DAY-1 Hands-On Activities

Problem 1: **Personal Notes Saver using LocalStorage (Level-1)**

Scenario

You are building a simple web page where users can write daily notes and save them in their browser without using a server.

📌 Requirements

- A textarea for writing notes.
- A **Save** button (using onclick).
- A **Clear** button.
- Notes must:

    Be stored in localStorage

    Automatically load when the page refreshes

- Display stored note on page load.

🛠 Technical Constraints

- Must use:
    - onclick inline event
    - localStorage.setItem()
    - localStorage.getItem()
    - localStorage.removeItem()
- No backend/database.
- Pure HTML + JavaScript only.
- Data stored as key-value pair.

🎯 Learning Outcome

You should be able to:

- Inline event handling (onclick)
- Browser Storage APIs
- Storing & retrieving key-value data
- Page load event handling
- Basic DOM manipulation

Problem 2: Live Form Validation with Events (Level-1)

Scenario

Create a simple registration form that validates user input when fields change.

📌 Requirements

- Fields:
    - Name
    - Email
    - Age
- Use:
    - onchange
    - onclick
- Validate:
    - Name cannot be empty
    - Email must contain "@"
    - Age must be greater than 18
- Display validation message dynamically.
- Store valid user data in sessionStorage.

🛠 Technical Constraints

- Must use inline onchange events.
- Store valid data using:
    - sessionStorage.setItem()
- No external libraries.
- Use basic JavaScript only.

🎯 Learning Outcome

You will be able to:

- onchange event usage
- Form validation logic
- Difference between localStorage and sessionStorage
- Dynamic DOM updates

Problem 3: Location-Based Weather Logger (Level-2)

Scenario
Create a web application that fetches the user's geographic location and stores location history in localStorage.

📌 Requirements

- Button: **Get My Location**
- Use:
  navigator.geolocation.getCurrentPosition()
- Display:
  Latitude
  Longitude
- Handle:
  Permission denied
  Timeout
  Location unavailable
- Save last 5 location entries in localStorage.
- Display location history on page load.

🛠 Technical Constraints

- Must handle:
  Success callback
  Error callback
- Use browser permission handling.
- Store data as JSON using:
  JSON.stringify()
  JSON.parse()
- Use inline event (onclick).

🎯 Learning Outcome

Learners should be able to:

- Geolocation API
- Handling browser permissions
- Error handling in APIs
- Managing structured data in localStorage
- JSON parsing and stringifying

Problem 4: Mini Expense Tracker using Client-Side Database (Level-2)

Scenario

Develop a client-side expense tracker where users can add, view, and delete expenses using a browser-supported database.

📌 Requirements

1. Fields:
    - Expense Title
    - Amount
    - Date
2. Buttons:
    - Add Expense
    - View Expenses
    - Delete Expense
3. Use:
    - Client-side database (Web SQL or IndexedDB)
4. Execute SQL-like queries:
    - CREATE TABLE
    - INSERT
    - SELECT
    - DELETE
5. Maintain transaction handling.
6. Display expense list dynamically.

🛠 Technical Constraints

- Must use:
    - Client-side DB transactions
    - SQL execution methods
- Must handle:
    - Transaction errors
    - Query errors
- No server/database allowed.
- Pure JavaScript + HTML.

🎯 Learning Outcome

You will be able to:

- Client-side database concepts
- Executing SQL queries in browser
- Managing transactions
- Advanced DOM rendering
- Persistent structured storage