MEDICO
Tele-Consulting Platform
(Audio/Video/Chat)

Healthcare Application Development

Indain Institute of Information Technology, Allahabad

under guidance of

Prof. Dr. Ranjana Vyas

TEAM 1:

- Vaibhaw Kori (MHC2023008)
- Akshit Kumar (MML2023002)

Contents

Project Demo & GitHub Links

Project Demo Link:

- There are 5 videos in total.
- Link: Click Here

GitHub Links:

- Medico Frontend Server: Click Here
- Medico Backend Server: Click Here

Problem Statement

1. Patients can “call” platform - serviced by one of available doctors
2. Call and conversation can be supported by either chat or audio/video (WebRTC) channel. Should simulate incoming/outgoing calls, as well as waiting/busy conditions
3. Online consultation, recording of clinical details, issue of prescription
   - Patient can upload clinical records (history)
   - Mechanism for doctor to see prior history and add details, and register follow-ups
   - Mechanism for doctors to view callers and follow-ups
   - Prescriptions only for video calls or where identity of caller can be verified
4. Handling repeat callers
5. Consent for sharing record with other hospitals
6. Ability for senior doctor to monitor activities across channels
7. *Audio calls can also be done from regular phones (non-WebRTC - no app needed). No appointments needed for this
8. *Chatbot as a possible initial interaction for chats. Can help patients navigate the app

Project Scope

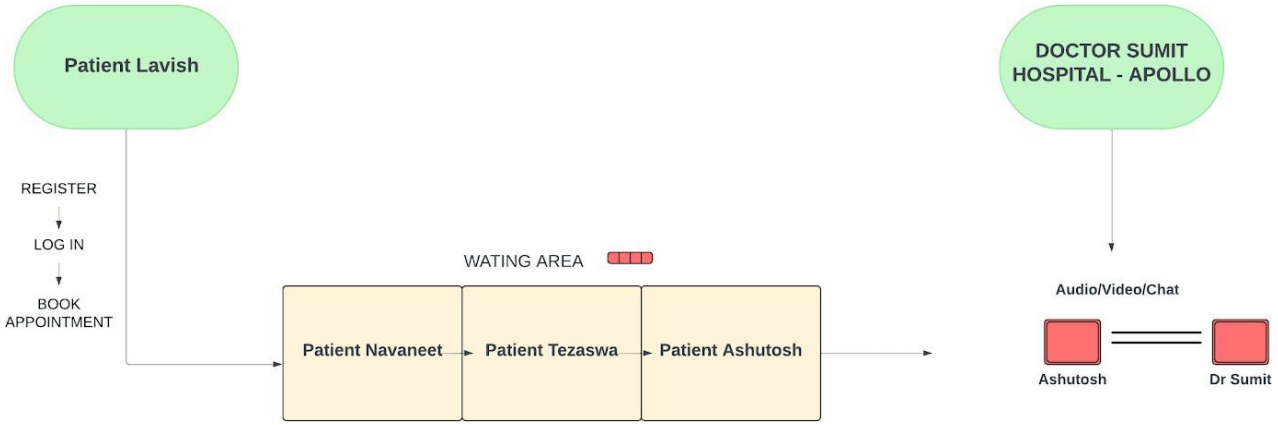
Objective:

- Develop and implement a comprehensive Tele-Consulting Platform to facilitate remote medical consultations with multiple doctors.
- Enhance patient engagement and accessibility by providing support for both audio/video and chat channels.
- Ensure secure and efficient online consultations, including clinical record management, prescription issuance, and follow-up scheduling.
- Implement mechanisms to facilitate seamless continuity of care.
- Establish a consent mechanism for patients to authorize the sharing of their medical records with other healthcare facilities.
- Enable real-time monitoring of platform activities by senior doctors to ensure quality control and compliance.
- Integrate a chatbot to improve user experience and guide patients through the platform.

Deliverables:

- Fully Functional Tele-Consulting Platform: A user-friendly platform capable of supporting audio/video and chat channels seamlessly.
- Interactive Call Simulation: Simulated scenarios for incoming/outgoing calls, waiting/busy conditions, and smooth channel transitions.
- Comprehensive Online Consultation Module: Incorporation of features for clinical record upload, history viewing, prescription issuance, and follow-up registration.
- Repeat Caller Management System: Tools and protocols for managing repeat callers and ensuring uninterrupted care delivery.
- Consent Mechanism for Record Sharing: Secure mechanism enabling patients to grant consent for sharing medical records with authorized healthcare providers.
- Real-time Monitoring Dashboard: Dashboard interface for senior doctors to monitor platform activities and ensure adherence to standards.
- Accessible Audio Calls: Integration of audio call functionality accessible via regular phones, eliminating the need for additional applications or appointments.
- User friendly chatbot integrated

High-Level Overview:



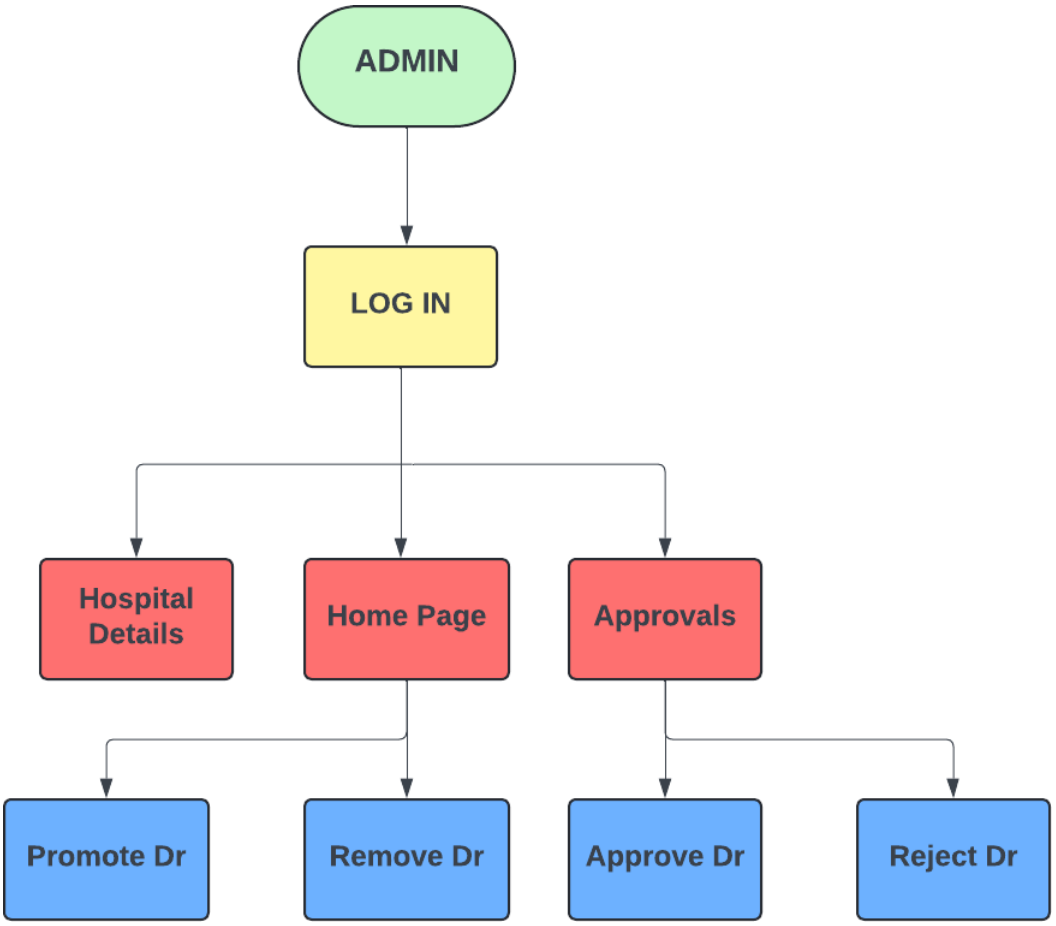
- Patient Lavish seems to be a patient management system or module. It has features such as Registration, Log in, and Appointment, which suggests that patients can register, log in to the system, and book appointments.

- The Doctor Sumit in Hospital Apollo component likely represents the hospital's management system or module. Patient Lavish , Patient Navaneet , Patient Tezaswa and Patient Ashutosh are waiting to connect to the doctor in the “Waiting Area”.

- It includes an "Audio/Video-Chat" feature and "Patient Ashutosh" and "Dr Sumit" are connected in the call where they can video chat or audio chat and even they can only chat as well

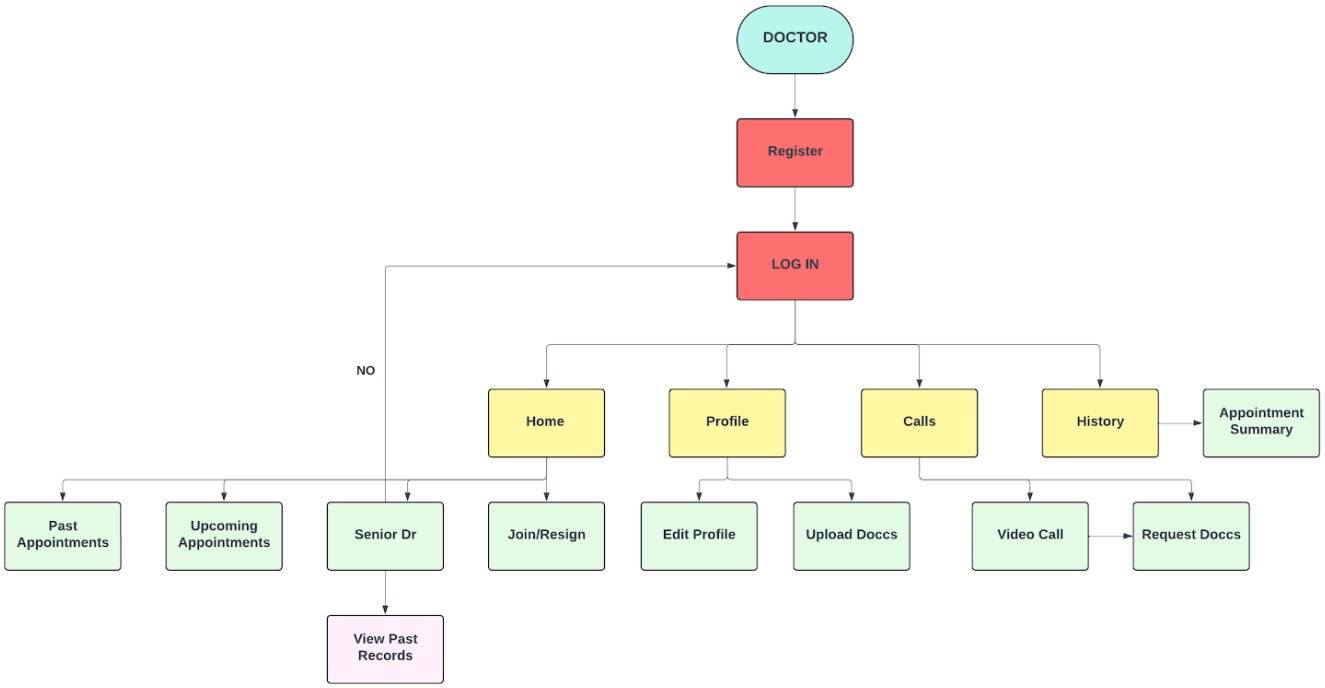
Application Flow

Admin Flowchart:



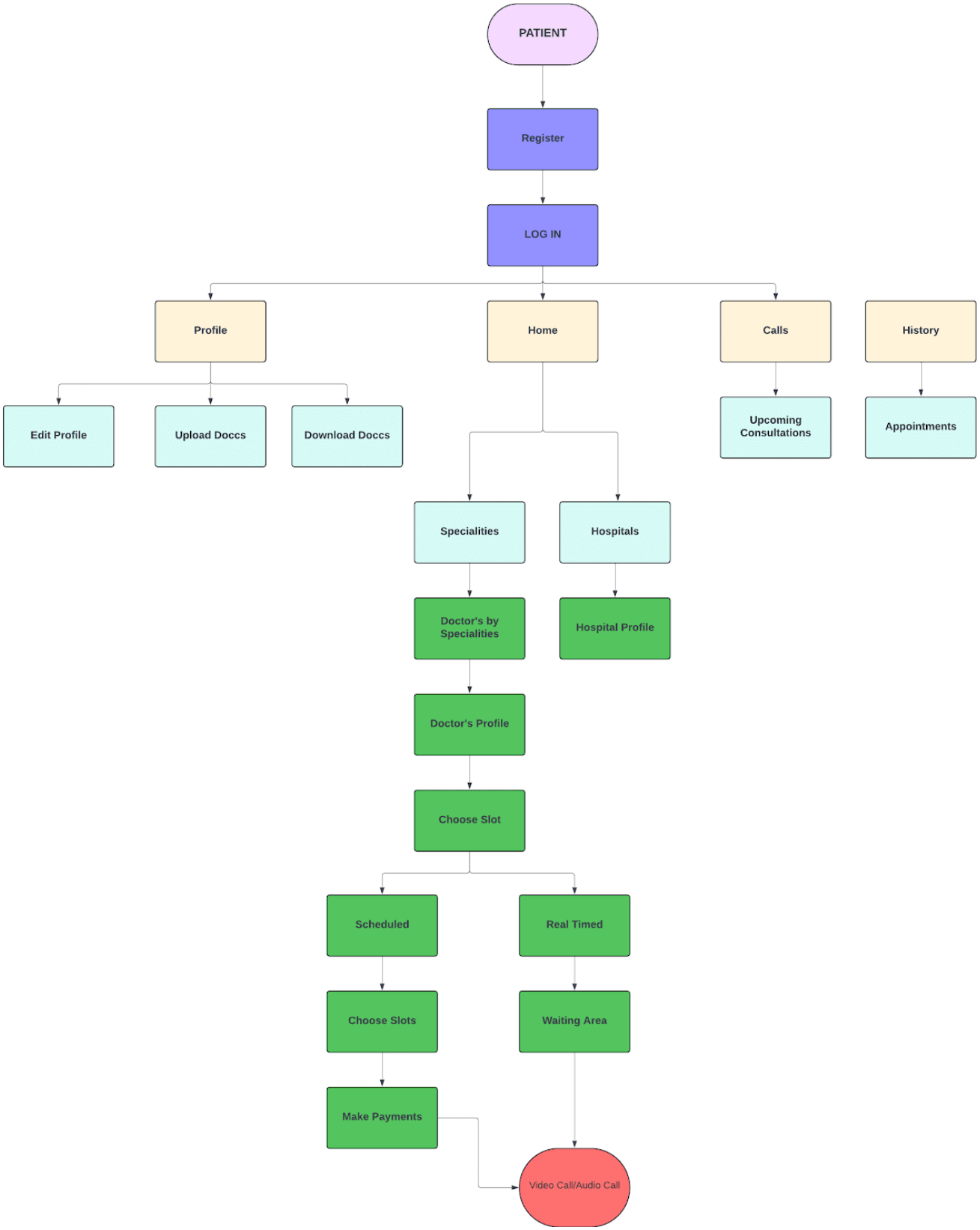
- **Login:** The admin needs to log in to access the system.
- **Hospital Details:** The admin can access and manage details related to hospitals.
- **Home Page:** There is a home page section that the admin can navigate to.
- **Approvals:** The admin has a section dedicated to handling approvals, likely related to doctors or medical professionals.
- **Promote Doctor:** The admin has the ability to promote or grant a higher position to doctors.
- **Remove Doctor:** The admin can remove or revoke the privileges of doctors from the system.
- **Approve Doctor**: The admin can approve or grant access to doctors, possibly new hires or applicants.
- **Reject Doctor:** The admin has the authority to reject or deny doctors, likely based on certain criteria or requirements

Doctor Flowchart:



- **Registration:** A doctor needs to register to gain access to the system.
- **Login:** After registration, the doctor can log in to access their account and features.
- **Home:** There is a home section or landing page for the doctor's interface.
- **Profile:** The doctor can view and manage their profile details within the system.
- **Calls:** There is a dedicated section for handling video calls or consultations with patients.
- **History:** The doctor can access and view the medical history or records of patients.
- **Appointment Booking:** Doctors can book appointments or schedule consultations with patients.
- **Upcoming Appointments**: Doctors can view a list or schedule of their upcoming appointments.
- **Senior Dr:** There is an option for doctors to consult with or seek guidance from senior or more experienced doctors.
- **Join Design:** Doctors can potentially join or participate in the design or development of certain aspects of the system
- **Edit Profile:** Doctors can edit and update their profile information as needed.
- **Upload Docs:** Doctors can upload relevant medical documents or files to the system.
- **Video Call:** Doctors can initiate or conduct video calls, likely for consultations or follow-ups with patients.
- **Request Doctors:** Doctors can request or seek assistance from other doctors within the system.
- **View Past Records**: Doctors can view and access past medical records or case histories of patients.

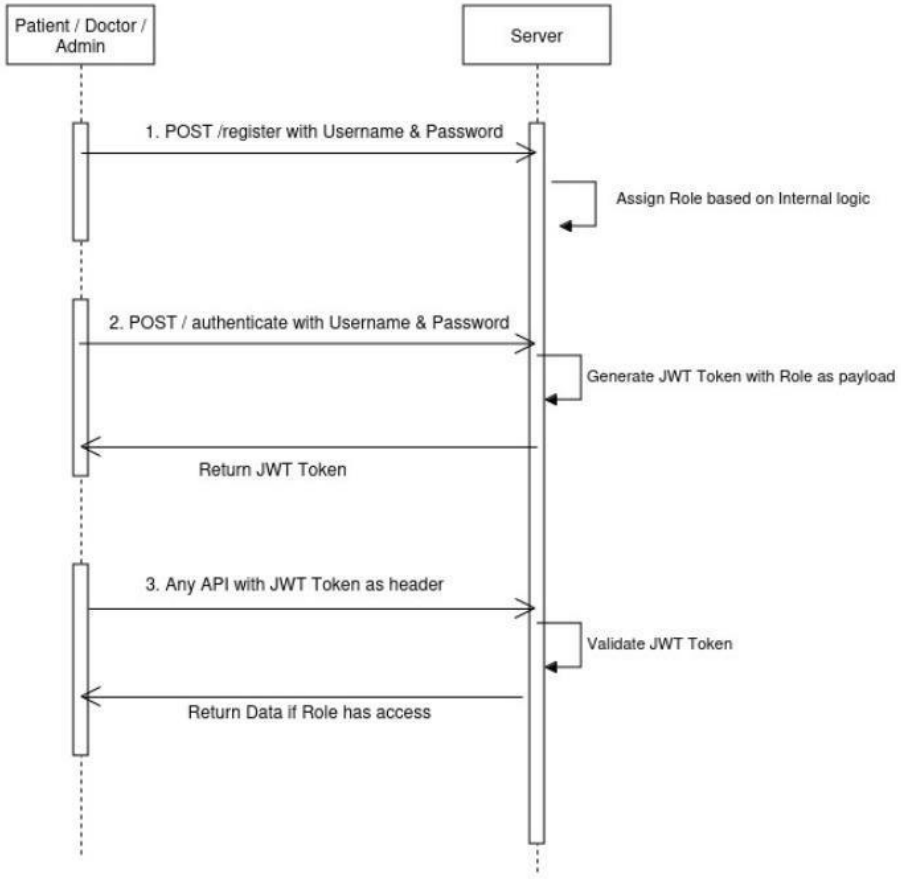
Patient Flowchart:



- **Patient:** The start point is the "Patient" node, indicating that the flow is centered around the patient.
- **Register:** The first step is for the patient to register in the system.
- **Login:** After registration, the patient can log in to access various features and functionalities.
- **Profile:** Patients can view and manage their profile details within the system.
- **Home:** There is a dedicated home section or landing page for patients.
- **Calls:** Patients have access to a feature for making calls, likely for consultations or appointments.
- **History:** Patients can view their medical history or records within the system.
- **Specialities:** Patients can explore or choose from various medical specialties or departments.
- **Hospitals:** Patients can access information or choose from different hospitals.
- **Request Dr:** Patients can request or book appointments with specific doctors.
- **Request Facility:** Patients can request access to specific medical facilities or services.
- **Update Profile:** Patients can update or edit their profile information as needed.
- **Book Lab:** Patients can book or schedule laboratory tests or procedures.
- **Add Card:** Patients can add or manage payment card information within the system.
- **View Bills:** Patients can view and access their medical bills or invoices.
- **Pay Online:** Patients have the option to make online payments for their medical bills or services.
- **View Transcripts:** Patients can view and access transcripts, potentially related to their medical reports or records.
- **Exit:** The final node represents the patient exiting or completing their journey within the healthcare system.

Sequence Diagram

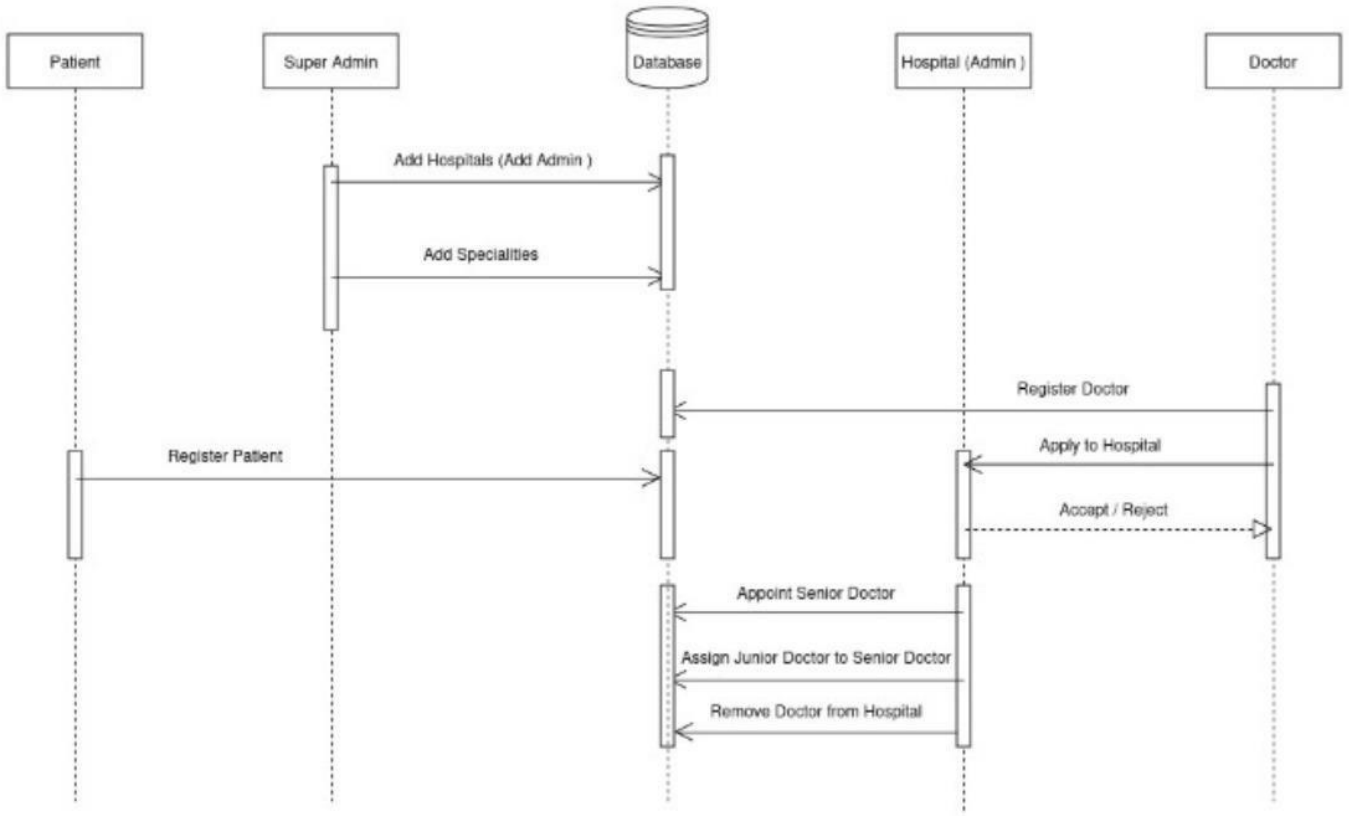
User Authentication



1. The client initiates the process by sending a POST request with their username and password to register on the server.
2. Once registered, the client sends another POST request with their username and password to authenticate themselves with the server.
3. The server receives the authentication request and performs an internal logic to assign a specific role to the client based on their credentials. This role determines the client's access permissions role to the client based on their credentials. This role determines the client's access permissions within the application.
4. After assigning the role, the server generates a JSON Web Token (JWT) with the assigned role encoded as the payload.
5. The server responds to the client by returning the generated JWT token.
6. For any subsequent API requests, the client includes the received JWT token in the request header as an authentication and authorization mechanism.
7. When the server receives an API request with the JWT token, it validates the token to ensure its authenticity and integrity.
8. If the JWT token is valid, the server checks if the client's assigned role (encoded in the token) has the necessary access permissions for the requested data or resource.
9. If the client's role has the required access permissions, the server returns the requested data or performs the requested operation.

This sequence diagram illustrates a common authentication and authorization approach using JWT tokens in modern web applications. The JWT token acts as a secure and self-contained mechanism to transmit the client's identity and assigned role between the client and server. By validating the JWT token and checking the associated role permissions, the server can enforce access control and ensure that clients can only access resources they are authorized to access based on their assigned role.

Admin (Hospital)



Super Admin:

- The Super Admin can add new hospitals to the system by adding hospital admins.
- The Super Admin can also add specialties or areas of expertise for doctors.

Hospital (Admin):

- The Hospital Admin can register new doctors into the system.
- Doctors can apply to the hospital, and the Hospital Admin can accept or reject their applications.
- The Hospital Admin can appoint senior doctors.
- The Hospital Admin can assign junior doctors to work under the supervision of senior doctors.
- The Hospital Admin also has the authority to remove doctors from the hospital.

Doctor:

- Doctors can apply to hospitals in the system.
- Upon application, the Hospital Admin can either accept or reject their application.

Patient:

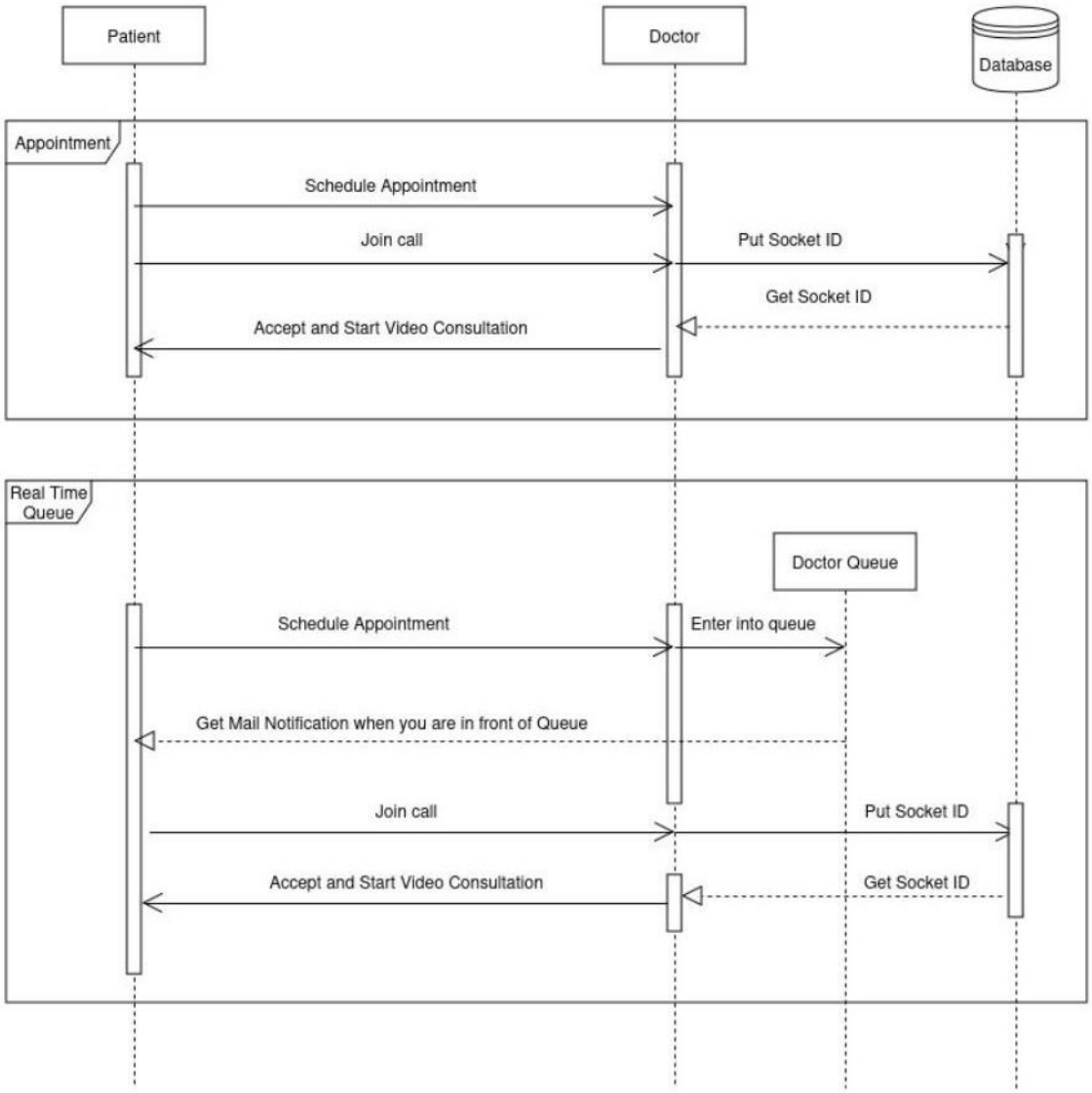
- Patients can register themselves in the healthcare management system.

Database:

- The Database component indicates that the system stores and manages data related to hospitals, doctors, patients, and other relevant information.

The use case diagram illustrates the various actors, their roles, and the key interactions or use cases within the healthcare management system. It provides a high-level overview of the functionalities and relationships between the different entities involved, without delving into the specific sequences or detailed flows.

Appointments



- Patient schedules an appointment.
- The appointment is added to the Real Time Queue.
- When it's the patient's turn, they receive a mail notification to join the call.
- The patient joins the call.
- On the doctor's side, the schedule appointment request is received.

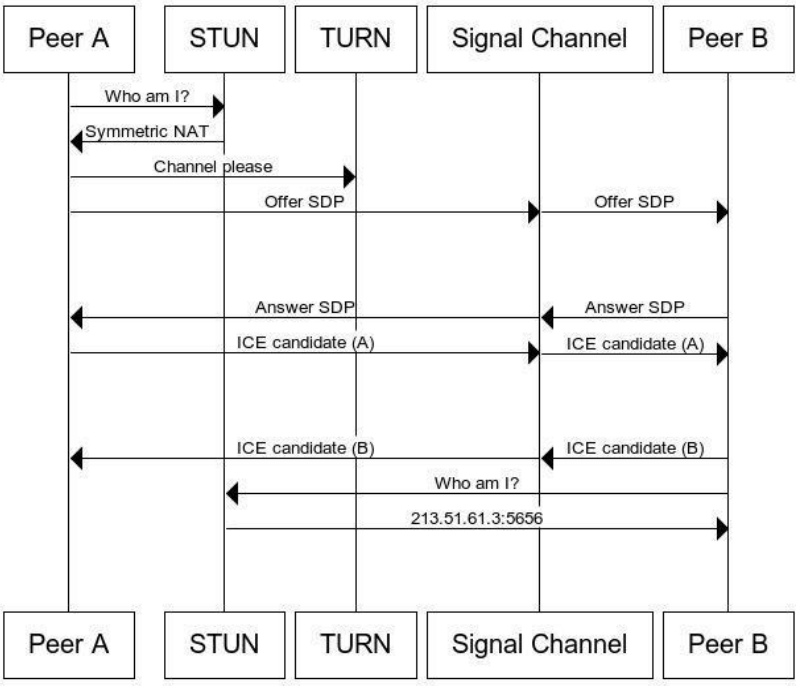
- The doctor gets the Socket ID from the database to connect with the patient.
- The doctor puts the Socket ID to accept and start the video consultation with the patient.

Additionally, there's an alternative path where the doctor is directly available:

- The patient schedules an appointment.
- The doctor receives the schedule appointment request.
- The doctor gets the Socket ID from the database.
- The doctor puts the Socket ID to join the call.
- The patient accepts and starts the video consultation.

The workflow involves components like the Patient, Doctor, Database for storing Socket IDs, Real Time Queue for managing appointment requests, and a Doctor Queue for handling multiple appointment requests simultaneously. Overall, it facilitates the scheduling and execution of video consultations between patients and doctors in an organized manner.

Video Call



- Peer A initiates the call by sending a "Who am I?" message to the STUN server. This is done to discover its public IP address and port mapping.

- The STUN server responds with Peer A's public IP address and port mapping (213.51.61.3:5656 in the example).

- Peer A sends a "Channel please" message to the TURN server, requesting a relay address and port for the media session.

- The TURN server responds with an "Offer SDP" (Session Description Protocol) message, which contains the relay address and port information.

- Peer A sends the "Offer SDP" message to Peer B, proposing the media session details.

- Peer B receives the "Offer SDP" and responds with an "Answer SDP" message, accepting the proposed media session details.

- Peer A and Peer B exchange ICE (Interactive Connectivity Establishment) candidates. These candidates are potential IP addresses and ports that can be used for the media session.

- Peer A and Peer B perform connectivity checks using the exchanged ICE candidates to determine the best path for the media session.

- Once a valid media path is established, the video call can start, and media data is exchanged directly between Peer A and Peer B using the negotiated IP addresses and ports.

- The STUN server is used to discover the public IP address and port mapping of the peers, which is necessary for establishing a direct peer-to-peer connection. The TURN server acts as a relay server, providing a fallback option if a direct peer-to-peer connection cannot be established due to network constraints or firewalls. The signaling channel is used to exchange control messages and negotiate the media session parameters.

- This workflow ensures that the video call can be established even in the presence of Network Address Translation (NAT) devices or firewalls, by using the STUN and TURN servers to facilitate the connection setup and relay media data when necessary.

API Documentation

AuthController

🔑 Authentication

Method	Endpoint	Description
POST	/api/auth/registerAdmin	Register a new admin
POST	/api/auth/loginAdmin	Login as an admin
POST	/api/auth/registerDoctor	Register a new doctor
POST	/api/auth/loginDoctor	Login as a doctor
POST	/api/auth/registerPatient	Register a new patient
POST	/api/auth/loginPatient	Login as a patent

AdminController

Method	Endpoint	Description
GET	/api/admin/getDoctorsOfHospital/{adminId}	Get doctors of a hospital
DELETE	/api/admin/removeDoctorFromHospital/{adminId}/{docId}	Remove a doctor from hospital
POST	/api/admin/acceptOrRejectDoctor	Accept or reject a doctor
POST	/api/admin/assignJrDoctorsToSrDoctor	Assign junior doctors to senior doctor
GET	/api/admin/getAppliedDoctorsList/{adminId}	Get list of applied doctors
GET	/api/admin/appointSrDoctor/{docId}	Appoint a senior doctor

DoctorController

Method	Endpoint	Description
GET	/api/doctor/getDoctorDetails/{doctorId}	Get details of a doctor
POST	/api/doctor/editDoctorDetails	Edit details of a doctor
GET	/api/doctor/getAllConsultationOfDoc/{docId}	Get all consultations of a doctor
GET	/api/doctor/getPendingConsultationsOfDoc/{docId}	Get pending consultations of a doctor
POST	/api/doctor/putSocketOfDoctor	Update socket of a doctor
GET	/api/doctor/resignFromHospital/{doctorId}	Resign from a hospital as a doctor
GET	/api/doctor/applyToHospital/{doctorId}/{hospitalId}	Apply to a hospital as a doctor
GET	/api/doctor/getSocketOfNextPatient/{doctorId}	Get socket of the next patient
GET	/api/doctor/deleteQueueOfDoctor/{doctorId}	Delete queue of a doctor
GET	/api/doctor/getJrDoctorsOfSrDoctor/{srDoctorId}	Get junior doctors of a senior doctor
POST	/api/doctor/uploadDoctorFiles/{docId}	Upload files for a doctor
GET	/api/doctor/downloadDoctorFiles/{docId}	Download files of a doctor

HomeController:

Method	Endpoint	Description
GET	/api/home/allSpecialities	Get all specialities
GET	/api/home/docBySpeciality/{specialityId}	Get doctors by speciality

PatientController:

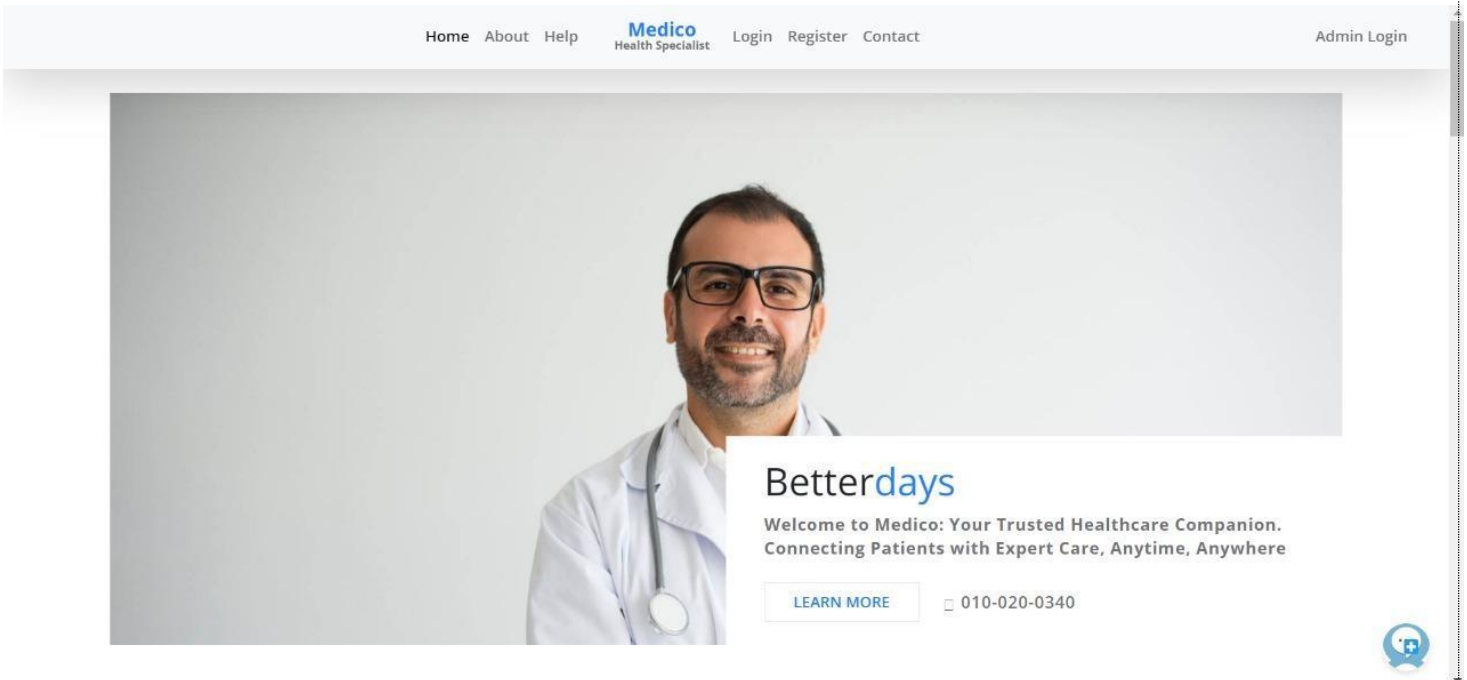
Method	Endpoint	Description
POST	/api/patient/getDoctorSlots	Get available slots for a doctor
POST	/api/patient/bookConsultation	Book a consultation with a doctor
GET	/api/patient/getAllHospitals	Get all hospitals
GET	/api/patient/getAllSpecialityByHospital/{hospitalId}	Get all specialities by hospital
GET	/api/patient/getDocBySpecialityandHospital/{specialityId}/{hospitalId}	Get doctors by speciality and hospital
GET	/api/patient/getPatientDetails/{patientId}	Get details of a patient
POST	/api/patient/editPatientDetails	Edit details of a patient
GET	/api/patient/getAllConsultationsOfPat/{patientId}	Get all consultations of a patient
POST	/api/patient/setRating	Set rating for a consultation
GET	/api/patient/docBySpeciality/sortedR/{specialityId}	Get doctors by speciality sorted by rating
GET	/api/patient/docBySpeciality/sortedP/{specialityId}	Get doctors by speciality sorted by popularity
GET	/api/patient/getSocketOfDoctor/{doctorId}	Get socket of a doctor
POST	/api/patient/enterIntoQueue	Enter into queue for a doctor
POST	/api/patient/getWaitingList	Get waiting list for a patient
POST	/api/patient/uploadPatientsFiles/{patientId}/{placeholder}	Upload files for a patient
GET	/api/patient/downloadPatientFiles/{patientId}	Download files of a patient
GET	/api/patient/downloadOnePatientFile/{filename}	Download one file of a patient
GET	/api/patient/files/{patientId}	Get files of a patient

SuperAdmin-Controller:

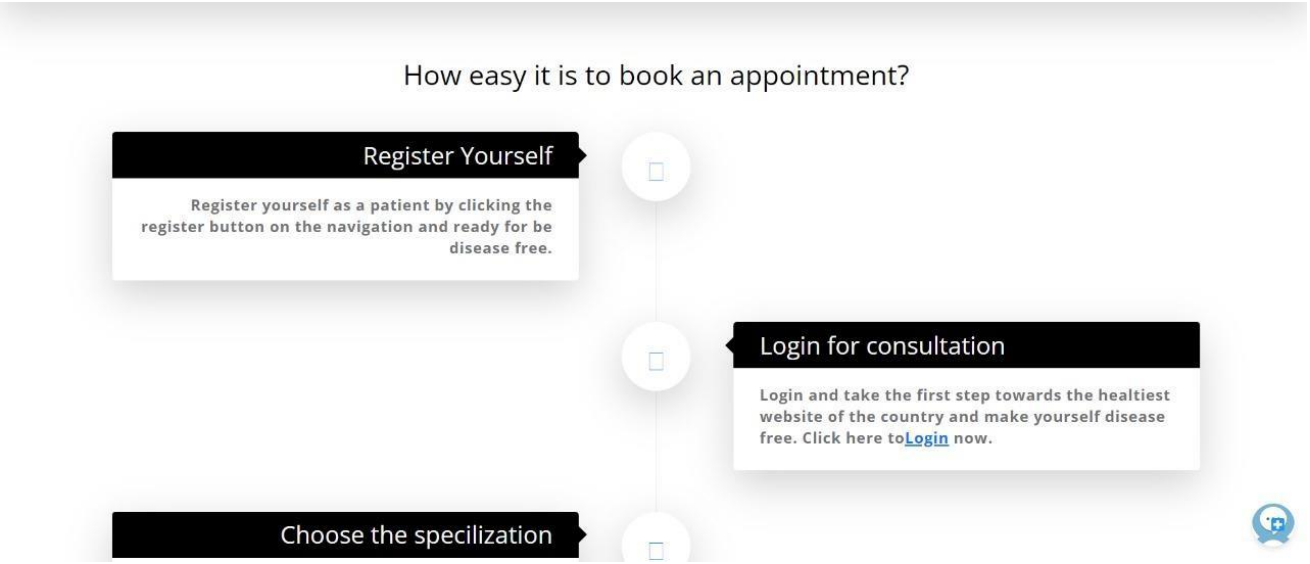
Method	Endpoint	Description
POST	/api/superAdmin/addHospital	Add a hospital
POST	/api/superAdmin/addSpeciality	Add a speciality
POST	/api/superAdmin/addHospitals	Add multiple hospitals
POST	/api/superAdmin/addSpecialities	Add multiple specialities

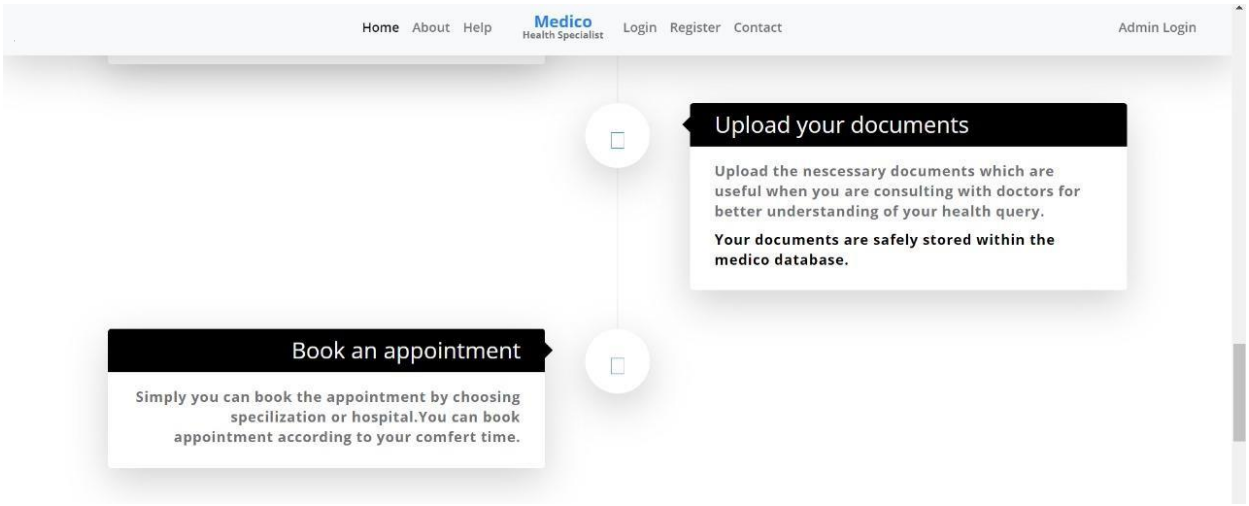
User Interface

General Landing Page

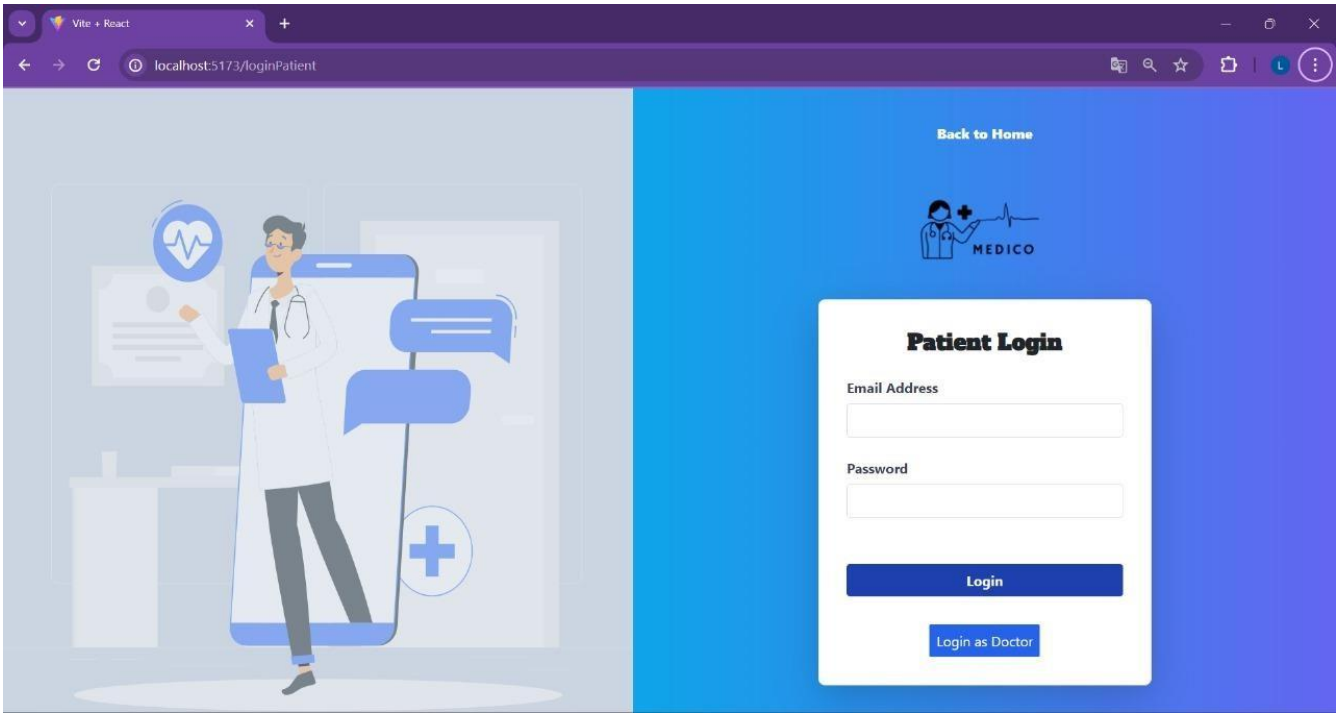


Navigation

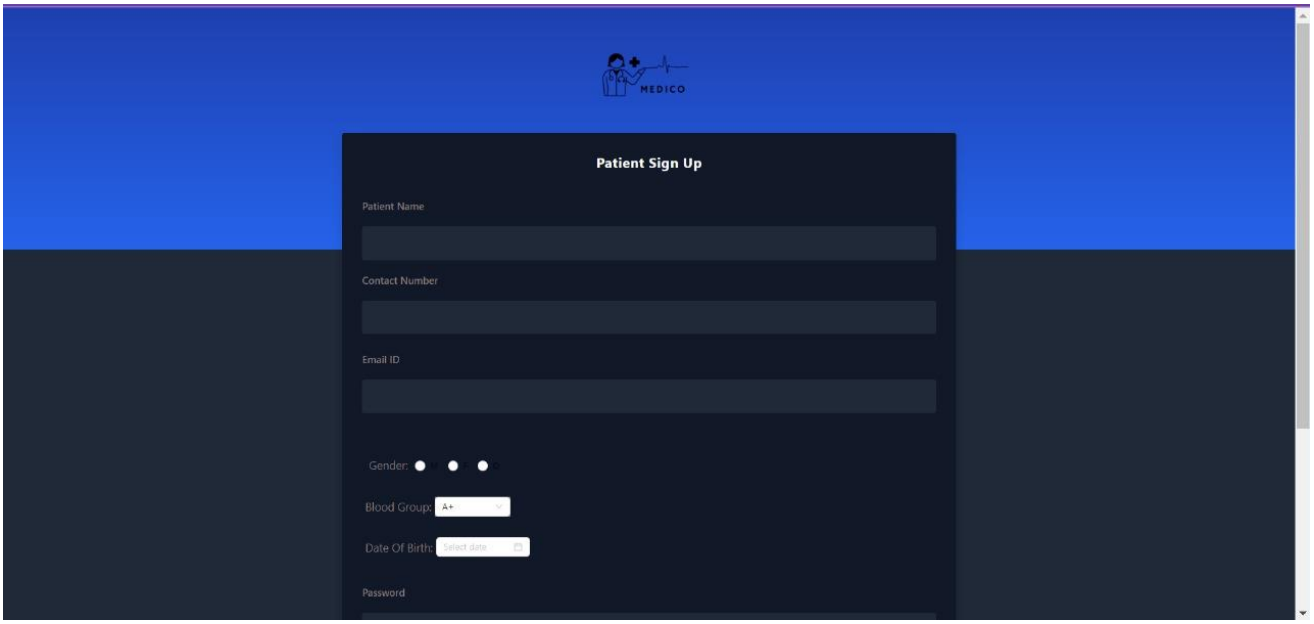



Home About Help
Medico
Health Specialist
Login Register Contact
Admin Login
Upload your documents
Upload the nescessary documents which are useful when you are consulting with doctors for better understanding of your health query.
Your documents are safely stored within the medico database.
Book an appointment
Simply you can book the appointment by choosing specilization or hospital.You can book appointment according to your comfert time.

Login

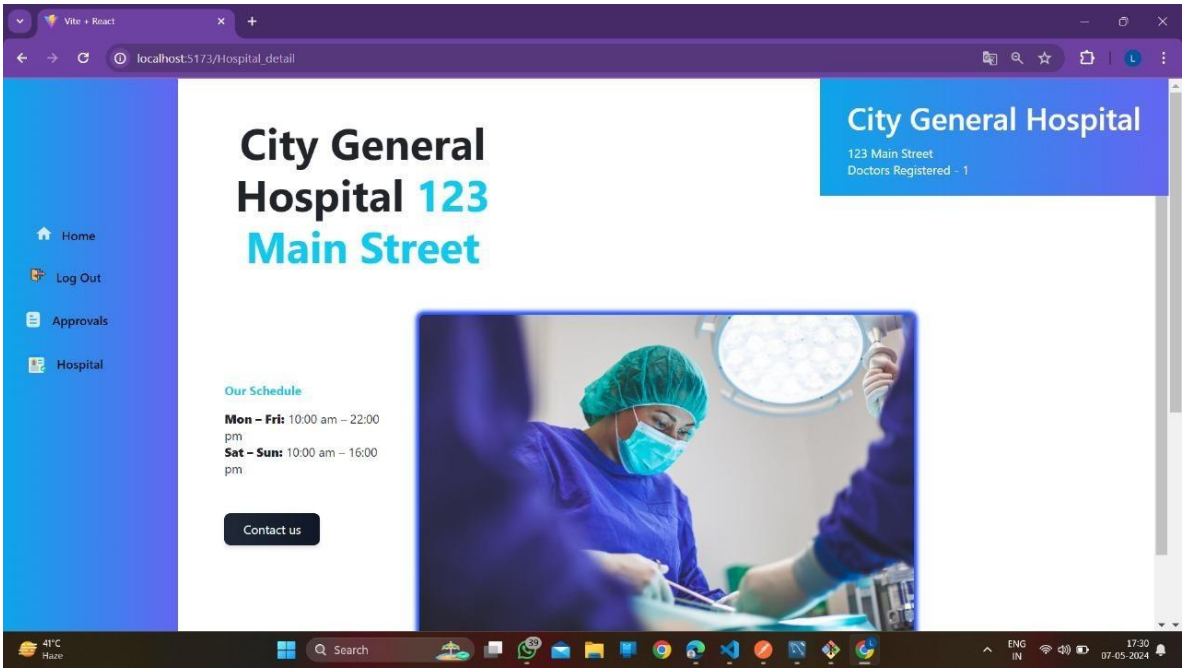

Vite + React
localhost:5173/loginPatient
Back to Home
MEDICO
Patient Login
Email Address
Password
Login
Login as Doctor

Register

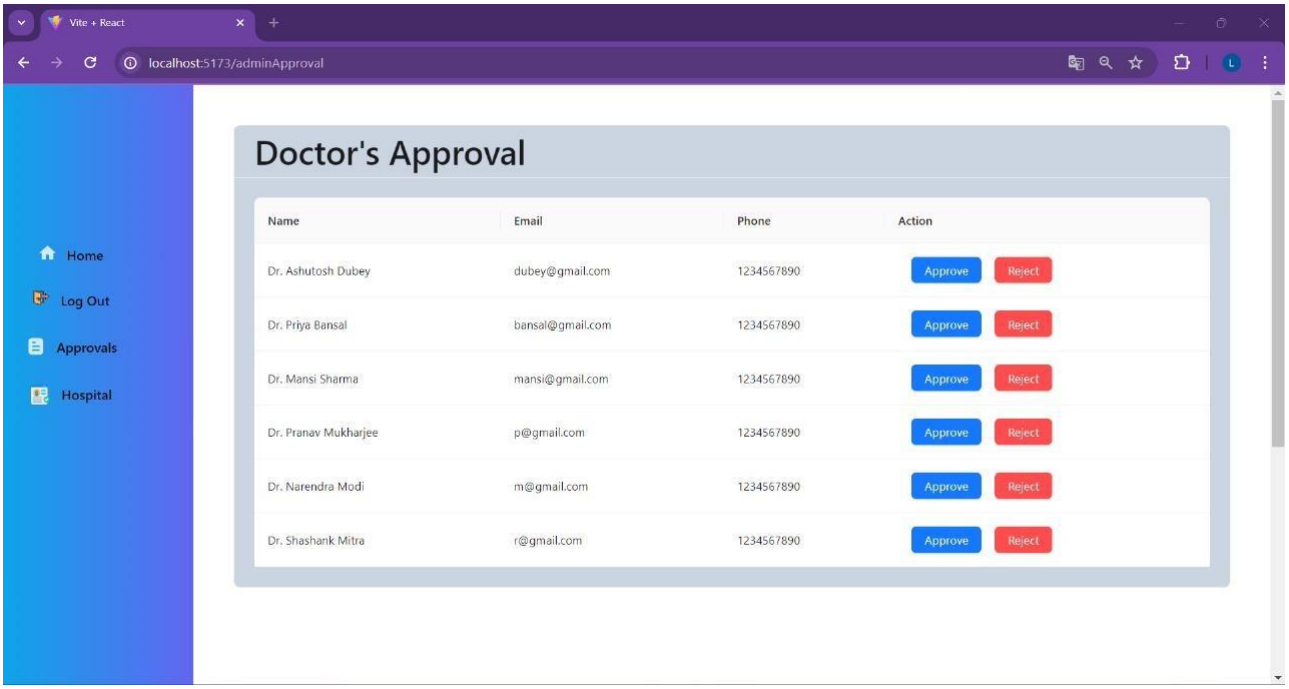


Admin:

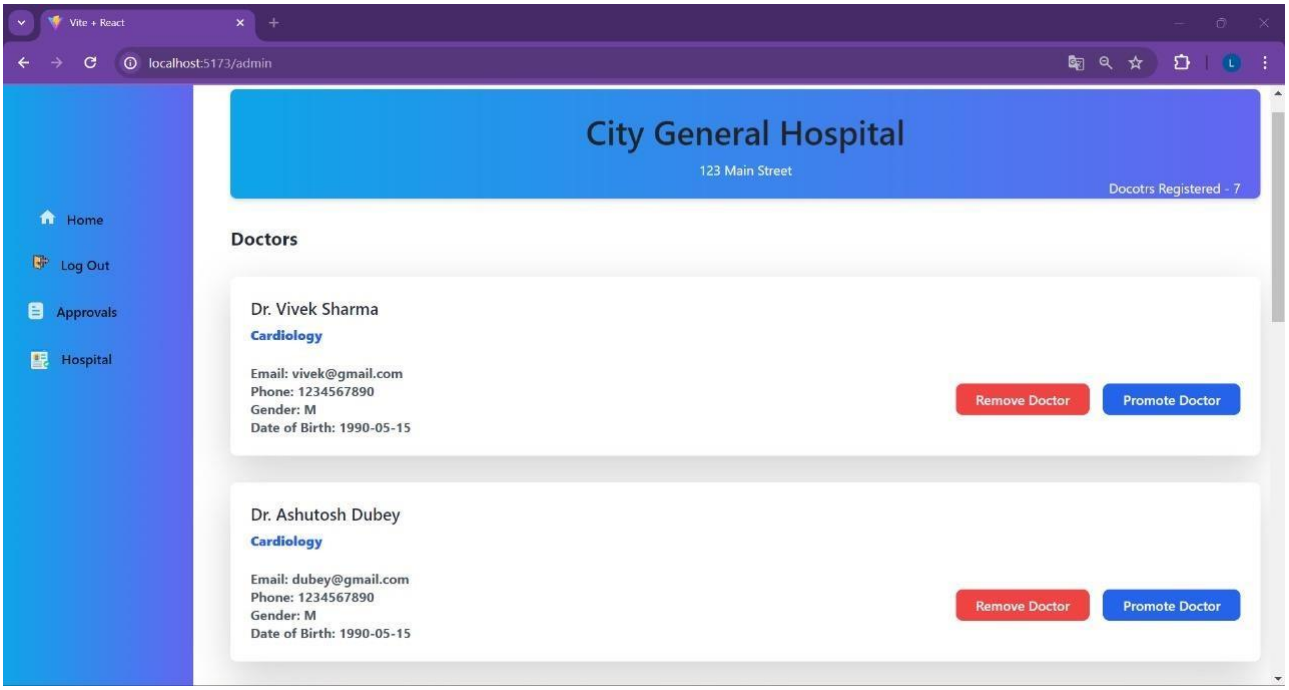
Hospital Profile



Doctor’s Approval

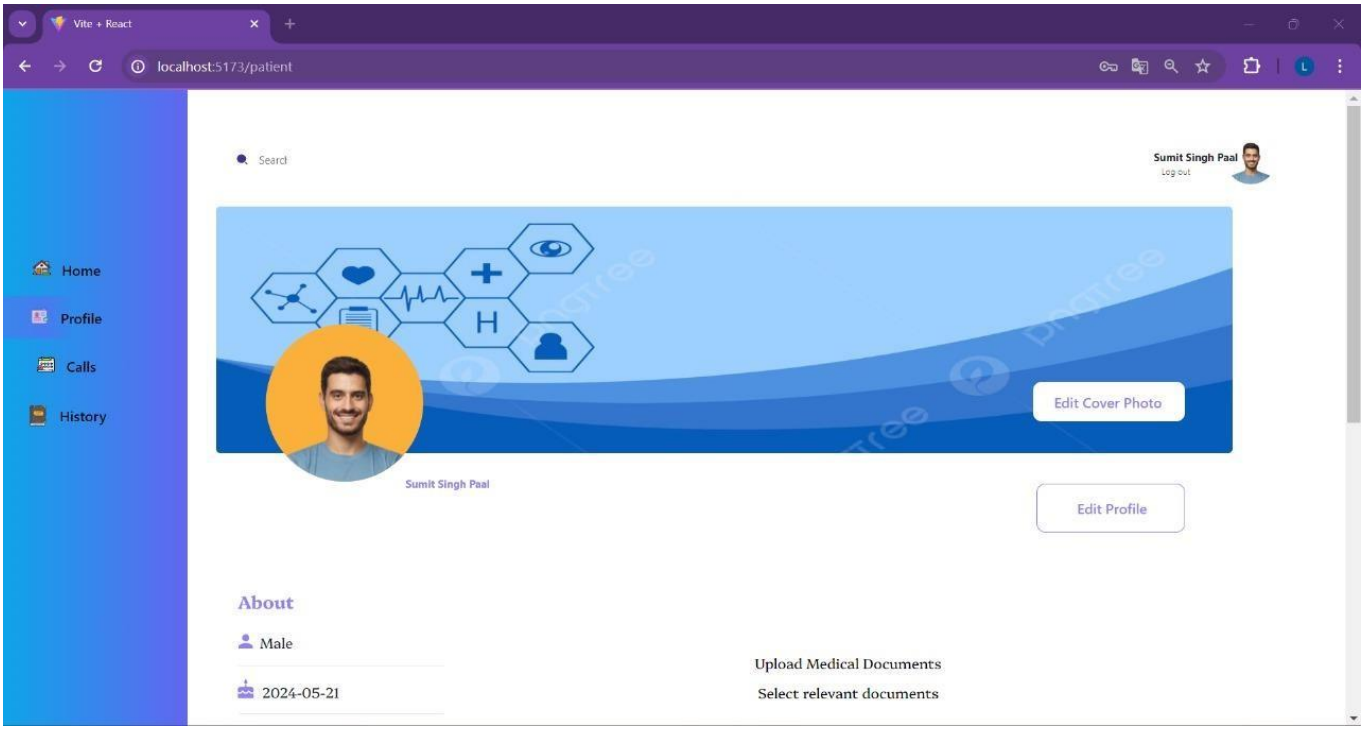


Promote Doctor / Remove Doctor

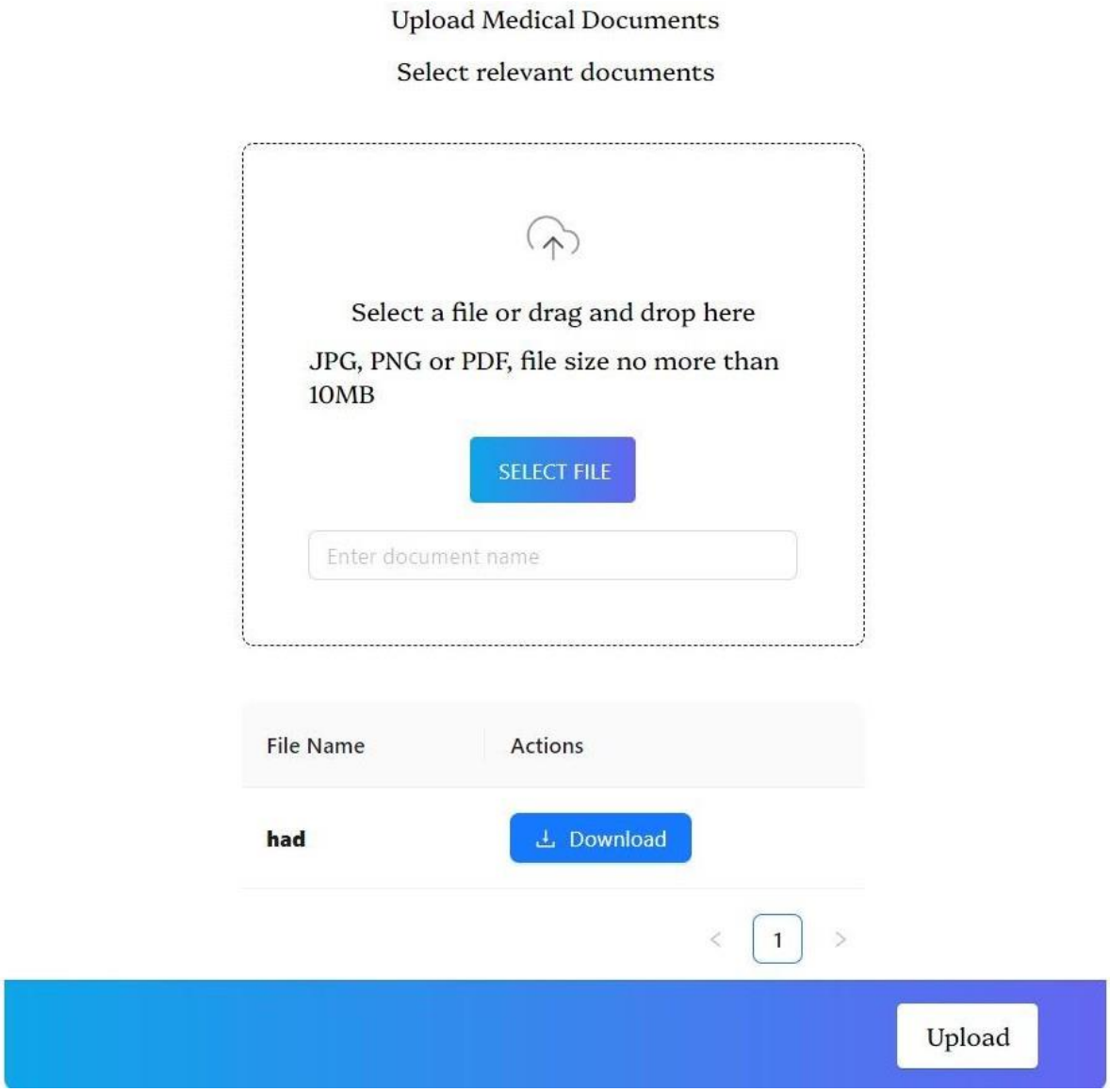


Patient:

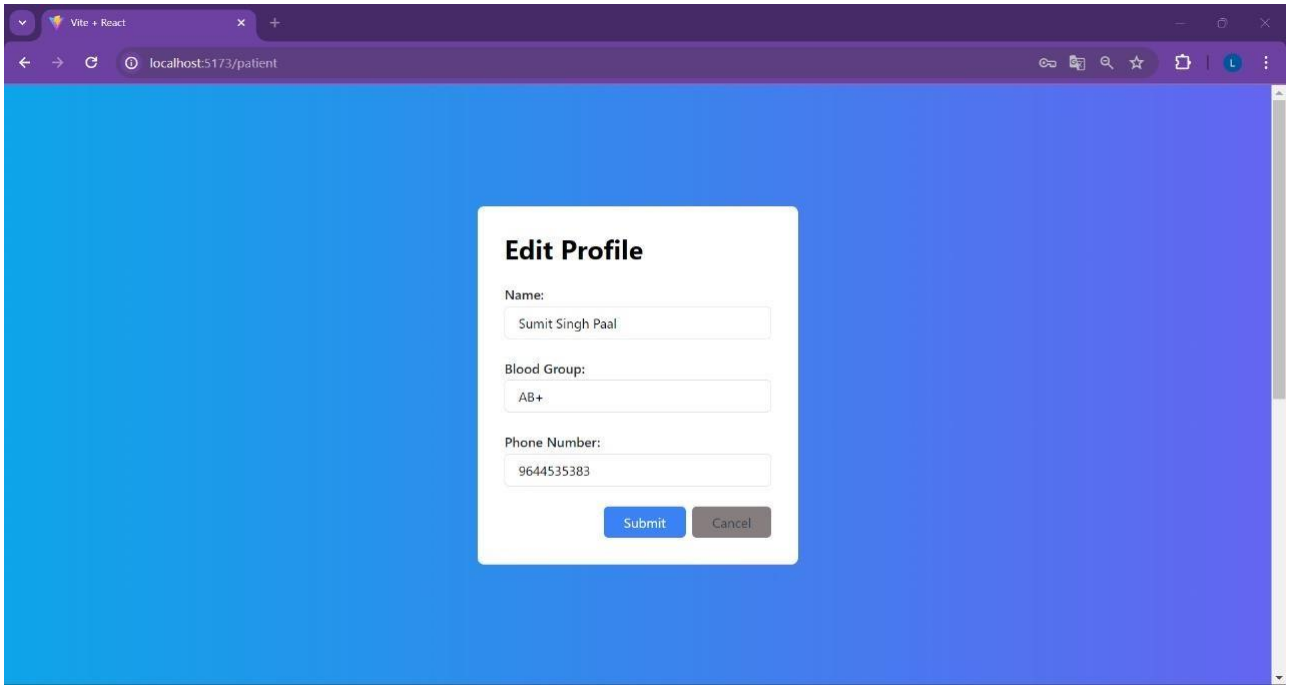
Patient's Profile



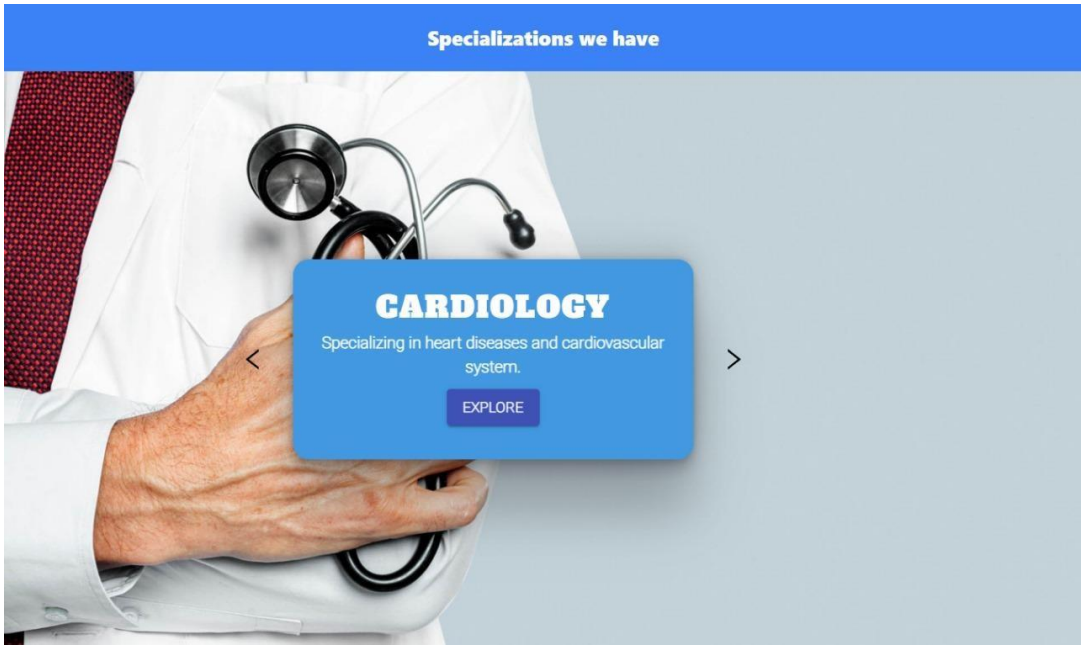
Upload Doccuments



Edit Profile

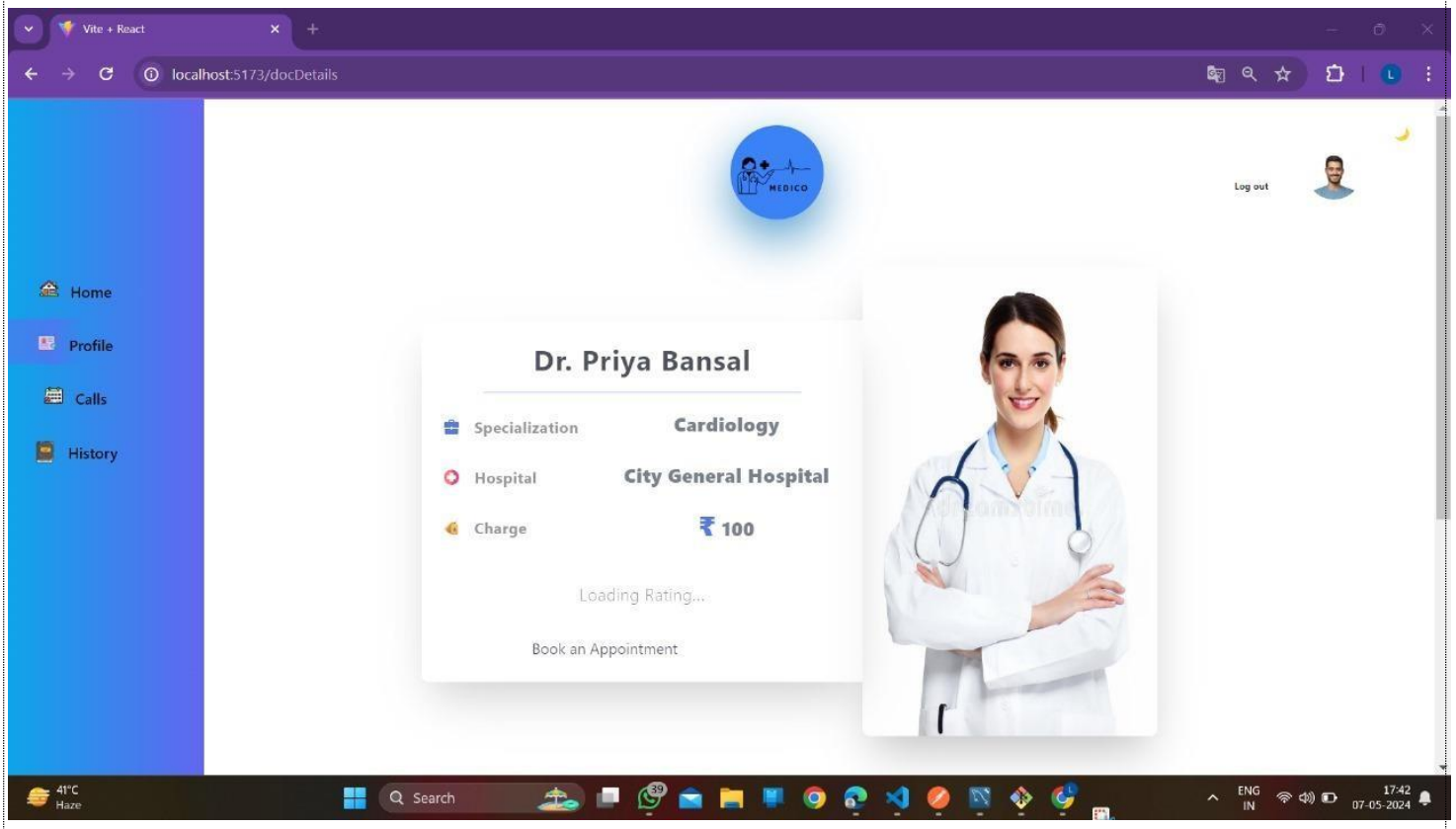


Specializations

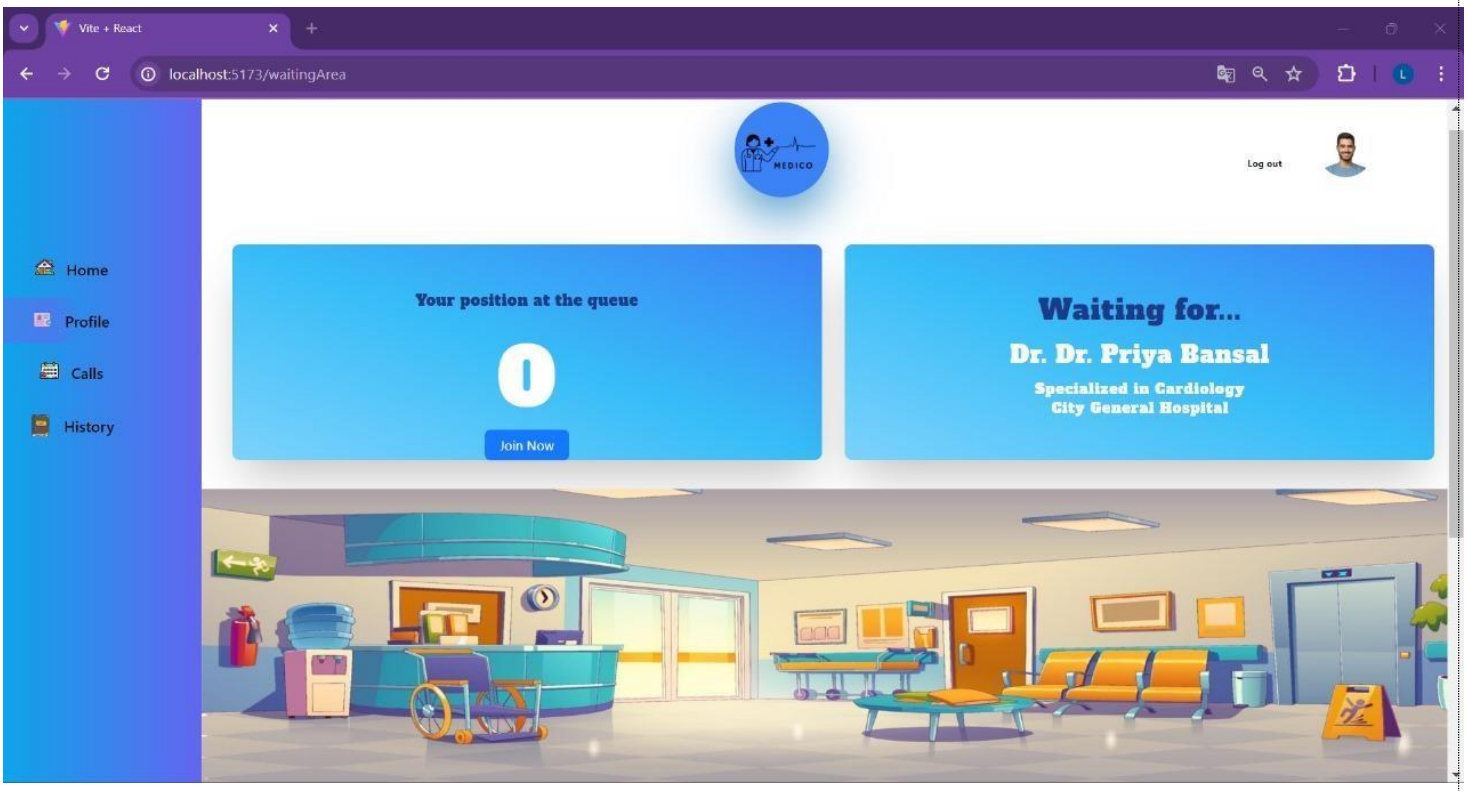


Hospitals

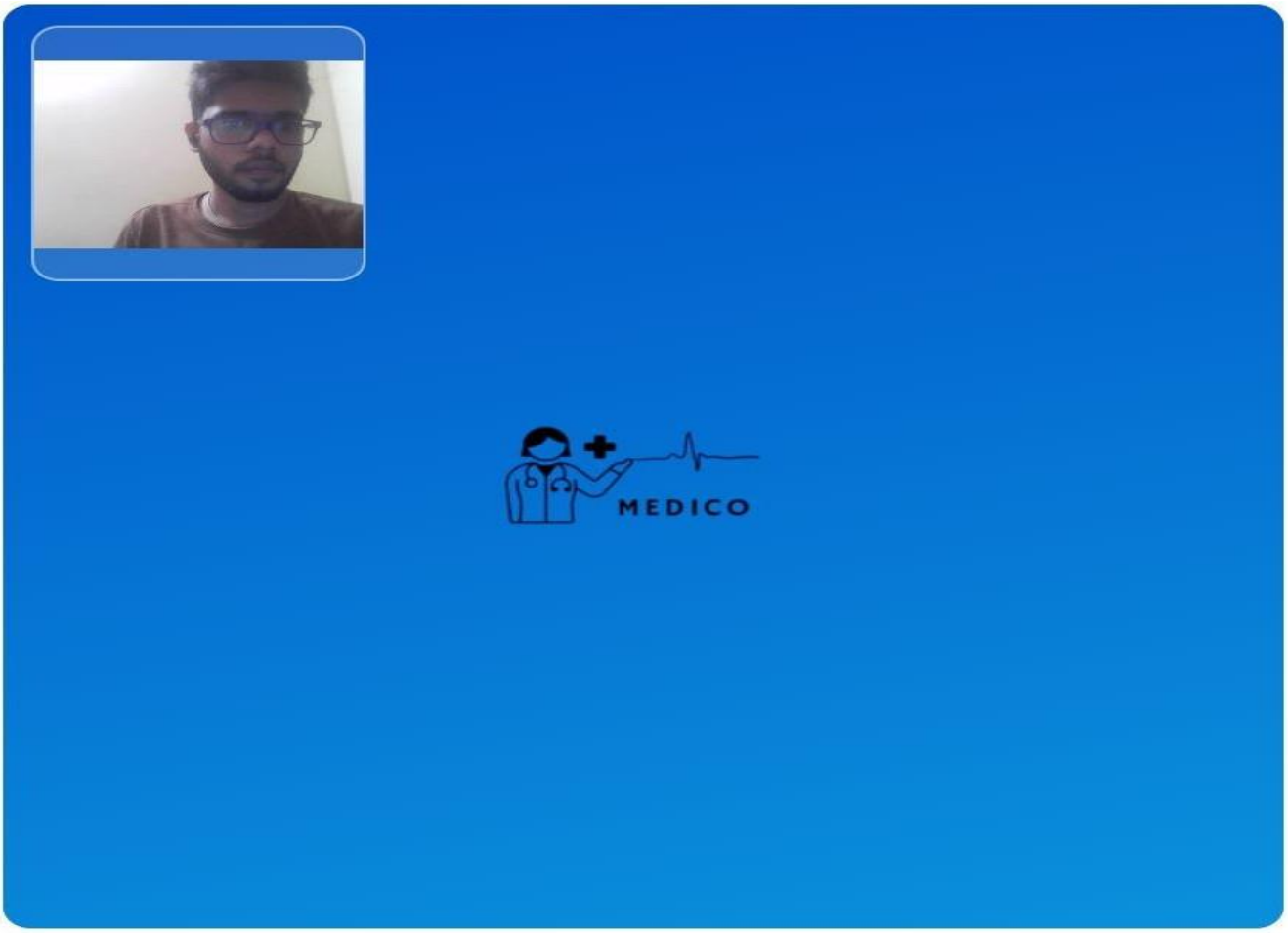
Doctors



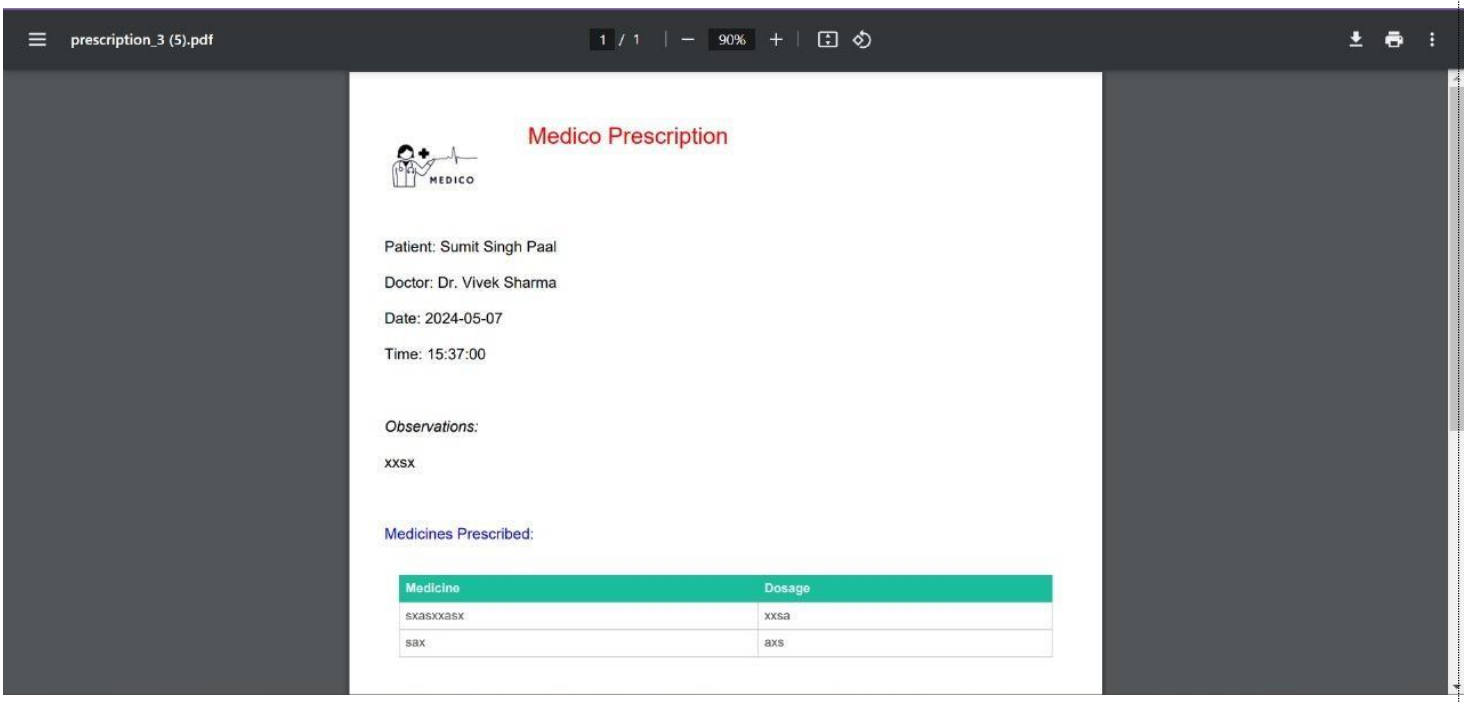
Waiting Area



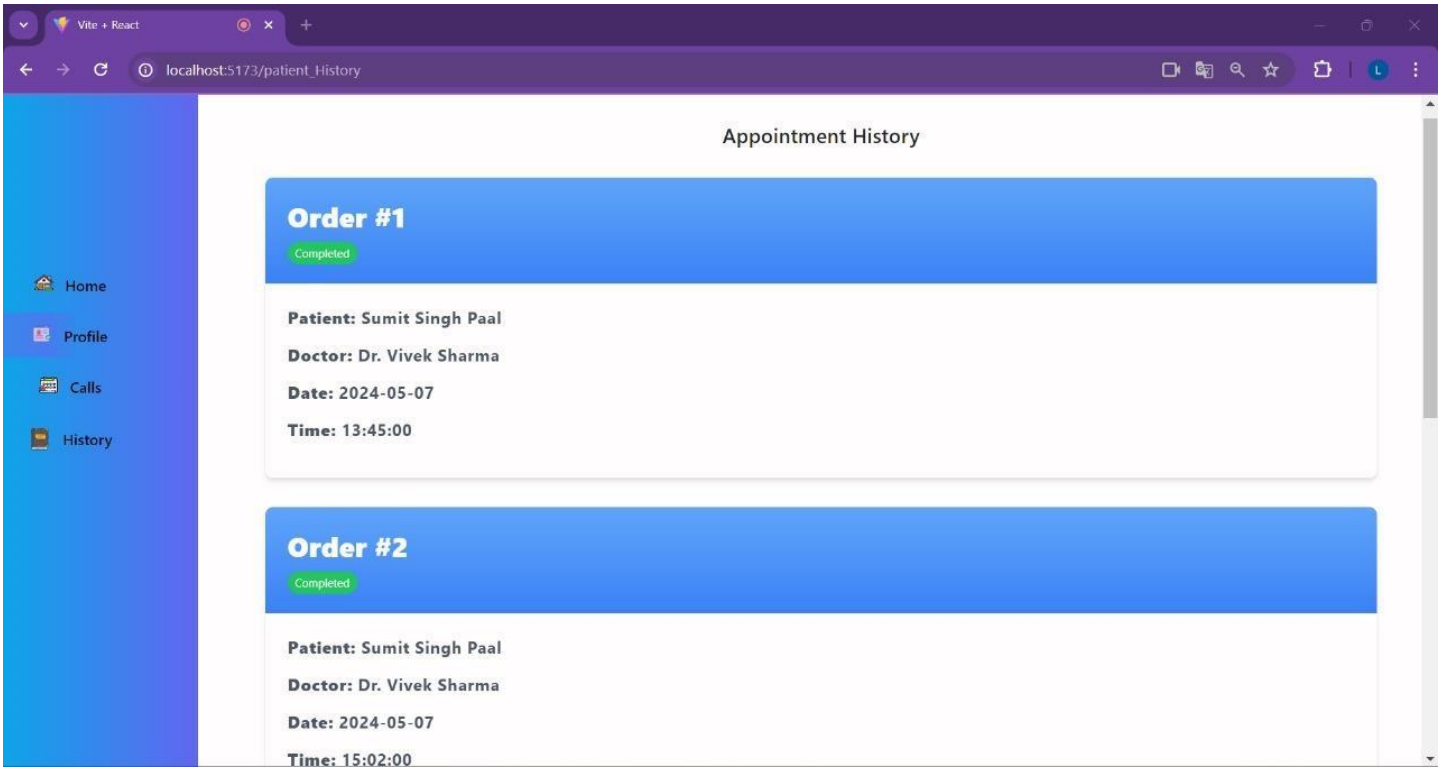
Video Call Screen



Download Prescription

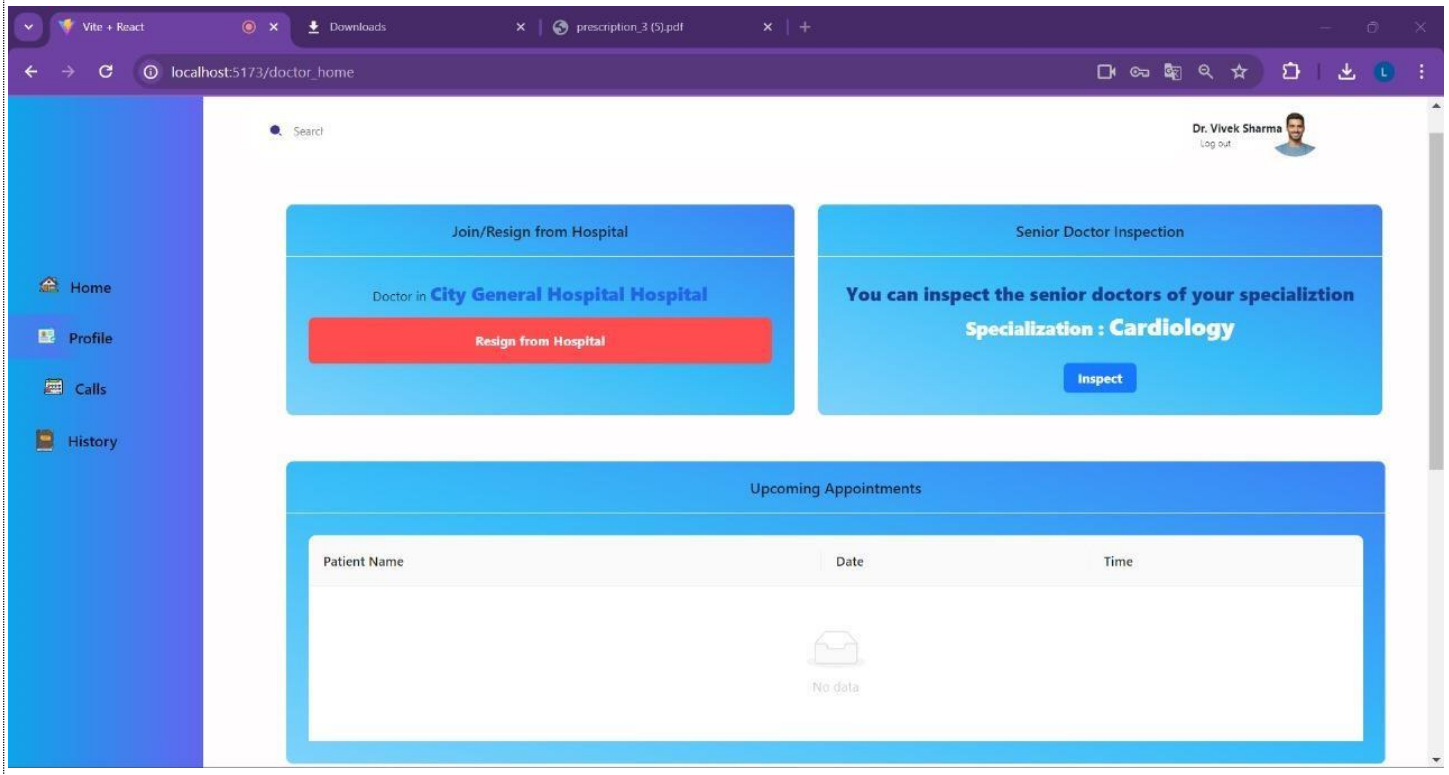


Patient History

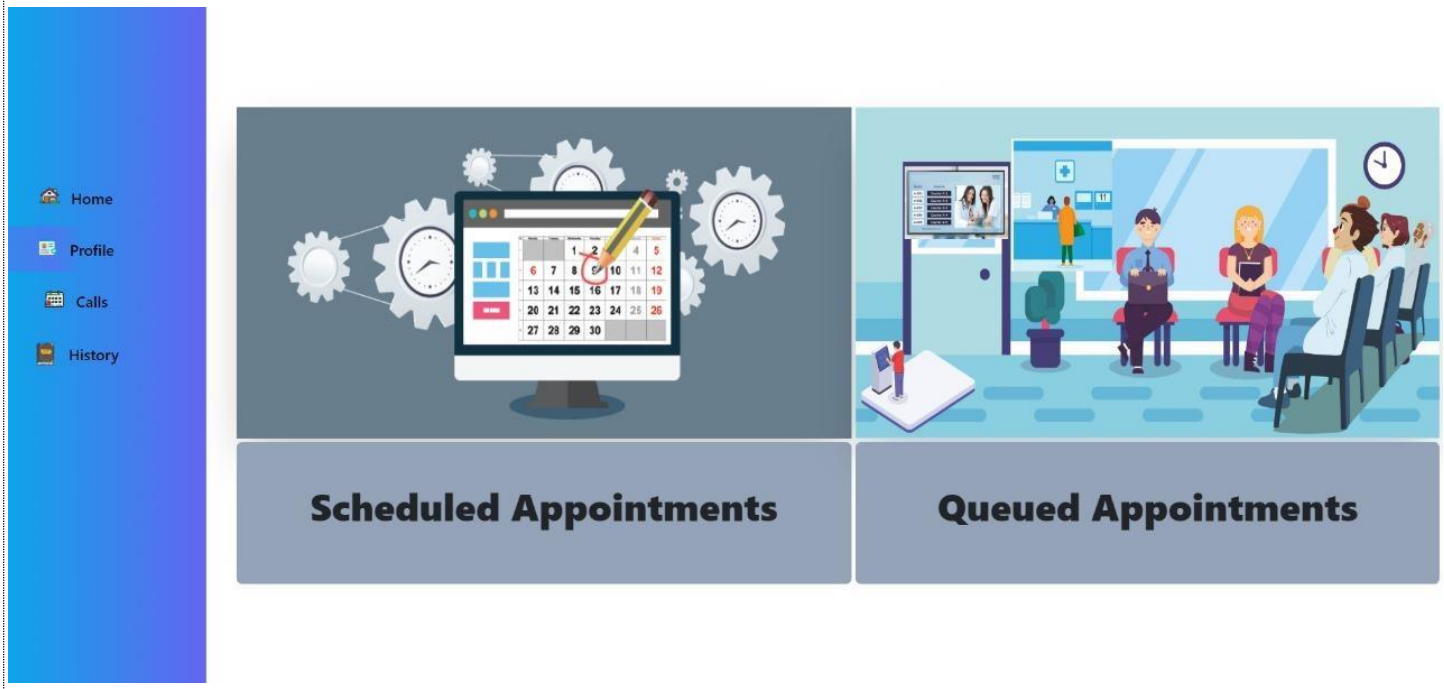


Doctor :

Home

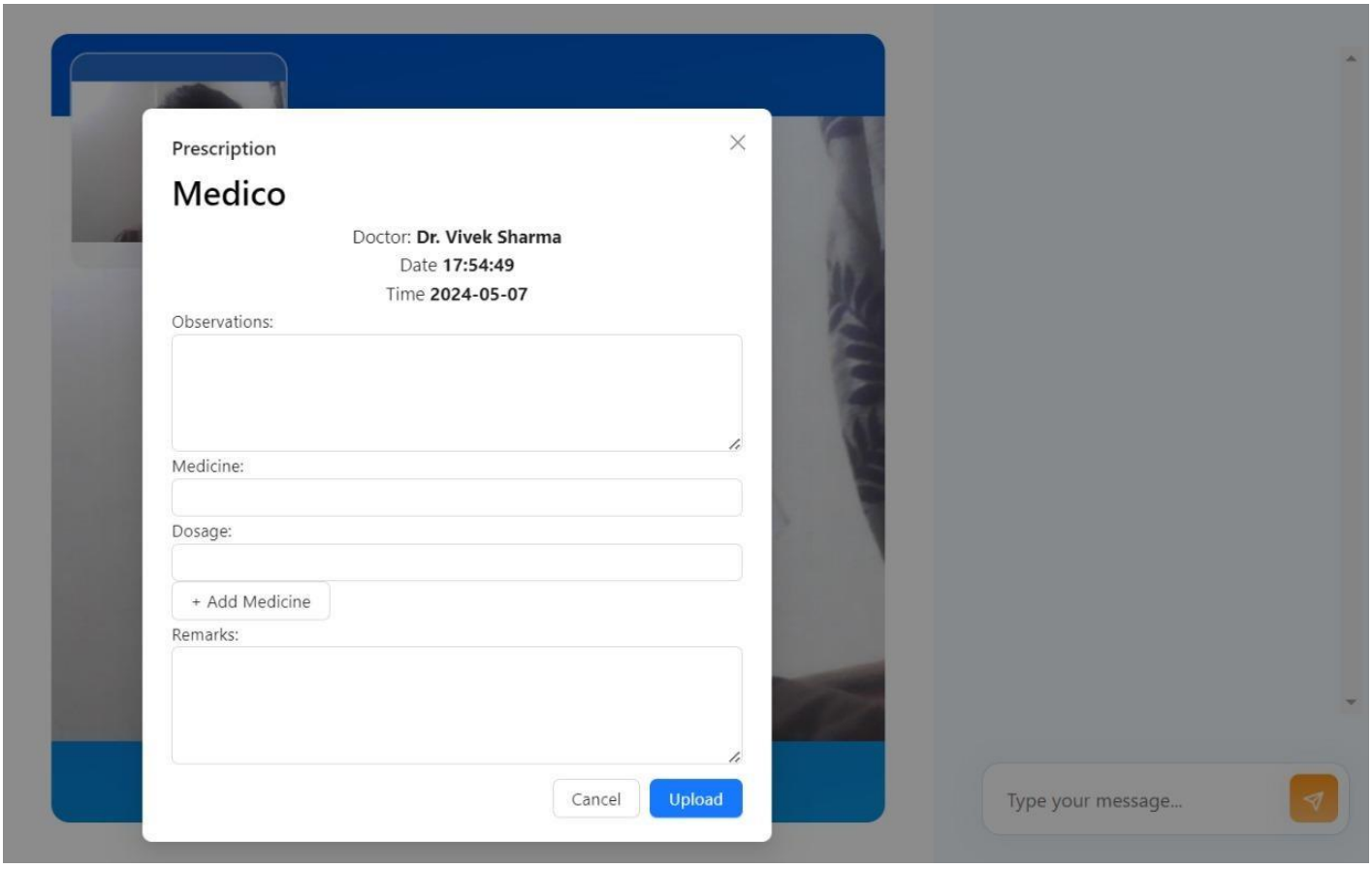


Appointments



Inspection Of Jr Doctor

Upload Prescription(On Call)



Database Design

Database Tables

1. **Consultation:** This table likely serves as a central repository for details regarding patient consultations or appointments. It includes information such as the date and time of the consultation, the reason for the visit, any notes taken during the consultation, and possibly references to the involved doctor and patient. This table facilitates the organization and tracking of patient appointments and medical history within the system.

2. **Doctor Tokens:** This table is likely used for authentication or access control purposes related to doctors. It stores tokens or identifiers assigned to doctors, which may be used for secure authentication when accessing the system or specific functionalities within it. These tokens help ensure that only authorized doctors can access sensitive information or perform certain actions within the system.

3. **One-Time Password:** As the name suggests, this table is likely used to store one-time passwords generated for users, such as administrators or patients. These passwords are typically used for secure authentication or password reset functionality. They provide an added layer of security by ensuring that each password can only be used once, reducing the risk of unauthorized access to user accounts or sensitive information.

4. **Patient Tokens:** Similar to the Doctor Tokens table, this table likely stores tokens or identifiers associated with patients for authentication or other purposes. These tokens help authenticate patients when accessing their medical records, scheduling appointments, or using other services within the system. Like doctor tokens, patient tokens contribute to the overall security and integrity of the system by ensuring that only authorized users can access patient-related information.

5. **Slots:** This table is likely used to manage and track available time slots for appointments or consultations. It may include information such as the date, time, duration, and availability status of each slot. The Slots table facilitates efficient scheduling of appointments between patients and doctors, allowing for the optimal utilization of healthcare resources. It also supports functionalities such as booking, rescheduling, and cancelling appointments within the system.

6. **Admin:** This table likely contains information about administrative users or staff within the system. It may include fields such as login credentials, names, roles, and possibly access permissions. The Admin table facilitates the management of administrative accounts and permissions, ensuring proper access control and oversight of system operations.

7. **Doctor Files:** This table is likely used to store files or documents associated with doctors, such as their credentials, certifications, or other relevant documents. It provides a centralized repository for

storing and managing important documents related to doctors' qualifications and professional credentials.

8. **Hospitals:** This table holds information about different hospitals or healthcare facilities. It may include fields such as names, addresses, contact details, and potentially associated doctors or services. The Hospitals table facilitates the management and organization of hospital information within the system, aiding in patient referrals, scheduling, and resource allocation.

9. **Patient:** This table likely contains personal and demographic information about patients. It may include fields such as names, dates of birth, contact details, medical histories, and other relevant data. The Patient table serves as a central repository for storing comprehensive patient information, enabling healthcare providers to deliver personalized care and track patient health over time.

10. **Prescriptions:** This table stores prescription information, including medication names, dosages, instructions, and potentially the prescribing doctor and patient details. It facilitates the management of medication orders and prescriptions within the system, ensuring accurate tracking and administration of medications to patients.

11. **Sockets:** While the purpose of this table is unclear from the name alone, it could potentially be related to real-time communication or data transfer. This could include functionalities such as video consultations, messaging, or other forms of real-time interaction between patients and healthcare providers.

12. **Admin Tokens:** Similar to the Doctor Tokens and Patient Tokens tables, this table likely holds authentication tokens or identifiers for administrative users. It facilitates secure access to administrative functionalities within the system, ensuring proper authentication and authorization of administrative accounts.

13. **Doctors:** This table likely contains information about doctors, such as their names, specialties, qualifications, and contact details. The Doctors table serves as a comprehensive directory of healthcare providers within the system, enabling patients to find and connect with doctors based on their specific needs and preferences.

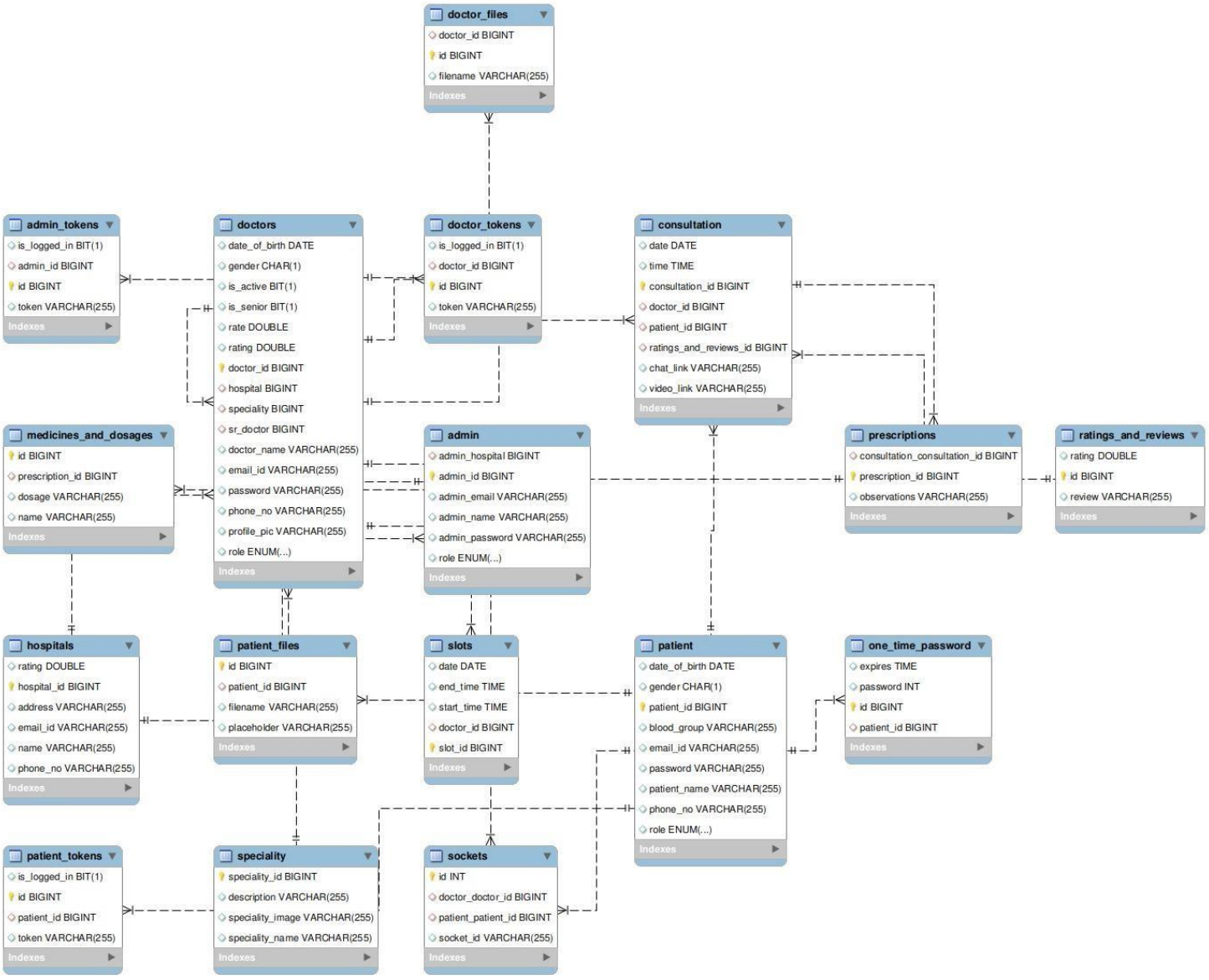
14. **Medicines and Dosages:** This table stores information about different medications and their recommended dosages. It may be used in conjunction with the Prescriptions table to ensure accurate and standardized prescribing practices within the system.

15. **Patient Files:** This table holds files or documents associated with patients, such as medical reports, test results, or any other relevant files. It provides a secure repository for storing and managing patient-related documents, ensuring easy access and retrieval when needed for clinical decision-making or documentation purposes.

16. **Ratings and Reviews:** This table stores ratings and reviews provided by patients for doctors or healthcare services. It includes rating scores, review text, and potentially additional metadata. The Ratings and Reviews table facilitates patient feedback and quality assessment of healthcare services within the system, helping to improve patient satisfaction and overall service quality.

17. **Specialty:** This table likely contains information about different medical specialties or departments. It may be used to categorize or associate doctors with their areas of expertise, facilitating patient referrals, scheduling, and resource allocation based on specialty requirements.

DB Desgin:



Security

JWT:

JWT, or JSON Web Token, is a widely adopted method for securing communication between different components of web applications. It serves as adigital identification card, containing essential information about users and additional data.

JWT consists of three main components:

- **Header:** Specifies the algorithm used for token signing.
- **Payload:** Contains claims about the user and additional information.
- **Signature:** Generated by signing the header and payload with a secret key.

Authentication Process:

- **User Login:** Upon successful authentication, the server generates a JWT token.
- **Token Issuance:** The server sends the JWT token to the user.
- **Token Inclusion:** Subsequent requests from the user include the JWT token.
- **Token Verification:** The server verifies the authenticity of the JWT token,checking the signature and examining the claims in the payload.

JWT authentication safeguards API endpoints from unauthorized access by ensuring that only authenticated users with valid JWTtokens can access protectedresources. It enhances security by preventing unauthorized users from interactingwith sensitive parts of the application.

In addition to JWT authentication, the system implements session management using a Redis server. After successful login, the server generates the JWT token and sets it to its header as an HttpOnly cookie (It will not be accessed by the client-side scripts, such as JavaScript). The Redis server is utilized to store sessioninformation, including the username, access token, session expiry, and the IP address of the user.

Upon each API request, the system verifies the access token sent by the user. It also validates the username and checks the session for its validation status. Furthermore, it compares the IP address of the request with the IP stored in the session. If the IP addresses do not match, the session data is cleared, the access token is nullified, and the user is redirected to the login

page. This multi-layeredapproach ensures robust security and prevents unauthorized access to protected resources.

Role-Based Authentication:

Role-based authentication is a security model that limits access to resources basedon the roles of individual users.

In our project, we have four types of roles:

1. **PATIENT**
2. **DOCTOR**
3. **ADMIN**
4. **SR_DOCTOR**

Each role is assigned specific privileges to access their respective APIs. By restricting access to resources based on user roles, we can mitigate the risk of databreaches and other security threats. This approach ensures that users only have access to the functionalities and data that are relevant to their role within the application.

Data Security Measures:

In addition to JWT authentication and session management, our systemincorporates robust data security measures to protect sensitive user information:

BcryptPasswordEncoder for Password Encryption:

One of the critical components of our security architecture is the utilization of the BcryptPasswordEncoder, a cryptographic hashing function specifically designed for password hashing

Secure Password Hashing:

BcryptPasswordEncoder securely hashes user passwords before storing them in the database. Unlike simple hashing algorithms, such as MD5 or SHA-1, Bcrypt employs a one-way hashing process that makes it extremely difficult for attackers to reverse-engineer the original password from its hash.

Salting:

Bcrypt incorporates a technique called salting, where a unique random salt is generated for each password before hashing. This ensures that even if two users have the same password, their hashed

values will be different due to the unique salt, preventing attackers from using precomputed hash tables, known as rainbow tables, to crack passwords efficiently.

Iteration:

Bcrypt also includes an adjustable cost factor that determines the number of iterations used in the hashing process. Increasing the number of iterations makes the hashing process more computationally intensive, thereby increasing the time and resources required for brute-force attacks. This effectively mitigates the risk of brute-force attacks against hashed passwords.

How Bcrypt Secures Passwords:

Resistance to Brute-Force Attacks: Bcrypt's adjustable iteration count significantly increases the computational effort required to hash passwords, making it impractical for attackers to perform brute-force attacks within a reasonable timeframe.

Protection Against Rainbow Table Attacks: The use of unique salts for each password ensures that even if an attacker manages to obtain the hashed passwords, they cannot be easily cracked using precomputed rainbow tables.

Adaptability to Future Threats: Bcrypt's flexibility allows us to adjust the cost factor over time to stay resilient against evolving computational capabilities of attackers. By employing BcryptPasswordEncoder for password hashing, our system ensures the security of user passwords, protecting them from common attacks such as brute-force and rainbow table attacks. This adds an additional layer of defense to safeguard sensitive user information stored in our application's database

Key Functionalities Achieved

Senior doctor monitoring junior doctor records:

- Ability for senior doctors to provide feedback or guidance directly within the record system.
- Option for junior doctors to request consultations or advice from senior doctors within the platform.
- Feature for senior doctors to track junior doctors' progress and performance over time.

Patient appointment scheduling:

- Option for patients to set appointment reminders via email or SMS..

Real-time appointment queue system:

- Estimated wait time feature to inform patients about their expected wait before connecting with a doctor.
- Queue position notification via SMS or push notification to keep patients informed while they wait.
- Option for patients to request callbacks or leave the queue if their situation changes.

Secure site with authentication and authorization:

- Role-based access control to ensure that users only have access to the information relevant to their role.
- Activity logging to track and audit user actions within the system for security purposes.

Consultation via audio/video/chat:

- Transcript or summary generation feature for consultations conducted via chat for easy reference.
- Screen sharing capability during video consultations for better communication of medical information.

Prescription provision and download:

- Prescription renewal reminders for patients to ensure continuity of care.
- Prescription history tracking for both doctors and patients to monitor medication usage over time.

Hospital registration and doctor independence:

- Option for hospitals to customize their profiles and services offered on the platform.

Patient access to past records and rating system:

- Anonymous feedback option for patients to provide constructive criticism without fear of repercussions.
- Trend analysis of patient ratings to identify areas for improvement and recognize exemplary performance among doctors.

Secure document upload and profile editing:

- Document version control to track changes and revisions made to uploaded files.
- Audit trail feature to monitor profile edits and document uploads for compliance and security purposes.
- Encrypted storage and transmission of documents to protect sensitive medical information.

Document sharing during video calls via OTP:

- Time-limited OTPs for enhanced security during document sharing sessions.
- After the call gets ended there is no opportunity for the doctor's or the hospital to see the document shared by the patient during the call

Additional Functionalities:

Implemented Chatbot to analyze patients symptoms and suggest appropriate specialist doctor

Future Works:

- Integration with translation services for multilingual consultations.
- Ability for patients to specify preferred communication channels or language preferences when scheduling appointments.
- Two-factor authentication (2FA) for an added layer of security during login.
- Integration with pharmacies for direct prescription delivery and fulfillment.
- Prescription renewal reminders for patients to ensure continuity of care.
- Centralized billing and invoicing system for hospital-affiliated doctors for seamless financial management.
- Collaboration tools for hospital staff and independent doctors to coordinate patient care efficiently.
- Patient education resources integrated within the platform to promote health literacy and informed decision-making.
- Option for patients to revoke access to shared documents at any time.
- Integration with digital signature tools for signing and validating shared documents during consultations.

Privacy Policy & Security Documentation:

Data Fiduciaries: Identify the organizations/entities that "determine the purpose and means of processing of personal data"

In accordance with our commitment to privacy and data protection, we act as a data fiduciary for the personal data collected and processed through our teleconsulting platform. As a data fiduciary, we hold a position of trust and responsibility in managing and safeguarding the personal data of our users. Our role as a data fiduciary involves the following key principles:

Purpose of Data Processing: We determine the purpose of processing personal data within our teleconsulting platform, which includes but is not limited to facilitating communication between users and counselors, providing personalized counseling services, and ensuring the security and functionality of the platform.

Data Protection and Security: We prioritize the protection and security of personal data entrusted to us. We implement robust technical and organizational measures to safeguard against unauthorized access, disclosure, alteration, or destruction of personal data. This includes encryption, access controls, regular security assessments, and employee training on data protection best practices.

Nature of the application/platform: Summarize the overall purpose of your solution – what it broadly does, and not how it does it.

Our application enables the users who stay in remote areas of the country access medical consultation services remotely without any physical interaction. By integrating hospital data, and doctor data we're pioneering a transparent healthcare ecosystem. This empowers patients with comprehensive access to their medical information, fostering informed decisions about their health and treatment options. Our commitment to purposeful processing ensures that every data handling activity aligns with our mission of enhancing healthcare accessibility and transparency.

Personal data collected For each item of personal and/or sensitive data, mention the item and the purpose for which it is collected List one data item per line.

1: Patient's data:

- Name: Patient's name is stored for viewing purpose.
- Email: Email is stored for authentication purpose.
- Password: Password is stored for authentication purpose.
- Date of Birth:Date of birth is stored for easy resolution of age-related illness.
- Blood Group:Blood Group is used for disease related issues.
- Gender: Gender is stored to determine the gender of the patient.
- Previous Prescriptions: Previous prescriptions may be stored if the consulting doctor asks for it

2: Doctor's data:

- Name: Patient's name is stored for viewing purpose
- Email:Email is stored for authentication purpose.
- Password:Password is stored for authentication purpose.
- Date of Birth:Date of birth is stored to show the age of the doctor
- Gender: Gender is stored to determine the gender of the doctor.
- Fees: Rate is stored to show the consultation fees of the doctor.
- Speciality: Speciality is stored to show the doctor's specialization.

3: Hospital's data:

- Name: Name is stored for viewing purpose.
- Address: Address is stored for viewing purpose.
- Phone Number: Phone number is stored so that patients can directly call the hospital.
- Email id: Email is stored so that patients can directly mail the hospital

Informed consent : List the steps at which user consent is obtained, and the mechanism of consent collection.

1. During an ongoing consultation, if the doctor requires to have a look at the patient's previous records from another hospital then the doctor sends a request for an OTP to the patient. Upon verification of the OTP the doctor can see the patient's previous prescriptions.

Data Minimization : Are all data items collected required for the proposed processing? Identify those that are not needed for the stated purposes
Yes, all the data items collected are required for the smooth operation of our application ensuring there is no loss in the user experience of the doctor or the patient.

Confidentiality : How is this ensured. List any 3rd parties with whom the data is shared, and the purpose
We are not using any other 3rd party services other than Amazon S3 for static data storage.

Amazon is contractually obligated to adhere to strict confidentiality and data protection standards, including but not limited to encryption, access controls, and data retention policies, to safeguard user data against unauthorized access, disclosure, or misuse.

Purpose Limitation : Identify mechanisms in place to prevent data use for other than stated purposes
Amazon implements robust access controls and encryption measures to safeguard user data shared with them. Access to user data is restricted to authorized personnel with a legitimate need to access it

List the security related mechanisms and technical safeguards to protect privacy :

1. JWT: We have integrated our JWT based authentication and role based authorization for the validation of API requests to our server.
2. Session Management: At the time of login, We are taking username, access token and then maintaining a session for limited time. We are using cookies for storing access token, this is helpful for us to clear the cookie from the application when the user logs out.
3. For the storing of sensitive data like password we are using BcryptPasswordEncoder for the password. During the consultation we are not showing any information about the patient to the doctor.

Data Retention For what period of time is the collected data stored. What happens to the data at the end of that period:

For now we are not putting any time limit for the data collected , because all the data are connected to all the tables.

Data Deletion Requests Describe the mechanism by which users can request deletion of some or all of their data. If no such mechanism exists, state so

There is no mechanism with which the patient or the doctor can delete their individual records. This has been done to maintain the user experience. However they can permanently delete their accounts upon which all their records will get lost.

Account Deletion How are requests for deletion of accounts by users handled. What happens to the data of accounts that are deleted

The patients and doctors have the option of deleting their accounts permanently. Upon deleting their accounts all their information including the past consultations will be removed.