CHAPTER 1

INTRODUCTION

MEDEX MEDICAL CENTER is a website to automate & manage patient appointments booking.

The existing system consist of booking patient appointment manually and storing patient data in paper files. To reduce paper work and time we come up with a web application. The website consists of 3 modules such as patient, doctor and admin. In patient module the patient can book their appointment and cancel the appointment. In doctor module doctor can view their appointment and view the patient details. In admin module admin can generate report such as no of patient visited the hospital and no of doctor working in the hospital etc.

The following modules, clearly explains how **WEB APPLICATION FOR MEDEX MEDICAL CENTER** developed to meet the various requirements.

1.1 EXISTING SYSTEM

In existing system, the patient wants to book their appointment manually. The records are stored in paper files. It is very difficult to search the old records. To overcome this difficulty, we will automate the patient booking and storing their data.

1.2 OBJECTIVE

Medex Medical center is a process of implementing all the activities of the hospital in a computerized automated way to fasten the performance. The main purpose of our system is to make hospital task easy and is to develop software that replaces the manual hospital system into automated hospital management system.

1.4 SCOPE

The proposed system consists of three major modules i.e.

- Patient
- Doctor
- Admin

1. The Scope of **Patient module** is to

    - **Login:** Patient login
    - **Register:** New Patient registration
    - **Patient**: This will capture information about Patient register as well as Patient details
    - **Book a Doctor's Appointment:** Patient can book an appointment with the Doctor for consultation along with date and time preferred, and also we have option to select the specialty to consult

2. The Scope of **Doctor module** is to

    - **Check Appointments:** Doctor can check the appointment scheduled
    - **Add Diagnosis Info:** Doctor can add diagnosis details in form.

3. The Scope of the **Medex Admin module** is to

    - **Reports:** The Admin will have the options to generate the vise kind of reports giving summary details of patients & doctors
    - **Registration:** The Admin can register the doctor details in database.

CHAPTER 2

GENERAL DESCRIPTION

2.1 PROJECT PERSPECTIVE

The main perspective of the project is to reduce the paper work, data retrieval through automation. Our Project will replace all traditional and outdated means of tracking patient information and other data useful to the hospital. This system shall replace forms of databases using manual or outdated hardcopy databases. Accessing data can be better monitored, organized, and time conscientious.

2.2 USER CHARACTERISTICS

In this application, the admin can register the doctor details, the patient can register and login then they can book their appointment, view their appointment and cancel their appointment. The doctor can login and check their appointments, view patient details.

2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

The application is designed for Medex Medical Center to make the data entry and manipulation process simple, this application is portable and then the relation should satisfy the key constraints, domains, referential constraints and integrity constraints.

2.4 PROPOSED SYSTEM

The proposed system is a web application, which is completely related to automation. It consists of 3 modules such as Patient, Doctor and Admin module. The patient can register and login into the system, then they can book their appointment. The doctor can view their appointment and view the details about the patient. The admin only has the rights to register doctor details in database and they can generate report such as no of patient visited to the hospital and no of doctors who are working on the hospital.

2.5 ADVANTAGES

- Saves users time.
- portable
- Security - User id is provided with password or access only to the kind information.
- Data integrity

CHAPTER 3

REQUIREMENTS SPECIFICATIONS

3.1 FUNCTIONAL REQUIREMENTS

When the users enter the link http://localhost:8070/MedexMedical in the browser, Home page appears which will have a short welcome message; a brief introduction to the purpose of the C-Medex, the user can choose appropriate options. Medex Medical Center super-specialty services where specialist care is effectively administered under one roof. At Medex Medical Center Hospital, there is a continuous drive for quality in order to meet patients' standards through quality healthcare delivery.

3.1.1 HOME PAGE INTERFACE REQUIREMENTS

1. Home
2. Patient
3. Doctor
4. Medex Admin
5. About Us
6. Contact Us

3.1.2 MODULE DESCRIPTION

3.1.2.1 PATIENT

3.1.2.1.2 PATIENT - REGISTRATION PAGE

- When the patient/user clicked on the registration for free link, it should re-direct to registration for free form and the users/ patient need to fill some of the basic attributes/fields as mentioned below in requirement.
- Login Id, Password, First Name, Last Name, Age, Gender (Male or Female), Contact Number, E-mail, Address, Zip code, City.

3.1.2.1.2 PATIENT - LOGIN PAGE

When the users enter link http://localhost:8070/MedexMedical in the browser, the patient tab will have provided the features like, Login id and Register for free. Below are the fields:

- Login id
- Password

3.1.2.1.3 PATIENT - LOGGED IN PAGE

When the patient logged in, they should able to view the following options in the page. They are:

- Book Appointment
- View Appointment
- Cancel Appointment

3.1.2.2 DOCTOR

- In the doctor logged in page we have to give the upcoming weekend seminars details.
- The doctor can also check the appointment by choose the date pattern and should able to view patients details like, patients name, age, gender, address, appointment booked date etc.
- The doctor can update the diagnosis detail of the appropriate patient in the application.
- The doctor can also have a feature to view all the patient details by selecting the name.

3.1.2.3 ADMIN

- Admin only register the doctor and give their login id.
- The Medex admin should have a feature to view and to generate reports, like No of doctor in the hospital, No of patient visited to hospital and o their varies kind of reports should be able to generated by Medex Admin.

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 PERFORMANCE REQUIREMENTS

- All front-end pages and reports should be served up in less than 5 seconds post click when up to 1000 users are on the application concurrently

3.2.2 OPERATIONAL REQUIREMENTS

- Application Java and J2EE for deployment with Mysql database for persistence .
- This system must be up 24X7.

3.2.3 MAINTAINABILITY & PORTABILITY REQUIREMENTS

- If the allocated memory space for the existing website needs to be increased, then it should be possible without any impact to performance.

3.2.4 SECURITY REQUIREEMENTS

- No unauthorized users should be able to log on to the system
- User Id/Login Id should have combination of letter and numeric values, none

3.3 USER INTERFACE

3.3.1 USER INTERFACES

- Front-end software: Browser
- Scripting Languages: Front-end: JSP, HTML, Bootstrap
- Technologies: Servlet
- Back-end software: MySQL

3.3.2 HARDWARE INTERFACES

•	Processor	: Intel(R) Core(TM)i5
•	RAM	: 4 GB
•	Memory	: 1 TB

3.3.3 SOFTWARE INTERFACES

Following are the software used for the student information management online application.

TABLE 3.1 SOFTWARE DESCRIPTION

SOFTWARE USED	DESCRIPTION
Operating system	Windows operating system is chosen for its best support and user-friendliness.
Server	Application needs Apache Tomcat server for deployment with MySQL 5.5 database for persistence
Database	SQLs database is chosen to save the customer records.

3.3.4 SOFTWARE AND PROGRAMS USED

A number of software and programs have been used in order to develop the sample online booking system. The types of programs and software used were programming languages such as Java, frameworks such as Servlets, development platform such as Eclipse, script language such as JavaScript and bootstrap, markup language such as HTML and database program such as SQLite, among others. Proper installation of all the programs used were required and basic understanding of the programs were utilized in the completion of the project.

3.3.4.1 JAVA

Java is a set of technologies (programming language and computing platform) for creating and running software. Java is a popular general-purpose programming language and computing platform. It is fast, reliable, and secure. According to Oracle, the company that owns Java, Java runs on 3 billion devices worldwide. Java was built with the philosophy of "write once, run anywhere" (WORA). The Java code (pure Java code and libraries) you write on one platform (operating system) will run on other platforms with no modification. To run Java, an abstract machine called Java Virtual Machine (JVM) is used. The JVM executes the Java bytecode. Then, the CPU executes the JVM. Since all JVMs works exactly the same, the same code works on other operating systems as well, making Java platform-independent. There are different styles of programming. Object-oriented approach is one of the popular programming styles. In object-oriented programming, a complex problem is divided into smaller sets by creating objects. This makes your code reusable, has design benefits, and makes code easier to maintain.The earlier versions of Java were criticized for being slow. However, things are completely different now. The new JVMs are significantly faster. And, the CPU that executes JVM are also getting more and more powerful. The Java platform provides various features for security of Java applications. Some of the high-level features that Java handles are:
-provides secure platform for developing and running applications
-automatic memory management, reduces memory corruption and vulnerabilities
- provides secure communication by protecting the integrity and privacy of data transmitted

3.3.4.2 SERVLETS

Today we all are aware of the need of creating dynamic web pages i.e the ones which have the capability to change the site contents according to the time or are able to generate the contents according to the request received by the client. If you like coding in Java, then you will be happy to know that using Java there also exists a way to generate dynamic web pages and that way is Java Servlet. But before we move forward with our topic let’s first understand the need for server-side extensions.

Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server.Servlets work on the server-side. Servlets capable of handling complex request obtained from web server.

1. The clients send the request to the web server.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generate the response in the form of output.
5. The servlet sends the response back to the web server.
6. The web server sends the response back to the client and the client browser displays it on the screen.

3.3.4.3 HYPERTEXT MARKUP LANGUAGE

Hyper Text Markup Language (HTML) is a popular markup language used in web pages. HTML can be simply written in a text editor and tested through web browser. Writing in html is easy; with html it's also possible to add media and images to the web page. HTML contains special markup tags like <title>, <h>, <p> etc. To declare the title of the page for example, the title has to be included in the title tags. Similarly, the paragraphs, headings and other different contents in website pages have to be included inside the respective HTML tags. It is easy to store HTML code; in a simple text file with filename followed by .html or .htm extension. HTML gives developers the possibility of creating sections in the document. As <title> tag gives the title for the web page, <H1> for example specifies the main content of the web page. Similarly, with H2, H3 and so on tags, HTML creates minor contents. There are tags for other features such as paragraphs (<p>), style of font (<b>bold</b>) and tables (<table>) etc.

3.3.4.4 JAVASCRIPT

JavaScript is one of the most popular script languages for webpages today. JavaScript (JS) is an object-oriented language that supports features such as imperative and functional programming. The syntax is similar to other object-oriented languages such as Java and C++, hence making it easy to learn for people who know these languages. JS is used to add interactive features such as buttons, animation, games etc. JS was invented by Brendon Eich. JavaScript is capable of creating many different features ranging from beginner to advanced such as 2D and 3D features on a website.

3.3.4.5 BOOTSTRAP

Build responsive, mobile-first projects on the web with the world's most popular front-end component library. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery. Bootstrap employs a handful of important global styles and settings that you'll need to be aware of when using it, all of which are almost exclusively geared towards the normalization of cross browser styles.

3.3.4.6 CASCADING STYLE SHEET

CSS is a language that was developed in 1992-1993. CSS helps design each and every element of the mark-up elements language such as HTML by giving complete control to the designer. While the HTML elements enables the web page designers to add what content they want, CSS makes it possible how to display the content to the user. CSS covers the areas such as colours, layout, advanced positions of elements, fonts and also allows the content to adapt the content to different devices such as phones, tablets, bigger screens and printers. CSS can operate independently as well as be used with any mark-up languages based in XML. CSS uses simple, everyday English words and has an easy syntax. CSS is crucial in advanced web designing as it gives control to the layout and offers numerous techniques to make the web page look sophisticated. Currently, the basic features of CSS are supported by all main browsers such as Internet Explorer, Safari, 10Opera, Chrome and Firefox. CSS has been used in the web pages for the development of sample online booking system for this thesis for positioning, layout, margins and colours for the HTML elements.

3.3.4.7 SQLITE DATABASE

Databases are collections of similar data. Databases are used for organized collection and storing of similar data, to be later used for specific purposes. A database contains tables with rows and columns populated with objects, which displays connection between them. A database acts as a shared resource for the programs which can use the information from the database. Many enterprises rely on databases today to store a wide range of information systematically. Databases are used almost everywhere: in small companies which can use database to save customers' information and also for more advanced scientific and military areas. Databases facilitate the task of searching hundreds and thousands of records much simpler by storing them in an organized manner. Among many database programs available today, SQLite Database has been used in the development work.

CHAPTER 4

DETAILED DESIGN

4.1 ARCHITECTURAL DESIGN

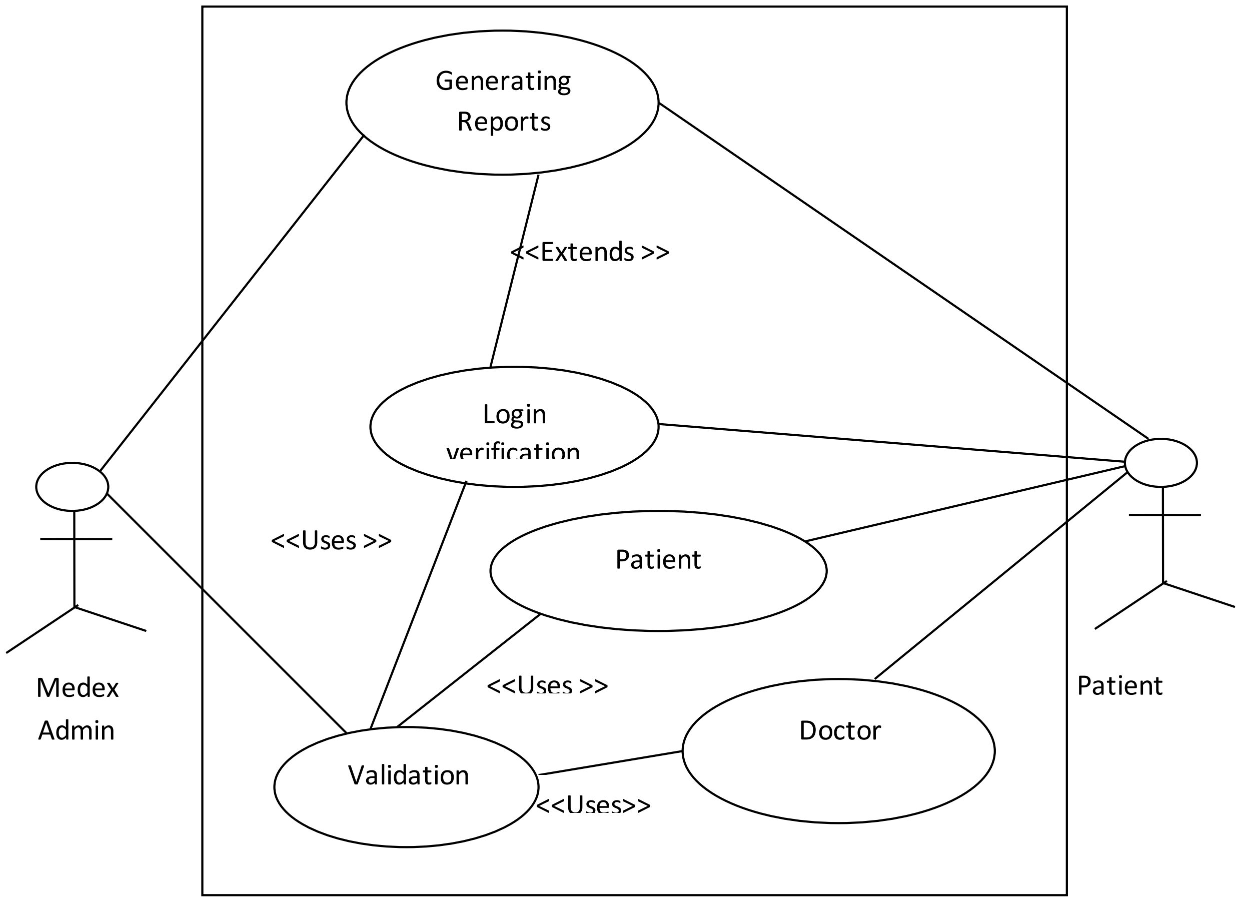


Figure 4.1 Use Case Diagram

4.2 DATA FLOW DIAGRAM

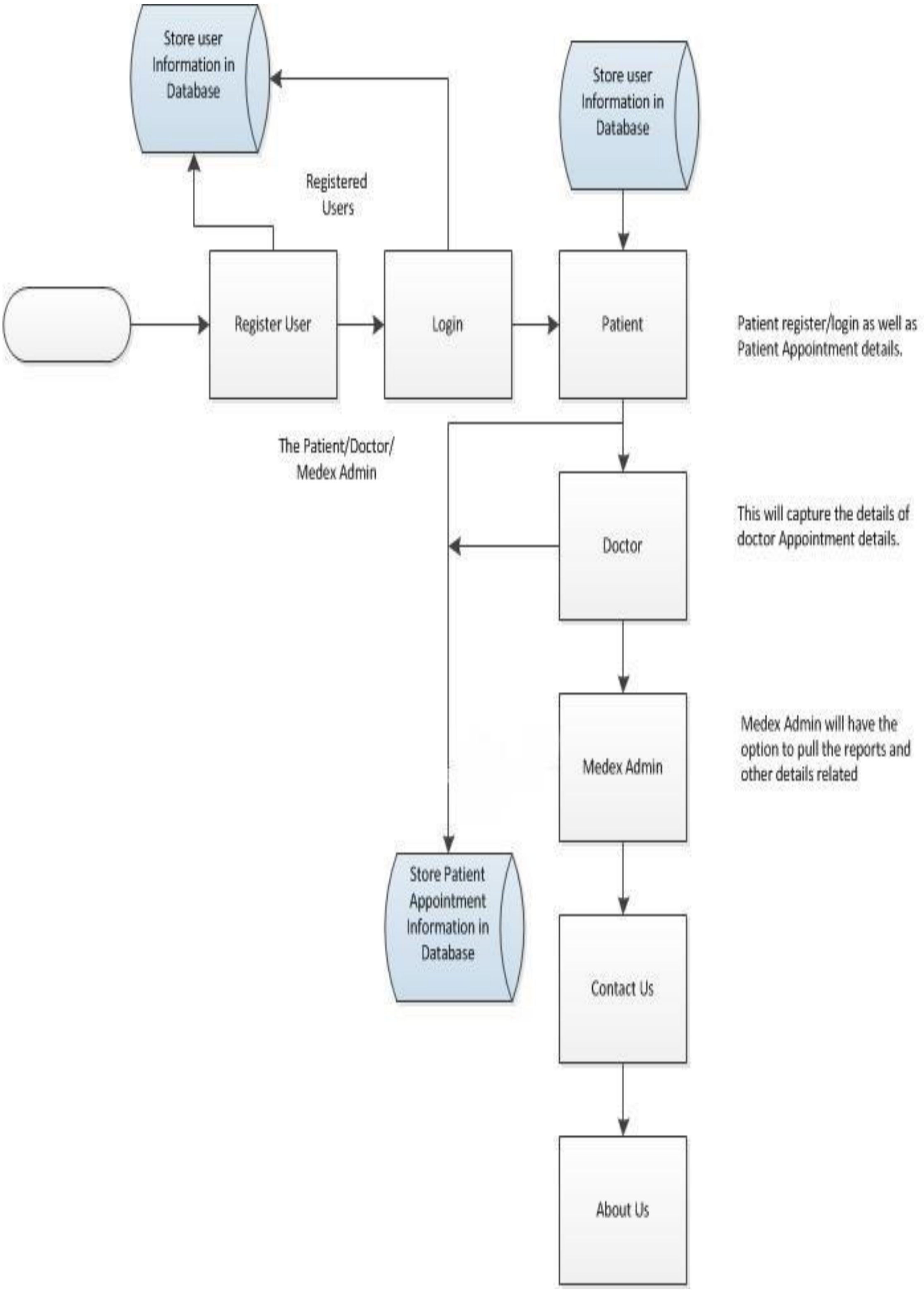

Store user Information in Database
Store user Information in Database
Registered Users
Register User
Login
Patient
Patient register/login as well as Patient Appointment details.
The Patient/Doctor/ Medex Admin
Doctor
This will capture the details of doctor Appointment details.
Medex Admin
Medex Admin will have the option to pull the reports and other details related
Store Patient Appointment Information in Database
Contact Us
About Us

Figure 4.2 Data Flow Diagram

4.3 DATABASE DESIGN

This project follows relational database model. The relational model represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship. The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized. Some popular Relational Database management systems are:

- DB2 and Informix Dynamic Server - IBM
- Oracle and RDB – Oracle
- SQL Server and Access - Microsoft

In our project the database consists of 5 tables such as patient, doctor, user, appointment and feedback. The patient acts as the major table in our database.

4.3.1 ER DIAGRAM

An ER diagram is a diagram that helps to design database in efficient way. Attributes in ER diagrams are usually modelled as an oval with the name of the attribute, linked to entity or relationship that contains the attribute.

Within the relation model the final step can generally be broken down into two further steps, that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects.

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other. ER diagrams are most often associated with complex databases that are used in software engineering and IT networks. In particular, ER diagrams are frequently used during the design stage of a development process in order to identify different system elements and their relationships with each other. For example, inventory software used in a retail shop will have a database that monitors elements such as purchases, item, item type, item source and item price.

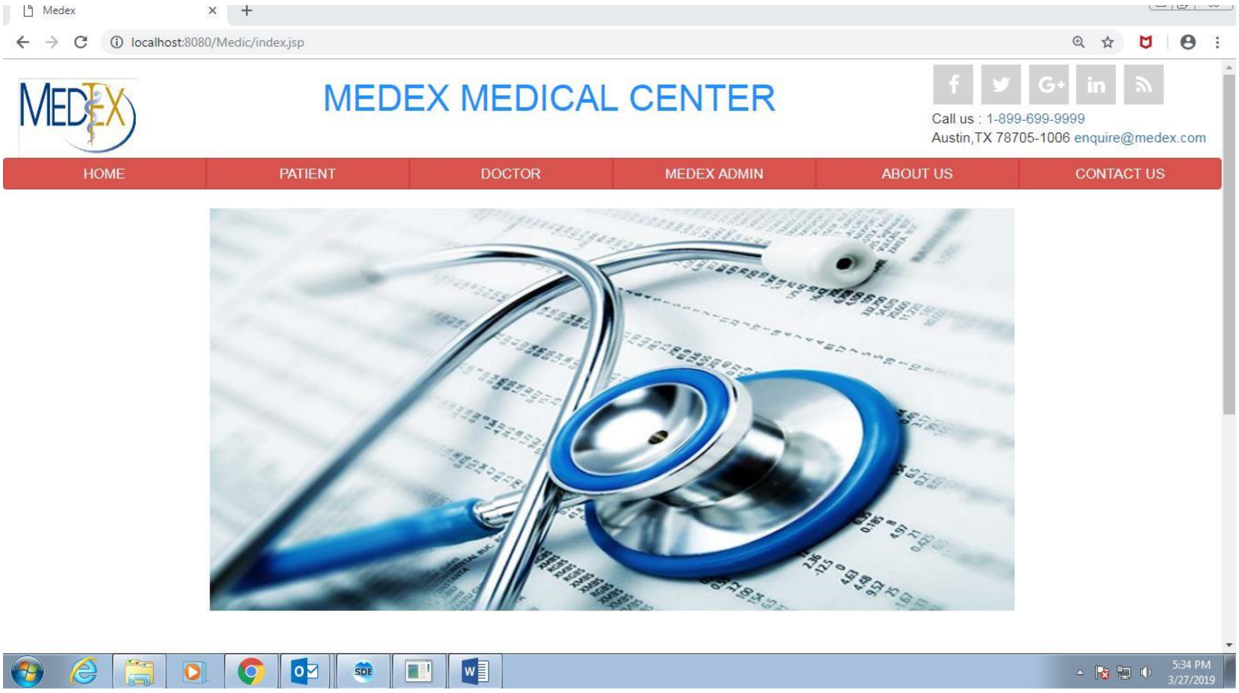
4.3.2 DATABASE MANIPULATION

Data Manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases. The functional capability of DML is organized in manipulation commands like SELECT, UPDATE, INSERT INTO and DELETE FROM, as described below:

- SELECT: This command is used to retrieve rows from a table. The syntax is SELECT [column name(s)] from [table name] where [conditions]. SELECT is the most widely used DML command in SQL.
- UPDATE: This command modifies data of one or more records. An update command syntax is UPDATE [table name] SET [column name = value] where [condition].
- INSERT: This command adds one or more records to a database table. The insert command syntax is INSERT INTO [table name] [column(s)] VALUES [value(s)].
- DELETE: This command removes one or more records from a table according to specified conditions. Delete command syntax is DELETE FROM [table name] where [condition].

4.4 OUTPUT DESIGN

HOME PAGE



Medex

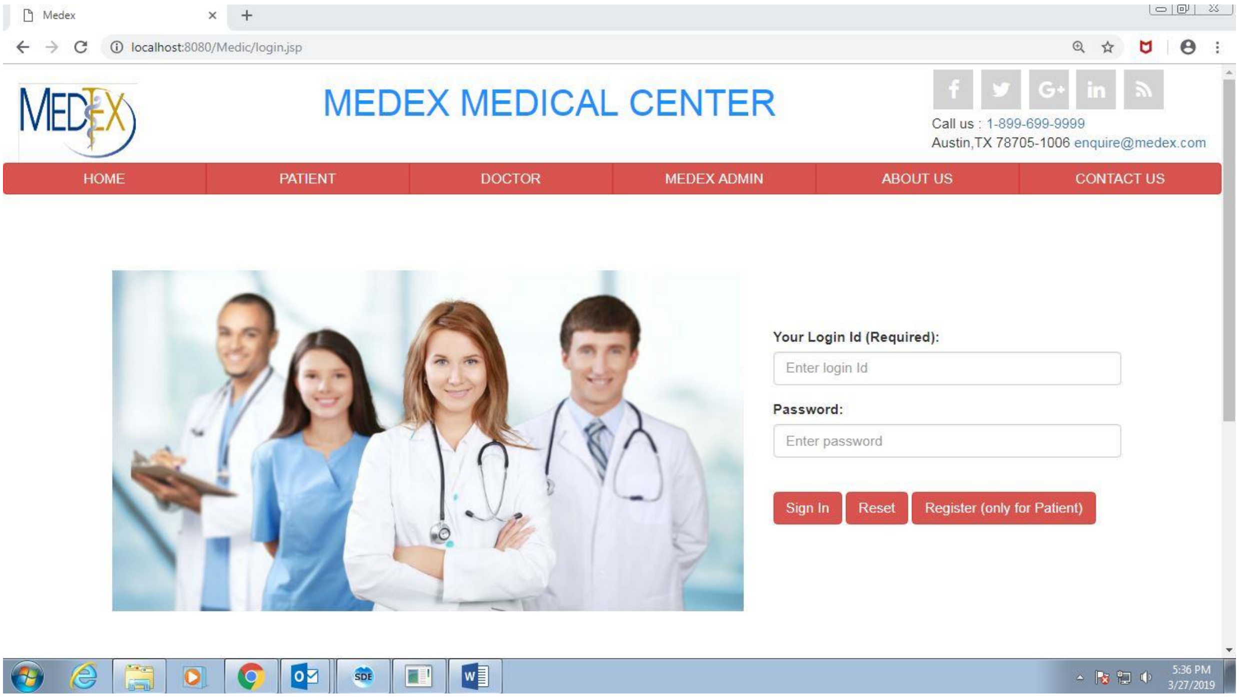
localhost:8080/Medic/index.jsp

MEDEX MEDICAL CENTER

Call us : 1-899-699-9999

Austin,TX 78705-1006 enquire@medex.com

LOGIN PAGE



Medex

localhost:8080/Medic/login.jsp

MEDEX MEDICAL CENTER

Call us : 1-899-699-9999

Austin,TX 78705-1006 enquire@medex.com

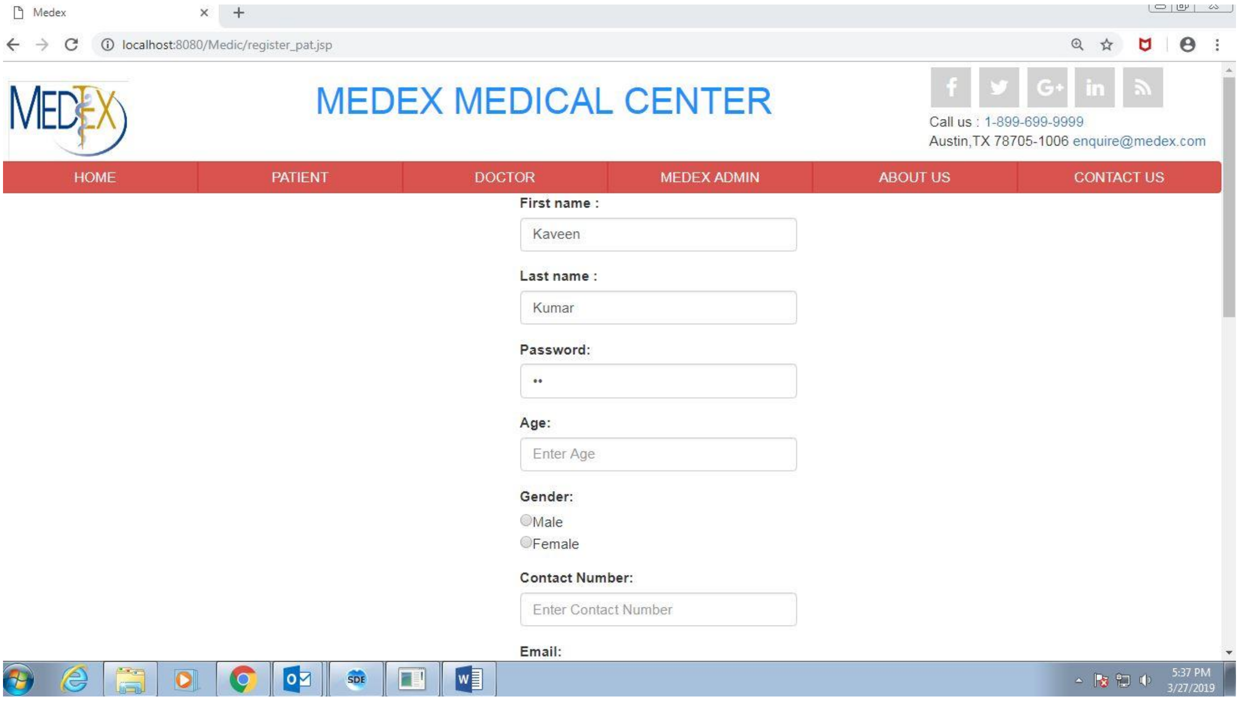
Your Login Id (Required):

Enter login Id

Password:

Enter password

REGISTER PAGE



MEDEX MEDICAL CENTER

Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

First name :
Kaveen

Last name :
Kumar

Password:
••

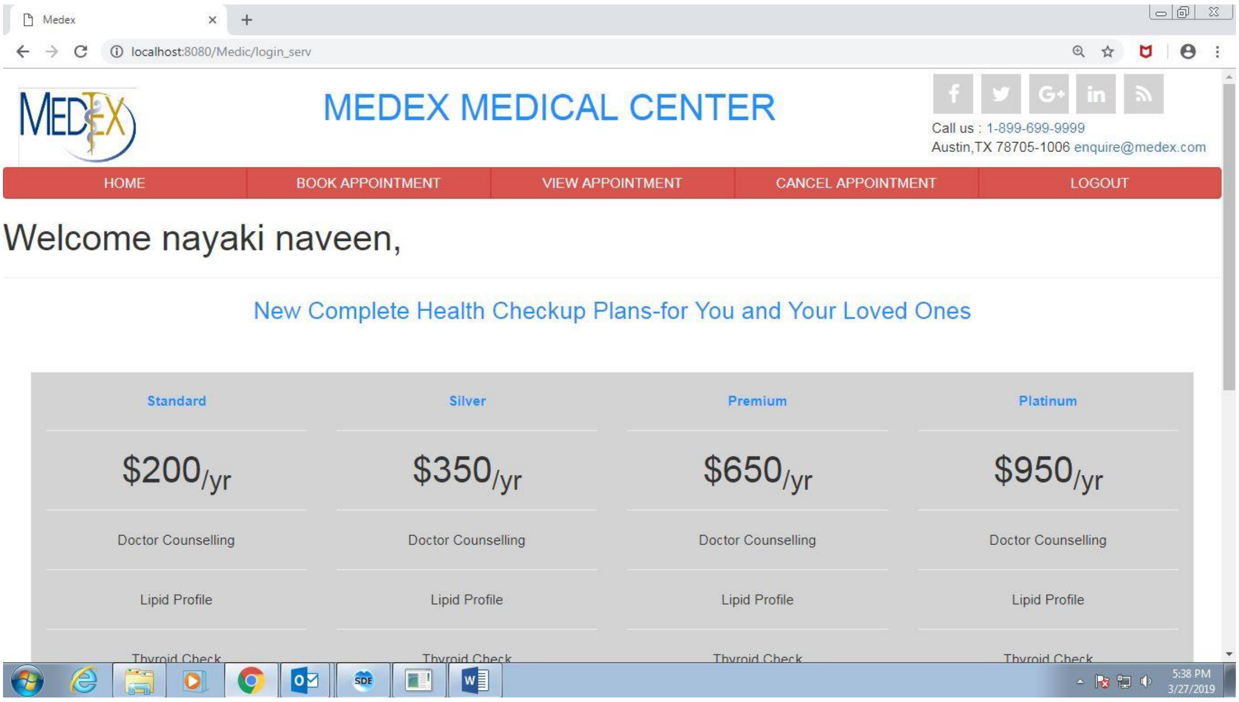
Age:
Enter Age

Gender:
○Male
○Female

Contact Number:
Enter Contact Number

Email:

PATIENT HOME PAGE



MEDEX MEDICAL CENTER

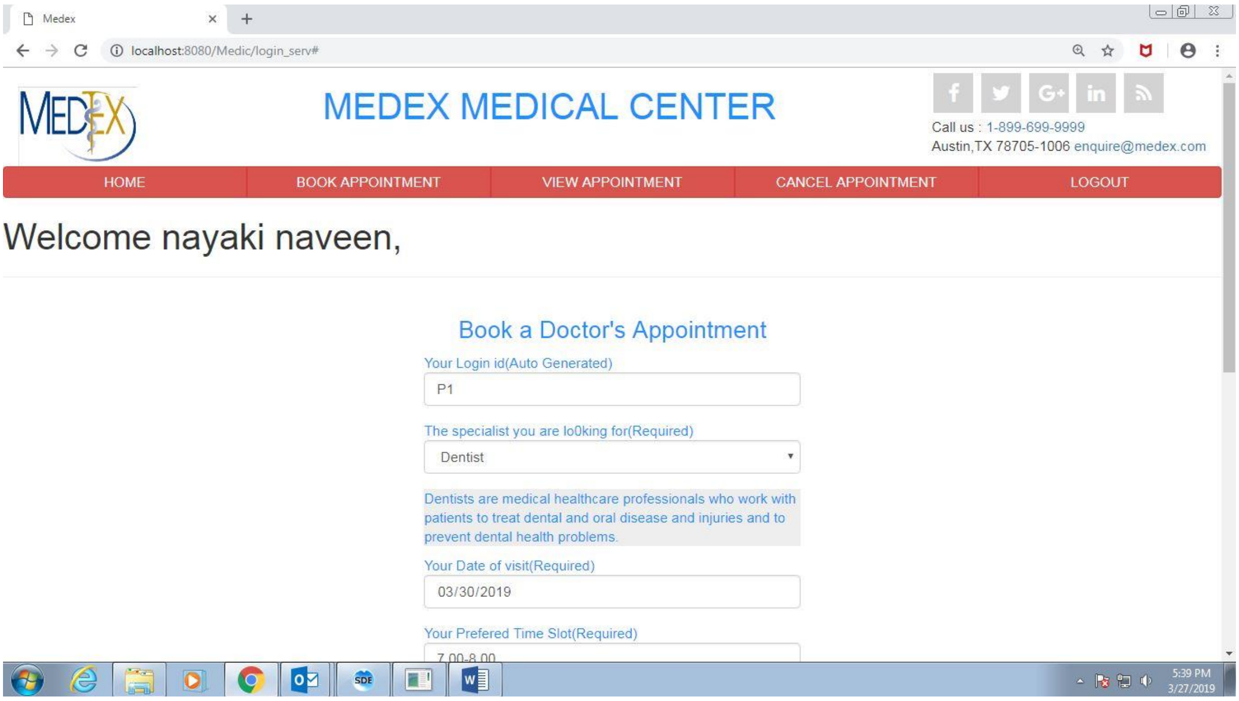
Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

Welcome nayaki naveen,

New Complete Health Checkup Plans-for You and Your Loved Ones

Standard	Silver	Premium	Platinum
$200/yr	$350/yr	$650/yr	$950/yr
Doctor Counselling	Doctor Counselling	Doctor Counselling	Doctor Counselling
Lipid Profile	Lipid Profile	Lipid Profile	Lipid Profile
Thyroid Check	Thyroid Check	Thyroid Check	Thyroid Check

APPOINTMENT BOOKING PAGE



MEDEX MEDICAL CENTER

Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

Welcome nayaki naveen,

Book a Doctor's Appointment

Your Login id(Auto Generated)

P1

The specialist you are lo0king for(Required)

Dentist

Dentists are medical healthcare professionals who work with patients to treat dental and oral disease and injuries and to prevent dental health problems.

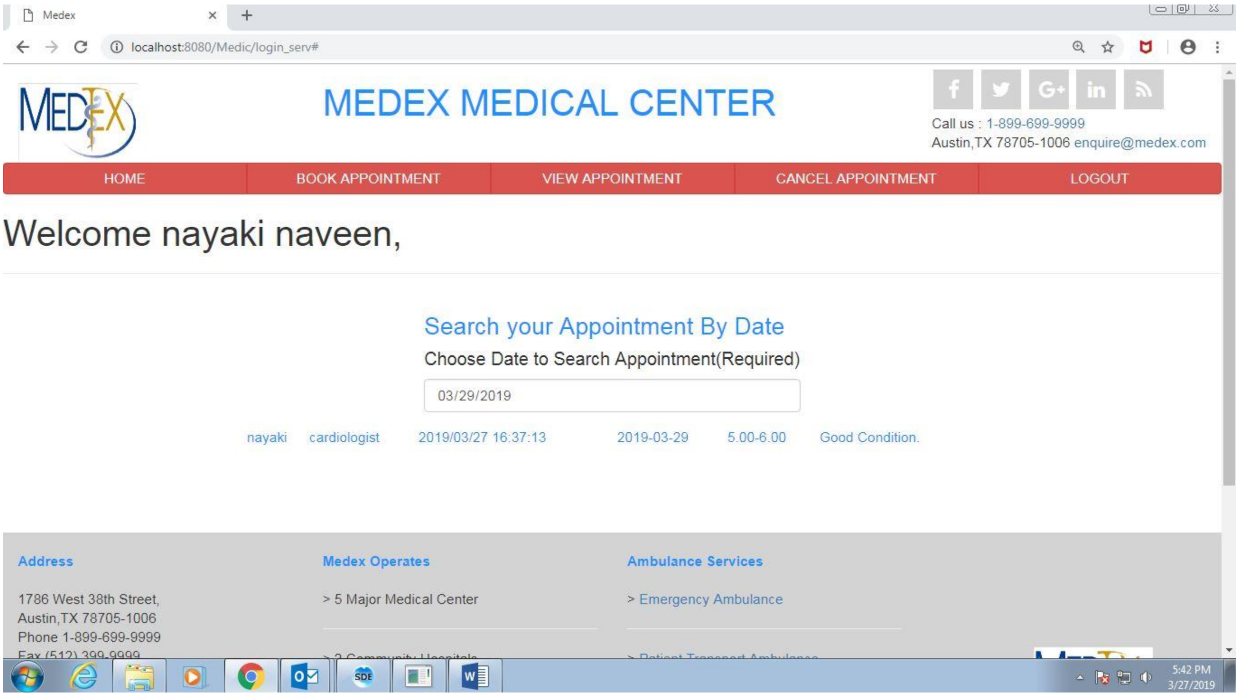
Your Date of visit(Required)

03/30/2019

Your Prefered Time Slot(Required)

7.00-8.00

VIEW APPOINTMENT PAGE



MEDEX MEDICAL CENTER

Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

Welcome nayaki naveen,

Search your Appointment By Date

Choose Date to Search Appointment(Required)

03/29/2019

nayaki	cardiologist	2019/03/27 16:37:13	2019-03-29	5.00-6.00	Good Condition.

Address

1786 West 38th Street,
Austin,TX 78705-1006
Phone 1-899-699-9999
Fax (512) 399-9999

Medex Operates

> 5 Major Medical Center

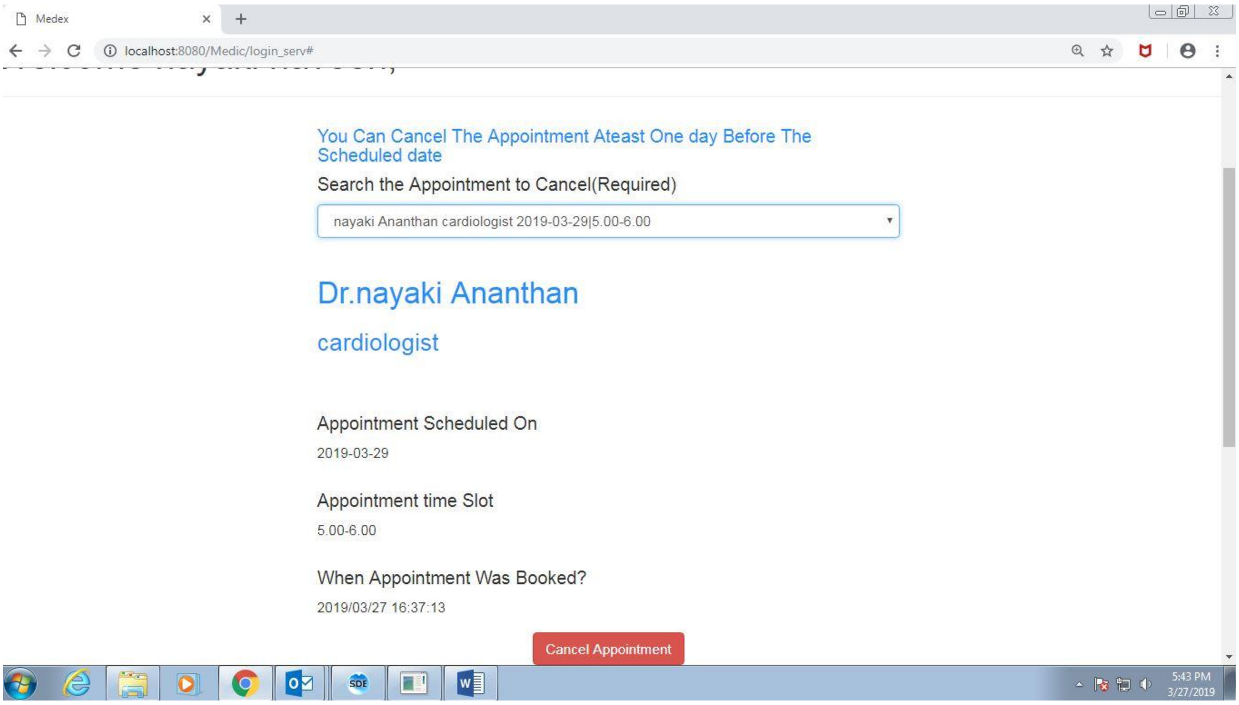
> 2 Community Hospitals

Ambulance Services

> Emergency Ambulance

> Patient Transport Ambulance

CANCEL APPOINTMENT PAGE



You Can Cancel The Appointment Ateast One day Before The Scheduled date

Search the Appointment to Cancel(Required)

nayaki Ananthan cardiologist 2019-03-29|5.00-6.00

Dr.nayaki Ananthan

cardiologist

Appointment Scheduled On

2019-03-29

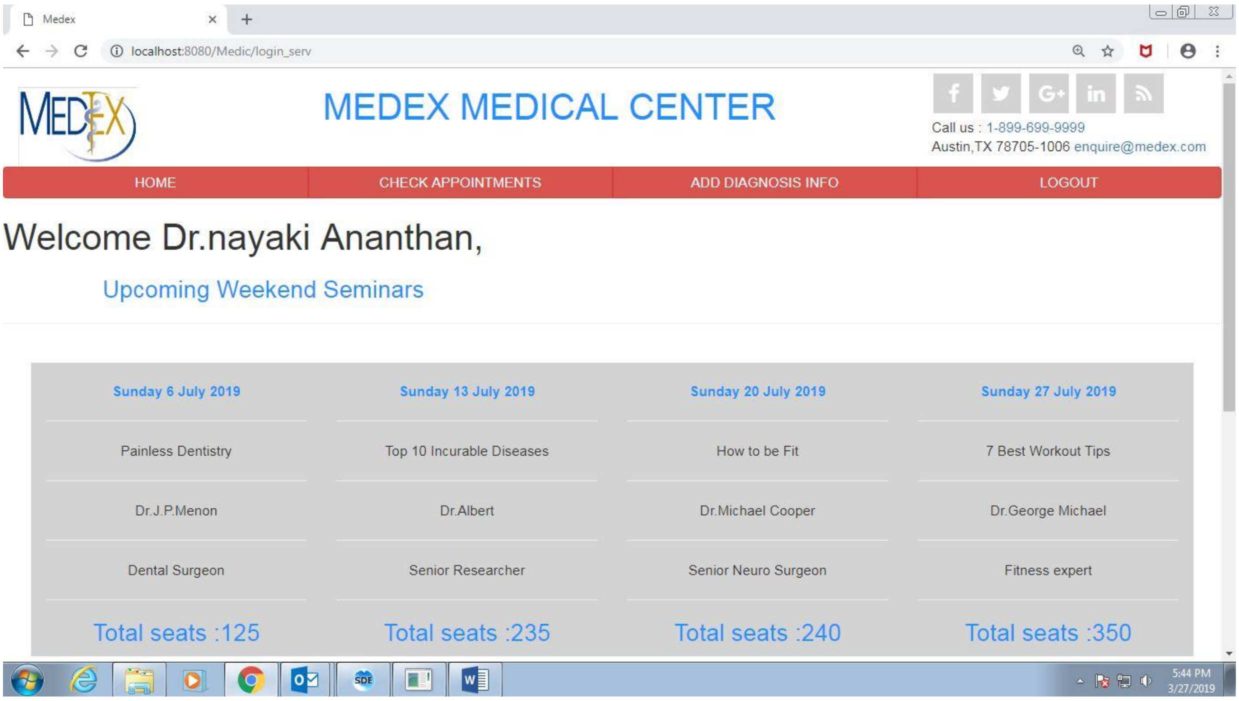
Appointment time Slot

5.00-6.00

When Appointment Was Booked?

2019/03/27 16:37:13

DOCTOR HOME PAGE



MEDEX MEDICAL CENTER

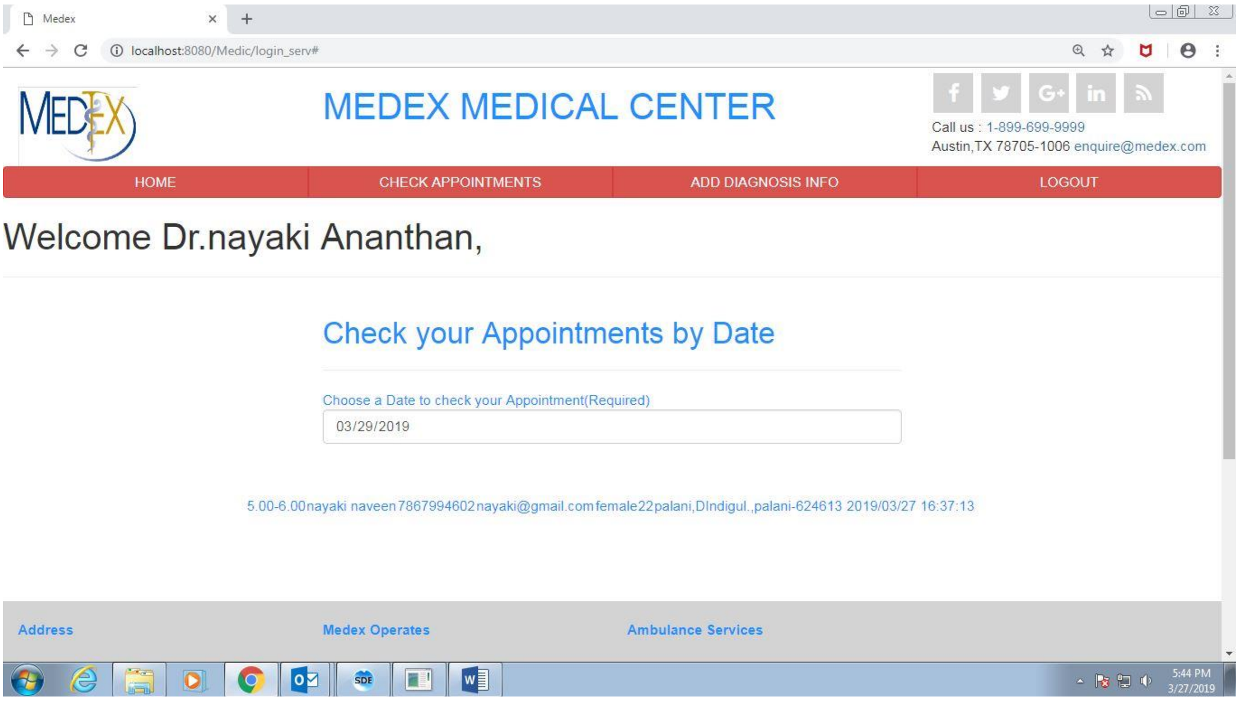
Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

Welcome Dr.nayaki Ananthan,

Upcoming Weekend Seminars

Sunday 6 July 2019	Sunday 13 July 2019	Sunday 20 July 2019	Sunday 27 July 2019
Painless Dentistry	Top 10 Incurable Diseases	How to be Fit	7 Best Workout Tips
Dr.J.P.Menon	Dr.Albert	Dr.Michael Cooper	Dr.George Michael
Dental Surgeon	Senior Researcher	Senior Neuro Surgeon	Fitness expert
Total seats :125	Total seats :235	Total seats :240	Total seats :350

CHECK APPOINTMENT PAGE



MEDEX MEDICAL CENTER

Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

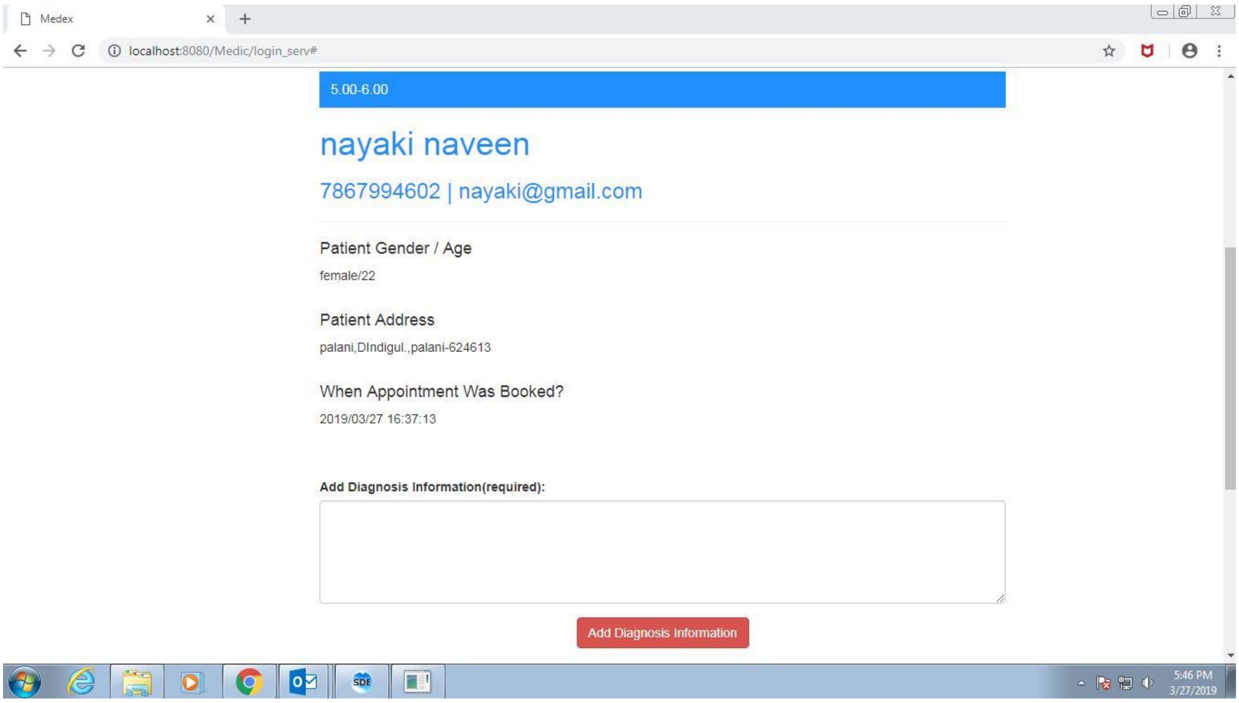
Welcome Dr.nayaki Ananthan,

Check your Appointments by Date

Choose a Date to check your Appointment(Required)
03/29/2019

5.00-6.00nayaki naveen7867994602nayaki@gmail.comfemale22palani,DIndigul.,palani-624613 2019/03/27 16:37:13

ADD DIAGNOSIS INFO PAGE



5.00-6.00

nayaki naveen

7867994602 | nayaki@gmail.com

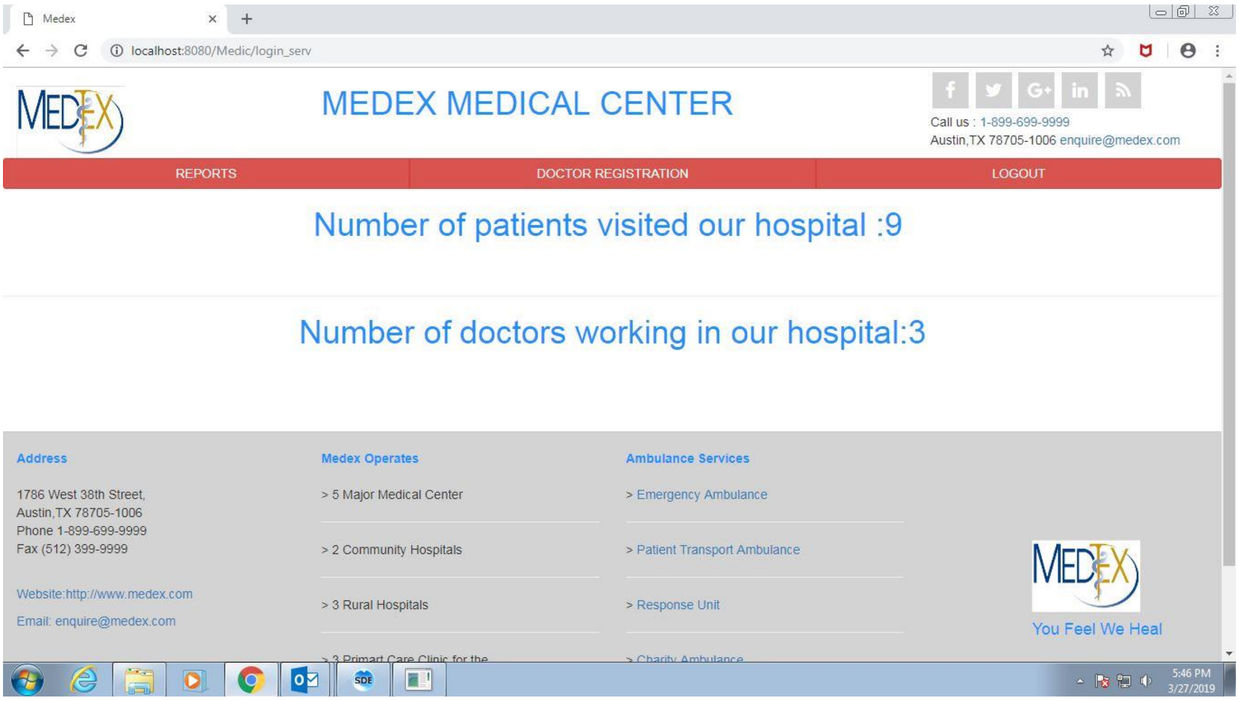
Patient Gender / Age
female/22

Patient Address
palani,DIndigul.,palani-624613

When Appointment Was Booked?
2019/03/27 16:37:13

Add Diagnosis Information(required):

ADMIN HOME PAGE



MEDEX MEDICAL CENTER

Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

Number of patients visited our hospital :9

Number of doctors working in our hospital:3

Address

1786 West 38th Street,
Austin,TX 78705-1006
Phone 1-899-699-9999
Fax (512) 399-9999

Website:http://www.medex.com
Email: enquire@medex.com

Medex Operates

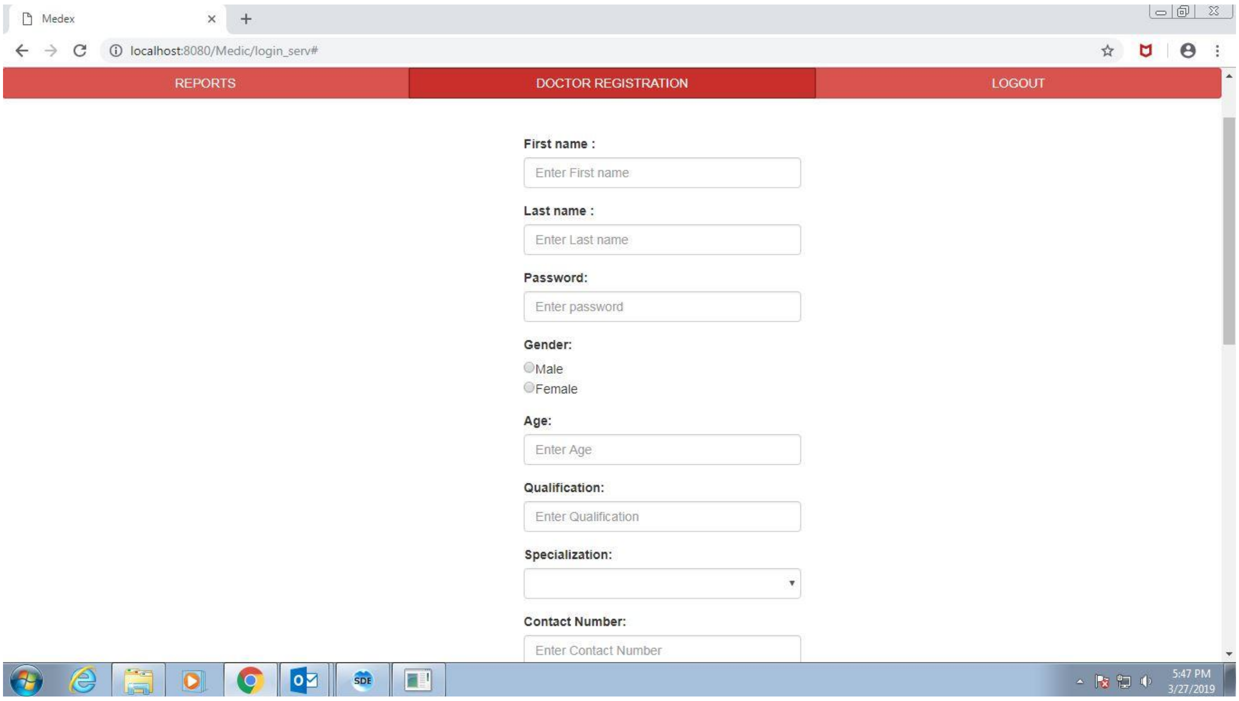
> 5 Major Medical Center
> 2 Community Hospitals
> 3 Rural Hospitals
> 3 Primart Care Clinic for the

Ambulance Services

> Emergency Ambulance
> Patient Transport Ambulance
> Response Unit
> Charity Ambulance

You Feel We Heal

DOCTOR REGISTERATION PAGE



First name : Enter First name

Last name : Enter Last name

Password: Enter password

Gender:
○Male
○Female

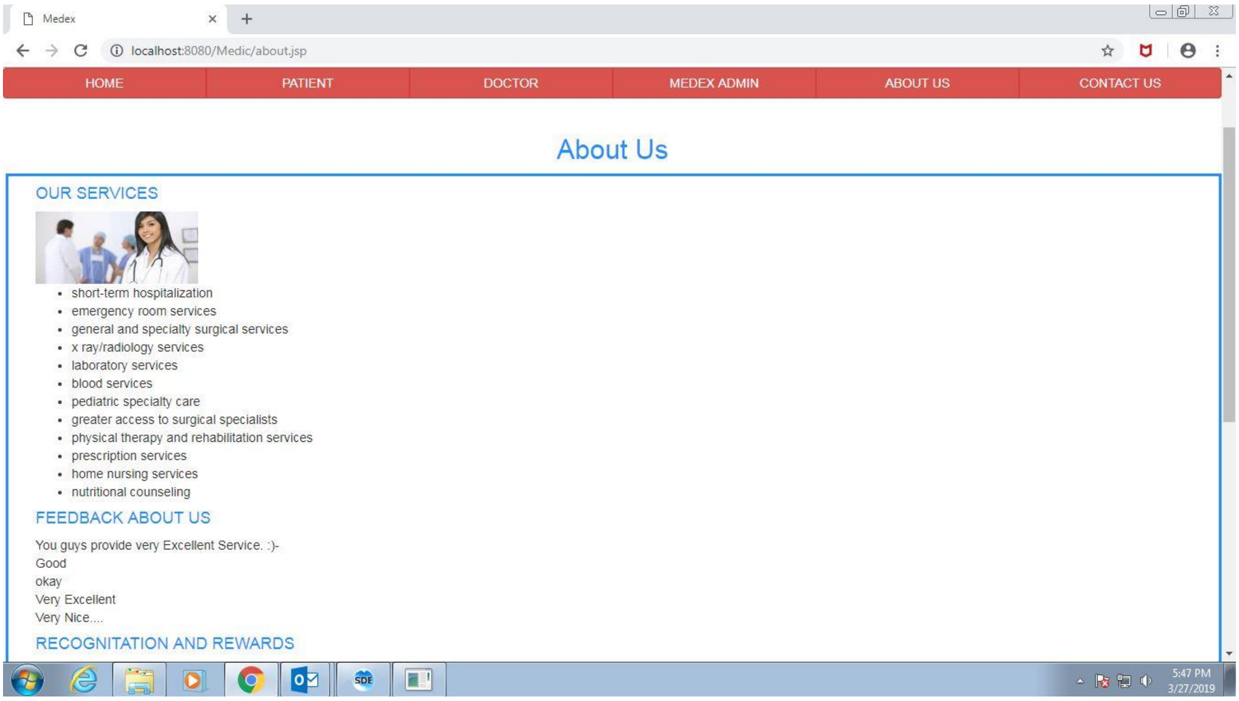
Age: Enter Age

Qualification: Enter Qualification

Specialization:

Contact Number: Enter Contact Number

ABOUT US PAGE



About Us

OUR SERVICES

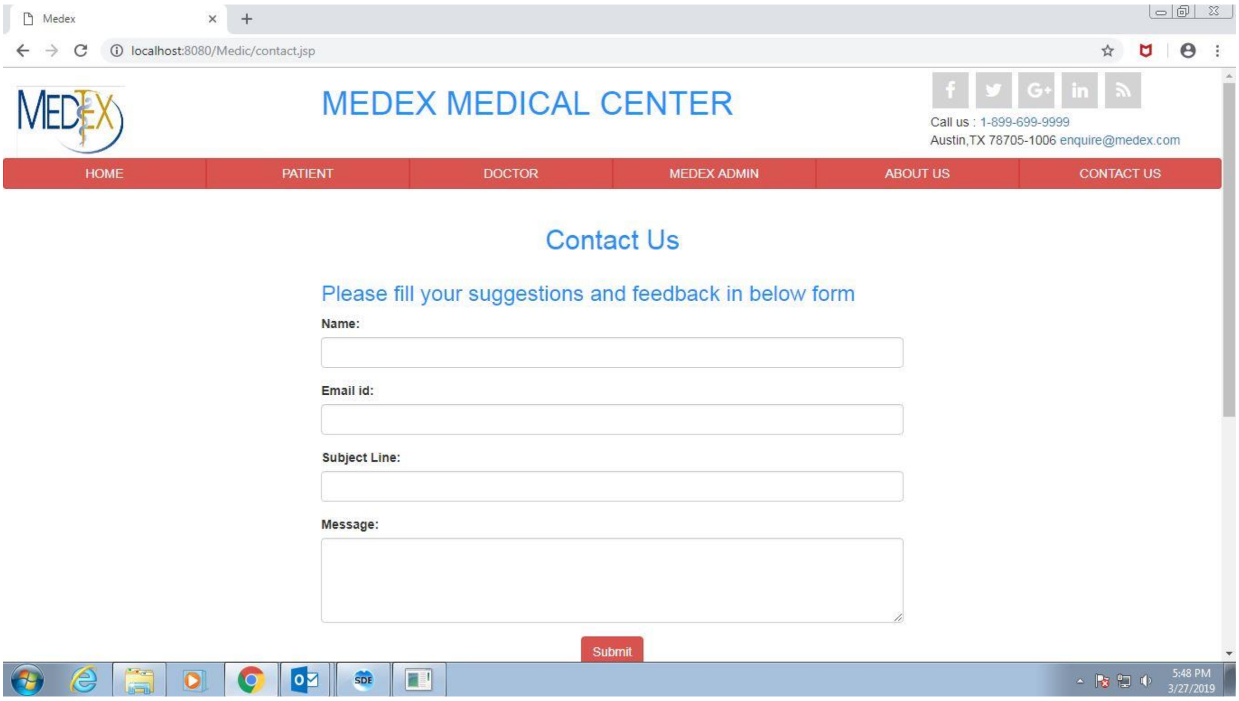
- short-term hospitalization
- emergency room services
- general and specialty surgical services
- x ray/radiology services
- laboratory services
- blood services
- pediatric specialty care
- greater access to surgical specialists
- physical therapy and rehabilitation services
- prescription services
- home nursing services
- nutritional counseling

FEEDBACK ABOUT US

You guys provide very Excellent Service. :)-
Good
okay
Very Excellent
Very Nice....

RECOGNITATION AND REWARDS

CONTACT US PAGE



MEDEX MEDICAL CENTER

Call us : 1-899-699-9999
Austin,TX 78705-1006 enquire@medex.com

Contact Us

Please fill your suggestions and feedback in below form

Name:

Email id:

Subject Line:

Message:

CHAPTER 5

5.1 RESULTS AND DISCUSSION

This Java Web Application automates and manages patient appointments booking. The application helps doctor to view the patient's data easily. It saves the patient's and doctor's time. As a result, the application is highly secured and increased in performance, the system is user friendly, efficient, reliable and easily maintainable.

CHAPTER 6

TESTING

6.1 UNIT TESTING

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated. In this process, each module (such as admin, user etc.) are considered and individual units are tested for proper operation. If each module meets up with the user's requirement, then it is subjected to integration testing where more than one module is integrated and tested. In Servlet Framework Test Complete allows to run automated tests for desktop applications.

6.2 REGRESSION TESTING

This testing is carried out to ensure that the later developed code will not affect the features of the earlier developed module. For example, the booking invoice that is to be generated is carried out only after all the calculations are accurately processed. If any error may reflect in the calculations due to the generation, then the code must be changed accordingly. Each module must be tested to produce a satisfactory outcome.

6.3 VERIFICATION

Verification is a static practice of verifying documents, design, code and program. It includes all the activities associated with producing high quality software: inspection, design analysis and specification analysis. It is a relatively objective process. Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful. Verification is concerned with whether the system is well-engineered and error-free.

6.4 VALIDATION

The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. Validation is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

CHAPTER 7

CONCLUSION AND FUTURE WORKS

CONCLUSION

Web application MEDEX MEDICAL CENTER manages and automates the patient appointment booking. It will help to reduces the time of patients and doctors. It will also reduce the paper work in managing data. Doctor can easily view the patient details. It is efficient, secured and user friendly.

FUTURE WORKS

Research, as we know is a continuous process and this is not an exception. This has successfully been carried out and now in a better position to make necessary recommendation and suggestion. The recommendation for the future research and addition to the project are given below.

1. As the technology emerges, it is possible to upgrade the system.
2. Security can be improved using emerging technologies.
3. Additionally modules can be added.

APPENDICES

SOURCE CODE

JSP CODE

```
<%@ page language="java" contentType="text/html; charset=ISO-8851"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head> <title>Medex</title><meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css"><script src="https://code.jquery.com/jquery-
1.12.4.min.js"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
<script class="jsbin
src="https://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<link rel="stylesheet" href="styles/header_style.css">
<link rel="stylesheet" href="styles/pat_reg.css">
<link rel="stylesheet" href="styles/about_style.css">
</head><body><div class="header">
<div class="container col-xs-12" style="width:100%"><div class="row" >
<div class="col-xs-2 "><br/><img src="images/logo.png" width=120 height=80>
</div><div class="col-xs-6 col-xs-offset-1">
<h1 style="color:dodgerblue;">MEDEX MEDICAL CENTER</h1></div>
<div class="col-xs-3 "><a href="#" class="fa fa-facebook"></a>
<a href="#" class="fa fa-twitter"></a><a href="#" class="fa fa-google-plus"></a>
<a href="#" class="fa fa-linkedin"></a><a href="#" class="fa fa-rss"></a><p >
Call us : <a href="#">1-899-699-9999</a><br>Austin,TX 78705-1006
```

```
<a href="#">enquire@medex.com</a></p></div></div></div><br><br><br><br>
<div class="container row"><div class="col-xs-7 col-xs-offset-1">
<a href=""><img src="images/img.jpg" width=100% height=100% ></a></div>
<div class="col-xs-4"><br><br><br>
<form method="post" action="login_serv" onsubmit="validate();"><div class="form-group">
<label for="login_id">Your Login Id (Required):</label>
<input type="text" class="form-control" id="pid" placeholder="Enter login Id"
name="pid"
value=<%= ( (session.getAttribute("id")!=null) ? session.getAttribute("id") : "" )
%>></div>
<div class="form-group"><label for="pwd">Password:</label>
<input type="password" class="form-control" id="pwd" placeholder="Enter password"
name="pwd" ></div><br>
<input type="submit" class="btn btn-danger " value="Sign In" onclick="validate();">
<input type="submit" class="btn btn-danger " value="Reset" >
<a href="register_pat.jsp" class="btn btn-danger " type="button" id="register">Register
(only for Patient)</a></form></div></div><script type="text/javascript">
function validate()
{var a = document.getElementById("pid");
var b = document.getElementById("pwd");
if(a.value.length<=0 && b.value.length<=0)
window.alert("Error:Login Id and Password Cannot be Blank");
else if(b.value.length<=0)
window.alert("Error:Password Cannot be Blank");
else if(a.value.length<=0)
window.alert("Error:Login Id Cannot be Blank");}
</script>
```

BEAN CODE

```
package org.medex.beans;
public class User {
private String id;
private String pwd;
private String role;
private String name;
public User() {
super();}
public User(String id, String pwd, String role, String name) {
super();
this.id = id;
this.pwd = pwd;
this.role = role;
this.name = name;}
public String getId() {
return id;}
public void setId(String id) {
this.id = id;}
public String getPwd() {
return pwd;}
public void setPwd(String pwd) {
this.pwd = pwd;}
public String getRole() {
return role;}
public void setRole(String role) {
this.role = role;}
public String getName() {
return name;}
public void setName(String name) {
this.name = name;}}
```

CONTROLLER CODE

```
package org.medex.controller;
import java.io.IOException;
import java.io.PrintWriter;
import java.lang.reflect.Type;
import java.util.List;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.medex.beans.Display;
import org.medex.beans.User;
import org.medex.service.DoctorService;
import org.medex.service.DoctorServiceImpl;
import org.medex.service.LoginService;
import org.medex.service.LoginServiceImpl;
import org.medex.service.PatientService;
import org.medex.service.PatientServiceImpl;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
@WebServlet("/login_serv")
public class LoginServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
LoginService log_serv = new LoginServiceImpl();
DoctorService doc_serv=new DoctorServiceImpl();
User u = new User();
User res = new User();
RequestDispatcher rd=null;
List<Display> app_id=null;
List<Display> name=null;
Display d=new Display();
PatientService pat_serv=new PatientServiceImpl();
PrintWriter pw=null;
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

```
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
String id = request.getParameter("pid");
String pwd = request.getParameter("pwd");
String role = "";
u.setId(id);
u.setPwd(pwd);
res = log_serv.validateUser(u);
if (res != null) {
role = res.getRole();
HttpSession session=request.getSession();
session.setAttribute("id", id);
session.setAttribute("name",res.getName());
session.setAttribute("role",res.getRole());
if (role.contentEquals("patient")) {
rd = request.getRequestDispatcher("patient.jsp");
d.setPatId(id);
app_id=pat_serv.viewAppointment(d);
Type listType = new TypeToken<List<Display>>() {}.getType();
Gson gson = new Gson();
String j = gson.toJson(app_id, listType);
request.setAttribute("list",j);
} else if (role.contentEquals("doctor")) {
rd = request.getRequestDispatcher("doctor.jsp");
d.setDocId(id);
name=doc_serv.viewDiag(d);
Type listType = new TypeToken<List<Display>>() {}.getType();
Gson gson = new Gson();
String j = gson.toJson(name, listType);
System.out.println(j);
request.setAttribute("list",j);
}else if (role.contentEquals("admin")) {
rd = request.getRequestDispatcher("admin.jsp");}
}else {
rd = request.getRequestDispatcher("login.jsp");}
rd.forward(request, response);
}
}
```

DBUTIL CODE

```
package org.medex.util;
public class DBConstants {
public static final String DRIVER="com.mysql.jdbc.Driver";
public static final String URL="jdbc:mysql://localhost:3306/medex";
public static final String UNAME="root";
public static final String PWD="root";
}
package org.medex.util;
import java.sql.Connection;
import java.sql.DriverManager;
public class DBUtil {
public static Connection getConnection(String driver, String url,
String uname, String pwd) {
Connection con = null;
try {
Class.forName(driver);
con = (Connection) DriverManager.getConnection(url, uname, pwd);
}catch (Exception e) {
e.printStackTrace();}
return con;
}
}
```

SERVICE CODE:

```
package org.medex.service;
import org.medex.beans.User;
public interface LoginService {
```

```
User validateUser(User u);
}
package org.medex.service;
import org.medex.beans.User;
import org.medex.dao.LoginDao;
import org.medex.dao.LoginDaoImpl;
public class LoginServiceImpl implements LoginService {
LoginDao log_dao = new LoginDaoImpl();
@Override
public User validateUser(User u) {
User res_user = log_dao.selectUser(u);
return res_user;
}
}
```

DAO CODE

```
package org.medex.dao;
import org.medex.beans.User;
public interface LoginDao {
User selectUser(User u);
}
package org.medex.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import org.medex.beans.User;
import org.medex.util.DBConstants;
import org.medex.util.DBUtil;
public class LoginDaoImpl implements LoginDao {
@Override
public User selectUser(User u) {
String name = "";
String role = "";
```

```
Connection con = null;
PreparedStatement pst1 = null, pst2 = null;
ResultSet rs1 = null, rs2 = null;
User res = new User();
try {
con = DBUtil.getConnection(DBConstants.DRIVER, DBConstants.URL,
DBConstants.UNAME, DBConstants.PWD);
pst1 = (PreparedStatement) con
.prepareStatement("select role from user where id=? and pwd=?");
pst1.setString(1, u.getId());
pst1.setString(2, u.getPwd());
rs1 = pst1.executeQuery();
if (rs1.next()) {
role = rs1.getString(1);
if (role.contentEquals("patient")) {
pst2 = (PreparedStatement) con
.prepareStatement("select fname,lname from patient where pid=?");
pst2.setString(1, u.getId());
rs2 = pst2.executeQuery();
if (rs2.next()) {
name = rs2.getString(1) + " " + rs2.getString(2);
res.setName(name);
res.setRole(role);
}
rs2.close();
pst2.close();
} else if (role.contentEquals("doctor")) {
pst2 = (PreparedStatement) con
.prepareStatement("select fname,lname from doctor where did=?");
pst2.setString(1, u.getId());
rs2 = pst2.executeQuery();
if (rs2.next()) {
name = rs2.getString(1) + " " + rs2.getString(2);
res.setName(name);
res.setRole(role);}
rs2.close();
pst2.close();}
else  {
role="admin";
res.setRole(role);
}
}
else {
res=null;}
rs1.close();
pst1.close();
rs2.close();
pst2.close();
con.close();
} catch (Exception  e) {
e.printStackTrace();
}
return res;
}
}
```