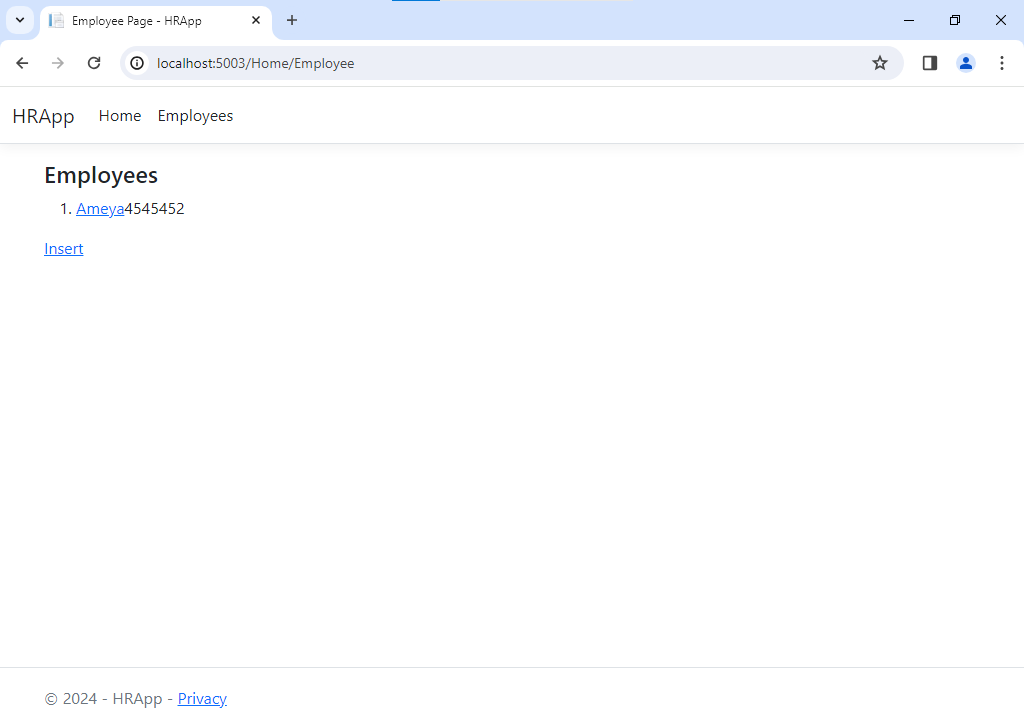
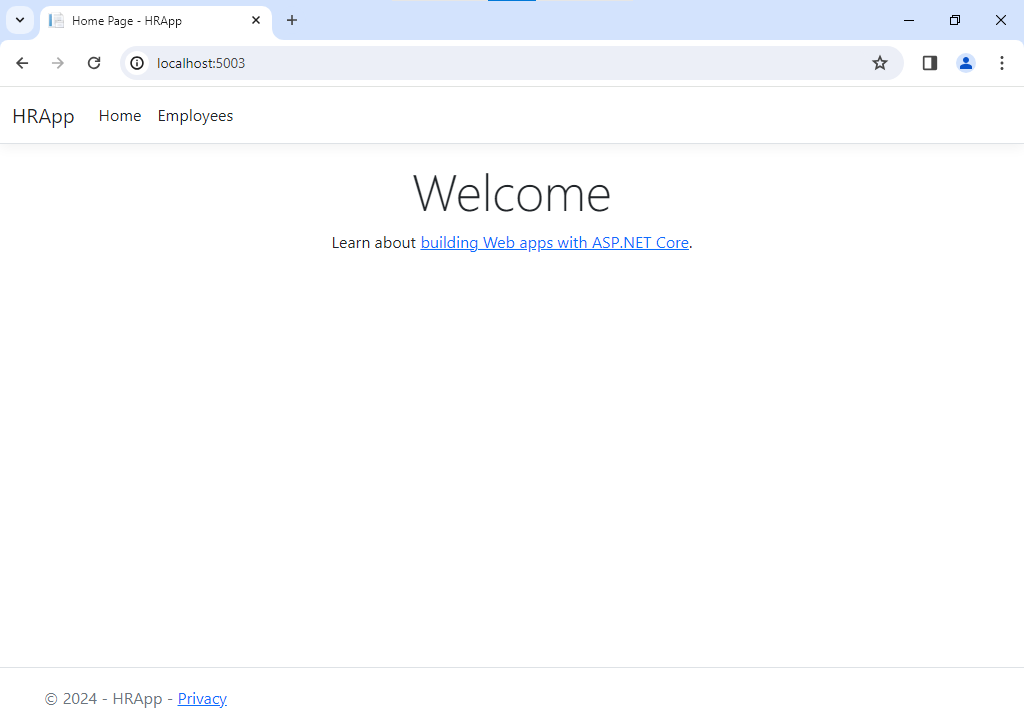
**Name: Ameya Kulkarni**

**PRN: 230945920050**

**Employee**



Code

@using BOL;

@{

    ViewData["Title"] = "Employee Page";

}

<div>

    <div>

        <h4>Employees</h4>

    </div>

    <ol>

        @{

            List<Employee> elist = ViewData["employees"] as List<Employee>;

            @foreach (Employee e in elist)

            {

                <li><a href="Details/@e.Id">@e.Name</a>@e.ContactNumber</li>

            }

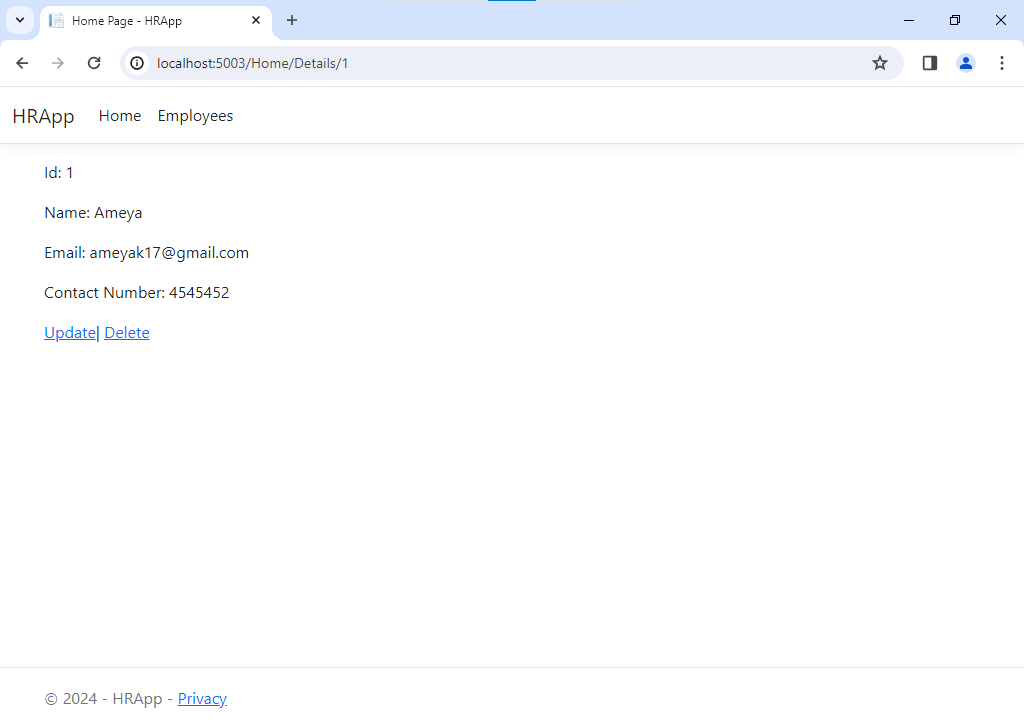
        }

    </ol>

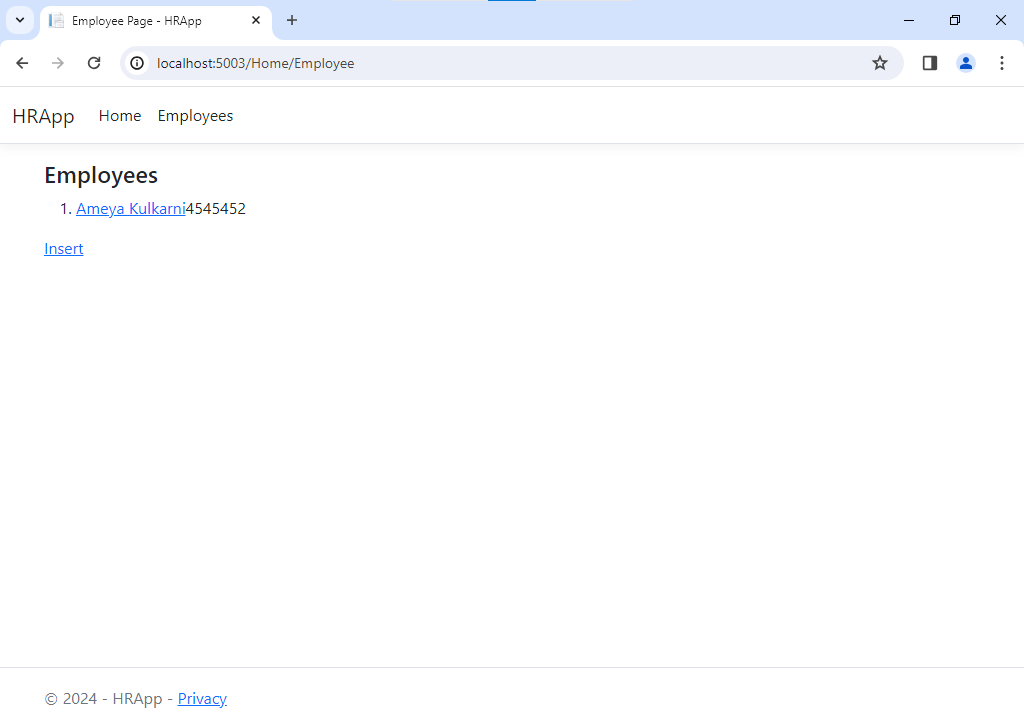
    <a href="Insert">Insert</a>

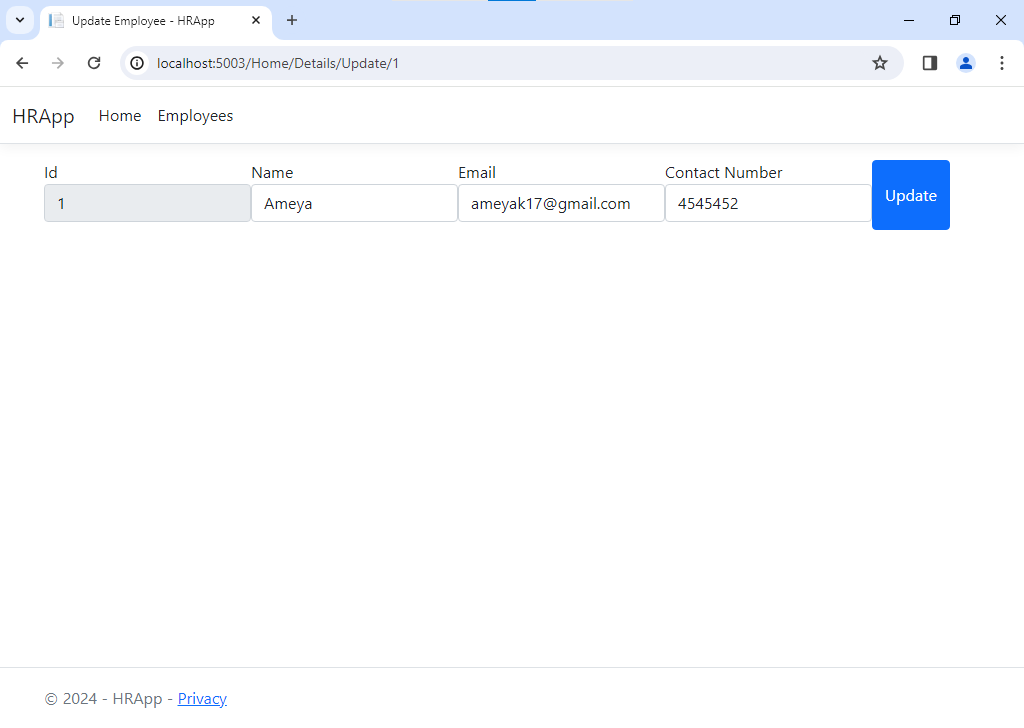
</div>

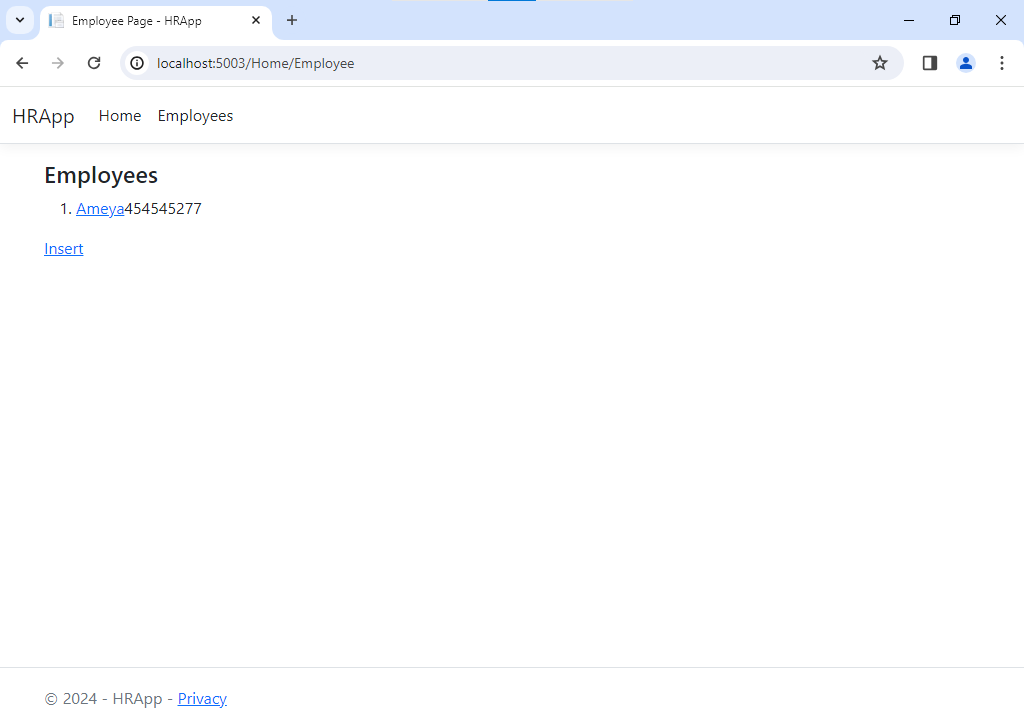
**Details**

****

Update







View:

@{

    ViewData["Title"] = "Update Employee";

}

<div>

    <form method="post" style="display: flex; align-items: 'center';">

        @{

            <label class="form-label">

                Id<input type=" number" name="Id" value="@Model.Id" class="form-control" readonly>

            </label>

            <label class="form-label">

                Name<input type="text" name="Name" value="@Model.Name" class="form-control">

            </label>

            <label class="form-label">

                Email<input type="email" name="Email" value="@Model.Email" class="form-control">

            </label>

            <label class="form-label">

                Contact Number<input type="text" name="contactNumber" value="@Model.ContactNumber" class="form-control">

            </label>

            <input type="submit" value="Update" class="btn btn-primary">

        }

    </form>

</div>

Controller:

[Route("Home/Details/Update/{id}")]

    public IActionResult Update(int id)

    {

        List<Employee> elist = EmployeeManager.GetAll();

        Employee e = elist.Find((emp) => emp.Id == id);

        return View(e);

    }

    [Route("Home/Details/Update/{id}")]

    [HttpPost]

    public IActionResult Update(int id, string name, string email, string contactNumber)

    {

        bool status = DBManager.Update(id, name, email, contactNumber);

        if (status)

        {

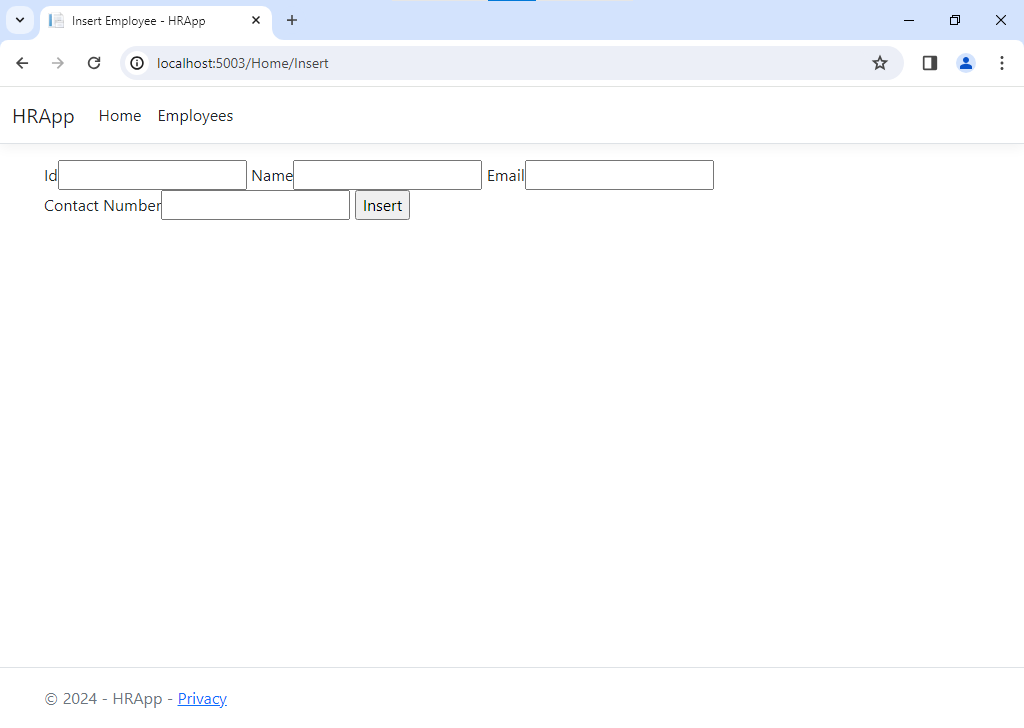
            return this.RedirectToAction("Employee");

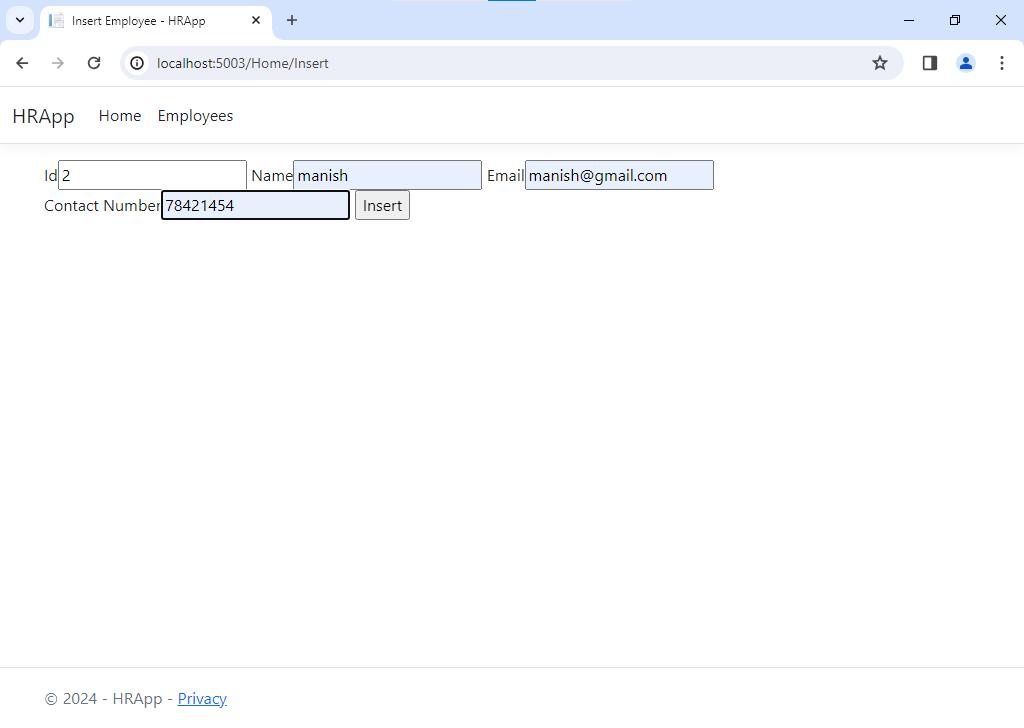
        }

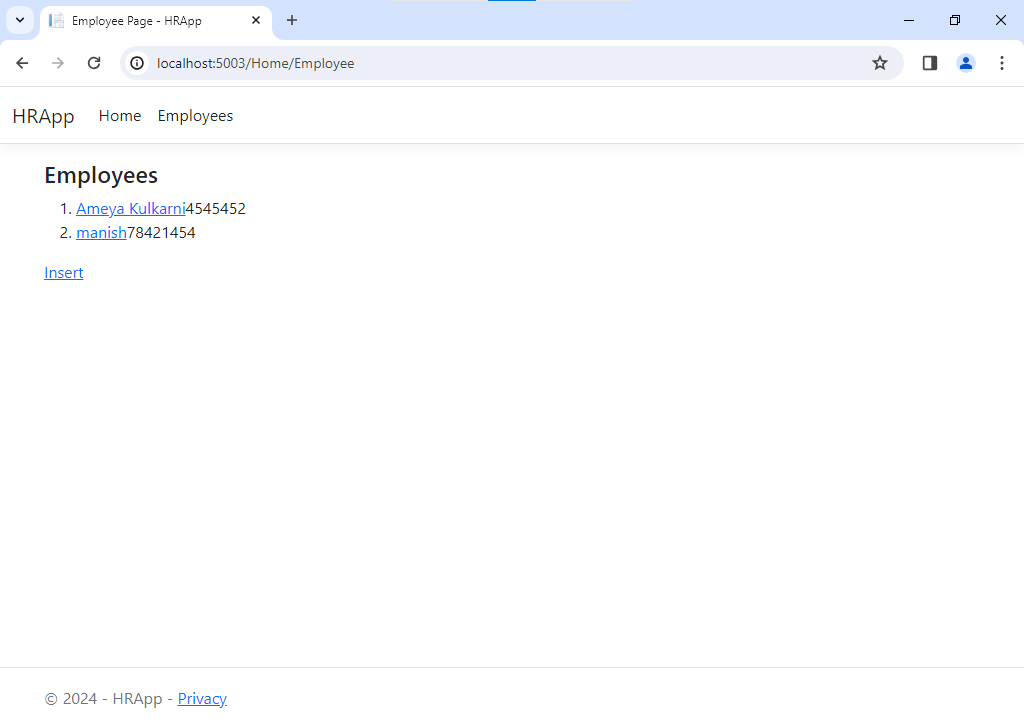
        return View();

    }

Insert:





Output

Controller:

public IActionResult Insert()

    {

        return View();

    }

    [HttpPost]

    public IActionResult Insert(int id, string name, string email, string contactNumber)

    {

        bool status = EmployeeManager.Insert(id, name, email, contactNumber);

        if (status)

        {

            return this.RedirectToAction("Employee");

        }

        return View();

    }

View:

@{

    ViewData["Title"] = "Insert Employee";

}

<div>

    <form method="post">

        <label>Id<input type="number" name="Id"></label>

        <label>Name<input type="text" name="Name"></label>

        <label>Email<input type="text" name="Email"></label>

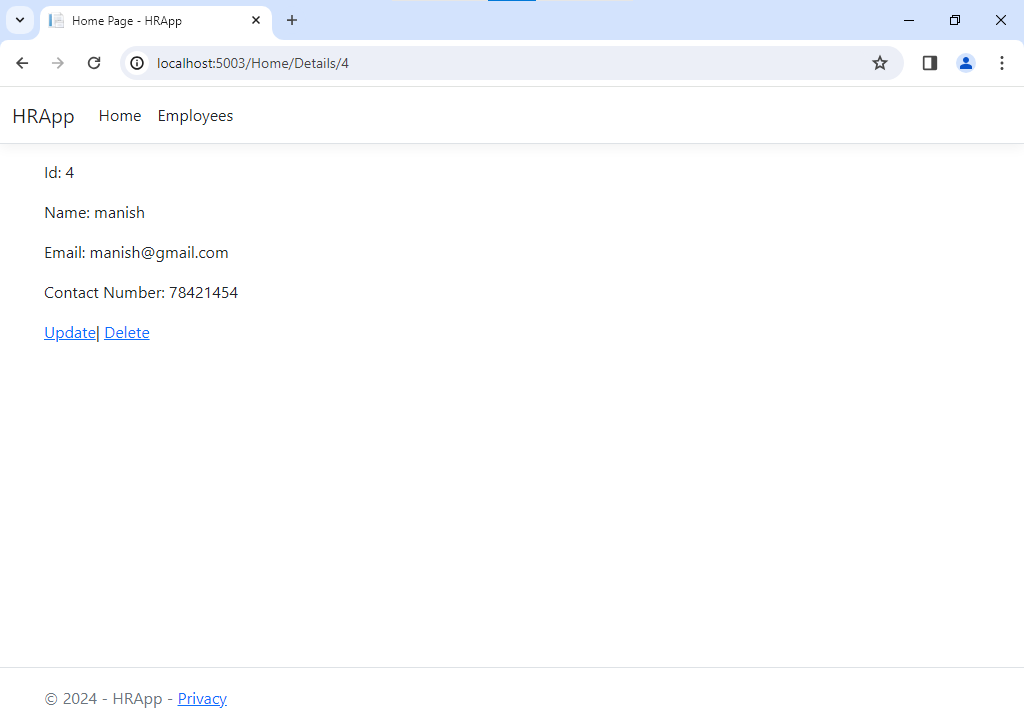
        <label>Contact Number<input type="text" name="contactNumber"></label>

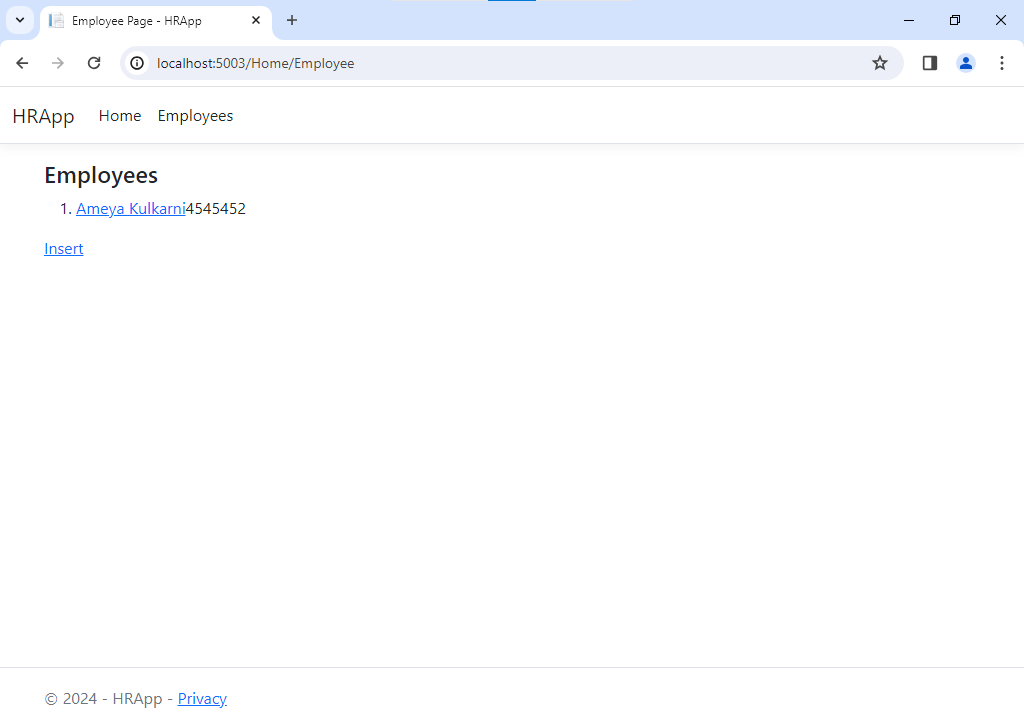
        <input type="submit" value="Insert">

    </form>

</div>

Delete:





Code:

View:

[Route("Home/Details/Delete/{id}")]

    public IActionResult Delete(int id)

    {

        bool status = EmployeeManager.Delete(id);

        if (status)

        {

            return this.RedirectToAction("Employee");

        }

        return View();

    }

Controller:

[Route("Home/Details/Delete/{id}")]

    public IActionResult Delete(int id)

    {

        bool status = EmployeeManager.Delete(id);

        if (status)

        {

            return this.RedirectToAction("Employee");

        }

        return View();

    }

DAL:

namespace DAL;

using BOL;

using MySql.Data.MySqlClient;

public class DBManager

{

    public static string conString = "server=192.168.10.150;port=3306;user=dac49;password=welcome;database=dac49";

    public static List<Employee> GetAll()

    {

        List<Employee> elist = new List<Employee>();

        MySqlConnection con = new MySqlConnection();

        con.ConnectionString = conString;

        string query = "select \* from employee1";

        MySqlCommand cmd = new MySqlCommand(query, con);

        try

        {

            con.Open();

            MySqlDataReader reader = cmd.ExecuteReader();

            while (reader.Read())

            {

                int id = int.Parse(reader["id"].ToString());

                string? name = reader["name"].ToString();

                string? email = reader["email"].ToString();

                string? contactNumber = reader["contactnumber"].ToString();

                Employee e = new Employee(id, name, email, contactNumber);

                elist.Add(e);

            }

        }

        catch (Exception e)

        {

            Console.WriteLine(e.Message);

        }

        finally

        {

            con.Close();

        }

        return elist;

    }

    public static bool Update(int id, string name, string email, string contactNumber)

    {

        MySqlConnection con = new MySqlConnection();

        con.ConnectionString = conString;

        string query = "update employee1 set name=@name, email=@email, contactnumber=@contactNumber where id =@id";

        MySqlCommand cmd = new MySqlCommand(query, con);

        try

        {

            con.Open();

            cmd.Parameters.AddWithValue("@id", id);

            cmd.Parameters.AddWithValue("@name", name);

            cmd.Parameters.AddWithValue("@email", email);

            cmd.Parameters.AddWithValue("@contactnumber", contactNumber);

            int n = cmd.ExecuteNonQuery();

            if (n > 0)

            {

                return true;

            }

        }

        catch (Exception e)

        {

            Console.WriteLine(e.Message);

        }

        finally

        {

            con.Close();

        }

        return false;

    }

    public static bool Insert(int id, string name, string email, string contactNumber)

    {

        MySqlConnection con = new MySqlConnection();

        con.ConnectionString = conString;

        string query = "insert into employee1 values(@id,@name,@email,@contactNumber)";

        MySqlCommand cmd = new MySqlCommand(query, con);

        try

        {

            con.Open();

            cmd.Parameters.AddWithValue("@id", id);

            cmd.Parameters.AddWithValue("@name", name);

            cmd.Parameters.AddWithValue("@email", email);

            cmd.Parameters.AddWithValue("@contactnumber", contactNumber);

            int n = cmd.ExecuteNonQuery();

            if (n > 0)

            {

                return true;

            }

        }

        catch (Exception e)

        {

            Console.WriteLine(e.Message);

        }

        finally

        {

            con.Close();

        }

        return false;

    }

    public static bool Delete(int id)

    {

        MySqlConnection con = new MySqlConnection();

        con.ConnectionString = conString;

        string query = "delete from employee1 where id=@id";

        MySqlCommand cmd = new MySqlCommand(query, con);

        try

        {

            con.Open();

            cmd.Parameters.AddWithValue("@id", id);

            int n = cmd.ExecuteNonQuery();

            if (n > 0)

            {

                return true;

            }

        }

        catch (Exception e)

        {

            Console.WriteLine(e.Message);

        }

        finally

        {

            con.Close();

        }

        return false;

    }

}

BOL:

Code:

namespace BOL;

public class Employee

{

    public int Id { get; set; }

    public string Name { get; set; }

    public string Email { get; set; }

    public string ContactNumber { get; set; }

    public Employee(int id, string name, string email, string contactNumber)

    {

        this.Id = id;

        this.Name = name;

        this.Email = email;

        this.ContactNumber = contactNumber;

    }

}

BLL:

Code:

namespace BLL;

using BOL;

using DAL;

public static class EmployeeManager

{

    public static List<Employee> GetAll()

    {

        List<Employee> elist = DBManager.GetAll();

        return elist;

    }

    public static bool Update(int id, string name, string email, string contactNumber)

    {

        return DBManager.Update(id, name, email, contactNumber);

    }

    public static bool Insert(int id, string name, string email, string contactNumber)

    {

        return DBManager.Insert(id, name, email, contactNumber);

    }

    public static bool Delete(int id)

    {

        return DBManager.Delete(id);

    }

}