MindLift: An Intelligent Role-Based Digital Mental Wellness Care and Consultation Platform

by

Shoumik Barua Shoumo
ID: CSE 2301028101

Md. Arif Babu
ID: CSE 2301028119

Md. Samir Alam
ID: CSE2301028030

Nazmus Sakib Chowdhury
ID: CSE2301028038

Supervised by
Tanjil Mahmud

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2026

APPROVAL

The project titled "**MindLift: An Intelligent Role-Based Digital Mental Wellness Care and Consultation Platform**" submitted by Shoumik Barua Shoumo (CSE2301028101), Md. Arif Babu (CSE2301028119), Md Samir Alam (CSE2301028030) and Nazmus Sakib Chowdhury (CSE2301028038) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Tanjil Mahmud
Lecturer & Assistant Coordinator,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Tanjil Mahmud, Lecturer** and **Asst. Coordinator,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Tanjil Mahmud)
Supervisor

Signature

Shoumik Barua Shoumo
ID: CSE2301028101

Md. Arif Babu
ID: CSE2301028119

Nazmus Sakib Chowdhury
ID: CSE2301028038

Md. Samir Alam
ID: CSE2301028030

ABSTRACT

In spite of increasing priority given to stress-related disorders such as depression and anxiety throughout the world, the mental health services that people get in Bangladesh are still in their rudimentary stage. People in our country either gravitate toward informal counseling or, worse, neglect mental soundness altogether. Even if a student in Dhaka, or a jobholder from Rajshahi overcomes social stigma and decides to book a professional consultation, they can't get the satisfactory and continuing support they hoped for. This project proposes a Mental Wellness Care and Consultation Platform named MindLift that fills the aforesaid voids. Building a role-based platform where patients, doctors, and administrators interact within a well-protected digital environment is the fundamental objective of this system. Patients can fill up questionnaires for mental wellness assessment, get doctor recommendations on the basis of their needs, schedule appointments with email OTP confirmation, attend online consultations and continue their care by completing wellness tasks that their doctors assign. Meanwhile, doctors have a much easier time tracking a patient's past sessions, keeping track of appointments or writing prescriptions. Everything is in one place, not scattered across paper files. In contrast, administrators are responsible for the overall platform operation by managing users, monitoring appointments, payment records and moderating platform content. The platform further supports continuity of care by allowing doctors to assign follow-up wellness tasks and patients to track them as part of their day-to-day mental wellness activities. The emergency support feature serves a different purpose by providing AI-assisted mental wellness support through a contextual Retrieval-Augmented Generation (RAG) approach, together with immediate-help resources intended for situations where timely guidance can matter. The platform also provides mental wellness articles, semantic blog search, and AI-assisted quick summaries to help users discover relevant information more effectively. When it comes to the technical side, the platform is developed using Django, HTML, CSS, SQLite, and JavaScript. Unlike generic appointment systems, this project encompasses continuous well-being support through assessment, personalized doctor recommendation, consultation, digital prescription, follow-up wellness tasks, and AI-assisted mental wellness guidance to make mental wellness a day-to-day practice. This platform, on the whole, aspires to bring forth a practical, tailored, coordinated, and reliable digital mental wellness service that can cater to the varying needs of patients.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Tanjil Mahmud**, Lecturer and Asst. Coordinator, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to our parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

CPU	Central Processing Unit
RAM	Random Access Memory
ROM	Read Only Memory
SDT	Self-Determination Theory
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SQL	Structured Query Language
JS	JavaScript
RBAC	Role-Based Access Control
URL	Uniform Resource Locator
ORM	Object-Relational Mapping
JSON	JavaScript Object Notation
API	Application Programming Interface
LTS	Long-Term Support
UI	User Interface
IDE	Integrated Development Environment
PHQ	Patient Health Questionnaire
GAD	Generalized Anxiety Disorder
UCD	User-Centered Design
SDLC	Software Development Life Cycle
CSRF	Cross-Site Request Forgery
CRUD	Create, Read, Update, and Delete

TABLE OF CONTENTS

Title			Page No.
DECLARATION…….……..			iii
ABSTRACT ……..……...…			iv
ACKNOWLEDGMENT……			v
LIST OF ABBREVIATIONS ……..…			vi
CHAPTER 1			
INTRODUCTION TO MENTAL WELLNESS CONSULTATION PLATFORM			1-9
1.1	Introduction ……..……		1
1.2	Objectives ……..……..		1-4
	1.2.1	General Objectives ……..…….	2-3
	1.2.2	Specific Objectives ……..…….	3-4
1.3	Project Timeline and Milestones ……..…		5-7
	1.3.1	Work Plan Gantt Chart……	5-6
	1.3.2	Implementation Milestones ……	6-7
1.4	Problem Statement……..…		7-8
1.5	Overview of Existing Mental Wellness Services and Gaps ……		8-9
CHAPTER 2			
BACKGROUND OF MENTAL WELLNESS CONSULTATION PLATFORM			10-22
2.1	System Architecture Approach ……		10-11
2.2	Technology Stack Core……		11-14
2.3	AI-Assisted Mental Wellness Support Chatbot ……		15-20
	2.3.1	Purpose, Scope, and Safety-First Processing ……	15-16
	2.3.2	RAG and Knowledge Base ……	16-17
	2.3.3	AI Response Generation and Conversation Context ……	17-19
	2.3.4	Crisis Handling, Safe Fallback, and Reliability ……	19-20
2.4	Comparative Analysis of Existing Mental Wellness Platforms ……		20
2.5	Requirement Specification ……		21-22

2.5.1 Hardware Requirement …… 21
2.5.2 Software Requirements …… 22

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN 23-49
3.1 System Architecture Diagram …… 23-25
3.2 Use Case Diagram……..… 25-27
3.3 Data Flow Diagrams…….. 27-30
3.3.1 Level 0 Data Flow Diagram…….. 27-29
3.3.2 Level 1 Data Flow Diagram…….. 29-30
3.4 Activity Diagrams……..…… 30-33
3.4.1 Patient Activity Diagram …… 31-32
3.4.2 Doctor and Admin Activity Diagram …… 32-33
3.5 Class Diagram……. 33-38
3.6 Sequence Diagrams……. 38-42
3.6.1 Consultation Booking and Care Sequence Diagram…… 38-41
3.6.2 AI Support and Blog Sequence Diagram …… 41-42
3.7 Entity-Relationship (ER) Diagram…… 42-46
3.8 Database Table Structure Diagram …… 46-49

CHAPTER 4

CONCEPTUAL FRAMEWORK AND SYSTEM DESIGN PRINCIPLES 50-59
4.1 Iterative and Incremental Development in the Project 50-51
4.2 Modular Design in the Project Structure 51-53
4.3 User-Centered Design in the Project Interface 53-54
4.4 Task-Technology Fit in the Mental Wellness Consultation Process 54-55
4.5 Cognitive Load and Interface Simplicity 55-56
4.6 Progressive Disclosure in User Interaction 56-57
4.7 Self-Determination and User Engagement 57-58
4.8 Error Prevention and Recovery in Core Workflows 58-59

CHAPTER 5

TESTING 60-62

5.1 Testing Strategy 60
5.2 Automated Testing 60-61
5.3 Test Execution and Results 62
5.4 Verified Functional Areas 62
5.5 Chapter Summary 62

CHAPTER 6

SYSTEM INTERFACE AND FEATURE DEMONSTRATION 63-94
6.1 Public Access and Authentication Interfaces 63-68
6.1.1 Home Page 64
6.1.2 Patient Registration 65
6.1.3 Role-Based Login 66-68
6.2 Patient Assessment and Personalized Care Entry 68-74
6.2.1 Patient Dashboard 68
6.2.2 Mental Wellness Assessment 69-71
6.2.3 Assessment Results, Doctor and Blog Recommendations 71-73
6.2.4 Doctor Discovery and Specialty Filtering 73-74
6.3 Appointment, Payment, and Consultation Workflow 74-79
6.3.1 Appointment Scheduling 74-75
6.3.2 Email OTP Payment Verification 75-77
6.3.3 Appointment Confirmation 77-78
6.3.4 Patient Appointment Management 78-79
6.3.5 Online Video Consultation 79
6.4 AI-Assisted Mental Wellness and Resource Support 79-84
6.4.1 Mental Wellness Blogs 79-80
6.4.2 Semantic Blog Search 80
6.4.3 AI Quick Summary 80-81
6.4.4 AI Support Chatbot 81-83
6.4.5 Breathing Exercise 83
6.5 Doctor Professional and Care Management 84-89
6.5.1 Doctor Dashboard 84-89

6.6 Administrative Management 89-94
6.6.1 Admin Dashboard 89-93
6.6.2 Role-Based Account Creation 93-94
6.7 Supporting Platform Information 94
6.7.1 About MindLift 94

CHAPTER 7

PROJECT CONCLUSION AND FUTURE DEVELOPMENT 95-99
7.1 Conclusion 95-97
7.1.1 Project Summary 95
7.1.2 System Realization 95-96
7.1.3 Key Functional Contributions 96
7.1.4 Overall Impact 96-97
7.2 Further Development 97-98
7.2.1 Functional Extension 97
7.2.2 Interface Improvement 97-98
7.2.3 System Security Enhancement 98
7.2.4 Long-Term Scalability 98
7.3 Final Remarks 98-99

REFERENCES 100-102

LIST OF TABLES

Table No.	Title	Page No.
Table 1.1	Milestone Table	7
Table 2.1	Supporting Libraries	13
Table 2.2	External Services and Their Purpose	14
Table 2.3	Comparative Analysis of Existing Mental Wellness Platforms	20
Table 2.4	Hardware Requirement	21
Table 2.5	Software Requirements	22
Table 4.1	Development Iterations and Implemented Features	51
Table 4.2	Modular Design of the Proposed System	53
Table 4.3	Application of UCD Principles in Platform Design	54
Table 4.4	Application of Self-Determination Theory in the System	58
Table 5.1	Automated Test Coverage	61
Table 5.2	Automated Test Results	62

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.1.1	Gantt Chart	5
Fig.3.1	System Architecture Diagram	23
Fig.3.2	Use Case Diagram	25
Fig.3.3	Level 0 Data Flow Diagram	27
Fig.3.4	Level 1 Data Flow Diagram	29
Fig.3.5	Patient Activity Diagram	31
Fig.3.6	Doctor and Admin Activity Diagram	32
Fig.3.7	Class Diagram	34
Fig.3.8	Consultation Booking and Care Sequence Diagram	39
Fig.3.9	AI Support and Blog Sequence Diagram	41
Fig.3.10	Entity-Relationship (ER) Diagram	43
Fig.3.11	Database Table Structure Diagram	46
Fig.5.1	Django Automated Test Execution Result	61
Fig.6.1	MindLift Home Page Interface	64
Fig.6.2	Patient Registration	65
Fig.6.3	Patient Login	66
Fig.6.4	Doctor Login	66
Fig.6.5	Admin Login	67
Fig.6.6	Role-Based Login Access Restriction	67
Fig.6.7	Patient Dashboard	68
Fig.6.8	Mental Wellness Assessment	70
Fig.6.9	Assessment Results	71
Fig.6.10	Doctor Recommendations	72
Fig.6.11	Blog Recommendations	72
Fig.6.12	Doctor Discovery	73
Fig.6.13	Specialty Filtering	73
Fig.6.14	Appointment Scheduling	74
Fig.6.15	Email OTP Payment Verification	75
Fig.6.16	Email OTP Delivery for Payment Verification	75
Fig.6.17	Appointment Confirmation Email	76
Fig.6.18	Appointment Confirmation	77
Fig.6.19	QR-Enabled Payment Receipt	78
Fig.6.20	Patient Appointment Management	78
Fig.6.21	Online Video Consultation	79
Fig.6.22	Mental Wellness Blogs	79

Figure No.	Title	Page No.
Fig.6.23	Semantic Blog Search	80
Fig.6.24	AI Quick Summary	80
Fig.6.25	AI Support Chatbot	81
Fig.6.26	Context-Aware AI Mental Wellness Support Conversation	81
Fig.6.27	Safety-Oriented Emergency Response	82
Fig.6.28	Breathing Exercise	83
Fig.6.29	Doctor Dashboard	84
Fig.6.30	Patient Management	84
Fig.6.31	Doctor Appointment Management	85
Fig.6.32	Availability and Schedule Management	85
Fig.6.33	Professional Profile Management	86
Fig.6.34	Doctor Blog Creation Interface	87
Fig.6.35	Doctor Blog Management	87
Fig.6.36	Patient Care Record and Consultation History	88
Fig.6.37	Follow-Up Wellness Task Assignment	88
Fig.6.38	Doctor Performance Analytics	89
Fig.6.39	Admin Dashboard	89
Fig.6.40	Recent Platform Activity Overview	90
Fig.6.41	User Management	90
Fig.6.42	Doctor Management	91
Fig.6.43	Appointment Management	91
Fig.6.44	Assessment Management	92
Fig.6.45	Financial Management	92
Fig.6.46	Role-Based Account Creation	93
Fig.6.47	About MindLift	94

CHAPTER 1

INTRODUCTION TO MENTAL WELLNESS CONSULTATION PLATFORM

1.1 Introduction

A distressing reality is becoming transparent in the aftermath of pandemic. COVID-19 being brought under control may have ended a crisis in Bangladesh, but it has marked the inception of a lingering psychological impact. An incalculable number of people have been struggling to make the transition from isolation back to the hyper-competitive environment. As per recent findings, our country sees approximately four hundred student suicides annually—the greater part of which are due to mental health issues [1]. Seeking professional assistance is hardly possible for the average person in our current socio-economic context. It's no wonder that 90% of sufferers of mental illness remain beyond the reach of treatment [2].

Since solving a 21st-century psychological crisis with 19th-century administrative tools is evidently futile, we have taken the initiative of building MindLift: Mental Wellness Care and Consultation Platform. Rather than just being another booking site, this project places emphasis on clinical persistence. The platform has been designed around the idea that digital technology can make professional mental wellness services more organized, accessible, and easier to continue over time. Patients, doctors, and administrators are provided with separate roles so that each of them can interact with the system according to their responsibilities while remaining part of the same care process.

Another important consideration behind MindLift is that mental wellness support should not end once a consultation session is completed. The system therefore connects professional consultation with assessment, follow-up care, wellness resources, and intelligent assistance. AI-supported components are incorporated where they can improve access to contextual information and immediate guidance, while professional consultation remains central to the overall care process. In this way, MindLift attempts to combine structured healthcare management with continuous digital wellness support rather than treating mental health care as a collection of isolated activities.

With all the aforementioned considerations, the platform's architecture prioritizes the enduring and contextual care of mental illness. MindLift is intended to provide a practical digital environment in which the different stages of mental wellness support can remain connected, helping users move from identifying their needs toward professional consultation and continued care in a more organized manner.

1.2 Objectives

We did not want to build just another static website that behaves like a basic data entry form. From the beginning, the aim was to create a system that feels active in the care process. A mental wellness consultation platform should not simply collect information and leave the user alone with it. It should guide the user, organize the next step, and help the right people notice important signals at the right time. Bearing that in mind, this project sets out to develop a

proactive, role-based digital platform for mental wellness support. The system is intended to connect patients, practitioners, and administrators through one organized service flow. A patient may begin with a private assessment, receive a result summary, view suitable doctor recommendations, book a consultation, complete payment, and continue with prescribed follow-up tasks. That journey may seem straightforward, but in mental wellness support it can be quite important. Many people do not begin with clear understanding of what they are experiencing. They may only know that sleep feels disturbed, concentration is poor, stress is rising, or daily routines are becoming harder to manage. Another objective is to make the platform responsive to user activity. If a patient leaves an assessment incomplete, misses assigned tasks, stops engaging after consultation, or submits responses that may suggest serious distress, the system should support timely attention. This does not mean the platform replaces professional judgement. It means the software can act as a protective support layer by making important patterns easier to see. What the project tries to do, in part, is take mental wellness information that is usually scattered around and make it easier to work with. Assessment responses, appointment histories, consultation notes, prescriptions, task progress, doctor recommendations, even notifications, can all sit in one linked space instead of being pulled from different corners of the system. That kind of setup can save people from a fair bit of confusion.

1.2.1 General Objectives

What this project is really trying to do is make mental wellness support feel less disordered, less hard to move through, and a bit more practical for the people inside it. Patients, doctors, and administrators all need different things, so the platform is meant to keep those roles separate without turning the whole service into a maze. A person might start with a self-assessment, get a recommendation, book a consultation, return for follow-up, while administrators keep an eye on how the service is holding together in the background. It is not just a storage space for information. The bigger idea is to make that information show up in a way that actually helps someone make a decision.

For patients, the first step is often the hardest one. Some people know something feels wrong, yet they hesitate before talking to a professional. That hesitation can come from uncertainty, fear of being judged, or simply not knowing whether their concern is serious enough. A private assessment gives them a quieter way in. A guided recommendation can take a little pressure off too. It may not solve the whole problem, of course, but it can make the next move feel less heavy.

Doctors need a different kind of support. In a real consultation, context matters a lot. If a doctor has to move between several patients during a busy schedule, nobody wants them digging through disconnected records just to figure out what happened previously. Assessment results, appointment history, consultation information, prescriptions, and follow-up tasks — when those pieces sit together, the conversation usually starts from a better place. Even a small detail, such as an unfinished wellness task or a concern discussed during an earlier consultation, may provide useful context for subsequent care.

The administrative side cannot be treated as an afterthought either. Mental wellness services do not run on appointments alone. Someone has to keep the service organized at a higher level

and manage the users, doctors, appointment records, payments, and content on the platform. The patient may not always see a structured administrative layer but it is necessary to make the platform work properly.

The platform makes that separation easier, because different roles need different levels of access. Patients should not be dealing with administrative tools, while administrators do not require the same interface or functions as doctors. That separation reduces unnecessary complexity while preserving a connected care flow.

Another general objective is to extend mental wellness support beyond scheduled consultations. MindLift therefore incorporates mental wellness articles, semantic blog search, AI-assisted quick summaries, and an AI-assisted emergency support service based on contextual Retrieval-Augmented Generation (RAG). The components are designed to enhance access to relevant information and supportive guidance while keeping professional mental health care at the center of the system.

Ultimately the project is trying to stay grounded. Answering an assessment, finding a doctor, booking a session, joining the consultation, receiving guidance, completing follow-up tasks, and managing service work — these are the things the system is built around.

1.2.2 Specific Objectives

Supporting Patients Through an Organized Care Journey:
Engaged in the decision-making process, users become empowered e-patient [3]. Rather than restricting patients to appointment booking alone, MindLift allows them to move through a connected sequence involving mental wellness assessment, personalized doctor recommendation, consultation, prescription, and follow-up wellness activities. The idea is to provide patients with a more clear pathway from identification of a concern to professional help and continuing care afterward.

Doctor Recommendations Based on Assessment:
One particular goal is to reduce the uncertainty of choosing a mental health professional. Assessment results on mental wellness are used to support personalized doctor recommendations and help patients find professionals that are more relevant to their needs instead of choosing blindly without contextual guidance. This follows the broader digital mental-health approach of using screening results to guide users toward appropriate treatment options and relevant services [4]. Developing a systematic consultation and Follow up procedure: The system aims to integrate appointment scheduling, online consultation, digital prescription, and doctor-assigned wellness tasks in a single workflow. Doctors can check the relevant patient information and past consultation records and patients can continue to do the assigned wellness activities post a session. This provides for a greater continuity from one consultation to the next.

Defining a Formal User-to-Role Assignment:
Implementation of a mapping between the set of Users and the set of Roles (Doctor, Patient, and Admin). This objective ensures that access rights are not assigned to individuals but are

dependent on the user's role. With this standard RBAC approach, long-term administration becomes much easier [5].

Well-Organized Appointment Management:
The platform aims to add more reliability to appointment handling by way of structured scheduling and email OTP confirmation. The appointment information is available to the related users, so that the patients and doctors can keep track of the consultations through their respective interfaces and the administrator can monitor the appointment related operations from the management side of the system.

Providing Contextual AI-Assisted Mental Wellness Support:
A specific objective of MindLift is to provide accessible mental wellness guidance through an AI-assisted emergency support feature. The chatbot employs a contextual Retrieval-Augmented Generation (RAG) approach so that responses can draw upon curated mental wellness information rather than depending only on generic generated responses. The system is not intended to replace professional mental health care, but is supportive and has built-in safety-oriented handling and instant help resources for discussions that might include crisis-related issues.

Enhancing access to relevant Mental Wellness information:
The platform is designed to help you discover and make sense of educational content about mental wellness. Doctors' articles offer structured wellness information. Semantic blog search allows users to search based on the meaning and context of the queries, rather than on exact matching of keywords alone. AI-assisted quick summaries also help users find the central points of an article without first having to process lengthy content to assess its relevance.

Keeping Centralized Records During the Care Process:
Another goal is to reduce fragmentation, by having related information in one coordinated system. Assessment results, doctor recommendations, appointments, consultation information, prescriptions, follow-up wellness tasks, and relevant administrative records are maintained in connected workflows. This allows each authorized role to access the information necessary for its responsibilities without depending on scattered records.

Supporting Effective Administration of Platforms:
The system is intended to provide a structured environment to the administrator to manage users, monitor the appointments and payment records and moderate the content of the platform. This administrative oversight helps to keep the day-to-day operation of the service running, while clearly separating patient-facing, doctor-facing and administrative responsibilities.

1.3 Project Timeline and Milestones

This section outlines the structured implementation schedule for the development and deployment of the platform. The project spans a six-month duration, broken down into distinct developmental phases to ensure systematic progress and timely delivery of key outputs.

1.3.1 Work Plan Gantt Chart

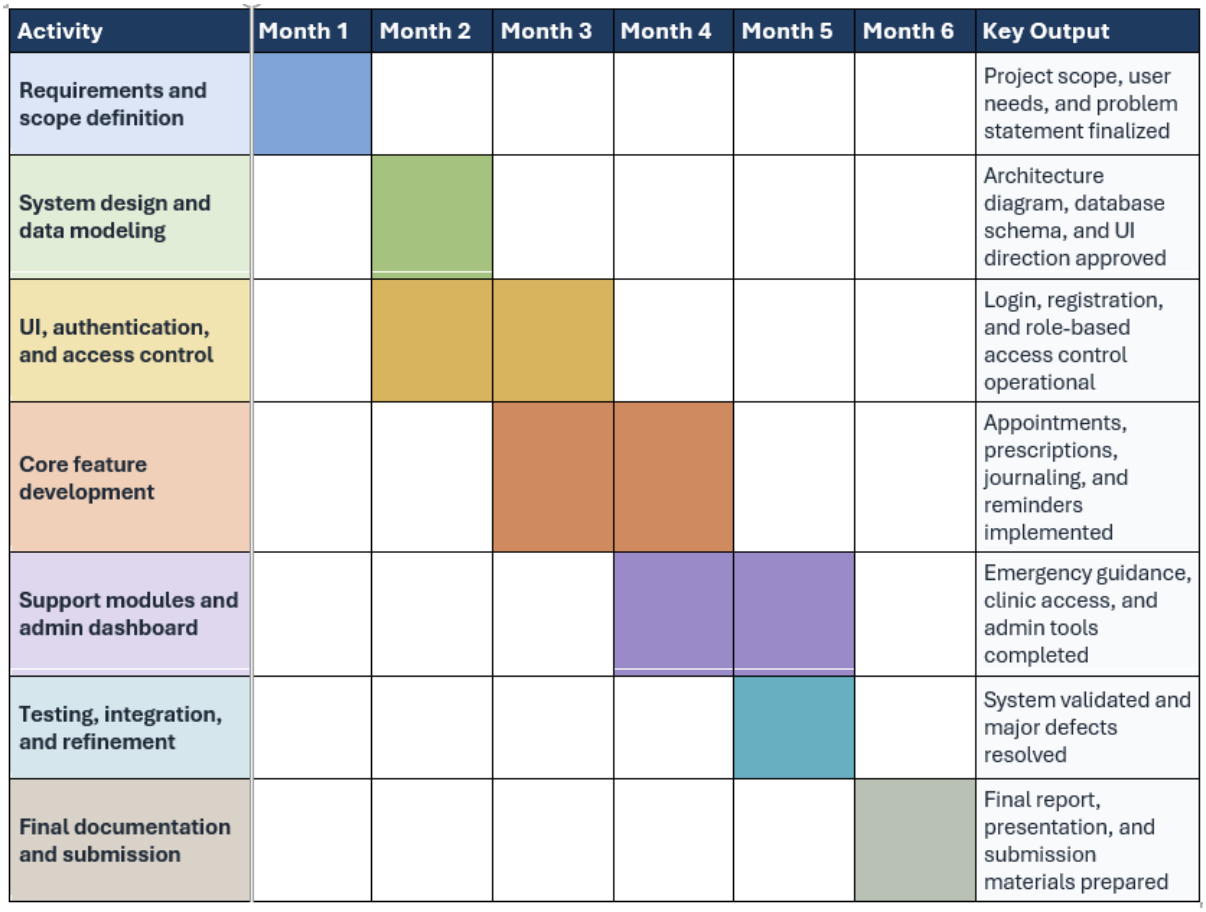


Figure 1.1: Gantt Chart

Month 1: Requirements gathering and scope definition
This is where the work begins, with the team sorting out what the project needs to do, where it should stop, and what problems it is actually meant to solve.

Month 2: System design and initial interface work

By Month 2, the work shifts toward system design and data modeling. At the same time, the UI, authentication, and access control begin to take shape. That overlap is intentional, since the interface and security layer probably need to develop in step rather than one waiting on the other.

Months 3 and 4: Core development
This is the main build period. The core features come in here: mental wellness assessments, personalized doctor recommendations, appointment scheduling with email OTP confirmation, online consultation, digital prescriptions, and doctor-assigned follow-up wellness tasks. In the same stretch, the supporting modules and admin dashboard are developed too, including AI-assisted emergency support using RAG, mental wellness articles, semantic blog search, AI-assisted quick summaries, appointment and payment monitoring, and content management.

Month 5: Testing and integration
This is the checking stage. This is usually where small mismatches start showing up — an assessment result that does not flow correctly into doctor recommendations, an appointment status that does not update as expected, an OTP confirmation that fails to behave consistently, or a dashboard view that looks fine on its own but feels off once everything is connected. This stage also checks that consultation, prescription, follow-up tasks, AI-assisted features, and role-based access work properly as part of one integrated system.

Month 6: Documentation and submission
The final month is kept for documentation and submission. That leaves room to prepare the report, presentation, and handover package without having to rush the last part.

1.3.2 Implementation Milestones

Milestone	Timeframe	Deliverable	Verification / Remarks
M1	Month 1	Project proposal approved	Problem statement, scope, and research direction confirmed
M2	Month 2	System design finalized	System architecture, database design, UI plan, and role structure finalized
M3	Months 2-3	Authentication and role-based access completed	Patients, doctors, and administrators can register/log in and access role-based features
M4	Months 3-4	Core feature development completed	Assessment, doctor recommendation, appointments, email OTP confirmation, online consultation, prescriptions, and follow-up wellness tasks implemented

Milestone	Timeframe	Deliverable	Verification / Remarks
M5	Month 5	Testing and integration completed	Core modules, AI/RAG support, semantic blog search, AI summaries, admin functions, and role-based workflows tested as an integrated system
M6	Month 6	Final submission ready	Documentation, presentation slides, testing evidence, and final demonstration prepared

Table 1.1: Milestone Table

The milestone table divides the project into a six-month schedule. That pacing seems pretty fair, because it lets for each stage instead of trying to cram everything toward the deadline. The first month is devoted to requirements gathering and scope definition. Before any development begins, the team needs to settle what the platform is expected to do, where its boundaries lie, and which problems deserve attention first. By the second month, attention turns to system design, data modeling, and early interface work. Authentication and role-based access control begin during the same period, so the foundation starts forming before the larger features arrive.

Most of the implementation takes place during months three and four. Mental wellness assessments, personalized doctor recommendations, appointment scheduling with email OTP confirmation, online consultation, digital prescriptions, follow-up wellness tasks, AI-assisted emergency support, semantic blog search, AI quick summaries, and administrative functions gradually come together over this stage instead of appearing all at once. Month five tends to be where smaller issues become visible. An appointment status may not update as expected, an OTP confirmation may behave inconsistently, an assessment result may not flow correctly into doctor recommendations, or a dashboard page may work perfectly on its own yet feel slightly inconsistent after the rest of the system is connected. Those are the kinds of problems that often stay hidden until everything begins working as one application.

The final month shifts the focus away from development. Documentation is completed, the presentation receives its finishing touches, testing evidence is organized, and there is still enough time to deal with loose ends before the project is submitted.

1.4 Problem Statement

Average mental health systems in Bangladesh have a propensity to operate on a fixed structure. That can be overwhelming for an adolescent, or even for an adult to follow along. Insufficiency of therapists is also one of the critical issues. A recent report shows that 60,000 patients on average get only one psychiatrist in our country [6]. To make the situation worse,

a significant number of existing platforms struggle to account for a patient's necessity for unconditional privacy. Like these weren't enough, mental health conditions cost up to 0.76% of the GDP due to the reduced workforce productivity [7]. The problem, however, is not limited to the availability of professionals. People seeking mental wellness support may also face difficulty understanding their condition, deciding which professional would be appropriate, maintaining continuity after a consultation, and keeping different stages of their care connected. Separate processes for handling assessment results, appointments, consultations, prescriptions and follow-up activities can result in fragmentation. This makes the whole care journey harder for patients, but can also restrict the contextual information available to doctors in subsequent consultations. Another challenge is the timely and relevant support when consultation hours are not scheduled. Users may need access to reliable mental wellness information or immediate supportive guidance without having to search through unrelated online resources. At the same time, such digital assistance needs to remain contextual and safety-conscious without attempting to substitute professional judgement. Our project responds to the need for a practical mental wellness system that addresses these gaps through one coordinated digital environment. MindLift brings assessment, professional consultation, follow-up care, mental wellness resources, and AI-assisted support into a connected workflow while maintaining separate role-based access for patients, doctors, and administrators. We have built a secure space which functions amidst the relative infrastructural gaps. The aim is therefore not merely to digitize appointment booking, but to reduce fragmentation and provide a more organized path through different stages of mental wellness care.

1.5 Overview of Existing Mental Wellness Services and Gaps

There were various mental wellness services that were used as references for this project. Each had its own features. LifeSpring is a leading mental-health and wellness provider in Bangladesh providing online counseling, child development services, corporate wellbeing assistance and free mental-health tests for general wellbeing [8]. The platform gives the impression of being more of a referral and information portal, supported by educational content and self-assessment tools, rather than a tightly integrated healthcare workflow system. Moner Bondhu follows a broader counselling-oriented approach, offering mental wellness screening, counsellor matching, appointment booking, and online counselling services [9]. Although these services make professional support easier to access, the publicly available platform features do not indicate the same level of integration between assessment, consultation, prescription, follow-up wellness tasks, AI-assisted support, and administrative management found within MindLift. There is also MindSheba which offers psychotherapy, psychiatrist consultation, nutritionist services, and access to licensed psychologists and psychiatrists through a substantial number of branches across the country [10]. Its workflow seems to be contingent mainly on appointment booking and specialist selection rather than a unified process connecting assessment-based recommendations, consultation, follow-up activities, intelligent content discovery, and administrative oversight.

International platforms such as BetterHelp and Talkspace are well established in remote therapy delivery [11], [12]. They are for the most part designed to make professional therapy accessible online in a simple and continuous way, putting emphasis on a straightforward therapist–client interaction. Both platforms provide mechanisms for connecting users with suitable professionals, while Talkspace also extends its services into psychiatry and AI-assisted mental wellness support. However, the focus and organization of these platforms differ from the system proposed in this project.

Compared with these services, MindLift is designed around an integrated mental wellness care journey rather than a single consultation or communication function. The platform connects mental wellness assessment, assessment-based doctor recommendation, appointment management, online consultation, digital prescriptions, and doctor-assigned wellness tasks within one role-based environment. It further introduces contextual RAG-assisted mental wellness support, semantic search for mental wellness articles, AI-assisted article summaries, email OTP appointment confirmation, and downloadable appointment receipts with QR codes. Patients, doctors, and administrators interact through separate role-specific interfaces while remaining part of the same coordinated workflow. This ultimately brings different aspects of mental wellness support together into a single, coordinated platform that makes navigation and management pragmatic for the users.

CHAPTER 2

BACKGROUND OF MENTAL WELLNESS CONSULTATION PLATFORM

2.1 System Architecture Approach

MindLift is a modular Django web architecture. Backend logic, database interactions, templates, static assets, mental wellness assessment routines, appointment workflows, consultation features, AI-assisted services, and content-management functions are organized in a modular Django-based structure. At first glance this structure looks conventional but it is actually a balance between simplicity and flexibility.

Users can register, log in, complete assessments, discover doctors, book appointments, process payments, attend online consultations, access prescriptions, manage wellness tasks, browse mental wellness articles, and use AI-assisted support—all by means of browser-based pages. The requests are handled by Django's URL routing and go through authentication and role checks. Then the business logic is executed in the views and the ORM is used to interact with the database. Depending on the request the result is either rendered HTML pages or JSON responses. SQLite is the main database to store application data and Django's ORM is the layer that the application uses to read and update these records.

There are also several services that are not part of the normal request-response flow in the architecture. The appointment process involves OTP verification through email and Jitsi for the video consultation platform. The platform has AI-assisted features including contextual mental wellness support, semantic blog search and article quick summaries. The emergency-support chatbot is based on a Retrieval-Augmented Generation (RAG) approach that retrieves relevant information from curated mental wellness knowledge before generating a contextual response. This allows the AI piece to operate within the larger platform rather than as a single chatbot.

Three main user roles are distributed by the platform: patient, doctor, and administrator. Patients are at liberty to complete mental health assessments, inspect recommendations, set up appointments, attend consultations, acquire prescriptions, complete wellness tasks, browse through educational blogs, and access AI-assisted mental wellness support. Doctors are able to manage profiles, availability, appointments, consultation records, prescriptions, follow-up tasks, and blog posts. Lastly, administrators keep watch over users, doctors, appointments, payment records, and platform content so that everything is functioning as had been expected. Structuring the platform around roles is likely to cut back the likelihood of accidental data exposure and not to mention, it also aligns with web security best practices [13].

Overall, the architecture keeps assessment, recommendation, appointment management, consultation, prescriptions, follow-up care, content services, administrative operations, and AI-assisted functionality connected within one Django-based system while still keeping the responsibilities of individual modules and user roles clearly separated.

2.2 Technology Stack Core

MindLift is based on a lean technology stack and doesn't rely on a lot of unrelated frameworks. The backend is written in Python using Django and the application data is persisted in SQLite. The browser-based interface is built using HTML, CSS, JavaScript, and Django templates. Additional libraries and external services are integrated only where they support particular functions of the platform.

The main technologies used in the system are described below.

Python

The backend of MindLift is mainly programmed in Python. It is used for:

- Application and business logic
- Calculation and processing of mental wellness assessment scores
- Processing of personalized doctor recommendations
- Appointment scheduling and payment workflow handling
- PDF generation for receipts and prescriptions
- RAG knowledge retrieval and chatbot processing
- Blog-search processing and AI-assisted features
- Communication with external AI and email services

Python’s integration with Django allows most backend functions to remain within a consistent programming environment while supporting the different patient, doctor, and administrative workflows of MindLift.

Django Framework

The project uses Django 5.2.12 as its primary web framework. Django provides a number of core facilities used throughout MindLift, including:

- URL routing
- Views for request processing
- Template rendering
- Authentication and sessions
- Forms and data validation
- Database models and ORM
- Middleware and CSRF protection

- Role-based application logic
- Administrative functionality

The application contains Django models representing entities such as Doctor, PatientAssessment, ClinicalAssessment, Appointment, Consultation, Payment, Prescription, DailyTask, BlogPost, and ChatSession. In the system, these records are managed through Django models, where each model generally maps to a single database table, and Django provides an automatically generated database-access API [14]. Django forms and backend validation are also used before submitted information is processed or stored. This parallels the standard web application practice of validating submitted data before using it in application logic or saving it in the database [15]. MindLift provides its own role-specific administrative interfaces while also retaining Django's underlying administrative facilities for application-data management. Django also provides an administrative interface that can be customized for managing application data and keeping an eye on records [16].

Django ORM and SQLite

MindLift's current relational database is SQLite. It was chosen for academic implementation because it is:

- Light weight
- Self-contained
- Configurable with Django easily
- Suitable for local development and demo
- Easy to save and transport with the project.

Communication layer between backend Application and SQLite is built using Django ORM. Instead of writing raw SQL again and again in the application, backend functions are implemented with Django models written in python. This keeps the database access consistent with the rest of the Django application and makes it easier to maintain the data layer. Also, the ORM-based structure gives more flexibility in the future if the database system needs to be changed, because a lot of the application level data access logic can be kept independent of the underlying database engine.

HTML, CSS, JavaScript

The user interface is built primarily with standard web technologies:

- HTML5 defines the structure of pages, forms, tables and other interface components.
- CSS3 controls layout, responsiveness, typography, buttons, cards, dashboards, tables, and overall visual presentation.
- JavaScript provides interactive browser behaviour and asynchronous communication.

The JavaScript Fetch API is particularly useful where the interface needs to communicate with the Django backend without completely reloading the page. This is used in areas such as

chatbot interaction, consultation-room behaviour, email OTP actions, AI-assisted blog functions, and selected appointment-related operations. The interface is largely constructed using custom HTML, CSS, JavaScript, and Django templates.

Supporting Libraries
Several additional Python packages support particular functions of MindLift:

Technology	Main Use
django-allauth	Account management and Google authentication
ReportLab	Generation of PDF receipts and prescriptions
Pillow	Image handling and processing support
WhiteNoise	Static-file serving and management

Table 2.1: Supporting libraries

Jitsi Meet
Jitsi Meet is used to provide browser-based video consultation between doctors and patients. The consultation room is integrated into MindLift through the Jitsi Meet IFrame API.

Its role includes:

- Live patient-doctor video communication
- Microphone and camera control
- Participant joining and leaving
- Screen-sharing capability
- Consultation-session events

MindLift still controls the appointment and consultation workflow. Jitsi handles the real-time audiovisual communication, while Django determines when a consultation becomes available, which appointment the room belongs to, and how the consultation status is handled within the platform.

Google Authentication
The project supports optional Google sign-in through django-allauth and Google OAuth. In the current implementation, Google authentication is available as an additional sign-in method for patient accounts. Google is the external identity provider and MindLift manages the user’s platform role, authorization and access after authentication. Traditional username and password authentication remains available. Therefore, Google authentication provides an alternative method of signing in rather than replacing MindLift's normal account system.

Generative AI and RAG Services
MindLift integrates a Gemini-compatible generative AI model through an OpenAI-compatible API interface for its AI-assisted functions. The generative AI service contributes to:

- Mental wellness chatbot responses
- Contextual mental wellness assistance
- AI Quick Summary generation
- Semantic rescue for difficult natural-language blog searches

The emergency-support chatbot does not depend entirely on the external AI model. MindLift manages local safety rules, Retrieval-Augmented Generation (RAG), curated knowledge retrieval, conversation state, crisis-oriented handling, and fallback responses within the application. Retrieved context can then be combined with generative AI when an AI-generated response is appropriate.

The blog-search system also follows a hybrid approach. Local weighted relevance matching is performed first, while AI-assisted semantic retrieval is used when a natural-language query cannot be matched confidently through the local search mechanism. This allows semantic assistance to complement rather than completely replace local search.

External Services

Service	**Purpose**
Jitsi Meet	Online video consultation
SMTP / Gmail	Platform email and OTP communication
Google OAuth	Optional patient Google sign-in
Gemini-compatible AI service	AI-assisted chatbot, summaries, and semantic assistance

Table 2.2: External Services and Their Purpose

These external services expand MindLift’s capabilities while allowing Django to remain responsible for the platform’s central workflow, access control, and stored application information. Overall, the selected technology stack remains technically manageable for an academic project while supporting MindLift's assessment, recommendation, consultation, follow-up, administrative, content-management, and AI-assisted functions.

2.3 AI-Assisted Mental Wellness Support Chatbot

One of the most distinctive capabilities of MindLift is the AI-powered chatbot. It is available in the Emergency Support environment but is not limited to emergencies. The chatbot can provide general mental health advice, explain MindLift features, refer users to an assessment or professional help, provide access to breathing exercises and identify when immediate attention is needed.

Unlike a simple chatbot that sends every message directly to a generative AI model, MindLift follows a layered architecture consisting of:

- Safety and emergency detection
- Intent analysis
- Retrieval-Augmented Generation (RAG)
- Platform-specific and wellness knowledge retrieval
- Conversation context
- Generative AI
- Local fallback responses

This structure makes the chatbot more controlled, context-aware, and suitable for a mental wellness environment. Different types of messages are handled through different paths instead of depending on generative AI for every interaction.

2.3.1 Purpose, Scope, and Safety-First Processing

The chatbot can be an immediate touch point for users who may not be ready to enter into a formal consultation.

It can provide support or guidance for topics such as:

- Anxiety and stress
- Low mood and depression-related concerns
- Sleep difficulties
- Breathing and grounding exercises
- Panic-related concerns
- Guidance toward mental wellness assessments
- Doctor and appointment-related questions
- MindLift platform information
- Emergency and crisis-related situations

A major design principle is that safety checks occur before ordinary AI generation. When a message is received, the system first sanitizes and normalizes the text before examining it for high-risk expressions related to:

- Suicide or self-harm
- Inability to remain safe
- Violence toward others
- Severe breathing difficulty
- Choking or fainting
- Severe chest pain
- Other expressions suggesting an immediate medical emergency

If a serious risk is detected, the normal conversational path is interrupted and the chatbot provides controlled safety guidance rather than sending the message through an ordinary AI-response path. For non-emergency messages, the system also identifies several common types of user intent. These include greetings, casual conversation, vague requests for help, breathing-exercise requests, doctor or appointment questions, platform-specific questions, and general mental wellness concerns.This allows simple requests to receive simple responses. For example, a greeting does not need to activate RAG and generative AI, while an emergency statement should not be handled like an ordinary wellness question. Similarly, users directly asking how to find a doctor or book a consultation are guided toward completing the mental wellness assessment first, allowing the platform's assessment-based recommendation process to determine the next step.

2.3.2 Retrieval-Augmented Generation (RAG) and Knowledge Base

Retrieval-Augmented Generation, or RAG, allows MindLift to provide responses based on its own controlled knowledge rather than relying entirely on the general knowledge of the AI model.

The basic process is:

User Question → Knowledge Retrieval → Relevant Context → AI or Local Response

MindLift's chatbot knowledge is collected from several sources, including:

- Internal MindLift platform information
- Relevant website template content
- Published mental wellness blog posts
- Doctor profiles
- Doctor availability and schedule information
- Mental wellness assessment questions
- Doctor specialization information
- Doctor primary-focus information
- Wellness task templates
- Platform-specific documentation

- Curated mental wellness Markdown documents
- Crisis and safety Markdown documents

These sources are divided into smaller knowledge chunks and stored through the ChatbotKnowledgeChunk model. Each stored chunk contains its content, source information, and a local numerical representation that can be used during retrieval. This means the chatbot does not send the entire website or all stored information to the AI for every question. Instead, it only finds the most relevant knowledge chunks for the current query of the user.

MindLift uses a hybrid retrieval strategy, consisting of:

- Lexical similarity. It measures how well meaningful query terms match stored knowledge.
- Local vector similarity, which compares locally generated hash-based numerical representations of the query and knowledge chunks.

The retrieval system combines these signals and ranks matching knowledge before selecting the most relevant chunks.

For example:

- A question about appointments can prioritize appointment-related platform information.
- A question about breathing can retrieve breathing, anxiety, panic, or grounding material.
- A question about platform services can retrieve relevant MindLift-specific information.
- A question related to doctors can retrieve appropriate platform knowledge while directing the user toward the assessment-based recommendation process.
- A crisis-related message is handled through the dedicated safety layer before ordinary RAG generation.

MindLift does not currently depend on a separate vector database. Knowledge chunks and their vector information are stored through the existing Django data layer. This keeps the RAG architecture comparatively lightweight while still allowing the chatbot to respond using project-specific and curated mental wellness information.

2.3.3 AI Response Generation and Conversation Context

Once safety checks are completed and relevant knowledge has been retrieved, generative AI may be used to produce a natural response. MindLift uses a Gemini-compatible generative

AI service, while the application remains responsible for determining when AI generation is appropriate and what context should be supplied to it.

The AI model receives controlled instructions describing how the MindLift assistant should behave. These instructions include important boundaries such as:

- Maintain a calm, supportive, and professional tone.
- Do not diagnose mental disorders.
- Do not prescribe medication.
- Do not claim to replace doctors or therapists.
- Use retrieved MindLift context for platform-specific details.
- Initially, prioritize immediate safety if any indications of a crisis or emergency are present.
- Refrain from offering detrimental guidance or hazardous medical recommendations.
- Maintain responses that are concentrated and easily understood.
- Do not disclose confidential system details such as API keys or internal errors.

When context retrieval is necessary, the pertinent RAG information is conveyed alongside the user's message. This anchors the reply in concrete MindLift data, instead of allowing the AI to speculate on project details.

Conversation continuity is maintained using:

- ChatSession
- ChatMessage

Recent user and assistant messages are retained as conversation context for subsequent responses.

For example, if a user initially says:
“I have been feeling very anxious lately.”

and later writes:
“It happened again today.”

The recent conversation context helps the chatbot interpret the follow-up in relation to the earlier discussion rather than treating it as an entirely unrelated message. The chatbot does not continuously send the complete historical conversation. Instead, a limited recent context window is used so that responses remain relevant without unnecessarily increasing the amount of information supplied for generation. Generative AI is therefore only one part of the chatbot. Greetings, vague-help prompts, breathing-exercise directions, crisis responses, assessment guidance, and several platform-specific answers can be handled through controlled local logic without requiring generative AI.

2.3.4 Crisis Handling, Safe Fallback, and Reliability

Mental wellness support requires the chatbot to remain useful even when an external AI service fails or when a message is too sensitive to leave entirely to generative output. For this reason, MindLift maintains deterministic crisis and medical-emergency handling. When serious self-harm, suicide, violence, or safety-related language is detected, the ordinary AI-response path is bypassed and the chatbot prioritizes actions such as:

- Seeking immediate real-world assistance
- Contacting a trusted person
- Moving away from potential means of harm
- Using available emergency resources
- Reaching professional support
- Going to an emergency department where immediate medical attention is required

Medical warning signs are handled separately from ordinary anxiety symptoms. Severe breathing difficulty, fainting, choking, rapidly worsening breathing, or serious chest pain should not be treated merely as routine stress or panic. MindLift also records detected emergency situations through the EmergencyLog model. The log maintains structured information such as the detected indicators, risk level, recommended action, and associated chat session. Sensitive emergency message content is not required to be exposed directly in the administrative record.

Another important feature is fallback behaviour. If the configured generative AI service is unavailable or does not return a usable response, the chatbot can continue to rely on:

- Local RAG knowledge
- Crisis and medical-emergency rules
- Platform-specific responses
- Curated wellness guidance
- Assessment and consultation guidance
- Safe fallback templates

This means the chatbot does not become completely unusable simply because the external AI service cannot respond.

The overall chatbot architecture can therefore be summarized as:
Input Processing → Safety Detection → Intent Analysis → RAG Retrieval → Context Preparation → AI Generation or Controlled Local Response → Safe Fallback

This layered design is one of the major technical strengths of MindLift because the chatbot is not merely a generic AI interface. It combines platform knowledge, curated mental wellness

information, conversational context, deterministic safety controls, Retrieval-Augmented Generation, and generative AI assistance within the same mental wellness support system.

2.4 Comparative Analysis of Existing Mental Wellness Platforms

MindLift is compared here with some selected local and international mental wellness platforms like LifeSpring, Moner Bondhu, BetterHelp, and Talkspace. The comparison focuses on the publicly available features that are most pertinent to the scope and objectives of the proposed system.

Feature	**MindLift**	**LifeSpring**	**Moner Bondhu**	**BetterHelp**	**Talkspace**
Mental wellness assessment / screening	Yes	Yes	Yes	Yes	Yes
Assessment-based provider recommendation	Yes	Limited	Yes	Yes	Yes
Appointment booking	Yes	Yes	Yes	Yes	Yes
Online consultation	Yes	Yes	Yes	Yes	Yes
Digital prescription support	Yes	Limited	Limited	Limited	Yes
Doctor-assigned wellness tasks	Yes	No	No	Limited	Limited
AI-assisted mental wellness support	Yes	No	Limited	No	Yes
Semantic mental wellness article search	Yes	No	No	No	No
AI-assisted article quick summary	Yes	No	No	No	No
Email OTP appointment confirmation	Yes	No	No	No	No
Downloadable appointment receipt with QR code	Yes	No	No	No	No

Table 2.3: Comparative Analysis of Existing Mental Wellness Platforms

2.5 Requirement Specification

The system requirements define the essential hardware and software components needed to successfully build, deploy, and operate the platform. These specifications ensure that the environment is optimized for Django's backend logic and the system's security features.

2.5.1 Hardware Requirement

The hardware requirements specify the physical computing resources and devices needed for the development, operation and proper use of the platform. MindLift is a web-based system, which means it has modest hardware requirements and does not require specialized computing hardware. The primary requirement is a computer capable of running the Django development environment and a modern web browser and able to handle multiple platform functions at the same time. The hardware requirements also mirror the nature of MindLift's services. For online video consultation, patients and doctors require audio-visual devices to interact directly, whereas internet connectivity is necessary for external services such as Jitsi Meet, email communication, Google authentication, and AI-assisted functions. The generative AI processing is performed via an external service, not on the device itself, so there is no need for specific AI hardware or a high-performance graphics processor on the platform. The following table provides recommended specifications for reliable development and use of MindLift.

Component	**Minimum / Recommended Specification**
Processor	Intel Core i3 / i5 or equivalent
RAM	8 GB minimum, 16 GB recommended
Storage	256 GB SSD or higher
Display	1366 × 768 resolution or higher
Internet	Required for online services and stable platform access
Webcam	Required for video consultation
Microphone	Required for video consultation
Audio Output	Speaker or headphones
Peripherals	Keyboard and mouse

Table 2.4: Hardware Requirement

2.5.2 Software requirements

Software Requirements Software requirements specify the technologies and tools needed to develop and run MindLift. Python and Django are used for backend development. Data is stored in SQLite. User Interface is made with HTML, CSS, JavaScript and Django templates. Supporting services and libraries also provide support for video consultation, authentication, email communication, PDF generation and AI assisted features.

Component	**Technology**	**Specification**
Programming Language	Python	Python 3.13
Web Framework	Django	5.2.12
Database	SQLite	SQLite 3.x
Frontend	HTML5, CSS3, JavaScript	Standard browser-based interface
Authentication	django-allauth	65.17.0
Social Login	Google OAuth	Optional patient Google sign-in
Video Consultation	Jitsi Meet	Jitsi Meet IFrame API
AI Service	Gemini-compatible AI	OpenAI-compatible API integration
Email Service	SMTP / Gmail	Email and OTP communication
PDF Generation	ReportLab	5.0.0
Image Support	Pillow	12.1.1
Static Files	WhiteNoise	6.12.0
Virtual Environment	venv / Virtualenv	Python environment isolation
IDE / Code Editor	VS Code / PyCharm	Development environment
Browser	Chrome / Edge / Firefox	Modern JavaScript-enabled browser

Table 2.5: Software requirements

CHAPTER 3

System Analysis and Design

3.1 System Architecture Diagram

The MindLift System Architecture shows how the main components of the platform interact to provide an integrated environment for mental wellness care and consultation. The Django backend orchestrates the core platform's functions while patients, doctors and administrators access the system via a browser-based interface. It connects user activities with stored application data, local knowledge sources, and external services so that assessment, appointment management, consultation, follow-up care, content access, and AI-assisted support can operate within one connected system.

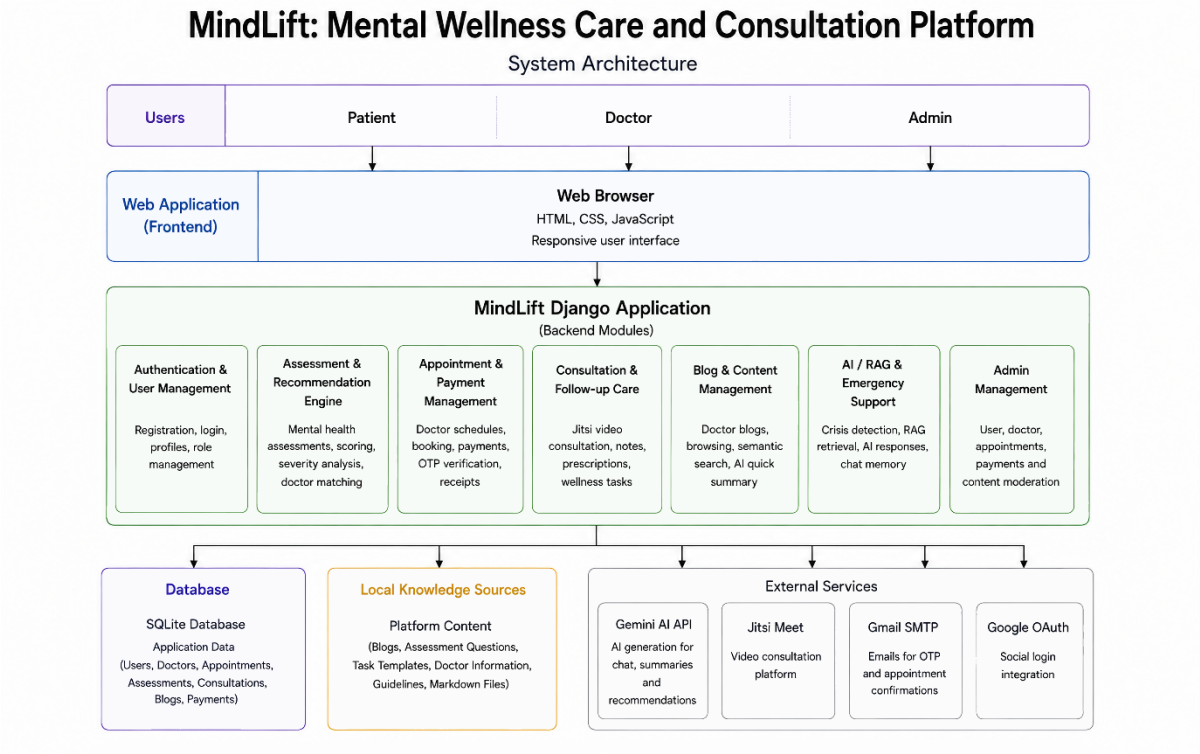


Figure 3.1: System Architecture Diagram

Users: MindLift supports three main user roles: Patient, Doctor, and Administrator. Each user interacts with the platform according to the responsibilities and permissions assigned to that role.

Web Application (Frontend) : The browser interface is developed using HTML, CSS and Java Script. It provides interactive access to the different functions of MindLift and shows role specific pages for patients, doctors and administrators.

Django Application (Backend): The Django application is the main process layer of the platform. It gets requests from users, does authentication and role checks, applies business logic, talks to a database and coordinates the different modules and external services.

Authentication & User Management: This module manages registration, login, user profiles, authentication, and role-based access for patients, doctors, and administrators.

Assessment & Recommendation Engine: This component processes the mental wellness assessments, computes the scores and severity levels, and uses the assessment result to enable personalized doctor recommendations.

Appointment & Payment Management: This module manages the doctor's schedule, appointment booking, payment records, email OTP confirmation, and digital appointment receipts. It manages the process of a free time for consultation and appointment confirmation.

Consultation & Follow up Care: This section contains online video consultation through Jitsi, consultation notes, digital prescriptions and wellness tasks prescribed by the doctor. This helps to ensure continuity of care across more than one consultation.

Blog & Content Management: Doctors are able to write mental wellness articles for users to browse and search for. The module also provides semantic blog search and AI-assisted Quick Summary to help users more easily find and understand relevant content.

AI / RAG & Emergency Support: This module provides safety-centric processing, Retrieval-Augmented Generation, local information retrieval, conversation context and generative AI to provide contextual mental wellness support. Crisis messages are routed on a controlled path of safety, before normal AI responses are applied.

Admin Management: Administrators oversee the broader operation of the platform by managing users and doctors, monitoring appointments and payment records, and moderating platform content.

Database: MindLift stores the structured application records required in SQLite including information related to user, doctor, assessment, appointment, consultation, prescription, wellness-task, blog, payment and chatbot.

Local Knowledge Sources: This layer contains the platform-specific and curated knowledge that is consumed by the AI/RAG system. Includes mental wellness content, assessments, doctor info, task templates, blogs, platform guidelines and curated markdown knowledge resources.

External Services: MindLift integrates select external services to augment its core capabilities. Jitsi Meet is utilized for online video consultations, Gmail SMTP for sending OTP and appointment-related emails, Google OAuth for optional patient Google sign-in and the

Gemini-compatible AI service for generating chatbot responses, AI Quick Summary, and semantic assistance.

Data Flow In General The main architecture flow is Users → Web Application → Django Backend. The backend will work with the SQLite database, local knowledge resources or external services based on the requested function. This architecture allows MindLift to consolidate assessment, personalized doctor recommendations, appointment booking, consultation, post-care follow-up, educational content, administrative functions and AI-enabled mental wellness support into a seamless platform.

3.2 Use Case Diagram

MindLift Use Case Diagram illustrates the interaction of Patients, Doctors and Administrators with the platform in accordance with their roles. The system offers a unified coordinated environment for mental wellness assessment, personalized doctor recommendation, appointment booking, consultation, follow-up care, mental wellness content, AI-assisted support and administrative management.

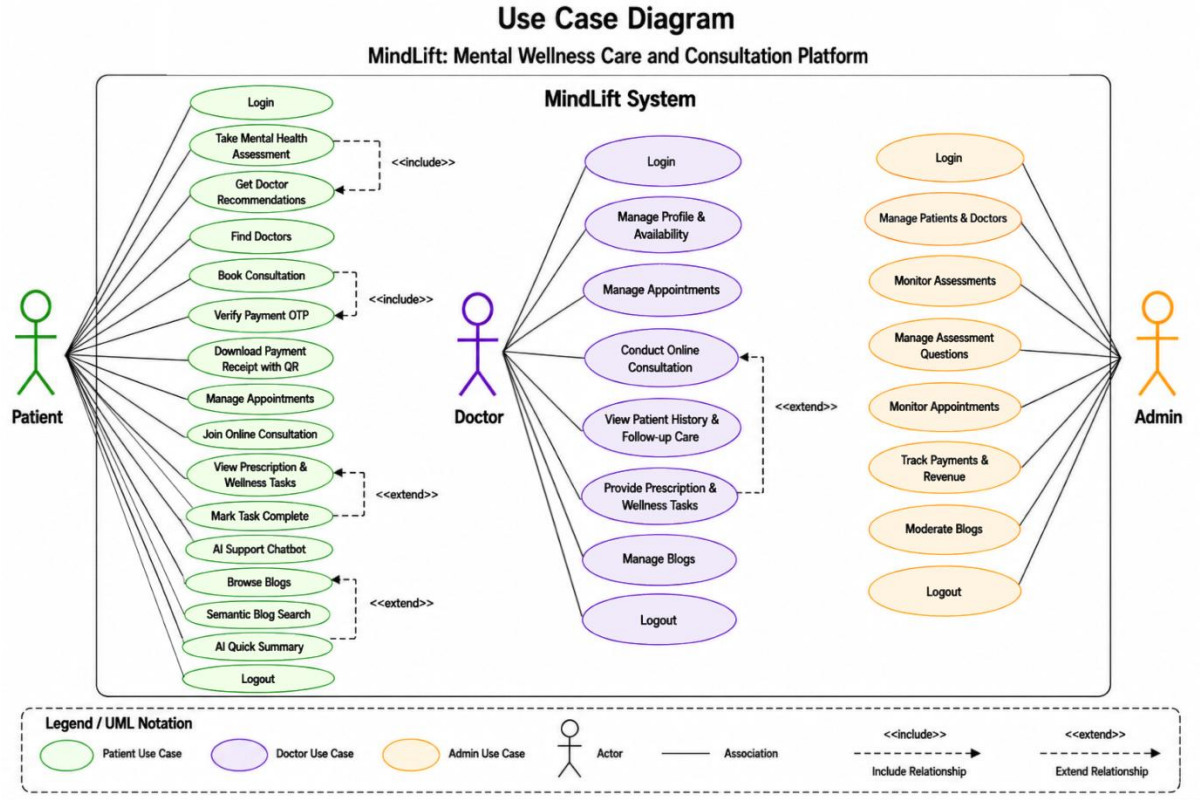


Figure 3.2: Use Case Diagram

Patient-Side Actions

The Patient is the primary user of MindLift for mental wellness and consultation services. After logging in, a patient can complete a mental health assessment and receive doctor recommendations based on the assessment outcome. Patients may also browse available doctors independently and select a suitable professional for consultation.

A patient can book a consultation, complete payment verification through email OTP confirmation, and download a payment receipt containing a QR code. When the appointment is confirmed, the patient can manage the appointment information and enter the online consultation when the session is scheduled. Post consultation, doctor can also assign wellness tasks and digital prescriptions for further follow up care. Patients can view these instructions through their account and mark assigned tasks as completed. They can also use the AI Support Chatbot for contextual mental wellness guidance and emergency-oriented assistance.

The content section allows patients to browse doctor-authored mental wellness blogs, perform semantic blog searches, and use AI Quick Summary to understand relevant articles more efficiently.

Doctor-Side Actions

MindLift helps doctors to manage their professional activities and to provide consultation and follow-up care. After logging in, doctors can update their profile and availability so that patients can see their professional details and available consultation schedules. The platform allows doctors to manage appointments and conduct online consultations. They can view relevant patient history and follow up information, provide digital prescriptions and assign wellness tasks on the basis of the patient's consultation requirements. Doctors can also run their own blogs on mental health, sharing professional knowledge and useful information with users outside the scope of formal consultation.

Admin-Side Actions

The Administrator shall be responsible for the overall operation, management and monitoring of MindLift. Once logged in, the administrator can manage patient and doctor records, monitor assessments and appointments and manage assessment questions.

The administrator can also track payment and revenue information related to consultation activities. In addition, blog moderation allows administrators to review and manage platform content so that the published information remains organized and appropriate for users.

So, the Use Case Diagram illustrates the primary duties of Patients, Doctors, and Administrators and how MindLift supports the entire mental wellness journey, including assessment, doctor recommendation, consultation, follow-up care, content access, AI-assisted support, and platform administration.

3.3 Data Flow Diagrams

MindLift Data Flow Diagrams (DFDs) are used to show the flow of information between the platform and its users, internal processes, data stores and external services. At the most top level, the Patients, Doctors and Administrators interact with the MindLift system. The supporting services like the AI service, email service and Jitsi video platform provide added functionality. The Level 1 diagram decomposes the overall view into the major internal processes that control accounts and availability, perform assessments and doctor recommendations, manage appointments and payments, provide consultation and follow-up care, provide AI-assisted emergency support, and conduct administrative monitoring. Overall, the DFDs provide a good picture of how MindLift receives, modifies, stores and returns data, all while staying connected to its core mental health and counseling services.

3.3.1 Level 0 Data Flow Diagram

The Level 0 DFD provides a high-level overview of the entire MindLift system. It addresses the information exchange between the central platform, the three main user roles and the external services that support specific functions.

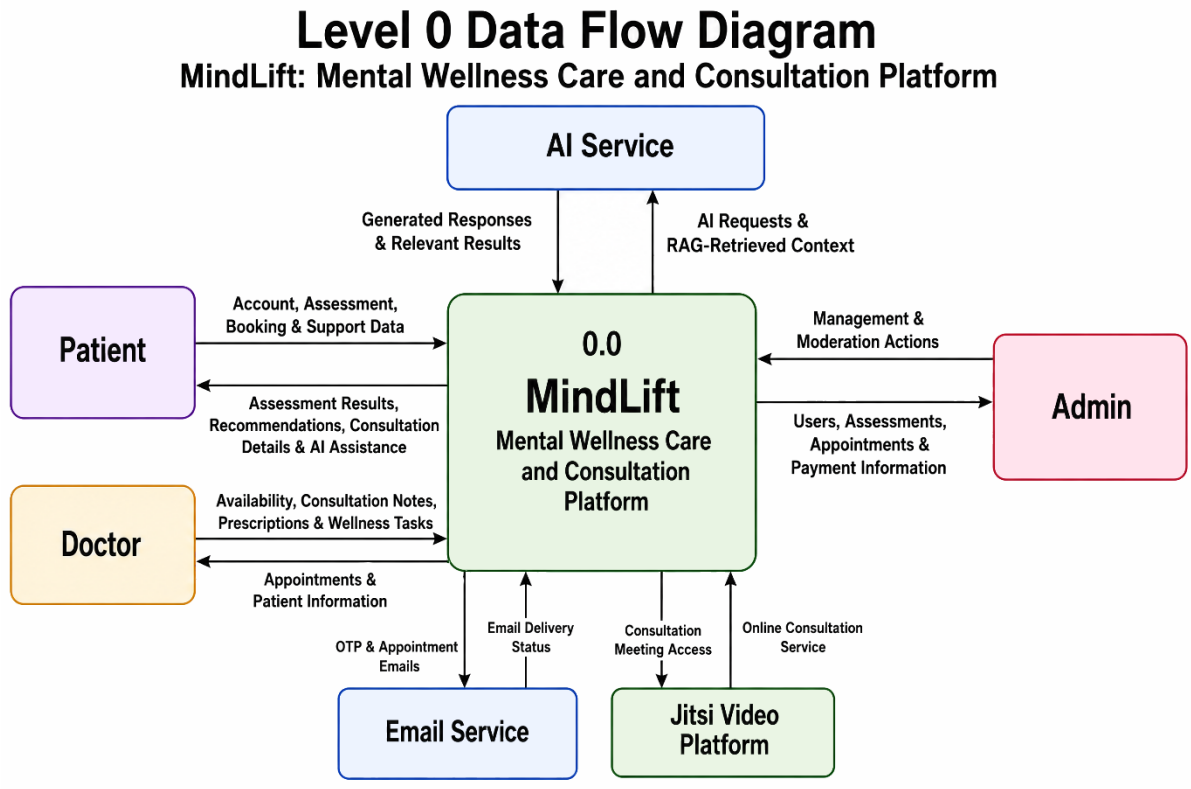


Figure 3.3: Level 0 Data Flow Diagram

Patients provide account information, responses to mental wellness assessment, booking information and support queries. In return, MindLift offers assessment results, doctor recommendations, consultation-related information and AI-assisted guidance. Doctors provide availability info, consultation notes, prescriptions, and wellness tasks, and get appointment and relevant patient info needed for consultation and follow-up care.

Administrators interact with the platform through management and moderation activities. They are provided with relevant information concerning users, assessments, appointments, and payments so that the overall operation of the platform can be monitored and managed from the administrative side.

MindLift also communicates with several external services. The AI service receives AI requests together with relevant RAG-retrieved context and returns generated responses or other relevant results. The email service handles OTP and appointment-related emails and returns delivery information to the platform. The Jitsi video platform provides the meeting environment required for online doctor-patient consultations.

The main data exchanges represented in the Level 0 DFD include:

- Patient → MindLift: Account, assessment, booking, and support data
- MindLift → Patient: Assessment results, doctor recommendations, consultation details, and AI assistance
- Doctor → MindLift: Availability, consultation notes, prescriptions, and wellness tasks
- MindLift → Doctor: Appointments and patient information
- Admin → MindLift: Management and moderation actions
- MindLift → Admin: User, assessment, appointment, and payment information
- MindLift ↔ AI Service: AI requests, RAG-retrieved context, generated responses, and relevant results
- MindLift ↔ Email Service: OTP and appointment emails together with delivery status
- MindLift ↔ Jitsi Video Platform: Consultation meeting access and online consultation service

The Level 0 diagram therefore provides a simplified view of MindLift as one complete system, showing the major information exchanged with users and external services without describing the internal processing in detail.

3.3.2 Level 1 Data Flow Diagram

The Level 1 DFD provides a more detailed view of how the major functions of MindLift process and exchange information. The overall system is divided into six main processes: Manage Accounts & Doctor Availability, Process Assessment & Doctor Recommendations, Manage Appointments & Payments, Manage Consultation & Follow-up, AI & Emergency Support, and Admin Monitoring & Moderation. Each process communicates with the relevant users, data stores, or external services according to its responsibility.

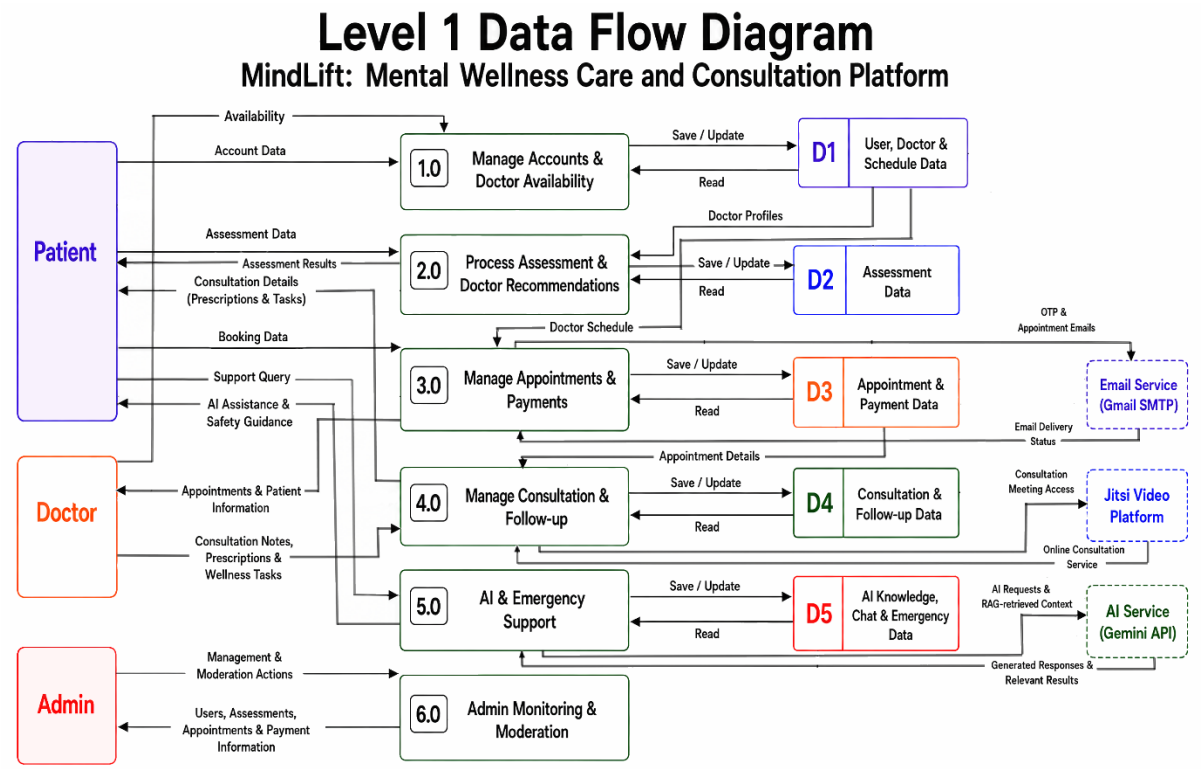


Figure 3.4: Level 1 Data Flow Diagram

Account & Doctor Management

This process takes care of the information of the account and the availability of the doctor. The doctor and schedule information is pulled from the respective data store as needed and profiles and consultation availability is maintained.

Doctor recommendations & process evaluation

Assessment answers from patients are used to calculate assessment results and to give personalized doctor recommendations. Assessment information is stored in the assessment data store, while relevant doctor profile and schedule information is retrieved from the user, doctor, and schedule data store when needed for the recommendation process.

Manage Appointments & Payments

This process takes care of booking information, doctor schedules, appointment records, information related to payments and appointment confirmation. It talks to the appointment and payment data store and uses email service for OTP and appointment related communication.

Manage Consultations & Follow-up

Consultation activities are managed through this process. Consultation notes, prescriptions and wellness tasks assigned by the doctor provide the follow-up information. Appointment details provide the consultation support. Jitsi offers access to meetings and the environment for online video consultations.

AI & Emergency Support

This process receives support queries from patients and provides AI-supported advice or safety-focused answers. The corresponding data store contains the relevant AI knowledge, chat information and emergency related records. If generative AI is required, the RAG context and user prompt are sent to the AI service, which responds with generated responses or relevant outputs.

Admin Monitoring & Moderation

This process enables administrators to perform management and moderation activities. It provides access to core platform data such as users, assessments, appointments and payments, as well as enabling the wider supervision of MindLift.

The Level 1 DFD shows how MindLift's major processes, data stores, users, and external services work together to support assessment, doctor recommendation, appointment management, consultation, follow-up care, AI-assisted support, and administrative monitoring within one coordinated system.

3.4 Activity Diagrams

The Activity Diagrams illustrate the complete activity flows of MindLift for Patients, Doctors and Administrators. Patients can register, login, fill mental wellness assessments, get doctor recommendations, book appointments, make payments and do OTP verification, join online consultations via Jitsi, get prescriptions and wellness tasks, explore mental wellness blogs and get AI supported help. Doctors can schedule appointments, have consultations, review patient records, provide prescriptions and wellness tasks, manage their profile and availability, and publish or update blog content. Administrators handle patients and doctors, monitor appointments and assessments, keep track of payments and revenue and moderate blogs. Taken together, the diagrams show how Patients, Doctors, and Administrators carry out their activities in the MindLift system.

3.4.1 Patient Activity Diagram

The Patient Activity Diagram presents the main sequence of actions a patient can perform in MindLift, beginning with registration or login and continuing through assessment, doctor recommendation, appointment booking, online consultation, follow-up care, blog access, and AI-assisted support.

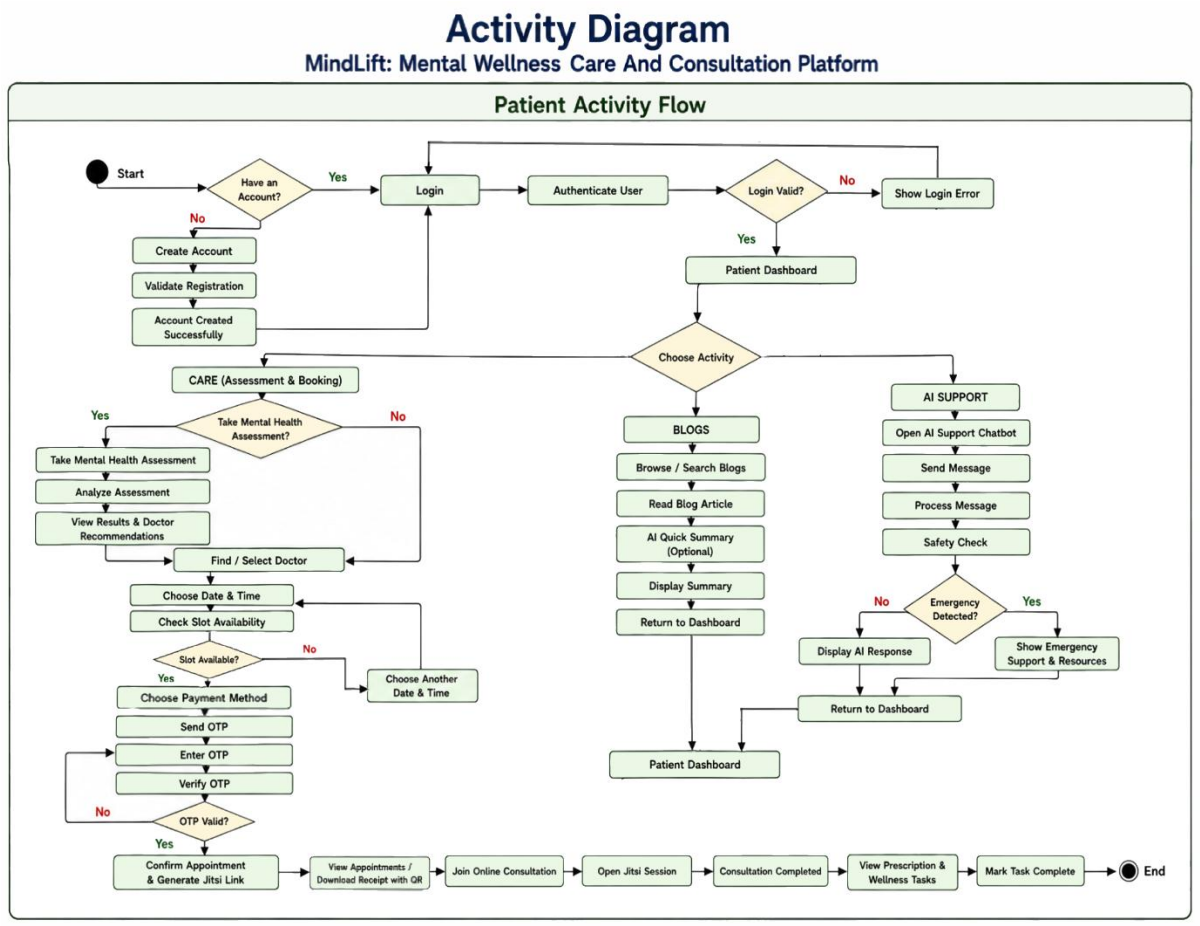


Figure 3.5: Patient Activity Diagram

The Patient Activity Diagram depicts the full range of activities that a patient can engage in on MindLift from creating or logging into an account to the Patient Dashboard. The patient can then select between care-oriented activities, access to the blog, and AI-based support. The patient can take a mental wellness assessment, see the results and personalized doctor recommendations, choose a suitable doctor, pick an available slot for consultation, make the payment and OTP verification, confirm the appointment, download a QR enabled receipt and join the scheduled Jitsi consultation. After the consultation the patient can view the digital prescriptions and assigned wellness tasks and mark tasks as completed. The diagram also illustrates how patients can browse or search mental wellness blogs, use the AI Quick

Summary, and engage with the AI Support Chatbot, which conducts safety checks and provides emergency support resources when a potential crisis is detected.

3.4.2 Doctor and Admin Activity Diagram

The Doctor and Admin Activity Diagram outlines the principal activities carried out by both professional roles within MindLift. The doctors are mainly responsible for the consultation, patient follow-up, profile and availability management, blog activities. The role of the administrator is to maintain the system, administer users, track appointments and assessments, manage financial records, and moderate content. Here is a diagram of their activity flows:

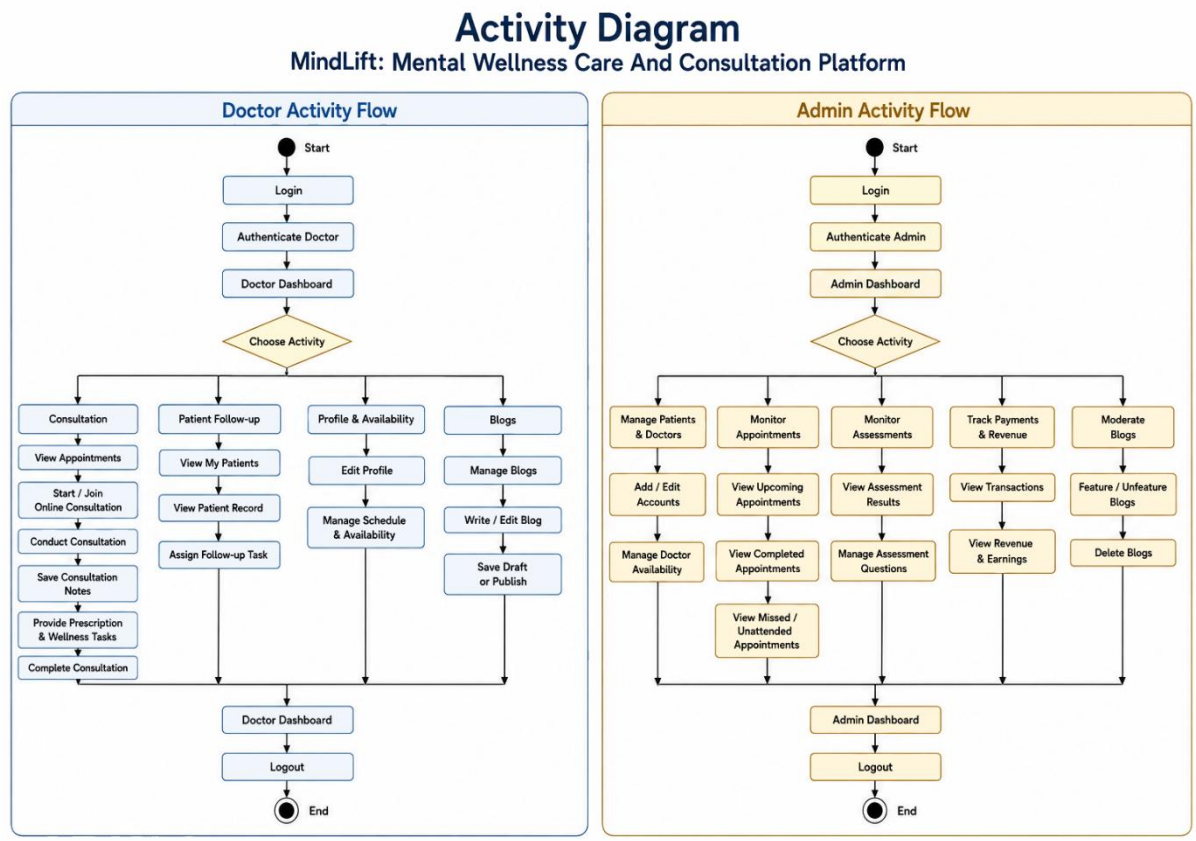


Figure 3.6: Doctor and Admin Activity Diagram

Doctor Side Flow

The Doctor Activity Diagram presents the main professional activities performed by doctors in MindLift. The doctor, after login and authentication, reaches the dashboard and can select from consultation, patient follow-up, profile and availability management and blog related

activities. In the consultation flow, doctors can view appointments, create or join scheduled online sessions, perform consultations, save consultation notes, give digital prescriptions, assign wellness tasks and complete the consultation process. The patient follow-up section allows doctors to see their patients, check relevant patient records, and set follow-up tasks if needed. Doctors are also able to edit their professional profile, manage schedules and availability so that patients can view accurate consultation information. Furthermore, the blog section allows doctors to create a new post and edit the existing ones. They can write or edit articles and save as drafts or publish for users.

Admin Side Flow

The Admin Activity Diagram presents the main administrative and monitoring activities carried out within MindLift. After the login and authentication, the admin enters the Admin Dashboard where he can manage the patients and doctors, monitor appointments and assessments, track payment and revenue, and moderate blog content. The user-management flow allows the administrators to add/edit accounts as well as manage doctor availability when needed. Monitoring of Appointments consists of checking of the upcoming consultations, finished appointments and the cases which are marked as missed or unattended. Administrators can also view assessment results and manage assessment questions used by the platform. On the financial side, they can review transactions together with revenue and earnings information. Blog moderation allows administrators to feature or unfeature selected posts and remove inappropriate or unnecessary content. These activities help the administrator maintain the operational, financial, and content-related areas of MindLift from one centralized management environment.

3.5 Class Diagram

The Class Diagram shows the structure design of MindLift, which explains the main classes, attributes, basic operations, and their relations. It explains how core entities such as User, Doctor, PatientAssessment, Appointment, Consultation, Payment, Prescription, DailyTask, BlogPost, ChatSession, ChatMessage, EmergencyLog, and TaskCompletion interact to provide the core features of the platform. The diagram also indicates the association of assessment information with the users, the association of doctors with schedules and appointments, the association of consultations with prescriptions and follow-up wellness tasks, and the association of chatbot sessions with messages and emergency-related records. Thus, the class structure reveals a clear picture of MindLift's data and functional relations in a single integrated system.

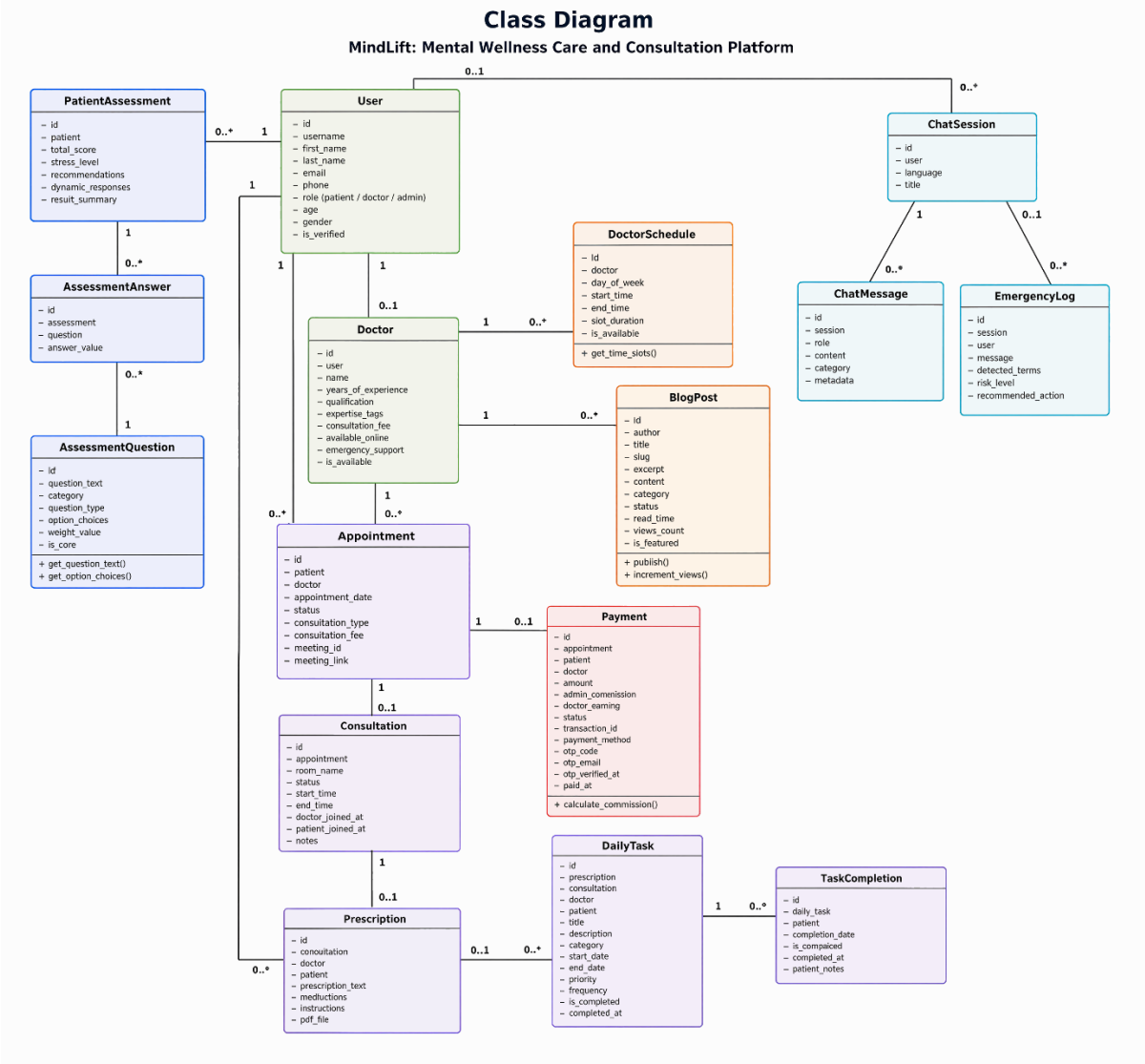


Figure 3.7: Class Diagram

The diagram mainly focuses on the structural relationship between users, doctors, assessments, appointments, consultations, payments, prescriptions, wellness tasks, blog content and AI-assisted support system. The User class is the main account entity for patients, doctors and administrators and the Doctor class extends the professional information needed for doctor-specific activities. Patients can do mental wellness assessments, get recommendations, schedule appointments, do payment verification, have consultations, get prescriptions, and follow wellness tasks assigned by doctors. Doctors can control their professional information and schedules, hold consultations, issue prescriptions and wellness tasks, and publish articles related to mental wellness. The chatbot part is represented by ChatSession, ChatMessage and EmergencyLog, which allow conversation history and emergency-related logs. The Class

Diagram shows how the main data entities of MindLift are organized and connected within the platform.

Important Information from the Class Diagram

User

- The User class stores fundamental account information such as username, first name, last name, email, phone, age, and gender.
- The role field distinguishes between Patient, Doctor, and Admin users.
- The is_verified field indicates the verification state of the user account.
- A user may have multiple assessments, appointments, prescriptions, and chatbot sessions depending on the activities performed within the platform.

Patient Assessment

- PatientAssessment stores the result of a patient's mental wellness assessment.
- It includes information such as total score, stress level, recommendations, dynamic responses, and result summary.
- A single user can complete multiple assessments over time.
- Assessment results are also used as part of the personalized doctor-recommendation process.

Assessment Question and Answer

- AssessmentQuestion stores the questions used during mental wellness assessments.
- It includes question text, category, question type, available choices, weight value, and whether the question is a core question.
- The operations get_question_text() and get_option_choices() support retrieval of question information and available response options.
- AssessmentAnswer stores individual patient responses to assessment questions.
- One assessment can contain multiple answers, while each answer is associated with a particular assessment question.

Doctor

- The Doctor class stores professional information including name, years of experience, qualification, expertise tags, consultation fee, availability status, and emergency-support availability.
- A doctor profile is associated with a user account.
- Doctors are connected with schedules, appointments, prescriptions, and blog posts.

- This class provides the professional information required for doctor discovery and assessment-based recommendation.

Doctor Schedule

- DoctorSchedule is the days and time periods of a doctor's availability for consultation. It contains information like day of week, start time, end time, slot duration and availability. A doctor can have many schedule records. The get_time_slots() operation is used to retrieve consultation slots from the stored schedule.

Appointment

- The Appointment class is for appointments between a patient and doctor for consultation. It contains information such as appointment date, appointment status, type of consultation, consultation fee, meeting id, and meeting link.
- A patient may have multiple appointments, while a doctor can also be associated with multiple appointments.
- The appointment serves as an important connecting entity between booking, payment, and consultation activities.

Consultation

- Consultation represents the actual consultation session associated with an appointment.
- It stores room name, consultation status, start and end times, doctor and patient joining times, and consultation notes.
- An appointment may lead to a consultation record.
- Consultation information provides the basis for subsequent prescriptions and follow-up care.

Payment

- The Payment class handles payment-related information associated with consultation booking.
- It stores the amount, admin commission, doctor earning, payment status, transaction ID, payment method, OTP code, OTP email, verification time, and payment time.
- An appointment can have zero or one associated payment record.
- The calculate_commission() operation supports calculation of the platform commission and doctor earning.

Prescription

- Prescription is connected with the consultation and follow-up care process.
- It contains prescription text, medicines, instructions, and PDF file information.

- Prescriptions are linked with both the doctor and patient.
- A prescription may also be associated with wellness tasks assigned as part of continued care.

Daily Task / Wellness Task

- DailyTask represents doctor-assigned wellness or follow-up activities within MindLift.
- It contains information such as title, description, category, start date, end date, priority, frequency, and completion-related information.
- A task may be associated with a prescription or consultation.
- These records allow follow-up guidance to remain connected with the patient's consultation process.

Task Completion

- TaskCompletion records the completion of an assigned wellness task by a patient.
- It stores completion date, completion status, completion time, and patient notes.
- A single wellness task can have multiple completion records where applicable.
- This structure allows patient task progress to be recorded separately from the original task assignment.

Blog Post

- BlogPost allows doctors to publish mental wellness content within MindLift.
- It contains information including title, slug, excerpt, content, category, publication status, read time, view count, and featured status.
- A doctor can publish multiple blog posts.
- The operations publish() and increment_views() support content publication and view tracking.

Chat System

- ChatSession represents a conversation between a user and the AI-assisted support system.
- A single session can contain multiple ChatMessage records.
- The ChatMessage contains information about the sender, the content of the message, its category, and metadata.
- These classes allow recent conversation information to be maintained so that chatbot interactions can preserve relevant context.

Emergency Support

- EmergencyLog records information related to emergency or high risk chatbot interactions. It has information such as the related session, user, message, detected terms, risk level and recommended action. Emergency logs enable the platform to keep structured records of risk-related situations that are detected during AI-assisted support. This class works with the chatbot's safety-aware processing to enable emergency-related handling in MindLift.

3.6 Sequence Diagrams

The Sequence Diagrams show the main MindLift workflows through the interactions between the Patient, MindLift System, Database, Doctor, Email Service, Jitsi Video Platform and AI Service. The first sequence includes registration and login, mental wellness assessment, doctor recommendation, booking appointment, payment and OTP verification, online consultation, prescriptions and wellness tasks. The second is about the blog browsing and semantic search, AI Quick Summary, and the AI Support Chatbot with conversation history, safety checks, emergency handling, RAG-based context retrieval, and AI-generated responses. The diagrams illustrate together how consultation, follow-up care, content discovery, and AI-powered support are coordinated across the components of MindLift.

3.6.1 Consultation Booking and Care Sequence Diagram

The Consultation Booking and Care Sequence Diagram shows how the Patient, MindLift System, Database, Email Service, Doctor, and Jitsi Video Platform interact throughout the consultation workflow, from assessment and booking to online consultation and follow-up care.

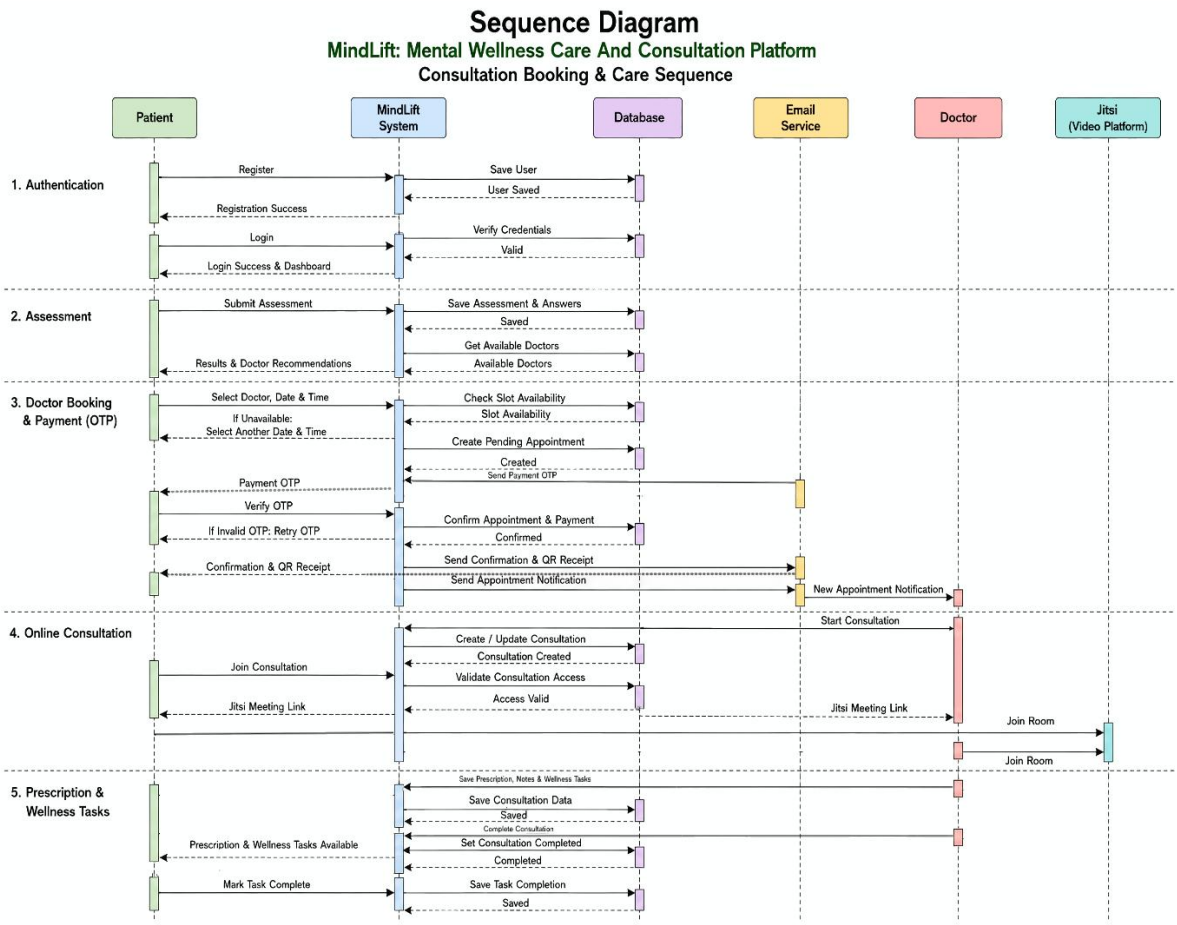


Figure 3.8: Consultation Booking and Care Sequence Diagram

Authentication

- The patient registers via MindLift and the user information is saved in the database.
- After registration the patient logs in and the credentials entered are verified by the system.
- After successful authentication of the patient, he is redirected to the dashboard.

Mental Wellness Assessment

- The patient submits a mental wellness assessment through the platform.
- MindLift stores the assessment and corresponding answers in the database.
- The system retrieves relevant available doctor information and returns the assessment results together with personalized doctor recommendations.

Doctor Booking and Payment Verification

- The patient selects a doctor, consultation date, and time.
- MindLift checks the database to see if the selected slot is available.
- If the slot is not available the patient will be asked to choose another date and time.
- When a slot is available, the platform creates the appointment and sends a payment OTP via the Email Service.
- The patient enters the OTP, and the system verifies it and confirms the appointment and payment.
- Once successfully confirmed, the patient gets confirmation of the appointment and receipt with QR-enabled and the doctor gets notification of the new appointment.

Online Consultation

- The doctor starts the consultation process and MindLift creates or updates the consultation record in the database.
- When the patient attempts to join, the platform validates consultation access.
- Once access is valid, the Jitsi meeting link becomes available.
- Both the patient and doctor join the same online consultation room through Jitsi.

Prescription and Wellness Tasks

- During or after the consultation, the doctor provides consultation notes, prescriptions, and wellness tasks.
- MindLift stores the consultation-related information in the database.
- The consultation then ends.
- After the session, the patient can see the prescription and assigned wellness tasks.
- Then the patient marks the wellness task completed. The information about the completion is saved in the database.

The sequence together illustrates how MindLift integrates authentication, assessment, doctor recommendation, appointment confirmation, online consultation, and follow-up care into one coordinated patient-care process.

3.6.2 AI Support and Blog Sequence Diagram

The AI Support and Blog Sequence Diagram shows how the Patient, MindLift System, Database, and AI Service interact during blog discovery, AI Quick Summary, and AI-assisted support.

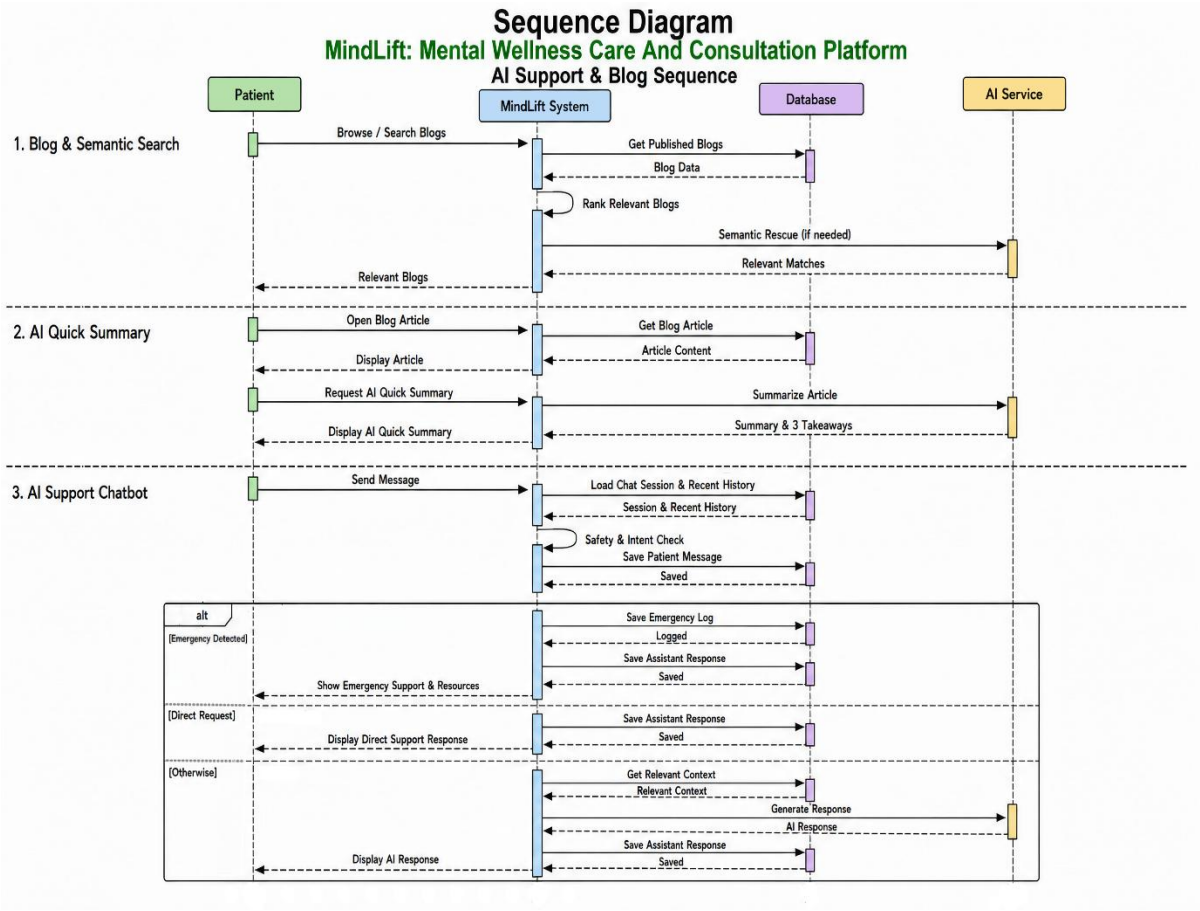


Figure 3.9: AI Support and Blog Sequence Diagram

Blog and Semantic Search

- The patient browses or searches mental wellness blogs through MindLift.
- The system retrieves published blog information from the database.
- MindLift ranks the available articles according to their relevance.
- If the local search result is not sufficient, semantic assistance is used to identify more relevant matches.

- The most relevant blog results are then displayed to the patient.

AI Quick Summary

- The patient opens a blog article through the platform.
- MindLift retrieves the selected article content from the database and displays it to the patient.
- The patient can request an AI Quick Summary for the article.
- The article content is sent to the AI Service for summarization.
- The AI Service returns a summary together with three key takeaways.
- MindLift displays the generated summary to the patient.

AI Support Chatbot

- The patient sends a message through the AI Support Chatbot.
- MindLift retrieves the applicable chat session and recent conversation history from the database. The system evaluates safety and intent , and determines how to respond to the message .
- The patient message is stored in the database.
- If an emergency is detected, MindLift records the emergency information and provides emergency support and relevant resources.
- If the message can be handled through a direct response, the system provides the appropriate support without requiring ordinary AI generation.
- For other requests, MindLift retrieves relevant RAG context from its stored knowledge.
- The relevant context and user request are supplied to the AI Service for response generation.
- The response is created and sent back to MindLift, stored in the database and displayed to the patient. The flow shows how MindLift uses local content retrieval, semantic assistance, AI-generated summaries, conversation context, safety processing, RAG retrieval, and generative AI for its blog and mental wellness support functions.

3.7 Entity-Relationship (ER) Diagram

Entity Relationship Diagram is the representation of the design of the database of MindLift. It contains the main entities, the key attributes of those entities and their relationships. It shows the relationships among users, doctors, assessments, appointments, payments, consultations, prescriptions, wellness tasks, blog posts, chat sessions, chat messages and emergency logs in the system. These relationships represent how MindLift manages the

patient's care, the consultation activities, follow-up information, content, payment records and AI-assisted support information in a coordinated database structure.

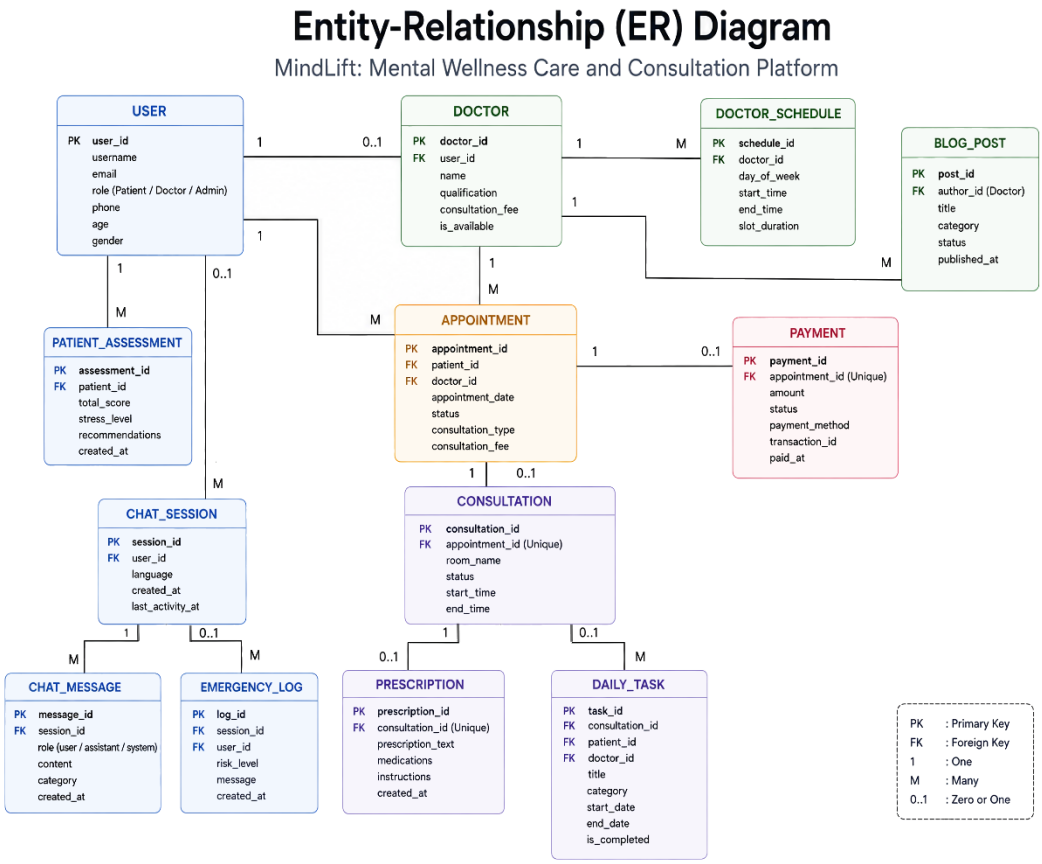


Figure 3.10: Entity-Relationship (ER) Diagram

The ER model of MindLift has several entities that represent specific parts of the platform's data and functionality. These entities work together to keep user information, consultation records, follow-up care, content, payments and AI-assisted support organized.

User

- USER is the central account entity of MindLift.
- It stores basic information such as username, email, role, phone, age, and gender.
- A user may be associated with a doctor profile, multiple patient assessments, appointments, and chat sessions.

Doctor

- DOCTOR stores professional information such as qualification, consultation fee, and availability.
- A doctor profile is linked with a user account.
- A doctor can maintain multiple schedules, handle multiple appointments, and publish multiple blog posts.

Patient Assessment

- PATIENT_ASSESSMENT stores assessment data on a patient.
- It includes total score, stress level, recommendations, and result-related data.
- A user can complete multiple assessments over time.

Doctor Schedule

- DOCTOR_SCHEDULE lists the available consultation days and time slots of a doctor. It contains information like day of week, start time, end time and slot duration.
- A doctor can maintain multiple schedule records.

Appointment

- APPOINTMENT connects a patient with a doctor for a consultation.
- It stores appointment date, status, consultation type, and consultation fee.
- An appointment may be associated with a payment record and a consultation record.

Payment

- PAYMENT stores the financial information related to an appointment.
- It includes the amount, status, payment method, transaction ID, payment time.
- An appointment can have zero or one associated payment record.

Consultation

- CONSULTATION represents the consultation session connected with an appointment.
- It stores room information, consultation status, start time, and end time.
- A consultation may lead to prescription and follow-up wellness task records.

Prescription

- PRESCRIPTION stores the prescription information provided after consultation.
- Includes prescription text, medicines and instructions. It is still connected to the related consultation.

Daily Task / Wellness Task

- DAILY_TASK means wellness activities prescribed by doctors as follow up.
- It stores consultation, patient, doctor, task title, category, start date, end date, and completion status.
- Multiple wellness tasks may be assigned as part of continued care.

Blog Post

- BLOG_POST stores doctor-authored mental wellness articles.
- It contains information such as title, category, status, and publication date.
- A doctor can publish multiple blog posts.

Chat Session

- CHAT_SESSION represents a user's conversation with the AI-assisted support system.
- A user can have multiple chat sessions.
- Each session can be linked with multiple chat messages and emergency-related records.

Chat Message

- CHAT_MESSAGE - individual messages that are exchanged in a chat session. It contains information like the sender's role, the message, the category and the creation time. A single chat session can contain multiple message records.

Emergency Log

- EMERGENCY_LOG contains information related to emergency or high risk chatbot interactions.
- It contains the associated session, user, message, and risk level.
- These records support the platform's safety-oriented emergency handling process.

Taken together, the ER Diagram shows how MindLift organizes and connects its major database entities to support assessment, doctor management, appointment booking, consultation, payment, follow-up care, mental wellness content, and AI-assisted support within one structured system.

3.8 Database Table Structure Diagram

Within the underlying database., MindLift keeps its core data as shown in the Database Table Structure diagram. Every table is a specific feature of the platform like Users, Doctors, Assessments, Appointments, Consultations, Payments, Prescriptions, Wellness Tasks, Blogs, Chat Sessions and Emergency Records. The diagram shows the relationship between these tables using keys and constraints to store related information in a consistent manner and to retrieve the same at the time of need.

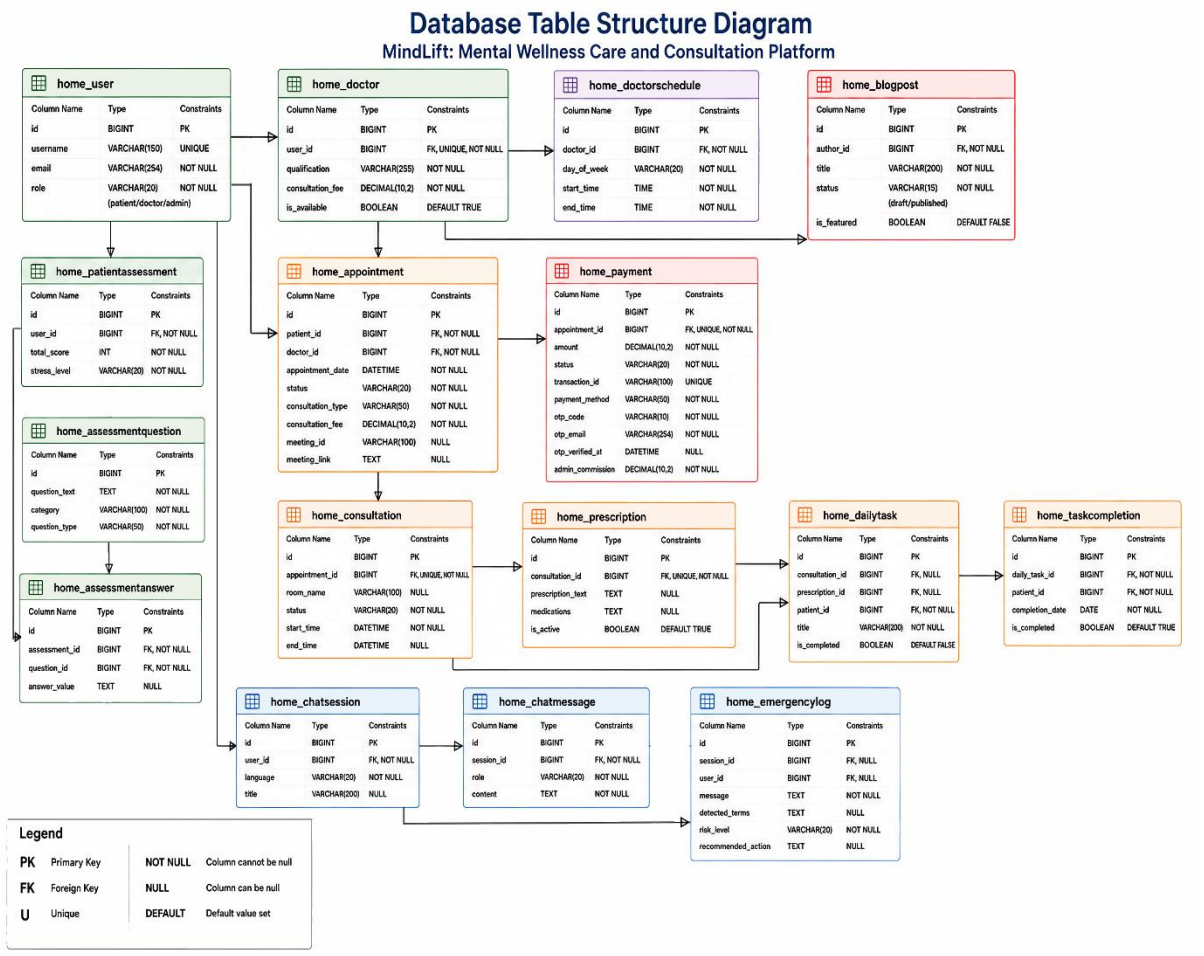


Figure 3.11: Database Table Structure Diagram

The MindLift database is structured with multiple related tables. Each table has information about a part of the platform. The relations between these tables ensure the connection of data from assessment, consultation, payment, follow-up care, blog and AI-support activities.

User Table

- home_user stores the basic account details of patients, doctors and admins. It has fields like user_name, email and role. The user record is linked to doctor profiles, patient assessments, appointments and chatbot sessions.

Doctor Table

- home_doctor stores professional information about doctors. It covers linked user account, qualification, consultation fee and availability status.
- Doctor records connect with schedules, appointments, and blog posts.

Doctor Schedule Table

- home_doctorschedule stores the working schedule of each doctor.
- It includes day of week, start time, end time, and slot-related information.
- These records are used when determining available consultation times.

Patient Assessment Tables

- home_patientassessment stores the main assessment result of a patient, including total score and stress level.
- home_assessmentquestion stores the questions used in the mental wellness assessment.
- home_assessmentanswer stores the answers submitted for individual assessment questions.
- Together, these tables maintain the assessment process from question presentation to result generation.

Appointment Table

- home_appointment contains information of consultation booking between patients and doctors. It includes appointment date, status, type of consultation, consultation fee, meeting ID and meeting link. The appointment record is an important link between doctor schedule, payment and consultation.

Payment Table

- The home_payment table stores payment information related to appointments.
- It includes amount, status, transaction ID, payment method, OTP information, verification time, and admin commission.
- The appointment relationship is unique so that one appointment is associated with at most one payment record.

Consultation Table

- home_consultation stores information about the actual consultation session.
- It includes the associated appointment, room name, consultation status, start time, and end time.
- Consultation records are linked with prescriptions and follow-up wellness tasks.

Prescription Table

- home_prescription stores prescription information created after consultation.
- It includes prescription text, medications, and active-status information.
- Each prescription remains linked with the consultation from which it was created.

Daily Task Table

- home_dailytask stores doctor-assigned wellness and follow-up tasks.
- It connects the task with the consultation, prescription, patient, and doctor.
- Task information includes title and completion status.

Task Completion Table

- home_taskcompletion logs completion of wellness tasks by patients.
- It holds the associated task, patient, completion date and completion status.
- This allows task progress to be kept separate from the original task assignment.

Blog Post Table

- home_blogpost stores mental wellness articles published by doctors.
- It includes the author, title, publication status, and featured status.
- Each blog post remains connected with its doctor author.

Chat Session and Message Tables

- home_chatsession stores chatbot conversation sessions associated with users.
- home_chatmessage stores the individual messages belonging to each session.
- MindLift uses these tables to store conversation data for AI assisted support.

Emergency Log Table

- home_emergencylog records emergency-related information encountered in conversations with chatbots.
- It can be associated with a chat session and a user.

- It logs data such as the detected message, terms, risk level and recommended action.

These tables together constitute the central data structure of MindLift. They allow information to be linked across modules through primary keys, foreign keys, unique constraints and other database rules.

CHAPTER 4

Conceptual Framework and System Design Principles

4.1 Iterative and Incremental Development in the Project

MindLift: Mental Wellness Care and Consultation Platform was developed by means of an iterative and incremental process, which practically means the system did not appear all at once. It was instead built in stages, with each stage producing something usable before proceeding forward to the next [17]. That kind of approach tends to make sense for a project like this, because needs can shift as the system takes shape, and some problems only become visible once the first working version is in front of you.

In this project, we decided to start out with the most basic functions: registration, login, and role-based access. As those became stable enough, development gradually moved toward the patient, doctor, and administrator dashboards, followed by mental wellness assessment, personalized doctor recommendation, appointment booking, payment handling, and email OTP confirmation. The consultation stage was then extended with Jitsi-based video sessions, digital prescriptions, and doctor-assigned wellness tasks. Later iterations introduced the mental wellness blog system, semantic search, AI Quick Summary, and the RAG-assisted mental wellness support chatbot.

The sequence was not just a planning choice; it was also a practical one. Mental health platforms are not the sort of systems where every requirement is perfectly clear from day one. In that sense, the development process had to leave space for correction and adjustment, which recent digital mental health design discussions also seem to support [18], [19].

This staged development made it easier to catch issues early. Something as small as a login redirect problem or a mismatch in appointment status might look minor on its own, but left unchecked it could affect the rest of the system. The same applies to assessment result display, doctor recommendation logic, double-booking risks, OTP verification, consultation availability, or inconsistent dashboard data. Resolving these issues progressively made the later integration of the platform more manageable and reduced the likelihood of several interconnected problems appearing at the final stage.

Development Phase	**Features Implemented**	**Main Activities**	**Outcome**
Phase 1	Authentication and Registration	Login, registration, logout, role assignment	Basic access control established
Phase 2	Role-Based Dashboards	Patient, doctor, and admin dashboards	Role-specific navigation implemented

Phase 3	Assessment and Recommendation	Mental wellness assessment, scoring, doctor recommendation	Personalized care-entry support added
Phase 4	Appointment and Payment	Doctor schedules, booking, payment, email OTP confirmation	Structured appointment workflow established
Phase 5	Consultation and Follow-Up	Jitsi consultation, consultation records, prescriptions, wellness tasks	Continuous consultation workflow enabled
Phase 6	AI-Assisted Support	RAG chatbot, safety handling, contextual knowledge retrieval	Intelligent mental wellness support introduced
Phase 7	Content and Administration	Blogs, semantic search, AI Quick Summary, content moderation, admin monitoring	Information access and management expanded
Phase 8	Testing and Refinement	Integration testing, bug fixing, workflow and UI refinement	Reliability and presentation readiness improved

Table 4.1: Development Iterations and Implemented Features

4.2 Modular Design in the Project Structure

The project was organized using modular design, which basically signifies the system was broken into smaller parts, each with its own responsibility. That may sound like a simple idea, but it makes a big difference once a project starts growing. A system with too many responsibilities packed into one place quickly becomes difficult to maintain, and even small changes can create unexpected problems [20].

For this platform, modularity was especially important because the system serves different users with different needs. A patient does not need the same tools as a doctor, and neither of them needs the same control panel as an administrator. Django's own structure supports this kind of separation through its models, views, templates, and admin features, which made the architecture of the project feel fairly natural [21].

In the implementation, the system is functionally separated into authentication, patient, doctor, administrator, assessment and recommendation, appointment and payment,

consultation and follow-up, blog and content, and AI-assisted emergency-support components. The separation keeps responsibilities clearer while still allowing information to move between related parts of the platform.

If something went wrong in appointment booking, for example, we did not have to search through the entire platform. We could concentrate on the part of the system that handled bookings and afterwards see how it connected to the related modules. The interconnection between these modules is also important because the platform does not treat each function as a completely separate feature. Authentication controls access to patient, doctor, and administrator areas. Assessment results contribute to personalized doctor recommendations. The appointment component connects patients with doctors and leads into payment and OTP confirmation. Consultation functionality relies on an existing appointment and then allows prescriptions and wellness tasks to become part of follow-up care. Blog content is linked with semantic search and AI Quick Summary, while the chatbot combines platform knowledge, RAG retrieval, safety controls, and generative AI. This illustrates that modularity in the project is not merely about dividing files; it concerns keeping responsibilities separate while still permitting the service flow to work as one connected system.

Module Name	**Responsibility**	**Inputs**	**Outputs**	**Interconnection**
Authentication Module	Handles registration, login, and access	User credentials	Authenticated session	Connects all role-based areas
Patient Module	Supports assessment, appointments, prescriptions, and wellness tasks	Patient actions	Care-related information	Connects assessment, doctors, appointments, and consultation
Doctor Module	Manages profiles, availability, appointments, consultations, and content	Doctor data and actions	Schedules, prescriptions, tasks, articles	Connects patients, appointments, consultation, and blogs
Admin Module	Oversees platform operations	Platform records	Monitoring and management information	Connects major administrative functions
Assessment and Recommendation Module	Processes assessment responses and doctor recommendations	Patient answers	Scores, severity information, recommended doctors	Connects patients with doctor discovery

Appointment and Payment Module	Handles scheduling, booking, payment, and OTP confirmation	Doctor, date, time, payment information	Confirmed appointment and payment record	Connects patient, doctor, consultation, and admin
Consultation and Follow-Up Module	Supports Jitsi sessions, prescriptions, and wellness tasks	Appointment and consultation information	Consultation and follow-up records	Connects doctor and patient care
Blog and Content Module	Manages mental wellness articles and discovery	Doctor-authored content and user queries	Articles, semantic results, AI summaries	Connects doctors, patients, and AI-assisted features
AI and Emergency Support Module	Provides contextual wellness support and safety handling	User query and retrieved knowledge	Guidance, RAG response, or safety response	Connects knowledge sources, AI service, and emergency support

Table 4.2: Modular Design of the Proposed System

4.3 User-Centered Design in the Project Interface

Because this platform is meant to support mental wellness, user-centered design was never really dispensable; it was absolutely vital. People coming to a system like this may already be stressed, overloaded, or emotionally worn down, so the interface needs to greet them graciously rather than demand extraneous effort. An ample amount of research on digital mental health systems keeps circling back to the same point: usability, privacy, accessibility, and a sense of safety are not side concerns. They are often what decide whether people keep using the platform at all [19], [22].

That exact idea influenced the interface design in this project. Instead of giving every user the identical layout, the platform basically utilizes separate dashboards for patients, doctors, and administrators. Each role sees only what seems relevant to it. A patient can complete assessments, receive doctor recommendations, manage appointments, join consultations, access prescriptions and wellness tasks, read mental wellness articles, and reach emergency support. Doctors can manage availability, appointments, consultations, prescriptions, wellness tasks, and their published articles. Administrators are provided with a broader management environment for users, doctors, appointments, payment records, and platform content.

Simplicity was another priority. Mental health tools can feel exhausting when someone is already under pressure, so the navigation was kept clear and the steps were reduced as much as possible. Assessment questions are presented progressively rather than forcing the user to process everything at once. Appointment and consultation actions appear within their relevant context, while emergency support remains readily accessible. The platform seeks to cut down unwanted friction during user interactions by emphasizing accessibility and ease of navigation.

UCD Principle	**Project Application**
Role-Specific Design	Separate dashboards and functions for patients, doctors, and administrators
Simplicity	Clear assessment, booking, consultation, and follow-up workflows
Safety Visibility	Accessible emergency support and immediate-help resources
Reduced Cognitive Load	Focused assessment steps and task-oriented pages
Engagement	Wellness tasks, doctor-authored content, and AI-assisted wellness resources
Accessibility	Responsive layouts, readable typography, and clearly visible controls

Table 4.3: Application of UCD Principles in Platform Design

4.4 Task-Technology Fit in the Mental Wellness Consultation Process

Task-Technology Fit explains whether a system's functions are suitable for the tasks users need to perform [23]. In this project, the platform was designed around a practical mental wellness service process rather than as a simple informational website. The main tasks include mental wellness assessment, personalized doctor recommendation, appointment booking, payment and OTP confirmation, online consultation, prescription management, follow-up wellness tasks, mental wellness information discovery, AI-assisted support, and administrative monitoring.

The patient side of the system is mainly concerned with help-seeking and continuity of care. A patient can register, complete an assessment, receive suitable doctor recommendations, browse doctors, book an appointment, complete the required booking confirmation process, attend an online consultation, receive a prescription, and complete assigned wellness tasks. Patients can also browse doctor-authored mental wellness articles, use semantic search to locate relevant content, obtain AI-assisted article summaries, and access the mental wellness support chatbot when needed.

The doctor side supports a different type of work. Doctors can maintain professional information and availability, review appointments, participate in consultations, record consultation information, create prescriptions, assign wellness tasks, and manage their blog posts. Administrators have broader supervision through user, doctor, appointment, payment, and content-management functions.

The data model upholds this task fit too. The appointment structure associates patients, doctors, appointment dates, consultation fees and appointment status information. Payment records remain associated with the corresponding booking, while prescription and wellness-task records allow professional guidance to continue after consultation. The content system similarly links doctor-authored articles with search and AI-assisted summarization.

The platform moreover supports Task-Technology Fit through its care-continuity structure. Assessment can lead toward doctor recommendation, booking leads toward consultation, and consultation can lead toward prescriptions and follow-up wellness activities. Semantic article search and AI Quick Summary support information-discovery tasks, while the RAG-assisted chatbot provides a separate contextual support channel. Consequently, the system design presents mental wellness support as a connected process extending from initial help-seeking to professional consultation and continued care.

4.5 Cognitive Load and Interface Simplicity

Cognitive load is the amount of mental effort required to process information while performing a task [24]. Cognitive Load Theory states that the human working memory has a limited capacity. Interfaces should therefore avoid unnecessary complexity and present information in a way that users can understand without excessive mental effort. This is important, because people who already have emotional or mental strain may use a mental wellness consultation platform. If the interface requires too much effort, users may find it harder to complete important tasks.

The platform reduces this burden mainly by separating activities into focused pages and role-based dashboards. The patient interface does not place assessment, doctor selection, appointment booking, payment, prescriptions, and wellness-task completion into a single screen. Instead, each activity has its own page or workflow. This makes the experience easier to follow because the user is usually dealing with one main decision at a time.

The same pattern appears in the doctor interface. Appointment review, availability management, patient information, blog writing, and consultation work are separated, so doctors are not forced to handle unrelated activities within one crowded interface.

The patient dashboard provides a central view of care-related information such as upcoming consultations, current wellness tasks, assessment-related information, and recent prescriptions. This means patients don’t have to repeatedly search through unrelated areas of the platform and creates a clearer picture of what has happened, what is scheduled and what may require attention.

Prior research on digital health systems has found that poor information retrieval, inefficient interface design, and unclear workflows can adversely affect user experience [25]. MindLift's dashboard and navigation structure try to avoid some of these problems by presenting related information in recognizable groups and keeping unrelated operations apart.

On a more generic interface level, the platform is biased toward simplicity, giving each workflow a well-defined space. Patient dashboards focus on care activities, doctor interfaces on duties, and administrative interfaces on tracking and management. Such separation helps users orient themselves more swiftly and reduces unnecessary mental effort while navigating the system.

4.6 Progressive Disclosure in User Interaction

Progressive Disclosure means showing users the most relevant choices first and revealing more detailed options only when needed [26]. This theory fits the system due to the fact that the platform contains several complex workflows, but most users do not need to see every option immediately.

The public-facing part of the system begins with general pages such as the homepage, doctor information, blogs, and emergency support. More specific functions become available once a user signs in. The visible options then change according to role. Patients gain access to assessments, recommendations, appointments, prescriptions, wellness tasks, and dashboard functions. Doctors receive access to appointment, availability, consultation, patient, blog, and dashboard functions. Administrators receive access to the relevant management areas. This gradual exposure helps prevent the interface from becoming overloaded at the first point of contact.

The consultation journey also follows this idea. A patient may first complete an assessment and receive appropriate doctor recommendations or browse available doctors. After selecting a doctor, the user chooses an available appointment slot and proceeds through the required booking and payment steps. Email OTP confirmation is introduced only when the booking reaches that stage, while consultation tools become relevant only when a valid appointment exists.

The doctor availability feature is another example. The system does not need to display every schedule for every doctor at once. A patient selects the relevant doctor and appointment context, after which the available slots are presented. This keeps detailed scheduling information hidden until it becomes useful and reduces the number of unnecessary choices presented at one time.

Progressive disclosure appears in the doctor workflow as well. Prescription and wellness-task functions are associated with consultation-related activities rather than appearing as unrelated controls throughout the system. This keeps professional actions within an appropriate context and reduces the chance of users encountering functions before they become relevant.

In essence, Progressive Disclosure in this project is not only a visual design choice. It is rooted within the navigation flow, authentication logic, role separation, and the order of patient-care activities.

4.7 Self-Determination and User Engagement

Self-Determination Theory explains motivation through three basic psychological needs: autonomy, competence, and relatedness. Autonomy refers to the user's feeling of being free to choose and in charge of their actions. Competence is the feeling of being able to understand what is going on and to do tasks well. Relatedness is about feeling close to other people or being supported by a social relationship [27]. These three ideas can be helpful to understand how the platform can help the user engagement. More than data collectors, mental wellness systems need to help people feel in control, track their progress, and connect them to support.

The platform provides patients with multiple ways to engage with it, facilitating autonomy. A patient may complete an assessment, view doctor recommendations, browse professionals, read mental wellness articles, book an appointment, view prescriptions, complete wellness tasks, or access emergency support. The system does not pressure every user through one strict path. Because people look for mental wellness support in different ways and at different levels of readiness.

Through feedback and structured follow-up, competence is strengthened. The assessment results convert user responses into meaningful severity information, and recommendations. Appointment information lets users know the status of their consultation. After the professional consultation, the prescriptions and wellness tasks prescribed by the doctor give more concrete next steps. Wellness tasks can help patients feel like follow-up care is actionable, rather than just general advice. Research on digital mental health interventions suggests that feedback, personalization, ease of use and reminders can promote engagement [28].

The doctor-patient relationship is about relatedness. Automated information is not only being provided to patients. They can search for doctors and make appointments with specific doctors. They can attend video sessions and get prescriptions or wellness tasks from their doctors. This creates a sense of mediated human support within the digital environment. Doctor-authored mental wellness articles may also extend professional guidance beyond scheduled consultation sessions.

Self-Determination Need	**System Support**
Autonomy	Choice among assessments, doctors, articles, appointments, and support resources
Competence	Assessment feedback, doctor recommendations, appointment information, prescriptions, and wellness-task completion

Relatedness	Doctor profiles, online consultations, follow-up guidance, and doctor-authored content

Table 4.4: Application of Self-Determination Theory in the System

4.8 Error Prevention and Recovery in Core Workflows

Error prevention and recovery matter here because a mistake in the wrong place can chip away at trust, or even interrupt care. Registration, password management, appointment booking, payment, schedule creation, prescriptions, and the everyday tasks people return to often all need some guardrails against invalid input, confusing steps, or broken states. In a mental health setting, the stakes feel a little higher than in a typical app; users may already be under stress, so the system should try to catch errors early and make recovery feel simple rather than punishing. Clear guidance, validation mechanisms, and recovery options can help users complete tasks successfully while reducing frustration and minimizing the impact of errors [29].

The registration process prevents common mistakes before the account is set up. The information submitted is validated, password requirements are checked and any existing account information can be detected before duplicate records are created. If the validation fails, the user stays within the form, and is given an appropriate message, instead of having invalid information stored silently.

Doctor availability also falls under prevention logic . When schedule information is created or updated, the system checks the relevant time information and prevents conflicting availability. When booking an appointment, the platform checks doctor schedules and current reservations so that a time slot already assigned to another patient is not re-selected. Invalid booking options are appointment times in previous days or otherwise unavailable.

There is one more step of validation in the booking process with email OTP confirmation. Instead of just taking it as done when the information is submitted, the email entered is verified with a one time password to finally confirm the appointment. This reduces the chance that a booking is successfully completed with an invalid or unavailable email address and provides a more obvious point of verification in the workflow. Structured status handling is also used for appointment/consultation records, to distinguish between different outcomes instead of using the same state. This will result in a cleaner presentation on the dashboard and easier determination of whether a consultation is upcoming, completed, or requires another appropriate outcome based on session activity by the platform.

The prescription and wellness-task workflow is not based only on informal messages, but on structured records. Doctors can record prescription information and assign specific follow-up wellness tasks, while patients can view and complete those tasks through their own interface. This preserves follow-up information in an organized form and reduces the likelihood of important guidance becoming scattered across unrelated parts of the system.

The AI-assisted support system applies a different kind of error prevention. Safety and emergency checks are performed before ordinary generative responses, and local fallback behaviour remains available if the external AI service cannot provide a usable response. This prevents the chatbot from relying on a single external service for every type of interaction and gives sensitive messages a controlled processing path.

Across its main workflows, MindLift therefore applies validation, slot-conflict prevention, email OTP verification, structured records, role-based interfaces, safety-first chatbot processing, and fallback behaviour to reduce avoidable errors. These mechanisms contribute to a more dependable experience while keeping recovery and corrective actions understandable for patients, doctors, and administrators.

CHAPTER 5

TESTING

5.1 Testing Strategy

Testing was performed to verify the main functions and workflows of MindLift. We built the automated test suite using Django's built-in testing framework, TestCase and SimpleTestCase. When you run tests , Django creates a separate database for testing . This allows you to run tests without affecting your main application data .

The tests cover major areas of the platform, including assessment and doctor recommendation, authentication, appointment booking and OTP verification, consultation access, wellness tasks, chatbot reliability, semantic blog search, AI Quick Summary, and administrative functions.

5.2 Automated Testing

The current home/tests.py contains 100 automated test cases. These tests examine both individual components and interactions between related modules.

The principal areas covered include:

- Assessment and personalized doctor recommendation
- Authentication and role-based access
- Appointment booking, payment, and email OTP verification
- Consultation access and session handling
- Prescription and wellness-task related behaviour
- AI Support Chatbot safety and fallback behaviour
- Semantic blog search
- AI Quick Summary
- Administrative functions

Test Area	Main Verification
Assessment	Assessment questions, submission, and result processing
Recommendation	Personalized doctor matching
Authentication	Login, role access, and recovery behaviour
Booking and OTP	Appointment validation and email OTP confirmation
Consultation	Authorized access and consultation-state handling
Wellness Tasks	Assignment and completion behaviour
AI Support Chatbot	Safety checks, fallback, and invalid-input handling
Blog Search	Search relevance and content visibility
AI Quick Summary	Patient access, summary generation, and safe failure
Administration	Selected administrative access and management behaviour

Table 5.1: Automated Test Coverage

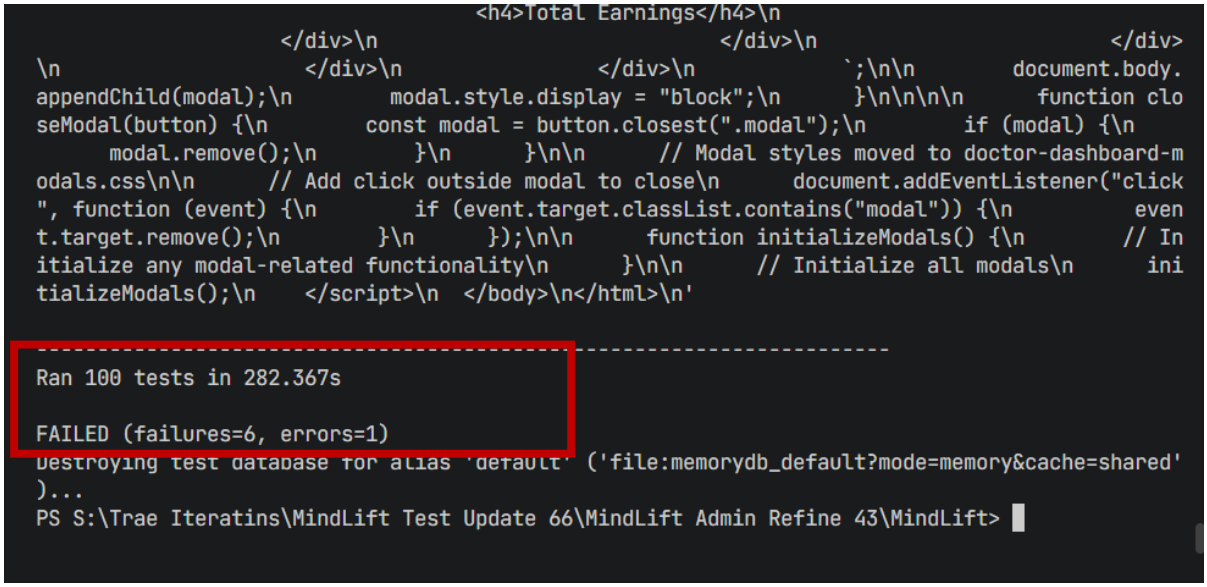


Figure 5.1: Django Automated Test Execution Result

5.3 Test Execution and Results

The complete automated suite was executed using:

python manage.py test home.tests --verbosity=2

The latest execution produced the following result:

Metric	Value
Total Tests	100
Passed	93
Failed	6
Errors	1
Pass Rate	93.0%
Execution Time	282.367 seconds

Table 5.2: Automated Test Results

5.4 Verified Functional Areas

The automated suite contains tests for several important MindLift workflows. Among the tested areas are the current 10-question mental wellness assessment, doctor recommendation, booking and OTP verification, role-based access, consultation authorization, wellness-task handling, chatbot fallback and safety behaviour, semantic blog search, and AI Quick Summary.

These tests provide repeatable verification of important application behaviour without depending only on manual observation of the interface.

5.5 Chapter Summary

This chapter presented the automated testing performed on MindLift using Django's testing framework. The current suite contains 100 tests covering the principal care, authentication, consultation, AI-assisted content, and administrative functions of the platform. In the latest execution, 93 tests passed, 6 failed, and 1 produced an error, resulting in a 93.0% pass rate. The test output provides direct evidence of the current testing status of the system.

CHAPTER 6

SYSTEM INTERFACE AND FEATURE DEMONSTRATION

6.1 Public Access and Authentication Interfaces

The public interface introduces the MindLift and gives access to the main entry functions of the platform before entering any role-specific workflow. In this section we show:

- **Home Page:** Public platform to discover MindLift, the available doctors, mental wellness resources, and emergency support for users.
- **Patient Registration:** New User Register and begin using the platform.
- **Different login portals:** For Patients, Doctors and Administrators as per their role.
- **Role based access control:** Prevents users from accessing wrong portal.
- **Successful authentication:** Automatically takes the app to the appropriate dashboard.
- **Different access paths:** So that each user role can access only the functions relevant to its responsibilities.

These interfaces are the first access to MindLift, before users go to assessment, consultation, professional care or administrative functions.

6.1.1 Home Page

The MindLift Home Page serves as the main public entry point to the platform. It introduces the care journey and provides direct access to registration, login, doctor discovery, mental wellness resources, and emergency support.

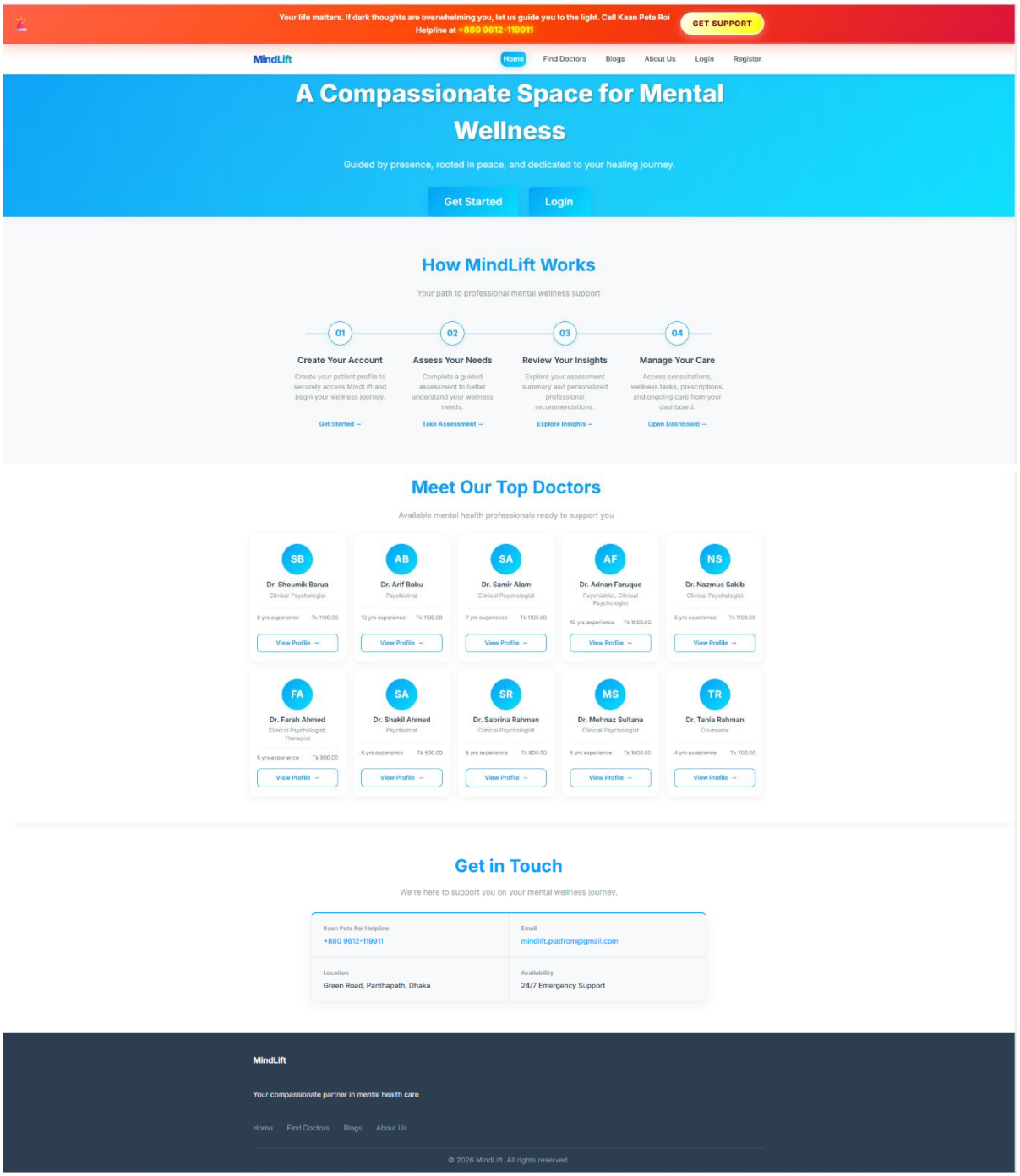


Figure 6.1: MindLift Home Page Interface

6.1.2 Patient Registration

The Patient Registration interface allows a new user to create a MindLift account by entering the required personal and account information.

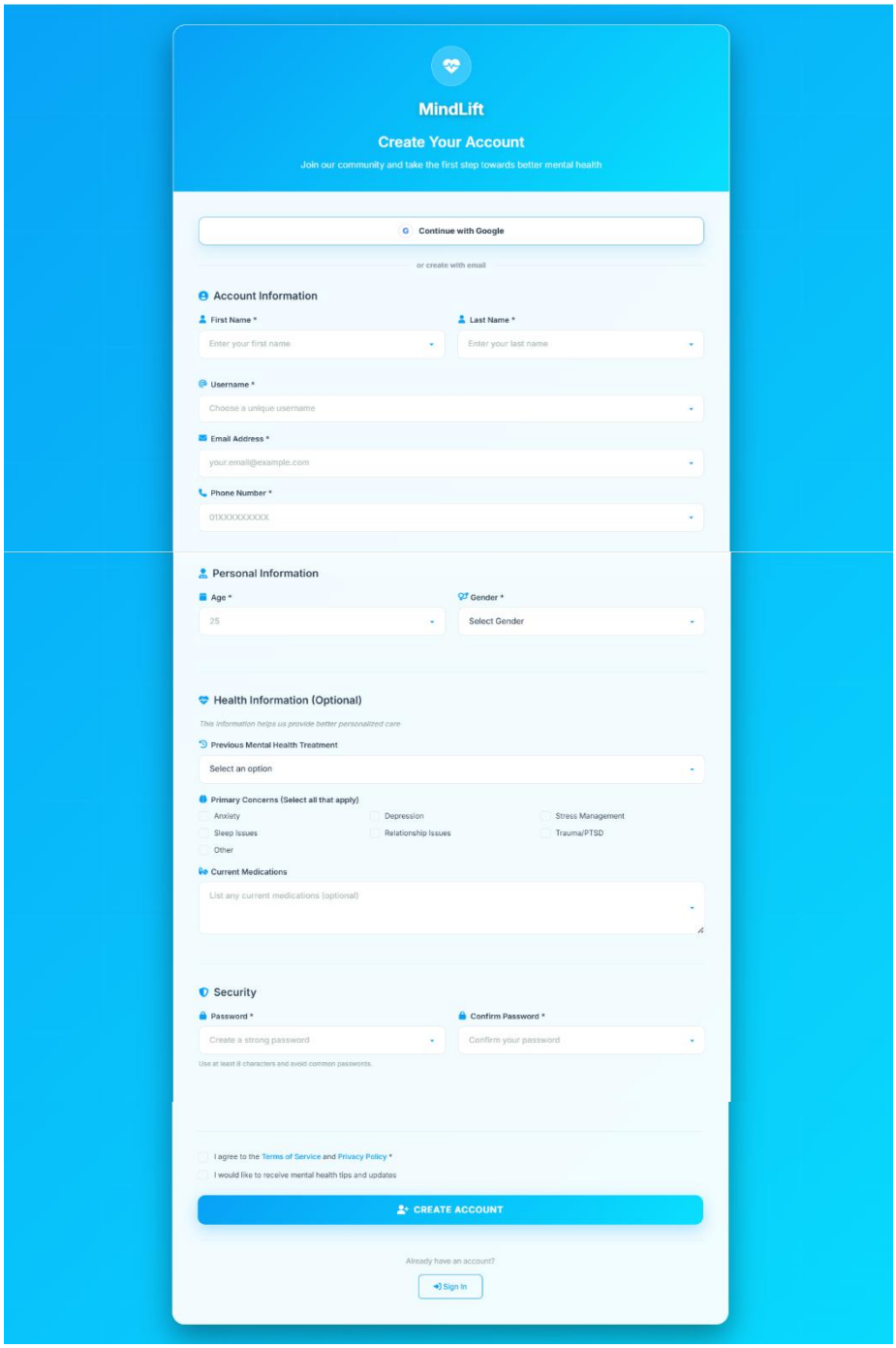


Figure 6.2: Patient Registration

6.1.3 Role-Based Login

MindLift provides separate login interfaces for Patients, Doctors, and Administrators, directing each user to the appropriate role-specific dashboard.

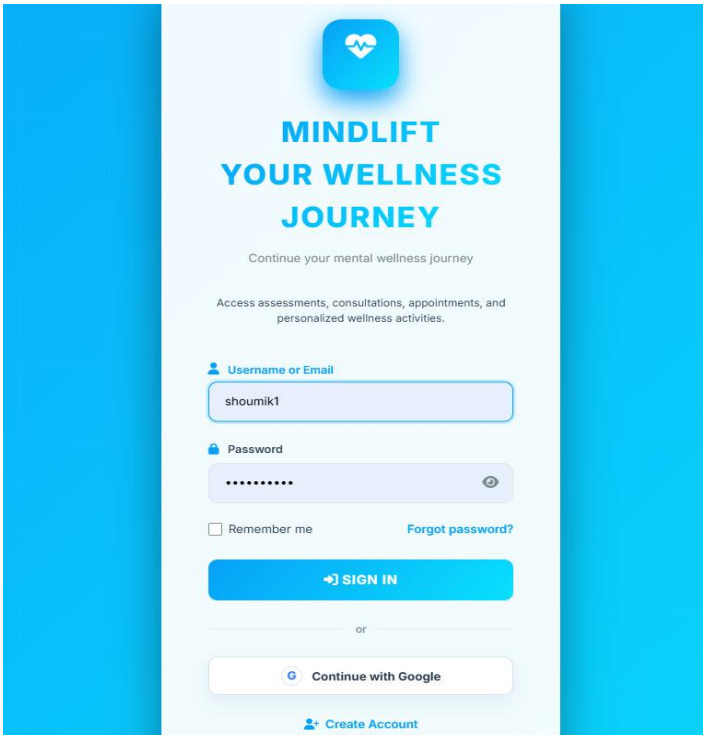


Figure 6.3: Patient Login

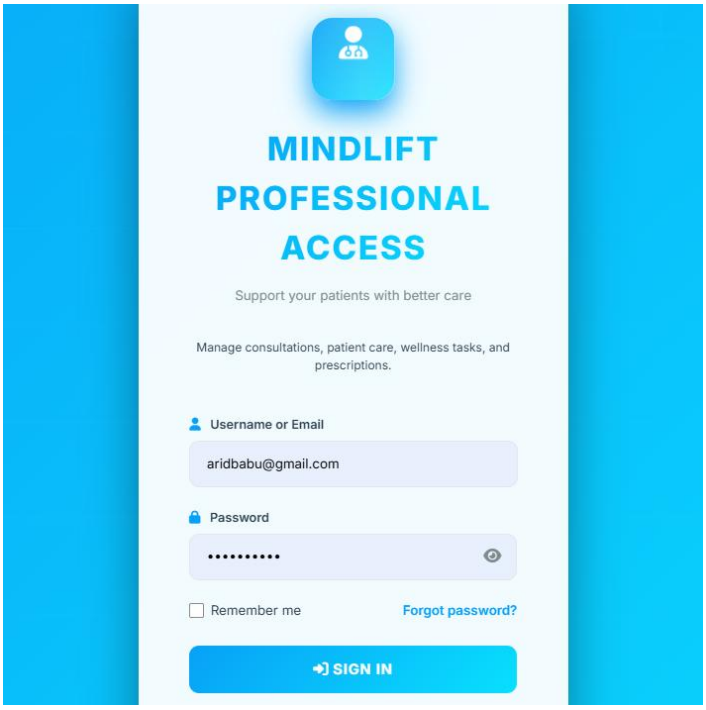


Figure 6.4: Doctor Login

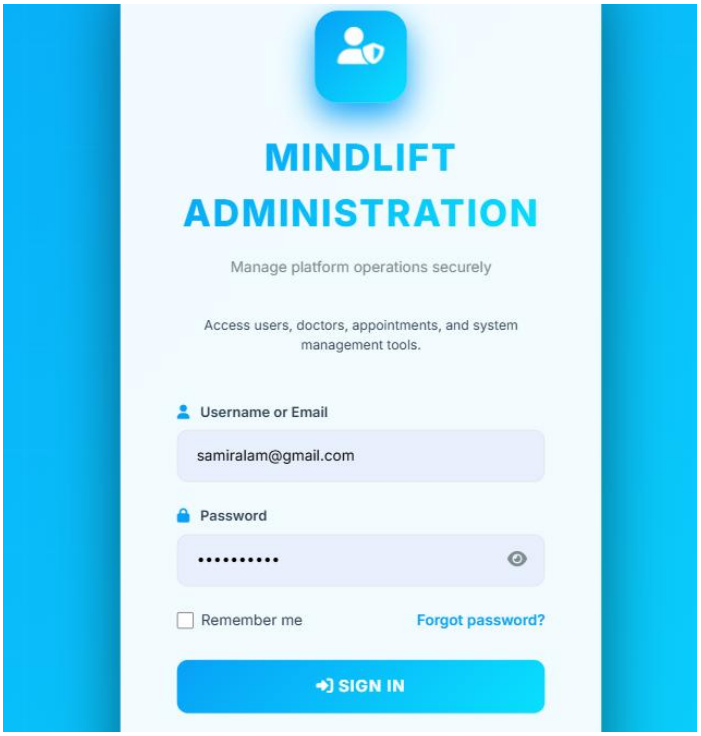


Figure 6.5: Admin Login

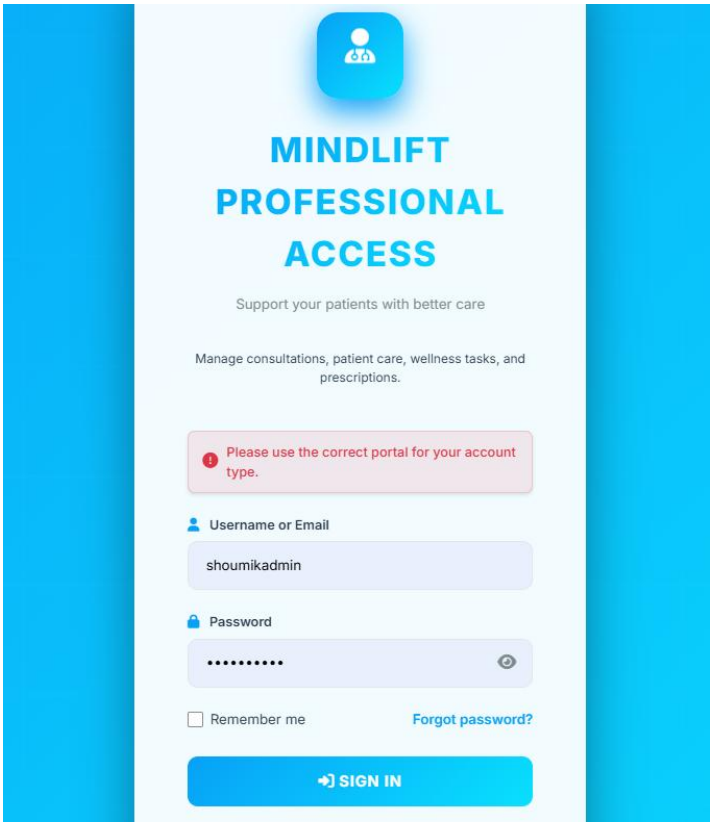


Figure 6.6: Role-Based Login Access Restriction

Role-Based Login Access Restriction: The system restricts users to their designated login portals, preventing Patients, Doctors, and Administrators from accessing the platform through an incorrect role-based login interface.

6.2 Patient Assessment and Personalized Care Entry

6.2.1 Patient Dashboard

The Patient Dashboard provides quick access to essential care activities, including wellness tasks, mental wellness assessment, doctor discovery, appointment management, and mental health blogs

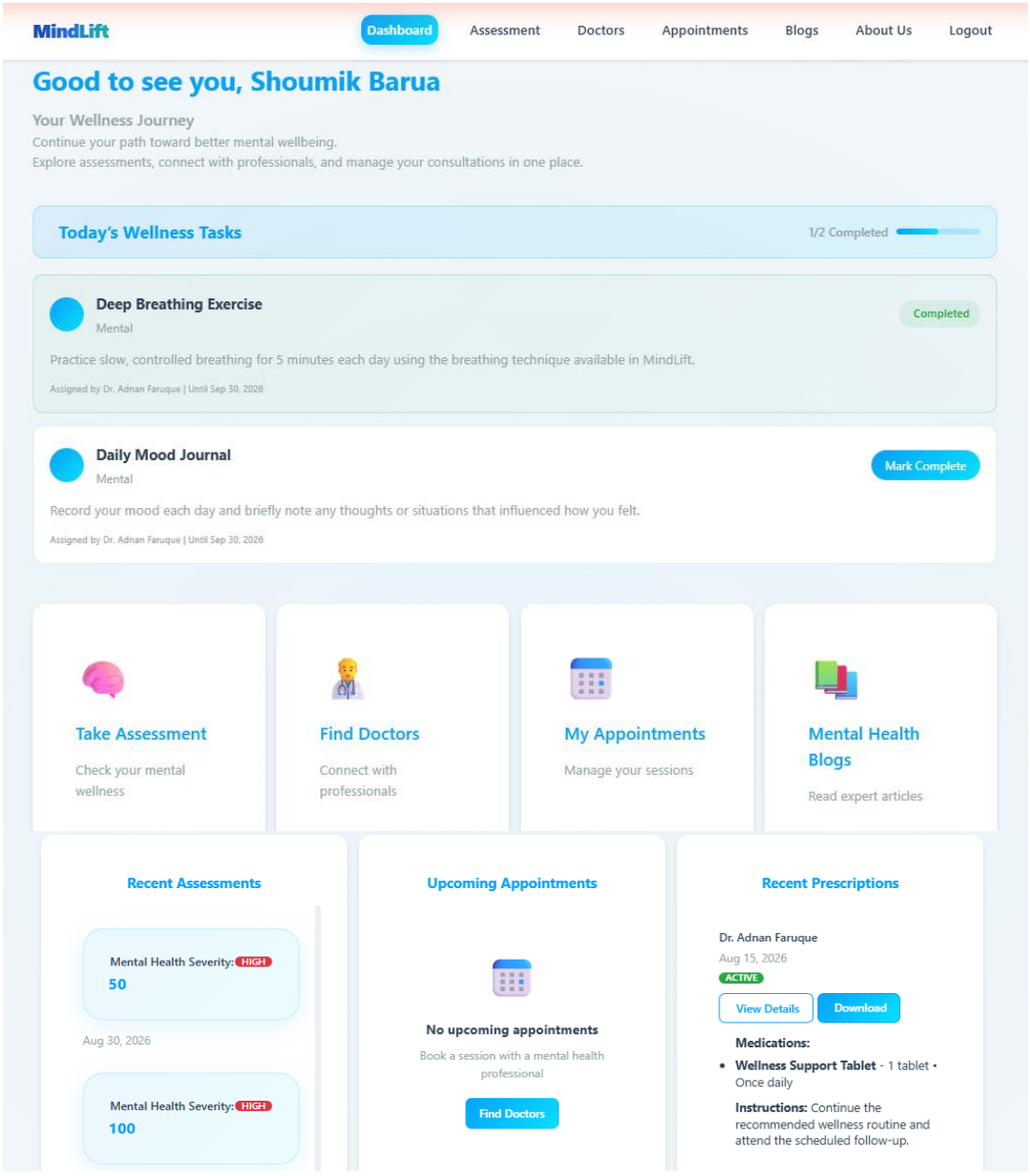
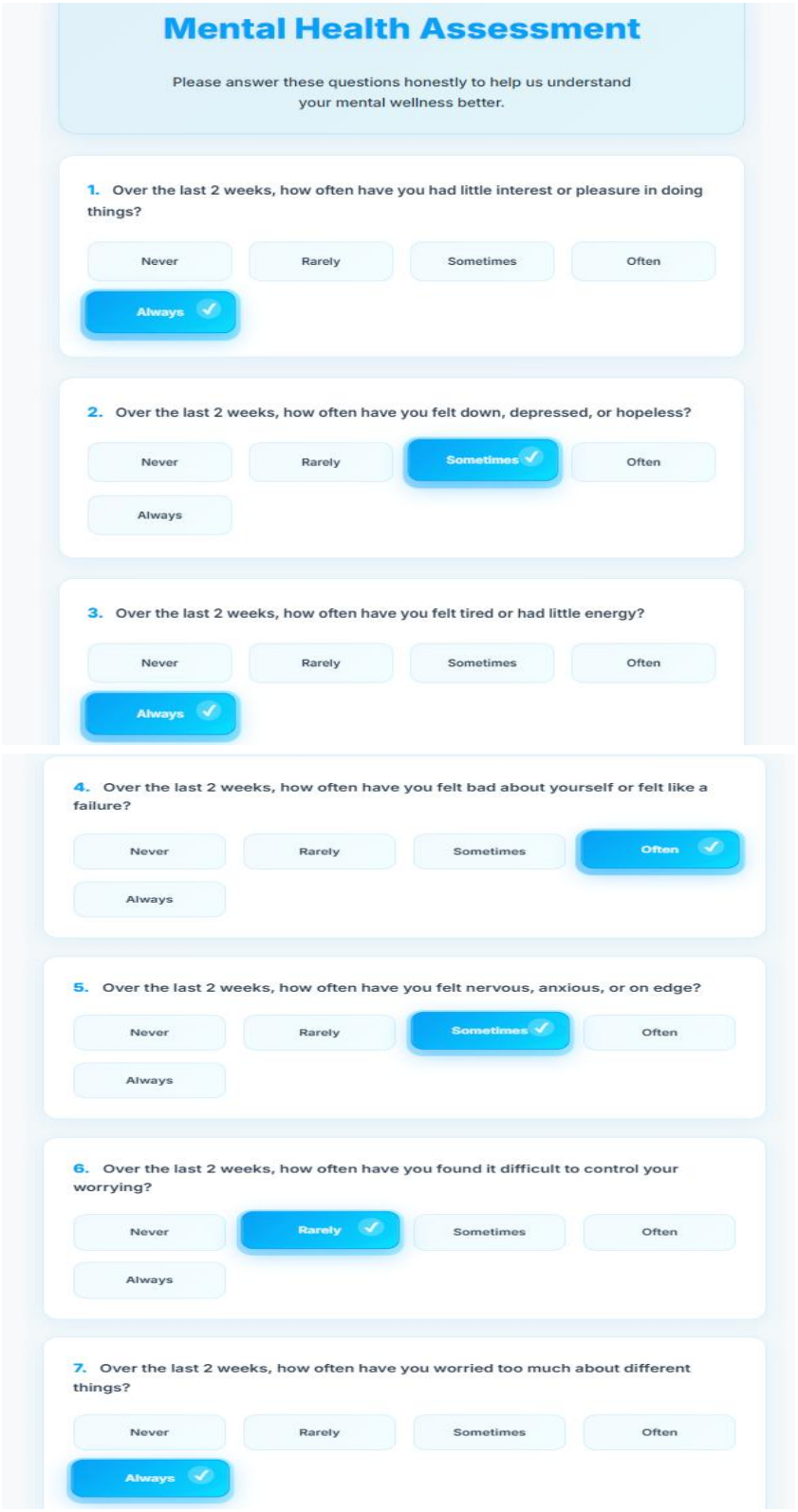


Figure 6.7: Patient Dashboard

6.2.2 Mental Wellness Assessment



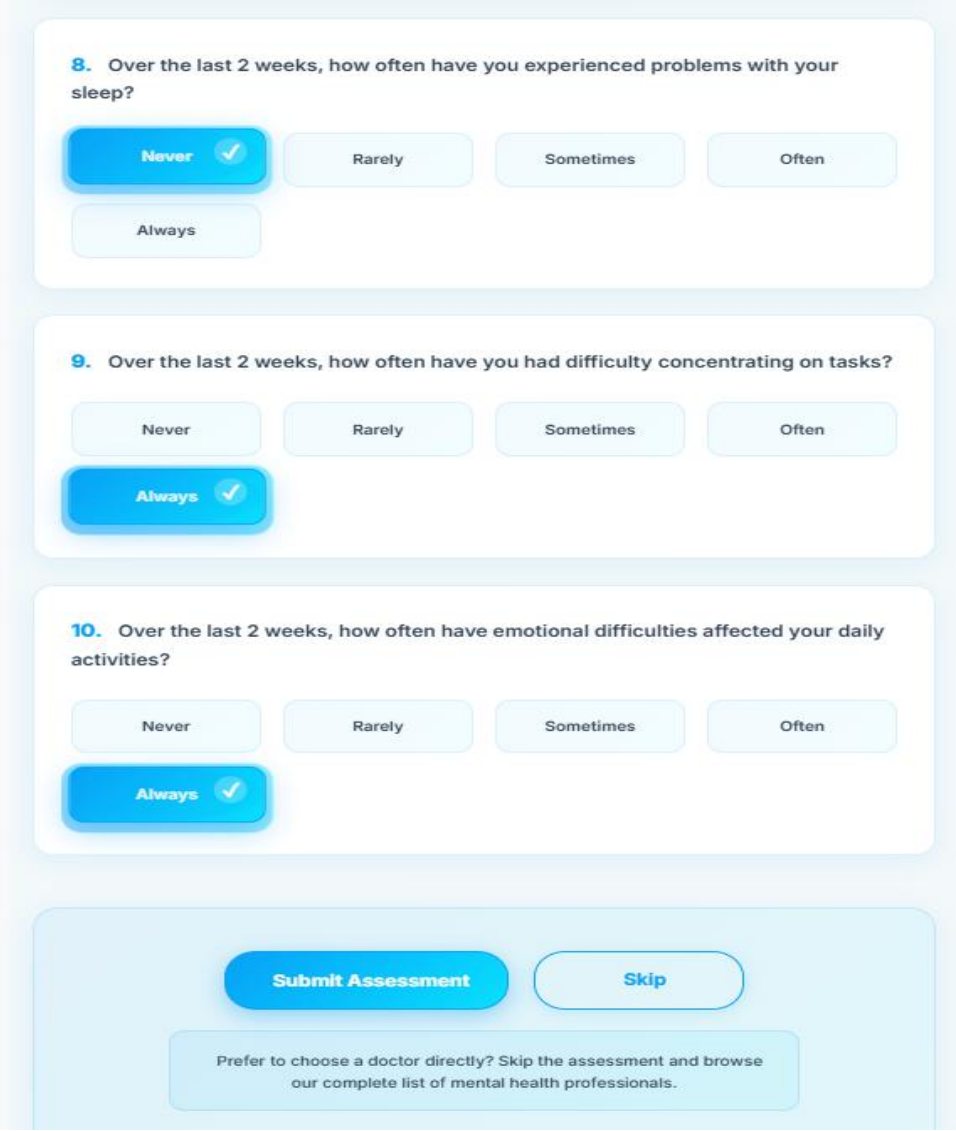


Figure 6.8: Mental Wellness Assessment

The Mental Wellness Assessment is the first step in MindLift's care journey. It uses 10 structured questions to evaluate the patient's current condition and generate an assessment result. The result then supports personalized doctor and mental wellness resource recommendations.

6.2.3 Assessment Results, Doctor and Blog Recommendations

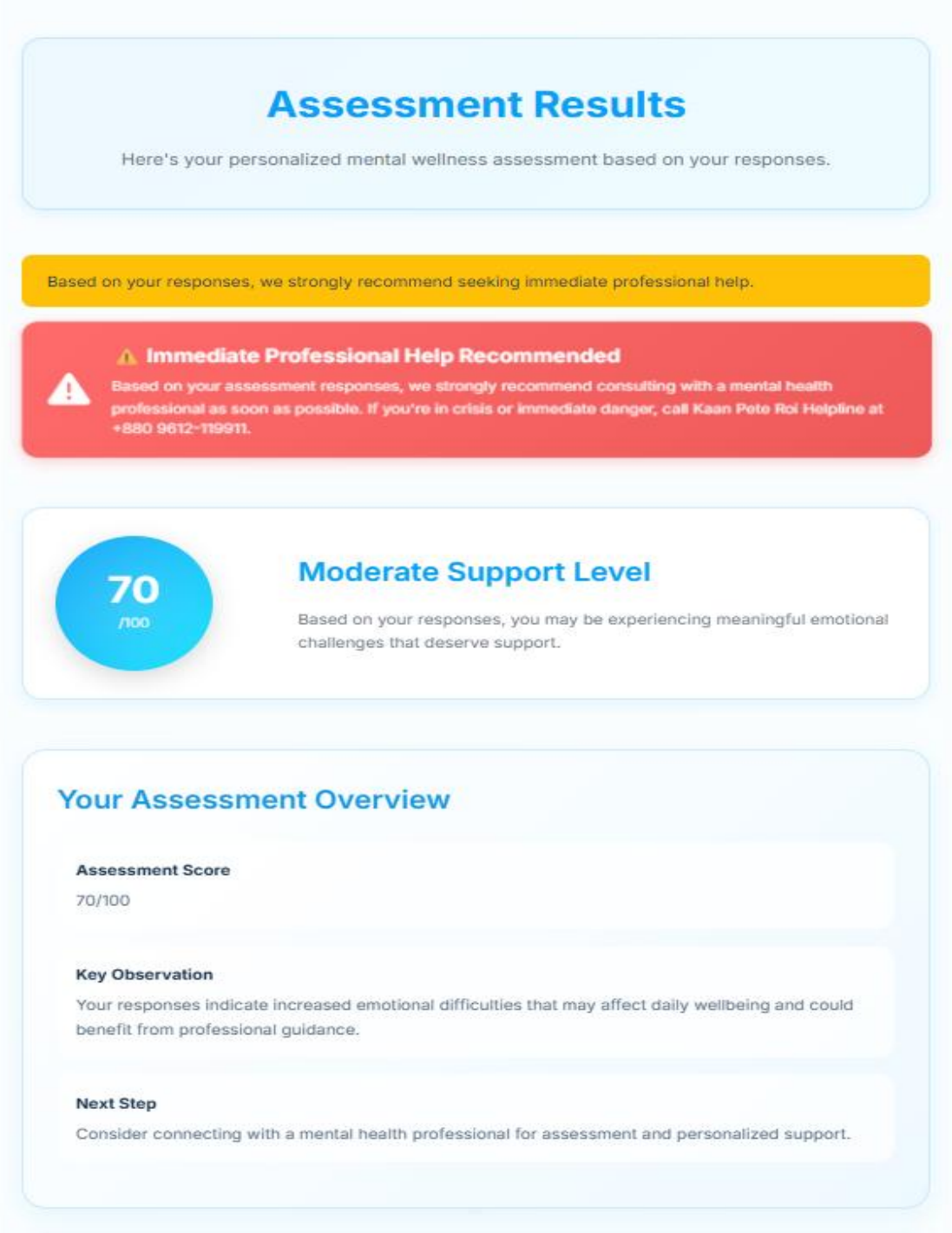


Figure 6.9: Assessment Results

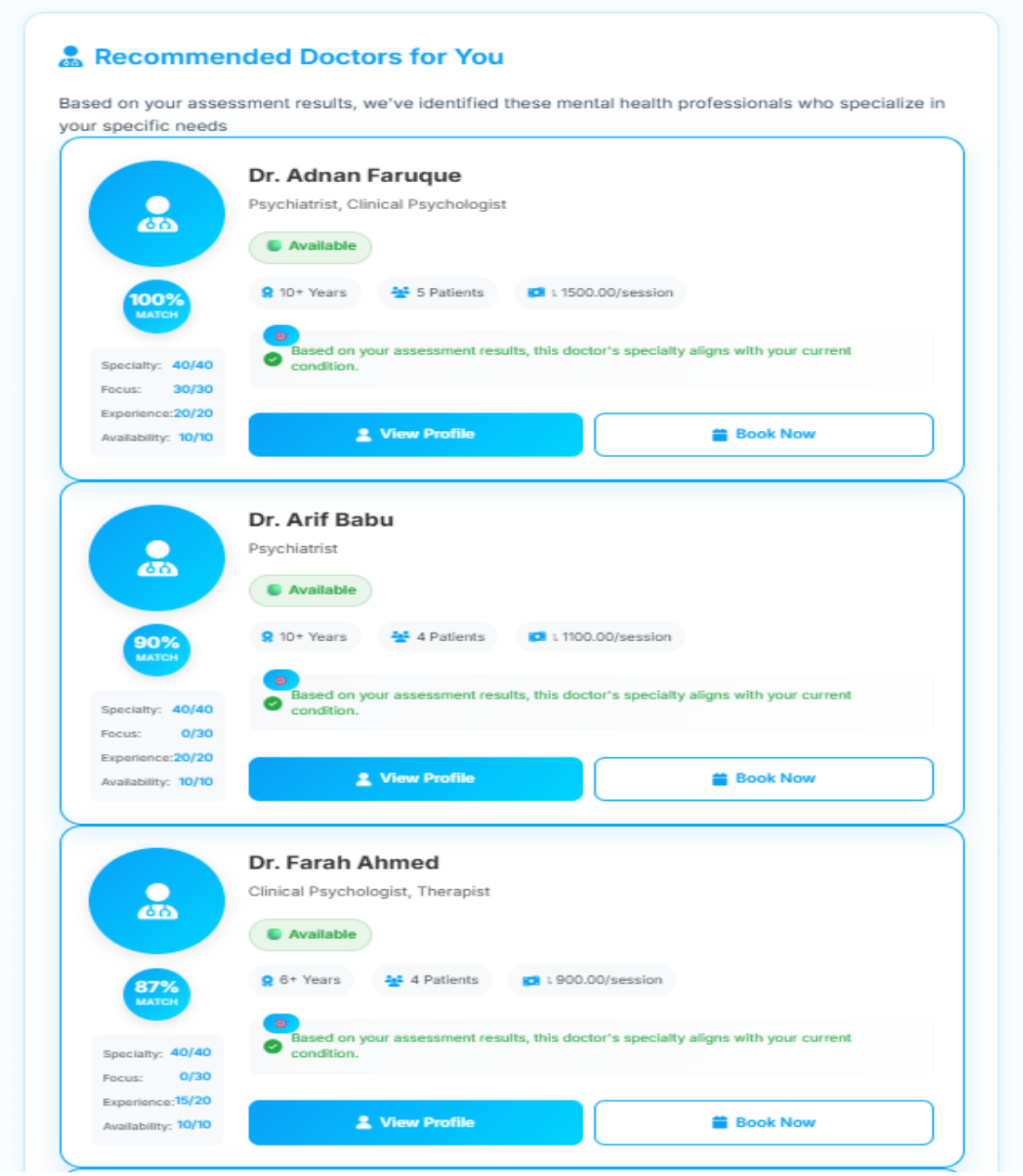


Figure 6.10: Doctor Recommendations

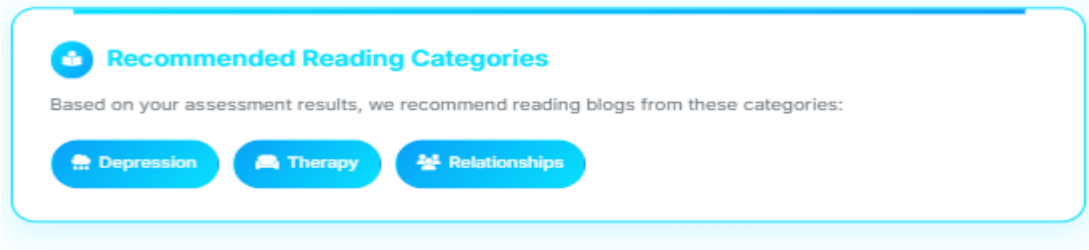


Figure 6.11: Blog Recommendations

6.2.4 Doctor Discovery and Specialty Filtering

The Doctor Discovery interface allows patients to browse available professionals and quickly narrow results using search and specialty filters to find suitable doctors.

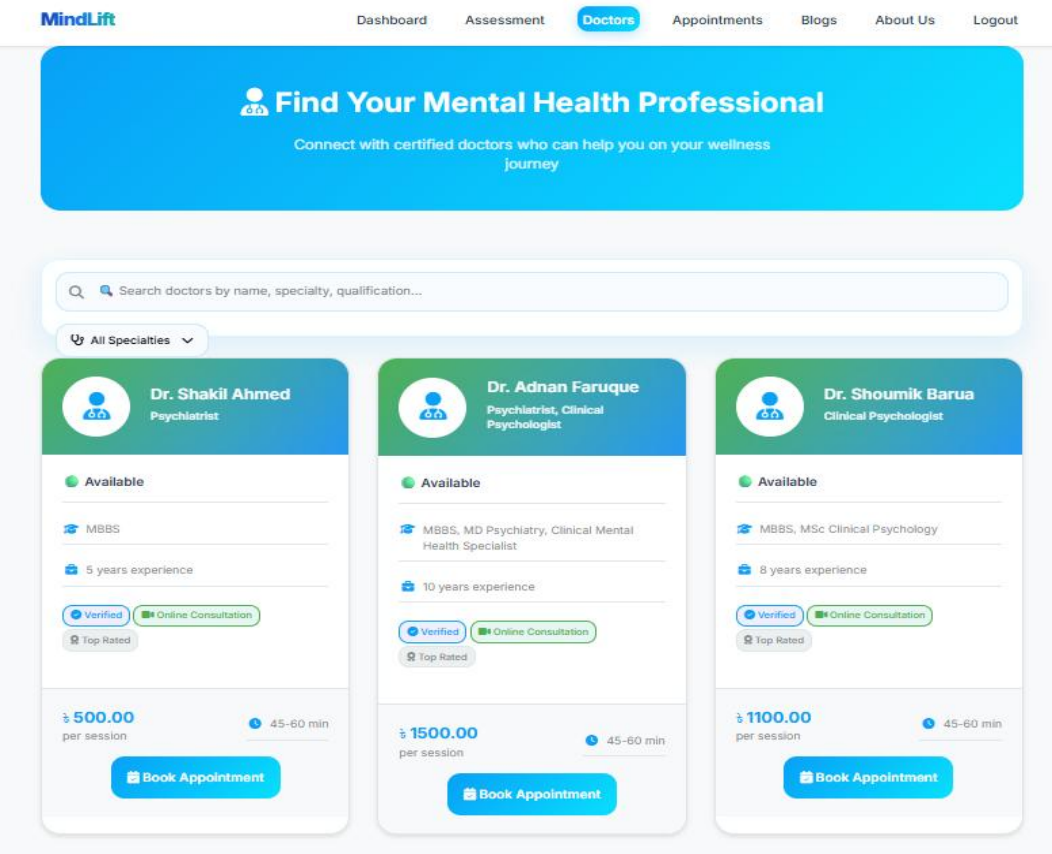


Figure 6.12: Doctor Discovery

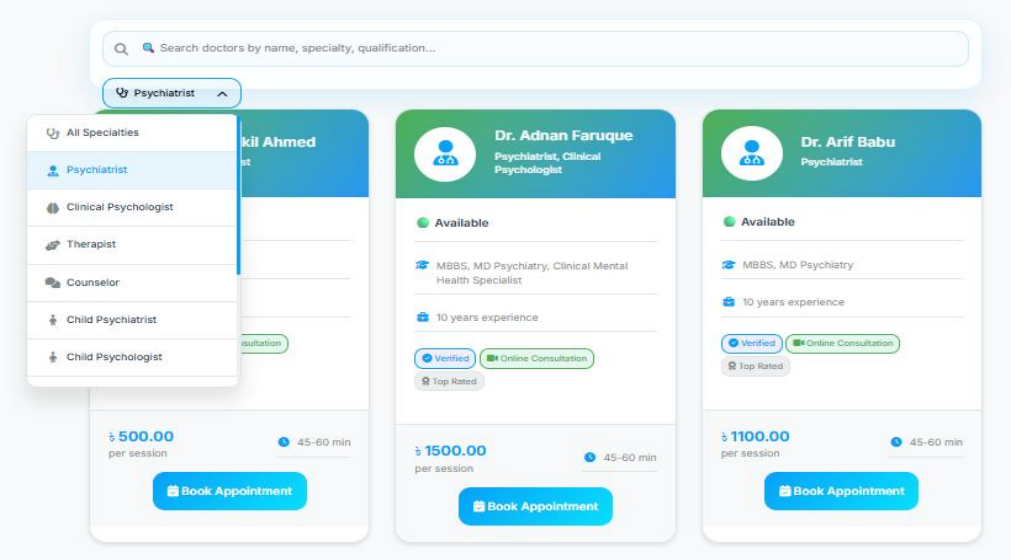


Figure 6.13: Specialty Filtering

6.3 Appointment, Payment, and Consultation Workflow

6.3.1 Appointment Scheduling

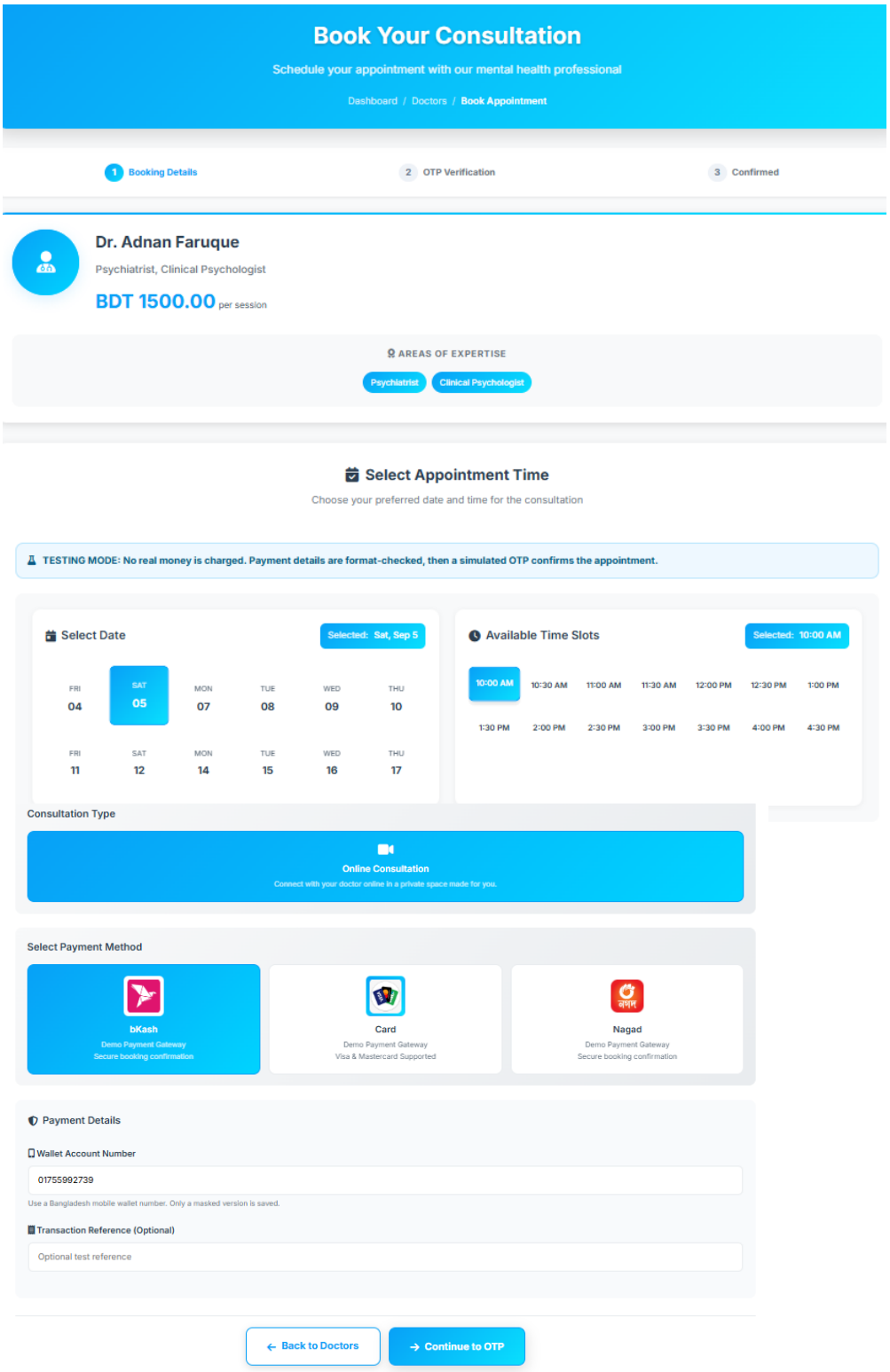


Figure 6.14: Appointment Scheduling

6.3.2 Email OTP Payment Verification

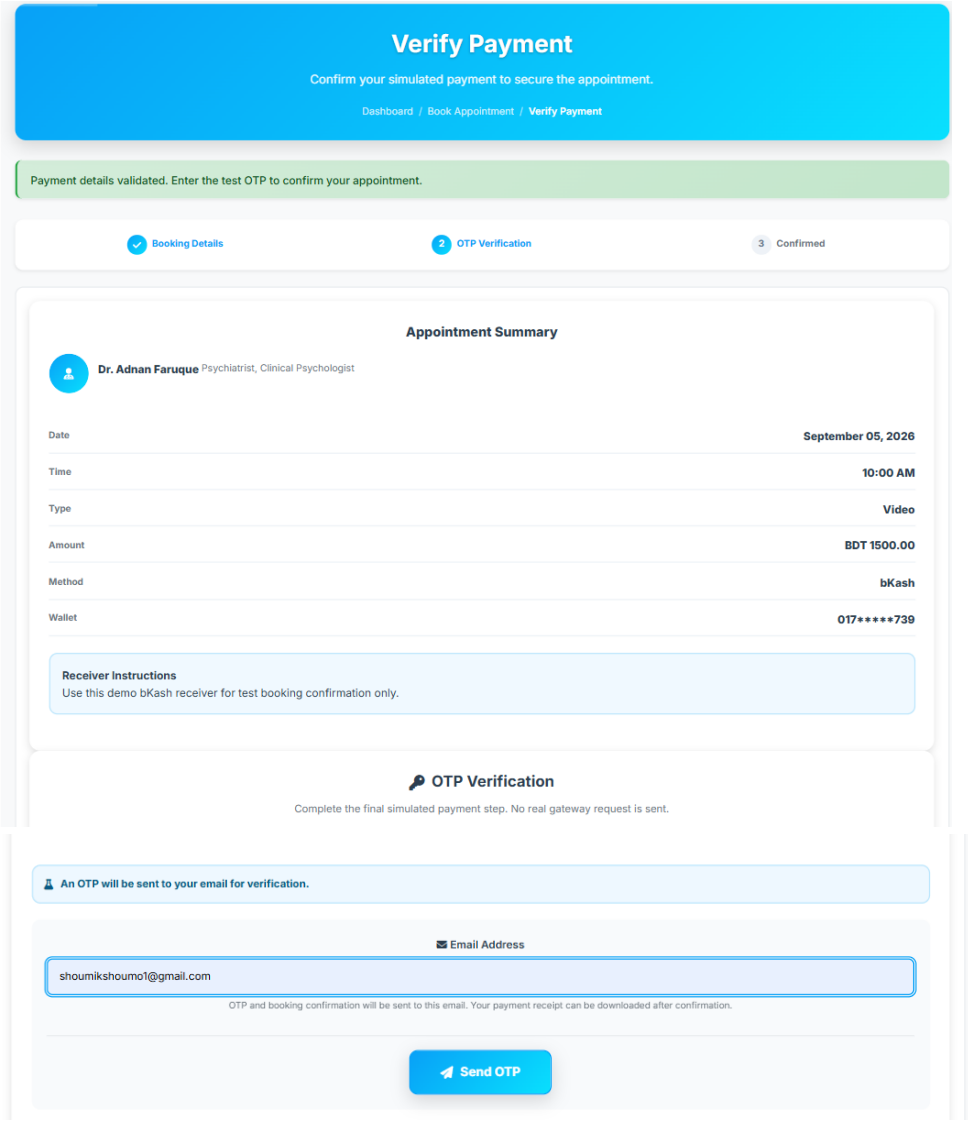


Figure 6.15: Email OTP Payment Verification

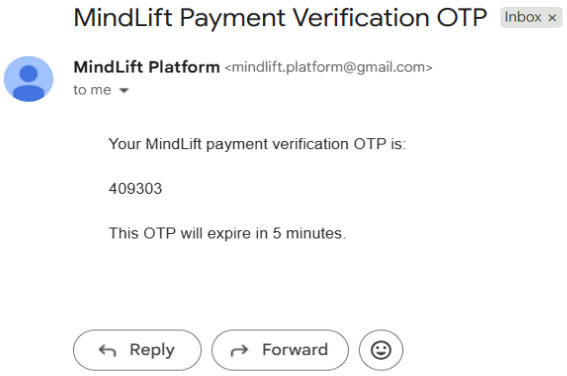


Figure 6.16: Email OTP Delivery for Payment Verification

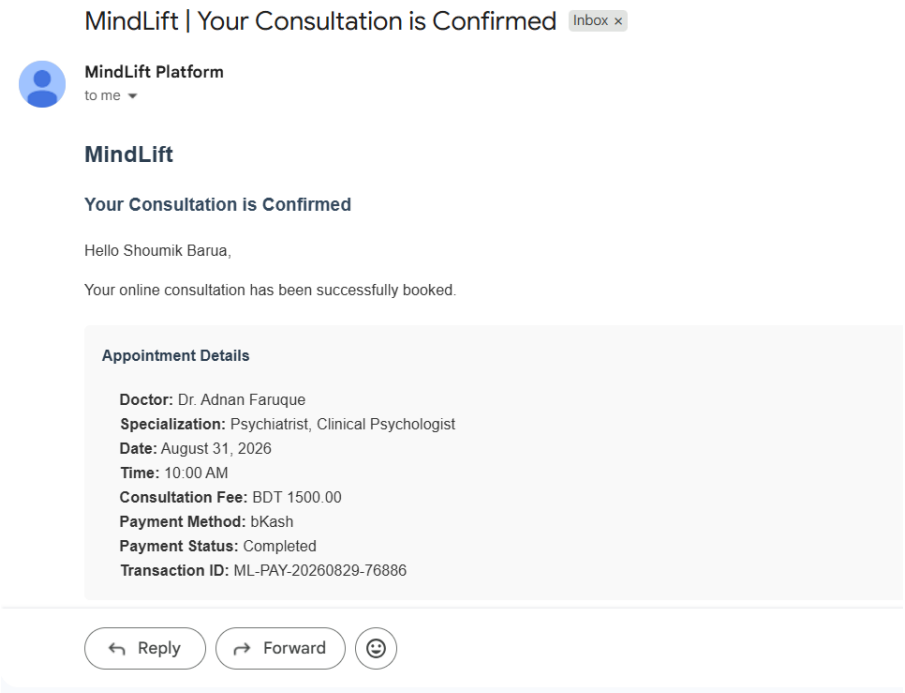


Figure 6.17: Appointment Confirmation Email

Figure 6.15–6.17 shows how MindLift employs email communication for the appointment confirmation workflow. Once the patient has finished the necessary booking and mock payment information, the system checks the submitted details and asks for an email address for verification. When the patient clicks on Send OTP, the Django backend generates the one-time password and sends it to the given mail address through the configured Gmail SMTP service. The OTP can be valid for a short amount of time to allow the system to verify whether the patient has access to the email address given before confirming the appointment.

After the correct OTP is submitted, MindLift verifies the code, completes the payment-verification stage, and confirms the appointment. The system then automatically sends a separate e-mail to the same verified address confirming the appointment. The email contains the important consultation details such as doctor, specialisation, date and time of appointment, consultation fee, payment method, payment status and transaction ID. The email process thus serves a dual purpose,

- To verify the booking via a time-limited OTP
- To provide the patient with a permanent confirmation record of the successfully scheduled consultation.

6.3.3 Appointment Confirmation

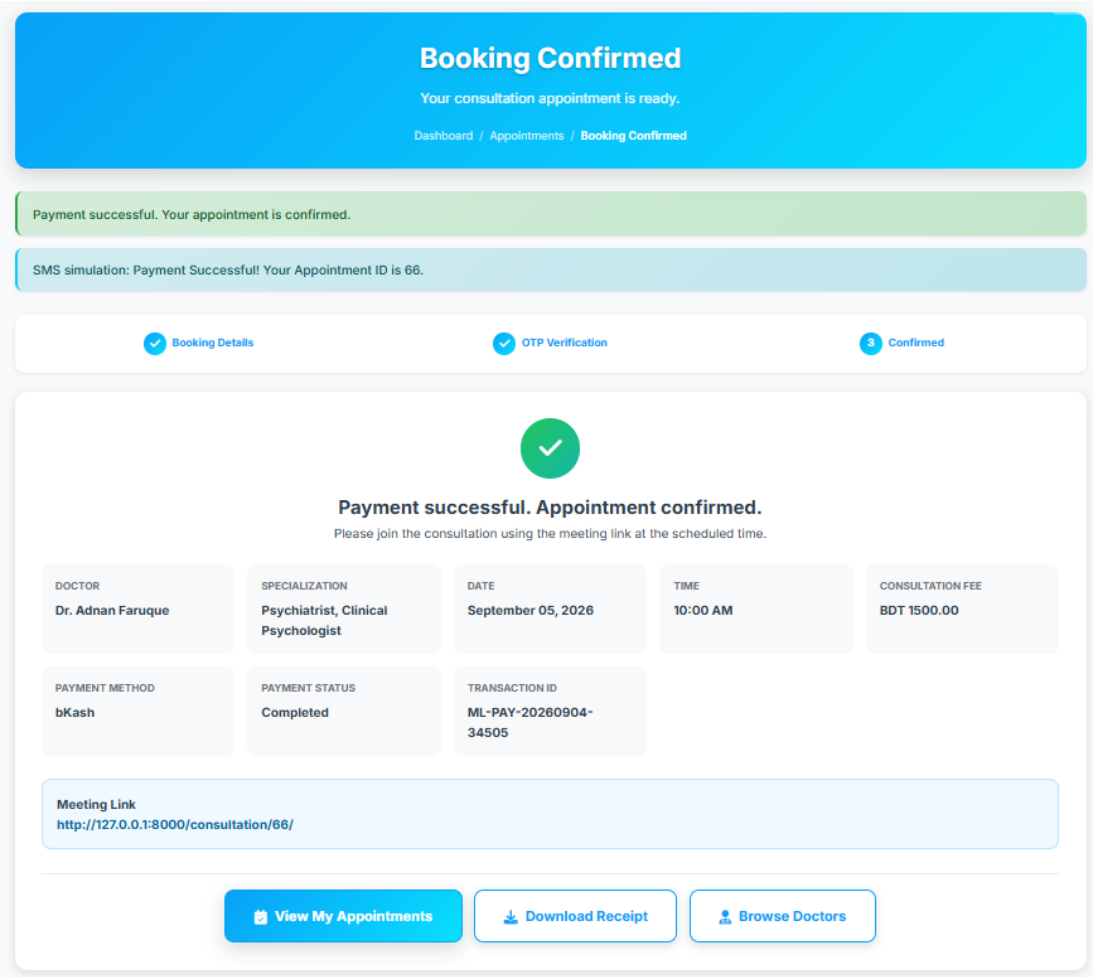


Figure 6.18: Appointment Confirmation

After successful OTP confirmation and payment, MindLift will send the final appointment confirmation with complete details of the consultation. The page shows the selected doctor, specialization, date and time of the consultation, consultation fee, payment method, payment status, transaction ID and meeting link. It also confirms the successful completion of the booking process and gives direct options to view upcoming appointment, download the appointment receipt or continue browsing doctors.

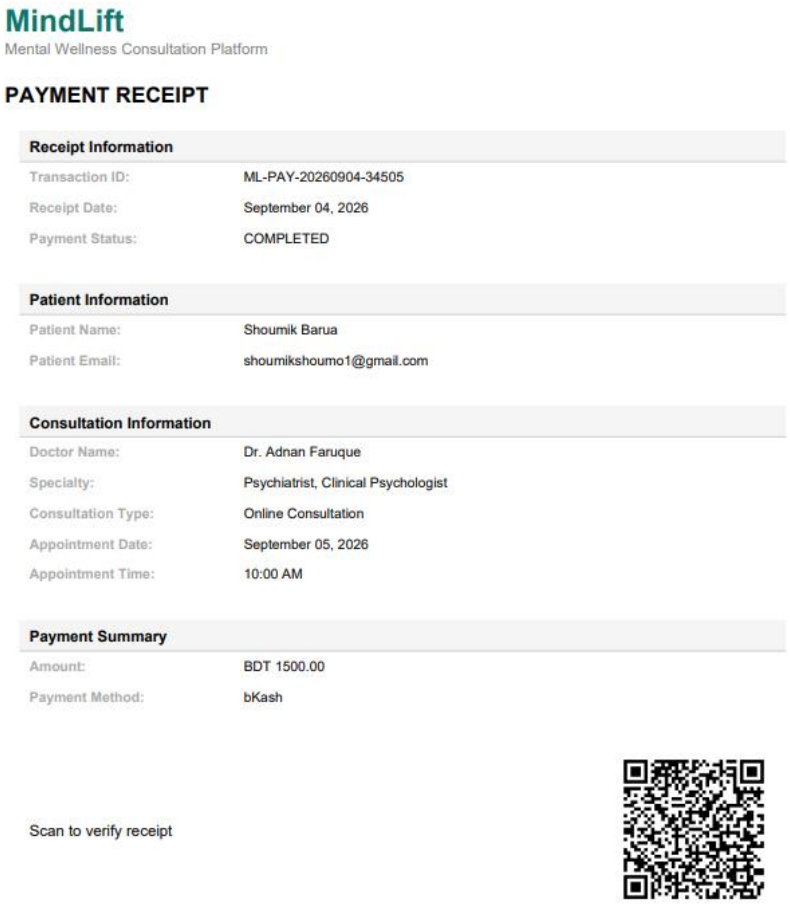


Figure 6.19: QR-Enabled Payment Receipt

6.3.4 Patient Appointment Management

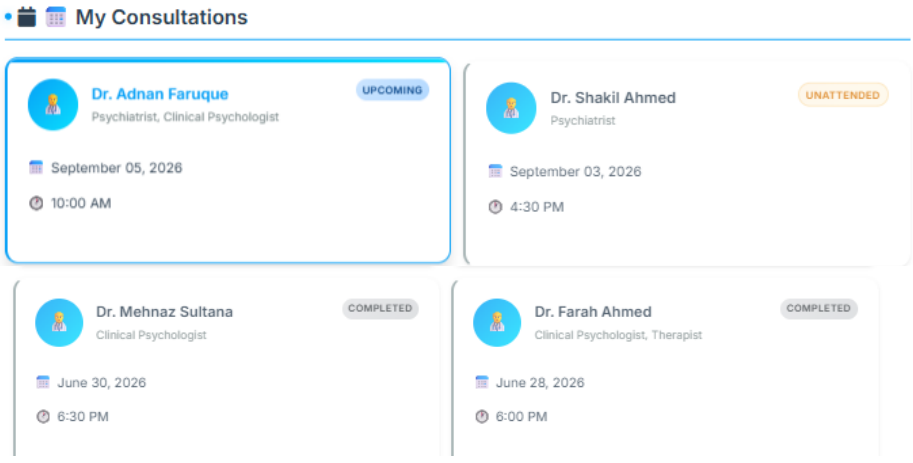


Figure 6.20: Patient Appointment Management

6.3.5 Online Video Consultation

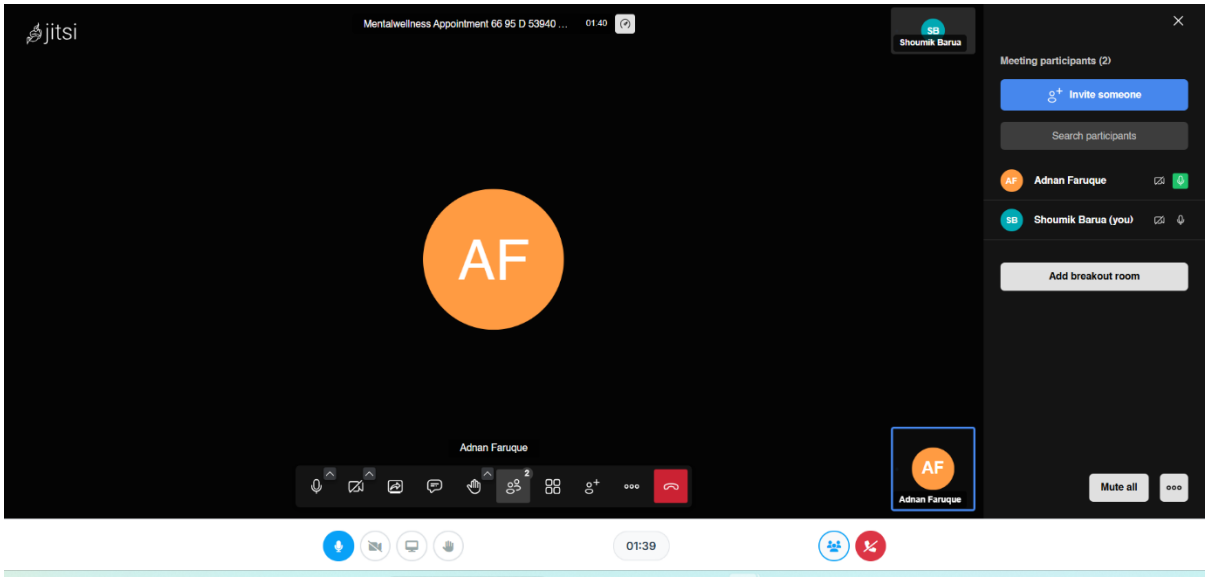


Figure 6.21: Online Video Consultation

6.4 AI-Assisted Mental Wellness and Resource Support

6.4.1 Mental Wellness Blogs

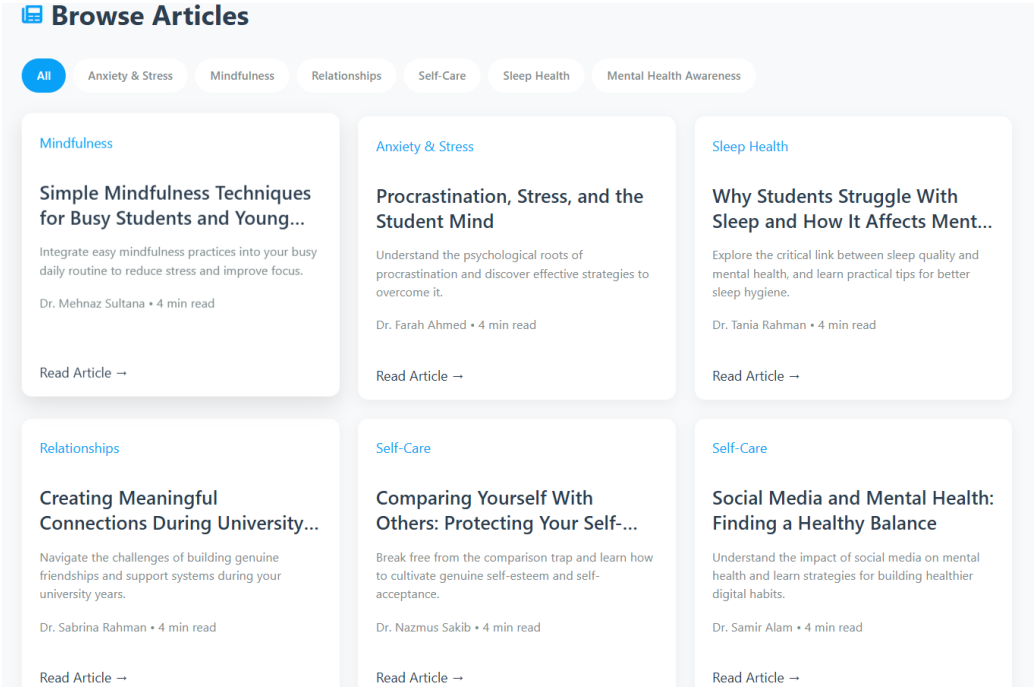


Figure 6.22: Mental Wellness Blogs

6.4.2 Semantic Blog Search

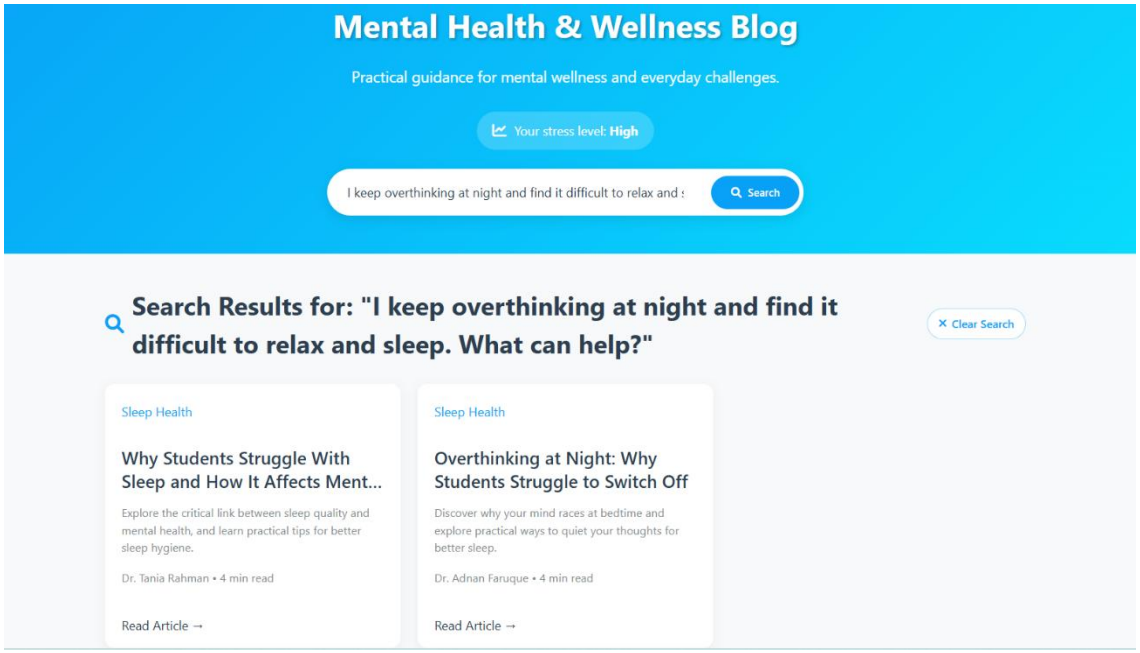


Figure 6.23: Semantic Blog Search

The Semantic Blog Search allows users to state their concerns in natural language and receive relevant mental wellness articles based on the meaning and context of the query. It doesn't just match exact keyword matches, it helps users to find more relevant content based on how they feel.

6.4.3 AI Quick Summary

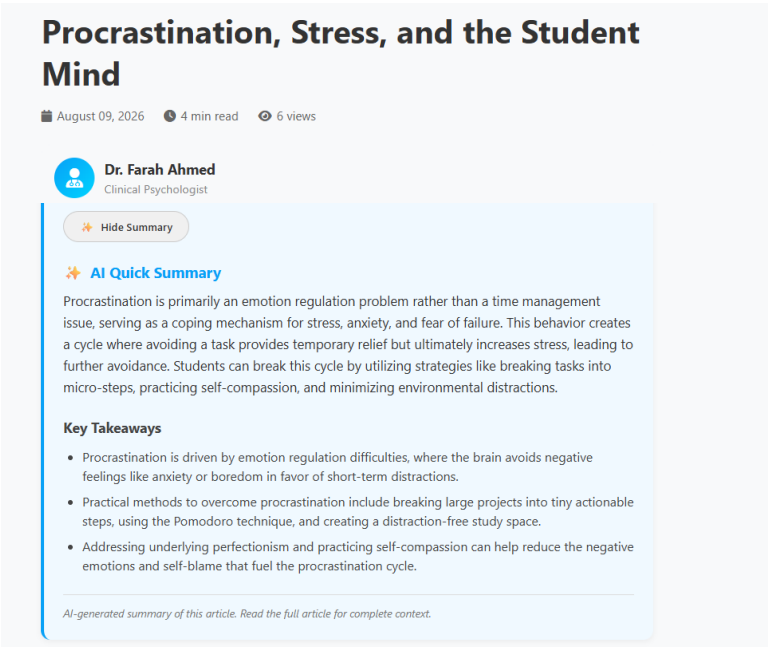


Figure 6.24: AI Quick Summary

6.4.4 AI Support Chatbot

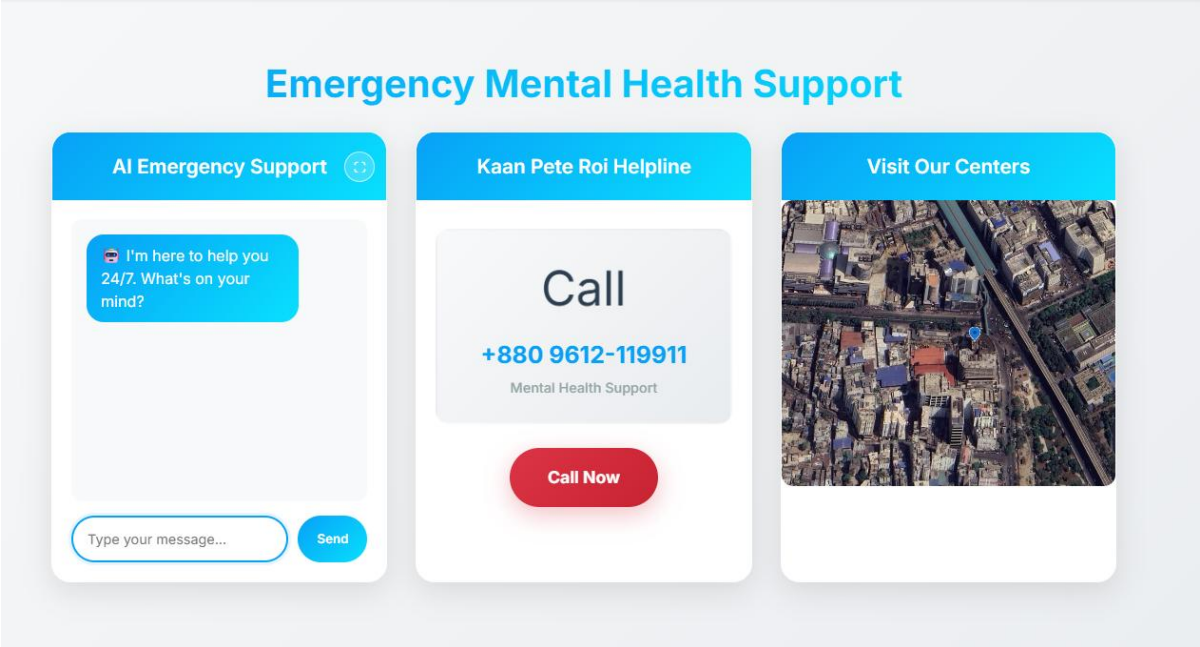


Figure 6.25: AI Support Chatbot

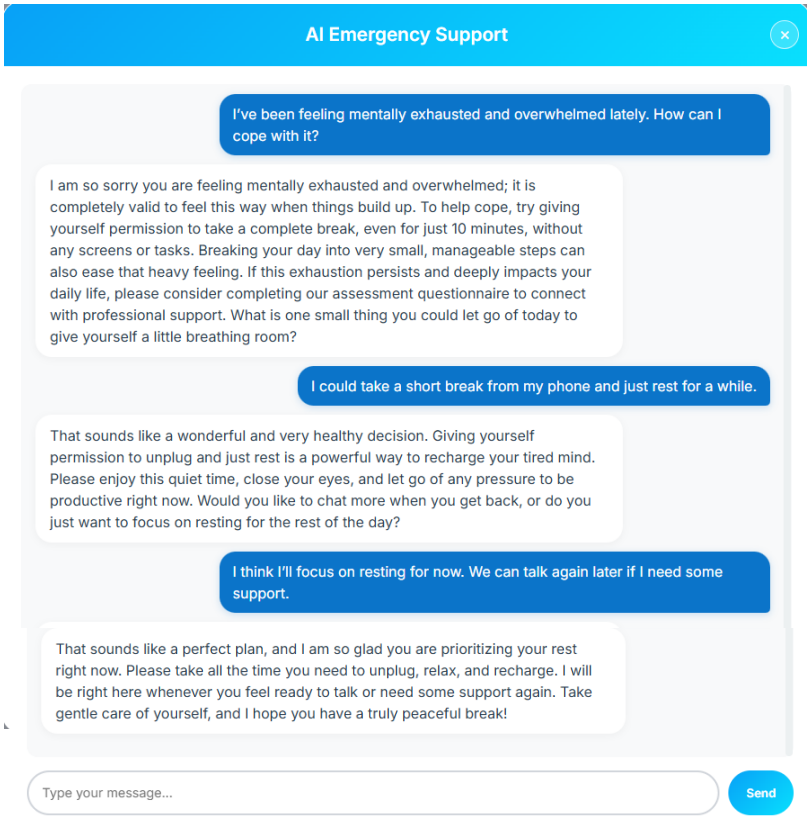


Figure 6.26: Context-Aware AI Mental Wellness Support Conversation

The AI Support Chatbot maintains the context of a conversation across multiple exchanges, ensuring that each reply is relevant to what the user has already told us. It also has Retrieval-Augmented Generation (RAG) that allows the system to access relevant curated mental wellness and platform-specific knowledge before generating a response. As shown in **Figure 6.26**, the chatbot identifies the user's mental fatigue, offers sympathetic and pragmatic advice, affirms the user's preferred coping strategy, and maintains the tone of the conversation and the awareness of context. This shows how MindLift uses the conversation history, retrieved knowledge and AI-generated responses to provide more relevant and ongoing mental wellness support.

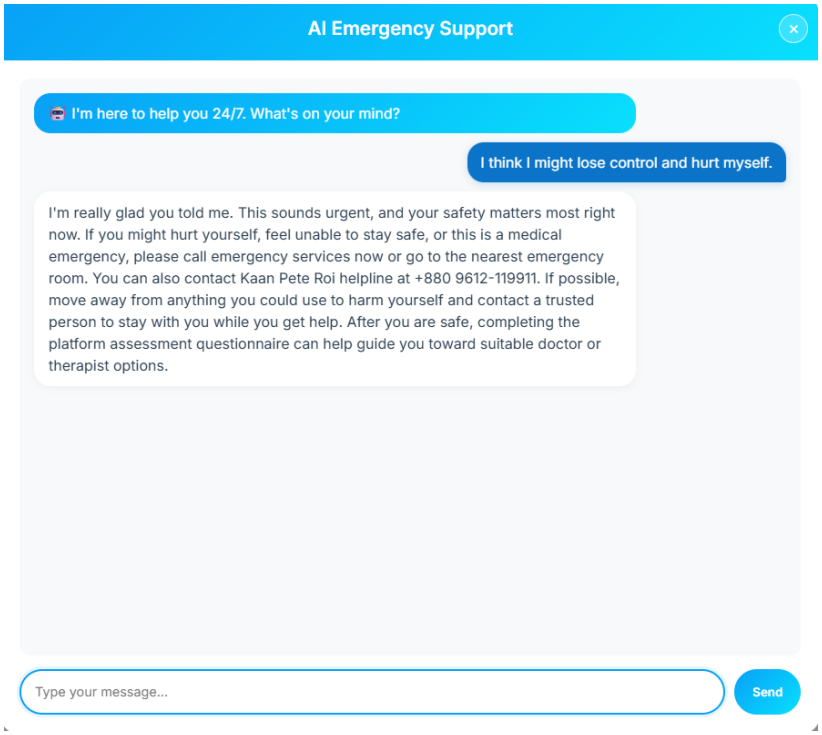


Figure 6.27: Safety-Oriented Emergency Response

As shown in **Figure 6.27**, MindLift identifies high-risk messages and activates a dedicated safety-oriented response path instead of treating them as ordinary chatbot conversations. In this example, the system recognizes a potential self-harm situation and immediately prioritizes the user's safety by recommending urgent real-world assistance, emergency services, nearby support, removal from possible means of harm, and contact with a trusted person. It also presents crisis-support information and guides the user toward professional care after immediate safety has been addressed.

6.4.5 Breathing Exercise

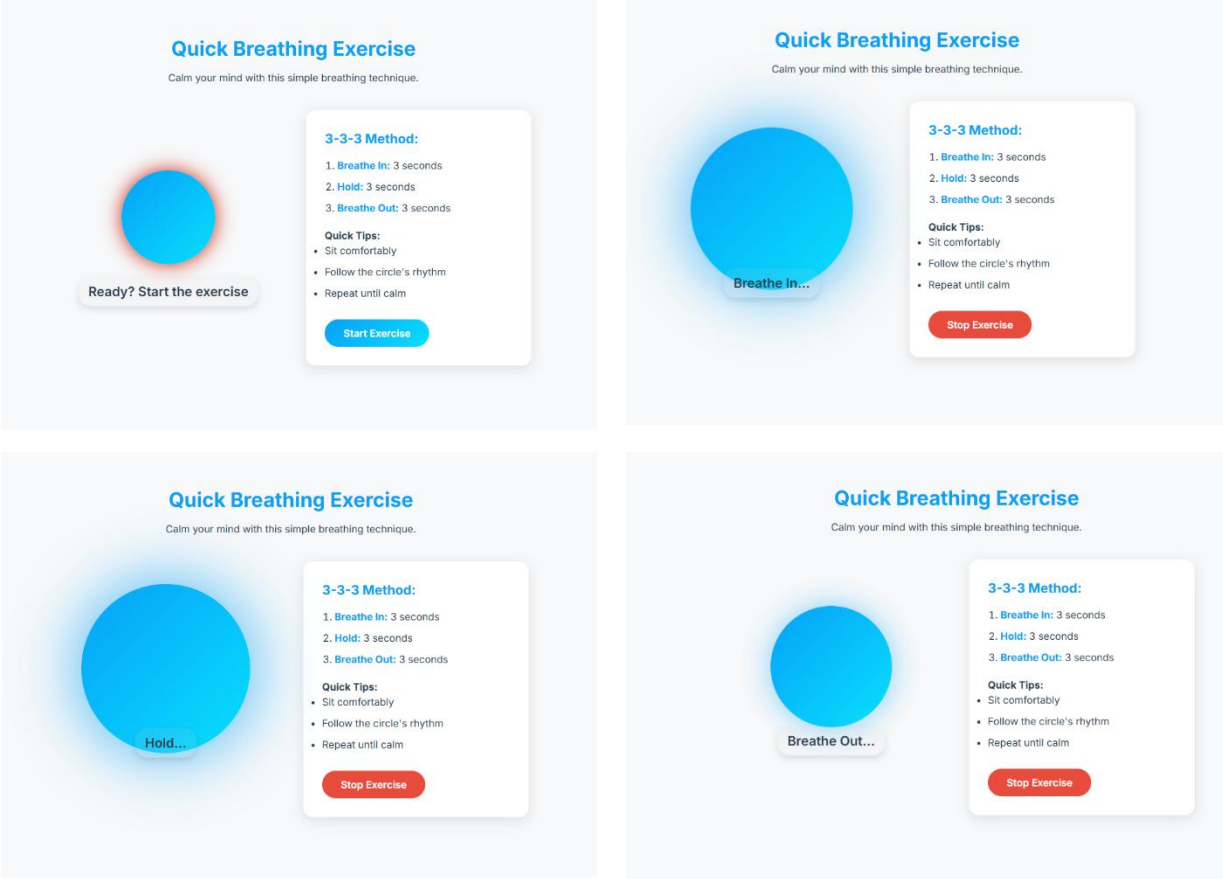


Figure 6.28: Breathing Exercise

The Breathing Exercise is a pulsating scale animation The circular visual slowly expands, pauses and contracts to show the inhale, hold and exhale phases of the 3-3-3 breathing cycle. The size is animated which helps with the timing visually . This makes it easier for the user to follow the breathing rhythm .

6.5 Doctor Professional and Care Management

6.5.1 Doctor Dashboard

The Doctor Dashboard is the main page for professional information and important activities of the doctor in MindLift. It shows specialization, experience and consultation fee. Quick Actions section gives direct access to patient management, appointments, schedule management, profile editing, blog creation and management and performance analytics. Doctors can handle their consultation duties and their professional platform activities in one place.

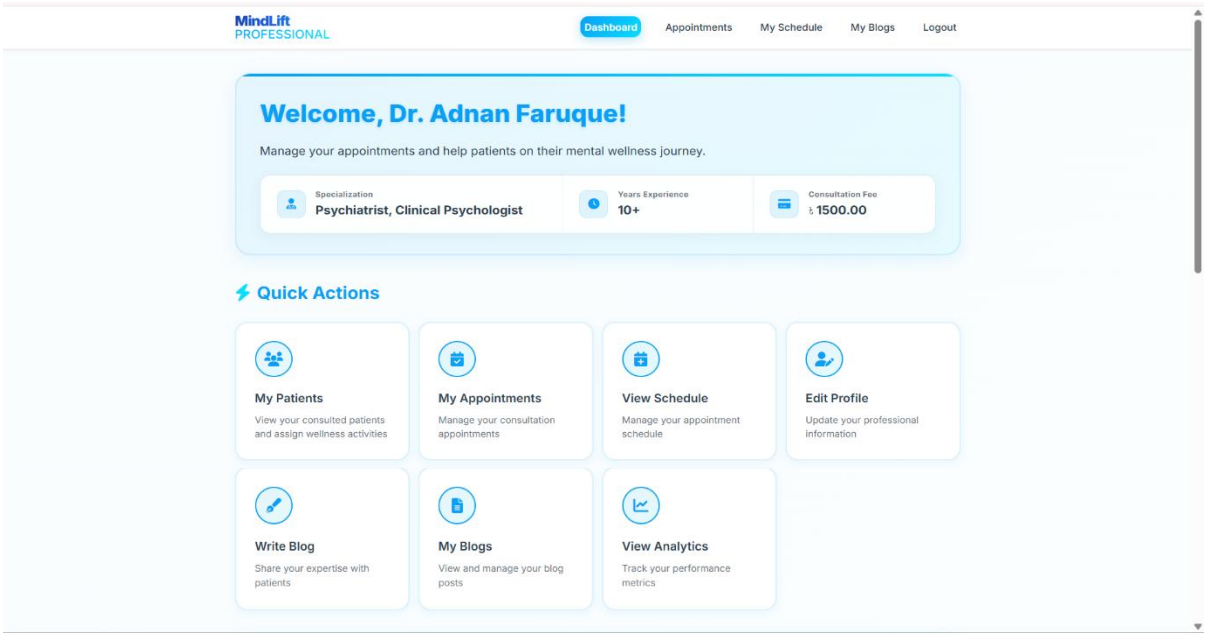


Figure 6.29: Doctor Dashboard

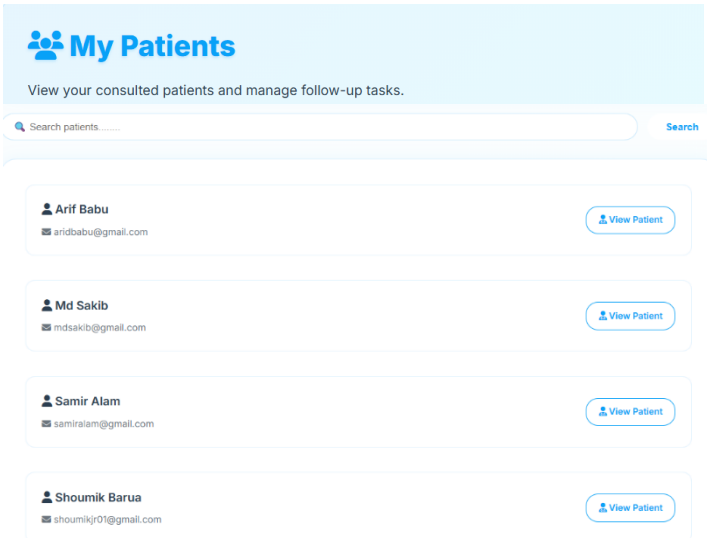


Figure 6.30: Patient Management

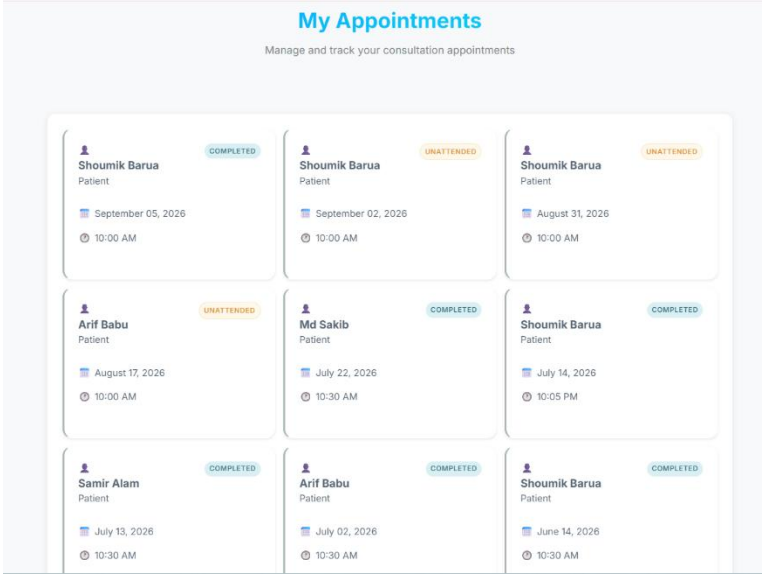


Figure 6.31: Doctor Appointment Management

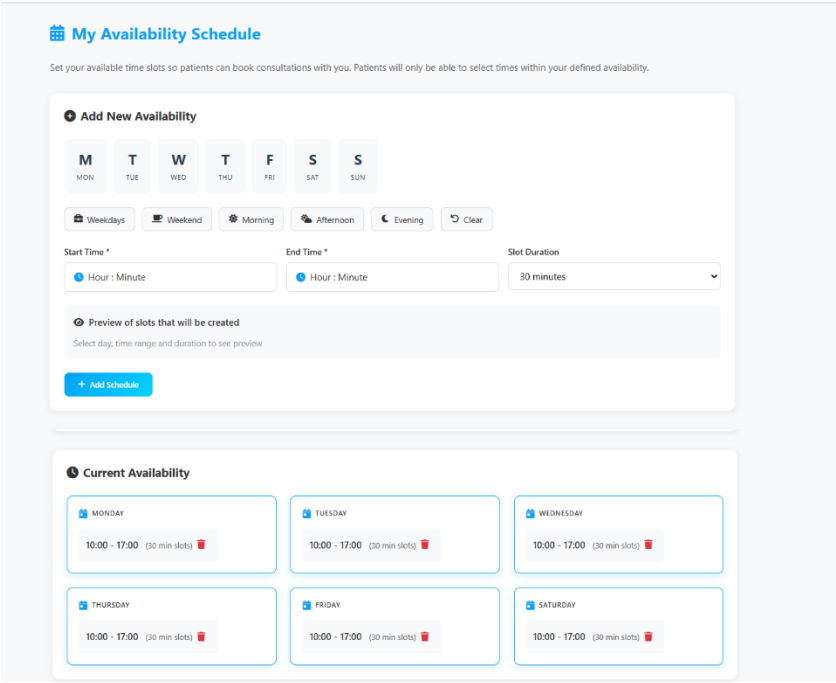


Figure 6.32: Availability and Schedule Management

Figure 6.32 Availability and Schedule Management interface Doctors can define the days of consultation, start and end time, and the slot duration. Doctors can add new availability and view their current schedules ensuring that patients only see and book consultation times which fall within the doctor's configured availability.

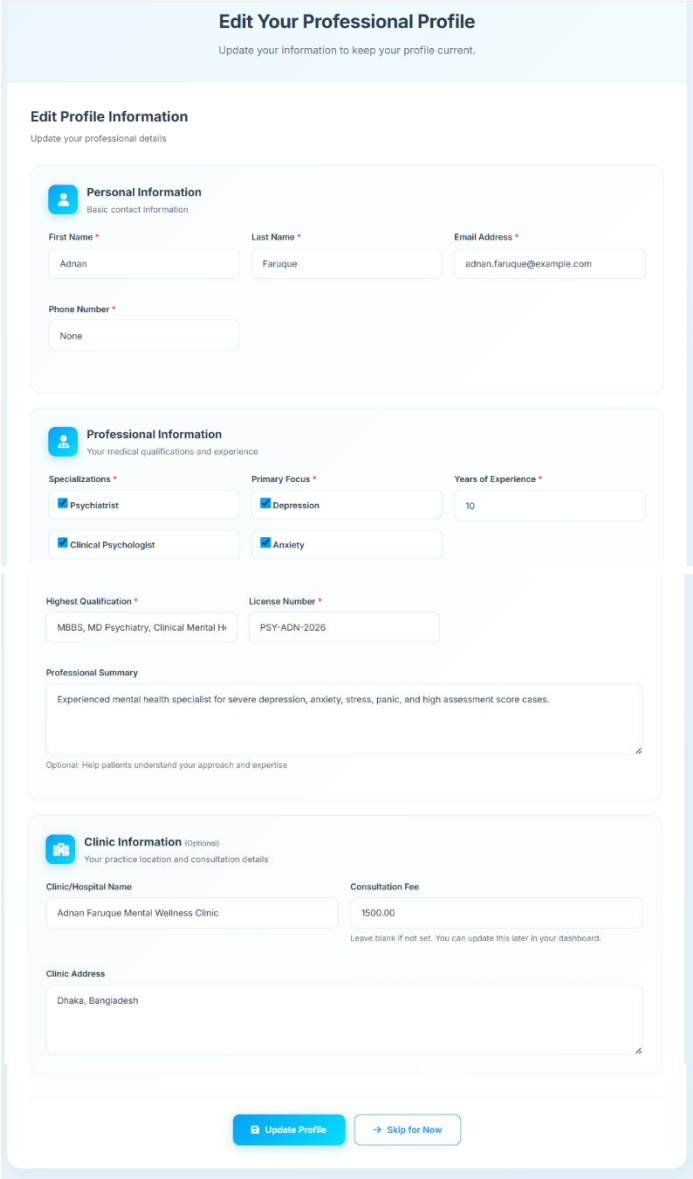


Figure 6.33: Professional Profile Management

The Professional Profile interface allows doctors to update their personal and professional information. This includes specialization, primary focus, experience, qualifications, license number, professional summary, consultation fee and clinic details. This also helps to keep the doctor's public profile accurate and current.

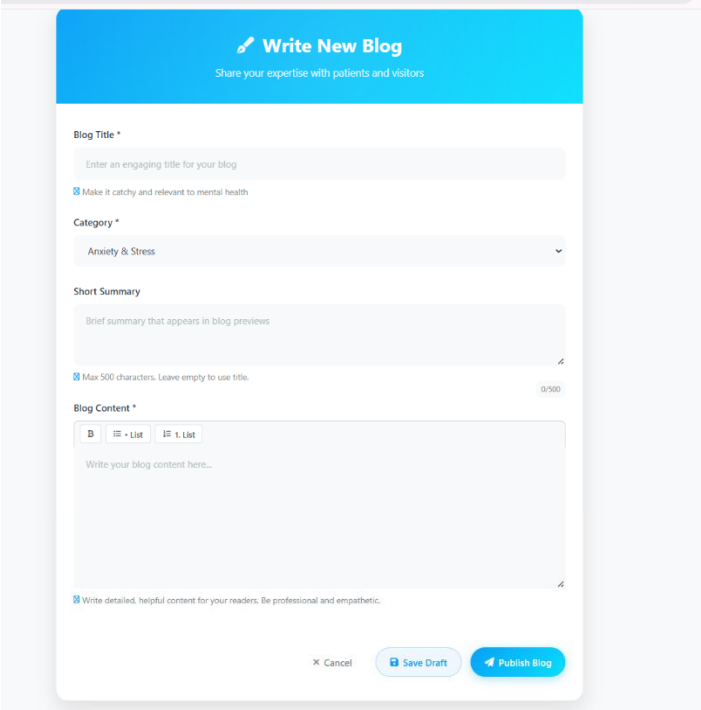


Figure 6.34: Doctor Blog Creation Interface

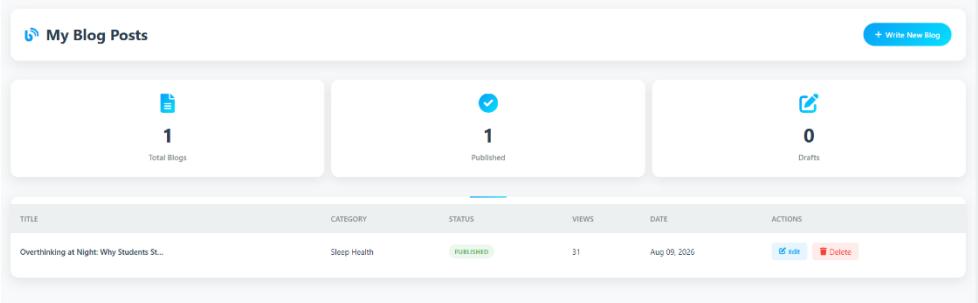


Figure 6.35: Doctor Blog Management

The Doctor Blog Management interface allows doctors to create, save, publish, edit, and manage their mental wellness articles. It also provides a summary of total posts, published blogs, drafts, views, and available management actions.

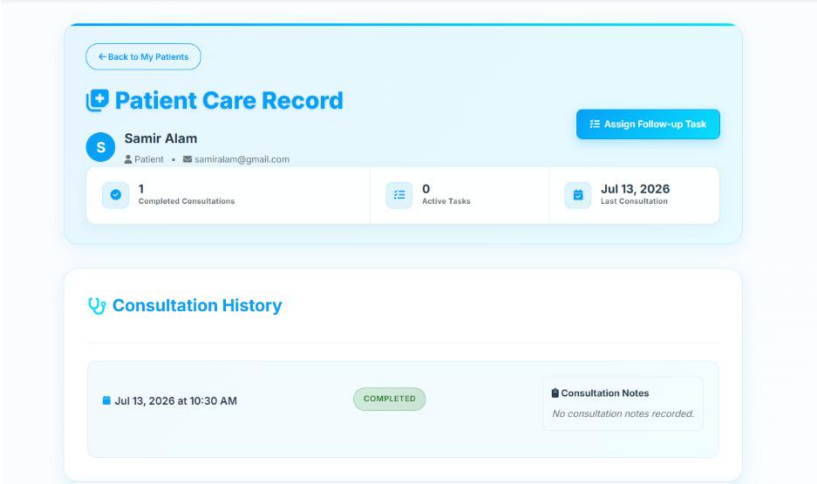


Figure 6.36: Patient Care Record and Consultation History

The Patient Care Record provides doctors with a consolidated view of a patient's consultation history, completed consultations, active wellness tasks, and recent care information. This helps the doctor review previous interactions before planning further follow-up.

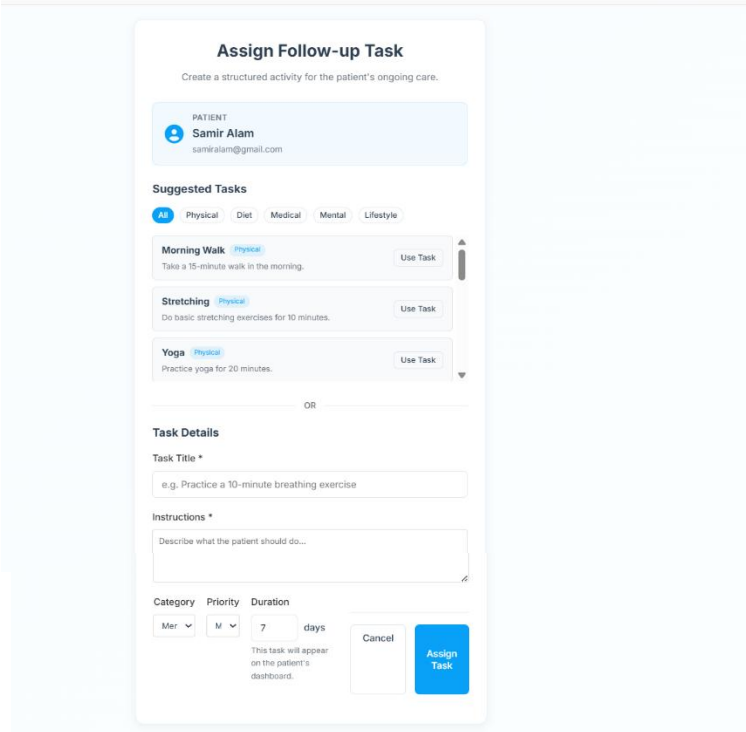


Figure 6.37: Follow-Up Wellness Task Assignment

The Follow-Up Task interface is designed to allow the doctor to assign structured wellness activities to the patient after the consultation. Doctors can select suggested tasks, or create a custom task with instructions, category, priority and duration, allowing for continued care beyond the consultation session.

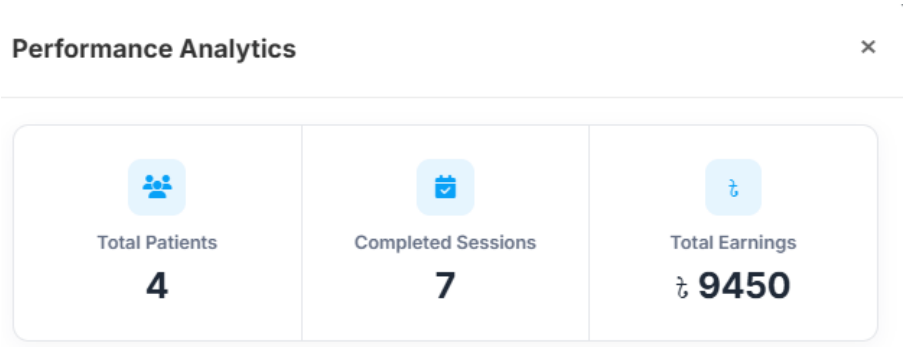


Figure 6.38: Doctor Performance Analytics

6.6 Administrative Management

6.6.1 Admin Dashboard

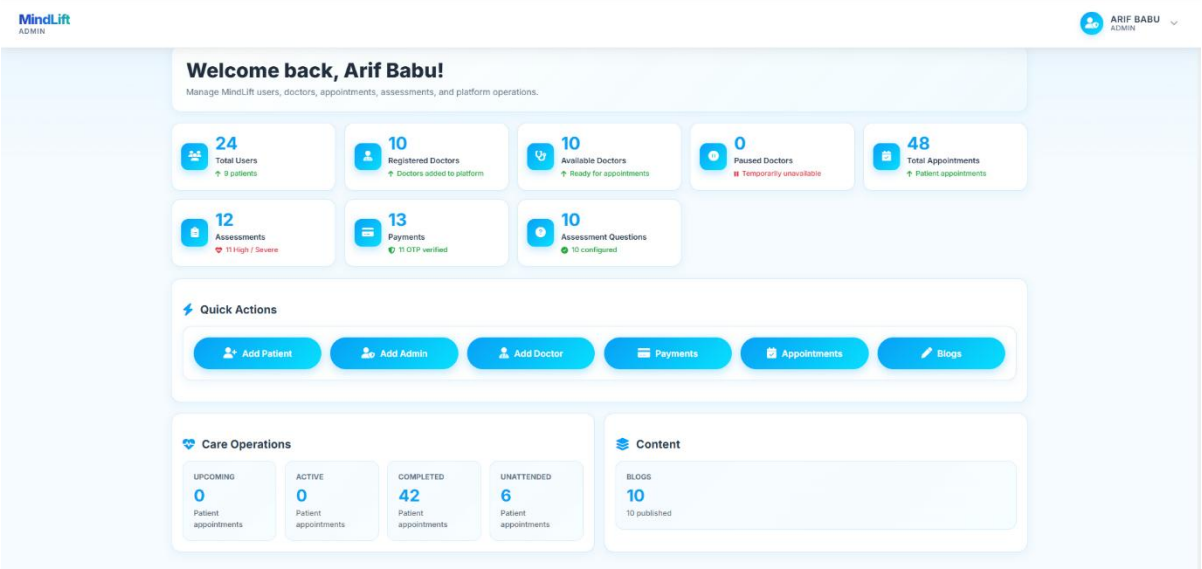


Figure 6.39: Admin Dashboard

The Admin Dashboard provides a centralized overview of MindLift's platform operations. It summarizes key information such as users, doctors, appointments, assessments, payments, assessment questions, and blog content, while also offering quick access to major administrative management functions.

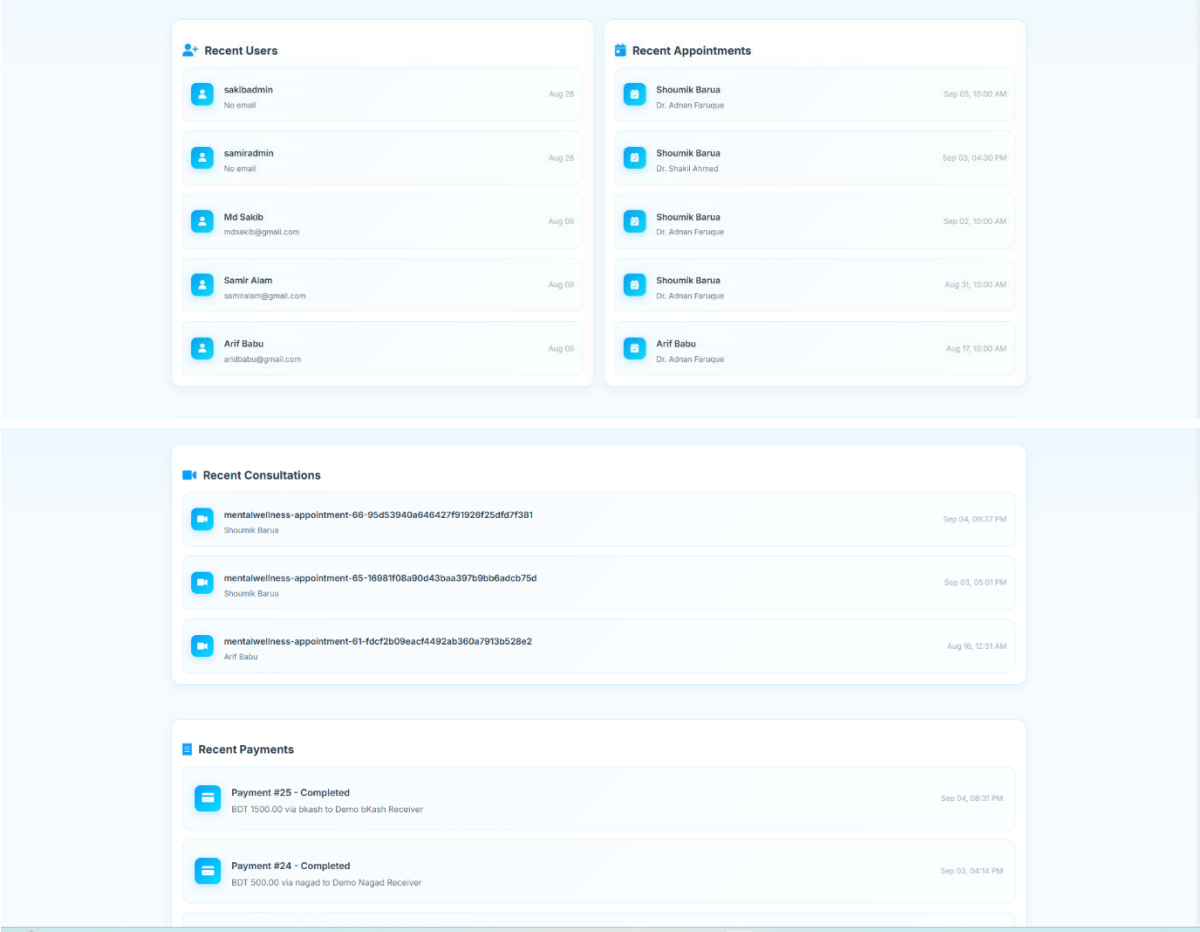


Figure 6.40: Recent Platform Activity Overview

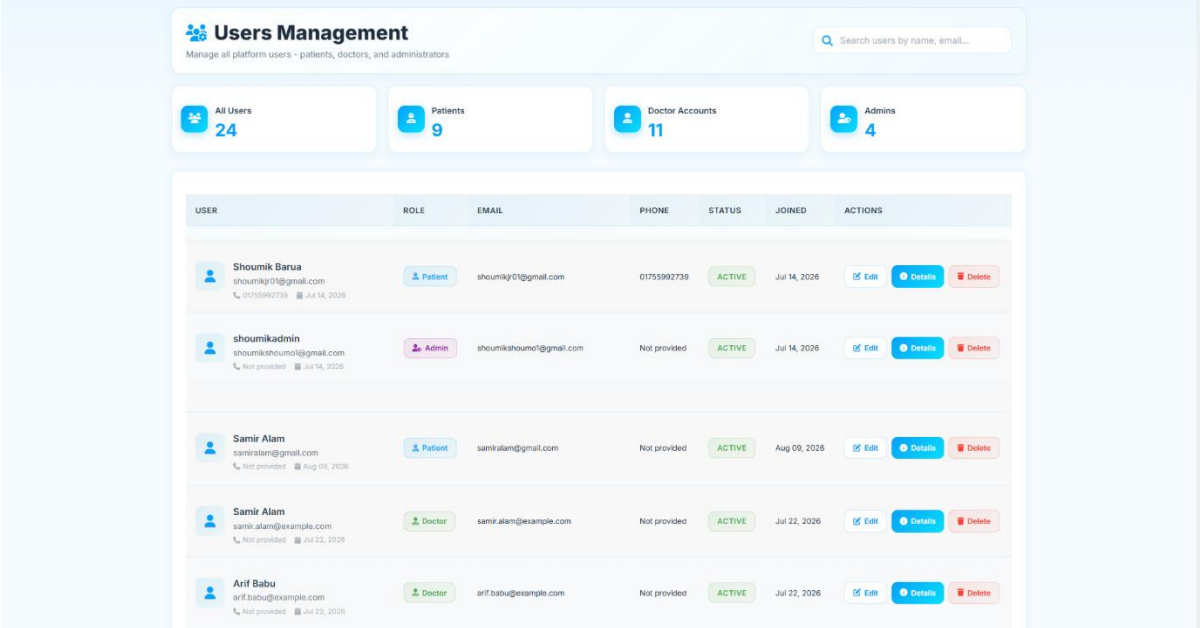


Figure 6.41: User Management

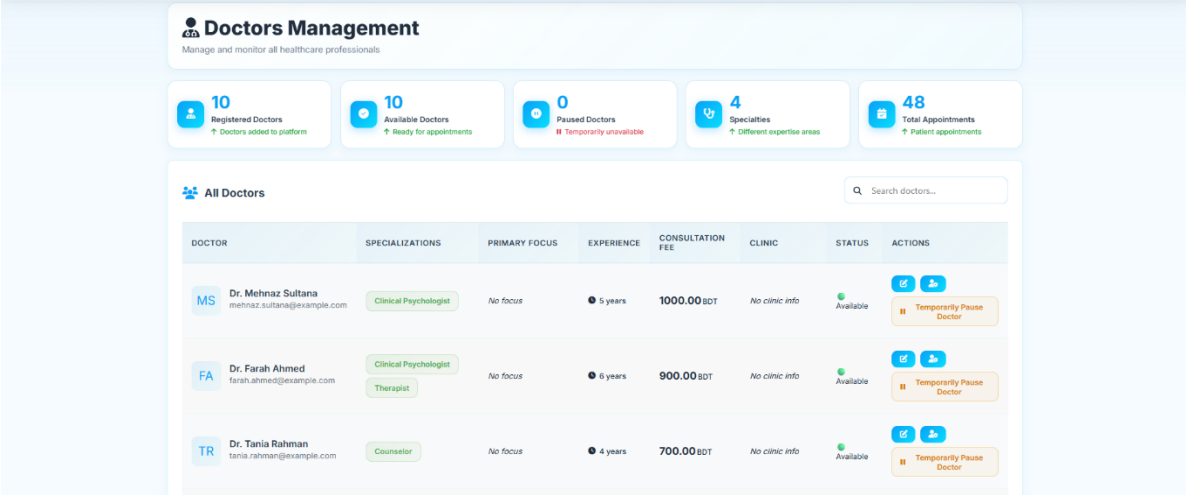


Figure 6.42: Doctor Management

The administrator can follow and manage healthcare professionals on the whole platform thanks to the Doctor Management interface. It includes registered, available and paused doctors and specialties and the number of appointments. Admins can view doctor details like specialization, experience, consultation fee, clinic details and availability status. The Temporarily Pause Doctor control lets the administrator make a doctor unavailable for new appointments without deleting the doctor's account or existing records, which is useful for short-term unavailability.

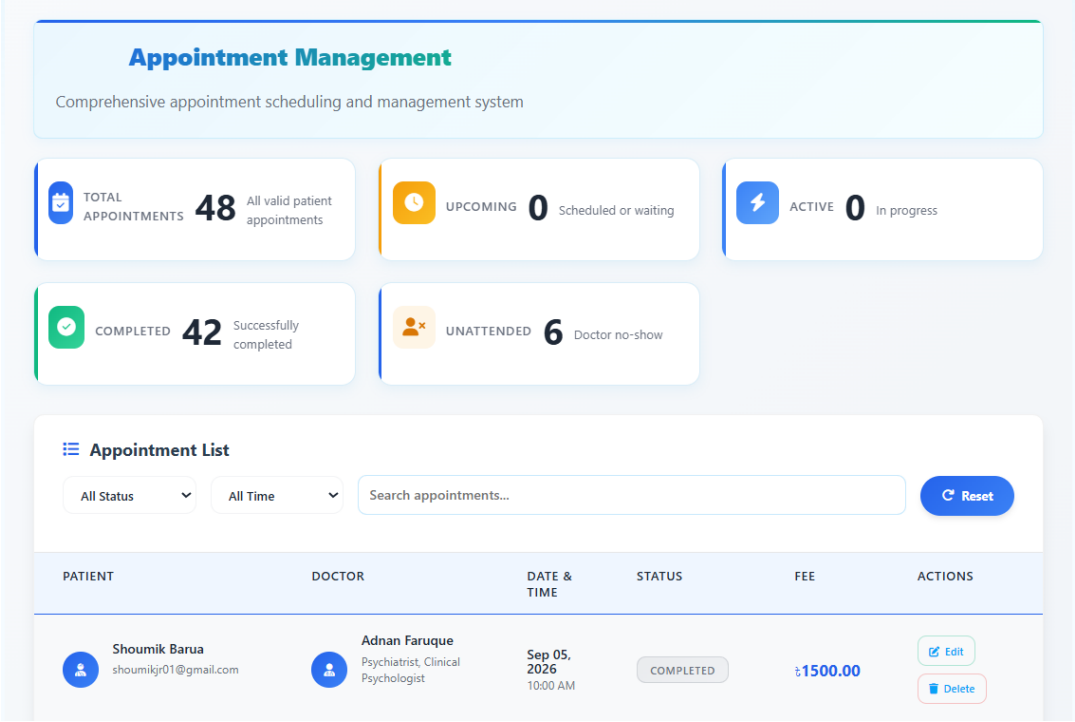


Figure 6.43: Appointment Management

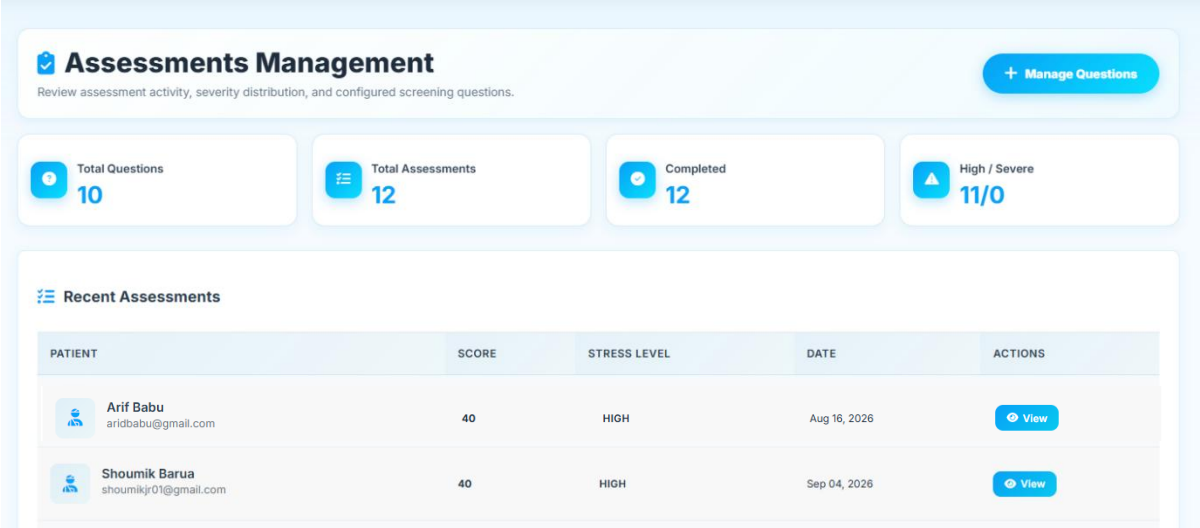


Figure 6.44: Assessment Management

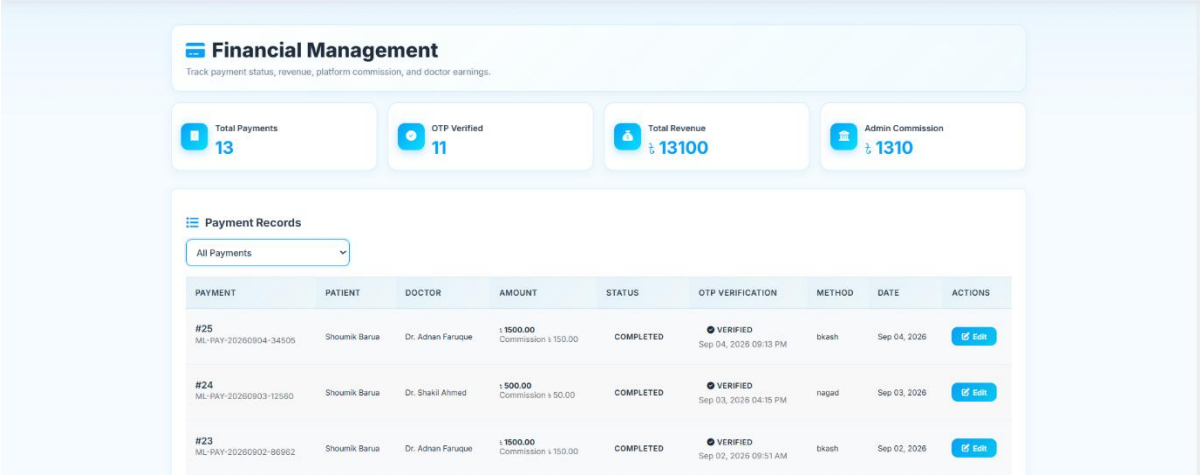


Figure 6.45: Financial Management

As shown in Figure 6.43–6.45, the administrative interfaces allow MindLift administrators to monitor several key operational areas from one management environment. Figure 6.35 presents Appointment Management, where administrators can review consultation records, appointment statuses, patient–doctor assignments, dates, times, and fees. Figure 6.36 Assessment Management Assessment Management provides a summary of completed assessment, scores, stress levels and recent assessment records of patients. Figure 6.37 Financial Management Admin can see the payment records, OTP verified transactions, total revenue, admin commission, payment methods and transaction details. Collectively these interfaces provide a central hub for monitoring consultation activity, patient assessments and financial operations.

6.6.2 Role-Based Account Creation

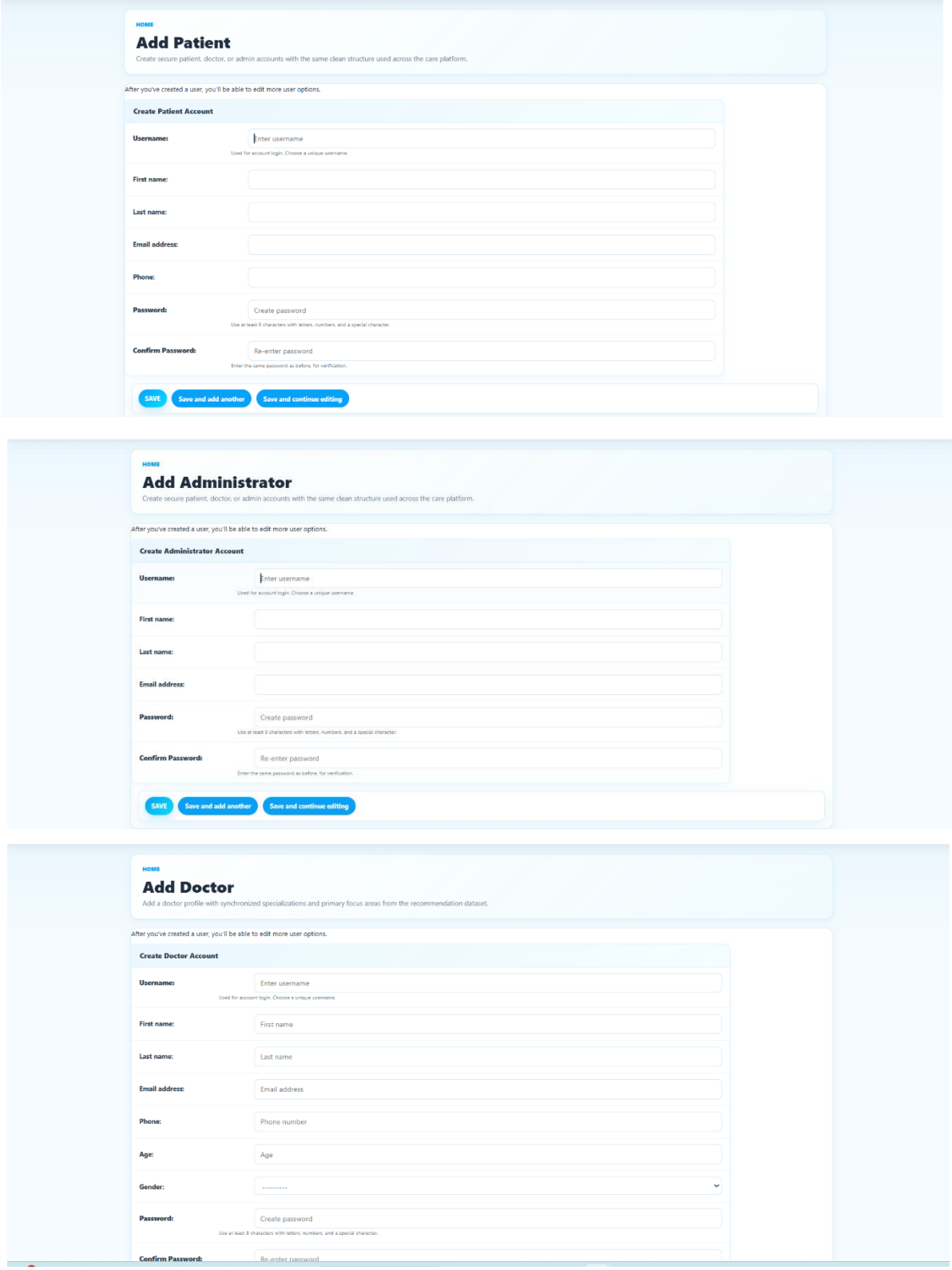


Figure 6.46: Role-Based Account Creation

Admin can create new accounts for Patients, Doctors or Administrator using role specific form. The administrator will then enter the required personal, login and role information. Depending on the account type you choose, the system will provide a particular level of access. This enables to add new users to MindLift in a structured manner, not mixing the different responsibilities and permissions of each of the roles.

6.7 Supporting Platform Information

6.6.1 About MindLift

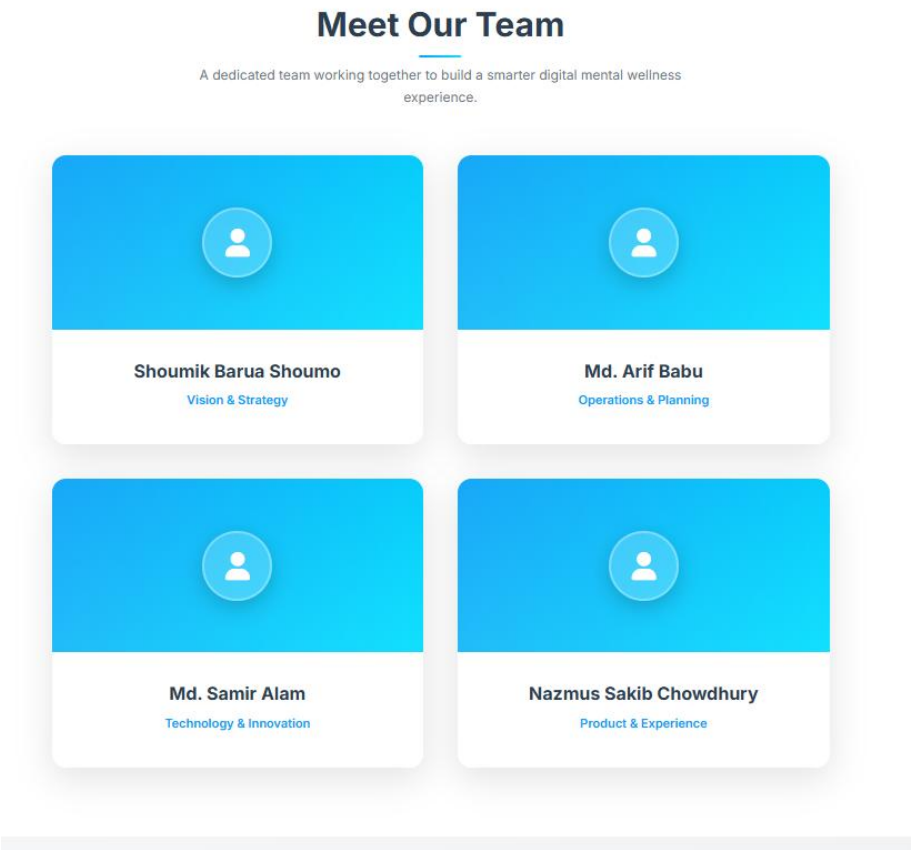


Figure 6.47: About MindLift

CHAPTER 7

PROJECT CONCLUSION AND FUTURE DEVELOPMENT

7.1 Conclusion

7.1.1 Project Summary

MindLift: An Intelligent Role-Based Digital Mental Wellness Care and Consultation Platform was developed to provide a connected environment for mental wellness assessment, professional consultation, follow-up care, educational resources, and AI-assisted support. The platform is designed around three roles—Patient, Doctor, and Administrator—each with separate responsibilities while remaining part of the same service workflow.

Patients can take mental wellness assessments, receive personalized doctor recommendations, book consultations, make payments, verify email OTP, attend online sessions using Jitsi Meet, access prescriptions and complete wellness tasks assigned by doctors. Doctors can manage their profiles, availability, appointments, consultations, prescriptions, wellness tasks, and educational articles, while administrators can manage users, assessments, appointments, payments, revenue information, and blog moderation.

MindLift also extends support beyond scheduled consultations through doctor-authored mental wellness articles, semantic blog search, AI Quick Summary, and an AI Support Chatbot. The chatbot combines safety detection, Retrieval-Augmented Generation (RAG), local knowledge, conversation context, generative AI, and controlled fallback responses. In this way, the platform supports users from initial assessment through professional care and continued wellness activities within one coordinated system.

7.1.2 System Realization

MindLift was implemented as a Django-based web application using Python, SQLite, HTML, CSS, JavaScript, and Django templates. Django handles authentication, role-based access, application logic, database interaction and the core workflows, while external services provide Jitsi video consultation, email communication, Google authentication and generative AI.

The assessment module analyzes the answers of the patient and supports the personalized recommendation of doctors based on the results of the assessment and the information about the doctors. The booking process links doctor calendars, booking, payment logs, email OTP verification, confirmation and QR-enabled appointment slips.

Confirmed appointments are moving to online consultations based on Jitsi. Doctors can save the information of consultation, prepare digital prescriptions and assign wellness tasks so that the care does not stop at the live session.

The intelligent functions were integrated within the same architecture. Semantic blog search supports contextual content discovery, AI Quick Summary generates concise summaries with

key takeaways, and the chatbot combines safety processing, RAG retrieval, recent conversation context, generative AI, and local fallback handling.

Administrators manage the broader operation of MindLift through role-specific interfaces for users, doctors, assessments, appointments, payments, revenue information, and blog content. The implemented platform therefore reflects the proposed design as a functioning and interconnected system.

7.1.3 Key Functional Contributions

The major functional contributions of MindLift can be summarized as follows:

- **Assessment and Personalized Recommendation:** Mental wellness assessment results contribute directly to doctor recommendations, helping users identify professionals whose specialization and focus are more relevant to their needs.
- **Connected Consultation Journey:** Doctor schedules, appointment booking, payment verification, email OTP confirmation, QR-enabled receipts, Jitsi consultation, digital prescriptions, and wellness tasks operate as successive stages of one care process.
- **AI-Assisted Mental Wellness Support:** The chatbot combines safety detection, intent analysis, RAG, local knowledge, recent conversation context, generative AI, and controlled fallback responses. There is a different path for interactions concerning safety and emergency situations.
- **Intelligent Content Access:** Doctor-authored articles with contextually relevant content and quick summaries with key takeaways are provided through Semantic blog search and AI Quick Summary.
- **Role-Based Administration:** Separate management environment for admins to manage patients, doctors, assessments, appointments, payments, revenue data and blogs moderation. These contributions mean that MindLift isn’t just an appointment booking system but rather a combination of professional care, follow up support, educational information and AI-assisted functionality and administration all working together into one platform.

7.1.4 Overall Impact

MindLift provides users with a structured pathway from identifying a mental wellness concern to receiving professional consultation and continuing follow-up care. Assessment and personalized recommendation can reduce uncertainty at the beginning of the process, while appointment management, online consultation, prescriptions and wellness tasks maintain the continuity afterward.

For doctors, the platform organizes professional information, schedules, appointments, consultation records, prescriptions, wellness tasks, and educational content within one environment. Administrators receive the broader tools needed to supervise users, appointments, assessments, payments, and content. A mental health platform cannot depend

only on attractive pages; it needs governance, review, permissions, logs, and accountability [30].

The project also demonstrates how AI-assisted functions can complement professional care. Alternative means to retrieve relevant information include semantic search, article summarization, and RAG-assisted chatbot, but professional consultation remains the foundation of the care process. This gives MindLift a practical foundation to grow further as an integrated digital mental wellness service.

7.2 Further development

7.2.1 Functional Extension

The existing platform can be extended in future developments without changing the basic workflow. Priority areas are:

- Tracking assessment results, appointments, and wellness task completion over time.
- More refined doctor matching based on consultation fee, preferred time, language preference, workload, consultation history and patient feedback.
- Improved emergency resources and well-designed, consent-based trusted contact features.
- Reminders of appointments and assigned wellness activities at scheduled times.
- Support for Mobile or Progressive Web Application for easy access to assessments, appointments, prescriptions, wellness tasks, articles and AI assisted services.

These additions would expand the utility of MindLift while maintaining its current assessment-to-consultation and follow-up format.

7.2.2 Interface Improvement

The future of interface development should extend, not replace, the existing role-based design. A more context aware patient dashboard could emphasize different information, depending on whether the user is currently completing an assessment, has a consultation coming up or has active wellness tasks. The assessment results could have more visuals to explain and compare progress. Doctor discovery could have more filters like specialization, fee and availability. The consultation environment could also bring together patient information, consultation notes, prescription preparation and wellness-task assignment into a more consolidated workspace for doctors.

As the amount of platform data increases, the administrator interface could benefit from stronger filtering, reporting, export, bulk-management, and audit-history functions. Such improvements would reduce unnecessary navigation while keeping the established role structure unchanged.

7.2.3 System Security Enhancement

The current implementation already applies authentication, role-based access, server-side validation, CSRF protection, and controlled access to sensitive workflows. A real-world deployment, however, would require additional protection for assessment, consultation, prescription, payment, and emergency-related records.

Future security improvements could include:

- field-level encryption for sensitive information;
- more context-specific access permissions;
- stronger session and re-authentication controls;
- suspicious-login monitoring;
- verified payment-gateway callbacks and transaction auditing;
- encrypted backups and tested recovery procedures;
- clearer privacy and data-management controls for patients.

These measures would strengthen the existing security foundation without requiring a redesign of the platform's current role-based architecture.

7.2.4 Long-Term Scalability

The current Django and SQLite environment is suitable for academic development and demonstration. For larger deployment, a production-oriented database such as PostgreSQL could provide stronger concurrent access, indexing, backup management, and long-term data handling.

More extensive use would also require better observation of Jitsi consultations, application performance, email delivery, AI-service availability, and database activity. A dedicated API could be used for future mobile apps and external integrations. Cloud infrastructure could be used to separate the application hosting, database services, file storage, backups and monitoring. As AI-assisted features proliferate, RAG retrieval, semantic search, chatbot processing, and article summarization might eventually be moved to more specialized service components. These enhancements would allow the platform to scale to a larger number of users without degrading the current care process.

7.3 Final Remarks

MindLift demonstrates how assessment, personalized doctor recommendation, appointment confirmation, online consultation, prescriptions, wellness tasks, educational content, and AI-assisted support can operate as parts of one connected mental wellness platform.

Its principal strength lies in continuity. A patient can move from assessment to recommendation, booking, email OTP confirmation, Jitsi consultation, and follow-up care,

while doctors and administrators perform their responsibilities through separate role-specific interfaces. Semantic blog search, AI Quick Summary, and the RAG-assisted chatbot further extend support beyond scheduled consultations without replacing professional care.

The completed project provides a practical foundation for future development in security, scalability, mobile access, progress tracking, and service integration. More importantly, it demonstrates a structured approach to making digital mental wellness support easier to access, navigate, and continue over time.

REFERENCES

[1] "২০২৫ সালে ৪০৩ শিক্ষার্থীর আত্মহত্যা, অর্ধেকই স্কুলের," *Samakal*, Bangladesh,Mar.1,2026. [Online] Available: https://samakal.com/bangladesh/article/340806/

[2] "মানসিক রোগীর ৯০ শতাংশই চিকিৎসার বাইরে," *Dainik Amader Shomoy*, Bangladesh, Oct. 10, 2024. [Online] Available: https://www.dainikamadershomoy.com/details/01927285b8a62/

[3] A. Dang, D. Arora, and P. Rane, "Role of digital therapeutics and the changing future healthcare," *J. Family Med. Prim. Care*, vol. 9, pp. 2207-13, 2020. [Online] Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7380804/

[4] J. Sin, G. Galeazzi, E. McGregor, J. Collom, A. Taylor, B. Barrett, V. Lawrence, and C. Henderson, "Digital Interventions for Screening and Treating Common Mental Disorders or Symptoms of Common Mental Illness in Adults: Systematic Review and Meta-analysis," *J. Med. Internet Res.*, vol. 22, no. 9, p. e20581, Sep. 2020. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7495259/

[5] A. Ben Fadhel, D. Bianculli, and L. Briand, "A Comprehensive Modeling Framework for Role-based Access Control Policies," Software Verification and Validation Laboratory: A Comprehensive Modeling Framework for Role-based Access Control Policies, TR-SnT-2014-15, Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, pp. 5-6, Apr. 2015. [Online] Available: https://orbilu.uni.lu/bitstream/10993/18874/1/SnT-TR-2014-15.pdf

[6] S. Inam, "Mental health matters," *The Financial Express*, May 10, 2025. [Online] Available: https://today.thefinancialexpress.com.bd/editorial/mental-health-matters-1746802866.

[7] World Health Organization, United Nations Development Programme, and ARK Foundation, "A case for investment: Prevention and management of mental health conditions in Bangladesh," *Bangladesh Mental Health Investment Case*, pp. 11–12, 2021. [Online]. Available: https://cdn.who.int/media/docs/default-source/searo/bangladesh/publications/mental-health-investment-case-in-bangladesh.pdf

[8] LifeSpring, Bangladesh, [Online]. Available: https://www.lifespringint.com/

[9] Moner Bondhu, Bangladesh, [Online]. Available: https://monerbondhu.com/

[10] MindSheba, Bangladesh, [Online]. Available: https://mindsheba.com/

[11] BetterHelp, [Online]. Available: https://www.betterhelp.com/

[12] Talkspace, [Online]. Available: https://www.talkspace.com/

[13] OWASP, "A01:2025 - Broken Access Control," OWASP Top 10:2025, 2025. [Online]. Available: https://owasp.org/Top10/2025/

[14] Django Software Foundation, "Models," Django documentation, Dec. 2025. [Online]. Available: https://docs.djangoproject.com/en/6.0/topics/db/models/

[15] OWASP Foundation, "Validate All Inputs," OWASP Developer Guide, 2023. [Online]. Available: https://devguide.owasp.org/en/04-design/02-web-app-checklist/05-validate-inputs/

[16] Django Software Foundation, "The Django admin site," Django Documentation 4.2, 2023. [Online]. Available: https://docs.djangoproject.com/en/4.2/ref/contrib/admin/

[17] S. G. Tetteh, “Empirical study of agile software development methodologies: A comparative analysis,” *Asian Journal of Research in Computer Science*, vol. 17, no. 5, pp. 30–42, 2024. [Online]. Available: https://www.researchgate.net/profile/Samuel-Gbli-Tetteh/publication/378534256_Empirical_Study_of_Agile_Software_Development_Methodologies_A_Comparative_Analysis/links/66057340f5a5de0a9fe7a256/Empirical-Study-of-Agile-Software-Development-Methodologies-A-Comparative-Analysis.pdf

[18] L. J. Boulos, A. Mendes, A. Delmas, and I. Chraibi Kaadoud, “An iterative and collaborative end-to-end methodology applied to digital mental health,” *Frontiers in Psychiatry*, vol. 12, pp. 1–21, 2021. [Online]. Available: https://doi.org/10.3389/fpsyt.2021.574440

[19] S. Z. Mbawa, R. A. J. de Vries, L. C. Siebert, K. van Turnhout, and W.-P. Brinkman, “Designing and Evaluating Digital Mental Health Interventions: Scoping Review,” *JMIR Mental Health*, vol. 13, pp. 1–24, 2026. [Online]. Available: https://doi.org/10.2196/77038

[20] Nivedhaa, “Software architecture evolution: patterns, trends, and best practices,” *International Journal of Computer Sciences and Engineering*, vol. 1, no. 2, pp. 1–14, 2024. [Online]. Available: https://iaeme.com/MasterAdmin/Journal_uploads/IJCSE/VOLUME_1_ISSUE_2/IJCSE_01_02_001.pdf

[21] Django Software Foundation, “Django documentation,” *Django documentation*, version 6.0, 2025. [Online]. Available: https://docs.djangoproject.com/en/6.0/

[22] S. Vial, S. Boudhraâ, and M. Dumont, “Human-Centered Design Approaches in Digital Mental Health Interventions: Exploratory Mapping Review,” *JMIR Mental Health*, vol. 9, no. 6, p. e35591, 2022. [Online]. Available: https://mental.jmir.org/2022/6/e35591

[23] D. Hidayat, C. H. Pangaribuan, O. B. Putra, and I. Irawan, “Contemporary studies of task-technology fit: a review of the literature,” in *2021 International Conference on Information Management and Technology (ICIMTech)*, pp. 309–313, 2021. [Online]. Available: https://www.researchgate.net/publication/354597050_Contemporary_Studies_of_Task-Technology_Fit_A_Review_of_the_Literature

[24] J. Sweller, “Cognitive Load Theory: What We Learn and How We Learn,” *The Encyclopedia of Child and Adolescent Development*, S. Hupp and J. D. Jewell, Eds. Cham, Switzerland: Springer, pp. 137–152, 2023. [Online]. Available: https://link.springer.com/rwe/10.1007/978-3-319-17461-7_50

[25] C. Panakaduwa, P. Coates, M. Munir, and O. De Silva, “Optimising visual user interfaces to reduce cognitive fatigue and enhance mental well-being,” in *Proceedings of the 35th Annual Conference of the International Information Management Association (IIMA)*, Manchester, UK, Sep. 2–4, 2024, pp. 1–16. [Online]. Available: https://www.researchgate.net/publication/384392176_Optimising_Visual_User_Interfaces_to_Reduce_Cognitive_Fatigue_and_Enhance_Mental_Well-being

[26] H. Forsey, D. Leahy, B. Fields, S. Minocha, S. Attfield, and T. Snell, “Designing for learnability: improvement through layered interfaces,” *Ergonomics in Design: The Quarterly of Human Factors Applications*, vol. 33, no. 3, pp. 135–141, 2025. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/10648046241273291

[27] V. W. S. Cheng, S. E. Piper, A. Ottavio, T. A. Davenport, and I. B. Hickie, “Recommendations for designing health information technologies for mental health drawn from self-determination theory and co-design with culturally diverse populations: template analysis,” *Journal of Medical Internet Research*, vol. 23, no. 2, e23502, 2021. [Online]. Available: https://www.jmir.org/2021/2/e23502/

[28] M. Saleem, L. Kühne, K. K. De Santis, L. Christianson, T. Brand, and H. Busse, “Understanding engagement strategies in digital interventions for mental health promotion: scoping review,” *JMIR Mental Health*, vol. 8, no. 12, e30000, Dec. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S236879592100072X

[29] C. Berardi, M. Antonini, Z. Jordan, H. Wechtler, F. Paolucci, and M. Hinwood, “Barriers and facilitators to the implementation of digital technologies in mental health systems: a qualitative systematic review to inform a policy framework,” *BMC Health Services Research*, vol. 24, pp. 1–19, 2024. [Online]. Available: https://link.springer.com/article/10.1186/s12913-023-10536-1

[30] World Health Organization, *World mental health report: transforming mental health for all*. Geneva, Switzerland: World Health Organization, pp. 52–54, 2022. [Online]. Available: https://www.who.int/publications/i/item/9789240049338