SIGNUP

1. HTML Structure

* <!DOCTYPE html>: Declares this as an HTML5 document.
* <html lang="en">: Sets the document language to English.

2. <head> Section

* External CSS:
  + signup-style.css: Custom styling for the signup form.
  + Font Awesome CDN: Provides icons (Instagram, Google, Facebook).

3. Navigation Menu (<ul>)

* Topics: HTML List + Button + Navigation
* Contains menu items: Home, About, Language, Cart, Contact.
* Use: Creates a simple horizontal navigation bar with buttons.

4. Background Video

* <video src="./vedioa.mp4" muted autoplay loop>:
  + Topics: HTML Media Element
  + Use: Adds an autoplaying, muted, looping video background to the page.

5. Signup Form (<form>)

Input Fields:

* Name: <input type="text">
* Email: <input type="email">
* Password: <input type="password">

Error Spans:

* <span id="name\_error">, etc.
* Topics: Form Validation UI
* Use: Show validation errors next to each input.

Submit Button:

* <button type="submit">SIGN-UP</button>
* <fieldset> groups and outlines form elements.
* <legend> adds a title to the group.

6. Social Media Sign-Up Options

* Font Awesome Icons with Links:
  + Instagram
  + Google
  + Facebook
* Topics: Icon Integration + External Linking
* Use: Redirects users to social login pages.

7. Navigation to Login

* <p>Already have an account? <a href="...">Login</a></p>
* Use: Provides a link to login page for existing users.

8. JavaScript Integration

* <script src="./signup-script.js"></script>
* Topics: Form Validation / Interactivity
* Use: Handles form validation (like checking empty fields, valid email, etc.).

Css Summery

1. Navigation Bar

* Topic: Flexbox Layout
* Purpose: Creates a horizontal menu bar with items like HOME, ABOUT, LANGUAGE, etc.
* Benefit: Helps users quickly navigate different sections of your site.

2. Background & Media

* Topic: Background Images and Videos
* Purpose: A background image and a sticky video element are used to enhance the visual design.
* Benefit: Makes your website look modern and interactive.

3. Signup Form

* Topic: HTML Forms
* Purpose: Collects user data (name, email, password) for signing up.
* Benefit: Essential for user authentication and account creation.

4. Input Fields and Labels

* Topic: Form Elements Styling
* Purpose: Customized styling for inputs and labels using borders, padding, colors, and spacing.
* Benefit: Improves readability and user experience.

5. Form Validation Feedback

* Topic: Error Message Styling
* Purpose: Displays error messages when users submit incomplete or incorrect data.
* Benefit: Guides users to correct mistakes before submitting.

6. Social Media Integration

* Topic: Font Awesome Icons
* Purpose: Adds login links with icons for Instagram, Google, and Facebook.
* Benefit: Allows users to sign up using social accounts.

7. Responsive Design Elements

* Topic: Percentage-based Widths & Flexbox
* Purpose: Layout adapts to different screen sizes to some extent.
* Benefit: Enhances accessibility on various devices.

8. CSS Transitions (Hover Effects)

* Topic: Hover Styling
* Purpose: Buttons change appearance on mouse hover.
* Benefit: Adds interactivity and feedback to user actions.

9. Visual Separation Using Fieldset

* Topic: Visual Structure
* Purpose: Adds a line with “or” text between the form and social icons.
* Benefit: Clearly separates different login/signup options.

10. JavaScript (Expected Features in Script)

* Topics:
  + Accessing HTML elements (getElementById)
  + Listening to form submission (addEventListener)
  + Validating input data
  + Showing error messages dynamically
* Purpose: Makes the form interactive and prevents empty or invalid form submissions.
* Benefit: Improves usability and prevents incorrect data entry.

JAVASCRIPT SUMMERY

### Topics and Their Purpose:

#### 1. **Getting HTML Elements**

* You connect JavaScript with the form fields (like name, email, password) and error message areas.
* This helps your script check what's entered in each field.

#### 2. **Event Listener**

* You add an action that says:  
  **“When the form is submitted (button clicked), run this function.”**
* It checks if the values are valid or not.

#### 3. **Preventing Default Submit**

* Normally, a form will refresh the page when submitted.
* If the data is **invalid**, this behavior is stopped to show error messages instead.

#### 4. **Name Field Validation**

* It checks if the name field is **empty**.
* If it's empty, it shows an error message under the input field.

#### 5. **Email Validation (Using Pattern)**

* It checks if the entered email looks like a real one (example: user@gmail.com).
* If it doesn’t match the expected pattern, it shows an error like “Valid email is required”.

#### 6. **Password Validation (Strong Password)**

* Checks whether the password is:
  + 8–15 characters long
  + Contains **uppercase**, **lowercase**, **numbers**, and **special characters** (like @, #, etc.)
* If it fails, it shows an error saying the password must be valid.

#### 7. **Error Message Display**

* If a field is wrong, an error message is shown below that field.
* If it's correct, any old error message is cleared.

Login Page

## **Login Page Overview**

This HTML file creates a **login page interface** using:

* HTML for structure
* External CSS for design
* Font Awesome for icons
* JavaScript (linked externally) for validation or interactivity

## **Topics Covered and Their Purpose**

### 1. **Header Links**

* **External CSS link**: Loads your custom styles for design (colors, layout).
* **Font Awesome link**: Provides social media icons like Instagram, Google, and Facebook.

### 2. **Navigation Bar**

* A horizontal menu bar with buttons like:
  + **HOME**, **ABOUT**, **LANGUAGE**, **CART**, **CONTACT**
* Used to **navigate** the website easily.

### 3. **Video Background**

* A video plays in the background with muted autoplay.
* Adds a modern, interactive feel to your page.

### 4. **Login Form Section**

* Centered form box where users can log in using their email/phone and password.

#### Contains:

* **Email Input**: Where users enter their login email or mobile.
* **Password Input**: For entering secure password.
* **Submit Button**: Triggers the login process.
* **Error Messages**: Empty spans to display validation errors from JavaScript.

### 5. **Visual Separation**

* A horizontal line with the word **“or”** to separate login from social media options.

### 6. **Social Media Login Icons**

* Icons linked to:
  + Instagram login
  + Google login
  + Facebook login
* These can be used for **social login** (if implemented later).

### 7. **Additional Links**

* **“Don't have an account?”** → Redirects to Signup page.
* **“Forgot Password?”** → Placeholder link for future password recovery page.

### 8. **JavaScript Integration**

* External JavaScript file (login-script.js) is included at the end.
* This script is responsible for:
  + Validating the input fields
  + Showing error messages
  + Preventing form submission if validation fails

CSS SUMMERY

1. Global Reset

* All elements start with zero margin and padding.
* Ensures consistent layout across different browsers.

2. Body Styling

* Background image used (yellow-themed).
* It covers the entire top portion (800px height).
* Background does not repeat and is fully stretched across width.

3. Video Styling

* Positioned to the left side.
* Has a fixed size (90% width, 500px height).
* Uses sticky positioning to keep it in view.
* Slightly shifted left using negative margin.

4. Navigation Bar (#main)

* A horizontal flex layout is applied.
* No list bullet points (unstyled <ul>).
* Buttons are spaced with margin and use monospace font.
* Hover effect adds a visible red border around buttons.

5. Form Box (#fo)

* Placed to the left beside the video.
* Set to a fixed size (30% width, 500px height).
* Has:
  + Rounded corners
  + Black border
  + Box shadow for 3D effect

6. Input Fields

* Email Input (#login-em)
  + Top spacing, left padding, and clear red border.
* Password Input (#login-ps)
  + Slightly lower than email field.
  + Same transparent background and border style.
* Rounded edges and modern appearance.
* Maintains red theme consistency.

7. Labels and Headings

* Labels are aligned using margin-left.
* Heading (LOGIN)
  + Centered
  + Large font size (50px)
  + Bright red color to match overall theme

8. Login Button (#lo)

* Styled similar to input fields.
* Transparent but bordered with red.
* Positioned further right using margin.

9. Fieldset and Legend

* #fi removes borders except the top, giving a cleaner line.
* The word “or” in #or appears in the middle, styled in red.

10. Social Media Icons

* Icons are spaced apart and aligned in a row.
* All icons are red-colored to match theme.
* Each icon (Instagram, Google, Facebook) has its own margin alignment.

11. Additional Links

* "Don't have an account?" is styled with padding.
* "Forgot password?" is aligned and styled with:
  + Black text
  + No underline (clean look)

12. Error Messages

* Error spans (#email\_error, #pass\_error) appear in red below fields.
* Positioned using margin and padding so they don’t overlap the layout.

JAVASCRIPT SUMMERY

Login Form Validation Summary

Purpose:

This script validates email and password fields when the user submits the login form. If anything is invalid, it prevents the form from submitting and displays error messages.

1. Selects Elements:
   * Gets references to the email input, password input, form, and the two span areas where error messages will appear.
2. Listens to Form Submission:
   * Watches when the user clicks the "LOGIN" button or submits the form.
3. Checks Email Format:
   * Uses a regular expression (pattern) to make sure the email follows correct format like example@gmail.com.
   * If it's invalid, it:
     + Stops the form from submitting.
     + Shows the error: "valid email is required".
   * If it's valid, it removes the error message.
4. Checks Password Format:
   * Password must meet these rules:
     + At least one lowercase letter
     + At least one uppercase letter
     + At least one digit (0–9)
     + At least one special character (like @ . # $ ! % \* ? &)
     + Length should be 8 to 15 characters
   * If password is invalid, it:
     + Stops form submission
     + Shows: "Password is invalid"
   * If valid, it clears the error.

Example of a Valid Input:

* Email: raju123@gmail.com
* Password: Raju@1234

Forget Password Page

Page Purpose:

This is a "Forgot Password / Get OTP" page. It allows a user to enter their email address and request a One-Time Password (OTP) to reset their password.

Key Elements on the Page:

1. Email Label and Input Field
   * The label tells the user to enter their email.
   * The input box is where the user types their email address.
2. Error Message Placeholder
   * There's a space below the input box where an error message can be shown if the email is invalid (like missing "@", etc.).
3. Submit Button
   * The button says “GET AN OTP.”
   * When clicked, it will trigger a JavaScript function (in the external forget.js file) to check if the email is correct and, if valid, proceed to send an OTP.
4. Form Container
   * The entire form is wrapped inside a container (div) to help with layout and design (handled using CSS from the forget.css file).

What Happens When You Submit:

* The page checks if the email is entered correctly.
* If not, it shows an error message without reloading the page.
* If valid, it proceeds to the next step (like sending OTP or redirecting to another page).

**Explanation** of CSS file:

### Purpose:

This CSS styles your **“Forgot Password / OTP”** form and gives it a colorful, visually appealing look with gradients, rounded corners, shadows, and layout adjustments.

### 🔍 Explanation of Each Section:

#### 1. **Global Styling**

* Removes default spacing (margin, padding) from all elements.
* box-sizing: border-box ensures that padding and borders are included in an element’s total size (avoids layout issues).

#### 2. **Body Styling**

* Full screen height (100vh) is used.
* A **beautiful gradient background** is applied (diagonal blend of orange, pink, and purple colors).

#### 3. .main **(Form Container)**

* Acts as the box that holds the form.
* Width: 300px, Height: 200px — a small card-like size.
* Background: Another gradient (red to green).
* Rounded corners (border-radius: 20px) give it a soft, smooth edge.
* Centered horizontally (margin-left: 600px) and pushed down vertically (margin-top: 250px).
* box-shadow: Adds a 3D floating effect (black shadow).

#### 4. **Label (**#email**)**

* The "Email" text is moved to the right for alignment.
* Font size is made large for visibility.

#### 5. **Input Box (**#in**)**

* Rounded edges for smooth design.
* No border (clean look).
* Adjusted margin to align neatly with the label.

#### 6. **Submit Button (**#btn**)**

* Positioned in the center horizontally.
* Green background with white text (eye-catching).
* Padding inside the button for better size.
* Rounded edges and no border (modern look).
* **Box-shadow** gives a 3D effect, making the button pop out.

#### 7. **Error Message Area (**#email\_error**)**

* Text color is dark (black).
* Spaced and padded to show clearly below the input if an error occurs.

Js File

Purpose:

This script checks whether the email entered by the user in the "Forgot Password" form is valid before allowing the form to submit.

Step-by-Step Explanation:

1. Getting the Email Input:
   * The script targets the input box where the user types their email.
2. Getting the Form:
   * It accesses the entire form so it can prevent the default behavior (submitting the form) if the email is not valid.
3. Getting the Error Display Area:
   * It uses a <span> to show error messages if needed (below the email input).
4. When the Form is Submitted:
   * A function is triggered to check if the email is written properly.
5. Email Pattern Check:
   * It uses a regular expression (a pattern rule) to verify that the email follows this format:  
     example@domain.com
6. If Email is Invalid:
   * It stops the form from submitting (preventDefault()).
   * It shows a message: "valid email is required" in red (from CSS).
7. If Email is Valid:
   * The error message is cleared.
   * The form continues submitting (e.g., may go to server to send OTP).

Summary:

This is a simple email validation check that:

* Stops submission if email is invalid.
* Shows an error message to guide the user.
* Ensures only valid email formats go through.