**Week 12**

**Project frontend and backend connection.**

* Project frontend.
* Adding fetch function to code.

**FrontendCodes:**

**Register.js**

import React, { useState } from "react";

import "./Register.css";

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

import logo from './Image/logo.png'; // Ensure the logo path is correct

function Register() {

  const [firstName, setFirstName] = useState("");

  const [lastName, setLastName] = useState("");

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

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

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

  const [errorMessage, setErrorMessage] = useState(""); // State for error messages

  const navigate = useNavigate();

  const handleSubmit = async (e) => {

    e.preventDefault();

    const employeeData = {

      empId: email,

      name: `${firstName} ${lastName}`,

      email: email,

      password: password,

      contactNo: phone,

      role: "Employee",

      status: "Active"

    };

    try {

      const response = await fetch("http://localhost:8080/emp", {

        method: "POST",

        headers: {

          "Content-Type": "application/json",

        },

        body: JSON.stringify(employeeData),

      });

      if (response.ok) {

        const data = await response.json();

        console.log("Success:", data);

        navigate("/Login");

      } else {

        const errorData = await response.json();

        console.error("Error:", errorData);

        setErrorMessage(errorData.message || "Registration failed");

      }

    } catch (error) {

      console.error("Error:", error);

      setErrorMessage("An error occurred during registration.");

    }

  };

  return (

    <div className="register-container">

      <div className="logo">

        <h1>Hardik Herbals</h1>

        <img src={logo} alt="logo" />

      </div>

      <div className="form-section">

        <h2>Let's Get Started</h2>

        {errorMessage && <p className="error-message">{errorMessage}</p>} {/\* Display error message \*/}

        <form onSubmit={handleSubmit}>

          <div className="input-group">

            <label>First Name:</label>

            <input

              type="text"

              value={firstName}

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

              required

            />

          </div>

          <div className="input-group">

            <label>Last Name:</label>

            <input

              type="text"

              value={lastName}

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

              required

            />

          </div>

          <div className="input-group">

            <label>Email:</label>

            <input

              type="email"

              value={email}

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

              required

            />

          </div>

          <div className="input-group">

            <label>Password:</label>

            <input

              type="password"

              value={password}

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

              required

            />

          </div>

          <div className="input-group">

            <label>Phone Number:</label>

            <input

              type="tel"

              value={phone}

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

              placeholder="+91"

              required

            />

          </div>

          <button type="submit" className="submit-btn">Finish Up</button>

        </form>

      </div>

    </div>

  );

}

export default Register;

**Login.js**

* Make changes in Login.js
* Adding fetch function to fetch the data.

import React, { useState } from 'react';

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

import './login.css';

import logo from './Image/logo.png';

function Login() {

    const [email, setEmail] = useState(''); // Can be email for admin or empId for employee

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

    const [isAdmin, setIsAdmin] = useState(false);

    const [errorMessage, setErrorMessage] = useState('');

    const navigate = useNavigate();

    const handleSubmit = async (event) => {

        event.preventDefault();

        setErrorMessage(''); // Clear previous error message

        const loginData = isAdmin

            ? { email, password }  // Admin login expects email and password

            : { empId: email, password }; // Employee login expects empId and password

        const endpoint = isAdmin

            ? "http://localhost:8080/login/admin"  // Admin login endpoint

            : "http://localhost:8080/login";      // Employee login endpoint

        try {

            const response = await fetch(endpoint, {

                method: "POST",

                headers: {

                    "Content-Type": "application/json",

                },

                body: JSON.stringify(loginData),

            });

            if (response.ok) {

                if (!isAdmin) {

                    localStorage.setItem('empId', email); // Store empId

                    navigate("/employee/home"); // Redirect to Employee Home

                } else {

                    navigate("/admin/HomeAdmin"); // Redirect to Admin Home

                }

            } else {

                if (response.status === 401) {

                    setErrorMessage("Invalid credentials. Please try again.");

                } else if (response.status === 500) {

                    setErrorMessage("Server error. Please try again later.");

                } else {

                    setErrorMessage("An error occurred while logging in. Please try again later.");

                }

            }

        } catch (error) {

            setErrorMessage("An error occurred during login. Please try again later.");

            console.error("Network Error:", error);

        }

    };

    return (

        <div className="login-container">

            <div className="logo-container">

                <img src={logo} alt="logo" width="100vh" />

                <h1>HARDIK HERBAL</h1>

            </div>

            <div className="login-box">

                <h2 style={{ fontFamily: 'WonderThaw' }}>Continue As</h2>

                {/\* User Type Toggle \*/}

                <div className="user-type-toggle">

                    <label>

                        <input

                            type="radio"

                            value="employee"

                            checked={!isAdmin}

                            onChange={() => setIsAdmin(false)}

                        />

                        Employee

                    </label>

                    <label>

                        <input

                            type="radio"

                            value="admin"

                            checked={isAdmin}

                            onChange={() => setIsAdmin(true)}

                        />

                        Admin

                    </label>

                </div>

                {/\* Display error message if exists \*/}

                {errorMessage && <div className="error-message">{errorMessage}</div>}

                <form onSubmit={handleSubmit}>

                    <div className="input-field">

                        <i className="fas fa-user"></i>

                        <input

                            type={isAdmin ? "email" : "text"} // Use email input for Admin, text for Employee

                            placeholder={isAdmin ? "Email" : "Employee ID"} // Display appropriate placeholder

                            value={email}

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

                            required

                        />

                    </div>

                    <div className="input-field">

                        <i className="fas fa-lock"></i>

                        <input

                            type="password"

                            placeholder="Password"

                            value={password}

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

                            required

                        />

                    </div>

                    <button type="submit">Log In</button>

                    <a href="https://example.com">Forgot password?</a>

                </form>

            </div>

        </div>

    );

}

export default Login;

**When Employee Login then EmployeeHome page open then Employee can apply leave request .**

**Leaveapply.js**

import React, { useState } from 'react';

import './Leaveapply.css';

function LeaveApply() {

    const [employeeID, setEmployeeID] = useState('');

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

    const [startDate, setStartDate] = useState('');

    const [endDate, setEndDate] = useState('');

    const [leaveType, setLeaveType] = useState('Vacation');

    const [reasonForLeave, setReasonForLeave] = useState('');

    const [errorMessage, setErrorMessage] = useState('');

    const [successMessage, setSuccessMessage] = useState('');

    const handleSubmit = async (e) => {

        e.preventDefault();

        if (new Date(startDate) > new Date(endDate)) {

            setErrorMessage("End date must be later than start date.");

            return;

        }

        const leaveData = {

            empId: employeeID,

            fullName: fullName,

            startDate: startDate,

            endDate: endDate,

            type: leaveType,

            reason: reasonForLeave

        };

        try {

            const response = await fetch("http://localhost:8080/leave", {

                method: "POST",

                headers: {

                    "Content-Type": "application/json"

                },

                body: JSON.stringify(leaveData)

            });

            if (response.ok) {

                setSuccessMessage("Leave request submitted successfully!");

                setEmployeeID('');

                setFullName('');

                setStartDate('');

                setEndDate('');

                setLeaveType('Vacation');

                setReasonForLeave('');

            } else {

                const errorData = await response.json();

                setErrorMessage(errorData.message || "Failed to apply for leave.");

            }

        } catch (error) {

            setErrorMessage("An error occurred while submitting the leave request.");

        }

    };

    return (

        <div className="leave-form-container">

            <h2>Leave Application</h2>

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

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

            <form onSubmit={handleSubmit}>

                <input

                    type="text"

                    value={employeeID}

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

                    placeholder="Employee ID"

                    required

                />

                <input

                    type="text"

                    value={fullName}

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

                    placeholder="Full Name"

                    required

                />

                <label htmlFor="startDate">Start Date:</label>

                <input

                    type="date"

                    id="startDate"

                    value={startDate}

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

                    required

                />

                <label htmlFor="endDate">End Date:</label>

                <input

                    type="date"

                    id="endDate"

                    value={endDate}

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

                    required

                />

                <select

                    value={leaveType}

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

                >

                    <option value="Vacation">Vacation</option>

                    <option value="Sick Leave">Sick Leave</option>

                    <option value="Personal Leave">Personal Leave</option>

                </select>

                <textarea

                    value={reasonForLeave}

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

                    placeholder="Reason for Leave"

                    required

                />

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

            </form>

        </div>

    );

}

export default LeaveApply;

**Inbox.js**

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

import './Inbox.css';

const Inbox = () => {

    const [notifications, setNotifications] = useState([]);

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

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

    useEffect(() => {

        const fetchNotifications = async () => {

            const empId = localStorage.getItem('empId');  // Assuming empId is stored in localStorage upon login

            try {

                const response = await fetch(`http://localhost:8080/employee/${empId}/notifications`);

                if (!response.ok) {

                    throw new Error('Failed to fetch notifications');

                }

                const data = await response.json();

                setNotifications(data);

            } catch (err) {

                setError(err.message);

            } finally {

                setLoading(false);

            }

        };

  fetchNotifications();

    }, []);

    if (loading) {

        return <p>Loading notifications...</p>;

    }

    if (error) {

        return <p>Error: {error}</p>;

    }

    return (

        <div className="inbox">

            <h2>Inbox</h2>

            <div className="inbox-container">

                {notifications.length > 0 ? (

                    notifications.map((notification, index) => (

                        <div key={index} className="notification">

                            <p>{notification.message}</p>

                            <p>{new Date(notification.timestamp).toLocaleString()}</p>

                        </div>

                    ))

                ) : (

                    <p>No notifications.</p>

                )}

            </div>

        </div>

    );

};

export default Inbox;

**Profile.js**

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

import './Profile.css';

function Profile() {

    const [userData, setUserData] = useState(null);

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

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

    useEffect(() => {

        const fetchProfileData = async () => {

            const empId = localStorage.getItem('empId');

            console.log('Employee ID:', empId);

            if (!empId) {

                setError('Employee ID not found. Please log in again.');

                setLoading(false);

                return;

            }

        try {

                const response = await fetch(`http://localhost:8080/employee/profile/${empId}`);

                console.log('Response Status:', response.status); // Log response status

                if (!response.ok) {

                    throw new Error('Failed to fetch profile data');

                }

                const data = await response.json();

                console.log('Profile data:', data);

                setUserData(data);

            } catch (error) {

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

                setError(error.message);

            } finally {

                setLoading(false);

            }

        };

        fetchProfileData();

    }, []);

if (loading) {

        return <p>Loading profile...</p>;

    }

    if (error) {

        return <p>Error: {error}</p>;

    }

    if (!userData) {

        return <p>No profile data available.</p>;

    }

    return (

        <div className="profile-container">

            <h2>Employee Profile</h2>

            <div className="profile-card">

                <h3>{userData.name}</h3>

                <p><strong>Emp ID:</strong> {userData.empId}</p>

                <p><strong>Email:</strong> {userData.email}</p>

                <p><strong>Contact No:</strong> {userData.contactNo}</p>

                <p><strong>Status:</strong> {userData.status}</p>

            </div>

        </div>

    );

}

export default Profile;

**Adminpage:**

**Employeelist.js**

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

import './Employee.css';

const EmployeeList = () => {

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

    const [searchTerm, setSearchTerm] = useState('');

    const [form, setForm] = useState({

        name: '',

        gender: '',

        empId: '',

        contactNo: '',

        status: '',

    });

    const [isEditing, setIsEditing] = useState(false);

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

    useEffect(() => {

        const fetchEmployees = async () => {

            setLoading(true);

            try {

                const response = await fetch("http://localhost:8080/employee");

                if (!response.ok) throw new Error('Network response was not ok');

                const data = await response.json();

                console.log("Fetched employees:", data);

                setEmployees(data);

            } catch (error) {

                console.error("Error fetching employee data:", error);

            } finally {

                setLoading(false);

            }

        };

        fetchEmployees();

    }, []);

    const handleInputChange = (e) => {

        const { name, value } = e.target;

        setForm((prevForm) => ({

            ...prevForm,

            [name]: value,

        }));

    };

    const handleSave = async () => {

        if (!form.name || !form.gender || !form.empId || !form.contactNo || !form.status) {

            alert('Please fill in all fields');

            return;

        }

        if (!/^\d{10}$/.test(form.contactNo)) {

            alert('Contact number must be 10 digits.');

            return;

        }

        try {

            if (isEditing) {

                const response = await fetch(`http://localhost:8080/employee/${form.empId}`, {

                    method: 'PUT',

                    headers: {

                        'Content-Type': 'application/json',

                    },

                    body: JSON.stringify(form),

                });

                if (!response.ok) throw new Error('Failed to update employee');

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

                    emp.empId === form.empId ? form : emp

                );

                setEmployees(updatedEmployees);

            } else {

                const response = await fetch("http://localhost:8080/employee", {

                    method: 'POST',

                    headers: {

                        'Content-Type': 'application/json',

                    },

                    body: JSON.stringify(form),

                });

                if (!response.ok) throw new Error('Failed to add employee');

                setEmployees([...employees, form]);

            }

        } catch (error) {

            console.error("Error saving employee data:", error);

            alert("Failed to save employee. Please try again.");

        }

        handleCancel();

    };

    const handleCancel = () => {

        setForm({

            name: '',

            gender: '',

            empId: '',

            contactNo: '',

            status: '',

        });

        setIsEditing(false);

    };

    const handleDelete = (empId) => {

        if (window.confirm("Are you sure you want to delete this employee?")) {

            // Update the employees state without a backend call

            setEmployees(employees.filter(emp => emp.empId !== empId));

            console.log(`Deleted employee with ID: ${empId}`);

        }

    };

    const filteredEmployees = employees.filter((emp) =>

        emp.name.toLowerCase().includes(searchTerm.toLowerCase())

    );

    if (loading) {

        return <div>Loading...</div>;

    }

    return (

        <div className="employee-manager-container">

            <div className="content">

                <h2>Employee List</h2>

                <form className="employee-form" onSubmit={(e) => e.preventDefault()}>

                    <input

                        type="text"

                        placeholder="Search"

                        value={searchTerm}

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

                    />

                    <input

                        type="text"

                        placeholder="Name"

                        name="name"

                        value={form.name}

                        onChange={handleInputChange}

                    />

                    <input

                        type="text"

                        placeholder="Employee ID"

                        name="empId"

                        value={form.empId}

                        onChange={handleInputChange}

                    />

                    <input

                        type="text"

                        placeholder="Contact No"

                        name="contactNo"

                        value={form.contactNo}

                        onChange={handleInputChange}

                    />

                    <input

                        type="text"

                        placeholder="Status"

                        name="status"

                        value={form.status}

                        onChange={handleInputChange}

                    />

                    <div className="form-buttons">

                        <button className="save-button" type="button" onClick={handleSave}>

                            {isEditing ? 'Update' : 'Save'}

                        </button>

                        <button className="cancel-button" type="button" onClick={handleCancel}>

                            Cancel

                        </button>

                    </div>

                </form>

                <table className="employee-table">

                    <thead>

                        <tr>

                            <th>ID</th>

                            <th>Name</th>

                            <th>Contact No</th>

                            <th>Status</th>

                            <th>Action</th>

                        </tr>

                    </thead>

                    <tbody>

                        {filteredEmployees.map((employee) => (

                            <tr key={employee.empId}>

                                <td>{employee.empId}</td>

                                <td>{employee.name}</td>

                                <td>{employee.contactNo}</td>

                                <td>{employee.status}</td>

                                <td>

                                    <button onClick={() => handleDelete(employee.empId)}>Delete</button>

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

};

export default EmployeeList;

**AdminInbox.js**

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

import { useNotification } from '../NotificationContext';

import './AdmInbox.css';

const AdminInbox = () => {

    const { fetchNotifications } = useNotification();

    const [leaveRequests, setLeaveRequests] = useState([]);

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

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

    const [loadingApprove, setLoadingApprove] = useState(false);

    const [loadingReject, setLoadingReject] = useState(false);

    useEffect(() => {

        const fetchLeaveRequests = async () => {

            setLoading(true);

            try {

                const response = await fetch('http://localhost:8080/leaves');

                if (!response.ok) {

                    throw new Error('Failed to fetch leave requests');

                }

                const data = await response.json();

                setLeaveRequests(data);

            } catch (err) {

                setError(err.message);

            } finally {

                setLoading(false);

            }

        };

        fetchLeaveRequests();

    }, []);

    const sendNotification = async (empId, message) => {

        try {

            const response = await fetch(`http://localhost:8080/notify/${empId}`, {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ message }),

            });

            if (!response.ok) {

                throw new Error('Failed to send notification');

            }

            console.log(`Notification sent to employee ${empId}: ${message}`); // Debug logging

        } catch (error) {

            console.error('Error sending notification:', error);

        }

    };

    const handleLeaveAction = async (empId, startDate, action) => {

        const actionVerb = action === 'approve' ? 'approved' : 'rejected';

        const endpoint = `http://localhost:8080/leave/${empId}/${action}?startDate=${startDate}`;

        const message = `Your leave has been ${actionVerb}.`;

        setLeaveRequests((prevRequests) =>

            prevRequests.filter(

                (leave) => !(leave.empId === empId && leave.startDate === startDate)

            )

        );

        try {

            const response = await fetch(endpoint, { method: 'PUT' });

            if (!response.ok) {

                throw new Error(`Failed to ${action} leave`);

            }

            await sendNotification(empId, message);

            alert(message);

            fetchNotifications();

        } catch (error) {

            console.error(`Error ${action} leave:`, error);

            alert(`Failed to ${action} leave`);

            setLeaveRequests((prevRequests) => [

                ...prevRequests,

                { empId, startDate } // You may consider adding more details if necessary

            ]);

        }

    };

    const handleApprove = async (empId, startDate) => {

        setLoadingApprove(true);

        await handleLeaveAction(empId, startDate, 'approve');

        setLoadingApprove(false);

    };

    const handleReject = async (empId, startDate) => {

        const confirmation = window.confirm('Are you sure you want to reject this leave request?');

        if (confirmation) {

            setLoadingReject(true);

            await handleLeaveAction(empId, startDate, 'reject');

            setLoadingReject(false); // Set loadingReject to false after operation

        }

    };

    if (loading) {

        return <p>Loading leave requests...</p>;

    }

    if (error) {

        return <p>Error: {error}</p>;

    }

    return (

        <div className="admin-inbox">

            <h2>Admin Inbox</h2>

            <div>

                {leaveRequests.length > 0 ? (

                    leaveRequests.map((leave, index) => (

                        <div key={index} className="request-card">

                            <p>

                                Leave request from {leave.empId}: {leave.type} from {leave.startDate} to {leave.endDate}

                            </p>

                            <button

                                className="approve-button"

                                onClick={() => handleApprove(leave.empId, leave.startDate)}

                                disabled={loading || loadingApprove}

                            >

                                {loadingApprove ? 'Approving...' : 'Approve'}

                            </button>

                            <button

                                className="reject-button"

                                onClick={() => handleReject(leave.empId, leave.startDate)}

                                disabled={loading || loadingReject}

                            >

                                {loadingReject ? 'Rejecting...' : 'Reject'}

                            </button>

                        </div>

                    ))

                ) : (

                    <p>No leave requests.</p>

                )}

            </div>

        </div>

    );

};

export default AdminInbox;

**NotificationContext.js**

import React, { createContext, useContext, useState, useEffect } from 'react';

const NotificationContext = createContext();

export const useNotification = () => {

    return useContext(NotificationContext);

};

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

    const [notifications, setNotifications] = useState([]);

    const fetchNotifications = async () => {

        try {

            const response = await fetch('http://localhost:8080/notifications');

            const data = await response.json();

            setNotifications(data);

        } catch (error) {

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

        }

    };

    const addNotification = (notification) => {

        setNotifications((prevNotifications) => [...prevNotifications, notification]);

    };

    useEffect(() => {

        fetchNotifications();

    }, []);

    return (

        <NotificationContext.Provider value={{ notifications, addNotification, fetchNotifications }}>

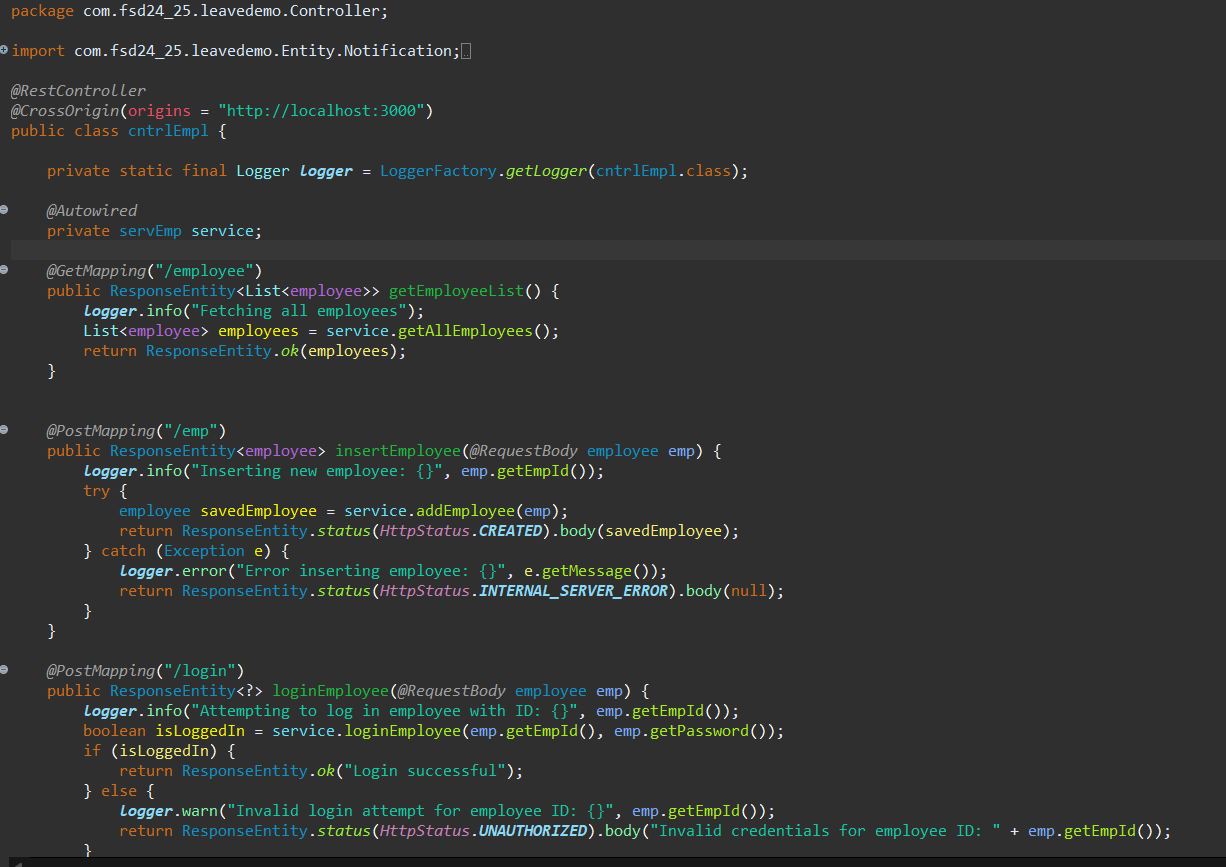
            {children}

        </NotificationContext.Provider>

    );

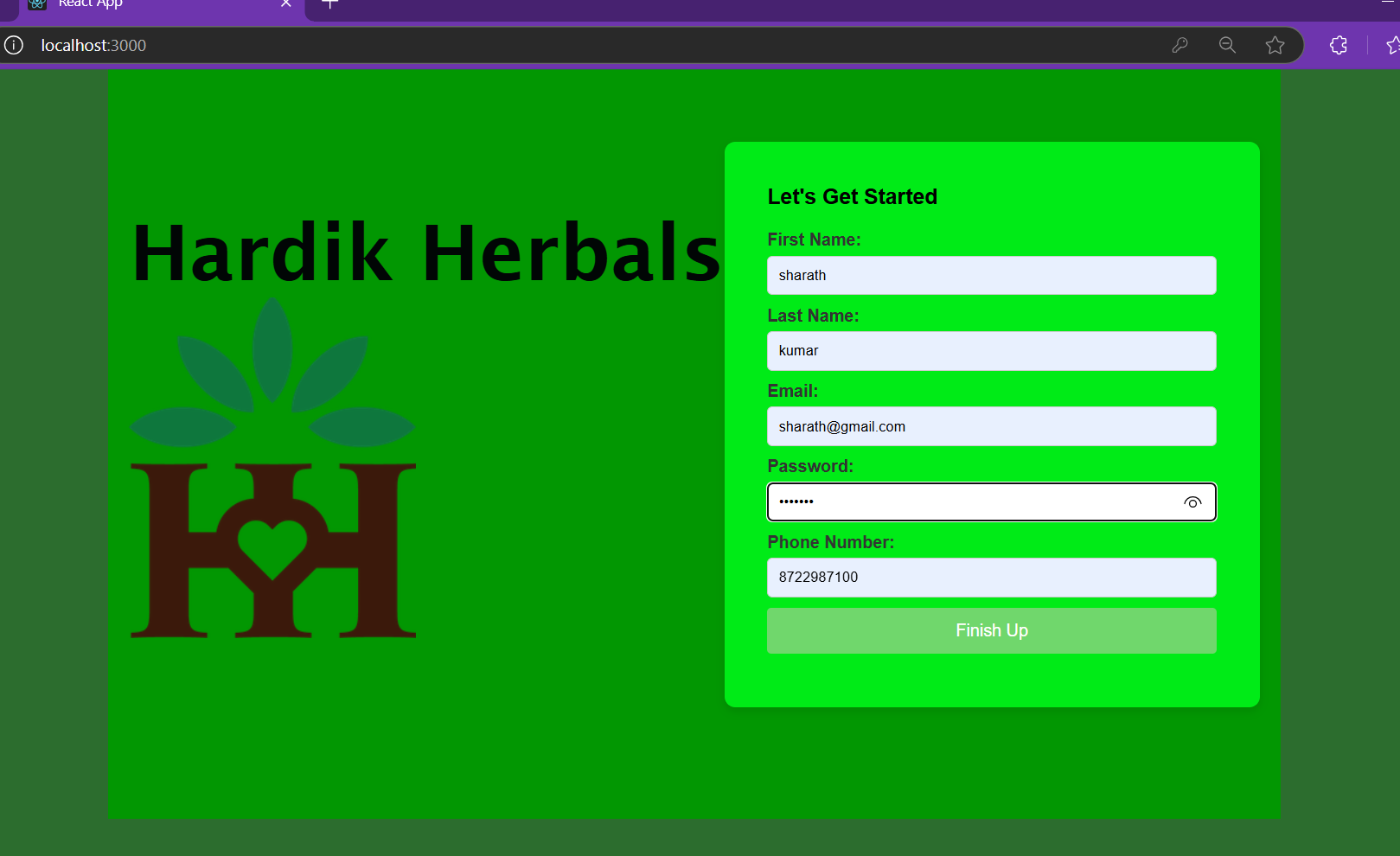
};

**Backendcode**

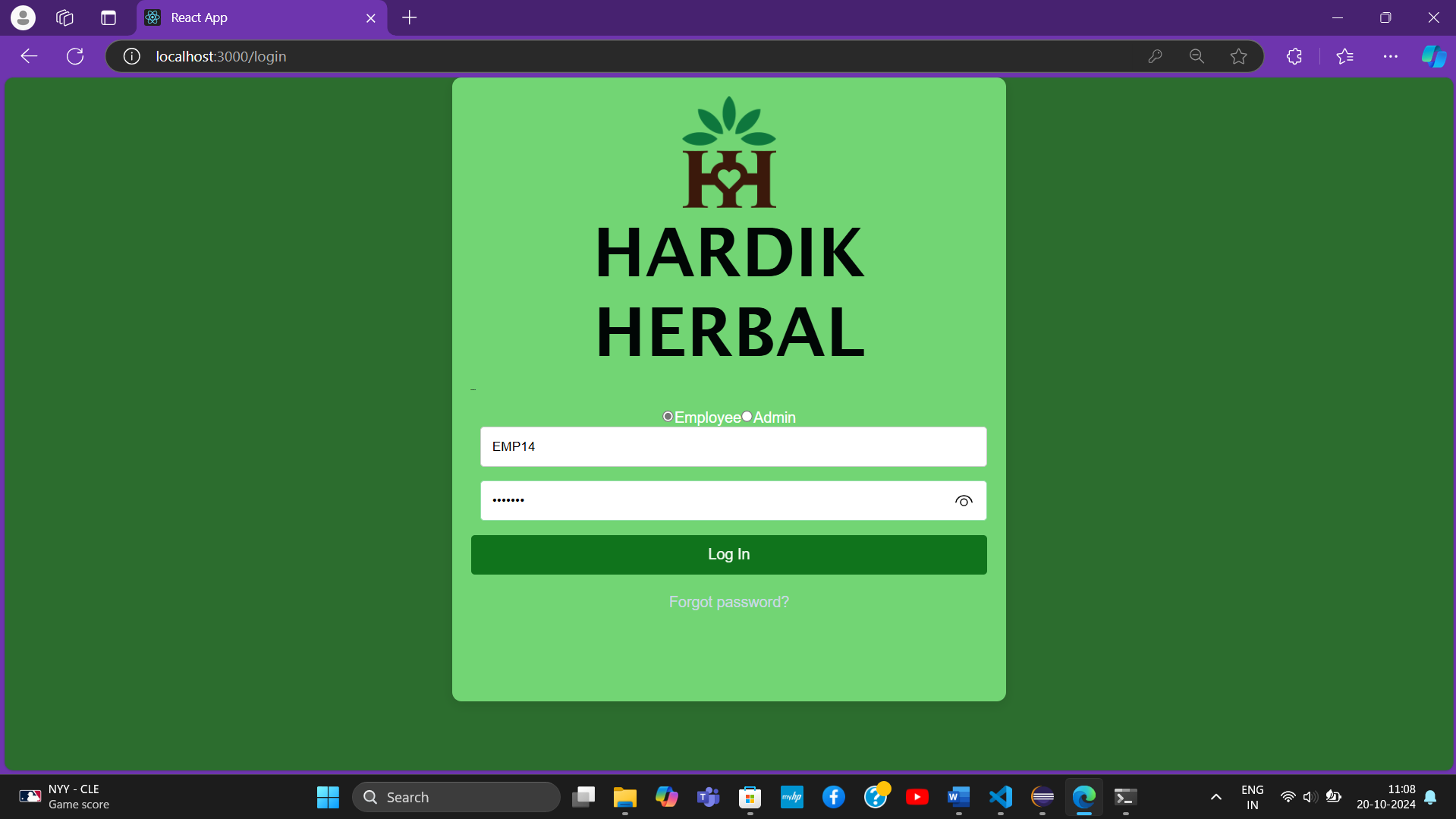
****

**Output:**

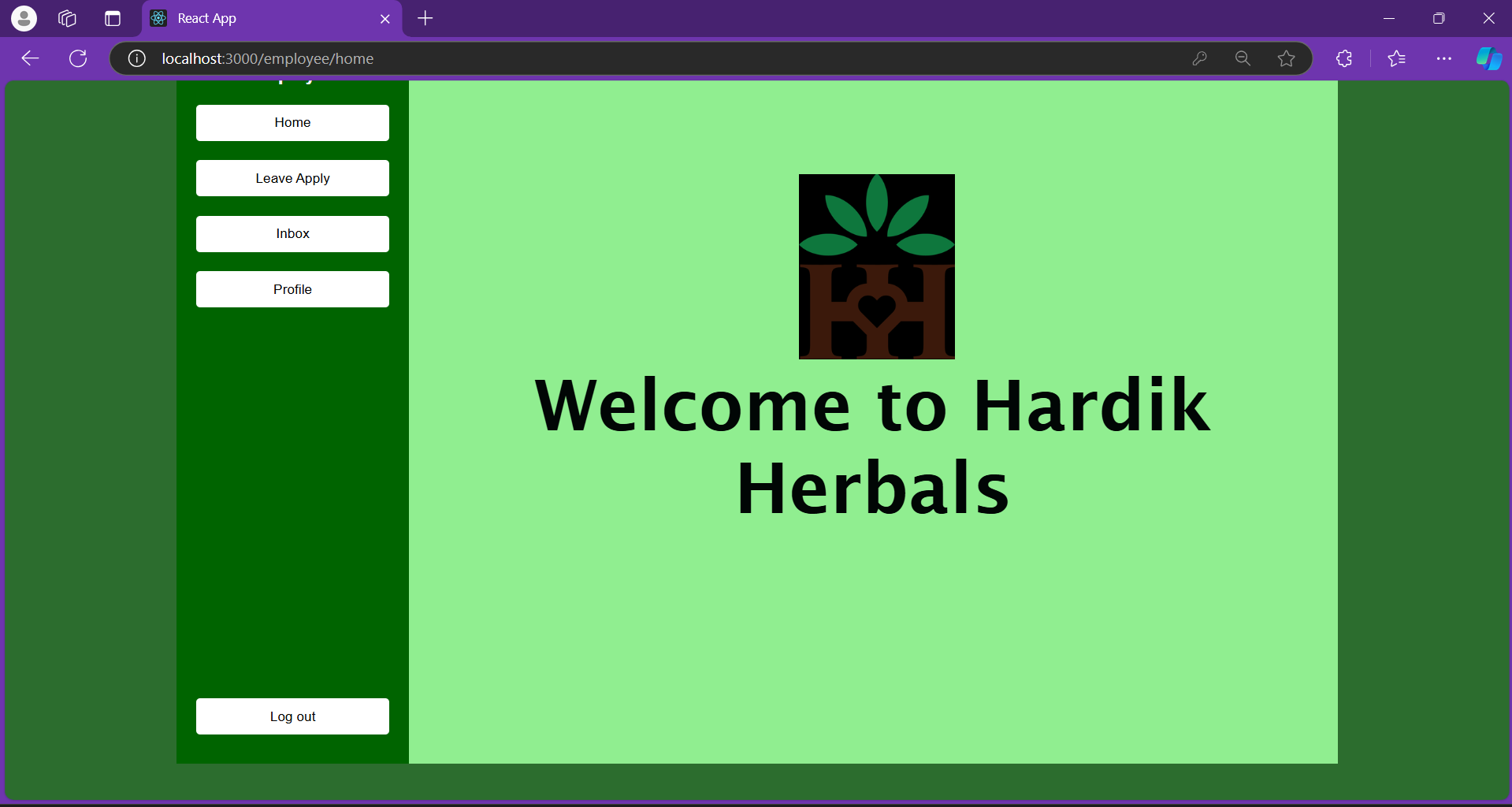
* **Run eclipse backend🡪Run the front end**
* **Then employee register his details here🡪then click on finishup.**

****

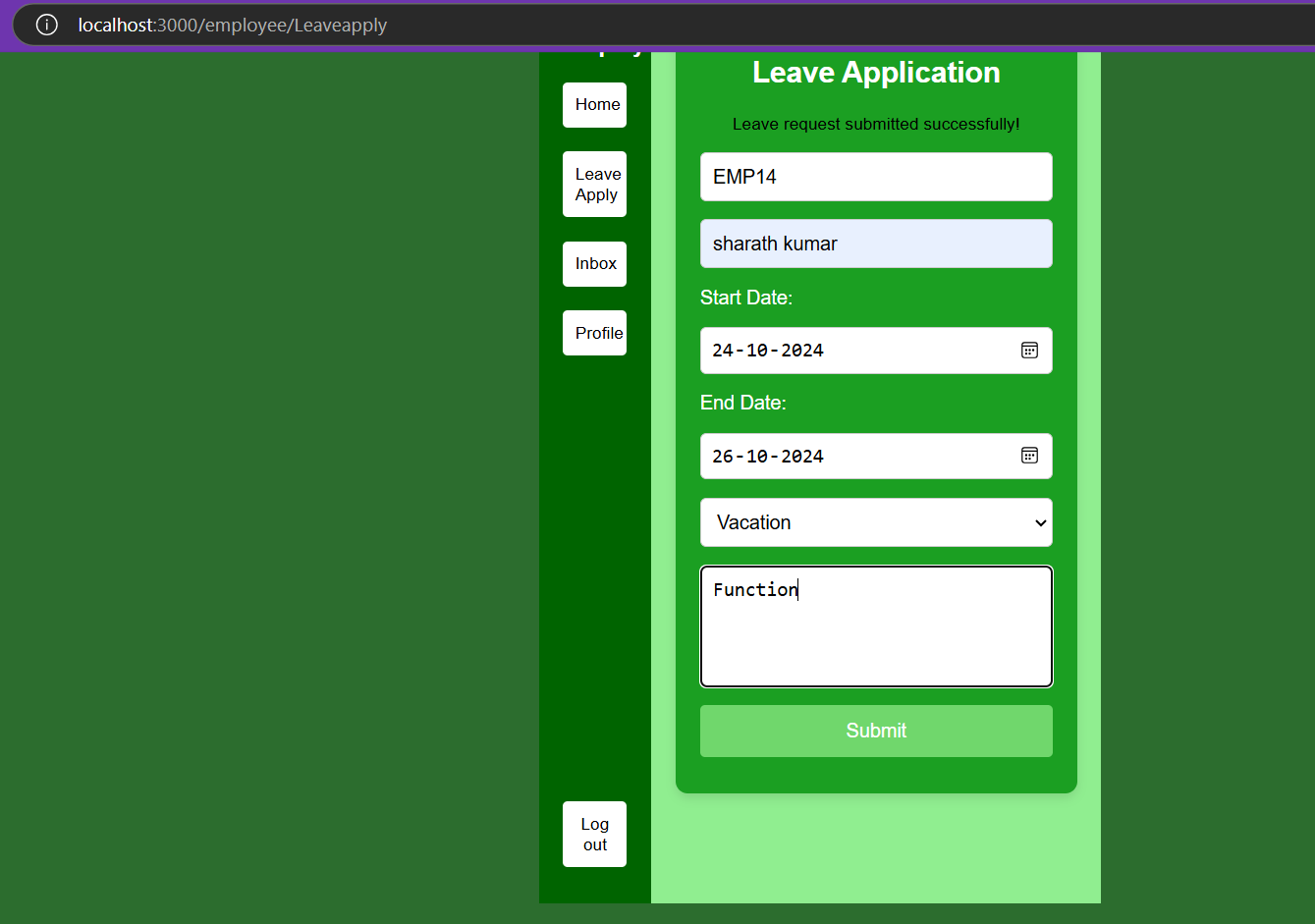
**Login page:🡪Employee login 🡪Then click on login.**

****

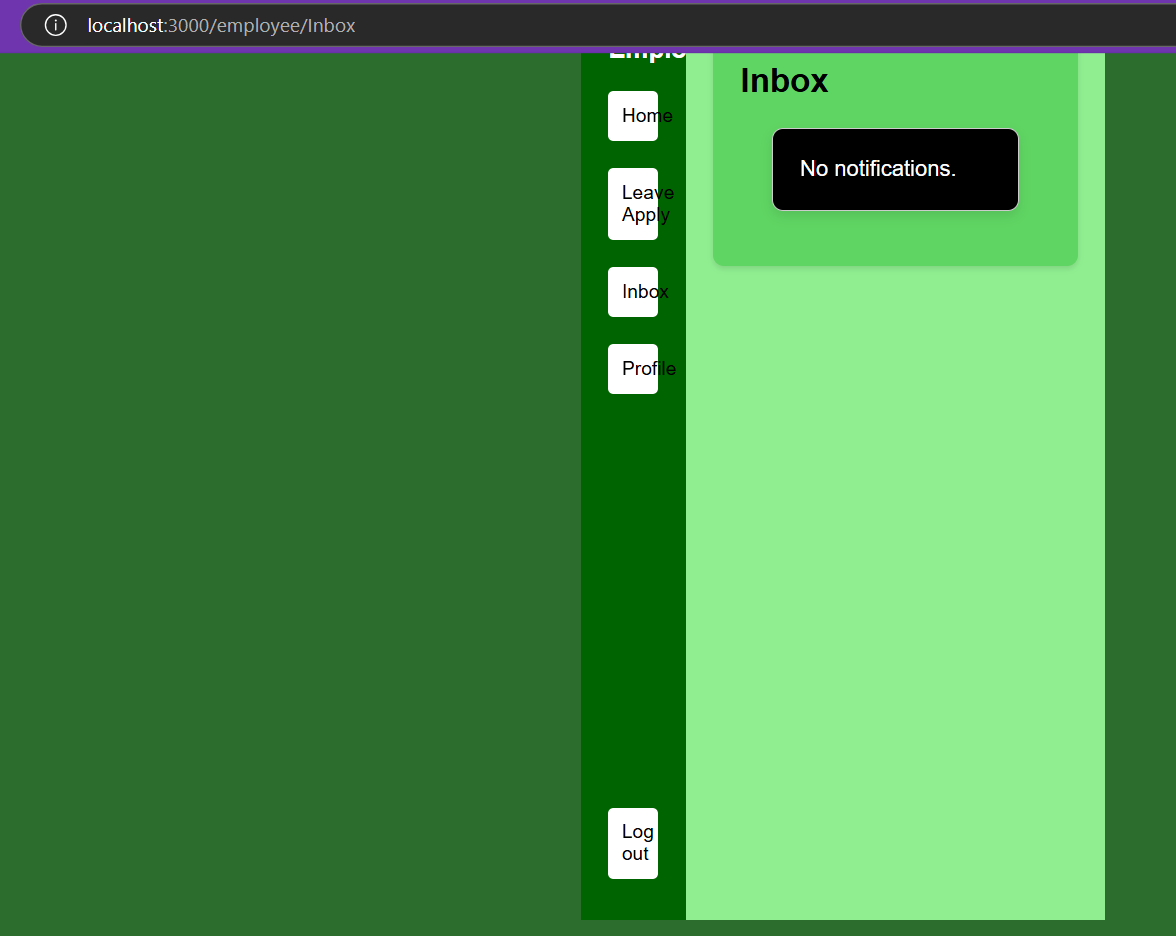
**EmployeeHome page will Display.**

****

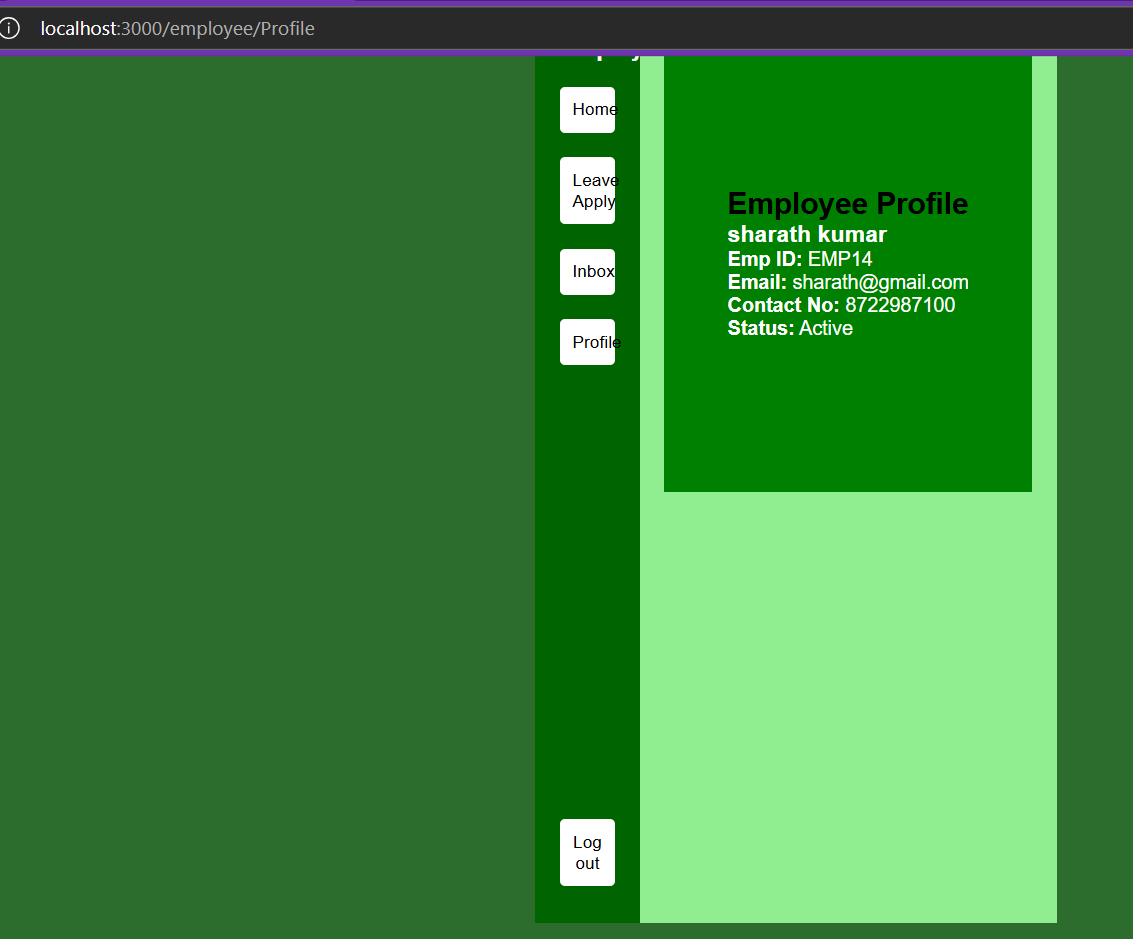
**Leaveapply🡪Employee apply leavea application to admin🡪then click on submit.**

****

**Employee Inbox.**

****

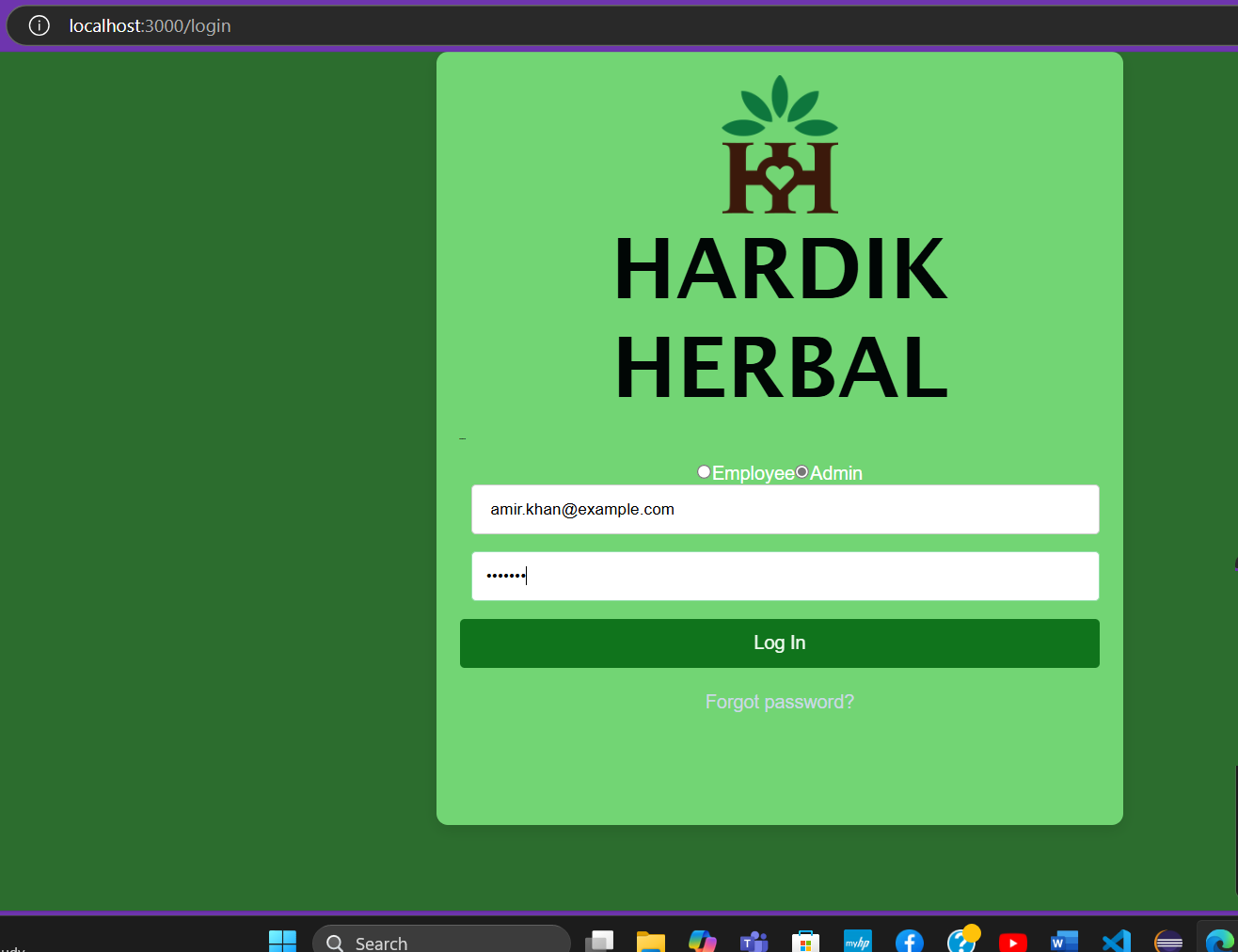
**Click on Profile 🡪display his profile.**

****

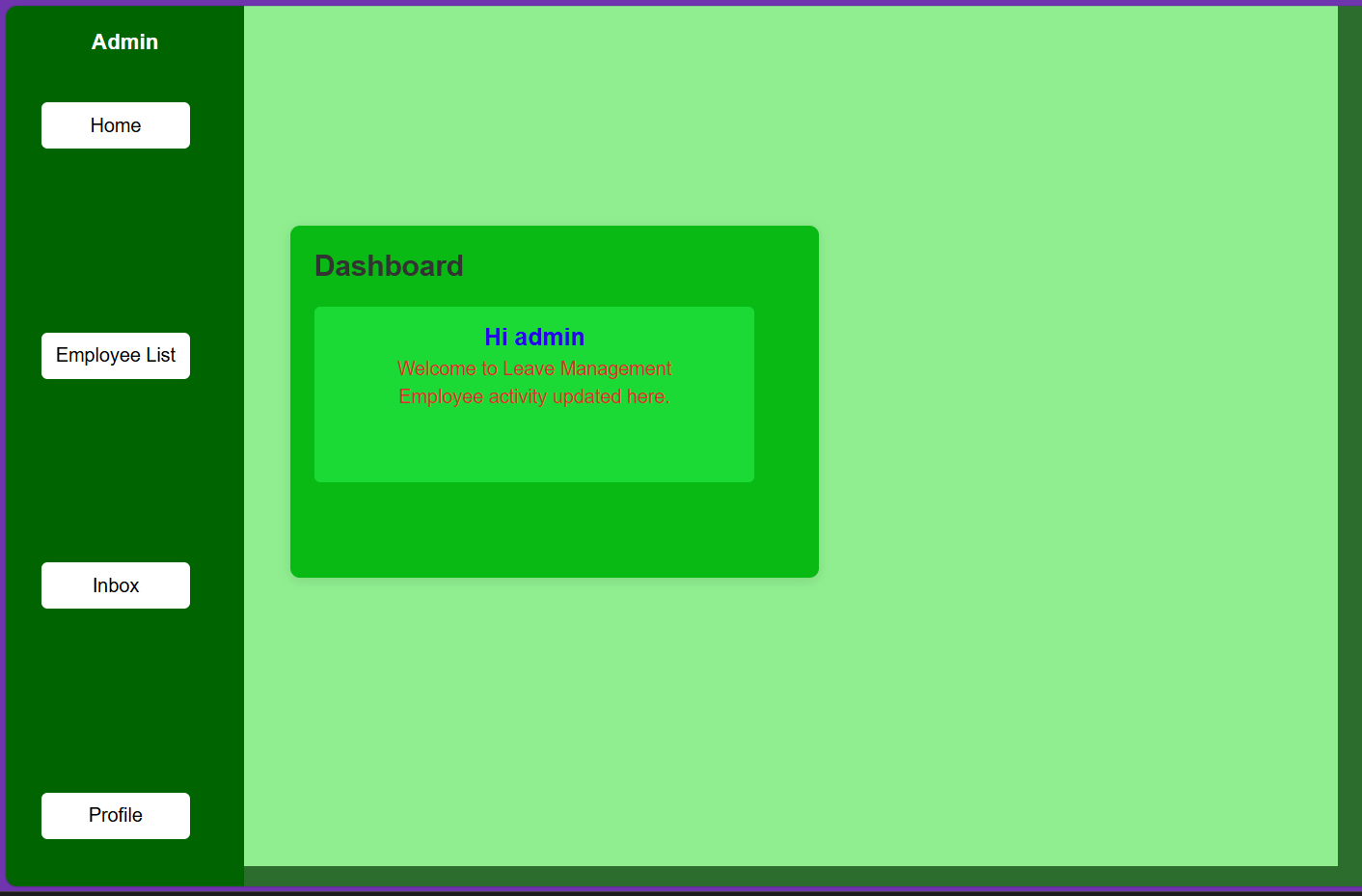
**Click on logout 🡪 then go login page.**

**Click on Admin**

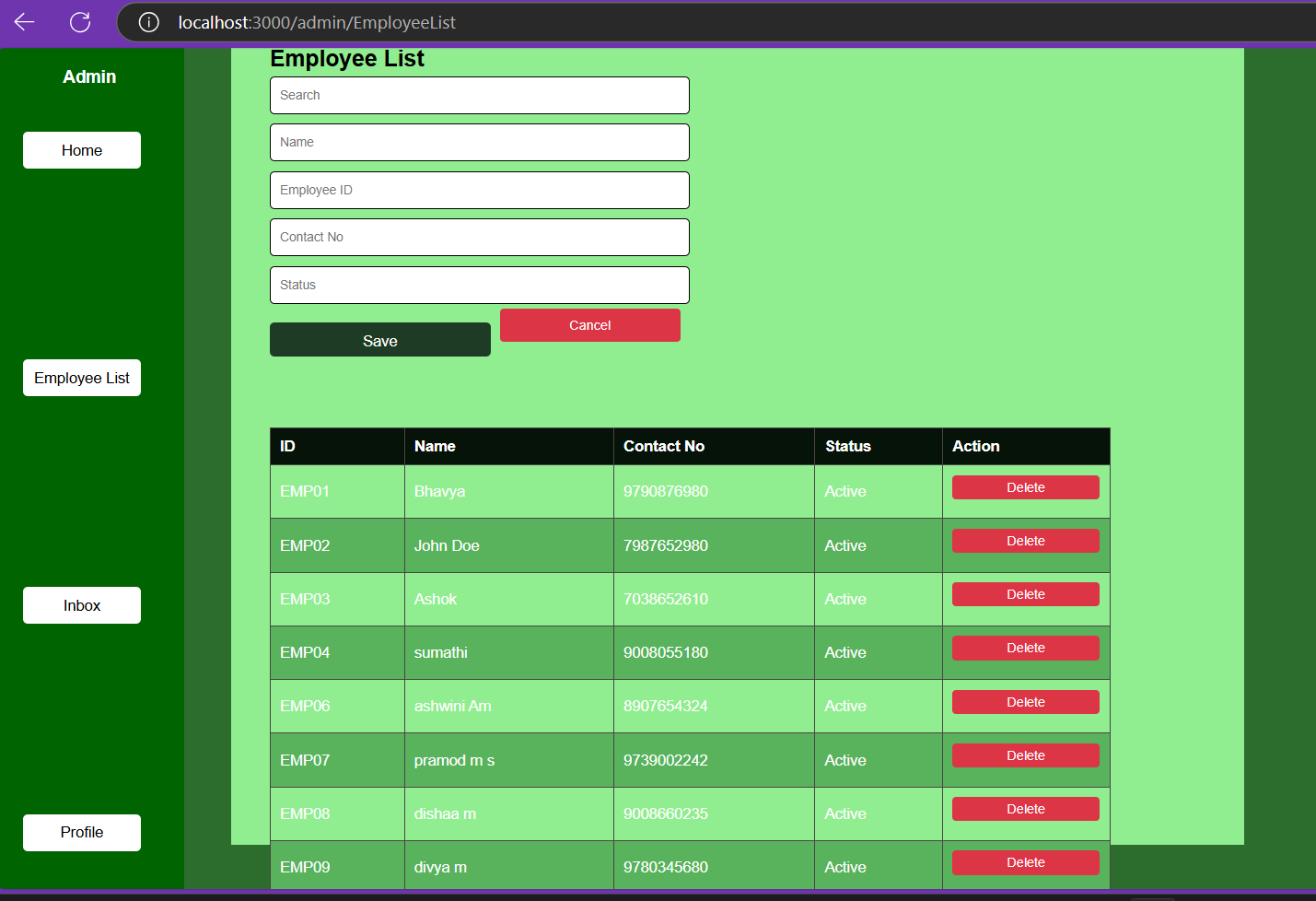
**Admin login:**

****

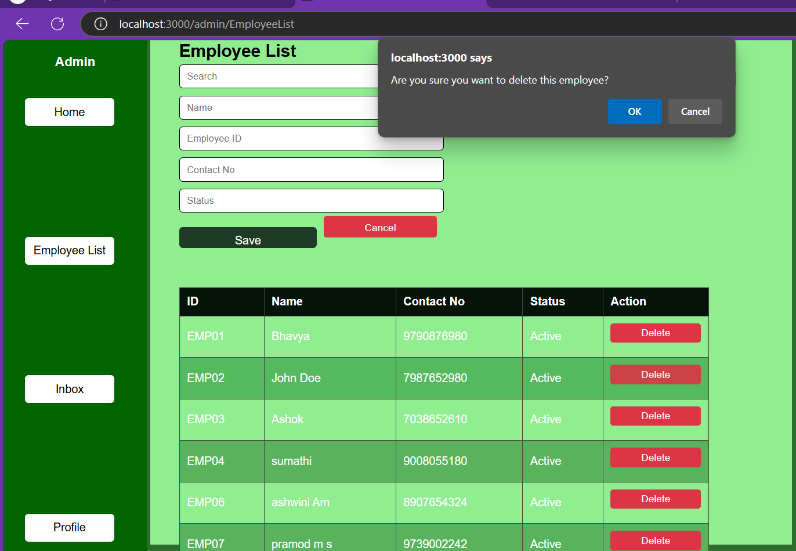
**AdminHomepage**

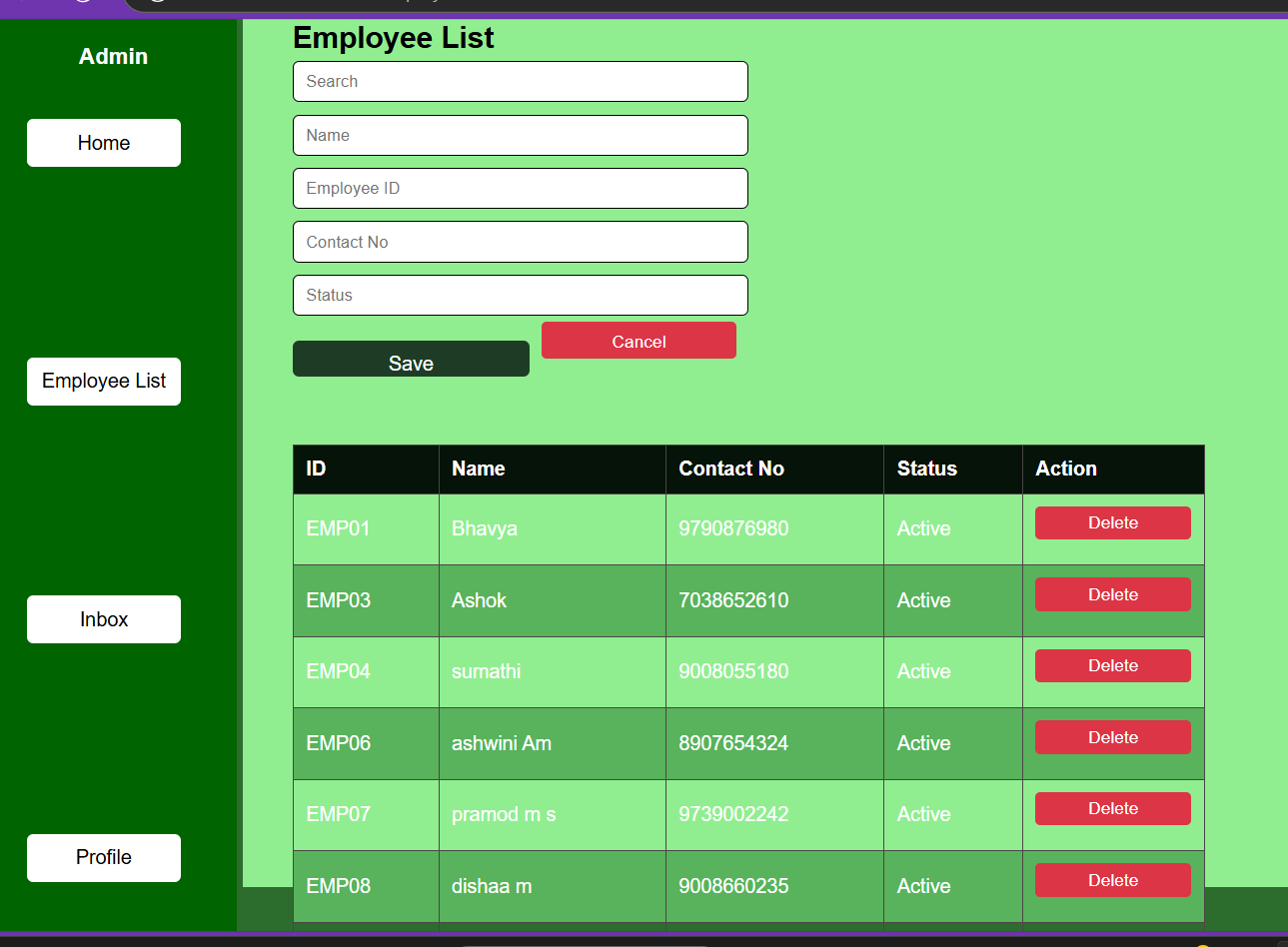
****

**EmployeeList**

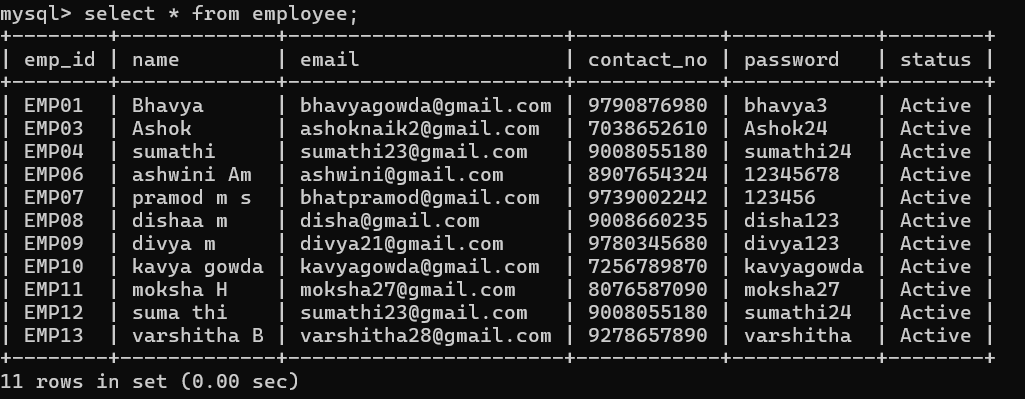
****

**When Click on delete button 🡪Employee will delete from the table.**

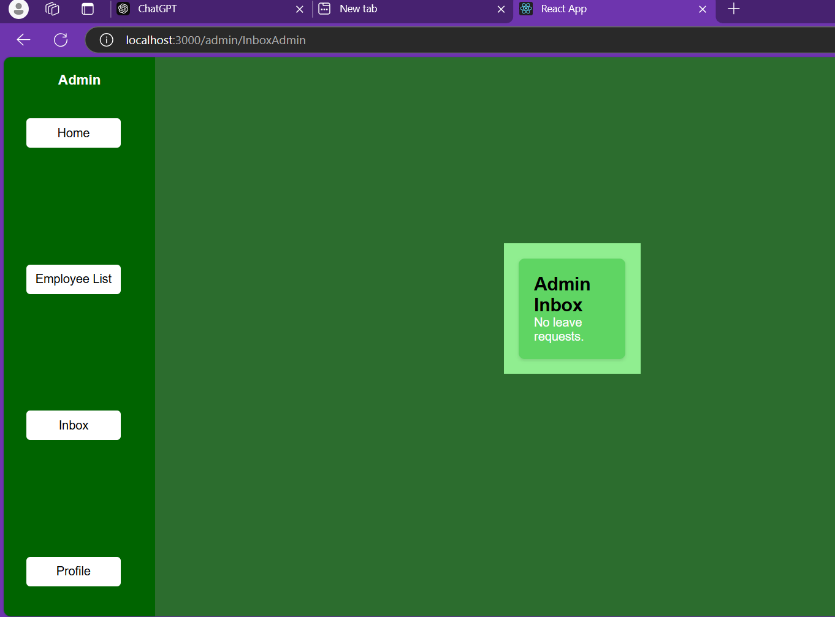
****

****

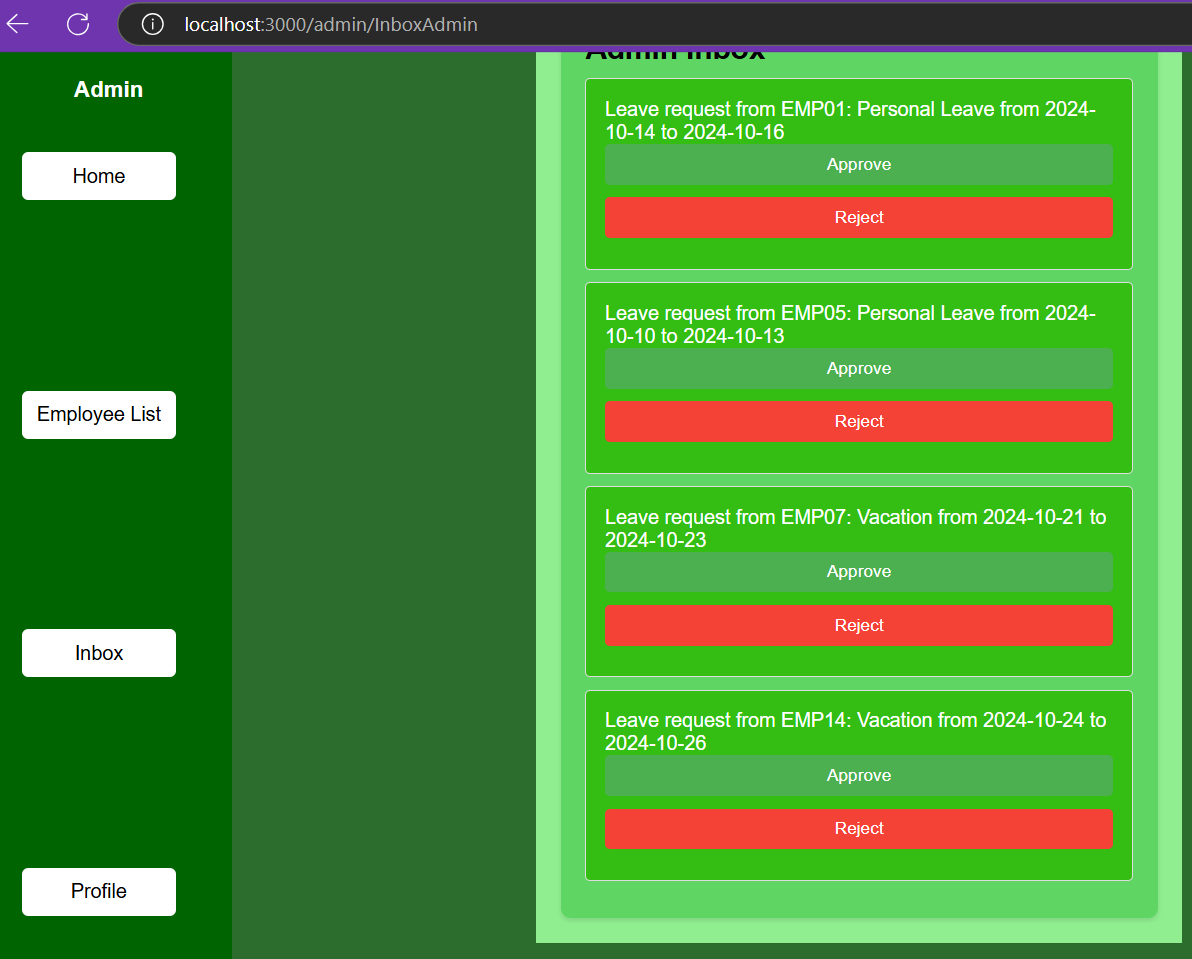
**In Mysql:**

****

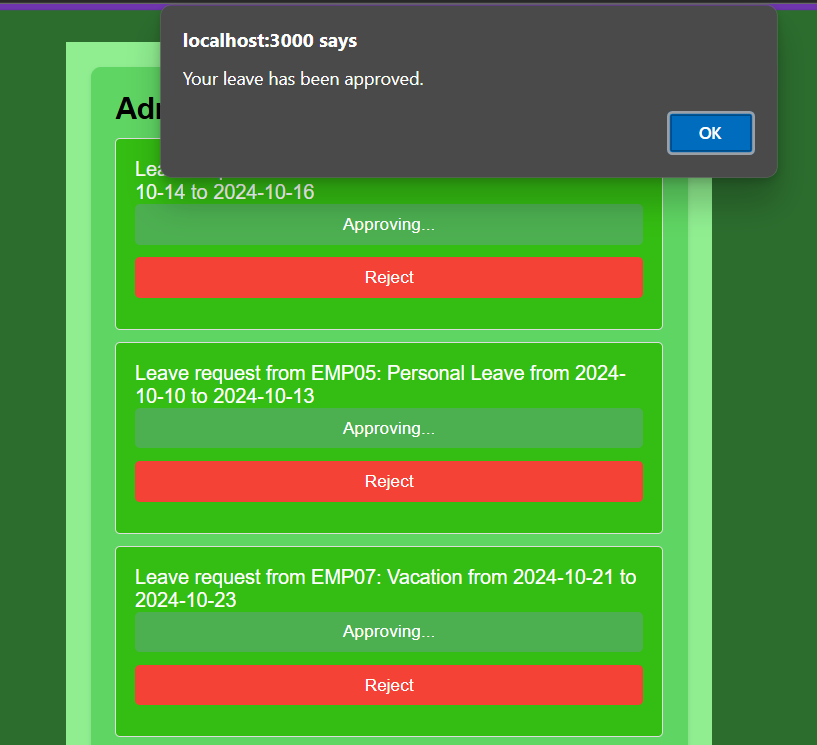
**AdminInbox.**



**Employee Send leaveapplication.**

****

**When Admin Click on Approve.**

****

**Then Employee can check his inbox .**



In backend all are updated

