

# Progetto Teconologie Web - A.A. 2013/2014

Luca De Franceschi, Michele Dal Santo, Nicolò Tresoldi, Davide Quaglio

25 Marzo 2014

- *Email referente: nicolo.tresoldi@hotmail.it*
- *URL sito:*
- *Credenziali di accesso:*
  - *User: admin@levecchiecredenze.it*
  - *Password: admin*

# 1 Informazioni generali

Il seguente progetto segue le linee guide indicate dal corso di studi **Tecnologie Web**. Tali specifiche sono reperibili alla seguente url:

<http://docenti.math.unipd.it/gaggi/tecweb/progetto.html>

Fin dalla prima linea di codice è stata seguita una stringente separazione tra **contenuto, presentazione e comportamento**. Il contenuto è strettamente fornito dall'**XHTML**, così come la presentazione è delegata totalmente al **CSS** e il comportamento agli script **Javascript** e **Perl/CGI**. Questo ha portato enormi vantaggi in termini di manutenibilità lungo tutto l'arco del progetto.

Il sito web in questione è stato progettato usando lo standard **XHTML 1.0 Strict**. Per scelta del gruppo non è stata implementata nessuna pagina con **HTML5**, in quanto seppur in parte supportato non è ancora standard. La scelta di usare tecnologie che rispondano a uno standard ufficiale rende il sito più accessibile, e l'accessibilità, come verrà ampiamente ripetuto lungo la seguente relazione, è stato il punto cardine su cui ha ruotato l'intero progetto. Questo non significa che il progetto non guarda al futuro e si "tira indietro" di fronte alle tecnologie emergenti. Lavorare con la coscienza di utilizzare strumenti riconosciuti come standard porta diversi vantaggi in termini di manutenibilità e di accessibilità.

Il database di riferimento, come da specifica, è stato implementato con **XML** e ad ognuno dei file è stata associata uno schema definito tramite **XMLSchema**.

Ogni pagina che necessita di contenuti dinamici è stata realizzata tramite script lato server **Perl/CGI**. Lo *scaffolding*, allo stato attuale del progetto, si presenta con la struttura seguente:

- **public-html**, in cui risiedono tutte le pagine statiche, i file **CSS**, gli script **javascript** e le immagini;
- **cgi-bin**, in cui risiedono tutti gli script **Perl/CGI** e i file di template da popolare;
- **database**, in cui risiedono tutti i file **XML** e i loro schemi associati;
- **relazione**, in cui risiedono i file **L<sup>A</sup>T<sub>E</sub>X** della relazione;

Lungo tutta la relazione verranno discusse alcune *features* che non sono state implementate. Riteniamo che il progetto abbia soddisfatto pienamente gli obiettivi richiesti dal corso. Ciononostante i tempi ristretti e gli altri corsi/esami sovrapposti hanno impedito di estendere ulteriori funzionalità e apportare altri miglioramenti. Il progetto è stato realizzato con un *occhio al futuro*, focalizzando l'attenzione sulla manutenibilità; pertanto se in futuro esso venisse raffinato e migliorato, le modifiche richiederebbero una manutenzione non troppo onerosa.

## 2 Abstract

I ristoranti al giorno d'oggi, soprattutto in Italia, nonostante la crisi economica martellante restano ancora una meta molto gradita e quasi irrinunciabile per la maggior parte delle persone. La ristorazione è un settore molto interessante per chiunque ami la buona cucina e la buona compagnia. Ma qualsiasi ristoratore sa benissimo che non basta una cucina raffinata per avere successo. Gran parte dell'appetibilità di un esercizio di questo tipo dipende moltissimo da **come esso appare** e da **come la gente ne parla**. In mezzo a questi due pilastri si colloca perfettamente il web. Gli utenti navigano ininterrottamente nella rete e oggi essa è diventata la fonte primaria di informazioni. Risulta dunque fondamentale per qualsiasi ristoratore dare una buona immagine di sé e del proprio locale nel web. Viviamo nell'era dei *social network* quindi siamo perfettamente consapevoli del fatto che gran parte della pubblicità che un esercizio può ottenere lo attinge da essi. Inutile citare *facebook* o *twitter*, ormai padroni assoluti. Più interessante in questo settore sarebbe parlare di *tripadvisor* ma lo scopo di questo progetto non è certo quello di analizzare questi aspetti. La domanda che viene spontanea, date queste considerazioni, è come un sito web si inserisce in questo insieme e quale sia la sua reale utilità.

Bisogna essere realisti: l'utilità di un sito web per un ristorante è **molto bassa**, o almeno questa è la situazione al giorno d'oggi. Se fino a qualche anno fa per un ristoratore l'avere o il non avere un sito web online poteva fare un'enorme differenza nel 2014 questa differenza si è molto assottigliata. Pensiamo per esempio ai contenuti significativi che il sito web da noi sviluppato offre:

- Possibilità di visualizzare una **galleria** di immagini;
- Possibilità di visualizzare **news**;
- Possibilità di visualizzare il **menu**;
- Possibilità di visualizzare gli **eventi**.

Chiunque abbia un minimo di familiarità con *facebook* sa benissimo che tutto ciò può essere facilmente gestito tramite esso. Anzi, considerando che un utente medio appena apre il proprio browser una delle prime pagine che visita è proprio *facebook* risulta chiaro come l'inglobare il tutto su di esso porti a una visibilità notevolmente maggiore. Nessun utente aprirebbe mai *lavecchiecredenze.it* di routine, semmai lo farà una prima volta e in seguito solamente quando lo riterrà necessario.

Detto questo sicuramente un pensiero che potrebbe scorrere nella testa di chiunque è questa:

*Perchè mai dovrei sviluppare un sito web di un ristorante?*

La risposta, viste le precedenti considerazioni, non sembra così immediata. Naturalmente risulta chiaro come nell'interesse primario di un ristoratore ci sia l'afflusso di persone nel suo ristorante. Anzi, probabilmente questa è l'unica finalità. Se la domanda ci venisse posta in questo momento queste sarebbero le risposte che daremmo:

- Anzitutto si tratta di un progetto didattico e la realizzazione delle sue componenti rispecchia in modo coerente gli obiettivi e gli argomenti presentati dal corso;
- In secondo luogo riteniamo che i social network in questo contesto abbiano un grosso difetto: forniscono un'**interfaccia comune** e non personalizzabile. Vedere la pagina *facebook* di un ristorante piuttosto che di un altro non farebbe grossa differenza se non per il contenuto. Riteniamo dunque che la componente **presentazionale** sia fondamentale per il sito web di un ristorante;
- Riteniamo che, se un ristorante possiede un sito web, allora i social network possono fungere solamente da **passerella** tra l'utente ed esso. Se il sito web risulta gradevole alla vista, accattivante e presenta i contenuti in maniera ordinata allora esso può risultare molto più efficace rispetto ad un social network ed invoglierebbe maggiormente l'utente a frequentare il locale;
- Per ultimo, ma non meno importante, riteniamo che l'**accessibilità** sia un concetto fondamentale nel web, e molto spesso i social network non rispettano questo canone. Il sito web è stato realizzato prestando attenzione massima all'accessibilità, favorendo tutte le categorie di utenti possibili nel migliore dei modi e il tutto senza compromettere il suo stile elegante. Riteniamo che le categorie di utenti svantaggiati possano trarre un'esperienza migliore visitando il sito piuttosto che visualizzando la sua pagina in un social network.

*"Le vecchie credenze"* risiede appieno in questa realtà. È in primo luogo un ristorante *"di lusso"*, con prezzi molto alti e le cui aspettative sono di conseguenza elevate. Proprio per creare grosse aspettative si è scelto di sviluppare questo sito, in modo da presentare il ristorante esattamente com'è: elegante, lussuoso e dalla cucina trascendentale. Con questo non vogliamo certo fare della pubblicità nel presente documento, si tratta solamente di giustificare al meglio le scelte e lo stile adottato. Se avessimo dovuto sviluppare un sito per una pizzeria avremmo certamente utilizzato uno stile differente. Ma, dal momento che si tratta di un ristorante elitario, abbiamo concentrato molto l'attenzione nel presentare al meglio ciò che esso offre, in modo che chiunque lo frequenti sappia cosa esso offre e si crei un'aspettativa alta.

Questo è in sostanza *"levecchiecredenze.it"*, un sito di un ristorante che si pone come obiettivo quello di presentare i contenuti in maniera accessibile e strutturata e allo stesso tempo fornire un'interfaccia elegante e pomposa, allo scopo di invogliare chiunque effettui una visita a pranzare o cenare presso di esso.

### 3 Analisi dell'utenza

Una delle prime operazioni che sono state svolte nella realizzazione del progetto è stata un'attenta analisi dell'**utenza**. Da ciò ne sono derivate diverse considerazioni, che hanno fatto sì che il progetto si sviluppasse in una direzione rispetto ad un'altra. Naturalmente quando si parla di utenza il *focus* viene messo sull'interfaccia del sito, che comprende, oltre al modo in cui esso appare all'utente, anche al modo in l'utente interagisce con esso.

Essendo nel mondo della ristorazione possiamo assumere che la quasi totalità degli utenti che effettuano una visita e una navigazione all'interno del sito sono dei **possibili clienti**, ovvero utenti che hanno come scopo quello di prendere in considerazione l'idea di cenare o pranzare nel ristorante. Ecco perché, come già evidenziato nel capitolo precedente, si è scelto di rendere il sito come una **vetrina**.

L'utente deve poter entrare nel sito e vivere una esperienza il più simile possibile a quella che avrebbe nell'entrare nel ristorante. Naturalmente stiamo parlando di due esperienze diverse, la prima delle quali non comprende sensazioni come l'udito e l'olfatto, anch'essi determinanti, ma dal punto di vista visivo la sensazione dovrebbe essere analoga, almeno per chi ha la fortuna di percepire questo senso. Ecco perché lungo tutta la navigazione è stato scelto di far scorrere una *slideshow* di immagini raffiguranti delle sezioni topiche, ovvero:

- Un'immagine dell'esterno del ristorante;
- Un'immagine dell'interno del ristorante;
- Un'immagine della cantina;
- Un'immagine di un piatto;

Allo stato attuale del progetto le immagini fornite risultano poco definite e quindi vengono visualizzate sgranate. L'effetto è chiaramente poco gradevole e per quanto riguarda l'accessibilità è chiaramente un danno. In futuro, non appena si otterranno immagini migliori il problema verrà indubbiamente risolto. Allo stato attuale è stato scelto di mantenerle per raggiungere l'obiettivo spiegato.

Per quanto riguarda gli utenti svantaggiati sono stati effettuati gli opportuni test di accessibilità ed è stato verificato che il sito sia accessibile a utenti ipovedenti.

La parte di amministrazione del sito ha un'utenza limitata a pochi elementi, che potranno presumibilmente essere il gestore del ristorante e/o altri componenti del personale. Per questa sezione è stato mantenuto lo stesso stile, ma per ovvi motivi non si è puntato fortemente il dito sull'estetica così come è stato fatto per la restante parte.

## 4 Suddivisione ruoli

All'interno del gruppo si è scelto di suddividere i ruoli per componenti, in modo da parallelizzare al massimo il lavoro. Tutti i membri del gruppo hanno conoscenze e hanno fatto un minimo di esperienza su tutte le tecnologie utilizzate. Ad ogni modo è risultato necessario dividere il più possibile per sezioni il lavoro, in modo che ciascuno fosse quanto più indipendente nello svolgere i propri compiti e le proprie modifiche fossero minimamente intrusive. La suddivisione è la seguente:

- **Luca De Franceschi** si è occupato dei seguenti compiti:
  - Configurazione dell'ambiente di lavoro;
  - Configurazione e mantenimento della struttura del progetto;
  - Codifica e manutenzione degli script Perl/CGI;
  - Creazione di pagine di template per gli script Perl/CGI.
- **Michele Dal Santo** si è occupato dei seguenti compiti:
  - Codifica e manutenzione del codice CSS;
  - Codifica e manutenzione del codice XHTML delle pagine statiche della vetrina.
- **Nicolò Tresoldi** si è occupato dei seguenti compiti:
  - Definizione della struttura dei file XML del database;
  - Definizione degli schemi associati ai file XML;
  - Esecuzione dei test sull'accessibilità della parte pubblica;
  - Esecuzione dei test sulla validazione del codice XHTML;
  - Esecuzione dei test sulla validazione del codice CSS;
  - Esecuzione dei test sulla validazione del codice XML associato allo schema.
- **Davide Quaglio** si è occupato dei seguenti compiti:
  - Creazione delle pagine XHTML di amministrazione del sito;
  - Esecuzione dei test sull'accessibilità della parte di amministrazione;
  - Codifica degli script javascript.

Tutti i membri del gruppo hanno lavorato alla relazione e steso i capitoli assegnati. In generale tutti hanno inoltre contribuito alla correzione e all'individuazione dei *bugs* e al suggerimento di nuove *features* tramite le issues di github.

## 5 Progettazione

### 5.1 XML e XMLSchema

Si è scelto di adottare XMLSchema al posto di DTD come linguaggio per descrivere la struttura dei file XML perchè più espressivo, e a fini didattici, essendo più complesso è stato più utile da imparare e data la sua complessità dimostra come il gruppo sa lavorare con XML. Di seguito verranno spiegate alcune scelte riguardanti la struttura.

**Scarso utilizzo degli attributi** I motivi per cui gli attributi sono poco presenti sono due:

- Un elemento è maggiormente estendibile rispetto ad un attributo;
- Un attributo viene utilizzato per descrivere metadati, mentre i dati in sé sono inseriti negli elementi;

**Formato della data** In molti file è richiesta una data. Si è scelto di salvare tale data in modo esteso, suddividendo l'elemento data in sottoelementi, così facendo è possibile indicare dei range di valore accettabili per giorno, mese ed anno. Mese viene scritto in lettere e non come numero, si è quindi applicata una *restriction*, partendo da string si accettano solo le stringhe equivalenti ai mesi. Non è stato utilizzato il tipo *Data* offerto da XMLSchema per evitare di dover utilizzare il formato americano. Per il tempo è invece stato utilizzato *Data*, precisamente nel file log.xml.

**Elementi opzionali** Nel file riguardante il menù sono presenti elementi non obbligatori. Nel caso di elementi semplici si è scelto di utilizzare l'attributo *nilable="true"* per indicare che tale elemento è opzionale. Nel file XML dunque l'elemento dovrà sempre essere presente, e nel caso in cui non lo si voglia definire vi si assegnerà *xsi:nil="true"*. Nel caso di elementi complessi si è scelto di utilizzare l'attributo *minOccurs="0"* che permette di non inserire l'elemento nel file XML. Tale scelta è stata fatta per rendere più leggibili i file XML, che risulterebbero altrimenti appesantiti.

Questi approcci sono necessari in quanto lasciare semplicemente vuoto l'elemento comporterà un errore di validazione, tranne che per gli elementi di tipo stringa.

**Modello adottato** Il modello adottato è *Bambole russe* in quanto non si è ritenuto il riutilizzo del codice di primaria importanza. Non si è aderito a tale modello solamente nel file menu.xsd. Questo perchè sono presenti quattro elementi (antipasti, primi, secondi, dessert) tutti con la stessa struttura interna, si è quindi dato un nome al tipo e lo si è messo esterno.

### 5.2 Social

In ogni pagina, con particolare risalto nella pagina contattaci, è stato reso disponibile un link a Facebook Twitter, Google plus e TripAdvisor. Le motivazioni sono spiegate nella sezione Abstract. I link non portano a nessuna

pagina, in quanto al momento tali pagine non sono presenti, ma verranno create non appena il sito verrà accettato e reso pubblico.

### 5.3 Sezione d' amministrazione

In questa sezione sarà possibile accedere solamente dopo aver effettuato la login nell' apposita pagina e qualora si tenti di accedere a questa sezione senza essere autenticati si verrà reindirizzati alla home page del sito. Sarà anche presente in ogni pagina un menu laterale per navigare tra le varie pagine di amministrazione e per tornare alla sezione pubblica.

**Dashboard** Questa sarà la pagina a cui si verrà reindirizzati una volta effettuata la log in, qui ci verrà fornito un messaggio che ci avverterà con che account siamo loggati e sotto di esso sarà presente una *tabella di log* che avviserà l' amministratore delle ultime azioni effettuate dagli amministratori del sito quali ad esempio l' inserimento di un nuovo evento, l' eliminazione o l' aggiunta di foto segnalando la data e l' ora di tali modifiche. Tali operazioni vengono effettuate leggendo da un file xml, ma per motivi di tempo il file xml non viene aggiornato.

**Modifica photogallery** In questa sezione verrà data la possibilità all' amministratore di aggiungere nuove immagini alla galleria e modificare i dati di un' immagine quali nome, testo alternativo e titolo, di un'immagine preesistente. Sono stati implementati dei controlli per non permettere il submit del form qualora ci sia uno o più campi lasciati vuoti.

**Inserimento nuovo evento** Qui l' amministratore potrà creare un nuovo evento fornendone il titolo, la descrizione, il costo del menu, la data a cui si riferisce l'evento e le portate che lo compongono. Tutti questi elementi sono obbligatori, e come tali qualora si cerchi di eseguire l' input di tale form, i campi vuoti verranno evidenziati all' amministratore con un' aura rossa intorno ad ogni input box vuota e quando si andrà ad inserire un carattere valido, tale aura verrà rimossa.

**Inserimento nuova news** Qui i campi saranno tutti necessari, quindi l' amministratore dovrà inserire tutti i campi, e qualora cerchi di eseguire il submit in presenza di campi vuoti non verrà permessa tale azione e verranno evidenziati i campi come nelle altre pagine della sezione privata.

## 6 Accessibilità

**Linee guida** I testi alternativi alle immagini sono stati impostati secondo le seguenti regole:

- Il testo deve presentare il contenuto e la funzione dell'immagine;
- Il testo deve essere breve, non più di due sentenze;
- Evitare di ripetere quanto già riportato nel testo vicino l'immagine;
- Non iniziare con "immagine di.." o "fotografia di.." .

### 6.1 Accessibilità sezione pubblica

**Cecità ai colori** Per testare che il sito sia accessibile alle persone daltoniche, tetracromiche o con altri problemi riguardanti i colori si è utilizzato l'estensione Vischeck, scaricabile dal sito Vischeck.com. Tale estensione, eseguibile con ImageJ, data un'immagine o una pagina web, mostra come viene percepita da chi soffre di cecità ai colori. È bastato verificare un paio di pagine, in quanto i colori e le immagini di background utilizzate sono le medesime in tutte le pagine.

**Utilizzo della sola tastiera** Per essere accessibile un sito deve poter essere navigabile utilizzando solamente la tastiera. Ogni pagina è stata testata ed il risultato è che l'intero sito è navigabile utilizzando *Tab* ed *Enter*. Non sono stati utilizzati gli attributi *tabindex* per i link in quanto il flusso logico del sito è semplice e lineare.

**Cecità** Per testare il sito rispetto agli *screen reader* è stata utilizzata l'estensione *Fangs* del browser Firefox. Tale estensione simula come un generico screen reader interpreta le varie pagine. Un problema che è sorto è stata la mancanza dell'italiano tra le lingue supportate da Fangs. Nonostante ogni pagina indichi l'italiano come lingua di default, tramite l'attributo *lang*, Fangs imposta l'inglese come lingua di default, e ad esempio legge i numeri decimali come centinaio (Ecco nella pagina cantina 2,00 diventa two hundred).

Nonostante ciò Fangs ha permesso di verificare che:

- tutti i link vengano catturati, e che ogni link abbia un significato chiaro;
- cambi lingua, le abbreviazioni e le tabelle fossero gestiti correttamente;
- visualizzare gli headers della pagina, funzionalità offerta da alcuni screen reader e utile per capire a grandi linee il contenuto della pagina e la sua struttura. Un esempio dell'utilità di tale funzionalità è data dalla pagina news, in cui grazie agli header è possibile accedere immediatamente ai titoli di tutte le news presenti, per poi selezionare quella di proprio interesse.s
- Ogni immagine abbia il rispettivo testo alternativo.

**Accessibilità dei form** Possiamo vedere un esempio di accessibilità dei form nella pagina contatti.html.

Per chi utilizza uno screen reader, nel compilare un form è fondamentale sapere come compilare gli input. Per questo motivo ad ogni *input type text* è associata una label con attributo for, il cui valore è l'id dell'input a cui fa riferimento. Con questa tecnica lo screen reader sa certamente collegare nel modo corretto un'input alla sua label. Tale tecnica è stata utilizzata anche per i *select menus*.

Per ogni bottone è stato utilizzato il tipo submit in quanto button necessita che javascript sia bilitato per funzionare, inoltre in ogni bottone è stato valorizzato l'attributo value, in modo che lo screen reader dia un significato esplicito all'azione del bottone.

**Total Validator** Total validator è uno strumento, che tramite un'apposita estensione di Firefox permette di verificare l'accessibilità di una pagina web secondo alcuni parametri. Le impostazioni di total validator scelte sono le più stringenti, richiedendo un'accessibilità aderente allo standard WCAG 2.0 AAA. Di seguito sono riportati alcuni errori segnalati da total validator, ma non corretti per i motivi riportati.

#### Pagina storia.html

**Riga: Errore 63:** The 'alt' attribute is for short descriptions. Use 'longdesc' for long ones.

Non è stato utilizzato longdesc per due motivi:

- Non è supportato da tutti gli screen reader;
- Il valore di tale attributo solitamente è un link ad un'altra pagina, anche esterna al sito che contiene la descrizione dell'immagine, e non un testo descrittivo.

## 6.2 Accessibilità sezione amministrazione

A differenza della parte pubblica del sito per poter usufruire della parte privata è necessario aver abilitato Javascript sul browser, le ragioni di tale scelta si possono identificare nel fatto che a seguito dell' analisi dell' utenza effettuata si è ritenuto che l' amministratore non disattivi mai javascript. Nonostante sia un progetto didattico è stato dimostrato che il gruppo sa gestire il caso in cui l' utente disattivi javascript nella parte pubblica.

**Accessibilità dei form** Per quanto riguarda l' accessibilità dei vari form, si è deciso di adottare questi accorgimenti:

- in ogni label di ogni input text obbligatoria sarà presente il carattere \*, il quale viene letto dagli screen reader che notificano all' utente che il campo è un campo obbligatorio come si può notare dall' immagine "screen reader required input";
- nel caso in cui si cerchi di inserire un carattere diverso da un numero o da delete all' interno dei campi che richiedono un numero, verrà impedito e l' utente verrà avvisato che si possono inserire solo numeri;

- se l' amministratore cercherà di inviare un form, nel quale non si siano riempiti campi obbligatori, l' azione verrà bloccata, evidenziando ogni input text non riempita correttamente con un' aura rossa intorno ad essa rendendo evidenti i campi vuoti. L' aura è volutamente molto larga in quanto dopo aver testato con vischeck si è ritenuta efficace questa ampiezza, inoltre il colore è accessibile in presenza di cecità ai colori. Quando si andrà a inserire un valore valido in uno di questi campi evidenziati si andrà a rimuovere l' aura dall' input text.
- per gli utilizzatori di screen reader, la compilazione necessita di label con un campo for adeguato per sapere come compilare una input text, e ciò è stato implementato in tutti i form della parte privata, anche in quelle sezioni dove i campi verranno aggiunti dinamicamente con l' uso di Jquery si è prestata attenzione a creare id adeguati e diversi per ogni campo e assegnarlo in modo corretto alla rispettiva label;

**Accessibilità da tastiera** L' intera parte privata è stata sviluppata in modo da poter essere interamente navigabile solamente con l' uso della tastiera, muovendosi con tab e selezionando gli elementi con invio, per permettere tale feature agli script javascript si è permesso nelle input text che richiedono solamente numeri anche la pressione del tasto tab, infatti qualora fossimo arrivati premendo tab in una input box che richiedeva solo numeri non sarebbe stata permessa la pressione del tasto tab.



Figure 1: Esempio di pagina vista da chi soffre di deuteranopia

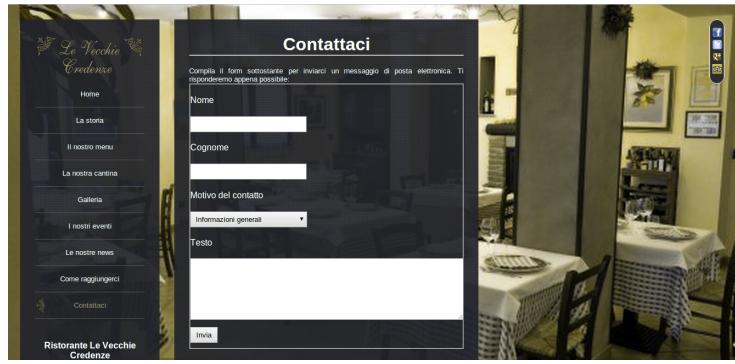


Figure 2: Esempio di pagina vista da chi soffre di protanopia



Figure 3: Esempio di pagina vista da chi soffre di tritanopia



Figure 4: Output di Fangs

Fangs output for: Contattaci

Reload Preferences

Screen reader output Headings list Links list

**Links list**

- Home
- La storia
- Il nostro menu
- La nostra cantina
- Galleria
- I nostri eventi
- Le nostre news
- Come raggiungerci
- Credits
- Admin

- Facebook
- Twitter
- Google Plus
- Tripadvisor

Figure 5: Elenco di link della pagina, come visualizzati da Fangs

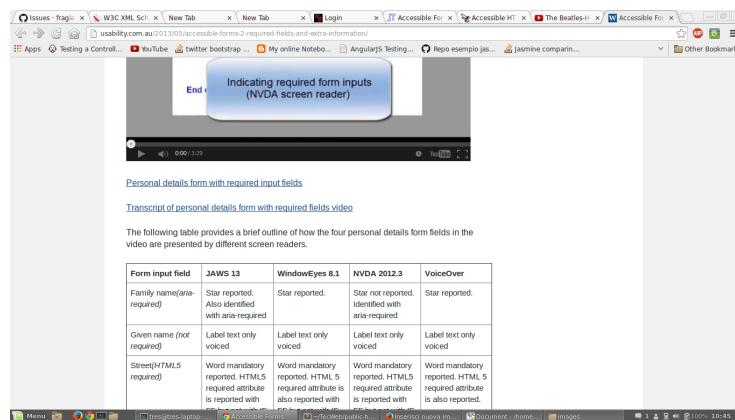


Figure 6: "screen reader required input" reazioni degli screeen reader a \*

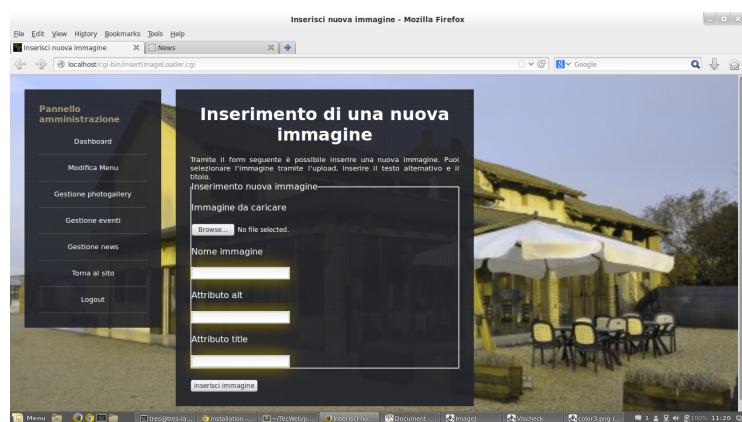


Figure 7: aura intorno ai campi lasciati vuoti vista in presenza di cecità ai colori