

Salone Anna

Relazione Salone Anna

Nome Progetto Salone Anna
Redazione Pietro Gabelli
Sebastiano Marchesini
Andrea Grendene

Sito:

<http://tecweb2016.studenti.math.unipd.it/smarches/>

Login: admin

Password: admin

Sommario

Relazione progetto Salone Anna destinato a Tecnologie Web.

Contenuti:

1	Introduzione	4
1.1	Descrizione generale	4
1.2	Caratteristiche degli utenti	4
1.3	Vincoli generali	4
1.4	Requisiti	5
1.5	Riferimenti	5
2	Suddivisione dei lavori	6
3	Linguaggi Utilizzati	7
4	Architettura	8
4.1	Progettazione layout	8
4.2	Sviluppo layout	8
4.3	Layout per dispositivi mobili	9
4.4	Layout di stampa	9
5	Struttura	10
6	Gestione Dati	13
6.1	Introduzione	13
6.2	Classi	13
6.3	Associazioni	15
6.4	Codice PHP per l'interazione con il database	16
7	Presentazione	18
8	Comportamento	19
9	Accessibilità	20
9.1	Implementazione	20
9.2	Combinazione dei colori	21
10	PHP	23
10.1	Descrizione generale	23
10.2	Pagine XHTML	23
10.3	Sessioni	23
11	Verifica e Test	25
11.1	Ambiente di lavoro	25
11.2	Dispositivi utilizzati	25
11.3	Verifica del codice prodotto	25
11.4	Conclusioni	26

Contenuti:

Lista delle tabelle

1	Elenco dei requisiti	5
---	--------------------------------	---

Lista delle figure

1	Schema del sito	9
2	Schema Relazioni	16
3	Pagina vista da persone con problemi nel distinguere i colori	22

1 Introduzione

1.1 Descrizione generale

Il progetto SaloneAnna è strutturato su una base di dati MySQL per la gestione di un negozio di parrucchieri. Fulcro del programma risiede nell'amministrazione dei clienti, salvati nel database e collegati agli appuntamenti gestiti dall'amministratore.

Il contesto su cui si appoggia la scelta di questo programma riguarda un salone di piccole dimensioni, si parla di un prodotto accessibile da tutte le dipendenti e titolari per tenere aggiornato il negozio con la massima efficienza.

Inoltre l'amministratore del salone e chi possiede l'accesso al sito hanno la possibilità di gestire un magazzino, cancellando, inserendo o modificando ogni singolo prodotto, in modo da avere un inventario in evoluzione e ben gestibile. Un utente non registrato invece potrà visualizzare le informazioni generiche del negozio, sfogliare una galleria dinamica delle realizzazioni e contattare tramite form il negozio.

1.2 Caratteristiche degli utenti

Gli utenti del sito saranno persone in ricerca di un nuovo negozio nella zona, in grado di utilizzare gli strumenti per la navigazione web o alle prime armi.

Il sito è rivolto ad un pubblico generico, all'interno del quale possiamo individuare le seguenti categorie:

Categoria di utenti: privati;

Funzionalità: Informarsi sulla locazione e sul numero del salone. Consultare realizzazioni e contattare il negozio lasciando il proprio indirizzo email.

Termini generali: Non eccessivamente distante dal punto vendita, in un raggio di circa 70 Km.

Categoria di utenti: amministratori;

Funzionalità: area riservata in cui poter aggiungere, rimuovere o aggiornare i prodotti, inserire nuovi clienti e conseguentemente gestire gli appuntamenti.

1.3 Vincoli generali

- Il sito dev'essere accessibile da parte di categorie d'utenti diversificate ed utilizzando dispositivi di vario tipo, compresi smartphones e tablet;
- Il sito deve presentare possibilità di stampa flessibile a seconda della pagina richiesta;
- Il sito dev'essere visitabile tramite i seguenti browser:
 - Firefox 3.6;
 - Internet Explorer dalla versione 7 alla versione 11; Edge 13;
 - Chrome 14;
 - Opera 12.16;
 - Safari 9.
- Separazione tra struttura, presentazione, comportamento;
- Conformità agli standard W3C per XHTML, CSS, JS;
- Sito comprensibile da screen-reader.

1.4 Requisiti

Di seguito sono presentati i requisiti emersi dall'analisi iniziale e quelli che si sono aggiunti nel corso dello svolgimento del progetto. Ciascuno è identificato da un numero progressivo per semplificarne l'individuazione successiva.

ID Req.	Descrizione
1	Il sito dev'essere visualizzabile sui browser elencati all'interno di "Vincoli generali"
2	Il sito dev'essere accessibile indipendentemente dalla grandezza dello schermo del dispositivo
3	Il sito dev'essere fruibile anche senza richiedere un foglio di stile
4	Le figure significative dovranno essere comprensive di un attributo alt per favorire l'accesso ad utenti non vedenti
5	Ai tag quali <input> e <textarea> devono essere associati tabindex e accesskey
6	Le gradazioni di colori non devono risultare sgradevoli o di intralcio a persone affette da daltonismo
7	Il layout deve risultare fluido nel ridimensionamento del carattere tramite i tasti Ctrl + e Ctrl -
8	Il sito deve essere validato per la parte di XHTML2.0, CSS3 e secondo gli standard WAI

Table 1: Elenco dei requisiti

1.5 Riferimenti

Per la progettazione del sito, si è fatto riferimento alle seguenti normative e specifiche:

- Legislazione e linee guida per accessibilità dei siti web istituzionali, 2011 <http://www.math.unipd.it/~artico/direttiva.htm>
- Specifiche Web Accessibility Initiative (WAI) <http://www.w3.org/WAI>;
- Specifiche Web Content Accessibility Guidelines (WCAG) 2.0 www.w3.org/TR/WCAG20/;
- Specifiche Sezione 508 <https://www.section508.gov/>;
- Ruota dei colori accessibile <http://colorfilter.wickline.org/>;
- Slides del corso: <http://docenti.math.unipd.it/gaggi/tecweb/materiale.html>.
- Risorse di WebAIM, Web Accessibility In Mind: <http://webaim.org/resources>

2 Suddivisione dei lavori

Per realizzare il progetto abbiamo cercato di distribuire il carico di lavoro in modo quanto più possibile uniforme, dividendo il progetto in sezioni indipendenti così da procedere nello sviluppo in modo quanto più possibile parallelo e libero da conflitti.

La suddivisione è stata la seguente:

- **Sebastiano Marchesini**

- Title, Meta-Title e Keywords;
- Creazione fogli di stile (home.css, print.css, mobile.css, expolerer.css);
- Creazione logo, footer del sito;
- Creazione pagine **Contattaci**, **Chi Siamo**, **Home**;
- Accessibilità;
- Stesura sezione Introduzione, Associazioni, Classi, Linguaggi Utilizzati, Architettura, Presentazione nella relazione.

- **Andrea Grendene**

- Creazione e modifica del database MySQL;
- Gestione sessioni;
- Comportamento del sito tramite PHP;
- Stesura sezioni Linguaggi utilizzati, Struttura, Gestione dati, PHP nella relazione.

- **Pietro Gabelli**

- Struttura del sito tramite PHP;
- Gestione errori;
- Comportamento del sito (JS);
- Accessibilità;
- Testing;
- Stesura delle sezioni Descrizione Generale, Suddivisione lavori, Accessibilità, Verifica e test nella relazione.

3 Linguaggi Utilizzati

La struttura del sito è stata realizzata rispettando lo standard XHTML 1.0 ed è stato validato secondo gli standard del W3C.

La presentazione è stata costruita in CSS, cercando di utilizzare quanto più possibile CSS2, che non valida per poche proprietà usate per una maggiore accessibilità. Risulta invece valido con CSS3, secondo gli standard W3C.

Javascript è stato utilizzato per definire funzioni di utilità alle pagine ed effettuare controlli dinamici sui dati inseriti nelle form.

La gestione dei dati è stata affidata al linguaggio SQL. Il database è stato caricato in uno spazio riservato tramite “phpmyadmin”, un’applicazione che permette di gestire la base di dati direttamente dal browser. Non c’è un file di riferimento tra quelli del sito perché il database è stato creato tramite degli appositi comandi, quindi esso è situato in uno spazio riservato a parte.

Per il comportamento, oltre a JS è stato utilizzato il linguaggio PHP. Nello specifico esso è stato usato per gestire le sessioni e le operazioni da effettuare sul database.

4 Architettura

Il layout è stato strutturato allo scopo di rendere il sito fruibile indipendentemente dal dispositivo, definendo:

- layout per dispositivi desktop;
- layout per dispositivi mobili;
- layout di stampa.

4.1 Progettazione layout

Si è deciso di utilizzare un layout di tipo responsive a singola colonna principale, adattabile secondo la dimensione dello schermo, impostando un limite sulla larghezza per passare al layout per dispositivi mobili.

4.2 Sviluppo layout

Nell'immagine che segue viene mostrata la struttura data ai vari blocchi *div* che compongono il sito, in cui sono contenute le informazioni divise per area tematica le informazioni.

Il layout fluido orizzontalmente si adatta in base alla larghezza dello schermo, con l'obiettivo di far sviluppare verso il basso il sito.

Il foglio di stile standard viene utilizzato sui browser per computer desktop e portatili, fino a che la larghezza dello schermo rimane maggiore di 650 px: al di sotto si passa automaticamente ad usare il CSS per al mobile.

Analizzando il sito spostandosi dall'alto verso il basso, gli elementi che si incontrano sono:

- Il div **header**, informa l'utente su ciò che sta visitando; comprende:
 - Titolo e logo; nella pagina relativa alle vendite è presente inoltre il pulsante d'accesso all'area riservata del sito, che conduce alla pagina da cui effettuare il login.
 - Il div **breadcrumbs**: aiuta l'utente ad identificare la posizione in cui si trova all'interno del sito, a partire dalla homepage;
 - Il menù utente, identificato con il div **menu**, presenta scelte differenziate a seconda che si abbia effettuato l'accesso. Agli utenti non autenticati sono disponibili i collegamenti a *Home*, *Foto*, *Chi Siamo*, *Prezzi*, *Contattaci*.
Dopo aver effettuato l'accesso si hanno a disposizione le seguenti scelte *Home*, *Prezzi*, *Immagini*, *Prodotti*, *Clients*, *Appuntamenti*, *Utilità*.
Viene evidenziata la posizione corrente e si trova in posizione centrale rispetto alla pagina; si estende orizzontalmente. Per agevolare lo sviluppo questa parte viene costruita da funzioni PHP;
- Segue il contenuto vero e proprio della pagina, inserito nel div **content**: ha il compito di esporre le informazioni che si stanno trattando;
- Alla fine della pagina, il **footer**, costruito con un layout a 2 colonne; al suo interno sono presenti:
 - i principali riferimenti all'azienda;
 - un piccolo logo dell'azienda ed i simboli di validità per l'XHTML ed il CSS;

Anche il footer è stato costruito con una funzione, in modo da ottenere un template valido per tutte le pagine del sito e potersi concentrare nello sviluppo del contenuto delle pagine.

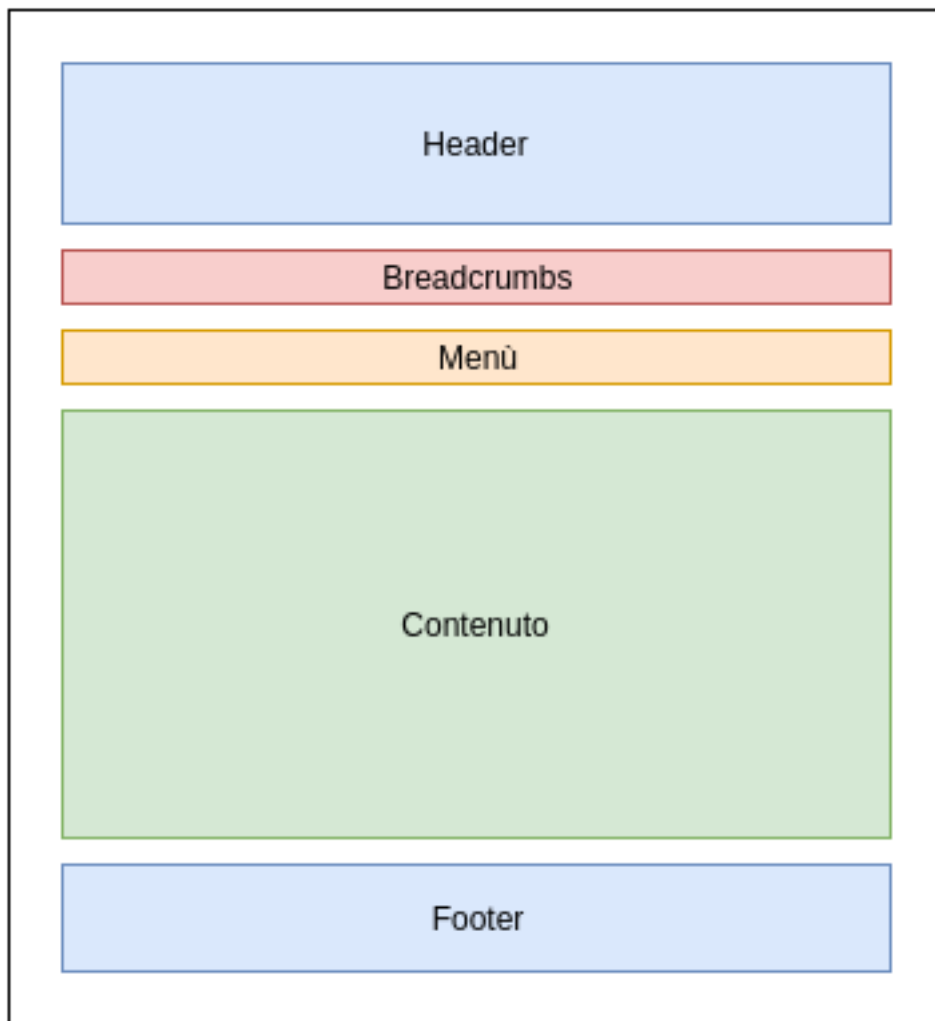


Figure 1: Schema del sito

4.3 Layout per dispositivi mobili

Il layout per dispositivi mobili è stato sviluppato in modo da favorire l'incolonnamento degli elementi, rimuovendo quanto possibili margini e padding, sfruttando al meglio l'area disponibile; sono state infine ridimensionate le immagini presenti.

4.4 Layout di stampa

Nel layout di stampa sono stati tolti gli elementi che non portavano informazioni significative; i contenuti sono stati privati dei colori; è stato rimosso il menù.

5 Struttura

La struttura del sito è stata divisa secondo le operazioni disponibili, così da semplificarne l'uso ed aiutare l'utente a trovare più facilmente le informazioni; inoltre per molte operazioni sono presenti più pagine, perché richiedono più passaggi e ognuno di essi deve stare su una pagina a parte.

Il sito è stato sviluppato in XHTML 1.0 Strict, supportato con PHP per semplificare la definizione e la stampa di codice comune a più pagine.

Di seguito sono elencate le pagine sviluppate, insieme ai link per poterle visualizzare da browser. Le pagine da cui vengono fatte operazioni di modifica sui dati presenti nel database, si occupano anche di fornire messaggi d'errore nel caso l'operazione non sia andata a buon fine o che i dati passategli non facciano riferimento ad oggetti presenti.

Alcuni file non hanno un link associato perché richiedono dei dati in input per poter funzionare:

- [index.php](#): questa pagina rappresenta la “copertina” del sito, il cui scopo è di attirare il cliente;
- [chi_siamo.php](#): in questa pagina sono state inserite le informazioni aziendali ed una breve descrizione dell'attività;
- [foto.php](#): in questa pagina si trova una galleria fotografica di alcuni lavori eseguiti;
- [listino.php](#): in questa pagina si trova il listino dei prezzi del Salone;
- [contattaci.php](#): qui dentro si trovano una form con cui contattare l'azienda, i numeri di telefono ed una mappa con le indicazioni per raggiungere il Salone;
- [login.php](#): in questo file si trova una form che permette di effettuare il login; la pagina si occupa anche di fornire un messaggio che informa dell'avvenuto successo o meno dell'operazione;
- [Immagini.php](#): in questa pagina si trova un menù da cui l'utente può accedere alle pagine per aggiungere, eliminare, modificare o visualizzare le immagini di [foto.php](#);
- [NuovaFoto.php](#): in questa pagina l'amministratore può inserire una nuova foto per la galleria, insieme alla sua descrizione;
- [EliminaFoto.php](#): in questa pagina l'amministratore può eliminare una o più foto della galleria;
- [SelezionaFoto.php](#): in questa pagina l'amministratore può selezionare una foto da modificare tra quelle già presenti nella galleria;
- [ModificaFoto.php](#): in questa pagina l'amministratore può modificare i dati della foto selezionata nella pagina precedente, ovviamente richiede che le sia trasmesso il codice dell'immagine da cambiare; può inoltre caricare una foto che andrà a sostituire quella già esistente;
- [Prodotti.php](#): in questa pagina è presente un menù da cui l'amministratore può eseguire varie operazioni sui prodotti;
- [ProdottiQuery.php](#): in questa pagina sono elencati i prodotti prossimi ad esaurirsi, insieme alla rispettiva quantità presente in magazzino;
- [ProdottiMax.php](#): in questa pagina sono elencati i prodotti più usati durante gli appuntamenti;
- [ProdottiClienteAppuntamento.php](#): in questa pagina l'amministratore può inserire i dati di un cliente per poter aggiungere i prodotti usati in uno dei suoi appuntamenti;
- [SelezionaAppuntamentoCliente.php](#): in questa pagina l'amministratore può selezionare uno degli appuntamenti del cliente che ha scelto nella pagina del punto precedente, in modo da poter aggiungere i prodotti che sono stati usati in quell'appuntamento, ovviamente richiede che siano forniti dei dati alla pagina;

- [SelezionaProdottiAppuntamento.php](#): in questa pagina l'amministratore può completare l'iter (iniziato con i due punti precedenti), inserendo la quantità di ogni prodotto che è stato usato durante l'appuntamento scelto, ovviamente richiede che siano forniti dei dati alla pagina;
- [EliminaProdotti.php](#): in questa pagina l'amministratore può eliminare uno o più prodotti presenti nell'inventario;
- [NuovoProdotto.php](#): in questa pagina l'amministratore può aggiungere un nuovo prodotto all'inventario inserendone i dati;
- [StoricoProd.php](#): in questa pagina l'amministratore può inserire il nome e il cognome di un utente per visualizzare i prodotti da lui usati;
- [Inventario.php](#): in questa pagina l'amministratore può visualizzare tutti i dati dei prodotti dell'inventario e modificarne le quantità presenti;
- [Clienti.php](#): in questa pagina è presente un menù da cui l'amministratore può eseguire delle operazioni sui clienti salvati nel database;
- [QueryCompleanno.php](#): in questa pagina sono elencati tutti gli utenti che compiono gli anni entro un mese;
- [ElencoClienti.php](#): in questa pagina sono elencati tutti i dati relativi ai clienti salvati nel database;
- [NuovoCliente.php](#): in questa pagina l'amministratore può inserire i dati per aggiungere un nuovo cliente;
- [ScegliCliente.php](#): in questa pagina l'amministratore può selezionare un cliente per modificarne i dati;
- [ModificaCliente.php](#): in questa pagina l'amministratore può inserire i dati del cliente scelto nel punto precedente per aggiornarli, ovviamente questa pagina richiede che le siano forniti dei dati;
- [ConfermaModificaCliente.php](#): in questa pagina viene confermato all'amministratore che i dati del cliente sono stati aggiornati con quelli che ha inserito nei due punti precedenti, ovviamente richiede che le siano forniti dei dati;
- [EliminaCliente.php](#): in questa pagina l'amministratore può eliminare uno o più clienti;
- [Appuntamenti.php](#): in questa pagina è presente un menù da cui l'utente può scegliere di eseguire alcune operazioni sugli appuntamenti salvati nel database;
- [Toptype.php](#): in questa pagina è presente una tabella con la frequenza dei tipi di appuntamento scelti dai clienti;
- [AppuntamentiSettimana.php](#): in questa pagina l'amministratore può visualizzare la lista degli appuntamenti divisi per giorno;
- [RicercaAppuntamenti.php](#): in questa pagina l'amministratore può inserire i dati relativi ad un cliente o ad un orario per visualizzare gli appuntamenti di quel cliente o prenotato a quell'orario;
- [AppClienteGiorno.php](#): in questa pagina l'amministratore può visualizzare gli appuntamenti selezionati in base ai dati inseriti nel punto precedente, ovviamente questa pagina richiede che le siano forniti dei dati;
- [NuovoAppuntamento.php](#): in questa pagina l'amministratore può inserire i dati di un nuovo appuntamento per aggiungerli al database;
- [ConfermaNuovoAppuntamento.php](#): in questa pagina viene confermato all'utente che l'aggiunta dell'appuntamento con i dati inseriti nel punto precedente è avvenuta con successo, ovviamente richiede che le siano forniti dei dati;

- [ScegliAppuntamento.php](#): in questa pagina l'amministratore può selezionare l'appuntamento a cui vuole modificare i dati;
- [ModificaAppuntamento.php](#): in questa pagina l'amministratore può inserire i nuovi dati dell'appuntamento scelto nel punto precedente, ovviamente questa pagina richiede che le siano forniti dei dati;
- [ConfermaModificaAppuntamento.php](#): in questa pagina viene confermato all'utente che la modifica dell'appuntamento con i dati inseriti nei due punti precedenti è avvenuta con successo, ovviamente richiede che le siano forniti dei dati;
- [EliminaAppuntamenti.php](#): in questa pagina l'amministratore può eliminare uno o più appuntamenti tra quelli salvati nel database;
- [Utilita.php](#): in questa pagina è presente un menù in cui l'amministratore può scegliere di eseguire delle operazioni che non sono presenti nei menù precedenti;
- [Messaggi.php](#): in questa pagina l'amministratore può visualizzare tutti i messaggi che sono stati ricevuti;
- [MostraMessaggio.php](#): questa pagina viene aperta quando l'amministratore clicca sul contenuto di uno dei messaggi, e qui egli può visualizzare tutti i dati relativi al messaggio, compreso il contenuto integrale, dato che nella pagina precedente esso veniva troncato ad un certo punto, ovviamente questa pagina richiede che le siano forniti dei dati;
- [EliminaMessaggi.php](#): in questa pagina l'amministratore può eliminare uno o più messaggi tra quelli salvati nel database;
- [CambioPassword.php](#): in questa pagina l'amministratore può cambiare la password del proprio account;
- [errore.php](#): questa pagina segnala all'utente che si è verificato un errore abbastanza grave, come l'impossibilità di accedere al database, richiede che le sia fornito il codice dell'errore. Si arriva a questa pagina anche se

6 Gestione Dati

6.1 Introduzione

Il database è stato scritto con SQL, un tipo di linguaggio che permette di modificare e interrogare la base di dati con facilità e sicurezza. Tale tecnologia infatti permette di inserire, modificare, cancellare ed eseguire altre operazioni attraverso dei comandi specifici, eseguiti tramite delle istruzioni apposite di PHP. I dati salvati riguardano vari aspetti del sito, di cui la maggior parte viene usata per fornire dei servizi aggiuntivi all'amministratore.

6.2 Classi

Ogni entità ha un proprio codice identificativo, anche quando si potevano usare altri attributi come chiave primaria. In questo modo il sistema è solido anche in previsioni di modifiche future, ad esempio perché cambiano le necessità dell'azienda o si presentano dei casi particolari. Di seguito verranno descritte brevemente le entità utilizzate.

- **Clienti:** rappresenta il cliente del Salone, i dati che contiene sono:
 - CodCliente: INT, chiave primaria;
 - Nome: VARCHAR(30);
 - Cognome: VARCHAR(30);
 - Telefono: VARCHAR(10);
 - Email: VARCHAR(50);
 - DataNascita: DATE, il formato salvato in questo caso è del tipo yyyy-m-gg, dove 'y' rappresenta le cifre degli anni, 'm' quelle dei mesi e 'g' quelle dei giorni.
- **TipoAppuntamento:** rappresenta il tipo di lavoro che il cliente può chiedere al Salone di fare. I dati contenuti sono:
 - CodTipoAppuntamento: SMALLINT, chiave primaria;
 - NomeTipo: VARCHAR(30), ovvero il nome dell'intervento;
 - Costo: DOUBLE, rappresenta il prezzo del lavoro;
 - Sconto: DOUBLE, è lo sconto che il responsabile può decidere di applicare al tipo di intervento, a seguito per esempio di una promozione o di un anniversario.
- **Appuntamenti:** rappresenta la prenotazione del cliente per fissare un orario in cui effettuare un determinato intervento. I dati che contiene sono:
 - CodAppuntamento: INT, chiave primaria;
 - CodCliente: INT, chiave esterna;
 - DataOra: DATETIME, è un formato che contiene una data e un'ora, la prima è strutturata in modo uguale a DataNascita di Clienti, l'ora invece ha un formato del tipo hh:mm:ss, dove 'h' rappresenta le cifre dell'ora, 'm' quelle dei minuti e 's' quelle dei secondi;
 - CodTipoAppuntamento: INT, chiave esterna.
- **Prodotti:** rappresenta il prodotto che il Salone usa e vende durante gli appuntamenti. I dati contenuti sono:
 - CodProdotto: INT, chiave primaria;

- Nome: VARCHAR(20), è il nome che identifica il prodotto;
 - Marca: VARCHAR(30), rappresenta il nome dell'azienda che crea il prodotto;
 - Tipo: VARCHAR(30), è il tipo di utilizzo a cui il prodotto è destinato;
 - Quantita: INT, è un valore che distingue i prodotti uguali ma con contenitori di dimensioni diverse, ad esempio per uno stesso tipo di shampoo possono esserci una confezione da 100 ml e un'altra da 250 ml;
 - Prezzo: DOUBLE, indica quanto il Salone spende per comprare il singolo prodotto;
 - PRivendita: DOUBLE, indica quanto il cliente deve spendere per acquistare il prodotto dal Salone.
- **ProdApp**: collega i prodotti agli appuntamenti in cui essi vengono usati, i dati che contiene sono:
 - CodAppuntamento: INT, chiave primaria e esterna;
 - CodProdotto: INT, chiave primaria e esterna;
 - Utilizzo: INT, rappresenta quanto prodotto viene usato durante l'appuntamento, come ad esempio 30 ml di shampoo.
 - **Account**: rappresenta l'account che un amministratore possiede e con cui può accedere al sito. I dati che contiene sono:
 - CodAccount: SMALLINT, chiave primaria;
 - username: VARCHAR(20);
 - password: VARCHAR(32), contiene la versione cifrata della password dell'utente.
 - **Messaggi**: rappresenta il messaggio che il cliente può scrivere all'azienda, tramite l'apposito form presente nel sito. I dati contenuti sono:
 - CodMessaggi: INT, chiave primaria;
 - CodCliente: INT, chiave esterna;
 - Contenuto: VARCHAR(512);
 - DataOra: DATETIME, rappresenta la data e l'ora in cui il server riceve il messaggio;
 - ToRead: BOOLEAN, è un flag che indica se il messaggio è già stato letto o no.
 - **Images**: contiene i dati relativi alle immagini presenti nella galleria fotografica di [public_html/foto.php](#). I dati che contiene sono:
 - Img_title: INT, chiave primaria;
 - Img_desc: VARCHAR(200), rappresenta la descrizione associata all'immagine;
 - Img_filename: VARCHAR(160), è il nome dell'immagine, serve per poter eseguire tutte le operazioni di eliminazione e modifica dell'immagine.
 - **Contatori**: è una View, ossia è una specie di entità i cui dati non vengono inseriti direttamente dall'utente, bensì sono ricavati dai valori di altre entità. In questo caso le informazioni riportate riguardano la statistica degli appuntamenti in base al tipo, gli attributi previsti sono:
 - Parziali BIGINT, è il numero di appuntamenti del tipo con CodTipoAppuntamento uguale all'attributo Tipo;
 - Tipo SMALLINT, è il codice del tipo di appuntamento di cui si vuole visualizzare la statistica.

6.3 Associazioni

Le Associazioni (o Relazioni) rappresentano legami logici fra due o più entità dell'applicazione.

Le Associazioni create al fine di modellare il progetto sono:

- **Appuntamento - Clienti**

Ogni cliente ha uno o più appuntamenti (o nessuno);

Ogni Appuntamento ha un cliente;

Molteplicità 1:N;

Totalità:

- Parziale verso Appuntamenti;
- Totale verso Clienti;

- **Appuntamento - Prodotti**

Durante un Appuntamento possono venir usati più prodotti (o nessuno);

Ogni prodotto può essere usato in più appuntamenti in diverse quantità (per cliente, o anche lo stesso cliente in appuntamenti diversi);

Molteplicità N:M;

Totalità:

- Parziale verso Prodotti;
- Parziale verso Appuntamenti.

- **TipoAppuntamento - Appuntamento**

Ogni TipoAppuntamento può essere usato su più appuntamenti;

Un Appuntamento ha un e uno solo TipoAppuntamento;

Molteplicità 1:N;

Totalità:

- Parziale verso Appuntamento;
- Totale verso TipoAppuntamento.

- **Messaggi - Clienti**

Ogni Cliente ha uno o più messaggi (o nessuno);

Ogni Messaggio ha un Cliente;

Molteplicità 1:N;

Totalità:

- Parziale verso Messaggi;
- Totale verso Clienti.

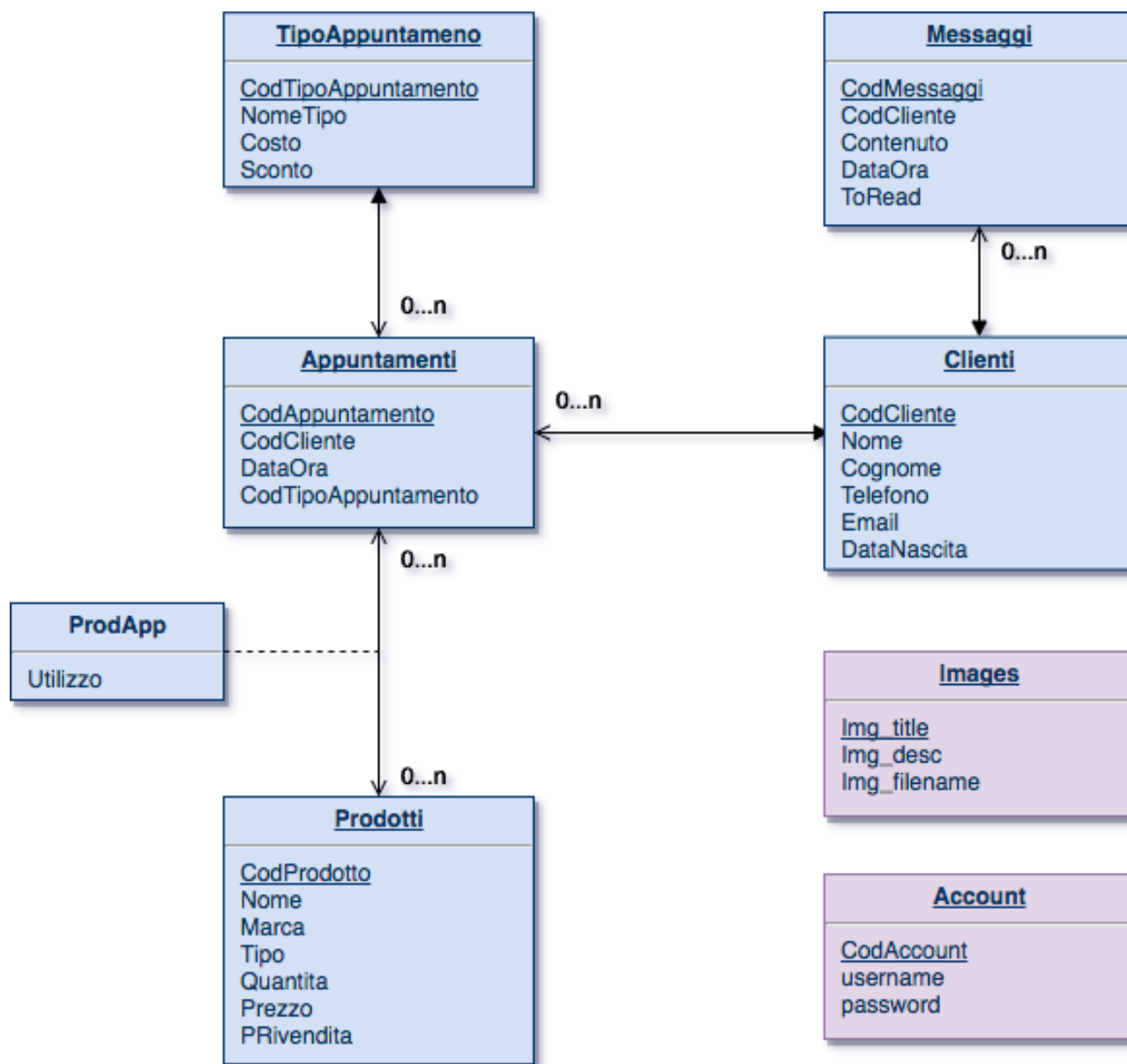


Figure 2: Schema Relazioni

6.4 Codice PHP per l'interazione con il database

Quasi tutte le funzioni PHP che gestiscono l'interazione tra il sito e il database sono contenute nel file "DBlibrary.php" dentro la cartella "utils". Tra di esse le più importanti sono "dbconnect", in cui sono contenute tutte le istruzioni per poter collegarsi al database SQL, in modo da poter eseguire facilmente le query, e "cleanString", che esegue delle pulizie di base sulla stringa passata come parametro, ovvero trasforma i caratteri speciali in entità HTML ed elimina gli spazi e le tabulazioni che possono essere presenti all'inizio o alla fine della stringa.

Tutte le altre funzioni invece rappresentano delle query da dover eseguire sul database, a parte un paio di esse che sono di supporto. È possibile suddividerle in alcuni gruppi, dato che molte di esse lavorano allo stesso modo ma su entità diverse: le funzioni del tipo "lista" (listaClienti, listaAppuntamenti, eccetera) prelevano dal database tutti i dati relativi all'entità a cui fanno riferimento, li salvano in un array di oggetti e restituiscono quest'ultimo all'utente; le funzioni del tipo "aggiungi" inseriscono una nuova istanza nel database in base ai parametri forniti alla funzione, ovviamente non prima che essi vengano controllati; le funzioni del tipo "elimina" cancellano un'istanza del database in base al codice fornito; le funzioni del tipo "aggiorna" modificano un'istanza in base al codice e ai parametri forniti, dopo che questi ultimi vengono

controllati; le funzioni “mostra” restituiscono i dati relativi ad una precisa istanza, in base al codice fornito alla funzione; infine le altre funzioni eseguono delle query più specifiche, che non hanno nulla in comune né tra loro né con le funzioni descritte prima.

I tipi restituiti si possono dividere in due categorie: la prima è rappresentata da un singolo oggetto o un array di oggetti, dove il valore che segnala la presenza di errori a chi chiama la funzione è “NULL”; la seconda è rappresentata da un valore booleano, dove “TRUE” indica che l’esecuzione della query è andata a buon fine, mentre “FALSE” segnala che ci sono stati dei problemi. Questi ultimi possono essere di vario tipo, ad esempio se il database non è accessibile, se la connessione si interrompe all’improvviso o se la query eseguita non è corretta. I controlli sui dati vengono effettuati in ogni funzione che prevede dei parametri, perché ciascuno di essi generalmente viene ottenuto da un input richiesto all’utente, che può essere errato e quindi può generare problemi. In particolare se JavaScript è disattivato avere questi controlli è fondamentale, in quanto non c’è alcuna garanzia che il formato dei dati non faccia danni. Un altro problema è causato dai caratteri speciali, che non possono essere salvati nel database e quindi devono essere trasformati nella corrispondente entità HTML; questa operazione inoltre aiuta la stampa degli oggetti nelle pagine HTML, perché i caratteri speciali risultano essere già in una forma corretta. Nelle stringhe inoltre viene controllato se ci sono degli spazi all’inizio o alla fine, perché sono caratteri inutili che possono comparire per vari motivi, come un errore dell’utente, e possono provocare dei problemi, ad esempio nel database le dimensioni degli attributi sono limitate e quindi dei caratteri di troppo possono rischiare di tagliare una parte dei dati inseriti dall’utente. Per i prezzi viene fatto un controllo di formato apposito, in quanto possono avere 0, 1, 2 cifre dopo la virgola, ma non di più, mentre la parte prima può essere lunga quanto serve; inoltre può essere presente la virgola al posto del punto, in tal caso essa viene sostituita con il punto, garantendo la riuscita della query.

Solo i parametri di tipo stringa vengono controllati, dato che in genere gli altri tipi segnalano errore se sono presenti dei caratteri errati; ad esempio se l’utente deve inserire un numero e aggiunge un carattere speciale allora la query fallirà, in quanto viene inviato un valore di tipo stringa dove è atteso un numero. Un altro caso particolare riguarda le date, che subiscono un controllo del formato molto più severo e quindi non vengono verificate con i metodi delle altre stringhe, in quanto i casi in cui tali istruzioni sono necessarie non passano i controlli usati dalle date. Infine gli attributi “Telefono” di “Cliente” e “Img_filename” di “Images” subiscono un’ulteriore operazione: vengono individuati ed eliminati tutti gli spazi contenuto al loro interno; questo perché altrimenti il loro formato risulterebbe errato rispetto a quello previsto.

7 Presentazione

Per presentare al meglio le informazioni disponibili abbiamo posto la nostra attenzione sulla precisione e l'accessibilità.

Avendo separato contenuto, presentazione e struttura, l'uso del codice CSS ha permesso di curare l'aspetto delle pagine; abbiamo usato per la maggior parte CSS versione 2 e solo alcuni elementi di CSS versione 3 compatibile comunque con la maggior parte dei browser in uso attualmente.

Per rendere migliore la presentazione, abbiamo suddiviso i file CSS in base alle loro funzioni, arrivando ad avere 4 differenti fogli di stile:

- **home.css**, utilizzato per la maggior parte dei dispositivi e browser con risoluzione maggiore, quali computer portatili e fissi;
- **print.css**, destinato a semplificare la stampa delle pagine. Giustifica e modifica il testo, ingrandendo e cambiando il tipo di carattere in uno di più semplice lettura; porta le immagini al centro della pagina. Toglie infine gli sfondi decorativi e sezioni inutilizzabili su carta per ottenere una stampa più chiara.
- **small-devices.css**: viene usato per i dispositivi mobili quali telefoni e tablet che non offrono schermi ampi e richiedono una visualizzazione chiara dell'informazione. Questo foglio di stile viene attivato a risoluzioni inferiori ai 640px. Particolare attenzione si è posta sulle tabelle, infatti la versione mobile, grazie all'uso di comandi CSS versione 3, abbiamo dato scelto una politica reattiva senza l'uso di Javascript e ad usabilità massima.
- **explorer.css**: quest'ultimo foglio è destinato ad *Internet Explorer*, in versione 8 od inferiore; rispetto al foglio di stile home si differenzia per un posizionamento differente di alcuni elementi e per l'aver sostituito gli attributi non compatibili.

Abbiamo usato font particolari, ma mantenendo come seconda scelta prioritaria quelli di sistema, solo nei titoli principali. I caratteri utilizzati hanno una dimensione espressa in "em" al fine di renderli più adattabili alle preferenze dell'utente senza peggiorare l'aspetto del sito. I contenuti del sito si presentano accessibili ed il sito si presenta utilizzabile anche tenendo i fogli di stile disattivati. Nelle sezioni in cui il layout è a più colonne si è scelto di utilizzare percentuali (%), questo per avere un sito più flessibile. Tenendo in considerazione problematiche di auto-posizionamento che alcuni browser subiscono.

Il sito può essere tracciato sulla base del numero di colonne e della loro disposizione ed è stato utilizzato layout a singola colonna (o layout monolitico). I contenuti della pagina sono distribuiti all'interno di un'unica area in quanto non è prevista una sidebar.

Mentre lo sviluppo orizzontale si è scelto il layout fluido, questo effetto si è ottenuto impostando dei valori percentuali per la larghezza del contenitore oppure applicando un semplice padding al corpo della pagina. Capacità di adattamento, cosiddetti layout responsivi (o adattivi). Layout in grado di adattarsi automaticamente alle caratteristiche del display in uso mediante l'utilizzo di vari CSS. L'utilizzo di questa tecnica (responsive design) consente di realizzare un'unica versione del sito web che potrà essere visualizzata efficacemente (e con risultati ottimali) sia sui classici computer che sui moderni device mobili (come smartphone e tablet).

8 Comportamento

Le funzionalità offerte tramite Javascript sono:

- Il pulsante per risalire ad inizio pagina, creato tramite una funzione jQuery;
- L'inserimento della mappa nella pagina "contattaci" ottenuta aggiungendo all'immagine nella pagina, la cartina fornita da Google.
- Il controllo lato client dei dati inseriti nella form nelle pagine Contattaci, nelle pagine da cui modificare e creare un appuntamento od un cliente, nella pagina da cui creare un prodotto ed in quelle da cui far iniziare una ricerca su di un cliente od un giorno particolare. Questo è stato fatto creando delle matrici per ogni form che si desiderava controllare, al cui interno erano riportati i nomi dei dati e le espressioni regolari che ne stabilivano la validità; fatto questo, sono state abbinate a delle funzioni che data la matrice dei dati, si occupavano d'invocare una funzione generica che s'occupa di reperire il dato all'interno della pagina, effettuare il controllo e segnalare gli errori trovati. Le funzioni invocate dalle form si occupano di impedire l'invio dei dati segnalando anche la presenza di errori vicino al pulsante d'invio della form.

Sapendo che Javascript più può non essere disponibile od essere stato disabilitato dall'utente:

- abbiamo sviluppato le pagine in modo da fornire messaggi d'errore qualora le invocazioni delle funzioni PHP che s'occupano della modifica dei dati non vadano a buon fine; questo per tutte le pagine che prevedevano un'interazione con l'utente.
- inserito all'interno della pagina contattaci, un tag `<noscript>` contenente degli aiuti alla compilazione della form.

Avendo utilizzato JS per un limitato numero di funzionalità la cui assenza non preclude l'uso del sito (la mappa è presente come immagine, mentre gli errori sugli inserimenti sono segnalati in ogni caso), riteniamo di aver raggiunto l'obiettivo di garantire un degrado elegante delle pagine, in assenza del supporto a Javascript da parte del browser in uso.

9 Accessibilità

9.1 Implementazione

Al fine di garantire l'utilizzo del sito ad utenti con disabilità, si sono:

- Validati i file che compongono il sito con i validatori XHTML1.0-Strict e CSS3 del W3C;
- Separata struttura, presentazione e comportamento;
- Fonita la possibilità per la tecnologia assistiva di ignorare contenuti non testuali puramente decorativi;
- Potuti ridimensionare i testi fino al 200%, senza perdita di contenuto e funzionalità;
- Creati contenuti rappresentabili in modalità differenti senza perdita d'informazioni o struttura;
- Utilizzato un rapporto di contrasto di almeno:
 - 7:1 tra testo e sfondo;
 - 4,5:1 tra testo grande ed immagini contenenti testo grande nella home;
- Reso le funzionalità del sito utilizzabili tramite tastiera, mediante i tabindex e accesskeys;
- Inseriti:
 - informazioni relative alla posizione dell'utente (breadcrumbs);
 - titoli appropriati per le pagine web;
 - testo appropriato e testo alternativo per i collegamenti;
 - per quanto riguarda le tabelle, è stato:
 - * Aggiunto attributo **summary** ed elemento **caption**;
 - * Aggiunti THEAD, TBODY; per tabelle che si prevedeva particolarmente lunghe anche TFOOT.
 - * Inseriti elementi *th* per le intestazioni e riservati *td* per i soli dati.
 - * Utilizzato l'attributo **scope** per rendere esplicita la relazione tra le celle d'intestazione e le celle di dati: indicando l'estensione di celle per cui l'intestazione si trova nella cella marcata con l'attributo **scope**.
Abbiamo marcato **scope** con il valore *col* le celle d'intestazione, assicurando che le informazioni di intestazione si applichino a tutte le celle della colonna; col valore *colspan*, invece, le celle d'intestazione relative ad un gruppo di colonne.
 - * Utilizzato serie di **id** e di **headers**: il primo per identificare una cella che contiene informazioni di intestazione a cui fare riferimento con headers serve per risalire alle celle d'intestazione da porre in relazione con la cella corrente.
Questo per aumentare la comprensione dei contenuti, diminuire il disorientamento ed agevolare l'utente che fa uso di screen reader, visto che questi ultimi che tengono traccia delle intestazioni.
 - alternative testuali per il contenuto non testuale:
 - * corredata ogni immagine portatrice di contenuto, con l'attributo alt che la descrive;
 - * aggiunta di una label ad ogni campo di input della form, in aiuto dello screen reader; corredata ogni form di una legend, per i medesimi motivi.
 - intestazioni ed etichette appropriate;
 - l'indicatore del focus nelle interfacce utilizzabili da tastiera;

- lingua predefinita per il contenuto delle pagine; usati gli attributi `xml:lang` per definire parole o blocchi in lingua diversa da quella predefinita della pagina; non sono state utilizzate abbreviazioni od acronimi;
 - definiti i tag meta: *Description*, *Keywords*, *Copyright*, *Author*; le parole chiave contenute nel tag "keywords" sono in lingua italiana, visto il pubblico a cui il sito si rivolge.
 - È stato definito infine un link di ritorno ad inizio pagina.
- Mantenuto un meccanismo di navigazione coerente all'interno delle pagine web del sito, grazie al template ottenuto con le funzioni PHP;
 - Individuati eventuali errori di inserimento e descritti con del testo; fornite etichette (o istruzioni) per l'input dell'utente.
 - Inserite modalità per saltare i blocchi di contenuto che si ripetono su più pagine;

Inoltre, non si sono:

- inseriti contenuti audio e video (contenuti multimediali basati sul tempo);
- posti vincoli di tempo all'utente per consultare i contenuti o compilare i campi dati;
- utilizzato il colore come modalità visiva per rappresentare le informazioni, indicare azioni, elemento di distinzione visiva; inserito contenuto audio eseguito automaticamente all'interno della pagina;
- sviluppati contenuti che possano causare attacchi epilettici (non s'è inserito contenuto lampeggiante);
- inseriti cambiamenti del contesto su alcun componente che riceve il focus;

Al fine di facilitare l'utilizzo del sito da parte di utenti con disabilità, si è:

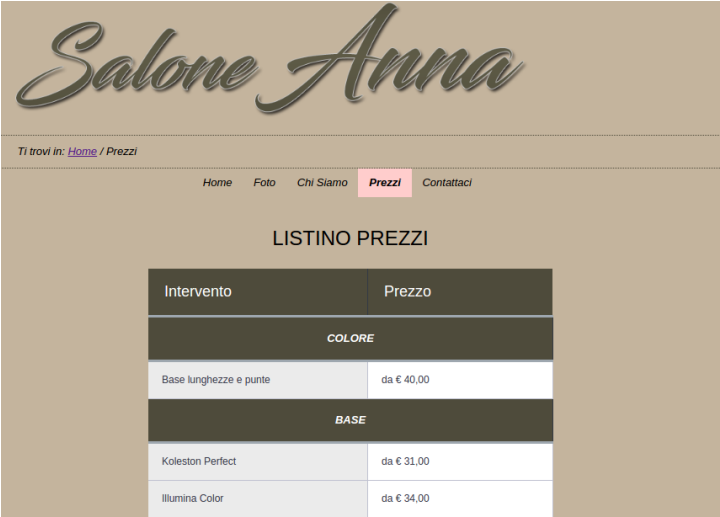
- aiutata la navigazione tra le pagine, con le seguenti tecniche:
 - *un path o breadcrumb*, per individuare il contesto;
 - *un menù di link* per mostrare dove si può andare;
 - *uso dello stesso stile* per tutti i link del sito;
 - *link di ritorno ad inizio pagina*.
 - ridefiniti i tabindex per la navigazione tra i vari link.
 - definite scorciatoie con *accesskey* per la navigazione tra le pagine.
- Mantenuti chiari i link.
- Creato scorciatoie per dispositivi mobili per favorire la navigazione.

9.2 Combinazione dei colori

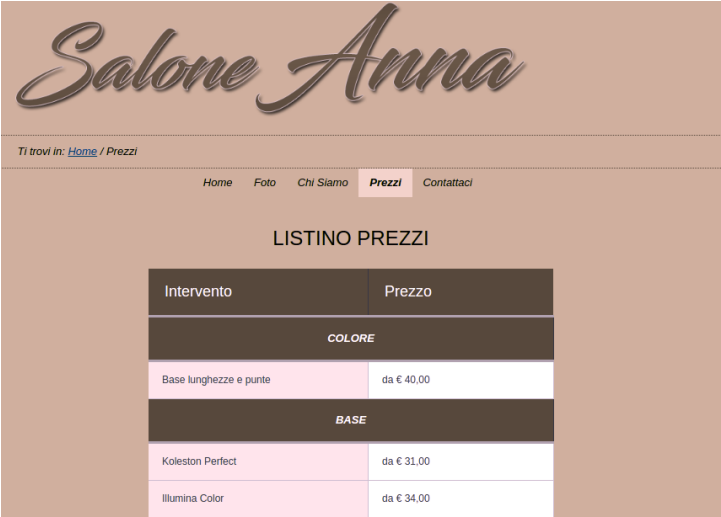
È stata utilizzato uno schema di colori che garantisca un contrasto di almeno 7:1 tra sfondo e testo; per testare le scelte fatte è stato utilizzato il servizio offerto da <http://wave.webaim.org>.

Il servizio offerto da <http://www.color-blindness.com/coblis-color-blindness-simulator/> ha permesso di capire come utenti affetti da disturbi nella visione dei colori visualizzino il sito.

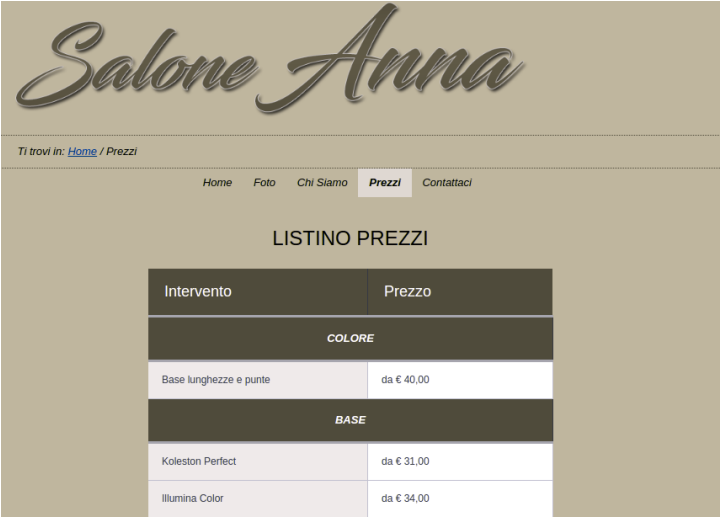
Di seguito vengono riportati i risultati ottenuti sulla pagina relativa ai prezzi, che contiene molti elementi utilizzati anche nelle altre pagine.



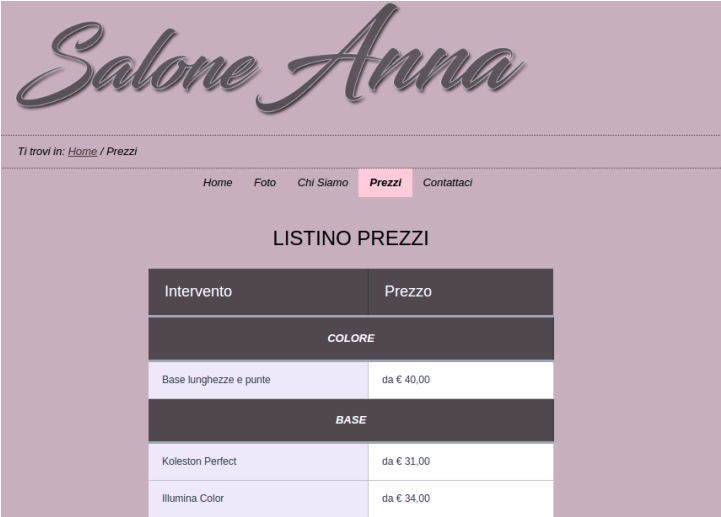
(a) Pagina originale



(b) Pagina vista da un deutranope



(c) Pagina vista da un protranope



(d) Home Page vista da un tritranope

Figure 3: Pagina vista da persone con problemi nel distinguere i colori

Abbiamo giudicato il risultato accettabile.

10 PHP

10.1 Descrizione generale

Il linguaggio PHP è stato utilizzato nel sito per moltissimi scopi, dalla stampa delle pagine HTML all'interazione con il database. Abbiamo implementato una parte front-end, che si occupa di costruire direttamente le pagine e passare i dati ad un back-end che esegue queries sul database e passa al front-end i risultati di queste operazioni.

La parte front-end si trova al primo livello della cartella “public_html”, mentre le funzioni che realizzano il back-end risiedono nella cartella “utils”.

Nelle sottosezioni successive verranno approfonditi i vari casi in cui questo linguaggio è stato usato, ad eccezione dell'interazione con il database che viene spiegata nella sezione GestioneDati.

10.2 Pagine XHTML

Il codice PHP è stato usato per semplificare la creazione delle pagine XHTML, ad esempio racchiudendo in apposite funzioni le parti di codice comuni a tutte le pagine, contenute nel file “library.php”: in questo modo la creazione della pagina XHTML risulta più semplice, veloce e facile da modificare; ad esempio se risulta necessario cambiare qualcosa nell'header o nel footer della pagina basta modificare il codice contenuto nell'apposita funzione, e in automatico verrà applicata la modifica a tutte le pagine. Nel caso di sezioni in cui alcune parti cambiano, come ad esempio la definizione del doctype e dei tag meta, in cui le parole chiavi, il nome della pagina e altri dati sono diversi per ogni singolo caso, sono stati previsti dei parametri, in modo da dover definire ogni volta solo le parti che cambiano.

Un'altra semplificazione che è stata resa possibile dall'utilizzo di PHP è la modifica della grafica delle pagine quando risulta che l'utente abbia effettuato il login: anziché dover creare una copia della pagina riadattata al caso o scrivere una funzione che modifichi ogni elemento da cambiare viene utilizzata una variabile booleana, il cui valore è determinato a inizio file, che indica quale versione della pagina bisogna caricare; quando è necessario sapere cosa stampare basta analizzare il valore di tale variabile. Questo metodo è particolarmente efficace anche perché spesso le differenze tra le due versioni della pagina sono molto piccole, ad esempio possono differire per una semplice scritta o una certa struttura.

10.3 Sessioni

Le sessioni sono state usate per permettere all'utente di eseguire il login come amministratore; dal punto di vista del sito determinano se è possibile o meno rendere visibili le sezioni dedicate alle operazioni che un amministratore può eseguire, come la modifica del database. La gestione delle sessioni avviene interamente in PHP e comincia con il file “login.php”, contenuto nella cartella “public_html”, dove la nuova sessione viene creata se l'utente inserisce uno username e una password corretti. L'unico controllo effettuato su questi ultimi è la verifica della corrispondenza con i dati degli account presenti nel server, se uno di essi risulta valido allora avviene la creazione della sessione; in particolare non viene verificato cosa contengono i due input in quanto il controllo della corrispondenza risulterà falso se il loro formato non è corretto. Se la login ha successo allora viene mostrato un avviso apposito, altrimenti viene stampato il tipo di errore che si è verificato. Per comodità il codice che stampa queste due versioni è contenuto nello stesso file, ed è proprio per questo che esso è contenuto nella cartella “public_html” e non “utils”, perché altrimenti bisognerebbe modificare dinamicamente tutti i link della pagina da stampare, come ad esempio quelli relativi ai file CSS. Per verificare se l'utente ha effettuato la login c'è la funzione “authentication”, contenuta nel file “library.php” dentro “public_html”; di fatto essa controlla se è stata istanziata la variabile “username”, contenuta dentro la sessione solo se essa è stata creata, e se il tempo di vita della sessione è scaduto, tramite il metodo descritto sotto, e restituisce il valore booleano corrispondente. Per limitare il tempo di vita della sessione, in fase di login viene salvata una variabile in cui è memorizzata l'ora di inizio; ogni

volta che viene effettuato un controllo dalla funzione **authenticate**, se è attiva la sessione viene invocata la funzione “checkSessionLifetime”, contenuta nel file “library.php”, in cui viene verificato se l’orario presente ha un valore inferiore a mezz’ora, rispetto all’ora corrente: in caso affermativo, la sessione viene eliminata e viene restituito “FALSE”, altrimenti viene riscritta la variabile contenente l’orario, con l’ora corrente e viene restituito “TRUE”.

Infine per effettuare il logout c’è il file “logout.php”, contenuto dentro a “utils”, in cui prima vengono distrutte le variabili della sessione e poi la sessione stessa.

11 Verifica e Test

Al fine di riuscire a garantire una corretta visualizzazione del sito ed una sua fruizione da parte di un numero di browser quanto più ampio possibile, è stata verificata la validità di tutte le pagine, indipendentemente dal tipo statico o dinamico; si è infine visualizzato il sito su browser meno recenti.

11.1 Ambiente di lavoro

Dall'inizio dello sviluppo abbiamo utilizzato il servizio di hosting GitHub per ospitare il nostro progetto e lavorare in contemporanea, potendo eventualmente far regredire il prodotto in caso di modifiche errate. Questo ci ha permesso di poter lavorare anche da remoto avendo sempre la versione più aggiornata a disposizione.

Il repository può essere trovato al seguente link: <https://github.com/sebastianomarchesini/SaloneAnna>.

11.2 Dispositivi utilizzati

Oltre a testare il sito sui computer del laboratorio, per verificarne la compatibilità su un gran numero di dispositivi ci siamo affidati ai servizi offerti da [BrowserStack](#), correggendo di volta in volta gli errori trovati. Di seguito l'elenco dei browser e dispositivi testati:

- Chrome 56.0.2924.76, Opera 39.0.2556.71, Firefox 51.0.1;
- Edge 13, IE 11, IE10, IE9, IE8, IE7
- Opera 42.0
- Safari versione 10.
- Dispositivi mobili: Apple Iphone 5 e 6Plus, Google Nexus 6, 9, Nokia Lumia 920, Samsung Galaxy Y, Google Nexus 7, Apple Ipad Mini.

11.3 Verifica del codice prodotto

Durante lo sviluppo il codice PHP è stato verificato usando <http://phpcodechecker.com/> e <http://phpfiddle.org/>

Per la verifica del codice XHTML e CSS prodotto ci siamo affidati ai validatori messi a disposizione dal W3C.

Tutto il codice XHTML prodotto è stato validato correttamente alla versione XHTML 1.0 Strict, senza errori.

Il codice CSS è stato validato in versione 2, 2.1 e 3; CSS 3 ha validato correttamente i fogli di stile prodotti, senza errori, nè warning; CSS 2 e CSS 2.1 riportavano errori determinati da alcune istruzioni non pienamente compatibili, che però non ostacolano la visualizzazione o l'accessibilità del sito. Abbiamo fatto in modo di avere un sito fruibile ed utilizzabile anche in assenza di fogli di stile.

Il codice Javascript è stato analizzato tramite <http://jshint.com/>, superando il controllo.

Per essere certi d'aver prodotto un sito web accessibile a tutte le tipologie d'utente, sono stati usati i seguenti validatori, che effettuano un controllo run-time del sito:

- [Achecker](#);

- wave.webaim.org;

Tutte le pagine del sito sono risultate valide.

Siamo riusciti ad ottenere la certificazione WCAG 2.0 AAA, la più alta possibile. Al momento della validazione sono stati prodotti dei warnings, ma solo nella parte riguardante il contenuto.

11.4 Conclusioni

Il sito, in seguito alle validazioni, si è dimostrato accessibile e compatibile con un ampio numero di browser e dispositivi; ci riteniamo soddisfatti del lavoro svolto.