PRN: 2241032

Practical 8

Title: Create a login form with two text fields called “login” and “password”.
When user enters “Galgotias” as a user name and “university” as a password it should be redirected to a Welcome.HTML page or to Sorry.HTML in case of wrong username/password.

Theory:

HTML Forms:

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax −

```
<form action = "Script URL" method = "GET|POST">
   form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

Sr.No	Attribute & Description
1	**action** Backend script ready to process your passed data.
2	**method** Method to be used to upload data. The most frequently used are GET and POST methods.
3	**target** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

4	**enctype** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are − application/x-www-form-urlencoded − This is the standard method most forms use in simple scenarios. mutlipart/form-data − This is used when you want to upload binary data in the form of files like image, word file etc.

Text Input Controls:

There are three types of text input used on forms −

- **Single-line text input controls** − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.
- **Password input controls** − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Password input controls:

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to password.

Example

Here is a basic example of a single-line password input used to take user password -

```
<!DOCTYPE html>
<html>
  <head>
    <title>Password Input Control</title>
  </head>
  <body>
    <form >        User ID : <input type = "text" name = "user_id" /><br>
                Password: <input type = "password" name = "password" />
    </form>
  </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for <input> tag for creating password field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for password input control it will be set to password.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
```

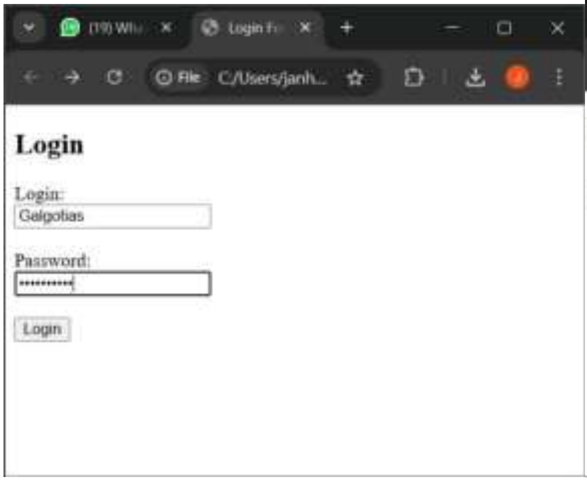
```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
    <script>
        function validateLogin() {
            var username = document.getElementById("username").value;
            var password = document.getElementById("password").value;

            // Check if username and password match the correct values
            if (username === "Galgotias" && password === "university") {
                window.location.href = "Welcome.html"; // Redirect to Welcome.html
            } else {
                window.location.href = "Sorry.html"; // Redirect to Sorry.html
            }
        }
    </script>
</head>
<body>
    <h2>Login</h2>
    <form onsubmit="event.preventDefault(); validateLogin();">
        <label for="username">Login:</label><br>
        <input type="text" id="username" required><br><br>

        <label for="password">Password:</label><br>
        <input type="password" id="password" required><br><br>

        <button type="submit">Login</button>
    </form>
</body>
</html>
```

Output:

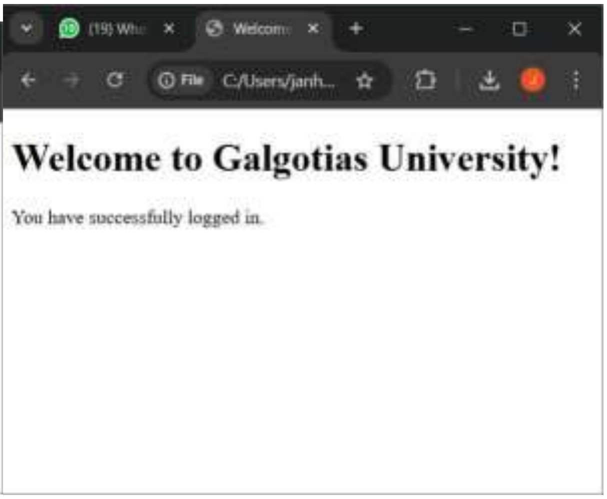


Login

Login:
Galgotias

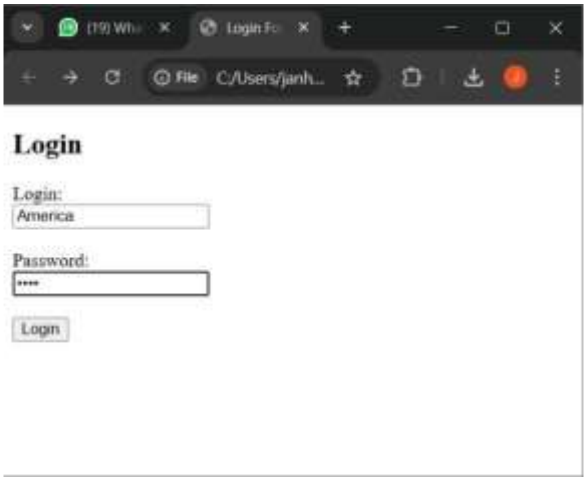
Password:
••••••••••

Login



Welcome to Galgotias University!

You have successfully logged in.

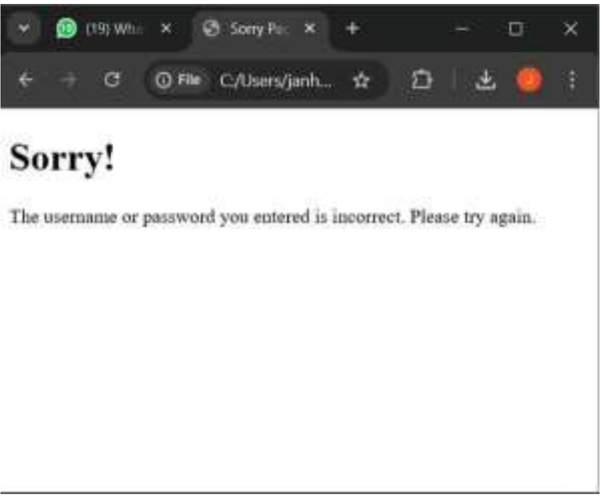


Login

Login:
America

Password:
••••

Login



Sorry!

The username or password you entered is incorrect. Please try again.

Conclusion:

Hence, created a login form with two text fields called “login” and “password”.When user enters “Galgotias” as a user name and “university” as a password it is redirected to a Welcome.HTML page or to Sorry.HTML in case of wrong username/password.