Ok here is what I want you to do based of the frontend logic I want you to update the backends, and viewprofile.js is the way I want the profile to be viewed so let us adjust everything on this. And freelancerprofile.js is what the profile.js should include  
  
wait for me till I send you everyfile that is needed and you will send me the solution

Freelancerprofile.js  
  
import React, { useState } from 'react';

import { Editor } from '@tinymce/tinymce-react';

const FreelancerProfile = () => {

  const [description, setDescription] = useState('');

  const [profileImage, setProfileImage] = useState(null);

  const [fullName, setFullName] = useState('');

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

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

  const [location, setLocation] = useState('');

  const [jobTitle, setJobTitle] = useState('');

  const [minRate, setMinRate] = useState('');

  const [maxRate, setMaxRate] = useState('');

  const [gender, setGender] = useState('');

  const [dob, setDob] = useState('');

  const [freelancerType, setFreelancerType] = useState('');

  const [category, setCategory] = useState('');

  const [friendlyAddress, setFriendlyAddress] = useState('');

  const [gallery, setGallery] = useState(null);

  const [resume, setResume] = useState(null);

  const [videoUrl, setVideoUrl] = useState('');

  const [awards, setAwards] = useState([{ id: 1, value: '' }]);

  const [skills, setSkills] = useState([{ id: 1, value: '' }]);

  const [faqs, setFaqs] = useState([{ id: 1, value: '' }]);

  const handleEditorChange = (content) => {

    setDescription(content);

  };

  const handleAddField = (setState, state) => {

    setState([...state, { id: state.length + 1, value: '' }]);

  };

  const handleRemoveField = (id, setState, state) => {

    setState(state.filter((item) => item.id !== id));

  };

  const handleSaveProfile = () => {

    const profileData = {

      profileImage,

      fullName,

      email,

      phone,

      location,

      jobTitle,

      minRate,

      maxRate,

      description,

      gallery,

      resume,

      videoUrl,

      awards,

      skills,

      faqs,

      gender,

      dob,

      freelancerType,

      category,

      friendlyAddress,

    };

    console.log('Profile Data:', profileData);

  };

  return (

    <div className="max-w-6xl mx-auto p-6">

      <h1 className="text-3xl font-bold mb-8 text-center">Edit Profile</h1>

      {/\* Profile Image \*/}

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Profile Image</label>

        <input type="file" onChange={(e) => setProfileImage(e.target.files[0])} className="block w-full" />

      </div>

      {/\* Full Name \*/}

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Full Name</label>

        <input

          type="text"

          value={fullName}

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

          className="w-full p-2 border border-gray-300 rounded-md"

        />

      </div>

      {/\* Gender, Date of Birth \*/}

      <div className="mb-6 grid grid-cols-1 md:grid-cols-2 gap-6">

        <div>

          <label className="block text-lg font-semibold mb-2">Gender</label>

          <select

            value={gender}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          >

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

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

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

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

          </select>

        </div>

        <div>

          <label className="block text-lg font-semibold mb-2">Date of Birth</label>

          <input

            type="date"

            value={dob}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

      </div>

      {/\* Email, Phone \*/}

      <div className="mb-6 grid grid-cols-1 md:grid-cols-2 gap-6">

        <div>

          <label className="block text-lg font-semibold mb-2">Email</label>

          <input

            type="email"

            value={email}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

        <div>

          <label className="block text-lg font-semibold mb-2">Phone Number</label>

          <input

            type="text"

            value={phone}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

      </div>

      {/\* Freelancer Type, Category \*/}

      <div className="mb-6 grid grid-cols-1 md:grid-cols-2 gap-6">

        <div>

          <label className="block text-lg font-semibold mb-2">Freelancer Type</label>

          <select

            value={freelancerType}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          >

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

            <option value="full-time">Full-time</option>

            <option value="part-time">Part-time</option>

            <option value="contract">Contract</option>

          </select>

        </div>

        <div>

          <label className="block text-lg font-semibold mb-2">Category</label>

          <select

            value={category}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          >

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

            <option value="web-development">Web Development</option>

            <option value="graphic-design">Graphic Design</option>

            <option value="content-writing">Content Writing</option>

            <option value="digital-marketing">Digital Marketing</option>

          </select>

        </div>

      </div>

      {/\* Job Title, Rates \*/}

      <div className="mb-6 grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">

        <div>

          <label className="block text-lg font-semibold mb-2">Job Title</label>

          <input

            type="text"

            value={jobTitle}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

        <div>

          <label className="block text-lg font-semibold mb-2">Min Hourly Rate</label>

          <input

            type="number"

            value={minRate}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

        <div>

          <label className="block text-lg font-semibold mb-2">Max Hourly Rate</label>

          <input

            type="number"

            value={maxRate}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

      </div>

      {/\* Location and Friendly Address \*/}

      <div className="mb-6 grid grid-cols-1 md:grid-cols-2 gap-6">

        <div>

          <label className="block text-lg font-semibold mb-2">Location</label>

          <input

            type="text"

            value={location}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

        <div>

          <label className="block text-lg font-semibold mb-2">Friendly Address</label>

          <input

            type="text"

            value={friendlyAddress}

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

            className="w-full p-2 border border-gray-300 rounded-md"

          />

        </div>

      </div>

      {/\* Description \*/}

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Description</label>

        <Editor

          value={description}

          init={{

            height: 300,

            menubar: false,

            plugins: [

              'advlist autolink lists link image charmap print preview anchor',

              'searchreplace visualblocks code fullscreen',

              'insertdatetime media table paste code help wordcount',

            ],

            toolbar:

              'undo redo | formatselect | bold italic backcolor | \

                alignleft aligncenter alignright alignjustify | \

                bullist numlist outdent indent | removeformat | help',

          }}

          onEditorChange={handleEditorChange}

        />

      </div>

      {/\* Media Upload \*/}

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Gallery Image</label>

        <input type="file" onChange={(e) => setGallery(e.target.files[0])} className="block w-full" />

      </div>

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Resume</label>

        <input type="file" onChange={(e) => setResume(e.target.files[0])} className="block w-full" />

      </div>

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Introduction Video URL</label>

        <input

          type="text"

          value={videoUrl}

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

          className="w-full p-2 border border-gray-300 rounded-md"

        />

      </div>

      {/\* Awards, Skills, FAQ \*/}

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Awards</label>

        {awards.map((award) => (

          <div key={award.id} className="flex items-center mb-2">

            <input

              type="text"

              value={award.value}

              onChange={(e) =>

                setAwards(awards.map((item) => (item.id === award.id ? { ...item, value: e.target.value } : item)))

              }

              className="w-full p-2 border border-gray-300 rounded-md"

            />

            <button

              onClick={() => handleRemoveField(award.id, setAwards, awards)}

              className="ml-2 px-4 py-2 bg-red-500 text-white rounded-md"

            >

              Remove

            </button>

          </div>

        ))}

        <button onClick={() => handleAddField(setAwards, awards)} className="px-4 py-2 bg-blue-500 text-white rounded-md">

          Add Award

        </button>

      </div>

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">Skills</label>

        {skills.map((skill) => (

          <div key={skill.id} className="flex items-center mb-2">

            <input

              type="text"

              value={skill.value}

              onChange={(e) =>

                setSkills(skills.map((item) => (item.id === skill.id ? { ...item, value: e.target.value } : item)))

              }

              className="w-full p-2 border border-gray-300 rounded-md"

            />

            <button

              onClick={() => handleRemoveField(skill.id, setSkills, skills)}

              className="ml-2 px-4 py-2 bg-red-500 text-white rounded-md"

            >

              Remove

            </button>

          </div>

        ))}

        <button onClick={() => handleAddField(setSkills, skills)} className="px-4 py-2 bg-blue-500 text-white rounded-md">

          Add Skill

        </button>

      </div>

      <div className="mb-6">

        <label className="block text-lg font-semibold mb-2">FAQ</label>

        {faqs.map((faq) => (

          <div key={faq.id} className="flex items-center mb-2">

            <input

              type="text"

              value={faq.value}

              onChange={(e) =>

                setFaqs(faqs.map((item) => (item.id === faq.id ? { ...item, value: e.target.value } : item)))

              }

              className="w-full p-2 border border-gray-300 rounded-md"

            />

            <button

              onClick={() => handleRemoveField(faq.id, setFaqs, faqs)}

              className="ml-2 px-4 py-2 bg-red-500 text-white rounded-md"

            >

              Remove

            </button>

          </div>

        ))}

        <button onClick={() => handleAddField(setFaqs, faqs)} className="px-4 py-2 bg-blue-500 text-white rounded-md">

          Add FAQ

        </button>

      </div>

      {/\* Save Button \*/}

      <button

        onClick={handleSaveProfile}

        className="px-6 py-3 bg-green-500 text-white font-semibold rounded-md hover:bg-green-600"

      >

        Save Profile

      </button>

    </div>

  );

};

export default FreelancerProfile;

viewprofile.js  
  
import React from 'react';

import { FaMapMarkerAlt, FaClock, FaStar, FaEnvelope, FaPhone, FaUserCircle, FaBriefcase, FaCalendarAlt, FaLanguage, FaTransgender } from 'react-icons/fa';

const ViewProfilePageReal = () => {

  return (

    <div className="min-h-screen bg-gray-100">

      {/\* Header Section \*/}

      <div className="bg-orange-50 text-black py-10 px-6 lg:flex lg:items-center lg:justify-between">

        <div className="lg:flex lg:items-center">

          <img

            src="https://via.placeholder.com/150"

            alt="Freelancer"

            className="w-32 h-32 lg:w-48 lg:h-48 rounded-full object-cover mr-6"

          />

          <div>

            <h1 className="text-3xl font-bold">Agent Pakulla</h1>

            <p className="text-lg mt-2">Nursing Assistant</p>

            <div className="flex items-center space-x-6 mt-4">

              <span className="flex items-center">

                <FaCalendarAlt className="mr-2 text-gray-600" />

                DOB: January 1, 1990

              </span>

              <span className="flex items-center">

                <FaMapMarkerAlt className="mr-2 text-gray-600" />

                Location: New York

              </span>

              <span className=" px-3 py-1 rounded-full text-sm">

                ⭐ 4.9 (52 Reviews)

              </span>

            </div>

          </div>

        </div>

        <div className="mt-6 lg:mt-0">

          <button className="bg-orange-100 text-green-00 py-2 px-6 rounded-lg shadow-lg mr-4 border-2 border-green-900 hover:bg-orange-200">

            Hire Me

          </button>

          <button className="bg-green-800 text-white py-2 px-6 rounded-lg hover:bg-green-900">

            Contact Me

          </button>

        </div>

      </div>

      <div className="container mx-auto mt-10 p-4 lg:p-10 lg:flex lg:space-x-8">

        {/\* Left Column: Freelancer Information \*/}

        <div className="lg:w-2/3 space-y-8">

          {/\* About Me Section \*/}

          <div className=" p-6">

            <h3 className="text-2xl font-bold mb-4">About Me</h3>

            <p>

              I am a Nursing Assistant with over 10 years of experience in providing patient care and healthcare services.

              I have worked in various healthcare settings and excel in delivering compassionate and professional care.

              My passion for healthcare has led me to continuously improve my skills and ensure that I am always at the forefront of best practices.

            </p>

          </div>

          {/\* Education Section \*/}

          <div className="bg-white shadow-md rounded-lg p-6">

            <h3 className="text-xl font-bold mb-4">Education</h3>

            <ul className="list-disc ml-5">

              <li>Bachelor of Nursing - XYZ University, 2012</li>

              <li>Master of Healthcare Management - ABC University, 2015</li>

            </ul>

          </div>

          {/\* Work Experience Section \*/}

          <div className="bg-white shadow-md rounded-lg p-6">

            <h3 className="text-2xl font-bold mb-4">Work Experience</h3>

            <ul className="list-disc ml-5">

              <li>Senior Nursing Assistant at XYZ Hospital, 2015-Present</li>

              <li>Healthcare Consultant, Freelance, 2018-Present</li>

            </ul>

          </div>

          {/\* Awards Section \*/}

          <div className="bg-white shadow-md rounded-lg p-6">

            <h3 className="text-2xl font-bold mb-4">Awards</h3>

            <ul className="list-disc ml-5">

              <li>Best Healthcare Assistant Award, 2019</li>

              <li>Outstanding Nursing Services Award, 2020</li>

            </ul>

          </div>

          {/\* Posted Services Section \*/}

          <div className=" p-6">

            <h3 className="text-2xl font-bold mb-4">Posted Services</h3>

            <div className="grid grid-cols-1 md:grid-cols-2 gap-6">

              <div className="bg-white border border-gray-200 rounded-lg shadow-lg p-4">

                <img

                  src="https://freeio.blogdu.de/wp-content/uploads/2022/11/service11-768x576.jpg"

                  alt="Service Image"

                  className="w-full h-40 object-cover rounded-md mb-4"

                />

                <div className="text-gray-500 text-sm">Design & Creative</div>

                <h3 className="text-lg font-semibold text-gray-900">Developers drop the framework folder into a...</h3>

                <div className="flex items-center mt-2 mb-4">

                  <FaStar className="text-yellow-400" />

                  <span className="ml-1 text-gray-700 font-bold">4.5</span>

                  <span className="ml-2 text-gray-500">(2 Reviews)</span>

                </div>

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

                  <div className="text-sm text-gray-500">John Paul</div>

                  <div className="text-gray-900 font-bold">Starting at: $128.00</div>

                </div>

              </div>

              {/\* Add more services here if needed \*/}

            </div>

          </div>

          //

        </div>

        {/\* Right Column: About the Seller and Skills \*/}

        <div className="lg:w-1/3 space-y-8">

          {/\* About the Seller Widget \*/}

          <div className="bg-white shadow-md rounded-lg p-6">

            <ul className="space-y-4">

              <li className="flex items-center">

                <FaBriefcase className="text-gray-500 mr-2" />

                $60 - $65 / hr

              </li>

              <li className="flex items-center">

                <FaMapMarkerAlt className="text-gray-500 mr-2" />

                Location: New York

              </li>

              <li className="flex items-center">

                <FaUserCircle className="text-gray-500 mr-2" />

                Type: Agency Freelancers

              </li>

              <li className="flex items-center">

                <FaLanguage className="text-gray-500 mr-2" />

                English Level: Professional

              </li>

              <li className="flex items-center">

                <FaTransgender className="text-gray-500 mr-2" />

                Gender: Male

              </li>

              <li className="flex items-center">

                <FaEnvelope className="text-gray-500 mr-2" />

                Email: agentpakulla@apus.com

              </li>

              <li className="flex items-center">

                <FaPhone className="text-gray-500 mr-2" />

                Phone Number: (+88) 123-456-789

              </li>

            </ul>

          </div>

          {/\* Skills Section \*/}

          <div className="bg-white shadow-md rounded-lg p-6">

            <h3 className="text-xl font-bold mb-4">Skills</h3>

            <div className="flex flex-wrap">

              <span className="bg-gray-200 text-sm text-gray-600 px-4 py-2 mr-4 mb-4 rounded-lg">

                Patient Care

              </span>

              <span className="bg-gray-200 text-sm text-gray-600 px-4 py-2 mr-4 mb-4 rounded-lg">

                Nursing

              </span>

              <span className="bg-gray-200 text-sm text-gray-600 px-4 py-2 mr-4 mb-4 rounded-lg">

                CPR

              </span>

              <span className="bg-gray-200 text-sm text-gray-600 px-4 py-2 mr-4 mb-4 rounded-lg">

                Medical Assistance

              </span>

            </div>

          </div>

        </div>

      </div>

    </div>

  );

};

export default ViewProfilePageReal;

const mongoose = require('mongoose');

const bcrypt = require('bcryptjs');

const UserSchema = new mongoose.Schema({

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

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

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

  role: { type: String, enum: ['freelancer', 'client'], default: 'freelancer', required: true },

  portfolio: [String],  // Only freelancers will use this

  rating: { type: Number, default: 0 }, // Average rating

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

});

// Password hashing before saving

UserSchema.pre('save', async function (next) {

  if (!this.isModified('password')) {

    return next();

  }

  const salt = await bcrypt.genSalt(10);

  this.password = await bcrypt.hash(this.password, salt);

  next();

});

const User = mongoose.model('User', UserSchema);

module.exports = User;

const mongoose = require('mongoose');

const profileSchema = new mongoose.Schema({

  user: {

    type: mongoose.Schema.Types.ObjectId,

    ref: 'User',

    required: true

  },

  bio: {

    type: String,

    required: true

  },

  skills: {

    type: [String],

    required: true

  },

  contact: {

    email: String,

    phone: String,

  },

  socialLinks: {

    linkedin: String,

    github: String,

    website: String

  },

  profilePicture: {

    type: String,

    default: ''

  },

  experience: {

    type: Number,

    default: 0

  }

}, { timestamps: true });

module.exports = mongoose.model('Profile', profileSchema);

const Profile = require('../models/Profile');

const multer = require('multer');

const path = require('path');

const fs = require('fs');

// Ensure the 'uploads/profile-pictures' directory exists

if (!fs.existsSync('uploads/profile-pictures')) {

  fs.mkdirSync('uploads/profile-pictures', { recursive: true });

}

// Set up multer storage for profile pictures

const storage = multer.diskStorage({

  destination: (req, file, cb) => {

    cb(null, 'uploads/profile-pictures');

  },

  filename: (req, file, cb) => {

    cb(null, `${req.user.id}-${Date.now()}${path.extname(file.originalname)}`);

  },

});

const upload = multer({ storage });

// Create or update profile

const createOrUpdateProfile = async (req, res) => {

  const { userId, bio, skills, email, phone, linkedin, github, website } = req.body;

  try {

    let profile = await Profile.findOne({ user: userId });

    if (profile) {

      profile.bio = bio;

      profile.skills = skills.split(',').map(skill => skill.trim());

      profile.contact = { email, phone };

      profile.socialLinks = { linkedin, github, website };

      if (req.file) {

        profile.profilePicture = `/uploads/profile-pictures/${req.file.filename}`;

      }

      await profile.save();

      return res.json(profile);

    } else {

      profile = new Profile({

        user: userId,

        bio,

        skills: skills.split(',').map(skill => skill.trim()),

        contact: { email, phone },

        socialLinks: { linkedin, github, website },

        profilePicture: req.file ? `/uploads/profile-pictures/${req.file.filename}` : ''

      });

      await profile.save();

      return res.status(201).json(profile);

    }

  } catch (error) {

    console.error('Error creating or updating profile:', error);

    res.status(500).json({ message: 'Error creating or updating profile' });

  }

};

// Get profile by user ID

const getProfile = async (req, res) => {

  try {

    const profile = await Profile.findOne({ user: req.params.id }).populate('user', ['name', 'email']);

    if (!profile) {

      return res.status(404).json({ message: 'Profile not found' });

    }

    res.json(profile);

  } catch (error) {

    console.error('Error fetching profile:', error);

    res.status(500).json({ message: 'Error fetching profile' });

  }

};

// Update profile

const updateProfile = async (req, res) => {

  const { bio, skills, email, phone, linkedin, github, website } = req.body;

  try {

    const profilePicture = req.file ? `/uploads/profile-pictures/${req.file.filename}` : null;

    const updatedProfile = await Profile.findOneAndUpdate(

      { user: req.user.id },

      {

        bio,

        skills: skills.split(',').map(skill => skill.trim()),

        contact: { email, phone },

        socialLinks: { linkedin, github, website },

        profilePicture

      },

      { new: true }

    );

    res.status(200).json(updatedProfile);

  } catch (error) {

    console.error('Error updating profile:', error);

    res.status(500).json({ message: 'Error updating profile' });

  }

};

module.exports = {

  createOrUpdateProfile,

  getProfile,

  updateProfile,

  upload

};

const express = require('express');

const router = express.Router();

const { createOrUpdateProfile, getProfile, updateProfile, upload } = require('../controllers/profileController');

const protect  = require('../middleware/authMiddleware')

// Create or update profile (with file upload)

router.post('/', protect, upload.single('profilePicture'), createOrUpdateProfile);  // Create or update profile

router.get('/:id', getProfile);  // Get profile by ID

// Update profile (with file upload)

router.put('/edit',protect, upload.single('profilePicture'), updateProfile);

module.exports = router;

ok he