A Major Project Final Report on

Sahara: Mental Health Support Application with Emotion Classification using DistilBERT

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Bachelor of Engineering in IT Engineering
under Pokhara University

Submitted by:
Bikram Kharal, 201511
Raaz Gupta, 201545
Sandesh Khatiwada, 201547
Aayusha Regmi, 201502

Under the supervision of:
Er. Umang Pandey

Date:
24 July 2025

Department of Computer Engineering
NEPAL COLLEGE OF INFORMATION TECHNOLOGY
Balkumari, Lalitpur, Nepal

ACKNOWLEDGEMENT

This project would not have been possible without the joint efforts of the college, the teachers, the supervisor and the faculty members. It has been a pleasure for us to acknowledge the assistance and contributions that were very important and supportive throughout the project.

We are deeply grateful to our project supervisor, Er. Umang Pandey, Assistant Professor of IT Engineering, for his invaluable guidance, technical expertise, and constructive feedback, which significantly elevated the project's quality and scope. His mentorship was pivotal to our success.

We sincerely thank the IT Engineering Department faculty for providing us with adequate resources, guidance and encouragement, which fostered a productive environment. We also acknowledge our institution for nurturing academic excellence and innovation. Additionally, we thank our teachers, parents, and peers for their foundational knowledge, moral support, and constructive feedback, which sustained us throughout the project.

This project stands as a testament to the collective efforts of all those mentioned above. Their guidance, resources, and encouragement have been indispensable, and we are profoundly thankful for their contributions to the success of this endeavor.

ABSTRACT

Mental health is a crucial aspect of overall well-being, yet access to mental health services remains a substantial challenge, particularly in under-resourced and stigmatized communities.To address these barriers, Sahara is a mobile application designed to provide accessible and personalized mental health support.The application is developed using React Native for the frontend and Express.js and Flask for the backend. Utilizing a fine-tuned DistilBERT model, Sahara offers emotion classification through daily journaling, enabling users to track their emotional well-being. Key features include daily journaling with mood tracking, sleep tracking, counselor booking system with video call sessions, a feedback and rating system for counselors, a secure payment gateway via eSewa and a chatbot powered by Gemini. The platform supports three user roles: Users, who can add journal, track sleep, book counselors, and engage with the chatbot; Counselors, who manage availability, accept bookings, and conduct sessions; and Admins, who oversee certified counselor onboarding and monitor platform activities. By integrating AI-driven tools with human-centered care, Sahara aims to revolutionize mental healthcare in Nepal, where such comprehensive platforms are scarce, fostering inclusivity and affordability.

Keywords: Mental health, Emotion classification, DistilBERT, Accessibility, Nepal

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Mental health remains a pressing yet underserved issue in Nepal, where limited access to professional support, social stigma, and geographical barriers prevent many from seeking help. To address this gap, Sahara, an AI-powered mental health app, is designed to provide accessible, and personalized care through daily mood tracking, journaling, and connections to human counselors. By blending technology with human expertise, Sahara aims to democratize mental health support, offering scalable and stigma-free assistance. This project seeks to empower users to manage their well-being while bridging the critical divide between mental health resources and those in need.

1.1 Project Overview

Sahara is a mobile application aimed at addressing the significant mental health crisis in Nepal through the integration of AI technology and human-centered care. The app provides users with personalized mental health support, leveraging tools like mood tracking, journaling with emotion detection, and access to certified human counselors.

Designed with a focus on accessibility and affordability, Sahara caters specifically to the unique challenges in Nepal's mental healthcare landscape—where stigma, geographic isolation, and lack of professional services limit access to timely support.

By combining scalable AI technologies with expert human oversight, Sahara seeks to democratize mental healthcare, and offer a relevant solution for emotional well-being. The project envisions transforming mental health outreach and care delivery, not just for urban populations but also for underserved rural communities across Nepal.

1.2 Problem Statement

Nepal's mental healthcare system faces severe structural shortcomings, with only 0.22 psychiatrists and 0.06 psychologists per 100,000 people. As a result, over 90% of the population lacks access to professional mental health services [1]. The situation is especially critical in rural areas where services are virtually non-existent, and even in urban centers, patients face waitlists exceeding 12 months. Cultural stigma further compounds the crisis, with the majority of individuals avoiding treatment despite evident psychological distress.

Digital interventions have shown global promise, yet existing mental health apps are

largely ineffective in the Nepali context. This systemic failure has led to dire outcomes, including annual rise in suicide rates. To address this urgent public health gap, there is a critical need for an inclusive, and technologically advanced solution. The proposed project, Sahara, aims to develop Nepal's first hybrid AI-human mental health support platform, integrating clinical safeguards with real-time AI assistance to provide private, accessible, and relevant care to those in need.

1.3 Objectives

The project has put forward the following objectives:

- To develop a hybrid AI-human mental health mobile application, integrating a DistilBERT-based emotion classification with a counselor booking system, to provide immediate and accessible mental health support.
- To address Nepal's mental health care gap by combining AI-driven features with affordable access to certified counselors, fostering help-seeking and crisis support in a low-resource, high-stigma environment.

1.4 Significance of the Study

The development of the Sahara represents a groundbreaking advancement in Nepal's mental healthcare system, pioneering an innovative digital solution that bridges critical gaps in accessibility and quality of care. As the nation's first AI-enhanced mental health platform designed specifically for local needs, Sahara transcends traditional barriers by delivering immediate, and personalized support to underserved populations while maintaining clinical standards. This initiative carries transformative potential across multiple dimensions of Nepal's healthcare ecosystem. By leveraging AI technology along with professional care, Sahara provides instant mental health support to communities that currently lack any access to professional services, particularly in remote regions where healthcare infrastructure is scarce or nonexistent. Sahara's unique hybrid model combines the scalability of artificial intelligence with the expertise of human counselors, establishing a sustainable framework for mental healthcare delivery that can adapt to Nepal's resource constraints while ensuring clinical appropriateness. The system generates invaluable insights into mental health trends and treatment outcomes across diverse Nepali demographics, equipping policymakers with evidence to shape more effective public health strategies and resource

allocation decisions. As a replicable model for low-income countries, Sahara contributes to global mental health innovation by demonstrating how technology can expand care access without compromising quality, particularly in settings with severe provider shortages.

1.5 Scope and Limitations

The Sahara project will develop a mental health platform that combines automated emotional support with professional counseling services, delivering a complete digital mental healthcare solution through an accessible mobile application.

The scope of this project includes:

i. Developing intuitive and responsive mobile interface for users to access mental health services, ensuring accessibility across various devices and platforms.

ii. Developing a comprehensive mental health support platform featuring counsellor booking,secure journaling with emotion detection to help users monitor their psychological well-being.

iii. Enabling a hybrid care model that connects users with certified human counselors for scheduled sessions, addressing the limitations of automated support and ensuring personalized care when needed.

The limitations of this project are:

i. While the platform is accessible nationwide to anyone with internet connectivity, service quality and reliability may vary depending on local network infrastructure and stability, particularly in remote regions with intermittent connectivity.

ii. The effectiveness of AI support components depends significantly on users' language proficiency and ability to clearly articulate mental health concerns, which may vary considerably across different age groups, education levels, and regional dialects.

iii. Platform utilization requires basic digital literacy and consistent access to smartphones or computers with modern operating systems, currently excluding population segments without technological access or familiarity, particularly older adults and economically disadvantaged groups.

2. Literature Review

2.1 Review of Existing Systems

In this section, we review existing applications and platforms that provide similar services to those proposed by the Sahara project, focusing on AI-powered mental health support and related features. This review highlights the strengths and limitations of these platforms and identifies the gaps that Sahara aims to address.

Wysa
Wysa is an AI-powered chatbot designed to provide mental health support [2]. It uses evidence based cognitive-behavioral techniques (CBT) to assist users in managing stress, anxiety, and depression. Users can interact with the chatbot for self-help.

Strengths:

i. Offers an intuitive and anonymous platform for mental health support.

ii. Provides CBT-based exercises and mood tracking.

iii. Supports multiple languages, which makes it accessible to a diverse audience.

Limitations:

i. Access to human therapists is a premium feature, which may limit affordability for users in low-resource settings.

ii. Lacks integration of AI-driven tools with human counseling, potentially limiting personalized and comprehensive care.

Woebot
Woebot is another AI-powered mental health chatbot that provides conversational support and evidence-based interventions [3] . It focuses on helping users manage their mental health.

Strengths:

i. Uses natural language processing (NLP) to engage users in meaningful conversations.

ii. Includes psychoeducational content to help users understand their mental health better.

Limitations:

i. Lacks integration with human counselors for personalized support.

ii. Does not provide features like sleep tracking or journaling.

BetterHelp
BetterHelp is an online platform that connects users with licensed therapists through text, audio, and video sessions [4] . It provides user professional mental health support. Strengths:

i. Offers access to licensed therapists for personalized support.

ii. Provides multiple communication channels (text, audio, video).

iii. Includes tools for journaling and self-reflection.

Limitations:

i. An expensive subscription model makes it less accessible to low-income users.

ii. Does not incorporate features like mood tracking or emotion classification.

Sanvello
Sanvello is a mobile app that combines self-help tools, peer support, and access to licensed therapists [5] . It offers mood tracking, guided meditation, and cognitive behavioral therapy exercises.

Strengths:

i. Comprehensive suite of features, including mood tracking and guided meditation.

ii. Offers an interactive community feature for peer support and stress management tools.

Limitations:

i. Premium features require a subscription, which may not be affordable for all users.

ii. Does not integrate with local mental health resources or services.

Headspace
Headspace is a mindfulness and meditation app that helps users manage stress and anxiety [6]. While it is not specifically a mental health support app, it offers tools that contribute to overall mental well-being.

Strengths:

i. Focuses on mindfulness and meditation to improve mental health.

ii. Provides structured programs for stress and anxiety management.

iii. Engages users with a user-friendly interface and gamified experience.

Limitations:

i. Lacks features like AI-driven support or access to human counselors.

ii. Primarily targets users seeking mindfulness, not those in need of clinical mental health support.

2.2 Comparison with Existing Systems

From the review of these existing platforms, several gaps are identified that Sahara aims to address:

Affordability and Accessibility: Many platforms rely on subscription models or premium features that limit access for low-income users. Sahara aims to provide low-cost services to ensure accessibility for underserved communities.

Integration of Human and AI Support: While some platforms offer either AI-driven or human centered support, Sahara will combine both, allowing users to interact with AI services and access human counselors when needed.

Mood tracking and Journaling: Existing platforms lack features such as journaling with mood tracking. Sahara will include a journaling feature to allow users to add their daily journal and detect emotion using fine-tuned DistilBERT model.

Comprehensive Feature Integration: Unlike existing systems, Sahara provides a

robust set of features, including sleep tracking, a feedback and rating system for counselors to ensure quality, and a role-based platform with User, Counselor, and Admin functionalities. The Admin dashboard enables certified counselor onboarding and platform oversight, while Users and Counselors benefit from tools like appointment tracking and availability management, creating a holistic and user-centric mental health ecosystem.

3. Methodology

This section describes the methodology that have been followed during the development of the project.

3.1 Software Development Life Cycle

For the software development process of our project, we have adopted the Incremental Model, which structures the system's development over multiple progressive iterations. Each iteration includes the phases of Requirement Analysis, Design, and Implementation, facilitating early delivery of foundational features and gradual system enhancement.

In the initial iteration, our primary focus was on implementing critical functionalities such as journaling feature with emotion classification, sleep tracking, and counsellor onboarding feature in the admin's end, establishing the core framework of the application.

Subsequent iterations progressively introduced advanced features like counselor booking system,rating and feedback functionality, integration of payment gateway, development of the functionality for real time video session between user and counselor etc. The following subsection outlines how the incremental SDLC model was applied throughout the development of the system.

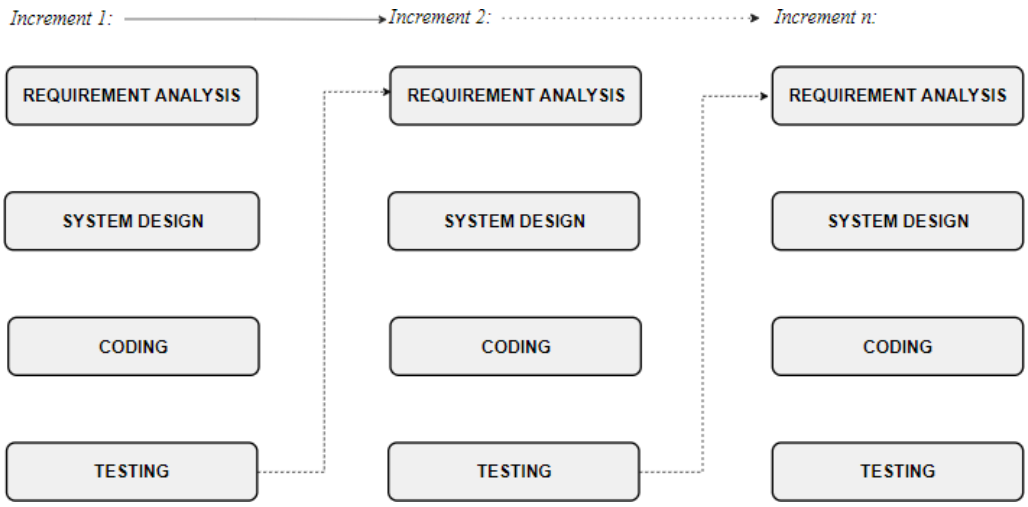

Figure 1: Incremental Model of Software Engineering

3.1.1 Requirement Analysis

In the Requirement Analysis phase, we focused on gathering and documenting all the necessary features and expectations for the app. This includes identifying the core features like counselor booking system, sleep tracking and journaling with emotion classification, secure payment gateway, and feedback with rating functionality supporting both Users and Counselors. The outcome of this phase is a comprehensive System Requirement Specification (SRS) document that outlines what the app must achieve and serves as a blueprint for the development process.

3.1.2 System Design

Transitioning from requirement analysis, the Design phase focuses on visualizing the app's structure and user experience. We utilized tools like Figma to create interface designs. The app's internal workings, including the APIs that connect the mobile frontend to the backend, are carefully planned. Additionally, we designed the database schemas and UML diagrams to ensure data is handled efficiently. This step helps us align the design with the functional requirements before development begins.

3.1.3 Coding

In the Coding phase, we started translating the designs and requirements into functional code. The web application for admin dashboard was built using ReactJS, while

the mobile app was developed with React Native. Backend APIs for the core functionalities were developed using Express.js and AI features were exposed as a Flask API. Mongoose was used to interact with the MongoDB database. Each feature was developed step by step, forming the foundation of the operational core of the project.

3.1.4 Testing and Evaluation

After the development phase, each feature of the application is carefully tested during the Testing and Evaluation stage. This involves verifying the correct functionality of each component, checking the app's behavior with real-world inputs, and identifying and fixing any bugs or errors. These steps ensure that the app remains stable, performs reliably, and provides a smooth user experience.

3.2 Technologies Used

Table 1 consists of the major technologies that are proposed to be used during development and deployment of the application. They are briefly described in the subsections that follow.

Table 1: Technologies Used

Subject	**Tools and Technologies Used**
Backend Database	MongoDB
REST API Service	Node and Express JS; Postman
AI Model Service	Python and Flask API
UI/UX Design	Figma
Android / iOS application	React Native
IDE / Code Editor	Vscode
Version Control System	GitHub
DistilBERT Documentation	LaTeX

3.2.1 MongoDB and Mongoose

MongoDB is a NoSQL database that stores data in a flexible, JSON-like format. Mongoose is an ODM (Object Data Modeling) library that makes it easier to define and work with MongoDB data in our project.

3.2.2 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to use JavaScript to write server-side code, enabling the development of scalable and high-performance web applications. Express.js is a web application framework for Node.js that simplifies the process of building APIs and web applications. The combination of Node.js and Express.js provides a powerful platform for developing RESTful APIs and handling HTTP requests.

3.2.3 Express.js

Express.js, commonly referred to as Express, is a minimal and flexible web application framework for Node.js. Express.js is being used to develop a REST API for the system.

3.2.4 Python and Flask

The AI models to be used in the platform is being exposed as a Flask API. Flask is a lightweight web framework for Python. Python is a high-level, versatile programming language used in web development, automation, data science, etc.

3.2.5 Postman

Postman is an application software that is used for testing API web services. It lets users provide URL, parameters, body and headers to send the request and also shows the response in raw as well as pretty form.

3.2.6 React Native

React Native is an open-source framework used to build mobile apps for both Android and iOS using a single codebase. It allows us to create fast, responsive, and native-like apps using JavaScript

3.2.7 Vscode

Visual Studio Code (VSCode) is a lightweight, cross-platform code editor developed by Microsoft. It is highly popular among developers due to its extensive customization options and rich ecosystem of extensions. This editor was chosen for backend development because of our familiarity with it and its powerful features.

3.2.8 Figma

Figma is a web-based UI/UX design application that allows for collaborative design and prototyping. It was chosen for designing the user interface of the mobile application due to its ease of use and real-time collaboration features.

3.2.9 Git and GitHub

Git is a distributed version control system that is being used in this project to manage different versions of the project and to collaborate with each other. GitHub is a platform that uses git for version control. We are using GitHub to host the project repository, to track changes, and manage different versions of the project.

3.2.10 LaTeX

LaTeX is widely used documentation preparation system for preparation of scientific documents, books and technical papers. It uses plain text for formatting unlike other document creation systems. The source code is compiled by a compeller to generate the printable/viewable document. The reason for using LaTeX for documentation was to learn this new form of documentation.

3.3 Overview and Details of Technologies

This section describes the working principles of the two AI components in the Sahara app: the emotion classifier model and AI chatbot. The emotion classifier is trained on publicly available datasets to classify emotions in journal entries, while the chatbot leverages pre-trained models without additional training.

3.3.1 Emotion Classifier

The emotion classifier classifies the emotional tone of journal entries to provide insights for counsellors. It was built on DistilBERT, a lightweight transformer model that processes English text for emotion classification. DistilBERT uses a distilled version of the BERT architecture, it tokenizes journal entries into word embeddings, passes them through transformer layers to capture contextual relationships, and outputs probabilities for emotional tones.

Working Principle

The process involves:

1. Text Preprocessing: Journal entries are cleaned (remove special characters, normalize case), and tokenized using DistilBERT's tokenizer.
2. Feature Extraction: DistilBERT's encoder generates contextual embeddings, capturing semantic and emotional nuances of the text.
3. Classification: A fully connected layer with softmax activation classifies the text into emotions.
4. Output Storage: The predicted emotion is stored in MongoDB and can be shared with the counsellor.

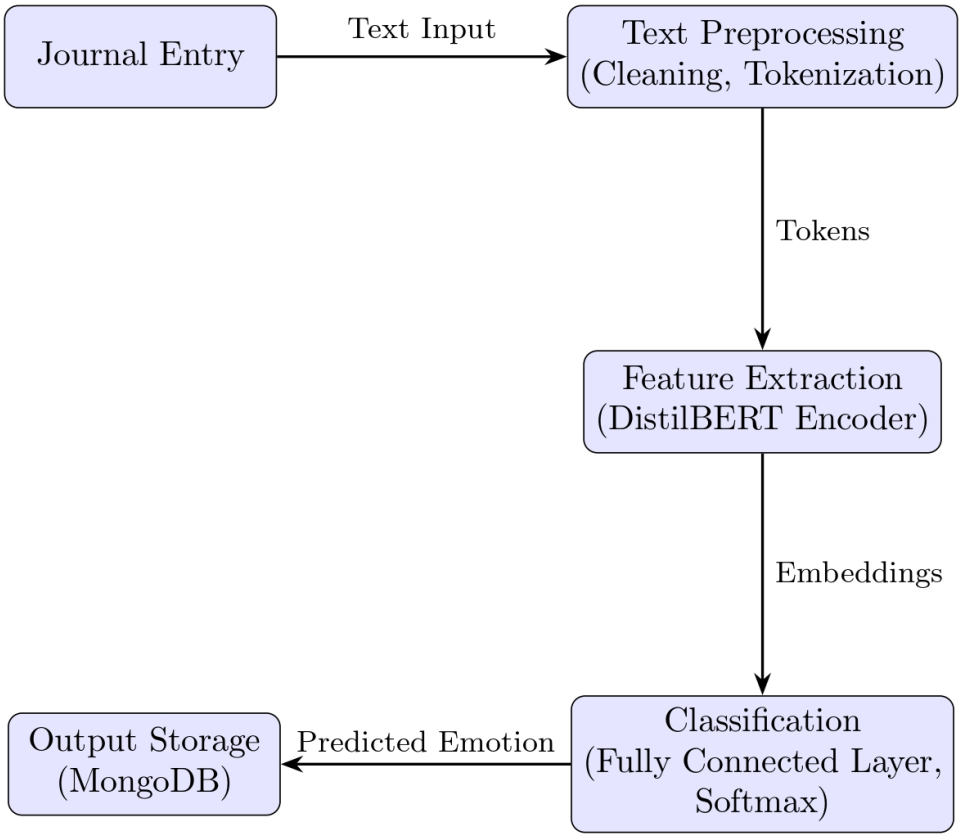

Figure 2: Working of Fine Tuned DistilBERT for Emotion Classification

The model is trained on public datasets to ensure reproducibility and ethical data use.

Data Sources

To enable the emotion classification feature in the Sahara mental health support application, a publicly available dataset was utilized to train the DistillBERT model, ensuring accurate detection of emotional states from user journal entries.

Emotion Dataset for Emotion Recognition Tasks

This dataset, comprising tweets annotated with six distinct emotions (anger, fear, joy, love, sadness, and surprise) , was used to fine-tune the DistillBERT model for emotion classification. The dataset's text-based nature and emotional annotations made it suitable for training the model to analyze user-generated journal entries, a core feature of Sahara's personalized mental health support system. The fine-tuned DistilBERT model processes journal inputs to identify emotional patterns, enabling users to gain insights into their mental well-being and receive tailored support through the application's AI-driven and human-centered features.

Source: https://www.kaggle.com/datasets/parulpandey/emotion-dataset/data

The dataset was preprocessed to align with the project's requirements, including tokenization and formatting for compatibility with DistilBERT's input structure.

The training process involved fine-tuning the model on the annotated tweet data to optimize its performance for detecting emotions relevant to mental health contexts, thereby enhancing the application's ability to provide meaningful feedback to users in Nepal's low-resource, high-stigma environment.

Data Distribution

Total Samples: Training: 15,969 unique samples, Validation: 1998 unique samples, Test: 2000 unique samples

Emotion Classes and Distribution (Training Set):

- Joy: 5362 instances
- Sadness: 4666 instances
- Anger: 2159 instances
- Fear: 1937 instances
- Love: 1304 instances
- Surprise: 572 instances

Training and Evaluation

The model was fine-tuned using the Hugging Face transformers library. A total of 3 epochs were used for training.

Performance Metrics

The first figure below shows overall metrics for training, validation and test respectively, second figure shows per-class metrics and third figure shows the confusion matrix.

Overall Performance Metrics:

To assess the general effectiveness of the DistilBERT model for emotion classification, we computed overall performance metrics — accuracy, precision, recall, and F1-score — on the training, validation, and test datasets.

i. **Accuracy:** The percentage of total correct predictions over all predictions.

   Accuracy = Number of Correct Predictions / Total Predictions

ii. **Precision:** The average proportion of correct positive predictions (i.e., how many predicted emotions were actually correct).

   Precision = True Positives/ (True Positives + False Positives)

iii. **Recall:** The average proportion of actual class instances that were correctly predicted.

   Recall = True Positives/ (True Positives + False Negatives)

iv. **F1-Score:** The harmonic mean of precision and recall, representing the overall balance between the two

   F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

The model achieves high accuracy across all splits, indicating strong overall classification ability.

The training set shows excellent performance (96% accuracy and 0.94 F1-score), suggesting that the model has effectively learned to distinguish between emotional categories. On the validation and test sets, the performance remains robust (F1-scores of 0.918 and 0.888, respectively), which implies that the model generalizes well and does not overfit excessively. The slight drop in metrics on the test set (compared to train/validation) is expected and indicates real-world complexity in unseen data.

Split	Accuracy	Precision	Recall	F1-Score
Train	0.9617	0.9228	0.9646	0.9411
Validation	0.9400	0.9044	0.9340	0.9181
Test	0.9275	0.8706	0.9115	0.8884

Figure 3: Overall Performance Metrics

Per-Class Performance Metrics:

To evaluate the model's performance across each emotion category, we calculated the precision, recall, and F1-score for every class. These metrics help us understand how

well the model performs on each specific emotion, especially in the presence of class imbalance.

The model performs exceptionally well for sadness, joy, and anger, with F1-scores well above 0.95. For love and surprise, the model achieves high recall but lower precision — indicating it often detects these emotions, but sometimes confuses them with other classes (false positives). The relatively lower precision for 'love' and 'surprise' (0.824 and 0.804) suggests some semantic overlap with similar emotions (like joy or fear), which is also visible in the confusion matrix.

Emotion	Precision	Recall	F1-Score
Sadness	0.9965	0.9801	0.9882
Joy	0.9968	0.9407	0.9680
Love	0.8248	0.9893	0.8996
Anger	0.9646	0.9838	0.9741
Fear	0.9500	0.9324	0.9411
Surprise	0.8041	0.9615	0.8758

Figure 4: Per-Class Performance Metrics

Confusion Matrix:

A confusion matrix is a performance measurement tool for classification problems. It provides insight into not just how often a model is correct, but how it makes its errors. Each row of the matrix represents the true label, while each column represents the predicted label. This helps us see where the model confuses one emotion for another.

The model performs very well on major classes like joy (641 correctly classified) and sadness (556 correct). Love and anger are also predicted quite accurately with only a few misclassifications.

Some confusion exists between joy and love, likely due to semantic similarities in text. Surprise, being a minority class, suffers from misclassification and fewer correct predictions.

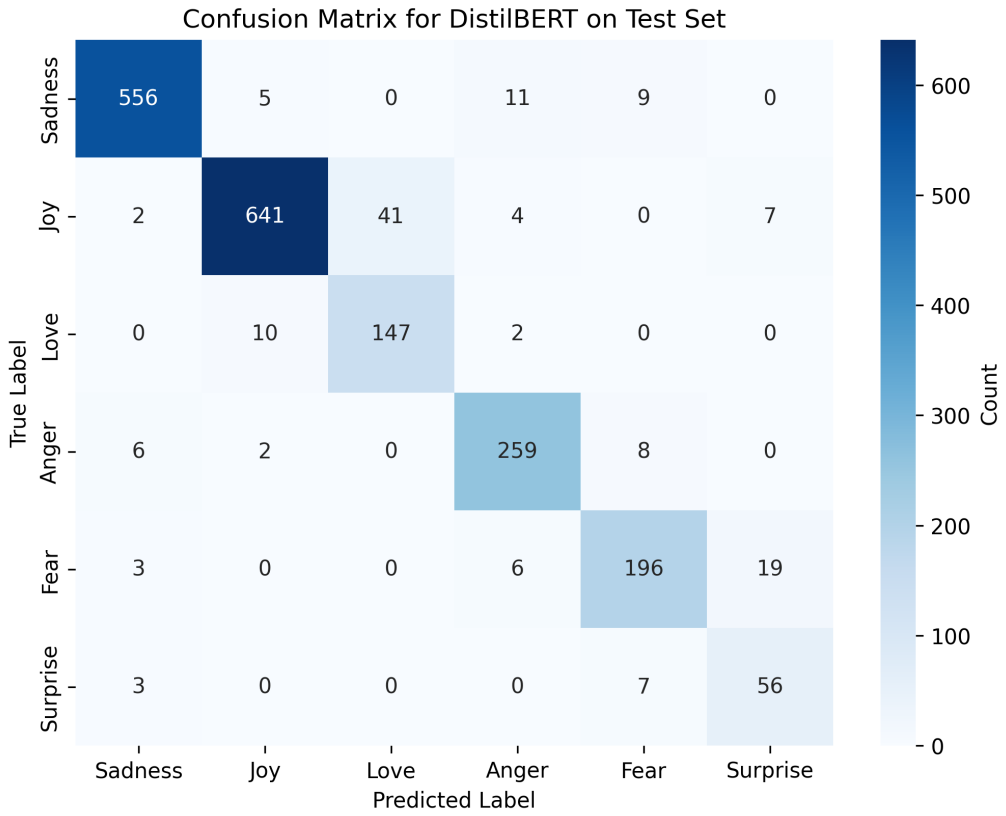

Figure 5: Confusion Matrix on Test Set

3.3.2 AI Chatbot

The AI chatbot in the Sahara application functions as a virtual counselor, providing empathetic and context-aware responses to user queries. It leverages Google's Gemini API, which provides access to a pre-trained generative AI model, eliminating the need for custom training.

Working Principle

The chatbot utilizes the Gemini API to generate coherent and empathetic responses for open-domain dialogue. The process comprises the following steps:

1. **Input Submission:** User messages are sent as text prompts to the Express backend which then provides it the Flask API which in turn provides it to Gemini API via an HTTP request.

2. **Response Generation:** The Gemini API processes the input prompt and generates a text response, leveraging the model's pre-trained capabilities to produce contextually relevant dialogue.

3. **Output Delivery:** The generated response is returned to the Flask backend which in turn provides it to the Express.js server which processes it further and then delivers it to the React Native mobile application, ensuring integration with the user interface.

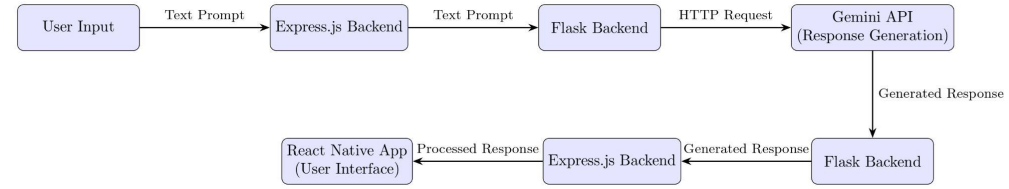

Figure 6: AI Chatbot Workflow and Integration with the System

4. System Design

This section provides a comprehensive overview of the proposed system's architecture focusing on key diagrams: Use Case Diagram, Class Diagram and Schema Diagram. These diagrams illustrate the core components and the interactions between them on the system.

4.1 Use Case Diagram

A use case diagram is a visual representation of the interactions between actors (In our case: User, Counsellor and Admin) and a system under consideration. It shows the different use cases or functionalities provided by the system and the relationships between the actors and the use cases.

Users can register in the platform, and then login with the provided credentials. After logging in, users can log daily sleep patterns, add daily journals, add booking requests for the appointment with the counselor, make payments, participate in video sessions with the counselor, and provide feedback with rating to the counselor.

Counselor can log in with the credentials provided him through the mail by the Admin, add availability, view booking requests, accept/reject them, track appointment bookings and participate in the video sessions.

Admin can add counselors, view registered counselors and users as well as get the overview of the platform.

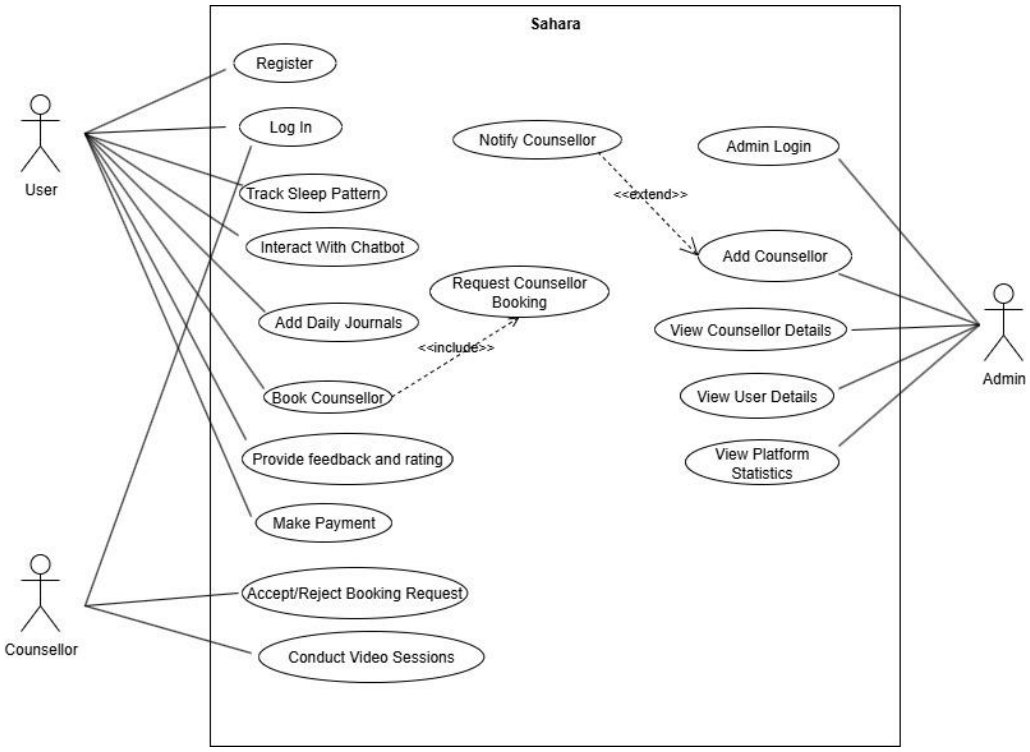

Figure 7: Use Case Diagram

4.2 Schema Design

The schema diagram below illustrates the data structure of the Sahara mental health app. It provides the relationships between entities that support user interactions, counseling sessions, journal entries, sleep tracking, chatbot conversations, and administrative oversight.

It represents the database organization for storing and managing data handled by the system.

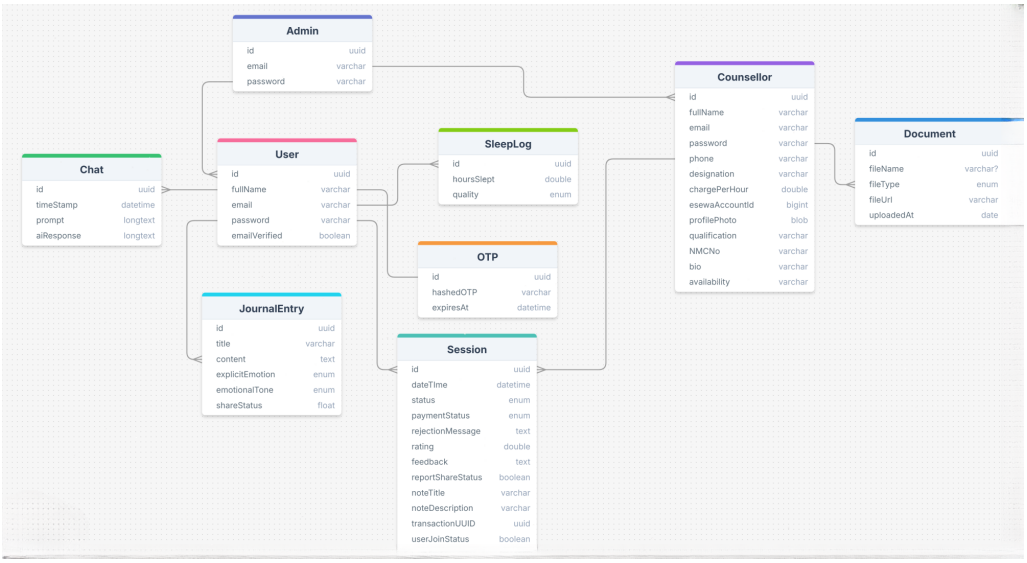

Figure 8: Schema Design

4.3 Class Diagram

The class diagram below represents the data model of the Sahara mental health app. It captures entities and their relationships for users, counselors, sessions, journal entries, sleep logs, chatbot interactions, and administrative management.

Users can have multiple sessions with the Counsellor, can provide multiple JournalEntires, SleepLogs. Chat Message represents Chat of User with AI Chatbot.

The Sahara mental health app's data model is depicted in the class diagram below. For users, counselors, sessions, journal entries, sleep logs, chatbot interactions, and administrative oversight, it records entities and their relationships.

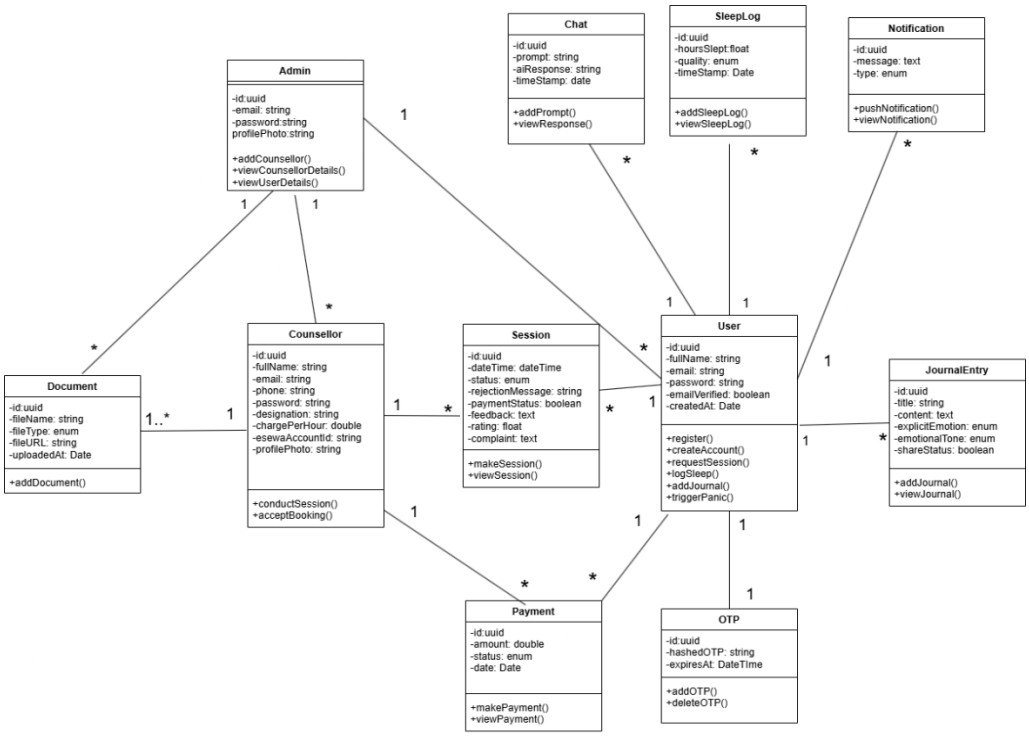

Figure 9: Class Diagram

5. Result and Discussion

The Sahara- Mental Health Support Application was developed as a comprehensive platform aimed at bridging the gap between individuals seeking mental health assistance and professional counselors. The application integrates multiple functionalities spanning emotional wellness tracking, counselor engagement, and AI-based emotion classification to provide personalized support and insights.

5.1 Implemented Features

The Key features successfully implemented in the Sahara application include:

- **Counselor Appointment Booking and Video Sessions:** Users can browse available counselors and request appointments based on the counselor's availability. Upon confirmation, users and counselors can conduct real-time video sessions for one-on-one counseling.
- **Counselor Availability and Session Management:** Counselors can manage their availability by selecting time slots, view incoming booking requests, and accept or reject them accordingly.
- **eSewa Payment Gateway Integration:** Users are required to complete payment prior to session commencement. We integrated the eSewa payment system, enabling secure transaction handling for booking confirmations.
- **Post-Session Feedback and Rating:** Users can rate counselors and provide feedback after each session, which helps maintain quality and transparency within the platform.
- **Emotion-Aware Daily Journaling:** Users can maintain daily journals, which are automatically analyzed using the fine-tuned DistilBERT model. The emotion classifier identifies the emotional tone of each journal entry across different emotion classes: anger, fear, joy, surprise, love, and sadness, or neutral.
- **Sleep Tracking and Emotional Insights:** Users can log their daily sleep patterns and track them over time. Both emotional trends and sleep patterns can be shared with counselors to support a more holistic and informed counseling experience.
- **Admin Dashboard:** The admin interface allows for management of platform participants (users and counselors), counselor onboarding, and provides an overall overview of platform usage and statistics.

5.2 Challenges Encountered

During development and integration, we faced several technical and logistical challenges:

- **Real-time Video Call and Booking Flow Synchronization:** Coordinating session statuses and access control between users and counselors involved managing state consistency across frontend and backend systems.
- **Payment Integration with eSewa:** The payment workflow required integrating with eSewa's sandbox and handling edge cases like OTP expiration and payment confirmation delays.
- **Emotion and Sleep Data Sharing Privacy:** Implementing privacy-aware consent mechanisms to allow users to control what information is shared with counselors added complexity but was essential for ethical design.
- **Multi-role System Architecture:** Designing and separating user, counselor, and admin interfaces while maintaining a unified backend API was crucial for scalability and maintainability.

6. Project Task and Time Schedule

The project schedule for the Sahara AI-Powered Mental Health Support App was developed by considering the technical complexity and integration of multiple features. The time allocation ensures foundational components like the AI chatbot and user data handling are prioritized to support the other modules.

Table 2: Project Task Duration

TASK	**APPROX DURATION IN DAYS**
Requirement Analysis	14
System Design	4
AI Chatbot & Mood Tracker Development	17
Application (Web & Mobile) Development	26
Frontend & Backend Integration	10
Testing & Debugging	8
Documentation	28

6.1 Gantt Chart

The Gantt chart in Figure 10 and Figure 11 shows the time schedule of the project. The chart shows the tasks and their respective time duration in the project. The tasks are shown in the vertical axis, and the time duration is shown in the horizontal axis. The tasks are color coded to show the different phases of the project. The chart shows the start and end date of each task, and the duration of each task is shown in days. The chart also shows the dependencies between the tasks, which helps in understanding the flow of the project.

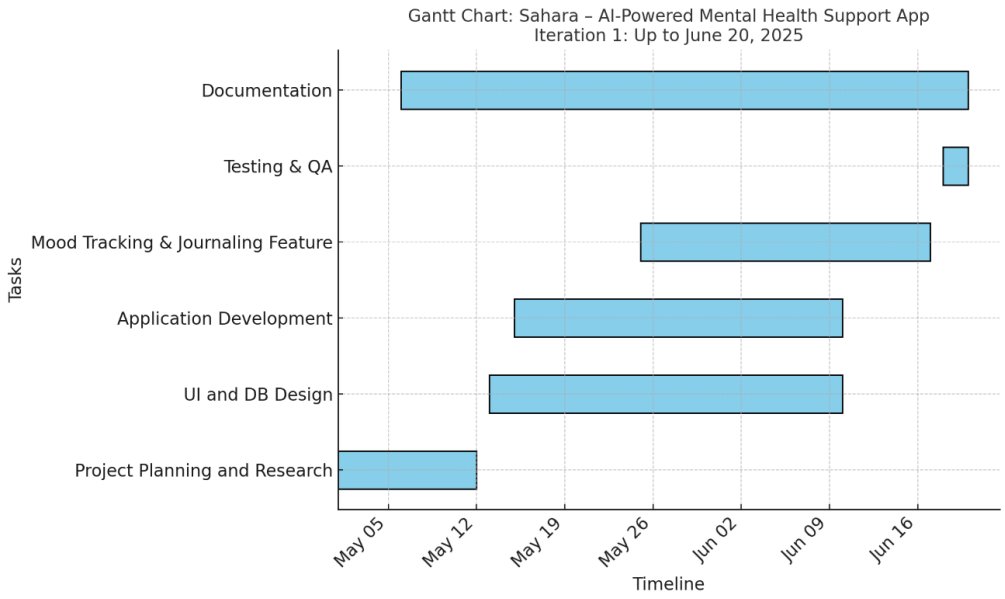

Figure 10: Gantt Chart Iteration I

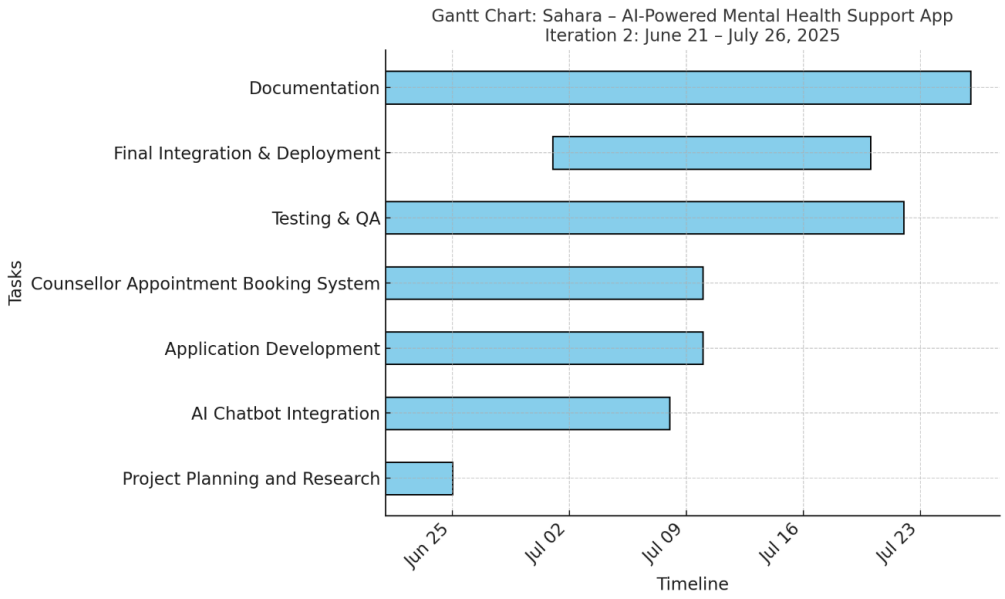

Figure 11: Gantt Chart Iteration II

7. Conclusion

The Sahara project effectively illustrates how artificial intelligence can be integrated with practical mental health support services. The platform seeks to bridge the gap between people seeking mental wellness and professional counseling services by incorporating counselor booking and engagement systems along with emotion-aware journaling tools.

Over time, users and counselors can get important emotional insights from automatic sentiment analysis of journal entries made possible by the DistilBERT-based emotion classification model.

The system architecture — built using React Native for mobile, React.js for the admin dashboard, Express.js for backend services, and Flask for AI model integration — highlights the effective use of modern technologies to solve meaningful problems. The application successfully underwent performance evaluation, with promising accuracy, precision, recall, and F1-scores in the emotion classification task, proving the viability of applying NLP techniques in mental health applications.

Overall, Sahara lays the foundation for AI-assisted mental health platforms in the Nepali context and beyond, and it opens up several possibilities for future improvements and scalability.

7.1 Further Work and Recommendations

While the current implementation of Sahara provides essential mental health support through AI-powered chat, mood tracking, and access to counselors, there are several areas where future improvements can be made:

- **Multilingual Support:** TTo reach a broader audience across Nepal, especially in rural regions, the app should include support for Nepali and other local languages. This would improve accessibility and user engagement.
- **Enhanced AI Capabilities:** The current emotion classification model can be further advanced by leveraging more sophisticated Natural Language Processing (NLP) architectures. This would enable the system to provide not only more accurate emotion detection but also offer context-aware, empathetic, and therapeutic responses, making the AI support more human-like and effective in addressing users' emotional needs.

- **Integration with Wearables:** Future versions could integrate with smartwatches or fitness trackers to gather physiological data (e.g., heart rate, sleep patterns) and provide better mental health insights.
- **Data Privacy and Security:** Further research into advanced encryption techniques and data anonymization methods should be conducted to strengthen user privacy and trust.
- **Community Forums and Peer Support:** Creating moderated community spaces for users to connect with peers facing similar challenges can enhance social support and engagement.
- **Collaboration with Mental Health Institutions:** Partnering with NGOs, hospitals, and educational institutions could increase credibility, reach, and impact of the app.

In conclusion, while Sahara lays a strong foundation for accessible mental health support in Nepal, ongoing research, development, and stakeholder collaboration are essential to scale its impact sustainably.

References

[1] N. P. Luitel, M. J. D. Jordans, A. Adhikari, N. Upadhaya, C. Hanlon, C. Lund, and I. H. Komproe, "Mental health care in nepal: Current situation and challenges for development of a district mental health care plan," *Conflict and Health*, vol. 9, no. 1, p. Article 3, 2015. [Online]. Available: https://doi.org/10.1186/s13031-014-0030-5

[2] Wysa, (n.d.), "Wysa: Ai mental health support," https://www.wysa.com/.

[3] Woebot Health, (n.d.), "Woebot: Your ai friend for mental health," https://woebothealth.com/.

[4] BetterHelp, (n.d.), "Betterhelp: Online therapy and counseling services," https://www.betterhelp.com/.

[5] Sanvello, (n.d.), "Sanvello: Your personal mental health app," https://www.sanvello.com/.

[6] Headspace, (n.d.), "Headspace: Meditation and mindfulness for mental wellness," https://www.headspace.com/.

8. Appendix

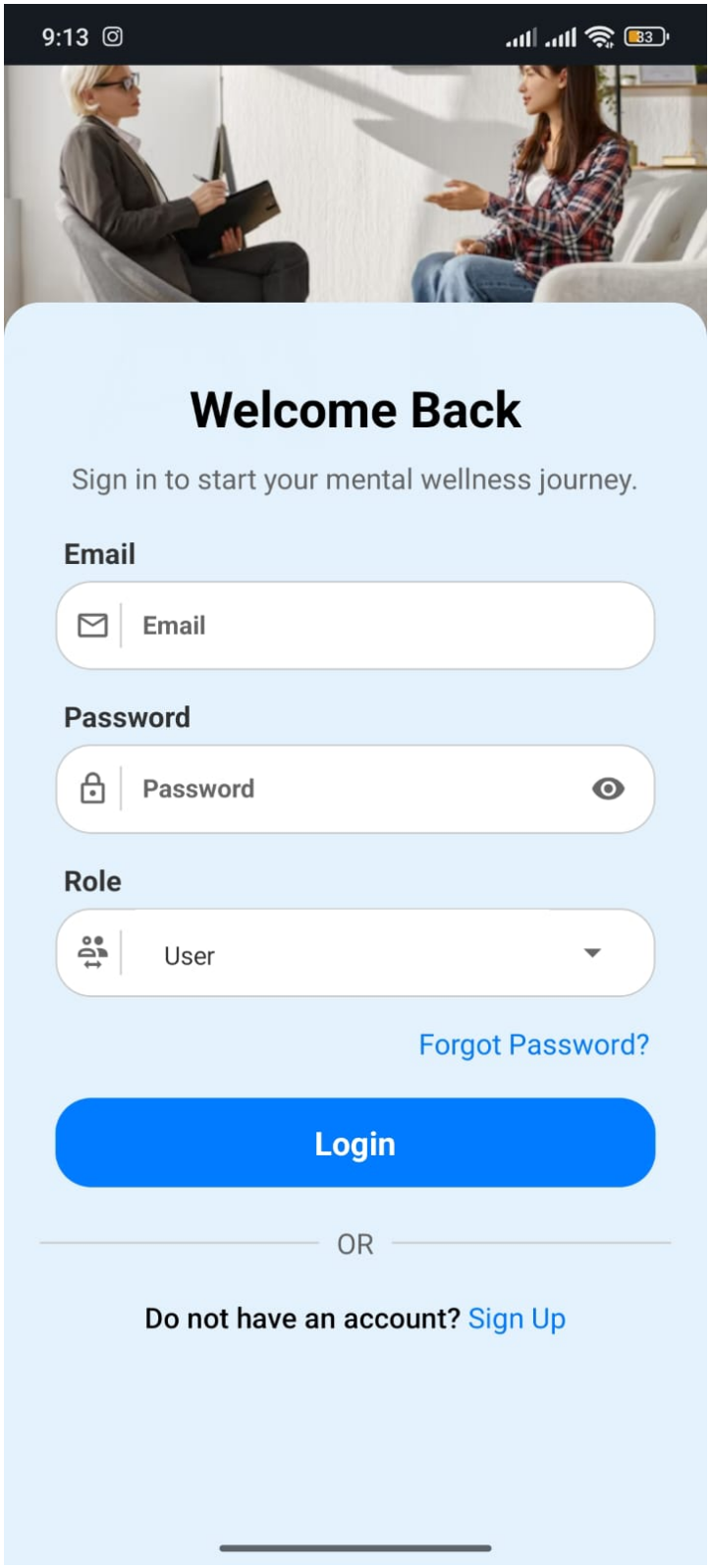

Figure 12: Login Page

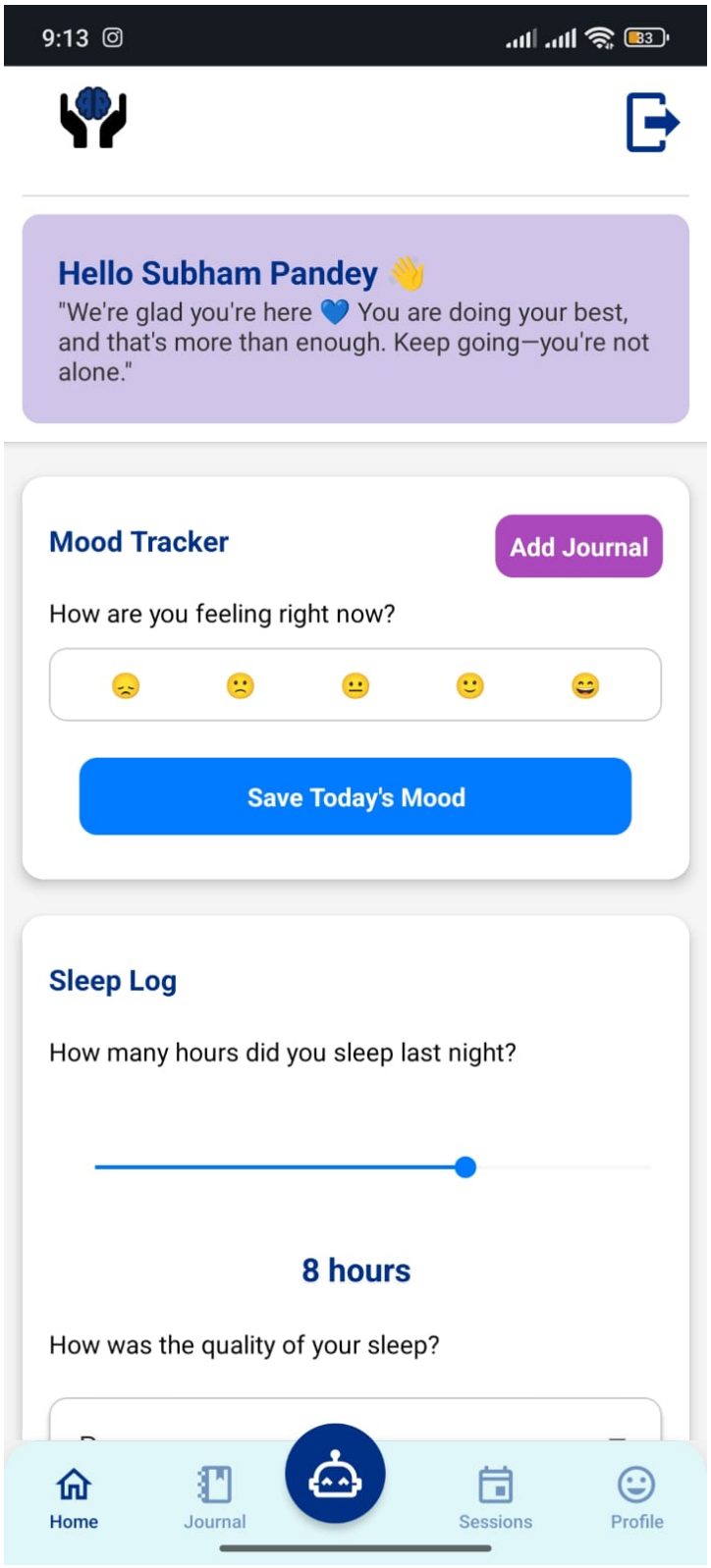

Figure 13: User Dashboard

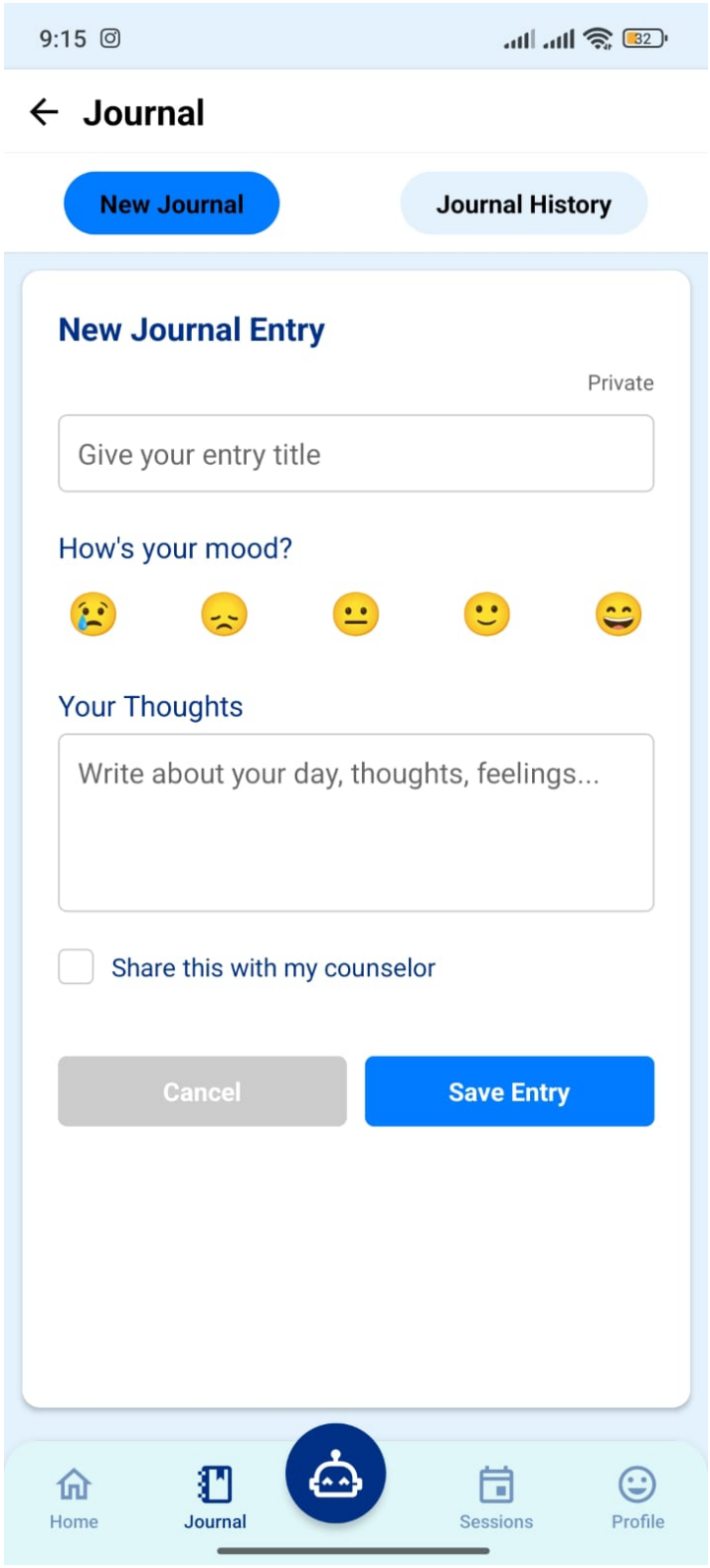

Figure 14: Journal Page

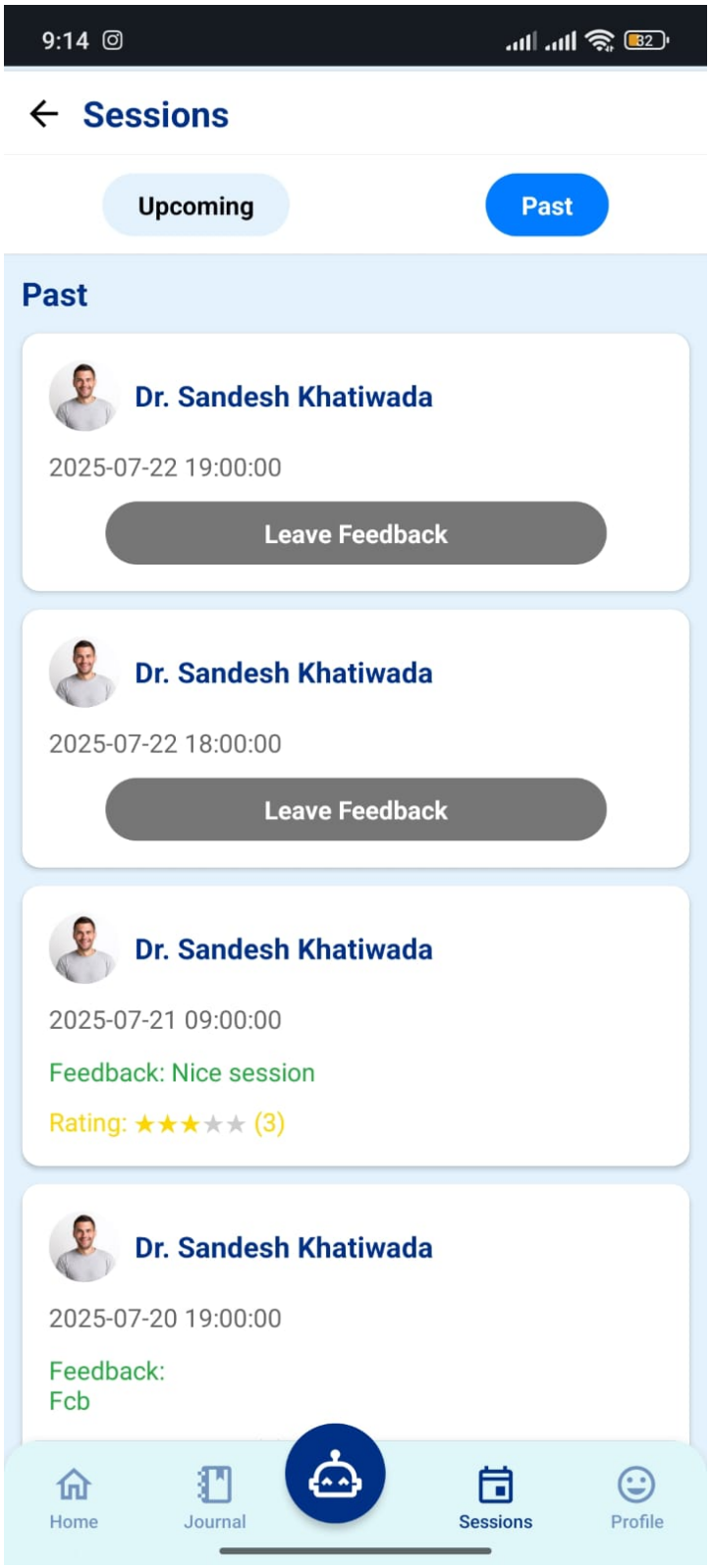

Figure 15: Session Page

Counsellor Side UI

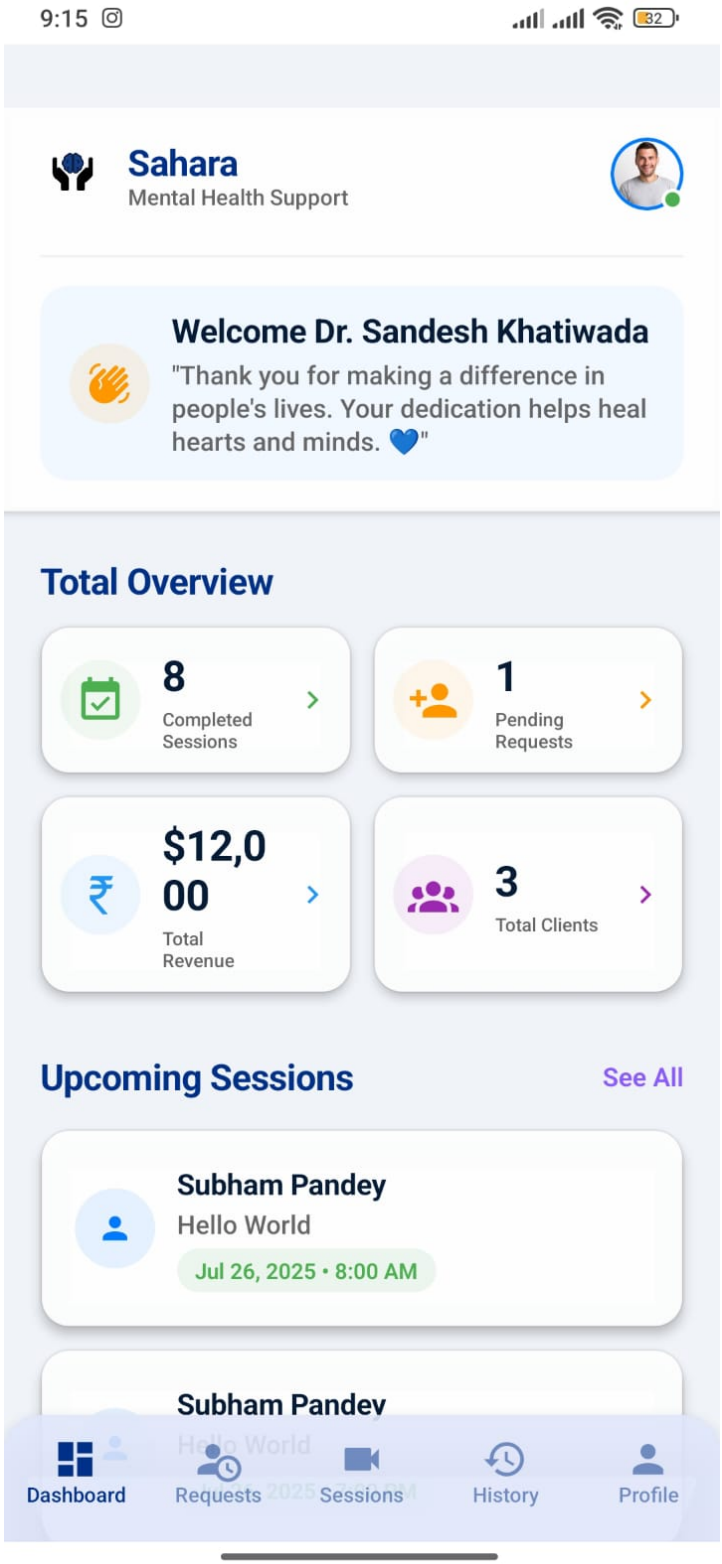

Figure 16: Counsellor Dashboard

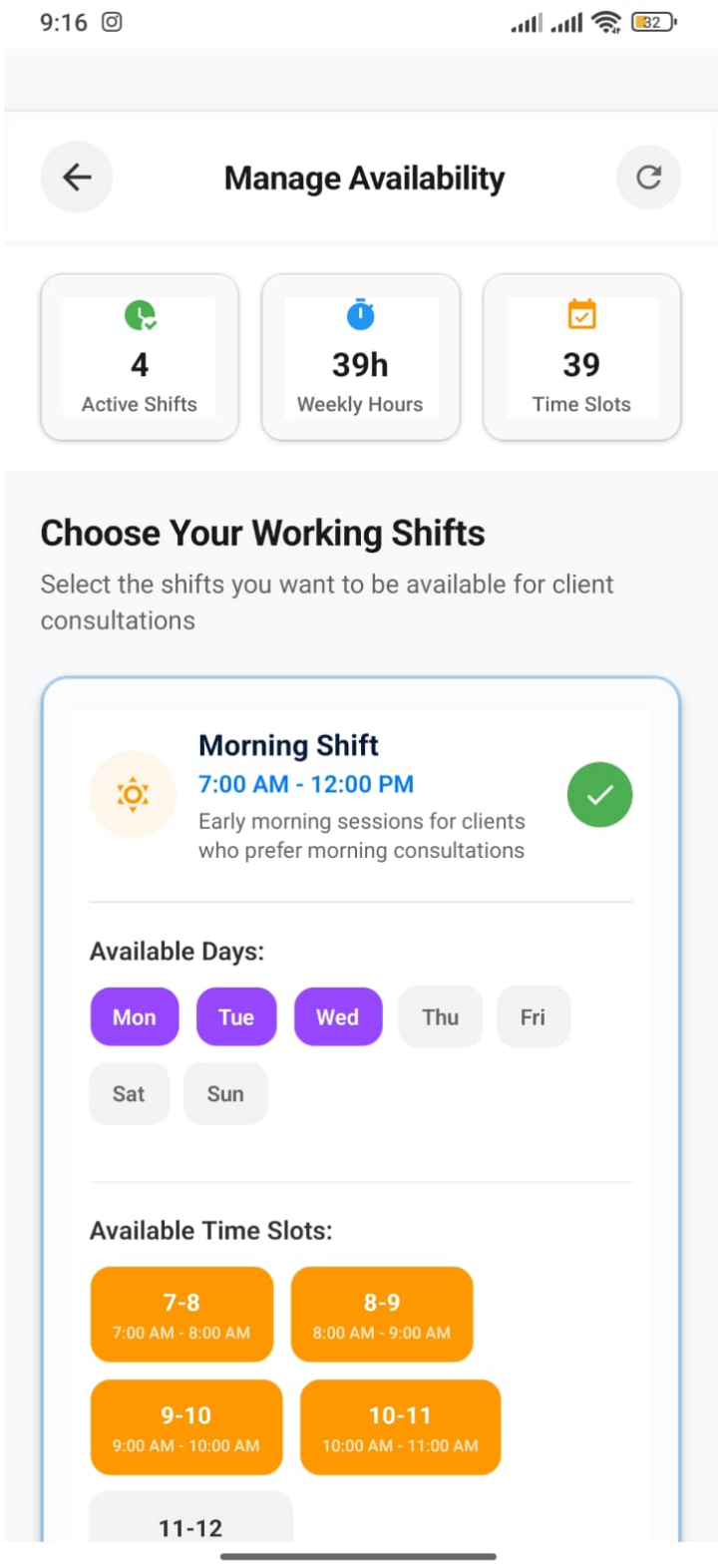

Figure 17: Counsellor Availability

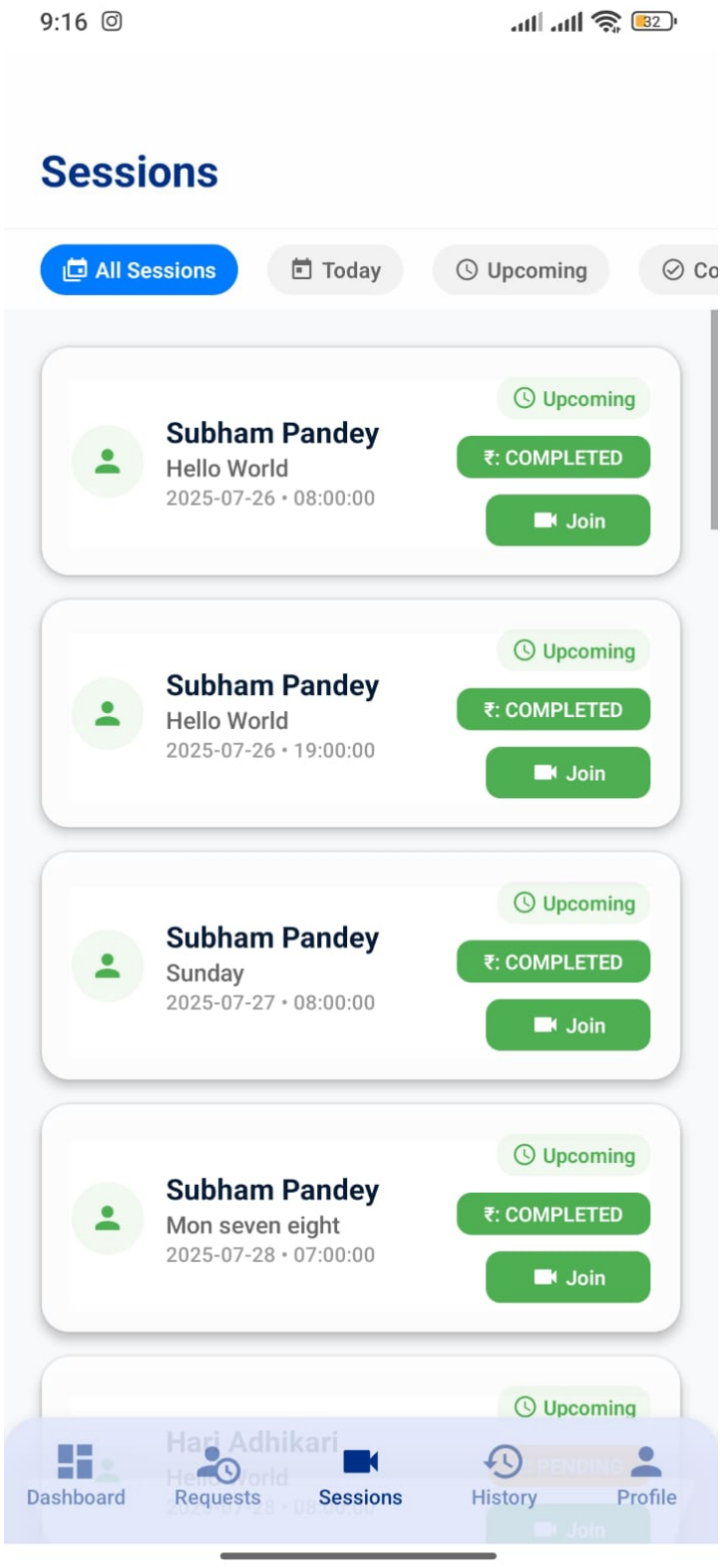

Figure 18: Counsellor Session

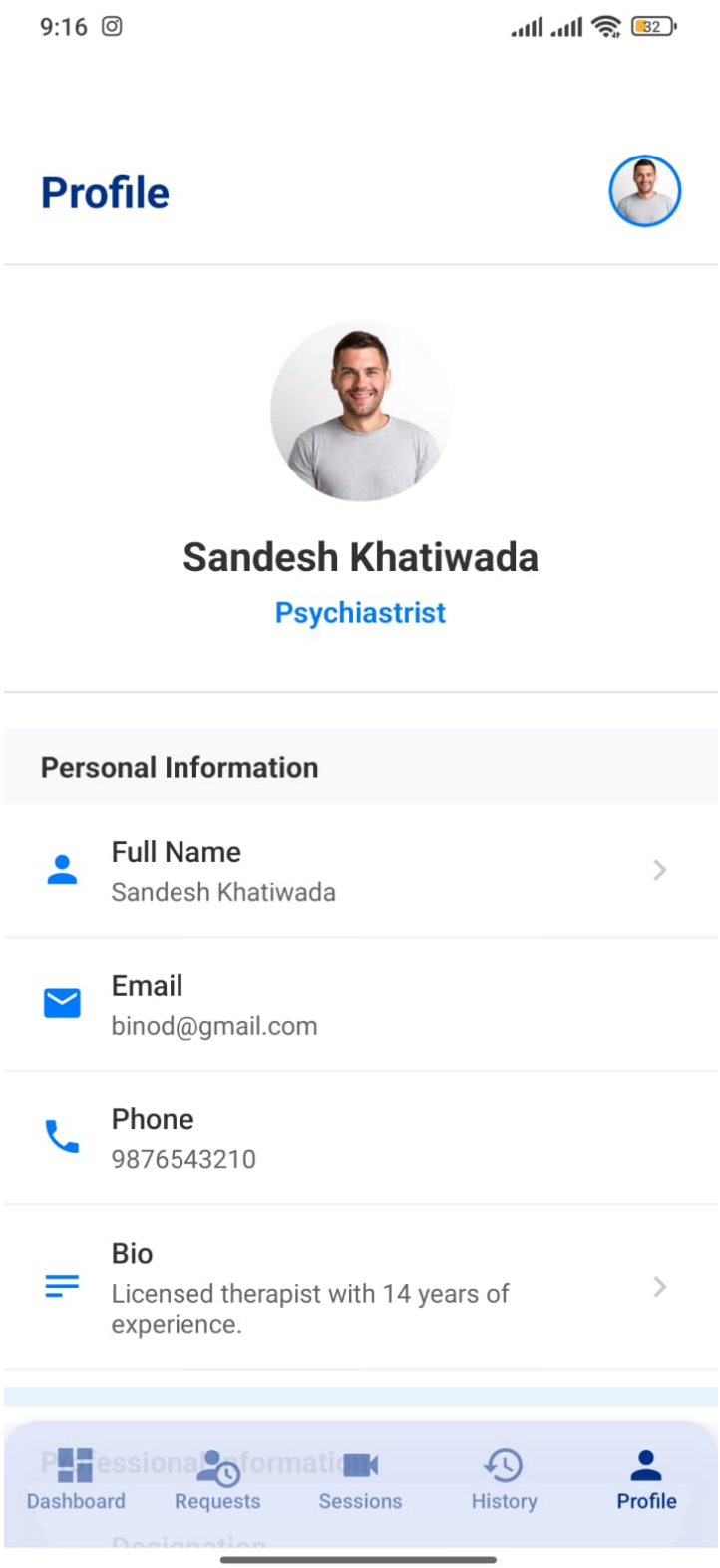

Figure 19: Profile Page

Admin Dashboard UI

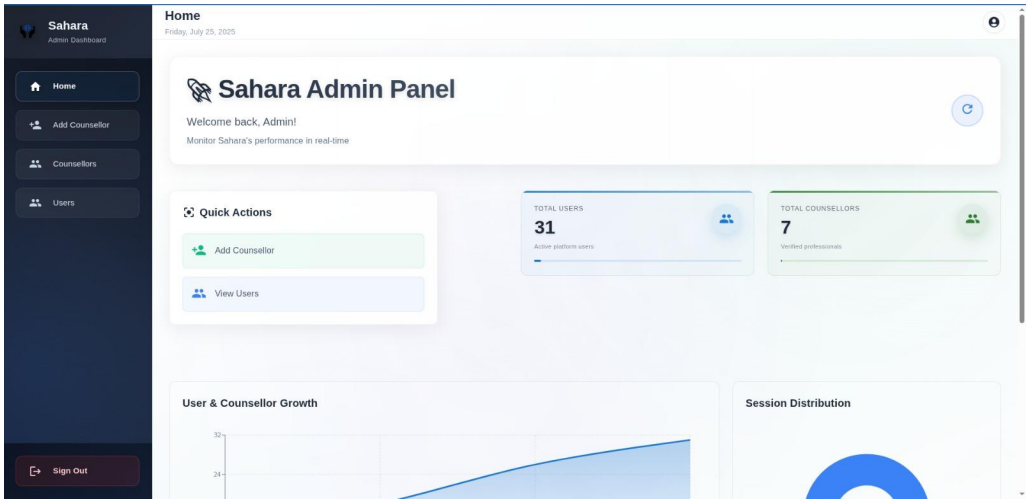

Figure 20: Home Page

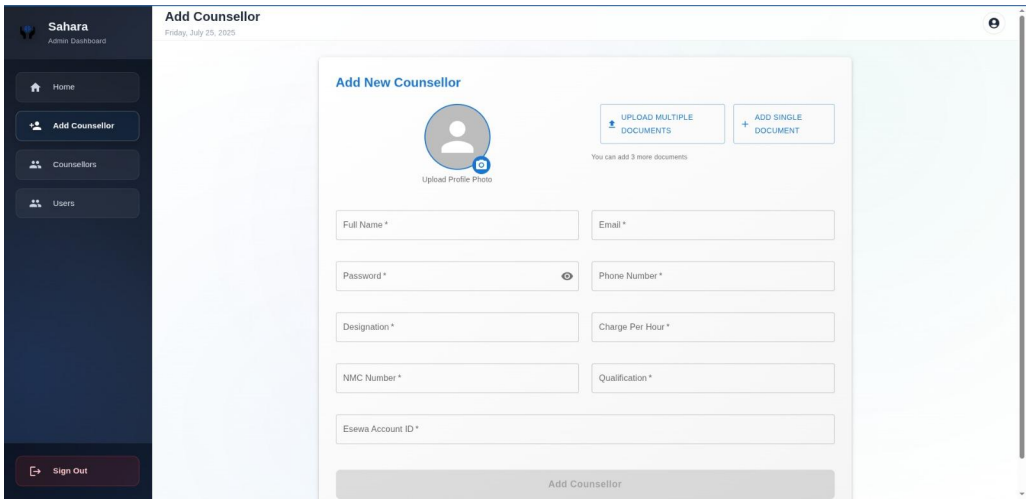

Figure 21: Add Counsellor

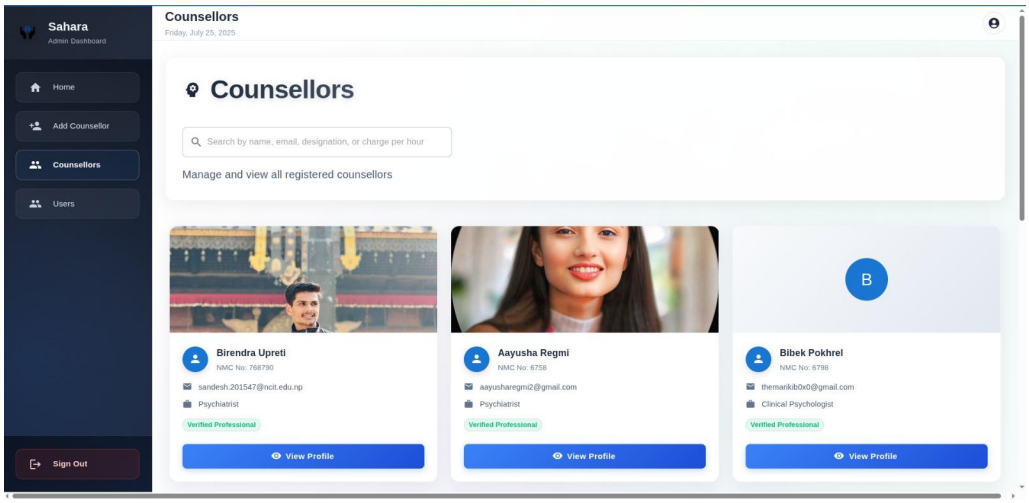

Figure 22: Browse Counsellor

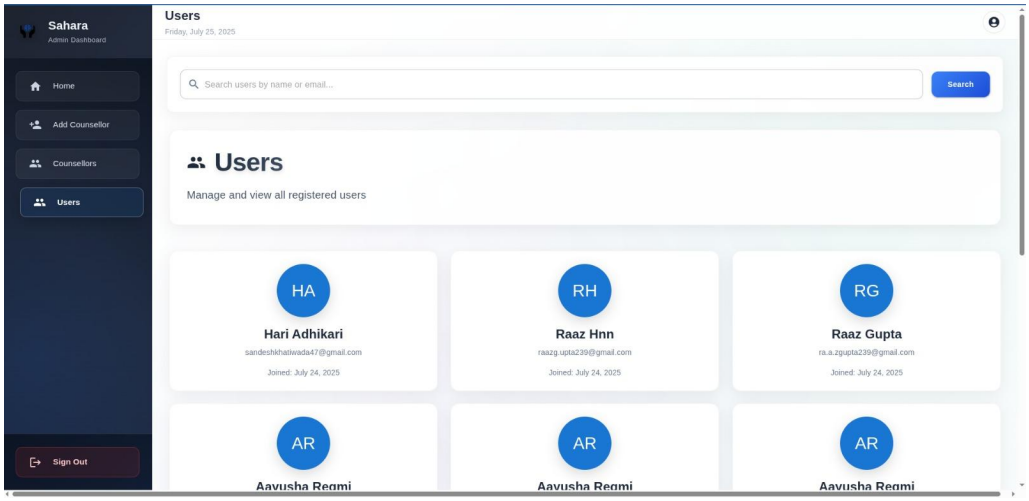

Figure 23: Browse Users