Core Html

Quest.1 Build a simple webpage that displays text as shown in the below image.

This text will be bolded.

This text will be italic.

This text will be underlined

This text will be highlighted

This is normal text $^{\text{This will be super scripted}}$ This is normal again

x^2 y

H_2o

Normal Text Small Text

Quest.2 Build a simple webpage that helps users navigate different web development-related websites. Note: On clicking the hyperlink the web pages should open in a new tab. Below is a reference image.

Take me to PW skills to buy a course
Take me to MDN Web Docsto buy a course
Take me to PW SKILLS LABto buy a course

Quest.3 Build a simple blog web page with 3 pages home, web development, and web design. Each page must contain hyperlinks to other pages in the top, a heading of the page topic and a paragraph of information. For the home page you can add some information about yourself

My Simple Blog

Home | Web Development | Web Design

Web Development

Web development is the process of building websites and web applications. It involves technologies like HTML, CSS, JavaScript, and backend programming languages such as Python, PHP, and Node.js.

Question 4 : Create an ordered list of HTML tags. Each list item must include the tag name and some information about the tag.

1. **<html>** - The root element that defines an HTML document.
2. **<head>** - Contains metadata like title, links, and scripts.
3. **<title>** - Sets the title of the web page.
4. **<body>** - Contains the main content of the webpage.
5. **<h1> to <h6>** - Defines headings, with <h1> being the largest and <h6> the smallest.

6. **<p>** - Represents a paragraph of text.
7. **<a>** - Defines a hyperlink to navigate between pages.
8. **<img>** - Displays an image on the webpage.
9. **<ul> <ol> <li>** - Used to create unordered and ordered lists.
10. **<table>** - Creates a table for organizing data.
11. **<form>** - Defines a form for user input.
12. **<input>** - Allows users to input data into a form.
13. **<button>** - Represents a clickable button.
14. **<div>** - A container used to group elements for styling.
15. **<span>** - An inline container used for styling text.

Question 5 : Create a description list of full stack web development tech stack, using the <dl> tag. Each term should be a tech stack name and each description should be a brief explanation of what the tech stack is used for. the tag.

Full Stack Web Development Tech Stack

HTML (HyperText Markup Language)
Used to structure web pages with elements like headings, paragraphs, and links.

CSS (Cascading Style Sheets)
Styles web pages by controlling colors, fonts, layout, and responsiveness.

JavaScript
Makes web pages interactive with dynamic content, animations, and event handling.

React.js
A JavaScript library for building fast and dynamic user interfaces.

Node.js
Allows JavaScript to run on the server-side for backend development.

Express.js
A lightweight framework for building web applications using Node.js.

MongoDB
A NoSQL database that stores data in flexible, JSON-like documents.

MySQL
A relational database management system used for structured data storage.

Git & GitHub
Used for version control, collaboration, and code management.

Docker
A tool for containerizing applications, making them easy to deploy.

Question 6: Create an ordered list of the full stack web development tech stack HTML, CSS, and JS. For each tech stack, create a table that lists the tech stack name, its primary use cases, and some key features or benefits. Below is a reference image.

1. HTML (HyperText Markup Language)
2. CSS (Cascading Style Sheets)
3. JavaScript

HTML

Primary Use Cases	Key Features/Benefits
Building the structure of webpage	• simple and easy to learn

	• Compatible withall web browsers • Allows for sementic markup

CSS

Primary Use Cases	Key Features/Benefits
Styling and layout of web pages	• Allows for separation of content and presentation • Offers a wide range of styling options

JavaScript

Primary Use Cases	Key Features/Benefits
Adds interactivity and dynamic content to web pages	Supports event handling, DOM manipulation, and API integration

Question 7 Build a complex nested list structure representing a multi-level table of contents. Use unordered lists (<ul>) and list items (<li>) with inline-block styling to create a structured layout. Apply formatting tags to enhance the presentation of list items.

Table of Contents

- Part 1: Introduction
- Part 2: Getting Started
    - 2.1 Installing the Software
    - 2.2 Creating a New Project
        - 2.2.1 Project Templates
        - 2.2.2 Customizing Settings
    - 2.3 Exploring the Interface
        - 2.3.1 Toolbar Features
        - 2.3.2 Panel Layout
            - 2.3.2.1 Docking Panels
            - 2.3.2.2 Tabbed Interface
- Part 3: Advanced Topics
    - 3.1 Working with Plugins
        - 3.1.1 Installing Plugins
        - 3.1.2 Plugin Configuration
    - 3.2 Customizing the UI
        - 3.2.1 Changing Themes
        - 3.2.2 Configuring Shortcuts
    - 3.3 Optimizing Performance
        - 3.3.1 Caching Strategies
        - 3.3.2 Resource Minification
- Part 4: Conclusion

Question 8: 5 Create a table to display a conference schedule. Each row corresponds to a time slot, and each column corresponds to a room. Some time slots might have multiple sessions running simultaneously in different rooms. Utilize rowspan and colspan attributes as necessary to accommodate this complex schedule. (use table attribute "cellpadding" to give extra padding in each table cell).

Conference Schedule

Time	Room 1	Room 2	Room 3	Room 4
9:00 AM - 10:00 AM	Keynote	Session A	Session B	Session C
		Session D	Session E	
10:30 AM - 11:30 AM	Session F			
12:00 PM - 1:00 PM	Lunch Break			
1:00 PM - 2:00 PM	Session G	Session H	Session I	Session J
	Session K		Session L	Session M