# Chapter 8

# Introducing Geolocation and Offline Support for Data

## Activity 8.1: Implementing Offline Support

### Problem Statement

You have been assigned a task to create the Web page, Register, in the ShowOnWheels website displaying two forms, Create an Account and Sign In. The Create an Account form contains the Name, Password, and Confirm Password fields. The Sign In form contains the Name and Password fields. You need to implement such functionality in the Create an Account form that upon filling the form, the details of the user should not be lost, if the browser is closed. In addition, when the user enters the name in the Name field in the Sign In form, the value in the Password field should be fetched automatically.

How will you perform this task?

**Prerequisite**: To perform this activity, you need to use the solution of **ShowOnWheels** that you have created in Activity 7.2. Ask your faculty to provide you the same.

### Solution

To store the users’ details locally, you need to perform the following tasks:

1. Create the login.html file.
2. Update the nav.html file.
3. View the login.html file.

#### Task 1: Creating the login.html File

To create the **login.html** file, you need to perform the following steps:

1. Open Notepad.
2. Add the following code snippet:

<HTML>

<HEAD>

<LINK type="text/css" rel="stylesheet" href="ExternalStylesheet\homestylesheet.css" />

<STYLE type="text/css">

th{

color:orangered;

}

#form2{

position:absolute;

top:50px;

left:400px;

width:380px;

height:200px;

border:3px solid orangered;

margin-left:190px;

}

#form1{

width:380px;

height:200px;

border:3px solid orangered;

margin-left:60px;

margin-top:50px;

}

button{

margin-left:80px;

color:white;

background-color:orangered;

}

</STYLE>

</HEAD>

<BODY>

<FORM id="form1" name="Sign Up" onsubmit="checkPassword()">

<TABLE>

<TR>

<TH colspan="2"><B><I>Create an Account</I></B></TH>

</TR><TR>

<TD>Name:</TD><TD> <Input type="text" name="username" onkeyup="checkName();" required></TD>

</TR>

<TR>

<TD>Password:</TD><TD> <Input type="password" name="pass1" required></TD>

</TR>

<TR>

<TD>Confirm Password:</TD><TD> <Input type="password" name="pass2" required></TD>

</TR>

<TR><TD><BR><Button type="Submit">Register</TD></TR>

</TABLE>

</FORM>

<FORM id="form2" name="Sign In" onsubmit="onSubmit();">

<TABLE>

<TR>

<TH colspan="2"><B><I>Sign In</I></B></TH>

</TR><TR>

<TD>Name:</TD><TD> <Input type="text" name="username1" onkeyup="checkName();" onchange="login();" required></TD>

</TR>

<TR>

<TD>Password:</TD><TD> <Input type="password" name="pass3" required ></TD>

</TR>

<TR><TD><BR><Button type="Submit">Sign In</TD></TR></TABLE>

</FORM>

</BODY></HTML>

The preceding code snippet creates two forms named Sign Up and Sign In and defines the styles to be applied on the form elements.

1. Add the following code snippet after the <HEAD> tag:

<SCRIPT>

function checkName()

      {

       if (!form1.username.value.match(/^[a-zA-Z]+$/) && form1.username.value !="")

           {

              form1.username.value="";

              form1.username.focus();

              alert("Please enter only alphabets in text");

            }

      }

      function checkPassword()

      {

      if(form1.pass1.value!=form1.pass2.value)

      {

        alert("Passwords entered do not match");

      }

      else

      if (typeof (Storage) !== "undefined") {

            localStorage.userid = form1.username.value;

            localStorage.passwrd = form1.pass1.value;

            alert("Registered");

      }

      }

     function login()

     {

          if (typeof (Storage) !== "undefined")

          {

          if (form2.username1.value== localStorage.userid) {

            form2.pass3.value=localStorage.passwrd;

      }

        }

      }

    function onSubmit()

    {

    if((form2.username1.value!="")&&(form2.pass3.value==localStorage.passwrd))

      alert("Signed In");

    else

      alert("Password do not match");

    }

</SCRIPT>

The preceding code snippet creates functions that implement local storage on the forms.

1. Save the file with the name **login.html** in the **ShowOnWheels** folder.
2. Close the **login.html** file.

#### Task 2: Updating the nav.html File

To update the **nav.html** file, you need to perform the following steps:

1. Open the **nav.html** file.
2. Add the following code snippet after the <A href="Offers.html" target="frame5">Offers</A> | line of code:

<A href="login.html" target="frame5">Register</A> |

The preceding code snippet creates a link for the Register Web Page on the home page.

1. Save and close the **nav.html** file.

#### Task 3: Viewing the login.html File

To view the **login.html** file, you need to perform the following steps:

1. Open Google Chrome.
2. Type **D:\HTML\ShowOnWheels\mainframe.html** in the address bar.
3. Press the **Enter** key. The home page appears.
4. Click the **Register** link. The **Register** Web page appears, as shown in the following figure.

*The Register Web Page*

1. Type **John** in the **Name** field in the **Create an Account** form.
2. Type **Password@1234** in the **Password** field.
3. Type **Password@1234** in the **Confirm Password** field.
4. Click the **Register** button. An alert message appears, as shown in the following figure:

An Alert Message

1. Click the **Ok** button.
2. Type **John** in the **Name** field of the **Sign In** form.
3. Press the **Tab** key. The password automatically gets filled in the **Password** field, as shown in the following figure.

*The Create an Account and Sign In Forms*

1. Close the Web browser.

Now, again open the **Register** Web page and type **John** in the **Name** field of the **Sign In** form. The password automatically gets filled in the **Password** field. This shows that even if the browser is closed, the users’ details are not lost.

# Exercises

### Exercise 1

You have been assigned a task to create a Web page, Sign In/Up, in the SkyLight University website. The Sign In/Up Web page contains two forms, Create an Account and Sign In. The Sign Up form contains the Name, Password, and Confirm Password fields. The Create an Account form contains the Name and Password fields. You need to implement functionality in the Create an Account form in such a way that the users’ details are not stored locally if the browser is closed. However, if the browser is not closed, the users’ details should be stored locally and the password gets automatically filled in the Password field of the Sign In form when the user enters the name. The Sign In/Up Web page is shown in the following figure.

*The Sign In/Up Web Page*

How will you perform this task?

**Prerequisite**: To perform this exercise, you need to use the solution file of Exercise 7.2.

### Exercise 2

You have been assigned a task to display the highest score of the game. This implies that even if the user closes the browser, the highest score should get stored locally and should be displayed when the user next time plays the game, as shown in the following figure.

*The Game Screen Displaying the Highest Score*

How will you perform this task?

**Prerequisite**: To perform this exercise, you need to use the solution of Activity 6.2.