**CHAPTER 1**

**INTRODUCTION**

**1.1 Abstract**

The purpose of the project entitled as **“Hospital Management System”** is to computerize the Front Office Management of Hospital to develop website which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Patil can be register and login using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

**1.2 Existing System**

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

**1.3 Scope of system**

The system will be used as the application that serves hospitals, clinic, dispensaries or other health institutions. The intention of the system is to increase the number of patients that can be treated and managed properly.

If the Hospital Management System system is file based, management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also, could be misplaced by losing data and information.

* Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.
* Bills are generated by recording price for each facility provided to patient on a separate sheet and at last they all are summed up.
* Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
* Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

**1.4 Detail Description of Technology used:**

* **HTML (5.0)**
* **Why to learn HTML?**

HTML stands for Hyper Text Mark-up Language. HTML is the standard mark up language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

* **Advantages of learning HTML:**

HTML helps to build structure of a website and is a widely used Mark-up language. It is easy to learn. Every browser supports HTML Language. HTML is light weighted and fast to load. Storage of big files is allowed because of the application cache feature. Do not get to purchase any extra software because it’s by default in every window. Loose syntax (although, being too flexible won’t suit standards). HTML is simple to edit as being a plain text. It integrates easily with other languages such as JavaScript, CSS etc.HTML is that it is easy to code even for novice programmers. HTML also allows the utilization of templates, which makes designing a webpage easy. It is fast to download as the text is compressible. Very useful for beginners in the web designing field. HTML can be supported to each and every browser, if not supported to all the browsers. HTML is built on almost every website, if not all websites. HTML is increasingly used for data storage as like XML syntax. HTML has much tag and attributes which can short your line of code.

* **CSS (3.0)**
* **Why to learn CSS?**

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.

* **Advantages of learning CSS:**

Faster Page Speed. More code means slower page speed. Better User Experience. CSS not only makes web pages easy on the eye, it also allows for user-friendly formatting. Quicker Development Time. Easy Formatting Changes Compatibility across Devices. The Script offer consistent platform independence and can support latest browsers as well. To make a global change, simply change the style, and all elements in all the web pages will be updated automatically. You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

* **Bootstrap (5.0):**

Bootstrap is a free and open source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript based design templates for typography, forms, button, navigation, and other interface components. Bootstrap also comes with several JavaScript components which do not require other libraries like JQuery. They provide additional user interface elements such as dialog boxes, toolkit progress bars, navigation drop-downs, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

* **JavaScript (2022):**
* **Why to learn JavaScript?**

It's the most popular programming language. It's in your browser. JavaScript also exists outside of the internet. JavaScript is Ideal for Newbies. JavaScript is Easy to Learn. You Can Create Visual Effects and Other Eye-catching. Aesthetic Features. JavaScript is Versatile.

* **Advantages of JavaScript:**

Speed, Since JavaScript is an 'interpreted' language. It reduces the time required by other programming languages like Java for compilation. Simplicity, JavaScript is easy to understand and learn. Popularity Interoperability, Server Load, Rich Interfaces, Extended Functionality, Versatility.

* **Python (3.11):**

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn’t specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. Python is often used to develop the back end of a website or application—the parts that a user doesn’t see. Python’s role in web development can include sending data to and from servers, processing data and communicating with databases, URL routing, and ensuring security. Python offers several frameworks for web development. Commonly used ones include Django and Flask.

* **SQLite (3.41):**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

* **Django (4.1):**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

**1.5 Operating Environment**

**1.5.1 Hardware:**

* **Client Side-Hardware**
* RAM:2 GB
* CPU Speed:2.4 GHZ
* Hard Disk: 250 GB
* **Server Side-Hardware**
* RAM: 8 GB
* CPU Speed: 2.4 GHZ
* Hard Disk: 500 GB

**1.5.2 Software:**

* **Client Side-Software**
* Web-browser(any)
* **Server Side-Software**
* Python (3.11)
* SQLite (3.41)
* Django (4.1)

**CHAPTER 2**

**PROPOSED SYSTEM**

**2.1 Proposed System**

The **Hospital Management System** is designed for any hospital to replace their manual paper-based system. The new system is to control the information of patients as well as doctors. These services are to be provided in an efficient, cost-effective manner, with the goal of reducing the time and resources currently required for such tasks.

The complete set of rules & procedures related to Hospitals, day to day activities and generating report is called “Hospital Management System”. It is a computerized management system. This system also keeps the records of hardware assets besides software of this organization. The proposed system will keep a track of Doctors, Patients & Receptionist. This project has GUI based website that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules.

**Goals of Proposed System:**

* The system should be easy to operate.
* The working in the organization will be well planned and organized.
* The level of accuracy in the proposed system will be higher.
* The reliability of the proposed system will be high due to proper storage of information.
* Provide quick and efficient retrieval of information.
* It will reduce human efforts.
* Insertion of data will be entered with validation and without human errors.

**2.2 Objectives of Proposed System**

* To provide the registration and login page for admin, doctors and patients.
* To manage the information of patients.
* To store the details of doctors.
* To store the details of equipment’s and machines.
* To manage appointments of doctors.
* To store the maintenance details of hospital.
* To download discharge invoice.

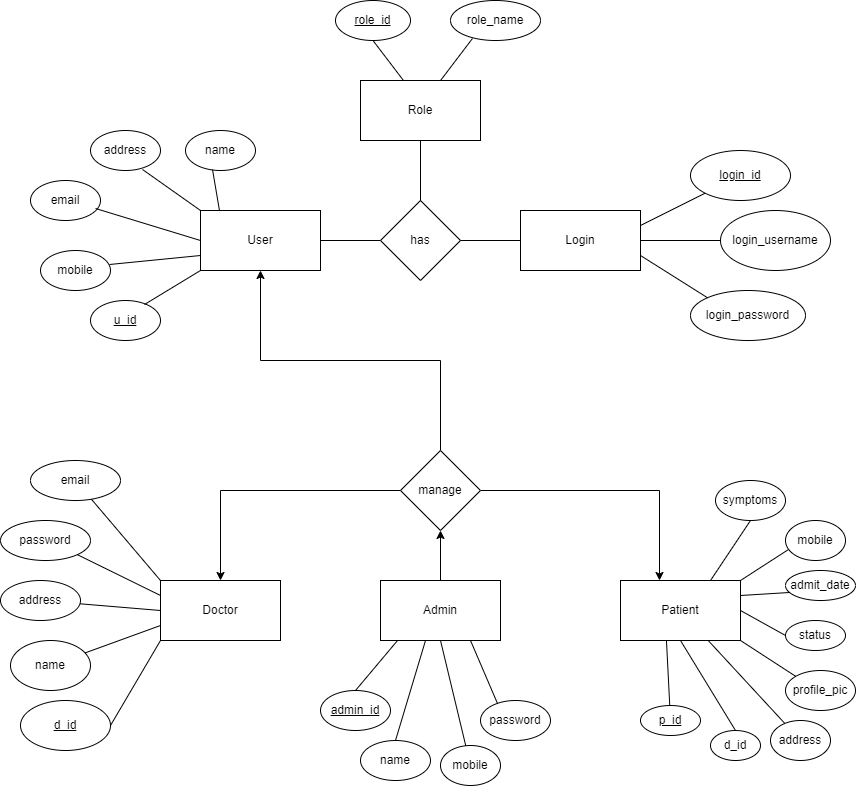
**2.3 Users Requirement**

* The system should create a login/logout page for patient or admin to access their account.
* The system should provide a profile module to create & update a customer profile.
* The system should provide book appointment module to manage appointments.

**CHAPTER 3**

**ANALYSIS AND DESIGN**

**3.1 Entity Relationship Diagram-**

****

**3.2 Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| a\_id | int | Appointment Id |
| address | varchar | Address |
| admin\_id | int | Admin Id |
| adminName | string | Admin Name |
| admitDate | date | Admit Date |
| appointmentDate | date | Appointment Date |
| d\_id | int | Doctor Id |
| d\_name | string | Doctor Name |
| daySpent | int | Day Spent |
| department | string | Department Name |
| description | string | Description |
| doctorFee | int | Doctor Fee |
| medicineCost | int | Medicine Cost |
| mobile | int | Mobile Number |
| otherCharges | int | Other Charges |
| password | varchar | Password |
| p\_id | int | Patient Id |
| p\_name | string | Patient Name |
| pd\_id | int | Patient Discharge id |
| profile\_pic | blob | Profile Picture |
| status | bool | Status |
| symptoms | string | Symptoms Name |
| totalAmount | int | Total Amount |
| releaseDate | date | Release Date |
| roomCharge | int | Room Charge |

**3.3 Table Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | | admin\_table | | | |
| **Primary Key** | | admin\_id | | | |
| **Foreign Key** | | - | | | |
| **Description of table** | | Information about Admin details | | | |
| **Sr.No** | **Field Name** | | **Datatype with size** | **Constraints** | **Description** |
| 1 | admin\_id | | int(3) | Primary Key | Admin Id |
| 2 | adminName | | string(20) | Not Null | Admin Name |
| 3 | mobile | | int(13) | Not Null | Admin Mobile |
| 4 | password | | varchar(20) | Not Null | Admin Password |

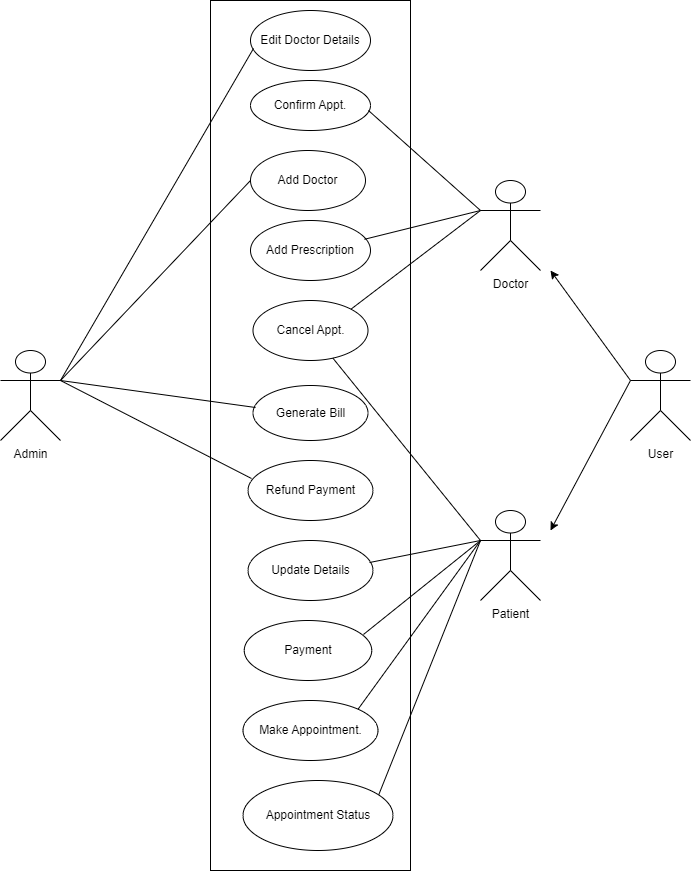
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | | Patient\_table | | | |
| **Primary Key** | | p\_id | | | |
| **Foreign Key** | | d\_id | | | |
| **Description of table** | | Information about Patient details | | | |
| **Sr.No** | **Field Name** | | **Datatype with size** | **Constraints** | **Description** |
| 1 | p\_id | | int(3) | Primary Key | Patient Id |
| 2 | assigned\_doctor\_id | | int(3) | Foreign key | Doctor Id |
| 3 | address | | varchar(25) | Not Null | Patient Address |
| 4 | mobile | | int(13) | Not Null | Patient Mobile |
| 5 | symptoms | | string(25) | Not Null | Patient Disease |
| 6 | status | | bool | Not Null | Patient Status |
| 7 | admitDate | | date | Not Null | Patient Admit Date |
| 8 | profile\_pic | | blob | Not Null | Patient Profile Picutre |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | | doctor\_table | | | |
| **Primary Key** | | d\_id | | | |
| **Foreign Key** | | - | | | |
| **Description of table** | | Information about Doctor details | | | |
| **Sr.No** | **Field Name** | | **Datatype with size** | **Constraints** | **Description** |
| 1 | d\_id | | int(3) | Primary Key | Doctor Id |
| 2 | address | | varchar(25) | Not Null | Doctor Address |
| 3 | department | | varchar(15) | Not Null | Doctor Department |
| 4 | status | | bool | Not Null | Doctor Status |
| 5 | profile\_pic | | blob | Not Null | Doctor Profile Picture |
| 6 | mobile | | int(13) | Not Null | Doctor Password |

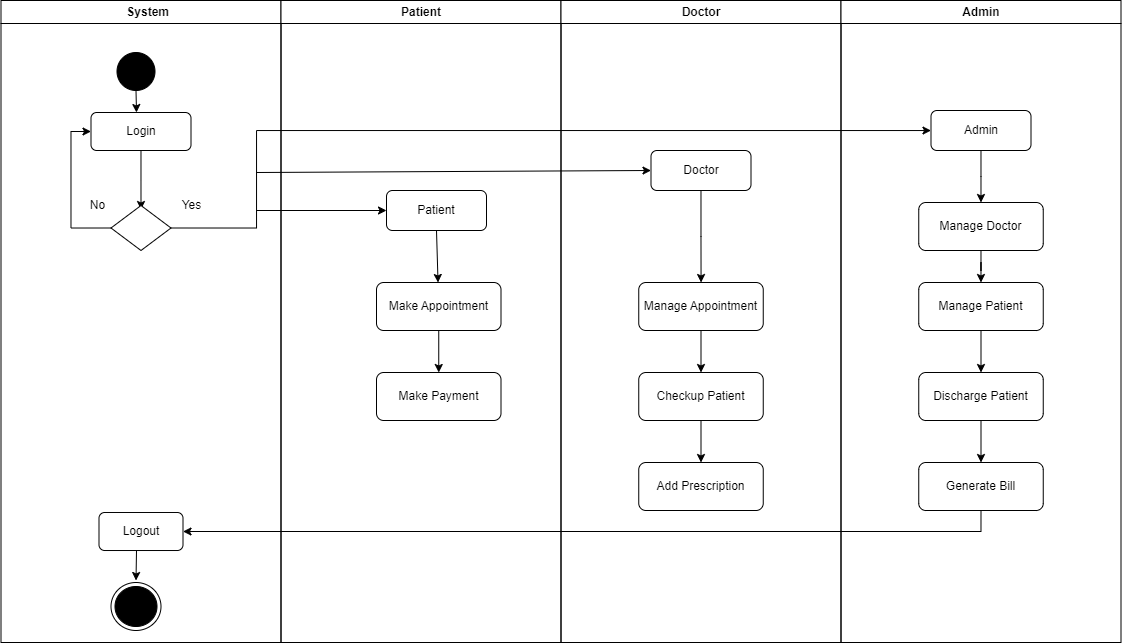
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | | patient\_discharge\_table | | | |
| **Primary Key** | | pd\_id | | | |
| **Foreign Key** | | p\_id, d\_id | | | |
| **Description of table** | | Information about Patient discharge details | | | |
| **Sr.No** | **Field Name** | | **Datatype with size** | **Constraints** | **Description** |
| 1 | pd\_id | | int(3) | Primary Key | Patient Discharge Id |
| 2 | p\_id | | int(3) | Foreign Key | Patient Id |
| 3 | p\_name | | varchar(50) | Not Null | Patient Name |
| 4 | assign\_d\_id | | int(3) | Foreign Key | Assign Doctor Name |
| 5 | address | | varchar(25) | Not Null | Address Details |
| 6 | mobile | | int(13) | Not Null | Mobile Number |
| 7 | symptoms | | varchar(25) | Not Null | Symptoms Name |
| 8 | admitDate | | date | Not Null | Admit Date |
| 9 | releaseDate | | date | Not Null | Release Date |
| 10 | daySpent | | int(2) | Not Null | Day Spent |
| 11 | roomCharge | | int(10) | Not Null | Room Charge |
| 12 | medicineCost | | int(10) | Not Null | Medicine Cost |
| 13 | doctorFee | | int(10) | Not Null | Doctor Fee |
| 14 | otherCharges | | int(10) | Not Null | Other Charges |
| 15 | totalAmount | | int(15) | Not Null | Total Bill |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | | appointment\_table | | | |
| **Primary Key** | | a\_id | | | |
| **Foreign Key** | | d\_id, p\_id | | | |
| **Description of table** | | Information about Appointment details | | | |
| **Sr.No** | **Field Name** | | **Datatype with size** | **Constraints** | **Description** |
| 1 | a\_id | | int(3) | Primary Key | Admin Id |
| 2 | d\_id | | int(3) | Foreign Key | Doctor Id |
| 3 | d\_name | | string(25) | Not Null | Doctor Name |
| 4 | p\_id | | Int(3) | Foregn Key | Patient Id |
| 5 | p\_name | | string | Not Null | Patient Name |
| 6 | appointmentDate | | date | Not Null | Appointment Date |
| 7 | description | | string | Not Null | Description |
| 8 | status | | bool | Not Null | Status |

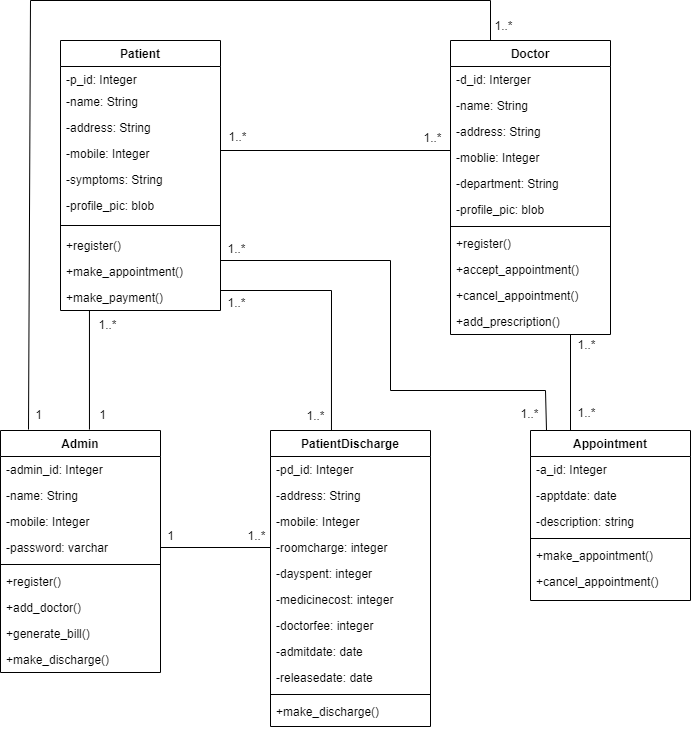
**3.4 Use Case Diagram-**

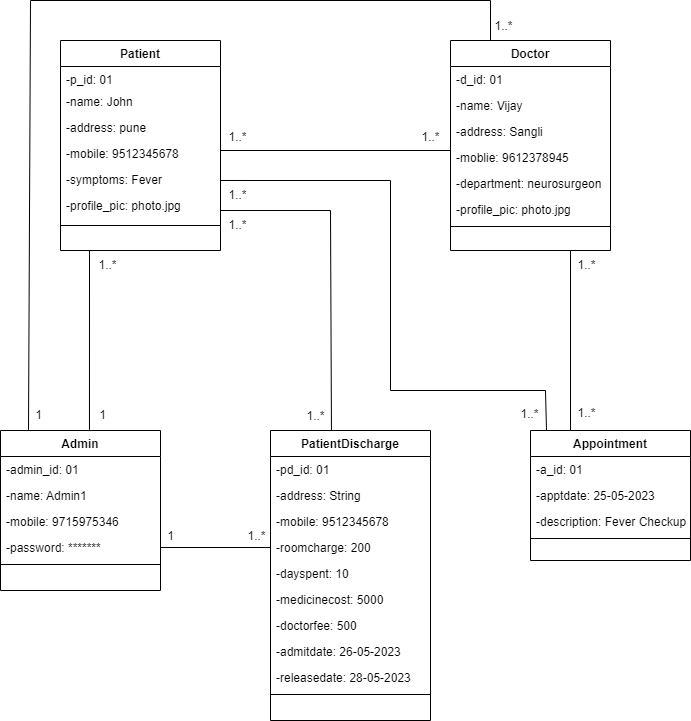
****

**3.5 Activity Diagram-**

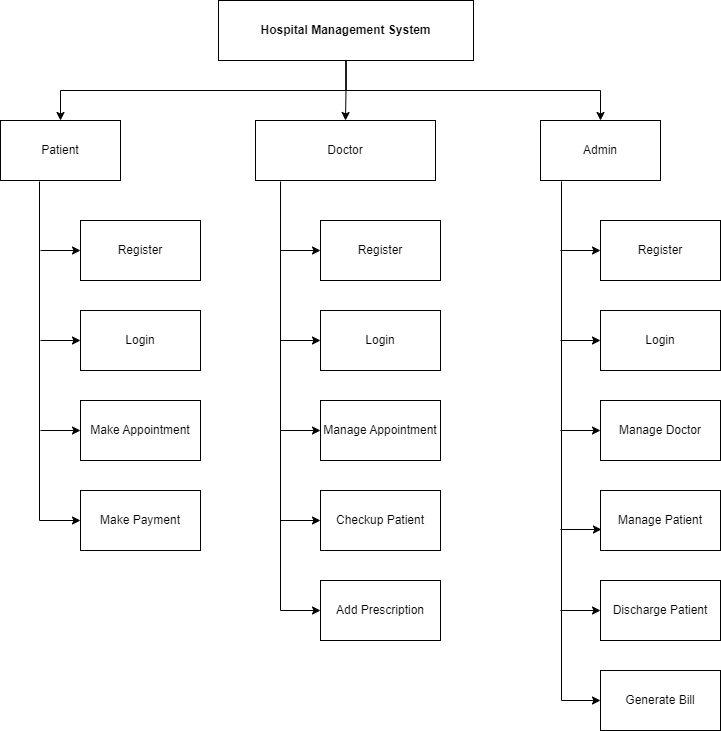
****

**3.7 Class Diagram –**

**3.8 Object Diagram –**

****

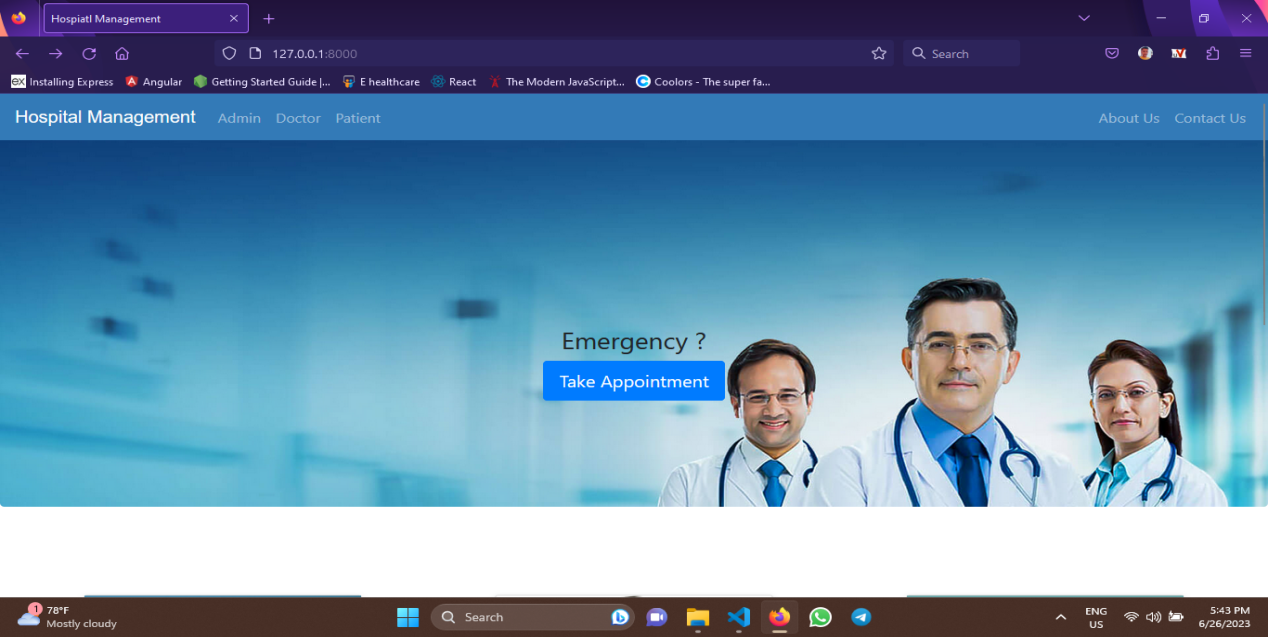
**3.9 Module Hierarchy Diagram-**

****

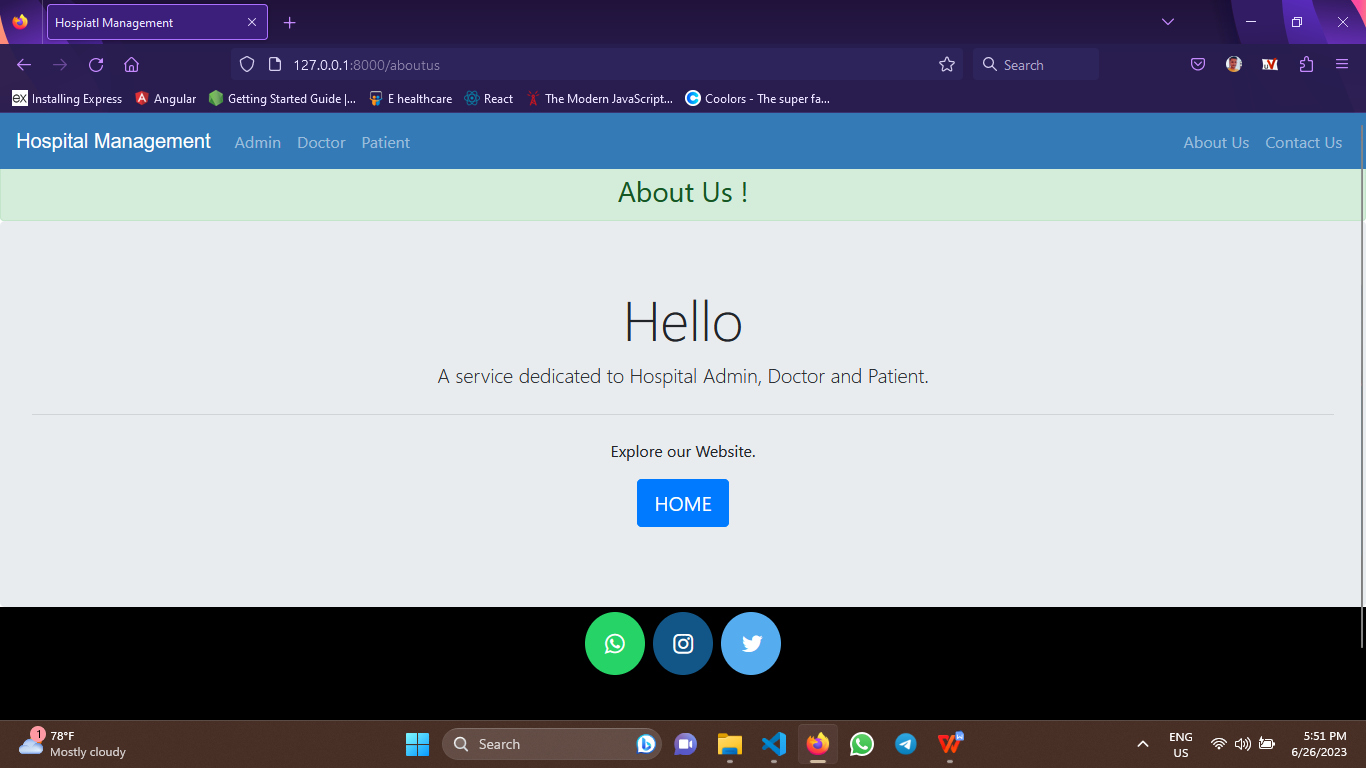
**CHAPTER 4**

**USER MANUAL**

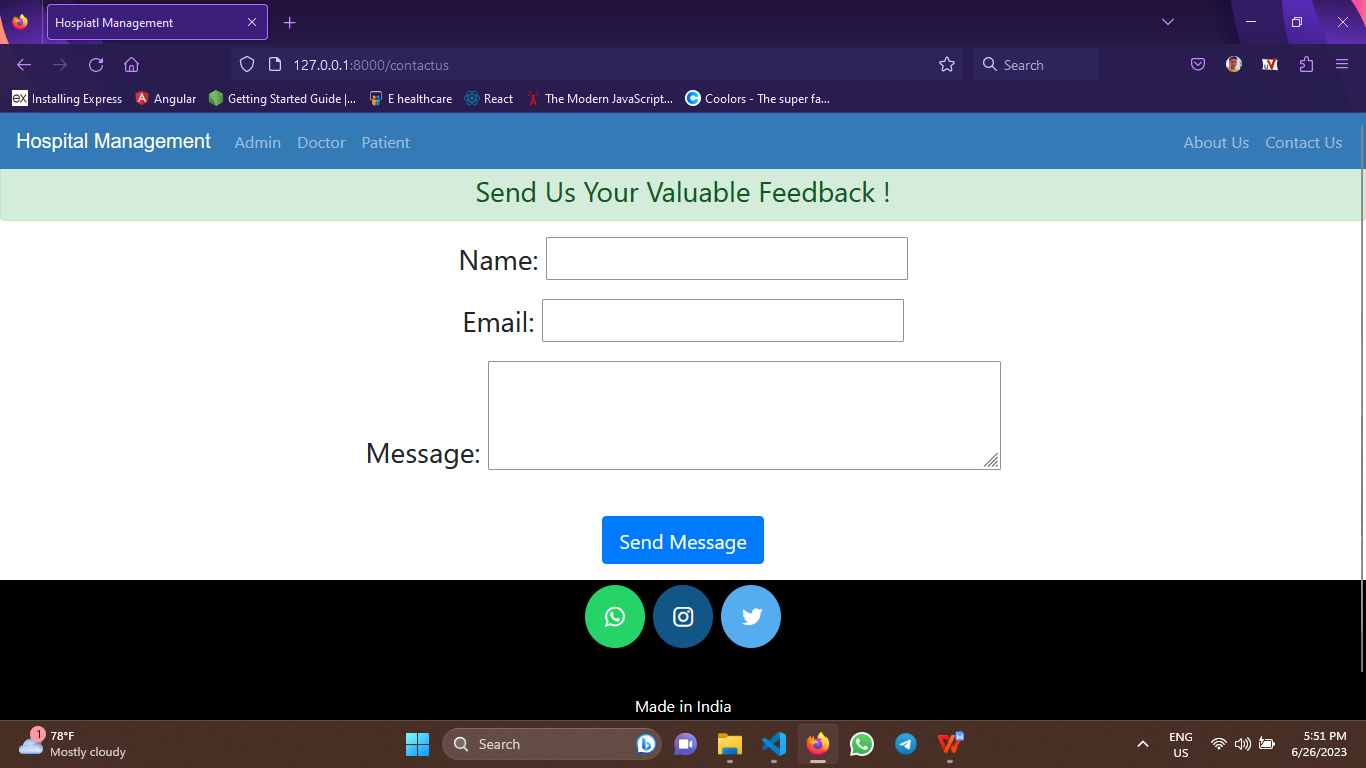
* 1. **User Interface Design (Screens):**
     1. **Home Page -**

****

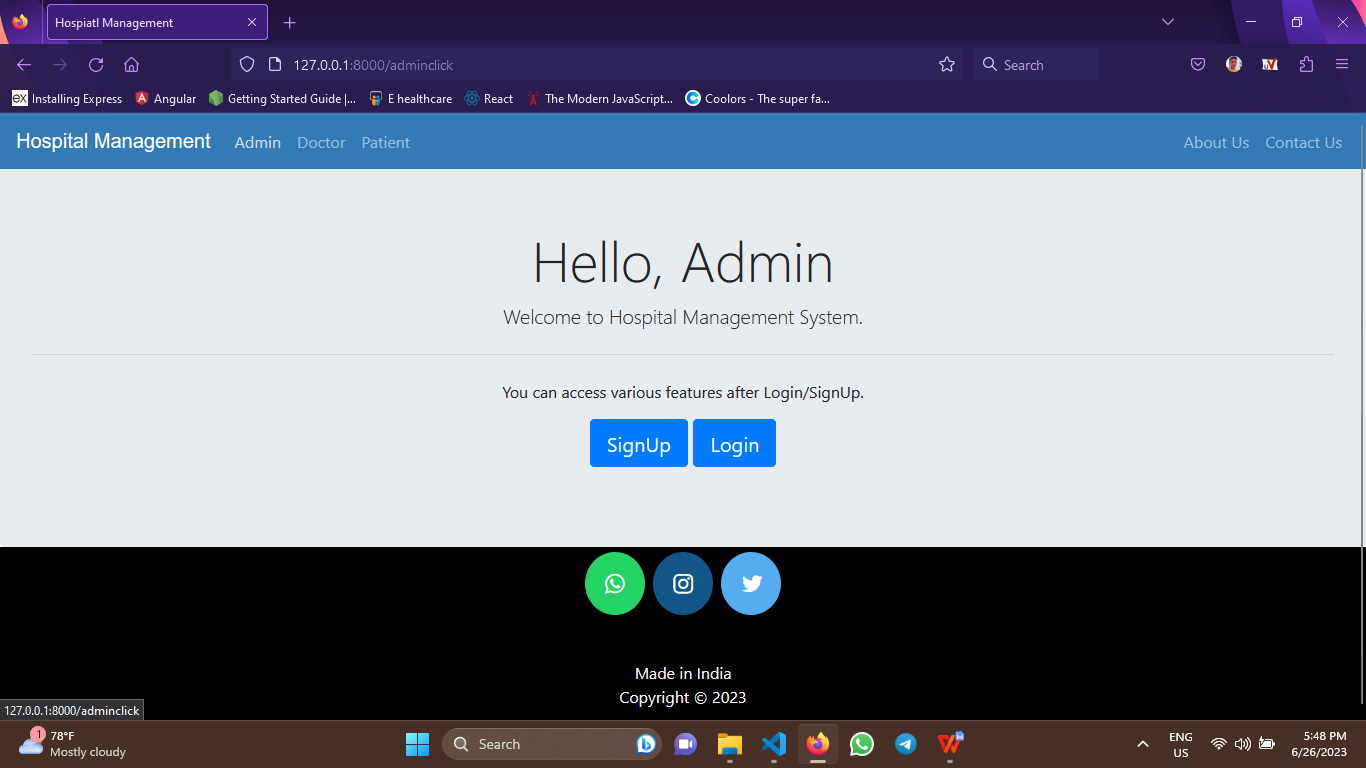
* + 1. **About us -**

****

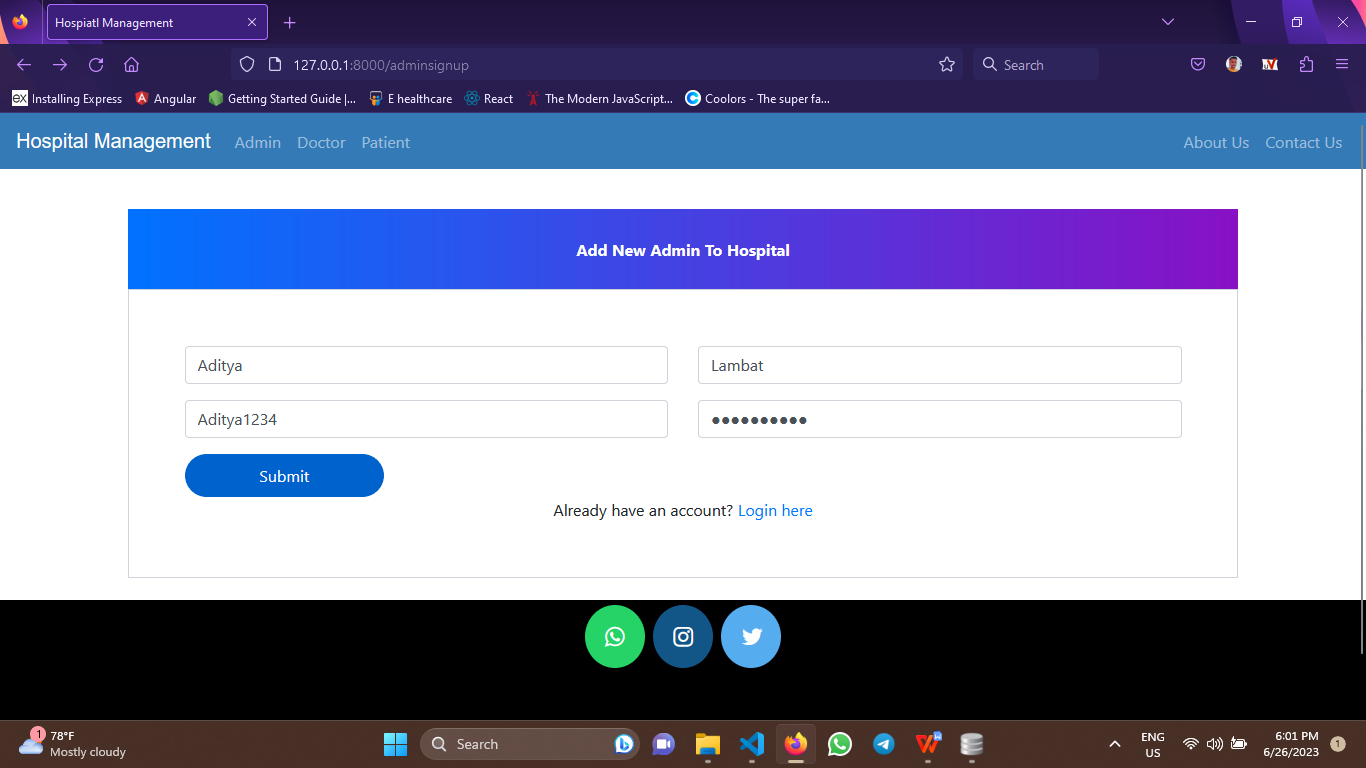
**Contact us -**

****

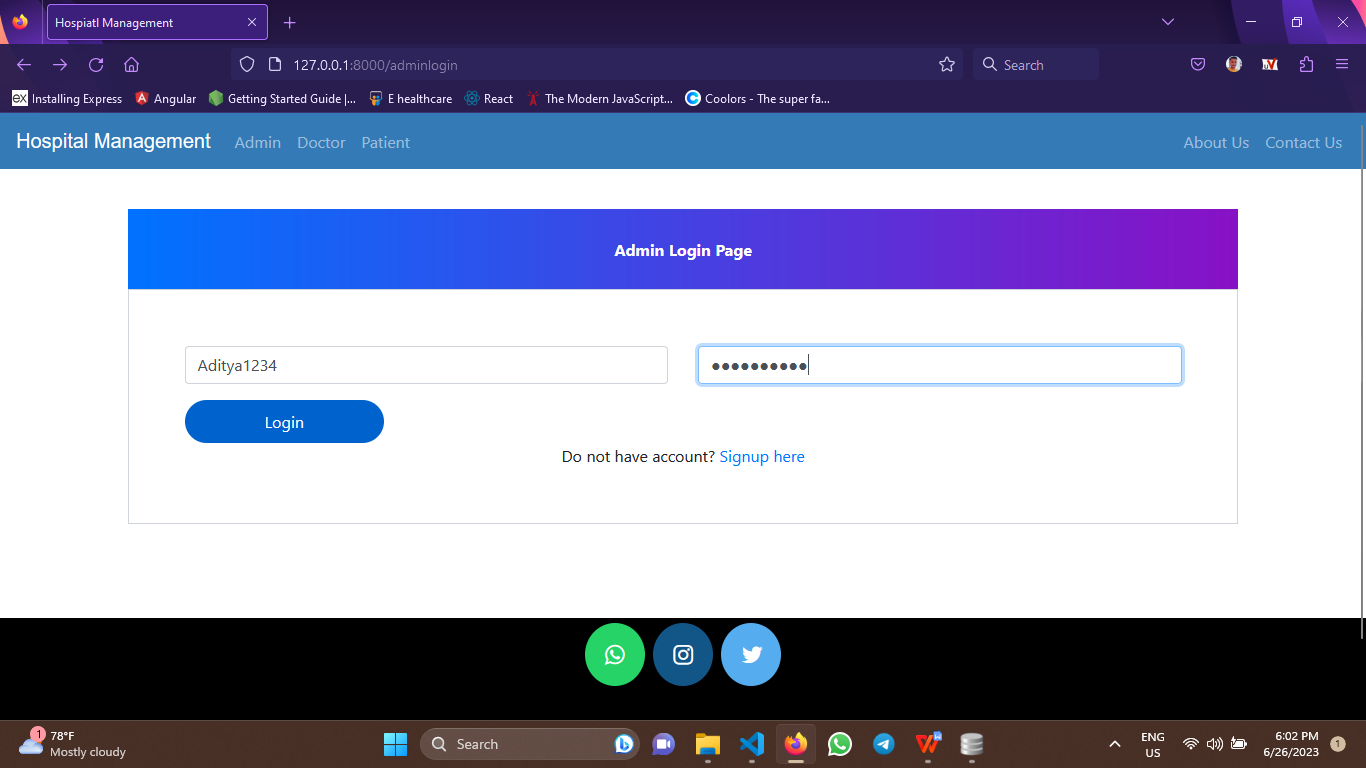
* + 1. **Admin -**

****

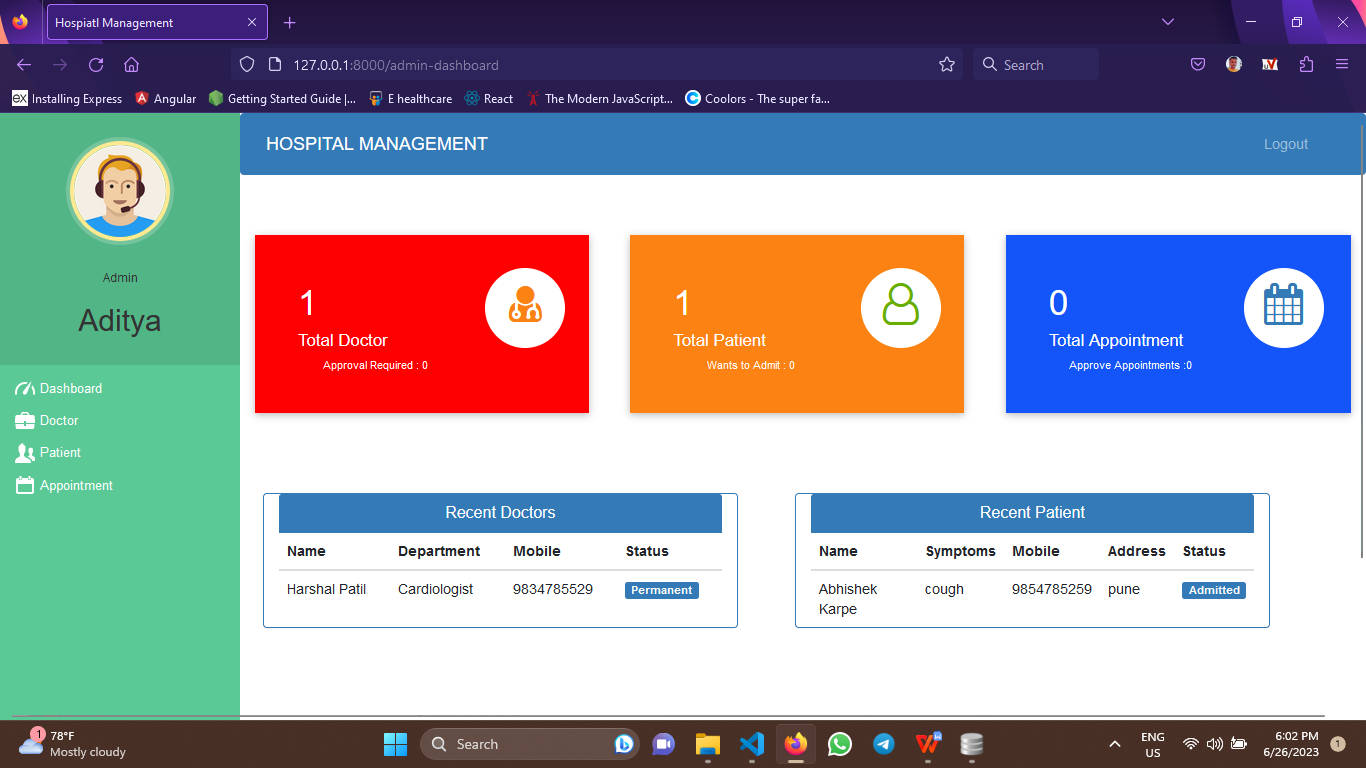
* + - 1. **Admin Sign up-**

****

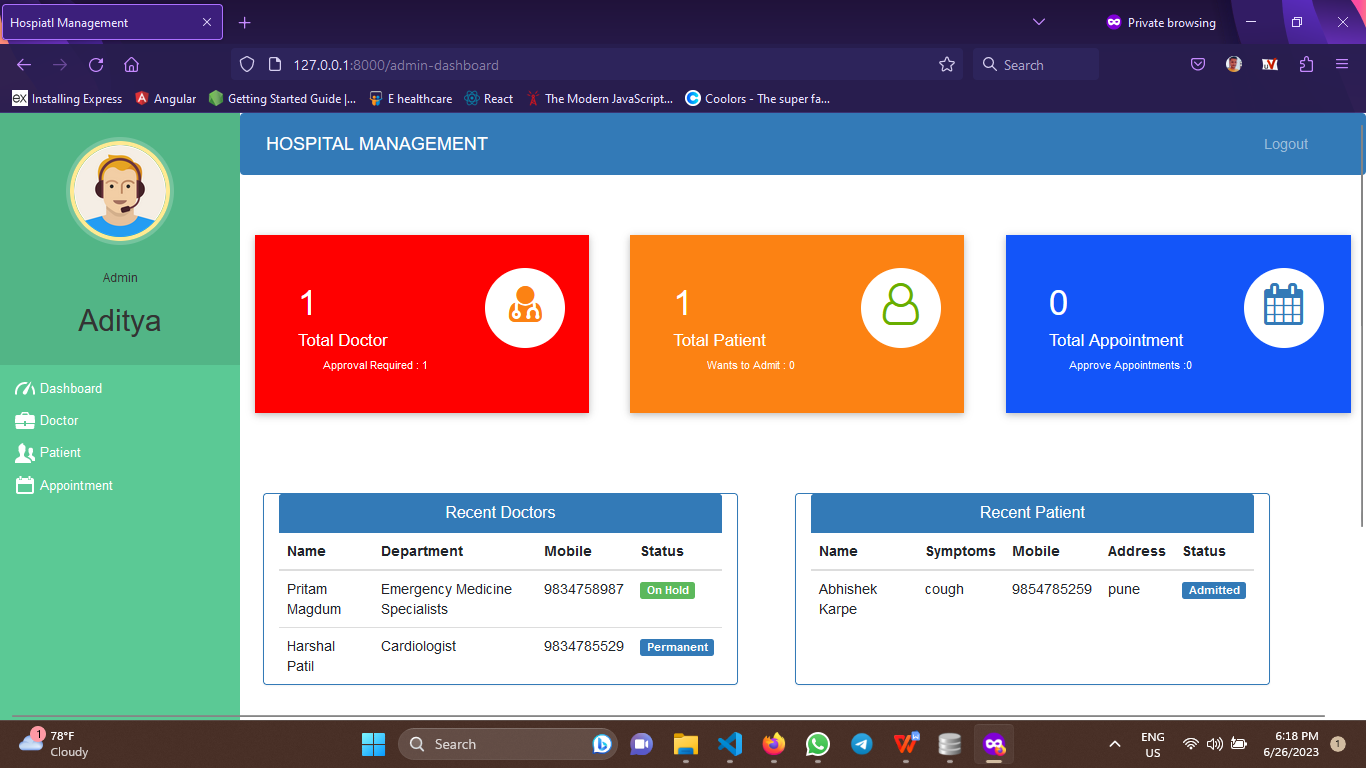
* + - 1. **Admin Login -**

****

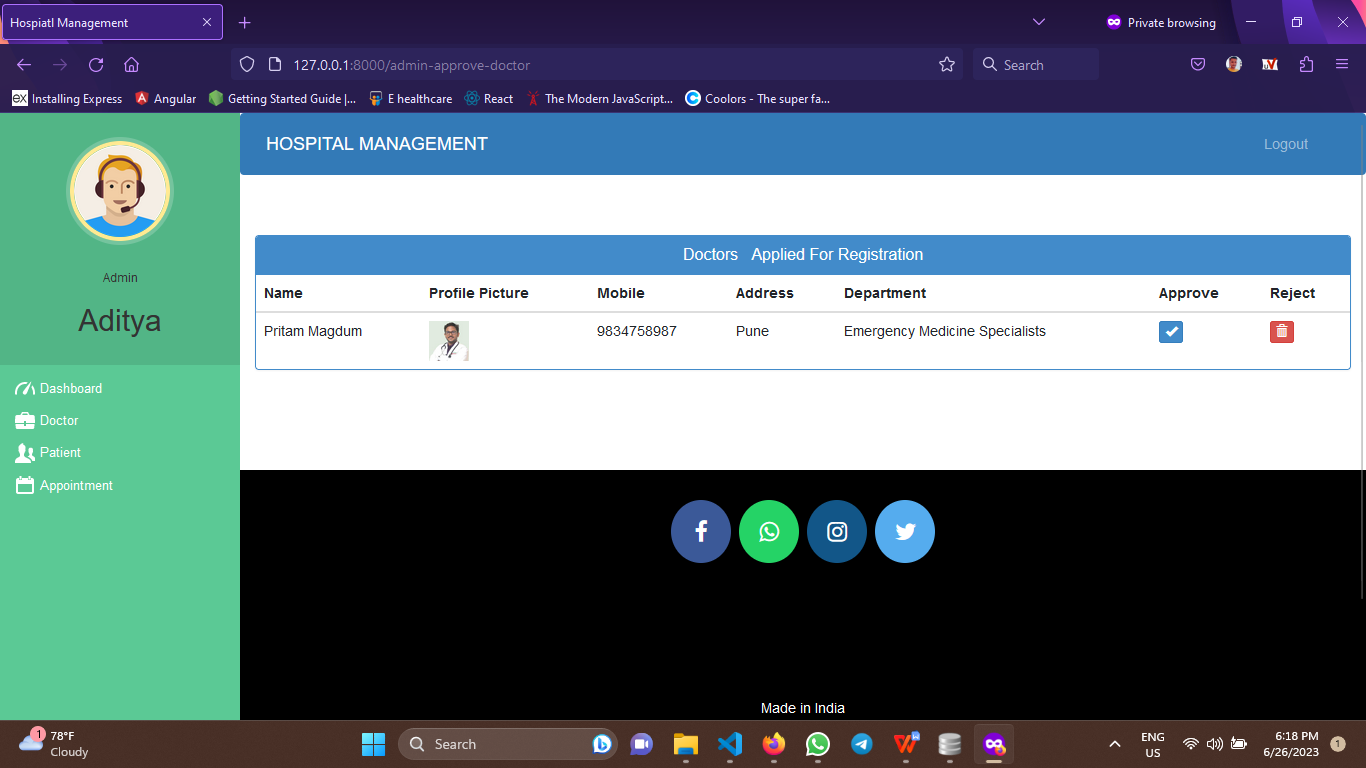
* + - 1. **Admin Dashboard -**

****

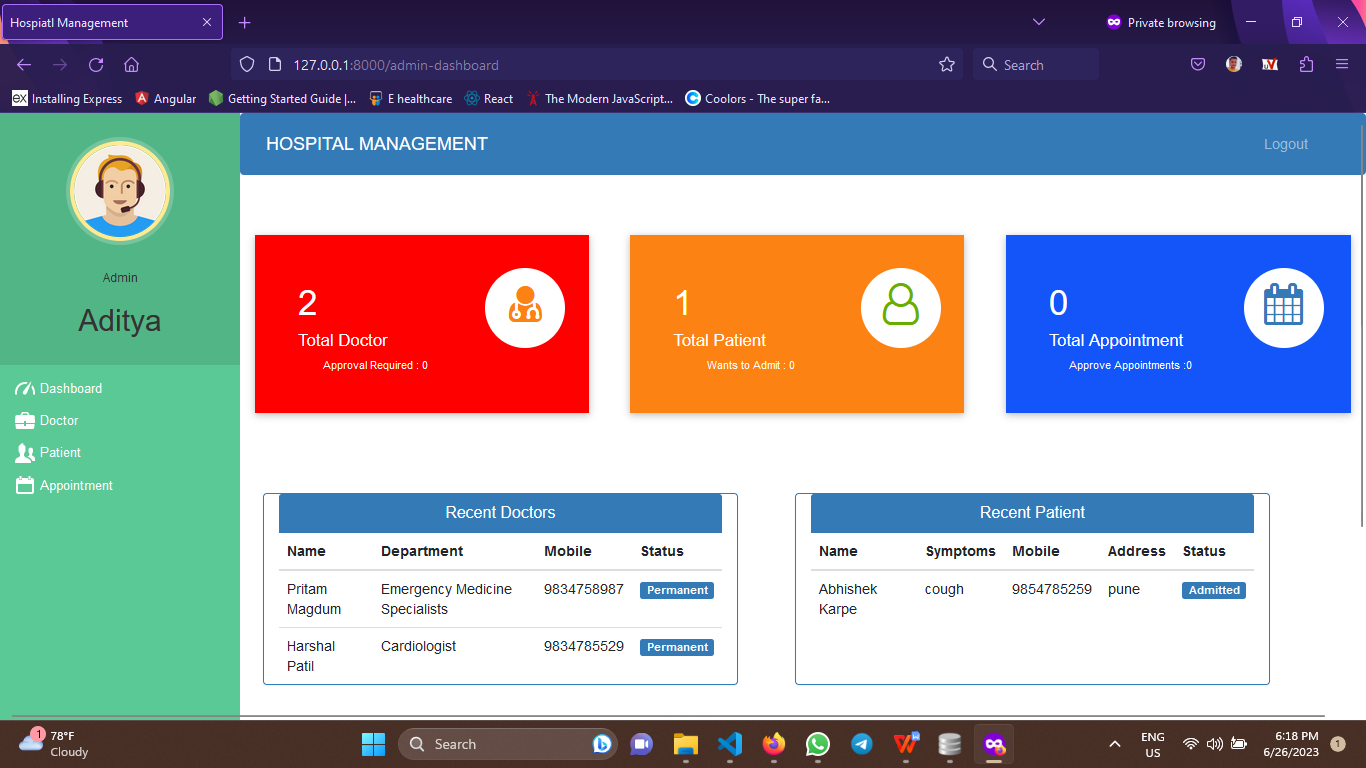
* + - 1. **Admin Dashboard Pending Request of Doctor -**

****

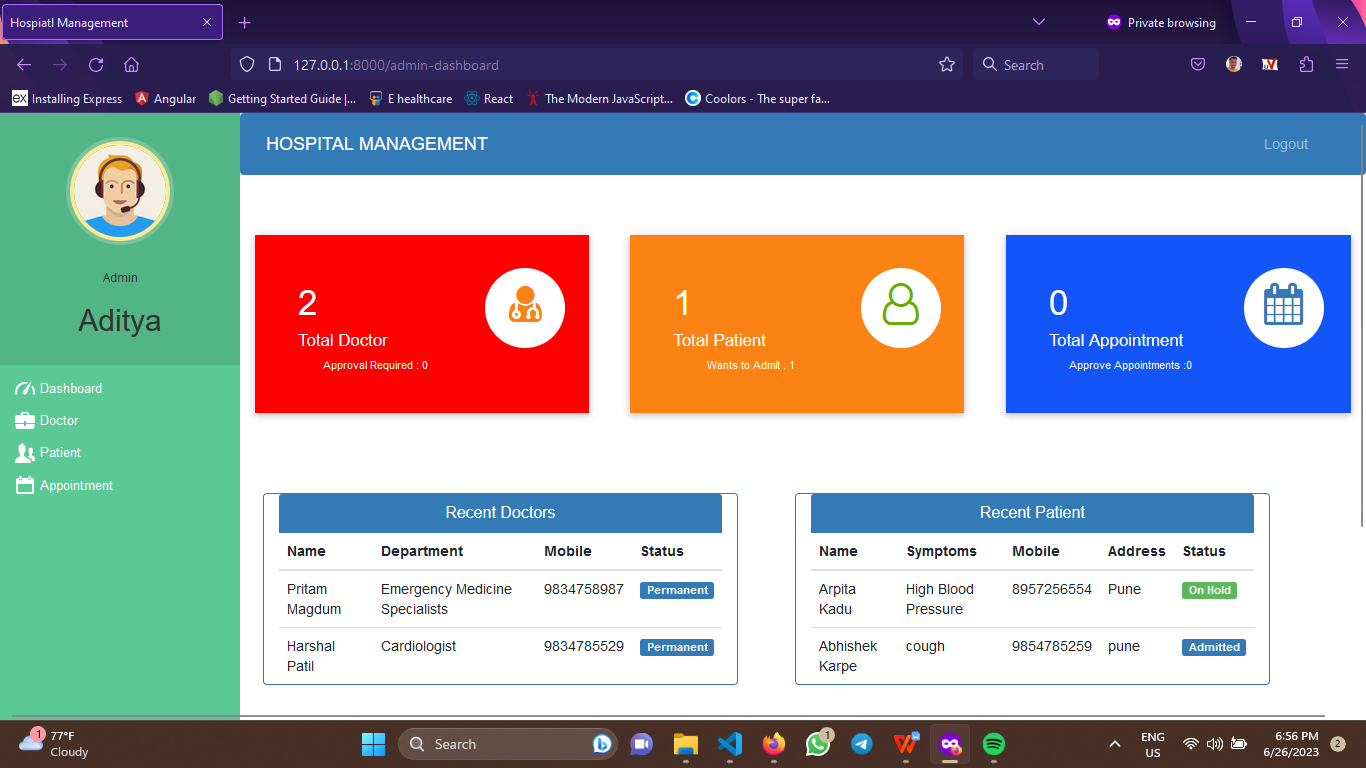
* + - 1. **Admin Doctor Approval -**

****

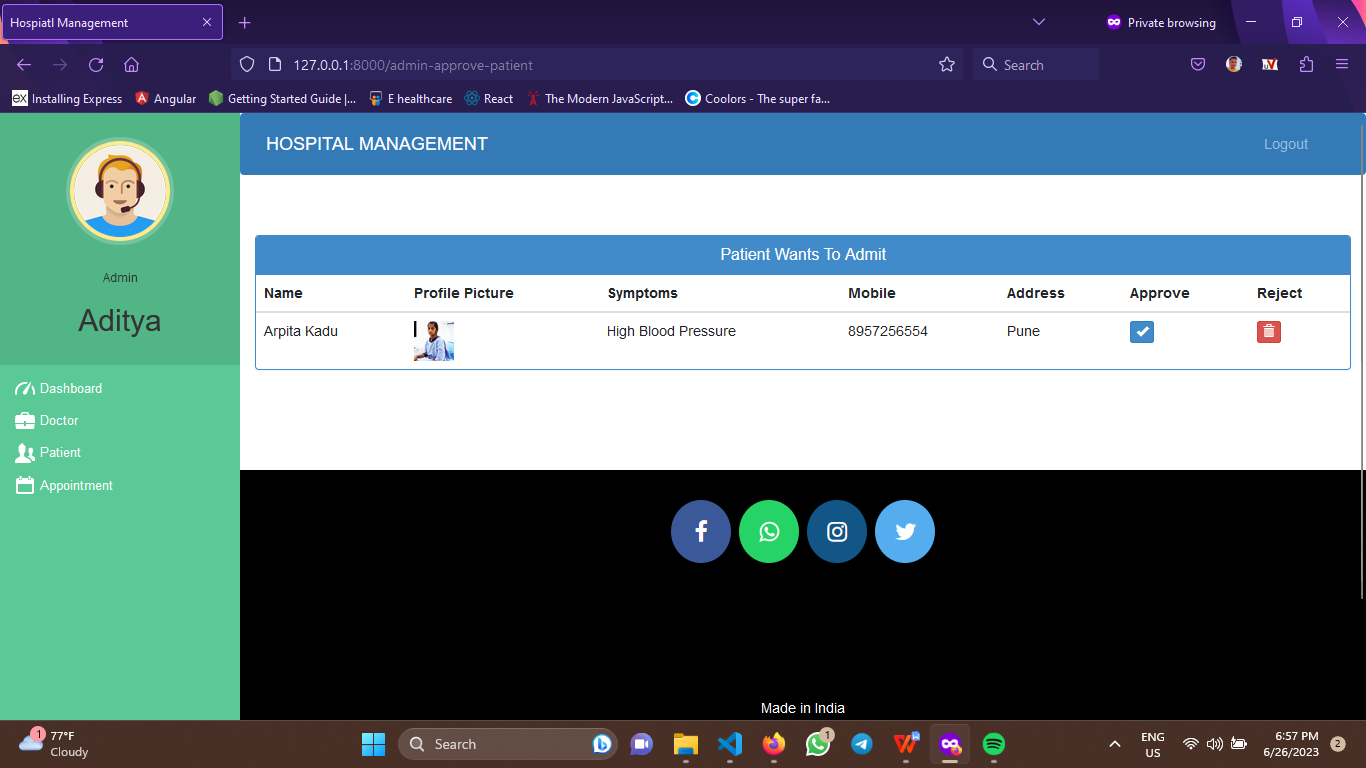
* + - 1. **Admin Dashboard After Doctor Request Approval -**

****

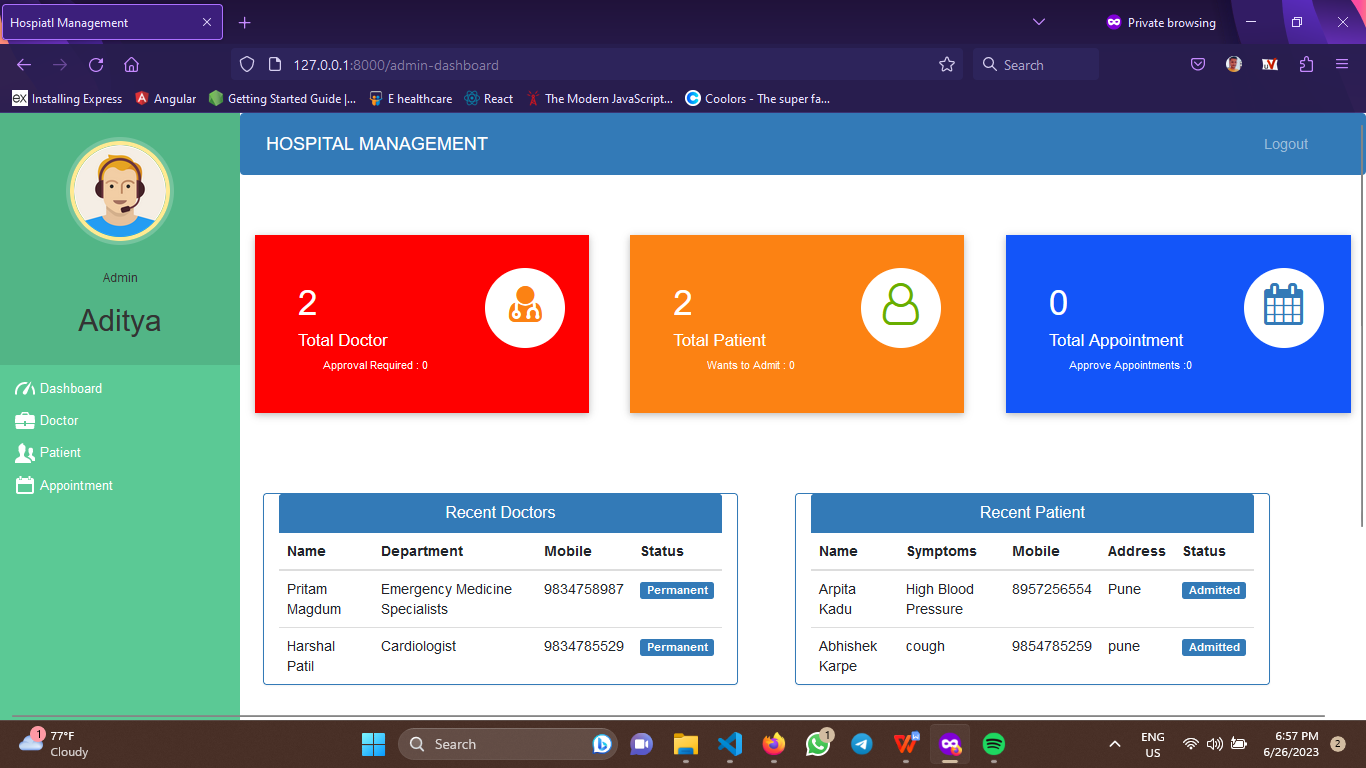
* + - 1. **Admin Dashboard Pending Request of Patient -**

****

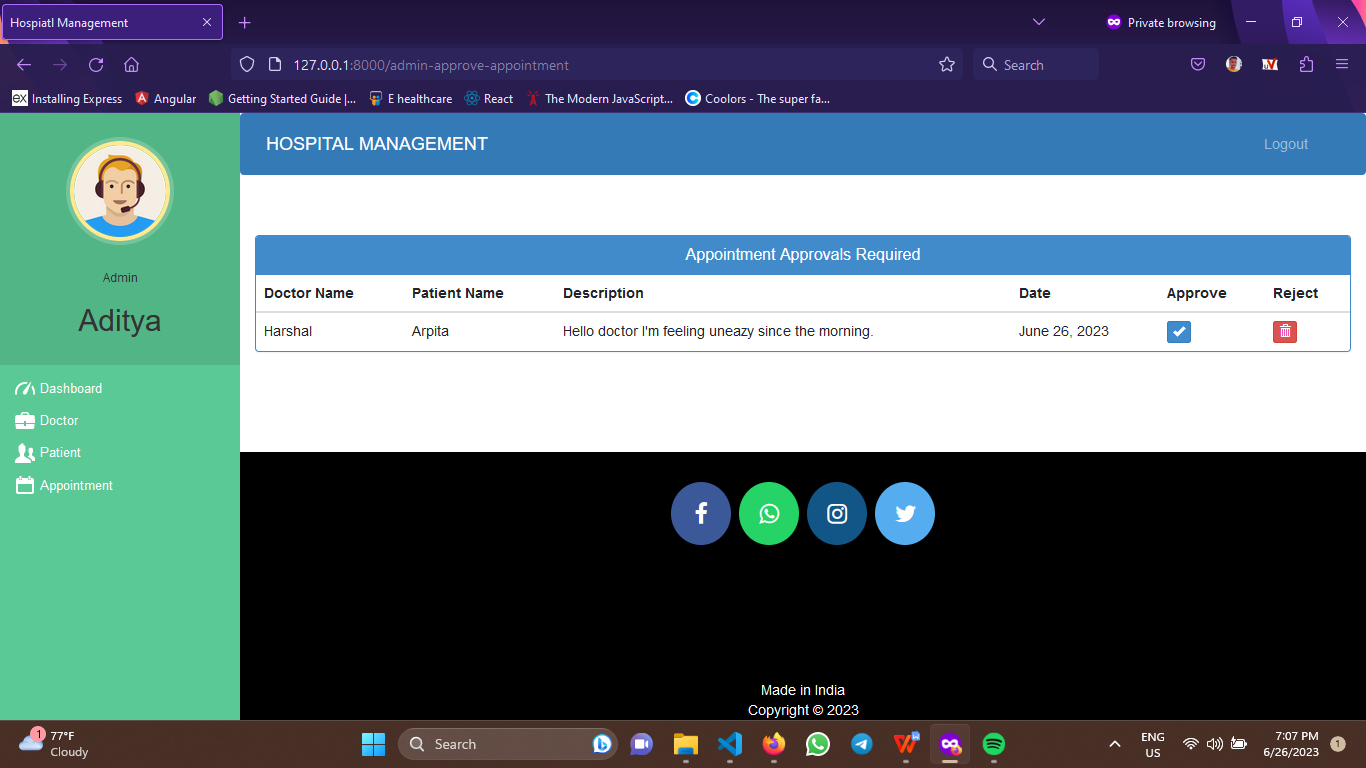
* + - 1. **Admin Patient Approval -**

****

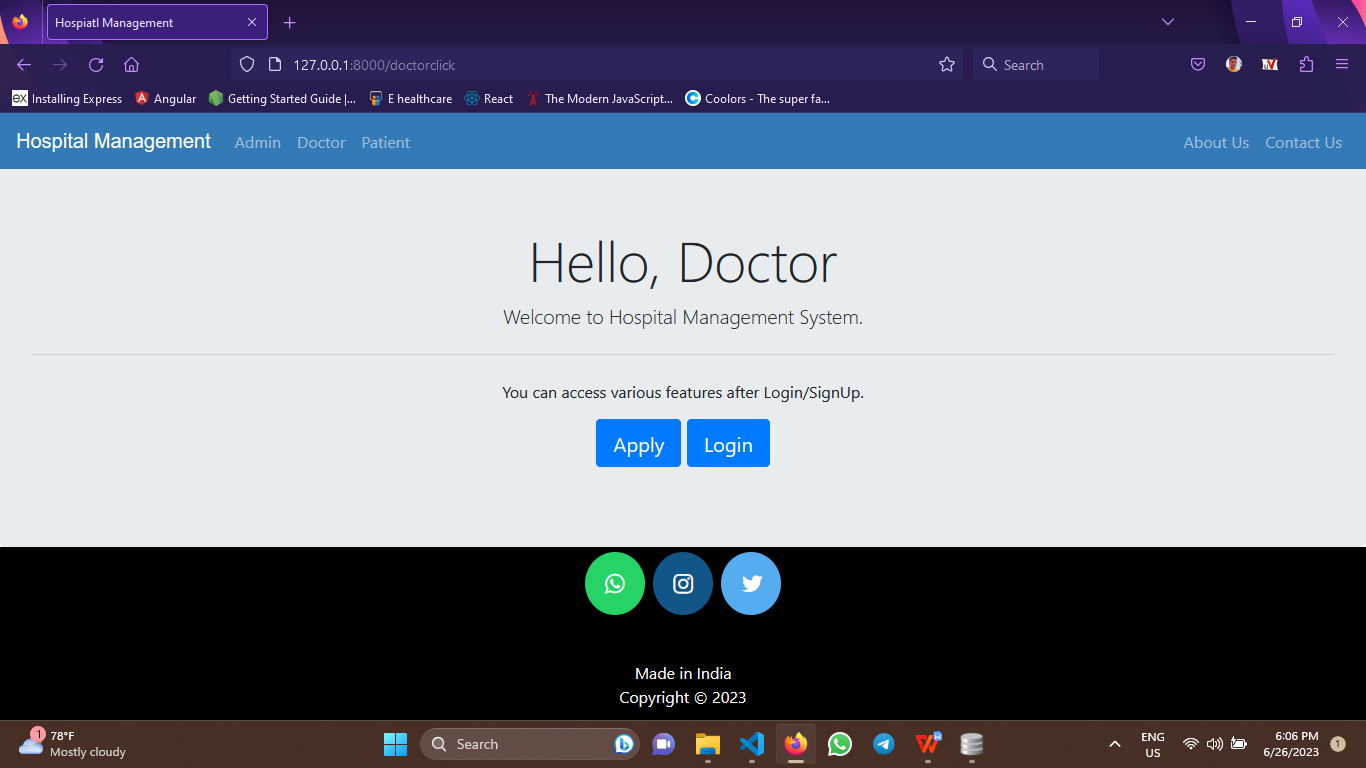
* + - 1. **Admin Dashboard After Patient Request Approval -**

****

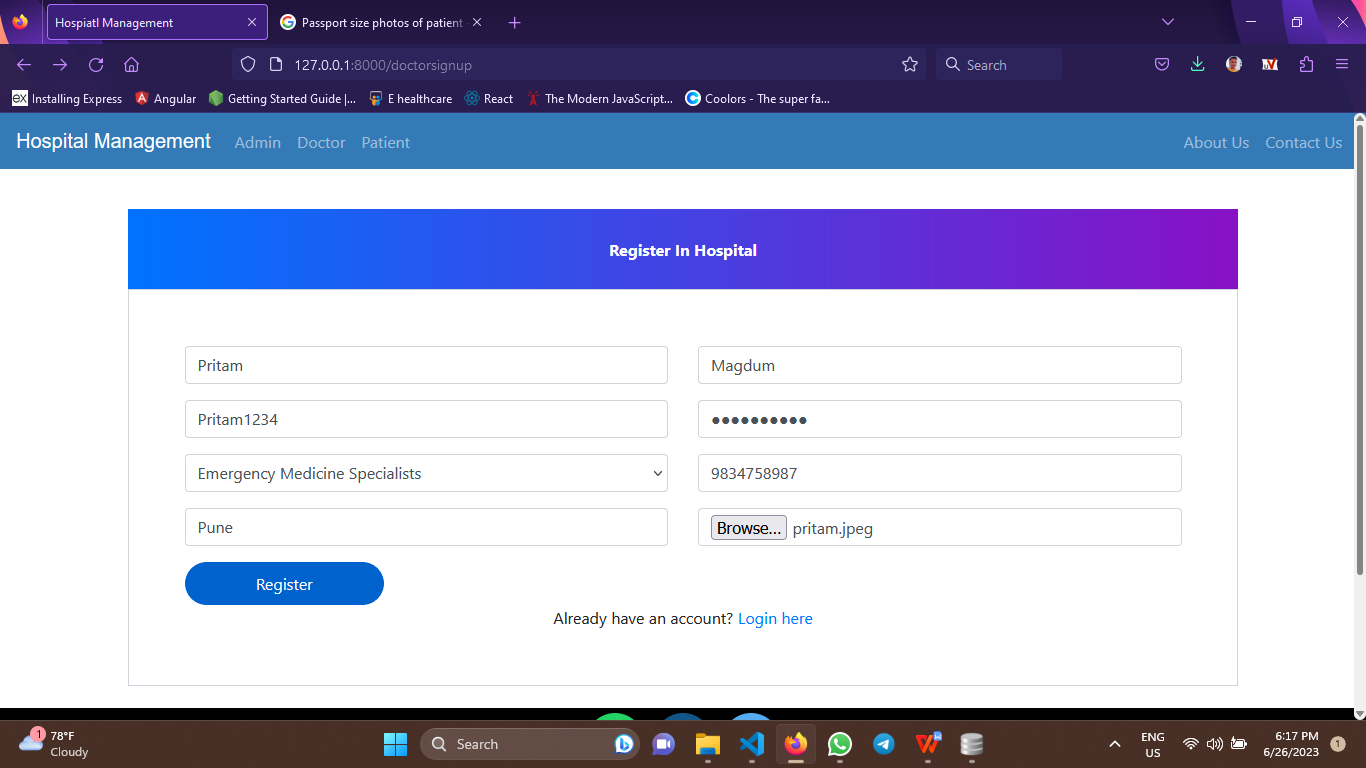
* + - 1. **Admin Appointment Approval -**

****

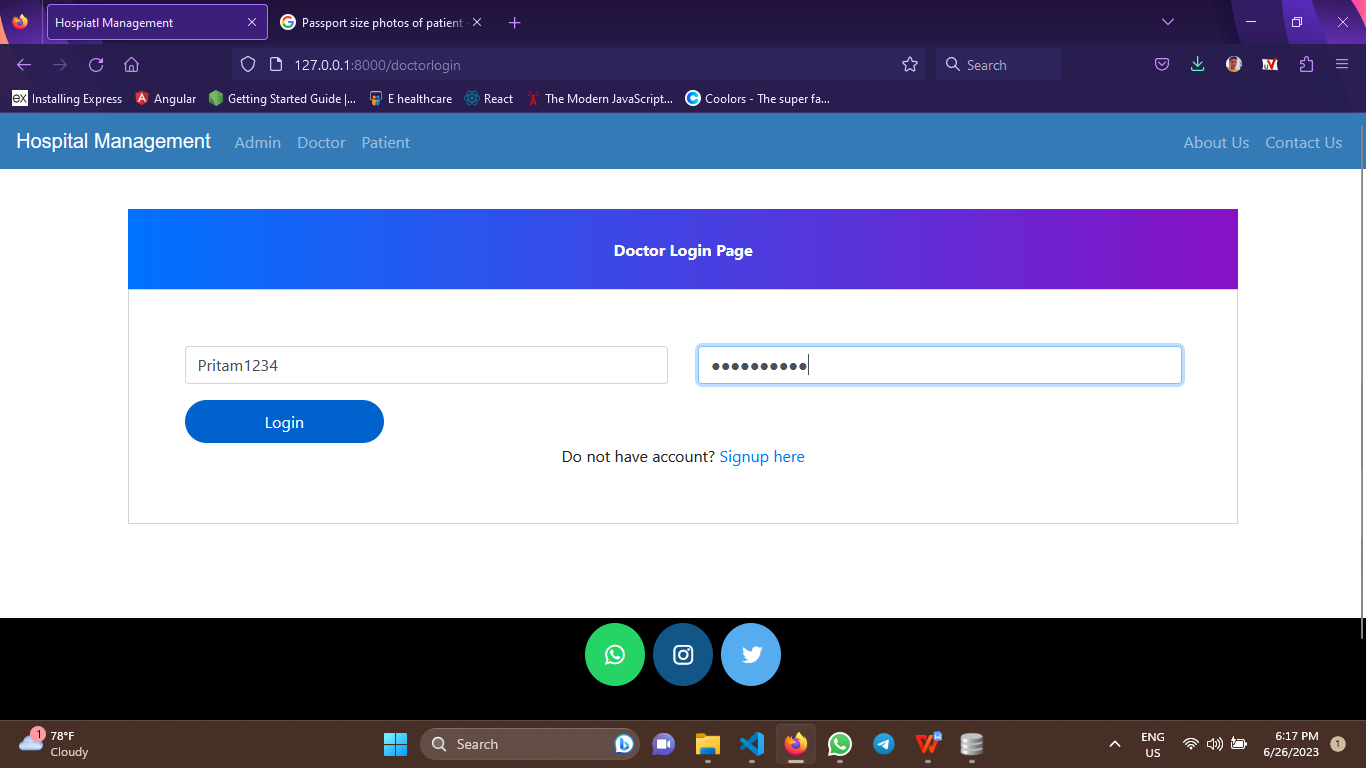
* + 1. **Doctor -**



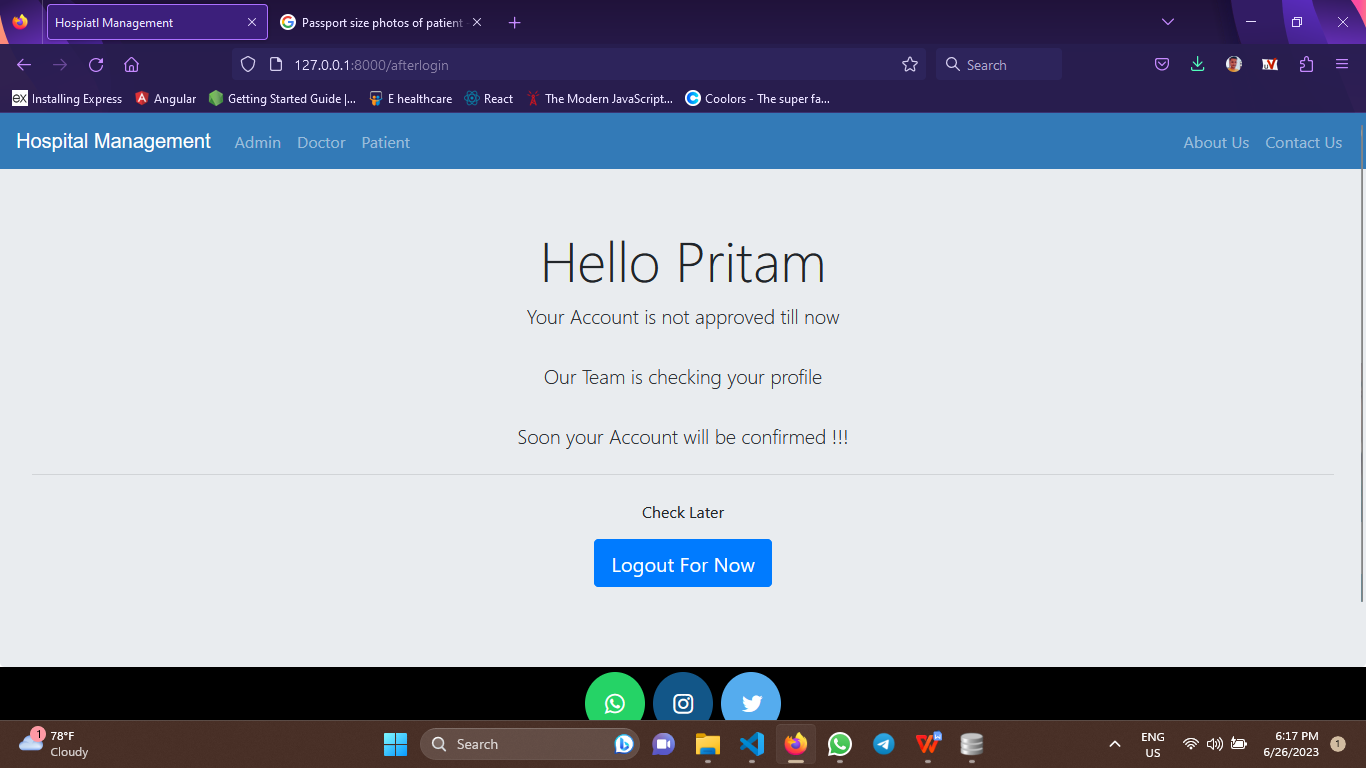
* + - 1. **Doctor Sign up -**

****

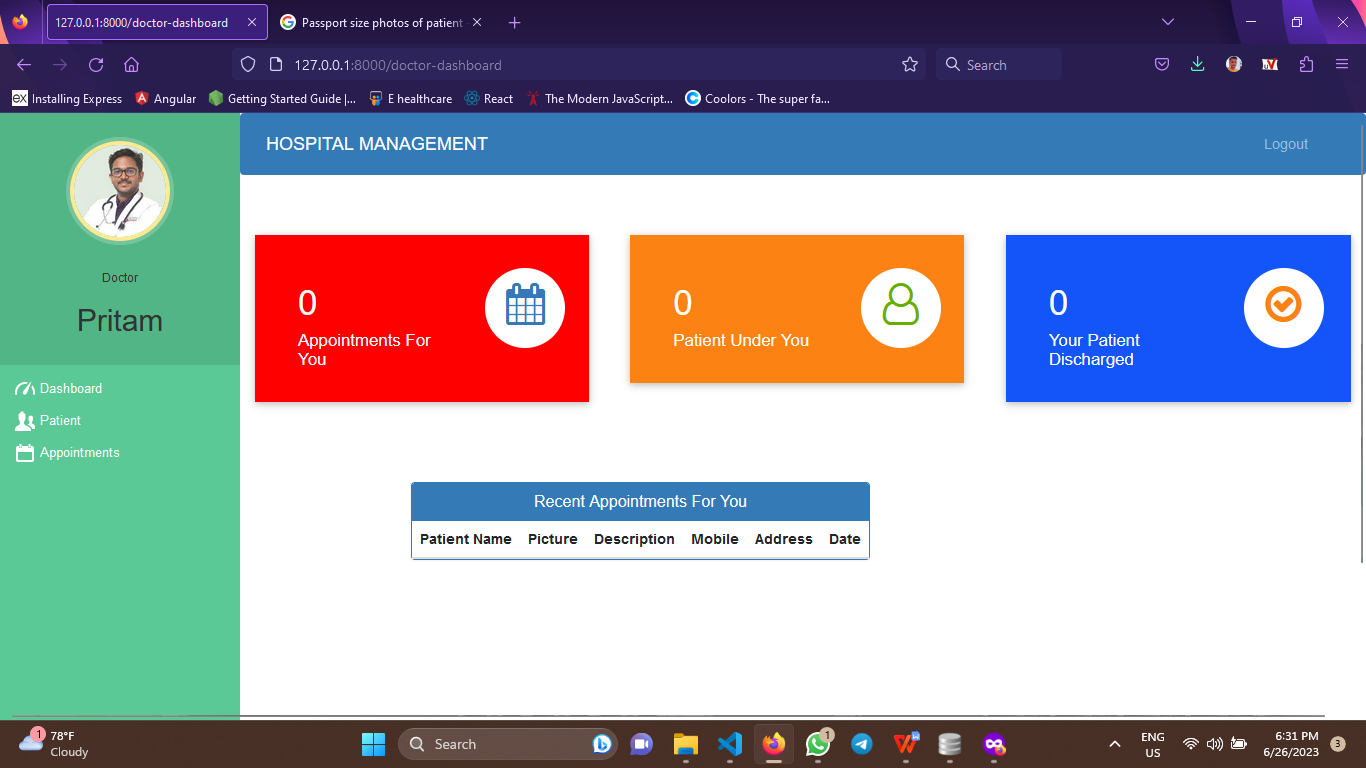
* + - 1. **Doctor Login -**

****

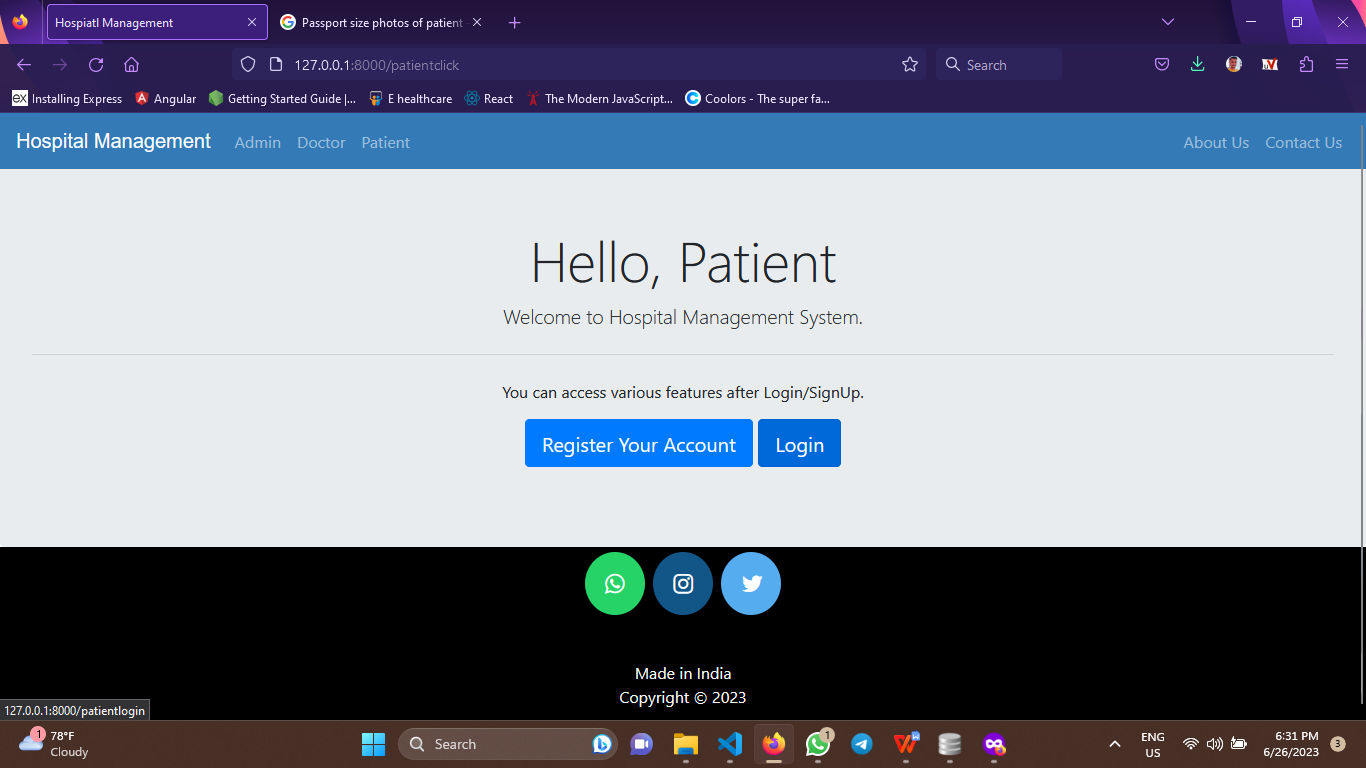
* + - 1. **Doctor need approval of admin -**

****

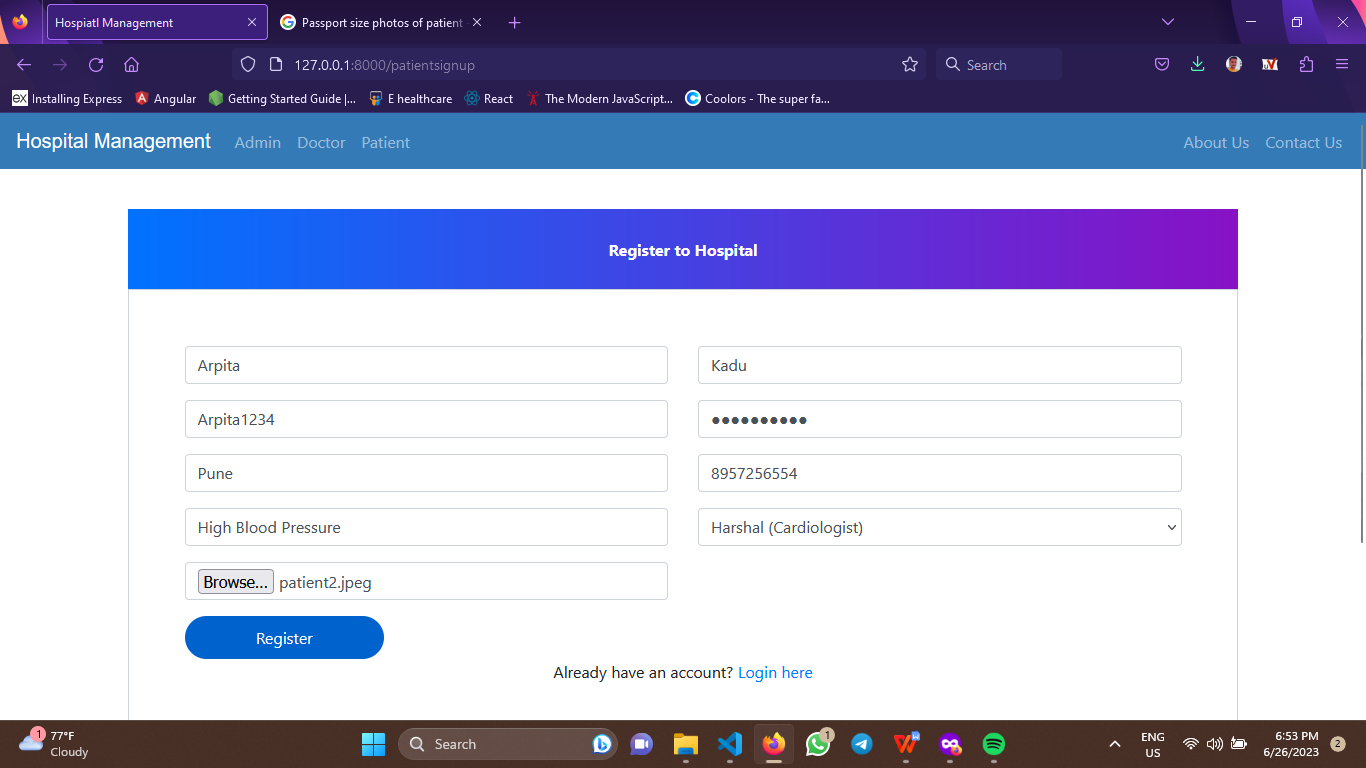
* + - 1. **Doctor Dashboard after approval of admin -**



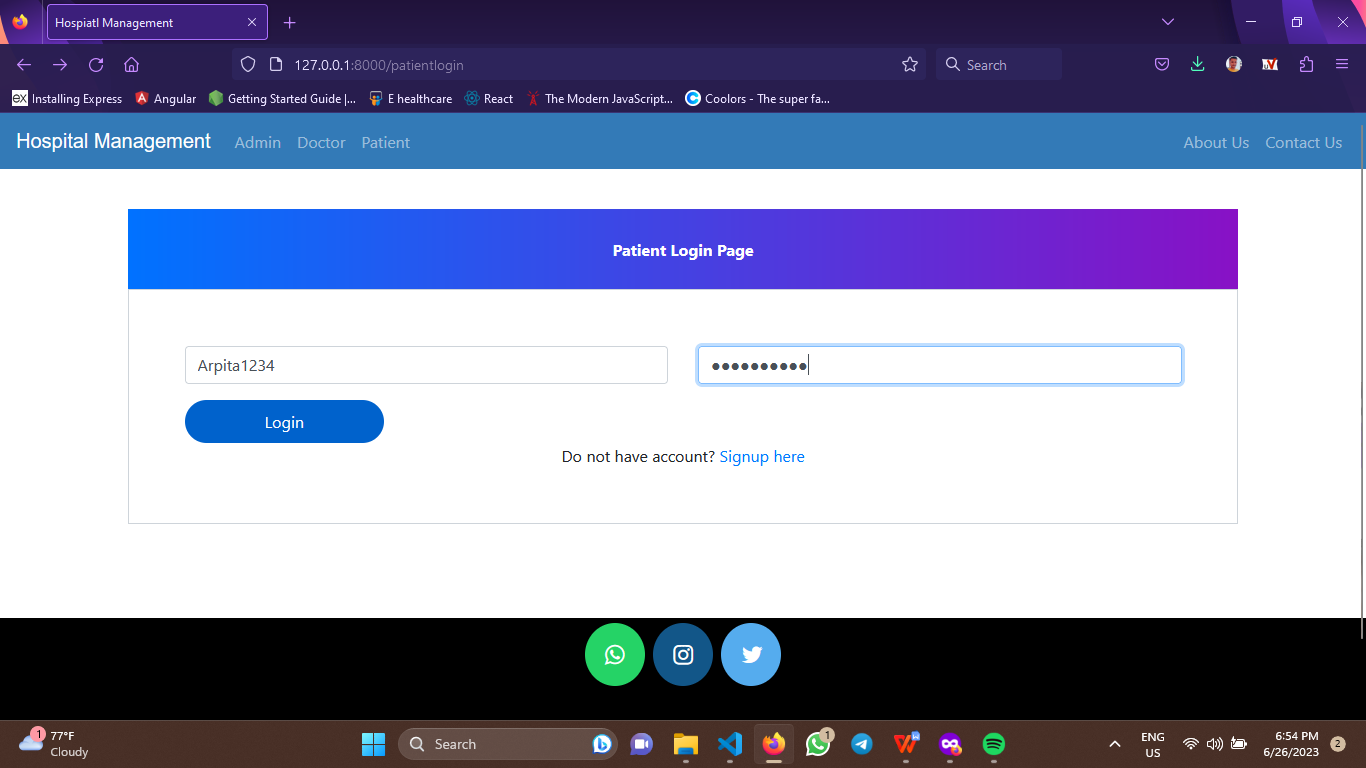
* + 1. **Patient -**



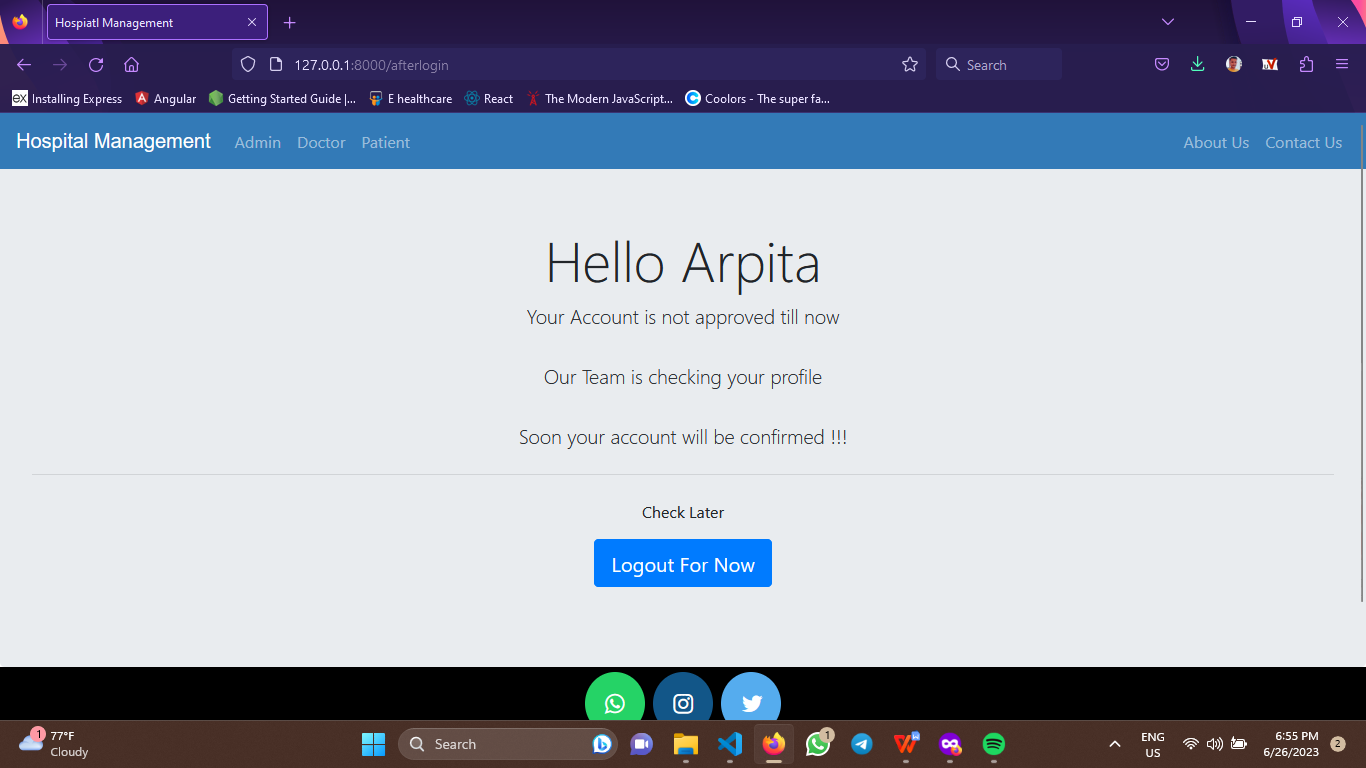
* + - 1. **Patient Sign up -**



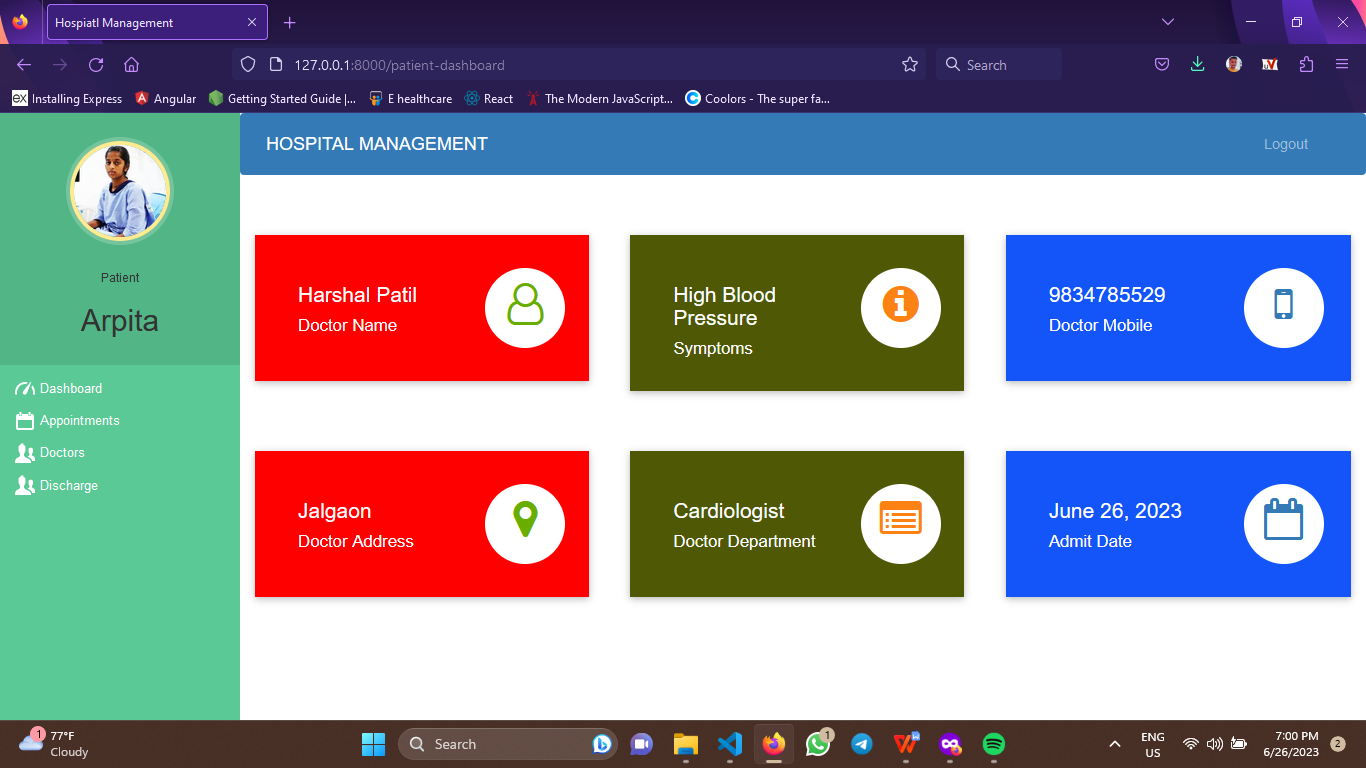
* + - 1. **Patient Login -**



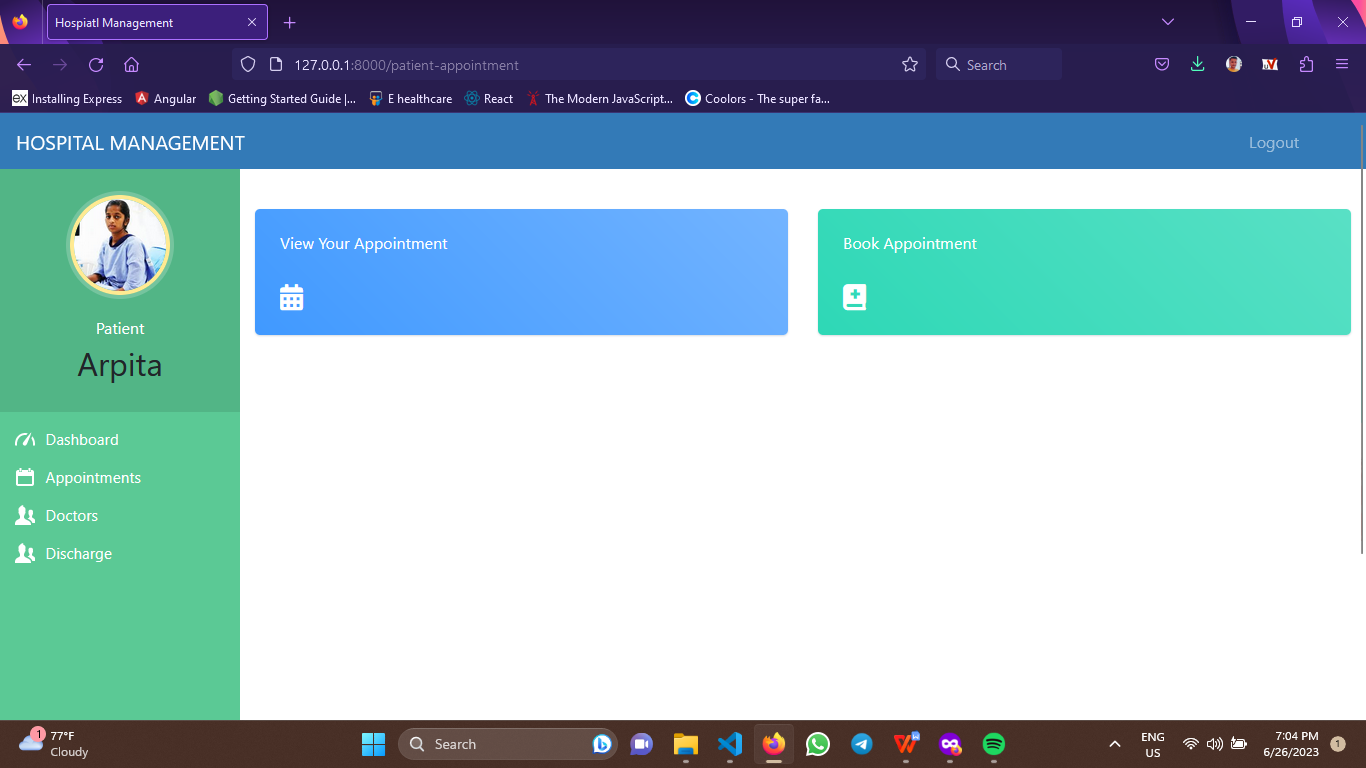
* + - 1. **Patient need approval of admin -**

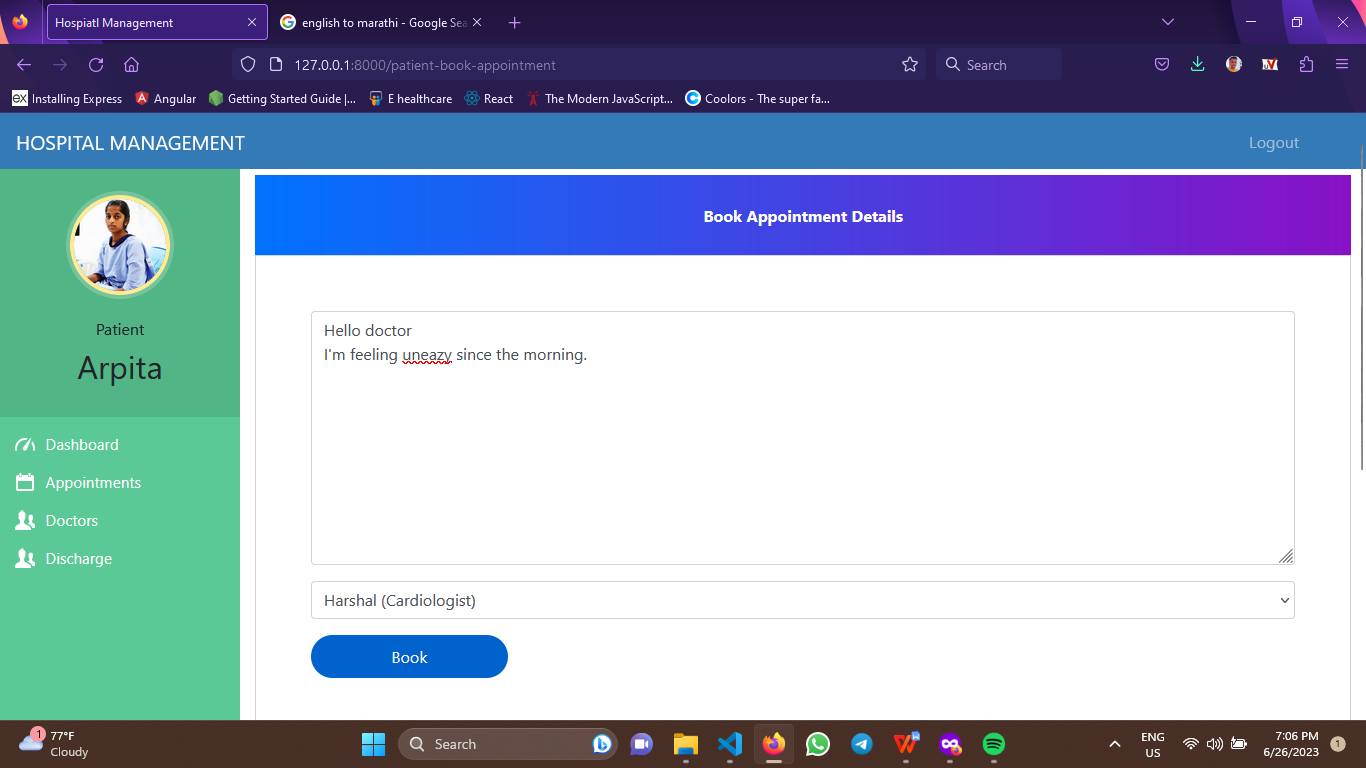


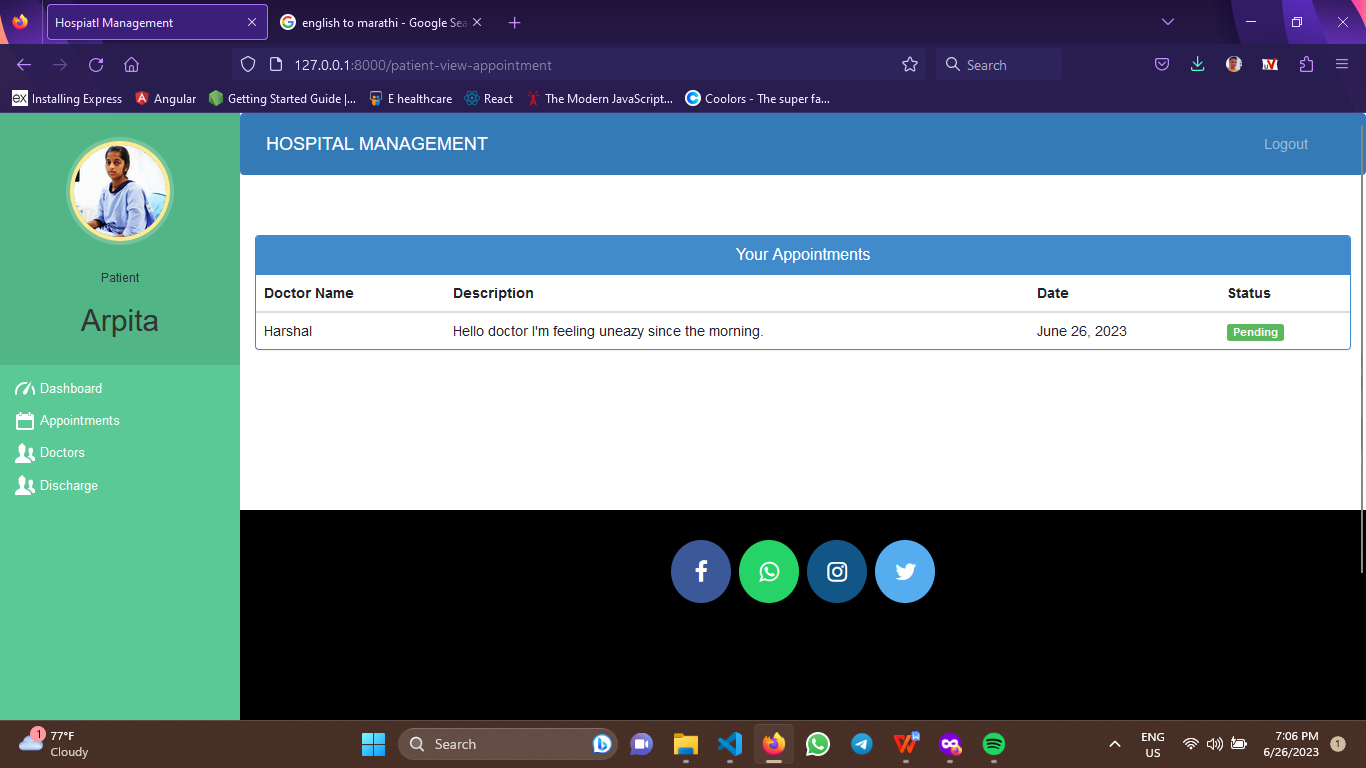
* + - 1. **Patient dashboard after approval of admin -**



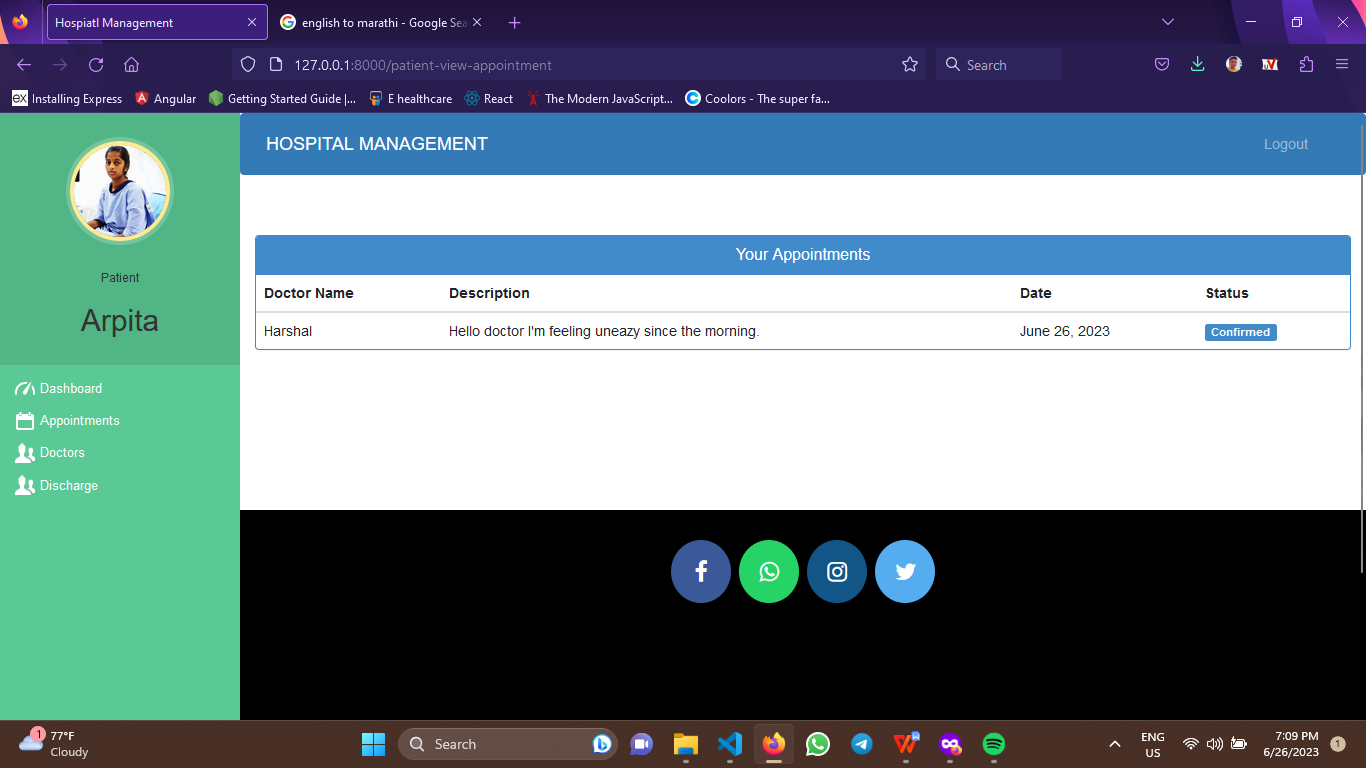
* + - 1. **Patient can Book Appointment -**



****



* + - 1. **Patient Appointment Confirmation After Admin Approval-**

****

**CHAPTER 5**

**CONCLUSION**

**5.1 Limitations -**

There are some limitations in this system which makes system sometimes ineffective in working.

* This process is automated but little-bit time consuming.
* Admin cannot accept payment online.
* System is not able to send messages to users.
* System has not been implemented with third-party payment app.

**5.2 Future Enhancement -**

The project has very big scope in the future implementation. The project needs to be updated in the future and with the new features, the system working will be more efficient and can be used more effectively. New components can be added to extend the use of the system very effectively.

* Sending the email to the users.
* Adding online payment method for the payment.
* The payments can be done by using UPI, Debit Cards and other payment Options.
* Will create android app for easy accessibility for user

**5.3 Conclusion -**

The conclusion of an hospital management project would depend on various factors and would be specific to the goals and outcomes of the project. However, here are some possible conclusions:

* Improved Access to Hospital: The Hospital Management project aimed to provide convenient access to hospital services through digital platforms. If it has successfully achieved this goal by offering online consultations, medical information, appointment booking, or remote monitoring, it can be considered a success in terms of improving healthcare accessibility.
* Enhanced Patient Experience: The project might have focused on improving the overall patient experience by providing user-friendly interfaces, personalized services, and efficient communication channels. Positive feedback from patients regarding the ease of use, convenience, and quality of services can indicate a successful implementation.
* Efficient Hospital Management: If the project has successfully implemented features to streamline hospital management processes, such as electronic health records, appointment scheduling, prescription management, or billing systems, it can contribute to the efficiency of hospital services providers and enhance the overall project success.
* Cost-effectiveness: The project's success can also be measured in terms of cost-effectiveness. If it has helped reduce hospitals costs by minimizing unnecessary hospital visits.

**CHAPTER 6**

**BIBLIOGRAPHY**

**6.1 References -**

1. Books-
2. Python programming: Powerful Object-Oriented Programming 4th edition by Mark Lutz.
3. OOSE and Design with UML by Michael R Blaha.
4. Web design with HTML, CSS & JS by Pearson.
5. Websites-
6. <https://stackoverflow.com/>
7. <https://www.python.org/>
8. <https://www.djangoproject.com/>
9. <https://www.geeksforgeeks.org/>
10. <https://www.tutorialspoint.com/index.html>

**6.2 ANNEXURE- SAMPLE PROGRAM CODE**

**Code -**

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

{% load widget\_tweaks %}

{% block content %}

<head>

<style media="screen">

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

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

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

E-Health Care System39

.form-control {

border-radius: 1.5rem;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

.menu {

top: 50px;

}

</style>

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

<script

src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script><script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script></head>

**<br><br>**

<form method="post">

{% csrf\_token %}

<div class="container register-form">

<div class="form">

<div class="note">

<p>Book Appointment Details</p>

</div>

<div class="form-content">

<div class="row">

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

<div class="form-group">

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

</div>

<div class="form-group">

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

</div>

<div class="form-group">

{% render\_field appointmentForm.patientId class=”form-control” placeholder=”

patient”%}

</div>

</div>

</div>

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

E-Health Care System41

</div>

</div>

</div>

</form>

{% endblock content %}

<!DOCTYPE html>

{% load static %}

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

<head>

<meta charset="utf-8">

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

<title>Hospiatl Management</title>

<style media="screen">

.jumbotron {

margin-top: 0px;

margin-bottom: 0px;

}

.jumbotron h1 {

text-align: center;

}

.alert {

margin: 0px;

}

E-Health Care System42

</style>

<title>sumit</title>

</head>

<body>

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

<br><br>

<center>

<h3 class='alert alert-success' style="margin-bottom:0px;">About Us !</h3></center>

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

<h1 class="display-4">Hello</h1>

<p class="lead">A service dedicated to Hospital Admin, Doctor and Patient.</p><hr class="my-4">

<p>Explore our Website.</p>

<p class="lead">

<a class="btn btn-primary btn-lg" href="/" role="button">HOME</a>

</p>

</div>

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

</body>

</html>