**HR Portal:**

Login.jsx

import React, { useState } from 'react';

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

import employeesData from './hr.json'; // Import employee data

import '../App.css'; // Ensure you have the CSS file imported

const LoginPage = () => {

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

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

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

  const navigate = useNavigate();

  const handleSubmit = (e) => {

    e.preventDefault();

    setError('');

    // Check credentials against hardcoded values

    const user = employeesData.find(emp => emp.email === email && emp.password === password);

    if (user) {

      localStorage.setItem('isAuthenticated', 'true');

      localStorage.setItem('loggedInUser', JSON.stringify(user));

      navigate('/dashboard');

    } else {

      setError('Invalid credentials!');

    }

  };

  return (

    <div className="login-page">

      <div className="form-container">

        <h3>Login</h3>

        <form onSubmit={handleSubmit}>

          <div className="form-control">

            <label htmlFor="email">Email:</label>

            <input

              type="email"

              id="email"

              value={email}

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

              required

            />

          </div>

          <div className="form-control">

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

            <input

              type="password"

              id="password"

              value={password}

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

              required

            />

          </div>

          {error && <p className="error">{error}</p>}

          <button type="submit">Login</button>

        </form>

      </div>

    </div>

  );

};

export default LoginPage;

2. EmployeeChart

import React from 'react';

import { Bar } from 'react-chartjs-2';

import {

  Chart as ChartJS,

  CategoryScale,

  LinearScale,

  BarElement,

  Title,

  Tooltip,

  Legend,

} from 'chart.js';

ChartJS.register(

  CategoryScale,

  LinearScale,

  BarElement,

  Title,

  Tooltip,

  Legend

);

const EmployeeChart = ({ data }) => {

  // Ensure to use 'workedHours' field (from updated HrDashboard component)

  const generateColors = (numColors) => {

    return Array.from({ length: numColors }, () =>

      `rgba(${Math.floor(Math.random() \* 255)}, ${Math.floor(

        Math.random() \* 255

      )}, ${Math.floor(Math.random() \* 255)}, 1)`

    );

  };

  const borderColors = generateColors(data.length);

  const chartData = {

    labels: data.map((emp) => emp.name),

    datasets: [

      {

        label: 'Hours Worked',

        data: data.map((emp) => emp.workedHours || 0), // Ensure we use 'workedHours'

        backgroundColor: 'transparent', // Transparent bars

        borderColor: borderColors, // Dynamic border colors

        borderWidth: 2,

      },

    ],

  };

  const options = {

    scales: {

      y: {

        beginAtZero: true,

        title: {

          display: true,

          text: 'Hours Worked',

        },

      },

    },

    plugins: {

      legend: {

        display: true,

      },

      tooltip: {

        enabled: true,

      },

    },

    layout: {

      padding: 20,

    },

  };

  return (

    <div style={{ padding: '1rem' }}>

      <Bar data={chartData} options={options} style={{ backgroundColor: 'transparent' }} />

    </div>

  );

};

export default EmployeeChart;

3. HrDashboard.jsx

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

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

import employeesData from "./hr.json";

import EmployeeChart from "./EmployeeChart";

import '../App.css'; // Ensure you have the CSS file imported

const HrDashboard = () => {

  const [employees, setEmployees] = useState(employeesData);

  const [selectedEmployee, setSelectedEmployee] = useState(null);

  const [timing, setTiming] = useState({ in: "", out: "" });

  const [leave, setLeave] = useState("");

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

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

  const [loggedInUser, setLoggedInUser] = useState(null);

  const navigate = useNavigate();

  useEffect(() => {

    const user = JSON.parse(localStorage.getItem('loggedInUser'));

    if (user) {

      setLoggedInUser(user);

    } else {

      navigate('/login');

    }

  }, [navigate]);

  // Function to calculate worked hours based on check-in and check-out time

  const calculateWorkedHours = (checkInTime, checkOutTime) => {

    const checkIn = new Date(`1970-01-01T${checkInTime}:00`);

    const checkOut = new Date(`1970-01-01T${checkOutTime}:00`);

    const diff = checkOut - checkIn;

    return diff / (1000 \* 60 \* 60); // Return hours

  };

  const handleSelectEmployee = (employee) => {

    setSelectedEmployee(employee);

    setTiming({ in: "", out: "" });

    setLeave("");

    setError("");

  };

  const handleTimingChange = (e) => {

    setTiming({ ...timing, [e.target.name]: e.target.value });

  };

  const handleLeaveChange = (e) => {

    setLeave(e.target.value);

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    setLoading(true);

    // Calculate worked hours and update the employee's timing and leave data

    const workedHours = calculateWorkedHours(timing.in, timing.out);

    const updatedEmployees = employees.map((emp) =>

      emp.id === selectedEmployee.id

        ? { ...emp, timing, leave, workedHours }

        : emp

    );

    setEmployees(updatedEmployees);

    setLoading(false);

    alert("Leave and Timings updated successfully!");

  };

  const handleLogout = () => {

    localStorage.removeItem('isAuthenticated');

    localStorage.removeItem('loggedInUser');

    navigate('/login');

  };

  return (

    <div className="dashboard-container">

      <div className="logout-container">

        {loggedInUser && (

          <div className="logged-in-user">

            <p>Welcome, {loggedInUser.name}!</p>

            <button onClick={handleLogout}>Logout</button>

          </div>

        )}

      </div>

      <h2>HR Dashboard</h2>

      <h3>Select Employee</h3>

      <select onChange={(e) => handleSelectEmployee(employees[e.target.value])}>

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

        {employees.map((employee, index) => (

          <option key={employee.id} value={index}>

            {employee.name}

          </option>

        ))}

      </select>

      {selectedEmployee && (

        <div className="employee-details">

          <h3>Employee Details: {selectedEmployee.name}</h3>

          <p>Email: {selectedEmployee.email}</p>

          <p>Role: {selectedEmployee.role}</p>

          <form onSubmit={handleSubmit}>

            <div className="form-control">

              <label>Office Timing (In):</label>

              <input

                type="time"

                name="in"

                value={timing.in}

                onChange={handleTimingChange}

                required

              />

            </div>

            <div className="form-control">

              <label>Office Timing (Out):</label>

              <input

                type="time"

                name="out"

                value={timing.out}

                onChange={handleTimingChange}

                required

              />

            </div>

            {error && <p className="error">{error}</p>}

            <button type="submit" disabled={loading}>

              {loading ? 'Saving...' : 'Save Timings and Leave'}

            </button>

          </form>

        </div>

      )}

      <div className="chart-container">

        <h3>Employee Hours Worked</h3>

        <EmployeeChart data={employees} />

      </div>

    </div>

  );

};

export default HrDashboard;

4.PrivateRoute.jsx

import React, { useEffect } from 'react';

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

const PrivateRoute = ({ children }) => {

  const isAuthenticated = localStorage.getItem('isAuthenticated') === 'true';

  useEffect(() => {

    // Clear localStorage to simulate logout on page refresh

    if (!isAuthenticated) {

      localStorage.removeItem('isAuthenticated');

    }

  }, [isAuthenticated]);

  return isAuthenticated ? children : <Navigate to="/login" />;

};

export default PrivateRoute;

5.signup.jsx

import React, { useState } from "react";

import '../App.css';

const SignupPage = () => {

  const [name, setName] = useState("");

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

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

  const [role, setRole] = useState("");

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

  const [success, setSuccess] = useState("");

  const handleSubmit = (e) => {

    e.preventDefault();

    setError("");

    setSuccess("");

    // Basic validation

    if (!name || !email || !password || !role) {

      setError("All fields are required!");

      return;

    }

    // Simulate an API call

    setTimeout(() => {

      console.log("Sign-Up Data:", { name, email, password, role });

      setSuccess("Sign-up successful!");

    }, 1000);

  };

  return (

    <div className="form-container">

      <h3>Sign Up</h3>

      <form onSubmit={handleSubmit}>

        <div className="form-control">

          <label htmlFor="name">Name</label>

          <input

            type="text"

            id="name"

            value={name}

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

            required

          />

        </div>

        <div className="form-control">

          <label htmlFor="email">Email</label>

          <input

            type="email"

            id="email"

            value={email}

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

            required

          />

        </div>

        <div className="form-control">

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

          <input

            type="password"

            id="password"

            value={password}

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

            required

          />

        </div>

        <div className="form-control">

          <label>Role</label>

          <div>

            <input

              type="radio"

              id="hr"

              name="role"

              value="HR"

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

              required

            />

            <label htmlFor="hr">HR</label>

          </div>

          <div>

            <input

              type="radio"

              id="employee"

              name="role"

              value="Employee"

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

              required

            />

            <label htmlFor="employee">Employee</label>

          </div>

        </div>

        {error && <p className="error">{error}</p>}

        {success && <p className="success">{success}</p>}

        <button type="submit">Sign Up</button>

      </form>

    </div>

  );

};

export default SignupPage;