#VIRTUNEXA WEEK 1 TASK:

COUNTDOWN TIMER

Overview:

This script is a countdown timer application that provides both a graphical user interface (GUI) and a console-based interface for user interaction. It supports various input formats for specifying durations (e.g., 5m30s, arithmetic expressions like 2*60+30, etc.). Key features include:

- A user-friendly GUI built with tkinter.
- Text-to-speech announcements via the pyttsx3 library.
- Arithmetic evaluation for flexible duration inputs.
- Countdown functionality with pause and resume options.

Dependencies

The following libraries are required to run the script:

1. **tkinter**: For GUI development.
2. **ttk**: For themed widgets in the GUI.
3. **Pillow (PIL)**: To handle image operations for the background.
4. **pyttsx3**: For text-to-speech functionality.
5. **time**: To create delays during the countdown.
6. **re**: To parse and validate time input strings.

Install missing dependencies using pip:

```
pip install pillow pyttsx3      #COMMAND TO INSTALL DEPENDENCIES
```

Components

1. TimeParser Class:

Handles parsing and evaluation of time input strings.

Methods

- **parse_time_input(input_str)**
    - Parses strings like 5m30s, 5m, 30s, or arithmetic expressions.
    - Returns the total duration in seconds.
    - Raises ValueError for invalid formats.

- **evaluate_arithmetic(expr)**
    - Evaluates arithmetic expressions (e.g., 2*60+30).
    - Converts m (minutes) to *60 and removes s (seconds) for compatibility.
    - Ensures only valid mathematical expressions are evaluated.
    - Returns the computed result in seconds.

---

2. CountdownTimerApp Class:

The primary GUI application for the countdown timer.

Attributes

- **root**: The main tkinter window.
- **time_left**: Tracks the remaining time in seconds.
- **running**: Indicates if the timer is active.
- **paused**: Tracks if the timer is paused.
- **engine**: Text-to-speech engine.

Methods

- **__init__(root):**
    - Initializes the GUI, including widgets and layout.
    - Loads a background image and sets up the tkinter frame.
- **start_timer(event=None):**
    - Starts the countdown based on user input.
    - Validates input using TimeParser.
- **update_timer():**
    - Updates the countdown timer every second.
    - Handles the end of the timer and announces "Time's up."
- **announce_time(mins, secs):**
    - Uses text-to-speech to announce the remaining time during the last 5 seconds.
- **pause_timer():**
    - Pauses the countdown.
- **resume_timer():**
    - Resumes the countdown if it was paused.
- **exit_app():**

- Closes the application.

---

3. Console Functionality:

Provides a text-based countdown timer.

countdown_timer_console(duration):

- Accepts a duration string.
- Parses and starts the countdown in the console.
- Displays the timer in real time.

---

4. Main Menu:

A simple command-line menu to choose between:

1. Console timer.
2. GUI timer.
3. Exit.

main_menu():

- Loops until the user exits the application.
- Allows the user to choose between GUI and console modes.

Usage:

Running the Application

Run the script directly:

Python countdown_timer_app.py

Input Formats

- **Time Formats**: 5m30s, 5m, 30s.
- **Arithmetic Expressions**: 2*60+30, 3*60-10.

Options in the Main Menu

1. **Console Mode**:
    - Enter a duration when prompted.
    - The timer runs in the terminal.
2. **GUI Mode**:
    - Launches the graphical interface for the timer.

3. **Exit**:
    - Quits the application.

Features

1. **GUI Timer:**
    - Background image support.
    - Buttons for starting, pausing, resuming, and exiting.
    - Real-time display of the countdown.
    - Announcements for the last 5 seconds and timer completion.
2. **Console Timer:**
    - Minimal, terminal-based interface.
    - Displays countdown in real time.
3. **Text-to-Speech:**
    - Announces time remaining for the last 5 seconds.
    - Says "Time's up!" upon completion.
4. **Input Parsing:**
    - Supports natural time formats and arithmetic expressions.
    - Rejects invalid inputs with error messages.

Error Handling:

1. Invalid inputs (e.g., abc, -5s) result in:
    - **Console**: Prints an error message.
    - **GUI**: Displays an error dialog.
2. Arithmetic errors or negative durations are handled gracefully.