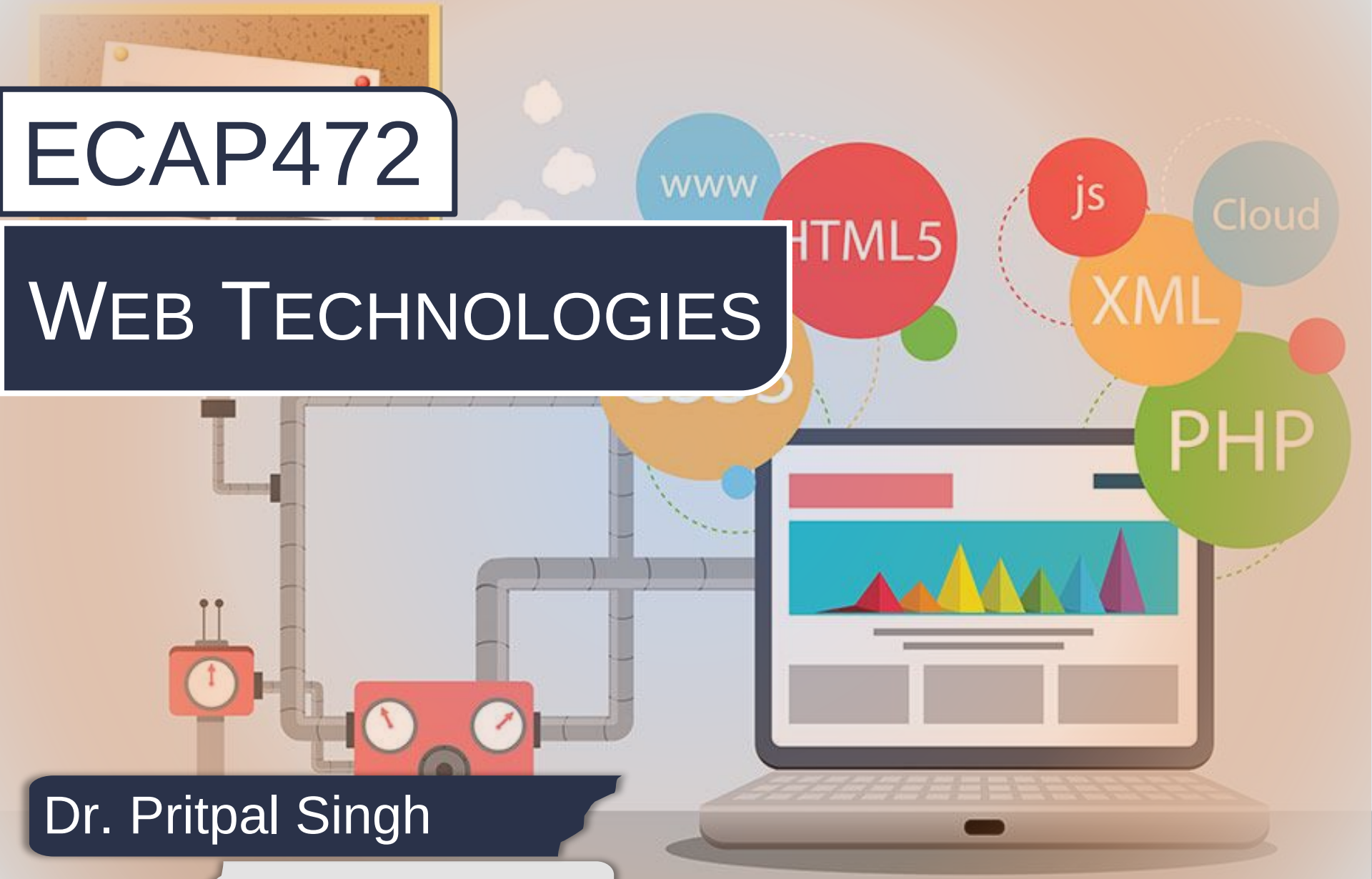
ECAP472

Web Technologies



Dr. Pritpal Singh

Associate Professor

Learning Outcomes

After this lecture, you will be able to

- understand concept of JavaScript events,
- understand concept of JavaScript cookies.

Web Applications

- As browsers and personal computers have continued to improve, JavaScript gained the ability to create robust web applications.
- Consider applications like Google Maps. If you want to explore a map in Google Maps, all you have to do is click and drag with the mouse. You will see the part of the map that is less detailed and then fills itself in. That’s the work of JavaScript behind the scene.

Games

- While the browser hasn’t been the traditional games platform in the past, recently it has become robust for games. Additionally, with the addition of HTML5 canvas, the level of complexity that is possible in the browser-based games has increased exponentially.

Why JavaScript Projects?

- Good knowledge of JavaScript can get you a range of challenging and interesting career options like developing mobile and desktop apps, building dynamic websites from scratch.
- If you follow a bit of HTML & CSS, you will understand most of the Javascript projects

JavaScript Events

- The change in the state of an object is known as an Event. In html, there are various events which represents that some activity is performed by the user or by the browser. When javascript code is included in HTML, js react over these events and allow the execution. This process of reacting over the events is called Event Handling. Thus, js handles the HTML events via Event Handlers.
- For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.

Mouse Events:

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	Onkeydown & onkeyup	When the user press and then release the key

Form Events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Window/Document Events

Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

JavaScript Cookies

- A cookie is an amount of information that persists between a server-side and a client-side. A web browser stores this information at the time of browsing.
- A cookie contains the information as a string generally in the form of a name-value pair separated by semi-colons. It maintains the state of a user and remembers the user's information among all the web pages.

How Cookies Works?

- When a user sends a request to the server, then each of that request is treated as a new request sent by the different user.
- So, to recognize the old user, we need to add the cookie with the response from the server.
- browser at the client-side.
- Now, whenever a user sends a request to the server, the cookie is added with that request automatically. Due to the cookie, the server recognizes the users.

How to Create a Cookie in JavaScript?

- In JavaScript, we can create, read, update and delete a cookie by using document.cookie property.
- The following syntax is used to create a cookie:
- document.cookie="name=value";

What are Cookies?

- Cookies are data, stored in small text files, on your computer.
- When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user.
- Cookies were invented to solve the problem "how to remember information about the user":

What are Cookies?

- When a user visits a web page, his/her name can be stored in a cookie.
- Next time the user visits the page, the cookie "remembers" his/her name.

Create a Cookie with JavaScript

- JavaScript can create, read, and delete cookies with the document.cookie property.
- With JavaScript, a cookie can be created like this:
- document.cookie = "username=John Doe";

Create a Cookie with JavaScript

- You can also add an expiry date (in UTC time). By default, the cookie is deleted when the browser is closed:
- document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC";

Read a Cookie with JavaScript

- With JavaScript, cookies can be read like this:
- let x = document.cookie;
- document.cookie will return all cookies in one string much like: cookie1=value; cookie2=value; cookie3=value;

The Cookie String

- The document.cookie property looks like a normal text string. But it is not.
- Even if you write a whole cookie string to document.cookie, when you read it out again, you can only see the name-value pair of it.
- If you set a new cookie, older cookies are not overwritten. The new cookie is added to document.cookie, so if you read document.cookie again you will get something like:
- cookie1 = value; cookie2 = value;

Advantages of Cookies:-

1. Cookies do not require any server resources since they are stored on the client.
2. Cookies are easy to implement.
3. You can configure cookies to expire when the browser session ends (session cookies) or they can exist for a specified length of time on the client computer (persistent cookies).

Disadvantages of Cookies:-

1. Users can delete a cookies.
2. Users browser can refuse cookies,so your code has to anticipate that possibility.
3. Cookies exist as plain text on the client machine and they may pose a possible security risk as anyone can open and tamper with cookies.

That’s all for now…