Mern full stack developer

- **Course Structure** **Html**

- Desing
- Coding
- Debugging
- Tools

Intro & basic
Elements
Tag
Attributes & Custom attributes
Forms (and) Validations (and) Frames
Responsive Design (and) SEO (and) etc…

What is application: An application is a software component to performed space task.

Web Application: A web application is a software component to perform data exchange over internet.

- Web application by using **web technologies** and run on the **web browser.**

Web browser: A web browser is web technologies compiler. Its compliers HTML, CSS & JS to browser understandable format and display the data.

Web Technologies:

HTML, CSS, Java Script.

- Web technologies are developed by **world wide web**.
- WWW given restriction we shouldn't create any alternative for web tech (HTML, CSS, JS).
- WWW given feasibility to develop library/frame work on the top of web.

Note: compilations are 2 types.

Compilation means converting code to human understandable.

1. Compilation – Done by compiler (Java, Net,...)
2. Transpilration –Done by transpire (Web technologies, python script, ...)

Types of web applications:

1. Web applications(internet)
2. Window applications (interanet (internal))
3. mobile applications(internet)
4. gaming applications(internet)
5. testing applications
6. networking applications
7. embedded applications.

CSS frame work:

- Bootstrap.
- Sass/ SCSS frame work
- Material UI. Etc….

JS Frame work: Angular

JS Libraries: React js, Vue js, kendo js, Jquey, etc…

Difference between programming language and Script:

JS is called programming language and it is called also scripting language.

- Programming language compiled by the complier and converts to **binary format.**
- Scripting language compiled by the transpier. Transpire translate script at run time (while program executing).

What is HTTP and HTTPS: (Hypertext transfer protocol & Secure)

HTTP: Hypertext Transfer Protocol is a protocol used for transmitting web pages and other information over the internet without encryption.

HTTPS: Hypertext Transfer Protocol Secure is the secure version of http, encryption data to safely transmit sensitive information over the internet.

3 Tier Architecture: Any software application fallows this 3 tier architecture.

- It having 3 tiers.
- Tire1 -> UI
- Tire2 -> Server
- Tire3 -> Database (DB).

Data Centre: Data centre providing space to store the information. Database, server will run on data centre.

Data Base: DB is a physical information storage.

Server: Server is a bridge between the DB and UI to exchange the data. Server is used to perform data validations.

UI: user interface display the data.

Programming language naming conventions:

- Camel Case: Starts with lower latter. Ex: firstName.
  - Variables and functions names.
- Pascal Case: Starts with Upper case latter. Ex: EmployeeDetails.
  - Class name and inter face.
- Chain Case: represented by - Ex: employee-detail
  - CSS class, folder names files names.
- Snake Case: represented by _ Ex: employee detail
  - Folder, files name.

How maney types relation between the files: there are two types.

1. Sibling files -> ./
2. Parent and child -> ../

Basics Html structure & Skeleton of Html

```
<!DOCTYPE html>

<html>

    <head>

        To add meta information (CSS, JS, Meta keywords)

    </head>

    <body>

        Display on the content.

    </body>

</html>
```

<!DOCTYPE html> Defines the document type of **version** of html.

<html></html> The **root element** of an html page.

***Structure of web application*:**

Create a folder

- Index.html -> root html / html file.
- Assets
  - css
  - Images
  - js
- Source folder – app specific files (login,..)
  - Login folder

- Login.html
- Login.js
- Login.css

Types of web applications: there are two types.

1. Static web application – HTML & CSS
2. Dynamic web application – JS (user intractable web app.)

HTML: Hyper Text Markup language.

- Purpose: Display the content in the web browser.

CSS: Cascading style sheet.

- Purpose: Beautification, beautification the content to be display

JS – JavaScript: JS adds events & actions to make user intractable web app.

React: React was developed by Facebook, to develop web & mobile app's.

- **React js**: Develop web application.
- **React Native**: Developed mobile application.

Core Components:

There are 3 core components.

1. Elements
2. Tags
3. Attributes

Elements: In the html element represented to content to be displayed on the browser. (An html element is defined by a start tag to some content and any end tag).

Ex: welcome page

Explain the above example:

- <a> this is open tag.
- this is close tag.
- Href -> this is the **attribute name**.
- “ashok.html” -> this is the **attribute value**.
- Welcome page -> this is the **content.**
- welcome page -> over all this path calling element.

Tag: tag is represented the element.

List of tags->

<html>,<head>,<title>,<script>,<style>,<link>,<body>,<h1toh6>,<hr>,<a>,<p>,
,<u>,<label>,<img>,<iframe>, ,<table>,<thead>,<tbody>,<tr>, <td>,<tfoot>,<form>,<col>,,<ol>,<ul>,<li>,<i>,<b>,<header>,<footer>, <blockquote>,<dl>,<dt>,<dd>,<pre>,<base>,<input>,<button>,<select>,<sub>, <sup>,<meta>,<section>,<nav>,<article>,<aside>,<address>,<main>,<figure>, <mark>,<figcaption>,,<small>,<s>,<city>,<q>,<dfn>,<abbr><data>,<time> ,<code>,<var>,<samp>,<kbd>,<ruby>,<rt>,<rp>,<bdi>,<embed>,<ins>,<object>, <video>,<audio>,<source>,<tract>,<map>,<area>,<svg>,<meth>,<colgroup>, <details>,<text area>,<output>,<summary>,<template>,<dialog>

This all tag represents in html tags.

Note: we can create our custom element. React works on custom element principle custome element – user defined element.

Attribute: Attribute to adding appearance and behaviour to an element.

Ex: style, id, class, onclick, alt, title, img(src), a(href).

Interview question:

1. What is application.
2. Type of application.

3. What is different between http and https.
4. What is web and web technologies.
5. How maney types of core components and this explain.
6. Difference between element and tag.

Types of elements: there are 3 types.

1. Inline element.
2. Block level element.
3. Empty element.

Inline element: there element will take width as per the same content. (inline element are those that do not start on a new line and only take up as much width as necessary)

Ex: <label> <button> <a> <img>   <sub> <sup> etc….

Block level element: there element will take end to end width of the browser. (block level element are those that start on a new line and typically occupy the full width available.)

Ex:<p> <h1-h6> <table>   <form> <header> <footer> <section> <aside> <article> <blockquote>.

Empty element: there element will not represent any content. (there element do not have an end tag or any content between tags.)

Ex: <img>
 <hr> <input> <meta> <link> <area>.

Semantic elements: semantic elements provide meaning to the web browser about the type of content that is inside them, which helps with accessibility and SEO (search engine optimization).

Ex: <footer> <header> <mani> <nav> <mark> <section> <article> <aside> <details> <figure>.

Form elements: there elements specifically handle user input and from submission functionalities.

Ex: <form> <input> <textarea> <label> <button> <select> <option> <fieldset> <legend>.

Tages full forms: <p>-paragraph <h1&h6>-headings -ordered list -unordered list -list item <a>-anchor&hyperlink -emphasis -strong emphasis
-linebreak <sub>-subscript <sup>-superscript <img>-image <hr>-horizontal rule <input>-from input <meta>-metadata <link>-link external resources like CSS files <area>-part of image maps <button>-clickable button [<optional>-an option within a<select>list] <select>-drop down list [<label>-label for an <input>element].

There 3 types are categorized into 2 types.

1. Schematic elements.
2. Non schematic elements.

Schematic elements: the element which are used for specific purpose.

Ex: Table,image, iframe, link , button, etc...

Non schematic elements: the element which are used general purpose.

Ex: p, label, span, div, etc...

Interview question:

1. How maney types of elements.
2. Types of tags
3. Can you explain self-closing tag
4. What is title tag & what is table grid tag
5. what is different between the inline element and block level element.

Heading tag: Defines the **heading** in a document ranges from <h1> to <h6> then <h1> is most important, <h6> is least and helps in structring content.

EX:

```
<body>
<h1>heading 1</h1>      // heading 1
<h2>heading 2</h2>      // heading 2
<h3>heading 3</h3>      // heading 3
<h4>heading 4</h4>      // heading 4
<h5>heading 5</h5>      // heading 5
<h6<heading 6</h6>      // heading 6
</body>
```

Paragraph tag: Used for defining paragraphs enclosed within <P>to</p> tags add automatic spacing before and after and text wraps to next line inside tag common in text-heavy content.

EX:

```
<body>
    <p> Learning to code is an invaluable skill in today's digital age. Whether
        you want to build websites, develop mobile apps, or analyze data,
            programming opens up a world of possibilities. Not only does it
            enhance problem-solving skills, but it also offers a creative
outlet to      turn ideas into reality. As technology continues to evolve, the
            demand for coding proficiency grows, making it a crucial skill
for both      personal and professional development.
    </p>
</body>
```

**
,<HR>:**

**
:** Adds a line or link break within text is empty no closing tag needed and
 and
 are both valid.

<HR>: Create a horizontal rule or line in <hr> also empty acts as a divider.

EX:

 Learning to code is an invaluable skill in today's digital age. Whether you want to build websites,
 develop mobile apps, or analyze data, programming opens up a world of possibilities. <hr> Not only does it enhance problem-solving skills, but it also offers
 a creative outlet to turn ideas into reality. <hr> As technology continues to evolve,
 the demand for coding proficiency <hr> grows, making it a crucial skill for both personal and professional development.

Images: there are three types of images

1. image
2. background image
3. button image.

Image: image is defined the **img tag.** Whene referring to file inside of another file.

Ex:

```
<body>
    <img scr=". /Your image file name">
</body>
```

Background image: background image is added to an element by using background image style.

Ex:

```
<body style=" background-image: url(". /Image url");">
    <h2>welcome to the image page </h2>
```

</body>

Button image: create a button with an image use the `<button>` element and place an `<img>` tag inside it.

EX:

```
<body>
        <button>
        <img src=”give image link” alt=” button image”>
        </button>
</body>
```

Video tag: create a video tag <video> embeds video files on page used src attribute for video URL supports multiple formats like mp4, webm allows for built-in controls via attributes like autoplay, controls, loop.

EX:

```
<body>
<video src=” video link” alt=” video play” controls></video>  </body>
```

Anchor tag: Used to creating hyperlinks requires href attribute for url can link to external sites or internal pages supports target attribute to control link behaviour.

EX:

```
<body>
<a href=”give me link”>name of link</a>

        <a href="https://www.google.co.in/">google</a>
        <a href="https://www.instagram.com/">instgram</a>
        <a href="https://www.facebook.com/">facebook</a>
```

```
        <a href="https://www.flipkart.com/">flipkart</a>

        <a href="https://www.youtube.com/">youtub</a>

    </body>
```

Some more tags: Bold/Italic/underline/strikethrough/small/big/per and also so more tags are in html.

EX:

```
<body>

    <b>this is bold tag</b>

    <i>this is italic tag</i>

    <u>this is under line tag </u>

    <strike>this is strike tag</strike>

    <s> this also strike tag</s>

    <small>this is small tag</small>

    <big>this is big tag</big>

    <per> dear sir,

    thoo.

    Thanks, you

    </per>

</body>
```

<u>Superscript and Subscript tag:</u> create <sup> makes text superscript and <sub> makes text subscript used for mathematical questions, footnotes does not change font size just position.

EX:

```
<body>
```

(a+b)2=a2+b2+2ab

(a+b)²=a²+b²+2ab

CH4+O2=>H2O+CO2

CH₄+O₂=>H₂O+CO₂

</body>

Semantic tags:	**Non-semantic tags:**
Meaningful: Describe content.	Generic: No specific meaning.
SEO: Good for search engines.	For Styling: Used for layout.
Accessibility: Useful for screen readers.	No SEO: Not SEO-friendly.
Ex: <header><footer><article><section><nav>.	Ex: <i><b><a>.