Project Report



**HealthCare Hub**

Submitted in partial fulfilment of requirement for the degree of MCA

Ajay Kumar Garg Engineering College- MCA,Ghaziabad

Under Guidance of

Submitted By:

**MS. Aman Gupta** **Abhishek Malik 2200270140001**

**Akansha Arora 2200270140003**

**Ashish Rai 2200270140009**

**Kajal Teetwal 2200270140020**

**Dr. A.P.J. Abdul Kalam Technical University,Uttar Pradesh, Lucknow**

# ABSTRACT

**Health Care Management System**

A Healthcare Management System (HMS) is a comprehensive software solution designed to streamline and optimize the operations of healthcare organizations, including hospitals, clinics, and medical facilities. This integrated system encompasses various modules and functionalities to efficiently manage patient care, administrative tasks, and financial aspects within the healthcare ecosystem. The primary objectives of an HMS are to enhance patient care quality, improve operational efficiency, ensure regulatory compliance, and facilitate seamless communication among healthcare professionals. Key components of an HMS include patient registration, appointment scheduling, electronic health records (EHR), billing and invoicing, pharmacy and inventory management, laboratory and radiology information systems, ambulance booking system, blood bank system, organ doner system, chat box system for helpline and many more functionalities. With the incorporation of advanced technologies and secure data management, a Healthcare Management System plays a pivotal role in transforming the delivery of healthcare services, ultimately leading to better patient outcomes and enhanced healthcare administration.

A healthcare management system refers to a set of processes, technologies, and tools that are used to efficiently manage and deliver healthcare services within a healthcare organization or system. These systems play a crucial role in improving patient care, reducing costs, and enhancing overall operational efficiency.

Implementing an effective healthcare management system requires careful planning, customization to the specific needs of the healthcare organization, and training for staff members. The ultimate goal is to enhance patient care, streamline operations, reduce errors, and improve the overall efficiency and quality of healthcare services.

|  |  |
| --- | --- |
| **Technology:**  React  Node Js  Javascript  Mongodb |  |
| **GROUP MEMBER –** |

**Abhishek Malik 2200270140001**

**Akansha Arora 2200270140003**

**Ashish Rai 2200270140009**

**Kajal Teetwal 2200270140020**

**Chapter 1:- INTRODUCTION**

The healthcare sector stands at the intersection of cutting-edge medical science, compassionate patient care, and transformative technological advancements. As societies evolve, so too must healthcare systems to meet the demands of an ever-growing and diverse population. This introduction explores the multifaceted landscape of healthcare systems, encompassing a myriad of components that collectively aim to enhance the quality, efficiency, and accessibility of healthcare services.

### Evolution of Healthcare Systems: A Historical Perspective

The roots of healthcare systems trace back through the annals of history, evolving from rudimentary healing practices to sophisticated, technology-driven systems. Over the centuries, healthcare has transitioned from localized and often unstandardized approaches to organized systems influenced by medical discoveries, societal needs, and the imperative to provide comprehensive care to individuals across the globe.

### The Core Components of Modern Healthcare Systems

Today's healthcare systems are intricate webs of interconnected elements designed to address the holistic needs of patients. Central to this complexity is the adoption and integration of electronic health records (EHRs). EHR systems streamline data management, providing a comprehensive repository of patient information accessible to healthcare professionals, facilitating informed decision-making, and improving continuity of care.

Telemedicine has emerged as a transformative force, bridging geographical gaps and bringing medical expertise to remote or underserved areas. The integration of telehealth technologies facilitates virtual consultations, remote patient monitoring, and efficient healthcare delivery, fostering a paradigm shift in how healthcare services are accessed and delivered.

Data analytics and artificial intelligence (AI) have become pivotal tools in healthcare system enhancement. These technologies empower healthcare professionals with predictive analytics, personalized treatment plans, and insights derived from vast datasets. The synergy of data analytics and AI contributes to evidence-based medicine, driving innovations in diagnostics, treatment strategies, and healthcare research.

### Patient-Centric Approaches and Empowerment

A hallmark of modern healthcare systems is the emphasis on patient-centric care. Initiatives that prioritize the patient experience include the development of user -friendly interfaces, patient portals, and communication tools. These innovations empower individuals to actively engage in managing their health, fostering a collaborative relationship between patients and healthcare providers. Additionally, patient-centric approaches aim to reduce health disparities, promote preventive care, and enhance overall well-being.



### Challenges and Opportunities in Healthcare Systems

Despite significant progress, healthcare systems grapple with challenges such as interoperability issues, data security concerns, and the need for widespread adoption of emerging technologies. Furthermore, the global landscape introduces additional complexities, including varying healthcare infrastructures, cultural considerations, and socioeconomic factors that influence healthcare delivery.

However, within these challenges lie opportunities for innovation and improvement. The ongoing quest for interoperability, advancements in cybersecurity measures, and the integration of emerging technologies present avenues for transformative change.

Collaborative efforts among healthcare stakeholders, policymakers, and technological innovators contribute to a resilient and adaptable healthcare ecosystem.

### Scope and Objectives of the Exploration

This exploration of healthcare systems aims to dissect the intricate components that collectively contribute to the functioning of modern healthcare. It seeks to analyze the evolution of healthcare practices, evaluate the impact of technological advancements, and assess the potential of patient-centric approaches. The examination of challenges and opportunities within healthcare systems aims to inform future developments and initiatives that strive for a more inclusive, efficient, and patient-focused healthcare landscape.

As we embark on this exploration, it is crucial to recognize the dynamic nature of healthcare systems, where ongoing innovation and collaboration pave the way for a healthier and more connected global community.

# Chapter:-2

**Problem Statement And Description**

## Problem Statement:

The healthcare industry faces multifaceted challenges that impact the quality, efficiency, and accessibility of healthcare services. Key issues include fragmented health information systems, limited interoperability, security concerns, and disparities in healthcare delivery. Addressing these challenges is paramount to achieving a cohesive and patient-centric healthcare ecosystem.

## Description:

The contemporary healthcare landscape is marked by remarkable advancements in medical science, yet the industry grapples with challenges that hinder the seamless delivery of care. The proliferation of disparate health information systems often results in fragmented patient data, impeding the holistic understanding of individual health profiles. This lack of interoperability hinders healthcare providers' ability to access comprehensive patient information in real-time, leading to suboptimal decision-making and potentially compromising patient outcomes.

Security concerns surrounding electronic health records (EHRs) and other health-related data sources pose additional challenges. Protecting sensitive patient information from cyber threats is crucial to maintaining the integrity and trustworthiness of healthcare systems. Ensuring robust cybersecurity measures is essential for safeguarding patient privacy and preventing unauthorized access to confidential health data.

Furthermore, disparities in healthcare delivery persist, influenced by socioeconomic factors, geographical barriers, and cultural considerations. These disparities contribute to unequal access to medical services, hindering efforts to provide equitable healthcare for all individuals.

The overarching problem lies in the need for an integrated and patient-centric healthcare system that addresses these challenges comprehensively. The project seeks to develop solutions that enhance interoperability, strengthen cybersecurity measures, and promote inclusive healthcare practices. By doing so, we aim to contribute to the creation of a healthcare ecosystem that prioritizes patient well-being, fosters collaboration among healthcare stakeholders, and embraces technological innovations to improve overall healthcare delivery.

## Objectives:

* **Interoperability Enhancement:** Develop and implement strategies to improve interoperability among disparate health information systems, enabling seamless data exchange and access to comprehensive patient records.
* **Cybersecurity Measures:** Implement robust cybersecurity protocols to safeguard electronic health records and sensitive health data, ensuring the confidentiality, integrity, and availability of patient information.
* **Patient-Centric Approaches:** Introduce initiatives and technologies that empower patients, enhance their engagement in healthcare management, and contribute to a more patient-centric model of care delivery.
* **Equity and Accessibility:** Explore solutions to reduce healthcare disparities, addressing factors such as socioeconomic status, geographical location, and cultural nuances to improve the accessibility and quality of healthcare services for diverse populations.

Through these objectives, the project endeavors to contribute to the ongoing evolution of healthcare systems, fostering a more integrated, secure, and patient-focused approach to healthcare delivery.

# Chapter:-3

**Software Requirements Specification**

1. **Introduction**

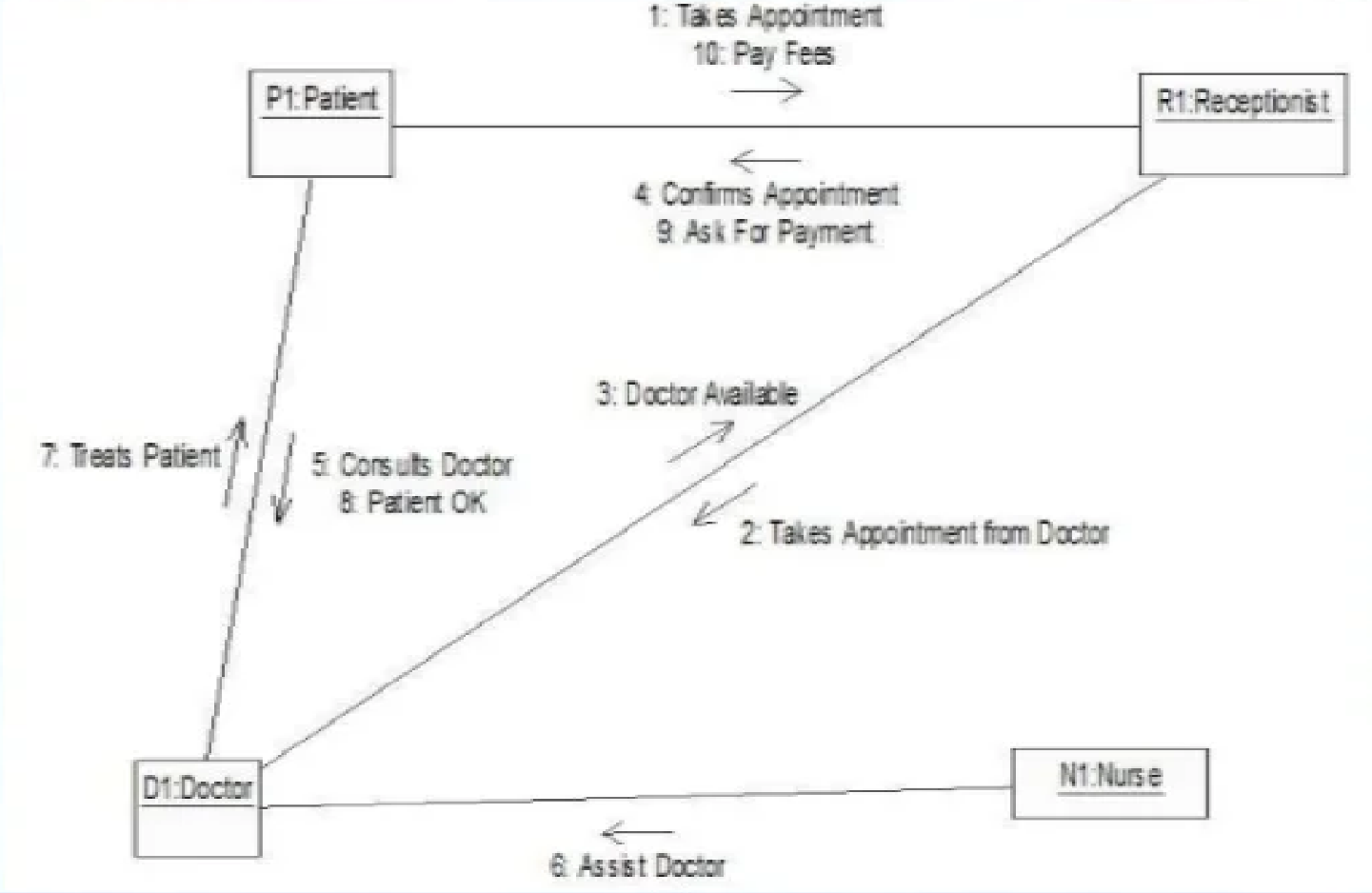
In contemporary society, the ubiquity of mobile devices, particularly Android phones, has led to a surge in the utilization of online services. As the demand for efficient and accessible healthcare solutions grows, there arises a need to harness technology for streamlining healthcare processes and minimizing waste. This project aims to address the issue of leftover medication and enhance the overall healthcare experience by developing a comprehensive Healthcare Management System.

# Purpose

The current healthcare landscape is marked by challenges such as inefficient medication management and lack of a cohesive system for patients to access healthcare services remotely. After prescriptions are fulfilled, any leftover medications often go unused, leading to waste and potential hazards. Additionally, patients face time-consuming processes when visiting healthcare facilities, which can be cumbersome and contribute to inefficiencies in the healthcare delivery system.

## Chapter:- 4

### Collaboration Diagram



A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

To develop a health care system website which shows the flows between the patient and hospital and live bed update .Main features of provide ambulance and bed website are as follows:

-Admin user can search hospital,view description of selected hospital,book appoint ment,check bed availbility and book ambulance.

-Its shows the activity flow of editing,adding and updating of details related to patient

And hospital.

-Its shows the full description and flow of blood donar and bed avaibility.

**Chapter:-4**

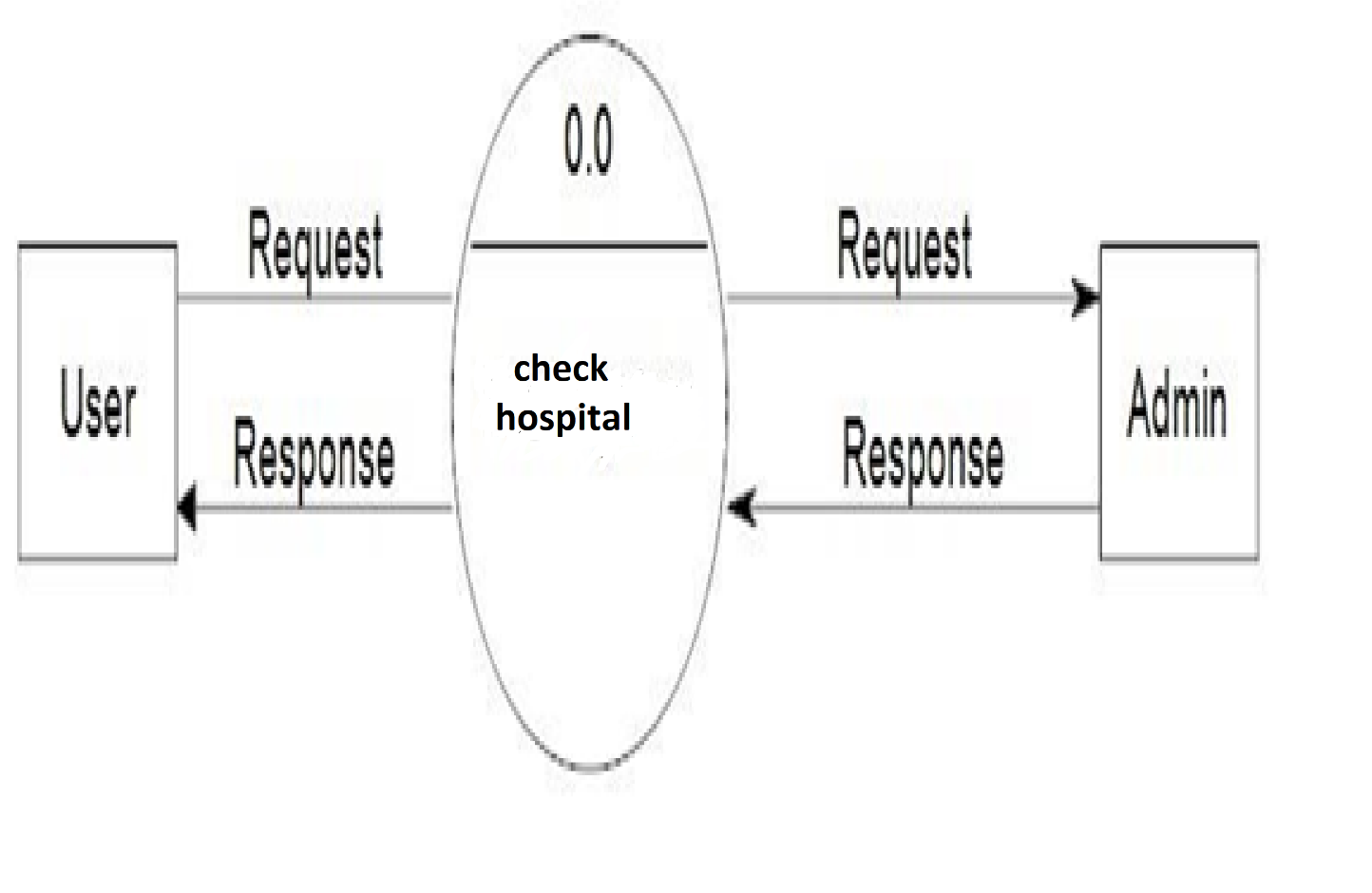
**4.8 Data Flow Diagram**

Health care Website is actually a type of software that allows the staff of hospital to manage and accept the request over the Internet or in the hospital. Let us understand the working of the hospital care website by using data flow diagram. DFD for health care System is shown below.

In the below diagram, the Input and Output of the system are shown. The system is designed and established across the world with input and output at this level. Health care System has the following input :

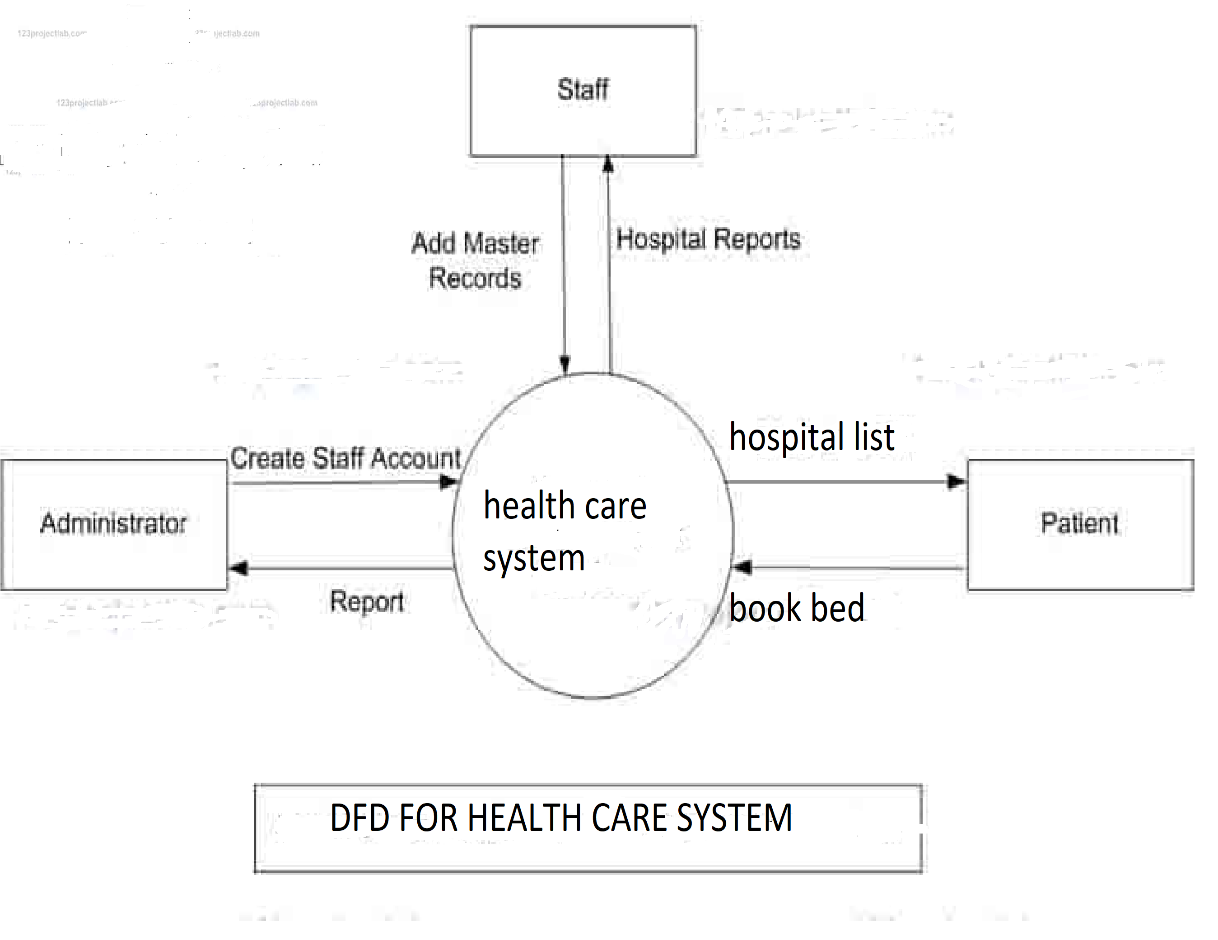
-book bed is input as the user it request to book bed and then get reponse and as output admin get all the details of hospital and further response if when needed. In the end, with the use of the amount bed available and daily bed booking depletion, it is easy to prepare a report of management. Further, the admin gets this report of management.

#### DFD zero level



#### DFD first level

For processing the request, process 1.0 is responsible. For bed, the blood donation activities involved are represented by processes 2.0, 3.0, and 4.0. The detailed information about bed availability should be available to create and report management and the list of hospital that are available ‘in-stock’ should be kept by maintaining the in data (describes the records of datasets such as their name, their content, source, many useful information, etc.) at the same time.



**Chapter**

**Home Page**

import React from "react";

import web from "../src/images/image1.png";

import { NavLink } from "react-router-dom";

import one from "../src/images/hospital1.png";

import two from "../src/images/hospital2.png";

import ambulance from "../src/images/ambulance.png";

import image from "../src/images/image3.png";

import GetStarted from "./GetStarted";

import Ambulance from "./Ambulance";

function Home() {

return (

<div >

<section id="header" className="d-flex align-items-center col " >

<div className="container-fluid nav\_bg">

<div className="row">

<div className="col-10 mx-auto">

<div className="row">

<div className="col-md-6 pt-5 pt-lg-0 order-2 order-lg-1 d-flex justify-content-center flex-column">

<h1>Specialists Healing Humankind</h1>

<h1>Experts To Provive Advance </h1>

<h1><strong className="br">Medical Treatments And Facilities</strong></h1>

<a href="/GetStarted" className="my-5" >Get Started</a>

</div>

<div className="col-lg-6 order-1 order-lg-2 header-img">

<img src={web} className="img=fluid animated" alt="home-img"/></div>

</div>

</div>

</div>

</div></section>

<section className="col mx-6 my-6">

<h1 className="mx-6 my-6">Emergency Medical Services offered by Every Hospital</h1>

<div className="row">

<h4>

<ul>

<li>Stroke</li>

<li>Roat Traffic Accident</li>

<li>Heart Attack</li>

<li>Fits or Epileptic Seizures</li>

<li>Respiratory Problem</li>

<li>Burns</li>

<li>Bone Fractures</li>

<li>Diabetic Coma</li>

<li>Poisoning</li>

<li>Mother and Child Emergency</li>

<li>Gastritis</li>

<li>Gall Bladder Stone Emergency</li>

<li>Kidney Stone Emergency</li>

<li>Animal Bite</li>

<li>Allegric Reactions</li>

<li>Appendicitis</li>

<li>Bleeding</li>

<li>Severe Pain</li>

<li>Sepsis</li>

<li>Severe Trauma</li>

<li>High Grade Fever</li>

<li>Dehydration</li>

</ul>

</h4>

</div>

</section>

<section className="mx-5 my-5">

<div class="gap">

<h3 className="text-center text-primary mx-5 my-5">At Felix, We Are Equipped To Handle Critically Ill Or Injured Patients With Life-Threatening Conditions

</h3>

<h2 id="right"><img className="img-fluid" src={ambulance}></img> 24X7 Emergenct Number<button type="button"class="btn btn-danger">Emergency</button></h2>

</div>

</section>

</div>

);

}

export default Home;

**Login Page For User**

import react,{useState} from "react";

import photo from "../src/images/login.png";

import { NavLink } from "react-router-dom";

import axios from "axios";

import { useNavigate } from "react-router-dom";

function LoginUser()

{

const navigate = useNavigate();

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const login = async () => {

let resp = await axios.post("http://localhost:3001/login", {

email: email,

password: password});

console.log("Login Response.");

console.log(resp);

if(resp.data.message){

window.sessionStorage.setItem("email", resp.data.message.email);

window.sessionStorage.setItem("firstname", resp.data.message.firstname);

setTimeout(() => {

navigate("/home");

}, 1000)

}

}

return (

<div >

<div className="container-sm">

<section className="mx-5 my-5"><section className="vh-100">

<div className="container-fluid">

<div className="row">

<div className="col-sm-6 text-black">

<div className="px-5 ms-xl-4">

<i className="fas fa-crow fa-2x me-3 pt-5 mt-xl-4" ></i>

<span className="h1 fw-bold mb-0">HealthcareHub</span>

</div>

<div className="d-flex align-items-center h-custom-2 px-5 ms-xl-4 mt-5 pt-5 pt-xl-0 mt-xl-n5">

<form >

<h3 className="fw-normal mb-3 pb-3" >Log in</h3>

<div className="form-outline mb-4">

<input value={email} onChange={(e) => setEmail(e.target.value)} type="email" id="form2Example18" className="form-control form-control-lg"

/>

<label className="form-label" for="form2Example18">Email address</label>

</div>

<div className="form-outline mb-4">

<input value={password} onChange={(e) => setPassword(e.target.value)} type="password" id="form2Example28" className="form-control form-control-lg" />

<label className="form-label" for="form2Example28">Password</label>

</div>

<div className="pt-1 mb-4">

<button onClick={() => login()} className="btn btn-info btn-lg btn-block" type="button"

>Login</button>

</div>

<p className="small mb-5 pb-lg-2"><a className="text-muted" href="#!">Forgot password?</a></p>

<p>Don't have an account? <a href="./Register" className="link-info">Register here</a></p>

</form>

</div>

</div>

<div className="col-sm-6 px-0 d-none d-sm-block">

<img src={photo}

alt="Login image" className="w-100 vh-100" />

</div>

</div>

</div>

</section></section>

</div>

</div>

)

}

export default LoginUser;

**Register**

import React, { useEffect, useState } from "react";

import axios from "axios";

import { toast } from "react-toastify";

import { useNavigate } from "react-router-dom";

import 'react-toastify/dist/ReactToastify.css';

function Register() {

const navigate = useNavigate();

const [fname,setFName]=useState('');

const [lname,setLName]=useState('');

const [phone,setphone]=useState('');

const [email,setemail]=useState('');

const [pass,setpass]=useState('');

const [cPass,setcPass]=useState('');

const handleRegistrationSubmit = async () => {

const resp = await axios.post("http://localhost:3001/adduser", {firstname:fname,lastname:lname,phonenumber:phone,email:email,password:pass,confirmpassword:cPass})

console.log(resp);

toast.success(resp.data.message, {

position: "top-right",

autoClose: 5000,

hideProgressBar: false,

closeOnClick: true,

pauseOnHover: true,

draggable: true,

progress: undefined,

theme: "light",

});

setTimeout(() => {

navigate("/Home");

}, 6000)

}

return (

<>

<section>

<section className="h-100 bg-dark ">

<form >

<div className="container py-5 h-100">

<div className="row d-flex justify-content-center align-items-center h-100">

<div className="col">

<div className="card card-registration my-4">

<div className="row g-0">

<div className="col-xl-6 d-none d-xl-block">

<img

src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-registration/img4.webp"

alt="Sample photo"

className="img-fluid"

id="dca"

/>

</div>

<div className="col-xl-6" method="Post">

<div className="card-body p-md-5 text-black">

<h3 className="mb-5 text-uppercase">Registration</h3>

<div className="row">

<div className="col-md-6 mb-4">

<div className="form-outline">

<input

type="text"

name="firstname"

className="form-control form-control-lg"

autoComplete="off"

value={fname}

onChange={(e) => setFName(e.target.value)}

required

/>

<label className="form-label">First name</label>

</div>

</div>

<div className="col-md-6 mb-4">

<div className="form-outline">

<input

type="text"

name="lastname"

id="form3Example1n"

class="form-control form-control-lg"

value={lname}

onChange={(e) => setLName(e.target.value)}

required

/>

<label className="form-label">Last name</label>

</div>

</div>

</div>

<div className="row">

<div className="col-md-6 mb-4">

<div className="form-outline">

<input

type="number"

name="phonenumber"

id="form3Example1m1"

className="form-control form-control-lg"

value={phone}

onChange={(e) => setphone(e.target.value)}

required

/>

<label className="form-label">Phone No</label>

</div>

</div>

</div>

<div class="form-outline mb-4">

<input

type="email"

name="ema"

id="form3Example8"

className="form-control form-control-lg"

value={email}

onChange={(e) => setemail(e.target.value)}

required

/>

<label className="form-label">Email id</label>

</div>

<div className="form-outline mb-4">

<input

type="password"

name="password"

className="form-control form-control-lg"

autoComplete="off"

value={pass}

onChange={(e) => setpass(e.target.value)}

required

/>

<label className="form-label">Password</label>

</div>

<div className="form-outline mb-4">

<input

type="password"

name="confirmpassword"

className="form-control form-control-lg"

value={cPass}

onChange={(e) => setcPass(e.target.value)}

required

/>

<label

className="form-label"

name="ConfirmPassword"

>

Confirm Password

</label>

</div>

<div className="d-flex justify-content-end pt-3">

<button

type="button"

className="btn btn-warning btn-lg ms-2"

value="Submit"

onClick={() => handleRegistrationSubmit()}

>

Submit

</button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

</section>

</section>

</>

);

}

export default Register;

**Login Page For Hospital**

import react,{useState} from "react";

import photo from "../src/images/login.png";

import { NavLink } from "react-router-dom";

import axios from "axios";

import { useNavigate } from "react-router-dom";

function HospitalLogin()

{

const navigate = useNavigate();

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const login = async () => {

let resp = await axios.post("http://localhost:3001/hospitallogin", {

email: email,

password: password});

console.log("Login Response.");

console.log(resp); if(resp.data.message){

window.sessionStorage.setItem("email", resp.data.message.email);

window.sessionStorage.setItem("firstname", resp.data.message.firstname);

setTimeout(() => {

navigate("/home");

}, 6000)

}

}

return (

<div >

<div className="container-sm">

<section className="mx-5 my-5"><section className="vh-100">

<div className="container-fluid">

<div className="row">

<div className="col-sm-6 text-black">

<div className="px-5 ms-xl-4">

<i className="fas fa-crow fa-2x me-3 pt-5 mt-xl-4" ></i>

<span className="h1 fw-bold mb-0">HealthcareHub</span>

</div>

<div className="d-flex align-items-center h-custom-2 px-5 ms-xl-4 mt-5 pt-5 pt-xl-0 mt-xl-n5">

<form >

<h3 className="fw-normal mb-3 pb-3" >Log in</h3>

<div className="form-outline mb-4">

<input value={email} onChange={(e) => setEmail(e.target.value)} type="String" id="form2Example18" className="form-control form-control-lg" />

<label className="form-label" for="form2Example18">LoginId</label>

</div>

<div className="form-outline mb-4">

<input value={password} onChange={(e) => setPassword(e.target.value)} type="password" id="form2Example28" className="form-control form-control-lg" />

<label className="form-label" for="form2Example28">Password</label>

</div>

<div className="pt-1 mb-4">

<button onClick={() => login()} className="btn btn-info btn-lg btn-block" type="button">Login</button>

</div>

<p className="small mb-5 pb-lg-2"><a className="text-muted" href="#!">Forgot password?</a></p>

<p>Don't have an account? <a href="./HosRegister" className="link-info">Register here</a></p>

</form>

</div>

</div>

<div className="col-sm-6 px-0 d-none d-sm-block">

<img src={photo}

alt="Login image" className="w-100 vh-100" />

</div>

</div>

</div>

</section></section>

</div>

</div>

)

}

export default HospitalLogin;

**Schema for Hospital**

const mongoose=require("mongoose")

const HospitalSchema=new mongoose.Schema({

name:{

type:String,

required:true

},

pincode:{

type:String,

required:true

},

phone:{

type:String,

required:true

},

emergency:{

type:String,

required:true

},

email:{

type:String,

required:true

},

address:{

type:String,

required:true

},

state:{

type:String,

required:true

},

district:{

type:String,

required:true

},

password:{

type:String,

required:true

},

confirmPassword:{

type:String

}

})

const HospitalRegister=mongoose.model('Hospital',HospitalSchema);

module.exports=HospitalRegister;

**Schema for Login**

const mongoose=require("mongoose");

const userSchema=new mongoose.Schema({

firstname:{

type:String,

required:true

},

lastname:{

type:String,

required:true

},

phonenumber:{

type:Number,

required:true,

unique:true

},

email:{

type:String,

required:true,

unique:true

},

password:{

type:String,

required:true

},

confirmpassword:{

type:String,

required:true

}

})

//now we need to create collection

const UserRegister =mongoose.model("register",userSchema);

module.exports=UserRegister;

**Api**

const express = require("express");

const mongoose = require("mongoose");

const app = express();

const User = require("./database/User");

const Port = 3001;

const UserRegister = require("./database/registers");

const HospitalRegister=require("./database/Hospital");

const connectDB = require("./database/registerConnection");

const DonerFindSchema=require("./database/DonerFindSchema");

const DonerRegisterSchema=require('./database/DonerRegisterSchema');

const AmbulanceRegister=require('./database/Ambulance')

var cors = require("cors");

app.use(cors());

//middlewarefor parsing

app.use(express.urlencoded({ extended: true }));

app.use(express.json());

connectDB();

//for registration page

app.post("/home", async (req, res) => {

/\* const DataSave=new connectDB(req.body);

await DataSave.save();

res.send("Save Data");\*/

console.log(req.body);

});

//for registration page

app.post("/adduser", async (req, res) => {

try {

const {

firstname,

lastname,

email,

confirmpassword,

phonenumber,

password,

} = req.body;

console.log(req.body);

console.log(password);

console.log(confirmpassword);

if (password === confirmpassword) {

const userToBeAdded = new UserRegister({

firstname,

lastname,

phonenumber,

email,

password,

confirmpassword,

});

await userToBeAdded.save();

console.log("Saved successfully");

res.status(201).json({ message: "Registration Successful" });

} else {

//log error.

console.log("Not Saving.")

res.status(500).json({ error: error });

}

} catch (error) {

res.status(500).json({ error: error });

}

});

//for contact us page

app.post("/contactus", async (req, res) => {

try {

const {

Name,

Phoneno,

Email,

Message, } = req.body;

console.log(req.body);

const userContact= new User({

Name,

Phoneno,

Email,

Message,

});

await userContact.save();

console.log(Email);

console.log(req.body);

res.status(201).json({ message: "Registration Successful" });

} catch (error) {

res.status(500).json({ error: error });

}

});

//Find Doner

app.post("/finddoner", async (req, res) => {

try {

const {

Name,

Phoneno,

Email,

Address,

State,

District,

BloodGroup } = req.body;

console.log(req.body);

const donerfind= new DonerFindSchema({

Name,

Phoneno,

Email,

Address,

State,

District,

BloodGroup

});

await donerfind.save();

console.log(req.body);

res.status(201).json({ message: "Registration Successful" });

} catch (error) {

res.status(500).json({ error: error });

}

});

//Register Doner

app.post("/registerdoner", async (req, res) => {

try {

const {

Name,

Phoneno,

Email,

Address,

State,

District,

BloodGroup } = req.body;

console.log(req.body);

const donerRegister= new DonerRegisterSchema({

Name,

Phoneno,

Email,

Address,

State,

District,

BloodGroup

});

await donerRegister.save();

console.log(req.body);

res.status(201).json({ message: "Registration Successful" });

} catch (error) {

res.status(500).json({ error: error });

}

});

//Hospital Registration

app.post("/hospitalreg", async (req, res) => {

try {

const {

name,

pincode,

phone,

emergency,

email,

address,

state,

district,

password,

confirmPassword

} = req.body;

console.log(req.body);

console.log(password);

console.log(confirmPassword);

if (password === confirmPassword) {

const hospital = new HospitalRegister({

name,

pincode,

phone,

emergency,

email,

address,

state,

district,

password,

confirmPassword

});

await hospital.save();

console.log("Saved successfully");

res.status(201).json({ message: "Registration Successful" });

} else {

//log error.

console.log("Not Saving.")

res.status(500).json({ error: error });

}

} catch (error) {

res.status(500).json({ error: error });

}

});

//Login

app.post('/login',async(req,res)=>{

try{

const{email,password}=req.body;

const login=await UserRegister.findOne({email});

if(!login || login.password !== password){

return res.status(401).json({error:'invalid username or password'});

}

res.status(200).json({message:login.toJSON()});

}

catch(error){

res.status(500).json({error:"Login Failed"});

}

})

//Ambulance Register

app.post("/ambulancereg", async (req, res) => {

try {

const {

name,

phone,

email,

carno,

drivername,

driverphone

} = req.body;

console.log(req.body);

console.log(email);

const ambulance = new AmbulanceRegister({

name,

phone,

email,

carno,

drivername,

driverphone

});

await ambulance.save();

console.log("Saved successfully");

res.status(201).json({ message: "Registration Successful" });

} catch (error) {

res.status(500).json({ error: error });

}

});

app.listen(Port, () => {

console.log("server is listening");

});

Chapter

Snapshot of Forms

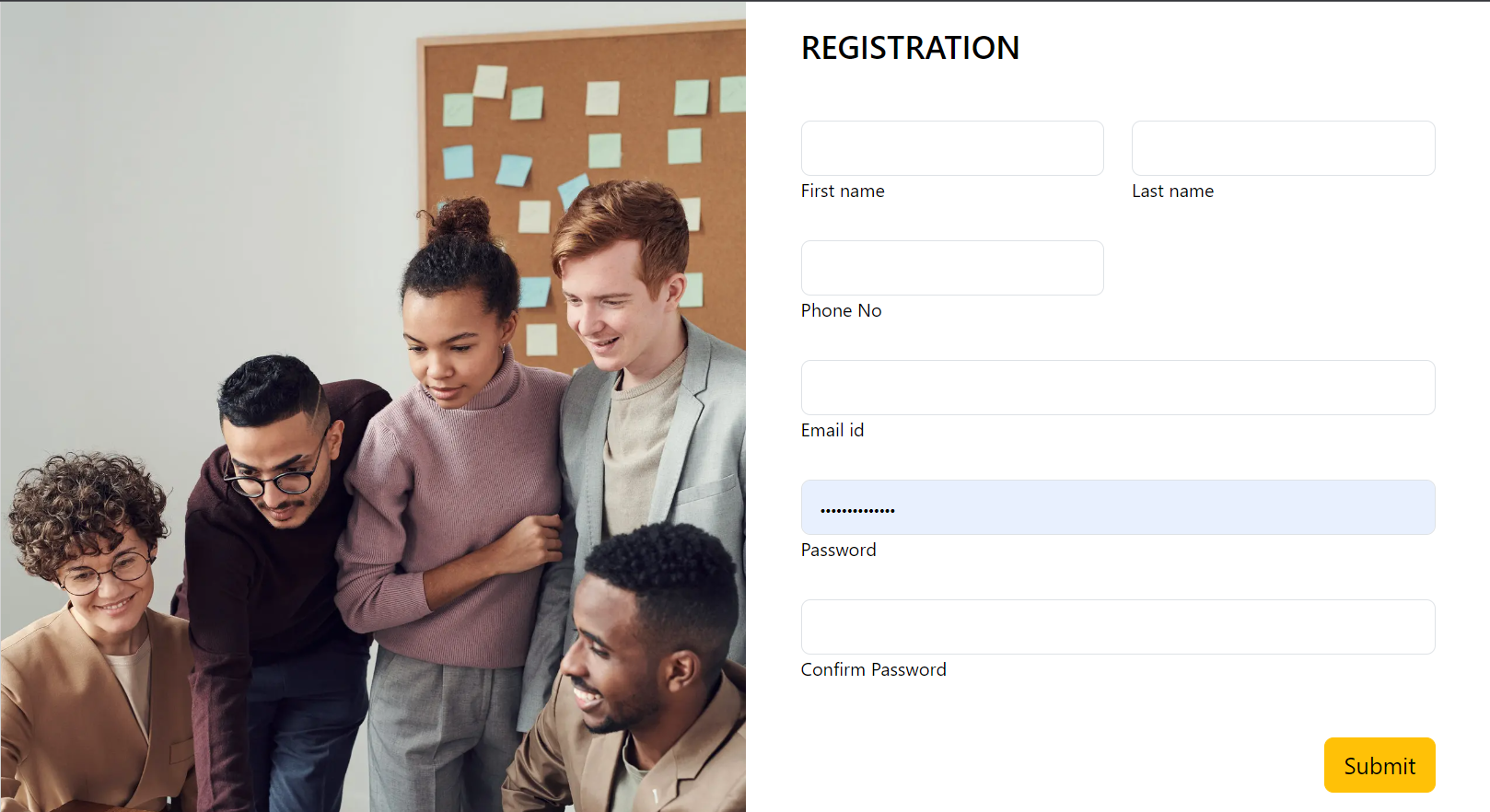
Home Page



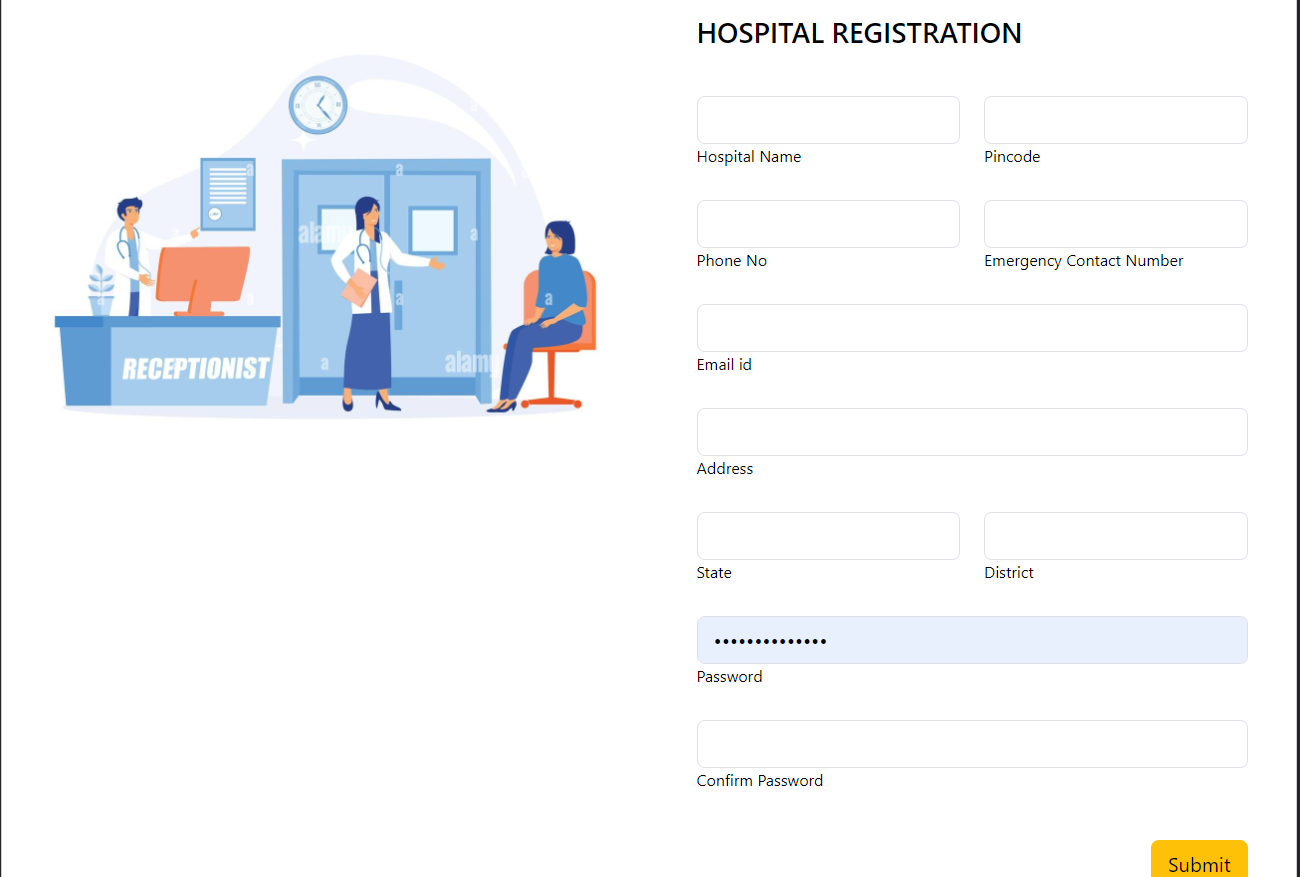
Login Page For User



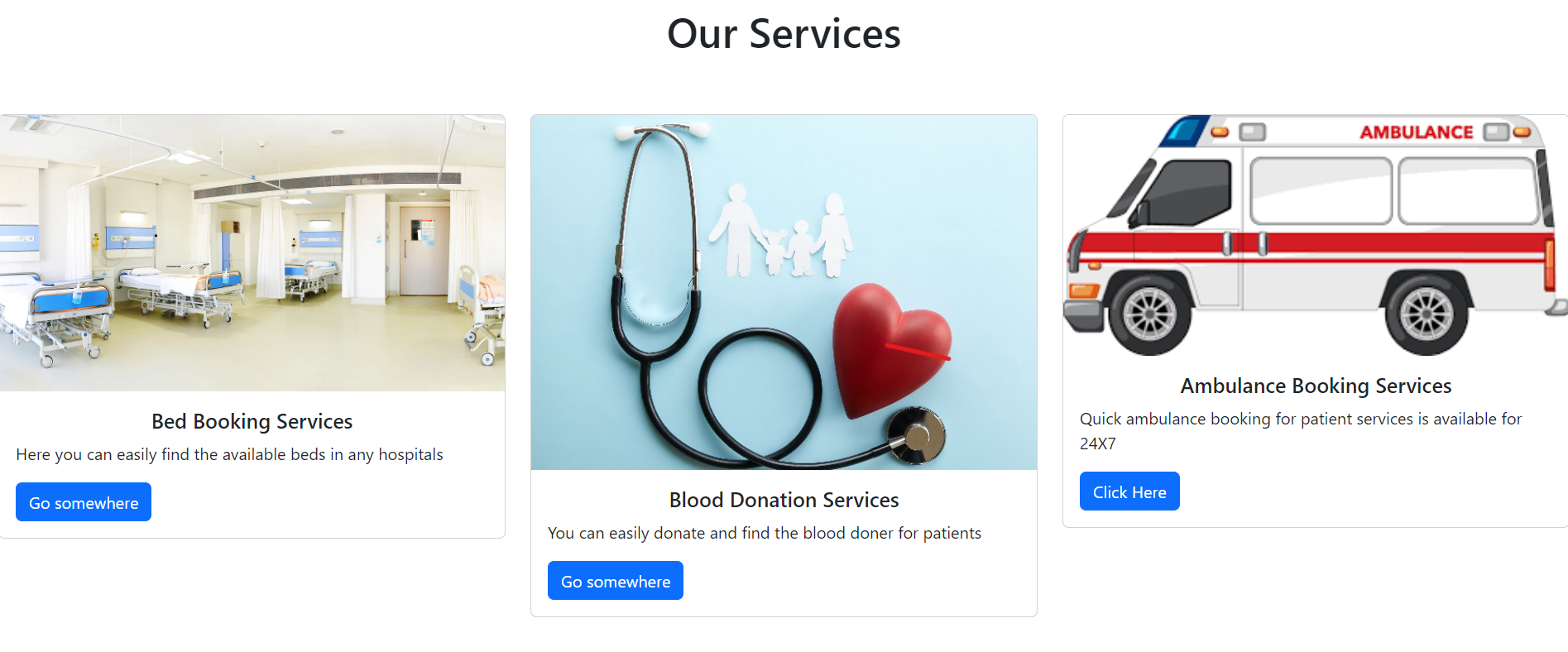
New User Registration Form

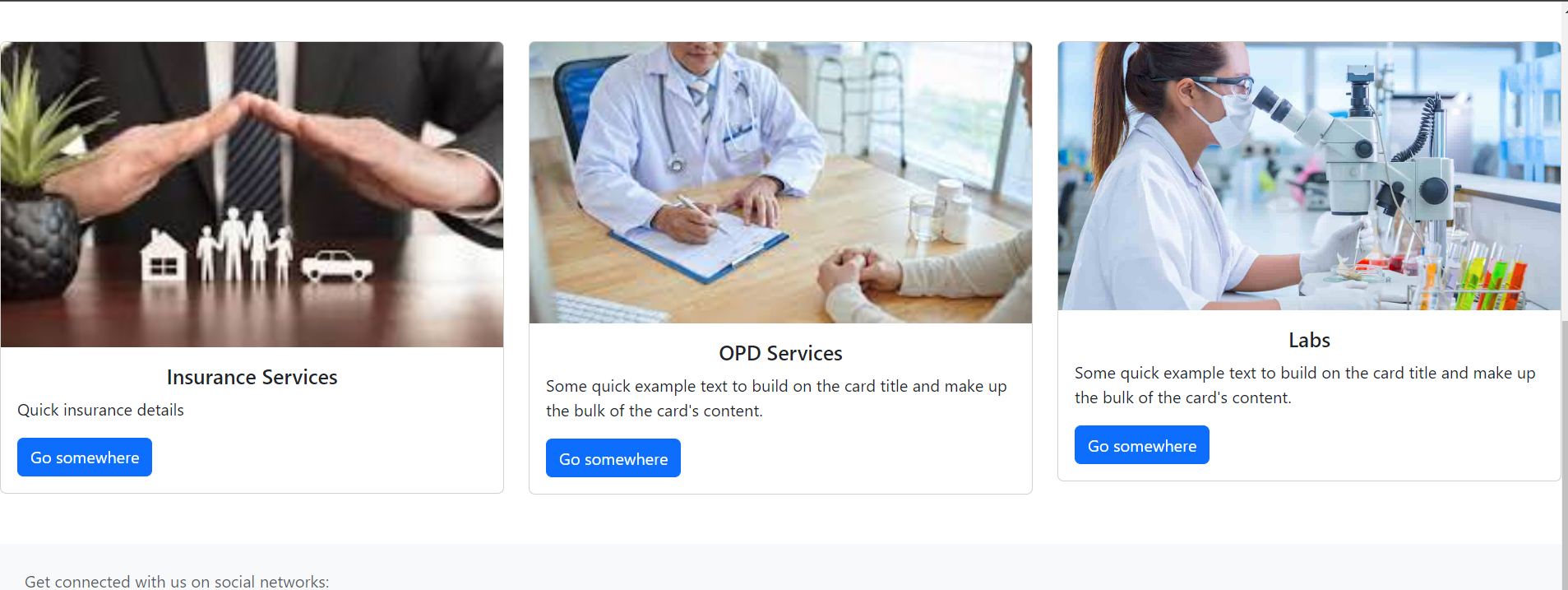
****

**Hospital Registration Form**

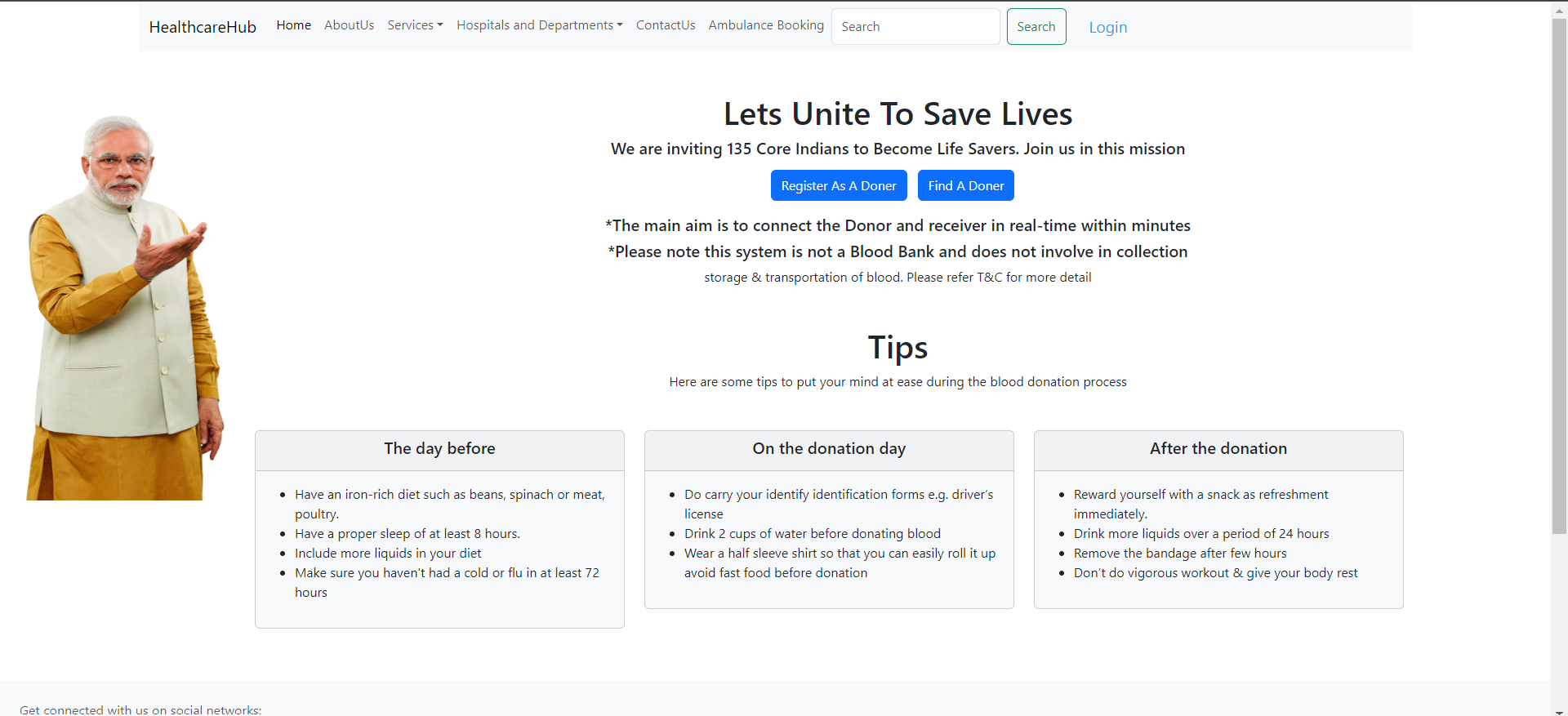
****

**Services**

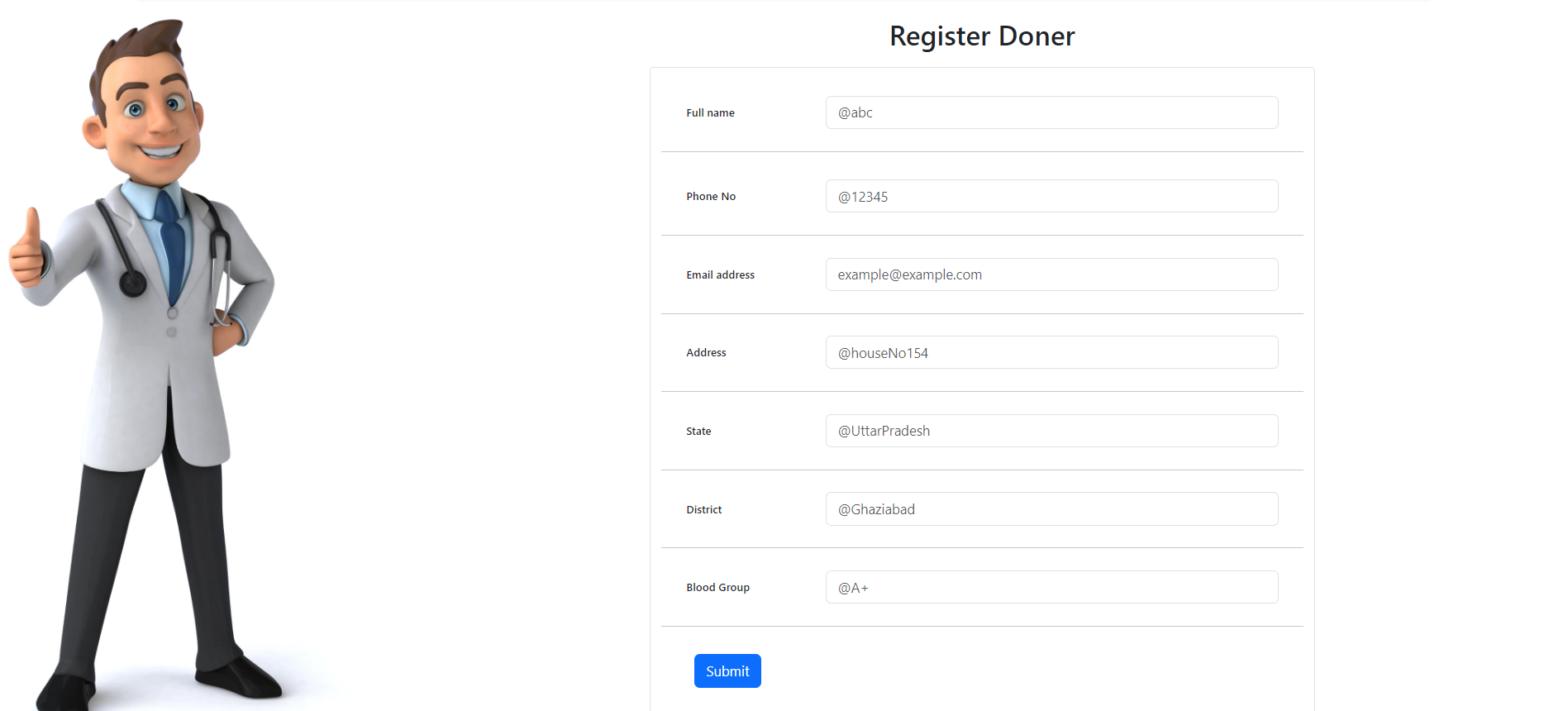
****

****

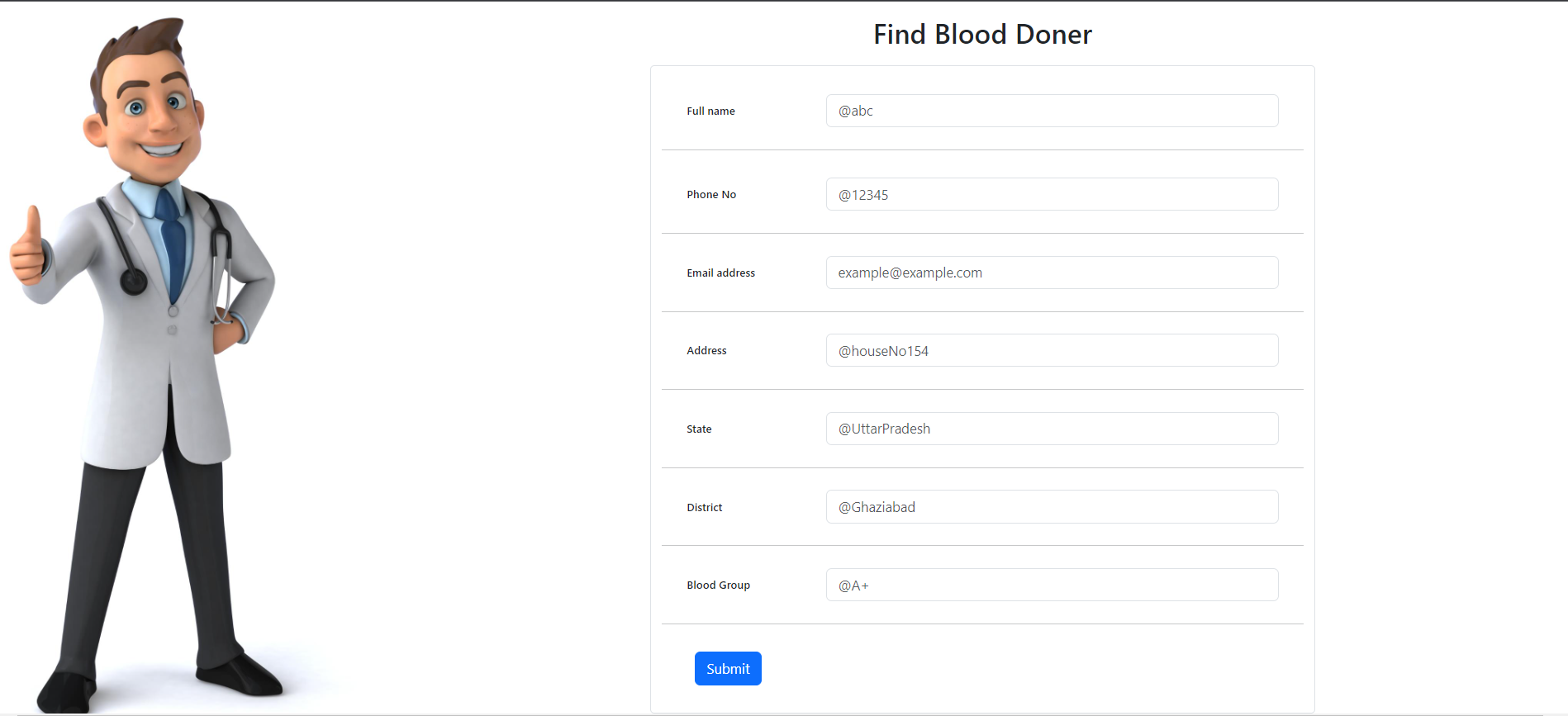
**Blood Donation**

****

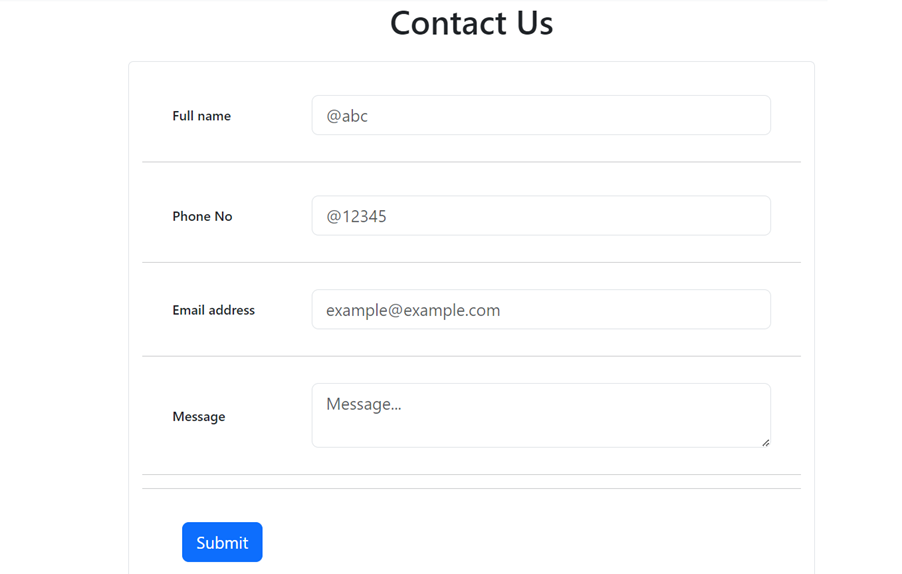
**Doner Register**

****

**Find Doner**

****

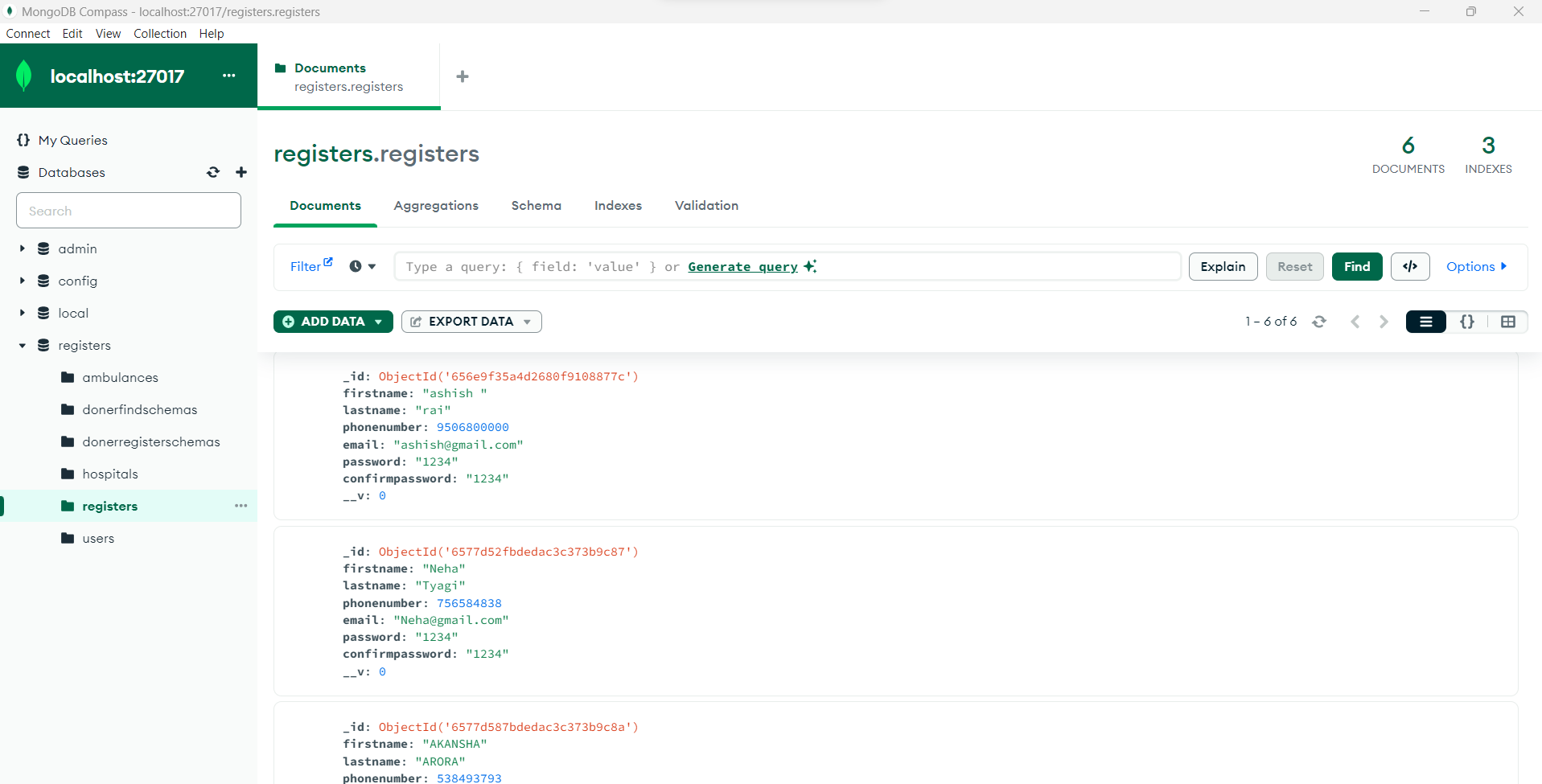
**Contact Us Page**



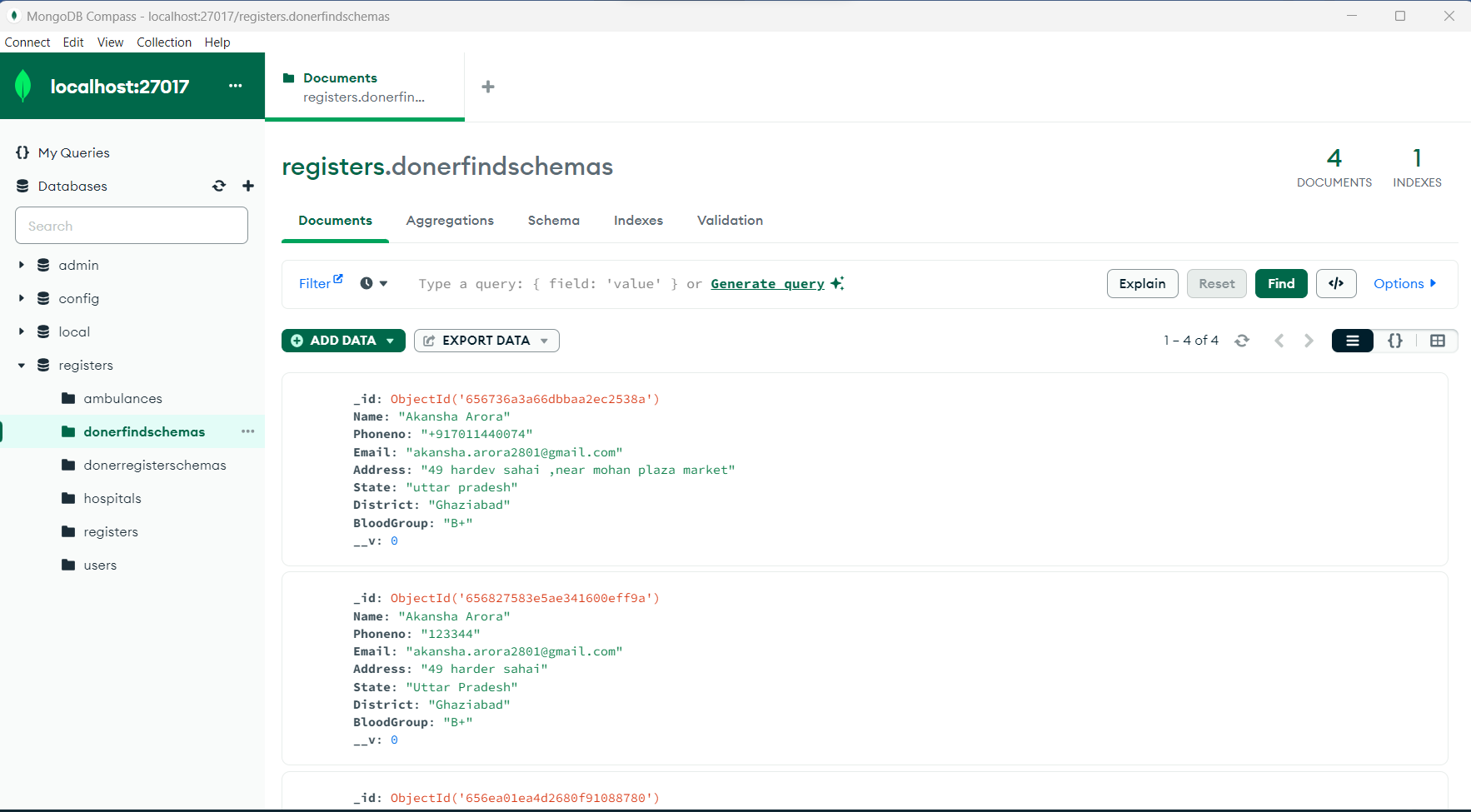
**Chapter**

**Snapshot of Database**

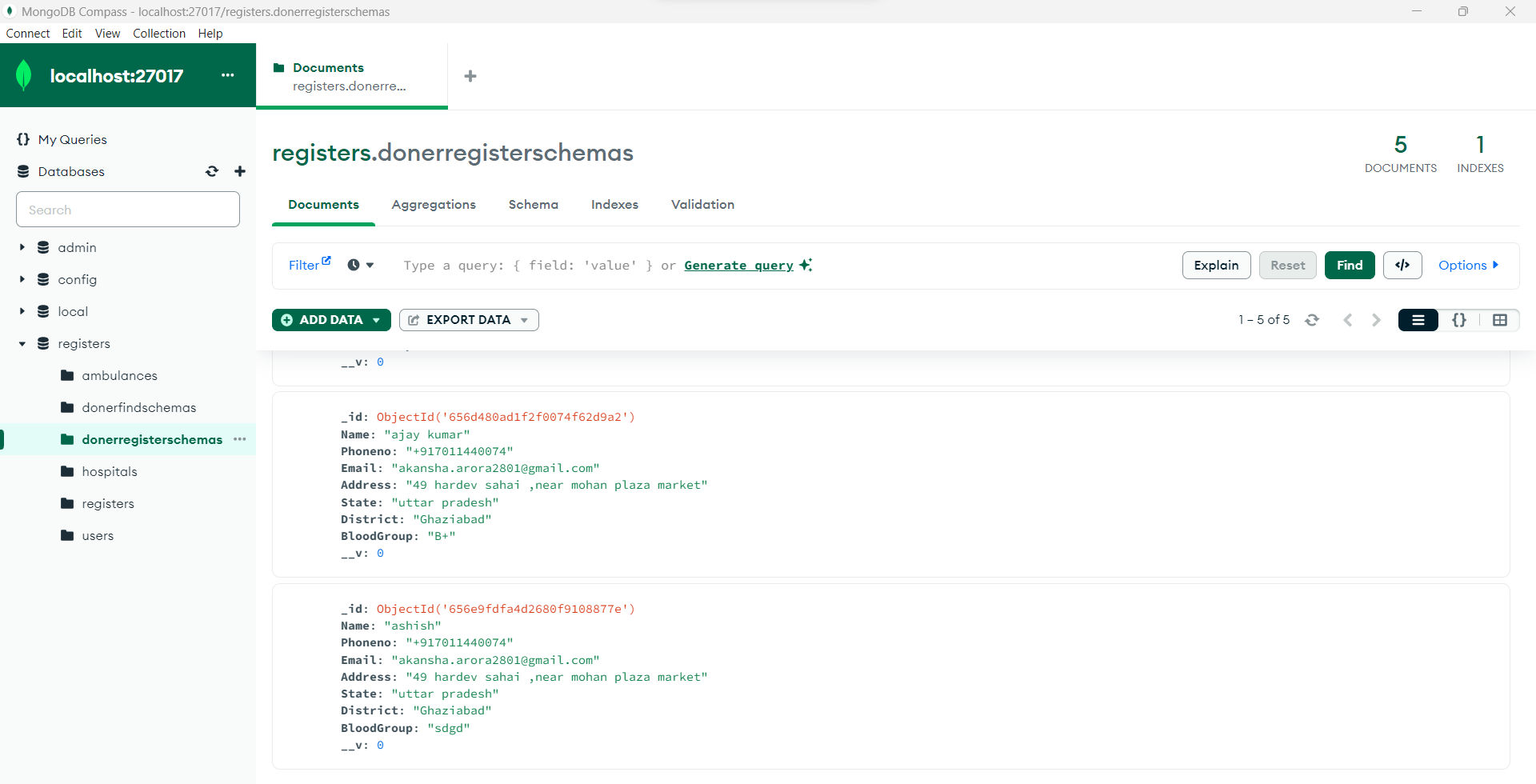
**User Register Database**

****

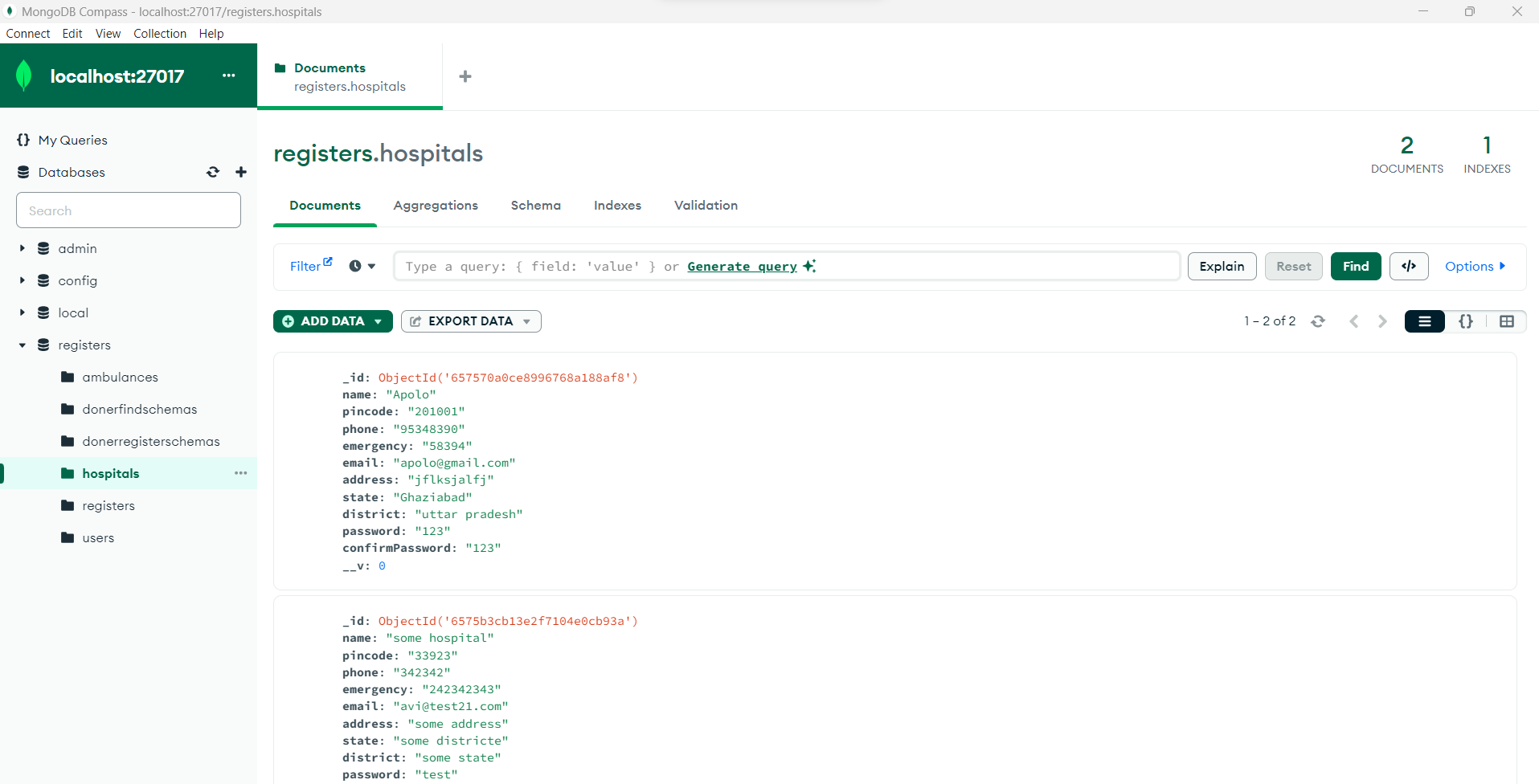
**Finding Doner Database**



**Doner Register Database**

****

**Hospital Registration Database**

****

# Chapter:-

**Reference**

## Google.

* + [**https://www.w3schools.com/**](https://www.w3schools.com/)
  + [**https://react.dev/**](https://react.dev/)
  + [**https://www.mongodb.com/try/download/community**](https://www.mongodb.com/try/download/community)
  + [**https://nodejs.org/en/learn/getting-started/introduction-to-nodejs**](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)

1. **You tube.**

