

Progetto

Specifiche tecniche

- Il sito web deve essere realizzato con lo standard HTML5. Le pagine devono degradare in modo elegante e devono rispettare la sintassi XML;
- il layout deve essere realizzato con CSS puri (CSS2 o CSS3);
- L'uso dei layout Flex e Grid, se sviluppati in maniera corretta ed utilizzati ragionevolmente, vengono valutati molto positivamente;
- il sito web deve rispettare la completa separazione tra contenuto, presentazione e comportamento;
- il sito web deve essere accessibile a tutte le categorie di utenti;
- il sito web deve organizzare i propri contenuti in modo da poter essere facilmente reperiti da qualsiasi utente;
- il sito web deve contenere pagine che utilizzino script PHP per collezionare e pubblicare dati inseriti dagli utenti (deve essere sviluppata anche la possibilità di modifica e cancellazione dei dati stessi);
- tra gli input richiesti all'utente deve esserci almeno un campo di testo libero;
- deve essere presente una forma di controllo dell'input inserito dall'utente, sia lato client che lato server;
- i dati inseriti dagli utenti devono essere salvati in un database;
- è preferibile che il database sia in forma normale.

1. Struttura del Documento e Semantica HTML

Questa sezione riguarda le fondamenta del codice e come viene interpretato dai browser e dagli screen reader.

- Validazione W3C:** Il codice deve essere privo di errori sintattici (tag non chiusi, attributi errati). Usa il *W3C Markup Validation Service*.
- Doctype e Lingua:** Dichiarare `<!DOCTYPE html>` e l'attributo `lang` nel tag `<html>` (es. `<html lang="it">`). Segnalare cambi di lingua interni con l'attributo `lang` sui tag specifici.
- Tag Semanticci HTML5:** Utilizzare `<header>`, `<nav>`, `<main>`, `<article>`, `<aside>`, `<footer>` invece di `<div>` generici.
- Contenuto Principale (`<main>`):** Deve essere unico e visibile per pagina.
- Gerarchia Intestazioni (H1-H6):** Ordine logico e sequenziale (non saltare da H1 a H3). L'H1 deve essere unico per pagina.

Paragrafi e Liste: Usa `<p>` per il testo (non usare `
` multipli per distanziare). Usa `` e `` per gli elenchi, mai semplici ritorni a capo.

Attributi Data: Nominati correttamente (es. `data-nome`) senza maiuscole dopo il prefisso.

2. Navigazione e Orientamento

Garantire che l'utente possa muoversi nel sito con facilità, specialmente tramite tastiera.

Skip Links: Inserire un link "Salta al contenuto" all'inizio della pagina, visibile solo al focus del tasto TAB.

Ordine del Focus: L'ordine di tabulazione nel codice deve corrispondere all'ordine visivo della pagina.

Focus Visible: L'elemento selezionato tramite tastiera deve avere un indicatore visivo chiaro (es. bordo colorato).

Breadcrumbs (Briciole di pane): Presenti per facilitare l'orientamento in strutture profonde.

Meta Tag: `<title>` unico, breve (max 60 car.) e descrittivo. Meta description presente e chiara.

3. Elementi Interattivi (Link, Bottoni e Form)

Assicurarsi che ogni interazione sia comprensibile e utilizzabile da tutti.

Chiarezza dei Link: Testi comprensibili fuori contesto (evitare "clicca qui"). Specificare se aprono file (es. "Scarica PDF, 2MB") o portano a siti esterni.

Etichette Form (Labels): Ogni `input` deve avere una `<label>` associata tramite `for` e `id`. I placeholder non bastano.

Campi Obbligatori ed Errori: Segnalati con testo o `aria-required`, non solo con il colore. I messaggi di errore devono suggerire come correggere l'input.

Accessibilità Bottoni: Usare tag `<button>`. Verificare che siano attivabili tramite la barra spaziatrice.

Gestori di Eventi JS: Tutte le funzioni (es. menu a tendina, modali) devono essere attivabili sia via mouse che via tastiera.

4. Contenuti Non Testuali e Media

Fornire alternative per chi non può vedere o sentire i contenuti.

Testo Alternativo (Alt Text): Presente e descrittivo per immagini informative.
`alt=""` (vuoto) per immagini puramente decorative.

5. Design, Colore e Font

Aspetti visivi e leggibilità del testo.

Contrasto Cromatico: Rapporto minimo 4.5:1 per testo normale e 3:1 per testo grande.

Indipendenza dal Colore: Non usare il solo colore per dare informazioni (es. "i campi in rosso sono obbligatori"). Usare icone o testo aggiuntivo.

Layout Fluido (Responsive): Nessun scorrimento orizzontale fino a un certo livello di zoom; il layout deve adattarsi ai vari schermi.

Caratteristiche del Font:

- Evitare font decorativi o troppo sottili.
- Altezza della "x" elevata.
- Aperture ampie (lettere c, a, e).
- Tracking (spazio tra lettere) adeguato e legature non forzate.

Strumenti Consigliati per il tuo Audit

1. **Validazione:** W3C Markup & CSS Validator.

2. **Screen Reader:** NVDA (Windows) o VoiceOver (Mac/iOS).
3. **Contrasto:** Colour Contrast Analyzer, ColorSafe.
4. **Simulazione:** Color Oracle (per daltonismo).

Attributo data: Il nome dell'attributo deve iniziare con data- non contenere alcuna lettera maiuscola, e almeno un carattere oltre il prefisso. Esempio:

```
<input type="password" name="pin"  
       required="required" aria-required="true"  
       data-msg-required="Per favore, inserisci la tua  
                         password"  
       data-msg-invalid="Formato non corretto" >
```