Web Application Lab.

Assignment 1

1. Download apache tomcat (.zip) and install(unzip) it. Read documentation and configure tomcat. Check the port it uses. Start the web server. Check if it is working or not. Now, write a simple HTML file (say first.html) containing your name and put it in the web server's ROOT directory. Now check if it is available through the web browser and appropriate URL. Change the port of the web server to 8002 and restart the web server. Check if it is working or not.

2. Using telnet application and HTTP, obtain the file first.html from a remote machine.

3. Design a web page containing one text box (for login name) and one password field (for password) and a submit button. Label these items.

4. Write a necessary JavaScript functions to verify the content of text box and password field in previous assignments. If there is any error, show appropriate message in red color in a label. Use following rules:
   A login name can only have alphanumeric characters and '_'(underscore)
   Number of characters in the login must not exceed 10
   A password must have 6 to 10 characters containing at least one digit and one uppercase letter.

5. Write a web page home.html that loads another page (say logout.html) if user is ideal for 10 seconds.

6. Write a simplified web page for online bill payment. The page typically contains a group of two radio buttons labeled "pay now" and "pay later". When the radio button labeled "pay later" is selected, an additional text box (to hold the date of payment) appears.

7. Write a web page that contains two drop-down lists (i.e. combo box) captioned year and semester. The first one contains values 1st, 2nd, 3rd and 4th and second one contains values 1st and 2nd. When a particular year and semester is selected, a list of subjects taught in that specific year and semester should appear.

8. Write a web page that contains two drop-down lists captioned states and districts(initially empty). The first one contains the names of states in India. When a specific state is selected, the second will have names of districts of the selected state. When a specific state and district is selected, show the information of that district in very brief.

9. Write a question paper as an XML file (question.xml say). The DTD is given below:

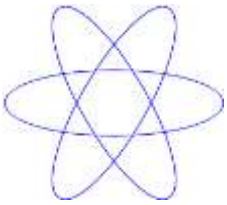
```
<!ELEMENT question-paper (question*)>
<!ELEMENT question (text, optionA, optionB, optionC, optionD, answer)>
<!ELEMENT text (#PCDATA)>
```

```
<!ELEMENT optionA (#PCDATA)>
<!ELEMENT optionB (#PCDATA)>
<!ELEMENT optionC (#PCDATA)>
<!ELEMENT optionD (#PCDATA)>
<!ELEMENT answer (#PCDATA)>

<!ATTLIST question no CDATA #REQUIRED>
<!ATTLIST answer value  (optionA|optionB|optionC|optionD) #REQUIRED>
```

Now, validate your XML file against this DTD.

10. Use HTML5 canvas and <SVG> tag to create following picture. Create one and use transformation to create others.



11. Consider a web page that locally stores (using IndexDB) the details (such as name, manufacturer, price etc) of different components(e.g. HDD, Monitor etc) of the computer. The web page contains buttons captioned “add”, “show”, “update” and “delete”. When “add” button is clicked, the pages shows a dialog box to get the details of the item to be added and inserts it to the IndexDB. Implement the functionality of the other buttons. Consider (name, manufacturer) as the primary key.

12. Write a simple web page in a web application running under tomcat. The web page displays one image and refers to one JavaScript source file. Use HTML5 application cache to cache all these in the web browser. Check if they are really cached or not. Now shutdown the tomcat and test the page is working or not.

13. Write a web page that generates a factorial table using a web worker. The main page sends an integer number (say 5) and the worker returns the factorial table (1 to 5) and the main page finally displays it.

14. Develop a simple chat web page for using WebSocket.

15. Develop a simple web page (use SSE) showing price of at least two stocks.