DAY-1 Hands-On Activities

Problem 1: Restaurant Menu Webpage (Level-1)

Scenario

A small restaurant wants a **basic menu webpage** to display their offerings online before moving to a full website.

📌 Requirements

Create an HTML page that displays:

1. **Restaurant Name** (Heading)
2. **About the Restaurant** (Paragraph)
3. **Menu Categories** (Unordered List)
4. **Price List** (Table)

📋 Table Structure

Item Name	**Category**	**Price (₹)**
Paneer Butter Masala	Main Course	220
Veg Biryani	Main Course	180
Masala Dosa	Breakfast	90
Cold Coffee	Beverages	120

🛠️ Technical Constraints

- Use proper **HTML boilerplate**
- Use at least **5 HTML elements**
- Use **HTML attributes** such as border, title, align
- Use:
    - <table>, <tr>, <th>, <td>
    - <ul> and <li>

🎯 Learning Outcome

You should be able to:

- Build a complete HTML page structure
- Use tables for structured data
- Use lists for grouped information

Problem 2: Personal Grocery Checklist (Level-1)

Scenario

You are building a **simple webpage for personal use** to plan your weekly grocery shopping. The page should clearly show **priority items** and **optional items**, so it's easy to decide what to buy first.

📌 Requirements

Create an HTML webpage that includes:

1. A **page title**:
   Weekly Grocery Checklist
2. A **main heading** displaying the same title.
3. An **Ordered List** showing **high-priority grocery items**, such as:
   - Rice
   - Milk
   - Vegetables
   - Cooking Oil
4. An **Unordered List** showing **optional or non-essential items**, such as:
   - Snacks
   - Ice cream
   - Soft drinks

🛠️ Technical Constraints

- Use proper **HTML boilerplate**:
  - <!DOCTYPE html>
  - <html>, <head>, <body>
- Use:
  - <ol> and <ul> correctly
  - <li> for each item
- Add at least **one HTML attribute** (example: title)
- Ensure **proper indentation and readability**

🎯 Learning Outcome

You will be able to:

- Create structured content using HTML lists
- Choose the correct list type based on real-world requirements
- Understand how HTML represents **logical order and grouping**
- Build confidence in writing basic but meaningful HTML pages

Problem 3: Employee Onboarding Page (Level-2)

Scenario

A company wants a **basic onboarding page** for new employees that HR can later style using CSS.

📌 Requirements

Use Semantic HTML:

- <header> → Company name & welcome message
- <section> → Employee details
- <article> → Company policies
- <footer> → Contact information

Content Structure

1. **Employee Information (Table)**
   - Employee ID
   - Name
   - Department
   - Joining Date
2. **Company Policies (Ordered List)**
   - Working hours
   - Leave policy
   - Code of conduct
3. **Facilities Provided (Unordered List)**
   - Laptop
   - Internet access
   - Training materials

🛠️ Technical Constraints

- Use **semantic tags only** (no <div> for layout)
- Add **meaningful attributes** (title, lang, etc.)
- Proper indentation & readability

🎯 Learning Outcome

Learners should be able to:

- Explain **why semantic HTML matters**
- Differentiate between structural and non-structural tags
- Build readable, SEO-friendly HTML

Problem 4: College Department Information Page (Level-2)

Scenario

A college wants to create a **basic informational webpage** for one of its departments (e.g., Computer Science, Information Technology).
The page will be used by **students and parents** to understand faculty details, subjects offered, and the weekly timetable before the site is enhanced with CSS and backend features.

📌 Requirements

Create an HTML webpage that includes the following sections:

1. **Header**
   - Department Name
   - College Name
2. **Section 1: Faculty Details**
   - Display faculty information in a **table** with columns:
     - Faculty Name
     - Designation
     - Subject Handled
3. **Section 2: Subjects Offered**
   - Display the list of subjects using an **unordered list**

4. **Section 3: Weekly Timetable**
    - Display timetable details in a **table** with columns:
        - Day
        - Subject
        - Time Slot
5. **Footer**
    - College address
    - Contact information

🛠️ Technical Constraints

- Use proper **HTML document structure**:
    - <!DOCTYPE html>
    - <html>, <head>, <body>
- Use **semantic HTML elements**:
    - <header>, <section>, <footer>
- Use:
    - <table>, <tr>, <th>, <td>
    - <ul> and <li>
- Add meaningful **HTML attributes** such as lang or title
- Avoid CSS and JavaScript (HTML only)

🎯 Learning Outcome

You will be able to:

- Build real-world HTML pages with structured content
- Understand how semantic HTML improves readability and maintenance
- Organize information logically using tables and lists
- Prepare HTML content that is ready for CSS styling and backend integration