



TECNOLOGIE WEB  
A.A. 2024/2025

# Leggende Animate

Relazione su progettazione, realizzazione e test

---

<b>Referente</b>	Bergamin Elia - 2068230 - elia.bergamin.1@studenti.unipd.it
<b>Componenti</b>	Bergamin Elia - 2068230 - elia.bergamin.1@studenti.unipd.it Corradin Samuele - 2068235 - samuele.corradin@studenti.unipd.it Milan Riccardo - 2068231 - riccardo.milan.4@studenti.unipd.it Peruzzi Uncas - 2068239 - uncas.peruzzi@studenti.unipd.it
<b>Indirizzo web</b>	<a href="http://tecweb.studenti.math.unipd.it/ebergami">http://tecweb.studenti.math.unipd.it/ebergami</a>
<b>Utenti</b>	Username: <i>user</i> , Password: <i>user</i> Username: <i>admin</i> , Password: <i>admin</i>

# Indice

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Analisi</b>	<b>2</b>
2.1	Caratteristiche dell'utenza . . . . .	2
2.2	Funzionalità . . . . .	2
<b>3</b>	<b>Progettazione</b>	<b>3</b>
3.1	Struttura del sito . . . . .	3
3.2	Pagine di errore . . . . .	3
<b>4</b>	<b>Realizzazione</b>	<b>3</b>
4.1	HTML . . . . .	3
4.2	CSS . . . . .	4
4.3	Validazione dell'input . . . . .	4
4.4	JavaScript . . . . .	4
<b>5</b>	<b>Database</b>	<b>5</b>
5.1	PHP . . . . .	5
5.2	Elementi dell'interfaccia utente . . . . .	6
5.2.1	Immagini e icone . . . . .	6
5.2.2	Font . . . . .	6
5.2.3	Colori . . . . .	6
<b>6</b>	<b>Accessibilità</b>	<b>7</b>
6.1	Accessibilità visiva . . . . .	7
6.2	Accessibilità cognitiva . . . . .	8
6.3	Accessibilità motoria . . . . .	8
6.4	Accessibilità generale . . . . .	8
<b>7</b>	<b>SEO</b>	<b>8</b>
<b>8</b>	<b>Test</b>	<b>9</b>
<b>9</b>	<b>Organizzazione del lavoro</b>	<b>9</b>

## 1 Abstract

Il sito presentato è una vetrina per un museo fittizio chiamato “Leggende Animate”, dedicato ai cartoni animati che hanno segnato l’infanzia dei membri del gruppo, come Yu-Gi-Oh, Pokémon, Dragon Ball e Gormiti. L’obiettivo del progetto è quello di proporre un’idea innovativa di museo, che espone immagini e carte reali legate a questi cartoni animati. Per alcuni, questi materiali possono essere considerati vere e proprie opere d’arte, create da artisti il cui talento spesso non riceve il riconoscimento che meriterebbe da parte del grande pubblico.

Il sito offre tutte le funzionalità essenziali per un museo, come la possibilità di prenotare una visita, lasciare una recensione e consultare le mostre in programma. Inoltre, è stato integrato un tour virtuale, che permette agli utenti di esplorare le diverse sale del museo online, stimolando curiosità e incentivando l’interesse di potenziali visitatori.

Il progetto è stato sviluppato seguendo rigorosamente le linee guida sull’accessibilità e le best practice di progettazione web, con l’obiettivo di renderlo fruibile da chiunque sia interessato al tema.

## 2 Analisi

### 2.1 Caratteristiche dell’utenza

Il sito è pensato principalmente per un pubblico giovane, familiare con i cartoni animati o le carte protagoniste del museo. Tuttavia, considerata la popolarità e la vasta diffusione di alcuni di questi franchise, è probabile che attragga anche appassionati adulti e genitori interessati ad accompagnare i propri figli nella visita. Questo rende lo spettro di utenza molto ampio.

Per rispondere a questa diversificazione, il gruppo ha scelto di evitare contenuti troppo superficiali, che andrebbero a diminuire l’interesse da parte del pubblico adulto, mantenendo però uno stile allegro e innovativo, adatto ai giovani. L’aspetto grafico del sito è stato curato con l’uso di colori vivaci e linee morbide, simulando l’atmosfera dei cartoni animati e le forme dei loro personaggi.

### 2.2 Funzionalità

Il sito dispone di varie funzionalità, alcune disponibili solo a certe categorie di utenti:

- un utente **non autenticato** può:
  - visualizzare la homepage del sito, comprensiva di carosello mostre e breve descrizione del museo;
  - visualizzare le mostre in corso, in programma e passate del museo;
  - visualizzare le recensioni del museo scritte da altri utenti;
  - registrarsi al sito per avere accesso ad ulteriori funzionalità;
  - accedere al sito, dopo essersi registrato;
- un utente **autenticato**, oltre alle funzionalità descritte sopra, può:
  - visualizzare il virtual tour, comprensivo di ogni sala e di ogni opera interna alla sala;
  - prenotare la visita al museo, per se stesso o per un massimo di 15 visitatori;
  - recensire la visita fisica effettuata al museo, o l’esperienza del virtual tour;
  - visualizzare la sua area personale, comprensiva dei suoi dati di registrazione e delle prenotazioni effettuate;
  - cancellare una prenotazione effettuata, tramite l’area personale;
- un utente **amministratore**, oltre alle funzionalità descritte sopra, può:
  - visualizzare nell’area admin tutte le mostre esposte nel sito;
  - rimuovere le mostre esposte nel sito;
  - modificare le mostre esposte nel sito;
  - aggiungere mostre da esporre nel sito.

## 3 Progettazione

### 3.1 Struttura del sito

Lo schema organizzativo è ibrido: principalmente è per argomento e in alcuni casi per task.

La struttura organizzativa utilizzata è di tipo gerarchico: l'ampiezza massima del sito non supera le 7 voci, mentre la profondità massima è di 3 livelli (home → virtual tour → sala → opera).

In generale in ogni pagina del sito troviamo questi elementi:

- **header:** contiene il logo del sito, presentato tramite un tag `<h1>` con il nome del museo. Nonostante non sia visibile, il testo è stato inserito per essere raggiungibile da screen reader e motori di ricerca, migliorando accessibilità e indicizzazione;
- **menù:** nel menù sono presenti tutte le pagine raggiungibili dalla homepage, con qualche differenza che dipende dalla tipologia di utente. Le voci “Accedi” e “Registrati” vengono sostituite da “Il mio profilo” nel caso di utente autenticato e l'utente admin ha accesso ad una voce aggiuntiva “Admin” per gestire le mostre esposte nel sito. La voce relativa alla pagina corrente non contiene link, per evitare link circolari;
- **breadcrumb:** permette all'utente di visualizzare il percorso di navigazione e orientarsi facilmente. Anche qui, la pagina attuale non è cliccabile, per evitare link circolari;
- **contenuto:** la sezione centrale della pagina contiene il contenuto principale, introdotto da un tag `<h1>` con un titolo descrittivo che risponde immediatamente alla domanda dell'utente: “Di cosa si tratta?”;
- **“Torna su”:** elemento che compare alla fine del contenuto per tornare rapidamente a inizio pagina, migliorando l'usabilità;
- **footer:** nel footer è presente nuovamente il logo del sito tramite tag `<h2>`, una lista di informazioni essenziali del museo, come indirizzo e contatti, e una tabella con i giorni e gli orari di apertura.

### 3.2 Pagine di errore

Oltre alle pagine di contenuto principali, il sito include anche pagine dedicate alla gestione degli errori, ciascuna progettata per fornire un messaggio simpatico che rassicura l'utente e lo aiuta a risolvere il problema. In particolare gli errori gestiti sono:

- Errore 400: causato da una richiesta non valida;
- Errore 403: causato da un tentativo di accesso non autorizzato;
- Errore 404: causato da una pagina non trovata;
- Errore 500: causato da un errore interno del server.

## 4 Realizzazione

### 4.1 HTML

Il sito è stato sviluppato in HTML5, rispettando la sintassi XML.

Per le parti dinamiche dei vari file HTML, abbiamo sviluppato un sistema di templating, che consiste nella suddivisione delle parti dinamiche in sezioni riconoscibili come commento HTML (`<!-- nomesezione_start -->` e `<!-- nomesezione_end -->`), nelle quali i valori dinamici sono indicati con ancora; nella sezione PHP viene spiegato in dettaglio l'utilizzo delle funzioni che permettono queste sostituzioni.

## 4.2 CSS

Per la parte di presentazione è stato utilizzato CSS, utilizzando un file unico chiamato `style.css`. Per tutte le pagine abbiamo utilizzato un layout basato su grid ed è stato utilizzato anche il layout flex-box, entrambi implementati con un livello di annidamento non eccessivo. Inoltre per aumentare la responsiveness generale del sito, abbiamo creato altri due file: `middle.css` (max-width: 1024px) e `mini.css` (max-width: 768px), che vanno a specificare le nuove regole di stile dove necessario. Si noti che questi non sono gli unici punti di rottura individuati, ma sono i principali: altri punti di rottura, per i quali sono state definite specifiche regole, sono, ad esempio, 600px e 500px. In questo modo il layout ottenuto è un ibrido tra fluido (misure in percentuale), elastico (misure in em) e responsive (più punti di rottura). Infine è presente un file per la stampa chiamato `print.css`.

## 4.3 Validazione dell'input

Le validazioni sull'input dell'utente, per assicurare consistenza, sono effettuate sia lato client con JavaScript, che lato server usando PHP. Le segnalazioni degli errori lato client non sono invasive per non disturbare l'utente durante l'inserimento o la modifica, in quanto l'errore viene segnalato dopo che l'utente ha immesso l'input, e i messaggi di errore indicano chiaramente quale sia il problema da risolvere. Il sito utilizza diversi moduli, perciò i controlli sono vari e mirati. Essi utilizzano espressioni regolari per assicurarsi ad esempio che non ci siano caratteri speciali pericolosi in termini di sicurezza, che nome e cognome non abbiano alcun carattere speciale, che l'email dell'utente sia valida e che la password scelta sia abbastanza sicura, cioè contenga almeno una lettera, un carattere speciale tra quelli consentiti e un numero. Inoltre ci sono controlli che verificano la validità delle date (ad esempio che una mostra non termini prima di essere iniziata) e che, grazie alla tecnica AJAX, verificano la disponibilità di biglietti per prenotare una visita al museo in una determinata fascia oraria di una determinata data. A tale proposito si è ipotizzato che nella stessa fascia oraria non possano essere dentro al museo più di 90 visitatori.

## 4.4 JavaScript

Oltre che per la validazione dell'input e la tecnologia AJAX per realizzarla, JavaScript è usato anche per altri scopi:

- cambio tema da chiaro a scuro, con priorità alla preferenza sul localStorage;
- aggiornamento dinamico del carosello nella pagina iniziale che, esponendo le immagini delle mostre in programma e permettendo di visualizzare il dettaglio di quella mostra con il pulsante "scopri di più", deve cambiare le immagini e la destinazione del tasto a seconda delle mostre nel database in quel momento (che possono essere aggiunte o eliminate dall'utente admin);
- apertura e chiusura del menù ad hamburger;
- gestione dell'accordion nella pagina delle mostre, per aprire e chiudere le mostre in programma, correnti e passate (con l'opzione dell'apertura automatica dell'accordion se si arriva da un link che portava ad una mostra nascosta);
- in fase di inserimento di una recensione, quando al click/tap o hover su un pallino per il voto, accanto ad esso appare automaticamente la corrispondente valutazione testuale;
- nella pagina virtual tour, quando si preme sull'immagine della sala, la sua descrizione viene enfatizzata, con un colore che ha maggiore contrasto rispetto allo sfondo;
- per la cancellazione di una mostra (admin) o prenotazione (utente), essendo operazioni irreversibili, appare un dialog di conferma operazione.

Tutto il codice JavaScript è organizzato in unico file `script.js` per garantire la corretta separazione tra struttura, presentazione e comportamento. La scelta di utilizzare un unico file è motivata dal fatto che in questo modo tutto il codice che governa il comportamento del sito è reperito tramite un'unica richiesta fatta dal client al server, scenario preferito rispetto al dover fare tante richieste più piccole al server.

## 5 Database

Il database, realizzato in terza forma normale, è semplice ma molto efficace per gestire tutti i dati necessari per il nostro museo.

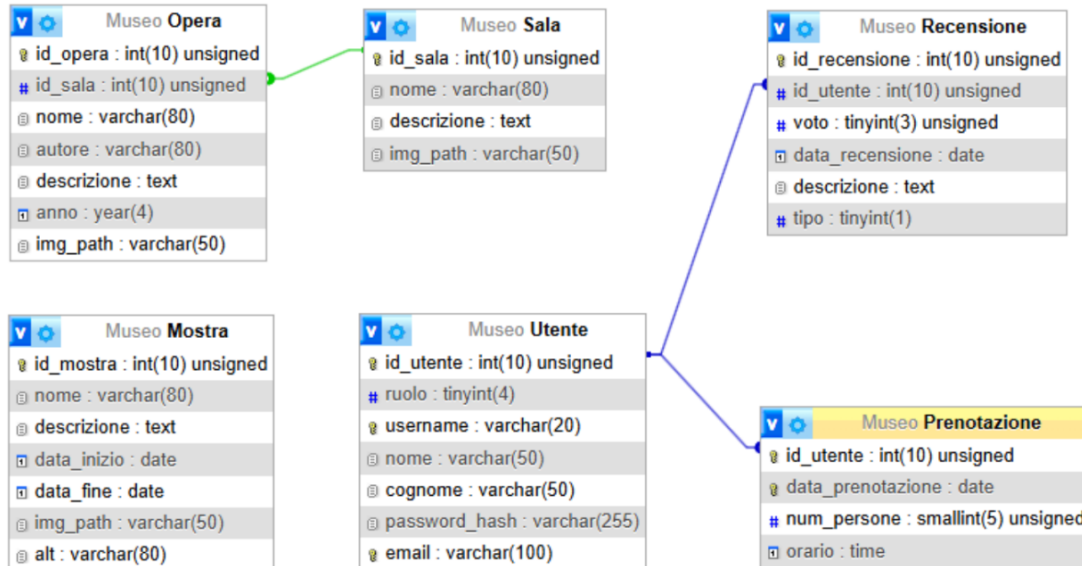


Figura 1: Schema del database

Nella cartella "database" dalla root del progetto si può accedere al file `init.sql`, che contiene le query necessarie alla creazione del database e delle tabelle e un dump con dei dati realistici per popolare la base di dati.

Per il popolamento del database, considerando che poi i dati esposti devono essere accessibili, è stato adottato un meccanismo di replacing per permetterci di inserire link, abbreviazioni e vocaboli in lingua in modo accessibile.

- Per inserire un link usiamo `aitexampleText/ahttps://wiki.pokemoncentral.it/exampleDest`, dove "a" sta per il tag del link, "it" per il codice ISO della lingua, il testo tra "" per l'indirizzo e la stringa `exampleText` per il testo che vede l'utente;
- Per le abbreviazioni usiamo `abbrexampleTitle;exampleAbbreviation/abbr`, sostituendo `exampleTitle` con il testo esteso e `exampleAbbreviation` con l'abbreviazione;
- Per il linguaggio invece usiamo `enexampleText/en`, dove al posto di "en" mettiamo il codice ISO della lingua che ci interessa e al posto di `exampleText` il testo in lingua.

### 5.1 PHP

Le pagine, come anticipato nella sezione HTML, sono rese dinamiche utilizzando PHP e il sistema di templating. È presente una cartella `php libs` che contiene delle classi di supporto allo sviluppo delle pagine `php`, in particolare:

- `DatabaseService` (`databaseService.php`): che gestisce il collegamento al database tramite `mysql` e utilizza le prepared query per rendere il sistema più veloce e più sicuro da eventuali vulnerabilità;
- `controllaDisponibilita.php` : riceve richieste di tipo post e risponde con la disponibilità effettiva in termini di capienza del museo, dopo aver consultato il database. Questo file è consultato dinamicamente durante la fase di prenotazione tramite una chiamata che parte da un controllo javascript per validare l'input della form;
- `Templating` (`templatingService.php`): questa classe racchiude la logica che permette ai vari file `php`, di generare dinamicamente le sezioni di codice HTML dei corrispettivi file. Il workflow di ogni pagina `php` potrebbe essere descritto in maniera procedurale:

1. Viene interrogato il database con la query necessaria alla specifica pagina php e HTML;
2. Viene caricata la pagina HTML con le parti dinamiche da rimpiazzare, tramite la funzione `getHTMLFileContent` (utilizza il nome del file php, con il presupposto che ogni file php si differenzia dal corrispettivo HTML, tramite l'estensione e il path);
3. Vengono selezionate le diverse sezioni da elaborare della pagina HTML utilizzando `getContentBetweenPlaceholder`, che riceve come parametro nome sezione (ricordando il formato `<!-- nome_sezione_start -->`);
4. Vengono rimpiazzate le variabili ottenute al punto 1, con le variabili che si trovano tra le `{{variabile_dinamica}}` tramite `replaceAnchor`;
5. La vecchia sezione viene sostituita con `replaceContentBetweenPlaceholder`;
6. Viene mostrata la pagina e cancellati eventuali commenti (inizio e fine delle sezioni) con `showHTMLPageWithoutPlaceholders`

Inoltre dentro a questo file vengono gestite le conversioni dei dati inseriti nelle query che andrebbero ad utilizzare tag HTML (si veda sez. Database). È presente una funzione `uploadImg`, che riguarda la gestione dell'inserimento delle immagini Client Side, che effettua diversi controlli sulla size ed estensioni. Un'altro punto rilevante realizzato tramite PHP è la dinamicità delle immagini del carosello. Nella pagina home, inizialmente le immagini raffigurano le mostre presenti e future inserite nel db (il bottone scopri di più riporta quindi alla pagina mostre e alla sezione specifica della mostra su cui si trovava il carosello). Nel caso in cui il database fosse privo di questo tipo di mostre, vengono illustrate alcune immagini del museo di default, e se cliccato scopri di più, avviene il reindirizzamento alla pagina virtual tour dove l'utente può comunque esplorare sale e opere del museo.

## 5.2 Elementi dell'interfaccia utente

### 5.2.1 Immagini e icone

Le immagini sono salvate in formato WEBP e hanno una dimensione dell'ordine 100 Kb per permettere buone performance sia da mobile che con desktop anche in presenza di bassa latenza. La favicon è nel formato .ico, supportato da tutti i browser.

### 5.2.2 Font

Come font principale è stato scelto Lexend, senza grazie e accessibile. Per la stampa, invece, è stato scelto Bitter, con grazie, più accessibile rispetto al predefinito Times New Roman (ad esempio, 1 e l sono piuttosto simili).

### 5.2.3 Colori

La gamma di colori scelti è piuttosto ampia, anche in linea con il tema del museo. Per il tema chiaro sono stati usati principalmente i seguenti colori:

- #205E86 (blu pastello): link non visitati menù e main, sfondo breadcrumb, testo e link visitati footer;
- #95E98A (verde pastello): sfondo header e footer, testo e link non visitati breadcrumb, sfondo ausiliario per alcune tabelle;
- #000 (nero): testo main, link visitati menù e main;
- #FFF (bianco): sfondo principale;
- #F2F4F7 (grigio molto chiaro): sfondo nella home e sfondo ausiliario per alcune tabelle;
- #FBFF50 (giallo): sfondo link visitati menù, link visitati breadcrumb;
- #EB0000 (rosso): messaggi d'errore.

Per il tema scuro sono stati usati principalmente i seguenti colori:

- #2B7BB1 (blu): sfondo breadcrumb e footer, link non visitati main;

- #95E98A (verde pastello): sfondo link visitati menù;
- #000 (nero): sfondo principale e header, link non visitati footer;
- #FFF (bianco): testo e link visitati footer, link visitati breadcrumb;
- #DDD (grigio chiaro): testo e link visitati main;
- #232323 (grigio molto scuro): sfondo ausiliario per alcune tabelle;
- #FF3D3D (rosso): messaggi d'errore.

I contrasti tra sfondo e testo (anche per i colori non qui specificati) rispettano sempre il minimo di 4.5:1, e i contrasti tra link visitati e non visitati (o relativo sfondo) rispetta il minimo di 3:1. L'unica eccezione è rappresentata dalla coppia #FBFF50 - #95E98A (link visitati - non visitati nelle breadcrumb) con contrasto 1.37:1. Tuttavia, testando tramite Silktide con gli appositi filtri, si è rilevata una distinzione visibile anche con il filtro più critico, riportato nello screenshot, che mostra il link visitato Home (#FBFF50), ben distinguibile dal colore del testo Le mostre (#95E98A, stesso colore dei link non visitati). Inoltre questa eccezione non viola le WCAG (<https://www.w3.org/WAI/WCAG22/Understanding/use-of-color.html>).

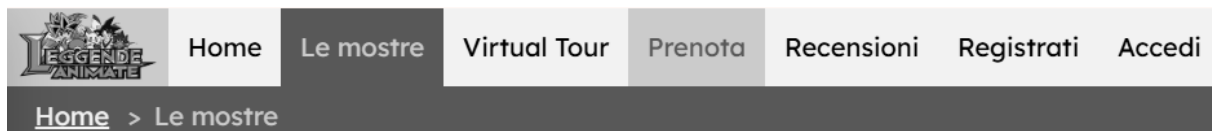


Figura 2: Visione di menù e breadcrumb con filtro acromatopsia

## 6 Accessibilità

### 6.1 Accessibilità visiva

Per migliorare la leggibilità del testo:

- viene usato il font accessibile Lexend (senza grazie);
- l'interlinea minima è 1.5;
- il contrasto minimo tra lo sfondo e il testo è 4.5:1;
- le dimensioni del testo sono espresse in em e rem.

Per migliorare l'accessibilità visiva in generale:

- è stato implementato il tema scuro;
- è possibile ingrandire il testo senza perdere funzionalità o leggibilità del sito.
- le immagini decorative hanno l'alt (testo alternativo) vuoto, mentre per le immagini di contenuto vengono seguiti due approcci: le immagini del carosello (nella home) sono inserite in un tag **figure** e accompagnate da didascalia in **figcaption**; nella pagina di dettaglio opera le immagini hanno l'alt che contiene il nome dell'opera e l'attributo **aria-describedby** con l'id della descrizione, che segue l'immagine.
- le immagini non contengono testo, eccetto il logo, che viene mostrato tramite CSS con image replacement (quindi lo screen reader è in grado di leggere il testo).



## 6.2 Accessibilità cognitiva

Per rendere il sito più comprensibile e ridurre il sovraccarico cognitivo dell'utente:

- il design è semplice e intuitivo, con il logo e il menù principale in alto (che per mobile diventa un menù a tendina), successivamente le breadcrumb, il contenuto principale e il footer;
- il linguaggio è generalmente semplice, anche se vengono usati termini specifici relativi ai cartoni animati;
- sono stati evitati contenuti lampeggianti o in movimento: le uniche animazioni avvengono in seguito a un'azione dell'utente (ad esempio hover del mouse su un pulsante);
- i messaggi di errore, in particolare nei moduli, sono facilmente comprensibili e dettagliati;
- i pulsanti e i link hanno etichette descrittive e chiare;
- i link presenti nel testo sono sottolineati (convenzione esterna);
- i link visitati sono distinguibili dai link non visitati;
- nell'area sicura vengono sempre mostrati logo (identità), menù principale (interazione), breadcrumb, titolo principale (contenuto), favorendo l'orientamento.

## 6.3 Accessibilità motoria

Per facilitare l'interazione con il sito agli utenti con difficoltà motorie:

- il sito è navigabile interamente da tastiera;
- il focus è ben visibile sugli elementi selezionati;
- i click target sono di dimensioni superiori o uguali a 24x24 px;
- non sono presenti interazioni basate su movimenti complessi (es. swipe o pinch);
- per le operazioni irreversibili (ad es. cancellazione di una prenotazione), viene chiesta la conferma o l'annullamento.

## 6.4 Accessibilità generale

Per migliorare l'accessibilità in generale:

- viene usato HTML5 valido e semanticamente corretto;
- vengono usati gli attributi ARIA;
- nei dispositivi mobile, i link del menù a tendina non sono raggiungibili via tab se il menù non è espanso (attributo inert);
- tutti i campi dei moduli sono associati ad un'etichetta;
- vengono forniti gli aiuti allo screen reader: tutte le pagine includono link "Vai al contenuto" e "Torna su". Quest'ultimo è anche visibile, perché può essere utile anche a chi non usa tecnologie assistive, specialmente su dispositivi mobile.

# 7 SEO

Il sito è stato progettato per rispondere efficacemente a ricerche come museo cartoni animati, Leggende Animate, esposizioni carte collezionabili e mostra Pokémon/Yu-Gi-Oh/Gormiti/Dragon Ball. Per migliorare il posizionamento nei motori di ricerca, sono stati utilizzati i metatag:

- **title:** strutturati dal particolare al generale;
- **description:** con una call to action che stimola l'interesse e invoglia l'utente a visitare il sito;

- **keywords:** inserite in ogni pagina (anche dinamicamente), scelte in base alla coerenza con il contenuto e con il tema del museo.

Inoltre è stata rispettata la gerarchia delle intestazioni (h1, h2, h3), includendo anche in esse parole chiave rilevanti per migliorare il posizionamento nelle SERP. Altri fattori migliorativi sono la presenza di alt (eventualmente vuoto) nelle immagini; i link forniti di un testo descrittivo e scansionabili dai motori di ricerca; la velocità di caricamento (misurata con Lighthouse).

## 8 Test

Tutto il sito è stato realizzando secondo le specifiche del livello AA delle Web Content Accessibility Guidelines (WCAG). I test sono stati eseguiti con vari strumenti:

- Total Validator, il quale ha unicamente rilevato l'errore E400 Link: problem with link: Bad Request. Il problema deriva dal link che il sito mette a disposizione nel footer di ogni pagina per visualizzare sulla mappa l'indirizzo del museo. La segnalazione di Total Validator, però è un falso positivo, in quanto il link <https://maps.app.goo.gl/xqD3pfvXG4ockxUeA> porta correttamente all'indirizzo indicato;
- achecker per eseguire ulteriori test sulla validazione del codice;
- WAVE rileva alcuni errori di contrasto nella home, ma si tratta di testo non visibile. Inoltre segnala un alert relativo al div con id="overlay", presente in ogni pagina: esso ha la funzione di oscurare parzialmente il contenuto mentre il menù ad hamburger è espanso, e permette di chiudere il menù semplicemente facendo click/tap fuori dal menù stesso. È volutamente non raggiungibile via tab;
- Toptal, con cui sono stati eseguiti test per protanopia, deuteranopia, tritanopia e acromatopsia;
- Silktide per un test generale sull'accessibilità;
- WCAG contrast checker: anch'esso segnala errori di contrasto, ma si tratta di testo non visibile, oppure dovuti a un'errata rilevazione dei colori da parte del tool (ad esempio con il tema scuro);
- Lighthouse, per testare prestazioni, best practices, accessibilità e SEO;
- Compatibilità con browser testata sui browser principali: Chrome, Firefox, Edge, Safari;
- NVDA, screen reader per Windows.

## 9 Organizzazione del lavoro

Il lavoro è stato suddiviso tra i membri del team nel modo più equilibrato possibile, con ciascuno che ha assunto specifiche responsabilità e ha contribuito attivamente alla creazione del sito in tutte le sue fasi. Questa distribuzione ha permesso di ottimizzare il tempo a disposizione e di affrontare in maniera organizzata ogni aspetto del progetto. Per facilitare la collaborazione e il controllo del codice, è stata utilizzata GitHub come piattaforma centrale, consentendo al gruppo di condividere il lavoro, tracciare le modifiche e gestire la suddivisione dei compiti in modo efficiente. La ripartizione del lavoro veniva suddivisa ogni due settimane circa, tramite dei meeting online, permettendo la distribuzione del lavoro abbastanza equa, evitando di affrontare un accumulo di lavoro all'ultimo minuto. Di seguito è riportata una lista delle attività svolte da ciascun membro del team:

- Bergamin Elia:
  - HTML;
  - CSS;
  - popolamento del database;
  - PHP;
  - JavaScript;
  - stesura della relazione.

- Corradin Samuele:
  - HTML;
  - CSS;
  - popolamento del database;
  - JavaScript;
  - stesura della relazione.
- Milan Riccardo:
  - HTML;
  - CSS;
  - popolamento del database;
  - PHP;
  - JavaScript;
  - stesura della relazione.
- Peruzzi Uncas:
  - HTML;
  - CSS;
  - popolamento del database;
  - PHP;
  - JavaScript;
  - stesura della relazione.