

UNIVERSITÀ DEGLI STUDI DI PADOVA

LAUREA IN INFORMATICA

CORSO DI TECNOLOGIE WEB

PROGETTO DI FINE CORSO

Monte Verde

Studente:
Luca ALESSIO

Matricola:
1070690

Studente:
Carlo SINDICO

Matricola:
1069322

LINK AL SITO:

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

MAIL DEL REFERENTE:

sindycarlo@gmail.com

LOGIN ADMIN:

USERNAME: *admin*
PASSWORD: *admin*

Indice

1 Abstract	1
2 Utenti destinatari	1
3 Struttura del sito	2
4 Amministrazione del sito	2
5 da qualche parte bisogner parlare bene anche di header e footer, sezione apposita?	2
6 Materiale consegnato DA FARE	2
7 HTML DA FARE	4
8 Perl DA FARE	4
9 CSS DA FARE	7
10 XML DA FARE	8
11 Javascript DA FARE	8
12 Spiegazioen archivio news? forse la includo in un altra sezione non so ancora	10
13 Accessibilità DA FARE MEGLIO STAVOLTA	11
13.1 Separazione tra struttura,presentazione e comportamento . . .	11
13.2 Colori	11
13.3 Tag meta	13
13.4 Screen reader	13
14 Usabilità DA FARE MEGLIO STAVOLTA	14

1 Abstract

- Il sito realizzato riguarda il fittizio Parco Naturale Monte Verde. L'obiettivo principale che il portale monteverde.it si pone è quello di fornire tutte le informazioni possibili sul parco agli utenti visitatori. Il sito è in lingua italiana.
- L'utenza che accede al sito ha la possibilità di visualizzare pagine informative che descrivono flora, fauna, storia e conformazione del parco e le attività che si possono svolgere al suo interno.
- Nel sito sono presenti anche pagine contenenti contenuto dinamico come gli orari e i prezzi d'ingresso al parco ed una pagina dedicata alle notizie.
- L'amministratore del sito ha la possibilità di alterare prezzi ed orari del parco e di creare nuove notizie. Una dettagliata spiegazione sulle funzionalità accessibili dall'amministratore presente nella sezione "Amministrazione del sito". Credenziali di accesso *Login:* admin *Password:* admin
- **Disclaimer:** Il Parco Naturale Monte Verde è un luogo puramente fantastico e senza alcuna corrispondenza nel mondo reale. La maggior parte dei testi contenuti nel nostro sito sono stati presi in prestito da <http://www.pnab.it/> perché, nonostante il Parco Naturale Monte Verde sia un luogo fittizio, ci è sembrato poco elegante riempire le pagine di "lorem ipsum" e allo stesso tempo la stesura di testo verosimile e originale avrebbe richiesto troppo tempo. Per quanto riguarda le immagini, esse sono state ricercate tramite Google Images, su richiesta del docente possiamo fornire la provenienza di ciascuna immagine (non includeremo tali dati in questa relazione in quanto superflui). Il logo è stato preso dal sito <https://greenmountainbaptist.org/>.

2 Utenti destinatari DA FARE

- Il target di riferimento di questo sito è molto ampio, un sito di cucina viene visitato solitamente da persone appartenenti a varie fasce d'età e con differenti bagagli culturali perciò è stato scelto di rendere il sito il più semplice ed intuitivo possibile, in particolare è stata posta molta

attenzione all'accessibilità per utenti non vedenti (ma questo verrà approfondito in una sezione apposita).

3 Struttura del sito

Il sito è diviso nelle seguenti aree primarie:

- **Home:** pagina di benvenuto, contiene una breve descrizione del parco e vari link alle sezioni principali del sito
- **Chi Siamo:** pagina puramente a scopo informativo, contiene un elenco degli avvenimenti importanti nella storia del parco, dall'inaugurazione al giorno d'oggi
- **Natura e Territorio:** altra pagina descrittiva, fornisce informazioni sulla conformazione territoriale del parco e su flora e fauna presenti in esso
- **News e Attività:** pagina divisa in due sezioni, un primo settore riguardante le attività che è possibile svolgere nel parco ed una seconda zona contenente le notizie più recenti
- **Orari e Prezzi:** come suggerisce il nome, da questa pagina l'utente può conoscere orari e prezzi d'ingresso al parco
- **Info e Contatti:** contiene informazioni quali il regolamento del parco, le istruzioni per raggiungerlo e contatti utili

Da ogni pagina è possibile raggiungere la mappa del sito dal link posto prima del footer.

4 Amministrazione del sito DA FARE

5 da qualche parte bisognerà parlare bene anche di header e footer, sezione apposita?

6 Materiale consegnato DA FARE

I file consegnati sono organizzati su quattro cartelle:

- cgi-bin: cartella nella quale sono presenti i file .cgi
- data: in questa cartella sono contenuti i file xml ed i relativi XMLSchema.
- public-html: contiene index.html e le sotto-cartelle:
 - css: cartella contenente i file .css;
 - images: cartella contenente tutte le foto del sito;
 - js: cartella contenente i vari script realizzati in JavaScript.
- relazione: cartella contenente la presente relazione in formato pdf e le immagini in essa contenute a dimensioni originali

Andiamo ora ad esaminare nel dettaglio le applicazioni di ciascuna tecnologia impiegata nel sito e le loro interazioni.

7 HTML DA FARE

La struttura HTML del sito viene interamente "stampata" da vari file .cgi: questa scelta è dovuta alla grande presenza di contenuto dinamico all'interno del sito per cui non sono presenti file .html "puri" se non index.html che è però un semplice redirect a menu.cgi, file Perl che effettivamente stampa la homepage.

Le pagine HTML del sito (stampate dal cgi) sono state realizzate rispettando lo standard XHTML 1.0 Strict.

8 Perl DA FARE

Le pagine scritte in Perl si dividono principalmente in due tipologie: pagine "dinamiche" di rappresentazione e pagine di elaborazione dei dati.

Alla prima tipologia appartengono i file .cgi che eseguono il "print" della pagina html con il contenuto richiesto (ne è un esempio la pagina page_template.cgi che si occupa della stampa a video di ogni ricetta) mentre le pagine della seconda tipologia sono solitamente "pagine di servizio" ovvero codice che esegue operazioni dietro le quinte come salvataggi ed eliminazioni di dati sui file xml.

Nel dettaglio, questo è ciò di cui ciascun file si occupa:

- menu.cgi - pagina principale del sito, contiene del contenuto dinamico in quanto le "ricette consigliate" che compaiono nella pagina sono casuali e variano ad ogni accesso alla pagina (inoltre in ogni pagina del sito il footer varia leggermente a seconda se l'amministratore sia loggato oppure meno).
- antipasti.cgi, secondi.cgi, primo.cgi, dessert.cgi - semplici sottomenù interni che si occupano di raggruppare le ricette appartenenti alla stessa categoria per facilitare la navigazione dell'utente
- page-template.cgi - pagina dinamica che visualizza la ricetta richiesta, funziona nel seguente modo: il link cliccato dall'utente che lo indirizza a questa pagina contiene un parametro (id) differente per ogni ricetta, page_template.cgi cerca il parametro passato tra gli attributi

IDCode delle singole ricette nel file 4forchette.xml (vedi sezione XML) ed identifica quale era la ricetta richiesta di cui va poi a restituirne le informazioni su schermo.

- proponericetta.cgi - stampa la pagina "Proponi una ricetta" attraverso la quale l'utente ha la possibilità di proporre nuove ricette da inserire nel sito. La pagina contiene un form di discrete dimensioni e delle istruzioni sul suo completamento. Un'analisi più dettagliata su come avviene il processo di creazione, memorizzazione e approvazione delle ricette verrà effettuata in un successivo paragrafo.
- handle_proposta.cgi - quando l'utente ha compilato correttamente (vedi sezione Javascript) il form per la proposta di una nuova ricetta, la ricetta viene salvata nel "database" xml. La nuova ricetta non verrà tuttavia ancora visualizzata nel sito, prima dovrà infatti venire approvata dall'amministratore (vedi accept_ricetta.cgi).
- contatti.cgi - stampa la pagina dei contatti, qui l'utente ha la possibilità di vedere i messaggi lasciati da altri utenti ed eventualmente può lasciarne uno proprio compilando il form apposito dove viene richiesto un nome ed il testo effettivo del commento (sarebbe stato interessante implementare un sistema di registrazione e gestione degli utenti nel sito ma date le scarse dimensioni del gruppo e il tempo disponibile si è optato per una soluzione meno impegnativa).
- inserisci_commento.cgi - gestisce l'inserimento di un nuovo commento traducendo l'input dell'utente in dati xml.
- amministratore_login.cgi - contiene il form che l'amministratore utilizza per effettuare l'accesso all'area riservata. Questa pagina è accessibile solamente dal link posto nel footer della pagina principale (menu.cgi).
- controllo_login.cgi - controlla la correttezza delle credenziali inserite nel form del punto precedente. Al momento è previsto un solo amministratore per il sistema. L'amministratore viene identificato come autenticato quando un parametro apposito (\$auth) è impostato ad un preciso valore e ciò viene settato in degli appositi cookie (vedi codice del file per i dettagli) per rendere l'informazione nota tra le diverse pagine.

- console_admin.cgi - stampa l'hub amministrativo da cui è possibile gestire semplicemente l'intero sito. Da questa pagina l'amministratore ha la possibilità di accettare o rifiutare le nuove ricette, eliminare vecchie ricette ed eliminare eventuali commenti ritenuti offensivi.
- accept_ricetta.cgi - concede ad una ricetta il permesso di venire visualizzata nel sito (semplicemente il suo attributo accepted viene settato positivamente).
- delete_ricetta.cgi - elimina una ricetta dal file xml. Operazione irreversibile.
- delete_commento.cgi - elimina un commento dal file xml. Operazione irreversibile.
- logout.cgi - breve script che effettua il logout dell'amministratore. Viene invocato alla pressione del tasto "logout" visibile sul footer di ogni pagina qualora l'amministratore sia attualmente autenticato nel sistema.
- cercaricetta.cgi - nell'header di ogni pagina presente una barra di ricerca, qualora l'utente digitasse una stringa di caratteri e ne richiedesse l'elaborazione, questo è lo script che verrebbe eseguito: cercaricetta.cgi stampa una pagina contenente un elenco di tutte le ricette il cui titolo matcha in parte o totalmente il parametro di ricerca inserito dall'utente. La ricerca avviene semplicemente scorrendo l'xml isolandone nomi ed id di tutte le ricette e restituendo solamente quelle che contengono una corrispondenza.
- funzioni.pl contiene funzioni di servizio usate in diversi contesti (ad es. trim degli input).

9 CSS DA FARE

- Nella realizzazione dell'interfaccia grafica del sito stato usato lo standard CSS3.
- Allo stesso tempo si è fatta molta attenzione alla compatibilità con browser più datati, e si è cercato di utilizzare un numero ristretto delle nuove funzionalità offerte dallo standard.
- Alcune delle funzionalità CSS3 che sono state utilizzate: Border-radius e Box shadow: per realizzare i pulsanti delle form , e per le immagini.
- Nella cartella public-html/css sono presenti i seguenti fogli di stile:
 - style.css: modella lo stile di visualizzazione del sito sia per gli utenti desktop (che hanno uno schermo largo al massimo 1200px) che per gli utenti mobile con dispositivi con schermo piccolo (che hanno uno schermo largo massimo 390px) e dispositivi con dimensioni schermo intermedie (max-width=730px).
 - print.css: modella lo stile di stampa delle pagine del sito.(particolare attenzione si è data alla stampa delle pagine delle ricette).

10 XML DA FARE

Sono presenti 3 file xml principali con i rispettivi xsd:

- amministratore.xml - semplice file ausiliario per il controllo del corretto login nell'area amministrativa, contiene solamente due campi per il login e la password.
- commenti.xml - contiene i commenti contenuti nella sezione contatti; dalla console amministrativa possibile intervenire indirettamente sul file attuando operazioni di eliminazione
- 4forchette.xml - cuore centrale del sistema, questo documento xml svolge la funzione di database per le ricette: anche qui l'amministratore pu effettuare operazioni di eliminazione mentre l'inserimento riservato all'utente (questo aspetto è esaminato nel dettaglio nelle sezioni successive)

11 Javascript DA FARE

- Javascript é stato utilizzato principalmente per il controllo degli input dei form.
- Nel sito sono presenti 3 differenti form, quello per il login nell'area amministrativa, un secondo per il submit dei commenti nella pagina dei contatti ed infine il più complesso nella sezione per la proposta di nuove ricette.
- Per quanto riguarda i primi due form (gestiti rispettivamente dai file controllo_login.js e valida_commento.js), una volta che l'utente clicca su "Submit" vengono analizzati in ordine di apparizione tutti i campi del form, se ne viene identificato uno vuoto allora l'invio viene bloccato e l'utente ne viene notificato da un messaggio d'errore specifico per quel campo.
- Per quanto riguarda il form "Proponi ricetta" invece (gestito dal file proponi_ricetta.js), oltre a questi controlli basilari ne viene effettuato un altro più approfondito nell'area di inserimento degli ingredienti.

- Dato che ogni ricetta presenta un numero di ingredienti sempre differente era impossibile prevedere un numero definito di campi per ciascun ingrediente nel form (in quanto sarebbero stati spesso insufficienti oppure troppo numerosi e comunque sgradevoli visivamente), per risolvere questo problema si è inizialmente pensato di realizzare un form dinamico che aggiungesse campi su richiesta dell’utente. Questa soluzione è stata per scartata per difficoltà tecniche nella realizzazione e principalmente per il fatto che non fosse accessibile agli screen reader.
- Si è optato quindi per lasciare la zona di inserimento degli ingredienti come una semplice area di testo all’interno della quale però vengono applicate delle regole per l’identificazione dei singoli ingredienti: dopo ogni ingrediente è necessario inserire il carattere di separazione ";" (punto e virgola senza virgolette) ed andare a capo. Se il testo inserito dall’utente rispetta questa sintassi l’input verrà accettato, altrimenti verrà fornito un messaggio d’errore. Nella pagina sono presenti delle istruzioni sull’utilizzo del form che spiegano anche l’appropriato inserimento degli ingredienti; riteniamo che la maggior parte degli utenti non trovi difficoltà ad usare questa sintassi che è inoltre pienamente accessibile ed utilizzabile da utenti non vedenti.

Come interagiscono le varie tecnologie tra di loro? Vediamo un esempio...

12 Spiegazioen archivio news? forse la includo in un'altra sezione non so ancora

13 Accessibilità DA FARE MEGLIO STAVOLTA

13.1 Separazione tra struttura,presentazione e comportamento

- Per una maggiore accessibilitá al sito da parte di utenti disabili e per favorire gli algoritmi dei motori di ricerca si é deciso di separarare la struttura dalla presentazione e dal comportamento. Infatti il contenuto del sito é rappresentato dai file HTML e CGI, i quali richiamano i fogli di stile CSS e si utilizzano (anche in questo caso attraverso percorsi esterni), controlli in JavaScript in particolare per la compilazione dei form.
- Il contenuto rimane accessibile anche se JavaScript é disabilitato. Infatti opportuni controlli in Perl ne verificano la validitá.
- Tutto il codice é stato scritto seguendo le disposizioni W3C con opportuna validazione sui loro validatori.

13.2 Colori

- Si é scelto uno schema di colori non particolarmente vivace (un mix di azzurri chiaro); anche se non sono colori di base comunque la lettura dei testi risulta accessibile. Per un sito di cucina inoltre opportuno non utilizzare colori troppi aggressivi e decisi ma attenersi ad uno stile pi sobrio e rilassato.
- I link sono sempre sottolineati, e diventano di colore viola quando vengono cliccati.

Di seguito sono riportate le visualizzazioni del sito attraverso alcuni disturbi visivi:



Figure 1: vedi file deuteranopia.png



Figure 2: vedi file protanopia.png



Figure 3: vedi file tritanopia.png

13.3 Tag meta

Sono stati inseriti per ogni pagina i tag meta:

- title: descrive la pagina corrente dal particolare al generale.
- description: da una descrizione del contenuto del sito
- keywords: contiene parole chiave per i motori di ricerca
- language: indica che il sito è stato interamente scritto in italiano.
- author: indica l'autore/i del sito
- content-type: contiene direttive per il browser
- viewport: esprime indicazioni per la visualizzazione

13.4 Screen reader

- Ogni foto ha il suo attributo alt che descrive ciò che l'immagine ritrae. Si è evitato di utilizzare immagini per visualizzare il testo, quindi il contenuto informativo rimane accessibile anche quando fallisce il caricamento delle immagini o del CSS.
- Si è fornita particolare attenzione alle parole straniere che sono state segnalate agli screenreader attraverso il tag "span xml:lang=en" segnalando la lingua con cui leggere correttamente i vocaboli.
- Inoltre, nelle pagine contenenti form, è stato inserito un link skip nav per saltare direttamente al contenuto qualora l'utente dello screen reader non voglia riascoltare nuovamente il menù

14 Usabilità DA FARE MEGLIO STAVOLTA

L'utente riesce ad orientarsi nel sito? Analizziamo alcuni degli assi principali del giornalismo. Ometteremo gli assi Who e When dato che nel nostro contesto non è necessario dare all'utente informazioni su chi c'è dietro al sito (è un progetto didattico) né sono presenti novit continue.

- What?: Un utente appena entra nella home capisce subito che si tratta di un sito di ricette, dalla barra dei menù (Proponi ricetta, Cerca ricetta), e dal contenuto in primo piano che mette in evidenza alcune ricette proposte.
- Where?: L'utente riesce sempre a capire dove si trova grazie al breadcrumb, inoltre l'header costituisce un importante punto di riferimento per la navigazione dell'utente.
- Why?: Perché un utente dovrebbe rimanere nel sito o dovrebbe ritornarci? Il sito è principalmente espositivo,(gli utenti possono liberamente visualizzare le ricette), si è cercato di renderlo più interessante, aggiungendo una sezione proponi ricetta (l'utente ha la possibilità di inserire la propria ricetta), e anche una sezione commenti.
- How?: La barra di navigazione mostra tutte le sezioni principali del sito alle quali un utente può accedere. Nella barra menù è sempre evidenziata la voce della pagina in cui ci troviamo,e si vede attraverso una diversa colorazione dei link in quali altre pagine si è stati.