SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI

Engineer to Excel

Course Code /Title: CSA4399 – Internet Programming
Programme : Computer Science and Engineering

ASSIGNMENT QUESTIONS

SET -2

1	You are updating a legacy web application to take advantage of modern web standards. The project involves transitioning from an earlier HTML version to HTML5. The goal is to leverage HTML5's new features to improve web development practices and enhance the functionality of web forms.	10	CO1	2
2	Designing a user registration form for a new web application. The form needs to capture essential information such as name, email, and a message from users. It is crucial to implement client-side validation to ensure that the data entered is accurate and complete before the form is submitted.	10	CO1	3
3	Designing a website that needs to function effectively across various devices and screen sizes. The design should include different types of layouts such as fixed, fluid, and responsive. To achieve this, you need to use CSS techniques like Flexbox or Grid to ensure that the layout adapts well to different screen sizes and maintains a consistent user experience.	10	CO1	2
4	To create a webpage that offers a seamless user experience across devices, responsive design principles must be applied. This involves using fluid grids, flexible images, and media queries to ensure the layout adjusts gracefully to different screen sizes. For instance, on a desktop, the webpage might display multiple columns with detailed content, while on a smartphone, the same content could stack vertically for easy scrolling. Navigation menus should transform into a hamburger icon on smaller screens, ensuring they remain accessible without taking up too much space. Additionally, touch-friendly elements and optimized images ensure fast loading times and a smooth experience on all devices.	10	CO1	3
5	Given a scenario (e.g., creating a blog post, a product listing), design a webpage using appropriate elements, tables, lists, and images for optimal readability and user experience. Also, describe step-by-step the sequence of HTTP requests and responses that occur when a user accesses a webpage containing multiple resources (HTML, CSS, JavaScript, images).	10	CO1	3

ASSIGNMENT - I

1. You are updating a legacy web application to take the advantage to take modern web standards. The project involves transitioning from an earlier version to HTML 5.

When updating a legacy web application to HTML 5.
We can take advantage of several features like.

1) Semantic elements:

⇒ Use Semantic Tags: Replace non-semantic divs and spans with HTML 5.

Eg:

```
<header>
<h1> Website Title </h1>
</header>
<nav>
<ul>
<li> <a href="#home"> Home </a> </li>
</ul>
</nav>
<section>
<article>
<p> Content goes here </p>
</article>
</section>
<footer>
<p> Copy : 2024 Company name </p>
</footer>
```

2. Enhanced form controls:

New input types: utilize new input types such as 'email', 'date', 'url' to provide better user input validation and UI enhancements.

Eg:

```
<form action = "/submit" method = "post">
<label for = "email"> Email: </label>
<input type = "email" id = "email" name = "email" required>
```

3. Multimedia Integration

Native audio and video:- Replace flash or other plugins with native HTML5.

Eg:

```
<video controls>
<source src = "movie.mp4" type = "video/mp4">
</video>
```

4. Offline Capabilities and Storage:

use localStorage and SessionStorage.

Eg

```
localStorage.setItem('username', 'Bhaskar');
SessionStorage.setItem('session ID', '12345');
Const username = localstorage.getItem('username');
Const SessionID = SessionStorage.getItem('session ID');
```

5. Responsive Design:

→ Viewport and Media Queries:

Ensure the application is responsive by defining the viewport.

eg:

```
<meta name = "viewport" content = "width = device width,
        initial scale = 1.0">
<style>
  body {
  font-family : Arial, sans-serif;
  }
```

6. Improved Accessibility:

ARIA Roles: Enhance accessibility using ARIA

Eg:

```
<nav aria-label = "main Navigation">
<ul>
<li><a href = "#Home"> Home </a></li>
</ul>
</nav>
```

7. Progressive Enhancement and graceful degradation:

Enhance new features: Implement new HTML 5 features.

```
<script src = "modernizr.js"></script>
<script>
if (!modernizr.inputtypes.date) {
}
</script>
```

2. Design a user registration form for a new web application.

When designing a user registration form for a new web application, capturing essential info like name, email. requires careful to both usability and data validation.

HTML Structure:

```
<!DOCTYPE html>
<html lang = "en">
 <head>
  <meta charset = "UTF-8">
  <meta name : "viewport" content = "width = device. width
  <title> user Registration form </title>
  <style>
   body {
   font family: Arial, sans serif;
   background-color : #ffffff;
   Margin : 0;
   padding : 20px;
  }
  . container {

  max width : 500px;
```

```
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
h1 {
  text-align: center;
  margin-bottom: 20px;
}
label {
  display: block;
  margin-bottom: 8px;
}
input, textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 16px;
}
button {
  width: 100%;
  padding: 10px;
  color: white;
  border: none;
  border-radius: 5px;
  font-size: 18px;
  cursor: pointer;
}
```

```
button : hover {
background : color : #21883 ;
}

.error {
    Color : red ;
    font size : 14 px;
    margin top : -15 px;
    margin bottom : 10 px;
    }
</style>
</head>
<body>

<div class = "container ">
<h2> user Registration form </h2>
<form id = " registration form ">
  label for = "name "> Name: </label>
  <input type = " text " id = "name " name = "name "
  <div class = "error " id = "Nameerror "></div>
  <label for = "email "> email : </label>
  <button type = "submit "> Register </button>
  </form>
  </div>
```

3. Designing a website that needs to function effectively across various devices and screen sizes. The design should include different types of layouts such as fixed, fluid and responsive.

Understanding layout types:

Fixed layout: A fixed layout uses specific pixel values for width, meaning the layout remains the same regardless of the screen size.

Fluid layout: A fluid layout uses percentage based widths, allowing the layout to adapt to the different screen sizes.

Responsive layout:

A responsive layout combines both fixed and fluid layouts, along with media queries to adapt to various screen sizes, ensuring a consistent user experience.

Using CSS Techniques:

Flexbox: Flexbox is ideal for one dimensional layouts, where you need to align items along a single axis.

Grid: CSS grid is perfect for two dimensional layouts where both rows and columns are essential. It provides more control over the layouts.

Combining layout types:

fixed header with fluid content

CSS

```
body, html {
  margin : 0;
  padding : 0;
}
header {
  width : 100%;
  position : fixed;
  top : 0;
  background-color : #333;
  color : white;
  padding : 10px 0;
  text-align : center;
}
main content {
  margin-top 60px; /* push content
  below the header */ padding : 20px;
  width : 80%; /* Fluid width */
  max-width : 1200px; /* cap the
  width for larger screens */
  margin-left : auto;
  margin-right : auto;
}
```