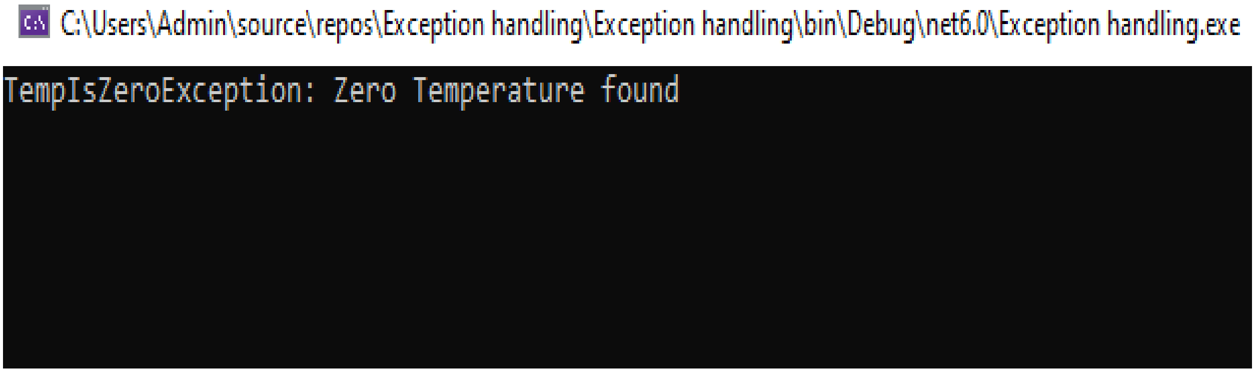
# LAB 1

## TITLE:  IMPLEMENTING EXCEPTION HANDLING

**//CODE**

```
using System;

namespace UserDefinedException
{
    class TestTemperature
    {
        static void Main(string[] args)
        {
            Temperature temp = new Temperature();
            try
            {
                temp.showTemp();
            }
            catch (TempIsZeroException e)
            {
                Console.WriteLine("TempIsZeroException: {0}", e.Message);
            }
            Console.ReadKey();
        }
    }
}
public class TempIsZeroException : Exception
{
    public TempIsZeroException(string message) : base(message)
    {
    }
}
public class Temperature
{
    int temperature = 0;

    public void showTemp()
    {

        if (temperature == 0)
        {
            throw (new TempIsZeroException("Zero Temperature found"));
        }
        else
        {
            Console.WriteLine("Temperature: {0}", temperature);
        }
    }
}
```

**//Output**



C:\Users\Admin\source\repos\Exception handling\Exception handling\bin\Debug\net6.0\Exception handling.exe

```
TempIsZeroException: Zero Temperature found
```

**LAB 2**

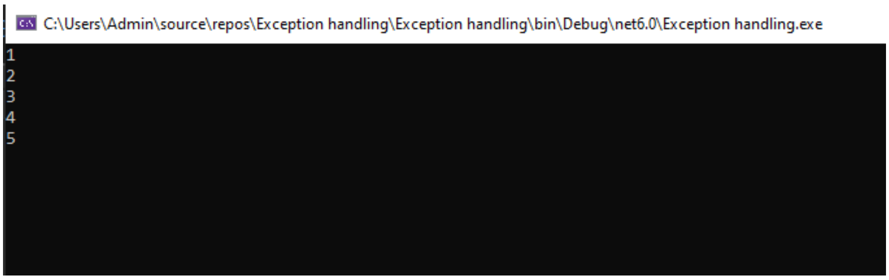
## TITLE: IMPLEMENTING GENERIC IN C#

**//CODE**

```
using System;
using System.Collections.Generic;
namespace GenericApp
{
    public class TestClass<T>
    {
        // define an Array of Generic type with length 5
        T[] obj = new T[5];
        int count = 0;

        // adding items mechanism into generic type
        public void  Add(T item)
        {
            //checking length
            if (count + 1 < 6)
            {
                obj[count] = item;
            }
            count++;
        }
        //indexer for foreach statement iteration
        public T this[int index]
        {
            get { return obj[index]; }
            set { obj[index] = value; }
  }
    }
    class Program
    {
```

```
static void Main(string[] args)
 {
    //instantiate generic with Integer
    TestClass<int> intObj = new TestClass<int>();

    //adding integer values into collection
    intObj.Add(1);
    intObj.Add(2);
    intObj.Add(3);   //No boxing
    intObj.Add(4);
    intObj.Add(5);
    //displaying values
    for (int i = 0; i < 5; i++)
    {
       Console.WriteLine(intObj[i]);   //No unboxing
    }
    Console.ReadKey();
 }
```

**// OUTPUT**



C:\Users\Admin\source\repos\Exception handling\Exception handling\bin\Debug\net6.0\Exception handling.exe

```
1
2
3
4
5
```

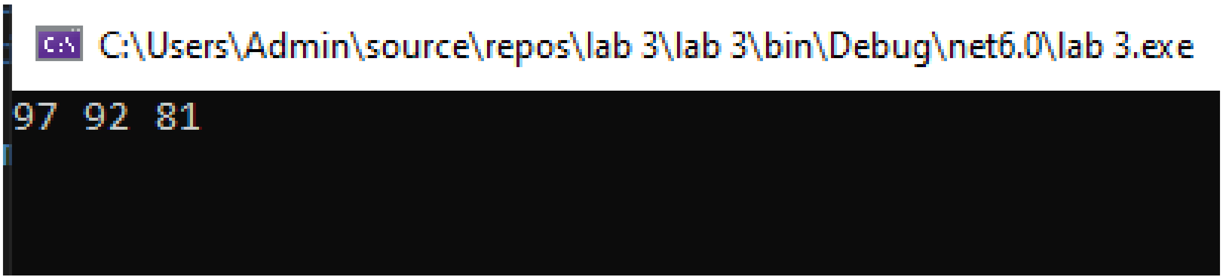
# LAB 3

## TITLE: IMPLEMENTING LINQ IN C#

**//CODE**

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

namespace Operators

{

    class LINQQueryExpressions

    {

     static void Main ()

        {

            // Specify the data source.

             Int [] scores = new int[] { 97, 92, 81, 60 };

            // Define the query expression.

            IEnumerable<int> scoreQuery = from score in scores where score > 80 select
score;

            // Execute the query.

            foreach (int i in scoreQuery)

            {

                Console.Write(i + " ");

            }

            Console.ReadLine();

        }

    }

}
```

**//OUTPUT**



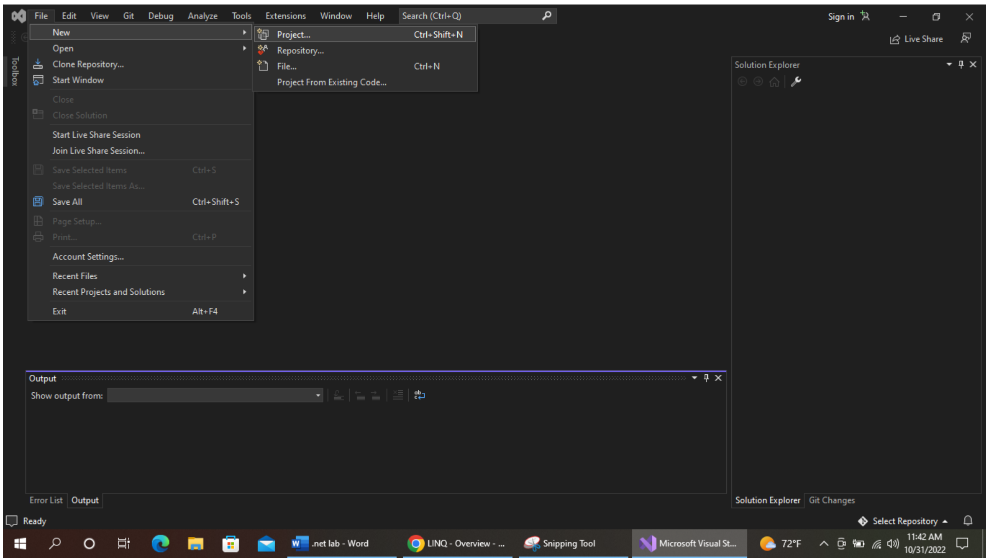
C:\Users\Admin\source\repos\lab 3\lab 3\bin\Debug\net6.0\lab 3.exe

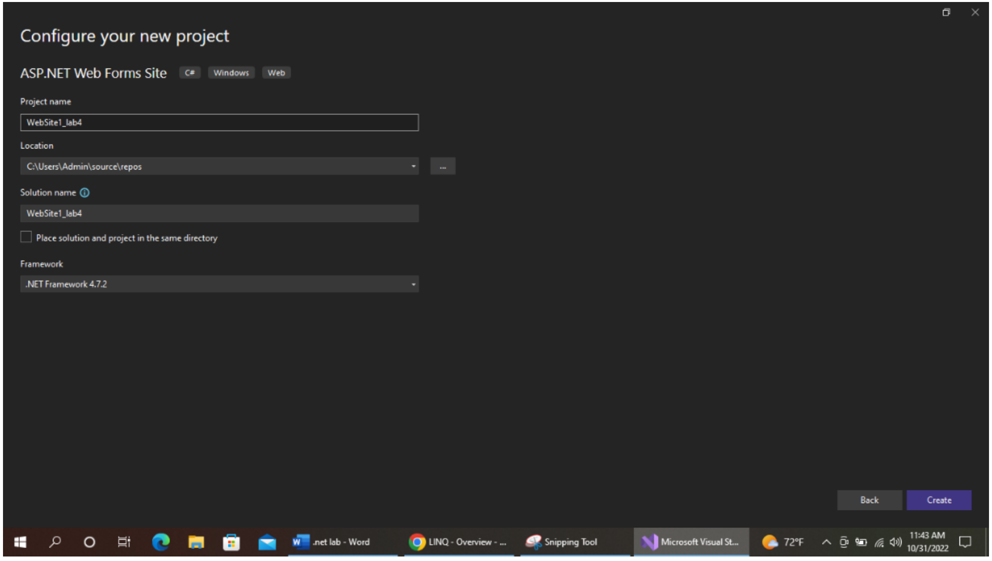
```
97 92 81
```

# LAB 4

# TITLE: METHOD TO CREATE ASP.NET FRAMEWORK

## STEPS:





Configure your new project

ASP.NET Web Forms Site C# Windows Web

Project name

WebSite1_lab4

Location

C:\Users\Admin\source\repos

Solution name

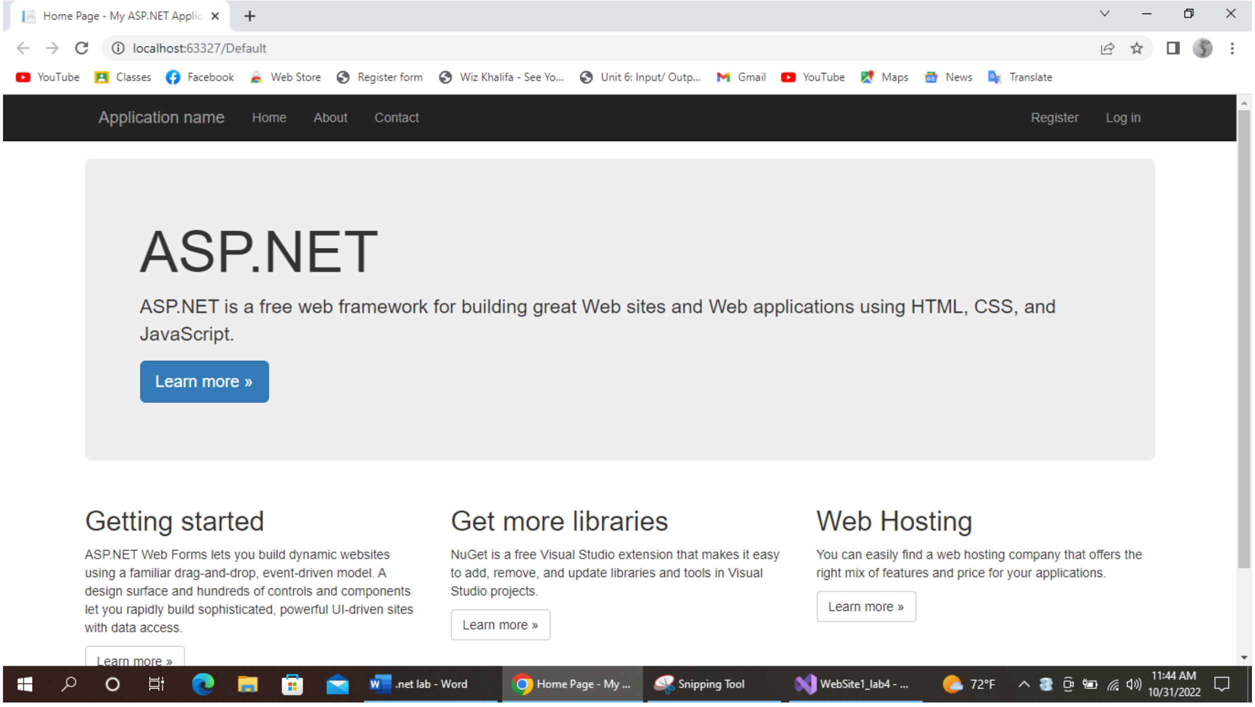
WebSite1_lab4

□ Place solution and project in the same directory

Framework

.NET Framework 4.7.2

## //OUTPUT



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Learn more »

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

Learn more »

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

Learn more »

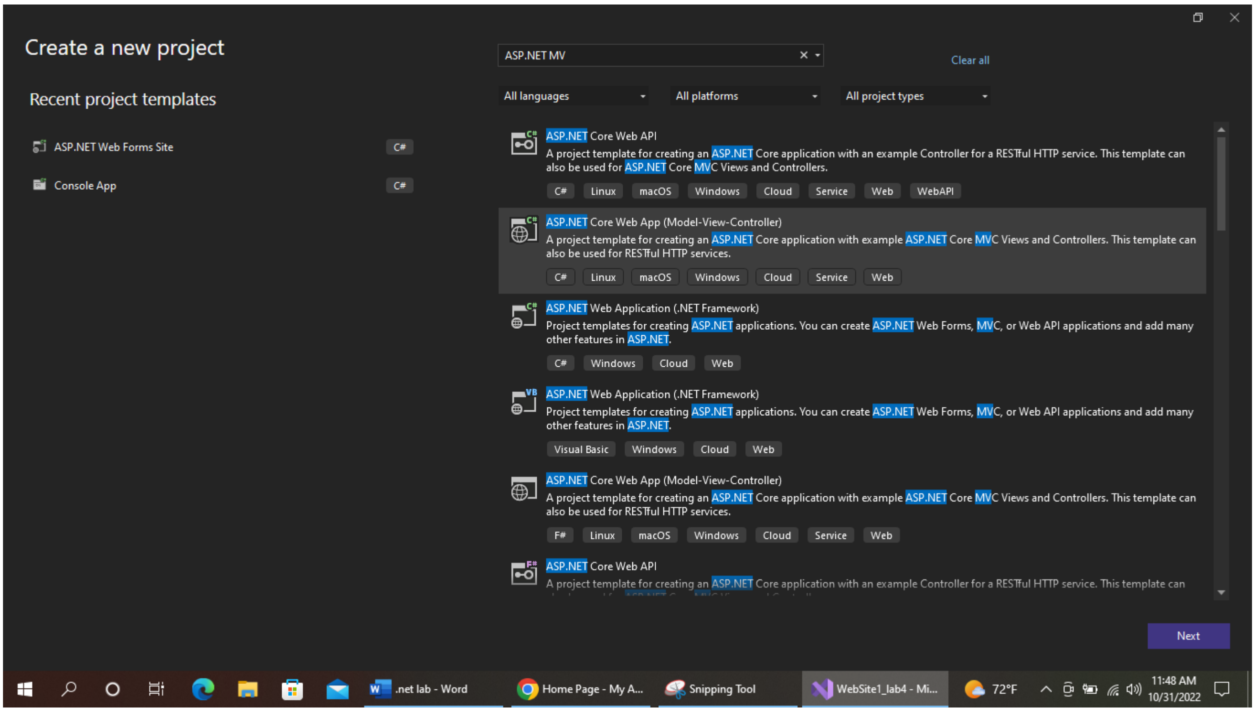
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

Learn more »

## TITLE: ASP.NET MVC

STEPS



ASP.NET MV

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# Linux macOS Windows Cloud Service Web WebAPI

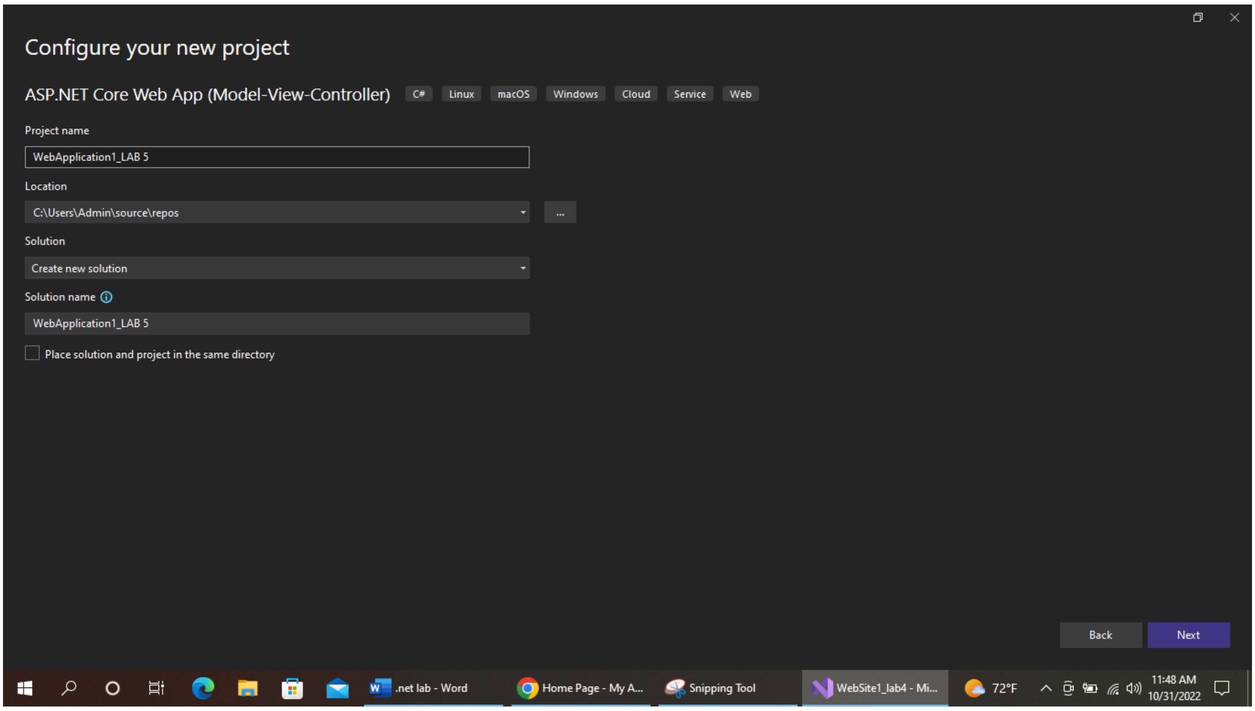
ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web

ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Cloud Web

ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Visual Basic Windows Cloud Web

ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
F# Linux macOS Windows Cloud Service Web

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can



Configure your new project

ASP.NET Core Web App (Model-View-Controller)

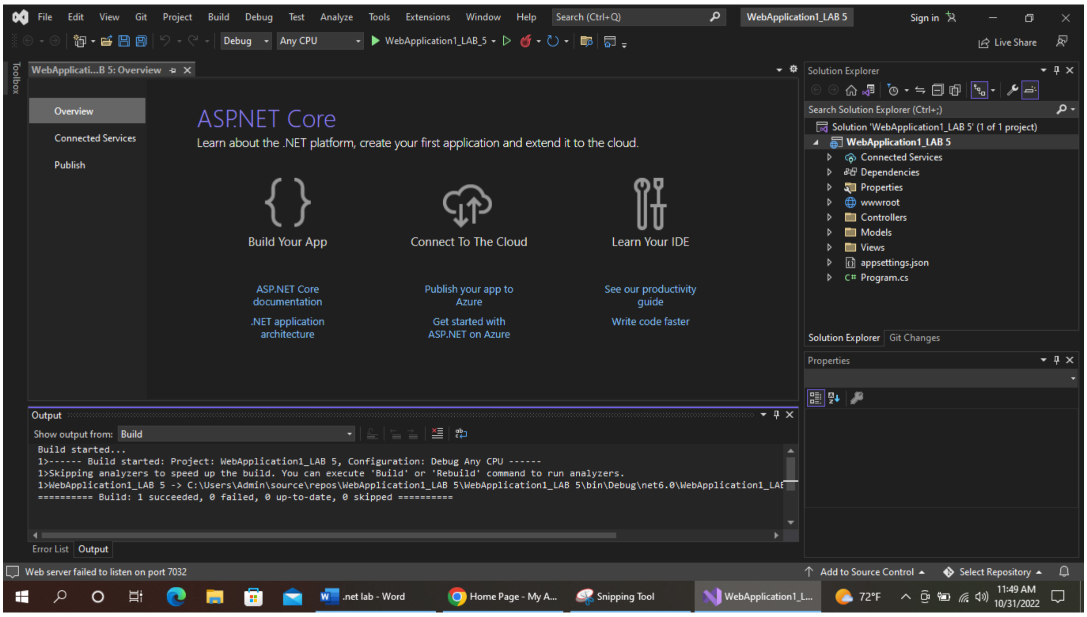
Project name: WebApplication1_LAB 5

Location: C:\Users\Admin\source\repos

Solution: Create new solution

Solution name: WebApplication1_LAB 5

□ Place solution and project in the same directory



ASP.NET Core

Learn about the .NET platform, create your first application and extend it to the cloud.

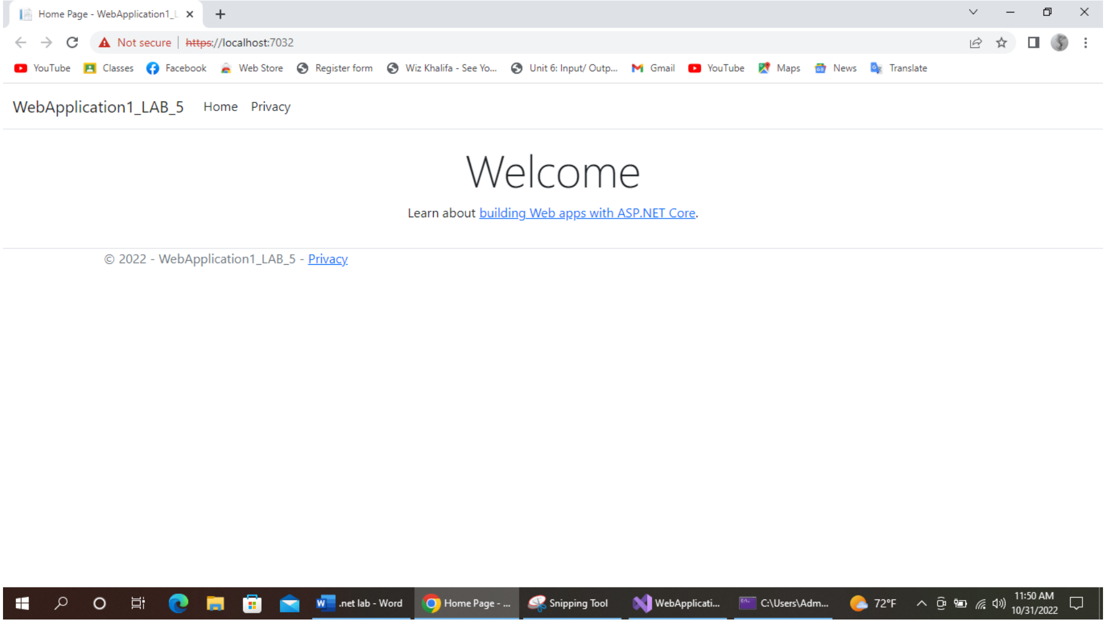
Output

Show output from: Build

```
Build started...
1>------ Build started: Project: WebApplication1_LAB 5, Configuration: Debug Any CPU ------
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>WebApplication1_LAB 5 -> C:\Users\Admin\source\repos\WebApplication1_LAB 5\WebApplication1_LAB 5\bin\Debug\net6.0\WebApplication1_LAE
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Web server failed to listen on port 7032

# OUTPUT



WebApplication1_LAB_5 Home Privacy

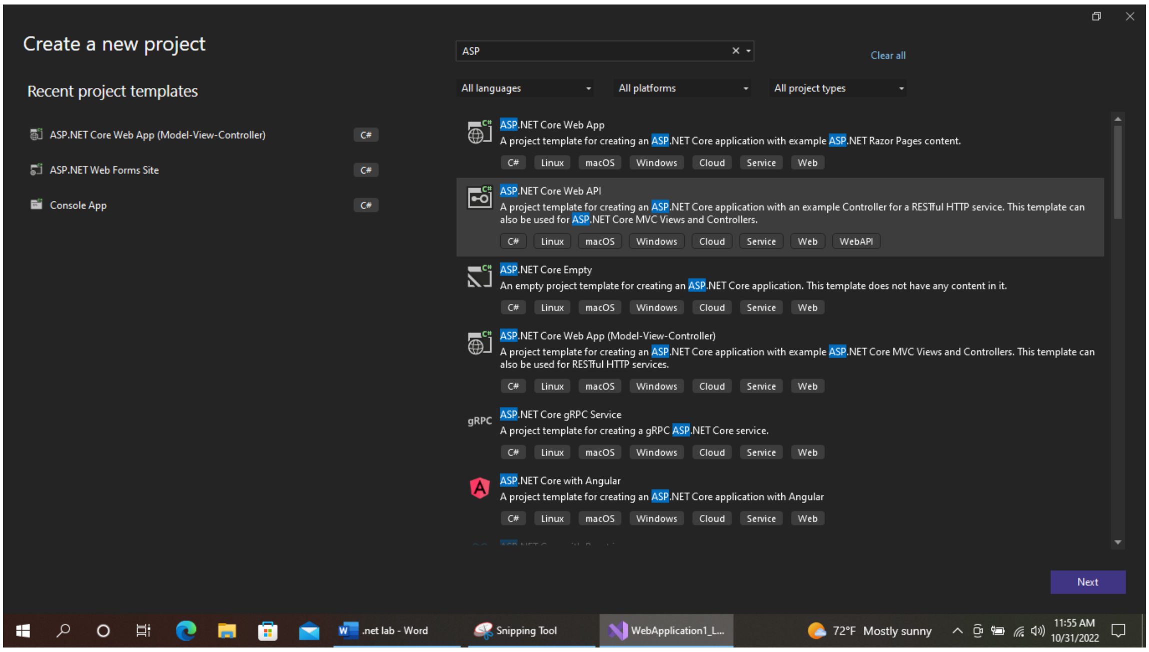
Welcome

Learn about building Web apps with ASP.NET Core.

© 2022 - WebApplication1_LAB_5 - Privacy

# TITLE: ASP.NET WEB API

## STEPS:


Create a new project

Recent project templates

- ASP.NET Core Web App (Model-View-Controller) C#
- ASP.NET Web Forms Site C#
- Console App C#

ASP

ASP.NET Core Web App
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

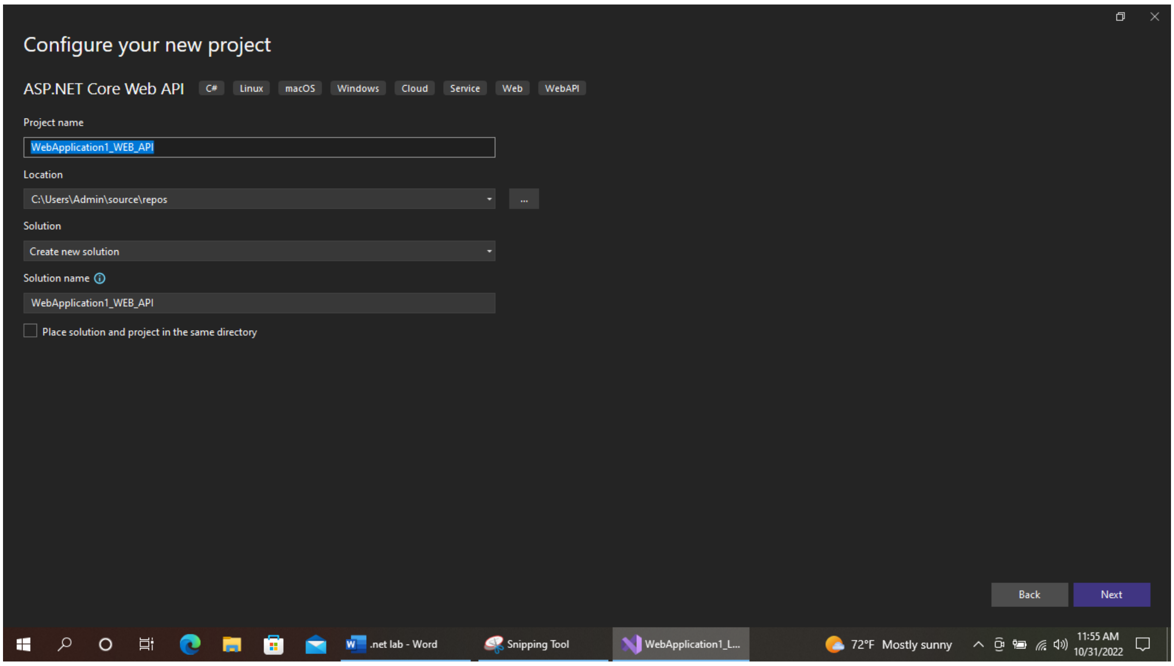
ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

ASP.NET Core gRPC Service
A project template for creating a gRPC ASP.NET Core service.

ASP.NET Core with Angular
A project template for creating an ASP.NET Core application with Angular



Configure your new project

ASP.NET Core Web API

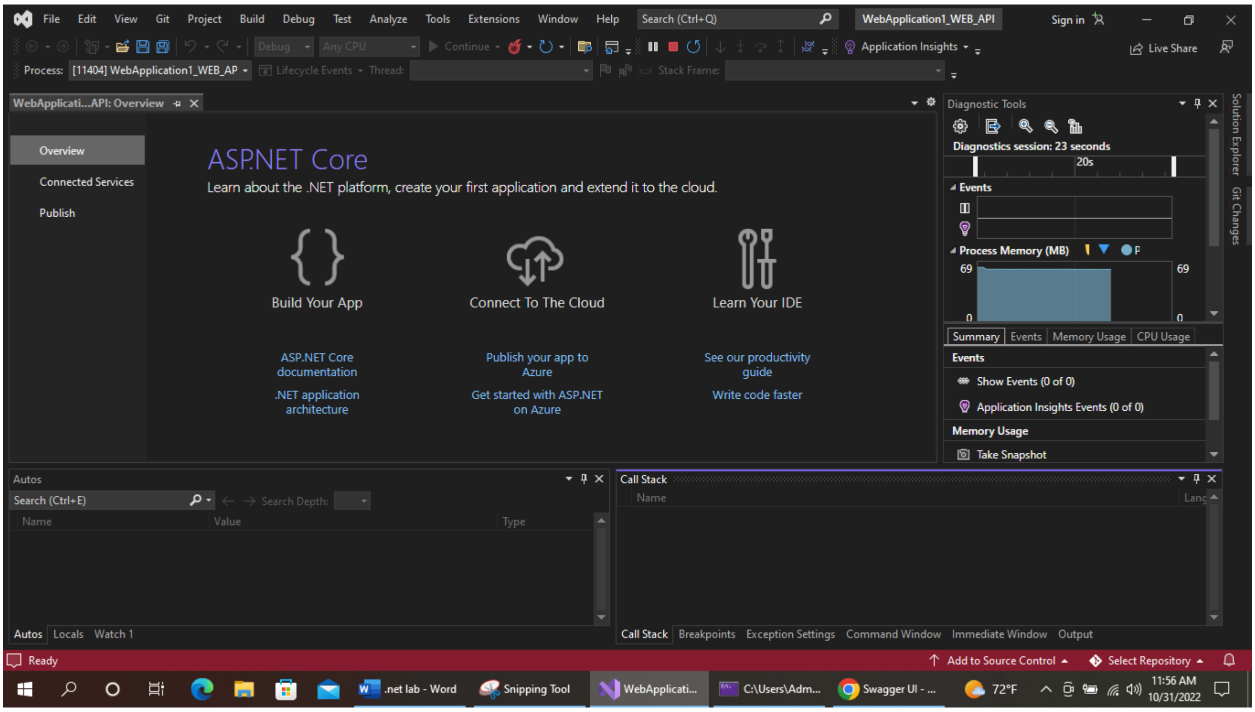
Project name: WebApplication1_WEB_API

Location: C:\Users\Admin\source\repos

Solution: Create new solution

Solution name: WebApplication1_WEB_API

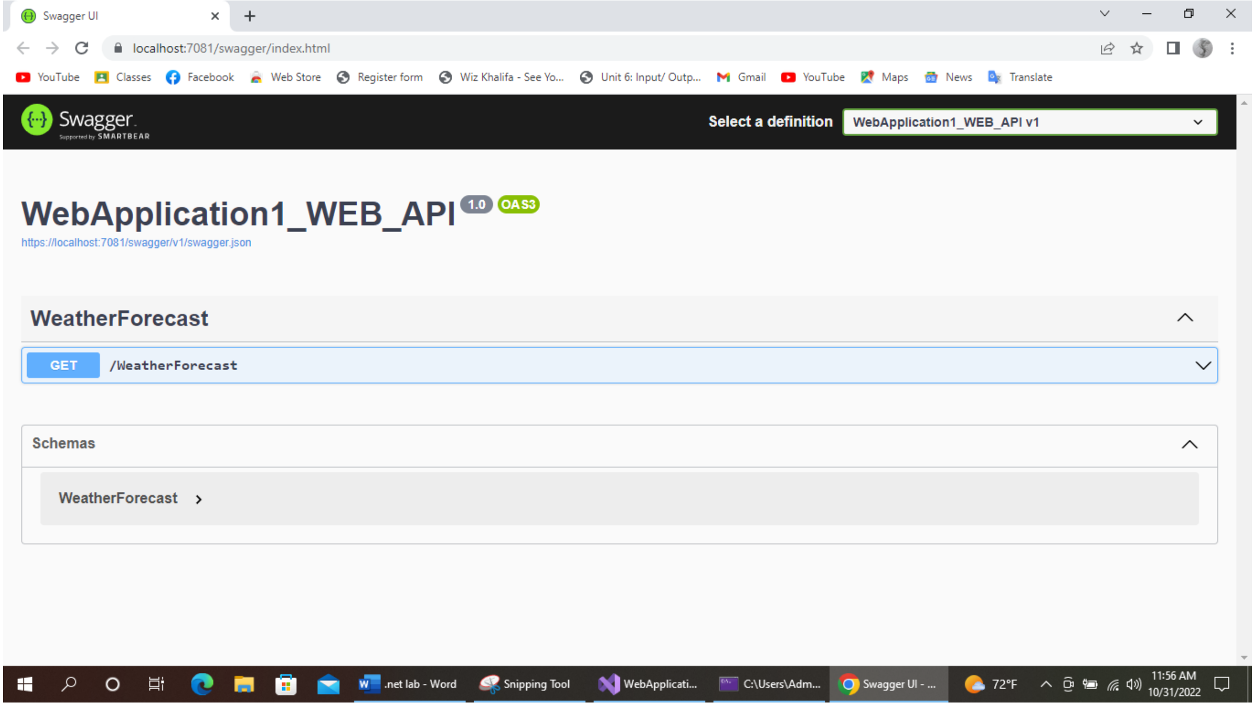
□ Place solution and project in the same directory



ASP.NET Core

Learn about the .NET platform, create your first application and extend it to the cloud.

## OUTPUT:



WebApplication1_WEB_API 1.0 OAS3

https://localhost:7081/swagger/v1/swagger.json

WeatherForecast

GET /WeatherForecast

Schemas

WeatherForecast

## LAB 5

## TITLE: JQUERY FORM VALIDATION

**//CODE:**

**Index.htlm**

```
<!DOCTYPE html>
<html>
<head>
        <!-- Latest compiled and minified CSS -->
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/
        4.0.0/css/bootstrap.min.css">
        <!-- jQuery library -->
        <script src="https://ajax.googleapis.com/ajax/libs/
        jquery/3.3.1/jquery.min.js">
        </script>
        <!-- Popper JS -->
        <script src="https://cdnjs.cloudflare.com/ajax/libs/
        popper.js/1.12.9/umd/popper.min.js">
        </script>
        <!-- Latest compiled JavaScript -->
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/
        4.0.0/js/bootstrap.min.js">
        </script>
</head>

<body><br>
        <p class="text-center">
                FORM VALIDATION USING JQUERY
        </p>



        <div class="container">
                <div class="col-lg-8
        m-auto d-block">
            <form>
                <div class="form-group">
                    <label for="user">
                        Username:
                    </label>
            <input type="text" name="username" id="usernames" class="form-
    control">
                    <h5 id="usercheck" style="color: red;">
                        **Username is missing
                    </h5>
                </div>

                <div class="form-group">
                    <label for="user">
                        Email:
                    </label>
            <input type="email" name="email" id="email" required class="form-
control">
                    <small id="emailvalid" class="form-text
                text-muted invalid-feedback">
                        Your email must be a valid email
                    </small>
                </div>

                <div class="form-group">
                    <label for="password">
                        Password:
                    </label>
            <input type="password" name="pass" id="password" class="form-
control">
                    <h5 id="passcheck" style="color: red;">
                        **Please Fill the password
                    </h5>
                </div>

                <div class="form-group">
                    <label for="conpassword">
                        Confirm Password:
                    </label>
        <input type="password" name="username" id="conpassword" class="form-control">
                    <h5 id="conpasscheck" style="color: red;">
                        **Password didn't match
                    </h5>
                </div>

            <input type="submit" id="submitbtn" value="Submit" class="btn btn-primary">
            </form>
        </div>
    </div>

    <!-- Including app.js jQuery Script -->
    <script src="app.js"></script>
</body>

</html>
```

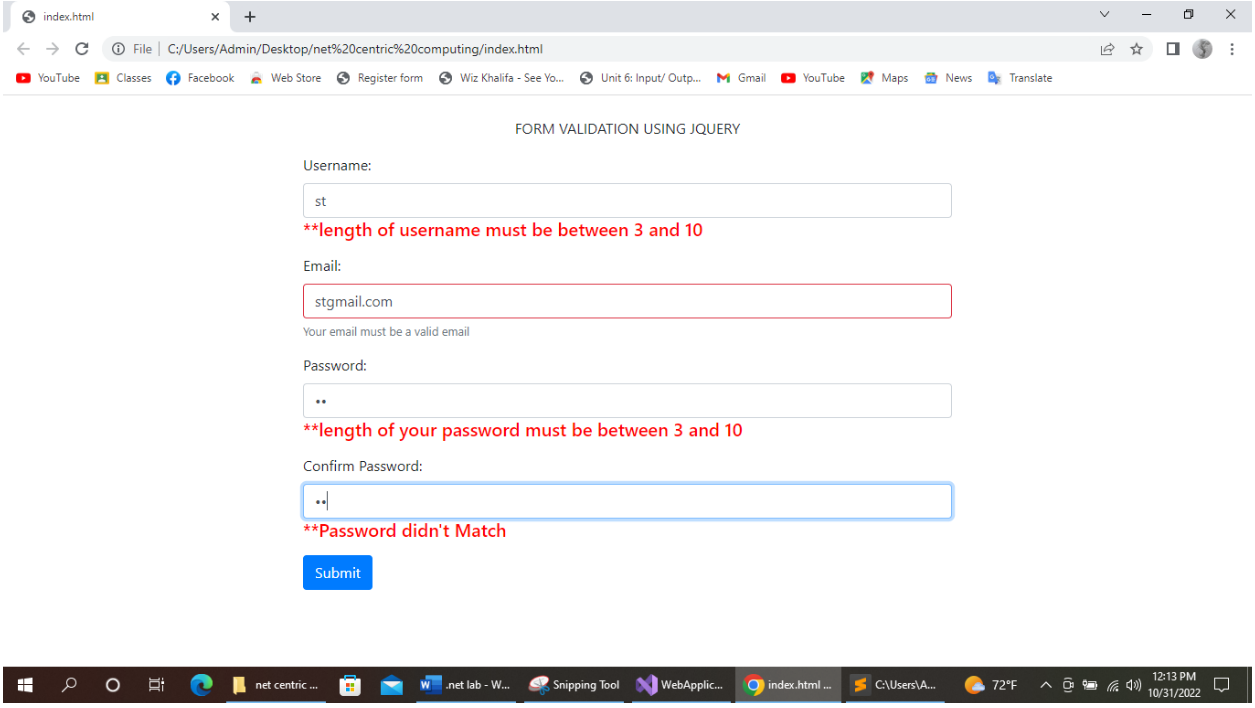
APP.JS

```
// Document is ready
$(document).ready(function () {
// Validate Username
$("#usercheck").hide();
let usernameError = true;
$("#usernames").keyup(function () {
    validateUsername();
});

function validateUsername() {
    let usernameValue = $("#usernames").val();
    if (usernameValue.length == "") {
    $("#usercheck").show();
    usernameError = false;
    return false;
    } else if (usernameValue.length < 3 || usernameValue.length > 10) {
    $("#usercheck").show();
    $("#usercheck").html("**length of username must be between 3 and 10");
    usernameError = false;
    return false;
    } else {
    $("#usercheck").hide();
    }
}

// Validate Email
const email = document.getElementById("email");
email.addEventListener("blur", () => {
    let regex = /^([_\-\.0-9a-zA-Z]+)@([_\-\.0-9a-zA-Z]+)\.([a-zA-Z]){2,7}$/;
    let s = email.value;
    if (regex.test(s)) {
    email.classList.remove("is-invalid");
    emailError = true;
    } else {
    email.classList.add("is-invalid");
    emailError = false;
    }
});
```

```
// Validate Password
$("#passcheck").hide();
let passwordError = true;
$("#password").keyup(function () {
        validatePassword();
});
function validatePassword() {
        let passwordValue = $("#password").val();
        if (passwordValue.length == "") {
        $("#passcheck").show();
        passwordError = false;
        return false;
        }
        if (passwordValue.length < 3 || passwordValue.length > 10) {
        $("#passcheck").show();
        $("#passcheck").html(
                "**length of your password must be between 3 and 10"
        );
        $("#passcheck").css("color", "red");
        passwordError = false;
        return false;
        } else {
        $("#passcheck").hide();
        }
}

// Validate Confirm Password
$("#conpasscheck").hide();
let confirmPasswordError = true;
$("#conpassword").keyup(function () {
        validateConfirmPassword();
});
```

```
function validateConfirmPassword() {
        let confirmPasswordValue = $("#conpassword").val();
        let passwordValue = $("#password").val();
        if (passwordValue != confirmPasswordValue) {
        $("#conpasscheck").show();
        $("#conpasscheck").html("**Password didn't Match");
        $("#conpasscheck").css("color", "red");
        confirmPasswordError = false;
        return false;
        } else {
        $("#conpasscheck").hide();
        }
}

// Submit button
$("#submitbtn").click(function () {
        validateUsername();
        validatePassword();
        validateConfirmPassword();
        validateEmail();
        if (
        usernameError == true &&
        passwordError == true &&
        confirmPasswordError == true &&
        emailError == true
        ) {
        return true;
        } else {
        return false;
        }
});
});
```

## //OUTPUT:



FORM VALIDATION USING JQUERY

Username:

st

**length of username must be between 3 and 10

Email:

stgmail.com

Your email must be a valid email

Password:

••

**length of your password must be between 3 and 10

Confirm Password:

••

**Password didn't Match

## LAB 6

## TITLE: ROLE BASED AUTHORIZATION

**//CODE:**

**1. HomeController.cs**

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using prjAspCoreMVCSecurity.Models;
using System.Diagnostics;

namespace prjAspCoreMVCSecurity.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }
        [Authorize(Roles = "Admin, User")]
        public IActionResult Index()
        {
            return View();
        }
        [Authorize(Roles = "Admin")]
        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}
```

**2. Program.cs**

```
using Microsoft.AspNetCore.Authentication.Cookies;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();
builder.Services.Configure < CookiePolicyOptions > (options =>
{
    options.CheckConsentNeeded = context => true;
    options.MinimumSameSitePolicy = SameSiteMode.None;
});
builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme).AddCookie();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}
app.UseCookiePolicy();
app.UseAuthentication();
app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Login}/{action=Index}/{id?}");

app.Run();
```

3. Index.cshtml

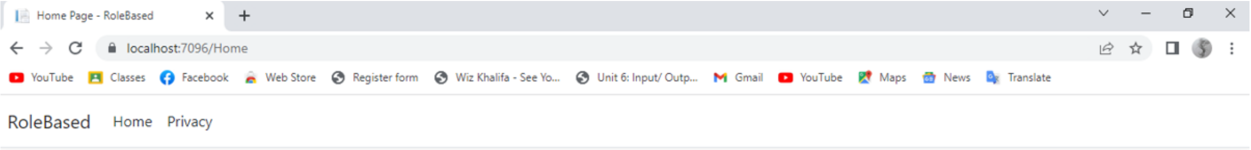
```
<form asp-action = "Index" method = "post">
    User Id <input type = "text" name = "userid" class = "form-control" />
    Password <input type = "text" name = "password" class = "form-control" />
    <input type = "submit" value = "login" />
</form>
```

**4. LoginController.cs**

```
using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.AspNetCore.Mvc;
using System.Security.Claims;

namespace prjAspCoreMVCSecurity.Controllers
{
```

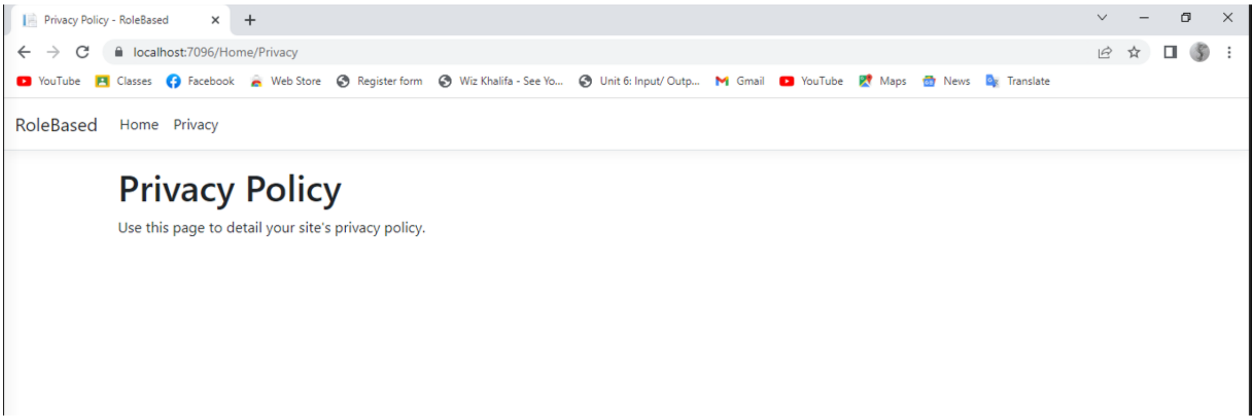
```
public class LoginController : Controller
{
    public IActionResult Index()
    {
        return View();
    }

    [HttpPost]
    public IActionResult Index(string userid, string password)
    {
        if (userid == null || password == "")
            return View();

        ClaimsIdentity identity = null;
        bool IsAuthenticate = false;
        if (userid =="admin" && password =="admin")
        {
            identity = new ClaimsIdentity(new[]
            {
                new Claim(ClaimTypes.Name, userid),
                new Claim(ClaimTypes.Role, "Admin"),

            }, CookieAuthenticationDefaults.AuthenticationScheme);
            IsAuthenticate = true;
        }

        if (userid == "user" && password == "user")
        {
            identity = new ClaimsIdentity(new[]
            {
                new Claim(ClaimTypes.Name, userid),
                new Claim(ClaimTypes.Role, "User"),
                }, CookieAuthenticationDefaults.AuthenticationScheme);
                IsAuthenticate = true;
            }
            if(IsAuthenticate)
            {
                var principal = new ClaimsPrincipal(identity);
                var login =
HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, principal);
                return RedirectToAction("Index", "Home");
            }
            return View();
        }
    }
}
```

//OUTPUT



Welcome

Learn about building Web apps with ASP.NET Core.



Privacy Policy

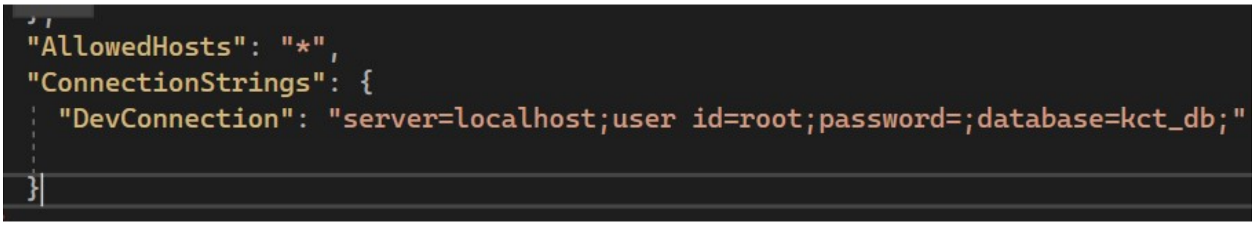
Use this page to detail your site's privacy policy.

# LAB 7

## TITLE: Take student information (rollno, firstname, lastname) of database and select from database using ASP.NET Core MVC Entity Framework

**// code**

### appsettings.json



```
},
"AllowedHosts": "*",
"ConnectionStrings": {
  "DevConnection": "server=localhost;user id=root;password=;database=kct_db;"

}
```

### DataDbContext.cs

```
using Microsoft.EntityFrameworkCore; using
prjAspcoreMVC.Models;

namespace prjASPCoreMVC.Models
{
    public class DataDbContext : DbContext
    {
        public DataDbContext(DbContextOptions options) : base(options)
        { }
        public DbSet<StudentInfo> student { get; set; }
    }
}
```

### Program.cs

```
using prjASPCoreMVC.Models; using
Microsoft.EntityFrameworkCore;

builder.Services.AddDbContext<DataDbContext>(option =>
{
    option.UseMySQL(builder.Configuration.GetConnectionString("DevConnection"));
});
```

### Model (studentInfo.cs)

```
using System.ComponentModel.DataAnnotations;


namespace prjAspcoreMVC.Models

{

    public class StudentInfo

    {

        [Key]

        public int id { get; set; } public
        string firstname { get; set; } public
        string lastname { get; set; }

    }

}
```

**View (Create.cshtml)**

```
@model prjAspcoreMVC.Models.StudentInfo

<form asp-action="Create">

    Firstname<input asp-for="firstname">

    Lastname<input asp-for="lastname">

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

</form>
```

**(List.cshtml)**

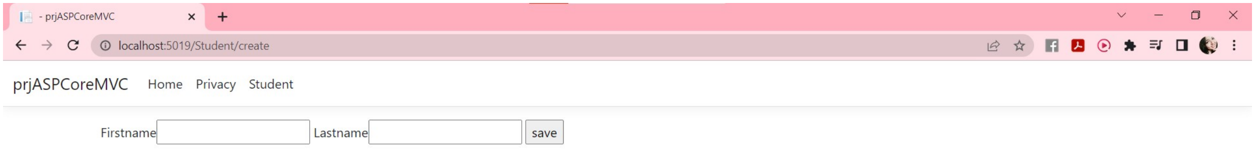
```
@model List<prjAspcoreMVC.Models.StudentInfo>

@foreach(var v in Model)

{

    @v.id

    @v.firstname;

    @v.lastname;

}
```

**Controller(StudentController.cs)**

```
using Microsoft.AspNetCore.Mvc;
using prjAspcoreMVC.Models;
using prjASPCoreMVC.Models;
```

```
        }

namespace prjASPCoreMVC.Controllers

{

    public class StudentController : Controller

    {

        DataDbContext db; public
        StudentController(DataDbContext _db)

        {

            db = _db;


        }


        public IActionResult Index()

        {

            return View();

        }


        public IActionResult List()

        {

            List<StudentInfo> students = db.student.ToList(); return
            View(students);

        }

        [HttpGet]

        public IActionResult Create()

        {

            return View();

        }

        [HttpPost] public IActionResult
        Create(StudentInfo info)

        {

            db.student.Add(info);
            db.SaveChanges();
            return View();

        }

     }

}
```

## //Output



Firstname □ Lastname □ save

LAB: 8

## TITLE: Make the CRUD operation for student in ASP.Net Core Web Application

**//source code**

### appsettings.json

```
"AllowedHosts": "*",
  "ConnectionStrings": {
    "Dev_Connection": "server=localhost;user id=root;password=;database=kct_db;"
  }
}
```

**DataContext.cs** using Microsoft.EntityFrameworkCore; using prjAspcoreMVC.Models;

```
namespace prj2ASPcoreWebApp.Models
{
    public class DataContext : DbContext
    {
        public DataContext(DbContextOptions options) : base(options)
        {

        }
        public DbSet<StudentInfo> student { get; set; }
    }

}
```

### Program.cs

```
using
prjASPCoreMVC.Models
; using
Microsoft.EntityFramewo
rkCore;
```

```
builder.Services.AddDbContext<DataDbContext>(option =>
{
    option.UseMySQL(builder.Configuration.GetConnectionString("DevConnection"));
});
```

**StudentInfo.cs** using

```
System.ComponentModel.DataAnnotations;

namespace prjAspcoreMVC.Models
{
    public class StudentInfo
    {
        [Key]
        public int id { get; set; } public
    string firstname { get; set; } public
    string lastname { get; set; } }

}
```

**Create.cshtml**

```
@page
@model prj2ASPcoreWebApp.Pages.Student.CreateModel
@{
}
<a a asp-page="list">Go To List</a>
<h1>Create Student </h1>
<form method="post">
    Firstname<input type="text" asp-for="info.firstname" class="form-control">
    Lastname<input type="text" asp-for="info.lastname" class="form-control">
    <input type="submit" value="save" class="btn-primary" />

</form>
```

**Create.cshtml.cs**

```
using Microsoft.AspNetCore.Mvc; using
Microsoft.AspNetCore.Mvc.RazorPages; using
prj2ASPcoreWebApp.Models; using
prjAspcoreMVC.Models;
```

```
namespace prj2ASPcoreWebApp.Pages.Student
{
    public class CreateModel : PageModel
    {
        public void OnGet()
        {
        }
        private DataContext db;
        public CreateModel(DataContext _db)
        {
            db = _db;
        }
        [BindProperty] public StudentInfo
        info { get; set; } public
        IActionResult OnPost()
        {
            db.student.Add(info); db.SaveChanges();
            HttpContext.Session.SetString("Message", "Saved Successfully"); return
            RedirectToPage("List");


        }

    }
}
```

**List.cshtml**

```
@page
@model prj2ASPcoreWebApp.Pages.Student.ListModel
@{
}

<h2>@Model.Message</h2>

<a asp-page="Create">Go To Create</a>

<table class="table">
    <tr>
```

```
        <th>Firstname</th>

        <th>Lastname</th>

        <th>Action</th>

    </tr>

    @foreach(var item in Model.list)

    {

        <tr>

            <td>@item.firstname</td>

            <td>@item.lastname</td>

            <td>

                <a asp-route-id="@item.id" asp-page="Delete" class="btn btn-
danger">Delete</a>

                    <a asp-route-id="@item.id" asp-page="Edit" class="btn btn-primary">Edit</a>
            </td>

        </tr>

    }

</table>
```

**List.cshtml.cs**

```
using Microsoft.AspNetCore.Mvc; using
Microsoft.AspNetCore.Mvc.RazorPages; using
prj2ASPcoreWebApp.Models; using
prjAspcoreMVC.Models;

namespace prj2ASPcoreWebApp.Pages.Student

{

    public class ListModel : PageModel

    {

        public DataContext db;

        public ListModel(DataContext _db)

        {

            db = _db;

        }
        public string Message = "";

        public List<StudentInfo> list { get; set; } public
        void OnGet()
        {
            Message = HttpContext.Session.GetString("Message"); if(db.student !=
            null)
            { list = db.student.ToList();

            }
        }
    }
}
```

**__Edit.cshtml__**

```
@page
@model prj2ASPcoreWebApp.Pages.Student.EditModel
@{
}
<h1>Update Student </h1>
<form method="post">
    Firstname<input type="text" asp-for="info.firstname" class="form-control">
    Lastname<input type="text" asp-for="info.lastname" class="form-control">
    <input type="submit" value="Update" class="btn-primary" />

</form>
```

**__Edit.cshtml.cs__**

```
_using Microsoft.AspNetCore.Mvc; using
Microsoft.AspNetCore.Mvc.RazorPages; using
prj2ASPcoreWebApp.Models; using
prjAspcoreMVC.Models;

namespace prj2ASPcoreWebApp.Pages.Student
{
    public class EditModel : PageModel
    {
        public DataContext db; public
        EditModel(DataContext _db)

        {

            db = _db;

        }

        [BindProperty] public StudentInfo
        info { get; set; }


        public void OnGet(int id)

        { info = db.student.Find(id);

        }

        public IActionResult OnPost(int id)

        {

            StudentInfo s = db.student.Find(id); s.firstname
            = info.firstname;

            s.lastname = info.lastname;
            db.SaveChanges(); return
            RedirectToPage("List");

        }

    }

}
```

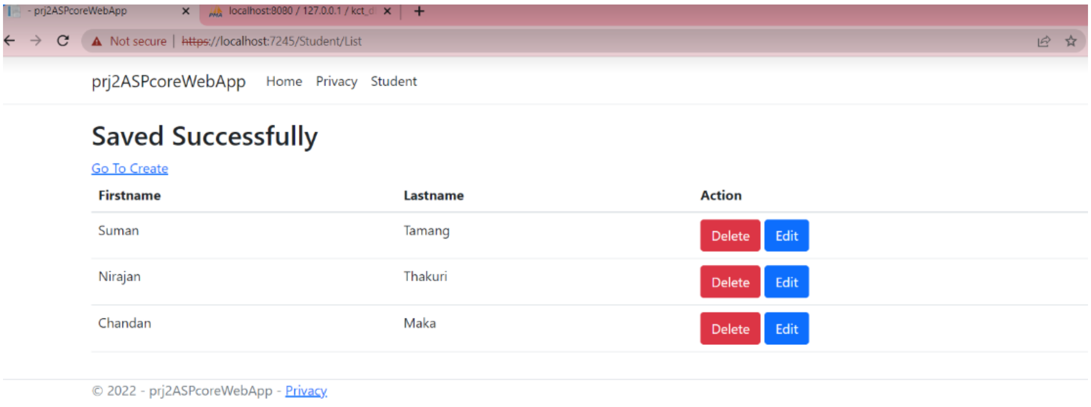
**Delete.cshtml**

```
@page

@model prj2ASPcoreWebApp.Pages.Student.DeleteModel

@{

}
```

**Delete.cshtml.cs**

```
using Microsoft.AspNetCore.Mvc; using
Microsoft.AspNetCore.Mvc.RazorPages; using
prj2ASPcoreWebApp.Models;


namespace prj2ASPcoreWebApp.Pages.Student

{

    public class DeleteModel : PageModel

    {
        private DataContext db; public
        DeleteModel(DataContext _db)

        {

            db = _db;

        }

        public IActionResult OnGet(int id)

        {

            db.student.Remove(db.student.Find(id));
            db.SaveChanges(); return
            RedirectToPage("list");

        }

    }

}
```

# Output



- prj2ASPcoreWebApp
localhost:8080 / 127.0.0.1 / kct_d
Not secure | https://localhost:7245/Student/List

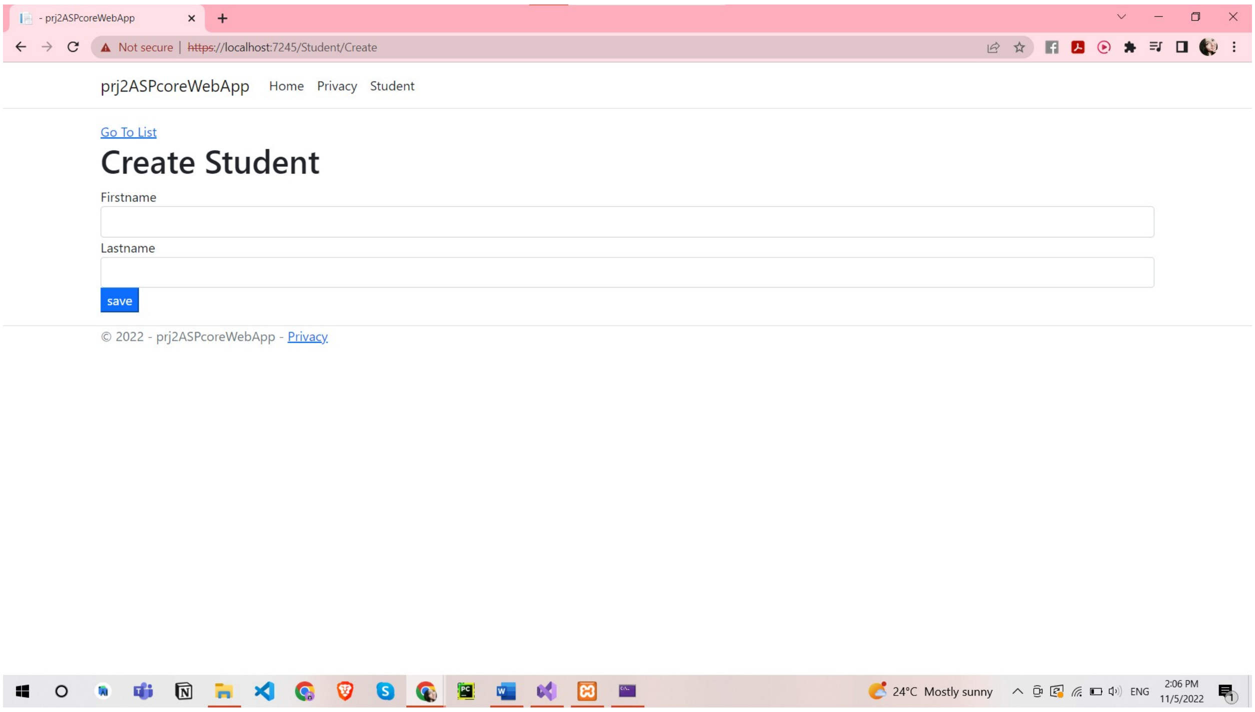
prj2ASPcoreWebApp Home Privacy Student

## Saved Successfully

Go To Create

| Firstname | Lastname | Action |
|---|---|---|
| Suman | Tamang | Delete Edit |
| Nirajan | Thakuri | Delete Edit |
| Chandan | Maka | Delete Edit |

© 2022 - prj2ASPcoreWebApp - Privacy



prj2ASPcoreWebApp Home Privacy Student

Go To List

## Create Student

Firstname

Lastname

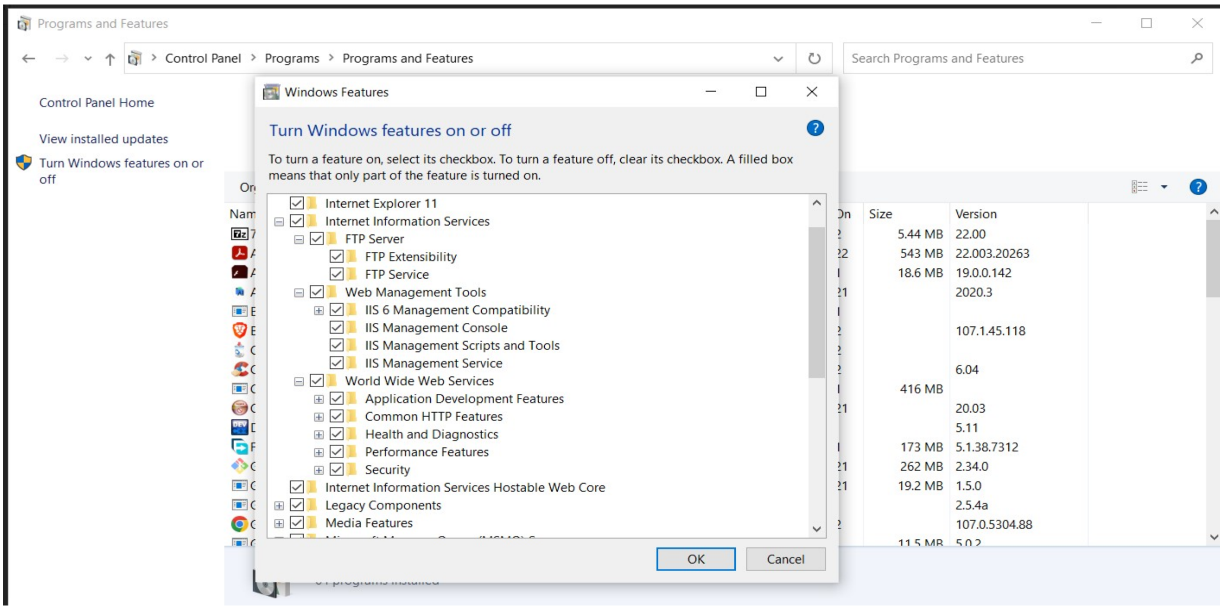
save

© 2022 - prj2ASPcoreWebApp - Privacy

# LAB 9
# TITLE : HOSTING AND DEPLOYING ASP.NET CORE APPLICATION IN IIS SERVER

## //source code

1. We turn-on IIS: Control Panel>Programs>Programs and Features>Turn Windows features on or off



Programs and Features
Control Panel > Programs > Programs and Features
Search Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Size
Version

Windows Features

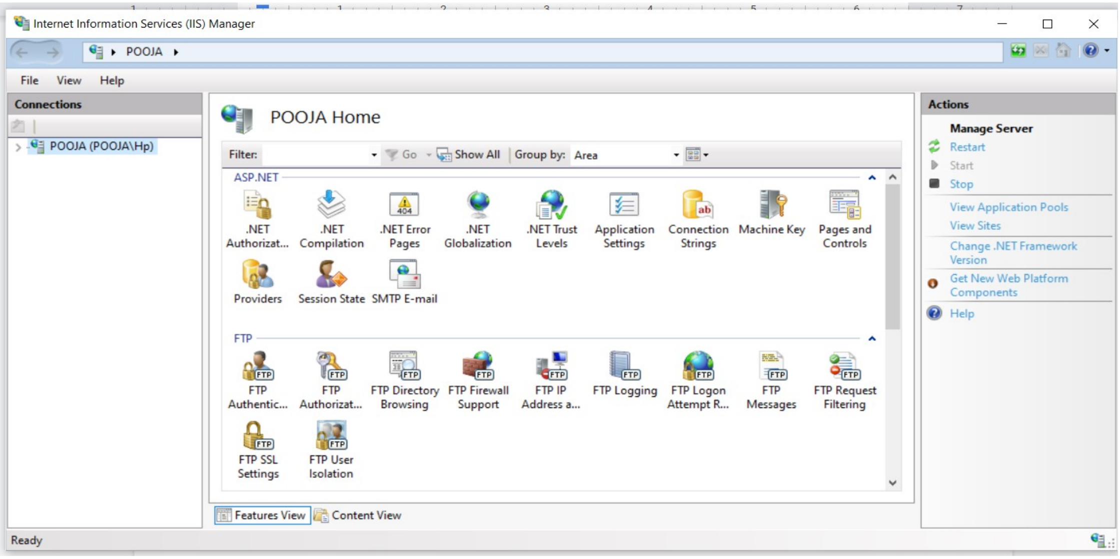
Turn Windows features on or off

To turn a feature on, select its checkbox. To turn a feature off, clear its checkbox. A filled box means that only part of the feature is turned on.

- ☑ Internet Explorer 11
- ☑ Internet Information Services
  - ☑ FTP Server
    - ☑ FTP Extensibility
    - ☑ FTP Service
  - ☑ Web Management Tools
    - ☑ IIS 6 Management Compatibility
    - ☑ IIS Management Console
    - ☑ IIS Management Scripts and Tools
    - ☑ IIS Management Service
  - ☑ World Wide Web Services
    - ☑ Application Development Features
    - ☑ Common HTTP Features
    - ☑ Health and Diagnostics
    - ☑ Performance Features
    - ☑ Security
- ☑ Internet Information Services Hostable Web Core
- ☑ Legacy Components
- ☑ Media Features

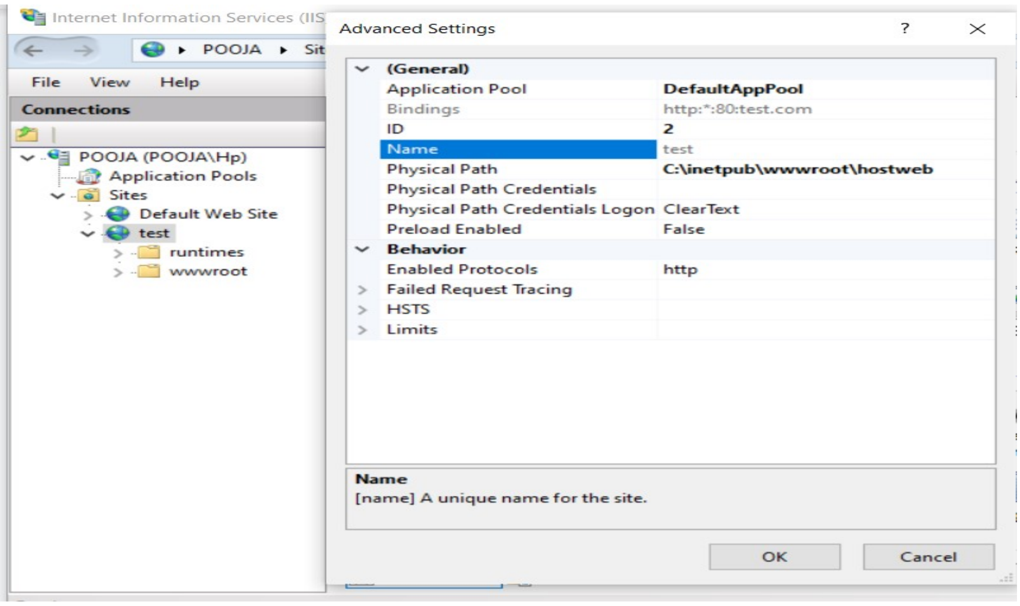
OK Cancel

2. After restarting our device, we Press Win+R and type “inetmgr” or Search IIS Manager



3. Right click on sites and choose “Add Website” ● Fill the site name and host name ● Set physical path as :

   C:>inetpub>wwwroot>hostweb(created_folder_name)



Advanced Settings

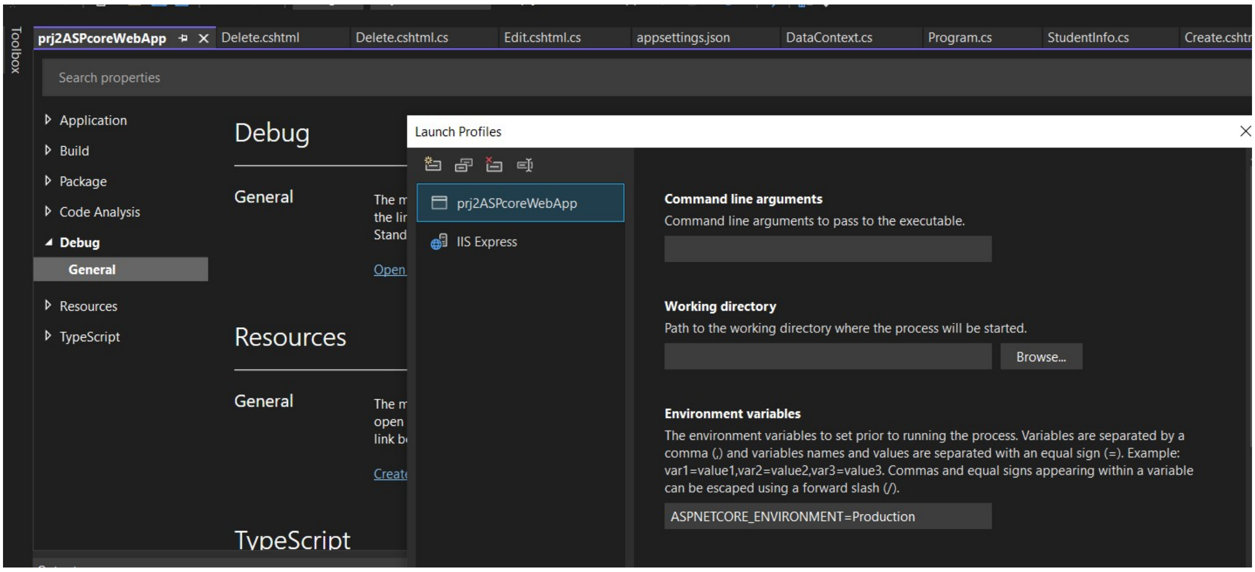
| | |
|---|---|
| **(General)** | |
| Application Pool | **DefaultAppPool** |
| Bindings | http:*:80:test.com |
| ID | **2** |
| Name | test |
| Physical Path | **C:\inetpub\wwwroot\hostweb** |
| Physical Path Credentials | |
| Physical Path Credentials Logon | ClearText |
| Preload Enabled | False |
| **Behavior** | |
| Enabled Protocols | http |
| Failed Request Tracing | |
| HSTS | |
| Limits | |

**Name**
[name] A unique name for the site.

4. We opened our project in visual studio and check build error:
   Goto Build>Build Solution….

   If Build Succeeded, then OK.

5. Right click our main project and choose properties>Debug>Open debug launch profiles UI>IIs Express>Environment variables: ASPNETCORE_ENVIRONMENT=Production



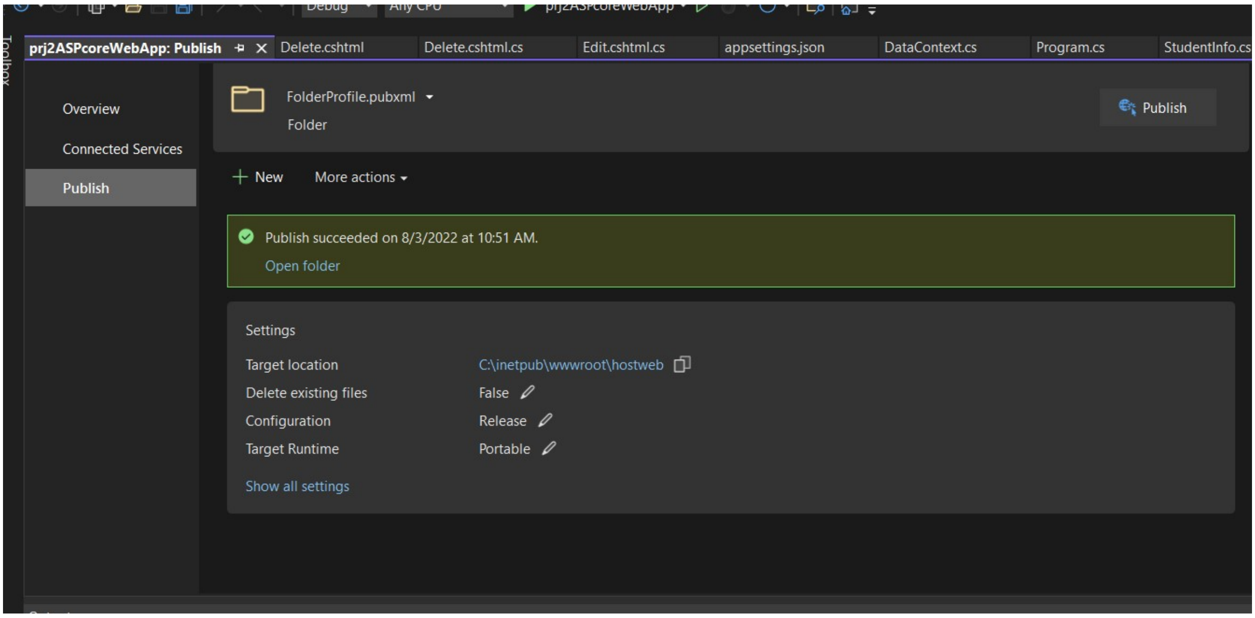
**Command line arguments**
Command line arguments to pass to the executable.

**Working directory**
Path to the working directory where the process will be started.

**Environment variables**
The environment variables to set prior to running the process. Variables are separated by a comma (,) and variables names and values are separated with an equal sign (=). Example: var1=value1,var2=value2,var3=value3. Commas and equal signs appearing within a variable can be escaped using a forward slash (/).

ASPNETCORE_ENVIRONMENT=Production

6. Right click on our project and choose publish



FolderProfile.pubxml
Folder

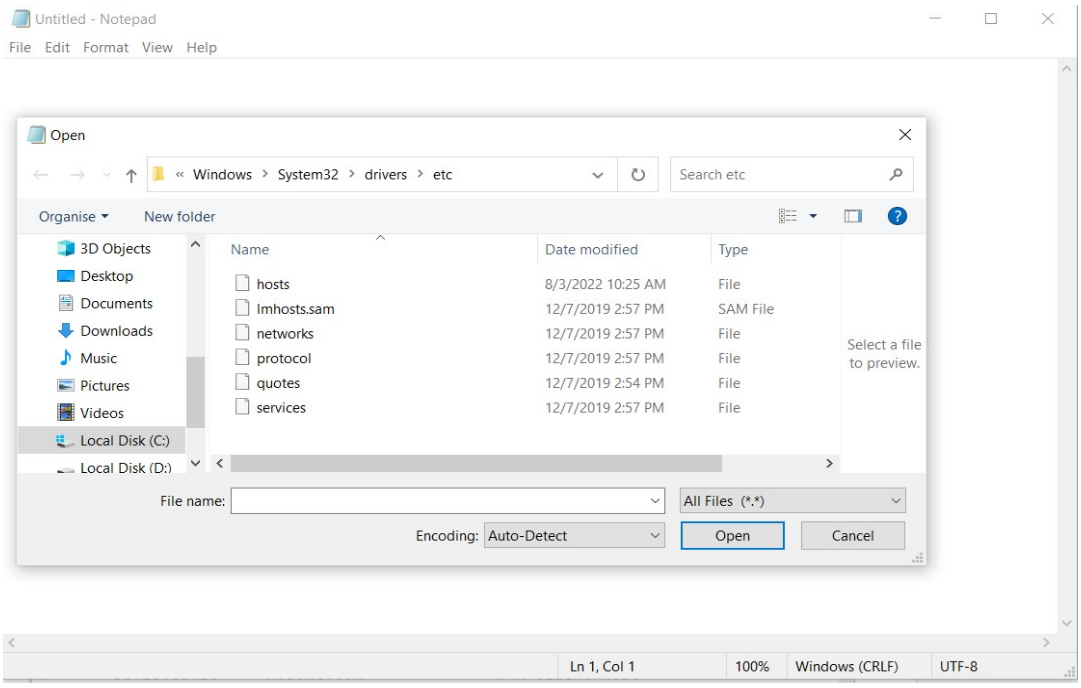
Publish succeeded on 8/3/2022 at 10:51 AM.
Open folder

Settings

| | |
|---|---|
| Target location | C:\inetpub\wwwroot\hostweb |
| Delete existing files | False |
| Configuration | Release |
| Target Runtime | Portable |

Show all settings

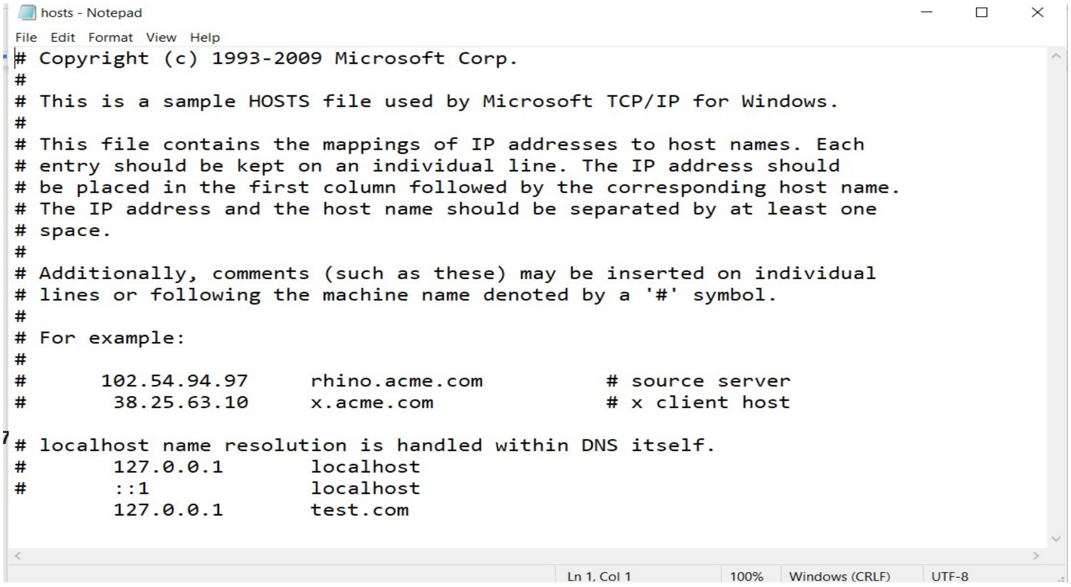
7. Configure domain name in Window OS



| Name | Date modified | Type |
|---|---|---|
| hosts | 8/3/2022 10:25 AM | File |
| lmhosts.sam | 12/7/2019 2:57 PM | SAM File |
| networks | 12/7/2019 2:57 PM | File |
| protocol | 12/7/2019 2:57 PM | File |
| quotes | 12/7/2019 2:54 PM | File |
| services | 12/7/2019 2:57 PM | File |

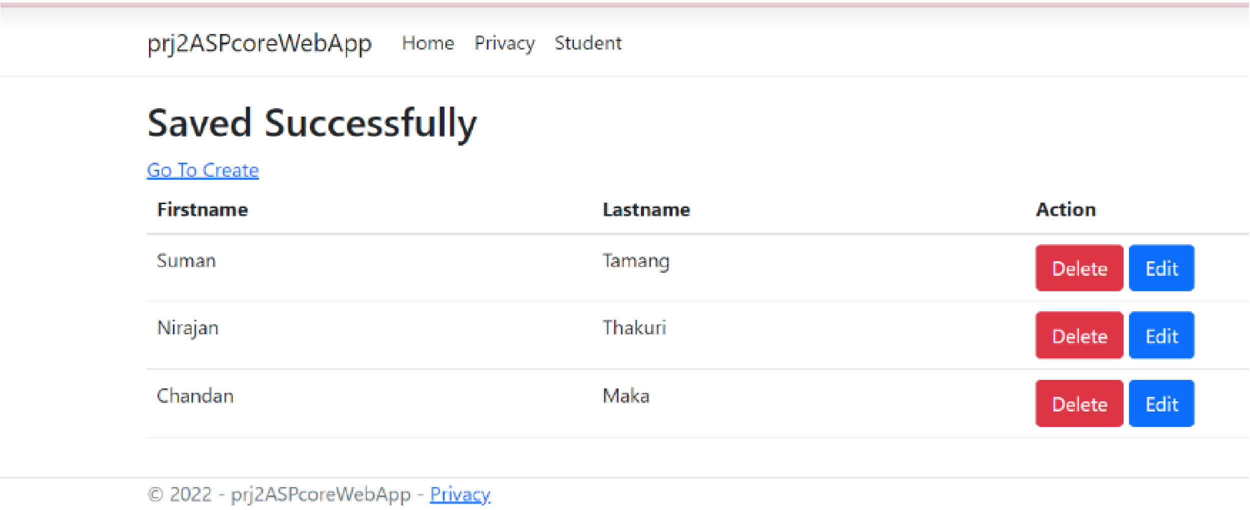
File name:
All Files (*.*)
Encoding: Auto-Detect

8. Place 127.0.0.1 test.com



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
        127.0.0.1       test.com
```

9. Open browser and type URL “test.com”



Saved Successfully

Go To Create

| Firstname | Lastname | Action |
|---|---|---|
| Suman | Tamang | Delete Edit |
| Nirajan | Thakuri | Delete Edit |
| Chandan | Maka | Delete Edit |

© 2022 - prj2ASPcoreWebApp - Privacy

## Conclusion

Thus, the ASP.NET Core application is hosted and deployed in IIS server.