Module - 5

PAGE NO :
DATE :

* **Ajax principles.**

As a new web application model. Ajax is still in it's infancy. However, several web developers have taken this new development as a challenge. The challenge is to define what makes a good ajax web application

- Minimal traffic
- No surprises
- Established conventions
- No distraction
- Accessibility
- Avoid entire page downloads
- User first.

- **Minimal traffic.**

Ajax application should ~~send~~ send and receive as little information as possible to and form the server, In short. Ajax can minimize the amount of traffic between the client and the server.

- **No surprises.**

Ajax applications typically introduce different user interaction model than traditional web applications. some ajax applactions use other user interface such as drag and drop or double clicking.

- **Established Conventions:**

Don't waste time inventing new user intrac-tions model that your user untamilion with. Borrow heavily from traditional web applicatio

and desktop applications, so there is minimal learning curve.

- No Distractions: Avoid unnecessary and distracting page elements such as looping animations and blinking page sections. such gimmicks distract the user from what he or she is trying to accomplish.

- Accessibility: Consider who your primary and secondary users will be and how they most likely will access your ajax application.

- Avoid Entire page and downloads:
All server communication after the initial page download should be managed by the ajax engine. Don't ruin the user experience by downloading small amounts of data in one place but reloading the entire page in others.

- User first:
Design the ajax application with the users in mind before anything else. Try to make the common use cases easy to accomplish & don't to caught up with how youre going to fit in advertising or cool effects.

PAGE NO :
DATE :

* Ajax Communication Technique

There a lot of request going back and forth between the browser and the server while you are surfing the web. Initially all these requests happens because the user made an action that required such a step.

1] The Hidden frame techniques :-

The basic idea behind this technique is to create a frameset that has a hidden frame that is used for client - server communication

we can hide the frame by setting it's width or height to zero.

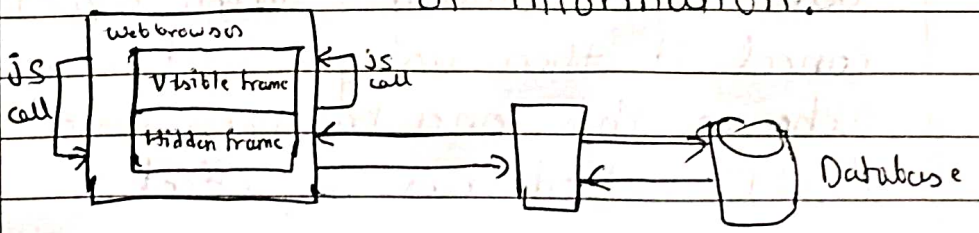
2] Hidden frame GET request :

for example :

The user enter a customer Id and receive in return information about the customer, for this we can use server-side programming language PHP and open source MySQL Database (Retrive information from the Databases)

3] Hidden Frame POST request :-

- Request Data from from the server
- A post request can send up to 2MB of information.



4] Hidden iFrames :-

It is introduced in HTML 4.0, the iframe technique can be applied to pages not originally created as a frameset. iframes are ideal for ajax communication.

5] Hidden iFrame Post Request.

To accomplish a post request using hidden iFrames, the approach is a load page that contains a form into the hidden frame.

* Cancel Pending requests with respect to Ajax :-

Ajax request cancellation is the process of stopping an on going a synchronous Http request made with Ajax technology. it can be achieved by using the abort() method of the XHR (XMLHTTP Request) object.

The abort() method stops the request & terminates the response

This is useful in situations where the user initiates a request but then decides to cancel it before it has been completed.

for example, the user request a some data then sometimes later want to cancel it then using abort() we can achieve this. once has been canceled, it's abort() method has no effect.

PAGE NO :
DATE :

Syntax :-

```
var Xhr = new XMLHttpRequest();
Xhr.abort();
```

* Control patterns and Predictive fetch with an example.

1) Predictive fetch

The Ajax application guess what the user is going to do next and retrieve appropriate Data. in perfect world. it would be wonderfull to always know what the user is going to do and make sure that the next data is readily available when needed. Determining future user actions is just a guessing game depending on your intentions

for example : Suppose you are reading online articles and that contains 3 pages. If you are interested in reading the first page, it logically assume that you are also interested in reading 2nd & 3rd pages, then it download 3 pages in background.

PAGE NO.
DATE :

2] Page Reloading

One of the simplest and most logical uses of the predictive fetch pattern is in the preloading of pages in an online article.

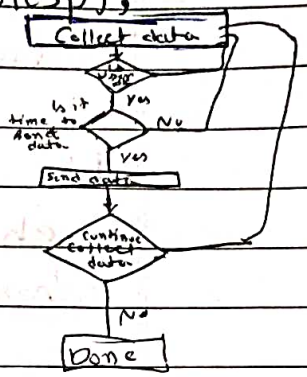
Reading long articles online is the very difficult on the eyes. so many sites split them into many pages but it takes longer time to load because each new page wants to load menus and all. predictive fetch uses the load on both client and server by loading only the text for the next pages while the reader is still reading the first page

Example

```
function make preloader (Parent node)
{
 var img = document. create element ("img");
img.src = "images/preloader.gif";
parent Node. append Child (img);
}

 function getData().
 {
  var parentNode = document. getElementById
       ("myParent Node");
  make preloader (Parent Node);
  $. ajax ("http://www.example.com");
      {
```

```
success : function(resp):
    {
      $(ParentNode).html(resp);
    }
    async : false
  }
 };
}
```



* Submission throttling

Predictive fetch is one pattern for retrieving data from the server, the other side of an ajax solution is the sending of data to the server.
In ajax model, the user interacts with the site or application without addition -al request generated for each click.

For example

when the user types a letter, each time they submit to the server and request made to the server because of this server busy happened. the submission throttling is an alternative approach to this problematic issue.

- Using Submission throttling, your buffer the data to be sent to the server on the client and then send the data at predefined times.

PAGE NO :
DATE :

* **Field Validation**

The popular form of the submission throttling design pattern involves submitting a single field periodically as changes are made.

The submission throttling validates the each field Instead of whole field when everytime changes are made.

for example

Instead of asking to fill the form, you need to signup for the application using username, unless untill the username format is correct we didn't submit the form.

Ex :-

```
<form method = "post" action = "Success.php">
<table>
   <tr>
      <td> Username </td>
      <td> input type = "text" id = "usr"
                  name = "usr" >
      <img src = "error.gif", alt = "Error">
   </td>
   </tr>
</table>
<input type = "submit" >
</form>
var xmlhttp = null;
var Timeoutid = null;
function validateField(OEvent)
   {
```

```
oEvent = oEvent || window.event
var txtfield = oEvent.target ||
    oEvent.srcElement;
if (iTimeoutId != null)
  {
    clearTimeOut(iTimeOut);
    iTimeoutId = null;
  }
}
```

* Multi Stage download:

One of the lasting problems on the web has been the speed at which pages download.

Multi stage download is an Ajax pattern where in only the most basic functionality is loaded into a page initialy. then begins to download other components that should appear on the page. If the user should leave the page before all the components are downloaded. it's of no consequences.

PAGE NO :
DATE :

HTML V/S Ajax

HTML

- HTML is the standard markup language used to create the structure and content of web pages.
- It consists of set of elements or tags that define the structure and layout of a webpage such as headings, paragraphs, links, images and forms.

Ajax

Ajax is a web development technique that allow web applications to retrieve and send data to a server.

HTML	Ajax
HTML is used for creating the static structure and content of web pages	Ajax is used for making asynchronous requests to a server and updating parts of a webpage.
HTML is a Markup language used to structure content.	Ajax is a technique that primarily involves javascript for making asynchronous requests.
HTML is not designed for data retrieval & manipulation.	Ajax is specifically designed for data retrieval and can update webpage content
HTML loads an entire web page when requested resulting in a full page reload.	Ajax loads only the data needed from the server.