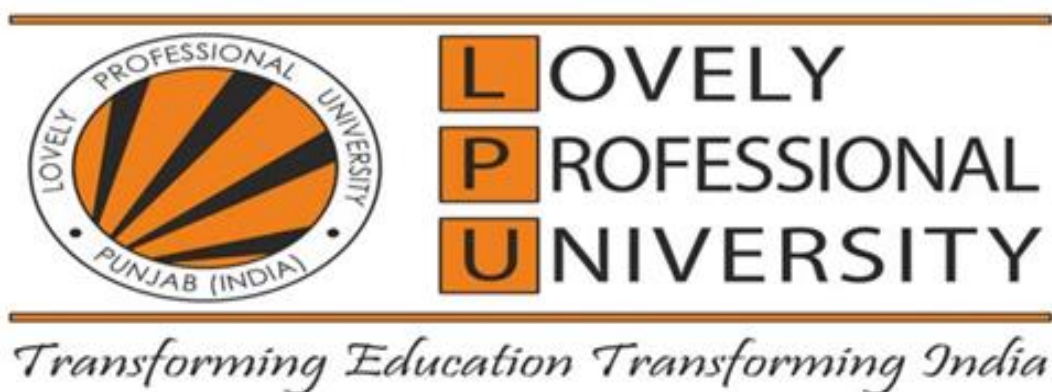


Alarm Tool

END TERM REPORT

By

Name	Registration No.	Roll No.	Section No.
Sourav Kumar Kalita	11803718	B65	K18MS
Kusu Mohan Krishna Hruday	11803784	B66	K18MS
Gautam Kumar	11815892	B67	K18MS
Sonu Kumar	11815568	B68	K18MS



**Department of Intelligent Systems
School of Computer Science Engineering
Lovely Professional University, Jalandhar
April – 2020**

Student Declaration

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

SOURAV KUMAR KALITA
Roll No.: B65

MOHAN KRISHNA
Roll No.: B66

GAUTAM KUMAR
Roll No.: B67

SONU KUMAR
Roll No.: B68

TABLE OF CONTENTS

Title	Page No.
1. Background.....	05
2. Motivation.....	06
3. Outcome.....	06
4. Concrete Goals And Objectives.....	06
5. Project Description.....	07
6. Description Of Project In Terms Of Work Among Student.....	08
7. Technologies And Framework To Be Used.....	08
8. Code and Output.....	09

BONAFIDE CERTIFICATE

Certified that this project report “Alarm Tool” is the bonafide work of Mr. SONU KUMAR, Mr. GAUTAM KUMAR & his team who carried out the project work under my supervision.

Name of Supervisor: Ms. Jasleen Kaur

ID of Supervisor=25340

Department of Intelligent Systems

1. Background

An **alarm clock** (or sometimes just an **alarm**) is a clock that is designed to alert an individual or group of individuals at a specified time. The primary function of these clocks is to awaken people from their night's sleep or short naps; they are sometimes used for other reminders as well. Most use sound; some use light or vibration. Every clock has to specify the time at which the alarm will ring. Alarm clocks are also used in mobile phones, watches, and computers.



2. Motivation

The motivation for doing this project was primarily an interest in undertaking a challenging project in an interesting area of research. The opportunity to learn about a new area of computing not covered in lectures was appealing. This area is possibly an area that I might study at next level of AI(Automation of Clocks).

3. Outcome of the project

Alarm tool has been created which will further help users to set the alarm in enhanced way in which system will give suggestions accordingly with some music recommendations.

4. Concrete goals and objectives:

To create an alarm clock with improved timing system and ease to use for the users.

5. Project Description:

import webbrowser

The `webbrowser` module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the `open()` function from this module will do the right thing. The script **`webbrowser`** can be used as a command-line interface for the module. It accepts a URL as the argument. It accepts the following optional parameters: `-n` opens the URL in a new browser window, if possible; `-t` opens the URL in a new browser page ("tab"). The options are, naturally, mutually exclusive.

import random

You can generate random numbers in Python by using random module. Python offers random module that can generate random numbers. These

are pseudo-random number as the sequence of number generated depends on the seed. If the seeding value is same, the sequence will be the same.

import time

This module provides various time-related functions. In which **datetime** and **calendar** module are very important.

import os

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see `open()`, if you want to manipulate paths, see the `os.path` module, and if you want to read all the lines in all the files on the command line see the `inputoutput()` module. For creating temporary files and directories see the `tempfile()` module, and for high-level file and directory handling see the `shutil()` module.

All functions in this module raise `OSError` (or subclasses thereof) in the case of invalid or inaccessible file names and paths, or other arguments that have the correct type, but are not accepted by the operating system.

6. Description of Work Division in terms of Roles among Students:

Sonu Kumar (Report, loom video, webbrowser module)

Gautam Kumar (Report, time&date module)

Kusu Mohan Krishna Hruday (os module, report)

Sourav Kumar Kalita (time module, report)

7. Technologies and Framework to be used

We have used Pycharm and Anaconda (jupyter), and chrome browsing for the completion of the project.

8. Code:

```

import datetime
import os
import time
import random
import webbrowser

print("=====
=====")
print()
print("                WELCOME TO ALARM TOOL SYSTEM
")
print()
print("=====
=====")
print("Hello there! Set an Alarm \n")

# If the video URL file doesn't exist, creates one
if not os.path.isfile("youtube_alarm_videos.txt"):
    print('Creating "youtube_alarm_videos.txt"...')
    with open("youtube_alarm_videos.txt", "w") as alarm_file:
        alarm_file.write("https://youtu.be/h1C-qA7BkP0")

def check_alarm_input(alarm_time):
    """Check to see if the user has entered during a legitimate alarm time*"""
    if len(alarm_time) == 1: # [Hour] Format.
        if alarm_time[0] < 24 and alarm_time[0] >= 0:
            return True
    if len(alarm_time) == 2: # [Hour:Minute] Format.
        if alarm_time[0] < 24 and alarm_time[0] >= 0 and \
            alarm_time[1] < 60 and alarm_time[1] >= 0:
            return True
    return False

# Get the user input for the alarm time
print("Set a time for the alarm (Ex. 10:30)")
while True:
    alarm_input = input(">> ")
    try:
        alarm_time = [int(n) for n in alarm_input.split(":")]
        if check_alarm_input(alarm_time):
            break

```



```

        else:
            raise ValueError
    except ValueError:
        print(">> ERROR : Please Enter time in HH:MM or HH:MM:SS format only
<<")

# Convert alarm time from [H:M] or [H:M:S] to seconds
seconds_hms = [3600, 60, 1] # No. of seconds in an Hour, Minute, and Second
alarm_seconds = sum([a*b for a,b in zip(seconds_hms[:len(alarm_time)], alarm_time)])

# Get this time of the day in seconds
now = datetime.datetime.now()
current_time_seconds = sum([a*b for a,b in zip(seconds_hms, [now.hour, now.minute,
now.second])])

# Calculate the number of seconds until alarm burst
time_diff_seconds = alarm_seconds - current_time_seconds

if time_diff_seconds < 0:
    time_diff_seconds += 86400 # number of seconds during on a daily basis

# Display the number of some time until the alarm burst
print("The Alarm set to travel off in %s" %
datetime.timedelta(seconds=time_diff_seconds))

# Sleep until the alarm burst
time.sleep(time_diff_seconds)

# Time for the alarm to travel off
print("Wake Up! Wake Up!")

# Load list of possible video URLs
with open ("youtube_alarm_videos.txt", "r") as alarm_file:
    videos = alarm_file.readlines()

# Open a random video from the list
webbrowser.open(random.choice(videos))

```

Output:

```

# Time for the alarm to travel off
print("\nWake Up! Wake Up!")

# Load list of possible video URLs
with open("youtube_alarm_videos.txt", "r") as alarm_file:
    videos = alarm_file.readlines()

# Open a random video from the list
webbrowser.open(random.choice(videos))

=====

--:  WELCOME TO ALARM TOOL SYSTEM  :--

=====

Hello there! Set an Alarm

Set a time for the alarm (Ex. 10:30)

>> 17:56

The Alarm set to travel off in 0:00:43

Wake Up! Wake Up!

Out[1]: True

```

- - - -