



# RELAZIONE TEC. WEB

A.A. 2021-2022

*Indirizzo web del sito:*

<http://tecweb.studenti.math.unipd.it/mmidena/index.php>

*Utente-password:*

admin-admin

trainer-trainer

client-client

*Componenti del gruppo:*

Casonato Matteo, matr. 1227270

Midena Matteo, matr. 1227272

Stojkovic Danilo, matr. 1222399

*Email referente del gruppo:*

[matteo.midena@studenti.unipd.it](mailto:matteo.midena@studenti.unipd.it)





## Indice

<b>1 Abstract</b>	<b>2</b>
1.1 La palestra: FDA . . . . .	2
1.2 Idea principale . . . . .	2
<b>2 Analisi dei requisiti</b>	<b>3</b>
2.1 Analisi dell'utenza . . . . .	3
2.2 SEO . . . . .	3
2.3 Requisiti . . . . .	3
<b>3 Progettazione e Organizzazione del lavoro</b>	<b>5</b>
3.1 Ambiente di Lavoro . . . . .	5
3.2 Database . . . . .	5
3.3 Struttura e Funzionalità . . . . .	6
3.4 Classi Utente . . . . .	6
<b>4 Front-end</b>	<b>7</b>
4.1 Sezioni implementate . . . . .	7
4.2 HTML . . . . .	9
4.3 CSS . . . . .	10
4.4 Javascript . . . . .	12
<b>5 Accessibilità e Usabilità</b>	<b>14</b>
5.1 Contenuto . . . . .	14
5.2 Dispositivi . . . . .	14
5.3 Alt Immagini . . . . .	15
5.4 No JavaScript . . . . .	15
5.5 Screen reader . . . . .	15
5.6 Ipovedenza . . . . .	16
5.7 Contrasto colore . . . . .	16
5.8 Daltonismo . . . . .	17
5.9 No CSS . . . . .	17
5.10 ARIA tags . . . . .	18
5.11 Navigazione . . . . .	18
5.12 Tabelle . . . . .	18
5.13 Peso pagine . . . . .	19
<b>6 Implementazione lato back-end</b>	<b>20</b>
6.1 PHP . . . . .	20
<b>7 Distribuzione Ruoli</b>	<b>22</b>
<b>8 Estensioni Future</b>	<b>23</b>



## 1 Abstract

Volendo basare il progetto su un potenziale sito da usare nel mondo reale, il gruppo ha deciso di progettare il sito di una palestra di Padova che alcuni membri frequentano. Questo ci ha fornito abbastanza conoscenze e materiale sul luogo per concentrarci sulla produzione web.

### 1.1 La palestra: FDA

Fitness & Dance Academy è una palestra situata a Padova che attualmente non dispone di un sito che permetta agli utenti di interagirci, bensì delle semplici pagine social. Nel periodo attuale una fruizione dei contenuti e un sistema di prenotazione/iscrizione a distanza potrebbe avere un grande impatto sui guadagni di una struttura del genere.

### 1.2 Idea principale

Offrire un sito nel quale ci siano le informazioni principali reperibili (orari, posizione, corsi offerti) ma anche una sviluppata area privata nella quale sia possibile richiedere schede, prenotare sessioni di allenamento e altro (per i clienti) e aggiungere corsi, compilare schede, visualizzare le prenotazioni dei clienti (per i trainer).



## 2 Analisi dei requisiti

### 2.1 Analisi dell'utenza

Il pubblico al quale si rivolge questo progetto consiste in chiunque voglia accedere alla struttura o interagirci, l'area geografica non è quindi molto larga e si può presupporre un'interesse preesistente degli utenti e una conoscenza della lingua italiana. Essendo Padova una città universitaria un potenziale sviluppo futuro potrebbe consistere nell'implementazione anche della lingua inglese.

Il linguaggio da utilizzare non sarà troppo tecnico per facilitare l'avvicinamento di individui più estranei al mondo dell'allenamento.

Verrà rivolto un particolare riguardo per l'utilizzabilità e la grafica del sito per permettere a FDA di innalzarsi al di sopra della concorrenza di fronte a clienti indecisi.

I linguaggi scelti sono HTML5 e CSS3, ciò è motivato da un'utenza prevalentemente giovane, ciononostante ci si assicurerà che le pagine degradino elegantemente verso versioni meno recenti.

### 2.2 SEO

Nello sviluppo di un sito è importante assicurarsi che esso sia raggiungibile da chi utilizza motori di ricerca (l'obiettivo è quello di presentare FDA anche a coloro che hanno espresso ricerche più generiche).

Il posizionamento SEO è una statistica fondamentale per assicurarsi che le pagine abbiano un pubblico vasto.

Le keywords (accuratamente scelte) per ogni pagina e la description correlata permettono ai motori di ricerca di indicizzare correttamente FDA. Le parole che idealmente richiamano questa tematica sono ad esempio: palestra, allenamento, padova, gym, fitness, corso di ballo, ecc...

### 2.3 Requisiti

Lo sviluppo è iniziato dall'elencazione delle funzionalità che il gruppo desiderava implementare per poi successivamente passare alla creazione di un database sulla base di queste ultime. Definire per bene quali sono i requisiti di progetto è fondamentale per un'andamento più tranquillo dei lavori e una suddivisione dei compiti più efficiente.

Ecco quindi i requisiti individuati durante le prime riunioni:

- parte statica:
  - pagina home;
  - pagina corsi;
  - pagina su di noi;
- parte area privata:
  - login;
  - signup;
  - visualizzazione e modifica profilo;
- sezione cliente:



- prenotazione sessione;
- prenotazione/visualizzazione scheda esercizi;
- prenotazione corsi;
- sezione trainer:
  - visualizzazione sessioni prenotate;
  - visualizzazione schede richieste e loro creazione;
  - aggiunta e dettaglio corsi;
- sezione admin:
  - visualizzazione sessioni prenotate;
  - visualizzazione schede richieste e loro creazione;
  - visualizzazione globale stato corsi;



### 3 Progettazione e Organizzazione del lavoro

#### 3.1 Ambiente di Lavoro

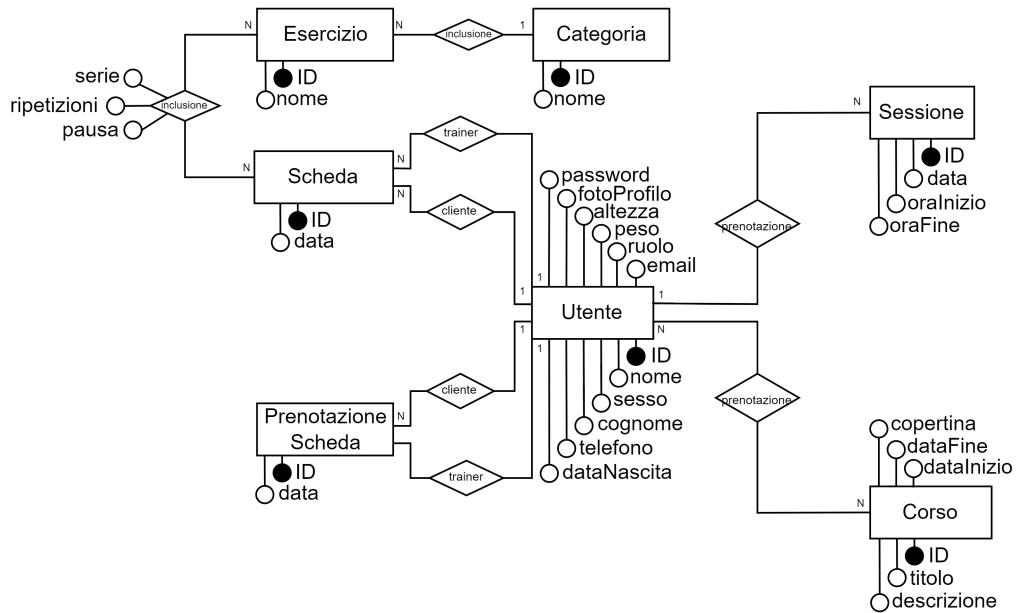
Per facilitare la collaborazione è stata creata una repository GitHub contenente ogni file del progetto. Questo strumento ha permesso ad ogni membro di lavorare in parallelo senza quasi mai causare conflitto con il lavoro altrui.

Inoltre è stato predisposto un container con Docker, all'interno del quale è stata svolta la maggior parte del lavoro al fine di utilizzare una versione comune di php e del database. Su questo container sono state fatte combaciare anche le versioni dei server dell'università indicate.

Un'altra piattaforma molto importante per il lavoro è stata Discord che ha ospitato la maggior parte delle ore di lavoro permettendo confronto immediato e motivazione per ognuno.

#### 3.2 Database

Una volta definite le funzionalità da implementare si è potuti passare alla costruzione di un database le cui entità rispecchiassero i requisiti, è stato naturalmente messo in forma normale. Qui di seguito è presente il suo diagramma E/R.

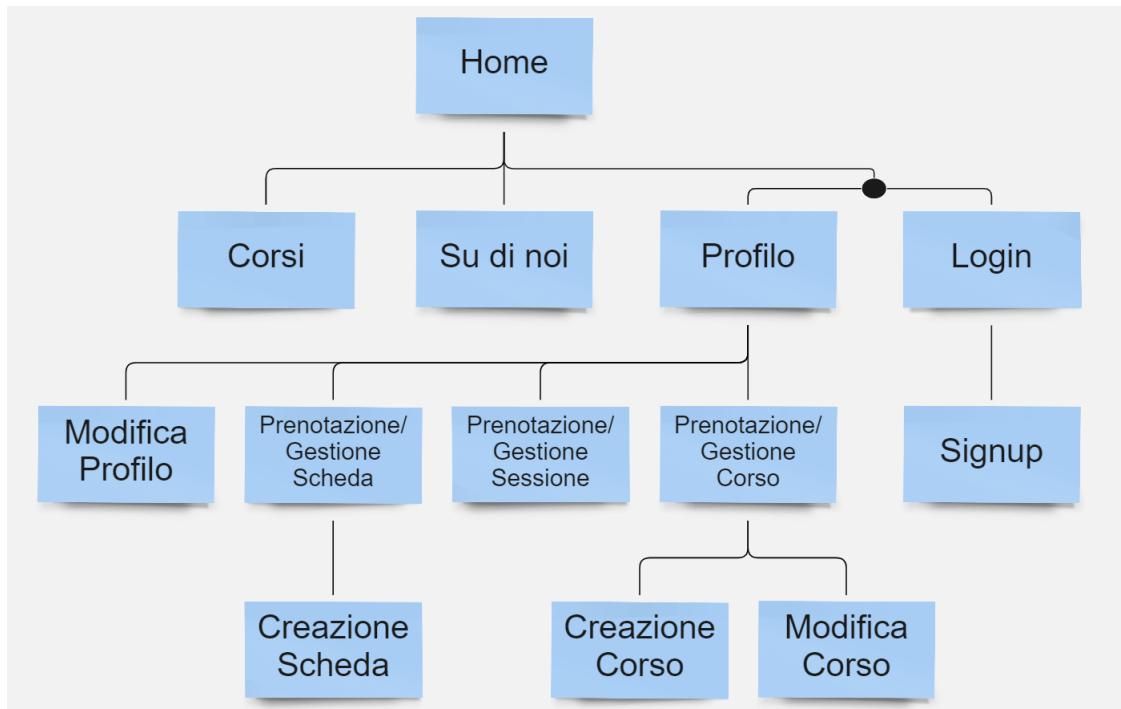


Ne è seguita la scrittura di un file *create\_db.sql* e il suo popolamento con tuple utili.



### 3.3 Struttura e Funzionalità

La struttura decisa è diretta conseguenza dei requisiti di progetto, la mappa del sito (strumento fondamentale per l'orientamento) è quindi così articolata:



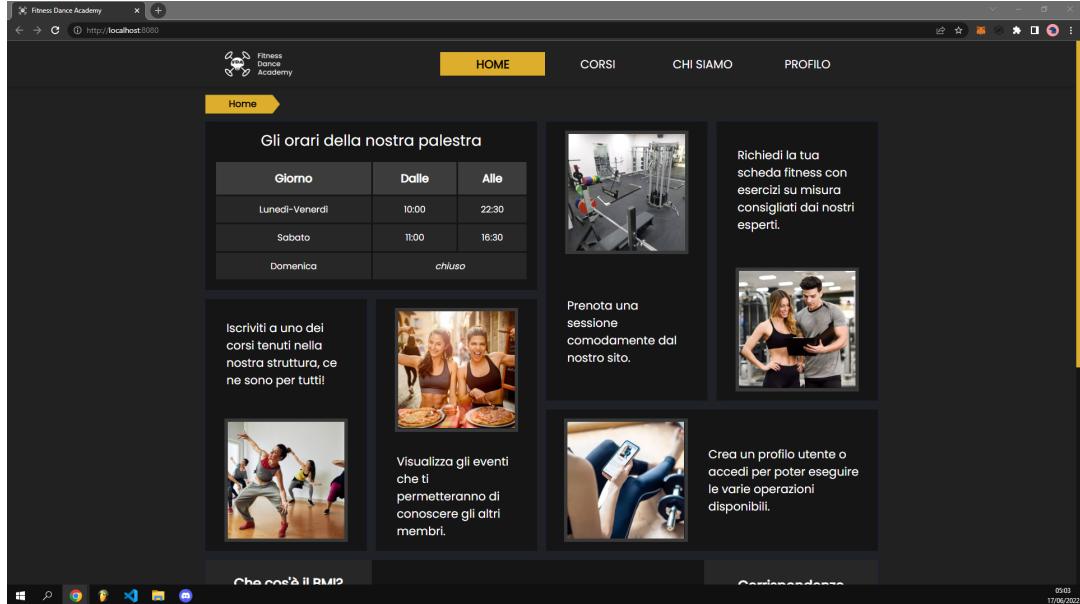
### 3.4 Classi Utente

Il sito verrà utilizzato da gruppi eterogenei di utenti che saranno distinti dalla tabella ruolo del database.

- ruolo=3 (Cliente): classe più numerosa, ad essi sono dedicate le varie pagine di prenotazione;
- ruolo=2 (Trainer): impiegati della palestra, predispongono corsi e schede per gli utenti base;
- ruolo=1 (Admin): gestiscono il sito web.



## 4 Front-end



### 4.1 Sezioni implementate

Le sezioni principali del sito web sono:

- **Navbar:** elemento costante nelle varie pagine, permette la navigazione, ed è composto dalle voci: HOME, CORSI, SU DI NOI, PROFILO.
- **Breadcrumb:** elemento dinamico delle varie pagine che descrive il percorso per mezzo del quale è stata raggiunta la posizione attuale per impedire il disorientamento e permettere all'utente di tornare sui suoi passi.
- **Footer:** elemento statico in fondo ad ogni pagina, contiene indirizzo, contatti telefonici, social e copyright.
- **Home Page:** è la pagina principale, include:
  - "Grid rappresentativa", dove sono presenti gli orari di apertura della palestra e le varie possibilità offerte dal sito;
  - "Calcolo BMI", in cui si possono leggere varie informazioni sul Body Mass Index ed è possibile calcolarlo.
- **Corsi:** pagina dedicata ai corsi offerti dalla struttura, si divide in:
  - "Orari dei corsi": consiste in una semplice tabella della distribuzione settimanale dei corsi;
  - "Offerta corsi": card con immagine e descrizione del corso recuperati dinamicamente dal database.
- **Su di noi:** semplice collezione di informazioni su FDA che permette agli utenti di conoscere qualche informazione in più sulla palestra.



- **Login:** punto nel quale è possibile effettuare l'accesso per gli utenti che hanno già un profilo, inserendo email e password.
- **Signup:** punto nel quale qualsiasi utente può creare un profilo, vengono richieste alcune semplici informazioni e gli input sono subito validati per facilitarne la correttezza e l'inserimento.
- **Profilo:** sezione utente nel quale vengono ripresentate le proprie informazioni anagrafiche; è possibile effettuare il logout o selezionare una delle voci disponibili:
  - "prenotazione corso (cliente)": visualizzazione dei corsi disponibili con bottone prenota associato e lista in forma tabellare dei corsi ai quali l'utente si è già iscritto;
  - "gestione corso (trainer)": qui è possibile per un trainer visualizzare i corsi di cui è responsabile con il relativo dettaglio dei clienti iscritti; inoltre è presente l'opzione per aggiungere un nuovo corso.
  - "creazione corso (trainer)": form da compilare per aggiungere un nuovo corso sul database del sito (il corso aggiunto verrà poi mostrato dinamicamente nella pagina dei corsi).
  - "prenotazione scheda (cliente)": visualizzazione delle eventuali schede prodotte dai trainer relative all'utente e breve form con il quale è possibile richiedere una nuova scheda ad uno dei trainer;
  - "gestione scheda (trainer)": visualizzazione di una tabella contenente le schede richieste al trainer con bottone tramite il quale è possibile raggiungere la pagina per la loro creazione;
  - "creazione scheda (trainer)": visualizzazione di un form per l'aggiunta degli esercizi e tabella nella quale sono ricapitolati gli esercizi inseriti finora;
  - "prenotazione sessione (cliente)": contiene un form per mezzo del quale è possibile prenotare una sessione di allenamento seguito da una semplice tabella elencante le sessioni già prenotate dall'utente;
  - "gestione sessione (trainer)": pagina nella quale è possibile verificare l'affluenza di clienti per una data selezionata, basata sulle prenotazioni di sessione effettuate;
  - "modifica profilo": pagina che permette di modificare i dettagli anagrafici inseriti al momento della registrazione.



## 4.2 HTML

La struttura delle pagine è stata scritta in HTML5, seppur rispettando la sintassi XML, aumentando la semantica del codice tramite un largo uso di semantica di nuova introduzione di questa versione (come nav o main). Diverse pagine sono di natura dinamica, in quanto variabili in base al contenuto del database e a quale utente ha effettuato l'accesso (le uniche pagine statiche sono: "home" e "su di noi"). Questo ha causato una considerevole quantità di codice HTML importato direttamente dagli script PHP, ciononostante nel momento della validazione ci si è assicurati che il risultato finale prodotto dal server fosse valido e passasse i test con validator.w3.org, di seguito: validazione della home.

The screenshot shows the W3C Markup Validation Service interface. The input area contains the HTML code for the home page. The validation results are displayed below, showing no errors or warnings. The status bar at the bottom of the window says "Document checking completed. No errors or warnings to show."

**Nota:** alcune pagine non passano la validazione per via dell'utilizzo dell'attributo *abbr* all'interno delle celle delle tabelle, è stato comunque scelto di tenere questi valori per ottenere una lettura più efficace ed espressiva da parte degli screen reader.

### Meta-Tag

Particolare attenzione è stata dedicata alla compilazione della sezione meta-tag dell'head, il gruppo ha infatti:

- scritto i *titoli* partendo dallo specifico al generale;
- utilizzato *description* per spiegare con cura il contenuto della pagina e permettere agli utenti di avere un'idea già dall'anteprima dei motori di ricerca;
- inserito nel campo *keywords* parole chiave con le quali ci attendiamo che un utente associa il nostro sito.

### Tabelle

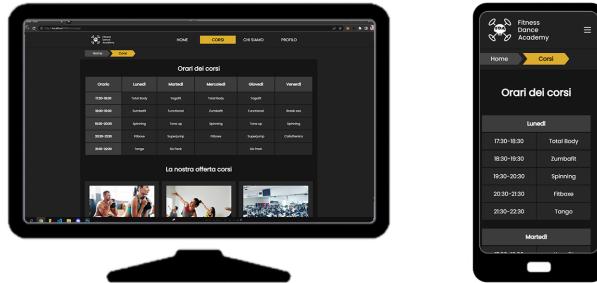
Diverse informazioni presentate sulle varie pagine sono in forma tabellare in quanto compatibili con la natura dai dati rappresentati (es: gli orari di aperture e dei corsi sono generalmente associati ad una rappresentazione tabellare).

Il gruppo si è assicurato di rispettare la sintassi HTML, durante la fase di sviluppo:

- suddivisione in thead, tbody e tfoot;
- utilizzo del tag caption;
- dichiarazione dello scope nei th per assicurare l'accessibilità;
- impiego dell'attributo colspan quando necessario;



Inoltre ogni tabella è comodamente visualizzabile da dispositivo mobile attraverso una trasformazione fluida, di seguito riportiamo un esempio.



Questa trasformazione si basa sull'utilizzo dell'attributo *data-title*.

### 4.3 CSS

La parte di presentazione in CSS3 è stata sviluppata in contemporanea con l'HTML essendo esse molto legate tra loro (i tag, gli id e le classi dichiarate nelle pagine HTML sono utilizzate per definire le regole di stile). Nella suddivisione del carico di lavoro chi scriveva una pagina ne curava anche l'aspetto estetico.

Utilizzare la versione più recente del linguaggio ci ha permesso di fare largo utilizzo delle sue novità come *flex-box* o *grid*.

Le regole di stile sono incluse nei file *style.css* e *mobile.css* (versione telefono) incorporati in ogni pagina.

#### Grid

La pagina principale è strutturata tramite CSS per mezzo di una grid, questo ha permesso una distribuzione più dinamica dei grid-item che si assemblano in modo particolare. Dal punto di vista della struttura gli elementi sono semplici div messi in successione ma poche regole di presentazione li hanno completamente ridistribuiti.

È stato scelto questo stile per la *home* per dare una panoramica semplice e non rigidamente ordinata dei servizi offerti dalla palestra. L'unica informazione a cui è data maggior importanza è la tabella che rappresenta gli orari di apertura che sarà chiaramente la prima informazione ad essere letta.

```
grid-container{
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-gap: 10px;
    grid-template-areas:
        "a a b c"
        "d e b f"
        "g g g g";
    margin: auto;
    margin-top: 0;
}
grid-container::content {
    background-color: transparent;
}
grid-item {
    display: flex;
    background-color: var(--transparent-dark);
    box-shadow: 0 0 10px var(--color-dark);
    margin-bottom: 0em;
}
grid-item-a {
    width: 100px;
    flex-direction: column;
    grid-area: a;
}
grid-item-b {
    width:100px;
    flex-direction: column;
    grid-area: b;
}
grid-item-c {
    width:100px;
    flex-direction: column;
    grid-area: c;
}
grid-item-d {
    width:100px;
    flex-direction: column;
    grid-area: d;
}
grid-item-e {
    width:100px;
    flex-direction: column;
    grid-area: e;
}
grid-item-f {
    width:100px;
    flex-direction: column;
    grid-area: f;
}
grid-item-g {
    width:100px;
    flex-direction: column;
    grid-area: g;
}
```

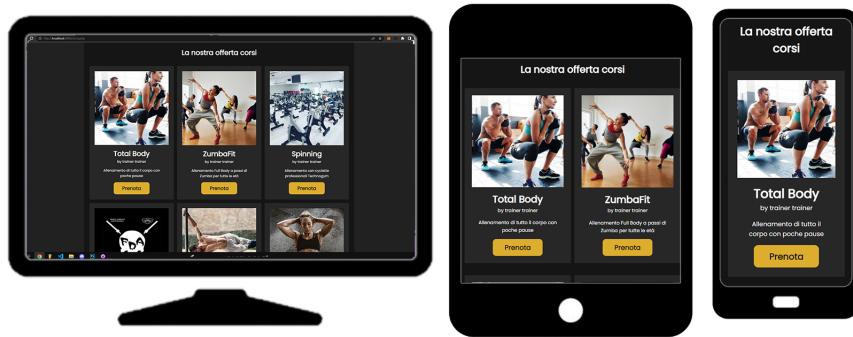


## Flex

Le flexbox sono state uno strumento fondamentale durante lo sviluppo dell'aspetto della maggior parte delle pagine, soprattutto nelle occasioni in cui è stato necessario allineare diversi elementi in verticale.

In particolare la proprietà `flex-wrap` ha assicurato un ambiente responsive nel caso dell'utilizzo da cellulare.

L'impiego di flexbox può essere riscontrato nel menù di navigazione, nelle breadcrumb, nel footer e in molti altri punti. Prendiamo come esempio la pagina corsi, le cui card occupano dinamicamente lo schermo in orizzontale.



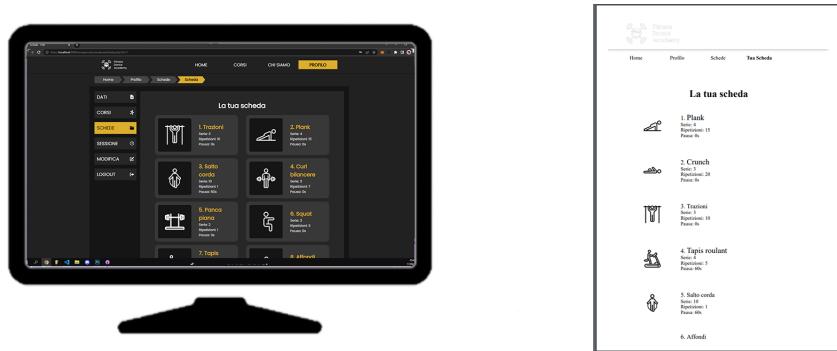
## print.css

Nel caso in cui l'utente voglia stampare alcune pagine per avere una copia fisica delle informazioni è stato predisposto un file `print.css` contenente delle regole specifiche per la carta stampata.

Tra gli accorgimenti di queste regole si citano:

- impostazione di un font con grazie (Times New Roman);
- rimozione di bottoni non utili alla comprensione della pagina stampata e del menu;
- inclusione delle breadcrumb.

Poichè la stampa generalmente viene eseguita in verticale, lo sviluppo degli accorgimenti per mobile ha facilitato anche la versione stampata. Riportiamo di seguito un esempio:





## Elementi Standard

Per alleggerire il codice css si è cercato di definire delle classi standard da utilizzare in diversi punti, per diminuire il numero di regole e fornire più uniformità tra le pagine del sito, riportiamo un esempio di questa scelta relativamente alla classe *button*.

```
/*BUTTON*/
700 .button {
701   font-size: 1.6em;
702   font-weight: bold;
703   border:none;
704   display: block;
705   text-align: center;
706   border-radius: 8px;
707   padding: 10px 20px;
708   background-color: transparent; /*background-color: var(--color-transparent-dark);*/ 709   font-family: inherit;
710   margin: 10px 0;
711   margin-top: 1.5em;
712   padding: 0.5em 1em;
713   transition: background-color 0.25s;
714   text-align: center;
715 }

716 .button blue {
717   color: white; /*background-color: var(--color-blue-light);*/ 718   color: #007bff;
719 }
720 .button blue-hover {
721   background-color: var(--color-blue-verylight);
722   color: var(--color-blue);
723 }

724 .button button-purple {
725   color: white; /*background-color: var(--color-purple);*/ 726   color: #6f42c1;
727 }
728 .button button-purple-hover {
729   background-color: var(--color-purple);
730   color: white;
731 }
732 .button button-transparent {
733   background-color: transparent; 734   color: #007bff;
735 }
```

## Unità di misura

Le dimensioni dichiarate nelle regole sono quasi interamente specificate per mezzo di unità di misura relative (em e %) essendo consapevoli del fatto che la precisione "al pixel" è irraggiungibile per via della presenza di diversi browser, sistemi operativi e dispositivi stessi... Questo metodo di lavoro ha facilitato l'adattività del sito nella sua versione mobile o comunque al variare della dimensione dello spazio disponibile.

## Pseudoelementi

Talvolta è stato fondamentale l'utilizzo dei pseudoelementi css come *:first-child* (per aumentare il font del nome nella lista di informazioni della pagina profilo) o *:before*(per inserire i data-title nelle tabelle mobile).

## 4.4 Javascript

Il linguaggio JavaScript è stato utilizzato principalmente per la validazione input lato client e per le chiamate AJAX al database.

### Validazione Input Front-end

Queste funzioni JavaScript permettono all'utente di avere feedback immediato sulla correttezza degli input dei form e alleggeriscono le attività di validazione del server (che eseguirà comunque una validazione nel caso in cui JavaScript sia disattivato o un utente malevolo cercasse di raggirarlo).

Queste funzioni non coprono ovviamente tutti i casi non avendo a disposizione il database (login non può controllare la presenza o meno di un'iscrizione effettuata per una certa email, sessione non può verificare che non sia già stata prenotata in quell'orario).

Ciononostante è stato compiuto un lavoro preciso di filtraggio degli input prima ancora che richiedano l'intervento del server, ad esempio sulla pagina *prenotazione\_sessione* la giornata non può essere passata, gli orari devono essere entrambi compresi nella fascia oraria di apertura della palestra per il giorno specificato e quello di fine non può precedere quello di inizio.

Gli input sono collegati alla validazione per mezzo delle keyword *onchange* o *onblur*.



## Calcolo BMI

Il calcolo del BMI è una delle funzionalità offerte dalla pagina home la quale non prevede elaborazione lato server (oltre all'inserimento del footer dinamicamente). La preparazione del risultato è infatti fatta esclusivamente lato client da una funzione che valida gli input ed esegue il calcolo per poi stamparlo sulla pagina.

**Che cos'è il BMI?**

Il Body Mass Index è un sistema di valutazione del peso attraverso la soluzione di una formula che richiede due valori noti (altezza e peso) che permette di stabilire la classe di appartenenza dell'utente. Il posizionamento nella classe peso-forma è statisticamente collegato al minor rischio di ammalarsi di patologie metaboliche e non solo.

**Calcola il tuo BMI!**

Altezza: 184 cm

Peso: 70 kg

**Calcola**

Risultato: 20.68 (peso forma)

**Corrispondenze**

Classe	Valore BMI
sottopeso	$x \leq 18.5$
pesoforma	$18.5 < x \leq 25$
sovrapeso	$25 < x \leq 30$
obesità	$30 < x \leq 35$
obesità severa	$x \geq 35$

## Menu Mobile

Per l'implementazione del menu di navigazione da telefono si è scelto un modello "ad hamburger" che è certamente noto agli utenti per la sua popolarità e che ha il pregio di non occupare troppo spazio e non distrarre quando non viene utilizzato. Questo comporta un click "di profondità" in più che il gruppo ha deciso di accettare.



## AJAX

Abbreviazione di Asynchronous JavaScript and XML, viene utilizzato in pagine dinamiche per raggiungere il back-end (in questo caso eseguire codice PHP) senza la necessità di dover effettuare un refresh della pagina. L'implementazione di alcune funzionalità si basano sul suo utilizzo. Ad esempio questa funzionalità è stata usata per la cancellazione di una sessione prenotata; la funzione predisposta comunica al server l'intenzione di cancellare una tupla chiamando un file php che si occupa della rimozione del record dal database mentre il codice JavaScript si occupa di cancellare la riga dalla tabella html senza ricaricare la pagina.



## Modifiche non salvate

Un ultimo esempio di codice JavaScript implementato, consiste nello script *checkUnsavedChanges.js* che in alcune pagine si assicura che l'utente non perda dei dati che ha inserito/modificato ma non ha ancora confermato.

# 5 Accessibilità e Usabilità

Abbiamo posto particolare attenzione all'accessibilità del sito sia per non precludere l'utilizzo a nessun potenziale cliente, sia per sottolineare l'importanza di questo aspetto troppo spesso trascurato nello sviluppo web.

Il maggior numero di classi utente possibile è stato considerato per mezzo dei seguenti test effettuati.

## 5.1 Contenuto

Un importante lato dell'accessibilità è l'utilizzo di un linguaggio semplice e facilmente comprensibile. Per la comprensione dei contenuti non è necessario possedere un lessico complesso e la lunghezza delle frasi si limita al minimo indispensabile per risultare comprensibile e utile.

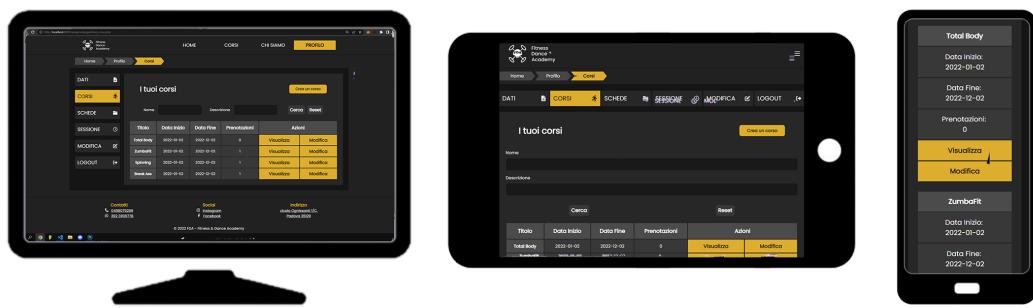
La lingua utilizzata è l'italiano ma nel mondo dell'allenamento alcuni esercizi o attrezzi sono conosciuti per il loro nome straniero, ogni caso di parola in altra lingua è stato associato dal corretto tag *span* con gli attributi *xml:lang* e *lang*.

Alcuni esempi di parole soggette a questo trattamento sono stretching, cyclette, fitness, tapis roulant...

## 5.2 Dispositivi

Una delle difficoltà di un sito web consiste nel fatto che non è possibile conoscere a priori il dispositivo sul quale verrà visualizzato, si deve cercare di prevedere ogni schermo, ogni browser, ogni sistema operativo per mantenere l'accessibilità in ogni variante.

Il design reattivo alle modifiche di ambiente è assicurato da regole CSS specifiche incluse in media-query legate ad una dimensione massima dello schermo; in particolare per questo progetto ci sono fondamentalmente due punti di rottura: 1200px (finestra desktop ristretta/tablet) e 600px (schermo del telefono e piccoli tablet).





Di seguito la serie di test effettuati suddivisi per categoria:

- Browser:

- Chrome;
- Brave;
- Edge;
- Safari;
- Firefox.

**Nota:** non sono stati eseguiti test su Internet Explorer in quanto largamente inutilizzato e non supportato più neanche da Windows stesso che consiglia il passaggio ad Edge.

- Dispositivi (con SO):

- Desktop (Windows 10);
- Desktop (Linux Ubuntu);
- Desktop (MacOS Monterey): solo per breve durata poiché il gruppo non ha la disponibilità fisica del dispositivo;
- Mobile (Android);
- iPad (iOS).

### 5.3 Alt Immagini

Ogni immagine significativa e non puramente decorativa è stata associata ad un attributo *alt* non vuoto, nel quale è presente una dettagliata descrizione del contenuto. Questo accorgimento aiuta molto chi non basa la sua navigazione sulla vista ma sull'udito.

### 5.4 No JavaScript

Alcuni utenti scelgono di disattivare il codice JavaScript ed è importante assicurare le funzionalità principali anche in sua assenza, infatti la validazione degli input non si limita al front-end ma viene eseguita anche nel codice php.

### 5.5 Screen reader

I test su screen reader vanno fatti per assicurare un'esperienza soddisfacente a coloro che non hanno il senso della vista. Questi software trasmettono via audio il contenuto del sito, ma richiedono accorgimenti nella scrittura del codice html, ad esempio, in questo sito si fa largo uso dell'attributo *role*, di *aria labels*, le tabelle sono state semplificate per gli screen reader ed è presente un bottone nascosto "skip to content" che permette di saltare direttamente al contenuto evitando un'eventuale rilettura di navbar e breadcrumb.

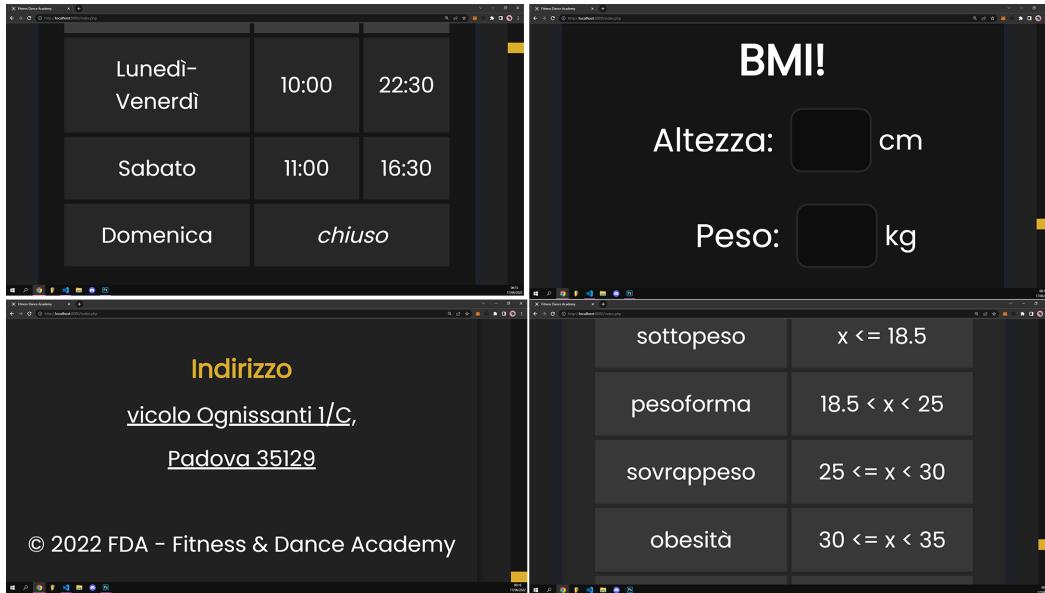
Premere invio per saltare al contenuto

Gli screen reader sui quali sono stati effettuati i vari test sono, *Orca* di Linux Ubuntu e quello incorporato in silktide.com.



## 5.6 Ipovedenza

L'ipovedenza è una condizione che influenza negativamente l'utilizzo di siti web. Questa classe di utenti è stata tutelata del css adattivo e dai test effettuati impostando lo zoom del browser a 500%.



## 5.7 Contrasto colore

Nella scelta della palette colori è molto importante verificare che colori sovrapposti abbiano confini definiti e perciò anche che il loro contrasto sia sufficiente ( $>3.5$ ).

Queste considerazioni sono state fatte fin da subito, durante la scelta dei colori principali da applicare al sito.

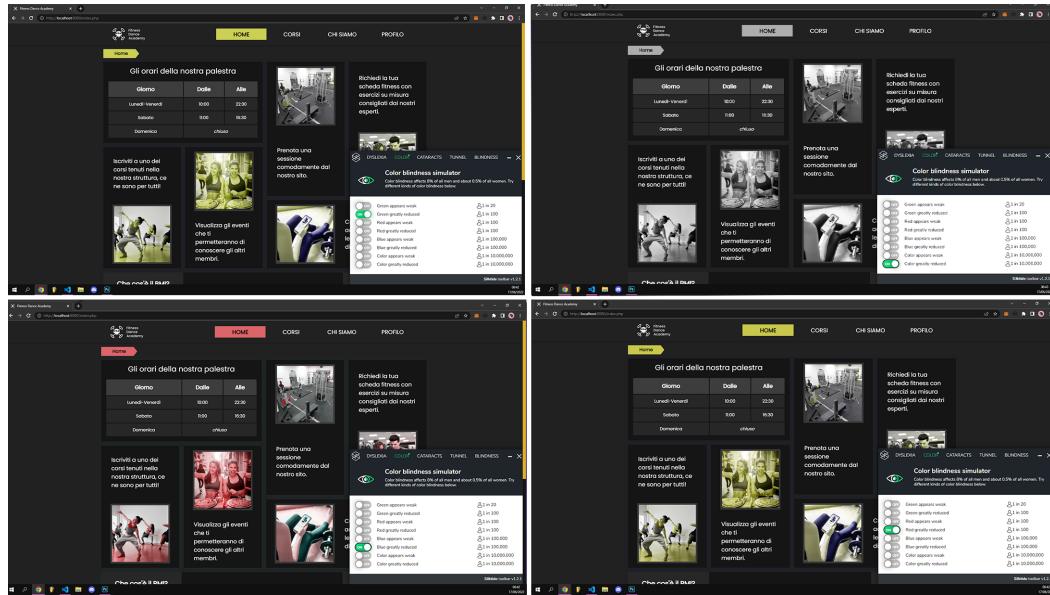
Nel momento della validazione di questo aspetto, l'utilizzo di colori con opacità inferiore a 100% ha reso più difficoltoso il processo, infatti i validatori come quello WCAG confrontano ignorando la trasparenza e perciò segnalano falsi positivi. Per il calcolo del contrasto tra singoli colori è stato utilizzato [webaim.org/resources/contrastchecker/](http://webaim.org/resources/contrastchecker/). Di seguito l'esempio del colore principale del testo sullo sfondo più chiaro, bottone e link visitato.

Text Type	Color Scheme	Contrast Ratio	WCAG AA	WCAG AAA
Normal Text	White text on black background	11.03:1	Pass	Pass
Large Text	White text on yellow background	10.16:1	Pass	Pass
Graphical Objects and User Interface Components	White text on orange background	5.82:1	Pass	Fail



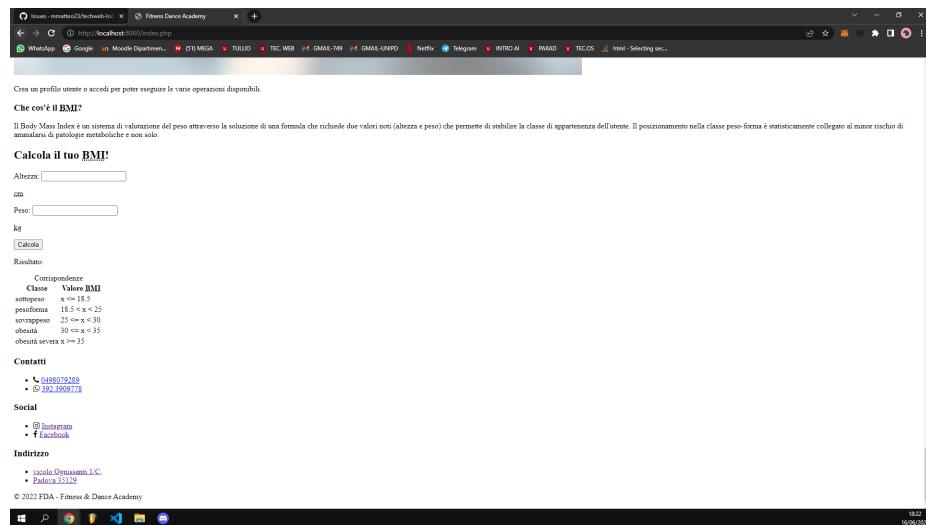
## 5.8 Daltonismo

L'utilizzabilità del sito è stata verificata anche in assenza di colori per assicurare accessibilità anche a chi soffre di daltonismo. Anche in questo caso *silk tide* è risultato utile per testare le pagine.



## 5.9 No CSS

In assenza dei fogli di stile abbiamo testato che il sito sia in grado di presentare nella sua interezza i contenuti della struttura in modo sensato. Ecco uno screen di esempio.





## 5.10 ARIA tags

Gli ARIA tags sono attributi inclusi nel codice HTML che permettono di rendere la pagina usufruibile per ogni classe di utente. Ad esempio *aria-hidden="true"* indica allo screen reader che quel tag non è da leggere. Un esempio di utilizzo di questi tags all'interno del sito, sono quelli utilizzati per evitare che venissero lette le icone nel menu dell'area privata, che non avrebbero aggiunto nessuna ulteriore informazione e avrebbero solo rallentato la lettura dello screen reader. Un altro esempio differente sono gli *aria-label="nome"* che identificano un particolare tag (sono stati utilizzati nel menu ad hamburger).

## 5.11 Navigazione

Per evitare il disorientamento dell'utente è fondamentale che la presentazione del sito risponda a tre domande fondamentali: dove sono? dove posso andare? di cosa si tratta? Per rispondere a questi quesiti abbiamo posto attenzione sul logo, le breadcrumb ed il menù; tutti questi elementi sono posizionati nei punti "standard", ovvero dove un utente è abituato a vederli.

## 5.12 Tabelle

Le tabelle sono state rese accessibili, sia dal punto di vista dell'adattamento a schermi differenti sia nell'utilizzo corretto della sintassi html (suddivisione in thead-tbody-tfoot, attributi scope, colspan dove è necessario).

Titolo	Data Inizio	Data Fine	Prenotazioni	Azioni	
Total Body	2022-01-02	2022-12-02	0	<a href="#">Visualizza</a>	<a href="#">Modifica</a>
ZumbaFit	2022-01-02	2022-12-02	1	<a href="#">Visualizza</a>	<a href="#">Modifica</a>
Spinning	2022-01-02	2022-12-02	1	<a href="#">Visualizza</a>	<a href="#">Modifica</a>
Break Ass	2022-01-02	2022-12-02	1	<a href="#">Visualizza</a>	<a href="#">Modifica</a>

```
<table class="table-prenotazione lista corsi full-button">
<thead>
</thead>
<tbody>
<tr>
<td><span unescape="en" lang="en">Total Body</span></td>
<td><td><span>Data Inizio:</span><span>2022-01-02</span></td>
<td><td><span>Data Fine:</span><span>2022-12-02</span></td>
<td><td><span>Prenotazioni:</span><span>0</span></td>
<td><a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Visualizza</a>
<a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Modifica</a>
</a></td>
</td>
</tr>
<tr>
<td><span unescape="en" lang="en">ZumbaFit</span></td>
<td><td><span>Data Inizio:</span><span>2022-01-02</span></td>
<td><td><span>Data Fine:</span><span>2022-12-02</span></td>
<td><td><span>Prenotazioni:</span><span>1</span></td>
<td><a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Visualizza</a>
<a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Modifica</a>
</a></td>
</td>
</tr>
<tr>
<td><span unescape="en" lang="en">Spinning</span></td>
<td><td><span>Data Inizio:</span><span>2022-01-02</span></td>
<td><td><span>Data Fine:</span><span>2022-12-02</span></td>
<td><td><span>Prenotazioni:</span><span>1</span></td>
<td><a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Visualizza</a>
<a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Modifica</a>
</a></td>
</td>
</tr>
<tr>
<td><span unescape="en" lang="en">Break Ass</span></td>
<td><td><span>Data Inizio:</span><span>2022-01-02</span></td>
<td><td><span>Data Fine:</span><span>2022-12-02</span></td>
<td><td><span>Prenotazioni:</span><span>1</span></td>
<td><a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Visualizza</a>
<a href="#" class="button button-purple button-filter" data-cs="2" data-kind="parent"><a href="#">Modifica</a>
</a></td>
</td>
</tr>
</tbody>
</table>
```



### 5.13 Peso pagine

Il gruppo ha avuto un occhio di riguardo anche nel cercare di minimizzare le dimensioni di ogni pagina al fine di velocizzare il caricamento e ridurre il consumo di dati. Ogni pagina è stata tenuta sotto 1MB.

In particolare la pagina più pesante risulta essere la pagina dei "Corsi" con 771KB, mentre il peso del resto delle pagine risulta essere quasi sempre inferiore ai 200KB.



## 6 Implementazione lato back-end

### 6.1 PHP

La parte dinamica del sito è stata codificata utilizzando PHP utilizzando la tecnica dei placeholder html per inserire contenuti basati su ciò che c'è nel database. L'area privata è molto legata agli input utente e fa quindi molto uso di codice PHP per renderla dinamica e il più possibile adattiva. La parte generale invece (Home, Corsi, Chi siamo) lo usa solo in parte. Per facilitare e seguire una linea il più possibile comune per la scrittura del codice PHP è stata adottata la scelta di associare a ogni tabella significativa (entità) una classe specifica che ne racchiude i metodi principali (create, validator, delete...) per la relativa interazione. Oltre ai documenti PHP legati ad ogni pagina, sono presenti anche:

- *db.php*: racchiude le principali funzioni di accesso al db e per l'esecuzione delle query;
- *deletePrenotazioneScheda.php* e *deleteSession.php*: script chiamati dalle funzioni AJAX;
- *AuthController.php*: che verifica l'autenticazione utente;
- *logout.php*: cancella la sessione utente;
- *utilities.php*: racchiude le funzioni php che vengono utilizzate su più pagine;
- *validSession.php*: verifica che l'utente sia loggato.

#### Gestione della sessione utente

L'array globale `$SESSION` è fondamentale in qualsiasi sito web che offre la possibilità di login e autenticazione all'utente. Il suo utilizzo permette il riconoscimento dell'utente autenticato e di gestire dinamicamente molte pagine e operazioni, ad esempio:

- nella scelta dei link da offrire all'utente nell'area privata (trainer/cliente);
- per identificare quali tuple eliminare dal database;
- gestire le operazioni in base al ruolo dell'utente autenticato.

#### Validazione input utente

Oltre ai controlli via JavaScript sul front end, anche nel back-end sono stati replicati i controlli necessari sull'input, tramite la codifica delle funzioni di validazione (validator). La validazione front-end è infatti replicata in PHP con la copertura dei test non mancanti (ad esempio verificare che una email non sia già occupata). Questo permette di essere molto più elastici agli input utente e di avere una protezione ulteriore che non si può fermare solamente ai controlli via javascript, facilmente superabili o che possono essere addirittura disattivati.

#### Protezione codice malevolo

La funzione `preventMaliciousCode()` impedisce input utente nocivi provenienti da attacanti malintenzionati eliminando i tag HTML con la funzione `preg_replace()` e prevenendo l'inserimento di apici con `htmlspecialchars()` (particolarmente importante per prevenire SQL injection).

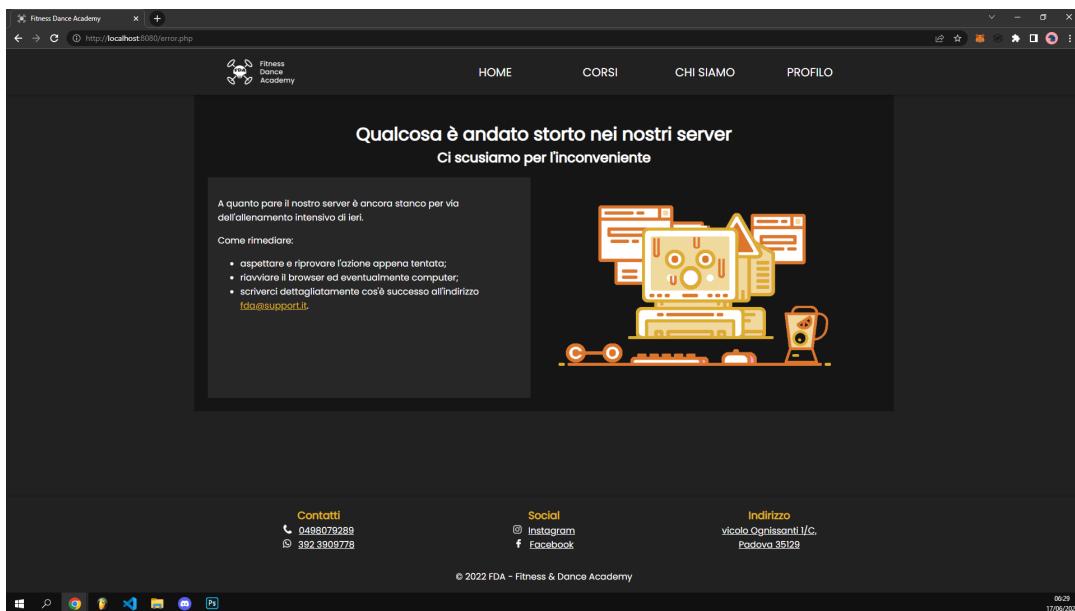


## Server Errors

Quando un errore è causato da un mancato riscontro dei nostri sistemi e non a causa di un errato input utente, questi viene riportato ad un'unica pagina di errore che fornisce consigli utili su come risolvere.

Inizialmente l'obiettivo di questa pagina consisteva nel fungere da punto di redirect degli errori 404, ma il redirect registrato nel file .htaccess non sembra funzionare correttamente.

Si è scelto inoltre di creare un'immagine simpatica che distraiga positivamente l'utente da una situazione che molto probabilmente non è positiva. In questo caso il soggetto scelto (maschotte) è un computer, che rappresenta i nostri server, che è affaticato dall'allenamento svolto in palestra.





## 7 Distribuzione Ruoli

Ogni membro del gruppo ha partecipato proficuamente ad ogni aspetto del sito, è stato quindi difficile presentare i singoli contributi senza andare troppo nello specifico e che rappresentano la reale distribuzione del lavoro.

- Matteo Casonato:
  - css desktop;
  - pagine e logica legate all'utente;
  - gestione sessione utente;
- Matteo Midena:
  - architettura php;
  - pagine e logica legate ai corsi;
  - javascript
- Danilo Stojkovic:
  - css mobile;
  - pagine e logica legate alle schede a alla sessione;
  - relazione;



## 8 Estensioni Future

Il gruppo ritiene che il progetto sia completo di tutte le funzioni fondamentali che deve avere un gestionale di una palestra. Sono state individuate comunque delle sezioni del sito che potrebbero essere estese e migliorate in futuro, che si è deciso di non affrontare per non eccedere in complessità e deviare dallo scopo principale del progetto. Riportiamo qui un elenco di possibili estensioni che sono state individuate e la relativa idea associata:

- Aumento delle funzionalità del ruolo **admin**: attualmente un utente admin ha delle funzioni che non discostano molto da quelle di un trainer (nel caso del progetto si è deciso di espandere maggiormente il ruolo del trainer perché direttamente collegato ai clienti e alla relativa interazione). Alcune delle funzionalità di cui un admin potrebbe essere fornito sono: creazione di utenti trainer, gestione più complessa dei clienti e modifica degli orari della palestra.
- Aumento delle funzionalità legate alle **schede**: attualmente la gestione di una scheda è più che sufficiente per la tipologie di allenamenti che la palestra (target del progetto) prepara per i propri clienti. Se un giorno si volesse ampliare questo servizio si potrebbe provvedere la possibilità di creare nuovi esercizi da quelli base, organizzare uno storico delle schede fornite e magari anche la possibilità da parte di un utente di rilasciare un feedback.

Per lo svolgimento del progetto si è deciso inoltre di non permettere la cancellazione di un utente trainer, questo per evitare conflitti con i corsi e le schede associate. Il gruppo ritiene infatti importante la possibilità di non cancellare un utente trainer in quanto entità principale all'interno della palestra; in caso di una futura necessità di escludere un trainer (ad esempio per un licenziamento, malattia prolungata, ...) un'opzione a cui il gruppo ha pensato è quella di gestire uno stato, in modo tale da identificare i trainer come *attivi* o *inattivi*, così facendo si potrà mantenere uno storico dei corsi tenuti e dei dipendenti assunti.