Admin

```
import { Outlet } from "react-router-dom";
import SideBar from "./SideBar";
import AdminNav from "./AdminNav";



const Admin = () => {

  return (
    <div className="admin-container">
      <SideBar />
      <div className="admin-inner-container">
        <AdminNav />
        <Outlet />
      </div>


    </div>
  );
}

export default Admin;
```

```
import { useState } from "react";
import sun from '../../images/light-mode.png'
import moon from '../../images/dark-mode.png'
import { useLocation, useNavigate } from "react-router-dom";
import homeWhite from '../../images/home-white.png'
import homeDark from '../../images/home-dark.png'

const AdminNav = () => {
    const [darkTheme, setDarkTheme] = useState(false)

    const [title, setTitle] = useState('')

    const location = useLocation()

    const navigate = useNavigate()


    function toggleTheme() {

        setDarkTheme(prevTheme => !prevTheme)

        document.body.classList.toggle("light-theme")

    }


    const loc = location.pathname


    return (

        <div className="admin-nav-container">

            <h2 className="location">{loc === '/admin/users' ? 'Users' :

                loc === '/admin' ? 'Dashboard':

                loc === '/admin/departments' ? 'Departments' :

                loc === '/admin/complaints' ? 'Complaints' : ''}

            </h2>

            <div className="admin-inner-nav-container">

                <h2 className="profile-name admin-nav-elements">Welcome {localStorage.getItem('fname')}</h2>

                <img className="theme-image admin-nav-elements" onClick={() => navigate('/')} src={darkTheme ? homeDark : homeWhite} alt="" />

                <img className="theme-image admin-nav-elements" onClick={toggleTheme} src={darkTheme ? sun : moon} alt="" />

            </div>

        </div>

    );

}
```

```
export default AdminNav;
```

```
import axios from "axios"
import { useEffect, useState } from "react"

const Complaints = () => {
  const [data, setData] = useState('')

  useEffect(() => {
    axios.get('http://localhost:2005/complaintList')
      .then(response => {
        setData(response.data)
      })
  }, [])

  return (
    <div className="complaint-list-container-admin">

    {/* Check if data is an array and map over it */}
    {data.length > 0 ? (
      data.map((complaint, index) => (
        <div key={index} className="complaint-list-item">

          <p>Complaint description: {complaint.description}</p>
          <p>Complaint severity: {complaint.severity}</p>
          <p>{complaint.attachment}</p>
          <p>To department: {complaint.department}</p>
          <img src="complaint.attachment" alt="" />
          {/* Display other complaint properties */}
```

```
                </div>
            ))
        ) : (
            <p>No complaints found.</p>
        )}
    </div>    );
}



export default Complaints;
```

```
import axios from "axios"
import { useState, useEffect } from "react"
import SimpleLineChart from './SimpleLineChart'



const Dashboard = () => {
    const [userCount, setUserCount] = useState(0)
    const [departmentCount, setDepartmentCount] = useState(0)
    const [complaintCount, setComplaintCount] = useState(0)
    const [data, setData] = useState({
        users: [],
        departments: [],
        complaints: []
    })
    const [clicked, setClicked] = useState(1)

    useEffect(()=>{
        //user count
```

```
axios.get('http://localhost:2005/userList')
  .then(result => {
    setUserCount(result.data.length)
    setData(prevState => {
      return ({
        ...prevState,
        users: result.data
      })
    })
    // console.log(data.complaints)
  })
  .catch(err => err)

  // department count
  axios.get('http://localhost:2005/departmentList')
  .then(result => {
    setDepartmentCount(result.data.length)
    setData(prevState => {
      return ({
        ...prevState,
        departments: result.data
      })
    })
  })
  .catch(err => err)

  //complaints
  axios.get('http://localhost:2005/complaintList')
  .then(result => {
      setComplaintCount(result.data.length)
      setData(prevState => {
        return ({
          ...prevState,
          complaints: result.data
        })
      })
    })
    .catch(err => err)
}, [])

return (
  <div className="dashboard-container">
    <div className="top-container">
      <div className="div-1 metrics" onClick={() => setClicked(1)}>
        <h2>Users</h2>
        <h2>{userCount}</h2>
      </div>
      <div className="div-2 metrics" onClick={() => setClicked(2)}>
        <h2>Departments</h2>
        <h2>{departmentCount}</h2>
      </div>
      <div className="div-3 metrics" onClick={() => setClicked(3)}>
        <h2>Complaints</h2>
        <h2>{complaintCount}</h2>
      </div>
    </div>
    <div className="bottom-container">
      <SimpleLineChart data={data} clicked={clicked} />
      </div>
    </div>
  );
}

export default Dashboard;
```

```
import axios from "axios"
import { Toaster, toast } from "sonner"
import { useEffect, useState } from "react"
import deleteWhite from '../../images/delete-white.png'
import deleteDark from '../../images/delete-dark.png'
import editWhite from '../../images/edit-white.png'
import editDark from '../../images/edit-dark.png'
import saveDark from '../../images/save-dark.png'
import saveWhite from '../../images/save-white.png'
import cancelWhite from '../../images/cancel-white.png'
import cancelDark from '../../images/cancel-dark.png'

const Departments = () => {
  const [data, setData] = useState([])
  const [newData, setNewData] = useState([])
  const [isOpen, setIsOpen] = useState(false)
  const [formData, setFormData] = useState({
    name:'',
    head: '',
    members: ''
  })

  const togglePopup = async() => {
```

```
  setIsOpen(prevState => !prevState)
  setFormData({})
}

const handleChange = (event) => {
  setFormData(prevData => {
    return ({
      ...prevData,
      [event.target.name]: event.target.value
    })
  })
}

const handleAddSubmit = async (e) => {
  e.preventDefault()
  try {
    const newDep = await axios.post(`http://localhost:2005/department/`, formData)
    console.log(newDep)
    togglePopup()
    toast.success('New Department Successfully added')
  } catch (error) {
    console.log(error)
    toast.error('error occured')
  }
}

const handleDelete = async (id) => {
  try {
    const result = await axios.delete(`http://localhost:2005/deleteDepartment/${id}`)
```

```
     console.log(result)
     setNewData((prevItems) => prevItems.filter(item => item._id !== id));
     toast.success('Department successfully deleted')
   } catch (error) {
     console.log(error.message)
     toast.error('there was an error please try again')
   }

 }



const [editDepId, setEditDepId] = useState(null);
const [editFormData, setEditFormData] = useState({
 name: '',
 head: '',
 members: ''
});

 useEffect(()=>{
   axios.get('http://localhost:2005/departmentList')
     .then(result => {
       setData(result.data)
     })
     .catch(err => console.log(err))
 }, [editDepId, newData, isOpen])

 const handleFormChange = (e) => {
   setEditFormData({
    ...editFormData,
    [e.target.name]: e.target.value,
   });
 }

 const handleEditClick = (department) => {
   setEditDepId(department._id);
   setEditFormData(department);
  };

  const handleFormSubmit = async (e) => {
   e.preventDefault();
   try {
    const response = await axios.patch(` http://localhost:2005/updateDepartment/${editDepId}` , editFormData);
    setData(data.map(dep => (dep._id === editDepId ? response.data : dep)));
    setEditDepId(null); // Exit edit mode
    toast.success('successfull')
   } catch (error) {
    console.error('Error updating user:', error);
    toast.error('error occured')
   }
  };

 return (
   <div className="departments-container">
     <Toaster richColors expand={false} position="bottom-center" />
     <div className="inner-container">
       <table className="departments-table">
```

```
          <thead className="table-head">
            <tr>
              <th className="table-head-data r-border">ID</th>
              <th className="table-head-data r-border">Department Name</th>
              <th className="table-head-data r-border">Department Head</th>
              <th className="table-head-data ">Members</th>
            </tr>
          </thead>
          <tbody>
            {data.length > 0 ? (
              data.map( department => (

                <tr key={department._id} className="table-row">
                  {editDepId === department._id ? (
                    <>
                      <td className="table-row-data r-border">{department._id}</td>
                      <td className="table-row-data r-border">
                        <input className="table-inputs inputs em-pas" type="text" name='name' value={editFormData.name} onChange={handleFormChange}/>
                      </td>
                      <td className="table-row-data r-border">
                        <input type="text" className="table-inputs inputs em-pas" name='head' value={editFormData.head} onChange={handleFormChange}/>
                      </td>
                      <td className="table-row-data td-last">
                      <input type="text" name='members' className="table-inputs inputs em-pas" value={editFormData.members} onChange={handleFormChange}/>
                      </td>
                      <div className="update-div">
                        <img src={saveDark} className="row-btn" onClick={handleFormSubmit} />
                        <img src={cancelDark} className="row-btn" onClick={() => setEditDepId(null)} />
                      </div>
                      </>
                    ) : (
                    <>
                      <td className="table-row-data r-border">{department._id}</td>
                      <td className="table-row-data r-border">{department.name}</td>
                      <td className="table-row-data r-border">{department.head}</td>
                      <td className="table-row-data">{department.members}</td>
                      <div className="hidden-btn">
                        <img src={editDark} className="row-btn" onClick={()=> handleEditClick(department)} alt="" />
                        <img src={deleteDark} className="row-btn" onClick={()=>handleDelete(department._id)} alt="" />
                      </div>
                    </>
                  )}
                </tr>

              ))
            ) : (
              <p>No departments found.</p>
            )}
          </tbody>
        </table>

        <button className="dep-btn add-btn" onClick={togglePopup}>+ Add Department</button>
        {isOpen &&
          <form onSubmit={handleAddSubmit} className="department-form">
            <label htmlFor="name">Name</label>
            <input type="text" id="name" name="name" value={formData.name} className="inputs em-pas" required onChange={handleChange} />
            <label htmlFor="head">Head</label>
            <input type="text" id="head" name="head" value={formData.head} className="inputs em-pas" required onChange={handleChange} />
            <label htmlFor="members">Members</label>
            <input type="text" id="members" name="members" value={formData.members} className="inputs em-pas" required onChange={handleChange} />
            {/* later make this a select with list of checkbexes nested inside options */}
            <div className="add-btn-div">
              <button className="dep-inputs dep-btn dep-one" onClick={togglePopup}>Cancel</button>
              <input type="submit" name="submit" value='Submit' className="dep-inputs dep-btn dep-two" />
            </div>
          </form>
        }
      </div>


    </div>
  );
}
export default Departments;
```

```
import dashboardDark from '../../../images/dashboard-dark.png'
import dashboardLight from '../../../images/dashboard-white.png'
import loginDark from '../../../images/login-dark.png'
import loginLight from '../../../images/login-white.png'
import usersDark from '../../../images/users-dark.png'
import usersLight from '../../../images/users-white.png'
import logoutDark from '../../../images/logout-dark.png'
import logoutLight from '../../../images/logout-white.png'
import departmentDark from '../../../images/department-dark.png'
import deprtmentLight from '../../../images/department-white.png'
import complaintDark from '../../../images/complaint-dark.png'
import complaintLight from '../../../images/complaint-white.png'
import { Link, Outlet, useNavigate } from 'react-router-dom'


const SideBar = () => {

    const navigate = useNavigate()
    const handleLogout = () => {
    localStorage.removeItem('token')
    localStorage.removeItem('email')
    localStorage.removeItem('fname')
    localStorage.removeItem('role')
    navigate('/login')
}


    return (
        <>
            <div className="sidebar-container">
                <ul>
                    <li className='sidebar-item'><Link className='sidebar-link' to='/admin'><img className='sidebar-img' src={dashboardLight} alt="" />Dashboard</Link></li>
```

```
          <li className='sidebar-item'><Link className='sidebar-link' to='/admin/users'><img className='sidebar-img' src={usersLight} alt=""/>Users</Link></li>
          <li className='sidebar-item'><Link className='sidebar-link' to='/admin/departments'><img className='sidebar-img' src={deprtmentLight} alt=""/>Departments</Link></li>
          <li className='sidebar-item'><Link className='sidebar-link' to='/admin/complaints'><img className='sidebar-img' src={complaintLight} alt=""/>Complaints</Link></li>
        </ul>
        <ul>
          <li className='sidebar-item last-sidebar-item'><Link className='sidebar-link' to='/login' onClick={handleLogout} ><img className='sidebar-img' src={logoutLight} alt='' />Logout</Link></li>
        </ul>
      </div>
    </>
  );
}

export default SideBar;
```

```
import React, { useState, useEffect } from 'react';
import { LineChart, Line, XAxis, YAxis, CartesianGrid, Tooltip, Legend, ResponsiveContainer } from 'recharts';


// const data = [
//   { name: 'Jan', uv: 4000, pv: 2400, amt: 2400 },
//   { name: 'Feb', uv: 3000, pv: 1398, amt: 2210 },
//   { name: 'Mar', uv: 2000, pv: 9800, amt: 2290 },
//   { name: 'Apr', uv: 2780, pv: 3908, amt: 2000 },
// ];
```

```
const SimpleLineChart = (props) => {

    const [selected, setSelected] = useState(props.data.users)
    // Update selected data based on `props.clicked`
  useEffect(() => {
    if (props.clicked === 1) {
      setSelected(props.data.users);
    } else if (props.clicked === 2) {
      setSelected(props.data.departments);
    } else {
      setSelected(props.data.complaints); // Fixed typo from "compliants" to "complaints"
    }
  }, [props.clicked, props.data]); // Re-run effect when `clicked` or `data` changes



    console.log(selected)
  return (
    <ResponsiveContainer width="100%" height={400}>
      <LineChart data={selected} margin={{ top: 5, right: 30, left: 20, bottom: 5 }}>
        <CartesianGrid strokeDasharray="3 3" />
        <XAxis dataKey="createdAt" />
        <YAxis domain={[0, selected.length]} />
        {/* {console.log(props.data.createdAt)} */}
        <Tooltip labelFormatter={(value) => `Date: ${new Date(value).toLocaleDateString()}` } />
        <Legend />
        <Line type="monotone" dataKey="name" activeDot={{ r: 8 }}  stroke="#000000" strokeWidth={1}
/>
      </LineChart>
  </ResponsiveContainer>
 );
};

export default SimpleLineChart;
```

```
import axios from "axios";
import { useEffect, useState } from "react";
import deleteWhite from '../../images/delete-white.png'
import deleteDark from '../../images/delete-dark.png'
import editWhite from '../../images/edit-white.png'
import editDark from '../../images/edit-dark.png'
import { Toaster, toast } from "sonner";
import saveDark from '../../images/save-dark.png'
import saveWhite from '../../images/save-white.png'
import cancelWhite from '../../images/cancel-white.png'
import cancelDark from '../../images/cancel-dark.png'

const Users = () => {
  const [data, setData] = useState('')
  const [editUserId, setEditUserId] = useState(null)
  const [editFormData, setEditFormData] = useState({
    fname: '',
    lname: '',
    email: '',
    role: ''
   });



  useEffect(()=>{
    axios.get('http://localhost:2005/userList')
      .then(result => {
        setData(result.data)
      })
      .catch(err => err)
  }, [editUserId])

  const handleFormChange = (e) => {
    setEditFormData({
     ...editFormData,
     [e.target.name]: e.target.value,
    });
  }

  const handleEditClick = (department) => {
    setEditUserId(department._id);
    setEditFormData(department);
   };

   const handleFormSubmit = async (e) => {
    e.preventDefault();
    try {
     const response = await axios.patch(`http://localhost:2005/updateUser/${editUserId}`,
editFormData);
     setData(data.map(user => (user._id === editUserId ? response.data : user)));
     setEditUserId(null); // Exit edit mode
     toast.success('User successfull updated')
    } catch (error) {
     console.error('Error updating user:', error);
     toast.error('error occured')
    }
   };

  const handleDelete = async (id) => {
    try {
      const result = await axios.delete(` http://localhost:2005/deleteUser/${id}` )
      console.log(result)
      setData((prevItems) => prevItems.filter(item => item._id !== id));
      toast.success('User successfully deleted')
    } catch (error) {
      console.log(error.message)
      toast.error('there was an error please try again')
    }
  }

  return (
    <div className="users-container">
      <Toaster richColors expand={false} position="bottom-center" />
      <table className="users-table">
        <thead className="table-head">
          <th className="table-head-data r-border">ID</th>
          <th className="table-head-data r-border">First Name</th>
          <th className="table-head-data r-border">Last Name</th>
          <th className="table-head-data r-border">Email</th>
          <th className="table-head-data">Role</th>
        </thead>
        <tbody>
          {data.length > 0 ? (
              data.map(user => (

                <tr key={user._id} className="table-row">
                  {editUserId === user._id ? (
                    <>
                    <td className="table-row-data r-border">{user._id}</td>
                    <td className="table-row-data r-border">
                      <input className="table-inputs inputs em-pas" type="text" name='fname' value={editFormData.fname} onChange={handleFormChange}/>
                    </td>
                    <td className="table-row-data r-border">
                      <input type="text" className="table-inputs inputs em-pas" name='lname' value={editFormData.lname} onChange={handleFormChange}/>
                    </td>
                    <td className="table-row-data r-border">
                    <input type="text" name='email' className="table-inputs inputs em-pas" value={editFormData.email} onChange={handleFormChange}/>
                    </td>
                    <td className="table-row-data td-last">
                      <select className="table-inputs inputs em-pas" name="role" onChange={handleFormChange} value={editFormData.role}>
                        <option value="admin" >admin</option>
                        <option value="head" >head</option>
                        <option value="user" >user</option>
                      </select>
                    {/* <input type="text" name='email' className="table-inputs inputs em-pas" value={editFormData.email} onChange={handleFormChange}/> */}
                    </td>
                    <div className="update-div">
                      <img src={saveDark} className="row-btn" onClick={handleFormSubmit} />
                      <img src={cancelDark} className="row-btn" onClick={() => setEditUserId(null)} />
                    </div>
                    </>
                  ):(
                  <>
                    <td className="table-row-data r-border">{user._id}</td>
                    <td className="table-row-data r-border">{user.fname}</td>
                    <td className="table-row-data r-border">{user.lname}</td>
                    <td className="table-row-data r-border">{user.email}</td>
                    <td className="table-row-data">{user.role}</td>
                    <div className="hidden-btn">
                        <img src={editDark} className="row-btn" onClick={() => handleEditClick(user)} alt="" />
                        <img src={deleteDark} className="row-btn" onClick={() => handleDelete(user._id)} alt="" />
                      </div>
                    </>
                  )}
                </tr>

              ))
            ) : (
              <p>No users found.</p>
            )}
          </tbody>
        </table>
      </div>
    );
  }
```

```
export default Users;
```

Authentication

```
function ForgetPassword() {
  return(
    <div className="apps-container">
      <div className="forget-password-container">
        <form action="#">
          <label htmlFor="reset-email">Enter your email</label>
          <input type="email" required placeholder="john@example.com" className="inputs em-pas" name="reset-email" />
          <input type="submit" name="submit" className="inputs btn btn1" value="Send reset link" />
        </form>
      </div>
    </div>
  )
}



export default ForgetPassword
```

```
import photo from './google-icon.png'
import React, {useEffect, useState} from 'react'
import Register from './Register';
import { Link, useNavigate } from 'react-router-dom';
import axios from 'axios';
import { toast, Toaster } from 'sonner';
```

```
function Login(props) {
  const [formData, setFormData] = useState({
    email: "",
    password: ""
  })
  const [isLoading, setIsLoading] = useState(false)
  const navigate = useNavigate()
  const [showPassword, setShowPassword] = useState(false);

    const togglePasswordVisibility = () => {
      setShowPassword((prevState) => !prevState);
    }

  const handleChange = (event) => {
    setFormData(prevData => {
      return ({
        ...prevData,
        [event.target.name]: event.target.value
      })
    })
  }

  const handleClick = () => {
    navigate("/Register")
  }
```

```
const handleSubmit = (e) => {
  e.preventDefault()
  axios.post('http://localhost:2005/login', formData)
    .then(result => {
      setIsLoading(true)
      const token = result.data.token
      const email = result.data.email
      const fname = result.data.fname
      const role = result.data.role
      localStorage.setItem('token', token)
      localStorage.setItem('email', email)
      localStorage.setItem('fname', fname)
      localStorage.setItem('role', role)
      props.handleIsLoged()
      if(result.data.role === 'admin'){
        navigate('/admin')
      } else {
        navigate('/')
      }
    })
    .catch(err => {
      console.log(err)
      toast.error(err.response.data.message)
    })
}


return (
  <div className="apps-container">
```

```
      <Toaster richColors expand={false} position='bottom-center' />

      <div className="login-container">

        <h2>Welcome Back!</h2>

        <form onSubmit={handleSubmit}>

          <label htmlFor="username" >User Name</label>

          <input type="email" id="username" onChange={handleChange}  required
placeholder="john@example.com" className="inputs em-pas" name="email"
value={formData.email} />

          <label htmlFor="password">Password</label>

          <input type={showPassword ? "text" : "password"} onChange={handleChange} required
placeholder="password" id="password" className="inputs em-pas" value={formData.password}
name="password" />

          <div className="passwordVisibility">

          <input type="checkbox" id="passwordVisibility" onClick={togglePasswordVisibility}
name="passwordVisibility" />

            <label htmlFor="passwordVisibility" className="showPasswordLabel">
{showPassword ? "Hide password" : "Show password"}</label>

            <Link to={"/ForgetPassword"} className="forget-password">forget password?</Link>

          </div>


          <input type="submit" className="inputs btn btn1" name="Login" disabled={isLoading}
value={isLoading ? "Processing..." : "Login"} />


        </form>

        <h4><span>or</span></h4>

        <div className="login-options">

          {/* <button className="btn button" ><img src={photo} alt="google icon"
className="google-icon" /><p>Login with Gmail</p></button> */}

          <button className="btn create-account button" onClick={handleClick} >Create a new
account</button>

        </div>

      </div>
```

```
        </div>
    )
}

export default Login;
```

```
import axios from "axios";
import React, {useState} from "react";
import { useNavigate } from "react-router-dom";
import { toast, Toaster } from "sonner";
import * as Yup from "yup"

function Register() {
    const [showPassword, setShowPassword] = useState(false);
    const [error, setError] = useState({})
    const [userData, setUserData] = useState({
        fname: "",
        lname: "",
        email: "",
        password: "",
        confirmPassword: ""
    })
    const navigate = useNavigate()

    // Validation schema
    const schema = Yup.object().shape({
        fname: Yup.string().required('Name is required'),
        lname: Yup.string().required('Name is required'),
        email: Yup.string().email('Invalid email').required('Email is required'),
    password: Yup.string().min(6, 'Password must be at least 6 characters').required('Password is required'),
    confirmPassword: Yup.string().oneOf([Yup.ref('password'), null], 'Passwords must match').required('Confirm password is required')
   });



  const handleChange = (event) => {
    setUserData(prevUser => {
      return ({
        ...prevUser,
        [event.target.name]: event.target.value
      })
    })
  }

  const togglePasswordVisibility = () => {
    setShowPassword((prevState) => !prevState);
  }

  const handleSubmit = async (e) => {
    e.preventDefault()
    setError({})
    try {
      //validating form data
      await schema.validate(userData, {abortEarly: false})

      //call an api if valid
      axios.post('http://localhost:2005/users', userData)
      .then(response => console.log(response))
      .catch(error => console.log(error.message))
      console.log("success")
      navigate('/login')
    } catch (err) {
      const newErrors = {};
      err.inner.forEach((error) => {
       newErrors[error.path] = error.message;
      });
      setError(newErrors);
      toast.error(newErrors.confirmPassword)
    }
  }

  return (
    <div className="apps-container">
      <Toaster richColors expand={false} position="bottom-center" />
      <div className="register-container">
        <form onSubmit={handleSubmit} >
          <label htmlFor="fname">First Name</label>
          <input type="text" id="fname" className="inputs em-pas" minLength={3} required name="fname" onChange={handleChange} />
          <label htmlFor="lname">Last Name</label>
          <input type="text" id="lname" className="inputs em-pas" minLength={2} required name="lname" onChange={handleChange} />
          <label htmlFor="email">Email</label>
          <input type="email" id="email" required placeholder="john@example.com" className="inputs em-pas" name="email" onChange={handleChange} />
            <label htmlFor="password">Password</label>
            <input type={showPassword ? "text" : "password"} required placeholder="password" id="password" className="inputs em-pas" name="password" onChange={handleChange} />

            <div className="passwordVisibility">

            <input type="checkbox" id="passwordVisibility" onClick={togglePasswordVisibility} name="passwordVisibility" />

              <label htmlFor="passwordVisibility" className="showPasswordLabel">{showPassword ? "Hide password" : "Show password"}</label>

            </div>

          <label htmlFor="#confirm-password">Confirm Password</label>

          <input type={showPassword ? "text" : "password"} id="confirm-password" className="inputs em-pas" required placeholder="Confirm password" onChange={handleChange} name="confirmPassword" />

          <input type="submit" className="btn1 inputs" name="create-account" value="Create account" />

        </form>

      </div>

    </div>

  )

}


export default Register
```

---

complaint submission

```
import axios from "axios"

import { useEffect, useState } from "react"

import { Toaster, toast } from 'sonner';




function Complaint() {

  const [depList, setDepList] = useState([])

  const [formData, setFormData] = useState({
```

```
 department: "",

 severity: "",

 description: "",

 file: null,

 reporter: localStorage.getItem("email") || ''

})



useEffect (() => {

 axios.get('http://localhost:2005/departmentList')

  .then(response => {

   setDepList(response.data)

  })

  .catch (error => {

   console.log(error)

  toast.error(error.response.data.message || 'error fetching departments')

  })


 }, [])


console.log(formData)


console.log(formData.reporter)


function handleChange(event) {

 setFormData((prevData) => {

  return {

   ...prevData,

   [event.target.name]: event.target.value,
    }
  })
}

function handleFileChange (event) {
  setFormData(prevState => {
    return ({
      ...prevState,
      file : event.target.files[0]
    })
  })
}

function handleSubmit(e) {
  e.preventDefault()

  const data = new FormData()
  data.append('department', formData.department)
  data.append('severity', formData.severity)
  data.append('description', formData.description)
  data.append('reporter', formData.reporter)
  data.append('file', formData.file)

  axios
    .post("http://localhost:2005/complaint", data, {
      headers: {
        'Content-Type': 'mutipart/form-data'
      }
    })
  .then(response => {
   console.log(response.data)
   toast.success("your complaint is successfully submited")
  })
  .catch(error => {
   console.log(error.message)
   toast.error("there was an error please try again")
})
}

return (
 <div className="apps-container">
  <div className="complaint-container">
   <form onSubmit={handleSubmit}>
    <label htmlFor="departments">Departments</label>
    <select
     id="departments"
     className="inputs em-pas"
     name="department"
     required
     onChange={handleChange}
     value={formData.department}
    >
     <option value="" disabled>
      Select department
     </option>
     <option value="admin">Admin </option>
     {depList.map(dep => (
      <option key={dep._id} value={dep.name} >{dep.name}</option>
  ))}
</select>
<label htmlFor="severity">severity</label>
<select
  id="severity"
  className="inputs em-pas"
  name="severity"
  required
  onChange={handleChange}
  value={formData.severity}
>
  <option value="" disabled>
    Select severity
  </option>
  <option value="very sever">Very severe</option>
  <option value="severe">Sever</option>
  <option value="normal">Normal</option>
  <option value="easy">Easy</option>
</select>

<label htmlFor="desc">Description</label>
<textarea
  name="description"
  className="inputs em-pas txt-area"
  required
  id="description"
  placeholder="describe your complaint"
  onChange={handleChange}
></textarea>
```

```
     <label htmlFor="file">Attach file</label>

     <input

      type="file"

      id="file"

      className="inputs"

      name="file"

      // value={formData.file}

      onChange={handleFileChange}

     />

     <input

      type="submit"

      className="inputs btn btn1"

      name="submit"

      value="Submit"

     />

    </form>

    <Toaster position="bottom-center" expand={false} richColors />

   </div>

  </div>

 )

}


export default Complaint
```

```
import axios from "axios";

import { useEffect, useState } from "react";


const ComplaintList = () => {


  const [data, setData] = useState('')
```

```
useEffect(() => {
  axios.get('http://localhost:2005/complaintList')
    .then(response => {
      setData(response.data)
    })
}, [])

return (
  <div className="complaint-list-container">
  <h1>Complaint List</h1>

  {/* Check if data is an array and map over it */}
  {data.length > 0 ? (
    data.map((complaint, index) => {
      if(complaint.status === 'read'){
        return (
          <div key={index} className="complaint-list-item">

            <p><strong>Complaint description:</strong> {complaint.description}</p>
            <p><strong>Complaint severity:</strong> {complaint.severity}</p>
            <p>{complaint.file}</p>
            <p><strong>To department:</strong> {complaint.department}</p>
            <p><strong>Solution:</strong> {complaint.solution}</p>

            {/* Display other complaint properties */}
          </div>
        )
      }
      })
    ) : (
      <p>No complaints found.</p>
    )}
  </div>   );
}

export default ComplaintList;
```

---

departments

```
import Notifications from "./Notifications"
import DepComplaints from "./depComplaints"
import homeWhite from '../../images/home-white.png'
import homeDark from '../../images/home-dark.png'
import sun from '../../images/light-mode.png'
import moon from '../../images/dark-mode.png'
import { useNavigate } from "react-router-dom"
import { useEffect, useState } from "react"
import axios from "axios"

const DepartmentPanel = () => {

 const [darkTheme, setDarkTheme] = useState(false)
 const [refresher, setRefersher] = useState(true)
 const [department, setDepartment] = useState([])

 const navigate = useNavigate()

 useEffect(()=> {
  axios.get(`http://localhost:2005/getDepHead/${localStorage.getItem('fname')}`)
```

```
   .then(response => {
    setDepartment(response.data)
    console.log(response.data)
   })
   .catch(error => {
    console.log(error)
   })
 }, [refresher])

 function handleRefresher() {
  setRefersher(prevState => !prevState)
 }

 function toggleTheme() {
  setDarkTheme(prevTheme => !prevTheme)
  document.body.classList.toggle("light-theme")
}

return (
  <div className="department-pannel-container">
   <div className="new-complaints">
    <div className="header">
    <img className="theme-image admin-nav-elements" onClick={() => navigate('/')} src={!darkTheme ? homeDark : homeWhite} alt="" />
    <h2>New Complaints</h2>
    <img className="theme-image admin-nav-elements" onClick={toggleTheme} src={darkTheme ? sun : moon} alt="" />
    </div>
    <Notifications
```

```
     handleRefresher={handleRefresher}
     refresher={refresher}
     department={department}
     />
   </div>
   <div className="old-complaints">
    <h2>Old Complaints</h2>
    <DepComplaints
     refresher={refresher}
     department={department}
     />
   </div>
  </div>
 )
}

export default DepartmentPanel
```

```
import axios from "axios";
import { useEffect, useState } from "react";

const DepComplaints = (props) => {
 const [data, setData] = useState([]);
 const [clickedId, setClickedId] = useState(null); // State to track the ID of the clicked item

 useEffect(() => {
  axios.get('http://localhost:2005/complaintList')
   .then(response => {
    setData(response.data);
   })
```

```
      .catch(error => console.error("Error fetching data:", error));
  }, [props.refresher]);


  const handleClick = (id) => {
    // Toggle expansion: if the same item is clicked again, collapse it; otherwise, expand it
    setClickedId(clickedId === id ? null : id);
  };


  return (
    <div className="department-complaints-container">
      {data.map(item => {
        if (props.department.length > 0 && item.department === props.department[0].name && item.status !== 'unread') {
          return (
            <div
              className="item-container"
              key={item._id}
              onClick={() => handleClick(item._id)}
            >
              <p><strong>Description: </strong>{item.description}</p>
              <p><strong>Created at: </strong>{new Date(item.createdAt).toLocaleDateString()}</p>


              {clickedId === item._id && ( // Show extra details if this item is clicked
                <>
                  <p><strong>Severity: </strong>{item.severity}</p>
                  <p><strong>Reporter: </strong>{item.reporter}</p>
                  <p><strong>Solution: </strong>{item.solution}</p>
                  {/* <button className="chat-btn" onClick={(e) => { e.stopPropagation();}}>Chat</button> */}
        </>
       )}
      </div>
     );
    }
    return null;
   })}
  </div>
 );
};

export default DepComplaints;
```

```
import axios from "axios";
import { useEffect, useState } from "react";
import arrowDark from '../../images/arow-dark.png'
import arrowWhite from '../../images/arow-white.png'
import arroUpWhite from '../../images/arrow-up-white.png'
import arrowUpDark from '../../images/arrow-up-dark.png'
import { toast, Toaster } from "sonner";

const Notifications = (props) => {
  const [messages, setMessages] = useState([])
  const [clickedId, setClickedId] = useState(null); // State to track the ID of the clicked item
  const [popUp, setPopUp] = useState(null)
  const [textArea, setTextArea] = useState({
    solution: ""
  })

  useEffect(() => {
```

```
 axios.get('http://localhost:2005/unreadedComplaintList')
    .then(response => {
      setMessages(response.data)
      // console.log(response.data)
    })
}, [props.refresher])

const handleSolved = (id) => {
  setPopUp(popUp === id ? null : id)
  setTextArea({
    solution: ""
  })
}

const handleClick = (id) => {
  // Toggle expansion: if the same item is clicked again, collapse it; otherwise, expand it
  setClickedId(clickedId === id ? null : id);
 };

const handleTextArea = (event) => {
  setTextArea({ solution: event.target.value })
}

const handlePost = (id) => {
  axios.put(` http://localhost:2005/solution/${id}` , textArea)
    .then(response => {
      console.log(response)
      toast.success('Posted successfully')
      props.handleRefresher()
```

```
        setTextArea({solution: ""})
      })
      .catch(error => {
        console.log(error)
        toast.error('there was a problem posting the solution')
        // console.log(id, textArea)
      })
  }
  // console.log(props.department.name)
  return (
    <div className="notifications-container">
      <Toaster richColors expand={false} position="bottom-center" />
      {messages.map(item => {
        if(props.department.length > 0 && item.status === 'unread' && item.department === props.department[0].name){
          return (
            <div className="item-outer-container">
            <div className="notification-item" key={item._id} >
              <p><strong>Description: </strong>{item.description}</p>
              <p><strong>Created at: </strong>{item.createdAt}</p>

              {clickedId === item._id && (
               <>
                  <p><strong>Severity: </strong>{item.severity}</p>
                  <p><strong>Reporter: </strong>{item.reporter}</p>
                  <div className="buttons">
                    {/* {popUp !== item._id && <button className="chat-btn" onClick={(e) => { e.stopPropagation(); }}>Chat</button>} */}
                    {popUp !== item._id && <button className="chat-btn" onClick={() => handleSolved(item._id)} >Solution</button>}
                  </div>

                  {popUp === item._id && (

                    <div className="solution">

                      <label htmlFor="txt">Solution:</label>

                      <textarea id="txt" className="inputs em-pas sol" onChange={e => handleTextArea(e)} value={textArea.solution} ></textarea>

                      <button className="chat-btn sol" onClick={() => handlePost(item._id)} >Post</button>

                    </div>

                  )}

               </>

              )}

              <img src={clickedId === item._id ? arrowUpDark : arrowDark} onClick={() => {

                handleClick(item._id)

                {clickedId === item._id && handleSolved(item._id)}

                }} className="arrow" alt="down arrow" />

            </div>

            </div>

          )

        }

      })}

    </div>

  );

}


export default Notifications;
```

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

import bg from '../images/home-bg.png'

import bgDark from '../images/home-bg.png'
```

```
const Home = () => {
  const navigate = useNavigate()
  return (
    <div className="home-container" >
      <img src={bg} alt="" className='bg' />
      <h1 className="cms-header">Welcome to <br/>Complaint Management System</h1>
      <div className="btn-container">
        <button className="comp-btn" onClick={()=> navigate('/complaint')}>Report a
complaint</button>
        <button className="comp-btn" onClick={() => navigate('/complaintList')}>Complaint
List</button>
        {localStorage.getItem('role') === 'admin' && <button className="comp-btn" onClick={() =>
navigate('/admin')}>admin</button> }
        {localStorage.getItem('role') === 'head' && <button className="comp-btn" onClick={() =>
navigate('/departmentPanel')}>Department Complaints</button> }
      </div>
    </div>
  );
}

export default Home;
```

```
import { Link, Outlet, useLocation, useNavigate } from "react-router-dom";


const NavBar = (props) => {
  const navigate = useNavigate()
  const location = useLocation()


  const handleClick = () => {
   navigate("/login")
  }
  const handleLogout = () => {
    localStorage.removeItem('token')
    localStorage.removeItem('email')
    localStorage.removeItem('fname')
    localStorage.removeItem('role')
    navigate('login')
  }

  const showLink = location.pathname !== '/'
  const showWelcome = location.pathname === '/'
  const fname = localStorage.getItem('fname')

  return (
    <div className="nav">
      <nav className="nav-list">
        <h1 className="logo">CMS</h1>
        <ul>
          {showLink && <>
            <li><Link className="nav-link" to={'/'} >Home</Link></li>
            <li><Link className="nav-link" to={'/complaint'} >Complaint</Link></li>
            <li><Link className="nav-link" to={'/ComplaintList'} >Complaint List</Link></li>
            </>
          }
          {props.isLoged && showWelcome ? <li>Welcome {fname}</li>: <li></li>}
        </ul>
        {props.isLoged ? <button className="nav-btn" onClick={handleLogout}>Logout</button> :
<button className="nav-btn" onClick={handleClick}>Signin/ Signup</button>}
      </nav>
      <Outlet />
    </div>
  );
}

export default NavBar;
```

```
import { Link } from "react-router-dom";


const NotFound = () => {
  return (
    <div className="not-found-container">
      <p>The Page You're Looking For Is Not Found</p>
      <Link to={'/'} >Go To Home</Link>
    </div>
  );
}

export default NotFound;
```

```
* {
 box-sizing: border-box;
}


:root {
 --primary-color: white;
 --secondary-color: #005ccc;
 --tertiary-color: black;
 --forth-color: #589df1;
}
```

```
.light-theme {
  --primary-color: rgb(185, 175, 175);
  --secondary-color: #4D8FD1;
  --tertiary-color: #121212;
  --forth-color: #357ABD;
}

body {
  margin: 0;
  height: 100vh;
  background-color: var(--primary-color);
}

/* ------- NavBar ------- */

nav {
  padding-top: 20px;
}

.logo {
  color: var(--secondary-color);
  font-size: 60px;
  margin: -8px 0 0 0;
  vertical-align: top;
}

.nav-list {
  width: 100%;
  height: 8vh;
  display: flex;
  justify-content: space-around;
  /* border-bottom: 3px solid black; */
}

.nav-btn {
  border: none;
  padding: 5px 15px;
  height: 50px;
  background-color: var(--secondary-color);
  color: var(--primary-color);
  border-radius: 10px;
}

.nav-btn:hover {
  cursor: pointer;
  /* transform: scale(1.05); */
  background-color: var(--forth-color);
}

.nav-list ul {
  display: flex;
  padding: 0;
  list-style-type: none;
}

.nav-link {
  color: var(--tertiary-color);
```

```css
  text-decoration: none;
  margin-left: 20px;
  height: 50px;
  padding: 24px;
}

.nav-link:hover,
nav-link {
  cursor: pointer;
  background: linear-gradient(
    0deg,
    var(--secondary-color),
    10%,
    var(--primary-color)
  );
  color: linear-gradient(
    180deg,
    var(--primary-color),
    10%,
    var(--secondary-color)
  );
}

/* ------ Home ------ */

.cms-header {
  color: var(--secondary-color);
  font-weight: 900;
  font-size: 60px;
  text-align: center;
}

.home-container {
  width: 100%;
  height: 86.5vh;
  display: flex;
  flex-direction: column;
  padding-top: 50px;
  position: relative;
  /* background-image: url("../public/home-bg.png"); */
  /* background-color: aqua; */
}

.bg {
  position: absolute;
  width: 100%;
  height: 100%;
  z-index: -1;
}

.btn-container {
  align-self: center;
  width: 500px;
  display: flex;
  justify-content: space-around;
  align-items: center;
  margin-top: 50px;
}
```

```
.comp-btn {
  border: none;
  background-color: var(--secondary-color);
  color: var(--primary-color);
  padding: 20px;
  border-radius: 10px;
}

.comp-btn:hover {
  cursor: pointer;
  background-color: var(--forth-color);
}

/* ----- Login, Register, Forget Password ----- */

.apps-container {
  width: 100%;
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  height: 89.8vh;
  margin: 0;
  display: flex;
  justify-content: center;
  align-items: center;
}

h2 {
  text-align: center;
  margin-bottom: 40px;
```

```
}

.login-container,
.register-container,
.forget-password-container,
.complaint-container {
  width: 400px;
  height: auto;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0px 0px 5px 5px lightgray;
  border: none;
  background-color: var(--primary-color);
}

.google-icon {
  width: 16px;
  margin-right: 5px;
}

form {
  display: flex;
  flex-direction: column;
}

.inputs {
  margin: 5px 0 20px 0;
  height: 40px;
  border-radius: 10px;
  border: 1px solid lightgrey;
}

.button {
  height: 40px;
  width: 100%;
  border-radius: 10px;
  border: 1px solid lightgrey;
  display: flex;
  justify-content: center;
  align-items: center;
}

.btn1 {
  background-color: var(--secondary-color);
  color: var(--primary-color);
  margin-bottom: 10px;
}

.em-pas {
  text-indent: 10px;
}

.em-pas:focus {
  box-shadow: 0px 0px 5px 5px lightskyblue;
  outline: none;
}

h4 {
  width: 100%;
  text-align: center;
  border-bottom: 1px solid lightgray;
  line-height: 0.1em;
  margin: 10px 0 20px;
}

h4 span {
  background: #fff;
  padding: 0 10px;
}

.btn:hover {
  cursor: pointer;
  transform: scale(1.02);
}

.passwordVisibility {
  margin-bottom: 20px;
  display: flex;
  align-items: center;
}

.create-account {
  margin-top: 20px;
}

.forget-password {
  text-decoration: none;
```

```
}

.showPasswordLabel {
  margin-right: auto;
}

input::file-selector-button {
  height: 40px;
  background-color: lightgray;
  border: none;
  border-radius: 10px;
  margin-left: 0px;
  text-indent: 0;
}

.txt-area {
  min-height: 100px;
}

/* ----- complaint-list ----- */
.complaint-list-container {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

.complaint-list-item {
  border-radius: 10px;
  margin-bottom: 20px;
  box-shadow: 5px 5px 10px 5px var(--forth-color);
  width: 80%;
  padding-left: 20px;
}

/* ---- side bar ---- */
.sidebar-container {
  width: 200px;
  height: 100vh;
  padding: 80px 0 50px 0;
  position: fixed;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  background-color: var(--secondary-color);
  border-right: 2px solid var(--primary-color);
  z-index: 2;
}

.sidebar-container ul {
  margin: 0;
  padding: 0;
  overflow: hidden;
}

.sidebar-item {
  list-style: none;
  display: flex;
```

```
}

.sidebar-img {
  width: 50px;
  margin: 30px 20px;
}

.sidebar-link {
  text-decoration: none;
  color: var(--primary-color);
  font-weight: 900;
  font-size: 15px;
  width: 100%;
  display: flex;
  align-items: center;
}

.sidebar-link:hover {
  background-color: var(--forth-color);
}

.sidebar-link:focus {
  background-color: var(--forth-color);
}

/* ----- Admin ----- */
.admin-container {
  display: flex;
}
```

```
.admin-inner-container {
  width: 100%;
}

/* ---- Dashboard ---- */
.dashboard-container {
  width: 100%;
  height: 100vh;
  padding: 90px 0 0 200px;
  display: flex;
  flex-direction: column;
}

.top-container {
  width: 100%;
  height: 30%;
  padding: 10px;
  display: flex;
  justify-content: space-between;
  background-color: var(--primary-color);
  border-bottom: 2px solid var(--tertiary-color);
}

.metrics {
  width: 33%;
  border-radius: 10px;
  margin: 10px;
  background-color: var(--secondary-color);
  color: var(--primary-color);
  display: flex;
  flex-direction: column;
  justify-content: center;
}

.metrics h2 {
  margin: 0 0 10px 0;
}

.bottom-container {
  width: 100%;
  height: 70%;
  padding: 100px 20px 20px 20px;
  background-color: var(--primary-color);
}

/* ---- Users ---- */
.users-container {
  width: 100%;
  height: 100vh;
  padding: 100px 20px 20px 220px;
}

.users-table {
  width: 100%;
}

.table-head,
```

```css
.table-row {
  height: 70px;
  box-shadow: 5px 5px 15px 5px #9ec0e9;
  border-radius: 10px;
}

.table-row:hover .hidden-btn {
  cursor: pointer;
  transform: scale(1.02);
  display: flex;
  align-items: center;
  justify-content: space-around;
}

.row-btn {
  width: 30px;
  margin-right: 30px 5px 0 0;
}

.row-btn:hover {
  transform: scale(1.12);
}

.table-head-data {
  text-indent: 25%;
  font-weight: 900;
  min-width: 150px;
}
```

```
.table-row-data {
  text-indent: 15%;
  overflow: hidden;

}

.td-last {
  padding-right: 100px;
}

.r-border {
  border-right: 3px solid #9ec0e9;
  border-radius: 10px 10px 10px 0;
}

/* ---- complaint list container ---- */
.complaint-list-container-admin {
  width: 100%;
  height: 90vh;
  padding: 100px 20px 20px 220px;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.complaint-list-item:hover {
  cursor: pointer;
  transform: scale(1.02);
}
```

```
/* ---- departments ---- */
.departments-container {
  width: 100%;
  height: 100vh;
  padding: 100px 20px 20px 220px;
  /* position: relative; */
}

.inner-container {
  position: relative;
}

.departments-table {
  width: 100%;
}

.hidden-btn {
  display: none;
  height: 70px;
  width: 70px;
  position: absolute;
  right: 50px;
}

.update-div {
  display: flex;
  justify-content: space-around;
  align-items: center;
  height: 70px;
```

```css
  width: 70px;
  position: absolute;
  right: 50px;
}

.dep-btn {
  border: none;
  padding: 5px 15px;
  height: 50px;
  background-color: var(--secondary-color);
  color: var(--primary-color);
  border-radius: 10px;
}

.add-btn {
  margin-top: 15px;
  margin-bottom: 15px;
}

.add-btn-div {
  width: auto;
  display: flex;
  justify-content: space-between;
  gap: 15px;
}

.dep-one, .dep-two {
  flex: 1;
}
```

```
.dep-btn:hover {
  cursor: pointer;
  background-color: var(--forth-color);
}

.department-form {
  width: 400px;
  height: auto;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0px 0px 5px 5px #9ec0e9;
  border: none;
  position: absolute;
  top: 70%;
  left: 40%;
  background-color: var(--primary-color);
  z-index: 2;
}

/* .dep-inputs {
  margin: 5px 0 20px 0;
  height: 40px;
  border-radius: 10px;
  border: 1px solid lightgrey;
} */

.table-inputs {
  width: 80%;
  margin: auto 0px;
}
/*
.edit-form {
  display: flex;
  width: 800px;
} */

/* --- Admin Nav Bar --- */
.admin-nav-container {
  width: 100%;
  height: 80px;
  padding-left: 200px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: var(--secondary-color);
  color: var(--primary-color);
  /* border-left: 2px solid var(--primary-color); */
  position: fixed;
}

.admin-inner-nav-container {
  height: 80px;
  display: flex;
  align-items: center;
}

.location {
  margin: 0 0 0 10%;
}

.admin-nav-elements {
  margin: 0 20px 0 0;
}

.theme-image {
  width: 50px;
  height: 50px;
}

.theme-image:hover {
  cursor: pointer;
  transform: scale(1.1);
}

/* --- chat ---- */
.chat-container {
  width: 100%;
  height: 100vh;
  padding: 50px;
  background-color: white;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
```

```
.chat-block-container {
  padding: 10px;
  height: 80%;
  width: 400px;
  border-radius: 10px;
  box-shadow: 0px 0px 5px 5px lightgray;
  border: none;
  margin-bottom: 20px;
}

.chat {
  background-color: #005ccc;
  margin-bottom: 20px;
  border-radius: 50px;
  padding: 5px;
  color: white;
}

.message-input {
  width: 330px;
  margin: 5px 0 20px 0;
  height: 40px;
  border-radius: 10px;
  border: 1px solid lightgrey;
  text-indent: 10px;
  display: inline;
}

.message-input:focus {
  box-shadow: 0px 0px 5px 5px lightskyblue;
  outline: none;
}
.chat-form {
  display: flex;
  flex-direction: row;
  width: 400px;
  justify-content: space-between;
}

#send-image {
  width: 50px;
  height: 50px;
  background-color: #005ccc;
  border-radius: 25px;
  padding: 5px;
  display: inline;
}

/* --- department pannel --- */
.department-pannel-container {
  width: 100vw;
  height: 100vh;
  display: flex;
  flex-direction: column;
}

.new-complaints {
  flex: 1;
  background-color: var(--primary-color);
}

.old-complaints {
  flex: 2;
  background-color: var(--primary-color);
}

.header {
  display: flex;
  justify-content: space-around;
  padding: 20px 20px;
}

/* --- department complaints --- */
.department-complaints-container {
  padding-top: 20px;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.item-container {
  border-radius: 10px;
  margin-bottom: 20px;
  box-shadow: 5px 5px 10px 5px var(--forth-color);
  width: 80%;
  padding-left: 20px;
  padding: 10px;
```

```css
}

.item-container:hover {
  transform: scale(1.02);
  cursor: pointer;
}

/* --- Notifications --- */
.notifications-container {
  padding-top: 20px;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.item-outer-container {
  width: 100%;
  display: flex;
  justify-content: center;
}

.notification-item {
  border-radius: 10px;
  margin-bottom: 20px;
  box-shadow: 5px 5px 10px 5px var(--forth-color);
  width: 80%;
  padding-left: 20px;
  padding: 10px;
}
```

```
.notification-item:hover {
  transform: scale(1.02);
}

.arrow {
  width: 30px;
  height: 30px;
  z-index: 2;
  margin-left: 47%;
  cursor: pointer;
  margin: 2px 47%;
}

.arrow:hover {
  transform: scale(1.3);
}

.bottons {
  display: flex;

}

.chat-btn {
  border: none;
  padding: 0px 105px;
  height: 50px;
  background-color: var(--secondary-color);
  color: var(--primary-color);
  border-radius: 25px;
  margin-right: 20px;
}

.chat-btn:hover {
  background-color: var(--forth-color);
  cursor: pointer;
}

.solution {
  display: flex;
  flex-direction: column;
  margin-top: 20px;
}

#txt {
  width: 70%;
}

.sol {
  max-width: 500px;
}

.not-found-container {
  margin: 30% 0 0 20%;
}
```

```
import './App.css';
import ForgetPassword from './components/Authentication/forgetPassword';
import Login from './components/Authentication/Login';
import Register from './components/Authentication/Register';
import ComplaintList from './components/ComplaintSubmission/complaintList';
import Complaint from './components/ComplaintSubmission/complaint';
import { Navigate, Route, Routes } from 'react-router-dom';
import NavBar from './components/Navbar';
import { useEffect, useState } from 'react';
import Home from './components/Home';
import Admin from './components/AdminPanel/Admin';
import Dashboard from './components/AdminPanel/Dashboard';
import Complaints from './components/AdminPanel/Complaints';
import Departments from './components/AdminPanel/Departments';
import Users from './components/AdminPanel/Users';
import Chat from './components/Departments/Chat';
import NotFound from './components/NotFound';
import Notifications from './components/Departments/Notifications';
import DepComplaints from './components/Departments/depComplaints';
import DepartmentPanel from './components/Departments/departmetnPanel';


function App() {
  const [isLoged, setIsLoged] = useState(false);

  useEffect(() => {
    // const token = localStorage.getItem('token');
    // if (token) {
    //   setIsLoged(true);
    // }
    const checkToken = () => {
      const token = localStorage.getItem('token');
      setIsLoged(!!token); // Set isLoged to true if token exists
    };
    checkToken();
    window.addEventListener('storage', checkToken); // For multi-tab detection
    return () => window.removeEventListener('storage', checkToken);
  }, []);

  const handleIsLoged = () => {
    setIsLoged(prevState => !prevState)
  }

  return (
    <div className="app-container">
      <Routes>
        {/* Department Routes */}
        <Route path='/departmentPanel' element={isLoged ? <DepartmentPanel /> : <Navigate replace to="/login" />} />
        <Route path='/notifications' element={isLoged ? <Notifications /> : <Navigate replace to="/login" />} />
        <Route path='/depComplaints' element={isLoged ? <DepComplaints /> : <Navigate replace to="/login" />} />
        {/* <Route path='/chat' element={isLoged ? <Chat /> : <Navigate replace to="/login" />} /> */}

        {/* Authentication Routes */}
        <Route path="/login" element={<Login handleIsLoged={handleIsLoged} />} />
        <Route path="/forget-password" element={<ForgetPassword />} />
        <Route path="/register" element={<Register />} />

        {/* Admin Routes */}
        <Route path="/admin" element={isLoged ? <Admin /> : <Navigate replace to="/login" />}>
```

```
          <Route index path="" element={ <Dashboard />} />
          <Route path="complaints" element={ <Complaints />} />
          <Route path="departments" element={ <Departments />} />
          <Route path="users" element={ <Users />} />
        </Route>

        {/* Main Routes */}
        <Route path="/" element={<NavBar isLoged={isLoged} />}>
          <Route path="complaint" element={isLoged ? <Complaint /> : <Navigate replace to="/login" />} />
          <Route path="complaintList" element={isLoged ? <ComplaintList /> : <Navigate replace to="/login" />} />
          <Route index element={<Home />} />
        </Route>
        {/* not Found Route */}
        <Route path='*' element={<NotFound />} />
      </Routes>
    </div>
  );
}

export default App;
```

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import { BrowserRouter as Router } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
```

```
root.render(
  <React.StrictMode>
    <Router >
      <App />
    </Router>
  </React.StrictMode>
);
```

Backend

```
const express = require('express')
const http = require('http')
const mongoose = require('mongoose')
const cors = require('cors')
const bcrypt = require('bcrypt')
const jwt = require('jsonwebtoken')
const { Server } = require('socket.io')
const { type } = require('os')
const multer = require('multer')
const path = require('path')
require('dotenv').config()

const app = express()
const { Schema, model } = mongoose
const jwtSecret = process.env.JWT_SECRET;

//middlewares
app.use(express.json())
app.use(cors())
app.use('/uploads', express.static('uploads'))
```

```
// database connection
mongoose.connect('mongodb+srv://abdumh:AmhkbwgA@cluster0.rfkxh.mongodb.net/cms', {
   useNewUrlParser: true,
   useUnifiedTopology:true
})
   .then(() => console.log('connected to mongodb'))
   .catch(err => console.log('error connecting to mongodb', err))

// Schemas

const userSchema = new Schema({
   fname: {type: String, required: true},
   lname: {type: String, required: true},
   email: {type: String, required: true, unique: true},
   password: {type: String, required: true},
   role:{type: String, default: 'user'},
   createdAt: {type: Date, default: Date.now}
})

userSchema.index({ email: 1 }, { unique: true }) // this is added so that the email becomes unique

//hashing password with bcrypt
userSchema.pre('save', async function(next) {
   const salt = await bcrypt.genSalt()
   this.password = await bcrypt.hash(this.password, salt)
   next()
})
// end of hashing
```

```
const complaintSchema = new Schema({
    department: {type: String, required: true},
    severity: {type: String, required: true},
    description: {type: String, required: true},
    reporter: {type: String},
    fileUrl: {type: String},
    status: {type: String, default: 'unread'},
    createdAt: {type: Date, default: Date.now},
    solution: {type: String, default: 'Not solved yet'}
})

const departmentSchema = new Schema({
    name: {type: String, required: true},
    head: {type: String, required: true},
    members: {type: Array, required: true},
    createdAt: {type: Date, default: Date.now}
})

const User = model('User', userSchema)
const Complaint = model('Complaint', complaintSchema)
const Department = model('Department', departmentSchema)

//configuring multer storage
const storage = multer.diskStorage({
    destination : (req, file, cb) => {
        cb(null, 'uploads/')
    },
    filename : (req, file, cb) => {
        cb(null, Date.now() + path.extname(file.originalname))
    }
})

const upload = multer({ storage })

// Routes

app.get('/userList', async (req, res) => {
    try {
        const users = await User.find()
        console.log(users)
        res.status(200).json(users)
    } catch (error) {
        res.status(400).json(error)
    }
})

app.get('/complaintList', async (req, res) => {
    try {
        const complaints = await Complaint.find()
        res.json(complaints)
    } catch (error) {
        res.status(500).json(error)
    }
})

app.get('/unreadedcomplaintList', async (req, res) => {
    try {
        const complaints = await Complaint.find({status: 'unread'})
        res.json(complaints)
    } catch (error) {
        res.status(500).json(error)
    }
})

app.get('/departmentList', async (req, res) => {
    try {
        const departments = await Department.find()
        res.json(departments)
    } catch (error) {
        res.status(500).json(error)
    }
})

app.get('/getDepHead/:name', async (req, res) => {
    const {name} = req.params
    try {
        const depHead = await Department.find({head: name})
        res.status(200).json(depHead)
    } catch (error) {
        res.status(404).json(error)
    }
})

app.post('/login', async (req, res) => {
    const {email, password} = req.body
    try {
        const user = await User.findOne({email})

        //check if user exists
        if( !user ) return res.status(400).json({message: 'The user does\'t exist'})

        //compare the password
        const isMatch = await bcrypt.compare(password, user.password)

        //check if password matches
        if( !isMatch ) return res.status(400).json({message: 'Invalid credentials'})

        // generate jwt
        const token = jwt.sign({id: user._id}, jwtSecret, {expiresIn: '1h'})
        res.json({token, email, fname: user.fname, role: user.role})
    } catch (error) {
        res.status(500).json(error)
    }
})

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

    try {
        const newUser = new User(req.body)
        const savedUser = await newUser.save()
        res.status(201).json(savedUser)
    } catch (error) {
        if (error.code === 11000) { // Duplicate key error
          res.status(400).json({ error: 'Email already exists' });
        } else {
```

```
     res.status(400).json(error);
    }
   }
})

app.post('/complaint', upload.single('file'), async (req, res) => {
  const { department, severity, description, reporter} = req.body
  const fileUrl = req.file ? req.file.path : null
  try {
    const newComplaint = new Complaint({
      department, severity, description, fileUrl, reporter
    })
    const savedComplaint = await newComplaint.save()
    res.status(201).json(savedComplaint)
  } catch (error) {
    res.status(400).json(error)
  }
})

app.post('/department', async (req, res)=> {
  try {
    const newDepartment = new Department(req.body)
    const savedDepartment = await newDepartment.save()
    res.status(201).json(savedDepartment)
  } catch (error) {
    res.status(400).json(error)
  }
})
```

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

  try {

    const deps = await Department.insertMany(req.body)

    res.status(201).json(deps)

  } catch (error) {

    res.status(400).json(error)

  }

})


app.delete('/deleteDepartment/:id', async (req, res) => {

  const {id} = req.params

  try {

    const deleteDep = await Department.deleteOne({ _id: id })

    res.status(200).json(deleteDep)

  } catch (error) {

    res.status(400).json(error)

  }

})


app.delete('/deleteUser/:id', async (req, res) => {

  const {id} = req.params

  try {

    const deleteUser = await User.deleteOne({ _id: id })

    res.status(200).json(deleteUser)

  } catch (error) {

    res.status(400).json(error)

  }

})
```

```
app.patch('/userList/:id', async (req, res) => {
  const { id } = req.params; // Retrieve id from route parameters
  const updateData = req.body; // Data to update

  try {
    const updateUser = await User.updateOne(
      { _id: id },          // Use `_id` if `id` is the MongoDB ObjectID
      { $set: updateData }    // Spread the update data correctly inside `$set`
    );

    if (updateUser.nModified === 0) {
      return res.status(404).json({ error: 'User not found or no changes made' });
    }

    res.status(200).json(updateUser);
  } catch (error) {
    res.status(400).json(error);
  }
});

app.patch('/updateDepartment/:id', async (req, res) => {
  const{ id } = req.params
  const updateData = req.body
  try {
    const updateDep = await Department.updateOne(
      {_id: id},
      {$set: updateData}
    )
    if(updateDep.nModified === 0){
            res.status(404).json({error: "Department not found or no changes made"})
        }
        res.status(200).json(updateDep)
    } catch (error) {
        res.status(404).json(error)
    }
})

app.patch('/updateUser/:id', async (req, res) => {
    const{ id } = req.params
    const updateData = req.body
    try {
        const updateUser = await User.updateOne(
            {_id: id},
            {$set: updateData}
        )
        if(updateUser.nModified === 0){
            res.status(404).json({error: "Department not found or no changes made"})
        }
        res.status(200).json(updateUser)
    } catch (error) {
        res.status(404).json(error)
    }
})

app.put('/solution/:id', async (req, res) => {
    const {id} = req.params
    const {solution} = req.body
    console.log(id, solution)
```

```
  try {
    const response = await Complaint.updateOne(
      {_id: id},
      {$set: {solution, status: 'read'}}
    )
    if(response.nModified === 0){
      return res.status(404).json({error: "complaint not found or no changes made"})
    }
    return res.status(200).json(response)
  } catch (error) {
    return res.status(404).json({ err: "not posting", error})
  }
})



// chat app
// const server = http.createServer(app)
// const io = new Server(server, {
//   cors: {
//     origin: "http://localhost:3001",
//     methods: ['POST']
//   }
// })

// io.on('connection', socket => {
//   console.log('a user connected')

//   socket.on('chat message', msg => {
```

```
//    socket.broadcast.emit('chat message', msg)

//   })

//   socket.on('disconnect', ()=>{

//    console.log('disconnected')

//   })


// })


app.listen(2005, () => console.log('server listening on port 2005...'))
```