curabl - Virtual Clinic

Stage-II Mid-Sem Report

Submitted by

Ghanshyam Patil **111803091**

Chaitanya Shinge **111803099**

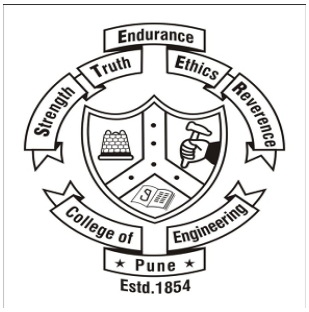
Adarsh Ninganur **111803115**

Under the guidance of

Prof. Satish S. Kumbhar

Prof. Uttam Chaskar

College of Engineering, Pune



DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE-5

March, 2022

Abstract

The current COVID-19 pandemic has caused significant strain on medical institutions worldwide. This has radically altered how medical practitioners provide care to their patients.

As a consequence, there is a crucial need for adoption of digital tools and technologies such as telemedicine and virtual care. Moreover, rural areas of India don't have access to proper medical resources and specialists. They have to travel from remote areas to cities to have access to these resources, which needs significant time and money.

Our virtual platform comes into the picture here. We aim to provide the most affordable and the best medical resources accessible to everyone. With our platform, anyone can login and book an appointment with the best doctors and specialists and also check reviews to get some assurance. Patient can search for doctors by entering their symptoms in our system. Our platform predicts the disease using a trained model and suggest relevant doctors to patients. This will not only save the patient's time and money but also help prioritize the doctor's schedule and aid to be accessible to more patients. Moreover, our platform will include all the past medical records of the patient so the diagnosis of the patient can be smooth and hassle-free. With the help of IoT devices, the doctor can view real time health data of patients while in a video conference, thereby simulating a physical check-up.

Through research, we found that existing healthcare systems either support only patient-doctor communication, or doctor-professional communication, or IoT based health-monitoring systems for patients. There is a need to combine all these services under one platform.

Contents

List of Tables

List of Figures

List of Symbols

Chapter 1

Introduction

1.1 Motivation

Technology has been rapidly evolving in all sectors in the last few years. Using technology to deliver health care has several advantages, including cost savings, convenience, and the ability to provide care to people with mobility limitations, or those in rural areas who don't have access to a local doctor or clinic.

Technologies like IoT, real-time communication, and visualization techniques have risen to prominence in the healthcare sector. They have a simplistic approach and provide comfort to the end users - Doctors and Patients. There are various systems right from hospital management to remote monitoring that target specific use-cases but they don't cover all aspects.

So, we came up with a solution to combine all the services under one common platform. We aim to build a system which will be more user-friendly and be accessible to everyone.

1.2 Product Scope

The product is accessible to all patients who seek any kind of consultation or medication. Also, Doctors from Primary Health Care Center, Specialists and other professionals have access to the product.

Chapter 2

Literature Review

2.1 Literature Review

Year of Paper	Title of the Paper	Journal/ conference details	Methodology	Proposed idea	Advantages/ achieved objectives in paper	Disadvantages/ Limitations
2020	Use of Telemedicine and Virtual Care for Remote Treatment in Response to COVID-19 Pandemic Link	Journal of Medical Systems	The study aims to provide recommendation and guidance to medical practitioners on the use of telemedicine to address COVID-19 response and similar challenges during possible future disasters.	The proposed idea focuses on the current legislations for use of telemedicine and virtual applications during COVID-19 era, guidelines for use of telemedicine and virtual applications, and challenges and recommendations for use of telemedicine.	➢ Telemedicine and virtual care can be used to provide needed medical-care to patients even after the pandemic. ➢ Findings suggest that telemedicine allows for examination of a patient's health. ➢ It is also indicated that stored patient health information can provide guidance for future examination.	➢ Some hospitals and clinics in some countries may not have the resources to manage the surge in telemedicine and virtual patient care.

2020	Virtual Consultations and the Role of Technology During the COVID-19 Pandemic for People With Type 2 Diabetes: The UK Perspective Link	Journal of Medical Internet Research	The study evaluates the feasibility and effectiveness of virtual clinics in diabetes care before and during the COVID-19 pandemic. Moreover, it discusses steps to establish a virtual diabetes clinic, a guide for health care professionals conducting virtual clinics, and a patient guide.	The proposed idea is to digitize diabetes care by setting up proper infrastructure, so that virtual consultations can be made more reachable at the national level.	➢ Better care, cost savings and a reduced environmental impact in telemedicine. ➢ Some centers have seen a rapid growth in their virtual consultations. ➢ Improved biochemical parameters are reported for people in virtual clinics.	➢ The staff faced major barriers in the transfer of their usual processes to the digital platform. ➢ Routine diabetes clinic tests which include checking BP, urine sample, HbA1c, spot checks are not possible in the same way in the virtual mode.
2020	How effective is the virtual primary healthcare centers? An experience from rural India Link	Journal of Family Medicine and Primary Care	The case study discusses the E-mitra clinic (Jiyyo) which aims to expand health outreach in rural areas of India and provide primary health care services by connecting local practitioners with qualified allopathic doctors online.	The idea of Jiyyo e-clinic is to make affordable healthcare services accessible to the people of rural areas and connect them with specialists when required.	➢ Jiyyo has made telemedicine available to areas with poor infrastructure. ➢ There is no need for costly equipment - only a smartphone. ➢ The patient turnover is high in these e-clinics indicating they are satisfied.	➢ Many patients require multiple tests before diagnosis which is limiting in the case of e-clinics. ➢ Small sample size was taken for the study.

2011	Mobile based Primary Health Care System for Rural India Link Journal Link	International Journal of Nursing Education	The study explores the present status of Mobile based Health Care systems in different countries. And in India how it will be beneficial in rural areas .	The proposed idea focuses on the shortfalls in Primary Health Care Management in rural India, and the potential solution to fill it with the enabling of Mobile Web technologies for Primary Health Care management.	➢ Increased quality of primary health care (PHC) services. ➢ Improved epidemiological surveillance and control. ➢ Reduction of maternal and perinatal mortality and morbidity.	➢ Not implemented yet. ➢ Problem in the accuracy that physical check-up can have.
2016	E-health monitoring system Link	International Conference on Applied Internet and Information Technologies	The study aims to propose an e-health system that allows doctors to closely monitor patients' vital parameters, no matter where they are located.	The proposed idea focuses on remote capabilities that enhance the level of medical support a patient receives while enabling them to be monitored in the comfort of their home.	➢ Beneficial to patients as it reduces costs and travel times. ➢ Using this system the patient's health can be monitored from any location on a daily or weekly basis. ➢ Can be used for health care of elderly living alone in their homes, patients in rural areas, as well as chronically ill patients.	➢ Common Database/Replica/ Single point failure. ➢ Application not user friendly. ➢ Security information not provided.

2000	CoMed: a real-time collaborative medicine system Link	International Journal of Medical Informatics	A study of telemedicine based web-systems was done. The system was developed to share multimedia patient databases between medical specialists and communicate in real-time on the Internet. Technologies trending at that time like synchronous interaction technology, CORBA, Java Applet were used.	The proposed system consists of a multimedia medical database relevant to laryngeal diseases and a real time collaboration system consisting of a video teleconferencing system and a whiteboard and chatting system.	➢ Real time communication between doctors through chat, white board and video call. ➢ Flexible and extensible device-independent distributed application.	➢ Does not include booking appointments and e-payment features. ➢ Patients cannot search doctors for treatment. ➢ Specific to patients diagnosed with laryngeal diseases. ➢ Does not use IoT and cloud computing features.
2021	A study and an implementation of online doctor consultation system Link	Journal of Applied Technology and innovation	A study of how booking systems operate in a clinic was done and a successful attempt to virtually recreate it was made. The proposed system focused on the patient's privacy and was developed using Rapid Application Development (RAD) methodology.	The proposed idea is focused on booking consultation time with doctors and can meet the doctor as per the consultation time.	➢ Maintaining patient's privacy with two security protocols which is SQL injection prevention and encryption. ➢ Scheduling appointments and feedback system	Major changes needed in the system to add following features- ➢ Online consultation through Video Conferencing in a secure way. ➢ E-payment feature

2020	A home hospitalization system based on the Internet of things, Fog computing and cloud computing Link	Informatics in Medicine	A study of benefits of the concept of Fog computing, Cloud Computing and IoT in healthcare systems were identified and objectives were carved. Precision/Accuracy in sensory measurements and real time monitoring systems was the primary focus.	The proposed idea focuses on an home hospitalization system based on the IoT, Fog computing, and Cloud computing. It integrates monitoring patient's health and environmental factors of the hospitalization rooms in one system, letting doctors diagnose the state of their patients remotely.	➢ Easy to apply, low cost, reliable and safe. ➢ Enables doctors, patients, and their family members to manage & monitor hospitalization operations through their mobile applications. ➢ Proposed system is scalable and can support new features like video conferencing, e-payments as well.	➢ App is compatible with Android platform only. ➢ It is just an IoT and cloud/fog-based monitoring system.
2017	An enhanced device-transparent real-time teleconsultation environment for radiologists Link	IEEE Journal of Biomedical and Health Informatics	A study of existing healthcare systems and a survey of radiologists was done. More focus was given on creating a distributed system supporting video call and chat functionality between professionals.	The proposed idea focuses on building a scalable, extensible system using microservices architecture, P2P networking and the SPA to promote collaboration among geographically distributed healthcare professionals.	➢ Increased workloads can be easily handled by dynamically spawned service instances. ➢ Further functionality extension could not affect the performance and responsiveness of the platform.	➢ System lacks patient to doctor communication and related features

2.2 Conclusion

After reviewing these research papers, it is clear that if a virtual clinic system is successfully created and deployed, doctors and patients both would benefit a lot. All the research papers highlighted that there is a crucial need for a virtual healthcare system that can integrate all the requirements and minimize the limitations mentioned.

Failures and weaknesses in the past implementations of such healthcare systems have given us insights on problems in rural areas and patient's privacy. The literature study helped identify additional requirements that can comfort patients and doctors like appointment scheduling, payments, Real-time health data, etc. and also guided us in designing our system architecture.

Now that we have a good understanding regarding virtual healthcare systems, we will develop the system accordingly.

Chapter 3

Research Gaps and Problem Statement

3.1 Research Gaps

- Most of the existing applications only consist of online consultancies. There don't have any remote checkup of patients with medical devices.
- There is currently no platforms which support patient-doctor and doctor-doctor communication and discussion.
- The implementation of client software with IoT devices in a commercial environment is limited.
- There are not many services aimed at the rural population where medical resources are inaccessible.

3.2 Problem Statement

3.2.1 Scope

To build a virtual clinic for doctors to address the patients' issues, give them proper medication / prescription or give consultation and collaborate with professionals to discuss patients' issues.

3.2.2 Objectives

- Build a platform for doctors to address the patients' issues and give them proper medication / prescription or give consultation.
- Seamless experience for patient issue resolution in video calls with chat feature.
- Integrate instruments with databases to successfully display real-time data to the doctor.
- Doctors should be able to communicate with other specialists, experts over a patient's diagnosis, medication or other issues.

3.2.3 Outcomes

- The user (patient or doctor) should be able to sign up and log in to their accounts.
- The user should be able to search for a doctor for whatever illness or specialist the user wants, nearest hospitals.
- The patient user should be easily able to book appointments.
- Doctor User should be able to see the previous medical history of the patient.
- Doctor User should be able to see live patient health data.
- Get a Record of the Prescription from the Doctor which should be visible to the patient.
- Patients must be able to view the Ratings of various other patients. Also, they would be able to give their own ratings and reviews of the doctor.
- The Doctor should be able to view the payments received from all the patients.

Chapter 4

Proposed Methodology/ Solution

4.1 Methodology

The existing healthcare system has limitations in its approach. With our software, we aim to connect patients and doctors where they can get reliable consultancy remotely. This is especially much more of a significant problem in rural areas where patients don't have access to proper medical resources. Through our virtual clinic, we provide facilities like :

- Examination through ECG
- Stethoscope
- Pulse Oximeter
- Temperature measure
- And other such IOT devices.

Patients can input their symptoms and our ML model will give disease relevant to those symptoms and recommendations for the appropriate doctor. The patient sees a list of appropriate doctors which he/she chooses to consult.
After this step, the patient will book an appointment with the doctor according to the available slots and payment is done. At the slot time, both doctor and patient are expected to join. They are connected through live video call along with all the sensor data which will be displayed to the doctor virtually on their desktop.
Literature Survey of several research papers was done and we identified the various services needed to implement in the project. We followed the Incremental Model SDLC for this project.
Technologies used:

- Backend - ExpressJS, Nginx, Node.js, Docker,
- Frontend - Ant-design - React design library, ReactJs,
- Jitsi - secure, open source video conferencing system,
- Rocketchat - open source, secure and integrable chat system,
- Razorpay - Popular integrable Payment APIs,
- Database - MongoDB.

4.2 Diagrams

4.2.1 Architecture Diagram

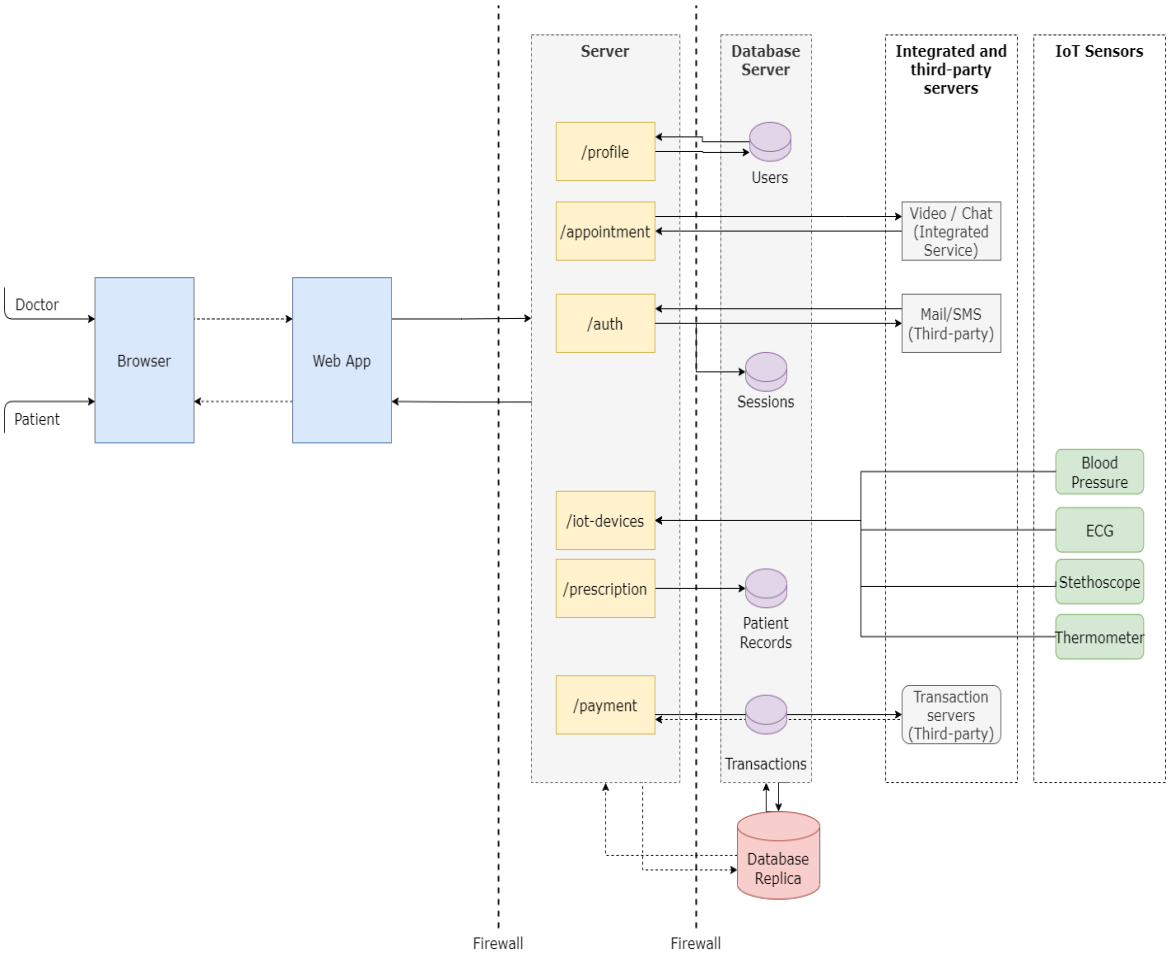


Figure 4.1: Architecture Diagram

4.2.2 ER Diagram

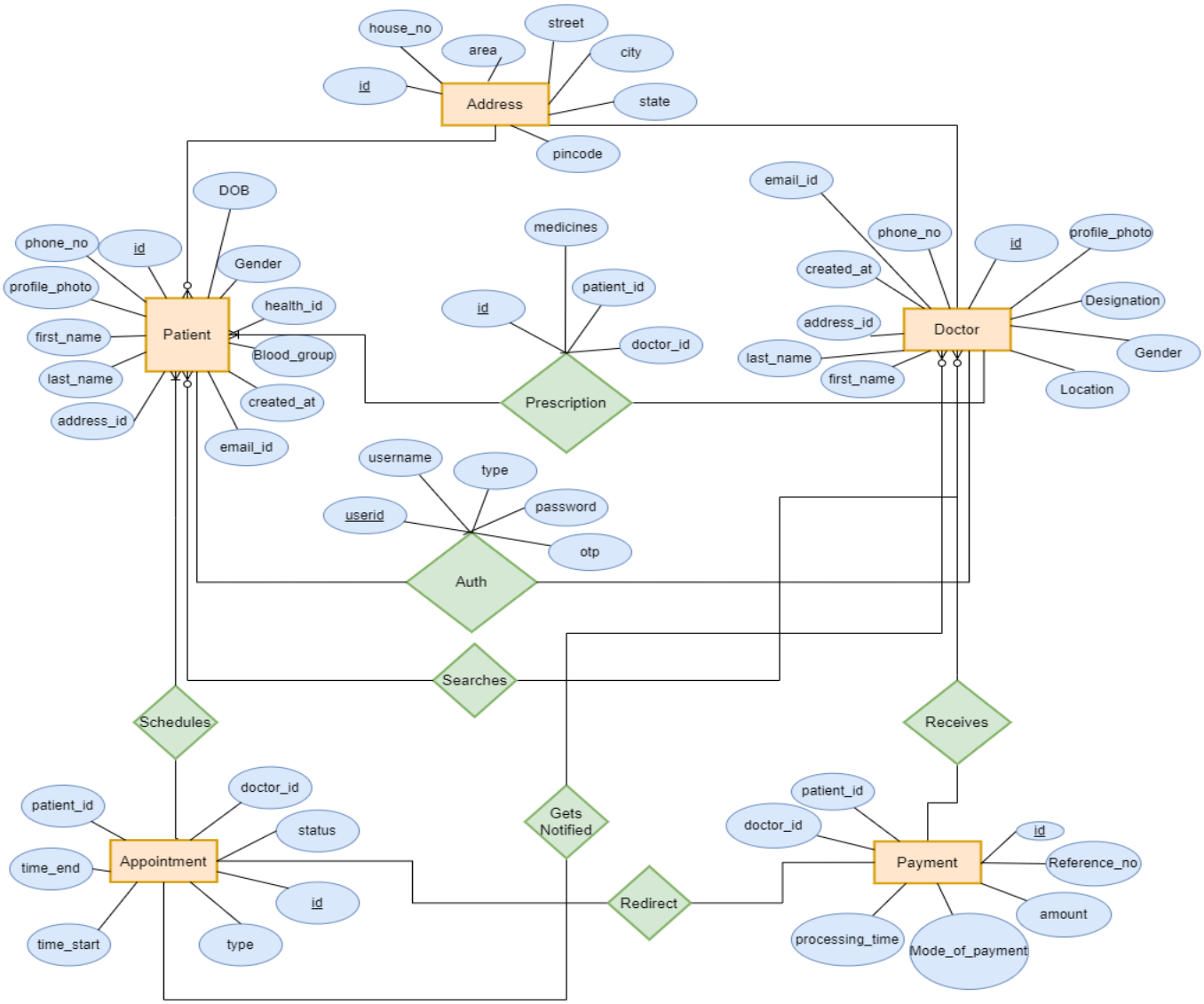


Figure 4.2: ER Diagram

4.2.3 Class Diagram

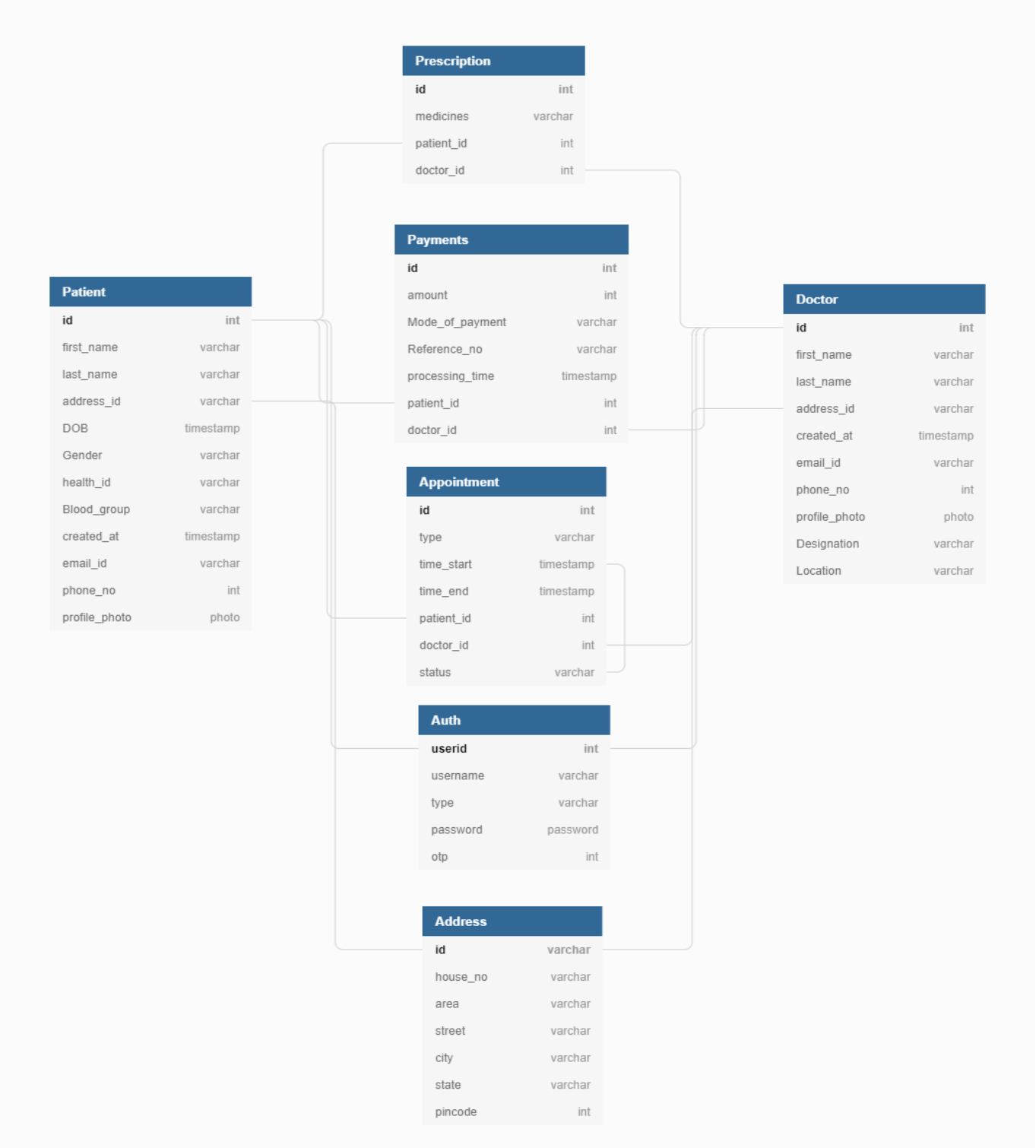


Figure 4.3: Class Diagram

4.2.4 Activity Diagram

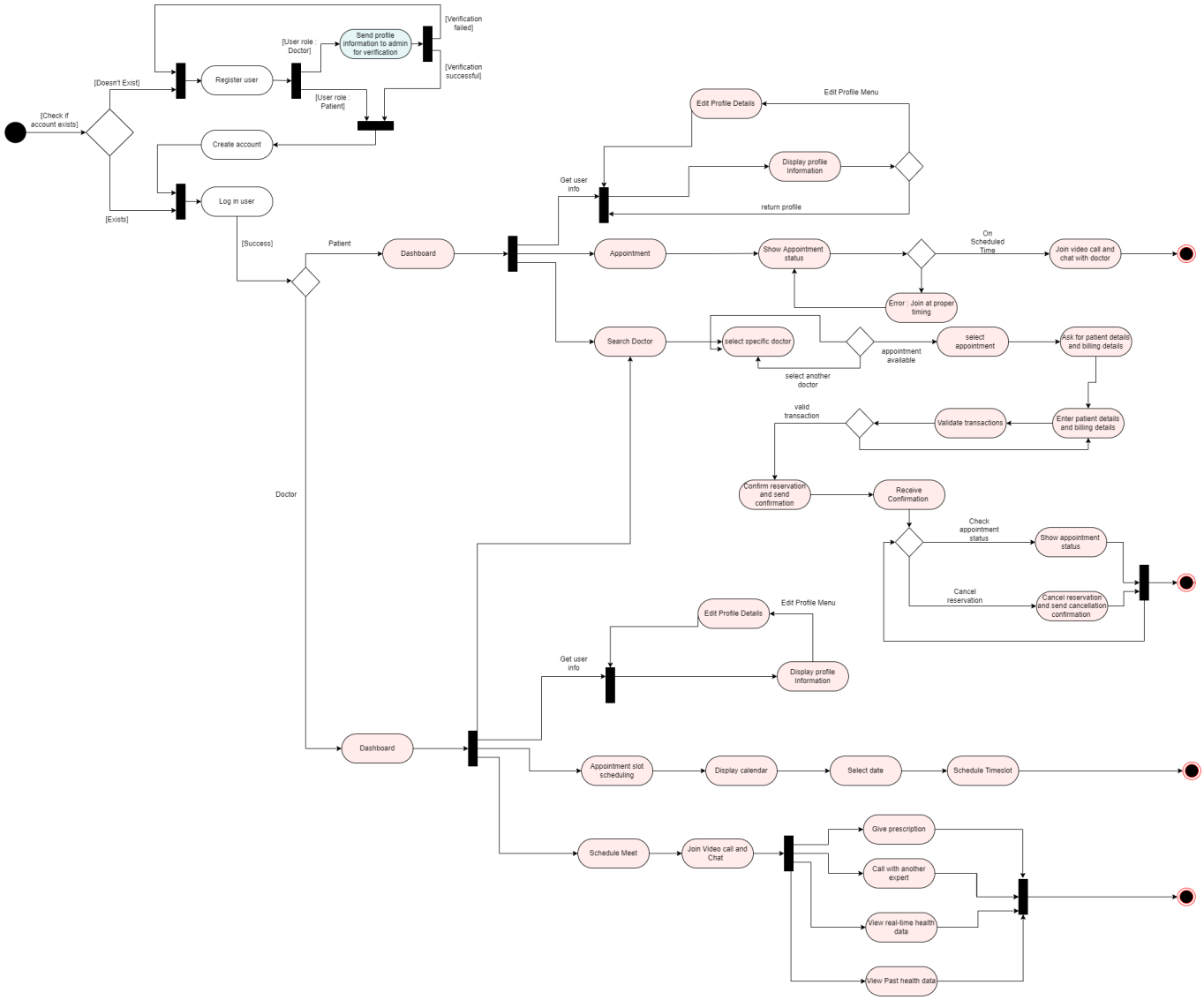


Figure 4.4: Activity Diagram

Chapter 5

Experimental Setup

5.1 Git Setup

We are maintaining our codebase using Git - a version control system. Git is the de facto industry standard compared to others like Mercurial. We have pushed our codebase to a remote repository on Github. We preferred Github because it is a social platform. It has features like CI/CD through Github Actions which enables automation of deployment processes.

5.2 Client environment

React is a popular frontend Javascript library among many IT companies. We have used React to build our Frontend. React also has a strong developer community. To make our frontend design more uniform, we have used the Ant Design library. Ant Design is a React UI Library which defines ready-to-use high quality React UI components. Moreover, Ant Design has good documentation on how to use its React Components. All the libraries and supporting packages are managed by npm - a package manager for Node runtime-based packages.

5.3 Server environment

Express is a flexible backend Web Framework based on the Node runtime environment. Companies like PayPal, Uber and IBM use Expressjs. Express has many supporting packages for different tasks like managing sessions, payload parsing and validation, routing, etc. We have built our backend using Express. There are several packages that support integrating

third-party applications. Express keeps the codebase more organized. The API routes created using Express were tested using the Postman application. Postman is a Chrome engine based client. All the HTTP routes were tested using Postman and were refined. All the libraries and supporting packages are managed by npm - a package manager for Node runtime-based packages.

5.4 Hosting and Configuration

We have created a virtual machine using Microsoft Azure's IaaS(Infrastructure as a Service). On this VM, we have hosted our server and other third-party services like Rocket.Chat, Jitsi. Azure provides us with a Public IP address to access the VM. We have mapped a domain name 'curabl.me' from 'namecheap.com' to this public IP. We connect to this VM through the SSH command-line program using this domain. To handle incoming HTTP/HTTPS traffic in the VM, we have installed Nginx Reverse Proxy service. We have configured it to route the incoming requests to appropriate services.

5.4.1 Hosting the server

We are currently using the Nodemon program to run the server continuously on the VM. We have added a Nginx configuration file for the server which allows users to access the API server using the subdomain 'api.curabl.me'.

```
dev@curabl:~$ ls /etc/nginx/sites-enabled/
api.curabl.me.conf  chat.curabl.me.conf  default  video.curabl.me.conf
dev@curabl:~$ cat /etc/nginx/sites-enabled/api.curabl.me.conf
#The Nginx server instance
server{
    server_name api.curabl.me;
    location / {
        root /home/github-actions/static;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
        proxy_pass http://127.0.0.1:9000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        # location /overview {
        #     proxy_pass http://127.0.0.1:3000$request_uri;
        #     proxy_redirect off;
        # }
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/api.curabl.me/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/api.curabl.me/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server{
    if ($host = api.curabl.me) {
        return 301 https://$host$request_uri;
    } # managed by Certbot
```

Figure 5.1: Nginx Configuration File for curabl Server

5.4.2 Hosting third-party services

- **Rocket.Chat** Rocket.Chat is a popular, highly configurable and open-source chat service that is free to use if hosted separately. We have rebranded Rocket Chat. We are running the Rocket Chat service on a VM in a Docker Container. We have added a Nginx configuration file for the server which allows users to access the Rocket Chat service using the subdomain ‘chat.curabl.me’.

```
dev@curabl:~$ ls /etc/nginx/sites-enabled/
api.curabl.me.conf  chat.curabl.me.conf  default  video.curabl.me.conf
dev@curabl:~$ cat /etc/nginx/sites-enabled/chat.curabl.me.conf
upstream backend {
  server 127.0.0.1:3000;
}

server {
#    listen 443 ssl;
#    listen [::]:443 ssl;
    server_name chat.curabl.me;

    client_max_body_size 200M;
    root /var/www/rocket.chat;
    access_log /var/log/nginx/rocketchat-access.log;
    error_log /var/log/nginx/rocketchat-error.log;

#    ssl_certificate /etc/letsencrypt/live/chat.curabl.me/fullchain.pem;
#    ssl_certificate_key /etc/letsencrypt/live/chat.curabl.me/privkey.pem;
#    include /etc/letsencrypt/options-ssl-nginx.conf;
#    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        # Simple requests
        if ($request_method ~* "(GET|POST)") {
                add_header "Access-Control-Allow-Origin"  *;
        }
        # Preflighted requests
        if ($request_method = OPTIONS ) {
                add_header "Access-Control-Allow-Origin"  *;
                add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, HEAD";
                add_header "Access-Control-Allow-Headers" "Authorization, Origin, X-Requested-With, Content-Type, Accept";
                return 200;
```

Figure 5.2: Nginx Configuration File for Rocket Chat

- **Jitsi** Jitsi is a popular, open-source video conferencing service that is free to use if hosted separately. We are running it on the VM as an individual service like our server. We have added a Nginx configuration file for the server which allows users to access the Jitsi service using the subdomain 'video.curabl.me'.

```
dev@curabl:~$ ls /etc/nginx/sites-enabled/
api.curabl.me.conf  chat.curabl.me.conf  default  video.curabl.me.conf
dev@curabl:~$ cat /etc/nginx/sites-enabled/video.curabl.me.conf
#server_names_hash_bucket_size 64;

types {
# nginx's default mime.types doesn't include a mapping for wasm
    application/wasm     wasm;
}
server {
    listen 80;
    listen [::]:80;
    server_name video.curabl.me;

    location ^~ /.well-known/acme-challenge/ {
        default_type "text/plain";
        root         /usr/share/jitsi-meet;
    }
    location = /.well-known/acme-challenge/ {
        return 404;
    }
    location / {
        return 301 https://$host$request_uri;
    }
}
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name video.curabl.me;

    # Mozilla Guideline v5.4, nginx 1.17.7, OpenSSL 1.1.1d, intermediate configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA
```

Figure 5.3: Nginx Configuration File for Jitsi

5.4.3 Securing the Connections

We have upgraded the HTTP connections to HTTPS by generating X.509 certificates from Let's Encrypt Certificate Authority. Let's Encrypt provides free X.509 certificates for TLS encryption. We have generated certificates for the domain and all the subdomains.

```
dev@curabl:~$ certbot certificates
Saving debug log to /var/log/letsencrypt/letsencrypt.log
The following error was encountered:
[Errno 13] Permission denied: '/etc/letsencrypt/.certbot.lock'
Either run as root, or set --config-dir, --work-dir, and --logs-dir to writeable paths.
dev@curabl:~$ sudo certbot certificates
Saving debug log to /var/log/letsencrypt/letsencrypt.log

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Found the following certs:
  Certificate Name: api.curabl.me
    Domains: api.curabl.me
    Expiry Date: 2022-04-29 04:40:27+00:00 (VALID: 60 days)
    Certificate Path: /etc/letsencrypt/live/api.curabl.me/fullchain.pem
    Private Key Path: /etc/letsencrypt/live/api.curabl.me/privkey.pem
  Certificate Name: chat.curabl.me
    Domains: chat.curabl.me
    Expiry Date: 2022-04-28 15:27:20+00:00 (VALID: 60 days)
    Certificate Path: /etc/letsencrypt/live/chat.curabl.me/fullchain.pem
    Private Key Path: /etc/letsencrypt/live/chat.curabl.me/privkey.pem
  Certificate Name: curabl.me
    Domains: curabl.me
    Expiry Date: 2022-04-24 18:17:00+00:00 (VALID: 56 days)
    Certificate Path: /etc/letsencrypt/live/curabl.me/fullchain.pem
    Private Key Path: /etc/letsencrypt/live/curabl.me/privkey.pem
  Certificate Name: video.curabl.me
    Domains: video.curabl.me
    Expiry Date: 2022-05-14 05:06:09+00:00 (VALID: 75 days)
    Certificate Path: /etc/letsencrypt/live/video.curabl.me/fullchain.pem
    Private Key Path: /etc/letsencrypt/live/video.curabl.me/privkey.pem
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure 5.4: Certbot showing list of certificates

5.5 Integrating third-party services

5.5.1 RazorPay

We are using Razorpay as our payment service provider for our website. It is mainly used for paying consultation fees to the doctor. It accepts Internet payments by credit card, debit card, online banking services, digital wallets and UPI payments. On successful payment, users will get redirected to the success page and their appointment slot will be scheduled with the doctor. Also, they can view their past payments in the “My Payments” tab.

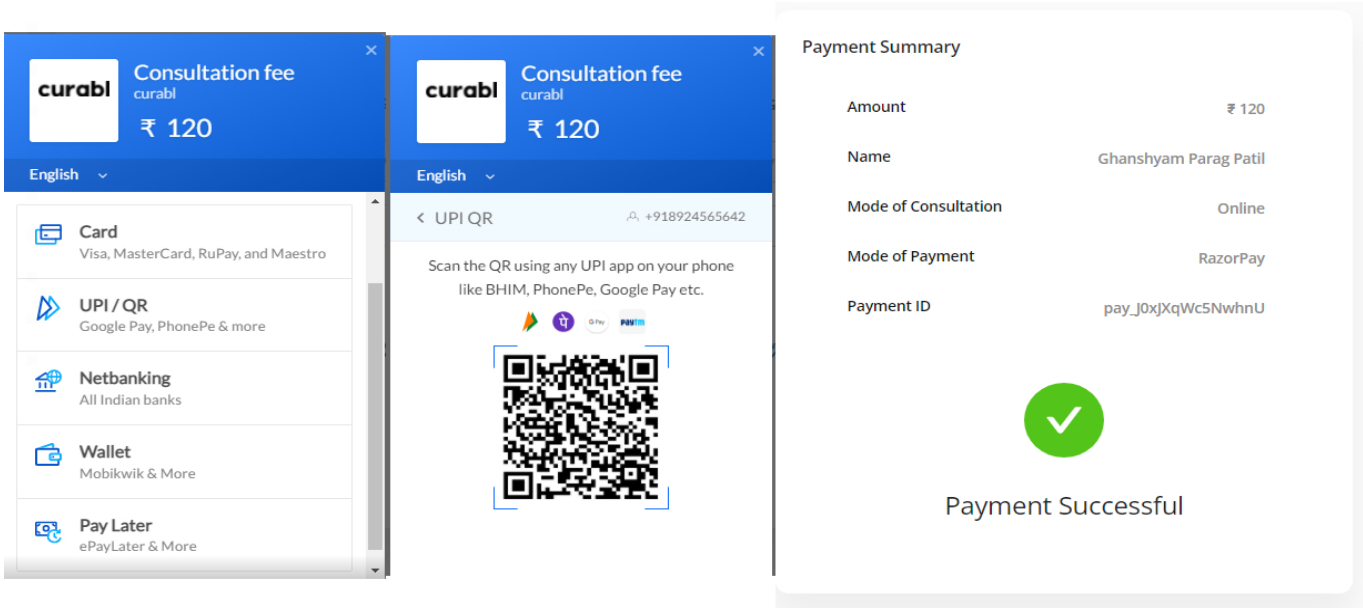


Figure 5.5: Razorpay Payment UI

5.5.2 RocketChat

We have integrated Chat Service into our Web Application using Iframe Integration provided by Rocket Chat. Users can view their previous chats under the Chat Section in our web app. Users can share any video, audio, file or any normal message using the chat service. There is an admin user role in Rocket Chat which can manage all users, their roles, and other important features.

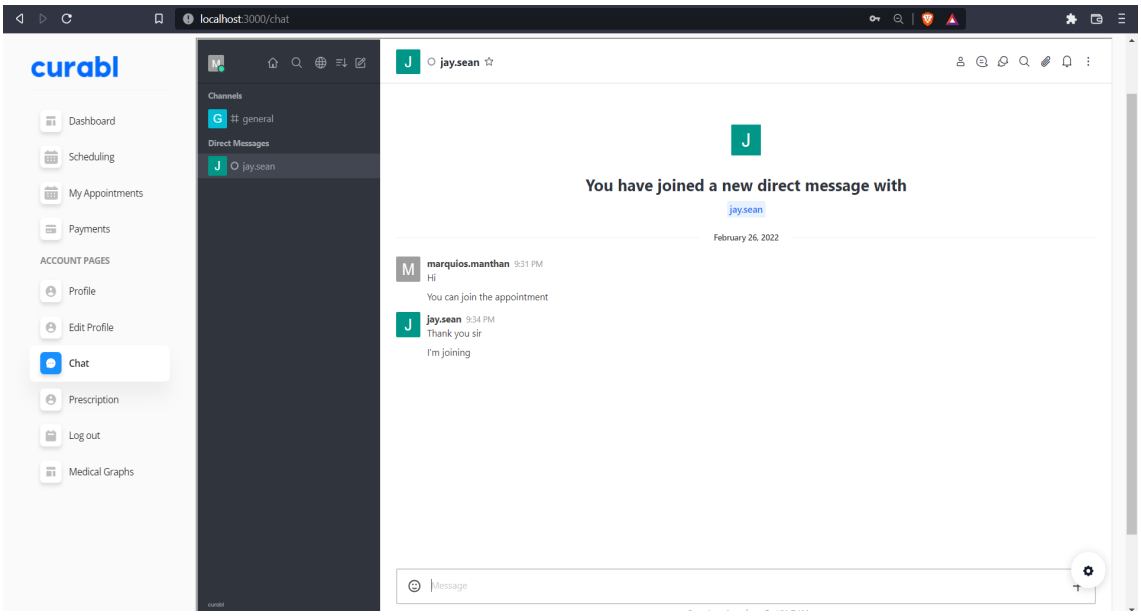


Figure 5.6: Users chatting on Rocket Chat integrated in Iframe

5.5.3 Jitsi

We have integrated the Video Call Service into our Web Application using api provided by Jitsi. Users can join at the appointment slot time and connect over the video conference. Multiple doctors can attend a single conference over discussion regarding diagnosis of a patient. Moderator rights are given through which a user can mute others, remove anyone from meeting or disable camera for others.

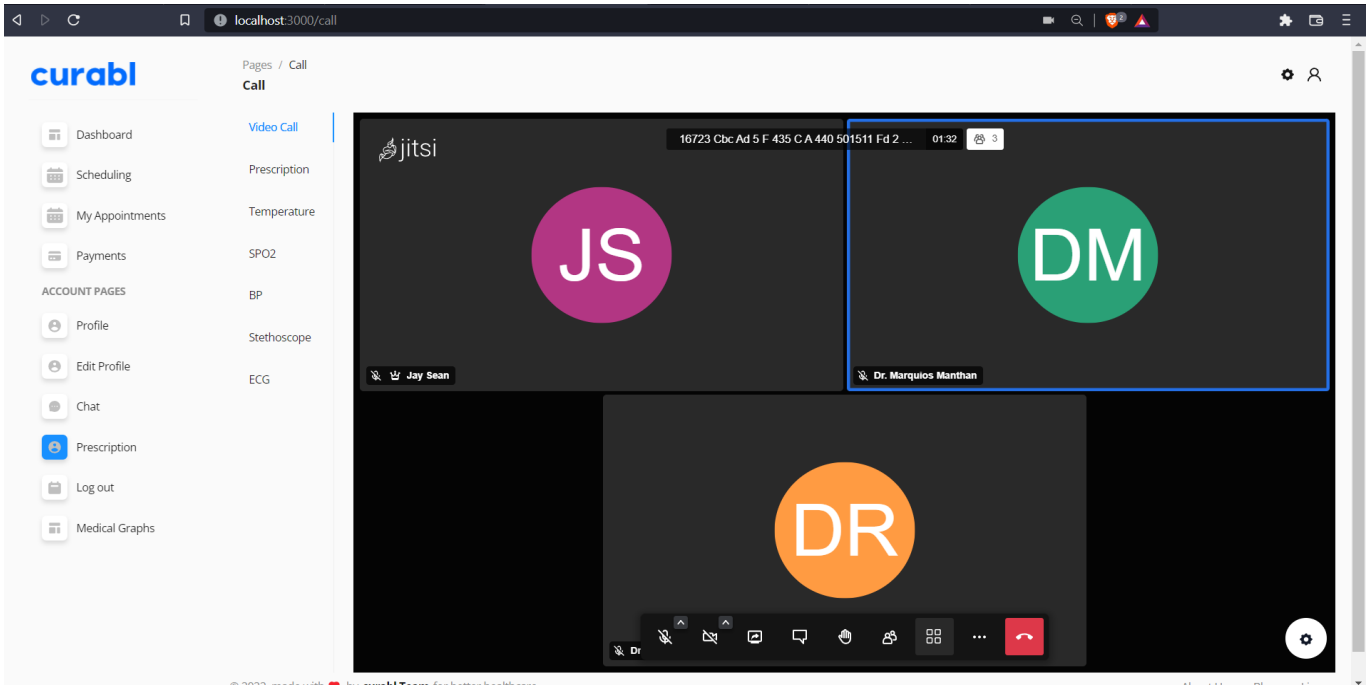


Figure 5.7: Users joined in a Video Conference

5.6 Database

5.6.1 MongoDB

We are using MongoDB Atlas service as our backend database. It is a fully-managed cloud database and it is a No-SQL database. We are storing the data regarding patient, doctor, appointment, payments and other data in this database. Passwords are hashed and stored due to security reasons.

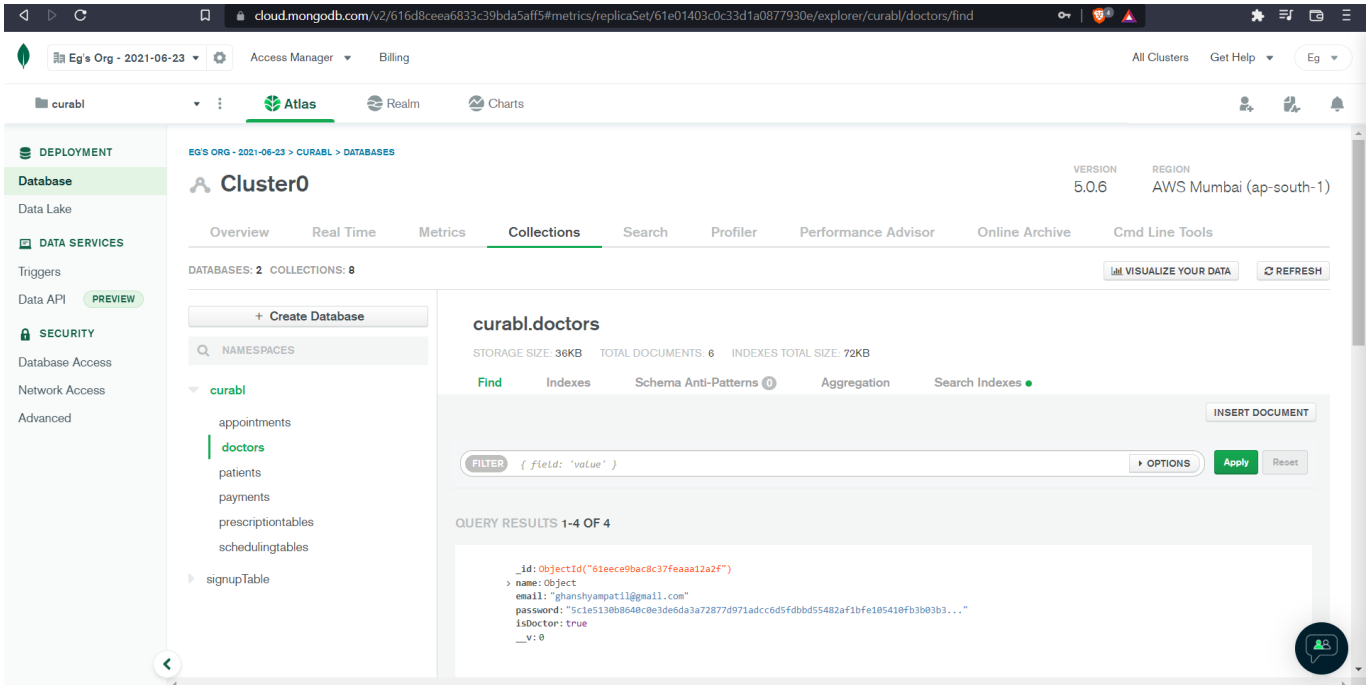


Figure 5.8: Collections in MongoDB

5.7 Development Environment and Tools

5.7.1 Operating System

Server - Ubuntu 20.04 Server Operating System

Client - Any Operating System that can run any web browser

5.7.2 Hardware

Server and Third-party hosted apps are running on Azure Virtual Machine with 2 CPU s and 4GB Memory.

5.7.3 Frameworks Libraries

Client and server are built using Javascript - based Web Frameworks.

Server - Expressjs Framework

Client - Reactjs library and other supporting libraries.

5.7.4 Runtime Environment

The above mentioned frameworks are based on Node Runtime Environment.

5.7.5 Other Software or IDEs used

- Visual Studio Code as Editor,
- Docker as Container Manager,
- Postman as API Tester,
- Developer Tools in Web Browser for testing Frontend / User Interface,
- MongoDB Atlas to temporarily host the database and check the remote connection to database.

Chapter 6

Results and Discussion

Any user, be it patient or doctor can sign up on the platform. The patient can search for any disease or doctor and he will be displayed a list of doctors. They can connect remotely with the doctor at the preferred appointment slot time after booking. The user is provided with the modern payment options to offer a seamless experience. The doctor can prescribe medicines to the patient while in video conference, that the patient can view later. We have attached the screenshots of every module and its respective function.

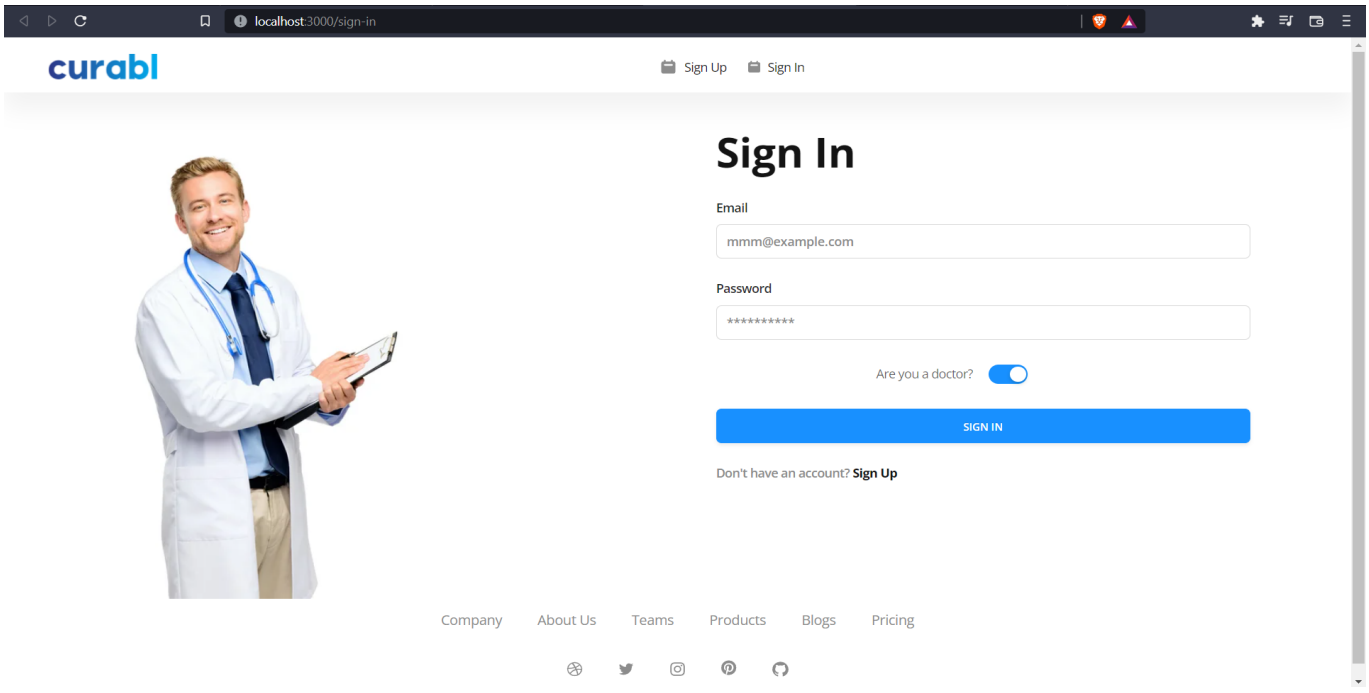


Figure 6.1: Sign-In Page

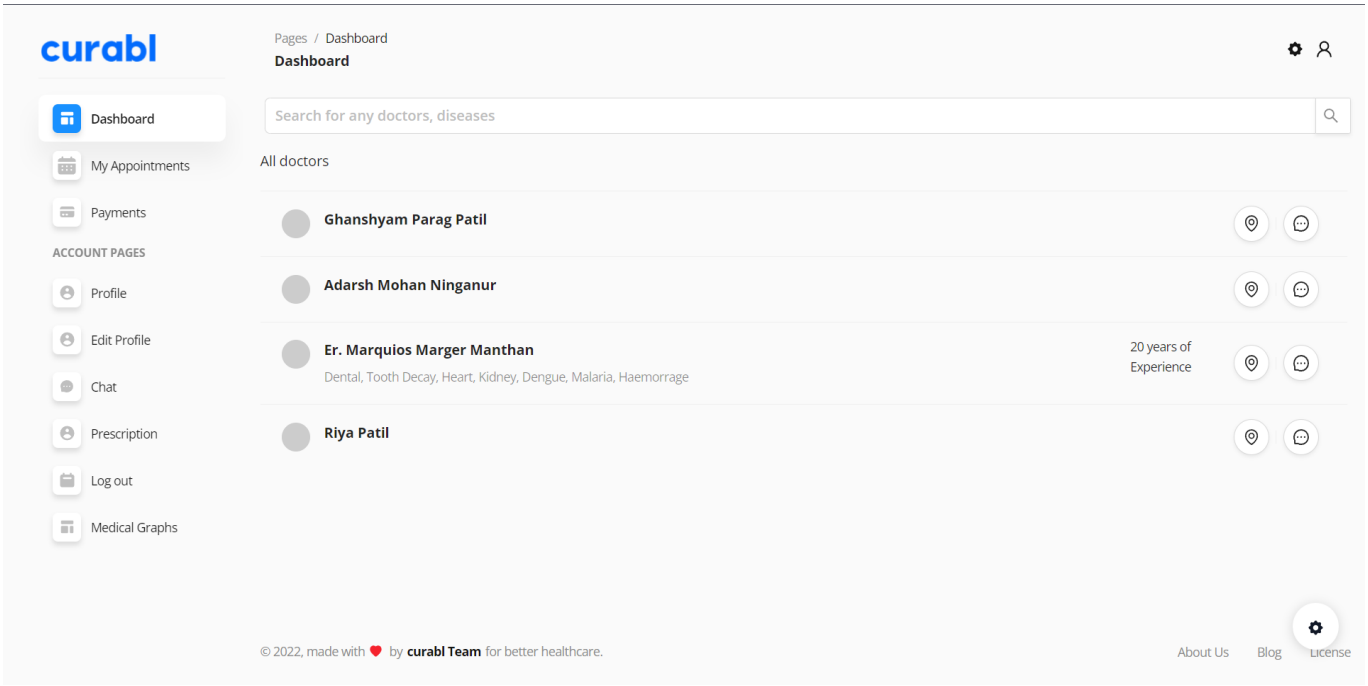


Figure 6.2: Searching for Doctors

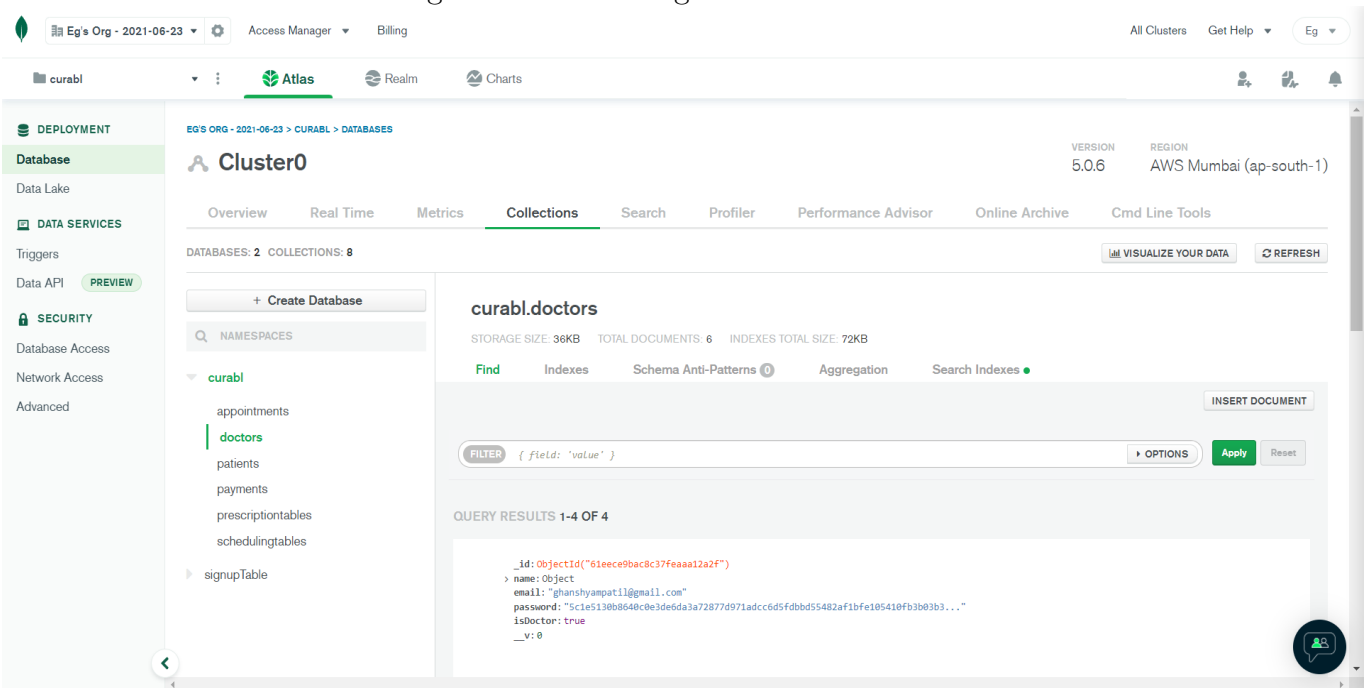


Figure 6.3: MongoDB Database

It is a No-SQL database. We are storing the data regarding patient, doctor, appointment, payments and other data in this database.

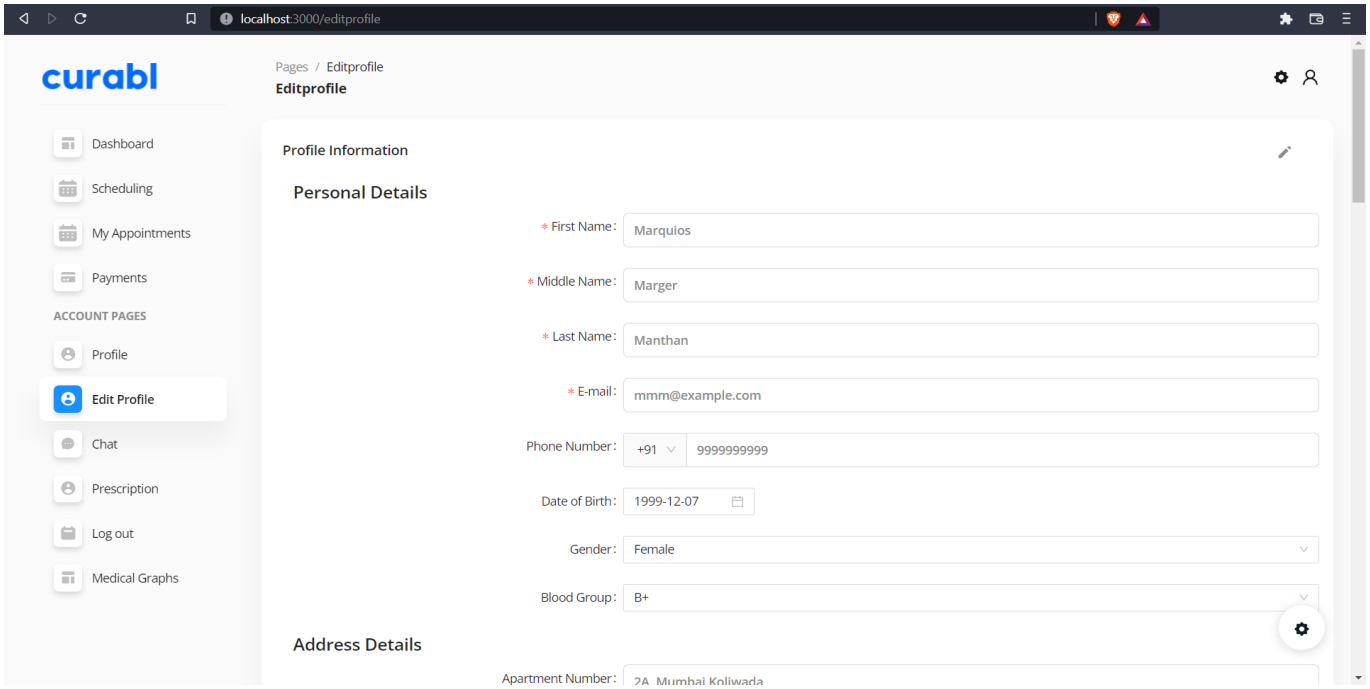


Figure 6.4: Edit Profile

Users can edit and update their profile information.

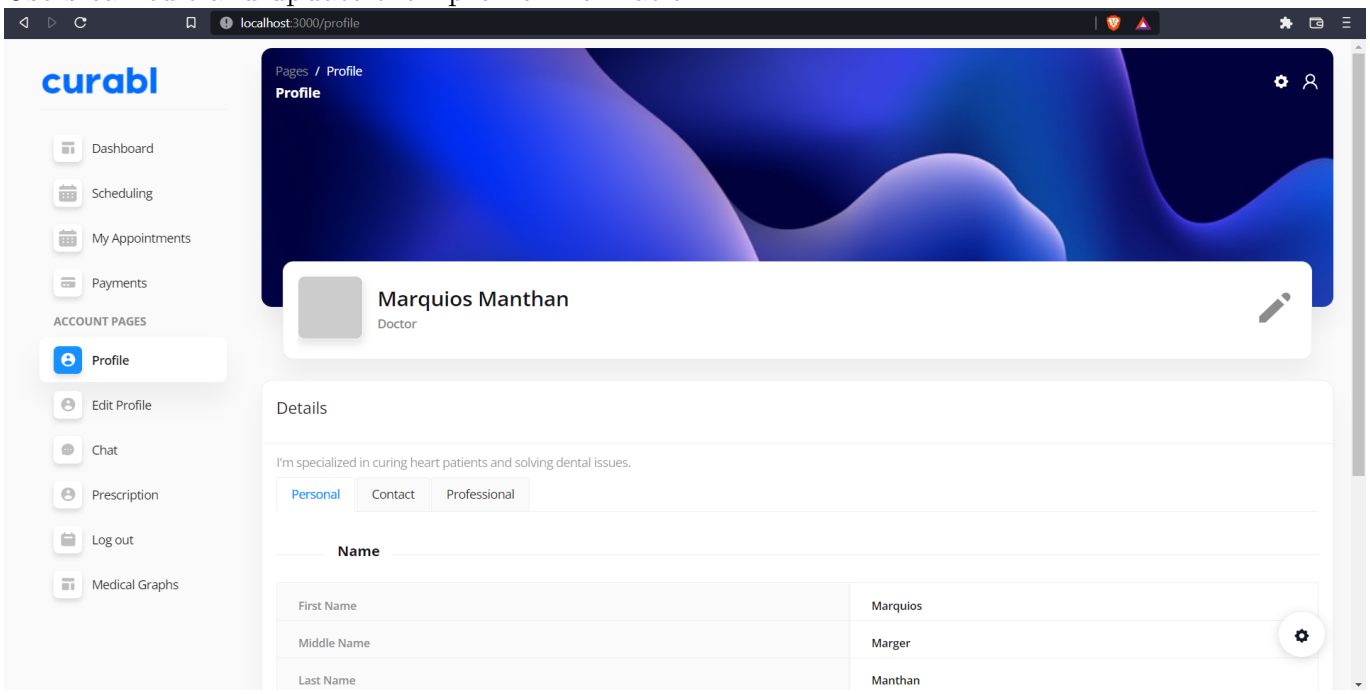


Figure 6.5: My Profile

Users can view their own profile information.

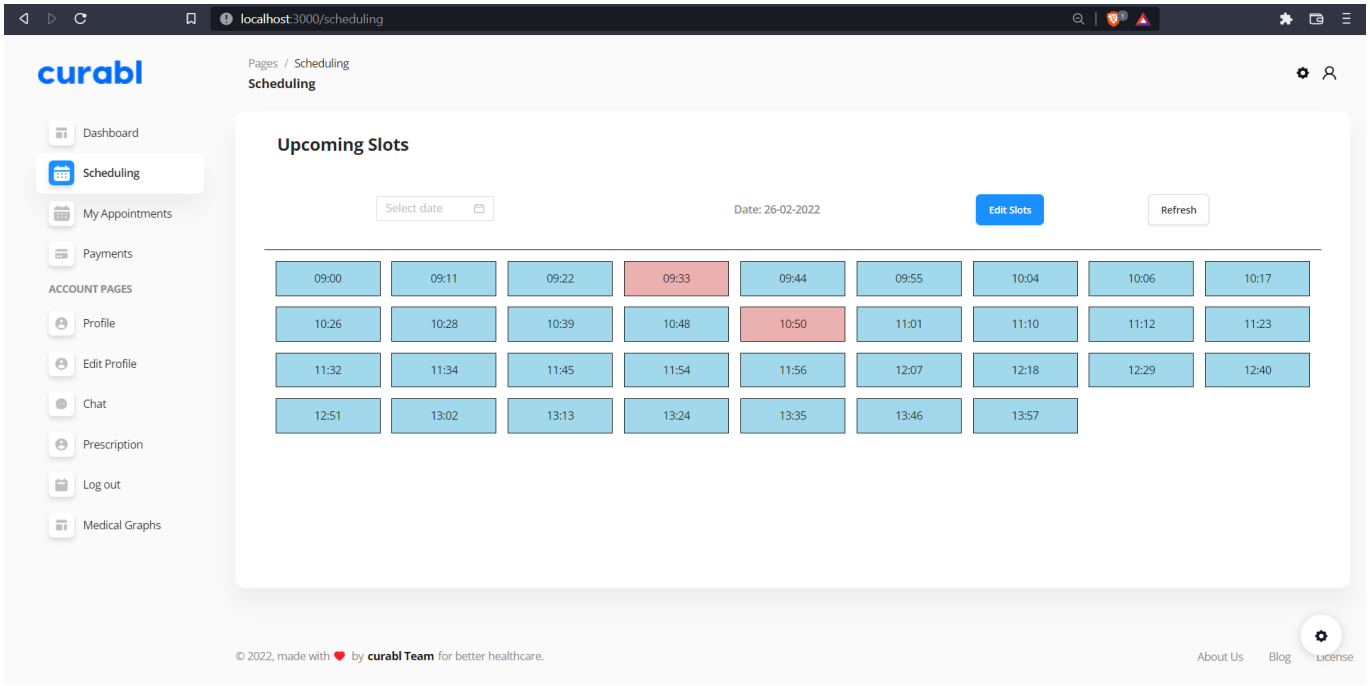


Figure 6.6: View Slots

Red slots indicate that the slot is booked by a patient. Blue ones indicate free slots.

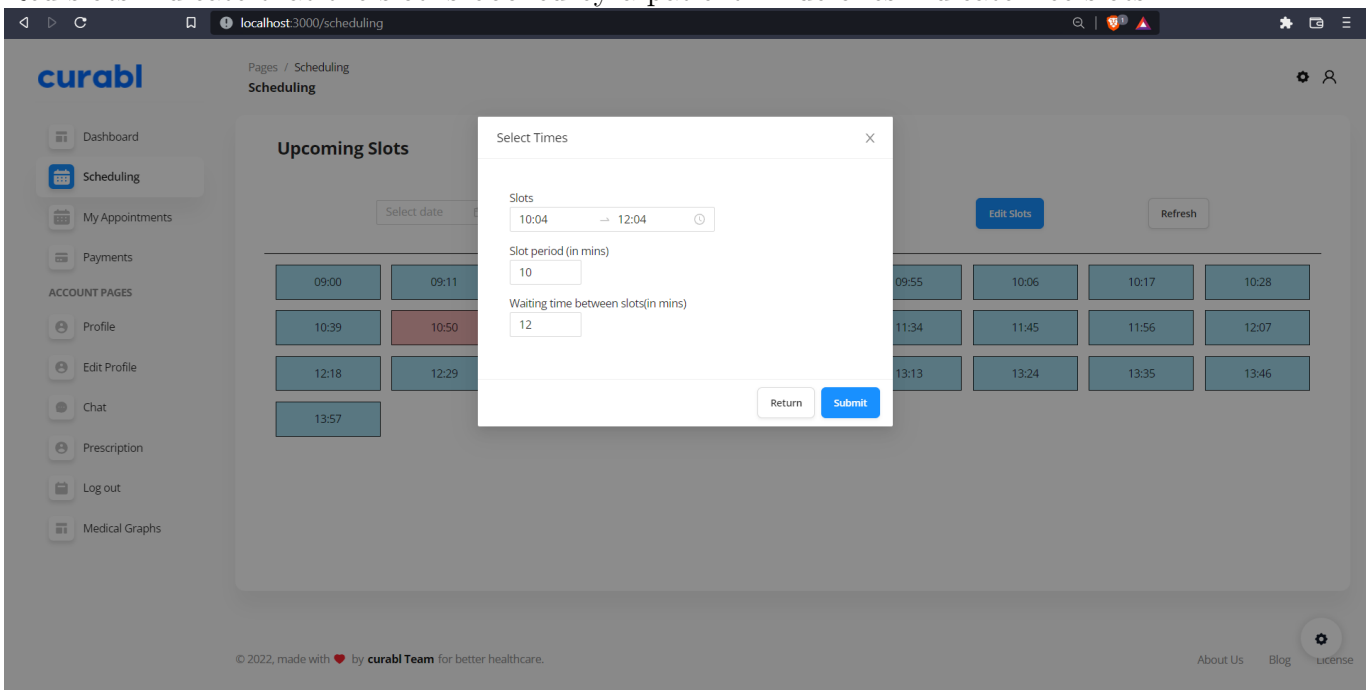


Figure 6.7: Schedule Slots

Doctors can schedule time slots when they are available for appointments.

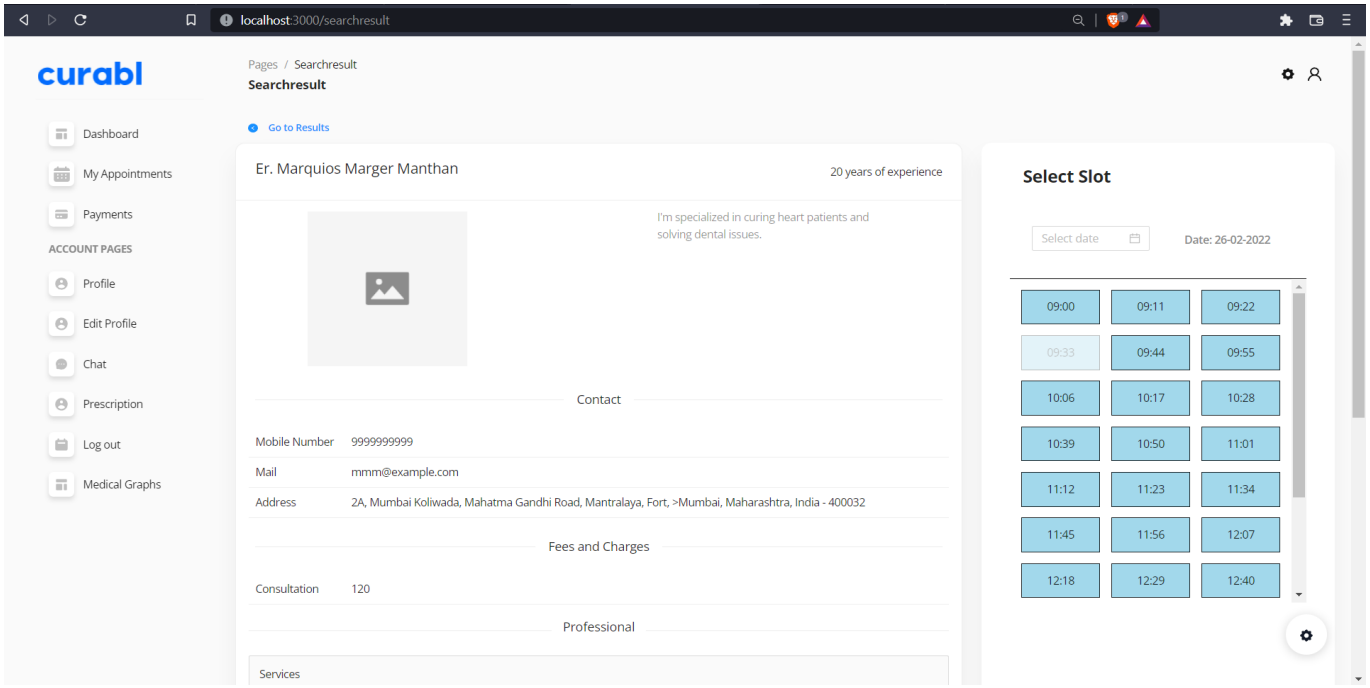


Figure 6.8: Selecting Slots

Patients can view the doctor information on the left side. Patients can book a slot with the doctor for a particular time slot on the right side.

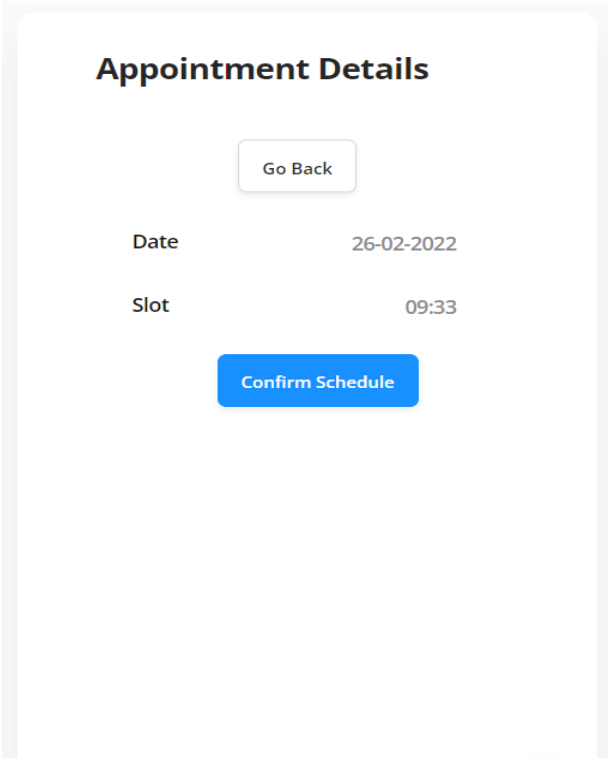


Figure 6.9: Confirm Slot Page

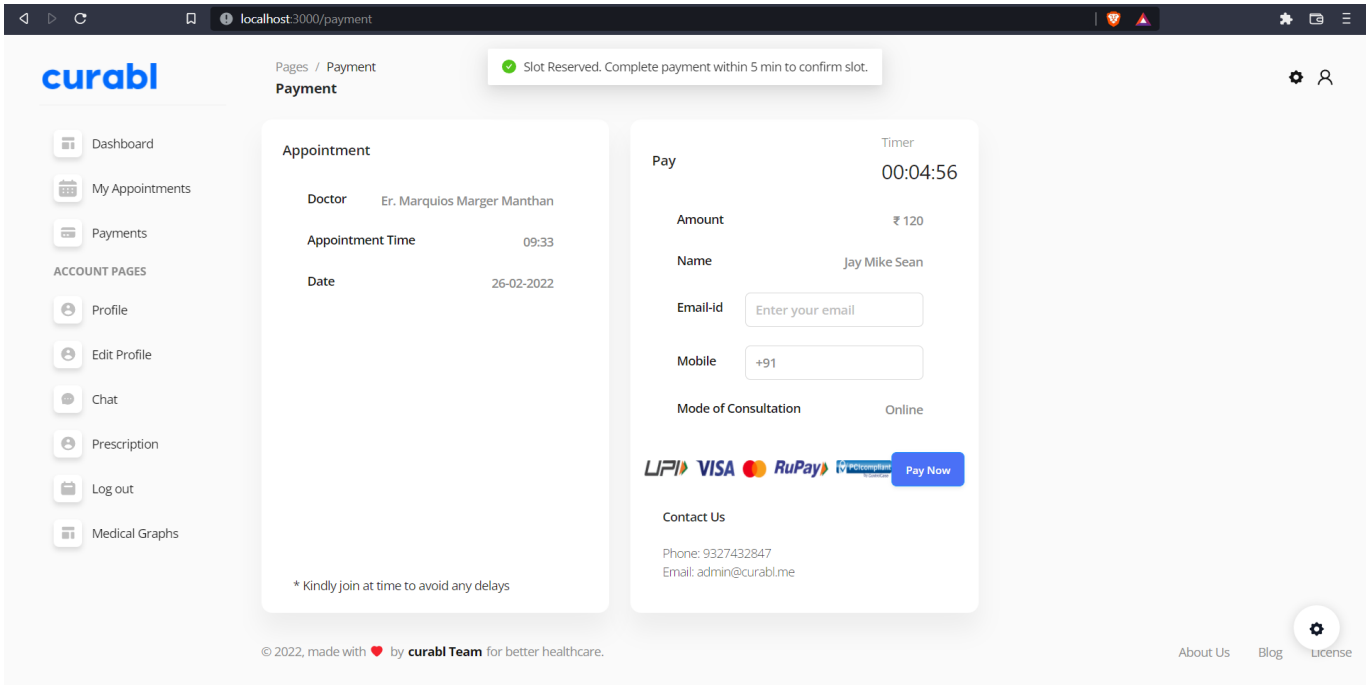


Figure 6.10: Payment Page

Patients can view the Appointment summary and consultation fee.

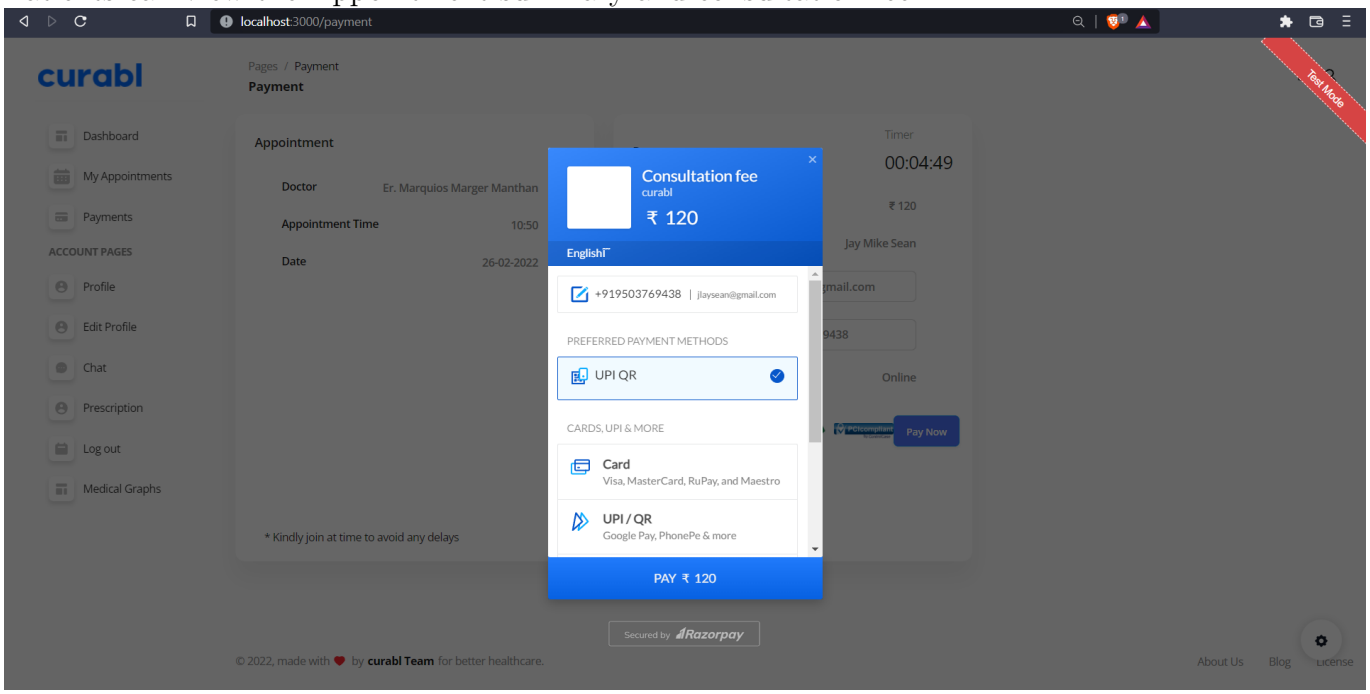


Figure 6.11: Payment Options

Patients can view a number of payment options on the RazorPay pop-up. They can pay by Card, UPI-ID, Scan UPI QR Code, Wallets.

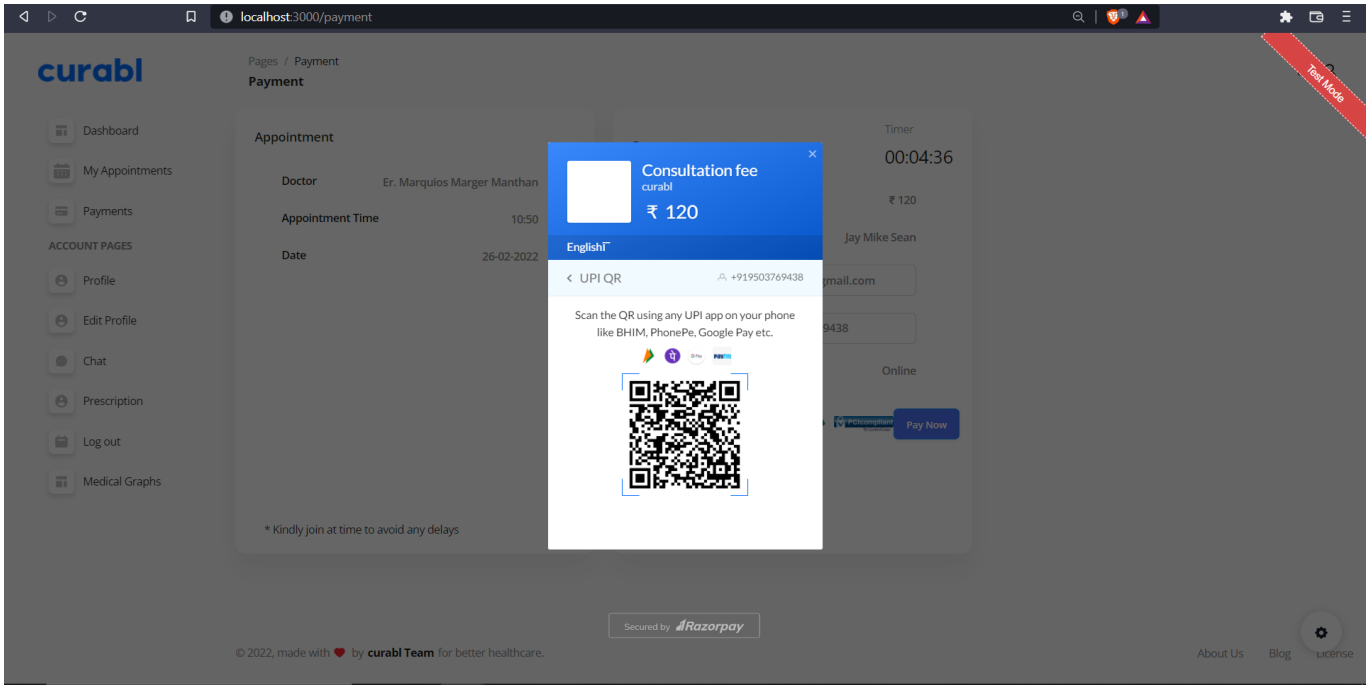


Figure 6.12: UPI Scan

Patients can scan via any UPI-enabled app to pay for it.

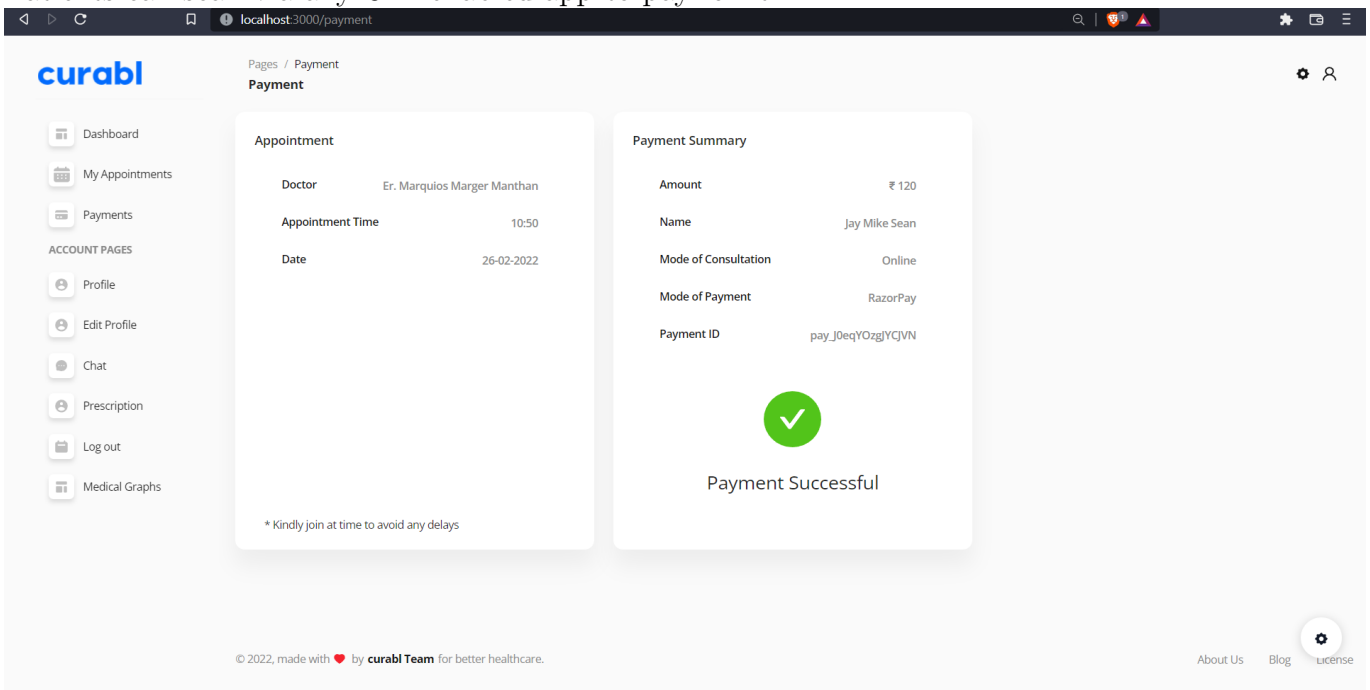


Figure 6.13: Payment Success Page

After successful payment, the patient is greeted by this page.

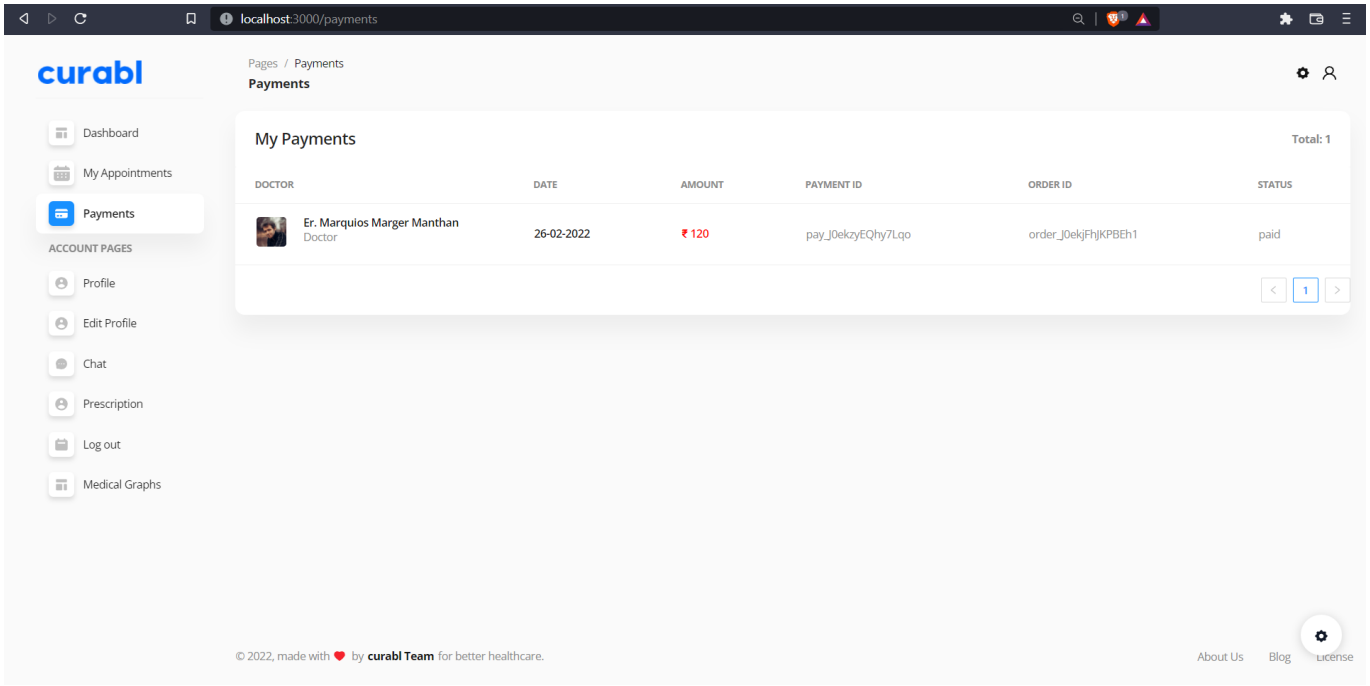


Figure 6.14: My Payments

Patients can view the payments which they have done for various appointments.

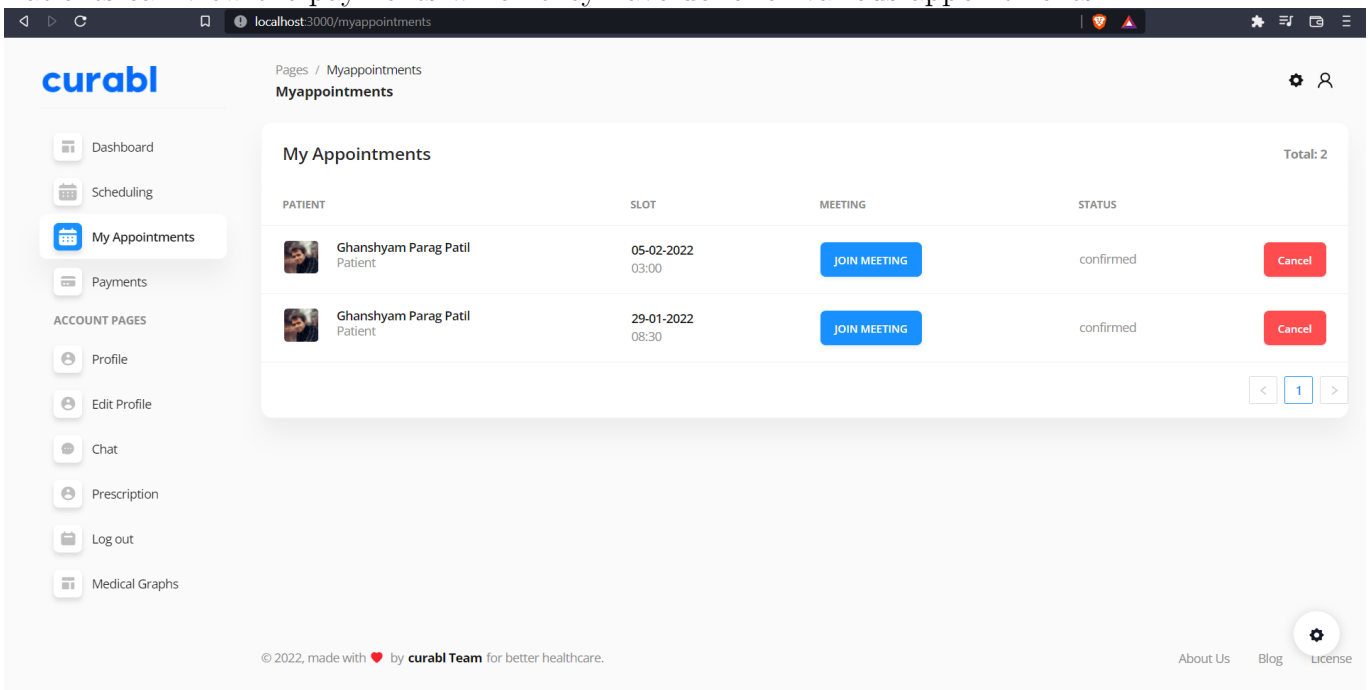


Figure 6.15: My Appointments

Patients and Doctors can join the appointments at the slot time for the video conference with the doctor.

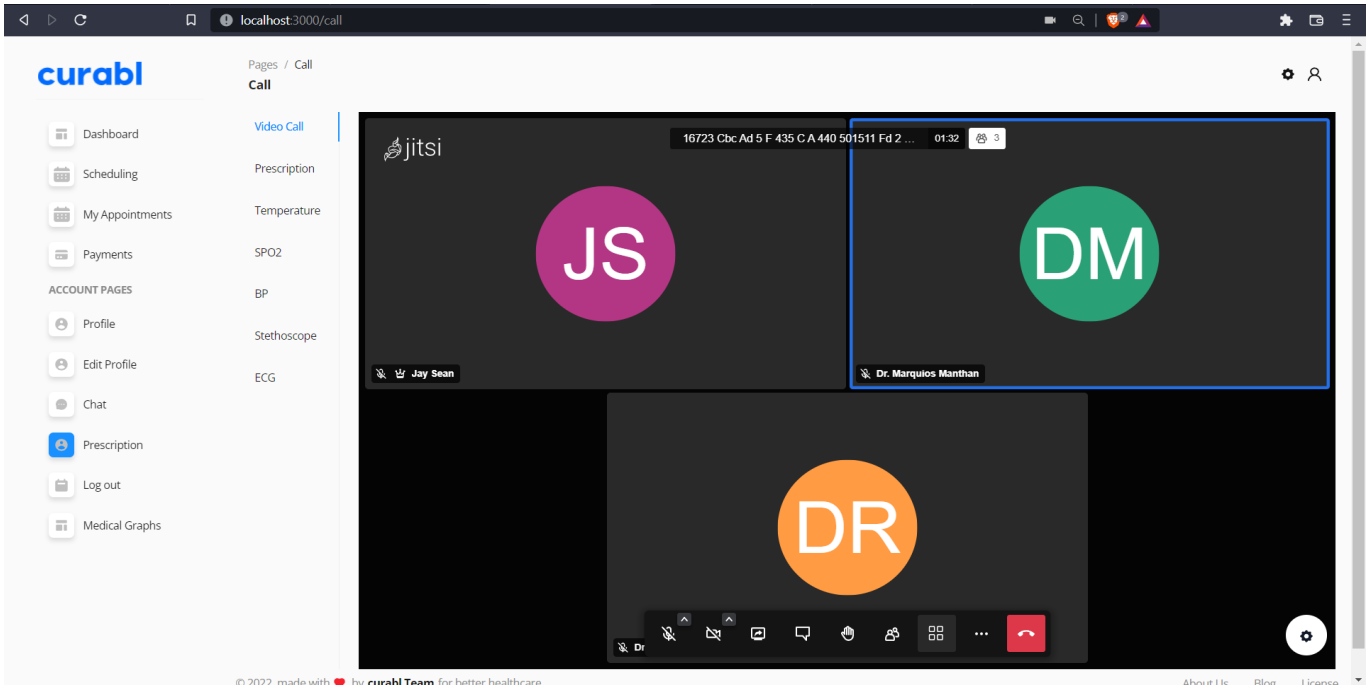


Figure 6.16: Video Call

Doctors and Patients can video chat here for the consultation with JitsiMeet integration.

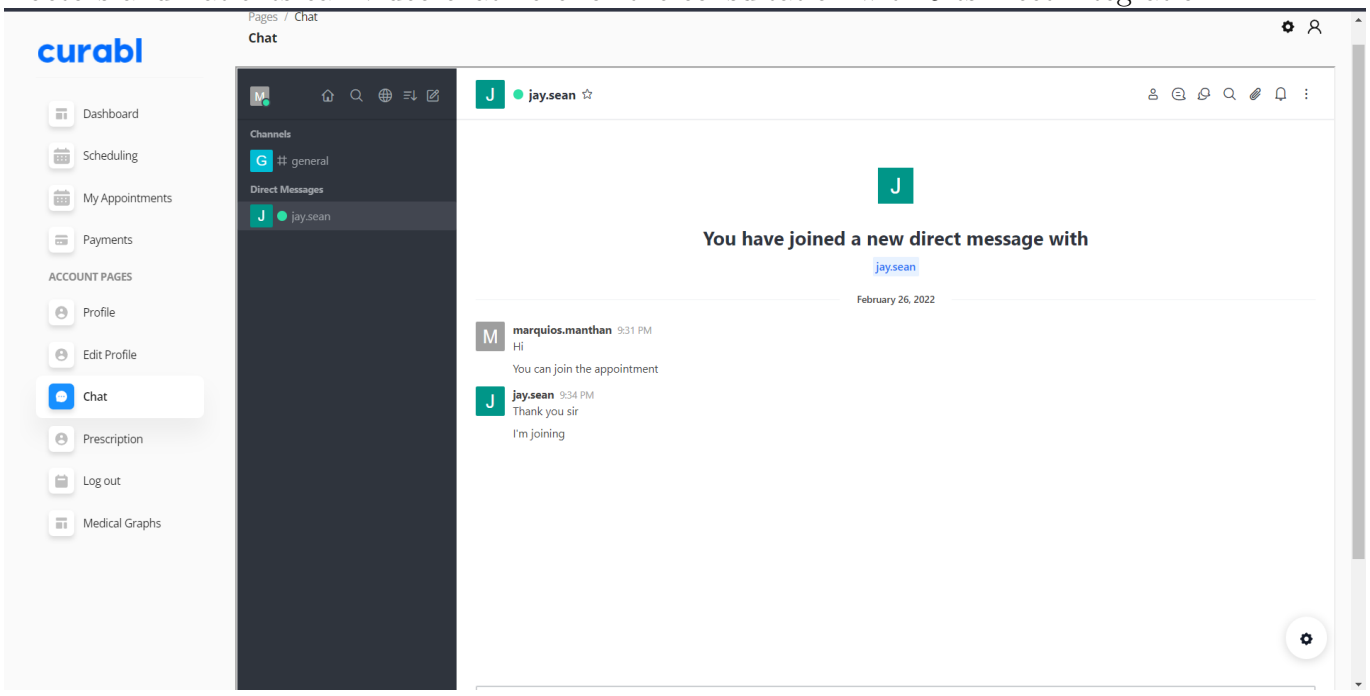


Figure 6.17: Chat

Patients and Doctors can chat here one-to-one with the help of RocketChat integration. Multiple doctors can also group chat to discuss an issue or regarding diagnosis.

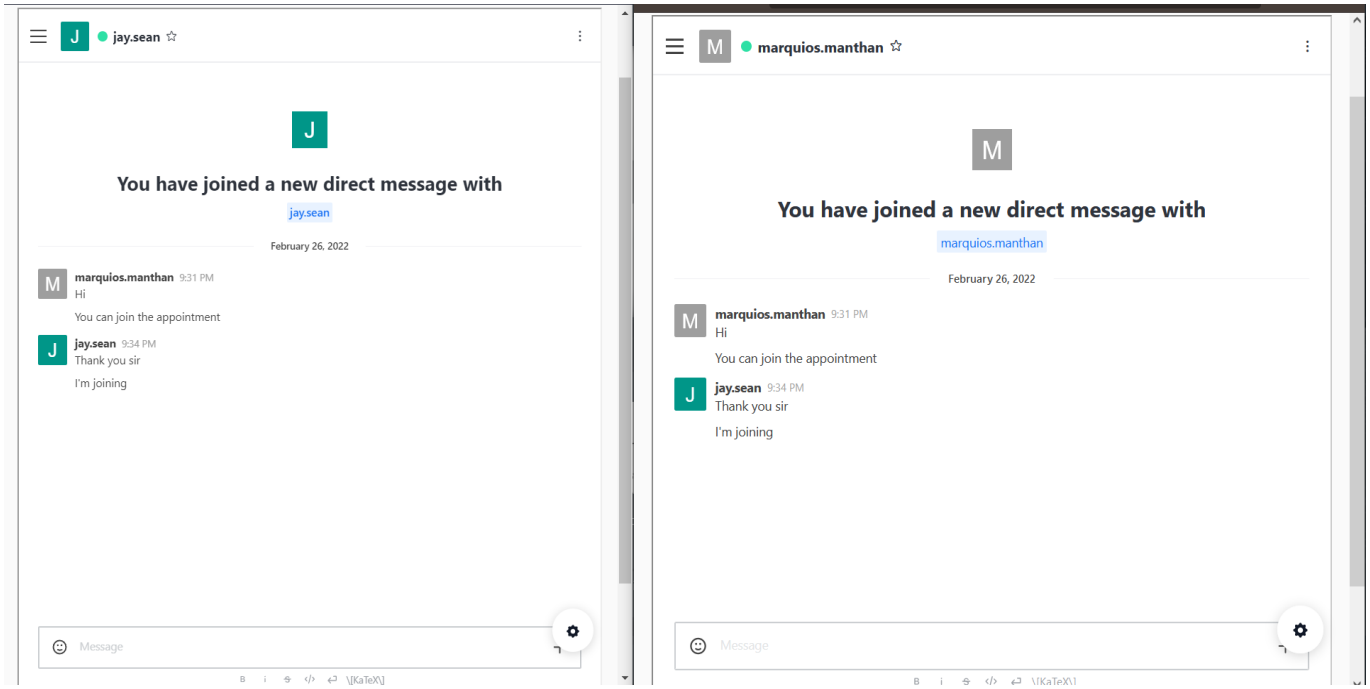


Figure 6.18: Rocket.Chat - Doctor-Patient chatting

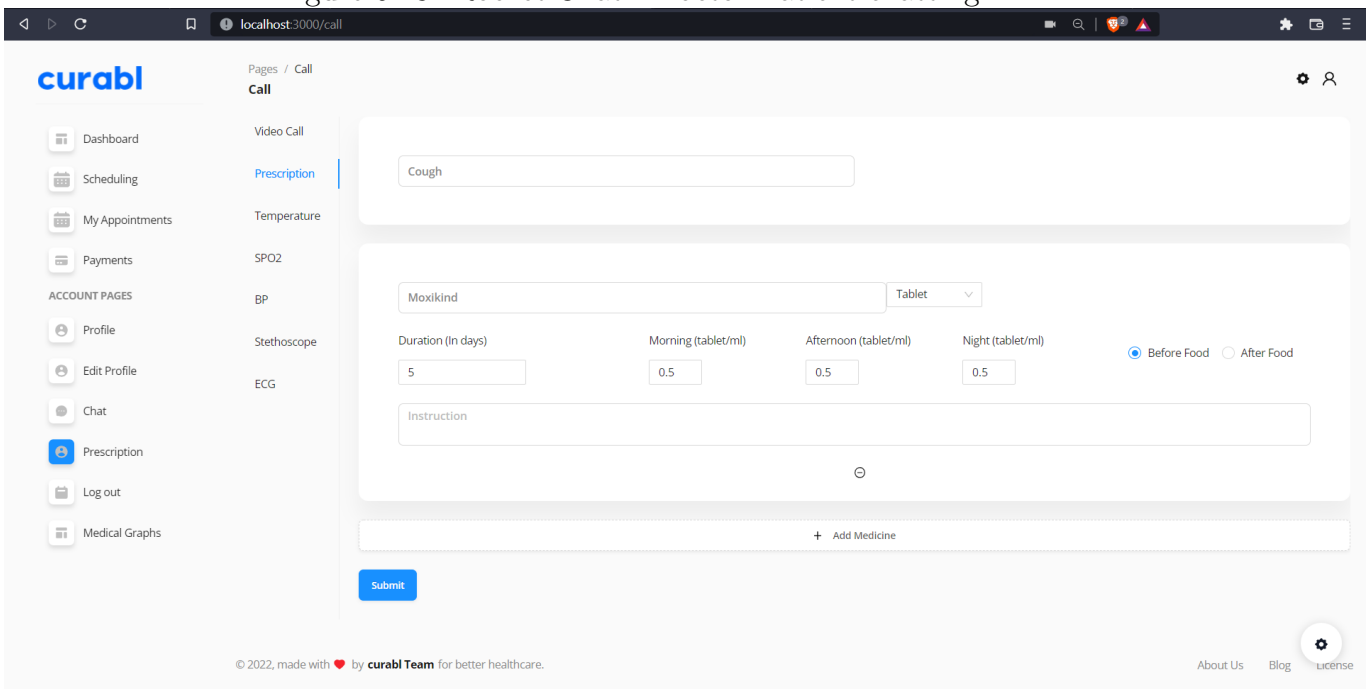


Figure 6.19: Prescription Page

Doctor can prescribe medicines to the patient at the time of meeting.

Chapter 7

Conclusion and Future Scope

After the COVID-19 pandemic, there was a radical change in the field of telemedicine. Many applications provide medical consultation and medicines. However, in rural areas, there is still a problem with people having access to the most basic medical facilities. With our telemedicine application, we aspire to provide the best medical consultation to everyone with the help of a network of leading doctors and specialists.

With the use of the IoT devices, the doctor would be able to monitor the patient's live health data, thus simulating a physical checkup. This includes Heart rate, blood pressure, temperature, oxygen saturation and other data.

The data collected from the use of this software by doctors and patients has wide-ranging applications. It can be used to predict the onset of diseases or to gain insight on which months have a specific disease is widespread. Furthermore, it can be used to manage inventory in hospital pharmacies based on the prescription data by doctors.

The data from symptoms can be used to recommend the doctors to the patient. This would ensure the patient does not have to manually search for any specific doctor or disease and would greatly improve the overall experience.

Moreover, we can visualize data in the form of charts to get a month-wise view of the diseases. A city specific data can also be filtered out to get greater insight of diseases in the area. This could then be shared with the hospitals in the area, so they can be better equipped to manage their medical resources accordingly.

Bibliography

[1] Bokolo Anthony Jnr, "Use of Telemedicine and Virtual Care for Remote Treatment in Response to COVID-19 Pandemic", Journal of Medical Systems, 2020

[2] Lauren M Quinn, Melanie J Davies, Michelle Hadjiconstantinou, "Virtual Consultations and the Role of Technology During the COVID-19 Pandemic for People With Type 2 Diabetes: The UK Perspective", Journal of Medical Internet Research, 2020

[3] Siddharth Angrish, Meghna Sharma, Md Abu Bashar, Shailesh Tripathi, Md Mahbub Hossain, Sudip Bhattacharya, Amarjeet Singh, "How effective is the virtual primary healthcare centers? An experience from rural India", Journal of Family Medicine and Primary Care, 2020

[4] Meenakshi Gautham, M Sriram Iyengar, Craig W Johnson, "Mobile phone–based clinical guidance for rural health providers in India", Health Informatics Journal, 2011

[5] Aleksandar Kotevski, Natasa Koceska, Saso Koceski, "E-health monitoring system", International Conference on Applied Internet and Information Technologies, 2016

[6] Mee Young Sung, Moon Suck Kim, Myung-Whun Sung, Eom Joon Kim, Jae Hong Yoo, "CoMed: a real-time collaborative medicine system", International Journal of Medical Informatics, 2000

[7] Wang Jack Huan, Dr Kuruvikulam Chandrasekaran Arun, "A study and an implementation of online doctor consultation system", Journal of Applied Technology and Innovation, 2021

[8] Hafedh, Ben Hassen, Nadia Ayari, Belgacem Hamdi, "A home hospitalization system based on the Internet of things, Fog computing and cloud computing", Informatics in Medicine Unlocked, 2020

[9] Christos Andrikos, Georgios Rassias, Panayiotis Tsanakas, Ilias Maglogiannis, "An Enhanced Device-Transparent Real-Time Teleconsultation Environment for Radiologists", IEEE Journal of Biomedical and Health Informatics, 2017

[10] Ant Design Documentation - https://ant.design/

[11] Rocket.Chat Documentation - https://docs.rocket.chat/

[12] Jitsi Documentation - https://jitsi.github.io/handbook/docs/intro

[13] RazorPay Documentation - https://razorpay.com/docs/