RAJKIYA PRATIBHA VIKAS VIDYALAYA, SECTOR-10 DWARKA

CENSUS MANAGEMENT SYSTEM

A Project Report Submitted
in Partial Fulfillment of the Requirements.

Academic Year: 2022-2023

In
COMPUTER SCIENCE (083)

By:

1. **AMAN KUMAR - XII B - ROLL NO: 11**
2. **ANSHUMAN - XII B - ROLL NO: 21**
3. **CHIRAG - XII B - ROLL NO: 26**

Acknowledgements

We would like to thank Dr. Atul Kumar, Principal Rajkiya Pratibha Vikas Vidyalaya,Sector 10 Dwarka. We are deeply indebted to our mentor Mr. OM Prakash Rai. We further thank all the staff members of Rajkiya Pratibha Vikas Vidyalaya,Sector-10 Dwarka.

We also express our deepest gratitude to our parents.

Finally, we would like to wind up by paying our heartfelt thanks to all our near and dear ones.

1. AMAN KUMAR - XII B - ROLL NO: 11
2. ANSHUMAN - XII B - ROLL NO: 21
3. CHIRAG - XII B - ROLL NO: 26

Contents

1. Introduction of the Project.

2. System Requirements of the Project.

3. Python Coding. 4.Output of the Project. 5.References.

Introduction of the Project

We the students of CLASS XII B of Rajkiya Pratibha Vikas Vidyalaya,Sector 10 Dwarka have been assigned the work of CENSUS MANAGEMENT SYSTEM.

We are so glad that this work has been assigned to us, yet we haven't done this work before .SH. OM PRAKASH RAI, our subject teacher has also helped us a lot to complete this project. We feel so blessed that we have learnt all this work with the help of our sir. We are also thankful to our respected principal DR.ATUL KUMAR for providing us various facilities to complete this project.

As we are the students of CLASS XII B and we haven't done this type of project before, we have performed all that which we have learnt from our CBSE PROGRAMMING .Hence, we know that this programming would be further done on a big platform. Since we have started this programming from SEPTEMBER month ,we believe that this programming would further help us a lot in our future .
We are also thankful to our group mates for cooperating with each other while performing this task. We have also polished the skills of group activity.

PROCESS

FIRSTLY, we have done the planning in a paperwork regarding what have to do on the assigned project HOTEL MANAGEMENT SYSTEM.

SECONDLY, we discussed our planning with our subject teacher and then he provided us the right path to perform the work.

NEXT, we started our project on the foot paths of our subject teacher.

THEN, we started our coding, coding took around 2 and half months for completion.

NEXT, we analyzed the mistakes done and then we corrected them. THEN, we prepared the project format as shown above.

System Requirements of the Project

Recommended System Requirements

Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz. Disk space: 2 to 4 GB.
Operating systems: Windows® 10, MACOS, and UBUNTU. Python Versions: 3.9 or Higher.
External Python Libraries: Custom Tkinter

Minimum System Requirements

Processors: Intel Atom® processor or Intel® Core™ i3 processor.

Disk space: 1 GB.

Operating systems: Windows 7 or later, MACOS, and UBUNTU. Python Versions: 3.9 or Higher.
External Python Libraries: Custom Tkinter

Prerequisites before installing MySQL Connector Python

You need root or administrator privileges to perform the installation process.

Python must be installed on your machine.

Note: – MySQL Connector Python requires python to be in the system's PATH. Installation fails if it doesn't find Python.

On Windows, If Python doesn't exist in the system's PATH, please manually add the directory containing python.exe yourself.

SAINIK SCHOOL AMARAVATHINAGAR

CERTIFICATE

This is to certify that Cadet **U.SHARAN**Roll No: 6012 has successfully completed the project Work entitled **UPSC REGISTRATION INFORMATION SYSTEM** in the subject Computer Science (083) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in Sainik School Amaravathinagar on______________.

(PM Jigajinni)
PGT Comp Sci
Master IC

Examiner:

Name: _______________

Signature:

TABLE OF CONTENTS [T O C]

SER	DESCRIPTION	PAGE NO
01	ACKNOWLEDGEMENT	**04**
02	INTRODUCTION	**05**
03	OBJECTIVES OF THE PROJECT	**05**
04	PROPOSED SYSTEM	**06**
05	SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)	**07**
06	PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE	**08**
07	FLOW CHART	**15**
08	SOURCE CODE	**16**
09	OUTPUT	**19**
10	TESTING	**20**
11	HARDWARE AND SOFTWARE REQUIREMENTS	**23**
12	BIBLIOGRAPHY	**24**

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws,

I express my deep sense of gratitude to the luminary The Principal, Rajkiya Pratibha Vikas Vidhyalaya who has been continuously motivating and extending their helping hand to us.

I express my sincere thanks to the academician The Vice Principal, Rajkiya Pratibha Vikas Vidhyalaya, for constant encouragement and the guidance provided during this project

I am overwhelmed to express my thanks to The Administrative Officer for providing me an infrastructure and moral support while carrying out this project in the school.

My sincere thanks to **Mr. OM Prakash Rai**, Teacher In-charge, A guide, Mentor all the above a friend,who critically reviewed my project and helpedin solving each and every problem, occurred during implementation of the project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

PROJECT ON CENSUS MANAGEMENT SYSTEM

INTRODUCTION

This programme helps in data collection of the people in country. Census Management system project to automate the counting process of people.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.

5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

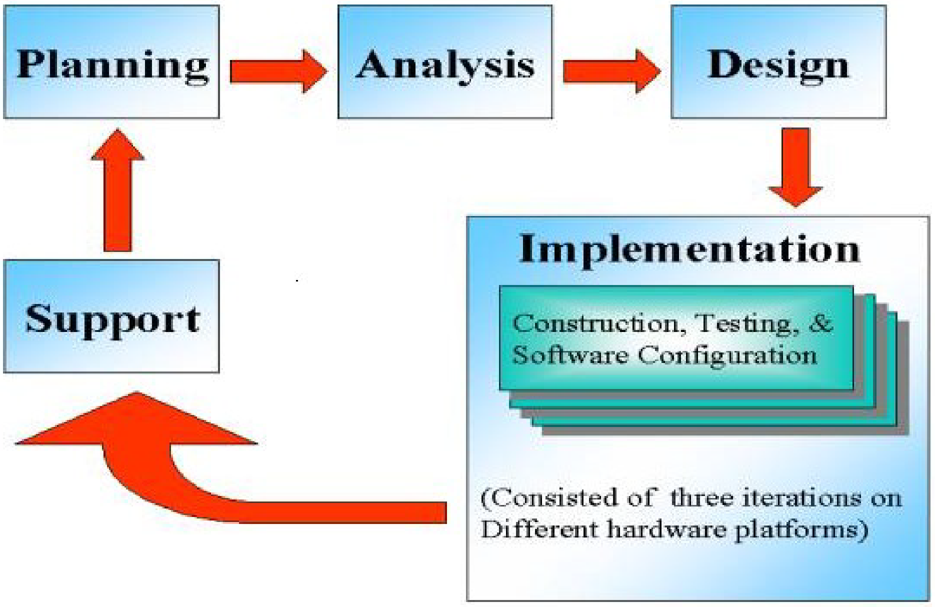
PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying **"to err is human"** no longer valid, it's out-dated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a muchsophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomizationvarious organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially hadto maintain a lot of ledgers and a lot of paperwork has to be done but now software producton this organization has made their work fasterand easier. Now only this software has to beloaded on the computer and work can be done.

This prevents a lot of time and money. Thework becomes fully automated and any information regarding the organization can beobtained by clicking the button. Moreover, nowit’s an age of computers of and automatingsuch an organization gives the better look.

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)


Planning
Analysis
Design
Implementation
Construction, Testing, & Software Configuration
(Consisted of three iterations on Different hardware platforms)
Support

The systems development life cycle is a projectmanagement technique that divides complexprojects into smaller, more easily managedsegments or phases. Segmenting projectsallows managers to verify the successfulcompletion of project phases before allocatingresources to subsequent phases.

Software developmentprojects typically include initiation, planning,design, development, testing, implementation,and maintenance phases. However, the phasesmay be divided differently depending on theorganization involved.

For example, initialproject activities might be designated asrequest, requirements-definition, and planningphases, or initiation, concept-development,and planning phases. End users of the systemunder development should be involved inreviewing the output of each phase to ensurethe system is being built to deliver the neededfunctionality.

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

INITIATION PHASE

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

The purpose of the Initiation Phase is to:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution?
- Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization.

- Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin
  the project.

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal’s purpose, identify expected benefits, and explain how the proposed system supports one of the organization’s business strategies. The business case should also identify alternative solutions and detail as many informational, functional, and network requirements as possible.

SYSTEM CONCEPT DEVELOPMENT PHASE

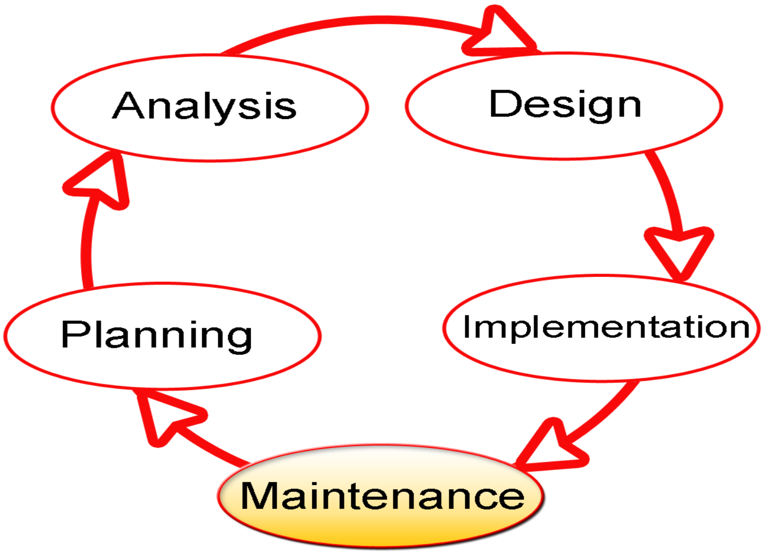
The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

The purpose of the System Concept Development Phase is to:

- Determine the feasibility and appropriateness of the alternatives.
- Identify system interfaces.
- Identify basic functional and data requirements to satisfy the business need.
- Establish system boundaries; identify goals, objectives, critical success factors, and performance measures.
- Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements
- Assess project risks

- Identify and initiate risk mitigation actions, and Develop high-level technical architecture, process models, data models, and a concept of operations. This phase explores potential technical solutions within the context of the business need.
- It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, onetime deployment.
- Construction of executable prototypes is encouraged to evaluate technology to support the business process. The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process.
- The ITPR must be approved by the State CIO before the project can move forward.

PICTORIAL REPRESENTATION OF SDLC:



PLANNING PHASE

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, isnecessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specificactivities and resources required to complete a project.

A critical part of a project manager'sjob is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and networkrequirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

REQUIREMENTSANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
- Develop detailed data and process models (system inputs, outputs, and the process.
- Develop the test and evaluation requirements that will be used to determine acceptable system performance**.**

DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs areconstructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.
- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.
- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.
- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.
- This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements. Concurrent with the development of the system design, the Agency Project Manager begins development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system.

INTEGRATION AND TEST PHASE

- Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assesses the system security and issue a security certification and accreditation prior to installation/implementation.

Multiple levels of testing are performed, including:

- Testing at the development facility by the contractor and possibly supported by end users

- Testing as a deployed system with end users working together with contract personnel

- Operational testing by the end user alone performing all functions. Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

OPERATIONS AND MAINTENANCE PHASE

The system operation is on-going. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization's needs. When modifications or changes are identified, the system may re-enter the planning phase.

The purpose of this phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

FLOW CHART FOR LOGIN TABLE

```
#backend
import csv
from tkinter import simpledialog
import mysql.connector
from tkinter import *
from tkinter import messagebox
import customtkinter

customtkinter.set_appearance_mode("System")
customtkinter.set_default_color_theme("blue")

root=customtkinter.CTk()
root.title("Census Record System")
root.geometry("1000x1200+10+10")
root.config(bg="#bca0dc")

def table1():

    def c1():
        tu = Label(root,text="Enter user :
",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        tu.pack(padx=10,pady=10)
        ent1 = Entry(root,bd=2,width=40,font=(18))
        ent1.pack(padx=10,pady=10)
        tpass = Label(root,text="Enter password :
",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        tpass.pack(padx=10,pady=10)
        ent2 = Entry(root,bd=2,width=40,font=(18),show = "*")
        ent2.pack(padx=10,pady=10)
        tdb = Label(root,text="Database name :
",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        tdb.pack(padx=10,pady=10)
        ent3 = Entry(root,bd=2,width=40,font=(18),)
        ent3.pack(padx=10,pady=10)

        def c1del():
            tu.after(1,tu.destroy)
            tpass.after(1,tpass.destroy)
            tdb.after(1,tdb.destroy)
            ent1.after(1,ent1.destroy)
            ent2.after(1,ent2.destroy)
            ent3.after(1,ent3.destroy)
            btn5.after(1,btn5.destroy)

        def dbsetup():
            global entT1,entT2,entT3
            entT1 = ent1.get()
            entT2 = ent2.get()
            entT3 = ent3.get()

            mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2)
            mycursor = mydb.cursor()
            mycursor.execute("CREATE DATABASE {}".format(entT3))
            mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("create table if not exists census_info(First_Name
varchar(64) not null,Last_Name varchar(64) null,Date_Of_Birth date not
null,Email_ID varchar(255) not null unique,Aadhar_Number varchar(16) not null
primary key,Father_Name varchar(64) not null,Mother_Name varchar(64)  not
null,City_And_State varchar(100) not null);")

        btn5 = Button(root,text="Save.",activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32),command=lambda:
[table2(),c1del(),dbsetup()])
        btn5.pack(padx=10,pady=10)

    def c2():
        tu = Label(root,text="Enter user :
",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        tu.pack(padx=10,pady=10)
        ent1 = Entry(root,bd=2,width=40,font=(18))
        ent1.pack(padx=10,pady=10)
        tpass = Label(root,text="Enter password :
",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        tpass.pack(padx=10,pady=10)
        ent2 = Entry(root,bd=2,width=40,font=(18),show = "*")
        ent2.pack(padx=10,pady=10)
        tdb = Label(root,text="Database name :
",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        tdb.pack(padx=10,pady=10)
        ent3 = Entry(root,bd=2,width=40,font=(18))
        ent3.pack(padx=10,pady=10)

        def vals():
            global entT1,entT2,entT3
            entT1 = ent1.get()
            entT2 = ent2.get()
            entT3 = ent3.get()

        def c2del():
            tu.after(1,tu.destroy)
            tpass.after(1,tpass.destroy)
            tdb.after(1,tdb.destroy)
            ent1.after(1,ent1.destroy)
            ent2.after(1,ent2.destroy)
            ent3.after(1,ent3.destroy)
            btn4.after(1,btn4.destroy)

        btn4 = Button(root,text="Save.",activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32),command=lambda:
[table2(),c2del(),vals()])
        btn4.pack(padx=10,pady=10)

    def btncdel():
        btnc1.after(1,btnc1.destroy)
        btnc2.after(1,btnc2.destroy)

    btnc1 = Button(root,text="Automatic Configuration (An Automatic Configuration
from Scratch.)",command = lambda:[c1(),btncdel()],activebackground="#800080",bg =
"#7c5295",bd=2,width=60,height=2,font=(18))
    btnc1.pack(pady=10)

    btnc2 = Button(root,text="Manual Configuration (For Pre-Made Database Only,You
can setup table later on in pre-made database.)",command = lambda:
[c2(),btncdel()],activebackground="#800080",bg =
"#7c5295",bd=2,width=86,height=2,font=(18))
```

```
    btnc2.pack(pady=10)




def btn1del():
    btn1.after(1,btn1.destroy)

btn1 = Button(root,text="Click me to Start Setup",command = lambda:
[table1(),btn1del()],activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32))
btn1.pack(pady=10)

def newwin():
    root=customtkinter.CTk()
    root.title("Test Win.")
    root.geometry("100x200+10+10")
    root.config(bg="#bca0dc")
    root.mainloop()

#setup

def dcensus(): #DELETE A RECORD FROM DATABASE #COMPLETE

    a = 0
    mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
    mycursor = mydb.cursor()
    dchoice = simpledialog.askstring("Delete A Record from Database","Enter Aadhar
Number to Delete the Record(XXXX-XXXX-XXXX)")
    if len(dchoice) == 0:
        messagebox.showerror('Error', 'Error: No Aadhar Number provided.')
    else:
        mycursor.execute('SELECT * FROM census_info')
        myresult = mycursor.fetchall()
        for i in myresult:
            if dchoice == i[4]:
                a = a+1
                mycursor.execute("DELETE FROM census_info WHERE
Aadhar_Number='{}'".format(dchoice))
                mydb.commit()
                messagebox.showinfo("Successful",  "Deleted record from the
Database.")
                break
        if a == 0:
            messagebox.showerror("Error",  "No Entries exists with this Aadhar
number in database.")

def lcensus(): #SHOW ALL THE RECORDS IN DATABASE #COMPLETE
    mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
    mycursor = mydb.cursor()
    mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
    myresult = mycursor.fetchall()
    if len(myresult) == 0:
        messagebox.showerror("Error","No Records to show in database.")
    else:
        def swin():
            mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            a = 1
            mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
            myresult = mycursor.fetchall()
            root=customtkinter.CTk()
            root.title("Records in Database.")
            root.geometry("400x600")
            root.config(bg="#D586FF")
            scrollbar = Scrollbar(root)
            scrollbar.pack( side = RIGHT, fill = Y )
            listbox = Listbox(root, yscrollcommand =
scrollbar.set,bg="#D586FF",font="32",height = "33")
            listbox.pack(fill = BOTH)
            for i in myresult:
                listbox.insert(END,"Record Number : "+str(a))
                listbox.insert(END,"First Name : "+i[0])
                listbox.insert(END,"Last Name : "+i[1])
                listbox.insert(END,"Date Of Birth : "+str(i[2]))
                listbox.insert(END,"Email ID : "+i[3])
                listbox.insert(END,"Aadhar Number : "+i[4])
                listbox.insert(END,"Father's Name : "+i[5])
                listbox.insert(END,"Mother's Name : "+i[6])
                listbox.insert(END,"City & State : "+i[7])

                listbox.pack()
                scrollbar.config( command = listbox.yview )

                a = a+1
            root.resizable(False,False)
            root.mainloop()
        swin()

def scensus():#SEARCH FOR A RECORD IN DATABASE #COMPLETE
    mydb =
mysql.connector.connect(host="localhost",user="root",passwd="root",database="census
_info")
    mycursor = mydb.cursor()
    a = 0
    schoice = simpledialog.askstring("Search for a record","Enter Aadhar number to
search record+")
    mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
    myresult = mycursor.fetchall()
    for i in myresult:
        if schoice == i[4]:
            a = 1
            b = i
            break
    if a == 1:
        def show():
            root=customtkinter.CTk()
            root.title("Records in Database.")
            root.geometry("400x600")
            root.config(bg="#D586FF")
            scrollbar = Scrollbar(root)
            scrollbar.pack( side = RIGHT, fill = Y )
            listbox = Listbox(root, yscrollcommand =
scrollbar.set,bg="#D586FF",font="32",height="33")
            listbox.pack(fill = BOTH)
            listbox.insert(END,"First Name : "+b[0])
            listbox.insert(END,"Last Name : "+b[1])
            listbox.insert(END,"Date Of Birth : "+str(b[2]))
            listbox.insert(END,"Email ID : "+b[3])
            listbox.insert(END,"Aadhar Number : "+b[4])
            listbox.insert(END,"Father's Name : "+b[5])
            listbox.insert(END,"Mother's Name : "+b[6])
            listbox.insert(END,"City & State : "+b[7])

            root.resizable(False,False)
            root.mainloop()
        show()
    else:
        messagebox.showerror("Error","No Records exists with this Aadhar number in
database.")

def icensus(): #ADD A RECORD TO THE DATABASE #COMPLETE

            mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()

            mycursor.execute('SELECT Aadhar_Number,Date_Of_Birth FROM census_info
ORDER BY First_Name')
            myresult = mycursor.fetchall()

            def addvals():
                x = []
                First_Name = FNameEnt.get()
                Last_Name = LNameEnt.get()
                Date_Of_Birth = DOBEnt.get()
                Email = EIDEnt.get()
                Aadhar_Number = ANumberEnt.get()
                Father_Name = FatherNameEnt.get()
                Mother_Name = MNameEnt.get()
                City_And_State = CASEnt.get()

                a = 0
                if len(myresult) == 0:
                    y = (First_Name, Last_Name,
Date_Of_Birth,Email,Aadhar_Number,Father_Name,Mother_Name,City_And_State)
                    x.append(y)
                else:
                    for i in myresult:
                        if Aadhar_Number == i[0]:
                            a = a + 1
                            break
                        else:
                            y = (First_Name, Last_Name,
Date_Of_Birth,Email,Aadhar_Number,Father_Name,Mother_Name,City_And_State)
                            x.append(y)

                if a == 1:
                    messagebox.showerror("Error","Same Aadhar Number address
already exists in database,Try Again.")
                else:
                    sqlform = "INSERT INTO
census_info(First_Name,Last_Name,Date_Of_Birth,Email_ID,Aadhar_Number,Father_Name,M
other_Name,City_And_State) VALUES (%s, %s, %s, %s,%s, %s, %s, %s)"
```

```
                    mycursor.executemany(sqlform,x)
                    mydb.commit()
                    messagebox.showinfo("Successful","Updated Entries in 
Database.")

            def addi():
                global 
FNameEnt,LNameEnt,DOBEnt,EIDEnt,ANumberEnt,FatherNameEnt,MNameEnt,CASEnt
                root=Tk()
                root.title("Records in Database.")
                root.geometry("400x1000")
                root.config(bg="#D586FF")



                FName = Label(root,text="Enter your First 
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                FName.pack(padx=10,pady=10)
                FNameEnt = Entry(root,bd=2,width=40,font=(18))
                FNameEnt.pack(padx=10,pady=10)
                LName = Label(root,text="Enter your Last 
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                LName.pack(padx=10,pady=10)
                LNameEnt = Entry(root,bd=2,width=40,font=(18))
                LNameEnt.pack(padx=10,pady=10)
                DOB = Label(root,text="Enter your Date Of Birth(YYYY-MM-
DD)",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                DOB.pack(padx=10,pady=10)
                DOBEnt = Entry(root,bd=2,width=40,font=(18))
                DOBEnt.pack(padx=10,pady=10)
                EID = Label(root,text="Enter your Email 
ID",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                EID.pack(padx=10,pady=10)
                EIDEnt = Entry(root,bd=2,width=40,font=(18))
                EIDEnt.pack(padx=10,pady=10)
                ANumber = Label(root,text="Enter your Aadhar 
Number",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                ANumber.pack(padx=10,pady=10)
                ANumberEnt = Entry(root,bd=2,width=40,font=(18))
                ANumberEnt.pack(padx=10,pady=10)
                FatherName = Label(root,text="Enter your Father's 
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                FatherName.pack(padx=10,pady=10)
                FatherNameEnt = Entry(root,bd=2,width=40,font=(18))
                FatherNameEnt.pack(padx=10,pady=10)
                MName = Label(root,text="Enter your Mother's 
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                MName.pack(padx=10,pady=10)
                MNameEnt = Entry(root,bd=2,width=40,font=(18))
                MNameEnt.pack(padx=10,pady=10)
                CAS = Label(root,text="Enter your City and 
State(City/State)",bd=2,width=40,height=2,font=(18),bg="#3c1361")
                CAS.pack(padx=10,pady=10)
                CASEnt = Entry(root,bd=2,width=40,font=(18))
                CASEnt.pack(padx=10,pady=10)

                BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = addvals)
                BTN.pack(padx=10,pady=10)
                root.resizable(False,False)
                root.mainloop()

            addi()

def rcensus(file1): #READ A CSV AND APPEND ITS ALL ENTRY TO DATABASE #COMPLETE
    mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
    mycursor = mydb.cursor()
    y=[]
    x = t = d = 0
    mycursor.execute('SELECT Aadhar_Number,Email_ID FROM census_info ORDER BY 
First_Name')
    myresult = mycursor.fetchall()
    with open(file1, "r") as f:
        reader = csv.reader(f)
        trash = f.readline()
        for j in reader:
            if len(j) == 0:
                a = list(j)
            else:
                y.append(j)
        if len(y) == 0:
            d = d + 1

        if d == 0:
            if len(myresult) > len(y):

                for z in myresult:
                    for f in y:
                        if z[0] == f[4] or z[1] == f[3]:
                            def error1():
                                messagebox.showerror("Error",  "Database already 
contains data,Remove this entry either from database or from the CSV file.")
                                root=customtkinter.CTk()
                                root.title("Records in Database.")
                                root.geometry("400x600")
                                root.config(bg="#D586FF")
                                scrollbar = Scrollbar(root)
                                scrollbar.pack( side = RIGHT, fill = Y )
                                listbox = Listbox(root, yscrollcommand = 
scrollbar.set,bg="#D586FF",font="32",height="33")
                                listbox.pack(fill = BOTH)
                                listbox.insert(END,"First Name : "+f[0])
                                listbox.insert(END,"Last Name : "+f[1])
                                listbox.insert(END,"Date Of Birth : "+str(f[2]))
                                listbox.insert(END,"Email ID : "+f[3])
                                listbox.insert(END,"Aadhar Number : "+f[4])
                                listbox.insert(END,"Father's Name : "+f[5])
                                listbox.insert(END,"Mother's Name : "+f[6])
                                listbox.insert(END,"City & State : "+f[7])

                                root.resizable(False,False)
                                root.mainloop()
                            error1()
                            t = t+1
                            break
                        else:
                                x = x+1
            else:
                for f in y:
                    for z in myresult:
                        if z[0] == f[4] or z[1] == f[3]:
                            def error2():
                                messagebox.showerror("Error",  "Database already
contains data,Remove this entry either from database or from the CSV file.")
                                root=customtkinter.CTk()
                                root.title("Records in Database.")
                                root.geometry("400x600")
                                root.config(bg="#D586FF")
                                scrollbar = Scrollbar(root)
                                scrollbar.pack( side = RIGHT, fill = Y )
                                listbox = Listbox(root, yscrollcommand =
scrollbar.set,bg="#D586FF",font="32",height="33")
                                listbox.pack(fill = BOTH)
                                listbox.insert(END,"First Name : "+f[0])
                                listbox.insert(END,"Last Name : "+f[1])
                                listbox.insert(END,"Date Of Birth : "+str(f[2]))
                                listbox.insert(END,"Email ID : "+f[3])
                                listbox.insert(END,"Aadhar Number : "+f[4])
                                listbox.insert(END,"Father's Name : "+f[5])
                                listbox.insert(END,"Mother's Name : "+f[6])
                                listbox.insert(END,"City & State : "+f[7])

                                root.resizable(False,False)
                                root.mainloop()
                            error2()
                            t = t+1
                            break
                        else:
                                x = x+1

            if t == 0:
                sqlform = "INSERT INTO
census_info(First_Name,Last_Name,Date_Of_Birth,Email_ID,Aadhar_Number,Father_Name,M
other_Name,City_And_State) VALUES (%s, %s, %s, %s,%s, %s, %s, %s)"
                mycursor.executemany(sqlform,y)
                mydb.commit()
                messagebox.showinfo("Successful","Imported All Data.")

        else:
            messagebox.showerror("Error","No Data Available in CSV file.")

def wcensus(file1): #APPEND ALL THE REC FROM DATABASE TO CSV FILE #COMPLETE
    mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
    mycursor = mydb.cursor()
    j = ["First Name","Last Name","Date Of Birth","Email ID","Aadhar
Number","Father's Name","Mother's Name","City & State"]
    mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
    myresult = mycursor.fetchall()
    with open(file1, "w+") as f:
        if len(myresult) > 0:
            writer = csv.writer(f)
            writer.writerow(j)
            for i in myresult:
                y = [i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7]]
                writer.writerow(y)
            messagebox.showinfo("Successful","Exported All Data to CSV file.")
        else:
            messagebox.showerror("Error","No Records in Database to Export.")

def ucensus(): #UPDATE ANY ENTRY INTO DATABASE #COMPLETE

    root=Tk()
    root.title("Update Records in Database.")
    root.geometry("400x565")
    root.config(bg="#D586FF")

    def fup():
        mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        FName = Label(root,text="Enter your First
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        FName.pack(padx=10,pady=10)
        FNameEnt = Entry(root,bd=2,width=40,font=(18))
        FNameEnt.pack(padx=10,pady=10)

        def fvals():
            global ufname,ufsearch
            ufsearch = SAadharEntry.get()
            ufname = FNameEnt.get()

        def execute():
            fvals()
            mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET First_Name='{}' WHERE
Aadhar_Number='{}'".format(ufname,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated First Name in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def lup():
        mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        LName = Label(root,text="Enter your Last
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        LName.pack(padx=10,pady=10)
        LNameEnt = Entry(root,bd=2,width=40,font=(18))
        LNameEnt.pack(padx=10,pady=10)

        def lvals():
            global ulname,ufsearch
            ufsearch = SAadharEntry.get()
            ulname = LNameEnt.get()

        def execute():
            lvals()
            mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET Last_Name='{}' WHERE
Aadhar_Number='{}'".format(ulname,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated Last Name in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def dobup():
        mydb =
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        DOB = Label(root,text="Enter your Date Of Birth(YYYY-MM-
DD)",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        DOB.pack(padx=10,pady=10)
        DOBEnt = Entry(root,bd=2,width=40,font=(18))
        DOBEnt.pack(padx=10,pady=10)
        def dobvals():
            global udob,ufsearch
            ufsearch = SAadharEntry.get()
            udob = DOBEnt.get()

        def execute():
            dobvals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET Date_Of_Birth='{}' WHERE 
Aadhar_Number='{}'".format(udob,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated Date of Birth in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def eidup():
        mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for 
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        EID = Label(root,text="Enter your Email 
ID",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        EID.pack(padx=10,pady=10)
        EIDEnt = Entry(root,bd=2,width=40,font=(18))
        EIDEnt.pack(padx=10,pady=10)
        def eidvals():
            global uemail,ufsearch
            ufsearch = SAadharEntry.get()
            uemail = EIDEnt.get()

        def execute():
            eidvals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET Email_ID='{}' WHERE 
Aadhar_Number='{}'".format(uemail,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated Email ID in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)
        root.resizable(False,False)
        root.mainloop()

    def aadharup():
        mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for 
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        ANumber = Label(root,text="Enter your Aadhar 
Number",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        ANumber.pack(padx=10,pady=10)
        ANumberEnt = Entry(root,bd=2,width=40,font=(18))
        ANumberEnt.pack(padx=10,pady=10)
        def aadharvals():
            global uadhaar,ufsearch
            ufsearch = SAadharEntry.get()
            uadhaar = ANumberEnt.get()

        def execute():
            aadharvals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET Aadhar_Number='{}' WHERE 
Aadhar_Number='{}'".format(uadhaar,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated Aadhar Number in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def fatherup():
        mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for 
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        FatherName = Label(root,text="Enter your Father's 
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        FatherName.pack(padx=10,pady=10)
        FatherNameEnt = Entry(root,bd=2,width=40,font=(18))
        FatherNameEnt.pack(padx=10,pady=10)
        def f1vals():
            global uf1name,ufsearch
            ufsearch = SAadharEntry.get()
            uf1name = FatherNameEnt.get()

        def execute():
            f1vals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET Father_Name='{}' WHERE 
Aadhar_Number='{}'".format(uf1name,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated Father's Name in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def mup():
        mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for 
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        MName = Label(root,text="Enter your Mother's 
Name",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        MName.pack(padx=10,pady=10)
        MNameEnt = Entry(root,bd=2,width=40,font=(18))
        MNameEnt.pack(padx=10,pady=10)

        def mvals():
            global umname,ufsearch
            ufsearch = SAadharEntry.get()
            umname = MNameEnt.get()

        def execute():
            mvals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET Mother_Name='{}' WHERE 
Aadhar_Number='{}'".format(umname,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated Mother's Name in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def csup():
        mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x600")
        root.config(bg="#D586FF")
        SAadhar = Label(root, text="Enter Aadhar Number to search for 
record.",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        CAS = Label(root,text="Enter your City and 
State(City/State)",bd=2,width=40,height=2,font=(18),bg="#3c1361")
        CAS.pack(padx=10,pady=10)
        CASEnt = Entry(root,bd=2,width=40,font=(18))
        CASEnt.pack(padx=10,pady=10)
        def csvals():
            global ucsname,ufsearch
            ufsearch = SAadharEntry.get()
            ucsname = CASEnt.get()

        def execute():
            csvals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET City_And_State='{}' WHERE 
Aadhar_Number='{}'".format(ucsname,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated City & State in database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    def crup():
        mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
        mycursor = mydb.cursor()
        mycursor.execute('SELECT * FROM census_info ORDER BY First_Name')
        root=customtkinter.CTk()
        root.title("Update Complete Record.")
        root.geometry("400x800")
        root.config(bg="#D586FF")

        SAadhar = Label(root, text="Enter Aadhar Number to search for 
record.",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        SAadhar.pack(padx=10,pady=10)
        SAadharEntry = Entry(root,bd=2,width=40,font=(18))
        SAadharEntry.pack(padx=10,pady=10)
        FName = Label(root,text="Enter your First 
Name",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        FName.pack(padx=10,pady=10)
        FNameEnt = Entry(root,bd=2,width=40,font=(18))
        FNameEnt.pack(padx=10,pady=10)
        LName = Label(root,text="Enter your Last 
Name",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        LName.pack(padx=10,pady=10)
        LNameEnt = Entry(root,bd=2,width=40,font=(18))
        LNameEnt.pack(padx=10,pady=10)
        DOB = Label(root,text="Enter your Date Of Birth(YYYY-MM-
DD)",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        DOB.pack(padx=10,pady=10)
        DOBEnt = Entry(root,bd=2,width=40,font=(18))
        DOBEnt.pack(padx=10,pady=10)
        EID = Label(root,text="Enter your Email 
ID",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        EID.pack(padx=10,pady=10)
        EIDEnt = Entry(root,bd=2,width=40,font=(18))
        EIDEnt.pack(padx=10,pady=10)
        ANumber = Label(root,text="Enter your Aadhar 
Number",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        ANumber.pack(padx=10,pady=10)
        ANumberEnt = Entry(root,bd=2,width=40,font=(18))
        ANumberEnt.pack(padx=10,pady=10)
        FatherName = Label(root,text="Enter your Father's 
Name",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        FatherName.pack(padx=10,pady=10)
        FatherNameEnt = Entry(root,bd=2,width=40,font=(18))
        FatherNameEnt.pack(padx=10,pady=10)
        MName = Label(root,text="Enter your Mother's 
Name",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        MName.pack(padx=10,pady=10)
        MNameEnt = Entry(root,bd=2,width=40,font=(18))
        MNameEnt.pack(padx=10,pady=10)
        CAS = Label(root,text="Enter your City and 
State(City/State)",bd=2,width=40,height=1,font=(18),bg="#3c1361")
        CAS.pack(padx=10,pady=10)
        CASEnt = Entry(root,bd=2,width=40,font=(18))
        CASEnt.pack(padx=10,pady=10)

        def ucvals():
            global 
ufsearch,ufname,ulname,udob,uemail,uadhaar,uf1name,umname,ucsname
        ufsearch = SAadharEntry.get()
        ufname = FNameEnt.get()
        ulname = LNameEnt.get()
        udob = DOBEnt.get()
        uemail = EIDEnt.get()
        uadhaar = ANumberEnt.get()
        uf1name = FatherNameEnt.get()
        umname = MNameEnt.get()
        ucsname = CASEnt.get()

        def execute():
            ucvals()
            mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
            mycursor = mydb.cursor()
            mycursor.execute("UPDATE census_info SET 
First_Name='{}',Last_Name='{}',Date_Of_Birth='{}',Email_ID='{}',Aadhar_Number='{}',
Father_Name='{}',Mother_Name='{}',City_And_State='{}' WHERE 
Aadhar_Number='{}'".format(ufname,ulname,udob,uemail,uadhaar,uf1name,umname,ucsname
,ufsearch))
            mydb.commit()
            root.destroy()
            messagebox.showinfo("Successful","Updated complete Record in 
database.")

        BTN = Button(root,text="SAVE",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=1,font=(32),command= execute)
        BTN.pack(padx=10,pady=10)

        root.resizable(False,False)
        root.mainloop()

    bt1 = Button(root,text="Update First Name.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = fup).pack()
    bt2 = Button(root,text="Update Last Name.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = lup).pack()
    bt3 = Button(root,text="Update Date of Birth.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = dobup).pack()
    bt4 = Button(root,text="Update Email ID.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command=eidup).pack()
    bt5 = Button(root,text="Update Aadhar Number.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = aadharup).pack()
    bt6 = Button(root,text="Update Father's Name.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = fatherup).pack()
    bt7 = Button(root,text="Update Mother's Name.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = mup).pack()
    bt8 = Button(root,text="Update City & State.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = csup).pack()
    bt9 = Button(root,text="Update Complete Entry.",activebackground="#800080",bg = 
"#7c5295",bd=2,width=40,height=2,font=(32),command = crup).pack()

    root.resizable(False,False)
    root.mainloop()
    root.destroy()

def setupcensus():#SETUP TABLE IN A PRE-MADE DATABASE #COMPLETE
    mydb = 
mysql.connector.connect(host="localhost",user=entT1,passwd=entT2,database=entT3)
    mycursor = mydb.cursor()
    mycursor.execute("create table if not exists census_info(First_Name varchar(64) 
not null,Last_Name varchar(64) null,Date_Of_Birth date not null,Email_ID 
varchar(255) not null unique,Aadhar_Number varchar(16) not null primary 
key,Father_Name varchar(64) not null,Mother_Name varchar(64)  not 
null,City_And_State varchar(100) not null);")
    messagebox.showinfo("Successful","Created table.Don't repeat this command
again.")

#Frontend

def rfilename():
    file1 = simpledialog.askstring("File Input","Enter File Name with .CSV in the
end.")
    if len(file1) == 0:
        messagebox.showerror('Error', 'Error: No file provided.')
    else:
        rcensus(file1)

def wfilename():
    file1 = simpledialog.askstring("File Input","Enter File Name with .CSV in the
end.")
    if len(file1) == 0:
        messagebox.showerror('Error', 'Error: No file provided.')
    else:
        wcensus(file1)

def table2():
    Tlabel = Label(text="Main Menu : ",bg="#bca0dc",font=("Arial Rounded MT
Bold",56,"bold")).pack()
    button1 = Button(text="Show Database
Records.",command=lcensus,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button2 = Button(text="Add A New Record in
Database.",command=icensus,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button3 = Button(text="Update A Record in
Database.",command=ucensus,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button4 = Button(text="Delete A Record from
Database.",command=dcensus,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button5 = Button(text="Search for A Record in
Database.",command=scensus,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button6 = Button(text="Import Data from
CSV.",command=rfilename,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button7 = Button(text="Export Records from Database to
CSV.",command=wfilename,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button8 = Button(text="Setup Table(from premade database
only.).",command=setupcensus,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)
    button9 = Button(text="Quit
Program.",command=quit,activebackground="#800080",bg =
"#7c5295",bd=2,width=40,height=2,font=(32)).pack(padx=10,pady=10)

root.resizable(False,False)

root.mainloop()

#made by Aman
```

TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test[1] , with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

TESTING METHODS

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

BLACK BOX TESTING

Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

SPECIFICATION-BASED TESTING

Specification-based testing aims to test the functionality of software according to the applicable requirements.[16] Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks

ADVANTAGES AND DISADVANTAGES

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.

That's why there are situations when (1) a black box tester writes many test cases to check something that can be tested by only one test case, and/or (2) some parts of the back end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

WHITE BOX TESTING

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these)

Types of white box testing:-

The following types of white box testing exist:

- api testing - Testing of the application using Public and Private APIs.
- Code coverage - creating tests to satisfy some criteria of code coverage.

For example, the test designer can create tests to cause all statements in the program to be executed at least once.

- fault injection methods.

- mutation testing methods.
- static testing - White box testing includes all static testing.

CODE COMPLETENESS EVALUATION

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

- FunctionCoverage: Which reports on functions executed and
- StatementCoverage:Which reports on the number of lines executed to complete the test.

They both return coverage metric, measured as a percentage

HARDWARE AND SOFTWARE REQUIREMENTS

I.OPERATING SYSTEM	:	WINDOWS 7 AND ABOVE
II. PROCESSOR	:	PENTIUM(ANY) OR AMD ATHALON(3800+- 4200+ DUALCORE)
III. MOTHERBOARD	:	1.845 OR 915,995 FOR PENTIUM 0R MSI K9MM-V VIAK8M800+8237R PLUS CHIPSET FOR AMD ATHALON
IV. RAM	:	512MB+
V. Hard disk	:	SATA 40 GB OR ABOVE
VI. CD/DVD r/w multi drive combo:		(If back up required)
VII. FLOPPY DRIVE 1.44 MB	:	(If Backup required)
VIII. MONITOR 14.1 or 15 -17 inch		
IX. Key board and mouse		
X. Printer	:	(if print is required – [Hard copy])

SOFTWARE REQUIREMENTS:

I. Windows OS

II. Python

III. MySQL

IV. Visual Studio Code

V. Custom TKinter

.

BIBLIOGRAPHY

1. ***Computer Science - Class XII***
2. ***Website:*** **https://www.w3resource.com,https://www.geeksforgeeks.org**