DAYS CALCULATOR IN PYTHON

## A PROJECT REPORT

***Submitted by***

### PRABUKUMAR V (2303811720521039)

***In partial fulfillment for the completion of the course***

# CGB1121-PYTHON PROGRAMMING

***in***

# INFORMATION TECHNOLOGY

**K.RAMAKRISHNANCOLLEGEOFTECHNOLOGY**

(An Autonomous Institution ,Affiliated to Anna University Chennai and Approved by AICTE ,NewDelhi)

**SAMAYAPURAM– 621 112**

**MAY,2024**

i

# K.RAMAKRISHNANCOLLEGEOFTECHNOLOGY (AUTONOMOUS)

**SAMAYAPURAM– 621 112**

# BONAFIDECERTIFICATE

Certified that this project report titled **“DAYS CALCULATOR IN PYTHON IN PYTHON”** is the bonafide work of **PRABUKUMAR V (2303811720521039)** who carried out the project under my supervision. Certified further , that to the best

of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a course was conferred on an earlier occasion on this or any other candidate.



##### SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D., **HEAD OF THE DEPARTMENT** PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

**SIGNATURE** Ms.P.Sudha,ME.,Ph.D., **SUPERVISOR** ASSISTANT PROFESSOR

Department of ECE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.



Submitted for the viva –voce examination held on 18.06.2024 INTRENAL EXAMINER

# DECLARATION

I declare that the project report on **“ DAYS CALCULATOR IN PYTHON”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF TECHNOLOGY** . This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1121- PYTHON PROGRAMMING**.

### Signature



PRABUKUMARV

Place:Samayapuram Date:18.06.2024

### ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project guide Ms.P.SUDHA ME.,(Ph.D)Department of **ELECTRONICS AND COMMUNICATION ENGINEERING,**

for her incalculable suggestions,creativity,assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

##### VISIONOF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education.

##### MISSIONOF THE INSTITUTION

* Producesmarttechnocratswithempiricalknowledgewhocansurmounttheglobal challenges.
* Create a diverse,fully-engaged,learner-centric campus environment to provide quality education to the students.
* Maintainmutuallybeneficialpartnershipswithouralumni,industry,and Professional associations.

##### VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

##### MISSIONOF DEPARTMENT

**M1:Industry Specific:**Tonurturestudentsinworkingwithvarioushardwareandsoftwareplatforms inclined with the best practices of industry.

**M2:Research:**To prepare students for research-oriented activities.

**M3:Society:**Toempowerstudentswiththerequiredskillstosolvecomplextechnologicalproblems of society.

##### PROGRAMEDUCATIONALOBJECTIVES

1. **PEO1:Domain Knowledge**

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

##### PEO2:Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

##### PEO3:Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

##### PROGRAMOUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as amember or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

##### ABSTRACT

The provided Python script demonstrates a basic days calculator using the datetime module. The script includes a function, days\_between\_dates, which calculates the number of days between two specified dates. The dates are passed as strings and converted to datetime objects using the strptime method. The difference between these dates is computed using subtraction, yielding a timedelta object from which the number of days is extracted. An example usage is included, showcasing how to use the function to find the difference in days between two specific dates. This approach is straight forward and leverages Python's robust date handling capabilities to perform date arithmetic efficiently.

**TABLEOFCONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGENO.** |
|  | **ABSTRACT** | viii |
|  | **LIST OF FIGURES** | ix |
|  | **LISTOF ABBREVIATIONS** | x |
| **1** | **INTRODUCTION** |  |
|  | **1.1INTRODUCTION TO PYTHON** | 1 |
|  | 1.1.1. Overview | 1 |
|  | 1.1.2.Programming Paradigms | 1 |
|  | 1.1.3.Standard Library | 1 |
|  | 1.1.4.Third-Party Libraries and Frameworks | 1 |
|  | 1.1.5.Versionsof Python | 1 |
|  | 1.1.6. Python Tools | 2 |
|  | 1.1.7.Versatility and Adoption | 2 |
| **2** | **PROJECTDESCRIPTION** |  |
|  | 2.1. PROJECT INTRODUCTION | 3 |
|  | 2.1. PROJECT OBJECTIVE | 3 |
|  | 2.3. PROBLEM STATEMENT | 3 |
|  | 2.4. LIBRARIESUSED | 3 |
| **3** | **SYSTEMANALYSIS** |  |
|  | **3.1. EXISTINGSYSTEM** | 5 |
|  | 3.1.1. Disadvantages | 5 |
|  | **3.2PROPOSED SYSTEM** | 6 |
|  | 3.2.1. Advantage | 6 |
| **4** | **SYSTEMDESIGN &MODULES** |  |
|  | 4.1. **BLOCK DIAGRAM** | 7 |
|  | 4.2. **MODULEDESCRIPTION** | 7 |
|  | 4.2.1.InputText Module | 7 |
|  | 4.2.2.EmojiDetection Module | 7 |
|  | 4.2.3.EmojitoTextConversion Module | 7 |
|  | 4.2.4.OutputText Module | 8 |
| **5** | **CONCLUSION&FUTUREENHANCEMENT** |  |
|  | 5.1. CONCLUSION | 9 |
|  | 5.2. FUTUREENHANCEMENT | 9 |
| **6** | **APPENDICES** |  |
|  | AppendixA- Source code | 11 |
|  | AppendixB-Screenshots | 12 |

**LISTOF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURENO** | **TITLE** | **PAGENO.** |
| 4.1 | BLOCKDIAGRAM | 18 |

## LISTOFABBREVIATIONS

**ABBREVIATIONS**

IDE - Integrated Development Environment

VS Code - VisualStudio Code

re - Regular Expression

iOS - Iphone Operating System

NLP - Natural Language Processing

GUI - Graphical UserInterface

APIs - Application Programming Interface

**CHAPTER1**

## INTRODUCTION

##### INTRODUCTIONTO PYTHON

**Overview**

Python is a widely-used, high-level programming language renowned for its readability and simplicity, making it an ideal choice for both novice and seasoned programmers. Created by Guido van Rossum and released in 1991, Python's core philosophy emphasizes code readability and straightforward syntax, allowing developers to write clear and concise code more efficiently compared to other languages like C++ or Java.

##### Programming Paradigms

Python supports various programming paradigms,including procedural,object-oriented, and functional programming. This flexibility, combined with a dynamic type system and automatic memory management, facilitates the development of a wide range of applications, from simple scripts to complex software systems.

##### Standard Library

The language's comprehensive standard library, often referred to as "batteries-included," provides built-in modules and functions for handling many programming tasks, such as file I/O, system calls, and even web services. This extensive library helps streamline the development process by offering ready-to-use solutions for common programming challenges.

##### Third-Party Libraries And Frameworks

One of Python's significant strengths is its extensive ecosystem of third-party libraries and frameworks. Popular libraries such as NumPy and Pandas enable efficient data manipulation and analysis, while frameworks like Django and Flask streamline web development. In the realm of machine learning and artificial intelligence, libraries like TensorFlow and PyTorch are widely adopted for building and deploying sophisticated models.

##### Versions Of Python

Pythonhasundergonesignificantevolutionsinceitsinception,with two major versions in use

today:

Python2:Releasedin2000,Python2.x series was a major milestone and widely used for

Many years. However,it reached its end of life on January1,2020,and is no longer maintained.

Python 3: Introduced in 2008, Python 3.x series brought substantial improvements and changes to the language, such as better Unicode support, a more consistent syntax, and enhanced standard libraries. Python 3 is the recommended version for all new projects.

##### Python Tools

Python's ecosystem includes numerous tools that enhance productivity and development experience:

* **IDEs and Code Editors:**Popular options include PyCharm,VSCode,and Jupyter Notebook, which offer features like syntax highlighting, code completion, and debugging.
* **Package Management:** Tools like pip and cond a facilitate the installation and management of Python libraries and dependencies.
* **Virtual Environments**: virtualenv and venv allow developers to create isolated environments for different projects, ensuring dependency conflicts are avoided.
* **Testing Frameworks:** unittest, pytest, and nose are commonly used for writing and running tests to ensure code reliability and correctness.
* **Build Tools:** setup tools and wheel help in packaging Python projects, making them easy to distribute and install.
* **Documentation Generators:**Tools like Sphinx are used to create comprehensive documentation for Python projects.
* **Linters and Formatters:** pylint,flake8,and black help maintain code quality and consistency by enforcing coding standards and formatting.

##### Versatility And Adoption

Python's simplicity and versatility have led to its widespread adoption in various fields, including web development, data science, artificial intelligence, automation, and scientific computing. Its active community continually contributes to a rich repository of resources, tutorials, and documentation, making it easier for developers to learn and apply Python effectively.

**CHAPTER 2 PROJECT DESCRIPTION**

**PROJECT INTRODUCTION**

The datetime module in Python is a versatile and powerful tool for handling dates and times. One common task that can be easily performed using this module is calculating the number of days between two dates. This can be useful in a variety of applications, such as determining deadlines, measuring the duration of events, or simply tracking the passage of time between two points.The script presented here provides a straightforward implementation of a days calculator using the datetime module. The key function, days\_between\_dates, accepts two date strings and a date format, converts these strings into datetime objects, and calculates the difference in days between the two dates. This difference is then returned as an integer, representing the number of days.

### PROJECT OBJECTIVE

The objective of this project is to develop a Python-based days calculator that accurately computes the number of days between two given dates. This calculator aims to:

Enhance Date Manipulation Skills:

Demonstrate practical usage of the datetime module for handling and manipulating dates in Python.

Provide a Useful Tool:

Offer a simple yet effective utility that can be utilized in various real-world applications, such as project management, event planning, and historical data analysis.

Ensure Flexibility: Allow for customiz able date formats to accommodate different date representations, enhancing the utility's adaptability to various user requirements.

Promote Code Re-usability: Implement a modular function (days\_between\_dates) that can be easily integrated into larger projects or reused in different contexts where date difference calculations are needed.

### PROBLEM STATEMENT

In many applications,there is a need to calculate the number of days between two dates. This can be required for tasks such as scheduling, tracking progress, calculating durations, or comparing historical data. While date arithmetic can be complex due to factors like leap years and varying month lengths, a robust solution is essential for accurate and reliable computations.

SpecificChallenges:

Date Input Handling: Users may provide dates in different formats, necessitating a flexible solution that can parse various date string formats.Date Validation: Ensuring that the provided dates are valid and correctly formatted to prevent calculation errors.

LeapYearsandMonthLengths:

Accounting for leap years and the varying number of days in different months to ensure accurate day difference calculations.

### LIBRARIES USED

datetime.datetime

This class is used for working with dates and times. It includes methods for creating, manipulating, and formatting date and time objects.strptime(date\_string, format):Parses a string representing a date/time according to a format.strftime(format): Formats a datetime object as a string according to a format.

1. datetime.timedelta

This class is used for representing the difference between two dates or times.Attributes: days, seconds, microseconds

### CHAPTER 3 SYSTEM ANALYSIS

**EXISTING SYSTEM**

In the context of calculating the number of days between two dates, several systems and libraries already exist that provide this functionality. Here, we will out line some of the most commonly used methods and tools available.

### DISADVANTAGES

##### A] Limited Functionality Compared to Advanced Libraries:

The custom solution focuses on basic date difference calculations and lacks the advanced features offered by third-party libraries like pand as or date utill,such as timezone handling, business day calculations, and more complex date manipulations.

##### Manual Date Format Handling:

The function relies on the user to specify the correct date format. If the date format is incorrect or inconsistent,it can lead to errors or incorrect results.Third-party libraries often have more robust parsing capabilities that handle various date formats more gracefully.

##### NoBuilt-in Validation:

Thefunctionassumesthattheinputdatesarevalidandcorrectlyformatted.Additional code is required to handle invalid dates, out-of-range values, or other input errors, whereas advanced libraries provide built-in validation mechanisms.

##### Scalability:

For large-scale applications or datasets involving complex date operations, the custom solutionmaynotbeasefficientorscalableasoptimizedthird-partylibrariesdesignedfor performance.

### PROPOSED SYSTEM

To address the limitations of the existing solutions and provide a more flexible and user-friendly days calculator, the proposed system aims to create a Python-based application using the datetime module. This system will be designed to hand levarious date formats, include basic validation, and offer clear feedback to users.

### ADVANTAGES

These advantage smake it a practical,flexible,and educational tool for users needing to calculate the number of days between two dates.

1. Flexibility:
2. Customizable Date Formats
3. The system allows users to input dates invarious formats,accommodating different regional and personal preferences.

### Adaptability:

Itcaneasilybeextendedormodifiedtohandleadditionaldateformatsormore complex date manipulations if needed.

### User-Friendly Interface: Simple Input and Output:

The straight forward interface ensures user scan easily input dates and receive results without needing extensive technical knowledge.

### Clear Feedback:

Informativeerrormessageshelpuserscorrectinputerrorsquickly,enhancing the overall user experience.

### Basic Validation Error Handling:

The system includes basic validation to ensure input dates are correctly formatted and valid, reducing the likelihood of calculation errors.

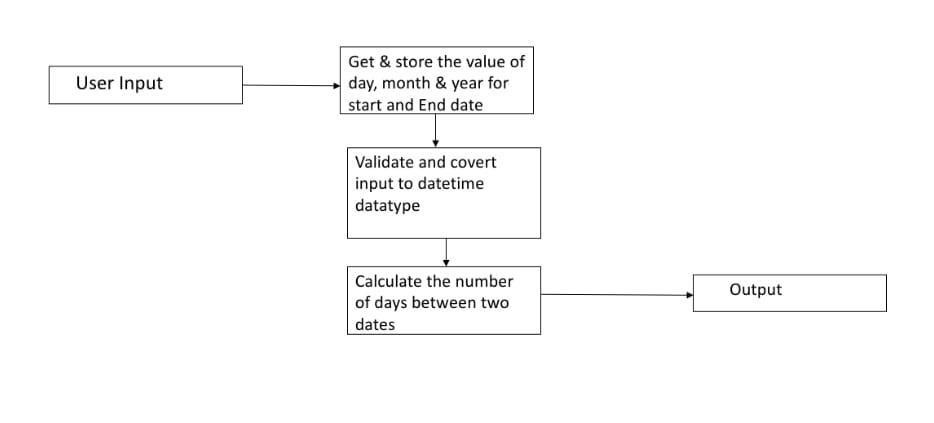
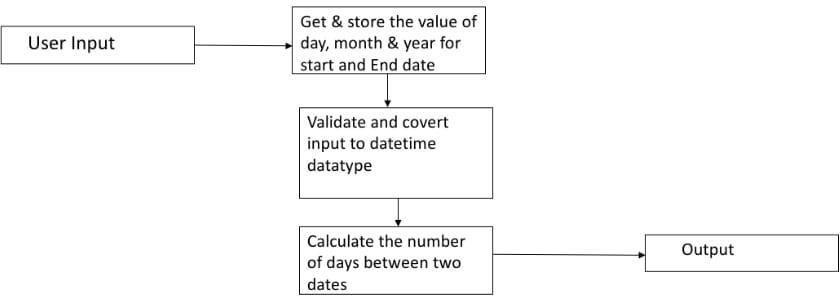
### Informative Errors:

Users are provided with clear error messages when invalid dates or formats are entered, improving usability.

### CHAPTER 4

**SYSTEM DESIGN & MODULES**

### BLOCK DIAGRAM



**Fig.4.1.BlockDiagram**

### MODULE DESCRIPTION INPUT TEXT MODULE

**InputTextModule(input\_text.py)**

### Description:

The Input Text Module handles user inputs for dates, ensuring they are in the correct form at and can be processed further by other modules.It includes functions for prompting user input, validating, and parsing dates.

### Functions: get\_date\_input(prompt):

Prompts the user for a date input and returns the entered date.validate\_date(date\_text):

Validates if the provided date text matches the expected format.parse\_date(date\_text):

Parses a date text in to a datetime object

### DateDifference Calculation Module (date\_difference.py) Description:

The Date Difference Calculation Module computes the difference in days between two dates using the datetime module.

It includes a function to calculate and return the number of days between two datetime objects. This module ensures accurate date arithmetic and is essential for applications that need to measure intervals between dates. By providing a straightforward interface, it simplifies the process of date difference calculations.

### Functions:

calculate\_days\_between(date1, date2): Calculates the number of days between two date time objects. **Business Days Calculation Module (business\_days.py) Description:**

The Business Days Calculation Module computes the number of business days between two dates, excluding weekends and optionally specified holidays. It includes afunctionthatgenerateseachdayinthedaterangeandcountsonlytheweekdaysthat are not holidays.

This module is crucial for applications that need to account for working days, providing an accurate measure of business days between dates. By handling both weekends and holidays , it ensures precise and useful calculations for business-related purposes.

Functions:calculate\_business\_days(date1,date2,holidays=[]):Calculatesthe number of business days between two dates, excluding weekends and specified holidays.

### Output Results Module (output\_results.py) Description:

The Output Results Module handles the display of calculated results to the user, ensuring clear and formatted output. It includes methods for displaying messages and formatting the number of days into readable text.

This module is essential for presenting information in a user-friendly manner, enhancing the over all user experience.By providing a structured way to output results, it ensures that users can easily understand and interpret the calculated data.

### OUTPUT TEXT MODULE

The Output Text Module handles the display and formatting of calculated results to the user, ensuring that the output is clear and easy to understand. This module provides methods for displaying messages and formatting the number of days into readable text. It plays a crucial role in enhancing the overall user experience by presenting information in a user-friendly manner.

* display\_message (message): Displays a message to the user.
* format\_days (days,description="days"):

Formats the number of days into area able message.

### CHAPTER 5

**CONCLUSION & FUTURE ENHANCEMENT**

**CONCLUSION**

The Days Calculator in Python is a versatile and efficient tool designed to accurately compute the number of days between two dates, as well as the number of business days, excluding weekends and holidays.

By breaking down the system into modular components, it ensures code clarity, maintainability, and ease of use. Each module—Input Text, Date Difference Calculation ,Business Days Calculation ,and Output Results—plays a crucial role in the overall functionality of the application.

## Input Text Module handles date inputs and ensures they are in the correct format.

Date Difference Calculation Module accurately computes the total number of days between two dates. Business Days Calculation Module calculates the number of business days, excluding weekends and specified holidays.Output Results Module for mats and displays the results clearly to the user.

This modular approach not only simplifies development and maintenance but also enhances the user experience by providing clear and accurate information.

The Days Calculator is a valuable tool for both personal and bus in issuse , offering reliable date calculations for various applications.

# FUTURE ENHANCEMENTS

To further enhance the functionality and user experience of the Days Calculator, several improvements can be considered:

Graphical User Interface(GUI):

Developing a user-friendly GUI to make the application more accessible to users who prefer a visual interface over a command-line interface.

Timezone Handling:

Addings upport for timezones to allow for more accurate calculations in applications involving international dates.

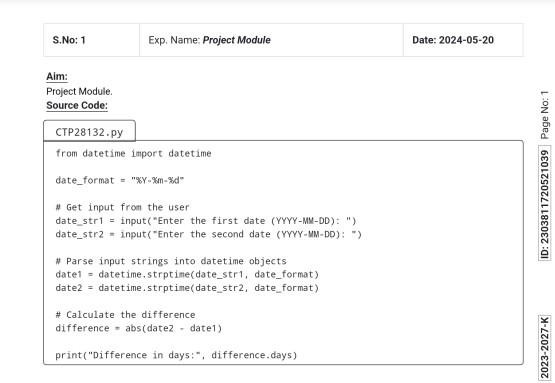
Localization:

Supporting multiple languages and regional date formats to a global audience.

Customizable Work weeks:

Allowing users to define custom workweeks and holidays for more accurate business days calculations in different in dustries. Integration with Calendar Applications: Providing options to import dates from and export results to popular calendar applications like Google Calendar or Microsoft Outlook.

### APPENDICES APPENDIX A-SOURCE CODE



#### APPENDIX B-SCREENSHOTS

