Assignment :- HTML,CSS,SQL in PHP

HTML Basics :

1. What is HTML? Explain its structure.

   html stands for hypertext markup language. html first time released in 1986 by tim berner lee .HTML describes the structure of a webpage using tags and elements, which tell the browser how to display content like text, images, links, forms, etc.
   html is not any any language it is a markup language
   html file extension .html or .htm

   Basic structures of html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,initial-
scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1 align="center">Hey : My name is Brijesh kumar pandey</h1>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis
saepe, quod
      consectetur aliquid magni odit perferendis ut minus. Labore eos
quae dolor
      fugiat facilis eius perspiciatis recusandae nam ducimus expedita!
    </p>  </body></html>
```

   Explanation of Structure:

<!DOCTYPE html> – Tells the browser this is an HTML5 document
<html> – The main wrapper for the whole webpage
<head> – Holds info like title, styles, and settings (not shown on the page)
<title> – Sets the name shown on the browser tab
<body> – Holds everything you see on the page (text, images, etc.)

2. Describe the purpose of HTML tags and provide examples of commonly used tags.

Purpose of html tags:-

HTML tags are used to define the structure and content of a webpage.They tell the browser what kind of content is being displayed (like text, images, links) and how to display it.

Commonly used tags:-

<!DOCTYPE html> – Says this is an HTML page.
<html> – The main part that holds everything.
<head> – Has page info like title and links to CSS.
<title> – Shows the page name on the browser tab.
<body> – Has everything you see on the page.
<h1> to <h6> – Headings. <h1> is biggest, <h6> is smallest.
<p> – A paragraph.
<b> or – Makes text bold.
<i> or – Makes text italic.

 – Jumps to the next line.
<hr> – Adds a line across the page.
 – A list with bullet points.
 – A list with numbers.
 – Each item in the list.
<a> – A link to another page.
<img> – Shows an image.

<form> – A form to collect data.
<input> – A box to type or pick something.
<button> – A button you can click.
 – A box to group things.
 – A small part of text to style.
<table> – A table.
<tr> – A table row.
<th> – A table heading.
<td> – A table cell.

3. What are the differences between block-level and inline elements? Give examples of each.

Block-level Elements:-

Block-level elements start on a new line and take up the full width of the page (by default). They create a block or section on the page.

Ex. , <p>, <h1> to <h6>, <ul>, <ol>, <li>, <form>, <table>

Inline Elements :-

Inline elements stay on the same line and only use the space they need. They are usually placed inside block elements.

Ex. , <a>, , , <img>, <input>, <label>

4. Explain the concept of semantic HTML and why it is important.

Concept :-

Semantic HTML means using HTML **tags** that describe the meaning of the content inside them, rather than just its appearance.

It tells both the browser and developer what each part of the webpage is.

Ex**.**

<header> – The top part of a page, usually with a title or logo
<nav> – A section with links or menu for navigation
<main> – The main part of the page content
<section> – A part of the page with related content
<article> – A complete piece of content like a blog post or news article
<aside> – Extra information like a sidebar or tips
<footer> – The bottom part of the page, often with contact info copyright
<figure> – A box for images, charts, or diagrams

Why is Semantic HTML Important? :-

Improves readability – The code is easier for developers to read and understand.
Better SEO – Search engines can understand your content better and may rank your site higher.
Accessibility – Helps screen readers read the page properly for users with disabilities.
Clean structure – Makes the code more organized by reducing extra tags.
Future-proof – Uses standard tags, so it’s easier to update and works well in the future.

CSS Fundamentals :

1. What is CSS? How does it differ from HTML?

- ➔ CSS stands for Cascading Style Sheets.
- ➔ It is used to style and design HTML content—like colors, fonts, layouts, spacing, and more.

How it differ from HTML?

HTML:

- ➔ HTML is used to create the content of webpage.
- ➔ Example: headings, paragraphs, images, buttons.
- ➔ HTML = What you see

CSS:

- ➔ CSS is used style the content made by HTML.
- ➔ Example: colors, fonts, spacing, layout.
- ➔ CSS = How it looks.

2. Explain the three ways to apply CSS to a web page.

There is 3 ways to apply CSS :

1. Inline CSS
2. Internal CSS
3. External CSS

1. Inline CSS:

- ➔ In this method, CSS is written directly inside the HTML tag using the style attribute. It is used for quick and small styling.
- ➔ Example: This is red text.

2. Internal CSS:

➔ CSS is written inside a <style> tag within the <head> section of the HTML file. It is used to style elements on a single HTML page.
➔ Example: <head> <style> p {color: blue;}</style></head>

3. External CSS:

➔ CSS is written in a separate file with a .CSS extension. It is linked to the HTML file using the <link> tag in the <head>. This method is best for styling multiple pages.
➔ Example: <head> <link rel="stylesheet" href="style.css"></head>

3. What are CSS selectors? List and describe the different types of selectors.

➔ CSS selectors are used to select HTML elements so that we can style them using CSS.

Types of CSS Selectors:

1. Universal Selector (*)

   - Selects all elements on the page.
   - Example: * { margin: 0; }

2. Element Selector (Tag name)

   - Selects elements by their tag name like p, h1, etc.
   - Example: p { color: blue; }

3. Class Selector (.)

   - Selects elements that have a specific class.
   - Example: .title { font-size: 20px; }

4. ID Selector (#)

   - Selects an element with a specific ID.
   - Example: #main { background: yellow; }

5. Group Selector

    - Selects multiple elements at once.
    - Example: h1, p { text-align: center; }

6. Descendant Selector (space)

    - Selects elements inside another element.
    - Example: div p { color: red; }

7. Child Selector (>)

    - Selects only direct child elements.
    - Example: ul > li { list-style: none; }

8. Attribute Selector

    - Selects elements by their attribute.
    - Example: input[type="text"] { }

9. Pseudo-class Selector (:hover, etc.)

    - Selects elements in a certain state like hover or focus.
    - Example: a:hover { color: red; }

10. Pseudo-element Selector (::first-letter, etc.)

    - Selects and styles part of an element.
    - Example: p::first-letter {font-size:24px;}

4. What is the box model in CSS? Explain its components.

➔ The CSS Box Model is a fundamental concept in web design.

➔ It describes how the size and spacing of elements are calculated on a webpage.

➔ Every HTML element is treated as a rectangular box, and it consists of four layers**:**

I. Content
   - The actual content of the element (text, image, etc.).
   - Its size can be set using width and height.

II. Padding
   - The space between the content and the border.
   - It adds space *inside* the element, increasing its size.
   - Padding is transparent but takes up space.

III. Border
   - The line surrounding the padding and content.
   - Can be styled using border-width, border-style, and border-color.

IV. Margin
   - The space outside the border, between the element and others.
   - Used to create space between elements.
   - Like padding, margin is also transparent.

Responsive Web Design:

1. What is responsive web design? Why is it important?

   ➔ Responsive Web Design (RWD) means building websites that adjust automatically to different screen sizes like mobile phones, tablets, laptops, and desktops.
   ➔ Why is Responsive Web Design Important?
      a. Works on All Devices
         - Users can view your website comfortably on any device.
      b. Improves User Experience
         - No zooming or side-scrolling. Everything fits perfectly.
      c. Better SEO (Search Engine Optimization)
         - google prefers mobile-friendly websites in search results.
      d. Saves Time and Cost
         - You don't need to create a separate site for mobile and desktop.
      e. Future-Proof
         - Adapts to new screen sizes and devices easily.

2. Explain the use of media queries in CSS. Provide an example.
   ➔ Media queries in CSS are used to apply different styles depending on the device's screen size, resolution, or type. They help make websites responsive, so the layout and design adjust properly on different devices like phones, tablets, and desktops.
   ➔ They allow you to set conditions like "If the screen is smaller than 600px, use these styles." This helps create mobile-friendly designs without changing the HTML.

   Example:

   /* Default style for all devices */

   body {background-color: white; font-size: 18px;}

```
/* Style for screens smaller than 600px */
    @media (max-width: 600px) {
     body {  background-color: lightblue; font-size: 16px;}}
```

In this example:

- ➔ On large screens, the background is white and text size is 18px.
- ➔ On screens smaller than 600px (like mobiles), the background changes to light blue and font size becomes 16px.

3. What are the benefits of using a mobile-first approach in web design?

- ➔ A mobile-first approach in web design means designing the website for small screens (like mobile phones) first, then gradually adding styles for larger screens (like tablets and desktops).
- ➔ Benefits of Mobile-First Design:
    - **a.** Better performance:
        - Mobile devices often have slower internet and less power. Designing for mobile first ensures your site loads faster and runs smoothly.
    - **b.** Improved user experience:
        - Most users access websites from their phones. Mobile-first design focuses on clean, simple layouts that are easy to use on small screens.
    - **c.** Responsive from the start:
        - Starting with mobile ensures your design automatically adapts as the screen gets bigger, making your site responsive by default

d. Higher SEO ranking:
    - Google uses mobile-first indexing, meaning it checks the mobile version of your site first when ranking in search results.

e. Content-focused design:
    - You're forced to prioritize what's most important, removing clutter and focusing on essential content.

SQL:

Multiple Tables and SQL Queries:-

➔ Create multiple tables and perform queries using:
- SELECT, UPDATE, DELETE, INSERT
- WHERE, LIKE, GROUP BY, HAVING
- LIMIT, OFFSET, Subqueries, AND, OR, NOT, IN

1. Create Database

```
CREATE DATABASE company_db;
```

2. Create Tables

```
CREATE TABLE customers (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100),
   email VARCHAR(100),
   city VARCHAR(50)
);

CREATE TABLE orders (
   id INT AUTO_INCREMENT PRIMARY KEY,
   customer_id INT,
   order_date DATE,
   total_amount DECIMAL(10, 2),
   FOREIGN KEY (customer_id) REFERENCES customers(id)
);
```

3. Insert Data

```
INSERT INTO customers (name, email, city) VALUES
('Aryan Khan', aryan@example.com', 'pune'),
('Neha Patel', 'neha@example.com', 'Surat'),
('Vikram Joshi', 'vikram@example.com', 'Ahmedabad'),
```

('Anita Mehta', 'anita@example.com', 'Rajkot');

INSERT INTO orders (customer_id, order_date, total_amount) VALUES
(1, '2025-07-01', 2500.00),
(2, '2025-07-03', 3200.50),
(1, '2025-07-10', 1500.00),
(3, '2025-07-15', 4000.75),
(4, '2025-07-20', 1800.00);

4. SQL Queries with Outputs

Query: SELECT * FROM customers;

Output:

id	name	email	city
1	Aryan Khan	aryan@example.com	pune
2	Neha Patel	neha@example.com	Surat
3	Vikram Joshi	vikram@example.com	Ahmedabad
4	Anita Mehta	anita@example.com	Rajkot

Query: SELECT * FROM customers WHERE city = 'Rajkot';

Output:

id	name	email	city
4	Anita Mehta	anita@example.com	Rajkot

Query: SELECT name FROM customers WHERE name LIKE 'A%';

Output:

name
Aryan Khan
Anita Mehta

Query: UPDATE customers SET city = 'Baroda' WHERE id = 3;

Output:

id	name	email	city
1	Aryan Khan	aryan@example.com	pune
2	Neha Patel	neha@example.com	Surat
3	Vikram Joshi	vikram@example.com	Baroda
4	Anita Mehta	anita@example.com	Rajkot

Query: DELETE FROM orders WHERE id = 5;

Output:

id	customer_id	order_date	total_amount
1	1	2025-07-01	2500.00
2	2	2025-07-03	3200.50
3	1	2025-07-10	1500.00
4	3	2025-07-15	4000.75

Query: INSERT INTO orders (customer_id, order_date, total_amount)
VALUES (2, '2025-07-25', 2800.00);

Output:

id	customer_id	order_date	total_amount
1	1	2025-07-01	2500.00
2	2	2025-07-03	3200.50
3	1	2025-07-10	1500.00
4	3	2025-07-15	4000.75
6	2	2025-07-25	2800.00

Query: SELECT customer_id, COUNT(*) AS total_orders FROM orders
GROUP BY customer_id HAVING COUNT(*) > 1;

Output:

customer_id	total_orders
1	2
2	2

Query: SELECT * FROM orders LIMIT 2;

Output:

id	customer_id	order_date	total_amount
1	1	2025-07-01	2500.00
2	2	2025-07-03	3200.50

Query: SELECT * FROM orders LIMIT 2 OFFSET 2;

Output:

id	customer_id	order_date	total_amount
3	1	2025-07-10	1500.00
4	3	2025-07-15	4000.75

Query: SELECT * FROM customers WHERE city IN ('Rajkot', 'Surat');

Output:

id	name	email	city
2	Neha Patel	neha@example.com	Surat
4	Anita Mehta	anita@example.com	Rajkot

Query: SELECT * FROM customers WHERE NOT city = 'Ahmedabad';

Output:

id	name	email	city
1	Aryan Khan	aryan@example.com	pune
2	Neha Patel	neha@example.com	Surat
3	Vikram Joshi	vikram@example.com	Baroda
4	Anita Mehta	anita@example.com	Rajkot

Query: SELECT name FROM customers WHERE id = (SELECT customer_id FROM orders ORDER BY total_amount DESC LIMIT 1);

Output:

name
Vikram Joshi