

Progetto Sistemi Informativi - A.A. 2013/2014

tecnologie-web.studenti.math.unipd.it/tecweb/~maduso/

Matteo Aduso, Damiano Costa, Leonardo Bortoletti

25 Marzo 2014

1 Introduzione

Questo progetto nasce come sito per un Pub, chiamato **GitPub**.

È un pub fittizio, ma data la struttura del sito si potrebbe adattare abbastanza facilmente ad un qualsiasi pub che ha gli stessi contenuti da mostrare. Abbiamo ipotizzato un pub che debba gestire degli eventi (una band che deve suonare, una serata Karaoke, etc), abbia un proprio menu con una foto della pietanza ed eventualmente una storia del locale. Per la gestione delle varie sezioni informative del sito abbiamo costruito una parte per gli amministratori.

Il nome del progetto deriva da git, il software di versioning usato per lo sviluppo del codice nella maniera più collaborativa e veloce possibile.

Il sito è stato creato rispettando gli standard di accessibilità del W3C, utilizzando CSS2.1, XHTML1.0 Strict e database XML con relativo XMLSchema(XSD).

2 Analisi e Progettazione

2.1 Analisi delle utenze

Dall'analisi dell'utenza è risultato che il nostro sito è pensato per qualsiasi target d'età: che tu sia un bambino che consulta il menu per vedere cosa mangiare con i genitori o un ragazzo che vuole vedere negli eventi se viene trasmessa la partita Champions League della sera. L'elenco degli eventi futuri, non solo quelli dell'immediato, e il menù del locale, estende sicuramente i possibili utenti che potranno scegliere la serata e il piatto più adatto a loro.

2.2 Scelta dei contenuti

Secondo una nostra analisi, prendendo spunto da qualche sito, abbiamo stilato una lista di possibili contenuti di interesse per un'azienda che debba gestire un pub:

- **Eventi:** l'insieme di eventi promossi dal pub;
- **Menu:** il menu culinario tipico del pub;
- **Contatti:** i vari contatti (e-mail, telefono, indirizzo ed eventuali social);
- **Chi siamo:** la storia del locale (com'è nato, chi ci lavora, etc).

3 XML & XMLSchema

Abbiamo creato un file XML per ogni contenuto: essendo un sito dinamico, ogni file XML viene utilizzato come database per memorizzare ogni modifica fatta dall'amministratore del sito.

Diversamente, nel file Users.xml sono memorizzati l'username e password dell'amministratore.

Lo schema scelto per ogni file XML è il modello Tende alla Veneziana.

Per ogni elemento definito come string nell'XMLSchema è prevista la gestione della doppia lingua (IT, EN) tramite l'attributo lang. Site.xml definisce per ogni pagina tutti i metatag d'installazione, i collegamenti ai fogli di stile e i link alle altre pagine. Esso contiene anche le immagini comuni a tutte le pagine (logo, prefooter).

3.1 File XML

- Site.xml
- Events.xml
- Menu.xml
- About.xml
- Contats.xml
- Users.xml

4 CSS

4.1 Struttura XHTML

Le varie pagine del sito sono state strutturate in due modi diversi:

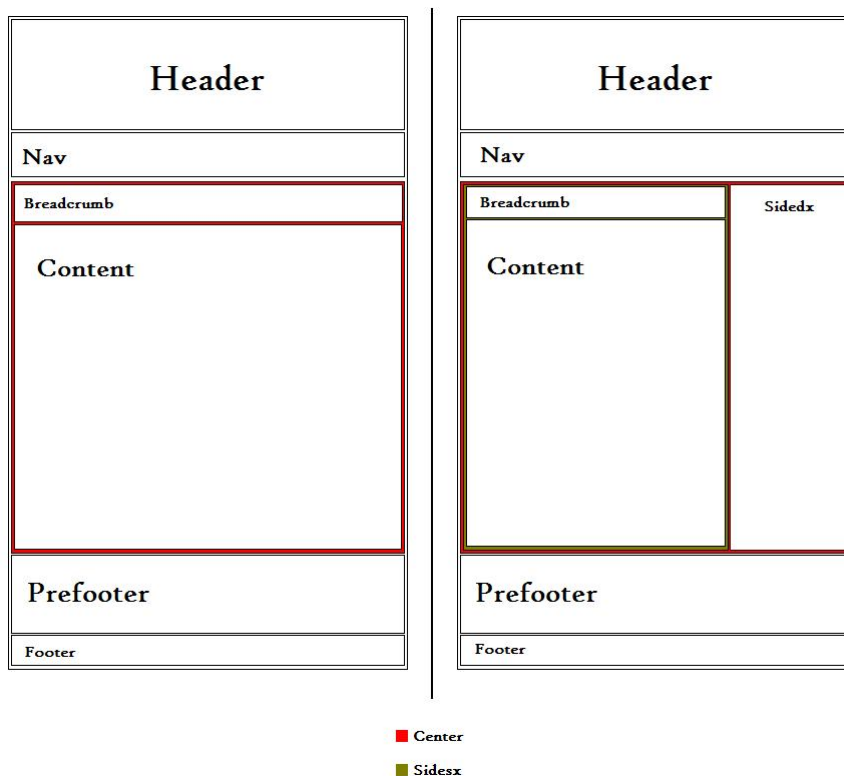


Figura 1: fig:CSS Pagina

La differenza tra le due strutture sta nel contenuto del tag Center:

- Nel primo grafico troviamo la struttura della pagina index e delle parti del back-end. Nel tag Center troviamo solo il tag Breadcrumb e il tag Content (Breadcrumb non compare nelle pagine di back-end).
- Nel secondo grafico troviamo la struttura delle altre pagine (Menu, Eventi, Contatti, Chi Siamo), dove abbiamo diviso in due parti (sidesx, sidedx) l'interno della parte centrale del sito. Il tag sidesx è strutturato come il center del primo grafico, mentre il tag sidedx contiene l'ultimo evento in programma e una pietanza scelta casualmente dal menù.

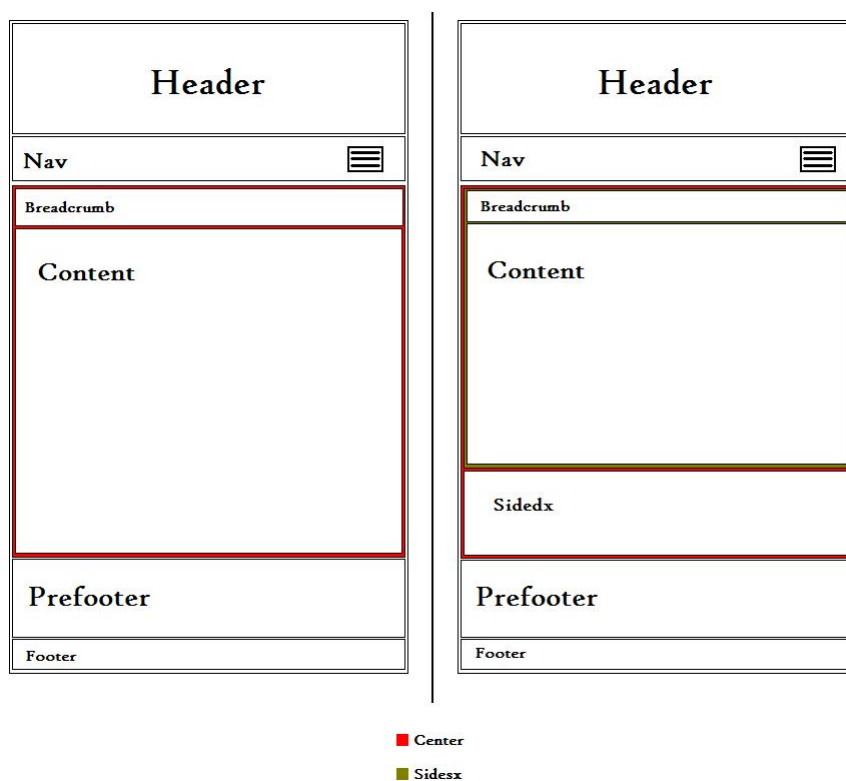
Descrizione dei vari tag del sito

- L'Header presenta due elementi, uno contenente il logo e uno l'opzione della lingua (Italiano, Inglese).
- Il tag Nav contiene il menu del sito: Home, Eventi, Menù, Chi Siamo, Contatti (con il login al back-end compaiono anche i link alle pagine di gestione).

- Il tag Breadcrumb serve all'utente per sapere in che pagina si trova, e da la possibilità di accedere alle pagine visitate precedentemente grazie ai link presenti.
- Il tag Content contiene le informazioni della pagina che sto visitando.
- Il tag Sidedx riporta l'ultimo evento in programma e una pietanza scelta casualmente dal menù.
- Nel teg Prefooter sono esposti gli sponsor del Pub.
- Il Footer riporta l'immagine riguardante gli standard W3C e i copyright.

4.2 Struttura Mobile

Le varie pagine del sito (parte Mobile) sono state strutturate in due modi diversi:



La differenza sta nella struttura del Center:

- Nel primo grafico troviamo la struttura della pagina index e delle parti del back-end, che ricorda la struttura della relativa pagina screen.
- Nel secondo grafico troviamo la struttura delle altre pagine (Menu, Eventi, Contatti, Chi Siamo), dove i contenuti non cambiano, ma la struttura è pensata per un display ristretto, quindi il sidedx è stato accodato al sidesx.

La differenza sostanziale sta nel Nav, il quale non è sempre in primo piano ma oscurabile tramite il tasto del menù a tendina che rende la navigazione più piacevole lasciando più spazio agli altri contenuti.

4.3 Grafica CSS

Si è deciso di suddividere in tre file il CSS: screen, print, mobile.

Screen

Il foglio di stile principale screen utilizza esclusivamente valori percentuali ed em per rendere la pagina più dinamica, elastica.

Mobile

Il foglio di stile mobile rende più accessibile la pagina per schermi inferiori a 768px. Si differenzia nell'uso del menu a tendina nel Nav.

Print

La versione per la stampa è stata creata semplificando al massimo la pagina e dando valore solo alle informazioni più importanti. Abbiamo tolto il Nav, l'opzione della lingua e il footer per rendere più snello il contenuto. In più abbiamo modificato la grandezza del font del corpo a 12pt, perché è una grandezza standard sui fogli A4.

5 Il motore del sito, PERL

5.1 Il Front-End

Il front-end è composto dalle pagine:

- index.pl: la home page del sito;
- events.pl: la pagina di presentazione degli eventi, e presentazione di un singolo evento selezionato;
- menu.pl: la pagina di presentazione delle varie pietanze;
- about.pl: la pagina che racconta la storia del pub;
- contacts.pl: la pagina che raccoglie tutte le informazioni utili per contattare il pub.

Nella home viene mostrato l'ultimo evento in programma e una pietanza scelta casualmente.

Nella pagina eventi vengono mostrati tutti gli eventi con la descrizione lunga massimo 80 caratteri, con la possibilità di sceglierne uno e vedere quindi la descrizione per intero.

Nella pagina menu è possibile vedere la lista di tutte le pietanze offerte dal pub in ordine alfabetico.

La pagina contatti mostra le informazioni di contatto con una piccola finestra di google maps(tag **object**).

5.2 Il Back-End

Il back-end è accessibile solo entrando nella pagina backIndex.pl, dove poi si può fare il login. Se si tenta di entrare in una delle altre pagine del back-end senza essersi autenticati, si viene reindirizzati alla pagina di accesso(backIndex.pl).

Le pagine di back-end sono:

- backIndex: pagina di login, logout e cambio password;
- backEvents: pagina per l'aggiunta, la modifica e la cancellazione di eventi;
- backMenu: pagina per l'aggiunta, la modifica e la cancellazione di pietanze;
- backAbout: pagina per la modifica del database about.xml;
- backContacts: pagina per la modifica del database contacts.xml;
- backLogos: pagina per l'aggiunta o cancellazione di loghi presenti nel preFooter.

Tutte le pagine di back-end ovviamente modificano solo i relativi file xml, senza andare a modificare nessuna pagina perl e nessuna pagina html (per altro nel sito nemmeno esistono).

Ogni pagina del back-end fa un controllo degli input prima di modificare qualcosa nel database, in particolare controlla la lunghezza del testo, il formato della data(come fa checkDate, descritto nel paragrafo relativo a Javascript) e il controllo dei metatag sui testi.

5.3 Libreria di funzioni, lib.pl

Inoltre è presente una libreria di funzioni usata da tutti gli script perl del sito. Le funzioni presenti sono le seguenti:

- `by_date`: usato per l'ordinamento degli eventi in `events.pl`;
- `by_title`: usato per l'ordinamento alfabetico in `menu.pl`;
- `startPage`: setta la lingua e poi chiama `divHeader` e `divNav`;
- `sessionHeader`: usato solo da `backIndex.pl`, dopo il login genera l'header per il passaggio delle cookies e reindirizza sulla stessa pagina;
- `divHeader`: stampa la parte iniziale della pagina, dalla codifica al header dello stile;
- `divNav`: stampa il menu, evidenziando la pagina che si sta visitando;
- `divSideDx`: stampa il pannello di destra, chiamato da `endPage` solo nelle pagine in cui deve comparire;
- `divBreadcrumb`: stampa il breadcrumb evidenziando la pagina che si sta visitando;
- `endPage`: stampa la parte finale della pagina, chiama `sidedx` se deve essere presente, `divPreFooter` e `divFooter`;
- `divPreFooter`: stampa il prefooter dove sono presenti i loghi;
- `divFooter`: stampa il footer con i copyright e i loghi di XHTML 1.0 e CSS;
- `food_random`: usato in `sidedx` e nella home per mostrare una pietanza a caso;
- `toXmllang`: converte i metatag presenti nel testo in tag span con il corretto attributo `xml:lang`;
- `checkXmllang`: controlla se in una stringa i tag sono chiusi correttamente, aperti correttamente e che non ci siano tag annidati;

6 Java Script

6.1 Dinamicità wrapper con adattatoreWidth.js

Questo script serve a rendere il design delle pagine responsivo calcolando dinamicamente la larghezza del wrapper.

Infatti esso viene calcolato in base alla larghezza della finestra del browser: il wrapper occuperà l' 80% della finestra se essa è più larga di 1280px, il 100% se la finestra è larga meno di 768px (primo punto di rottura nel CSS) e infine una percentuale che cresce in modo lineare tra 80% e il 100% se la larghezza della finestra è compresa tra 1280px e i 768px.

6.2 util.js, il controllore dei form

Controlla l'input dei form nella parte di back-end quando escono dal focus, e prima di inviare l'input.

Nel caso dell'uscita dal focus(`onBlur`) viene richiamata la funzione `checkInput` passandogli come argomento l'oggetto input chiamante, questa funzione controlla se a questo input sono associati controlli da fare, per farlo controlla la matrice `eventsErrType` e chiama le relative funzioni per effettuare i controlli. `eventsErrType` è una matrice di tipo hash, a cui ogni nome di input è associata un'altra matrice, contenente un'altra associazione biunivoca tra il nome della funzione e una relativa lista di parametri per la funzione (per esempio in `checkLength` è il numero minimo di caratteri e il relativo messaggio di errore da dare).

Nel caso di invio dell'input, viene richiamata la funzione `checkAllInput` passandogli come argomento l'oggetto form chiamante, in questo modo la funzione esegue `checkInput` per ogni elemento input presente in quel specifico form.

Le funzioni sono:

- `checkDate`: funzione che controlla il formato della data, cioè se composta di 4 cifre, il separatore -, 1 o 2 cifre, il separatore, 1 o 2 cifre (YYYY-MM-DD);
- `checkLength`: controlla se la lunghezza del testo è maggiore uguale di un numero prefissato di caratteri (dipende dalla matrice `eventsErrType`);
- `checkTagUnclosed`: funzione che verifica la corretta apertura e relativa chiusura dei metatag di lingua presenti nel testo;
- `checkPasswd`: viene usato per controllare se la password nuova e la sua ripetizione sono uguali, usato solo in `backIndex.pl` per il cambio password.

7 Accessibilità

Per rendere accessibile il sito a tutti gli utenti, in particolare ai diversamente abili, abbiamo seguito le varie regole del W3C. In particolare ci siamo soffermati sui seguenti aspetti:

- Separazione della struttura, presentazione e comportamento;
- Presenza del Breadcrumb per segnalare la posizione attuale dell'utente;
- Regole per le sintassi dello screen-reader con abbreviazioni e acronimi, e possibilità di saltare la barra di navigazione con un link;
- Selezione della lingua Italiano - Inglese delle pagine, per rendere la pagina accessibile a più utenti, essendo l'Inglese la lingua più diffusa nel Web;
- Utilizzo degli attributi Alt;
- Inserimento tabIndex per facilitare la navigazione;
- Possibilità per l'amministratore di definire parole inglesi o italiane (a seconda dell'ambito) con xml:lang usando dei semplici metatag([it]Parola italiana[/it] [en]English word[/en])

8 Testing

Il sito è stato testato nei seguenti browser:

- **Internet Explorer** v 7
- **Google Chrome** v 14.0
- **Mozilla Firefox** v 3.5
- **Opera** v 10.0
- **Safari** v 5

Test per l'accessibilità effettuati:

- Validazione dei fogli CSS e delle pagine XHTML tramite W3C;
- Validazione di XMLSchema e relativi XML;
- Validazione tramite TotalValidator;
- Contrasto dei colori: sono stati testati tutti i contrasti cromatici e non abbiamo riscontrato errori;

Sono stati testati i vari colori del sito; i risultati sono i seguenti:

- Vista sito normale;
- Vista sito con riproduzione della deuteranopia;
- Vista sito con riproduzione della protanopia;
- Vista sito con riproduzione della tritanopia.

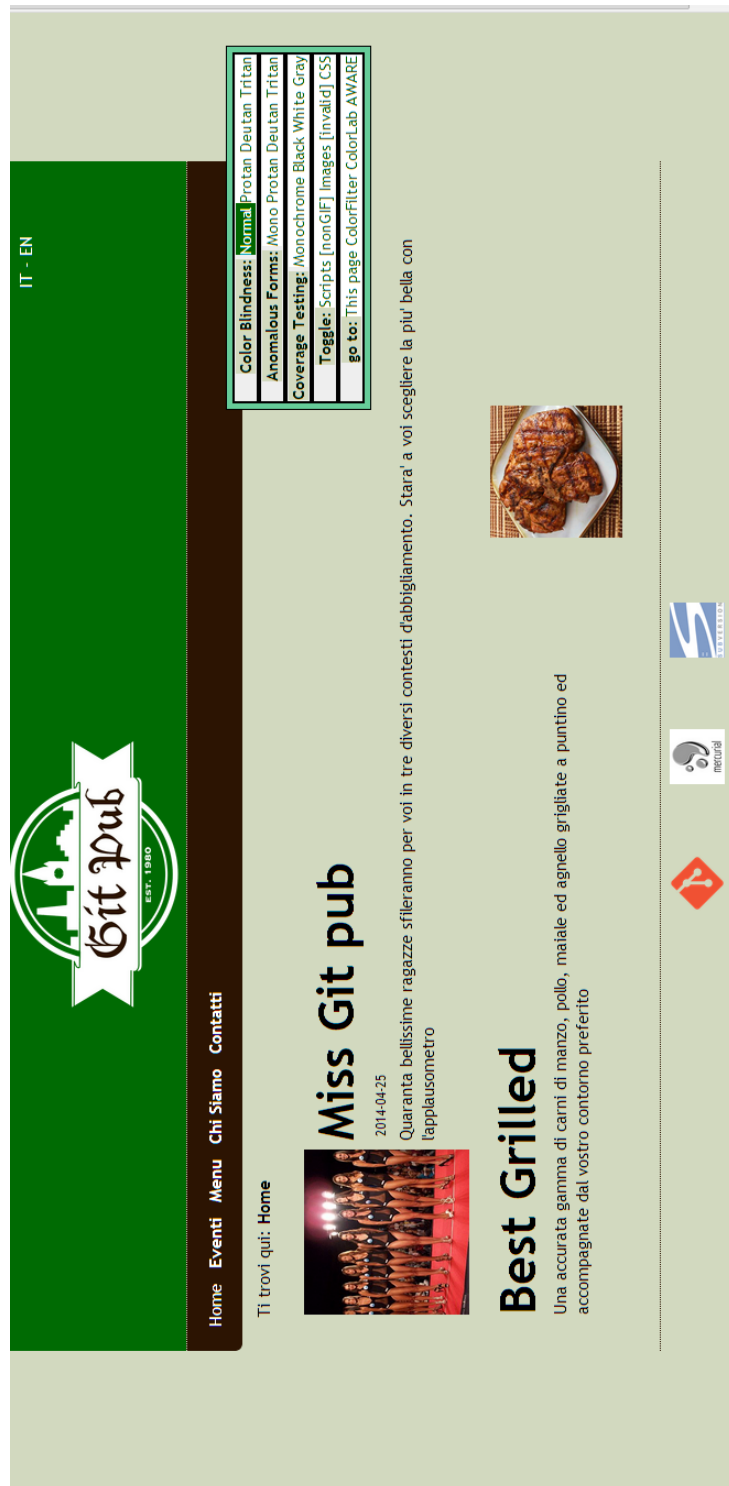


Figura 2: fig: Normale

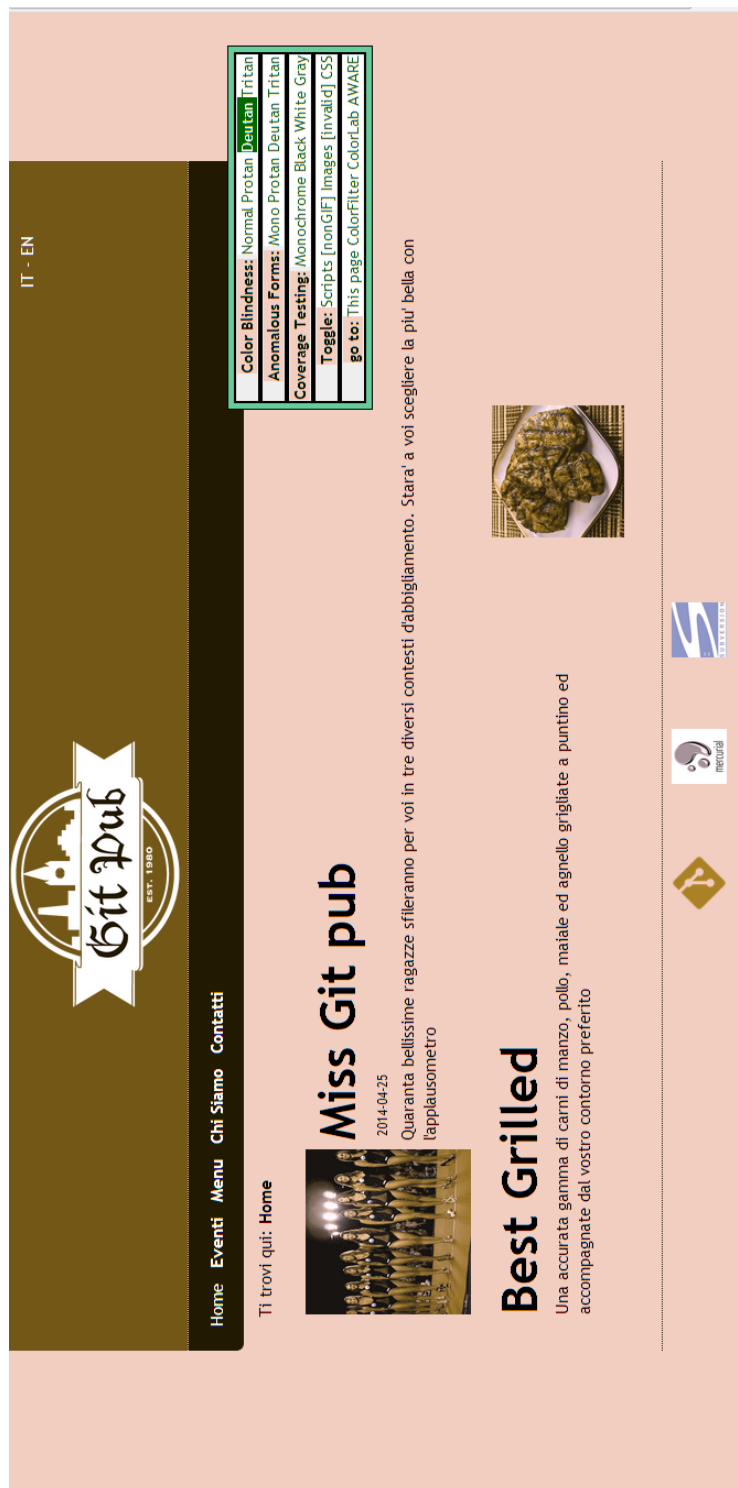


Figura 3: fig: Deuteranopia

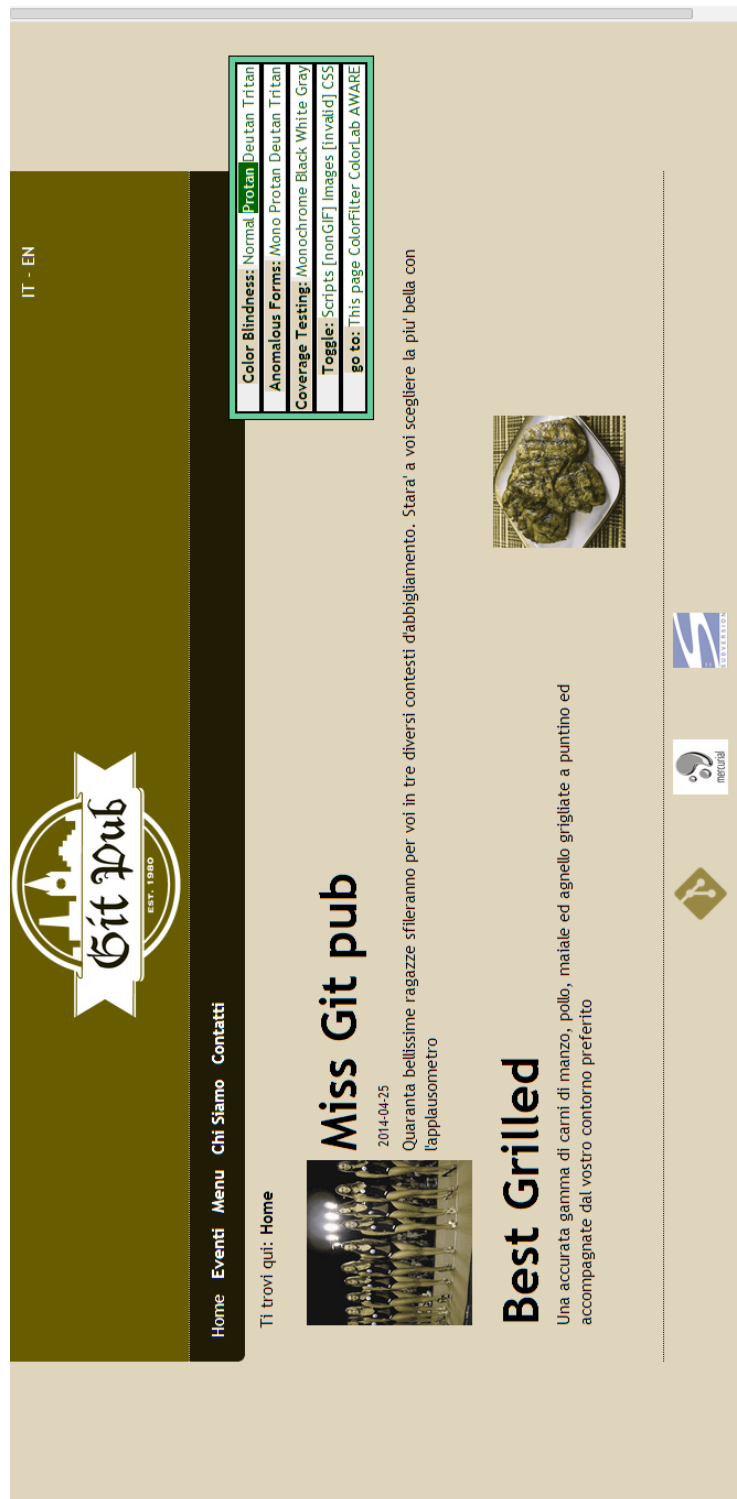


Figura 4: fig: Protanopia

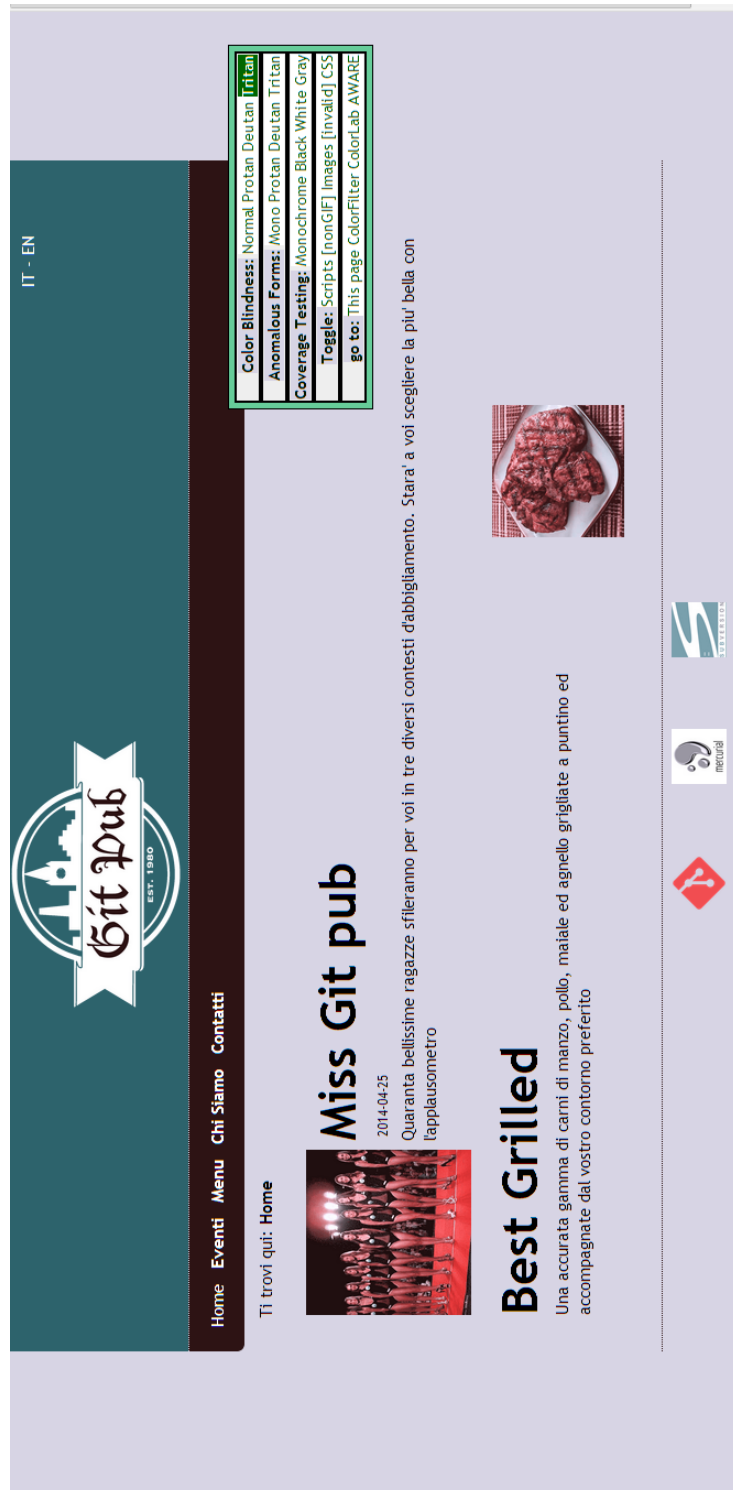


Figura 5: fig: Tritanopia

9 Conclusioni

Le access key non sono state messe volutamente in quanto riteniamo possano creare conflitti con le access key del browser utilizzato dall'utente.

È stato usato CSS3, in realtà inizialmente era previsto il solo uso di CSS2.1, infatti è presente uno script navDisplay.js che doveva implementare il menù a tendina che ora è implementato in maniera nativa con CSS3; in ogni caso i CSS usati usano poche proprietà del CSS3 (bordi, menù a tendina).