VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590 018

An Internship Project Report
on

"CARE AND CURE - ONLINE HOSPITAL MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirements for the 6th semester of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi

Submitted By

MUDDULUR LIKHITH VARMA 1RN19CS083

KUMMARA RAGHU LOCHAN 1RN19CS070

Carried out at

TechieAid

Under the guidance of:

Internal Guide:
Mrs. Chethana H R
Assistant Professor
Department of CSE

External Guide:
Mr. Girish
TechieAid
Bengaluru

ESTD:2001
An Institute with a Difference

Department of Computer Science and Engineering

RNS Institute of Technology

(NBA Accredited for academic years 2018-19, 2019-20,2020-21, 2021-22)

Dr. Vishnuvardhan Road, Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvardhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru - 560098

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(NBA Accredited for academic years 2018-19, 2019-20,2020-21, 2021-22)

CERTIFICATE

Certified that the Internship/Professional Practice work entitled "Care and Cure - Online Hospital Management System" has been successfully carried out at "TechieAid" by Muddulur Likhith Varma (1RN19CS083) and Kummara Raghu Lochan (1RN19CS070), Bonafide students of RNS Institute of Technology, Bengaluru in partial fulfillment of the requirements of 6th Semester Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during academic year 2021-2022. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mrs. Lakshmi R **Internal Guide** **Assistant Professor** **Department of CSE**	**Dr. Kiran P** **Professor and HoD** **Department of CSE** **RNSIT**	**Dr. M K Venkatesha** **Principal** **RNSIT**

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ABSTRACT

Care And Cure - An Online Hospital Management website is a computerized system that allows patients to book doctor appointments easily online. The project is aimed at exposing the relevance and importance of Online Hospital Management System. It is so convenient and easy, that anyone can book an appointment with a doctor on the time slot of their own choice for any problem right from a living room, with just a few clicks.

Online appointment booking has evolved more with the emergence of smart phones, where now, you can book from anywhere and anytime, with a wireless device connected to the Internet. You can search for any service online, without having to go anywhere physically or call up someone at the hospital to book an appointment.

This is a user-friendly website where the patient can register himself giving his details and log in with the same details and search for the desired doctor of a particular department according to their issue and book a slot of desired time from the available slots. Admin can log in with his details and add or remove the staff from the website and he can view all details of current appointments, staff and the profits received from all the appointments.

ACKNOWLEDGEMENT

At the very onset, I would like to place on record my gratitude to all those people who have helped me in making this Internship work a reality. Our Institution has played a paramount role in guiding in the right direction.

I would like to thank our beloved Principal, Dr. M K Venkatesha, for providing the necessary facilities to carry out this work.

I am extremely grateful to Dr. Kiran P, Professor and Head, Department of Computer Science and Engineering for having accepted to patronize me in the right direction with all his wisdom.

I would like to express my sincere thanks to our Coordinator and guide, Mrs. Lakshmi R, Assistant Professor, for her constant encouragement that motivated me for the successful completion of this work.

Last but not the least, I am thankful to all the teaching and non-teaching staff members of the Computer Science and Engineering Department for their encouragement and support throughout this work.

Muddulur Likhith Varma
1RN19CS083

Kummara Raghu Lochan
1RN19CS070

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Acronym	Description
ASP	**Active Server Pages**
SQL	**Structured Query Language**
MSSQL	**Microsoft SQL Server**
HTML	**Hypertext Markup Language**
CSS	**Cascading Style Sheets**
CLR	**Common Language Runtime**
IE	**Internet Explorer**
VB	**Visual Basics**
ISO	**International Organization of Standardization**
ANSI	**American National Standard Institutes**

CHAPTER 01

1. INTRODUCTION

1.1 ORGANIZATION

1.1.1 Company Profile

TechieAid is a software company based in Banashankari, Bengaluru. They specialize in Software Development, Software Testing, Technology Training, Leadership Training, and Staffing Solutions. They are IT services company registered under Govt. of India, Ministry of Micro, Small & Medium Enterprises. They offer IT Staffing Solutions, Technology Trainings, Leadership & Soft Skills Training, Software Development Services and Coaching & Mentoring. Their clients are reputed MNC companies and leading academic institutions.

1.1.2 Domain/Technology

The project is a Full stack web development project in C#, ASP.NET. The project also uses HTML, CSS, Bootstrap, JavaScript for front-end development and for the connectivity between front-end and back-end. The project uses MS-SQL database for the storage of data.

C# is a general-purpose, object-oriented programming language that is structured and easy to learn. It runs on Microsoft's .Net Framework and can be compiled on a variety of computer platforms.

C# is a boon for developers who want to build a wide range of applications on the .NET Framework Windows applications, Web applications, and Web services—in addition to building mobile apps, Windows Store apps, and enterprise software. It is thus considered a powerful programming language and features in every developer's cache of tools. The following are a few of the .NET applications that use ASP.NET to connect to a database, execute commands and retrieve data from the database.

- ASP.NET Web Applications
- Console Applications
- Windows Applications

Merits of C#:

- Being an object-oriented language, C# allows you to create modular, maintainable applications and reusable codes.
- Easy to develop as it has a rich class of libraries for smooth implementation of functions.
- Enhanced integration as an application written in .NET will integrate and interpret better when compared to other NET technologies
- As C# runs on CLR, it makes it easy to integrate with components written in other languages.
- It's safe, with no data loss as there is no type-conversion so that you can write secure codes.
- The automatic garbage collection keeps the system clean and doesn't hang it during execution and cross-platform support as it requires to run on NET Framework.

1.2 PROBLEM STATEMENT

1.2.1 Existing System

The current system was having static data and minimal operations made available to the user such as reading the information about the doctors and available slots. It had various information like about the doctor. It had various information about the company, and contact details rendered in the single web page of the application.

1.2.2 Proposed Solution

The existing web application has been enhanced to make the application more interactive and user-friendly with added functionality. The features like user login and registration have been made available for the patients and the doctors. Patients will be able to book new appointments and also view the details of previous appointments, billing history, prescriptions given by the doctor during the previous visit etc. Doctors will be able to accept or deny the appointments requested by the patients and give prescriptions to patients.

The admin will be able to view the complete data of the hospital i.e., information about all the patients, doctors and staff and can add or delete staff as needed.

1.2.3 Problem Formulation

This system is designed to assist in strategic planning and will help you ensure that your organization is equipped with the right level of information and details for your future goals. For those persons who tend for last minute appointment booking from their home, working area, etc., our system enables them to do so at their fingertips. These systems will allow you to better manage resources.

The system collects all necessary details from the user as well as logs every new appointment detail that is booked on the website. The procedure of collection of data will be obvious, simple, and sensible. It reduces the manual efforts of a person who must go to the hospital and wait in queues for the booking the appointment. This system allows the patient to book appointments with the available doctors on the available time slots sitting at the place of his comfort.

CHAPTER 02

2. REQUIREMENTS ANALYSIS, TOOLS, AND TECHNIQUES

2.1 Hardware and Software Requirements

2.1.1 Software Requirements

Name of Components	Specification
Operating System	Windows 10
Language	HTML, CSS, C#, JavaScript, Bootstrap
Database	MSSQL
Browser	Google Chrome
Integrated Development Environment	Microsoft Visual Studio 2019, Microsoft SQL Server 2019

2.1.2 Hardware Requirements

Name of Components	Specification
Processor	8^{TH} Gen CORE i3 Processor
RAM	8GB
Hard Disk	128 GB SSD

2.2 Functional Requirements

- **Input:** The patient has to register himself. The admin who is already registered will allocate the role to the user at the back-end. The admin can manage the users' details while the patients can book appointments and the doctors can accept or deny the appointment requests and give the prescription according to the patients needs which will get stored in the database.
- **Output:** The user will be able to get details about available doctors, appointment time slots etc if he is a patient and details about the pending appointments and current appointments if he is a doctor.

- **User Interface:** The user interface is in HTML and JavaScript calls are often used for defining functions. The interface is user friendly and hence can be used even by novice users.

2.3 Tools, Languages and Platform

The project uses C#, ASP.Net languages as a base. HTML, CSS, Bootstrap used for better front-end experience. The project uses MS-SQL for storing data in the database that is obtained through the user at the front-end. The whole project was done on Microsoft Visual Studio 2019 and Microsoft SQL Server.

CHAPTER 03

3. SYSTEM DESIGN

3.1 Schema of the database

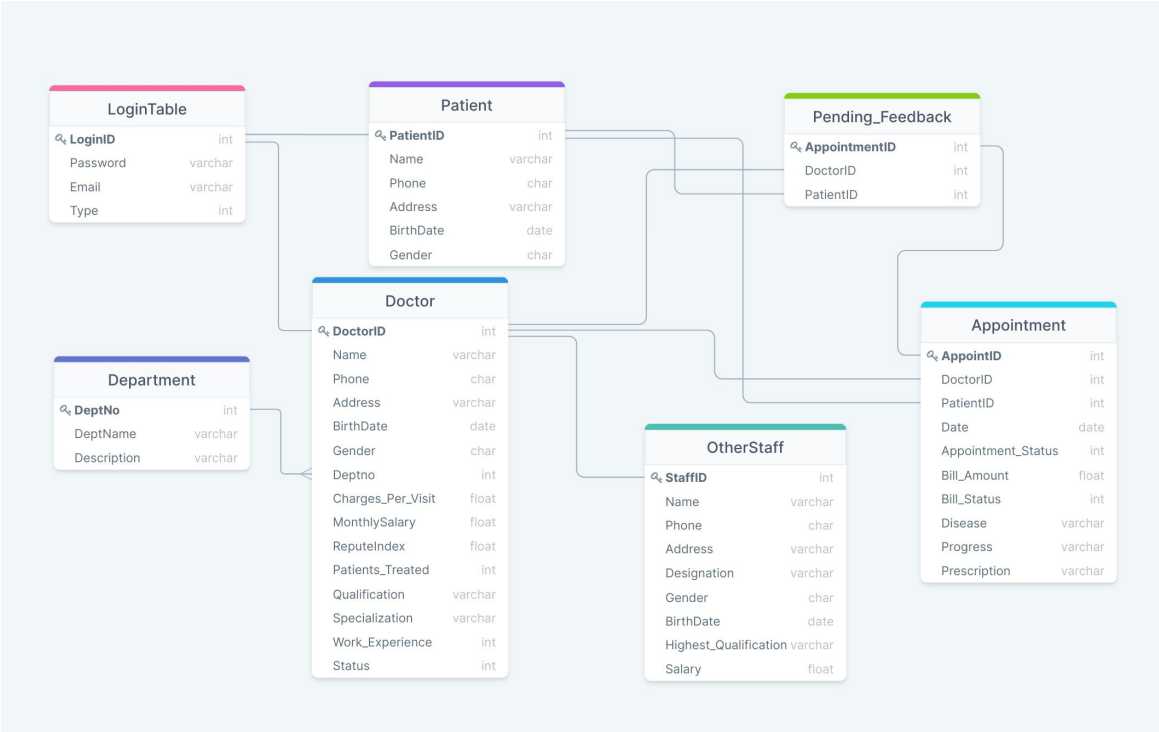


Fig 3.1.1: Relational Schema Diagram

This schema shows the relationship between the primary and foreign keys of all the tables.

3.2 Block Diagram

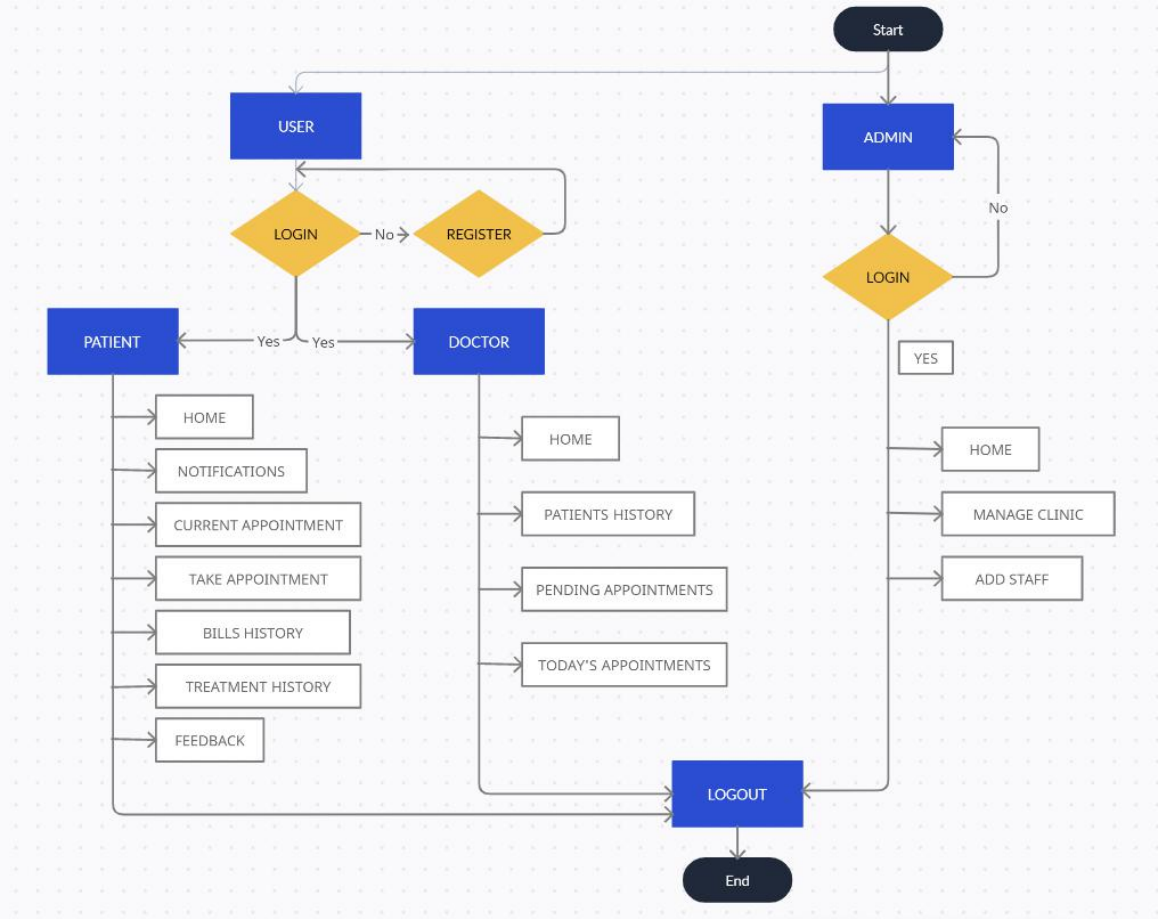


Fig 3.2.1: Block diagram

This diagram shows the entire flow of the working of the program. If it's a new user, then they need to register or if they are an existing user then they can login to access the home page. This page has all the features that the website provides. The user can choose the doctor of their choice and book an appointment if the user is a patient. If the user is a doctor, he can accept or deny appointment requests and give prescriptions to the patients. If the admin logs in, he can view all details of patients, doctors and other staff and add or remove staff members and doctors.

3.3 Database Description

1. Tables:

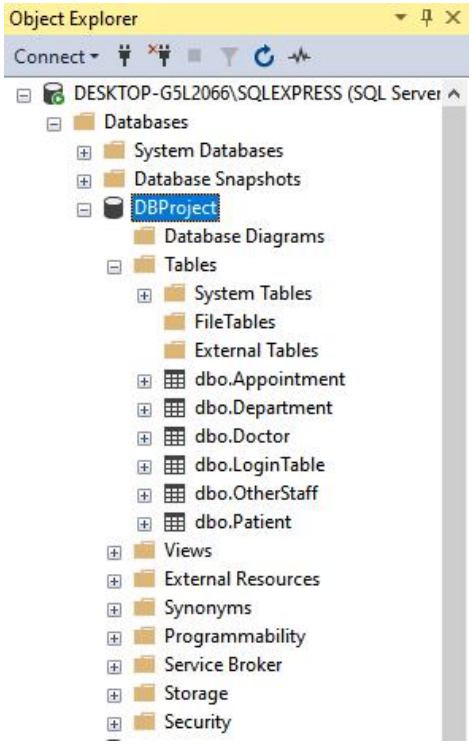


Fig 3.3.1.1 Tables

2. Patient Table Data: It contains all the details of patients registered.

DESKTOP-G5L2066\S...ject - dbo.Patient | DESKTOP-G5L2066\S...t - dbo.OtherStaff | DESKTOP-G5L2066\S...

	PatientID	Name	Phone	Address	BirthDate	Gender
▶	12	Sai Teja	61536516	Chennai	1996-04-04	M
	13	Ananth	61536516	Mysore	1996-04-04	M
	14	Uday	61536516	Tirupati	1996-04-04	M
	29	test	09696969696	TEST	2001-01-02	M
*	NULL	NULL	NULL	NULL	NULL	NULL

Fig 3.3.2.1 Patient Table

3.Appointment Table Data: It contains data about the scheduled appointments till date.

DESKTOP-G5L2066\...- dbo.Appointment | DESKTOP-G5L2066\S...t - dbo.LoginTable

	AppointID	DoctorID	PatientID	Date	Appointment_...	Bill_Amount
▶	1	9	29	2022-06-14 12:0...	3	1000
	2	9	29	2022-06-14 14:0...	3	1000
	3	4	14	2022-06-14 09:0...	3	1500
*	NULL	NULL	NULL	NULL	NULL	NULL

Fig 3.3.3.1 Appointment Table

Bill_Status	DoctorNotifica...	PatientNotific...	FeedbackStatus	Disease	Progress	Prescription
paid	2	2	2	NULL	NULL	NULL
Unpaid	2	1	1	Tongue	Diagnosed	Dolo
paid	2	1	1	Heart Disease	Treatment in pr...	Dolo
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 3.3.3.2 Appointment Table

4.Login Table Data: It contains the data about the users login details.

DESKTOP-G5L2066\S...t - dbo.LoginTable

	LoginID	Password	Email	Type
▶	1	admin	admin@clinic.com	3
	2	noor	noor@gmail.com	2
	3	raghu	raghu@gmail.com	2
	4	likhith	likhith@gmail.com	2
	5	sharon	sharon@gmail.com	2
	6	nikhil	nikhil@gmail.com	2
	7	koushik	koushik@gmail.com	2
	8	chirag	chirag@gmail.com	2
	9	sumanth	sumanth@gmail.com	2
	10	yashwanth	yashwanth@gmail.com	2
	11	preetham	preetham@gmail.com	2
	12	saiteja	saiteja@gmail.com	1
	13	ananth	ananth@gmail.com	1
	14	uday	uday@gmail.com	1
	29	test	test@gmail.com	1
*	NULL	NULL	NULL	NULL

Fig 3.3.4.1 Login Table

5.Doctor Table Data: It contains the available doctors details.

DESKTOP-G5L2066\S...oject - dbo.Doctor | DESKTOP-G5L2066\S...- dbo.Department | DESKTOP-G5L2066\...-

DoctorID	Name	Phone	Address	BirthDate	Gender
2	Noor	156133213	Pune	1996-04-12	M
3	Raghu	156133213	Bangalore	1996-12-12	M
4	Likhith	156133213	Hyderabad	1996-12-12	M
5	Sharon	156133213	Vijayawada	1990-05-04	M
6	Nikhil	156133213	Vizag	1990-05-04	M
7	Koushik	156133213	Tirumala	1990-05-04	M
8	Chirag	156133213	Mysore	1990-05-04	M
9	Sumanth	156133213	Tumkur	1990-05-04	M
10	Yashwanth	156133213	Bangalore	1990-05-04	M
11	Preetham	156133213	Tirupati	1990-05-04	M
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Fig 3.3.5.1 Doctor Table

DeptNo	Charges_Per_V...	MonthlySalary	ReputeIndex	Patients_Treated	Qualification
1	2500	30000	4	0	PHD IN EVERY ...
1	3000	25000	3.5	0	PHD IN EVERY ...
1	1500	20000	0	1	PHD IN EVERY ...
1	1000	15000	4.5	0	PHD IN EVERY ...
3	1000	15000	4.5	0	PHD IN EVERY ...
2	1000	15000	4.5	0	PHD IN EVERY ...
2	1000	15000	4.5	0	PHD IN EVERY ...
3	1000	15000	2.25	2	PHD IN EVERY ...
4	1000	15000	4.5	0	PHD IN EVERY ...
5	1000	15000	4.5	0	PHD IN EVERY ...
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Fig 3.3.5.2 Doctor Table

6.Departments Table Data: It contains the data about available departments.

DESKTOP-G5L2066\S...- dbo.Department

	DeptNo	DeptName	Description
▶	1	Cardiology	We have the be...
	2	Orthopaedics	Orthopedic sur...
	3	ENT	They are gentle....
	4	Physiotherapy	Physiotherapis...
	5	Neurology	A medical speci...
*	NULL	NULL	NULL

Fig 3.3.6.1 Department Table

7.Other Staff Table Data: It contains the data on other staff.

DESKTOP-G5L2066\S...t - dbo.OtherStaff | DESKTOP-G5L2066\S...oject - dbo.Doctor | DESKTOP-G5L2066\S...

	StaffID	Name	Phone	Address	Designation	Gender
▶	1	JEEVAN	03227561002	tirupati	GUARD	M
	2	HAMSA	03227561002	hyderabad	SWEEPER	M
	3	RAMESH	03227561002	tirupati	SECURITY	M
	4	GIRISH	03227561002	bangalore	GUARD	M
	5	KIRAN	03227561002	mysore	GUARD	M
	6	SURESH	03227561002	bangalore	GUARD	M
	7	PRAKASH	03227561002	pune	GUARD	M
	8	DINESH	03227561002	mumbai	SWEEPER	M
	9	PAVAN	03227561002	delhi	SECURITY	M
	10	VIKKY	03227561002	kashmir	GUARD	M
	11	HAPPY	03227561002	darjeeling	GUARD	M
	12	SIRAJ	03227561002	gangtok	GUARD	M
*	NULL	NULL	NULL	NULL	NULL	NULL

Fig 3.3.7.1 OtherStaff Table

3.4 User Interface Description

The front-end is built using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript, Bootstrap and Cascading Style Sheets (CSS). Front-end developers design and construct the user experience elements on the web page or app including buttons, menus, pages, links, graphics and more.

3.4.1 Hypertext Markup Language

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages, HTML, describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images, and other objects, such as interactive forms, may be embedded into the rendered page. It provides a direct link. means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img> and <input/> introduce content into the page Others such as ... surround and provide information about document text and may include other tags as sub -elements. Browsers do not display the HTML tags but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

3.4.2 Cascading Style Sheets

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML. document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate CSS file, and reduce complexity and repetition in the structural content.

3.4.3 JavaScript

JavaScript is the Programming Language for the Web. It can update and change both HTML and CSS. JavaScript can calculate, manipulate, and validate data. JavaScript is a dynamic computer programming language. It is lightweight and most used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

3.4.4 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

- Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing a responsive and mobile friendly website.
- It is free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.
- It facilitates you to create responsive designs.

3.5 DATABASE IMPLEMENTATION

The data store has been designed and developed by creating the entity relation diagram and schema design. The table structure and its underlying back-end layer has been implemented using Structured Query Language using MSSQL Server.

3.5.1 MSSQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

3.5.2 SQL

SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See-Quell. This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). It is also designed for stream processing in RDSMS.

You can easily create and manipulate the database, access, and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 198 6 and ISO in the year of 1987.If you want to get a job in the field of data science, then it is the most important query language to learn. Big enterprises like Facebook, Instagram, and LinkedIn, use SQL for storing the data in the back end.

- The basic use of SQL for data professionals and SQL users is to insert, update, and delete the data from the relational database.

- SQL allows the data professionals and users to retrieve the data from the relational database management systems.
- It also helps them to describe the structured data.
- It allows SQL users to create, drop, and manipulate the database and its tables.
- It also helps in creating the view, stored procedure, and functions in the relational database.
- It allows you to define the data and modify that stored data in the relational database.
- It also allows SQL users to set the permissions or constraints on table columns, views, and stored procedures.

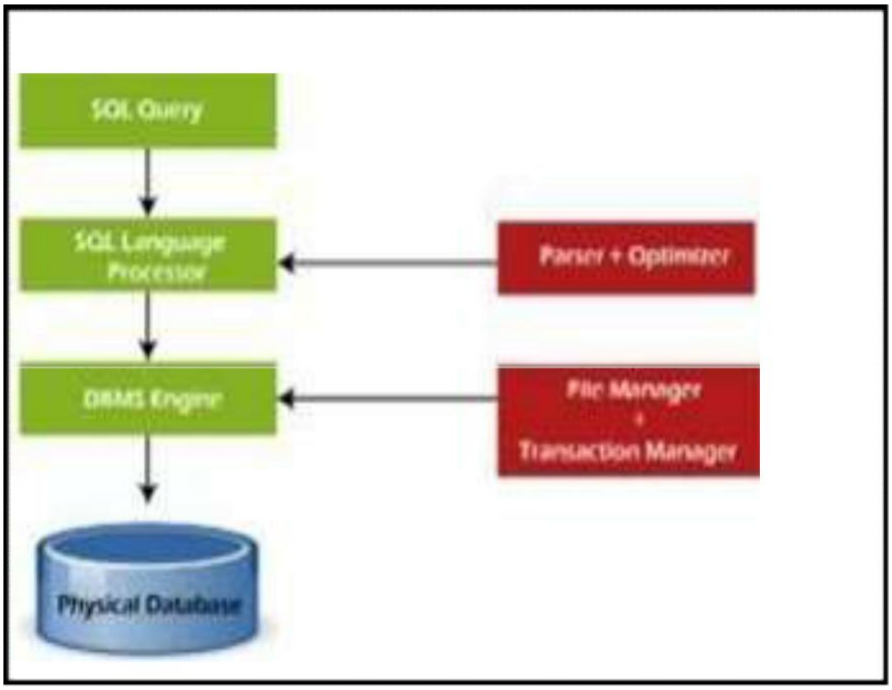
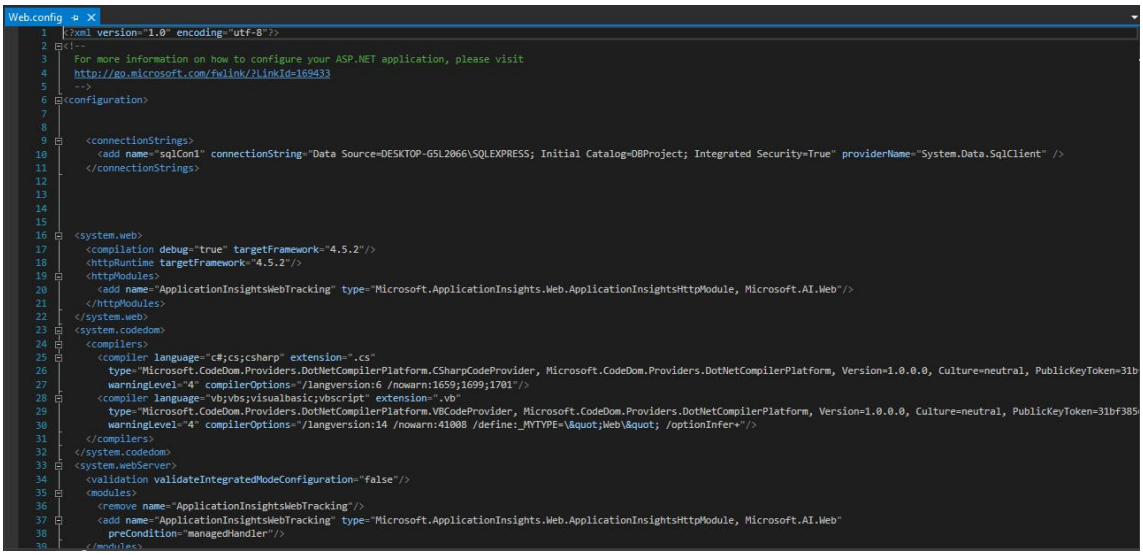


Fig 3.5.1 SQL Process Architecture diagram

3.6 CODE SEGMENT/ALGORITHM

Web Config: The below code defines how the database should be rendered onto the website. The code also includes the connection string that is used to connect the controllers from Visual Studio to one of the databases from MSSQL server.



Views: The Views are coded to design the webpage.

Below is the code snippets for different views:

1. AdminHome.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Admin/Admin.Master"
AutoEventWireup="true" CodeBehind="AdminHome.aspx.cs"
Inherits="DBProject.AdminHome" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder2"
runat="server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="head" runat="server">
</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
    <form runat ="server">
        <br />
        <h1 style="font-family: 'Times New Roman', Times, serif;border-
radius:5px; text-decoration: underline; background-color: #CCCCCC"><strong
style="margin:37%">Hospital Statistics</strong></h1>
        <br /><br />
           <div style="margin-left: 70px">

            <h4><strong>Total Number of Registered Doctors: </strong></h4>
            <asp:Label ID="TotalPatients" runat="server" Font-Bold="true"
Font-Size="Medium"></asp:Label>
            <br /><br />
```

```
        <h4><strong>Total Registered Patients: </strong></h4>
                    <asp:Label ID="Total_Doctors" runat="server"  Font-Bold="true"
Font-Size="Medium"></asp:Label>
                    <br /><br />

                    <h4><strong>Total Income: </strong></h4>
                    <asp:Label ID="TotalIncome" runat="server"  Font-Bold="true"
Font-Size="Medium"></asp:Label>
                    <br /><br />
                <h3><strong style="margin:5%">Current Appointments</strong></h3>

                <asp:gridview ID="Appointment_view" runat="server" CellPadding="4"
ForeColor="Black" GridLines="Vertical" BackColor="White" BorderColor="#DEDFDE"
BorderStyle="None" BorderWidth="1px">
                    <AlternatingRowStyle BackColor="White" />
                    <FooterStyle BackColor="#CCCC99" />
                    <HeaderStyle BackColor="#6B696B" Font-Bold="True"
ForeColor="White" />
                    <PagerStyle BackColor="#F7F7DE" ForeColor="Black"
HorizontalAlign="Right" />
                    <RowStyle BackColor="#F7F7DE" />
                    <SelectedRowStyle BackColor="#CE5D5A" Font-Bold="True"
ForeColor="White" />
                    <SortedAscendingCellStyle BackColor="#FBFBF2" />
                    <SortedAscendingHeaderStyle BackColor="#848384" />
                    <SortedDescendingCellStyle BackColor="#EAEAD3" />
                    <SortedDescendingHeaderStyle BackColor="#575357" />
                </asp:gridview>
            </div>

            <div style="margin:20%">

                <h2><strong style="margin:20%">Department Information</strong></h2>

                <asp:gridview ID="department_View" runat="server" CellPadding="4"
ForeColor="Black" Height="50px" BackColor="White" BorderColor="#DEDFDE"
BorderStyle="None" BorderWidth="1px" GridLines="Vertical">
                    <AlternatingRowStyle BackColor="White" />
                    <FooterStyle BackColor="#CCCC99" />
                    <HeaderStyle BackColor="#6B696B" Font-Bold="True"
ForeColor="White" />
                    <PagerStyle BackColor="#F7F7DE" ForeColor="Black"
HorizontalAlign="Right" />
                    <RowStyle BackColor="#F7F7DE" />
                    <SelectedRowStyle BackColor="#CE5D5A" Font-Bold="True"
ForeColor="White" />
                    <SortedAscendingCellStyle BackColor="#FBFBF2" />
                    <SortedAscendingHeaderStyle BackColor="#848384" />
                    <SortedDescendingCellStyle BackColor="#EAEAD3" />
                    <SortedDescendingHeaderStyle BackColor="#575357" />
                </asp:gridview>
                </div>
                </form>
        </asp:Content>
```

2. DoctorHome.aspx:

```
        <%@ Page Title="" Language="C#"
MasterPageFile="~/Doctor/doctormaster.Master" AutoEventWireup="true"
CodeBehind="DoctorHome.aspx.cs" Inherits="doctor.doctorhome" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

<title>Doctor's Home</title>

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="Cp1" runat="server">

    <div style="background-
image:url(/assets/img/backgrounds/PatientHome.jpg); background-position:center;
background-size:20px; margin-left:50px">

                <h1>Your Profile</h1>
                 <br />
                <h3>Name:       <asp:label id="Label1" runat="server"   Font-
Bold="true" Font-Size="Medium"  /><br /><br /> </h3>
                <h4>Phone:      <asp:label id="Label2" runat="server"   Font-
Bold="true" Font-Size="Medium"
/>            
              
       Address: <asp:label id="Label3" runat="server"
Font-Bold="true" Font-Size="Medium"   />  </h4>
                 <br />
                <h4> BirthDate:   <asp:label id="Label4" runat="server"  Font-
Bold="true" Font-Size="Medium" /></h4>

                <br />
                <br />
                <h4>Gender:  <asp:label id="Label5" runat="server"   Font-
Bold="true" Font-Size="Medium"
/>            
             
             
        Department No: <asp:label id="Label6"
runat="server"   Font-Bold="true" Font-Size="Medium"  /></h4>

                <br />
                <br />
               <h4>Charges Per Visit: <asp:label id="Label7" runat="server"
Font-Bold="true" Font-Size="Medium"
/>            
               &n
bsp   Monthly Salary: <asp:label id="Label8" runat="server"   Font-Bold="true"
Font-Size="Medium"  /></h4>
                <br />
                <br />
               <h4>Repute Index: <asp:label id="Label9" runat="server"   Font-
Bold="true" Font-Size="Medium"
/>            
             
             Patients Treated: <asp:label
id="Label10" runat="server"   Font-Bold="true" Font-Size="Medium"  /></h4>
                <br />
                 <br />

                <h4>Qualification: <asp:label id="Label11" runat="server"
Font-Bold="true" Font-Size="Medium"  /> <br /><br /> Specialization: <asp:label
id="Label12" runat="server"   Font-Bold="true" Font-Size="Medium"
/>            
               </h4>
```

```
        <h4>Work Experience: <asp:label id="Label13" runat="server"   Font-
Bold="true" Font-Size="Medium"
/>             
                
Status: <asp:label id="Label14" runat="server"   Font-Bold="true" Font-
Size="Medium"  /></h4>
                    <br />
        </div>

            </asp:Content>
```

3. PatientHome.aspx:

```
        <%@ Page Title="" Language="C#"
MasterPageFile="~/Patient/PatientMaster.Master" AutoEventWireup="true"
CodeBehind="PatientHome.aspx.cs" Inherits="DBProject.PatientHome" %>

        <asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

        <title>Patient's Home</title>

        </asp:Content>

        <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">

            <div style="background-
image:url(/assets/img/backgrounds/PatientHome.jpg); background-position:center;
background-size:20px">

                <br />
                <h1><strong style="margin:37%">Your Information</strong></h1>
                <br /><br />

                <div style="margin-left: 70px">

                    <h4><strong>Name: </strong></h4>
                    <asp:Label ID="PName" runat="server" Font-Bold="true" Font-
Size="Medium"></asp:Label>
                    <br /><br />

                    <h4><strong>Phone: </strong></h4>
                    <asp:Label ID="PPhone" runat="server"  Font-Bold="true" Font-
Size="Medium"></asp:Label>
                    <br /><br />

                    <h4><strong>Birth Date: </strong></h4>
                    <asp:Label ID="PBirthDate" runat="server"  Font-Bold="true"
Font-Size="Medium"></asp:Label>
                    <br /><br />

                    <h4><strong>Age: </strong></h4>
                    <asp:Label ID="PatientAge" runat="server" Font-Bold="true"
Font-Size="Medium"></asp:Label>
                    <br /><br />

                    <h4><strong>Gender:</strong></h4>
                    <asp:Label ID="PGender" runat="server" Font-Bold="true" Font-
Size="Medium"></asp:Label>
                    <br /><br />
```

```
                    <h4><strong>Address: </strong></h4>
                    <asp:Label ID="PAddress" runat="server"  Font-Bold="true" Font-
Size="Medium"></asp:Label>
                    <br /><br />

                </div>

            </div>

        </asp:Content>
```

CHAPTER 04

4. OBSERVATIONS AND RESULTS

4.1 Home Page: This is the home page where users are allowed to register or login.

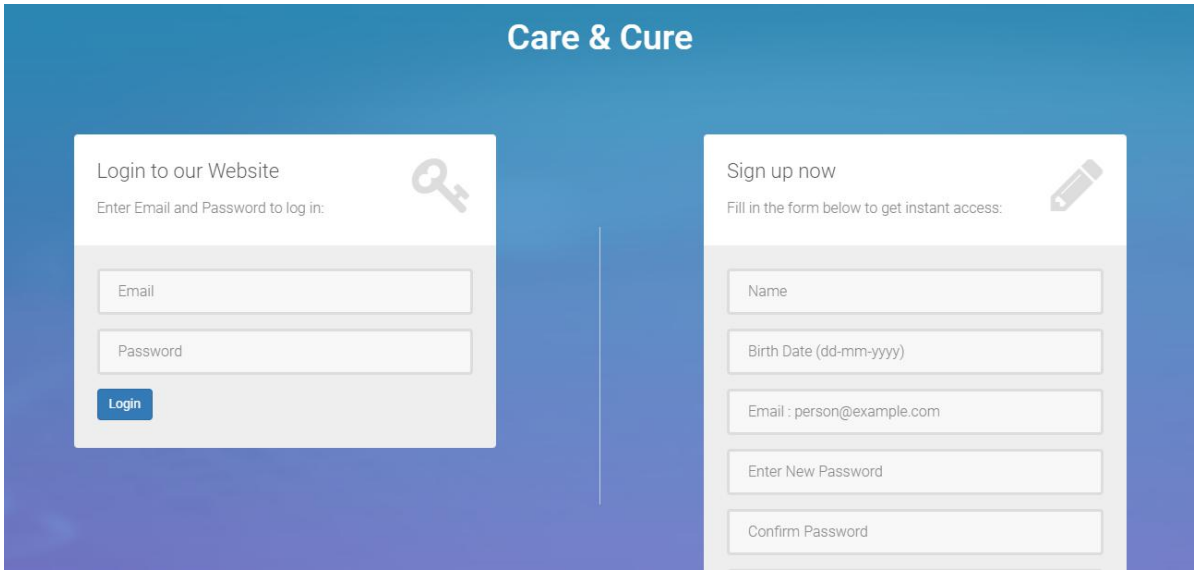


Fig 4.1.1: Homepage

4.2 Patient Homepage: This is the homepage of patient after login.

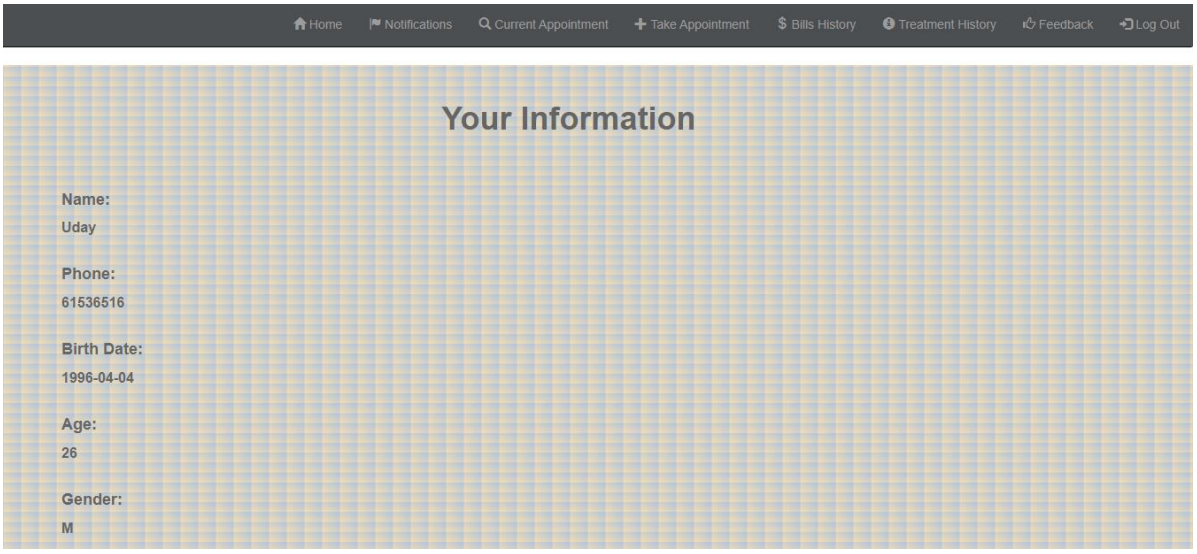


Fig 4.2.1 Valid Email Address and password

4.3 Patient Notifications: Patients can view their notifications on this page.

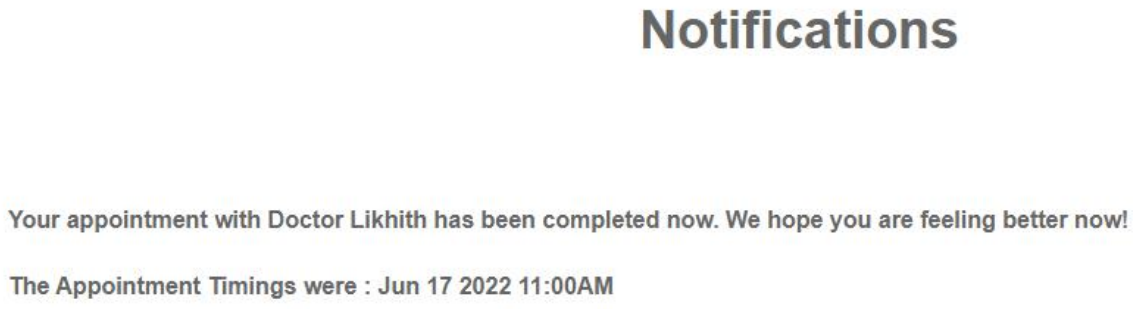


Fig 4.3.1 Patient Notifications

4.4 Current Appointments: Patients can view the status of their current appointments here.

Fig 4.4.1 Current Appointments

4.5 Take Appointment: Patients can book a new appointment on this page.

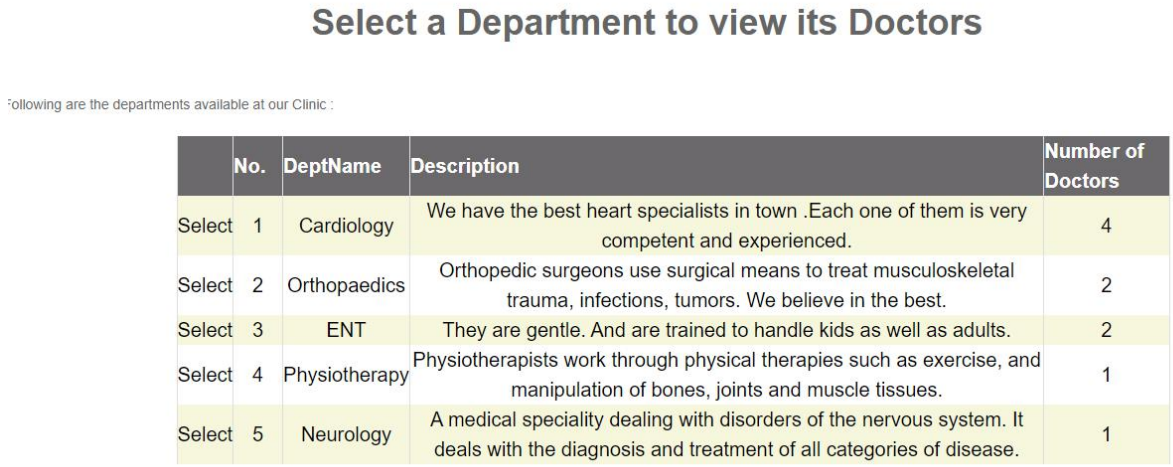


Fig 4.5.1 Take Appointment

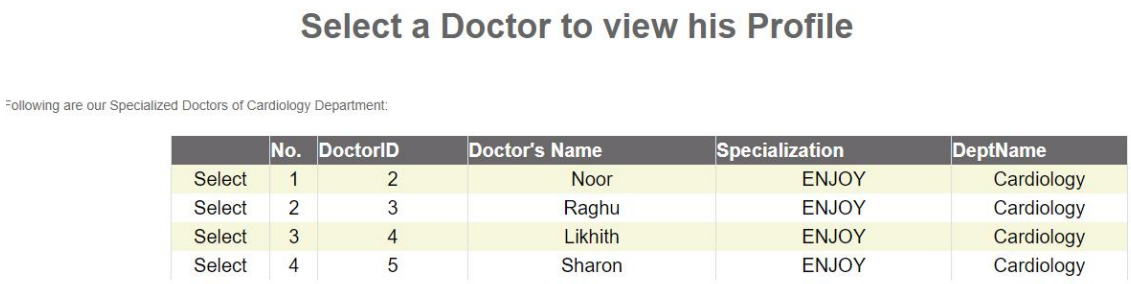


Fig 4.5.2 Take Appointment

Fig 4.5.3 Take Appointment

4.6 Bill History: The patient can view his billing history here.

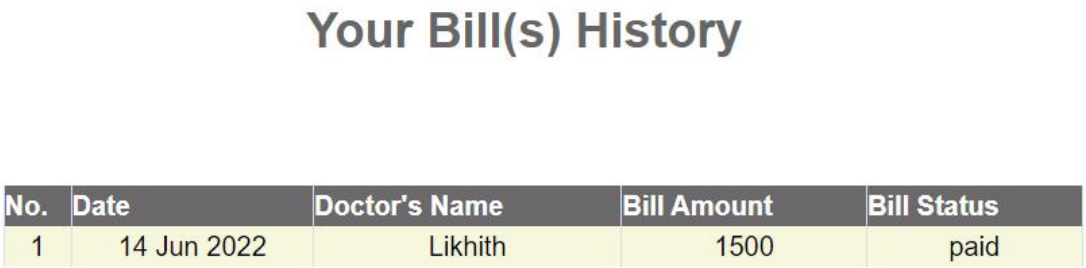


Fig 4.6.1 Bill History

4.7 Feedback: In this page patient can give feedback to the doctor he took appointment with.

Fig 4.7.1 Bill History

4.8 Doctor Home: This is the homepage of doctor after he logs in.

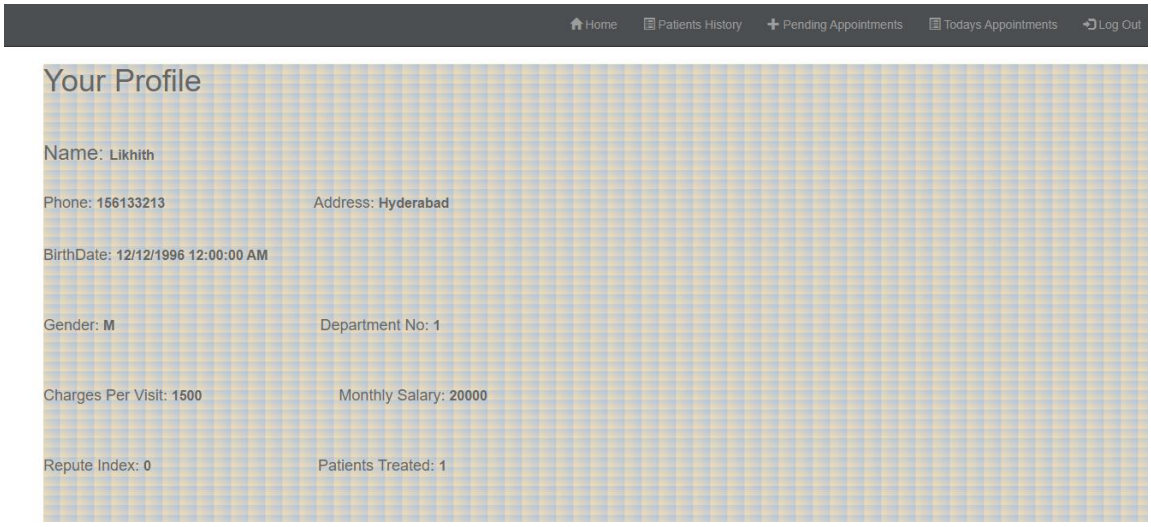


Fig 4.8.1 Doctor Home

4.9 Patient History: This page shows the patients history of the logged in doctor.

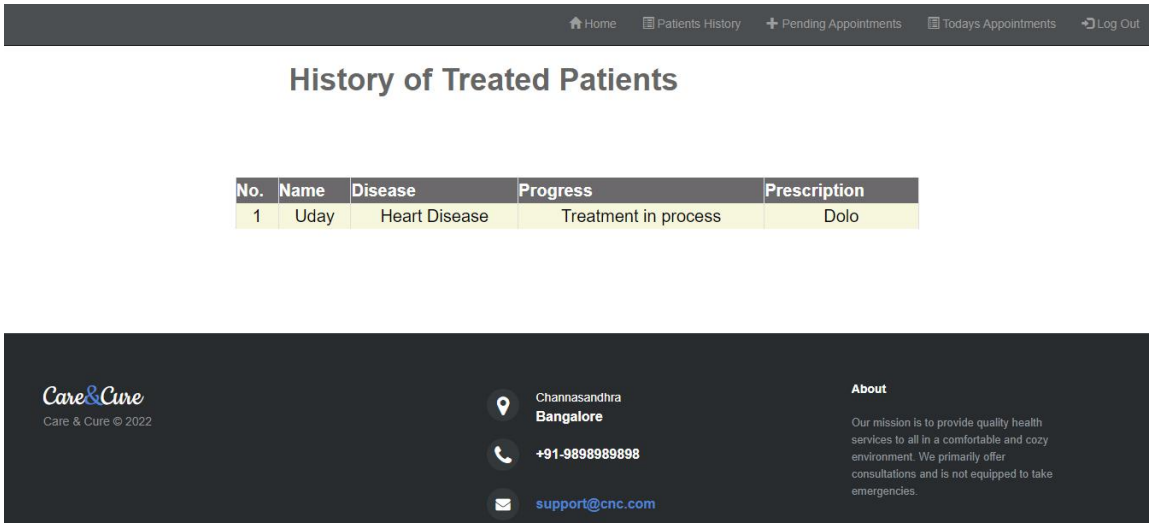


Fig 4.8.1 Doctor Home

4.10 Pending Appointments: On this page the doctor can view, accept or deny the appointment requests by the patients.

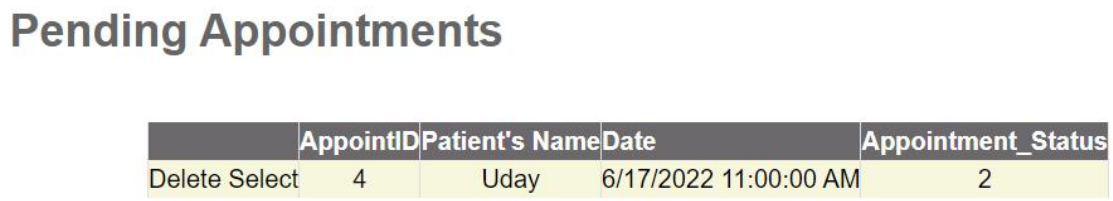


Fig 4.10.1 Pending Appointments

4.11 Today's Appointments: This page shows accepted appointments.

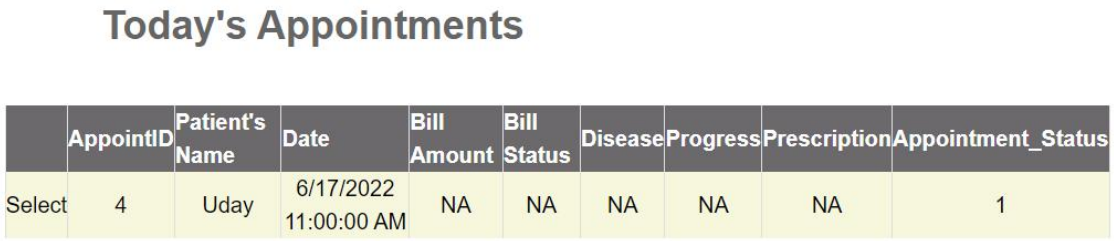


Fig 4.11.1 Today's Appointments

CHAPTER 05

5. CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Care and Cure is a user-friendly website where the patients can search for their desired doctor and book an appointment on their desired time slot and doctors can view and accept or deny pending appointment requests and write prescriptions for the accepted appointments after examining the patient. Admin can view the complete details of all the patients, doctors and other staff of the hospital and add or delete the existing staff members and doctors. It helps to decrease the workload of the user. The website is more descriptive and simpler so the user with little or no experience can also book appointments easily.

5.2 FURTHER WORK

As a future enhancement we have decided to include a genuine payment system. Front end will be more attractive. Payment options and document checking such as ID proofs can be added. Applications can be upgraded by improving performance as per user feedback.

REFERENCES

[1] https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-controlle r?view=aspnetcore-6.0&tabs=visual-studio

[2] https://stackoverflow.com/questions/31224206/predefined-type-system-object-is-not-defined-or-imported-net-4-6

[3] https://www.c-sharpcorner.com/article/create-mvc-project-step-by-step-from-scratch-part-one/

[4] https://docs.microsoft.com/enus/aspnet/mvc/overview/getting-started/introduction/ad ding-a-view