Chapter 1

Introduction

1.1 Overview

Information Technology (IT) is playing an increasingly important role in all sectors of the economy, healthcare sector being no exception. In fact, the increased role of IT in the healthcare sector has led to coining of new terminology, "Health Informatics," which deals with the use of IT for better healthcare services. Health informatics applications often deal with the health record of individuals, in digital form, which is referred to as the Electronic Health Record (EHR). An electronic health record (EHR) is more than a digital version of a patient's paper chart. EHRs are real-time, patient-centred records that make information available instantly and securely to authorized users. While an EHR does contain the medical and treatment histories of patients, an EHR system is built to go beyond standard clinical data collected in a provider's office and can be inclusive of a broader view of a patient's care. EHRs can:

- Contain a patient's medical history, diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory and test results.
- Allow access to evidence-based tools that providers can use to make decisions about a patient's care.
- Automate and streamline provider workflow.

In our work, we are going to build a centralized cloud-based storage system backed with authorization for users of the system, thus making it convenient for both the patient and hospital to handle their respective clinical data.

1.2 Need for work

Our world has been radically transformed by digital technology – smartphones, tablets, and web-enabled devices have transformed our daily lives and the way we communicate. Medicine is an information-rich enterprise. A greater and more seamless flow of information within a digital health care infrastructure, created by electronic health records (EHRs), encompasses and leverages digital progress and can transform the way care is delivered and compensated. Thus, we can tackle following problems that are currently being faced in this sector:

- **Patient Inconvenience:**

  Managing a bunch of reports is really a tedious task for a patient to deal with. So, with the help of this system, we can improve health care quality and convenience for patients by:

  1. Reducing the need to fill out the same forms at each visit.
  2. The convenience of e-prescriptions electronically sent to pharmacy.
  3. Patient portals with online interaction for providers.
  4. Electronic referrals allowing easier access to follow-up care with specialists.

- **Inconvenience for Hospitals:**

  On the other hand, hospitals find it difficult to deal with this physical data for every patient it treats. EHR system can help hereby:

  1. Providing quick access to patient records from inpatient and remote locations for more coordinated, efficient care.
  2. Enhancing decision support, clinical alerts, reminders, and medical information.
  3. Performance-improving tools, real-time quality reporting.
  4. Legible, complete documentation that facilitates accurate coding and billing.
  5. Providing interfaces with labs, registries, and other EHRs.
  6. Safer, more reliable prescribing.

- **Diagnostics & Patient Outcomes:**

  With EHRs, providers can have reliable access to a patient's complete health information. This comprehensive picture can help providers diagnose patients' problems sooner.

- **Care Coordination:**

  Electronic health record (EHR) systems can decrease the fragmentation of care by improving care coordination. EHRs have the potential to integrate and organize patient health information and facilitate its instant distribution among all authorized providers involved in a patient's care.

- **Practice Efficiencies and Cost Savings:**

  EHRs can benefit medical practices in a variety of ways, including:

  1. Reduced transcription costs.
  2. Reduced chart pulls, storage, and re-filing costs.
  3. Reduced medical errors through better access to patient data and error prevention alerts.

1.3 Objectives

- To implement and organize a centralized system over cloud, which can be shared among different health organizations.
- Provide analysis for stored data on the basis of different diseases over regions
- Provide a secure way to save and access of patient information.
- Provide access to authorized users only.
- To improve Patient Care.
- To enhance the management experience for hospitals.
- To improve Care Coordination.
- To improve Diagnostics.

Chapter 2

Requirement Analysis and Specification

2.1 Information gathering

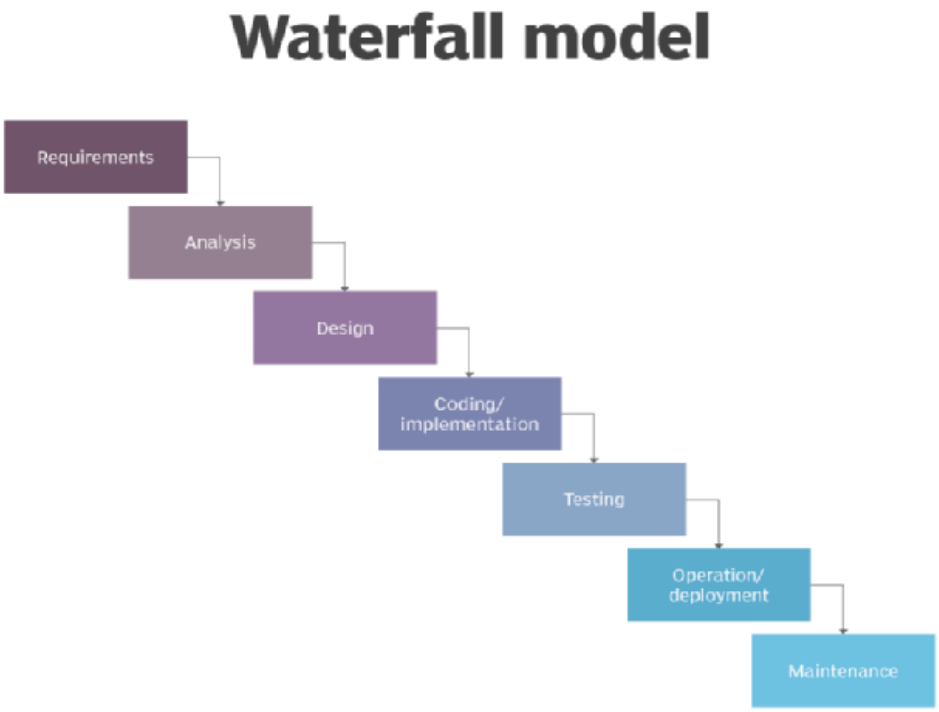
Automating the medical health record management system has received a lot of research focus. The privacy and security of the patient health record being of utmost importance, this field of research has seen various approaches being suggested. In a typical scenario, an EHR records the vital stats, diagnoses, medications, history of immunizations, laboratory and radiology reports, doctor notes, etc. The above information is gathered was gathered during the hospital visits.

2.2 Survey regarding the project

According to the International Organization for Standardization, electronic health records (EHR) are “repository of patient data in digital form, stored and exchanged securely, and accessible by multiple authorized users. In the past decade, it has been seen as an explosion in the amount of digital information stored in electronic health records (EHRs). While primarily designed for archiving patient information and performing administrative healthcare tasks like billing, many researchers have found secondary use of these records for various clinical informatics applications. In this review, we survey the current research on clinical tasks based on EHR data, where we find a variety of techniques being applied to several types of clinical applications including information extraction and outcome prediction.

2.3 SDLC model (Waterfall Model)

The SDLC life cycle model that we have used in our project is waterfall model. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.



The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

2.4 SRS (Software Requirement Specification)

2.4.1 Introduction

2.4.1.1 Purpose:

Maintaining medical records is a tedious task. Medical records can be stored in many different ways e.g. tradition file-based system i.e. paper-based. The problem gets more difficult when the number of records increases. Our goal is to provide a secure way to save and access patient information. The purpose is to implement and organize centralized system over cloud, which can be shared among different health organizations. Then provide access to authorized users only and improve patient care. In the end, our aim towards regional analysis on various records.

2.4.1.2 Project Scope:

Such information can be used for many purposes such as the regional classification of diseases. Electronic health record (EHR) systems can decrease the fragmentation of care by improving care coordination. Reduced medical errors through better access to patient data and error prevention alerts.

2.4.1. Project Features:

The system will provide following features:

- Secure way to save and access of patient information
- Diagnostics
- Acts as a Management tool for hospitals
- Security to patient credentials
- Authorization
- Platform Independent

2.4.2 Operating Environment

1. Operating System: Windows, Linux, Unix, Mac, etc.
2. Languages and technologies: JavaScript, html, CSS, reactjs.
3. NODE JS with MySQL
4. Web browser: Chrome, Opera, Mozilla and Safari.

2.4.3 Functional Requirements

Functional requirements define the capabilities and functions that a system must be able to perform successfully. The functional requirements of EHR system includes:

- Authentication and authorization of users.
- Provide diagnostics and report.
- Privacy and security to patient credentials.

2.3.4 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific it's behaviour. Some of the non-functional requirements include:

- Maintainability-ease to maintain.
- Performance/response time- fast response.
- Usability by target user community- easy to use.
- Expandability- needs to be future proof or upgradable.

Chapter 3
System Design

3. Design

The design of the system is described using the following diagrams:

- System Architecture
- Modules
- Data Flow Diagrams
- Database Diagram

3.1 System Architecture

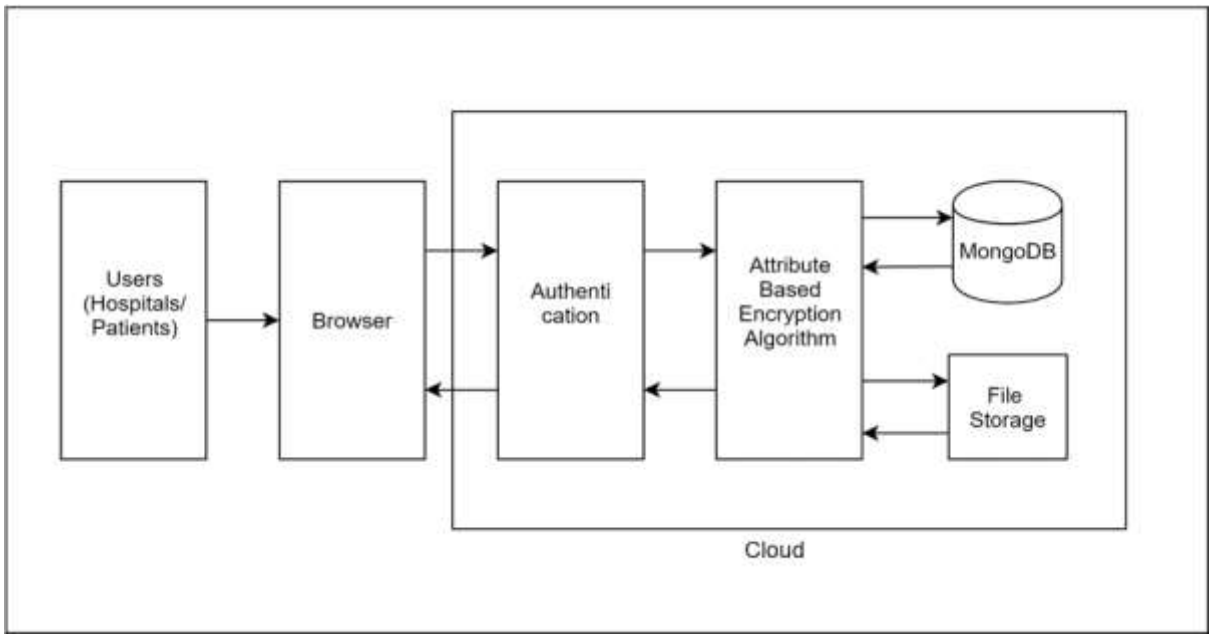


Fig 4.1: System Architecture

The Users of the system are patients and hospitals. The web interface is provided on browser through which user interacts with the system. The Cloud block specifies centralized system consisting of modules including authentication, encryption/decryption, database, and file storage. The authentication module allows authorized users to access to provided services. Encryption/ Decryption module facilitates to securely store and retrieve respective data. Database is used to store user's information and file storage stores reports, images, clinical data, etc.

3.2 Modules

3.2.1 User Interface:

This module provides users with a secured interface to access their respective data. This user interface will be designed with the technologies like React.js, HTML, CSS, JS and will interact with back end through AJAX API calls.

3.2.2 Encryption:

Every initiated process by the users of the system (i.e. patients and hospitals) gets through authorization phase. Processes including:

1. Patient data upload and retrieval by hospitals over the cloud.
2. Patient access to its own data.
3. Hospital's access to patient data provided with a patient key.

This encryption will be implemented through Attributed based encryption.

3.2.3 Storage:

For a centralized storage system, elastic file system (EFS) is used which is a service provided by amazon web service (AWS). Every module's instance is connected to this centralized storage. Any instance can opt for any other instance with the help of NFS-V4 (Network file system) protocol. For storing data, we are using MongoDB because of its simplicity and scalability. For storing reports and other medical files we are using a file system which is AWS's EFS which is network file system. This file system solution will help us to store reports. This FS will be shared among various EC2 instances.

3.2.4 Decryption:

With every retrieval, whether by the patient or hospital data decryption is processed. The data stored on EFS will be encrypted. So, when hospital or patient makes request for data the data needs to be decrypted. First, we are checking authorization and then we are decrypting that data using our algorithm and sending reports back to patients.

3.2.5 Analytics:

Healthcare providers want to make the best possible decision for their patients, and they often need some extra help to do so. Now that the vast majority of providers have adopted electronic health records, they have access to the basic big data that will allow them to engage in clinical analytics. Clinical analytics can be patient-focused, such as using the EHR to compare a diabetic individual's HbA1C readings over the past two years to benchmark data from other non-diabetic patients, using algorithms to create risk scores for post-surgical infections and 30-day readmissions based on vital signs, or using large-scale genomic data to match patients with rare cancers to personalized treatments.

3.3 Data Flow Diagram

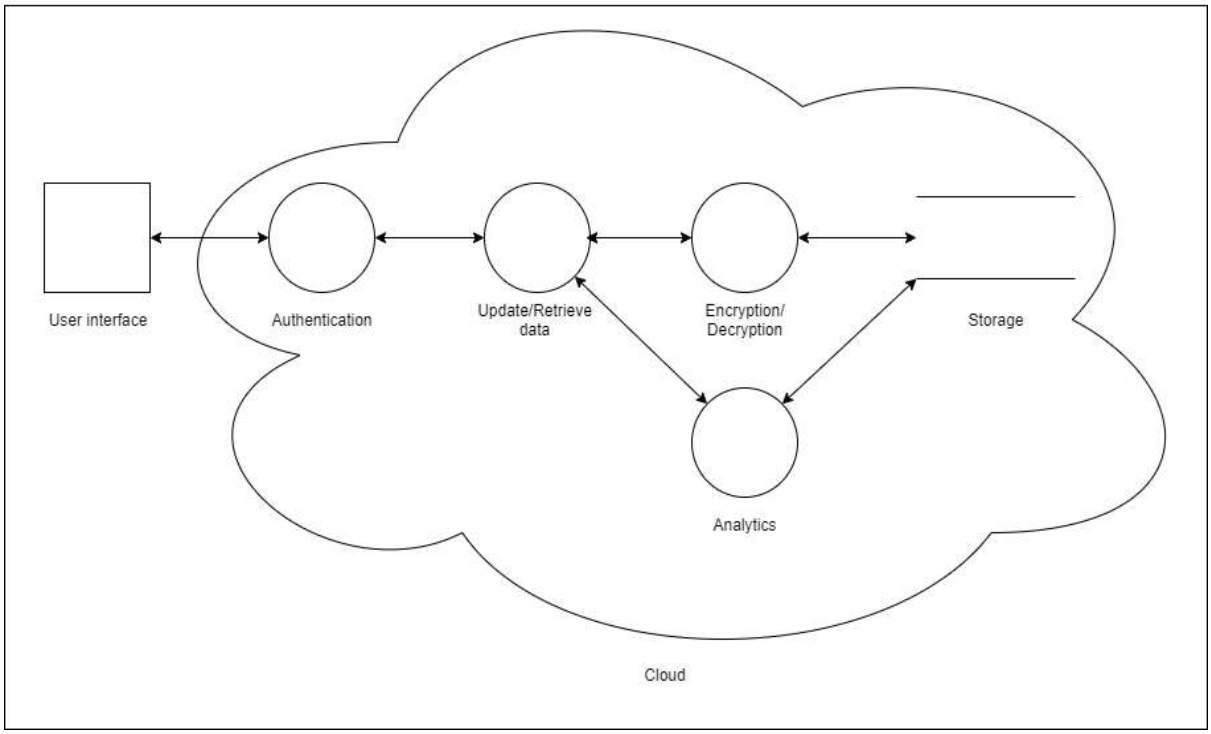


Fig 3.3.1: 0-Level DFD

The user interacts with the system through browser. Providing valid credentials user can use the features of the system. The operations carried out by user are protected and privacy is maintained.

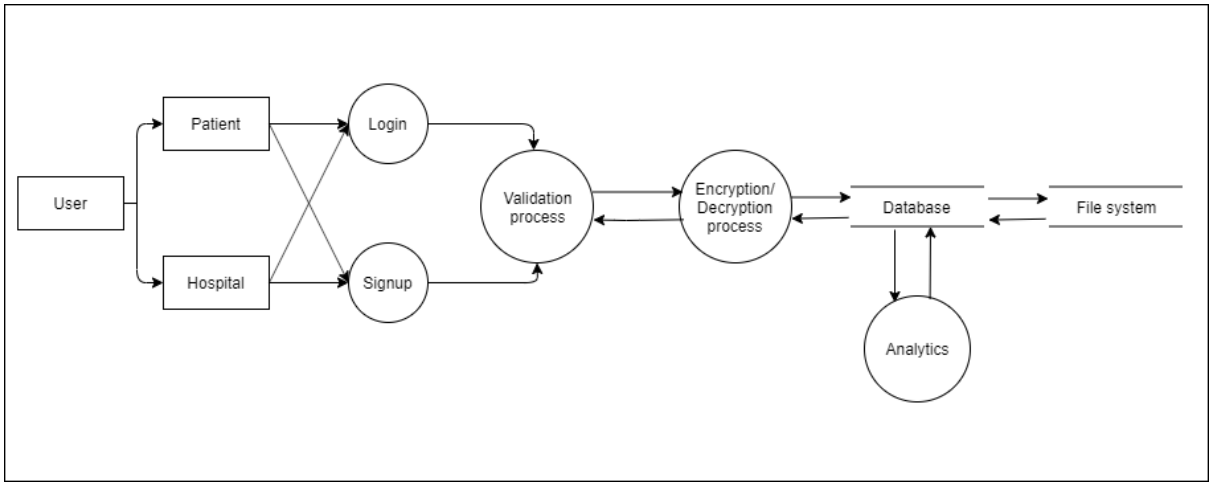


Fig 4.3.1: 1-Level DFD

The user of system can use the facilities of system by registering as new patient or hospital. If the user has already registered then he/she can directly login to proceed to dashboard. The operations performed by user are encrypted before being stored in storage (database) and decrypted at the time of retrieval. The system uses file system to store records which are accessed by user via database.

3.4 Database Design

Tables in database:

- Hospital
- Doctor
- Patient
- Report

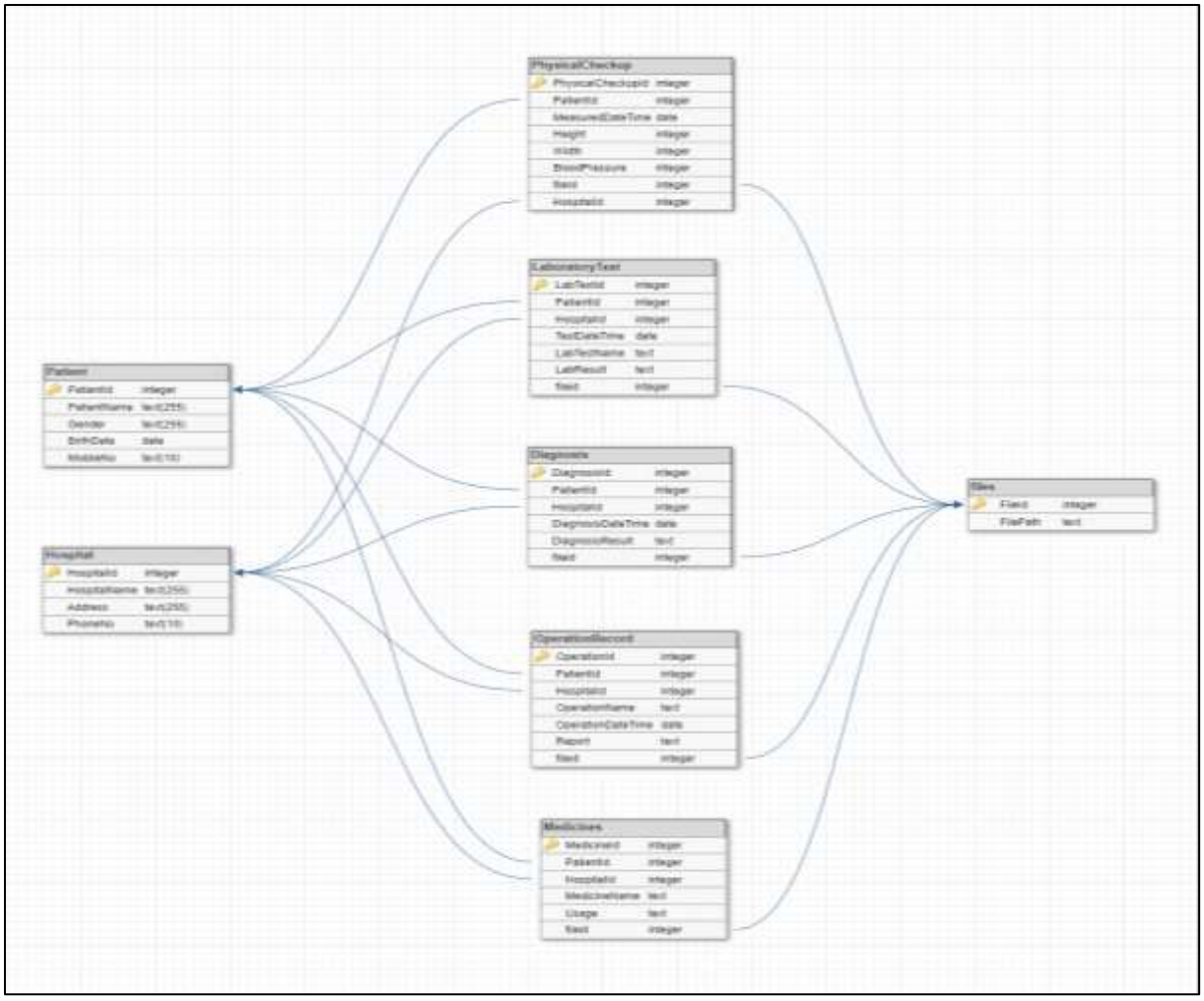


Fig 3.4: Database Design

Denotes primary key

Denotes Foreign Key

Chapter 4

Implementation

4. Implementation and Coding:

4.1Tools and Technologies used

1. **HTML**

   The hypertext mark-up language is used to develop the front end. The first page of UI allows user to grab image file from client system and allows it to be submitted as query image. The result page of the system is dynamically created depending on obtained results from image recognition and retrieval module.

2. **CSS**

   To make UI more appealing to the user CSS is used.

3. **JavaScript**

   JavaScript is an object-based scripting language which is lightweight and cross-platform JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

4. **React**

   React is an open-source JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

5. **Node.js**

   Node.js is an open source server environment.Node.js is free**.** Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.). Node.js uses JavaScript on the server.

6. **Express**

   Express is a small framework that sits on top of Node.js's web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middleware and routing. it adds helpful utilities to Node.js's HTTP objects. it facilitates the rendering of dynamic HTTP objects.

7. **MySQL**

   MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

4.2 Hardware & Software Requirements

- Processor: Intel Pentium 4 processor or later that's SSE2 capable
- Ram: 512MB of RAM / 2GB for better browser support
- Operating System: Windows, Linux, Unix, Android, Mac.
- Browser: Chrome, Opera, Firefox, Microsoft edge.

Chapter 5

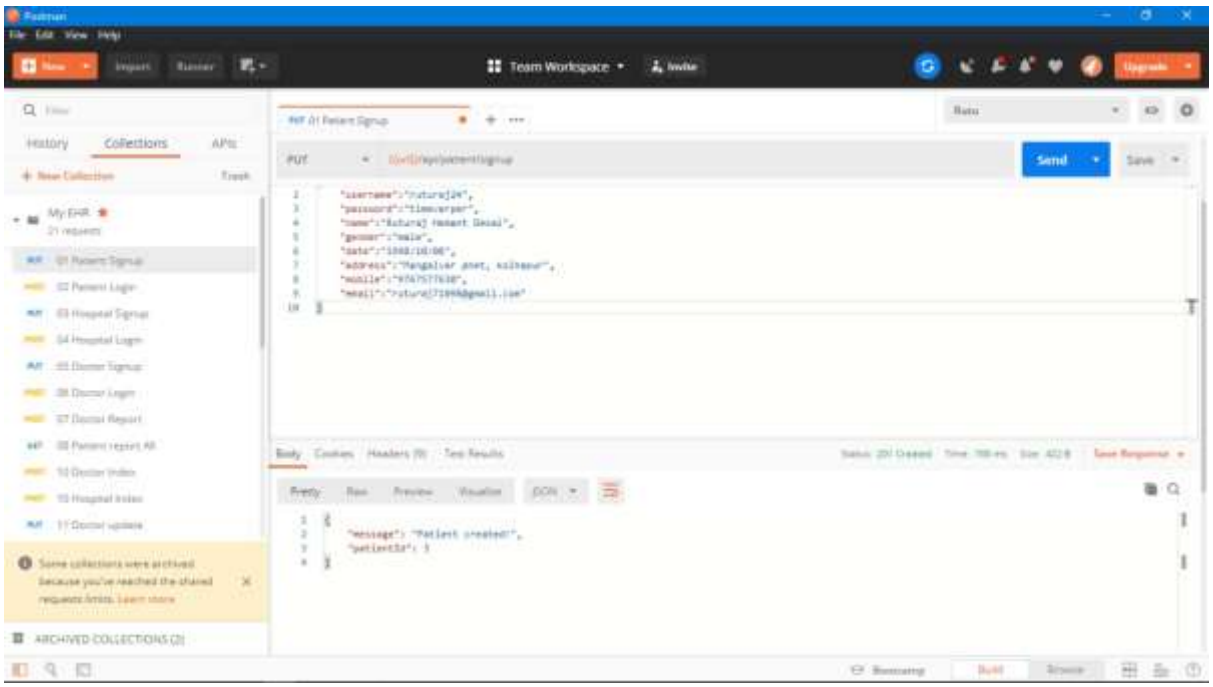
Testing and Results

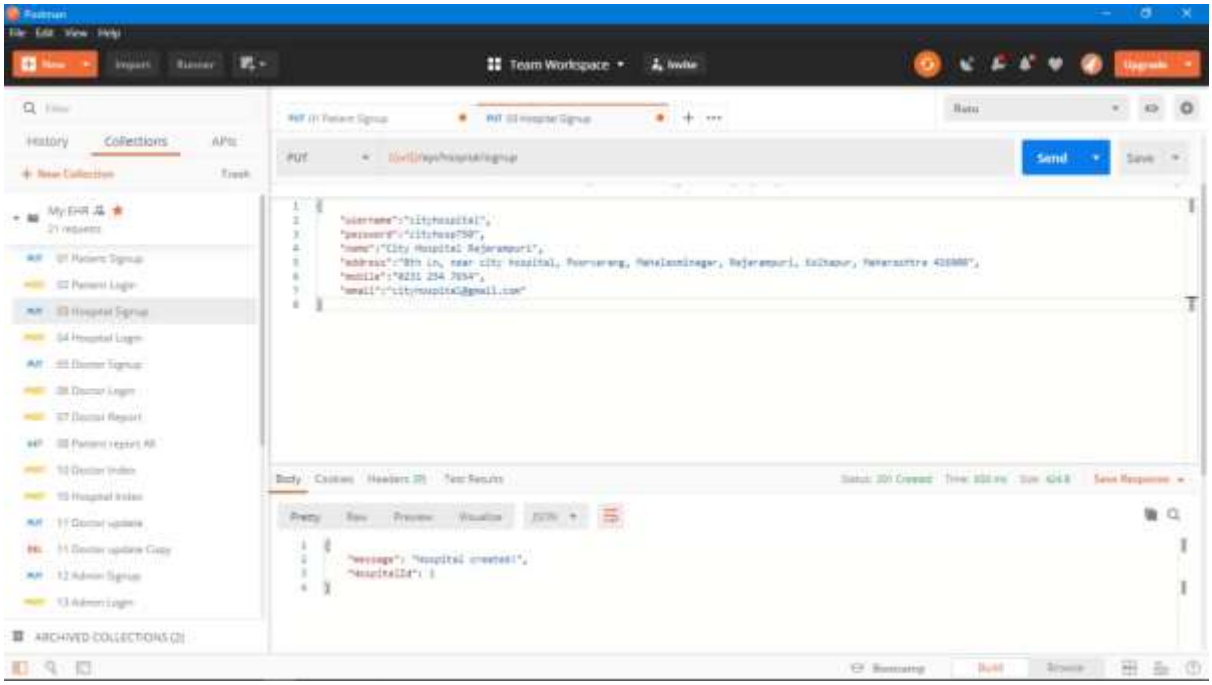
5. Testing

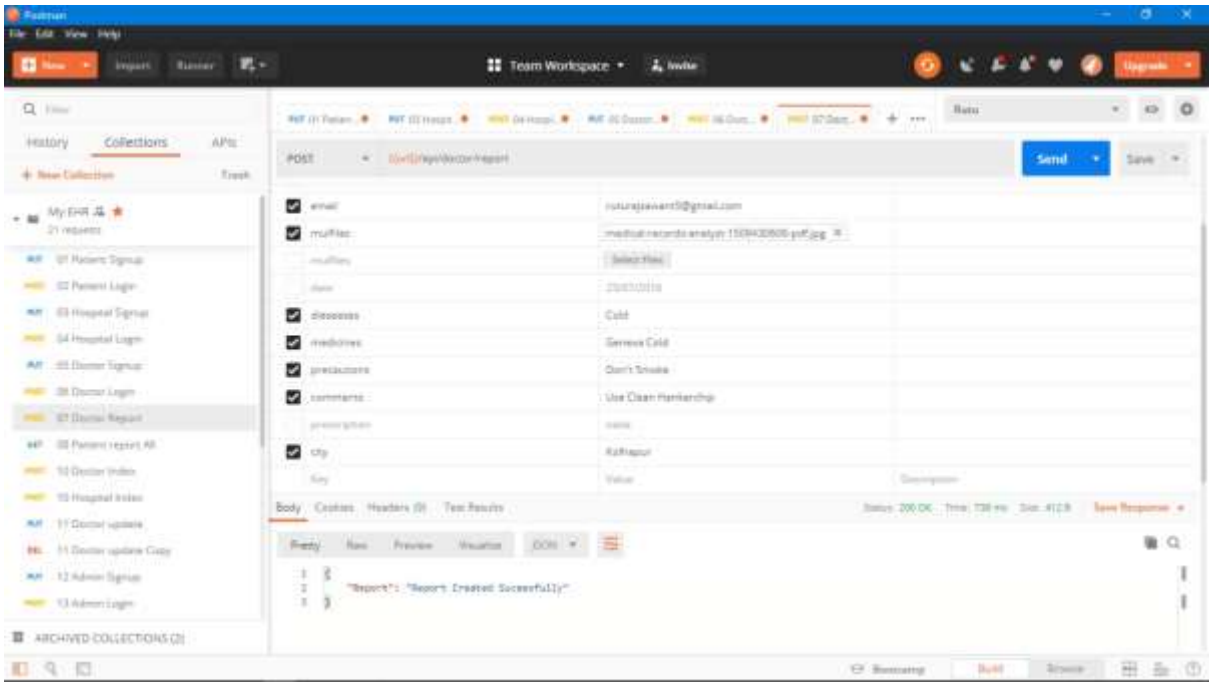
6.1 Testing mechanisms used

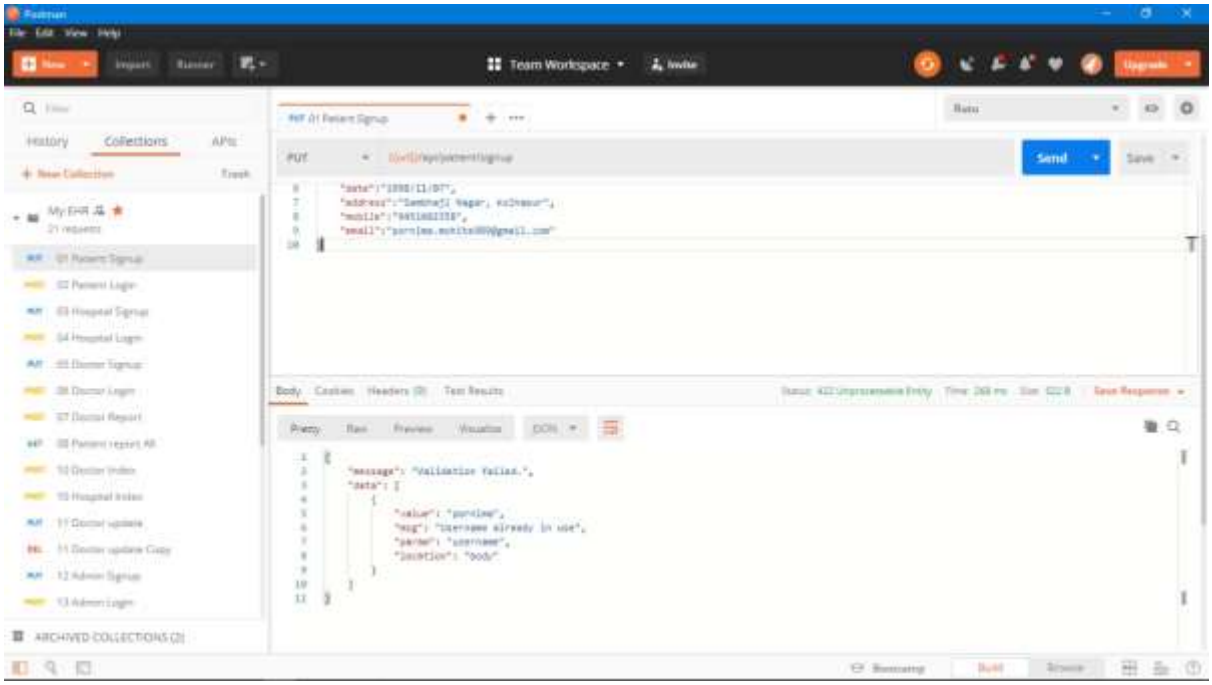
- Testing is done with the help of Postman to test login and sign up for patients, hospitals, report generation, and regional analysis.
- Postman is a scalable API testing tool that quickly integrates into CI/CD pipeline. It started in 2012 to simplify API workflow in testing and development. API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.
- Testing routes

1. Patient Signup
2. Patient Login
3. Hospital Signup
4. Hospital Login
5. Doctor Signup
6. Doctor Login
7. Doctor Report
8. Patient Report all
9. Doctor Index
10. Doctor Index
11. Doctor Update




Team Workspace
Invite
Send
Save
History
Collections
APIs
My EHR
"username":"cityhospital",
"password":"cityhosp750",
"name":"City Hospital Rajarampuri",
"address":"8th cn, near city hospital, Poorvarang, Mahalaxminagar, Rajarampuri, Kolhapur, Maharashtra 416008",
"mobile":"8231 254 7654",
"email":"cityhospital@gmail.com"
"message": "Hospital created!",
"hospitalId": 1


Team Workspace
Invite
Send
Save
History
Collections
APIs
My EHR
POST
email
rutujapawar0@gmail.com
diseases
Cold
medicines
Geneva Cold
precautions
Don't Smoke
comments
Use Clean Handwashing
city
Kolhapur
"Report": "Report Created Successfully"



5.2 Results obtained

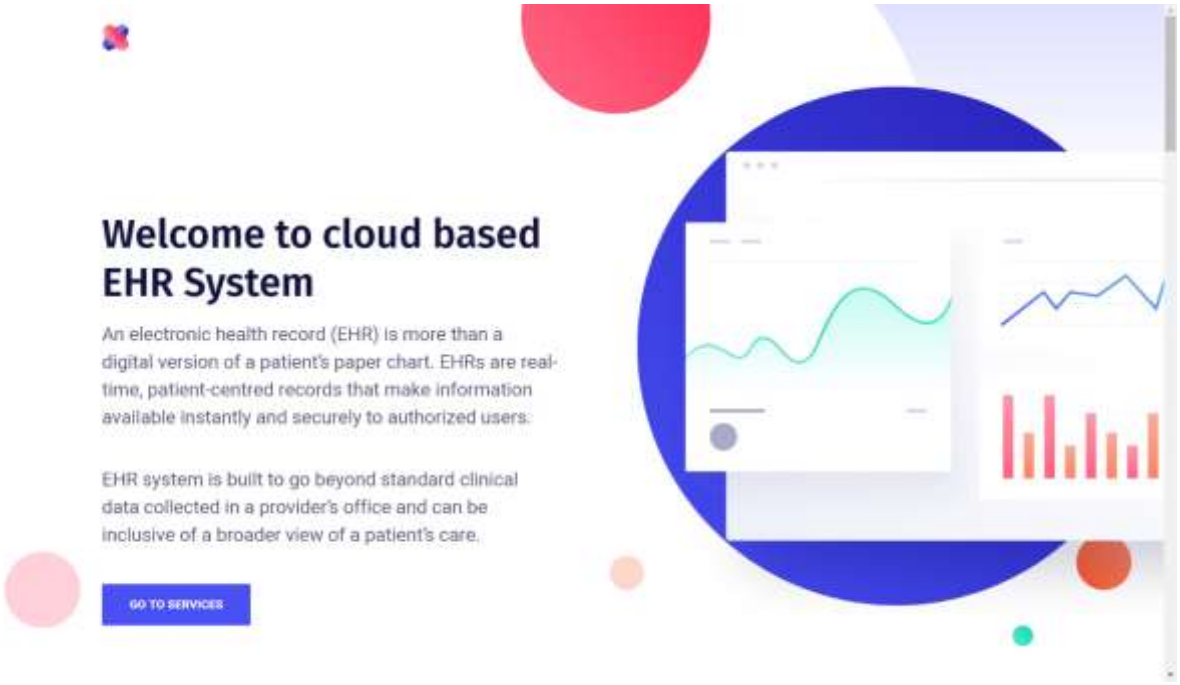


Fig. 5.2.1 Landing page

5.2.1 Landing page

Landing page provides user with system navigation and features.

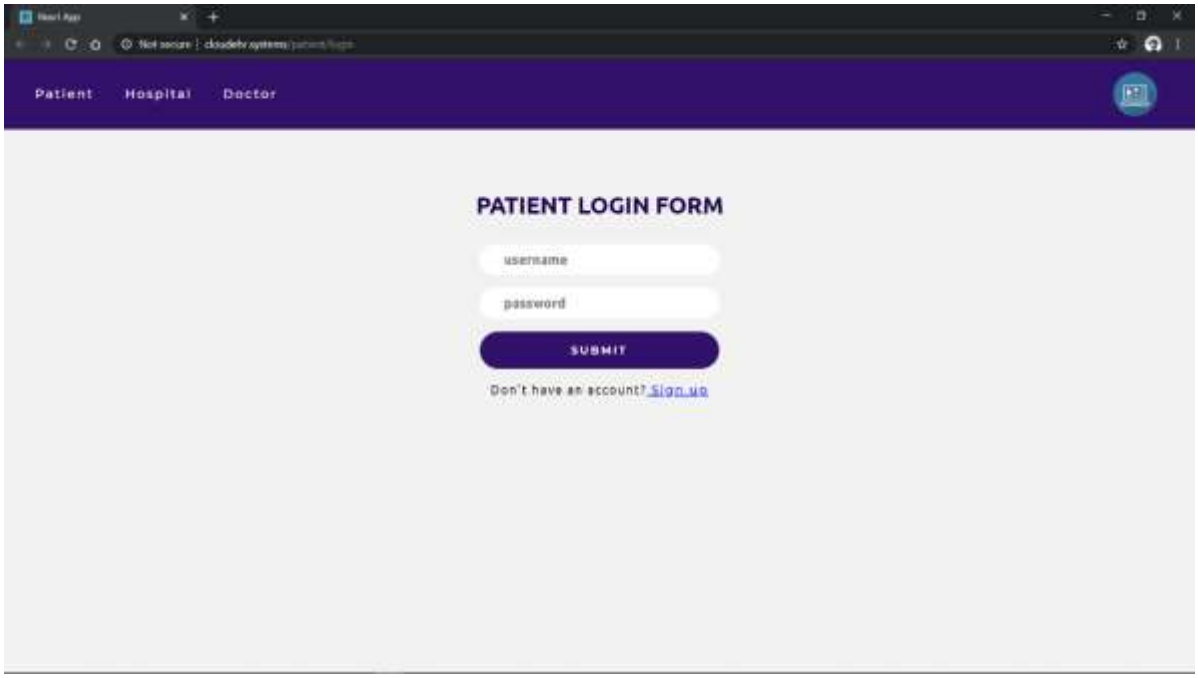


Fig. 5.2.2 Patients login page

5.2.2 Patients login page

Allows user to login as patient.

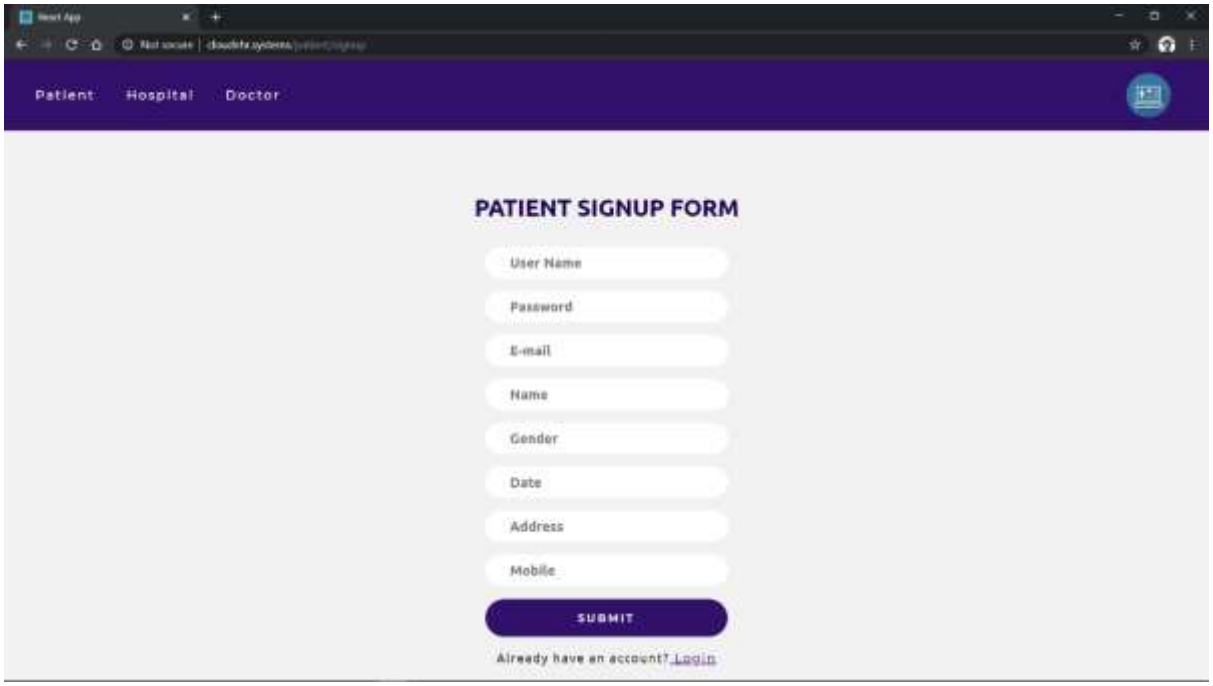


Fig. 5.2.3 Patients signup page

5.2.3 Patients signup page

Allows user to create a new profile as patient.

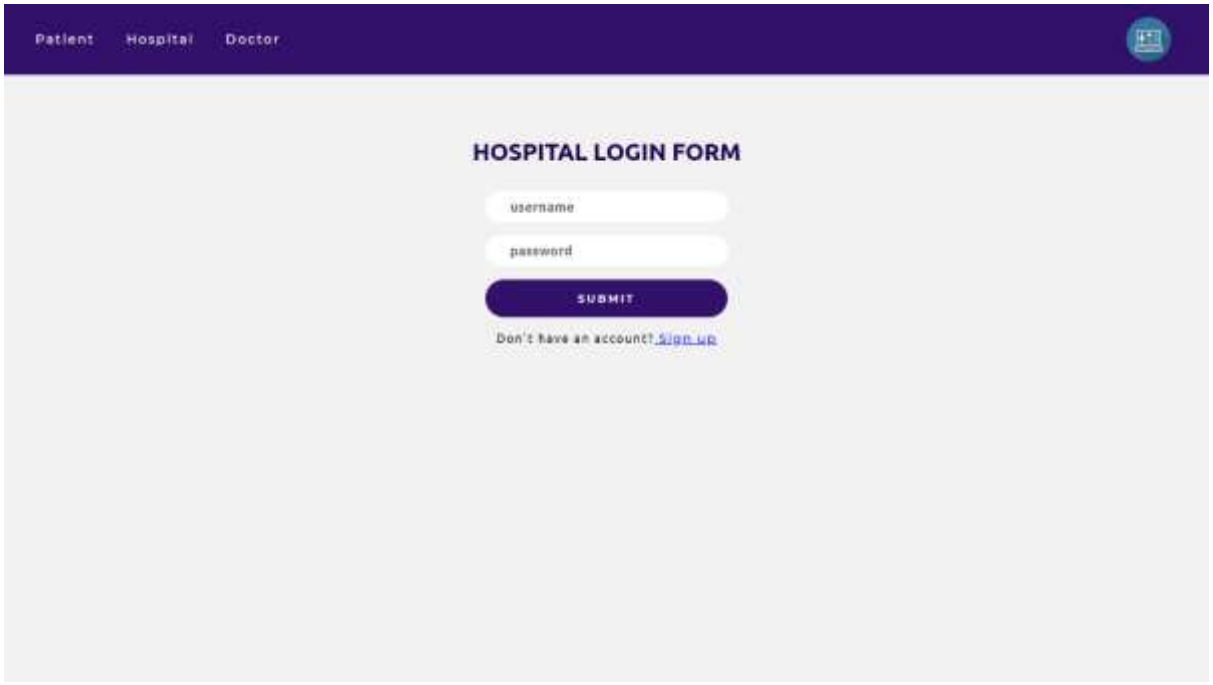


Fig. 5.2.4 Hospital login page

5.2.4 Hospital login page

Allows user to login as hospital.

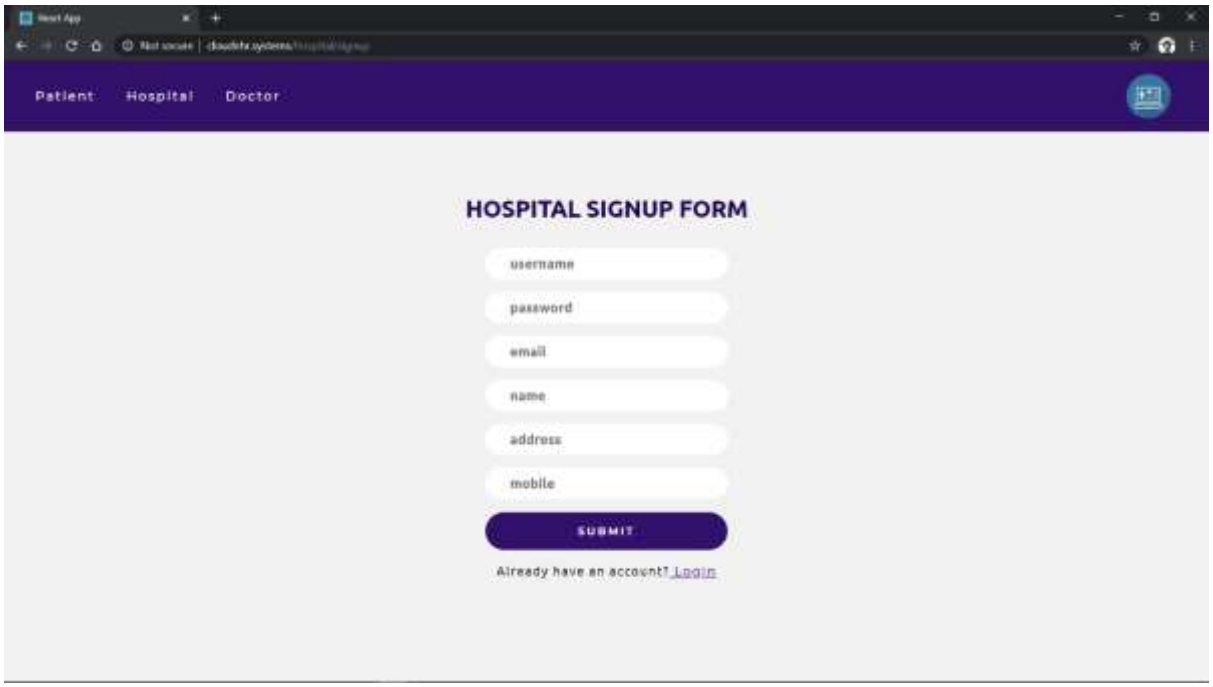


Fig. 5.2.5 Hospital Sign up page

5.2.5 Hospital signup page

Allows user to create a new profile as Hospital.

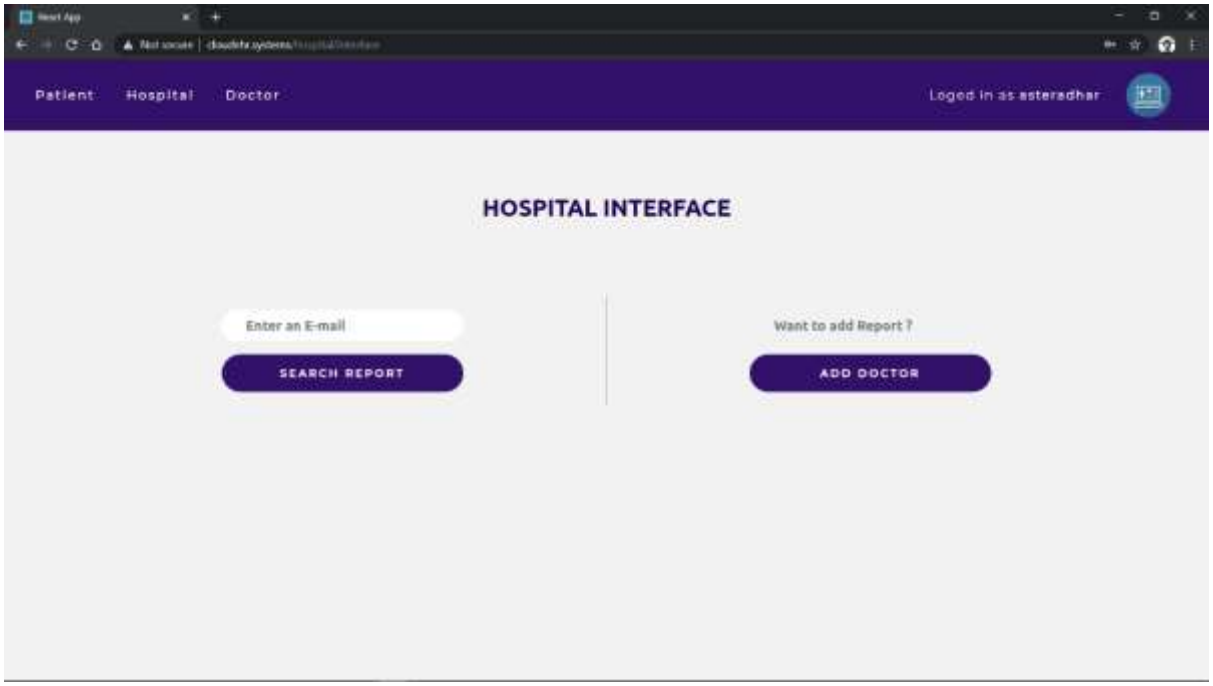


Fig.5.2.6 Hospital Interface

5.2.6 Hospital Interface

With the help of hospital interface doctors of respective hospitals can sign up and interact with patients.

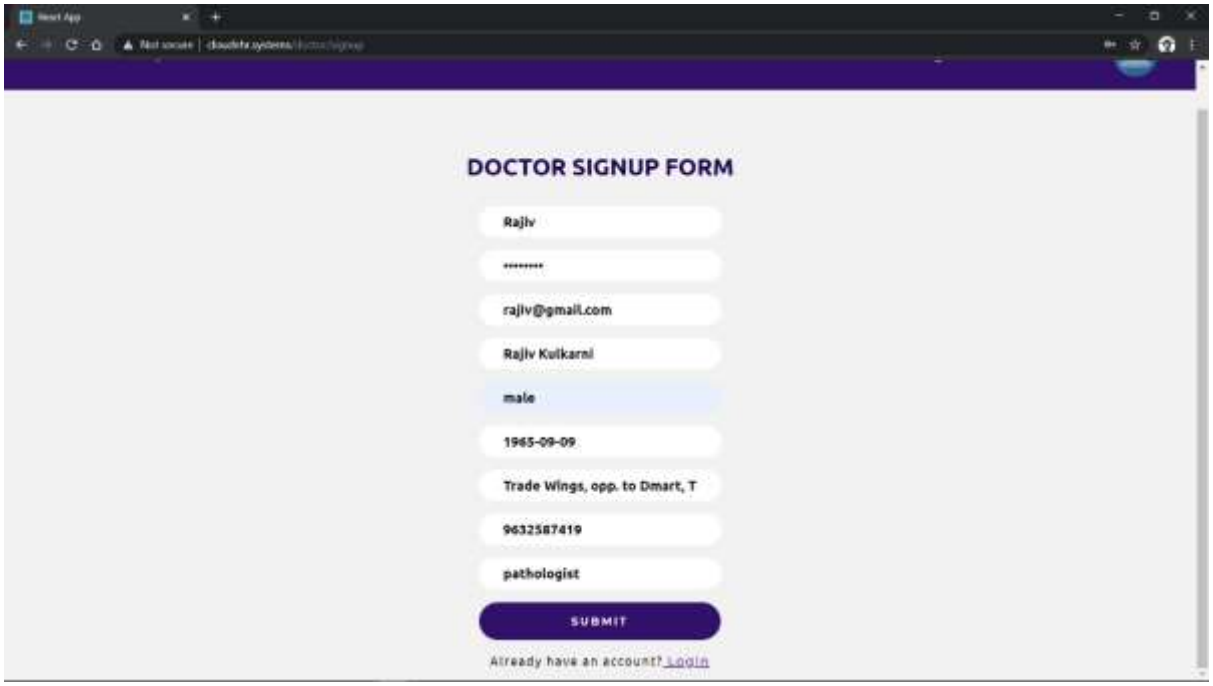


Fig. 5.2.7 Doctor Sign Up

5.2.7 Doctor Sign Up

Allows doctors in a hospital to sign up under that particular hospital.

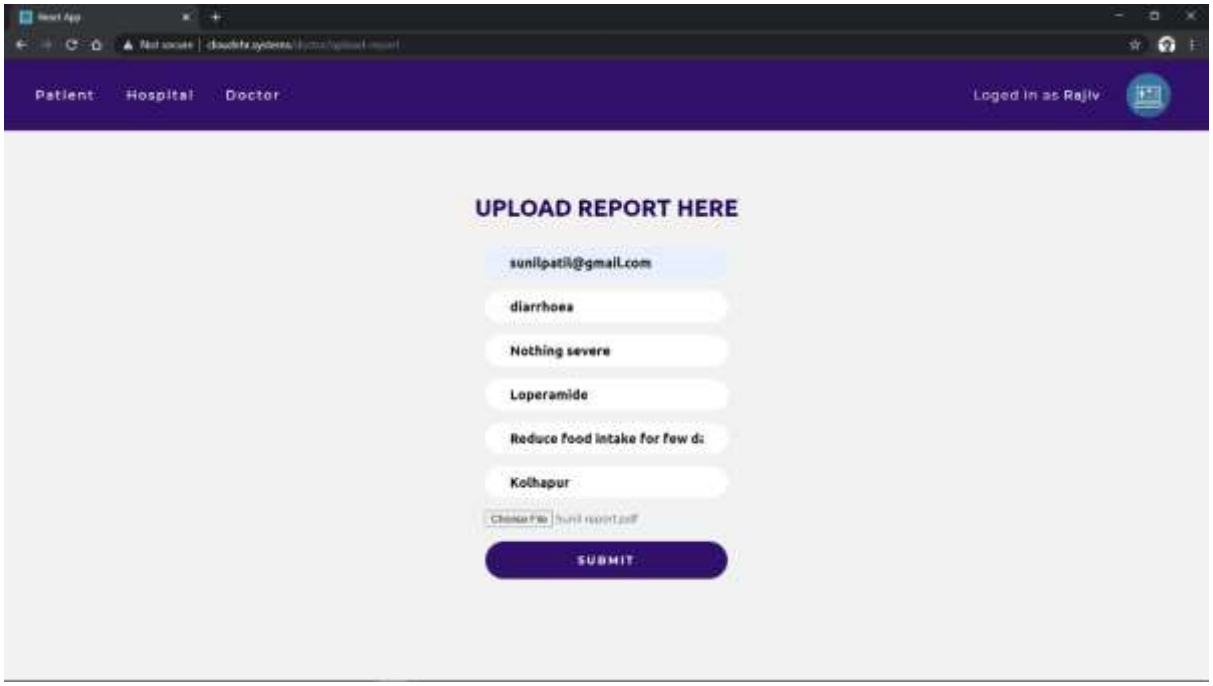


Fig. 5.2.8 Report Upload

5.2.8 Report Upload

With the help of report upload facility doctors can provide a detailed diagnosed report to their respective patients.

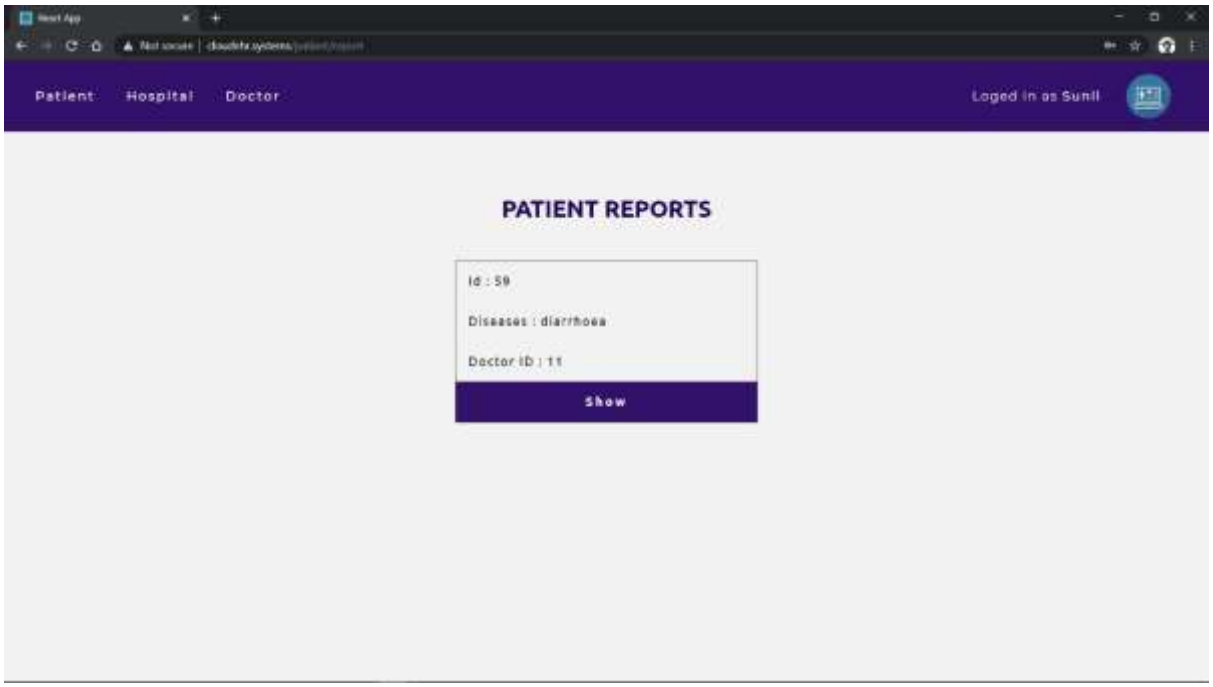


Fig. 5.2.9 Patient Report Access

5.2.9 Patient Report Access

Reports uploaded by the doctor can be accessed the in-patient dashboard.

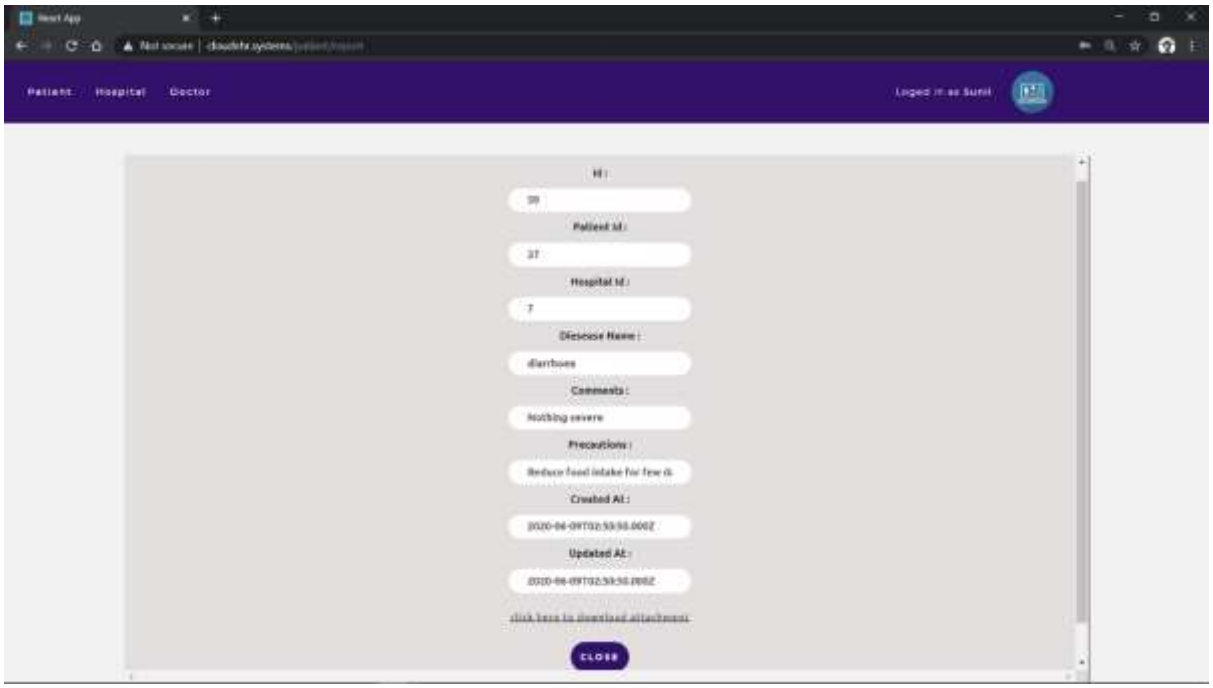


Fig. 5.2.10 Report details and download

5.2.10 Report details and download

Patient is provided with a privilege to download and access the details of his/her report.

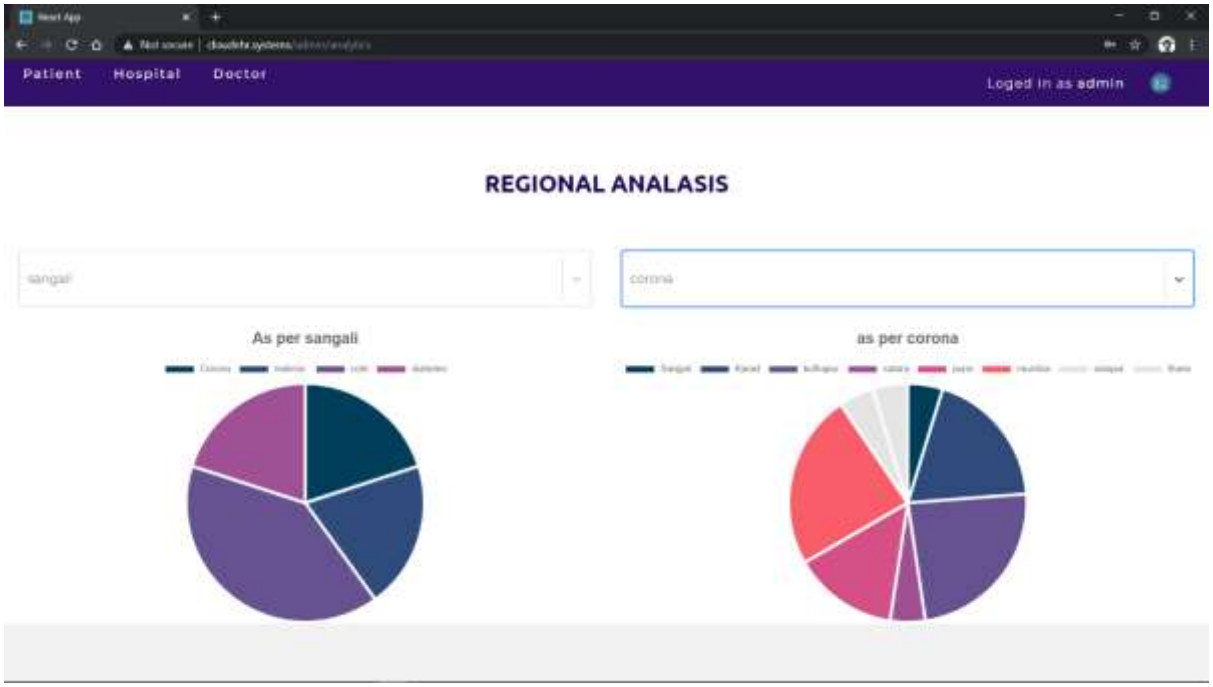


Fig. 5.2.11 Regional Analysis

5.2.11 Regional Analysis

Regional analysis is provided on the basis of diseases in cities.

Chapter 6

Conclusion & Future Work

6. Conclusion and Further work

6.1 Conclusion

Electronic health record (EHR) system hold great promise for improved patient care, but thus far they have not yet fulfilled their potential. Because contemporary EHR systems suffer from many shortcomings, the medical big data that they produce is also often flawed. Cloud based EHR System provides users with better patient care, efficient hospital management, and provides a whole view with the help of regional analysis.

6.2 Further work

1. Maintenance of system
2. System Upgrade
3. System Updates (feature expansion)
4. Security Improvisation
5. Improve client support

Chapter 7

References

7.References

7.1 Links:

1. https://nodejs.org/en/
2. https://expressjs.com/
3. https://reactjs.org/docs/getting-started.html
4. https://en.wikipedia.org/wiki/Data_Encryption_Standard

7.2 Paper:

Attribute-Based Encryption for Secure Access to Cloud-Based EHR Systems. Maithilee Joshi, Karuna P. Joshi and Tim Finin University of Maryland, Baltimore, County, Baltimore, MD 21250, USA Email: {maithi1 Karuna.joshi,finin}@umbc.edu.