A

Project Report

On

MOOD ANALYSIS AND HEALTH MONITORING SYSTEM

Submitted in partial fulfillment of the requirement for the IVth semester

Bachelor of Technology

By

Harshit Waldia

2161170

Under the Guidance of

Mr. Lalit Kumar

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI,

DISTRICT- NAINITAL-263132

2022- 2023

STUDENT'S DECLARATION

I, **Harshit Waldia** here-by declare the work, which is being presented in the project, entitled "**MOOD ANALYSIS AND HEALTH MONITORING SYSTEM**" in partial fulfillment of the requirement for the award of the degree **B. Tech** in the session **2022-2023**, is an authentic record of our own work carried out under the supervision of "**Mr. Lalit Kumar", Assistant Professor, Department of CSE, Graphic Era Hill University, Bhimtal.**

The matter embodied in this project has not been submitted by us for the award of any other degree.

Date: …… …….

(Harshit Waldia)

CERTIFICATE

The project report entitled "MOOD ANALYSIS AND HEALTH MONITORING SYSTEM" being submitted by Harshit Waldia to Graphic Era Hill University Bhimtal Campus for the award of Bonafede work conducted by him. He have worked under my guidance and supervision and fulfilled the requirement for the submission of report.

(Mr. Lalit Kumar) **(Dr. Ankur Bisht)**

Project Guide **(HOD, CSE Dept.)**

ACKNOWLEDGEMENT

We take immense pleasure in thanking Honorable **"Mr. Lalit Kumar"** (**Assistant Professor, CSE, GEHU Bhimtal Campus**) to permit me and conduct this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me everything that we need. We again want to extend thanks to our President **"Prof. (Dr.) Kamal Ghanshala"** for providing us all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to Professor **"Dr. Manoj Chandra Lohani"** (Director Gehu Bhimtal),

other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this thesis.

Finally, yet importantly, we would like to express my heartiest thanks to our beloved parents,

for their moral support, affection and blessings. We would also like to pay our sincere thanks to all our friends and well-wishers for their help and wishes for the successful completion of this research.

Harshit Waldia

ABSTRACT

"The Mental Health Monitoring And Stress Relieving Program "

To develop a software for solving mental fatigue of a person in their day-to-day busy life. Mental health is an important public health concern worldwide and should be a prominent part of the healthcare industry. However, the pace of development in this field appears to be quite slow. Recently AI techniques have attracted a lot of attention in different industries, including mental health. With the help of advanced AI techniques and Non - Machine Learning Algorithms, a personalized care that focuses on providing emotional support catered to a specific individual has been facilitated. In this paper, we analyze various systems for mental health monitoring namely virtual counselling, mood therapy, and motivational and gaming systems by reviewing the algorithms and parameters used in each system. We conclude by proposing a system which combines the above-mentioned systems and is planned to provide personalized mental care.

CONTENTS

INTRODUCTION

The program called "The Mental Health Monitoring and Stress Reduction “to create a software that addresses a person's daily stress-related mental tiredness. Because it concerns everyone's health, mental illness should take center stage in the healthcare sector. However, it seems like this field is developing at a somewhat modest rate. Recently, various businesses, including the field of mental health, have paid a lot of attention to AI techniques. Advanced AI methods and non-machine learning algorithms have made it possible to provide individualized care that is focused on offering emotional support tailored to a particular person. In this research, we explore numerous systems for tracking mental health, including virtual counselling, mood therapy, and gaming and incentive systems.

WHAT IS MENTAL HEALTH?

Mental health refers to a person's emotional, psychological, and social well-being. It encompasses how individuals think, feel, and behave, as well as their ability to cope with everyday challenges and stressors. Good mental health allows for positive self-esteem, healthy relationships, and the ability to make sound decisions. Conversely, poor mental health can lead to various difficulties, such as depression, anxiety, addiction, and impaired functioning in daily life. It is important to prioritize mental health through self-care, seeking support from loved ones or professionals, and practicing strategies like stress management, mindfulness, and maintaining a balanced lifestyle to promote overall well-being.

TOOLS/ PLATFORM, HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATIONS

Platform

- Windows 7/8/10/11

Hardware Requirement Specification

Criterion	Description
Disk Space	500 MB disk space for Android Studio, at least 1.5 GB for Android SDK, emulator system images, and caches
RAM	3GB RAM minimum, 8 GB RAM recommended, plus 1GB for the Android Emulator
Python Version	Python 3.7 and above

Table 1

Software Requirement Specification

Any version of Python mentioned above.

LIBRARIES USED IN MOOD ANALYSIS AND STRESS MANAGEMENT

There are 3 in – built libraries used for working with this system and analysis.

1. PIL: The PIL (Python Imaging Library) is a popular library for manipulating and processing images in Python. It provides a wide range of functions for tasks such as opening, saving, and resizing images, as well as applying filters and performing basic image enhancements. With PIL, developers can easily incorporate image processing capabilities into their Python applications.

2. MYSQL CONNECTOR: The MySQL Connector library is a Python module that allows developers to connect and interact with MySQL databases. It provides a simple and efficient way to execute SQL queries, manage database connections, and retrieve results. With the MySQL Connector library, Python applications can seamlessly integrate with MySQL databases for data storage and retrieval operations.

3. WEBBROWSER: The web browser library in Python provides a high-level interface for opening web browsers programmatically. It allows developers to launch URLs, open specific web pages, and perform basic web browsing operations from within Python scripts. This library enables easy automation of web browsing tasks and integration of web content into Python applications.

Apart from these in-built libraries 4 self-built modules and 1 html webpage were also used :

MUSIC_3

- GUI BASED MUSIC PLAYER
- TKINTER BASED

.

MODULE_MOOD

- DATA SET OF ALL MOODS
- MOOD LIBRARY

PROPERTIES OF PIL(PYTHON IMAGING LIBRARY)

The PIL (Python Imaging Library) is a popular library for manipulating and processing images in Python. It provides a wide range of functions for tasks such as opening, saving, and resizing images, as well as applying filters and performing basic image enhancements. With PIL, developers can easily incorporate image processing capabilities into their Python applications.

ADVANTAGE

Comprehensive Image Processing: PIL offers a wide range of functions and methods for image manipulation and processing. It allows developers to perform tasks such as opening, saving, resizing, cropping, and applying various filters to images. This comprehensive set of features makes it a powerful tool for working with images in Python.

Cross-Platform Compatibility: PIL is a cross-platform library, which means it can be used on different operating systems such as Windows, macOS, and Linux. This makes it a versatile choice for developers working on different platforms, as they can utilize the same codebase to perform image-related tasks.

DISADVANTAGE

Limited Development and Maintenance: PIL development has been stagnant since 2011, and it does not support Python 3.5 and above. Although there are alternative libraries, such as Pillow, which is a fork of PIL and is actively maintained, the original PIL library lacks ongoing development and updates.

Limited File Format Support: PIL supports a decent range of image file formats; however, it lacks support for some modern formats such as WebP. This limitation can be problematic when working with specific file formats that may be required in certain projects.

CODE

```
# MOOD ANALYSIS

### Importing Modules (Self - Made) and In-built Libraries

from module_Mood import check_mood

import mysql.connector


import PIL

from PIL import Image


import module_img

from module_img import *


import music_3

from music_3 import *


import module_music as music

from music_3 import MusicPlayer


import webbrowser

import re

#### Creating connection object

mydb = mysql.connector.connect(

        host = "localhost",

        user = "root",

        password = "Harshit",

   database ="healthsys"

)
```

```
# checking connection establishment
if (mydb.is_connected):
    print("connection established")

# Printing the connection object
print(mydb)
### Main Program
USER=input("NAME >>> ")



__Name__=USER.title() #new code edited resently

import re
if(bool(re.match('^[a-zA-Z" "]*$',__Name__))==True):
    print("valid")

else:
    print("invalid")
    quit()
### USER'S DATA
print("HI !! ",__Name__," HOW ARE YOU FEELING ....")

which_mood = {
    'happy':0,
    'sad':1,
    'angry':2,
```

```
    'gaming':3,
    'neutral':4,
    'all':5,
    'foody':6
}
while True:
    try:
        mood = input(">>> ")
        mood=mood.lower()


        if not check_mood(mood,which_mood['all']):
            print('invalid')
        else :
            print("oh!",mood)
            break
    except ValueError:
        print("Provide an string value...")
        continue
while True:
  try:
    age = int(input("Age: "))
    if age>1 and age<100:
      print("Age entered successfully...")
      break;
    else:
      print("Age should be >1 and <100...")
```

```
  except ValueError:
    print("Provide an integer value...")
    continue


while True:
    try:
        gender = str(input('Gender:')).lower()
        if gender in ['male', 'female']:
            break
        else:
            print("error")
    except ValueError:
        print("Provide an string value...")
        continue


address=input("City:")


mood=mood.lower()
### Condition For Health
if check_mood(mood,which_mood['sad']) or check_mood(mood,which_mood['angry']):


    health = "bad"



elif check_mood(mood,which_mood['happy']) or check_mood(mood,which_mood['gaming'])
or check_mood(mood,which_mood['neutral']) or check_mood(mood,which_mood["foody"]):
```

```
    health = "good"
### creating cursor{establishes conn between sql and python IDLE}
cursor=mydb.cursor()


cursor.execute ("INSERT INTO moodlog (Name,Age,Gender,City,Mood,Health) VALUES
('{}','{}','{}','{}','{}','{}');".format(__Name__,age,gender,address,mood,health))



mydb.commit()



print("Data Entered Successfully!")
### STARTING OF MOOD DEPENDENCIED
if check_mood(mood,which_mood['sad']):


    print("SO, what can i do to make your day !!\n")


    print("I Can do some cool things ,,\n1_Play music \n2_Give you motivation \n3_Make you
laugh\n")


    lang = int(input(">>>"))
    match lang :
        case 1:
            print(" So, you like to hear some music : ")
            __ans__=input("Y/N\n>>>")
            __ans__=__ans__.lower()
```

```
if (__ans__=="y" or __ans__== "yes"):

    music_player = MusicPlayer()
    music_player.run()

else:
    print("Would you like to see some memes ....?\n ")
    __ans__=input("Y/N\n>>>")
    __ans__=__ans__.lower()


    if (__ans__=="y" or __ans__=="yes"):
        import webbrowser

        def open_html_page(file_path):
          webbrowser.open(file_path, new=2)

        # Example usage
        html_file_path = r"C:\Users\harsh\mini_project_2\Untitled-1.html"
        open_html_page(html_file_path)

    else:
        print("Boy you really need some motivation ? ")
        __ans__=input("Y/N\n>>>")
        __ans__=__ans__.lower()
        if (__ans__=="y" or __ans__=="yes"):
            import module_music as music
```

```
            music.motivational()


    case 3:
        print("So ,you like to see some memes ....?\n ")
        __ans__=input("Y/N\n>>>")
        __ans__=__ans__.lower()


        if (__ans__=="y" or __ans__=="yes"):
            import webbrowser
            def open_html_page(file_path):
                webbrowser.open(file_path, new=2)


            # Example usage
            html_file_path = r"C:\Users\harsh\mini_project_2\Untitled-1.html"
            open_html_page(html_file_path)


        else:
            #(__ans__=="n" or __ans__=="no"):
            print("THERE IS NOTHING I CAN DO NOW .....!!!\n You really need to
seek G.O.D's attention ")


    case 2:
        print("So, motivation !!\n Here we GO!! ")
        import module_music as music
        music.motivation()
```

```
        case _:
            #(__ans__=="n" or __ans__=="no"):
            print("THERE IS NOTHING I CAN DO NOW .....!!!")


elif check_mood(mood,which_mood['happy']):

    print("Hope you make someone SMILE !!\n")

    print("SO, HOW can i make your day more joyful , here are certain things i can do - i can play music  , make you laugh , or i can give you motivation \n")

    print("Would you like to hear some music ???\n ")

    __ans__=input("Y/N\n>>>")
    __ans__=__ans__.lower()

    if (__ans__=="y" or __ans__== "yes"):
        import module_music as music
        from music_3 import MusicPlayer


        music_player = MusicPlayer()
        music_player.run()


    else:
```

```
    pass


print("Would you like to see some memes ....?\n ")
__ans__=input("Y/N\n>>>")
__ans__=__ans__.lower()


if (__ans__=="y" or __ans__=="yes"):
    import webbrowser
    def open_html_page(file_path):

        webbrowser.open(file_path, new=2)

            # Example usage
        html_file_path = r"C:\Users\harsh\mini_project_2\Untitled-1.html"
        open_html_page(html_file_path)


else:
    pass
print(" DO need some motivation ? ")
__ans__=input("Y/N\n>>>")
__ans__=__ans__.lower()


if (__ans__=="y" or __ans__=="yes"):


    import module_music as music
    music.motivational()
```

```
else:
    pass

print("Your chooice is : \n1- Play GAME \n2-  GAMING MUSIC")

ch=int(input(">>>"))

#gaming web site

def play_game():
    import webbrowser
    webbrowser.open('http://poki.com', new=2)

#memes

def memes():
    import webbrowser
    def open_html_page(file_path):
        webbrowser.open(file_path, new=2)

            # Example usage
    html_file_path = r"C:\Users\harsh\mini_project_2\Untitled-1.html"
    open_html_page(html_file_path)



#gaming music
```

```
    def gaming_music():
        music.gaming()



    if(ch==1):
        play_game()



    elif(ch==2):
        import module_music as music
        gaming_music()
    else :

      print("ALPHA TO CHARLIE !!! ITS MAY_DAY BUT YOU ARE NOOB.....")



elif check_mood(mood,which_mood["gaming"]):
    print("Alpha to Charli ! Alpha to Charli!!! ")
    print("\n")
    print("Your chooice is : \n1- Play GAME \n2- MEMES\n3- GAMING MUSIC")
```

```
#gaming web site


def play_game():
    import webbrowser
    webbrowser.open('http://poki.com', new=2)


#memes


def memes():


    import webbrowser
    def open_html_page(file_path):
        webbrowser.open(file_path, new=2)


            # Example usage
    html_file_path = r"C:\Users\harsh\mini_project_2\Untitled-1.html"
    open_html_page(html_file_path)


#gaming music


def gaming_music():
    music.gaming()



_ch_made_=int(input(">>>"))


match _ch_made_:
```

```
        case 1:
            play_game()



        case 2:
            memes()


        case 3:
            import module_music as music
            gaming_music()
        case _:
            print("ALPHA TO CHARLIE !!! ITS MAY_DAY BUT YOU ARE NOOB.....")


elif check_mood(mood,which_mood["foody"]):

    websites = {
    "pizza": "https://www.dominos.co.in",
    "burger": "https://www.mcdonalds.co.in",
    "samosa": "https://www.haldirams.com",
    "biryani": "https://www.behrouzbiryani.com",
    "dosa": "https://www.saravanabhavan.com",
    "chaat": "https://www.bikanervala.com",
    "idli": "https://www.udupi.com",
    "vada pav": "https://www.gajananvadapav.com",
    "pani puri": "https://www.lokmithbhavan.com",
    "paratha": "https://www.paranthe.com",
```

"chole bhature": "https://www.kwalityrestaurant.in",

"butter chicken": "https://www.punjabgrill.in",

"samosa pav": "https://www.chaipeenilane.com",

"misal pav": "https://www.bedekar.com",

"dhokla": "https://www.khaman.com",

"tandoori chicken": "https://www.nandos.co.in",

"kulfi": "https://www.kuremal.com",

"vada sambhar": "https://www.sagar.com",

"kathi roll": "https://www.kusumrolls.com",

"pav bhaji": "https://www.sardarspavbhaji.com",

"rasgulla": "https://www.kc-das.com",

"momo": "https://www.wowmomo.in",

"jalebi": "https://www.rawatsweets.com",

"papdi chaat": "https://www.kanha.com",

"chicken biryani": "https://www.biryanihouse.com",

"masala dosa": "https://www.malleshwarmasaladosa.com",

"rabri": "https://www.halwakadhai.com",

"rasgulla": "https://www.kcda.com",

"kathi roll": "https://www.kusumrollcenter.com",

"bhelpuri": "https://www.swastikbhelpuri.com",

"taco": "https://www.tacobell.com",

"pasta": "https://www.olivegarden.com",

"ice cream": "https://www.baskinrobbins.com",

"steak": "https://www.outback.com",

"ramen": "https://www.ichiraku.com",

"fried chicken": "https://www.kfc.com",

"samosa": "https://www.haldirams.com"

```
}


food_item = input("What would you like to have : ")
if food_item in websites:
    website = websites[food_item]
    print("Website for", food_item, "is:", website)


    import webbrowser
    webbrowser.open(website,new=1)


else:
    print("Website for", food_item, "is not available.")


    site=input("Enter Your Favourite site with domain : ")
    import webbrowser
    webbrowser.open(site)
```

CODE

```
import tkinter as tk
from tkinter import filedialog
import pygame
import os


class MusicPlayer:
    def __init__(self):
        self.file_paths = []
        self.current_song_index = 0

        pygame.init()
        pygame.mixer.init()

        self.window = tk.Tk()
        self.window.title("Music Player")
        self.window.configure(bg="#EEEEEE")
        self.window.geometry("400x200")

        self.create_buttons()

    def create_buttons(self):
        browse_button = tk.Button(self.window, text="Browse Folder",
command=self.browse_folder)
        browse_button.grid(row=0, column=0, padx=10, pady=10)
```

```
        previous_button = tk.Button(self.window, text="Previous",
command=self.play_previous_song)
        previous_button.grid(row=1, column=0, padx=10, pady=10)

        play_button = tk.Button(self.window, text="Play", command=self.play_music)
        play_button.grid(row=1, column=1, padx=10, pady=10)

        pause_button = tk.Button(self.window, text="Pause", command=self.pause_music)
        pause_button.grid(row=1, column=2, padx=10, pady=10)

        resume_button = tk.Button(self.window, text="Resume", command=self.resume_music)
        resume_button.grid(row=1, column=3, padx=10, pady=10)

        stop_button = tk.Button(self.window, text="Stop", command=self.stop_music)
        stop_button.grid(row=1, column=4, padx=10, pady=10)

        next_button = tk.Button(self.window, text="Next", command=self.play_next_song)
        next_button.grid(row=1, column=5, padx=10, pady=10)

    def browse_folder(self):
        folder_path = filedialog.askdirectory()
        if folder_path:
            self.file_paths = self.get_mp3_files(folder_path)
            if self.file_paths:
                self.current_song_index = 0
                self.play_music()
```

```
def get_mp3_files(self, folder_path):
    mp3_files = []
    for root, dirs, files in os.walk(folder_path):
        for file in files:
            if file.endswith(".mp3"):
                mp3_files.append(os.path.join(root, file))
    return mp3_files

def play_music(self):
    pygame.mixer.music.load(self.file_paths[self.current_song_index])
    pygame.mixer.music.play()

def pause_music(self):
    pygame.mixer.music.pause()

def resume_music(self):
    pygame.mixer.music.unpause()

def stop_music(self):
    pygame.mixer.music.stop()

def play_next_song(self):
    self.current_song_index += 1
    if self.current_song_index >= len(self.file_paths):
        self.current_song_index = 0
    self.play_music()
```

```
    def play_previous_song(self):
        self.current_song_index -= 1
        if self.current_song_index < 0:
            self.current_song_index = len(self.file_paths) - 1
        self.play_music()

    def run(self):
        self.window.mainloop()
        pygame.quit()

if __name__ == "__main__":
    music_player = MusicPlayer()
    music_player.run()
```

TESTING

TEAM INTERATION

The following describes the level of team interaction necessary to have a successful product.

- The test team will work closely with the Development team to achieve a high quality design and user interface specifications based on customer requirements. The test team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.

- The Test team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.

- Since the application with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

TESTING OBJECTIVE

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program.

ENHENCEMENTS

INTRODUCTION to framework and Django, the future more software companies will hire this software program because now a days the need for the speed in the day-to-day life has become essential. As competition increases, companies by considering outdated version, they develop more efficient versions for individual success. In future we can make a link of this project with networking. We can also convert this project in Hibernate. Hibernate is a framework. It store large amount of database.

CONCLUSION

We have successfully implemented MOOD ANALYSIS AND STRESS RELIEVING using Python .

REFERENCES

1. https://book.pythontips.com/en/latest/
2. https://docs.python.org/3/