JavaScript Assignment-3

1. Inline Function:

Inline function is powerful concept that is commonly used with classes. If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time. The compiler can ignore the inline qualifier in case defined function is more than a line. It is to increase the execution time of a program.

When we use Inline Function:

1.Use inline function when performance is needed.
2. Use inline function over macros.
3. Prefer to use inline keyword outside the class with the function definition to hide implementation details.

Which Languages use Inline Function:

C,c++,JavaScript

Inline Function in JavaScript:

In JavaScript, inline function is a special type of anonymous function which is assigned to a variable, or in other words, an anonymous function with a name. ... Unlike normal function, they are created at runtime.

```
var inline_func = function() {
 alert("Hello I'm inside inline function")
}
$('#inline-function').click(inline_func)
```

2. Process Of DOM updation:

The 'update' action will generally always consist of three things happening:

1. Listen for an action to occur to initiate the '*update*'
2. Perform a '*fetch*' with the method of 'PATCH' to update the back-end
3. Have the '*update*' reflected on the DOM

1. Listen for an action to occur to initiate the update:

In most cases, for an update to occur, an action needs to occur to initiate the actual '*updating*'. Whether it's the user clicking on a checkbox or entering something into a form and hitting submit, we have to add an *eventListener* to a specific element in the HTML to '*listen*' for a specific event to occur.

Ex: checkbox.addEventListener("click",(),=>{})

An *eventListener* takes a callback function to run when the event happens and this is where step 2 comes in.

2. **Perform a '*fetch*' with the method of '**PATCH**' to update the back-end**

Inside of this callback function is generally where we would want the '*fetch*' to occur. This is the step that will be most dependent and specific to the way your DOM is set up. Sometimes, the element that the *eventListener* was added to can already identify the specific object that is to be updated .

where ID is being string-interpolated into the URL. The second argument of the '*fetch*' will be an object with three keys: method, headers and body.

fetch(url){

method: "PATCH',

headers:{ }

body:{}

}

3. **Have the '*update*'** reflected **on the DOM**

Just like with a simple GET request with '*fetch*', we can use the '*.then*' method to get a response back and use that response to change the DOM.

3. Differences between webpage & web application:

WebPage:

A webpage is defined as a single document or a solitary page of any website.

It can be developed and maintained by an individual, business or organization. The website aims to serve a variety of purposes.

A website is hosted on a single or multiple web server. It is accessible via a network like the Internet or a private local area network via IP address.

Every webpage is attached to a unique URL address used to render or access that particular page.

Web Application:

A web application is a software or program which is accessible using any web browser.

In a web application, the user not only read the page content but also manipulate the restricted data.

Its frontend is usually created using languages like HTML, CSS, Javascript, which are supported by major browsers.