**Department of Electrical and Computer Engineering**

**North South University**



**Junior Design Project**

**Application Of Blockchain for A Secure Online Telemedicine**

|  |  |
| --- | --- |
| **S M Gazzali Arafat Nishan** | **ID # 1831513642** |
| **Rofiqul Alam Shehab** | **ID # 1831185642** |
| **Shafiul Bashar** | **ID # 1831446642** |
| **Md Asif Hossain** | **ID # 1831737642** |

**Faculty Advisor**

**Dr. Mohammad Monirujjaman Khan**

**Professor**

**ECE Department**

**Fall, 2021**

**DECLARATION**

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

**Students’ name & Signature**

1. **S M Gazzali Arafat Nishan**
2. **Rofiqul Alam Shehab**
3. **Shafiul Bashar**
4. **Md Asif Hossain**

**APPROVAL**

The Junior Design project entitled “**Application of Blockchain for A Secure Online Telemedicine System**” by **S M Gazzali Arafat Nishan (ID#1831513642), Rofiqul Alam Shehab (ID#1831185642), Shafiul Bashar (ID#1831446642),** and **Md Asif Hossain (ID #1831737642)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on May and has been accepted as satisfactory.

|  |
| --- |
|  |
| **Supervisor’s Signature** |
|  |
| **Dr. Mohammad Monirujjaman Khan Professor**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |
| **Department Chair’s Signature** |
| **Dr. Rezaul Bari**  **Associate Professor**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |
|  |

**ACKNOWLEDGMENT**

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report.

The Junior Design project program is very helpful to bridge the gap between the theoretical knowledge and real-life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding.

We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding.

It is imperative to show our appreciation for our honorable faculty member Dr. Mohammad Monirujjaman Khan for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as CSE 299 in which we could really work on this project and materialize it the way we have dreamt of.

We thank our friends and family for their moral support to carve out this project and always offer their support.

We would also thank “Kazi Hasib” for providing the necessary support and advising us during the development of this project.

**ABSTRACT**

In the twenty-first century, telemedicine introduced a fresh revolution to the medical field. The proportion of people aged 18–49 who visited a telehealth center increased from 68% in 2010 to 71% in 2020.Telemedicine made people's lives simpler, raised the likelihood of providing care to the impoverished, and made it easier to reach rural areas where doctors are few. The majority of developed countries have already begun to use telemedicine. In the 21st century, people are open to accepting new technologies as the need for modernization increases every day. As well, as we know, security is the main part of our cyber world. Blockchain technology improves the security of our telemedicine system. Blockchain is a relatively recent development and application. Blockchain software is open source and usually available. This technology will be suitable for the next generation. Five significant benefits of blockchain are enhanced security, greater transparency, instant traceability, increased efficiency and speed, and automation. These benefits fill up our conditions; that's why we adopted the blockchain. The use of encryption, hashing, and maintaining multiple copies of the decentralized ledger across the blockchain will bolster security and reduce opportunistic cyberattacks. Cybersecurity mechanisms to safeguard patient data have yet to be fully implemented in the healthcare industry. Thousands of transactions are made every day in the medical sector of Bangladesh, where security is the main cause of concern. Patient records were exposed in data breaches, and patients faced hassles while making payments. Patient's sensitive data like health and genomic testing records, identity, credit card, and banking information must be secure. The objective of this research paper is to develop a secure telemedicine web application that is integrated with blockchain technology to make the system more secure and robust. This website was created using HTML, CSS, and Python, JavaScript for the frontend and backend, respectively. Django was utilized for the framework's backend. For blockchain, we used the Ethereum framework, as well as the Brownie API. The functionality makes advantage of WebRTC for video calling. The doctor can provide a link to the patients for them to download the prescription, and then the patient can do so. In our research paper, the overall web application will have three different dashboards for the patient, Doctor & admin. Admin can manage all the users of the entire system. Patient & Doctor will be able to communicate through video calling & doctors will be able to examine the condition of the patient from a live video call. Then, doctors can also give suggestions & provide a prescription to the patient. The patient will also find the Doctor's availability and their department. In our system, there will be an option to make payment which will be integrated with the blockchain technology, and the patient will be able to view their bills & medicine costs through our website. Patients will also be able to download the payment slip. Overall, our web application has produced satisfactory results & we have successfully implemented blockchain technology for the payment option while also fulfilling the objective of our research paper to make the system more secure & robust

Keywords: Blockchain, Smart Contracts, Ethereum, Telemedicine, Peer-to-Peer.

**Table of Content**

|  |  |  |
| --- | --- | --- |
|  |  | Page |
|  | **Chapter 1. Introduction**………………………………....................... | 11 |
| 1.1 | Background………………………………………………………………………… | 12 |
| 1.2 | Problem Statement……………………………………………………………......... | 14 |
| 1.3 | Existing Similar System/ Literature Review………………...................................... | 14 |
| 1.4 | Gap of our objective innovation……………………………………………............. | 15 |
| 1.5 | Organization of the paper………………………………………………………....... | 17 |
|  |  |  |
|  | **Chapter 2: Methodology** ……………………………………….......... | 19 |
| 2.1 | Introduction………………………………………………………………………… | 20 |
| 2.2 | Web-Based System…………………………………………………........................ | 20 |
| 2.3 | Block Diagram of the entire System ………………………………………............. | 20 |
| 2.4 | Outline of the ER Diagram for the entire System...................................................... | 22 |
| 2.5 | Use Case Diagram and Analysis…………………………………………………… | 25 |
| 2.6 | Technologies……………………………………………………………………….. | 29 |
| 2.7 | Project Development Approach……………………………………………………. | 31 |
|  | **Chapter 3: Design results and analysis** ……………………………... | 34 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Chapter 4: Conclusion**…………………………………….................. | 51 |
|  |  |  |
|  | **References**…………………………………………………………...... | 54 |
|  | **Appendix: Software Listing**……………………………….……........ | 57 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure No. | Figure caption | Page No. |
| 1 | Block diagram of the entire system | 21 |
| 2 | ER diagram of the entire system | 24 |
| 3 | UML use case diagram | 27 |
| 4 | MVT architecture | 28 |
| 5 | Waterfall model for the project development | 32 |
| 6 | Landing page of the website. | 35 |
| 7 | Patient registration page of the website. | 36 |
| 8 | Patient login page of the website. | 37 |
| 9 | Patient’s dashboard. | 37 |
| 10 | Patient views appointment. | 39 |
| 11 | Patient’s payment slip | 40 |
| 12 | Doctor’s dashboard. | 41 |
| 13 | Admin’s dashboard. | 42 |
| 14 | Video calling & chatting feature. | 43 |
| 16 | Doctor’s prescription. | 45 |
| 17 | Patient downloads prescription in pdf format. | 47 |
| 18 | Smart contract to interact. | 48 |
| 20 | Snapshot of a transaction hash. | 49 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table No. | Table caption | Page No. |
| 1. | Software Comparison with Other Papers | 49 |
| 2 | Comparison table between this paper and others paper | 50 |

**CHAPTER 1**

**Introduction**

**1.1 Introduction**

*1.1 Background:* Telemedicine is described as a video and phone call appointment between patients and doctors. It has created a more likely medical environment in which doctors can readily see patients and provide them with proper medical care and medication, as well as advise if necessary. Our medical system has been upgraded and made it easier to seek care for poor and lower-class families, reducing doctor demand in communities thanks to blockchain technology employed in the telemedicine industry and safeguarding health-related data. As a result, blockchain provides a safe digital environment for the modern world. During the pandemic, governments around the world began to implement more telehealth options, with the majority attempting to do so as soon as feasible. While the number of telehealth contacts has been continuously increasing since 2019, and even before that, the pandemic has boosted access to remote care and electronic health records. The United States (50%) and Sweden (58%) have significant percentages of available online medical activities such as scheduling appointments, refilling prescriptions, and checking tests. ([1] A blockchain is a distributed database shared across nodes in a computer network. The data on the blockchain is retained in a digital format. Without the assistance of a third party, the blockchain always provides security. That is why blockchain technology is so popular. The goal of blockchain is to keep your data secure and disseminated, yet it cannot be modified by a third party. That's how blockchain keeps your data unalterable, which means it can't be removed, destroyed, or changed. Can you believe that 0.5 percent of the world's population is using blockchain? ([2] The usage of blockchain technology is growing every day, which tells us that it will revolutionize our modern telehealth and may bring us revolutionary improvements, which are much needed for the current health system. This COVID-19 circumstance can provide us with some inspiration. Blockchain technology is becoming popular in every sector of technology since it ensures a secure system. In health care, the impact of blockchain technology is essential since it is well embedded in global finance, healthcare, and defense. Thus, the review focuses on the relevance of blockchain technology to the health care sector. For example, In the current COVID-19 environment, clinicians will rely more on telehealth to reduce person-to-person contact. The clinical team can deliver this service with confidence in the veracity of the patient data made accessible through the blockchain platform [3]. Thus, Blockchain can play an essential role in preventing data breaches in health care & telemedicine systems. Similarly, blockchain technology can be utilized to create tamper-proof registries of an implanted medical device in addition to monitoring the provenance of medical devices supply chains [4]. Smart contracts on the Ethereum blockchain are one of the fastest digital technologies which are used in Telemedicine. But Telemedicine can be more efficient when seeking treatment because patients do not need to see a doctor face to face. When using blockchain technology, the stored data becomes more transparent for each node in the blockchain network but has verification on every transaction, taking time and gas costs. Thus, the idea of the decentralized application in telemedicine image record system with the contract on Ethereum. However, blockchain technology is reliable because this system decentralizes information data [5]. Thus, Blockchain is not managed centrally but is instead managed by each user in the network [6]. The main advantage of blockchain systems is the decentralization of data. As data centralization has many vulnerabilities, Blockchain can be a solution. The consensus mechanism will validate the transaction [6]. Security is a major issue in the healthcare industry. In this era only blockchain has the functionality to ensure the ultimate level of security. Telemedicine is an information technology-based health service that allows patients to consult with doctors or other health experts without having to meet [7]. This innovation in health services with the internet can help patients use their time more efficiently because they do not have to come to a hospital or health facility for a consultation. During the remote consultation, the doctor helps the patient to get information regarding the suspected diagnosis, treatment, or first treatment for illness or injury, as well as tips to improve body health.[7]

*1.2 Problem Statement:* In our nation, a large number of people live in villages, and there are fewer doctors than patients. Basically, most doctors reside in the city, so if we have any ailments, we must travel to the city for treatment. Once there, we must wait for a doctor and provide a serial number on the day we wish to meet with the doctors. As a result, we established a telemedicine system. People can take advantage of it to help lessen the country's doctor shortage. Between 2009 and 2017, more than 176 million patient records were exposed in data breaches [7]. However, in the modern era, we can fix those concerns with telemedicine systems, and for security, modern scientists have employed blockchain to record the dialogue and treatment procedure. So, by utilizing this facility to serve the poor and middle-class people, we can solve the problem of a lack of doctors and thereby minimize the number of deaths due to a lack of effective care. In this manner, the entire system can be organized. On the other hand, there is a secure payment transaction facility in our project. Because of information sharing and consistent decision-making, transaction information between entities is open and transparent and cannot be falsified.

*1.3 Existing similar system & literature review:* During the pandemic period, there was a high demand for telemedicine in 2020. The coronavirus has created a massive demand for telemedicine as most people are still reluctant to go to hospitals, fearing contracting the coronavirus. COVID-19 Patients need instant health care service as early as possible. According to the Dhaka Tribune, about 15 digital healthcare service providers are currently working in the country. [8]: Doctors usually provide consultations to their patients by video conference. Several healthcare service providers are working relentlessly on this telemedicine platform. Sebaghar, a telemedicine app, was launched in 2020 to treat patients remotely. Patients can only speak to the doctors and ask for advice and prescriptions. About 75,000 people have used telemedicine services through the Sebaghar app in the last six months [8]. But the security protocol is a significant issue in this application. They didn’t use blockchain in their application to secure patients’ sensitive information. On the other hand, Praava Health, Daktarbari, a telemedicine app, also offers healthcare services via online doctor’s appointment services with facilities for electronic health records. These telemedicine applications can remove hassles but can’t ensure a secure conversation, video call, patient data records, or a secure payment procedure. A blockchain-based application, on the other hand, allows for encrypted peer-to-peer communication. In 2021, Praava Health Care raised $10.6 million to increase access to quality healthcare in Bangladesh, and it comprises about 40% of Praava’s services, including telemedicine. [9]

Research [10] suggests that making electronic health records (EHRs) more efficient and safer is a new model for health information exchange (HIE). As a result of this research, only blockchain was used in our system. For example, in the current COVID-19 environment, clinicians will rely more on telehealth to reduce person-to-person contact. The clinical team can deliver this service with confidence in the veracity of the patient data made accessible through the blockchain platform [11]. Similarly, the following research [11] improves the digital security of sensitive patient data, which is currently a lucrative target for cybercriminals. In the present COVID-19 setting, clinicians will rely increasingly on telemedicine to limit person-to-person contact. [11] This concept is comparable to ours. Three family medicine clinics investigated the influence of blockchain on provider prescribing efficiency using a distributed ledger electronic prescribing platform called Prescription Abuse Greatly Reduced (PAGR) Prescriptions, according to the paper [12]. But we didn't go with that plan. The current level of blockchain research in the field of healthcare is examined in this publication [13]. The goal is to showcase the technology's potential applications as well as the challenges and future directions of blockchain research in healthcare. [13] As a result of this paper, blockchain was implemented in our system. According to the paper [14], the drive of this developing technology has now spread to the medical sphere. While the primary use of blockchain in practice has been to create distributed ledgers based on virtual currencies, the technology's popularity has recently spread to the medical field. [14] This is an idea that is similar to ours. According to research [15], blockchain technology can play a critical role in providing information security and privacy, operational transparency, the immutability of health records, and traceability to uncover fraud involving patients' insurance claims and physician credentials. Our system is based on a similar concept. [15] If we use blockchain in telemedicine, the patient can believe that their payment method and personal data are secure. Blockchain technology is important in developing new systems, such as development tools and backend database systems. Blockchain-based systems are capable of adding, changing, enhancing functionality, features, etc. Blockchain-based systems are capable of handling large amounts of data; the backend and frontend technologies have greater importance for providing accurate, error-free, and frequent data. The Blockchain provides benefits in terms of saving time, minimizing errors, and providing efficiency in economically feasible work. We know about the upsides of blockchain innovation for commercial banks. Blockchain innovation has the potential to reduce exchange costs for both parties while also increasing the working proficiency of business banks and executives’ activities.

*1.4 Gap our objective innovation:* Blockchain technology uses blockchain data structures to validate and store data, as well as distributed node consensus mechanisms to produce and update data, encryption to ensure data transmission and access security, and intelligent script code. It's a brand-new computer and distributed infrastructure paradigm for data manipulation and programming. Since we have lost relatives and friends as a result of this pandemic, this is not the end of it. We've seen people die right in front of our eyes. They were powerless to intervene. Some people are dying because there is no suitable transportation for them while their relatives are crying. People could not handle an immediate doctor to get the right therapy, whether they lived in a village with no doctor for a large area or in the modern day.People have access to every convenience and modern technology, but why should they have to beg for adequate health care? So, based on these circumstances, we attempt to take the first step in developing telemedicine to address these issues. So, based on these scenarios, we attempt to take the first step in developing telemedicine to address the current challenges with blockchain. where people can meet with their doctors and receive therapy without fear of being judged. To maintain confidence, we deployed blockchain technology to protect the system. Encryption, hashing, and numerous copies of the decentralized ledger throughout the blockchain will improve security and lower the risk of opportunistic cyberattacks. However, in the healthcare business, cybersecurity methods to protect patient data have yet to be effectively adopted. A traditional supply chain system is a manual process with insufficient data and transaction security. It also takes a significant amount of time, making the entire procedure lengthy. The undivided process is ineffective and untrustworthy for consumers. If blockchain and smart contract technologies are integrated into traditional supply chain management systems, data security, authenticity, time management, and transaction processes will all be significantly improved [11]. In terms of work organization, the healthcare system has not changed much over the past century. It is simply a matter of time before technology such as blockchain is implemented to provide the highest level of security, immutability, and transparency in the healthcare industry [12]. Since securing data is the key feature of blockchain, this technology can be implemented wherever security is the main priority. In recent years, with the deepening of the global division of labor, the supply chains of modern enterprises have been continuously extended, resulting in the fragmentation, complexity, geographical dispersion, and other characteristic tics that have brought great challenges to supply chain management [13]. According to the DGHS, some of the high-quality telemedicine services have been provided at different levels of hospitals in Bangladesh. But there are only two specialized hospitals (Bangabandhu Sheikh Mujib Medical University and the National Institute of Cardiovascular Diseases), three district hospitals (Shatkhira, Nilphamari, and Gopalganj), and three sub-district hospitals (Pirgonj, Dakota, and Debate). [14] But, these hospitals can provide telemedicine services using only Skype or other social media. So, it is often a complicated procedure to run a telemedicine service. However, these hospitals do not have their own web application for telemedicine.Sometimes, they even use third-party applications that are not secure enough to encrypt patient information, and the payment transaction procedure is less secure. So, our goal is to develop a secure blockchain-based web application that could be a possible solution to overcome this problem in the telemedicine sector. Patients can rely on it to provide a reliable service. Technically, the blockchain is complicated. Some organizations, such as CallHealth, Mediledger, and Embleema, have developed blockchain-based systems to offer trust and operational transparency in healthcare systems. [15] If we use blockchain in telemedicine, the patient can believe that their payment method and personal data are secure. Blockchain technology is important in developing new systems, such as development tools and backend database systems. Blockchain-based systems are capable of adding, changing, enhancing functionality, features, etc. Blockchain-based systems are capable of handling large amounts of data; the backend and frontend technologies have greater importance for providing accurate, error-free, and frequent data. The Blockchain provides terms of saving time, minimizing errors, and providing efficiency in economically feasible work. We know about the upsides of blockchain innovation for commercial banks. Blockchain innovation has the potential to reduce exchange costs for both parties while also increasing the working proficiency of business banks and executive’s activities.[16] Blockchain-based systems have improved healthcare industry services by enabling healthcare professionals to perform business operations in a reliable, trusted, transparent, auditable, trackable, and secure way. Blockchain is a relatively recent development and application.[17]

*1.5 Organization of the paper:* This paper presents the research and making of a secure online telemedicine system that is integrated with blockchain technology. The introduction of the paper has been presented in Section 1. Section 2 describes the method and methodology. Section 3 provides the results and analysis, and Section 4 provides the conclusion of the presented work.

**CHAPTER**

**Methodology**

*2.1 Introduction.*

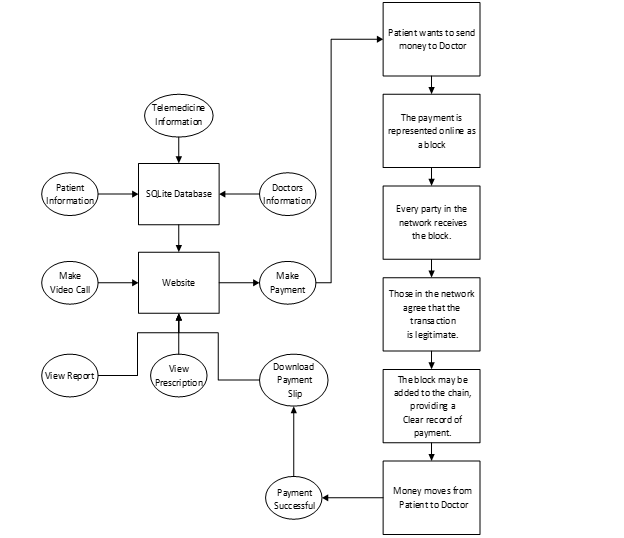
This section goes through the many strategies and materials that were employed to attain the aim. The modeling technique is based on the blockchain notion. When payment operations start and stop, the blockchain keeps track of them. As a result, patients' and doctors' activities might be considered as information services provided to the blockchain architecture. In this way, smart contract design may be viewed as the computation of start and completion times for information services on a blockchain. All materials and techniques, including architectural patterns, system architecture, and a use case diagram for developing the system, are addressed in this part.

*2.2 Web-Based System****.***

This study proposes a website that would include all of the general uses of remote patient telemedicine services. Patient registration, log-in using e-mail and password, accessing and updating patients' information, and viewing and uploading health-related information to the database are all elements of the online application. Doctor and patient registration and log-in, review of the patient's report by the doctor, issuing prescriptions, and video conferencing and calling between the doctor and patient are all aspects of the website.

*2.3 Block Diagram of the entire System****.***

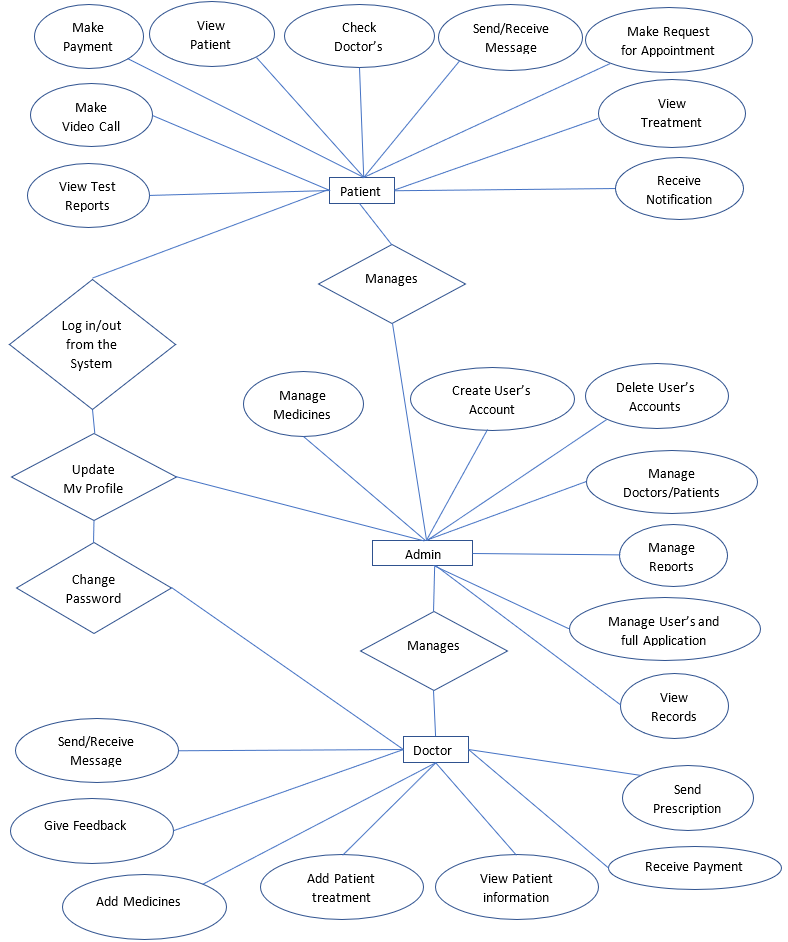
Figure 1, depicts the overall system's block diagram. The system is separated into two components: a blockchain-integrated payment system and a web application. Blockchain will be linked to the website, allowing for a secure transaction between the patient and the doctor. The Web application is linked to the SQLite Database, which stores health-related data, patient information, and doctors' notes. That data is shown again on the website for patient observation through video conferencing. This information is available to doctors in the form of patient reports. Doctors can offer appropriate feedback following the evaluation by issuing medicines. Video calling is an alternative for patients who need to communicate with a doctor in real time but cannot come to the hospital. If a patient requests to pay, he will be sent to a secure payment window. The payment is essentially reflected online as a block. Every time a block is received, the network is updated. Those in the network agree that the transaction's legal block should be added to the chain so that a clear record of payment may be kept. The funds are sent from the patient's account to the doctor's account. Then a notification stating that the payment was successful will appear. After that, the patient can get the payment slip from the website.



**Figure 1:** Block diagram of the entire system.

*2.4 Outline of the ER Diagram for the entire System.*

As we could see in Figure 2, the entire system had been separated into three primary actors, as we could see. The Super Admin is in charge of the entire system and has the most power. All new accounts, such as patients and doctors, are managed by the super admin, who also has the ability to cancel all existing accounts. In addition, the administrator may double-check all of the accounts before they are created. Essentially, the super admin is in charge of the physicians and patients, as well as information such as medicine for patients who have previously been referred to the doctors. Prescription is the most important aspect of the doctor-patient relationship since doctors will prescribe medicine to patients and patients will follow the doctor's instructions. Patients' and doctors' records will be kept on file so that when patients return, doctors may review past reports and assess how their physical condition has improved. The most popular characters in this system are doctors and patients. When the admin handles physicians for the patient's needs and the doctors' accounts are verified by the admin, the virtual link is entered, and the doctor can only view the patient. Because the virtual link is extremely secure and monitored by the administrator, no one else can access it. The doctor will begin by gathering information on the patients since it is necessary to know where they are in terms of illness severity. A doctor can observe patients and communicate with them through chat to learn more about their illnesses. Following that, Doctors would give patients medication and listen to their speech before providing comments. Patients, on the other hand, should create an account and have it verified by an administrator before entering a secure virtual link supplied by the physicians. Before joining the virtual link, the patient must schedule an appointment with the admin for the doctor, who will then email the patient a notification. They can also do a background check on all of the physicians prior to a meeting. Patients may tell their selected doctor anything they want in the chat box or have a conversation with a doctor. Patients can either present or fax their previous prescription to their doctor. According to the physicians, if a patient needs a diagnosis, they may obtain it from their preferred hospital, and they must bring it with them and show it to the doctor before accessing the virtual connection for the second time. Finally, customers must finalize the payment to the doctor after exiting the virtual connection. After then, patients can disconnect from the virtual link and resume their therapy.



**Figure 2:** ER diagram of the entire system.

*2.4 Use Case Diagram and Analysis.*

Use Case Analysis (UCA) is a method for determining a system's requirements as well as the responsibilities and relationships among different classes that will be depicted in a use case diagram.

*2.4.1 Use Case Diagram for Admin.*

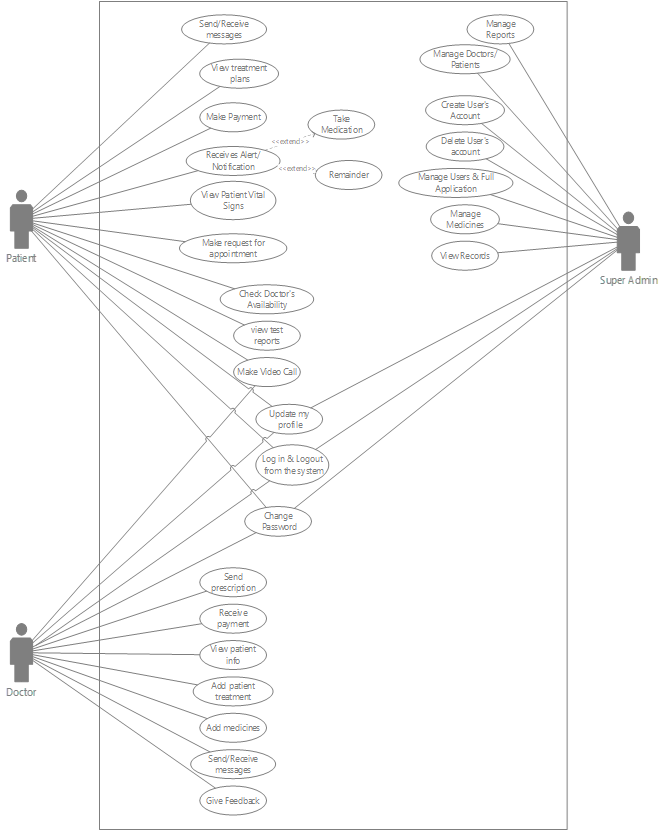
The administrator is in charge of all of the systems. The admin will be assigned to the admins in the system or on the website. All pending registration requests are displayed on an admin's dashboard. The admin can accept a patient's registration because anybody can be a patient. The admin must manually accept a doctor's registration request. The admin can authorize the doctor's registration if the credentials match. As seen in Figure 3, the admin can easily update, delete, or add new users from the dashboard. Admin can check complete information and applications of patients and has the power to collect doctor’s and gather patients.

*2.4.2 Use Case Diagram for Patients.*

Patients must first complete registration before logging into the system in this application. Patients can create an account on the system. Basic information such as the patient's complete name, password, phone number, symptoms, department, and e-mail address are necessary. After creating an account, a patient can log in to the Android app using their e-mail and password. The patient can use an internet connection to collect data from various sensors and save it in the SQLite Real-time Database. As shown in Figure 3, the patient can also log in to the website to examine reports and initiate a video conversation with the doctor. Patients can find doctors who are suitable for their treatment. They can communicate through video calls, use chat boxes, discuss their treatment plans, receive doctor's prescriptions, and have them mailed to them when they return. Patients can also display their essential symptoms to the doctor, as well as their reports.

*2.4.3 Use Case Diagram for Doctors.*

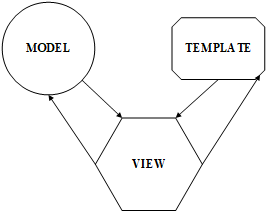
Doctors must go through a more rigorous registration process. The name and department are used to register as a doctor along with other important information such as e-mail, password, degree, and visiting hours. As seen in Figure 3, the doctor can also initiate a video call with the patient. Doctors can use this system to log in with their credentials, change their passwords, and perform services such as sending or receiving prescriptions, administering therapy, and adding necessary medication. Doctors have access to the patient's data. This system gives them a distinct advantage. It has the potential to assist them in collecting their own funds, and it is a safe system based on Blockchain. As a result, they will be able to refer to or view the patient's reports.



**Figure 3.** UML use case diagram of the entire system.

*2.5 Architecture of the system.*

Python is used to create the online telemedicine web application. Django develops web applications using the Model View Template (MVT) software design paradigm. Below is a thorough description of the architectural pattern.



**Figure 4.** MVT architecture of the system.

*2.5.1 Model.*

The website's UI is the model. It is in charge of data coordination and maintenance. The model contains the logical data structure that is used in the backend of the program. A database that depicts a relational database in general is also used in the model. Django models make jobs easier to complete and arrange tables into models. In most cases, each model corresponds to a single database table. This article focuses on how to utilize Django models to store data in a database with ease. Furthermore, we may utilize Django's admin panel to add, edit, remove, or retrieve model fields, among other things. Simplicity, consistency, version control, and comprehensive metadata management are all features of Django models.

*2.5.2 View.*

Python The view serves as the web application's user interface (UI). When a user browses a website, the view's principal function is to render everything. HTML, CSS, and JavaScript are used to represent the view (JS). Django Views are an important component of the Django MVT Structure. According to the Django documentation, a view function is a Python function that receives a web request and delivers a web response. This answer might be the HTML content of a web page, a redirect, a 404 error, an XML document, an image, or anything else that a web browser can display.

*2.5.3 Template.*

Python is used to create the online telemedicine web application. Django develops web applications using the Model View Template (MVT) software design paradigm. Below is a thorough description of the architectural pattern. Django is mostly a backend framework; thus, we utilize templates to offer a frontend and a layout for our website. Depending on our demands, there are two ways to incorporate the template to our website. We can utilize a single template directory that will be distributed throughout the project. We may make a separate template directory for each app in our project.

*2.6 Technologies.*

* + - * Front-end HTML, CSS, SCSS, JavaScript, Bootstrap, jQuery
      * Back-end languages: Python and Solidity.
      * Framework: Django, Ethereum, and Hyperledger.
      * SQLite database is used.
      * Operating Systems: Windows OS, Mac OS, Linux, and Android OS.
      * Browsers: Google Chrome, Mozilla Firefox, Yandex, Safari.
      * Code Editor: VS Code
      * MS Visio Pro, MS Word, and MS Excel are all included.
    - Blockchain

*2.6.1 Frontend Tools.*

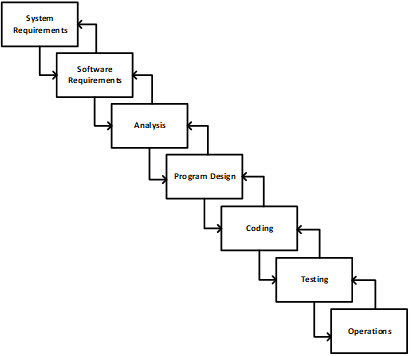
The website's front end is an HTML5 template. Later, this template was included in a telemedicine website. The frontend of this website is built with HTML, CSS, JavaScript, and jQuery. "HTML stands for Hypertext Markup Language. It is a programming language. It serves as the foundation for the entire website. With the use of numerous tags, users may develop and arrange their own web apps. CSS is utilized to give the website a more alive appearance. Each button's behavior is specified using JavaScript, a programming language. A JavaScript library is jQuery. The goal of jQuery is to make it easy to utilize JavaScript by supporting it. The frontend development was likewise done with Bootstrap.

*2.6.2 Backend Tools.*

Backend technologies were employed to make the website dynamic. The website's backend allows it to execute a variety of functions. It also organizes and displays all the web pages on the website, allowing users to view, send, and edit data. The Django framework is used to create the website, which is written in Python. Django was picked because of its efficiency and scalability. Django structures and stores data using its SQLite relational database. At first, WebRTC technology and the PeerJS library were used for video calling, and Node.js and Socket.IO were utilized for video chatting. However, we discovered several problems that we were unable to repair, causing our server to crash. So, for video chatting, we switched to Google Meet because it is now more stable and quicker than WebRTC. It's simple because the majority of consumers' mobile phones are now Android. Google Meet is supported on Android. Android is presently used by the majority of consumers. On the other hand, we deployed blockchain for decentralization in our project. The backend is built with Solidity and Solidity compilers. We've also implemented a blockchain using the Ethereum framework. The payment gateway makes use of smart contracts.

*2.7 Project Development Approach.*

Our main goal is to develop our blockchain project by following a specific software development lifecycle. We have found that a software development approach is really suitable for our project, depending on some software requirements and factors. So, we have selected a software process model for the development strategy to achieve our goal. This goal can be achieved by abiding by some constraints. For example, the iterative waterfall model is a great model to achieve our goal. This model is really user-friendly and suitable for understanding the software designing model. Due to the high rigidity of the iterative waterfall model, it’s easy to manage. In this model, most of the phases are followed one time, and they don’t overlap again and again.

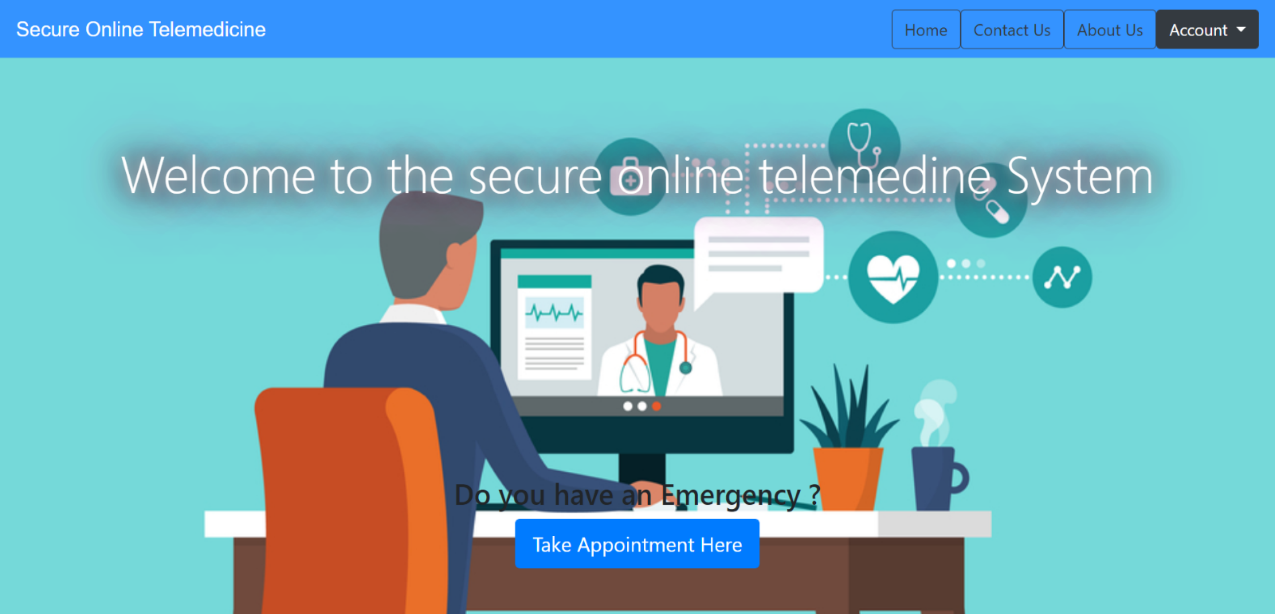


**Figure 5.** Waterfall model for the project development of the system.

The goal of the requirement analysis is to make the user's and patient's requirements comprehensible, as well as to appropriately record such requirements. The initial step in our project is to gather as many needs as possible. Each of the requirements must be examined. During the data collection and analysis process, the customer's needs should be recognized. A software developer may rapidly determine an organization's requirements by creating a software requirements specification. The design phase's goal is to convert the requirements described in the SRS document into a structure that can be implemented in a programming language. In other words, the software architecture is replicated from the SRS document during the design process.

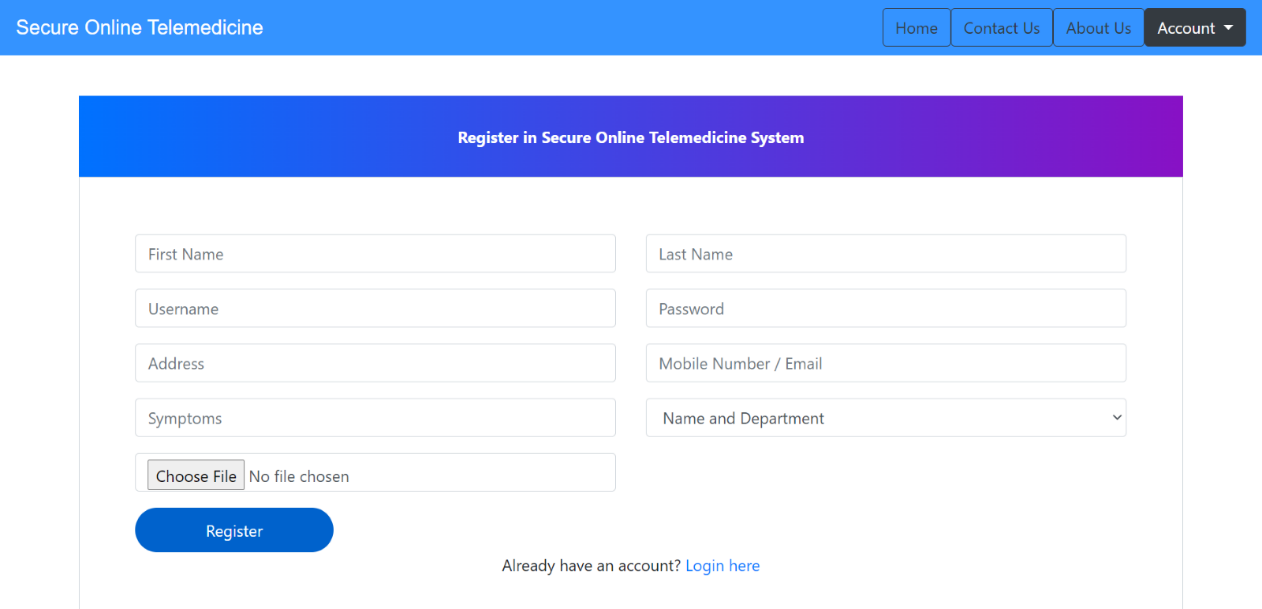
The object-oriented design method (OOD) is a relatively recent methodology. Numerous items that appear in the issue domain and the solution domain are initially recognized, followed by the various connections between these things. The goal of software development coding is to convert the program design into source code. Because the concept is transformed into a usable solution, the coding step is also known as the implementation phase. Each design element is implemented as a separate software module. The ultimate result of this step is a collection of independently tested software modules. Every software development firm often develops its own coding standards to allow engineers to produce high-quality programs. A coding standard covers topics like conventional ways of putting out computer code, a template for laying out functions and variables, module headers, commenting rules, and function name conventions, among other things. A system test is normally carried out in a certain manner according to a system test plan document during the testing process. Maintaining traditional software may be difficult at times, as it needs far more effort than developing a web application. As a result, our system may have the option of having a super admin manage it on a regular basis.

**3 Results and Analysis:**



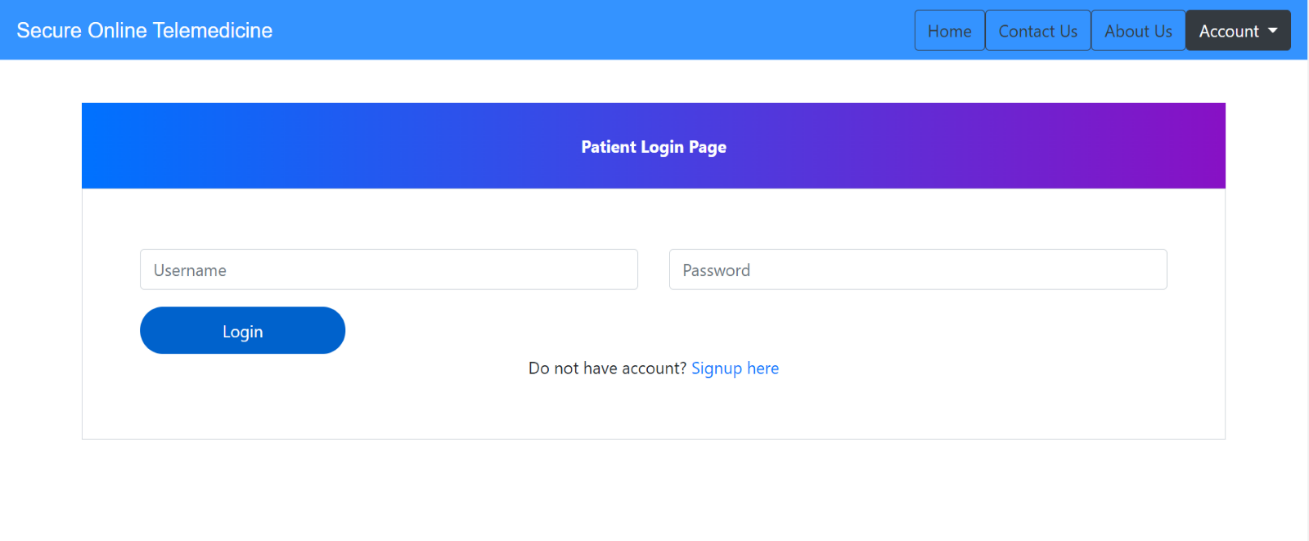
**Figure 6.** Landing page of the website.

The website's landing page is seen in Figure 6. This page is open to the public. The main page gives basic information about the services offered by this website. Appointments, telemedicine, and online prescriptions are all part of the program. The doctor and admin buttons are located in the right corner of the website's navbar inside the account button. The patient "login" and "signup" buttons are located on the right side of the navigation bar inside the account button. The user is transported to a new web page after selecting the "About Us" button, which contains further information from the system administrator. The "Contact Us" page contains important contact information.



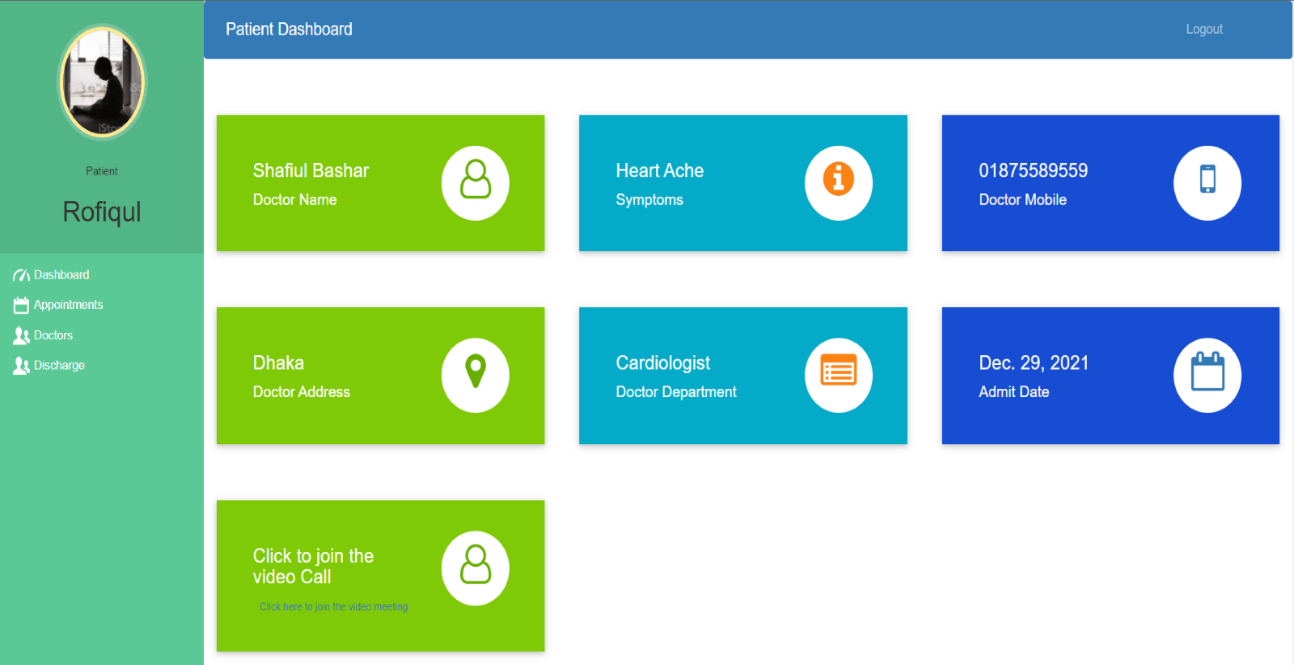
**Figure 7.** Patient registration page of the website.

Figure 7 shows the registration page of the website. There are "signup" and "login" buttons in the upper right corner, inside the account button. Users select their responsibilities initially on the registration page. Users have the option of registering as either doctors or patients. Users must give information such as their complete name, last name, username, e-mail/phone number, address, symptoms, doctor's name and department, profile avatar, and password in order to register. Doctors must wait for authentication when enrolling. Both the Doctors and the patients will not be able to use the system unless they have received administrative authorization. Patients and doctors will be returned to the home page after completing the registration process.



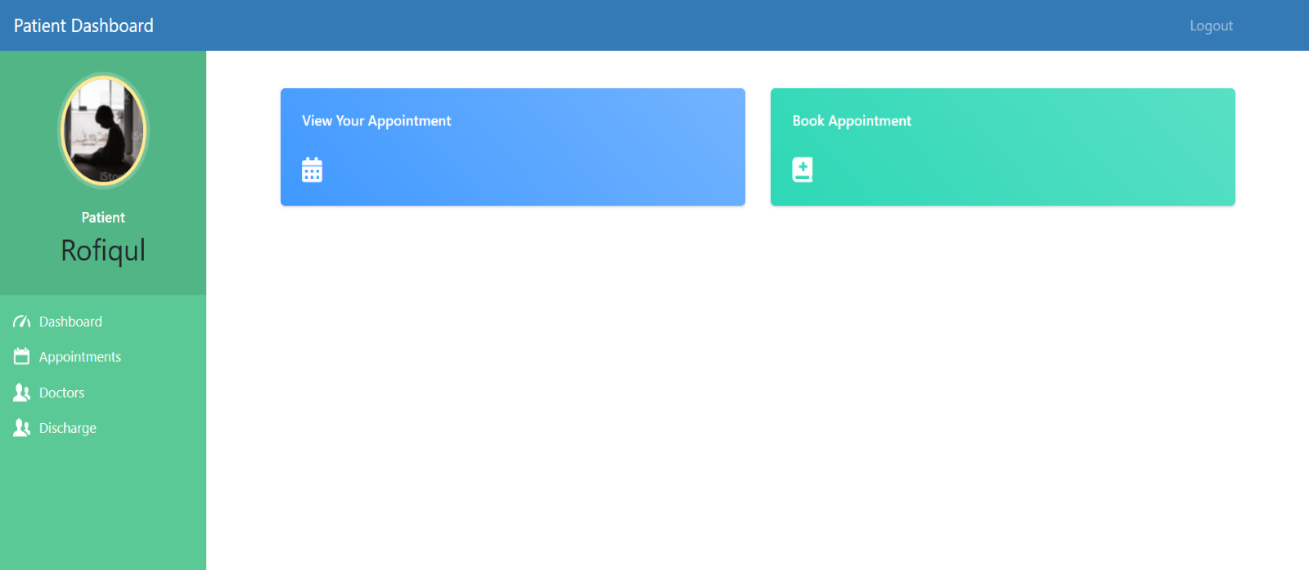
**Figure 8.** Patient login page of the website.

The website's log-in page is shown in Figure 8. A patient must first register and then provide a valid username and password to gain access to the system. A valid e-mail address must also be provided by the patient. Admins and doctors must also provide their login and password to get access to the system.



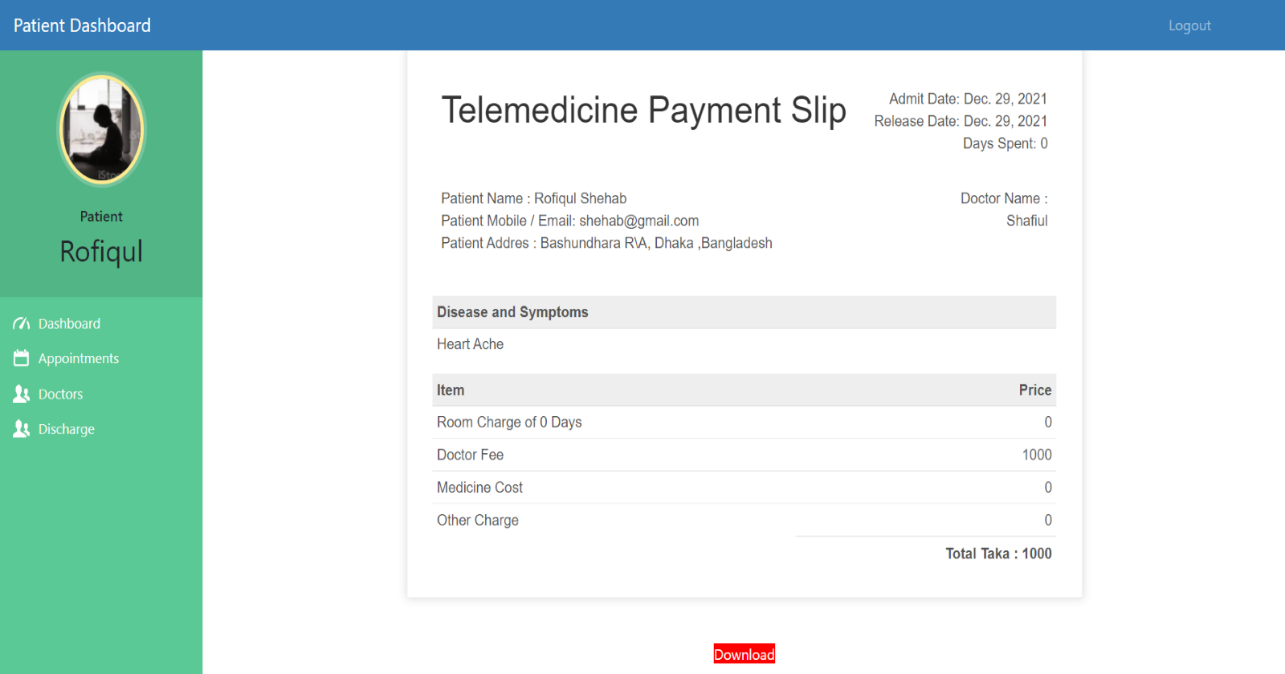
**Figure 9.** Patient’s dashboard.

Figure 9 depicts the patient's dashboard. Patients may access crucial information on the dashboard, such as the doctor's name, symptoms, contact information, address, and department, as well as participate in a video conversation. After reviewing a patient's report, a doctor can issue a prescription for them. The doctor must take notes on the patient's symptoms. After that, the doctor must actually write the prescription. A doctor can only write a prescription for a patient who has scheduled an appointment. background in medicine After reviewing a patient's report, a doctor can issue a prescription for them. Patients can also save time by printing their prescription or downloading it as a PDF file. Patients may view the date and time the prescription was written. can learn about the symptoms they're experiencing as well as the medication they're taking. In addition, the patient will be able to download the payment slip and participate in a video chat. Figure 9 shows the dashboard of the patients. Patients will be able to check their appointment, doctors, phone numbers, and locations. There is also the option of using the making video call link to make a video call. patients can also view and remove their scheduled appointments. Patients also can view the doctor list of which doctors are available on this system.



**Figure 10.** Patient views appointment.

Figure 10 shows the patient's appointment. Patients can easily book an appointment from this section. To book an appointment, patients have to describe a few of their problems, and then they might need to select the doctor and department to book the appointment. Then, if the booking is approved by the admin, the patient will be able to see the appointment status, date, and doctor's name in the "view your appointment" section.



**Figure 11.** Patient’s payment slip.

Figure 11 shows the payment slip. Patients can easily download a payment slip after completing the video conference and completing the appointment with the doctor. This payment slip is generated by the admin. On the pay slip, the patient can see the appointment date and the number of days he or she met with the doctor. The patient's name and address are also indicated on the payment slip. The patient can also see his or her disease and symptoms on the payment slip. The main things that are on the payment slip are fees and charges. Fees are summed up to give the total amount to be paid by the patient.

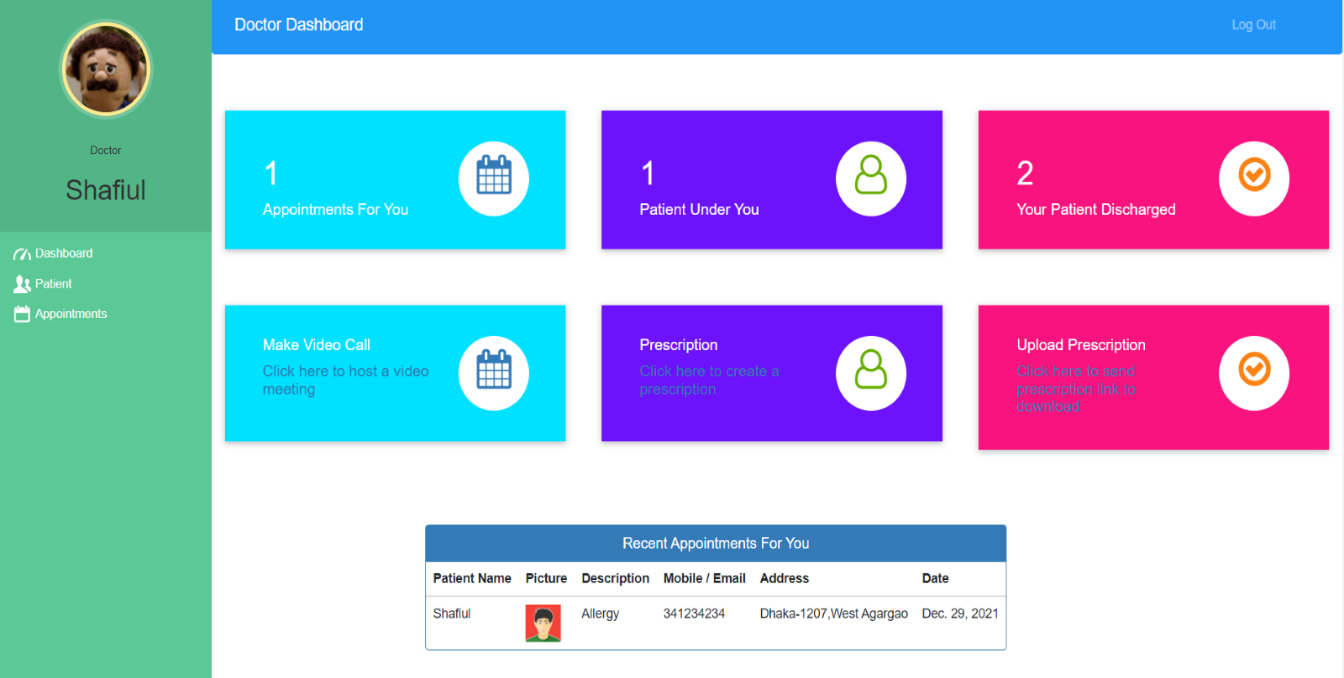
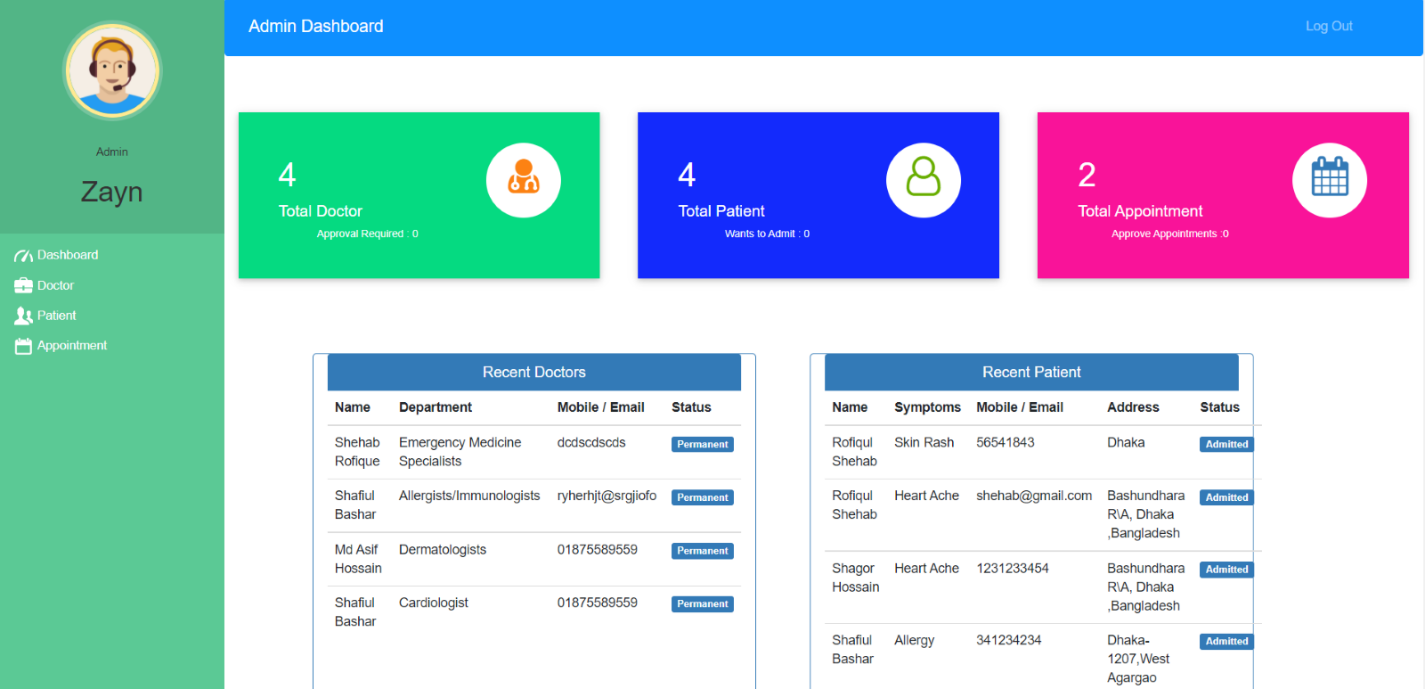
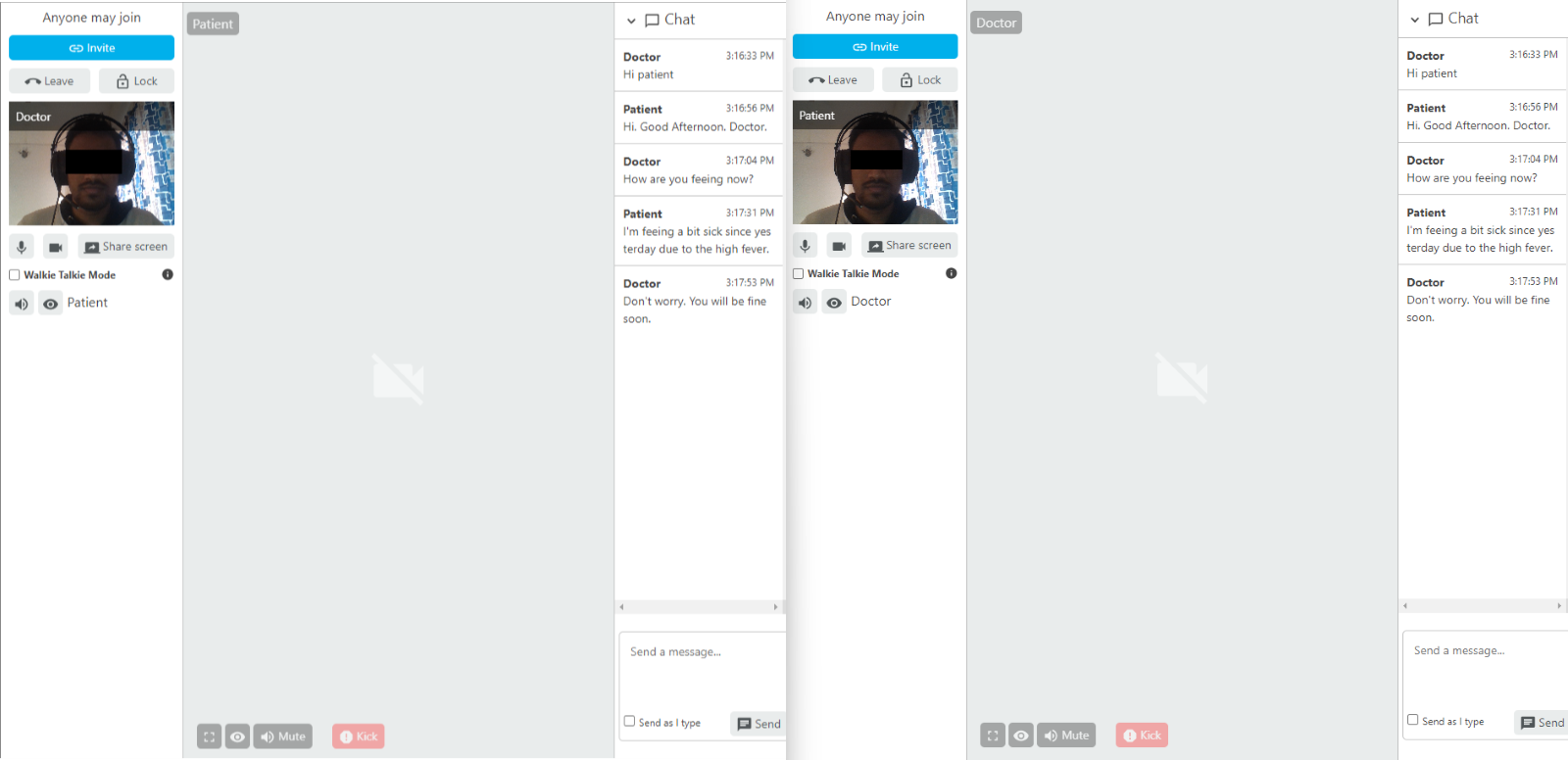
  
**Figure 12.** Doctor’s dashboard.

Figure 12 depicts the doctor's dashboard. On their dashboard, doctors will be able to see how many recent appointments they have and how many patients they have. There's also the option of setting up a video conference call. Doctors may also create prescriptions for their patients and email them to them with a link to download them in pdf format. Through the patient information, doctors will be able to view patient records. Doctors will be able to see the date of the patient's appointment and may also remove it from the list.



**Figure 13.** Admin’s dashboard.

Figure 13 shows the admin dashboard. The admin will be provided with a list of open appointments and patients after logging in. The admin can see the recent numbers of patients and doctors on the dashboard and can also see the total number of appointments. The administrator will be able to schedule doctor-patient appointments. The admin must choose a day and time, as well as the patient's and doctor's names, to establish an appointment. The overall number of appointments, the number of completed appointments, and the number of scheduled appointments may all be viewed by the administrator. The administrator will then be able to see patient records, admit patients, and approve them. An administrator can also make changes to a patient's personal information and remove them from the system. Admin can even discharge them by creating a payment slip. The admin can also delete the records of patients and doctors from the system. Admins will be able to access the doctors' records, as well as register, authorize, and view the specializations of the doctors. Admins will also be able to view and book doctor's appointments as well as authorize assistants. Admin will be able to check doctor's records and register doctors. Doctors can be approved and their specializations can be viewed by the administrator. Admins will also be able to add new doctors to the system.



**Fig 14.** Video calling & chatting feature.

The figure shows the doctor's video call feature in this system. This feature uses a video call API based on WebRTC. Doctors can create rooms according to their department name. Also, doctors can chat with patients and share links with other patients. Doctors can share device screens and kick out any patient at any time. By this time, doctors will also be able to send a link to the patient to download the prescription at the end of the video conversation. In the figure, the patient's videocall feature of this system. Patients can join their booked doctor's room. Also, patients can do chat with doctors and share links with other patients with share screens of their device. Patient can mute and unmute himself & herself during the video meeting. There is also a real time chat option for sending machine

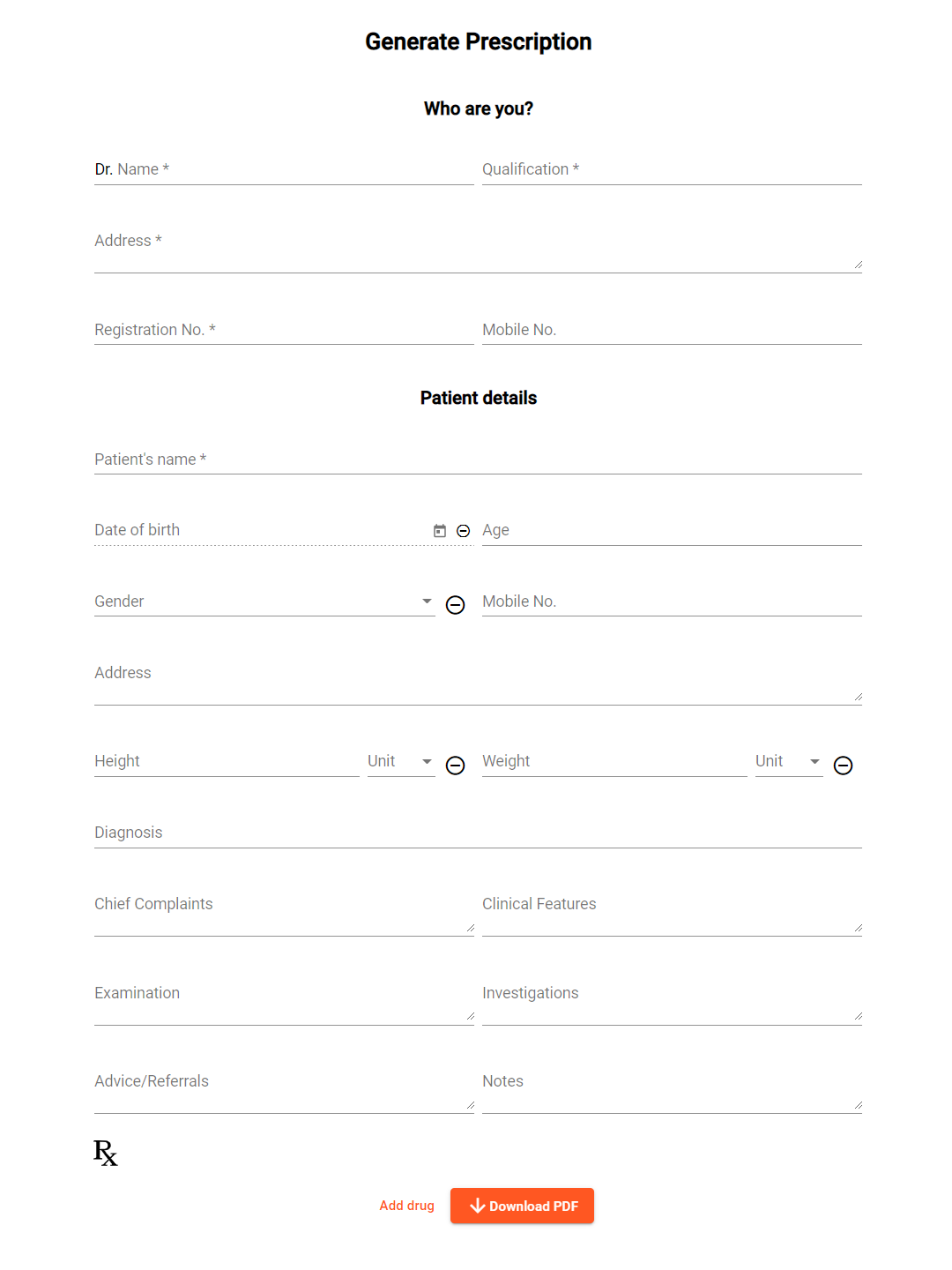
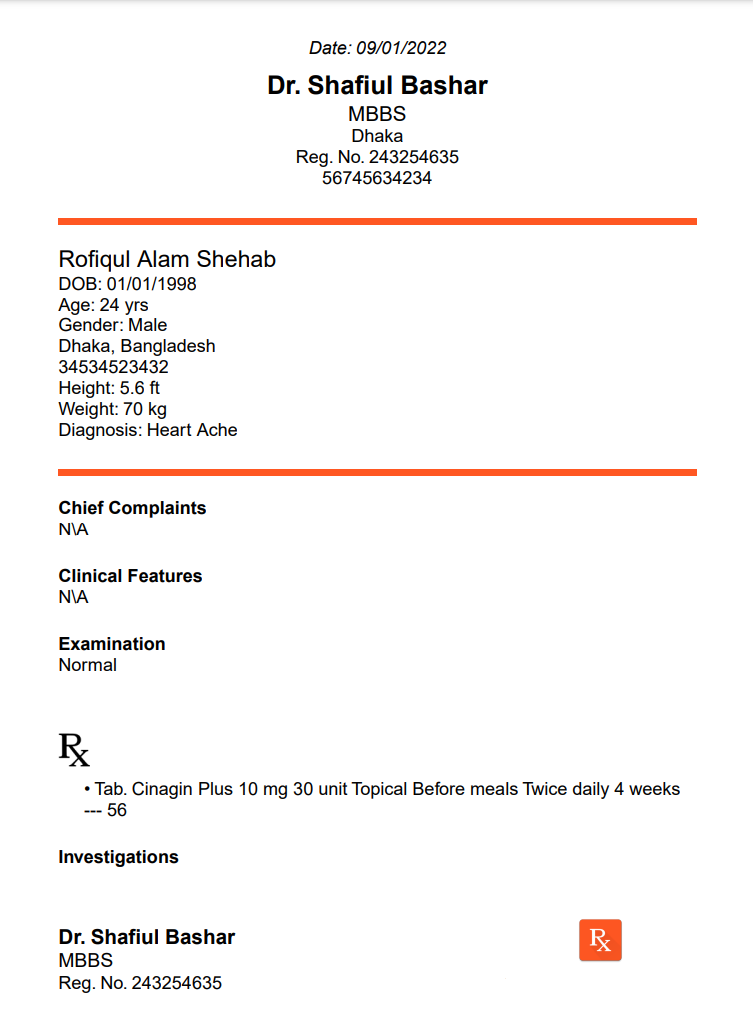
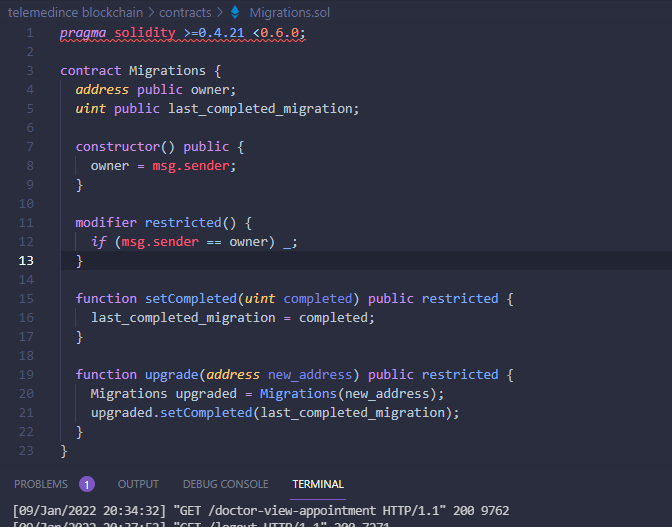
  
**Fig 16.** Doctor’s prescription.

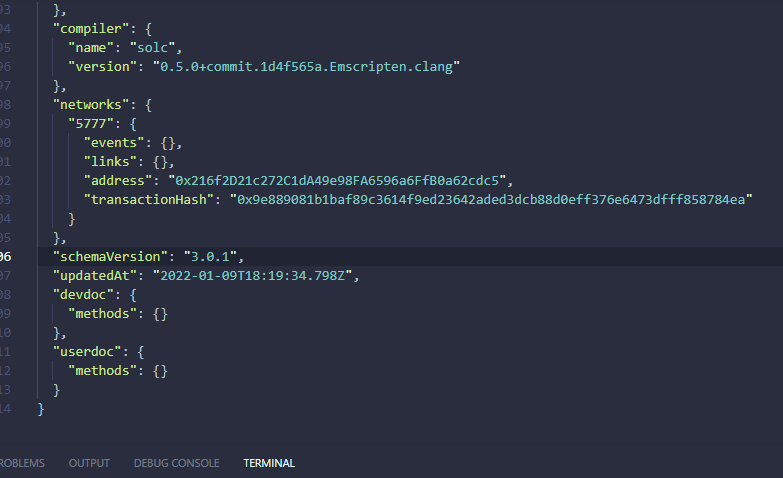
Figure 16 depicts the patient's prescription. The doctor will be the one to write the prescription. At the top of the prescription, there will be information from the doctors. Basic patient information and records, such as name, date of birth, gender, address, diagnosis, primary complaints and clinical characteristics, examination, investigation, advice, and notes, will be included in the prescription. A list of medications can also be added to the prescription by the doctor. It is also available in pdf format for the doctor to download. Finally, Figure 17, depicts the patient's prescription in a pdf format. A pdf version of the prescription will be prepared and provided to the patient through a link that will allow them to download the prescription in pdf format.



**Fig 17.** Patient downloads prescription in pdf format.



In figure 18 depicts the smart contracts. We have constructed some smart contracts to make an interaction with system during the transaction between doctor and patient. In figure 19, The input data is then “hashed” by the node, which converts it into a hash value or “hash,” which must always include a specific number of zeros. +commit node determines if a hash satisfies the difficulty requirements. +commit hash must begin with the appropriate number of zeroes. If the hash satisfies the difficulty requirements, it is disseminated to the rest of the network’s users. +commit first user to discover a valid hash converts the block into a new block and is paid for the block reward and fees.



**Fig 19.** Snapshot of a transaction hash.

**Table 1:** Software Comparison with Other Papers

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Name | | | Website | Video calling | Prescription | Telemedicine service | Blockchain |
| 1 | | This Paper | YES | | YES | YES | YES | YES |
| 2 | | Ref. [3] | NO | | NO | NO | YES | YES |
| 3 | | Ref. [4] | NO | | NO | NO | YES | NO |
| 4 | | Ref. [6] | NO | | NO | YES | YES | YES |
| 5 | | Ref. [7] | NO | | NO | NO | NO | NO |

**Table 2:** Comparison table between this paper and others paper

|  |  |  |  |
| --- | --- | --- | --- |
| Points of this paper | Points of others papers | | |
| (1) As a result of the implementation of the Django framework and the blockchain technology described in this article, third-party interventions are eliminated, and strong peer connections are developed. | | (1) Not every paper suggests their own coin as a money, third-party interventions are common in telemedicine systems. |
| (2) The video calling process is both straightforward and safe. | | (2) The broker can occasionally cause the transaction system to malfunction. The doctors' and patients' faith in each other is jeopardized. |
| (3) This article uses the most advanced MVT technology to create a user-friendly web application for the doctor and patient. It can also protect the site against cyberattacks during visits, resulting in a trusted site. | | (3). Some other paper is devoid of blockchain technology. If there is any, it is insufficiently secure. As a result, the website has been hacked on occasion, and money has been taken by the hackers. |
| (4) Uses a smart contract correctly. As a result, the system has improved reliability. | | (4) In the telemedicine sector, smart contracts are not used properly. As a result, there's a chance that sensitive patient information will be tampered with. |
| (5) This article is designed to facilitate open and transparent communication between doctors and patients at any time when the patient wants assistance. | | (5) When a payment is sent to the incorrect address, they have an extremely complicated refund procedure. As a result, patients are forced to deal with a variety of unpleasant situations. |

**4.0 Conclusion**

The purpose of this article is to improve the intelligence, robustness, and security of our telemedicine application. This blockchain has the potential to provide complete payment transparency. It guards against unwanted access and data tampering on our website. Smart contracts also reduce the amount of time spent on time-consuming documentation. In traditional telemedicine, a lot of paperwork is normally required. Smart contracts are immutable because of the blockchain, which retains the information as proof. This paradigm has the most impact on telemedicine payment transactions and refunds processes. This study examines the design and implementation of a secure telemedicine system. In this age of technology everyone is eager to adopt new technology. So, it is time to renovate medical industry through technology. we have developed an online telemedicine system using blockchain technology to take this renovation one step further. Many people die in remote areas due to not getting proper treatment. Some people die on the way of hospital. Emergency medical situations like Cardiac arrest, choking, sprains, nosebleeds can be treated by the proper instructions of a doctor and it doesn’t require any equipment. Through our online telemedicine system people can get proper consultation and save their beloved one's life. Since it is an online system and medical records are sensitive data so our main concern is the security. That’s why we have used blockchain technology, which is the most reliable and transparent technology to secure data. People can benefit from this website in a variety of ways, there is no need to take time off work, and there is less time spent in the waiting room., lower risk of contracting a new illness, improved clinical workflows, and increased practice efficiency, among others. As a result, we should take steps to make use of this website and spread it among the general public in order to improve their treatment. In terms of storage and processing expenses, blockchain transactions will be tremendously advantageous. This research promises to speed up and reduce payment procedure costs, as well as increase financial inclusion by providing more options for patients who do not have easy access to financial services, among its many benefits the most important of which is the ability to keep payment procedures secure. This is a small-scale, one-of-a-kind piece of work. The delay may be altered if there is a huge volume of data. In terms of storage and processing expenses, blockchain transactions in telemedicine will be tremendously advantageous. Another option to increase performance is to use decentralized databases like BigchainDB and HBasechainDB.

# **Data Availability**

No data were used to support the findings of this study.

# **Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the study.

# **Acknowledgements**

The authors are thankful for the support from the North South University Researchers Supporting Project (NSURSP-2021), North South University, Dhaka, Bangladesh.

# **5 References**

[1] K. Roadruck, “Telehealth statistics and telemedicine trends 2021”, The Checkup,2021 [Online]. Available:<https://www.singlecare.com/blog/news/telehealth-statistics/>

[2] C. Petrov, “91+ Blockchain Statistics: Understand Blockchain in 2021”. TechJury,2022, [Online]. Available: <https://techjury.net/blog/blockchain-statistics/#gref>

[3] Y.M. Tang and G. Ho, “Integrating blockchain for improving data sharing in implant surgery”. Hamburg, Germany: Proceedings of ISER 147th International Conference”, pp.27–31, 2018.Available: https://research.polyu.edu.hk/en/publications/integrating-blockchain-for-improving-data-sharing-in-implant-surg

[4] D. Yonatan and D. Husna, “Design of Decentralized Application for Telemedicine Image Record System with Smart Contract on Ethereum”, International Journal of Advanced Computer Science and Applications, vol 12, no 10, pp. 272-281, 2021. Available:10.14569/ijacsa.2021.0121030

[5] A. Abugabah and N. Nizamuddin, "Decentralized Telemedicine Framework for a Smart Healthcare Ecosystem,” IEEE Access, vol. 8, pp. 166575-166588, 2020, Available:10.1109/ACCESS.2020.3021823

[6] R. W. Ahmad, S. Khaled, R. Jayaraman, I. Yaqoob and S. Ellahham, "The Role of Blockchain Technology in Telehealth and Telemedicine”, International Journal of Medical Informatics, vol. 148, no. 18:104399, pp. 1-8, 2021, Available: 10.1016/j.ijmedinf.2021.104399

[7] “Healthcare Data Breach Statistics”. HIPAA Journal, 2021[Online]. Available at: https://www.hipaajournal.com/healthcare-data-breach-statistics/

[8] M. Hasan, “High demand for telemedicine in pandemic”, Dhaka Tribune, 2020. [Online]. Available at:https://archive.dhakatribune.com/bangladesh/2020/08/29/high-demand-for-telemedicine-in-pandemic[Accessed 9 January 2022].

[9] Praava Health raises $10.6m to increase access to quality healthcare in Bangladesh. (2021, March 8). Dhaka Tribune. https://www.dhakatribune.com/health/2021/03/08/praava-health-raises-10-6m-to-increase-access-to-quality-healthcare-in-bangladesh

[10] A. H. da Costa Mayer & R. D. R Righi, “Electronic health records in a Blockchain: A systematic review”. Health Informatics Journal, vol 26, no. 2, pp. 1273–1288, 2019, Available: 10.1177/1460458219866350

[11] L. Tan, D. Tivey and T. Kopunic, “Blockchain technology in health care”, ANZ Journal of Surgery, vol. 90, no 12, pp. 2415–2419, 2020. Available: 10.1111/ans.16455

[12] T. R. Seaberg & D. C. Seaberg, “Use of Blockchain Technology for Electronic Prescriptions”, Blockchain in Healthcare Today, vol. 4., no. 183, 2021, Available: 10.30953/bhty.v4.183

[13] M. Hölbl, M. Kompara, A. Kamišalić, & Z. L. Nemec, “A Systematic Review of the Use of Blockchain in Healthcare”, Symmetry, vol. 10, no. 10, pp. 470, 2018, Available: 10.3390/sym10100470

[14] M. Kassab, J. DeFranco, T. Malas, P. Laplante, G. Destefanis and Neto, “Exploring Research in Blockchain for Healthcare and a Roadmap for the Future”, IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 4, pp. 1835–1852, 2019, Available: https: 10.1109/tetc.2019.2936881

[15] R. W. Ahmad, K. Salah, R. Jayaraman, I. Yaqoob and S. Ellahham, “The role of blockchain technology in telehealth and telemedicine”, International Journal of Medical Informatics, vol. 148, Article ID. 104399, 2021, Available: 10.1016/j.ijmedinf.2021.104399

[16] M. D. Turjo, M. M. Khan and M. Kaur, “Smart Supply Chain Management Using the Blockchain and Smart Contract”. Scientific Programming, vol 2021, no 6092792, pp. 1–12, 2021, Available: 10.1155/2021/6092792

[17] K. Koshechkin, G. Lebedev, G. Radzievsky and R. Seepold “Blockchain Technology Projects to Provide Telemedical Services: Systematic Review”, Journal of Medical Internet Research, vol. 23, no. 8, 2019, Available:10.2196/17475

[18] M. Lou, X. Dong and Z. Cao, “SESCF: A Secure and Efficient Supply Chain Framework via Blockchain-Based Smart Contracts”, Security and Communication Networks, Vol. 2021, pp. 1–18, 2021, Available: 10.1155/2021/8884478

[19]D.G.H.S.(n.d.).TelemedicineService.[Online].Available:https://dghs.gov.bd/index.php/en/publications/other-publication/84-english-root/ehealth-eservice/490-telemedicine-service

[20] M.U. Chowdhury, K. Suchana, S.M.E. Alam and M.M. Khan, “Blockchain Application in Banking System.”, Journal of Software Engineering and Applications, vol. 14, no. 7, pp. 298-311, 2021, Available: 10.4236/jsea.2021.147018

[21] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, & T. Hayajneh, “Healthcare Blockchain System Using Smart Contracts for Secure Automated Remote Patient Monitoring”, Journal of Medical Systems, vol. 42, no. 130, pp. 1-7, 2018, Available: 10.1007/s10916-018-0982-x

**Landing page**

**Navbar**

<!DOCTYPE html>

{% load static %}

<html lang="en">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<style type="text/css">

.bs-example {

margin: 0px;

}

.navbar-brand {

font-size: 20px;

font-family: sans-serif;

}

</style>

</head>

<body>

<div class="bs-example">

<nav class="navbar navbar-expand-md navbar-dark fixed-top" style="background:#3493ff;">

<a href="/" class="navbar-brand">Secure Online Telemedicine</a>

<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">

<div class="navbar-nav">

</div>

<div class="navbar-nav">

<a href="/" class="btn btn-outline-dark">Home</a>

<a href="/contactus" class="btn btn-outline-dark">Contact Us</a>

<a href="/aboutus" class="btn btn-outline-dark">About Us</a>

<div class="dropdown show">

<a class="btn btn-outline-dark dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Account

</a>

<div class="dropdown-menu" aria-labelledby="dropdownMenuLink">

<a href="/adminclick" class="dropdown-item">Admin</a>

<a href="/doctorclick" class="dropdown-item">Doctor</a>

<a href="/patientsignup" class="dropdown-item">Sign Up</a>

<a href="/patientlogin" class="dropdown-item">Log In</a>

</div>

</div> </div>

</div> </nav></div></body></html>

**Homepage**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Home</title>

</head>

<body>

{% include "hospital/navbar.html" %}

{%block content%}

{%endblock content%}

{% include "hospital/footer.html" %}

</body>

</html>

**Index**

{% extends 'hospital/homebase.html' %}

{% load static %}

{% block content %}

<head>

<style media="screen">

.jumbotron {

margin-top: 0px;

margin-bottom: 0px;

background-image: url('{% static "images/bg1.jpg" %}');

background-size: cover;

background-repeat: no-repeat;

}

.jumbotron h5,

h3 {

text-align: center;

}

.alert {

margin: 0px;

}

.glow {

font-size: 50px;

color: rgb(255, 255, 255);

text-align: center;

-webkit-animation: glow 1s ease-in-out infinite alternate;

-moz-animation: glow 1s ease-in-out infinite alternate;

animation: glow 1s ease-in-out infinite alternate;

}

@-webkit-keyframes glow {

from {

text-shadow: 0 0 10px #eeeeee, 0 0 20px #000000, 0 0 30px #000000, 0 0 40px #000000, 0 0 50px #9554b3, 0 0 60px #9554b3, 0 0 70px #9554b3;

}

to {

text-shadow: 0 0 20px #eeeeee, 0 0 30px #ff4da6, 0 0 40px #ff4da6, 0 0 50px #ff4da6, 0 0 60px #ff4da6, 0 0 70px #ff4da6, 0 0 80px #ff4da6;

}

}

</style>

</head>

<br>

<br>

<div class="jumbotron" style="margin-bottom: 0px;margin-top: 0px;">

<br>

<h5 class="display-3 glow">Welcome to the secure online telemedine System</h5>

<br><br><br><br><br>

<br><br><br><br><br>

<p>

<h3>Do you have an Emergency ?</h3>

<p class="lead">

<a class="btn btn-primary btn-lg" href="/patientclick" role="button">Take Appointment Here</a>

</p>

<br><br>

</div>

<br><br><br><br>

{% include "hospital/admin\_doctor\_patient\_card.html" %}

<br><br><br>

{% endblock content %}

**footer**  
<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

<style>

/\*---------------------------------------

Social section

-----------------------------------------\*/

footer {

padding: 0px 0px 0px 0px;

background-color: black;

margin: 0px;

}

.fa {

padding: 20px;

font-size: 23px;

width: 60px;

text-align: center;

text-decoration: none;

margin: 5px 2px;

border-radius: 50%;

}

.fa:hover {

opacity: 0.5;

text-decoration: none;

}

.fa-facebook {

background: #3B5998;

color: white;

margin-top: 30px;

}

.fa-whatsapp {

background: #25d366;

color: white;

}

.fa-twitter {

background: #55ACEE;

color: white;

}

.fa-instagram {

background: #125688;

color: white;

}

.center {

display: flex;

justify-content: center;

}

p {

text-align: center;

}

</style>

</head>

<footer>

<p>

<a href="https:/" class="fa fa-facebook"></a>

<a href="https:/" class="fa fa-whatsapp"></a>

<a href="https:/" class="fa fa-instagram"></a>

<a href="https:/" class="fa fa-twitter"></a>

</p>

<br>

<div class="container">

<div class="row">

<div class="col-md-12 col-sm-12">

<div style="color:#ffffff;" class="wow fadeInUp footer-copyright">

<p>A Secure Online Telemedicine System <br>

Copyright &copy; 2021 </p>

</div>

</div>

</div> </div></footer></html>

**Patient side**

**Patient Dashboard**

{% extends 'hospital/patient\_base.html' %}

{% load static %}

{% block content %}

{%include 'hospital/patient\_dashboard\_cards.html'%}

<br><br><br><br>

{% endblock content %}

**Patient Book Appointment**

{% extends 'hospital/patient\_base.html' %}

{% load widget\_tweaks %}

{% block content %}

<head>

<style media="screen">

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

.form-control {

border-radius: 1.5rem;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

.menu {

top: 50px;

}

</style>

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>

<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

<br><br>

{% if message %}

<script>

var a = "{{message}}";

alert(a);

</script>

{% endif %}

<!------ add appointment page by patient ---------->

<form method="post">

{% csrf\_token %}

<div class="container register-form">

<div class="form">

<div class="note">

<p>Book Appointment Details</p>

</div>

<div class="form-content">

<div class="row">

<div class="col-md-12">

<div class="form-group">

{% render\_field appointmentForm.description class="form-control" placeholder="Description" %}

</div>

<div class="form-group">

{% render\_field appointmentForm.doctorId class="form-control" placeholder="doctor" %}

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Book</button>

</div>

</div>

</div>

</form>  
{% endblock content %}

**Patient Final Bill**

{% extends 'hospital/admin\_base.html' %}

{% load static %}

{% block content %}

<head>

<meta charset="utf-8">

<title>A simple, clean, and responsive HTML invoice template</title>

<style>

.invoice-box {

max-width: 800px;

margin: auto;

padding: 30px;

border: 1px solid #eee;

box-shadow: 0 0 10px rgba(0, 0, 0, .15);

font-size: 16px;

line-height: 24px;

font-family: 'Helvetica Neue', 'Helvetica', Helvetica, Arial, sans-serif;

color: #555;

}

.invoice-box table {

width: 100%;

line-height: inherit;

text-align: left;

}

.invoice-box table td {

padding: 5px;

vertical-align: top;

}

.invoice-box table tr td:nth-child(2) {

text-align: right;

}

.invoice-box table tr.top table td {

padding-bottom: 20px;

}

.invoice-box table tr.top table td.title {

font-size: 45px;

line-height: 45px;

color: #333;

}

.invoice-box table tr.information table td {

padding-bottom: 40px;

}

.invoice-box table tr.heading td {

background: #eee;

border-bottom: 1px solid #ddd;

font-weight: bold;

}

.invoice-box table tr.details td {

padding-bottom: 20px;

}

.invoice-box table tr.item td {

border-bottom: 1px solid #eee;

}

.invoice-box table tr.item.last td {

border-bottom: none;

}

.invoice-box table tr.total td:nth-child(2) {

border-top: 2px solid #eee;

font-weight: bold;

}

@media only screen and (max-width: 600px) {

.invoice-box table tr.top table td {

width: 100%;

display: block;

text-align: center;

}

.invoice-box table tr.information table td {

width: 100%;

display: block;

text-align: center;

}

}

/\*\* RTL \*\*/

.rtl {

direction: rtl;

font-family: Tahoma, 'Helvetica Neue', 'Helvetica', Helvetica, Arial, sans-serif;

}

.rtl table {

text-align: right;

}

.rtl table tr td:nth-child(2) {

text-align: left;

}

.menu {

top: 50px;

}

.download {

text-align: center;

display: block;

}

</style>

</head>

<br><r><br>

<div class="invoice-box">

<table cellpadding="0" cellspacing="0">

<tr class="top">

<td colspan="2">

<table>

<tr>

<td class="title">

<h1>Telemedicine Payslip</h1>

</td>  
 <td>

Admit Date: {{admitDate}}<br>

Release Date: {{todayDate}}<br>

Days Spent: {{day}}

</td>

</tr>

</table>

</td>

</tr>

<tr class="information">

<td colspan="2">

<table>

<tr>

<td>

Patient Name : {{name}}<br>

Patient Mobile : {{mobile}}<br>

Patient Addres : {{address}}<br>

</td>

<td>

Doctor Name :<br>

{{assignedDoctorName}}<br>

</td>

</tr>

</table>

</td>

</tr>

<tr class="heading">

<td>

Disease and Symptoms

</td>

<td> </td></tr>

<tr class="details">

<td>

{{symptoms}}

</td>

</tr>

<tr class="heading">

<td>

Item

</td>

<td>

Price

</td>

</tr>

<tr class="item">

<td>

Room Charge of {{day}} Days

</td>

<td>

{{roomCharge}}

</td>

</tr>

<tr class="item">

<td>

Doctor Fee

</td>

<td>

{{doctorFee}}

</td>

</tr>

<tr class="item">

<td>

Medicine Cost

</td>

<td>

{{medicineCost}}

</td>

</tr>

<tr class="item last">

<td>

Other Charge

</td>

<td>

{{OtherCharge}}

</td>

</tr>

<tr class="total">

<td></td>

<td>

Total Taka : {{total}}

</td>

</tr>

</table>

</div>

<br><br>

<div class="download">

<a style="background:red; width:500px;" href="{% url 'download-pdf' patientId %}">Download</a>

</div>

{% endblock content %}

**Patient Signup**

<!DOCTYPE html>

{% load widget\_tweaks %}

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Telemedicine</title>

<style type="text/css">

body {

color: #aa082e;

background-color: #b6bde7;

font-family: 'Roboto', sans-serif;

}

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

.form-control {

border-radius: 1.5rem;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

</style>

</head>

<body>

{% include "hospital/navbar.html" %}

<br> <br> <br> <br>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

<div class="container register-form">

<div class="form">

<div class="note">

<p>Register in Secure Online Telemedicine System</p>

</div>

<div class="form-content">

<div class="row">

<div class="col-md-6">

<div class="form-group">

{% render\_field userForm.first\_name class="form-control" placeholder="First Name" %}

</div>

<div class="form-group">

{% render\_field userForm.username class="form-control" placeholder="Username" %}

</div>

<div class="form-group">

{% render\_field patientForm.address class="form-control" placeholder="Address" %}

</div>

<div class="form-group">

{% render\_field patientForm.symptoms class="form-control" placeholder="Symptoms" %}

</div>

<div class="form-group">

{% render\_field patientForm.profile\_pic required="required" class="form-control" placeholder="Profile Picture" %}

</div>

</div>

<div class="col-md-6">

<div class="form-group">

{% render\_field userForm.last\_name class="form-control" placeholder="Last Name" %}

</div>

<div class="form-group">

{% render\_field userForm.password class="form-control" placeholder="Password" %}

</div>

<div class="form-group">

{% render\_field patientForm.mobile class="form-control" placeholder="Mobile Number / Email" %}

</div>

<div class="form-group">

{% render\_field patientForm.assignedDoctorId class="form-control" placeholder="Doctor" %}

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Register</button>

<div class="text-center">Already have an account? <a href="patientlogin">Login here</a></div>

</div>

</div>

</div>

</form>

{% include "hospital/footer.html" %}

</body>

</html>

**Doctor Side**

**Doctor Dashboard**

{% extends 'hospital/doctor\_base.html' %}

{% load static %}

{% block content %}

{%include 'hospital/doctor\_dashboard\_cards.html'%}

<br><br><br><br>

<div class="container">

<div class="row">

<div class="panel panel-primary" style="margin-left:15%;">

<div class="panel-heading" style="text-align:center;">

<h6 class="panel-title">Recent Appointments For You</h6>

</div>

<table class="table table-hover" id="dev-table">

<thead>

<tr>

<th>Patient Name</th>

<th>Picture</th>

<th>Description</th>

<th>Mobile / Email</th>

<th>Address</th>

<th>Date</th>

</tr>

</thead>

{% for a,p in appointments %}

<tr>

<td>{{a.patientName}}</td>

<td> <img src="{% static p.profile\_pic.url %}" alt="Profile Pic" height="40px"width="40px"/></td>

<td>{{a.description}}</td>

<td>{{p.mobile}}</td>

<td>{{p.address}}</td>

<td>{{a.appointmentDate}}</td>

</tr>

{% endfor %}

</table>

</div>

</div>

</div>

{% endblock content %}

**Doctor View Appointment**

{% extends 'hospital/doctor\_base.html' %}

{% block content %}

{%load static%}

<head>

<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>

<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>

<style media="screen">

a:link {

text-decoration: none;

}

h6 {

text-align: center;

}

.row {

margin: 100px;

}

</style>

</head>

<div class="container">

<div class="panel panel-primary">

<div class="panel-heading">

<h6 class="panel-title">Your Appointments</h6>

</div>

<table class="table table-hover" id="dev-table">

<thead>

<tr>

<th>Patient Name</th>

<th>Picture</th>

<th>Description</th>

<th>Mobile / Email</th>

<th>Address</th>

<th>Appointment Date</th>

</tr>

</thead>

{% for a,p in appointments %}

<tr>

<td>{{a.patientName}}</td>

<td> <img src="{% static p.profile\_pic.url %}" alt="Profile Pic" height="40px" width="40px" /></td>

<td>{{a.description}}</td>

<td>{{p.mobile}}</td>

<td>{{p.address}}</td>

<td>{{a.appointmentDate}}</td>

</tr>

{% endfor %}

</table>

</div>

</div>

{% endblock content %}

**Doctor Signup**

<!DOCTYPE html>

{% load widget\_tweaks %}

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Telemedicine</title>

<style type="text/css">

body {

color: #aa082e;

background-color: #b6bde7;

font-family: 'Roboto', sans-serif;

}

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

.form-control {

border-radius: 1.5rem;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

</style>

</head>

<body>

{% include "hospital/navbar.html" %}

<br>

<br>

<br>

<br>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

<div class="container register-form">

<div class="form">

<div class="note">

<p>Register In Secure Online Telemedicine System</p>

</div>

<div class="form-content">

<div class="row">

<div class="col-md-6">

<div class="form-group">

{% render\_field userForm.first\_name class="form-control" placeholder="First Name" %}

</div>

<div class="form-group">

{% render\_field userForm.username class="form-control" placeholder="Username" %}

</div>

<div class="form-group">

{% render\_field doctorForm.department class="form-control" placeholder="Department" %}

</div>

<div class="form-group">

{% render\_field doctorForm.address class="form-control" placeholder="Address" %}

</div>

</div>

<div class="col-md-6">

<div class="form-group">

{% render\_field userForm.last\_name class="form-control" placeholder="Last Name" %}

</div>

<div class="form-group">

{% render\_field userForm.password class="form-control" placeholder="Password" %}

</div>

<div class="form-group">

{% render\_field doctorForm.mobile class="form-control" placeholder="Mobile" %}

</div>

<div class="form-group">

{% render\_field doctorForm.profile\_pic required="required" class="form-control" placeholder="Profile Picture" %}

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Register</button>

<div class="text-center">Already have an account? <a href="doctorlogin">Login here</a></div>

</div>

</div>

</div>

</form>

{% include "hospital/footer.html" %}

</body>

</html>

**Admin Dashboard**

{% extends 'hospital/admin\_base.html' %}

{% load static %}

{% block content %}

{%include 'hospital/admin\_dashboard\_cards.html'%}

<br><br><br><br>

<div class="container">

<div class="row">

<div class="panel panel-primary col-md-5" style="margin-left:2%;">

<div class="panel-heading" style="text-align:center;">

<h6 class="panel-title">Recent Doctors</h6>

</div>

<table class="table table-hover" id="dev-table">

<thead>

<tr>

<th>Name</th>

<th>Department</th>

<th>Mobile / Email</th>

<th>Status</th>

</tr>

</thead>

{% for d in doctors %}

<tr>

<td> {{d.get\_name}}</td>

<td>{{d.department}}</td>

<td>{{d.mobile}}</td>

{%if d.status%}

<td> <span class="label label-primary">Permanent</span></td>

{% else %}

<td> <span class="label label-success">On Hold</span></td>

{% endif %}

</tr>

{% endfor %}

</table>

</div>

<div class="panel panel-primary col-md-5" style="margin-left:5%;">

<div class="panel-heading" style="text-align:center;">

<h6 class="panel-title">Recent Patient</h6>

</div>

<table class="table table-hover" id="dev-table">

<thead>

<tr>

<th>Name</th>

<th>Symptoms</th>

<th>Mobile / Email</th>

<th>Address</th>

<th>Status</th>

</tr>

</thead>

{% for p in patients %}

<tr>

<td> {{p.get\_name}}</td>

<td>{{p.symptoms}}</td>

<td>{{p.mobile}}</td>

<td>{{p.address}}</td>

{%if p.status%}

<td> <span class="label label-primary">Admitted</span></td>

{% else %}

<td> <span class="label label-success">On Hold</span></td>

{% endif %} </tr>

{% endfor %}

</table>

</div>

</div>

</div>

{% endblock content %}

**Log IN**

<!DOCTYPE html>

{% load widget\_tweaks %}

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Telemedicine</title>

<style type="text/css">

body {

color: #aa082e;

background-color: #b6bde7;

font-family: 'Roboto', sans-serif;

}

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

.form-control {

border-radius: 1.5rem;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

</style>

</head>

<body>

{% include "hospital/navbar.html" %}

<br>

<br>

<br><br>

<!--- login page for admin by admin ---------->

<form method="post">

{% csrf\_token %}

<div class="container register-form">

<div class="form">

<div class="note">

<p>Admin Login Page</p>

</div>

<div class="form-content">

<div class="row">

<div class="col-md-6">

<div class="form-group">

{% render\_field form.username class="form-control" placeholder="Username" %}

</div>

</div>

<div class="col-md-6">

<div class="form-group">

{% render\_field form.password class="form-control" placeholder="Password" %}

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Login</button>

<div class="text-center">Do not have account? <a href="adminsignup">Signup here</a></div>

</div>

</div>

</div>

</form>

<br><br><br>

{% include "hospital/footer.html" %}

</body>

</html>

**Video Call**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta

name="viewport"

content="width=device-width, initial-scale=1, shrink-to-fit=no"

/>

<!-- IMPORTANT SETUP -->

<!-- Change the YOUR\_API\_KEY section of the config URL to match your API key -->

<meta

name="simplewebrtc-config-url"

content="https://api.simplewebrtc.com/config/guest/492c9fa0a4b959099ceb1959"/>

<meta name="simplewebrtc-sound-message-receive" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-message-send" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-peer-enter" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-peer-exit" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-test-output" content="/url-of-mp3-file" />

<!--

The Talky Sample app can be configured to play sounds in some cases:

<meta name="simplewebrtc-sound-message-receive" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-message-send" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-peer-enter" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-peer-exit" content="/url-of-mp3-file" />

<meta name="simplewebrtc-sound-test-output" content="/url-of-mp3-file" />

-->

<title>Telemedicine Video Call</title>

<style>

body {

margin: 0;

padding: 0;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

min-height: 100%;

font-family: -apple-system,

BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen', 'Ubuntu', 'Cantarell',

'Fira Sans', 'Droid Sans', 'Helvetica Neue', sans-serif;

font-size: 16px;

line-height: 1.5;

color: #4d5659;

text-rendering: optimizeSpeed;

} code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

[hidden] {

display: none;

}

a {

background-color: transparent;

text-decoration: none;

}

a:active,

a:hover {

outline: 0;

}

img {

border: 0;

}

svg:not(:root) {

overflow: hidden;

}

button,

input {

color: inherit;

font: inherit;

margin: 0;

}

button {

overflow: visible;

}

button {

text-transform: none;

}

button,

html input[type='button'],

input[type='reset'],

input[type='submit'] {

-webkit-appearance: button;

cursor: pointer;

}

button[disabled],

html input[disabled] {

cursor: default;

}

button::-moz-focus-inner,

input::-moz-focus-inner {

border: 0;

padding: 0;

}

input {

line-height: normal;

}

\*,

\*:after,

\*:before {

box-sizing: border-box;

}

p {

margin-top: 0;

margin-bottom: 30px;

}

img,

video {

max-width: 100%;

}

html {

box-sizing: border-box;

}

\*,

\*:before,

\*:after {

box-sizing: inherit;

}

\* {

margin: 0;

padding: 0;

.create-room-form {

text-align: center;

}

.create-room-form-input-wrapper {

border: 1px #00b0e9 solid;

padding: 10px;

border-radius: 3px;

}

.create-room-form-input-wrapper:focus {

border: 1px #f00 solid;

}

.domain {

color: #b7c0c3;

}

.create-room-form-input {

border: none;

background: none;

color: #00b0e9;

}

.create-room-form-input:focus {

outline: none;

}

.create-room-form-button {

margin-left: 5px;

padding: 8px 10px !important;

display: inline-block;

}

@media (max-width: 420px) {

.create-room-form-input-wrapper {

width: 100%;

display: block;

text-align: left;

}

.create-room-form-button {

width: 100%;

margin-top: 20px;

display: block;

margin-left: 0px;

}

.create-room-form-input {

width: 75%;

}

}

.container {

margin: 60px auto;

width: 85%;

max-width: 800px;

text-align: center;

}

</style>

</head>

<body>

<noscript>

You need to enable JavaScript to run this app.

</noscript>

<div id="root"></div>

<template id="haircheck-header">

<h2 style="text-align: center">Ready to join a video chat?</h2>

</template>

<template id="empty-peer-grid">

<h2 style="text-align: center; color: rgba(68, 68, 68, 0.25)">

There isn't anyone here (yet!)

</h2>

</template>

<template id="empty-roster">

<div></div>

</template>

<template id="homepage">

<div class="container">

<form class="create-room-form" method="GET" action="/">

<span class="create-room-form-input-wrapper"

><span class="domain">localhost/</span

><input

type="text"

name="room"

placeholder="choose a room name"

class="create-room-form-input"/></span

><button

class="create-room-form-button button button-default button-undefined"

type="submit"

>

Start a chat

</button>

</form>

</div>

</template>

<script type="text/javascript">

const params = new URLSearchParams(window.location.search);

SimpleWebRTC.run({

roomName: params.get('room'),

root: document.getElementById('root'),

gridPlaceholder: () => SimpleWebRTC.loadTemplate('empty-peer-grid'),

haircheckHeaderPlaceholder: () =>

SimpleWebRTC.loadTemplate('haircheck-header'),

emptyRosterPlaceholder: () => SimpleWebRTC.loadTemplate('empty-roster'),

homepagePlaceholder: () => SimpleWebRTC.loadTemplate('homepage')

});

</script>

</body>

</html>

**Smart Contract**

pragma solidity >=0.4.21 <0.6.0;

contract Migrations {

address public owner;

uint public last\_completed\_migration;

constructor() public {

owner = msg.sender;

}

modifier restricted() {

if (msg.sender == owner) \_;

}

function setCompleted(uint completed) public restricted {

last\_completed\_migration = completed;

}

function upgrade(address new\_address) public restricted {

Migrations upgraded = Migrations(new\_address);

upgraded.setCompleted(last\_completed\_migration);

}

}

**Transaction Hash**

{

"contractName": "TodoList",

"abi": [

{

"constant": true,

"inputs": [],

"name": "taskCount",

"outputs": [

{

"name": "",

"type": "uint256"

}

],

"payable": false,

"stateMutability": "view",

"type": "function",

"signature": "0xb6cb58a5"

}

],

"bytecode": "0x608060405260008055348015601357600080fd5b506099806100226000396000f3fe608060405260043610603e5763ffffffff7c0100000000000000000000000000000000000000000000000000000000600035041663b6cb58a581146043575b600080fd5b348015604e57600080fd5b5060556067565b60408051918252519081900360200190f35b6000548156fea165627a7a723058202177827df92f167243a038636cdcab7c2185d8ccd7aca5ba504b4aeb3cf865740029",

"deployedBytecode": "0x608060405260043610603e5763ffffffff7c0100000000000000000000000000000000000000000000000000000000600035041663b6cb58a581146043575b600080fd5b348015604e57600080fd5b5060556067565b60408051918252519081900360200190f35b6000548156fea165627a7a723058202177827df92f167243a038636cdcab7c2185d8ccd7aca5ba504b4aeb3cf865740029",

"sourceMap": "25:50:1:-;;;71:1;47:25;;25:50;8:9:-1;5:2;;;30:1;27;20:12;5:2;25:50:1;;;;;;;",

"deployedSourceMap": "25:50:1:-;;;;;;;;;;;;;;;;;;;;;;;47:25;;8:9:-1;5:2;;;30:1;27;20:12;5:2;47:25:1;;;;;;;;;;;;;;;;;;;;;;;;:::o",

"source": "pragma solidity ^0.5.0;\n\ncontract TodoList {\n uint public taskCount = 0;\n}\n",

"sourcePath": "/Users/gregory/code/eth-todo-list/contracts/TodoList.sol",

"ast": {

"absolutePath": "/Users/gregory/code/eth-todo-list/contracts/TodoList.sol",

"exportedSymbols": {

"TodoList": [

62

]

},

"id": 63,

"nodeType": "SourceUnit",

"nodes": [

{

"id": 58,

"literals": [

"solidity",

"^",

"0.5",

".0"

],

"nodeType": "PragmaDirective",

"src": "0:23:1"

},

{

"baseContracts": [],

"contractDependencies": [],

"contractKind": "contract",

"documentation": null,

"fullyImplemented": true,

"id": 62,

"linearizedBaseContracts": [

62

],

"name": "TodoList",

"nodeType": "ContractDefinition",

"nodes": [

{

"constant": false,

"id": 61,

"name": "taskCount",

"nodeType": "VariableDeclaration",

"scope": 62,

"src": "47:25:1",

"stateVariable": true,

"storageLocation": "default",

"typeDescriptions": {

"typeIdentifier": "t\_uint256",

"typeString": "uint256"

},

"typeName": {

"id": 59,

"name": "uint",

"nodeType": "ElementaryTypeName",

"src": "47:4:1",

"typeDescriptions": {

"typeIdentifier": "t\_uint256",

"typeString": "uint256"

}

},

"value": {

"argumentTypes": null,

"hexValue": "30",

"id": 60,

"isConstant": false,

"isLValue": false,

"isPure": true,

"kind": "number",

"lValueRequested": false,

"nodeType": "Literal",

"src": "71:1:1",

"subdenomination": null,

"typeDescriptions": {

"typeIdentifier": "t\_rational\_0\_by\_1",

"typeString": "int\_const 0"

},

"value": "0"

},

"visibility": "public"

}

],

"scope": 63,

"src": "25:50:1"

}

],

"src": "0:76:1"

},

"legacyAST": {

"absolutePath": "/Users/gregory/code/eth-todo-list/contracts/TodoList.sol",

"exportedSymbols": {

"TodoList": [

62

]

},

"id": 63,

"nodeType": "SourceUnit",

"nodes": [

{

"id": 58,

"literals": [

"solidity",

"^",

"0.5",

".0"

],

"nodeType": "PragmaDirective",

"src": "0:23:1"

},

{

"baseContracts": [],

"contractDependencies": [],

"contractKind": "contract",

"documentation": null,

"fullyImplemented": true,

"id": 62,

"linearizedBaseContracts": [

62

],

"name": "TodoList",

"nodeType": "ContractDefinition",

"nodes": [

{

"constant": false,

"id": 61,

"name": "taskCount",

"nodeType": "VariableDeclaration",

"scope": 62,

"src": "47:25:1",

"stateVariable": true,

"storageLocation": "default",

"typeDescriptions": {

"typeIdentifier": "t\_uint256",

"typeString": "uint256"

},

"typeName": {

"id": 59,

"name": "uint",

"nodeType": "ElementaryTypeName",

"src": "47:4:1",

"typeDescriptions": {

"typeIdentifier": "t\_uint256",

"typeString": "uint256"

}

},

"value": {

"argumentTypes": null,

"hexValue": "30",

"id": 60,

"isConstant": false,

"isLValue": false,

"isPure": true,

"kind": "number",

"lValueRequested": false,

"nodeType": "Literal",

"src": "71:1:1",

"subdenomination": null,

"typeDescriptions": {

"typeIdentifier": "t\_rational\_0\_by\_1",

"typeString": "int\_const 0"

},

"value": "0"

},

"visibility": "public"

}

],

"scope": 63,

"src": "25:50:1"

}

],

"src": "0:76:1"

},

"compiler": {

"name": "solc",

"version": "0.5.0+commit.1d4f565a.Emscripten.clang"

},

"networks": {

"5777": {

"events": {},

"links": {},

"address": "0x216f2D21c272C1dA49e98FA6596a6FfB0a62cdc5",

"transactionHash": "0x9e889081b1baf89c3614f9ed23642aded3dcb88d0eff376e6473dfff858784ea"

}

},

"schemaVersion": "3.0.1",

"updatedAt": "2022-01-09T18:19:34.798Z",

"devdoc": {

"methods": {}

},

"userdoc": {

"methods": {}

}

}