Sreenidhi Institute of Science and Technology
Department of Computer Science and Engineering
B.Tech Computer Science and Engineering

Name of the lab: Web Technologies Lab

Lab Number: 2214A

Number of systems: 72

HARDWARE DETAILS

S.NO	NAME OF THE HARDWARE	MANUFACTURER
1	Make	ACER
2	Model	ACER Vertion M200 H81
3	Memory	4GB RAM
4	Processor	Inter Core i3,3.70GHZ
5	Hard Disk(HDD)	1 TB HDD
6	SMPS	ATX MERCURY
7	Monitor	ACER 18.5
8	Keyboard	ACER QUERTY MODEL
9	Mouse	ACER QTICAL

SOFTWARE DETAILS

S.NO	NAME OF THE SOFTWARE	SOURCE
1	ORACLE Database 10g	Licensed
2	Windows Operating System	Licensed

List of Experiments in Web Technologies Lab

Week-1:

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 Web Site

Fig 1.1

2) LOGIN PAGE:

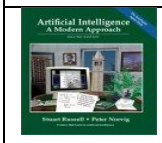
This page looks like below:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Login : Password: Submit Reset			

3) CATOLOGUE 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:
Snap shot of Cover Page.

Author Name.
Publisher.
Price.
Add to cart button.

Logo	**Web Site Name**			
Home	**Login**	**Registration**	**Catalogue**	**Cart**
CSE		Book : XML Bible Author : Winston Publication : Wiely	$ 40.5	Add to cart
ECE		Book : AI Author : S.Russel Publication : Princeton hall	$ 63	Add to cart

EEE CIVIL		Book : Java 2 Author : Watson Publication : BPB publications	$ 35.5	Add to cart
		Book : HTML in 24 hours Author : Sam Peter Sam publication	$ 50	Add to cart

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 ECE EEE CIVIL	Book name Java 2 XML bible $40.5 Total amount -	Price $35.5 $130.5	Quantity 2 $40.5	Amount $70 1

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) Gender (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 fields of the above registration page.

<html>
<head>
<style>
Create the internal stylesheet using the following properties:

1. Apply font-family for all input and tr selectors- as Monotype Corsiva with font-size:30, color:brown, text-align: center
2. Use the following properties for option and select selectors with font-size:24 and color:blue

</style>
<script type="text/javascript">
function validate()
{
Write an internal javascript to validate fields of the registration form with the conditions:

A. Name field - must not be null, - must have only alphabets, - size must not be greater than 45 characters, - must not have special characters apart from spaces.
B. Password field - must not be null, - size must not be less than 6 characters, - must be combination of digits, special characters and alphabets.
C. Email field - must not be null, - first character should be an alphabet,-must follow standard format: name@domain.com
D. Phone Number field – must not be null, - must have exactly 10 digits, - must not have either the special characters or alphabets.
E. Gender field – values must be Male, Female and Transgender, user must select only one.
F. Date of Birth (DOB) field – must be 3 select boxes (DD, MONTH, YYYY). Else create the calendar for DOB selection.
G. Languages known – values must be English, Telugu, Hindi, Tamil, etc, - user must be able to select known languages.
H. Address field- should be a box of 5 rows and 50 columns, -must not be empty.

}
</script>
</head>
<body bgcolor=yellow>
<form name=" " method=" " action=" ">
Do the following:

1. Design the above mentioned fields: (Name, Password, Email, Phone Number, Gender, DOB, Languages Known and Address).
2. All should be under one form with the appropriate attributes of form.
3. Above field must be neatly aligned using the table tag.
4. Place the buttons:

- Submit- On clicking submit, validation function must execute.
- Reset- All field must be cleared of the contents.

</form>
</body>
</html>

Note: Students can make use of the different events to validate the fields.

Week-4:

Design a web page using CSS (Cascading Style Sheets) which includes the following and also create the External CSS for the same and apply this to the earlier created web-pages.

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.

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

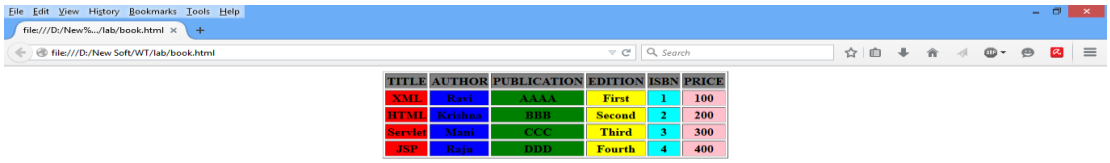
Week-5:

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) or XML Schema Definition (XSD) to validate the above XML file.
- Display the XML file.
- 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



TITLE	AUTHOR	PUBLICATION	EDITION	ISBN	PRICE
XML	Ravi	AAAA	First	1	100
HTML	Krishna	BBB	Second	2	200
Servlet	Mani	CCC	Third	3	300
JSP	Raju	DDD	Fourth	4	400

Week 6:

Install a database (Mysql or Oracle). And perform the following:

1. Write a Java program to create a table using the fields of the Registration page created earlier in week 2 (program 5).
2. Write a java program to connect to that database and experiment with various (select, insert, create, update, delete, etc) SQL queries. Also make use of ***Statement*** and ***PreparedStatement***.
3. Write a java program to call a procedure using the ***CallableStatement.***
4. Write jdbc program to get the metadata of the database table. Use M***etaData.***

   For above programs, jdbc connectivity can be done as given below:

   import the required java and jdbc packages.

   public static void main(String args[]) throws Exception

   {

   try{

   1. Pass the driver name using Class.forName("<driver-name>").
   2. Create the Connection using Connection con=DriverManager.getConnection("<url of the driver>","<username of the database>","<password of the database>") method.
   3. Create Statement using Statement st= con.createStatement("<sql statement>");

      Use the appropriate Statement with the corresponding methods of the statement object.
   4. Execute statement using appropriate methods. (execute(), executeUpdate(), executeQuery()).
   5. Make use of ResultSet class for select Query.

6. Close all opened objects.

}

Catch(Exception e)

{}

}

Week-7:

1. Install APACHE TOMCAT web server and while installation, assign port number 8181. Make sure that this port is available i.e., no other process is using this port.

2. Access the above developed static web pages for books web site, using this server by putting the web pages developed in week-1 and week-2 in the document root. Access the pages by using the url : http://localhost:8181/rama/books.html.

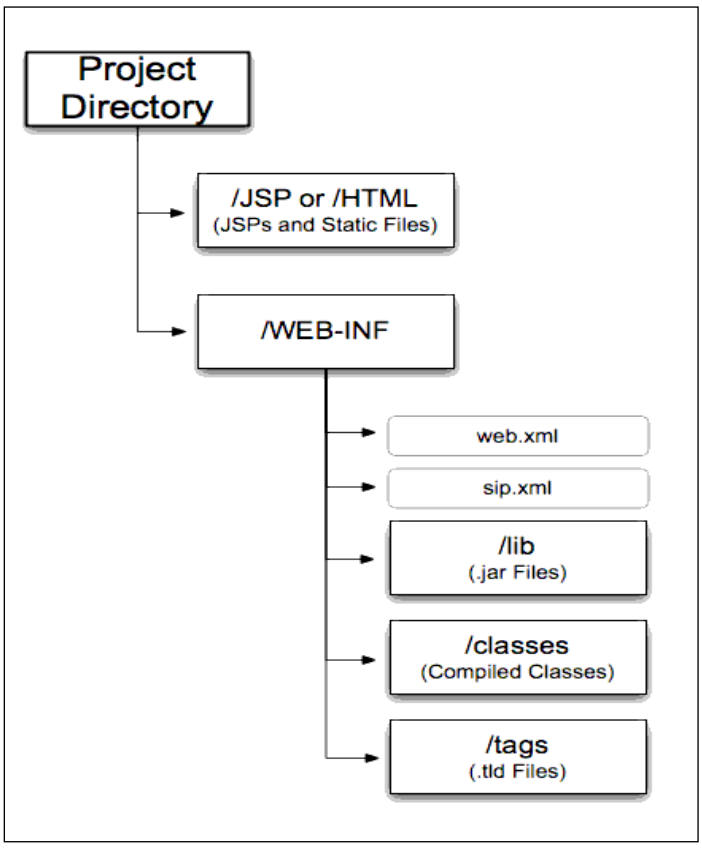


Figure 1 Directory structure for servlet

3. Write a servlet program to print welcome message on the browser.(Files to be developed-java file, and deployment descriptor)

4. Develop a web application to pass the parameters from the HTML page and display them using servlet.(Files to developed- Html,Java, Web.xml)
5. 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) .(Files to be developed- Html,Java, Web.xml)
6. Develop a web application using servlet to perform Create, Update, Retrieve and Delete (CURD) operations on the data in database from HTML form.(Files to be developed- Html,Java, Web.xml)

Servlet Prototype:

1. Servlet File:

```
import required java,sql and servlet packages;
public class Class-Name implements GenericServlets
{
Public void init( initialization parameters to a servlet)
{
Initial configurations to a servlet- if required;
}
Pubic void service(ServletRequest req, ServletResponse res)throws IOException,ServletException
{
try{
res.setContentType(“text/html”);
PrintWriter pw=res.getWriter();
//all the necessary actions of a servlets with the respective concept.
//make use of jdbc code if your are interacting with the database – same as weak-6 with respective application.
// close all the opened objects.
}
Catch(Exception e)
{}
Public void destroy(){}}
```

2. Web.xml

```
<web-app>
<servlet>
<servlet-name>--------</servlet-name><servlet-class>---------</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>-----------</servlet-name><url-pattern>-----------</url-pattern>
</servlet-mapping>
</web-app>
```

Note:

- if you are using Http-Servlets, then extend class with HttpServlets and make use of respective service method - like doGet(),doPost() etc – with request and response objects of Http-Servlets-like(HttpServletRequest req, HttpServletResponse res)
- Make use of init-parameter and context-parameter with the requirement of your

application.

- Make use of appropriate HTML file with your servlet application.

Week 8:

1. Develop a web application using servlet to perform Session Tracking with hidden form fields, cookies and url-rewriting and http sessions. (Files to developed- Html,Java, Web.xml)
2. Write a servlet using RequestDispatcher class. (Files to developed- Html,Java, Web.xml)

Week- 9:

Develop a web application using servlet to perform the user Authentication:

A. 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.

B. Authenticate the user when he submits the login form using the user name and password from the database.

Week-10:

Write a JSP which does the following job:

1. Program to print welcome message on the browser.
2. 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).
3. Develop a web application to perform Create, Update, Retrieve and Delete (CURD) operations on the data in database from HTML form.

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:

1. **Write a PHP to test the database connection**

```
<?php
$servername = "localhost";
$username = "username";
```

```
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

2. **Write a PHP to create Database**

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
   echo "Database created successfully";
} else {
   echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

3. **Write a php to create Table (for Registration Page of Week-2)**

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// sql to create table
$sql = "CREATE TABLE Registration(
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL, Phone_no number(30) NOT NULL,
email VARCHAR(50), gender varchar2(20), DOB date, Lang varchar2(20),
Address varchar2(40)
)";
if ($conn->query($sql) === TRUE) {
   echo "Table MyGuests created successfully";
} else {
   echo "Error creating table: " . $conn->error;
}
 $conn->
close();
?>
```

Week 13:

1. Write a PHP to insert values form HTML to database(registration Page)

HTML PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Records Form</title>
</head>
<body>
<form action="insert.php" method="post">
          <p>
          <label for="firstName">First Name:</label>
      <input type="text" name="first_name" id="firstName">
   </p>
   <p> <label for="phone_no">Phone number:</label>
      <input type="text" name="ph_no" id="ph_no">    </p>
   <p> <label for="emailAddress">Email Address:</label>
      <input type="text" name="email" id="emailAddress">   </p>
   <p> <label for="gender">Gender:</label>
```

```
 <input type="text" name="gender" id="genderder">  </p>
          <p>
          <label for="Address">Address:</label>
      <input type="text" name="address" id="Address">
   </p>
   <p>
          <label for="DOB">DOB:</label>
      <input type="text" name="dob" id="dob">
   </p>
   <p>
          <label for="Language">Language Known:</label>
      <input type="text" name="lang" id="lang">
   </p>

   <input type="submit" value="Add Records">
</form>
</body>
</html>
```

PHP to insert values

```
<?php
$link = mysqli_connect("localhost", "username", "password", "demo");
// Check connection
if($link === false){
   die("ERROR: Could not connect. " . mysqli_connect_error());
}
 //Fetching values form html
$first_name = mysqli_real_escape_string($link, $_REQUEST['first_name']);
$phone_no = mysqli_real_escape_string($link, $_REQUEST['phone_no']);
$email = mysqli_real_escape_string($link, $_REQUEST['email']);
$gender = mysqli_real_escape_string($link, $_REQUEST['gender']);
$address = mysqli_real_escape_string($link, $_REQUEST['address']);
$dob = mysqli_real_escape_string($link, $_REQUEST['dob']);
$lang = mysqli_real_escape_string($link, $_REQUEST['lang']);
// attempt insert query execution
$sql = "INSERT INTO registration VALUES ('$first_name', '$phone_no',
'$email','$gender','$address','$dob','$lang')";
if(mysqli_query($link, $sql)){
   echo "Records added successfully.";
} else{
```

```
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}
 // close connection
mysqli_close($link);
?>
```

2. Write a PHP to select values form database table.

```
<?php
$link = mysqli_connect("localhost", "username", "password", "demo");
// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
// Attempt select query execution
$sql = "SELECT * FROM registration";
if($result = mysqli_query($link, $sql)){
    if(mysqli_num_rows($result) > 0){
        echo "<table>";echo "<tr>";echo "<th>id</th>";echo "<th>first_name</th>";
echo "<th>phone_no</th>";
                echo "<th>email</th>";
                echo "<th>gender</th>";
                echo "<th>ADDRESS</th>";
                echo "<th>Dob</th>";
                echo "<th>lang</th>";
                echo "</tr>";
        while($row = mysqli_fetch_array($result)){
            echo "<tr>";
                echo "<td>" . $row['id'] . "</td>";
                echo "<td>" . $row['first_name'] . "</td>";
                echo "<td>" . $row['phone_no'] . "</td>";
                echo "<td>" . $row['email'] . "</td>";
                echo "<td>" . $row['gender'] . "</td>";
                echo "<td>" . $row['address'] . "</td>";
                echo "<td>" . $row['dob'] . "</td>";
                echo "<td>" . $row['lang'] . "</td>";
               echo "</tr>";
        }
        echo "</table>";
        // Free result set
        mysqli_free_result($result);
```

```
    } else{
        echo "No records matching your query were found.";
    }
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}
 // Close connection
mysqli_close($link);
?>
```

3. Write a PHP to update existing records of a database table.

```
<?php
$link = mysqli_connect("localhost", "username", "password", "demo");
// Check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
 // Attempt update query execution
$sql = "UPDATE registration SET email='xyz@abcdmail.com' WHERE id=1";
if(mysqli_query($link, $sql)){
    echo "Records were updated successfully.";
} else {
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);}
 // Close connection
mysqli_close($link);
?>
```

4. Write a PHP to validate user login

a). <html>
<title>login page</title>

```
<form method="post" action="login.php">
Username:<input type="text" name="username" placeholder="Enter username"><br>
password:<input type="password" name="password" placeholder="Enter the
password"><br>
    <input type="submit" name="submit">
    <input type="reset" name="reset">
    </form>

    </html>
```

b.)

```
<?php
$con=mysqli_connect('localhost','username','password','database');
if(isset_POSt['submit']){
$name=$_POST['username'];
$password=$_POST['password'];
$sql=mysqli_query("select name,password from table name where name=$name and
password=$password");
$result=mysqli_query($con,$sql);
if(mysqli_num_rows($result)>0)
  {
                    header("location: home.php");  }else  {     echo "login denied";  }}?>
```