

EXPERIMENT – 4

AIM: To create an external CSS file and style all pages with same background etc. and NAV tag and create an ordered list and unordered list, checkbox and radio button.

Nav →

```
<body>
  <div class="fixed-header">
    <div class="container">
      <nav>
        <a href="index.html">Home</a>
        <a href="signup.html">Sign Up</a>
        <a href="shoppingcart/index.html">Products</a>
        <a href="allservices.html">Services</a>
        <a href="#cont_section">Contact Us</a>
      </nav>
    </div>
  </div>
<div class="container">
```

in css→

```
.container{
  width: 80%;
  margin: 0 auto;
}
.fixed-header, .fixed-footer{
  width: 100%;
  position: fixed;
  background: #333;
  padding: 10px 0px;
  color: #fff;
}
.fixed-header{
  top: 0;
}
.fixed-footer{
  bottom: 0;
}
nav a{
  color: #fff;
  text-decoration: none;
  padding: 7px 80px;
  display: inline-block;
}
```



Ordered list and unordered list, checkbox and radio button. →

<h3 align='center'>SIGN UP HERE</h3>

```
<form name="signupform" align='center' onsubmit="return validateform();" method="post" >
First name:<br>
<input type="text" name="firstname"><br>
Last name:<br>
<input type="text" name="lastname">
<br>
Email:<br>
<input type="text" name="email">
<br>
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other <br>
Which Servies would you like to avail:<br>
<input type="checkbox" name="service" value="InfluencerMarketing"> Influencer Marketing<br>
<input type="checkbox" name="service" value="OtherServices" "> Other Services<br><br>
Password: <input type="password" name="password"><br/>
<br>
<input type="submit" value="submit">
</form>
```

```
<h4>Terms and Conditions:</h2>
<ol>
<li>Your Information is secure with us.</li>
<li>We use your information for:
<ul>
<li>Sending you better offers</li>
<li>Contacting you for business collabrations</li>
</ul>
</li>
</ol>
```

```
<section id="cont_section"></section>
<div class="fixed-footer">
<div align=center class="container">Copyright &copy; 2018 Developed by Akshay -
akshaykhatter@yahoo.in</div>
</div>
</body>
</html>
```

SIGN UP HERE

First name:
Last name:

☒ Male
☐ Female
☐ Other

Which Servies would you like to avail:
☐ Influencer Marketing
☐ Other Services

Submit

Terms and Conditions:

1. Your Information is secure with us.
2. We use your information for:
 - Sending you better offers
 - Contacting you for business collabrations

EXPERIMENT – 5

AIM: Send data to backend(MYSQL) using JSP.

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. JavaServer Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with `<%` and end with `%>`.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically.

Elements of JSP

The elements of JSP have been described below –

The Scriptlet

A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.

Following is the syntax of Scriptlet –

```
<% code fragment %>
```

JSP Declarations

A declaration declares one or more variables or methods that you can use in Java code later in the JSP file. You must declare the variable or method before you use it in the JSP file.

Following is the syntax for JSP Declarations –

```
<%! declaration; [ declaration; ]+ ... %>
```

JSP Directives

A JSP directive affects the overall structure of the servlet class. It usually has the following form –

```
<%@ directive attribute="value" %>
```

JSP Actions

JSP actions use **constructs** in XML syntax to control the behavior of the servlet engine. You can dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin.

There is only one syntax for the Action element, as it conforms to the XML standard –

```
<jsp:action_name attribute="value" />
```

In signup.html we add the action as →

```
<h3 align='center'>SIGN UP HERE</h3>
```

```
<form align='center' method="get" action="signup1.jsp">  
First name:<br>  
<input type="text" name="firstname"><br>  
Last name:<br>  
<input type="text" name="lastname">  
<br>  
<input type="radio" name="gender" value="male" checked> Male<br>  
<input type="radio" name="gender" value="female"> Female<br>  
<input type="radio" name="gender" value="other"> Other <br>  
Which Services would you like to avail:<br>  
<input type="checkbox" name="service" value="InfluencerMarketing"> Influencer Marketing<br>  
<input type="checkbox" name="service" value="OtherServices" "> Other Services<br>  
<br>  
<input type="submit" value="Submit">  
</form>
```

signup1.jsp →

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"  
pageEncoding="ISO-8859-1"%>  
<%@page import="java.sql.*,java.util.*"%>
```

```
<%  
String firstname=request.getParameter("firstname");  
String lastname=request.getParameter("lastname");  
String gender=request.getParameter("gender");  
String services=request.getParameter("service");  
  
try  
{  
Class.forName("com.mysql.jdbc.Driver");  
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Influencers", "root", "");  
Statement st=conn.createStatement();  
  
int i=st.executeUpdate("insert into  
user(firstname,lastname,gender,services)values('"+firstname+"','"+lastname+"','"+gender+"','"+services+"')  
");  
out.println("Data is successfully inserted!");  
}  
catch(Exception e)  
{  
System.out.print(e);  
e.printStackTrace();  
}  
%>
```



+ Options

firstname	lastname	gender	services
AKSHAY	khatter	male	InfluencerMarketing
Bhavy	Khatter	male	InfluencerMarketing
Bhavy	Khatter	male	OtherServices

EXPERIMENT – 6

AIM: Retrieve data from backend(MYSQL) using JSP and display it on screen.

showData.jsp →

```
<%--
    Document   : showData
    Created on  : Mar 31, 2018, 12:45:18 AM
    Author      : akshaykhatter
--%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page import="java.sql.*" %>
<%@ page import="java.io.*" %>
<html>
<head>
<title>display data from the table using jsp</title>
</head>
<body>
<h2>Data from the table 'user' of database 'Influencers'</h2>
<%
try {
/* Create string of connection url within specified format with machine
name, port number and database name. Here machine name is localhost and
database name is student. */
String connectionURL = "jdbc:mysql://localhost:3306/Influencers";
// declare a connection by using Connection interface
Connection connection = null;
/* declare object of Statement interface that is used for executing sql
statements. */
Statement statement = null;
// declare a resultset that uses as a table for output data from the table.
ResultSet rs = null;
// Load JDBC driver "com.mysql.jdbc.Driver"
Class.forName("com.mysql.jdbc.Driver").newInstance();
/* Create a connection by using getConnection() method that takes parameters
of string type connection url, user name and password to connect to database.*/
connection = DriverManager.getConnection(connectionURL, "root", "");
/* createStatement() is used for create statement object that is used for
sending sql statements to the specified database. */
statement = connection.createStatement();
// sql query to retrieve values from the specified table.
String QueryString = "SELECT * from user";
rs = statement.executeQuery(QueryString);
%>
<TABLE cellpadding="15" border="1" style="background-color: #ffffcc;">
<%
while (rs.next()) {
%>
<TR>
<TD><%=rs.getString(1)%></TD>
<TD><%=rs.getString(2)%></TD>
<TD><%=rs.getString(3)%></TD>
<TD><%=rs.getString(4)%></TD>
</TR>
<% } %>
<%
// close all the connections.
rs.close();
statement.close();
connection.close();
```

```
} catch (Exception ex) {  
%>  
</font>  
<font size="+3" color="red"></b>  
<%  
out.println("Unable to connect to database.");  
}  
%>  
</TABLE>  
  
<TABLE>  
<TR>  
<TD><FORM ACTION="index.html" method="get" >  
<button type="submit"><-- back</button></TD>  
</TR>  
</TABLE>  
</font>  
</body>  
</html>
```

The screenshot shows a web browser with two tabs. The left tab is titled 'Sign Up Form' and displays a registration form. The right tab is titled 'display data...' and displays a table of data from the 'user' table of a database named 'Influencers'.

SIGN UP HERE

First name:

Last name:

☒ Male
☐ Female
☐ Other

Which Services would you like to avail:
☐ Influencer Marketing
☐ Other Services

Terms and Conditions:

1. Your Information is secure with us.
2. We use your information for:
 - Sending you better offers
 - Contacting you for business collaborations

[Show Existing Influencers and Brands](#)

Data from the table 'user' of database 'Influencers'

Akshay	Khatter	male	InfluencerMarketing
Rohan	Sharma	male	OtherServices
Ishita	Sharma	female	InfluencerMarketing

EXPERIMENT – 7

AIM: To do form validation using JavaScript.

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

What is JavaScript ?

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Advantages of JavaScript

The merits of using JavaScript are –

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multithreading or multiprocessor capabilities.
- JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>**.

```
<script ...>  
    JavaScript code  
</script>
```

[validate.js](#) →

```
function validateform(){  
    var firstname=document.signupform.firstname.value;  
    var lastname=document.signupform.lastname.value;
```

```
var password=document.signupform.password.value;
var x=document.signupform.email.value;
var atposition=x.indexOf("@");
var dotposition=x.lastIndexOf(".");

if (firstname==null || firstname=="" || lastname==null || lastname==""){
    alert("Name can't be blank");
    return false;
}
else if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
    alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
    return false;
}
else if(password.length<6){
    alert("Password must be at least 6 characters long.");
    return false;
}
}
```

The image displays three screenshots of a web form titled "SIGN UP HERE". The form includes input fields for "First name:", "Last name:", and "Email:", along with radio buttons for "Male", "Female", and "Other", and checkboxes for "Influencer Marketing" and "Other Services".

- Top Left Screenshot:** Shows the form with empty input fields. A white error box at the bottom displays the message "Name can't be blank" and a "Close" button.
- Top Right Screenshot:** Shows the form with "First name:" filled as "akshay", "Last name:" filled as "khatter", and "Email:" filled as "aaan@g.c". A white error box at the bottom displays the message "Please enter a valid e-mail address" followed by "atpostion:4" and "dotposition:6" (note the typo in the image), and a "Close" button.
- Bottom Screenshot:** Shows the form with "First name:" filled as "akshay", "Last name:" filled as "khatter", and "Email:" filled as "akshay@gmail.com". The "Male" radio button is selected, and "Influencer Marketing" is checked. A white error box at the bottom displays the message "Password must be at least 6 characters long." and a "Close" button.

EXPERIMENT – 8

AIM: XML data display and styling using XSLT stylesheet.

THEORY:

- XML stands for eXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.
- XSL (eXtensible Stylesheet Language) is a styling language for XML.
- XSLT stands for XSL Transformations.
- The World Wide Web Consortium (W3C) started to develop XSL because there was a need for an XML-based Stylesheet Language.

hello.xml →

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="abc.xsl"?>
<books>
  <book>
    <name>John Abraham</name>
    <category>Fitness</category>
    <location>Mumbai</location>
    <followers>10 m</followers>
  </book>
  <book>
    <name>Priyanka Chopra</name>
    <category>Fashion</category>
    <location>New York</location>
    <followers>16 m</followers>
  </book>
  <book>
    <name>Shahid Kapoor</name>
    <category>Travel</category>
    <location>Mumbai</location>
    <followers>8 m</followers>
  </book>
  <book>
    <name>Kriti Sanon</name>
    <category>Lifestyle</category>
    <location>Delhi</location>
    <followers>6 m</followers>
  </book>
</books>
```

abc.xsl →

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8"/>
<xsl:template match="/">
<html>
<head><title>Books</title>
<style>
body {
  background: lightblue url("images/123.jpg") no-repeat fixed center;
}
.fixed-header, .fixed-footer{
  width: 100%;
  position: fixed;
  background: #333;
  padding: 10px 0px;
  color: #fff;
}
.fixed-header{
```

```

    top: 0; }
.fixed-footer{
    bottom: 0; }
nav a{
    color: #fff;
    text-decoration: none;
    padding: 7px 90px;
    display: inline-block;
}
</style>
</head>
<body >
    <div align="center" class="fixed-header">
    <div class="container">
        <nav>
            <a href="index.html">Home</a>
            <a href="signup.html">Sign Up</a>
            <a href="shoppingcart/index.html">Products</a>
            <a href="allservices.html">Services</a>
            <a href="#cont_section">Contact Us</a>
        </nav>
    </div>
    <div align="center" style="margin-left:20%;margin-right:20%;margin-top:10%;">
    <h1>FULL LIST</h1>
    <table width="100%" border="1">
    <THEAD>
        <TR>
            <TD width="35%"><B>Name</B></TD>
            <TD width="15%"><B>Category</B></TD>
            <TD width="10%"><B>Location</B></TD>
            <TD width="10%"><B>Followers</B></TD>
        </TR>
    </THEAD>
    <TBODY>
        <xsl:for-each select="books/book">
            <TR>
                <TD width="35%"><xsl:value-of select="name" /></TD>
                <TD width="15%"><xsl:value-of select="category" /></TD>
                <TD width="10%"><xsl:value-of select="location" /></TD>
                <TD width="10%"><xsl:value-of select="followers" /></TD>
            </TR>
        </xsl:for-each>
    </TBODY>
    </table>
    </div>

</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Home

Sign Up

Products

Services

Contact Us

FULL LIST

Name	Category	Location	Followers
John Abraham	Fitness	Mumbai	10 m
Priyanka Chopra	Fashion	New York	16 m
Shahid Kapoor	Travel	Mumbai	8 m
Kriti Sanon	Lifestyle	Delhi	6 m

EXPERIMENT – 9

AIM: Do calculation using JavaScript. (Make a Shopping Cart)

shoppingcart/index.html →

```
<!DOCTYPE html>
<html>
<head>
<title>Digital Marketing</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
<script type="text/javascript" src="js/jquery.shop.js"></script>
</head>

<body>

<div id="site">
<header id="masthead">
<div class="fixed-header">
<div class="container">
<nav>
<a href=" ../index.html">Home</a>
<a href=" ../signup.html">Sign Up</a>
<a href="#">Products</a>
<a href=" ../allservices.html">Services</a>
<a href=" ../index.html">Contact Us</a>
</nav>
</div>
</div>

<h1>Digital Marketing<span class="tagline">Specializing in Marketing since 1999</h1>
</header>
<div id="content">
<div id="products">
<ul>
<li>
<div class="product-image">

</div>
<div class="product-description" data-name="SEO" data-price="5">
<h3 class="product-name">Search Engine Optimization</h3>
<p class="product-price">&euro; 19</p>
<form class="add-to-cart" action="cart.html" method="post">
<div>
<label for="qty-1">Durations (Months)</label>
<input type="text" name="qty-1" id="qty-1" class="qty" value="1" />
</div>
<p><input type="submit" value="Add to cart" class="btn" /></p>
</form>
</div>
</li>

<li>
<div class="product-image">

</div>
<div class="product-description" data-name="Influencer Marketing" data-price="8">
<h3 class="product-name">Influencer Marketing</h3>
<p class="product-price">&euro; 29</p>
<form class="add-to-cart" action="cart.html" method="post">
<div>
<label for="qty-2">Durations (Months)</label>
```

```
<input type="text" name="qty-2" id="qty-2" class="qty" value="1" />
</div>
<p><input type="submit" value="Add to cart" class="btn" /></p>
</form>
</div>
</li>
<li>
<div class="product-image">

</div>
<div class="product-description" data-name="Email Marketing" data-price="11">
<h3 class="product-name">Email Marketing & Automation</h3>
<p class="product-price">&euro; 39</p>
<form class="add-to-cart" action="cart.html" method="post">
<div>
<label for="qty-3">Durations (Months)</label>
<input type="text" name="qty-3" id="qty-3" class="qty" value="1" />
</div>
<p><input type="submit" value="Add to cart" class="btn" /></p>
</form>
</div>
</li>
</ul>
</div>
</div>
</div>
<footer id="site-info">
Copyright &copy; SplashFluence
</footer>
</body>
</html>
```

shoppingcart/cart.html →

```
<!DOCTYPE html>
<html>
<head>
<title>Digital Marketing</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
<script type="text/javascript" src="js/jquery.shop.js"></script>
</head>

<body>

<div id="site">
<header id="masthead">
<h1>Digital Marketing<span class="tagline">Specializing in Marketing since 1999</h1>
</header>
<div id="content">
<h1>Your Shopping Cart</h1>
<form id="shopping-cart" action="cart.html" method="post">
<table class="shopping-cart">
<thead>
<tr>
<th scope="col">Item</th>
<th scope="col">Qty</th>
<th scope="col" colspan="2">Price</th>
</tr>
</thead>
<tbody>

</tbody>
</table>
<p id="sub-total">
```

```
        <strong>Sub Total</strong>: <span id="stotal"></span>
    </p>
    <p id="sub-total">
        <strong>Discount</strong>: <span id="stotal"></span>
    </p>
    <ul id="shopping-cart-actions">
        <li>
            <input type="submit" name="update" id="update-cart" class="btn" value="Update Cart" />
        </li>
        <li>
            <input type="submit" name="delete" id="empty-cart" class="btn" value="Empty Cart" />
        </li>
        <li>
            <a href="index.html" class="btn">Continue Shopping</a>
        </li>
        <li>
            <a href="#" class="btn">Go To Checkout</a>
        </li>
    </ul>
</form>
</div>

</div>

<footer id="site-info">
Copyright &copy; SplashFluence
</footer>

</body>
</html>
```

shoppingcart/js/jquery.shop.js →

```
(function( $ ) {
$.Shop = function( element ) {
this.$element = $( element );
this.init();
};

$.Shop.prototype = {
init: function() {
// Properties
this.cartPrefix = "Scart-"; // Prefix string to be prepended to the cart's name in the session storage
this.cartName = this.cartPrefix + "cart"; // Cart name in the session storage
this.shippingRates = this.cartPrefix + "shipping-rates"; // Shipping rates key in the session storage
this.total = this.cartPrefix + "total"; // Total key in the session storage
this.storage = sessionStorage; // shortcut to the sessionStorage object

this.$formAddToCart = this.$element.find( "form.add-to-cart" ); // Forms for adding items to the cart
this.$formCart = this.$element.find( "#shopping-cart" ); // Shopping cart form
this.$subTotal = this.$element.find( "#stotal" ); // Element that displays the subtotal charges
this.$shoppingCartActions = this.$element.find( "#shopping-cart-actions" ); // Cart actions links
this.$updateCartBtn = this.$shoppingCartActions.find( "#update-cart" ); // Update cart button
this.$emptyCartBtn = this.$shoppingCartActions.find( "#empty-cart" ); // Empty cart button

this.currency = "&euro;"; // HTML entity of the currency to be displayed in the layout
this.currencyString = "€"; // Currency symbol as textual string

this.createCart();
this.handleAddToCartForm();
this.emptyCart();
this.updateCart();
this.displayCart();
this.deleteProduct();
},
```

```
createCart: function() {
    if( this.storage.getItem( this.cartName ) == null ) {

        var cart = {};
        cart.items = [];

        this.storage.setItem( this.cartName, this._toJSONString( cart ) );
        this.storage.setItem( this.shippingRates, "0" );
        this.storage.setItem( this.total, "0" );
    }
},

deleteProduct: function() {
    var self = this;
    if( self.$formCart.length ) {
        var cart = this._toJSONObject( this.storage.getItem( this.cartName ) );
        var items = cart.items;

        $( document ).on( "click", ".pdelete a", function( e ) {
            e.preventDefault();
            var productName = $( this ).data( "product" );
            var newItems = [];
            for( var i = 0; i < items.length; ++i ) {
                var item = items[i];
                var product = item.product;
                if( product == productName ) {
                    items.splice( i, 1 );
                }
            }
            newItems = items;
            var updatedCart = {};
            updatedCart.items = newItems;

            var updatedTotal = 0;
            var totalQty = 0;
            if( newItems.length == 0 ) {
                updatedTotal = 0;
                totalQty = 0;
            } else {
                for( var j = 0; j < newItems.length; ++j ) {
                    var prod = newItems[j];
                    var sub = prod.price * prod.qty;
                    updatedTotal += sub;
                    totalQty += prod.qty;
                }
            }

            self.storage.setItem( self.total, self._convertNumber( updatedTotal ) );
            self.storage.setItem( self.shippingRates, self._convertNumber( self._calculateShipping( totalQty ) ) );

            self.storage.setItem( self.cartName, self._toJSONString( updatedCart ) );
            $( this ).parents( "tr" ).remove();
            self.$subTotal[0].innerHTML = self.currency + " " + self.storage.getItem( self.total );
        });
    }
},

displayCart: function() {
    if( this.$formCart.length ) {
        var cart = this._toJSONObject( this.storage.getItem( this.cartName ) );
        var items = cart.items;
        var $tableCart = this.$formCart.find( ".shopping-cart" );
        var $tableCardBody = $tableCart.find( "tbody" );

        if( items.length == 0 ) {
            $tableCardBody.html( "" );
        } else {

            for( var i = 0; i < items.length; ++i ) {
                var item = items[i];
                var product = item.product;
                var price = this.currency + " " + item.price;
```

```

        var qty = item.qty;
        var html = "<tr><td class='pname'>" + product + "</td>" + "<td class='pqty'><input type='text'
value='" + qty + "' class='qty'/></td>";
        html += "<td class='pprice'>" + price + "</td><td class='pdelete'><a href='" + product
+ "'>&times;</a></td></tr>";

        $tableCardBody.html( $tableCardBody.html() + html );
    }

}

if( items.length == 0 ) {
    this.$subTotal[0].innerHTML = this.currency + " " + 0.00;
} else {

    var total = this.storage.getItem( this.total );
    this.$subTotal[0].innerHTML = this.currency + " " + total;
}
} else if( this.$checkoutCart.length ) {
    var checkoutCart = this._toJSONObject( this.storage.getItem( this.cartName ) );
    var cartItems = checkoutCart.items;
    var $cartBody = this.$checkoutCart.find( "tbody" );

    if( cartItems.length > 0 ) {

        for( var j = 0; j < cartItems.length; ++j ) {
            var cartItem = cartItems[j];
            var cartProduct = cartItem.product;
            var cartPrice = this.currency + " " + cartItem.price;
            var cartQty = cartItem.qty;
            var cartHTML = "<tr><td class='pname'>" + cartProduct + "</td>" + "<td class='pqty'>" + cartQty +
"</td>" + "<td class='pprice'>" + cartPrice + "</td></tr>";

            $cartBody.html( $cartBody.html() + cartHTML );
        }
    } else {
        $cartBody.html( "" );
    }

    if( cartItems.length > 0 ) {

        var cartTotal = this.storage.getItem( this.total );
        var subTot = this._convertString( cartTotal ) + this._convertString( cartShipping );

        this.$subTotal[0].innerHTML = this.currency + " " + this._convertNumber( subTot );
    } else {
        this.$subTotal[0].innerHTML = this.currency + " " + 0.00;
    }

}
},

emptyCart: function() {
    var self = this;
    if( self.$emptyCartBtn.length ) {
        self.$emptyCartBtn.on( "click", function() {
            self._emptyCart();
        });
    }
},

updateCart: function() {
    var self = this;
    if( self.$updateCartBtn.length ) {
        self.$updateCartBtn.on( "click", function() {
            var $rows = self.$formCart.find( "tbody tr" );
            var cart = self.storage.getItem( self.cartName );
            var total = self.storage.getItem( self.total );

            var updatedTotal = 0;
            var totalQty = 0;
            var updatedCart = {};
            updatedCart.items = [];

```

```
$rows.each(function() {
    var $row = $( this );
    var pname = $.trim( $row.find( ".pname" ).text() );
    var pqty = self._convertString( $row.find( ".pqty > .qty" ).val() );
    var pprice = self._convertString( self._extractPrice( $row.find( ".pprice" ) ) );

    var cartObj = {
        product: pname,
        price: pprice,
        qty: pqty
    };

    updatedCart.items.push( cartObj );

    var subTotal = pqty * pprice;
    updatedTotal += subTotal;
    totalQty += pqty;
});

self.storage.setItem( self.total, self._convertNumber( updatedTotal ) );
self.storage.setItem( self.cartName, self._toJSONString( updatedCart ) );

});
},

handleAddToCartForm: function() {
var self = this;
self.$formAddToCart.each(function() {
    var $form = $( this );
    var $product = $form.parent();
    var price = self._convertString( $product.data( "price" ) );
    var name = $product.data( "name" );

    $form.on( "submit", function() {
        var qty = self._convertString( $form.find( ".qty" ).val() );
        var subTotal = qty * price;
        var total = self._convertString( self.storage.getItem( self.total ) );
        var sTotal = total + subTotal;
        self.storage.setItem( self.total, sTotal );
        self._addToCart({
            product: name,
            price: price,
            qty: qty
        });
    });
});
},

_emptyCart: function() {
    this.storage.clear();
},

_extractPrice: function( element ) {
    var self = this;
    var text = element.text();
    var price = text.replace( self.currencyString, "" ).replace( " ", "" );
    return price;
},

_toJSONObject: function( str ) {
    var obj = JSON.parse( str );
    return obj;
},

_toJSONString: function( obj ) {
    var str = JSON.stringify( obj );
    return str;
},

_addToCart: function( values ) {
    var cart = this.storage.getItem( this.cartName );
```



```
var cartObject = this._toJSONObject( cart );  
var cartCopy = cartObject;  
var items = cartCopy.items;  
items.push( values );  
  
this.storage.setItem( this.cartName, this._toJSONString( cartCopy ) );  
  
},  
  
};  
  
$(function() {  
var shop = new $.Shop( "#site" );  
});  
  
})( jQuery );
```

