

**ADVANCED JAVA & WEB TECHNOLOGIES LAB**

**Week-1:**

Design the following static web pages required for an online book store web site.

**1) HOMEPAGE:**

The static home page must contain three frames.

**Top frame:**

Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

**Left frame:**

At least four links for navigation, which will display the catalogue of respective links.

**For e.g.:** When you click the link “CSE” the catalogue for CSE Books should be displayed in the Right frame.

**Right frame:**

The pages to the links in the left frame must be loaded here. Initially this page contains description of the web site.

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Description of the Web Site			

**2) LOGINPAGE:** This page looks like below




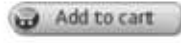
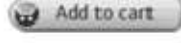
Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	<p>Login : <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Submit"/> <input type="button" value="Reset"/></p>			

**3) CATOLOGUEPAGE:**

The catalogue page should contain the details of all the books available in the web site in a table.

The details should contain the following:

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE		Book : XML Bible Author : Winston Publication : Wiley	\$ 40.5	
ECE				
EEE				
CIVIL		Book : AI Author : S.Russel Publication : Princeton hall	\$ 63	
		Book : Java 2 Author : Watson Publication : BPB publications	\$ 35.5	
		Book : HTML in 24 hours Author : Sam Peter Publication : Sam publication	\$ 50	

Note: Week 2 contains the remaining pages and their description.

### Week-2:

**4) CART PAGE:** The cart page contains the details about the books which are added to the cart. The cart page should look like this:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE	<b>Book name</b>	<b>Price</b>	<b>Quantity</b>	<b>Amount</b>
ECE	Java 2	\$35.5	2	\$70
EEE	XML bible	\$40.5	1	\$40.5
CIVIL				
		<b>Total amount -</b>		<b>\$130.5</b>

### 5)REGISTRATIONPAGE:

Create a “*registration form* “with the following fields

- 1) Name (Text field)      2) Password (password field)
- 3) E-mail id (text field)    4) Phone number (text field)
- 5) Sex (radio button)      6)Date of birth (3 select boxes)
- 7) Languages known (check boxes – English, Telugu, Hindi, Tamil)
- 8) Address (text area)

### WEEK3:

#### VALIDATION:

Write *JavaScript* to validate the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

Note : You can also validate the login page with these parameters.

Use PHP to connect with the database to store the above details.

#### Week-4:

Design a web page using CSS (Cascading Style Sheets) which includes the following:

- 1) Use different font, styles: In the style definition you define how each selector should work (font, color etc.). Then, in the body of your pages, you refer to these selectors to activate the styles.

For example:

```
<HTML>
<HEAD>
<style type="text/css">
B.headline {color:red; font-size:22px; font-family:arial; text-decoration:underline}
</style>

</HEAD>
```

```
<BODY>
<b>This is normal bold</b><br>
Selector {cursor:value}

For example:

<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>

<b class="headline">This is headline style bold</b>
</BODY>

</HTML>
```

- 2) Set a background image for both the page and single elements on the page.

```
BODY {background-image:url(myimage.gif);}
```

You can define the background image for the page like this:

- 3) Control the repetition of the image with the background-repeat property. As background-repeat: repeat Tiles the image until the entire page is filled, just like an ordinary background image in plain HTML.

- 4) Define styles for links as

A:link

A:visited

A:active

A:hover

Example:

```
<style type="text/css">
```

```
A:link {text-decoration: none}
```

```
A:visited {text-decoration: none}
```

```
A:active {text-decoration: none}
```

```
A:hover {text-decoration: underline; color: red;}
```

```
</style>
```

5) Work with layers:

For example:

LAYER 1 ON TOP:

```
<div style="position: relative; font-size:50px; z-index:2;">LAYER 1</div>
```

```
<div style="position: relative; top:-50; left:5; color: red; font-size:80px; zindex:1">LAYER 2</div>
```

LAYER 2 ON TOP:

```
<div style="position: relative; font-size:50px; z-index:3;">LAYER 1</div>
```

```
<div style="position: relative; top:-50; left:5; color:red; font-size:80px; zindex:4">LAYER 2</div>
```

6) Add a customized cursor:

```
Selector {cursor:value}
```

**For example:**

```
<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
```

```
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
```

```
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>
```

**Week-5:**

Write an XML file which will display the Book information which includes the following:

- 1) Title of the book
- 2) Author Name
- 3) ISBN number
- 4) Publisher name
- 5) Edition
- 6) Price

Write a Document Type Definition (DTD) to validate the above XML file.

Display the XML file as follows.

The contents should be displayed in a table. The header of the table

should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose.

Note: Give at least for 4 books. It should be valid syntactically.

Hint: You can use some xml editors like XML-spy

### **Week-6:**

#### **VISUALBEANS:**

Create a simple visual bean with a area filled with a color. The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false. The color of the area should be changed dynamically for every mouse click. The color should also be changed if we change the color in the “property window“.

### **Week-7:**

1) Install TOMCAT web server and APACHE.

While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.

2) Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root. Access the pages by using the urls : <http://localhost:4040/rama/books.html> (for tomcat) <http://localhost:8080/books.html> (for Apache)

### **Week-8:**

#### **User Authentication:**

Assume four users user1, user2, user3 and user4 having the passwords pwd1,pwd2,pwd3 and pwd4 respectively. Write a servlet for doing the following.

1. Create a Cookie and add these four user id's and passwords to this Cookie.

2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords ) available in the cookies. If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display “ You are not an authenticated user “. Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the `getInitParameters ()` method.

### **Week-9:**

Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form). Practice ‘JDBC’ connectivity.

Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

### **Week-10:**

Write a JSP which does the following job:

Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (similar to week8 instead of cookies).

**Week-11:**

Create tables in the database which contain the details of items (books in our case like Book name, Price, Quantity, Amount)) of each category. Modify your catalogue page (week 2) in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using JDBC.

**Week-12:**

**HTTP** is a stateless protocol. Session is required to maintain the state. The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time (i.e., from different systems in the LAN using the ip-address instead of local host). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method `session.invalidate()`). Modify your catalogue and cart JSP pages to achieve the above mentioned functionality using sessions.