



# GagGym

*Progetto corso di Tecnologie Web A.A. 2023/2024*

## Realizzato da:

| Matricola | Nome e Cognome                        | Email                                 |
|-----------|---------------------------------------|---------------------------------------|
| 2042346   | Riccardo Carraro ( <b>referente</b> ) | riccardo.carraro.10@studenti.unipd.it |
| 2042346   | Endi Hysa                             | riccardo.carraro.10@studenti.unipd.it |
| 2042346   | Andrea Giurisato                      | riccardo.carraro.10@studenti.unipd.it |
| 2042346   | Michele Ogniben                       | riccardo.carraro.10@studenti.unipd.it |

## Indirizzo del sito

<http://tecweb.studenti.math.unipd.it/rcarraro>

## Repository del progetto

<https://github.com/Carraro-Riccardo/Progetto-Tecnologie-Web>

## Credenziali di utilizzo

Username: *admin*, Password: *admin*

Username: *user*, Password: *user*

# Indice dei contenuti

|       |  |   |
|-------|--|---|
| 1     | Introduzione .....                     | 3 |
| 2     | Approccio allo sviluppo .....          | 4 |
| 2.1   | Mockup .....                           | 4 |
| 2.2   | Impatto grafico .....                  | 4 |
| 3     | SEO .....                              | 4 |
| 4     | Progettazione .....                    | 5 |
| 4.1   | Schema organizzativo e struttura ..... | 5 |
| 4.2   | Utenti .....                           | 5 |
| 4.3   | Convenzioni interne .....              | 5 |
| 5     | Implementazione .....                  | 6 |
| 5.1   | Linguaggi utilizzati .....             | 6 |
| 5.2   | HTML5 .....                            | 6 |
| 5.2.1 | Struttura e Contenuto .....            | 6 |
| 5.2.2 | Componenti .....                       | 6 |
| 5.3   | CSS3 .....                             | 6 |
| 5.4   | PHP .....                              | 7 |
| 5.4.1 | Costruzione delle pagine .....         | 7 |
| 5.4.2 | Collegamento database .....            | 7 |
| 5.4.3 | SQL injection .....                    | 7 |
| 5.4.4 | Librerie esterne .....                 | 7 |
| 5.5   | Javascript .....                       | 8 |
| 5.6   | Validazione input utente .....         | 8 |
| 5.7   | Database .....                         | 8 |
| 5.7.1 | Schema .....                           | 8 |
| 5.8   | Pagine di errore .....                 | 8 |
| 6     | Accessibilità .....                    | 9 |
| 6.1   | Font .....                             | 9 |
| 6.2   | Colori e Contrasti .....               | 9 |
| 6.3   | Ulteriori accorgimenti .....           | 9 |
| 7     | Test e verifica .....                  | 9 |

# 1 Introduzione

GagGym (letto “*Gaggim*”, come scherzoso gioco di parole tra il cognome della professoressa Ombretta Gaggi e il termine inglese “*Gym*”) è un sito web ideato come sito di presentazione, esplorazione, iscrizione e amministrazione di una palestra, con l’obiettivo di fornire un’esperienza utente semplice e intuitiva.

L’utente può visualizzare le informazioni essenziali della palestra, i macchinari di cui dispone, lo staff presente e procedere, qualora fosse interessato, all’iscrizione, ottenendo così accesso alla sua area riservata.

Nella sua area personale, l’utente potrà visualizzare le informazioni del profilo, la scheda di allenamenti seguita, verificare l’abbonamento sottoscritto e caricare il certificato medico da far validare dall’amministratore. Tra le informazioni del profilo dell’utente sarà presente anche un QR code identificativo, utilizzabile per l’accesso rapido alla palestra mediante un lettore fisico dedicato, pratica ormai comune in molte palestre.

L’amministratore, invece, potrà accedere all’area amministrativa, con metriche e statistiche sulla palestra, e gestire i macchinari presenti, gli utenti e la validazione dei loro certificati medici.

## 2 Approccio allo sviluppo

### 2.1 Mockup

La fase iniziale dei lavori ha visto come primo passo la realizzazione di un mockup del sito, per avere un punto di partenza per l'organizzazione, il contenuto e lo stile delle pagine. Il gruppo ha utilizzato un approccio *mobile-first*, non solo perchè consapevole che la maggior parte degli accessi ad internet avviene tramite dispositivi mobili, ma anche perchè risulta più semplice, in fase implementativa, adattare un sito web da mobile a desktop che viceversa, dato che le dimensioni ridotte della viewport mobile potrebbero rappresentare un fattore critico se non sufficientemente considerate.

### 2.2 Impatto grafico

Il gruppo ha imposto uno stile *cartoonesco* e simpatico, con una struttura semplice e intuitiva, adoperando un lessico facilmente comprensibile e un linguaggio informale, per rendere l'esperienza utente più piacevole e divertente.

Il sito è stato progettato per essere utilizzabile da un ampio raggio di utenza, sia da coloro che già conoscono la palestra, sia da coloro che la stanno scoprendo per la prima volta. Il range di età è compreso tra i 15 e i 50 anni, considerando una maggioranza di utenti tra i 18 e i 40 anni.

## 3 SEO

La SEO è un elemento fondamentale per un sito web, in quanto permette di posizionarsi in modo ottimale nei risultati di ricerca, aumentando la visibilità e il traffico sul sito. GagGym intende a rispondere a ricerche come:

- GagGym (nome del sito);
- Ricerche generali su palestre;
- Ricerche in merito a corsi di fitness (yoga, pilates, ecc.);
- Ricerche in merito a macchinari ed esercizi.

Per raggiungere questo obiettivo, ogni pagina è stata progettata seguendo le *good practice* viste a lezione:

- HTML5 valido e semanticamente corretto;
- Utilizzo di tag *meta description* per descrivere il contenuto della pagina;
- Utilizzo di tag *meta keywords* per descrivere le parole chiave della pagina;
- Utilizzo di tag *title* in modo pertinente e conciso.

Il gruppo è altresì consapevole che la SEO è un processo che non termina con la pubblicazione del sito, e che il suo miglioramento dipende anche da fattori esterni come la bontà dei link d'ingresso.

## 4 Progettazione

### 4.1 Schema organizzativo e struttura

Il sito è stato progettato per essere facilmente navigabile. Si utilizza uno schema organizzativo esatto, che predispone ogni elemento in modo coerente identificando i contenuti in modo chiaro ed evidente (corsi, abbonamenti, macchinari, schede, staff).

La struttura del sito è ampia e poco profonda, permettendo non solo il raggiungimento rapido delle informazioni ricercate (con una media di 3 click per raggiungere la pagina desiderata) ma anche una manutenzione più semplice e veloce.

La navigazione tra le pagine del sito è resa possibile tramite un menù a tendina contenente 6 voci (Home, Corsi, Abbonamenti, Macchinari, Staff, Login), rispettando le *best practice* in merito al numero di voci consigliate (non più di 7) per non appesantire l'overhead cognitivo dell'utente.

### 4.2 Utenti

Il sito è stato pensato per rispondere alle esigenze di 3 tipologie di utenti:

- **Utente non autenticato:** utente che accede al sito senza aver effettuato il login. Può visualizzare le informazioni pubbliche della palestra, i macchinari presenti e lo staff. Può inoltre registrarsi al sito, ottenendo così un account utente e la sua area personale.
- **Utente autenticato:** utente che ha effettuato il login. Può visualizzare le informazioni pubbliche della palestra, i macchinari presenti e lo staff. Può inoltre visualizzare le informazioni del proprio profilo, la scheda di allenamenti seguita, verificare l'abbonamento sottoscritto e caricare il certificato medico da far validare dall'amministratore.
- **Amministratore:** utente che ha effettuato il login con privilegi di amministratore. Può accedere all'area amministrativa, con metriche e statistiche sulla palestra, e gestire i macchinari presenti, gli utenti e la validazione dei loro certificati medici.

### 4.3 Convenzioni interne

- Su schermi con larghezza della viewport superiore a 660px, il menù a tendina viene mostrato orizzontalmente anziché verticalmente, per sfruttare al meglio lo spazio disponibile.
- All'interno dell'header, il logo della palestra rispetta la convenzione esterna di essere cliccabile e di condurre alla homepage. Questo approccio creerebbe una ripetizione del link, presente sia nel logo che nel nome della palestra presente nella nav-bar principale. Per agevolare la navigazione del sito agli utenti utilizzatori di screen reader (o di navigazione da tastiera (tab)) è stato deciso di mantenere il link anche nel logo, ma di nascondere allo screen reader e alla tabulazione per evitare ripetizioni. Questo è stato possibile tramite l'attributo `aria-hidden="true"` e `tabindex="-1"`.
- Al fine di rimuovere i link circolari, all'interno del menù a tendina, il link che condurrà alla pagina corrente sarà sostituito da uno span. Questa sostituzione avviene tramite un controllo lato server, e sfrutta la natura *inline* dello span, che non altera la struttura del menù.
- L'utente amministratore non possiede una classica pagina di profilo, bensì, una volta effettuato il login, entra direttamente nella pagina di amministrazione. Questa scelta, seppur possa sembrare una rottura delle convenzioni interne, è dettata dalla distinzione netta delle due tipologie utente: l'amministratore non è un cliente della palestra, bensì ne è il gestore, e pertanto non possiede dettagli come certificato medico, scheda, qr di accesso o abbonamento, accedendo direttamente alla pagina di amministrazione. Inoltre, si assume che l'amministratore sia un utente che, in qualità di gestore, sia consapevole delle funzionalità amministrative. Per rendere evidente il cambio di contesto, non solo lo sfondo della pagina cambia, ma anche il testo del menù a tendina, da "Menù" diventa "Menù admin", contenendo le nuove voci di amministrazione.

- A seguito del login, la voce corrispondente nel menù viene sostituita dallo username dell'utente loggato: siamo consapevoli che il login con username e password “*user*” possa in questo modo far mostrare la voce “*user*” nel menù, ma si tratta di un caso limite dettato dai requisiti del progetto.

## 5 Implementazione

### 5.1 Linguaggi utilizzati

Il sito è stato realizzato utilizzando i seguenti linguaggi:

- HTML5: per la struttura delle pagine;
- CSS3: per la presentazione delle pagine;
- PHP: per il lato server;
- JavaScript: per il lato client.

### 5.2 HTML5

#### 5.2.1 Struttura e Contenuto

La struttura del sito è stata realizzata utilizzando HTML5. Le pagine HTML sono utilizzate come template per la generazione dinamica dei contenuti, utilizzando PHP, pertanto è stato necessario inserire dei placeholder per i contenuti dinamici:

- “@@placeholder@@” per semplici stringhe;
- “<!-- inizio contenuto dinamico -->” e “<!-- fine contenuto dinamico -->” per contenuti più complessi, come tabelle, cards o liste.

Questo non solo contribuisce alla leggibilità del codice, ma permette anche di distinguere facilmente i contenuti statici da quelli dinamici, creando inoltre una netta separazione tra struttura, presentazione e comportamento. Inoltre, l'utilizzo delle pagine HTML come template per l'elaborazione dinamica dei contenuti, permette di dover contattare il server solo una volta per pagina, riducendo il carico di lavoro del server e velocizzando il caricamento delle pagine.

#### 5.2.2 Componenti

Alcuni elementi comuni delle pagine sono stati resi componenti, ossia salvati come file HTML separati e aggiunti dinamicamente alle pagine tramite PHP, sfruttando i placeholder precedentemente descritti. Questo approccio permette di evitare la ripetizione di codice, rendendo anche più semplici eventuali modifiche a componenti condivisi. Gli elementi resi componenti sono:

- navbar principale di navigazione (*./HTML\_pages/componenti/navbar-principale.html*);
- footer (*./HTML\_pages/componenti/footer.html*);

### 5.3 CSS3

Lo stile del sito è stato realizzato utilizzando CSS3 utilizzando un approccio *mobile-first*. Questo ha permesso la realizzazione di un solo file CSS per tutti i dispositivi, gestendo dapprima il layout per mobile e successivamente, mediante *media query*, per i dispositivi desktop. Seguendo le pratiche viste a lezione e in laboratorio, il gruppo si è impegnato ad organizzare il file CSS commentando le sezioni secondo la pagina a cui si riferiscono, e ad raccogliere tutte le variabili di colore in un'unica sezione iniziale, in modo che la personalizzazione o l'aggiornamento dei colori, possa avvenire in modo rapido e semplice.

Mediante CSS sono stati gestiti tutti gli aspetti decorativi, in particolare tutte le immagini che prevedevano una natura decorativa e non informativa. Allo stesso modo anche le emoji presenti nella pacomprendendohtmogina dei corsi sono aggiunte mediante pseudo-elementi CSS come `::before` e `::after`.

Il gruppo inoltre ha valutato attentamente il ruolo delle immagini all'interno del sito: all'interno della homepage, le immagini che rappresentano le varie voci di navigazione sono state interpretate come decorative: difatti non rappresentano informativamente il contenuto della pagina di destinazione, ma sono unicamente un elemento decorativo che permette di rendere più piacevole la navigazione.

Lo stesso ragionamento è stato svolto per la pagina dei corsi: l'immagine del corso di yoga (ad esempio) non rappresenta informativamente il corso, ma è unicamente un elemento decorativo.

Pertanto, siccome il contenuto informativo rimane invariato anche senza l'immagine, queste sono state inserite come immagini decorative.

Al contrario, le immagini presenti nella pagina dei macchinari e dello staff sono state interpretate come informative: difatti rappresentano informativamente il contenuto della pagina e del contenuto riferito, e sono pertanto state inserite come tag `<img>` (il macchinario nell'immagine è l'unità informativa ricercata e concreta).

Le immagini sono in formato *webp*, permettendo un miglioramento anche in termini di performance e peso della pagina. Maggiori informazioni in merito sono presenti nella sezione *Test e verifica*.

## 5.4 PHP

Il lato server del sito è stato realizzato utilizzando PHP. Le pagine vengono generate dinamicamente, leggendo il contenuto delle corrispondenti pagine HTML e sostituendo i placeholder con i contenuti dinamici. L'utilizzo principale di PHP risiede nel collegamento con il database, la creazione della sessione, la generazione dinamica delle pagine e la gestione dei form. In particolare, per la gestione dei form, è stato utilizzato un approccio mediante il quale i dati inseriti dall'utente vengono processati in una pagina PHP dedicata, che si occupa di validare i dati e di restituire eventuali errori, che vengono mostrati all'utente nella pagina di origine. Esempi di questo approccio possono essere la pagina di *login* e la pagina di *registrazione*, dove i form presenti richiamano le rispettive pagine PHP *login\_handler.php* e *registrazione\_handler.php*.

### 5.4.1 Costruzione delle pagine

Per la costruzione delle pagine, si utilizza un file PHP dedicato, *pages\_builder.php*, il quale, visto il nome della pagina richiesta, legge il contenuto della pagina HTML corrispondente e ne costruisce l'impianto iniziale. Successivamente, questo contenuto viene restituito alla pagina che ne ha richiesto la costruzione, che si occuperà di sostituire i placeholder con i contenuti dinamici.

### 5.4.2 Collegamento database

Per il collegamento al database, è prevista una classe dedicata, *db\_connection.php*, che si occupa non solo di stabilire la connessione al database, ma anche di effettuare le query necessarie: questo permette la gestione centralizzata della logica di comunicazione con il database, rendendo più semplice e veloce la manutenzione del codice.

### 5.4.3 SQL injection

Per prevenire attacchi di tipo SQL injection, il gruppo ha utilizzato le *prepared statements* per tutte le query che prevedono l'inserimento di dati forniti dall'utente. Questo approccio permette di separare la query SQL dalla sua esecuzione, evitando che i dati forniti dall'utente possano essere interpretati come codice SQL.

### 5.4.4 Librerie esterne

Il progetto fa utilizzo di 2 librerie PHP esterne:

- *phpqrcode*: libreria per la generazione di QR code identificativo dell'utente.
- *jpggraph*: libreria per la generazione di grafici e metriche per la schermata dell'amministratore.

## 5.5 Javascript

Il gruppo ha cercato di rendere il sito indipendente da Javascript, utilizzandolo solamente per la validazione dei dati lato client. Il menù a tendina infatti, è impostato per essere mostrato aperto di default, e al caricamento della pagina, esso viene chiuso tramite Javascript. In questo modo, se Javascript è disabilitato, le funzionalità di navigazione rimangono comunque disponibili, e l'intero utilizzo del sito rimane possibile.

Si osserva però che tale soluzione non è ottimale, in quanto il servizio LightHouse di Google segnala un calo di performance su mobile, dovuto ad un problema di *cumulative layout shift* (CLS), causato dal caricamento del menù aperto e successivamente chiuso.

## 5.6 Validazione input utente

Per la validazione dei dati inseriti dall'utente, il gruppo ha impostato i medesimi controlli sia lato client che lato server, mostrando all'utente gli errori riscontrati in modo che possa essere consapevole della motivazione che ha portato alla generazione dell'errore.

Gli script di validazione lato client sono situati all'interno della cartella `./scripts`, aventi nome significativo in base al compito di validazione che svolgono.

Per la validazione lato server, il compito è demandato al file `./server_side_validator.php`, il quale fornisce funzioni di validazione mediante le quali è possibile ottenere l'eventuale messaggio di errore da mostrare all'utente.

Il messaggio di errore è cumulativo, pertanto, al presentarsi di più errori in un singolo form, il messaggio di errore mostrato indicherà tutti gli errori riscontrati, in modo che l'utente possa sapere con certezza le motivazioni dell'errore e come risolvere.

## 5.7 Database

Il database rispetta i vincoli dettati dalla consegna (terza forma normale). All'interno della cartella è presente il file `./db_progetto_tw.sql` per l'importazione del database.

### 5.7.1 Schema

Il database è composto da 10 tabelle:

- **utenti**: contiene le informazioni degli utenti registrati al sito;
- **abbonamenti**: contiene le informazioni sugli abbonamenti offerti dalla palestra;
- **utenti\_abbonamenti**: contiene le informazioni sugli abbonamenti sottoscritti dagli utenti;
- **scheda**: contiene le informazioni sulle schede di allenamento;
- **esercizi**: contiene le informazioni sugli esercizi;
- **schede\_esercizi**: contiene le informazioni sugli esercizi presenti nelle schede di allenamento;
- **schede\_utenti**: contiene le informazioni sulle schede di allenamento seguite dagli utenti;
- **macchinari**: contiene le informazioni sui macchinari presenti nella palestra;
- **allenatori**: contiene le informazioni sugli allenatori della palestra;
- **gruppimuscolari**: contiene elenco dei gruppi muscolari a cui gli esercizi fanno riferimento.

## 5.8 Pagine di errore

Al fine di migliorare l'esperienza utente, il gruppo ha deciso di implementare delle pagine di errore personalizzate, che vengono mostrate all'utente in caso di errore 404 o 500. Questo permette di mantenere l'utente all'interno del sito, evitando che venga reindirizzato ad una pagina di errore generica del browser. Le pagine di errore contengono un messaggio di errore semplice e simpatico, in tema con il sito, con suggerimenti in merito a come risolvere il problema.



Data la natura delle pagine di errore ben distinta dalle pagine di navigazione, il gruppo ha deciso di non mostrare la breadcrumb in queste pagine, fornendo però tutti i mezzi necessari per riprendere la navigazione, quali *header* (con il menù di navigazione principale) e il *footer*.

## 6 Accessibilità

### 6.1 Font

Il font utilizzato è *Oswald*, un font *sans serif* che supporta un ampio range di lingue, compresi alfabeti non latini come il cirillico. Al fine di rendere l'esperienza utente più piacevole, il testo non possiede sempre dimensioni superiori ai 16pixels (9pt).

### 6.2 Colori e Contrasti

La palette di colori utilizzata è stata scelta con attenzione al fine di permettere un contrasto sufficiente tra testo e sfondo, e in grado di distinguere nitidamente gli elementi presenti nella pagina. Ogni contrasto testo/background è stato verificato essere superiore a 7:1 (il minimo richiesto per il livello AAA di WCAG). Si osserva che il contrasto tra testo e sfondo è stato verificato utilizzando il tool *Color Contrast Checker* di WebAIM.

Sfortunatamente, il contrasto tra il colore dei link visitati e il color dei link non visitati, non presenta un contrasto sufficiente: seppur in entrambi i casi il contrasto con lo sfondo sia superiore a 7:1, il contrasto tra i due colori è inferiore a 3:1.

### 6.3 Ulteriori accorgimenti

- **Navigazione mediante tastiera:** il sito è navigabile mediante tastiera, utilizzando la tabulazione per spostarsi tra i vari elementi della pagina in modo coerente al contenuto presentato.
- **Utilizzo di attributi *aria*:** al fine di migliorare l'accessibilità del sito, sono stati utilizzati gli attributi *aria* per fornire informazioni aggiuntive agli screen reader. Un esempio è il menù a tendina, che fa utilizzo dell'attributo *aria-expanded* per indicare se il menù è aperto o chiuso.
- **Tabelle accessibili** seguendo le *best practice* viste a lezione.
- **<span lang="en">**: per le parole in lingua inglese, è stato utilizzato l'attributo *lang="en"* per indicare la lingua utilizzata, in modo da permettere agli screen reader di leggere correttamente il testo.
- **Classe *screen-reader-only*:** per permettere agli screen reader di leggere il contenuto di un elemento, senza che questo sia visibile all'utente, è stata utilizzata la classe *screen-reader-only*, che rende l'elemento invisibile, ma non nascosto agli screen reader.

## 7 Test e verifica

Il sito è stato sottoposto a diversi test atti a verificarne sia il funzionamento che la correttezza e validità del codice. Strumenti fondamentali durante la fase di test sono stati:

- **W3C Markup Validation Service:** per la validazione del codice HTML:
  - la validazione ha restituito 0 errori e 0 warning, ma un alcuni *info* dovuti allo *slash* / finale dei tag meta e link, ma che abbiamo visto a lezione essere buona pratica chiudere ogni tag.
- **W3C CSS Validation Service:** per la validazione del codice CSS
- **Contrast Checker di WebAIM:** per la verifica del contrasto tra testo e sfondo;
- **LightHouse** di Google Chrome: per la verifica delle performance, accessibilità, SEO e *best\_practice* (un report dettagliato sarà disponibile trovarlo per ogni pagina nel file dedicato *REPORT\_LIGHTHOUSE.pdf*).
  - Media Performance: 80% (per i motivi descritti in precedenza riguardo al menù a tendina);

- Per migliorare le performance, il sito utilizza immagini in formato *webp*, che permette di ridurre le dimensioni delle immagini senza perdita di qualità, risultando più leggere di *png* e *jpeg*. Le immagini inoltre sono state ulteriormente ottimizzate tramite strumenti come *TinyPNG* e *Simple Image Resizer*.
- Media Accessibility: 100%;
- Media Best Practice: 100%;
- Media SEO: 100%;
- **Screen reader:** per la verifica dell'accessibilità del sito:
  - Assistente vocale di Windows 10;
  - NVDA (seppur in modo molto limitato data l'inseperienza nell'utilizzo);
- **Compatibilità browser:** per la verifica della compatibilità del sito con i principali browser:
  - Google Chrome;
  - Mozilla Firefox;
  - Microsoft Edge;
  - Safari (desktop e mobile);
  - Opera;
  - Samsung Internet.

Si osserva inoltre che i test non sono stati effettuati solamente su *desktop* o su dispositivi simulati, ma anche su diversi dispositivi reali posseduti dal gruppo, in modo da avere un riscontro quanto più realistico possibile.