# Ex. No 4: Build a web client-side Login, Registration form and Dashboard with drop down menus.

**Aim:**

To build a web client-side Login, Registration form and Dashboard with drop down menus.

# Background Theory:

1. **HTML Forms and Structure in Basic HTML**

Forms in HTML are used to collect input data from users, such as login credentials or registration details. Basic HTML forms do not have the enhanced validation and input types of HTML5, so we rely on the most basic tags to structure the form.

# Key Elements:

* + **<form>**: The container for all form input elements.
    - **action**: Specifies the URL where form data is sent.
    - **method**: Defines how form data is sent, commonly set to "post" for secure data transfer.
  + **<label>**: Provides a textual description of each input field, helping users understand the purpose of each field.
  + **<input>**: Defines a field where users can enter information. Common type attributes in basic HTML include:
    - "text": Used for single-line text fields.
    - "password": Masks input characters, useful for password fields.
    - "submit": Creates a button to submit the form.
  + **Basic Form Structure**: Each form field should have an associated label to improve accessibility and clarity for users.

# Building the Login Form

A login form in basic HTML consists of fields for the username and password. The form is structured without additional validation, so checking for correct credentials would be handled on the server-side after submission.

 **Username and Password Fields**: Both fields are simple <input> elements without validation. type="text" shows the password input.

**Submit Button**: The form includes a <input type="submit"> button to send data to the specified action endpoint.

# Building the Registration Form

A registration form generally collects more information. In basic HTML, additional fields are included, but without automatic email or password validation, as would be possible in HTML5.

**Text Fields (<input type="text">)**: This can be used to collect basic information like username, email, and password.

**Radio Buttons**: The name attribute for all gender options is the same (gender) to ensure only one option can be selected at a time.

**Checkboxes**: Each checkbox has the same name (e.g., hobbies) but different values to allow multiple selections.

**Dropdown Menu**: The <select> element provides a dropdown list, and each <option> represents e.g., a country.

**Submit Button**: An <input type="submit"> button allows users to submit their registration data to the specified action URL.

# Creating the Dashboard with Dropdown Menus

Without CSS or JavaScript, dropdown menus are created using the <select> tag, which allows users to choose from predefined options. This approach is not dynamic but provides a simple way to simulate dropdowns.

* + **Dropdown Menu Structure**: Each dropdown is created with a <select> element containing multiple <option> elements. This lets users pick one of several choices in a menu.
  + **Options (<option>)**: Each <option> element represents an item in the dropdown menu. The value attribute specifies the data sent when an option is selected.

# Result:

Thus a web client-side Login, Registration form and Dashboard with drop down menus is created using HTML.