VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

21CSMP67

MINI PROJECT REPORT

on

HarmonyHub: Mental health assistant

Submitted in partial fulfillment of the requirements for 6th Semester in
Bachelor of Engineering in Computer Science and Engineering
of Visvesvaraya Technological University, Belagavi

Submitted by

AMITH SAGAR L 1RN21CS019

CHIRAG N SUNDAR 1RN21CS047

Under the Guidance of:

Dr. Sampada K S

Associate Professor

Dept. of CSE, RNSIT

Department of Computer Science and Engineering

RNS Institute of Technology

(Accredited by NBA upto 30-06-2025)

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

2023-2024

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(Accredited by NBA upto 30-06-2025)

CERTIFICATE

Certified that the Mini Project work has been successfully carried out by **AMITH SAGAR** and **CHIRAG N SUNDAR** bearing USN's **1RN21CS019** and **1RN21CS047** ,bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements of 6th Semester in **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work for the said degree.

Dr. Sampada K S	**Dr. Kiran P**	**Dr. Ramesh Babu H S**
Associate Professor	Professor and HOD	Principal
Dept. of CSE	Dept. of CSE	RNSIT
RNSIT	RNSIT	

Abstract

HarmonyHub: Mental Health Assistant is an innovative application designed to support and enhance users' mental well-being through a comprehensive suite of tools and resources. This project integrates advanced technologies such as natural language processing and generative AI to create a supportive environment for users to manage their mental health effectively. The platform provides features like mood tracking, goal setting, and a community forum, allowing users to gain insights into their mental health, set achievable goals, and connect with others facing similar challenges.

The HarmonyHub app is built with a user-friendly interface that includes a cheerful and supportive chatbot, designed to provide positive affirmations and mental health advice based on the user's inputs and uploaded mental health resources. The platform also offers a habit tracker to help users maintain healthy routines and track their progress over time. By leveraging AI and machine learning, HarmonyHub ensures personalized and adaptive support, making it a valuable tool for anyone looking to improve their mental health and well-being.

Acknowledgement

At the very onset, I would like to place on record my gratitude to all those people who have helped me in making this Mini Project work a reality. Our Institution has played a paramount role in guiding in the right direction.

I would like to profoundly thank **Sri. Satish R Shetty**, Managing Director, RNS Group of Companies, Bengaluru for providing such a healthy environment for the successful completion of this Mini Project work.

I would like to thank our beloved Director, **Dr. M K Venkatesha**, for his constant encouragement which motivated me to complete this work.

I would also like to thank our beloved Principal, **Dr. Ramesh Babu H S** , for providing the necessary facilities to carry out this work.

I am extremely grateful to **Dr. Kiran P**, Professor and Head, Department of CSE for his constant encouragement and motivation which helped me to accomplish this work.

I would like to express my sincere thanks to our Mini Project Coordinator, **Mrs. Mamatha Jajur S**, Assistant Professor, Department of CSE, RNSIT.

My heartfelt thanks to my guide **Dr. Sampada K S** , Associate Professor, Department of CSE, for her continuous guidance and constructive suggestions for this work.

I am thankful to all the teaching and non-teaching staff members of the Computer Science and Engineering Department for their encouragement and support throughout this work.

Amith Sagar L

1RN21CS019

Chirag N Sundar

1RN21CS047

EVALUATION SHEET

SL. NO	PARTICULARS	MAX MARKS	MARKS AWARDED
1	**PHASE 1** Title Scrutiny	10	
2	**PHASE 2** Implementation (50% Completion)	20	
3	**PHASE 3** Complete Implementation	50	
4	**REPORT SUBMISSION**	20	
	TOTAL	100	

Signature of Guide

Contents

Chapter 1

Introduction

Mental health is a critical component of overall well-being, yet it often remains under-addressed due to stigma, lack of resources, and limited accessibility to support systems. In response to these challenges, HarmonyHub has been developed as an innovative mental health assistant, aiming to bridge the gap between individuals and the mental health support they need. Unlike traditional mental health tools, HarmonyHub offers a comprehensive suite of features designed to empower users to take control of their mental health journey.

HarmonyHub combines mood tracking, goal setting, access to mental health resources, and crisis intervention into a single, user-friendly platform. By leveraging the power of technology, HarmonyHub seeks to make mental health support more accessible, personalized, and effective. This project highlights the importance of integrating mental health support into everyday life, providing users with the tools and insights necessary to foster a proactive and informed approach to mental health care.

The development of HarmonyHub was driven by the understanding that mental health is not a one-size-fits-all issue. Therefore, the platform is designed to be adaptable to the diverse needs of its users. Whether someone is looking to track their mood, set personal goals, access reliable mental health information, or seek immediate help during a crisis, HarmonyHub is equipped to provide comprehensive support. This introduction outlines the vision behind HarmonyHub, its core functionalities, and its potential to revolutionize the way individuals approach their mental health and well-being.

1.1 Background

1.2 Problem Statement

Despite growing awareness of the importance of mental health, many individuals still face significant barriers to accessing effective mental health support. These barriers include stigma, lack of resources, and the inability to find personalized and immediate assistance. Traditional mental health services often fall short in providing the continuous, easily accessible, and tailored support that individuals need to manage their mental well-being proactively.

Moreover, there is a critical gap in integrating comprehensive mental health tools into a single, user-friendly platform that caters to diverse needs. Current solutions may offer mood tracking, goal setting, access to resources, or crisis intervention independently, but rarely do they combine all these features effectively. This fragmentation makes it challenging for users to maintain consistent mental health management and can lead to missed opportunities for timely intervention and support.

The challenge, therefore, is to develop an inclusive and holistic mental health assistant that consolidates these essential services into a cohesive, accessible, and intuitive platform. The goal is to empower users to take control of their mental health journey by providing them with the tools and insights needed to manage their well-being proactively and effectively. This project aims to address these issues by creating HarmonyHub, an innovative solution designed to overcome the limitations of existing mental health support systems.

1.3 Objectives

- To develop a comprehensive mental health assistant that integrates mood tracking, goal setting, access to resources, and crisis intervention into a single platform.
- To create an intuitive and user-friendly interface that allows individuals to easily navigate and utilize various mental health support tools.
- To provide personalized insights and recommendations based on user data to enhance the effectiveness of mental health management.

- To ensure continuous accessibility to mental health support, enabling users to manage their well-being proactively and consistently.
- To reduce the stigma associated with seeking mental health support by providing a private and secure platform for users.
- To leverage advanced technologies, such as machine learning and data analytics, to offer tailored mental health solutions.
- To integrate real-time crisis intervention features to offer immediate assistance during critical moments.
- To evaluate the effectiveness of HarmonyHub through user feedback and continuous improvement based on the insights gathered.

1.4 Scope

The scope of this project, *HarmonyHub: A Mental Health Assistant*, encompasses several key aspects aimed at addressing the mental health needs of users through a comprehensive digital platform. The scope is defined as follows:

- **Feature Development:**
  - *Mood Tracking:* Implementing tools to allow users to record and monitor their daily mood and emotional states.
  - *Goal Setting:* Providing functionalities for users to set, track, and achieve personal mental health goals.
  - *Resource Access:* Creating a repository of mental health resources, including articles, videos, and guided exercises, accessible to users.
  - *Crisis Intervention:* Developing real-time support features for users in crisis, including immediate access to helplines and emergency contacts.
- **User Interface and Experience:**

  - Designing an intuitive and user-friendly interface to ensure ease of navigation and accessibility.
  - Incorporating personalized dashboards that reflect individual user data and progress.
  - Ensuring the platform is accessible across multiple devices, including smartphones, tablets, and computers.

- **Personalization and Insights:**
  - Utilizing machine learning algorithms to analyze user data and provide personalized recommendations.
  - Offering tailored insights based on mood tracking and goal-setting activities to enhance mental health management.

- **Privacy and Security:**
  - Implementing robust data security measures to protect user information.
  - Ensuring compliance with relevant data privacy regulations and standards.
  - Providing users with control over their data, including options for data export and deletion.

- **Continuous Improvement:**
  - Collecting user feedback to refine and enhance platform features.
  - Conducting regular updates to integrate new mental health resources and tools.
  - Evaluating the platform's effectiveness through user surveys and usage analytics.

- **Stakeholder Engagement:**
  - Engaging with mental health professionals to ensure the accuracy and relevance of provided resources.
  - Collaborating with users to understand their needs and preferences better.

By addressing these aspects, the project aims to create a comprehensive, user-friendly, and secure mental health assistant that effectively supports users in managing their mental well-being.

Chapter 2

Methodology

The development of *HarmonyHub: A Mental Health Assistant* follows a structured methodology that ensures the creation of a comprehensive and effective mental health support platform. The methodology is divided into several key phases:

- **Phase 1: Requirements Analysis**
  - *Stakeholder Interviews:* Conduct interviews with potential users, mental health professionals, and stakeholders to gather requirements and expectations.
  - *Surveys and Questionnaires:* Distribute surveys to collect data on user needs, preferences, and existing challenges related to mental health management.
  - *Market Research:* Analyze existing mental health applications and services to identify gaps and opportunities for differentiation.
- **Phase 2: System Design**
  - *Feature Specification:* Define the features and functionalities based on the requirements analysis, including mood tracking, goal setting, resource access, and crisis intervention.
  - *User Interface Design:* Create wireframes and prototypes of the user interface to ensure an intuitive and user-friendly experience.
  - *Architecture Design:* Develop the system architecture, including backend services, database schema, and integration points for external APIs and services.
- **Phase 3: Implementation**

- *Frontend Development:* Implement the user interface components using appropriate web technologies (e.g., HTML, CSS, JavaScript) and frameworks (e.g., Streamlit).
- *Backend Development:* Develop backend services and APIs to handle data processing, user authentication, and communication with external systems.
- *Integration of Machine Learning Models:* Incorporate machine learning algorithms for personalized insights and recommendations using libraries such as Sentence Transformers and Generative AI.

- **Phase 4: Testing**
  - *Unit Testing:* Perform unit tests on individual components and functions to ensure they work as expected.
  - *Integration Testing:* Test the integration of various system components and external services to ensure seamless functionality.
  - *User Acceptance Testing (UAT):* Conduct testing with a group of end-users to validate the system's usability, functionality, and overall user experience.

- **Phase 5: Deployment**
  - *Deployment Planning:* Prepare deployment scripts and procedures for rolling out the application to production environments.
  - *Launch:* Deploy the application to the live environment and ensure all components are functioning correctly.
  - *Post-Deployment Monitoring:* Monitor the system for any issues or performance bottlenecks and address them promptly.

- **Phase 6: Maintenance and Improvement**
  - *User Feedback Collection:* Continuously gather feedback from users to identify areas for improvement and additional features.
  - *Regular Updates:* Implement updates and enhancements based on user feedback and emerging trends in mental health support.

– *Performance Monitoring:* Regularly assess system performance and make necessary adjustments to ensure optimal operation.

This methodology ensures that *HarmonyHub* is developed systematically, with a focus on meeting user needs and providing effective mental health support.

2.1 Software Requirements

Table 2.1: Software Requirements for *HarmonyHub: A Mental Health Assistant*

Component	Description
Operating System	Windows 10/11, macOS 10.15 (Catalina) or later, Ubuntu 20.04 LTS or later
Programming Languages	Python (for backend development and machine learning), Python (Streamlit) (for frontend development)
Frontend Framework	React.js or Vue.js for interactive UI, Streamlit for rapid prototyping
Backend Framework	Flask or Django for RESTful API development
Database	PostgreSQL or MySQL for relational data storage, SQLite for development and testing
Machine Learning Libraries	Sentence Transformers for NLP tasks, TensorFlow or PyTorch for model training
Version Control	Git for version control, GitHub or GitLab for repository hosting
Deployment Platform	Heroku, AWS, or Azure for deploying the application
Integrated Development Environment (IDE)	VS Code, PyCharm, or any other suitable IDE for development
API Integration	External APIs for additional functionalities (e.g., mental health resources)
Testing Frameworks	PyTest or unittest for backend testing, Jest or Mocha for frontend testing

2.2 Algorithms

2.2.1 Leaky Bucket Algorithm

The Leaky Bucket algorithm is used for traffic shaping and rate limiting. It can be described by the following steps:

1. **Initialization:** Initialize the bucket with a maximum capacity B and a leak rate R. The bucket starts empty.

2. **Arrival of Packets:** When a packet of size p arrives, check if there is enough space in the bucket.

- **If the bucket has enough space:** Add the packet to the bucket.
- **If the bucket does not have enough space:** Discard the packet.

3. **Leakage:** Continuously leak packets from the bucket at a constant rate R.
4. **Output:** The packets that are leaked are sent to the output.

Mathematically, the amount of water (or packets) in the bucket at time t can be expressed as:

$$W(t) = \max(0, W(t-1) + A(t) - R \cdot \Delta t)$$

where:

- $W(t)$ is the amount of water in the bucket at time t,
- $A(t)$ is the amount of water added to the bucket at time t,
- R is the leak rate,
- Δt is the time interval.

2.2.2 Chat Interface with Streamlit

The chat interface is implemented using Streamlit and consists of the following components:

1. **Login UI:** The login interface uses the `streamlit_login_auth_ui` widget to authenticate users. If authenticated, the app displays the main content.
2. **Sidebar Navigation:** Users can navigate between different sections of the app, such as Home, Resources, Assistant, and Habit Tracker, using a sidebar menu.
3. **Chat History:** The chat history is maintained in `st.session_state['messages']`. The chat messages are displayed and processed using the Streamlit chat components.
4. **PDF Processing:** PDFs are uploaded and processed to extract text. The extracted text is used to provide contextual responses in the chat.
5. **Response Generation:** Responses are generated using the Gemini model from `google.generativeai`. The context is derived from the most similar document based on cosine similarity.

2.2.3 Habit Tracker Algorithm

The Habit Tracker component uses the following steps:

1. **Data Loading:** Users can load or create a new CSV file to store habit data.
2. **Data Entry:** Users can enter data for various habits (e.g., sleep, mood, exercise) and add it to the loaded dataset.
3. **Data Removal:** Users can remove entries based on the date.
4. **Data Visualization:** The app provides visualizations of habit data over time using Plotly charts.

2.2.4 Mathematical Operations for Habit Tracker

The average performance metric is calculated as follows:

$$\text{avg\_performance} = \frac{\text{sum of all habits}}{28}$$

where 28 is the number of days in a typical month.

2.3 Data Collection and Analysis procedure

2.3.1 PDF Data Collection

File Upload

Users can upload PDF files through the sidebar interface of the app. This functionality allows for the integration of mental health resources directly into the application. The file upload process involves:

- **Interface:** A file uploader widget provided by Streamlit (`st.file_uploader`) allows users to select multiple PDF files.
- **File Handling:** Uploaded files are temporarily stored in the application's session state for further processing.

Text Extraction

The app uses the `PyPDF2` library to extract text from the uploaded PDF files. The procedure is as follows:

- **PDF Reading:** Each page of the PDF is read, and text is extracted using the `PdfReader` class.
- **Text Aggregation:** Text from all pages of a PDF file is concatenated into a single string for each document.
- **Data Storage:** The aggregated text is stored in the session state for later use in generating context-aware responses.

Code Example

```
def extract_text_from_pdf(file):
    pdf_reader = PyPDF2.PdfReader(file)
    text = ""
    for page in pdf_reader.pages:
        text += page.extract_text()
    return text
```

2.3.2 Habit Tracking Data Collection

Data Entry

Users can input data related to their daily habits through the habit tracker interface. This includes various metrics such as sleep hours, mood level, and more. The data entry process involves:

- **Date Input:** Users specify the date for which they want to record their habits.
- **Habit Metrics:** Users rate or provide information on various aspects of their day, including sleep, mood, energy, and other activities.
- **Submission:** Data is submitted and stored in a CSV file format.

File Handling

Users have the option to either upload an existing CSV file or create a new one. The process includes:

- **File Upload:** An existing CSV file can be loaded into the app, allowing users to continue from where they left off.
- **File Creation:** Users can specify a filename for a new CSV file which will be created and initialized for data entry.

Code Example

```
def get_file(input_file):
    HabitData = data.HabitData()
    HabitData.load(file=input_file)
    return HabitData


def create_file(filename):
    HabitData = data.HabitData()
    HabitData.create(filename=filename)
    return HabitData
```

2.3.3 Data Analysis

Text Similarity Analysis

Embedding Generation Text from the PDFs and user queries are converted into vector embeddings using the `SentenceTransformer` model. This process involves:

- **Model Initialization:** The `SentenceTransformer` model is loaded with a pre-trained model, such as `'all-MiniLM-L6-v2'`.
- **Embedding Computation:** The text data and user queries are transformed into embeddings (vectors) that capture semantic meaning.

Similarity Calculation Cosine similarity is used to measure the similarity between the user query embedding and the embeddings of the uploaded documents. The procedure is:

- **Cosine Similarity Formula:**

$$\text{cosine\_similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||||\mathbf{v}||}$$

  where **u** and **v** are the vector embeddings of the user query and document, respectively.

- **Context Selection:** The document with the highest similarity score is chosen to provide context for generating a response.

Code Example

```
def get_embeddings(documents):
    return sentence_model.encode(documents)
```

Habit Data Analysis

Data Aggregation The habit data collected is aggregated to compute average performance metrics. This involves:

- **Summation:** Summing up the values for each habit across all recorded entries.
- **Average Calculation:** Dividing the summed values by the number of days to get an average score.

Visualization Data visualization is performed using Plotly to display trends and patterns in habit data. This involves:

- **Plot Generation:** Creating bar charts or other visualizations to show changes in habit metrics over time.
- **Interactive Features:** Users can select different metrics to visualize and analyze their habits dynamically.

Code Example

```
px_chart = px.bar(HabitData.data, x="date", y=selectbox_columns)
st.plotly_chart(px_chart, use_container_width=True)
```

The data collection and analysis procedures implemented in the Mental Health App are designed to provide a comprehensive understanding of user interactions and habits. By integrating PDF processing, habit tracking, and text similarity analysis, the app supports users in managing their mental health effectively and provides meaningful insights based on their inputs.

Chapter 3

Implementation

Code Example

left=1in, right=1in, top=1in, bottom=1in

Listing 3.1: Streamlit Code Example

```
import streamlit as st
from streamlit_login_auth_ui.widgets import __login__
import os
import json
import PyPDF2
from streamlit_chat import message
from dotenv import load_dotenv
import google.generativeai as genai
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
import plotly.express as px
from modules import data
from PIL import Image
from io import BytesIO
import time
from sentence_transformers import SentenceTransformer
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
```

```
import random

# Login UI setup
__login__obj = __login__(auth_token="AUTH_TOKEN",
                         company_name="HarmonyHub:␣Mental␣health␣assistant",
                         width=200, height=250,
                         logout_button_name='Logout', hide_menu_bool=False,
                         hide_footer_bool=True,
                         lottie_url='https://assets2.lottiefiles.com/packages/lf20_jcikwtu

LOGGED_IN = __login__obj.build_login_ui()

if LOGGED_IN:
    # Define the CSS for the background color
    page_bg_css = """
    <style>
        .stApp {
            background-color: #000000;
            color :#00ff00;
        }
    </style>
    """
    st.markdown(page_bg_css, unsafe_allow_html=True)

    # Initialize chat open state
    if 'chat_open' not in st.session_state:
        st.session_state['chat_open'] = False

    # Function to toggle chat panel
    def toggle_chat():
        st.session_state['chat_open'] = not st.session_state['chat_open']

    # Sidebar navigation
    st.sidebar.title("Mental␣Health␣App")
    nav = st.sidebar.selectbox("Navigation", ["Home", "Resources", "Assistant", "␣Habit␣Tr
```

```
# Main container
main_container = st.container()

# Containers for different sections
header_container = main_container.container()
hero_container = main_container.container()
features_container = main_container.container()
call_to_action_container = main_container.container()
footer_container = main_container.container()

# Content for Home navigation
if nav == "Home":
    def animated_section(content_func, alignment):
        container = st.empty()
        with container.container():
            if alignment == "left":
                st.markdown('<div class="left-section">', unsafe_allow_html=True)
                content_func()
                st.markdown('</div>', unsafe_allow_html=True)
            else:
                st.markdown('<div class="right-section">', unsafe_allow_html=True)
                content_func()
                st.markdown('</div>', unsafe_allow_html=True)
        return container

    def home_content():
        st.title("Welcome to Our Mental Health App")
        st.write("Your journey to better mental health starts here.")
        st.header("Take Control of Your Mental Health")
        st.write("""
        Our app is designed to help you manage your mental health and wellbeing. With
        you'll be able to track your progress, set goals, and connect with others who
        Whether you're looking to improve your mood, set and achieve goals, or find co
        for you.
        """)
        st.write("""
```

```
    Mental health is a crucial part of overall well-being, and taking steps to man
    quality of life. Our app offers features like mood tracking, goal setting, and
    By regularly engaging with these tools, you can gain insights into your mental
    decisions.
    """)
    st.write("""
    We understand that everyone's mental health journey is unique, and our app is
    to different needs. Whether you're looking for daily journaling, setting speci
    app is here to help. Start today and take the first step towards better mental
    and resources.
    """)

def mood_tracking_content():
    st.title("Mood Tracking")
    st.write("""
    Our Mood Tracking feature is designed to help you understand and manage your e
    interface, you can easily log your daily moods, thoughts, and experiences. The
    mood patterns over time, allowing you to identify trends and triggers.
    """)
    st.write("""
    By regularly tracking your mood, you'll gain valuable insights into your emoti
    make informed decisions about your self-care and when to seek additional suppo
    in mood due to stress, life changes, or other factors, mood tracking can provi
    """)
    st.write("""
    In addition to tracking your mood, the app offers personalized insights based
    you in understanding the underlying causes of your emotions and help you devel
    Start using the Mood Tracking feature today and take a proactive step towards
    """)

def goal_setting_content():
    st.title("Goal Setting")
    st.write("""
    The Goal Setting feature empowers you to take proactive steps towards improvin
    (Specific, Measurable, Achievable, Relevant, Time-bound) framework to help you
    it's practicing mindfulness, improving sleep habits, or reducing stress, our a
```

```
    """)
    st.write("""
    You can track your progress with interactive charts and receive timely reminde
    achieving your mental health objectives. Setting and achieving goals can provi
    confidence, contributing positively to your overall well-being.
    """)
    st.write("""
    The app also allows you to adjust your goals as needed, ensuring they remain r
    and updating your goals, you can maintain a forward momentum in your mental he
    today and take control of your mental health with our structured and supportiv
    """)

def community_forum_content():
    st.title("Community Forum")
    st.write("""
    Our Community Forum is a safe and supportive space where you can connect with
    Join discussion groups based on common interests or experiences, share your st
    is moderated to ensure a respectful and constructive environment.
    """)
    st.write("""
    Additionally, you'll have access to valuable resources and occasional expert a
    These resources can provide guidance and support, helping you navigate your me
    and understanding. Engaging with the community can also reduce feelings of iso
    """)
    st.write("""
    Remember, while peer support can be incredibly helpful, it's not a substitute
    Our forum is designed to complement your mental health care by providing addit
    Community Forum today and start building meaningful connections with others wh
    """)

def cta_content():
    st.title("Get Started Today")
    st.write("""
    Sign up for our app and start your journey to better mental health. Our compre
    to support you every step of the way. Whether you're looking to track your moo
    support, our app has everything you need.
```

```
    """)
    st.write("""
    Taking the first step towards better mental health can be daunting, but you're
    providing the structure and support you need to make meaningful progress. With
    goal setting, and a supportive community forum, you'll have all the tools you
    """)
    st.write("""
    Don't wait any longer to take control of your mental health. Sign up today and
    a healthier, happier you. Start your journey now and see the difference that d
    in your life.
    """)

def footer_content():
    st.write("Copyright 2023 Mental Health App. All rights reserved.")

# Display sections with alternating alignment
animated_section(home_content, alignment="left")
st.markdown('<div class="fade-in"></div>', unsafe_allow_html=True)
time.sleep(0.5)

animated_section(mood_tracking_content, alignment="right")
st.markdown('<div class="fade-in"></div>', unsafe_allow_html=True)
time.sleep(0.5)

animated_section(goal_setting_content, alignment="left")
st.markdown('<div class="fade-in"></div>', unsafe_allow_html=True)
time.sleep(0.5)

animated_section(community_forum_content, alignment="right")
st.markdown('<div class="fade-in"></div>', unsafe_allow_html=True)
time.sleep(0.5)

animated_section(cta_content, alignment="left")
st.markdown('<div class="fade-in"></div>', unsafe_allow_html=True)
time.sleep(0.5)
```

```
    footer_container.write(footer_content())
if nav == "Resources":
    with header_container:
        st.title("Resources␣for␣Mental␣Health")
        st.write("Here␣are␣some␣resources␣that␣may␣be␣helpful␣for␣managing␣mental␣heal
        with hero_container:
            st.header("Mental␣Health␣Organizations")

            st.write("1.␣Understanding␣Mental␣Health")
            st.write("␣␣␣-␣Provides␣free␣24/7␣mental␣health␣counseling␣in␣India")
            st.markdown("[Download](https://www.un.org/en/healthy-workforce/files/Unde

            st.write("2.␣Mental␣Health␣Wellbeing")
            st.write("␣␣␣-␣Offers␣general␣mental␣health␣resources")
            st.markdown("[Download](https://www.education.gov.in/covid-19/assets/img/p

            st.write("3.␣Mental␣Health␣Resources␣")
            st.write("␣␣␣-␣Contains␣all␣the␣required␣mental␣health␣resources.")
            st.markdown("[Download](https://iris.who.int/bitstream/handle/10665/42823/

            st.write("4.␣Mental␣Health␣Tips")
            st.write("␣␣␣-␣Information␣on␣finding␣help,␣providers,␣and␣treatment␣for␣m
            st.markdown("[Download](https://www.mentalhealth.org.uk/sites/default/file

            st.write("5.␣Tips␣for␣improving␣mental␣health")
            st.write("␣␣␣-␣Contains␣all␣the␣tips␣related␣to␣mental␣health")
            st.markdown("[Download](https://www.heretohelp.bc.ca/sites/default/files/i
if nav == "Assistant":
    load_dotenv()
    api_key = os.getenv("GENAI_API_KEY") or st.secrets.get("GENAI_API_KEY")

    if not api_key:
        st.error("Please␣set␣the␣GENAI_API_KEY␣environment␣variable.")
        st.stop()

    genai.configure(api_key=api_key)
```

```
# Set up the model
generation_config = {
    "temperature": 0.9,
    "top_p": 1,
    "top_k": 32,
    "max_output_tokens": 8192,
}

safety_settings = [
    {"category": "HARM_CATEGORY_HARASSMENT", "threshold": "BLOCK_MEDIUM_AND_ABOVE"
    {"category": "HARM_CATEGORY_HATE_SPEECH", "threshold": "BLOCK_MEDIUM_AND_ABOVE
    {"category": "HARM_CATEGORY_SEXUALLY_EXPLICIT", "threshold": "BLOCK_MEDIUM_AND
    {"category": "HARM_CATEGORY_DANGEROUS_CONTENT", "threshold": "BLOCK_MEDIUM_AND
]

model = genai.GenerativeModel(
    model_name="gemini-pro",
    generation_config=generation_config,
    safety_settings=safety_settings,
)

# Initialize Sentence Transformer model
sentence_model = SentenceTransformer('all-MiniLM-L6-v2')

# Function to extract text from PDF
def extract_text_from_pdf(file):
    pdf_reader = PyPDF2.PdfReader(file)
    text = ""
    for page in pdf_reader.pages:
        text += page.extract_text()
    return text

# Function to process uploaded PDFs
def process_pdfs(uploaded_files):
    all_text = []
```

```
    for file in uploaded_files:
        text = extract_text_from_pdf(file)
        all_text.append(text)
    return all_text

# Function to get embeddings for documents
def get_embeddings(documents):
    return sentence_model.encode(documents)

st.title("␣Cheerful␣Mental␣Health␣Buddy␣")

# Sidebar for PDF uploads
with st.sidebar:
    st.header("Upload␣Mental␣Health␣Resources")
    uploaded_files = st.file_uploader("Choose␣PDF␣files", accept_multiple_files=Tr

    if uploaded_files:
        documents = process_pdfs(uploaded_files)
        document_embeddings = get_embeddings(documents)
        st.session_state['documents'] = documents
        st.session_state['document_embeddings'] = document_embeddings
        st.success(f"Yay!␣Processed␣{len(documents)}␣documents.␣")

# Main chat interface
greeting_messages = [
    "Hello␣there!␣I'm␣your␣friendly␣Mental␣Health␣Buddy.␣How␣can␣I␣brighten␣your␣d
    "Welcome!␣I'm␣here␣to␣chat␣and␣support␣you.␣What's␣on␣your␣mind␣today?␣",
    "Hi␣friend!␣It's␣great␣to␣see␣you.␣How␣about␣we␣start␣with␣something␣positive?
    "Greetings!␣I'm␣your␣cheerful␣chat␣companion.␣Ready␣for␣an␣uplifting␣conversat
]

if 'greeted' not in st.session_state:
    st.session_state['greeted'] = False

if not st.session_state['greeted']:
    st.write(random.choice(greeting_messages))
```

```
    st.session_state['greeted'] = True

# Initialize chat history
if 'messages' not in st.session_state:
    st.session_state['messages'] = []

# Display chat history
for message in st.session_state['messages']:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

# Chat input
user_input = st.text_input("You:", key="user_input", placeholder="Type your messag

if user_input:
    # Add user message to chat history
    st.session_state['messages'].append({"role": "user", "content": user_input})
    with st.chat_message("user"):
        st.markdown(user_input)

    with st.spinner("Thinking..."):
        if 'documents' in st.session_state and st.session_state['documents']:
            # Encode query
            query_embedding = sentence_model.encode([user_input])[0]

            # Calculate similarities
            similarities = cosine_similarity([query_embedding], st.session_state['

            # Get the most similar document
            most_similar_idx = np.argmax(similarities)
            most_similar_doc = st.session_state['documents'][most_similar_idx]

            # Generate response using Gemini
            context = f"""You are a cheerful and supportive mental health chatbot
            Use the following context to answer the user's question. If the contex
            use your general knowledge about mental health to provide a helpful an
```

```
                Always maintain a compassionate, professional, and upbeat tone. Start
                positive affirmation or encouragement when appropriate. Remember to us

                Context: {most_similar_doc[:2000]}"""

                response = model.generate_content(
                    contents=[
                        {"role": "user", "parts": [{"text": context}]},
                        {"role": "user", "parts": [{"text": user_input}]}
                    ]
                )
                bot_response = response.text
            else:
                bot_response = "I'm excited to chat with you! However, I don't have an

        # Add assistant response to chat history
        st.session_state['messages'].append({"role": "assistant", "content": bot_respo
        with st.chat_message("assistant"):
            st.markdown(bot_response)

    # Clear chat history button
    if st.button("Start Fresh "):
        st.session_state['messages'] = []
        st.session_state['greeted'] = False
        st.success("Chat history cleared. Let's start anew with positivity! ")

    # Motivational message
    st.markdown("---")
    motivational_messages = [
        "Remember, every step forward is progress. You're doing great! ",
        "Your mental health matters. Be kind to yourself today. ",
        "You have the strength within you to overcome any challenge. Believe in yourse
        "Taking care of your mind is a beautiful act of self-love. Keep it up! "
    ]
    st.markdown(f"*{random.choice(motivational_messages)}*")
if nav == " Habit Tracker":
```

```
st.title("␣Habit␣Tracker")

dt_now = dt.datetime.now()
dt_str = dt_now.strftime("%Y-%m-%d")
dt_weekday = dt_now.strftime("%A")
dt_day = dt_now.strftime("%d")
dt_month = dt_now.strftime("%b")

st.markdown(f"Today␣is␣{dt_weekday}␣-␣{dt_day}.␣of␣{dt_month}.")

# Sidebar
st.sidebar.title("Mission␣Control")

sidebar_data_container = st.sidebar.expander("␣Load␣or␣Create␣a␣file", expande

with sidebar_data_container:
    st.markdown("###␣␣Load␣data")
    sidebar_uploaded_file = st.file_uploader("Choose␣your␣.csv␣file")

    st.markdown("###␣   ␣or␣create␣a␣new␣file")
    sidebar_create_file_name = st.text_input("Choose␣a␣file␣name", value="habi
    sidebar_create_file_button = st.button("*␣Create␣new␣file")

sidebar_input_container = st.sidebar.expander("␣How␣did␣you␣do␣today?", expand

with sidebar_input_container:
    sidebar_date = st.date_input("␣Which␣day␣you␣want␣to␣make␣an␣entry␣for?",
    sidebar_sleep = st.slider("␣How␣much␣did␣you␣sleep?", min_value=0.0, max_v
    sidebar_mood = st.slider("␣What␣mood␣were␣you␣in?", min_value=1, max_value
    sidebar_energy = st.slider("      ␣How␣energized␣did␣you␣feel?", min_value
    sidebar_food = st.radio("␣Did␣you␣eat␣healthy?", (0, 1))
    sidebar_exercise = st.radio("␣Did␣you␣exercise?", (0, 1))
    sidebar_meditation = st.radio("Did␣you␣meditate?", (0, 1))
    sidebar_reading = st.radio("␣Did␣you␣read?", (0, 1))
    sidebar_journaling = st.radio("      ␣Did␣you␣journal?", (0, 1))
    sidebar_learning = st.radio("␣Did␣you␣learn␣something␣new?", (0, 1))
```

```
    sidebar_work = st.radio("␣Did␣you␣work␣towards␣your␣goals?", (0, 1))
    add_row = st.button("   ␣Add␣values")

@st.cache_resource
def get_file(input_file):
    HabitData = data.HabitData()
    HabitData.load(file=input_file)
    return HabitData

@st.cache_resource
def create_file(filename):
    HabitData = data.HabitData()
    HabitData.create(filename=filename)
    return HabitData

if sidebar_uploaded_file is not None:
    st.markdown("Data␣loaded.")
    HabitData = get_file(sidebar_uploaded_file)
elif sidebar_create_file_button:
    st.markdown("   ␣File␣created.")
    HabitData = create_file(sidebar_create_file_name)
else:
    st.markdown("###␣Create␣or␣load␣a␣file␣to␣continue.")
    st.stop()

if add_row:
    if sidebar_date in HabitData.data["date"].values:
        st.warning("Data␣for␣this␣date␣already␣exists.")
    else:
        append_dict = {
            "date": [sidebar_date],
            "sleep": [sidebar_sleep],
            "mood": [sidebar_mood],
            "energy": [sidebar_energy],
            "food": [sidebar_food],
            "exercise": [sidebar_exercise],
```

```
            "meditation": [sidebar_meditation],
            "reading": [sidebar_reading],
            "journaling": [sidebar_journaling],
            "learning": [sidebar_learning],
            "work": [sidebar_work],
        }
        append_df = pd.DataFrame(append_dict)
        HabitData.data = pd.concat([HabitData.data, append_df], ignore_index=T

sidebar_remove_data_container = st.sidebar.expander("Delete␣data", expanded=Fa

with sidebar_remove_data_container:
    try:
        opt = HabitData.data.date.unique()
    except AttributeError:
        opt = [None]

    selectbox_remove_date = st.selectbox("Remove␣data:", options=opt)
    drop_row = st.button("   ␣Remove␣values")

if drop_row:
    HabitData.drop(date_index=selectbox_remove_date)

sidebar_download_data = st.sidebar.button("      ␣Download␣data")

if sidebar_download_data:
    st.sidebar.markdown(HabitData.download(), unsafe_allow_html=True)

HabitData.data["avg_performance"] = HabitData.data.set_index("date").sum(axis=

data_container = st.expander("Display␣your␣data", expanded=True)
with data_container:
    dataframe = st.dataframe(HabitData.data)

st.markdown("###␣Plot␣your␣habits␣over␣time")
```

```
plot_container, info_container = st.columns([8, 2])

with info_container:
    opt = HabitData.data.columns.to_list()
    opt.remove("date")
    selectbox_columns = st.selectbox("Select␣which␣column␣to␣plot:", options=o

    m = round(HabitData.data[selectbox_columns].mean(), 2)
    metric_dict = {
        "mood": "lvl",
        "energy": "lvl",
        "sleep": "h",
        "food": "%",
        "exercise": "%",
        "meditation": "%",
        "reading": "%",
        "journaling": "%",
        "learning": "%",
        "work": "%",
    }
    st.markdown(f"""# avg // {m}""")

with plot_container:
    px_chart = px.bar(HabitData.data, x="date", y=selectbox_columns)
    st.plotly_chart(px_chart, use_container_width=True)
```

Chapter 4

Results

4.1 Login Page

The login page is the entry point for users to access the app. It includes fields for username and password, and options for creating an account, resetting the password, and retrieving a forgotten password.

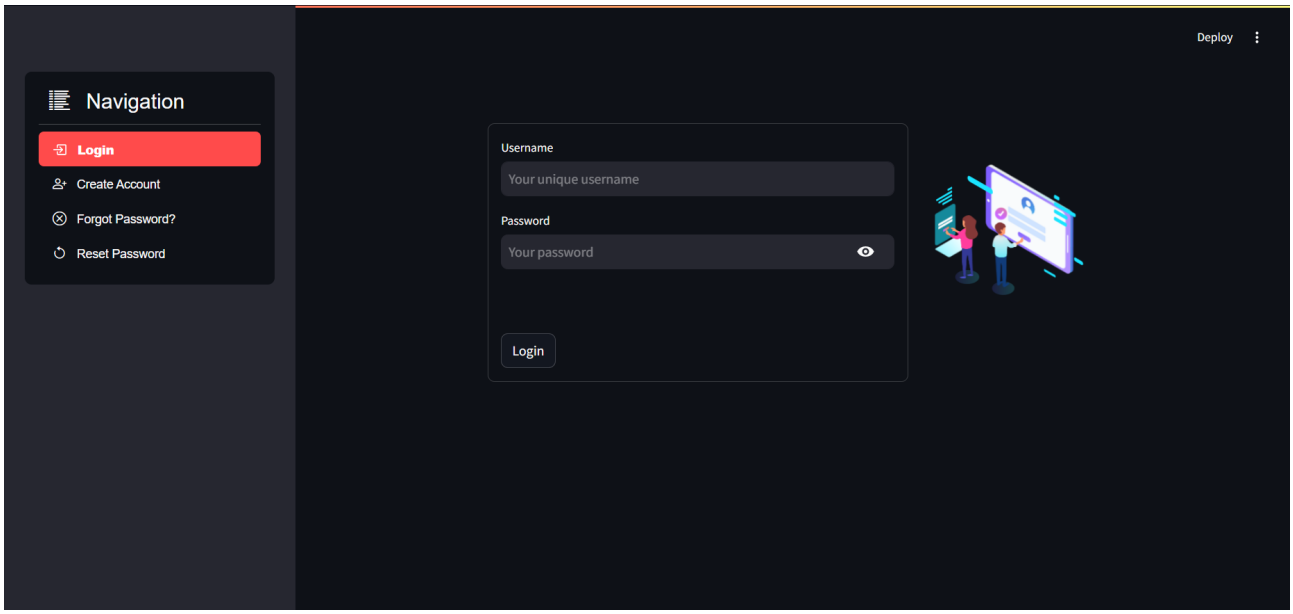

Figure 4.1: Login Page

4.2 Home Page

After logging in, users are greeted with a welcome message and an introduction to the app's purpose and features. The sidebar includes navigation options and a logout button.

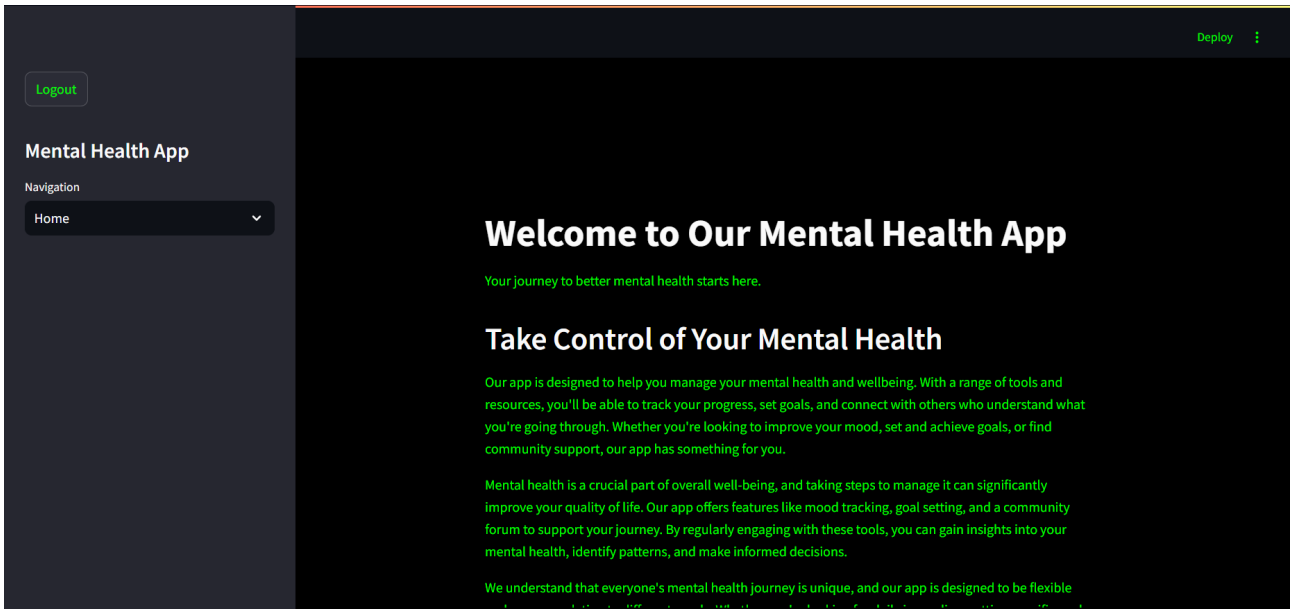

Figure 4.2: Home Page 1

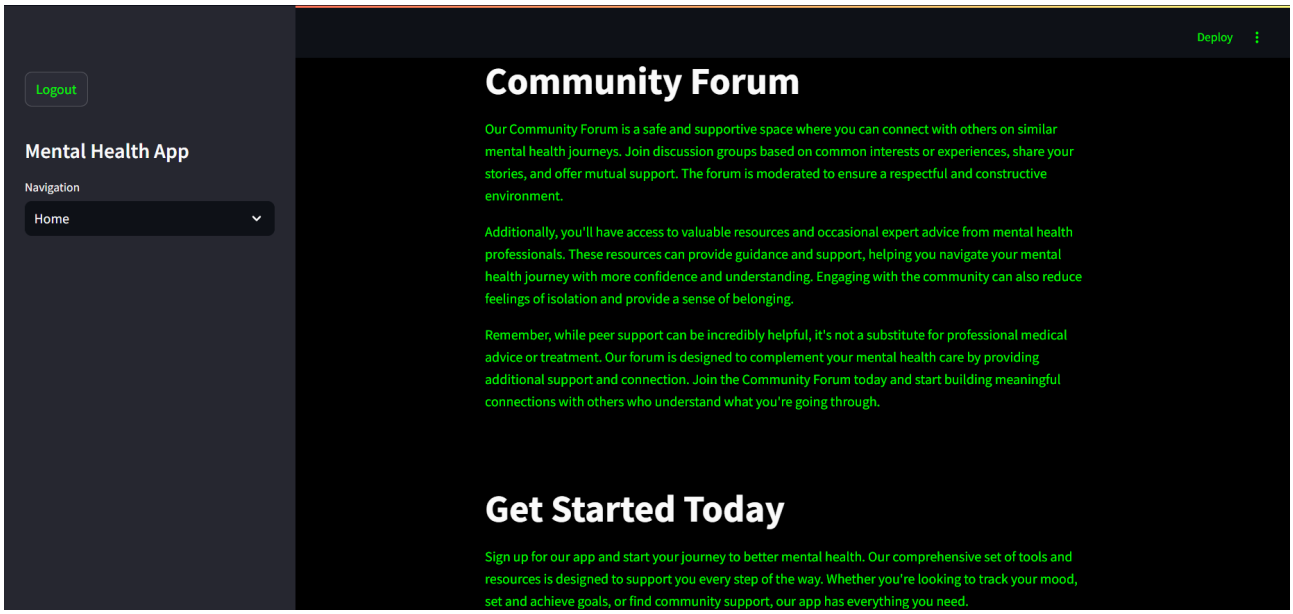

Figure 4.3: Home Page 2

4.3 Resources Page

The resources page contains PDFs and other materials related to mental health. Users can access and download these resources to support their mental health journey.

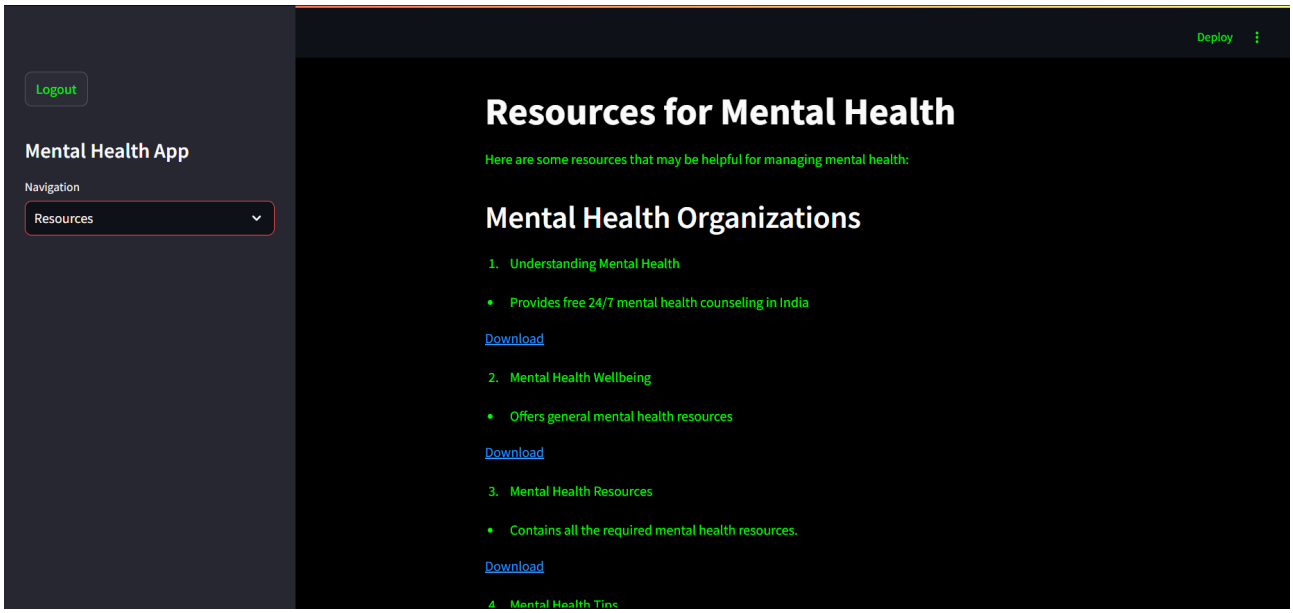

Figure 4.4: Resources Page

4.4 Assistant Page

The assistant page features a mental health chatbot that provides support and answers questions related to mental health. Users can interact with the chatbot to get personalized advice and support.

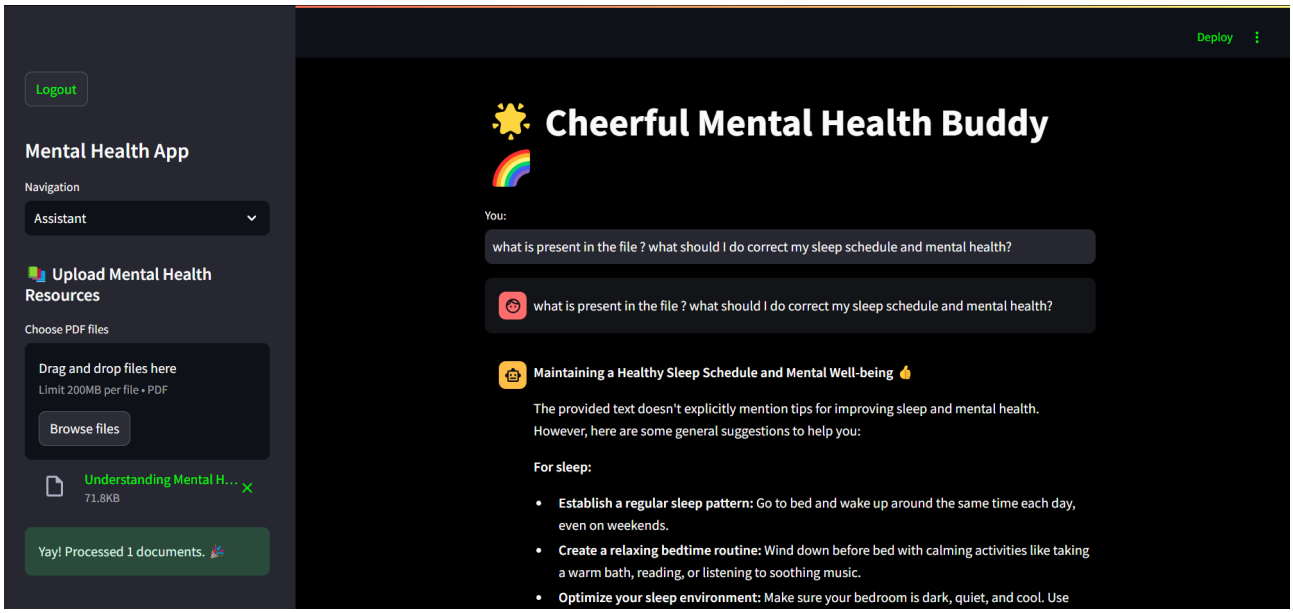

Figure 4.5: Assistant Page

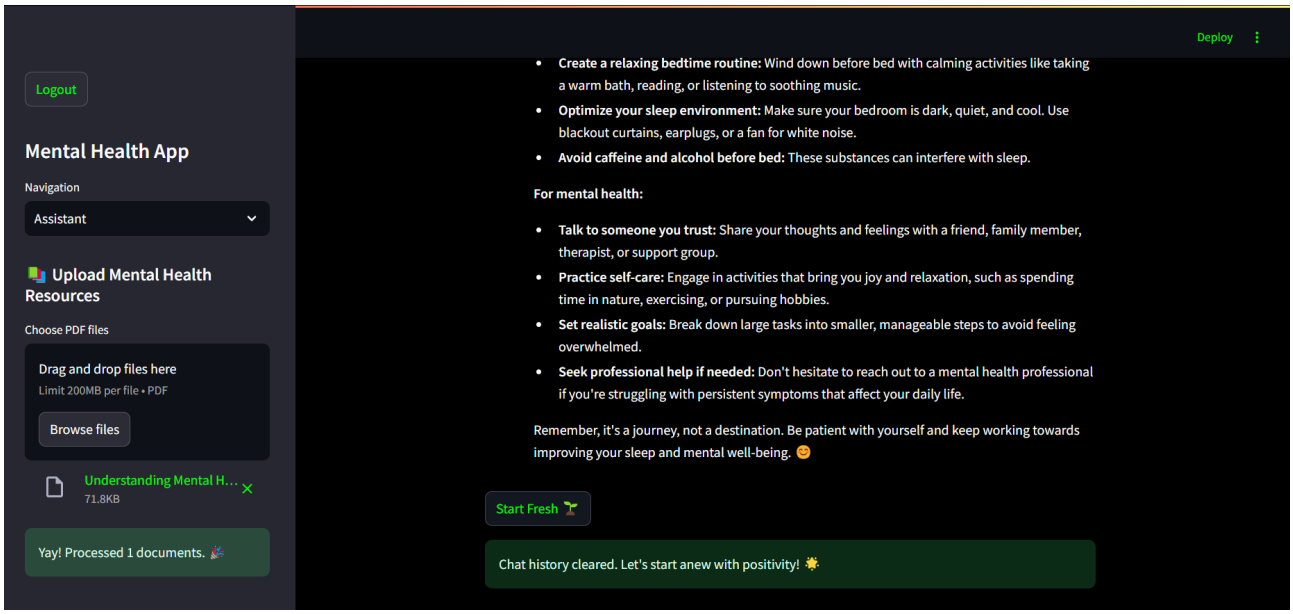

Figure 4.6: Assistant Page

4.5 Habit Tracker

The habit tracker helps users monitor and manage their habits. It provides tools to track progress, set goals, and analyze habits over time.

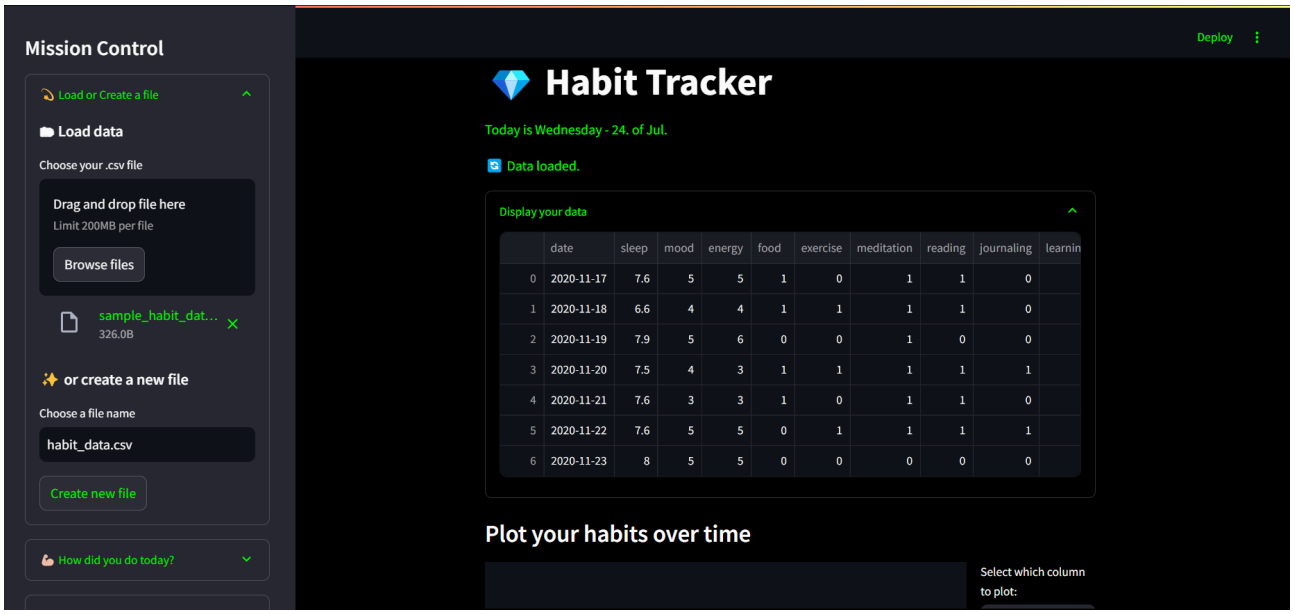

Figure 4.7: Habit Tracker

Chapter 5

Conclusion

The implementation of the mental health application demonstrates a comprehensive and user-centric approach to supporting mental well-being. By integrating a variety of features such as mood tracking, goal setting, community forums, and habit tracking, the application provides a multi-faceted platform for users to manage their mental health effectively. The use of interactive elements and personalized content enhances user engagement and supports diverse needs, from tracking daily habits to connecting with a supportive community.

The application leverages advanced technologies like natural language processing and machine learning to offer personalized assistance and insights. The integration of a generative AI model for chat interactions, combined with the ability to process and analyze user-uploaded documents, adds a layer of responsiveness and adaptability to the platform. This innovative approach not only improves the user experience but also aligns with best practices in mental health support.

Overall, the application stands out as a valuable tool for individuals seeking to enhance their mental health through structured support and personalized interaction. Its well-rounded features and thoughtful design reflect a commitment to addressing various aspects of mental well-being, making it a significant contribution to the field of mental health technology. Future enhancements could further refine its capabilities, ensuring even greater alignment with users' evolving needs and preferences.

References

- National Institute of Mental Health. (n.d.). Retrieved from https://www.nimh.nih.gov.
- Streamlit Documentation. (n.d.). Retrieved from https://docs.streamlit.io.
- World Health Organization. (2017). *Mental Health Atlas 2017*. Retrieved from https://www.who.int/mental_health/evidence/atlas/mental_health_atlas_2017/en/.
- PyPDF2 Documentation. (n.d.). Retrieved from https://pypdf2.readthedocs.io/.
- Plotly Documentation. (n.d.). Retrieved from https://plotly.com/python/.
- Coursera's *The Science of Well-Being* by Yale University. (n.d.). Retrieved from https://www.coursera.org/learn/the-science-of-well-being.
- World Health Organization's Global Health Observatory Data Repository. (n.d.). Retrieved from https://www.who.int/data/gho.