A PRELIMENERY REPORT ON

TELEMEDICINE APPOINTMENT SCHEDULAR

SUBMITTED TO THE VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

BACHELOR OF TECHNOLOGY (COMPUTER ENGINEERING(AIML))

SUBMITTED BY

Sr. NO	Student Name	Roll NO	Email-id
1	Sushant Nadvade	391060	sushant.22210661@viit.ac.in
2	Varun Vangar	391063	varun.22210185@viit.ac.in
3	Shiromani Waghmare	391067	shiromani.22210441@viit.ac.in
4	Vedank Wakalkar	391068	vedank.22210400@viit.ac.in

VISHWAKARMA INSTITUTES

VI

DEPARTMENT OF COMPUTER ENGINEERING(AIML)

BRACT'S
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY

SURVEY NO. 3/4, KONDHWA (BUDRUK), PUNE – 411048, MAHARASHTRA (INDIA).

1.INTRODUCTION

1.1 Overview

The *Telemedicine Appointment Scheduler* is an AI-driven healthcare platform designed to transform the way patients interact with medical professionals. In today's fast-paced world, timely and accessible medical care is essential—especially in areas with limited healthcare infrastructure. This system bridges that gap by enabling secure, efficient, and intelligent virtual consultations.

Through the integration of modern technologies such as WebRTC for real-time video communication, machine learning for personalized doctor recommendations, and cloud infrastructure for scalable deployment, the platform offers a comprehensive solution for remote medical services. Patients can book appointments, consult doctors, receive prescriptions, and manage medical records—all through a single, user-friendly interface.

The system supports a wide range of features including family account management, multilingual access, digital payments, and role-based access control, ensuring that it caters to diverse user needs while maintaining high standards of privacy and data security.

In essence, this project aims to enhance healthcare accessibility, reduce physical barriers to consultation, and offer a reliable, AI-assisted medical experience to users across various demographics.

1.2 Motivation

The increasing demand for accessible and timely healthcare services—especially in remote and underserved regions—has highlighted the limitations of traditional, in-person medical consultations. The COVID-19 pandemic further accelerated the need for robust telemedicine solutions, making virtual healthcare not just a convenience but a necessity.

Patients often face challenges such as long travel distances, waiting times, and difficulty in finding the right specialist. Moreover, many healthcare facilities are overwhelmed with administrative tasks that could be streamlined using digital tools. These inefficiencies not only affect patient satisfaction but also compromise the quality of care delivered.

This project was driven by the desire to create a seamless, intelligent, and secure system that empowers patients and doctors alike. By leveraging advancements in artificial intelligence, cloud computing, and secure video communication, the *Telemedicine Appointment Scheduler* aims to reduce these inefficiencies, provide better access to medical expertise, and ensure continuity of care from the comfort of one's home.

Ultimately, the goal is to create a scalable and inclusive digital health platform that supports modern healthcare delivery and improves patient outcomes across geographies.

1.3 Problem Definition and Objectives

Problem Definition

The conventional healthcare system often struggles to meet the growing needs of patients—particularly in regions with limited access to medical professionals or infrastructure. Issues such as unavailability of doctors, inefficient appointment management, long waiting periods, lack of personalized care, and fragmented communication between patients and healthcare providers continue to hinder the delivery of timely and effective treatment.

Additionally, with the increased adoption of digital tools in healthcare, there remains a significant gap in integrating these technologies into a unified, user-friendly, and secure system that ensures consistent and high-quality virtual consultations.

Objectives

To address these challenges, the *Telemedicine Appointment Scheduler* project was initiated with the following key objectives:

- **Enable seamless appointment scheduling** through real-time calendar views and conflict-free slot allocation.
- **Facilitate secure and encrypted teleconsultations** using WebRTC technology for high-quality video and audio communication.
- **Implement AI-powered doctor recommendation systems** based on patient history, symptoms, preferences, and location.
- **Digitally manage prescriptions** for efficient sharing, viewing, and integration with pharmacies.
- **Ensure scalability, accessibility, and user-friendly interactions** across devices and languages.
- **Provide a secure environment** through encrypted communications, JWT authentication, and role-based access control.
- **Support diverse user groups** including families, multi-profile management, and multilingual patients.
- **Streamline payments and insurance integration** for a smooth end-to-end consultation experience.

1.4 Project Scope & Limitations

Project Scope

The *Telemedicine Appointment Scheduler* project aims to create an end-to-end platform for remote healthcare services, focusing on accessibility, intelligence, and security. The system is designed to serve both patients and healthcare providers through the following scope:

- **Virtual Consultations**: Enables video-based doctor-patient consultations using secure WebRTC protocols.

- **AI-Based Recommendations**: Uses machine learning and similarity search (FAISS) to suggest the most suitable doctors based on user input and medical history.
- **Appointment Management**: Supports scheduling, rescheduling, and cancellation of appointments with real-time doctor availability tracking.
- **Digital Prescription Handling**: Allows doctors to issue prescriptions online and enables patients to access them anytime, with optional pharmacy integration.
- **Multilingual and Multi-Profile Support**: Offers a multi-language interface and allows users to manage multiple family profiles under one account.
- **Chatbot Assistance**: Provides instant support through an AI-powered chatbot for symptom guidance and platform navigation.
- **Cloud Hosting and Scalability**: Hosted on AWS EC2 using Docker containers to ensure high availability, performance, and ease of scaling.
- **Secure Transactions**: Integrates encrypted payment gateways for consultation charges, supporting multiple payment options.

Limitations

While the platform addresses a wide range of healthcare challenges, it has a few limitations:

- **Dependency on Internet Access**: A stable internet connection is essential for smooth video consultations and platform usage.
- **AI Limitations**: AI-based recommendations and chatbots may not always deliver perfect accuracy, especially in complex or rare cases.
- **Regulatory Constraints**: Compliance with health data regulations (such as HIPAA or local laws) may vary by region and require adaptations.
- **Limited Physical Examination**: Certain diagnoses and treatments that require physical interaction are not feasible through virtual consultations.
- **Language Translation Nuances**: Although multi-language support is implemented, real-time translations may still carry semantic or medical inaccuracies in rare cases.

1.5 Methodologies of Problem Solving

The *Telemedicine Appointment Scheduler* leverages a combination of advanced technologies and structured problem-solving approaches to deliver a smart, secure, and user-centric healthcare platform. The development of the system involved the following methodologies:

1.5.1 Machine Learning for Doctor Recommendation

To intelligently match patients with suitable doctors, a machine learning-based recommendation engine is implemented. It utilizes historical consultation data, user preferences, symptoms, doctor specialties, and ratings to suggest the most relevant healthcare providers. The recommendation accuracy improves over time through continuous learning.

1.5.2. Similarity Search with FAISS

FAISS (Facebook AI Similarity Search) is used to perform efficient similarity searches. When a patient provides input symptoms or preferences, FAISS rapidly retrieves doctors with matching profiles from the database, significantly reducing the time to find appropriate matches.

1.5.3. Secure Video Communication via WebRTC

WebRTC (Web Real-Time Communication) enables encrypted, peer-to-peer video consultations between patients and doctors. It supports high-quality, low-latency communication, screen sharing, and cross-platform compatibility for secure and seamless virtual visits.

1.5.4. Secure Authentication and Authorization

To ensure privacy and data protection, the platform uses JSON Web Tokens (JWT) for secure user authentication. Role-Based Access Control (RBAC) is implemented to restrict access to sensitive data and features based on user roles (e.g., patient, doctor, admin).

1.5.5. Modular System Design

A modular architecture is followed to ensure scalability, maintainability, and flexibility. Each module (e.g., booking, consultation, chatbot, prescription, payments) is independently developed and integrated, enabling faster development and easier updates.

1.5.6. DevOps and Cloud Deployment

The entire platform is containerized using Docker and deployed on AWS EC2 instances. This ensures high availability, ease of scaling, and continuous integration/deployment capabilities for future feature updates.

1.5.7. Chatbot with NLP

An AI-powered chatbot is integrated using Botpress, which employs Natural Language Processing (NLP) to assist users with symptom-based guidance, FAQs, and platform navigation.

By combining these methodologies, the system achieves a balance between technical robustness, user experience, and operational scalability—delivering a modern solution to traditional healthcare challenges.

---

2. LITERATURE SURVEY

The concept of telemedicine has gained significant attention in recent years due to advancements in communication technologies, the increasing demand for remote healthcare, and the global push for digital transformation in the medical field. Various research studies and systems have laid the foundation for modern telemedicine platforms. This section highlights key contributions and gaps identified in existing literature.

2.1. Traditional Telemedicine Systems

- Focused primarily on providing video or audio consultations.
- Lacked AI integration, smart scheduling, or real-time doctor availability tracking.
- Systems like early versions of **eVisit** and **Doxy.me** offered basic functionality.
- Limitations included poor scalability, minimal personalization, and no backend intelligence.

2.2. Artificial Intelligence in Healthcare

- Machine learning models have been used for diagnosis prediction, symptom analysis, and treatment suggestions.
- Commonly used models: Decision Trees, Support Vector Machines (SVM), Neural Networks.
- AI applications include chatbots, triage tools, and risk assessments.
- Most AI implementations exist in silos and are not integrated into telehealth platforms.

2.3. Appointment Scheduling Techniques

- Traditional platforms rely on static or rule-based scheduling methods.
- These approaches don't adapt to dynamic availability or patient load.
- AI-driven scheduling can optimize appointments and reduce wait times.
- Real-time rescheduling and slot recommendations are rare in existing systems.

2.4. Secure Video Consultation Tools

- WebRTC is the industry standard for secure, real-time video communication.
- Platforms such as **Doxy.me** and **Amwell** use WebRTC but offer limited features.
- Many systems lack document sharing, multi-language support, and session recording options.
- Not all platforms implement full end-to-end encryption.

2.5. Existing Commercial Telemedicine Platforms

- Platforms like **Practo**, **Teladoc**, **MDLIVE**, and **1mg** dominate the market.
- Offer features like booking, consultation, and digital prescriptions.
- Most are subscription-based with limited customization or open-source availability.
- Accessibility in rural and low-bandwidth regions remains a challenge.

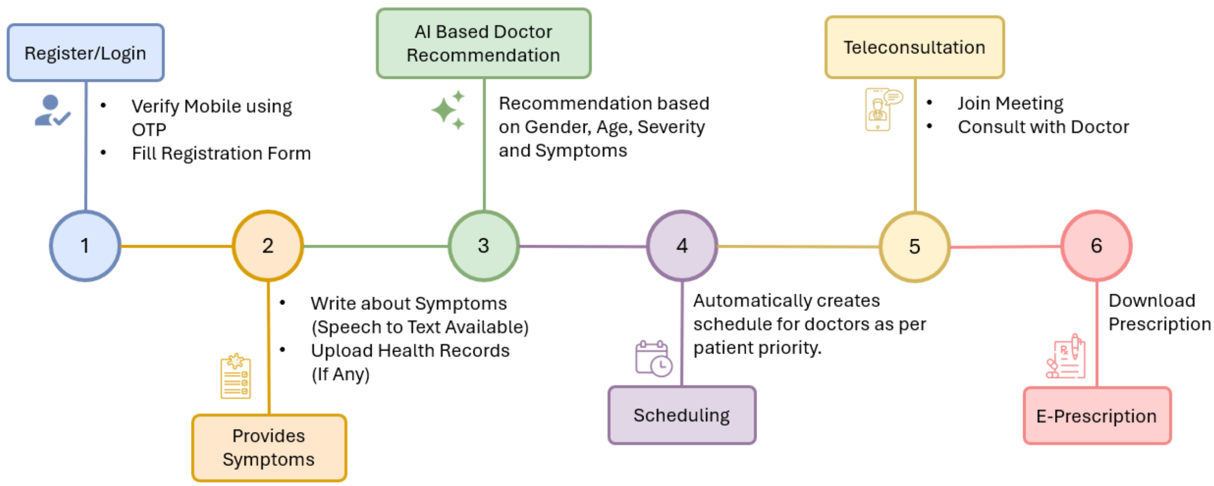
2.6. Identified Gaps

- Poor AI integration for personalized doctor recommendations.
- Lack of multilingual and family management features.
- Inadequate support for real-time scheduling adjustments.
- Few platforms offer scalable, modular, cloud-hosted architectures.

---

3. SYSTEM DESIGN

3.1 System Architecture



4. PROJECT IMPLEMENTATION

4.1 Overview of Project Modules

The telemedicine platform can be broken down into several key modules, each handling a specific aspect of the user journey and system functionality:

4.1.1 User Authentication & Registration Module

- **Purpose**: Manages user onboarding and secure access to the platform.
- **Functionality**:
  - User registration and login with mobile verification using OTP.
  - Form filling for user details (e.g., name, age, gender).
- **Technologies**:
  - **Frontend**: React Native (UI), Redux (state management), Axios (API calls).
  - **Backend**: Node.js, Express.js (server-side logic), JWT (authentication).
  - **Database**: MongoDB (storing user profiles).

4.1.2 Symptom Input & Health Record Management Module

- **Purpose**: Collects user symptoms and health records for doctor recommendations and consultations.
- **Functionality**:
  - Users input symptoms via text or speech (speech-to-text functionality).
  - Option to upload health records for additional context.
- **Technologies**:
  - **Frontend**: React Native (UI for input forms), Redux (state management).
  - **AI/ML**: Google Speech to Text (converting speech to text), NLTK (natural language processing of symptoms).

    - **Storage**: Cloudinary (storing uploaded health records).
    - **Backend**: Node.js, Express.js (handling data submission).

4.1.3 AI-Based Doctor Recommendation Module

- **Purpose**: Analyzes user data to recommend an appropriate doctor.
- **Functionality**:
    - Uses AI to evaluate symptoms, gender, age, and severity.
    - Recommends a doctor based on the analysis.
- **Technologies**:
    - **AI/ML**: TensorFlow (machine learning model for recommendations), FastAPI (serving the AI model), NLTK (symptom analysis).
    - **Backend**: Node.js, Express.js (integrating AI recommendations with the system).
    - **Database**: MongoDB (storing doctor profiles and recommendation data).

4.1.4 Scheduling Module

- **Purpose**: Automates the scheduling of doctor consultations.
- **Functionality**:
    - Creates a consultation schedule based on patient priority and doctor availability.
- **Technologies**:
    - **Backend**: Node.js, Express.js (scheduling logic).
    - **Database**: MongoDB (storing schedules and availability data).

4.1.5 Teleconsultation Module

- **Purpose**: Facilitates real-time virtual consultations between users and doctors.
- **Functionality**:
    - Users join a virtual meeting to consult with the doctor.
    - Supports real-time video and audio communication.
- **Technologies**:
    - **Real-Time Communication**: React Native WebRTC, Kurento (for video/audio streaming).
    - **Frontend**: React Native (UI for joining meetings).
    - **Backend**: Node.js, Express.js (managing consultation sessions).

4.1.6 E-Prescription Module

- **Purpose**: Provides users with electronic prescriptions post-consultation.
- **Functionality**:
    - Doctors generate e-prescriptions during or after the consultation.
    - Users can download the prescription.
- **Technologies**:
    - **Backend**: Node.js, Express.js (prescription generation and delivery).
    - **Storage**: Cloudinary (storing e-prescriptions).
    - **Frontend**: React Native (UI for downloading prescriptions).
    - **Database**: MongoDB (storing prescription records).

4.1.7 Multilingual Support Module (Optional)

- **Purpose**: Ensures accessibility for users speaking different languages.
- **Functionality**:
    - Translates user input (symptoms, communication) and doctor responses if needed.
- **Technologies**:
    - **AI/ML**: Google Translation API (language translation).

4.2 Tools and Technology Used

4.2.1 Frontend

- **React Native**: For building the mobile app interface (e.g., registration, symptom input, teleconsultation UI).
- **Redux**: For state management within the app (e.g., managing user data, symptoms, schedules).
- **Axios**: For making API requests to the backend (e.g., submitting symptoms, fetching recommendations).

4.2.2 AI/ML

- **TensorFlow**: For building and running the AI model for doctor recommendations based on gender, age, severity, and symptoms.
- **FastAPI**: For creating a high-performance API to serve the AI model and handle requests.
- **NLTK (Natural Language Toolkit)**: For processing and analyzing user-entered symptoms (natural language processing).
- **Google Speech to Text**: For converting user speech to text when entering symptoms.
- **Google Translation API**: For enabling multilingual support by translating user input or doctor communication.

4.2.3 Backend & Authentication

- **Node.js**: For server-side logic and handling requests across all modules (e.g., registration, scheduling, e-prescription).
- **Express.js**: A Node.js framework for building RESTful APIs and managing backend routes.
- **JWT (JSON Web Token)**: For securing user authentication and session management during login/registration.

4.2.4 Real-Time Communication

- **React Native WebRTC**: For enabling real-time video and audio streaming in teleconsultation (used on the mobile app).
- **Kurento**: A WebRTC media server for managing video/audio streams during virtual consultations.

4.2.5 Database & Cloud Storage

- **MongoDB**: For storing user data, symptoms, health records, schedules, prescriptions, and doctor profiles.
- **Cloudinary**: For managing the storage and retrieval of uploaded health records and e-prescriptions.

4.3 Algorithm Details

4.3.1 Algorithm 1

- Calculate Priority Score using: ***Priority Score = (f1×w1)+(f2×w2)+(f3×w3)+(f4×w4)***

(where F1=Urgency, F2=Demographics, F3=History, F4=Time of Booking)

- Optimize scheduling to prioritize urgent requests for simultaneous appointment bookings.

4.3.2 Algorithm 2

- AI Doctor Recommendation leverages symptoms, medical history, language preference, and time zones, using Random Forest and Logistic Regression to recommend top N doctors based on specialty, language compatibility, and availability.

5. Results

5.1 Outcomes

5.1.1 Improved Access to Healthcare

- Users can consult with doctors remotely, eliminating geographical barriers.
- The platform supports multilingual communication (via Google Translation API), making healthcare accessible to diverse linguistic groups.

5.1.2 AI-Driven Efficiency in Doctor Matching

- The AI-based recommendation system (powered by TensorFlow, NLTK) matches patients with doctors based on gender, age, severity, and symptoms, ensuring more relevant and effective consultations.
- Reduces the time patients spend searching for the right specialist.

5.1.3 Streamlined User Experience

- A seamless user journey from registration to e-prescription, facilitated by an intuitive mobile app interface (React Native, Redux).
- Features like speech-to-text (Google Speech to Text) for symptom input and OTP-based mobile verification simplify user interaction.

5.1.4 Enhanced Scheduling and Prioritization

- Automated scheduling (Node.js, Express.js, MongoDB) prioritizes patients based on the severity of their symptoms, optimizing doctor availability and reducing wait times.

5.1.5 Secure and Real-Time Teleconsultations

- Real-time video/audio consultations (React Native WebRTC, Kurento) provide a reliable and high-quality interaction between patients and doctors.
- Secure user authentication (JWT) ensures privacy and data protection during consultations.

5.1.6 Efficient Health Record and Prescription Management

- Users can upload health records and download e-prescriptions (Cloudinary), creating a centralized and accessible medical history.
- Doctors can generate and share e-prescriptions instantly, improving post-consultation care.

5.1.7 Scalability and Reliability

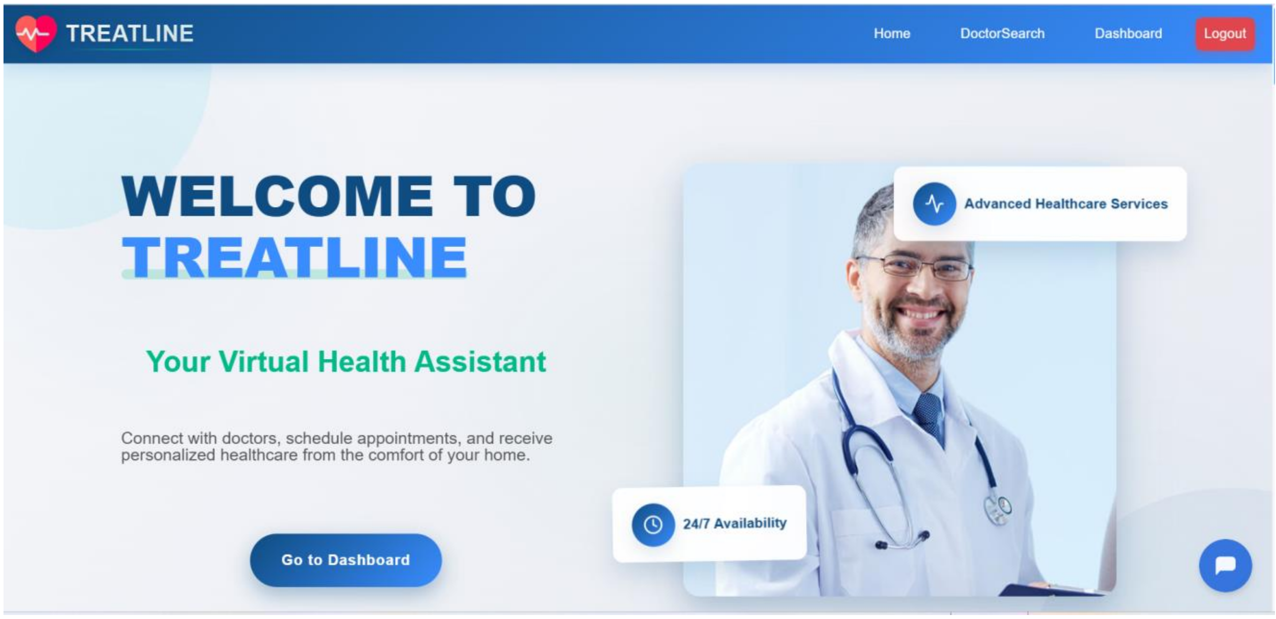
- The use of modern technologies like Node.js, Express.js, and MongoDB ensures the platform can handle a growing number of users and data.
- FastAPI enables quick and efficient AI model serving, maintaining performance even with high user demand.

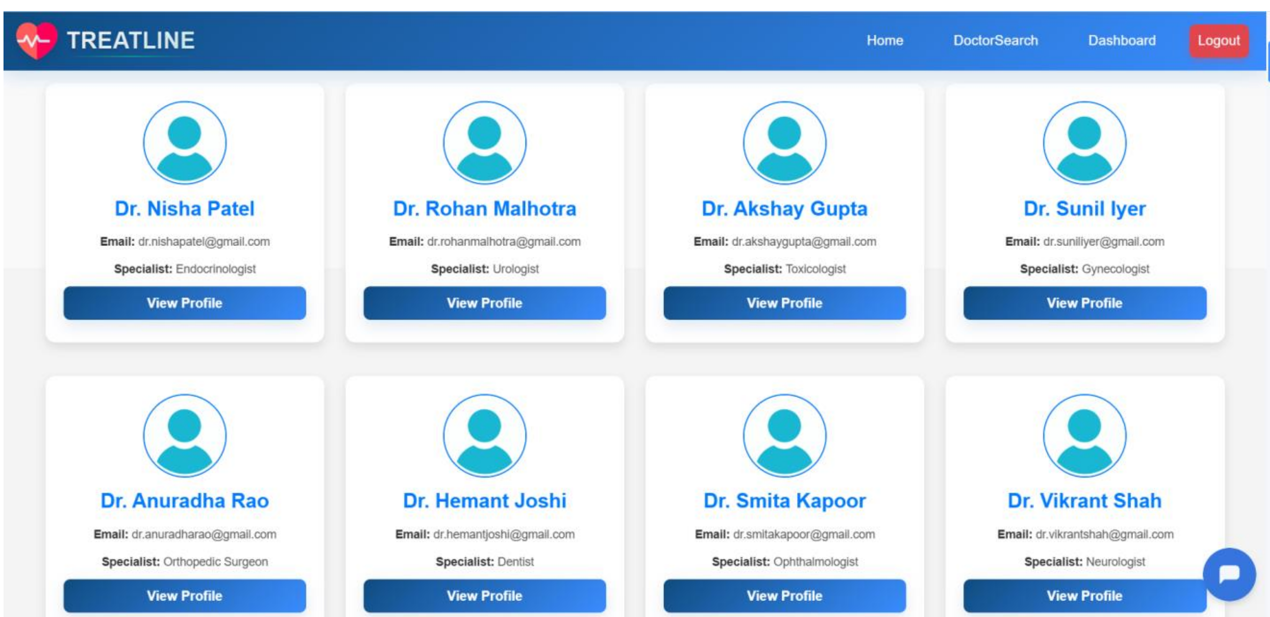
5.1.8 Cost and Time Savings

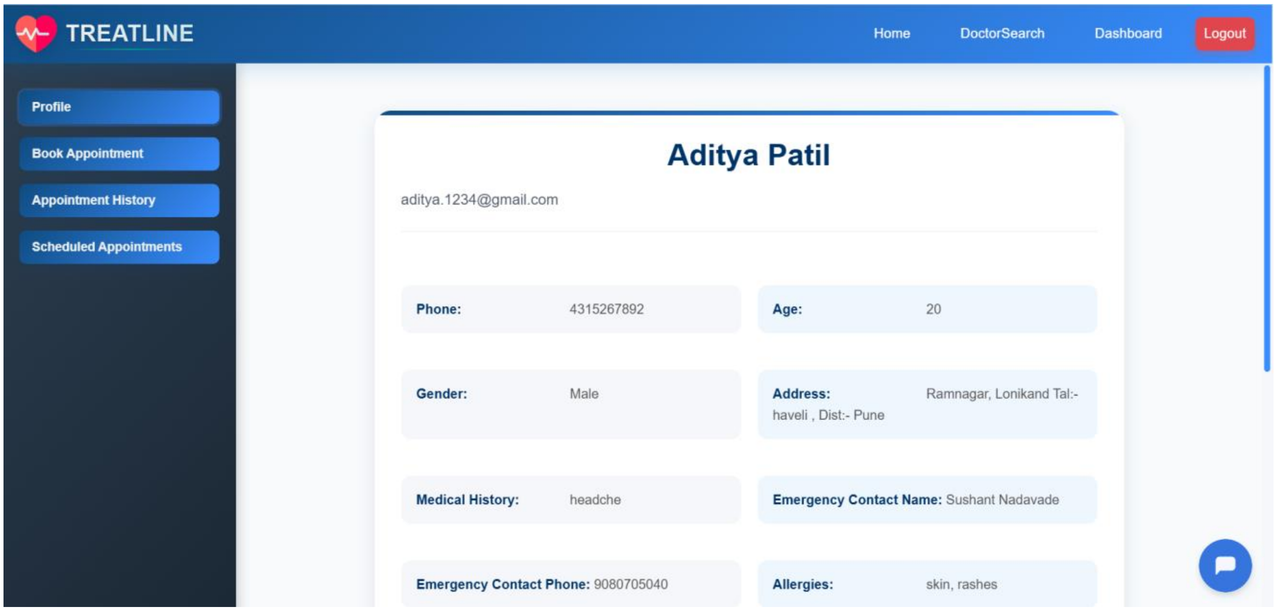
- Patients save on travel costs and time by consulting doctors remotely.
- Doctors can manage more consultations efficiently through automated scheduling and virtual meetings.

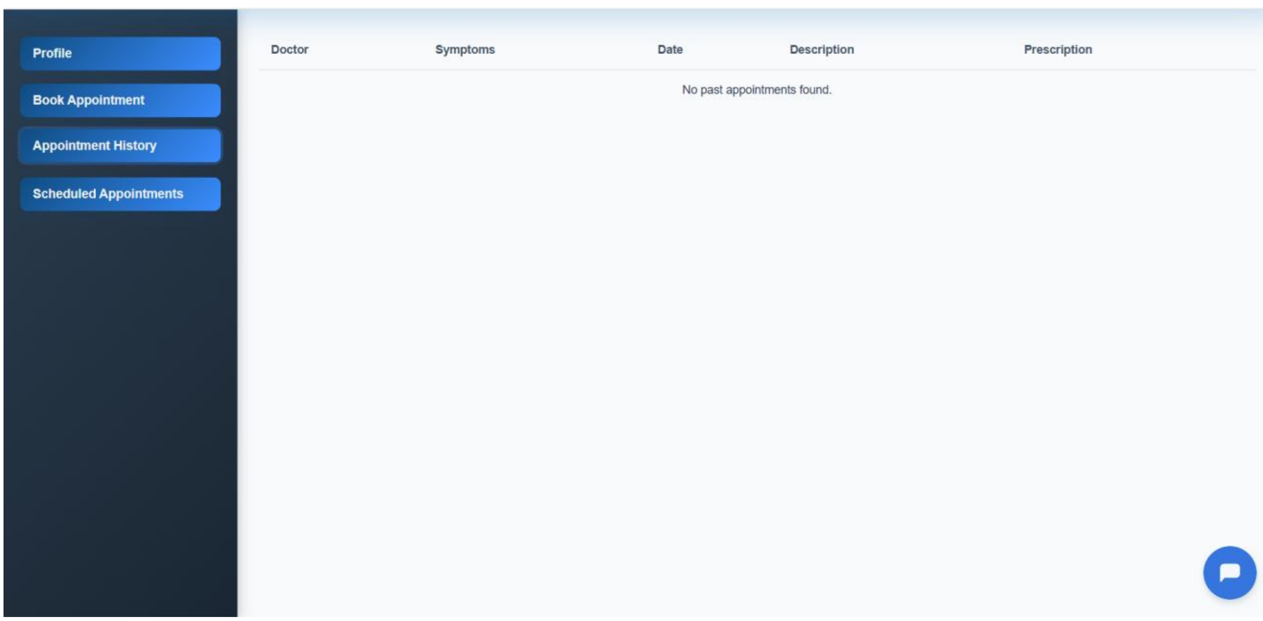
5.2 Screenshots

5.2.1 Web App Screenshots:

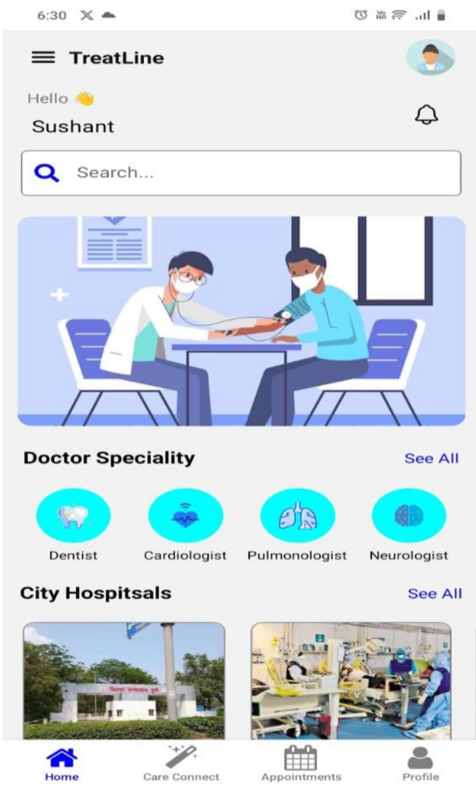


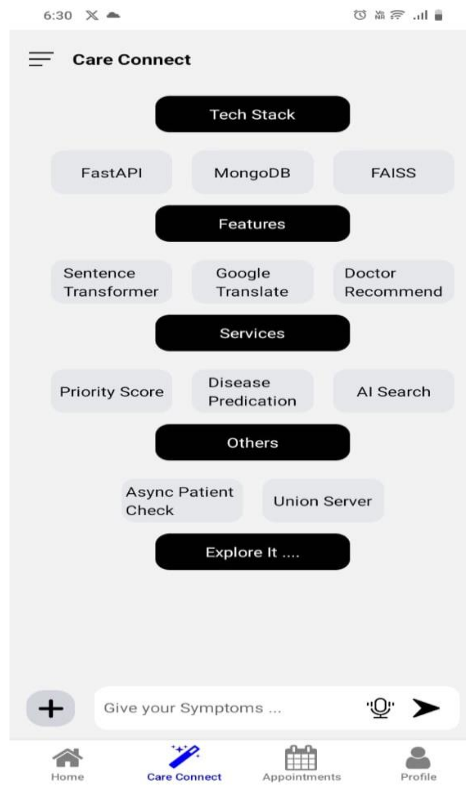


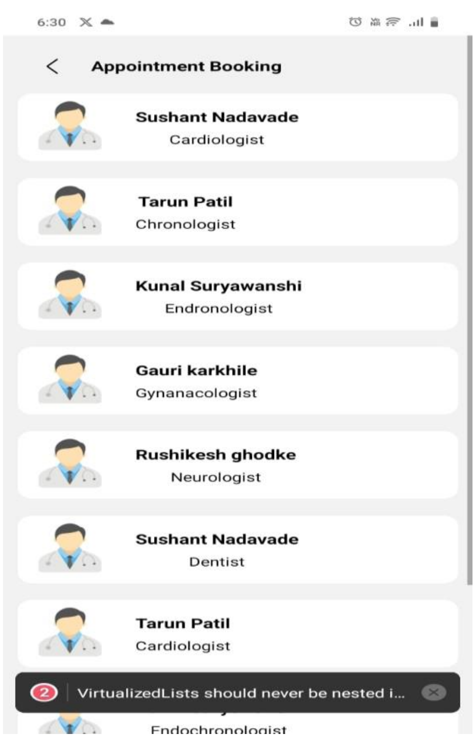

TREATLINE
Home
DoctorSearch
Dashboard
Logout
Profile
Book Appointment
Appointment History
Scheduled Appointments
Aditya Patil
aditya.1234@gmail.com
Phone: 4315267892
Age: 20
Gender: Male
Address: Ramnagar, Lonikand Tal:- haveli , Dist:- Pune
Medical History: headche
Emergency Contact Name: Sushant Nadavade
Emergency Contact Phone: 9080705040
Allergies: skin, rashes

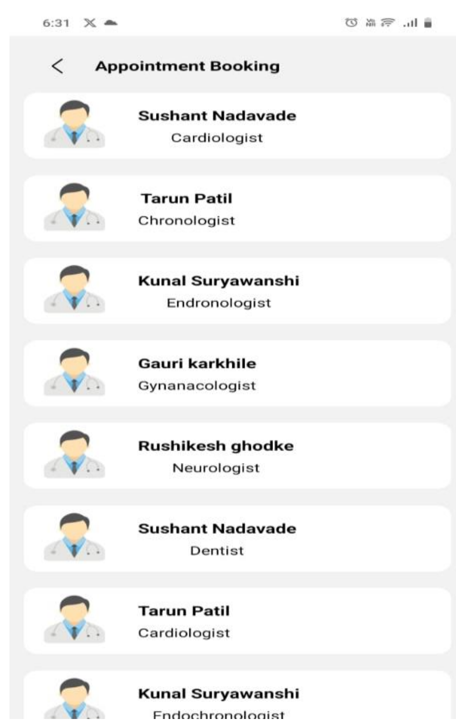

Profile
Book Appointment
Appointment History
Scheduled Appointments
Doctor
Symptoms
Date
Description
Prescription
No past appointments found.

5.2.1 Mobile App Screenshots:









6. CONCLUSIONS

6.1 CONCLUSIONS

The *Telemedicine Appointment Scheduler* project successfully addresses the need for efficient, accessible, and intelligent remote healthcare services. By integrating machine learning, secure video communication, real-time scheduling, and multilingual support, the system offers a modern solution that enhances the overall patient and doctor experience.

The modular design, cloud-based deployment, and AI-powered features provide a scalable foundation that can evolve with the growing demands of the healthcare sector. Security and privacy have been ensured through encrypted communications, authentication protocols, and role-based access controls.

The system not only reduces consultation delays but also brings convenience to patients by offering seamless digital experiences—from appointment booking to receiving prescriptions—at their fingertips.

6.2 FUTURE WORK

To further enhance the platform's capabilities and reach, the following advancements are proposed:

Blockchain Integration: For secure and tamper-proof medical record storage and sharing.

Voice-Based Assistant: Implementing voice commands and accessibility tools for elderly or differently-abled users.

Offline Mode Support: Allowing appointment bookings and record access in low-connectivity areas.

Advanced AI Symptom Checker: Incorporating NLP and large-scale symptom-disease mappings for pre-consultation self-assessments.

Integration with Wearable Devices: For real-time health monitoring and remote diagnostics.

Global Language Support: Expanding multilingual features to include regional dialects and voice-to-text translations.

6.3 APPLICATIONS

The system can be deployed in various healthcare-related environments, including:

Hospitals and Clinics: To digitize patient flow and reduce administrative workload.

Rural and Remote Areas: Providing access to specialists and quality healthcare where physical infrastructure is limited.

Corporate Health Portals: For employee health management and wellness consultations.

Elderly Care Facilities: Offering routine checkups and doctor access for senior citizens with mobility constraints.

School and University Health Systems: To support student health services and mental health counseling.