

1 2 3 Coupon!

Sito WEB di offerte promozionali di prodotti e servizi

Docente: prof. Alessandro Cucchiarelli

UNIVERSITÀ POLITECNICA DELLE MARCHE

16 giugno 2017

Gruppo 06

Bruno Davide 15%

Cameli Alessandro 40%

Micek Radoslaw Daniel 5%

Pioli Martina 40%

1 2 3 Coupon!

Sito WEB di offerte promozionali di prodotti e servizi

Descrizione del sito

Il nostro sito WEB ha l'obiettivo di permettere a degli utenti iscritti di acquisire dei coupon, associati a delle offerte, presentate dalle aziende e/o dai negozi associati con il nostro sito, per poi poterne usufruire al momento dell'acquisto vero e proprio.

Inoltre, se non si è registrati, è comunque possibile vedere i nostri contenuti ed è data la possibilità di registrarsi a chiunque voglia acquisire dei coupon.

Se una nuova azienda vuole entrare a far parte della nostra "community" gli basterà contattare gli amministratori del sito, i quali daranno la possibilità ad un utente scelto tra quelli già inseriti, o creato appositamente, di poter effettuare la CRUD delle promozioni da una specifica area riservata. Tale utente farà parte dello staff.

Anche l'utente admin avrà una sua area riservata, molto più complessa, dove potrà inserire, modificare, eliminare delle aziende, categorie, faq, inserire un nuovo utente staff o eliminare e modificare utenti.

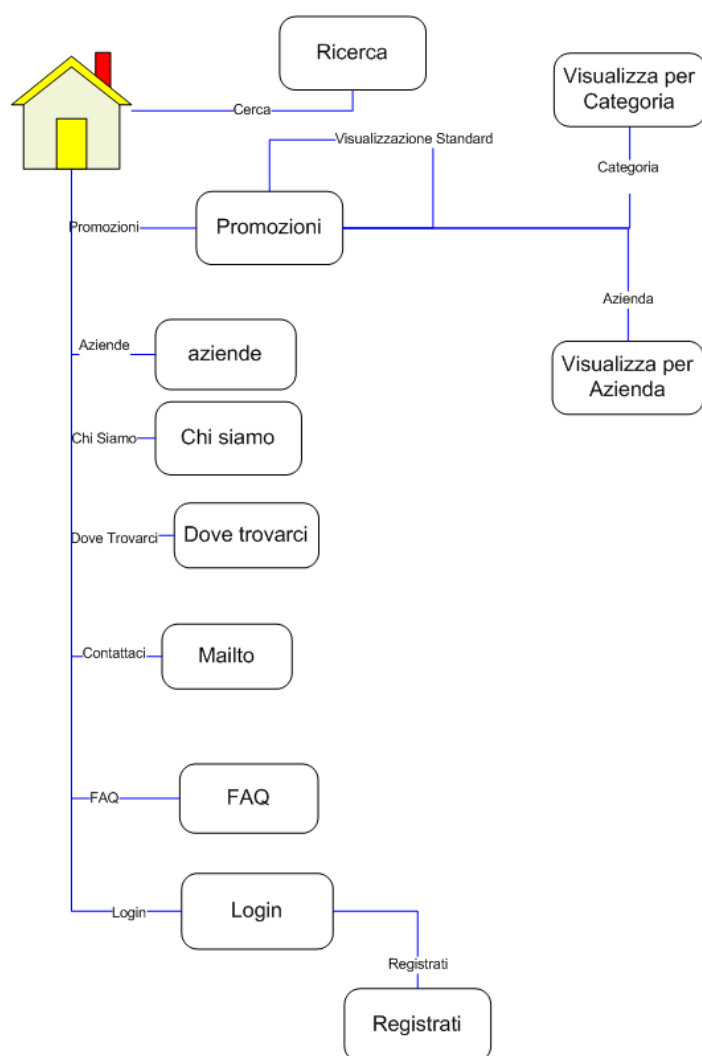
Schema di link

NOTE:

- Nel nostro schema di link i collegamenti sono organizzati nel seguente modo:
il nodo corrisponde alla pagina alla quale l'ancora è collegata, mentre i collegamenti rappresentano le ancore stesse.
- In ogni pagina del nostro sito è possibile tornare alla Home cliccando nel logo in alto a sinistra.

Nello schema di link abbiamo considerato i vari livelli di utenza, facendone uno per ogni tipo di utente.

Livello PUBBLICO



Il livello pubblico è quello in cui ci si trova al momento dell'apertura del sito, alla sinistra c'è il menù di navigazione, nel quale sarà possibile accedere alle pagine relative alle promozioni, aziende, faq, chi siamo, dove siamo, alla pagina per il login ed il link "contattaci" ci permetterà di inviare direttamente un'email all'amministratore.

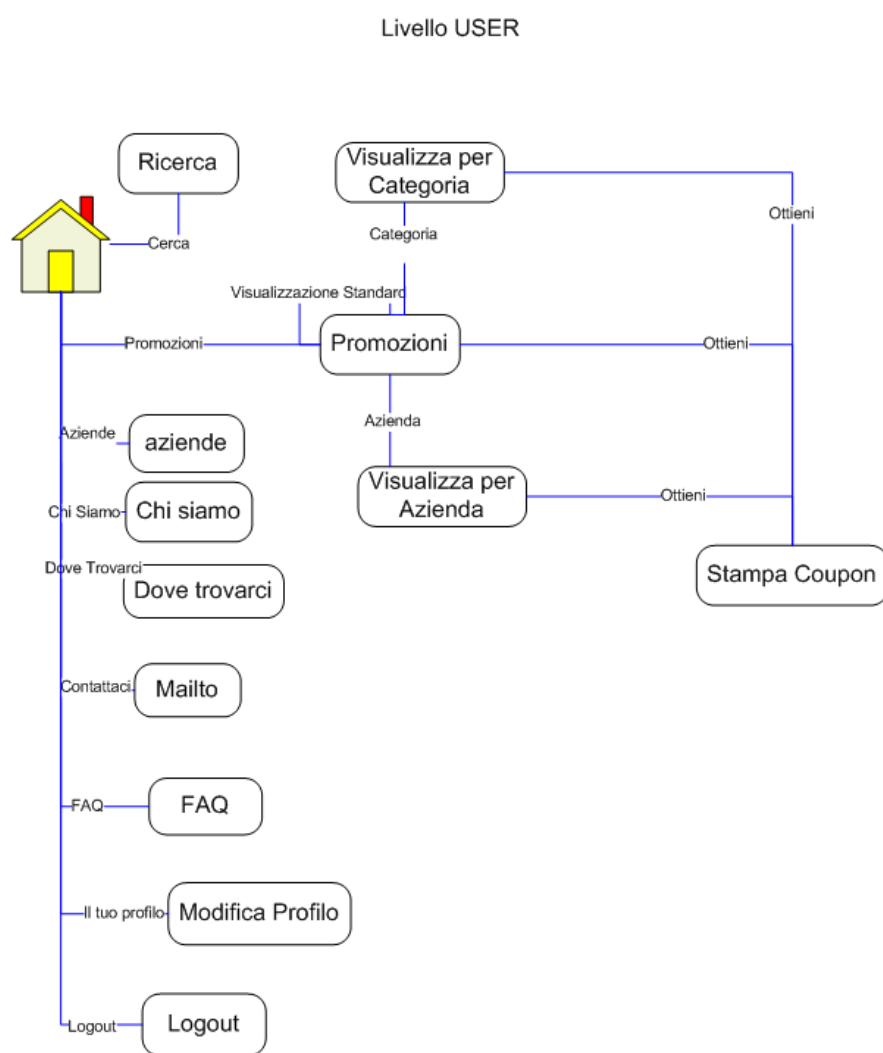
La prima pagina sarà proprio quella che permette di visualizzare tutte le promozioni, le quali potranno essere ordinate per categoria, azienda oppure con la visualizzazione standard. Cliccando infatti su "categoria" o "azienda" verrà fatta una divisione delle promozioni in gruppi e sarà possibile estendere un sottogruppo alla volta.

Nella header c'è una form per la ricerca delle promozioni, sarà possibile inserire la categoria e/o una o più parole.

Nella pagina “chi siamo” saranno descritte in breve le funzionalità del sito mentre nella pagina “Dove trovarci” c'è un iframe con la mappa presa da google, con indicato il luogo, in questo caso la sede dell'univpm.

Cliccando nel link “login” si potrà effettuare l'accesso, se si ha già un utente registrato con le proprie credenziali, oppure creare un nuovo account cliccando su “Registrati”. In questo modo si verrà direzionati alla pagina con la form relativa.

Al momento dell'inserimento dei dati per l'autenticazione, il sistema controlla che siano esatti ed estrae il ruolo di quell'utente.



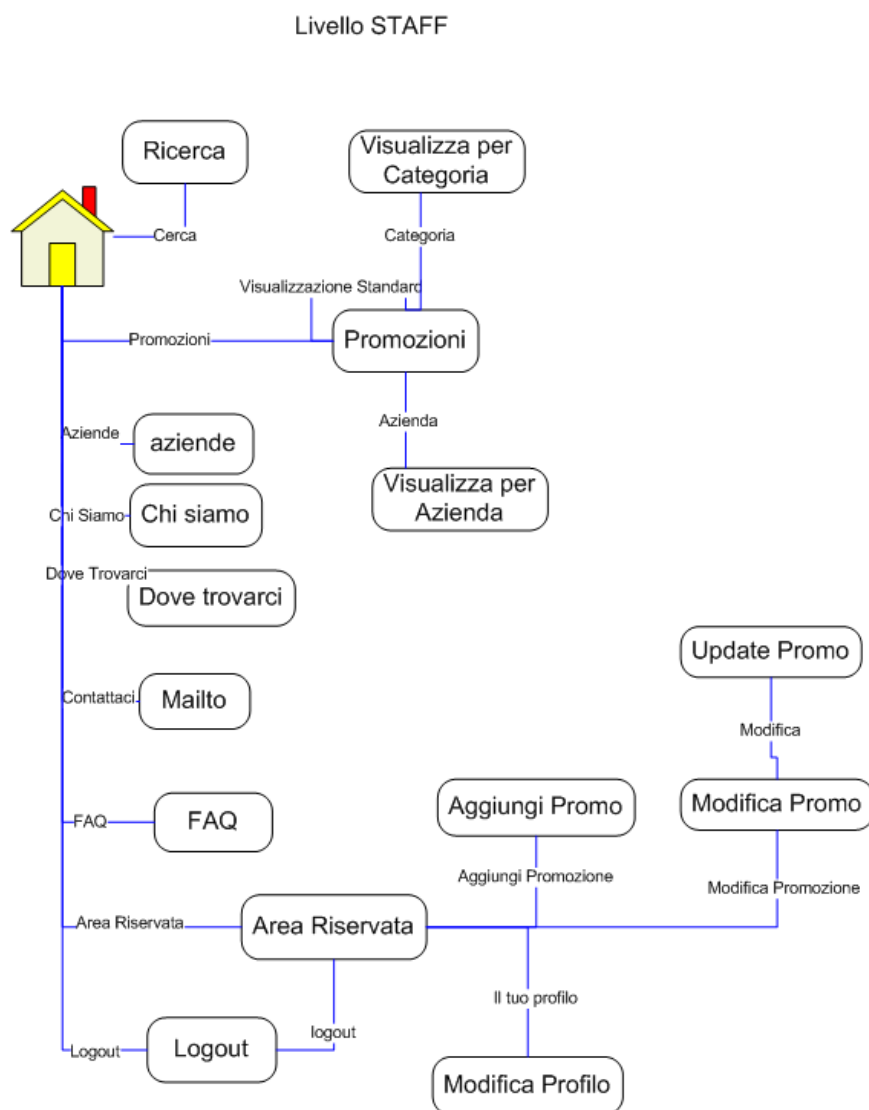
Al momento del login l'utente verrà rimandato ad una pagina Home dove potrà scegliere l'operazione da effettuare.

Il livello user estende il livello pubblico, troveremo quindi un menù simile, gli unici cambiamenti saranno i seguenti: al posto del bottone login, troveremo logout, e avremo un link alla pagina “modifica profilo”.

Il bottone logout sarà presente nel menù di ogni utente autenticato, qualsiasi sia il suo livello, fare click su questo bottone, terminerà la sessione corrente e riporterà, l'utente alla home del livello pubblico.

Facendo click su “Modifica Profilo” verrà caricata una form, già popolata con i dati dell’utente che ha fatto l’accesso in quel momento, e ci sarà la possibilità di cambiare i dati anagrafici e la password.

Una volta fatto l’accesso come utente sarà possibile acquisire dei coupon. Apparirà infatti su ogni visualizzazione delle promozioni un bottone “Ottieni” che, quando premuto, attiverà automaticamente la procedura di salvataggio di un nuovo coupon. Nella tabella ne verrà inserito uno nuovo con l’id dell’utente che l’ha acquisito e l’id della promozione nella quale si è cliccato. Inoltre verrà visualizzata una pagina “Stampa Coupon” che si potrà stampare per poi mostrare al momento dell’acquisto in negozio.

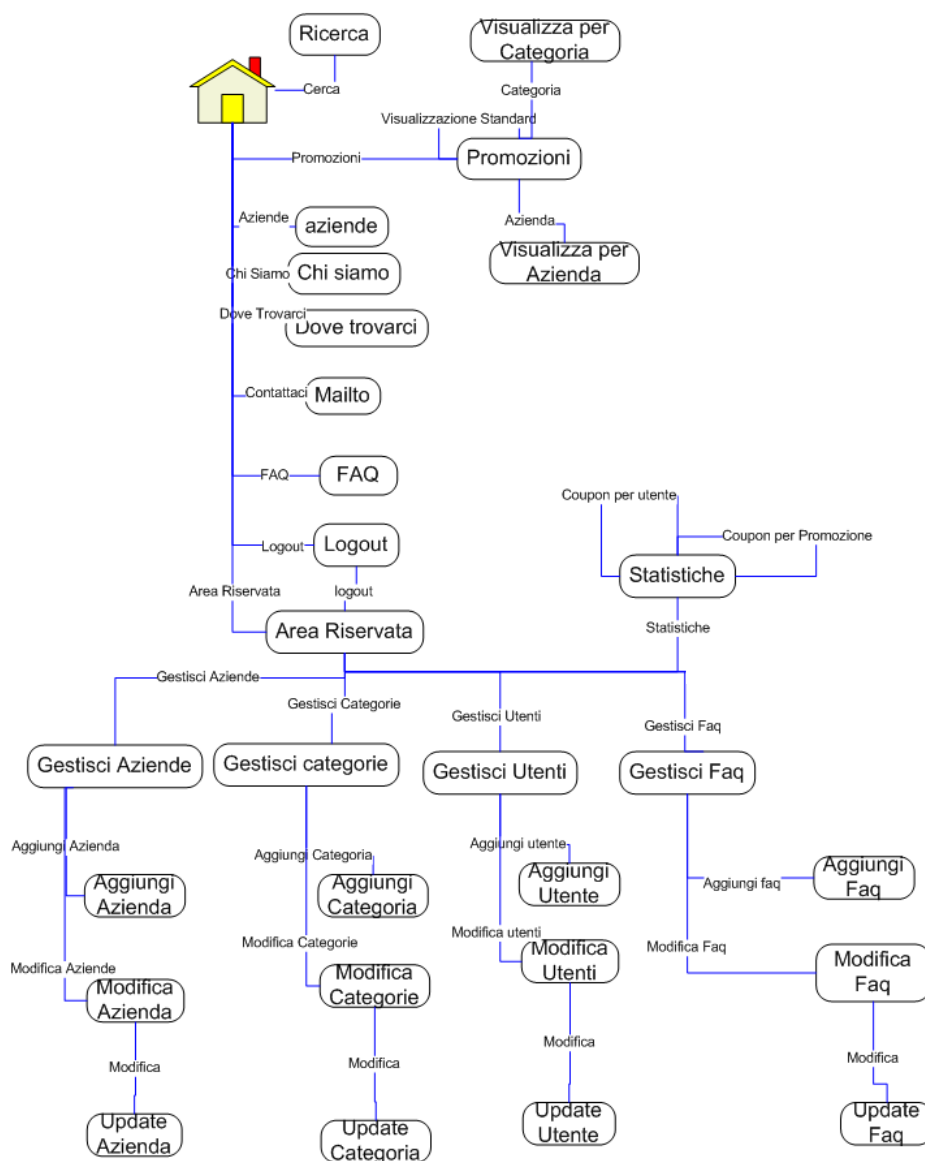


Il livello staff, come il livello user, estende quello pubblico. Al momento dell’autenticazione verrà visualizzata la pagina dell’area riservata che avrà un menù, attraverso il quale sarà possibile modificare il profilo utente secondo la stessa procedura dell’utente base, aggiungere una nuova promozione attraverso “Aggiungi Promozione” e modificare o eliminarne una attraverso “Modifica Promozione”.

In quest’ultima pagina di modifica, verrà visualizzata una tabella con tutte le promozioni inserite nel sito. Alla fine di ogni record sarà possibile scegliere se modificare, si aprirà una form specifica secondo la stessa procedura della modifica utente, oppure eliminare una determinata promozione.

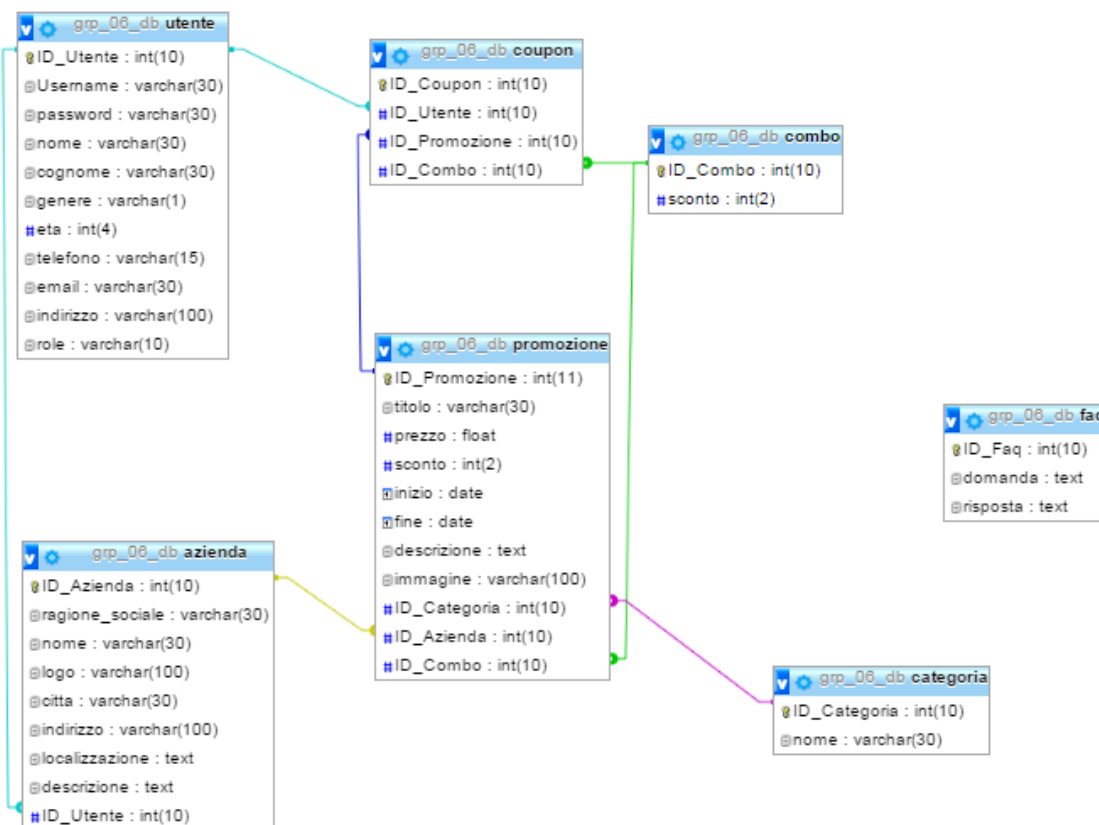
Sarà inoltre possibile tornare al menù principale, simile a quello pubblico, cliccando nell'ancora "Home". Varierà solo per la presenza di: "Area Riservata", la quale farà parte anche dello stesso menù per l'admin.

Livello ADMIN



Al momento dell'autenticazione con credenziali admin, l'utente verrà rimandato alla pagina principale dell'area riservata. Ci sarà un menù specifico formato dalle ancore che porteranno alla gestione CRUD delle aziende, categorie, faq, utenti oltre che alla visualizzazione di alcune statistiche del sito. Le statistiche permetteranno all'admin di vedere quanti coupon sono stati emessi in totale, quanti emessi per ogni promozione e chi, fra gli utenti registrati, ha ottenuto coupon e quanti.

Diagramma ER



Il database l'abbiamo sviluppato con MySQL. Abbiamo inserito le tabelle sopra riportate con i relativi campi attenendoci alle specifiche del progetto.

La tabella combo, che avevamo aggiunto per un'eventuale opzione aggiuntiva che ne richiedeva l'utilizzo, non viene mai utilizzata.

Tecnologie utilizzate per la realizzazione del progetto

- Linguaggio **HTML**, utilizzato per la realizzazione della struttura delle pagine web;
- Linguaggio **CSS**, utilizzato per le decorazioni della pagina web, per gestire i colori e i posizionamenti all'interno delle pagine. Non abbiamo mai aggiunto codice di questo tipo all'interno dei documenti html, ma ogni attributo è scritto nell'apposito file style.css all'interno alla cartella public/css;
- **JavaScript** e **jQuery** utilizzati per la validazione in modalità asincrona di tutte le form, attraverso una validazione **AJAX**. **jQuery** è stato impiegato anche per altre funzionalità che descriveremo in seguito;
- Codifica **JSON**, utilizzata per la validazione ajax. Quando il client fa la richiesta al server in modalità asincrona, quest'ultimo restituisce un file con la risposta in questa codifica, è molto più leggera dell'**XML** e facilmente leggibile dal client;
- **ZendFramework** è stata la nostra base per lo sviluppo del sito, attraverso il modello **MVC**. Tutte le funzionalità sono state implementate con questa tecnologia. Compreso anche il controllo degli accessi con **ACL**.

Soluzioni adottate

Ecco alcune soluzioni degne di nota, che abbiamo adottato per lo sviluppo del nostro progetto.

Registrazione nuovo account

Per la registrazione di un nuovo account, richiamiamo la form “Signin.php” e, al click del bottone “Registrami”, come prima cosa verrà mandata una query per cercare se quell’username inserito corrisponde già ad un account (per poter avere l’unicità del campo), in seguito verranno inviati i dati, salvati nel db, verrà effettuato automaticamente il login con le credenziali appena inserite. L’utente verrà rimandato alla pagina di Home dell’utente, con la scritta di Benvenuto.

Questo è possibile perché viene richiamata l’”authenticate” anche all’interno dell’azione “adduserAction” del “PublicControl”.

Validazioni AJAX

Abbiamo usato **AJAX** per la validazione asincrona delle form. Ci è servito cioè per poter dare all’utente un riscontro in tempo reale nei dati inseriti in quel momento. Il modello client-server implementato dal protocollo **HTTP** non ci permetterebbe di fare un’operazione del genere in quanto opera in modalità sincrona; cioè quando il client manda una richiesta al server, poi non può più fare niente finché non arriva la risposta che si aspetta. Usando **AJAX** invece è possibile scavalcare questo problema ma non solo. Grazie a delle funzioni **jQuery** possiamo controllare se il dato è valido quando la text-box in cui lo stavamo inserendo perde il focus. In quel momento infatti viene richiamata la funzione “doValidation” che si trova nel file public/js/functions.js, attraverso la quale verrà eseguito il procedimento di validazione vero e proprio attraverso **AJAX**. Il server restituisce un file di codifica **JSON** che sarà facilmente interpretabile.

La funzione **jQuery** ed il successivo richiamo della funzione “doValidation” sono state necessariamente implementate all’interno di ogni vista che richiama una form: login.phtml, signin.phtml (registrazione), retrieve.phtml (modifica profilo autenticato), newpromo.phtml, populate.phtml, newazienda.phtml, populateaz.phtml, newcategoria.phtml, populatecat.phtml, newfaq.phtml, populatefaq.phtml, newutente.phtml, populateusr.phtml.

Form di modifica

Per le form di modifica ci siamo fondamentalmente basati su una funzione che ci mette a disposizione **ZendFramework**: populate().

Per farlo abbiamo utilizzato la stessa form che avevamo creato per l’inserimento (usato per utenti, categorie, faq, aziende e promozioni, tutte con questo stesso procedimento). Vengono richiamate tre distinte action all’interno dello stesso controller. Cliccando dalla vista “modpromo” nell’icona relativa alla modifica di un campo, verrà attivata, ad esempio per la modifica delle promozioni, la “populateAction”.

A questo punto viene inviata una query per poter prendere i dati dal database, della promozione scelta dall’utente. Questi dati verranno convertiti in un array e, attraverso la funzione “getUpdatePromoForm”, verranno inseriti con la populate(), in una nuova istanza di form, uguale a quella per l’inserimento.

Dopo questo passaggio, la form compilata con i giusti dati viene visualizzata attraverso la vista richiamata automaticamente dalla “populateAction”.

Come ultimo passaggio c’è la “updpromoAction” che vede se i dati inseriti nella form sono validi, in caso affermativo viene richiamata la query per poter modificare quel campo all’interno della giusta tabella, altrimenti verrà visualizzato un messaggio di errore.

Ricerca

Un meccanismo molto importante quanto complesso da implementare è stato quello della ricerca.

Come prima cosa abbiamo dovuto instanziare un oggetto `searchForm` nella `init` del `PublicController`, così da poterla richiamare direttamente dalla `topbar` di `layout` che abbiamo preimpostato per l'utente pubblico.

La form è composta di 3 elementi: una `select` per la scelta della categoria, una `text-box`, per poter inserire una o più parole da ricercare nella descrizione o nel titolo, ed ovviamente il bottone `cerca`, per inviare i dati inseriti.

Al momento del click nel bottone `cerca`, viene attivata la `"searchAction"` dove verranno salvati in una variabile locale i dati inseriti per poter esser poi elaborati.

A questo punto viene richiamata la `"findAction"` che a sua volta richiama una funzione che si trova nel `model`, `"search"`, passando i valori della form insieme ai dati utili per il paginatore.

Nella `"search"` verrà deciso quale interrogazione fare al database, in base ai dati che sono stati inseriti, infatti c'è la possibilità di inserire una categoria, scrivere una o più parole, oppure completare entrambi i campi.

Se viene compilato solo il campo della scelta della categoria, verrà eseguita una query con la ricerca per `ID_Categoria`, se viene compilata solo la `text-box` verrà eseguita una query per la ricerca delle parole inserite sia nella descrizione che nel titolo, infine, se saranno inseriti entrambi, abbiamo implementato la chiamata di una query che le comprende tutte e due: `"fullSearch"`.

La ricerca per parola, è case-insensitive.

Se l'interrogazione effettuata avrà esito, negativo, ovvero non esistono corrispondenze con la ricerca nel database, viene visualizzata la scritta: `'nessun risultato trovato'`

Una delle cose più difficili in questa parte è stata quella della gestione del paginatore: ogni volta bisognava tenere conto dei dati inseriti, quindi nel file `/paginator/_pagSearch.phtml` abbiamo dovuto passare ad ogni link anche questi dati.

Per poter provare questa funzionalità del paginatore basterà ricercare le promozioni per la categoria `"elettronica"`.

Bottone ottieni sulle promozioni

Il bottone ottieni nelle promozioni diventa visibile solo quando ha effettuato l'accesso un utente base, se sarà autenticato un admin o uno staff non sarà disponibile. Abbiamo fatto questa scelta in quanto è data la possibilità solo a un utente base di acquisire dei coupon.

Per implementare questa funzione abbiamo utilizzato un view-helper: `"Ottieni.phtml"`.

Tale file ci permette di instanziare un'applicazione `Auth` attraverso la quale potremmo estrarre il ruolo dell'utente che è autenticato, se esiste un utente autenticato.

Nel caso in cui ci si trovi ancora nella parte pubblica, cioè nessuno ha ancora effettuato l'autenticazione, verrà visualizzata l'ancora per la pagina di autenticazione.

Recupero del menù e della topbar relativa all'utente autenticato

In base all'utente che è autenticato in quel momento nel sito, dovranno essere caricati un menù ed una topbar diversi.

Questo viene gestito nella `init` del `PublicController`, ogni volta che si ritorna nella Home principale viene attivata quest'azione e attraverso un `"if"` verrà controllato se è autenticato un utente, ed in caso affermativo, viene estratto il ruolo che verrà usato per redirezionare l'applicazione a prendere il giusto menù e la giusta topbar che si troveranno nella cartella nominata con gli stessi nomi dei ruoli.

Tutto questo è possibile grazie alla funzione `getIdentity()`.

Modifica Promozione\Azienda: problema immagine

Per motivi di sicurezza non possiamo popolare il campo immagine della form, è stato inserito un controllo nell' 'updatePromoAction', in modo che se il campo immagine non è stato settato durante la modifica della promozione, prima di procedere all'update viene ricaricato il valore precedentemente presente nel database, altrimenti il valore verrebbe aggiornato come stringa vuota, perdendo di fatto il nome dell'immagine, così che successivamente non potrà più essere visualizzata.

Questo controllo libera l'utente dall'obbligo di reinserire l'immagine ogni volta che modifica una promozione.

Lo stesso controllo è stato implementato per la modifica delle aziende.

Inserimento data promozione

Per implementare il corretto inserimento della data per le promozioni, ci siamo serviti della funzione **jQuery** 'datepicker()', definita nel file 'function.js', richiamata dalle viste: 'staff\populate.phtml' e 'staff\newpromo.phtml'.

Questa funzione istanzia gli input relativi alla data un datepicker, che mostrerà un calendario quando faremo click su uno dei due campi data.

per il campo 'data inizio' è stato inserito l'attributo minDate: 0, che fa in modo che non si possa inserire una promozione che inizia in una data precedente ad 'oggi', è inoltre associata un'azione all'evento onSelect, che permette di impostare per il campo 'data fine' come valore minimo quello selezionato per la data di inizio.

Righe a colori alterni delle tabelle

In ogni vista contenente una tabella di modifica, viene richiamata la funzione **striper()**, definita nel file 'function.js', che applica, o meno la classe css 'striped', alternativamente agli elementi con tag <tr>

Visualizzazione promozioni per categorie/aziende e faq a scomparsa

Vengono usate rispettivamente le funzioni **togglePromo()** e **toggleFAQ()**, definite in 'jfunctions.js', facendo click su una categoria/azienda nella vista **ordered.phtml**, o sul titolo di una faq, vengono nascosti tutti gli altri elementi descrittivi eventualmente aperti, e nel caso delle promozioni, viene effettuato lo **slideToggle()** su tutte le promozioni associate a quella categoria/azienda, mentre per le faq viene effettuato sulla risposta alla domanda selezionata.

Stampa del coupon

Al click sul bottone stampa nella vista **cupon.phtml** è associata la funzione **stampaCoupon()** che nasconde dalla pagina i bottoni, e l'eventuale avviso per l'utente riguardante la stampa di coupon multipli per la stessa promozione, e viene richiamata la funzione **window.print()**, che permette di stampare la pagina web.

Note del sito

Ogni utente può acquisire un determinato coupon una sola volta, abbiamo deciso che se un utente vorrà ottenere un coupon che ha già ottenuto in precedenza, verrà stampato il coupon già emesso precedentemente con un messaggio in basso che avviserà il cliente che quel coupon è già stato emesso e che potrà usufruirne una sola volta.

Essendo infatti il codice del coupon univoco, abbiamo pensato che in questi casi potrebbe essere una buona idea far gestire la possibilità di fruizione direttamente dal negozio in cui il cliente andrà a comprare il prodotto/servizio. In questo modo l'utente potrà stampare quante volte vorrà il suo coupon ma potrà usufruirne una sola volta.