



WHAT TO VISIT

Graziano Grespan 1003760

Carlo Munarini 1050128

Federica Speggiorin 1051031

Sebastiano Valle 1050123

8 febbraio 2015

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

Indice

1	Abstract	4
1.1	Suddivisione ruoli	4
1.2	Schema organizzativo	4
2	Individuazione degli utenti e delle loro esigenze	5
3	Struttura	6
3.1	Struttura organizzativa	6
3.2	Progettazione della struttura delle pagine	6
3.2.1	XHTML and Semantic Markup	6
3.2.2	Tighter, Firmer Pages Guaranteed: Structure and Semantics	7
3.3	Parti comuni a tutte le pagine	7
3.3.1	Head	8
3.3.2	Body	8
3.4	Pagine di primo livello	8
3.5	Pagine di secondo livello	9
3.5.1	Pagine secondarie	9
3.5.2	Categorie di località	9
3.6	Pagine di terzo livello	10
3.7	Pagina di ricerca fallita	10
4	Presentazione	11
4.1	I colori	11
4.2	CSS e Parti comuni a tutte le pagine	12
4.2.1	Header	12
4.2.2	Navigation menù	12
4.2.3	Searchbar	13
5	Front-end	14
5.1	Gestione cookie	14
5.2	Inserimento placeholder in barra di ricerca	14
5.3	Gestione dei commenti	15
5.3.1	Visualizzazione commenti	15
5.3.2	Rendere invisibili i commenti	15
5.3.3	Pubblicazione dei commenti	16
5.4	Geolocalizzazione	16
5.4.1	L'implementazione del modulo in dettaglio	17
6	Back-end	18
6.1	Inserimento commenti	18
6.2	Barra di ricerca	18
7	Accessibilità	19
7.1	Trasformazione elegante	19
7.1.1	Separazione struttura-comportamento	19
7.1.2	Separazione struttura-presentazione	19
7.1.3	Separazione presentazione-comportamento	19
7.1.4	Porzioni di sito visibili	19
7.2	Linee guida per l'accessibilità	19
7.2.1	Alternative a contenuti audio e visivi	19
7.2.2	Non fare affidamento sul colore	19
7.2.3	Uso appropriato dei tag	20
7.2.4	Linguaggi naturali	20
7.2.5	Trasformazione elegante delle tabelle	20
7.2.6	Trasformazione elegante delle nuove tecnologie	20

7.2.7	Contenuti che cambiano nel corso del tempo	20
7.2.8	Interfacce utente	20
7.2.9	Indipendenza da dispositivo	20
7.2.10	Meccanismi di fallback	21
7.2.11	Raccomandazioni W3C	21
7.2.12	Orientamento	21
7.2.13	Navigazione	21
7.2.14	Semplicità dei contenuti	21
8	Test effettuati	22
8.1	Validazione XHTML	22
8.2	Validazione CSS	22
8.3	Validazione XML e XSD	22
8.4	Validazione XSLT	22
8.5	JavaScript	22
8.6	Perl e CGI	22
8.7	Cynthia Says	22
8.8	WAVE	23
8.9	Vischeck	23
8.10	Fangs	24
8.11	Lynx	24
8.12	Compatibilità con i vari Browser	24
8.12.1	Internet Explorer	25
8.12.2	Geolocalizzazione	25
8.12.3	Visualizzazione commenti degli utenti	25
8.13	Performance	25

Elenco delle figure

1	Organizzazione gerarchica del sito	6
2	Gli Header di WTV	12
3	I Navigation Menù di WTV	13
4	Test con Vischeck su una pagina di una località	24
5	Homepage: tempi di caricamento	25
6	Homepage: Grafico a 'torta'	26
7	Performance in <i>citta.html</i> e <i>london.html</i>	26

Elenco delle tabelle

1	Conformità allo standard WAI 2.0 X WCAG	22
---	---	----

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, Carlo Munarini, 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 stati 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):

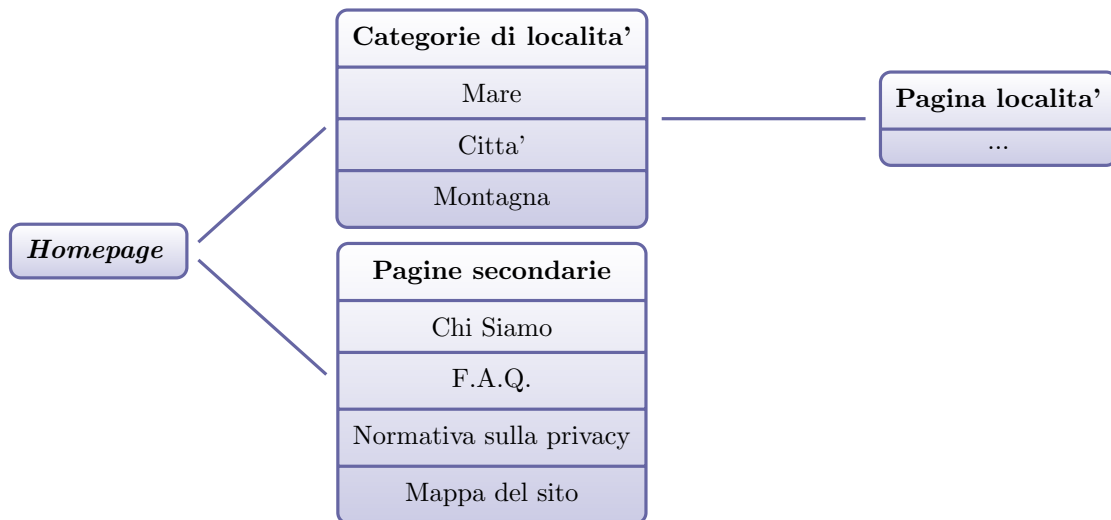


Figura 1: Organizzazione gerarchica del sito

Per ogni 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 ancora 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.

In tutte le pagine sono presenti due pulsanti nella sezione header che svolgono una funzionalità di trigger per dei menù: alla pressione di questi, vengono assegnate tramite JavaScript delle classi a dei menù che verranno visualizzati nella versione mobile del sito, come descritto nella sezione 4.2.2.

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 riescano a dedurre che si trovano nella homepage.

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 affianco al 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 6);
- 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*
- **F.A.Q.:** 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 i 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; 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 la 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 e visualizzare i commenti precedenti;
- un blocco laterale, dove sono presenti la parte dedicata alla geolocalizzazione, alcune informazioni generali 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.

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 Presentazione

In questa sezione vedremo come abbiamo deciso che l'utente debba visualizzare il nostro sito. Come anticipato nel punto 3.3.1, ogni pagina può essere visualizzata in maniera diversa a seconda del dispositivo utilizzato, vi illustreremo quindi le scelte che abbiamo attuato nelle diverse pagine (I, II, III Livello) in maniera tale che potessero essere visualizzate nel migliore dei modi. Essendo il nostro sito orientato ad un target non specifico che vuole apprendere informazioni su diverse località, abbiamo deciso di permettere 4 diversi modi di visualizzare la pagina, introducendo quindi 4 fogli di stile CSS:

- **Foglio di stile layout.css:** Questo layout viene applicato a tutti i dispositivi che visualizzano il nostro sito con più di 780px di larghezza, abbiamo però deciso di ottimizzare tale layout in maniera che potesse essere accessibile anche a coloro che utilizzano screen-reader
- **Foglio di stile playout.css:** Una persona che trova informazioni riguardo ad una località potrebbe decidere scaricare o stampare il contenuto della nostra pagina così da poterlo guardare in un secondo tempo, su un PDF od un foglio stampato. Questo stile serve proprio affinché un utente possa stampare ogni pagina del nostro sito, visualizzando al meglio le informazioni chiave della pagina
- **Foglio di stile tlayout.css:** Questo foglio di stile entra in gioco quando la pagina viene visualizzata in dispositivi con larghezza inferiore a 780px. In tal modo il nostro sito offre una visualizzazione pulita anche su Tablet e Cellulari
- **Foglio di stile mlayout.css:** Nel caso What To Visit venga visualizzato su un cellulare o in una finestra con larghezza inferiore ai 480px, ecco che verrà chiamato in causa questo quarto foglio di stile, molto simile al precedente, che permette però un'eccellente visualizzazione delle pagine di III Livello anche sui cellulari. Il sito è quindi riconosciuto da Google come *Mobile-Friendly*²

4.1 I colori

Prima di iniziare a guardare nel dettaglio l'aspetto grafico di ogni pagina, vogliamo presentare quelli che sono i colori adottati nel nostro sito. In What To Visit infatti gli utenti avranno di fronte a se delle pagine che utilizzano pochi colori, ai quali abbiamo cercato di attribuire un significato:

- **Verde scuro (#1D653C):** Con questo colore, simile ad un Verde primavera scuro, abbiamo voluto indicare gli elementi non attivi ma che possono essere attivati (come pulsanti per i commenti o i link), gli elementi fissi della pagina (come footer ed header) oppure elementi non attivabili che però caratterizzano una località o un collegamento (come succede per indicare a quale categoria appartiene la località o per indicare un link che non è stato visitato e porta in una pagina esterna al nostro sito)
- **Verde chiaro (#2ECC71):** Con questa via di mezzo tra un verde primavera ed un verde smeraldo, si è deciso di indicare quegli elementi che sono già stati attivati o che potrebbero esserlo poiché puntati dal cursore. Un esempio possono essere i link già visitati (esterni o interni al sito), link puntati dal cursore o ancora, i pulsanti per i commenti qualora siano stati attivati o possano portare al cambiamento/essere frutto di un cambiamento della pagina
- **Grigio Scuro (#444444):** Questo grigio scuro viene utilizzato come colore del testo di contenuto e come colore di background per le caselle di testo nel quale l'utente deve per l'appunto inserire informazioni o commenti

²Il Test di compatibilità con dispositivi mobili è stato eseguito all'indirizzo: <https://www.google.com/webmasters/tools/mobile-friendly/>

- **Bianco (#FFFFFF):** È presente in tutte le pagine in quanto, colore di background del sito e della barra di ricerca presente nella breadcrumb (Vedi punto 3.5), unica eccezione riguardante il grigio scuro come background-color delle caselle di testo
- **Rosso (#E84444):** Utilizzato solamente in due casi, nelle pagine di III Livello, questo colore indica all'utente che qualcosa non va, o che premendo un determinato pulsante potrebbe cancellare i dati inseriti in una form
- **Grigio Chiaro (#C9C9C9):** Questo colore viene utilizzato solamente in un caso, quello in cui un pulsante, anche se premuto, non cambierebbe la pagina. Questa scelta è dovuta al fatto che si vuol cercare di far capire all'utente che quel pulsante è sostanzialmente inutile in quel determinato momento ma che potrebbe essere utilizzato in un secondo momento

4.2 CSS e Parti comuni a tutte le pagine

Durante la realizzazione del sito abbiamo cercato di creare un layout semplice, accessibile, utilizzabile ma soprattutto visualizzabile al meglio sul maggior numero possibile di Browser. Il nostro sito limita quindi l'utilizzo del linguaggio CSS3, un linguaggio presente ma allo stesso tempo non fondamentale, che consente quindi un degrado elegante del sito nel caso di mancato supporto a determinate funzioni.

CSS3 è usato in diverse parti del sito, specialmente nelle parti comuni a tutte le pagine, già descritte nella sezione 3.3, e specialmente nell'ambito della visualizzazione del sito per dispositivi mobili. Ora però vediamo come vengono visualizzati tali elementi a seconda dei dispositivi utilizzati:

4.2.1 Header

L'**Header** è un elemento presente in tutte le pagine del nostro sito (I, II e III Livello) ma viene visualizzato in maniera differente a seconda che venga utilizzato un foglio di stile piuttosto che un altro. La differenza principale è il modo di mostrare il nome del sito, infatti nella versione **textScreen** e **Print**, l'utente vedrà sempre comparire in *alto a sinistra* il nome del sito come una scritta, mentre nel layout per **Dispositivi Mobili**, tramite una tecnica di *image-replacement*³ abbiamo deciso di far visualizzare all'utente il logo di What To Visit.



Figura 2: Gli Header di WTV

4.2.2 Navigation menù

Come accade per l'**Header** anche il **Menù di Navigazione** è un elemento presente in tutte le pagine del nostro sito. Questo elemento però viene visualizzato in maniera diversa a seconda del livello della pagina e anche del dispositivo utilizzato. Nella *Homepage* infatti esso è visualizzato nell'**Header** affianco al nome del sito, mentre in tutte le altre pagine rimane sempre nella *parte alta della pagina* ma si trova sotto alla **Breadcrumb**, affianco al contenuto. Il nostro intento però è quello di soffermarsi su come si presenta all'utente questo elemento, quindi è importante dire che le voci di questo menù, a differenza degli altri link, saranno di colore bianco su sfondo verde. Nella versione **Screen** e nella *Homepage*, le voci del **Menù di Navigazione** riprendono quelli che sono i caratteri distintivi dei box che indicano le categorie, creando così uno stile particolare per le pagine di I livello, cercando quindi di dare un punto di riferimento all'utente che dovrà saper distinguere la sua posizione una volta che arriva o ritorna nella *Homepage*.

³ Abbiamo utilizzato una piccola variante (Text-indent: -10em e non -9999px) del *Phark's Method* - citato da Zeldman nel 2003: <http://www.zeldman.com/daily/0703b.shtml#au1103>

La versione per **Dispositivi Mobili** invece, non fa distinzione tra pagine di *I, II o III livello* e visualizza il **Menù di Navigazione** solamente se l'utente attiverà il pulsante posto sempre in *alto a destra* sullo schermo del dispositivo, affianco a quello per la **Casella di ricerca**⁴. In questo caso il menù apparirà in *posizione fissa* coprendo il contenuto sotto ad esso e, nel caso di menù a più livelli di profondità cercherà di rimarcare questa gerarchia aumentando lo spessore dei bordi superiori ed inferiori delle voci del sotto-menù.

Altre due particolarità delle voci del **Menù di Navigazione** sono:

- Il fatto che un link sia già stato visualizzato o meno, non cambia le caratteristiche dell'ancora, questo poichè il **Menù di Navigazione** serve all'utente per poter spostarsi all'interno del sito e quindi abbiamo ritenuto, non necessario che sapesse se fosse o meno già stato in una determinata categoria o pagina secondaria;
- La seconda caratteristica è che, nel caso il visitatore si trovi in una *pagina di II livello*, la voce che indica la posizione in cui si trova, non sarà cliccabile, ma soprattutto verrà caratterizzata da un'immagine posta alla *sinistra* di essa. Anche nel caso di *pagine di III livello*, il visitatore vedrà un'immagine affianco alla voce che indica la categoria in cui si trova, questa volta però l'ancora sarà utilizzabile.



(a) Navigation Menù in Screen



(b) Navigation Menù su Dispositivi Mobili

Figura 3: I Navigation Menù di WTV

4.2.3 Searchbar

La **Searchbar** è un elemento presente in *tutte le pagine del sito* e, nonostante possa trovarsi in posizioni diverse, nell'**Header** per l'*Homepage* e nella **Breadcrumb** nelle *pagine di II e III livello*, essa sarà sempre riconoscibile in quanto, oltre ad essere l'unica casella di testo con sfondo bianco, il testo sarà verde scuro e soprattutto il *submit-botton* sarà rappresentato da una lente d'ingrandimento, immagine spesso usata nel web per indicare la ricerca, che verrà posta a destra della casella. Questo elemento non viene visualizzato in maniera completamente diversa a seconda del foglio di stile, eccezion fatta per il foglio `playout.css` (Vedi punto 4.) che elimina dalla stampa tale elemento in quanto ritenuto non rilevante nel caso l'utente voglia stampare informazioni su località o pagine secondarie. Nelle visualizzazioni per **Screen** e per **Dispositivi Mobili**, la searchbar si differenzia solamente per la posizione del *submit-botton*, posto a destra nella versione **Screen** ed a sinistra in quella per **Dispositivi Mobili**.

⁴Entrambi i pulsanti sono sempre visibili e sono posizionati in *alto a destra* grazie all'attributo *Position: fixed*. Questi pulsanti, per cui abbiamo utilizzato la tecnica di *image-replacement* precedentemente descritta, non cambiano all'attivazione, poichè pensiamo che l'utente capisca cosa abbia portato all'apertura del menù e cosa dovrà premere per chiudere i box

5 Front-end

Gli script 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 script.

`Require.js` si è dimostrato un valido *module loader* anche per la sua estesa compatibilità con i browser (IE 6+, Firefox 2+, Safari 3.2+, Chrome 3+, Opera 10+).

Nello script `main.js` sono presenti delle variabili globali; tuttavia, queste vengono inizializzate ed utilizzate solo quando ciò è sia possibile che necessario (ad esempio una variabile relativa ad un elemento presente solo nelle pagine di terzo livello non viene inizializzata al caricamento della homepage).

5.1 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 script 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.

5.2 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 una funzione JavaScript per riuscire nel nostro intento di includerlo.

Si è quindi deciso di collegare la gestione degli eventi *onBlur()* e *onFocus()* a due funzioni definite nello script `main.js`. Queste due funzioni assicurano un comportamento tale che se la barra di ricerca non possiede il focus e non contiene testo inserito dall'utente, visualizza un **placeholder** "Nome località per aiutare l'utente a capire cosa inserire in quella barra di ricerca, fornendo una classe diversa a questo testo per enfatizzarla e distinguerla dal testo inserito dall'utente.

Se invece la barra di ricerca acquisisce il focus, vengono aumentate le dimensioni della stessa, scompare il placeholder (se presente) ma non un eventuale testo inserito dall'utente. A questo punto l'utente può inserire un testo, al quale verrà attribuita la classe predefinita prevista per il testo.

Se la barra di ricerca perde il focus, allora questa continuerà a visualizzare il testo inserito dall'utente oppure verrà nuovamente visualizzato un placeholder (con classe di enfasi), nel caso in cui l'utente non avesse inserito testo.

5.3 Gestione dei commenti

Nelle pagine delle località (sezione 3.6) sono stati inseriti dei pulsanti per visualizzare, nascondere e pubblicare commenti; qui di seguito viene descritto il modo con cui queste tre operazioni (visualizzazione, rendere invisibili e pubblicazione dei commenti) sono state gestite sul lato client di What To Visit.

5.3.1 Visualizzazione commenti

I commenti sono visibili per tutti i browser fuorchè Internet Explorer, nel cui caso viene mostrato un messaggio di scuse all'utente per la mancanza della funzionalità.

Questo è dovuto al fatto che i commenti non sono già presenti nella pagina al momento del caricamento, ma vengono importati da un file XML e quindi trasformati tramite un foglio di stile XSLT, che trasforma i commenti in liste prive di ordine i commenti presenti nel file; in particolare:

- lo script che applicava il foglio di stile XSLT al file XML forniva degli output parziali e incompleti utilizzando la funzione prevista per Internet Explorer, mentre la funzione supportata da tutti gli altri browser forniva un frammento di documento facilmente manipolabile;
- il risultato fornito dalla funzione prevista per IE non riusciva ad essere gestito come elemento nodo ma solamente come stringa, mentre con la funzione supportata da tutti gli altri browser veniva prodotto un elemento di tipo nodo sul quale era possibile ricercare tra i nodi figli la lista contenente i commenti relativi solamente alla località della pagina.

Se non vi sono commenti da visualizzare viene visualizzata la stringa “Non vi sono commenti da visualizzare, altrimenti i commenti vengono visualizzati a gruppi di due ad ogni pressione del pulsante “Visualizza commenti fin tanto che ci sono commenti; la divisione di documento contenente i commenti viene creata ed allocata nel documento solo alla prima pressione del pulsante “Visualizza commenti.

I pulsanti “Visualizza commenti può essere in tre stati e questi vengono assegnati via JavaScript tramite delle classi:

- **com-runnable**: non sono visualizzati commenti e si può richiedere la loro visualizzazione (se l'utente sta usando Internet Explorer o se non vi sono commenti, solo in seguito alla pressione del pulsante sarà avvisato dell'impossibilità di mostrargli dei commenti sulla località)
- **com-running**: sono attualmente visualizzati alcuni commenti ma è possibile visualizzarne altri
- **com-notrunnable**: non sono presenti ulteriori commenti da visualizzare

5.3.2 Rendere invisibili i commenti

Dopo aver visualizzato tutti i commenti di una località o parte di questi, l'utente ha la possibilità di nasconderli tramite la semplice pressione di un pulsante.

Ciò è stato realizzato collegando l'evento onclick del pulsante “Nascondi commenti ad una funzione `nascondi_c()`, che toglie i commenti presenti nella divisione in cui erano visualizzati e pone il pulsante “Visualizza commenti nello stato **com-runnable** se erano presenti commenti.

Il pulsante “Nascondi commenti può essere in due stati, assegnati anch'essi attribuendo delle classi al pulsante:

- **com-runnable**: sono visualizzati commenti e si offre all'utente la possibilità di nasconderli
- **com-notrunnable**: non sono visualizzati commenti, perciò per non creare confusione all'utente si segnala che il pulsante è inutilizzabile (un'eventuale pressione di questo non produrrebbe alcun output visibile)

5.3.3 Pubblicazione dei commenti

Nel sito è prevista per gli utenti la possibilità di inserire un proprio commento inerentemente ad una determinata località. A questo scopo, al termine della sezione della pagina dedicata alla visualizzazione dei commenti, vi è il pulsante “Pubblica il tuo commento che permette di visualizzare un form per l’inserimento. Tale pulsante può trovarsi in due stati:

- **Disattivo:** il form non viene visualizzato
- **Attivo:** il pulsante è stato attivato dall’utente causando quindi la visualizzazione del form per l’inserimento.

Il form ha inizialmente, quando disattivo, una classe `class=hidden` che viene gestita in CSS per nascondere. Nel momento in cui l’utente attiva il pulsante “Pubblica il tuo commento il codice JavaScript si occupa di modificare tale classe eliminandone il valore. In questo modo il form compare permettendo all’utente di inserire il proprio commento. Il form è costituito da due campi e due pulsanti così caratterizzati:

- **Username:** campo in cui viene richiesto uno username di identificazione eventualmente compilato automaticamente tramite il valore del cookie precedentemente creato
- **Testo del commento:** campo in cui l’utente può inserire il testo del proprio commento
- **Pubblica:** pulsante la cui attivazione determina l’invio del commento
- **Annulla:** pulsante la cui attivazione determina l’annullamento dell’operazione

A ciascuno dei campi sono associate una o più funzioni che si occupano di controllare i dati in input. E’ prevista infatti l’obbligatorietà di inserire un valore per ognuno dei campi **Username** e **Testo del commento**. Alla perdita del focus dei campi, se il loro contenuto è vuoto, vengono visualizzati i relativi messaggi di errore. Il pulsante **Pubblica** si occupa di attivare il codice Perl che esegue l’effettivo inserimento del commento all’interno dell’apposito file XML. Il pulsante **Annulla** invece permette all’utente di annullare l’operazione di pubblicazione del proprio commento. Lo script ad esso associato, dopo aver eliminato il contenuto dei campi, nasconde il form e annulla l’operazione.

5.4 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 Google⁵, per la qualità della documentazione relativa alle API offerta da Google e perché l'utente con queste mappe riesce facilmente a spostarsi, aumentare il dettaglio di zoom oppure passare in modalità StreetView in modo semplice ed immediato.

5.4.1 L'implementazione del modulo in dettaglio

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 JSON 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;
- 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.

⁵Fonte: comScore Mobile Metrix, analisi effettuata negli Stati Uniti a Giugno 2014.

6 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.

6.1 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.

6.2 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.

7 Accessibilità

Come standard di accessibilità è stato scelto di raggiungere il grado WAI-AA WCAG 2.0. Di seguito sono descritte gli accorgimenti adottati dal nostro sito per perseguire questo obiettivo.

7.1 Trasformazione elegante

Le pagine del nostro sito rimangono accessibili anche se l'utente non può utilizzare alcune tecnologie o funzionalità per scelta personale, se il suo dispositivo di navigazione non le supporta o per cause di forza maggiore (e.g. svantaggi dal punto di vista fisico e psichico).

7.1.1 Separazione struttura-comportamento

In ogni pagina del sito non è stato introdotto alcun attributo per la gestione degli eventi associata ad un elemento strutturale e gli script sono inclusi nella pagina e non fanno parte del corpo di questa. Gli script JavaScript, a loro volta, non si preoccupano di creare intere pagine dinamicamente, ma si limitano a creare frammenti di queste in modo opportuno, ovvero quando ciò **non** è possibile staticamente.

7.1.2 Separazione struttura-presentazione

Le pagine del sito non contengono alcun foglio di stile ma ogni foglio di stile viene incluso in queste. Oltre a ciò, ogni elemento strutturale contiene solamente attributi relativi al suo significato semantico e non presentazionale (ad esempio non sono stati utilizzati gli attributi `bgcolor` e `font`).

Ogni id e ogni classe è stata dichiarata utilizzando nomi relativi alla semantica e non al modo in cui verranno presentati.

7.1.3 Separazione presentazione-comportamento

Gli script JavaScript non vanno a modificare le regole con cui gli elementi sono presentati all'interno delle pagine del sito, ma si limitano ad attribuire o aggiungere ad essi classi che verranno trattate con dei fogli di stile.

7.1.4 Porzioni di sito visibili

L'area cosiddetta "*above the fold*" contiene sempre i contenuti di maggior rilievo della pagina visualizzata, mentre elementi di minore importanza sono presenti in parti come il footer che non sempre possono essere visibili all'apertura di una nuova finestra.

7.2 Linee guida per l'accessibilità

In questa sottosezione viene descritto come il gruppo ha rispettato le linee guida del WAI.

7.2.1 Alternative a contenuti audio e visivi

Nel sito non sono presenti contenuti audio e video né applet, quindi non ci si pone il problema per questi. Al contrario, ogni immagine ha un attributo `alt` che fornisce l'equivalente testuale nel caso in cui non fosse possibile visualizzare l'immagine.

7.2.2 Non fare affidamento sul colore

Sono stati utilizzati tool per la verifica dell'accessibilità che controllano anche l'uso corretto del colore nel sito.

7.2.3 Uso appropriato dei tag

I tag sono stati utilizzati per il loro significato semantico, non sono state utilizzate tabelle per definire il layout delle pagine e, togliendo i fogli di stile, il sito rimane accessibile anche con le impostazioni predefinite dei browser.

7.2.4 Linguaggi naturali

Sono stati utilizzati marcatori sia per la pronuncia di parole in lingua straniera che per estendere la pronuncia di abbreviazioni e acronimi.

7.2.5 Trasformazione elegante delle tabelle

Non sono presenti tabelle nel sito.

7.2.6 Trasformazione elegante delle nuove tecnologie

1. CSS3
2. **JavaScript**: se un utente ha disattivato JavaScript sul proprio browser non è in grado di visualizzare i commenti (parte secondaria di contenuto), non visualizza il placeholder nella barra di ricerca, quando clicca su un link “esterno, questo viene aperto sulla stessa finestra e le immagini sulla home non fungono da link alle pagine delle categorie; tuttavia, senza queste funzionalità il sito continua ad offrire una più che buona user experience
3. Geolocalizzazione

7.2.7 Contenuti che cambiano nel corso del tempo

1. la searchbar cambia dimensione quando assume il focus e viene visualizzato un placeholder solamente se quando questa perde il focus non è presente testo (altrimenti rimane l’input immesso dall’utente);
2. premendo sui pulsanti “Visualizza commento, “Pubblica commento e “Nascondi commenti vengono rispettivamente visualizzati i commenti presenti, compare la form di inserimento commenti e vengono nascosti i commenti precedentemente visualizzati.

Questi elementi non causano problemi ad eventuali utenti che soffrono di epilessia poichè il loro cambio di stato non è troppo rapido. Allo stesso tempo, questi elementi si aggiornano dopo dei click e mantengono il loro stato fino ad una successiva interazione tramite click non causando un senso di disagio nell’utente che altrimenti vedrebbe il layout modificarsi di continuo sotto i suoi occhi.

7.2.8 Interfacce utente

Non sono previste delle interfacce utenti quali comandi vocali ed access key per il sito. Al contrario, sono stati previsti dei tab index per:

1. poter saltare delle voci di navigazione ridondanti;
2. accedere con priorità ai link potenzialmente più interessanti per l’utente (secondo previsioni dei componenti del gruppo).

7.2.9 Indipendenza da dispositivo

Come detto nelle sezioni 8.10 e 8.11, le pagine del sito sono state provate rispettivamente anche con emulatori di screen reader e browser testuali.

Non state utilizzate aree di immagini come link.

7.2.10 Meccanismi di fallback

Ogniquale volta che una certa tecnologia non fosse disponibile per visualizzare i contenuti previsti, si è scelto di non fornire il contenuto previsto (perché di secondaria importanza) oppure il sito offre dei meccanismi per i quali la degradazione è elegante (ad esempio la searchbar, se acquista il focus, si ingrandisce istantaneamente anziché effettuare una transizione).

7.2.11 Raccomandazioni W3C

Come detto in precedenza, le immagini presentano sempre l'alternativa testuale inserita utilizzando la tecnologia offerta da W3C.

Non sono presenti formati come shockwave e PDF.

7.2.12 Orientamento

Vengono inoltre forniti una mappa del sito ed una pagina di F.A.Q. riferite in ogni pagina del sito. Oltre a questo, in tutte le pagine eccetto la home (dove non è ritenuta necessaria) è presente una breadcrumb che indica all'utente la sua posizione all'interno del sito.

7.2.13 Navigazione

I link sono evidenziati in modo che siano distinguibili attraverso un test di *Drue Miller* e forniscono sempre un attributo "title" che informa l'utente sul contenuto della destinazione. In tutte le pagine eccetto la home (dove la navigazione viene indirizzata volutamente verso le categorie), è presente un menù di navigazione contenente i "sibling della pagina o, se l'utente si trova in una pagina di una località, visualizza i riferimenti alla homepage e alle categorie. In questo elemento viene indicata anche la pagina corrente con un'icona oppure, se l'utente è presente in una pagina da una località, viene visualizzata un'icona di colore differente dalla precedente vicino alla voce relativa alla categoria di appartenenza.

7.2.14 Semplicità dei contenuti

Il layout è coerente, consistente e riconoscibile in tutte le pagine del sito, come descritto nelle sezioni 3 e 4.

Si è cercato di tenere un linguaggio semplice nei contenuti del sito.

8 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.

8.1 Validazione XHTML

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

8.2 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.

8.3 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>.

8.4 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>.

8.5 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.

8.6 Perl e CGI

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

8.7 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

Tabella 1: Conformità allo standard WAI 2.0 X WCAG

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.

8.8 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.

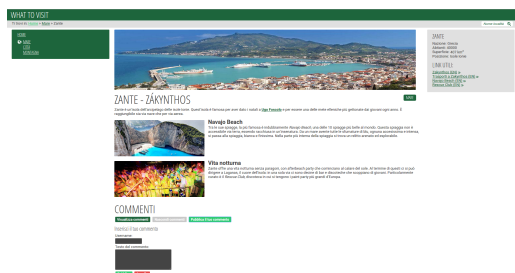
I contrasti dei colori sono stati regolati per riuscire a superare almeno il livello di accessibilità WCAG 2.0 AA.

8.9 Vischeck

Vischeck è uno strumento che permette di visualizzare immagini e pagine web come le vedrebbe una persona che presenta deficit visivi quali:

- Deuteranope: una forma di deficit di colore rosso/verde
- Protanope: un'altra forma di deficit di colore rosso/verde
- Tritanope: una forma di parziale o insufficiente discriminativa per il blu e il violetto

A causa del fatto che lo strumento per analizzare le pagine web era momentaneamente inutilizzabile sono state effettuate delle prove attraverso degli screen delle pagine web valutando la visione generale che avrebbero avuto utenti affetti dai deficit precedentemente esposti.



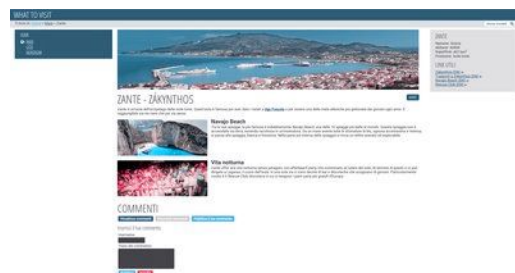
(a) Pagina originale (nessun deficit visivo presente)



(b) Deuteranope



(c) Protanope



(d) Tritanope

Figura 4: Test con Vischek su una pagina di una località

Dalle quattro immagini si può vedere come le pagine rimangano comunque accessibili e in particolare i link sono comunque visibili e ben distinguibili dall'utente.

8.10 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.

8.11 Lynx

Le pagine del sito sono state anche navigate tramite il browser testuale **lynx**, affinché si potesse verificare la facilità d'uso di *What To Visit* anche per gli utenti che fanno uso di questo strumento di navigazione.

8.12 Compatibilità con i vari Browser

In sito *What To Visit* è stato testato su vari browser e su diverse versioni di questi (sia desktop che mobile). In particolare risulta compatibile in tutte le sue funzionalità, eccetto la geolocalizzazione e la visualizzazione dei commenti degli utenti, con i seguenti in versione desktop:

- **Google Chrome:** versione 40.0+
- **Mozilla Firefox:** versione 32.0.3+
- **Internet Explorer:** versioni 9+
- **Opera:** versione 12.16+
- **Safari:** versione 5.0+

- **Maxthon:** versione 0.9.0.31+ (per Linux)

Inoltre, in versione mobile su:

- **Google Chrome:** versione 40.0+
- **Mozilla Firefox:** versione 35+

8.12.1 Internet Explorer

Per testare il sito sulle varie versioni di Internet Explorer è stato utilizzato lo strumento *IETester* disponibile gratuitamente online. Attraverso questo è stato rilevato che, fino alla versione 8 compresa del browser, gli script che si occupano di visualizzare i commenti e di rendere cliccabili le immagini delle categorie in home non funzionano.

8.12.2 Geolocalizzazione

La funzionalità di geolocalizzazione offerta da *What To Visit* è stata testata con tutti i browser precedentemente elencati tuttavia non risulta compatibile, in quanto non supportata, da Internet Explorer 8 (e versioni precedenti) e Maxthon per Linux (le altre versioni di Maxthon non sono state provate).

8.12.3 Visualizzazione commenti degli utenti

Nel sito è prevista la visualizzazione, per ogni località, dei commenti inseriti dagli utenti. Tale funzionalità tuttavia è supportata in tutti i browser testati ad eccezione di Internet Explorer, come spiegato nella sezione 5.3.1.

8.13 Performance

Per terminare la fase di testing sono state eseguite, con strumenti automatici, una serie di verifiche riguardo le performance del sito.

Ci siamo avvalsi di tools online⁶ che analizzano in profondità le pagine html volute e restituiscono: consigli di ottimizzazione, scale di punteggi per ogni aspetto del sito e una serie di statistiche che vedremo in dettaglio.⁷

Al fine di uniformare i risultati raccolti da questi strumenti abbiamo deciso di mostrare solo alcune delle pagine analizzate. In particolare le più significative (Homepage, *citta.html*, *london.html*) hanno dato i seguenti esiti:

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete			Fully Loaded		
						Time	Requests	Bytes In	Time	Requests	Bytes In
First View	1.052s	0.121s	0.493s	773	60	1.052s	16	355 KB	1.137s	17	356 KB
Repeat View	0.002s	0.000s	0.087s	181	60	0.002s	0	0 KB	0.015s	0	0 KB

Figura 5: Homepage: tempi di caricamento

In homepage come evidenziato in fig.6 dal diagramma circolare delle richieste e della quantità di dati trasmessi dal server, si nota la prevalenza del flusso delle immagini. Come è facile aspettarsi, le immagini occupano in termini di peso la maggior parte del sito ed essendo molteplici, le richieste per il loro recupero rappresentano più del 40% delle richieste totali.

⁶<https://developers.google.com/speed/pagespeed/insights/>, <http://www.webpagetest.org/>, <http://gtmetrix.com/>

⁷NB: Per poter utilizzare i tester il sito è stato hostato sulla piattaforma Github.io



Figura 6: Homepage: Grafico a 'torta'

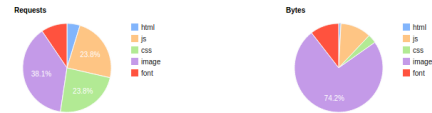
Lo stesso è replicato in *citta.html* e *london.html* (fig.7), eccezion fatta per *citta.html* dove viene fatto un uso più intenso dei fogli di stile.

Dopo aver eseguito i test sulle stesse pagine per 3 volte abbiamo stimato che il tempo medio di caricamento completo si attesta sui 1.4 secondi circa.

I controlli sulla velocità di caricamento sono stati effettuati su di un client residente negli Stati Uniti avente browser Chrome.

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete	Time	Requests	Bytes In	Fully Loaded	Time	Requests	Bytes In
First View	0.974s	0.186s	0.893s	968	89	0.974s	19	288 KB	1.078s	21	292 KB		
Repeat View	0.008s	0.000s	0.009s	101	89	0.008s	0	0 KB	0.035s	0	0 KB		

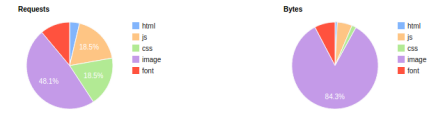
(a) *citta.html*: tempi di caricamento



(b) *citta.html*: grafico a 'torta'

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete	Time	Requests	Bytes In	Fully Loaded	Time	Requests	Bytes In
First View	1.456s	0.204s	0.894s	1016	186	1.456s	25	584 KB	1.553s	27	588 KB		
Repeat View	0.002s	0.000s	0.004s	143	186	0.002s	0	0 KB	0.002s	0	0 KB		

(c) *london.html*: tempi di caricamento



(d) *london.html*: grafico a 'torta'

Figura 7: Performance in *citta.html* e *london.html*

Complessivamente il sito ha totalizzato una media di 71 punti su 100 totali in Google PageSpeed Insights.

Le ottimizzazioni suggerite per aumentarne il punteggio andavano contro standards W3C e/o compatibilità con i browsers più obsoleti, abbiamo quindi cercato di alleggerire il sito (es: minificazione della libreria Require.js, immagini a risoluzione inferiore ecc.) trovando un buon compromesso tra velocità di caricamento e compatibilità con più browsers.