PROJECT REPORT

PROSENSIA

TEAM MEMBERS:
DANIAL AFRIDI (LEADER)
AYESHA LIAQAT (MEMBER)
EMAN ZEB (MEMBER)

[Email address]

Project Report:

Mood Tracker and Personal Growth Assistant

1. Introduction:

The project, titled **Mood Tracker and Personal Growth Assistant**, was developed by a dedicated team of three members: Danial (Team Leader), Ayesha Liaqat, and Eman Zeb. The project aims to assist users in tracking their daily moods, setting personal growth goals, and receiving mood-boosting tips based on sentiment analysis of their diary entries.

2. Project Overview:

2.1. Purpose

The main purpose of this project is to provide users with a personal diary application that not only helps them document their daily thoughts and feelings but also offers insights and suggestions to improve their well-being. The application includes features like sentiment analysis of diary entries, goal setting, mood analysis, badge awarding for achievements, and daily reminders.

2.2. Features

The application offers a range of features designed to enhance the user experience:

- **Diary Entries:** Users can add daily diary entries, which are analyzed for sentiment.
- **Mood Analysis:** Sentiment analysis is performed on each entry to categorize the user's mood as Positive, Negative, or Neutral.
- **Goal Setting:** Users can set and view personal growth goals.
- **Achievement Badges:** Badges are awarded based on user activity, such as consistent journaling or maintaining a positive mood streak.
- **Mood-Boosting Tips:** The application provides random tips to help improve the user's mood.
- **Mood Graph:** A simple text-based graph displays trends in the user's mood over time.

3. Team Roles and Contributions:

3.1. Danial - Team Leader

Danial led the team, focusing on the overall methodology of the project. He ensured that the project was completed on time and that the objectives were met. His leadership and guidance were crucial in keeping the team focused and organized.

3.2. Ayesha Liaqat - Feature Development

Ayesha was responsible for the development and implementation of the features within the application. She worked diligently to ensure that each feature was functional and user-friendly. Her attention to detail and commitment to excellence were evident in the quality of the features she developed.

3.3. Eman Zeb - Development Techniques

Eman focused on the development techniques used in the project. She was responsible for ensuring that the code was optimized and that best practices were followed. Her technical expertise and problem-solving skills were instrumental in overcoming challenges during development.

4. Team Dynamics:

Ayesha and Eman both exhibited exemplary behavior throughout the project. Their supportive nature and willingness to collaborate made for a positive working environment. The team worked well together, with each member contributing their unique skills and expertise to the project.

5. PROJECT CODE:

```
import datetime

import random

import tkinter as tk

from tkinter import messagebox, simpledialog, Text


# Initialize data storage

entries = []

goals = []

badges = set()


# Mood suggestions

suggestions = {

    'Positive': ['Keep up the good work!', 'Continue with your current activities.'],

    'Negative': ['Try going for a walk.', 'Consider talking to a friend or family member.', 'Engage in a hobby you enjoy.'],
```

```
    'Neutral': ['Try something new or different today.', 'Reflect on what you are grateful for.']
}

mood_boosting_tips = [
    "Take a short walk outside.",
    "Listen to your favorite music.",
    "Try meditation or deep breathing exercises.",
    "Write down three things you're grateful for.",
    "Reach out to a friend or loved one."
]

# Add a diary entry
def add_entry(description):
    """Add a diary entry with mood analysis."""
    date = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    mood = analyze_sentiment(description)
    entries.append({
        'date': date,
        'description': description,
        'mood': mood
    })

    # Check for badges
    award_badges()
```

```
    messagebox.showinfo("Entry Added", f"Entry added. Today's mood: {mood}\nSuggestion: {', '.join(suggestions.get(mood, ['No suggestions available']))}")


# Add a goal
def add_goal(goal):
    """Add a personal growth goal."""
    goals.append({
        'date': datetime.datetime.now().strftime("%Y-%m-%d"),
        'goal': goal
    })
    messagebox.showinfo("Goal Added", "Goal added.")


# Analyze entries
def analyze_entries():
    """Analyze sentiment and provide suggestions."""
    if not entries:
        messagebox.showinfo("Analyze Entries", "No entries to analyze.")
        return

    analysis_text = ""
    for entry in entries:
        mood = entry['mood']
        analysis_text += f"Date: {entry['date']}\nSentiment: {mood}\nSuggestion: {', '.join(suggestions.get(mood, ['No suggestions available']))}\n{'-' * 40}\n"

    show_text_window("Analyze Entries", analysis_text)
```

```
# Simple sentiment analysis function
def analyze_sentiment(text):
    positive_words = {'happy', 'joy', 'love', 'great', 'fantastic', 'good', 'excited'}
    negative_words = {'sad', 'angry', 'hate', 'terrible', 'bad', 'depressed'}

    positive_count = sum(word in positive_words for word in text.lower().split())
    negative_count = sum(word in negative_words for word in text.lower().split())

    if positive_count > negative_count:
        return 'Positive'
    elif negative_count > positive_count:
        return 'Negative'
    else:
        return 'Neutral'

# View goals
def view_goals():
    """View personal growth goals and progress."""
    if not goals:
        messagebox.showinfo("View Goals", "No goals set.")
        return

    goals_text = ""
    for goal in goals:
```

```
        goals_text += f"Date: {goal['date']}\nGoal: {goal['goal']}\n{'-' * 40}\n"

    show_text_window("View Goals", goals_text)

# Award badges based on mood history and goals
def award_badges():
    """Award achievement badges based on user activity."""
    if len(entries) >= 7 and "Consistent Journaler" not in badges:
        badges.add("Consistent Journaler")
        messagebox.showinfo("Badge Earned", "You've earned the 'Consistent Journaler' badge for journaling for 7 days!")

    positive_moods = [entry['mood'] for entry in entries if entry['mood'] == 'Positive']
    if len(positive_moods) >= 5 and "Positive Streak" not in badges:
        badges.add("Positive Streak")
        messagebox.showinfo("Badge Earned", "You've earned the 'Positive Streak' badge for having 5 positive mood entries!")

# Display a random mood-boosting tip
def mood_boosting_tip():
    """Display a random mood-boosting tip."""
    messagebox.showinfo("Mood-Boosting Tip", random.choice(mood_boosting_tips))

# Daily reminders to journal
def daily_reminder():
    """Remind user to journal daily."""
```

```
    last_entry_date = datetime.datetime.strptime(entries[-1]['date'], "%Y-%m-%d %H:%M:%S") if entries else None
    today_date = datetime.datetime.now().strftime("%Y-%m-%d")

    if last_entry_date and last_entry_date.strftime("%Y-%m-%d") != today_date:
        messagebox.showinfo("Daily Reminder", "Reminder: You haven't added a diary entry today. Don't forget to journal!")

# Mood graphs (text-based visualization)
def mood_graph():
    """Display a simple text-based graph of mood trends."""
    if not entries:
        messagebox.showinfo("Mood Graph", "No mood data to display.")
        return
    mood_counts = {'Positive': 0, 'Neutral': 0, 'Negative': 0}
    for entry in entries:
        mood_counts[entry['mood']] +=
    graph_text = "Mood Graph:\n"
    for mood, count in mood_counts.items():
        graph_text += f"{mood}: {'*' * count}\n"
    show_text_window("Mood Graph", graph_text)

# Show text in a new window
def show_text_window(title, text):
    """Show text in a new window."""
    text_window = tk.Toplevel(root)
```

```
    text_window.title(title)
    text_area = Text(text_window, wrap=tk.WORD)
    text_area.insert(tk.END, text)
    text_area.pack(expand=True, fill='both')

# Main function to interact with the diary
def main():
    """Main function to interact with the personal diary."""
    global root
    root = tk.Tk()
    root.title("Personal Diary")

    frame = tk.Frame(root)
    frame.pack(padx=20, pady=20
    tk.Button(frame, text="Add Entry", width=25, command=lambda:
add_entry(simpledialog.askstring("Add Entry", "Enter diary entry:"))).pack(pady=5)
    tk.Button(frame, text="Analyze Entries", width=25, command=analyze_entries).pack(pady=5)
    tk.Button(frame, text="Add Goal", width=25, command=lambda:
add_goal(simpledialog.askstring("Add Goal", "Enter personal growth goal:"))).pack(pady=5)
    tk.Button(frame, text="View Goals", width=25, command=view_goals).pack(pady=5)
    tk.Button(frame, text="View Achievement Badges", width=25, command=lambda:
messagebox.showinfo("Achievement Badges", ", ".join(badges) if badges else "No badges earned
yet.")).pack(pady=5)
    tk.Button(frame, text="Get Mood-Boosting Tip", width=25,
command=mood_boosting_tip).pack(pady=5)
    tk.Button(frame, text="View Mood Graph", width=25, command=mood_graph).pack(pady=5)
    tk.Button(frame, text="Exit", width=25, command=root.quit).pack(pady=5)
```

```
    daily_reminder()

    root.mainloop()

if __name__ == "__main__":

    main()
```

6. Conclusion:

The **Mood Tracker and Personal Growth Assistant** project was successfully completed thanks to the hard work and dedication of the team. Each member played a critical role in bringing the project to fruition, and the final product is a testament to their collaborative effort. The application not only meets the initial project objectives but also has the potential to make a positive impact on users' lives by helping them track and improve their emotional well-being.