

DEVELOPMENT OF WEB USER INTERFACE: QUIZ:-

Section I:

1. a) **HTML**:- is the structure of the webpage, it defines the context like headings, paragraphs, images e.t.c.
CSS:- is the design and style of the page, it controls colour, fonts e.t.c.
Javascript:- it is the interactiveness and behaviour of the page, making things like buttons react or links.

b) The browser loads HTML, making a document object model. Then, it loads CSS, making CSS object model, then javascript can update what you see and how it looks on a webpage.
3. a) The browser looks for the real IP address of the website so knows where to go. The browser then talks to the website's server. The browser sends a message asking if it could give it its webpage. The website's computer sends files back, in a way of the HTML, CSS, and Javascript way. The browser then puts everything together and shows the website on your screen.

b) **DOM**: - Created from HTML for structure
CSSOM: - Created from CSS for styling
Render Tree: - Combination of DOM & CSSOM for visual effects
Then the browser paints it on the screen and displays it.
4. **Causes**:- Some features are supported only in modern browsers, and some browsers may show different CSS features or rules differently.
Solutions:- Test your website across different browsers like Chrome or Edge. And make sure that there is a consistent base style.

Section II:

7. a) **Action**: Specifies where data is sent
Method: It's how data is sent
Name: Identifies each input
Required: Makes sure each input is filled.

```
b) <form action="/submit" method="POST">
  <label for="username">Name:</label>
  <input type="text" id="username" name="username" placeholder="Enter name"
required>
  <button type="submit">Send</button>
</form>
```

Nesting: Means placing one tag inside another in the correct order.

Hierarchy: Defines parent-child relationships between elements.

Importance:

Proper nesting ensures the browser displays the layout correctly. Incorrect nesting may break design or make code unreadable.

Example (Correct):

```
<div>
  <p><strong>Hello, World!</strong></p>
</div>
```

Incorrect:

```
<p><strong>Hello</p></strong> <!-- Wrong closing order -->
```

8. a) Attributes give *extra meaning* or *control* to HTML elements. They can be **global** (e.g., id, class, style) or **specific** (e.g., src for images).

Examples:

```
 <!-- 'src' and 'alt' attributes -->
<a href="https://example.com" target="_blank">Visit</a> <!-- 'href' & 'target' -->
```

Attributes make pages interactive and user-friendly.

9. <body>

<div> // <section>

<h5> // <h1> Main Title</h1>

<p> // This page explains HTML.

</section>

</body>