ABSTRACT

The Phone Book Management System works specifically for tracking people. In older days we stored all our information contact details in books and papers. Which was very difficult in persevering book for long period, In order to resolve this problem. we proposed a new system using this application we can store all the contact details in this system.

In manual method if we forget information book then it was very difficult to get the contact details. By using this system, we can see our contacts from anywhere in the world, here we have security to our login page and no one can see our contacts without proper authentication. In this project we can save our contacts, address and we can search them by name and also, we can view all the contacts at a time.

This System is developed using the general required by all the user while using the phone book. This system contains a set of basic functions for adding, searching, updating, viewing and deleting the contacts that already exists in the application.

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

Phone Book is a project that is provided by the technical assignment help to us in that we get a simple SQL based solutions to store our contacts. we can use it to replace hard phonebook or even use it as an office-wide phone directory. This will help user to easily search and manage contacts using this system. This system is developed using the general need required by the user while using the phone directory book. In order to keep updated the phone book directory, the admin will have the authority to add and delete as well as modify the existing record with the phone book directory.

1.2 PROBLEM STATEMENT

To design and develop a Phone Book Management System project using Python as both front end and back end and database used is Oracle Sql.

1.3 OBJECTIVES

1. User can search details of the new phone number, list of contacts, contact details.
2. In this database management system user can easily manage new phone number details, address details, city name details etc...
3. User can add, delete and update the records of stored data.
4. User can restore the data, once if deleted.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

- Processor: Dual core/Core duo processors and Above
- Ram: 2GB and Above
- Hard Disk: 256GB and Above

2.2 SOFTWARE REQUIREMENTS

- Operating System: Windows XP and Above
- Software: Visual Studio Code, PyQt5 (Designer)
- Programming Language: Python and SQL
- Database: Oracle SQL

2.3 FUNCTIONAL REQUIREMENTS

CONTACT: This table includes the information of attributes C_ID, NAME, EMAIL, PHNO, RELATIONSHIP, TYPE, DOB, PINCODE. This tables include ADD, DELETE, UPDATE operations.

ADDRESS: This table includes the attributes C_ID, HNO, HNAME, AREA, PINCODE. This tables include ADD, DELETE, UPDATE operations.

DETAILS: This table includes the attributes C_ID, PHNO, ZONE, COMPANY, LOCATION, PHNO_TYPE. This tables include ADD, DELETE operations.

ZIPCODE: This table includes the attributes PINCODE, CITY, STATE. This table include ADD, DELETE operations.

RESTORE: This table includes the information of attributes C_ID, NAME, P_L_NUMBER, EMAIL, RELATIONSHIP, TYPE, DOB, HNO, HNAME, AREA, CITY, STATE, PINCODE, TIME_ZONE, COMPANY, LOCATION, P_TYPE. This table include ADD, DELETE operations.

2.4 NON-FUNCTIONAL REQUIREMENTS

Performance: The Application is built in such a way by taking all the necessary precautions and this as gone through rigorous testing in order to ensure the user a swift feel.

Availability: It will be available only when there is an internet connection because it is an online application.

Usability: The GUI is made so responsive which would make the user to easily work with the software.

CHAPTER 3

SYSTEM ARCHITECTURE DESCRIPTION

3.1 ER DIAGRAM

Entity	Attributes
CONTACT	C_ID,NAME,EMAIL,PHNO,REATIONSHIP, TYPE,DOB,PINCODE.
ADDRESS	C_ID,HNO,HNAME,AREA,PINCODE.
DETAILS	C_ID,PHNO,ZONE,COMPANY,LOCATION, PHNO_TYPE.
ZIPCODE	PINCODE, CITY, STATE.
RESTORE	C_ID, NAME, P_L_NUMBER, EMAIL, RELATIONSHIP, TYPE, DOB, HNO, HNAME, AREA, CITY, STATE, PINCODE, TIME_ZONE, COMPANY, LOCATION, P_TYPE.

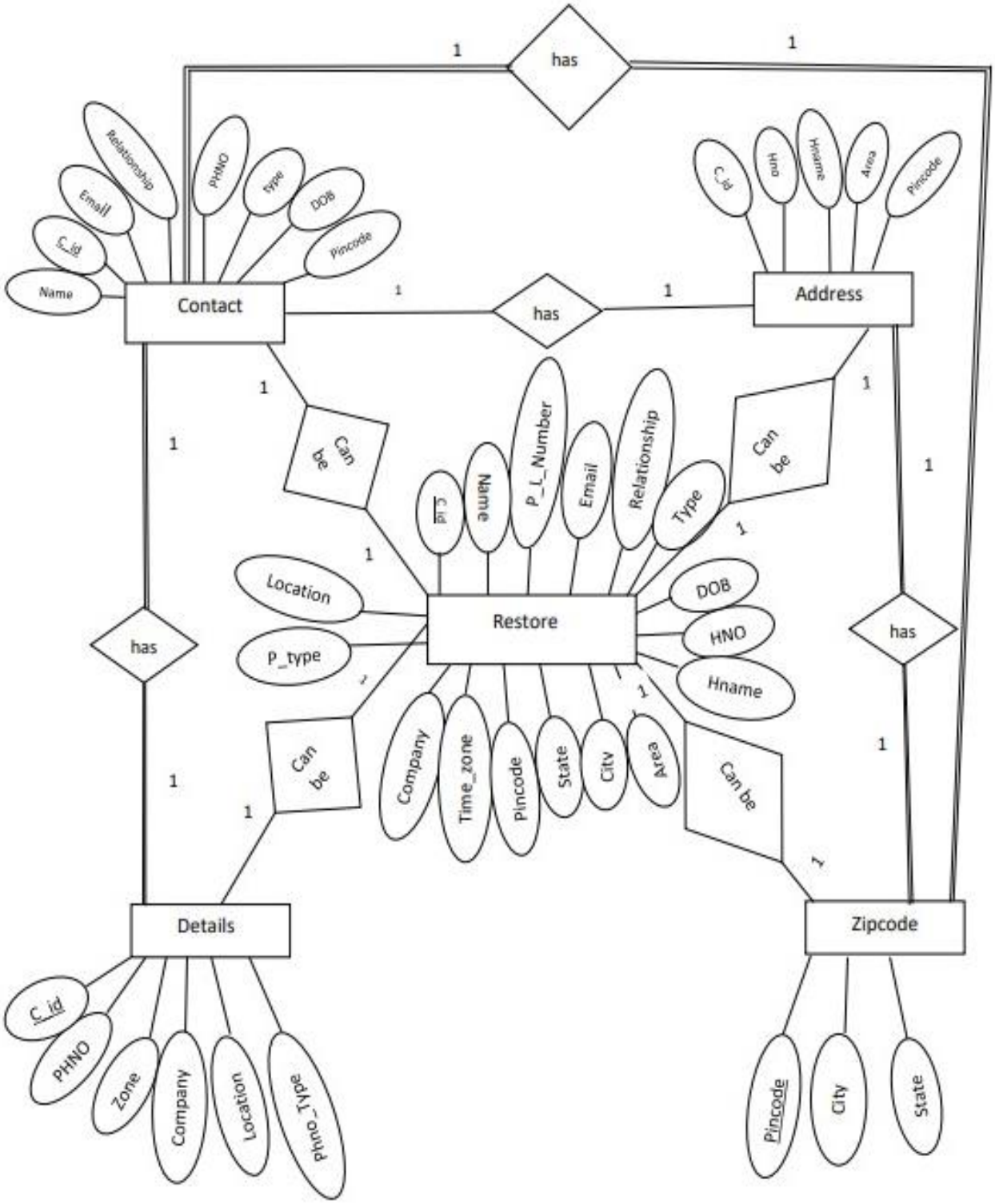


FIG: ER DIAGRAM

SCHEMA DIAGRAM:

CONTACT

C_ID	NAME	EMAIL	PHNO	RELATIONSHIP	TYPE	DOB	PINCODE

ADDRESS

C_ID	HNO	HNAME	AREA	PINCODE

DETAILS

C_ID	PHNO	ZONE	COMPANY	LOCATION	PHNO_TYPE

ZIPCODE

PINCODE	CITY	STATE

RESTORE

C_ID	NAME	P_L_NUMBER	EMAIL	RELATIONSHIP	TYPE	DOB
HNO	HNAME	AREA	CITY	STATE	PINCODE	TIME_ZONE
COMPANY	LOCATION	P_TYPE				

CHAPTER 4

IMPLEMENTATION

4.1 OVERVIEW OF MODULES / COMPONENTS

The proposed system has following modules

Login window: This module is the start of the application. It consists of two fields Username and Password by which the application validates verified users who can access the application's benefits. If the user enters a wrong username or the password, then the application warns the user by popping up a message saying you have entered a wrong username and password. Please try again by entering the correct username and password. After logging in the user can use our application. Hence this module plays an important role in the application.

Main window: This is the window where the user can start working with the features of the application. The following are our main features:

1.HOME: This feature displays the existing contacts in our application. You can select the type of number you need like Mobile Number or Landline Number. After you click on show button it will displays you all the details of the contact.

2.ADD CONTACT: This feature is used to save new contact number and the details of the particular person. The data will be stored in the SQL database. For our application we used ORACLE as our database. When you press add button the contact will be added and stored in SQL database. If you enter wrong inputs u will get a popup message that you have not filled the details correctly. If you try to add the same number again then it will give you the popup message saying that the number already exists and content will be displayed in the table. We have a clear button, it will clear the details entered by the user.

3.UPDATE: This feature is used to update the existing contact in our database. First we have to search a contact by its number then it will display the details in the respective fields then below we have the options where we can modify the details such as NAME, HNO, HNAME, EMAIL, AREA. We have a clear button, it will clear the details and it will also clear the number entered by the user to search a particular contact.

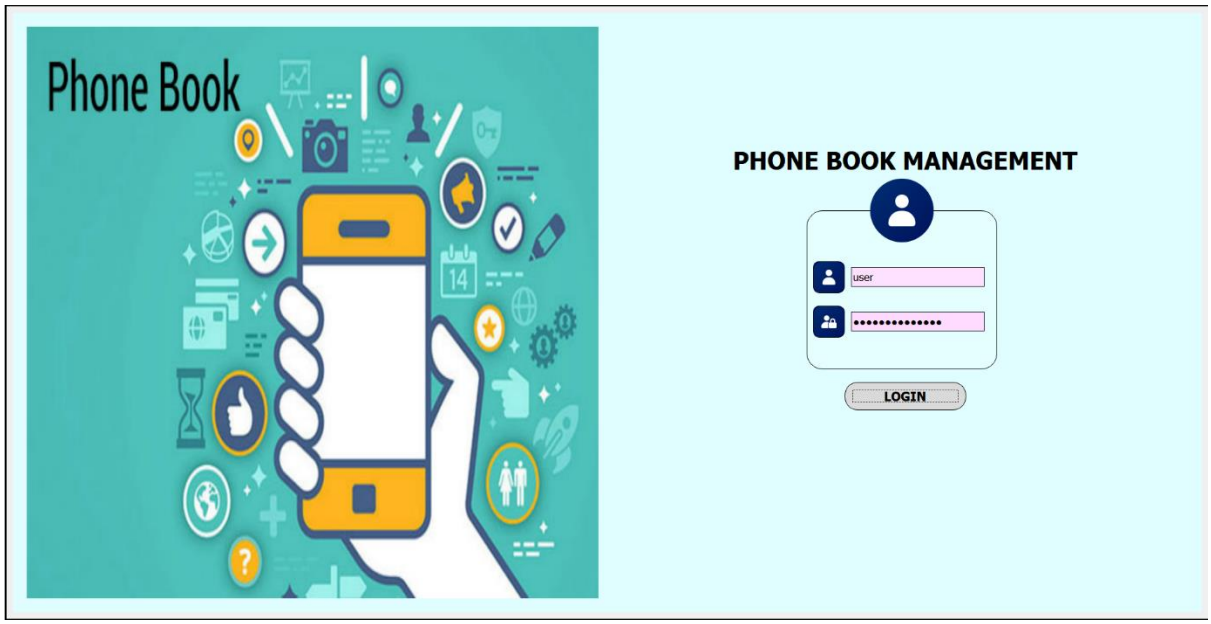
3.DELETE: This feature is used to delete the existing contact from the database. First the user has to search the contact by entering Mobile/Landline number or C_ID. After clicking on serach button the particular contact details will be displayed in the table. After pressing the delete button the data of the particular data will be deleted from the database and then the deleted data is stored in the table called as restore when you press the restore button the data will be restored in the existing database. Once you press permanent delete button then the data will be deleted even from the restore table, the data cannot be restored back.

4.SEARCH: This feature is used to search the existing contact from the database. In this operation we have two options first is Single Search, second is Search All.

- Single Search: Here we are searching the details of the contact either by name or by Mobile/Landline number. Once you enter the name or Mobile/Landline number press search button, It will search the related data from the database and displays the details in the below table.
- Search All: Once you press on search all button it will directly retrieve all the details of contacts present in the database.

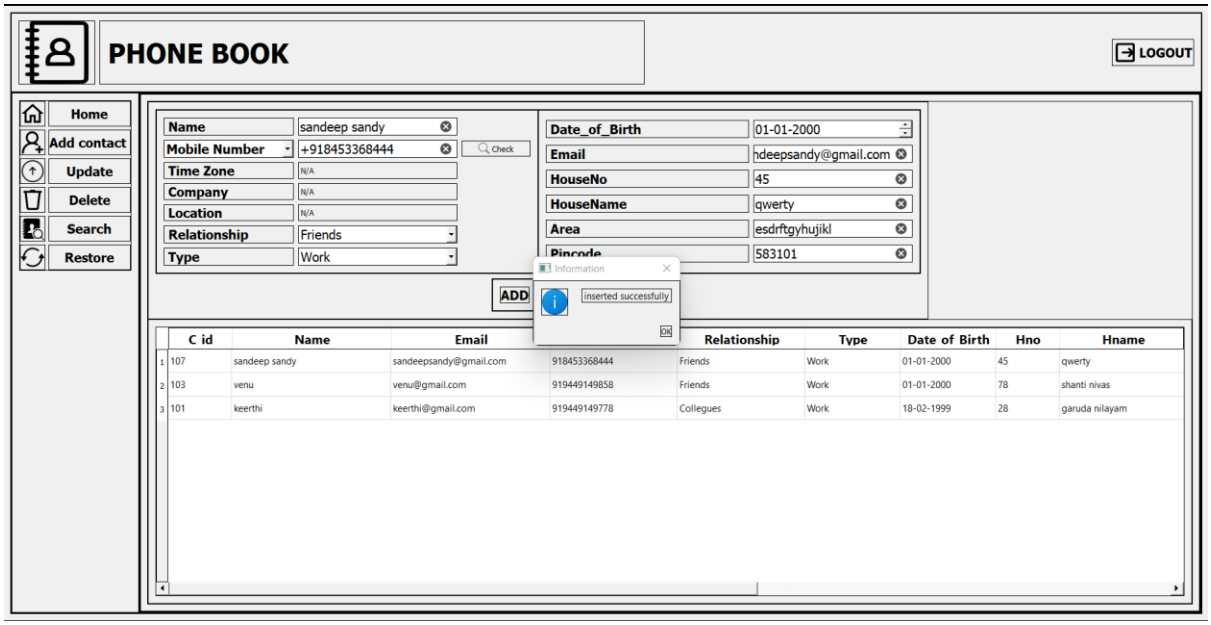
4.2 RESULTS

LOGIN WINDOW

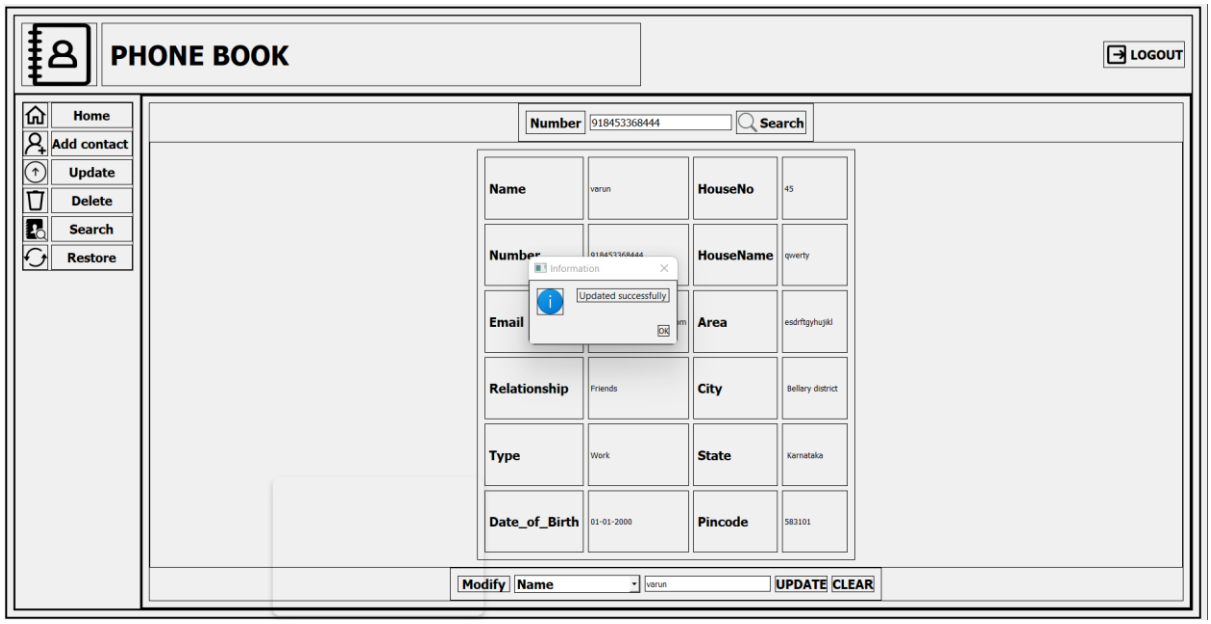


HOME PAGE

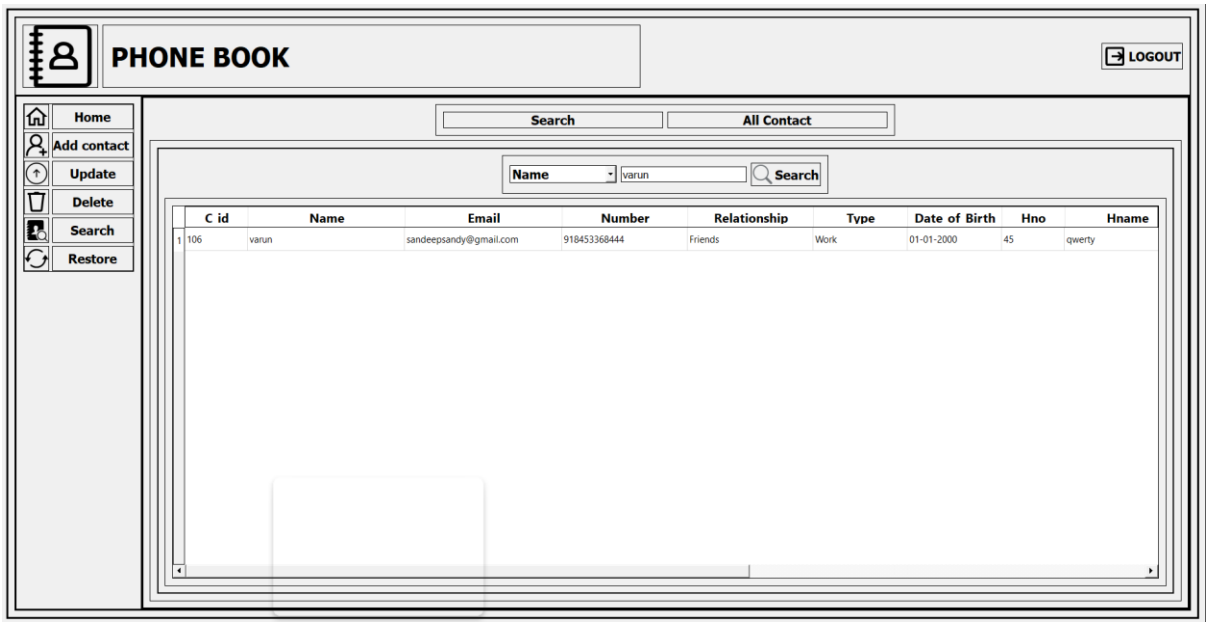
ADD CONTACT



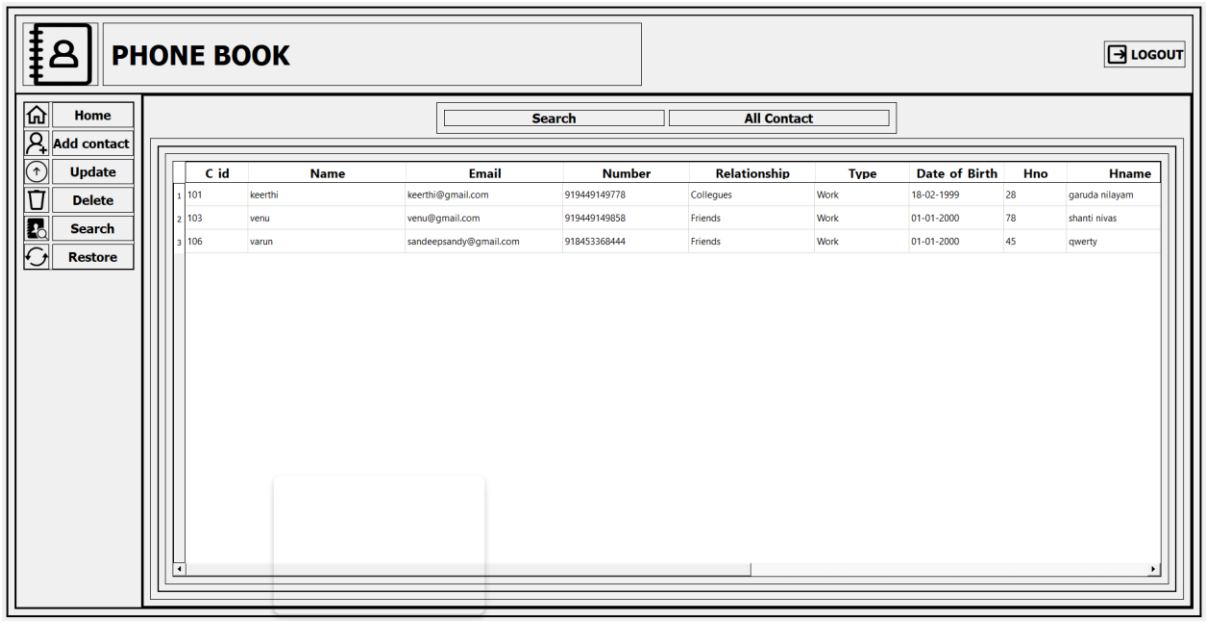
UPDATE



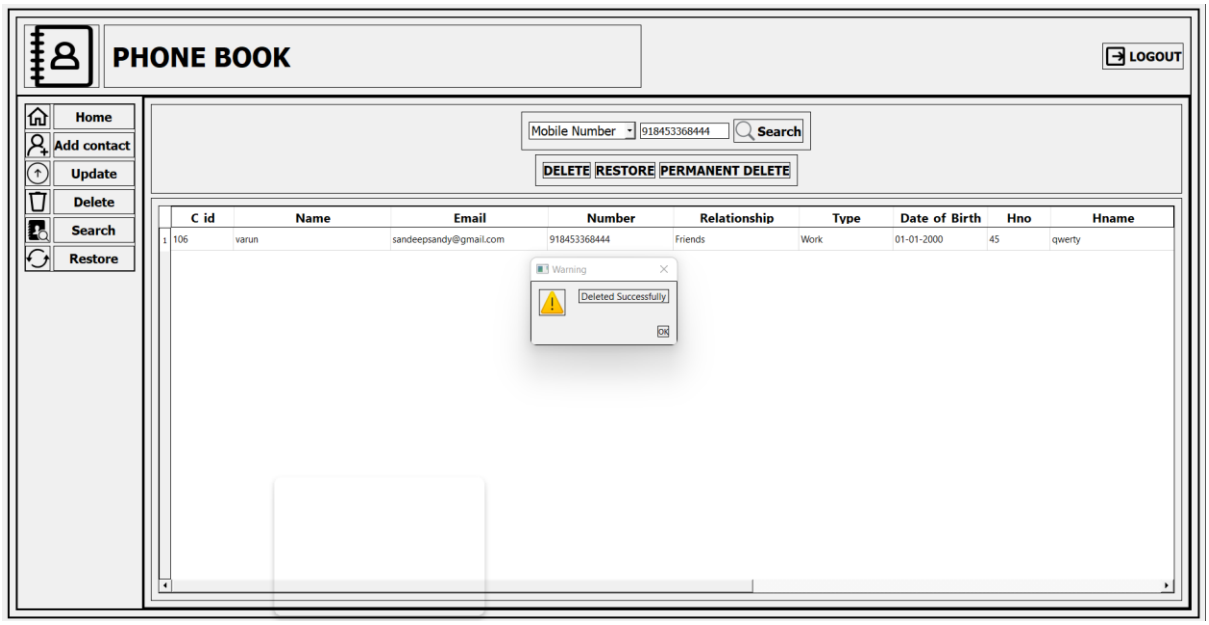
SINGLE SEARCH



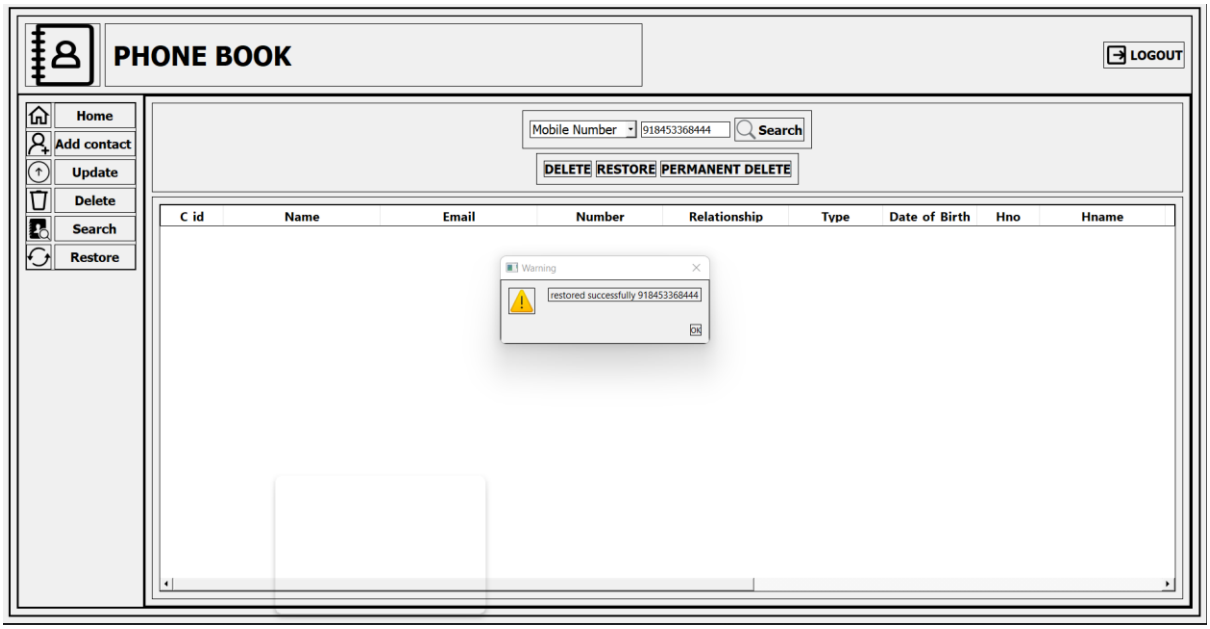
SEARCH ALL



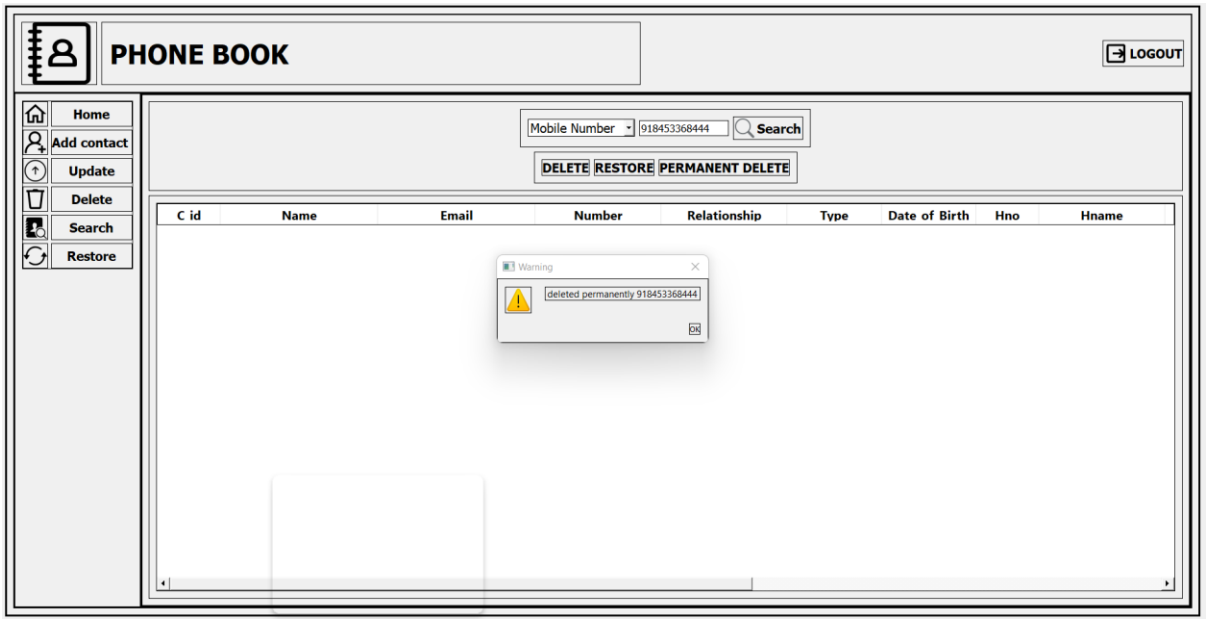
DELETE



RESTORE



PERMANENT DELETE



CONCLUSION

It is concluded that the application works well and satisfy the end users. The application is tested very well and errors are properly debugged. This system is user friendly so everyone can use easily. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully. Further enhancements can be made to the application, so that the application functions very attractive and useful manner than the present one.

REFERENCES

[1] https://www.w3schools.com/sql/default.asp
[2]. "Fundamentals of Database System'' by Shamkant.B.Navathe, 5th Edition, Published by Dorling Kindersley.
[3] https://www.geeksforgeeks.org/oracle-database-connection-in-python/
[4] https://stackoverflow.com/
[5] https://wiki.python.org/moin/PyQt/Tutorials