1. Explain the concept of Full Stack Web Development. Discuss the key layers involved in Full Stack Development and provide examples of technologies used in each layer.

2. Explain the concept of Cascading Style Sheets and its role in web development. Compare and contrast inline, internal, and external CSS, highlighting their use cases and advantages.

3. Explain in detail Layers of Full Stack Development

4. Define WEB,URL,HTTPS,Name Resolution

5. Explain 5 types of selectors with examples

6. Write the syntax of below mentioned HTML elements and explain with an example

      (i) <div> (ii) <span>

7. Write 5 types of selectors with examples

8. Illustrate CSS box Model, Be sure to label and explain each component of the model with an example

9. Understand the concept of "distributed" in the context of version control systems like Git, and outline the different states that files can exist in within the Git repository

10. Differentiate between static and dynamic websites. Provide examples of use cases for each type and explain the technologies commonly used to build them.

11. Explain the importance of version control in software development. Describe any two basic Git commands and how GitHub enhances collaboration among developers.

12. Write the basic structure of an HTML document. Explain the purpose of the <head> tag, <body> tag and other tags, and provide examples of common HTML elements.

13. Discuss the importance of CSS style properties in web design. Provide examples of commonly used properties for text styling, box model, and layout, and explain their effects.

14. Create an HTML document for a blog post. The document should include:
    1. A proper HTML structure with <DOCTYPE>, <html>, <head>, and <body> tags.
    2. A heading (<h1>) for the blog title.
    3. A paragraph (<p>) for the blog content with at least 5 sentences.
    4. An image (<img>) related to the blog with appropriate attributes.
    5. A hyperlink (<a>) to a related article with the text "Read more".
    6. Use semantic elements like <header>, <main>, and <footer> to structure the document.

15. Design an HTML web form for a feedback submission page. The form should include:

A <form> tag with the action attribute set to "/submit".

Input fields for:

1. Name (text input, required)   2. Email (email input, required)   3. Rating (number input, range 1 to 5) 4. Feedback (text area with a character limit of 500)   5. A radio button for subscribing to the newsletter. 6. A submit button with the text "Submit Feedback". 7. Use appropriate attributes for validation (e.g., required, min, max).

16. Create a responsive webpage layout using CSS Flexbox and Grid. The layout should include:

1. A header section with a centered title.   2. A main content section divided into two columns using CSS Grid. 3. A sidebar on the left and a content area on the right, styled using CSS Flexbox. 4. A footer section with three equally spaced links. 5. Ensure the layout is responsive using media queries for different screen sizes (mobile, tablet, desktop).

17. Write HTML code to design the table as shown below. Add embedded CSS to the page such that

(i) All table headings are displayed in red font
(ii) All other cells have blue text

Sl No	Course	Subject	Marks		Category
			Internal	External	
1	B.Tech (AIML)	Game Design	30	70	T
		Game Programming	30	70	P

18. Justify your answer with proper explanation

(i) <html>
    <title> What is the color of the text </title>
    <link rel="stylesheet" href="./style.css">
    <h1 class="a-class" id="an-d"> Sixth Sem Students are in what color?</h1>
    </html>

    style.css
    .a-class{
     color:blue;
     }
    #an-id{
     color:green;

```
        }
```

(ii) <html>
 <title> What is the color of the text </title>
 <link rel="stylesheet" href="./style.css">
<h1 class="a-class another-class"> Sixth Sem Students are in what color? </h1>
 </html>

style.css
.a-class{
 color:green;
 }
 .another-class{
 color:blue;
 }
What is the color of the h1 element ?

19. Describe a scenario where you would choose to use Float, Flexbox, and CSS Grid respectively, and explain why each layout method is most appropriate for its specific scenario