

**CureBridge**

**BS(CS) Fall 2024**

**Project Supervisor**

**Nida Munawar**

**Project Co-supervisor**

**Dr Nauman Durrani**

**Project TEAM**

**Kuljeet Khatri K21-3362**

**Hassan Abbas K21-3286**

**Syed Muhammad Zaid K21-3348**

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

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES KARACHI CAMPUS

May 2025

| Project Supervisor | Miss Nida Munawar |
| --- | --- |
| Project Team | Kuljeet Khatri Name K21-3362  S.M. Zaid Name K21-3348  Hassan Abbas Name K21-3286 |
| Submission Date | May 15, 2025 |

Miss. Nida Munawar

Supervisor

Dr. Nauman Durrani

Co-Supervisor

Dr. Ghufran Ahmed

Head of Department

**FAST SCHOOL OF COMPUTING**

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES KARACHI CAMPUS**

**Acknowledgement:**

All thanks to the supervisor, Miss. Nida Munawar and co-supervisor Mr. Dr. Nauman Durrani, the members of the project, Kuljeet Khatri, Syed Muhammad Zaid, Hasan Abbas, and people who cooperated in providing the data for the project. All the mentioned people have contributed to the project and helped us on this journey. Without their help, we could not have done this.

Table of Contents

[Abstract 5](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.lmfy5d50ka0c)

[Introduction 6](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.lhkrozxqe9g)

[Related Work 7](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.eg01ztofqzq1)

[Requirements 8](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.yv68gbdg4x60)

Functional Requirements [8](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.493lxx8robkf)

User Authentication (Log-In) [8](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.pcypld7s8yvx)

Call Integration8

Real-Time Transcription9

Symptom Extraction & Disease Prediction9

Sentiment Analysis9

Profile Management10

[Non-Functional Requirements 1](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.n51xlike5vwb)0

[Design](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.3fiuhie42rqx) 12

System Architecture [1](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.g5r5f1om9xj5)2

System Flow 13

Use Case Diagram 14

Database Schema15

User Interfaces (Patient/Doctor/Admin) 16

[Implementation](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.u4aqipg0ooj) 18

[Testing](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.p7pcv4ww9r9n) 22

[Conclusion](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.2bd92pnyrj28) 23

[References](https://docs.google.com/document/d/1phyHy8YnpJ3NLbdqh96mLX9gbm3n8WMVPAfcHALLivg/edit#heading=h.9qccf6pdoez) 24

**Abstract**

Telehealth platforms have improved healthcare accessibility but lack robust monitoring, diagnostic support, and oversight of patient-doctor interactions. CureBridge bridges these gaps by integrating real-time transcription, symptom extraction, disease prediction, and sentiment analysis to enhance telehealth consultations. The platform allows patients to book online consultations and receive real-time conversation transcriptions. AI models then extract symptoms, assist doctors in disease prediction, and detect unprofessional behavior through sentiment analysis, promoting a professional and secure telehealth experience. Additionally, CureBridge centralizes patient records, enabling easy access for both patients and doctors, thereby improving continuity of care. Unlike traditional telehealth platforms that primarily facilitate communication, CureBridge enhances diagnostic accuracy, accountability, and efficiency. By addressing critical gaps in telehealth, it aims to improve patient outcomes, streamline doctor workflows, and uphold high healthcare standards.

**Introduction**

With the increasing demand for remote consultations, telehealth has become one of the defining segments of modern health care. It has brought medical access closer to many people. However, the existing telehealth platforms have many issues. They mainly include a lack of effective monitoring tools for consultations, limited support for real-time disease diagnosis, and dialysis about efficient access and maintenance of patient records.

As a response to the above-mentioned problems, our group has been developing a telehealth platform CureBridge based on AI and web technologies, which we believe should improve the entire experience of telehealth. The enhancement in the efficiency, trustworthiness, and accessibility of healthcare services to patients, providers, and administrators is what really drives us.

Thus, some of the project components include:

Real-time transcription: This converts conversations into text for better reference and secure record-keeping.

Symptom Extraction: Automatically identifying symptoms from live transcription data during consultations.

Disease prediction: Providing real-time suggestions for possible diseases based on the extracted symptoms to the doctor.

Sentiment Analysis: Detecting toxic or unprofessional behavior in consultations so that a healthy, respectful environment is ensured.

Centralized patient records: Allow doctors and patients to access the entire medical history and previous consultations with ease.

This report is for the progress we have thus far made towards our goals by focusing on the development of the symptom extraction and disease prediction models. This also contains a review of the related works, the methods we followed, testing results, and the design of the system. Goals for the next phase of the project and improvement plans were explicitly laid down.

**Related Work / (SRS/SDS)**

Telehealth platforms have come a long way in improving healthcare by making remote consultations possible. However, most of the existing platforms such as Amwell, MDLive lack a combination of features that are essential for providing a truly effective and reliable telehealth experience. Based on ongoing research trends and industry needs, features like real-time transcription, symptom extraction, disease prediction, and sentiment analysis are increasingly recognized as essential for improving telehealth services. This is the gap our project, CureBridge, aims to fill.

In the field of symptom extraction, a lot of research has focused on using natural language processing (NLP) techniques to improve accuracy. Studies have explored tools like MetaMap, MedLEE, and machine learning models such as BERT, which are designed to process medical conversations and identify symptoms effectively. However, these methods still face challenges, especially when dealing with complex or ambiguous medical terms.

Similarly, research in disease prediction is rapidly advancing, with efforts aimed at assisting doctors by providing real-time AI-driven diagnostic suggestions. Techniques like decision trees, random forests, and neural networks are being employed to predict diseases based on extracted symptoms. While these methods have shown promising results, but overall result depends on the dataset.

Another critical area of focus is sentiment analysis in healthcare, which is essential for maintaining professionalism in doctor-patient interactions. Research in sentiment analysis leverages models like LSTM, GRU, XLNet, and transformers to detect emotions such as frustration, anger, or disrespect in textual data. Studies have shown that sentiment analysis can help flag unprofessional or toxic behavior, thereby improving the overall quality of telehealth services.

To ensure our project addresses these gaps effectively, we developed two key documents:

1. Software Requirements Specification (SRS): This document lists all the functional and non-functional requirements, including use cases, user stories, constraints, and interfaces, to ensure clarity in the project’s scope.
2. Software Design Specification (SDS): This document outlines the technical design and architecture, including diagrams like class and sequence diagrams to guide implementation.

By combining insights from existing platforms, research advancements in symptom extraction, disease prediction, and sentiment analysis, CureBridge aims to create a telehealth platform that addresses critical shortcomings while providing a reliable and innovative healthcare solution.

**Functional Requirements**

### **User Authentication (Log-In)**

**Token-Based Authentication:**

* The platform uses a secure token-based login system that ensures authenticated access to different user roles (patient, doctor, admin). On successful login, a JWT (JSON Web Token) is generated and used for subsequent requests to maintain session security without storing sensitive data on the client side.

**Secure Session Management:**

* All tokens are time-bound and auto-expire, minimizing unauthorized access risks. Refresh tokens are used for maintaining longer sessions securely.

### **Call Integration**

**Room API & Socket Communication:**

* CureBridge utilizes Room APIs combined with WebSockets to establish secure, low-latency video call sessions between doctors and patients. This ensures seamless, real-time communication.

**Session Management:**

* The backend manages session states and signaling via sockets to maintain synchronized audio/video streams and support call initiation, joining, and termination functionalities.

### **Real-Time Transcription**

**Deepgram Integration:**

* Real-time voice-to-text transcription is powered by Deepgram, enabling instant transcription of doctor-patient conversations. This functionality supports both parties in keeping track of medical communication without delay.

**Live Display:**

* Transcripts are displayed live on the frontend, enhancing accessibility and improving post-consultation record-keeping.

**Symptom Extraction & Disease Prediction**

**BioBERT for NER:**

* A fine-tuned BioBERT Named Entity Recognition (NER) model is used to automatically extract symptoms from the live transcribed text.

**Stacked Ensemble Model for Disease Prediction:**

* The extracted symptoms are fed into an ensemble of machine learning models (Random Forest, Decision Tree, and SVM) with Logistic Regression acting as a meta-estimator to predict possible diseases. This enhances diagnostic support and decision-making.

### **Sentiment Analysis**

**XLNet-Based Sentiment Classifier:**

* An advanced XLNet model is used to analyze the emotional tone and language used during doctor-patient conversations. This model helps detect unprofessional behavior, stress indicators, or emotionally charged language in real time.

**Flagging & Toxicity Detection:**

* If any portion of the conversation is detected as toxic or unprofessional, it is automatically flagged by the system. These flags are stored along with the conversation transcript and associated sentiment scores.

**Admin Review Panel:**

* Admins have exclusive access to a dedicated sentiment analysis dashboard. This panel allows them to:
  + View flagged consultations.
  + Analyze sentiment scores and keywords contributing to the flagged status.
  + Take necessary actions such as issuing warnings, conducting reviews, or contacting the involved parties.

**Quality Assurance & Compliance:**

* This functionality ensures the platform upholds ethical standards during consultations, reinforces professionalism, and supports compliance with healthcare communication norms.

### **Profile Management**

**Admin-Centric Control:**

* Admins are responsible for creating, updating, or disabling profiles of doctors and patients. This ensures controlled access and avoids unauthorized account creation.

**Role-Specific Views:**

* Each user role (patient, doctor, admin) sees a tailored interface and functionality set, ensuring user-friendly navigation and role-specific tools.

**Non-functional Requirements**

**Performance Requirements**

**Real-Time Response:**

* Video calls should be smooth and without noticeable lag.
* Real-time transcription must process and display text with minimal latency.

**Throughput:**

* The platform should handle multiple consultations simultaneously without performance bottlenecks.

**AI Model Efficiency:**

* Symptom extraction and disease prediction models should provide accurate results in real time.

**Load Handling:**

* The system must manage a high volume of users during peak hours without crashing or slowing down.

**Safety Requirements**

**Patient Data Accuracy:**

* Ensure transcription and diagnosis suggestions are reliable to avoid miscommunication or errors in patient care.

**System Reliability:**

* Avoid unexpected system crashes during live consultations, which could disrupt healthcare services.

**Backup and Recovery:**

* Regular data backups and a recovery plan must be in place to prevent data loss.

**Security Requirements**

**Data Encryption:**

* All communication, including video calls, transcriptions, and stored data, must be encrypted.

**Role-Based Access:**

* Different access levels should ensure that users only have access to data and functionalities relevant to their role (e.g., patients cannot access admin tools).

**Authentication:**

* Secure login mechanisms, such as two-factor authentication, should be implemented.

**Data Privacy:**

* Ensure compliance with healthcare data regulations (e.g., HIPAA) to protect sensitive patient information.

**Anomaly Detection:**

* Flag and log unauthorized access attempts for security review.

**User Documentation**

**Patient Guide**:

* Instructions for booking consultations, accessing records, and viewing real-time transcriptions.

**Doctor Guide:**

* Steps for managing appointments, accessing patient histories, using AI diagnostic tools, and reviewing transcripts.

**Administrator Guide:**

* Guidelines for overseeing consultations, managing flagged interactions, and maintaining system operations.

**Troubleshooting Guide:**

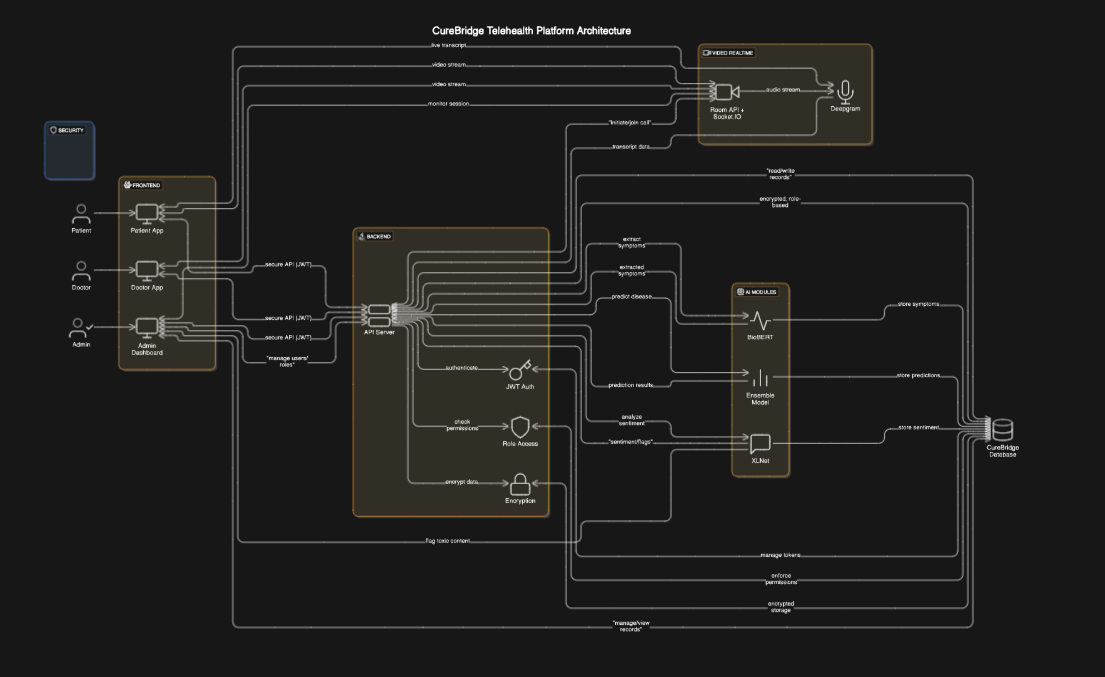
* Common issues and solutions for video calls, transcription errors, and accessing records.

**Setup Guide:**

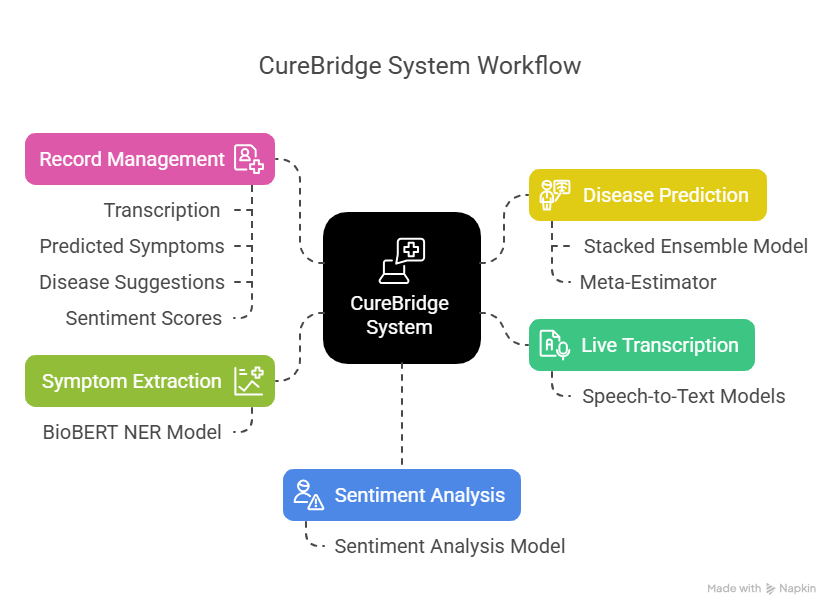
* Instructions for users to set up their accounts and configure the platform on their devices.

**Design**

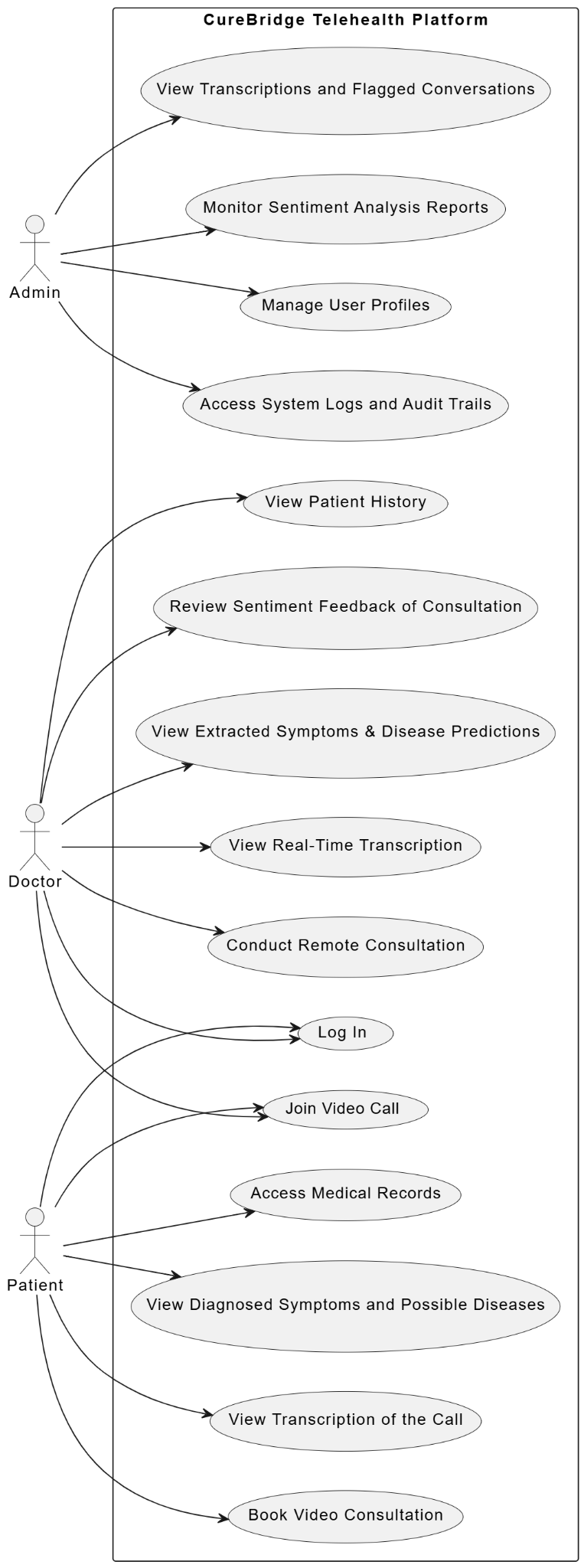
**System Architecture**

****

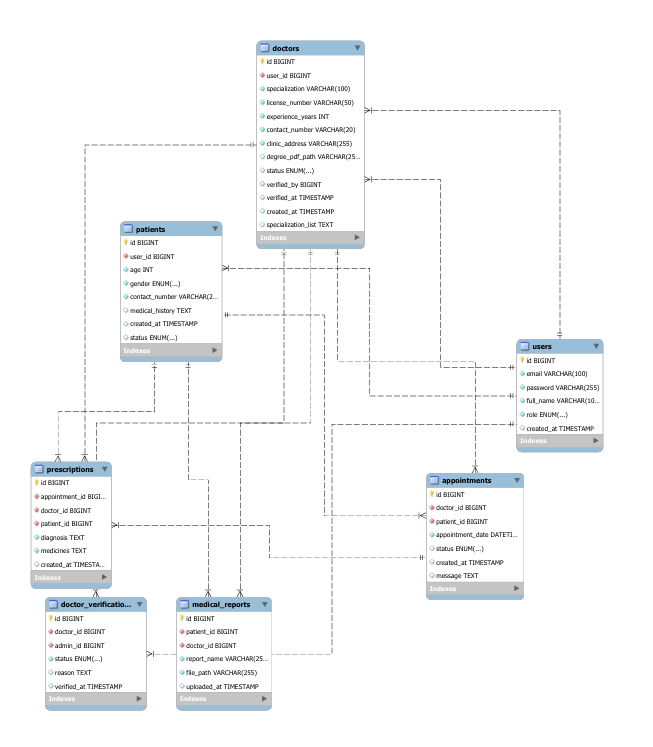
**System Flow**

****

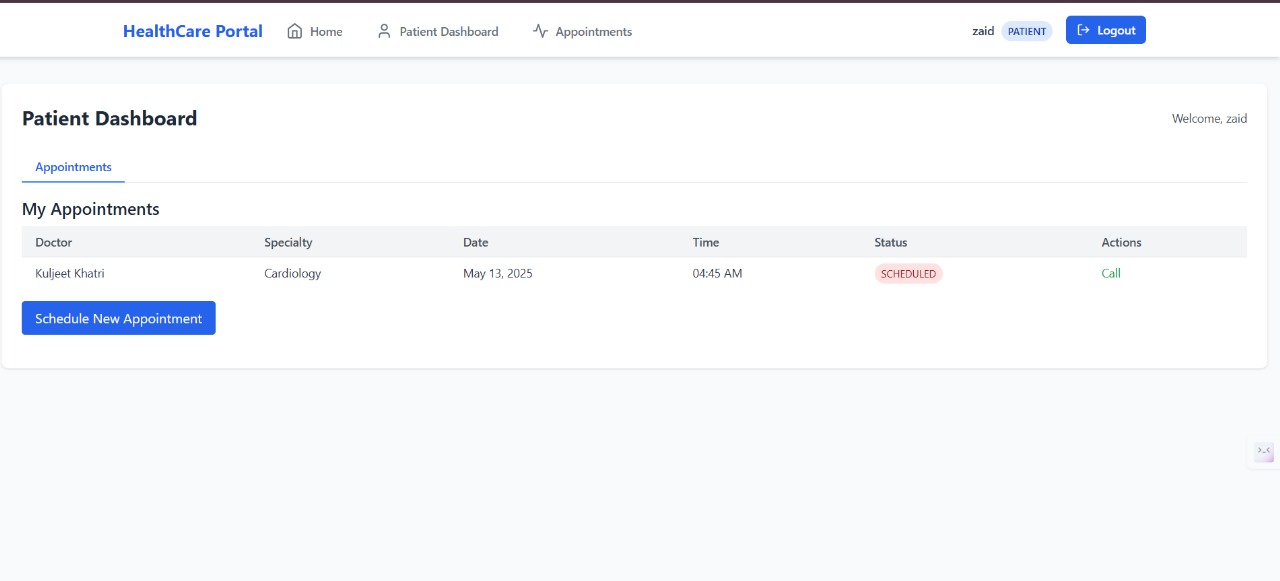
**Use Case Diagram**

****

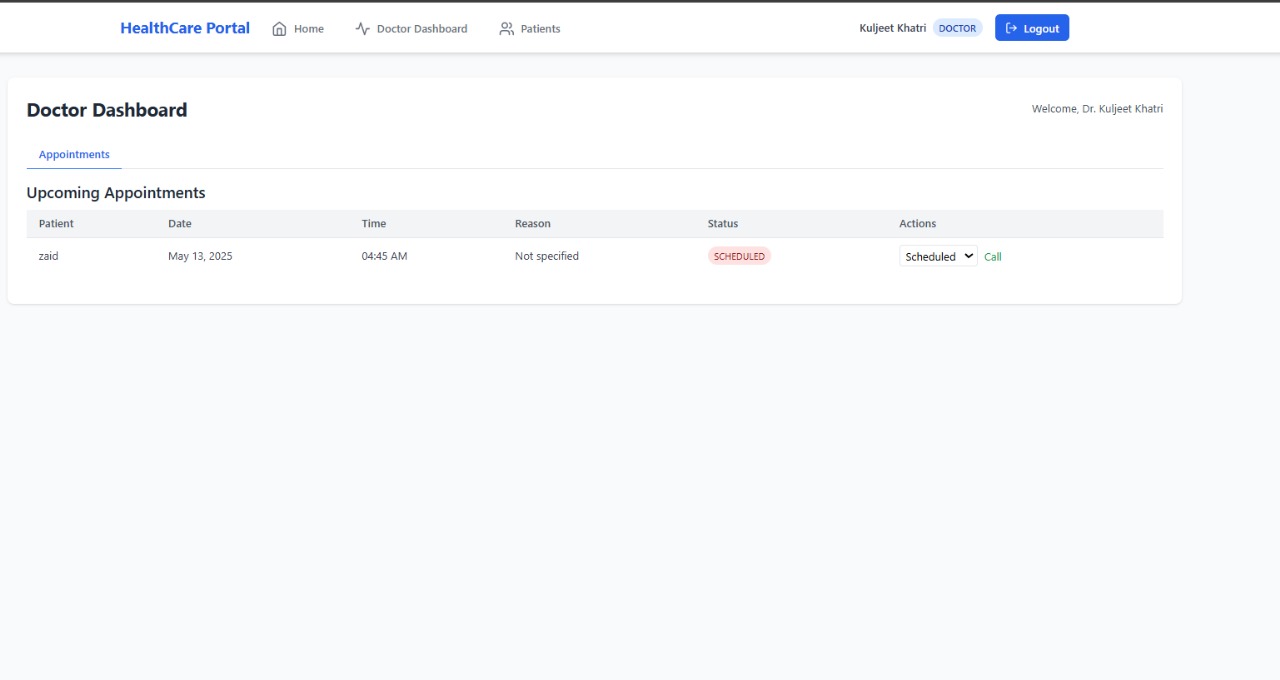
**Database Schema**

****

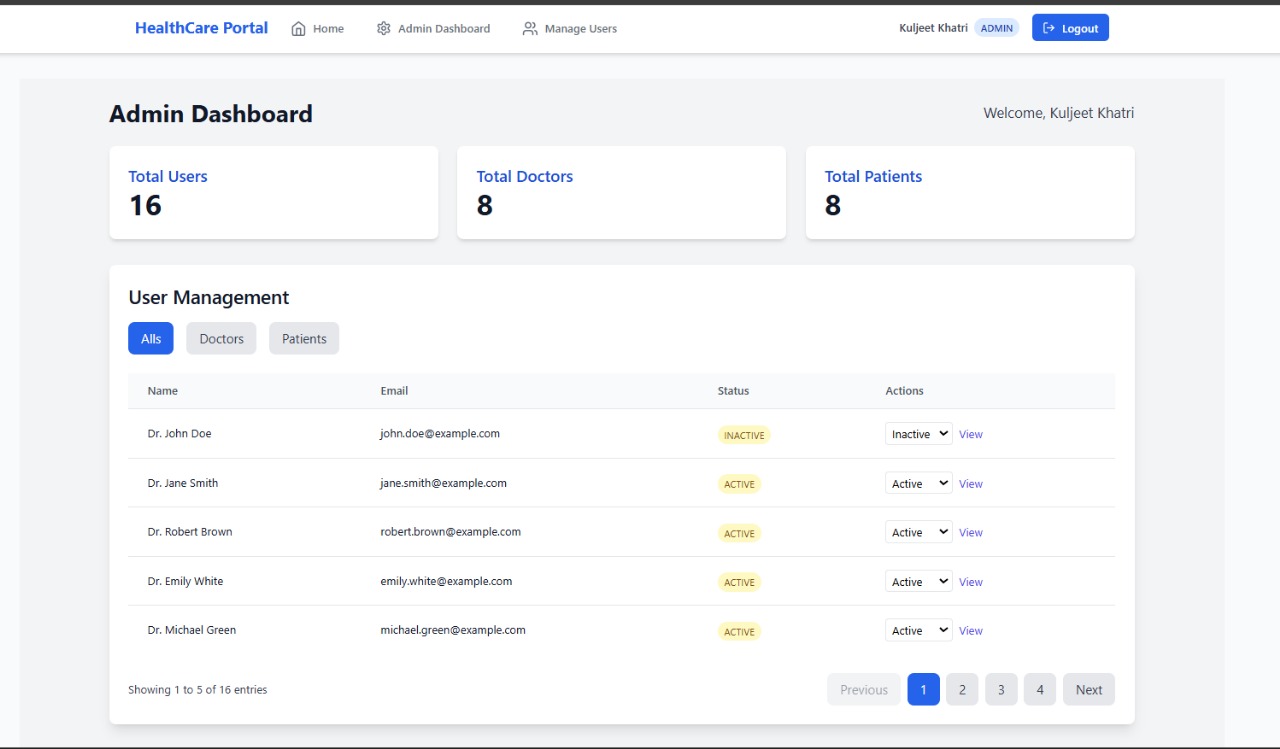
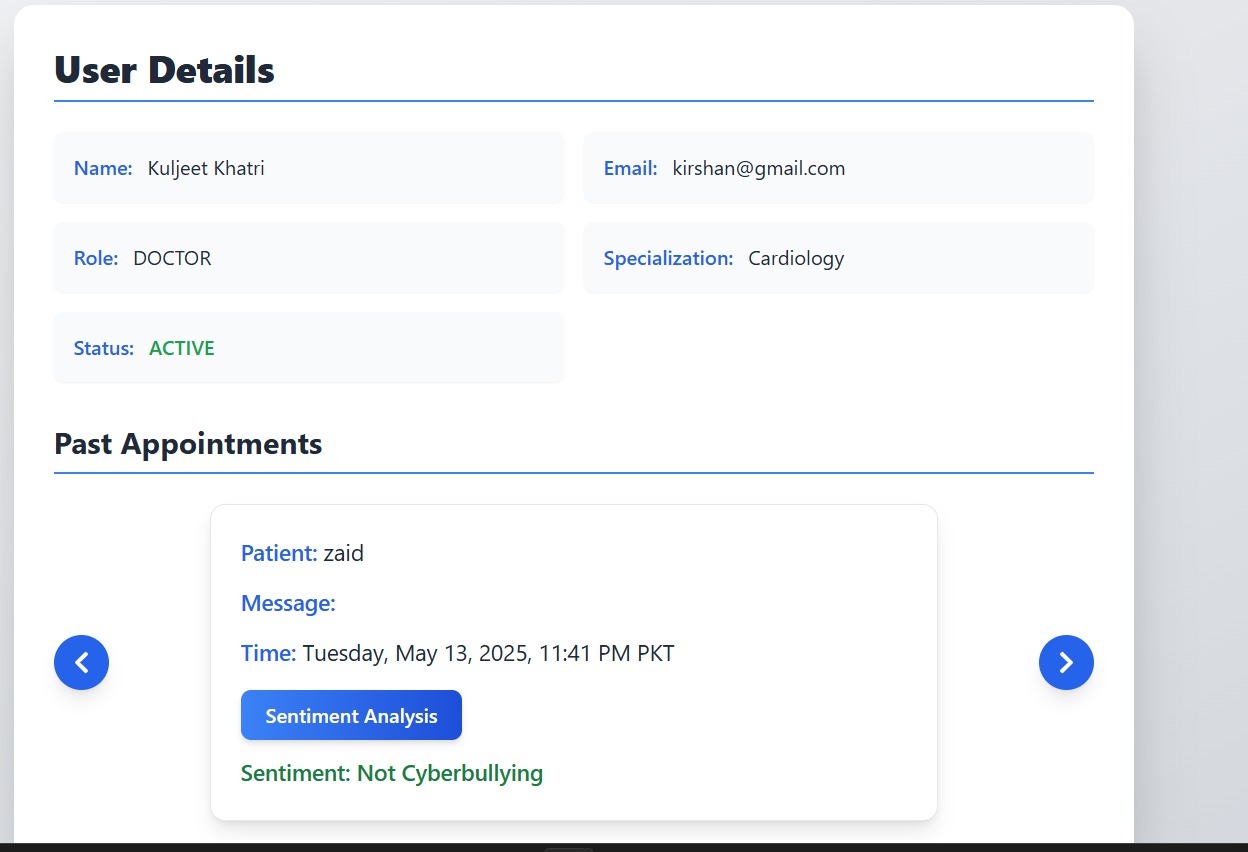
**Pateint Interface**

****

**Doctor Interface**

****

**Admin Interface**

****

**Implementation**

The implementation phase of the CureBridge telehealth platform focused on building a full-stack AI-driven system that enhances remote medical consultations through real-time transcription, intelligent symptom extraction, disease prediction, and sentiment analysis. The system was developed using modern web technologies and advanced machine learning models to ensure real-time performance and high accuracy.

### **Frontend Implementation**

**Technologies Used:**

* ReactJS with TypeScript for modular, strongly typed UI development
* TailwindCSS for responsive and modern styling
* Socket.IO client for real-time communication

**Features Implemented:**

* User interfaces for login, video calls, patient records, and sentiment flag viewing
* Role-based navigation and views for patients, doctors, and admins
* Live display of transcription during video consultations
* Dynamic rendering of symptoms and predicted diseases
* Admin dashboard for managing users and flagged consultations

### **Backend Implementation**

**Technologies Used:**

* Java Spring Boot for secure and scalable REST APIs
* JWT Authentication for token-based login and access control
* Socket.IO (Java server-side) for real-time video call handling
* Room API for WebRTC-based video communication
* MySQL for storing user profiles, consultation records, symptoms, predictions, and sentiment logs

**Key Components:**

* Authentication Module: Handles secure login and generates tokens for session management
* Call Management Module: Creates and manages video call rooms using WebSockets and Room API
* Data Storage: Stores user data, medical records, and conversation logs with proper encryption and indexing

### **Real-Time Transcription Module**

**Technology Used:**

* Deepgram Speech-to-Text API

**Integration Details:**

* The audio stream from the ongoing call is sent to Deepgram in real time
* Transcribed text is returned via Deepgram’s WebSocket API
* The transcript is displayed live in the frontend and also stored in the database for future analysis

### **Symptom Extraction Module**

**Technology Used:**

* Fine-tuned BioBERT (en\_biobert\_ner\_symptom)

**Implementation Flow:**

* The real-time transcript is passed through the BioBERT NER model
* Named entities corresponding to symptoms are extracted
* These symptoms are then sent to the disease prediction module for further processing

### **Disease Prediction Module**

**Technology Used:**

* Stacked Ensemble Machine Learning Model  
  + Base models: Random Forest, Decision Tree, SVM
  + Meta-estimator: Logistic Regression

**Implementation Flow:**

* Input: Symptoms extracted by the BioBERT model
* Output: A list of probable diseases ranked by confidence scores
* The model was trained on medical datasets and evaluated for high precision and recall

### **Sentiment Analysis Module**

**Technology Used:**

* XLNet-based Sentiment Classifier

**Implementation Flow:**

* Transcripts from conversations are passed through the XLNet model
* The model classifies the conversation tone as: Professional, Toxic, or Neutral
* Flagged consultations are automatically logged and made accessible to the admin panel for review

### **Admin Dashboard and Monitoring**

**Features Implemented:**

* Admin can manage doctors and patients (create, update, delete)
* Admin can view all consultations, transcriptions, symptoms, and predictions
* Flagged sentiment analysis results are shown with transcript context
* Role-based access is enforced to maintain system security

### **Integration & Deployment**

* All modules were integrated using REST APIs and WebSockets
* Frontend communicates securely with the backend using JWT tokens
* AI models run as background services or integrated Python microservices
* The application was containerized using Docker for easy deployment

**Testing**

| **Test Case ID** | **Module** | **Test Description** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC001 | User Authentication (Login) | Verify that users can log in using valid credentials | JWT token is generated and user is redirected to dashboard | As Expected | Pass |
| TC002 | User Authentication (Login) | Attempt login with invalid credentials | Display error message and prevent access | As Expected | Pass |
| TC003 | Call Integration | Start a call between the doctor and the patient | Call is successfully established using Room API and Socket.IO | As Expected | Pass |
| TC004 | Call Integration | Test disconnection and reconnection scenarios | User reconnects automatically or gets a retry option | As Expected | Pass |
| TC005 | Real-Time Transcription | Transcribe live conversation using Deepgram | Display real-time text transcription with minimal delay | As Expected | Pass |
| TC006 | Symptom Extraction | Extract symptoms from transcription using BioBERT | Symptoms such as "fever", "headache" are highlighted and stored | As Expected | Pass |
| TC007 | Disease Prediction | Predict disease based on extracted symptoms | Ensemble model returns top diseases with confidence scores | As Expected | Pass |
| TC008 | Sentiment Analysis | Analyze conversation sentiment using XLNet | Admin can view flagged "toxic" or "unprofessional" segments | As Expected | Pass |

**Conclusion**

The successful completion of the CureBridge telehealth platform marks a significant milestone in our journey to develop an AI-powered remote consultation system. Through this project, we have designed and implemented a comprehensive end-to-end solution that integrates real-time transcription, symptom extraction, disease prediction, and sentiment analysis into a single cohesive platform.

Leveraging **Deepgram** for live transcription, a **BioBERT-based NER model** for symptom extraction, a **stacked ensemble machine learning model** for disease prediction, and a **fine-tuned XLNet model** for sentiment analysis, CureBridge provides an intelligent and responsive diagnostic support system. These AI components have been seamlessly integrated into a robust **Java-based backend** with a dynamic and user-friendly **React + TypeScript frontend** interface, ensuring both scalability and usability.

Furthermore, the inclusion of real-time **video calling**, secure record management, and professional-grade UI/UX elevates the platform into a complete telehealth solution. This project demonstrates not only technical competency across machine learning, natural language processing, and web development, but also a clear vision for improving remote healthcare delivery.

CureBridge stands as a testament to the power of interdisciplinary innovation, and we are confident that this platform can serve as a foundation for future research, development, and real-world deployment in the telemedicine domain.

References

A. Haleem, M. Javaid, R.P. Singh, and R. Suman, “Telemedicine for healthcare: Capabilities, features, barriers, and applications,” *ScienceDirect*, Elsevier, India, 2021.

* X Luo, P. Gandhi, S. Storey, and K. Huang, “A Deep Language Model for Symptom Extraction From Clinical Text and its Application to Extract COVID-19 Symptoms From Social Media,” *IEEE*, USA, April 2022.
* D. Guo, M. Li, Y. Yu, Y. Li, G. Duan, F. Wu, and J. Wang, “Disease Inference with Symptom Extraction and Bidirectional Recurrent Neural Network,” *IEEE*, Spain, 2018.
* A. Schiegl, “Disease-Symptom relation extraction from medical text corpora with BERT,” *reposiTUm*, 2021.
* S. Wada, R. Iida, K. Torisawa, T. Takeda, S. Manabe, and Y. Matsumura, “Extracting Symptom Names and Disease-Symptom Relationships from Web Texts Using a Multi-Column Convolutional Neural Network,” *IOS Press*, 2019.
* K.S. Kumar, M.S. Sathya, A. Nadeem, and S. Rajesh, “Diseases Prediction based on Symptoms using Database and GUI,” *IEEE*, India, 2022.
* P. Hema, N. Sunny, R.V. Naganjani, and A. Darbha, “Disease Prediction using Symptoms based on Machine Learning Algorithms,” *IEEE*, India, 2022.
* P. Hamsagayathri and S. Vigneshwaran, “Symptoms-Based Disease Prediction Using Machine Learning Techniques,” *IEEE*, India, 2021.
* A.C.M.V. Srinivas, C. Satyanarayana, C. Divakar, and K.P. Sirisha, “Sentiment Analysis using Neural Network and LSTM,” *IOP Publishing*, India, 2021.
* D. Li and J. Qian, “Text sentiment analysis based on long short-term memory,” *IEEE*, China, 2016.
* P.A. Naidu, K.D.K. Yadav, B. Meena, and Y.V.N. Meesala, “Sentiment Analysis By Using Modified RNN And A Tree LSTM,” *IEEE*, India, 2022.
* Karen Walker, “About The Future Of Telehealth And Technology,” *Forbes*, 2024.

MARKETSANDMARKETS, “Telehealth & Telemedicine Market Size, Share & Trends - Global Forecast to 2028.”