

Relazione progetto Tecnologie Web

Rosa Rosae

Enrico Ceron

Giacomo Beltrame

23 giugno 2014

Indirizzo web del sito

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

Credenziali d'accesso

Utente amministratore admin, admin

Utente senza privilegi utente, utente

Indirizzo e-mail referente

`enrico.ceron.1@studenti.unipd.it`

Sommario

L'obiettivo che ci ha guidato nello sviluppo del sito web denominato *Rosa Rosae* è ovviamente l'utilizzo della più grossa porzione di contenuti e competenze ricavati dalla frequenza del corso di Tecnologie Web e dallo studio individuale sulla materia. La scarsa importanza del contenuto, rispetto a quella di struttura e presentazione, ha fatto ricadere la scelta tematica su di un argomento dal quale fosse facile copiare senza scrupolo delle informazioni: da qui la decisione di realizzare un sito sul giardinaggio. La maggior parte delle forze la si è, così, potuta spendere nella realizzazione di una struttura il più possibile funzionale e facilmente amministrabile nel lungo periodo e di una presentazione il più possibile "classica" e funzionale, che non rompesse alcuno schema mentale dell'utente, ma allo stesso tempo moderna e accattivante. Le pagine che compongono il sito sono molto poche, nella forma simile ma diversificate nella sostanza, alcune di esse esclusive a utenti con privilegi di amministratore.

Indice

1	Descrizione generale	3
1.1	Di cosa stiamo parlando	3
1.2	Struttura dei database	3
1.3	Le pagine che compongono <i>Rosa Rosae</i>	4
2	Analisi dell'utenza	4
3	Scendendo nel dettaglio	4
3.1	I punti d'incontro	4
3.2	Home page	5
3.3	La pagina dedicata	5
3.4	I contatti	7
3.5	Le form	7
3.5.1	Registrati e <i>logga!</i>	7
3.5.2	E ora come commento?	7
3.5.3	Crea contenuto	8
4	Dietro le quinte!	8
4.1	Script server-side: <i>Perl</i>	8
4.2	Script client-side: <i>JavaScript</i>	10
5	Mobile	10
6	Accessibilità e testing	11
7	Conclusione	11

1 Descrizione generale

In questa sezione sarà descritta la struttura generale del sito e i principali motivi che hanno portato alla sua definizione. Saranno elencati i file che lo compongono per permettere di orientarsi seguendo i link e gli effetti degli script.

1.1 Di cosa stiamo parlando

Il sito si chiama *Rosa Rosae* e consiste, nel cuore, di un database XML *plants.xml* che contiene una moltitudine di piante con le relative specifiche e descrizione, ognuna accompagnata dai commenti degli utenti. Gli utenti sono registrati all'interno di un secondo file XML *users.xml*. Tutte le pagine dinamiche del sito, generate da script perl di cui si discuterà in seguito, pescano i dati necessari dal database *plants.xml* e dal foglio di stile XSLT relativo alla pagina. Le pagine statiche sono scritte in XHTML strict. Sono presenti alcune funzioni JavaScript all'interno del file *script.js* opportunamente collegato a tutte le pagine del sito.

1.2 Struttura dei database

I file XML e i relativi schemi XMLSchema sono salvati all'interno della cartella *data*.

Il file *plants.xml* contiene i dati e le specifiche che andranno a comporre le pagine dinamiche del sito. In particolare è formato da un elemento radice `<plants>` con infiniti figli `<plant>`, i quali hanno attributo `id` di tipo `positiveInteger`, unico e sempre crescente. L'elemento `<plant>` contiene:

date di tipo `date`;

author, title di tipo `string`;

img di tipo `anyURI`;

specs di tipo complesso, padre di altri elementi;

description di tipo `string`;

paragraph di tipo `string` con attributo `title`;

comments di tipo complesso, genitore di infiniti `<comment>`.

L'elemento `<specs>` ha come figli:

family, height, diameter di tipo `string`;

light, terrain di tipo enumerazione su base `string`;

temperature di tipo `integer`;

ph, hiquality di tipo vuoto con `minOccurs=0`.

Può sembrare poco corretto aver definito gli elementi `height` e `diameter` di tipo stringa e non intero positivo o float, ma ciò è stato fatto poichè in fase di popolamento ci si è accorti che l'unità di misura era variabile, nello specifico si passava dai centimetri ai metri. L'idea di trovare scritto *0.1 metri* o *1 metri* non ci piaceva, così abbiamo applicato la soluzione più semplice, ma non per questo migliore: dichiarare `height` e `diameter` di tipo stringa e

inserire l'unità di misura all'interno del database.

L'elemento `<comment>` ha attributo `valid` settato di default a `true` e contiene:

`date` di tipo `date`;

`time` di tipo `time`;

`user`, `text` di tipo `string`.

Il file *users.xml* contiene i dati degli utenti registrati al sito, in particolare: nickname, indirizzo e-mail e password. Un attributo booleano `admin` decide se l'utente possiede o no i privilegi di amministratore.

1.3 Le pagine che compongono *Rosa Rosae*

Il sito consta esattamente di sei pagine, alcune in doppia versione —utente e amministrazione— e altre esclusive all'uno o all'altro tipo d'utente. Esse sono:

Home page Pagina dinamica, disponibile in doppia versione, presenta delle piccole differenze che verranno esplicate in seguito.

Pagina dedicata Dinamica anch'essa, è la pagina comune a tutti gli elementi che compongono il sito, ovvero le piante. Ne descrive le caratteristiche e mostra i commenti degli utenti.

Contattaci Pagina statica, contiene le informazioni per contattare gli amministratori del sito.

Login Statica, contiene la form per effettuare l'accesso.

Registrati Statica, contiene la form per effettuare la registrazione.

Crea contenuto Dinamica, esclusiva agli amministratori, consente la creazione di nuovo contenuto ampliando il database.

2 Analisi dell'utenza

Data la natura del sito l'utenza è sicuramente un insieme molto ampio e diversificato di individui. A grandi linee l'età potrebbe essere compresa in un intervallo 20–80 anni dove, per questioni legate al digital divide, più l'età si alza più la densità di utenti dirada. Molto probabilmente, trattandosi di un sito sul giardinaggio, l'utente medio pretenderà che la presentazione delle pagine web sia di sufficiente gusto estetico. Si intuisce inoltre che le pagine dovranno essere abbastanza chiare ed esplicative per permetterne un facile utilizzo da parte di un pubblico meno giovane e quindi meno pratico nell'utilizzo dei servizi web.

3 Scendendo nel dettaglio

3.1 I punti d'incontro

Osservando la figura 1 si possono vedere alcuni elementi comuni a tutte le pagine del sito. In alto sono visibili il logo, il path, il menu di navigazione e il menu dedicato all'utente. Nel momento in cui un utente effettua l'accesso i link *Login* e *Registrati* scompaiono e al

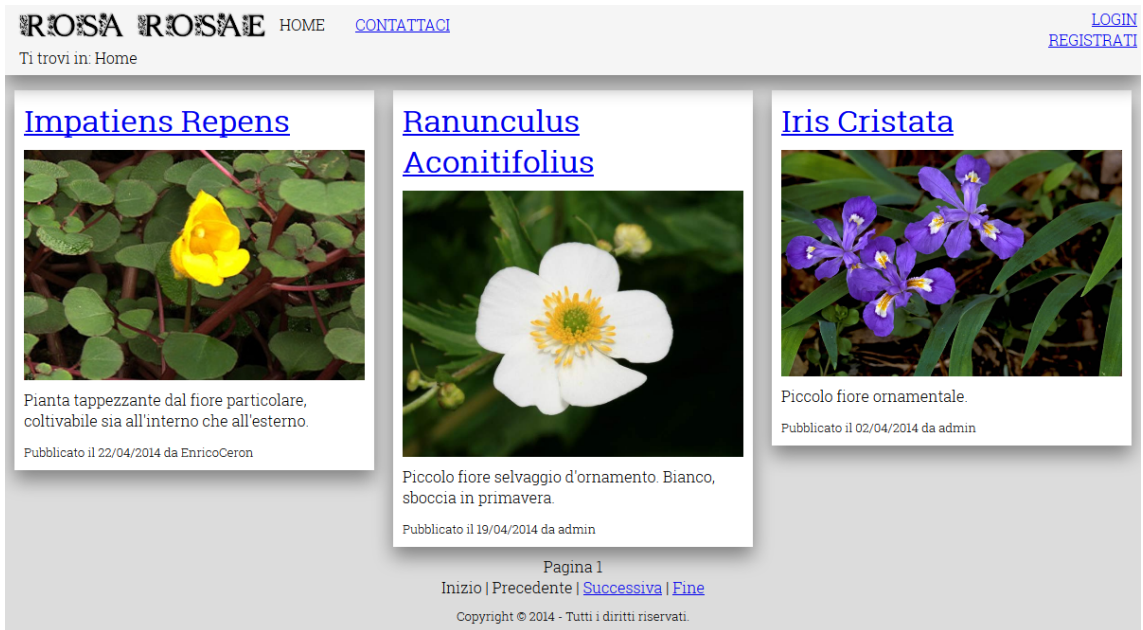


Figura 1: La home page.

loro posto vengono visualizzati il messaggio “Ciao [utente]” e *Logout*. Nel caso un amministratore effettui l’accesso, al di sotto di *Logout* compare un terzo link *Crea contenuto*. Inoltre, per rendere il tutto più esplicito, in coda al path viene aggiunta la stringa “(amministrazione)”.

Ora saranno analizzate singolarmente tutte le pagine che compongono il sito.

3.2 Home page

La home page prevede un layout di tipo ad album fotografico, à la Pinterest. Lo script perl che genera la pagina pesca dal database le ultime tre piante aggiunte, quindi con id compreso tra `last()` e `last()-3` all’interno di *plants.xml*. La mattonella di ogni pianta mostra le informazioni essenziali a capire quale sia il contenuto della pagina.

È possibile navigare tra le pagine della home tramite i link a fondo pagina. Ci si può muovere di pagina in pagina cliccando su *Precedente* e *Successiva*, oppure spostarsi direttamente alla prima o ultima pagina con *Inizio* e *Fine*.

3.3 La pagina dedicata

La pagina dedicata mostra nel dettaglio tutto ciò che interessa una singola pianta: il nome, un’ampia immagine, una tabella contenente le sue specifiche e una descrizione delle informazioni suddivisa in paragrafi molto schematici e concisi.

Al di sotto di quanto descritto sono visibili i commenti degli utenti ordinati in maniera decrescente secondo la data di pubblicazione, quindi dal più recente al più datato. Nel caso si sviluppi una conversazione tra due o più utenti che si scambiano messaggi in stretta correlazione l’uno con l’altro, con questo tipo d’ordinamento, la lettura risulterebbe molto scomoda e pedante. Ma data la natura e l’argomento del sito abbiamo supposto che ciò

Impatiens Repens ★

[Vai ai commenti](#)**Aspetto**

Strisciante con fusti che radicano al suolo.

Fiori

A 5 petali con grande sperone ricoperto di sottile peluria, giallo intenso, in estate.

Foglie

Sempreverdi, dentellate, reniformi, di colore verde.

Zona d'origine

Sri Lanka e India.

Coltivazione

In zone rigide come pianta da serra o appartamento, in terriccio umido e ben drenato. Se coltivato all'esterno richiede un'esposizione soleggiata. Ricopre il terreno in un bordo erbaceo o misto.

Propagazione

Per seme o per talea di stelo, in primavera o in estate.

Pubblicato il 22/04/2014 da EnricoCeron

Famiglia	balsaminaceae
Altezza	10 centimetri
Diametro	indefinito
Luce	soleggiato
Terreno	umido
Temp. min.	-10°C

Commenti

Esegui il [login](#) o [registrati](#) per commentare.

GiacomoBeltrame

lol

Pubblicato il 16/05/2014 alle 10:31:11

EnricoCeron

basta, non tocco più niente

Pubblicato il 16/05/2014 alle 09:55:32

Figura 2: La pagina dedicata.

si possa verificare con bassissime probabilità e quindi ignorato il problema. Se non fosse stato così sarebbe stato certamente utile proporre all'utente un'ordinamento inverso dei commenti o un sistema di citazione del commento a cui si fa riferimento.

La pagina è generata dinamicamente grazie allo script *showpage.cgi* che ricevendo il parametro *id* pesca le informazioni dal database e gli applica il foglio di stile *admin_plant.xsl* nel caso un amministratore abbia effettuato l'accesso, se così non fosse applica *plant.xsl*.

3.4 I contatti

È stata creata una pagina dove si possono trovare le informazioni per poter contattare gli amministratori del sito. Data la semplicità della pagina ci si aspetterebbe fosse del tutto statica ma durante lo sviluppo ci siamo accorti che nel momento in cui un utente effettua l'accesso, perché in alto a destra venga mostrato il messaggio “Ciao [utente]”, per uniformità nella navigazione del sito, la pagina doveva necessariamente essere generata da uno script perl ed essere quindi dinamica.

3.5 Le form

Di seguito sarà illustrato il funzionamento di tutte le form presenti nel sito, strumento fondamentale per l'interazione tra gli utenti. Qui non tratteremo degli script perl che raccolgono i dati delle form, il tutto sarà rimandato nell'apposita sezione 4.

3.5.1 Registrati e *logga!*

Come è facile supporre, le form di login e registrazione sono fondamentali per poter permettere agli utenti di lasciare la propria traccia. Non è possibile, infatti, commentare —o creare nuovi contenuti per utenti amministratori— senza essersi preventivamente registrati e autenticati.

Passiamo ora ad esaminare le due form:

Login Consta di quattro tag `<input>`, uno di tipo testo per il nickname, uno di tipo password per la password e due di tipo hidden funzionali all'invio di parametri che possono essere modificati o meno tramite la funzione JavaScript `CheckForJavaScript()` richiamata al caricamento del body della pagina *login.html*. Lo script *login.cgi* si comporterà di conseguenza a seconda dei valori dei parametri di tipo hidden che riceverà. Rimandiamo alla sezione 4.2 per la spiegazione degli script client-side.

Registrazione Non prevede tag `<input>` di tipo hidden bensì due tag di tipo testo per nickname ed e-mail e due di tipo password per la digitazione della stessa e il controllo che il primo inserimento sia corretto. In questo caso non è previsto l'utilizzo di JavaScript: eventuali errori sono segnalati tramite la stampa di una pagina generata dal server.

3.5.2 E ora come commento?

Come già osservato, raggiungendo le pagine dedicate ad ogni singola pianta, al di sotto della descrizione della stessa si possono leggere i commenti che gli utenti registrati al sito hanno lasciato in merito. Per poter commentare nasce la necessità di aggiungere una form da visualizzare in cima ai commenti con, però, alcune limitazioni per impedirne un uso improprio ed incontrollato.

Le situazioni nelle quali un utente che volesse commentare (amministratore o meno) può incorrere sono le seguenti:

- stringa di testo che invita a registrarsi o ad autenticarsi con relativi collegamenti alle pagine di registrazione e login nel caso di utenti non autenticati;
- form con textarea e bottone per l'invio del commento nel caso di utenti autenticati.

La form, inoltre, prevede una funzione JavaScript che permette all'utente di controllare in tempo reale quanti caratteri ha ancora a disposizione per commentare (massimo 150) e blocca la digitazione una volta raggiunto questo limite.

Nel caso in cui qualche malintenzionato disabilitasse l'uso del JavaScript sul proprio browser, il file *comment.cgi*, che si occupa dell'inserimento del commento, controlla in ogni caso che la condizione sia rispettata (viene impedito anche l'inserimento di un commento vuoto).

3.5.3 Crea contenuto

Questa form è stato deciso di generarla dinamicamente tramite lo script *addContent.cgi* in quanto si vuole evitare che utenti non autenticati o senza privilegi possano raggiungere tale form per questioni di sicurezza del database.

Si tratta di una form molto articolata ma non per questo complicata. Essa consta di vari tipi di campi input tra cui *text*, *file*, *checkbox* e di *select* e *textarea*.

4 Dietro le quinte!

4.1 Script server-side: *Perl*

Dopo aver analizzato in dettaglio le form presenti all'interno del sito, passiamo alla trattazione degli script perl, i moduli CGI, che si occupano della raccolta dei dati e analizzano la situazione nella quale si trova l'utente.

Questo è necessario per far in modo che la parte di presentazione e comportamento del sito si adattino in maniera consona ad ogni situazione in cui si trova l'utente e, di conseguenza, per renderlo il più possibile accessibile (di questo verrà trattato nell'apposito paragrafo 6 nel quale si discuterà, appunto, di accessibilità).

Ciò che accomuna quasi tutti gli script è il controllo delle sessioni che viene fatto sistematicamente in tutte le pagine dinamiche. Questo permette di mostrare in alto a destra sulla barra di navigazione i link alla pagina di login e registrazione per utenti ospiti oppure il proprio nickname ed il link per effettuare il logout per utenti autenticati. Non rientrano tra questi *login.cgi* che si occupa della creazione della sessione, *logout.cgi* che si limita a distruggerla e *register.cgi* che si occupa solo della registrazione di un nuovo utente.

Per quanto riguarda, invece, gli script che agiscono sui database modificandoli (apertura in scrittura) è stato scelto di adottare il blocco del file XML in questione (*plants.xml* o *users.xml*) onde evitare che la scrittura contemporanea di tali file da parte di due o più utenti invalidasse definitivamente l'intero sito per via della sua caratteristica dinamica.

Ora passiamo alla trattazione dettagliata di ogni script:

register.cgi Questo script raccoglie i sei parametri inviati dalla form in *register.html* ed effettua scrupolosamente su di essi tutti i controlli del caso dato che la registrazione viene effettuata una sola volta. Al momento è stato scelto di non dare la possibilità all'utente di modificare i propri dati.

Nel caso tutti i test vadano a buon fine viene creato il nuovo nodo utente e aggiunto al database *users.xml* con successivo reindirizzamento alla pagina dove ci si trovava prima di effettuare la registrazione (nel caso in cui JavaScript sia abilitato, altrimenti alla home page). Nel caso contrario lo script genera una pagina dove vengono elencati tutti gli errori commessi in fase di compilazione della form.

login.cgi Lo script raccoglie i quattro parametri della form, ne effettua i controlli (utente e password devono coincidere con quelle di un utente registrato nel database) e crea la sessione dell'utente con relativo cookie nel client se in *users.xml* è stata trovata una corrispondenza, altrimenti viene mostrata una schermata indicante il fatto che non è stata trovata una corrispondenza dei dati all'interno del database utenti.

Infine, nel caso di autenticazione riuscita, il modulo effettua un reindirizzamento alla pagina da dove si è voluto effettuare il login nel caso di JavaScript attivo, altrimenti il reindirizzamento viene fatto verso la home page.

index.cgi Si tratta di uno script volto all'interazione con il database per la creazione di pagine dinamiche. Una volta creato lo script esso è in grado di generare le pagine del sito che gli competono indipendentemente da cambiamenti nel database.

Lo script sfrutta la libreria *LibXSLT* che permette di applicare un determinato foglio di stile XSL ad un file XML (nel nostro caso *empty_plants.xml*) inizialmente vuoto e riempito temporaneamente coi nodi che si desidera visualizzare nel browser. Una serie di calcoli viene svolta per analizzare il contenuto del database ed organizzare la suddivisione delle pagine.

showpage.cgi Il comportamento e la funzione di questo script sono molto simili a quelli di *index.cgi* con la differenza che non è necessario nessun file XML di supporto in quanto esso si occupa di mostrare nel dettaglio la pianta selezionata e tutti i commenti ad essa associati (quindi un nodo e i suoi figli).

comment.cgi Utile al salvataggio dei commenti degli utenti, questo script può essere invocato soltanto da un utente preventivamente autenticato all'interno del sito.

Come detto in precedenza, esso controlla in ogni caso che il commento non superi i 150 caratteri o che non sia vuoto (condizioni facilmente evitabili disabilitando il controllo JS client-side). Se il commento non è valido genera una pagina di errore.

addContent.cgi In questo caso ci troviamo di fronte ad uno script il cui compito è solamente quello di stampare la form i cui dati verranno raccolti da *addPlant.cgi*. Rendendolo dinamico, oltre a permettere la visualizzazione del nome dell'utente (in uniformità alle altre pagine), possiamo rendere la pagina raggiungibile soltanto dagli amministratori dato che si possono inserire dati che vanno a modificare il database e quindi il contenuto del sito.

addPlant.cgi Per aumentarne il livello di sicurezza, il controllo dell'identità di amministratore dell'utente autenticato viene effettuata anche da questo script prima di modificare il database con i dati inseriti nella form *addContent.cgi*.

Cio di cui si occupa poi questo script è fondamentalmente di raccogliere i dati e manipolarli (ove necessario) per comporre mano a mano il nuovo nodo `<plant>` da

inserire in *plant.xml*. Al termine si viene reindirizzati alla pagina di visualizzazione del contenuto appena inserito.

logout.cgi Questo script non stampa niente a video ma si occupa soltanto di terminare la sessione dell'utente con successivo reindirizzamento alla home page.

about.cgi Genera una pagina con i dettagli verosimili degli sviluppatori e stampa nome dell'utente e link di logout in uniformità alle altre pagine del sito.

4.2 Script client-side: *JavaScript*

Nello sviluppo di questo sito ci siamo avvalsi anche dell'uso di alcuni script JS client-side. In particolare, la funzione `CheckForJavaScript()` si occupa di modificare gli hidden field della form di login in modo tale che i valori raccolti da *login.cgi* gli permettano il reindirizzamento alla pagina dalla quale l'utente ha deciso di effettuare l'accesso al sito (ad esempio per commentare un articolo). Qualora JavaScript non fosse abilitato l'utente verrebbe reindirizzato alla home.

La funzione `addtext()`, invece, inserisce nella textview della form di *addContent.cgi* la sintassi esatta per aggiungere i paragrafi descrittivi della pianta in questione in modo da rispettare perfettamente quella che è la struttura del database.

`CountChar()` è una funzione molto utile all'utente per aiutarlo nella scrittura del commento in quanto egli può sempre tenere d'occhio quanti caratteri ha a disposizione ed evitare che inserisca un commento che non venga accettato e doverlo riscrivere daccapo.

5 Mobile

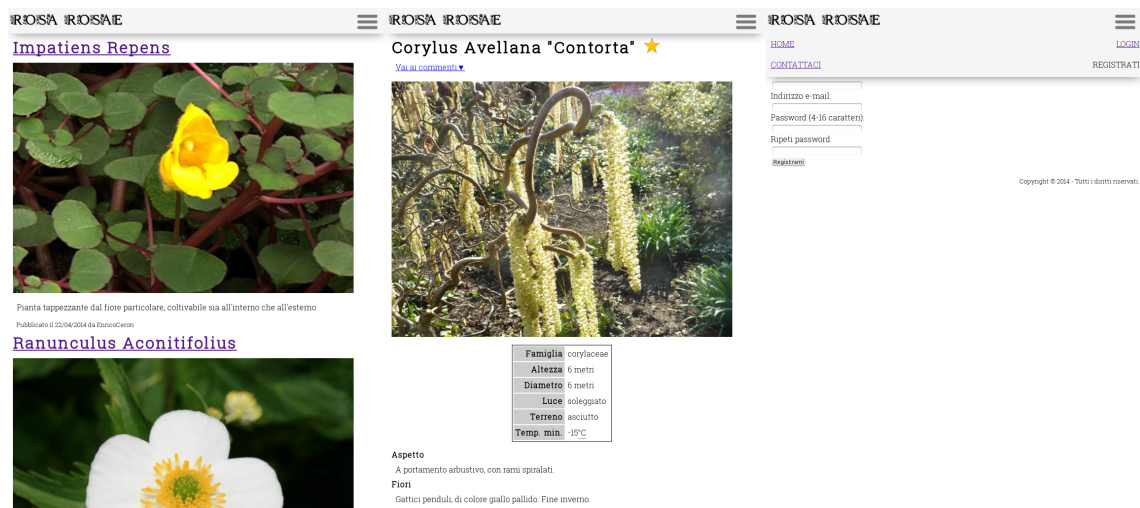


Figura 3: L'interfaccia mobile su una risoluzione di 768x1024.

Parecchia attenzione è stata rivolta allo sviluppo di una buona interfaccia mobile, rispettando i canoni "classici" di tale ambiente per offrire una buona immediatezza nell'utilizzo.

Le pagine sono minimali, veloci nel caricamento e facili da leggere. Il menu è a scomparsa, raggruppato in alto visualizzabile premendo un'icona. Unica nota negativa: è stato sacrificato il path poichè negli schermi più piccoli, andando a capo, diventava confusionario e poco comprensibile oltre ad occupare uno spazio eccessivo.

6 Accessibilità e testing

Accessibilità incorpora anche il concetto di usabilità. Sono state adoperate delle scelte tali che il sito potesse essere consultabile da utenti con disabilità nella vista: abbiamo svolto dei test per verificare che il contrasto fra i colori di testo e sfondi fosse sufficientemente buono utilizzando l'estensione *WCAG Contrast Checker* di Firefox con risultati soddisfacenti. È stato inserito un link nascosto all'inizio di ogni pagina che sposta l'attenzione direttamente al contenuto facendo, ad esempio, saltare la lettura dei menu da parte di uno screen reader. Le funzioni `JS CheckForJavaScript()` e `CountChar()` descritte nella sezione 4.2 aggiungono delle comodità che l'utente di certo non disprezza.

Tutte le pagine sono state validate utilizzando *Total Validator* ottenendo una valutazione WCAGv2 AAA.

In ambiente desktop Windows 7 è stato testato su: IE 11, Chrome 34+, Firefox 28+, Safari 5 e Opera 20 senza riscontrare problemi. In Linux Kubuntu su: Firefox 28+, Chromium e Lynx. In ambiente mobile Android su: AOSP e Chrome. In iOS su: Safari e Chrome. Sembra fili tutto liscio...

7 Conclusione

Al progetto abbiamo partecipato in maniera vicendevole. Giacomo Beltrame si è concentrato più nella realizzazione degli script perl e le form HTML che ad essi fanno riferimento, mentre Enrico Ceron si è focalizzato nella scrittura delle pagine statiche e dinamiche (XHTML e XSL) e creazione degli schemi del database principale e degli utenti. In ogni caso, entrambi abbiamo apportato modifiche al codice dell'altro e generato codice JavaScript per soddisfare esigenze di vario genere.