Report On

GUI Age Calculator

Submitted in partial fulfillment of the requirements of the Course project in

Semester III of Second Year Computer Engineering

by

Ayush Dahiwale (Roll No.35)

Ved Chaudhari (Roll No.33)

Pranali Chakor (Roll No.22)

Supervisor

Prof. Sneha Mhatre

**Vidyavardhini's College of Engineering & Technology**

**Department of Computer Engineering**



**(2023-24)**

**Vidyavardhini's College of Engineering & Technology**

**Department of Computer Engineering**

**CERTIFICATE**

This is to certify that the project entitled “GUI Age Calculator” is a bonafide work of " Ayush Dahiwale (Roll No.35), Ved Chaudhari (Roll No.33), Pranali Chakor (Roll No.22)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester IV of Second Year Computer Engineering.

**Supervisor**

|  |  |  |
| --- | --- | --- |
| Prof. Sneha Mhatre |  |  |
| Dr Megha Trivedi  Head of Department |  | Dr. H.V. Vankudre  Principal |

**Abstract**

The "Age Calculator" application is a user-friendly and visually appealing tool designed to calculate a person's age based on their birth date. Developed using the Tkinter library in Python, the application provides a seamless and intuitive interface for users to input their name and birth date (year, month, day) and obtain their age in both years and days. Furthermore, the application incorporates a MySQL database to store the user's information, ensuring efficient record-keeping and easy access to age-related data.

The graphical user interface (GUI) of the "Age Calculator" application is thoughtfully designed to enhance user experience. It features a background image that adds aesthetic appeal and sets a pleasant tone for the user interaction. The input fields for name and birth date are prominently displayed, making it convenient for users to enter their information. Additionally, the "Calculate Age" button is strategically placed, making it easily accessible for users to initiate the age calculation process.

Upon clicking the "Calculate Age" button, the application performs a series of calculations based on the provided birth date to determine the user's age. The calculated age is then displayed on the GUI along with a personalized greeting, adding a delightful touch to the user experience. Furthermore, the application ensures data integrity and security by inserting the user's information, including their name, birth date, and calculated age in years, into the MySQL database. This database integration enables users to retrieve their age-related information at a later time, providing a valuable resource for personal record-keeping and reference.

The "Age Calculator" application serves as an exemplary showcase of the capabilities of Tkinter for creating interactive GUIs in Python and MySQL for efficient database management. Its user-friendly interface and robust functionality make it a valuable tool for users seeking a convenient and reliable method to calculate and store their age-related information. As technology continues to evolve, the "Age Calculator" application can be further enhanced with additional features and functionalities to meet the changing needs and preferences of users.

**Contents**

**Pg. No**

|  |  |
| --- | --- |
| 1 Problem Statement | 1 |
| 2 Block Diagram | 3 |
| 3 Module Description | 5 |
| 4 Software and Hardware Requirements | 6 |
| 5 Code | 8 |
| 6 Result | 14 |
| 7 Conclusion | 15 |
| 8 Reference | 16 |

**Problem Statement**

In today's digital age, there is a growing need for efficient and user-friendly tools to manage personal information, including age-related data. Existing solutions for calculating and storing age often lack intuitive interfaces and integration with databases, making it challenging for individuals to track their age accurately and conveniently. Additionally, these solutions often do not offer personalized greetings or visual enhancements, diminishing the overall user experience.

To address these shortcomings, we aim to develop an "Age Calculator" application that provides a comprehensive solution for calculating, storing, and managing age-related information. The application will be developed using the Tkinter library in Python, offering a visually appealing and intuitive interface for users to input their name and birth date and obtain their age in years and days. Furthermore, the application will integrate with a MySQL database to ensure efficient storage and retrieval of user information, enabling users to access their age-related data easily.

The key objectives of the "Age Calculator" application are as follows:

1. Develop a user-friendly interface: The application will feature an intuitive design with input fields for name and birth date, a button to calculate age, and a display area for the calculated age and personalized greeting. Visual enhancements, such as background images, will be incorporated to improve the overall user experience.
2. Implement age calculation functionality: Upon user input, the application will calculate the user's age based on their birth date and display the result in both years and days. The application will also provide a personalized greeting to enhance user engagement.
3. Integrate with a MySQL database: The application will store user information, including name, birth date, and calculated age, in a MySQL database for efficient storage and retrieval. This integration will enable users to access their age-related data at any time, ensuring data integrity and security.
4. Ensure scalability and flexibility: The application will be designed to accommodate future enhancements and modifications, allowing for scalability and flexibility to meet the evolving needs of users.

By developing the "Age Calculator" application, we aim to provide a comprehensive and user-friendly solution for calculating and managing age-related information.

**Block Diagram**

****

Store Input Using Mysql\_connector

Take Input From the Website



The "Age Calculator" application consists of several key components that work together to provide a user-friendly interface for calculating and storing age-related information. The block diagram below illustrates the main components and their interactions.

1. User Interface (UI):

* Input Fields: Allows users to enter their name, birth year, month, and day.
* Calculate Age Button: Triggers the age calculation process.
* Result Display: Displays the user's calculated age in years and days along with a personalized greeting.
* Visual Enhancements: Includes background images and a visually appealing layout to enhance user experience.

2. Tkinter Library:

* Provides tools for creating the graphical user interface (GUI) of the application.
* Enables the creation of input fields, buttons, labels, and other UI elements.
* Facilitates the integration of visual enhancements and interactive elements into the application.

3. Age Calculation Module:

* Receives input from the UI and calculates the user's age based on their birth date.
* Utilizes date calculations to determine the user's age in years and days.
* Sends the calculated age back to the UI for display.

4. MySQL Database:

* Stores user information, including name, birth date, and calculated age.
* Enables efficient storage and retrieval of age-related data.
* Ensures data integrity and security by managing user information in a structured format.

5. Database Integration Module:

* Handles the interaction between the application and the MySQL database.
* Inserts user information into the database upon calculation of age.
* Retrieves age-related data from the database for display on the UI.

6. Application Logic:

* Coordinates the flow of data between the UI, age calculation module, and database integration module.
* Ensures that user input is processed correctly and that the calculated age is displayed accurately.
* Manages the overall functionality of the application, including error handling and data validation.

7. Personalized Greeting Module:

* Generates a personalized greeting for the user based on their name and age.
* Enhances user engagement and provides a customized user experience.

8. Visual Enhancements Module:

* Manages the display of visual enhancements, such as background images and layout design.
* Improves the aesthetics of the application and enhances user interaction.

In summary, the "Age Calculator" application utilizes the Tkinter library for creating a user-friendly interface, integrates with a MySQL database for efficient storage and retrieval of age-related information, and includes modules for age calculation, personalized greetings, and visual enhancements. This comprehensive approach ensures that the application provides a seamless and engaging experience for users seeking to calculate and manage their age-related data.

**Module Description**

1. User Interface (UI) Module:

* Description: This module provides the graphical user interface (GUI) for the application, allowing users to interact with the "Age Calculator" features.
* Features:
  + Input Fields: Name, birth year, month, and day.
  + Calculate Age Button: Initiates the age calculation process.
  + Result Display: Shows the calculated age in years and days along with a personalized greeting.
  + Visual Enhancements: Background images and layout design for an attractive UI.

2. Age Calculation Module:

* Description: This module calculates the user's age based on the provided birth date.
* Features:
  + Uses date calculations to determine the age in years and days.
  + Returns the calculated age to the UI module for display.

3. MySQL Database Integration Module:

* Description: Manages the interaction between the application and the MySQL database for storing and retrieving user information.
* Features:
  + Inserts user information (name, birth date, age) into the database.
  + Retrieves age-related data from the database for display on the UI.

4. Personalized Greeting Module:

* Description: Generates a personalized greeting for the user based on their name and age.
* Features:
  + Enhances user engagement by providing a customized message.
  + Adds a personal touch to the user experience.

5. Visual Enhancements Module:

* Description: Manages the visual aspects of the application, such as background images and layout design.
* Features:
  + Integrates visual elements to enhance the overall aesthetic appeal of the UI.
  + Improves user interaction by creating an engaging visual experience.

6. Application Logic Module:

* Description: Coordinates the flow of data and logic between the various modules of the application.
* Features:
  + Ensures that user input is processed correctly and that the calculated age is displayed accurately.
  + Manages error handling and data validation to ensure the application runs smoothly.

7. Tkinter Library Module:

* Description: Provides tools and functionality for creating the GUI of the application using the Tkinter library in Python.
* Features:
  + Creates UI elements such as input fields, buttons, and labels.
  + Enables the integration of visual enhancements and interactive elements into the application's interface.

8. Database Module:

* Description: Represents the MySQL database used for storing user information.
* Features:
  + Stores user data, including name, birth date, and calculated age.
  + Ensures data integrity and security by managing user information in a structured format.

9. Error Handling Module:

* Description: Manages errors and exceptions that may occur during the execution of the application.
* Features:
  + Detects and handles errors related to user input, database operations, and application logic.
  + Provides feedback to the user to help them understand and resolve any issues.

**Software and Hardware Requirements**Top of Form

**Software Requirements:**

1. Operating System: The "Age Calculator" application is compatible with various operating systems, including Windows (7 or later), macOS (10.12 or later), and Linux distributions
2. Python: The application requires Python version 3.6 or later to run. Python is a versatile programming language that provides robust support for GUI development through libraries like Tkinter.
3. Tkinter: Tkinter is Python's standard GUI toolkit, providing a set of modules and classes for creating graphical user interfaces. The "Age Calculator" application utilizes Tkinter to design the user interface, including input fields, buttons, and labels.
4. MySQL Server: The application requires a MySQL server to be installed The application uses MySQL to store user data, including names, birth dates, and calculated ages, ensuring efficient data storage and retrieval.
5. MySQL Connector/Python: To enable communication between the Python application and the MySQL database, the application utilizes the MySQL Connector/Python module. The module can be installed using pip, a Python package installer, by running the command **pip install mysql-connector-python**.

**Hardware Requirements:**

1. Processor: The application can run on a processor with a clock speed of 1 GHz or faster. The processor should be capable of handling the computational requirements of the application, including age calculations and database operations.
2. RAM: A minimum of 1 GB of RAM is recommended for running the application smoothly. However, to ensure optimal performance, especially when dealing with large datasets, it is advisable to have at least 2 GB of RAM.
3. Storage: The application requires approximately 100 MB of available hard disk space for installation. This includes space for the application files, the MySQL database, and any additional resources required for operation.

**Code**

**Python**

import tkinter as tk

from tkinter import \*

import mysql.connector

from datetime import date

# Connect to the MySQL database

conn = mysql.connector.connect(

host='127.0.0.1',

user='root',

password='Ayush@123',

database='agedata'

)

cursor = conn.cursor()

root = tk.Tk()

root.title("Age Calculator")

root.geometry("600x400")

# Create a canvas widget

canvas = Canvas(root, width=600, height=400, bg="#f0f0f0", highlightthickness=0)

canvas.pack()

def calculateAge():

today = date.today()

birth\_date = date(int(year\_value.get()), int(month\_value.get()), int(day\_value.get()))

age\_years = today.year - birth\_date.year - ((today.month, today.day) < (birth\_date.month, birth\_date.day))

age\_days = (today - birth\_date).days

years\_str = "year" if age\_years == 1 else "years"

days\_str = "day" if age\_days == 1 else "days"

result\_label.config(text=f"Hi {name\_value.get()}! Your age is: {age\_years} {years\_str} and {age\_days} {days\_str}")

# Insert the user's information into the database

cursor.execute("INSERT INTO Age (name, birth\_date, age) VALUES (%s, %s, %s)",

(name\_value.get(), f"{year\_value.get()}-{month\_value.get()}-{day\_value.get()}", age\_years))

conn.commit()

cursor.close()

# Load the background image

bg\_image = PhotoImage(file="background.png")

# Display the background image on the canvas

canvas.create\_image(0, 0, anchor=NW, image=bg\_image)

tk.Label(root, text="Name", font=("arial", 15, "bold")).place(x=50, y=50)

name\_value = tk.StringVar()

tk.Entry(root, textvariable=name\_value, font=("arial", 15)).place(x=200, y=50)

tk.Label(root, text="Year", font=("arial", 15, "bold")).place(x=50, y=100)

year\_value = tk.StringVar()

tk.Entry(root, textvariable=year\_value, font=("arial", 15)).place(x=200, y=100)

tk.Label(root, text="Month", font=("arial", 15, "bold")).place(x=50, y=150)

month\_value = tk.StringVar()

tk.Entry(root, textvariable=month\_value, font=("arial", 15)).place(x=200, y=150)

tk.Label(root, text="Day", font=("arial", 15, "bold")).place(x=50, y=200)

day\_value = tk.StringVar()

tk.Entry(root, textvariable=day\_value, font=("arial", 15)).place(x=200, y=200)

result\_label = tk.Label(root, text="", font=("arial", 15, "bold"))

result\_label.place(x=50, y=250)

tk.Button(root, text="Calculate Age", font=("arial", 15, "bold"), fg="white", bg="#21130d", command=calculateAge).place(x=200, y=300)

root.mainloop()

conn.close()

**SQL**

CREATE TABLE AGE (

id INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

BIRTH\_DATE DATE NOT NULL,

AGE Int

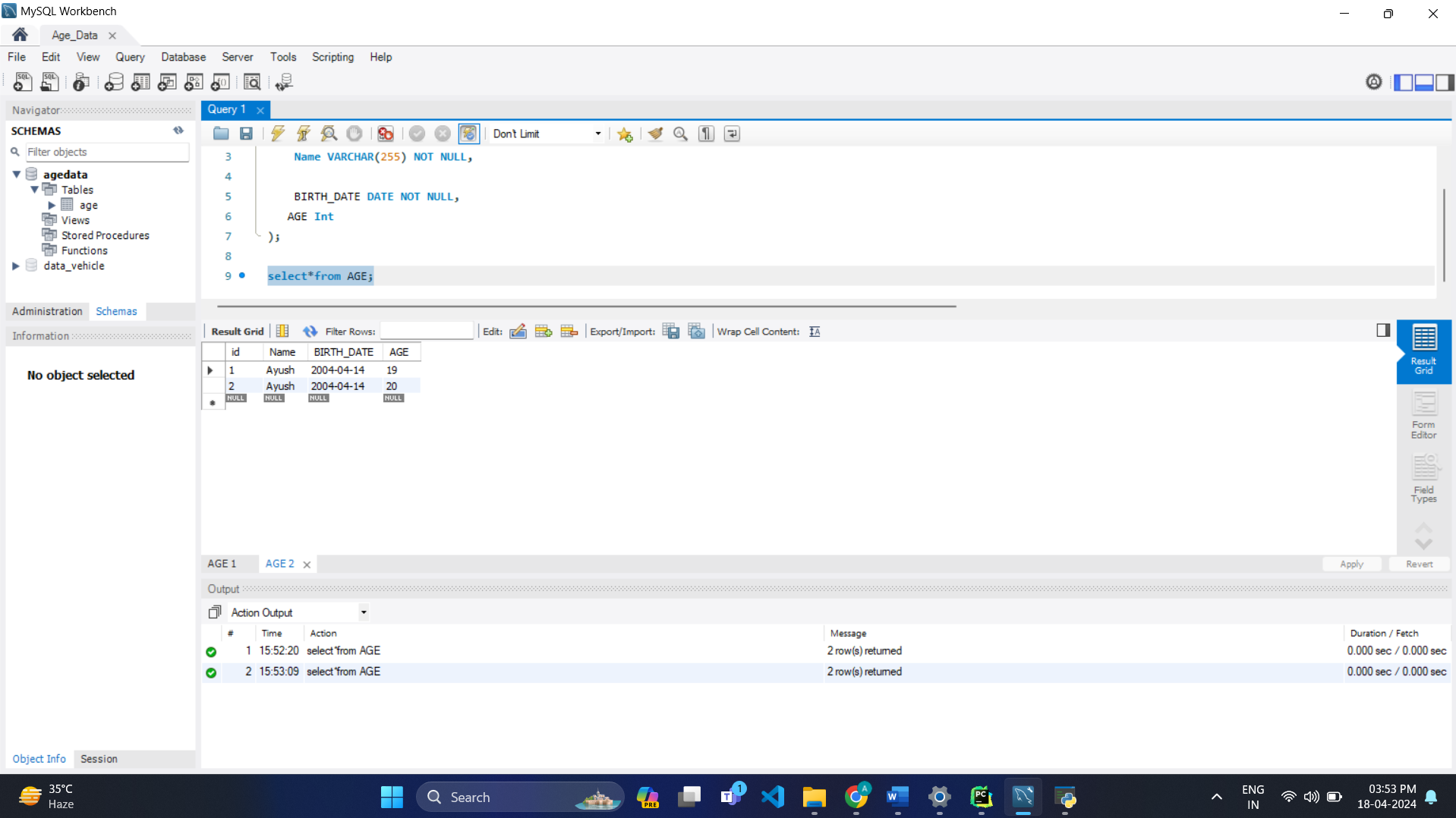
);

select\*from AGE;

**Result**



When users enter age details the system validates the input and stores it in the MySQL database.



**Conclusion**

In conclusion, the "Age Calculator" application is a user-friendly and efficient tool developed using Python and Tkinter for calculating and managing age-related information. The application provides a simple and intuitive interface for users to input their name and birth date, calculate their age in years and days, and receive a personalized greeting. Additionally, the application integrates with a MySQL database for storing and retrieving user information, ensuring data integrity and security.

The development of the "Age Calculator" application demonstrates the capabilities of Python and Tkinter for creating interactive GUIs and integrating with databases. The application's modular design allows for easy maintenance and scalability, making it adaptable to future enhancements and modifications.

Overall, the "Age Calculator" application serves as a valuable tool for individuals seeking a convenient and reliable method to track their age and manage age-related data. Its user-friendly interface, robust functionality, and database integration make it a practical solution for users of all ages.

Top of Form

**References**

1. Python Software Foundation. (n.d.). Python. Retrieved from <https://www.python.org/>
2. Python Software Foundation. (n.d.). Tkinter. Retrieved from <https://docs.python.org/3/library/tkinter.html>
3. Oracle Corporation. (n.d.). MySQL. Retrieved from <https://www.mysql.com/>
4. MySQL Documentation. (n.d.). MySQL Connector/Python Developer Guide. Retrieved from <https://dev.mysql.com/doc/connector-python/en/>
5. W3Schools. (n.d.). Python MySQL Tutorial. Retrieved from https://www.w3schools.com/python/python\_mysql\_getstarted.asp

**Top of Form**