UNIT-7

Client-side Development in ASP.NET Core

⊛ Common client-side web technologies:

ASP.NET Core applications are web applications and they typically rely on client-side web technologies like HTML, CSS and JavaScript. By separating the content of the page (the HTML) from its layout and styling (the CSS), and its behaviour (via JavaScript), complex web apps can leverage the Separation of Concerns principle. While HTML and CSS are relatively stable, JavaScript, by means of the application frameworks and utilities developers work with to build web-based applications. Angular, jQuery, React, Vue are javascript frameworks and libraries.

⊛ JQuery:

jQuery is a fast, small and feature-rich JavaScript library included in a single .js file. It provides many built-in functions using which developers can accomplish various tasks easily and quickly. Some of the jQuery important features are:

DOM Selection: jQuery provides Selectors to retrive DOM element based on different criteria like tag name, id, css class name, attribute name, value, nth child in hierarchy etc.

DOM Manipulation: We can manipulate DOM elements using various built-in jQuery functions. For example, adding or removing elements, modifying html content, css class etc.

Special Effects: We can apply special effects to DOM elements like show or hide elements, fade-in or fade-out of visibility, sliding effect, animation etc.

Events: jQuery library includes functions which are equivalent to DOM events like click, dbclick, mouseenter, mouseleave, blur, keyup, keydown etc. These functions automatically handle cross-browser issues.

Ajax: jQuery also includes easy to use AJAX functions to load data from servers without reloading whole page.

Cross-browser support: jQuery library automatically handles cross-browser issues, so the user does not have to worry about it.

⊛ Advantages of jQuery:

Easy to learn: jQuery is easy to learn because it supports same JavaScript style coding.

Write less do more: jQuery provides a rich set of features that increase developers productivity by writing less and readable code.

Excellent API Documentation: jQuery provides excellent online API documentation.

Cross-browser support: jQuery provides excellent cross-browser support without writing extra code.

Unobtrusive: jQuery is unobtrusive which allows separation of concerns by separating html and jQuery code.

⊛. jQuery Syntax:

The jQuery syntax is used to select HTML elements and perform some action on those element(s). Basic syntax is: $(selector).action()

→ A $ sign to define/access jQuery.

→ A (selector) to "query (or find)" HTML elements.

→ A jQuery action() to be performed on the element(s).

Example: Using id selector with id=test will be hidden on button click.

```
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
```

⊛. jQuery vs SPA Framework:

Factor	jQuery	Angular
Abstracts the DOM	Yes	Yes
AJAX Support	Yes	Yes
Declarative Data Binding	No	Yes
MVC-style Routing	No	Yes
Templating	No	Yes
Deep-link Routing	No	Yes

⊛ Single Page Application (SPA) Frameworks:

1) Angular: Angular remains one of the world's most popular JavaScript frameworks. Angular applications are built from components. Components combine HTML templates with special objects and control a portion of the page. Components are defined using the @Component decorator function, which takes ~~element on the page where~~ in metadata about the component. The selector property identifies the ID of the element on the page where this component will be displayed.

The template property is a simple HTML template that includes a placeholder that corresponds to the component's name property, defined on the last line. By working with components and templates, instead of DOM elements, Angular apps can operate at a higher level of abstraction. Angular also imposes some order on how we organize our client-side script files.

Advantages of Angular JS:

→ Open source JavaScript MVC framework.
→ Supported by Google.
→ No need to learn another scripting language. It's just pure JavaScript and HTML.
→ Built-in attributes makes HTML dynamic.
→ Easy to extend and customize.
→ Easy to Unit test.

2) React: Unlike Angular, which offers a full Model-View-Controller pattern implementation, React is only concerned with views. Its not a framework, just a library. There are a number of libraries that are designed to be used with React to produce rich single page applications.

One of React's most important features is its use of a virtual DOM. The virtual DOM provides React with several advantages, including performance and testability. Rather than having a strict seperation between code and markup, React adds HTML directly within its JavaScript code as JSX. JSX is HTML-like syntax that can compile down pure JavaScript.

⊛. Choosing a SPA Framework:

When considering which JS framework will work best to support our SPA, keep in mind the following considerations:
→ Is your team familiar with the framework and dependencies?
→ How opinionated is the framework, and do you agree with its default way of doing things?
→ Does it include all of the features our app requires?
→ Is it well documented?
→ How active is its community?
→ Are issues being resolved and new versions shipped regularly?

⊛ Forms and Validation:

Q. Design a page to show client-side validation for login page using jquery or angular or react. [Model set Imp]

Solution:

Client Side Validation using jQuery:

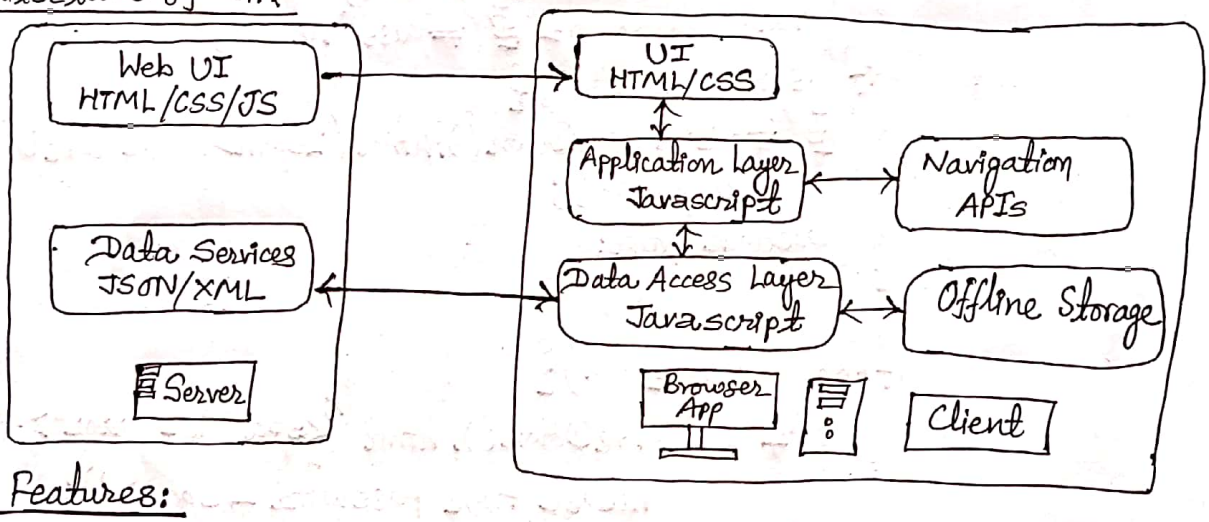
```
<html>
  <head>
      <title>Jquery login page validation </title>
      <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.
              min.js" type="text/javascript">
      </script>
      <script type="text/javascript">
        $(document).ready(function(){
          $('#submit').click(function(){
              var username=$('#user').val();
              var password=$('#pass').val();
            if(username==""){
              $('#dis').slideDown().html("<span>Please type Username
                 </span>");
              return false;
            }

            if(password==""){
              $('#dis').slideDown().html('<span id="error"> Please
                                type password </span>');
              return false;
            }
          });
        });
      </script>
  </head>

  <body>
      <fieldset style="width:250px;">
        <form method="POST" action="">
          <label id="dis> </label> <br>
```

```
Username: <input type="text" name="user" id="user" /> <br/>
Password: <input type="password" name="pass" id="pass" /> <br/>
<input type="submit" name="submit" id="submit" />
</form>
</fieldset>
</body>
</html>.
```

Q. What is Single Page Application? Draw architecture of SPA and explain its features. [Model Set Question Imp.]

Ans: SPA is a web application taking a single HTML page. Delivering dynamic updates, it allows interacting with the page without refreshing it. Making use of single page app we can significantly decrease the server load and enhance loading speed for a better UX. This is possible because SPA only boots data on demand while the entire page has previously been loaded.

Architecture of SPA



SPA Features:

i) No server roundtrip: SPA can easily reshape any part of UI to restore the full HTML page preventing server roundtrip.

ii) Templating and routing on the client side: Supported by JS-based routers, SPA can monitor the user's actual state and location within the whole navigation experience.

iii) The compatibility to work offline: A SPA can work offline in the event of a lost internet connection, connection is restored when online.

iv) Cross-platform functionality: SPA work on different operating systems from any browser.

v) A reduced throughput: The connection with server is carried out in small fragments. Data is packed into smaller objects like JSON data format.