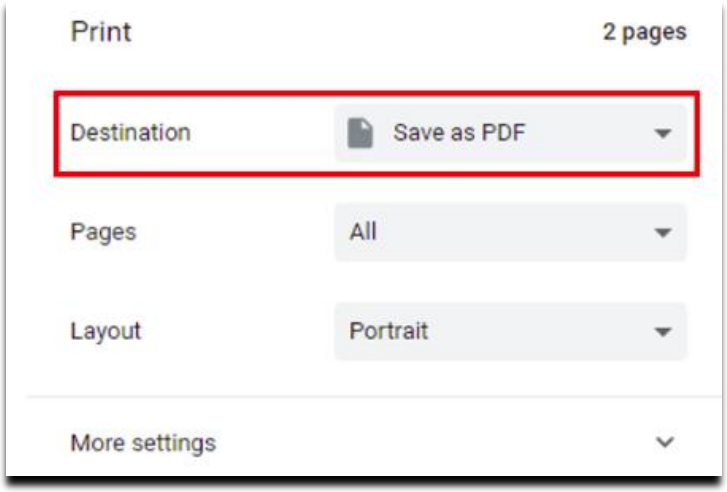


Figure 34:Printing DIGIKYAR reports