

## **ANALISI DEI REQUISITI**

Nome del cliente: Provincia di Vercelli

Tipo di attività del cliente: Promozione del territorio

Obiettivo del sito: Promozione del territorio

### **Descrizione del sito:**

Numero di pagine complessive:

La richiesta si articola in una home page con galleria modificabile e la sezione delle ultime news.

2 Pagine statiche per la promozione delle passeggiate e slow food.

Ulteriori due pagine dinamiche per contatti e visualizzazione dati territorio organizzati.

Un modulo di amministrazione del sito.

Richiesta anche la sitemap.xml da poter inviare a principali motori di ricerca.

### **Principali funzionalità**

Home page:

Galleria di immagini recuperate dal database con bottoni prev e next.

Sezione delle ultime 6 news modificabili nel modulo di gestione.

Nella pagina dei dati del territorio si richiede una visualizzazione in tabella di dati generici organizzati secondo un aggregatore.

Contatti avrà un modulo di racconta email, oggetto e richiesta i quali verranno inoltrati ad una mail.

### **Testi e immagini**

Relativi testi ed immagini devono essere forniti dal cliente.

### **Configurazione hardware/software**

Avendo scelto come primo hosting AlterVista, il sito verrà sviluppato in php con MySQL come database.

Per le eventuali limitazioni si rimanda alle condizioni contrattuali dello spazio gratuito fornito da AlterVista.

### **Attività SEO-SEM, Web Marketing**

Come da richiesta verrà integrato il robots.txt che limita l'indicizzazione per la pagina login.php.

Verranno integrate le keyword concordate in riunione oltre che rispettati i principali canoni di attività SEO.

Infine verrà sviluppata la sitemap.xml da poter consegnare ai principali motori di ricerca per indicizzazione.

### **Assistenza e formazione post vendita**

Comprensiva della fornitura è prevista una giornata di formazione al personale redattore. Ulteriori giornate verranno concordate e fatturate a consuntivo.

## **Dettagli tecnici di implementazione (Per progetto universitario)**

### **File “application.setting.ini”**

File di configurazione sito, qui devono essere modificati i parametri per connettersi al database. Tale file viene letto ed interpretato dal configuration.php il quale ottiene la risorsa effettua un parse che valorizza la variabile \$configuration.

Tale istanza viene fornita alla pagina **db.php** che utilizza il metodo get per recuperare i parametri e creare la connessione.

Aspetti rilevanti di tale pagina sono la funzione connect che permette di ottenere con questa struttura il pattern Singleton fornendo la risorsa al database in return se già attiva.

Le ulteriori funzioni sono state implementate per essere più user-friendly e per una gestione degli errori che evita il lancio di eccezioni incontrollate (Struttura analoga per la gestione degli errori si troveranno all'interno di tutte le entità).

### **File “dati.php / gallery.php / news.php / user.php”**

I sopracitati file non sono altro che la rappresentazione in oggetti del mio database. Essi implementano nient'altro che le principali operazioni CRUD che ogni oggetto deve presentare.

Una particolare attenzione è stata dedicata alla classe user il quale permette di effettuare il CheckLogin (richiamato da login.php) che si occupa di verificare l'autenticità di chi effettua il login e creare l'eventuale sessione.

Tale sessione verrà riportata nel database e, attraverso la funzione CheckAuthentication, l'utente verrà riconosciuto se si trova in una pagina protetta (manage.php)

### **File “AjaxDati.php, AjaxNews.php”**

Sono due file, rispettivamente per le entità dati e news che mi permettono di restituire l'entità selezionata sotto oggetto json.

### **File “startup.php”**

Le attività principali di questa pagina è istanziare la connessione che successivamente verrà fornita agli oggetti e implementare il pattern Lazyload attraverso la funzione \_\_autoload()

Tale funzione include le classi automaticamente quando queste vengono richieste attraverso l'operatore "New NomeOggetto()". Questa semplice implementazione vincola che le classi si devono chiamare come il file e che devono essere tutte nella cartella lib/.

## **Manage.php**

Utente di test: "admin" "Password1"

Effettuata l'autenticazione si viene riportati su questa pagina qui che permette di gestire le immagini caricando un nuovo file o cancellando uno vecchio.

Una particolare attenzione è stata dedicata per i dati e per le new in quanto il "Load" effettua una chiamata \$.getJSON alle pagine sopracitate.

Questa funzione è offerta dalla libreria jquery il quale effettua una chiamata ajax e in caso di esito positivo effettua un json decode valorizzando la var "data" (contenuta nella function). Questo mi permette di accedere alle proprietà come nel seguente esempio "data.valore".

## **Contatti.php**

In questa pagina è stata implementata la mappa attraverso i metodi forniti dalla libreria leaflet che recupera i dati dal servizio openstreetmap.org

E' stato implementato come test il metodo per l'invio delle mail, ma per una evitare errori a video in quanto funzione mail() non configurata è stato implementato il controllo if(1==1)

La validazione della mail è stata effettuata utilizzando la funzione di php filter\_var con option = FILTER\_VALIDATE\_EMAIL.

## **Index.php**

All'interno del modulo è stata integrata la galleria come da richiesta, e fornita nel materiale di studio. Unica piccola modifica sono i bottoni di prev e next che stilizzati invisibili con top o bottom 0 per una più facile fruizione.