GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

# PASSWORD PROTECTED MY CONTACT BOOK


**Submitted by**

**Brijesh Kumar Shukla**

**(1613101228 / 16SCSE101087)**


**in partial fulfillment for the award of the degree**

**of**

**Bachelor of Technology**

**IN**

**Computer Science and Engineering**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**Under the Supervision of**

Mr. S.P. Ramesh

**<Assistant Professor >**

**MONTH & YEAR**

**MAY,2020**

GALGOTIAS UNIVERSITY

# SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

## BONAFIDE CERTIFICATE

**Certified that this project report "PASSWORD PROTECTED MY CONTACT BOOK" is the bonafide work of "BRIJESH KUMAR SHUKLA [1613101228]" who carried out the project work under my supervision.**

**SIGNATURE OF HEAD**

**Dr. MUNISH SHABARWAL**
**PhD (Management), PhD (CS)**
**Professor & Dean,**
**School of Computing Science & Engineering**

**SIGNATURE OF SUPERVISOR**

**Mr. S. P. Ramesh B.Tech.,**
**M.E.,**
**Assistant Professor**
**School of Computing Science & Engineering**

## STUDENT DECLARATION

I am **BRIJESH KUMAR SHUKLA**, student of B.TECH in Computer Science Department of 8th semester hereby declare that the presented report of final project titled "**PASSWORD PROTECTED MY CONTACT BOOK**". The information and data give in the report is authentic to the best of my knowledge after the completion of seven semester I complete my website.

I also confirm that the report is only prepared for my academic requirement, not for any other purpose. Its my not be used with the interest of the opposite party of the corporation.

NAME: BRIJESH KUMAR SHUKLA

ROLL NO: 1613101228

BRANCH: CSE

## ACKNOWLEDGEMENT

"*Gratitude is not a thing of expression; it is more matter of feeling*."

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of study.

I express my sincere gratitude to Mr. S. P. Ramesh worthy Principal for providing me an opportunity to undergo final project at final semester "GALGOTIAS UNIVERSITY".

I am thankful to Mr. S. P. Ramesh for his support, cooperative, and motivation provided to me during the project for constant inspiration, presence and blessings.

I also extend my sincere appreciation to Mr. R. Vishwanathan who provided his valuable suggestions and precious time in accomplishing my project report.

Lastly, I would like to thanks the almighty and my parents for this moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.

# ABSTRACT

**Customer, Data, Record Security**: User can Store his data and contact in this system. Through this system data store in the Database. When user want then they can easily retrieve from the database and also they can remove and edit the data into the database.

**Report Generation**: A Report Generation system will be developed for the user of contact and his detail. This My contact book System will have both details and summary reports of his contact's person. To develop a system for the management of data, Store and maintenance processes that will be performed with a click of mouse button by given user id and password. To develop a system that has a good management of data along with integrity and minimizing redundancy. To develop a system that will be user friendly in all possible ways. To develop a system that provides easier work than existing system for the user. To develop a secure system that can be accessed only by authorized users.

**TABLE OF CONTENTS**

## LIST OF FIGURES

# CHAPTER 1

## INTRODUCTION

The computer has brought revolution in every sphere of human life. Whether it is business, education field, governance, medical science etc. The computer has reduced the human work load, business are going global and everything is available at the click of mouse. The concept of contact book has been introduced and we can store the contact online and make secure through my contact book.

Presently I am proposing the system “Password Protected My Contact Book”. The generally issue clients handwritten contacts and they enter details in manual registers. And maintain MS Excel file for product rate. So, the proposed system will computerize their manual contact generation system.

As stated above the generally clients presently use manual bills and hand written record to maintains their contact list, customer list, and keep the invoice, there is lot of duplicate work, and chance of mistake. When the client is changed his mail id, they need to update each and every hand written record.

There is no security; anybody can access any report and sensitive data, also there are no reports to find out the data volume, stock list, and summary report. This My Contact Book system is used to overcome the entire problem which the client is facing currently, and making complete atomization of manual and XL contact system.

## CHAPTER 2

## LITERATURE SURVEY

1. TITLE: - Document Security Solutions

AUTHORS: -Rogerio Pontes, Francisco Maia, Joao˜ Paulo, and Ricardo Manuel Pereira Vilac¸a. Saferegions

For preventing reveal of the internal information, companies are deploying Data Loss Prevention solutions (DLP). The solution also has the functions that is dealing with the document containing critical data such as intellectual properties and personal information. Normally the DLP solutions are monitoring the transmission of critical files from internal systems. Some DLP solutions use encryption system for preserving the secrecy of documents. For making the encryption system work, DLP system orchestrates the employees' computer systems for managing keys used for encrypting and decrypting documents.

## 2. TITLE: **Varying password-based security scheme**

**AUTHORS:** Karen Scarfone and Murgiah Souppaya

This aims at providing the security more or less in the same traditional way but in a different fashion. This scheme eliminates the shouldering problem. Here the user is granted access to the system if and only if he enters the correct password according to the predefined specific positions in the entered text. The key idea behind this technique that separates it from ordinary traditional ones is the varying length of the password. Out of the text entered by the user only the positions defined will be read and matched to grant access. This scheme is inspired by the UNIX system, in which [12, 1] • UNIX systems permit passwords of unlimited length to be typed, but only first eight characters are significant, which converts some seemingly secure passwords into insecure ones, e.g., "Carolina71Duke59" becomes simply "Carolina". Here, we are giving the user more freedom i.e. the user can choose the starting position, his/her own password and then can write anything as postfix of the password. At the time of registration of user the system will ask about the position from which the password will start and at the time of authentication the system will trim the password string entered by user. For example, if we select even a single character password such as "s" and we select the size of the dummy prefix to be of 6 characters such that the character at the seventh position will be read and matched and the size of the dummy postfix can be of any length. Then the valid passwords are

- rewddesasklnfljfipsoejf
- jutghjsryjkbfrtumk
- htdbnxsghtrfghmkjuikyupoi

# CHAPTER 3

# SYSTEM ANALYSIS

## HARDWARE & SOFTWARE USED: -

Today we have huge no. of technologies used for software development these are C, C++, Python, Java, etc. as back end and HTML, CSS, JavaScript, etc. as front end tools. But Graphical User Interface (GUI) is used as front end and Python language is used as back end respectively in my Project.

I have chosen Python for Front end because it provides quick software development and it also provides quick error detection and correction. As web pro, developing is a core part and there is no shortage of programming languages, with Python being the trending one. Python is an interactive programming language and getting started with programming a GUI (Graphical User Interface) framework is not much of a difficult task. Python has a diverse range of options for GUI frameworks.

Tkinter is commonly bundled with Python, using Tk and is Python's standard GUI framework. It is popular for its simplicity and graphical user interface. It is open source and available under the Python License. One of the advantages of choosing Tkinter is that since it comes by default, there is an abundance of resources, both codes and reference books. Also with the community being old and active, there are many users who can help you out in case of doubts.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.

- Enter the main event loop to take action against each event triggered by the user.

**Tools: -**

**Hardware used**: -

| | |
|---|---|
| Monitor | : 15” color monitor |
| RAM | : 4GB |
| Hard Disk Space | : 500GB |
| **Processor** | **: intel core i3** |

**Software used:-**

| | |
|---|---|
| Operating System | : Windows 10 |
| IDLE | : Python 3.x, PyCharm |
| Front end | : Python GUI |
| Back end | : Python |
| Database | : MYSQL Server |
| Web Server | : Apache |

**Technology: -**

**Python** is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. Due to concern about the amount of code written for Python 2, support for Python 2.7 (the last release in the 2.x series) was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

## Existing System

The following sections summarize features and functionality performed by user. Please see below:

| Manage Customers - Analyze Customers |
| --- |
| Project is able to identify his contact details. |
| Project is able to handle customer's request with user id and password. |
| |
| Manage Contacts (Catalog, Categories) |
| Project is able to browse the contact data. |
| Project is able to add a new contact data. |
| Project is able to edit an existing data |
| User (System Admin.) is able to search the contact data |
| User (System Admin.) is be able to search the data by the email id |
| |
| Reports |
| User is able to store the data into the database. |
| User is able to check contact detail. |
| User is able to retrieve the data by mail id after login by id and password. |

# PROPOSED MODEL

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming on python GUI based applications and no some extent Windows Applications and SQLITE server, but also about all handling procedure related with "**PASSWORD PROTECTED MY CONTACT BOOK**". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

**Benefits:**

The project is identified by the merits of the system offered to the user. The merits of this projects are as follows:

- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the personal information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stage.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.

Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he

# CHAPTER 4

## SYSTEM DESIGN

**Data Flow Diagram (DFD): -**

**DFD for Contact Book System: -**

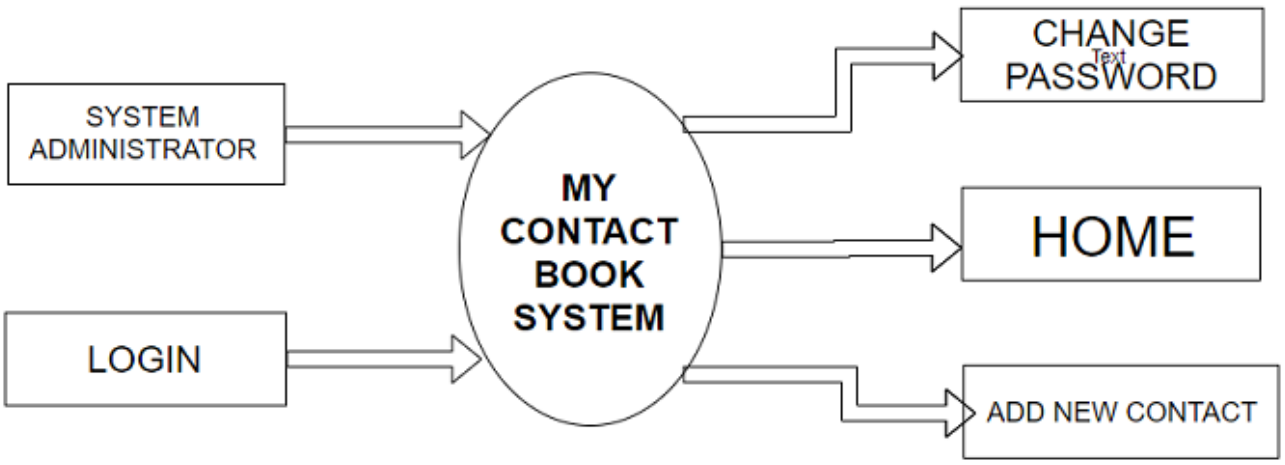



**Fig: ZERO$^{TH}$ LEVEL DFD**

**DFD for system admin: -**

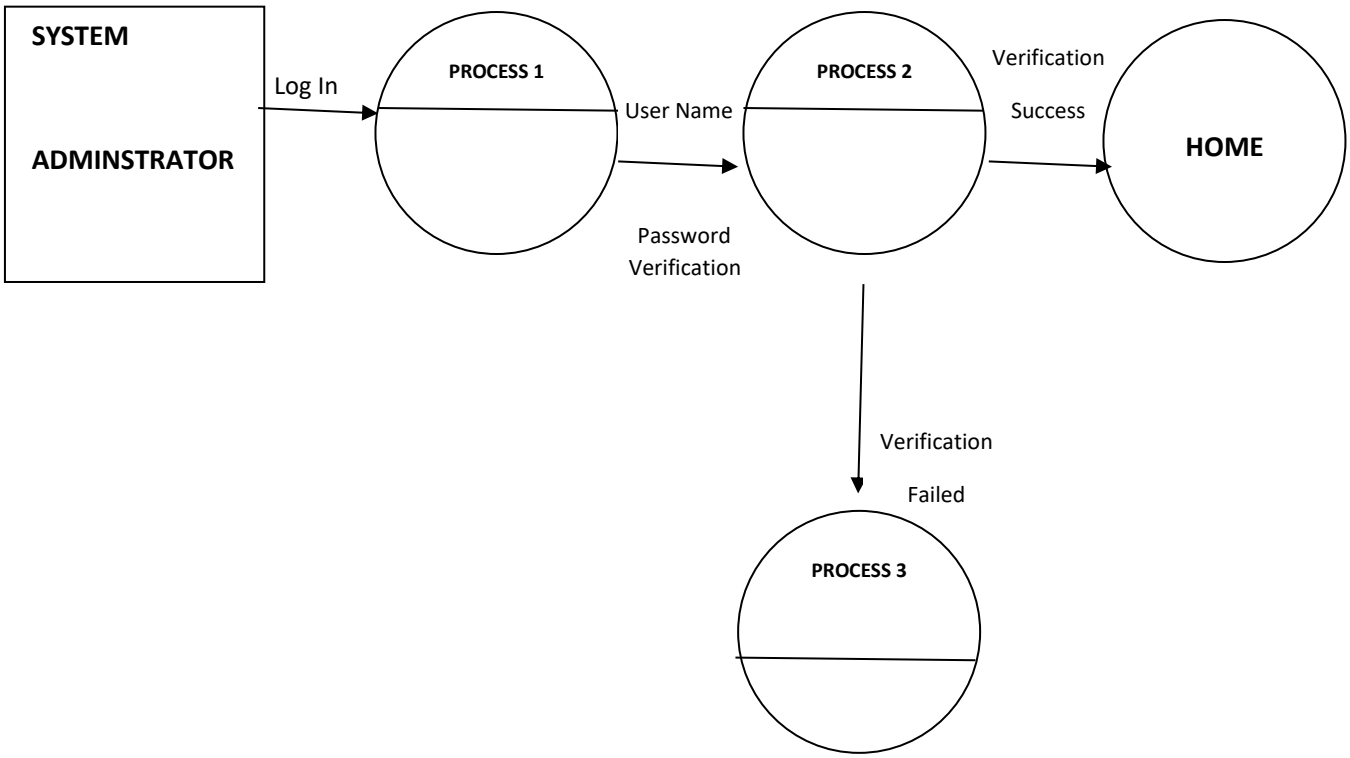



**Fig: 1ST LEVEL DFD**

## DFD for customer:-

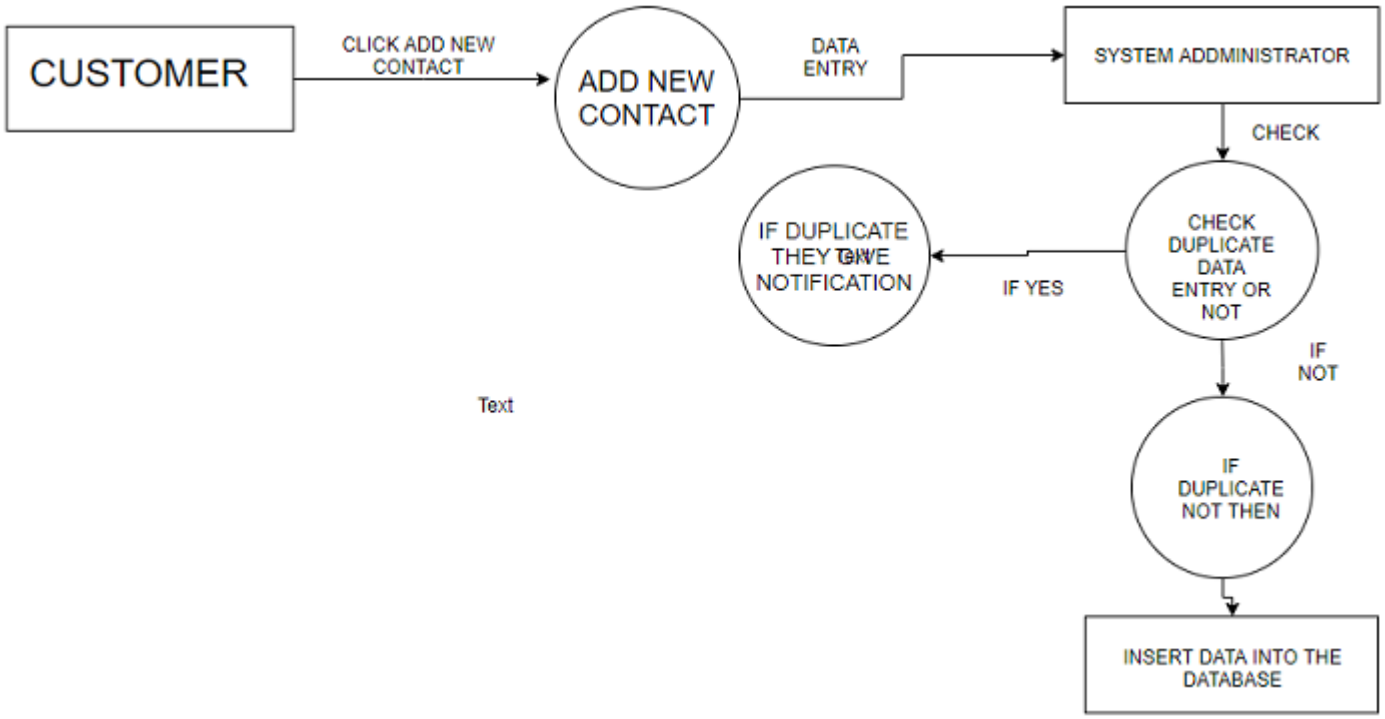



**Fig: 2ND LEVEL DFD**

# ER DIAGRAM FOR GENERAL STORE BILLING SYSTEM

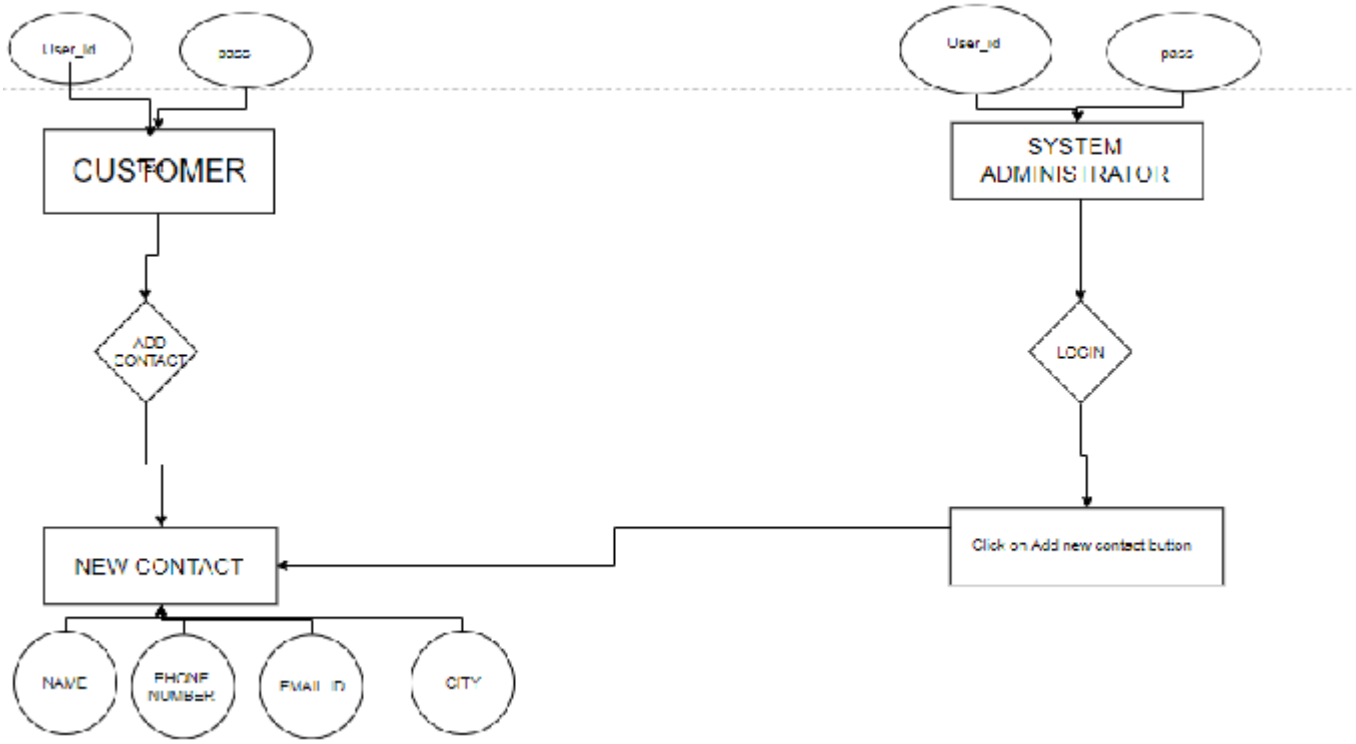



**Fig: ER-Diagram**

# CHAPTER 5

## CONCLUSION AND FUTURE ENHANCEMENT

### CONCLUSION: -

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming on python GUI based applications and no some extent Windows Applications and MYSQL server, but also about all handling procedure related with "**PASSWORD PROTECTED MT CONTACT BOOK**". It also provides knowledge about the latest technology used in developing web enabled application and data securing technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

**Benefits:**

The project is identified by the merits of the system offered to the user. The merits of this projects are as follows:

- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stage.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.
- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.

**FUTURE SCOPE: -**

So, there are many things for future enhancement of this project. This future enhancement that are possible in the project are as follows:

- To optimize the query this is embedded in the system.
- Linking and integration of any legacy system for accounting.
- This project future can develop as a password protected my contact book.
- This project will help the storing the data.
- This project enables user to maintain a great database of all information and secure that data by user id and password.
- Project will enable to see report regarding saved data.
- It is easy to maintain in future prospect.

## APPENDICES

### A. SOURCE CODING

```
from tkinter import *
from tkinter.ttk import *
from tkinter import filedialog
from PIL import Image,ImageTk
from sqlite3 import *
from tkinter import messagebox

class ManageContactsFrame(Frame):
    def __init__(self, parent):
        Frame.__init__(self, parent)

        self.pack(fill = BOTH, expand = TRUE)
        self.style = Style()
        self.style.configure('TFrame', background = 'white')
        self.con = connect('mycontacts.db')
        self.cur = self.con.cursor()
        self.create_view_all_contacts_frame()

    def create_view_all_contacts_frame(self):
        self.view_all_contacts_frame = Frame(self)
        self.view_all_contacts_frame.place(relx=.5, rely=.5, anchor=CENTER)

        self.style.configure('TButton', font=(NONE, 15), width=20)
        self.add_new_contact_button = Button(self.view_all_contacts_frame, text="Add New Contact",
        command=self.add_new_contact_button_click)


        self.add_new_contact_button.grid(row=0, column=1, sticky=E, pady=10)
        self.style.configure('TLabel', background='white', font=(NONE, 15))

        self.name_label = Label(self.view_all_contacts_frame, text="Name: ")
        self.name_label.grid(row=1, column=0, pady=10, sticky=W)

        self.name_entry = Entry(self.view_all_contacts_frame, font=(NONE, 15), width=62)
        self.name_entry.grid(row=1, column=1, sticky=W)
        self.name_entry.bind('<KeyRelease>', self.name_entry_key_released)
        self.style.configure('Treeview.Heading', font=(NONE, 15))
        self.style.configure('Treeview', font=(NONE, 14), rowheight=25)
        self.contacts_treeview = Treeview(self.view_all_contacts_frame,
        columns=('name', 'phone_no', 'email_id', 'city'), show='headings')
```

```
    self.contacts_treeview.heading('name', text="Name", anchor=W)
    self.contacts_treeview.heading('phone_no', text="Phone No", anchor=W)
    self.contacts_treeview.heading('email_id', text="Email Id", anchor=W)
    self.contacts_treeview.heading('city', text="City", anchor=W)
    self.contacts_treeview.column('name', width=250)
    self.contacts_treeview.column('phone_no', width=150)
    self.contacts_treeview.column('email_id', width=200)
    self.contacts_treeview.column('city', width=150)
    self.contacts_treeview.grid(row=2, column=0, columnspan=2, pady=10)
    self.contacts_treeview.bind('<<TreeviewSelect>>', self.create_update_delete_contact_frame)

    self.cur.execute("select * from Contact")
    self.fill_contacts_treeview()

def add_new_contact_button_click(self):
    self.view_all_contacts_frame.destroy()

    self.add_new_contact_frame = Frame(self)
    self.add_new_contact_frame.place(relx=.5, rely=.5, anchor=CENTER)

    self.style.configure('TLabel', background='white', font=(NONE, 15))

    self.name_label = Label(self.add_new_contact_frame, text="Name: ")
    self.name_label.grid(row=0, column=0, sticky=W)

    self.name_entry = Entry(self.add_new_contact_frame, font=(NONE, 15), width=20)
    self.name_entry.grid(row=0, column=1, pady=10)

    self.phone_no_label = Label(self.add_new_contact_frame, text="Phone No: ")
    self.phone_no_label.grid(row=1, column=0, sticky=W)

    self.phone_no_entry = Entry(self.add_new_contact_frame, font=(NONE, 15), width=20)
    self.phone_no_entry.grid(row=1, column=1, pady=10)

    self.email_id_label = Label(self.add_new_contact_frame, text="Email Id: ")
    self.email_id_label.grid(row=2, column=0, sticky=W)


    self.email_id_entry = Entry(self.add_new_contact_frame, font=(NONE, 15), width=20)
    self.email_id_entry.grid(row=2, column=1, pady=10)

    self.city_label = Label(self.add_new_contact_frame, text="City: ")
    self.city_label.grid(row=3, column=0, sticky=W)
```

```
    self.city_combobox = Combobox(self.add_new_contact_frame, font=(NONE, 15), width=20,
    values=('Greater Noida', 'Noida', 'Delhi', 'Gurgaon', 'Mumbai'))


    self.city_combobox.grid(row=3, column=1, pady=10)
    self.city_combobox.current(0
    self.profile_pic_label = Label(self.add_new_contact_frame, text="Profile Pic: ")

    self.profile_pic_label.grid(row=4, column=0)

    self.style.configure('TButton', font=(NONE, 15), width=20)

    self.profile_pic_button = Button(self.add_new_contact_frame, text="Choose your profile pic",
    command = self.profile_pic_button_click)
    self.profile_pic_button.grid(row=4, column=1)

    self.add_button = Button(self.add_new_contact_frame, text="Add", command = self.add_button_click)
    self.add_button.grid(row=5, column=1, pady=10)

def profile_pic_button_click(self):
    profile_pic = Image.open(filedialog.askopenfilename())
    self.profile_pic_path = "profile_pics/" + self.email_id_entry.get() + "." + profile_pic.format
    profile_pic.save(self.profile_pic_path)

def add_button_click(self):
    self.cur.execute("select * from Contact where EmailId = ?", (self.email_id_entry.get(),))
    contact = self.cur.fetchone()
    if contact is None:
        self.cur.execute("insert into Contact values(?, ?, ?, ?, ?)", (self.name_entry.get(),
        self.phone_no_entry.get(), self.email_id_entry.get(), self.city_combobox.get(),
        self.profile_pic_path))
        self.con.commit()
        messagebox.showinfo('Success Message', 'Contact details are added successfully')
        self.add_new_contact_frame.destroy()
        self.create_view_all_contacts_frame()
    else:
        messagebox.showerror('Error Message', 'Contact details are already added')

def create_update_delete_contact_frame(self, event):
    contact = self.contacts_treeview.item(self.contacts_treeview.selection())['values']
    self.view_all_contacts_frame.destroy()

    self.update_delete_contact_frame = Frame(self)
```

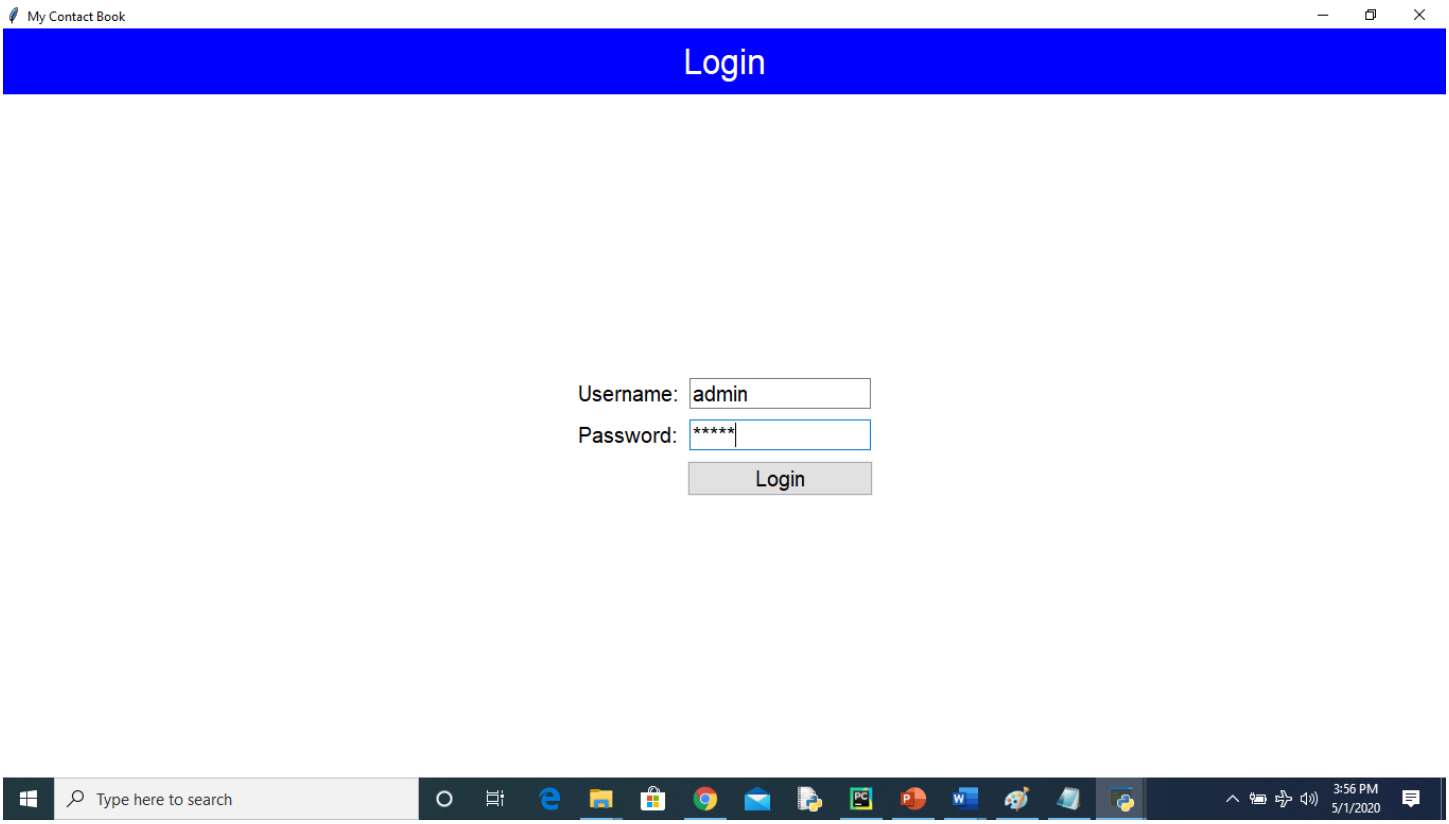
```
self.update_delete_contact_frame.place(relx=.5, rely=.5, anchor=CENTER)

self.style.configure('TLabel', background='white', font=(NONE, 15))

self.name_label = Label(self.update_delete_contact_frame, text="Name: ")
self.name_label.grid(row=0, column=0, sticky=W)

self.name_entry = Entry(self.update_delete_contact_frame, font=(NONE, 15), width=20)
self.name_entry.grid(row=0, column=1, pady=5)
self.name_entry.insert(END, contact[0])


self.phone_no_label = Label(self.update_delete_contact_frame, text="Phone No: ")
self.phone_no_label.grid(row=1, column=0, sticky=W)
self.phone_no_entry = Entry(self.update_delete_contact_frame, font=(NONE, 15), width=20)
self.phone_no_entry.grid(row=1, column=1, pady=5)
self.phone_no_entry.insert(END, contact[1])

self.email_id_label = Label(self.update_delete_contact_frame, text="Email Id: ")
self.email_id_label.grid(row=2, column=0, sticky=W)

self.email_id_entry = Entry(self.update_delete_contact_frame, font=(NONE, 15), width=20)
self.email_id_entry.grid(row=2, column=1, pady=5)
self.email_id_entry.insert(END, contact[2])
self.old_email_id = contact[2]

self.city_label = Label(self.update_delete_contact_frame, text="City: ")
self.city_label.grid(row=3, column=0, sticky=W)

self.city_combobox = Combobox(self.update_delete_contact_frame, font=(NONE, 15), width=20,
values=('Greater Noida', 'Noida', 'Delhi', 'Gurgaon', 'Mumbai'))
self.city_combobox.grid(row=3, column=1, pady=5)
self.city_combobox.set(contact[3])

self.profile_pic_label = Label(self.update_delete_contact_frame, text="Profile Pic: ")
self.profile_pic_label.grid(row=4, column=0)



self.update_delete_contact_frame.image = ImageTk.PhotoImage(Image.open(contact[4]))
self.profile_pic = Label(self.update_delete_contact_frame,
image=self.update_delete_contact_frame.image,
text="Here we will display image")
self.profile_pic.grid(row=4, column=1)
```

```
    self.style.configure('TButton', font=(NONE, 15), width=20)

    self.profile_pic_button = Button(self.update_delete_contact_frame, text="Choose new profile pic",
    command = self.profile_pic_button_click)
    self.profile_pic_button.grid(row=5, column=1, pady=5)

    self.update_button = Button(self.update_delete_contact_frame, text="Update",
    command = self.update_button_click)
    self.update_button.grid(row=6, column=1, pady=5)

    self.delete_button = Button(self.update_delete_contact_frame, text="Delete",
    command = self.delete_button_click)
    self.delete_button.grid(row=7, column=1, pady=5)

def update_button_click(self):
    self.cur.execute("""update Contact set Name = ?, PhoneNumber = ?, EmailId = ?, City = ?,
    ProfilePicPath = ? where EmailId = ?""", (self.name_entry.get(), self.phone_no_entry.get(),
    self.email_id_entry.get(), self.city_combobox.get(), self.profile_pic_path, self.old_email_id))
    self.con.commit()
    messagebox.showinfo('Success Message', 'Contact details are updated successfully')
    self.update_delete_contact_frame.destroy()
    self.create_view_all_contacts_frame()


def delete_button_click(self):
    if messagebox.askquestion('Confirmation Message', 'Are you sure to delete?') == 'yes':
        self.cur.execute("delete from Contact where EmailId = ?", (self.old_email_id,))
        self.con.commit()
        messagebox.showinfo('Success Message', 'Contact details are deleted successfully')
    self.update_delete_contact_frame.destroy()
    self.create_view_all_contacts_frame()

def name_entry_key_released(self, event):
    self.cur.execute("select * from Contact where Name like ?", ('%' + self.name_entry.get() + '%', ))
    self.fill_contacts_treeview()

def fill_contacts_treeview(self):
    for contact in self.contacts_treeview.get_children():
        self.contacts_treeview.delete(contact)

    contacts = self.cur.fetchall()
    for contact in contacts:
        self.contacts_treeview.insert("", END, values=contact)
```

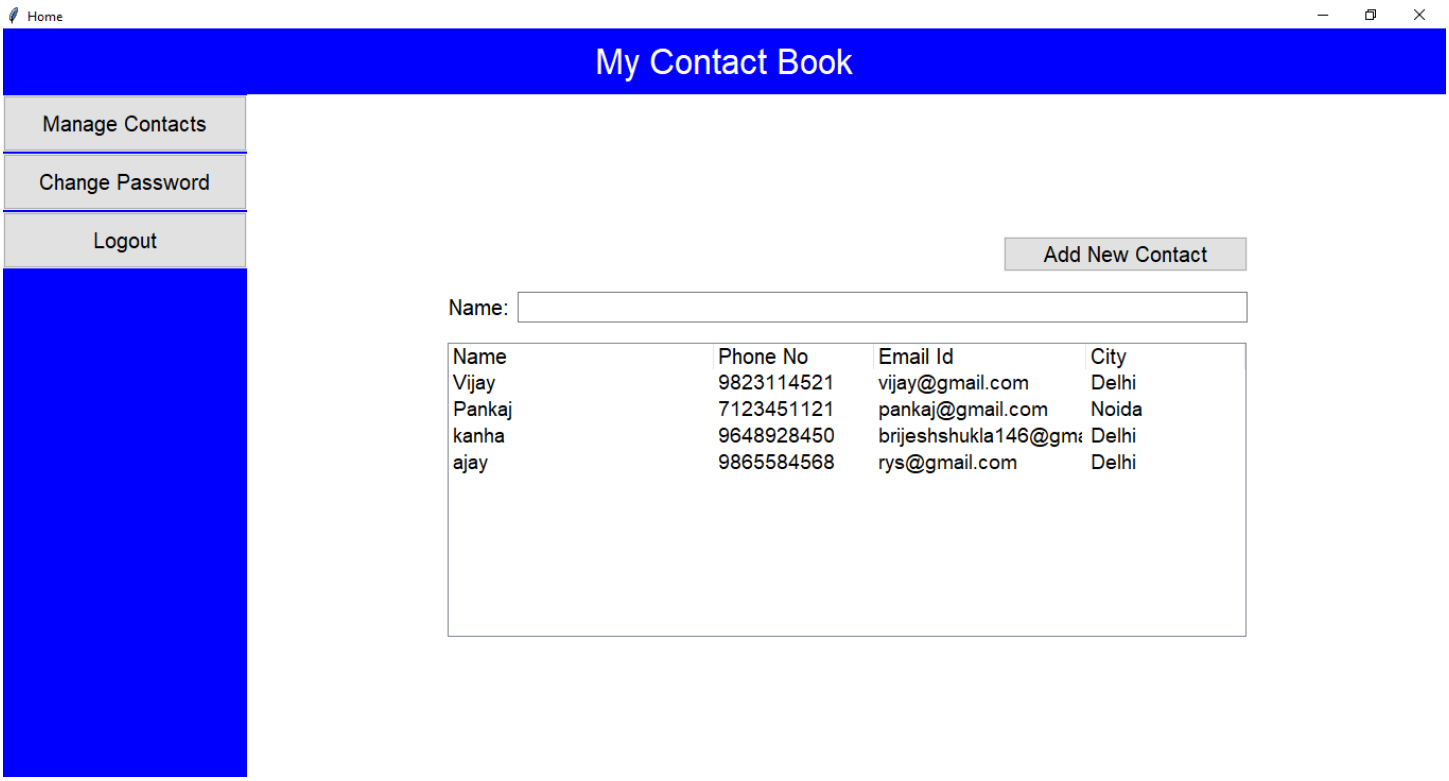
## B. SCREEN SHOTS

**SNAPSHOTS**

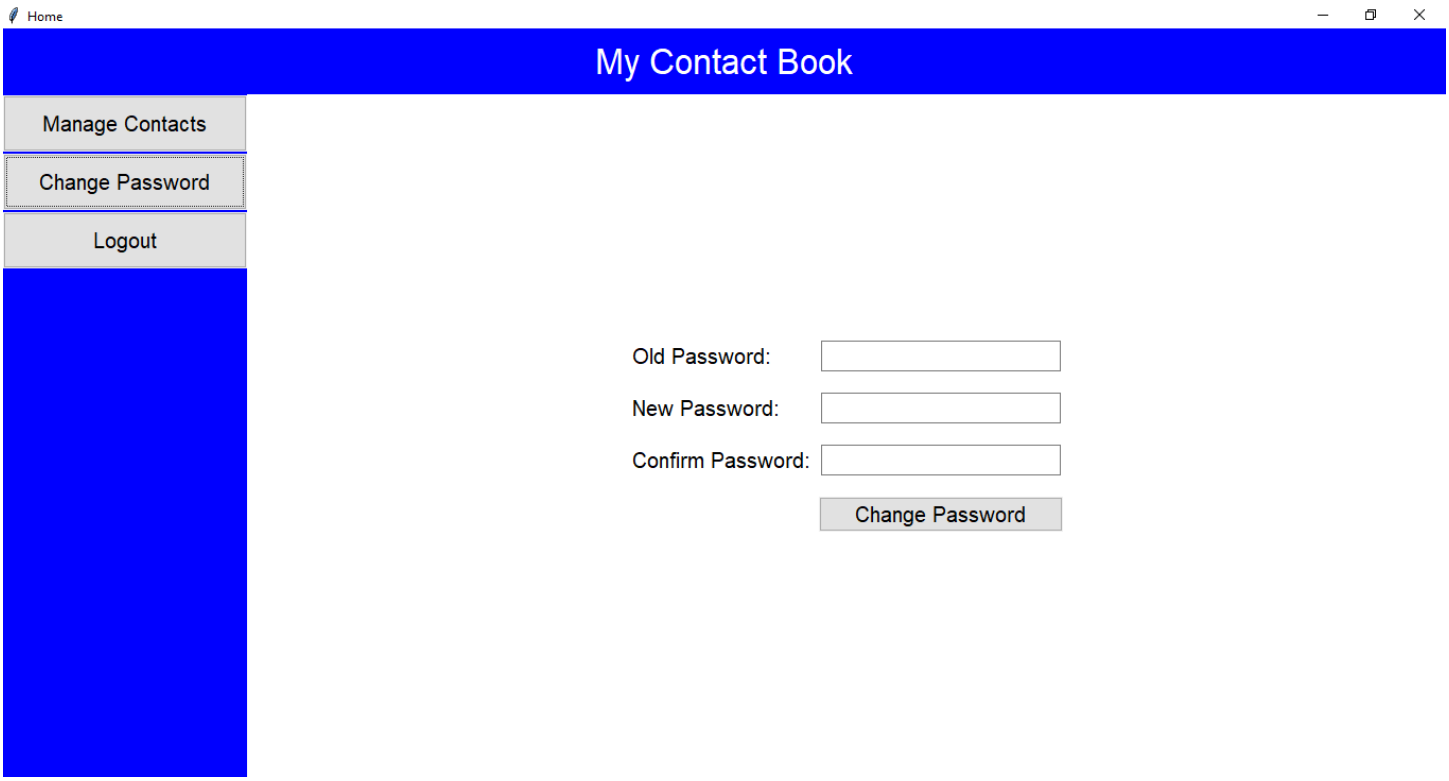
1.Login form:



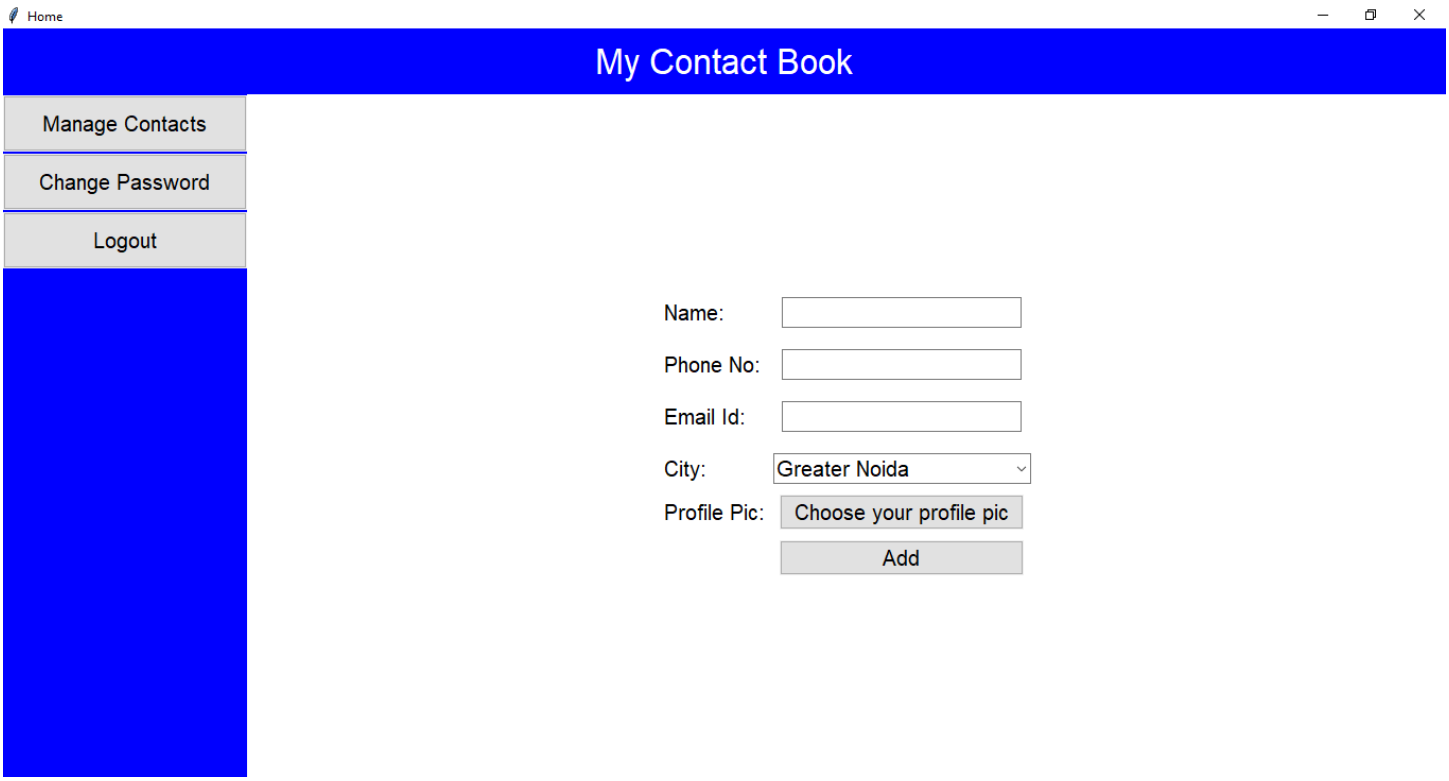
2. Home Page:


Home
My Contact Book
Manage Contacts
Change Password
Logout
Add New Contact
Name:
Name
Phone No
Email Id
City
Vijay
9823114521
vijay@gmail.com
Delhi
Pankaj
7123451121
pankaj@gmail.com
Noida
kanha
9648928450
brijeshshukla146@gma
Delhi
ajay
9865584568
rys@gmail.com
Delhi

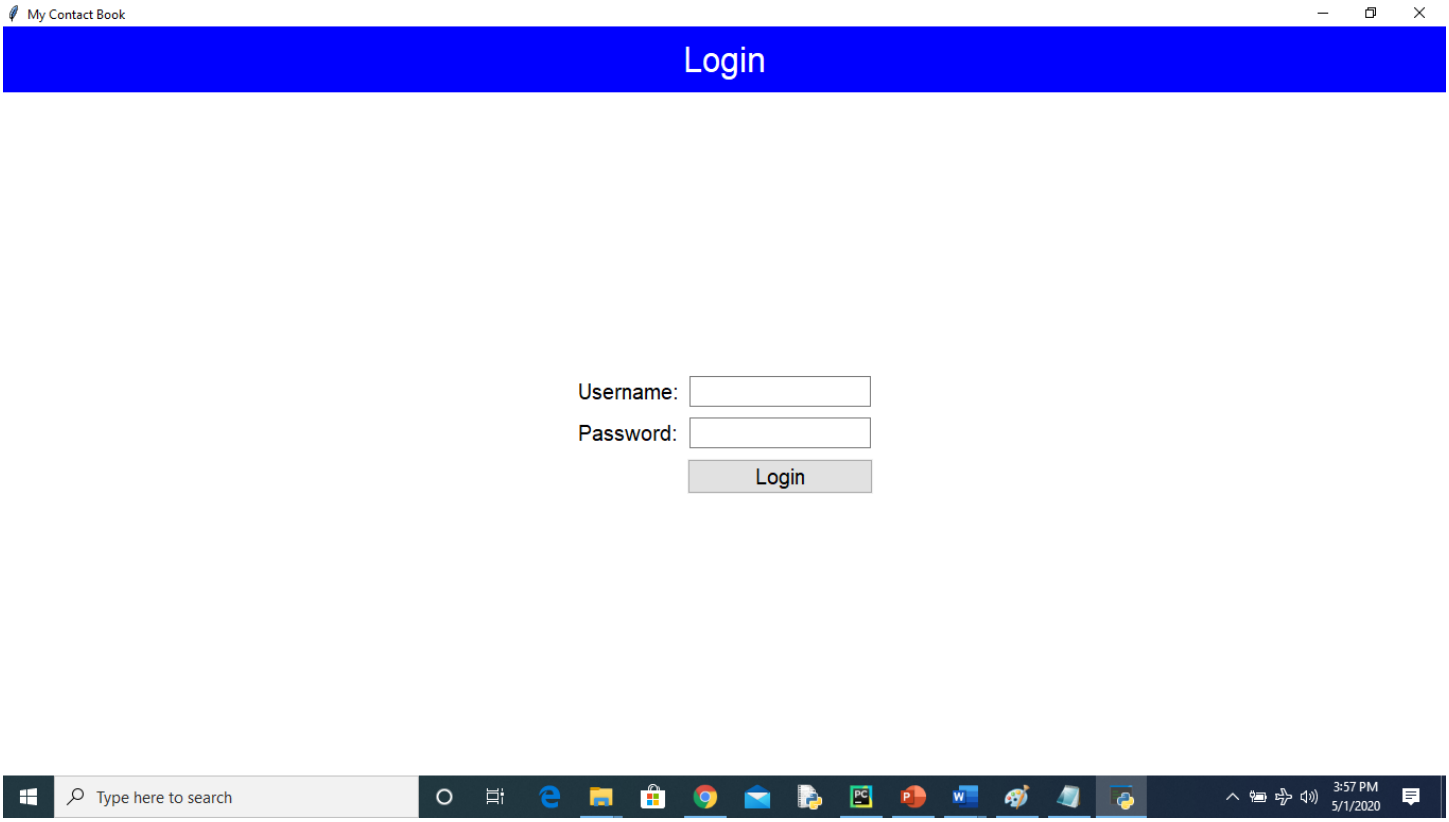
3.Change password:



3. Add New Contact:



4. Logout Frame:



## RESULT & DISCUSSION

In this project finally the data stored and also it will save where you want to see for future and remove the all kind of miss happening about the data because the saved data has been saved in computer as well as database of the system.

The save data you can see it this system

| Name | Phone No | Email Id | City |
|---|---|---|---|
| Vijay | 9823114521 | vijay@gmail.com | Delhi |
| Pankaj | 7123451121 | pankaj@gmail.com | Noida |
| kanha | 9648928450 | brijeshshukla146@gma | Delhi |
| ajay | 9865584568 | rys@gmail.com | Delhi |

## REFERENCES


1. Rogerio Pontes, Francisco Maia, Joao˜ Paulo, and Ricardo Manuel Pereira Vilac¸a. Saferegions: Performance evaluation of multi-party protocols on hbase. In *35th IEEE Symposium on Reliable Distributed Systems Workshops, SRDS 2016 Workshop, Budapest, Hungary, September 26, 2016*, pages 31–36, 2016.
2. Karen Scarfone and Murgiah Souppaya National Institute of Standard and Technology Special Publication 800-118(Draft), 38 pages (Apr. 2009).
3. Improving Data Protection with McAfee Drive Encryption, IT@Intel White Paper (Nov. 2013).

**WEBSITE: -**

4. www.google.com
5. www.slideshare.net
6. www.wikipedia.org