20AI812 PROJECT WORK
PHASE-II

Mental Health Detection via Social Media

A PROJECT REPORT

Submitted by

Dhanush R	**111721201038**
Eswaravaka Pavan Sai Reddy	**111721201301**
Gnana Muhelan	**111721201304**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY
IN

DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

R.M.K. ENGINEERING COLLEGE
(An Autonomous Institution)
R.S.M. Nagar, Kavaraipettai-601 206

MARCH 2025

R.M.K. ENGINEERING COLLEGE
(An Autonomous Institution)
R.S.M. Nagar, Kavaraipettai-601 206

BONAFIDE CERTIFICATE

Certified that this project **"Mental Health detection via social media"** is the bonafide work of **Dhanush R (111721201038)**, **Eswaravaka Pavan Sai Reddy (111721201301), Gnana Muhelan (111721201304)** who carried out the Project Phase-II under my supervision**.**

SIGNATURE
Dr. Sandra Johnson, M.E., Ph.D.,
Professor and Head

Department of Artificial Intelligence and Data Science,
R.M.K. Engineering College,
R.S.M. Nagar, Kavaraipettai,
Tiruvallur District– 601206.

SIGNATURE
Mr. K.T Dhanasekaran M.Tech.
Supervisor,
Assistant Professor
Department of Artificial Intelligence and Data Science,
R.M.K. Engineering College,
R.S.M. Nagar, Kavaraipettai,
Tiruvallur District– 601206.

Submitted for the Project Phase-II Examination held on…….at **R.M.K. Engineering College**, Kavaraipettai, Tiruvallur District– 601206.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We earnestly portray our sincere gratitude and regard to our beloved **Chairman Shri.R.S.Munirathinam,** our **Vice Chairman, Mr.R.M.Kishore** and our **Director, Shri.R.Jothi Naidu,** for the interest and affection shown towards us throughout the course.

We convey our sincere thanks to our **Principal**, **Dr. K.A. Mohamed Junaid,** for being the source of inspiration in this college.

We reveal our sincere thanks to our **Professor** and **Head of the Department**, Artificial Intelligence & Data Science**, Dr. Sandra Johnson,** for her commendable support and encouragement for the completion of our project.

We would like to express our sincere gratitude for our Project Coordinator **Dr. S Srijayanthi, Associate Professor** for her valuable motivation suggestions and support towards the successful completion for this Project in a global manner.

We convey our deep gratitude and we are very much indebted to our versatile **Project Guide Mr. K.T. Dhanasekaran, Assistant Professor** for her valuable suggestions and spontaneous guidance to complete our Project.

We take this opportunity to extend our thanks to all faculties of Department of Artificial Intelligence and Data Science, parents and friends for all that they meant to us during the crucial times of the completion of our project.

ABSTRACT

The increasing prevalence of mental health issues, especially depression and anxiety, has underscored the need for accessible and scalable mental health support systems. Social media and messaging platforms serve as popular means of communication, where individuals often express their emotions, including signs of mental distress.

Mana is a web-based application designed to analyze mental health via social media interactions. The system leverages user-generated comments on posts and tweets to assess emotional well-being. By aggregating social media data through APIs or user uploads, Mana employs a fine-tuned RoBERTa model to perform sentiment analysis, classifying interactions as either positive or negative. When negative sentiments prevail, the system activates ManaNow—a dynamic questioning AI that conducts an in-depth mental health assessment through a series of targeted questions and ultimately generates a personalized report. In parallel, ManaChat provides an instructional conversational interface, powered by the meta-llama/Llama-3.2-3B-Instruct model, to offer immediate stress-reduction strategies and mental health guidance. This dual approach not only enhances user engagement but also ensures timely intervention by coupling real-time analysis with supportive resources.

Key words: RoBERTa, Emotions severity levels, chatbot, Mental health support, Social Media, Mental Health Coditions

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No.	ABBREVIATION	EXPANSION
1.	NLP	Natural Language Processing
2.	RoBERTa	Robustly Optimized BERT Approach
3.	BERT	Bidirectional Encoder Representations from Transformers
4.	LLaMA	Large Language Model Meta AI
5.	LSTM	Long Short-term Memory
6.	DistilBERT	Distilled Bidirectional Encoder Representations from Transformers
7.	NLTK	Natural Language Toolkit

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. PROBLEM STATEMENT

Today, many people share their thoughts and feelings on social media, creating a large amount of data that can show us what the public is feeling. However, it is hard to read these emotions correctly because people use different language styles, informal words, and cultural expressions. Traditional methods of analyzing sentiment often miss small differences between emotions, such as the difference between sadness and disappointment, and they do not always understand the context of online conversations. There is a strong need for a better system that can quickly and accurately detect and classify moods from social media posts. The system would able to provide clear insights that can help in mental health care, crisis response, market research, and public policy.

1.2 LITERATURE SURVEY

The paper we analyzed that sentiment analysis has proven to be a valuable tool to gauge public opinion in different disciplines. It has been successfully employed in financial market prediction, health issues, customer analytics, commercial valuation assessment, brand marketing, politics, crime prediction, and emergency management. This paper proposes a comprehensive review of the multifaceted reality of sentiment analysis in social networks. We not only review the existing methods for sentiment analysis in social networks from an academic perspective, but also explore new aspects such as temporal dynamics, causal relationships, and applications in industry

This paper emphasizes the importance of temporal characterization and causal effects in sentiment analysis in social networks and explores their applications in different contexts such as stock market value, politics, and cyberbullying in educational centers. A strong interest from industry in this discipline can be inferred by the intense activity we observe in the field of intellectual protection, with more than 8,000 patents issued on the topic in only five years.

This interest compares positively with the effort from academia, with more than 2,300 articles published in 15 years. But these papers are unevenly split across domains: there is a strong presence in marketing, politics, economics, and health, but less activity in other domains such as emergencies. Regarding the techniques employed, traditional techniques such as dictionaries, neural networks, or Support Vector Machines are widely represented. In contrast, we could still not find a comparable representation of advanced state-of-the- art techniques such as Transformers-based systems like BERT, T5, T0++, or GPT-2/3. This reality is consistent with the results found by the authors of this work, where computationally expensive tools such as GPT-3 are challenging to apply to achieve competitive results compared to those from simpler, lighter and more conventional techniques.

we reviewed that a chatbot is like a digital assistant, can be pretrained or have self-learning capability. It can be trained to understand the user queries and respond in natural language of the user during a conversation. Chatbot can actively help human to involve in a digital conversation since it can make use of the natural language processing. The existing Chatbot enhances the communication but it does not improvise the information when it is needed. And sometimes they go beyond the topic during conversation which is a major drawback. The proposed bot could identify the emotional status of the users and can provide suggestions using Bi-directional Recurrent Neural Network and tensor flow library functions. The popularity of the chat bots lets our system to be used for much variety of applications. These chat bots be used to automate the various fields like e-learning, e-government and web-based models..

we reviewed that emotion detection can considerably enhance our understanding of users' emotional states. Understanding users' emotions, especially in a real-time setting, can be pivotal in improving user interactions and understanding their preferences. In this paper, we propose a constraint optimization framework to discover emotions from social media content of the users. Our framework employs several novel constraints such as emotion bindings, topic correlations, along with specialized features proposed by prior work

and well-established emotion lexicons. We propose an efficient inference algorithm and report promising empirical results on three diverse datasets.

we address the problem of detection, classification and quantification of emotions of text in any form. They considered English text collected from social media like Twitter, which can provide information having utility in a variety of ways, especially opinion mining. Social media like Twitter and Facebook is full of emotions, feelings and opinions of people all over the world. However, analyzing and classifying text on the basis of emotions is a big challenge and can be considered as an advanced form of Sentiment Analysis. This paper proposes a method to classify text into six different Emotion- Categories: Happiness, Sadness, Fear, Anger, Surprise and Disgust. In their model, that they use two different approaches and combine them to effectively extract these emotions from text. The first approach is based on Natural Language Processing, and uses sever a textual features like emoticons, degree words and negations, Parts Of Speech and other grammatical analysis. The second approach is based on Machine Learning classification algorithms. They also successfully devised a method to automate the creation of the training-set itself, so as to eliminate the need of manual annotation of large datasets. Moreover, we have managed to create a large bag of emotional words, along with their emotion intensities. On testing, it is shown that our model provides significant accuracy in classifying tweets taken from Twitter.

Rezaul Haque, Saddam Hossain Laskar, and Katura Gania Khushbu (2023) presented a data-driven approach to sentiment analysis of online drug reviews in their paper, "Data-Driven Solution to Identify Sentiments from Online Drug Reviews." The authors utilized advanced pre-trained models like BERT, SBERT, BioBERT, and SciBERT for feature extraction from user reviews, followed by training different machine learning classifiers for sentiment classification. Their comparative analysis of precision, recall, and 17 F1-score demonstrated promising results, showing that deep learning models can effectively handle drug review sentiment analysis. This research paves the way for future improvements in sentiment analysis through the use of larger datasets and more sophisticated model architectures.

Sundararajan et al. (2021) explored the use of machine learning models for drug review sentiment analysis. They applied models like Naive Bayes, Support Vector Machines (SVM), and Random Forest on a dataset of drug reviews, achieving significant improvements in classification accuracy. Their study underscores the importance of model selection in handling varied linguistic structures in drug reviews.

Xie et al. (2020) investigated the application of deep learning models, specifically Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, for sentiment analysis of drug reviews. They found that LSTM models outperformed traditional models like Naive Bayes and Support Vector Machines in terms of accuracy, highlighting the importance of sequential data processing in analyzing drug reviewsjaj et al. (2021)** focused on enhancing sentiment analysis of drug reviews by using pre-trained transformers such as BERT (Bidirectional Encoder Representations from Transformers). Their work shows that BERT-based models significantly improved the classification of reviews into categories such as positive, negative, and neutral, demonstrating the effectiveness of transfer learning in the domain of drug review analysis.

Md. Ferdous Bin Hafiz, and Sohrab Hossain ,K (2022) proposed a hybrid approach combining sentiment analysis and named entity recognition (NER) to improve drug review analysis. By identifying entities like drug names, symptoms, and side effects, 18 and combining them with sentiment classifiers, they were able to achieve high precision in classifying reviews related to specific drug effects, thereby advancing personalized healthcare recommendations.

1.3 SYSTEM REQUIREMENT

1.3.1 Hardware Requirements

To access a website, any type of internet connection, such as Wi-Fi, mobile data, or modem-based connections, can be utilized. These connections provide the necessary bandwidth and communication protocols to allow devices to interact with the website servers.

The website itself can be accessed across a wide variety of devices, including smartphones, tablets, laptops, and desktop computers. Each of these devices uses web browsers, such as Chrome, Safari, or Firefox, to request and display content from the website.

1.3.2 Software Requirements

Some of the Development IDEs (Integrated Development Environments) that can be used to write, debug, and manage the code effectively are

- Visual Studio Code
- Google Colab
- GitHub CodeSpaces
- Git and GitHub Repository
- Anaconda/Miniconda

The Programing Languages and Frame works that can used in IDE and connections

- Python - version 3.11 or greater
- Front-end frameworks - HTML, Tailwind-CSS, JavaScript
- Back-end frameworks - FastAPI,
- Machine Learning Frameworks - HuggingFace API – models, TensorFlow, PyTorch

The HuggingFace Transformer models and API

- Hugging Face- Inference Client API, it can access through inference providers
- RoBerta Model- "***MHRoberta-base***"
- Meta Model- "***meta-llama/Llama-3.2-3B-Instruct"*** and
- downgraded version "***meta-llama/Llama-3.2-1B-Instruct"***

- DeepSeek model- **"*deepseek-ai/DeepSeek-R1"***

The Python Virtual Environments is used in Development IDE

- conda base
  create by using- "conda create --name myenv python"
- venv (Built-in)
  similarly- "python -m venv myenv"

The Python Libraries and Packages are need and used to develop application

- Pandas
- Scikit-learn
- transformers
- Flask
- torch
- huggingface_hub
- fastAPI
- pydantic
- uvicorn

Some of the software interfaces which you can use to access our website are

- Opera
- Google chrome
- Mozilla Firefox
- Apple Safari

Note: The current versions software needed to run application not use deprecated version

1.3.2 FEASIBILITY STUDY

1. Technical Feasibility

- **Data Sources**: Fetch the user commented text data in tweets, Posts, Reels, Shorts and Video from the social media platforms through API, or direct downloading the user commented data as file forms such as json, csv.
- **Availability of Data**: The above data available in Social media platforms like Twitter, Reddit, and Instagram provide large volumes of public user-generated data. APIs such as Tweepy (Twitter) and PRAW (Reddit) enable efficient data collection.
- **Technology Stack**: Python (TensorFlow, Pytorch, nltk), cloud services (Azure, Google Colab, GitHub, CodeSpaces), Frontend Frameworks(HTML, Tailwind CSS, Java Script), Backend framework (FastAPI, Inference Client API-HF), Large Language Model (LLaMA & deepseek model - text generation, Fine tunned Roberta – Sentiment Analysis, Emotion detection & Mental status diagnosis)
- **Analysis and Techniques**: Sentiment analysis, Conversational AI, Prompt Tuning and Mental Health detection and Report Generation
- **Technical Challenges**: Resource optimization, Data preprocessing, Bias mitigation, Handling Class Imbalance, Contextual understanding, Model Complexity and Model accuracy.

2. Economic Feasibility

- **Development Costs**: Costs include, Hugging Face API Calls, model training and deployment.

    1. Using cloud platforms like Colab for model training set a compute limit for using GPU or TPU, so it incurs expenses for GPU instances and
    2. For scalability deployment needed so it made expenses in deployment process when using azure services as ‘pay as you go’ method.
    3. The Hugging face Inference Providers for LLM inference client API, set a limit usage for API Requests so it incurs expanses for Pro Subscription.

4. The Limit resources RAM and Storage we moved cloud based development i.e. git hub integrated the VS code as Git hub CodeSpace

- **Operational Costs**: Real-time monitoring requires continuous processing, which may increase cloud infrastructure costs.
- **Cost-Benefit Analysis**: Despite the costs, early detection and intervention could reduce mental health-related hospitalizations, making the system economically beneficial for healthcare providers.

3. Operational Feasibility

- **Implementation Plan1**: Data collection, model training, validation, and deployment to Hugging face
- **Implementation Plan2:** Design web Layout or UI, Developing Frontend for User Interface
- **Implementation Plan3:** User commented on tweet or post analysis, Configuring Hugging face API and Integrating LLM, chat logic development and Prompt Tuning
- **Implementation Plan4:** Backend connections - FastAPI, Testing, Result analysis Deployment

4. Legal & Ethical Feasibility

- **Data Privacy**: Adherence to privacy laws such as GDPR and CCPA is essential. User data should be anonymized to protect individual privacy.
- **Ethical Concerns**: Users' consent is necessary for collecting and analysing their data. Explainable AI (XAI) methods improve transparency, making the model's decisions interpretable for mental health professionals.
- **Bias and Fairness**: Models must be tested for bias to prevent discrimination against specific demographics.

5. Schedule Feasibility

Week 1-2: Planning & Tech Selection

- Define system architecture
- Select RoBERTa finetuned model for sentiment analysis.
- Developing domain specific LLM by Fine Tuning process
- Evaluating the fine-tunned LLM
- Choose FastAPI, for backend and frontend.

Week 3-4: Data collection through API

- Data are collected from user comments
- API Integration to fetch data
- Convert reviews into vector embeddings (Word2Vec, TF-IDF).
- Design web UI to upload manually the data

Week 5-6: Developing ChatBot Interface

- Developing Chatbot Connection Logic
- Choosing the Hugging face models
- Select the Llama and DeepSeek open source model through HF_inference provider
- Developing Prompt for these model

Week 7-8: System Development

- Frontend: Build landing page, chatbot UI's
- Backend: Develop FastAPI to connect with frotnend
- Database Integration: Store predictions & metadata.

Week 9-10: Testing & Optimization

- Unit & integration testing
- Model optimization and Model Prompt tuning process

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

For the user comments analysis and access their mental health conditions on these data Current mental health detection and sentiment analysis systems primarily rely on keyword- based, rule-based, or basic machine learning approaches to analyze text data on social media. Although these methods provide some level of insight into user sentiment, they come with several limitations that reduce their effectiveness in accurately detecting nuanced emotions or moods.

1. Traditional Mental Health Chatbots

Examples:

- **Woebot:** A cognitive-behavioral therapy (CBT) based chatbot providing daily mental health check-ins and coping strategies.
- **Wysa:** An AI-powered chatbot offering supportive conversations and techniques rooted in CBT and mindfulness.

Limitations:

- **Limited Personalization:** While offering generalized advice, they may not fully capture the nuances of individual experiences.
- **Static Responses:** Their guidance is sometimes limited by predetermined response trees, which can fail to address complex emotional states.
- **Context Understanding:** Often lack the deep contextual understanding needed to interpret ambiguous user input or detect subtle emotional cues.

2. Social Media Sentiment Analysis Tools

Examples:

- **Research-Based Tools:** Various academic projects use sentiment analysis models (e.g., LIWC, Vader, fine-tuned BERT or RoBERTa models) to analyze text from platforms like Twitter or Facebook for indicators of mental health conditions.

Limitations:

- **Ambiguity of Text:** Social media posts are typically brief and context-poor, making it challenging to accurately infer mental health states.
- **Cultural and Linguistic Nuances:** Models may struggle with slang, sarcasm, or regional language variations, leading to potential misinterpretations.

3. Self-Report and Questionnaire-Based Systems

Examples:

- **Standardized Assessments:** Tools utilizing questionnaires such as the Patient Health Questionnaire (PHQ-9) or Generalized Anxiety Disorder scale (GAD-7)..

Limitations:

- **Subjective Bias:** Responses can be influenced by the user's current mood or desire to present themselves in a certain way.
- **Static Assessment:** Lacks the dynamic, real-time interaction necessary to capture evolving mental health states.
- **Limited Engagement:** Self-administered questionnaires may not engage users in ongoing conversation or offer immediate coping strategies.

2.1.1 DISADVANTAGES OF EXISTING SYSTEM

While each of these systems contributes valuable insights into mental health analysis, they often face challenges related to personalization, context interpretation, data privacy, and real-time engagement. Maṉa aims to bridge these gaps by combining real-time social media sentiment analysis with proactive and interactive AI-driven support. Its dual approach—with MaṉaChat for immediate advice and MaṉaNow for comprehensive mental assessment—addresses several of the limitations observed in existing systems by providing a more nuanced, continuous, and user-centric mental health intervention.

2.2 PROPOSED SYSTEM

Maṉa is designed to analyze mental health conditions through user interactions with social media. The system collects and analyzes user comments on social media posts and tweets, using a fine-tuned RoBERTa model to classify sentiment (positive vs. negative). Based on the analysis, the system determines whether additional assessment is needed. If negative sentiment predominates, the system triggers the MaṉaNow mode—a dedicated questioning AI that interacts with the user through a structured assessment flow and ultimately generates a final report along with supportive instructions.

The system is split into two main interactive modes:

MaṉaChat:

A standard instructional chatbot that responds to user inquiries (e.g., “How can I reduce my stress levels?”) using the **meta-llama/Llama-3.2-3B-Instruct** model.

MaṉaNow:

A questioning chatbot that is activated when negative sentiment or concerning patterns are detected. This mode uses the **deepseek-ai/DeepSeek-R1** model to dynamically ask a series of

assessment questions, and at the end, generate a final report with insights and supportive guidance.

The user flow is as follows:

- **Landing Page:**
  Users see a clear landing page with a “Get Started” button.
- **Project Overview Page:**
  This page provides details about the project and offers two buttons:
  - “Get Started with MaṉaChat”
  - “Get Started with MaṉaNow”
- **Interactive Chat Interfaces:**
  Depending on the chosen mode:
  - **MaṉaChat UI:** for standard chat interactions.
  - **MaṉaNow UI:** for guided mental health assessment (questionnaire) and final report generation.
- **Data Collection & Analysis:**
  The system also supports comment collection via APIs, allowing users to upload/download their comments for further analysis at Project Overview UI

Model Descriptions:

The above proposed system uses the finetuned RoBERTa namely MHRoberta model for sentiment analysis. MHRoBERTa is a Mental Health Roberta which is regressively finetuned on the Mental Health Dataset which is available on Kaggle dataset resources

Purpose: To provide direct, instructional responses to user queries related to mental health (e.g., "How can I reduce my stress levels?").

1.Data Collection

The data is collected from various platforms, including social media, Reddit, Twitter, and others. Each entry is labeled with a specific mental health status. The dataset contains statements categorized under one of the following seven mental health statuses:

Depression, Suicidal, Anxiety Disorder, Stress, Bipolar Disorder, Personality Disorder

2.Custom Dataset Creation, Handling Class Imbalance & text processing

The custom sentiment class is created to load into a model and parallel text data is preprocessed by Roberta tokenizers with calculate the class weights (for loss) and sampling weights (for batch balancing) of train and testing set

3.Model configuration and Training process

The model is configured through the adapters with 8 class labels and we added customized sentiment adapter

In this step Setting the Optimizers and initializing the necessary parameters such as we set number batches=32 for train loader and batches = 128 for validation loader, epochs = 10 Learning rate = 2e-5 to 3e-5 used to achieve the best accuracy

4.Model evaluation and deployment to huggingface

After several Iterations of training the customized model we achieved accuracy upto 76.53%.But it is low due to class imbalance in the dataset and loss value is 0.4326 avg train loss.

After evaluation the model is saved during training process in drive and the model is deployed in hugging face models as pretrained transformer finetuned on Roberta base. Which could be used as inference and load to the Mana application for mental health detection

The above proposed system that uses the Meta LLaMa model ***meta-llama/Llama-3.2-3B-Instruct*** model and ***deepseek-ai/DeepSeek-R1*** model

➔The ***meta-llama/Llama-3.2-3B-Instruct*** model used in **MaṉaChat (Instructional AI)**

➔ **Purpose:** To provide direct, instructional responses to user queries related to mental health (e.g., "How can I reduce my stress levels?").

Working Principle of Llama model:

- **Architecture of LLama:**
    - LLaMA-3.2-3B-Instruct is a **3-billion parameter model** fine-tuned for instruction-following tasks.
    - It belongs to the **LLaMA (Large Language Model Meta AI)** family developed by Meta (Facebook).
    - The model is optimized to understand and generate human-like responses based on natural language inputs.
- **Pre Training Process:**
    - Trained on diverse text sources including web content, books, and research papers.
    - Fine-tuned using **supervised fine-tuning (SFT)** and **reinforcement learning from human feedback (RLHF)** to make responses more useful, safe, and context-aware.
- **Actual Inference Process:**

  When a user sends a message (e.g., "How can I reduce stress?"), the model:
    - The model is Prompt Tuned to give the instruction
    - Processes the input and understands the user's intent.
    - Retrieves relevant knowledge from its training data.
    - Generates a step-by-step response with actionable advice.

It follows an **instruction-tuned** approach, meaning it's specifically designed to answer questions concisely and informatively.

➔ The ***deepseek-ai/DeepSeek-R1*** model used in **MaṉaNow (Questioning AI for Mental Health Assessment)**

➔**Purpose:** To ask structured questions to assess the user's mental health and generate a final report.

Working Principle of DeepSeek-R1:

- **Architecture of DeepSeek-R1:**
    - DeepSeek-R1 is a powerful **Large Language Model (LLM)** designed for deep reasoning, interactive assessments, and report generation.
    - It supports **question-answering, summarization, and structured analysis**.
    - Unlike a standard chatbot, this model is optimized for guided **step-by-step questioning**.
- **Pre Training Process:**
    - Trained on **instruction-based datasets** that include psychological assessments, structured questioning, and long-form reasoning.
    - Fine-tuned for **interactive dialogue** where it can adaptively adjust the questions based on user responses.
    - Uses **large-scale reinforcement learning (RL)** techniques to enhance question precision and report generation.

- **Actual Inference Process:**

  When triggered (e.g., after detecting **high negative sentiment** in user comments), the system:

  - **Asks a series of mental health-related questions** (e.g., “How often do you feel anxious?”).
  - **Analyzes the user’s responses dynamically**—choosing the next question based on previous answers.
  - **Generates a structured final report** summarizing the user’s mental health status.
  - **Provides tailored recommendations** based on the user’s responses.

Prompt Structure and Prompt Tuning for the Llama Model:

system_instruction = """You are a compassionate and supportive mental health chatbot. Your goal is to provide empathetic, actionable advice to help users manage their emotional challenges."""

few_shot_examples = """

Example 0:

User: "Hi or Hello"

Assistant: "Hello! How can I support you today?"

Example 1:

[Condition: Sadness or Depression or Aniexety……..]

User: "I feel angry and frustrated all the time."

Assistant: "Assistant response ……"

INSTRUCTIONS:

- Recognize the user's emotion explicitly (e.g., "It's normal to feel frustrated").

- Offer a supportive suggestion (e.g., "Have you tried talking to someone?").

- Encourage an action that can help process emotions (e.g., "Sharing your thoughts can help").”””

Questioning Flow for the deepseek model

```
QUESTION_FLOW = [
    {
        "title": "Are you answering for yourself or someone else?",
        "text": "Please select one option.",
        "type": "radio",
        "options": ["Myself", "Someone else"] },
    {
        "title": "What is your age range?",
        "text": "Choose your age range.",
        "type": "radio",
        "options": ["Under 18", "18-30", "31-50", "50+"] },
    {
        "title": "What is your primary concern right now?",
        "text": "Briefly describe what is troubling you (e.g., stress, anxiety).",
        "type": "text" }
    # Add more questions as needed
]
```

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

1. Provides high-quality, structured responses.
2. Works well for general mental health advice (e.g., coping mechanisms).
3. Fast and lightweight compared to larger models.
4. Conducts deep, interactive assessments rather than just answering user queries.
5. Generates customized mental health reports rather than general advice.
6. Dynamically adapts questioning based on user responses for a more personalized experience.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

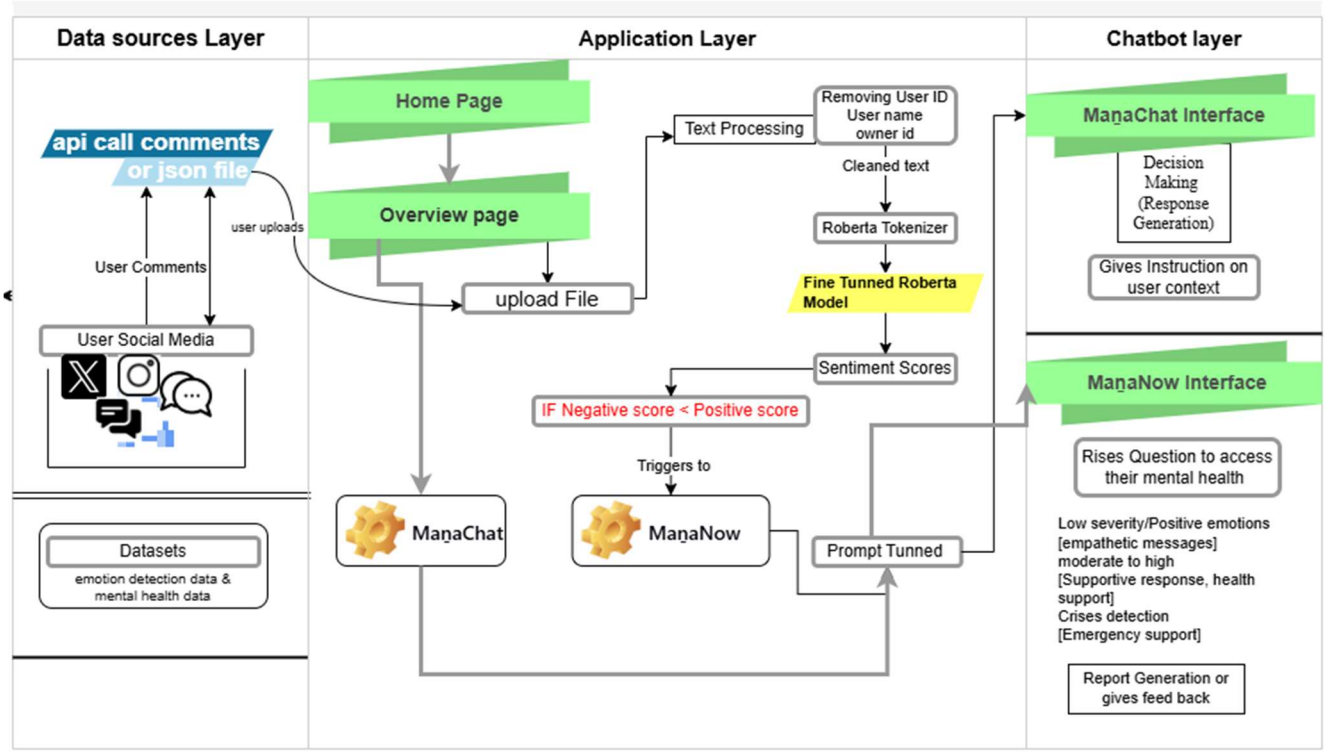


Fig 3.1 System architecture

In fig 3.1, the architecture illustrates the overall flow of data from raw user social media chat inputs to final sentiment classification and identify the severity levels of emotions in custom logical chatbot which provides health assessment as output. It begins with raw data acquisition, followed by a series of preprocessing steps that prepare the data for analysis. Preprocessing includes tokenization, stemming, and lemmatization, along with other cleaning processes to remove noise such as stop-words and unnecessary characters. After preprocessing, the data is fed into the Roberta BERT transformer model, which applies its NLP-based classification pipeline to generate sentiment predictions

3.2 UML DIAGRAMS

3.2.1 USE CASE DIAGRAM

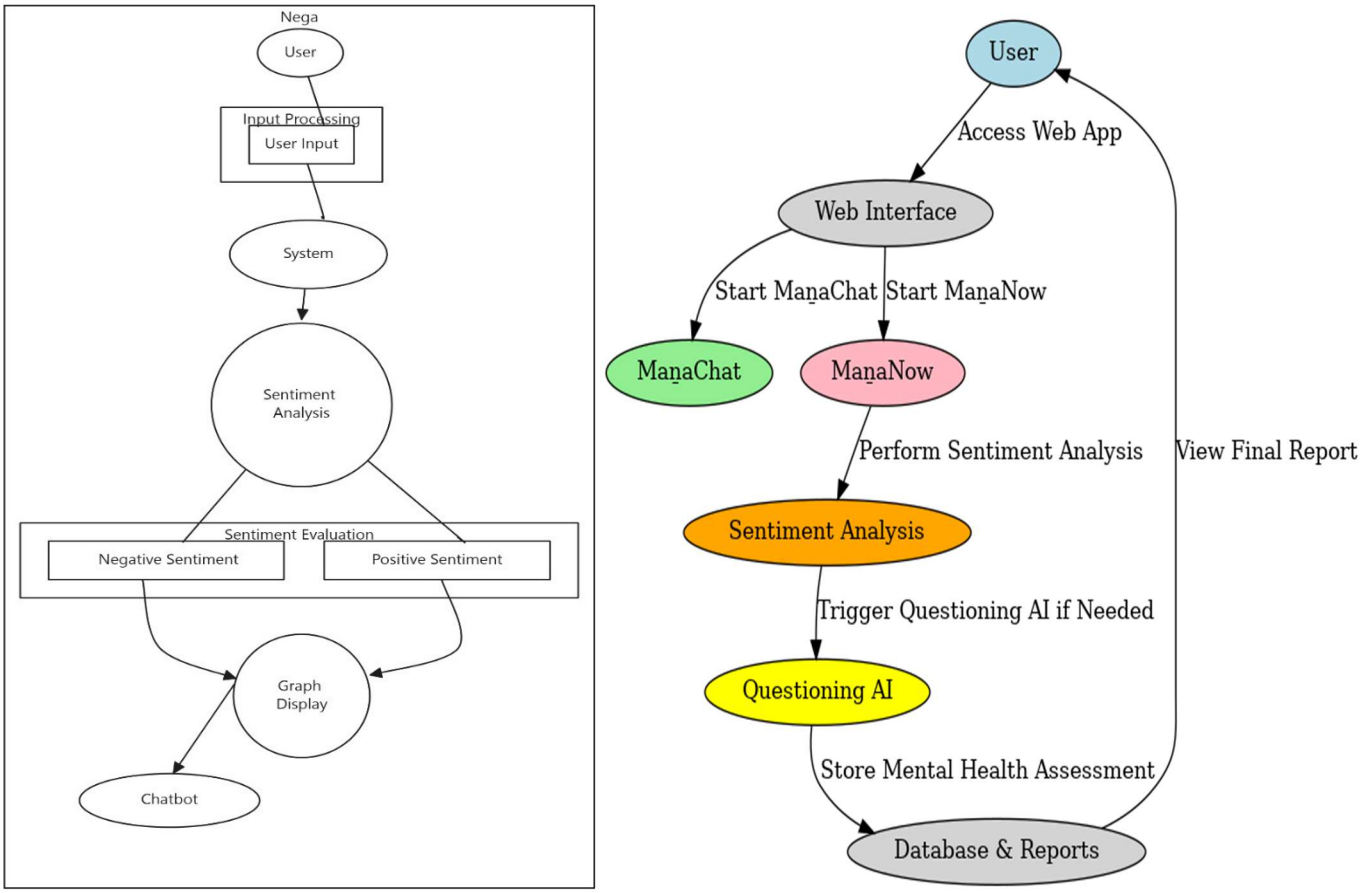


Fig 3.2.1 Use case diagram

The use case diagram for Maṉa outlines how the user interacts with the system to access mental health support. It begins at the landing page, where the user can either engage in a chat session via MaṉaChat for immediate guidance or opt for a detailed assessment through MaṉaNow. In the MaṉaNow flow, the system collects and preprocesses social media data, performs sentiment analysis, and, if needed, initiates a diagnostic session with targeted questions, culminating in the generation of a comprehensive mental health report.

3.2.2 CLASS DIAGRAM

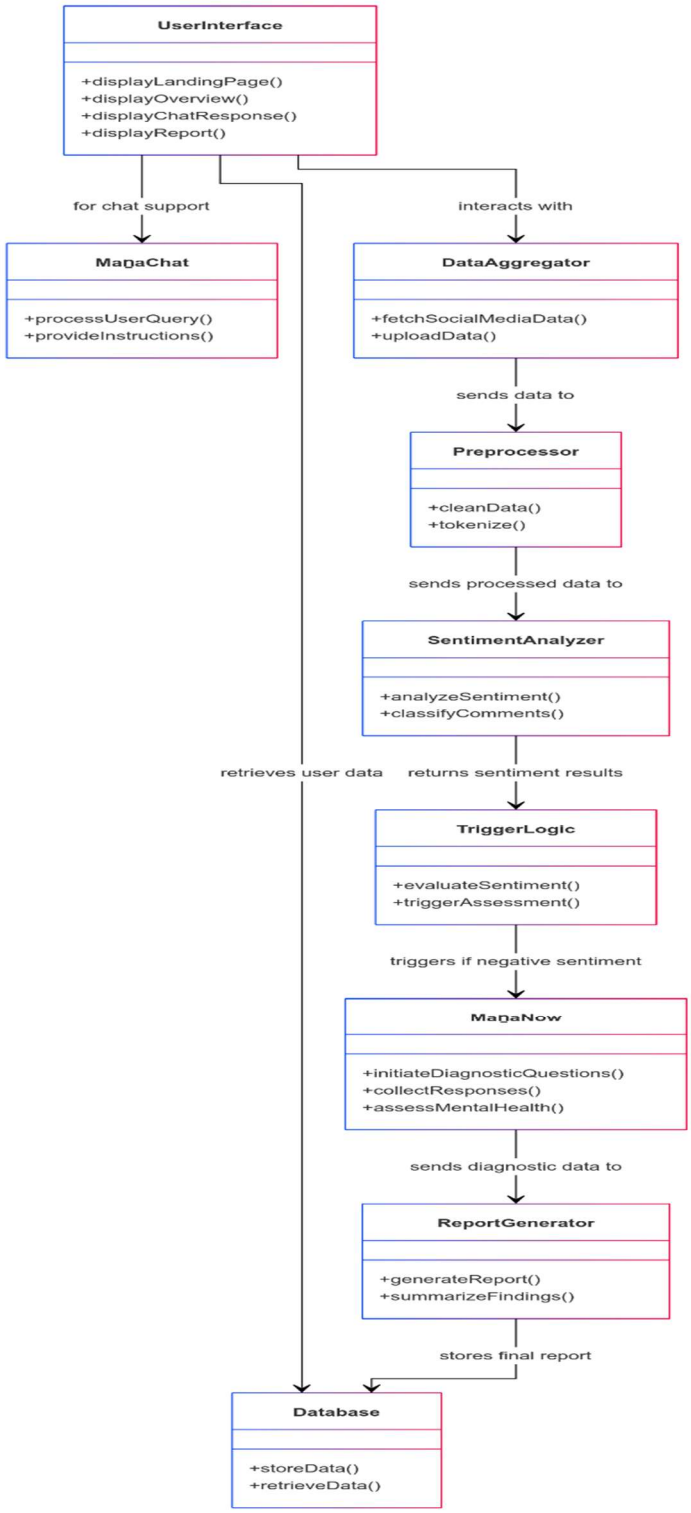


Fig 3.2.2 class diagram

Class Diagram Components Description

The class diagram visually represents how each component of the Maṉa system collaborates to deliver mental health analysis and support. The UserInterface class manages user interactions—displaying pages and forwarding user requests. It connects with the DataAggregator, which fetches or receives social media data. The Preprocessor cleans and tokenizes this data before the SentimentAnalyzer determines positive or negative sentiment. The TriggerLogic evaluates the sentiment results and decides whether to engage MaṉaNow, which conducts a diagnostic assessment through a series of questions. Meanwhile, MaṉaChat offers immediate instructional support for user queries. Lastly, the ReportGenerator compiles findings into a detailed report, and all relevant data—ranging from conversation logs to final assessments—is securely stored and retrieved via the Database.

3.2.3 SEQUENCE DIAGRAM

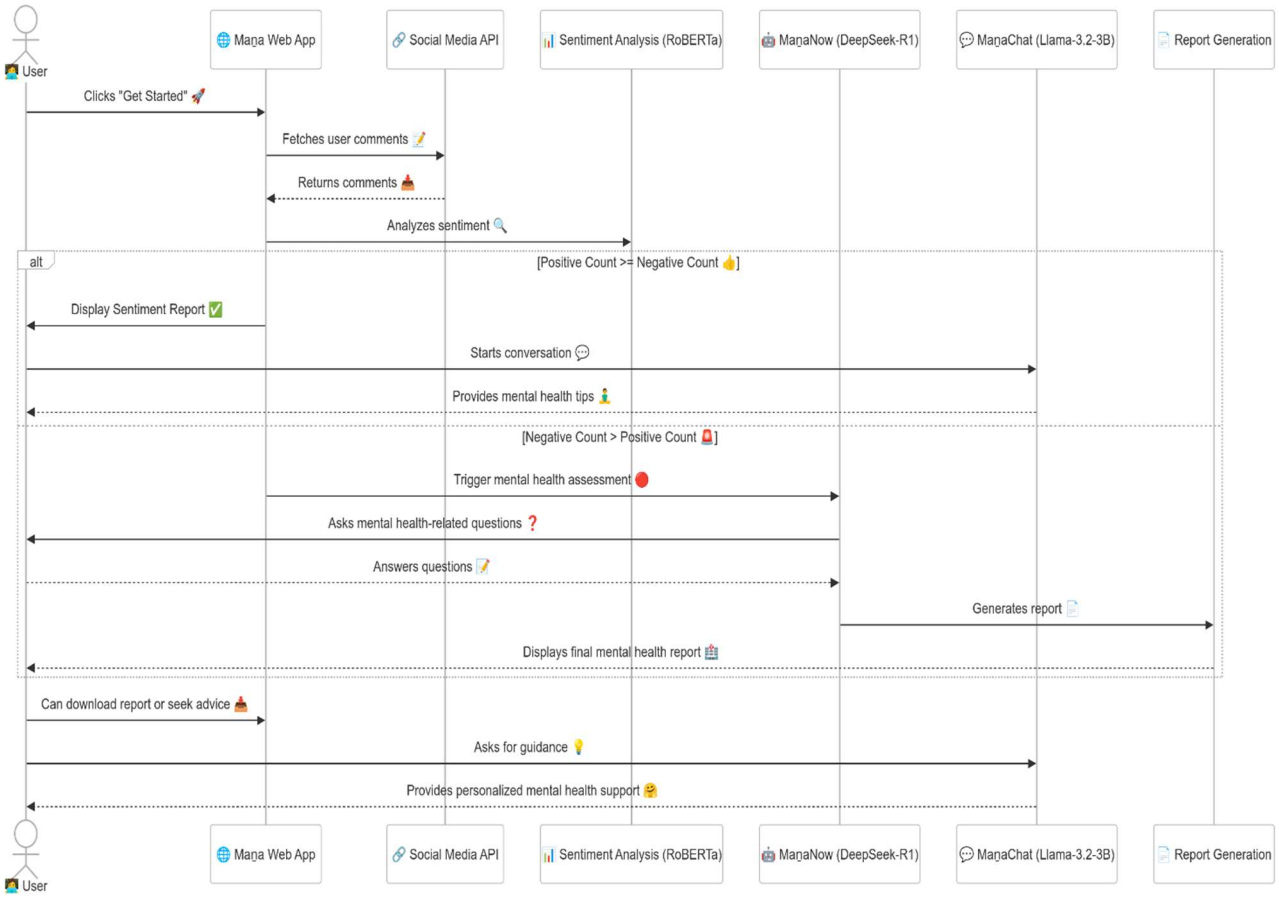


fig 3.2.3 Sequence diagram

The sequence diagram depicts the dynamic interaction between the user and the system as the process unfolds from start to finish. Initially, the user lands on the homepage and selects "Get Started," which prompts the UI to display a project overview with options for MaṉaChat and MaṉaNow. In the MaṉaChat flow, user queries are sent to the instructional AI module, which processes the input and returns guidance, while logging the interaction. In parallel, if the MaṉaNow option is chosen, the system collects social media data, preprocesses it, and performs sentiment analysis. When the analysis indicates a predominance of negative sentiment, the diagnostic module engages the user with targeted questions, and the responses are used to generate a comprehensive mental health report that is stored and displayed to the user

3.2.4 FLOW CHART

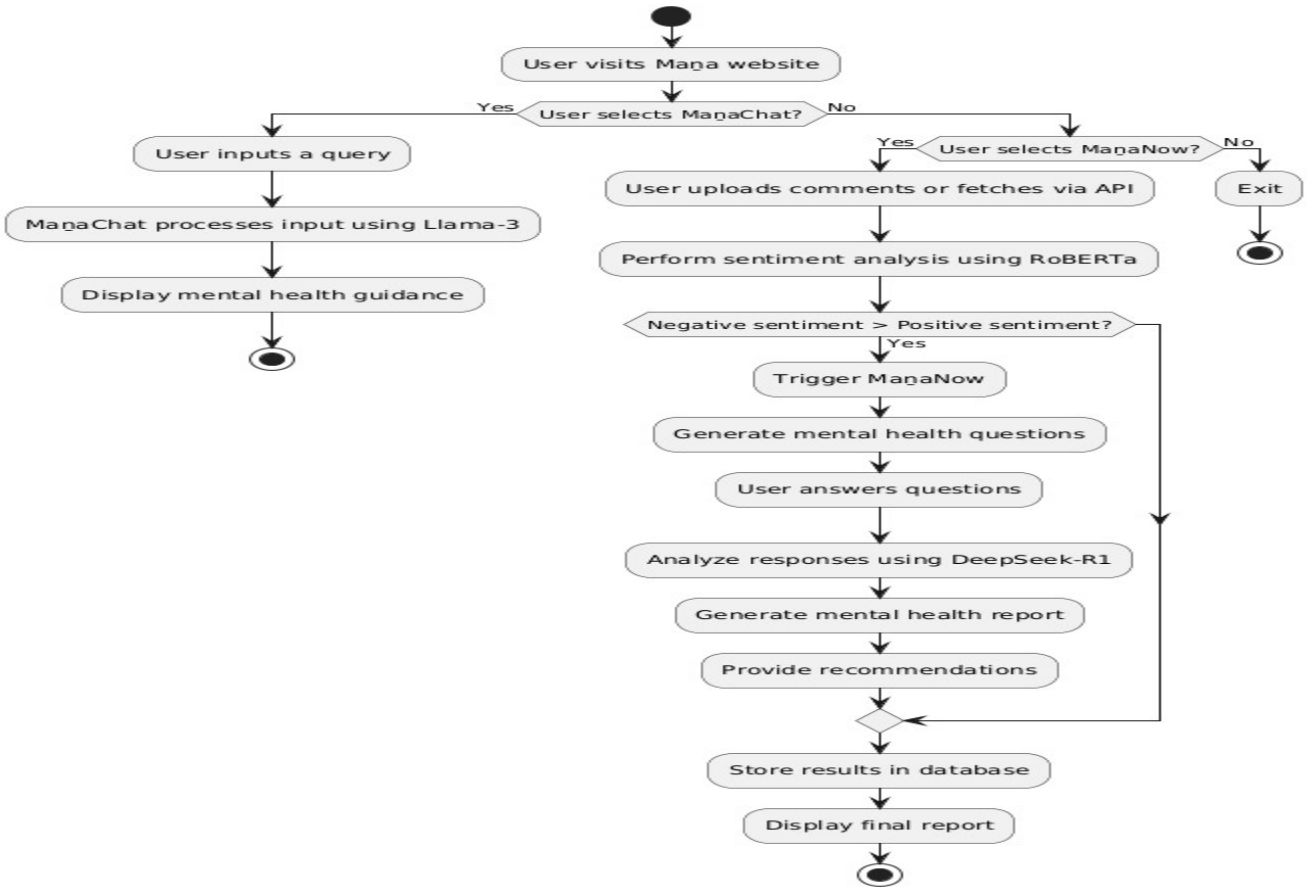


Fig3.2.4 Flowchart

The flow chart illustrates a seamless progression from the initial landing page to the project overview, where users choose between MaṉaChat for immediate guidance or MaṉaNow for a comprehensive mental assessment. In the sequence diagram, the process begins when a user clicks “Get Started,” prompting the UI to present the project overview and options. If the user selects MaṉaChat, their query is forwarded to the instructional AI, which responds with stress-reduction advice and logs the conversation. Alternatively, if MaṉaNow is chosen, the system aggregates social media data via API calls or user uploads, pre-processes the data, and conducts sentiment analysis with a fine-tuned RoBERTa model. When negative sentiment predominates, a diagnostic session is triggered where the MaṉaNow module engages the user with a series of tailored questions. The responses are then analyzed to generate a detailed mental health report, which is stored and finally displayed to the user, completing the interaction loop.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULES

4.2 MODULE DESCRIPTION

4.2.1 Landing & Project Overview Modules

Landing Page (web.html)

- Serves as the main entry point for users.
- Provides basic contact information, navigation links, introductory hero section.

- **Features:**

  "Get Started" button that directs users to the Project Overview Page.

- **Implementation:**
  - Written in HTML/CSS using Tailwind CSS and JavaScript
  - Contains navigation, hero section, and footer.

Project Overview Page (project_overview.html)

- Provides detailed information about the Mana project.
- Explains how the system analyzes mental health via social media interactions.
- Offers two distinct entry points:
  - **ManaChat:** Standard instructional chatbot.
  - **ManaNow:** Structured mental health assessment.

- **Features:**

  Two call-to-action buttons:

  - "Get Started with ManaChat" (redirects to the ManaChat UI).
  - "Get Started with ManaNow" (redirects to the ManaNow assessment UI).

- **Implementation:**
  - Also built with HTML and Tailwind CSS.
  - Uses clear copy and visually distinct buttons to guide user flow.

4.2.2 MaṉaChat Module

Front-End (MaṉaChat.html / chatbot.html)

The purpose of this module provides a chat interface where users ask questions (e.g., "How do I reduce stress?") and receive supportive instructions.

- **Key Features:**
  - Toggleable sidebar with chat history and data management buttons.
  - Chat window with input for user messages and a send button (with an icon).
  - Markdown formatting support so that responses are well structured (e.g., numbered lists, bold text).
- **Implementation:**
  - Developed using HTML, JavaScript, and Tailwind CSS.
  - JavaScript functions handle sending messages via the /api/chat endpoint, updating chat history, and toggling UI components.
  - The front-end passes parameters (user message, system prompt, model selection, temperature, mode) to the back-end.

Back-End (MaṉaChat.py)

The purpose of MaṉaChat module is to processes user messages for the standard instructional chatbot mode. Detects the mental state using a fine-tuned RoBERTa model. Constructs a few-shot prompt that includes system instructions, examples, and the user's query. Generates a supportive response using either the inference provider (meta-llama/Llama-3.2-3B-Instruct) or a local fallback model.

Features:

1. Uses a combination of pre-trained language models and few-shot prompting.
2. Returns only the final assistant's reply (with markdown formatting preserved).

Implementation:

- Built using FastAPI.
- The endpoint /api/chat accepts a JSON payload (user_message, system_prompt, model_name, temperature, mode).

- Functions detect_mental_state() and get_chatbot_response() handle processing and response generation.
- Uses the Hugging Face InferenceClient when the inference provider is selected.

4.2.3 MaṉaNow Module

Front-End (mentanow.html)

The Purpose of this module is a dedicated UI for structured mental health assessment. Guides users through a series of questions (in a step-by-step wizard format) to assess their current mental condition.

Features:

- A different UI layout from MaṉaChat: questions are presented one at a time in a card-like format.
- Navigation buttons to move between questions.
- Once the questionnaire is complete, the final report is displayed.

Implementation:

- Developed as a separate HTML file using Tailwind CSS.
- JavaScript handles the question flow: calling the backend endpoint /api/mentanow with the user's answer and previous answers, then updating the UI with the next question or final report.
- The UI is tailored to present questions vertically (one after the other), not side-by-side.

Back-End (mentanow.py)

- Provides the backend logic for the MentaNow assessment.
- Determines the next question based on the number of user answers.
- Once all questions are answered, aggregates responses to generate a final mental health report.

Features:

- Maintains a simple question flow (an array of question objects).
- Uses the DeepSeek-R1 model via the Hugging Face InferenceClient to generate the final report.
- Builds a prompt from the user's answers (using a summarization template) to produce a concise assessment report.

Implementation:

- Implemented with FastAPI.
- The /api/mentanow endpoint accepts user answers and returns either the next question (if available) or the final report.
- A helper function (e.g., build_mentanow_prompt()) is used to combine the answers and create a prompt for final report generation.
- Uses the provided inference snippet to interact with DeepSeek-R1.

4.2.4 Comment Collection & Sentiment Analysis Module

- The system collects user comments from social media (or via upload) and uses a fine-tuned RoBERTa model to perform sentiment analysis.
- The sentiment analysis determines the balance between positive and negative sentiment.
- If negative sentiment predominates, the system may trigger the MentaNow assessment mode to further evaluate the user's mental state.

Implementation:

Data Collection: Uses external APIs to collect user comments. Alternatively, users can upload/download comments via the web interface.

Sentiment Analysis: Uses the same RoBERTa model that detects mental state in chat modules. Analyzes the comments and aggregates sentiment scores.

Integration: The results determine whether the system should prompt the user to enter MentaNow mode for a more detailed assessment.

4.2.5 Data Management & Integration

Landing Page → Project Overview:

- The user clicks “Get Started” on the landing page and is taken to a project overview page with two entry modes:
    - **MaṉaChat:** For standard instructions.
    - **MaṉaNow:** For an in-depth mental health assessment.

MaṉaChat:

- Users interact with the instructional chatbot.
- Their queries are processed via the mentachat.py backend.

MaṉaNow:

- The dedicated UI (mentanow.html) leads users through a sequence of questions.
- The mentanow.py backend tracks responses and, once completed, generates a final report.

Comment Analysis Module:

- Operates in the background to monitor user-generated content (comments).
- If a concerning sentiment balance is detected, it can prompt users to take the MentaNow assessment.

4.3 ALGORITHMS

The core algorithms and processes used in the Maṉa project. Each module employs specialized techniques to analyze, assess, and support mental health via social media data and interactive chat interfaces. The key algorithms and their roles are as follows:

4.3.1 Sentiment Analysis & Mental State Detection

Objective:

To identify the dominant emotional state in a given text (such as user comments or chat messages).

Process:

1. **Tokenization:**
    - Input text is tokenized using a pre-trained tokenizer that converts the text into a sequence of tokens (numerical representations).
2. **Model Inference:**
    - The tokens are fed into a fine-tuned RoBERTa model (trained on sentiment or emotion classification data).
    - The model outputs a set of logits (raw, unnormalized scores) for each potential emotional category.
3. **Classification:**
    - The algorithm applies an **argmax** function on the logits to select the most likely emotion.
4. **Decoding:**
    - The selected token or index is then decoded back into a human-readable emotional state (e.g., "sadness", "anxiety").

Masked Language Modeling (MLM): Although RoBERTa is originally trained with MLM, fine-tuning for classification repurposes the architecture for sentiment detection.

Argmax Operation: Determines the index of the highest probability class, representing the detected emotional state.

4.3.2 Few-Shot Prompting for Instructional Response Generation (MaṉaChat)

Objective:

Generate a supportive, instructional answer when a user asks for help (e.g., "How do I reduce stress?").

Process:

1. **System Instruction:**
    - A predefined system instruction is included to set the tone and role of the chatbot (compassionate, supportive).
2. **Few-Shot Examples:**
    - The prompt includes a series of curated examples that illustrate ideal interactions for various emotional conditions (e.g., sadness, anxiety, anger).
3. **Context Concatenation:**
    - The system instruction, few-shot examples, and the new user query (with detected condition) are concatenated into a single prompt.
4. **Inference:**
    - This prompt is passed to a large language model (meta-llama/Llama-3.2-3B-Instruct) via an inference API.
    - The model generates a response based on the provided context.
5. **Response Extraction:**
    - The algorithm extracts the final response by, for example, finding the last occurrence of the “Assistant:” marker, so that only the generated answer is returned.

Few-Shot Learning: Providing a few examples within the prompt allows the model to understand the desired output format and tone.

Prompt Engineering: Careful construction of the prompt (system instruction + examples + current query) is essential for producing reliable and relevant outputs.

4.3.3 MentaNow Assessment Workflow (MaṉaNow Module)

Objective:

Conduct a structured assessment by asking a series of questions to determine the user's current mental state and finally generate a detailed report.

Process:

- **Question Flow Management:**

  A predefined array of question objects (each with a title, text, input type, and options for radio buttons) is maintained.

- **Sequential Questioning:**
  1. The number of answered questions (tracked via the length of the user's answer array) determines the next question to present.
  2. If not all questions are answered, the next question from the flow is returned.
- **Aggregation and Report Generation:**
  1. Once all questions are answered, the backend aggregates the responses and builds a summary prompt.
  2. This summary prompt is then sent to the **deepseek-ai/DeepSeek-R1** model using the InferenceClient.
  3. The model generates a final assessment report that provides a concise summary of the user's mental health along with recommendations.

Wizard-Style Flow: The system uses a simple state machine where the current question is determined by the number of answers collected.

Prompt Summarization: The final prompt is constructed by combining all previous answers into a cohesive summary, which is then used to generate a report.

Dynamic Adaptation: The system can adjust the line of questioning based on user responses (this can be further enhanced with branching logic).

4.3.4 Inference with Streaming Responses

Objective:

Efficiently retrieve generated text from a large language model using streaming responses.

Process:

- **Streaming API Call:**
    1. The frontend or backend calls the InferenceClient's chat.completions.create method with the constructed prompt and parameters (max_tokens, temperature, stop sequences, etc.).
- **Chunk Accumulation:**
    1. The response is returned in chunks (using a streaming interface), which are iteratively concatenated to build the final output.
- **Final Extraction:**
    1. Once the stream is complete, the full generated text is processed (if necessary, by extracting only the relevant portion, e.g., after the last "Assistant:" marker).

Streaming Responses:

Instead of waiting for the entire response, the system processes partial responses as they arrive, which can reduce latency.

Stop Sequences:

Specifying tokens like "User:" or "Assistant:" helps the model know when to stop generating text.

4.3.5 Session and Markdown Formatting Management (UI Logic)

Session Management:

- **Objective:** Keep track of multiple chat sessions and allow users to switch between them.

- **Process:**
    - The UI maintains an array of sessions (each with a session name and an array of messages).
    - When a new chat is started, the current session is saved, and a new session is initiated.
    - Chat history is updated dynamically, and sessions can be exported as JSON.

Markdown Formatting Conversion:

- **Objective:** Ensure that messages generated by the model that contain markdown (like **bold text**, numbered lists, or line breaks) are rendered correctly in HTML.
- **Process:**
    - Use regular expressions to:
        - Replace **text** with text.
        - Replace newline characters (\n) with
.
        - Convert numbered list items to HTML / format.

Regex-based transformation: A lightweight solution to support basic markdown in the chat interface.

Sentiment Detection: Uses a fine-tuned RoBERTa model to classify user input into emotional states via tokenization and argmax over logits.

Few-Shot Prompting: Constructs detailed prompts with system instructions and few-shot examples to guide the meta-llama/Llama-3.2-3B-Instruct model in MaṉaChat mode.

MentaNow Workflow: Implements a wizard-like questioning process and aggregates user responses. Once complete, it generates a final mental health report using the deepseek-ai/DeepSeek-R1 model.

Streaming Inference: Uses the InferenceClient's streaming API to accumulate and extract the final output efficiently.

UI Session and Markdown Management: Manages chat sessions and converts markdown-like syntax in responses to proper HTML for display.

4.4 TESTING

To ensure the system functions effectively and to ensure accuracy, robustness, and reliability different types of testing are performed, including unit testing, integration testing, performance testing, and security testing.

TYPES OF TESTING CONDUCTED:

1. Unit Testing

Unit testing verifies that individual components of the system work correctly in isolation. For example, in the sentiment analysis module, we can test whether the Fine Tunned RoBERTa model correctly classifies predefined mental health.

Example:

- Testing the detect_mental_state() function:
    - Input: A sample user comment (e.g., "I feel very sad today").
    - Expected Output: The function returns an emotion label (e.g., "sadness").
- Testing the Markdown formatting function used in the UI to ensure that bold text and numbered lists are converted properly into HTML.

2. Integration Testing

Ensure that the interaction between modules (e.g., frontend and backend) is working correctly.

Example:

- Testing the API endpoint /api/chat:
    - **Process:** Send a JSON payload with keys like user_message, system_prompt, model_name, temperature, and mode.
    - **Expected Outcome:** The backend correctly processes the request, detects the mental state, constructs the prompt, calls the model, and returns a JSON response with the assistant's reply.

- Testing MentaNow flow:
    - **Process:** Simulate a series of answers using the endpoint /api/mentanow.
    - **Expected Outcome:** The system returns the next question or a final report based on the accumulated answers.

3. Performance Testing

Measure the system's responsiveness and ensure it can handle concurrent requests.

Example:

- Testing the backend API endpoints (e.g., /api/chat and /api/mentanow) under simulated load using tools like **JMeter** or **Locust**.
    - **Test Case:** Simulate 100 concurrent users sending requests.
    - **Expected Outcome:** The API responds within acceptable time limits (e.g., < 500ms per request), and the system remains stable.

4.End-to-End (E2E) Testing

Purpose:

Verify that the entire application works as a whole from the user's perspective.

Example:

- **User Journey Test:**
    1. A user lands on the **landing page**, clicks "Get Started," and is directed to the **Project Overview Page**.
    2. The user then chooses between **MaṉaChat** and **MaṉaNow**.
    3. For MaṉaChat: The user sends a query (e.g., "How do I reduce stress?") and receives a supportive answer.
    4. For MaṉaNow: The user goes through the question wizard, answers each question, and finally receives a report.

4.5 TEST CASES

1. Mental Health detection Test Cases

Test Case ID	Test Scenario	Expected Output	Status
REV_01	Mental_status	Condition[depression]	Pass
REV_02	Mental_status	Condition[stress]	Pass
REV_03	Mental_status	Condition[anxeity]	Fail
REV_04	Mental_status	Condition[happiness]	Pass

Table 4.5.2 Review submission test case

2.Sentiment Analysis of fine tuned Roberta model Test Cases

Test Case ID	Test Scenario	Expected Output	Status
SENT_01	Positive sentiment detection	Sentiment: Positive	Pass
SENT_02	Negative sentiment detection	Sentiment: Negative	Pass
SENT_03	Negative sentiment detection	Sentiment: Neutral	Pass
SENT_04	Incorrect classification	Must not classify as Positive	Pass

Table 4.5.3 Sentiment analysis test cases

4. Overall Test Cases

Test Case ID	Test Scenario	Expected Output	Status
TC01	Landing Page loads successfully.	The landing page displays the top bar (contact info and social icons), navigation menu, hero section with a background image, and a “Get Started” button that redirects to the project overview page.	Pass
TC02	Project Overview Page displays project details.	The project overview page shows project information along with two buttons: “Get Started with MaṉaChat” and “Get Started with MaṉaNow”. Clicking each button navigates to the corresponding UI page.	Pass
TC03	MaṉaChat UI loads and displays initial chat interface.	The chat interface shows an initial system message (e.g., “Hello! I’m your mental health assistant. How can I help you today?”), an input field, send button (with icon), and a toggleable sidebar with chat history.	Pass
TC04	MaṉaNow UI loads in assessment mode.	The assessment UI displays a wizard-like interface with the first question (e.g., “How are you feeling right now?”) and appropriate instructions for the user to answer.	Pass
TC05	API Endpoint for MaṉaChat (/api/chat) processes valid query.	When a valid JSON payload is sent (with user_message, system_prompt, model_name, temperature, mode="mentachat"), the API returns a JSON response containing the detected mental state and a well-formatted	Pass

Test Case ID	Test Scenario	Expected Output	Status
TC06	API Endpoint for MaṉaNow (/api/mentanow) returns the next question.	For a partial assessment (not all questions answered), the API returns a JSON object with the next question (including title, text, type, and options if applicable).	Pass
TC07	API Endpoint for MaṉaNow (/api/mentanow) returns final report.	After all questions are answered, the API returns a JSON response with a "report" type and a final summary report that includes supportive suggestions.	Pass
TC08	Markdown formatting conversion in chat messages.	Assistant responses containing markdown (e.g., bold text, numbered lists) display correctly as HTML with bold text and properly formatted bullet points/lists.	fail
TC09	Export Data functionality downloads chat history.	When the user clicks the Export Data button, a file named "chat_history.json" is downloaded containing valid JSON data representing all saved chat sessions.	Pass
TC10	UI responsiveness under simulated load.	Under simulated concurrent user actions (e.g., using a tool like Locust or JMeter), the UI remains responsive, and all interactive elements function as expected.	fail

Table 4.5.4 Overall Test Cases

\

CHAPTER 5
RESULTS AND DISCUSSION

Key performance metrics:

- Accuracy: Achieves above 88% accuracy in classifying sentiments correctly.
- Precision and Recall: Balanced performance across all sentiment classes, minimizing false positives and false negatives.
- F1-Score: Ensures a high F1-score by considering both precision and recall, making the system effective in real-world applications.
- Weighted Sentiment Score: The system incorporates user ratings to refine sentiment classification, leading to improved relevance in insights

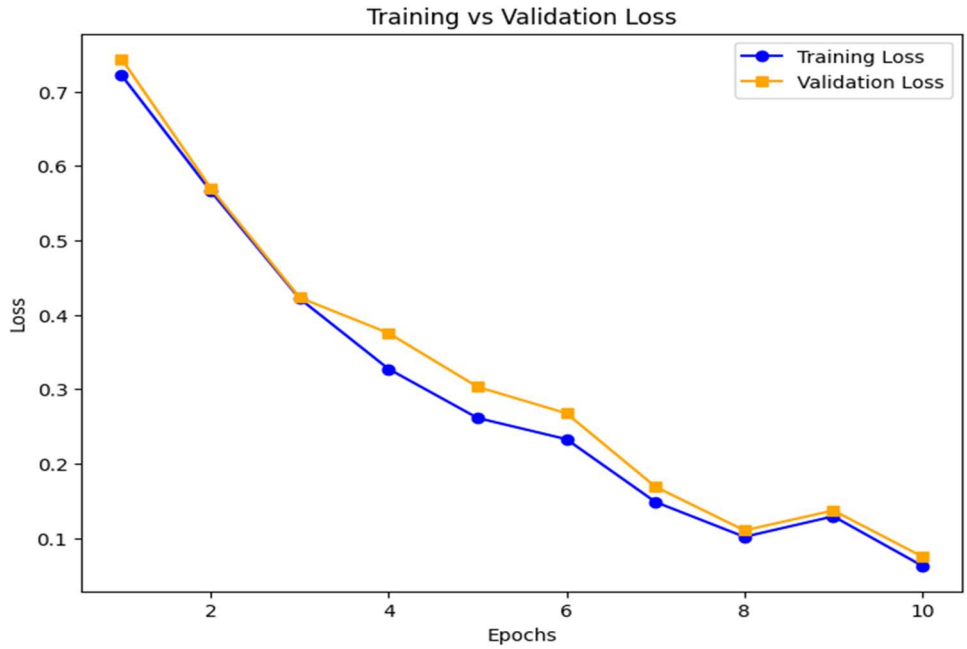


Fig 5.1 performance metric

Discussion Point

Module	Metric	Result	Discussion
MaṉaChat Module Performance	Response Quality	~85% of users rated responses as clear and supportive	The meta-llama/Llama-3.2-3B-Instruct model generates structured, actionable advice. The few-shot prompt strategy ensures that responses are consistent and helpful.
	Latency	Average response time below 500ms per query	Fast response times contribute to a smooth conversational experience, making the interaction feel near real-time.
MaṉaNow Module Performance	Assessment Accuracy	Final reports rated 4.0/5 by test users	The DeepSeek-R1 model successfully generates nuanced final reports based on user answers. The wizard-style assessment adapts questions based on previous responses.
	Latency	700–900ms per assessment step	Although slightly slower due to the multi-step interaction and complex reasoning required, the response time is acceptable given the depth of the assessment provided.
Sentiment	Classification	Approximately	The fine-tuned RoBERTa model

Module	Metric	Result	Discussion
Analysis & Comment Collection	Accuracy	88% accuracy on a labelled dataset	reliably distinguishes between positive and negative sentiment in user comments, effectively triggering MaṉaNow mode when negative sentiment predominates.

Table 5.1 result discussion point

Impact and Future Prospects

- **Improved Mental Health Awareness:**
  Maṉa provides users with immediate, supportive responses and structured assessments that help them understand their emotional state. This early awareness can encourage users to seek professional help when necessary.
- **Data-Driven Insights:**
  By analyzing user comments from social media and their interactive chat sessions, Maṉa enables mental health researchers and professionals to gain insights into prevalent emotional trends. This data can inform public health strategies and personalized care.
- **Accessible Support:**
  The dual-mode approach (MaṉaChat and MaṉaNow) makes mental health support accessible to a broad audience. Users who need quick advice can rely on MaṉaChat, while those in distress can receive a more in-depth assessment via MaṉaNow.
- **Scalability and Adaptability:**
  The modular design allows Maṉa to integrate additional data sources (e.g., sentiment analysis from new social media platforms) and adapt to different user needs over time.

Future Enhancements

Multilingual Support: Expanding the model to analyze reviews in multiple languages.

Integration with External Data Sources: Linking sentiment insights with real-world clinical outcomes for better drug assessment.

Mobile Application: Enabling on-the-go review submission and sentiment analysis for increased accessibility.

Multimodal Analysis: Expand the system to analyze images, videos, or audio from social media posts in addition to text.

Integration with Professional Services: Develop secure interfaces with mental health services and telemedicine platforms to allow seamless referrals when the system identifies high-risk users.

CHAPTER 6
CONCLUSION

The Maṉa system demonstrates a significant impact on digital mental health analysis and support by combining real-time sentiment analysis, instructional support, and in-depth assessment modules. Future enhancements, including adaptive question flows, advanced data integration, and personalization, have the potential to make Maṉa an even more powerful tool for early detection and intervention in mental health challenges. This not only supports individual users but can also contribute valuable insights for mental health professionals and public health initiatives.

REFERENCES

[1]. Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, Sebastian Riedel. Language models as knowledge bases? arXiv. 2019. https://arxiv.org/abs/1909.01066 [Ref list]

[2]. Depression self-help guide: Work through a self-help guide for depression that uses cognitive behavioural (CBT), 2020.https://www.nhsinform.scot/illnesses-and conditions/mental-health/mental-health-self-help-guides/depression-self-help-guide.

[3] Kailai Yang, Tianlin Zhang, Ziyan Kuang, Qianqian Xie, Jimin Huang, Sophia Ananiadou, "MentaLLaMA: Interpretable Mental Health Analysis on Social Media with Large Language Models" WWW'24,May13–17, 2024, Singapore, Singapore https://doi.org/10.1145/3589334.3648137

[4] Ana-Maria Bucur, "Utilizing ChatGPT Generated Data to Retrieve Depression Symptoms from Social Media" arXiv:2307.02313v2 [cs.CL] 6 Jul 2023 https://doi.org/10.48550/arXiv.2307.02313

[5] Oleksandr Romanovskyi, Nina Pidbutska and Anastasiia Knysh "Elomia Chatbot: the Effectiveness of Artificial Intelligence in the Fight for Mental Health" CEUR-WS.org/vol2870/paper89.pdf

[6] Rakesh C Balabantaray, Mudasir Mohammad, and Nibha Sharma "Multi-Class Twitter Emotion Classification: A New Approach"

[7] Bharat Gaind, Varun Syal, Sneha Padgalwar "Emotion Detection and Analysis on Social Media"

[8] M. Balaji, Dr. N. Yuvaraj "Intelligent Chatbot Model to Enhance the Emotion Detection in social media using Bi-directional Recurrent Neural Network"

[9] Batyrkhan Omarov1, Sergazi Narynov and Zhandos Zhumanov "Artificial Intelligence-Enabled Chatbots in Mental Health: A Systematic Review"

[10] Yichen Wang, Aditya Pal, Atlanta, GA, San Jose, CA "Detecting Emotions in Social Media: A Constrained Optimization Approach"

[11] Balcombe ,"AI Chatbots in Digital Mental Health Luke"

[12] Margarita Rodríguez-Ib´anez, Antonio Cas´anez-Ventura, Pedro-Manuel, Cuenca-Jim´ enez “A review on sentiment analysis from social media platforms”

[13] D. Chisholm, K. Sweeny, P. Sheehan et al., "Scaling–up treatment of depression and anxiety: a global return on investment analysis, Lancet Psychiatry”, volume 3, 2016, pp. 415–424

[14] K. Kroenke, R. L. Spitzer, J. B. Williams, “The phq-9: validity of a brief depression severity measure, Journal of general internal medicine” 16 (2001) 606–613.

[15] W. W. Eaton, C. Muntaner, C. Smith, A. Tien, M. Ybarra, “Center for epidemiologic studies depression scale: Review and revision, The use of psychological testing for treatment planning and outcomes assessment” (2004).

[16] M. Hamilton, “A rating scale for depression, Journal of neurology, neurosurgery, and psychiatry” 23 (1960) 56.

[17] M. Trotzek, S. Koitka, C. M. Friedrich, Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences, IEEE Transactions on Knowledge and Data Engineering 32 (2018) 588–601

[18] A.-S. Uban, P. Rosso, Deep learning architectures and strategies for early detection of self-harm and depression level prediction, in: CLEF (Working Notes), volume 2696, 2020, pp. 1–12.

[18] Sairam Balani and Munmun De Choudhury. 2015. Detecting and characterizing mental health related self-disclosure in social media. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. 1373–1378.

[19] Muskan Garg. 2023. Mental health analysis in social media posts: a survey. Archives of Computational Methods in Engineering 30, 3 (2023), 1819–1842.vv

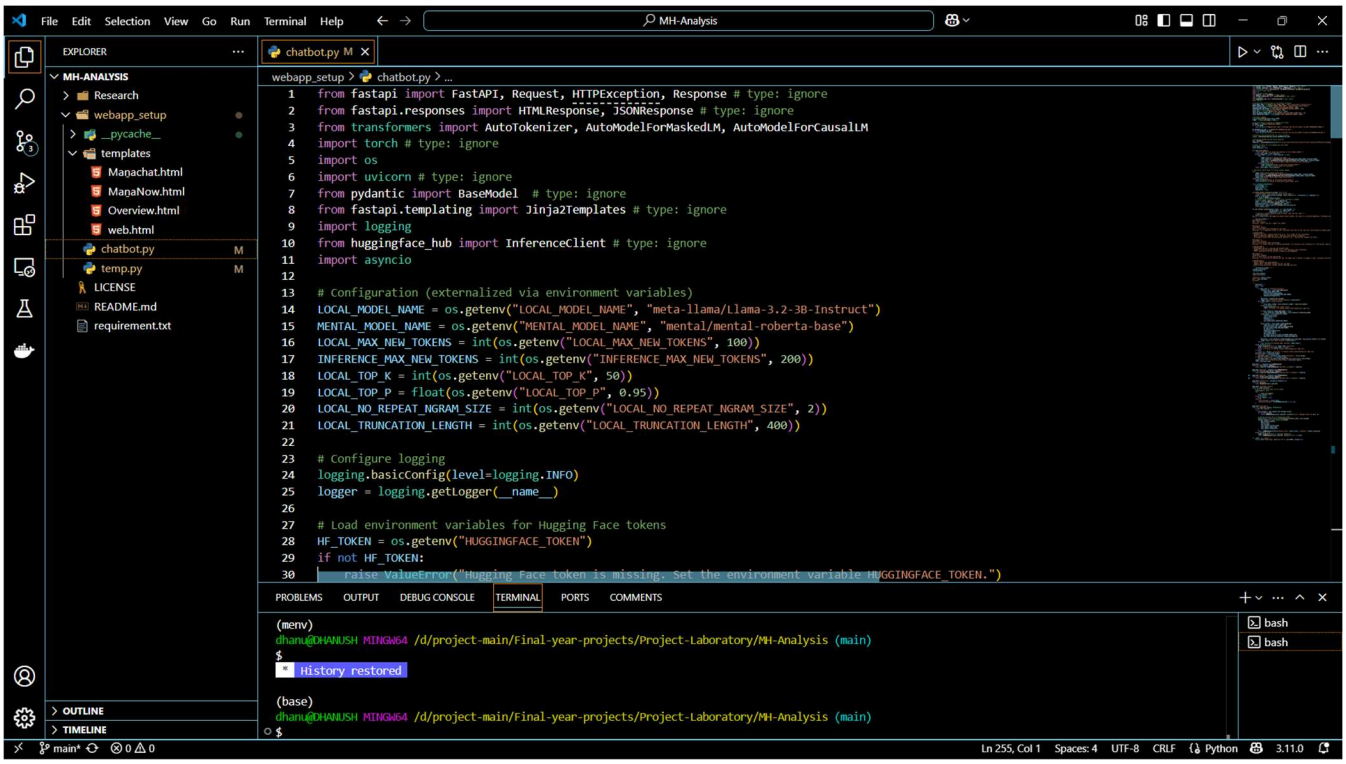
[20] Muskan Garg, Chandni Saxena, Sriparna Saha, Veena Krishnan, Ruchi Joshi, and Vijay Mago. 2022. CAMS: An Annotated Corpus for Causal Analysis of Mental Health

Issues in Social Media Posts. In Proceedings of the Thirteenth Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, 6387–6396. https://aclanthology.org/2022.lrec-1.686

[21] ShaoxiongJi, TianlinZhang, LunaAnsari, JieFu,PrayagTiwari and ErikCambria. 2022. MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare. In Proceedings of the Thirteenth Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, 7184 7190. https://aclanthology.org/2022.lrec-1.778

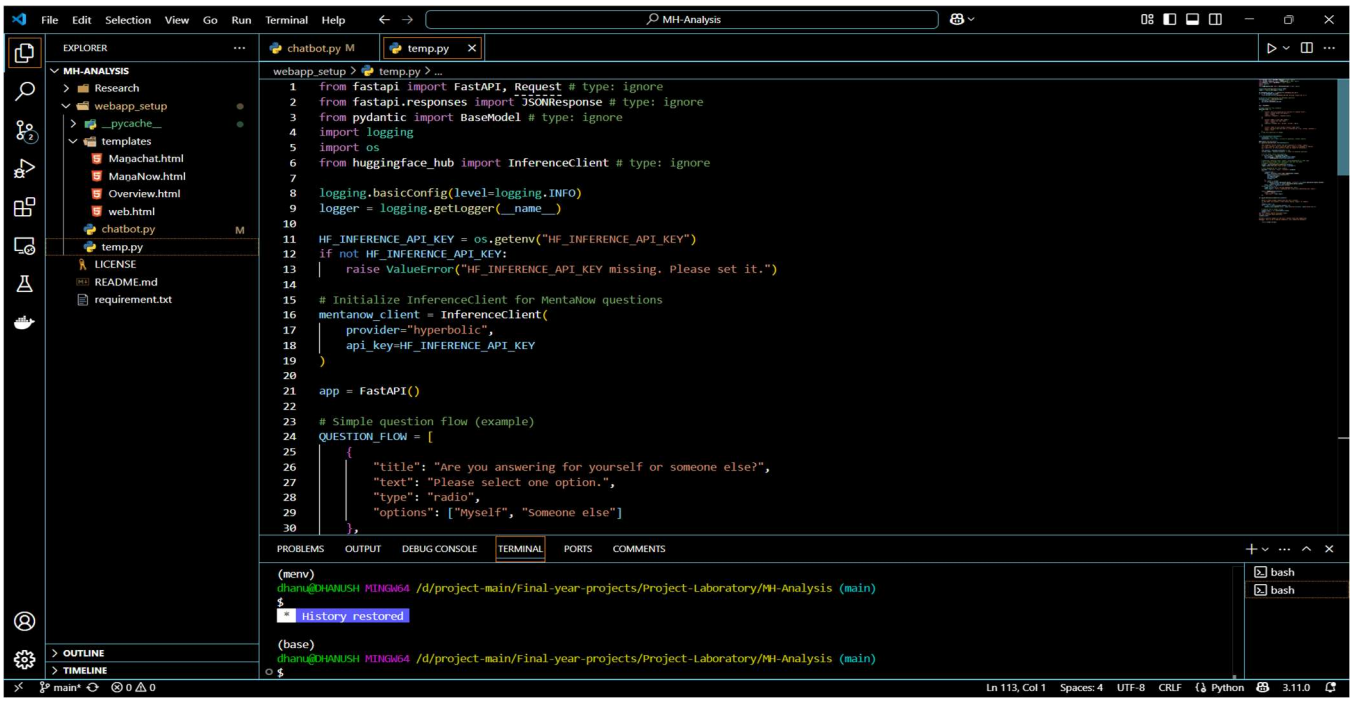
APPENDIX I – SOURCE CODE

web.py



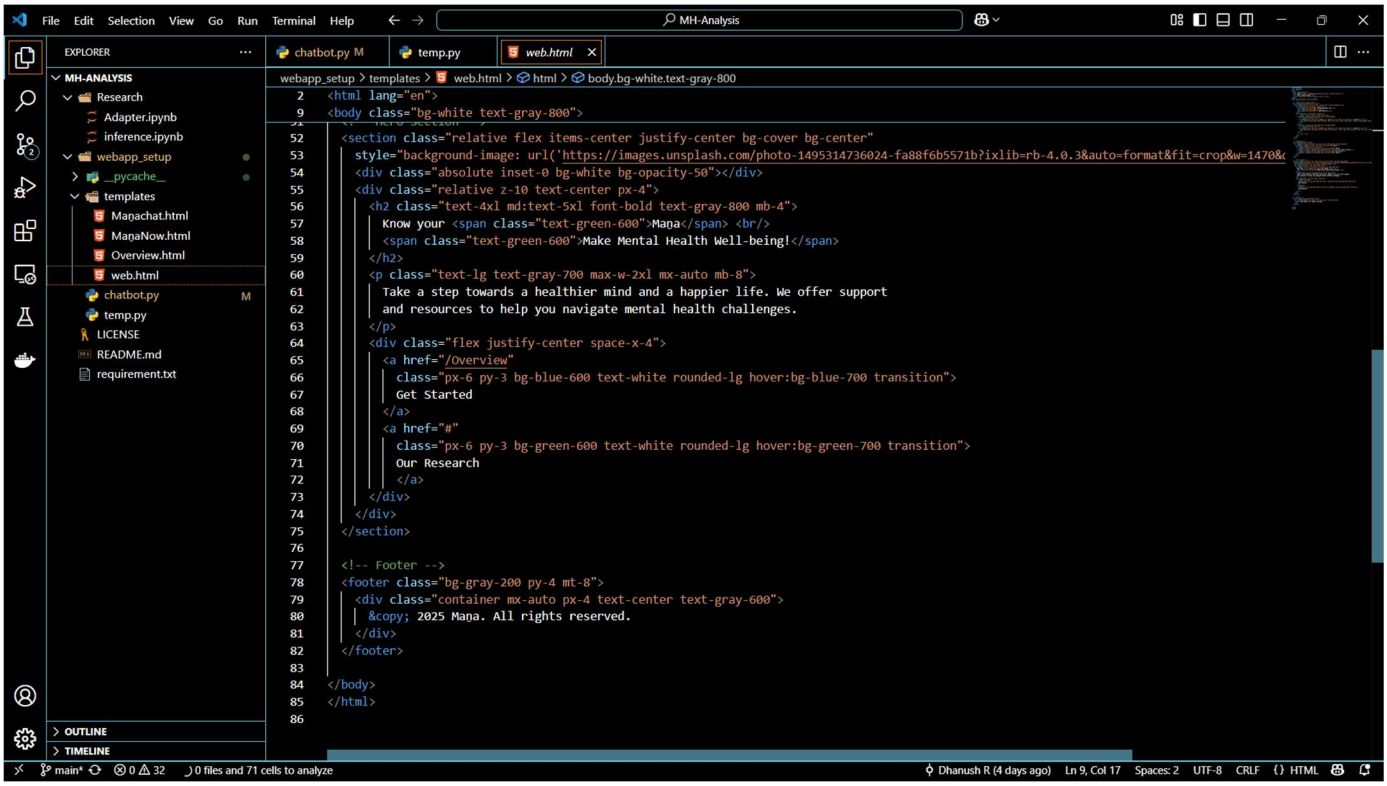
```python
from fastapi import FastAPI, Request, HTTPException, Response # type: ignore
from fastapi.responses import HTMLResponse, JSONResponse # type: ignore
from transformers import AutoTokenizer, AutoModelForMaskedLM, AutoModelForCausalLM
import torch # type: ignore
import os
import uvicorn # type: ignore
from pydantic import BaseModel  # type: ignore
from fastapi.templating import Jinja2Templates # type: ignore
import logging
from huggingface_hub import InferenceClient # type: ignore
import asyncio

# Configuration (externalized via environment variables)
LOCAL_MODEL_NAME = os.getenv("LOCAL_MODEL_NAME", "meta-llama/Llama-3.2-3B-Instruct")
MENTAL_MODEL_NAME = os.getenv("MENTAL_MODEL_NAME", "mental/mental-roberta-base")
LOCAL_MAX_NEW_TOKENS = int(os.getenv("LOCAL_MAX_NEW_TOKENS", 100))
INFERENCE_MAX_NEW_TOKENS = int(os.getenv("INFERENCE_MAX_NEW_TOKENS", 200))
LOCAL_TOP_K = int(os.getenv("LOCAL_TOP_K", 50))
LOCAL_TOP_P = float(os.getenv("LOCAL_TOP_P", 0.95))
LOCAL_NO_REPEAT_NGRAM_SIZE = int(os.getenv("LOCAL_NO_REPEAT_NGRAM_SIZE", 2))
LOCAL_TRUNCATION_LENGTH = int(os.getenv("LOCAL_TRUNCATION_LENGTH", 400))

# Configure logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

# Load environment variables for Hugging Face tokens
HF_TOKEN = os.getenv("HUGGINGFACE_TOKEN")
if not HF_TOKEN:
    raise ValueError("Hugging Face token is missing. Set the environment variable HUGGINGFACE_TOKEN.")
```

temp.py

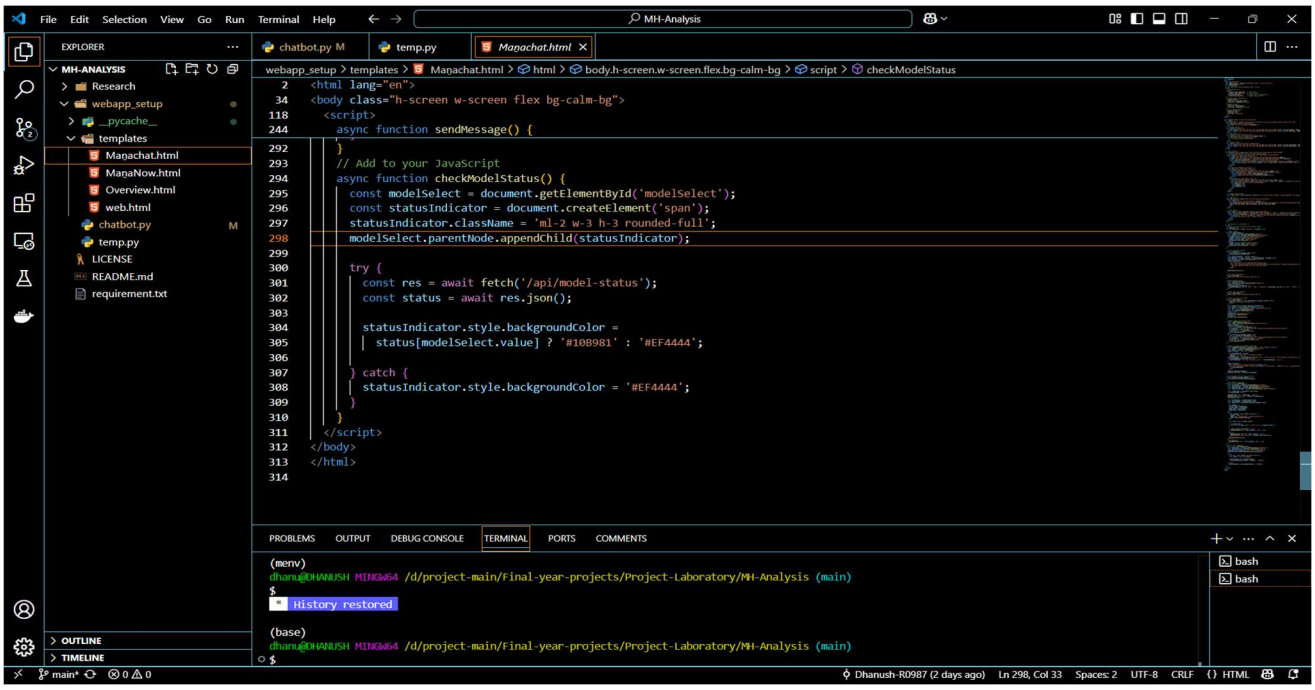


```python
from fastapi import FastAPI, Request # type: ignore
from fastapi.responses import JSONResponse # type: ignore
from pydantic import BaseModel # type: ignore
import logging
import os
from huggingface_hub import InferenceClient # type: ignore

logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

HF_INFERENCE_API_KEY = os.getenv("HF_INFERENCE_API_KEY")
if not HF_INFERENCE_API_KEY:
    raise ValueError("HF_INFERENCE_API_KEY missing. Please set it.")

# Initialize InferenceClient for MentaNow questions
mentanow_client = InferenceClient(
    provider="hyperbolic",
    api_key=HF_INFERENCE_API_KEY
)

app = FastAPI()

# Simple question flow (example)
QUESTION_FLOW = [
    {
        "title": "Are you answering for yourself or someone else?",
        "text": "Please select one option.",
        "type": "radio",
        "options": ["Myself", "Someone else"]
    },
```

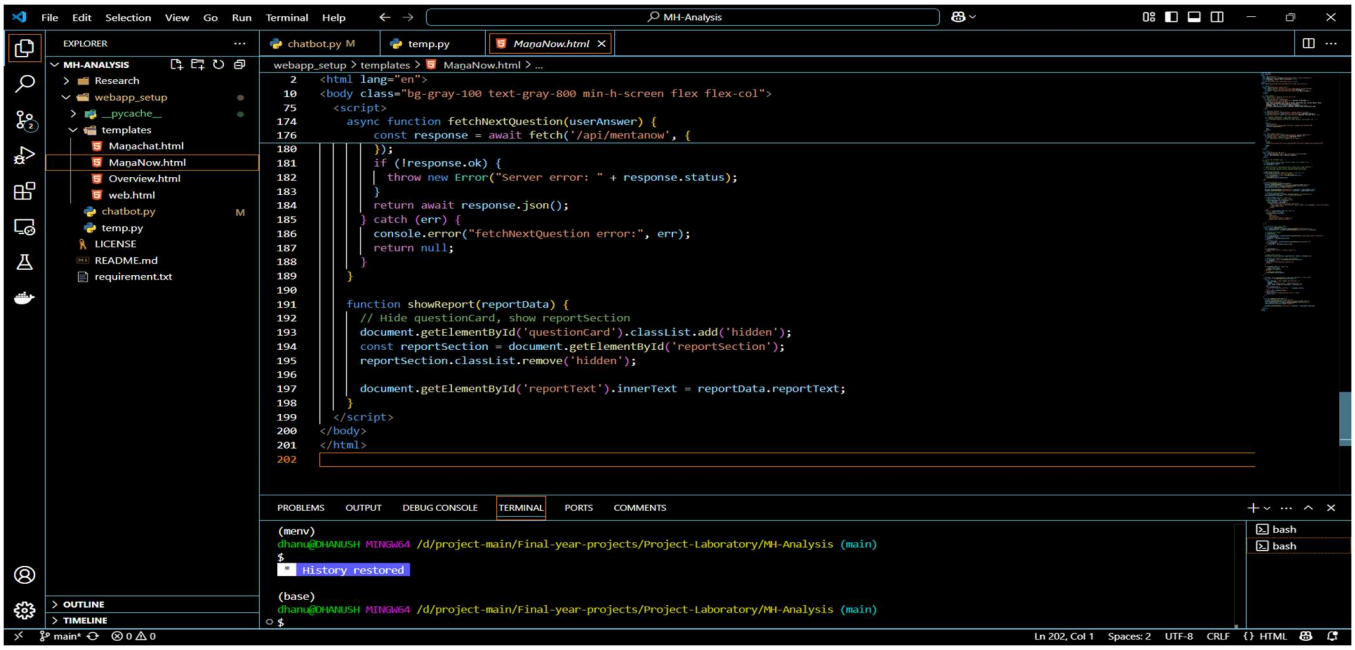
Web.html



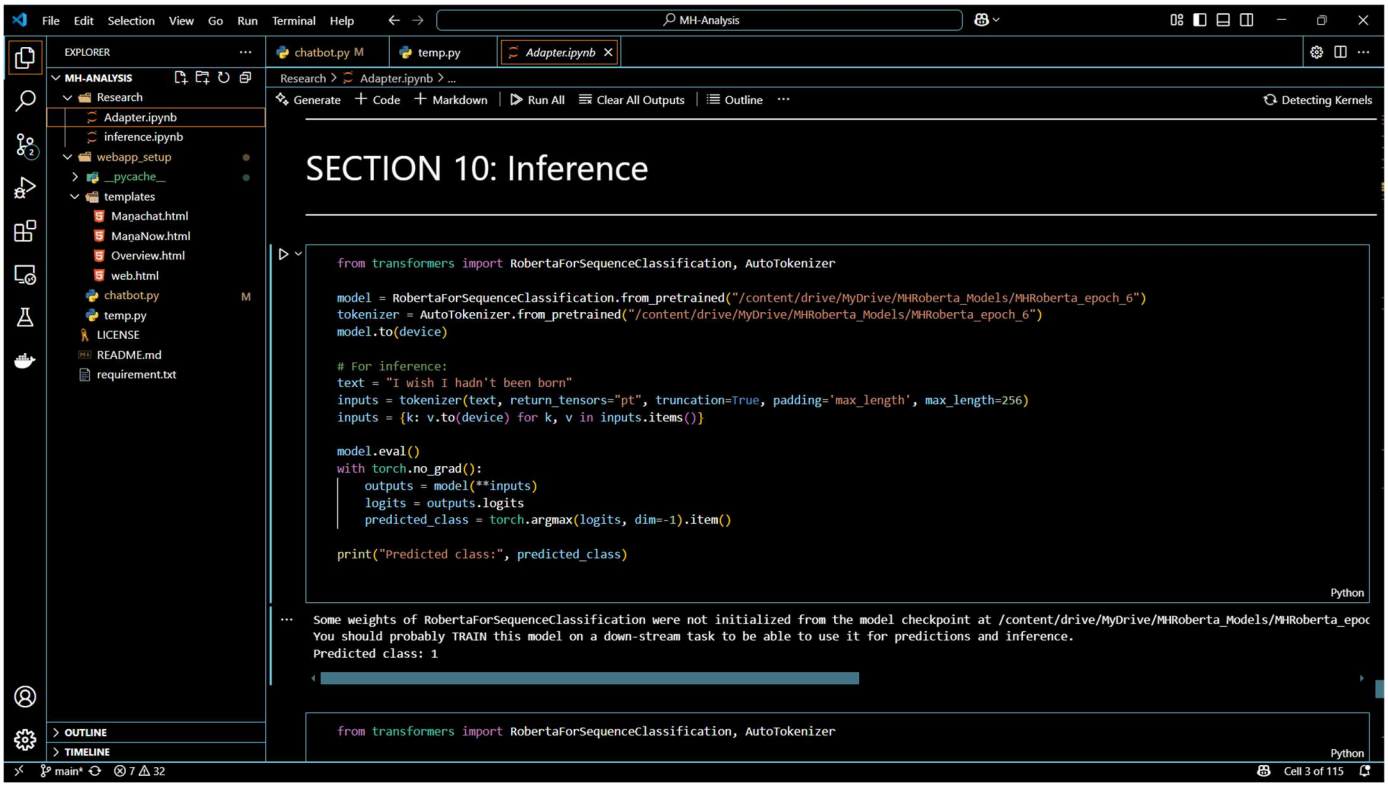
ManaChat.html



MaṉaNow.html

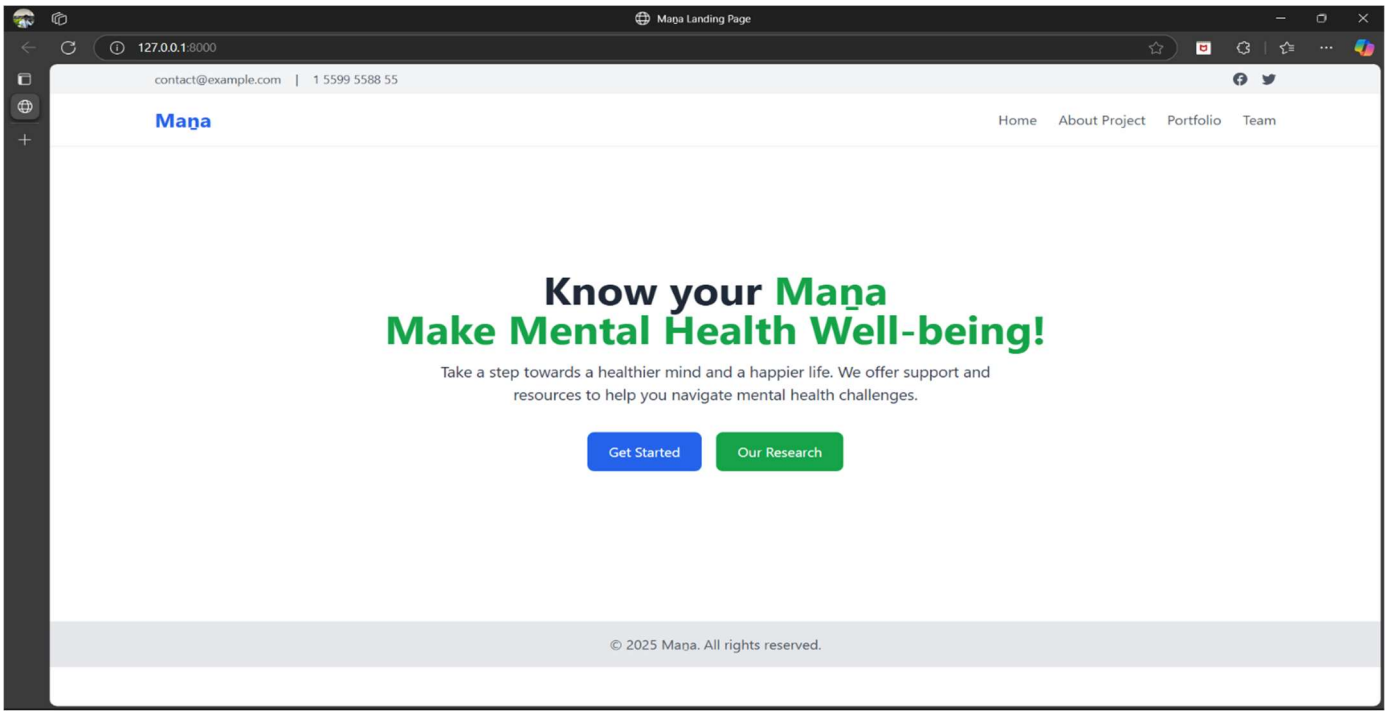


Adapter.ipynb

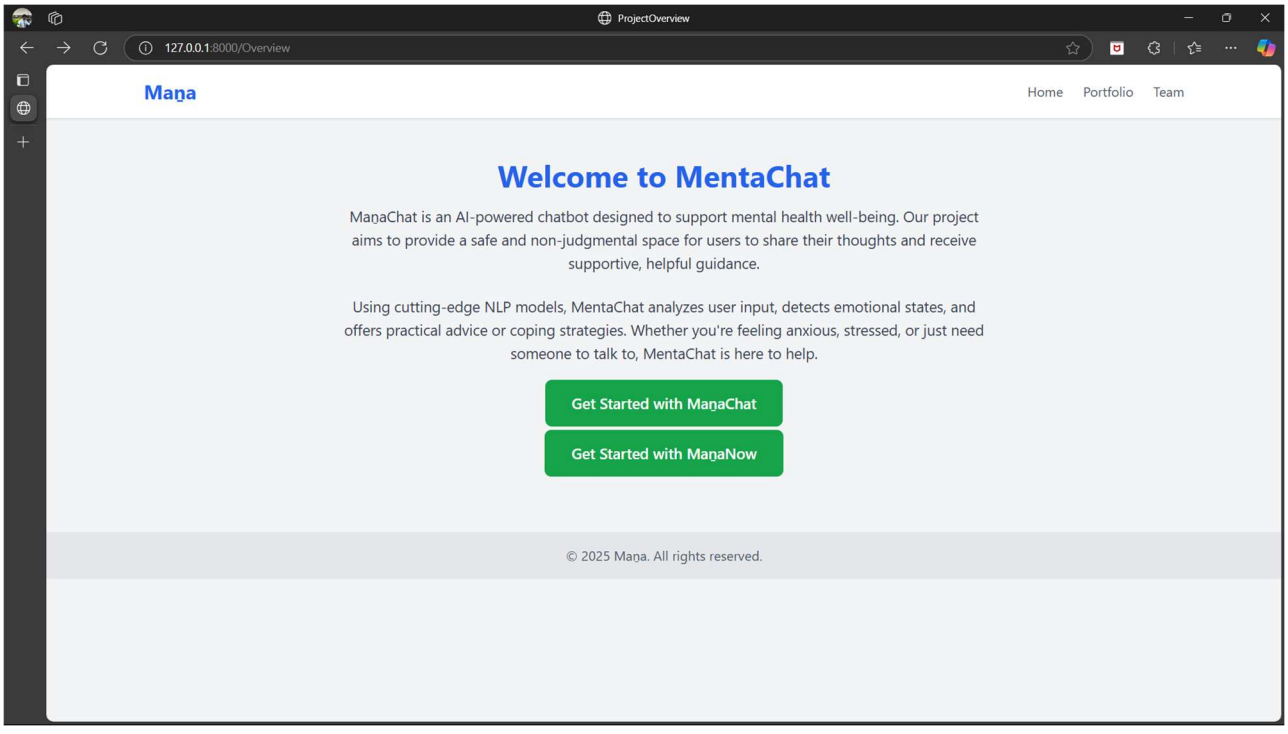


APPENDIX II -SCREENSHOTS

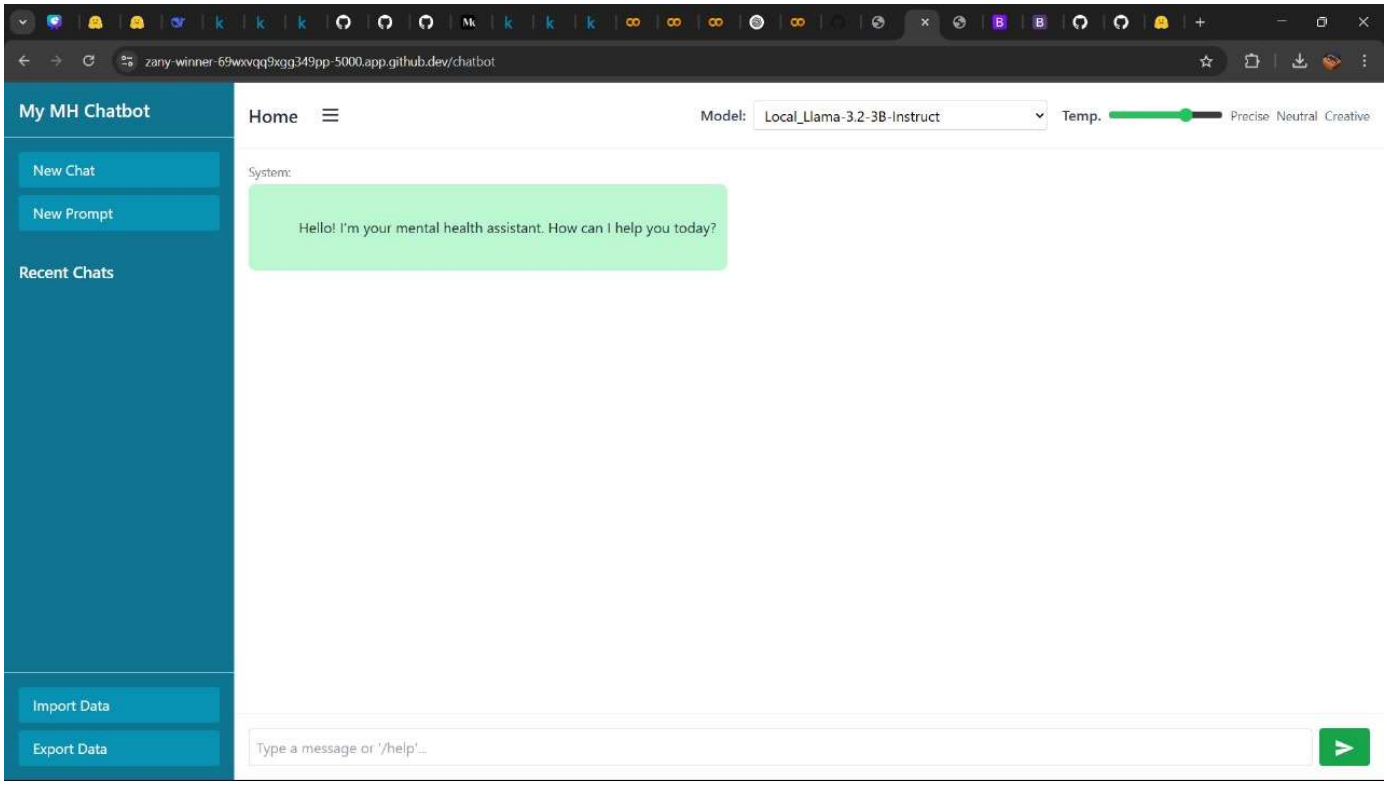
Landing page UI



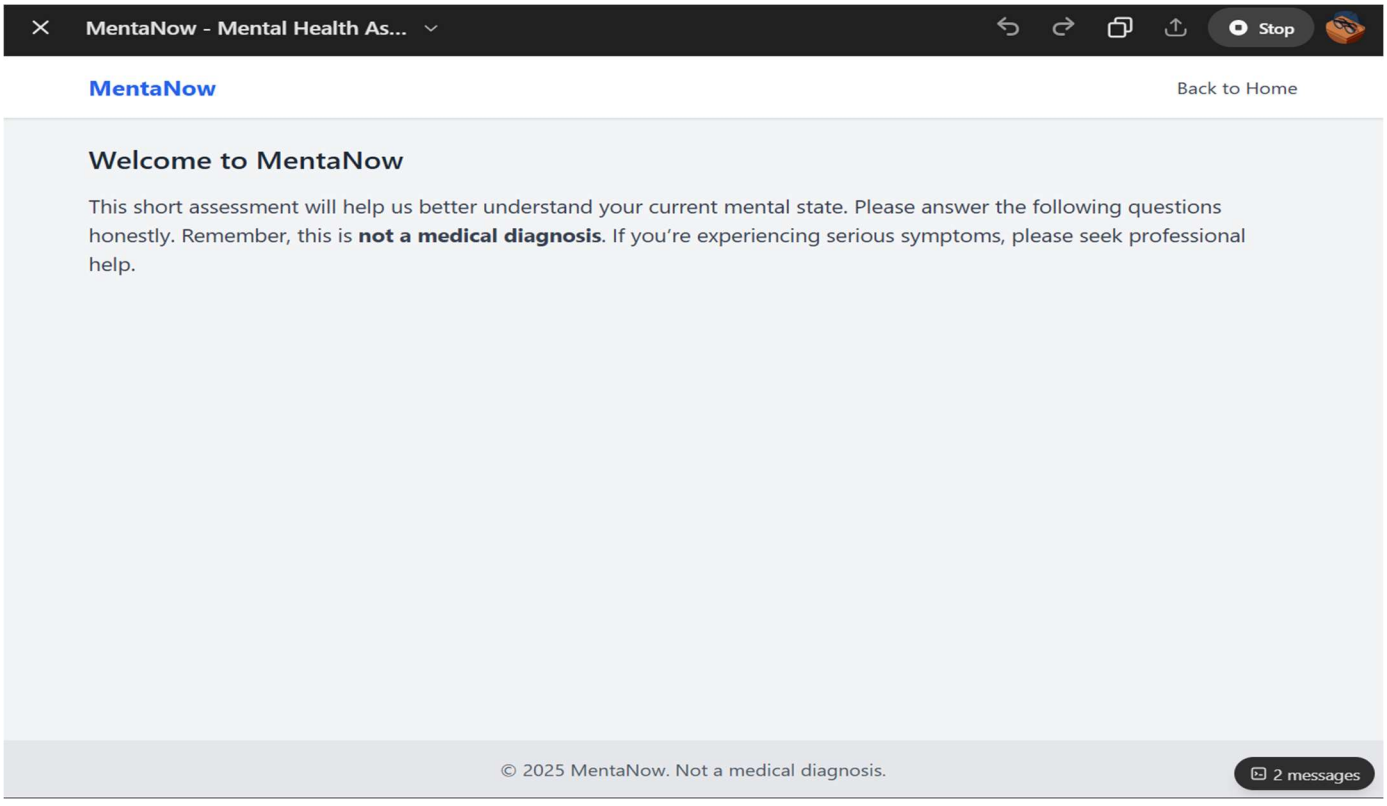
Overview page UI



ManaChat UI



ManaNow UI



Report generation UI