

Project Synopsis Report

Centre for Development of Advanced Computing, Kolkata
(Ministry of Electronics and Information Technology, Government of India)



A PROJECT REPORT ON
E-Health Care System in Virtual Environment

SUBMITTED BY

Orchid Ganguly (170270620001)

Prithwiraj Dutta (170270620002)

Mohammad Adil (170270620003)

Manish Kumar Gupta (170270620004)

GUIDED BY

Mr. Sangam Kumar Chaturvedi
(Senior Technical Officer)
(ICT&S-02)C-DAC, Kolkata

SUPERVISED BY

Mr. Asok Bandyopadhyay
Joint Director (ICT&S-02),
C-DAC, Kolkata

ACKNOWLEDGEMENT

We express our sincere gratitude to Mr. Sangam Kumar Chaturvedi, Senior Technical officer of ICT&S-02, CDAC, Kolkata, for extending your valuable times for us to take up these problems as a Project and guide us throughout the project tenure.

We are also indebted to Mr. Asok Bandyopadhyay, Section Head, ICT&S-02, CDAC, Kolkata for his continuous help in executing the task smoothly.

We are also highly obliged to all the faculty members and employees of CDAC, Kolkata, for their critical advice and guidance without which this project would not have been possible.

We are grateful to Dr. Nabarun Bhattacharya, Director, C-DAC Kolkata, for giving me a chance to work at this esteemed organization.

Orchid Ganguly(170270620001)

Prithwiraj Dutta(170270620002)

Mohammad Adil(170270620003)

Manish Kumar Gupta (170270620004)



Centre for Development of Advanced Computing, Kolkata
(Plot: E2/1, Block: GP, Sector: V, Salt Lake Electronics
Complex, Kolkata- 700091)

CERTIFICATE

This is to certify that the project titled **E-Health Care System in Virtual Environment** using **ADVANCE JAVA** submitted by **Mr. Orchid Ganguly (170270620001)**, **Mr. Prithwiraj Dutta (170270620002)**, **Mr. Mohammad Adil (170270620003)** and **Mr. Manish Kumar Gupta (170270620004)** has been completed under our supervision for the partial fulfilment for **PG-DACC** course, academic session **February 2017 - August 2017** at **C-DAC, KOLKATA**. This report is hereby forwarded.

Mr. Sangam Kumar Chaturvedi
Senior Technical Officer
ICT&S-02,C-DAC,Kolkata

Mr. Asok Bandyopadhyay
Joint Director & Section Head
ICT&S-02,C-DAC,Kolkata

DECLARATION

I hereby declare that the project work entitled "E-Health Care System in Virtual Environment" submitted to the Centre for Development of Advance Computing (CDAC), Kolkata, is a record of an original work done by us under the guidance of Mr. Sangam Kumar Chaturvedi, Senior Technical Officer, ICT&S-02, C-DAC, Kolkata and under the supervision of Mr. Asok Bandyopadhyay, Joint Director, ICTS-02, C-DAC, Kolkata. This project work has been performed for the award of PG-DAC course of C-DAC, Kolkata only and this or any similar project will not be used for any other Degree or Diploma's associateship/fellowship.

Signature:

Name: Orchid Ganguly

Roll No:170270620001

Signature:

Name: Prithwiraj Dutta

Roll No:170270620002

Signature:

Name: Mohammad Adil

Roll No:170270620003

Signature:

Name: Manish Kumar Gupta

Roll No:170270620004

CONTENTS

CONTENTS	PAGE NO
Abstract	06
Introduction	07
Architecture	08
▪ Conferencing	09
▪ Multimedia based audio-video streaming	09
Design and block diagram	10
Implementation and deployment	11
Requirement specification	12
▪ Software requirements	12
▪ Hardware requirements	12
Installation guide	13
▪ Apache Tomcat Server Installation	13
▪ Node Server Installation	13
▪ MySQL Essential GUI Installation	14
Technologies used	15
▪ Internet	15
▪ HTML5	15
▪ WebRTC	17
▪ Node.js	18
Use case diagram	19
Data flow diagram	20
Codes	21
▪ Code flow	21
▪ Code page description	22
Code Snippet	23

▪ Home Page Code (home.jsp)	23
▪ Registration Code (RegAction.java)	23
▪ Database Insert Code (DocDao.java)	24
▪ Chat Code (ChatDto.java)	25
▪ Connection Code (Connect.java)	26
Project screen shots	27
▪ Database snapshot	27
• Doctor's table	27
• Patient table	27
• Report table	28
▪ GUI snapshot	28
Doctor's section	28
• Home page (home.jsp)	28
• Doctor's Registration Page (canvas.jsp)	29
• Doctor's Welcome Page (welcome.jsp)	29
• Chat GUI (welcome.jsp)	30
Patient Section	30
• Patient Registration Page (canvas.jsp)	30
• Audio Video Conferencing (demo_audio_video_simple.html)	31
Conclusion	32
Scope for future enhancement	33
References	34

ABSTRACT

Implementation of an **E-Healthcare System in Virtual Environment** is an online system where doctor and patient both can interact to each other. It is observed that distance is a critical barrier for providing medical support in remote areas. This system will help to provide medical support to the people living in the rural areas or a remote location.

The goal is to design a user friendly, responsive and easy to maintain platform with a creative and interactive GUI supported by back-end data for both patient and doctor. The basic functionalities of the system includes Text chat, Video call, Online prescription, Medical document upload etc.

In this project we have used some latest technologies namely HTML, CSS, JSP and servlet for the front end development, MySQL for the backend database part and node.js is used for the video call interface.

INTRODUCTION

Telemedicine is the use of telecommunication and information technologies in order to provide clinical health care at a distance. It helps eliminate distance barriers and can improve access to medical services that would often not be consistently available in distant rural communities

Telemedicine service can be made effective using web based interaction where a centralized database contains the information about doctor and patient. System also includes some peripherals like computer, webcam and speaker etc. which doctors and patient can use to communicate with each other.

The approach of this system is basically to establish a communication system which enables users to communicate with each other like other web enabled application. But it is differ from those web applications. The main objective is it will serve the e-health care aspects. Here in this system user (Doctor) can communicate with another user (Patient) via audio-video conversation over the internet so that doctors will be able to write the proper prescription to send it to the corresponding patients or user.

The system is developed to spread the interactive system globally whereas the previous system is only worked in a geographical area that means it only works in LAN. So any user can access that system from anywhere.

One medical assistance will be available at patient station that will be doing the work as per doctor's instruction.

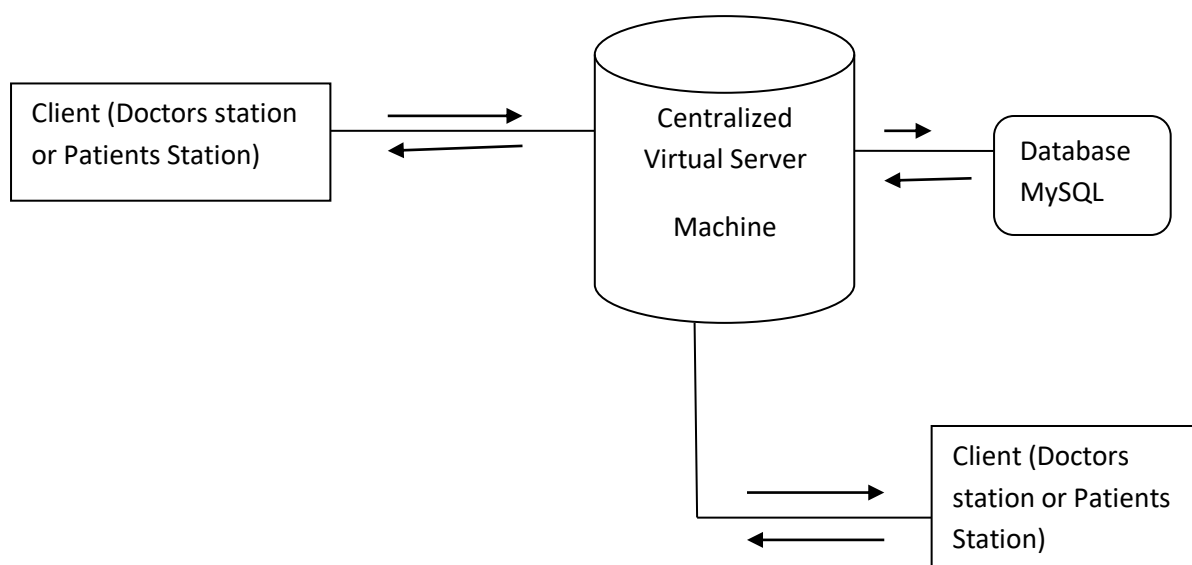
Telemedicine has the potential expedite and improve the delivery of high quality, cost effective care by extending the reach of health care practitioners beyond their local setting and using advanced information technologies.

The system the very first thing which we have to know what type of system it is. This system is basically a web application which is the most interactive application now a day because in this world every people can know the term internet and maximum of them can be able to use it frequently. Only this type application can run over the internet. So we can say using internet we can easily surf so many web application. Now the answer is cleared to everyone why we make this system as web application because it enables a large number

of users that is doctors and patients to access it over internet and use it. This telemedicine application provides a platform where a doctor can easily see and hear his or her patient and then prepare the actual prescription and send it to the corresponding patient so that he or she can download the prescription and take medications accordingly. This system will omit the distance barrier between the patient and the doctor. The people from the rural areas where good treatment is not available will be able to consult with several doctors and take their advice regarding their problems. The doctors will be able to reach a large number of patients without being physically present in a place.

ARCHITECTURE

The architecture of the system is based on client – server system architecture. It describes the user interaction with the web server or the server machine.



System Architecture

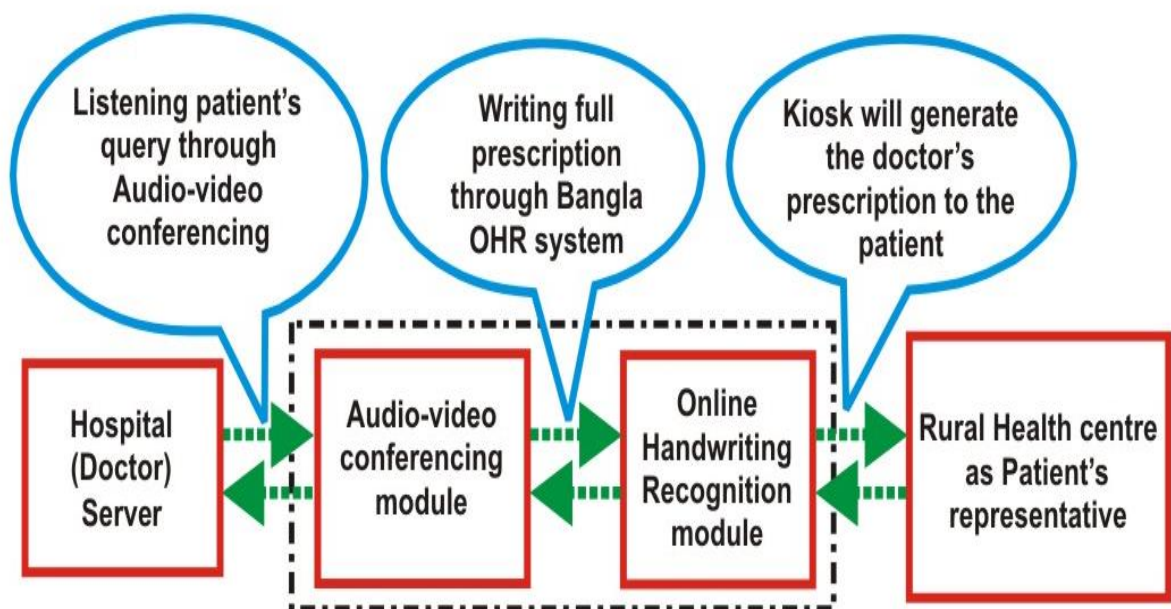
According to the above diagram the system follows the Client-Server architecture. Here in this diagram the rectangle boxes represent the Client machine that basically the patient or doctor station. The cylindrical figure represents the Server machine by which the communication established among two clients or user and the rounded rectangle represent the database that is the MySQL server which is used in this project. All the arrow represents the requests and responses to and from the server. The communication between server and database takes place whenever a particular query needed in the form of client request.

Conferencing

The use case below represents a Web-Conferencing scenario built with lightweight HTML5 components and WebRTC APIs with no plug-in installation. Such an application can allow plug-n-play of components such as chat, file transfer, screen share with few lines of JavaScript code.

Multimedia based audio-video streaming

This scenario enables audio, video, and data streams into environments with WebRTC, HTML5 and WebGL APIs. Such a combination provides options for combining real-time media with WebGL canvas innovatively



DESIGN AND BLOCK DIAGRAM

The system follows a traditional MVC architecture i.e Model, View and Controller.

MODEL:

The model part consists of the Back-end Database i.e. MYSQL Engine and classes which perform database oriented tasks. The Model package is used for the database connection set up. It consist of a class called connect which set up the connection between the database and the application.

View:

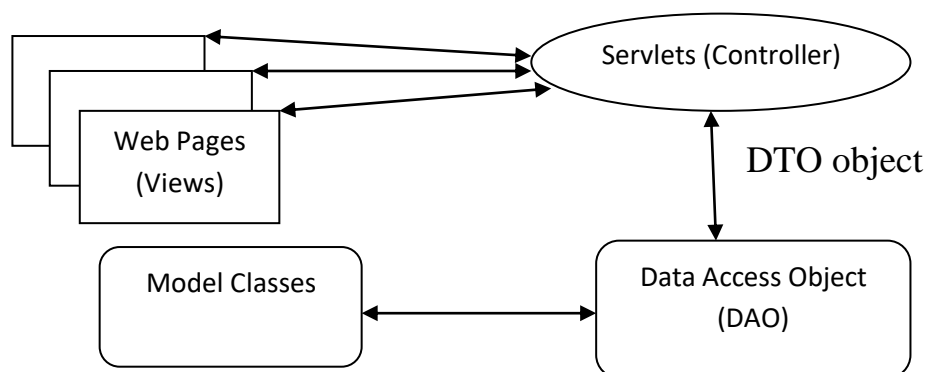
The view part consists of a number of web pages which are shown to the users. Here in this system the view part includes a number of JSP pages which accepts data or some particular request from the users.

Controller:

The controller portion consists of the Servlets which basically works as a middleware between the view and the model. It accepts the request from the user and process it accordingly in order to retrieve the result from the database.

There are also two additional packages are used to implement this MVC model and they are **DTO (Database Transfer Object)** package and **DAO (Database Access Object)** package.

DTO package contains classes which are used to carry data from webpage and send it to the servlet for a particular request and the DAO package is used to retrieve and insert data into the database.



MVC Architecture (In Block Diagram)

IMPLEMENTATION AND DEPLOYMENT

The system is implemented to serve the Telemedicine aspects over environment. To implement that system we here used the Java EE environment. There are various advantages of that environment for implementing the system. These are:-

1. Developers Productivity,
2. Feature Rich Platform, Java EE platform provides a rich set of features out of box, avoiding explosion of non-standard, non-integrated APIs. The result is pretested, pre-integrated platform offered by multiple vendors.
3. Web Profile, for applications that do not need the power and enterprise connectivity of the full Java EE platform, Java EE 6 introduced the lightweight Web Profile optimized for web applications. Developers can rapidly create rich user interfaces that integrate with transactional business components and databases for departmental or outward-facing web applications.
4. Faster solutions and delivery time to market.
5. Complete web service support,
6. Freedom of choice,
7. By offering one platform with faster solution delivery time to market, freedom of choice, and simplified connectivity, the Java EE platform helps IT by reducing Total Cost of Ownership (TCO) and simultaneously avoiding single-source lock-in for their enterprise software needs

REQUIREMENT SPECIFICATION

There are software and hardware requirements which needed to implement the project.

Software Requirements

1. Java Development Kit(JDK) 1.7,
2. Apache Tomcat 7.0,
3. MySQL server 1.5,
4. Net Beans IDE 8.2,
5. Web Browser.

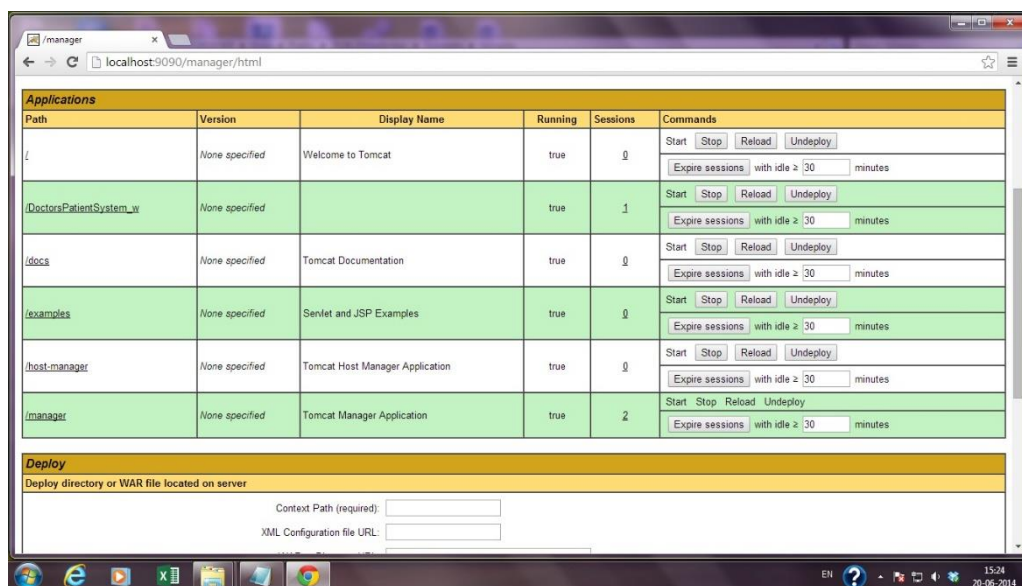
Hardware Requirements

1. 4 GB RAM,
2. Intel i5 processor,
3. Hard Disk 1 TB,
4. Advance Digital Web Cam,
5. High Definition Audio Speaker.

INSTALLATION GUIDE

1. Apache Tomcat Server Installation

- Double click on apache-tomcat-7.0.42.exe from
E:\PGDAC\Telemedicine\2017-05-29 Project Telemedicine\Softwares
- Change the HTTP connector port as 9090-->give username
(admin)andpassword (admin).
- Open chrome browser and give this address "localhost:9090"
- In Tomcat Click on Manager App.Give username(admin) and
password(admin) of Tomcat server
- WAR file to deploy-->click on choose file-->choose the WAR file and
click on delpoy
- Click on the link of application named -->"/doctorspatientsystem" in
<http://localhost:9090/manager/html>



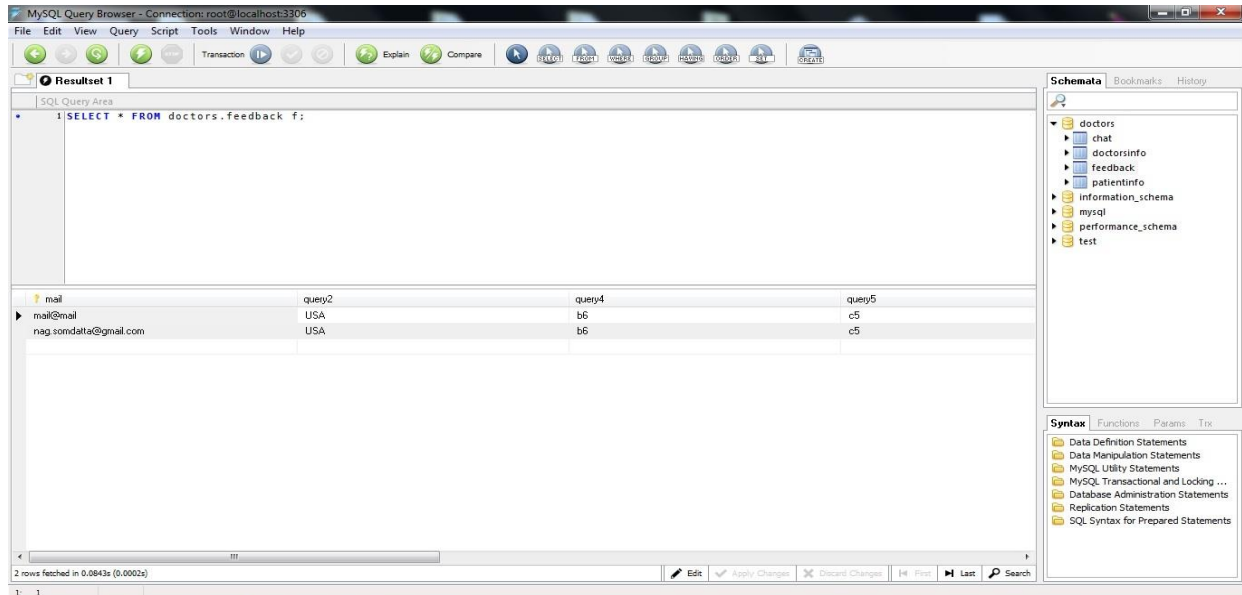
Apache Tomcat Server in Browser

2. Node Server Installation

- Install node.js
- Unrareasyrtc to c:\users\nodes
- Change name from easyrtc-master-->easyrtc
- In command prompt cd node\easyrtc
- Node\easyrtc>npm install
- Node\easyrtc\server_example node server.js
- Server will be started.

3. MySQL Essential GUI Installation.

- a. Install MySQL Essential server.
- b. Install MySQL query browser.
- c. Restore .sql file in mysql server.



Database Snapshot in MySQL Query Browser

TECHNOLOGIES USED

Internet

This software can also be access through internet when it is applicable for the users in allover the world.

The system is designed and coded on NetBeans IDE 6.8. This whole system works on the Apache Tomcat Server. The database created in MySQL server 1.5. Here the corresponding database name is Doctors which consists of the main table “doctor’s info (id, name, pswd, loginStatus, path)”. At the time of implementation the project the Apache Tomcat server used which is incorporate within the Net Beans IDE. After built the project simply the .war file is created of the project and then just deploys it in the server machine, so that the entire user can access it properly.

HTML5

Web Conferencing Using HTML5

Two-way video conferencing from each sides with supported peripherals. An audio-video conference can be set up between doctor and patient, when both of them are online over internet. Based on three API (GetUserMedia API, PeerConnection API, DataChannel API) web conferencing is achieved.

HTML:Hyper Text Markup Language (HTML) is the encoding scheme used to create and format a web document. A user need not be an expert programmer to make use of HTML for creating hypertext documents that can be put on the internet. HTML is the language of communication that flies over the network. Although files are not necessarily required to be stored in HTML formatting, it is often found the easiest way to do so. Servers may store files in other formats or in variations on HTML that include extra information of local interest only, and then generate HTML on the fly with each request.

The Web is evolving at a pace faster than ever before. The last few years has seen tremendous innovations in the Web Technologies, Applications, Infrastructure and Services. The advent of HTML5 has redefined the way Web

Applications work by bringing in the capabilities & richness of native applications to the Web platform.

HTML5 technologies such as Web Workers, Browser-Native Media, Web Sockets and the like are redefining the roles and capabilities of the browser and the Web, and creating experiences that rival native applications.

Building along similar lines, is the introduction of WebRTC/RTCWeb technological standards into the HTML5 standards basket, which is concerned with bringing rich real-time, interactive communications natively to the browsers.

Real-time communications applications like softphones, conferencing applications are not new to the Web. Applications such as WebEx, JabberWeb and Skype already provide ways for people to communicate and collaborate on the Web today.

These applications do come with limitations however:

1. Need to install plugin to get things working.
2. With plugins come the challenges of compatibility on the host platform.
3. With plugins comes along security issues, since the application no longer runs in the sandboxed environment of the browser.
4. With plugins comes the issue of maintenance. Newer versions of the browser or standards might break existing installations.
5. Applications based on plugins lack the rich flexibility of native browser resources due to privilege restrictions. This in turn limits the innovation possible with these applications.
6. A proprietary solution brings in interoperability issues.

WebRTC

RTCWeb (Owned by IETF) and the WebRTC (Owned by W3C) standards is an evolving proposal to bringing the “Rich Interactive Secure Peer to Peer Communications” to the Web in a “Plugin-less Fashion”. These standard bodies together are responsible for defining the following aspects for enabling real-time communications as inherent part of the web infrastructure.

1. APIs and access rules for end-user devices such as microphones, cameras etc.
2. End-to-end security architecture and protocol.
3. NAT traversal techniques for peer connectivity.
4. Signalling mechanisms for setting up, updating and tearing down the sessions.
5. Support for different media types.
6. Media transport requirements.
7. Quality of Service, congestion control and reliability requirements for the session over the Best-Effort Internet.
8. Identity architecture and mechanisms for peer identification.
9. Codecs for audio and video compression.
10. Last but not the least, HTML and JavaScript APIs enabling application developers.

With such a detailed charter, WebRTC/RTCWeb has the potential to impact the way people communicate on the web. With the tremendous increase in the usage of browsers and always available nature of the Web, the combination of “Browser and the Web” revolutionizes real-time communications on one end and possibly poses potential challenges to legacy/traditional solutions of today.

1. GetUserMedia API specification defines requirements for a Web application to access end-users media sources such as camera, microphone
2. PeerConnection API specifies SDP-based session description APIs and the state machine to session setup, update and tear-down between the peers.
3. Data Channel API will enable peer-to-peer exchange of arbitrary data, with low latency and high throughput.

Under the hood, the browser is responsible for:

1. Ensuring end-to-end security for media and data sessions via DTLS.
2. Performing NAT traversals procedures for connection setup based on Interactive Connection Establishment (ICE).
3. Establishing media transport based on RTP and UDP.
4. Setting up data-channel transport based on SCTP and UDP
5. Enabling feedback reports for the session based on RTCP.
6. Encoding and decoding audio and video streams.

easyRTC'sWebRTC Server

The video conferencing developed on easyRTC, a bundle of Open Source WebRTC joy, incorporates an easyRTC server install and client API, and working, HTML5 and Javascript, application source code under a BSD 2 license.

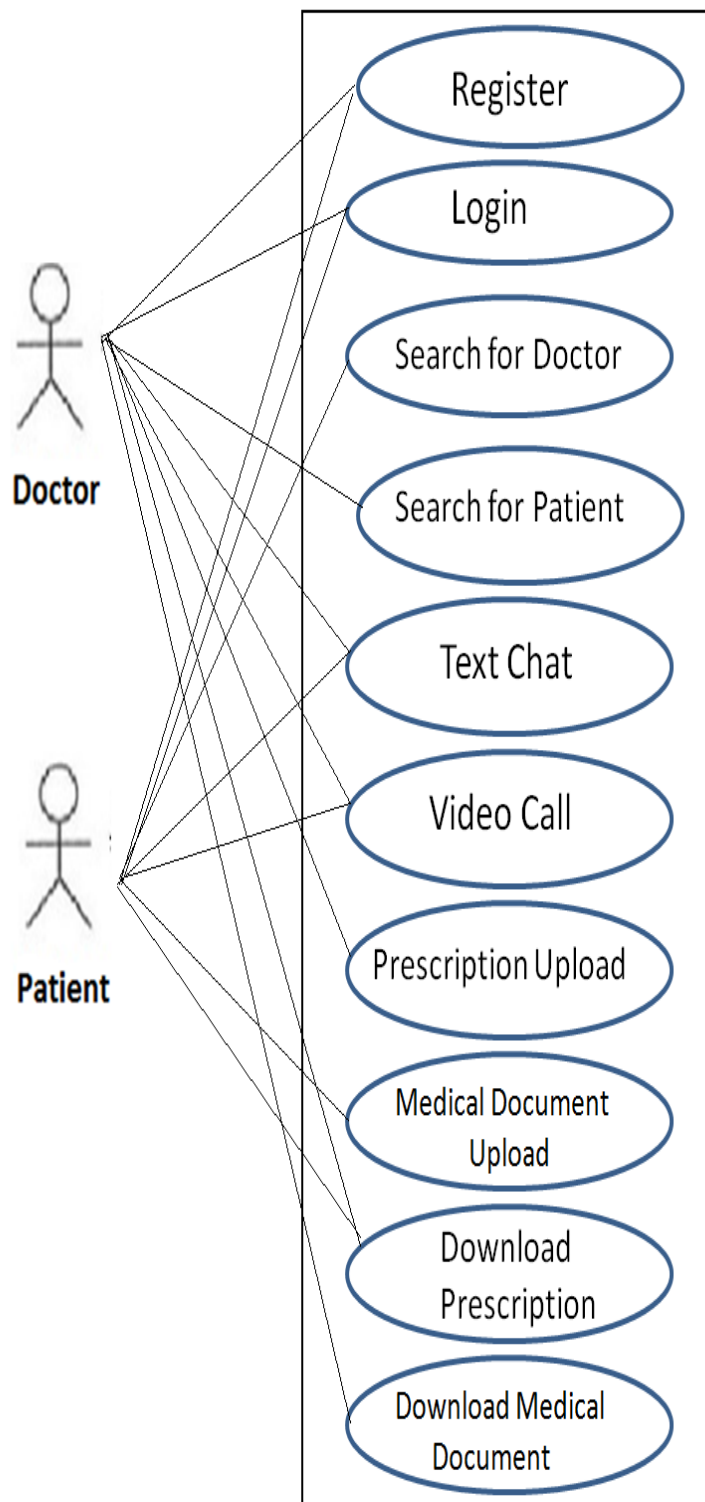
Features of easyRTC

1. Install easyRTC'sWebRTC Server Kit on our Windows, server in minutes not days.
2. Used easyRTC API and sample application code to build and deploy our WebRTC app in hours not weeks.
3. easyRTC is completely free and open source under a BSD 2 license. No usage costs or other hidden fees.

Node.js

Node's goal is to provide an easy way to build scalable network programs. Node tells the operating system (through `poll`, `kqueue`, `/dev/poll`, or `select`) that it should be notified when a new connection is made, and then it goes to sleep. easyRTC'sWebRTC server is hosted by Node.js. easyRTC is started by running `node server.js`.

USE CASE DIAGRAM



DATA FLOW DIAGRAM

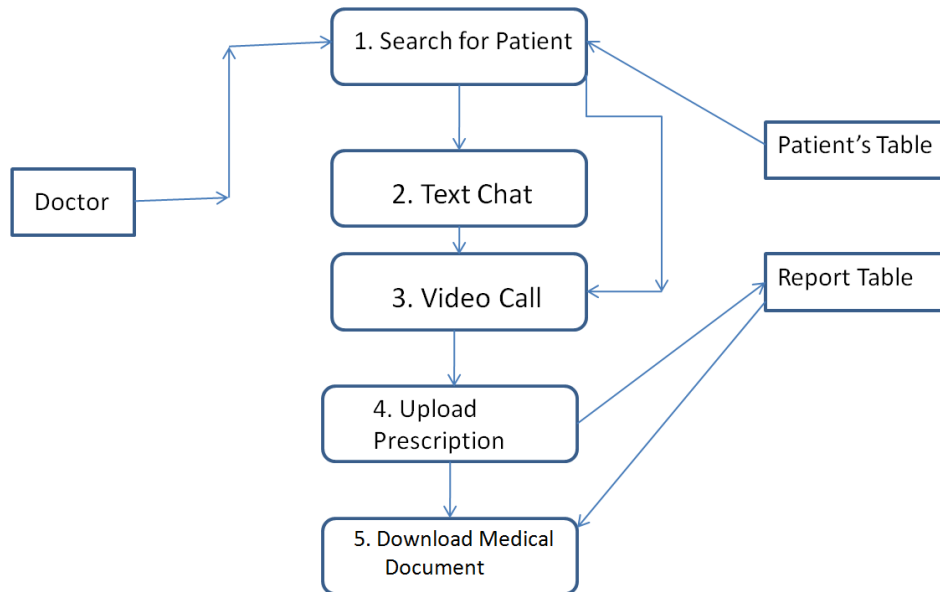


Fig 1: DFD for Doctor Login

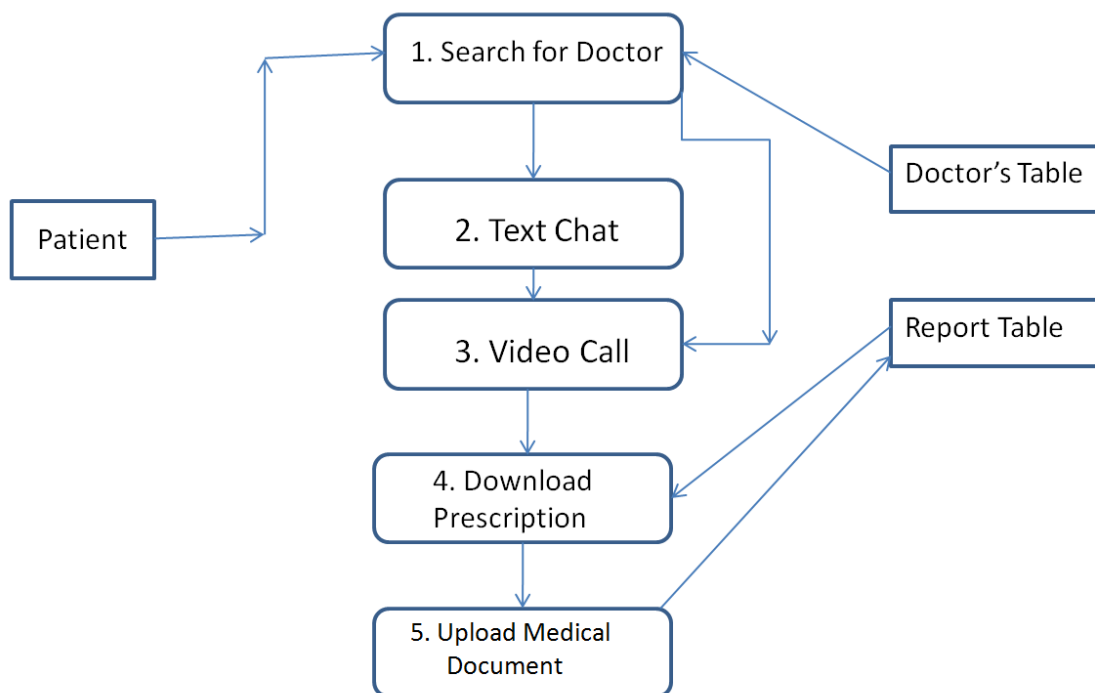
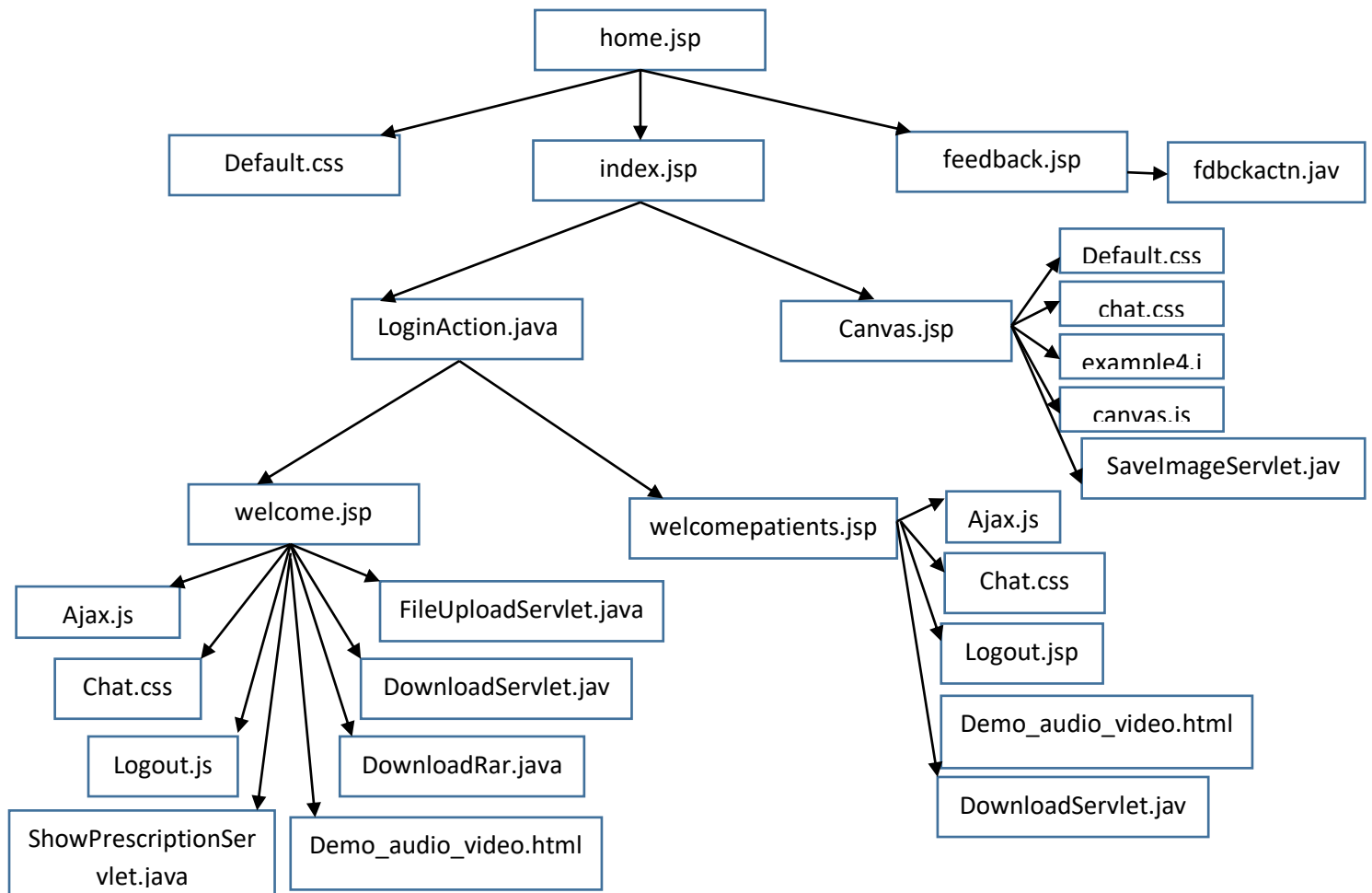


Fig 2: DFD for Patient Login

CODES

Code Flow



Code Flow Guide

Code Page Description

1. Home.jsp→front page of the application
2. Default.css→ style sheet file of html and jsp page backgrounds.
3. Index.jsp→login page of the application.
4. Feedback.jsp→users feedback form page of the application.
5. Loginaction.java→java servlet code being login and associated activities.
6. Canvas.jsp→new user registration page.
7. Welcome.jsp→doctor's page after login.
8. Welcomepatient.jsp→patient's page after login.
9. Ajax.js→ asynchronous JavaScript and xml odes for availability checking and associated activities.
10. Logout.jsp→jsp page after logging out of the application.
11. Showprescriptionservlet.java→ in doctor's page for viewing the pdf formatted Bengali prescription with respective doctor's signature and information's.
12. Fileuploadservlet.java→ in doctor's page for uploading the html prescription and changing it to pdf format.
13. Downloadservlet.java→Downloading prescription in pdf format in both doctor's and patient's end.
14. Downloadrar.java→downloading the application in .rar.
15. Demo_audio_video.html→for starting multimedia conferencing in both doctor's and patient's end using node server.
16. Example4.js→for doing signature.
17. Canvas.js→for editing signature and saving it.
18. Saveimageservlet.java→for new user's registration.
19. Fdbckactn.java→for user's feedback collecting action.

Code Snippet

Home Page Code (home.jsp)

```
<script type="text/javascript" language="javascript">
function call(){
alert("For Best Functionality, Please use Google Chrome");
varans=confirm("Do you want to Install Chrome?");
if(ans==true){
window.open("https://www.google.com/intl/en/chrome/browser/");
}
}
</script>
</head>
<body onload="call();">
<div id="wrapper">
<div id="header">
<div id="logo">
<%-- <h1><a href="#"><font face="brush script mt" size="6">E-HEALTHCARE</font></a></h1>--
%>
<p><imgsrc="images/ehealthcare.jpg" width="100%" height="auto" alt="E-HEALTHCARE"/></p>
</div>
<div id="menu" class="topnav">
<a href="home.jsp">Homepage</a>
<a href="feedback.jsp" >Your Feedback</a>
<a href="#" >About Us</a>
<a href="http://www.cdackolkata.in/">Contact Us</a>
<a href="javascript:void(0);" onclick="myFunction()"></a>
<%-- <imgsrc="images/building.jpg" width="200" height="85" alt="" />--%>
</div>
</div>
<div id="page">
```

Registration Code (RegAction.java)

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
DocDtoob = new DocDto();
ob.setId(request.getParameter("id"));
ob.setName(request.getParameter("name"));
ob.setPassword(request.getParameter("password"));
ob.setType(request.getParameter("select"));
sel=request.getParameter("select");
out.write("type"+sel);
DocDao ob1 = new DocDao();
if (request.getParameter("select").equals("Doctors")) {
if (ob1.insert(ob) == 1) {
```



```

response.sendRedirect("registration.jsp?msg=Registered Suceessfully");

        } else {
response.sendRedirect("registration.jsp?msg=Registration Failed");
        }
    } else {
if (ob1.insertpat(ob) == 1) {
response.sendRedirect("registration.jsp?msg=Pat Registered Suceessfully");

        } else {
response.sendRedirect("registration.jsp?msg=Registration Failed");
        }
    }
}

try {

    } finally {
out.close();
    }
}

```

Database Insert Code (DocDao.java)

```

public class DocDao {

    Connection con=null;
    PreparedStatementps=null;
    ResultSetsrs=null;
    publicint insert(DocDtoob){
    try {
con=Connect.MyConnect();
        // if(ob.getType().equals("Doctors")){
ps = con.prepareStatement("insert into
doctorsinfo(id,name,pswd,loginStatus,path,type,email,Address,Phone,Occupation,Institution,active)values(?,?,?,
?,?,?, ?, ?, ?, ?, ?)");
ps.setString(1,ob.getId());
ps.setString(2,ob.getName());
ps.setString(3,ob.getPassword());
ps.setInt(4,0);
ps.setString(5,ob.getPath());
ps.setString(6,ob.getType());
ps.setString(7,ob.getEmail());
ps.setString(8,ob.getAddress());
ps.setString(9,ob.getPhone());
ps.setString(10, ob.getOccupation());
ps.setString(11, ob.getInstitution());
ps.setInt(12,0);
        // ps.setString(13,ob.getfeedback());
        //}
        /*else{
ps = con.prepareStatement("insert into doctorsinfo(id,name,password) values(?,?)");

```

```

ps.setString(1,ob.getId());
ps.setString(2,ob.getName());
ps.setString(3,ob.getPassword());
    }*/
int c=ps.executeUpdate();
if(c==1){
return 1;
    }
    } catch (SQLException ex) {
Logger.getLogger(DocDao.class.getName()).log(Level.SEVERE, null, ex);
    }
return 0;
    }

public int insertpat(DocDto ob){
try {
con=Connect.MyConnect();
    // if(ob.getType().equals("Doctors")){
ps = con.prepareStatement("insert into
patientinfo(id,name,pswd,loginStatus,path,type,email,Address,Phone,Occupation,Institution,active)values(?,?,?,
?,?,?, ?, ?, ?, ?, ?)");
ps.setString(1,ob.getId());
ps.setString(2,ob.getName());
ps.setString(3,ob.getPassword());

```

Chat Code (ChatDto.java)

```

public class ChatDto {
    String sender=null;
    String reciever=null;
    String msg=null;
    Timestamp ts=null;

    public Timestamp getTs() {
return ts;
    }

    public void setTs(Timestamp ts) {
this.ts = ts;
    }

    public String getMsg() {
return msg;
    }

    public void setMsg(String msg) {
this.msg = msg;
    }

    public String getReciever() {
return reciever;
    }

```

```
public void setReciever(String reciever) {  
    this.reciever = reciever;  
}
```

```
public String getSender() {  
    return sender;  
}
```

Connection Code (Connect.java)

```
public class Connect {  
    public static Connection MyConnect(){  
        Connection con = null;  
        try {  
            Class.forName("com.mysql.jdbc.Driver");  
            try {  
                con = DriverManager.getConnection("jdbc:mysql://localhost:3306/doctors", "root", "root");  
            } catch (SQLException ex) {  
                Logger.getLogger(Connect.class.getName()).log(Level.SEVERE, null, ex);  
            }  
            return con;  
        } catch (ClassNotFoundException ex) {  
            Logger.getLogger(Connect.class.getName()).log(Level.SEVERE, null, ex);  
        }  
        return con;  
    }  
}
```

PROJECT SCREEN SHOTS

Database Snapshot

Doctor's Table

The screenshot shows the MySQL Query Browser interface. The query window contains the SQL statement: `SELECT * FROM doctorsinfo d;`. The results pane displays two rows of data from the 'doctorsinfo' table. The table has columns: id, name, pwd, loginStatus, path, type, and email. The first row shows a doctor with id 1001, name Manish Kumar, pwd 123@abc, loginStatus 0, path NULL, type Doctor, and email manish@gmail.com. The second row shows a doctor with id 1002, name Prithee Datta, pwd abc@123, loginStatus 0, path NULL, type Doctor, and email prithee@gmail.com. The status bar at the bottom indicates '2 rows fetched in 0.0043s (0.0002s)'.

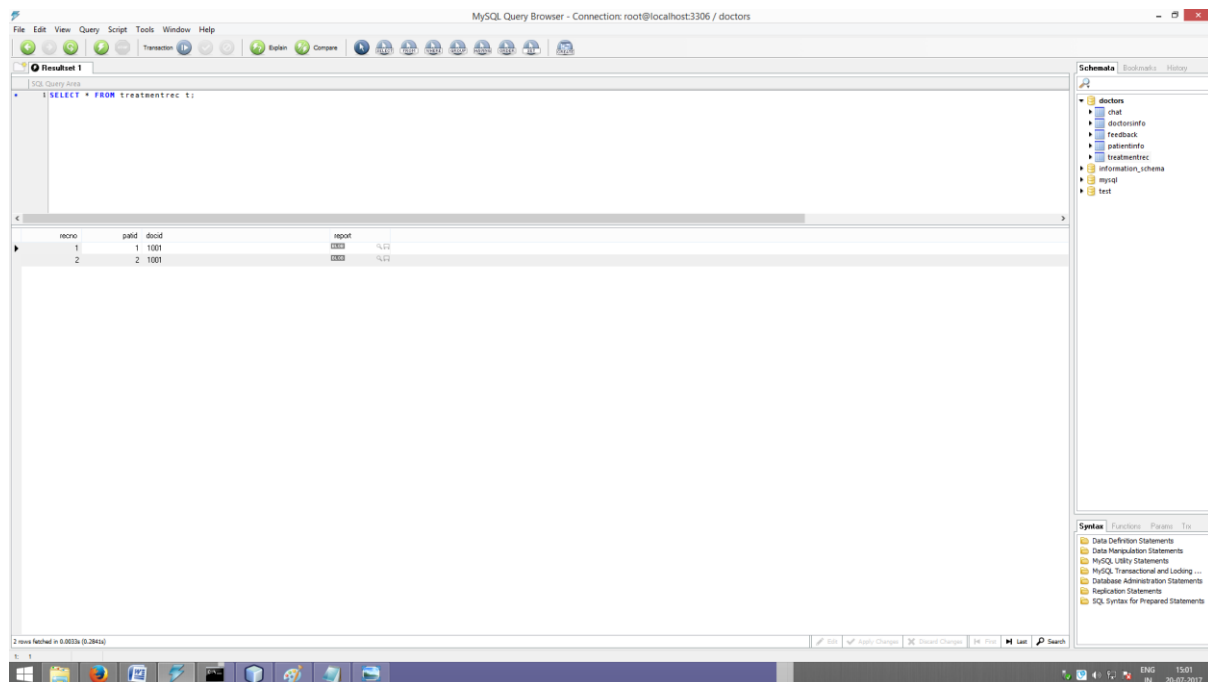
id	name	pwd	loginStatus	path	type	email
1001	Manish Kumar	123@abc	0	NULL	Doctor	manish@gmail.com
1002	Prithee Datta	abc@123	0	NULL	Doctor	prithee@gmail.com

Patient Table

The screenshot shows the MySQL Query Browser interface. The query window contains the SQL statement: `SELECT * FROM patientinfo p;`. The results pane displays two rows of data from the 'patientinfo' table. The table has columns: pafid, name, pwd, loginStatus, path, type, and email. The first row shows a patient with pafid 1, name Akshay Das, pwd 123@abc, loginStatus Active, path NULL, type Patients, and email akshay@gmail.com. The second row shows a patient with pafid 2, name Nishant Kumar, pwd 456@abc, loginStatus Active, path NULL, type Patients, and email nishant@gmail.com. The status bar at the bottom indicates '2 rows fetched in 0.0053s (0.0002s)'.

pafid	name	pwd	loginStatus	path	type	email
1	Akshay Das	123@abc	Active	NULL	Patients	akshay@gmail.com
2	Nishant Kumar	456@abc	Active	NULL	Patients	nishant@gmail.com

Report Table



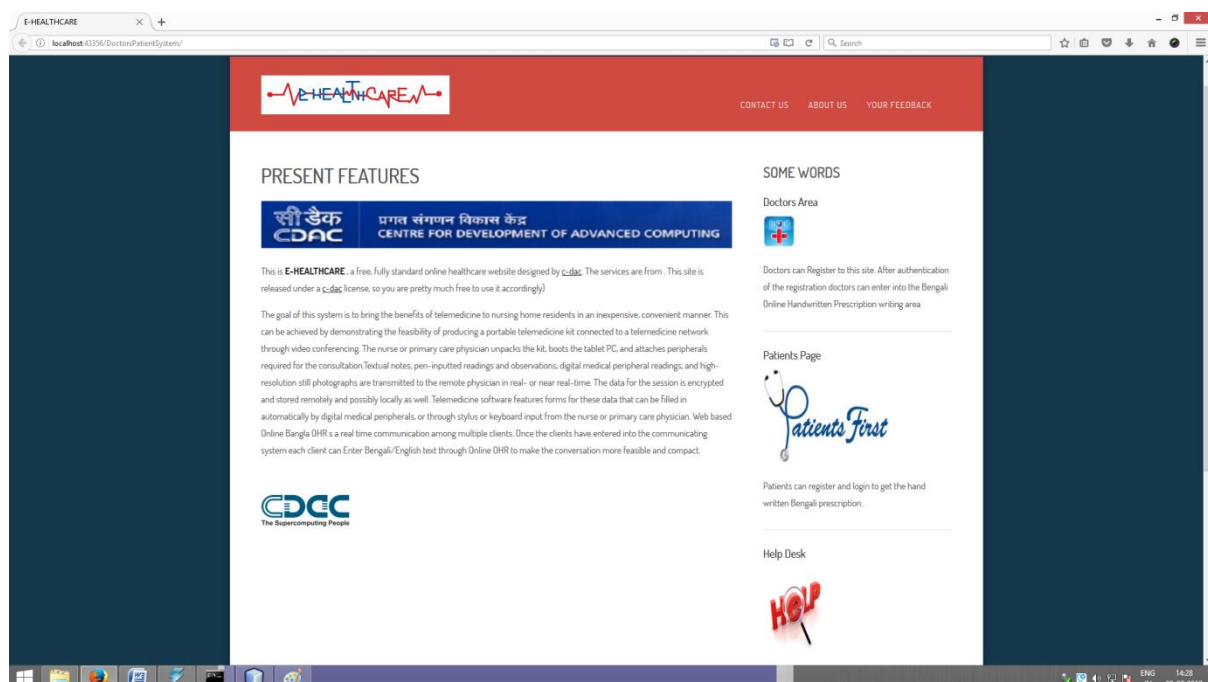
The screenshot shows the MySQL Query Browser interface. The main window displays the query result for the query `SELECT * FROM treatmentrec t;`. The result is a table with 4 columns: `idono`, `paid`, `docid`, and `report`. There are 2 rows of data.

idono	paid	docid	report
1	1	1001	1001
2	2	1001	1001

GUI Snapshot

Doctor's Section

Home Page (home.jsp)



Doctor's Registration Page (canvas.jsp)

E-HEALTHCARE

HOME PAGE YOUR FEEDBACK ABOUT US CONTACT US

PRESENT FEATURES

सी डैक CDAC प्रगत संगणन विकास केन्द्र
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

This is **E-HEALTHCARE**, a free, fully standard online healthcare website designed by **cdac**. The services are from. This site is released under a **cdac** license, so you are pretty much free to use it accordingly!

The goal of this system is to bring the benefits of telemedicine to nursing home residents in an inexpensive, convenient manner. This can be achieved by demonstrating the feasibility of producing a portable telemedicine kit connected to a telemedicine network through video conferencing. The nurse or primary care physician unpacks the kit, boots the tablet PC, and attaches peripherals required for the consultation. Textual notes, pen-inputted readings and observations, digital medical peripheral readings, and high-resolution still photographs are transmitted to the remote physician in real- or near real-time. The data for the session is encrypted and stored remotely and possibly locally as well. Telemedicine software features forms for these data that can be filled in automatically by digital medical peripherals, or through stylus or keyboard input from the nurse or primary care physician. Web based Online Bangla QHR is a real time communication among multiple clients. Once the clients have entered into the communicating system each client can Enter Bengali/English text through Online QHR to make the conversation more feasible and compact.

CDAC
The Supercomputing People

SOME WORDS

Home

DOCTORS/PATIENTS REGISTRATION FORM:-

User id-
Name
Password
Re type Ur Password
Email
Confirm Email
Address
City/Town/Village
State
Country
Zip/Postal Code
Phone
Occupation
Institution
Type

Signature Panel:

Doctor's WelcomePage (welcome.jsp)

E-HEALTHCARE

HOME PAGE YOUR FEEDBACK ABOUT US CONTACT US

PRESENT FEATURES

सी डैक CDAC प्रगत संगणन विकास केन्द्र
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING


This is **E-HEALTHCARE**, a free, fully standard online healthcare website designed by **cdac**. The services are from. This site is released under a **cdac** license, so you are pretty much free to use it accordingly!

The goal of this system is to bring the benefits of telemedicine to nursing home residents in an inexpensive, convenient manner. This can be achieved by demonstrating the feasibility of producing a portable telemedicine kit connected to a telemedicine network through video conferencing. The nurse or primary care physician unpacks the kit, boots the tablet PC, and attaches peripherals required for the consultation. Textual notes, pen-inputted readings and observations, digital medical peripheral readings, and high-resolution still photographs are transmitted to the remote physician in real- or near real-time. The data for the session is encrypted and stored remotely and possibly locally as well. Telemedicine software features forms for these data that can be filled in automatically by digital medical peripherals, or through stylus or keyboard input from the nurse or primary care physician. Web based Online Bangla QHR is a real time communication among multiple clients. Once the clients have entered into the communicating system each client can Enter Bengali/English text through Online QHR to make the conversation more feasible and compact.


CDAC
The Supercomputing People

SOME WORDS

1001
1002 Available
Check Patient's Availability:



Doctors can Register to this site. After authentication of the registration doctors can enter into the Bengali Online Handwritten Prescription writing area

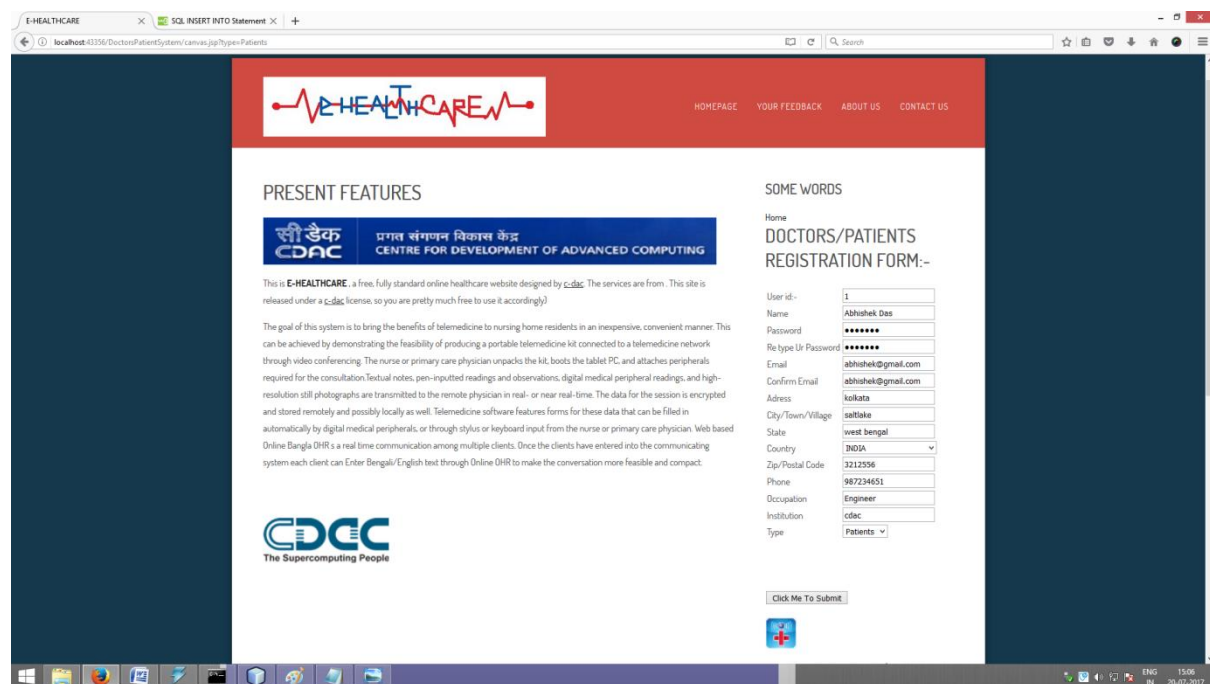
Help Desk


ChatGUI (welcome.jsp)

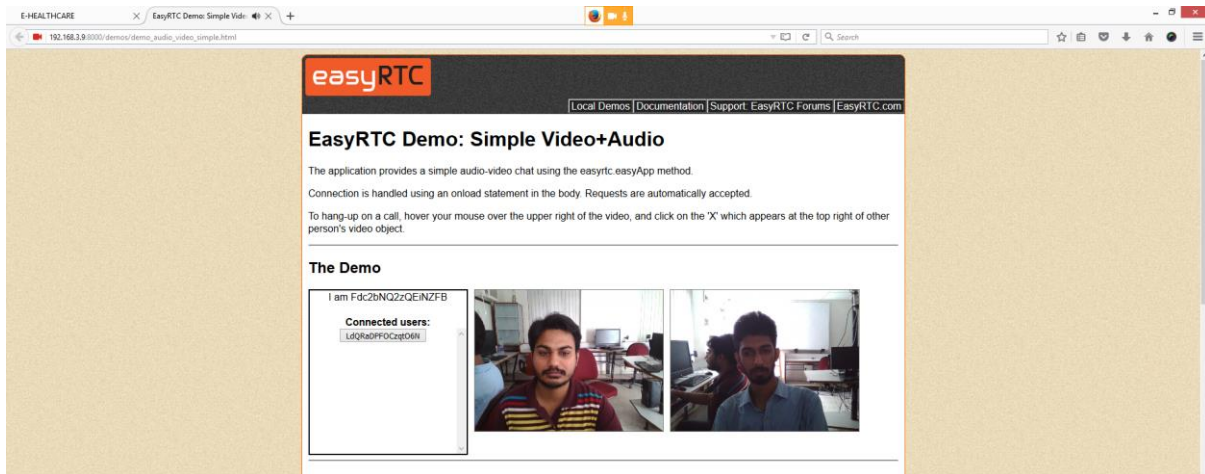


Patient Section

Patient Registration Page (canvas.jsp)



Audio Video Conferencing (demo_audio_video_simple.html)



CONCLUSION

The portal **E-Health Care System in Virtual Environment** reduces the considerable drawbacks like burden of human labour, long awaited appointments and hectic schedule of Doctors. This portal saves time and provides 24 hour accessibility even from a remote place. The portal offers a tremendous option to establish an interaction between doctor and patient through audio-video chat or text chat.

The implemented system provides a simple but effective telemedicine support for patients. The system will ensure a quality treatment for the people living in the rural areas or any remote location. It has been successfully implemented and tested in a similar environment with a few number of people. Due to it's user responsive and easy to understand GUI, the system is very much user friendly.

The system has to be tested with a large number of patients and doctors in order to test its scalability and performance. The acceptance of the system also needs to be verified.

SCOPE FOR FUTURE ENHANCEMENT

As we all know that no system is 100% perfectly developed, so there are a lot of areas where we need to work on to make the system more efficient. A separate module is also created for prescription upload which includes a tablet pc with hand writing recognition software. Using this tablet a doctor can write a prescription including his/her signature and the prescription will be converted to a .pdf file and then it will be uploaded to the database. In future we would like to add this module with the existing system.

In future we would like to make the system more responsive and also like to improve each module. For example after login the doctor will be able to see the number of patients in the queue. He can select the patients one by one and consult with them. On the patient's end the patient can see the number of doctors available on a particular day and according to his/her needs, he/she can choose the doctor etc.

These are few implementations we want to add in near future.

REFERENCES

1. <http://www.google.com>
2. <https://www.w3schools.com/>
3. <https://www.apachefriends.org/download.html>
4. <https://webrtc.org/>
5. Web based Bangla Online Handwritten Prescription system for Telemedicine through Multimedia Conferencing <http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-12-4128-4133.pdf>
6. Telemedicine: A New Horizon in Public Health in India <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2782224/>
7. Eysenbach G. What is E-Health?
<http://www.jmir.org/2001/2/e20/>
8. G Kaur, N Gupta, E-health: A new perspective on global health
https://www.researchgate.net/profile/Gurjit_Kaur4/publication/242247956_E-health_A_New_Perspective_on_Global_Health/links/5422b0230cf238c6ea6aaa00.pdf
9. Telemedicine System Interoperability Architecture
<https://www.itu.int/itudoc/itu-t/workshop/e-health/wcon/s8con001.pdf>
10. Edgar Pek and Sven Loncaric, Internet-based Medical Teleconsultation System
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=938709>
