

# Relazione Tecnologie Web

## Informazioni sul documento

Nome documento	Relazione Tecnologie Web
Versione	1.0.0
Data redazione	08/01/2020
Membri	Francesco Bari Ivan Furlan Francesco Pecile Lin Zhaohui
Users	admin@admin.com ( <b>password:</b> admin) user@user.com ( <b>password:</b> user) user2@user.com ( <b>password:</b> user)

## Informazioni

Indirizzo sito web: <http://tecweb1920.studenti.math.unipd.it/fbari/>

Email referente gruppo: [francesco.bari.2@studenti.unipd.it](mailto:francesco.bari.2@studenti.unipd.it)

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Abstract . . . . .	3
1.2	Analisi d'utenza . . . . .	3
1.3	Interazioni utente . . . . .	3
<b>2</b>	<b>Organizzazione del lavoro e sviluppo</b>	<b>5</b>
2.1	Progettazione . . . . .	5
2.2	Design . . . . .	5
2.2.1	Header . . . . .	6
2.2.2	Breadcrumbs . . . . .	6
2.2.3	Contenuto . . . . .	6
2.2.4	Footer . . . . .	8
2.2.5	Database . . . . .	9
<b>3</b>	<b>Implementazione lato front-end</b>	<b>10</b>
3.1	HTML . . . . .	10
3.2	CSS . . . . .	10
3.2.1	CSS desktop . . . . .	10
3.2.2	CSS mobile . . . . .	11
3.2.3	CSS stampa . . . . .	12
3.3	JavaScript . . . . .	13
3.4	Compatibilità con browser datati . . . . .	14
<b>4</b>	<b>Accesibilità</b>	<b>15</b>
4.1	Screen reader . . . . .	15
4.2	Colori . . . . .	16
4.3	Test . . . . .	17
<b>5</b>	<b>Implementazione lato back-end</b>	<b>18</b>
5.1	PHP . . . . .	18
<b>6</b>	<b>Ruoli</b>	<b>21</b>

## Elenco delle figure

1	Header . . . . .	6
2	Breadcrumbs utente non registrato . . . . .	6
3	Breadcrumbs utente registrato . . . . .	6
4	Esempio del contenuto di una pagina, in questo caso <i>Citologia Nasale</i>	7
5	Footer . . . . .	8
6	Esempio di una pagina mobile, in questo caso <i>Notizie</i> . . . . .	11
7	Esempio di una pagina di stampa, in questo caso <i>Notizie</i> . . . . .	12
8	<i>Color greatly reduced</i> . . . . .	16
9	<i>Blue greatly reduced</i> . . . . .	16

# 1 Introduzione

## 1.1 Abstract

Il progetto scelto dal gruppo ha come scopo la realizzazione di un sito dedicato ad un medico professionista di nome Marco Donati. Essendo un otorino, si specializza nei disturbi auditivi, posturali e nasali.

L'obiettivo della pagina è quello di far conoscere le capacità del dottore e metterlo in contatto con i propri pazienti. Il sito web offre quindi quattro funzionalità principali:

- La possibilità di creare un account privato con il quale inviare messaggi privati tra l'utente e il dottore;
- Una funzione di prenotazione visita online;
- Le informazioni sulla locazione dello studio in cui saranno svolti i vari servizi medici disponibili e tutte le informazioni relative ad essi;
- Una pagina nella quale verranno mostrate varie news, di tipo organizzativo piuttosto che informativo o di carattere divulgativo.

## 1.2 Analisi d'utenza

Il target del sito si concentra su uomini e donne di qualunque età in cerca di un consulto medico o di una prestazione medica specialistica. Data l'ampiezza d'utenza potenzialmente molto variegata, il numero di utenti non avvezzi alla navigazione web potrebbe essere elevato, quindi si è deciso di sviluppare un design molto semplice ed efficace. Le informazioni sono quindi fornite tramite l'utilizzo di un linguaggio informale e comprensibile, restando però nei limiti di comprensione imposti dal campo medico. Il sito infatti mantiene comunque la professionalità e la precisione nei contenuti che si aspetterebbe un medico in visita nel sito.

Le informazioni fornite dal nostro sito sono di carattere puramente informativo, pensate per dare un'idea all'utente prima che incontri direttamente il dottore per spiegazioni molto più dettagliate. Questa caratteristica fa sì che non sia necessaria una barra di ricerca, poiché la gran parte delle informazioni più complesse sarà data a voce o tramite il servizio di consulti online. Si è pensato quindi di non doverla aggiungere in quanto l'espressività della barra di navigazione e delle sue sottosezioni copre completamente tutti i contenuti e le necessità dell'utente.

## 1.3 Interazioni utente

Abbiamo individuato 3 tipi di utente:

- Utente non registrato;
- Utente registrato;
- Admin.

L'utente non registrato può accedere a tutta la parte non interagibile del sito, ma può comunque informarsi su tutto il contenuto chiave senza doversi registrare.

L'utente registrato sblocca le funzionalità cardine del nostro servizio: la possibilità di prenotare un appuntamento tramite il nostro form online e consultare



direttamente il dottore in modo privato. Inoltre dopo aver prenotato una visita, l'utente può visualizzarne privatamente le informazioni a riguardo, così da poterle anche stampare per documentarsi. Questa funzionalità è valida anche per tutte le conversazioni private con il dottore, le quali vengono mostrate tramite chat sullo schermo e possono essere eventualmente stampate.

L'admin del sito è lo stesso dottor Marco Donati, il quale ha poteri diversi dal semplice utente registrato. Esso può:

- Aggiornare la pagina notizie, eliminare quelle vecchie e scriverne di nuove;
- Rispondere alle domande degli utenti registrati e visualizzare un elenco delle chat con tutti gli utenti;
- Visualizzare una lista di tutte le prenotazioni delle visite, divise per giorno e tipologia.

## 2 Organizzazione del lavoro e sviluppo

### 2.1 Progettazione

Nella fase iniziale si sono individuate capacità e punti di forza dei membri del gruppo. In seguito a ciò si è deciso di non affidare a ciascuno una parte ben precisa del progetto ma di svolgerlo nella sua interezza in gruppo, in modo da mantenere la coerenza del codice e delle idee su tutto il piano lavorativo. Questa decisione inoltre garantisce che ogni membro del gruppo abbia lavorato su tutte le parti principali del progetto. Abbiamo deciso di svolgere il lavoro in parallelo con le lezioni del corso, usando quindi una strategia di progettazione classica:

1. Codifica pagine HTML;
2. Codifica pagine CSS;
3. Creazione e popolamento Database;
4. Codifica JavaScript;
5. Codifica pagine PHP.

Punto fondamentale degno di nota è l'accessibilità. Essa non è posta nell'indice della progettazione appena descritto, poiché sono stati fatti test continui ad ogni passo della progettazione. Così facendo, ogni blocco di lavoro compiuto risulta accessibile e valido. Questo ci ha consentito di non dover tornare sui nostri passi in fasi più avanzate del progetto, ma poter contare su una base di lavoro effettuato in modo costante.

Nello specifico sono state progettate le pagine dedicate alla due funzionalità più importanti del sito: la prenotazione della visita e il consulto online. I punti cardine di queste due pagine sono:

- **Confidenzialità:** le informazioni relative alle prenotazioni e tutte le conversazioni tra dottore e utente possono essere accedute solo dai diretti interessati. Questa caratteristica è garantita dal fatto che l'accesso alle due pagine di prenotazione e consulti è consentito solo agli utenti registrati;
- **Semplicità:** la pagina di consulti online utilizza un singolo form sul quale poter scrivere il proprio messaggio. Una volta fatto ciò la conversazione viene mostrata secondo i canoni standard della messaggistica online. La pagina prenota visita invece è composta da diversi form. In essi è possibile scegliere giorno, mese, anno e tipologia di prestazione. Controllando la disponibilità della data scelta sarà poi possibile visualizzare e quindi scegliere la fascia oraria libera più consona alle proprie esigenze orarie.

### 2.2 Design

Abbiamo deciso di separare completamente la parte statica da quella dinamica, in modo da poter rendere semplice l'aggiornamento del sito in qualunque momento. La struttura di ogni pagina web è templetizzata in modo tale da incrementare robustezza del sito web, sul quale vengono applicati fogli di stile CSS puri. Tali layout sono stati sviluppati per ottenere un design fluido, scalabile e funzionale per la maggior parte dei browser e dei dispositivi. Sul lato stilistico i colori scelti sono quelli standard che ci si aspetta da un sito web medico-sanitario. Comprendono infatti varie gradazioni di blu e bianco, le quali risultano rilassanti all'occhio.

### 2.2.1 Header

All'interno dell'header è presente il *logo* del sito, i pulsanti *accedi/registrati* e la *navbar*. Il *logo* è situato a sinistra ed è caratterizzato da un link non circolare, ovvero funziona come link in tutte le pagine fatta eccezione per la home. *(Una precisazione in questo senso: nell'index è stato messo un link a # che di fatto se schiacciato non compie nessuna azione. Questo è stato fatto per non eliminare completamente l'ancora, perché è attraverso questa che nel CSS viene definito lo stile del logo.)*

I pulsanti *accedi/registrati* sono posti in alto a destra e dopo aver effettuato il login o la registrazione, spariscono, lasciando spazio ad un singolo pulsante *esci* per uscire dal proprio account. Infine la *navbar* occupa una lunga linea che taglia tutto il sito orizzontalmente. Questa ospita tutti i link alle pagine del sito ed è l'elemento centrale della navigazione nel sito web. Bisogna notare che i link alle pagine non occupano tutta la lunghezza della navbar, ma sono posti il più possibile a sinistra. Abbiamo scelto questo posizionamento perché, dagli studi fatti sul movimento oculare sugli utenti in una pagina web, si è ricavata una termografia dove le zone calde rappresentano le parti della pagina in cui l'utente ha dedicato più tempo, le quali risultano avere una forma ad F. La disposizione dei nostri link fa uso di questa conoscenza per essere più accessibile dai nostri utenti.



Figura 1: Header

### 2.2.2 Breadcrumbs

Le *breadcrumbs* sono presenti in tutte le pagine del sito, sia mobile che desktop, e comprendono un insieme di campi che identificano la posizione dell'utente all'interno del sito. L'ultimo campo è la pagina corrente, ovvero la pagina che l'utente sta visualizzando. Per evitare link circolari, quest'ultimo campo è solo testuale.



Figura 2: Breadcrumbs utente non registrato

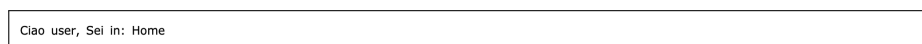


Figura 3: Breadcrumbs utente registrato

### 2.2.3 Contenuto

Lo schema che abbiamo utilizzato è quello a tre pannelli, i quali corrispondono alle seguenti domande:

1. **Dove mi trovo?** la prima risposta a questa domanda si trova nel titolo della pagina. La seconda si trova nelle *breadcrumbs*, che indicano il percorso fatto per arrivare in quel punto;

2. **Di cosa parla questa pagina?** la risposta a questa domanda si trova nello stesso contenuto della pagina, diviso solitamente in due colonne, una posizionata a sinistra per seguire le regole date dalla termografia e una a destra contenente un'immagine. Questa disposizione è dettata anche dal fatto che il contenuto delle pagine web si comporta in modo opposto a quello fisico. Ad esempio nei giornali le immagini prendono gran parte della pagina poiché attirano maggiormente l'attenzione, mentre nel web è il testo a fare da leader nell'attenzione ricevuta. Questo viene definito effetto zapping. Il nostro sito cerca di evitarlo;
3. **Dove posso andare?** la risposta a questa domanda si trova nella *navbar*, la quale non cambia mai di posizione.



**Figura 4:** Esempio del contenuto di una pagina, in questo caso *Citologia Nasale*

### Home

Questa pagina ha la funzione di accogliere l'utente. È composta da uno *slider* che mostra varie immagini del sito e da un testo che descrive il contenuto del sito sinteticamente. Il testo è composto da meno di 100 parole, questo segue le regole base studiate da alcuni membri del gruppo durante il corso di WIM. Abbiamo quindi posto grande attenzione a restare nei limiti di tempo adeguati per mantenere l'attenzione degli utenti. Infatti dopo vari studi è stato dimostrato che il tempo di permanenza in una pagina per il primo accesso da parte di un utente è 31 secondi; la nostra home in media rientra in questa tempistica.

### Pagina Contatti

La funzione di questa pagina è di dare all'utente tutte le informazioni critiche su come contattare il dottore o trovare il suo studio medico. Per fare ciò, oltre al testo contenente le varie informazioni, è stata inserita una mappa GoogleMaps interattiva.



**Pagina Dottore**

Questa pagina serve ad introdurre la figura del Dott. Marco Donati, illustrandone le competenze e achievement professionali.

**Pagina Area Medica**

La pagina Area Medica racchiude tutte le 4 prestazioni mediche fornite dal dottore. Si compone quindi di 4 link alle pagine singole (nelle quali le prestazioni vengono spiegate nel dettaglio) e di brevi descrizioni per ciascuna di esse.

**Pagina Notizie**

La pagina Notizie raccoglie tutte le news postate dal dottore. È fondamentale per gli utenti poiché comprende informazioni piuttosto importanti circa le chiusure straordinarie dello studio medico, oltre a fornire documentazione su nuove terapie e funzionalità del sito in sviluppo.

**Pagina Prenota Visita**

Questa pagina viene divisa in 2 colonne. Entrambe ospitano dei form; questi guidano l'utente registrato (gli utenti non registrati non possono accedere alla pagina) nella scelta di una data per una visita specialistica. Si deve scegliere anno, mese, giorno, prestazione specialistica e quindi l'orario.

**Pagina Consulti Online**

Questa pagina è dedicata interamente alla chat privata con il Dottore. Può essere acceduta solo da utenti registrati. Per ogni messaggio inviato viene mostrato nome utente, data e orario di invio.

**Pagina Accedi**

Questa pagina viene utilizzata per accedere al sito del dottore previa registrazione. Per accedere bisogna inserire email e password.

**Pagina Registrati**

Questa pagina viene utilizzata per registrarsi al sito del dottore. Per registrarsi è necessario fornire nome, cognome, numero di telefono, email e password.

**2.2.4 Footer**

All'interno del footer compaiono le immagini di validazione W3C HTML, W3C CSS, le informazioni relative al dottore (partita IVA, numero di iscrizione all'ordine dei medici) e i nomi degli sviluppatori. Le immagini per la validazione sono link cliccabili, che una volta premuti validano la pagina rispettivamente in xHTML 1.0 e CSS 3.



**Figura 5: Footer**



### 2.2.5 Database

Il nostro database è composto dalle tabelle:

1. **Messaggi:** contiene tutti i messaggi dei Consulti Online (sia degli utenti, sia del dottore);
2. **Notizie:** contiene tutte le notizie postate dal dottore;
3. **Utenti:** contiene tutte le informazioni legate agli utenti registrati;
4. **Visite:** contiene tutte le visite prenotate da utenti registrati.

## 3 Implementazione lato front-end

### 3.1 HTML

Si è deciso di usare xHTML 1.0 per garantire il supporto a tutti i tipi di web browser. La parte di codifica in xHTML 1.0 ha posto le basi della struttura del sito ed è stata completata, nella sua interezza, come prima cosa. Questa è stata templatizzata il più possibile per garantirne un riutilizzo efficace. Infatti sono stati creati in prima istanza l'header e il footer, così da poterli utilizzare in ogni pagina, e successivamente sono state creati ad hoc i contenuti di tutte le pagine. Si è deciso di utilizzare HTML 5 solo per l'utilizzo della mappa interattiva Google Maps tramite il tag *iframe*. Strumento fondamentale per validare la correttezza del codice xHTML 1.0 e HTML 5 è stato il validatore fornito da W3C. Il validatore è stato settato con Document Type xHTML 1.0 Strict.

Da menzionare il fatto che si è presa la decisione di definire il current link come classe e non come id, per evitare un conflitto con l'id dell'area medica precedentemente definito.

### 3.2 CSS

Fin da subito si è deciso di progettare 3 fogli di stile separati, rispettivamente:

1. Stile desktop;
2. Stile mobile;
3. Stile stampa.

Da notare il fatto che le funzionalità CSS3 sono state implementate per ultime, in maniera tale da preservare la struttura principale del sito nel caso in cui si utilizzi un browser che non supporta la suddetta versione di CSS. Questa caratteristica permette al nostro sito di "degradare elegantemente".

#### 3.2.1 CSS desktop

Scelte degne di nota sono state:

- Il logo, ritenuto molto importante in quanto associato dai clienti con lo studio, è stato inserito all'interno dell'HTML e poi modificato successivamente in CSS;
- Tutte le immagini sono state adattate alla pagina usando il CSS (dimensione e posizionamento);
- Si è deciso di utilizzare una impostazione a due colonne per la maggior parte delle pagine;
- Abbiamo creato una classe `.grassetto` in CSS e non utilizzato `strong/em` in HTML, permettendo così il giusto funzionamento della lettura attraverso lo screen reader, sia la correttezza grafica del testo;
- Per far sì che con la riduzione della grandezza della pagina, le immagini del footer non occupino troppo spazio, abbiamo creato un punto di rottura usando le media queries `@media (max-width: 906px)` per nasconderle sotto una certa dimensione della pagina.

### 3.2.2 CSS mobile

Le modifiche che abbiamo effettuato per l'adattamento alla versione mobile sono state:

- L'accesso/registrazione utente sono stati condensati in una singola icona, posta in alto a sinistra, la quale, se cliccata, mostra un piccolo menù a tendina che permette di scegliere se accedere o registrarsi. L'icona è posta a sinistra, quindi tecnicamente scomoda da raggiungere con il pollice, in quanto viene utilizzata soltanto per effettuare il login, o registrarsi. Una volta effettuato l'accesso però l'utente tenderà a restare collegato con il proprio account, ed ad utilizzare principalmente il menu di navigazione, spiegato nel punto successivo;
- Il menù di navigazione invece è stato implementato tramite un menù ad hamburger in alto a destra, quindi comodo da raggiungere con il pollice. Questo pulsante, al click, diventa un menù a tendina che occupa buona parte dello schermo ma senza nascondere completamente il contenuto della pagina;
- Sono stati pensati 2 differenti punti di rottura:
  1. Il primo a 630px orizzontali sancisce l'effettivo passaggio alla versione mobile;
  2. Il secondo a 500px orizzontali prevede l'adattamento del logo e del titolo che vengono ridotti di circa il 50%. Per questo fin da subito abbiamo reso anche il titolo del sito cliccabile, così da sopperire alla poca superficie occupata dal logo rimpicciolito.

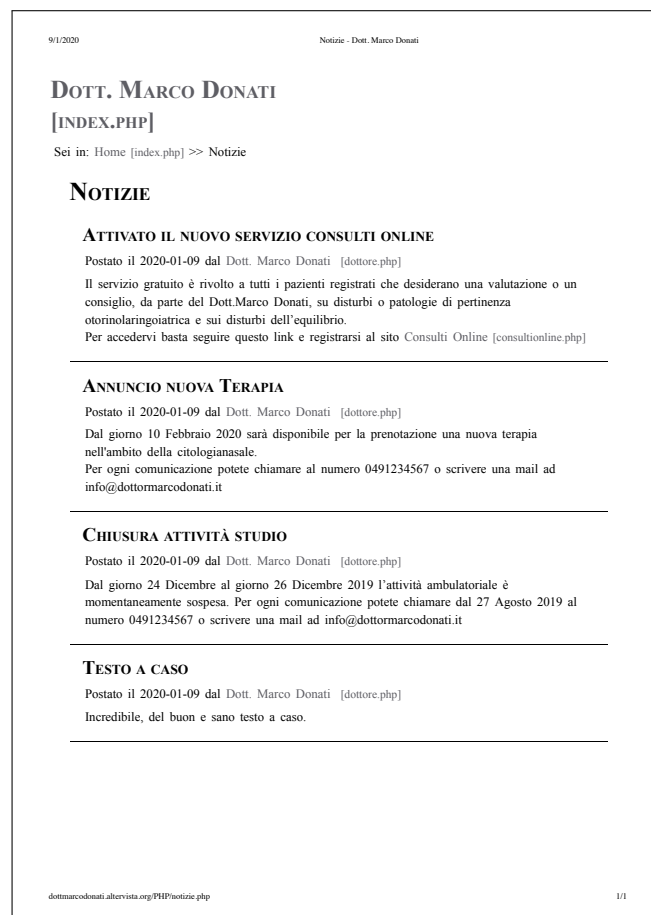


**Figura 6:** Esempio di una pagina mobile, in questo caso *Notizie*

### 3.2.3 CSS stampa

Il CSS di stampa è stato scritto seguendo tutte le linee guida illustrate a lezione:

1. I link sono stati colorati in grigio (#5E5E64) (per garantire una stampa full b/w) e l'indirizzo URL è stato esplicitato;
2. Il menù di navigazione è stato rimosso;
3. La mappa di Google Maps è stata rimossa dalla stampa, poiché non è ritenuta utile come immagine statica;
4. I form della pagina *Prenota Visita* e *Consulti Online* sono stati nascosti perché si è pensato che un utente che stampi le relative pagine voglia rispettivamente soltanto la lista delle visite prenotate, o lo storico dei messaggi scambiati con il dottore, e non abbia l'interesse di stampare i form di queste pagine;
5. In alto, accanto alla data, è stata lasciata la stampa dell'id pagina (Es. *Consulti Online - Dott. Marco Donati* per la pagina *consultionline.html*) per dare più contesto.



**Figura 7:** Esempio di una pagina di stampa, in questo caso *Notizie*

### 3.3 JavaScript

Per il comportamento di alcune pagine del sito, lato client, è stato utilizzato uno script JavaScript, comprendente diverse funzioni. È stato scelto di limitare al massimo l'utilizzo di queste funzionalità, in quanto non è possibile fare nessuna assunzione sulla disponibilità di questa tecnologia. Il sito infatti rimane completamente navigabile e utilizzabile nel caso in cui JavaScript venga disabilitato o non sia comunque disponibile. Sono state quindi fornite tutte le alternative, lato server, attraverso PHP (unica alternativa in questo senso è rappresentata dallo slideshow, che degrada elegantemente mostrando solo una immagine statica). Da mobile si presuppone ovvimante JavaScript attivo.

Le principali funzioni che abbiamo implementato (tutte contenute all'interno di un unico file) sono state:

- **validaAccedi()** : questa funzione controlla che il campo email immesso nel form specifico per l'accesso al sito sia corretto, mentre per la password controlla semplicemente che non sia vuota;
- **validaRegistrati()** : questa funzione controlla che tutti i campi immessi nel form specifico per la registrazione al sito siano corretti;
- **setPrenotaVisitaForJS()** : viene eseguita solo se c'è javascript; questa funzione fa sì che, per chi abbia JavaScript attivo, venga visualizzata la pagina come dovrebbe essere (con tutti i controlli e le chiamate alla pagina PHP per controllare la disponibilità), mentre per chi non dovesse avere JavaScript attivo, non venga eseguita questa funzione e quindi venga lasciata la pagina così com'è impostata in HTML, comunque utilizzabile;
- **dataValida(date)** : funzione che controlla che la data sia inserita nel giusto formato (d/m/y);
- **controllaDisponibilita()** : funzione che nasconde gli orari se non si riferiscono più al giorno selezionato, in caso di cambio data da parte dell'utente;
- **openCloseMenu(menu)** : funzione per chiudere e aprire i menù a tendina nella versione mobile del sito;
- **plusDivs(n)** : funzione per far scorrere lo slideshow;
- **showDivs(n)** : funzione per mostrare le diverse immagini dello slideshow;
- **changeFocusAccedi(event, campo)**: funzione che, al premere del tasto invio una volta che è stata inserita la email nel form per accedere al sito, passa il focus sul campo password, senza inviare direttamente il form.

### 3.4 Compatibilità con browser datati

Per garantire la compatibilità con browser datati:

- All'interno dei form sono stati usati *type text* per le email e il *type number* per i numeri telefonici, scegliendo quindi di non utilizzare *type email* e *type tel* che sono attributi di HTML5;
- *sticky* viene messa in coda al codice come tutte le altre funzioni CSS3, così che, se la pagina venisse aperta da un browser che non supporta CSS3 la funzionalità della pagina non verrebbe compromessa; semplicemente le funzioni CSS3 opzionali non sarebbero mostrate. Questo vale anche per proprietà come *border-radius*, *box-sizing*, *flex*, *transform*;
- Non usando CSS3 non è possibile fare il bordo dei radio button colorato quando è selezionato, quindi un user di IE8 o precedenti vedrà solo i pallini senza il bordo della label colorata. Questo però non inficerà sulla usabilità perché il radio button selezionato sarà comunque distinguibile da quelli non selezionati;
- Abbiamo usato *@media flex*. Tale proprietà CSS3, pur non essendo supportata dai vecchi browser, non degrada in modo particolare il sito. Anzi si parla solo di una questione puramente estetica;
- Per quanto riguarda la mappa di Google Maps, per utilizzarne la funzionalità non abbiamo avuto scelta se non usare funzioni HTML 5, che quindi risultano errate in un validatore xHTML 1.0 Strict. (Come è stato detto a lezione); Quindi nella pagina contatti, il documento è stato dichiarato come HTML 5, unica eccezione del nostro sito.

## 4 Accessibilità

Il sito mantiene la completa separazione tra struttura, presentazione e comportamento. La struttura è stata sviluppata tramite documenti xHTML, i quali richiamano i fogli di stile esterni CSS puri che implementano la presentazione, ed infine, per quanto riguarda il comportamento, sia script esterni realizzati con JavaScript che pagine dinamiche in PHP. Tutto il codice redatto è stato scritto secondo le raccomandazioni W3C WCAG 2.0 (Level AAA). Si sono evitati tag e attributi deprecati.

### 4.1 Screen reader

Sono stati effettuati vari test tramite screen reader, cercando il più possibile di simulare l'esperienza di un non vedente sul nostro sito. Così facendo sono state individuate le criticità e sono state quindi sviluppate tecniche per risolverle.

La criticità più fastidiosa era data dalla mancanza di *skipLink* efficaci; sono quindi stati introdotti 3 diversi *skipLink*:

- Il primo è posto all'inizio della pagina e fa saltare il logo e la navbar, andando al contenuto principale della pagina stessa;
- Il secondo è presente solo nella pagina contatti e permette di saltare la mappa di Google Maps;
- Il terzo è posto poco prima del footer e riporta l'utente alla barra di navigazione. Si è pensato infatti che l'utente, dopo aver visitato l'intera pagina, voglia poterne visitare un'altra e viene quindi ridirezionato al menù di navigazione stesso.

Lo *slideshow* è stato programmato in modo da non creare fastidi ai non vedenti. A tal proposito è stato illuminante il seminario svolto in classe. Il nostro *slideshow* infatti non reagisce con lo screen reader, poiché non cambia in automatico le immagini. Viene inoltre data la possibilità di saltarlo direttamente tramite *skipLink*.

Le parole in lingue diverse da quella italiana sono state racchiuse all'interno di tag con l'attributo *lang*, per permettere una corretta lettura da parte dello screen reader.

Abbiamo infine provato a navigare i form senza l'uso del mouse e siamo rimasti soddisfatti dal risultato dopo aver corretto vari errori nell'HTML. I *radio button* che normalmente sono colorati di verde se un'ora è disponibile, o di rosso se invece non lo è (totalmente inutile in caso di disabilità visiva), sono stati modificati aggiungendo l'attributo *disabled="disabled"* facendo in modo che al passaggio sul form il pulsante dichiarasse il suo stato di disabilitato o meno.



## 4.2 Colori

I colori, a tema medico, sono stati scelti in modo da seguire le regole del contrasto e raggiungono un livello di accettabilità tripla A. (In particolare, ratio 7.14:1 tra #2E5894 e #FFFFFF, e di 14.39:1 tra #192168 e #FFFFFF). Questi inoltre restano ben distinti anche usando una scala di grigi o con alterazione/mancanza di colori.

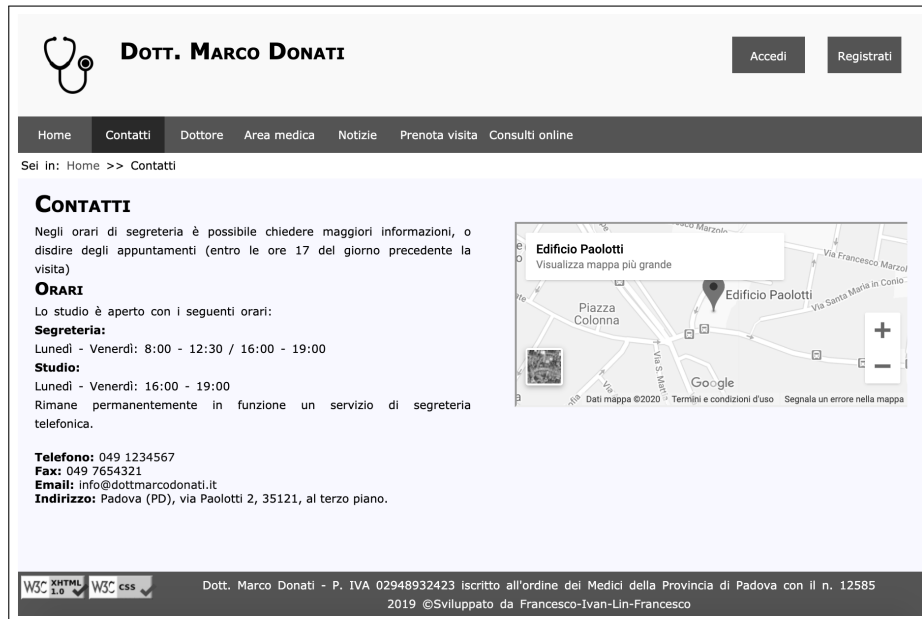


Figura 8: Color greatly reduced

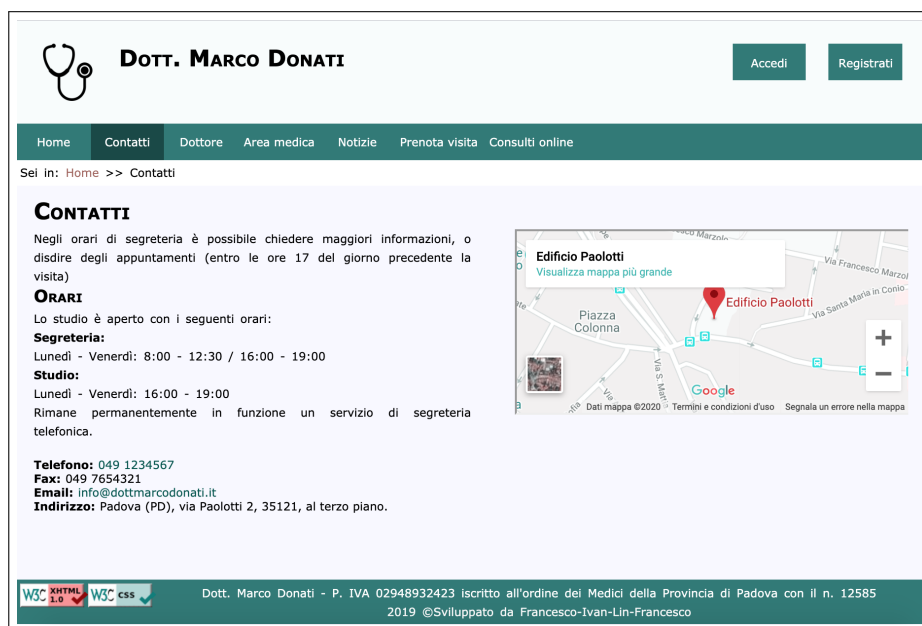


Figura 9: Blue greatly reduced

### 4.3 Test

Tutti i test riguardanti l'accessibilità sono stati svolti (e passati) con:

- *Total Validator*;
- *Silktide Screen Reader Simulator*;
- *Color Contrast Analyzer*;
- *Wawe*.

I feedback dati da questi test ci hanno quindi permesso di redarre le sezioni riguardanti l'accessibilità.

## 5 Implementazione lato back-end

### 5.1 PHP

Il comportamento del sito lato server è gestito da file PHP, i quali sostituiscono completamente i controlli JavaScript nell'eventualità che sia disabilitato o non disponibile. I file PHP interagiscono con il database, forniscono sessioni di utilizzo per utenti loggati (incluso l'admin) e caratterizzano il comportamento generale delle pagine. Le funzioni più utilizzate e generali sono state definite in un file a parte *funzioni.php*, mentre i file specifici per ogni pagina contengono le funzioni di utilità a loro associati.

Descriviamo quindi le funzioni di maggiore importanza contenute in *funzioni.php*:

- **getPaginaHTML(pageName)**: Funzione che, ricevuto in ingresso la pagina corrente, ne recupera il file HTML corrispondente e successivamente:
  - ci inserisce Header e Footer;
  - mostra il pulsante Esci se si è loggati;
  - inserisce i breadcrumbs
  - nel caso si sia admin modifica le voci del menu di navigazione "Prenota visita" e "Consulti online" rispettivamente in "Visite prenotate" e "Elenco chat"
  - elimina i link ricorsivi.

Una volta eseguiti questi passaggi ritorna una stringa contenente la pagina HTML, senza eventuale contenuto dinamico all'interno del main (che sarà da gestire poi nella relativa pagina).

- **controlloCampiDatiRegistrati(nome, cognome, telefono, email, password, confermapassword)**: Funzione che, ricevute in ingresso le credenziali del form di registrazione, ritorna true se e solo se tutti i dati sono stati compilati correttamente (non vuoti e conformi alle espressioni regolari a loro corrispondenti), altrimenti ritorna una stringa contenente gli errori relativi ai campi compilati erroneamente;
- **preparaHTMLListaVisite(arrayVisite)**: Funzione che riceve un array dove ogni elemento è a sua volta un array con le informazioni della singola visita. Ritorna una stringa contenente la lista delle visite in codice HTML.

Descriviamo inoltre le funzioni degne di nota relative alla connessione con il database contenute in *dbaccess.php*:

- **\_\_destruct()**: ridefinisce il distruttore della classe, in modo che chiuda la sessione corrente con il database (qualora sia stata aperta);
- **openDBConnection()**: apre la connessione con il database;
- **login(email, password)**: se il login avviene con successo ritorna l'email, nome e cognome dell'utente, altrimenti ritorna false;
- **registrazioneUtente(email, nome, cognome, telefono, password)**: ritorna true se la registrazione è andata a buon fine, altrimenti ritorna false;
- **getListaVisitePrenotatePeriodo(periodo, email, getNomeUtente)**: questa funzione ritorna un array con le visite prenotate in un determinato periodo che viene passato come parametro. Questo parametro è una stringa contenente un codice per identificare il periodo desiderato (Es. "f7" indica i prossimi 7 giorni da oggi, "20p" indica le ultime 20 visite già passate). I parametri email e getNomeUtente sono opzionali, infatti se non viene passato il campo email vengono considerate le visite di qualsiasi account, mentre getNomeUtente, che serve per dichiarare se si vuole ricevere anche i dati relativi a chi ha prenotato la visita, di default è true.

Infine, descriviamo le scelte implementative degne di nota relative alle singole pagine:

#### **registrati.php**

Nel contesto dell'inserimento errato delle credenziali, la pagina, una volta ricevute tramite form le informazioni da inviare al database, notifica gli errori relativi all'inserimento dei campi errati, mantenendo però le informazioni precedentemente inserite dall'utente. Questo per evitare all'utente di dover reinserire nuovamente tutte le informazioni.

#### **accedi.php**

Per quanto riguarda questa pagina, la funzionalità descritta nel paragrafo precedente non è stata implementata. Questo per non dare troppe informazioni ad un eventuale utente malevolo, relativo al campo immesso erroneamente. Infatti, se mostrassimo quale campo si è sbagliato, lasciando inserito quello corretto, daremmo delle informazioni che potrebbero essere usate in modo malintenzionato.

#### **controllaDisponibila.php**

Questa pagina serve solo per stampare un file in formato JSON che viene utilizzato come API da JavaScript in *prenotavisita.php*. Se non dovessero esserci variabili impostate, si viene reindirizzati alla pagina 404 (per esempio se si dovesse provare ad accedere alla pagina copiando un URL). Non serve controllare che la data sia corretta, perché questa pagina viene chiamata soltanto da JavaScript, che quindi dovrà essere attivo (e deve aver già fatto i controlli). Se JavaScript non dovesse essere attivo i controlli saranno semplicemente fatti lato server tramite PHP quando verrà inviata la richiesta, e quindi questa pagina non verrà chiamata.

#### **prenotavisita.php & consultationline.php**

Se non si è loggati, il sito lo riconosce e invita il visitatore ad effettuare l'accesso oppure a compiere la registrazione. Tutto questo è stato implementato tramite PHP.

#### **notizie.php**

Il form relativo all'inserimento e alla modifica di una notizia è lo stesso, cambia solo il valore del pulsante *submit*, che poi (sempre in PHP) serve ad identificare l'azione corrispondente.

Una ultima precisazione in merito alle pagine contenenti form: è presente un tag `<erroriDaMostrare />` che, nel caso ci siano errori, viene sostituito con gli errori che devono essere mostrati relativi alla singola pagina, mentre se non ce ne sono viene semplicemente sostituito dalla stringa vuota.

## 6 Ruoli

Di seguito si elencano le parti alle quali hanno contribuito i membri del gruppo:

- **Francesco Pecile**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. Alcune funzioni JavaScript;
4. Test di accessibilità;
5. Stesura relazione.

- **Francesco Bari**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. Alcune funzioni JavaScript;
4. PHP di alcune pagine;
5. Stesura relazione.

- **Ivan Furlan**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. Alcune funzioni JavaScript;
4. PHP delle pagine;
5. Contributo informazioni relazione.

- **Lin Zhaohui**

1. HTML di alcune pagine;
2. CSS di alcune parti del sito;
3. Funzioni JavaScript;
4. PHP di alcune pagine;
5. Contributo informazioni relazione.