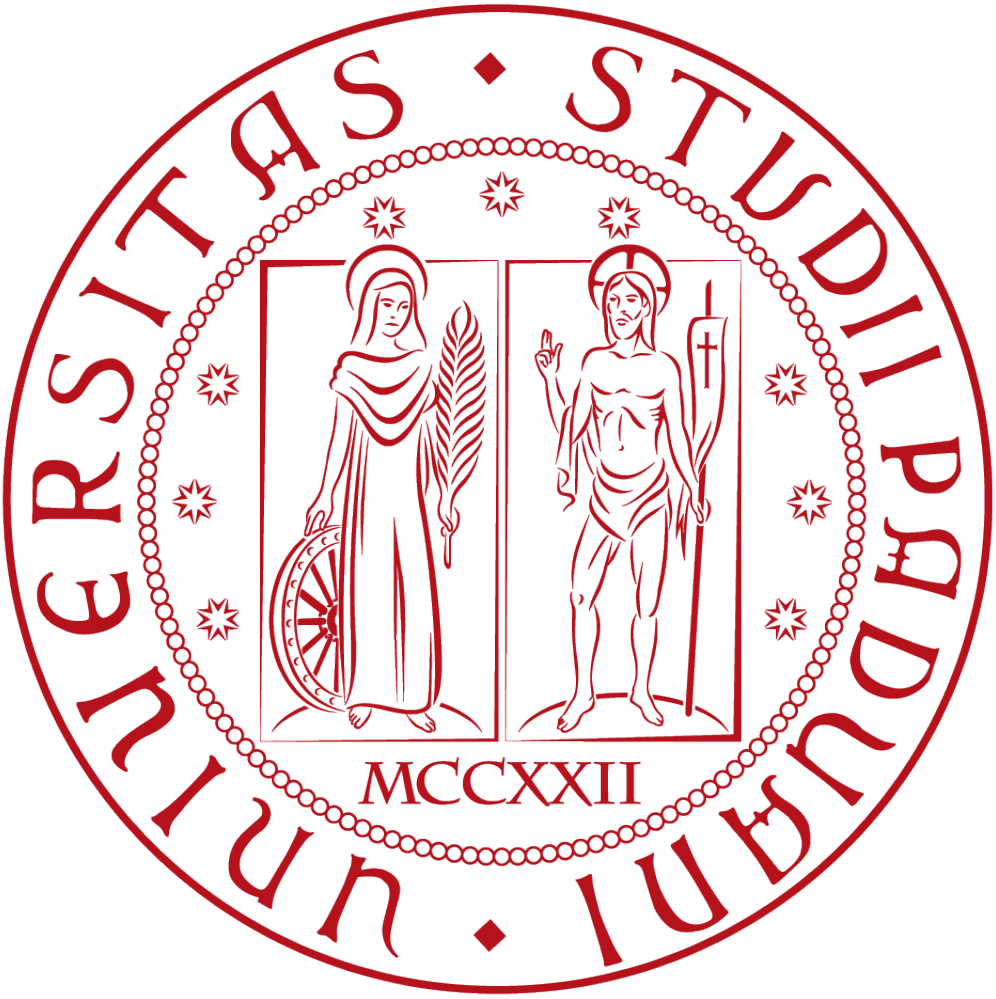

Università degli Studi di Padova

Relazione progetto corso di Tecnologie web: Cantina Benato

Informazioni sul documento

Nome documento	Relazione sito Cantina Benato
Scopo documento	Illustrare la progettazione e la realizzazione di tale sito web
Data documento	27/03/2012
Redattori	• Cattelan Giacomo • Cornaglia Alessandro • De Stefani Riccardo • Sotomayor Jorge
Anno Accademico	2011 - 2012
Indirizzo web del sito	`tecnologie-web.studenti.math.unipd.it/tecweb/~rdestefa`
Utenti precaricati	esame - password (utente senza ordinazioni) esame2 - password (utente con ordinazioni)
Email referente	giacomo.cattelan@gmail.com
Email gruppo	progetto-tecweb@googlegroups.com

Sommario

Il documento descrive le varie fasi che hanno portato alla realizzazione del sito web. A partire dalla sua analisi e la conseguente progettazione fino alla codifica.

Componenti del gruppo

Cognome e nome	Numero matricola
Cattelan Giacomo	610689
Cornaglia Alessandro	592764
De Stefani Riccardo	597316
Sotomayor Jorge	541740

Indice

1 Abstract **4**

2 Analisi dell'utenza **5**
2.1 Privati . . . 5
2.2 Aziende . . . 5

3 Layout **6**

4 Dati **7**
4.1 XML . . . 7
4.2 XMLSchema: . . . 7
4.3 Trasformazione(XSLT): . . . 7

5 Comportamento **8**
5.1 Perl . . . 8
5.1.1 Stampare codice xhtml . . . 8
5.1.2 Leggere e scrivere codice xml . . . 8
5.1.3 Trasformare codice xsl in codice xhtml . . . 8
5.1.4 Passare paramentri fra file .cgi . . . 8
5.2 Javascript . . . 9
5.3 Scelte implementative . . . 9
5.4 Sicurezza e controlli . . . 9
5.4.1 Controlli . . . 9
5.4.2 Sicurezza . . . 10

6 Accessibilità **11**

7 Test **12**

8 Meriti **13**

9 Ringraziamenti **13**

1 Abstract

Il progetto sviluppato si propone lo scopo di fornire un sito web per la cantina Benato, cantina vinicola realmente esistente e situata a Boccon di Vo', in provincia di Padova.

Tale sito web consente all'azienda di publicizzare sulla rete i propri prodotti, ed ai suoi clienti di effettuare prenotazioni online una volta che si siano registrati.

Nascendo da una necessità concreta l'intero progetto cerca di attenersi quanto più strettamente alla realtà, e di soddisfare le esigenze dei vari tipi di utilizzatori. Ci si aspetta che un utente che interagisce con il sito sia interessato ad ottenere informazioni sulla cantina e sui prodotti, oppure che voglia effettuare una prenotazione. Un ordine può essere effettuato solamente da utenti registrati ed autenticati, i quali hanno la possibilità di consultare lo storico personale delle ordinazioni.

In seguito all'analisi dell'utenza, riportata successivamente, si è scelto di concentrarsi maggiormente sulla versione stampabile del sito rispetto alla versione mobile.

Durante tutta la fase di sviluppo si è cercato di perseguire la massima accessibilità del sito si prestatandovi grande attenzione ed effettuando ricerche specifiche sul web.

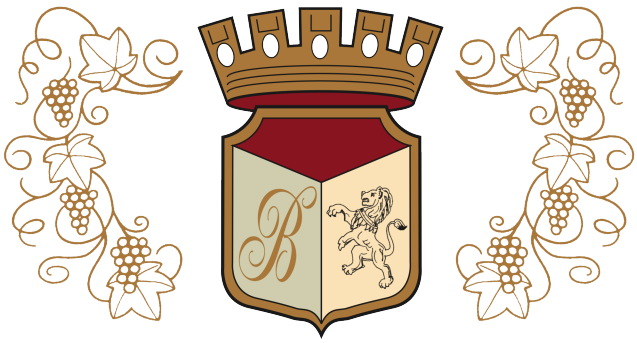

Figura 1: Logo ufficiale Cantina Benato

2 Analisi dell'utenza

Una volta che il gruppo ha scelto il tema su cui sviluppare l'intero progetto è stata realizzata un'attenta analisi dell'utenza del sito che si andava a sviluppare. Durante questa fase hanno impiegato alla pari tutti e quattro i membri del gruppo.

Ci si aspetta che un utente che interagisce con il sito web sia alla ricerca di informazioni su prodotti enologici della Cantina Benato o sulla cantina stessa oppure sia intenzionato a prenotare dei quantitativi di vino. I risultati dell'analisi indicano che i tipi di utenti che ci aspettiamo che interagiscano con il sito web siano i seguenti:

- Privati
- Aziende

2.1 Privati

I Privati possono essere, nella maggioranza dei casi, delle persone anziane o di media età che cercano prodotti enologici secondo le proprie esigenze, come consumarli o regalarli. Questo non toglie che anche giovani utenti possano approciarsi al sito ma si presume essi siano una netta minoranza. È possibile che gli utenti nominati siano affetti da una qualsiasi disabilità di conseguenza si è progettato la struttura del sito in modo che sia facile per tali utenti navigare ed effettuare delle operazioni sul sito.

2.2 Aziende

Un'altra classe di utenza sono le aziende come ristoranti, trattorie o qualunque azienda che voglia mettere a disposizione dei suoi clienti dei prodotti con garanzia di qualità. Essendo delle aziende si presupone che abbiano al loro interno del personale capace di navigare ed effettuare delle operazioni di prenotazione all'interno del sito. Quall'ora esso non si vero, comunque il sito è stato pensato per rendere l'interazione col'utente la più semplice possibile.

3 Layout

La struttura di layout del sito è stata realizzata applicando una semplice rivisitazione del classico template di layout a tre pannelli. La sezione di header, posta in alto nella pagina, è destinata a contenere il logo e l'intestazione dell'azienda per cui sviluppiamo il sito. La sezione di path, appena sotto alla sezione header, mostra il percorso preciso della pagina corrente.

La sezione di footer, posta nella parte bassa della pagina, contiene solo i nomi dei progettisti e i link alle validazioni html, CSS e accessibilità. Nella sezione centrale di contenuto della pagina un blocco div wrapper racchiude due parti, il blocco di navigazione che presenta la lista dei link necessari alla navigazione del nostro sito, e il blocco di contenuto, con il contenuto vero e proprio. Il blocco usato come wrapper (avente id "container") occupa l'85% della pagina in larghezza, lasciando ugual margine a sinistra e a destra. Il blocco dei link di navigazione (avente id "navigation") flotta a sinistra grazie alla proprietà float:left e mantiene una larghezza del 20%, mentre il blocco con i contenuti (avente id "content") mantiene un margine sinistro del 22%, onde evitare che il contenuto, una volta riempito lo spazio adiacente il blocco di navigazione, si disponga al di sotto di quest'ultimo.

La parte presentazionale del layout è stata implementata con CSS puro, utilizzando regole CSS 2.1 e CSS 3. Le fondamenta del layout sono state realizzate con regole CSS 2.1 in modo da assicurare compatibiità e robustezza, visto che layout definiti in CSS3 come ad esempio le griglie non sono ancora supportate. Successivamente sono state aggiunte regole CSS 3 con lo scopo di rendere l'intera interfaccia piu' gradevole e proporre un'esperienza utente migliore.

Tuttavia queste regole CSS 3 sono pensate e impiegate in modo tale che, nel caso in cui un browser non supporti queste CSS 3, il layout subisca un degrado elegante senza compromettere accessibilità e usabilità. Se un text-shadow non viene visto interpretato dal browser accadrà quindi che il testo risulterà ancora visibile e leggibile, ma senza ombreggiatura. Se una transition viene ignorata l'animazione prodotta del browser si mostrerà scattosa e non fluida, ma ancora non compromettendo l'esperienza di navigazione del nostro sito da parte dell'utente. Talvolta le regole CSS 3 sono impiegate in punti leggermente più critici: è il caso ad esempio dei background che fanno uso di gradienti di sfondo. In questi casi è stato previsto un fallback al fine di garantire il degrado elegante, aggiungendo una doppia della regola "background": la prima in ordine di scrittura imposta una sfondo solido usando CSS 2.1, la seconda sovrascrive la prima impostando lo sfondo con uso del gradiente. Così facendo i browser che non sanno interpretare un gradiente lasceranno applicata la prima regola di background a sfondo solido, mentre i browser aggiornati sapranno applicare il gradiente.

Altro esempio simile è il caso del colore del testo in RGBA piuttosto che nel solito spazio RGB. Anche in questo caso la regola color:RGBA(..) è preceduta dal corrispondente fallback che imposta il colore in RGB, secondo il ragionamento seguito nel caso dei gradienti. Il layout è stato impostato in maniera fluida, esprimendo le dimensione degli elementi e dei contenitori perlopiù in percentuali quando possibile, altrimenti in em. Solo alcune immagini come il logo dell'azienda sono impostate a pixel, poichè la si ritiene talmente importante da non rimpicciolirla.

Anche la dimensione del testo è stata fissata in em, adattandosi alle preferenze dell'utente. Grazie a questa struttura fluida il layout si rimpicciolisce e ingrandisce dinamicamente e coerentemente al resize della finestra del browser, ovviamente fino ad un determinato punto critico. In questa ottica però, non è stato ritenuto necessario definire un layout per dispositivi mobili, vista anche la classe d'utenza del sito; si è preferito invece privilegiare la realizzazione di un layout dedicato alla stampa, molto più utile visto il tema centrare del sito.

4 Dati

4.1 XML

Come richiesto dalle specifiche del progetto si è implementato una forma di database utilizzando un file XML il qualle contiene tre tipi principali di dati: Prodotti, Utenti e Prenotazioni. Lo scopo dell'XML è di memorizzazare i dati come gli utenti iscritti al sito, i prodotti disponibili e le prenotazioni che tali utenti hanno effettuato. La struttura del XML si può riassumere nel seguente diagramma:

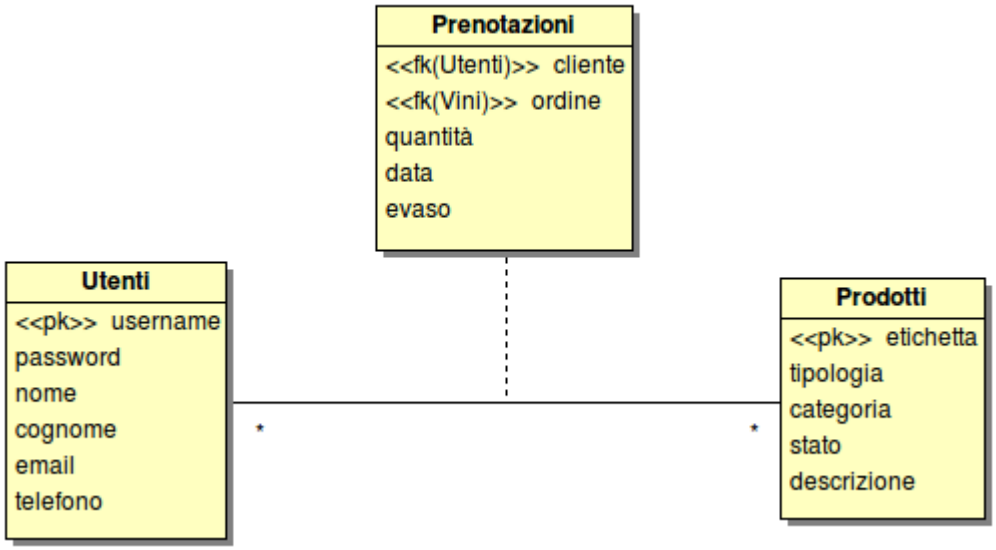

Figura 2: Diagramma relazionale

Dallo schema soprastante si può osservare che tra gli Utenti e i Prodotti c'è una relazione "molti a molti" che si traduce nell'entità Prenotazioni che è uno storico delle prenotazioni evase o no.

Essendo la gestione dell'integrità referenziale non ottimizzata nel XMLSchema si è deciso di demandare al server il controllo di coerenza tra i vincoli di chiave primaria e chiave esterna.

4.2 XMLSchema:

Dato i tipi di dati che si volevano rappresentare nel file XML c'era la possibilità di applicare molti tipi di modelli di schemi XML; di conseguenza si è optato per il modello di schema XML con la maggior estendibilità cioè il modello "Tende alla Veneziana".

4.3 Trasformazione(XSLT):

I fogli di stile XSLT sono stati applicati per stampare i Prodotti che si vuole presentare alle nostre utenze; questa decisione è stata presa poiché i Prodotti sono già stati precaricati nell'XML, non variano in base all'utente che li visualizza e non è stato previsto l'inserimento di altri vini. Per rendere i fogli di trasformazione compatibili con tutti i possibili browser la conversione da XSLT a HTML viene fatta lato server tramite le pagine catalogo.cgi e index.cgi . La presenza di menù diversificati tra utenti loggati e non ci ha costretto a creare due trasformate diverse che verranno applicate in base alla sessione che è attiva al momento (come utente loggato oppure no).

5 Comportamento

5.1 Perl

Il sito è stato sviluppato per la quasi sua interezza mediante pagine dinamiche, scritte in linguaggio Perl. Questa scelta è dovuta al fatto che una stessa pagina mostra contenuti diversi a seconda che l'utente sia registrato o meno. Inoltre il linguaggio Perl offre la possibilità di utilizzare e gestire le sessioni come metodo di scambio dati fra pagine.

Le sessioni, oltre al tipico utilizzo, sono state utili a mantenere e garantire l'integrità referenziale, durante la fase di prenotazione di un prodotto enologico, assicurando che l'utente che effettua l'ordine sia presente fra i registrati.

Le pagine realizzate consentono di:

- stampare codice xhtml
- leggere e scrivere codice xml
- trasformare codice xsl in codice xhtml
- passare parametri fra pagine .cgi

5.1.1 Stampare codice xhtml

Per soddisfare tale necessità sono state definite alcune funzioni all'interno del file *static.cgi*, situato in */cgi-bin/funzioni*, che si occupano di stampare le varie componenti che costituiscono le pagine del sito. Molte di queste ricevono in ingresso dei parametri, utili a personalizzare le pagine.

La scelta di di realizzare il codice xhtml tramite libreria CGI è dovuta al fatto che consente di mantenere un buon grado di disgiunzione fra il codice utile a realizzare la struttura di una pagina, e quello che gestisce il contenuto. Inoltre questa scelta di modellazione rende il codice facilmente riutilizzabile e manutenibile, semplificando così l'operazione di modifiche future.

5.1.2 Leggere e scrivere codice xml

Essendo i dati del sito memorizzati ed organizzati in file di tipo xml, si è reso necessario interagire con questi. Per semplificare alcune azioni in lettura è stato utile eliminare i namespace perchè così facendo si è reso possibile utilizzare la libreria XML::LibXML. Tale libreria mette a disposizione dei metodi intuitivi per interrogare i file xml.

Questa strategia non è una *best-practice* ma ha facilitato il lavoro del gruppo quindi, in seguito ad una scelta collettiva, è presente nella maggior parte dei file che compongono il sito.

5.1.3 Trasformare codice xsl in codice xhtml

Come descritto nella parte che tratta i dati, la trasformazione da codice xsl in xhtml è stata implementata nei file *index.cgi* e *catalogo.cgi* . Queste due pagine realizzano dei semplici parser che mediante la libreria XML::LibXSLT trasformano le singole righe xsl in righe di xhtml valido. La scelta è stata influenzata dal fatto che questo metodo rende più semplice e veloce la stampa di una grande mole di dati che sarebbe stata decisamente molto più onerosa se gestita solamente tramite perl.

Inoltre la trasformazione dei fogli di stile xslt, lato server così realizzati, rende il codice prodotto automaticamente accessibile a qualsiasi browser, nascondendo la struttura del foglio xml.

5.1.4 Passare paramentri fra file .cgi

Per rendere possibile lo scambio di dati fra le pagine *.cgi* sono stati utilizzati tre metodi offerti dal linguaggio Perl a seconda delle situazioni: le sessioni, ed i metodi post e get.

Il metodo get si è reso utile per la visualizzazione dettagliata del singolo prodotto, inserendo nella barra del URL, nel modo corretto, l'etichetta del vino di cui si volevano conoscere le informazioni. Si

è deciso di utilizzare tale metodo per la visualizzazione specifica del vino per far si che un utente possa facilmente ritornare a visitare una di queste pagine e perchè i dati che compaiono in chiaro non sono sensibili.

Il metodo post invece è stato utilizzato in ogni pagina in cui un utente ha la possibilità di interagire con il sito inserendo dati o inviado richieste al server. La tecnica scelta è stata utile per assicurare l'usabilità nelle pagine in cui un utente inserisce dei dati da inviare al server, perchè così facendo è possibile segnalare specificatamente gli errori di immissione e mantenere i dati corretti. È possibile osservare il comportamento appena descritto nei file *login.cgi* e *registrati.cgi*.

Le sessioni sono state usate per mantenere e controllare lo stato di un utente loggato, e su questo il gruppo ha dovuto superare il problema determinato dal fatto che la release installata nel server del laboratorio si è dimostrata meno recente di quella che si è usata nelle prove locali. Per ovviare a questo problema si sono dovute applicare le sessioni come richiesto dalla release *3.95*; l'uso delle sessioni in questa maniera richiede che venga passato l'id di una sessione da una pagina all'altra. Questo modo di agire è applicabile in tre diversi modi, tramite: get, post o cookie. Il metodo risultato più elegante e sicuro è risultato essere il metodo dei cookie, tralasciando il rischio che un utente possa disabilitare localmente l'uso dei coockie, perchè stando allo studio di analisi dell'utenza si presuppone che la maggior parte dei nostri utenti non abbia conoscenze sufficienti per far ciò.

5.2 Javascript

Si è scelto di utilizzare javascript per filtrare i prodotti mostrati al utente in base a dei criteri come per etichetta, per tipologia (bianco, rosso o rosato), per categoria(dolce o secco) e per stato(frizzante, fermo o spumante). Questo filtraggio viene fatto lato client in modo che non ci siano delle chiamate al server ogni volta che l'utente cambia criterio di ricerca.

La ricerca prende i diversi parametri e mostra solo i prodotti che soddisfano le proprietà scelte. Se il campo etichetta è vuoto, la ricerca va fatta solo su tipologia, categoria e stato tralasciando l'etichetta di tale prodotto.

Si è deciso di utilizzare javascript anche per rilevare se il browser che l'utente sta utilizzando supporta certi metodi. Quall'ora il browser non li supporti vengono nascoste certe funzionalità(es. non è possibile effetturare la ricerca dei prodotti se il browser scelto è Internet Explorer nelle versioni inferiori alla 9).

Comunque si sono utilizzati dei metodi standard e si è seguita una struttura consigliata dalla W3C riguardo a metodi e modalità di accesso all DOM HTML del documento.

5.3 Scelte implementative

Per garantire l'usabilità e una buona esperienza di navigazione del nostro sito, si sono rese necessarie alcune scelte.

Si è scelto di reindirizzare un utente alla propria pagina personale una volta che effettua correttamente un ordine, dove è visibile la buona riuscita dell'operazione e l'insieme delle varie ordinazioni non ancora evase. La scelta è dovuta al fatto che la tecnologia che ci consente di stampare la pagina *catalogo.cgi* non ci consente di stampare un messaggio che avvisi l'utente che la sua ordinazione è stata ricevuta correttamente.

Anche quando un utente effettua il login questo viene reindirizzato alla propria pagina personale, scelta che è sembrata la migliore all'interno del gruppo.

Quando invece un utente autenticato effettua l'operazione di logout, questo viene rimandato alla pagina *index.cgi* .

5.4 Sicurezza e controlli

5.4.1 Controlli

Il sito per garantire il buon funzionamento, effettua i dovuti controlli sui dati che un utente può immettere, lato server. Le pagine in cui un utente ha la possibilità di inserire dati che incidono sullo

stato del sito sono: *prenota.cgi* e *registrati.cgi* .

Nel file *prenota.cgi* viene opportunamente controllato, mediante espressione regolare, che la quantità di bottiglie di vino che un utente si accinge a prenotare si maggiore di zero. L'integrità referenziale fra l'utente, identificato da uno username unico, e la prenotazione di una determinata quantità di prodotti enologici viene garantita dall'uso delle sessioni.

All'interno del file *registrati.cgi*, invece, i controlli che vengono effettuati sono decisamente superiori. Per ogni campo dati richiesto per la registrazione viene controllato che non siano presenti parentesi angolari, così da evitare spiacevoli inconvenienti, e che i campi obbligatori non vengano lasciati vuoti. È previsto il caso che un utente abbia uno o più nomi o cognomi, purchè questi siano separati da spazi, e che siano costituiti da caratteri validi. Viene controllata la forma della email che viene inserita e che il numero di telefono fornito sia composto da soli numeri.

Prima che un utente venga effettivamente registrato, vie controllato che la password inserita corrisponda a quella di conferma e che lo username non sia già stato scelto da un altro utente.

Per ogni errore riscontrato viene visualizzato un messaggio esplicativo affianco al campo di input dato; invece i campi corretti, fatta eccezzione per la password e conferma, vengono riproposti così che l'utente non debba reinserirli.

5.4.2 Sicurezza

Si è scelto di memorizzare le password in maniera criptata utilizzando la funzione *crypt* messa a disposizione dal modulo CGI. La funzione implementa un algoritmo di sicurezza a chiave condivisa SHA1, e come chiave condivisa è stata scelta la stringa *"ciao"*.

È assicurato che le funzionalità offerte per gli utenti registrati siano raggiungibili solo da utenti autenticati e viceversa.

6 Accessibilità

Per quanto riguarda l'accessibilità del sito, le pagine sono state sottoposte al test automatico Cynthia di contentquality.com, tuttavia un controllo esaustivo non può essere fatto da un validatore automatico, ma deve essere accompagnato dal buon senso umano dei progettisti col fine di eliminare falsi positivi o falsi negativi del test.

La costruzione stessa del codice HTML è stata realizzata tenendo conto delle regole generali di accessibilità:

- Uso di XHTML strict, validando le pagine finali al validatore del W3C
- la struttura, la presentazione e il comportamento sono completamente separati, grazie ad un codice HTML semantico, css in fogli di stile esterni, funzioni javascript in un file esterno
- inserimento nelle pagine dei link "Torna su" per ritornare in alto nella stessa pagina
- inserimento di un link nascosto visivamente ma presente, che permette ai lettori di schermo di far saltare il menu di navigazione
- mantenimento coerente della presentazione visiva dei link visitati e non
- link del menu di navigazione corredati di accesskey e tab index. Inoltre il link che punterebbe alla pagina in cui siamo viene disabilitato onde evitare link circolari
- uso del path per mostrare all'utente dove si trova in ogni momento, al fine di favorire l'orientamento
- realizzazione di tabelle accessibili, come nella pagina "chi siamo", corredando i table header di "id", "abbr", e "scope". Ogni cella quindi ha il suo attributo "headers" con cui riferirsi alle intestazioni
- form divisi in vari fieldset per agevolare il compilamento dell'utente e facilitare la vista della form
- controllo dei colori con il software reperibile al sito colororacle.org, che mostra come vengono viste le pagine del nostro sito dagli utenti affetti da daltonismo, e con WCAG contrast analyzer di Mozilla Firefox
- test su degradi del sito effettuati con Web Developer Tool di Mozilla Firefox, simulando la disabilitazione di immagini, stili css e javascript
- test di navigazione su browser non visuali come Lynx
- test di lettura dello schermo con Fangs di Mozilla Firefox
- predisposizione di un layout css dedicato alla stampa

7 Test

Il sito è stato sviluppato perlopiù utilizzando Apple Safari 5.1, Mozilla Firefox 10, Google Chromium 17 e Google Chrome 16. Tuttavia sono stati svolti test di compatibilità e visualizzazione grazie a Microsoft Internet Explorer 9, il quale permette a sua volta di visualizzare le pagine come si stesse utilizzando Internet Explorer 8 o 7. Internet Explorer 6 è stato invece completamente ignorato, considerando la ormai nulla diffusione di questo browser.

Il risultato con la versione 9 di Internet Explorer è molto soddisfacente, mostrando il sito pressoché come in Chrome o Firefox aggiornati, mentre nelle versioni precedenti del browser di Microsoft si nota un degrado grafico, che però non intacca usabilità e accessibilità. Anche Opera 11 è stato usato per effettuare test, con risultati ottimi.

8 Meriti

Schema riassuntivo delle parti svolte e sviluppate da ogni componente del gruppo.

Cattelan Giacomo:

- struttura e validazione XML
- trasformazione XSL
- modellazione pagine Perl

Cornaglia Alessandro:

- modellazione pagine, controlli Perl
- inserimento dati XML

De Stefani Riccardo:

- struttura e validazione XHTML
- struttura e validazione CSS
- curatore grafico

Sotomayor Jorge:

- funzionalità client-side Javascript
- verificatore codice e relazione

Il gruppo tiene a precisare che le scelte intraprese durante la fase di progettazione e sviluppo sono sempre state prese in collettività e che ogni membro ha piena conoscenza anche delle parti del progetto che non ha svolto direttamente. Inoltre si informa che la relazione è stata stesa congiuntamente.

9 Ringraziamenti

Si ringrazia la Cantina Benato per l'idea, le informazioni e il materiale forniti.