

Topic 5 — HTML Forms : By Gagan Baghel

Concepts Covered:

`<form>`, `<label>`, `<input>`, `<textarea>`, `<select>`, `<option>`, `<button>`, `name`, `value`, `placeholder`, `required`, `checked`, `disabled`, and form organization with `<fieldset>` and `<legend>`.

🎯 Learning Objective:

To learn how to build and organize user input forms using various input elements, understand their attributes, and apply them in real-world use cases like signup, feedback, and contact forms.

🧩 Practice Tasks (10 Total)

Task 1: Create a Simple Contact Form

Goal: Learn basic form structure and text inputs.

Instructions:

1. Create `contact-form.html`.
2. Add a `<form>` with these fields:
 - Name (`<input type="text">`)
 - Email (`<input type="email">`)
 - Message (`<textarea>`)
3. Add a submit button (`<input type="submit" value="Send Message">`).
4. Use `<label>` for each field.
5. Add `placeholder` and `required` attributes.
6. output :

Name:

Email:

Message:

Send Message

Task 2: Create a “Sign Up” Form

Goal: Use multiple input types together.

Instructions:

1. Create `signup-form.html`.
2. Add:
 - Username
 - Password (`type="password"`)
 - Confirm Password
 - Email
 - Gender (Radio buttons)
 - Country (Dropdown using `<select>` and `<option>`)
 - Submit button
3. Add proper `<label>`s and use the `name` attribute for each input.
4. Mark required fields using `required`.
5. Output :

Username:

Password:

Confirm Password:

Email:

Gender:

Male

Female

Country:

Task 3: Create a Feedback Form

Goal: Learn radio buttons and checkboxes.

Instructions:

1. Create `feedback.html`.
2. Add:
 - Full Name
 - Email
 - Feedback Type (Radio buttons: Good, Average, Poor)
 - Features liked (Checkboxes: Design, Speed, Content, Usability)
 - Comment box (`<textarea>`)
 - Submit button
3. Group related fields using `<fieldset>` and `<legend>`.
4. Output :

Feedback Information

Full Name:

Email:

Feedback Type:
 Good Average Poor

Features Liked:
 Design Speed Content Usability

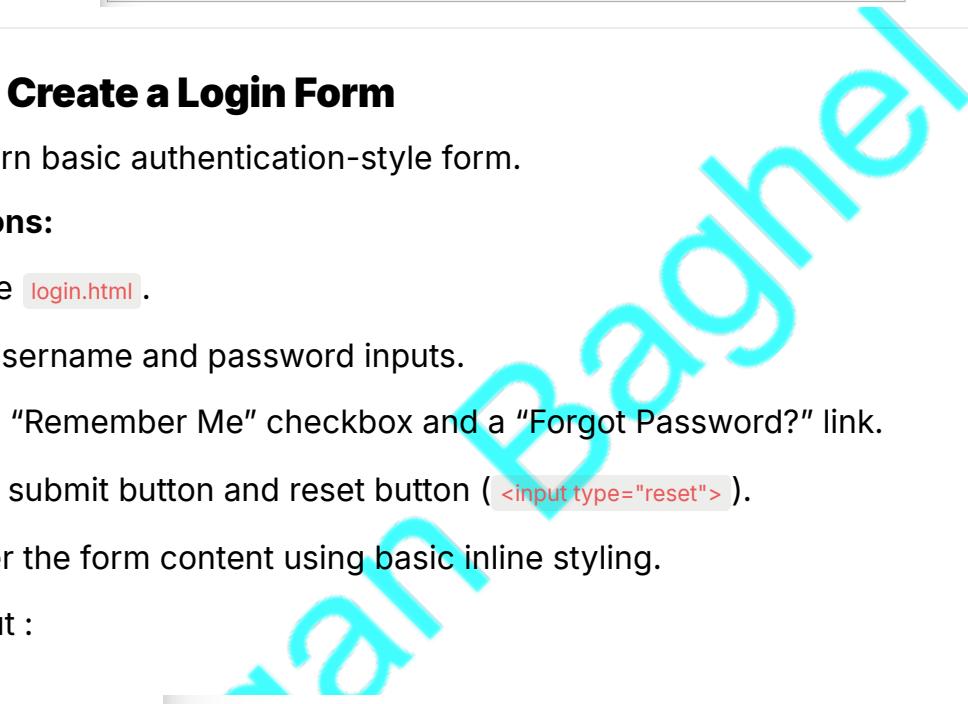
Comments:

Task 4: Create a Login Form

Goal: Learn basic authentication-style form.

Instructions:

1. Create `login.html`.
2. Add username and password inputs.
3. Add a "Remember Me" checkbox and a "Forgot Password?" link.
4. Use a submit button and reset button (`<input type="reset">`).
5. Center the form content using basic inline styling.
6. Output :



Username:

Password:

Remember Me

[Forgot Password?](#)

Task 5: Create a Job Application Form

Goal: Use advanced input attributes and grouping.

Instructions:

1. Create `job-form.html`.
2. Add:
 - Full Name
 - Email
 - Contact Number
 - Position Applying For (Dropdown)
 - Upload Resume (`<input type="file">`)
 - Availability Date (`<input type="date">`)
 - Experience Level (Radio buttons)
 - Submit button
3. Add `<fieldset>` and `<legend>` to organize sections (e.g., Personal Info, Job Details).
4. Test input types in your browser.
5. Output :

Personal Info

Full Name:

Email:

Contact Number:

Job Details

Position Applying For:

Upload Resume: No file chosen

Availability Date:

Experience Level:

Junior Mid-Level Senior

Task 6: Create a "Survey Form"

Goal: Combine all input types into one project.

Instructions:

1. Create [survey-form.html](#).
2. Add:
 - Name
 - Age (number input)
 - Favorite Color (color picker)
 - Rating (1–10 using number or range input)
 - Hobbies (checkboxes)
 - Country (dropdown)
 - Comments box (`<textarea>`)
3. Add a Submit and Reset button at the end.

4. Test all inputs by filling and submitting.

5. Output :

Name:

Age:

Favorite Color:

Rating (1–10):

Hobbies:

Reading Music Travel

Country:

Comments:



Task 7: Create a Course Enrollment Form

Goal: Practice dropdowns, radio buttons, and multiple selects.

Instructions:

1. Create `course-enroll.html`.

2. Add:

- Full Name
- Email
- Select Course (Dropdown with 4–5 options)

- Select Timing (Radio buttons: Morning, Afternoon, Evening)
 - Additional Modules (Multiple select dropdown)
 - Comments section
 - Submit button
3. Use `<label>` and `for` attributes for accessibility.
4. Use `multiple` attribute for multiple selections.

5. Output :



Full Name:

Email:

Select Course:

Select Timing:

Morning Afternoon Evening

Additional Modules:

Git & GitHub
 Node.js
 MongoDB
 Deployment

Comments:

Task 8: Create a Subscription Form

Goal: Learn inline forms and button types.

Instructions:

1. Create `subscription.html`.
2. Add:
 - Email field (`type="email"`)
 - Subscribe button (`<button type="submit">Subscribe</button>`)
3. Add a note: "We respect your privacy."
4. Output :

```
<form>
  <label>Email:</label> <input type="email" value="Enter your email"/> <button type="submit">Subscribe</button>
  We respect your privacy.
</form>
```

Task 9: Create a "Profile Settings" Form

Goal: Explore disabled, readonly, and prefilled values.

Instructions:

1. Create `profile-settings.html`.
2. Add:
 - Username (readonly)
 - Email (editable)
 - Phone number
 - Date of Birth
 - Gender (radio)
 - Newsletter Subscription (checkbox, pre-checked using `checked`)
3. Add Save and Reset buttons.
4. Disable one field intentionally using `disabled` attribute.
5. Output :

Username:

Email:

Phone:

Date of Birth:

Gender:

Male Female

Subscribe to Newsletter

Bio (Disabled Field):

Task 10: Create a Complete "Contact Us" Page

Goal: Combine design, structure, and input features.

Instructions:

1. Create `contact-us.html`.
2. Add:
 - Name, Email, Subject, Message
 - Reason for contact (Dropdown)
 - Checkbox for "Subscribe to updates"
 - Submit and Reset buttons
3. Group the form visually with `<fieldset>` and `<legend>`.
4. Add proper placeholder text and alt labels.
5. Add a small footer note — "We'll respond within 24 hours."

6. Output :

Contact Us

Name:

Email:

Subject:

Message:

Reason for Contact:

Subscribe to updates

We'll respond within 24 hours.



Pro Tips for Students

- Always wrap related inputs inside a `<form>` tag — otherwise, submission won't work.
- Use `<label>` with `for` attributes — improves accessibility and usability.
- Use `name` and `value` properly — these are essential for backend form processing.
- Prefer `<button>` over `<input type="submit">` when adding icons or text inside.
- Use `required` and `placeholder` for better UX.
- Use `<fieldset>` and `<legend>` to group related fields — gives a professional structure.
- Never leave form elements unlabelled — screen readers depend on labels.