AJWT

[Advanced Java & Web Technologies]

LAB EXERCISES

HTML

1. Create a web page which displays "Hello World!" text.
2. Create a web page which displays your personal details in the below format:
    - Name: xxxx
    - Gender: xxxx
    - e-mail ID: xxxx
    - Mob.No.: xxxx
    - Xth percentage: xxxx
    - Inter percentage: xxxx
    - Aggregate percentage in B.Tech: xxxx

    The main heading of the page should be "My Profile" below which the above mentioned details should be displayed. Also the left-hand side details like Name, Gender, etc., should be displayed in bold.
3. Create a web page to display your resume.
4. Create a web page to replicate Google's homepage.
5. Create a web page which displays your class time table.
6. Create a web page which displays the syllabus of AJWT as a bulleted list. Unit titles are displayed as contents of the main list and the topics of individual units are displayed as sub lists in the main list. (Hint: Use nested lists)
7. Create a web page which displays a hyperlink for each subject in your current semester. When the user clicks a link, it should open the respective subject's page. (Note: The subject pages can be empty)
8. Implement the above functionality using frames concept. Divide the web page into two parts using frames. Left frame should display the hyperlinks and when the user clicks a link, corresponding page should be loaded into the right frame.
9. Create a webpage to manage personal details like name, class, qualifications, photo, address etc., using tables and other suitable HTML tags.

CSS

1. Create a webpage which displays "Hello World" with font size 20 pixels, bold format, in "Times New Roman" font and green in colour using inline CSS, embedded CSS and external CSS.
2. Create a webpage which displays the class time table and apply the following effects on the table:
    - For the table header apply *blue* as the background colour and *white* for the colour of the text in the table header.
    - Display *day names* (Mon, Tue etc...) in bold format with the first letter in the day name in uppercase.
    - Display *lunch* slightly in bigger font other than the remaining text.
3. Create a webpage to manage personal details like name, class, qualifications, photo, address etc., using tables and other suitable HTML tags. Apply the following style information:
    - Display the heading of the page in *Times New Roman* font and with 24px size.
    - Align all the field names like Name, Class, Photo etc to *right* in the table.
    - Apply *light blue* as background colour for the left side cells in the table which contains field names like Name, Class etc...
    - Also display your college logo as background image in the top right position of the web page.
4. Create a web page containing two images, where one image overlaps another image by using the *z-index* CSS property.
5. Create a web page which displays the syllabus of AJWT as a bulleted list. Unit titles are displayed as contents of the main list and the topics of individual units are displayed as sub lists in the main list. (Hint: Use nested lists). Instead of bullets for unordered list, display arrows using CSS.
6. Create a web page which displays a hyperlink for each subject in your current semester. When the user clicks a link, it should open the respective subject's page. (Note: The subject pages can be empty). Use CSS pseudo classes on hyper links.

JavaScript

1. Create a simple script which displays "Hello World" using internal javascript and external javascript. (**Note:** Use both *document.write( )* and *alert( )* methods)
2. Create a script to read your name and mobile number and display the same on the web page. (**Hint:** Use *prompt( )* method for reading values)
3. Create a script to read a number, reverse that number and print the same on the web page.
4. Create a script to read a number and print whether the given number is an Armstrong number or not. (**Note:** You have to use *Math.floor( )* method)
5. Create a script which accepts a string and a character from the user. The script returns *True* if the character is present in the string, otherwise returns *False*.
6. Create a script to find the factorial of a given number.
7. Create a script which validates the following details in the webpage: *Name, Password, Mobile No.,* and *Email ID*. (*Name* should contains at least 6 characters, *Password* should contain at least 6 characters, *Mobile No.* should contain exactly 10 digits and *Email ID* should be in correct format).
8. Create a web page which displays some text initially and when the user hovers the mouse over that text, the background colour must change to yellow. (**Note:** This example demonstrates DHTML)
9. Create a web page which displays a *light bulb* image. Below the image display two buttons *ON* and *OFF*. When the user clicks *ON* button, the image (bulb) should glow and when the user clicks *OFF* button, the image (bulb) should return back to off state. (**Hint:** Grab two images of a light bulb, one represents ON state and the other represents OFF state. Change the images on the webpage using DHTML)

PHP

1. Write a PHP script to display 'Hello World'.
2. Create a PHP script which contains two variables which are initialized. Display the values of the variables on the webpage.
3. Create a PHP script which contains a constant named PI with value 3.142. Find the area and circumference of a circle using the constant PI.
4. Create a PHP script which finds out whether a given number is Armstrong number or not. (*Note:* Assign a number to a variable if needed)
5. Create a PHP script which validates username and password fields of a webpage. Username and password should be at least 6 characters.
6. Create a PHP script which verifies the username and password given by the user against details available in the database. (*Note:* Create a sample database and table using MYSQL)
7. Create a PHP script which stores the details given in a registration form in a table in the database. (*Note:* Create a sample database and table using MYSQL)

XML

For XML exercises store the details of a car like: id, company name, model, engine and mileage.

1. Create a well-formed XML document.
2. Create a valid XML document using DTD.
3. Create a valid XML document using XML Schema.
4. Create a XML document which contains details of cars and display the same as a table using XSLT.
5. Write a Java program to parse the XML document containing car details using SAX API.

Servlets

1. Create a servlet to display "Hello World" in the browser.
2. Create a servlet to store email-id as an initialization parameter and print the same email-id by reading the initialization parameter from the web.xml file.
3. Create a servlet to retrieve name and branch details from a html page and print the same using the servlet.
4. Create a HTML page which accepts book id, book name and book price and a submit button. When the user clicks the submit button, all the values assigned to the previous text fields must be stored in a session object and the control forwards to another servlet where the values stored in the session are retrieved and displayed.

JSP

1. Create a JSP page to display "Hello World" in the browser.
2. Create a JSP page to store email-id as an initialization parameter and print the same email-id by reading the initialization parameter from the web.xml file.
3. Create a JSP page to retrieve name and branch details from a html page and print the same using a servlet.
4. Create a HTML page which accepts book id, book name and book price and a submit button. When the user clicks the submit button, all the values assigned to the previous text fields must be stored in a session object and the control forwards to a JSP page where the values stored in the session are retrieved and displayed.

JDBC

1. Create a JSP page which accepts student regd.no. and prints the results of that student by retrieving the results from the database.
2. Create a HTML page which accepts book id, book name and book price and a submit button. When the user clicks the submit button, all the values assigned to the previous text fields must be stored in the database.

AJAX

1. Create a HTML page which accepts student regd.no. and prints the results of that student by retrieving the results from the database. Use AJAX to display the "please wait..." while the server is processing the request and print the result of the student when the server returns the result. Server resource can be either servlet or JSP or PHP.