Relazione del progetto didattico per l'insegnamento di Tecnologie Web

Università di Padova

Informazioni sul gruppo:

Gruppo	fsociety
Membri del gruppo	Bicego Eduard
	Castello Davide
	Tombolato Andrea
Email referente	davide.castello.1@studenti.unipd.it

URL sito

http://tecnologie-web.studenti.math.unipd.it/tecweb/~dcastell/

Dati di accesso:

Username	admin
Password	admin

Indice

1 Informazioni generali **1**

2 Abstract **2**

3 Analisi dell'utenza **3**

4 Suddivisione dei ruoli **4**

5 Progettazione **5**
5.1 XHTML 5
5.2 CSS 6
5.3 Perl/CGI 6
5.4 Javascript 9
5.5 XML e XMLSchema 9
5.6 Mobile 10

6 Accessibilità **12**
6.1 Chiarezza e semplicità 12
6.2 Colore 12
6.3 Parole straniere 14
6.4 Form 14
6.5 Accessibilità da tastiera 15
6.6 Accessibilità tramite screen reader 15
6.7 Compatibilità con i browser 17

7 Istruzioni per l'utilizzo su altri server **18**

Elenco delle figure

1 Esempio di pagina vista da dispositivo mobile 3
2 Esempio di pagina vista da dispositivo mobile con menu a tendina aperto 11
3 Esempio di pagina vista da chi soffre di protanopia 12
4 Esempio di pagina vista da chi soffre di deuteranopia 13
5 Esempio di pagina vista da chi soffre di tritanopia 13
6 Esempio di pagina vista da chi soffre di monochromacy 14
7 Esempio di come vengono mostrati gli errori nelle form 15
8 Esempio di come viene mostrata la pagina index.html da Lynx . 16
9 Esempio di come viene mostrata la pagina menu.cgi da Lynx . . 17

1 Informazioni generali

Per lo sviluppo del progetto abbiamo seguito le linee guida indicate sul sito del corso di Tecnologie Web:

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

Fin da subito si è deciso che durante lo sviluppo dell'intero progetto si sarebbe operata una forte suddivisione tra contenuto, comportamento e presentazione usando le tecniche apprese a lezione, al fine di migliorare l'accessibilità del sito ed il posizionamento sui motori di ricerca.

Lo standard adottato per il codice `HTML` è `XHTML 1.0 Strict`, mentre lo standard adottato per il `CSS` è `CSS2`.
Il comportamento del sito è stato definito interamente in `Perl/CGI`, con l'aggiunta di alcuni script `Javascript`.

Per quanto riguarda il database, esso è stato implementato in `XML` e si può trovare nella cartella `data`. È suddiviso in `menu.xml` e `admins.xml`.
Il file `menu.xml` contiene il menù del ristorante: nella stessa cartella è fornito lo schema secondo cui il documento XML è valido: `menu.xsd`; lo schema è stato definito in `XMLSchema` seguendo il modello Tende alla Veneziana.
Il file `admins.xml` contiene l'username e password per accedere all'area amministratore, prevedendo la possibile aggiunta di ulteriori amministratori in futuro: nella stessa cartella è fornito lo schema secondo cui il documento XML è valido: `admins.xsd`; lo schema è stato definito anch'esso in `XMLSchema` seguendo il modello Tende alla Veneziana.

Lo *scaffolding*, allo stato attuale del progetto, si presenta con la struttura seguente:

- `cgi-bin`, in cui risiedono tutti gli script `Perl/CGI` e i file di template da popolare;
- `data`, in cui risiedono tutti i file `XML` e i loro schemi associati;
- `public-html`, in cui risiedono tutte le pagine statiche, i file `CSS`, gli script `Javascript` e le immagini;
- `relazione`, in cui risiede il presente documento, `relazione.pdf`.

2 Abstract

Il gruppo ha deciso di realizzare un sito web per il ristorante Sakura, un ristorante giapponese di Bassano del Grappa.
Il tutto è nato dalla necessità di sviluppare un sito web per il progetto del corso di Tecnologie Web e, in contemporanea, l'incontro con Ming, il padrone del suddetto ristorante.
Dopo avergli parlato delle nozioni di accessibilità e di posizionamento sui motori di ricerca apprese a lezione, Ming è rimasto davvero impressionato e ha chiesto al gruppo di sfruttare l'occasione del progetto didattico per realizzare il nuovo sito del ristorante.
Durante la fase iniziale, abbiamo deciso di dedicare una parte di tempo per analizzare il sito attuale del ristorante, che è reperibile al seguente link:

http://www.ristorantesakura.com

Abbiamo riscontrato i seguenti problemi nel sito attuale:

- non vi è una completa separazione tra contenuto e presentazione;
- non vi è una completa separazione tra contenuto e comportamento;
- il layout è tabellare;
- il codice `XHTML` è dichiarato non vado dal validatore di W3C;
- tutto il menù del sito è riportato sotto forma di più immagini una di seguito all'altra, rendendolo completamente inaccessibile ad uno screen reader;
- non viene indicata attraverso l'attributo `xml:lang` la presenza di parole straniere, che sono moltissime all'interno del sito.

La domanda che ci siamo posti durante questa fase iniziale di analisi è: quali vantaggi può portare il nostro sito web al ristorante?
Sicuramente permetterà di raggiungere un maggior numeri di utenti: tenendo conto di questo, riteniamo fondamentale che il sito in questione sia accessibile per permettere a chiunque di visitarlo.
I social network sono molto utili in questo caso, ma non permettono un'interfaccia personalizzata, impedendo di mostrare la propria personalità attraverso il design.
Un'altra considerazione che abbiamo fatto è stata che se l'esperienza dell'utente sul sito è piacevole e i contenuti sono ben ordinati e facilmente reperibil, l'utente è invogliato ad usare il sito per le operazioni più frequenti: ad esempio, guardare il menù e i prezzi prima di ordinare, o guardare dal sito il numero di telefono e far partire da lì la chiamata.

3 Analisi dell'utenza

Quando si parla di utenza l'attenzione si sposta sull'interfaccia: come il sito appare all'utente ma anche come l'utente interagisce con esso.
Essendo il sito web di un ristorante si può supporre che una qualsiasi persona sia interessata ad informarsi sul ristorante o ordinare take away, per questo motivo è fondamentale l'accessibilità del sito.

Una grossa fetta di utenti è composta dai ragazzi più giovani, molti di cui sono assolutamente innamorati della cucina giapponese e tendono a frequentare il ristorante oppure ordinare take away. Per questo motivo il gruppo ha deciso di definire lo stile del sito anche per i dispositivi mobile, tenendo conto che la maggior parte degli accessi al sito da parte di ragazzi giovani verrà effettuata da smartphone o tablet.

Figura 1: Esempio di pagina vista da dispositivo mobile

Questi dati sono stati ricavati dall'esperienza personale di alcuni membri del gruppo ma anche da un confronto con lo stesso proprietario del ristorante Sakura: egli infatti ci ha dato molte informazioni interessanti riguardo il mondo della cucina giapponese che abbiamo deciso di sfruttare a nostro vantaggio per ottenere un sito *giusto* per il suo bacino d'utenza.

La parte di amministrazione del sito ha un utenza limitata al gestore del ristorante, quindi l'utente che utilizzerà l'area amministratore sarà un utente che ha conoscenze di `XHTML`.
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 dei ruoli

Inizialmente tutti i membri del gruppo hanno collaborato per effettuare un'analisi dettagliata del sito attuale del ristorante, della possibile utenza del sito, dei bisogni degli utente e dei bisogni del proprietario del ristorante.
Una volta ottenuti questi dati, è iniziata una fase di progettazione del sito dove tutti i membri hanno lavorato assieme per cercare di definire più in dettaglio possibile le varie pagine necessarie, sia statiche che dinamiche, e le funzionalità da fornire, sia lato desktop che lato mobile.
Successivamente, il gruppo si è diviso gli incarichi in modo da poter parallelizzare il lavoro.

Andrea Tombolato si è occupato dei seguenti compiti:

- configurazione dell'ambiente di lavoro;
- codifica e manutenzione del codice `CSS`;
- codifica e manutenzione del codice `XHTML`;
- esecuzione dei test sulla validazione del codice `CSS`.
- esecuzione dei test sulla compatibilità con più browser;

Davide Castello si è occupato dei seguenti compiti:

- definizione della struttura dei file `XML` del database;
- definizione degli schemi associati ai file `XML`;
- esecuzione dei test sulla validazione del codice `XML` associato allo schema e correzione errori;
- esecuzione dei test sulla validazione del codice `XHTML` e correzione errori;
- esecuzione dei test sull'accessibilità della parte pubblica e dell'area amministratore;
- stesura della relazione, raccogliendo le scelte progettuali importanti dagli altri membri del gruppo e tenendo conto della fase di analisi svolta ad inizio progetto.

Eduard Bicego si è occupato dei seguenti compiti:

- codifica e manutenzione degli script `Perl/CGI`;
- creazione di pagine di template per gli script `Perl/CGI`;
- codifica degli script `Javascript`.

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

In seguito sono riportate in forma di elenco tutte le scelte progettuali effettuate durante lo sviluppo del progetto che il gruppo ha ritenuto degne di nota, suddivise per ambito di interesse: `XHTML`, `CSS`, `Perl/CGI`, `Javascript`, `XML` e `XMLSchema`.
Tutte le scelte che riguardano puramente la parte mobile sono state riportante a parte nella sezione 5.6 *Mobile* del presente documento.

5.1 XHTML

- Il logo del sito è stato realizzato con una `background-image` sfruttando la tecnica di *image replacement* affinché possa costituire informazione per il motore di ricerca.
- Nella parte pubblica del sito, il `path` svolge sia funzione di breadcrumb che di titolo della pagina: avendo un solo livello profondità, abbiamo ritenuto non necessario il breadcrumb all'interno della parte pubblica del sito.
- Per indicare al motore di ricerca l'importanza dei termini caratteristici del ristorante e della cucina giapponese è stato usato `strong`: questa è un'operazione di SEO (*Search Engine Optimization*).
- La struttura di ogni pagina è stata pensata per evidenziare le informazioni che si ritengono più importanti per un utente che naviga su un sito di ristorazione. Quindi si è deciso di tenere in alto e visibile in maniera chiara in ogni pagina della parte pubblica del sito i seguenti elementi:
  - il *luogo* dove si trova il ristorante, che è un'ancora a `dove-siamo.html`: una pagina che presenta informazioni più dettagliate sulla posizione del ristorante
  - il *numero di telefono*, che è un'ancora alla funzione di inizio della telefonata (`tel:`)
- Nella pagina `index.html` sono stati messi in evidenza l'*orario*, le *informazioni generali* riguardo il ristorante e la possibilità di usufruire del servizio di *take away*.
- Per fidelizzare il cliente, nella pagina `chi-siamo.html` mostriamo le personalità principali che gestiscono l'attività, come il proprietario del ristorante e i cuochi, raccontando un sunto del loro curriculum o le loro esperienza passate nel campo della ristorazione.
- Nella pagina `dove-siamo.html` è presente una breve descrizione testuale di dove si trova il ristorante accompagnata da un'immagine che ne indica la posizione rispetto ai vicini punti di interesse. Per rendere accessibile l'immagine a chi utilizza uno screen reader è stato utilizzato l'attributo

`longdesc` che associa una descrizione testuale dettagliata (definita nel file `indicazioni.txt`) all'immagine. Oltre a questo, è presente un link a Google Maps per agevolare l'utente nella ricerca di indicazioni precise per raggiungere il ristorante: nel caso in cui stia navigando da un dispositivo mobile ed abbia l'applicazione di Google Maps installata, verrà automaticamente lanciata l'applicazione.

- Nella pagina `curiosita.html` abbiamo inserito una serie di curiosità sulla cucina giapponese per far conoscere all'utente un po' della sua storia.
- Tutti i nomi delle pietanze nella pagina `menu.cgi` sono inseriti in un tag `strong` per indicarne l'importanza al motore di ricerca: questa è un'operazione di SEO (*Search Engine Optimization*).
- L'area amministratore permette di aggiungere, modificare o rimuovere pietanze o bevande dal menù.

5.2 CSS

- Si è cercato di utilizzare meno colori possibili.
- Si è scelto di utilizzare i colori di bandiera del ristorante, ricavati dal logo fornitoci dal proprietario del ristorante.
- Tali colori sono stati scelti antagonisti, per venire incontro a utenti con problemi di colorblindness ed aumentando così l'accessibilità del sito.
- Lo sfondo in legno dell'intero sito web vuole ricordare la stuoia usata nel processo di creazione del sushi e, più in generale, lo stile nipponico.
- Si è deciso di non inserire nel `footer` del sito l'icona di validazione ottenuta dal W3C in quanto non in linea con i colori del layout. Il sito è stato comunque interamente validato sia nel CSS che nel XHTML.
- Al foglio di stile `style.css` è stato applicato il *minifier* per un miglioramento delle performance.
- Viene fornito anche il file `style_extended.css`, che sarebbe il foglio di stile prima dell'applicazione del *minifier*, per migliorare la lettura in fase di correzione del progetto.

5.3 Perl/CGI

- Script in `Perl/CGI` vengono usati per gestire la visualizzazione dei dati nel database `XML` e gestire eventuali inserimenti, modifiche e rimozioni su di esso.
- Gli script utilizzati vanno ad impattare in entrambe le aree del sito, sia pubblica che privata (Area amministratore).

- Gli script `CGI` sono così suddivisi nella cartella cgi-bin:
  - script di pagina riservata al login all'area amministratore;
  - script di pagine riservate all'accesso pubblico del menù del ristorante;
  - script di pagine riservate all'accesso dell'amministratore (simile al menù presentato all'utente ma con form e pulsanti aggiunti, questi script sono anche incaricati della rimozione);
  - script di pagine riservate per l'aggiunta (prefisso `add-`);
  - script di pagine riservate per la modifica (prefisso `mod-`);
  - script per effettuare il logout che termina la sessione e reindirizza il browser;
  - moduli perl all'interno della cartella `My` che racchiudono le subruotines utilizzate per gli altri script `Perl/CGI`;
- Le uniche pagine accessibili senza credenziali sono:
  - `menu.cgi`: crea, prelevando i dati dal database `XML`, il menù del ristorante;
  - `login.cgi`: permette tramite l'inserimento di admin e password l'accesso all'area amministratore.
- Si può accedere all'area amministratore tramite login da ogni pagina del sito tramite link situato nel footer. Tutte le pagine della sezione amministrativa che effettuano modifiche in scrittura nel database sono protette da credenziali (riportate in prima pagina della presente relazione). Se corrette verrà visualizzata la pagina `private-menu-cibi.cgi` per prima. Il `nav` del sito pubblico sarà sostituito da un `nav` di sole due sezioni:
  - Cibi
  - Bevande

  da cui sarà possibile effettuare le operazioni sul file `menu.xml`. Si è deciso di separare le due componenti per agevolare il lavoro dell'amministratore perché tutto il menù del ristorante creava una pagina troppo lunga con l'aggiunta di form e pulsanti.
- Per far ciò si è ricorso all'utilizzo delle sessioni `Perl` e relativa libreria. Ogni pagina effettua al caricamento un controllo della sessione aperta e se risulta non valida o scaduta (dopo 60 minuti) l'amministratore o qualsiasi utente che prova l'accesso sono reindirizzati verso la pagina `accesso-negato.html`. Questa pagina viene mostrata in tutti i casi in cui l'utente o non ha le credenziali o la sessione è scaduta.
- Poiché l'area amministrativa conterà un solo amministratore non si è ritenuto necessario l'utilizzo di parametri all'interno della sessione tranne l'username.

- Per l'uscita invece sono stati inseriti due link 'uscita' nel top e nel bottom delle pagine che chiudono la sessione e indirizzano alla pagina `index.html`.
- Per poter rimuovere un piatto o una bevanda è sufficiente utilizzare i pulsanti "Rimuovi" situati vicino ogni elemento del menù del ristorante nelle pagine:
  - `private-menu-cibi.cgi`
  - `private-menu-bevande.cgi`

  Una volta premuto il pulsante la pagina viene ricaricata e un messaggio ad inizio pagina avverte se tutto è andato a buon fine.
- Per poter aggiungere un piatto o una bevanda è necessario effettuare il submit sulle form di aggiungi dalle pagine:
  - `private-menu-cibi.cgi`
  - `private-menu-bevanda.cgi`

  perché da esse vengono prelevati i campi `hidden` necessari al corretto funzionamento dell'inserimento. È possibile aggiungere piatti e bevande ma non portate o liste di bevande (vini, birre, etc.) in quanto già sufficienti a comprendere eventuali piatti o bevande nuove. Non si è comunque escluso che in un futuro questa funzionalità non possa essere aggiunta. Prima di ogni aggiunta i nuovi dati inseriti sono opportunamente controllati tramite espressioni regolari per far sì che il database risulti ben formato e valido secondo il relativo schema.
- Per poter modificare un piatto o una bevanda è necessario effettuare il submit sulle form di modifica dalle pagine:
  - `private-menu-cibi.cgi`
  - `private-menu-bevanda.cgi`

  perché da esse vengono prelevati i campi `hidden` necessari al corretto funzionamento della modifica. Le form di modifica sono infatti già compilate con i dati presenti nel file `menu.xml`. Prima di ogni modifica i nuovi dati inseriti sono opportunamente controllati tramite espressioni regolari per far sì che il database risulti ben formato e valido secondo il relativo schema.
- Sono stati scritti in totale tre package `Perl` con lo scopo di riutilizzare il codice e di renderlo il più possibile manutenibile. Essi dividono le funzioni in base al loro luogo di utilizzo. Di seguito la lista dei package e una descrizione delle funzioni racchiuse all'interno:
  - `Base`: racchiude tutte le funzioni per la stampa dell'`HTML` più frequente (come ad esempio l'`header`) e tutte le funzioni per agevolare l'accesso sia in lettura che scrittura sul database `XML`;

  - `PrintMenu`: racchiude le funzioni che si occupano della raccolta di dati dal database `XML` e alla stampa con o senza la presenza di form per l'area amministratore;
  - `Operation`: raccoglie tutte le funzioni che costruiscono le form di aggiunta e modifica e controllano che i dati inseriti siano corretti e soprattutto che non invalidino il database.

5.4 Javascript

- Il menu del sito (chiamato d'ora in poi anche `nav`) è inizialmente visibile, così se `Javascript` non è attivo il sito ha una degradazione elegante.
- Se invece `Javascript` è attivo e la navigazione viene effettuata da mobile, allora il `nav` viene nascosto e rimane accessibile dall'apposita `icon hamburger`.
- È presente il controllo tramite `Javascript` dei dati inseriti nei campi delle form, per segnalare eventuali errori all'utente in maniera immediata, senza attendere una risposta del server.
- Nella pagina `menu.cgi` e `private-menu.cgi`, il menù è diviso in categorie inizialmente chiuse per diminuire la lunghezza della pagina e dare all'utente una visione generale delle categorie di pietanze e bevande presenti nel menù.
- Nel caso in cui il `Javascript` non sia attivo, il menù verrà mostrato per intero, ma viene aggiunta una funzionalità di ricerca veloce (in alto nella pagina) e, alla fine di ogni categoria, un'ancora per tornare all'inizio della pagina.

5.5 XML e XMLSchema

- Per modellare il menù del ristorante in un database XML si è deciso di fare una forte divisione logica tra cibi e bevande.
- Quest'ultimo è presente nella cartella `data` ed è il documento `menu.xml`: esso è ben formato e valido secondo lo schema `menu.xsd`, presente nella stessa cartella.
- Il modello scelto per modellare lo schema in `XMLSchema` è Tende alla Veneziana, in quanto sfrutta i tipi nominali e, basandosi sul menù da modellare, era necessario riutilizzare i tipi definiti in precedenza.
- È stato definito un tipo complesso `stringWithSpan` per permettere ai nomi (sia delle portate che delle bevande) e alle descrizioni di contenere, nel caso in cui sia presente una parola straniera, un tag `span` che definisca tramite l'attributo `xml:lang` la lingua della parola. `stringWithSpan` permette anche di contenere il tag `abbr` per le eventuali abbreviazioni.

- `stringWithSpan` è stato definito come un tipo misto contenente una `xsd:choice` (con `minOccurs="0"` e `maxOccurs="unbounded"`) che permetta una scelta tra l'elemento `span` e `abbr`.
- Il tipo di `span` è stato definito con un tipo complesso dal contenuto semplice, ricavato per estensione da `xsd:string` aggiungendo un attributo `xml:lang` : quest'ultimo fa riferimento all'attributo `xml:lang` dello schema `xml.xsd` contenuto in http://www.w3.org/2001/xml.xsd, quindi è stato necessario importare il namespace "http://www.w3.org/XML/1998/namespace".
- Il tipo di `abbr` è molto simile al tipo di `span`, con l'aggiunta di un attributo `title` che permette di indicare il nome completo della parola abbreviata.
- La definizione del tipo `stringWithSpan` è stata necessaria data l'elevata presenza di parole straniere all'interno del menù da modellare.
- È stato definito un tipo `Tpiatto` che è stato riutilizzato per ogni piatto presente all'interno del menù: ciò è stato possibile impostando `minOccurs="0"` sugli elementi `numero` e `descrizione` presenti al suo interno. Ciò ha permesso di riutilizzarlo in vari contesti.
- Ogni tag `portata`, `piatto`, `listaBevande` e `bevanda` presenta un attributo `id` che permette di identificarli in maniera univoca.
- Il tipo di prezzo è stato definito con un tipo complesso dal contenuto semplice, ricavato per estensione da `xsd:decimal` aggiungendo l'attributo `valuta` di tipo `xsd:string` : questo attributo non si trova nel file `menu.xml` in quanto è stato definito un valore di default impostato a `"euro"` e nel caso da modellare non erano presenti altre valute.
- Il file `admins.xml` invece contiene l'username e password per accedere all'area amministratore, prevedendo la possibile aggiunta di ulteriori amministratori in futuro: nella stessa cartella è fornito lo schema secondo cui il documento `XML` è valido: `admins.xsd`; lo schema è stato definito anch'esso in `XMLSchema` seguendo il modello Tende alla Veneziana.

5.6 Mobile

Una grossa fetta di utenza del sito è composta dai ragazzi più giovani, come dimostrato dall'analisi presente alla sezione 3 *Analisi dell'utenza* del presente documento. Per questo motivo il gruppo ha deciso di definire lo stile del sito anche per i dispositivi mobile, tenendo conto che la maggior parte degli accessi al sito da parte di ragazzi giovani verrà effettuata da smartphone o tablet.
Di seguito sono riportate le scelte progettuali effettuate durante lo sviluppo del progetto che il gruppo ha ritenuto degne di nota per quanto riguarda la parte mobile del sito.

- È stata usata la `hamburger icon` come simbolo del menu del sito (`nav`), ritenuta adeguata in quanto è ormai una consuetudine per gli utenti associare ad essa il menu del sito.
- Per agevolare la visita del sito e rendere più piacevole l'esperienza dell'utente, abbiamo dato all'utente la possibilità di aprire e chiudere il `nav`: per far ciò è stata utilizzata una funzione `Javascript` che mostra e nasconde il `nav` attraverso la `hamburger icon`.
- Abbiamo convenuto che la parte mobile sia dedicata interamente agli utenti esterni, quindi abbiamo impedito l'accesso all'area amministratore da mobile: esso è permesso solo da desktop.
- Il sito è stato testato sui seguenti browser mobile: Google Chrome, Safari.

Figura 2: Esempio di pagina vista da dispositivo mobile con menu a tendina aperto

6 Accessibilità

6.1 Chiarezza e semplicità

È stato scelto un font molto chiaro per permettere all'utente di leggere senza sforzi il contenuto. Si è cercato di mantenere la semplicità in ogni parte del sito, sia nel contenuto che nella presentazione.
Per maggiore chiarezza, ma anche per permettere all'utente di salvare una pagina del sito tra i preferiti senza che ciò crei confusione, il tag `title` nell'`head` di ogni pagina è stato definito dal particolare al generale.

6.2 Colore

Le disabilità visive testate sul sito sono le seguenti: deuteranopia, monochromacy, partialMonochromacy, protanopia e tritanopia.
L'esito è stato che indipendentemente dal tipo di malattia il sito si presenta in maniera chiara e accessibile.

Una nota particolare va fatta per i `link`: essi sono colorati di rosa salmone, però sono rafforzati dal grassetto proprio per renderli maggiormente distinguibili anche nel caso in cui l'utente non sia in grado di vedere il colore; a tal riguardo, il caso più lampante per il nostro sito è la monochromacy: se non fossero in grassetto i link sarebbero difficilmente distinguibili dal resto del testo.

Di seguito sono riportati una serie di esempi di una pagina del sito vista da una persona con disabilità visiva.

Figura 3: Esempio di pagina vista da chi soffre di protanopia

Figura 4: Esempio di pagina vista da chi soffre di deuteranopia

Figura 5: Esempio di pagina vista da chi soffre di tritanopia

Figura 6: Esempio di pagina vista da chi soffre di monochromacy

6.3 Parole straniere

All'interno del sito sono presenti molte parole giapponesi, per ovvii motivi: in ognuno di questi casi è stato definito l'attributo `xml:lang="ja"` tra gli attributi del tag all'interno del quale figuravano (ad esempio, `strong`) oppure tra gli attributi di uno `span` opportunamente piazzato.

All'interno del menù del ristorante invece sono presenti molte parole straniere: giapponesi, francesi, tedesche e spagnole: è stato definito uno `span` opportuno per ognuna di queste occorrenze, con attributo `xml:lang` impostato sulla lingua della parola straniera.

Queste due operazioni hanno permesso di favorire la lettura del sito da parte di uno screen reader.

6.4 Form

Nella pagina `private-menu.cgi`, presente nell'area amministratore, è possibile effettuare delle modifiche al menù del ristorante.
Nelle `form` di questa pagina si è preferito semplificare la parte di presentazione pur di mantenere la validazione dell'`XHTML` e quindi mantenere un buon livello di accessibilità.

Si deve tenere conto anche del fatto che questa pagina è accessibile unicamente all'amministratore, quindi il sacrificio in fatto di presentazione è accettabile in quanto non si toglie nulla all'utente che visita il sito.

Più in generale, in tutte le `form` del sito non è stata definita una `legend` perché il titolo della pagina, stampato con un `<h3>` proprio sopra la `form`, rende totalmente chiaro all'utente lo scopo della `form` ed il motivo per cui quei tag `input` sono raggruppati.

Gli errori trovati durante l'inserimento di dati nelle form sono stati segnalati in maniera elegante, sfruttando il `Perl` per il controllo lato server ma anche il `Javascript` per un controllo immediato (se attivo) lato client.

Figura 7: Esempio di come vengono mostrati gli errori nelle form

6.5 Accessibilità da tastiera

Una condizione necessaria ad un sito per essere accessibile è quella di essere navigabile utilizzando solamente la tastiera.
Ogni pagina del sito è stata testata ed il risultato è che il sito è navigabile utilizzando `Tab` ed `Enter`, ad eccezione delle pagine `menu.cgi` e `private-menu.cgi` (rispettivamente, parte pubblica e amministratore) con `Javascript` attivo.
Infatti, nel caso in cui `Javascript` non sia attivo, il sito è navigabile totalmente da tastiera.
Il gruppo ha deciso di non utilizzare gli attributi `tabindex` per i `link` del sito in quanto il flusso logico del sito è semplice e lineare.

6.6 Accessibilità tramite screen reader

Il sito è completamente navigabile attraverso il browser testuale Lynx, anche nell'area amministratore.
Sono stati presi alcuni accorgimenti che agevolavano di molto la fruizione del contenuto, come la funzionalità "Salta la navigazione", che permette di saltare

la lettura del `nav`, e "Ricerca veloce", che permette di "saltare" direttamente ad una sezione del menù del ristorante.

L'unico problema riscontrato è la codifica dei caratteri accentati che vengono interpretati da Lynx in una codifica diversa da UTF-8. Si suppone sia soltanto un problema di impostazioni di Lynx.

Di seguito gli screenshot riportati riguardano la pagina `index.html`, in cui si può notare l'ancora per saltare la navigazione, e la pagina `menu.cgi` (ritenuta la più importante), in cui si può fruire facilmente di ogni piatto grazie alla funzionalità di "Ricerca veloce" e all'ancora "Torna all'inizio" posizionata alla fine di ogni portata.

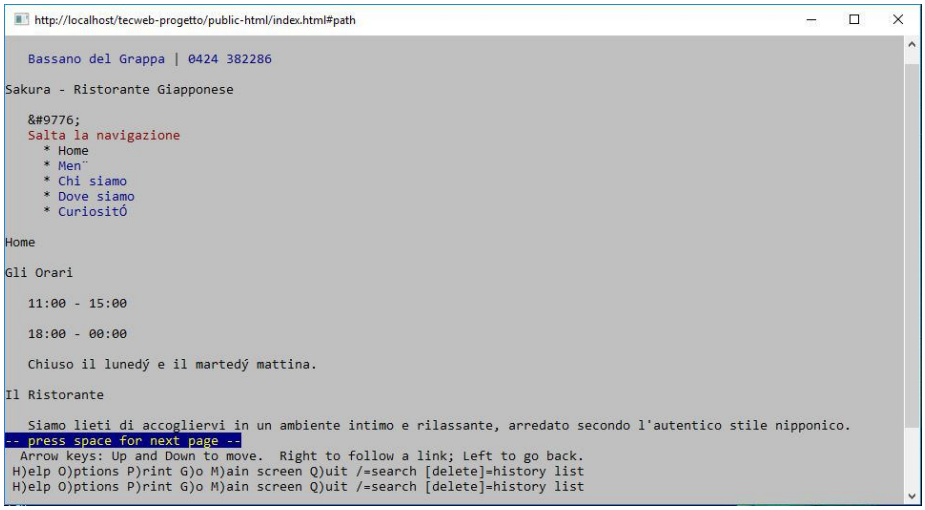

Figura 8: Esempio di come viene mostrata la pagina index.html da Lynx

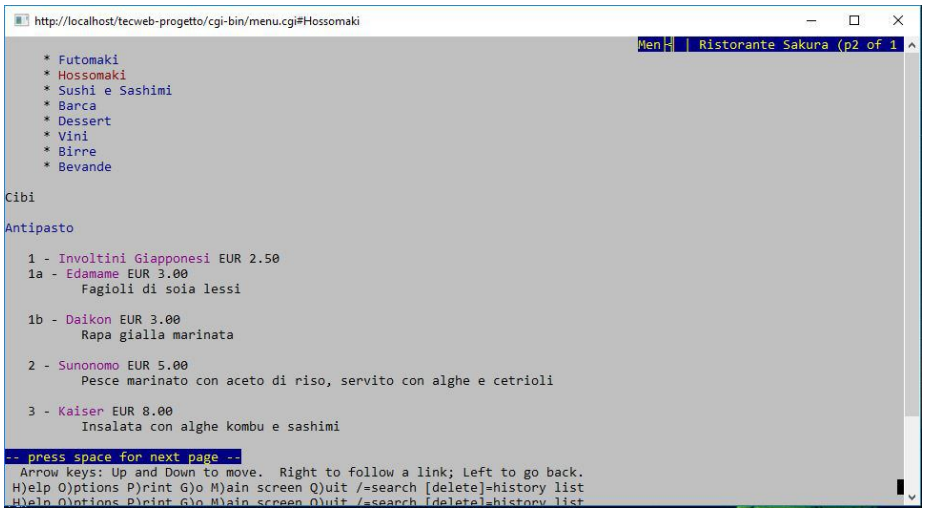

Figura 9: Esempio di come viene mostrata la pagina menu.cgi da Lynx

6.7 Compatibilità con i browser

Il sito è stato testato sui seguenti browser:

- Google Chrome versione 48.0.2564.103 m;
- Mozilla Firefox versione 44.0;
- Internet Explorer versione 11.63.10586.0;
- Microsoft Edge versione 25.10586.0.0;
- Safari (su Mac OS X) versione 9.0.2;
- Google Chrome per mobile su dispositivo Android;
- Safari per mobile su dispositivo iOS.

7 Istruzioni per l'utilizzo su altri server

Il sito è stato caricato su:

http://tecnologie-web.studenti.math.unipd.it/tecweb/~dcastell/

Il pacchetto consegnato è il sito che è stato caricato sul server nella sua interezza, con l'aggiunta degli schema in XMLSchema che validano i file XML della relazione.

Nel caso si volesse caricarlo su altri server è necessario effettuare le modifiche sui path relativi a ritroso segnalati nel sito http://www.studenti.math.unipd.it/index.php?id=corsi_tecweb, in particolare per gestire tutti i file e le cartelle contenuti in `public_html`.