

Python Final Report



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Course Name – **Python Project Course**

Code – **CSM216**

Section - **K23UP**

Student Group — **2**

Project Title – **VILLAGE DATA INFO HUB**

Student Name: **Vemireddy Venkatarami Reddy**

Reg No: **12307726**

Roll No: **47**

Submission Date: **22/11/2024**

Supervisor:

Aman Kumar

Acknowledgment :

I, at this moment, declare that the research work reported in the project entitled “Village data info hub”, submitted in partial fulfilment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara,

Punjab, is an original work carried out under the supervision of Aman kumar sir. The project utilizes Python programming and related technologies to achieve the stated objectives. I affirm that I have not submitted this work elsewhere for any degree or diploma. I understand that the work presented herewith adheres to Lovely Professional University’s Policy on plagiarism, intellectual property rights, and the highest standards of moral and ethical conduct. To the best of my knowledge, this project represents an authentic and honest effort, conducted in its entirety by me. I accept full responsibility for the content and results of this project.

Vemireddy Venkatarami Reddy

12307726

Table of Contents

1. Introduction
2. Objectives and Scope of the Project
3. Application Tools
4. Project Design
5. Flowchart
6. Code and output
7. Test cases
8. Conclusion
9. References

I. Introduction

The Village Data Information Hub is an application designed to help manage the data of villages, including family and individual records. It is developed to serve the needs of DEO (District Education Officers) and Village Officials, enabling them to access, manage, and update family information, member details, and village statistics. The project simplifies the data entry and retrieval processes for village officials and DEOs while maintaining security and user-specific access.

This system offers features like:

- Login functionality for both DEOs and Village Officials.
- Management of family records, including adding, updating, and viewing families.
- Tracking individual family members with details such as gender, age, qualification, occupation, etc.
- Role-based access control ensuring that Village Officials can only manage the data relevant to their specific village.

The purpose of this project is to make village-level data management easier and more efficient, using a centralized database accessible via a user-friendly graphical interface.

II. Objectives and Scope of the Project Objectives:

1. Develop a Python-based desktop application using Tkinter for a user-friendly interface.
2. Implement role-based login authentication for DEOs and Village Officials.
3. Enable DEOs to create user IDs and passwords for Village Officials.
4. Provide the ability for Village Officials to manage family and member data including adding, editing, and viewing records.
5. Store data in a MySQL database, ensuring persistence and easy retrieval of records.
6. Integrate the DEO's ability to oversee data and allow Village Officials to manage only specific data related to their respective villages.

Scope:

- The project focuses on the user interface for easy access to data and database management for secure data storage.
- It targets DEOs and Village Officials as primary users, allowing them to input and retrieve family and member records efficiently.
- The project is intended for local deployment with access to MySQL for data persistence.

III. Application Tools

Programming Language:

- Python – The main language used for the application development.

IDEs:

- PyCharm or Visual Studio Code for coding and debugging the Python application.

Libraries/Packages:

- Tkinter – For the graphical user interface (GUI).
- MySQL Connector – For connecting the Python application to a MySQL database.

Version Control:

- Git.

Other Tools:

- MySQL – For database management and storage.

IV. Project Design

The Village Data Information Hub project is structured into various components, with separate modules for handling database connections, user interfaces, and application logic. The major parts of the project are as follows:

1. Login and Authentication:

- The login page allows two types of users: DEO and Village Official.
- DEOs can create and manage user credentials for Village Officials. Upon login, DEOs can access a page for managing villages, creating accounts for officials, overseeing & also managing family data.
- Village Officials log in using their credentials to manage family records for their specific village.

2. Family Management:

- Village Officials can add a new family by entering the head of the family's name and ration card number.
- The system allows adding family members by selecting an existing family ID and entering personal details such as name, gender, date of birth, aadhar number, phone number, blood group, and occupation.
- Village Officials can view family records, including family heads and family members.

3. Database Operations:

- The project uses MySQL to store user credentials, family, and member information.
- It includes tables for DEOs, Village Officials, Families, and Family Members.

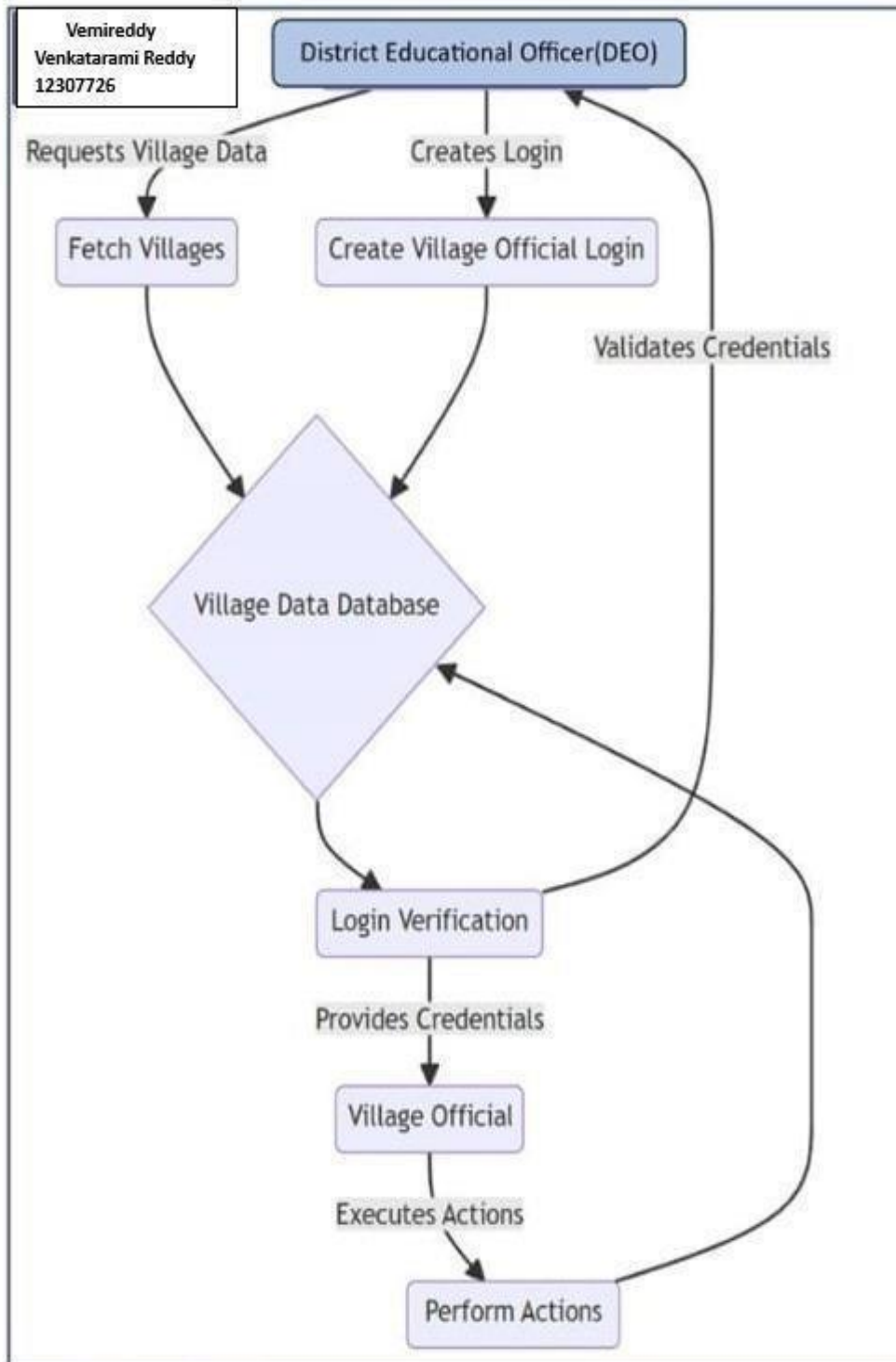
4. Navigation:

- The user interface is designed to have forward and backward navigation buttons to move between the pages for adding families, viewing families, and managing family members.

5. Security:

- Role-based access control ensures that Village Officials can only access data relevant to their villages, while DEOs have broader access to manage all villages.

V.FLOW CHART



VI.Code and output:

```
1 import tkinter as tk
2 from tkinter import messagebox, ttk
3 import mysql.connector
4 from mysql.connector import Error
5
6 # Database connection function
7 def connect_db():
8     try:
9         return mysql.connector.connect(
10             host="localhost",
11             user="root",
12             password="Srihari@143",
13             database="village_data"
14         )
15     except Error as e:
16         messagebox.showerror("Error", f"Database connection failed: {e}")
17         return None
18
19 # Login Verification
20 def verify_login(user_type, village_name, username, password, root):
21     db = connect_db()
22     if not db:
23         return
24     cursor = db.cursor()
25
26     try:
27         if user_type == 'DEO':
28             query = "SELECT * FROM deo WHERE username = %s AND password = %s"
29             cursor.execute(query, (username, password))
30             result = cursor.fetchone()
31             if result:
32                 root.open_deo_homepage() # Open DEO homepage
33             else:
34                 messagebox.showerror("Error", "Invalid DEO Credentials")
35         elif user_type == 'Village Official':
36             query = "SELECT * FROM village_officials WHERE village_name = %s AND username = %s AND password = %s"
37             cursor.execute(query, (village_name, username, password))
38             result = cursor.fetchone()
39             if result:
40                 root.user_village = village_name # Store village name
41                 root.open_official_homepage() # Open Village Official homepage
42             else:
43                 messagebox.showerror("Error", "Invalid Village Official Credentials")
44     except Error as e:
45         messagebox.showerror("Error", f"Database query failed: {e}")
46     finally:
47         db.close()
48
49 # Main Application Class
50 # Main Application Class
51 class VillageDataInfoHub(tk.Tk):
52     def __init__(self):
53         super().__init__()
54         self.title("Village Data Info Hub")
55         self.geometry("600x700")
56         self.user_village = None # Store the logged-in village (if applicable)
57         self.create_login_page()
58
59     def create_login_page(self):
60         self.clear_window()
61
62         login_frame = tk.Frame(self)
63         login_frame.pack(fill="both", expand=True)
64
65         tk.Label(login_frame, text="Select User Type:").pack(pady=5)
66         user_type = ttk.Combobox(login_frame, values=["DEO", "Village Official"])
67         user_type.pack(pady=5)
```



```
File Edit Selection View Go Run Terminal Help
village dat info hub.py 2 X
C:\Users\venke> OneDrive\ Desktop> New folder> python> village dat info hub.py > ...
51 class VillageDataInfoHub(tk.Tk):
60 def create_login_page(self):
69
70     self.village_dropdown = ttk.Combobox(login_frame, values=[])
71     self.village_dropdown.pack(pady=5)
72     self.village_dropdown.set("Select Village")
73     self.village_dropdown.pack_forget()
74
75     user_type.bind("<<ComboboxSelected>>", lambda e: self.on_user_type_select(user_type.get(), login_frame))
76
77     tk.Label(login_frame, text="Username:").pack(pady=5)
78     username_entry = tk.Entry(login_frame)
79     username_entry.pack(pady=5)
80
81     tk.Label(login_frame, text="Password:").pack(pady=5)
82     password_entry = tk.Entry(login_frame, show="*")
83     password_entry.pack(pady=5)
84
85     login_button = tk.Button(
86         login_frame,
87         text="login",
88         command=lambda: verify_login(
89             user_type.get(), self.village_dropdown.get(), username_entry.get(), password_entry.get(), self
90         )
91     )
92     login_button.pack(pady=20)
93
94 def on_user_type_select(self, user_type, login_frame):
95     if user_type == 'Village Official':
96         self.village_dropdown.pack(pady=5)
97         self.village_dropdown['values'] = self.get_villages()
98     else:
99         self.village_dropdown.pack_forget()
100
101 def open_deo_homepage(self):
102     self.clear_window()
103
104     tk.Label(self, text="DEO Homepage", font=("Arial", 20)).pack(pady=20)
105
106     tk.Button(
107         self,
108         text="Create Village Official login",
109         command=self.create_village_official_login
110     ).pack(pady=10)
111
112 def create_village_official_login(self):
113     self.clear_window()
114
115     tk.Label(self, text="Create Village Official login", font=("Arial", 16)).pack(pady=10)
116
117     tk.Label(self, text="Select Village:").pack(pady=5)
118     village_dropdown = ttk.Combobox(self, values=self.get_villages())
119     village_dropdown.pack(pady=5)
120
121     tk.Label(self, text="Username:").pack(pady=5)
122     username_entry = tk.Entry(self)
123     username_entry.pack(pady=5)
124
125     tk.Label(self, text="Password:").pack(pady=5)
126     password_entry = tk.Entry(self, show="*")
127     password_entry.pack(pady=5)
128
129     submit_button = tk.Button(
130         self,
131         text="Submit",
132         command=lambda: self.save_village_official(
133             village_dropdown.get(), username_entry.get(), password_entry.get()
134         )
135     )
136     submit_button.pack(pady=10)
137
138 def save_village_official(self, village, username, password):
139     # Save logic here
140
141 def get_villages(self):
142     # Get villages logic here
143
144 def clear_window(self):
145     # Clear window logic here
146
147 def __del__(self):
148     # Destructor logic here
149
150 if __name__ == '__main__':
151     info_hub = VillageDataInfoHub()
152     info_hub.create_login_page()
153     info_hub.open_deo_homepage()
154     info_hub.mainloop()
155
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
```

```
File Edit Selection View Go Run Terminal Help
village dat info hub.py 2 X
C:\Users\venke> OneDrive\ Desktop> New folder> python> village dat info hub.py > ...
51 class VillageDataInfoHub(tk.Tk):
101 def open_deo_homepage(self):
102     self.clear_window()
103
104     tk.Label(self, text="DEO Homepage", font=("Arial", 20)).pack(pady=20)
105
106     tk.Button(
107         self,
108         text="Create Village Official login",
109         command=self.create_village_official_login
110     ).pack(pady=10)
111
112 def create_village_official_login(self):
113     self.clear_window()
114
115     tk.Label(self, text="Create Village Official login", font=("Arial", 16)).pack(pady=10)
116
117     tk.Label(self, text="Select Village:").pack(pady=5)
118     village_dropdown = ttk.Combobox(self, values=self.get_villages())
119     village_dropdown.pack(pady=5)
120
121     tk.Label(self, text="Username:").pack(pady=5)
122     username_entry = tk.Entry(self)
123     username_entry.pack(pady=5)
124
125     tk.Label(self, text="Password:").pack(pady=5)
126     password_entry = tk.Entry(self, show="*")
127     password_entry.pack(pady=5)
128
129     submit_button = tk.Button(
130         self,
131         text="Submit",
132         command=lambda: self.save_village_official(
133             village_dropdown.get(), username_entry.get(), password_entry.get()
134         )
135     )
136     submit_button.pack(pady=10)
137
138 def save_village_official(self, village, username, password):
139     # Save logic here
140
141 def get_villages(self):
142     # Get villages logic here
143
144 def clear_window(self):
145     # Clear window logic here
146
147 def __del__(self):
148     # Destructor logic here
149
150 if __name__ == '__main__':
151     info_hub = VillageDataInfoHub()
152     info_hub.create_login_page()
153     info_hub.open_deo_homepage()
154     info_hub.mainloop()
155
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
```

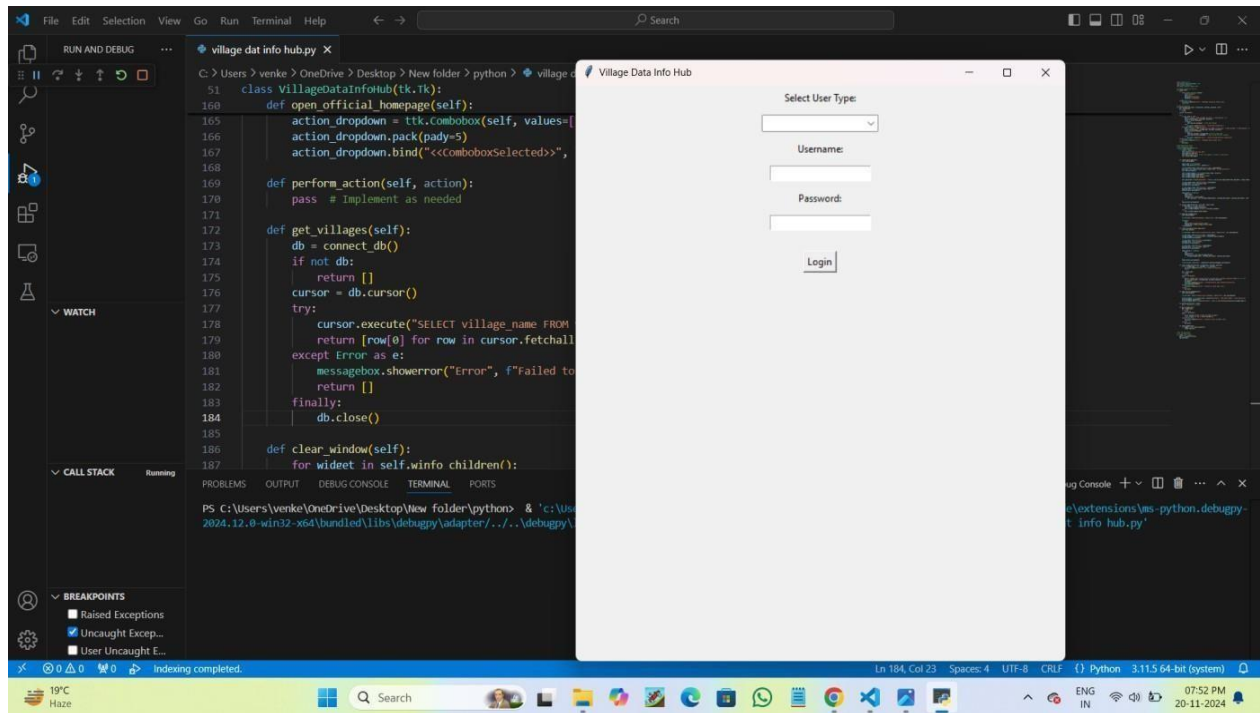
```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

village dat info hub.py 2 X
C:\Users> venke > OneDrive > Desktop > New folder > python > village dat info hub.py > ...
51 class VillageDataInfoHub(tk.Tk):
112     def create_village_official_login(self):
135     )
136     submit_button.pack(pady=20)
137
138     tk.Button(self, text="Back", command=self.open_deo_homepage).pack(pady=10)
139
140     def save_village_official(self, village_name, username, password):
141     if not village_name or not username or not password:
142         messagebox.showerror("Error", "All fields are required!")
143         return
144
145     db = connect_db()
146     if not db:
147         return
148     cursor = db.cursor()
149     try:
150         query = "INSERT INTO village_officials (village_name, username, password) VALUES (%s, %s, %s)"
151         cursor.execute(query, (village_name, username, password))
152         db.commit()
153         messagebox.showinfo("Success", "Village Official Login Created Successfully")
154         self.open_deo_homepage()
155     except Error as e:
156         messagebox.showerror("Error", f"Failed to create login: {e}")
157     finally:
158         db.close()
159
160     def open_official_homepage(self):
161     self.clear_window()
162
163     tk.Label(self, text="Village Official Homepage", font=("Arial", 20)).pack(pady=20)
164
165     action_dropdown = ttk.Combobox(self, values=["Add Family", "Add Family Member", "View Families"])
166     action_dropdown.pack(pady=5)
167     action_dropdown.bind("<<<ComboboxSelected>>>", lambda e: self.perform_action(action_dropdown.get()))
168
169     def perform_action(self, action):
170     pass # Implement as needed
171
172     def get_villages(self):
173     db = connect_db()
174     if not db:
175         return []
176     cursor = db.cursor()
177     try:
178         cursor.execute("SELECT village_name FROM villages")
179         return [row[0] for row in cursor.fetchall()]
180     except Error as e:
181         messagebox.showerror("Error", f"Failed to fetch villages: {e}")
182         return []
183     finally:
184         db.close()
185
186     def clear_window(self):
187     for widget in self.winfo_children():
188         widget.destroy()
189
190     # Run the application
191     # Run the application
192     if __name__ == "__main__":
193         app = VillageDataInfoHub()
194         app.mainloop()
195
196
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
19°C Haze
```

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

village dat info hub.py 2 X
C:\Users> venke > OneDrive > Desktop > New folder > python > village dat info hub.py > ...
51 class VillageDataInfoHub(tk.Tk):
160     def open_official_homepage(self):
165     action_dropdown = ttk.Combobox(self, values=["Add Family", "Add Family Member", "View Families"])
166     action_dropdown.pack(pady=5)
167     action_dropdown.bind("<<<ComboboxSelected>>>", lambda e: self.perform_action(action_dropdown.get()))
168
169     def perform_action(self, action):
170     pass # Implement as needed
171
172     def get_villages(self):
173     db = connect_db()
174     if not db:
175         return []
176     cursor = db.cursor()
177     try:
178         cursor.execute("SELECT village_name FROM villages")
179         return [row[0] for row in cursor.fetchall()]
180     except Error as e:
181         messagebox.showerror("Error", f"Failed to fetch villages: {e}")
182         return []
183     finally:
184         db.close()
185
186     def clear_window(self):
187     for widget in self.winfo_children():
188         widget.destroy()
189
190     # Run the application
191     # Run the application
192     if __name__ == "__main__":
193         app = VillageDataInfoHub()
194         app.mainloop()
195
196
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
19°C Haze
```

OUTPUT:



Select User Type:

Village Official

Username:

Durgaram

Password:

Login

Durgaram

Select User Type:

Village Official

Username:

Karlai

Password:

Login

Karlai

DEO Homepage

Create Village Official Login

Create Village Official Login

Select Village:

Konapuram

Username:

Konapuram

Password:

Submit

Back

Create Village Official Login

Select Village:

Konapuram

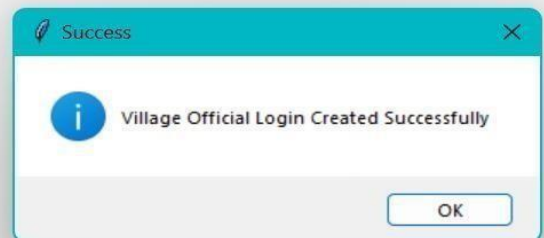
Username:

Konapuram

Password:

Submit

Back



Add Family Member

Family ID (Ration Card No):

Member Name:

Gender:

Date of Birth:

Aadhar Card Number:

Phone Number:

Qualification:

Blood Group:

Occupation:

Add Family

Head of Family Name:

Ration Card Number:

VII. TEST CASES:

Test Case ID	Scenario	Input	Expected Output		Status
TC001	Valid DEO Login	User Type: DEO, Username: valid_username(deo), Password: valid_password(deo@00)	DEO Homepage is displayed.	VALID	TESTED
TC002	Invalid DEO Login	User Type: DEO, Username: invalid_username(DEO), Password: invalid_password(DEO@)	Error message: "Invalid DEO Credentials".	INVALID	TESTED
TC003	Valid Village Official Login	User Type: Village Official, Village: valid_village, Username: valid_username, Password: valid_password	Village Official Homepage is displayed.	VAILD	TESTED
TC004	Invalid Village Official Login	User Type: Village Official, Village: valid_village, Username: invalid_username, Password: invalid_password	Error message: "Invalid Village Official Credentials".	INVALID	TESTED
TC005	DEO: Create Village Official with valid input	Village: valid_village, Username: valid_username, Password: valid_password	Message: "Village Official Login Created Successfully", Redirects to DEO Homepage.	VAILD	TESTED
TC006	DEO: Create Village Official with missing input	Village: valid_village, Username: "", Password: valid_password	Error message: "All fields are required!".	INVAILD	TESTED
TC007	Village Official Homepage action selection	Action: Add Family	Corresponding functionality (e.g., adding a family) is triggered.	VAILD	TESTED
TC008	DEO: Select Village dropdown with populated database	Database contains villages: VillageA, VillageB	Dropdown is populated with options "VillageA", "VillageB".	VAILD	TESTED

VIII.Conclusion:

This project has really been faithful and informative. It has made us learn and understand the many trivial concepts of python language. As we have used python Tkinter as a GUI it provides various controls such as buttons, labels, and tabs and text box to build a user-friendly application. Finally, it has taught us a valuable lifelong lesson about the improvements and working and interacting in a group. And also, it helps us to learn how to code in python and we are able to learn more about different module present in python. It also proved beneficial for us because we were able to design GUI inpython.

IX. References:

1. Python Documentation:

- o Python Official Documentation:

- <https://docs.python.org/>

- o Used for understanding language syntax, libraries, and error-handling mechanisms.

2. Tkinter Documentation:

- o **Python Tkinter GUI Programming:**

- <https://docs.python.org/3/library/tkinter.html>

3. Numeral Systems

- o **Wikipedia: Numeral Systems Overview:**

- https://en.wikipedia.org/wiki/Numeral_system

- o Provided foundational knowledge about binary, decimal, hexadecimal, and other numeral systems.

These references collectively guided the development, testing, and documentation of the Base Conversion System project.