**12CS404 - WEB 2.0 AND RICH INTERNET APPLICATIONS – LAB PROGRAM SET**

1. A Website provides for buying sports goods. The user can buy watches - Rado and Rolex or racquettes - Yonex or Wilson. The products are limited edition and hence there is a <div> at the top of the page which displays the total number of items available for each product. Since multiple users can access the website and buy products, this information must be frequently refreshed. Demonstrate the functionality by employing the periodic refresh pattern. Also implement and demonstrate the exponential back-off pattern for failure.

2. An HTML5 page has an animation running. The animation works as follows. A miniature red-coloured square starts at the center of a canvas element and expands (retaining its center coordinates). When it touches the outer walls of the canvas, it disappears and a miniature circle now starts at the center of the canvas. This circle starts expanding (while retaining its center coordinates) until it touches the canvas walls, whereupon the square takes over again. There is also a button which performs an involved calculation. Demonstrate the side effects of the calculation on the animation. Hence solve the problem using Web Workers.

3. Create an HTML page that has a form with 4 text fields. The fields are USN, NAME, DEPT and GPA. Only the "USN" field is editable. The "NAME", "DEPT" and "GPA" fields are NOT editable. They are READ-ONLY. The student can enter a valid USN in the USN field. As soon as the student exits the "USN" field, an AJAX call is made to a server script. The server returns the NAME, DEPT and GPA values corresponding to the USN entered. If the USN entered is invalid, the server returns "INVALID USN". This string is then appended (next to the USN field) on the screen. If proper values are returned for NAME, DEPT and GPA, the values are populated into the respective text fields. (Remember these are read-only and hence only the server can populate these using Javascript. The server side script can be trivial and return hard-coded values). Implement the functionality using hidden iframes. Demonstrate the unique advantage of this technique.

4. An HTML5 page has an animation running. The animation is a sine-wave. That is, a sine wave is drawn in real time. The wave starts at origin (you can take origin as 100,100). The wave has a period of "100px". Once the wave reaches the right end of the canvas, the wave is cleared and starts again at origin. This is perpetual (never ending).There is also a button which performs an involved calculation. Demonstrate the side effects of the calculation on the animation. Hence solve the problem using Web Workers.

5. An HTML5 page has an animation running. The animation works as follows. There are two balls bouncing off each other and also the walls of the canvas. This action is perpetual. There is also a button which performs an involved calculation. Demonstrate the side effects of the calculation on the animation. Hence solve the problem using Web Workers.

6. A book-search web page served by localhost is rich in content and GUI. It has multiple images to display, a search form for the front-end user to search for a book, a set of links which might interest the user and also general information on how to use the site. Use Multi-stage download pattern to demonstrate good user experience with this page. On entering an ISBN in the search box and searching for it, return the matching book from the server. The server maintains book information in a database.

7. A Website provides for Engg seat Counseling. The user can opt colleges - PESIT and RVCE and courses - CSE or ECE. The number of seats are limited and hence there is a <div> at the top of the page which displays the total number of seats available for each course/college. Since multiple users can access the website and opt for courses, this information must be frequently refreshed. Demonstrate the functionality by employing the periodic refresh pattern. Implement a simple fallback pattern for this using exponential back-off method.

8. Implement a photo slideshow using JQuery libraries. There must be atleast 3 photos in the album. When the first photo is displayed, only a "next" link is available. When the second photo is displayed, there is a "next" and "previous" link.When the last photo is displayed, we only have a "previous" link. Provide animations for displaying the photos.

9. Implement a Javascript/AJAX program to fetch RSS feeds from a well-known RSS feed site. Provide a scrolling display of latest news on your page. You can use xparser.js if you like.

10. Implement an RSS-based search feature. Have a text box and a button in your page for the same. Show the results in a separate <div> which has the results as hyperlinks, which the user can click.

11. Use the Reverse AJAX technique to build a web-based chat application. The application is two-way browser-based. That is, we have a window in which one user types his messages. From other other side, the second user also updates a browser window. Show the chat windows side-by-side and demo it.

12. A file on a server has information about cricket players. The fields represent name, country, matches, runs and centuries. The fields are separated by colons (:). The front end screen has a text field in which the user can enter a country. The server returns details of all players belonging to that country in the form of one big JSON object. The client parses the JSON object and builds an HTML table to print the results. Implement the server side script and the client code.

13. By implementing a single server side PHP script, demonstrate the working of a RESTful Web Service. The service must essentially provide the CREATE, READ and UPDATE operations. For each operation, the server returns data in json format. Also implement a PHP script which acts as client to this service. You need to show URL mapping using Apache’s configuration file.

14. Implement a simple Mashup using Google Map APIs and AJAX techniques. Show a map of Bangalore, centred on PESIT. Add markers at PESIT, PESSE, RVCE and BMSCE with appropriate titles for each college. The latitude and longitude of each of these colleges is stored in a file and should be read using AJAX techniques (for adding the markers).

The values are: PESIT: 12.9342678, 77.53432629999998, PESSE: 12.8614515, 77.66470809999998, RVCE: 12.9247501, 77.49875709999992, BMSCE: 12.9404973, 77.56568749999997