Project Sundial

Final Project

CSE 4202

Team Members

- **Mahajabin Tabassum**
  - ID: 200041132 Group: 1B
- **Jarin Tasnim Hridy**
  - ID: 200041108 Group: 1B
- **Amina**
  - ID: 200041155 Group: 1A

Project Sundial

The project is a clock enabling one to utilize time by using the basic features that one would find on a standard digital clock with a unique feature of Voice Generator.

C programming language is used to design the project.

Project Sundial

Setting Alarm	Timer	Stop Watch	Time Converter
• Take user input in Label and Time • Read out the label while ringing • Snooze for 5 minutes • Resetting alarm	• Take input in HH:MM:SS • Countdown • Color the terminal - Yellow after 50% - Red after 75% • Read aloud "Time is Over."	• Start • Propagate time in MM : SS : milliSec • Reset • Pause • Exit	• Connect to a file of Place and time difference • Follow GMT convention • Read Out the time

Project Sundial

Unimplemented feature:

- Display of Clock

Extra features added to compensate:

- Use of Lavenstein distance in Converter
- Substring matching in Converter
- Reset and Pause options in Stop watch and Alarm

Project Sundial

Work Distribution:

Mahajabin Tabassum	• Addition of eSpeak Voice Synthesizer • Combining works of other members.
Jarin Tasnim Hridy	• Implementation of Setting Alarm • Implementation of Stop watch
Amina	• Implementation of Timer • Implementation of Time Converter

Setting Alarm

Challenges	Solution
1. Alarm was not ringing continuously.	1. kbhit() function was used to solve the problem.
2. Time was running slower than real time for using sleep().	2. A user defined delay() function was used and time was managed by taking OS time using struct tm, time() and localtime() functions.
3. While giving input, if character was given, the program would crash.	3. Input is taken as string and filtered in digits only.

Stop Watch

Challenges	Solution
1. While displaying input in milliseconds, the stopwatch was running slower.	1. Clock() function was used which returns the approximate processor time based on speed of processor. Besides, '\r' which is carriage return escape character was used to avoid much glitching as '\r' can move the cursor back to the beginning of the line to overwrite it with new contents.

Timer

Challenges	Solution
1. If user presses timer accidentally, he/she can't come back to menu without setting a minimum timer,	1. kbhit() was added to remove the problem.

Time Converter

Challenges	Solution
1. If user does any spelling mistake, he won't see any results. 2. In the list more than one country can contain same name that could cause confusion.	1. lavenstein distance was used to find closeness between two words and help the user to detect correct place. 2. the substrings detection was used.

Project Sundial

Demonstration


thank you