<table>
<tr><td colspan="10">Faculty of Information Technology</td></tr>
<tr><td rowspan="5">I declare that I am familiar with, and will abide to the Examination rules of CTU

 
Signature</td><td colspan="9">SUBJECT NAME: Solutions Development
SUBJECT CODE: SLD521</td></tr>
<tr><td colspan="4">FORMATIVE ASSESSMENT
Duration: Sep 15 – 17 Oct
Date 17 October 2023
Total Marks: 100
Total pages:</td><td colspan="5">Examiner: Faith Muwishi
Moderator: Mr. Newton</td></tr>
<tr><td colspan="9">Student number</td></tr>
<tr><td>2</td><td>0</td><td>2</td><td>3</td><td>2</td><td>7</td><td>5</td><td>9</td><td></td></tr>
<tr><td colspan="4">Surname:
Poponi</td><td colspan="3">Initials:
M.P</td><td>/</td><td>%</td></tr>
</table>

Student Number:20221016 Name: Gontse Thelele

Student Number:20232605 Name: Gabriella Rakgotsoka

Table of Contents

Questions Paper

Create an MVC Search Weather project that will allow users to register their details such as email, and password, and then log into the application to enter a city in Gauteng.

Once the user enters the city, the application must display the city name, followed by the zip code of the city and the weather for that day.

Here are general steps you can use as a group to assist you:

1. Set up a new ASP.NET Web Application project in Visual Studio, selecting the MVC template.

2. Create a SQL Server database to store user information.

3. Define a User model class with properties for user ID, email, and password.

4. Create a database context class to interact with the database.

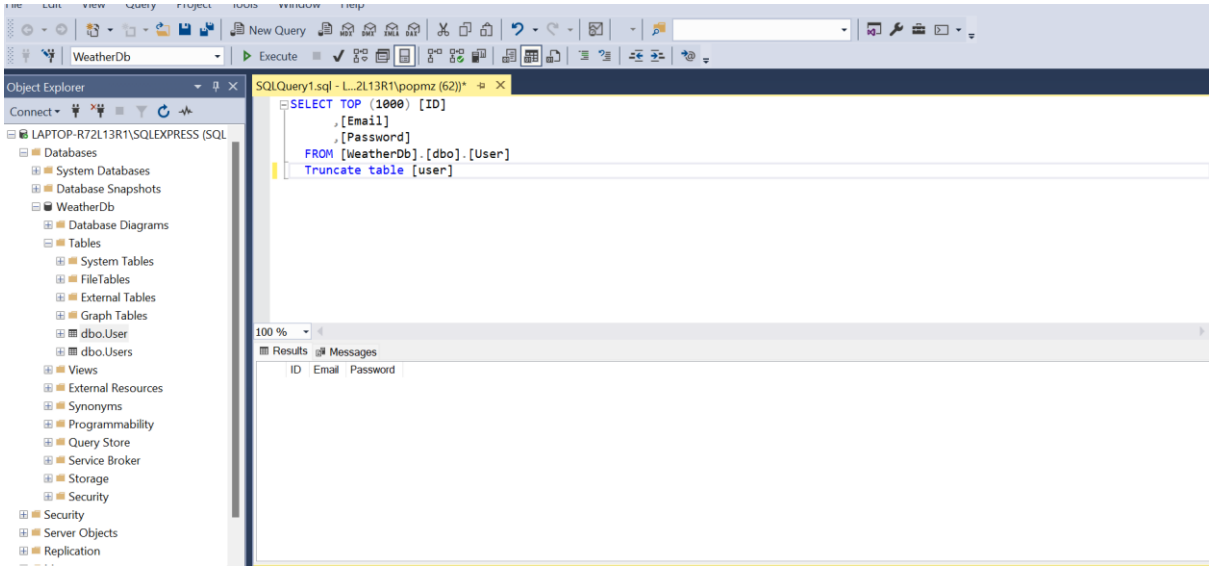
5. Implement user registration functionality in an AccountController with two actions: one to display the registration form, and one to handle the form submission and save the user to the database.

6. Implement user login functionality in the same AccountController with two actions: one to display the login form, and one to handle the form submission and authenticate the user using the database.

7. Obtain an API key from OpenWeatherMap and use it to implement a weather search feature in a new WeatherController with two actions: one to display the search form, and one to handle the form submission and display the weather information.

8. Add validation to the user registration and login forms using the data annotations in the User model class. 9. Use Bootstrap to style the user interface of the application

SQL Database: Screenshot of the SQL query with code



User Model Class

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace SearchWeatherApplication.ViewModel
{
   public class UserViewModel
   {
      [Key, Column(Order = 1)]
      [DatabaseGeneratedAttribute(DatabaseGeneratedOption.Identity)]
      public int ID { get; set; }

      [Required]
      [RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}")]
      public string Email { get; set; }

      [Required]
      [RegularExpression(@"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d).{8,15}$")]

      public string Password { get; set; }

      //[NotMapped]
      //[Required]
      //[System.ComponentModel.DataAnnotations.Compare("Password")]
      //public string ConfirmPassword { get; set; }

   }
}
```

Database context class to interact with the database

```
using System;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;
using SearchWeatherApplication.Models;

// Code scaffolded by EF Core assumes nullable reference types (NRTs) are not used or disabled.
// If you have enabled NRTs for your project, then un-comment the following line:
// #nullable disable

namespace SearchWeatherApplication.Data
{
    public partial class SearchWeatherDbContext : DbContext
    {
        public SearchWeatherDbContext()
        {
        }

        public SearchWeatherDbContext(DbContextOptions<SearchWeatherDbContext> options)
            : base(options)
        {
        }

        public virtual DbSet<User> User { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            if (!optionsBuilder.IsConfigured)
            {
#warning To protect potentially sensitive information in your connection string, you should move it out of source code. See http://go.microsoft.com/fwlink/?LinkId=723263 for guidance on storing connection strings.
                optionsBuilder.UseSqlServer("Server=LAPTOP-R72L13R1\\SQLEXPRESS;Database=WeatherDb;Trusted_Connection=True");
            }
        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<User>(entity =>
            {
                entity.Property(e => e.Id).HasColumnName("ID");

                entity.Property(e => e.Email)
                    .HasMaxLength(50)
                    .IsUnicode(false);

                entity.Property(e => e.Password)
                    .HasMaxLength(50)
                    .IsUnicode(false);
            });

            OnModelCreatingPartial(modelBuilder);
        }

        partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
    }
}
```

User login and registration functionality in an AccountController

```
using SearchWeatherApplication.Data;
using SearchWeatherApplication.ViewModel;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;

namespace SearchWeatherApplication.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {

            return View(); // (view) ~/Views/Home/Index.cshtml
        }

        [HttpGet]
        public ActionResult LogedIn()
        {
            return View(); // (view) ~/Views/Home/LogedIn.cshtml
        }

        [HttpPost]
        public ActionResult Login(Models.User user)
        {
            if (ModelState.IsValid)
            {
                if (IsValid(user.Email, user.Password))
                {
                    FormsAuthentication.SetAuthCookie(user.Email, false);
                    using (var dbContext = new SearchWeatherDbContext())
                    {
                        var getUser = dbContext.User.FirstOrDefault(x => x.Email == user.Email && x.Password ==
user.Password);
                        if (getUser != null)
                        {
                            ViewBag.Message = "Successfully Logged In";
                            return RedirectToAction("LogedIn", "Home");
                        }
                        ViewBag.Message = "Not logged in, try again.";
                        return RedirectToAction("Index", "Home");
                    }
                }
                else
                {
                    ModelState.AddModelError("", "Error!");
                }
            }
            return View(user); // (view) ~/Views/Home/LogedIn.cshtml
        }

        public ActionResult Register()
        {
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Register(UserViewModel vm)
```

```
        {
            if (ModelState.IsValid)
            {
                Models.User user = new Models.User { Email = vm.Email, Password = vm.Password };
                using (var dbContext = new SearchWeatherDbContext())
                {
                    dbContext.User.Add(user);
                    dbContext.SaveChanges();
                    ModelState.Clear();
                    ViewBag.Message = "Successfully Registration Done";
                }
            }
            return View(vm);
        }

        public ActionResult Logout()
        {
            FormsAuthentication.SignOut();
            return RedirectToAction("Index", "Home");
        }

        private bool IsValid(string userSol, string passSol)
        {
            return true;
        }
    }
}
```

Weather Controller

```
using SearchWeatherApplication.Data;
using SearchWeatherApplication.ViewModel;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;

namespace SearchWeatherApplication.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {

            return View(); // (view) ~/Views/Home/Index.cshtml
        }

        [HttpGet]
        public ActionResult LogedIn()
        {
            return View(); // (view) ~/Views/Home/LogedIn.cshtml
        }

        [HttpPost]
        public ActionResult Login(Models.User user)
        {
            if (ModelState.IsValid)
```

```
        {
            if (IsValid(user.Email, user.Password))
            {
                FormsAuthentication.SetAuthCookie(user.Email, false);
                using (var dbContext = new SearchWeatherDbContext())
                {
                    var getUser = dbContext.User.FirstOrDefault(x => x.Email == user.Email && x.Password ==
user.Password);
                    if (getUser != null)
                    {
                        ViewBag.Message = "Successfully Logged In";
                        return RedirectToAction("LogedIn", "Home");
                    }
                    ViewBag.Message = "Not logged in, try again.";
                    return RedirectToAction("Index", "Home");
                }
            }
            else
            {
                ModelState.AddModelError("", "Error!");
            }
        }
        return View(user); // (view) ~/Views/Home/LogedIn.cshtml
    }

    public ActionResult Register()
    {
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Register(UserViewModel vm)
    {
        if (ModelState.IsValid)
        {
            Models.User user = new Models.User { Email = vm.Email, Password = vm.Password };
            using (var dbContext = new SearchWeatherDbContext())
            {
                dbContext.User.Add(user);
                dbContext.SaveChanges();
                ModelState.Clear();
                ViewBag.Message = "Successfully Registration Done";
            }
        }
        return View(vm);
    }

    public ActionResult Logout()
    {
        FormsAuthentication.SignOut();
        return RedirectToAction("Index", "Home");
    }

    private bool IsValid(string userSol, string passSol)
    {
        return true;
    }
  }
}
```

Weather View Model

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace SearchWeatherApplication.Models
{
    public class ResultViewModel
    {
        public string City { get; set; }
        public string Country { get; set; }
        public string Lat { get; set; }
        public string Lon { get; set; }
        public string Description { get; set; }
        public string Humidity { get; set; }
        public string TempFeelsLike { get; set; }
        public string Temp { get; set; }
        public string TempMax { get; set; }
        public string TempMin { get; set; }
        public string WeatherIcon { get; set; }
    }

    public class Coord
    {
        public double lon { get; set; }
        public double lat { get; set; }
    }

    public class Weather
    {
        public int id { get; set; }
        public string main { get; set; }
        public string description { get; set; }
        public string icon { get; set; }
    }

    public class Main
    {
        public double temp { get; set; }
        public double feels_like { get; set; }
        public double temp_min { get; set; }
        public double temp_max { get; set; }
        public int pressure { get; set; }
        public int humidity { get; set; }
    }

    public class Wind
    {
        public double speed { get; set; }
        public int deg { get; set; }
    }

    public class Clouds
    {
        public int all { get; set; }
    }

    public class Sys
```

```
    {
        public int type { get; set; }
        public int id { get; set; }
        public string country { get; set; }
        public int sunrise { get; set; }
        public int sunset { get; set; }
    }

    public class RootObject
    {
        public Coord coord { get; set; }
        public List<Weather> weather { get; set; }
        public string @base { get; set; }
        public Main main { get; set; }
        public double visibility { get; set; }
        public Wind wind { get; set; }
        public Clouds clouds { get; set; }
        public double dt { get; set; }
        public Sys sys { get; set; }
        public double timezone { get; set; }
        public int id { get; set; }
        public string name { get; set; }
        public double cod { get; set; }
    }
}
```

Code for Homepage/index page (index.cshtml)

```
@model SearchWeatherApplication.ViewModel.UserViewModel
@{
    ViewBag.Title = "Home Page";
}

<main>
    <section class="row" aria-labelledby="aspnetTitle">

    </section>

    <div class="row">
        @using (Html.BeginForm("Login", "Home", FormMethod.Post))
        {
            <div class="prijavno-okno">
                <fieldset>
                    <legend>Login form:</legend>
                    <form method="get">
                        @*<label>Username:</label>@Html.TextBoxFor(u => u.Email)<br />
                        <label>Password:</label>@Html.PasswordFor(u => u.Password)<br />*@

                        <div class="editor-label">
                            @Html.LabelFor(model => model.Email)
                        </div>
                        <div class="editor-field">
                            @Html.EditorFor(model => model.Email)
                            @Html.ValidationMessageFor(model => model.Email)
                        </div>
                        <div class="editor-label">
                            @Html.LabelFor(model => model.Password)
                        </div>
                        <div class="editor-field">
```

```
                @Html.EditorFor(model => model.Password)
                @Html.ValidationMessageFor(model => model.Password)
            </div>

            <input class="gumb" type="submit" value="Login">
        </form>
    </fieldset>
</div>
}
@Html.ActionLink("Register", "Register", "Home")


</div>
</main>
```

Register page (Register.cshtml)

Code:

```
@model SearchWeatherApplication.ViewModel.UserViewModel

@{
    ViewBag.Title = "Register";
}


<h2>Register</h2>

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)


    <fieldset>
        @Html.AntiForgeryToken()
        @if (ViewBag.Message != null)
        {
            <div style="border:solid 1px green">
                @ViewBag.Message
            </div>
        }


        <div class="editor-label">
            @Html.LabelFor(model => model.Email)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Email)
            @Html.ValidationMessageFor(model => model.Email)
        </div>
        <div class="editor-label">
            @Html.LabelFor(model => model.Password)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Password)
            @Html.ValidationMessageFor(model => model.Password)
        </div>
        @*<div class="editor-label">
            @Html.LabelFor(model => model.ConfirmPassword)
```

```
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ConfirmPassword)
            @Html.ValidationMessageFor(model => model.ConfirmPassword)
        </div>*@
        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}


<div>
    @Html.ActionLink("Back to Index", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Login page (login.cshtml)

Code:

```
@{
    ViewBag.Title = "Login";
}

<h2>Error</h2>

@Html.ActionLink("Back", "Index", "Home")
```

Logged in page

Code:

```
@model SearchWeatherApplication.ViewModel.UserViewModel
@{
    ViewBag.Title = "LogedIn";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>LogedIn</h2>
<br />
@Html.ActionLink("View Weather Forecast", "Weather", "Weather")
```

Weather page (Weather.cshtml)

```
@{
    ViewBag.Title = "Weather";
}

<h1>Search City and Get Weather Forecast</h1>

<div>
    <strong>City Name  :</strong><input id="txtCity" type="text" />
    <br />
```

```
    <br />
    <button id="btnSubmit">Get Weather Forecast</button>
</div>


<div>
    <h2>Weather Forecast</h2>
    <table>
        <tr>
            <td>Weather Symbol Icon <img id="imgWeatherIconUrl" src="" title="Weather Icon" /></td>
        </tr>
        <tr>
            <td>
                <strong>City: </strong>
                <span id="lblCity"></span>  ,
                <span id="lblCountry"></span>
            </td>
        </tr>
        <tr>
            <td>
                <strong>Latitude: </strong>
                <label id="lblLat"></label><br />
                <strong>Longitude: </strong>
                <label id="lblLon"></label>

            </td>
        </tr>
        <tr>
            <td>
                <strong>Description:</strong>
                <label id="lblDescription"></label><br />
                <strong>Humidity:</strong>
                <label id="lblHumidity"></label>
            </td>
        </tr>
        <tr>
            <td>
                Temperature (Feels Like)<label id="lblTempFeelsLike"></label><br />
                Temperature <label id="lblTemp"></label><br />
                Temperature (Min)<label id="lblTempMin"></label><br />
                Temperature (Max)<label id="lblTempMax"></label><br />
            </td>
        </tr>


    </table>
</div>

<script>


    $("#btnSubmit").click(function () {

        var cityname = $("#txtCity").val();
        if (cityname.length > 0)
        {
        $.ajax({
            url: "http://localhost:52189/Home/WeatherDetail?City="+cityname,
            type: "POST",
            success: function (rsltval) {
                var data =JSON.parse(rsltval);
                console.log(data);
```

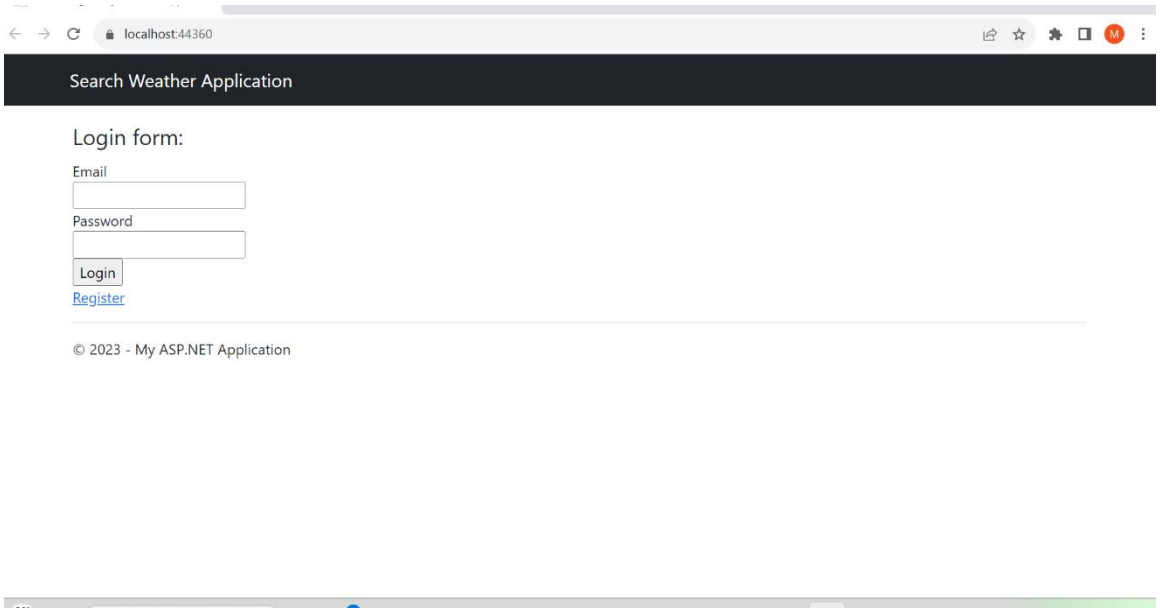
```
            $("#lblCity").html(data.City);
            $("#lblCountry").text(data.Country);
            $("#lblLat").text(data.Lat);
            $("#lblLon").text(data.Lon);
            $("#lblDescription").text(data.Description);
            $("#lblHumidity").text(data.Humidity);
            $("#lblTempFeelsLike").text(data.TempFeelsLike);
            $("#lblTemp").text(data.Temp);
            $("#lblTempMax").text(data.TempMax);
            $("#lblTempMin").text(data.TempMin);
            $("#imgWeatherIconUrl").attr("src", "http://openweathermap.org/img/w/" + data.WeatherIcon + ".png");
            //data - response from server
        },
        error: function () {

        }
    });
    }
    else
    {
        alert("City Not Found");
    }
});

</script>
```

Layoutpage (_layout.cshtml)

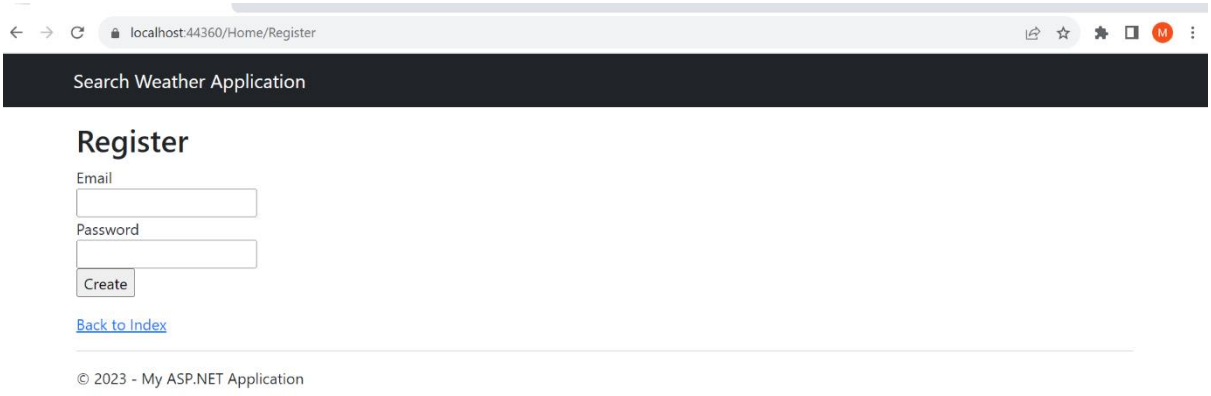
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
</head>
<body>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-dark">
        <div class="container">
            @Html.ActionLink("Search Weather Application", "Index", "Home", new { area = "" }, new { @class =
"navbar-brand" })
            <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"
title="Toggle navigation" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse d-sm-inline-flex justify-content-between">
                <ul class="navbar-nav flex-grow-1">
                    @*<li>@Html.ActionLink("Home", "Index", "Home", new { area = "" }, new { @class = "nav-link"
})</li>
                    <li>@Html.ActionLink("About", "About", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home", new { area = "" }, new { @class = "nav-link"
})</li>*@
                </ul>
            </div>
        </div>
```

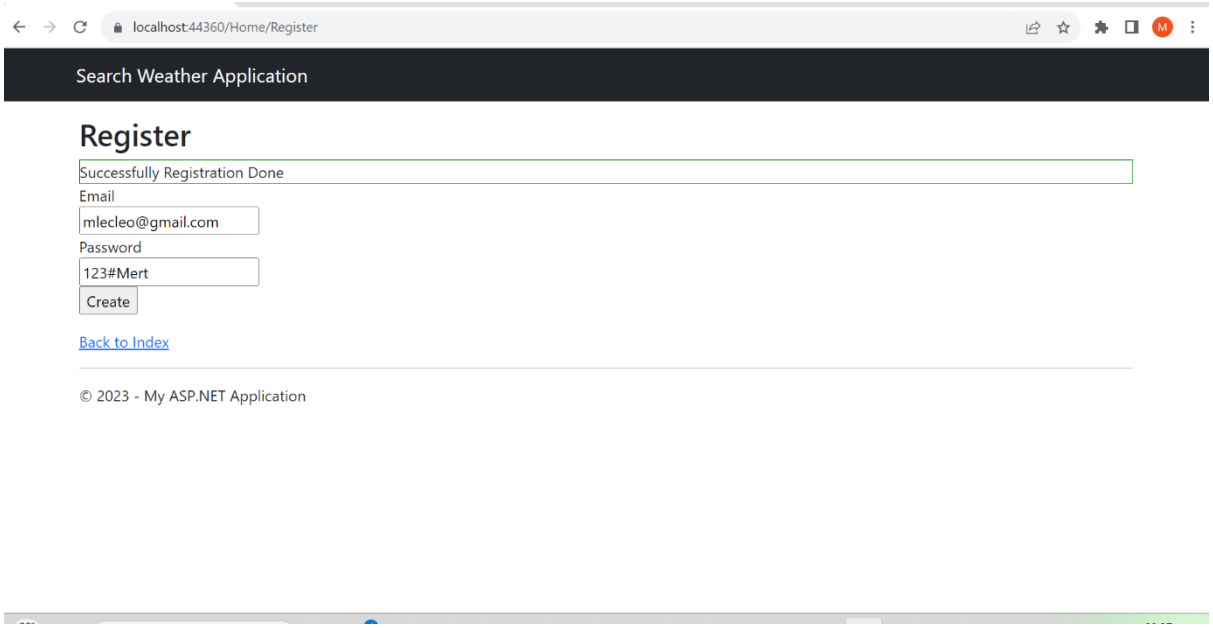
```
    </nav>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>


    @RenderSection("scripts", required: false)
</body>
</html>
```

HomePage screenshot

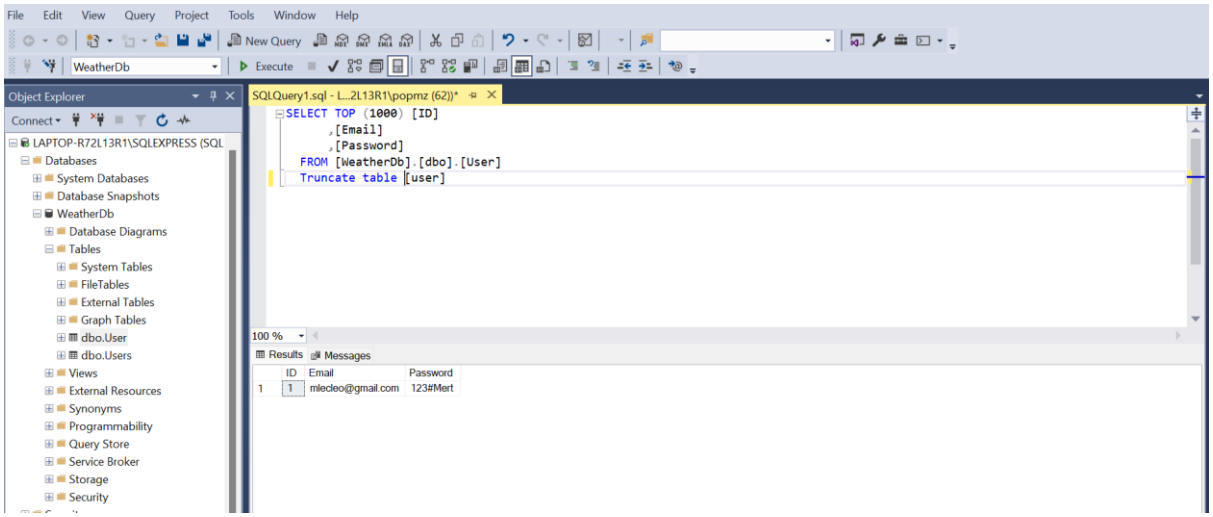


Register Page screenshot

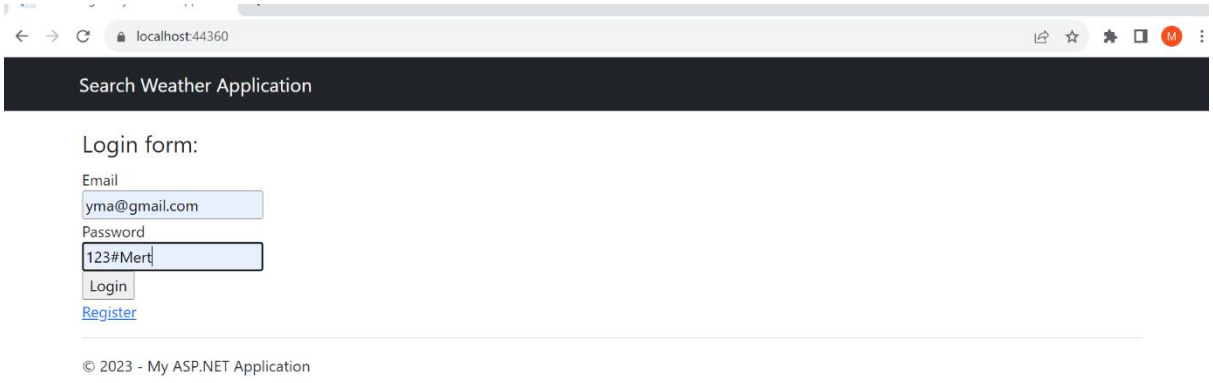


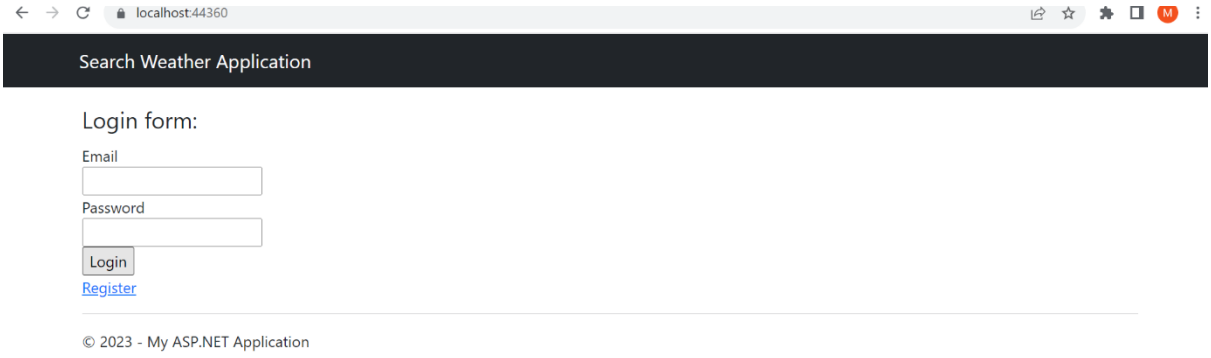


SQL Database: User successfully added to the database

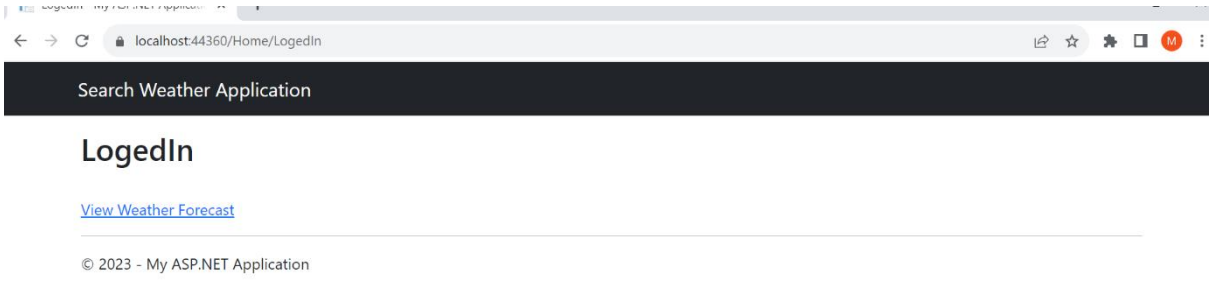


Validation with Validation

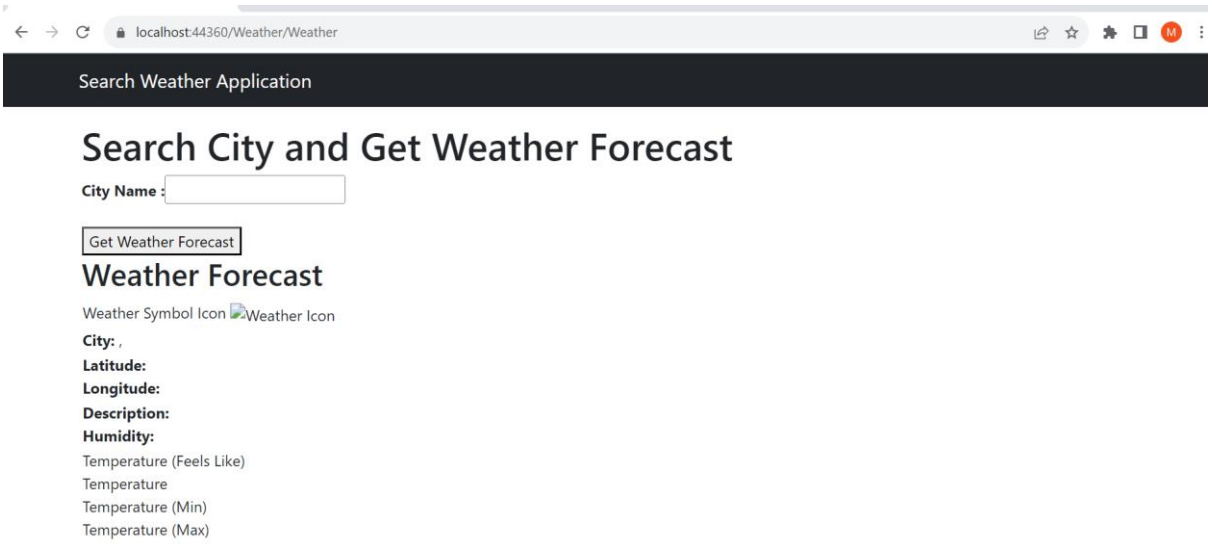




Page login functionality (successfully logged in)



Search Weather page



Functionality of the weather page