➢ HTML basics theory:-

A.1:- HTML (Hyper Text Markup Language) is the standard language used to create and structure content on the web. It uses a system of tags to define elements on a webpage, such as headings, paragraphs, links, images, tables, and forms.

- Purpose of HTML in web development :- is to provide the basic structure of web pages.

- **Structures the content**: It defines where different elements like text, images, videos, and forms are placed on a webpage.
- **Enables hyperlinks**: It allows linking to other pages or resources, creating the interconnected nature of the web.
- **Is the foundation of web content**: Without HTML, a webpage wouldn't have any structure or meaning. It's essential for creating a functional website.

A:-2

- The basic structure of an HTML document includes several key elements that work together to form a complete webpage.
- Basic HTML Document Structure:-

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Page Title</title>
 </head>
<body>
 <h1>Welcome to My Website</h1>
 <p>This is a paragraph of text on my webpage.</p>
</body>
</html>
```

○ Explanation of the Mandatory Tags and Their Purpose:

- <!Doctype html>:- Purpose: Declares the document type and version of HTML being used (HTML5 in this case). It tells the browser to render the page using HTML5 standards, ensuring proper display and functionality.
- <html>:- Purpose: The root element of an HTML document. All other HTML elements are nested inside this tag. It wraps the entire content of the webpage. Attribute: You can specify the language of the document using the lang attribute, e.g., <html lang="en"> for English.
- <head>:- Purpose: Contains meta-information about the document that isn't directly visible on the page. This includes things like the title, character encoding, and links to external resources like stylesheets and scripts.
- <meta>:- Purpose: <meta charset="UTF-8">: Defines the character encoding, ensuring that text displays correctly .<meta name="viewport" content="width=device-width, initial-scale=1.0">: Helps with responsive design, ensuring the page scales properly on different screen sizes.
- <title>:- Purpose: Defines the title of the webpage, which appears in the browser’s title bar or tab. It's essential for user navigation and search engine optimization (SEO).

A:-3

- **Block-Level Elements:-**
- **Definition**: Block-level elements take up the full width of their parent container (by default) and start on a new line. They often "block" other content from appearing next to them, causing elements to stack vertically.
- These elements are typically used for larger structural parts of a webpage, such as sections, headings, paragraphs, and dividers.
- EXAMPLES:- ,<p>,<ol>,<ul>,<h1>to<h6>,<li>…..
- **Inline Elements:-**
- **Definition**: Inline elements do not start on a new line and only take up as much width as necessary for their content. They can be placed inside block-level elements and will flow along with the surrounding content.
- These elements are typically used for smaller parts of the content, such as text, links, or styling elements, that are placed inside block-level elements without disrupting the flow of the page.
- EXAMPLES:- <img>,,<a>,……

➢ HTML-Forms-theory:-

- **A:-1** HTML forms are used to collect user input and submit it to a server for processing. They allow users to interact with a webpage by entering data, which can then be sent to a backend server for things like creating accounts, sending messages, making searches, etc. Forms are an essential part of web applications, allowing users to communicate with the website.
- <input>:- This is the most versatile element in a form. It can be used for various types of input depending on its type attribute.
- Types include :-
- text: for single-line text input (e.g., name, username)
- password: for password input (it hides the text)
- email: for email addresses, often with validation
- checkbox: for yes/no or true/false values
- radio: for a selection of one option from multiple choices
- file: to upload files
- submit: to submit the form.

- **<textarea>:-** This is used for multi-line text input, like when users need to write a comment or a message .It allows users to type larger amounts of text than an <input> field. You can set its size using the rows and cols attributes, or style it with CSS.
  <select>:- This creates a dropdown menu, allowing users to select from a list of options. It contains one or more <option> elements, which represent the available choices. For example, you might use a <select> for choosing a country, a product from a list, or selecting a category.
  <button>:- This is used to trigger an action, such as submitting the form or executing JavaScript code. It can be customized with a type attribute
  submit: to submit the form data to the server.
  reset: to reset all form fields to their initial values.
  button: for general purposes like triggering custom JavaScript actions.

A:-3

- The <label> element in a form is used to define a label for an input element, such as a text field, checkbox, radio button, etc. It provides a clear, accessible description of what the user is supposed to enter in the associated input field.
- **How <label> Improves Accessibility:-**
- Better Screen Reader Support
- Clickable Areas
- Improved Focus Management

```
<form>
<label for="username">Username</label>
<input type="text" id="username" name="username">
<label for="password">Password</label>
<input type="password" id="password" name="password">
<input type="submit" value="Submit">
</form>
```

➢ HTML-Tables-theory:-

- A:-1 An HTML table is used to organize and display data in a grid-like structure with rows and columns.
- **Basic Structure of an HTML Table:-**

```
<table>
 <thead>
  <tr>
  <th>Header 1</th>
  <th>Header 2</th>
  <th>Header 3</th>
  </tr>
  </thead>
  <tbody>
  <tr>
```

```
<td>Data 1</td>
<td>Data 2</td>
<td>Data 3</td>
</tr>
 <tr>
 <td>Data 4</td>
 <td>Data 5</td>
 <td>Data 6</td>
 </tr>
 </tbody>
 </table>
```

- **Explanation of Each Element:-**

  Purpose: This element defines the start of a table. It is the container that holds all other table-related elements, including rows, headers, and data cells. Example: <table> creates a table on the webpage.

  The <table> element holds the entire table.
  <thead> is used to group the header rows, often containing
  <th> elements to define column labels.
  <tr> defines each row, which can be a header or a data row.
  <th> defines headers for columns or rows, typically in the <thead>.
  <td> holds the actual data within the rows of the table.

A:-2

- The attributes colspan and rowspan are used in HTML tables to control how many columns or rows a cell should span across, allowing for more complex table layouts. Here's how each one works and their differences:-
- **colspan:-** Purpose: The colspan attribute is used to make a table cell span across multiple columns in the same row. How it works: When you set a cell to have a certain colspan, it will expand horizontally across the number of columns specified, essentially merging multiple columns into one cell.

```
<table border="1">
<tr>
<td colspan="3"> This cell spans 3 columns </td>
</tr>
<tr>
<td>Column 1</td>
<td>Column 2</td>
```

<td>Column 3</td>

</tr>

</table>

Explanation:- In the first row, the <td> has a colspan="3", so it will span across 3 columns, merging them into one cell. The second row contains three separate <td> elements.

rowspan:- Purpose: The rowspan attribute is used to make a table cell span across multiple rows. How it works: When you set a cell to have a certain rowspan, it will expand vertically across the number of rows specified, effectively merging multiple rows into one cell.

```
<table border="1">
<tr>
<td rowspan="2">This cell spans 2 rows</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 2</td>
</tr>
</table>
```

Explanation: In the first row, the <td> has a rowspan="2", so it will span across both the first and second rows. The second column in both rows will have separate cells, as usual.

- **Key Difference:** colspan is used to span columns horizontally, while rowspan is used to span rows vertically.

```
<table border="1">
<tr>
<td colspan="2">This cell spans 2 columns</td>
<td rowspan="2">This cell spans 2 rows</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
```

Combined Example:- In this example: The first row's first cell spans 2 columns (due to colspan="2").The second row's third cell spans 2 rows (due to rowspan="2").