

# Page Layout and CSS

## Tutorial – 1, HTML, CSS, Servlets and JSP

**CSP 595 – Enterprise Web Application**

**Dr. Atef Bader**

**Illinois Institute of Technology**

-Presentation by  
Sarath Kumar Prabhakaran  
sprabha7@hawk.iit.edu

# HTML:

- HTML is a markup language used to define the content of a web page
- The contents are described using HTML tags
- There are different kind of tags to describe different elements of a web page such as headings, paragraphs, images, tables, form etc.
- A html document is saved with '.html' extension
- Create a new text file, type in the following code and save it with a name (Eg. firstPage.html)
- Double click on the newly created file, this should open a new web page on your default browser

```
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

## HTML Elements:

- HTML elements begin with a start tag and ends with an end tag

`<tagname>content</tagname>`

- The browser renders everything that is within the start and the end tags
- HTML is a forgiving language, it does not generate any error if you misplace any of the tags or do any other mistakes

# HTML General Structure:

- A typical HTML file is based on the following structure
- There are 2 distinct parts
  - Head – Head comprises of a title element which describes the title of the page
  - Body – The body is usually composed of many elements with the intention of showing relevant information to the user

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
...
<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- To learn more on HTML, visit: [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

## Difference between HTML & HTML 5:

- HTML 1.0 - 4.0 were basically to design the layout of static content such as text and images
- HTML 5.0 aims more at cross platform application development. It has got support for including 2D, 3D graphics, playing audio and video
- Following are a few features of HTML 5;
  - New elements: article, aside, audio, bdi, canvas, command, data, datalist, details, embed, figcaption, figure, footer, header, hgroup, keygen, mark, meter, nav, output, progress, rp, rt, ruby, section, source, summary, time, track, video, wbr
  - Deprecated elements: A few older tags will be dropped altogether Eg. acronym, applet, basefont, big, center, dir, font, frame, frameset, isindex, noframes, strike, tt
  - Clear syntax and backward compatibility: Unlike Html 4.0 you don't have to write a longer doctype declaration. You just need to write <!DOCTYPE html> and that's all. It is also designed to be backward compatible.

## CSS:

- CSS, Cascading Style Sheet is used to design the HTML elements
- HTML defines the content of the web page, CSS is used to design the style of the web page
- There are 3 common ways to include CSS in the HTML page;
  - Inline CSS – Style is defined within the HTML element
  - Embedded CSS – The style is defined within the head section of the HTML page
  - External CSS – The CSS is stored in an external file and the reference is given in the head section of the HTML page
- To learn more on CSS, visit:  
[http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp)

## CSS – Inline CSS:

- The inline CSS is included within the HTML element
- Syntax:

```
< HTML tag style = "Style Attribute"> </ HTML tag>
```

- Following is an example of inline CSS, here the 'h1' heading element is given some style

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

## CSS – Embedded CSS:

- In Embedded CSS, the style is included in the head section of the HTML file.
- Syntax:

```
<html>  
  <head>  
    <style>  
      All the style logic is defined here  
    </style>  
  </head>  
  <body>  
  </body>  
</html>
```

- The style logic is incorporated inside the <style> tag. The <style> tag is within the head section



## CSS – Embedded CSS:

- Following is an example of Embedded CSS, here the style is defined in the head section of the html page

```
<head>  
<style>  
body {  
    background-color: linen;  
}  
  
h1 {  
    color: maroon;  
    margin-left: 40px;  
}  
</style>  
</head>
```

## CSS – External CSS:

- Here the CSS styles are written in an external file and is included in the HTML page as a link in the head section
- Create a new file and write the style logic, save that file with a name and '.css' extension
- Include the location of the file in the 'href' as highlighted below

```
<html>  
  <head>  
    <link rel = "stylesheet" type = "text/css" href = "location_of_css_file">  
  </head>  
  <body>  
  </body>  
</html>
```

## CSS – External CSS:

- Create a new file and enter the following, save it as 'mystyle.css'

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

- To include this in the html file, enter the following;

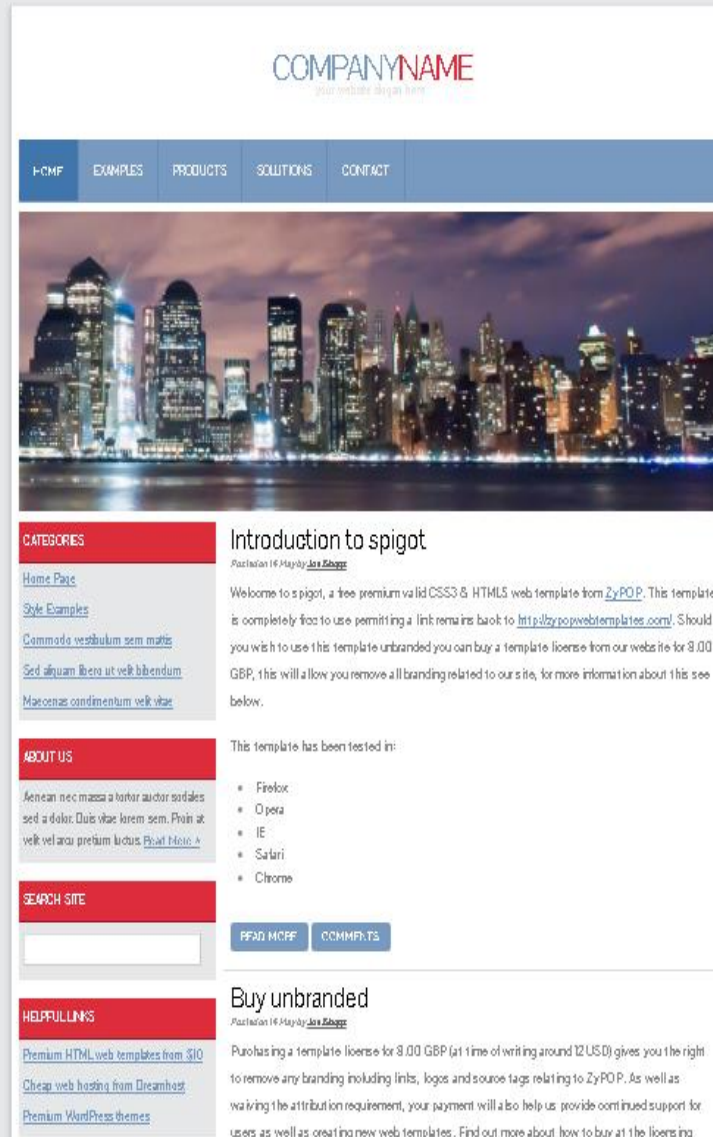
```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

## Servlets and JSPs:

- HTML and CSS together can produce only static web pages, pages that you can see but not interact with
- Technologies such as Servlets, JSP, JavaScript, provide the dynamic aspect to the web page
- Servlet is a Java program that can be used to collect the input from through web page forms, display records from a database or another source and thus provide a means for the user to interact with the web page
- JSP is a technology to create dynamic web page
- JSP has got some advantages over Servlets, it is easier to write and maintain JSP than Servlets
- A typical JSP file can contain both the HTML tags and Java logic

Let's See some Examples of  
CSS Templates

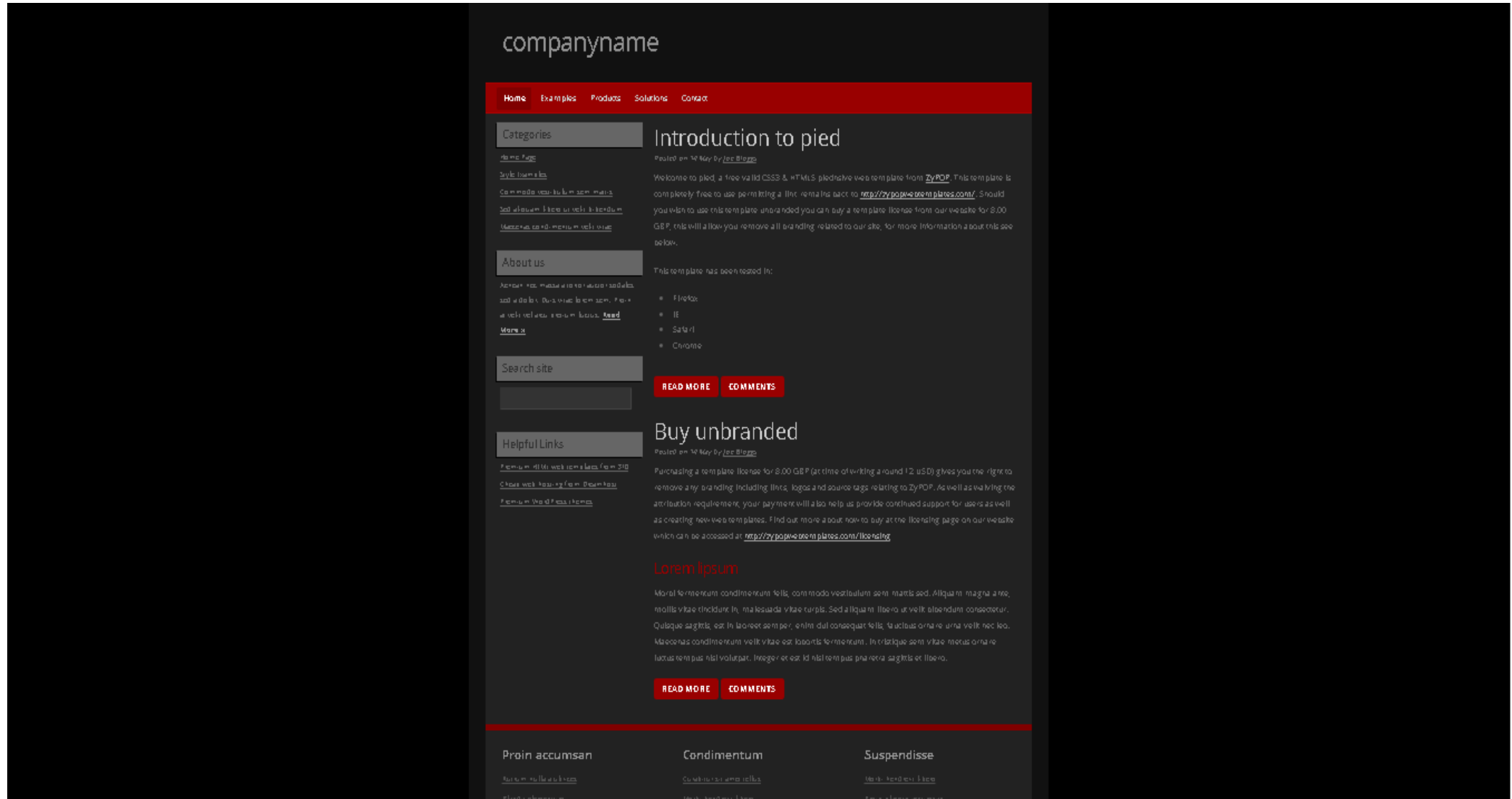
# Template 1 :



# Template 2 :



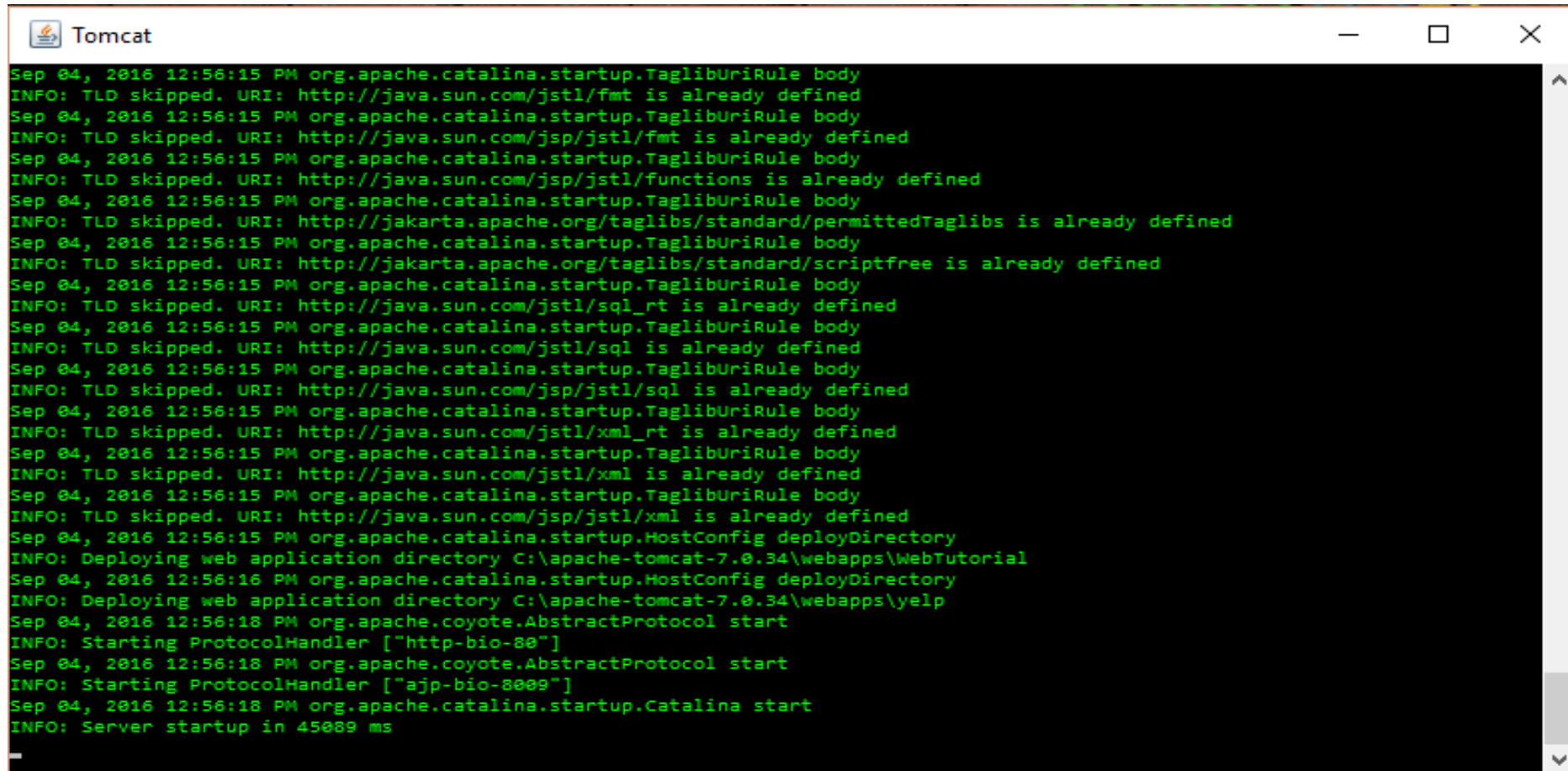
# Template 3 :





## Example:

- Before we begin, make sure that Apache Tomcat are running

A screenshot of a Windows command prompt window titled "Tomcat". The window displays a series of log messages from the Apache Tomcat startup process. The messages are green text on a black background. The logs show the startup of various components, including the TaglibUriRule, HostConfig, and Catalina, and the deployment of web applications. The final message indicates that the server startup took 45089 ms.

```
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jstl/fmt is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jsp/jstl/fmt is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jsp/jstl/functions is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://jakarta.apache.org/taglibs/standard/permittedTaglibs is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://jakarta.apache.org/taglibs/standard/scriptfree is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jstl/sql_rt is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jstl/sql is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jsp/jstl/sql is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jstl/xml_rt is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jstl/xml is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.TaglibUriRule body
INFO: TLD skipped. URI: http://java.sun.com/jsp/jstl/xml is already defined
Sep 04, 2016 12:56:15 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory C:\apache-tomcat-7.0.34\webapps\WebTutorial
Sep 04, 2016 12:56:16 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory C:\apache-tomcat-7.0.34\webapps\yelp
Sep 04, 2016 12:56:18 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-80"]
Sep 04, 2016 12:56:18 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Sep 04, 2016 12:56:18 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 45089 ms
```

# Web Application Home Page:

- welcome-file-list in web.xml : The **welcome-file-list** element of **web-app**, is used to define a list of welcome files. A **welcome file** is the file that is invoked automatically by the server, if you don't specify any file name.

By default server looks for the welcome file in following order:

1. welcome-file-list in web.xml
2. index.html
3. index.htm
4. index.jsp

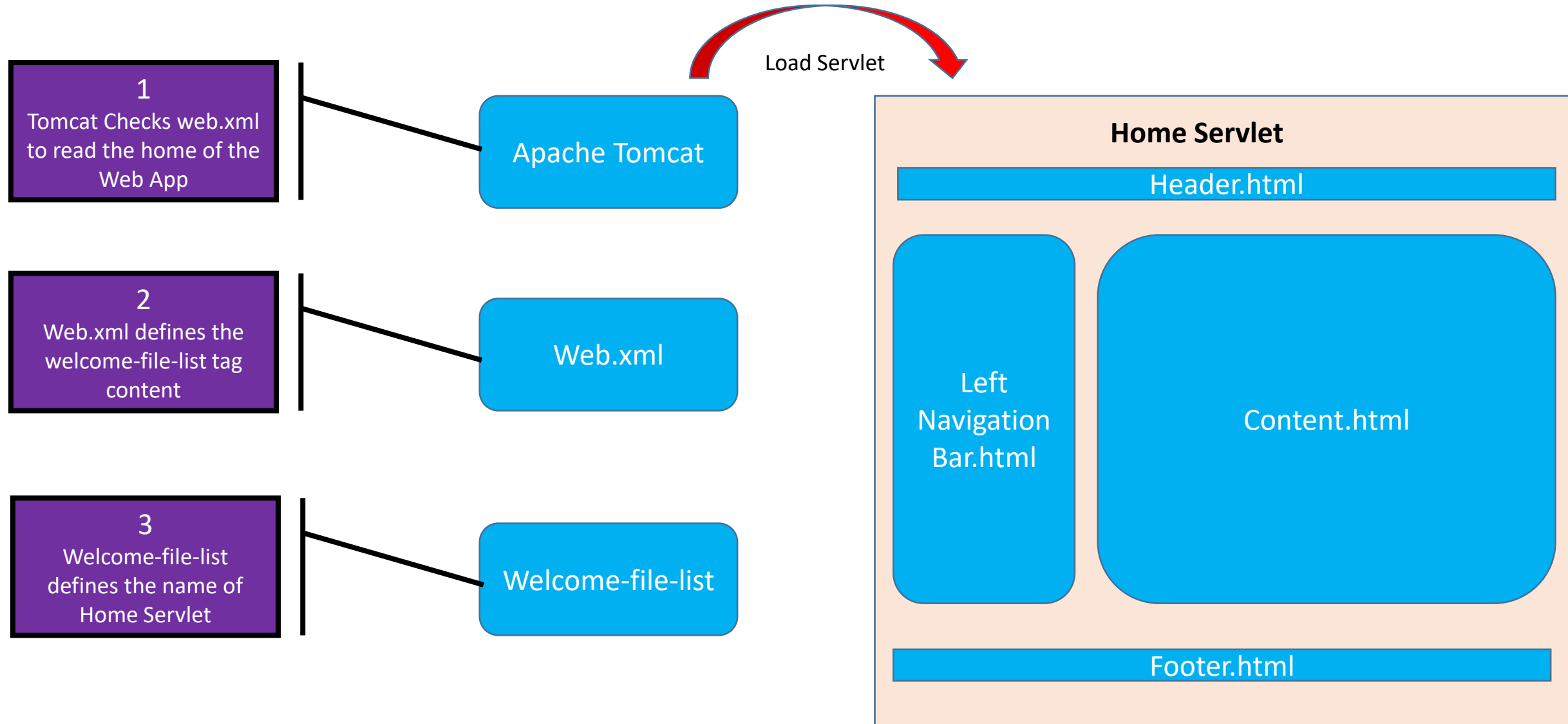
If none of these files are found, server renders 404 error.

- To learn more on welcome-file-list, visit:  
<http://www.javatpoint.com/welcome-file-list>

# How Apache Tomcat loads web App Homepage ?

- **Home** servlet is added in **welcome-file-list** of **web.xml**.
- We have 4 Important HTML pages in our Game Speed Web Application.
- **Header.html** - >It Contains Game Speed Application Logo, Search Box and Top Navigation Bar.
- **Footer.html**->It Contains Copyright Information of the Game Speed Application.
- **LeftNavigationBar.html** -> It contains Different Product Categories available in Game Speed Application (Consoles, Games, Tablets, Accessories).
- **Content.html** -> It Contains Products Available in Game Speed Application, you have the options of Buying and reviewing Product.
- These html pages are printed using the **utilities** function in the **Home** Servlet.

# How Apache tomcat loads the web App Homepage (Diagram) :



# Home Servlet:

```
public class Home extends HttpServlet {  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        response.setContentType("text/html");  
        PrintWriter pw = response.getWriter();  
  
        Utilities utility = new Utilities(request,pw);  
        utility.printHtml("Header.html");  
        utility.printHtml("LeftNavigationBar.html");  
        utility.printHtml("Content.html");  
        utility.printHtml("Footer.html");  
    }  
}
```

# Left NavigationBar.html :

```
<div id="sidebar">
  <ul>
    <li>
      <h2>Consoles</h2>
      <ul>
        <li id="first"><a href="ConsoleList?maker=microsoft">Microsoft</a>
        <li><a href="ConsoleList?maker=sony">Sony</a></li>
        <li><a href="ConsoleList?maker=nintendo">Nintendo</a></li>
      </ul>
    </li>
    <li>
      <h2>Games</h2>
      <ul>
        <li id="first"><a href="GamesList?maker=electronicArts">Electronic
          Arts</a></li>
        <li><a href="GamesList?maker=activision">Activision</a></li>
        <li><a href="GamesList?maker=takeTwoInteractive">Take-Two
          Interactive</a></li>
      </ul>
    </li>
    <li>
      ....
    </li>
  </ul>
</div>
```

For Consoles, ConsoleList servlet is Invoked with variable maker which is assigned a value "sony", when the user clicks Sony from the Left Navigation Bar.

For Games, GamesList servlet is Invoked with variable maker which is assigned the value "Activision", when the user clicks Activision from the Left Navigation Bar.

The href attribute specifies the URL of the page the link goes to.

# ConsoleList Servlet :

```
public class ConsoleList extends HttpServlet {  
    ...  
    PrintWriter pw = response.getWriter();  
    String name = null;  
    String categoryName = request.getParameter("maker");  
    HashMap<String, Console> hm = new HashMap<String, Console>();  
    if(categoryName==null){  
        ...  
    }  
    if(categoryName.equals("sony")){  
        hm.putAll(ConsoleHashMap.sony);  
        name = ConsoleHashMap.string_sony;  
    }  
    else if(categoryName.equals("nintendo")){  
        hm.putAll(ConsoleHashMap.nintendo);  
        name = ConsoleHashMap.string_nintendo; }  
    }  
    ...  
}
```

CategoryName gets the value stored in “maker” variable, With value of CategoryName the servlet checks whether it belongs to sony, Nintendo or Microsoft and then shows the consoles belonging to requested console Manufacturers.

# Content.html :

```
<div id="content">
```

```
....
```

```
<div class="post">
```

```
<div class="entry">
```

```
<table id="bestseller">
```

```
<tr>
```

```
<td><div id="shop_item">
```

```
<h3>XBox One</h3>
```

```
<strong>399.99$</strong>
```

```
<ul>
```

```
<li id="item">
```

```
</li>
```

```
<li><a class="btnreview" href="#">Buy Now</a></li>
```

```
<!--<li><a class="btnbuy"
```

```
href="AccessoryList?maker=microsoft&console=xboxone">View
```

```
Accessories</a></li>-->
```

```
<li><a class="btnreview" href="#">Reviews</a></li>
```

```
</ul>
```

```
</div>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
....
```



# Utilities Servlet:

```
public class Utilities extends HttpServlet{
```

```
...
```

```
public void printHtml(String file) {  
    String result = HtmlToString(file);  
    if (file == "Header.html") {  
        if (session.getAttribute("username")!=null){  
            String username = session.getAttribute("username").toString();  
            username = Character.toUpperCase(username.charAt(0)) + username.substring(1);  
            result = result  
                + "<li><a>Hello, "+username+"</a></li>"  
                + "<li><a href='Account'>Account</a></li>"  
                + "<li><a href='Logout'>Logout</a></li>";  
        }  
        else  
            result = result + "<li><a href='Login'>Login</a></li>";  
        result = result  
            + "<li><a href='Cart'>Cart("+CartCount()+")</a></li></ul></div></div><div id='page'>";  
        pw.print(result);  
    } else  
        pw.print(result);  
}
```

```
...
```

```
}
```

# Registration Servlet : How Customer Accounts are Stored in Hash Map ?

```
public class Registration extends HttpServlet {
    String username = request.getParameter("username");
    String password = request.getParameter("password");

    ...

    HashMap<String, User> hm=new HashMap<String, User>();
    try
    {
        FileInputStream fileInputStream = new FileInputStream(new File(TOMCAT_HOME+"\\webapps\\GameSpeedServletCSSwithSAXTutorial\\UserDetails.txt"));
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

        hm= (HashMap)objectInputStream.readObject();
    }
    if(hm.containsKey(username))
    { error_msg = "Username already exist as " + usertype;}
    else{
        User user = new User(username,password,usertype);
        hm.put(username, user);

        FileOutputStream fileOutputStream = new FileOutputStream(TOMCAT_HOME+"\\webapps\\GameSpeedServletCSSwithSAXTutorial\\UserDetails.txt");
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
        objectOutputStream.writeObject(hm);
        objectOutputStream.flush();
        objectOutputStream.close();
        fileOutputStream.close();

    }
    displayRegistration(request, response, pw, true);
    ...
}
```

## Utilities Servlet : How customer orders are stored in hash map ?

```
public void storePayment(int orderId,String orderName,double orderPrice,String userAddress,String creditCardNo){

HashMap<Integer, ArrayList<OrderPayment>> orderPayments= new HashMap<Integer, ArrayList<OrderPayment>>();
    try
    {
        FileInputStream fileInputStream = new FileInputStream(new File(TOMCAT_HOME+"\\webapps\\GameSpeedServletCSSwithSAXTutorial\\PaymentDetails.txt"));
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

        orderPayments = (HashMap)objectInputStream.readObject();
    }
    if(!orderPayments.containsKey(orderId)){
        ArrayList<OrderPayment> arr = new ArrayList<OrderPayment>();
        orderPayments.put(orderId, arr);
    }
    ArrayList<OrderPayment> listOrderPayment = orderPayments.get(orderId);
    OrderPayment orderpayment = new OrderPayment(orderId,username(),orderName,orderPrice,userAddress,creditCardNo);
    listOrderPayment.add(orderpayment);
    try
    {
        FileOutputStream fileOutputStream = new FileOutputStream(new File(TOMCAT_HOME+"\\webapps\\GameSpeedServletCSSwithSAXTutorial\\PaymentDetails.txt"));
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);
        objectOutputStream.writeObject(orderPayments);
        objectOutputStream.flush();
        objectOutputStream.close();
        fileOutputStream.close();
    }

    ...
}
```

# SaxParserDataStore Servlet:

```
public class SaxParserDataStore extends DefaultHandler {

    ...

    Console console;

    Game game;

    Tablet tablet;

    Accessory accessory;

    static HashMap<String,Console> consoles;

    static HashMap<String,Game> games;

    static HashMap<String,Tablet> tablets;

    static HashMap<String,Accessory> accessories;

    public SaxParserDataStore(String consoleXmlFileName) {

        this.consoleXmlFileName = consoleXmlFileName;

        consoles = new HashMap<String, Console>();

        games=new HashMap<String, Game>();

        tablets=new HashMap<String, Tablet>();

        accessories=new HashMap<String, Accessory>();

        accessoryHashMap=new HashMap<String,String>();

        parseDocument();

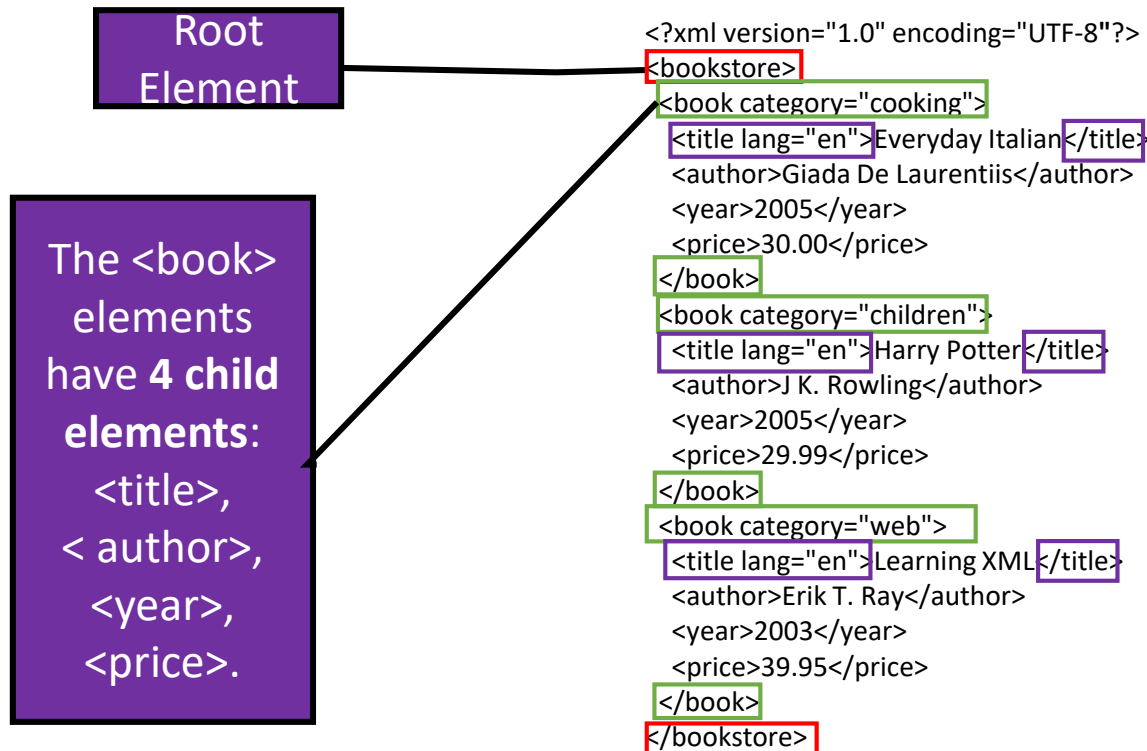
    } ... }
```

# What is XML?

- XML stands for EXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be self-descriptive.
- XML is used in many aspects of web development.
- XML is often used to separate data from presentation.
- To Learn More on How to Use XML , Visit : [http://www.w3schools.com/xml/xml\\_usedfor.asp](http://www.w3schools.com/xml/xml_usedfor.asp)

## XML Document :

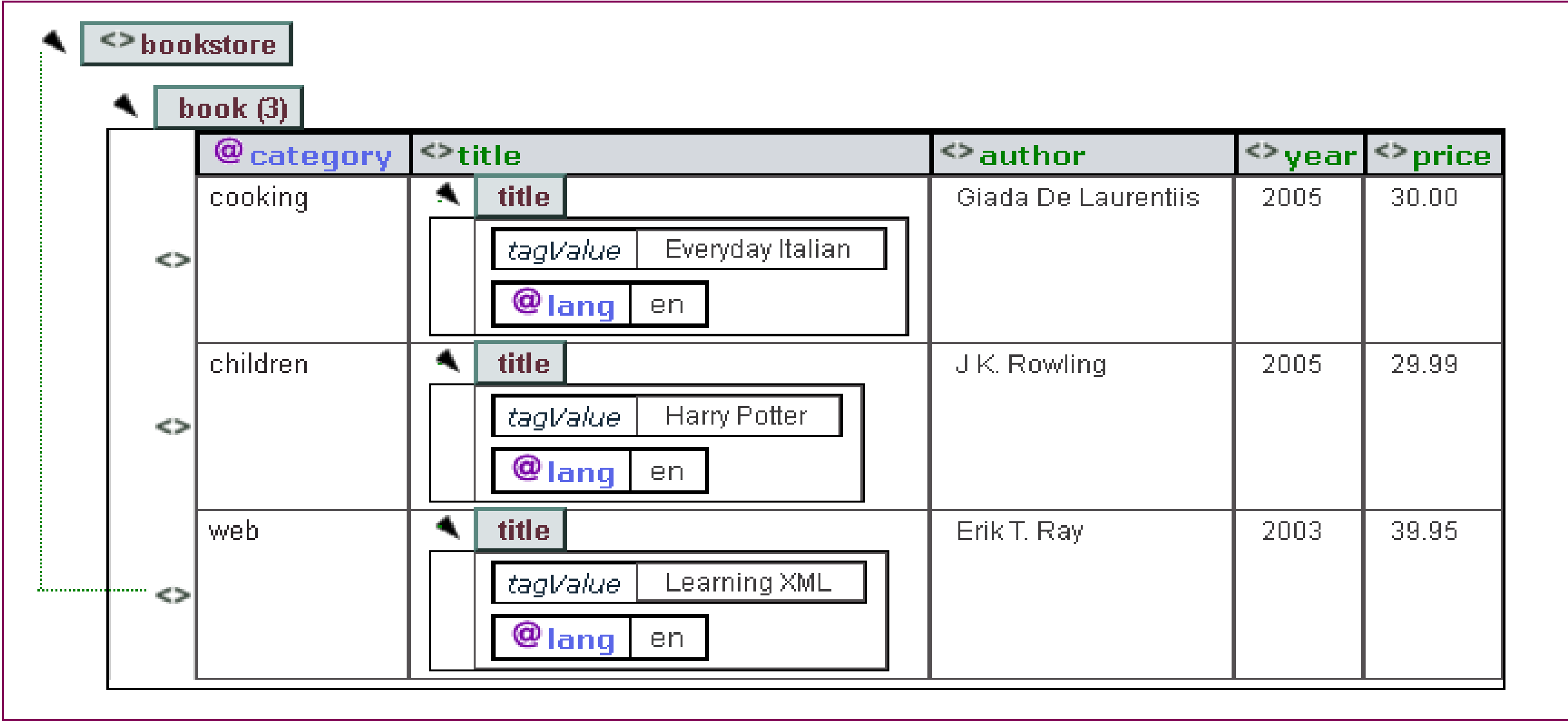
- XML documents form a tree structure that starts at "the root" and branches to "the leaves".



# XML Tree Structure:

An XML tree starts at a **root element** and branches from the root to **child elements**.

Let us Seen an Example of XML Tree Structure of XML Shown Before :



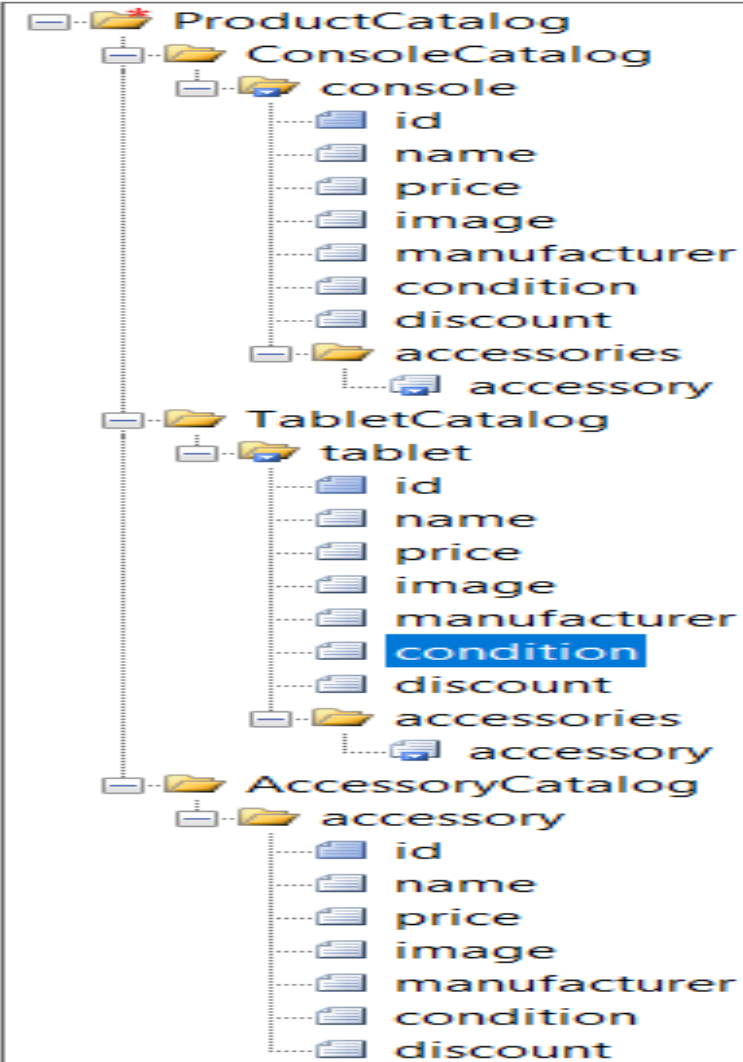
To Learn More about XML Tree , Visit: [http://www.w3schools.com/xml/xml\\_tree.asp](http://www.w3schools.com/xml/xml_tree.asp)

# XML Structure for ProductCatalog.xml :

## XML Source

XML maps in this workbook:

ProductCatalog\_Map



## ProductCatalog.xml :

<ProductCatalog>

<ConsoleCatalog>

```
<console id="xboxone">
  <name>Xbox One</name>
  <price>399.99</price>
  <image>xbox1.jpg</image>
  <manufacturer>Microsoft</manufacturer>
  <condition>New</condition>
  <discount>10.0</discount>
  <accessories>
    <accessory>xboxone_wc</accessory>
    <accessory>xboxone_sh</accessory>
  </accessories>
</console>
<console id="ps4">
  <name>PS4</name>
  <price>349.99</price>
  <image>PS4-console-bundle.jpg</image>
  <manufacturer>Sony</manufacturer>
  <condition>New</condition>
  <discount>10.0</discount>
  <accessories>
    <accessory>ps4_wc</accessory>
  </accessories>
</console>
```

</ConsoleCatalog>



## XML Document Tree Structure for ProductCatalog.xml :

<> ProductCatalog

<> ConsoleCatalog

console (2)

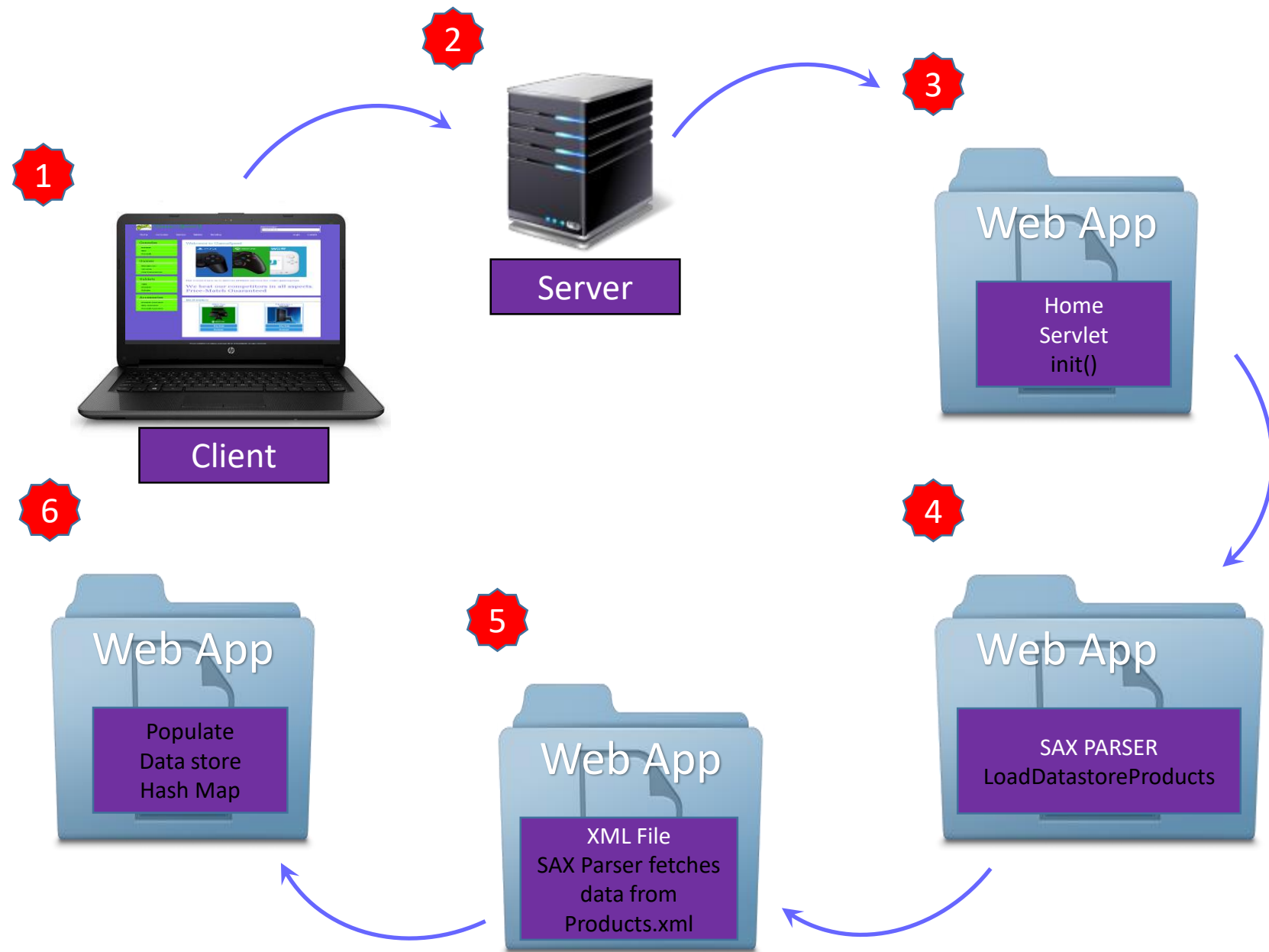
@id	<>name	<>price	<>image	<>manufacturer	<>condition	<>discount	<>accessories
xboxone	Xbox One	399.99	xbox1.jpg	Microsoft	New	10.0	<p>accessories</p> <p>accessory (2)</p> <ul style="list-style-type: none"><li>xboxone_wc</li><li>xboxone_sh</li></ul>
ps4	PS4	349.99	PS4-console-bundle.jpg	Sony	New	10.0	<p>accessories</p> <p>accessory ps4_wc</p>

<> TabletCatalog

tablet (2)

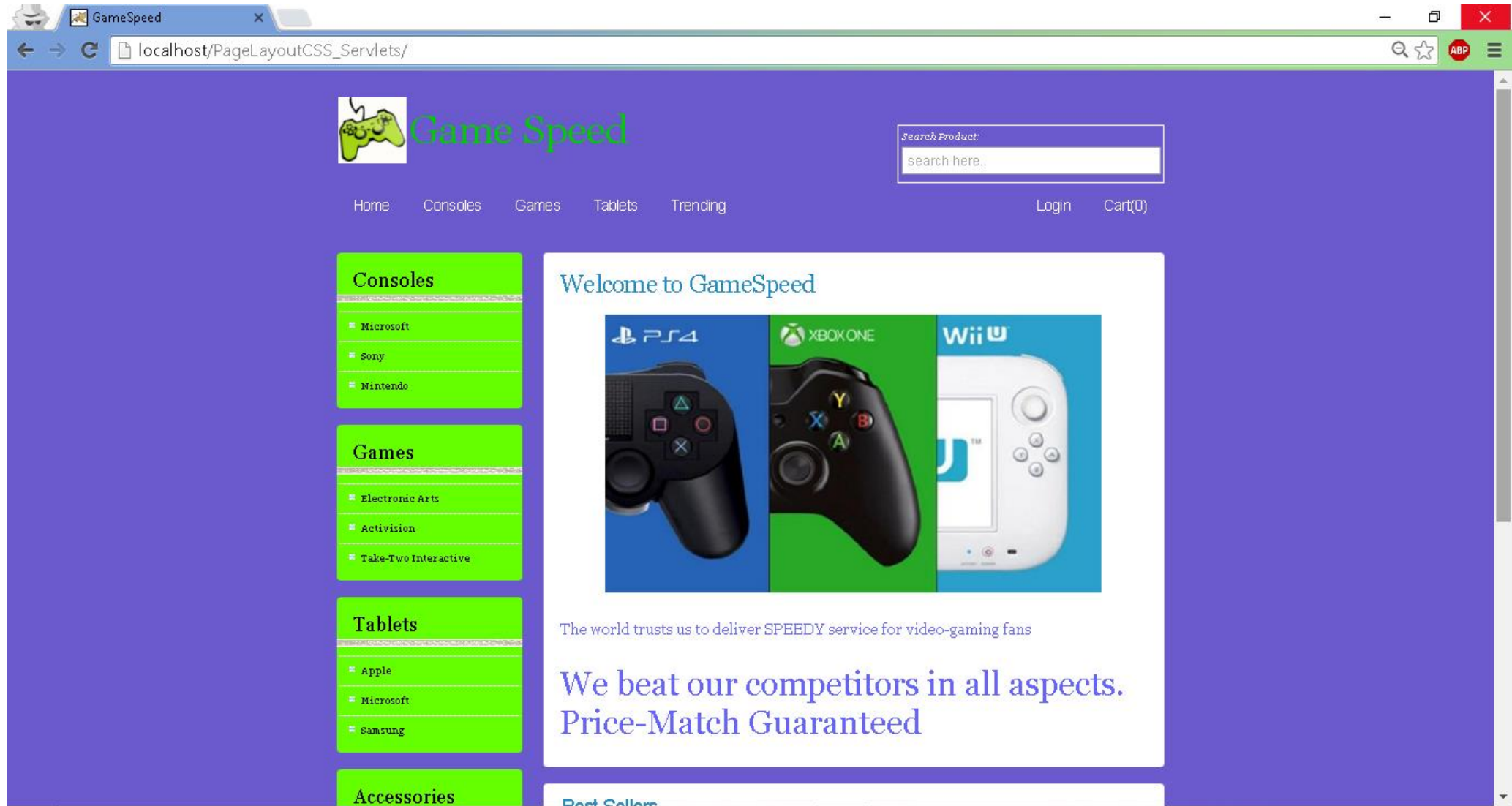
@id	<>name	<>price	<>image	<>manufacturer	<>condition	<>discount	<>accessories
ipad	ipad	189.99	ipad.jpg	Apple	Used	40.0	<p>accessories</p> <p>accessory ipad-ep</p>
Surface	Surface	299.99	surface.jpg	Microsoft	New	15.0	<p>accessories</p> <p>accessory (2)</p> <ul style="list-style-type: none"><li>surf_spkr</li><li>surf_hp</li></ul>

How Product Data Gets Populated:



## Example – Startup Page:

- Open your browser and type in 'http://localhost/PageLayoutCSS\_Servlets/'



# Example – Page Layout:

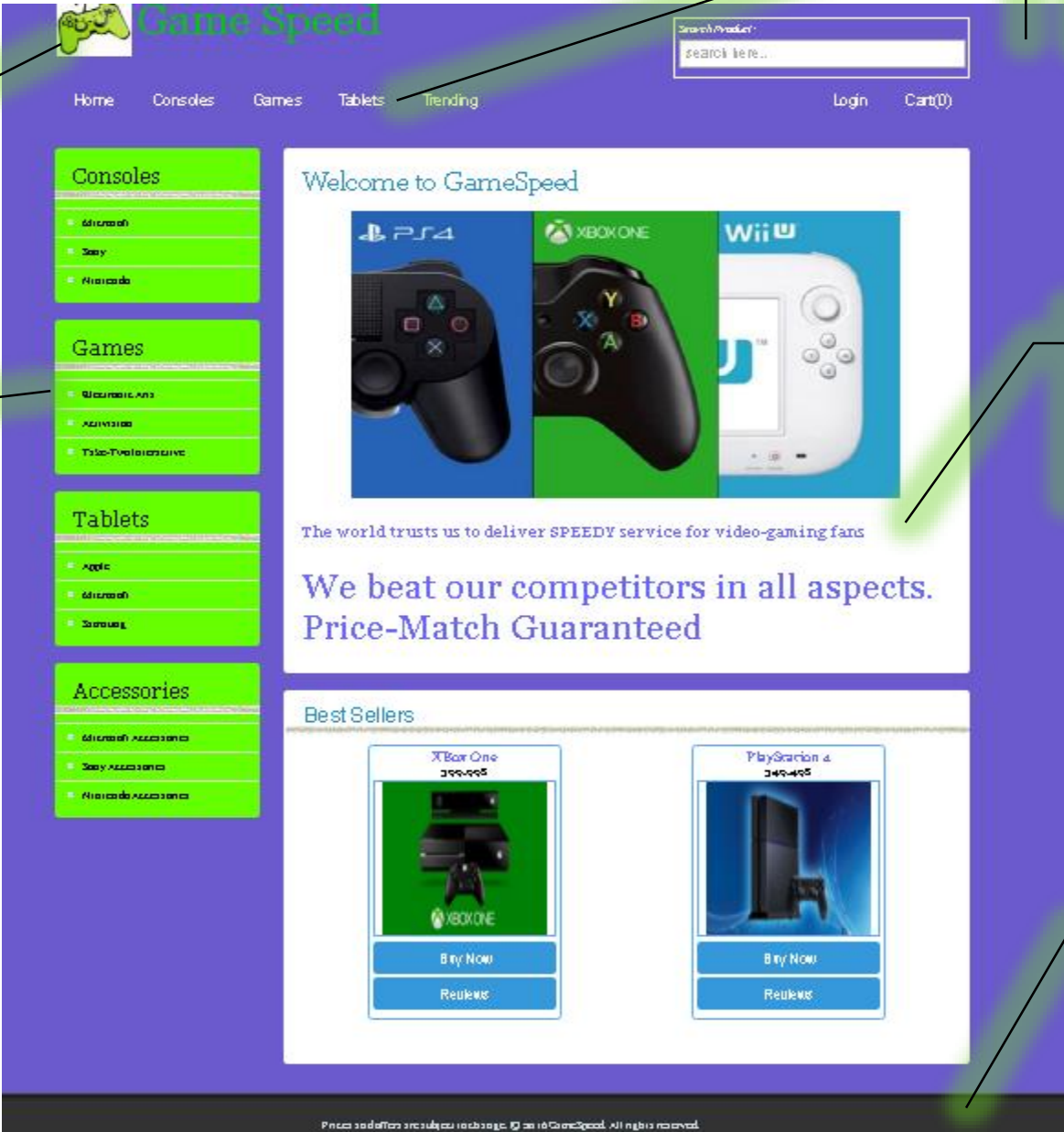
Top/Main Navigation Bar

Page Title

Left Navigation Bar

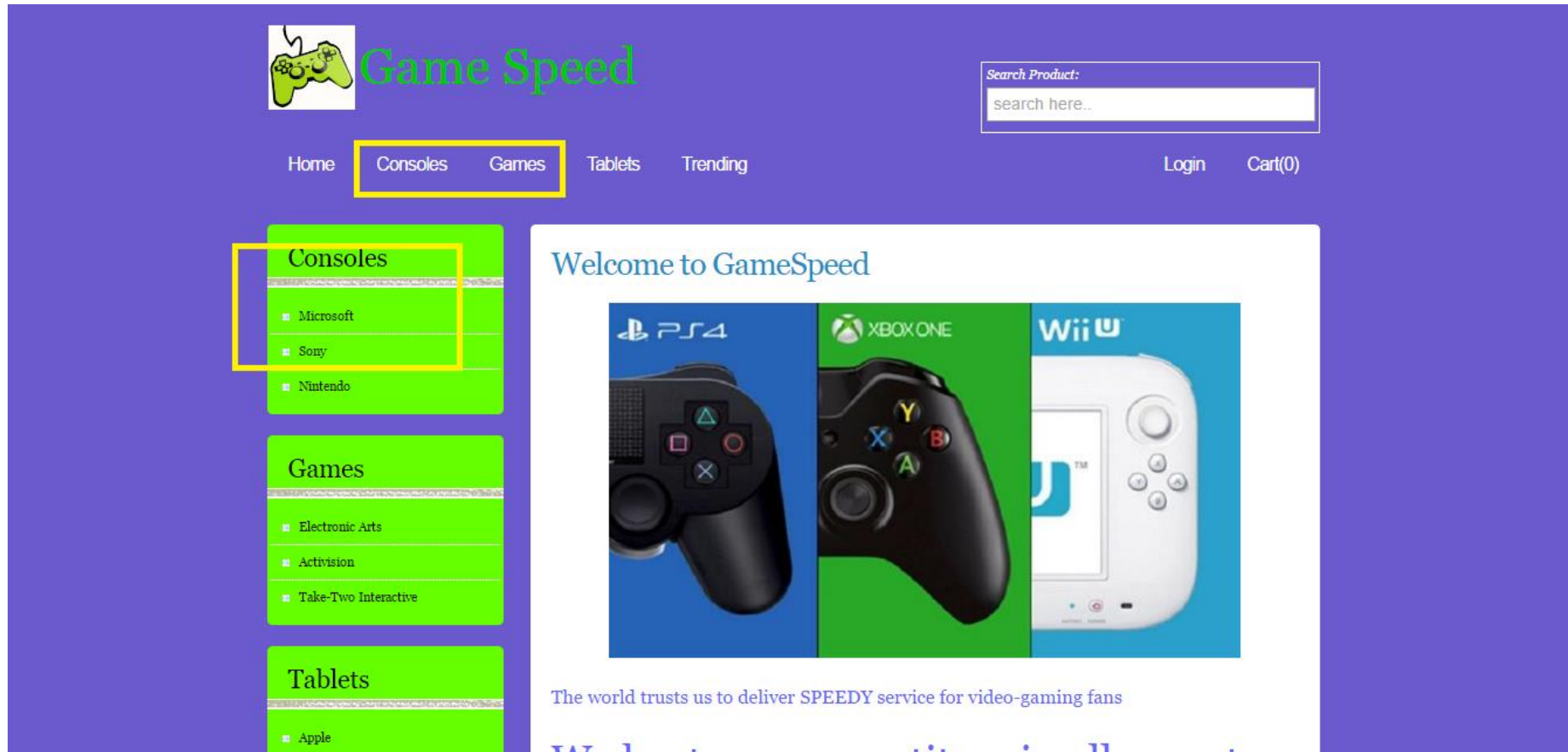
Main Content

Footer Section



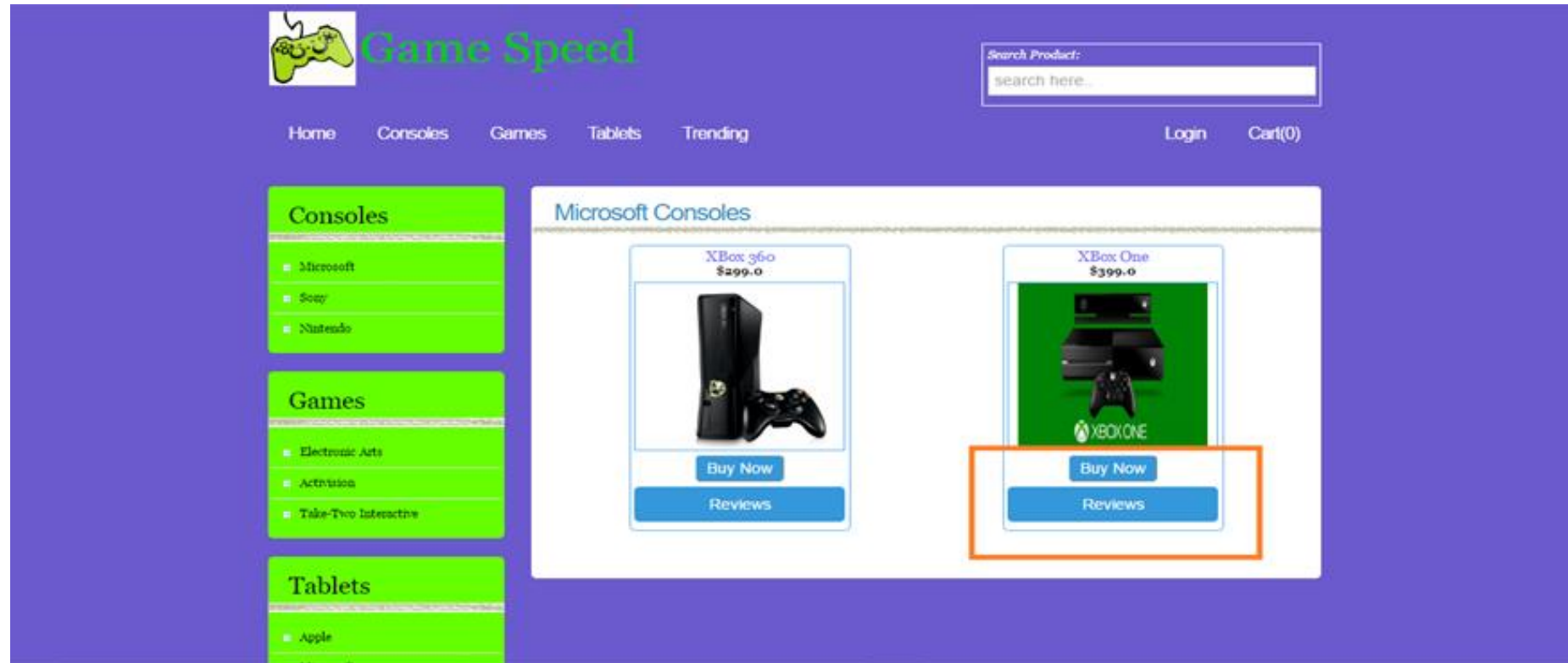
## Example – Navigation Bar:

- Click on the products available in the navigation bar
- You can also select the products from the left navigation bar



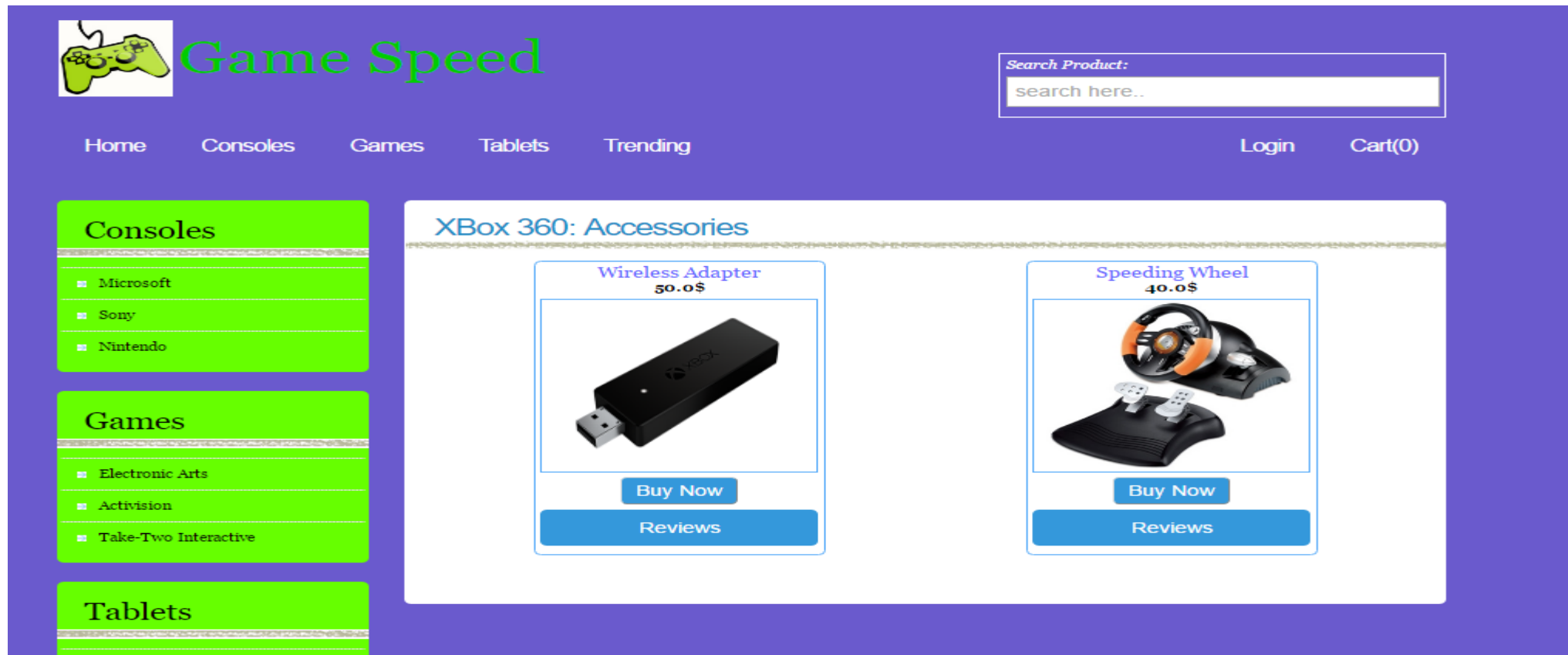
## Example – Products Catalog:

- Clicking on a product type will take you to the product page
- You have different options available such as buy a product , write reviews for the product.



## Example – Accessories:

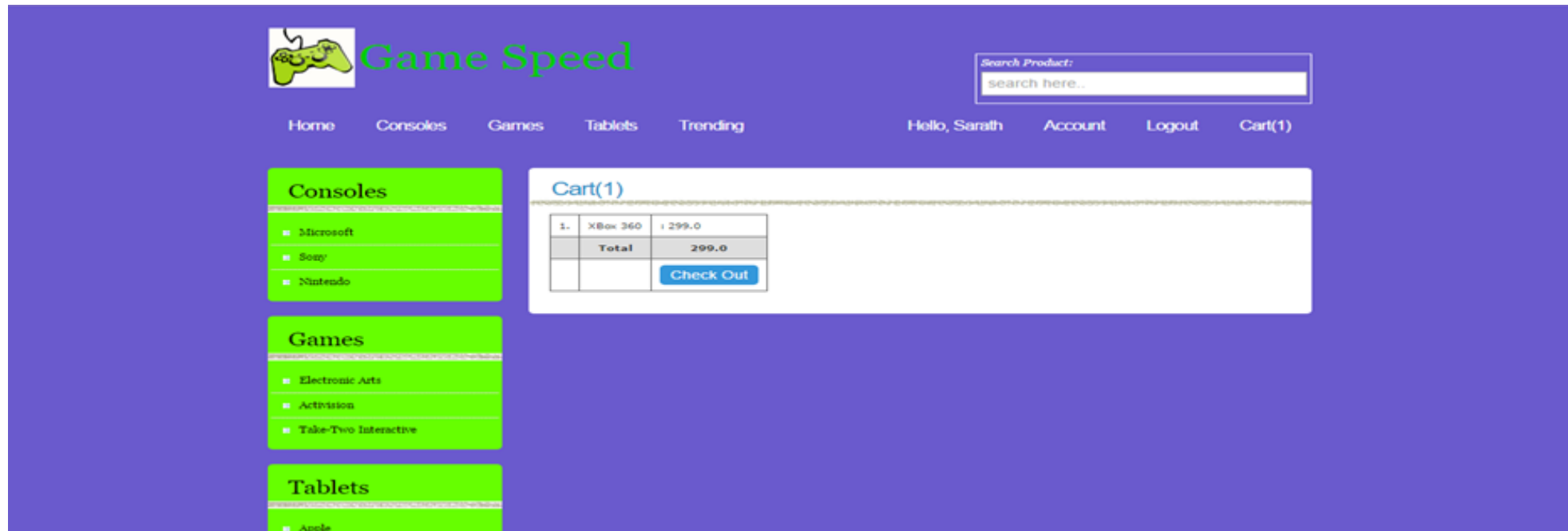
- Clicking on a Accessories link will take you to the Accessories page.
- You have different options available such as buy Accessories and review them.





## Example – Place Order:


- Click on 'Buy Now' button on the products page to purchase the product
- This should take you to a new page (Cart Servlet) where you can purchase the product
- Click on 'Check Out' to place the order for the selected product.





## Example – Purchase Order:

- Enter the Credit card Number or Account Number of the Customer.
- Enter the Customer address.
- Click on ‘Submit’ to place the order for the selected product.



# Game Speed

[Home](#) [Consoles](#) [Games](#) [Tablets](#) [Trending](#) [ViewOrder](#) [Hello,Sarath](#) [Account](#) [Logout](#) [Cart\(1\)](#)

### Consoles

- Microsoft
- Sony
- Nintendo

### Games

- Electronic Arts
- Activision
- Take-Two Interactive

### Order

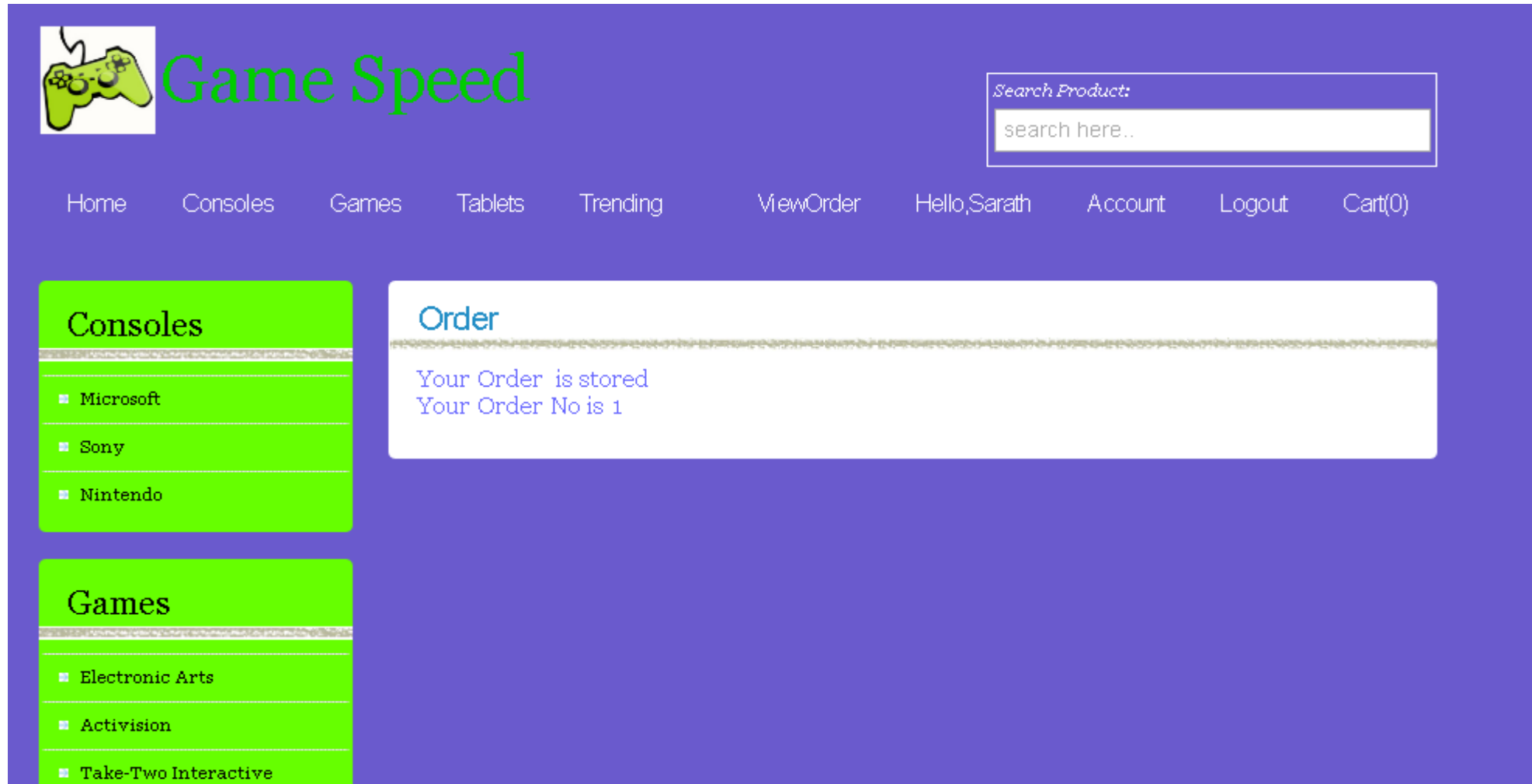
Customer Name:	Sarath
Product Purchased:	xbox360
Product Price:	399.99
Total Order Cost	399.99

Credit/accountNo	2456345678901245
Customer Address	Chicago , Illinois

Submit

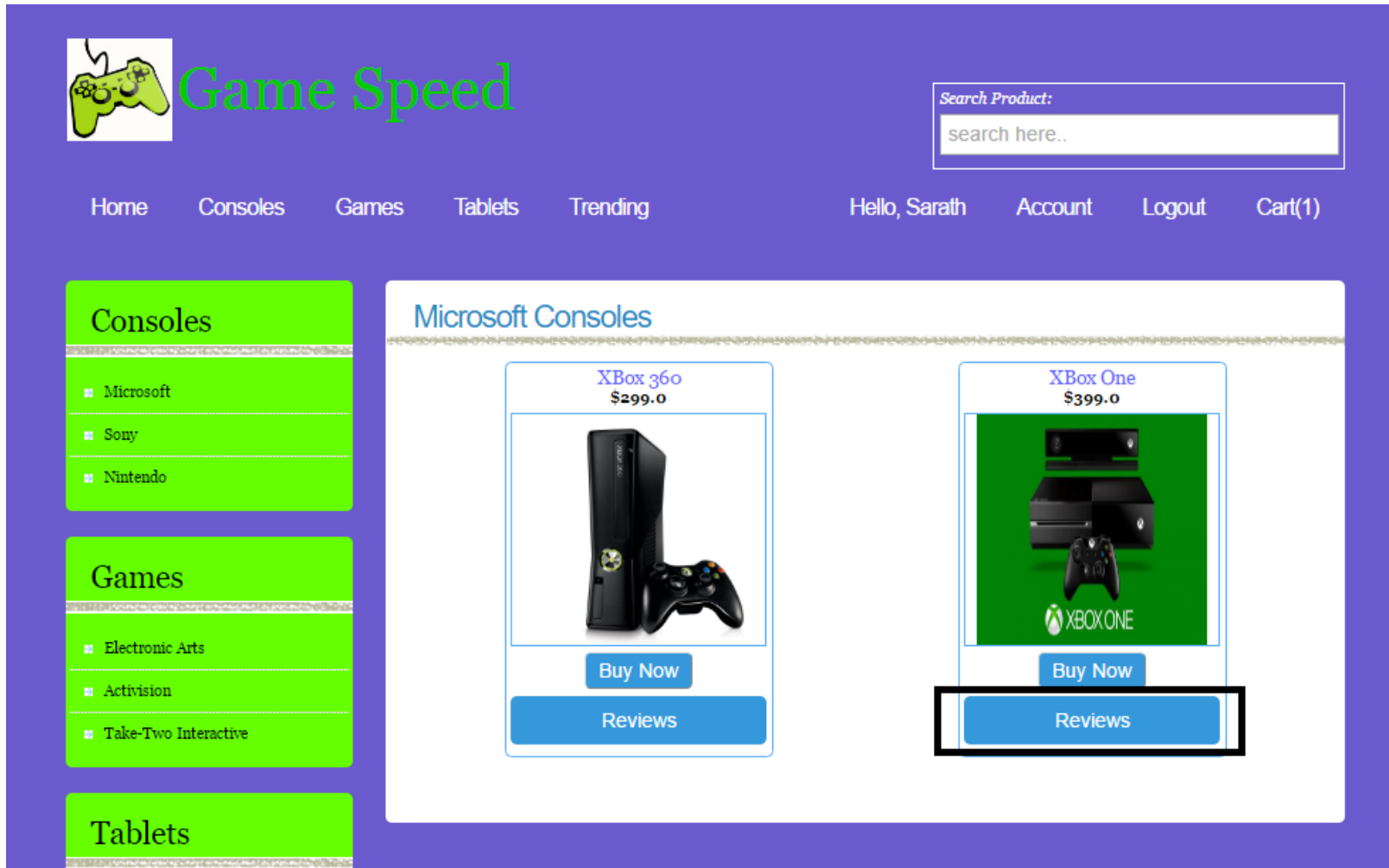
## Example – Order Confirmation:

- Order Number is Generated for the Customer Order.
- Customer Order is Stored.



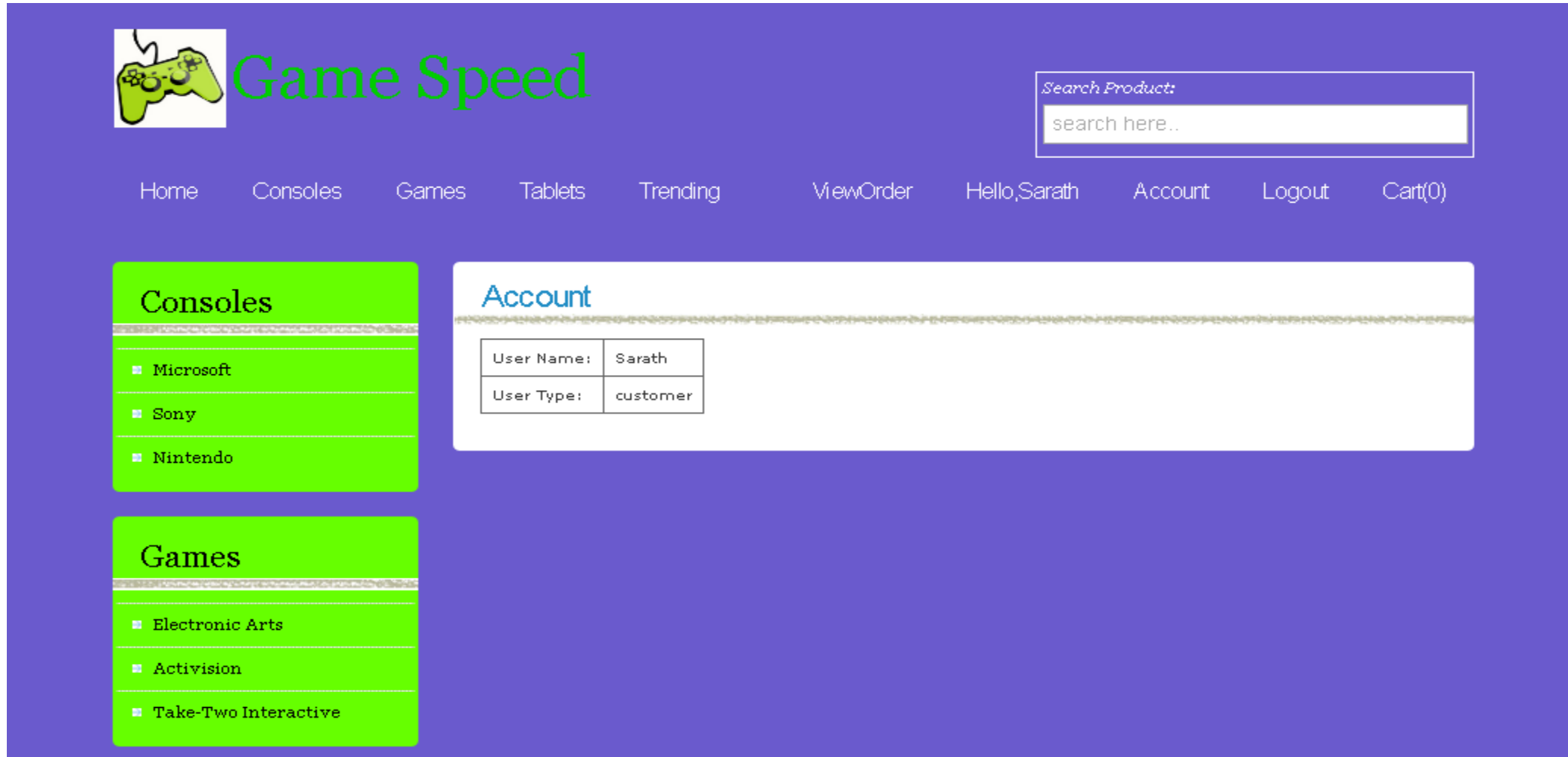
## Example – Write Review:

- To write a review for the product, click on 'Reviews' button



## Example – Account Tab:

- Account Details of the Customer is Shown.



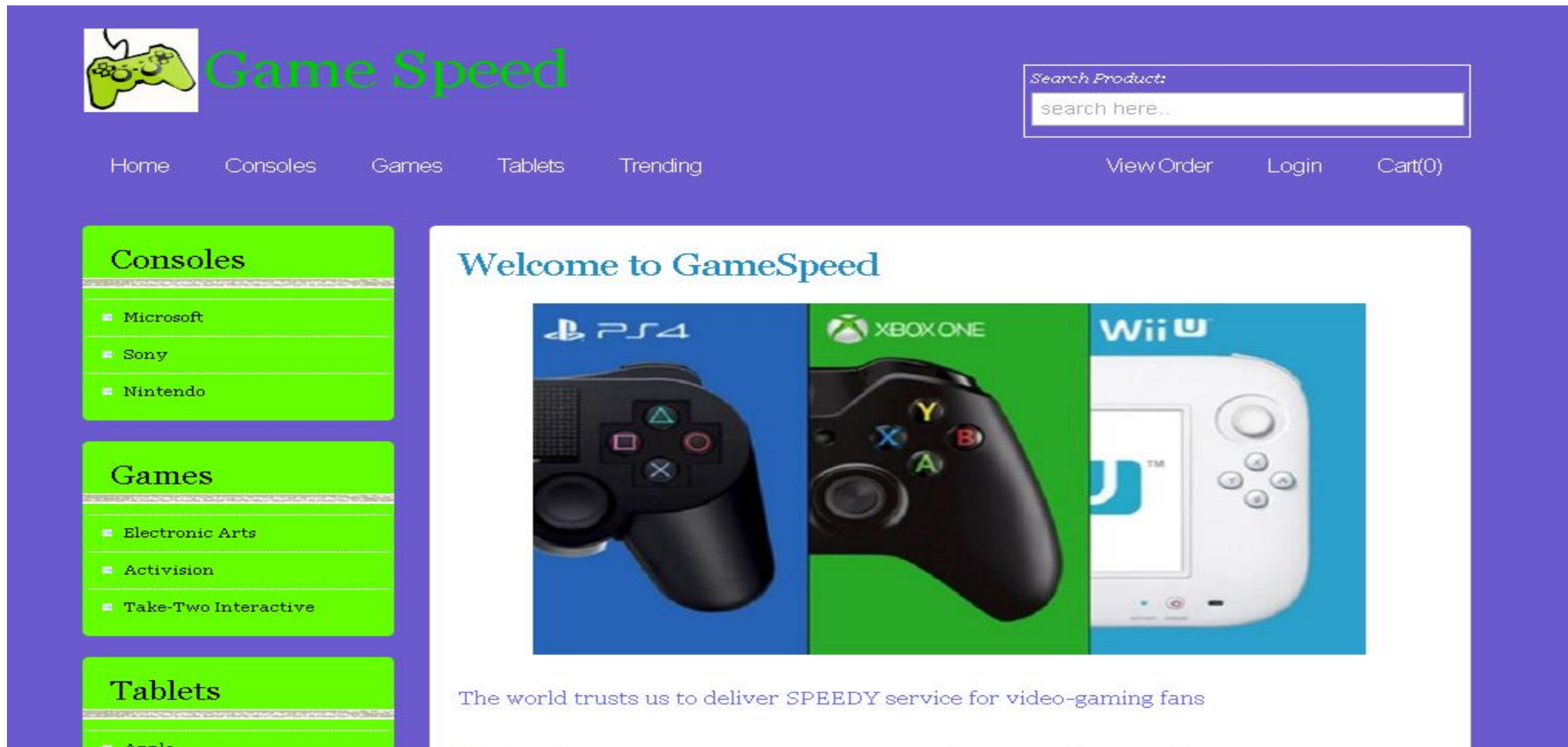
The screenshot displays the 'Game Speed' website interface. The header features a logo of a yellow game controller with a cursor icon, the text 'Game Speed' in green, and a search bar labeled 'Search Product:' with the placeholder 'search here..'. The navigation menu includes links for Home, Consoles, Games, Tablets, Trending, ViewOrder, Hello,Sarath, Account, Logout, and Cart(0). The 'Account' tab is selected, showing a white box with the title 'Account' and a table of user details.

User Name:	Sarath
User Type:	customer

On the left side, there are two green boxes. The 'Consoles' box lists Microsoft, Sony, and Nintendo. The 'Games' box lists Electronic Arts, Activision, and Take-Two Interactive.

## Example – Logout:

- User Successfully logged out of the Web Application.



Questions?