# In-House Training

# REPORT

***Submitted by***

## ADITYA RAJ (23BCS80083)

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

## BACHELOR OF ENGINEERING

**IN**

COMPUTER SCIENCE ENGINEERING



**Chandigarh University**

2024



**BONAFIDE CERTIFICATE**

Certified that this Mini project report **“ ”** is the bonafide work of “**Aditya Raj”** who carried out the project work under my/our supervision.

**SIGNATURE**

**HEAD OF THE DEPARTMENT**

**SIGNATURE**

**SUPERVISOR**

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## TABLE OF CONTENTS

[List of Figures](#_bookmark0) 4

[List of Tables](#_bookmark1) 4

[Abstract… 5](#_bookmark2)

CHAPTER 1. INTRODUCTION 6

1.1. Important Terms 6

CHAPTER 2. PROBLEM STATEMENT… 7

2.1. Approaches 8

CHAPTER 3. METHODOLOGY 9

3.1. Software Used 10

CHAPTER 4. LITERATURE REVIEW 11

CHAPTER 5. CONCLUSION 12

[REFERENCES 13](#_bookmark3)

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who contributed to the completion of this project. First and foremost, we extend our heartfelt appreciation to our college staff and class representatives for their invaluable guidance and support throughout the project.

Special thanks are due to our Python teacher, **Neha Kapur ma’am (E13446)**, whose expertise, encouragement, and coordination were instrumental in the successful completion of this project. We also wish to thank our classmates for their collaboration, feedback, and assistance during various stages of the project development.

Furthermore, we are grateful to the Python community for providing extensive documentation, tutorials, and resources that facilitated our learning and implementation of Tkinter GUI programming. Last but not least, we express our gratitude to our friends and family for their understanding, encouragement, and motivation throughout the project journey.

Thank you, everyone, for your unwavering support and encouragement.

### List of Figures

**Figure 1 7**

**Figure 2** **10**

**List of Tables**

**Table 1 14**

# ABSTRACT

This report presents the development process of an Age Calculator application using Tkinter, a widely-used GUI toolkit for Python. The application provides users with a straightforward interface to input their birthdate and obtain their current age, leveraging the flexibility and simplicity of Tkinter widgets. The report begins with an overview of Tkinter's key features and advantages for GUI development, highlighting its cross-platform compatibility and extensive documentation.

The development process is structured into several key components, starting with the design of the user interface. Tkinter's grid layout manager is utilized to arrange widgets such as labels, entry fields, buttons, and message boxes in a visually appealing and intuitive manner. The report delves into the implementation details, discussing how each widget is instantiated, configured, and placed within the GUI window.

The core functionality of the Age Calculator application revolves around accurately computing the user's age based on their input birthdate. This involves parsing the input string, converting it into a date object, and calculating the time difference between the birthdate and the current date. The report elucidates the algorithmic steps involved in this process, addressing potential edge cases such as leap years and invalid input formats.

## INTRODUCTION

Age Calculator using Tkinter in Python is a graphical user interface (GUI) application designed to calculate a person's age based on their date of birth. Tkinter is a standard GUI toolkit for Python, providing easy-to-use interfaces for creating desktop applications. This project addresses the need for a simple and intuitive tool to quickly calculate ages, which is relevant in various contexts such as age verification, age-based eligibility, and personal interest.

**Identification of Client/Need/Relevant Contemporary Issue**: The client for this project could be any individual, organization, or developer community seeking a user-friendly tool for calculating ages. The need arises from the common requirement to calculate ages accurately and efficiently in various scenarios, including:

* Age Verification: Businesses and organizations often need to verify the age of individuals for services such as purchasing alcohol, accessing age- restricted content, or entering into legal agreements.
* Age-based Eligibility: Certain services or benefits may have age-based eligibility criteria. An age calculator can help determine whether an individual meets these criteria.
* Personal Interest: Individuals may want to calculate their own or others' ages for personal reasons, such as celebrating birthdays, tracking milestones, or planning events.
* Legal and Medical Fields: In legal and medical contexts, accurate age calculation is crucial for determining eligibility for certain procedures, treatments, or legal rights.

## PROBLEM STATEMENT

Design an age calculator application using Tkinter in Python. The application should provide a user interface allowing individuals to input their birth date. Upon clicking a button labeled "Calculate Age", the application should compute their current age in years, months, and days and display the result in a clear and understandable format.

The application must include the following features:

An input field where users can enter their birth date.

A button labeled "Calculate Age" that triggers the age calculation.

Validation to ensure that the input date is in the correct format (YYYY-MM-DD). Error handling to display a message if the input date format is invalid.

Age calculation logic considering leap years and the varying number of days in each month.

A clear and user-friendly interface for interaction.

The application should be designed to provide accurate and reliable age calculations while ensuring a smooth and intuitive user experience.

## METHODOLOGY

* **Requirement Analysis:** Understand the requirements of the age calculator application, including user inputs, expected outputs, and any additional features or constraints.
* **GUI Design:** Design the graphical user interface (GUI) using Tkinter. This involves creating the necessary widgets such as labels, entry fields, buttons, etc., and arranging them in a logical layout.
* **Input Validation:** Implement input validation to ensure that the user enters the birth date in the correct format (YYYY-MM-DD). Handle cases where the input format is invalid and provide appropriate error messages to the user.
* **Age Calculation Logic:** Develop the logic to calculate the age based on the input birth date and the current date. Consider factors such as leap years and the varying number of days in each month to ensure accurate calculations.
* **Event Handling:** Define event handlers for the GUI components, such as button clicks. When the user clicks the "Calculate Age" button, trigger the age calculation process.
* **Output Display:** Display the calculated age in years, months, and days in a clear and understandable format. You may choose to use message boxes, labels, or other GUI components to present the output to the user.
* **Testing:** Test the application thoroughly to ensure that it works as expected in various scenarios. Verify that the input validation, age calculation, and output display functionalities are all functioning correctly.
* **User Experience (UX) Improvement:** Iterate on the design and functionality of the application to enhance the user experience. Consider factors such as usability, aesthetics, and responsiveness to make the application more intuitive and user-friendly.
* **Documentation:** Document the code, including explanations of key functionalities, input/output formats, and any dependencies or requirements for running the application.
* **Deployment:** Once the application is complete and tested, deploy it for use by end-users. You may choose to distribute the application as a standalone executable or package it for

distribution through platforms like PyPI or as source code on GitHub.

By following this methodology, you can systematically develop an age calculator application using Tkinter in Python, ensuring that it meets the requirements and provides a satisfactory user experience.

## SOFTWARE USED

For developing the age calculator application using Tkinter in Python, you would typically use the following software:

**Python:** Python is the programming language in which you'll write your code. Ensure you have Python installed on your system. You can download Python from the official website: https://[www.python.org/downloads/](http://www.python.org/downloads/)

**Integrated Development Environment (IDE):** An IDE provides an environment for writing and debugging your code efficiently. Some popular IDEs for Python development include:

Visual Studio Code

Tkinter Library: Tkinter is Python's standard GUI (Graphical User Interface) toolkit. It comes pre-installed with Python, so you don't need to install it separately. You'll be using Tkinter to create the graphical interface for your age calculator application.

**Text Editor:** If you prefer not to use a full-fledged IDE, you can use a simple text editor to write your code. Common text editors for coding include:

VS Code

**Command Line or Terminal:** You'll need a command-line interface or terminal to run your Python scripts. This comes with your operating system (Command Prompt on Windows, Terminal on macOS and Linux).

Ensure you have these software components installed and set up on your system before you start developing your age calculator application using Tkinter in Python.

## LITERATURE REVIEW

1. "Tkinter GUI Programming by Example" by David Love, Packt Publishing, 2018: This book provides a comprehensive guide to Tkinter GUI programming with Python. It covers various aspects of Tkinter, including building simple to complex GUI applications. Although it doesn't specifically cover age calculators, it offers valuable insights into building GUI applications using Tkinter, which can be helpful in developing an age calculator.
2. "Python GUI Programming with Tkinter" by Alan D. Moore, Apress, 2017: Moore's book is a practical guide to GUI programming using Tkinter. It offers explanations and examples of Tkinter's functionalities and provides hands-on exercises to reinforce learning. While not specifically focusing on age calculators, the book equips readers with the necessary skills to develop such applications.
3. "Programming Python: Powerful Object-Oriented Programming" by Mark Lutz, O'Reilly Media, 2019: This book covers various aspects of Python programming, including GUI development with Tkinter. It provides detailed explanations and examples, making it suitable for both beginners and experienced programmers. While not solely focused on Tkinter GUIs, it offers valuable insights into Python programming techniques that can be applied in developing an age calculator.
4. "Tkinter GUI Application Development Blueprints" by Bhaskar Chaudhary, Packt Publishing, 2015: Chaudhary's book offers a series of projects that demonstrate how to build GUI applications using Tkinter. While not specifically addressing age calculators, it provides step-by-step instructions on developing various types of applications, which can serve as a reference for creating an age calculator.
5. "Python Programming: An Introduction to Computer Science" by John Zelle, Franklin, Beedle & Associates Inc., 2016: Zelle's book is widely used as a textbook for introductory Python programming courses. It covers fundamental programming concepts using Python and includes chapters on GUI programming with Tkinter. Although it doesn't delve deeply into complex GUI applications like age calculators, it provides a solid foundation for understanding Tkinter's basics.

These resources offer valuable insights into Python programming, GUI development with Tkinter, and software design principles, which are essential for developing an age calculator application. By studying these materials, developers can gain the knowledge and skills necessary to create functional and user-friendly age calculator programs using Python and Tkinter.

## CONCLUSION

In conclusion, developing an age calculator application using Tkinter in Python is an excellent learning opportunity that covers various aspects of software development.

Throughout the process, you'll gain valuable skills in GUI programming, event handling, input validation, date and time manipulation, error handling, user experience design, testing, and documentation.

By successfully completing this project, you'll have a deeper understanding of how to create interactive graphical applications in Python, as well as the ability to handle user input effectively and provide meaningful feedback to users. Additionally, you'll strengthen your problem-solving skills by addressing challenges such as input validation and error handling.

Furthermore, the age calculator application serves as a practical example of real-world software development, demonstrating how to translate requirements into a functional and user-friendly application. This experience can be applied to future projects and provide a solid foundation for further exploration of GUI programming and application development in Python.

Overall, developing an age calculator application using Tkinter in Python is not only a rewarding learning experience but also a valuable addition to your portfolio as a Python developer. It showcases your proficiency in building practical applications and solving real- world problems, making you a more versatile and competent software engineer

## REFERENCES

* **Python Documentation:** [**https://docs.python.org/3/**](https://docs.python.org/3/)**.**
* **Tkinter Documentation:** [**https://docs.python.org/3/library/tkinter.html**](https://docs.python.org/3/library/tkinter.html)**.**
* **Real Python:** [**https://realpython.com/tkinter-python/**](https://realpython.com/tkinter-python/)**.**
* **GeeksforGeeks: https://**[**www.geeksforgeeks.org/python-gui-tkinter/**](http://www.geeksforgeeks.org/python-gui-tkinter/)

### Tutorialspoint: https://[www.tutorialspoint.com/python/python\_gui\_programming.htm](http://www.tutorialspoint.com/python/python_gui_programming.htm)

* **"Python GUI Programming Cookbook" by Burkhard A. Meier**