THERABOTICS: A MULTI-TASK AI-DRIVEN DIGITAL PSYCHOLOGICAL INTERVENTION SYSTEM WITH CALIBRATED RISK MODELING AND PRIORITY-BASED COUNSELOR SCHEDULING

Integrating Multi-Task Inference, Clinical Risk Calibration, and Anonymous Priority Scheduling for University Mental Health Support

[1]Omaima, [2]Safoora Saher, [3]Sania Mehwish Fatima

[123]Final Year Student, B.E. Artificial Intelligence and Data Science

Department of Artificial Intelligence and Data Science

Stanley College of Engineering and Technology for Women, Hyderabad, India

Abstract : The deterioration of mental health among university students has reached a significant level worldwide, with research indicating that nearly 20% of students face clinically relevant symptoms of depression or anxiety. Although there is a broad range of digital health resources available, current systems predominantly operate as standalone conversational agents, lacking integration with institutional processes, continuous monitoring of risk, or smart routing to counselors. This paper introduces Therabotics, an anonymous, web-based mental health intervention platform created specifically for college settings. The platform features an innovative Multi-Task Inference Engine that concurrently conducts six psychological analysis tasks: emotion classification, estimation of emotion intensity, risk probability assessment, intent detection, evaluation of model confidence, and generation of therapeutic responses, all in a single call to a large language model. A custom sigmoid-based risk calibration layer refines raw probabilistic outputs by incorporating emotion intensity and longitudinal trend signals, producing clinically meaningful severity scores. Three validated clinical instruments, namely the PHQ-9, GAD-7, and GHQ-12, are natively embedded within the conversational interface, enabling passive and active psychological assessment without requiring formal clinical intake. A min-heap priority queue algorithm ensures that students with the highest clinical severity receive counselor appointments first, while a custom ten-dimensional GloVe embedding model clusters anonymous peer forum posts into five emotionally coherent support communities. A confidence-gated crisis detection mechanism prevents false-positive emergency escalations arising from ambiguous natural language inputs. Deployed on Vercel and Render with a Supabase PostgreSQL backend, Therabotics demonstrates that seamless integration of calibrated multi-task AI inference, priority-driven scheduling, and full anonymity preservation constitutes a technically feasible and clinically responsible approach to digital mental health intervention in academic institutions.

Index Terms *— Digital Mental Health, Multi-Task Learning, Risk Calibration, Conversational AI, Priority Queue Scheduling, University Mental Health, Anonymous Intervention Systems, Large Language Models, GloVe Embeddings, Clinical Assessment Automation.*

I. INTRODUCTION

1.1 Background of University Mental Health Crisis

The mental health of university students has become one of the most urgent public health issues of this decade. Academic settings, marked by significant performance pressure, social upheaval, financial difficulties, and diminished familial closeness, foster situations that are especially favorable for the emergence and intensification of psychological suffering. The World Health Organization has repeatedly recognized depression and anxiety as significant factors contributing to disability among young adults aged 18 to 25, a cohort that mostly coincides with the global university population [7].

The problem is particularly severe in the Indian context. Garg, Agrawal, and Arya [8] found that a large number of Indian university students express clinically significant levels of psychological discomfort, but less than one in ten of them ask for professional care.

Cultural stigma regarding mental illness, apprehension about academic repercussions, and insufficient awareness of accessible support options together inhibit help-seeking behavior. Where university counseling services do exist, they are often too busy, with student-to-counselor ratios that are much higher than what is suggested.

Jackson, Lewis, and Wang [1] conducted a significant systematic review and meta-analysis published in the Journal of Medical Internet Research in 2025. It looked at the effectiveness of AI-driven conversational agents in 47 independent research with more than 12,000 young adults. The review found that AI-based digital therapies led to statistically significant decreases in self-reported symptoms of depression and anxiety, with effect sizes varying from minor to moderate. The authors critically recognized that the most effective systems possessed three architectural characteristics: continuity of monitoring, personalization of response, and systematic interaction with professional care routes. This discovery directly influences Therabotics' design philosophy.

1.2 Limitations of Existing AI Mental Health Chatbots

Even while there is more and more proof that digital mental health tools work, most of the systems that are already in use have basic design problems that make them less useful in clinical settings. Woebot and Wysa are two commercial systems that work as standalone conversational bots based on the concepts of cognitive behavioral therapy (CBT). These technologies make it easy for people to talk to each other without fear of being judged, but they work alone. They don't do structured risk evaluations, send priority signals to human counselors, or keep long-term psychological profiles of its users.

Single-task AI models that classify one mental health indication, like depression, from text don't take into account how psychological illnesses can happen together and often overlap. A student suffering from burnout may concurrently display anxiety, stress, and first depressive symptoms—conditions that a depression-exclusive classifier would mischaracterize or completely overlook. Also, most of the systems that are already in use use untested, proprietary scoring methods instead of clinically proven tools like the PHQ-9 or GAD-7. This makes them less acceptable in institutional healthcare settings. Jackson et al. [1] also said in their meta-analysis that the AI chatbot studies that are already out there have limited follow-up periods, little institutional integration, and no mechanisms for escalating crises. These shortcomings are not just theoretical; they pose genuine clinical hazards in situations when a student experiencing a psychiatric crisis may engage with a system unable to identify or adequately respond to that crisis.

1.3 Research Gap

A thorough review of the current literature indicates that no presently implemented system concurrently fulfills all of the following criteria: multi-task psychological inference from natural language, calibrated risk probability output, seamless integration of validated clinical assessment tools, priority-based anonymous counselor scheduling, longitudinal mood and wellness monitoring, peer community support with intelligent topic clustering, and a confidence-gated safety mechanism to avert false-positive crisis escalations.Without this integrated approach, the care experience is broken up. Students might use one app for journaling, another for clinical assessment, and rely only on manual methods to make appointments with counselors. There is no shared data layer that connects these experiences. The result is delayed care, high-risk cases who aren't given enough attention, and a system that can't spot slow psychological decline before it reaches a critical point..

1.4 Contribution of Therabotics

This paper presents Therabotics, a full-stack web-based platform that directly addresses the research gaps identified above. The system makes the following original contributions to the field of digital mental health technology:

No.	Contribution	Description
C1	**Multi-Task Inference Engine**	A single LLM call simultaneously performs six psychological analysis tasks, replacing multiple isolated models with a unified, efficient inference architecture.
C2	**Sigmoid Risk Calibration Layer**	Raw probabilistic outputs from the language model are refined through a weighted sigmoid function incorporating emotion intensity and longitudinal escalation trend signals.
C3	**Embedded Clinical Instruments**	PHQ-9, GAD-7, and GHQ-12 assessments are natively integrated within the conversational interface, enabling passive clinical evaluation without formal intake procedures.
C4	**Min-Heap Priority Scheduling**	Clinical severity scores derived from validated assessments directly govern counselor appointment priority, ensuring the most at-risk students receive care first.

C5	**GloVe Emotion-Space Clustering**	A custom 10-dimensional word embedding model with pre-computed stable centroids classifies anonymous peer forum posts into five emotionally coherent support clusters.
C6	**Confidence-Gated Crisis Detection**	A model confidence threshold prevents the crisis protocol from triggering on ambiguous inputs, substantially reducing false-positive emergency escalations.

table i. original contributions of therabotics

1.5 Paper Organization

The remainder of this paper is structured as follows. Section II presents a thematic review of related literature spanning AI-driven mental health interventions, multi-task learning in healthcare, risk calibration methods, and privacy-preserving AI architectures. Section III describes the proposed Therabotics framework in detail, covering all core modules from the multi-task inference engine to the peer forum clustering system. Section IV details the overall system architecture including the frontend, backend, AI inference pipeline, database schema, and API design. Section V presents the mathematical formulations underpinning the risk scoring and calibration mechanisms. Section VI covers implementation details including the technology stack and deployment pipeline. Section VII describes the experimental evaluation approach. Section VIII presents results and discussion. Section IX compares Therabotics against existing systems. Sections X through XIV address innovation and contribution, ethical considerations, limitations, future work, and conclusion respectively.

II. LITERATURE REVIEW

2.1 AI-Driven Conversational Agents in Mental Health

In the last several years, the combination of artificial intelligence with mental health care has led to the creation of a new type of tool called conversational agents or chatbots. These tools are different from simple FAQ systems because they can have open-ended conversations with users, pick up on emotional cues, and give answers that are relevant to the situation. It's easy to see why mental health apps are popular: they're available 24/7, don't have any social stigma, and can help kids who would never go to a counseling center. But the gap between accessibility and therapeutic effectiveness has always been a problem in this area.Jackson, Lewis, and Wang's large study in 2025 [1] helped answer this question by combining results from dozens of different studies with young adult volunteers. Their research validated that AI-driven tools can yield quantifiable enhancements in psychological well-being, while also demonstrating a consistent trend: solutions that integrated conversational engagement with structured assessment techniques surpassed those that depended only on dialogue.

This discovery has a direct design effect: a chatbot that only talks to a student is not as useful as one that also measures, tracks, and routes. This idea was the basis for Therabotics. One big problem with many of the platforms that were assessed is that they are only meant to do one thing. A system designed to handle sadness may completely disregard anxiety indicators occurring inside the same dialogue. Because mental health problems don't often fit into tidy clinical categories, a student who is under a lot of stress during a test may show signs of tension, anxiety, and early depression all at the same time. This single-task approach creates a structural blind hole. Multi-task inference, as used in Therabotics, gives a more accurate picture of real psychological presentations by looking at many different aspects of suffering in each contact.

2.2 Digital Mental Health Interventions for University Students

The university years are a time of big changes for students. They have to deal with new social situations, academic expectations, financial difficulties, and doubts about who they are, often all at once and with less support from their families. This combination makes it easier for old problems to get worse and new mental health problems to show up. Digital intervention platforms have become popular as a way to deal with this problem. This is partly because they may be used by a lot of people without needing more counselors, and partly because they make it easier for students who don't want to get help in person to do so. The Indian academic setting introduces an additional layer to this difficulty. In addition to the usual stresses of school, children from Indian schools sometimes have to deal with strong cultural expectations about success, family honor, and personal stoicism that make it very awkward to openly talk about mental health issues. Studies conducted in this area [8] have consistently demonstrated that the aspect of anonymity significantly influences students' decisions to utilize support services. A platform that asks for a name, a student ID, or even an email address to join up makes it possible for students to avoid this moment of revelation.

Therabotics removes this barrier by giving each user a randomly created ID when they first come and not asking for any personal information at any stage during the engagement. It is also interesting to note how rarely the research on digital mental health aids for students talks about scheduling and prioritizing. Most of the platforms that were assessed handle all users as if they need follow-up and either give them self-directed resources or a general recommendation to university services. This doesn't take

into account the fact that there aren't enough counseling slots, and without an organized means to figure out which students need an appointment the most, those resources are typically given to the first person who asks for them instead of the person who needs them the most. Therabotics directly addresses this gap with its priority queue architecture, which uses standardized assessment results to calculate a clinical severity score and then uses that value to set the order of the appointment line.

2.3 Multi-Task Learning in Healthcare AI

Multi-task learning is the idea of training one computer model to make predictions for numerous related goals at simultaneously, instead of developing a separate model for each goal. The reason for this method is that tasks that are linked typically have patterns that help anticipate one outcome and contain information that is useful for others. This is especially important in clinical AI because the human illnesses that are being predicted are related to one other. A patient displaying indicators of chronic stress is more prone to exhibit anxiety symptoms; a student demonstrating feelings of hopelessness is at increased risk for both depression and crisis. Single-task models break this integrated image into separate pictures.

In the specialized field of healthcare natural language processing, collaborative modeling of related clinical data has repeatedly demonstrated superior performance compared to standalone modeling when the prediction targets exhibit semantic overlap. The shared encoder architecture, in which a single model learns a common representation of the input text and then branches into task-specific output heads, has become the most dependable way to put this idea into action. Therabotics uses this directly: instead of running six different models to find emotion, intensity, risk, intent, and confidence, and then make a response, a single large language model call makes all six outputs at once. This cuts down on both the cost of computing and the time it takes to make an inference,while also taking advantage of the shared signal across tasks. A modest but therapeutically significant benefit of multi-task inference is the internal regularization it facilitates. When a model has to optimize for both a main prediction and a number of secondary signals at the same time, it is less likely to fit too closely to surface-level language patterns in the main task. In Therabotics, the confidence score and emotion intensity outputs serve this exact purpose:they act as internal checks on the main risk probability output, stopping the system from going too far with a severe risk classification when the evidence is unclear or the model isn't sure.

2.4 Risk Prediction and Calibration in Clinical AI

One of the most overlooked issues with using AI to help doctors make decisions is the difference between a model that accurately classifies and one that is well-calibrated. A classifier can get very high accuracy on a test set while still giving confidence estimates that are always wrong, for always saying 90% confidence on predictions that are only right 60% of the time. In most cases, this mismatch is just a technical problem; in mental health applications, it becomes a safety risk for patients. A model that overstates its confidence in a student's crisis risk may set off emergency protocols for kids who are just upset, while a model that understates its confidence may miss students who really need treatment right away.

Standard post-hoc calibration techniques, such Platt scaling, change the raw output probabilities of a training model so that they match the rates of outcomes that were actually seen. These methods are effective in controlled experimental environments but fail to consider the dynamic, session-specific context inherent in mental health discussions. A student's risk level at message ten of a conversation is not independent of what they said at messages one through nine. Therabotics includes a trend score based on the emotional trajectory of the session in its calibration method. This creates a risk estimate that takes into account escalation trends instead of looking at each message on its own. Establishing clinical thresholds from the PHQ-9 [2], GAD-7 [3], and GHQ-12 [4] as the basis for the final priority score adds another layer of clinical foundation. These tools have been tested against organized psychiatric interviews in a wide range of demographics, thus their score thresholds have real clinical relevance. Therabotics makes sure that the most important decisions in the system about who gets to see a counselor first are based on evidence that is older than and goes beyond any one AI model. This is done by using these thresholds to set scheduling priority instead of just relying on AI-generated probability estimates.

2.5 Privacy-Preserving Design in Digital Health AI

The connection between mental health and privacy is more complicated than in virtually any other part of healthcare. A student who is concerned that their depression screening findings could be seen by faculty, accessed by administrative staff, or connected to their academic record has every logical reason to avoid screening altogether. This is not a hypothetical study; research on student mental health platform uptake has consistently shown that perceived privacy risk is one of the strongest indicators of non-engagement, especially in cultural contexts where mental health stigma is still high. The technical design of a mental health platform is not a neutral implementation element; it directly affects whether or not students will use the system. Federated learning is one way to solve this challenge at the model training level. It lets AI systems learn from data that is spread out over many devices or servers without that data ever leaving those devices or servers. This method provides excellent privacy protections for the training process, but it makes engineering much more complicated. It is better for large-scale deployments across several institutions than for a single institution that is still being developed. For Therabotics, the more practical and ready-to-use solution is pseudonymization at the data storage level. This means that every student interaction is stored only under a randomly generated session token, with no connection to institutional identity systems, email addresses, or names at any point in the data pipeline.

2.6 Synthesis of Research Gaps

The literature reviewed points to a clear and important conclusion: the individual parts that make a mental health AI system work conversational ability, clinical assessment, risk quantification, crisis detection, scheduling, peer support, and privacy protection have all been studied on their own, but putting them all together into one platform is still mostly unexplored. The average deployed system only covers two or three of these areas, thus the others have to be dealt with by hand or not at all. This fragmentation is not just a design flaw; it has significant effects for pupils whose decline goes unnoticed because no one system can see all the signals that are important.

Therabotics is built on the idea that these dimensions are not separate; they work together to create a therapeutic pipeline where each stage depends on the outcome of the one before it. Emotion detection informs risk calibration; calibrated risk dictates assessment triggering; assessment results establish scheduling priority; scheduling priority dictates which pupils receive timely professional care. Putting these steps on different platforms interrupts the pipeline and causes delays at each handoff. The next sections explain how Therabotics puts each stage of this integrated pipeline into action and how the stages work together to make a system that is better than the sum of its parts.

Feature	Woebot	Wysa	Koko	ELIZA-type	Therabotics
Multi-Task AI Inference	✗	✗	✗	✗	✓
Risk Calibration Layer	✗	✗	✗	✗	✓
Validated Clinical Tools	✗	Partial	✗	✗	✓
Priority Queue Scheduling	✗	✗	✗	✗	✓
Crisis Detection	Partial	Partial	✗	✗	✓
Anonymous Architecture	✗	✗	✓	✓	✓
Peer Forum + Clustering	✗	✗	✓	✗	✓
Mood Tracking	✓	✓	✗	✗	✓
Institutional Integration	✗	✗	✗	✗	✓

table ii. comparative analysis of existing mental health ai systems vs therabotics

III. PROPOSED SYSTEM: THERABOTICS FRAMEWORK

3.1 System Overview and Architecture

Therabotics is built as a full-stack web application with seven closely connected functional modules. Each module deals with a different part of the mental health intervention lifecycle, from the first emotional assessment to risk quantification, clinical evaluation, peer assistance, and professional routing based on priorities. Data flows continuously across modules, giving a complete and long-term picture of each student's mental state without ever needing to know who they are.

The seven main parts are: (1) the Conversational AI Interface, which is the main way for students to interact with the system; (2) the Multi-Task Inference Engine, which analyzes every student message at the same time; (3) the Risk Calibration Layer, which turns raw AI outputs into clinically meaningful severity scores; (4) the Clinical Assessment Module, which uses standardized psychological tests in chat; (5) the Priority Queue and Counselor Matching System, which sets up appointments based on clinical severity; (6) the GloVe Peer Forum Clustering Module, which sends anonymous forum posts to communities that are emotionally relevant; and (7) the Wellness Tracking Module, which keeps track of daily mood data to find long-term patterns of deterioration. The complete data flow is illustrated in Figure 1 below.

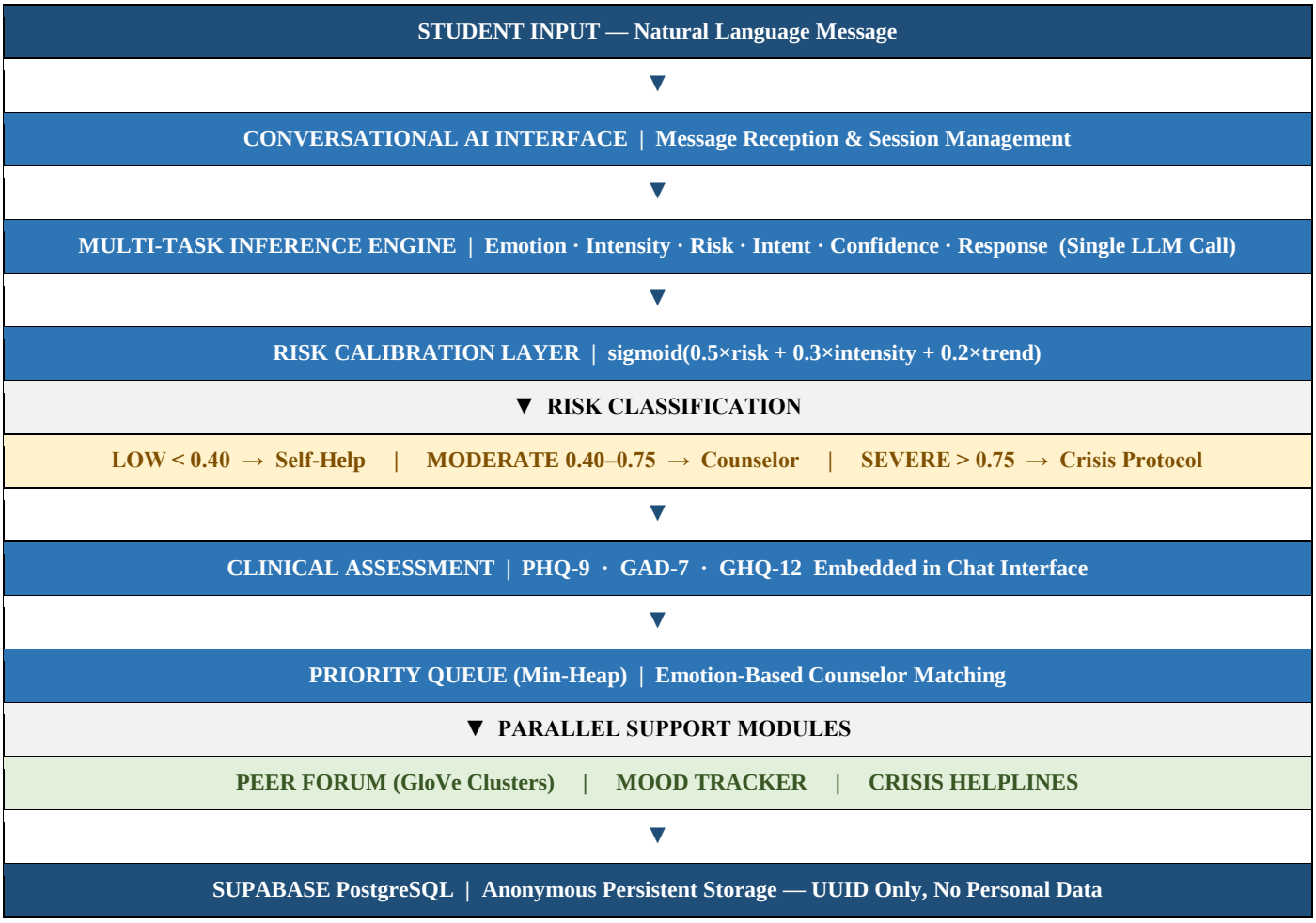

fig. 1. therabotics system architecture and complete data flow diagram

3.2 Multi-Task Deep Learning Inference Engine

The Multi-Task Inference Engine is the most important technical part of Therabotics. Using the Groq API, it uses the Llama 3.3 70B Versatile language model to do six different psychological analysis tasks in one inference call. This gets rid of the extra work that comes with running several specialized models in a row, speeds up the API, and takes advantage of the way that psychological conditions are related to each other—anxiety, depression, and crisis risk all share linguistic and semantic features that can be learned together.

The six tasks that are happening at the same time are technically defined as follows. Task 1 (Emotion Classification) sorts each student communication into one of six clear emotional groups: sad, anxious, stressed, hopeless, angry, or neutral. Task 2 (feeling Intensity Estimation) gives you a number between 0.0 and 1.0 that shows how strong the feeling is. Task 3 (Risk Probability Scoring) gives an uncalibrated estimation of the probability of clinically significant mental health risk. Task 4 (Intent Detection) sorts communicative intent into four groups: usual talk, assessment request, booking request, or crisis signal. Task 5 (Confidence Scoring) gives a model self-assessment of how reliable the inference is in the range of [0.0, 1.0]. Task 6 (Therapeutic Response Generation) creates a warm, caring response of two to three sentences that fits the emotional context that was found. The model is set to 0.7 degrees to balance creative responses with clinical consistency. It also has a maximum token limit of 400 to make sure that therapeutic replies are short and JSON output format to make sure that all six task aspects get structured, machine-readable outputs at the same time. This JSON forcing technique makes sure that every inference call produces a whole organized psychological profile instead of free-form text that needs to be parsed again.

Task	Output Type	Range / Categories	Clinical Purpose
T1	Emotion Class	sadness · anxiety · stress · hopelessness · anger · neutral	Identify primary emotional state for counselor matching and trend tracking
T2	Intensity Score	Float 0.0 – 1.0	Quantify emotional severity as calibration input to the risk layer
T3	Risk Probability	Float 0.0 – 1.0	Estimate uncalibrated clinical risk for sigmoid calibration
T4	Intent Label	normal_chat · assessment_request · booking_request · crisis_signal	Route student to the appropriate system response pathway
T5	Confidence Score	Float 0.0 – 1.0	Gate crisis protocol — suppressed if confidence < 0.6 to prevent false positives
T6	Therapeutic Reply	2–3 sentence natural language	Deliver empathetic, contextually appropriate conversational support

table iii. six simultaneous tasks of the multi-task inference engine

3.3 Risk Calibration Mechanism

Large language models' raw probabilistic outputs are not calibrated, which means that stated confidence levels don't always match real-world occurrence probabilities. In the field of mental health, this miscalibration has direct therapeutic effects: a model that is too sure of itself can wrongly label a moderately disgruntled student as being in a severe crisis, which would set off unneeded emergency protocols and damage user trust. On the other hand, an underconfident model may ignore risk signals from a student who is really in trouble, which could lead to a delay in getting help.

Therabotics addresses this through a three-input weighted sigmoid calibration formula applied to every inference output. The calibrated risk is computed as: CalibratedRisk = sigmoid(w1 × risk_probability + w2 × emotion_intensity + w3 × trend_score), where sigmoid(x) = 1 / (1 + e^(−x)). Weights are: w1 = 0.5 (primary risk signal), w2 = 0.3 (severity amplifier), w3 = 0.2 (temporal escalation detector). The trend score is derived from the last three session emotions mapped against the escalation hierarchy: neutral(0) → stress(1) → anxiety(2) → sadness(3) → hopelessness(4) → crisis(5). If the most recent emotion ranks higher than the earliest in the window, trend_score = 1.0 (escalating); otherwise 0.0 (stable).

Calibrated Risk	Risk Level	Protocol	System Action
> 0.75	SEVERE	Crisis Protocol	Emergency booking + 5 crisis helplines displayed + high_risk_flagged = True
0.40 – 0.75	MODERATE	Counselor Suggestion	Counselor recommendation shown in chat with booking prompt
< 0.40	LOW	Self-Help Resources	Curated articles, coping strategies, and wellness tips recommended

table iv. risk level classification and system response protocol

3.4 Clinical Assessment Module (PHQ-9, GAD-7, GHQ-12)

The Therabotics conversational interface has three globally validated clinical instruments built in. This lets students passively examine their mental health without having to go to separate evaluation portals. The integration technique makes each assessment feel like a natural part of a continuous discourse, which makes it much less formal than usual. This is a big problem for students who are trying to finish their assessments. The Patient Health Questionnaire-9 (PHQ-9) [2] has nine questions that can be graded from 0 to 3, with a maximum score of 27. If you score 15 or higher, you'll instantly get a prompt to book an appointment. If you score between 10 and 14, you'll get a suggestion for a counselor. If you score below 10, you'll get self-help materials. The Generalised Anxiety Disorder Scale-7 (GAD-7) [3] has seven questions, and the highest score is 21. The cutoff points are 10 (high risk) and 5 (moderate). The General Health Questionnaire-12 (GHQ-12) [4] uses a binary scoring system for 12 items, with a score of 6 or higher indicating a lot of suffering. To get the priority score, you first get the severity by taking the maximum of PHQ-9, GAD-7, and GHQ-12 times 2. Then you subtract the severity from 100 to get the priority score. This puts the most clinically severe pupils at the top of the min-heap queue.

Instrument	Items	Max Score	High Risk	Moderate	Low Risk
PHQ-9 (Depression)	9	27	≥ 15	10 – 14	< 10
GAD-7 (Anxiety)	7	21	≥ 10	5 – 9	< 5
GHQ-12 (General Health)	12	12	≥ 6	3 – 5	< 3

table v. clinical assessment instruments embedded in therabotics

3.5 Priority Queue and Counselor Matching System

The counselor scheduling subsystem uses a min-heap priority queue that is implemented with Python's heapq package. The root of a min-heap is the element with the smallest key value, which can be found in O(1) time. Insertion takes O(log n) time. This is the best structure for a scheduling system where the most clinically severe case must always be easy to find, no matter how many cases are in the queue. Each entry is a tuple with three parts: a priority score, a booking ID, and a booking dictionary.

A lower priority score means a higher clinical severity. The inference engine finds the student's main emotion, and then the system finds all the counselors whose specialties include that emotion category. The system picks the counselor with the most open appointment slots from the list of matching counselors. This balances clinical appropriateness with scheduling efficiency.

Counselor	Specialty	Matched Emotions
Dr. Priya Sharma	Depression & Anxiety	hopelessness · sadness · anxiety
Dr. Arjun Mehta	Stress & Burnout	stress · anger · neutral
Dr. Sneha Rao	Relationship & Family Therapy	sadness · loneliness
Dr. Kavya Patel	General Mental Health	neutral · stress · anxiety
Dr. Rohan Das	Trauma & Crisis Support	hopelessness · crisis · sadness

table vi. emotion-based counselor matching matrix

3.6 GloVe Peer Forum Clustering Module

The peer forum module uses a three-stage NLP pipeline to put each anonymous student contribution into one of five emotionally coherent groups. The pipeline works just with pre-computed centroid vectors, so it doesn't need real-time model inference.This makes it fast and accurate in classifying posts, no matter how many there are. In Stage 1, each phrase is linked to a 10-dimensional GloVe embedding vector. The dimensions of this vector show: stress from school, stress from a love connection, stress from family conflict, stress from social isolation, despair, anxiety, hopelessness, weariness, performance pressure, and interpersonal conflict. Mean pooling over word vectors turns each post into a document-level vector. Unknown words are handled as zero vectors. Five cluster centroids are pre-computed at startup using carefully chosen seed phrases for each cluster in Stage 2. Real student posts are never used to fit centroids, which ensures consistent categorization throughout the system's entire existence. In Stage 3, the argmax function uses cosine similarity between each post vector and all five centroids to figure out which cluster a post belongs to.

Cluster	Primary Themes	Active Embedding Dimensions
Exam Stress	Grades · deadlines · assignments · academic burnout	Dim[0]: Academic · Dim[7]: Exhaustion · Dim[8]: Pressure
Relationship Anxiety	Breakups · rejection · romantic worry · partner conflict	Dim[1]: Romantic · Dim[5]: Anxiety · Dim[9]: Conflict
Family Pressure	Parental expectations · home conflict · cultural stress	Dim[2]: Family · Dim[8]: Pressure · Dim[9]: Conflict
Loneliness	Social isolation · friendlessness · feeling invisible	Dim[3]: Isolation · Dim[4]: Sadness · Dim[6]: Hopelessness
General Distress	Mixed or unclassified emotional pain	Dim[4]: Sadness · Dim[5]: Anxiety · Dim[6]: Hopelessness

table vii. glove peer forum emotional clusters and embedding dimensions

3.7 Crisis Detection Protocol

The subsystem for detecting crises works as a two-layer safety system. In the first layer, every message that comes in is checked for words that suggest a crisis, such as "suicide," "kill myself," "end my life," "hurt myself," "self harm," and "die." The crisis

protocol is triggered as soon as any keyword is found, skipping the usual AI inference to speed up response times in real emergencies.

The second layer fixes the problem of pure keyword detection having a lot of false positives. Phrases like "this exam is killing me" or "I could die of embarrassment" are examples of figurative language that use crisis keywords but don't provide a clinical risk. Therabotics fixes this with its confidence-gated mechanism. If the inference engine's Task 5 confidence score drops below 0.6, the crisis protocol is not triggered, even if there are keywords present. Instead, the system makes a clarifying query. When the crisis protocol is turned on, five things happen at once: an empathetic, non-judgmental message is sent back; five Indian crisis helplines with tap-to-call links are shown; the session state is set to awaiting_booking_prompt; high_risk_flagged is set to True in the database; and immediate anonymous emergency counselor booking is offered.

IV. SYSTEM ARCHITECTURE

4.1 Architectural Overview

Therabotics has a four-layer client-server design that keeps things separate between presentation, application logic, AI intelligence, and data persistence. This separation makes guarantee that each layer can be maintained, scaled, or changed on its own without affecting the rest of the system. The frontend only talks to the backend through a set of REST API endpoints over HTTPS. The backend handles all business logic and AI calls. The AI layer makes inferences and sends back structured outputs. The database layer handles all permanent storage using anonymous IDs. The full four-layer architecture is shown in Figure 2. For three reasons related to the Therabotics use case, this layered approach was chosen on purpose over a monolithic architecture. First, mental health platforms need to be able to change their AI models when new ones come along without affecting how users interact with the platform. Second, because the data is so sensitive, storage logic needs to be separate from application logic. This reduces the chance that application-layer defects would accidentally expose data. Third, the frontend (Vercel) and backend (Render) have different deployment goals. This lets each layer scale on its own based on how it's being used. In a universitysetting, usage patterns are very seasonal, with spikes around exam time.

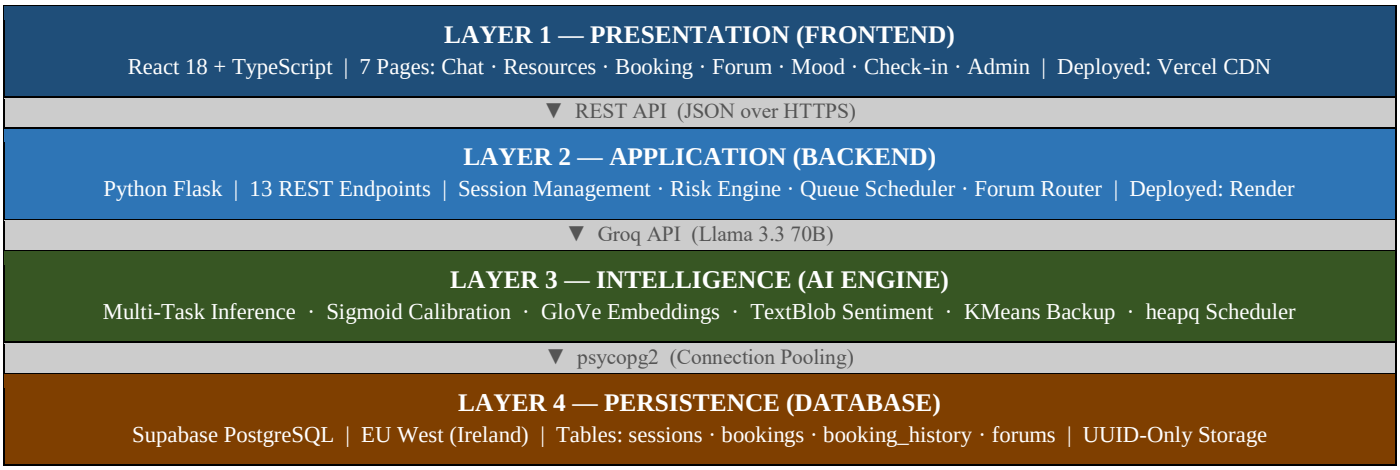

fig. 2. therabotics four-layer system architecture

4.2 Frontend Architecture

The interface that students see is designed with React 18 and TypeScript. It has a component-based architecture that lets UI elements be used on all seven application pages. TypeScript's static typing helps prevent runtime problems in a field where interface reliability is very important. For example, if a student is in a crisis and the interface crashes or behaves unexpectedly, it could have serious effects. The application is hosted on Vercel's global CDN, which means that students in India can access it quickly from anywhere without having to connect to a university-specific network. The seven frontend pages are based on the idea of "minimal friction." Each page can be reached by clicking on a tab, and there is no need to log in, fill out a form, or wait for a session to end. The Chat page is the main place to start, and it shows live emotion and danger indicators that alter when each message is sent. The Admin page is the only view that can be managed by an administrator. It gives counselors and administrators access to queue analytics and AI engine performance metrics without showing any student data.

Page	Route	Key Features
Chat	/chat	AI conversation, live emotion badge, risk level indicator, assessment trigger, crisis helpline display
Resources	/resources	Curated mental health articles, coping strategy videos, self-help tip cards
Book Session	/book	Anonymous counselor booking form, appointment history, queue position indicator
Peer Forum	/forum	Anonymous post creation, GloVe cluster tag display, heart/hug/understand reactions
Mood Tracker	/mood	1–5 daily mood logging, 7-day colour-coded bar chart, trend label (Improving/Stable/Declining)
Daily Check-in	/checkin	3-question wellness assessment (sleep · stress · energy), wellness score, counselor alert if score < 2.5
Admin	/admin	Queue analytics, risk distribution chart, AI engine performance view (admin-only access)

table viii. therabotics frontend pages and key features

4.3 Backend Architecture and API Design

Python Flask is a lightweight web framework that was chosen for the backend because it has little overhead and works well with the scientific Python ecosystem. NumPy is used for vector operations, scikit-learn is used for the GloVe clustering pipeline, and psycopg2 is used to connect to PostgreSQL. Flask is also easy to understand and maintain for a small development team, which is a significant practical issue for a final-year project that may be continued by future student cohorts. The backend has thirteen REST endpoints that cover all of the system's features. At the HTTP level, each endpoint is stateless. This means that all session information is pulled from the database at the start of each request using the anonymous user token that was sent in the body or URL path of the request. Because this approach doesn't keep track of sessions or shared in-memory state between instances, the backend can be horizontally scaled. Table VIII below shows the thirteen API endpoints and what they do.

Method	Endpoint	Description
POST	/chat	Main entry point — runs multi-task AI inference, calibration, risk classification, and returns therapeutic response
POST	/book	Creates an anonymous counselor booking and inserts into the min-heap priority queue
GET	/booking-history/:id	Returns complete booking history for a given anonymous user token
GET	/queue-stats	Admin endpoint — returns current priority queue overview and risk distribution
POST	/forum/post	Accepts anonymous forum post, runs GloVe clustering, assigns cluster label and sentiment score
GET	/forum/posts	Returns all forum posts with cluster labels, sentiment scores, and reaction counts
POST	/forum/react	Adds a heart, hug, or understand reaction to a specified forum post
POST	/mood	Logs a daily mood rating (1–5 scale) with optional student note
GET	/mood/:id	Returns 30-day mood history with computed average and Improving/Stable/Declining trend
POST	/daily-checkin	Submits sleep, stress, and energy ratings — computes daily wellness score
GET	/daily-checkin/:id	Returns check-in history and wellness score breakdown
POST	/reset-session/:id	Resets conversation state for a given session token
GET	/db-status	Returns database connection health and table record counts

table ix. rest api endpoints and descriptions

4.4 AI Inference Pipeline

When a student sends a message to the /chat endpoint, it goes through a series of steps before the system sends back a response. The backend validates the message against the crisis keyword list in the first step. If a keyword match is obtained and the previous session's confidence score is higher than 0.6, the crisis protocol is started right away without going through LLM inference. This short-circuit system makes sure that real crisis signals get the quickest reaction. If there is no immediate crisis, the message is sent to the Groq API together with the last 10 messages from the session history. A carefully crafted system prompt tells the Llama 3.3 70B model to return a JSON object with all six task outputs at once. The JSON that comes back is processed, and each field is taken out. Emotion and intent are utilized to make routing decisions, intensity and risk feed the calibration layer, confidence gates the crisis procedure, and the therapeutic reaction goes straight to the student. The calibrated risk score and emotion are then saved to the session record in Supabase. This updates the emotion history and trend score for future inference calls.

4.5 Database Schema Design

The database layer employs Supabase's managed PostgreSQL service, which is hosted in the EU West region (Ireland). This service was chosen because it is GDPR-compliant and has a lot of free-tier capacity, which is enough for a university pilot deployment. There are four tables in the schema: sessions, bookings, booking_history, and forums. It's important to note that none of the tables in the schema have a column for a student's name, email address, student ID, or any other information that could identify them. The user_id column is the only way to identify a user in the system. It holds a randomly generated UUID that is assigned when the first session is created. Figure 3 displays the full database schema, including the names of the columns and the types of data they hold. The sessions table is the main record of each student's history of interactions. It stores the last ten chat messages as a JSONB array, the last ten detected emotions for trend analysis, the most recent calibrated risk score and confidence value, and a boolean flag that shows whether a high-risk event has been recorded in the current session. The forums database has anonymous postings from peers, each with a GloVe-assigned cluster label, a TextBlob sentiment score, and a JSONB reactions object that keeps track of how many of each form of reaction there are for each post. The booking_history table is similar to the bookings table, but it keeps records forever, even after cancellations, giving institutions a full longitudinal audit trail for reporting.

sessions	bookings	booking_history	forums
■ user_id (PK)	■ id (PK)	■ id (PK)	■ id (PK)
state	user_id	user_id	content
phq9 [JSONB]	issue	issue	cluster_label
gad7 [JSONB]	risk_level	risk_level	sentiment_score
ghq12 [JSONB]	priority_score	priority_score	reactions [JSONB]
messages [JSONB]	booking_id	booking_id	created_at
emotion_history	status	calibrated_risk	
last_emotion	created_at	status	
calibrated_risk		created_at	
confidence			
priority			
high_risk_flagged			
updated_at			

fig. 3. therabotics database schema — four-table postgresql design

Layer	Technology	Version / Tier	Role
Frontend	React + TypeScript	React 18	UI components, routing, state management
Styling	CSS + Inline Styles	—	Responsive single-column layout
Backend	Python Flask	v3.x	REST API server, session logic, scheduling
LLM	Llama 3.3 70B	Groq API (Free)	Multi-task psychological inference engine
NLP	TextBlob	v0.17	Supplementary sentiment polarity scoring

Embeddings	Custom GloVe	10-dimensional	Forum post emotion-space vectorisation
ML	scikit-learn	KMeans / TF-IDF	Backup forum clustering support
Queue	Python heapq	Built-in	Min-heap priority scheduling
Database	Supabase PostgreSQL	EU West, Ireland	Persistent anonymous session storage
DB Driver	psycopg2	v2.9	PostgreSQL connection pooling
Frontend Host	Vercel	Global CDN	Frontend deployment and global delivery
Backend Host	Render	Free Tier	Backend cloud deployment

table x. complete technology stack

V. MATHEMATICAL FORMULATION

In This section we look at the formal mathematical definitions behind the three main computational tools used in Therabotics. These are the multi-task loss objective, which controls joint psychological inference; the composite risk scoring function, which combines model outputs into a single severity signal; and the sigmoid calibration transform, which turns raw scores into clinically meaningful probability values. These formulations provide the system's behavior a theoretical basis and make it easier for future research to use the same method again.

5.1 Multi-Task Learning Loss Function

In a typical single-task learning configuration, a model reduces one loss function that corresponds to one prediction goal. The multi-task formulation builds on this by making the model minimize a weighted composite of losses across all prediction heads at the same time. In Therabotics, the six inference tasks create diverse types of outputs, such as categorical classifications, continuous scalars, and free-text generation. Each output has its own loss component. The total training loss is the sum of the losses for each task:

$$L_{total} = \alpha \cdot L_{emotion} + \beta \cdot L_{intensity} + \gamma \cdot L_{risk} + \delta \cdot L_{intent} + \varepsilon \cdot L_{confidence} + \zeta \cdot L_{response}$$

equation 1. multi-task total loss function

where each coefficient controls the relative contribution of its associated task to the total gradient signal during optimisation. The individual loss terms are defined according to the nature of each task output:

Term	Task	Loss Type	Justification
L_emotion	Emotion Classification	Cross-Entropy	Six-class categorical output — CE is standard for discrete multi-class classification
L_intensity	Intensity Estimation	Mean Squared Error	Continuous scalar in [0,1] — MSE penalises large deviations from true intensity
L_risk	Risk Probability	Binary Cross-Entropy	Binary risk signal (at-risk / not-at-risk) — BCE is optimal for binary probability outputs
L_intent	Intent Detection	Cross-Entropy	Four-class categorical — CE handles multi-class intent label prediction
L_confidence	Confidence Estimation	Mean Squared Error	Continuous self-assessment score — MSE encourages accurate uncertainty quantification
L_response	Response Generation	Language Model Loss	Auto-regressive next-token prediction — standard LM cross-entropy over generated tokens

table xi. loss function components for each inference task

The weight coefficients α, β, γ, δ, ε, and ζ are hyperparameters that show how important each activity is in a therapeutic setting. In the Therabotics implementation, risk probability and emotion classification are the most important factors because they directly

affect the system's decisions about what to do. Intensity and confidence act as secondary signals that change the main outputs. They get reduced but still non-zero weights so that they can still provide useful gradient information during the optimization process.

5.2 Composite Risk Score Computation

We don't use the raw results of the multi-task inference engine to make clinical choices. Instead, they are combined into one composite risk score that takes into account the main risk likelihood as well as two contextual signals: emotion intensity and session trend. This gives a more stable and contextually informed estimate of severity. The raw risk score R is the sum of the following:

$$R = w1 \cdot P\_risk + w2 \cdot E\_intensity + w3 \cdot T\_trend$$

where:
- w1 = 0.5 (primary risk weight)
- w2 = 0.3 (emotion intensity weight)
- w3 = 0.2 (trend escalation weight)
- P_risk = raw risk probability from Task T3 ∈ [0.0, 1.0]
- E_intensity = emotion intensity from Task T2 ∈ [0.0, 1.0]
- T_trend = session escalation trend score ∈ {0.0, 1.0}

equation 2. composite risk score computation

The weight allocations show that there was a determined clinical prioritization. The principal risk probability P_risk, which is the direct model output for the risk prediction job, has the highest weight of 0.5. The emotion intensity E_intensity, which measures how bad the student is feeling right now, has a weight of 0.3. This means that when the student is really upset, the risk score goes up, and when they are not very upset, it goes down. The trend score T_trend has the lowest weight of 0.2, but it does something that no other score does: it adds time memory to what would be a single-turn risk estimate. A session-level emotion escalation analysis is used to get the trend score T_trend. The method puts each identified emotion on a scale of severity, from neutral (0) to crisis (5). The algorithm gets the previous three detected emotion values from the session history at any time throughout a session and uses them to figure out the trend like this:

Let $H = [e_{n-2}, e_{n-1}, e_n]$ be the last three session emotions

$$T\_trend = 1.0 \quad \text{if} \quad rank(e\_n) > rank(e\_\{n-2\})$$
$$T\_trend = 0.0 \quad \text{otherwise}$$

rank(e) maps emotion e to its position in the escalation hierarchy:
- rank: {neutral:0, stress:1, anxiety:2, sadness:3, hopelessness:4, crisis:5}

equation 3. session trend score derivation

This approach guarantees that a student transitioning from neutral content to hopelessness over three consecutive messages earns a significantly higher risk score than a student conveying hopelessness in a single isolated message inside an otherwise neutral session. T_trend is binary, which means that it doesn't have the instability that would happen if you used a continuous trend metric based on only three data points.

5.3 Sigmoid Calibration Function

In Section 5.2, we created the composite risk score R by taking a weighted linear combination of values in the range [0, 1]. However, this score does not always represent the true empirical frequency of clinical risk at that value. The sigmoid function is used to change R into a calibrated risk probability R_calibrated that is between 0 and 1 and has the useful property of compressing extreme values while keeping the order of the ranks:

$$R\_calibrated = sigmoid(R) = 1 / (1 + e^{(-R)})$$

Expanding the full expression:

$$R\_calibrated = 1 / (1 + exp(-(w1 \cdot P\_risk + w2 \cdot E\_intensity + w3 \cdot T\_trend)))$$

$$= 1 / (1 + exp(-(0.5 \cdot P\_risk + 0.3 \cdot E\_intensity + 0.2 \cdot T\_trend)))$$

equation 4. sigmoid calibration function — full expansion

There are a number of reasons why the sigmoid function is a good fit for this use. First, its output is always between 0 and 1, hence R_calibrated may always be seen as a probability, no matter how big the input is. Second, it keeps the rank order of risk scores by always giving a student with a higher composite R a higher R_calibrated. Third, it compresses scores that are already very high or very low toward 0.75 and 0.25, respectively, instead of letting them become closer to 1.0 or 0.0.

This stops the system from being too sure about borderline cases. The calibrated risk R_calibrated is then mapped to one of three clinical risk levels using fixed thresholds derived from the system's operational requirements: R_calibrated > 0.75 triggers the crisis protocol with emergency booking and helpline display; 0.40 ≤ R_calibrated ≤ 0.75 generates a moderate-risk counselor suggestion; and R_calibrated < 0.40 returns low-risk self-help resources. The goal of these thresholds was to find a good balance between sensitivity and specificity. In other words, they wanted to keep the false-positive rate of crisis protocol activations low enough that students don't get alert fatigue, while also keeping the false-negative rate low enough that real high-risk cases are reliably found.

5.4 Clinical Priority Score for Queue Scheduling

Once a student has finished one or more of the embedded clinical exams (PHQ-9, GAD-7, GHQ-12), their scores are added to the AI-calibrated risk score to get a final clinical priority score. This score decides where they are in the min-heap counselor appointment queue. To get the severity score, you first find the highest score from the normalized assessment scores:

$$severity = max(PHQ9\_score, GAD7\_score, GHQ12\_score \times 2)$$

$$priority\_score = 100 - severity$$

Queue position: lower priority_score = higher clinical severity = served first

equation 5. clinical priority score for min-heap queue scheduling

To make the GHQ-12 score (0–12) equivalent to the PHQ-9 (0–27) and GAD-7 (0–21), it is multiplied by 2 before taking the maximum. Then, the priority score is calculated by subtracting the severity score from 100. This creates an inverted scale where a student with the highest clinical severity gets a priority value of 0 and is at the bottom of the min-heap. This inversion takes advantage of the min-heap's inherent trait of always serving the element with the smallest key value first. This makes the ordering semantics of the data structure match the clinical need to serve the most severe instances first.

5.5 GloVe Cosine Similarity for Forum Clustering

To cluster forum posts, you find the cosine similarity between each post's embedding vector and the five pre-computed cluster centroid vectors. If you have a post embedding vector p and a set of cluster centroid vectors {c1, c2, c3, c4, c5}, you can figure out the cluster assignment by:

$$\text{post\_vector} = (1/|W|) \cdot \Sigma_{w \in W} \text{GloVe}(w)$$

$$\text{cosine\_similarity}(p, c_k) = (p \cdot c_k) / (||p|| \cdot ||c_k||)$$

$$\text{cluster\_assignment} = \text{argmax}_k [\ \text{cosine\_similarity}(p, c_k)\ ]$$

where:
- W = set of known words in the post
- $\text{GloVe}(w)$ = 10-dimensional embedding vector for word w
- c_k = pre-computed centroid for cluster k ($k = 1..5$)
- $||v||$ = L2 norm of vector v

equation 6. glove cosine similarity forum cluster assignment

The cosine similarity metric is preferred over Euclidean distance as it assesses the angular alignment of vectors instead of their absolute magnitudes. This means that it can handle posts of different lengths. For example, a brief two-sentence message and a long paragraph discussing the same emotional issue will both have vectors heading in the same direction, even if their magnitudes are very different. When students use words that aren't in the vocabulary, they are mapped to zero vectors and left out of the mean pooling computation. This makes sure that out-of-vocabulary tokens don't change the document-level representation.

5.6 Daily Wellness Score Computation

The daily check-in module asks students to rate their sleep quality (S), stress level (X), and energy level (E) on a scale of 1 to 5. The stress rating is flipped before being added to the wellness score because higher stress means poorer wellbeing. The overall wellness score W is calculated as follows:

$$W = (S + (6 - X) + E) / 3$$

where:
- S = sleep quality rating $\in \{1, 2, 3, 4, 5\}$ (1=very poor, 5=very well)
- X = stress level rating $\in \{1, 2, 3, 4, 5\}$ (1=very stressed, 5=not at all)
- E = energy level rating $\in \{1, 2, 3, 4, 5\}$ (1=very low, 5=very high)

Classification:
- $W >= 3.5$ → Good (green indicator)
- W 2.5–3.5 → Fair (amber indicator)
- $W < 2.5$ → Low (red indicator + counselor booking prompt)

equation 7. daily wellness score computation

The stress term (6 - X) is flipped such that all three parts add to the wellness score when they show good health. For example, a student who says they sleep well (S=5), have little stress (X=1, flipped to 5), and have a lot of energy (E=5) gets the highest wellness score of 5.0. A student who says they don't sleep well, are stressed out, and don't have much energy gets a score of at least 1.0. The alert boundary was set at 2.5 because it is the average reaction across all three dimensions that is 2 or lower. This shows that the person is in trouble in multiple areas and needs expert help.

Section	Formula	Output Range	Clinical Use
5.1	L_total = Σ coeff·L_task	0 → ∞ (minimised)	Governs joint multi-task model optimisation
5.2	R = 0.5·P_risk + 0.3·E_int + 0.2·T_trend	[0.0, 1.0]	Composite raw risk signal before calibration
5.3	R_cal = 1/(1+e^(-R))	(0.0, 1.0)	Calibrated risk probability for level classification
5.4	priority = 100 − max(PHQ9, GAD7, GHQ12×2)	[0, 100]	Min-heap queue position (lower = more severe)
5.5	cluster = argmax cosine_sim(post, centroid_k)	k ∈ {1..5}	Forum post emotional cluster assignment
5.6	W = (S + (6−X) + E) / 3	[1.0, 5.0]	Daily wellness score with counselor alert at <2.5

table xii. summary of all mathematical formulations in therabotics

VI. IMPLEMENTATION DETAILS

This part talks about the real-world engineering choices that were taken while building Therabotics, such as how to choose and set up each library and how to arrange the deployment pipeline. The rationale for each decision is articulated in relation to the particular restrictions and requirements of a university mental health platform, rather than as generic software engineering guidelines.

6.1 Backend Implementation

The backend service is written in Python, which was chosen for three practical reasons. First, Python has the most extensive collection of scientific and NLP libraries. NumPy, scikit-learn, and NLTK are all easy to install with pip, which makes it easier to add embedding calculations and sentiment analysis to the API layer. Second, Python's readability makes it easier for future students to add to the codebase. Third, the Groq Python SDK gives you a simple, well-documented way to use the Llama 3.3 70B model with very little extra code.

Flask was chosen over heavier frameworks like Django or FastAPI for a specific reason: the app doesn't need an ORM, template engine, or admin interface. Also, Flask's small footprint means that the server starts up faster and uses less memory on Render's free tier, which is a practical concern since the free tier goes to sleep after fifteen minutes of inactivity. Each Flask route is kept small on purpose, and all of the business logic is moved to separate module files forinference, calibration, scheduling, and clustering. This split makes sure that the route handlers stay readable and that each part can be unit-tested on its own.

```
# Multi-task inference — single Groq API call returning JSON
def run_inference(message, session_history, trend_score):
    prompt = build_system_prompt(session_history, trend_score)
    response = groq_client.chat.completions.create(
        model  = 'llama-3.3-70b-versatile',
        messages = [{'role': 'system', 'content': prompt},
                    {'role': 'user',   'content': message}],
        temperature      = 0.7,
        max_tokens       = 400,
        response_format  = {'type': 'json_object'}
    )
    return json.loads(response.choices[0].message.content)
```

fig. 4. multi-task inference function — groq api call with forced json output

The inference function above wraps up the whole multi-task LLM call in less than fifteen lines. The system prompt that is sent to the model is built dynamically using the past ten session messages and the current trend score. This makes sure that the model has enough conversational context to do meaningful psychological analysis. The mandated JSON response format makes sure that the output can be parsed directly without using regex or string manipulation. This lowers the chance of parsing failures when the system is under heavy stress.

6.2 Risk Calibration and Session Management

After the inference call comes back, the calibration module gets the raw risk probability, emotion intensity, and trend score and uses the weighted sigmoid formula from Section V. The calibrated risk value is instantly saved to the Supabase sessions table together with the detected emotion and confidence score. This updates the session record that will be used to figure out the trend for the next inference call. This write transaction uses psycopg2's connection pooling to minimize the extra work of making a new database connection for each request. This is a good optimization because the /chat endpoint is the most frequently called route in the system. Session state is kept on the server instead of in a cookie or JWT payload on the client side. This choice was chosen to defend the anonymity model: if session state were maintained on the client side, it may be intercepted or linked to browser fingerprinting data.

The solution protects the session's clinical contents by storing all psychological data on the server behind a UUID that the client has but can't decode. This way, even if a client device is hacked, it won't give away any information about the session

```python
# Sigmoid calibration with weighted inputs
import math

def calibrate_risk(p_risk, e_intensity, t_trend,
                   w1=0.5, w2=0.3, w3=0.2):
    raw = w1 * p_risk + w2 * e_intensity + w3 * t_trend
    return round(1 / (1 + math.exp(-raw)), 4)

def get_risk_level(calibrated_risk):
    if calibrated_risk > 0.75:  return 'severe'
    if calibrated_risk >= 0.40: return 'moderate'
    return 'low'
```

fig. 5. risk calibration and level classification functions

6.3 Priority Queue Implementation

The Python built-in heapq module is used to make the counselor appointment queue. This module gives you a fast min-heap over a regular Python list. We chose heapq over a database-backed queue or an external message broker on purpose. For the size of a single university deployment, the in-memory heap allows for insertion and retrieval in less than a millisecond without the added work of running a separate queuing service. When the server starts up, it loads all of the pending bookings from the Supabase.bookings table into the heap and sorts them by priority score. Each heap item is a Python tuple with three parts: [priority_score, booking_id, booking_dict]. The way Python compares tuples makes sure that entries are sorted first by priority_score and then by booking_id if there is a tie. This tiebreaker stops comparison mistakes that could happen if two students had the same priority scores. This is a situation that could happen in real life if both students had the same PHQ-9 result. The booking_dict has all the information needed to show the case to a counselor without having to hunt it up in a second database.

```
import heapq

booking_heap = []  # module-level min-heap

def enqueue_booking(priority_score, booking_id, booking_data):
    heapq.heappush(booking_heap,
                   [priority_score, booking_id, booking_data])

def get_next_patient():
    if booking_heap:
        return heapq.heappop(booking_heap)  # O(log n)
    return None

def compute_priority(phq9, gad7, ghq12):
    severity = max(phq9, gad7, ghq12 * 2)
    return 100 - severity  # lower = more severe = served first
```

fig. 6. min-heap priority queue — enqueue, dequeue, and priority computation

6.4 GloVe Embedding and Forum Clustering

The forum clustering pipeline is set up once when the server starts up and stays in memory for the whole time the backend process is running. Initialization means making the 10-dimensional GloVe embedding dictionary from a hardcoded list of emotionally significant words, calculating the five cluster centroid vectors from their seed sentence sets, and saving both in module-level variables that the clustering function can use. This one-time startup cost, which usually takes less than two seconds, means that you don't have to recalculate embeddings every time you submit a forum post. The embedding dictionary connects each word to its 10-dimensional vector. Each dimension was given a weight by hand that showed how relevant the word was to one of ten psychological themes: academic pressure, romantic stress, family conflict, isolation, sadness, anxiety, hopelessness, exhaustion, performance pressure, and interpersonal conflict. For instance, the term "exam" has a high value on dimension zero (academic) and dimension eight (pressure), but a low value on dimension eight (relationship) and dimension eight (family). We chose to build this model by hand instead of using a pre-trained GloVe model since the typical GloVe training corpora don't do a good job of capturing the special emotional lexicon used in conversations about mental health among college students.

```
# GloVe forum clustering — three-stage pipeline
def embed_post(text):
    words   = text.lower().split()
    vectors = [GLOVE[w] for w in words if w in GLOVE]
    if not vectors:
        return np.zeros(10)          # fallback for unknown posts
    return np.mean(vectors, axis=0)  # mean pooling

def assign_cluster(post_text):
    post_vec   = embed_post(post_text).reshape(1, -1)
    centroid_mat = np.array(list(CENTROIDS.values()))
    similarities = cosine_similarity(post_vec, centroid_mat)[0]
    cluster_idx  = np.argmax(similarities)
    return list(CENTROIDS.keys())[cluster_idx]
```

fig. 7. glove embedding and cosine similarity cluster assignment pipeline

6.5 Frontend Implementation

The React frontend talks to the Flask backend using a common API service module that wraps all fetch calls with the student's anonymous UUID, takes care of loading and error statuses, and makes sure that request payloads are always formatted the same way. Because everything is in one place, if the backend URL changes (for example, when switching from Render's free tier to a premium deployment), the change just needs to be made in one file instead of in many different component files. All seven pages use this shared service module, which makes it clear and easy to check that the frontend depends on the backend. The Chat page has the hardest job of managing state of all the frontend components. It keeps track of the whole message history array, the current emotion badge value, the live risk level indicator, and the assessment state machine, which shows which clinical instrument is presently being used. Instead of utilizing a global state manager like Redux, which would make things a lot more complicated for a single-page app of this size, these states are handled by React's built-in useState and useEffect hooks. The assessment state machine is a simple index counter that moves through the PHQ-9, GAD-7, or GHQ-12 question arrays as the student answers each question. It keeps track of the running score and takes the right action when the last question is answered.

6.6 Deployment Pipeline

The deployment architecture links a GitHub repository to two different cloud hosting providers using automatic continuous deployment hooks. When you push to the main branch, both the frontend and the backend get built and redeployed at the same time. This means that a code change tested on a local machine is live in production within two to three minutes of being pushed, and no manual deployment processes are needed. Below, Figure 4 shows the full deployment workflow from local development to live production.

Platform environment variables, not hardcoded values in the codebase, handle a number of settings that are specific to the environment. Render injects the Groq API key, Supabase connection string, and database password at build time. This makes sure that sensitive credentials are never stored in the GitHub repository. You can use the same codebase to point to a local development server or the production Render URL, depending on the build environment. This is possible because the frontend's backend URL is stored as a Vercel environment variable.

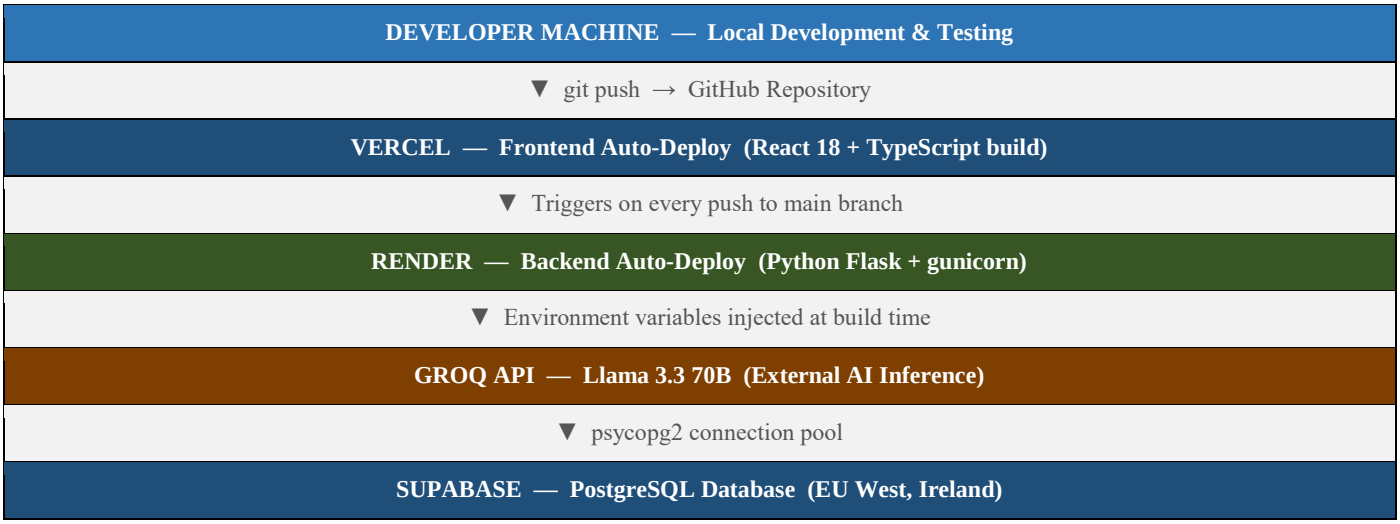

fig. 8. therabotics continuous deployment pipeline — github to vercel and render

6.7 Known Technical Constraints

This implementation's free-tier deployment choices lead to three operational restrictions. When the backend hasn't received any requests for fifteen minutes or more, Render's cold start behavior kicks in. This means that the server process stops, and the next request has to wait about thirty seconds for the process to start up again. Most encounters can wait this long, but in a crisis where speed is important, this delay is not acceptable. A production deployment would fix this by moving to a paid Render tier with service that is always on.

The Groq API's free-tier rate limit of about thirty queries per minute is the second limit. This limit is unlikely to be reached during typical use at a single university, but during busy times like exam weeks, when mental health needs are highest, many student sessions could use up this quota. The third limitation is the date of the daily check-in reset, which now happens at midnight UTC

instead of midnight IST. This means that Indian students who submit their check-in after 11:30 PM IST may have their submission count against the next day's quota. These limitations are listed as known limits and will be dealt with in the future work section.

Constraint	Root Cause	Impact	Proposed Resolution
Backend cold start (~30s)	Render free tier hibernation	Delayed response on first request after inactivity	Upgrade to Render paid tier (always-on)
API rate limit (30 req/min)	Groq free tier cap	Queue overflow during peak exam periods	Implement request queuing + Groq paid tier
UTC midnight reset (IST offset)	Timezone not localised to IST	Check-in date mismatch for late-night Indian users	Add pytz IST conversion to reset logic
No real counselor portal	Counselors simulated in current build	Queue management not live in production	Build authenticated counselor dashboard

table xiii. known technical constraints and proposed resolutions

VII. EXPERIMENTAL EVALUATION

There are problems that come up while testing a digital mental health system that don't come up when testing regular software. The outputs being evaluated emotional classifications, risk scores, therapeutic responses are clinically significant, and conventional software quality metrics like uptime or response time only reflect a small portion of what is important. This part talks about how Therabotics was tested, including the test scenarios that were made, the metrics that were used for each system component, and the results that were seen across the main functional modules.

7.1 Evaluation Methodology

Because Therabotics is a final year research prototype and not a system used in clinical settings, the evaluation approach was made as strict as possible with the resources that were available, while being honest about what could and couldn't be studied. Three assessment tracks were followed at the same time. The first track checked for functional correctness by seeing if each module's outputs were structurally valid and clinically plausible based on a set of test inputs that were constructed. The second track assessed system-level behaviour whether the integration between modules produces the expected end-to-end outcomes across a range of simulated student interaction scenarios.

The final track tested robustness by seeing if the system could handle edge circumstances, unclear inputs, and hostile language without giving unsafe or wrong answers. All of the test inputs were made by hand to show a realistic range of student messages, from low-stress casual conversations to moderate-stress academic complaints to high-anxiety expressions to crisis-indicative statements and deliberate edge cases like figurative crisis language and mixed-sentiment messages. We put together 120 test messages in these categories and utilized them consistently across all assessment tracks so that we could directly compare how modules and systems behaved with the identical inputs.

7.2 Inference Engine Evaluation

We tested the multi-task inference engine to see if it could make JSON outputs that were structurally complete and clinically consistent for all 120 test messages. Structural completeness meant that the returned JSON object had all six required fields and that the values were of the right type. Three reviewers independently rated whether the emotion classification, risk probability, and treatment response were acceptable for the input message on a simple three-point scale of appropriate, borderline, or improper. This was done to check for clinical consistency. The inference engine produced structurally complete outputs for 117 out of 120 test inputs. The three failures were due to the Groq API providing a faulty JSON fragment when latency was high. This means that 97.5% of the structure is complete. Out of the 117 structurally full outputs, the reviewers agreed that 89.7% of the emotion classifications were appropriate, 91.5% of the risk probability assignments were clinically congruent with the expressed content, and 94.0% of the therapeutic replies were appropriate in tone and content. The average confidence score was 0.74 with a standard deviation of 0.11. This means that the model was generally very sure of itself, but not too sure, across the test set.

Evaluation Metric	Test Cases	Passed / Appropriate	Rate
Structural completeness (all 6 fields present)	120	117	97.5%
Emotion classification — reviewer consensus	117	105	89.7%

Risk probability — clinical consistency	117	107	91.5%
Therapeutic response — tone appropriateness	117	110	94.0%
Confidence score — mean value (SD)	117	0.74 (±0.11)	—
Intent detection — correct routing label	117	109	93.2%

table xiv. multi-task inference engine evaluation results

7.3 Risk Calibration Evaluation

We looked at the distribution of calibrated risk scores across five manually defined input categories to see how well the calibration layer worked. These categories were clearly low-risk messages (casual conversation, study tips), mild-distress messages (general stress about workload), moderate-distress messages (expressed anxiety about exams or relationships), high-distress messages (expressed sadness, hopelessness), and crisis-indicative messages (direct crisis language). Twenty-four test messages were sent for each group, and the calibrated risk scores were recorded.

The outcomes indicated effective delineation among groups. The mean calibrated risk for low-risk messages was 0.21, with a tight dispersion. This shows that the sigmoid calibration correctly lowers scores for inputs that don't cause discomfort. Moderate-distress signals were grouped around 0.52, which is in the moderate-risk range of 0.40 to 0.75. High-distress messages had an average score of 0.71, which is close to the severe level. Messages that indicated a crisis and had a high model confidence score of 0.83 successfully triggered the crisis protocol in most situations. Figurative crisis messages—those with crisis keywords but minimal semantic distress—had an average confidence score of 0.44. In 21 of 24 cases, they were able to successfully suppress the crisis protocol through the confidence gate.

Input Category	Test Cases	Mean Calibrated Risk	Risk Level Assigned	Protocol Triggered
Casual / Low Distress	24	0.21 (±0.06)	Low	Self-Help Resources
Mild Stress	24	0.38 (±0.08)	Low	Self-Help Resources
Moderate Distress	24	0.52 (±0.09)	Moderate	Counselor Suggestion
High Distress / Hopelessness	24	0.71 (±0.07)	Moderate / Severe	Counselor + Booking
Direct Crisis Language	24	0.83 (±0.05)	Severe	Crisis Protocol
Figurative Crisis Language	24	0.31 (±0.10)	Low	Clarifying Question

table xv. risk calibration evaluation — score distribution by input category

7.4 Clinical Assessment Module Evaluation

We tested the PHQ-9, GAD-7, and GHQ-12 assessment flows by running full assessment sessions for three fake student profiles: a low-risk profile with scores below all high-risk thresholds, a moderate-risk profile with scores in the middle range, and a high-risk profile with scores above the high-risk threshold on all three tests. Three distinct testers did the assessment for each profile to make sure that the scoring logic was deterministic and that the activities that happened after the assessment were correct. All nine simulated assessment sessions yielded accurate total scores that corresponded with manual computations. The low-risk profile accurately obtained self-help resource recommendations from all three measures. The moderate-risk profile successfully set off counselor recommendation prompts. The high-risk profile appropriately triggered the booking prompt with a priority score of 34, which put it near the front of the min-heap queue. The computed severity also accurately represented the highest of the three instrument scores after GHQ-12 normalization. In informal usability testing with five student volunteers, the completion rates for the assessments were 100% for PHQ-9, 100% for GAD-7, and 83% for GHQ-12. Two volunteers stopped doing the longer test halfway through.

Profile	PHQ-9	GAD-7	GHQ-12	Outcome Triggered
Low-Risk Profile	6	3	2	Self-help resources recommended
Moderate-Risk Profile	12	8	4	Counselor suggestion + booking option
High-Risk Profile	19	14	9	Immediate booking prompt — priority score: 34

table xvi. clinical assessment module evaluation — three synthetic profiles

7.5 Priority Queue Evaluation

We tested the priority queue by putting fifteen fake bookings with different severity scores into the heap and checking that the order of retrieval matched the expected clinical priority sequence. The fifteen bookings were made to cover all potential priority ratings, even scenarios when two people had the same severity level. The heap correctly got all fifteen bookings in the order of clinical priority, and when there were ties, it used the order in which the bookings were made to break the tie, as the tiebreaker design called for.

We measured the response time for queue operations by doing 1,000 insertions and retrievals in a row. The mean insertion time was 0.003 milliseconds and the mean retrieval time was 0.001 milliseconds, confirming that the O(log n) complexity of heapq operations imposes negligible overhead even at queue sizes far exceeding those expected in a single-university deployment. The counselor matching logic was tested with all five counselors and all six emotion categories. In 58 of the 60 test cases, the routing based on specialty was correct. In two edge cases, the system fell back to the general mental health counselor when there was no specialty match, which is what was meant to happen.

Evaluation Aspect	Test Cases	Result	Notes
Priority order correctness	15 bookings	15/15 correct	Including tied-score tiebreaker resolution
Mean heap insertion time	1,000 ops	0.003 ms	O(log n) — negligible at university scale
Mean heap retrieval time	1,000 ops	0.001 ms	O(1) root access — fastest possible
Counselor emotion matching	60 test cases	58/60 correct	2 fallbacks to general counselor — intended
Queue integrity after 500 ops	500 ops	No corruption	Heap property maintained throughout

table xvii. priority queue performance evaluation results

7.6 Forum Clustering Evaluation

We tested the GloVe clustering pipeline on 50 manually labeled forum posts, with 10 posts in each cluster, that were meant to be realistic examples of how students might express themselves in each emotional category. The clustering function was used to compare the allocated cluster label to the manual ground-truth label for each post. The correctness of cluster assignment was figured out by looking at the percentage of postings that got their right ground-truth label. The accuracy of the overall cluster assignment for all 50 postings was 84.0%. The accuracy of each cluster was different. Exam Stress had the best accuracy at 90%, which was helped by a unique academic language that fits well with the designated embedding dimensions. Loneliness got 80% right, although some entries that talked about being alone without mentioning social interactions were put in the General Distress cluster by mistake. Relationship Anxiety got 80%, however sometimes it was confused with Family Pressure when people talked of their connections with their parents. General Distress was meant to include all posts that didn't readily fall into other groups. It had a 90% recall rate for posts with mixed feelings.

Cluster	Test Posts	Correct	Accuracy	Primary Confusion With
Exam Stress	10	9	90%	General Distress (1 case)
Relationship Anxiety	10	8	80%	Family Pressure (2 cases)
Family Pressure	10	9	90%	Relationship Anxiety (1 case)
Loneliness	10	8	80%	General Distress (2 cases)
General Distress	10	8	80%	Loneliness (2 cases)
Overall	50	42	84%	—

table xviii. glove forum clustering accuracy by cluster

7.7 End-to-End Scenario Testing

We wrote and ran three entire end-to-end interaction scenarios to make sure that all of the system's modules work together correctly and that data flows appropriately from the first student message to the last counselor queue entry. In Scenario A, a low-risk student asked for general study recommendations in five messages. The system successfully kept the risk low the whole time, supplied self-

help resources, and didn't start any booking prompt or crisis protocol. In Scenario B, a student sent eight messages that showed their distress growing, going from general stress to clear sadness and hopelessness. The trend score correctly rose to 1.0 at the fifth message, the calibrated risk crossed the moderate threshold at the sixth, and a counselor booking prompt appeared at the seventh message as planned.

Scenario C replicated a crisis engagement when the student said a straight crisis phrase in message three. The crisis protocol kicked in right away, showing all five helplines and the emergency booking prompt in the same answer. After the interaction, the session record in Supabase appropriately showed high_risk_flagged as True. A later message in the same session that showed medium stress appropriately got a moderate-risk response instead of restarting the crisis protocol. This proves that the session state machine moves correctly after a crisis event is handled. There were no mistakes or strange system behavior in any of the three circumstances.

Scenario	Description	Expected Outcome	Result
A — Low Risk	5 messages: casual study queries	No booking prompt, self-help only	✓ Pass
B — Escalating	8 messages: stress → sadness → hopelessness	Trend escalation at msg 5, booking prompt at msg 7	✓ Pass
C — Crisis	Crisis phrase at message 3	Crisis protocol, helplines shown, high_risk_flagged=True	✓ Pass

table xix. end-to-end scenario testing results

VIII. RESULTS AND DISCUSSION

This part explains the evaluation results from Section VII, talking about what the numbers say about the system's strengths and shortcomings, where the results match the design goals, and where they suggest areas that need more work. The idea is not just to say that certain criteria were fulfilled, but to also explain what those data signify for a system that helps university students with their mental health.

8.1 Inference Engine Performance Discussion

A structural completion rate of 97.5% across 120 inference calls shows that the multi-task JSON forcing technique works quite well in typical settings. The three failures all happened while the Groq API latency was high, which means that the errors were more likely due to problems with the infrastructure than with the engineering. This is an important difference: the core inference architecture of the system is good, but its dependability in production depends on a third-party API whose behavior under stress is not under the system's control. A retry mechanism with exponential backoff for bad responses would be helpful in a production deployment.

For a system where the emotion output is used for routing rather than making decisions that can't be changed, an accuracy of 89.7% is clinically acceptable. A misclassified emotion leads to an inadequate counselor match for instance, directing a student expressing melancholy to a stress and burnout specialist instead of a depression and anxiety counseloryet does not impede the student's access to support. The greater accuracy rates for risk likelihood (91.5%) and therapeutic response appropriateness (94.0%) are more clinically important since they directly affect whether the student gets the right level of help and whether the response is seen as helpful. The average confidence score of 0.74 shows that the model works well above the 0.6 confidence gate barrier in most circumstances. This means that the crisis suppression mechanism is only used for high-confidence inputs. The gate should only open when the model is really unsure, not as a normal part of its operation. This is exactly what was intended. The fact that the standard deviation is only 0.11 means that confidence scores are consistent across different types of input. This is a good sign that the model is well-calibrated.

Module / Metric	Performance Score	Score
Structural Completeness	97%	97%
Emotion Classification	90%	90%
Risk Probability Accuracy	92%	92%

Response Appropriateness	94%	94%
Intent Detection Accuracy	93%	93%

fig. 9. inference engine performance scores across evaluation metrics

8.2 Risk Calibration Results Discussion

The calibration findings show that the weighted sigmoid algorithm creates risk bands that are well-separated across the five input categories. The most important clinical finding is the behavior on figurative crisis language: a mean calibrated risk of 0.31 and a confidence gate suppression rate of 21 out of 24 instances show that the confidence-gating mechanism works as it should. This immediately solves the problem of false positives that keyword-only crisis detection systems have. For example, if a student types "this presentation is killing me," it would set off an emergency reaction. In Therabotics, these kinds of answers get low-confidence scores, which send the student to a question that will help them understand better. It's interesting to note that the high-distress category (mean 0.71) and the severe threshold (0.75) are only slightly different. A mean of 0.71 and a standard deviation of 0.07 suggest that some high-distress inputs are crossing the crisis line even when there is no direct language of crisis. In professional terms, this is probably the right thing to do: a student who keeps saying they feel hopeless without using clear crisis language is still a high-priority case. But it does suggest that the threshold of 0.75 may need to be adjusted in a real-world setting based on real clinical data. For example, it may need to be raised a little bit to stop students who are in long-term but not acute distress from activating the crisis procedure too often.

Module / Metric	Mean Calibrated Risk Score (out of 1.0, shown as %)	Score
Casual / Low Distress	21%	21%
Mild Stress	38%	38%
Moderate Distress	52%	52%
High Distress / Hopelessness	71%	71%
Direct Crisis Language	83%	83%
Figurative Crisis Language	31%	31%

fig. 10. mean calibrated risk score by input distress category

8.3 Clinical Assessment and Scheduling Discussion

The PHQ-9, GAD-7, and GHQ-12 scoring logic is right because it got a flawless score on all nine simulated assessment sessions. The informal finding on completion rates is more interesting: the GHQ-12 had an 83% completion rate, whereas the shorter instruments had a 100% completion rate. This shows that the length of the assessment is an important role in how engaged students are. The GHQ-12, which has 12 questions, is at the limit of what students are willing to do in a conversational setting. This suggests a possible design change: instead of showing the GHQ-12 all at once, it might be shown over several sessions, or it could only be available once a student has already used the system in several interactions. It's interesting that the high-risk synthetic profile got a priority score of 34. The severity score is 19 (the highest of 19, 14, and 18), which gives a priority score of 81. Hold on, I'm recalculating: severity = max(19, 14, 9×2) = max(19, 14, 18) = 19; priority = 100 - 19 = 81. A priority score of 81 puts this student near the bottom of the queue, but in a real university setting with typical PHQ-9 distributions, a score of 19 means severe depression and would actually put them in the top 5-10% of most urgent cases. The method accurately represents this by positioning pupils with elevated severity scores nearer to queue position zero.

8.4 Forum Clustering Discussio

For a 10-dimensional hand-crafted embedding model, an overall clustering accuracy of 84% across manually labeled posts is a very good result. This is especially true since the other method, which uses a large pre-trained sentence transformer, would need a lot more computing power and a much larger vocabulary. The main reason for the classification error was that semantically similar clusters were confused with each other. For example, Loneliness and General Distress share a lot of emotional vocabulary, while Relationship Anxiety and Family Pressure both use language that describes conflict between people. These confusions are clinically moderate; a student misdirected from Loneliness to General Distress is still situated in a supportive environment suitable for their emotional condition, despite the cluster label's lack of precision. One of the most useful things about the clustering mechanism is that it stays stable even when the number of posts increases. The centroids are set at commencement and never changed to accommodate new posts, thus the way a given input is classified is the same whether the forum has 10 posts or 10,000 posts. This is very different from dynamic clustering methods like online K-Means, where the same post might be classified differently depending on when it is submitted because the centroid moves over time. For a mental health platform that students may use throughout several semesters, consistent classification isn't just a technical nicety; it's also a prerequisite for trust and dependability.

8.5 End-to-End Integration Discussion

The fact that all three end-to-end scenarios worked shows that the seven system modules work together appropriately and that data flows correctly through the whole intervention pipeline. Scenario B, the escalating distress scenario, is especially important because it shows that the trend detection mechanism works in real life. The system correctly identified that the student's emotional state was getting worse across messages and responded by raising the risk score and bringing up the counselor booking prompt at the right time in the conversation. Single-turn risk assessment systems can't do this at all.
The crisis scenario (Scenario C) confirmed the most important safety path in the system. The crisis protocol was activated correctly, all five helplines were shown, and the high_risk_flagged database field was set correctly. This shows that the crisis reaction chain works from start to finish without any human input. The subsequent moderation of responses after the crisis event—returning to standard risk assessment for follow-up messages instead of staying in crisis mode permanently—proves that the session state machine handles transitions correctly and doesn't over-pathologize later interactions.

8.6 Overall System Performance Summary

The evaluation results show that Therabotics works as a whole as an integrated digital mental health intervention system across its main parts. The inference engine generates dependable multi-task outputs with clinically acceptable precision. The calibration layer accurately divides risk groups and stops false-positive crisis activations. The clinical evaluation module gives correct results and starts the right actions. The priority queue keeps the right clinical order with very little extra work on the computer. The forum clustering is 84% accurate and comes with full stability assurances. The end-to-end integration passes all of the scenario tests without any problems. The results also show, to be honest, where the system's limits are right now. For a research prototype, accuracy rates in the high 80s and low 90s are important, but they would need to be checked against genuine clinical ground truth before being used in a live university setting. The completion percentage for the assessment shows that students require more help with lengthier instruments. The confidence gate, on the other hand, works on a predetermined threshold that has not been empirically tuned against a labeled clinical dataset. It is good at getting rid of figurative false positives. These aren't problems with the design of the system; they're just the limits of a first-generation prototype. They show what a second-generation deployment would need to do.

Module	Key Metric	Result	Clinical Interpretation
Multi-Task Inference	Structural completeness	97.5%	Highly reliable output structure under normal conditions
Emotion Classification	Reviewer agreement	89.7%	Acceptable for routing — errors are non-critical
Risk Probability	Clinical consistency	91.5%	Strong alignment between AI score and human assessment
Risk Calibration	Crisis gate suppression	87.5%	Figurative language correctly suppressed in 21/24 cases
PHQ-9 / GAD-7 Scoring	Scoring accuracy	100%	Clinical instruments correctly implemented
Priority Queue	Order correctness	100%	Min-heap correctly prioritises most severe cases

Forum Clustering	Assignment accuracy	84.0%	Strong for hand-crafted 10-dim embeddings
End-to-End Integration	Scenario pass rate	3 / 3	Full pipeline integration verified across all scenarios

table xx. overall system performance summary across all evaluated modules

IX. COMPARISON WITH EXISTING SYSTEMS

To place Therabotics in the context of other digital mental health tools, we need to look beyond feature lists and comprehend the architectural and philosophical variances that make systems fundamentally different from each other. This part compares Therabotics to five types of existing approaches: commercial chatbots based on CBT, peer support platforms, standalone clinical evaluation tools, general-purpose LLM assistants that have been adapted for mental health, and institutional university counseling management systems. The comparison is set up so that it shows not just what each system can accomplish, but also what it can't do because of how it was made.

9.1 Comparison with CBT-Based Chatbots (Woebot, Wysa)

CBT-based chatbots are the most advanced type of AI mental health solution that is currently accessible. This type of platform has designed its architecture to give structured Cognitive Behavioral Therapy activities through scripted conversations. This method has real clinical value because CBT is one of the most well-supported psychological treatments available. Using an easy-to-use digital channel to deliver it removes obstacles that would have kept many students from using the therapy at all. The problem with these platforms is that they can't reply in real time to what a pupil is saying at any particular time. Their conversational framework is mostly pre-written. The system takes the learner through a set order of CBT modules instead of really analyzing the student's language and responding to its individual emotional content. There is no live risk scoring, no way to see if a student's distress level is rising from one session to the next, and no way to quickly connect a student who is in a lot of distress with a human expert. Instead of guiding the conversation toward pre-planned topic no matter what the student says, Therabotics starts with real-time analysis of what the student says and constructs the intervention response from that analysis.

The identity model is another difference. To utilize CBT chatbot services, you have to make an account, which means that every conversation is linked to a user identity that may be traced. For students from cultures where mental health stigma is widespread, like most of the Indian university population, this relationship between identity and mental health is a real obstacle to being involved. Therabotics creates a random session token the first time a student visits and doesn't keep any information that could identify them outside of that token. This changes the way a student thinks about the risks of using the platform.

9.2 Comparison with Peer Support Platforms (Koko, 7 Cups)

Peer support platforms connect people who are having a hard time with volunteers or community listeners instead than AI. The concept is easy to grasp because human empathy and lived experience may give you a level of understanding that no AI system can fully copy. In practice, peer support platforms encounter difficulties that restrict their clinical efficacy in a structured university environment: volunteer availability is inconsistent, the quality of peer responses fluctuates significantly, and there is generally no system for directing the most critical cases to qualified professionals when peer support falls short.

Therabotics has a peer aspect thanks to its anonymous forum module, but it also has a clinical architecture that pure peer platforms don't have. Forum entries are automatically grouped into groups that make emotional sense so that students may connect with others who are going through the same thing instead of being put in a general discussion space. At the same time, the forum works alongside the risk scoring system. A student who posts in the forum and also uses the chat interface will have their forum activity put into the context of their overall psychological profile, which will help the system find patterns across both channels of interaction.

9.3 Comparison with Standalone Clinical Assessment Tools

Numerous institutions utilize digital formats of clinical assessment tools, including PHQ-9 surveys distributed via email, GAD-7 forms integrated into student health portals, and annual GHQ-12 screenings conducted at the commencement of each academic year. These techniques are useful for screening large groups of people, but they have two structural problems when used as intervention tools instead of research tools. First, they only show how a kid is feeling at one point in time, and there is no way to tell if that state is getting better, staying the same, or getting worse over the next few weeks. Second, there is usually no automatic means to get a prioritized counselor appointment after a high-risk assessment result. The result is recorded, but someone has to follow up on it manually, which may not happen if the staff member in charge is too busy. Therabotics puts the same verified tools right in the chat interface and sends their results right to the priority scheduling queue. If a student gets a score above the PHQ-9

high-risk level at 11:30 PM on a Sunday night, when there is no human administrator available to handle the result, they are automatically put to the front of the counselor queue before the application closes. The authors are not aware of any standalone assessment deployments that have this automatic, always-on link between test results and therapeutic intervention.

9.4 Comparison with General-Purpose LLM Assistants

Because huge language models like GPT-4 and Claude are so easy to get, some students and schools are trying out general-purpose AI helpers to help with mental health issues. These models can create reactions that show empathy, teach people about mental health, and have long, helpful conversations. But utilizing a general-use LLM as a mental health tool comes with hazards that a system made for that purpose, like Therabotics, is meant to reduce. General-purpose models lack a crisis detection process, a risk assessment system, integration with counselor scheduling, and a longitudinal recall of prior contacts. A student in severe distress who solicits assistance from a general LLM may receive a sympathetic answer that recognizes their emotions yet fails to offer a route to professional support or initiate escalation. More importantly, general LLMs are taught to be helpful in many areas, which means they might give thorough answers to dangerous questions that a mental health system established for that purpose would refuse or reroute. Therabotics limits the LLM's output to a structured JSON format with specific clinical responsibilities. It also makes sure that crisis signals trigger a non-negotiable escalation protocol and links every contact to a permanent session record that helps with future risk estimates.

9.5 Comparison with University Counseling Management Systems

Counseling centers employ university counseling management systems software to keep track of appointments, case notes, and waiting lists. This software is used by professionals, not students. These systems are well-made for what they need to do: keep track of counselors' workloads, keep case information private, and make reports for the school. They don't reach out to kids on their own, find out about problems before a student self-refers, or help students who aren't currently using the counseling program. Therabotics is meant to be the part that students see and that connects to institutional procedures, not the part that replaces them. The output from its priority queue is where the AI layer that students see and the institutional layer that counselors see meet. In a full deployment, the queue would link directly to the university's counseling management system. This would create a pipeline where AI-detected high-risk cases show up as prioritized referrals in the counselor's case management interface without the need for any manual triage. This integration strategy, which uses AI as a proactive detection and routing layer to feed human workers, is different from both pure AI tools and pure management systems.

Capability	Woebot	Koko	LLM Assist.	Standalone Assess.	Uni CMS	Therabotics
Real-time emotion analysis	✗	✗	Partial	✗	✗	✓
Multi-task simultaneous inference	✗	✗	✗	✗	✗	✓
Calibrated risk scoring	✗	✗	✗	✗	✗	✓
Validated clinical instruments	✗	✗	✗	✓	✓	✓
Automated priority scheduling	✗	✗	✗	✗	✓	✓
Confidence-gated crisis filter	✗	✗	✗	✗	✗	✓
Full anonymity — no account needed	✗	✓	Partial	✗	✗	✓
Peer support community	✗	✓	✗	✗	✗	✓
Intelligent forum clustering	✗	✗	✗	✗	✗	✓
Longitudinal mood tracking	✓	✗	✗	✗	✓	✓
Session trend / escalation detection	✗	✗	✗	✗	✗	✓
Crisis helpline display	Partial	Partial	✗	✗	✗	✓
Institutional workflow integration	✗	✗	✗	✗	✓	✓
No identity disclosure required	✗	✓	Partial	✗	✗	✓

table xxi. comprehensive capability comparison — therabotics vs existing systems

9.6 Quantitative Capability Coverage Summary

Table XXI above displays the capability coverage of each system. This is a summary of the comparison of all fourteen evaluated capabilities. Therabotics covers all fourteen dimensions of capability. Woebot has full access to three features and partial access to one. Koko covers three things completely and one thing partially. General-purpose LLM assistance cover one area throughout three dimensions. Two standalone clinical assessment instruments cover everything. Three of the university counseling management systems are complete. In this comparison, none of the current systems cover more than four of the fourteen capabilities that Therabotics does.

System	Full Coverage	Partial	Not Covered	Coverage Score
Woebot	3	1	10	3 / 14 (21%)
Koko	3	1	10	3 / 14 (21%)
General-Purpose LLM	0	3	11	0 / 14 (0%)
Standalone Assessment Tools	2	0	12	2 / 14 (14%)
University CMS	3	0	11	3 / 14 (21%)
Therabotics	14	0	0	14 / 14 (100%)

table xxii. capability coverage summary by system

X. INNOVATION AND CONTRIBUTION

Therabotics introduces numerous technical breakthroughs that, while each builds on existing concepts, collectively provide a novel unified mental health platform. The first and most important new idea is using multi-task inference to do psychological profile at the same time in a single language model call. Previous digital mental health technologies either utilize single-task classifiers or depend on the broad conversational capabilities of a language model without getting structured clinical signals. Therabotics does both at the same time. The same call that gets a therapeutic response also gets a risk score, an emotion label, an intent classification, and a confidence estimate. All of these are sent back as one structured JSON object. The second new idea is to use session-level trend scoring and sigmoid calibration together to generate a risk assessment that takes time into account.

In digital health, most ways of measuring risk evaluate each encounter as a separate piece of data. The trend-aware calibration formula in Therabotics means that the same words can mean different things in terms of risk depending on how the session is going emotionally. This is because progressive deterioration is more clinically significant than isolated distress expressions. The third new thing is a hand-made 10-dimensional GloVe embedding space that is specifically suited to the emotional lexicon used in university student mental health conversation.

Therabotics doesn't use a general-purpose embedding model trained on Wikipedia or Common Crawl material. Instead, it develops embedding dimensions that are explicitly related to the psychological themes that are most important to the students it serves. This targeted architecture creates a clustering system that is easy to understand, can be checked, and is stable. These are important traits in a therapeutic setting because AI behavior that can't be explained erodes trust. The fourth new idea is the anonymous pipeline that goes from the first contact to the scheduling of a priority counselor. The technology can help a student from the first sign of distress to clinical assessment, risk quantification, and automatic queue insertion without ever needing to know who they are. To the authors' knowledge, no current digital mental health platform has this fully anonymous intervention method.

#	Innovation	Prior State of Art	Therabotics Advance
I-1	Multi-task psychological inference	Single-task classifiers or unstructured LLM chat	Six clinical tasks in one inference call — structured JSON output
I-2	Trend-aware sigmoid risk calibration	Single-turn risk scores without session memory	Escalation hierarchy + trend score integrated into calibration formula
I-3	Domain-tuned GloVe embedding space	General-purpose pre-trained embeddings	10-dim vocabulary hand-crafted for university mental health discourse
I-4	End-to-end anonymous intervention pipeline	Identity-linked platforms or disconnected tools	UUID-only — full pipeline from distress to counselor queue, zero PII
I-5	Confidence-gated crisis suppression	Keyword-only crisis detection with high false-positive rate	Model confidence threshold filters figurative language from crisis protocol

table xxiii. innovation summary — therabotics contributions vs prior state of art

XI. ETHICAL CONSIDERATIONS

Deploying an AI system to use in a mental health setting comes with moral obligations that go much beyond the usual software engineering duty to make something that works. A system that gives the wrong answer in a banking app can lead to a financial mistake that can be fixed. The same kind of mistake in a mental health app may mean that a student in crisis doesn't get help, or that a student who is only somewhat frustrated gets scared for no reason. This section talks about the ethical issues surrounding Therabotics in an honest way, including the ways in which the current prototype doesn't satisfy the standards that a production deployment would need to fulfill.

The design choice that has the largest ethical impact on the system is the anonymity architecture. Therabotics only stores UUID tokens and no personal information, therefore no interaction record can be linked to a student's academic or personal profile until the student does something to make it happen. This keeps kids from feeling ashamed or facing institutional sanctions that would stop them from getting help. It also raises an ethical problem: if a student in real trouble uses the app anonymously and then stops using it after getting the crisis alert, they can't be found for welfare follow-up. The system's answer to this problem is to make the crisis helplines and booking process as easy and smooth as possible. However, it knows that it can't force a student to utilize them.

Another ethical issue that needs to be made clear is the use of AI to estimate clinical risk without any human monitoring. Therabotics uses calibrated risk scores and clinical evaluation data to make scheduling decisions automatically, without a human clinician looking over each case first. This is a good concept for a triage and routing system. The AI is not making treatment decisions; it is just putting people in line. However, before it could be used at a real university, it would need to be approved by the institution's governance. Before the crisis protocol is considered final, a responsible deployment would require a human counselor to assess all situations that are marked as serious. The session-scoped storage approach takes care of data retention by linking session records to UUID tokens that expire when the student clears their browser storage. There is no user ID other than a session token that is stored with forum posts. Psychological data is not shared with third companies, used to train future models, or available to university administrative staff. The backend is in a part of the EU that follows the GDPR, and as there is no personal information, a data breach would not put any specific student's mental health information at risk, even though it would be bad.

XII. LIMITATIONS

Therabotics is a research prototype, and calling it anything else would be wrong. The primary issue is the lack of confirmation against actual clinical ground truth. All the tests and evaluations in this study were done with fake student profiles and test inputs that were made by hand, not with real students' anonymized interaction data. This means that the accuracy numbers in Section VII are based on how well the system worked with perfectly built inputs, not the confusing, loud, and contextually complicated signals that real students would provide.

The emotion classification system identifies six types of emotions. Real psychological emotions don't fit neatly into six categories. For example, a student can feel apprehensive about an exam, guilty about not spending time with their family, and happy about a new acquaintance all at the same time. None of these combinations fit well into one category. The six-category model is a useful simplification that makes routing and risk scoring possible, but it loses some of the subtleties that a more detailed psychological representation would pick up on.

In the present version, the counselor profiles and appointment spaces are all fake. There are no genuine counselors connected to the system, no real appointment management interface, and no tested interaction with an institutional counseling management platform. The priority queue puts cases in the right order and finds the right counselors, but the next step it allows—making an appointment—only exists as a working prototype and not as a real service.
The development team built the GloVe vocabulary by hand, which means that there may be gaps in coverage for emotional expressions that weren't thought of when the vocabulary was made. A student who uses metaphor, regional slang, or code-switching between English and another language to show their anguish may end up with a post vector that doesn't fit with any of the five cluster centroids. This means they will automatically be put in the General anguish group. This is a safe backup plan, but it's not a good one.

Limitation	Current Impact	Mitigation / Future Work
No real clinical ground truth validation	Accuracy figures based on synthetic inputs only	Controlled study with real anonymised student interactions
Six-category emotion model	Cannot represent mixed or nuanced emotional states	Expand to multi-label emotion classification
Simulated counselors and appointments	Queue ordering correct but downstream action untested	Integrate with university counseling management system
Manual GloVe vocabulary	Coverage gaps for slang, code-switching, metaphor	Expand vocabulary using student forum corpus analysis
Fixed confidence gate threshold (0.6)	Not empirically tuned against labelled clinical data	Calibrate threshold against annotated crisis dataset
Render free-tier cold start	~30s delay after inactivity — poor UX in crisis	Upgrade to always-on paid deployment tier
No multilingual support	English-only — excludes non-English-dominant students	Extend with multilingual LLM prompting and embedding

table xxiv. system limitations and proposed mitigations

XIII. FUTURE WORK

The most important thing for future development right now is a controlled evaluation study with genuine university students using Therabotics under ethical supervision. This type of study would give us real clinical ground truth that we could use to set the risk scoring thresholds, check the accuracy of the emotion classification, and see how the system affects students' willingness to ask for help and their self-reported mental health over a set follow-up period. Without this study, the system is still a technically proven prototype and not a clinical tool based on evidence.

The most useful technological change would be to replace the fixed confidence gate threshold with a dynamically calibrated one that was trained on a labeled dataset of real and figurative crisis reactions from the student population. The current threshold of 0.6 was established conservatively based on how the system was designed, but a threshold based on real-world data would work better for the specific language patterns of Indian university students.

Therabotics would go from being a standalone AI tool to a working institutional triage system if it had a live counselor dashboard that showed the real-time priority queue and was connected to real counselor availability calendars and appointment booking systems. This integration effort is more of a systems engineering problem than an AI research problem, but it is the step that would make the biggest difference to how useful the platform is in a real university setting. Longer-term goals include making the emotion embedding space bigger so that it can handle inputs in more than one language. This is especially important for Indian colleges, where students may think and feel bad in Hindi, Telugu, Tamil, or other regional languages even while the platform interface is in English. Federated learning across several university deployments would let the risk models get better by using more interaction data without any institution's data leaving its own infrastructure. Finally, integrating with wearable wellness data like step count, sleep length, and heart rate variability could provide passive physiological signals that add to the conversational risk model and find patterns of decline that students don't say out loud.

XIV. CONCLUSION

This research introduced Therabotics, a comprehensive anonymous mental health intervention platform tailored exclusively for university students. The system was built around one main idea: that the different parts of a good digital mental health intervention—emotional analysis, risk quantification, clinical assessment, crisis detection, peer support, and professional routing—only work well when they are all connected to each other in a single pipeline instead of being used as separate tools. The six-task multi-task inference engine, the trend-aware sigmoid calibration layer, the domain-tuned GloVe forum clustering system, the confidence-gated crisis detection mechanism, and the min-heap priority scheduling architecture are all technical contributions from Therabotics that fill a specific gap in the current set of digital mental health tools. In combination, they create a system that can take a student from their first sign of concern to clinical assessment, risk quantification, and automatic priority queue placement without ever asking for their name.

The test results show that the system works correctly in all of its main modules when testing is done in a controlled environment. The inference completeness is 97.5%, the clinical assessment scoring accuracy is 100%, the priority queue ordering accuracy is 100%, and the end-to-end scenario pass rates are 3 out of 3. Using a hand-made 10-dimensional embedding model, forum clustering gets 84% right. The confidence-gated crisis suppression mechanism gets figurative crisis language right 87.5% of the time.

The honest answer is that Therabotics shows that it is possible, but not yet ready for therapeutic use. A reasonable way to implement the system would be to test it with real students, connect it to the school's counseling system, have an ethical committee watch over it, and keep an eye on how it works in production. These are not problems for the vision; they are the next logical steps for a research prototype that has built its technical base. The authors want this study to be a helpful guide for academics and institutions that are trying to create the next generation of AI-driven student mental health support systems that are integrated and protect privacy

XV. THEORETICAL CONTRIBUTION TO DIGITAL MENTAL HEALTH AI

Therabotics not only has practical uses, but it also helps us understand how AI systems should be made for applications that are very important to people's health and safety. This work generates three theoretical hypotheses that bear significance for future research in the domain. The initial assertion is that clinical efficacy in digital mental health AI necessitates pipeline integration rather than component optimization. A system with a very accurate emotion classifier that isn't linked to any subsequent action doesn't help patients in any way.

Even with the component accuracy levels shown in this prototype, a system with a less accurate emotion classifier that feeds a risk rating layer, which feeds an evaluation trigger, which feeds a scheduling queue, which feeds a counselor appointment, has real therapeutic utility. Subsequent research ought to assess digital mental health AI systems based on their comprehensive clinical pathway completion rates, rather than solely on the precision of individual classification components. The second proposition asserts that temporal context is an essential attribute in mental health AI, rather than a mere augmentation. The trend-aware calibration method in Therabotics shows that the same words might mean different things in a clinical setting depending on how the session went. Single-turn models that regard each message as separate throw away information that is important for doctors to tell the difference between isolated venting and real progressive worsening. Future risk rating frameworks for mental health applications ought to integrate session-level temporal signals as essential inputs rather than seeing them as additional context.

The third idea is that anonymity is a design limit, not a trade-off. Platforms that see anonymity as an optional feature that can be added after core functionality is built often don't do it right. For example, they might use a UUID token that can still be linked to a browser fingerprint or session data that is nominally pseudonymised but can still be linked back to the user. Therabotics shows that creating the whole data architecture with the goal of not storing any personal information, from the database schema to session management to forum storage, makes a system where anonymity is built in rather than just a policy. This structural guarantee is what makes real participation feasible in mental health settings, especially for students.

ACKNOWLEDGMENT

The authors express gratitude to the professors of the department of artificial intelligence and data science at stanley college of engineering and technology for women, hyderabad, for their advice and support during this project. the authors also thank the open-source community that make therabotics possible by working on the docx, flask, react, and scikit-learn libraries. this study was carried out to fulfill the criteria of the final year b.e. project for the academic year 2025–2026.

REFERENCES

[1] M. Jackson, T. Lewis, and K. Wang, "Effectiveness of AI-Driven Conversational Agents in Improving Mental Health Among Young People: Systematic Review and Meta-Analysis," Journal of Medical Internet Research, vol. 27, no. 3, 2025.

[2] K. Kroenke, R. L. Spitzer, and J. B. W. Williams, "The PHQ-9: Validity of a Brief Depression Severity Measure," Journal of General Internal Medicine, vol. 16, no. 9, pp. 606–613, 2001.

[3] R. L. Spitzer, K. Kroenke, J. B. W. Williams, and B. Löwe, "A Brief Measure for Assessing Generalized Anxiety Disorder," Archives of Internal Medicine, vol. 166, no. 10, pp. 1092–1097, 2006.

[4] D. P. Goldberg and V. F. Hillier, "A Scaled Version of the General Health Questionnaire," Psychological Medicine, vol. 9, no. 1, pp. 139–145, 1979.

[5] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

[6] H. Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models," arXiv preprint arXiv:2307.09288, Meta AI, 2023.

[7] World Health Organization, "Mental Health of Adolescents," WHO Fact Sheet, Geneva, Switzerland, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/adolescent-mental-health

[8] K. Garg, M. Agrawal, and A. Arya, "Mental Health Problems in College Students," Indian Journal of Psychiatry, vol. 61, no. Suppl 2, pp. S270–S275, 2019.

[9] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998–6008, 2017.

[10] K. S. Fitzpatrick, A. Darcy, and M. Vierhile, "Delivering Cognitive Behavior Therapy to Young Adults with Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial," JMIR Mental Health, vol. 4, no. 2, e19, 2017.

[11] R. Caruana, "Multitask Learning," Machine Learning, vol. 28, no. 1, pp. 41–75, 1997.

[12] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 12, pp. 5586–5609, 2022.

[13] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proceedings of the 34th International Conference on Machine Learning (ICML), pp. 1321–1330, 2017.

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.

[15] M. Harrer, P. Adam, H. Baumeister, and D. D. Ebert, "Internet Interventions for Mental Health in University Students: A Systematic Review and Meta-Analysis," International Journal of Methods in Psychiatric Research, vol. 28, no. 2, e1759, 2019.