



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE3501-Information Security Analysis and Audit

Topic:- Authentication and Authorization

Assignment-5

Name: Adrija Mukhopadhyay

Register Number: 19BDS0159

Submitted Date: 21.11.2021

Course Instructor: Prof. SARAVANAGURU RA. K

Exercise-5: Authentication and Authorization

Study and Analyze the importance of authentication and authorization process with the following:

1. Create a database with username,password,role as fields (Role: faculty/student)
2. Use hashfunction to store password. (also may use salt as secret key)
3. Validate the user through forms based authentication
4. Based on user credentials try to redirect respective webpage to access the directory with files.
5. Share the authentication portion alone to your friends website similar to CORS (CROS - Cross Origin Resource Sharing)

• Creating a database and using hash function to store password:-

simple login form

```
<asp:Table ID="Table1" runat="server" Width="418px" Height="209px">
  <asp:TableRow>
    <asp:TableCell>
      User Name
    </asp:TableCell>
    <asp:TableCell>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>
      Password
    </asp:TableCell>
    <asp:TableCell>
      <asp:TextBox ID="TextBox2" runat="server"
TextMode="Password"></asp:TextBox>
    </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>
      ROLE
    </asp:TableCell>
    <asp:TableCell>
      <asp:TextBox ID="TextBox2" runat="server"
TextMode="ROLE"></asp:TextBox>
    </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>
```

```

        <asp:Button ID="Button2" runat="server" Text="login"
OnClick="login_click" />
    </asp:TableCell>
    <asp:TableCell>
        <asp:Label ID="Label1" runat="server" />
    </asp:TableCell>
</asp:TableRow>
</asp:Table>

```

create the Database table

```

Select MySQL 8.0 Command Line Client
mysql> create table labfive(username varchar(50),password varchar(100),role varchar(20));
Query OK, 0 rows affected (0.37 sec)

mysql> show tables;
+-----+
| Tables_in_lab5 |
+-----+
| labfive         |
+-----+
1 row in set (0.00 sec)

mysql> insert into labfive(username,password,role) values('pushwanth',md5('12345'),'student');
Query OK, 1 row affected (0.00 sec)

mysql> select * from labfive;
+-----+-----+-----+
| username | password | role |
+-----+-----+-----+
| pushwanth | 827ccb0eea8a706c4c34a16891f84e7b | student |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> insert into labfive(username,password,role) values('dinesh',md5('6789'),'student');
Query OK, 1 row affected (0.00 sec)

mysql> insert into labfive(username,password,role) values('madhavi',md5('13579'),'teacher');
Query OK, 1 row affected (0.00 sec)

mysql> insert into labfive(username,password,role) values('sujatha',md5('2468'),'teacher');
Query OK, 1 row affected (0.00 sec)

mysql> select * from labfive;
+-----+-----+-----+
| username | password | role |
+-----+-----+-----+
| pushwanth | 827ccb0eea8a706c4c34a16891f84e7b | student |
| dinesh | 46d045ff5190f6ea93739da6c0aa19bc | student |
| madhavi | e13dd027be0f2152ce387ac0ea83d863 | teacher |
| sujatha | e82c4b19b8151ddc25d4d93ba7b908f | teacher |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

Recent Favorites

- New
- example
- information_schema
- lab5
 - New
 - labfive
- mysql
- performance_schema
- sys

Showing rows 0 - 3 (4 total, Query took 0.0008 seconds.)

SELECT * FROM `labfive`

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

username	password	role
pushwanth	827ccb0eea8a706c4c34a16891f84e7b	student
dinesh	46d045ff5190f6ea93739da6c0aa19bc	student
madhavi	e13dd027be0f2152ce387ac0ea83d863	teacher
sujatha	e82c4b19b8151ddc25d4d93ba7b908f	teacher

☐ Show all | Number of rows: 25 | Filter rows: Search this table

//pushwanth and dinesh are students & madhavi and sujatha are teacher

Sign up button to get user name and password:

```

public void Message_click(object sender, EventArgs e)
{
    string username = TextBox1.Text.ToString();
    String password = TextBox2.Text;
    //Get the encrypt the password by using the class
    string pass = encryption(password);
    Label1.Text = pass;
    //Check whether the UserName and password are Empty
    if (username.Length > 0 && password.Length > 0)
    {
        //creating the connection string
        string connection =
        ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();
        SqlConnection con = new SqlConnection(connection);
        String passwords = encryption(password);
        con.Open();
        // Check whether the Username Found in the Existing DB
        String search = "SELECT * FROM UserAccount WHERE (UserName = '" +
username + "')";
        SqlCommand cmds = new SqlCommand(search, con);
        SqlDataReader sqldrs = cmds.ExecuteReader();
        if (sqldrs.Read())
        {
            String passed = (string)sqldrs["Password"];
            Label1.Text = "Username Already Taken";
        }
        else
        {
            try
            {
                // if the Username not found create the new user account
                string sql = "INSERT INTO UserAccount (UserName, Password)
VALUES ('" + username + "', '" + passwords + "')";
                SqlCommand cmd = new SqlCommand(sql, con);
                cmd.ExecuteNonQuery();
                String Message = "saved Successfully";
                Label1.Text = Message.ToString();
                TextBox1.Text = "";
                TextBox2.Text = "";
                Response.Redirect("Default2.aspx");
            }
            catch (Exception ex)
            {
                Label1.Text = ex.ToString();
            }
        }
    }
}

```

```

        con.Close();
    }
}

else
{
    String Message = "Username or Password is empty";
    Label1.Text = Message.ToString();
}
}

```

encrypt the password

```

public string encryption(String password)
{
    MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();
    byte[] encrypt;
    UTF8Encoding encode = new UTF8Encoding();
    //encrypt the given password string into Encrypted data
    encrypt = md5.ComputeHash(encode.GetBytes(password));
    StringBuilder encryptdata = new StringBuilder();
    //Create a new string by using the encrypted data
    for (int i = 0; i < encrypt.Length; i++)
    {
        encryptdata.Append(encrypt[i].ToString());
    }
    return encryptdata.ToString();
}

```

Login button

```

public void login_click(object sender, EventArgs e)
{
    String username = TextBox1.Text.ToString();
    String password = TextBox2.Text;
    string con =
    ConfigurationManager.ConnectionStrings["DefaultConnection"].ToString();
    SqlConnection connection = new SqlConnection(con);
    connection.Open();
    //encrypt the given password
    string passwords = encryption(password);
    String query = "SELECT UserName, Password FROM UserAccount WHERE (UserName
= ' + username + "') AND (Password = '"+passwords+"')";

    SqlCommand cmd = new SqlCommand(query, connection);
    SqlDataReader sqlldr = cmd.ExecuteReader();
}

```

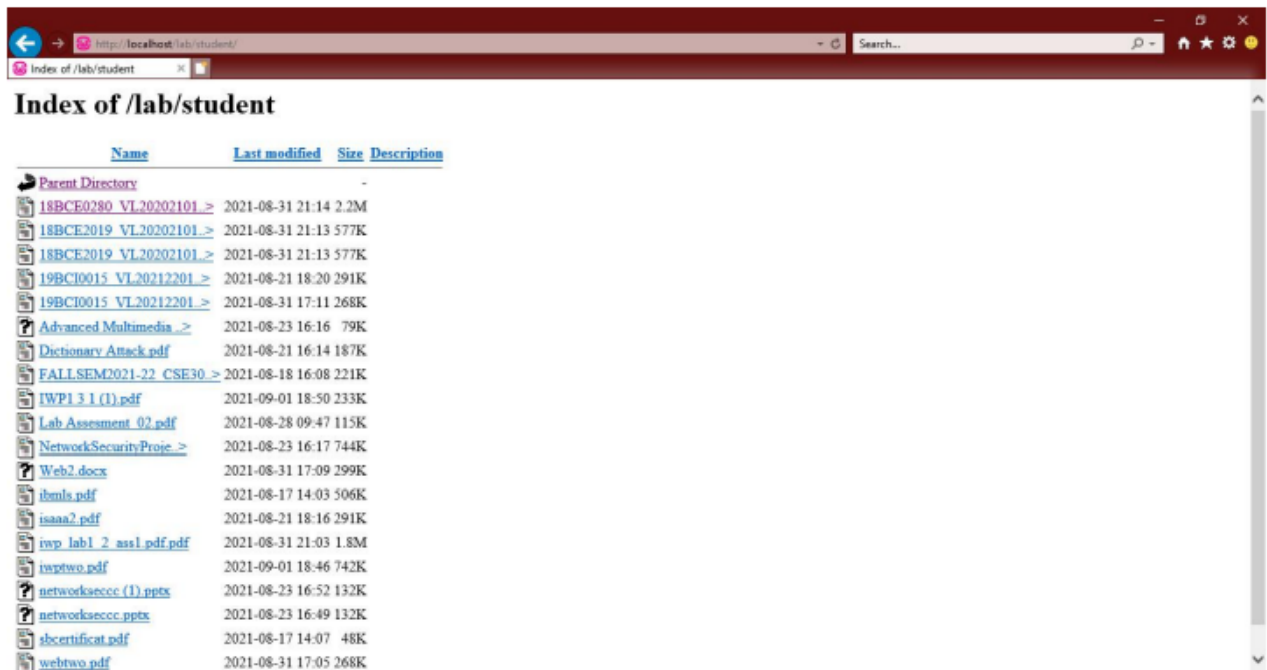
```

        if (sqlldr.Read())
        {
            Response.Redirect("Default3.aspx");
        }
        else
        {
            Label11.Text = "INVALID";
        }

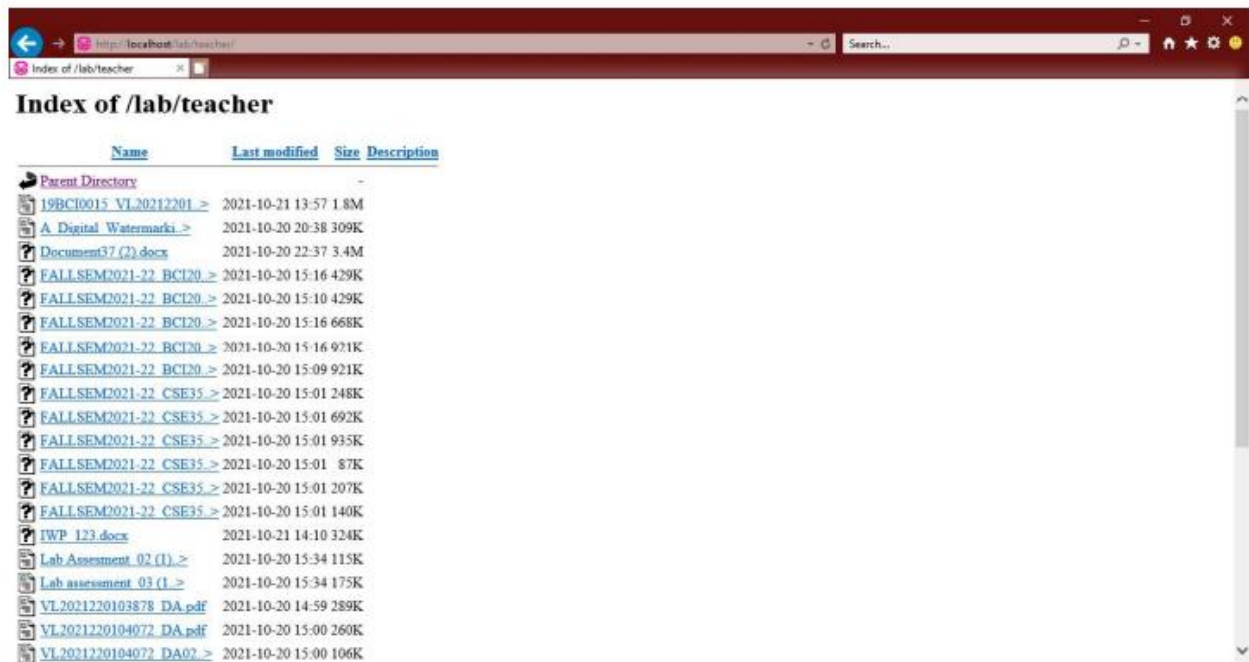
        connection.Close();
    }
}

```

- Created a directory(student) which will only available to students



- Created a directory(teacher) which will only



- Created a html page which will authenticate and authorize

For students

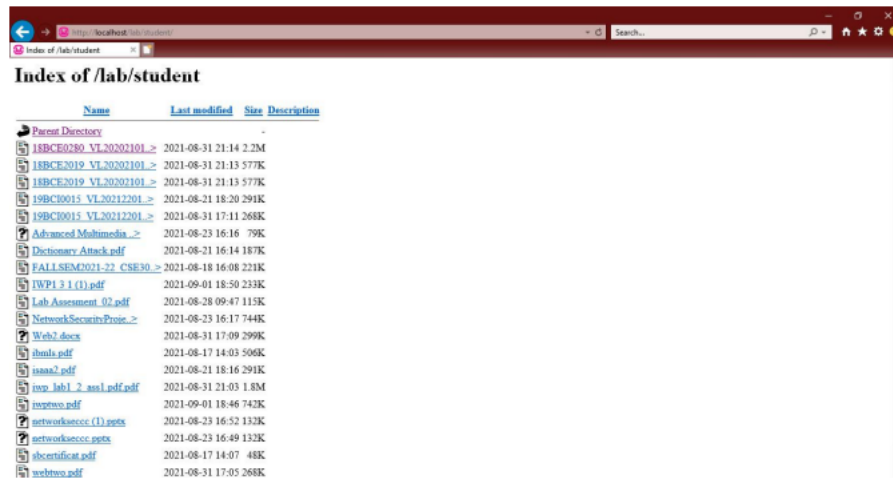
The screenshot shows a web browser window with the address bar displaying 'http://localhost/lab/issalab5.html'. The page contains a login form with the following fields and a submit button:

UserName:

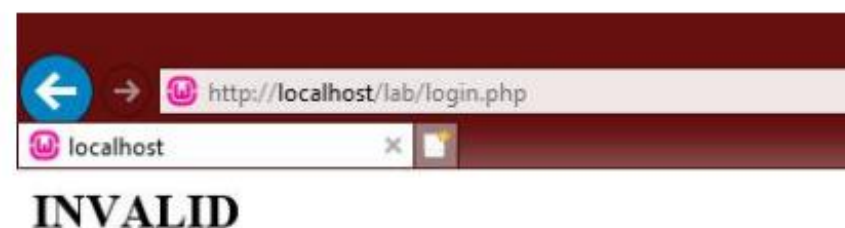
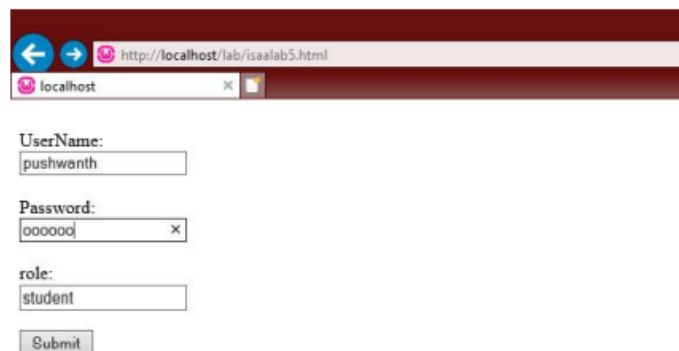
Password:

role:

After clicking on submit button it'll redirect to student directory

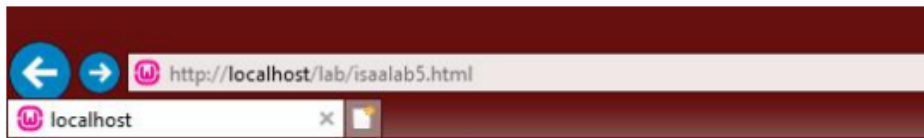


If we enter wrong password:-



If a student tries to access teacher-only accessible files:

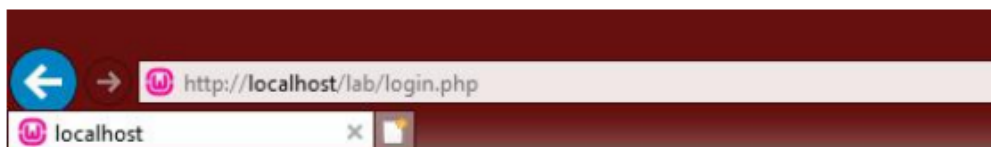
//entered role as teacher for a student//



UserName:

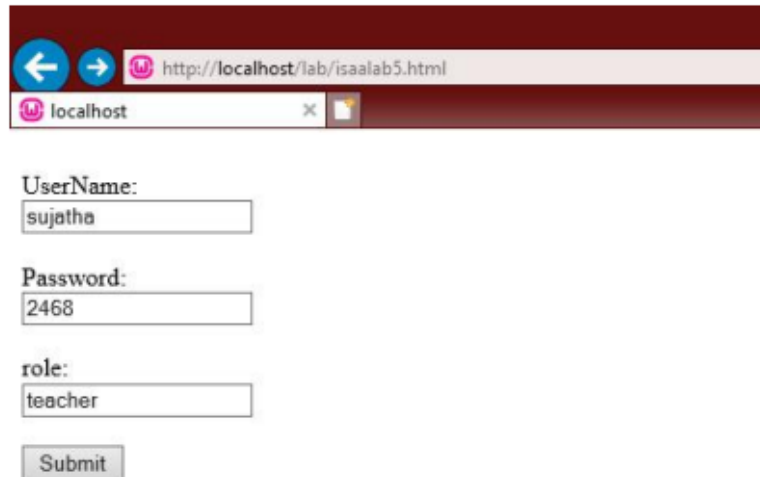
Password:

role:



INVALID

For teachers:-



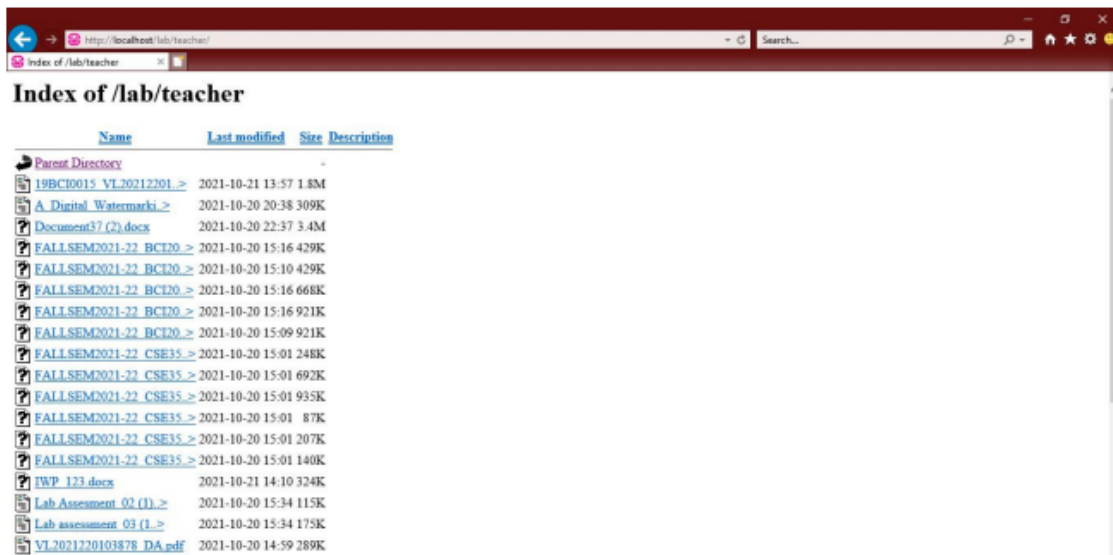
UserName:
sujatha

Password:
2468

role:
teacher

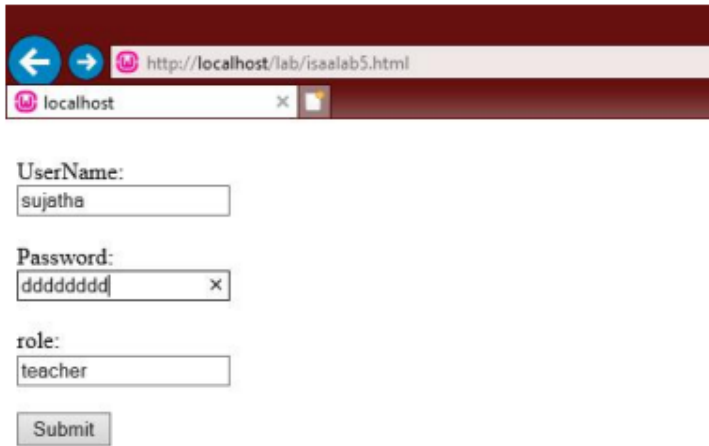
Submit

After clicking on submit button it'll redirect to teacher directory



Index of /lab/teacher

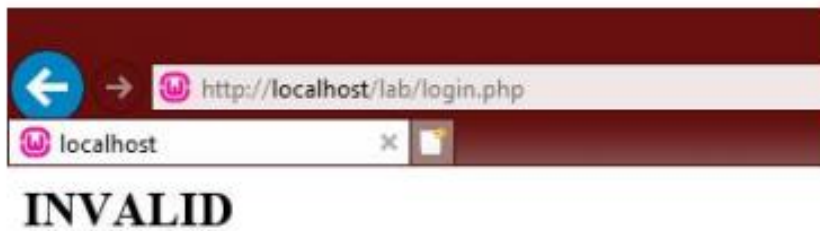
Name	Last modified	Size	Description
Parent Directory		-	
19BCI0015_VL20212201_>	2021-10-21 13:57	1.8M	
A_Digital_Watermarki_>	2021-10-20 20:38	309K	
Document37 (2).docx	2021-10-20 22:37	3.4M	
FALLSEM2021-22_BCT20_>	2021-10-20 15:16	429K	
FALLSEM2021-22_BCT20_>	2021-10-20 15:10	429K	
FALLSEM2021-22_BCT20_>	2021-10-20 15:16	668K	
FALLSEM2021-22_BCT20_>	2021-10-20 15:16	921K	
FALLSEM2021-22_BCT20_>	2021-10-20 15:09	921K	
FALLSEM2021-22_CSE35_>	2021-10-20 15:01	248K	
FALLSEM2021-22_CSE35_>	2021-10-20 15:01	692K	
FALLSEM2021-22_CSE35_>	2021-10-20 15:01	935K	
FALLSEM2021-22_CSE35_>	2021-10-20 15:01	87K	
FALLSEM2021-22_CSE35_>	2021-10-20 15:01	207K	
FALLSEM2021-22_CSE35_>	2021-10-20 15:01	140K	
IWP_123.docx	2021-10-21 14:10	324K	
Lab Assessment_02 (1)_>	2021-10-20 15:34	115K	
Lab assessment_03 (1)_>	2021-10-20 15:34	175K	
VL2021220103878_DA.pdf	2021-10-20 14:59	289K	

If we enter wrong password:

A screenshot of a web browser window. The address bar shows the URL `http://localhost/lab/isaaleb5.html`. Below the address bar, there is a tab labeled `localhost`. The main content area displays a login form with the following fields and values:

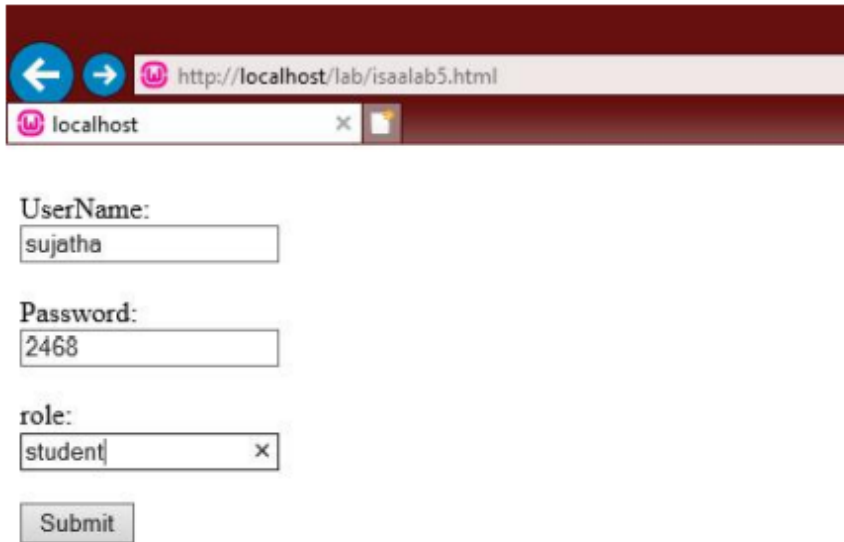
- UserName:** `sujatha`
- Password:** `ddddddd` (with a small 'x' icon to the right of the input field)
- role:** `teacher`

At the bottom of the form is a `Submit` button.



If a teacher tries to access student-only accessible files:

//entered role as student for a teacher/



Browser address bar: <http://localhost/lab/isaalab5.html>

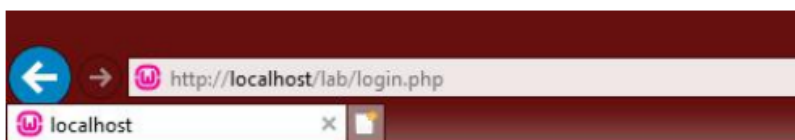
localhost

UserName:

Password:

role:

Submit



Browser address bar: <http://localhost/lab/login.php>

localhost

INVALID