

# Computer science Project

Session: 2024-2025

**A Project Report On “Contact Book”**

Submitted by-

Sneha Dubey

Class – XII (Science A)

Under the Guidance of-

Mrs. Bhavnita Postaria

PGT (Computer Science)

# Contents

# 

|  |  |  |
| --- | --- | --- |
| S.No. | Description | Pg.No. |
| 1. | Certificate | 3 |
| 2. | Acknowledgement | 4 |
| 3. | Introduction | 5 |
| 4. | Python Overview | 6 |
| 5. | Features of Python | 7 |
| 6. | Advantages of Python / How to Install Python | 8 |
| 7. | About MySQL / Installing MySQL | 9 |
| 8. | Advantages of Contact Book | 10 |
| 9. | Objective of the Project | 11 |
| 10. | Proposed System | 13 |
| 11. | Primary Aim | 14 |
| 12. | Secondary Aim | 15 |
| 13. | Source Code | 17 |
| 14. | Output Screen | 22 |
| 15. | Limitations and Future Scope | 31 |
| 16. | Bibliography | 32 |
| 17. | Computer Software & Hardware Requirements | 33 |
| 18. | Thank You!! | 34 |

# Certificate

This is to certify that **Sneha Dubey** student of class XII (Science) has successfully prepared the report on the Project entitled **“Contact Book”** under the guidance of **Mrs. Bhavnita Postaria** (PGT Computer Science). The report is the result of his efforts & endeavors. The report is found worthy of acceptance as final Project report for the subject Computer Science of class XII (Science).

Signature of Internal Examiner Signature of External Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Principal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Acknowledgement

I would like to express a deep sense of thanks and gratitude to my project guide Mrs. Bhavnita Postaria for guiding me immensely through the course of the project. She always evinced keen interest in my project. Her constructive advice & constant motivation have been responsible for the successful completion of this project.

My sincere thank goes to Mrs.Lincy Pancholi (Principal) madam for her coordination in extending every possible support for the completion of this project.

I must thanks to my classmates for their timely help and support for completion of this project.

Last but not the least, I would like to thank all those who had helped directly or indirectly towards the completion of this project.

**Sneha Dubey**

**Class- XII (Science)**

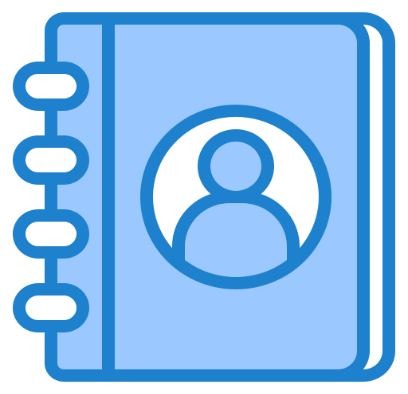
# Introduction

A contact book, also known as an address book, is a tool used to store and manage personal and professional contact information. It typically includes details such as names, phone numbers, email addresses, and physical addresses. While traditionally a physical notebook, contact books have evolved into digital tools accessible on smartphones, computers, and online platforms.

Modern contact books offer advanced features beyond basic information storage. They allow users to add more details like birthdays, notes, and social media profiles. Contacts can also be grouped into categories (e.g., family, work, friends) for easier organization. Many digital contact books, such as Google Contacts or Microsoft Outlook, integrate with other applications, enabling seamless communication through email, phone calls, or messaging directly from the app.

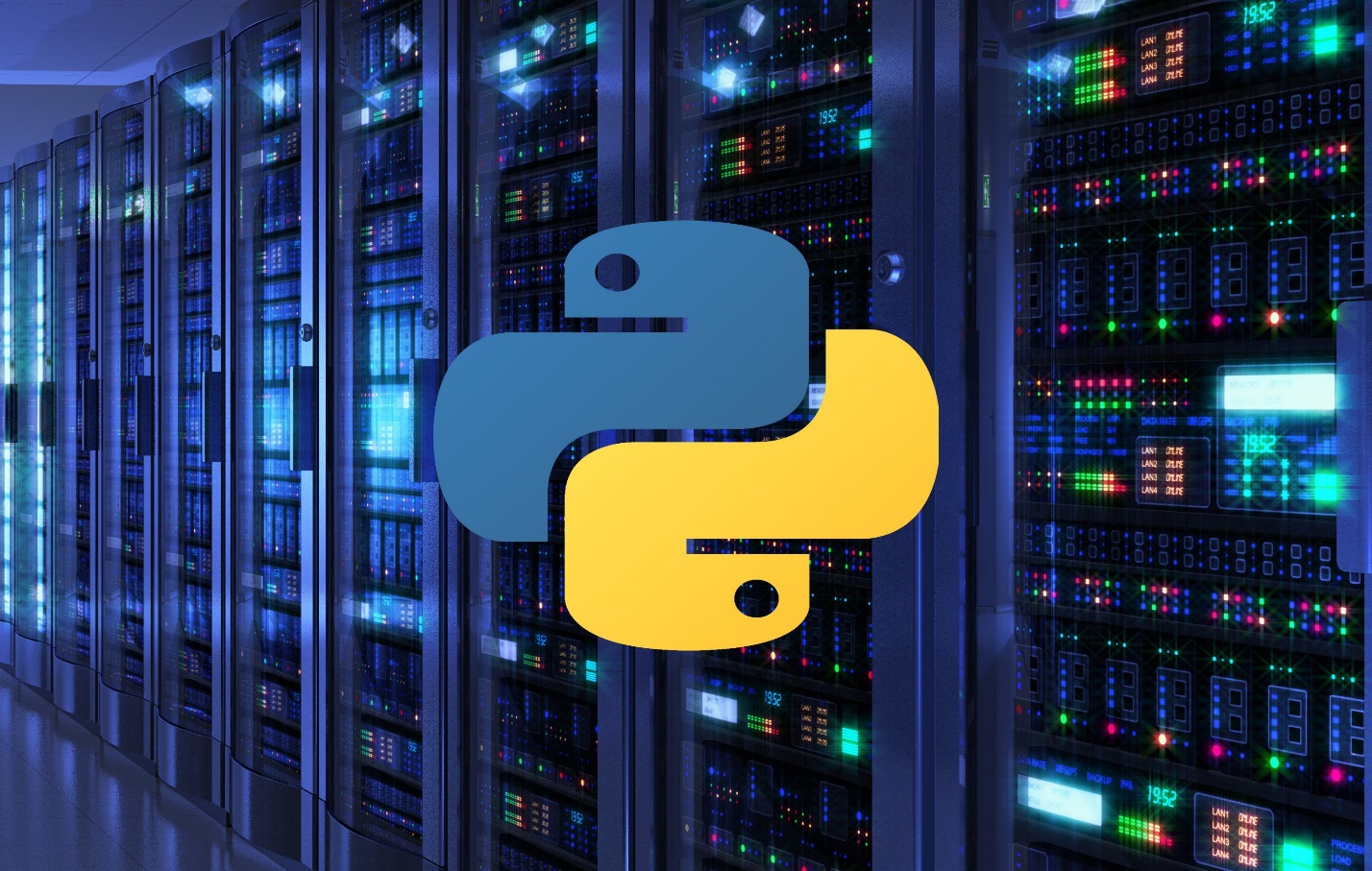
One key advantage of digital contact books is synchronization, which keeps information updated across devices and platforms. Cloud-based systems provide backup and ensure access to contacts from anywhere with an internet connection.

In both personal and professional settings, contact books are essential tools for managing relationships and communication. They make it easier to keep track of important contacts and details, enhancing efficiency and organization in our daily interactions.



# Python Overview

Python is a general-purpose high-level programming language. It is an open source language, released under a GPL-Compatible license. Python Software Foundation (PSF), a non-profit organization, holds the copy-right of python. Guido Van Rossum conceived python in the late 1980s. It was released in 1991 at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language. He named this language after a popular comedy show called “Monty Python’s Flying Circus” (and not after python - the snake). In the last few years, its popularity has increased immensely. According to stackoverflow.com ’s recent survey, python is in the top ten most popular technologies in 2018. It is also dynamically-typed because it carries out type-checking at run time. It does so to make sure that the type of construct matches what we except it to be. The distinctive feature of python is that it is an interpreted language. The Python IDLE (Integrated Development & Learning Environment) executes instruction one line at a time. The python programming language is one of the richest languages.



*Features of python*

**1.Easy:**  Python is a very easy to learn and understand; using this python tutorial, any beginner can understand the basics of python.

1. **Interpreted:** It is interpreted (executed) line by line. This makes it easy to test and debug.

1. **Object** **Oriented:** The python programming language supports classes and objects.

1. **Free and Open-Source:** The language and its source code are available to the public for free; there is no need to buy a costly license.

1. **Portable**: Since it is open source, we can run python on windows, mac, Linux, or any other platforms. Our programs will work without needing to the changed for every machine.

1. **GUI (Graphical User Interface) programming:** We can use it to develop a GUI (Graphical User Interface).

1. **Large Library:** Python provides us with a large standard library. We can use it to implement a variety of functions without needing to reinvent the wheel every time. Just pick the code we need and continue. This lets us to focus on other important tasks.

**Advantages of Python-**

**1. Extensible**

**2. Portable**

## 3. Free & Open-Source

## 4. Readable

## 5. Embeddable

## 6. Improved Productivity

## 7. Simple and Easy

## 8. Object Oriented

## 9. Interpreted

## 10. Extensive Libraries

## How to install Python-

Point your web browser to the download page on the Python website [(www.python.org)](http://www.python.org/)

1. Select the latest Windows x86 MSI Installer and click on the link to download the .msi installer.
2. Run the installer.
3. Select “Install for all users” and click the Next > button.
4. Keep the default option (C:\Python32\) as the destination directory and click Next > again.
5. Don’t make any changes in the “Customize Python 3.2.3” dialog, just click Next > again.
6. Click Yes if asked if this program should be allowed to install software on your system.
7. Click the Finish button when installation completes.

## About MySQL-

MySql is a fast, easy to use RDBMS (Relational Database Management System) being used for many small and big businesses. MySql is developed, marketed and supported by MySQL AB, which is a Swedish Company.

MySql is becoming is so popular because of many good reasons.

1. MySql is released under an open-source language. So, we have nothing to pay to use it.
2. MySql is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
3. MySql uses a standard form of the well-known SQL data language.
4. MySql works on many operating system and with many languages including PHP, PERL, C, C++, JAVA, etc.
5. MySql works very quickly and works well even with large data sets.
6. MySql is very friendly to PHP, the most appreciated language for web development.
7. MySql supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4 GB, but we can increase this (if our operating system can handle) to a theoretical limit of 8 million Tera-Bytes (TB).
8. MySql is customizable. The open-source GPL license allows programmers to modify the MySql software to fit their own specific environments. **For installing MySQL-**

1.For Python 2.7 or lower, install using pip aspip install mysql. connector

2. For Python 3 or higher version, install using pip3 as pip3 install mysql. Connector



**Advantages of “Contact Book”-**

1. **Quick Search**: Instant access to contacts with keywords and filters.
2. **Large Capacity**: Stores thousands of contacts without taking space.
3. **Integration**: Syncs with email, calendars, and apps.
4. **Backup**: Automatic cloud backups protect against data loss.
5. **Accessible Anywhere**: Available on multiple devices, anytime.
6. **Detailed Info**: Stores more than just phone numbers—emails, notes, etc.

**General:**

* **Time-Saving**: Organized contacts reduce search time.
* **Efficient Communication**: Easy to reach contacts quickly.
* **Professional**: Improves response times and networking.

# Objective of the project-

The objective of a "Contact Book" project using Python and MySQL is to develop a simple yet functional application that allows users to manage and store contact information (such as names, phone numbers, email addresses, etc.) in a structured and persistent manner using a database system. The project helps to demonstrate and practice key concepts in Python programming, database management with MySQL, and how to create an interactive user interface (CLI or GUI) for interacting with the system.

**Key Objectives of the Contact Book Project:**

**1.Data Storage and Retrieval:**

* Store user contact details in a relational database (MySQL).
* Retrieve, update, and delete contact information from the database using SQL queries.

**2.CRUD Operations**:

* Create: Add new contacts to the database.
* Read: View and search for existing contacts by various criteria (e.g., name, phone number).
* Update: Edit existing contact details.
* Delete: Remove contacts from the database.

**3.Database Design:**

* Design and implement a MySQL database with a table (e.g., contacts) to hold the contact information.
* Define relationships and constraints (e.g., ensuring unique phone numbers or email addresses).

**4.User Interface (UI):**

* Implement a command-line interface (CLI) or a basic graphical user interface (GUI) for interacting with the contact book.
* Allow users to add, view, search, and delete contacts through simple commands or buttons.

**5.Data Validation and Error Handling:**

* Implement basic input validation (e.g., checking for valid phone numbers, email formats).
* Handle exceptions like database connection errors, invalid input, and other edge cases.

**6.Persistence:**

* Ensure that data is stored in the MySQL database so that it persists even after the program is closed or restarted.
* Demonstrate knowledge of SQL and how to use Python libraries like mysql-connector or SQLAlchemy to interact with the database.

**7.Search and Filtering:**

* Provide features to search contacts by name, phone number, or email, and filter or sort results based on certain criteria.

**Possible Features:**

* Add Contact: Add a new contact to the database with details such as name, phone number, email, and address.
* Search Contact: Search for contacts based on criteria (e.g., name, phone number).
* Update Contact: Edit contact details like name, phone number, or email.
* Delete Contact: Remove a contact from the system.
* View All Contacts: Display all contacts stored in the database.
* Advanced Search: Implement search options that allow searching by specific fields, such as email or phone number.

**Technologies Involved:**

* Python: Used for writing the logic, interacting with the database, and providing a user interface.
* MySQL: Used for creating a relational database to store the contacts data.
* MySQL Connector for Python (mysql-connector-python or PyMySQL): Used to establish a connection between Python and the MySQL database to execute queries and manage data.

**Learning Outcomes:**

* Hands-on experience with MySQL and relational database management systems (RDBMS).
* Practice writing SQL queries (SELECT, INSERT, UPDATE, DELETE).
* Developing Python skills for integrating with external databases.
* Understanding how to build a simple application with CRUD functionality.
* Designing a user-friendly interface for managing contacts.

## Proposed System

The Contact Book project with Python and MySQL aims to create a simple yet effective application for managing contact information in a structured, persistent, and easily accessible way. This project leverages Python as the programming language for logic and user interaction, while MySQL serves as the relational database management system to store and manage the contact data.

**Primary Aims:**

1.Manage Contacts Efficiently:

* To develop a system that allows users to easily store, view, update, search, and delete contact information (such as name, phone number, email, and address).
* The system should organize contacts in a structured format, allowing users to efficiently manage and retrieve their contacts as needed.

2.Persistence of Data:

* To ensure that all contacts are stored persistently in a MySQL database, allowing data to remain available even after the application is closed or restarted.
* Users should be able to interact with the system across multiple sessions without data loss.

3.Practical Application of Python and MySQL:

* To provide hands-on experience with Python programming by working on real-world scenarios.
* To demonstrate the integration of Python with a MySQL database using tools like MySQL Connector to handle data storage and retrieval.
* To help the user understand CRUD operations (Create, Read, Update, Delete) within the context of a relational database.

4.Understanding Relational Database Design:

* To implement a relational database structure for storing contacts, where data is normalized and organized into tables with relationships.
* Users will gain an understanding of SQL queries (e.g., SELECT, INSERT, UPDATE, DELETE) and how they interact with a database.

5.User Interface Development:

* To provide a user-friendly interface for managing contacts. This could be a command-line interface (CLI) or a graphical user interface (GUI).
* Through this, users can easily add, search, and modify their contacts without interacting directly with the database.

**Real-World Use Case:**

The project mirrors a real-world scenario where people need to keep track of their contacts (friends, family, colleagues, etc.) and perform operations like updating contact details or deleting obsolete information.

**Data Validation and Error Handling:**

To implement input validation (e.g., checking for valid phone numbers, unique emails) and error handling (e.g., preventing SQL injection, managing database connection errors) to ensure the application works robustly.

**Secondary Aims:**

* Security and Data Integrity: Ensure that only valid data is entered into the database and implement basic error handling to manage edge cases.
* Extendibility: Provide a foundation that can be expanded with additional features in the future (e.g., backup/restore functionality, user authentication, cloud storage integration).
* Learning and Skill Development: Help the developer (or user) to gain practical experience in Python, database management, and building real-world applications.

**Project Scope:**

The contact book will have the following core functionalities:

* Create Contact: Allows users to add a new contact to the database.
* Read/View Contacts: Users can view all contacts or search for specific contacts based on name, phone number, or email.
* Update Contact: Users can update a contact’s information, such as phone number or email.
* Delete Contact: Users can delete a contact from the system.
* Search/Filter Contacts: Users can search or filter contacts by name, phone number, or email.
* Data Persistence: All contacts are saved in a MySQL database to persist data across sessions.

**In Summary:**

The aim of the Contact Book project is to develop a functional, real-world application that allows users to efficiently manage contact information with Python and MySQL. By integrating a relational database, the project demonstrates the importance of data persistence and provides practical experience in building CRUD-based applications. It also teaches key skills in database interaction, error handling, and user interface design.

****

# SOURCE CODE

#Create a database and table in MySQL

CREATE DATABASE contact\_book;

USE contact\_book;

CREATE TABLE contacts (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

phone VARCHAR(15),

email VARCHAR(100) );

#Intall pip using command prompt

pip install mysql-connector-python

#Python code

import mysql.connector

# Function to connect to the database

def create\_connection():

connection = mysql.connector.connect(

host='localhost',

user='root', # replace with your MySQL username

password='root', # replace with your MySQL password

database='contact\_book'

)

return connection

# Function to add a contact

def add\_contact(name, email, phone):

connection = create\_connection()

cursor = connection.cursor()

cursor.execute("INSERT INTO contacts (name, email, phone) VALUES (%s, %s, %s)", (name, email, phone))

connection.commit()

cursor.close()

connection.close()

print("Contact added successfully!")

# Function to view contacts

def view\_contacts():

connection = create\_connection()

cursor = connection.cursor()

cursor.execute("SELECT \* FROM contacts")

contacts = cursor.fetchall()

cursor.close()

connection.close()

return contacts

# Function to delete a contact

def delete\_contact(contact\_id):

connection = create\_connection()

cursor = connection.cursor()

cursor.execute("DELETE FROM contacts WHERE id = %s", (contact\_id,))

connection.commit()

cursor.close()

connection.close()

print("Contact deleted successfully!")

# Main program

def main():

while True:

print("\nContact Book")

print("1. Add Contact")

print("2. View Contacts")

print("3. Delete Contact")

print("4. Exit")

choice = input("Choose an option: ")

if choice == '1':

name = input("Enter name: ")

email = input("Enter email: ")

phone = input("Enter phone: ")

add\_contact(name, email, phone)

elif choice == '2':

contacts = view\_contacts()

print("\nContacts:")

for contact in contacts:

print(f"ID: {contact[0]}, Name: {contact[1]}, Email: {contact[2]}, Phone: {contact[3]}")

elif choice == '3':

contact\_id = int(input("Enter contact ID to delete: "))

delete\_contact(contact\_id)

elif choice == '4':

break

else:

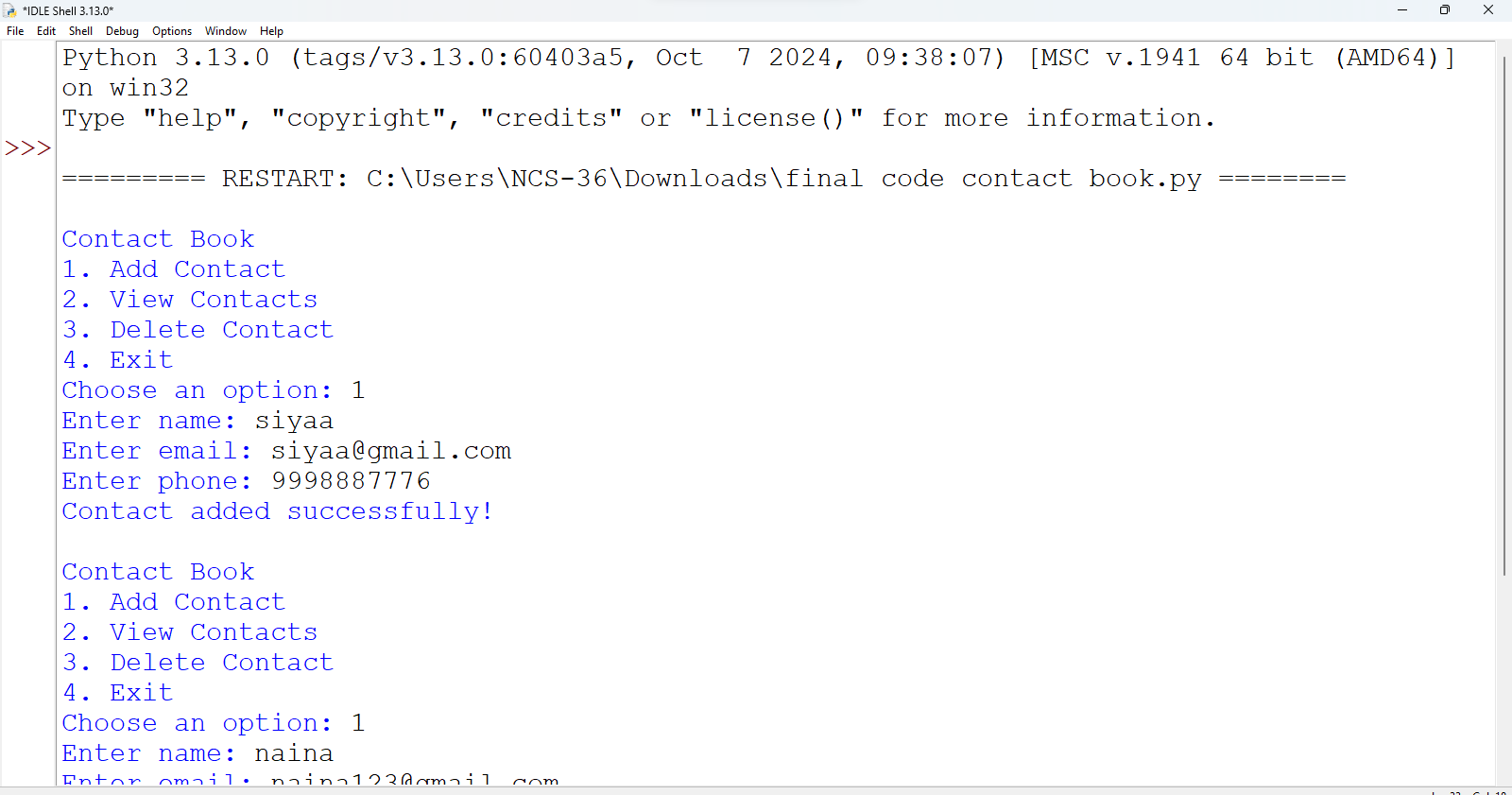
print("Invalid option. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

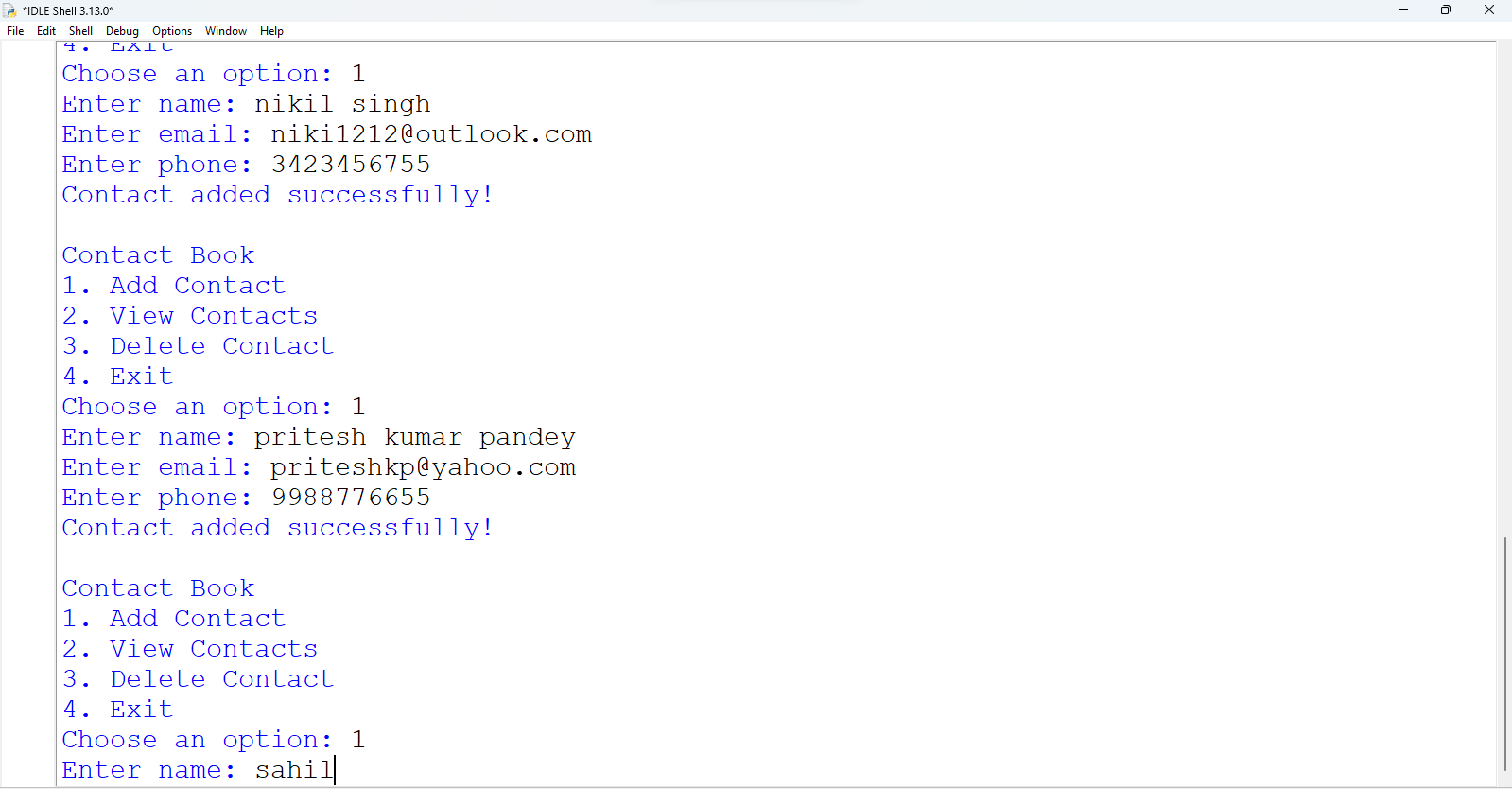
python contact\_book.py

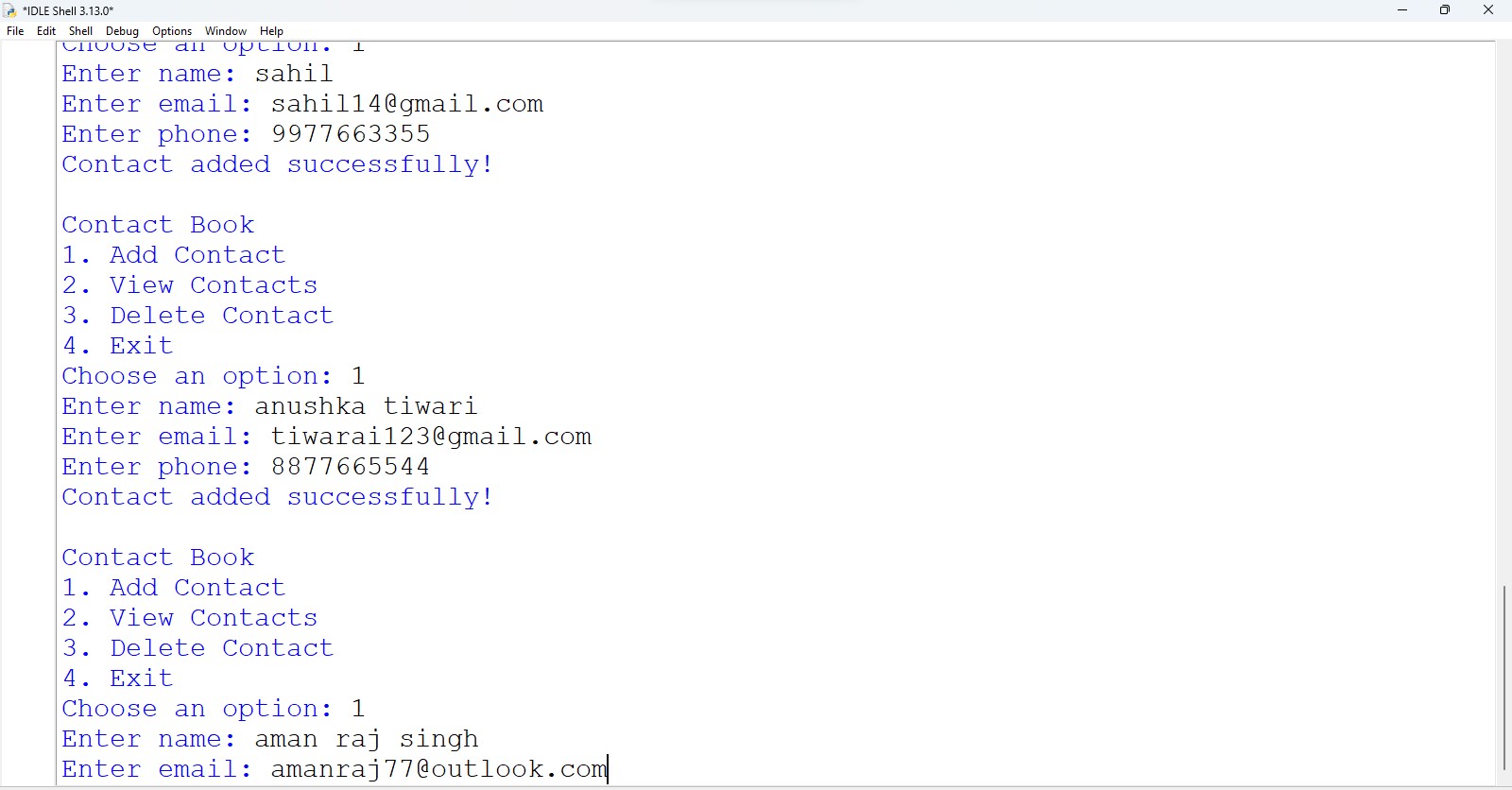
# Output Screens





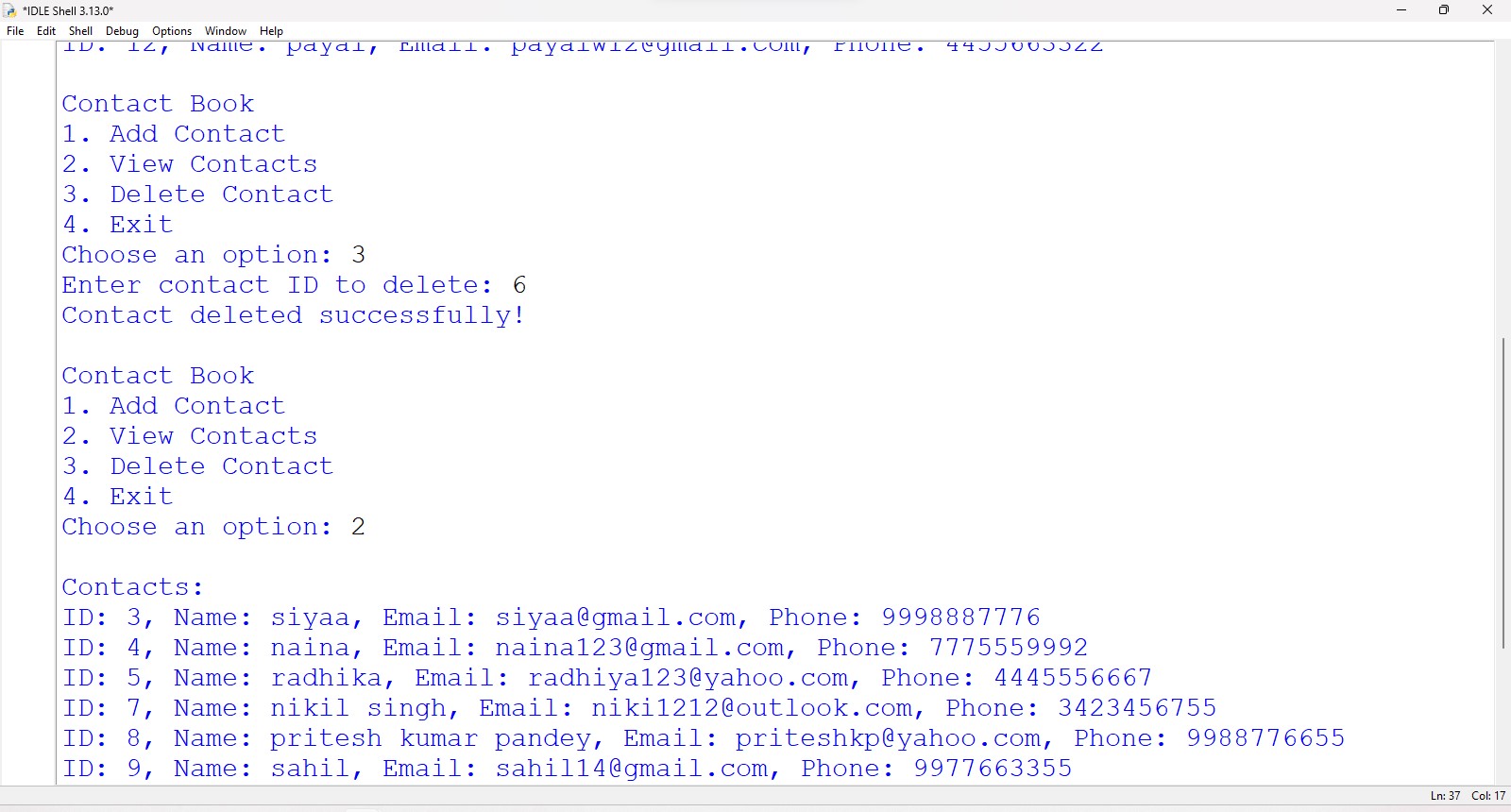


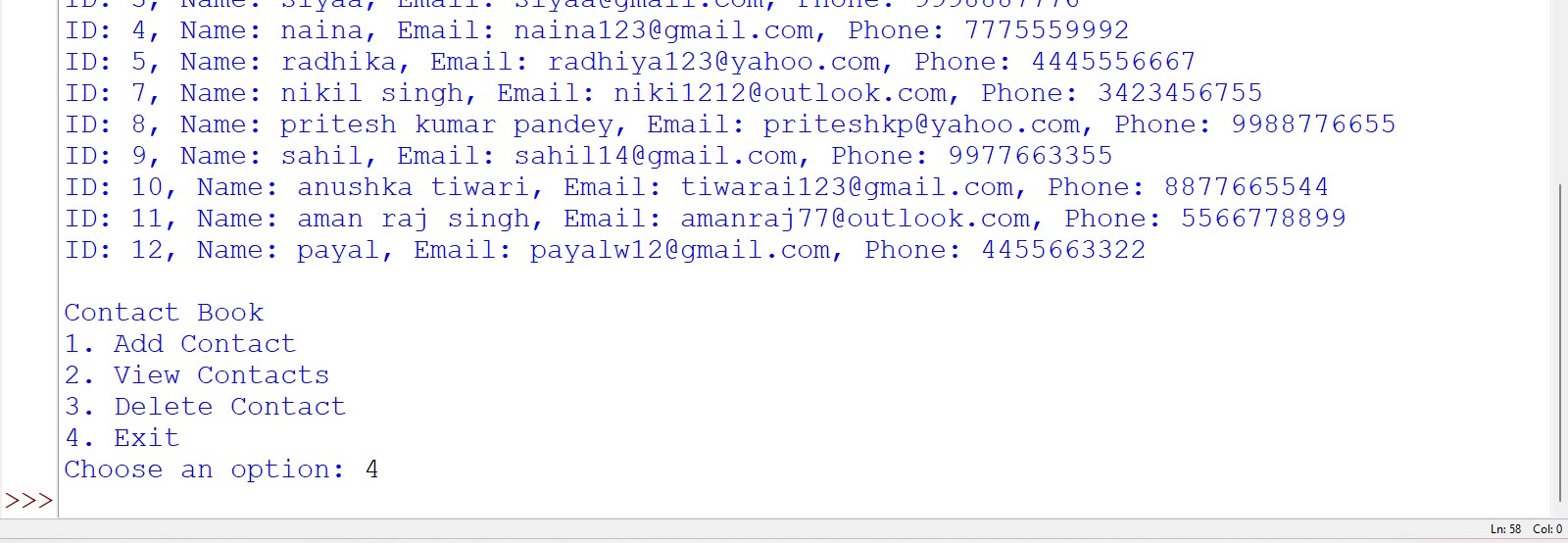












# Scope

## Limitations: -

* Technology Dependence: Requires devices and internet access.
* Security Risks: Vulnerable to hacking and data breaches.
* Data Loss: Risk of losing contacts without proper backups.
* User Error: Accidental deletion or mismanagement of contacts.
* Compatibility Issues: Poor syncing between different applications.
* Limited Features: May lack advanced functions like tagging.
* Over-Reliance: Users may struggle to remember contacts without it.

## Future Scope: -

* AI Integration: Smart management and reminders.
* Enhanced Security: Better encryption and privacy features.
* Cross-Platform Syncing: Seamless access across devices.
* Advanced Search: Intuitive search using natural language.
* Social Media Integration: Automatic updates from social profiles.
* Voice Assistants: Hands-free management.
* Data Analytics: Insights into communication trends.
* Customizable Features: User-defined fields and templates.
* Collaboration Tools: Secure sharing within teams.
* VR/AR Visualization: Immersive contact management experiences.

# Bibliography

1. IDLE Python 3.8.5
2. Pyforschool
3. cbsepython.in
4. [www.google.com](http://www.google.com/)
5. Computer Science with Python by Sumita Arora Class XII (Book)
6. Preeti arora class 12 computer science

# Computer Software & Hardware Requirements

## Hardware required: -

Windows 10 or 11

64-bit operating system, x64-based processor

Intel(R) Core (TM) i3-10100 CPU @ 3.60GHz 3.60 GHz)

Printer for printing project

## Software required: -

Windows 11 Pro

MS word, for preparing documentation

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32

*THANK YOU!!*