Index.js

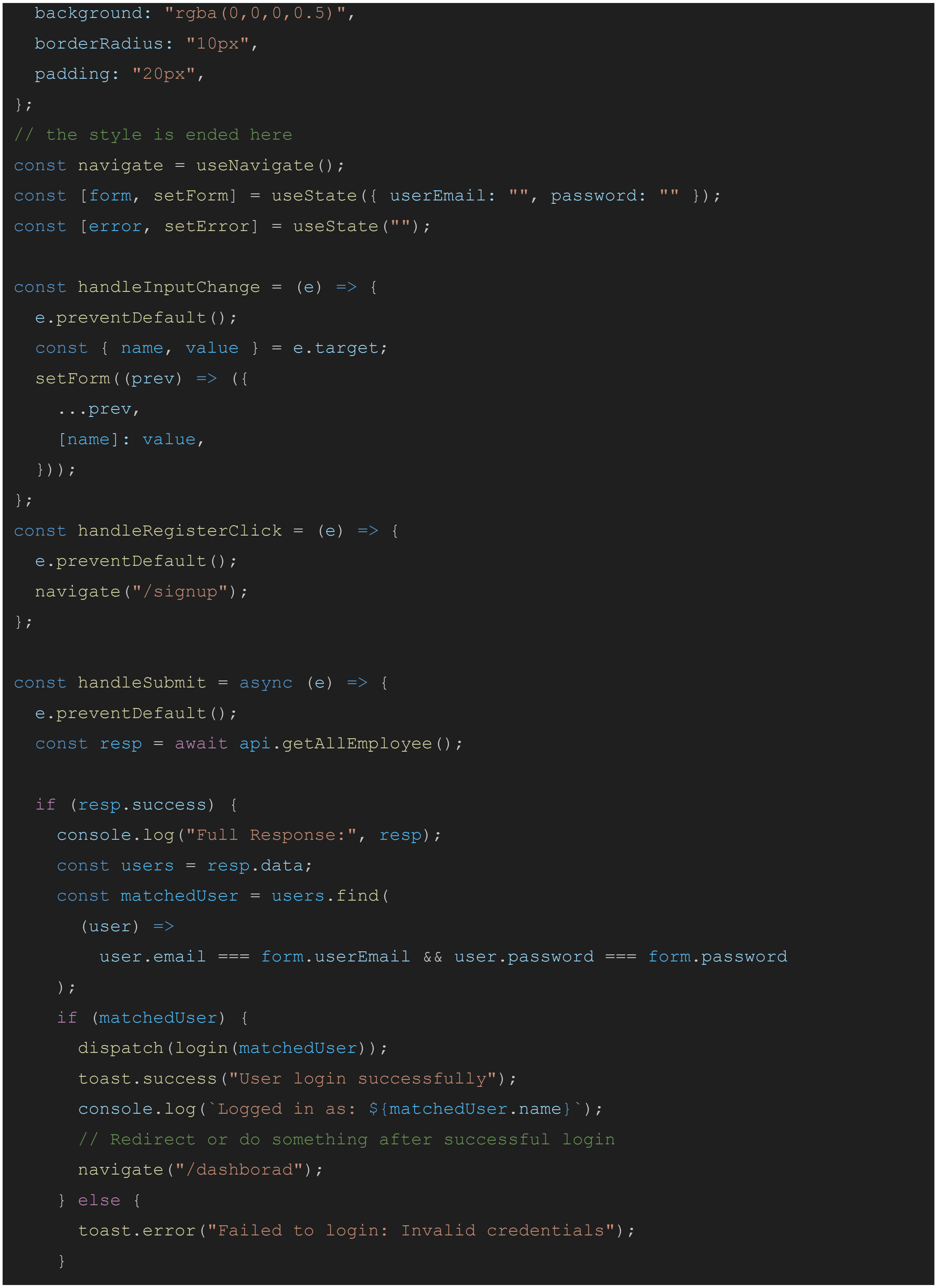
```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import "bootstrap/dist/css/bootstrap.min.css";
import "bootstrap/dist/js/bootstrap.bundle.min.js";
import { Provider } from "react-redux";
import { store } from "./services/store";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(
 <React.StrictMode>
   <Provider store={store}>
     <App />
   </Provider>
 </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

App.js

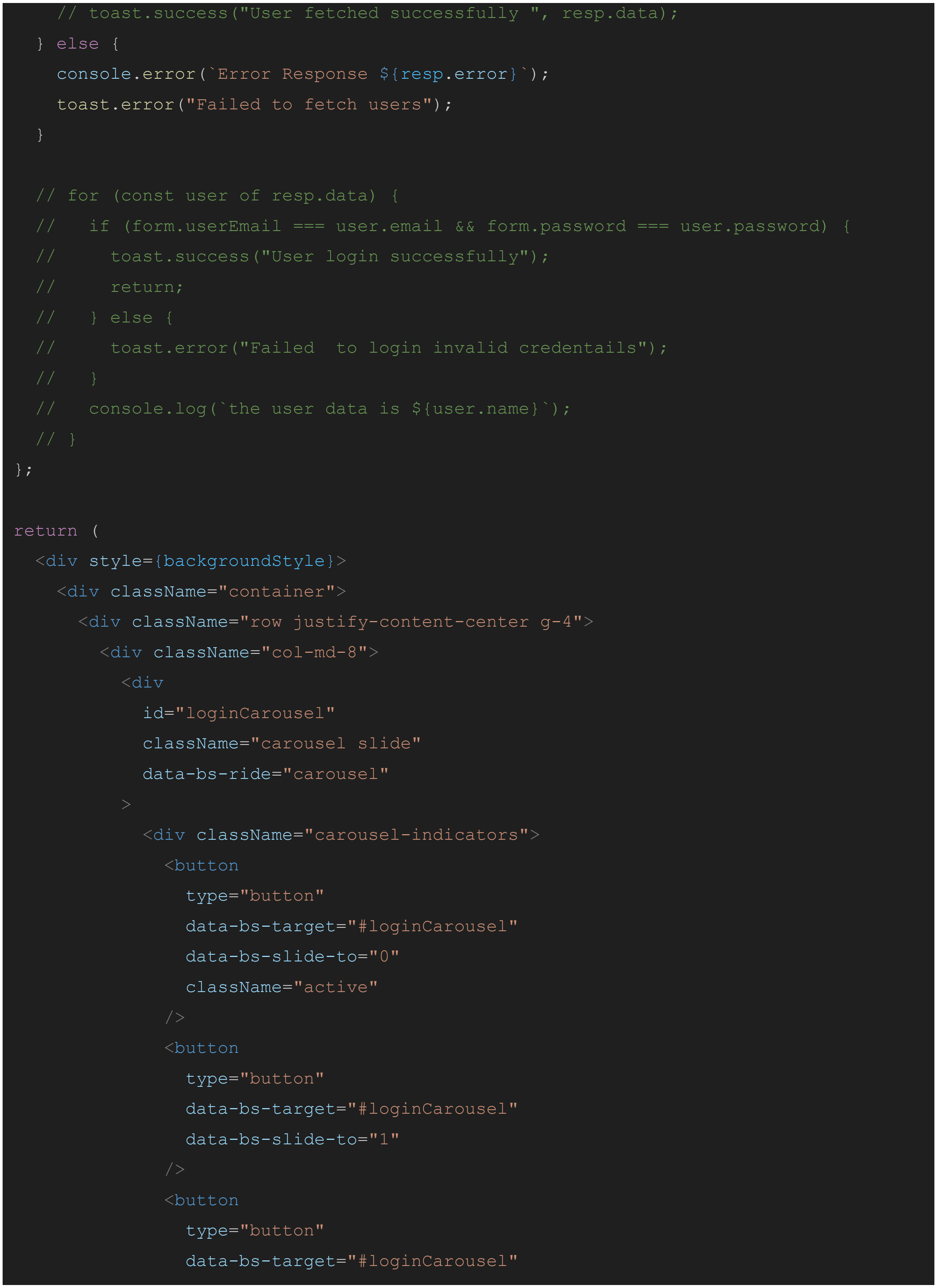
```
import "./App.css";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Login from "./component/Login";
import SignUp from "./component/SignUp";
import { ToastContainer } from "react-toastify";
import DashBoard from "./component/DashBoard";
import AllEmpolyee from "./component/AllEmpolyee";
import AddEmployee from "./component/AddEmployee";
import PunchInOut from "./component/PunchInOut";
import ViewAttendance from "./component/ViewAttendance";
import ViewLeave from "./component/ViewLeave";
import ManageLeave from "./component/ManageLeave";
function App() {
 return (
   <>
     <Router>
       <Routes>
         <Route path="/" element={<Login />} />
         <Route path="/signup" element={<SignUp />} />
         <Route path="/dashborad" element={<DashBoard />} />
         <Route path="/allemployee" element={<AllEmpolyee />} />
         <Route path="/addemployee" element={<AddEmployee />} />
         <Route path="/punchinout" element={<PunchInOut />} />
         <Route path="/viewattendance" element={<ViewAttendance />} />
         <Route path="/viewleave" element={<ViewLeave />} />
         <Route path="/manageleave" element={<ManageLeave />} />
       </Routes>
     </Router>
     <ToastContainer position="top-right" autoClose="3000"></ToastContainer>
   </>
 );
}

export default App;
```

Login.js

```
import React from "react";
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import { backgroundStyle } from "../customstyles";
import { api } from "../services/api";
import { toast } from "react-toastify";
import { useDispatch } from "react-redux";
import { login } from "../services/authslice";

const Login = () => {
 // the below are used to set the backgrund style
 const dispatch = useDispatch();
 const carouselImageSyle = {
   height: "400px",
   objectFit: "cover",
   borderRadius: "10px",
 };

 const carouselCaptionStyle = {
```



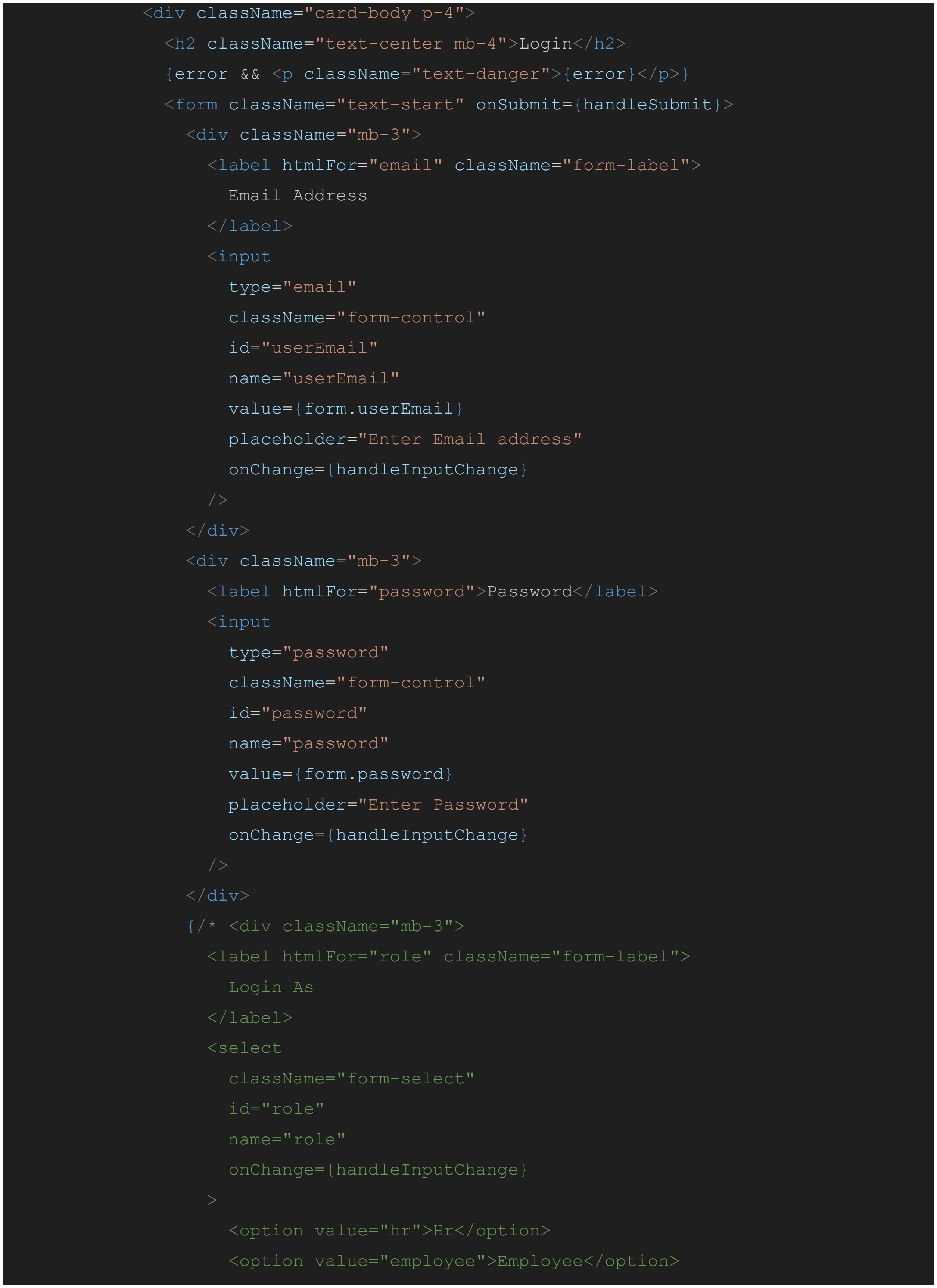
```
  background: "rgba(0,0,0,0.5)",
  borderRadius: "10px",
  padding: "20px",
};
// the style is ended here
const navigate = useNavigate();
const [form, setForm] = useState({ userEmail: "", password: "" });
const [error, setError] = useState("");

const handleInputChange = (e) => {
  e.preventDefault();
  const { name, value } = e.target;
  setForm((prev) => ({
    ...prev,
    [name]: value,
  }));
};
const handleRegisterClick = (e) => {
  e.preventDefault();
  navigate("/signup");
};

const handleSubmit = async (e) => {
  e.preventDefault();
  const resp = await api.getAllEmployee();

  if (resp.success) {
    console.log("Full Response:", resp);
    const users = resp.data;
    const matchedUser = users.find(
      (user) =>
        user.email === form.userEmail && user.password === form.password
    );
    if (matchedUser) {
      dispatch(login(matchedUser));
      toast.success("User login successfully");
      console.log(`Logged in as: ${matchedUser.name}`);
      // Redirect or do something after successful login
      navigate("/dashborad");
    } else {
      toast.error("Failed to login: Invalid credentials");
    }
```



```
    // toast.success("User fetched successfully ", resp.data);
  } else {
    console.error(`Error Response ${resp.error}`);
    toast.error("Failed to fetch users");
  }

  // for (const user of resp.data) {
  //   if (form.userEmail === user.email && form.password === user.password) {
  //     toast.success("User login successfully");
  //     return;
  //   } else {
  //     toast.error("Failed  to login invalid credentails");
  //   }
  //   console.log(`the user data is ${user.name}`);
  // }
};

return (
  <div style={backgroundStyle}>
    <div className="container">
      <div className="row justify-content-center g-4">
        <div className="col-md-8">
          <div
            id="loginCarousel"
            className="carousel slide"
            data-bs-ride="carousel"
          >
            <div className="carousel-indicators">
              <button
                type="button"
                data-bs-target="#loginCarousel"
                data-bs-slide-to="0"
                className="active"
              />
              <button
                type="button"
                data-bs-target="#loginCarousel"
                data-bs-slide-to="1"
              />
              <button
                type="button"
                data-bs-target="#loginCarousel"
```

```
                 data-bs-slide-to="2"
               />
             </div>
             <div className="carousel-inner">
               <div className="carousel-item active">
                 <img

src="https://images.unsplash.com/photo-1600880292203-757bb62b4baf?ixlib=rb-4.0.3&ixid=
M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"
                   className="d-block w-100"
                   alt="HRMS"
                   style={carouselImageSyle}
                 />
                 <div
                   className="carousel-caption"
                   style={carouselCaptionStyle}
                 >
                   <h3>Welcome To HRMS</h3>
                   <p>
                     streamline your hr operation with our comprenhensive
                     solution
                   </p>
                 </div>
               </div>
               <div className="carousel-item">
                 <img

src="https://images.unsplash.com/photo-1542744173-8e7e53415bb0?ixlib=rb-4.0.3&ixid=M3w
xMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"
                   className="d-block w-100"
                   alt="Team Management"
                   style={carouselImageSyle}
                 />
                 <div className="carousel-caption">
                   <h3>Employee Management</h3>
                   <p>Efficiently manage your workforce</p>
                 </div>
               </div>
               <div className="carousel-item">
                 <img
```

```
src="https://images.unsplash.com/photo-1552581234-26160f608093?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"
                   className="d-block w-100"
                   alt="Performance Analysis"
                   style={carouselImageSyle}
                 />
                 <div
                   className="carousel-caption"
                   style={{
                     background: "rgba(0,0,0,0.5)",
                     borderRadius: "10px",
                     padding: "20px",
                   }}
                 >
                   <h3>Performance Tracking</h3>
                   <p>Monitor and improve employee performance</p>
                 </div>
               </div>
             </div>
             <button
               className="carousel-control-prev"
               type="button"
               data-bs-target="#loginCarousel"
               data-bs-slide="prev"
             >
               <span className="carousel-control-prev-icon"></span>
               <span className="visually-hidden">Previous</span>
             </button>
             <button
               className="carousel-control-next"
               type="button"
               data-bs-target="#loginCarousel"
               data-bs-slide="next"
             >
               <span className="carousel-control-next-icon"></span>
               <span className="visually-hidden">Next</span>
             </button>
           </div>
         </div>
         <div className="col-md-4">
           <div className="card show h-100">
```

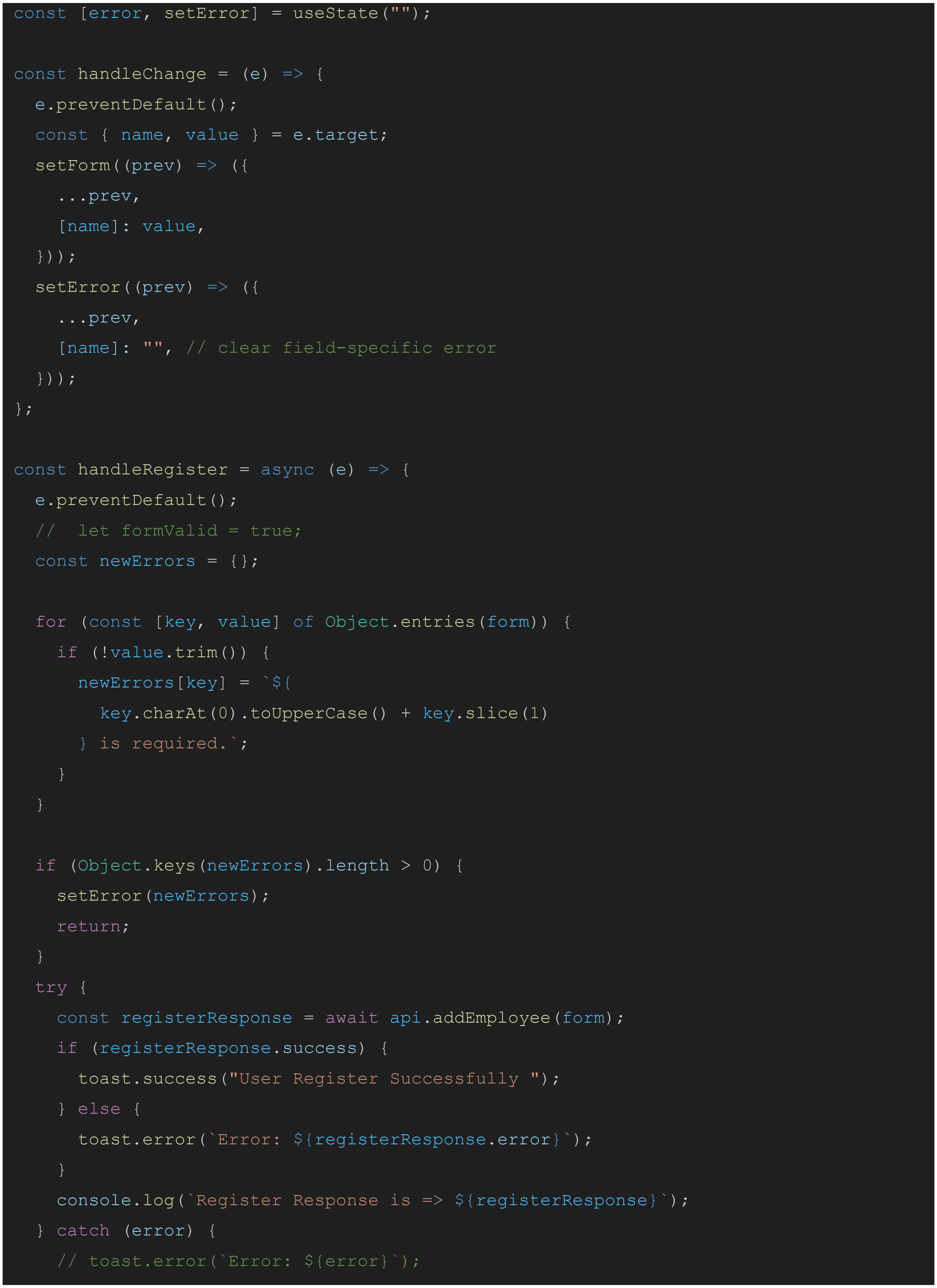


```
            <div className="card-body p-4">
              <h2 className="text-center mb-4">Login</h2>
              {error && <p className="text-danger">{error}</p>}
              <form className="text-start" onSubmit={handleSubmit}>
                <div className="mb-3">
                  <label htmlFor="email" className="form-label">
                    Email Address
                  </label>
                  <input
                    type="email"
                    className="form-control"
                    id="userEmail"
                    name="userEmail"
                    value={form.userEmail}
                    placeholder="Enter Email address"
                    onChange={handleInputChange}
                  />
                </div>
                <div className="mb-3">
                  <label htmlFor="password">Password</label>
                  <input
                    type="password"
                    className="form-control"
                    id="password"
                    name="password"
                    value={form.password}
                    placeholder="Enter Password"
                    onChange={handleInputChange}
                  />
                </div>
                {/* <div className="mb-3">
                  <label htmlFor="role" className="form-label">
                    Login As
                  </label>
                  <select
                    className="form-select"
                    id="role"
                    name="role"
                    onChange={handleInputChange}
                  >
                    <option value="hr">Hr</option>
                    <option value="employee">Employee</option>
```

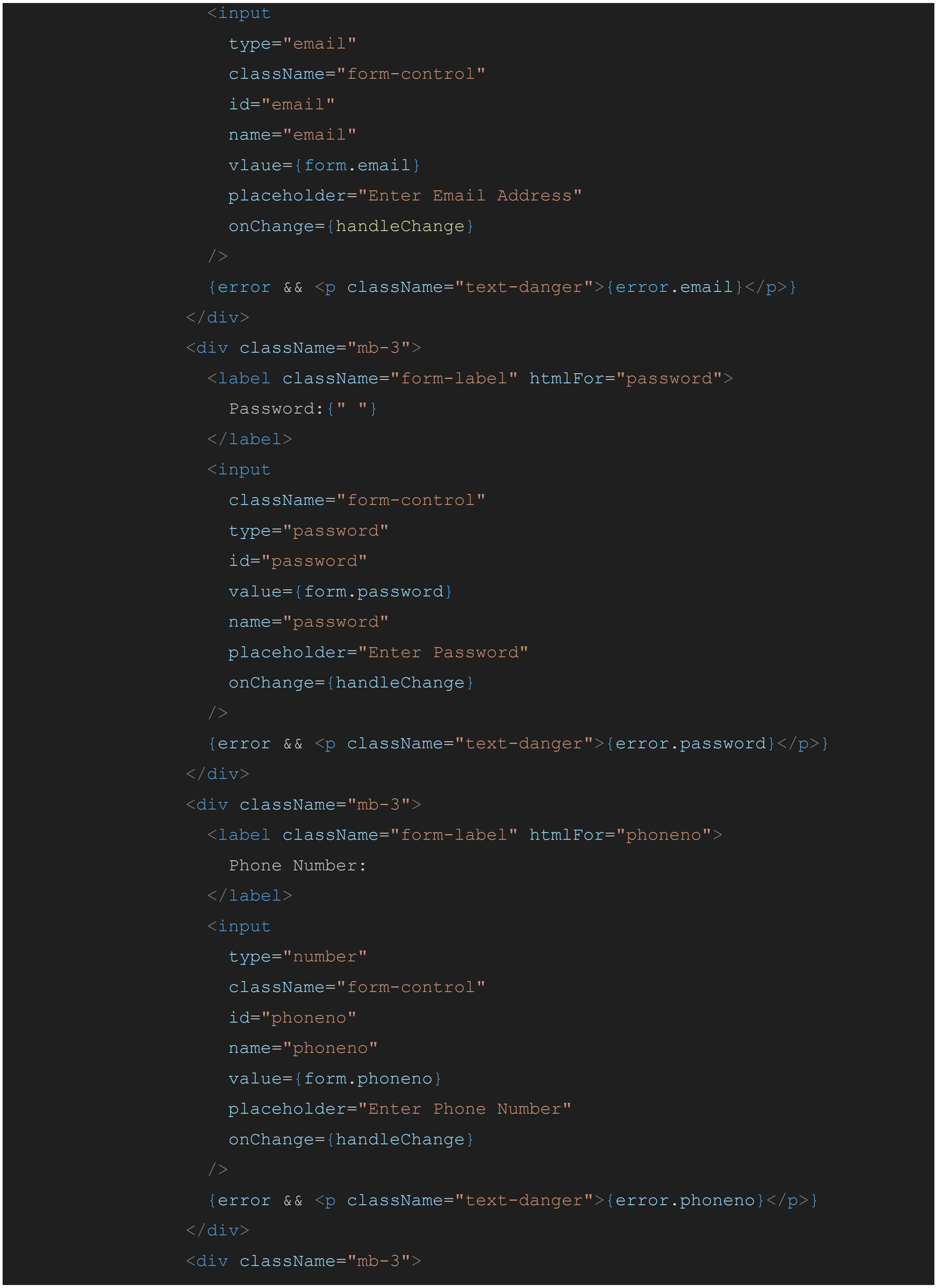
```
                     <option vlaue="admin">Admin</option>
                   </select>
                 </div> */}

                 <label
                   className="text-primary mb-4"
                   onClick={handleRegisterClick}
                 >
                   Yet not register click here to register!!
                 </label>
                 <button type="submit" className="btn btn-primary w-100">
                   Login
                 </button>
               </form>
             </div>
           </div>
         </div>
       </div>
     </div>
   </div>
 );
};

export default Login;
```

Register.js (Signup.js)

```
import React, { useState } from "react";
import { api } from "../services/api";
import { toast } from "react-toastify";
import { backgroundStyle } from "../customstyles";

function SignUp() {
 const [form, setForm] = useState({
   name: "",
   email: "",
   password: "",
   role: "",
   phoneno: "",
   department: "",
   address: "",
 });
```



```
const [error, setError] = useState("");

const handleChange = (e) => {
  e.preventDefault();
  const { name, value } = e.target;
  setForm((prev) => ({
    ...prev,
    [name]: value,
  }));
  setError((prev) => ({
    ...prev,
    [name]: "", // clear field-specific error
  }));
};

const handleRegister = async (e) => {
  e.preventDefault();
  //  let formValid = true;
  const newErrors = {};

  for (const [key, value] of Object.entries(form)) {
    if (!value.trim()) {
      newErrors[key] = `${
        key.charAt(0).toUpperCase() + key.slice(1)
      } is required.`;
    }
  }

  if (Object.keys(newErrors).length > 0) {
    setError(newErrors);
    return;
  }
  try {
    const registerResponse = await api.addEmployee(form);
    if (registerResponse.success) {
      toast.success("User Register Successfully ");
    } else {
      toast.error(`Error: ${registerResponse.error}`);
    }
    console.log(`Register Response is => ${registerResponse}`);
  } catch (error) {
    // toast.error(`Error: ${error}`);
    console.error(`error ading employee ${error}`);
  }
  setForm({
    name: "",
    email: "",
    password: "",
    role: "",
    phoneno: "",
    department: "",
    address: "",
  });
  setError("");
};
return (
  <div style={backgroundStyle}>
    <div className="container mt-5">
      <div className="row justify-content-center">
        <div className="col-md-6">
          <div className="card show">
            <div className="card-body">
              <h2 className="text-center mb-4">Register</h2>

              <form className="text-start" onSubmit={handleRegister}>
                <div className="mb-3">
                  <label className="form-label" htmlFor="name">
                    Name:{" "}
                  </label>
                  <input
                    type="text"
                    className="form-control"
                    id="name"
                    name="name"
                    value={form.name}
                    placeholder="Enter Name"
                    onChange={handleChange}
                  />
                  {error && <p className="text-danger">{error.name}</p>}
                </div>
                <div className="mb-3">
                  <label className="form-label" htmlFor="email">
                    Email Address:
                  </label>
```



```
                  <input
                    type="email"
                    className="form-control"
                    id="email"
                    name="email"
                    vlaue={form.email}
                    placeholder="Enter Email Address"
                    onChange={handleChange}
                  />
                  {error && <p className="text-danger">{error.email}</p>}
                </div>
                <div className="mb-3">
                  <label className="form-label" htmlFor="password">
                    Password:{" "}
                  </label>
                  <input
                    className="form-control"
                    type="password"
                    id="password"
                    value={form.password}
                    name="password"
                    placeholder="Enter Password"
                    onChange={handleChange}
                  />
                  {error && <p className="text-danger">{error.password}</p>}
                </div>
                <div className="mb-3">
                  <label className="form-label" htmlFor="phoneno">
                    Phone Number:
                  </label>
                  <input
                    type="number"
                    className="form-control"
                    id="phoneno"
                    name="phoneno"
                    value={form.phoneno}
                    placeholder="Enter Phone Number"
                    onChange={handleChange}
                  />
                  {error && <p className="text-danger">{error.phoneno}</p>}
                </div>
                <div className="mb-3">
```

```
              <label className="form-lable" htmlFor="department">
                Department:
              </label>
              <input
                type="text"
                className="form-control"
                id="department"
                name="department"
                value={form.department}
                placeholder="Enter Department"
                onChange={handleChange}
              />
              {error && <p className="text-danger">{error.department}</p>}
            </div>
            <div className="mb-3">
              <label className="form-label" htmlFor="address">
                Address:
              </label>
              <input
                type="text"
                className="form-control"
                id="address"
                name="address"
                value={form.address}
                placeholder="Enter Address"
                onChange={handleChange}
              />
              {error && <p className="text-danger">{error.address}</p>}
            </div>
            <div className="mb-3">
              <label htmlFor="role" className="form-label">
                Login As
              </label>
              <select
                className="form-select"
                id="role"
                name="role"
                onChange={handleChange}
              >
                <option value="hr">Hr</option>
                <option value="employee">Employee</option>
                <option vlaue="admin">Admin</option>
```

```
                   </select>
                   {error && <p className="text-danger">{error.role}</p>}
                 </div>
                 <button type="submit" className="btn btn-primary w-100">
                   Register{" "}
                 </button>
               </form>
             </div>
           </div>
         </div>
       </div>
     </div>
   </div>
 );
}
export default SignUp;
```

AdminHome.js

```
import React from "react";
import Header from "./Header";
import DashBoardCard from "./DashBoardCard";
import { useNavigate } from "react-router-dom";

function AdminHome() {
 const navigate = useNavigate();
 const hanldeGetAllEmployee = () => {
   console.log("handle clicked");
   navigate("/allemployee");
 };
 const handleAddEmployee = () => {
   navigate("/addemployee");
 };
 const handlePunchInOut = () => {
   navigate("/punchinout");
 };
 return (
   <>
     <Header />

     <div className="container mt-5">
```

```
       <div className="row">
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="Employee"
             onClick={hanldeGetAllEmployee}
             btnText="Get All employees"
           />
         </div>
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="Add New Employee"
             onClick={handleAddEmployee}
             btnText="Add Employee"
           />
         </div>
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="Punch In/Out And Apply Leave"
             onClick={handlePunchInOut}
           />
         </div>
       </div>
     </div>
   </>
 );
}

export default AdminHome;
```

DashBorad.js

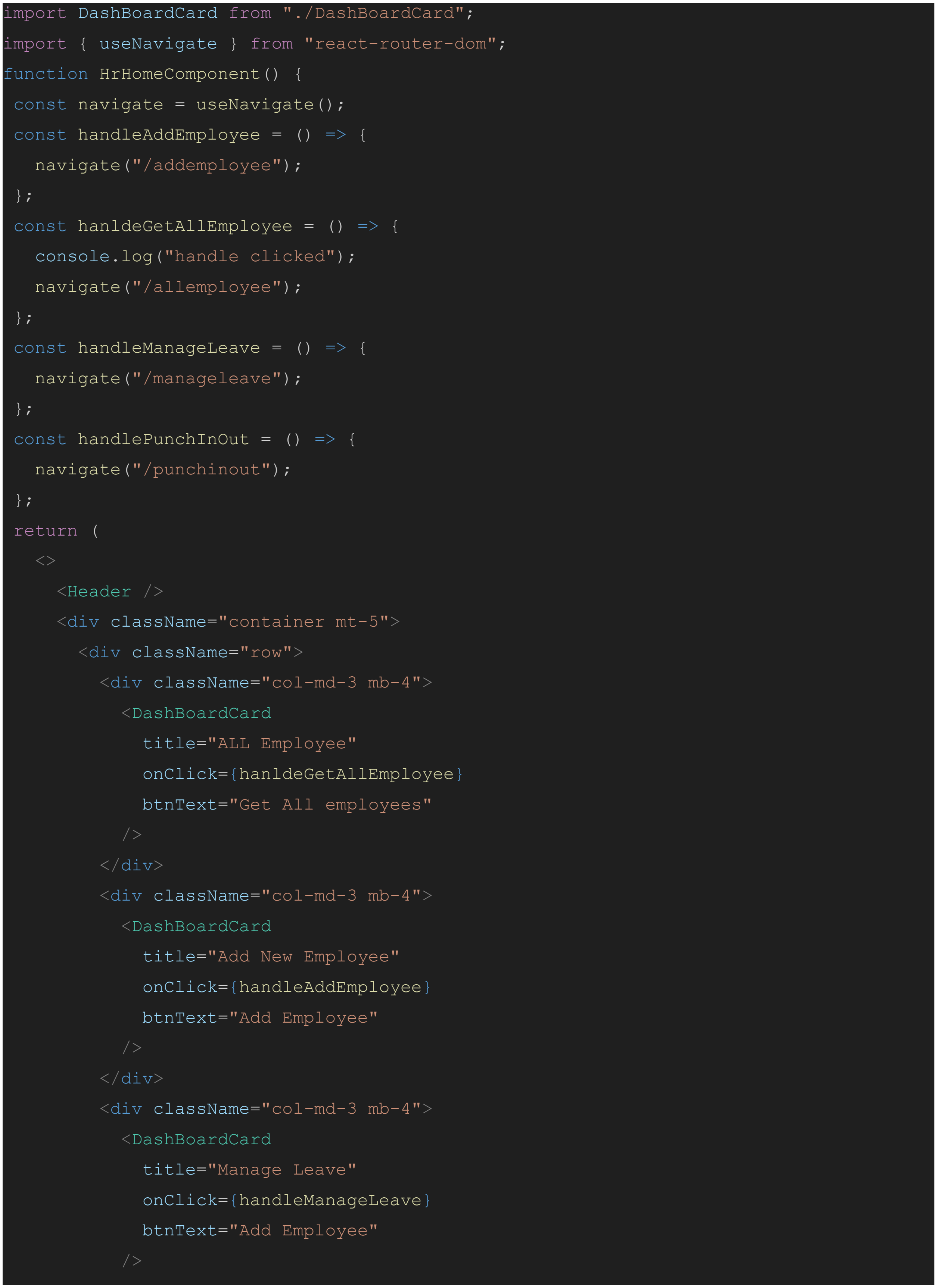
```
import { useSelector } from "react-redux";
import AdminHome from "./AdminHome";
import HrHomeComponent from "./HrHomeComponent";
import EmployeeHome from "./EmployeeHome";

function DashBoard() {
 const user = useSelector((state) => state.auth.user);
 if (!user) return <div>Please log in</div>;
 const role = user.role?.toLowerCase();

 switch (role) {
   case "admin":
     return <AdminHome />;
   case "hr":
     return <HrHomeComponent />;
   case "employee":
     return <EmployeeHome />;

   default:
     return <div>Role not recognized</div>;
 }
}

export default DashBoard;
```

EmployeeHome.js

```
import Header from "./Header";
import DashBoardCard from "./DashBoardCard";
import { useNavigate } from "react-router-dom";
function EmployeeHome() {
  const navigate = useNavigate();
  const handleApplyeeLeave = () => {};
  const handlePunchInOut = () => {
    navigate("/punchinout");
  };
  return (
    <>
      <Header />
      <div className="row mt-5">
        <DashBoardCard
          title="Punch In/Out And Apply Leave"
          onClick={handlePunchInOut}
        />
      </div>
    </>
  );
}
export default EmployeeHome;
```

HrHomeComponent.js

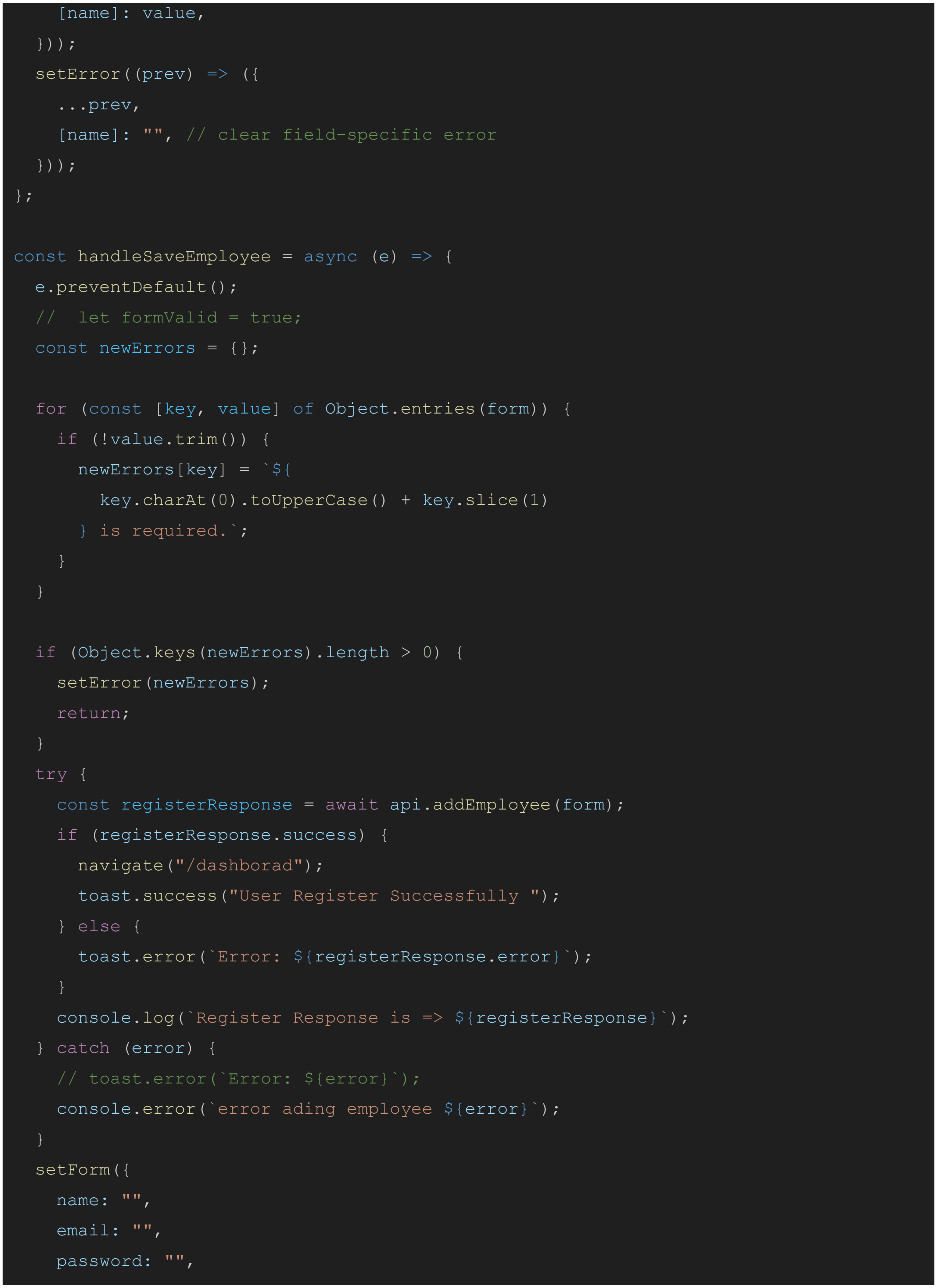
```
import Header from "./Header";
```



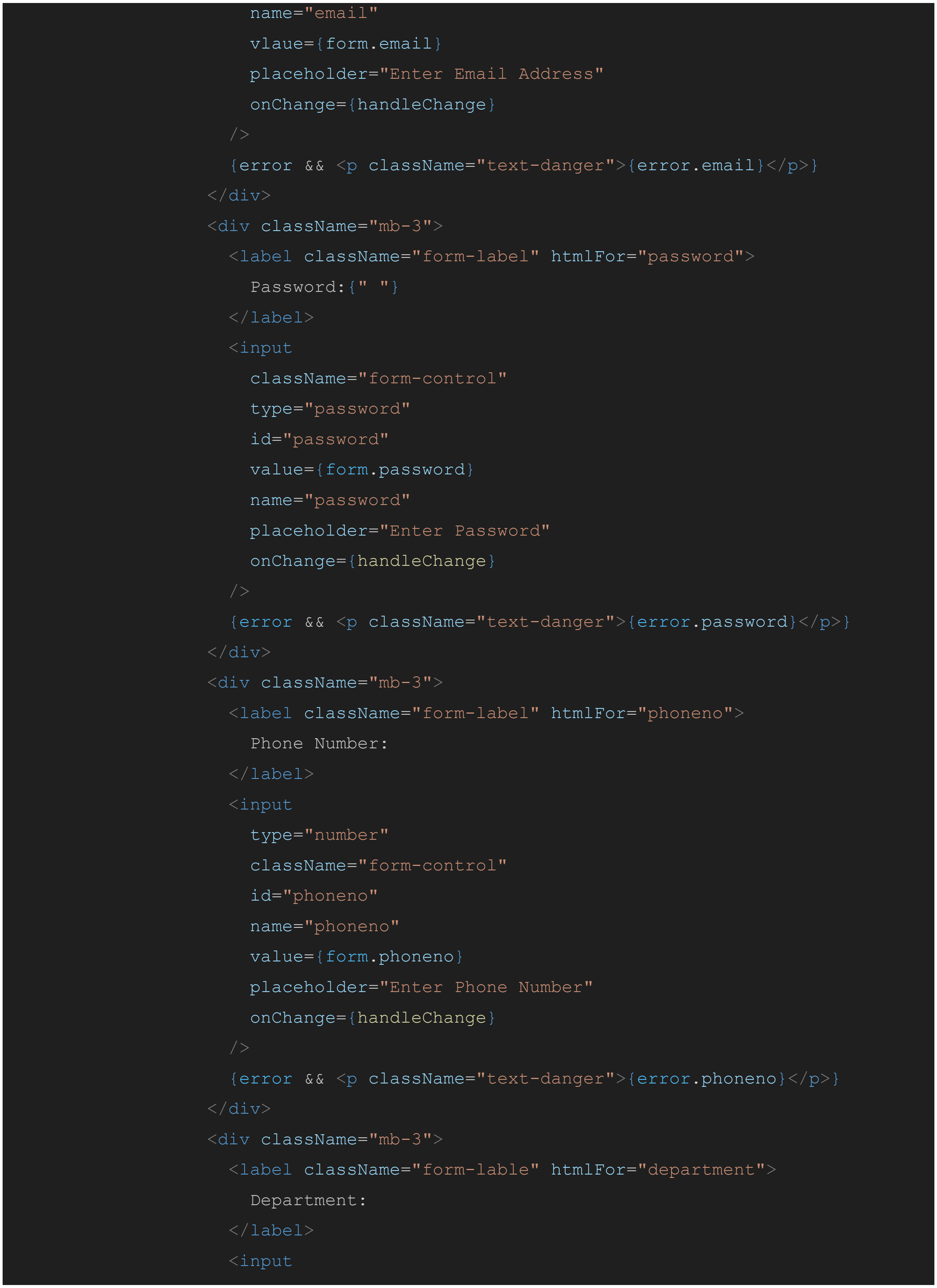
```
import DashBoardCard from "./DashBoardCard";
import { useNavigate } from "react-router-dom";
function HrHomeComponent() {
 const navigate = useNavigate();
 const handleAddEmployee = () => {
   navigate("/addemployee");
 };
 const hanldeGetAllEmployee = () => {
   console.log("handle clicked");
   navigate("/allemployee");
 };
 const handleManageLeave = () => {
   navigate("/manageleave");
 };
 const handlePunchInOut = () => {
   navigate("/punchinout");
 };
 return (
   <>
     <Header />
     <div className="container mt-5">
       <div className="row">
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="ALL Employee"
             onClick={hanldeGetAllEmployee}
             btnText="Get All employees"
           />
         </div>
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="Add New Employee"
             onClick={handleAddEmployee}
             btnText="Add Employee"
           />
         </div>
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="Manage Leave"
             onClick={handleManageLeave}
             btnText="Add Employee"
           />
         </div>
         <div className="col-md-3 mb-4">
           <DashBoardCard
             title="Punch In/Out And Apply Leave"
             onClick={handlePunchInOut}
           />
         </div>
       </div>
     </div>
   </>
 );
}

export default HrHomeComponent;
```

AddEmployee.js

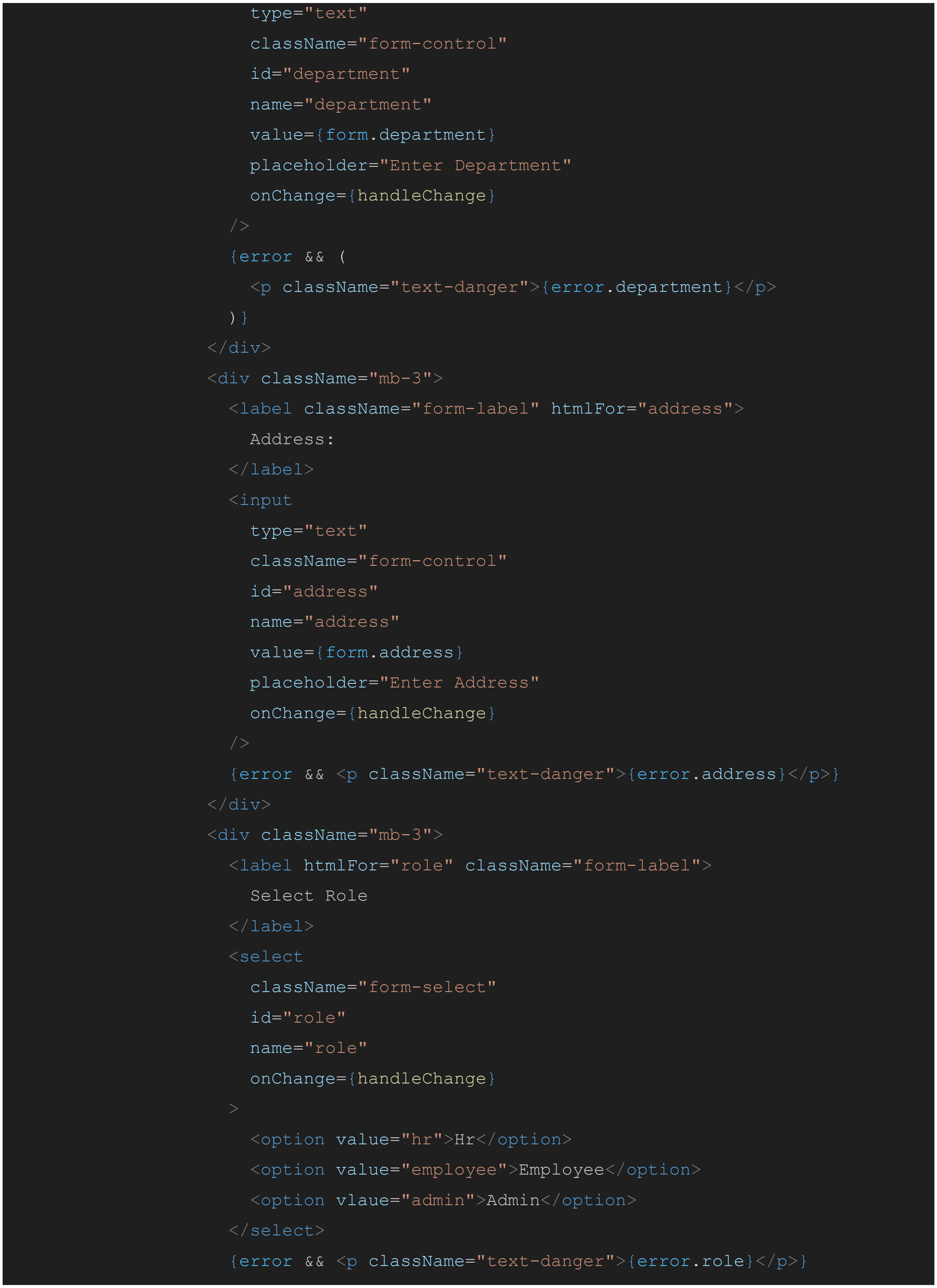
```
import React, { useState } from "react";
import { api } from "../services/api";
import { toast } from "react-toastify";
import { backgroundStyle } from "../customstyles";
import { useNavigate } from "react-router-dom";
import Header from "./Header";

function AddEmployee() {
 const navigate = useNavigate();
 const [form, setForm] = useState({
   name: "",
   email: "",
   password: "",
   role: "",
   phoneno: "",
   department: "",
   address: "",
 });
 const [error, setError] = useState("");

 const handleChange = (e) => {
   e.preventDefault();
   const { name, value } = e.target;
   setForm((prev) => ({
     ...prev,
```



```
    [name]: value,
  }));
  setError((prev) => ({
    ...prev,
    [name]: "", // clear field-specific error
  }));
};

const handleSaveEmployee = async (e) => {
  e.preventDefault();
  //  let formValid = true;
  const newErrors = {};

  for (const [key, value] of Object.entries(form)) {
    if (!value.trim()) {
      newErrors[key] = `${
        key.charAt(0).toUpperCase() + key.slice(1)
      } is required.`;
    }
  }

  if (Object.keys(newErrors).length > 0) {
    setError(newErrors);
    return;
  }
  try {
    const registerResponse = await api.addEmployee(form);
    if (registerResponse.success) {
      navigate("/dashborad");
      toast.success("User Register Successfully ");
    } else {
      toast.error(`Error: ${registerResponse.error}`);
    }
    console.log(`Register Response is => ${registerResponse}`);
  } catch (error) {
    // toast.error(`Error: ${error}`);
    console.error(`error ading employee ${error}`);
  }
  setForm({
    name: "",
    email: "",
    password: "",
    role: "",
    phoneno: "",
    department: "",
    address: "",
  });
  setError("");
};
return (
  <>
    <Header />
    <div>
      <div className="container mt-5">
        <div className="row justify-content-center">
          <div className="col-md-6">
            <div className="card show">
              <div className="card-body">
                <h2 className="text-center mb-4">Add New Employee</h2>

                <form className="text-start" onSubmit={handleSaveEmployee}>
                  <div className="mb-3">
                    <label className="form-label" htmlFor="name">
                      Name:{" "}
                    </label>
                    <input
                      type="text"
                      className="form-control"
                      id="name"
                      name="name"
                      value={form.name}
                      placeholder="Enter Name"
                      onChange={handleChange}
                    />
                    {error && <p className="text-danger">{error.name}</p>}
                  </div>
                  <div className="mb-3">
                    <label className="form-label" htmlFor="email">
                      Email Address:
                    </label>
                    <input
                      type="email"
                      className="form-control"
                      id="email"
```



```
    name="email"
    vlaue={form.email}
    placeholder="Enter Email Address"
    onChange={handleChange}
  />
  {error && <p className="text-danger">{error.email}</p>}
</div>
<div className="mb-3">
  <label className="form-label" htmlFor="password">
    Password:{" "}
  </label>
  <input
    className="form-control"
    type="password"
    id="password"
    value={form.password}
    name="password"
    placeholder="Enter Password"
    onChange={handleChange}
  />
  {error && <p className="text-danger">{error.password}</p>}
</div>
<div className="mb-3">
  <label className="form-label" htmlFor="phoneno">
    Phone Number:
  </label>
  <input
    type="number"
    className="form-control"
    id="phoneno"
    name="phoneno"
    value={form.phoneno}
    placeholder="Enter Phone Number"
    onChange={handleChange}
  />
  {error && <p className="text-danger">{error.phoneno}</p>}
</div>
<div className="mb-3">
  <label className="form-lable" htmlFor="department">
    Department:
  </label>
  <input
```



```
            type="text"
            className="form-control"
            id="department"
            name="department"
            value={form.department}
            placeholder="Enter Department"
            onChange={handleChange}
          />
          {error && (
            <p className="text-danger">{error.department}</p>
          )}
        </div>
        <div className="mb-3">
          <label className="form-label" htmlFor="address">
            Address:
          </label>
          <input
            type="text"
            className="form-control"
            id="address"
            name="address"
            value={form.address}
            placeholder="Enter Address"
            onChange={handleChange}
          />
          {error && <p className="text-danger">{error.address}</p>}
        </div>
        <div className="mb-3">
          <label htmlFor="role" className="form-label">
            Select Role
          </label>
          <select
            className="form-select"
            id="role"
            name="role"
            onChange={handleChange}
          >
            <option value="hr">Hr</option>
            <option value="employee">Employee</option>
            <option vlaue="admin">Admin</option>
          </select>
          {error && <p className="text-danger">{error.role}</p>}
```

```
                   </div>
                   <button type="submit" className="btn btn-primary w-100">
                     Save
                   </button>
                 </form>
               </div>
             </div>
           </div>
         </div>
       </div>
     </div>
   </>
 );
}
export default AddEmployee;
```

AllEmployee.js

```
import React, { useEffect, useState } from "react";
import { api } from "../services/api";
import { listStyle } from "../customstyles";
import Header from "./Header";
const AllEmpolyee = () => {
 const [users, setAllUsers] = useState([]);

 useEffect(() => {
   async function fetchEmployee() {
     const resp = await api.getAllEmployee();
     console.log(`the user resp form All Employee ${resp.data}`);
     if (resp.success) {
       setAllUsers(resp.data);
     }
   }
   fetchEmployee();
 }, []);

 return (
   <>
     <Header />
     <h3>All Employee</h3>
     <ul className="list-group list-group-flush" style={listStyle}>
       {users.map((user) => (
         <li key={user.id} className="list-group-item">
           {user.name}
         </li>
       ))}
     </ul>
   </>
 );
};

export default AllEmpolyee;
```

DashBoradCard.js

```
import React from "react";
import { cardstyle } from "../styles/card";
const DashBoardCard = ({ title = "Hrms", onClick, btnText = "" }) => {
 return (
   <div className="card" style={cardstyle} onClick={onClick}>
     <div className="card-body">
       <h5 className="card-title">{title}</h5>
     </div>
   </div>
 );
};

export default DashBoardCard;
```

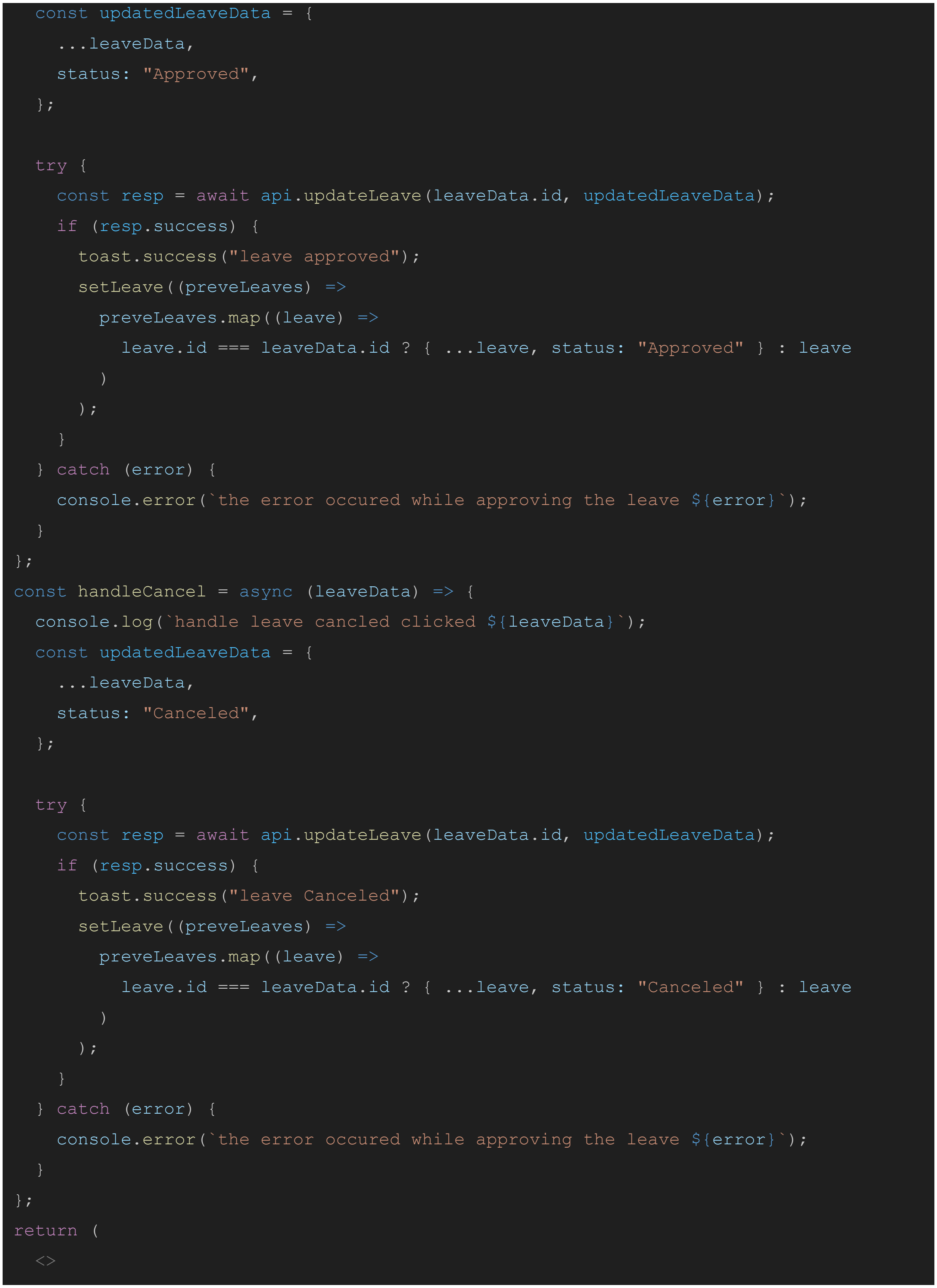
Header.js

```
import React, { useState, useEffect, useRef } from "react";
import { useSelector, useDispatch } from "react-redux";
import { logout } from "../services/authslice";
import { useNavigate } from "react-router-dom";
import "../styles/Header.css"; // keep your custom styles if needed

function Header() {
 const user = useSelector((state) => state.auth.user);
 const dispatch = useDispatch();
 const navigate = useNavigate();
 const [dropdownOpen, setDropdownOpen] = useState(false);
 const dropdownRef = useRef();

 const initials = user?.name
   ?.split(" ")
   .map((n) => n[0])
   .join("")
   .toUpperCase();

 const handleLogout = () => {
   dispatch(logout());
   navigate("/");
 };

 useEffect(() => {
   const handleClickOutside = (event) => {
     if (dropdownRef.current && !dropdownRef.current.contains(event.target)) {
       setDropdownOpen(false);
     }
   };
   document.addEventListener("mousedown", handleClickOutside);
   return () => document.removeEventListener("mousedown", handleClickOutside);
 }, []);

 if (!user) return null;

 return (
   <header className="d-flex justify-content-between align-items-center p-3 bg-light border-bottom">
     <div className="h5 m-0">
       Welcome, {user.name} — Role: {user.role}
     </div>
     <div className="dropdown" ref={dropdownRef}>
       <button
         className="btn btn-secondary dropdown-toggle"
         type="button"
         onClick={() => setDropdownOpen(!dropdownOpen)}
       >
         {initials}
       </button>
       <ul
         className={`dropdown-menu dropdown-menu-end ${
           dropdownOpen ? "show" : ""
```

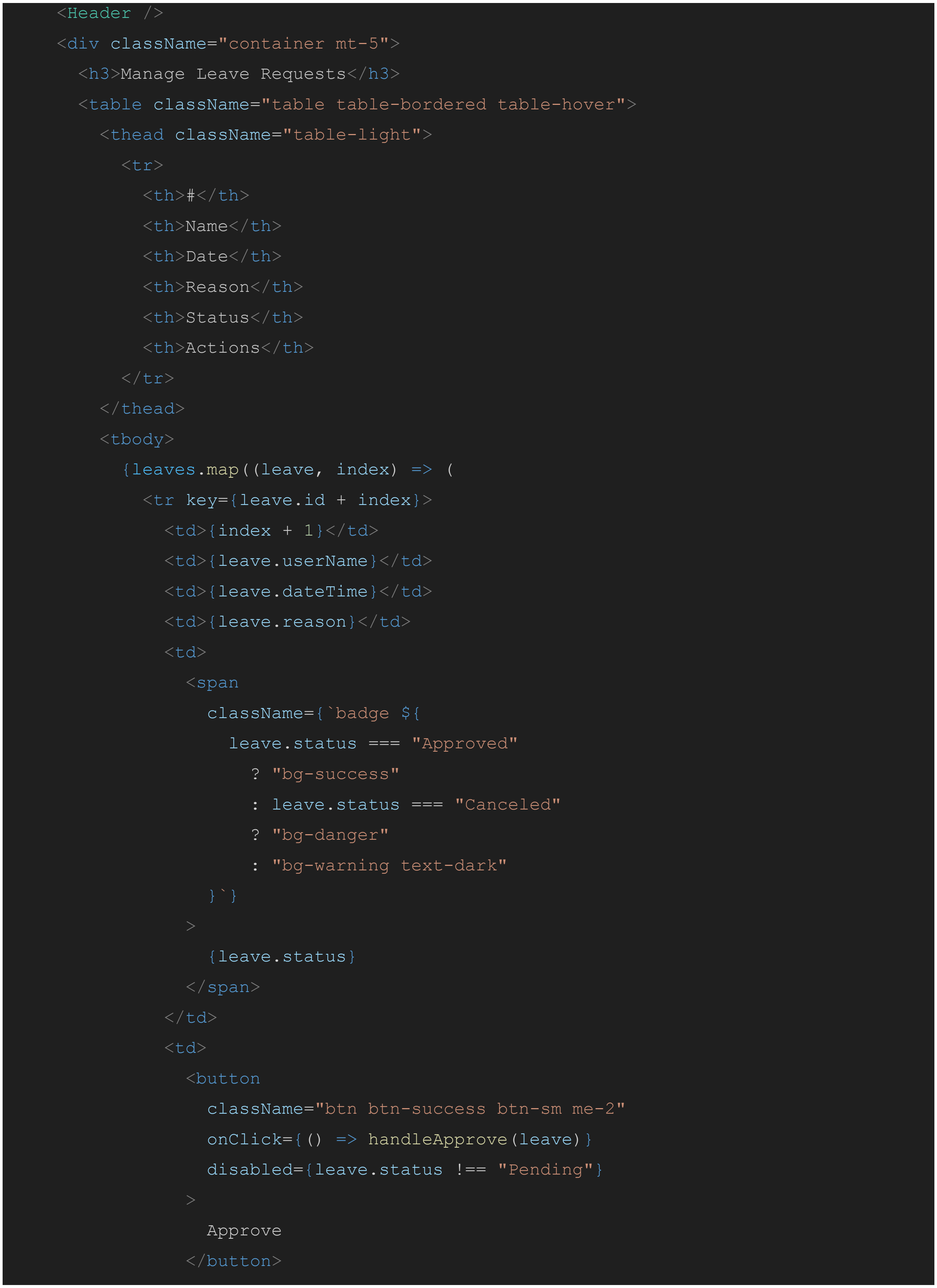
```
         }`}
       >
         <li>
           <button className="dropdown-item" onClick={handleLogout}>
             Logout
           </button>
         </li>
       </ul>
     </div>
   </header>
 );
}

export default Header;
```

ManageLeave.js

```
import React, { useState, useEffect } from "react";
import { api } from "../services/api";
import { toast } from "react-toastify";
import Header from "./Header";
const ManageLeave = () => {
 const [leaves, setLeave] = useState([]);

 useEffect(() => {
   async function fetchLeave() {
     const resp = await api.getAllLeaves();
     console.log(`Leave response : ${JSON.stringify(resp)}`);

     if (resp.success) {
       setLeave(resp.data);
     }
   }
   fetchLeave();
 }, []);

 const handleApprove = async (leaveData) => {
   console.log(
     `handle leave approve clicked and leave data is ${JSON.stringify(
       leaveData
     )}`
   );
```



```
  const updatedLeaveData = {
    ...leaveData,
    status: "Approved",
  };

  try {
    const resp = await api.updateLeave(leaveData.id, updatedLeaveData);
    if (resp.success) {
      toast.success("leave approved");
      setLeave((preveLeaves) =>
        preveLeaves.map((leave) =>
          leave.id === leaveData.id ? { ...leave, status: "Approved" } : leave
        )
      );
    }
  } catch (error) {
    console.error(`the error occured while approving the leave ${error}`);
  }
};
const handleCancel = async (leaveData) => {
  console.log(`handle leave cancled clicked ${leaveData}`);
  const updatedLeaveData = {
    ...leaveData,
    status: "Canceled",
  };

  try {
    const resp = await api.updateLeave(leaveData.id, updatedLeaveData);
    if (resp.success) {
      toast.success("leave Canceled");
      setLeave((preveLeaves) =>
        preveLeaves.map((leave) =>
          leave.id === leaveData.id ? { ...leave, status: "Canceled" } : leave
        )
      );
    }
  } catch (error) {
    console.error(`the error occured while approving the leave ${error}`);
  }
};
return (
  <>
```

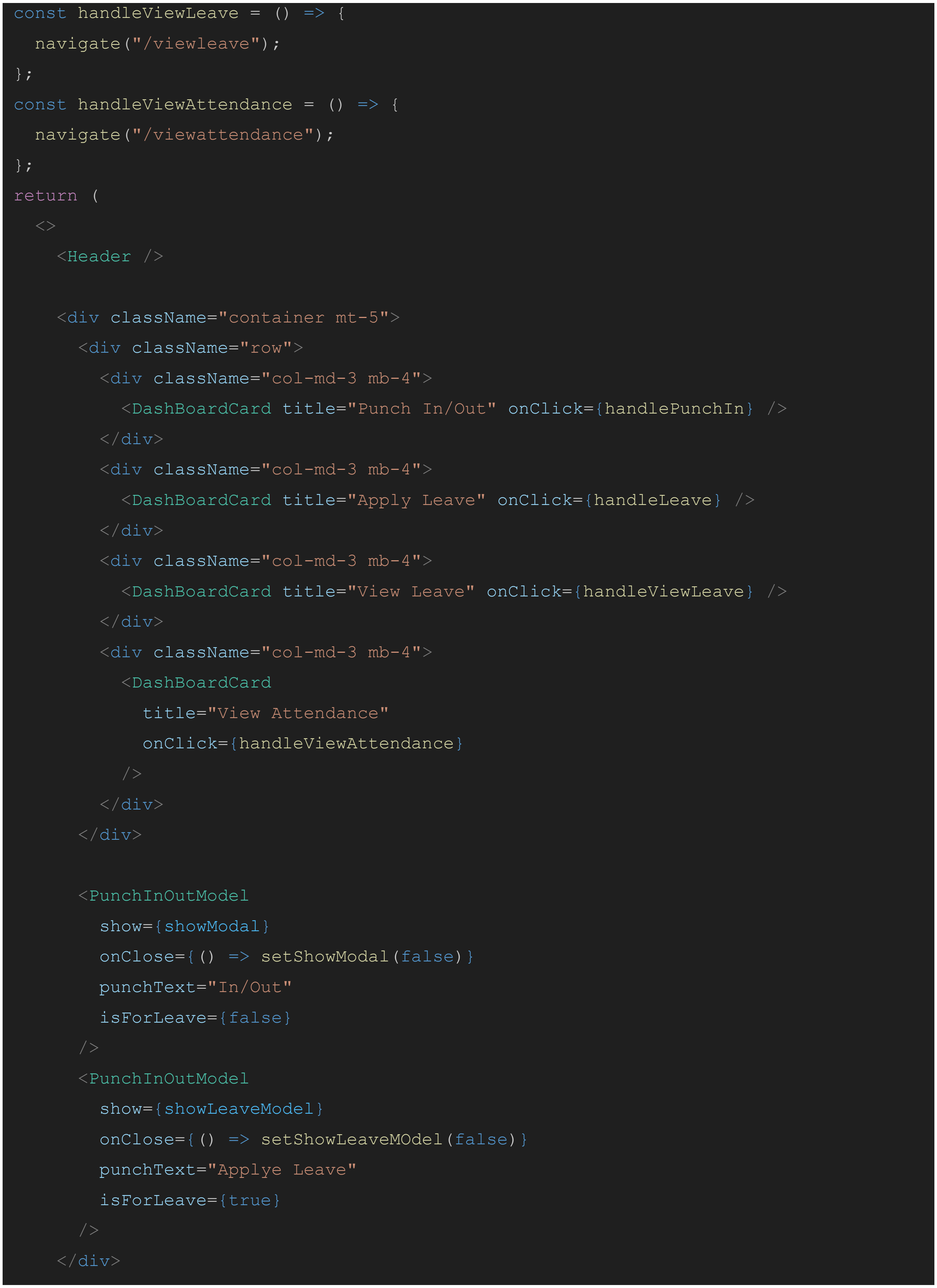


```
    <Header />
    <div className="container mt-5">
      <h3>Manage Leave Requests</h3>
      <table className="table table-bordered table-hover">
        <thead className="table-light">
          <tr>
            <th>#</th>
            <th>Name</th>
            <th>Date</th>
            <th>Reason</th>
            <th>Status</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody>
          {leaves.map((leave, index) => (
            <tr key={leave.id + index}>
              <td>{index + 1}</td>
              <td>{leave.userName}</td>
              <td>{leave.dateTime}</td>
              <td>{leave.reason}</td>
              <td>
                <span
                  className={`badge ${
                    leave.status === "Approved"
                      ? "bg-success"
                      : leave.status === "Canceled"
                      ? "bg-danger"
                      : "bg-warning text-dark"
                  }`}
                >
                  {leave.status}
                </span>
              </td>
              <td>
                <button
                  className="btn btn-success btn-sm me-2"
                  onClick={() => handleApprove(leave)}
                  disabled={leave.status !== "Pending"}
                >
                  Approve
                </button>
```

```
                 <button
                   className="btn btn-danger btn-sm"
                   onClick={() => handleCancel(leave)}
                   disabled={leave.status !== "Pending"}
                 >
                   Cancel
                 </button>
               </td>
             </tr>
           ))}
         </tbody>
       </table>
     </div>
   </>
 );
};

export default ManageLeave;
```

PunchInOut.js

```
import React from "react";
import Header from "./Header";
import DashBoardCard from "./DashBoardCard";
import { useState } from "react";
import PunchInOutModel from "./PunchInOutModel";
import { useNavigate } from "react-router-dom";

const PunchInOut = () => {
 const navigate = useNavigate();
 const [showModal, setShowModal] = useState(false);
 const [showLeaveModel, setShowLeaveMOdel] = useState(false);

 const handlePunchIn = () => {
   // console.log("handle puch in called");

   setShowModal(true);
 };

 const handleLeave = () => {
   setShowLeaveMOdel(true);
 };
```

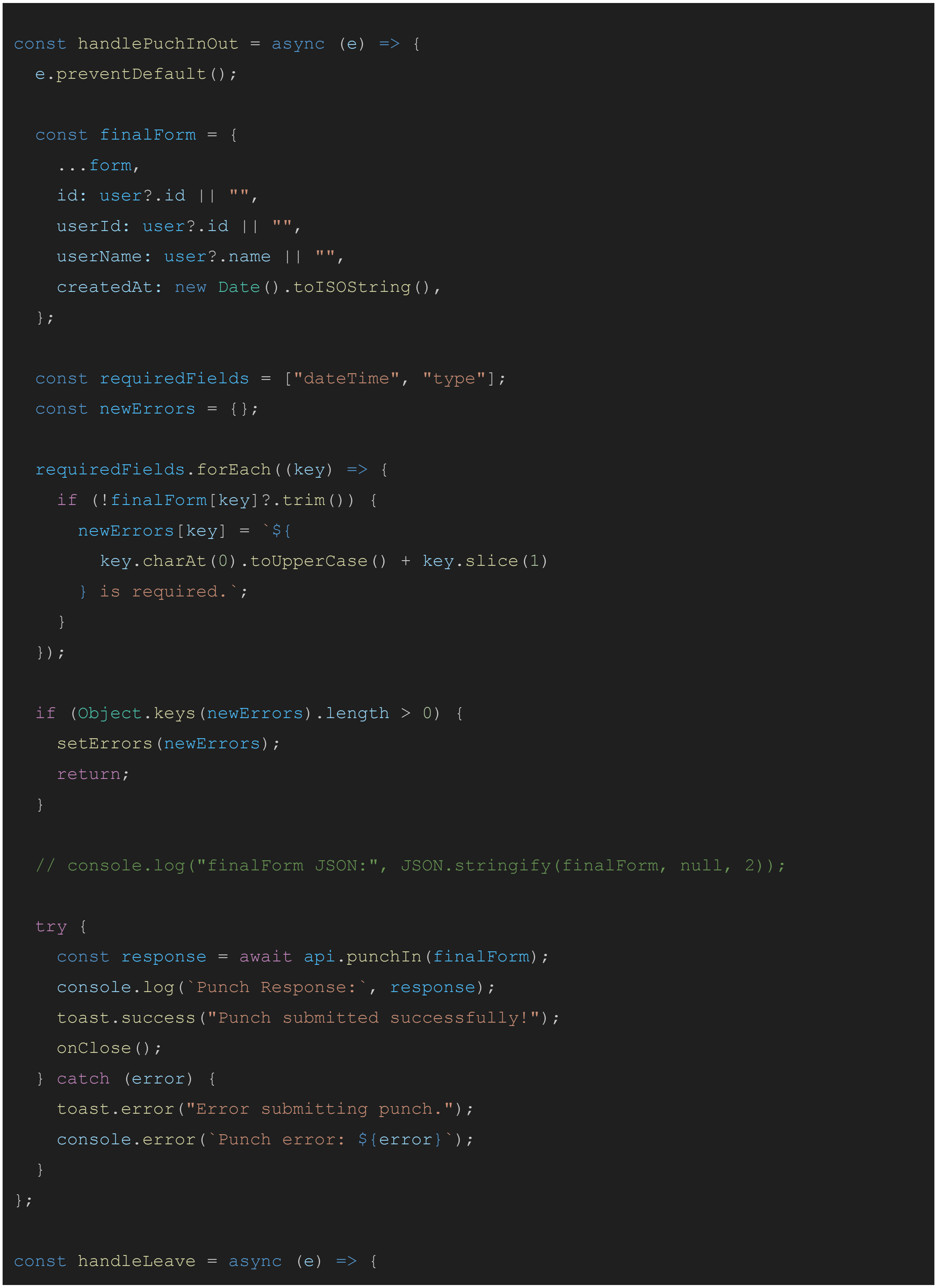


```
const handleViewLeave = () => {
  navigate("/viewleave");
};
const handleViewAttendance = () => {
  navigate("/viewattendance");
};
return (
  <>
    <Header />

    <div className="container mt-5">
      <div className="row">
        <div className="col-md-3 mb-4">
          <DashBoardCard title="Punch In/Out" onClick={handlePunchIn} />
        </div>
        <div className="col-md-3 mb-4">
          <DashBoardCard title="Apply Leave" onClick={handleLeave} />
        </div>
        <div className="col-md-3 mb-4">
          <DashBoardCard title="View Leave" onClick={handleViewLeave} />
        </div>
        <div className="col-md-3 mb-4">
          <DashBoardCard
            title="View Attendance"
            onClick={handleViewAttendance}
          />
        </div>
      </div>

      <PunchInOutModel
        show={showModal}
        onClose={() => setShowModal(false)}
        punchText="In/Out"
        isForLeave={false}
      />
      <PunchInOutModel
        show={showLeaveModel}
        onClose={() => setShowLeaveMOdel(false)}
        punchText="Applye Leave"
        isForLeave={true}
      />
    </div>
```

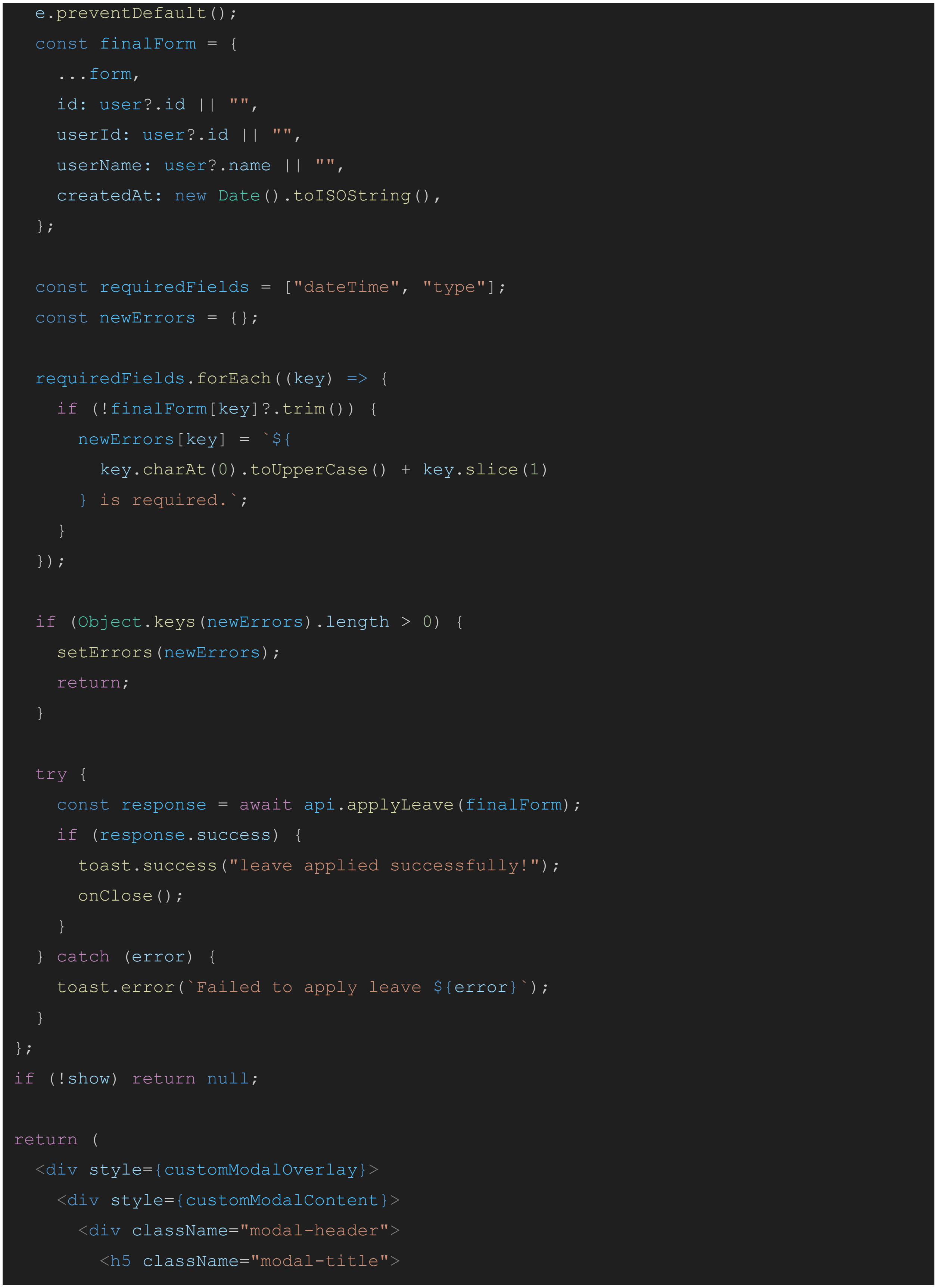
```
   </>
 );
};

export default PunchInOut;
```

PunchInOutModel.js

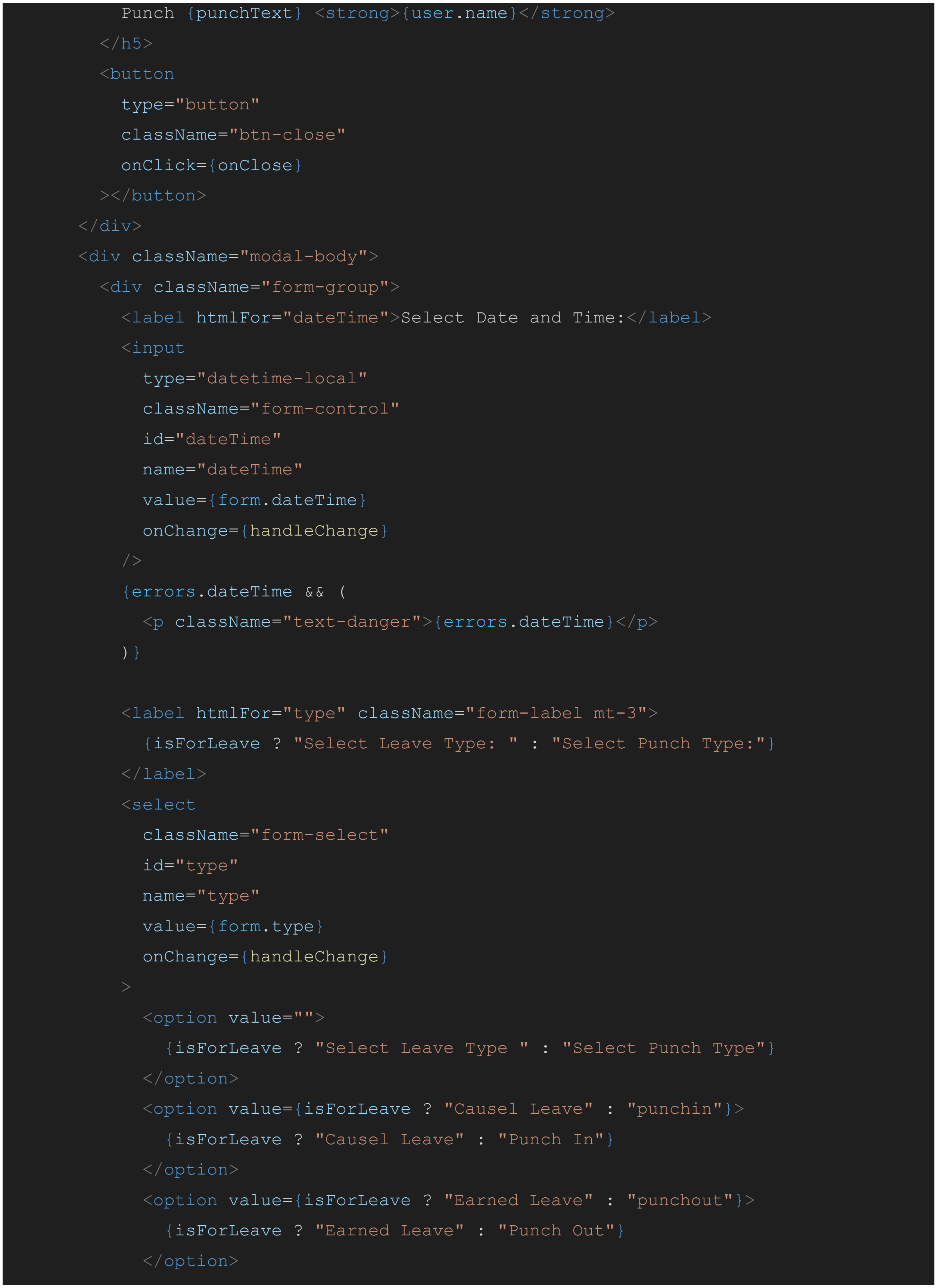
```
import React, { useState } from "react";
import { useSelector } from "react-redux";
import { customModalContent, customModalOverlay } from "../customstyles";
import { api } from "../services/api";
import { toast } from "react-toastify";

const PunchInOutModel = ({ show, onClose, punchText, isForLeave = false }) => {
 const user = useSelector((state) => state.auth.user);

 const [form, setForm] = useState({
   id: "",
   userId: "",
   userName: "",
   dateTime: "",
   type: "", // punchin / punchout
   location: "",
   notes: "",
   createdAt: "",
   reason: "",
 });

 const [errors, setErrors] = useState({});

 const handleChange = (e) => {
   const { name, value } = e.target;
   setForm((prev) => ({
     ...prev,
     [name]: value,
   }));
   setErrors((prev) => ({
     ...prev,
     [name]: "",
   }));
 };
```



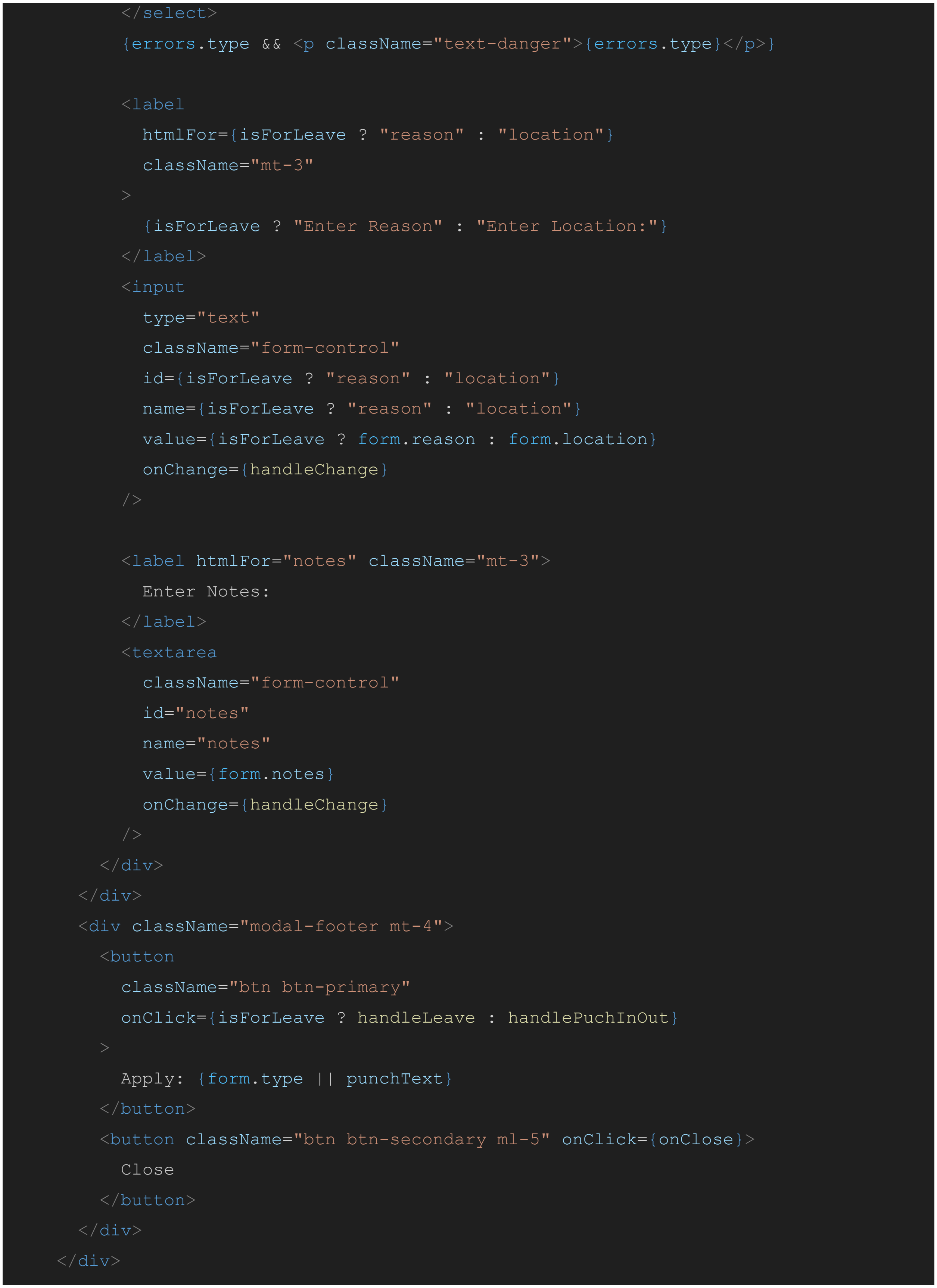
```
const handlePuchInOut = async (e) => {
  e.preventDefault();

  const finalForm = {
    ...form,
    id: user?.id || "",
    userId: user?.id || "",
    userName: user?.name || "",
    createdAt: new Date().toISOString(),
  };

  const requiredFields = ["dateTime", "type"];
  const newErrors = {};

  requiredFields.forEach((key) => {
    if (!finalForm[key]?.trim()) {
      newErrors[key] = `${
        key.charAt(0).toUpperCase() + key.slice(1)
      } is required.`;
    }
  });

  if (Object.keys(newErrors).length > 0) {
    setErrors(newErrors);
    return;
  }

  // console.log("finalForm JSON:", JSON.stringify(finalForm, null, 2));

  try {
    const response = await api.punchIn(finalForm);
    console.log(`Punch Response:`, response);
    toast.success("Punch submitted successfully!");
    onClose();
  } catch (error) {
    toast.error("Error submitting punch.");
    console.error(`Punch error: ${error}`);
  }
};

const handleLeave = async (e) => {
```



```
  e.preventDefault();
  const finalForm = {
    ...form,
    id: user?.id || "",
    userId: user?.id || "",
    userName: user?.name || "",
    createdAt: new Date().toISOString(),
  };

  const requiredFields = ["dateTime", "type"];
  const newErrors = {};

  requiredFields.forEach((key) => {
    if (!finalForm[key]?.trim()) {
      newErrors[key] = `${
        key.charAt(0).toUpperCase() + key.slice(1)
      } is required.`;
    }
  });

  if (Object.keys(newErrors).length > 0) {
    setErrors(newErrors);
    return;
  }

  try {
    const response = await api.applyLeave(finalForm);
    if (response.success) {
      toast.success("leave applied successfully!");
      onClose();
    }
  } catch (error) {
    toast.error(`Failed to apply leave ${error}`);
  }
};
if (!show) return null;

return (
  <div style={customModalOverlay}>
    <div style={customModalContent}>
      <div className="modal-header">
        <h5 className="modal-title">
```



```
          Punch {punchText} <strong>{user.name}</strong>
        </h5>
        <button
          type="button"
          className="btn-close"
          onClick={onClose}
        ></button>
      </div>
      <div className="modal-body">
        <div className="form-group">
          <label htmlFor="dateTime">Select Date and Time:</label>
          <input
            type="datetime-local"
            className="form-control"
            id="dateTime"
            name="dateTime"
            value={form.dateTime}
            onChange={handleChange}
          />
          {errors.dateTime && (
            <p className="text-danger">{errors.dateTime}</p>
          )}

          <label htmlFor="type" className="form-label mt-3">
            {isForLeave ? "Select Leave Type: " : "Select Punch Type:"}
          </label>
          <select
            className="form-select"
            id="type"
            name="type"
            value={form.type}
            onChange={handleChange}
          >
            <option value="">
              {isForLeave ? "Select Leave Type " : "Select Punch Type"}
            </option>
            <option value={isForLeave ? "Causel Leave" : "punchin"}>
              {isForLeave ? "Causel Leave" : "Punch In"}
            </option>
            <option value={isForLeave ? "Earned Leave" : "punchout"}>
              {isForLeave ? "Earned Leave" : "Punch Out"}
            </option>
```



```
      </select>
      {errors.type && <p className="text-danger">{errors.type}</p>}

      <label
        htmlFor={isForLeave ? "reason" : "location"}
        className="mt-3"
      >
        {isForLeave ? "Enter Reason" : "Enter Location:"}
      </label>
      <input
        type="text"
        className="form-control"
        id={isForLeave ? "reason" : "location"}
        name={isForLeave ? "reason" : "location"}
        value={isForLeave ? form.reason : form.location}
        onChange={handleChange}
      />

      <label htmlFor="notes" className="mt-3">
        Enter Notes:
      </label>
      <textarea
        className="form-control"
        id="notes"
        name="notes"
        value={form.notes}
        onChange={handleChange}
      />
    </div>
  </div>
  <div className="modal-footer mt-4">
    <button
      className="btn btn-primary"
      onClick={isForLeave ? handleLeave : handlePuchInOut}
    >
      Apply: {form.type || punchText}
    </button>
    <button className="btn btn-secondary ml-5" onClick={onClose}>
      Close
    </button>
  </div>
</div>
```

```
   </div>
 );
};

export default PunchInOutModel;
```

ViewAttendance.js

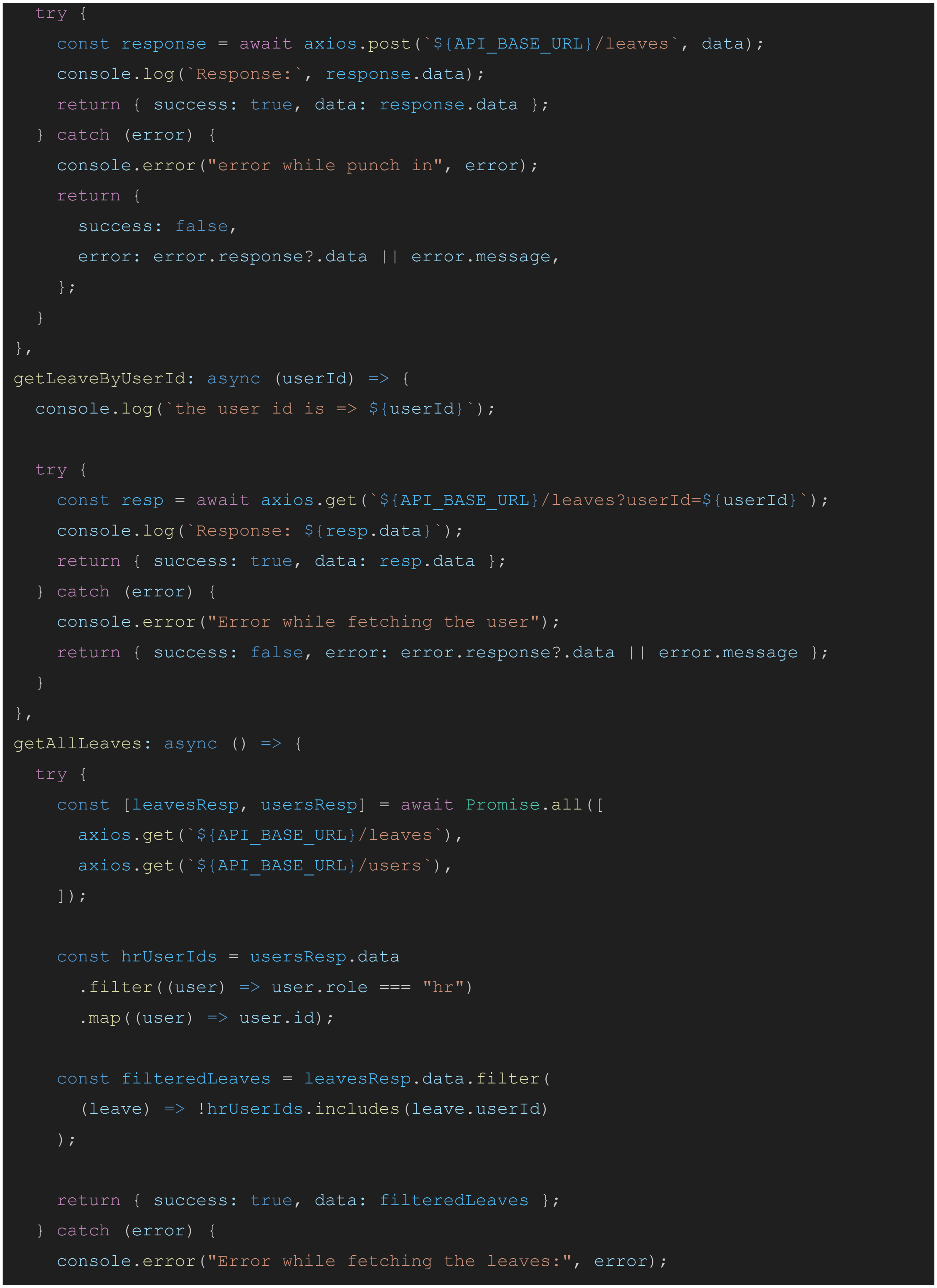
```
import React from "react";
import { useState, useEffect } from "react";
import { api } from "../services/api";
import Header from "./Header";
import { listStyle } from "../customstyles";
import { useSelector, useDispatch } from "react-redux";

const ViewAttendance = () => {
 const [attendance, setAttendance] = useState([]);
 const user = useSelector((state) => state.auth.user);

 useEffect(() => {
   async function fetchAttendance(userId) {
     const response = await api.getAllPunchInOutData(userId);
     if (response.success) {
       setAttendance(response.data);
     }
   }
   fetchAttendance(user.id);
 }, [user]);

 return (
   <>
     <Header />
     <h3 className="mt-2 mx-5">Attendance Data </h3>
     <ul className="list-group list-group-flush" style={listStyle}>
       {attendance.map((punchinut) => (
         <li key={punchinut.id} className="list-group-item">
           <strong>Punch DateTimeL</strong> {punchinut.dateTime}{" "}
           <strong>Punch Type: </strong>
           {punchinut.type}
         </li>
       ))}
     </ul>
   </>
 );
};

export default ViewAttendance;
```

ViewLeave.js

```
import React from "react";
import { useState, useEffect } from "react";
import { api } from "../services/api";
import Header from "./Header";
import { useSelector, useDispatch } from "react-redux";

const ViewLeave = () => {
 const [leaves, setLeave] = useState([]);
 const user = useSelector((state) => state.auth.user);
 console.log(`the loggin userd data is ${JSON.stringify(user)}`);

 useEffect(() => {
   async function fetchLeave(userId) {
     const resp = await api.getLeaveByUserId(userId);
     if (resp.success) {
       setLeave(resp.data);
     }
   }
   fetchLeave(user.id);
 }, [user]);

 return (
   <>
     <Header />
     <h3 className="mt-2 mx-5">Leave Data </h3>

     <ul className="list-group list-group-flush">
       {leaves.map((leave) => (
         <li key={leave.id} className="list-group-item">
           <strong>Leave Date Time: </strong>
           {leave.dateTime}
           <strong> Leave Type </strong>
           {leave.type}
         </li>
       ))}
     </ul>
   </>
 );
};

export default ViewLeave;
```

Api.js

```
import axios from "axios";
import { data } from "react-router-dom";

const API_BASE_URL = "http://localhost:3001";
export const api = {
 addEmployee: async (data) => {
   try {
     const response = await axios.post(`${API_BASE_URL}/users`, data);
     console.log("Response:", response.data);
     return { success: "true", data: response.data };
   } catch (error) {
     console.error("Error occurred while adding employee:", error);
     return { success: "false", error: error.response?.data || error.message };
   }
 },

 punchIn: async (data) => {
   // console.log("punc in data form api ", JSON.stringify(data, null, 2));

   try {
     const response = await axios.post(`${API_BASE_URL}/punches`, data);
     console.log(`Response:`, response.data);
     return { success: true, data: response.data };
   } catch (error) {
     console.error("error while punch in", error);
     return {
       success: false,
       error: error.response?.data || error.message,
     };
   }
 },
 applyLeave: async (data) => {
```



```
  try {
    const response = await axios.post(`${API_BASE_URL}/leaves`, data);
    console.log(`Response:`, response.data);
    return { success: true, data: response.data };
  } catch (error) {
    console.error("error while punch in", error);
    return {
      success: false,
      error: error.response?.data || error.message,
    };
  }
},
getLeaveByUserId: async (userId) => {
  console.log(`the user id is => ${userId}`);

  try {
    const resp = await axios.get(`${API_BASE_URL}/leaves?userId=${userId}`);
    console.log(`Response: ${resp.data}`);
    return { success: true, data: resp.data };
  } catch (error) {
    console.error("Error while fetching the user");
    return { success: false, error: error.response?.data || error.message };
  }
},
getAllLeaves: async () => {
  try {
    const [leavesResp, usersResp] = await Promise.all([
      axios.get(`${API_BASE_URL}/leaves`),
      axios.get(`${API_BASE_URL}/users`),
    ]);

    const hrUserIds = usersResp.data
      .filter((user) => user.role === "hr")
      .map((user) => user.id);

    const filteredLeaves = leavesResp.data.filter(
      (leave) => !hrUserIds.includes(leave.userId)
    );

    return { success: true, data: filteredLeaves };
  } catch (error) {
    console.error("Error while fetching the leaves:", error);
```

```
     return { success: false, error: error.response?.data || error.message };
   }
 },
 getAllPunchInOutData: async (userId) => {
   try {
     const resp = await axios.get(`${API_BASE_URL}/punches?userId=${userId}`);
     console.log(`Response: ${resp.data}`);
     return { success: true, data: resp.data };
   } catch (error) {
     console.error("Error while fetching the user");
     return { success: false, error: error.response?.data || error.message };
   }
 },

 updateLeave: async (leaveId, updatedData) => {
   try {
     const response = await axios.patch(
       `${API_BASE_URL}/leaves/${leaveId}`,
       updatedData
     );
     return { success: true, data: response.data };
   } catch (error) {
     console.error("Update Leave Error", error);
     return { success: false, error: error.response?.data || error.message };
   }
 },

 getAllEmployee: async () => {
   try {
     const resp = await axios.get(`${API_BASE_URL}/users`);
     console.log(`Response: ${resp.data}`);
     return { success: true, data: resp.data };
   } catch (error) {
     console.error("Error while fetching the user");
     return { success: false, error: error.response?.data || error.message };
   }
 },
};
```

AuthSlice.js

```
import { createSlice } from "@reduxjs/toolkit";
```

```
const authSlice = createSlice({
 name: "auth",

 initialState: {
   user: null,
 },
 reducers: {
   login: (state, action) => {
     state.user = action.payload;
   },
   logout: (state) => {
     state.user = null;
   },
 },
});

export const { login, logout } = authSlice.actions;
// authslice.js
export const authReducer = authSlice.reducer;
```

Store.js

```
import { configureStore } from "@reduxjs/toolkit";
import { authReducer } from "./authslice";

export const store = configureStore({
 reducer: {
   auth: authReducer,
 },
});
```

card.js style

```
export const cardstyle = {
 width: "18rem",
 padding: "20px",
 margin: "20px",
};
```

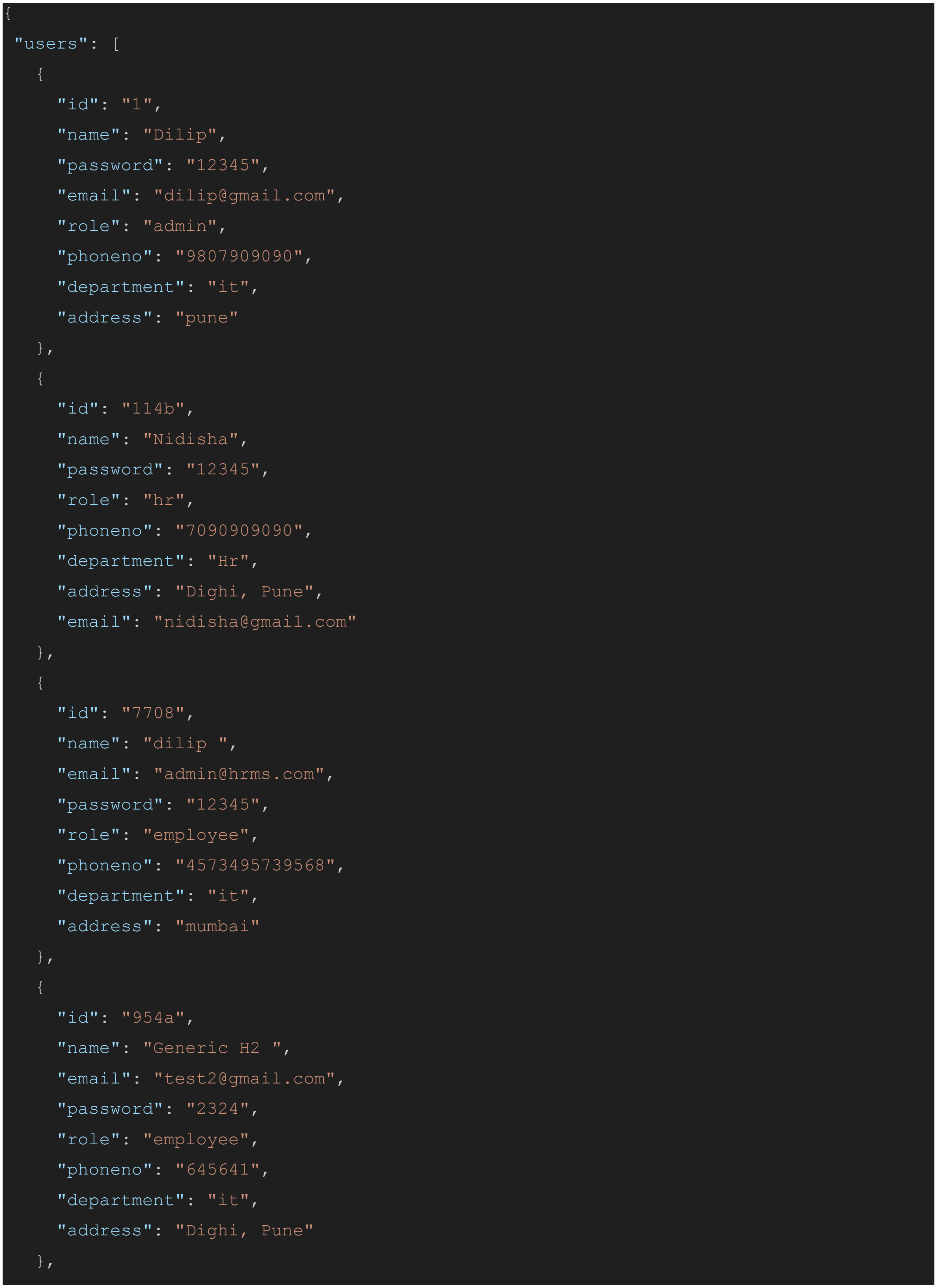
Header.css

```css
.header-container {
 display: flex;
 justify-content: space-between;
 align-items: center;
 padding: 16px 24px;
 background-color: #f8f9fa;
 box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
 position: relative;
}

.header-title {
 font-size: 18px;
 font-weight: 600;
}

.header-profile {
 position: relative;
}

.profile-circle {
 width: 40px;
 height: 40px;
 background-color: #007bff;
 color: white;
 border-radius: 50%;
 display: flex;
 align-items: center;
 justify-content: center;
 font-weight: bold;
 cursor: pointer;
}

.dropdown-menu {
 position: absolute;
 top: 50px;
 right: 0;
 background-color: white;
 padding: 10px;
 border: 1px solid #ddd;
 border-radius: 6px;
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.15);
 z-index: 1000;
}
```

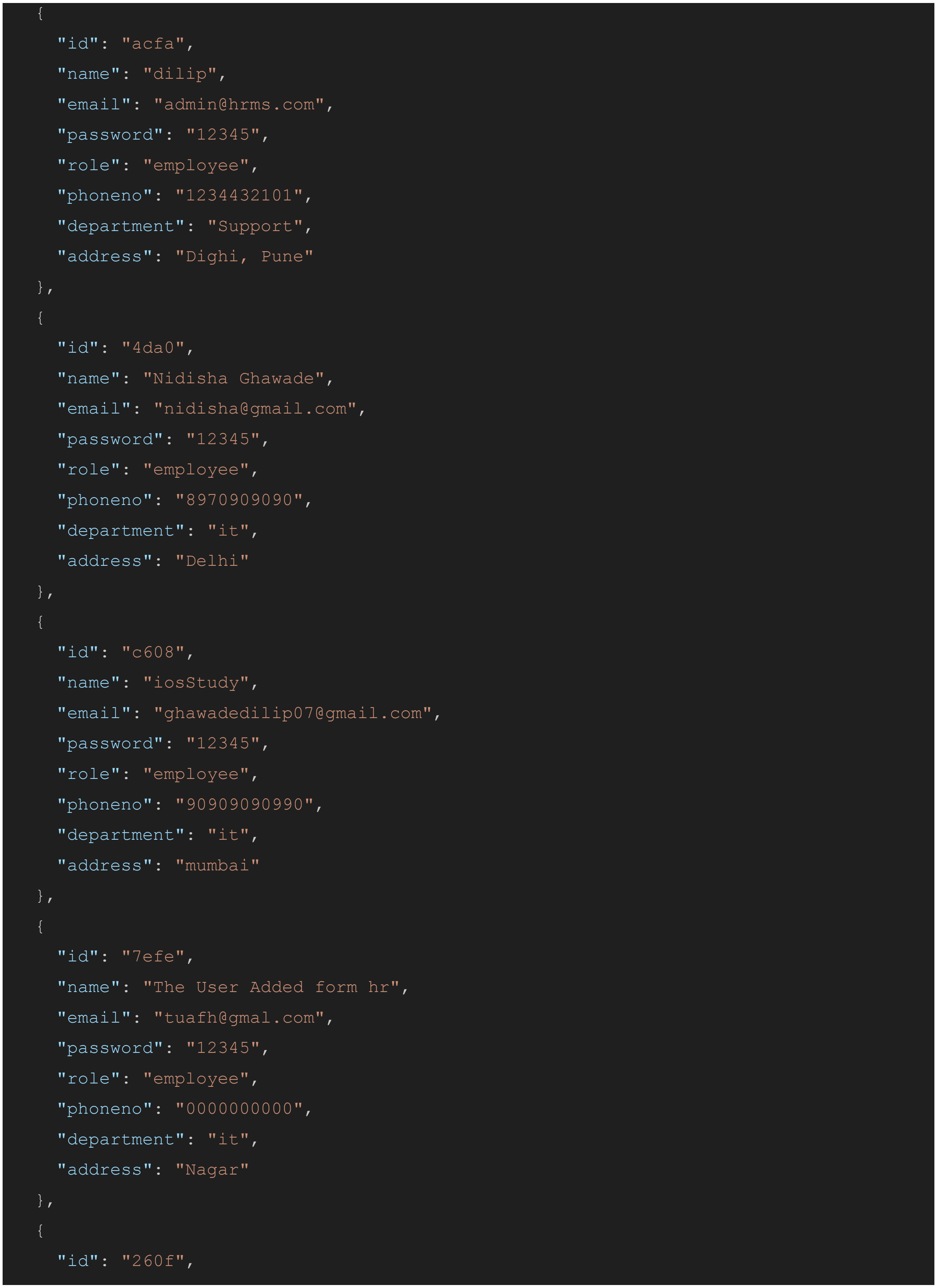
Customstyle.js

```
export const backgroundStyle = {
 background: "linear-gradient(135deg, #0033a0 0%, #00a4bd 100%)",
 minHeight: "100vh",
 display: "flex",
 alignItems: "center",
 justifyContent: "center",
};

export const listStyle = {
 width: "50%",
 padding: "10px",
};

export const customModalOverlay = {
 position: "fixed",
 top: "0",
 left: "0",
 width: "100%",
 height: "100%",
 backgroundColor: "rgba(0, 0, 0, 0.5)",
 zIndex: "1050",
 display: "flex",
 alignItems: "center",
 justifyContent: "center",
};

export const customModalContent = {
 background: "white",
 padding: "20px",
 borderRadius: "10px",
 maxWidth: "600px",
 width: "100%",
 zIndex: "1051",
 boxShadow: "0 5px 15px rgba(0, 0, 0, 0.3)",
};
```

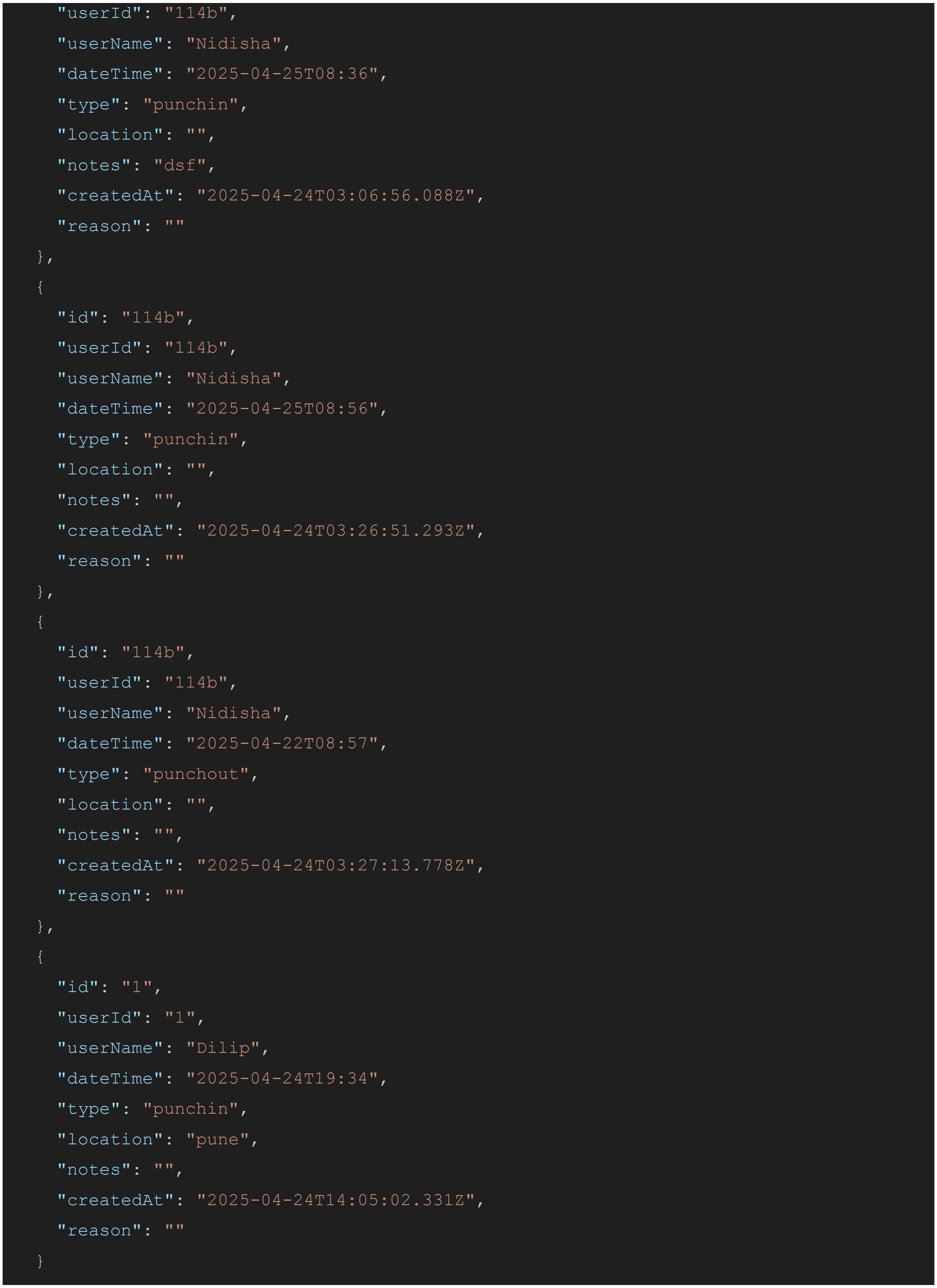
Db.json



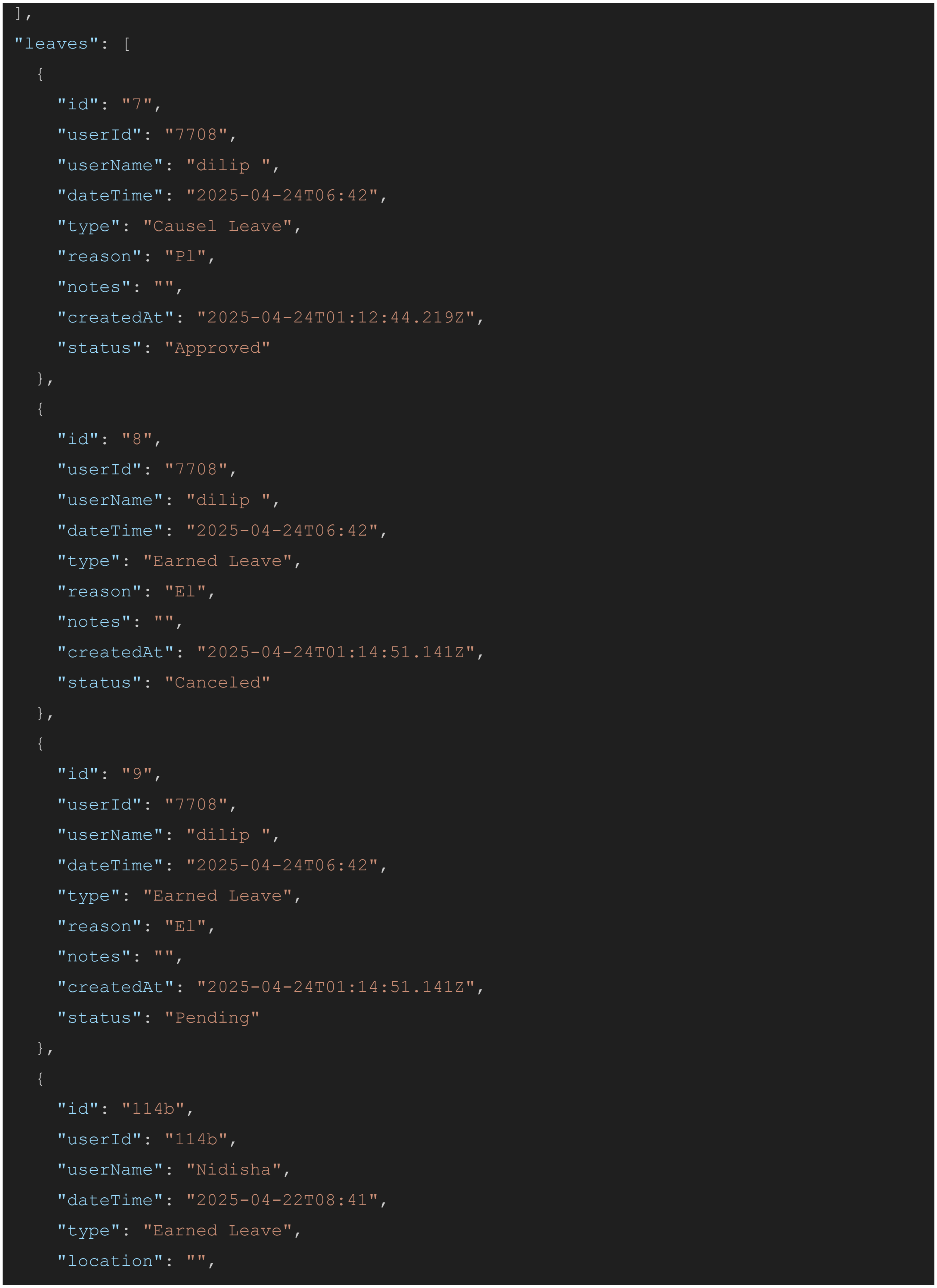
```
{
 "users": [
   {
     "id": "1",
     "name": "Dilip",
     "password": "12345",
     "email": "dilip@gmail.com",
     "role": "admin",
     "phoneno": "9807909090",
     "department": "it",
     "address": "pune"
   },
   {
     "id": "114b",
     "name": "Nidisha",
     "password": "12345",
     "role": "hr",
     "phoneno": "7090909090",
     "department": "Hr",
     "address": "Dighi, Pune",
     "email": "nidisha@gmail.com"
   },
   {
     "id": "7708",
     "name": "dilip ",
     "email": "admin@hrms.com",
     "password": "12345",
     "role": "employee",
     "phoneno": "4573495739568",
     "department": "it",
     "address": "mumbai"
   },
   {
     "id": "954a",
     "name": "Generic H2 ",
     "email": "test2@gmail.com",
     "password": "2324",
     "role": "employee",
     "phoneno": "645641",
     "department": "it",
     "address": "Dighi, Pune"
   },
```



```
  {
    "id": "acfa",
    "name": "dilip",
    "email": "admin@hrms.com",
    "password": "12345",
    "role": "employee",
    "phoneno": "1234432101",
    "department": "Support",
    "address": "Dighi, Pune"
  },
  {
    "id": "4da0",
    "name": "Nidisha Ghawade",
    "email": "nidisha@gmail.com",
    "password": "12345",
    "role": "employee",
    "phoneno": "8970909090",
    "department": "it",
    "address": "Delhi"
  },
  {
    "id": "c608",
    "name": "iosStudy",
    "email": "ghawadedilip07@gmail.com",
    "password": "12345",
    "role": "employee",
    "phoneno": "90909090990",
    "department": "it",
    "address": "mumbai"
  },
  {
    "id": "7efe",
    "name": "The User Added form hr",
    "email": "tuafh@gmal.com",
    "password": "12345",
    "role": "employee",
    "phoneno": "0000000000",
    "department": "it",
    "address": "Nagar"
  },
  {
    "id": "260f",
    "name": "Pradeep",
    "email": "pradeep@gmail.com",
    "password": "12345",
    "role": "employee",
    "phoneno": "9090909090",
    "department": "it",
    "address": "Beed"
  }
],
"punches": [
  {
    "id": "7708",
    "userId": "7708",
    "userName": "dilip ",
    "dateTime": "2025-04-23T09:23",
    "type": "punchin",
    "location": "pune",
    "notes": "This is for testing purpose",
    "createdAt": "2025-04-23T04:06:14.695Z"
  },
  {
    "id": "7708",
    "userId": "7708",
    "userName": "dilip ",
    "dateTime": "2025-04-24T06:31",
    "type": "punchin",
    "location": "new delhi",
    "notes": "na",
    "createdAt": "2025-04-24T01:06:19.985Z"
  },
  {
    "id": "7708",
    "userId": "7708",
    "userName": "dilip ",
    "dateTime": "2025-04-24T07:14",
    "type": "punchout",
    "location": "",
    "notes": "",
    "createdAt": "2025-04-24T01:44:35.605Z"
  },
  {
    "id": "114b",
```



```
    "userId": "114b",
    "userName": "Nidisha",
    "dateTime": "2025-04-25T08:36",
    "type": "punchin",
    "location": "",
    "notes": "dsf",
    "createdAt": "2025-04-24T03:06:56.088Z",
    "reason": ""
  },
  {
    "id": "114b",
    "userId": "114b",
    "userName": "Nidisha",
    "dateTime": "2025-04-25T08:56",
    "type": "punchin",
    "location": "",
    "notes": "",
    "createdAt": "2025-04-24T03:26:51.293Z",
    "reason": ""
  },
  {
    "id": "114b",
    "userId": "114b",
    "userName": "Nidisha",
    "dateTime": "2025-04-22T08:57",
    "type": "punchout",
    "location": "",
    "notes": "",
    "createdAt": "2025-04-24T03:27:13.778Z",
    "reason": ""
  },
  {
    "id": "1",
    "userId": "1",
    "userName": "Dilip",
    "dateTime": "2025-04-24T19:34",
    "type": "punchin",
    "location": "pune",
    "notes": "",
    "createdAt": "2025-04-24T14:05:02.331Z",
    "reason": ""
  }
```



```
],
"leaves": [
  {
    "id": "7",
    "userId": "7708",
    "userName": "dilip ",
    "dateTime": "2025-04-24T06:42",
    "type": "Causel Leave",
    "reason": "Pl",
    "notes": "",
    "createdAt": "2025-04-24T01:12:44.219Z",
    "status": "Approved"
  },
  {
    "id": "8",
    "userId": "7708",
    "userName": "dilip ",
    "dateTime": "2025-04-24T06:42",
    "type": "Earned Leave",
    "reason": "El",
    "notes": "",
    "createdAt": "2025-04-24T01:14:51.141Z",
    "status": "Canceled"
  },
  {
    "id": "9",
    "userId": "7708",
    "userName": "dilip ",
    "dateTime": "2025-04-24T06:42",
    "type": "Earned Leave",
    "reason": "El",
    "notes": "",
    "createdAt": "2025-04-24T01:14:51.141Z",
    "status": "Pending"
  },
  {
    "id": "114b",
    "userId": "114b",
    "userName": "Nidisha",
    "dateTime": "2025-04-22T08:41",
    "type": "Earned Leave",
    "location": "",
     "notes": "",
     "createdAt": "2025-04-24T03:11:53.554Z",
     "reason": "EL"
   },
   {
     "id": "1",
     "userId": "1",
     "userName": "Dilip",
     "dateTime": "2025-04-26T19:35",
     "type": "Earned Leave",
     "location": "",
     "notes": "",
     "createdAt": "2025-04-24T14:05:19.296Z",
     "reason": ""
   }
 ]
}
```