

WEB PROGRAMMING LAB

(Common to CSE & IT)

Course Code :13CT1121

| | | | |
|----------|----------|----------|----------|
| L | T | P | C |
| 0 | 0 | 3 | 2 |

Course Educational Objectives:

The main objective of the lab course is to expose the students to different programming aspects related to web designing using different technologies. Upon completion of this course, the student should be able to:

1. Understand web page site planning, designing, and maintenance.
2. Develop web sites which are secure and dynamic in nature and writing scripts which get executed on server as well.
3. Study the actual advanced Web methodologies, specifications and techniques.
4. Acquire the skills necessary to design, implement and deploy complex Web sites and applications.
5. Understands the concepts of PHP and AJAX to develop web pages

Course Outcomes:

At the end of the course the student should be able to

1. Get practical exposure on HTML, XHTML, CSS, JavaScript, XML and other web technologies.
2. Get practical exposure to develop XML Technologies such as XML Schemas, XSLT.
3. Get practical exposure to develop Server-Side Programming using Servlets and JSP's.
4. Develop web pages using AJAX and PHP.
5. Develop a website using the above technologies.

LIST OF PROGRAMS:

WEEK-1 & 2:

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

web site.

1) HOME PAGE:

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 website | | | |

2) Login Page





| Logo | Web Site Name | | | |
|----------------------------|---|--------------|-----------|------|
| Home | Login | Registration | Catalogue | Cart |
| CSE ECE EEE CIVIL | <p style="text-align: center;">Log in <input type="text"/></p> <p style="text-align: center;">Password <input type="text"/></p> <p style="text-align: center;">Submit <input type="button"/> Reset <input type="button"/></p> | | | |

3) CATALOGUE PAGE:

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.

| Login | Web Site Name | | | |
|-------|---|---|-----------|--|
| Home | Login | Registration | Catalogue | Cart |
| CSE |  | Book: XML Bible Author: Winston Publication: Wiley | \$ 40.5 | <input type="button" value="ADD TO CART"/> |
| ECE |  | Book: AI Author: S. Russel Publication: Princeton hall | \$ 63 | <input type="button" value="ADD TO CART"/> |
| EEE |  | Book: Java 2 Author: Watson Publication: BPB publications | \$ 35.5 | <input type="button" value="ADD TO CART"/> |
| CIVIL |  | Book: HTML in 24 hours Author: Sam Peter Publication: Sam publication | \$ 50 | <input type="button" value="ADD TO CART"/> |

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

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 | | | |
|-------|---------------|--------------|-----------|-----------------------|
| | Login | Registration | Catalogue | Cart |
| CSE | | | | |
| ECE | Book name | Price | Quantity | Amount |
| EEE | Java 2 | \$ 35.5 | 2 | \$ 70 |
| CIVIL | XML bible | \$ 40.5 | 1 | \$ 40.5 |
| | | | | Total amount- \$130.5 |

5) REGISTRATION PAGE:

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)

WEEK 3:

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.

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

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

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; z-index:1">LAYER 2</div>
```

LAYER 2 ON TOP:

```
<div style="position:relative; font-size:50px; z-index:3;">LAYER1</div>  
<div style="position:relative; top:-50; left:5; color:red; fontsize: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:

- 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 7:

USERAUTHENTICATION :

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 8:

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. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week1).

WEEK 9:

Write a JSP which does the following job: Insert the details of the 3 or 4 users who register with the web site (week7) 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 10:

1. Create a simple XMLHttpRequest, and retrieve data from a TXT file using AJAX
2. Create an XMLHttpRequest to retrieve data from an XML file using AJAX

WEEK 11:

1. How a web page can communicate with a web server while a user type characters in an input field. (Retrieve the content of an JSP file)
2. Retrieve content from a database using AJAX.

WEEK 12:

Design the web pages required for an online book store web site using PHP, AJAX, and HTML5.

REFERENCES:

1. Dietel and Dietel, “*Internet and World Wide Web - How to Program*”, 5th Edition, PHI/Pearson Education, 2011
2. Herbert Schildt, “*The complete Reference Java 2*”, 8th Edition, TMH, 2011.
3. Phil Hanna, “*The Complete Reference JSP*”, 1st Edition, TMH, 2003.

