

UNIVERSITÀ DEGLI STUDI DI PADOVA

CORSO DI LAUREA IN INFORMATICA

TECNOLOGIE WEB

Copisteria Berton

Componenti del Gruppo “Copisteria Berton”:

BERTON Franco - 1052574

CARNOVALINI Filippo - 1048335

FERRARA Alberto - 1049378

VAROTTO Mattia - 1051619

15 Gennaio 2016

*Il sito è caricato sullo spazio dell'utente **aferrara**
Account di accesso all'area di amministrazione del sito:*

AMMINISTRATORE:

Username: **admin**

Password: **pwd**

Indice

1	Introduzione	2
1.1	Scopo del sito	2
1.2	Classi di utenza	2
1.3	Pagine	2
2	Codice Perl	4
2.1	Suddivisione del codice	4
2.2	Pagine Dinamiche	4
2.3	Controlli sui Form	5
3	HTML	5
3.1	Contenuto del tag header	6
3.2	Esperienza utente	6
3.3	Validazione	6
4	CSS	7
4.1	Responsività	7
4.2	Suddivisione del codice	7
4.3	Validazione	7
5	Javascript	8
5.1	Validazione form	8
5.2	Ricerca tra i servizi	9
6	Accessibilità	9
6.1	Pratiche a livello di html	9
6.2	Pratiche a livello di CSS	10
6.3	Strumenti usati	10
7	Altri Aspetti	11
7.1	Database	11
7.2	Sicurezza	11
7.3	Norme	11
7.4	Suddivisione del lavoro	11

1 Introduzione

La presente relazione spiega i criteri, le tecniche, e le tecnologie usate per la creazione del sito della fittizia copisteria Berton, applicando i buoni principi della progettazione e programmazione per il Web, seguendo gli standard del W3C.

1.1 Scopo del sito

Lo scopo del sito è quello di fornire visibilità sul web all'ipotetica copisteria Berton.

Devono dunque essere presenti pagine in cui vengono dati i contatti e le indicazioni per trovare la copisteria, si deve dare il listino dei servizi offerti, nonché una breve storia della copisteria per migliorarne l'immagine. Si vuole dare inoltre la possibilità ai visitatori di lasciare commenti relativi alla copisteria in un Guestbook.

Per dei superutenti, deve essere possibile modificare il listino e moderare i commenti lasciati dai visitatori eliminando quelli indesiderati.

1.2 Classi di utenza

Agli utenti visitatori sarà possibile visitare ogni pagina del sito, eccetto quella di amministrazione. A loro sarà possibile effettuare ricerche tra i servizi presenti in listino, e potranno lasciare commenti al sito tramite un apposito form. Non potranno però modificare o eliminare commenti inseriti.

Agli utenti amministratori dotati di appositi dati di accesso sarà invece possibile modificare la lista dei servizi aggiungendone di nuovi oppure modificando o eliminando quelli presenti. Questi utenti potranno anche eliminare commenti tra quelli lasciati sulla pagina apposita. Potranno infine creare altri utenti amministratori creando nuovi dati di accesso. Non è però possibile modificare i propri dati di accesso o eliminare amministratori.

1.3 Pagine

Visitando la mappa del sito (per la quale è presente un link in calce a tutte le pagine) si trovano link a tutte le pagine del sito, inclusa quella di amministrazione. Tuttavia, se non si ha effettuato l'accesso al sito si verrà rimandati al login.

Le pagine del sito sono:

Home La pagina principale del sito è molto semplice, e non contiene molto contenuto, solo un messaggio di benvenuto e una breve introduzione alla copisteria, ma presenta (come ogni pagina) il pannello di navigazione per accedere alle altre pagine. Inoltre, nel blocco principale è presente un altro link alla pagina dei servizi, considerata la pagina più interessante per un visitatore;

Servizi In questa pagina è presente una lista dei servizi offerti dalla copisteria, corredata da una breve descrizione e dal prezzo del servizio. Se javascript è attivo, viene visualizzato anche un form per filtrare i servizi presenti in base a parole chiave. Accedendo come amministratore è presente anche un form per l'aggiunta di servizi e ogni elemento della lista è corredato da pulsanti per poter eliminare o modificare quell'elemento;

Storia La pagina di storia presenta un'immagine della copisteria, e una breve storia (a carattere più pubblicitario che storico) della stessa;

Dicono di Noi È la pagina del Guestbook, dove è presente la lista dei commenti dei visitatori del sito e un form per aggiungere nuovi commenti. Agli amministratori è disponibile anche un pulsante per eliminare i commenti;

Contatti Una pagina che riporta i contatti della copisteria, inclusa una mappa, e i nomi del personale della copisteria;

Informativa Cookie Pagina contenente le informazioni sui cookie usati dal sito prescritta dal garante della privacy (vedi par. 7.3);

Mappa del Sito Contiene link a ogni pagina del sito;

Amministrazione Se non si è acceduti viene visualizzata la pagina di login, altrimenti viene visualizzata una pagina contenente un form per aggiungere nuovi amministratori e rimandi alle altre pagine dove gli amministratori hanno poteri speciali (ovvero la pagina dei Servizi e quella del Dicono di Noi).

2 Codice Perl

In questa sezione vengono esposti i criteri e le pratiche con cui è stato pensato il codice del server, esponendone velocemente i contenuti.

2.1 Suddivisione del codice

Per la generazione del codice html delle pagine è stato creato un file chiamato `template.cgi` contenente funzioni per la generazione delle parti comuni di ogni pagina, come il contenuto del tag `head`, l'header della pagina, la barra di navigazione e il footer. Queste funzioni accettano argomenti per la differenziazione delle pagine, ad esempio la funzione per la generazione dell'head accetta argomenti per definire i tag meta, o la funzione che genera l'header e il breadcrumb prende in argomento la stringa che costituisce il breadcrumb vero e proprio. Queste funzioni non stampano direttamente il loro output ma restituiscono la stringa generata, in modo tale che se fosse necessario fare ulteriori modifiche (ad esempio per eliminare link circolari) sia possibile farlo. Esistono poi tanti file `.cgi` quante le pagine visitabili del sito, che richiamano le funzioni del file `template.cgi` e poi generano direttamente il codice html che va a riempire il blocco del contenuto principale della pagina (il div avente `id="content"`). Le pagine che effettuano operazioni su database si avvalgono a volte di file `.cgi` aggiuntivi per gestire queste operazioni (sono i file `add_admin.cgi` e `check_login.cgi`), che non generano codice html ma si limitano a reindirizzare. Esiste anche un file `logout.cgi` che gestisce le operazioni di distruzione di una sessione e reindirizza verso la home del sito. L'ultimo file perl rimasto è il file `funzioni.pl`, pensato per contenere le funzioni perl usate da più pagine ma che non generano codice html. Alla fine del progetto questo file contiene solo una funzione per eliminare caratteri indesiderati dall'input degli utenti.

2.2 Pagine Dinamiche

Anche se tutte le pagine sono dinamiche in quanto generate su server, solo alcune pagine possono variare notevolmente oltre alla presenza del messaggio che notifica l'aver effettuato accesso al sito come amministratore. Queste pagine sono quella dei servizi e quella del guestbook. In ambo le pagine è possibile effettuare operazioni di inserimento tramite un apposito form, di eliminazione, e sulla pagina dei servizi

anche di modifica. In questo caso tutte le operazioni sono gestite dallo stesso file .cgi che genera la pagina. In seguito ad ogni operazione, viene visualizzata una pagina con un messaggio di successo, e un'ancora per tornare a vedere la pagina vera e propria. L'altra pagina che mostra grandi differenze è la pagina di amministrazione. Quella che all'utente è presentata in maniera trasparente sempre come pagina di amministrazione è in realtà, a seconda del fatto che si sia autenticati o meno, la pagina login.cgi o la pagina admin.cgi. La prima, come suggerisce il nome, è un form di login per autenticarsi, la seconda invece è la pagina di arrivo una volta effettuato l'accesso e da cui è possibile aggiungere nuovi amministratori al sito. Contiene poi dei rimandi alle pagine dei servizi e del guestbook, dove l'amministratore può effettuare operazioni speciali. La pagina login.cgi controlla la presenza di una sessione e se la trova rimanda alla pagina admin.cgi, e viceversa se quest'ultima non trova una sessione aperta rimanda a login.cgi.

2.3 Controlli sui Form

Nel sito sono presenti varie form, e l'input di tutte queste è controllato tramite javascript come spiegato nella sezione 5.1. Tuttavia, data la poca affidabilità dei controlli lato client sono stati effettuati controlli anche lato server. Questi controlli non sono sempre user-friendly quanto quelli in javascript: in particolare non sempre vengono mostrati tutti gli errori presenti contemporaneamente. Questo è considerato meno grave data la presenza di controlli lato client appositamente pensati per migliorare l'esperienza d'uso degli utenti. Inoltre è importante notare come, prima di inserire qualsiasi input utente nel database, questo input è passato attraverso la funzione *sanitizza()* presente nel file funzioni.pl, per assicurarsi che nel database non possano entrare caratteri maliziosi per XML, per quando possano essere sfuggiti ad ogni controllo. (i caratteri in questione sono '<' e '>', riservati per i tag, e '"', attraverso il quale è possibile definire entità).

3 HTML

Per questo progetto è stato scelto di creare tutte le pagine in XHTML, perché era l'opzione meglio supportata dai vari browser. L'unico file avente estensione .html è

il file `index.html`, che reindirizza semplicemente alla home page. Si parlerà dunque di pagine html intendendo l'output dei relativi file `.cgi`.

3.1 Contenuto del tag header

Come già detto il tag head delle pagine del sito viene generato da una delle funzioni del file `template.cgi`. Questa funzione è stata parametrizzata in modo da rendere comunque possibile specificare per ogni pagina il tag title, e i tag meta title, keywords e description della pagina, in modo da ottimizzare ogni specifica pagina per i motori di ricerca. Per i tag title è stato scelto di includere (nell'ordine) il nome della pagina e il nome del sito, senza includere tutto il percorso della pagina, dal momento che molti browser mettono a disposizione solo poco spazio per la visualizzazione del title (per lo stesso motivo il nome della pagina viene scritto prima di quello del sito). Le keywords sono state scelte con parsimonia facendo attenzione che fossero anche presenti nel corpo della pagina, per non essere mal visti dai motori di ricerca, e la description è sempre stata mantenuta molto breve affinché non venisse tagliata.

3.2 Esperienza utente

Per migliorare la navigabilità del sito è stato inserito un breadcrumb in ogni pagina. Anche questo viene generato a partire da un parametro dalla funzione che genera l'header “visibile” del sito. In ogni pagina è inoltre presente un link alla home nascosto nel logo del sito (che per convenzione nota agli utenti rimanda alla home) ed è stato inserito in ogni pagina un footer con delle informazioni generali sul sito, e un link alla mappa dello stesso. È stata fatta grande attenzione a non inserire mai link circolari, che possono indurre l'utente in confusione.

3.3 Validazione

Ogni pagina è stata validata con il validatore online del W3C per lo standard XHTML 1.0 Strict, prestando attenzione alle varie “alternative” che potevano essere generate dal server di uno stesso indirizzo.

4 CSS

Per definire lo stile delle pagine html è stato fatto uso di CSS, incluso negli header delle varie pagine html.

4.1 Responsività

Nel creare il codice CSS per definire lo stile delle pagine è stata fatta attenzione ad usare il meno possibile riferimenti a distanze fisse espresse in pixel, preferendo distanze percentuali per avere un design quanto più possibile responsivo. È stata fatta attenzione anche al definire variazioni del CSS (tramite regole di tipo @media) quando la risoluzione del browser o del dispositivo su cui viene visualizzato il sito sia minore a certi limiti, per garantire una buona visualizzazione anche su mobile. In particolare è stata fatta molta attenzione alla pagina dei servizi, in modo da definire regole che visualizzassero sempre in forma “tabellare quello che in realtà è il contenuto di una lista, gestendo con attenzione la quantità variabile di elementi per riga, anche sfruttando regole definite per visualizzazioni intermedie nel file style.css.

4.2 Suddivisione del codice

Sono stati creati quattro file .css: i file style.css e small.css servono a definire lo stile del sito in visualizzazione normale (il file main.css) e nel caso di schermi piccoli (il file small.css).

Il terzo file, print.css, definisce delle regole di CSS da applicare solo in caso di stampa. È stato definito allora un file specifico per la stampa in modo tale che nella versione cartacea del sito non apparissero tutti i link presenti (che sarebbero inutili) ma solo il contenuto informativo della pagina.

L'ultimo file, aural.css, è stato usato per i media aural e speech. Sarà trattato meglio nella sezione 6.2.

4.3 Validazione

Tutto il codice CSS è stato validato tramite il validatore W3C per lo standard CSS3. Una volta validato per CSS3 è stato effettuato anche un controllo di va-

lidazione secondo CSS2, che valida la grande maggioranza del codice, eccetto i `border-radius`, che sono comunque supportati da molti browser anche non recenti e per i quali sono state incluse anche le varianti dei vari vendor, il valore `inline-block` per la proprietà `display`, anch'essa interpretata comunque correttamente da quasi la totalità dei browser anche non recenti, i font aggiunti e la proprietà `resize` per le textarea. Anche queste due ultime cose sono spesso interpretate correttamente, e comunque non comportano alterazioni capaci di rovinare l'aspetto del sito se non vengono intepretate.

5 Javascript

Il sito non fa uso massivo di javascript, ma è stato comunque usato in due casi per migliorare l'usabilità dello stesso. Tutte le funzioni usate sono contenute nel file `scripts.js`, incluso in ogni pagina. È stato inoltre incluso jQuery (nella sua versione 1.11.3 minimizzata) perché usato da alcune funzioni.

5.1 Validazione form

Come già accennato, ogni form viene validata tramite javascript prima che i suoi dati vengano inviati al server. Ogni form usa una specifica funzione per controllare gli errori, ma il funzionamento di base è comune a tutte: gli errori vengono segnalati sostituendo commenti html con span aventi la classe `error`. Questo sistema è stato usato per rendere totalmente compatibili gli errori generati via javascript con quelli generati via perl, infatti in ambo i casi sono state usate espressioni regolari per la scrittura di questi messaggi di errore. La prima cosa che queste funzioni effettuano è la pulizia di eventuali errori già presenti: tramite espressioni regolari ripristinano i commenti html originali, se questi erano stati sostituiti da messaggi. Dopodiché, viene verificato per ogni input se presenta errori, e questi errori vengono segnalati. Infine, viene restituito falso se è presente almeno un errore, e vero altrimenti. Questo dato booleano è usato per decidere se inviare o meno al server i dati del form.

5.2 Ricerca tra i servizi

Nella pagina dei servizi è stata inclusa una funzionalità (facente uso di jQuery) di filtraggio dei servizi presenti: un piccolo form prende una stringa di input, e la cerca nel contenuto html dei vari servizi, e nasconde quelli che non la contengono. Questa ricerca è implementata solo in javascript, e dunque non può funzionare se questo è disabilitato. Per questo motivo l'html generato da perl non include questo form, ma un messaggio che avverte che la ricerca è disponibile solo con javascript attivato. Se è effettivamente attivato, una funzione javascript effettuerà subito una modifica al DOM in modo da mostrare il form della ricerca. Di fatto, se javascript è attivo il messaggio d'errore non sarà visualizzato.

6 Accessibilità

Oltre alle buone pratiche di programmazione web, mantenendo separate struttura, presentazione e comportamento, e rispettando gli standard dettati dal W3C, sono state usati degli ulteriori accorgimenti e degli strumenti per valutare come il sito può essere percepito da determinate classi di utenza.

6.1 Pratiche a livello di html

È stata prestata attenzione alla localizzazione del sito: la lingua del sito dichiarata è l'italiano, ma ogni volta che un termine straniero appariva nel sito (sono capitati esclusivamente termini inglesi) questo è stato incapsulato in uno span (o a seconda dei casi altri tag) che specificasse la diversa lingua con l'apposito attributo lang. In certi casi sfortunatamente questo non è stato possibile, ad esempio nei tag dell'head per via delle specifiche dettate dal W3C.

In tutte le pagine sono stati specificati i tabindex in modo tale che premendo il tabulatore si fosse portati a selezionare nell'ordine più naturale i vari elementi della pagina, partendo da quelli del contenuto e solo dopo seguendo quelli di navigazione. Le immagini presenti sul sito sono state descritte nel loro attributo alt.

Ogni pagina dispone di un “torna su visibile, che rimanda all'inizio del div content, e altri aiuti alla navigazione nascosti ma comunque letti dagli screen reader.

6.2 Pratiche a livello di CSS

I colori del sito sono stati scelti in modo che passassero sempre i livelli di contrasto e luminosità prescritti dal WCAG, facendo riferimento allo standard AAA (ma accettando anche un livello AA).

Sono presenti dei testi di aiuto che non vengono visualizzati su browser ma che sono letti da screen reader. Questi testi (sono la classe `.reader`) non hanno regole sul foglio di stile aural, ma dato che pochi browser e screen reader supportano correttamente i fogli di stile aural e speech non è bastato eliminare (tramite `display:none`) questi testi nel foglio di stile normale. Per questo motivo nel file `style.css` hanno visibilità `collapsed`, ma anche questo valore non è interpretato correttamente in molti browser, e dunque sono state aggiunte le proprietà `position:absolute`; e `left:-100000px`; che permettono di ottenere l'effetto desiderato anche se non elegantemente quanto si sarebbe desiderato.

6.3 Strumenti usati

Per poter valutare come il sito dovesse essere interpretato da uno screen reader, è stato usato il plugin per Firefox Fang, che produce un testo simulante ciò che uno screen reader leggerebbe ad un eventuale utente non vedente, per renderci conto se alcune informazioni erano date per scontate per via di indicazioni visive.

Nello scegliere i colori da usare nel sito invece è stato usato il servizio presente all'indirizzo <http://www.accesskeys.org/tools/color-contrast.html> (è stato però necessario caricare su un hosting online il sito per poterlo usare). Questo tool permette una rapida valutazione generale, ma non riesce a controllare effettivamente tutti i contrasti presenti. Per controllare quelli mancanti è stato usato il tool Color Contrast Analyzer.

Infine, è stato fatto uso di Total Validator per validare ogni pagina nonostante fossero già state validate con il validatore ufficiale W3C, perché questo strumento segnala eventuali accorgimenti che possono essere utili per rendere il sito più accessibile.

7 Altri Aspetti

7.1 Database

Il database è stato implementato in XML, ed è stato diviso in tre file: DatabaseAmministratori.xml, che contiene gli username e le password degli amministratori del sito, DatabaseCommenti.xml, che contiene i commenti presenti nel Guestbook, e DatabasePrezzi.xml, che contiene i dati della pagina dei servizi. Anche se non vengono mai sfruttati dal sito, sono allegati i file .xsd con i quali i database sono stati validati.

7.2 Sicurezza

Le password nel sistema sono sempre criptate con l'algoritmo SHA256 nella sua variante a cifre esadecimali. Tuttavia, per avere una completa sicurezza il sito dovrebbe godere di una connessione sicura (tramite protocollo https), ma questo ovviamente non rientra nelle possibilità e nello scopo del progetto.

7.3 Norme

In accordo con i recenti cambiamenti della legislazione in materia, è stato valutato come rendere il sito a norma con i dettami della Cookie Law. Dal momento che il sito utilizza solo cookie tecnici, non è stato necessario inserire un banner con una informativa in forma breve e chiedere autorizzazione all'utente per l'utilizzo dei cookie, ma è stato necessario includere un'informativa estesa conforme alle indicazioni del Garante della Privacy, accessibile da un link incluso nel footer di ogni pagina del sito (come indica di fare il Garante).

7.4 Suddivisione del lavoro

Il carico del lavoro sul sito è stato suddiviso sui quattro membri del gruppo in quantità tentativamente uguali:

Berton Franco ha curato i fogli di stile CSS e la pagina dei servizi, inclusi i suoi aspetti lato server.

Carnovalini Filippo ha creato il file `template.cgi`, e si è occupato della validazione di tutto il codice e dei controlli e delle correzioni necessarie per garantire l'accessibilità. Ha fatto la funzione di ricerca tra i servizi.

Ferrara Alberto ha creato le pagine di login e di amministrazione, incluse tutte le operazioni lato server annesse, ha poi curato i database in XML e i loro schemi XSD, nonché le pagine dei contatti e della storia.

Varotto Mattia ha curato la homepage e la pagina del Dicono di Noi, inclusi i suoi aspetti lato server. Ha creato le funzioni in javascript per la validazione dei form.

Quanto non riportato è frutto di lavoro congiunto dei quattro membri.