# A

**Project Report On**

## Cab Booking & Management System

**By:** Kayum Parmar Semester – 6

# Project Guide:

PROF.



**Submitted To:** Geetanjali College of Computer Science and Commerce Rajkot.

# Academic Year:

2024-2025.

**ACKNOWLEDGEMENT**

I Am Happy to Submit My Idea Of "**Age Calculator in Python** " Application In Saurashtra University, Rajkot For BCA Degree In Computer Branch.

We take this occasion to thank God, almighty for blessing us with his grace and taking our Endeavour to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, PROF. ………………. for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank friends for the support and encouragement they have given us during the course of our work.

# Index

|  |  |
| --- | --- |
| * Project Profile | 01 |
| * Introduction | 02 |
| * Software Environment | 05 |
| * Working Environment | 07 |
| * Diagrams | 08 |
| * Data Dictionary | 11 |
| * Screen Shots | 14 |
| * Test Cases | 28 |
| * Implementation and Evaluation | 29 |
| * Future Enhancement | 30 |
| * Webliography | 31 |

**PROJECT PROFILE**

|  |  |
| --- | --- |
| Project Title | Age Calculator in Python |
| Development Software | VS Code |
| Project Language | Python |
| Documentation Tool | Microsoft Word |
| Operating System | Windows |
| Academic Year | 2024 – 2025 |
| Developed By | Gaurav Davare |
| Submitted To | Geetanjali College |

1. **Introduction**

Age Calculator is a simple Python-based application that calculates a user's age based on their date of birth. This project demonstrates fundamental programming concepts, such as working with date and time, user input handling, and simple GUI development. It is designed to provide an accurate age calculation in years, months, and days, considering leap years and different month lengths.

This project is particularly useful for applications that require age verification, such as online registrations, job applications, and health services. By implementing both a console-based and graphical user interface (GUI) version, the project aims to enhance user experience and accessibility. Additionally, the project serves as a stepping stone for beginners to understand how to integrate Python's built-in libraries with real-world applications.

**2. Objectives**

* To develop a Python application that calculates the age of a person.
* To enhance understanding of date and time manipulation in Python.
* To implement a user-friendly interface using Tkinter (optional).
* To improve problem-solving skills in programming.

**3. Software and Tools Used**

* **Programming Language:** Python 3.x
* **Libraries:**
* datetime (for date calculations)
* tkinter (for GUI development, optional)
* **Development Environment:** Any Python IDE (e.g., PyCharm, VS Code, Jupyter Notebook)

**4. Implementation**

**4.1 Algorithm**

1. Take the user's date of birth as input.
2. Retrieve the current date.
3. Calculate the difference between the current date and the birth date.
4. Extract the years, months, and days to determine the exact age.
5. Display the age as output.

**4.2 Code Implementation (Console-based)**

from datetime import datetime

def calculate\_age(dob):

today = datetime.today()

dob = datetime.strptime(dob, "%Y-%m-%d")

age = today.year - dob.year - ((today.month, today.day) < (dob.month, dob.day))

return age

# User input

dob = input("Enter your date of birth (YYYY-MM-DD): ")

age = calculate\_age(dob)

print(f"You are {age} years old.")

**4.3 GUI-based Implementation (Using Tkinter)**

**5. Features**

* Accepts date of birth input in YYYY-MM-DD format.
* Computes age accurately, considering leap years and different month lengths.
* Provides a console-based and GUI-based approach for user interaction.
* Simple and lightweight application with minimal dependencies.

**6. Applications**

* Can be used in online forms for age verification.
* Useful in schools, hospitals, and businesses for quick age calculations.
* Integrated into larger applications requiring date-based calculations.

**7. Future Enhancements**

* Adding a feature to calculate age in months and days for more precision.
* Allowing users to select their birth date using a date picker.
* Implementing a mobile-friendly web version using Flask or Django.
* Adding multi-language support for better accessibility.

**8. Conclusion**

The Age Calculator project demonstrates the application of Python's datetime module and GUI development using Tkinter. It serves as a beginner-friendly project to understand fundamental programming concepts, user input handling, and simple UI creation.

**References**

* Python Official Documentation: <https://docs.python.org/3/>
* Tkinter GUI Guide: <https://docs.python.org/3/library/tkinter.html>