Module 5 – Mernstack – HTML5

Question 1: Difference b/w HTML & HTML5?

ans)

Feature	HTML (Old)	HTML5
Multimedia support	Via plugins	Native `<audio>`, `<video>`
Semantic tags	No	Yes (`<header>`, `<nav>`, etc.)
Mobile-friendly design	Limited	Strongly supported
Form enhancements	Limited	New input types & validation
APIs	Not available	Web Storage, Canvas, Geolocation
Doctype	Complex	Simple

Question 2: What are the additional tags used in HTML5?
ans)

HTML5 introduced many new elements (tags) to enhance the structure, semantics, and functionality of web pages. Here's a categorized list of the most important ones:

1. Semantic Structural Tags

These tags define the structure and meaning of the content (SEO-friendly & accessible):

Tag	Purpose
`<header>`	Represents introductory content (title, logo, nav, etc.)
`<nav>`	Defines a navigation section
`<section>`	Represents a standalone section
`<article>`	Independent content (e.g., blog post, news)
`<aside>`	Side content (ads, sidebars, etc.)
`<footer>`	Footer content for section or page
`<main>`	Main content area (only one per page)

2. Media Tags

These allow embedding audio and video without external plugins:

Tag	Purpose
`<audio>`	Embeds audio content
`<video>`	Embeds video content

`<source>`	Defines multiple media sources
`<track>`	Provides subtitles/captions for media

3. Form Enhancements

HTML5 improved forms with new input types and attributes:

Tag	Purpose
`<datalist>`	Provides predefined options for `<input>`
`<output>`	Displays the result of a calculation
`<progress>`	Displays progress of a task
`<meter>`	Displays scalar measurement within a known range

New Input Types: `email`, `url`, `tel`, `number`, `range`, `date`, `time`, `color`, etc.

4. Graphics & Drawing

For rendering graphics without plugins:

Tag	Purpose
`<canvas>`	Draw graphics with JavaScript
`<svg>`	Scalable Vector Graphics (already existed, but more used in HTML5)

5. Scripting and Templates

Tag	Purpose
`<template>`	Holds client-side template content that is not rendered when the page loads
`<script type="module">`	Supports JavaScript modules