**WEB TECHNOLOGIES – 13CS357 - LAB PROGRAM SET**

1. Write a program to split a page into 2 frames. The left side frame will have 2 text boxes and a button. The text boxes can take a shape (square or triangle) and the size of the shape. Using the shape and the size, draw the figure using asterisks (\*) in the right frame.

2. Write a program to split a page into 2 frames. The left frame has a text box and a button. On entering some text in the text box and clicking the button, the string is encoded and printed in the right frame. Encoding is as follows: **abc** is encoded as **bdf** (first letter incremented by 1, 2nd by 2 etc). Similarly **HijK** is encoded as **IkmO**. Also take care of rollovers (if first letter is Z we should get an A as encoded value). That is, **wXyZ** is encoded as **xZbD**.

3. Write a Javascript program to draw a 4\*4 square matrix. The cells have different colours. On clicking any cell, it should swap colours with another cell that is **not** its neighbour. Also develop a tool tip for each cell. That is, if you move your mouse over any cell, its number and colour appear as a tool tip just next to the cell. (Do not use the "title" attribute available on a DOM element)

4. Develop an HTML page that simulates form submission. Demonstrate the use of onblur, onload, onsubmit events. Also develop your own dialog box to confirm whether the page should be submitted or not.

5. Develop an HTML page that simulates form submission. Also Demonstrate the use of the right-click event that shows a different menu compared to the default browser menu. Show atleast 4 items in the menu

6. Write a program to demonstrate the use of sessions and database in PHP. Build a "shopping cart"-kind-of application to depict the same. Hence demonstrate save/search functionality using MYSQL database (use customer id as key in the database)

7. Write a program to demonstrate the use of cookies and files in PHP. Use "text formatting" as a use case (color, fonts etc). The formatting information is provided once by the user. Later on, cookies are used. Initial cookie values are read from a file.

8. Write a Javascript program that demonstrates scrolling of text. The text should scroll from top to bottom or from left to right (or both). Change the colour of the text periodically as scrolling happens.

9. Develop an HTML5 program to draw an animated sinewave. Provide buttons to Start, Stop, Pause and Resume the wave.

10. Write a Javascript program for "Guess my magic number" game. The script should prompt the user for guessing the magic number (a number between 1 and 100).Depending on the user's input, the script should provide hints telling him whether his guess was more or less than the magic number. It should allow him another attempt. The script should provide a max of 10 attempts to the user after which it notifies him that he failed to guess the number. If the user does not enter any number on three consecutive attempts, the script ends with a message. The "guess-history" should be maintained and displayed in a table (along with whether the guess was more or less than the magic number). The header of the table should be yellow and all cells should have their contents in blue and be centrally aligned. Use an external style sheet for the purpose. Note: You can use the Math.random() function which generates a random number between 0 and 1, to generate the magic number.

11. Write an HTML5/Javascript program that accepts a number from user. The user is prompted for a number when the "DRAW" button is clicked. The number is the period of a triangular waveform which the program draws inside a canvas area. The period of the wave is 100 times the value entered by the user (if the user entered 2, the period is 100\*2 = 200px.). Provide Stop, Pause and Resume buttons.

12. Demonstrate the use of Local Storage by developing three pages. Each page displays a paragraph of text. All pages have a footer which says C PESIT, 2016. All rights reserved. (C is the Copyright symbol). The first page has 3 text fields - one for background colour and the second for changing the font of the paragraph and the third for changing the footer text. It also has a button. The 2nd and third pages do not have these fields and button. When the user enters values (valid values) for these fields and clicks the button, the first page sets three localStorage keys corresponding to these. The field values are reflected on the page as well. background colour is changed, font is changed on the paragraph and the footer text is also changed). The same changes are reflected in the other two pages as well. When the user closes the pages and launches them again, the recent values are automatically reflected in the browser. (If there are no keys in the localStorage, then the default background colour is ivory and the default font is "Arial" and 16px. The default footer is as specified in the first page)

13. Build an HTML5 form that has two text fields. The fields represent username and password. The form also has a "LOGIN" button. When the LOGIN button is clicked, a "POST" request is made to a servlet that resides on "localhost". The servlet is called "LoginServlet". The servlet accepts the request and authenticates the user. If the authentication succeeds, it sends back a new page with the content "LOGIN SUCCESSFUL. WELCOME!!". If the authentication fails, the server sends back an error page which says "LOGIN FAILED. SORRY". There is a link in the error page which says "Back to Login Page". If you click this page, it takes you back to the Login Page.

14. Demonstrate the use of Sessions in Servlets to simulate a Shopping-cart application. Build atleast 4 pages in sequence. Provide next and prev links on every page to navigate between pages

15. Develop an HTML page that has two text fields and a drop down menu. The user can enter two numbers in the text fields. The drop-down menu has 3 options - Add, Subtract and Multiply. There is a button labeled "PERFORM OP". When the button is clicked, a JSP is invoked. The JSP instantiates a Javabean and populates it with the two values sent. The Javabean has a method called performOperation(). This takes a string as parameter. The string can have three values (Add, Subtract and Multiply). The method performs the specified action (which was selected by the user from the drop-down menu) and returns the result. The JSP calls the performOperation() method on the bean object and prints out the result in a new HTML page. The instantiation of the bean, population of its members and the method call, MUST be done using the "JSP Action Tags".