Doctor-Patient-Appointment:

1)backend:

1)verifyToken.js:

import jwt from 'jsonwebtoken';

import Doctor from '../models/DoctorSchema.js';

import User from '../models/UserSchema.js';

// 🔐 Middleware to authenticate token

export const authenticate = async (req, res, next) => {

  const authHeader = req.headers.authorization;

  // Check if token exists and starts with Bearer

  if (!authHeader || !authHeader.startsWith('Bearer ')) {

    return res

      .status(401)

      .json({ success: false, message: 'No token, authorization denied' });

  }

  const token = authHeader.split(' ')[1];

  try {

    const decoded = jwt.verify(token, process.env.JWT\_SECRET\_KEY);

    // Store the decoded userId and role in request object

    req.userId = decoded.id;

    req.role = decoded.role;

    console.log('Authenticated user:', req.userId, 'Role:', req.role);

    next();

  } catch (err) {

    if (err.name === 'TokenExpiredError') {

      return res.status(401).json({ success: false, message: 'Token has expired' });

    }

    return res.status(401).json({ success: false, message: 'Invalid token' });

  }

};

// ⛔ Middleware to restrict access to specific roles

export const restrict = (roles) => {

  return async (req, res, next) => {

    const userId = req.userId;

    let user;

    try {

      const patient = await User.findById(userId);

      const doctor = await Doctor.findById(userId);

      if (patient) user = patient;

      if (doctor) user = doctor;

      // If user is not found or role is not authorized

      if (!user || !roles.includes(user.role)) {

        return res

          .status(403)

          .json({ success: false, message: 'Access denied' });

      }

      // Add user to request for downstream use if needed

      req.user = user;

      next();

    } catch (err) {

      return res.status(500).json({ success: false, message: 'Server error during role check' });

    }

  };

};

2)authController:

import User from '../models/UserSchema.js'

import Doctor from '../models/DoctorSchema.js'

import jwt from 'jsonwebtoken'

import bcrypt from 'bcryptjs'

const generateToken = user=>{

    return jwt.sign({id:user.\_id, role:user.role}, process.env.JWT\_SECRET\_kEY, {

        expiresIn: "15d",

    })

}

export const register = async(req,res)=> {

    const {email, password, name, role, photo, gender} = req.body

    try {

        let user= null

        if(role==='patient')

        {

            user = await User.findOne({email})

        }

        else if(role==='doctor'){

            user = await Doctor.findOne({email})

        }

        if(user){

            return res.status(400).json({message: 'User already exist.'})

        }

        const salt = await bcrypt.genSalt(10)

        const hashPassword = await bcrypt.hash(password, salt)

        if(role==='patient'){

            user = new User({

                name,

                email,

                password:hashPassword,

                photo,

                gender,

                role

            })

        }

         if(role==='doctor'){

            user = new Doctor({

                name,

                email,

                password:hashPassword,

                photo,

                gender,

                role,

                isApproved: 'pending'

            })

        }

        await user.save()

        res.status(200).json({success:true, message:'User successfully created'})

    } catch (error) {

        res.status(500).json({success:false, message:'Internal server error, Try again' })

    }

};

export const login = async(req,res)=> {

    const{email, password,} = req.body

    try {

        let user =  null

        const patient = await User.findOne({email})

        const doctor = await Doctor.findOne({email})

        if(patient){

            user = patient

        }

        if(doctor){

            user = doctor

        }

        if(!user){

            return res.status(404).json({message: "User Not Found"})

        }

        const isPasswordMatch =await bcrypt.compare(req.body.password, user.password)

        if(!isPasswordMatch){

            return res.status(400).json({status: false, message: "Invalid Credentials"})

        }

        const token = generateToken(user)

        const {password, role, appointments, ...rest} = user.\_doc

        res.status(200).json({status: true, message: "Successfully Login", token, data:{...rest, isApproved: user.isApproved,}, role})

    } catch (error)

    {

        res.status(500).json({success:false, message:'Failed to Login' })

    }

};

3)doctorController.js:

import Booking from '../models/BookingSchema.js';

import Doctor from '../models/DoctorSchema.js'

export const updateDoctor = async(req, res)=>{

    const id = req.params.id

    try{

    const updatedDoctor = await Doctor.findByIdAndUpdate(id, {$set:req.body}, {new:true})

    res.status(200).json({success:true, message:'Successfully updated', data:updateDoctor})

}catch{

    res.status(500).json({success:false, message:'Failed to update'})

}

};

export const deleteDoctor = async(req, res)=>{

    const id = req.params.id

    try{

    await Doctor.findByIdAndDelete(id)

    res.status(200).json({success:true, message:'Successfully deleted',})

}catch{

    res.status(500).json({success:false, message:'Failed to delete'})

}

};

export const getSingleDoctor = async(req, res)=>{

    const id = req.params.id

    try{

    const doctor = await Doctor.findById(id).populate("reviews").select("-password")

    res.status(200).json({success:true, message:'Doctor Found', data: doctor,})

}catch{

    res.status(404).json({success:false, message:'No Doctor Found'})

}

};

export const getAllDoctor = async(req, res)=>{

    try{

        const { query } = req.query;

        const doctors = query

        ? await Doctor.find({

      isApproved: 'approved',

      $or: [

        { name: { $regex: query, $options: 'i' } },

        { specialization: { $regex: query, $options: 'i' } }

      ]

    }).select("-password")

  : await Doctor.find({}).select("-password");

    res.status(200).json({success:true, message:'Doctors Found', data: doctors,})

}catch{

    res.status(404).json({success:false, message:'Not Found'})

}

};

export const getDoctorByProfile = async(req, res)=>{

    const doctorId = req.userId

        try {

            const doctor = await Doctor.findById(doctorId)

            if(!doctor){

                return res.status(404).json({successfull:false, message:'Doctor not found'})

            }

            const {password, ...rest} = doctor.\_doc;

            const appointments=await Booking.find({doctor: doctorId})

            res.status(200).json({success:true, message:'Profile info is getting', data:{...rest, appointments},})

        } catch (error) {

            res.status(500).json({success:false, message:"Something went wrong, cannot get"})

        }

    };

4)reviewController.js:

import Review from '../models/ReviewSchema.js'

import Doctor from '../models/DoctorSchema.js'

export const getAllReview = async (req, res) => {

  try {

    const reviews = await Review.find({})

    res.status(200).json({ success: true, message: "Successful", data: reviews })

  } catch (error) {

    res.status(404).json({ success: false, message: "Not Found" })

  }

};

export const createReview = async (req, res) => {

  if (!req.body.doctor) req.body.doctor = req.params.doctorId

  if (!req.body.user) req.body.user = req.userId // this should be set from the token

  const newReview = new Review(req.body)

  try {

    const savedReview = await newReview.save()

    await Doctor.findByIdAndUpdate(req.body.doctor, {

      $push: { reviews: savedReview.\_id }

    })

    res.status(200).json({ success: true, message: 'Review submitted', data: savedReview })

  } catch (err) {

    res.status(500).json({ success: false, message: err.message })

  }

};

5)userController.js:

import User from '../models/UserSchema.js'

import Booking from '../models/BookingSchema.js'

import Doctor from '../models/DoctorSchema.js'

export const updateUser = async(req, res)=>{

    const id = req.params.id

    try{

    const updatedUser = await User.findByIdAndUpdate(id, {$set:req.body}, {new:true})

    res.status(200).json({success:true, message:'Successfully updated', data:updatedUser})

}catch{

    res.status(500).json({success:false, message:'Failed to updated'})

}

};

export const deleteUser = async(req, res)=>{

    const id = req.params.id

    try{

    await User.findByIdAndDelete(id)

    res.status(200).json({success:true, message:'Successfully deleted',})

}catch{

    res.status(500).json({success:false, message:'Failed to delete'})

}

};

export const getSingleUser = async(req, res)=>{

    const id = req.params.id

    try{

    const user = await User.findById(id).select("-password")

    res.status(200).json({success:true, message:'User Found', data: user,})

}catch{

    res.status(404).json({success:false, message:'No User Found'})

}

};

export const getAllUser = async(req, res)=>{

    try{

    const users = await User.find({}).select("-password")

    res.status(200).json({success:true, message:'Users Found', data: users,})

}catch{

    res.status(404).json({success:false, message:'Not Found'})

}

};

export const getUserProfile = async(req, res)=>{

    const userId = req.userId

    try {

        const user = await User.findById(userId)

        if(!user){

            return res.status(404).json({successfull:false, message:'User not found'})

        }

        const {password, ...rest} = user.\_doc

        res.status(200).json({success:true, message:'Profile info is getting', data:{...rest}})

    } catch (error) {

        res.status(500).json({success:false, message:"Something went wrong, cannot get"})

    }

};

export const getMyAppointments = async(req,res)=>{

    try {

        const bookings = await Booking.find({user:req.userId})

        const doctorIds = bookings.map(el=>el.doctor.id)

        const doctors = await Doctor.find({\_id:{$in:doctorIds}}).select('-password')

        res.status(200).json({success:true, message:'Appointments are getting', data:doctors})

    } catch (error) {

        res.status(500).json({success:false, message:"Something went wrong, cannot get"})

    }

}

6)bookingSchema.js:

import mongoose from "mongoose";

const bookingSchema = new mongoose.Schema(

  {

    doctor: {

      type: mongoose.Types.ObjectId,

      ref: "Doctor",

      required: true,

    },

    user: {

      type: mongoose.Types.ObjectId,

      ref: "User",

      required: true,

    },

    ticketPrice: { type: String, required: true },

    appointmentDate: {

      type: Date,

      required: true,

    },

    status: {

      type: String,

      enum: ["pending", "approved", "cancelled"],

      default: "pending",

    },

    isPaid: {

      type: Boolean,

      default: true,

    },

  },

  { timestamps: true }

);

export default mongoose.model("Booking", bookingSchema);

7)doctorSchema.js:

import mongoose from "mongoose";

const DoctorSchema = new mongoose.Schema({

  email: { type: String, required: true, unique: true },

  password: { type: String, required: true },

  name: { type: String, required: true },

  phone: { type: Number },

  photo: { type: String },

  ticketPrice: { type: Number },

  role: {

    type: String,

  },

  // Fields for doctors only

  specialization: { type: String },

  qualifications: {

    type: Array,

  },

  experiences: {

    type: Array,

  },

  bio: { type: String, maxLength: 50 },

  about: { type: String },

  timeSlots: { type: Array },

  reviews: [{ type: mongoose.Types.ObjectId, ref: "Review" }],

  averageRating: {

    type: Number,

    default: 0,

  },

  totalRating: {

    type: Number,

    default: 0,

  },

  isApproved: {

    type: String,

    enum: ["pending", "approved", "cancelled"],

    default: "pending",

  },

  appointments: [{ type: mongoose.Types.ObjectId, ref: "Appointment" }],

});

export default mongoose.model("Doctor", DoctorSchema);

8)reviewSchema.js:

import mongoose from "mongoose";

import DoctorSchema from "./DoctorSchema.js";

const reviewSchema = new mongoose.Schema(

  {

    doctor: {

      type: mongoose.Types.ObjectId,

      ref: "Doctor",

    },

    user: {

      type: mongoose.Types.ObjectId,

      ref: "User",

    },

    reviewText: {

      type: String,

      required: true,

    },

    rating: {

      type: Number,

      required: true,

      min: 0,

      max: 5,

      default: 0,

    },

  },

  { timestamps: true }

);

reviewSchema.pre(/^find /,function(next){

  this.populate({

    path:'user',

    select:'name photo',

  });

});

reviewSchema.statics.calcAverageRatings = async function(doctorId){

  const stats = await this.aggregate([{

    $match:{doctor:doctorId}

  },

{

  $group:{

    \_id:'$doctor',

    numOfRating:{$sum:1},

    avgRating:{$avg:'$rating'}

  }

}])

await DoctorSchema.findByIdAndUpdate(doctorId, {

  totalRating: stats[0].numOfRating,

  averageRating: stats[0].avgRating,

})

}

reviewSchema.post('save', function(){

  this.constructor.calcAverageRatings(this.doctor)

})

export default mongoose.model("Review", reviewSchema);

9)userSchema.js:

import mongoose from "mongoose";

const UserSchema = new mongoose.Schema({

  email: { type: String, required: true, unique: true },

  password: { type: String, required: true },

  name: { type: String, required: true },

  phone: { type: Number },

  photo: { type: String },

  role: {

    type: String,

    enum: ["patient", "admin"],

    default: "patient",

  },

  gender: { type: String, enum: ["male", "female", "other"] },

  bloodType: { type: String, required: false },

  appointments: [{ type: mongoose.Types.ObjectId, ref: "Appointment" }],

});

export default mongoose.model("User", UserSchema);

10)auth.js:

import express from 'express'

import { register, login } from '../Controllers/authController.js'

const router = express.Router()

router.post('/register', register)

router.post('/login', login)

export default router

11)doctor.js:

import express from 'express'

import { updateDoctor, deleteDoctor, getAllDoctor, getSingleDoctor, getDoctorByProfile } from "../Controllers/doctorController.js";

import { authenticate, restrict } from '../auth/verifyToken.js';

import reviewRouter from './review.js'

const router = express.Router()

router.use("/:doctorId/reviews", reviewRouter);

router.get('/', getAllDoctor)

router.get('/:id', getSingleDoctor)

router.put('/:id', authenticate, restrict(['doctor']), updateDoctor)

router.delete('/:id', authenticate, restrict(['doctor']), deleteDoctor)

router.get('/profile/me', authenticate, restrict(["doctor"]), getDoctorByProfile)

export default router;

12)review.js:

import express from 'express';

import { getAllReview, createReview } from '../Controllers/reviewController.js';

import { authenticate, restrict } from '../auth/verifyToken.js';

const router = express.Router({ mergeParams: true });

router.route('/')

  .get(getAllReview)

  .post(authenticate, restrict(['patient']), createReview);

export default router;

13)user.js:

import express from 'express'

import { updateUser, deleteUser, getAllUser, getSingleUser,

    getUserProfile, getMyAppointments

 } from "../Controllers/userController.js";

import { authenticate, restrict } from '../auth/verifyToken.js';

const router = express.Router()

router.get('/', authenticate, restrict(['admin']), getAllUser)

router.get('/:id', authenticate, restrict(['patient']), getSingleUser)

router.put('/:id',  authenticate, restrict(['patient']), updateUser)

router.delete('/:id',  authenticate, restrict(['patient']), deleteUser)

router.get('/profile/me',  authenticate, restrict(['patient']), getUserProfile)

router.get('/appointments/my-appointments',  authenticate, restrict(['patient']), getMyAppointments)

export default router;

14)generateSecret.js:

import { randomBytes } from 'crypto'

const secret = randomBytes(256).toString('base64')

console.log(secret)

15)index.js:

import express from "express"

import cookieParser from "cookie-parser"

import cors from "cors"

import mongoose from "mongoose"

import dotenv from "dotenv"

import authRoute from './Routes/auth.js'

import userRoute from './Routes/user.js'

import doctorRoute from './Routes/doctor.js'

import reviewRoute from './Routes/review.js'

dotenv.config()

const app= express()

const port = process.env.PORT || 8000

const corsOptions={

    origin:true

}

app.get('/', (req,res)=>{

    res.send('Api is Working')

})

//database connection

mongoose.set('strictQuery', false)

const connectDB = async()=>{

    try{

        mongoose.connect(process.env.MONGO\_URL,

            { useNewUrlParser: true,

            useUnifiedTopology: true

        })

        console.log("MongoDB database is connected")

    }catch(err){

        console.log("MongoDB database connection failed", err.message);

    }

}

//middleware

app.use(express.json())

app.use(cookieParser())

app.use(cors(corsOptions))

app.use('/api/v1/auth', authRoute)

app.use('/api/v1/users', userRoute)

app.use('/api/v1/doctors', doctorRoute)

app.use('/api/v1/reviews', reviewRoute)

app.listen(port, ()=>{

    connectDB();

    console.log("Server is Running on port" + port)

})

Frontend:

1)About.jsx:

import React from 'react';

import aboutImg from '/src/assets/images/about.png';

import aboutCarding from '/src/assets/images/about-card.png';

import { Link } from 'react-router-dom'

const About = () => {

  return (

    <section>

      <div className='container'>

        <div className='flex justify-between gap-[50px] lg:gap-[130px] xl:gap-0 flex-col lg:flex-row'>

          {/\* Main Image (Kept as Original) \*/}

          <div className="relative w-3/4 lg:w-1/2 xl:w-[770px] z-10 order-2 lg:order-1">

            <img src={aboutImg} alt="About Section"/>

            {/\* Adjusted Overlay Card \*/}

            <div className='absolute z-[50] bottom-[10%] left-[10%] w-[180px] md:w-[280px]'>

              <img src={aboutCarding} alt="About Card" className='w-full' />

            </div>

          </div>

          <div className='w-full lg:w-1/2 xl:w-[670px] order-1 lg:order-2'>

            <h2 className='heading'>🌟Proud to be one of the nation's best!🏆✨Delivering excellence, innovation, and care every step of the way!🚀</h2>

            <p className='text\_\_para'>India has recognized MediCare as a top healthcare provider!🏆✨With world-class doctors, cutting-edge facilities, and a commitment to excellence, it sets new standards in patient care!🏥🩺🚀</p>

            <Link to='/'>

            <button className='btn'>Learn More</button>

            </Link>

          </div>

        </div>

      </div>

    </section>

  );

};

export default About;

2)DoctorCard.js:

import React from 'react'

import starIcon from '/src/assets/images/Star.png'

import{Link} from 'react-router-dom'

import { BsArrowRight } from 'react-icons/bs';

const DoctorCard = ({doctor}) => {

    const {name, avgRating,totalRating, photo, specialization, totalPatients, hospital} = doctor

  return (

    <div className='p-3 lg:p-5'>

      <div>

        <img src={photo} className='w-full'></img>

      </div>

    <h2 className='text-[18px] leading-[30px] lg:text-[26px] lg:leading-9 text-headingColor font-[700] mt-3 lg:mt-5'>{name}</h2>

    <div className='mt-2 lg:mt-4 flex items-center justify-between'>

        <span className='bg-[#CCF0F3] text-irisBlueColor py-1 px-2 lg:py-2 lg:px-6 text-[12px] leading-4 lg:text-[16px] lg:leading-7 font-semibold rounded'>

            {specialization}

        </span>

        <div className='flex items-center gap-[6px]'>

            <span className='flex items-center gap-[6px] text-[14px] leading-6 lg:text-[16px] lg:leading-7 font-semibold text-headingColor'>

                <img src={starIcon} />{avgRating}

            </span>

            <span className='text-[14px] leading-6 lg:text-[16px] lg:leading-7 font-[400] text-textColor'>({totalRating})</span>

        </div>

    </div>

        <div className='mt-[18px] lg:mt-5 flex items-centyer justify-between'>

            <div>

                <h3 className='text-[16px] leading-7 lg:text-[18px] lg:leading-[30px]  font-semimbold text-headingCOlor'>+{totalPatients} patients</h3>

                <p className='text-[14px] leading-6 font-[400] text-textColor'>At {hospital}</p>

            </div>

            <Link to='/doctors' className='w-[44px] rounded-full border border-solid border-[#181A1E] mt-[30px] mx-auto flex items-center justify-center group hover:bg-primaryColor hover:border-none'>

                <BsArrowRight className='group-hover:text-white w-6 h-5'/>

            </Link>

        </div>

    </div>

  )

}

export default DoctorCard

3)DoctorList.js:

import React from 'react'

import { doctors } from './../../assets/data/doctors'

import DoctorCard from './DoctorCard'

const DoctorList = () => {

  return (

    <div className='grid grid-cols-1 sm:grid -cols-2 md:grid-cols-3 gap-5 lg:gap-[30px] mt-[30px]

    lg:mt-[55px]'>{doctors.map((doctor)=>

    <DoctorCard key={doctor.id} doctor={doctor}/>

    )}

    </div>

  )

}

export default DoctorList

4)error.jsx:

import React from 'react'

const Error = ({errMessage}) => {

  return (

    <div className='flex items-center justify-center w-full h-full'>

      <h3 className='text-headingColor text-[20px] leadin-[30px] font-semibold'>{errMessage}</h3>

    </div>

  )

}

export default Error

5)faqitem.js:

import React from 'react'

import { useState } from 'react'

import { AiOutlineMinus, AiOutlinePlus } from 'react-icons/ai'

const FaqItem = ({item}) => {

  const [isOpen, setIsOpen] = useState(false)

  const toggleAccordion = ()=> {

    setIsOpen(!isOpen);

  }

  return (

    <div className='p-3 lg:p-5 rounded-[12px] border border-solid border-[#D9DCE2] mb-5 cursor-pointer'>

      <div className='flex items-center justify-between gap-5' onClick={toggleAccordion}>

        <h4 className='text-[16px] leading-7 lg:text-[22px] lg:leading-8 text-headingColor'>{item.question}</h4>

        <div className={`${isOpen && 'bg-primaryColor text-white border-none'} w-7 h-7 lg:w-8 lg:h-8 border border-solid border-[#141F21] rounded flex items-center justify-center `}>

          {isOpen ? <AiOutlineMinus /> : <AiOutlinePlus />}

        </div>

      </div>

      {isOpen &&  <div className='mt-4'>

        <p className='text-[14px] leading-6 lg:text-[16px] lg:leading-7 font-[400] text-textColor'>{item.content}</p>

        </div>}

    </div>

  )

}

export default FaqItem

6)faqlist.jsx:

import React from 'react'

import FaqItem from './FaqItem'

import { faqs } from './../../assets/data/faqs'

const FaqList = () => {

  return (

     <ul className='mt-[38px]'>

        {faqs.map((item,index)=>(

        <FaqItem item={item} key={index} />

        ))}

    </ul>

  );

};

export default FaqList;

7)footer.jsx:

import React from 'react';

import { Link } from 'react-router-dom';

import logo from '/src/assets/images/logo.png';

import { RiLinkedinFill } from 'react-icons/ri';

import { AiFillGithub } from 'react-icons/ai';

const socialLinks = [

  {

    path: "https://www.linkedin.com/in/krishika-v-7092b8335",

    icon: <RiLinkedinFill className='group-hover:text-white w-4 h-5' />,

  },

  {

    path: "https://github.com/krish-dotcom518",

    icon: <AiFillGithub className='group-hover:text-white w-4 h-5' />,

  },

];

const quickLinks01 = [

  { path: '/home', display: "Home" },

  { path: '/', display: "About Us" },

  { path: '/services', display: "Services" },

  { path: '/', display: "Blog" },

];

const quickLinks02 = [

  { path: '/find-a-doctor', display: "Find a Doctor" },

  { path: '/', display: "Request an Appointment" },

  { path: '/', display: "Find a Location" },

  { path: '/', display: "Get an Opinion" },

];

const quickLinks03 = [

  { path: '/', display: "Donate" },

  { path: '/contact', display: "Contact Us" },

];

const Footer = () => {

  const year = new Date().getFullYear();

  return (

    <footer className='pb-16 pt-10 bg-gray-100'>

      <div className='container'>

        <div className='flex justify-between flex-col md:flex-row flex-wrap gap-[30px]'>

          {/\* Logo & Social Links \*/}

          <div>

            <img src={logo} alt="Logo" className='w-24' />

            <p className='text-[16px] leading-7 font-[400] text-textColor mt-4'>

              Copyright © {year} developed by Krishika Venkatesan. All rights reserved.

            </p>

            {/\* Social Icons - Side by Side \*/}

            <div className='flex gap-4 mt-4'>

              {socialLinks.map((link, index) => (

                <Link

                  to={link.path}

                  key={index}

                  className='w-9 h-9 border border-solid border-[#181A1E] rounded-full flex items-center justify-center group hover:bg-primaryColor hover:border-none'

                >

                  {link.icon}

                </Link>

              ))}

            </div>

          </div>

          {/\* Quick Links \*/}

          <div className='grid grid-cols-3 gap-[20px]'>

            {/\* Column 1 \*/}

            <ul>

              {quickLinks01.map((link, index) => (

                <li key={index}>

                  <Link to={link.path} className='text-[16px] font-[500] text-textColor hover:text-primaryColor'>{link.display}</Link>

                </li>

              ))}

            </ul>

            {/\* Column 2 \*/}

            <ul>

              {quickLinks02.map((link, index) => (

                <li key={index}>

                  <Link to={link.path} className='text-[16px] font-[500] text-textColor hover:text-primaryColor'>{link.display}</Link>

                </li>

              ))}

            </ul>

            {/\* Column 3 \*/}

            <ul>

              {quickLinks03.map((link, index) => (

                <li key={index}>

                  <Link to={link.path} className='text-[16px] font-[500] text-textColor hover:text-primaryColor'>{link.display}</Link>

                </li>

              ))}

            </ul>

          </div>

        </div>

      </div>

    </footer>

  );

};

export default Footer;

8)header.jsx:

import React, { useEffect, useRef, useContext } from 'react';

import { NavLink, Link } from 'react-router-dom';

import { BiMenu } from 'react-icons/bi';

import logo from '/src/assets/images/logo.png';

import { AuthContext } from '../../context/AuthContext';

const navLinks = [

  { path: '/home', display: 'Home' },

  { path: '/doctors', display: 'Find a Doctor' },

  { path: '/services', display: 'Services' },

  { path: '/contact', display: 'Contact' },

];

const Header = () => {

  const headerRef = useRef(null);

  const menuRef = useRef(null);

  const { user, role, token } = useContext(AuthContext);

  // Sticky header on scroll

  useEffect(() => {

    const handleScroll = () => {

      if (window.scrollY > 80) {

        headerRef.current?.classList.add('sticky\_\_header');

      } else {

        headerRef.current?.classList.remove('sticky\_\_header');

      }

    };

    window.addEventListener('scroll', handleScroll);

    return () => window.removeEventListener('scroll', handleScroll);

  }, []);

  const toggleMenu = () => {

    menuRef.current.classList.toggle('hidden');

  };

  return (

      <header

      ref={headerRef}

      className="w-full bg-white py-4 shadow-md absolute top-0 left-0 z-50"

    >

      <div className="container mx-auto px-4 flex items-center justify-between">

        {/\* Logo \*/}

        <Link to="/home" className="flex items-center gap-2">

          <img src={logo} alt="Logo" className="w-[140px]" />

        </Link>

        {/\* Navigation Links \*/}

        <nav className="hidden md:flex gap-8 m-0 p-0 items-center" ref={menuRef}>

          {navLinks.map((link, index) => (

            <NavLink

              key={index}

              to={link.path}

              className={({ isActive }) =>

                isActive

                  ? 'text-primaryColor font-semibold text-[16px]'

                  : 'text-gray-800 hover:text-primaryColor font-medium text-[16px]'

              }

            >

              {link.display}

            </NavLink>

          ))}

        </nav>

        {/\* Login / Profile \*/}

        <div className="flex items-center gap-4">

          {token && user ? (

            <Link

              to={`/${role === 'doctor' ? 'doctors/profile/me' : 'users/profile/me'}`}

              className="flex items-center gap-2"

            >

              <figure className="w-[35px] h-[35px] rounded-full overflow-hidden border border-gray-300">

                <img

                  src={user?.photo}

                  alt="User"

                  className="w-full h-full object-cover"

                />

              </figure>

              <span className="text-sm font-semibold text-gray-800">{user?.name}</span>

            </Link>

          ) : (

            <Link to="/login">

              <button className="bg-primaryColor text-white px-6 py-2 rounded-full font-semibold hover:bg-blue-600 transition">

                Login

              </button>

            </Link>

          )}

          {/\* Mobile Menu Icon \*/}

          <button onClick={toggleMenu} className="md:hidden">

            <BiMenu className="w-6 h-6 text-gray-800" />

          </button>

        </div>

      </div>

      {/\* Mobile Nav \*/}

      <div className="md:hidden px-4 mt-2">

        <ul

          ref={menuRef}

          className="hidden flex-col gap-3 bg-white p-4 border rounded shadow-md"

        >

          {navLinks.map((link, index) => (

            <li key={index}>

              <NavLink

                to={link.path}

                className="text-gray-800 hover:text-primaryColor block"

              >

                {link.display}

              </NavLink>

            </li>

          ))}

        </ul>

      </div>

    </header>

  );

};

export default Header;

9)loading.jsx:

import React from 'react'

import HashLoader from 'react-spinners/HashLoader'

const Loading = () => {

  return (

    <div className='flex items-center justify-center w-full h-full'>

        <HashLoader color='#0067FF'/>

    </div>

  );

};

export default Loading;

10)serviceCard.jsx:

import React from 'react'

import { BsArrowRight } from 'react-icons/bs';

import { Link } from 'react-router-dom'

const ServiceCard =({item,index})=>{

    const {name, desc, bgColor, textColor} = item

    return (

  <div className={`py-[30px] px-3 lg:px-5 ${bgColor} ${textColor}`}>

    <h2 className='text-[26px] leading-9 text-headingColor font-[700]'>{name}</h2>

    <p className='text-[16px] leading-7 font-[400] text-textColor mt-4'>{desc}</p>

    <div className='flex items-center justify-between mt-[30px]'>

        <Link to='/doctors' className='w-[44px] rounded-full border border-solid border-[#181A1E] flex items-center justify-center group hover:bg-primaryColor hover:border-none'>

                    <BsArrowRight className='group-hover:text-white w-6 h-5'/>

                  </Link>

                  <span className='w-[44px] h-[44px] flex items-center justify-center text-[18px] leading-[30px] font-[600]' style={{background:`${bgColor}`, color: `${textColor}`, borderRadius: "6px 0 0 6px"}}>

                    {index+1}

                  </span>

    </div>

  </div>

);

};

export default ServiceCard

11)serviceList.jsx:

import React from 'react'

import {services} from './../../assets/data/services';

import ServiceCard from './ServiceCard';

const ServiceList = () => {

  return <div className='grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-5 lg:gap-[30px] mt-[30px] lg:mt-[55px]'>

    {services.map((item,index)=>(

    <ServiceCard item={item} index={index} key={index}/>

    ))}

    </div>

}

export default ServiceList

12)testimonial.jsx:

import React from 'react';

import { Pagination } from 'swiper/modules';

import { Swiper, SwiperSlide } from 'swiper/react';

import 'swiper/css';

import 'swiper/css/pagination';

import patientAvatar from '/src/assets/images/patient-avatar.png';

import { HiStar } from 'react-icons/hi';

const testimonials = [

  {

    name: "Mathew Thomas",

    feedback: "The doctors here truly care! They listened to my concerns and provided the best treatment. I feel healthier than ever!",

  },

  {

    name: "Michael Tae",

    feedback: "Excellent service and compassionate staff. My recovery was smooth, thanks to their guidance!",

  },

  {

    name: "Nick jonas",

    feedback: "I was hesitant about my procedure, but the doctors reassured me at every step. Highly recommend!",

  },

];

const Testimonial = () => {

  return (

    <div className='mt-[30px] lg:mt-[55px]'>

      <Swiper

        modules={[Pagination]}

        spaceBetween={30}

        slidesPerView={1}

        pagination={{ clickable: true }}

        breakpoints={{

          640: { slidesPerView: 1, spaceBetween: 0 },

          768: { slidesPerView: 2, spaceBetween: 20 },

          1024: { slidesPerView: 2, spaceBetween: 30 }, // Showing 2 side-by-side

        }}

      >

        {testimonials.map((patient, index) => (

          <SwiperSlide key={index}>

            <div className='py-[30px] px-5 rounded-3 border border-gray-200 shadow-lg flex flex-col lg:flex-row gap-5'>

              <img src={patientAvatar} alt="Patient Avatar" className='w-16 h-16 rounded-full' />

              <div>

                <h4 className='text-[18px] leading-[30px] font-semibold text-headingColor'>{patient.name}</h4>

                <div className='flex gap-1 mt-1'>

                  {[...Array(5)].map((\_, starIndex) => (

                    <HiStar key={starIndex} className='text-yellow-500 text-lg' />

                  ))}

                </div>

                <p className='mt-2 text-textColor text-sm'>{patient.feedback}</p>

              </div>

            </div>

          </SwiperSlide>

        ))}

      </Swiper>

    </div>

  );

};

export default Testimonial;

13)Authcontext.jsx:

import { createContext, useEffect, useReducer } from "react";

const initialState = {

    user: localStorage.getItem('user')!= undefined ?

    JSON.parse(localStorage.getItem('user')) : null,

    role:localStorage.getItem('role') || null,

    token:localStorage.getItem('token') || null,

};

export const AuthContext = createContext(initialState);

const authReducer = (state, action)=>{

    switch(action.type){

        case 'LOGIN\_START':

            return {

                user: null,

                role:null,

                token:null

            };

            case 'LOGIN\_SUCCESS':

                return{

                    user:action.payload.user,

                    token:action.payload.token,

                    role:action.payload.role

                };

                case 'LOGOUT':

            return {

                user: null,

                role:null,

                token:null

            };

        default:

            return state;

    }

};

export const AuthContextProvider = ({children})=>{

    const [state, dispatch] = useReducer(authReducer, initialState)

    useEffect(()=>{

        localStorage.setItem('user', JSON.stringify(state.user))

        localStorage.setItem('token', state.token)

        localStorage.setItem('role', state.role)

    }, [state])

    return <AuthContext.Provider value={{user:state.user, token:state.token, role:state.role, dispatch}}>

        {children}

    </AuthContext.Provider>

}

14)dashboard.jsx:

import React, { useState } from 'react'

import Loader from '../../components/Loader/Loading'

import Error from '../../components/Error/Error'

import useFetchData from '../../hooks/useFetchData'

import { BASE\_URL } from '../../config.js'

import Tabs from './Tabs.jsx'

import starIcon from '/src/assets/images/Star.png'

import DoctorAbout from './../../pages/Doctors/DoctorAbout.jsx'

import Profile from './Profile.jsx'

const Dashboard = () => {

  const {data, loading, error } = useFetchData(`${BASE\_URL}/doctors/profile/me`);

  const [tab, setTab] = useState('overview')

  return (

    <section>

      <div className='max-w-[1170px] mx-auto'>

        {loading && !error && < Loader/>}

        {error && !loading && <Error/>}

        {!loading && !error && (

          <div className='grid lg:grid-cols-3 gap-[30px] lg:gap-[50px]'>

            <Tabs tab={tab} setTab={setTab}/>

            <div className='lg:col-span-2'>

              {data.isApproved === 'pending' && (

                <div className='flex p-4 mb-4 text-yellow-800 bg-yellow-50

                rounded-lg'>

                  <svg aria-hidden

                  className='flex-shrink-0 w-5 h-5'

                  fill='currentColor' viewBox='0 0 20 20' xmlns='http://www.w3.org/2000/svg'>

                  <path fillRule='evenodd' d='M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9

                  9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z' clipRule='evenodd'>

                  </path>

                  </svg>

                  <span className='sr-only'>Info</span>

                  <div className='ml-3 text-sm font-medium'>

                    To get approval please complete your profile. We&apos;ll review

                    manually and approve within 3 days

                  </div>

                </div>

              )}

              <div className='mt-8'>

  {tab === 'overview' && (

    <div className='flex gap-4 mb-10'>

      <figure className='max-w-[200px] max-h-[200px] flex-shrink-0'>

        <img src={data?.photo} className='w-full h-full object-cover' alt='doctor' />

      </figure>

      <div>

        <span className='bg-[#CCF0F3] text-irisBlueColor py-1 px-4

          lg:py-2 lg:px-6 rounded text-[12px] leading-4 lg:text-[16px] lg:leading-6 font-semibold'>

          {data.specialization} Surgeon

        </span>

        <h3 className='text-[22px] leading-9 font-bold text-headingColor mt-3'>{data.name}</h3>

        <div className='flex items-center gap-[6px]'>

          <span className='flex items-center gap-[6px] text-headingColor text-[14px]

            leading-5 lg:text-[16px] lg:leading-6 font-semibold'>

            <img src={starIcon} alt='star' />4.5

          </span>

          <span className='text-textColor text-[14px] leading-5 lg:text-[16px] lg:leading-6 font-semibold'>

            (272)

          </span>

        </div>

        <p className='text\_\_para font-[15px] lg:max-w-[390px] leading-6 mt-2'>doctor bio</p>

        <DoctorAbout

          name={data.name}

          about={data.about}

          qualifications={data.qualifications}

          experiences={data.experiences}

        />

      </div>

    </div>

  )}

                {tab === 'appointments' && <div>appointments</div>}

                {tab === 'settings' && <div>,<Profile/></div>}

              </div>

            </div>

          </div>

        )}

      </div>

    </section>

  )

}

export default Dashboard

15)profile.jsx:

import { useState } from 'react'

import { AiOutlineDelete } from 'react-icons/ai'

const Profile = () => {

  const [formData, setFormData] = useState({

    name: '',

    email: '',

    phone: '',

    bio: '',

    gender: '',

    specialization: '',

    ticketPrice: 0,

    qualifications: [{ startingDate: '', endingDate: '', degree: '', university: '' }],

    experiences: [{ startingDate: '', endingDate: '', position: '', hospital: '' }],

    timeSlots: [{ day: '', startingTime: '', endingTime: '' }],

    about: '',

    photo: null,

  })

  const handleInputChange = e => {

    setFormData({ ...formData, [e.target.name]: e.target.value })

  }

  const handleFileInputChange = e => {

    const file = e.target.files[0]

    if (file) {

      const reader = new FileReader()

      reader.onload = () => setFormData(prev => ({ ...prev, photo: reader.result }))

      reader.readAsDataURL(file)

    }

  }

  const addItem = (key, item) => {

    setFormData(prevFormData => ({

      ...prevFormData,

      [key]: [...prevFormData[key], item],

    }))

  }

  const handleReusableInputChangeFunc = (key, index, event) => {

    const { name, value } = event.target

    setFormData(prevFormData => {

      const updatedItems = [...prevFormData[key]]

      updatedItems[index] = {

        ...updatedItems[index],

        [name]: value,

      }

      return {

        ...prevFormData,

        [key]: updatedItems,

      }

    })

  }

  const deleteItem = (key, index) => {

    setFormData(prevFormData => ({

      ...prevFormData,

      [key]: prevFormData[key].filter((\_, i) => i !== index),

    }))

  }

  const updateProfileHandler = async e => {

    e.preventDefault()

    // handle submit logic here

  }

  // Corrected add/delete/handle functions for qualifications and experiences:

  const addQualifications = e => {

    e.preventDefault()

    addItem('qualifications', { startingDate: '', endingDate: '', degree: '', university: '' })

  }

  const handleQualificationsChange = (e, index) => {

    handleReusableInputChangeFunc('qualifications', index, e)

  }

  const deleteQualifications = (e, index) => {

    e.preventDefault()

    deleteItem('qualifications', index)

  }

  const addExperiences = e => {

    e.preventDefault()

    addItem('experiences', { startingDate: '', endingDate: '', position: '', hospital: '' })

  }

  const handleExperiencesChange = (e, index) => {

    handleReusableInputChangeFunc('experiences', index, e)

  }

  const deleteExperiences = (e, index) => {

    e.preventDefault()

    deleteItem('experiences', index)

  }

  // Handle timeSlots change and delete similarly

  const handleTimeSlotsChange = (e, index) => {

    handleReusableInputChangeFunc('timeSlots', index, e)

  }

  const deleteTimeSlot = (e, index) => {

    e.preventDefault()

    deleteItem('timeSlots', index)

  }

  const addTimeSlot = e => {

    e.preventDefault()

    addItem('timeSlots', { day: '', startingTime: '', endingTime: '' })

  }

  return (

    <div>

      <h2 className="text-headingColor font-bold text-[24px] leading-9 mb-10 ">Profile Information</h2>

      <form>

        {/\* Name \*/}

        <div className="mb-5">

          <p className="form\_\_label">Name\*</p>

          <input

            type="text"

            name="name"

            value={formData.name}

            onChange={handleInputChange}

            placeholder="Full Name"

            className="form\_\_input"

          />

        </div>

        {/\* Email \*/}

        <div className="mb-5">

          <p className="form\_\_label">Email\*</p>

          <input

            type="email"

            name="email"

            value={formData.email}

            readOnly

            aria-readonly

            disabled

            className="form\_\_input"

          />

        </div>

        {/\* Phone \*/}

        <div className="mb-5">

          <p className="form\_\_label">Phone\*</p>

          <input

            type="number"

            name="phone"

            value={formData.phone}

            onChange={handleInputChange}

            placeholder="Mobile Number"

            className="form\_\_input"

          />

        </div>

        {/\* Bio \*/}

        <div className="mb-5">

          <p className="form\_\_label">Bio\*</p>

          <input

            type="text"

            name="bio"

            value={formData.bio}

            onChange={handleInputChange}

            placeholder="Bio"

            className="form\_\_input"

            maxLength={100}

          />

        </div>

        {/\* Gender, Specialization, Ticket Price \*/}

        <div className="mb-5">

          <div className="grid grid-cols-3 gap-5 mb-[30px]">

            <div>

              <p className="form\_\_label">Gender\*</p>

              <select

                name="gender"

                value={formData.gender}

                onChange={handleInputChange}

                className="form\_\_input py-3.5"

              >

                <option value="">Select</option>

                <option value="male">Male </option>

                <option value="female">Female </option>

                <option value="other">Other</option>

              </select>

            </div>

            <div>

              <p className="form\_\_label">Specialization\*</p>

              <select

                name="specialization"

                value={formData.specialization}

                onChange={handleInputChange}

                className="form\_\_input py-3.5"

              >

                <option value="">Select</option>

                <option value="surgeon">Surgeon </option>

                <option value="neurologist">Neurologist </option>

                <option value="gynaecologist">Gynaecologist</option>

              </select>

            </div>

            <div>

              <p className="form\_\_label">Doctor Fee\*</p>

              <input

                type="number"

                placeholder="100"

                name="ticketPrice"

                value={formData.ticketPrice}

                className="form\_\_input"

                onChange={handleInputChange}

              />

            </div>

          </div>

        </div>

        {/\* Qualifications \*/}

        <div className="mb-5">

          <p className="form\_\_label">Qualifications\*</p>

          {formData.qualifications?.map((item, index) => (

            <div key={index}>

              <div className="grid grid-cols-2 gap-5">

                <div>

                  <p className="form\_\_data">Starting Date</p>

                  <input

                    type="date"

                    name="startingDate"

                    value={item.startingDate}

                    className="form\_\_input"

                    onChange={e => handleQualificationsChange(e, index)}

                  />

                </div>

                <div>

                  <p className="form\_\_data">Ending Date</p>

                  <input

                    type="date"

                    name="endingDate"

                    value={item.endingDate}

                    className="form\_\_input"

                    onChange={e => handleQualificationsChange(e, index)}

                  />

                </div>

              </div>

              <div className="grid grid-cols-2 gap-5">

                <div>

                  <p className="form\_\_data">Degree\*</p>

                  <input

                    type="text"

                    name="degree"

                    value={item.degree}

                    className="form\_\_input"

                    onChange={e => handleQualificationsChange(e, index)}

                  />

                </div>

                <div>

                  <p className="form\_\_data">University\*</p>

                  <input

                    type="text"

                    name="university"

                    value={item.university}

                    className="form\_\_input"

                    onChange={e => handleQualificationsChange(e, index)}

                  />

                </div>

              </div>

              <button

                onClick={e => deleteQualifications(e, index)}

                className="bg-red-600 p-2 rounded-full text-white text-[18px] mt-2 mb-[30px] cursor-pointer"

              >

                <AiOutlineDelete />

              </button>

            </div>

          ))}

          <button

            onClick={addQualifications}

            className="bg-[#000] py-2 px-5 rounded text-white h-fit cursor-pointer"

          >

            Add Qualifications

          </button>

        </div>

        {/\* Experiences \*/}

        <div className="mb-5">

          <p className="form\_\_label">Experiences\*</p>

          {formData.experiences?.map((item, index) => (

            <div key={index}>

              <div className="grid grid-cols-2 gap-5">

                <div>

                  <p className="form\_\_data">Starting Date</p>

                  <input

                    type="date"

                    name="startingDate"

                    value={item.startingDate}

                    className="form\_\_input"

                    onChange={e => handleExperiencesChange(e, index)}

                  />

                </div>

                <div>

                  <p className="form\_\_data">Ending Date</p>

                  <input

                    type="date"

                    name="endingDate"

                    value={item.endingDate}

                    className="form\_\_input"

                    onChange={e => handleExperiencesChange(e, index)}

                  />

                </div>

              </div>

              <div className="grid grid-cols-2 gap-5">

                <div>

                  <p className="form\_\_data">Position\*</p>

                  <input

                    type="text"

                    name="position"

                    value={item.position}

                    className="form\_\_input"

                    onChange={e => handleExperiencesChange(e, index)}

                  />

                </div>

                <div>

                  <p className="form\_\_data">Hospital\*</p>

                  <input

                    type="text"

                    name="hospital"

                    value={item.hospital}

                    className="form\_\_input"

                    onChange={e => handleExperiencesChange(e, index)}

                  />

                </div>

              </div>

              <button

                onClick={e => deleteExperiences(e, index)}

                className="bg-red-600 p-2 rounded-full text-white text-[18px] mt-2 mb-[30px] cursor-pointer"

              >

                <AiOutlineDelete />

              </button>

            </div>

          ))}

          <button

            onClick={addExperiences}

            className="bg-[#000] py-2 px-5 rounded text-white h-fit cursor-pointer"

          >

            Add Experiences

          </button>

        </div>

        {/\* Time Slots \*/}

        <div className="mb-5">

          <p className="form\_\_label">Time Slots\*</p>

          {formData.timeSlots?.map((item, index) => (

            <div key={index} className="grid grid-cols-4 gap-5 mb-3">

              <div>

                <p className="form\_\_data">Day\*</p>

                <select

                  name="day"

                  value={item.day}

                  onChange={e => handleTimeSlotsChange(e, index)}

                  className="form\_\_input"

                >

                  <option value="">Select Day</option>

                  <option value="Monday">Monday</option>

                  <option value="Tuesday">Tuesday</option>

                  <option value="Wednesday">Wednesday</option>

                  <option value="Thursday">Thursday</option>

                  <option value="Friday">Friday</option>

                  <option value="Saturday">Saturday</option>

                  <option value="Sunday">Sunday</option>

                </select>

              </div>

              <div>

                <p className="form\_\_data">Starting Time\*</p>

                <input

                  type="time"

                  name="startingTime"

                  value={item.startingTime}

                  className="form\_\_input"

                  onChange={e => handleTimeSlotsChange(e, index)}

                />

              </div>

              <div>

                <p className="form\_\_data">Ending Time\*</p>

                <input

                  type="time"

                  name="endingTime"

                  value={item.endingTime}

                  className="form\_\_input"

                  onChange={e => handleTimeSlotsChange(e, index)}

                />

              </div>

              <div className="flex items-end">

                <button

                  onClick={e => deleteTimeSlot(e, index)}

                  className="bg-red-600 p-2 rounded-full text-white text-[18px] cursor-pointer"

                >

                  <AiOutlineDelete />

                </button>

              </div>

            </div>

          ))}

          <button

            onClick={addTimeSlot}

            className="bg-[#000] py-2 px-5 rounded text-white h-fit cursor-pointer"

          >

            Add Time Slot

          </button>

        </div>

        {/\* About \*/}

        <div className="mb-5">

          <p className="form\_\_label">About\*</p>

          <textarea

            rows="4"

            name="about"

            value={formData.about}

            onChange={handleInputChange}

            className="form\_\_input"

          />

        </div>

        {/\* Photo \*/}

        <div className="mb-5">

          <p className="form\_\_label">Upload Your Image</p>

          <input

            type="file"

            onChange={handleFileInputChange}

            className="form\_\_input"

            accept="image/\*"

          />

          {formData.photo && (

            <img

              src={formData.photo}

              alt="Profile preview"

              className="mt-3 w-32 h-32 object-cover rounded"

            />

          )}

        </div>

        <button

          className="bg-[#000] py-3 px-6 rounded text-white cursor-pointer"

          onClick={updateProfileHandler}

        >

          Update Profile

        </button>

      </form>

    </div>

  )

}

export default Profile

16)tabs.jsx:

import {useContext} from 'react'

import {BiMenu} from 'react-icons/bi'

import { AuthContext } from '../../context/AuthContext'

import { useNavigate } from 'react-router-dom'

const Tabs = ({tab, setTab}) => {

    const {dispatch} = useContext(AuthContext)

    const navigate = useNavigate()

    const handleLogout=()=>{

        dispatch({type: 'LOGOUT'});

        navigate('/')

    }

  return (

    <div>

        <span className='lg:hidden'>

            <BiMenu className='w-6 h-6 cursor-pointer'/>

        </span>

        <div className='hidden lg:flex flex-col p-[30px] bg-white shadow-panelShadow

        items-center h-max rounded-md'>

        <button

        onClick={()=>setTab('overview')}

        className={`${tab==='overview'

            ? 'bg-indigo-100 text-primaryColor'

            : "bg-transparent text-headingColor"

        } w-full btn mt-0 rounded-md`}>Overview</button>

        <button

        onClick={()=>setTab('appointments')}

        className={`${tab==='appointments'

            ? 'bg-indigo-100 text-primaryColor'

            : "bg-transparent text-headingColor"

        } w-full btn mt-0 rounded-md`}>Appointments</button>

        <button

        onClick={()=>setTab('settings')}

        className={`${tab==='settings'

            ? 'bg-indigo-100 text-primaryColor'

            : "bg-transparent text-headingColor"

        } w-full btn mt-0 rounded-md`}>Profile</button>

        <div className='mt-[100px] w-full ' >

                    <button

                    onClick={handleLogout}

                    className='w-full bg-[#181A1E] p-3 text-[16px] leading-7 rounded-md text-white'>

                    Logout

                    </button>

                    <button

                    className='w-full bg-red-600 mt-4 p-3 text-[16px] leading-7 rounded-md text-white'>

                    Delete Account

                    </button>

                </div>

        </div>

    </div>

  )

}

export default Tabs

17)myAccount.jsx:

import {useContext, useState} from 'react'

import { AuthContext } from '../../context/AuthContext'

import MyBookings from './MyBookings.jsx';

import Profile from '/src/Dashboard/user-account/Profile.jsx';

import useFetchData from '../../hooks/useFetchData.jsx'

import { BASE\_URL } from '../../config.js';

import Loading from '../../components/Loader/Loading.jsx';

import Error from '../../components/Error/Error.jsx';

const MyAccount = () => {

    const {dispatch} = useContext(AuthContext)

    const [tab, setTab] = useState('bookings')

    const {data:userData,

           loading,

           error

        } = useFetchData(`${BASE\_URL}/users/profile/me`)

        console.log(userData,'userdata');

    const handleLogout = ()=>{

        dispatch({type: 'LOGOUT'});

    }

  return (

    <section>

        <div className='max-w-[1770px] px-5 mx-auto'>

            {loading && !error && < Loading/>}

            {error && !loading && <Error errMessage={error}/> }

        {

            !loading && !error && (

                <div className='grid md:grid-cols-3 gap-10'>

            <div className='pb-[50px] px-[30px] rounded-md'>

                <div className='flex items-center justify-center'>

                    <figure className='w-[100px] h-[100px] rounded-full border-2 border-solid border-primaryColor'>

                        <img src={userData.photo} className='w-full h-full rounded-full'/>

                    </figure>

                </div>

                <div className='text-center mt-4'>

                    <h3 className='text-[18px] leading-[30px] text-headingColor font-bold '>{userData.name}</h3>

                    <p className='text-textColor text-[15px] leading-6 font-medium'>{userData.email}</p>

                    <p className='text-textColor text-[15px] leading-6 font-medium'>Blood Type: <span className='ml-2 text-headingColor text-[22px] leading-8 '>{userData.bloodType}</span></p>

                </div>

                <div className='mt-[50px] md:mt-[100px]' >

                    <button onClick={handleLogout}className='w-full bg-[#181A1E] p-3 text-[16px] leading-7 rounded-md text-white'>Logout</button>

                    <button className='w-full bg-red-600 mt-4 p-3 text-[16px] leading-7 rounded-md text-white'>Delete Account</button>

                </div>

            </div>

            <div className='md:col-span-2 md:px-[30px]'>

                <div>

                    <button onClick={()=>setTab('bookings')} className={`${tab==='bookings' && 'bg-primaryColor text-white font-normal'} p-2 mr-5 px-5 rounded-md text-headingColor font-semibold text-[16px] leading-7

                border border-solid border-primaryColor`}>My Bookings</button>

                <button onClick={()=>setTab('settings')} className={`${tab==='settings' && 'bg-primaryColor text-white font-normal'} py-2 px-5 rounded-md

                 text-headingColor font-semibold text-[16px] leading-7

                border border-solid border-primaryColor`}>Profile Settings</button>

            </div>

            {tab==='bookings' && <MyBookings/>}

            {tab==='settings' && <Profile user={userData}/>}

            </div>

        </div>)

        }

    </div>

    </section>

  )

}

export default MyAccount

18)Mybooking.jsx:

import React from 'react'

import useFetchData from '../../hooks/useFetchData'

import DoctorCard from '../../components/Doctors/DoctorCard'

import Loading from '../../components/Loader/Loading'

import Error from '../../components/Error/Error'

import { BASE\_URL } from '../../config.js';

const MyBookings = () => {

  const {data:appointements, loading, error} = useFetchData(`${BASE\_URL}/users/appointments/my-appointments`)

  return (

    <div>

      {loading && !error && < Loading/>}

      {error && !loading && <Error errMessage={error}/> }

      {!loading && !error && (<div className='grid grid-cols-1 lg:grid-cols-2 gap-5 '>

        {

          appointements.map(doctor=> <DoctorCard doctor={doctor} key={doctor.\_id}/>)

        }

        </div>)}

        {!loading && !error && appointements.length===0 && <h2 className='mt-5 text-center leading-7

        text-[20px] font-semibold text-primaryColor'>You did not book any doctors yet </h2>}

    </div>

  )

}

export default MyBookings

19)profile.jsx:

import { useEffect, useState } from 'react';

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

import uploadImageToCloudinary from '../../utils/uploadCloudinary';

import {BASE\_URL, token} from '../../config.js'

import {toast} from 'react-toastify';

import HashLoader from 'react-spinners/HashLoader';

const Profile = ({user}) => {

  const [selectedFile, setSelectedFile] = useState(null)

  const[loading,setLoading] = useState(false)

  const[formData, setFormData] = useState({

      email:'',

      password: '',

      photo:null,

      name:'',

      gender:'',

      bloodType:"",

    });

    const navigate = useNavigate()

    useEffect(() => {

  if (user) {

    setFormData({

      name: user.name || '',

      email: user.email || '',

      photo: user.photo || '',

      gender: user.gender || '',

      bloodType: user.bloodType || '',

      password: ''

    });

  }

}, [user]);

    const handleInputChange =e=>{

      setFormData({...formData, [e.target.name]:e.target.value})

    }

    const handleFileInputChange=async event=>{

      const file = event.target.files[0];

      const data = await uploadImageToCloudinary(file)

      setSelectedFile(data.url)

      setFormData({...formData, photo:data.url})

      console.log(data)

    }

    const submitHandler=async event=>{

      event.preventDefault()

      setLoading(true)

      try {

        const res = await fetch(`${BASE\_URL}/users/${user.\_id}`,{

          method:'put',

          headers:{

            'content-type':'application/json',

            Authorization: `Bearer ${token}`

          },

          body: JSON.stringify(formData),

        })

        const {message} = await res.json()

        if(!res.ok){

          throw new Error(message)

        }

        setLoading(false)

        toast.success(message)

        navigate('/users/profile/me')

      } catch (error) {

        toast.error(error.message)

        setLoading(false)

      }

    }

  return (

    <div className='mt-10'>

      <form onSubmit={submitHandler}>

              <div className='mb-5'>

            <input type='text' placeholder={`${user.name}`} name='name' value={formData.name} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

            <div className='mb-5'>

            <input type='email' placeholder={`${user.email}`} name='email' value={formData.email} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             aria-readonly

             readOnly></input>

          </div>

          <div className='mb-5'>

            <input type='password' placeholder='Enter Your password' name='password' value={formData.password} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'></input>

          </div>

          <div className='mb-5'>

            <input type='text' placeholder='Blood Type' name='bloodType' value={formData.bloodType} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

          <div className='mb-5 flex items-center justify-between'>

            <label className='text-headingColor font-bold text-[16px] leading-7 '>

              Gender

              <select name="gender" value={formData.gender} onChange={handleInputChange} className='text-headingColor font-semibold text-[15px] leading-7 px-4 py-3 focus:outline-none' >

                <option value=''>Select</option>

                <option value='female'>Female</option>

                <option value='Male'>Male</option>

                <option value='other'>Other</option>

              </select>

            </label>

          </div>

          <div className='mb-5 flex items-center gap-3'>

            {formData.photo && (<figure className='w-[60px] h-[60px] rounded-full border-2 border-solid border-primaryColor flex items-center justify-center'>

              <img src={formData.photo} className='w-full rounded-full'/>

            </figure>)}

            <div className='relative w-[160px] h-[50px]'>

              <input type='file' name='photo' onChange={handleFileInputChange}  id='customFile' accept='.jpg, .png'

              className='absolute top-0 left-0 w-full h-full opacity-0 cursor-pointer'/>

              <label htmlFor='customFile' className='absolute top-0 left-0 w-full h-full flex items-center px-[0.75rem] py-[0.375rem] text-[15px]

              leading-6 overflow-hidden bg-[#0066ff46] text-headingColor font-semibold rounded-lg truncate cursor-pointer' >{selectedFile ? selectedFile.name : 'Upload Photo'}</label>

            </div>

          </div>

          <div className='mt-7'>

            <button

            disabled={loading && true}

             type='submit' className='w-full

             bg-primaryColor text-white text-[18px] leading-[30px] rounded-lg px-4 py-3'>

              { loading ? <HashLoader size={25} color="#ffffff"/>:'Update Profile'}</button>

          </div>

            </form>

    </div>

  )

}

export default Profile;

20)useFetchData.jsx:

import { useEffect, useState } from 'react';

import { token } from '../config.js';

const useFetchData = (url) => {

  const [data, setData] = useState([]);

  const [loading, setLoading] = useState(false);

  const [error, setError] = useState(null);

  useEffect(() => {

    const fetchData = async () => {

      setLoading(true);

      try {

        const res = await fetch(url, {

        headers: {Authorization: `Bearer ${token}`},

        });

        const result = await res.json();

        if (!res.ok) {

          throw new Error(result.message + '🙂‍↔️');

        }

        setData(result.data);

        setTimeout(() => setLoading(false), 1500);

      } catch (err) {

        setTimeout(() => {

        setLoading(false);

        setError(err.message);

    }, 1500);

      }

    };

    if (url) fetchData();

  }, [url]);

  return { data, loading, error };

};

export default useFetchData;

21)layout.jsx:

import React from 'react'

import Header from '../components/Header/Header'

import Footer from '../components/Footer/Footer'

import Routers from '../routes/Routers'

const Layout = () => {

  return (<>

  <Header />

  <main>

    <Routers />

  </main>

  <Footer />

</>

  );

};

export default Layout

22)DoctorAbout.jsx:

import React from 'react'

import { formateDate } from '../../utils/formateDate'

const DoctorAbout = ({name, about, qualifications, experiences}) => {

  return (

    <div>

      <div>

        <h3 className='text-[20px] leading-[30px] text-headingColor font-semibold flex items-center gap-2'>About of

            <span className='text-irisBlueColor font-bold text-[24px] leading-9'>Shakthi</span></h3>

        <p className='text\_\_para'>

            A Medicare surgeon is a skilled professional committed to quality surgical care, balancing precision and

                  compassion to restore health. They excel in life-saving procedures, ethical standards, and continuous learning,

                  ensuring patients receive the best treatment.

        </p>

      </div>

      <div className='mt-12'>

        <h3 className='text-[20px] leading-[30px] text-headingColor font-semibold'>Education</h3>

        <ul className='pt-4 md:p-5'>

            <li className='flex flex-col sm:flex-row sm:justify-between sm:items-end md:gap-5 mb-[30px] '>

                <div>

                    <span className='text-irisBlueColor text-[15px] leading-6 font-semibold'>{formateDate('06-04-2010')} - {formateDate('06-04-2013')}</span>

                    <p className='text-[16px] leading-6 font-medium text-textColor'>PhD in Surgeon</p>

                </div>

                <p className='text-[14px] leading-5 font-medium text-textColor'>New Apollo Hospital, India</p>

            </li>

            <li className='flex flex-col sm:flex-row sm:justify-between sm:items-end md:gap-5 mb-[30px] '>

                <div>

                    <span className='text-irisBlueColor text-[15px] leading-6 font-semibold'>{formateDate('07-21-2010')} - {formateDate('08-01-2013')}</span>

                    <p className='text-[16px] leading-6 font-medium text-textColor'>PhD in Surgeon</p>

                </div>

                <p className='text-[14px] leading-5 font-medium text-textColor'>New Apollo Hospital, India</p>

            </li>

        </ul>

      </div>

      <div className='mt-12'>

        <h3 className='text-[20px] leading-[30px] text-headingColor font-semibold'>Experience</h3>

        <ul className='pt-4 md:p-5'>

            <li className='flex flex-col sm:flex-row sm:justify-between sm:items-end md:gap-5 mb-[30px] '>

                <div>

                    <span className='text-irisBlueColor text-[15px] leading-6 font-semibold'>{formateDate('06-04-2015')} - {formateDate('06-04-2020')}</span>

                    <p className='text-[16px] leading-6 font-medium text-textColor'>Surgeon</p>

                </div>

                <p className='text-[14px] leading-5 font-medium text-textColor'>New Apollo Hospital, India</p>

            </li>

        </ul>

      </div>

    </div>

  )

}

export default DoctorAbout

23)doctorDetails.jsx:

import React, { useState } from 'react'

import doctorImg from '/src/assets/images/doctor-img02.png'

import starIcon from '/src/assets/images/Star.png'

import DoctorAbout from './DoctorAbout'

import Feedback from './Feedback'

import SidePanel from './SidePanel'

const DoctorDetails = () => {

  const[tab,SetTab] = useState('about')

  return (

    <section>

      <div className='max-w-[1170px] px-5 mx-auto'>

        <div className='grid md:grid-cols-3 gap-[50px]'>

          <div className='md:col-span-2'>

            <div className='flex items-center gap-5'>

              <figure className='max-w-[200px] max-h-[200px]'>

                <img src={doctorImg} className='w-full'/>

              </figure>

              <div>

                <span className='bg-[#CCF0F3] text-irisBlueColor py-1 px-6 lg:py-2 lg:px-6 text-[12px] leading-4

                lg:text-[16px] lg:leading-7 font-semibold rounded'>Surgeon</span>

                <h3 className='text-headinColor text-[22px] leading-9 mt-3 font-bold'>John Doe</h3>

                <div className='flex items-center gap-[6px]'>

                  <span className='flex items-center gap-[6px] text-[14px] leading-5 lg:text-[16px] lg:leading-7

                  font-semibold text-headingColor'>

                    <img src={starIcon} /> 4.8

                  </span>

                  <span className='text-[14px] leading-5 lg:text-[16px] lg:leading-7

                  font-semibold text-textColor'>(272)</span>

                </div>

                <p className='text\_\_para text-[14px] leading-5 md:text-[15px] lg:max-w-[390px]'>A Medicare surgeon is a skilled professional committed to quality surgical care, balancing precision and

                  compassion to restore health. They excel in life-saving procedures, ethical standards, and continuous learning,

                  ensuring patients receive the best treatment.</p>

              </div>

            </div>

            <div className='mt-[50px] border-b border-solid border-[#0066ff34]'>

              <button onClick={()=>SetTab('about')} className={`${tab==='about' && 'border-b border-solid border-primaryColor '} py-2 px-5 text-[16px] leading-7 text-headingColor font-bold`}>About</button>

              <button onClick={()=>SetTab('feedback')} className={`${tab==='feedback' && 'border-b border-solid border-primaryColor '} py-2 px-5 text-[16px] leading-7 text-headingColor font-bold`}>Feedback</button>

            </div>

            <div className='mt-[50px]'>

               {tab==='about' && <DoctorAbout/>}

               {tab==='feedback' && <Feedback/>}

            </div>

          </div>

          <div>

            <SidePanel />

          </div>

        </div>

      </div>

    </section>

  )

}

export default DoctorDetails

24)doctor.jsx:

import React from 'react';

import DoctorCard from './../../components/Doctors/DoctorCard';

import { doctors } from './../../assets/data/doctors';

import Testimonial from './../../components/Testimonial/Testimonial';

const Doctors = () => {

  return (

    <>

      {/\* Search Section \*/}

      <section className='bg-[#fff9ea]'>

        <div className='container text-center'>

          <h2 className='heading'>Find a Doctor</h2>

          <div className='max-w-[570px] mt-[30px] mx-auto bg-[#0066ff2c] rounded-md flex items-center justify-between'>

            <input

              type='search'

              className='py-4 pl-4 pr-2 bg-transparent w-full focus:outline-none cursor-pointer placeholder=text-textColor'

              placeholder='Search Doctors'

            />

            <button className='btn mt-0 rounded-[0px] rounded-r-md'>Search</button>

          </div>

        </div>

      </section>

      {/\* Doctor Cards Section \*/}

      <section>

        <div className="container">

          <div className='grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-5'>

            {doctors.map((doctor) => (

              <DoctorCard key={doctor.id} doctor={doctor} />

            ))}

          </div>

        </div>

      </section>

      {/\* Patient Testimonials Section \*/}

      <section>

        <div className="container">

          <div className='xl:w-[470px] mx-auto text-center'>

            <h2 className='heading'>What Our Patients Say</h2>

            <ul className='text\_\_para'>

              <li>1. "The doctors here truly care! They listened to my concerns and provided the best treatment. I feel healthier than ever!" – Mathew Thomas.</li>

              <li>2. "I was nervous about my surgery, but the medical team made me feel at ease. Their expertise and kindness made all the difference!" – Michael Tae.</li>

              <li>3. "From consultation to recovery, everything was smooth and professional. Highly recommend this hospital!" – Nick Jonas.</li>

            </ul>

          </div>

          {/\* Swiper Testimonials \*/}

          <Testimonial />

        </div>

      </section>

    </>

  );

};

export default Doctors;

25)feedback.jsx:

import React, { useState } from 'react'

import avatar from '/src/assets/images/avatar-icon.png'

import { formateDate } from '../../utils/formateDate'

import { AiFillStar } from 'react-icons/ai';

import FeedbackForm from './FeedbackForm';

const Feedback = () => {

    const [showFeedbackForm,setShowFeedbackForm]= useState(false)

  return (

    <div>

      <div className='mb-[50px]'>

        <h4 className='text-[20px] leading-[30px] font-bold text-headingColor mb-[30px]'>All Reviews(272)</h4>

        <div className='flex justify-between gap-10 mb-[30px]'>

            <div className='flex gap-3'>

                <figure className='w-10 h-10 rounded-full'>

                    <img className='w-full' src={avatar} />

                </figure>

                <div>

                    <h5 className='text-[16px] leading-6 text-primaryColor font-bold'>Ali Ahmed</h5>

                    <p className='text-[14px] leading-6 text-textColor'>{formateDate('02-14-2023')}</p>

                    <p className='text\_\_para mt-3 font-medium text-[15px]'>Good services, Highly recommended</p>

                </div>

            </div>

            <div className='flex gap-1'>

                {[...Array(5).keys()].map((\_, index)=>(

                    <AiFillStar key={index} color="#0067FF"/>

                ))}

            </div>

        </div>

      </div>

      {!showFeedbackForm && <div  className='text-center'>

        <button className='btn' onClick={()=>setShowFeedbackForm(true)}>Give Feedback</button>

      </div>}

      {showFeedbackForm && <FeedbackForm/>}

    </div>

  )

}

export default Feedback

26)feedback.jsx:

import React, { useState } from 'react'

import avatar from '/src/assets/images/avatar-icon.png'

import { formateDate } from '../../utils/formateDate'

import { AiFillStar } from 'react-icons/ai';

import FeedbackForm from './FeedbackForm';

const Feedback = () => {

    const [showFeedbackForm,setShowFeedbackForm]= useState(false)

  return (

    <div>

      <div className='mb-[50px]'>

        <h4 className='text-[20px] leading-[30px] font-bold text-headingColor mb-[30px]'>All Reviews(272)</h4>

        <div className='flex justify-between gap-10 mb-[30px]'>

            <div className='flex gap-3'>

                <figure className='w-10 h-10 rounded-full'>

                    <img className='w-full' src={avatar} />

                </figure>

                <div>

                    <h5 className='text-[16px] leading-6 text-primaryColor font-bold'>Ali Ahmed</h5>

                    <p className='text-[14px] leading-6 text-textColor'>{formateDate('02-14-2023')}</p>

                    <p className='text\_\_para mt-3 font-medium text-[15px]'>Good services, Highly recommended</p>

                </div>

            </div>

            <div className='flex gap-1'>

                {[...Array(5).keys()].map((\_, index)=>(

                    <AiFillStar key={index} color="#0067FF"/>

                ))}

            </div>

        </div>

      </div>

      {!showFeedbackForm && <div  className='text-center'>

        <button className='btn' onClick={()=>setShowFeedbackForm(true)}>Give Feedback</button>

      </div>}

      {showFeedbackForm && <FeedbackForm/>}

    </div>

  )

}

export default Feedback

27)feedbackform.js:

import React, { useState } from 'react';

import { AiFillStar } from 'react-icons/ai';

const FeedbackForm = () => {

  const [rating, setRating] = useState(0);

  const [hover, setHover] = useState(null);

  const [reviewText, setReviewText] = useState("");

  const handleSubmitReview = async e=>{

    e.preventDefault()

  }

  return (

    <form action=''>

      <div>

        <h3 className='text-headingColor text-[16px] leading-6 font-semibold mb-4'>

          How would you rate the overall experience?\*

        </h3>

        <div className="flex gap-2">

          {[...Array(5)].map((\_, index) => {

            const starIndex = index + 1;

            return (

              <button

                key={starIndex}

                type='button'

                className={`text-[22px] cursor-pointer bg-transparent border-none outline-none

                  ${starIndex <= (hover || rating) ? "text-yellow-500" : "text-gray-400"}`}

                onClick={() => setRating(starIndex)}

                onMouseEnter={() => setHover(starIndex)}

                onMouseLeave={() => setHover(null)}

                onDoubleClick={() => {

                  setHover(null);

                  setRating(0);

                }}

              >

                <AiFillStar />

              </button>

            );

          })}

        </div>

      </div>

      <div className='mt-[30px]'>

        <h3 className='text-headingColor text-[16px] leading-6 font-semibold mb-4'>

          Share your fedback Suggestions\*

        </h3>

        <textarea className='border border-solid border-[#0066ff34] focus:outline outline-primaryColor w-full px-4 py-3

        rounded-md' rows="5" placeholder='Write your suggestions/feedback' onChange={(e)=>setReviewText(e.target.value)}></textarea>

      </div>

      <button type='submit' className='btn' onClick={handleSubmitReview}>Submit Feedback</button>

    </form>

  );

};

export default FeedbackForm;

28)sidepanel.jsx:

import React from 'react'

const SidePanel = () => {

  return (

    <div className='shadow-panelShadow p-3 lg:p-5 rounded-md'>

        <div className="flex items-center justify-between">

            <p className='text\_\_para mt-0 font-semibold'>Ticket Price</p>

            <span className='text-[16px] leading-7 lg:text-[22px] lg:leading-8 text-headingColor font-bold'>500 BDT</span>

        </div>

        <div className='mt-[30px]'>

            <p className='text\_\_para mt-0 font-semibold text-headingColor'>Available Time Slots:</p>

            <ul className='mt-3'>

                <li className='flex items-center justify-between mb-2'>

                    <p className='text-[15px] leading-6 font-semibold text-textColor'>Sunday</p>

                    <p className='text-[15px] leading-6 font-semibold text-textColor'>4:00 P.M. - 9:30 P.M.</p>

                </li>

                <li className='flex items-center justify-between mb-2'>

                    <p className='text-[15px] leading-6 font-semibold text-textColor'>Tuesday</p>

                    <p className='text-[15px] leading-6 font-semibold text-textColor'>4:00 P.M. - 9:30 P.M.</p>

                </li>

                <li className='flex items-center justify-between mb-2'>

                    <p className='text-[15px] leading-6 font-semibold text-textColor'>WEdnesday</p>

                    <p className='text-[15px] leading-6 font-semibold text-textColor'>4:00 P.M. - 9:30 P.M.</p>

                </li>

            </ul>

        </div>

        <button className='btn px-2 w-full rounded-md'>Book Appointment</button>

    </div>

  )

}

export default SidePanel

29)contact.jsx:

import React from 'react'

const Contact = () => {

  return (

    <section>

      <div className='px-4 mx-auto max-w-screen-md'>

        <h2 className='heading text-center '>Contact Us</h2>

        <p className='mb-8 lg:mb-16 font-light text-center text\_\_para'>Got a technical Issues? Want to send feedback about a beta feature? Let us know</p>

        <form action='#' className='space-y-8'>

          <div>

            <label htmlFor="email" className='form\_\_label'>Your Email</label>

            <input type='email' id='email' placeholder='example@gmail.com' className='form\_\_input'/>

          </div>

          <div>

            <label htmlFor="subject" className='form\_\_label'>Subject</label>

            <input type='text' id='subject' placeholder='Let Us know how we can help you!' className='form\_\_input'/>

          </div>

          <div className='sm:col-span-2'>

            <label htmlFor="message" className='form\_\_label'>Your Message</label>

            <textarea rows='6' type='text' id='message' placeholder='Leave a Comment..' className='form\_\_input'/>

          </div>

          <button type='submit' className='btn rounded sm:w-fit'>Submit</button>

        </form>

      </div>

    </section>

  )

}

export default Contact

30)home.jsx:

import React from 'react';

import heroImg01 from '/src/assets/images/hero-Img01.png'

import heroImg02 from '/src/assets/images/hero-Img02.png'

import heroImg03 from '/src/assets/images/hero-Img03.png'

import icon01 from '/src/assets/images/icon01.png'

import icon02 from '/src/assets/images/icon02.png'

import icon03 from '/src/assets/images/icon03.png'

import featureImg from '/src/assets/images/feature-Img.png'

import videoIcon from '/src/assets/images/video-icon.png'

import avatarIcon from '/src/assets/images/avatar-icon.png'

import faqImg from '/src/assets/images/faq-img.png'

import { Link } from 'react-router-dom';

import { BsArrowRight } from 'react-icons/bs';

import About from '../components/About/About';

import ServiceList from '../components/Services/ServiceList';

import DoctorList from '../components/Doctors/DoctorList';

import FaqList from '../components/faq/faqlist';

import Testimonial from '../components/Testimonial/Testimonial';

const Home = () => {

  return (

    <>

      <section className='hero\_\_section pt-[60px] 2xl:h-[800px]'>

        <div className='container'>

          <div className='flex flex-col lg:flex-row gap-[90px] items-center justify-between'>

            <div className='lg:w-[570px]'>

              <h1 className='text-[36px] leading-[46px] text-headingColor font-[800] md:text-[60px]'>

                We help patients live a healthy, long life

              </h1>

              <p className='text\_\_para'>

                Welcome to MediCare, your one-stop solution for hassle-free doctor appointments! Whether you need a routine check-up or specialized medical care, our platform makes booking consultations easy and convenient. Browse expert doctors, check availability, and schedule your appointment—all in just a few clicks. Stay connected with healthcare professionals and manage your appointments effortlessly. Your health, your convenience, our priority! 🚀💙

              </p>

              <button className='btn'>Request an Appointment</button>

              <div className='mt-[30px] lg:mt-[40px] flex flex-col lg:flex-row lg:items-center gap-5 lg:gap-[30px]'>

                <div>

                  <h2 className='text-[36px] leading-[56px] lg:text-[44px] lg:leading-[54px] font-[700] text-headingColor'>30+</h2>

                  <span className='w-[100px] h-2 bg-yellowColor rounded-full block mt-[-14px]'></span>

                  <p>Years of Experience</p>

                </div>

                <div>

                  <h2 className='text-[36px] leading-[56px] lg:text-[44px] lg:leading-[54px] font-[700] text-headingColor'>15+</h2>

                  <span className='w-[100px] h-2 bg-purpleColor rounded-full block mt-[-14px]'></span>

                  <p>Clinic Location</p>

                </div>

                <div>

                  <h2 className='text-[36px] leading-[56px] lg:text-[44px] lg:leading-[54px] font-[700] text-headingColor'>100%</h2>

                  <span className='w-[100px] h-2 bg-irisBlueColor rounded-full block mt-[-14px]'></span>

                  <p>Patients Satisfaction</p>

                </div>

              </div>

            </div>

            <div className='flex gap-[30px] justify-end'>

              <img className='w-full' src={heroImg01}></img>

            </div>

            <div className='mt-[30px]'>

              <img src={heroImg02} className='w-full mb-[30px]'></img>

                <img src={heroImg03} className='w-full'></img>

            </div>

          </div>

        </div>

      </section>

      <section>

        <div className="container">

          <div className='lg:w-[470px] mx-auto'>

            <h2 className='heading text-center'>

              Providing the best medical services!🏥 Your health, our priority!🩺✨

            </h2>

            <p className='text\_\_para text-center'>🌍 World-class care for everyone!🏥 Our health system offers unmatched expert healthcare with compassion and excellence!🩺✨ Your well-being is our priority!🚀</p>

          </div>

          <div className='flex flex-wrap items-center justify-between flex-col md:flex-row gap-5 lg:gap-[30px] mt-[30px] lg:mt-[55px]'>

            <div className='grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-5 lg:gap-[30px] mt-[30px] lg:mt-[55px]'>

              <div className='py-[30px] px-5'>

                <div className='flex items-center justify-center'>

                  <img src={icon01}></img>

                </div>

                <div className='mt-[30px]'>

                  <h2 className='text-[26px] leading-9 text-headingColor font-[700] text-center'>Find a Doctor</h2>

                  <p className='text-[16px] leading-7 text-textColor font-[400] mt-4 text-center'>World class care for everyone!Our health system offers unmatched, expert health care! From lab to Clinic</p>

                  <Link to='/doctors' className='w-[44px] rounded-full border border-solid border-[#181A1E] mt-[30px] mx-auto flex items-center justify-center group hover:bg-primaryColor hover:border-none'>

                    <BsArrowRight className='group-hover:text-white w-6 h-5'/>

                  </Link>

                </div>

              </div>

              <div className='py-[30px] px-5'>

                <div className='flex items-center justify-center'>

                  <img src={icon02}></img>

                </div>

                <div className='mt-[30px]'>

                  <h2 className='text-[26px] leading-9 text-headingColor font-[700] text-center'>Find a Location</h2>

                  <p className='text-[16px] leading-7 text-textColor font-[400] mt-4 text-center'>World class care for everyone!Our health system offers unmatched, expert health care! From lab to Clinic</p>

                  <Link to='/doctors' className='w-[44px] rounded-full border border-solid border-[#181A1E] mt-[30px] mx-auto flex items-center justify-center group hover:bg-primaryColor hover:border-none'>

                    <BsArrowRight className='group-hover:text-white w-6 h-5'/>

                  </Link>

                </div>

              </div>

              <div className='py-[30px] px-5'>

                <div className='flex items-center justify-center'>

                  <img src={icon03}></img>

                </div>

                <div className='mt-[30px]'>

                  <h2 className='text-[26px] leading-9 text-headingColor font-[700] text-center'>Book Appointment</h2>

                  <p className='text-[16px] leading-7 text-textColor font-[400] mt-4 text-center'>World class care for everyone!Our health system offers unmatched, expert health care! From lab to Clinic</p>

                  <Link to='/doctors' className='w-[44px] rounded-full border border-solid border-[#181A1E] mt-[30px] mx-auto flex items-center justify-center group hover:bg-primaryColor hover:border-none'>

                    <BsArrowRight className='group-hover:text-white w-6 h-5'/>

                  </Link>

                </div>

              </div>

            </div>

          </div>

        </div>

      </section>

      <About />

      <section>

        <div className="container">

          <div className='xl:w-[470px] mx-auto'>

            <h2 className='heading text-center'>Our medical Services</h2>

            <p className='text\_\_para text-center'>🏥MediCare provides expert medical care with compassion and innovation!🩺✨From routine check-ups to specialized treatments, our world-class doctors ensure top-notch healthcare for everyone!🚀Your health, our priority!🌍💪</p>

          </div>

          <ServiceList />

        </div>

      </section>

      <section>

        <div className="container">

          <div className='flex items-center  justify-between flex-col lg:flex-row'>

            <div className='xl:w-[670px]'>

              <h2 className='heading'>Get Virtual Treatment <br /> anytime.</h2>

              <ul className='pl-4'>

                <li className='text\_\_para'>

                  1.Schedule the appointment directly

                </li>

                <li className='text\_\_para'>

                  2.Search for your physician here, and contact their office.

                </li>

                <li className='text\_\_para'>

                  3.View our physicians who are accepting new patients, use the online scheduling tool to select an appointment time.

                </li>

              </ul>

              <Link to='/'><button className="btn">Learn more</button></Link>

            </div>

<div className='relative z-10 xl:w-[770px] flex justify-end mt-[50px] lg:mt-0'>

  <img src={featureImg} className='w-3/4' alt="Feature" />

  <div className='w-[150px] lg:w-[248px] bg-white absolute bottom-[50px] left-0 md:bottom-[100px]

  md:left-5 z-20 p-2 pb-3 lg:pt-4 lg:px-4 lg:pb-[26px] rounded-[10px] flex flex-col items-center'>

  <div className='flex items-center gap-[6px] lg:gap-3'>

    <p className='text-[10px] leading-[10px] lg:text-[14px] lg:leading-5 text-headingColor font-[600]'>

      Tue, 24

    </p>

    <p className='text-[10px] leading-[10px] lg:text-[14px] lg:leading-5 text-textColor font-[400]'>

      10:00

    </p>

    <span className='w-6 h-6 lg:w-[40px] lg:h-[40px] flex items-center justify-center bg-yellowColor rounded-full py-1 px-[6px] lg:py-3 lg:px-[9px] ml-auto shadow-md'>

  <img src={videoIcon} alt="Video Icon" className='w-4 h-4 lg:w-[20px] lg:h-[20px]' />

</span>

  </div>

  <div className='w-[65px] lg:w-[96px] bg-[#CCF0F3] py-1 px-2 lg:py-[6px] lg:px-[10px] text-[8px] leading-[8px]

  lg:text-[12px] lg:leading-4 text-irisBlueColor font-[500] mt-2 rounded-full'>

    Consultation

  </div>

  <div className='flex items-center gap=[6px] lg:gap-[10px] mt-2 lg:mt-[18px] '>

    <img src={avatarIcon} />

    <h4 className='text-[10px] leading-3 lg:text-[16px] lg:leading-[22px] font-[700] gtext-headingColor'>Manohar</h4>

  </div>

</div>

</div>

          </div>

        </div>

      </section>

      <section>

        <div className="container">

          <div className='xl:w-[470px] mx-auto'>

            <h2 className='heading text-center'>Our Great Doctors</h2>

            <p className='text\_\_para text-center'>👨‍⚕️Medicare doctors are the heart of healthcare!🏥✨

              With skill, dedication, and compassion, they bring healing and hope to countless lives.

              Their commitment to excellence shapes a healthier, brighter future.🩺💙</p>

          </div>

          <DoctorList />

        </div>

      </section>

      <section>

        <div className="container">

          <div className='flex justify-between gap-[50px] lg:gap-0'>

            <div className='w-1/2 hidden md:block'>

             <img src={faqImg}/>

            </div>

            <div className='w-full md:w-1/2'>

             <h2 className='heading'>Frequently asked questions and doubts by our beloved patients</h2>

             <FaqList />

            </div>

          </div>

        </div>

      </section>

      <section>

        <div className="container">

          <div className='xl:w-[470px] mx-auto'>

            <h2 className='heading text-center'>What Our Patients say</h2>

            <p className='text\_\_para text-center'>

              <ul>

                <li>

                 1. The doctors here truly care! They listened to my concerns and provided the best treatment. I feel healthier than ever!"  -Mathew Thomas.

                </li><br/>

                <li>

                  2. I was nervous about my surgery, but the medical team made me feel at ease. Their expertise and kindness made all the difference!"  -Michael Tae.

                </li><br/>

                <li>

                  3. From consultation to recovery, everything was smooth and professional. Highly recommend this hospital!  -Nick Jonas.

                </li>

                </ul></p>

          </div>

          <Testimonial />

        </div>

      </section>

    </>

  );

};

export default Home;

31)login.jsx:

import React, { useState, useContext } from 'react'

import {Link, useNavigate} from 'react-router-dom'

import {toast} from 'react-toastify';

import {BASE\_URL} from '../config.js'

import { AuthContext } from '../context/AuthContext.jsx';

import HashLoader from 'react-spinners/HashLoader.js';

const Login = () => {

  const[formData, setFormData] = useState({

    email:'',

    password: ''

  });

  const[loading,setLoading] = useState(false);

  const navigate = useNavigate()

  const{dispatch}= useContext(AuthContext)

  const handleInputChange =e=>{

    setFormData({...formData, [e.target.name]:e.target.value})

  }

  const submitHandler=async event=>{

        event.preventDefault()

        setLoading(true)

        try {

          const res = await fetch(`${BASE\_URL}/auth/login`,{

            method:'post',

            headers:{

              'content-type':'application/json'

            },

            body: JSON.stringify(formData),

          })

          const result = await res.json()

          if(!res.ok){

            throw new Error(result.message)

          }

          dispatch({

            type:'LOGIN\_SUCCESS',

            payload:{

              user:result.data,

              token: result.token,

              role: result.role,

            },

          });

          console.log(result, 'login data');

          setLoading(false)

          toast.success(result.message)

          navigate('/home')

        } catch (error) {

          toast.error(error.message)

          setLoading(false)

        }

      }

  return (

    <section>

      <div className='w-full max-w-[570px] mx-auto rounded-lg shadow-md md:p-10'>

        <h3 className='text-headingColor text-[22px] leading-9 font-bold mb-10'>Hello! <span className='text-primaryColor'>

          Welcome</span>Back 🎉</h3>

        <form className='py-4 md:py-0' onSubmit={submitHandler}>

          <div className='mb-5'>

            <input type='email' placeholder='Enter Your Email' name='email' value={formData.email} onChange={handleInputChange}

            className='w-full py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

          <div className='mb-5'>

            <input type='password' placeholder='Password' name='password' value={formData.password} onChange={handleInputChange}

            className='w-full py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

          <div className='mt-7'>

            <button type='submit' className='w-full bg-primaryColor text-white text-[18px] leading-[30px] rounded-lg px-4 py-3'>

              {loading ? <HashLoader size={25} color='#fff'/>:'Login'}</button>

          </div>

          <p className='mt-5 text-textColor text-center'>Don&apos;t have an account? <Link to='/register'

          className='text-primaryColor font-medium ml-1'>Register</Link></p>

        </form>

      </div>

    </section>

  )

}

export default Login

32)service.jsx:

import React from 'react'

import { services } from '../assets/data/services'

import ServiceCard from '../components/Services/ServiceCard'

const Services = () => {

  return <section>

  <div className="container">

    <div className='grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-5 lg:gap-[30px] mt-[30px] lg:mt-[55px]'>

    {services.map((item,index)=>(

    <ServiceCard item={item} index={index} key={index}/>

    ))}

  </div>

    </div>

  </section>

}

export default Services

33)signup.jsx:

import React, { useState } from 'react';

import signupImg from '/src/assets/images/signup.gif';

import { Link, useNavigate } from 'react-router-dom';

import uploadImageToCloudinary from '../utils/uploadCloudinary';

import {BASE\_URL} from '../config.js'

import {toast} from 'react-toastify';

import HashLoader from 'react-spinners/HashLoader';

const Signup = () => {

  const [selectedFile, setSelectedFile] = useState(null)

  const[previewURL, setPreviewURL] =useState('')

  const[loading,setLoading] = useState(false)

  const[formData, setFormData] = useState({

      email:'',

      password: '',

      photo:selectedFile,

      name:'',

      gender:'',

      role:'patient'

    });

    const navigate = useNavigate()

    const handleInputChange =e=>{

      setFormData({...formData, [e.target.name]:e.target.value})

    }

    const handleFileInputChange=async event=>{

      const file = event.target.files[0];

      const data = await uploadImageToCloudinary(file)

      setPreviewURL(data.url)

      setSelectedFile(data.url)

      setFormData({...formData, photo:data.url})

      console.log(data)

    }

    const submitHandler=async event=>{

      event.preventDefault()

      setLoading(true)

      try {

        const res = await fetch(`${BASE\_URL}/auth/register`,{

          method:'post',

          headers:{

            'content-type':'application/json'

          },

          body: JSON.stringify(formData),

        })

        const {message} = await res.json()

        if(!res.ok){

          throw new Error(message)

        }

        setLoading(false)

        toast.success(message)

        navigate('/login')

      } catch (error) {

        toast.error(error.message)

        setLoading(false)

      }

    }

  return (

    <section className='px-5 xl:px-0'>

      <div className='max-w-[1170px] mx-auto'>

        <div className='grid grid-cols-1 lg:grid-cols-2'>

          <div className='hidden lg:block bg-primaryColor rounded-l-lg'>

            <figure className='rounded-l-lg'>

              <img

              src={signupImg}

              className='w-full rouded-l-lg' />

            </figure>

          </div>

          <div className='rounded-l-lg lg:pl-16 py-10'>

            <h3 className='text-headingColor text-[22px] leading-9 font-bold mb-10'>Create an <span className='text-primaryColor'>Account</span></h3>

            <form onSubmit={submitHandler}>

              <div className='mb-5'>

            <input type='text' placeholder='Full Name' name='name' value={formData.name} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

            <div className='mb-5'>

            <input type='email' placeholder='Enter Your email' name='email' value={formData.email} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

          <div className='mb-5'>

            <input type='password' placeholder='Enter Your password' name='password' value={formData.password} onChange={handleInputChange}

            className='w-full pr-4 py-3 border-b border-solid border-[#0066ff61] focus: outline-none

             focus:border-b-primaryColor text-[20px] leading-7 text-headingColor placeholder:text-textColor rounded-md cursor-pointer'

             required></input>

          </div>

          <div className='mb-5 flex items-center justify-between'>

            <label htmlFor='' className='text-headingColor font-bold text-[16px] leading-7 '>

              Are you a:

              <select name="role"  value={formData.role} onChange={handleInputChange} className='text-headingColor font-semibold text-[15px] leading-7 px-4 py-3 focus:outline-none' >

                <option value='patient'>Patient</option>

                <option value='doctor'>Doctor</option>

              </select>

            </label>

            <label className='text-headingColor font-bold text-[16px] leading-7 '>

              Gender

              <select name="gender" value={formData.gender} onChange={handleInputChange} className='text-headingColor font-semibold text-[15px] leading-7 px-4 py-3 focus:outline-none' >

                <option value=''>Select</option>

                <option value='female'>Female</option>

                <option value='Male'>Male</option>

                <option value='other'>Other</option>

              </select>

            </label>

          </div>

          <div className='mb-5 flex items-center gap-3'>

            {selectedFile && <figure className='w-[60px] h-[60px] rounded-full border-2 border-solid border-primaryColor flex items-center justify-center'>

              <img src={previewURL} className='w-full rounded-full'/>

            </figure>}

            <div className='relative w-[160px] h-[50px]'>

              <input type='file' name='photo' onChange={handleFileInputChange}  id='customFile' accept='.jpg, .png'

              className='absolute top-0 left-0 w-full h-full opacity-0 cursor-pointer'/>

              <label htmlFor='customFile' className='absolute top-0 left-0 w-full h-full flex items-center px-[0.75rem] py-[0.375rem] text-[15px]

              leading-6 overflow-hidden bg-[#0066ff46] text-headingColor font-semibold rounded-lg truncate cursor-pointer' >Upload Photo</label>

            </div>

          </div>

          <div className='mt-7'>

            <button

            disabled={loading && true}

             type='submit' className='w-full

             bg-primaryColor text-white text-[18px] leading-[30px] rounded-lg px-4 py-3'>

              { loading ? <HashLoader size={35} color="#ffffff"/>:'Sign Up'}</button>

          </div>

          <p className='mt-5 text-textColor text-center'>Already have an account? <Link to='/login'

          className='text-primaryColor font-medium ml-1'>Login</Link></p>

            </form>

          </div>

        </div>

      </div>

    </section>

  );

};

export default Signup

33)protectedRoute.jsx:

import { useContext } from 'react'

import { Navigate } from 'react-router-dom'

import { AuthContext } from '../context/AuthContext'

const ProtectedRoute = ({children, allowedRoles }) => {

    const {token, role} = useContext(AuthContext)

    const isAllowed = allowedRoles.includes(role)

    const accessibleRoute =

    token && isAllowed ? children : <Navigate to= '/login' replace={true} />;

  return accessibleRoute;

};

export default ProtectedRoute;

34)routers.jsx:

import Home from '../pages/Home'

import Services from '../pages/Services'

import Login from '../pages/Login'

import Signup from '../pages/Signup'

import Contact from '../pages/Contact'

import Doctors from '../pages/Doctors/Doctors'

import DoctorDetails from '../pages/Doctors/DoctorDetails'

import MyAccount from '../Dashboard/user-account/MyAccount'

import Dashboard from '../Dashboard/doctor-account/Dashboard'

import {Routes,Route} from 'react-router-dom'

import ProtectedRoute from './ProtectedRoute'

const Routers = () => {

  return (

    <Routes>

      <Route path="/" element={<Home/>} />

      <Route path="/home" element={<Home/>} />

      <Route path="/doctors" element={<Doctors/>} />

      <Route path="/doctors/:id" element={<DoctorDetails/>} />

      <Route path="/login" element={<Login/>} />

      <Route path="/register" element={<Signup/>} />

      <Route path="/services" element={<Services/>} />

      <Route path="/contact" element={<Contact/>} />

      <Route path="/users/profile/me" element={<ProtectedRoute allowedRoles={['patient']}><MyAccount/></ProtectedRoute>} />

      <Route path="/doctors/profile/me" element={<ProtectedRoute allowedRoles={['doctor']}><Dashboard/></ProtectedRoute>} />

    </Routes>

  )

}

export default Routers

35)formatdate.js:

export const formateDate=(date, config)=>{

    const defaultOptions = {day:'numeric', month:'long', year:'numeric'}

    const options= config ? config : defaultOptions

    return new Date(date).toLocaleDateString('en-India', options)

}

36)uploadCloudiary.js:

const upload\_preset = import.meta.env.VITE\_UPLOAD\_PRESET;

const cloud\_name = import.meta.env.VITE\_CLOUD\_NAME;

const uploadImageToCloudinary = async (file) => {

  const uploadData = new FormData();

  uploadData.append("file", file);

  uploadData.append("upload\_preset", upload\_preset);

  try {

    const res = await fetch(`https://api.cloudinary.com/v1\_1/${cloud\_name}/image/upload`, {

      method: "POST",

      body: uploadData,

    });

    if (!res.ok) {

      throw new Error("Cloudinary upload failed");

    }

    const data = await res.json();

    return data;

  } catch (err) {

    console.error("Error uploading image to Cloudinary:", err);

    return null;

  }

};

export default uploadImageToCloudinary;

37)app.jsx:

import "./app.css";

import Layout from "./layout/Layout";

function App() {

  return <Layout />;

}

export default App;