



Office Automation for HBCSE

Arshdeep Singh, ID: 2018AAPS0436H

Prince Ajmera, ID: 2018A3PS0906H

Mentors – Prof. Nagarjuna G, Mr. Surendra Patil, Ms. Adithi M.
PS Instructor: Dr. Rajib Ranjan Maiti, BITS Pilani -Hyderabad
Campus



BITS Pilani
Hyderabad Campus

Acknowledgement



- We are using this opportunity to express our gratitude to everyone who supported us throughout the course of the project.
- We are thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work.
- We would like to thank our Project In-Charge **Prof. Nagarjuna G.** who provided us all the facilities required and conducive conditions for accomplishment of this project .
- We would like to thank **Ms. Adithi** for her constant guidance and valuable remarks.
- We would also like to thank our project mentor **Mr. Surendra Patil** for his constant support during the project.
- We would also like to thank our PS Faculty **Dr. Rajib Ranjan Maiti** for his support in completion of this project.

Abstract



- Our project was aimed to collecting data from employees at HBCSE and generate reports.
- All this work was previously done by hand but now it could be accomplished by the use of python scripts and a good data collecting platform.

Choosing the Platform for Data Collection



- We first went with google forms. But we had to drop the idea of google forms due to the following reasons:
 - Google forms do not provide much flexibility for making forms.
 - Issue of data security since all the data is stored on google's servers.
- We were then suggested to use KoBoToolbox. It is free data collection platform that provides much more flexibility for making forms.

KoBoToolbox



- KoBoToolbox is a suite of tools for field data collection. It is a free and open source software.
- The forms on KoBoToolbox are based on XLSForm.
- Forms can be made either by using the GUI provided by KoBo or simply can be made in XLS format and then uploaded.
- However we suggest making the forms in XLS format as they align better to the scripts and are easy and quick to make.



The UI for KoBoToolbox can be seen below.

The screenshot displays the KoBoToolbox web interface. At the top left is the KoBoToolbox logo. A search bar labeled 'Search Projects' is located at the top center. On the right side of the header, there is a user profile icon with the letter 'A'. The left sidebar contains navigation options: 'NEW' (highlighted in blue), 'Deployed' (28 items), 'Draft' (1 item), and 'Archived' (0 items). Below these are icons for help, home, and user profile. The main content area is titled 'Deployed' and contains a table with the following data:

Name	Shared by	Created	Last Modified	Submissions
Links For All The Forms	prince	Today at 9:45 AM	Today at 9:55 AM	0
Involvement in Administrative Bodies	prince	May 30, 2020	Yesterday at 11:22 PM	3
Awards, Honours, Distinctions	prince	May 30, 2020	Yesterday at 11:22 PM	3
Involvement in Academic Bodies	prince	May 30, 2020	Yesterday at 11:21 PM	3
Lectures / Lecture Courses Given Elsewhere	prince	May 30, 2020	Yesterday at 11:21 PM	3
Invited Talks	prince	May 30, 2020	Yesterday at 11:20 PM	4
Conference(s) / Meeting(s) Organised	prince	May 30, 2020	Yesterday at 11:19 PM	3
Courses Taught at TIFR (Including Centres and Facilities)	prince	May 30, 2020	Yesterday at 11:18 PM	3
Popular Science Lecture(s)/ Pedagogical Lectures	prince	May 30, 2020	Yesterday at 11:17 PM	5
Outreach Activity	prince	May 30, 2020	Yesterday at 11:16 PM	9

Making Forms on KoBoToolbox using XLSForm



- XLSForm is a form standard created to help simplify the authoring of forms in Excel. Authoring is done in a human readable format using a familiar tool that almost everyone knows - Excel. XLSForms provide a practical standard for sharing and collaborating on authoring forms. They are simple to get started with but allow for the authoring of complex forms by someone familiar with the syntax.
- The XLSForm is then converted to an [ODK XForm](#), a popular open form standard, that allows you to author a form with complex functionality like skip logic in a consistent way across a number of web and mobile data collection platforms.



Basic Format of XLSForm

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

The Syntax for XLSForm is case sensitive so the sheet names should be in lower case for the form to be properly executed.

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.

- 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.

Example of survey worksheet.

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

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).

- The **choices** worksheet has 3 mandatory columns as well: **list_name**, **name**, and **label**.
 - The **list_name** column lets you group together a set of related answer choices, i.e., answer choices that should appear together under a question.
 - The **name** column specifies the unique variable name for that answer choice.
 - The **label** column shows the answer choice exactly as you want it to appear on the form. Alternatively, [label translation columns](#) can be used.

Example of choices worksheet

list_name	name	label
gender	transgender	Transgender
gender	female	Female
gender	male	Male
gender	other	Other
surveychoicessettings		

The 'settings' Worksheet



- The settings worksheet is an optional worksheet that allows the user to define title, version, id etc. of the form. It also allows the user to change the theme of the form.

Question types available in XLSForm



- There are many question types available for XLSForm. All the question types can be viewed on this link: [Question types](#).
- Some important question types that were used for our project are:
 - text – to get a text input
 - Integer – to get integral input
 - select_one [list name] - for multiple choice question with single option
 - date – to get input of date

Some Important Functionalities of XLSForm



- There are several important functionalities of the XLSForm that were used for the project. We will go through them briefly one by one.
- The whole documentation for XLSForm can be viewed at <http://www.xlsform.org>.
- These functionalities are:
 - Hints
 - Constraints
 - Relevant
 - Required
 - Grouping Questions
 - Repeats

Creating Multiple Choice Questions



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?
survey choicesettings		

list name	name	label
yes_no	yes	Yes
yes_no	no	No
survey choicesettings		

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.	

surveychoicessettings

- There are other types of hints also. They are called ‘guidance hints’. They do not appear in the form normally. They are visible only in special views.
- They can be used for a form simply by adding a column ‘guidance_hint’.

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

surveychoicessettings

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.
- A constraint message can also be added by including 'constraint_message' column in the survey worksheet.

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

surveychoicessettings

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

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

surveychoicessettings

- The favourite toppings field will only be displayed if the question 'Do you like pizza?' is opted as yes.

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

surveychoicessettings

- A message can also be displayed if a required question is not answered. It can be done by including 'required_message' column in survey worksheet

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		
surveychoicessettings		

- 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.

Grouping Questions continued...



- Groups can also be nested. This can be done by writing another 'begin group' tag within a group. Note that 'end group' closes the recent group created.
- A particular can be skipped or made to appear under special conditions by using 'relevant' column.

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		
surveychoicessettings		

list name	name	label
male_female	male	Male
male_female	female	Female
surveychoicessettings		

Repeats continued...

- 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.
- It is always a good practice to group questions inside a repeat block by using 'begin group' and 'end group' construct inside a repeat construct.

- The user can also assign fixed repeat counts for a particular repeat construct by adding 'repeat_count' column and specifying number of repeats.
- Also repeat block can be skipped or made to appear under specific conditions by using 'relevant' column.

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

surveychoicessettings

list_name	name	label
yes_no	yes	Yes
yes_no	no	No

surveychoicessettings

Example of 'Workshops' Form



- In workshops form, each category selected maps to a different set of questions. We will briefly explain the implementation of this form using XLSForm and also will clear some points to remember while making new forms.

	A	B	C	D	E	F	G	H	I	J
1	type	name	label	hint	required	relevant	required_message			
2	text	reporting_mem	Reporting Member:		yes		This field is required			
3	text	designation	Designation:		yes		This field is required			
4										
5	begin repeat	rep								
6	select_one S_T_PST_TeR	type_w	Type of Workshop:							
7										
8	begin repeat	student_workshop					\$(type_w) = 'S'			
9	begin group	students	Student Workshop							
10	text	titile_s	Title of the Workshop:	eg: Learnir	yes		This field is required			
11	integer	number_s	No. of student participants:		yes		This field is required			
12	integer	numbert_s	No. of teacher participants (if any):							
13	text	coord_s	Workshop Coordinator from HBCSE:		yes		This field is required			
14	integer	nuberres_s	No. of HBCSE Resource Persons who conducted sessions:		yes		This field is required			
15	integer	numbrnonres_s	No. of external (non-HBCSE) Resource Persons who conducted sessions (if any):							
16	date	start_date	Starting Date of the Workshop:		yes		This field is required			
17	date	ending_date	Ending Date of the Workshop:		yes		This field is required			
18	text	wokshop_collab	Workshop Collaborators (if any):	eg: in collaboration with Nehru Science Centre						
19	text	venue_s	Venue of the Workshop:	eg: HBCSE	yes		This field is required			
20	end group									
21	end repeat									
22										
23	begin repeat	Teacher_workshop					\$(type_w) = 'T'			
24	begin group	teacher_w	Teacher Workshop							
25	text	titile_s	Title of the Workshop:	eg: Pedagogy	yes		This field is required			

lead

- Here, the fields 'Reporting Member' an 'Designation' are written for the non-repeating part.
- Then a multiple choice question asks for the category of workshop.
- Also, note that the 'begin repeat' construct only appears if a particular category is selected. This is implemented using the **relevant** column of survey sheet.
- It is always a good practice to include questions within a repeat construct to be within a group.
- Also, take a look at the name table of repeat constructs. The name that is written for the repeat constructs in the name column, it appears as the 'sheet name' in the exported data for this form .So, for correct implementation of the script, it is advisable to type the **name** column similar to the **label** column

The forms we made



All the forms were implemented in a similar fashion.
Some forms had single set of repeating questions and
some had none.

Now we will move to the script part of the project.

Script for Report Automation



- The process by which the script generates the report is as follows:
 - First, the data is downloaded off from KoBoToolbox
 - Then a template file is created which contains all the column headers for all the different forms. That file can be edited for changing the order of columns or even deleting some columns. Also a tabular or a list format can be specified for a particular form in that file itself.
 - Then the report is generated in Markdown format following the template provided in the template. The report can be visualised in any markdown editor.
- So, we made 3 different scripts for fulfilling the mentioned purposes.
 - Download_excel.py – for downloading data from KoBoToolbox.
 - Headers.py – for generating the template file.
 - Report.py – For generating the final report.
- First we will explain the working of these 3 scripts in brief then we will demonstrate the working of these scripts.

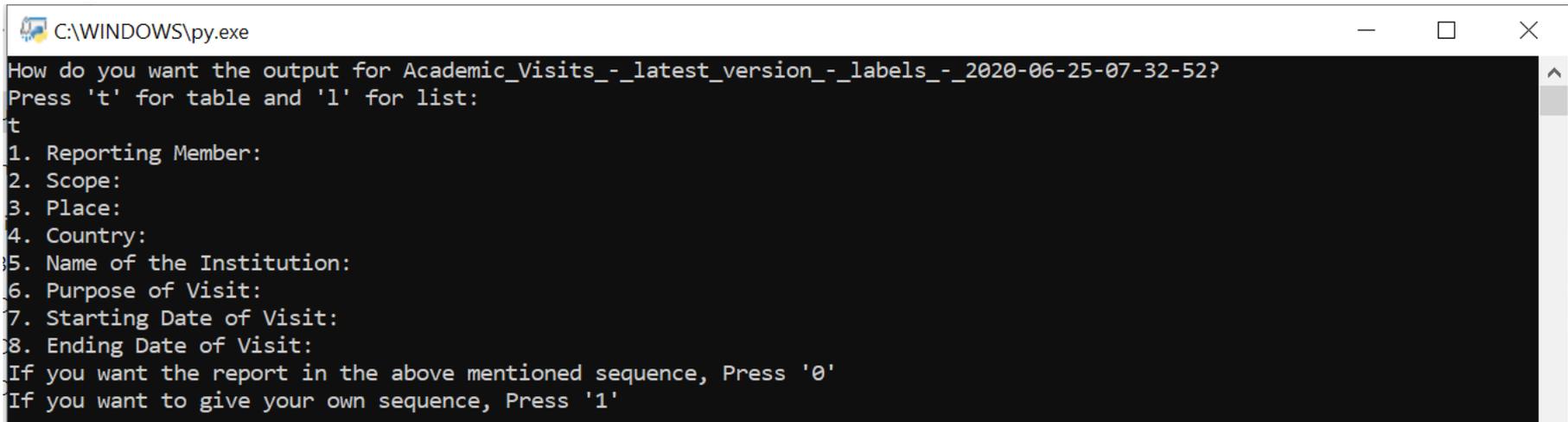
Download_excel.py

- This script features sample code/snippets to pull or push data using the new KoboToolBox API.
- Data is first exported and the URL containing the latest excel file is generated.
- All these URLs are then used to download all the excel files in the Excel Folder.

- We have currently three different kinds of forms:
 - Forms with no repeating set of questions.
 - Forms with a single set of repeating questions.
 - Forms with multiple sets of repeating.
- Headers.py works in different ways for these 3 different types of forms:
 - For forms with no repeating parts, data is present in only one excel sheet so it straightaway extracts the relevant columns from the spreadsheet.
 - For forms with single set of repeating questions, The questions under 'repeat' construct are present in a different sheet so the sheets are to be merged on the basis of the column containing unique id for each submission, which is '_uuid', and then the column headers are extracted.
 - For forms with multiple sets of repeating questions, data for each repeating set is present in a separate sheet so the sheets are to be merged sequentially. By doing this we extract columns for each separate category separately and reports for each category is generated separately.

Headers.py continued...

- When the script is run, it asks whether the report is to be generated in Table form or in List form which has to be entered by the user.
- Also, templates for the forms are shown which can be changed by the user.
- Preview:



```
C:\WINDOWS\py.exe
How do you want the output for Academic_Visits_-_latest_version_-_labels_-_2020-06-25-07-32-52?
Press 't' for table and 'l' for list:
t
1. Reporting Member:
2. Scope:
3. Place:
4. Country:
5. Name of the Institution:
6. Purpose of Visit:
7. Starting Date of Visit:
8. Ending Date of Visit:
If you want the report in the above mentioned sequence, Press '0'
If you want to give your own sequence, Press '1'
```

Example of template file



The template is generated in a text file called **template.txt**

```
template - Notepad
File Edit Format View Help
t;Academic_Visits_-_latest_version_-_labels_-_2020-06-25-07-32-52;acad_visits;Reporting Member;;Scope;;Place;;Country;;Name of the Institution;;Purpose of Visit;;Starting Date of Visit;;Ending Date of Visit;;
t;Any_other_Information_-_latest_version_-_labels_-_2020-06-25-07-31-45;any_info;Category;;Reporting Member(s);;Title of the Project;;Description;;
t;Awards_Honours_Distinctions_-_latest_version_-_labels_-_2020-06-25-07-22-30;group_wd2jl00;Reporting Member;;Scope;Name of the Item;Awarding/Electing Body;Full Citation (if relevant);Amount;Date;
t;Conference(s)___Meetings_Organised_-_latest_version_-_labels_-_2020-06-25-07-18-06;group_bd6jw63;Reporting Member;;Scope;Name of the Meeting;Host Institution(s);Place;Role;Funding Source(s);Starting Date;End Date;
t;Conferences_Workshops_Attended_-_latest_version_-_labels_-_2020-06-25-07-17-00;con_workshop;Reporting Member;;Scope;Name of the Meeting;;Organising Institution;;Role;;Place;;Country;;Date;;
t;Courses-Taught-at-TIFR-Including-Centres-and-Facilities_-_latest_version_-_labels_-_2020-06-25-07-15-54;group_tp36z97;Reporting Member;;Type;Semester;Subject Board;Name of the Course;Nature of Involvement;
t;Degree(s)_Awarded_Supervised_Ph.D._M.Sc.M.Phil._-_latest_version_-_labels_-_2020-06-25-07-07-09;degrees_awarded_supervised;Reporting Member;;Degree;Name of the Awardee (w. prefix);Awarding Institute;Title of the Theses;Name of Supervisor (if relevant);Date;
t;Degrees_awarded_to_Members_of_the_Unit_-_latest_version_-_labels_-_2020-06-25-07-30-40;rep;Broad area;;Degree;;Name of the Awardee;;Title of Thesis;;Name of Supervisor;;Awarding Institution;;Date;;
t;Graduate_Courses_offered_by_Members_of_the_Unit_-_latest_version_-_labels_-_2020-06-25-07-25-00;grad_courses;Broad Area;;Level;;Title of the Course;;Name of the Instructor(s);;Semester;;
t;Invited_Talks_-_latest_version_-_labels_-_2020-06-25-07-19-13;group_pp7zp94;Reporting Member;;Scope;Title of the Talk;Occasion;Delivered at (Institution);City;Country;Date;
t;Involvement_in_Academic_Bodies_-_latest_version_-_labels_-_2020-06-25-07-21-25;group_hf2md68;Reporting Member;;Scope;Role;Name of the Body;Organisation;Starting Date;End Date;
t;Involvement_in_Administrative_Bodies_-_latest_version_-_labels_-_2020-06-25-07-23-52;group_dp7zj85;Reporting Member;;Scope;Role;Name of the Body;Organisation;Starting Date;End Date;
t;Lectures_by_non-TIFR_Visitors_in_the_Unit_-_latest_version_-_labels_-_2020-06-25-07-28-29;lec_non_tifr;Nature of Talk;;Name of the Speaker;;Title of the Talk;;Home Institution;;Place;;Country;;Date;;
t;Lectures__Lecture_Courses_Given_Elsewhere_-_latest_version_-_labels_-_2020-06-25-07-20-18;group_xl66s13;Reporting Member;;Scope;Level;Occasion;Title of the Course / Set of Lectures;Place;Country;Starting Date;Ending Date;
t;List_of_Members_of_the_Unit_-_latest_version_-_labels_-_2020-06-25-07-29-35;List of Members of the Unit;Reporting Member;;Are you a Staff Member or, Student or Postdoc?;Category;;Category;.1;
t;Major_Conference(s)___Meetings_Organised_by_the_Unit_-_latest_version_-_labels_-_2020-06-25-07-26-17;Sheet1;
t;Mentoring_of_Students_and_Visiting_Fellows_TIFR_ONLY_-_latest_version_-_labels_-_2020-06-25-07-06-00;mentoring_of_students;Reporting Member;;Level;Name of the Student VF (with prefix);Destination (if relevant);Starting Date;Ending Date;
t;Non-TIFR_Research_Projects_Grants_-_latest_version_-_labels_-_2020-06-25-07-03-47;group_ej6oi05;Reporting Member;;Scope;Title of the Project;Role;Amount (Rs. Lakhs);Funding Agency;Starting Date;Duration (Months);
t;Outreach_Activity_-_latest_version_-_labels_-_2020-06-25-07-13-41;group_ir2fk74;Reporting Member;;Scope;Title of the Activity;Nature of the Activity;Host
```

Report.py



Report.py simply reads the template file for the sequence of the column headers and extracts the data accordingly. It also sorts the data in descending order of 'Date' or 'Year' column, if any, in the form.

Sample Report Generated by our Script



STUDENT PROJECTS GUIDED TIFR AND NON-TIFR

Reporting Member:	Level	Name of the Student (with prefix)	Student's Home Institution	Title of the Project	Duration (Months)
K. K. Mashood	Masters	Mr. Prithu Raj Ghosh	TISS Hyderabad	Students misconceptions in basic mechanic	1
Sugra Chunawala	Informal	Ms. Vijaylaxmi Bhat	University of Mumbai	Designing activities for school children	3
R. B. Khaparde	UG	Mr. Dhiman Biswas	RKVM, Kolkata	Understanding Mirage	2
R. B. Khaparde	UG	Mr. Y Aditya Varma	IITDM, Chennai	Magnetic Damping	3

VISITORS HOSTED

Reporting Member:	Scope	Name of the Visitor (with prefix)	Visitor's Home Institution	Country	Purpose of Visit	Starting Date	Ending Date
Adithi	National	Veena Vijayan	M.A. Sociology, University of Hyderabad	India	Purpose of Visit will come here	24.09.2019	20.03.2020
Adithi	National	Parbati Murmu	M. A. Education, TISS, Hyderabad	India	Purpose of Visit will come here	13.05.2019	13.06.2019
Nagarjuna G.	National	Fifteen students part of Practice School	Birla Institute of Technology, Pilani	India	Practise school supporting Gnowledge Lab projects	22.05.2018	13.07.2018

Everything as a Git-Hub Repository



As suggested by our mentors, we have uploaded all the work on GitHub:

https://github.com/Ar5h71/XLS_to_MARKDOWN_REPORT

Concluding Remarks:



- By the constant support and guidance by the mentors and our PS faculty, we have successfully completed the project.
- We were successfully able to generate reports in the desired fashion with very less editing required from the user end.
- We hope that this project will be carried forward and will help the people at HBCSE in a constructive way.



THANK YOU