LOGINMVC:

User.cs

@model Authenication.Models.User

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Login</title>

<style>

#login-div {

position: absolute;

left: 40%;

top: 40%;

border: 1px solid black;

padding: 10px 10px;

}

.field-validation-error {

color: red;

}

body{

background-color:aqua;

}

</style>

</head>

<body>

<div id="login-div">

@using (Html.BeginForm("Authorise", "Login", FormMethod.Post))

{

<table>

<tr>

<td>

</td>

<td style="text-decoration:underline"> login</td>

</tr>

<tr>

<td>

@Html.LabelFor(model => model.username)

</td>

<td>

@Html.EditorFor(model => model.username)

</td>

</tr>

<tr>

<td>

</td>

<td>

@Html.ValidationMessageFor(model => model.username)

</td>

</tr>

<tr>

<td>

@Html.LabelFor(model => model.password)

</td>

<td>

@Html.EditorFor(model => model.password)

</td>

</tr>

<tr>

<td>

</td>

<td>

@Html.ValidationMessageFor(model => model.password)

</td>

</tr>

<tr>

<td colspan="2">

<label class="field-validation-error">@Html.DisplayFor(model => model.LoginErrorMsg)</label>

</td>

</tr>

<tr>

<td>

</td>

<td>

<input type="submit" value="Login" />

<input type="reset" value="Clear" />

</td>

</tr>

</table>

}

<script src="~/lib/jquery/dist/jquery.js"></script>

<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.js"></script>

</div>

</body>

</html>

-------------------------------------------------------------------------------------------

UserController.cs:

using LoginMVC.Models;

using LoginMVC.Services;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace LoginMVC.Controllers

{

public class UserController : Controller

{

public object Session { get; private set; }

/\* private readonly IRepo<User> \_repo;

public UserController(IRepo<User> repo)

{

\_repo = repo;

}\*/

public IActionResult Index()

{

return View();

}

[HttpPost]

public IActionResult Authorize(User user)

{

HttpContext context;

using (UserContext db = new UserContext())

{

try

{

var userDetails = db.users.Where(x => x.Id == user.Id && x.Password == user.Password).FirstOrDefault();

if(userDetails==null)

{

user.LoginErrorMessage = "Invalid username or password";

return View("Index", user);

}

else

{

//Session["userId"] = user.Id;

return RedirectToAction("Index", "Message");

}

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

return View();

}

}

}

Index.csHtml:

@model LoginMVC.Models.User

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Login</title>

<style>

#login-div{

position:absolute;

left:40%;

top:40%;

border:1px solid #ccc;

padding:10px 10px;

}

</style>

</head>

<body>

<div id="login-div">

@using(Html.BeginForm("Authorize", "User",FormMethod.Post))

{

<table>

<tr>

<td style="text-decoration:underline;text-align:center">Login</td>

</tr>

<tr>

<td>

@Html.LabelFor(Model => Model.Id)

@Html.EditorFor(Model => Model.Id)

</td>

</tr>

<tr>

<td>@Html.ValidationMessageFor(Model => Model.Id)</td>

</tr>

<tr>

<td>

@Html.LabelFor(Model => Model.Password)

@Html.EditorFor(Model => Model.Password)

</td>

</tr>

<tr>

<td>@Html.ValidationMessageFor(Model => Model.Password)</td>

</tr>

<tr>

<td>

<input type="submit" value="Login" />

</td>

</tr>

</table>

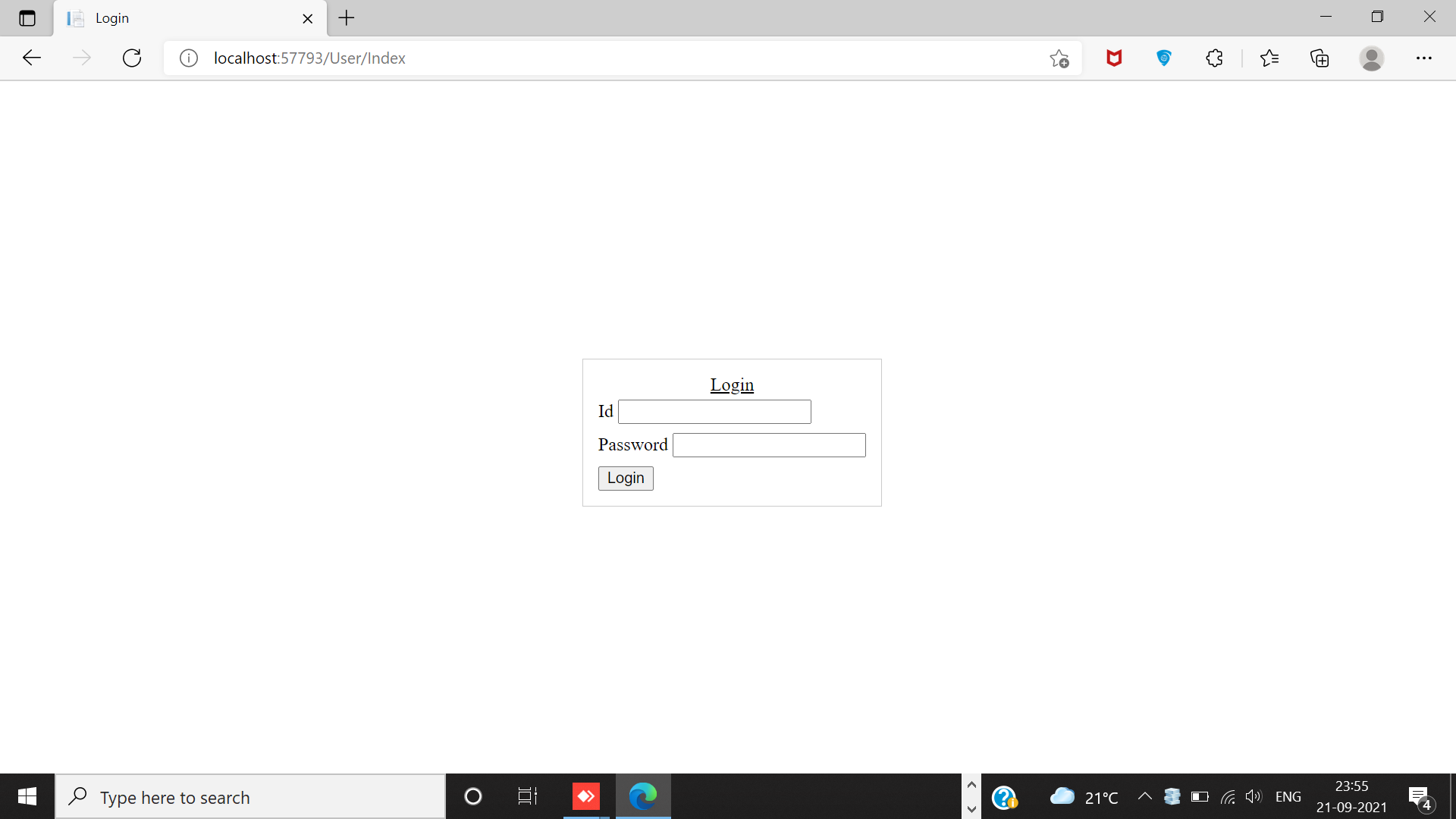
}

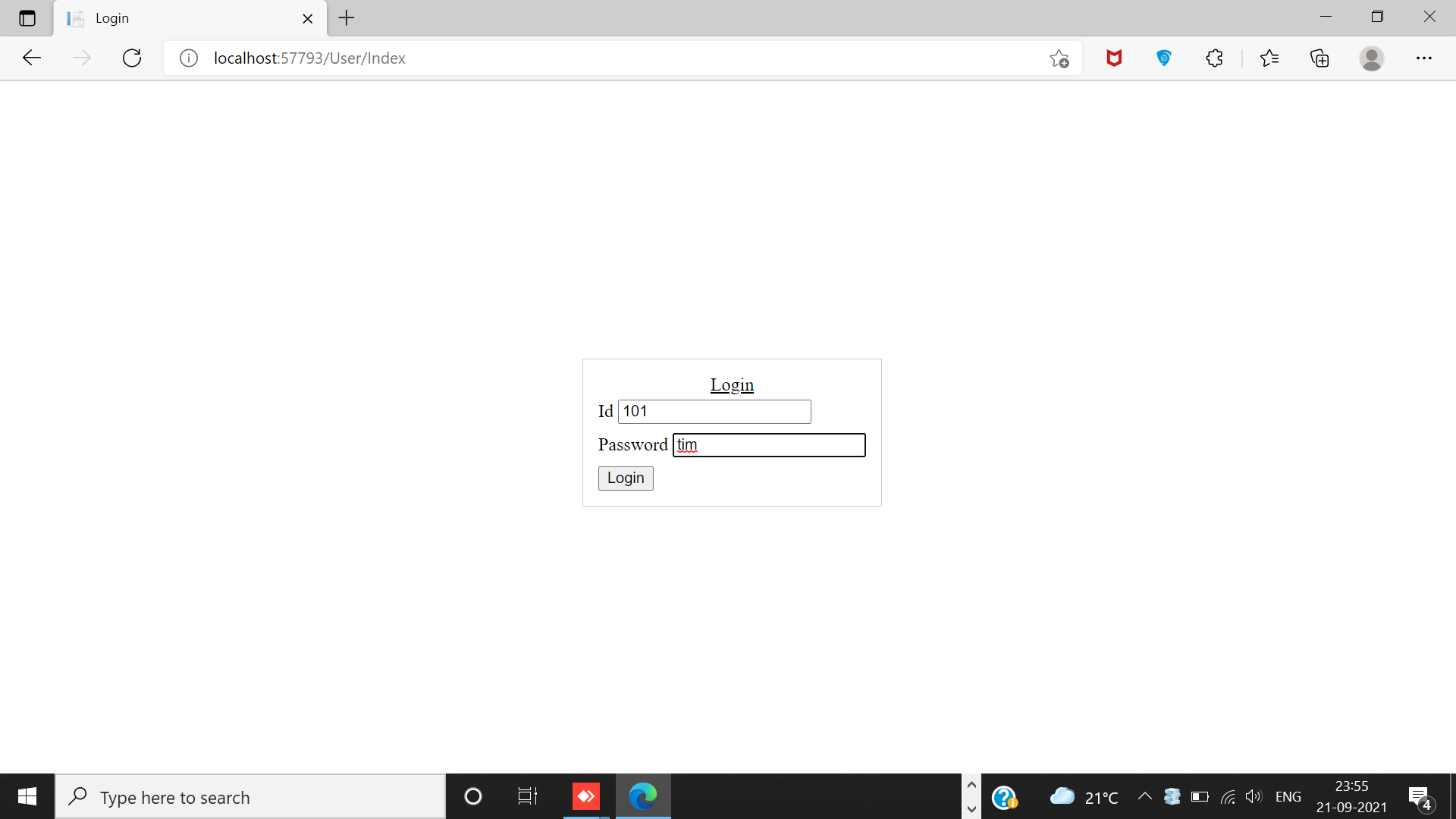
</div>

</body>

</html>

OUTPUT:





Goes to next page:

