

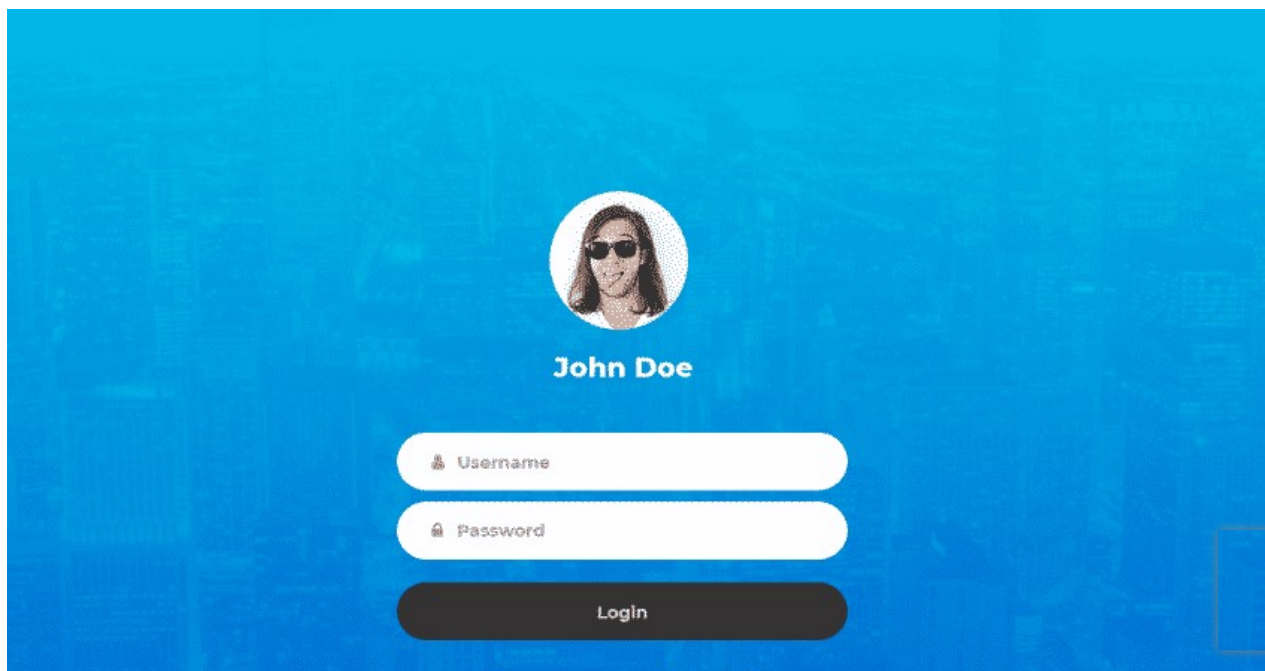


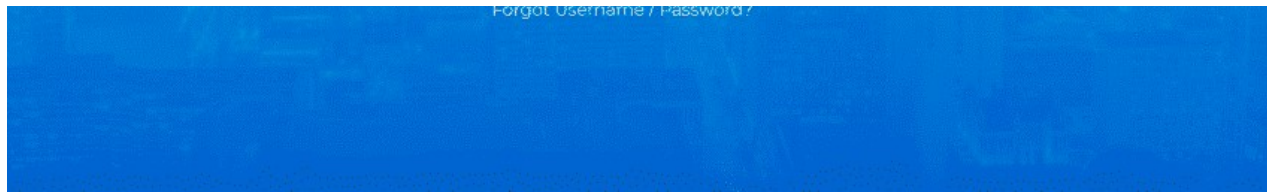
LOGIN PAGE IN ASP.NET CORE MVC WITH DATABASE

OCTOBER 31, 2018

Hey guys, In this tutorial I'm going to create a Login page in Asp.net MVC.

There are many Login pages designs are available, you can use of your choice. For this tutorial, I'm going to use [This](#) Beautiful Login page by colorlib.





Create Desktop, Web And

Pi With Xojo. Get Started Free Today! xojo.com

So, If you have not even created your **Asp.net Core** project then your first step is to create an **Empty project** using this command

```
dotnet new Web
```

& don't forget to create these three folders “**Controllers**”, “**Models**” & “**Views**” add this code to your “**Startup.cs**” to ready your project for using MVC

```
1.  using System;
2.  using System.Collections.Generic;
3.  using System.Linq;
4.  using System.Threading.Tasks;
5.  using Microsoft.AspNetCore.Builder;
6.  using Microsoft.AspNetCore.Hosting;
7.  using Microsoft.AspNetCore.Http;
8.  using Microsoft.Extensions.DependencyInjection;
9.
10. namespace AdminPanelTutorial
11. {
12.     public class Startup
13.     {
14.         public void ConfigureServices(IServiceCollection services)
15.         {
16.             services.AddMvc();
17.         }
18.
19.         // This method gets called by the runtime. Use this method to configure the
           HTTP request pipeline.
20.         public void Configure(IApplicationBuilder app, IHostingEnvironment env)
21.         {
22.             if (env.IsDevelopment())
23.             {
24.                 app.UseDeveloperExceptionPage();
```

```
25.         }
26.
27.     app.UseStaticFiles();
28.
29.     app.UseMvc(routes =>
30.     {
31.         routes.MapRoute(
32.             name: "default",
33.             template: "{controller=Home}/{action=Index}/{id?}");
34.     });
35.     }
36. }
37. }
```

Now you need to create models for your Database table.

Here's my table for users to login



id : int(11)
name : varchar(255)
email : varchar(255)
password : varchar(255)
phone : varchar(15)
created : timestamp

I'm using **Database First** approach to create Models from my existing Database, so If you have also created your Database & want to create models then I'll recommend you this Article to Scaffold your Database => [Asp.Net Core Scaffold DB](#)

Free Writing Tool

Improve grammar, word choice, and sentence structure in your writi

Now you need to download Login page Code from Colorlib. So download from here
=> [Login Form 12 by Colorlib](#)

unzip the folder and you'll find these folders & file





Copy all folders from here and paste in the “**wwwroot**” folder of your **Asp.net Core MVC** project.

Now create a Controller as “**AccountController.cs**” inside your Controllers Folder & Add this code.

```
1. using Microsoft.AspNetCore;
2. using Microsoft.AspNetCore.Hosting;
3. using Microsoft.AspNetCore.Mvc;
4.
5. namespace AdminPanelTutorial
6. {
7.     public class AccountController : Controller
8.     {
9.         public IActionResult Login()
10.        {
11.            return View();
12.        }
13.    }
14. }
```

Create a new Folder into Views as “**Account**” & inside this Folder create a file as “**Login.cshtml**” and put this code

```
1. @{
2.     Layout=null;
3. }
4.
5. <!DOCTYPE html>
6. <html lang="en">
7. <head>
8.     <title>Login V12</title>
9.     <meta charset="UTF-8">
10.    <meta name="viewport" content="width=device-width, initial-scale=1">
11.    <!--=====
12.    <link rel="icon" type="image/png" href="~/images/icons/favicon.ico"/>
13.    <!--=====
```

```

=====-->
14.     <link rel="stylesheet" type="text/css" href="~/vendor/bootstrap
/css/bootstrap.min.css">
15. <!--=====
=====-->
16.     <link rel="stylesheet" type="text/css" href="~/fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
17. <!--=====
=====-->
18.     <link rel="stylesheet" type="text/css" href="~/fonts/Linearicons-Free-v1.0.0
/icon-font.min.css">
19. <!--=====
=====-->
20.     <link rel="stylesheet" type="text/css" href="~/vendor/animate/animate.css">
21. <!--=====
=====-->
22.     <link rel="stylesheet" type="text/css" href="~/vendor/css-hamburgers
/hamburgers.min.css">
23. <!--=====
=====-->
24.     <link rel="stylesheet" type="text/css" href="~/vendor/select2/select2.min.css">
25. <!--=====
=====-->
26.     <link rel="stylesheet" type="text/css" href="~/css/util.css">
27.     <link rel="stylesheet" type="text/css" href="~/css/main.css">
28. <!--=====
=====-->
29. </head>
30. <body>
31.
32.     <div class="limiter">
33.         <div class="container-login100" style="background-image: url('../images
/img-01.jpg');">
34.             <div class="wrap-login100 p-t-50 p-b-0">
35.                 <form class="login100-form validate-form">
36.                     <div class="login100-form-avatar">
37.                         
38.                     </div>
39.
40.                     <span class="login100-form-title p-t-20 p-b-45">
41.                         John Doe
42.                     </span>
43.
44.                     <div class="wrap-input100 validate-input m-b-10" data-validate = "Email
is required">
45.                         <input class="input100" type="text" name="email" id="email"
placeholder="Email">
46.                         <span class="focus-input100"></span>
47.                         <span class="symbol-input100">
48.                             <i class="fa fa-user"></i>
49.                         </span>
50.                     </div>

```

```

51.
52.         <div class="wrap-input100 validate-input m-b-10" data-validate =
"Password is required">
53.             <input class="input100" type="password" name="password" id="password"
placeholder="Password">
54.             <span class="focus-input100"></span>
55.             <span class="symbol-input100">
56.                 <i class="fa fa-lock"></i>
57.             </span>
58.         </div>
59.
60.         <div class="container-login100-form-btn p-t-10">
61.             <input type="button" value="Login" class="login100-form-btn"
onclick="Validate()">
62.         </div>
63.
64.         <div class="text-center w-full p-t-25 p-b-80">
65.             <a href="#" class="txt1">
66.                 Forgot Username / Password?
67.             </a>
68.         </div>
69.
70.     </form>
71. </div>
72. </div>
73. </div>
74.
75. <!--=====
=====-->
76.     <script src="~/vendor/jquery/jquery-3.2.1.min.js"></script>
77. <!--=====
=====-->
78.     <script src="~/vendor/bootstrap/js/popper.js"></script>
79.     <script src="~/vendor/bootstrap/js/bootstrap.min.js"></script>
80. <!--=====
=====-->
81.     <script src="~/vendor/select2/select2.min.js"></script>
82. <!--=====
=====-->
83.     <script src="~/js/main.js"></script>
84.
85. </body>
86. </html>

```

Run your project & go to <http://localhost:8080/Account/Login> (your port can be different)

Now you need a method to validate user from Database.

In your “**AccountController.cs**”, Add this Validation method

```
1. public ActionResult Validate(Admins admin)
2. {
3.     var _admin = db.Admins.Where(s => s.Email == admin.Email);
4.     if(_admin.Any()){
5.
6.         if(_admin.Where(s => s.Password == admin.Password).Any()){
7.
8.             return Json(new { status = true, message = "Login Successfull!"});
9.         }
10.        else
11.        {
12.            return Json(new { status = false, message = "Invalid Password!"});
13.        }
14.    }
15.    else
16.    {
17.        return Json(new { status = false, message = "Invalid Email!"});
18.    }
19. }
```

We'll use ajax to validate our user from Login Page.

So the last thing is our ajax call. Put this Code at the End of your body

```
1. <script>
2.
3.     function Validate() {
4.
5.         $.ajax(
6.         {
7.             type: "POST",
8.             url: '@Url.Action("Validate", "Account")',
9.             data: {
10.                 email: $('#email').val(),
11.                 password: $('#password').val()
12.             },
13.             error: function (result) {
14.                 alert("There is a Problem, Try Again!");
15.             },
16.             success: function (result) {
17.                 console.log(result);
18.                 if (result.status == true) {
```

```
19.         window.location.href = '@Url.Action("Index", "Home")';
20.     }
21.     else {
22.         alert(result.message);
23.     }
24. }
25.
26. });
27. }
28.
29. </script>
```

Now run your project & Login with the email & password you have in your Admins Table.

I have saved Password without any Hashing just for keeping it simple. But keep in mind that this is really a bad practice. You should never save the password in your Database without hashing.

If you want to Hash password then follow this article => [C# Hashing algorithm class](#)

Or If you find any problem in this Tutorial, please do comment below & stay connected for the next part of this Tutorial.

[← Setup Admin Template](#)

[CRUD Operation →](#)



AUTHOR

SHEHRYAR KHAN

I'm passionate about learning new technologies as well as mentoring and helping others get started with their programming career. This blog is my way of giving back to the Community.



RELATED POSTS

**MULTI-THREADED TCP
SERVER USING DOTNET
CORE EXAMPLE | C#**

JUNE 3, 2019

Status	Latest release
Preview	3.0.0-preview5
Current	2.2.5
LTS	2.1.11
End of life	2.0.9
LTS	1.1.13
LTS	1.0.16

**SELECTING DOTNET CORE
SDK VERSION WHILE
CREATING A NEW
PROJECT**

MAY 23, 2019

**THE ULTIMATE GUIDE TO
THE BEST .NET CORE
E-COMMERCE OPEN
SOURCE PLATFORMS**

MAY 12, 2019

17 COMMENTS

**SOPRONO**

1 MONTH AGO

[↩ REPLY](#)

Is this source code uploaded anywhere? I see your Git repo does not have any repository with this AccountController in it?

**CHARLIE**

2 MONTHS AGO

[↩ REPLY](#)

In the login page, it does a call to Account controller, it connects to Database and return true, but just return to login page again, it only reload login page, not going to Home-Index. Whats wrong? Please help me. Thanks.

SHEHRYAR KHAN

POST AUTHOR





2 MONTHS AGO

Yeah, that was a mistake. Thanks for your comment.

**TALIS QUALIS**

3 MONTHS AGO

[↩ REPLY](#)

Thank you for tutorial! Great work!

In your "AccountController.cs" line 12 there should be "false" I think:
return Json(new { status = false, message = "Invalid Password!"});

**ANDRÉ**

5 MONTHS AGO

[↩ REPLY](#)

Having problem with ajax request on validade
Alwais sending There is a Problem, Try Again!

**MASOOM**

6 MONTHS AGO

[↩ REPLY](#)

Hi Great Work !! but i need a prob to connectivity in db 😞

**SHEHRYAR KHAN**

POST AUTHOR

6 MONTHS AGO

[↩ REPLY](#)

If you have downloaded my code from Github then you need to update
Database credentials in MyDbContext.cs File.

**JENNY**[↩ REPLY](#)

6 MONTHS AGO

Thank you for your tutorial! I'm following the whole tutorial from the beginning article in order to create a dashboard in ASP.net.

However I'm running into problems in this section when I need to create models of my DB.

How did you import the .sql file to your solution and what links should be established?

The second problem is that it seems not possible to create models of a db in Visual Studio 2017?

**SHEHRYAR KHAN**

POST AUTHOR

[↩ REPLY](#)

6 MONTHS AGO

It's possible to models of a DB in VS2017 in many ways. you can also use MySQL WorkBench to import .sql file & create models in your project using few commands following this tutorial. <https://codinginfinite.com/entity-framework-core-database-first-net-core-scaffold-mysql-db/>

**ALERON**[↩ REPLY](#)

6 MONTHS AGO

Do we have to add a class below account, because there isn't any `@Url.Action("Validate", "Account")`,

SHEHRYAR KHAN

POST AUTHOR



6 MONTHS AGO

Please check carefully. At the end of this Article, I'm adding "Validate" method in my "AccountController.cs" Class.

**MORRIS SMITH**

7 MONTHS AGO

Login page will not work

[↩ REPLY](#)**SHEHRYAR KHAN**

POST AUTHOR

7 MONTHS AGO

What is the problem you're facing? please share.

[↩ REPLY](#)**MORRIS**

7 MONTHS AGO

In the account controller, do we need to add `MyDBContext = new MyDBContext();`
Can't validate my user

[↩ REPLY](#)**SHEHRYAR KHAN**

POST AUTHOR

7 MONTHS AGO

yes, you need to create Models of your Database and use Database Context to Validate User.

[↩ REPLY](#)

**SHAYAN**

8 MONTHS AGO

[↩ REPLY](#)

bro its really helpful. thanks

**JACK**

8 MONTHS AGO

[↩ REPLY](#)

Thanks for sharing. This is much simpler than configuring the IdentityUser.

WRITE A COMMENT

Name

Email

Website

Enter your comment here..

☐ Save my name, email, and website in this browser for the next time I comment.☒ Notify me when new comments are added.**POST COMMENT**



1.6K Facebook



Twitter



YouTube



Instagram



#1 Backup and Recovery

Free powerful tool by Veeam
Replicate VMs, schedule b
restore to AWS and more!

Veeam Software

SUBSCRIBE

Subscribe to our newsletters to get an Email on every new Article

Email Address *

Choose Technology *

SUBSCRIBE





TOP ARTICLES



SIGNUP LOGIN PAGE IN PHP WITH DATABASE MYSQL SOURCE CODE

MAY 27, 2018



CRUD OPERATIONS IN ASP.NET CORE MVC

NOVEMBER 10, 2018



HERE ARE THE TEN BEST PROGRAMMING LANGUAGES TO LEARN IN 2019

DECEMBER 22, 2018



OPEN SOURCE BULK SMS SENDER ANDROID APP

APRIL 16, 2018



CAR LOCATION TRACKING ANDROID APP WITH FIREBASE TUTORIAL

AUGUST 28, 2018



LOGIN PAGE IN ASP.NET CORE MVC WITH DATABASE

OCTOBER 31, 2018



CAR LOCATION TRACKING ANDROID APP WITH FIREBASE TUTORIAL | PART 2

AUGUST 28, 2018



© 2018 CODING INFINITE - ALL RIGHTS RESERVED

[^ TOP](#)

