16.3

Ajax Web Applications – Uses client-side scripting to make web applications more responsive. Ajax applications separate client-side user interaction and server communication and run them in parallel, reducing the delays of server-side processing normally experienced by the user.

Traditional Web Applications – (Synchronous Request) User fills in and submits form. Next, browser generates a request to the server, which receives request and processes it. The server generates and sends a response containing the exact page the browser will render, which causes browser to load new page. Browser window is now temporarily blank as client waits for the server to respond and reloads the entire page with the data from the response.

Ajax applications – (Asynchronous Request) layer added between client and server to manage communication between the two. User interacts with page – client creates an XMLHttpRequest object to manage a request. XMLHttpRequest object sends the request to the server and awaits response. Requests are asynchronous so user can continue to interact with the application on the client side while the server processes the earlier request concurrently. Other user interactions can continue to take place. Once server responds to original request, XMLHttpRequest object that issued the request calls a client-side function to process the data returned by the server. (CallBack Function) uses partial page updates to display the data in the existing web page without reloading the entire page. At the same time the server can be responding to 2nd request and so on…

16.5

A) Partial Page Update is a step utilized by Ajax Applications. (CallBack Function) uses partial page updates to display the data in the existing web page without reloading the entire page. At the same time the server can be responding to 2nd request and so on… Such partial page updates help make web applications more responsive, making them feel more like desktop applications. Web application does not load a new page while the user interacts with it.

B) Asynchronous Request – Ajax applications separate client-side user interaction and server communication and run them in parallel, reducing the delays of server-side processing normally experienced by the user. User can interact with web application on client side while server processes earlier request concurrently.

C) XMLHttpRequest object - The keystone of AJAX is the XMLHttpRequest object.

All modern browsers support the XMLHttpRequest object.

The XMLHttpRequest object is used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

D) Ajax - AJAX stands for Asynchronous JavaScript and XML. It is the use of the XMLHttpRequest object to communicate with server-side scripts. It can send as well as receive information in a variety of formats, including JSON, XML, HTML, and even text files. AJAX’s most appealing characteristic, however, is its "asynchronous" nature, which means it can do all of this without having to refresh the page. This lets you update portions of a page based upon user events.

The two features in question are that you can:

Make requests to the server without reloading the page

Receive and work with data from the server

E) CallBack Function - The AJAX callback function – the function we assign as the onreadystatechange event handler – is the central part of our AJAX requests. The callback function is responsible for checking the progress of requests, identifying the result of the request and handling data returned from the server. Callback functions also serve as delegators, handing off to other areas of your application code.

F) Same-Origin Policy - The same-origin policy restricts how a document or script loaded from one origin can interact with a resource from another origin. It is a critical security mechanism for isolating potentially malicious documents.