Make Your Family-Tree App Explained

View, insert, delete & Update members
From anywhere around the globe

Imagine a comprehensive Family Tree with multiple generations, structured hierarchically and organized in an Excel spreadsheet, as illustrated below.

Gen-1	Gen-2	Gen-3	Gen-4	Gen-5	Gen-6	Gen-7	Gen-8
BABA JUDAN							
	Kangali						
		Mohar					
			Sahadul				
				Quayum			
					Bilquis		
						Tarique	
							Mayera
						Huma	
							Samara
					Zafar		
						Shadab	
							Aadab

To create a universally accessible, web-based Family Tree, we aim to develop a platform where family members from anywhere in the world can seamlessly view, add, update, or remove member details with ease.

Step-1:

- Install Python from Microsoft Store in your System
- Open Command Prompt as Administrator
- Run the following command:
  - **winget install Python.Python**
    - Python will be installed in your system
    - create a directory say 'Python' in drive C:
    - **C:/>cd Python**
    - Install Python dependency library *openpyxl*
  - **C:\Python>pip install pandas openpyxl**

Step-2:

Set database Columns as: *id, name, parent_id, generation*

Convert Hierarchical Family Tree in Excel **xlsx** to **csv** format using Python Script in command prompt:

Python Script File: **conv_csv.py** conversion from **xlsx** to **csv**

```
import pandas as pd

# Define input and output file paths
excel_path = "family_tree.xlsx"  # Change this to your actual
file path
csv_path = "family_tree.csv"

# Load the Excel file
df = pd.read_excel(excel_path, engine="openpyxl")

# Process hierarchical data into structured format
family_tree = []
id_counter = 1  # Unique ID counter
parent_stack = {0: None}  # Stack to track parent-child
relationships

# Iterate through rows to extract names and hierarchy levels
for index, row in df.iterrows():
    for level, value in enumerate(row):
        if pd.notna(value):  # If cell is not empty
            parent_id = parent_stack.get(level - 1, None)  #
Get parent ID from previous level
            family_tree.append({"id": id_counter, "name":
value, "parent_id": parent_id, "generation": level + 1})
            parent_stack[level] = id_counter  # Update parent
reference for next level
            id_counter += 1

# Convert to DataFrame
structured_df = pd.DataFrame(family_tree)

# Save as CSV
structured_df.to_csv(csv_path, index=False)

print(f"CSV file '{csv_path}' created successfully!")
```

Step-2 (Alternate):
Another Python script for conversion of Hierarchical data Excel **xlsx** to **csv** format can be downloaded from here:

https://sofexindia.in/jkmsq/python script for xlsx to csv.py

SAMPLE OUTPUT AS IN CSV FILE:

id	name	parent_id	generation
1	BABA JUDAN		1
2	Kangali	1	2
3	Mohar	2	3
4	Sahadul	3	4
5	Quayum	4	5
6	Bilquis	5	6
7	Zafar	5	6
8	Tarique	6	7
9	Huma	6	7
10	Shadab	7	7
11	Mayera	8	8
12	Samara	9	8
13	Aadab	10	8
---	----	---	---
591	Daug Rabbani	588	7
592	Maryam	581	5
593	Chhutu Mian	580	4

Step-3:

Our next step is to organize the large dataset from a .csv file in a structured format suitable for insertion into the jkmsq_data database table (assumed name). This can be achieved in multiple ways, but one of the simplest approaches is to arrange the data as follows.

For bulk data processing, the .csv file should be formatted so that each row represents an individual record, making it easy to insert into the database.

```
<?php
// Database connection
$conn = new mysqli("localhost", "username", "password", "database_name");
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// SQL query with multiple rows
$sql = "INSERT INTO jkmsq_data (id, name, parent_id, generation) VALUES
   (1, ' BABA JUDAN ', NULL, 1),
   (2, 'Kangali', 1, 2),
   (3, ' Mohar', 2, 3),
   (4, ' Sahadul', 3, 4),
   (5, ' Quayum', 4, 5)";
// Execute query
if ($conn->query($sql) === TRUE) {
   echo "Records inserted successfully!";
} else {
   echo "Error: " . $conn->error;
}
$conn->close();
?>
```

Step-4:

We have successfully inserted the entire family tree data into a database on my server. The next step is to create a PHP/HTML script to display this data as a structured Family Tree.

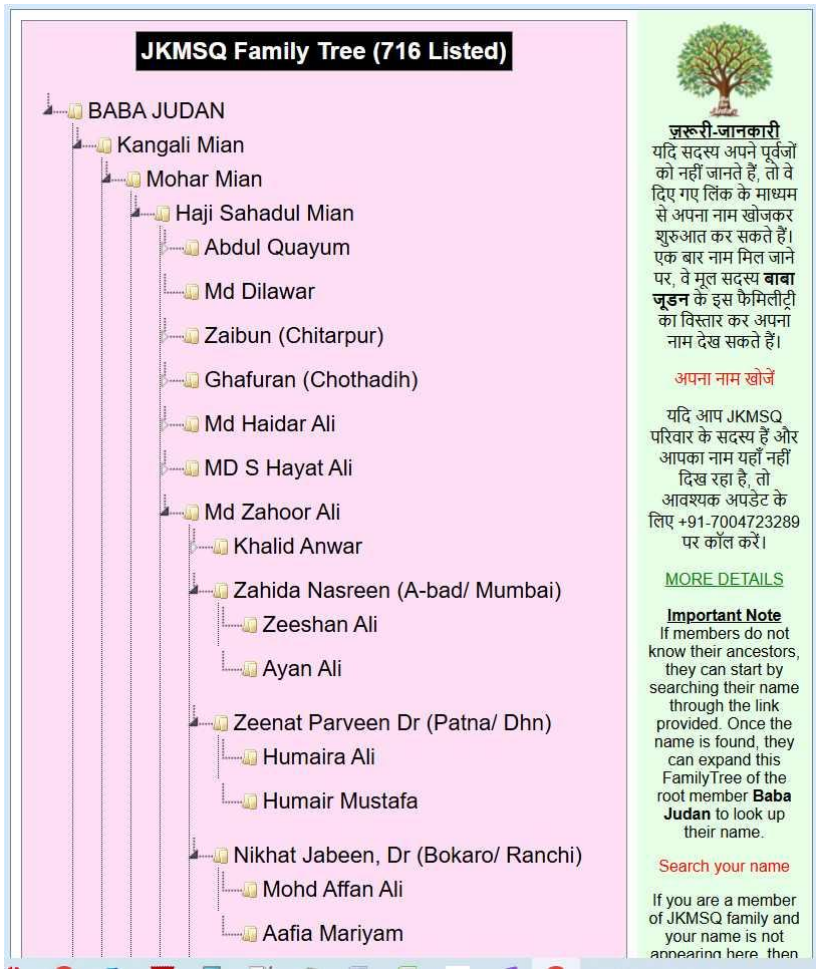
The following script, which I have personally used for my own Family Tree, can be downloaded from my website:

*https://sofexindia.in/jkmsq/php script for tree view.ph_*

Instructions:

1. Change the file extension from .ph_ to .php.
2. Open the script in a text editor and modify it according to your database settings.
3. Upload the updated script to your website.
4. Run the script in any web browser to visualize the Family Tree.

FAMILY TREE OUTPUT:

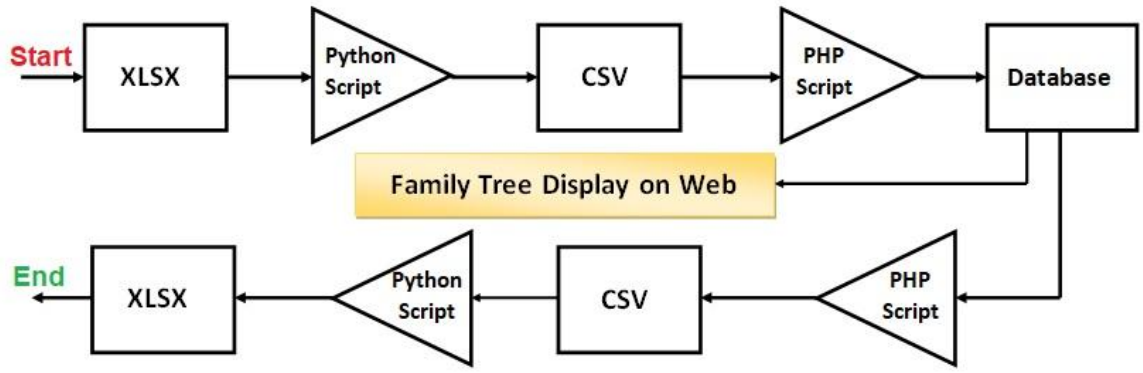


Additionally the, a dashboard can be created to extract the following outputs from the database. Here is **live links** of JKMSQ dashboard:

Know Your Forefathers	JKMSQ Tree View	View All Members
Add New Member	Update a Member	Remove a Member
View 1st Level Children	View 1st & 2nd Level Children	View 1st, 2nd & 3rd Level Children

This dashboard will enhance accessibility and provide an intuitive way to manage and analyze the family tree data.

Step-5:
Exporting database family tree data from server to create a .csv file and finally converting the .csv file to Hierarchical data in .xlsx format for easier addition of members in the database. A **flowchart** is provided below for better visualization of the process.



Users can download all the scripts of python and php by clicking the following links to use them accordingly:

Start side links: Already given and explained above.
End side links:
PHP Script for .**csv:** https://sofexindia.in/jkmsq/export_family_tree.php
Python Script for .**xlsx:** https://sofexindia.in/jkmsq/csv2xlsx.py

Important Note: Remove 1st Row and initial columns without data in the final xlsx data file, extracted from database. Cross-check parent-child relationships to confirm proper alignment in the hierarchical format.