CodeSoft

Topic: Python Programming

Start Date:01 October 2023

End Date:31 October 2023

Duration:30 Days

Submitted By: Deepa S R

Department of Computer Science and Engineering

Sri Jayachamarajendra College of Engineering

(JSSSTU),MYSURU

What is Python?

- Python is a high-level, versatile, and interpreted programming language that is known for its simplicity and readability.
- It was created by Guido van Rossum and was first released in 1991.
- Python is designed to be easy to understand and write, with a clear and concise syntax that emphasizes code readability.
- Python is also called Interpreted Language.
- Python is commonly used in various fields, such as web development (with frameworks like Django and Flask), data science (with libraries like NumPy, Pandas, and Matplotlib), artificial intelligence (with libraries like TensorFlow and PyTorch), and more.

TASK 1:TO DO LIST

A To-Do List application is a useful project that helps users manage command-line or GUI-based application using Python, allowing users to create, update, and track their to-do lists

- **Introduction**: The To-Do List Application is a Python project aimed at helping users manage and organize their tasks efficiently. It offers a simple GUI interface for users to create, update, and track their to-do lists.
- **Requirements**: Create a graphical user interface (GUI) using Tkinter, Allow users to add tasks to the to-do list, Enable users to remove tasks from the list,Provide user feedback through message boxes, Ensure that the application is user-friendly and intuitive
- **Design** :The application is designed using the Tkinter library for Python, It includes an Entry widget for adding tasks and a Listbox widget for displaying tasks,"Add Task" and "Remove Task" buttons are provided to facilitate user actions.

- **Implementation:-** The project is implemented in Python using the Tkinter library for the GUI.- Functions `add_task` and `remove_task` are defined to handle task addition and removal.- Message boxes are utilized for user feedback in case of invalid inputs.- The application is structured in a straightforward manner, making it easy to understand and modify.
- **Usage**: 1. Run the application.2. Use the "Add Task" button to add tasks to the to-do list.3. Select a task from the list, then use the "Remove Task" button to delete it.
- **Conclusion:** The To-Do List Application provides a simple and user-friendly solution for task management. It offers basic functionality for adding and removing tasks and serves as a starting point for further enhancements and improvements.
- **Code:**

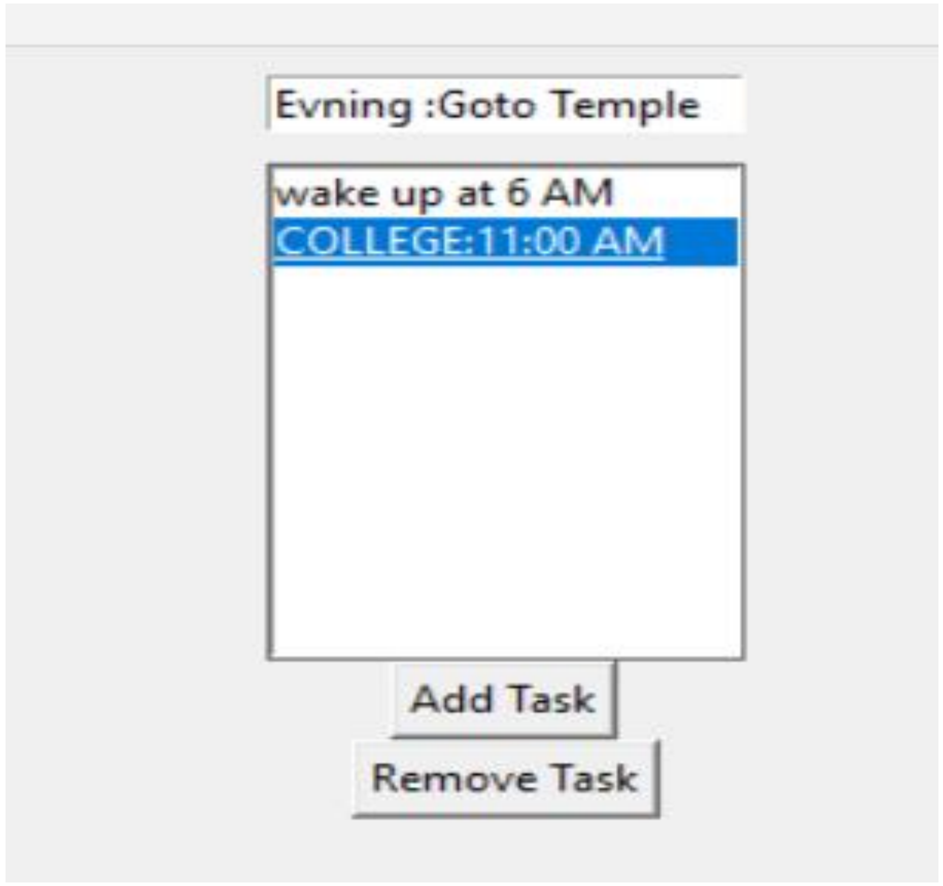
```
import tkinter as tk
from tkinter import messagebox
```

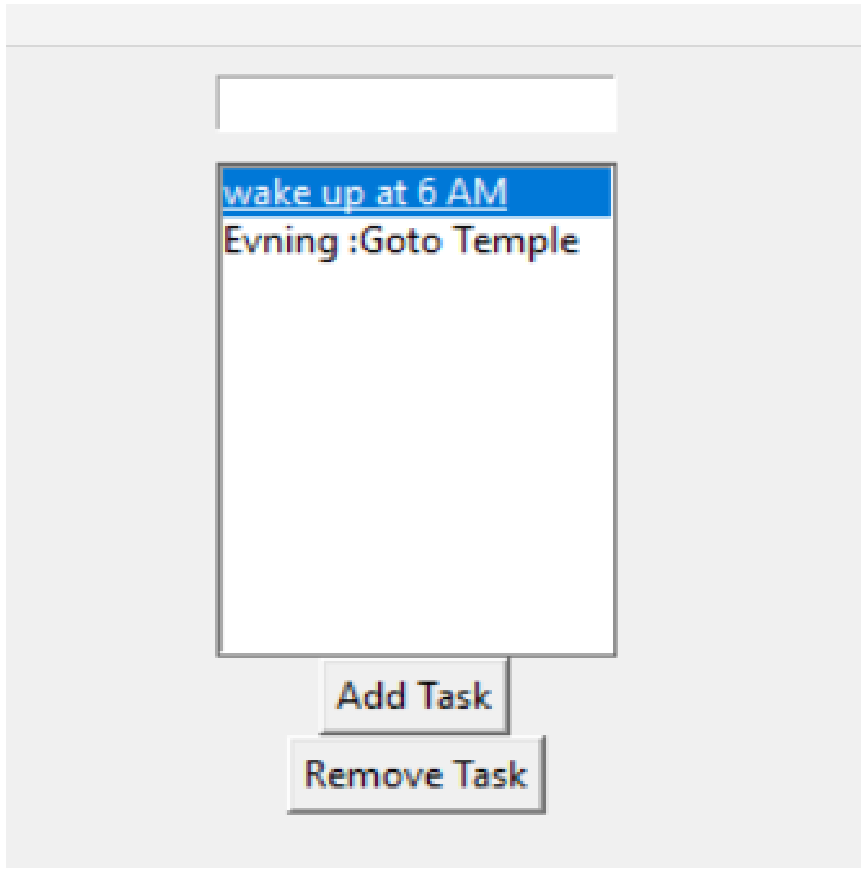
```python
# Function to add a task to the to-do list
def add_task():
    task = entry.get()
    if task:
        listbox.insert(tk.END, task)
       entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Warning", "Please enter a task.")

# Function to remove a selected task from the to-do list
def remove_task():
    try:
        selected_task_index = listbox.curselection()[0]
        listbox.delete(selected_task_index)
    except IndexError:
        messagebox.showwarning("Warning", "Please select a task to remove.")

# Create the main window
root = tk.Tk()
root.title("To-Do List")
```

```python
# Create an Entry widget for adding tasks
entry = tk.Entry(root)
entry.pack(pady=10)

# Create a Listbox widget to display tasks
listbox = tk.Listbox(root)
listbox.pack()

# Create buttons for adding and removing tasks
add_button = tk.Button(root, text="Add Task", command=add_task)
remove_button = tk.Button(root, text="Remove Task",
command=remove_task)
add_button.pack()
remove_button.pack()

# Run the application
root.mainloop()
```

OUTPUT

- ScreenShots





TASK 2:CALCULATOR

Design a simple calculator with basic arithmetic operations. Prompt the user to input two numbers and an operation choice. Perform the calculation and display the result.

- **Introduction:**The Simple Calculator Application is a Python project created with the Tkinter library, designed to perform basic arithmetic operations. This project provides users with a user-friendly graphical interface for performing calculations.
- **Requirement**s: Create a graphical user interface (GUI) for a basic calculator,Implement arithmetic operations such as addition, subtraction, multiplication, and division, Allow users to input numeric values and clear the display,Handle errors and provide user feedback for invalid inputs.
- **Design**-:The application features a graphical user interface (GUI) created with Tkinter, It includes a display area at the top for showing input and results,,Buttons for digits, arithmetic operators, and the equal sign (=) are arranged in a grid below the display.

- **Implementation-** The project is implemented in Python using the Tkinter library for the GUI,Functions for button clicks, clearing the display, and performing calculations are defined, The application is structured to handle user input and display results accordingly.
- **Usag:**1. Run the application.2. Click on the digit buttons (0-9) and arithmetic operation buttons (+, -, *, /) to enter your calculation.3. Click the '=' button to perform the calculation and display the result.4. Use the 'C' button to clear the display.5. Handle any errors or invalid inputs by ensuring the input is a valid mathematical expression.
- **Conclusion :**The Simple Calculator Application provides a straightforward solution for performing basic arithmetic calculations through a user-friendly graphical interface. While it meets the basic requirements, there is potential for further enhancement and expansion of features to cater to more complex mathematical needs.
- **Code**

```
import tkinter as tk
```

```python
# Function to update the display when a button is clicked
def button_click(number):
    current = entry.get()
    entry.delete(0, tk.END)
  entry.insert(0, current + str(number))

# Function to clear the display
def clear():
    entry.delete(0, tk.END)

# Function to perform the calculation
def calculate():
    current = entry.get()
    try:
        result = eval(current)
        entry.delete(0, tk.END)
        entry.insert(0, result)
    except Exception:
        entry.delete(0, tk.END)
        entry.insert(0, "Error")
```

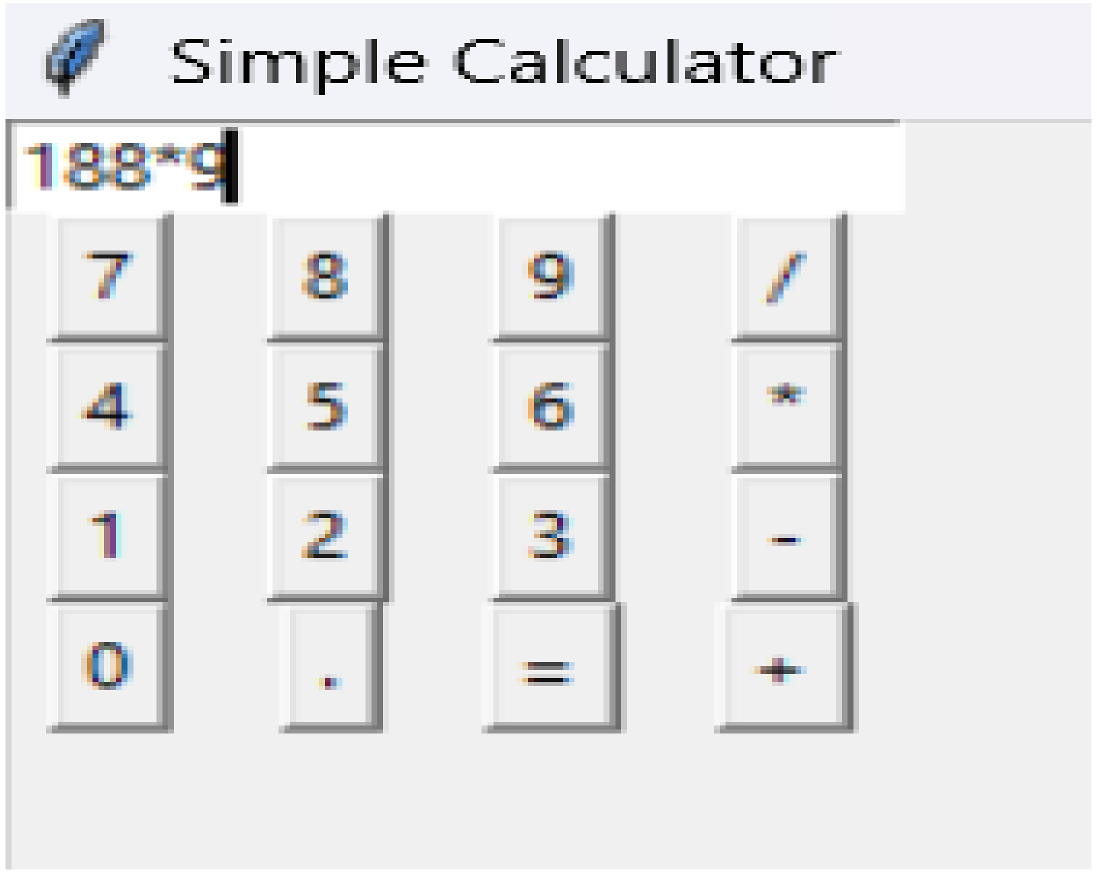
```python
# Create a tkinter window
window = tk.Tk()
window.title("Simple Calculator")

# Create an entry widget for the display
entry = tk.Entry(window, width=20)
entry.grid(row=0, column=0, columnspan=4)
# Define the buttons
buttons = [

    '7',  '8',  '9',  '/',

    '4',  '5',  '6',  '*',

    '1',  '2',  '3',  '-',

    '0',  '.',  '=',  '+'
]
# Create and place the buttons on the grid
row, col = 1, 0
for button in buttons:
   if button == '=':
        tk.Button(window, text=button, command=calculate).grid(row=row, column=col)
```

```
 elif button == 'C':
        tk.Button(window, text=button,
command=clear).grid(row=row, column=col)
    else:
        tk.Button(window, text=button, command=lambda
b=button: button_click(b)).grid(row=row, column=col)
    col += 1
    if col > 3:
        col = 0
        row += 1

# Start the GUI main loop
window.mainloop()
```

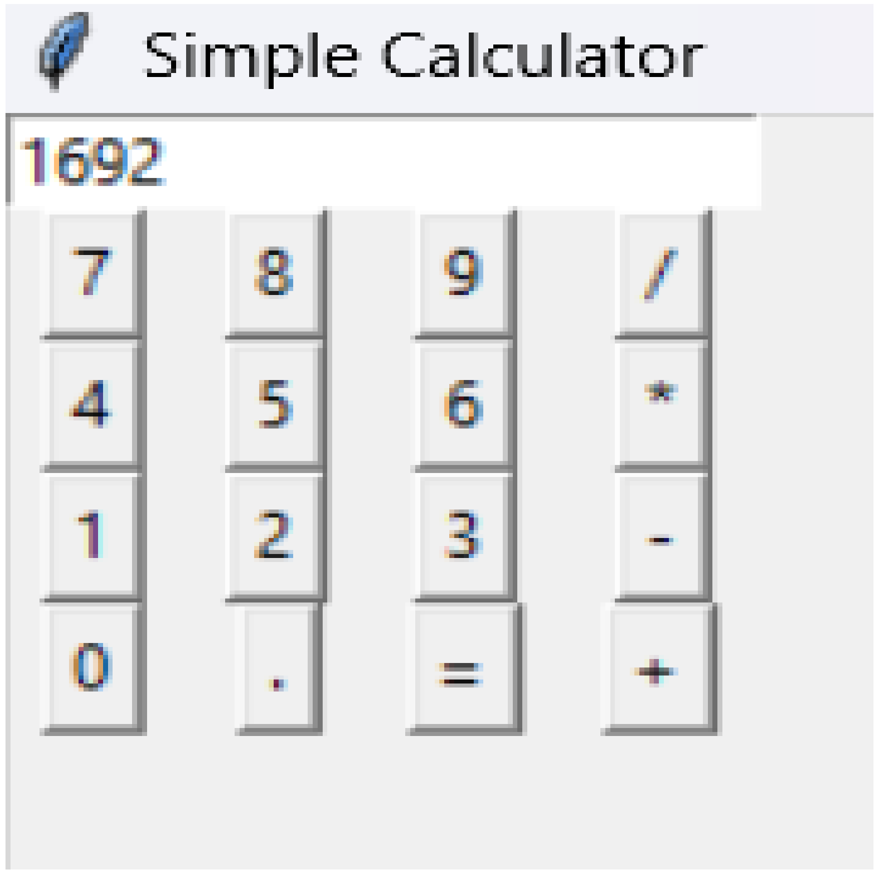
OUTPUT

- ScreenShots

Input



Output



TASK 3:PASSWORD GENERATOR

A password generator is a useful tool that generates strong and random passwords for users. This project aims to create a password generator application using Python, allowing users to specify the length and complexity of the password. User Input: Prompt the user to specify the desired length of the password. Generate Password: Use a combination of random characters to generate a password of the specified length. Display the Password: Print the generated password on the screen .

- **Introduction:**The Password Generator Application is a Python project created using the Tkinter library. It is designed to generate random and secure passwords based on user-defined parameters. The primary goal of this application is to provide users with a simple way to create strong and unique passwords.
- **Requirements:**Create a graphical user interface (GUI) for a password generator.- Allow users to specify the desired password length.- Generate random passwords containing letters, digits, and special characters.- Handle invalid input and provide user feedback**.**
- **Design:**The application features a GUI created with Tkinter, providing a user-friendly interface,Users can enter the desired password length via an entry field, Upon clicking the "Generate Password" button, random passwords are generated using a combination of letters, digits, and special characters**.**

- **Implementation:** The project is implemented in Python, utilizing the Tkinter library for GUI,The `generate_password` function generates random passwords based on user-defined length,User input validation and error handling are implemented to ensure valid password generation.
- **Usage:**1. Run the application.2. Enter the desired password length in the entry field.3. Click the "Generate Password" button.4. The application will display a randomly generated password of the specified length.5. Handle errors or invalid input as prompted by the application.
- **Conclusion:**The Password Generator Application provides a straightforward solution for generating strong and unique passwords. While it meets the basic requirements, there is potential for further enhancement and expansion of features to cater to diverse security needs.

- Code

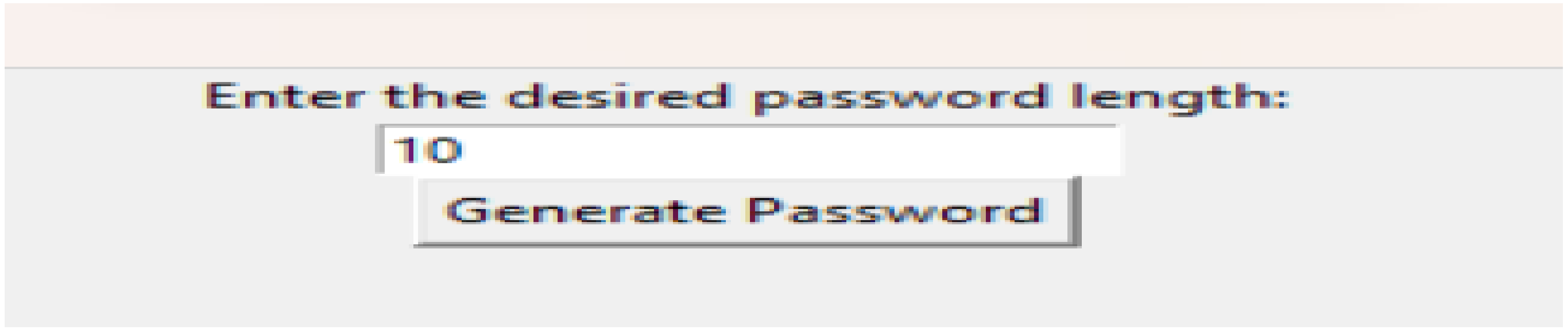
```
import random
import string
import tkinter as tk

def generate_password(length):
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for _ in range(length))
    return password

def generate_password_button_click():
    try:
        password_length = int(length_entry.get())
        if password_length <= 0:
            result_label.config(text="Password length must be greater than 0.")
        else:
            password = generate_password(password_length)
            result_label.config(text="Generated Password: " + password)
   except ValueError:
        result_label.config(text="Invalid input. Please enter a valid number for
password length.")
```

```python
# Create the main window
window = tk.Tk()
window.title("Password Generator")

# Create and place widgets
length_label = tk.Label(window, text="Enter the desired password length:")
length_label.pack()

length_entry = tk.Entry(window)
length_entry.pack()

generate_button = tk.Button(window, text="Generate Password",
command=generate_password_button_click)
generate_button.pack()

result_label = tk.Label(window, text="")
result_label.pack()

# Start the GUI application
window.mainloop()
```

OUTPUT

- ScreenShots

Input

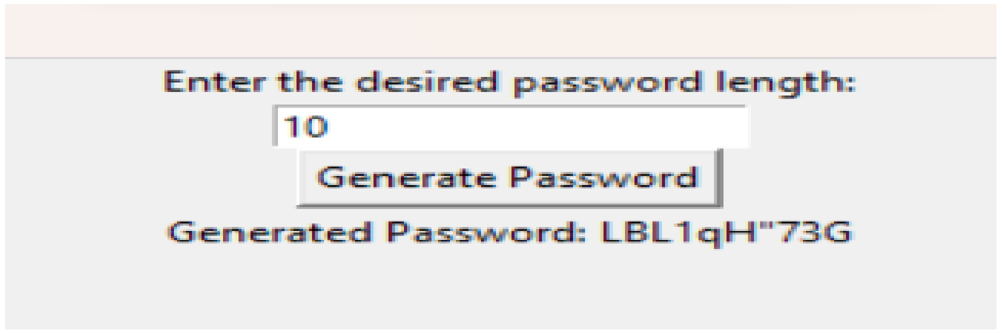


Enter the desired password length:

10

Generate Password

Output

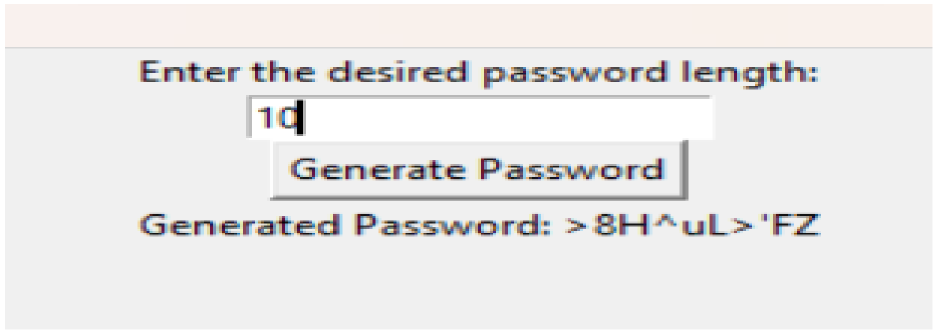


Enter the desired password length:

10

Generate Password

Generated Password: LBL1qH"73G



Enter the desired password length:

10

Generate Password

Generated Password: >8H^uL>'FZ

THANK YOU