

LDAP Authentication



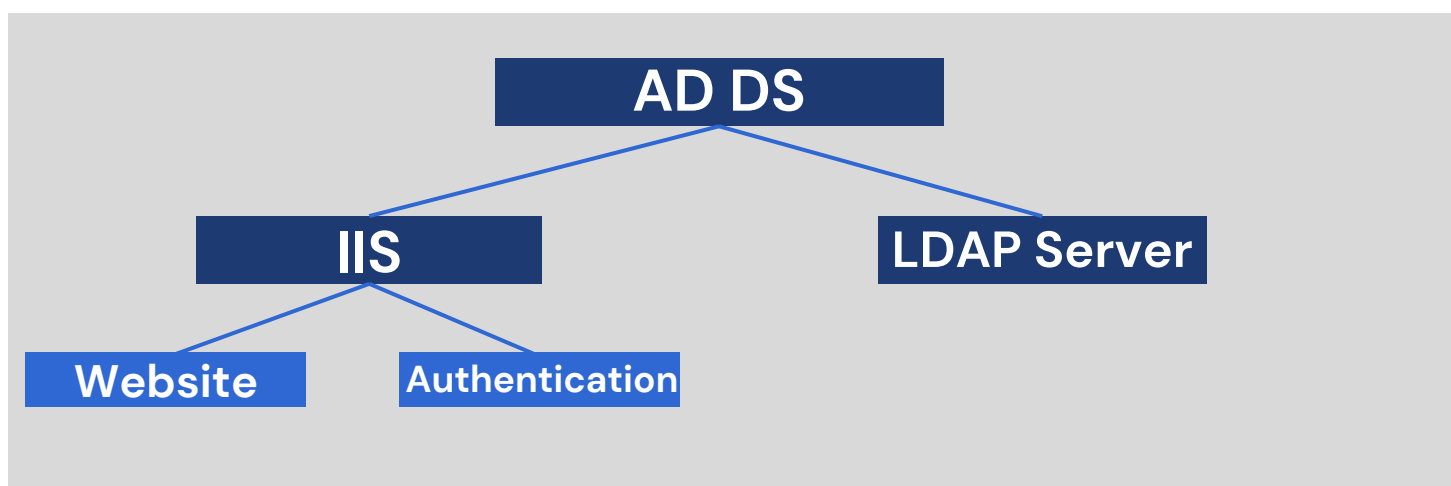
USERNAME

Aws Ghanem | Fares Qarali

PASSWORD

Configuration Structure

In order to Enable the LDAP authentication protocol, there are some few things that has to be configured, We will go through each one of them in details.



- 1) **Active Directory Domain Service:** Full Configuration statring from Configuring the server, Joining the domain and creating users.
 - 2) **Web Server IIS:** Configuring the IIS and Adding our Website and Enabling the Integrated Windows Based Authentication within it.
 - 3) **LDAP Server:** Installing the LDAP Server and Naming it propelry, and then Linking it to the AD DS.
-

Configuration Steps

1) Active Directory Domain Service:

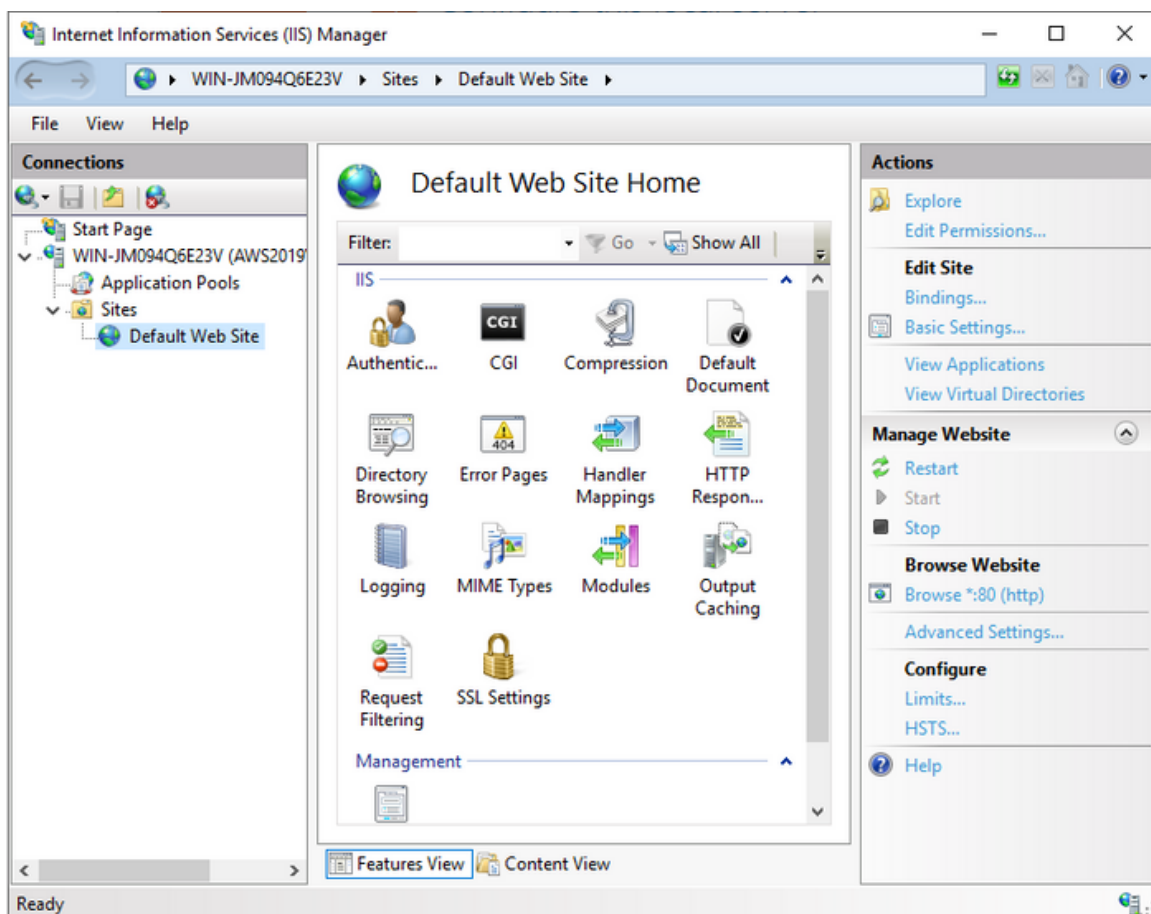
As We took on the Labs, nothing changed on that configuration

2) Web Server IIS:

First of all, we Installed the Web Server from Add Roles and Features
Then we configured the steps below:

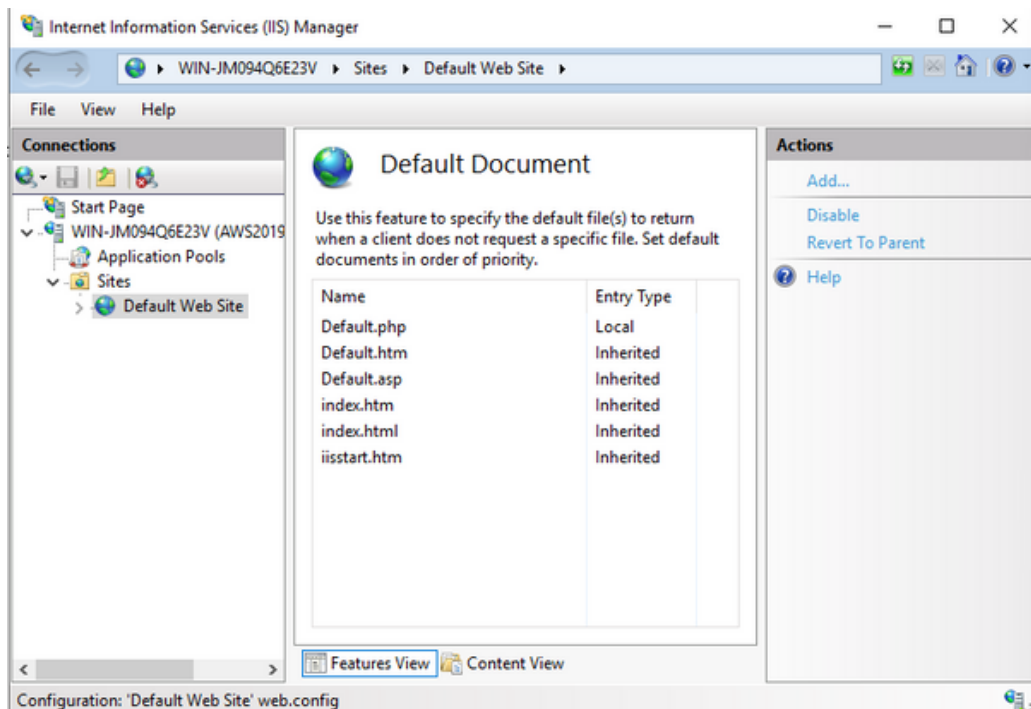
[A] Setting Up our Default Website on the IIS

From Tools → IIS Manager ,
then Sites → Default Web Site
then Default Document

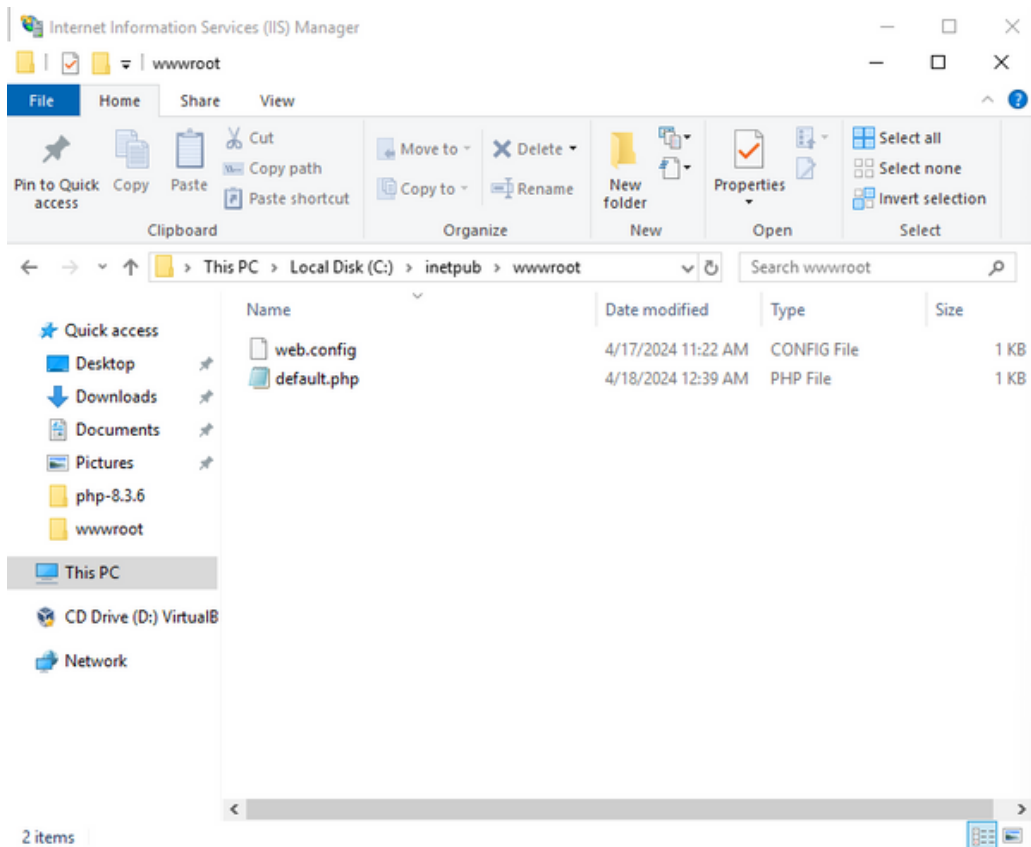


First of All we want to change the Extension of our Default Website to .php

Click on Add and name it as Default.php and give it the highest priority



the path of the files is C://intepub//wwwroot
we added our Code here as shown below

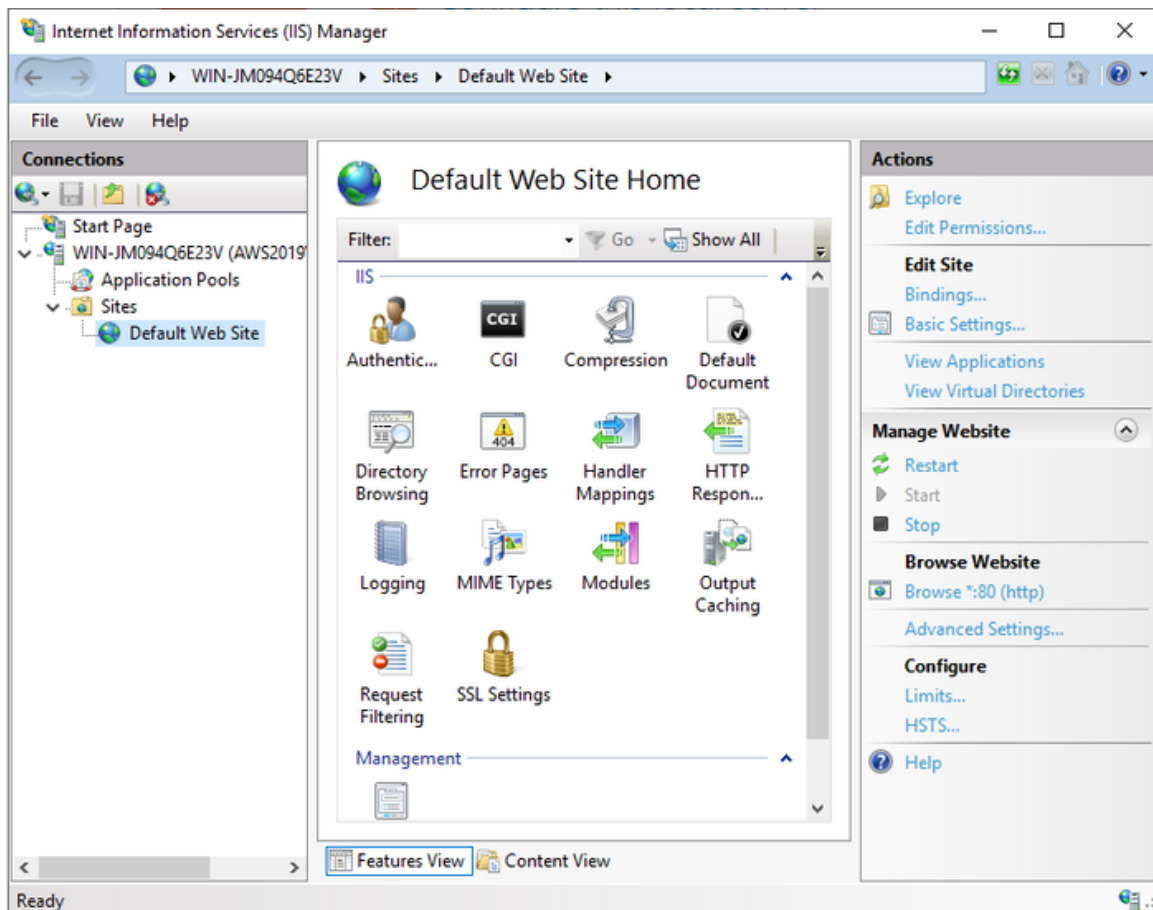


our HTML/PHP code for the website

```
default.php - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Welcome</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
  <style>
    body{ font: 14px sans-serif; text-align: center; }
  </style>
</head>
<body>
  <h1 class="my-5">Hi, Welcome to our site.</h1>
  <p>
    <a href="forget.php" class="btn btn-warning">Reset Your Password</a>
    <a href="logout.php" class="btn btn-danger ml-3">Sign Out of Your Account</a>
    <div><a href="email.php" class="btn btn-dark ml-3">Update Email</a></div>
  </p>
</body>
</html>
```

[B] Enabling the Windows Based Authentication on the IIS

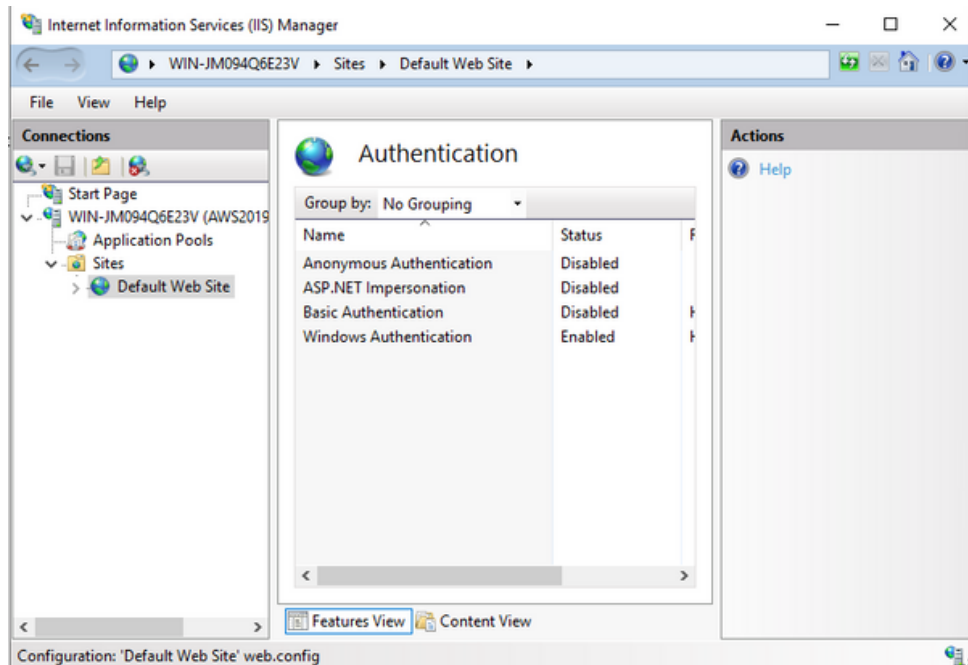
Double click on Authentication



We can see various methods for Authentication

We choose the Windows Authentication that is linked with the LDAP Server installed on AD DS

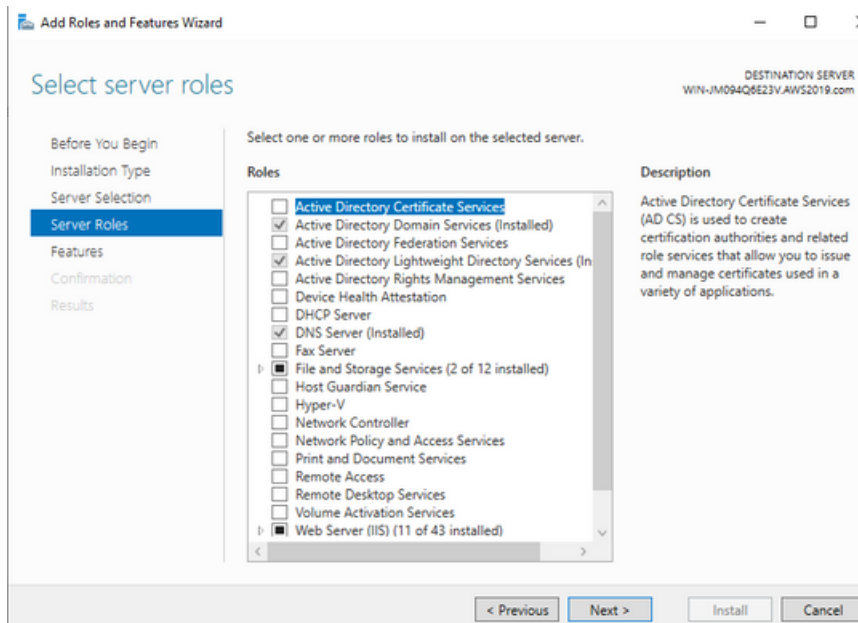
Click → Enable in the top right side



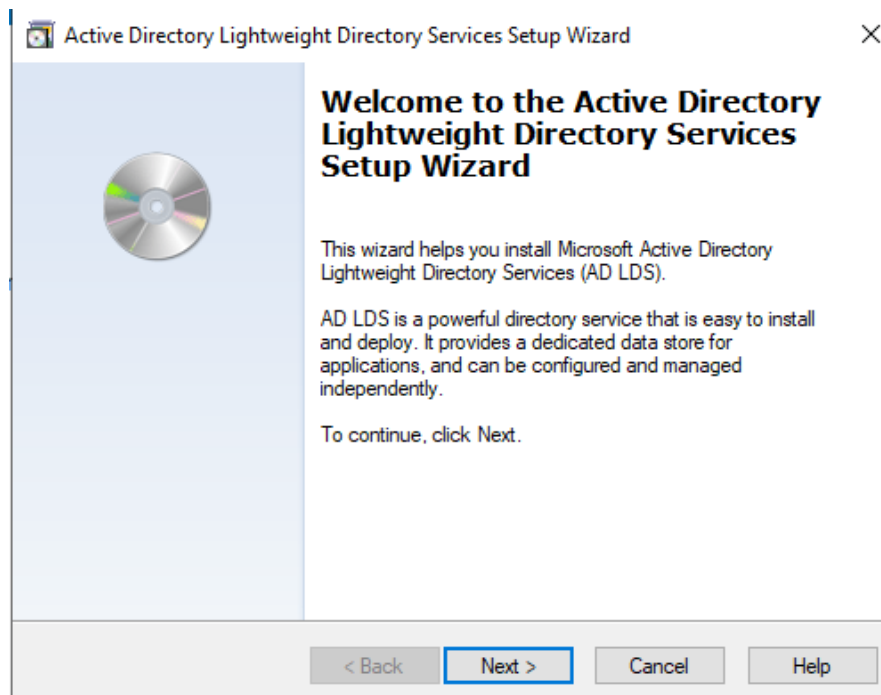
Now The IIS is Ready, but we still have to Install the AD LDS and link the LDAP Server into our Active Directory.

3) LDAP Server: In order to install the AD LDS

Manage → Add Roles and Features → Active Directory Lightweight Directory Services → then Next till finishing installation

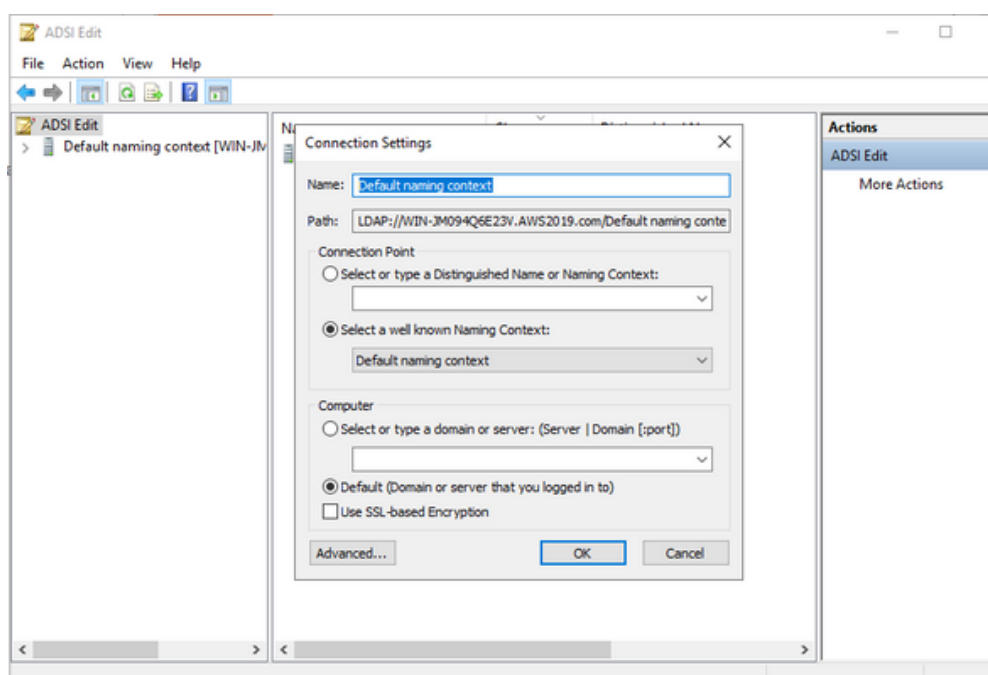


When the installation is done, The ADSI Wizard will launch automatically
Click Next and Specify Names and Ports then install.



Now We want to Link the LDAP server that we've created into the AD DS
from Tools → ADSI Edit → Right click on ADSI Edit file → Connect

We have to specify the name of our LDAP server that we created and the
DN (Distinguished Name) and the Computer to connect the LDAP to



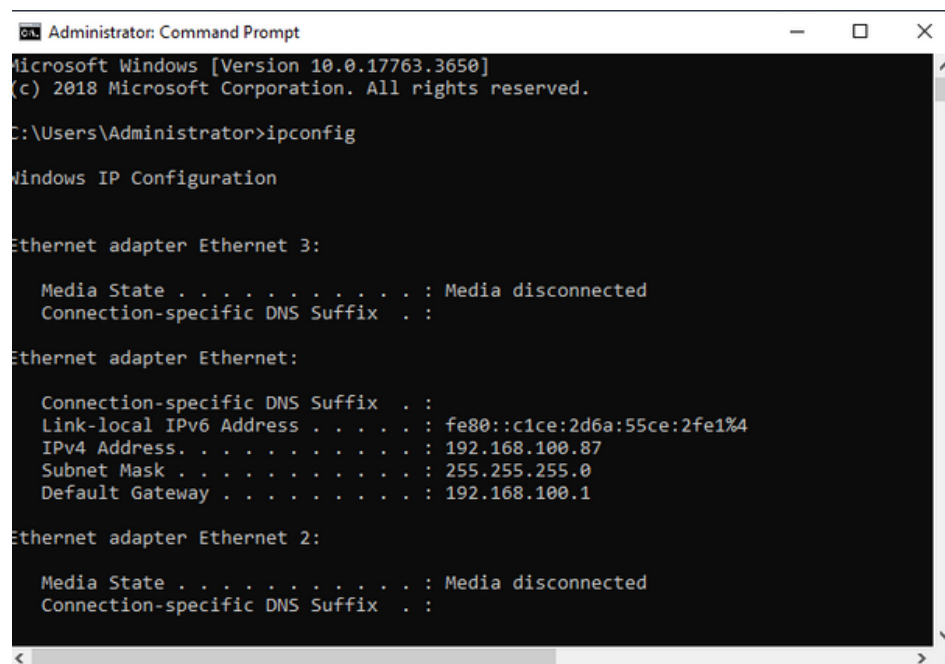
Experiments

Now Everything has been configured properly, we have our own website that are hosted on the Web IIS, and the IIS is connected to the AD DS and the LDAP Server that we rely on while the Authentication process in connected to the AD DS.

So Now we'll do the Experiment, the Following Scenario Should happen:

- 1) When Accessing the Website's IP address it should be up and prompt the user to enter the username and password.
- 2) Only the one's who has Credentials on the AD can access the website
- 3) The website can be accessed by anyone on the Same Domain Which is AWS2019, or on the Same Local Network (PC , Phone , etc ...)

The Windows Server IP is 192.168.100.87



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::c1ce:2d6a:55ce:2fe1%4
    IPv4 Address. . . . . : 192.168.100.87
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.100.1

Ethernet adapter Ethernet 2:

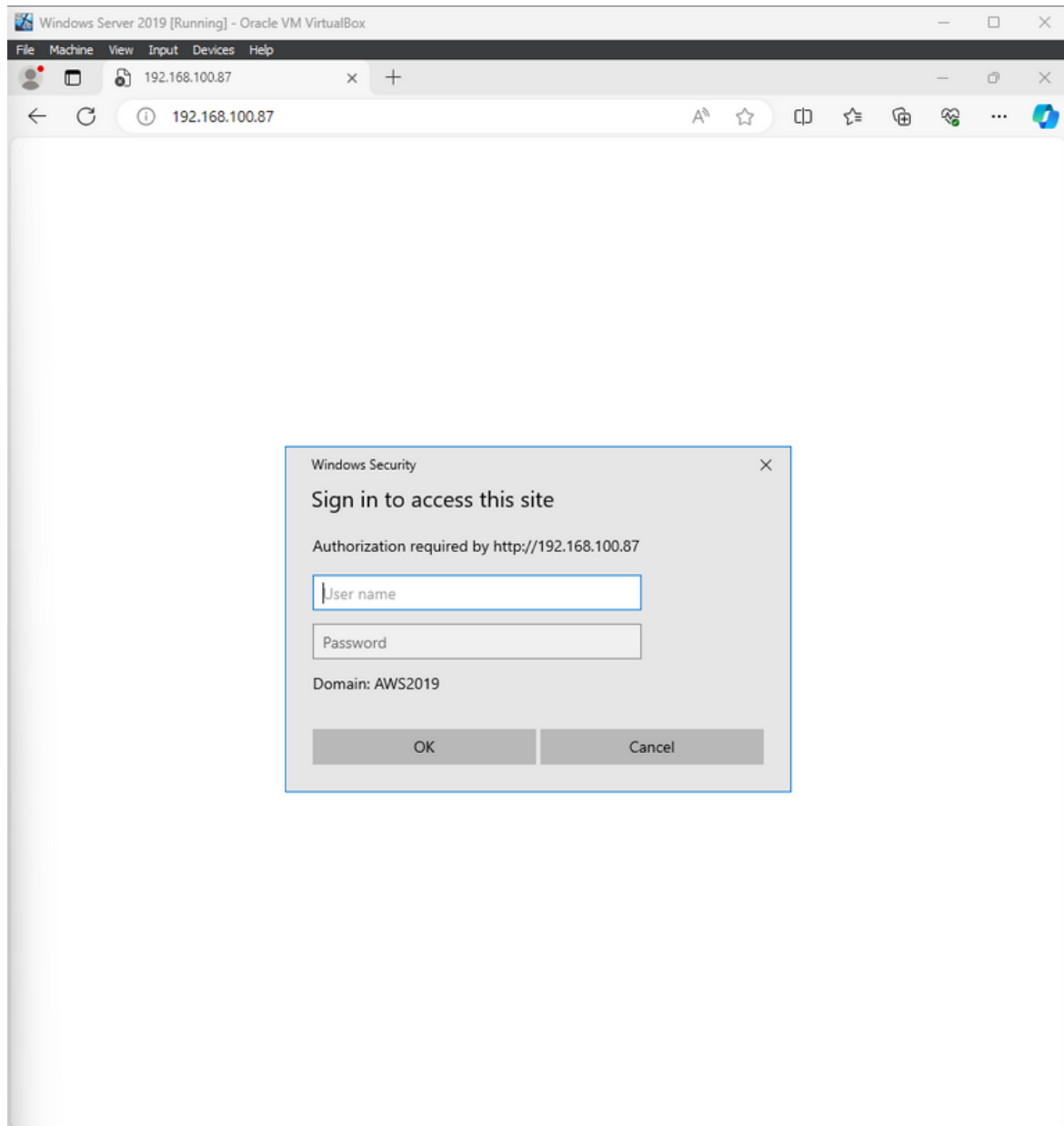
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```


Note that i have 2 Usernames on the Active Directory
Administrator and **AwsO216679**

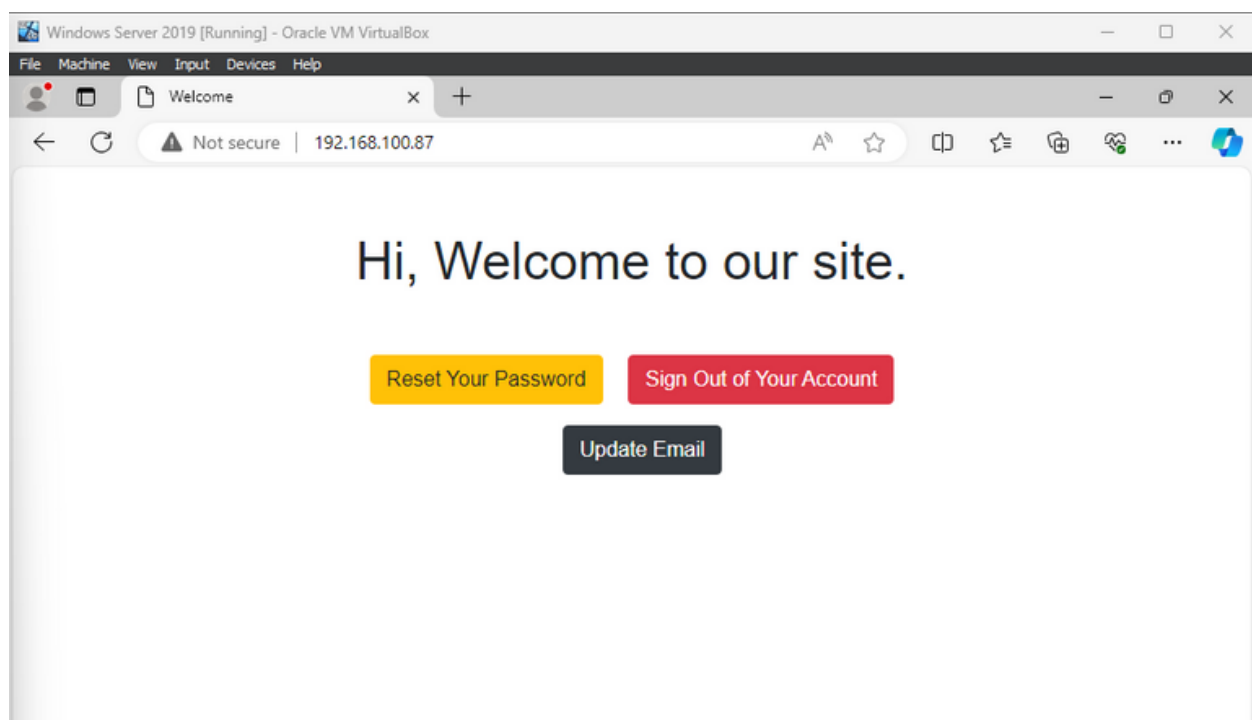
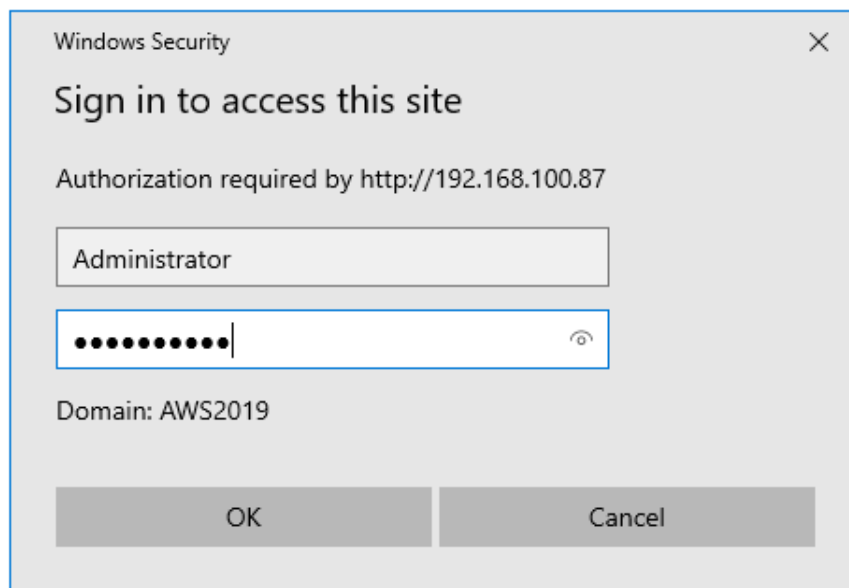
Now lets access the IP address from the Windows Server's Browser

1

We can see that it requires a username and password



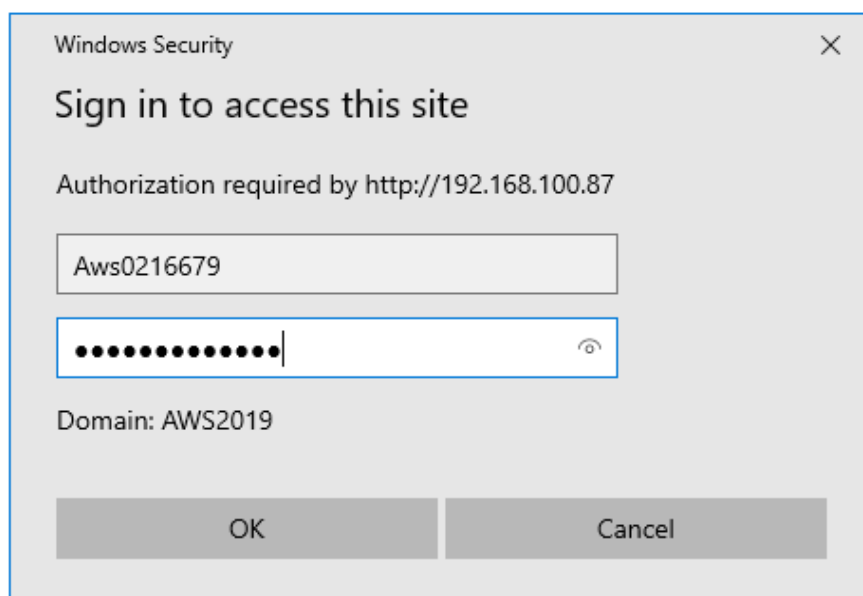
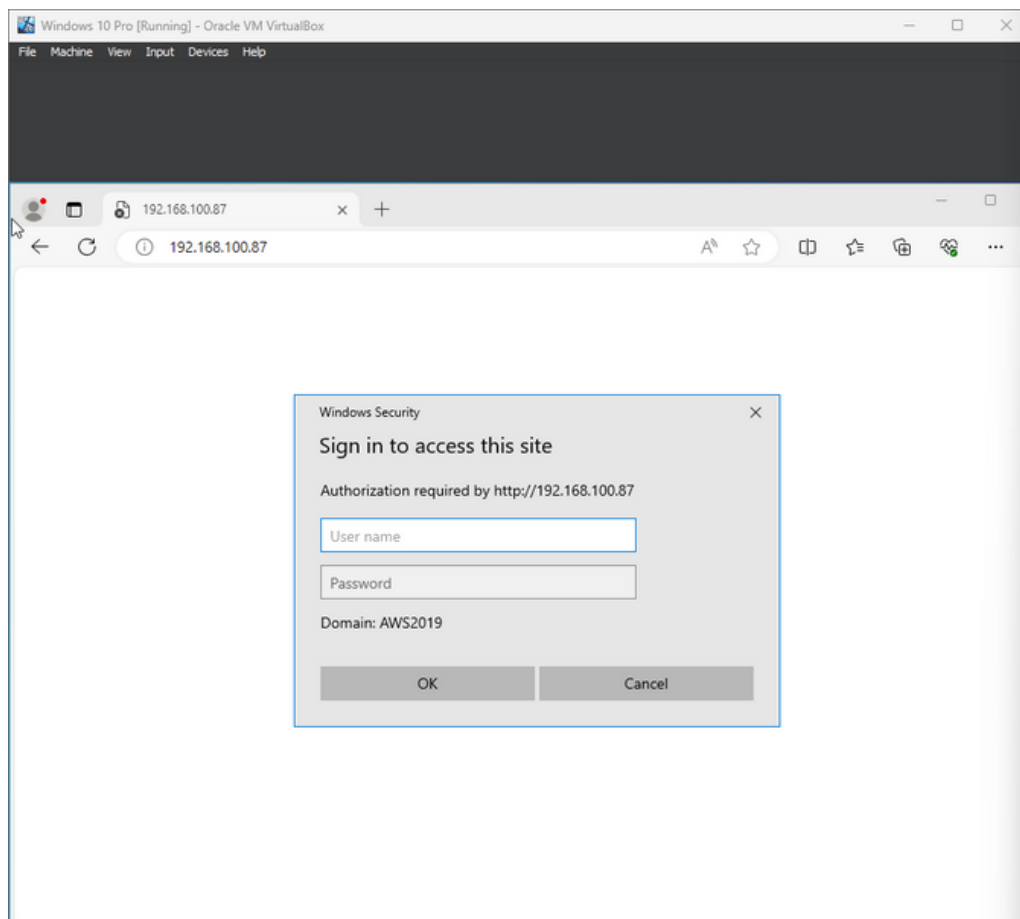
When we enter one of the valid Credentials, the user get access to the website



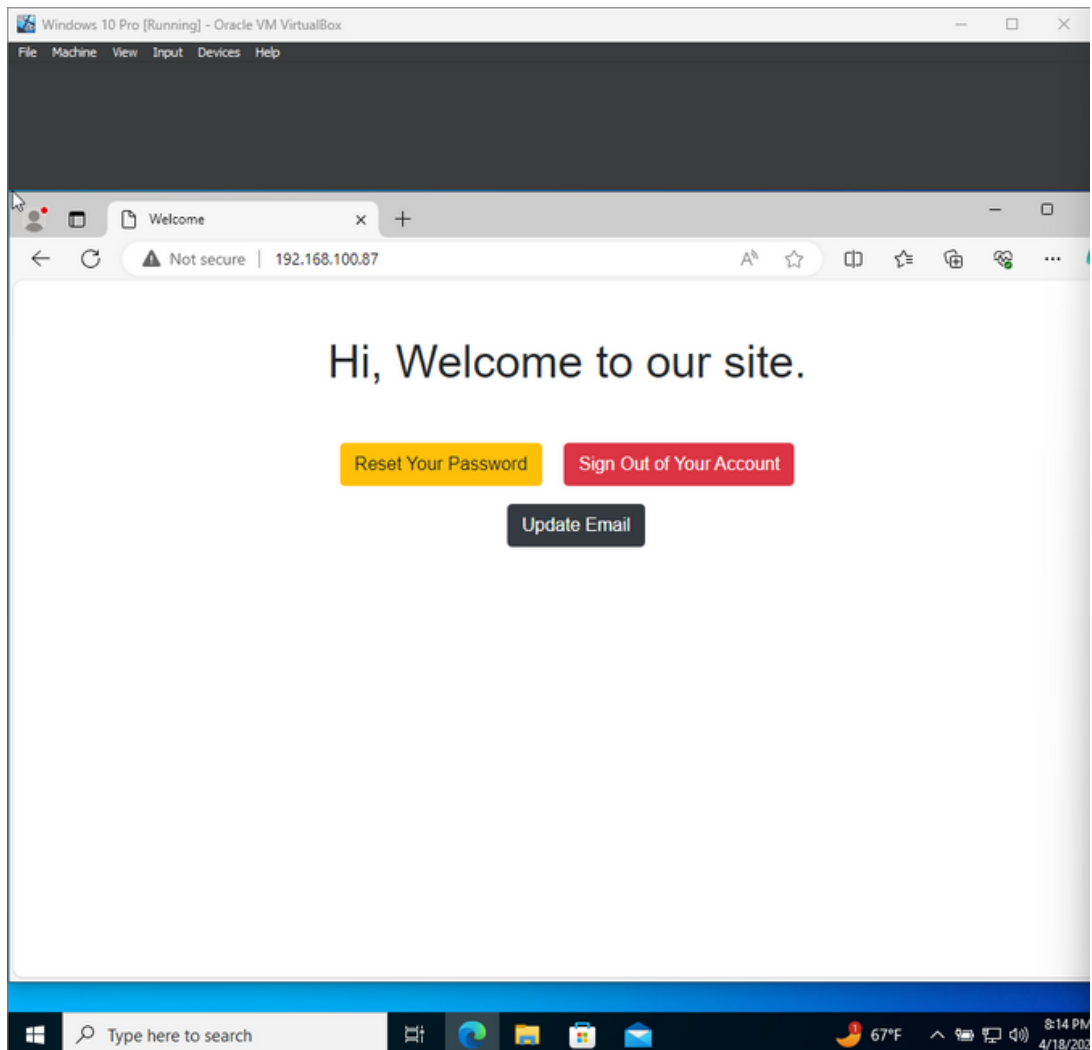
Now Lets try from another Windows 10 machine that is either joined the domain or not joined.

We enter the same IP address of the Server, just if they are on the same network

2

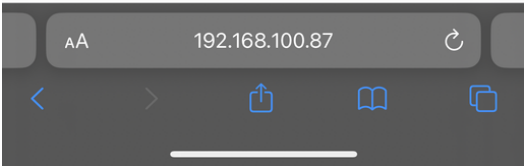
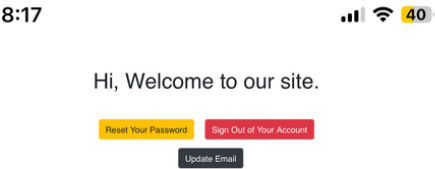
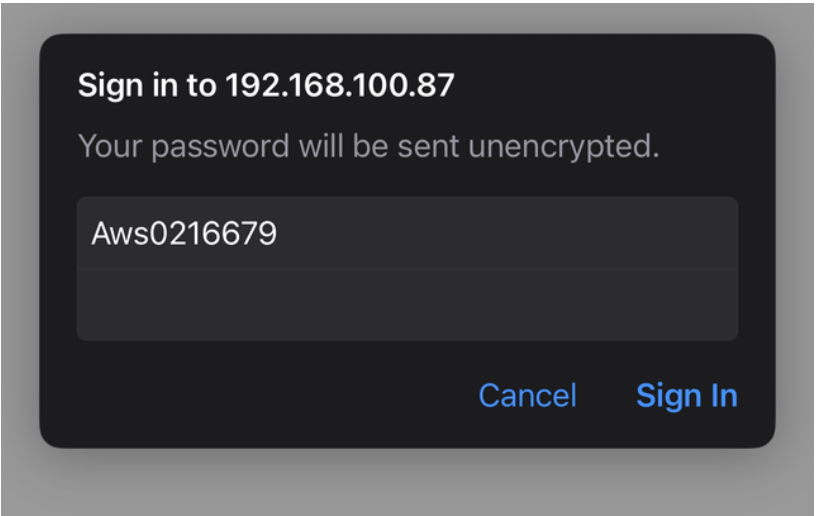
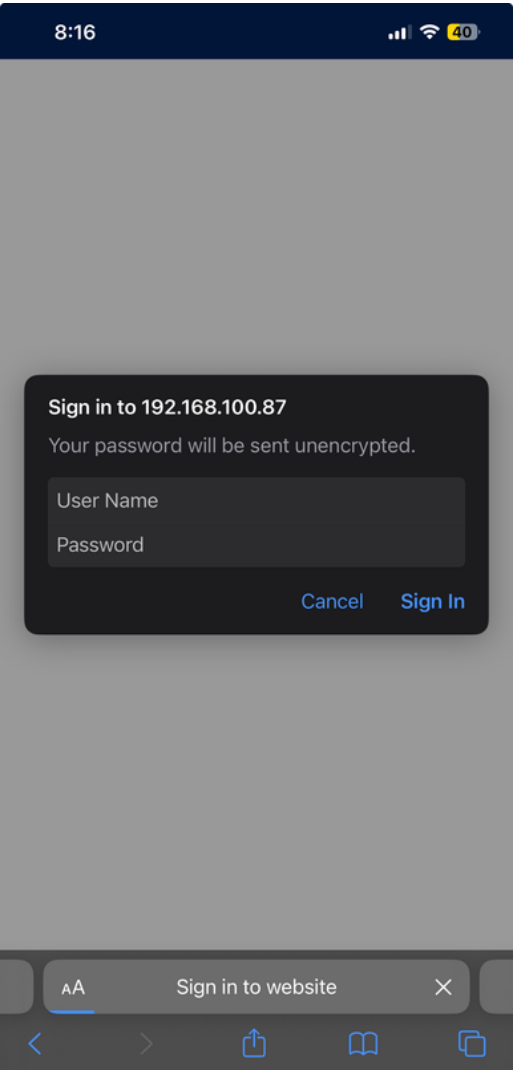


Remote user's Authentication



Also we can try to connect via the mobile phone

3



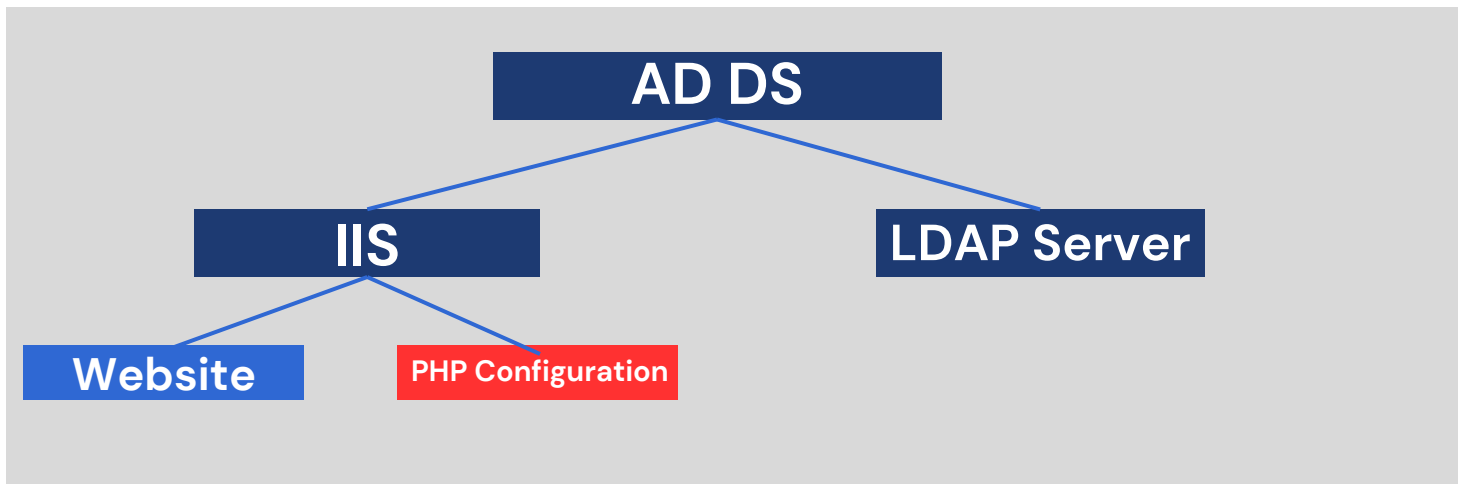
The Code

Its not that complex code that need to be reviewed
its just a simple HTML page run as php and the Authentication
we relied on as we said is based on Windows and its all about
Configuration and Linking everything together

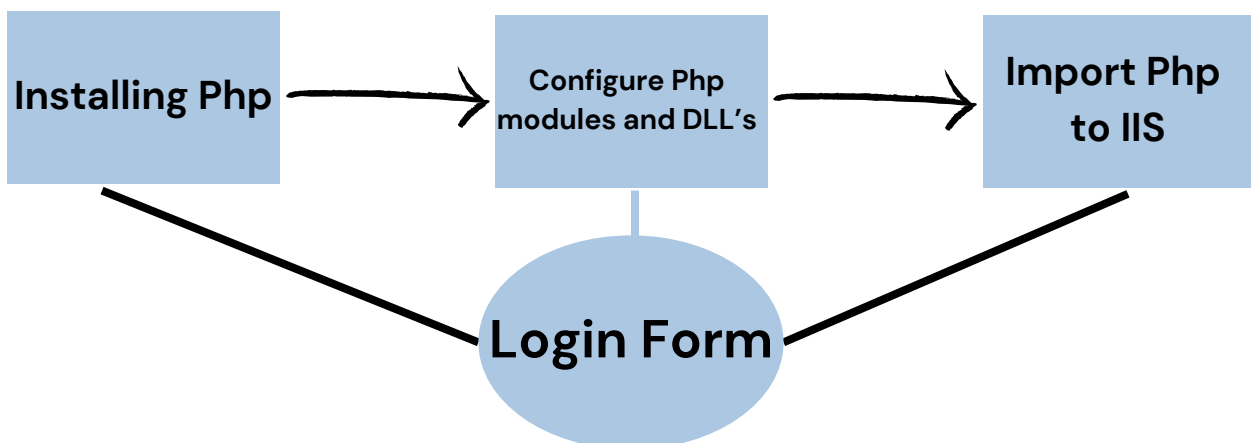
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Welcome</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    body{ font: 14px sans-serif; text-align: center; }
  </style>
</head>
<body>
  <h1 class="my-5">Hi, Welcome to our site.</h1>
  <p>
    <a href="forget.php" class="btn btn-warning">Reset Your Password</a>
    <a href="logout.php" class="btn btn-danger ml-3">Sign Out of Your Account</a>
    <div><a href="email.php" class="btn btn-dark ml-3">Update Email</a></div>

  </p>
</body>
</html>
```

Login Form Using PHP_LDAP



As the Web Server IIS on Windows Server is always installed as an empty box, there is a lot of configuration and Modules and Dynamic Link Libraries that has to be imported and configured properly in order to obtain access into the website through the LDAP authentication protocol (LDAP Server)



Installing Php

- 1) We have to install **Php** into our Windows Server 2019, from the Php Website. we choose the php-7.4 version due to some technical issues.
- 2) We should set up the **PATH** for the Php file in the Environment Variables on Windows Server.

Configuring Php Modules and DLL's

We have a file called **php.ini**, we have to Configure some stuff in it, this file is primarily acts as the settings for the php that the user can customize depending on the use case.

- to check if the ini file exists execute this command on cmd:
> **php --ini**

[A] We should include the line --> **extension=ldap** or **extension=php_ldap.dll**.

They are the base for this process to succeed as they import the most important libraries and modules for LDAP using Php.

[B] We should specify the path for the extra modules that are installed with Php in order to use them.

By editing this line --> **extension_dir= "path-to-ext-folder-in-Php-File"**.

Now Everything is configured on Php side.

```
C:\Users\Administrator>php --version
PHP 7.4.33 (cli) (built: Nov  2 2022 16:00:55) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with the ionCube PHP Loader v13.0.2, Copyright (c) 2002-2023, by ionCube Ltd.
```

```
C:\Users\Administrator>php --ini
Configuration File (php.ini) Path:
Loaded Configuration File:      C:\phpWin\php.ini
Scan for additional .ini files in: (none)
Additional .ini files parsed:    (none)
```

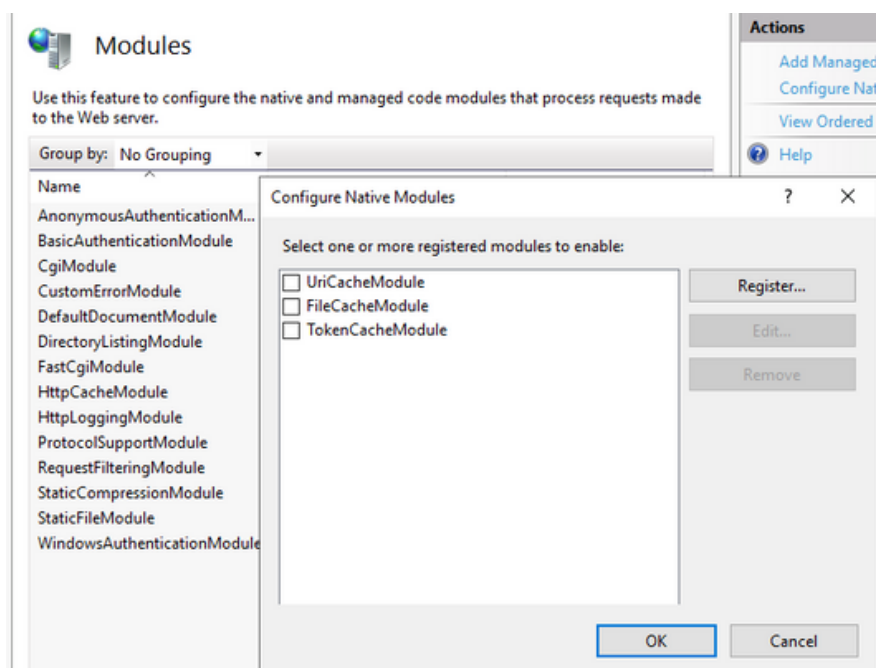
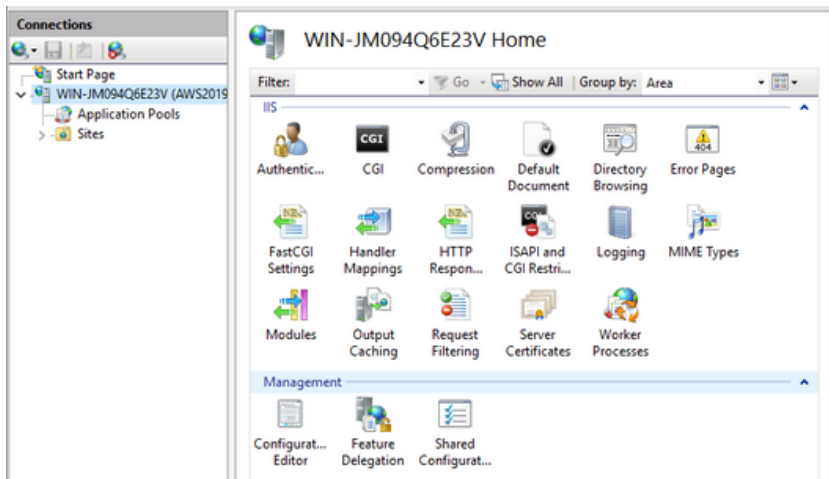

Import Php to IIS

As we said the IIS comes as a clear box, so everything that the user want to perform need to be configured fully manually.

- On our Web Server IIS We have three things has to be done:

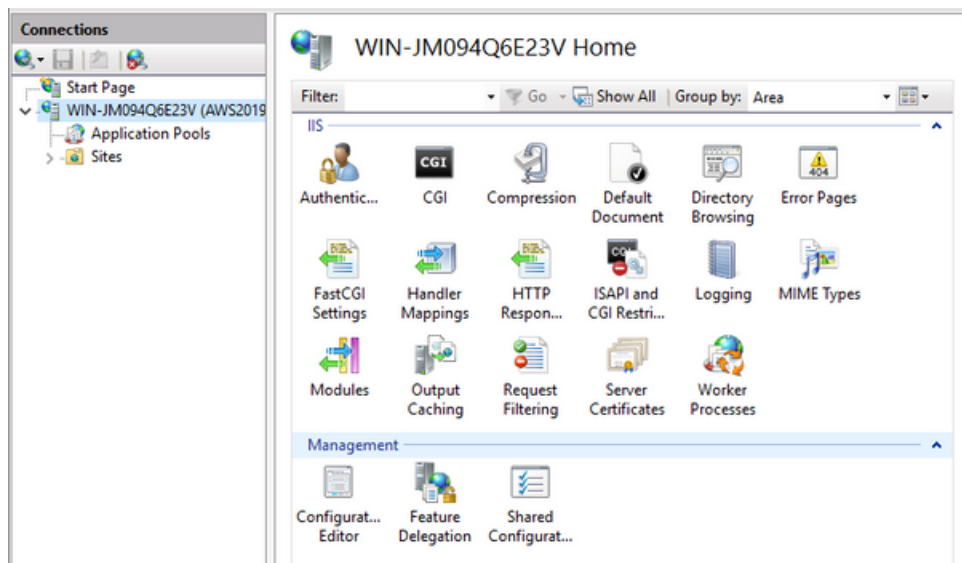
1) Uploading the FastCGI Module

Double Click on **Modules** → **Configure Native Modules** from side Actions then select **FastCGI Module** if its not selected by default



2) Updating the Handler Mappings

Double Click on **Handler Mappings** → **Add Module Mapping** from side Actions

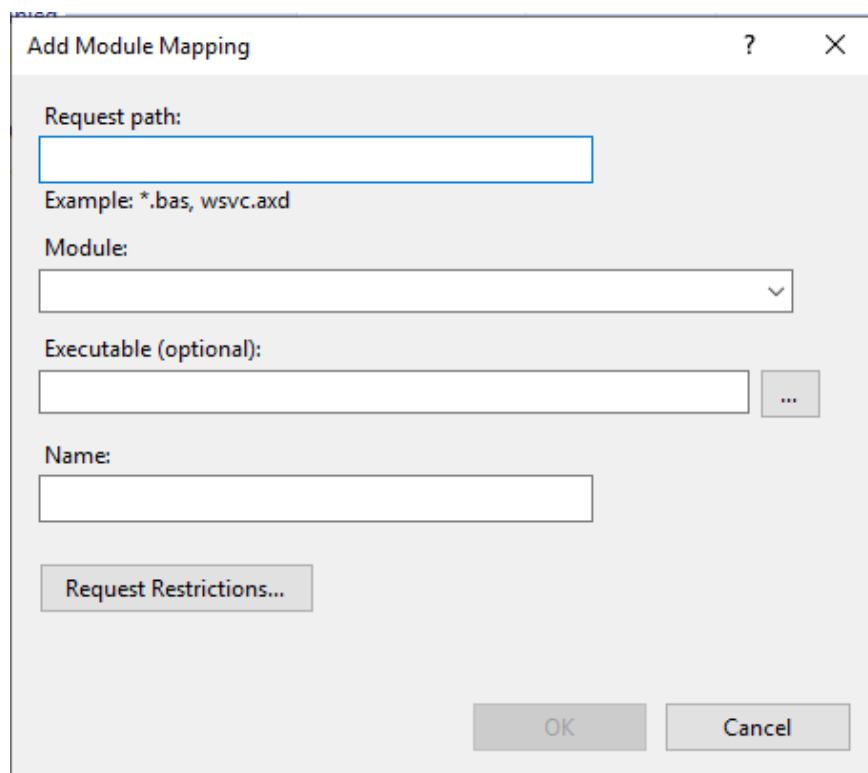


Request path: ***.php**

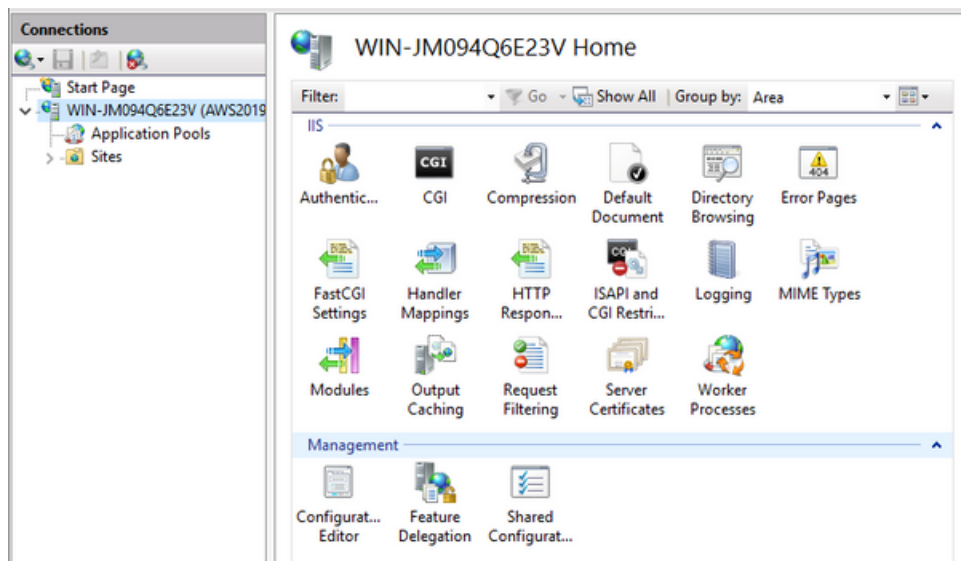
Module: **FastCGIModule**

Executable: **Path to php-cgi.exe**

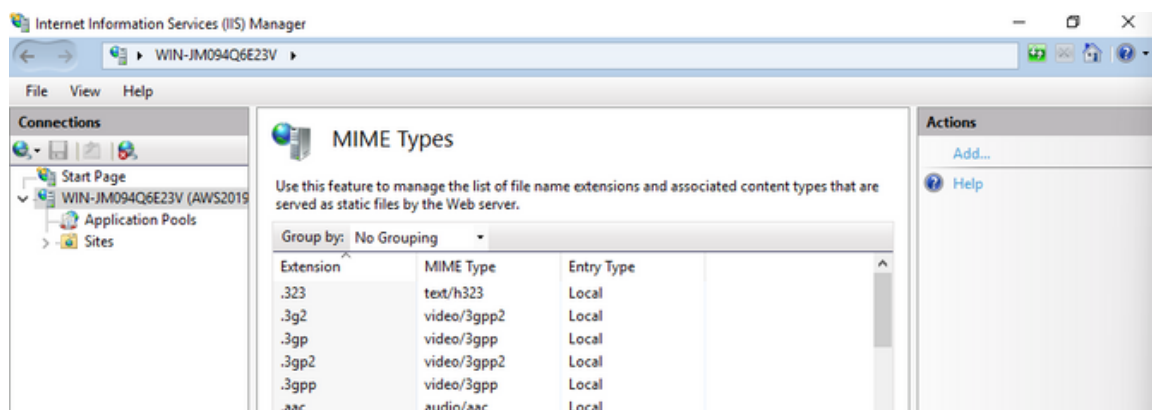
Name: -----



3) Updating the Mime Types

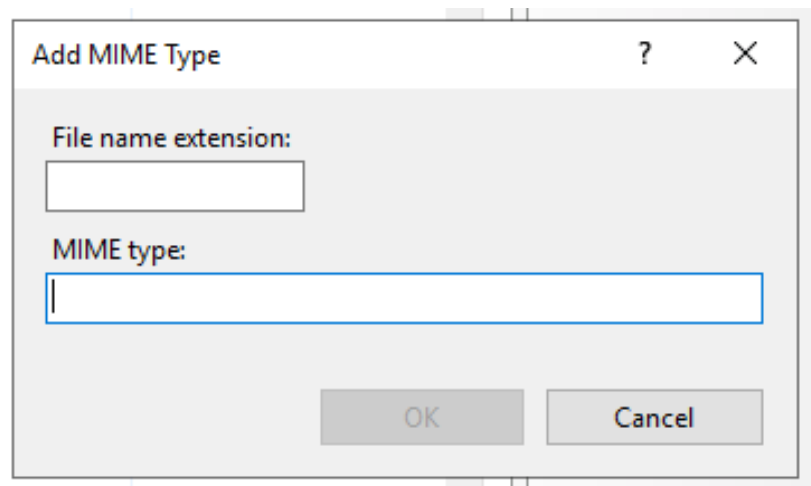


Double Click on **Mime Types** -> **Add** from side Actions



File name extension: **.php**

MIME type: **application/x-httpd-php**

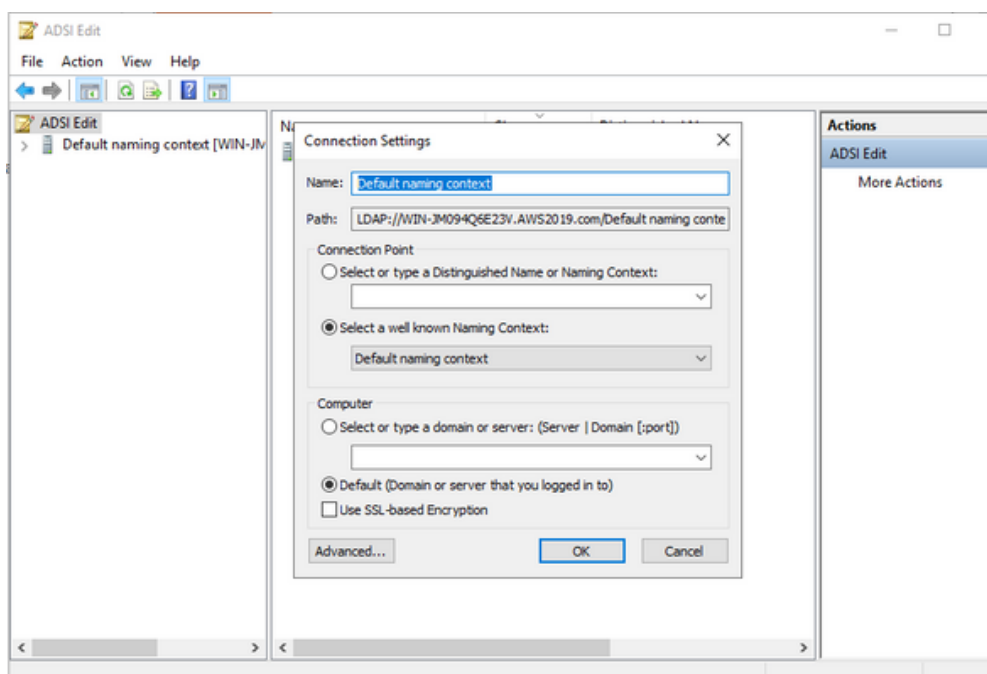


Login Form and LDAP Server Connection

No matter what the Login Form code is, what really matters is how to connect to our LDAP Server that we've created earlier.

By configuring the Php Modules which we already did in the steps before, we made the process easier to connect and validate users through the Active Directory based on the LDAP Server Scheme.

- We will rely mainly on the LDAP Server configuration to be able to connect to it by the php, so we went with the **Default Naming Context** to elementate complexity.



• Talking about the Code itself

```
<?php

// Function to redirect user to a specified URL
function redirect($url) {
    header("Location: $url");
    exit();
}

// Check if the form is submitted and username and password are provided
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['username']) && isset($_POST['password'])) {
    // LDAP server configuration
    $server = 'ldap://WIN-JM094Q6E23V.AWS2019.com';
    $port = 51278;

    // Get username and password from the form
    $username = $_POST['username'] . '@AWS2019.com';
    $password = $_POST['password'];

    // LDAP connection
    $ldap_connection = ldap_connect($server, $port);

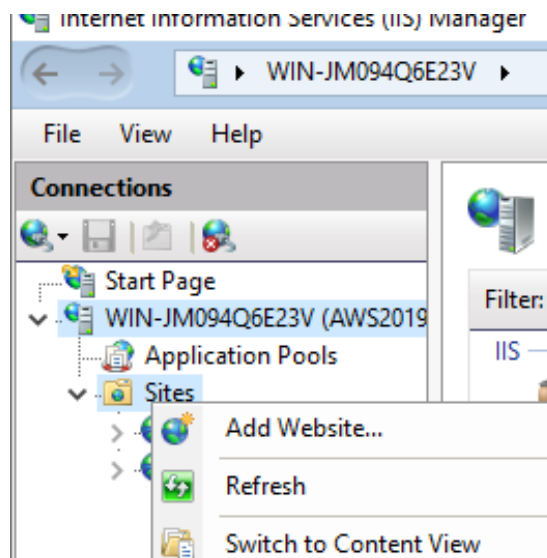
    // Check if LDAP connection is successful
    if ($ldap_connection) {
        // Attempt LDAP authentication only if username and password are provided
        if (!empty($username) && !empty($password)) {
            // Attempt to bind to the LDAP server with the provided credentials
            $ldap_bind = @ldap_bind($ldap_connection, $username, $password);

            // Check if LDAP authentication is successful
            if ($ldap_bind) {
                // Authentication successful, set session variable and redirect user to welcome.php
                session_start();
                $_SESSION['username'] = $_POST['username'];
                redirect("welcome.php");
            } else {
                // Authentication failed, display an error message
                $error_message = "Invalid username or password.";
            }
        } else {
            // Username or password is empty, display an error message
            $error_message = "Username or password cannot be empty.";
        }
    } else {
        // LDAP connection failed, display an error message
        $error_message = "LDAP connection failed.";
    }
}

?>
```

- If we redirecting into another Website

From IIS we can Add another Website into the Server

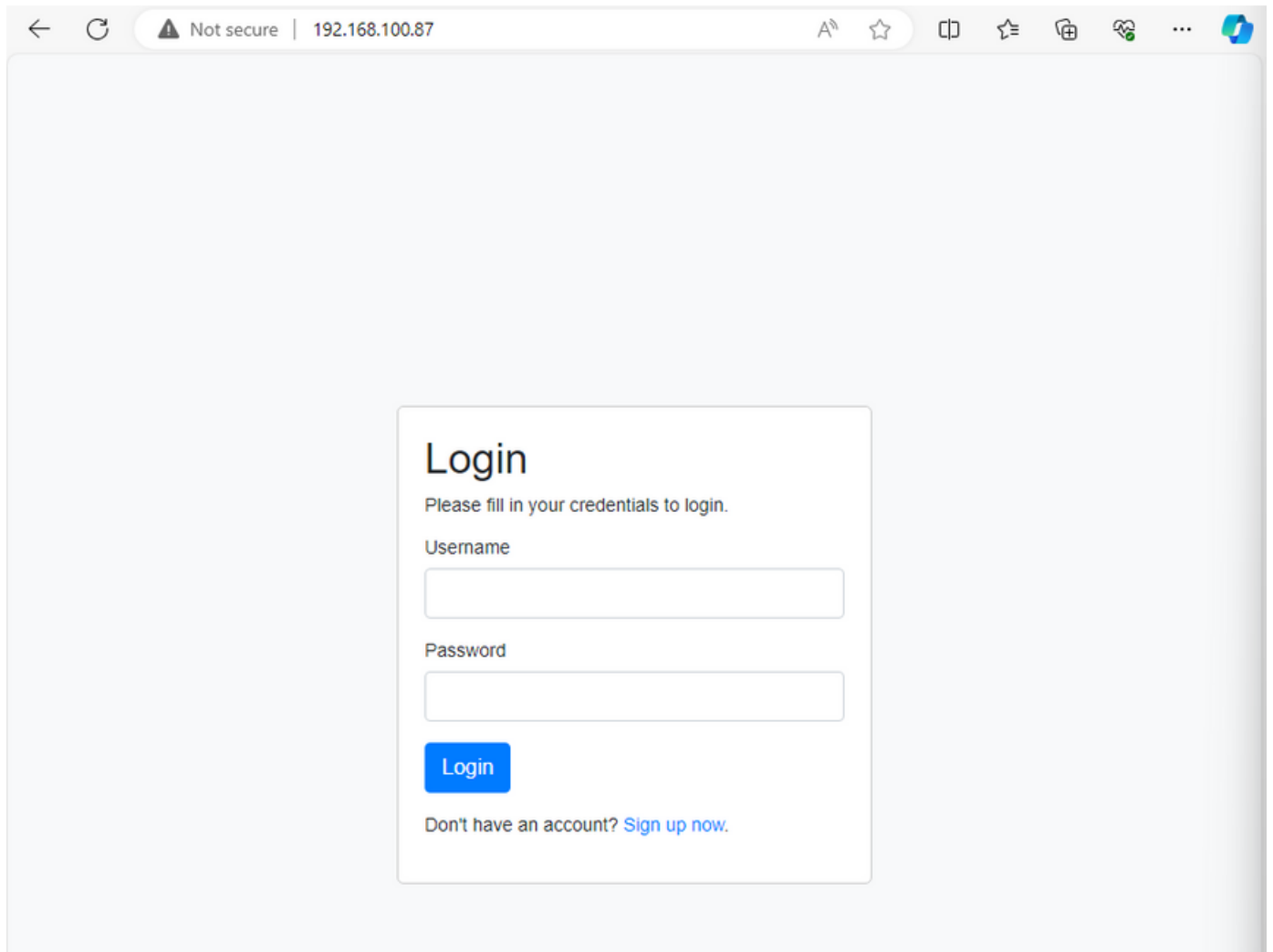
The screenshot shows the 'Add Website' dialog box. It has the following fields and options:

- Site name:** A text box for the website name.
- Application pool:** A dropdown menu showing 'DefaultAppPool' and a 'Select...' button.
- Content Directory:**
 - Physical path:** A text box with a browse button ('...').
 - Pass-through authentication:** A checkbox.
 - Connect as...** and **Test Settings...** buttons.
- Binding:**
 - Type:** A dropdown menu showing 'http'.
 - IP address:** A dropdown menu showing 'All Unassigned'.
 - Port:** A text box showing '80'.
 - Host name:** A text box.
 - Example:** www.contoso.com or marketing.contoso.com
- Start Website immediately:** A checked checkbox.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Application Pool: **Default**
Physical path: **C:\intepub\wwwroot**
Port: **Specify available port**
Host Name: **Optional**

Experiments on PHP_LDAP

- **Connecting to the IP or Domain**



A screenshot of a web browser window. The address bar shows "Not secure" and the IP address "192.168.100.87". The page content is a simple login form centered on a light gray background. The form has a title "Login", a subtitle "Please fill in your credentials to login.", and two input fields labeled "Username" and "Password". Below the fields is a blue "Login" button. At the bottom of the form, there is a link that says "Don't have an account? Sign up now.".

← ↻ ⚠ Not secure | 192.168.100.87

⌵ ☆ 📄 ⌵ 📄 📄 ⋮

Login

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? [Sign up now.](#)

- Entering valid username and password

Login

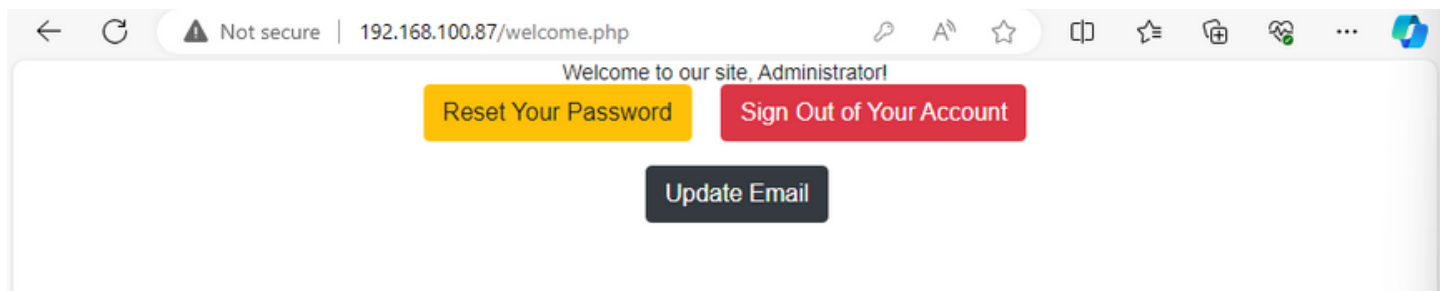
Please fill in your credentials to login.

Username

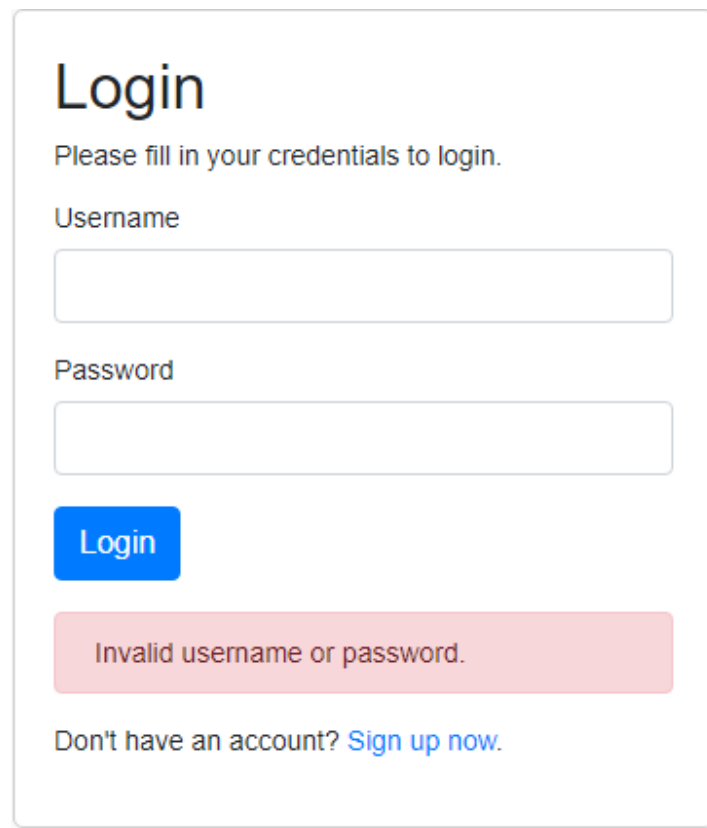
Password

Login

Don't have an account? [Sign up now.](#)



- **Entering invalid username and password**



Login

Please fill in your credentials to login.

Username

Password

Login

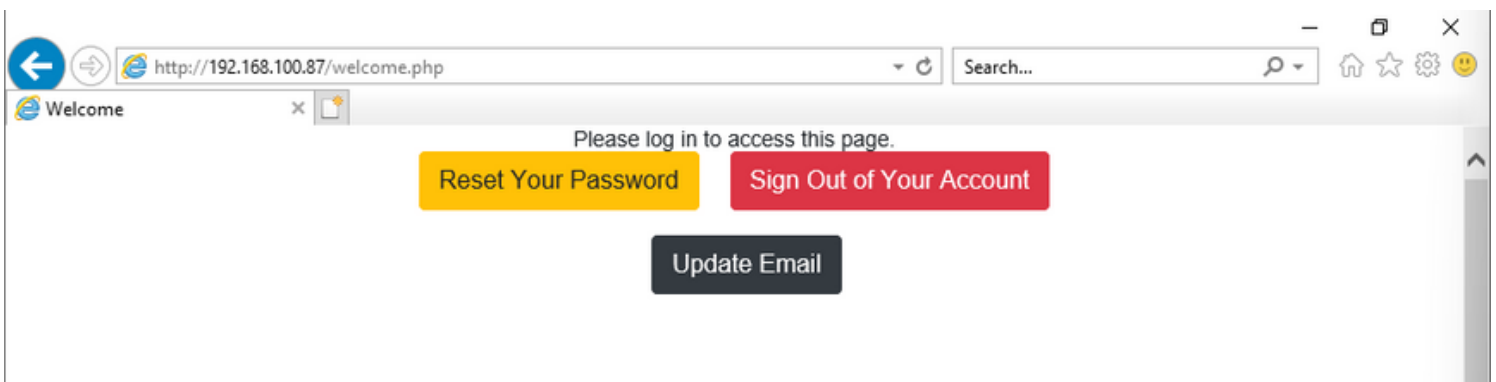
Invalid username or password.

Don't have an account? [Sign up now.](#)

- **All of the validation is done through the code we provided earlier and based on the LDAP Bind function which checks the validity of the credentials.**
 - **This experiment works on any device on the same network which perfectly fits the scope of this project and its purpose.**
 - **More experiments will be shown in the class during the presentation.**
-

Adding Layer of Security based on SESSIONS

- To wrap up everything, we would like to enhance the Security level of our project.
 - Session management is the most important thing on Web development when it comes to login or redirection between pages.
 - its just about checking if the user who tries to access specific website is Authorized before accessing it, so we want to prevent the bypassing of the Authentication Process using SESSIONS
- Before** When we try to access the path welcome.php, it bypass the authentication process and lets any user to access.

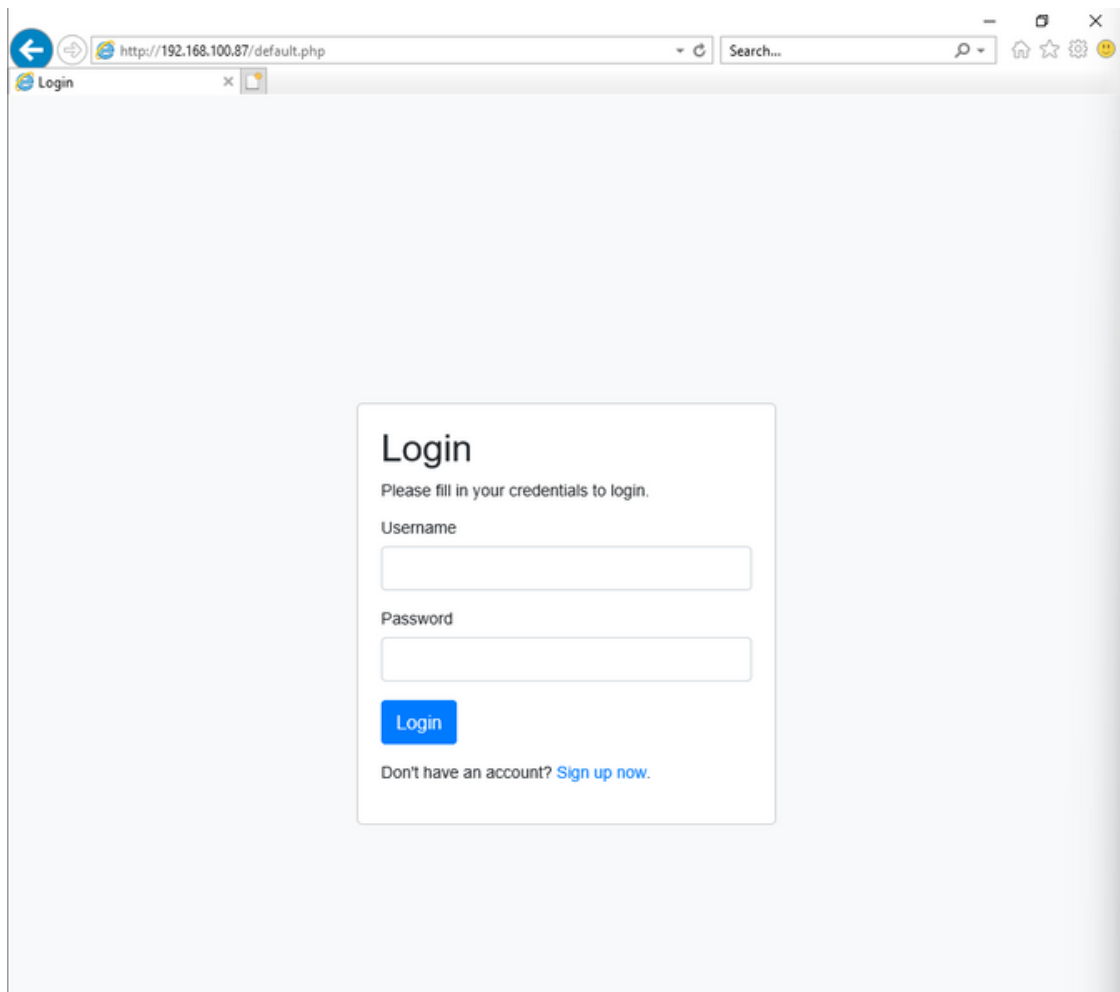


After

We enhanced the welcome.php code to only allow the logged users to access the website, by storing the username on the SESSION variable for later use, by this improvement we secured our website against the Authenticaiton Bypassing attacks (**Access Control**)

```
k?php
// Start the session
session_start();

// Check if the 'username' session variable is set
if (isset($_SESSION['username'])) {
    // Retrieve the username from the session
    $username = $_SESSION['username'];
    // Display a welcome message
    echo "Welcome to our site, $username!";
} else {
    // If 'username' session variable is not set, redirect to the login page
    header("Location: default.php");
    exit();
}
?>
```



The screenshot shows a web browser window with the address bar displaying `http://192.168.100.87/default.php`. The page contains a login form titled "Login" with the instruction "Please fill in your credentials to login." The form has two input fields: "Username" and "Password". Below the password field is a blue "Login" button. At the bottom of the form, there is a link: "Don't have an account? [Sign up now.](#)".

The End

Made By Aws Ghanem and Fares Qarali



GitHub: <https://github.com/AwsGhanem>



Aws Ghanem



Fares Qarali
