

What To Visit

Graziano Grespan 1003760

Carlo Munarini 1050128

Federica Speggiorin 1051031

Sebastiano Valle 1050123

4 febbraio 2015

Indirizzo: <http://tecnologie-web.studenti.math.unipd.it/tecweb/~svalle/homepage.html>
Email referente: sebastiano.valle@hotmail.it

Indice

1 Abstract	3
1.1 Suddivisione ruoli	3
1.2 Schema organizzativo	3
2 Individuazione degli utenti e delle loro esigenze	4
3 Struttura	5
3.1 Struttura organizzativa	5
3.2 Progettazione della struttura delle pagine	5
3.2.1 XHTML and Semantic Markup	5
3.2.2 Tighter, Firmer Pages Guaranteed: Structure and Semantics	6
3.3 Parti comuni a tutte le pagine	6
3.3.1 Head	7
3.3.2 Body	7
3.4 Pagine di primo livello	7
3.5 Pagine di secondo livello	8
3.5.1 Pagine secondarie	8
3.5.2 Categorie di località	8
3.6 Pagine di terzo livello	8
3.7 Pagina di ricerca fallita	9
4 Front-end	10
4.1 Gestione cookie	10
4.2 Inserimento placeholder in barra di ricerca	10
4.3 Visualizzazione dei commenti (lato client)	10
4.4 Geolocalizzazione	10
4.4.1 L'implementazione del modulo in dettaglio	11
5 Back-end	13
5.1 Inserimento commenti	13
5.2 Barra di ricerca	13
6 Test effettuati	14
6.1 Validazione XHTML	14
6.2 Validazione CSS	14
6.3 Validazione XML e XSD	14
6.4 Validazione XSLT	14
6.5 JavaScript	14
6.6 Perl e CGI	14
6.7 Cynthia Says	14
6.8 WAVE	15
6.9 Fangs	15

Elenco delle figure

1 Organizzazione gerarchica del sito	5
--	---

Elenco delle tabelle

1 Abstract

Il progetto consiste nella realizzazione di un sito Web che presenti ai visitatori alcune località turistiche suddivise tra località marittime, località montane e località cittadine. Gli utenti hanno anche la possibilità di esprimere le proprie opinioni sulle località esposte aggiungendo dei commenti.

1.1 Suddivisione ruoli

Durante il progetto le attività sono state ripartite nel seguente modo:

- **Struttura:** Sebastiano Valle, Federica Speggiorin, Graziano Grespan e Carlo Munarini
- **Presentazione:** Carlo Munarini, ed in misura minore gli altri componenti
- **Front-end:** Graziano Grespan, Sebastiano Valle, Federica Speggiorin ed in misura minore Carlo Munarini
- **Back-end:** Sebastiano Valle
- **Accessibilità:** Sebastiano Valle, Federica Speggiorin
- **Validazione e testing:** Tutti i componenti
- **Relazione:** Tutti i componenti

1.2 Schema organizzativo

Lo schema organizzativo adottato è di tipo ambiguo: sebbene qualche località potrebbe appartenere a più categorie (ad esempio sia città che mare), è stato deciso di associare ad ogni località un'unica categoria per la sua caratteristica di spicco. Questa scelta è dovuta al fatto che sono previste tre modalità di interazione con il sito:

1. l'utente sa già quale località cerca e con la barra di ricerca può direttamente trovare ciò che gli interessa (tiro perfetto nella metafora della pesca), altrimenti nel caso peggiore ha due categorie da esplorare se non trova subito ciò che cerca in una categoria;
2. l'utente ha un'idea precisa del tipo di vacanza che ricerca, ma si aspetta di aumentare le proprie conoscenze riguardo a delle mete di mare/città/montagna durante l'esplorazione del sito (trappola per aragoste nella metafora della pesca);
3. l'utente non sa ciò che cerca ma ha solamente un'idea vaga di ciò che gli interessa; in questo caso, permettendogli di scegliere subito ciò che gli interessa maggiormente, si può far avvertire nell'utente una sensazione di serendipità nell'esplorare nuove località di cui non conosceva nemmeno l'esistenza.

2 Individuazione degli utenti e delle loro esigenze

Il sito non si pone dei vincoli al target di utenza mirato; in particolare si individuano due macro-categorie di utenti di lingua italiana:

- L'utente che vuole fare una vacanza e desidera avere più informazioni su questa, conoscendo già la meta desiderata;
- L'utente che è alla ricerca di informazioni generiche su una località o di una vacanza senza conoscerne la meta, navigando senza un obiettivo preciso all'interno del sito.

Si è comunque cercato di rendere accessibile le pagine del sito in modo tale che questo potesse degradare elegantemente in caso di browser vecchi o testuali e che il sito fornisse supporti a persone svantaggiate sotto il profilo fisico o psichico. Per ogni località sono state individuati dei punti di interesse che sono ritenuti di notevole attrattiva per il target di utenti scelto, cercando di includere sia un pubblico giovane che uno più adulto. Tuttavia link esterni che portano a pagine non in lingua italiana sono stati affiancati da un'indicazione testuale della lingua in cui è scritta la pagina riferita. La data, tuttavia, è in formato big-endian (americano AAAA-MM-GG) perchè l'utenza a cui il sito si rivolge è più che in grado di apprendere facilmente questo formato, se non è addirittura già conosciuto (a differenza di una possibile soluzione come middle-endian, ovvero AAAA-GG-MM che potrebbe creare molta confusione nelle categorie interessate).

3 Struttura

Di seguito sono illustrate le modalità di progettazione della struttura delle pagine all'interno del sito.

3.1 Struttura organizzativa

Il sito è stato progettato secondo la seguente struttura gerarchica (non sono state indicate tutte le località, anche perché è la direzione in cui il sito potrebbe più facilmente estendersi): Per ogni

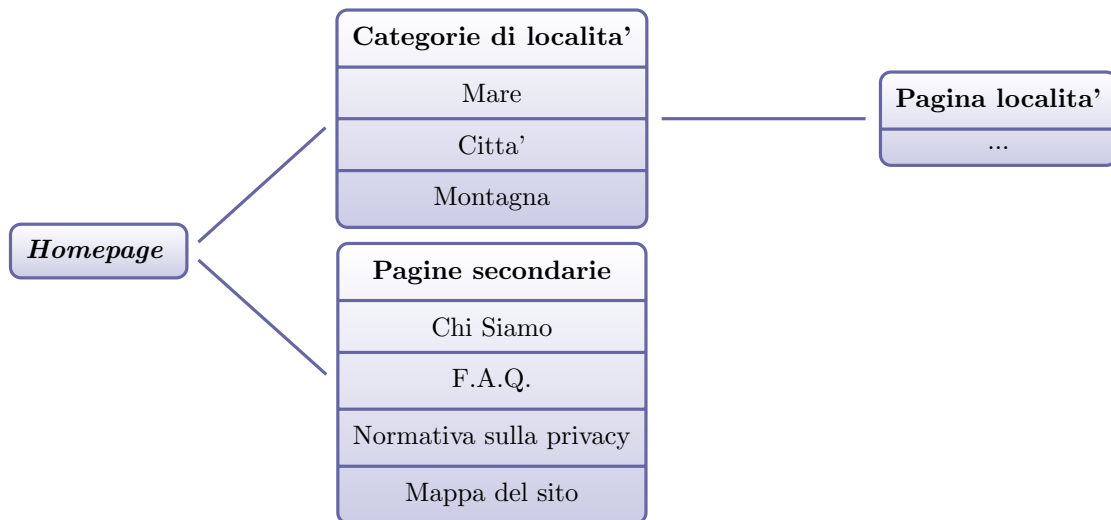


Figura 1: Organizzazione gerarchica del sito

pagina allo stesso livello gerarchico (ovvero sulla stessa fascia verticale in figura) è stata uniformata la struttura della pagina, così da permettere all'utente di orientarsi meglio all'interno del sito.

È opportuno osservare che solo dei sottoinsiemi di località mutuamente esclusivi sono figli di ogni categoria di località; la figura non rappresenta chiaramente questa distinzione per non essere di difficile comprensione.

3.2 Progettazione della struttura delle pagine

Durante le attività di progetto, le pagine del sito sono state codificate facendo riferimento ai capitoli 6 (*"XHTML and Semantic Markup"*) e 8 (*"Tighter, Firmer Pages Guaranteed: Structure and Semantics"*) di *Designing with Web Standards*¹.

3.2.1 XHTML and Semantic Markup

Sono state seguite le seguenti indicazioni:

- ogni documento XHTML inizia con un DOCTYPE corretto e facente riferimento a XHTML 1.0 Strict (come previsto dalle specifiche del progetto);
- ogni dichiarazione DOCTYPE è seguita dal namespace XHTML, con l'indicazione che nelle pagine verrà utilizzata la lingua italiana (se non diversamente specificato per alcune parti di testo, in cui sarà indicato con uno span apposito);
- per la codifica dei caratteri è stata utilizzata UTF-8, in quanto potevano essere presenti anche caratteri stranieri non presenti nel comune ISO Latin 1 oltre alle lettere accentate;

¹Jeffrey Zeldman, Ethan Marcotte - *Designing with Web Standards*, Terza Edizione [2009]

- tutti gli elementi e gli attributi sono stati scritti in *lowercase*, siccome XHTML è case sensitive, mentre il loro contenuto può avere indistintamente lettere maiuscole o minuscole;
- tutti i valori degli attributi hanno un valore (possibilmente vuoto) e sono racchiusi tra virgolette e tutti i caratteri speciali sono stati generati utilizzando le rispettive entità e non venendo inseriti direttamente;
- tutti i tag sono aperti e chiusi nell'ordine giusto, anche quelli vuoti;
- nessun commento ha due *dash* consecutivi al suo interno;
- gli elementi sono stati inseriti all'interno della pagina in una struttura tale che questa fosse appropriata per il contenuto che gli elementi rappresentano e non per come verranno visualizzati (a tal fine non sono stati utilizzati elementi o attributi presentazionali o di comportamento ma solamente semantici).

3.2.2 Tighter, Firmer Pages Guaranteed: Structure and Semantics

Sono state seguite le seguenti indicazioni:

- i tag `div` sono stati intesi come divisioni di elementi aventi un comune scopo all'interno del documento e non come blocchi da comporre affinché la pagina sia “bella da vedere”;
- si è cercato di scegliere nomi semanticamente appropriati per `id` e `class`, evitando di attribuire ad un `id` il valore “giallo” o il valore “pippo”;
- gli attributi `id` e `class` sono stati usati per il motivo per cui sono stati creati: gli `id` individuano un elemento che svolge un ruolo semanticamente unico nella pagina, mentre una classe attribuisce ad un elemento delle caratteristiche che condivide con altri elementi;
- per la “*Sticky Note Theory*”, ogni elemento che viene marcato con un `id` è preferibilmente soggetto a essere coinvolto in regole CSS o ad essere cercato in qualche script;
- dal momento che gli `id` sono molto potenti per creare ancori nel documento, effettuare query su CSS e JavaScript, si è cercato di usare un `id` anziché una classe ogniqualvolta possibile (usando identificatori ben-formati);
- le pagine del sito sono state strutturate in modo tale che i contenuti siano facilmente comprensibili logicamente sia per utenti svantaggiati che utilizzano screen reader, così come per i motori di ricerca, con titoli ordinati con importanza progressiva e altri aiuti che verranno approfonditi nella sezione “Accessibilità”;
- le pagine contengono elementi che sono riusabili in altre pagine, così da:
 1. non dover scrivere una generica struttura per ogni pagina prodotta;
 2. evitare disorientamento all'utente.
- proprio perchè vengono preferiti gli `id` agli attributi `class`, si è cercato di evitare il fenomeno chiamato “*classitis*” da Jeffrey Zeldman, ovvero quando troppi elementi facilmente raggruppabili come figli di un unico elemento condividono la stessa classe, basta applicare un `id` all'elemento padre anziché applicare la singola classe a tutti gli elementi figli;
- si è cercato di evitare anche il fenomeno chiamato “*divitis*”, ovvero una tendenza ad inserire tag `div` e `span` dovunque, anche dove non vi sono elementi in relazione tra loro o, più genericamente, dove non è strettamente necessario.

3.3 Parti comuni a tutte le pagine

Tutte le pagine sono state scritte seguendo lo standard *XHTML 1.0 Strict* e come codifica è stata scelta UTF-8 dal momento che nel sito sono presenti parole accentate. Una sezione **head** ed una sezione **body** sono state inserite in tutte le pagine con la stessa struttura.

3.3.1 Head

Sono presenti i seguenti tag nelle sezioni head di tutte le pagine:

- **title:** permette di visualizzare sulla finestra del browser il titolo della pagina visualizzata, dal particolare al generale
- **meta title:** indica il titolo della pagina in un eventuale snippet, anch'esso dal particolare al generale
- **meta description:** in questo tag viene inserita la breve descrizione della pagina visualizzata in un eventuale snippet
- **meta author:** in questo tag sono indicati i componenti del gruppo
- **meta keywords:** parole che aiutano un motore di ricerca a trovare la pagina grazie a dei termini di importanza focale
- **meta robots:** tag che indica ad un eventuale spider se indicizzare la pagina e se seguire i link da essa uscenti
- **meta reply-to:** indica l'indirizzo di posta elettronica dell'autore del documento
- **meta Classification:** tag che serve ad indicare l'argomento trattato dalle pagine del sito
- **meta viewport:** elemento orientato all'ottimizzazione del sito per dispositivi multipli, indicando al browser come controllare dimensioni e scala della pagina
- **link shortcut icon:** icona visibile a fianco al titolo della scheda nel browser, aiuta a identificare meglio le schede di *What To Visit* se un utente avesse più schede aperte nel suo browser
- **link stylesheet:** collegamento ai vari fogli di stile CSS, questo tag è stato utilizzato più volte in quanto abbiamo voluto consentire una diversa visualizzazione del sito in base al dispositivo utilizzato dall'utente.

3.3.2 Body

Affinchè l'utente si sentisse il meno disorientato possibile all'interno di *What To Visit*, si è cercato di progettare il sito con un layout essenziale e che mettesse in primo piano il contenuto aspettato in tutte le pagine. Sono presenti questi elementi strutturali nei corpi di tutte le pagine:

- un header, dove vi è il logo del sito;
- un'ampia parte centrale, dove vengono visualizzati i contenuti richiesti dall'utente;
- un footer, dove sono presenti link ed informazioni di poco rilievo e un'indicazione riguardo la validità della pagina.

3.4 Pagine di primo livello

Vi è un'unica pagina di primo livello, ovvero la homepage. Essendo la pagina principale del sito, si è pensato di esporre in primo piano all'utente la scelta delle tre categorie delle località. A partire da queste l'utente può arrivare nelle pagine delle categorie, dove può trovare le liste delle località presenti in queste.

Dal momento che la homepage è l'unica pagina facilmente riconoscibile data la sua struttura con tre titoli di indirizzamento, la breadcrumb è stata omessa perchè si è assunto che gli utenti riuscissero a dedurre che si trovano nella homepage quando vi sono dentro (anche grazie all'URL). Per poter comunque fornire collegamenti alle pagine che non sono di contenuto ma che sono significative (Chi Siamo e F.A.Q.), i link a queste sono stati inseriti nell'header della pagina a fianco del logo; in questo modo, anche se sono di importanza secondaria rispetto ai tre pannelli visualizzati nella pagina, rimangono comunque nella parte visibile del sito quando questo viene

aperto (http://en.wikipedia.org/wiki/Above_the_fold#In_web_design). Oltre a questi link, vi è anche una barra di ricerca dalla quale, digitando il nome della località e confermando la propria richiesta, si viene reindirizzati direttamente nella pagina della località cercata. Nel footer, oltre alle indicazioni di validità della pagina, sono stati lasciati i link restanti alle pagine che non sono di contenuto.

3.5 Pagine di secondo livello

Vi sono due tipi di pagine di secondo livello: quelle secondarie (che forniscono all'utente informazioni utili ma non fondamentali per la navigazione) e quelle che raggruppano le località per categorie in una lista.

Queste pagine presentano una breadcrumb sotto l'header, così che l'utente possa orientarsi semplicemente guardando questa. La barra di ricerca è stata spostata nella breadcrumb, ma siccome non vi sono altri elementi nell'angolo in alto a destra, questo fattore si ritiene che non crei disorientamento nell'utente.

La parte centrale della pagine è divisa a sua volta in due parti:

- un blocco di navigazione, dove è evidenziata la pagina in cui ci si trova e dove si vedono i “fratelli” della pagina in questione, ovvero le pagine facenti parte della stessa famiglia (come si vede in fig. 1 a pagina 5);
- una parte centrale, dove viene visualizzato il contenuto atteso della pagina se questa è secondaria, altrimenti la lista della località presenti in tale categoria.

3.5.1 Pagine secondarie

Il footer di queste pagine non presenta altre informazioni oltre alle indicazioni alla validazione. Questo poichè tutti i collegamenti alle altre pagine secondarie sono già presenti nel blocco di navigazione.

In particolare sono presenti quattro pagine secondarie all'interno del sito:

- Chi siamo: spiega brevemente l'obiettivo del sito *What To Visit*
- Faq: il cui contenuto è costituito da una lista di definizioni. Essa presenta una serie di domande frequenti sul sito e le relative risposte.
- Normativa sulla privacy: redatta sulla base delle reali informazioni che il sito raccoglie; è stata creata prendendo anche spunto dal sito www.freeprivacypolicy.com
- Mappa del sito: il suo contenuto è costituito da una lista non ordinata di tutti i link presenti all'interno del sito. Essi sono presentati in ordine di “importanza”: homepage, link alle tre categorie con relativa sottolista contenente le singole località, ed infine link alle pagine secondarie.

3.5.2 Categorie di località

In queste pagine, il blocco navigazione contiene anche una parte dove si può vedere in una mappa dove si trovano le località *solamente* per quella categoria di località; questa parte verrà approfondita nella sezione “Geolocalizzazione”.

Il footer di queste pagine contiene tutti i link alle pagine secondarie.

3.6 Pagine di terzo livello

Nelle pagine di terzo livello sono descritte le località. Sono stati pensati due modi per arrivare in queste pagine: tramite la searchbar, inserendo il nome della località in italiano o in lingua originale, oppure tramite il link presente sulla pagina della categoria alla quale appartiene alla località (in realtà attualmente vi sarebbe un terzo modo, ovvero dalla mappa del sito; tuttavia, i progettisti hanno pensato che se il sito si dovesse espandere, nella mappa del sito rimarrebbero solamente homepage, categorie e pagine secondarie per evitare disorientamento).

La parte alta della pagina (header e breadcrumb) è sostanzialmente uguale a quella delle pagine di secondo livello (sezione 3.5), solamente che vi è un livello di profondità in più.

La parte centrale è divisa in tre parti:

- un blocco di navigazione, dove è evidenziata la categoria alla quale la pagina fa parte e sono elencate le categorie delle località;
- una parte centrale, dove viene presentata la località, informazioni utili per questa (ad esempio potrebbero essere presenti i trasporti) e le maggiori attrattive di questa località; sotto la descrizione della località, gli utenti possono lasciare un commento con la loro opinione sulla località e visualizzare i commenti precedenti;
- un blocco laterale, dove sono presenti la parte dedicata alla geolocalizzazione, alcune informazioni generali sulla località e una serie di link utili riguardanti la località messi a disposizione per l'utente.

Il titolo principale della parte centrale è il nome della località, il titolo secondario la categoria di appartenenza ed infine i titoli minori sono gli argomenti presenti nella descrizione della località. Alcuni argomenti della descrizione della località, essendo vasti, sono stati sviluppati secondo una lista di definizioni, dove vi è un elemento da definire (ad esempio un parco) e la sua descrizione. Il footer è uguale a quello delle pagine di categorie di località (sezione 3.5.2).

3.7 Pagina di ricerca fallita

È stata prevista anche una pagina creata con CGI per assistere l'utente nel caso in cui la ricerca con la searchbar non andasse a buon fine; più precisamente:

1. l'utente può trovarvi i link per andare a una pagina di categoria delle località tramite il box di navigazione;
2. l'utente può tornare alla homepage tramite un link proposto nella parte centrale della pagina oppure tramite tre link nella stessa pagina;
3. l'utente viene invitato a riformulare la sua ricerca.

Questa pagina è stata pensata (con leggere modifiche) anche per assistere l'utente e fornirgli il link alla mappa del sito in caso di errore 404, ma naturalmente non è stato possibile adottare questa soluzione nel progetto poichè non si hanno i permessi per modificare l'.htaccess del server TecWeb.

4 Front-end

Gli scripts introdotti per la parte front-end, interamente sviluppata in **JavaScript**, offrono funzionalità che permettono una migliore esperienza per l'utente.

Nello specifico sono state inserite le seguenti features:

- un cookie per il salvataggio del nome utente al momento della pubblicazione di un nuovo commento
- il placeholder nella barra di ricerca località
- uno script relativo ai commenti utente che si occupa di controllare, caricare e nascondere i dati inseriti nella form di pubblicazione
- un modulo che effettua la geolocalizzazione

Per ogni caratteristica sopra elencata si è cercato di separare il codice nel modo più chiaro possibile; a tal fine si è scelto di utilizzare la libreria `require.js` (<http://requirejs.org>) che si occupa essenzialmente di caricare files situati in percorsi diversi modularizzando quindi i vari scripts. `Require.js` si è dimostrato un valido *module loader* anche per la sua estesa compatibilità con i browsers (IE 6+, Firefox 2+, Safari 3.2+, Chrome 3+, Opera 10+).

4.1 Gestione cookie

Per immagazzinare persistentemente la username dell'utente abbiamo deciso di creare un cookie. Lo scopo di questo è velocizzare la compilazione di un commento infatti: l'utilizzatore, avendo pubblicato una prima volta, alla sua seconda troverà già riempito il campo nome utente. Nei nostri scripts sono state implementate una funzione per l'inizializzazione ed una per il recupero/lettura del cookie appena creato.

In fase di creazione del cookie, oltre alla coppia chiave-valore abbiamo aggiunto come terzo parametro una data di scadenza impostata a 15 giorni. La funzione di recupero restituisce il valore del cookie se questo è presente, successivamente il valore verrà inserito nel documento html e l'utente vedrà la username già compilata.

4.2 Inserimento placeholder in barra di ricerca

Poichè il nostro progetto è conforme allo standard xhtml 1.0 strict ed il tag *placeholder* non vi è definito, abbiamo dovuto usare Javascript per riuscire nel nostro intento di includerlo. Grazie ai metodi *onblur* ed *onfocus* abbiamo potuto forzare il valore di input della nostra barra di ricerca a seconda che questa si trovi in focus oppure no.

4.3 Visualizzazione dei commenti (lato client)

4.4 Geolocalizzazione

In questo progetto abbiamo inoltre deciso di implementare un modulo che fa uso delle API HTML5 per la geolocalizzazione. L'introduzione di tale caratteristica ha il puro scopo di migliorare la qualità del sito; in assenza di questa, l'utente avrà comunque accesso a tutte le funzionalità di base.

Grazie al modulo in questione, dopo aver acconsentito a fornire i dati sulla propria posizione e a seconda della pagina in cui si trova, all'utente verranno mostrate diverse informazioni.

In particolare, spostandosi sulle pagine "mare.html" o "citta.html" oppure "montagna.html" verranno visualizzate:

- una mappa a livello stradale avente tanti marker quante sono le località recensite all'interno della sezione appena raggiunta ed il marker che segnala la posizione attuale;

- una descrizione recante per ognuna delle località della sezione, la distanza in km dalla posizione attuale.

In alternativa, spostandosi sulle pagine delle località recensite (es: "praga.html") verranno visualizzate:

- una mappa a livello stradale ad un livello di zoom maggiore rispetto al precedente ed un solo marker posizionato nel centro della città considerata;
- una piccola descrizione che informa sulla distanza tra la città considerata e la posizione attuale.

È stato scelto di utilizzare Google Maps poiché gli utenti hanno una maggiore familiarità con le mappe offerte da Google², per la qualità della documentazione relativa alle API offerta da Google e perchè l'utente con queste mappe riesce facilmente a spostarsi, aumentare il dettaglio di zoom oppure passare in modalità StreetView in modo semplice ed immediato.

4.4.1 L'implementazione del modulo in dettaglio

Per svolgere più leggeremente l'attività di codifica è stata inclusa una libreria di utilità, underscore.js (<http://underscorejs.org>). Vi è un unico script che viene inizializzato al caricamento della pagina ed è valido sia per le pagine di sezione, (es: "mare.html") sia per le pagine delle località turistiche (es: "praga.html"). Questo è stato possibile attraverso la ricerca di un id presente solamente nelle pagine relative alle categorie, ma assente nelle pagine delle località. In tal modo lo script è in grado di discriminare se la pagina è di una categoria o di una località.

Per ottenere i dati voluti abbiamo dovuto realizzare una base-dati con le informazioni delle località recensite attraverso un oggetto JavaScript così formato:

```
var località = {
  "Parigi": {
    "name": "Parigi",
    "loc": "Città",
    "lat": 48.856614,
    "lon": 2.3522219
  }
}
```

La funzione principale dalla quale estrapoliamo poi tutti i dati è *getLocation()*. Invocando questa funzione viene innanzitutto verificato se il browser supporta la geolocalizzazione, fornendo un messaggio di errore nel caso in cui non la supportasse; se invece il browser supporta questa funzionalità, viene chiamata la funzione *navigator.geolocation.getCurrentPosition(showPosition)* la quale recupera latitudine e longitudine della posizione attuale del dispositivo utilizzato dall'utente.

La funzione *CalcolaDistanza(myLat, myLon, lat, lon)*, come dice il nome, si occupa di calcolare la distanza fra 2 coordinate geografiche; essa prende come argomenti latitudine e longitudine dei due punti desiderati ed attraverso la formula dell'emisenoverso ne restituisce la distanza in km.

La funzione *showPosition(position)* ha lo scopo di creare l'interfaccia per i dati ed è infatti la funzione più corposa dello script; essa effettua :

- l'inizializzazione e successivamente la collocazione della mappa con i marker nel documento HTML;
- il filtraggio delle sole località da considerare;

²Fonte: comScore Mobile Metrix, analisi effettuata negli Stati Uniti a Giugno 2014.

- il riordinamento per distanza minore dalla posizione attuale delle località nelle pagine di sezione;
- la creazione e la collocazione degli elementi html che costituiscono le informazioni di distanza.

La funzionalità di geolocalizzazione offerta da *What To Visit* è stata testata con i browser Google-Chrome ver.40+, Firefox ver.35+ sia in versione desktop che mobile.

5 Back-end

Per quanto riguarda la programmazione lato server, il sito *What To Visit* offre due funzionalità: con la prima l'utente può inserire commenti, mentre con la seconda si permette all'utente di accedere alla pagina di una località tramite una searchbar.

5.1 Inserimento commenti

Dal momento che il sito offre la funzionalità di visualizzare commenti inseriti dagli utenti, questi devono essere salvati in uno o più file nel server. Più precisamente, ogni commento viene salvato in un file XML chiamato `commenti.xml` presente nel server. Questo documento, valido rispetto allo schema `commenti.xsd` (progettato seguendo il modello *Tende alla Veneziana*), contiene delle liste di commenti divisi per località; ogni commento, a sua volta, possiede un corpo, un utente che l'ha scritto e una data in cui è stato scritto (nel formato AAAA-MM-GG come anticipato nella sezione 2). Naturalmente, all'inizio una località può non avere alcun commento oppure, se qualche commento dovesse essere rimosso per cause di forza maggiore (ad esempio illegalità o perchè viola i diritti d'autore di un individuo, come specificato nella pagina della Normativa della Privacy), un tag `localita` può rimanere senza commenti.

Lo script di inserimento commenti è stato scritto in Perl utilizzando la tecnologia **CGI** (Common Gateway Interface), in modo da riuscire facilmente a ottenere i dati passati con la form per inserire un commento presente nelle pagine delle località.

Successivamente lo script procede utilizzando la libreria **LibXML**, con le cui funzionalità legge il file dei commenti e cerca il nodo `localita` in cui inserire il commento con una query XPath. Una volta trovato, viene creato il frammento XML da aggiungere alla lista dei commenti già presenti per la stessa località. Una volta aggiunto il commento, lo script scrive la modifica sul file XML e reindirizza l'utente alla pagina della località in cui ha inserito il commento.

5.2 Barra di ricerca

L'utente durante la navigazione nel sito ha a disposizione anche una barra di ricerca; tramite questa, può scrivere il nome di una località (case insensitive in lingua italiana o in lingua originale) e venire reindirizzato direttamente alla pagina della località ricercata; se la ricerca non dovesse trovare nessun risultato, l'utente deve visualizzare una pagina che lo informi che la ricerca non è andata a buon fine.

Per ottenere questo risultato, è stato scritto uno script in Perl utilizzando la tecnologia **CGI** per acquisire il parametro `$place` dove è presente la richiesta effettuata dall'utente e, in caso di ricerca fallita, per creare la pagina da far visualizzare all'utente.

Lo script inizialmente confronta la stringa ricevuta in input con delle alternative prefissate; se la stringa passata corrisponde ad una di queste, allora l'utente viene reindirizzato alla località cercata con successo. In caso contrario, si esegue una stampa di una pagina con la sezione head composta come quella nelle altre pagine, header come nelle pagine secondarie, breadcrumb, nav con possibilità di andare nelle pagine delle categorie, corpo (dove si informa l'utente di cosa è successo) e footer.

6 Test effettuati

In questa sezione si elencano i test effettuati per ogni prodotto di progetto. Per i test di accessibilità eseguiti in remoto, il sito è stato caricato su GitHub Pages.

6.1 Validazione XHTML

Ogni pagina è stata validata con il tool offerto da W3C all'indirizzo http://validator.w3.org/#validate_by_input.

6.2 Validazione CSS

Ogni foglio di stile CSS è stato validato con il tool offerto da W3C all'indirizzo https://jigsaw.w3.org/css-validator/#validate_by_input.

6.3 Validazione XML e XSD

Il file XML dove sono salvati i commenti è stato validato rispetto al suo schema XSD con il tool offerto da *freeformatter.com* all'indirizzo <http://www.freeformatter.com/xml-validator-xsd.html>.

6.4 Validazione XSLT

Il file XSLT che permette di trasformare il file XML dove sono salvati i commenti è stato validato (si è anche verificato il suo corretto funzionamento) con il tool offerto da *freeformatter.com* all'indirizzo <http://www.freeformatter.com/xsl-transformer.html>.

6.5 JavaScript

Ogni script JavaScript è stato analizzato tramite il tool di analisi statica di *WebStorm*, affinché venissero trovati errori e miglioramenti al codice. Ogni frammento di codice è stato testato in locale ed in remoto non tralasciando alcun valore nel dominio dei possibili valori in ingresso. Ogni test è stato effettuato tenendo visibile la **JavaScript Console** del browser, così da notare eventuali warning o errori nel codice. I test hanno avuto esito positivo sul prodotto finale.

6.6 Perl e CGI

Ciascuno degli script Perl e CGI è stato verificato in locale, con esiti positivi.

6.7 Cynthia Says

Uno dei tool che è stato utilizzato per l'accessibilità delle pagine è Cynthia Says, inserendo l'URL di ogni pagina caricata su GitHub Pages e richiedendo la conformità a WCAG 2.0 AAA. Questa scelta è dovuta al fatto che i tentativi per verificare l'accessibilità di una pagina sono limitati a 10 nell'arco di 24 ore. Di seguito vengono riportati i risultati ottenuti per **tutte** le pagine del sito:

Conformità	Esito
WCAG 2.0 A	Conforme
WCAG 2.0 AA	Conforme
WCAG 2.0 AAA	Non conforme

Le pagine del sito non sono conformi a WCAG 2.0 AAA poichè:

1. *Cynthia Says* non riconosceva come pagine di aiuto “Mappa del sito e “F.A.Q.”;
2. il contrasto richiesto per il terzo grado di accessibilità era troppo elevato per poter garantire una resa grafica piacevole del sito.

6.8 WAVE

WAVE (Web Accessibility Evaluation Tool) è uno strumento offerto da *WebAIM* (WEB Accessibility In Mind) per verificare l'accessibilità delle pagine. Questo strumento analizza una pagina e fornisce errori, avvertimenti (warnings), caratteristiche positive, struttura basilare di una pagina e indicazioni sul contrasto.

Più in particolare, sono stati corretti gli errori trovati (relativi agli standard 508 e WCAG) non relativi alla lingua delle pagine, dal momento che nonostante il tag html fosse ben-formato e contenente l'indicazione sulla lingua utilizzata nella pagina in modo corretto, il sito continuava a segnalare il falso positivo.

I warnings sono stati corretti solo se effettivamente utili; questo dal momento che sono stati introdotti volutamente titoli o link ridondanti per aiutare l'utente a capire dove si trova e che l'uso dei tabindex viene segnalato come warning, anche se questi sono stati messi per permettere all'utente di raggiungere subito i link della pagina più rilevanti, saltando elementi di navigazione o breadcrumb a seconda dei casi.

I contrasti dei colori sono stati regolati per riuscire a superare almeno il livello di accessibilità WCAG 2.0 AA.

6.9 Fangs

Fangs è un'estensione per i browser che consente di visualizzare un sito nello stesso modo in cui verrebbe visualizzato da uno screen reader, fornendo il testo come sarebbe letto da questo dispositivo, la lista delle intestazioni e la lista dei link presenti nella pagina.

Sono state controllate tutte le pagine, con esito positivo: le intestazioni seguono l'ordine logico pensato dai componenti del gruppo, i link sono visualizzati nell'ordine aspettato e non sono presenti elementi estranei nel testo corrispondente all'audio che sentirebbe un utente con screen reader, mentre gli aiuti come il "salta la navigazione" sono presenti.