

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Informatica - Scienza e Ingegneria

*CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA GESTIONALE*

TESI DI LAUREA

in
SISTEMI INFORMATIVI L

***Analisi, progetto e realizzazione di un sito Web di E-Commerce B2B
e B2C multilingue: il caso Inovit S.R.L. e lo sviluppo di
emgcompany.it***

CANDIDATO:

Filippo Petrisano

RELATORE:

Chiar.ma Prof. Wilma Penzo

CORRELATORI:

Chiar.mo Prof. Fabio Grandi
Dott. Davide Lepri

Anno Accademico 2016/2017

Sessione I

Sommario

Introduzione.....	4
1 – L’e-commerce	7
1.1 – Cos’è l’e-commerce?	7
1.2 – Quanto è importante l’e-commerce?	8
1.3 – Ragioni di crescita dell’e-commerce	9
2 – Supporto tecnologico alla realizzazione di siti e-commerce	10
2.1 – I Web Server	11
2.1.1 – Cos’è un server e cos’è un client?.....	11
2.1.2 – I fondamenti del protocollo HTTP.....	11
2.1.3 – La scelta del Web server	12
2.2 – DBMS.....	13
2.2.1 – MySql	13
2.2.2 – Oracle Database	14
2.2.3 – Microsoft SQL Server	14
2.2.4 – IBM DB2	14
2.2.5 – SQLite.....	15
2.2.6 – Alternative	15
2.3 – Linguaggi di programmazione	16
2.3.1 – HTML	16
2.3.2 – CSS	17
2.3.3 – JavaScript e AJAX	18
2.3.3.1 – La libreria jQuery	19
2.4 – La programmazione lato server.....	20
2.4.1 – Java Enterprise Edition	20
2.4.2 – La suite Microsoft .NET.....	21
2.4.3 – Il linguaggio PHP	22
2.4.4 – I Framework.....	23
2.4.5 – Il pattern MVC	24
3 – Il caso analizzato	26
3.1 – Le aziende.....	26
3.1.1 – NoWhere S.R.L.	26
3.1.2 – INOVIT S.R.L.	28
3.2 – I cerchi in lega e i parametri di riferimento.....	29
3.2.1 – I materiali e i processi produttivi utilizzati per i cerchi in lega.....	32
3.3 – Le richieste del cliente e la definizione del contratto.....	34
3.3.1 – Il logo e le immagini dei prodotti.....	34
3.4 – Strumenti utilizzati.....	35
3.4.1 – Laravel	35
4 – Realizzazione del progetto	37
4.1 – DBMS.....	38
4.1.1 – Le specifiche progettuali.....	38
4.1.2 – L’importazione dell’esistente	39
4.1.3 – Script di importazione	41
4.1.4 – La struttura del database.....	42
4.2 – Back-end.....	44
4.2.1 – Le pagine del CMS.....	44
4.2.2 – Le funzioni del back-end	51
4.2.3 – Sitemap del back-end	52
4.3 – Front-end.....	55

4.3.1 – La scelta del template.....	55
4.3.2 – Le funzioni del front-end	56
4.3.3 – Sitemap del front-end.....	57
4.3.4 – Struttura della homepage.....	58
4.3.5 – La ricerca intelligente	62
4.3.6 – La pagina dei risultati di ricerca	66
4.3.7 – La pagina prodotto	69
4.3.8 – Il carrello	71
4.3.9 – Check-out.....	72
4.3.10 – PayPal	76
4.3.11 – Carta di credito	79
5 – Un sito Web di successo.....	80
5.1 – User-friendly	80
5.2 – SEO	81
Conclusioni.....	83
Bibliografia.....	85

Introduzione

L'evoluzione della tecnologia ha modificato i comportamenti degli individui e delle aziende nel processo di comunicazione, informativo e d'acquisto. Tali cambiamenti hanno creato nuove opportunità e un nuovo mercato con caratteristiche e bisogni differenti rispetto a quello tradizionale.

Le nuove dinamiche hanno fatto sì che le imprese rivedessero le loro procedure d'acquisto, considerando sempre di più Internet come vantaggio sia in termini di efficienza che di efficacia.

Il numero di utenti connessi alla rete è in trend positivo, la banda larga a dicembre del 2016 ha raggiunto il 67,4% delle famiglie italiane e il 97% delle imprese con più di 10 dipendenti¹.

Internet rende disponibili molte più informazioni rispetto a quelle offerte dall'acquisto "classico", permettere l'integrazione dei servizi e la customizzazione dei prodotti. Inoltre, consente di ridurre i costi, gli intermediari e il customer lead time.

Il primo passo effettuato dalle aziende è stato creare un sito istituzionale, ovvero un "biglietto da visita" online; successivamente hanno timidamente intrapreso un'integrazione dei servizi offerti all'interno del Web. Lo sviluppo del fenomeno Internet nelle dinamiche aziendali è stato ormai compreso dai manager che non lo vedono più come qualcosa di distaccato dal sistema azienda, ma come parte integrante delle attività aziendali e un vantaggio competitivo da sfruttare.

I primi siti di vendita non presentavano le caratteristiche di sicurezza attualmente disponibili ed evidenziavano criticità nella semplicità di utilizzo e nella rapidità d'acquisto. Inoltre, la complessità nella fase di programmazione da parte degli operatori rappresentava un limite notevole durante la progettazione e l'implementazione dei siti. Grazie allo sviluppo della tecnologia e all'utilizzo di

¹ Fonte: Istat - Cittadini e nuove tecnologie.
(<https://www.istat.it/it/archivio/cittadini+e+nuove+tecnologie>)
(<https://www.istat.it/it/archivio/143073>)

nuovi strumenti software, oggi è possibile ottenere prodotti *user-friendly*, più sicuri ed esteticamente attraenti per il consumatore.

Attualmente si ha una crescita costante dell'utilizzo di tali mezzi per la vendita, anche se procede a ritmi molto diversi da Paese a Paese. I motivi di tale diversità risiedono nelle abitudini d'acquisto degli individui e dipendono da quanto essi siano disposti ad effettuare acquisti a distanza.

Nonostante durante l'acquisto su Internet il consumatore percepisca un livello di sicurezza più basso rispetto all'acquisto in negozio, si denota ugualmente da anni la crescita del numero di compravendite effettuate in rete. Per soddisfare la domanda in aumento, sempre più aziende decidono di dedicarsi al commercio elettronico, inserendo nel proprio sito una sezione dedicata agli acquisti online.

Tra i limiti dei negozi fisici vi è lo spazio per l'esposizione dei prodotti. Su Internet tale spazio non è più un limite e ogni articolo a magazzino può essere in vetrina. L'offerta dei prodotti aumenta stabilmente e chiunque può trovare ciò che cerca, direttamente da casa, senza doversi recare appositamente in negozio, che spesso si trova a grandi distanze. Capita, infatti, che l'acquirente non riesca a trovare ciò che vuole dietro l'angolo, soprattutto nelle piccole città in cui la scelta è limitata dalla bassa presenza di negozi fisici.

L'oggetto di questo lavoro è stato lo sviluppo di un sito di e-commerce, partendo da un'analisi dell'azienda e dei suoi obiettivi.

Sempre più spesso si sente parlare di siti di e-commerce. Nel primo capitolo si discuterà di cosa sono e che importanza rivestono a livello aziendale, analizzando dati statistici riferiti all'Italia e all'Europa.

Nel secondo capitolo si affronterà la tematica degli strumenti tecnologici attualmente disponibili per lo sviluppo di un sito di e-commerce. I principali argomenti saranno i protocolli, i Web server, i linguaggi di mark-up, di programmazione e i relativi framework.

Il terzo capitolo offrirà una visione delle aziende interessate nel lavoro progettuale, dei prodotti messi in commercio dall'azienda cliente e delle richieste del cliente.

Il quarto capitolo riguarda lo sviluppo del progetto nella pratica e le procedure attuate nella progettazione e nella realizzazione della soluzione software. Si

partirà definendo la struttura del DBMS utilizzando le specifiche progettuali identificate a seguito dell'analisi aziendale e dalle funzionalità richieste del cliente. A questo punto sarà possibile implementare la struttura del database nel sito e importarvi le specifiche dei prodotti, selezionando e riorganizzando i dati secondo la nuova struttura del database appena creato.

La fase della realizzazione del progetto continua con la creazione del back-end, utilizzando come riferimento le richieste del cliente in termini di funzionalità, ma anche la semplicità di utilizzo.

Si può procedere quindi alla scelta del template per il front-end e allo sviluppo di quest'ultimo. Per il front-end saranno utilizzati accorgimenti al fine di rendere l'esperienza d'acquisto rapida e intuitiva, come il guest-checkout e un motore di ricerca intelligente dei prodotti.

La fase finale del progetto sarà l'implementazione del processo di check-out e di pagamento.

Il quinto capitolo offrirà le nozioni basilari per creare un sito Web che risulti efficace e semplice nell'utilizzo, oltre che facile da trovare tramite i motori di ricerca.

1 – L'e-commerce

1.1 – Cos'è l'e-commerce?

“L'espressione *commercio elettronico* indica in generale il processo che consente di acquistare o vendere un prodotto o un servizio in una rete elettronica, in particolare attraverso Internet.”[1].

Un concetto strettamente connesso all' e-commerce è la *transazione commerciale*, che può essere di tre tipi:

- il primo tipo è la transazione tra azienda e consumatore, ed è detto business-to-consumer (*B2C*). Il tipico esempio di e-commerce che effettua transazioni di B2C è Amazon, che vende prodotti al dettaglio agli utenti di tutto il mondo. Oltre ai prodotti fisici, nell'e-commerce B2C è possibile vendere anche servizi, come fa per esempio il sito match.com (siti per appuntamenti online);
- il secondo tipo di transazione è detta business-to-business (*B2B*) ed avviene tra aziende. Un esempio di azienda che pratica transazioni business-to-business online è Maersk (gruppo danese leader mondiale nel trasporto container, con oltre 25.000 dipendenti e che opera in oltre 125 paesi);
- il terzo tipo si è sviluppato più di recente. Si tratta delle transazioni consumer-to-consumer (*C2C*), che avvengono tra consumatore e consumatore. L'esempio più noto di sito che si occupa di transazioni del genere è indubbiamente eBay, che permette agli utenti di mettere all'asta prodotti rivolti ad altri utenti.

1.2 – Quanto è importante l'e-commerce?

L'e-commerce - in senso stretto - rappresenta solo una piccola percentuale del tempo speso in Internet dalle persone. Nonostante questo, in Italia ha avuto e sta avendo un riscontro positivo:

“Gli acquirenti online italiani hanno compiuto in media più di un acquisto al mese nel 2016. Il numero di acquisti online cresce in modo lineare da diversi anni e ha superato il livello di 76 milioni di transazioni per trimestre a febbraio 2017. Ogni 100 acquisti online durante la stagione natalizia del 2016, 59,5 hanno riguardato prodotti fisici e 40,5 servizi e beni digitali. Il rapporto tra prodotti e servizi è molto variabile durante i dodici mesi, con un picco a fine anno a favore dei prodotti. Negli ultimi tre anni, l'incremento medio annuo del numero di acquisti di prodotti è stato superiore a quello di servizi e beni digitali.[...] Oltre l'85% della spesa online degli italiani transita sui circuiti delle Carte: una carta di credito o una prepagata (si intende incluso anche PayPal)”²

Anche in Europa l'utilizzo dell'e-commerce è molto elevato, sia in termini di numeri di acquisti che di fatturato.

Di seguito saranno riportati alcuni dati relativi all'Europa e all'e-commerce per dare un'idea dello sviluppo attuale³.

Dati relativi all'Europa:

- la popolazione europea ha superato i 743 milioni di individui;
- 685 milioni di persone hanno un'età inferiore ai 15 anni;
- 516 milioni di persone usano Internet;
- 296 milioni di persone sono e-shoppers (il 43% della popolazione europea);

² Fonte: Netcomm - Consorzio del commercio elettronico italiano (<http://www.consorzionetcomm.it/ImagePub.aspx?id=1229606>)

³ Fonte: E-commerce foundation – Organizzazione mondiale indipendente dell'e-commerce. Italy european B2C e-commerce report 2016

- nel sud Europa (Italia inclusa) il fatturato dell'e-commerce è 50 miliardi di euro ed è in crescita di circa il 17% annuo;
- in media, un e-shopper spende 1540€/anno;
- la media europea di crescita del fatturato dell'e-commerce è 13,3%/anno.

1.3 – Ragioni di crescita dell'e-commerce

Da cosa è dettato lo sviluppo costante dell'e-commerce?

I driver che inducono gli utenti a effettuare acquisti sono:

- desiderio di una gamma di scelta più ampia (con riferimento a prodotti, offerte, prezzi e canali);
- maggior fabbisogno informativo (dettagliate informazioni sul prodotto, sulla sua disponibilità, sullo stato dell'ordine);
- necessità di supporto online interattivo;
- sparisce la necessità di recarsi fisicamente al punto vendita, spendere del denaro per arrivarvi. Inoltre, non bisogna più considerare il problema della ricerca di un parcheggio;
- eliminazione dei limiti di tempo (orari di apertura, periodo intercorrente fra l'invio dell'ordine e la consegna).

Un esempio dei benefici di cui gode il cliente è fornito da Expedia (expedia.com), una nota agenzia di viaggi online che offre la massima libertà al cliente e ampio potere decisionale, fornendo informazioni comparative e permettendo di contattare direttamente le compagnie aeree, gli alberghi o gli autonoleggi. Tali benefici sono riassunti nello slogan pubblicitario dell'azienda: "Immaginate di poter pianificare il vostro viaggio, acquistare i biglietti aerei, prenotare gli alberghi e comperare pacchetti last-minute da soli, quando più vi piace".

2 – Supporto tecnologico alla realizzazione di siti e-commerce

Prima di parlare degli strumenti necessari allo sviluppo di un sito di e-commerce è necessario definire cosa serve per creare un generico sito Web dinamico, ovvero un sito che permetta all'utente di interagire con lo stesso.

Mentre un tempo era necessario avere delle competenze di programmazione, oggi tutti possono creare il proprio sito Web dinamico, tramite l'utilizzo di alcuni software detti Content Management System (*CMS*), che molto spesso sono messi a disposizione direttamente online sul sito del produttore del CMS. In questo caso si ottiene un ambiente preconfezionato e standardizzato, senza effettuare alcuna installazione, che permette di gestire i contenuti, come ad esempio i post di un blog.

I CMS attualmente più utilizzati sono *Wordpress*⁴, *Joomla!*⁵ e *Drupal*⁶, che tramite l'implementazione di alcuni plug-in (es. *WooCommerce*⁷ per Wordpress, *Virtuemart*⁸ per Joomla) possono diventare valide alternative di sviluppo per un sito Web di e-commerce.



Figura 1 - Loghi rispettivi di WordPress, Joomla! e Drupal

⁴ <https://wordpress.com>

⁵ <https://www.joomla.com>

⁶ <https://www.drupal.org>

⁷ <https://woocommerce.com/>

⁸ <https://www.virtuemart.net/>

Invece, per soluzioni ad hoc è necessario rivolgersi a società specializzate nell'*information technology* (IT), che analizzeranno le esigenze dell'impresa per rendere concrete le necessità di business.

Anche in tal caso, le alternative di sviluppo progettuale software sono variegate ed è necessario scegliere quali strumenti software utilizzare, ovvero il Web server, il DBMS, il linguaggio di programmazione e i relativi framework.

2.1 – I Web Server

Ultimamente, con lo sviluppo della tecnologia e di Internet si sente spesso parlare di server e in particolare di Web server anche tra i non addetti al settore, quindi è doveroso discutere anzitutto di cosa sia un server.

2.1.1 – Cos'è un server e cos'è un client?

Approcciamoci ai Web server partendo dal paradigma client-server, su cui si basa la comunicazione Web.

“Secondo questo paradigma, la computazione è distribuita su almeno due nodi: uno, definito client, richiede un servizio a un altro nodo, definito server, utilizzando lo stesso linguaggio comune detto protocollo. Il server risponde [...] indicando l'esito della richiesta e allegando eventuali risorse fornite come risultato.” [6].

Il protocollo usato dal paradigma client-server per il Web è HTTP.

I servizi consistono principalmente in richieste di pagine Web da parte del client, che si identifica nel *browser Web* (il software usato per la navigazione su Internet).

2.1.2 – I fondamenti del protocollo HTTP

Prima di poter parlare di Web server, bisogna affrontare un'introduzione sul protocollo utilizzato per lo scambio di dati.

L'*Hyper Text Transfer Protocol* (*HTTP*) è il protocollo sul quale lavora il Web. Ogni Web browser implementa questo protocollo per scambiare informazioni con il server.

HTTP definisce 9 tipi di metodi, ma quelli comunemente utilizzati sono *GET* e *POST*. *GET* è il metodo con cui vengono richieste la maggior parte delle informazioni ad un Web server. È possibile inviare dei parametri in query string, cioè la parte di URL dopo il punto interrogativo

Es. sito.com/pagina?id=123&page=3

In questo caso, i parametri sono *id* e *page* e i rispettivi valori sono 123 e 3.

Il metodo *POST*, invece, consente di inviare i dati senza mostrarli in query string.

Il protocollo *HTTP* è un protocollo senza stato (*stateless*): la connessione viene chiusa una volta terminata lo scambio richiesta/risposta e non vengono mantenuti dati di sessione o altro.

2.1.3 – La scelta del Web server

La scelta del Web server, per gestire le comunicazioni *HTTP* e *HTTPS* non è una scelta ardua, poiché di solito si ricade su *Apache 2*⁹, di tipo open source e compatibile con la maggior parte dei sistemi operativi in commercio.

Esistono anche altri Web server meno utilizzati, ma pur sempre molto affidabili. Alcuni esempi di ottimi prodotti con licenza BSD (software libero) sono: *lighttpd*¹⁰, server molto leggero sviluppato dalla Lighttpd Developers e *nginx*¹¹ (si pronuncia "engine-x"), scritto in C e C++.

⁹ <https://httpd.apache.org/>

¹⁰ <https://www.lighttpd.net/>

¹¹ <https://nginx.org/en/>

2.2 – DBMS

Il *database* (o *base di dati*, abbreviato anche in *DB*) è un insieme di dati omogeneo per contenuti e per formato, memorizzati in un elaboratore elettronico. Risulta essere il cuore dei moderni siti Web dinamici e si identifica con il *DBMS* (DataBase Management System), che rappresenta il sistema software per creare, manipolare e interrogare database.

Il linguaggio standard de facto per la creazione, l'interrogazione e la manipolazione nei DBMS relazionali è *SQL* (Structured Query Language) e vi è una vasta gamma di prodotti software che implementano tale linguaggio.

Inoltre, è necessario decidere se utilizzare un DBMS open source o acquistarne uno con licenza proprietaria.

Di seguito saranno riportati alcuni tra i più famosi DBMS utilizzati al mondo.

2.2.1 – MySql

È doveroso cominciare parlando di *MySql*¹², il “*DBMS open source più utilizzato al mondo*” [7]. È un DBMS di tipo relazionale (*RDBMS*).

La peculiarità dei RDBMS è la struttura del modello di memorizzazione dei dati. Questi, nei RDBMS sono memorizzati in tabelle aventi uno o più *attributi* o campi dato e un *valore* per ciascun attributo all'interno del dominio o tipo di dato consentito.

Un concetto importante è la *tupla* (*record* o riga), che rappresenta l'insieme non ordinato di valori assunti dagli attributi.

L'esempio classico è la tabella agenda telefonica, dove gli attributi sono nome, cognome e numero di telefono e i valori assunti da una delle righe potrebbero essere “Mario”, “Rossi” e “+3933333848”.

Rilasciato per la prima volta nel 23 maggio 1995, è divenuto proprietà della Sun Microsystems, che a seguito è stata acquisita dalla Oracle Corporation. È supportato da numerosissimi linguaggi di programmazione, come Java, .NET,

¹² <https://www.mysql.com>

PHP e Python (alcuni dei quali saranno discussi in seguito), ed è composto da un client a riga di comando e un server.

Può essere installato su sistemi operativi *Unix*, *Unix-like* e *Windows*.

Esistono vari *fork* (versioni modificate) di MySQL, come *MariaDB*¹³, *Drizzle*¹⁴ e *Percona server*¹⁵, ma risultano essere meno utilizzati rispetto al “genitore”.

2.2.2 – Oracle Database

*Oracle Database*¹⁶, anche questo prodotto dalla Oracle Corporation, è “il DBMS più diffuso al mondo” [7]. In questo caso si tratta di una suite software con licenza proprietaria. Differisce da MySQL per datatype, funzioni, strutture e tante altre caratteristiche. Solitamente è utilizzato per applicazioni enterprise di grosse dimensioni.

2.2.3 – Microsoft SQL Server

Anche Microsoft mette a disposizione il proprio Database Management System, in questo caso con licenza privata. *Microsoft SQL Server*¹⁷ risale al 1989, mentre l’ultimo rilascio RTM è avvenuto nel giugno 2016 con la versione 13.0. Inizialmente era utilizzato per piccole basi di dati, in seguito è stato utilizzato anche per gestire database di grosse dimensioni.

2.2.4 – IBM DB2

Oltre a Oracle, anche IBM offre una sua soluzione DBMS.

¹³ <https://mariadb.org/>

¹⁴ <https://launchpad.net/drizzle>

¹⁵ <https://www.percona.com/software/mysql-database/percona-server>

¹⁶ <https://www.oracle.com/it/database/>

¹⁷ <https://www.microsoft.com/it-it/sql-server/sql-server-2016>

La suite *DB2*¹⁸ è a pagamento, ma IBM dal 2006 distribuisce anche una versione gratuita detta “*Express-C*”, che limita l’utilizzo dell’hardware su cui risiede. L’attuale versione (10.5) permette di avere un database al massimo di 15TB, utilizza al massimo 2 processori e 16 GB di RAM.

2.2.5 – SQLite

È da tenere in considerazione anche *SQLite*¹⁹, libreria software gratuita, compatta (~500Kb) e molto veloce, che implementa al suo interno un DBMS. La particolarità di tale libreria è che, oltre alle applicazioni Web, supporta anche browser e istanze di vari software.

2.2.6 – Alternative

Come già detto, la scelta del DBMS può essere effettuata tra una grande varietà di prodotti e produttori. Per completezza è necessario accennare anche alla presenza di DBMS detti “non relazionali”, ovvero che non utilizzano la struttura classica a tabelle dei database relazionali. Tra questi, spicca *MongoDB*²⁰, software libero che utilizza documenti in stile *JSON* (JavaScript Object Notation) ed è utilizzato prevalentemente per la memorizzazione di documenti.

Per altre applicazioni particolari è possibile anche ricorrere a database basati sullo storage key-value. Il database di questo tipo più utilizzato è *Redis*²¹, open source con persistenza facoltativa.

¹⁸ <http://www.ibm.com/analytics/us/en/technology/db2/>

¹⁹ <https://www.sqlite.org/>

²⁰ <https://www.mongodb.com/>

²¹ <https://redis.io/>

2.3 – Linguaggi di programmazione

2.3.1 – HTML

Quando si parla di Programmazione Web, nonostante non si tratti di un linguaggio di programmazione, è bene partire dall'*HTML*²².

Come si intuisce dal nome, l'HTML (HyperText Markup Language) è un linguaggio di markup, vale a dire un insieme di regole che descrivono i meccanismi di rappresentazione del testo. Quindi, tramite una sequenza di caratteri si marcano gli elementi di un file di testo marcatori detti 'tag' per assegnare loro determinate caratteristiche o funzioni. Le regole sono stabilite dal World Wide Web Consortium (*W3C*), associazione che definisce gli standard del World Wide Web, e la versione attuale *HTML5* è stata rilasciata dal W3C nell'ottobre 2014. Tale rilascio è avvenuto a causa della necessità di fornire direttamente le funzionalità che in precedenza erano fruibili tramite estensioni proprietarie all'esterno dei browser, come Adobe Flash.

Sono state apportate molte modifiche rispetto ad *HTML4*, in particolare:

- l'aggiunta di regole stringenti per la strutturazione del testo in capitoli, paragrafi e sezioni;
- l'introduzione di elementi di controllo per i menu di navigazione;
- l'introduzione di elementi specifici per il controllo di contenuti multimediali (tag `<video>` e `<audio>`);
- settati come deprecati o eliminati alcuni elementi che hanno dimostrato scarso o nessun utilizzo effettivo;
- estensione a tutti i tag di una serie di attributi, specialmente quelli finalizzati all'accessibilità;
- viene supportata l'estensione *Canvas* che permette di creare animazioni e grafica bitmap;
- introduzione della geolocalizzazione, dovuta ad una forte espansione di sistemi operativi mobili più diffusi;

²² <https://www.w3.org/html/>

- sistema alternativo ai normali cookie, chiamato *Web Storage*, più efficiente, che consente un notevole risparmio di banda;
- sostituzione del lungo e complesso doctype, con un semplice `<!DOCTYPE html>`

2.3.2 – CSS

Il CSS²³ (Cascading Style Sheets) è un linguaggio di programmazione utilizzato per definire la formattazione dei documenti HTML.

La nascita del CSS è diventata un'esigenza quando ci si è resi conto che i produttori di browser sviluppavano tag propri per la formattazione del codice e questo era un problema per i webmaster, poiché la resa grafica era differente utilizzando client differenti e l'assenza di una standardizzazione non permetteva di poter sviluppare lo stesso codice per due browser differenti. Con la nascita dei palmari e degli smartphone aventi basse risoluzioni e differente rapporto d'aspetto dell'immagine, non era possibile visualizzare pagine create per essere mostrate su un computer. Inoltre, si sentiva la necessità di rendere il codice HTML più leggibile. Per questa ragione, W3C alla fine del 1996 rilascia le direttive (Recommendations) per comporre il CSS e separare il contenuto dalla sua formattazione.

Usando i fogli di stile è possibile controllare bordi, colori, margini, allineamenti, font e tante altre proprietà per ottenere l'effetto visivo desiderato.

La pratica migliore è posizionare i fogli di stile in un file esterno all'HTML che sarà richiamato dall'HTML stesso all'avvio del file, o tramite JavaScript (discusso in seguito) al verificarsi di un evento come il click del mouse. Inserendo i fogli di stile in file separati sarà possibile riutilizzare lo stesso CSS in più pagine. Il foglio di stile può essere inserito anche all'interno della pagina HTML, ma è necessario copiare ed incollare il codice da una pagina all'altra per riutilizzarlo. Diventa quindi difficile la sincronizzazione e l'aggiornamento.

Per questa ragione, il suddetto metodo è sconsigliato, a meno che non si tratti di modifiche grafiche da applicare solo alla pagina considerata.

Il terzo modo per inserire i fogli di stile all'interno dell'HTML è detto inline.

²³ <https://www.w3.org/Style/CSS/Overview.en.html>

L'inserimento di fogli di stile inline è un modo per inserire il linguaggio dei fogli di stile direttamente all'interno del tag di apertura di un elemento HTML.

Dopo aver detto che sarebbe sempre bene separare il contenuto dalla formattazione, può risultare strano sentirsi dire che il linguaggio CSS venga integrato nel contenuto, ma a volte risulta molto semplice e veloce l'inserimento del CSS inline per piccole modifiche che non saranno utilizzate in altre pagine. Gli elementi della pagina vengono individuati tramite l'utilizzo di "Selettori" che specificano uno o più elementi dell'HTML.

Esistono 3 tipi di selettori:

- *selettori universali*: utilizzati per gli elementi generici dell'HTML espressi nel codice html utilizzando tag, come `<h1>` o `<input>`
- *selettori di classe*: identificano gli elementi HTML aventi il nome di classe specificato all'interno dell'attributo class.
A titolo d'esempio `<input class="selettore_di_classe">`
- *selettori ID*: permettono di identificare in modo univoco un singolo elemento tramite una stringa testuale `<input id="selettore_ID">`

I colori all'interno della sintassi CSS possono essere espressi o in formato esadecimale o sotto forma RGB (Red Green Blue). Esiste anche un altro modo di esprimere i colori ed è utilizzando il loro nome in inglese, ma sfortunatamente questo sistema funziona soltanto con 16 colori predefiniti.

2.3.3 – JavaScript e AJAX

JavaScript è un linguaggio di scripting lato client, orientato agli oggetti e agli eventi, comunemente utilizzato all'interno del codice HTML di siti Web e applicazioni Web, per ottenere effetti dinamici interattivi tramite funzioni invocate da eventi innescati dall'utente sulla pagina Web in uso.

L'integrazione avviene direttamente nel file HTML racchiusa dal tag `<script>` o tramite appositi file separati con estensione ".js", richiamati inserendo nel tag `<script>` l'attributo `src` che abbia come valore l'*URL* del file (indirizzo univoco della risorsa online).

Nonostante Javascript sia usato prevalentemente per il Web, è possibile eseguirlo anche nel *VBS* (Visual Basic Script), che nella sua versione presente all'interno di Internet Explorer offre le stesse funzionalità di JavaScript.

Per questa ragione, il *VBScript* è spesso utilizzato anche dai cracker per creare virus avviabili su piattaforme Windows. Ne è un celebre esempio il virus *ILOVEYOU* che ha infettato con successo 10 milioni di computer come allegato di un messaggio e-mail avente come oggetto il testo “ILOVEYOU”.

2.3.3.1 – La libreria jQuery

*jQuery*²⁴ è una libreria JavaScript per Web Application, nata con l'obiettivo di semplificare la selezione, la manipolazione, la gestione degli eventi e l'animazione di elementi.

Il framework fornisce metodi e funzioni per gestire al meglio gli aspetti grafici e strutturali, come il posizionamento degli elementi, il click sulle immagini, la manipolazione del documento, la validazione lato client dei campi, l'esecuzione di animazioni e la modifica dei documenti di stile.

Inoltre, tra le API, è presente anche un gestore delle richieste *AJAX* (Asynchronous JavaScript and XML).

AJAX è una tecnica di sviluppo software che permette di effettuare lo scambio di dati in background tra client e server. Tramite *AJAX* è quindi possibile ottenere siti più interattivi, poiché lo scambio di dati avviene senza ricaricare la pagina. Inoltre, le richieste asincrone permettono di ottenere pagine più veloci, grazie alla possibilità di utilizzare meno banda, evitando di ricaricare tutta la pagina e aggiornando solo la parte interessata

²⁴ <https://jquery.com>

2.4 – La programmazione lato server

Un altro punto da scegliere quando si crea un sito dinamico è il linguaggio di programmazione utilizzato lato server. In questo caso l'URL richiesto dal browser punterà alla risorsa contenente il codice HTML farcito di frammenti di script. Tale codice è inserito all'interno dell'HTML utilizzando sintassi differenti che dipendono dal linguaggio ed usualmente è racchiuso da uno pseudo-tag.

Esempi di linguaggi di programmazione Web lato server sono PHP ed ASP, entrambi molto diffusi. È possibile usare anche JavaScript, ed eventualmente anche altri linguaggi (Ruby, Python, ecc.).

Un altro meccanismo è quello delle Java Server Page (JSP), che utilizza le servlet con una modalità simile allo scripting.

2.4.1 – Java Enterprise Edition

Java Enterprise Edition (*Java EE*²⁵) prevede, oltre alle caratteristiche di base del linguaggio incluse nella Standard Edition, una serie di tecnologie volte a creare applicazioni complesse client-server, non necessariamente basate sull'uso del Web.

Diversamente dai linguaggi di scripting, Enterprise Edition si basa su concetti orientati agli oggetti.

Gli elementi costitutivi sono i quattro seguenti:

- servlet: classe Java con metodi per la gestione delle richieste e delle risposte HTTP (in particolare, GET e POST)
- Java Servlet Pages (*JSP*): documenti in cui può essere presente codice Java mescolato all'HTML che viene compilato sul server. Si utilizza lo pseudo tag `<% %>` per racchiudere il codice che è compilato alla prima richiesta in una servlet
- Componenti software: *Java Beans* ed Enterprise Java Beans gestiscono funzioni specifiche e sono progettati per rendere semplice il riutilizzo. Nella versione Enterprise fruiscono di opzioni avanzate quali la persistenza degli oggetti e la distribuzione del carico.

²⁵ <http://www.oracle.com/technetwork/java/javaee/overview/index.html>

- **Contenitore:** il contenitore di servlet è un elemento software che lavora in cooperazione con un server Web, oppure lo incorpora. Le funzioni di un contenitore sono: decodificare le richieste dei client, la gestione del ciclo di vita delle servlet, gestione delle sessioni, supporto alla creazione della risposta http, distribuzione del carico sicurezza e persistenza.

Un'applicazione Web in Java è un insieme di file e directory disposti secondo uno schema standard descritto nella specifica delle servlet. L'insieme dei file contiene codice HTML, immagini, pagine JSP, classi Java (servlet e Java Beans) ecc., comunque sotto una directory radice ospitata nella directory webapps.

2.4.2 – La suite Microsoft .NET

Fino a qualche anno fa, l'approccio di Microsoft alle applicazioni Web consisteva nella tecnologia di scripting ASP, basata su *Visual Basic* e *JScript*. Negli ultimi anni ha però sviluppato un nuovo insieme di componenti destinati allo sviluppo di applicazioni Web, ma anche tradizionali, che va sotto il nome di *.NET*²⁶.

Microsoft .NET si basa su un framework che comprende:

- Compilatori per i principali linguaggi Microsoft
- Macchina virtuale
- Ampia libreria di classi.

L'approccio alla macchina virtuale ricorda quello di Java, così come il nuovo linguaggio di programmazione *C#* è simile a Java. Ciò che probabilmente fornisce il maggior pregio alla piattaforma è la libreria di classe, che semplifica molto la realizzazione di interfacce Web. Inoltre è presente un supporto notevole per i Web service.

²⁶ <https://www.microsoft.com/net>

2.4.3 – Il linguaggio PHP

Nessuno si aspettava che *PHP* sarebbe diventato tanto popolare com'è oggi, considerando che questo linguaggio vede timidamente la luce come progetto di un hacker per offrire una soluzione semplice



Figura 2 - Logo ufficiale PHP

e divertente per lo sviluppo di siti Web. Oggi conoscere la programmazione PHP è fondamentale per qualunque sviluppatore Web poiché attualmente “ha conquistato una quota variabile tra il 35 e il 59% dei linguaggi impiegati nello sviluppo Web” [5].

PHP è un linguaggio di programmazione orientata agli oggetti per lo sviluppo di applicazioni Web e mobile e la scalabilità delle sorgenti di dati. Si tratta di un linguaggio di alto livello fornito di moltissime API, oltre 3.000 funzioni del nucleo base.

L'interprete di PHP è distribuito sotto la licenza libera PHP License e l'ultima versione rilasciata è la 7.1.3.

PHP è in grado di interfacciarsi a innumerevoli database tra cui MySQL, MariaDB, Oracle, *Firebird* SQL (DBMS open source Mozilla) , IBM DB2, Microsoft SQL Server, ma anche con database di tipo NoSql come per esempio il già citato MongoDB, e supporta numerose tecnologie e protocolli (XML, SOAP, IMAP, FTP, CORBA).

Come tanti altri linguaggi moderni, è a tipizzazione debole (tipizzazione dinamica), ovvero le variabili possono riferirsi a valori di qualunque tipo, che possono cambiare dinamicamente in seguito a manipolazioni esterne.

Le variabili in PHP iniziano sempre con il simbolo del dollaro (\$) e il carattere successivo non deve essere un numero. I caratteri che si possono utilizzare per denominare una variabile sono A-Z , a-z , 0-9 e _.

Una volta dichiarate, esse vengono istanziate senza definirne esplicitamente il "tipo":

```
$hello = "Ciao mondo!";
```

Per mostrare il contenuto di \$hello, è possibile usare il comando echo:

```
echo $hello;
```

Tale istruzione visualizzerà a schermo “Ciao mondo!”.

Se la variabile \$hello non fosse esistita, echo avrebbe generato un errore non fatale (*Notice*).

PHP può essere eseguito anche da prompt dei comandi (in Linux o Windows) e pertanto può essere utilizzato in script pianificati senza interventi manuali (*CRON*).

2.4.4 – I Framework

Nello sviluppo di applicazioni spesso si ricorre all'utilizzo di strutture di supporto su cui un software può essere organizzato. Tale struttura di supporto è detta framework. Alla base di un framework c'è sempre una serie di librerie software che possono essere utilizzate da uno o più linguaggi di programmazione.

I framework di ultima generazione sono spesso corredati da una serie di strumenti di supporto allo sviluppo del software, come ad esempio il debugger²⁷.

Lo scopo di un framework è di far risparmiare tempo allo sviluppatore nella riscrittura di codice già steso per compiti simili.

Questa circostanza si è presentata sempre più spesso adesso che le interfacce utente sono diventate sempre più complesse, o più in generale man mano che è aumentata la quantità di software con funzionalità secondarie simili.

L'esempio classico è il menu a tendina, che per un'esperienza utente ottimale dovrebbe rimanere sempre uguale. Tramite un framework può essere inserito con poche righe in una o più pagine.

Un framework è definito da un insieme di classi astratte e dalle relazioni tra esse, quindi istanziare un framework significa fornire un'implementazione di queste.

²⁷ *Debugger*: strumento software progettato per analizzare e correggere gli errori software

Invece l'insieme delle classi concrete, proprie dell'applicazione creata, saranno definite ereditando quelle astratte del framework assieme alle loro relazioni.

In questo modo si ottiene un insieme di classi concrete collegate tra loro.

Ma cosa sono le librerie citate in precedenza?

In generale, una libreria software è un insieme di funzioni o strutture dati predisposte per essere collegate ad un programma software.

Lo scopo di una libreria software è quello di fornire una collezione di entità di base pronte per l'uso per ridurre il tempo di sviluppo. Un altro vantaggio che si può evincere è la possibilità di modificare le librerie separatamente dal programma, per ottenere output differenti dal programma stesso.

Tramite queste è possibile riutilizzare il codice, evitando al programmatore di dover riscrivere ogni volta le stesse funzioni o strutture dati e facilitando così le operazioni di manutenzione. Questa caratteristica si inserisce quindi nel più vasto contesto del "richiamo di codice" all'interno di programmi e applicazioni ed è presente in quasi tutti i linguaggi.

2.4.5 – Il pattern MVC

Il pattern Model-View-Controller (*MVC*, in italiano tradotto con Modello-Vista-Controllo) è uno schema architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti.

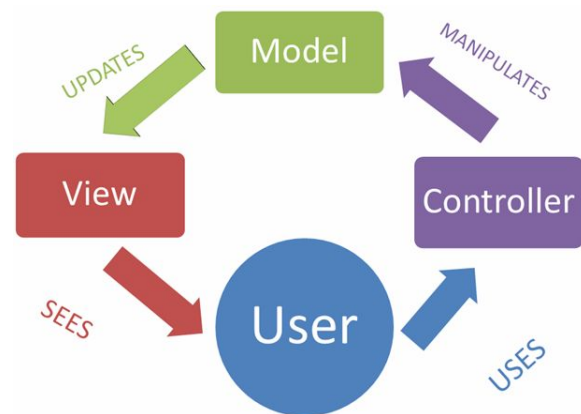


Figura 3 - Schema di funzionamento del pattern MVC

Il pattern è basato sulla separazione dei compiti fra i tre componenti software in figura:

- il *model* (*modello*) fornisce i metodi per accedere ai dati utili all'applicazione;
- la *view* (*vista*) visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;

- il *controller* (*controllo*) riceve i comandi dell'utente (in genere attraverso la view) e li attua modificando lo stato degli altri due componenti.

Storicamente il pattern MVC è stato implementato lato server. Recentemente, con lo sviluppo e la parziale standardizzazione di JavaScript sono nate le prime implementazioni lato client.

Ma affrontiamo il coding lato server, che è la parte davvero gravosa nello sviluppo di un sito web di e-commerce, o più in generale di una applicazione Web, in cui utilizzare framework MVC risulta la best practice.

Originariamente impiegato dal linguaggio *Smalltalk*²⁸, il pattern è stato esplicitamente o implicitamente sposato da numerose tecnologie moderne, come framework basati su PHP (*Symfony*²⁹, *Laravel*³⁰, *Zend Framework*³¹, *CakePHP*³², *Yii framework*³³, *CodeIgniter*³⁴), su Ruby³⁵ (*Ruby on Rails*³⁶), Python³⁷ (*Django*³⁸ ecc.), Java (*Spring*³⁹, *JSF*⁴⁰ e *Struts*⁴¹), *Swift*⁴² e .NET.

²⁸ <http://smalltalk.org/>

²⁹ <https://symfony.com/>

³⁰ <https://laravel.com/>

³¹ <https://framework.zend.com/>

³² <https://cakephp.org/>

³³ <https://www.yiiframework.it/>

³⁴ <https://www.codeigniter.com/>

³⁵ <https://www.ruby-lang.org/>

³⁶ <https://rubyonrails.org/>

³⁷ <https://www.python.org/>

³⁸ <https://www.djangoproject.com/>

³⁹ <http://projects.spring.io/spring-framework/>

⁴⁰ <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

⁴¹ <https://struts.apache.org/>

⁴² <https://developer.apple.com/swift/>

3 – Il caso analizzato

L'obiettivo di questo lavoro di tesi è stato sviluppare un sito Web di e-commerce, ovviamente dopo aver effettuato un'analisi dell'azienda cliente per conoscere le esigenze di business della stessa e soddisfarle in modo da ottenere un prodotto ad hoc.

Il lavoro è stato svolto presso INOVIT S.R.L., per conto di NoWhere S.R.L..

3.1 – Le aziende

3.1.1 – NoWhere S.R.L.

L'azienda in cui è stato svolto il tirocinio extracurricolare per la realizzazione finale del sito Web è denominata "*NoWhere S.R.L.*", con sede in Via Milazzo 24 a Bologna.

NoWhere S.R.L. è un'azienda indipendente attiva sul territorio da oltre 10 anni, specializzata in servizi Web-based e interattivi.

Attualmente vi lavorano 7 dipendenti, presenta un attivo di bilancio di circa 420.000€ (dati aggiornati al 2015).

Attualmente il presidente del consiglio di amministrazione è la Dott.ssa Giulia Sabbadini.

Come si può evincere dal grafico sottostante, si tratta di una società in crescita.

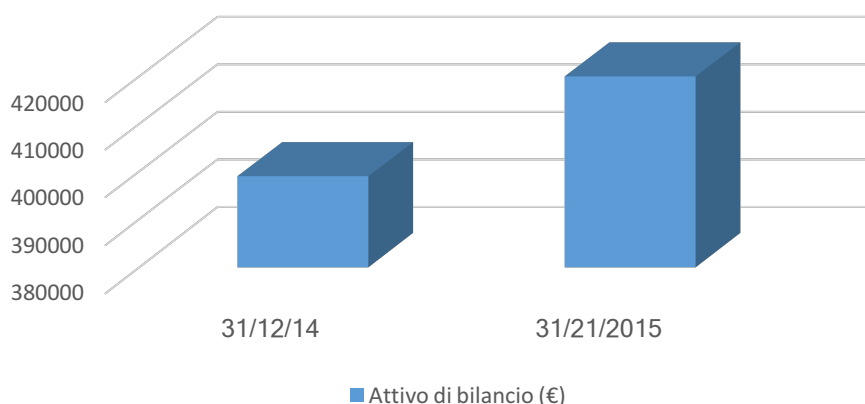


Grafico 1 Fonte: Bilanci ufficiali depositati alla Camera di Commercio di Bologna

Opera in un settore ad alto tasso di sviluppo e impiega personale qualificato e in continua formazione professionale.

Il clima che si respira in azienda è sereno e altamente collaborativo.

Fornisce vari servizi connessi all'IT, come:

- l'ottimizzazione e l'organizzazione dei flussi di lavoro;
- gestionali online;
- la realizzazione di siti di e-commerce;
- la progettazione e la realizzazione di Applicazioni Web e mobile;
- prodotti multimediali di vario genere.

La società si occupa inoltre di realizzazioni grafiche e audio-video e di soluzioni come il *Video Scribing* (un'attività di *storytelling* a grande impatto visivo ed emotivo utilizzata in ambiti come il marketing e l'apprendimento scolastico).

Il referente aziendale è stato il Dott. Davide Lepri, socio di NoWhere S.R.L. con responsabilità di co-amministratore, giovane, ma con una consolidata esperienza alle spalle. Il Dott. Lepri mi ha fornito i dati necessari a definire le specifiche progettuali e ha seguito costantemente l'evoluzione della realizzazione del progetto fino alla sua conclusione.

3.1.2 – INOVIT S.R.L.

Durante il primo incontro col referente in NoWhere S.R.L., dopo un primo colloquio conoscitivo, mi sono stati proposti diversi possibili lavori progettuali. La scelta è ricaduta su un progetto complesso di e-commerce per *INOVIT S.R.L.*, con sede a San Giorgio di Piano, una cittadina della provincia Bolognese. La INOVIT S.R.L. ha un totale attivo circolante di circa 355.000€ e un utile annuo di 232.723€⁴³.

Si tratta della filiale italiana di una multinazionale che attualmente rivende cerchi in lega per auto soprattutto a clienti di tipo *business* (le aziende), ma anche a una ristretta clientela di tipo *consumer* (i privati).

Dal mio punto di vista, nonostante fosse un progetto impegnativo, mi avrebbe permesso di applicare competenze acquisite durante il percorso di studi di ingegneria gestionale e la mia passione per la programmazione, interesse che ho coltivato fin da piccolo.

Da un'analisi condotta in loco assieme al Dott. Lepri, dopo una discussione con Pierluigi Greco, responsabile commerciale di INOVIT S.R.L., è emerso l'obiettivo prioritario dell'azienda cliente di creare un sito di e-commerce con particolari specifiche, che saranno approfondite nel prosieguo, per sviluppare anche il canale di vendita online.

Ci si è affidati alla figura di Pierluigi Greco per una spiegazione tecnica dell'impostazione della ricerca per i cerchi in lega sulla piattaforma, poiché, oltre a gestire interamente la parte commerciale, si occupa di consulenze tecniche sui prodotti.

⁴³ Fonte: Bilancio ufficiale di esercizio al 31-12-2015, depositato alla Camera di Commercio di Bologna

3.2 – I cerchi in lega e i parametri di riferimento

"La normativa europea definisce la ruota come un elemento rotante di sostegno del carico posto tra lo pneumatico e l'asse della vettura"[9].



Figura 4- Esempio di ruota con cerchio smontabile

La ruota è costituita da due parti:

- Il cerchio (chiamato canale)
- Il disco (chiamato flangia)

Se le parti costituiscono un unico elemento la ruota si definisce Monoblocco.

Se le parti sono due o più entità, la ruota si definisce Componibile (detta anche ruota con cerchio smontabile).

Nella completa designazione del canale sono presenti tutte le indicazioni che determinano il tipo di accoppiamento/calettamento dello pneumatico sulla ruota. Queste caratteristiche tecniche sono determinate da un'organizzazione europea che si chiama E.T.R.T.O. a cui tutti i produttori di cerchi e gomme si riferiscono.

Un esempio di specifiche tecniche di un cerchio è:

8.0 J x 18" H2

dove

- 8.0 rappresenta la dimensione del canale in pollici;
- J il profilo della flangia del cerchio. Esistono numerosi tipi di bordi dei cerchi e la sigla J indica le autovetture. Per questa ragione, molto spesso si omette se si tratta di cerchi per auto;
- 18" è il diametro nominale espresso in pollici;

- H2 è la sigla del profilo della sezione del cerchio. Tale parametro riguarda le curvature che si effettuano sulla costruzione al fine di renderla più solida. Anche questo parametro molto spesso si omette in dicitura.


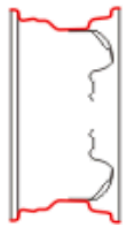
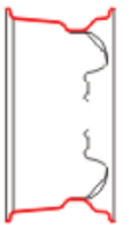




Indicazione	H	H2	FH	FH2	CH	EH2	EH2+
Profilo							
Descrizione	Hump	Double Hump	Flat Hump	Double Flat Hump	Combination Hump	Extended Hump	Extended Hump +
Forma	interna	Gobba	Gobba	Gobba piatta	Gobba piatta	Gobba estesa	Gobba estesa
	esterna	Normale	Gobba	Normale	Gobba piatta	Normale	Gobba estesa

Tabella 1 - Profili della sezione

Indicazione	Applicazione
A	Biciclette, motociclette, motocicli, scooter, furgoni, camion, carrelli elevatori.
B	Vecchi cerchi fino alla larghezza di 6", riscontrate a volte nei vecchi modelli di VW Maggiolino
D	Biciclette, motociclette, motocicli, scooter, furgoni, camion, carrelli elevatori.
E	Camion, furgoni, trattori agricoli (ruote anteriori) con i cerchi „semi-drop”
F	Furgoni, camion, carrelli elevatori con cerchi „semi-drop”.
G	Furgoni, camion - cerchi „semi-drop”
H	Furgoni, camion - cerchi „semi-drop”.
J	Autovetture.
JJ	Fuoristrada 4x4.
P	Riscontrate a volte nei veicoli più vecchi p. es. VW Maggiolino
K	Riscontrate a volte nei veicoli più vecchi p. es Jaguar
S	Veicoli utilitari con cerchi „flatbase”
T	Veicoli utilitari con cerchi „flatbase”.
V	Veicoli utilitari con cerchi „flatbase”
W	Rimorchi agricoli, principalmente sull'asse posteriore

Tabella 2 - Profili della flangia

La ruota è considerata una componente molto importante per la sicurezza del veicolo e per questo deve poter esserne garantita la rintracciabilità.

A tal fine le normative di riferimento (Italiane, Europee e internazionali) sposano le prescrizioni indicate nelle Certificazioni dei sistemi qualità che richiedono che sul prodotto ruota debbano essere indicati anche la Versione/Adattamento (ET, PCD, C.B.).

Discutiamo tali parametri, poiché saranno citati più volte nel corso della trattazione:

- *ET*: anche chiamato *offset*, è lo spazio espresso in mm tra la superficie dell'attacco del cerchio al *mozzo*⁴⁴ ed il piano centrale (di mezzeria) del cerchio stesso;
- *PCD*: “*Pitch Circle Diameter*”, indica il numero e la distanza delle viti di fissaggio. Es: un cerchio che presenta un PCD 5/100 indica che ha 5 fori distanti dal centro 100mm;
- *C.B.*: è la sigla di “*Center Bore*” e indica il diametro del foro centrale del cerchio, quello dove entra il mozzo ruota.

⁴⁴ Il mozzo è la parte centrale di una ruota o più in generale di un organo rotante, che porta generalmente cuscinetti alloggiati al suo interno per permettere alla componente che verrà montata sopra di esso di ruotare, nel nostro caso alla ruota.

3.2.1 – I materiali e i processi produttivi utilizzati per i cerchi in lega

Il materiale più usato per la realizzazione dei cerchi in lega leggera è l'alluminio. Le leghe principalmente usate per la produzione sono:

- **g-ALSi7**: Lega di alluminio e silicio usata per la produzione di cerchi in lega di primo impianto e after market (originali e ricambi). Questa lega si presta molto bene al trattamento termico che migliora del 20% le prestazioni meccaniche e che la rende attualmente la lega di fonderia migliore e più usata.
- **g-ALSi10/11**: Lega di alluminio e silicio usata per i cerchi in lega after market. Non è quasi più usata nel primo impianto perché non si presta al trattamento termico e senza questa fase la resistenza meccanica è minore rispetto a quella della lega g-ALSi7.

I cerchi in lega possono essere realizzati anche in lega di magnesio.

L'utilizzo delle leghe di magnesio però non è molto diffuso, in quanto non è adatto ad una vita duratura del prodotto e richiede molta cura nell'utilizzo. Viene così usato attualmente solo in particolari categorie racing e per pochissime "supercar".

La fusione dell'alluminio può essere eseguita utilizzando due tipi di tecnologie diverse:

- Fusione in gravità: usata principalmente per le leghe g-AlSi10/11. Questa tecnologia prevede che l'alluminio liquido a circa 700 °C entri nello stampo per gravità e lo riempia in modo completo prendendo la forma della ruota.



Figura 5 - Fusione in gravità

- Fusione in bassa pressione: usata principalmente per le leghe g-AlSi7. In questo caso l'alluminio liquido a circa 700 °C viene iniettato nello stampo con una pressione di circa 1.2/1.5 Bar. Il riempimento dello stampo viene assicurato quindi dalla pressione in questione e l'alluminio liquido, riempiendo tutte le aree dello stampo, prende la forma della ruota.

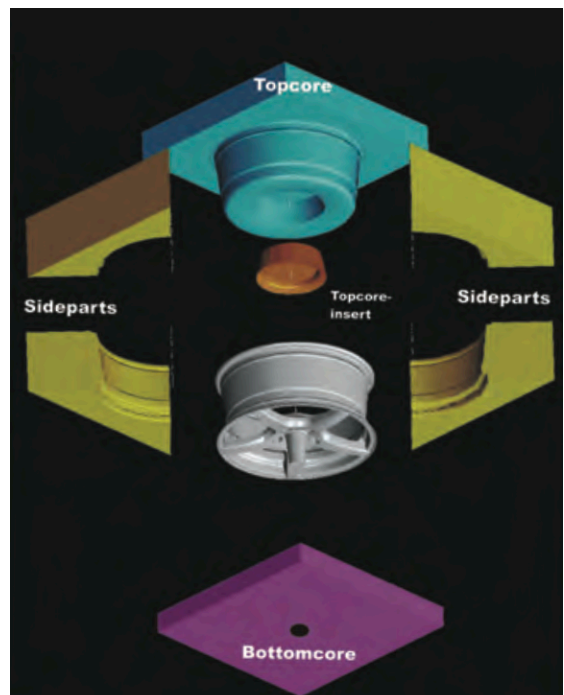


Figura 6 - Fusione in bassa pressione

3.3 – Le richieste del cliente e la definizione del contratto

Assieme al mio referente, durante i primi incontri col cliente sono state stilate le clausole contrattuali. Le richieste principali è stata quella di poter effettuare online qualunque tipo di vendita effettui attualmente, quindi la possibilità di lavorare con transazioni sia di tipo B2B⁴⁵ che B2C⁴⁶.

Inoltre, è stato richiesto che il sito fosse *design responsive*⁴⁷ e multilingue, ovvero ci fosse la disponibilità di cambiare la lingua tramite un menu a tendina, in modo da poter vendere potenzialmente in ogni nazione del mondo, soltanto inserendola tra le desiderate.

Per la gestione dei pagamenti sono state scelte esclusivamente piattaforme elettroniche, quali PayPal e carta di credito, omettendo il bonifico (richiesta aziendale per snellire le procedure di gestione degli ordini).

Sono state definite, inoltre, le date di incontro col cliente per la consegna di prototipi intermedi, per l'attivazione dell'account PayPal, per la scelta dell'interfaccia e per la spiegazione d'utilizzo del CMS.

In tali incontri è stato definito anche il dominio di pubblicazione del sito: "emgcompany.it".

Il marchio con cui saranno venduti i cerchi in lega sarà EMG, ma il sito è stato pensato in modo da poter permettere la vendita in futuro di prodotti con altri marchi.

3.3.1 – Il logo e le immagini dei prodotti

Dalla discussione col cliente è emerso che la produzione delle immagini riguardanti il logo aziendale e i cerchi in lega da mettere in stock sarebbe toccata a un'azienda terza, specializzata nella creazione di loghi e fotografie.

⁴⁵ B2B: Acronico che indica "Business-to-business", ovvero le transazioni commerciali tra imprese

⁴⁶ B2C: Acronico che indica "Business-to-consumer", ovvero le transazioni commerciali tra imprese e consumatori/clienti individuali

⁴⁷ Design responsive: tecnica di Web design per la realizzazione di un sito in grado di adattarsi graficamente in modo automatico al dispositivo col quale viene visualizzato (computer con diverse risoluzioni, tablet, smartphome, cellulari, Web tv)

3.4 – Strumenti utilizzati

Dopo aver discusso di cosa è possibile utilizzare per la creazione di un sito di e-commerce, saranno descritte le tecnologie utilizzate nel caso specifico per lo sviluppo dell'e-commerce considerato.

Come già detto, quando si cerca compatibilità e sicurezza per un server Web è bene scegliere Apache 2.0, ampiamente collaudato e gratuito.

Il DBMS su cui si è sviluppato il progetto è stato MySQL e, oltre al linguaggio di markup HTML5, sono stati utilizzati i linguaggi di programmazione PHP e JavaScript.

A integrazione degli strumenti citati sono intervenuti i framework Laravel (framework MVC per PHP), jQuery (framework per JavaScript) e Bootstrap (framework per CSS).

3.4.1 – Laravel

Laravel è un framework MVC basato su PHP per creare siti Web professionali. Lo slogan utilizzato da Laravel è "The PHP Framework For Web Artisans". L'ultima versione rilasciata è la 5.4.

Si basa su una struttura a directory (cartelle) e tra i file, tutti scritti in PHP, è possibile trovare "routes.php" (detto file di routing) utilizzato per richiamare le funzioni dei controller alla richiesta di un link.

La gestione delle richieste avviene tramite i metodi *get* e *post* della classe Route, che utilizzano le richieste HTTP GET e POST (vedi capitolo 2).

Un esempio di routing per ottenere una pagina col get:

```
Route::get('/login', 'FrontController@login');
```

In tal caso, sarà richiamato il FrontController e invocata la funzione login, che, dopo il processamento da parte del Web server, restituirà l'output della vista, ovvero il codice HTML da inviare al browser.

Per il post valgono le stesse regole. Un esempio che rimanda alla funzione `login_post`:

```
Route::post('/login', ' FrontController@login_post');
```

Le funzioni del controller utilizzeranno quindi i modelli, in questo caso `Images.php` (modello delle immagini della homepage), gestire i contenuti. Per riassumere, il modello permette quindi la visualizzazione, la modifica e l'eliminazione degli elementi del database.

4 – Realizzazione del progetto

Indipendentemente dall'impostazione dello scheduling delle attività di progetto, è necessario lo sviluppo di due sezioni distinte: il front-end e il back-end. Nell'ambito dello sviluppo di un sito Web, viene definita *front-end* la parte del sito visibile da chiunque e raggiungibile tramite link pubblici.

Invece, si definisce *back-end* la parte di amministrazione del sito, accessibile solo dall'amministratore e protetta da un'autenticazione, per la modifica dei contenuti da visualizzare agli utenti del front-end. In un sito strutturato in questo modo, il back-end è detto anche CMS, termine già affrontato nel secondo capitolo.

Poiché si tratta di una soluzione creata appositamente per l'azienda cliente, come il front-end, anche il CMS sarà strutturato e programmato appositamente per la stessa.

Sarà discusso anche il *template* HTML. Nella programmazione Web, quando si parla di template, ci si riferisce a un modello (gratuito o a pagamento) per lo sviluppo di altre pagine Web con grafica e formattazione identiche, ma contenuti diversi.

Per creare un sito Web di e-commerce, prima ancora di creare il front-end e il back-end, è necessario strutturare il DBMS avendo una chiara conoscenza dei prodotti da mettere in vendita e le altre caratteristiche che variano in base al caso aziendale specifico.

4.1 – DBMS

Per impostare la struttura del DBMS è necessario conoscere i prodotti, le esigenze aziendali e una visione generale delle funzioni che dovrà offrire il sito.

Poiché quando si acquista un cerchio in lega per un'auto è necessario sapere i dati di riferimento dell'auto per poter valutare se è possibile accoppiare il cerchio all'auto, la richiesta espressa dall'azienda è stata offrire la possibilità all'utente del front-end (sia esso business o privato) di poter acquistare senza dover conoscere tutti i parametri di riferimento dei cerchi in lega.

La ricerca dei prodotti disponibili per l'auto sarà effettuata semplicemente scegliendo da un menu a tendina la marca dell'auto e il modello, con il relativo anno di fabbricazione della stessa (per esempio: Audi A6 prodotta dal 02/2016 in poi).

In tal modo è possibile raggiungere anche un pubblico meno informato, che potrà effettuare l'acquisto senza competenze specifiche.

4.1.1 – Le specifiche progettuali

Il sito richiesto dal cliente dovrà avere le seguenti funzionalità (la tabella va letta partendo dagli utenti, verso le entità. (Es. l'amministratore può aggiungere, modificare e eliminare i prodotti):

Utenti / Entità	Prodotto	Auto	Colore	Homepage	Carrello
amministratore	<ul style="list-style-type: none">• Aggiunta, modifica eliminazione• Gestione della disponibilità• Gestione della visibilità sul front per il cliente finale	<ul style="list-style-type: none">• Aggiunta, modifica eliminazione	<ul style="list-style-type: none">• Aggiunta, modifica eliminazione	<ul style="list-style-type: none">• Gestione delle immagini	<ul style="list-style-type: none">• Ricezione ordini

	<ul style="list-style-type: none"> • Selezione colori • Gestione immagini 				
guest	<ul style="list-style-type: none"> • Ricerca intelligente • Visualizzazione (se visibile) • Aggiunta al carrello 			<ul style="list-style-type: none"> • Registrazione • Login 	<ul style="list-style-type: none"> • Acquisto senza pre-registrazione • Acquisto
user					<ul style="list-style-type: none"> • Acquisto

Tra le specifiche progettuali si è deciso di includere la creazione di un codice prodotto, che l'azienda non possedeva. Si è scelto di utilizzare un codice alfanumerico composto nella parte iniziale dalle tre lettere "EMG", seguite da un numero di 6 cifre composto da zeri e dall'id (es: EMG000140).

4.1.2 – L'importazione dell'esistente

Si è potuta offrire questa ricerca semplificata partendo da un foglio elettronico *Excel* (Software della suite *Microsoft-Office* per la gestione dei fogli di calcolo), già utilizzato dall'azienda INOVIT S.R.L. per stabilire gli accoppiamenti auto-cerchi.

Tale foglio elettronico è composto di un'unica tabella auto avente come tuple le singole auto con i relativi valori. Si seguito sono riportati gli attributi:

- *MARCA*
- *MODELLO*
- *TIPO*
- *ANNO*
- *DIAMETRO*
- *MISURA*
- *ET*
- *PCD*
- *CB*

- *VITE/DADO* (caratteristiche della vite e del dado per il montaggio)
- *ANELLO* (dimensioni del foro centrale)
- *MISURA 1* (le misure fanno riferimento alle dimensioni dello pneumatico)
- *MISURA 2*
- *MISURA 3*
- *MISURA 4*
- *NOTE* (dettagli testuali riferiti all'auto, da riportare al cliente del front-end)

Il file considerato è stato processato e convertito in un file di tipo *CSV* (*Comma-Separated Values*). Si tratta di un file utilizzato per l'importazione e l'esportazione dei database che separa i vari campi utilizzando un separatore specifico, che, nonostante non esista uno standard formale che lo definisca⁴⁸, solitamente è la virgola o il punto e virgola.

Il foglio elettronico di partenza aveva la dimensione di 12,3 MB, mentre dopo l'esportazione in CSV il file ha raggiunto la dimensione di circa 30 MB.

A un occhio poco attento, 30 MB possono sembrare una dimensione ridotta, ma quando si parla di un database di tipo testuale, si stanno considerando oltre due milioni e mezzo di caratteri. La dimensione, però, risulta giustificata, poiché si tratta di un database da 240.000 righe che riporta tutti i parametri riferiti ai cerchi della maggioranza dei modelli di auto prodotti dal 1982 a oggi e le relative note integrative.

Per la realizzazione del sito è stato utilizzato anche un altro foglio elettronico fornito dall'azienda cliente, di dimensione molto minore rispetto al precedente (circa 600 KB), contenente l'elenco prodotti attualmente a magazzino con le relative specifiche tecniche.

⁴⁸ <https://tools.ietf.org/html/rfc4180#section-2>

4.1.3 – Script di importazione

Tali file sono stati frazionati, riorganizzati e importati su varie tabelle MySql utilizzando script PHP e tramite il comando explode sono stati ottenuti più array denominati \$valori, ognuno dei quali rappresenta una tupla:

```
$valori = explode (',', $linea);
```

La variabile \$linea rappresenta una stringa di valori ordinati secondo l'ordine degli attributi della tabella di partenza al momento dell'esportazione, separati da virgole.

ES:

```
"ABARTH,GRANDE PUNTO,199,2007 >,RUOTE SPORGENTI DALLA  
CARROZZERIA - WHEELS PROUDE FROM FENDERS,18,8,35,4 X 100,72,B  
M12x1.5x27 K17,F (72-56.6),215/40 R18,,,"
```

In tal modo è stato possibile inserire le singole tuple in database molto rapidamente grazie a una funzione di base che PHP integra per la gestione di MySql:

```
mysqli_query($link, $query);
```

dove \$link è la variabile che contiene il link al database e, utilizzando i valori ottenuti in precedenza, una query semplificata di inserimento dati all'interno del DB potrebbe essere del tipo:

```
INSERT INTO `nometabella`(campo1, campo2)  
VALUES ('$valori[0]', '$valori[1]');
```

In questo caso \$valori[0] e \$valori[1] rappresentano i valori dei rispettivi attributi campo1 e campo2.

Il tempo di processamento e di importazione da parte dello script è stato di circa quattro ore, utilizzando un computer con processore Intel Core i7 2,7 GHz 16 GB di RAM a 2133 MHz, poiché si trattava di una grossa mole di dati.



Figura 7 - Specifiche tecniche dell'elaboratore utilizzato

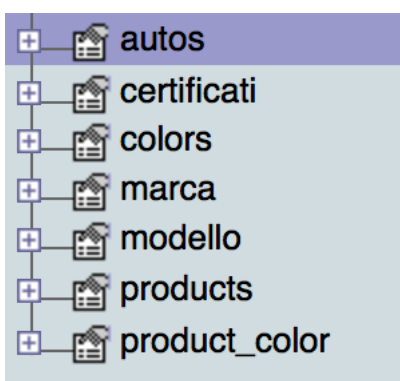


Figura 8 – Lista ottenuta da phpMyAdmin (software per la gestione del database) delle tabelle ottenute dalla frammentazione dei file offerti dall'azienda

Il risultato di questo lavoro è stato un database contenente le tabelle riportate in figura 8, contenenti al loro interno i dati di base sui prodotti e sulle auto fornite dall'azienda cliente.

4.1.4 – La struttura del database

Dopo aver importato il database cliente, è stato necessario creare le altre tabelle al fine di avere un database correttamente strutturato.

A questo punto sono state create le tabelle per la gestione degli utenti (tabella clienti), la gestione delle immagini sulla homepage (tabella images), del carrello (tabella shcart), degli ordini (tabella ordini), delle nazioni (tabella nazioni) con cui il cliente avrà intenzione di commerciare e delle destinazioni (tabella destinazioni), contenente gli indirizzi di spedizione per il cliente finale. Inoltre è stata aggiunta la tabella certificati, per permettere l'upload dei certificati di omologazione dei prodotti (ove necessario).

- *images* è la tabella di gestione delle immagini, dei link e delle descrizioni in homepage;
- *product_color* è la tabella dei prodotti realmente venduti e il codice di vendita sarà associato all'id di *product_color*. Quindi, in futuro, quando si farà riferimento al prodotto, si considererà il prodotto avente del codice prodotto l'id di *product_color*;
- *users* è la tabella di gestione degli utenti administrator e vi risiedono i dati di autenticazione degli utenti che possono accedere al CMS e modificare i contenuti del sito, controllare gli ordini, aggiornare i prodotti a magazzino ecc..

4.2 – Back-end

Dopo aver strutturato il database è stato necessario creare il back-end (CMS) per poter gestire il database tramite interfaccia utente e permettere le funzioni richieste dal cliente.

4.2.1 – Le pagine del CMS

Lo sviluppo del back-end è partito riciclando alcune vecchie viste⁴⁹, precedentemente utilizzate dalla società NoWhere S.R.L. in CMS sviluppati per altre imprese. Dal codice sorgente di tali viste è stato estrapolato e riutilizzato solo l'HTML, il JavaScript e il CSS, poiché il resto del codice non era compatibile con le nuove versioni dei linguaggi utilizzati per lo sviluppo del sito attuale.

La richiesta di riutilizzare tali viste è stata effettuata da NoWhere S.R.L., per poter applicare sempre gli stessi standard di manutenzione del codice su tutti i siti gestiti dalla stessa.

⁴⁹ Già discussa nel capitolo relativo all'MVC, la vista è un mix di tutti i codici che si è deciso di utilizzare (HTML5, PHP, CSS, JavaScript, jQuery, Laravel, Bootstrap) inseriti all'interno di un file con estensione "php", poiché è stato scelto il PHP come linguaggio lato server. Il risultato dell'elaborazione sarà la visione a schermo del contenuto processato e formattato.

EMG - Gestione Sito CAMBIA PASSWORD
LOGOUT

EMGCOMPANY.IT Crea

[HOMEPAGE](#)
[PRODOTTI](#)
[COLORI](#)
[AUTO](#)

COLORI

Gestione colori

Nome	Name	Modifica	Elimina
Antracite Lucido Diamantato	Gloss Anthracite Polished		
Antracite Opaco	Matt Anthracite		
Antracite Opaco Diamantato	Matt Anthracite Polished		
Hyper Silver	Hyper Silver		
Nero Lucido	Gloss Black		
Nero Lucido Diamantato	Gloss Black Polished		
Nero Opaco	Matt Black		
Nero Opaco Diamantato	Matt Black Polished		
Silver	Silver		
Silver con Bordo Diamantato	Silver Lip Polished		
Silver Diamantato	Silver Polished		

CMS - Content Management System v 3.0 - last update 4 Maggio 2017
 Nowhere via Milazzo 24 Bologna, Italia | Assistenza tecnica Tel. 051 587.93.55 Email: supporto@nowhere.it

Figura 9 - Pagina dei colori disponibili

Di seguito sarà riportato come esempio il sorgente della vista della pagina colori, la più semplice del CMS:

```

<html>
<head>
  <meta charset="UTF-8">
  <title>Colori</title>
  <link          rel="stylesheet"          type="text/css"
  href="http://emgcompany.it/css/reset.css"  media="screen"
  />
  <link          rel="stylesheet"          type="text/css"
  href="http://emgcompany.it/css/text.css"   media="screen"
  />
  <link          rel="stylesheet"          type="text/css"
  href="http://emgcompany.it/css/layout.css" media="screen"
  />
  <link          rel="stylesheet"          type="text/css"
  href="http://emgcompany.it/css/nav.css"    media="screen" />
  <link          rel="stylesheet"          type="text/css"
  href="http://emgcompany.it/css/styleGest.css" />
  <link type="text/css" href="http://emgcompany.it/css/ui-
  lightness/jquery-ui-1.8.17.custom.css"   rel="stylesheet"
  />

```

```

<script type="text/javascript"
src="http://emgcompany.it/js/jquery-
1.7.1.min.js"></script>
<script type="text/javascript"
src="http://emgcompany.it/js/jquery-ui-
1.8.17.custom.min.js"></script>
</head>

<body>
  <div class="container_12" style="min-width:900px;">
    <div id="header">
      <h1 style="font-size:35px;"><a
href="/cms/homepage"><strong>EMG</strong> - <span style="font-
size:22px;"><strong>Gestione Sito</strong></span></a></h1>
      <ul id="menu">
        <li><a
href="http://emgcompany.it/cms/password-change"><span
class="ui-button-icon-primary ui-icon ui-icon-
key"></span>CAMBIA PASSWORD</a></li>
        <li><a
href="http://emgcompany.it/logout"><span class="ui-button-
icon-primary ui-icon ui-icon-unlocked"></span>LOGOUT</a></li>
      </ul>
    </div>
    <div id="content">
      <p class="right">
        <a class="ui-state-default ui-corner-all"
id="dialog_link" href="/cms/colori/create">
          <span class="ui-icon ui-icon-
document"></span>Crea</a>
      </p>
      <h2><span class="ui-button-icon-primary ui-icon
ui-icon-clipboard"></span> Colori</h2>
      @if (count($colori) == 0)
        Nessun colore.
      @else
        Gestione colori
      <table class="display" cellpadding="0"
width="100%">
        <thead>
          <tr>
            <th>Nome</th>
            <th>Name</th>

```

```

                <th>Modifica</th>
                <th>Elimina</th>
            </tr>
        </thead>
        <tbody>
            @foreach($colori as $key => $value)
                <tr id="riga{{ $value->id }}">
                    <td>{{ $value->nome }}</td>
                    <td>{{ $value->name }}</td>
                    <td><a href="/cms/colori/{{ $value->id }}/edit" class="ui-button-icon-primary ui-icon ui-icon-pencil" title="Modifica">Modifica</a></td>
                    <td><a href="/cms/colori/{{ $value->id }}" class="ui-button-icon-primary ui-icon ui-icon-trash" title="Elimina">Elimina</a></td>
                </tr>
            @endforeach
        </tbody>
    </table>
    <div style="width: 50%; margin: 0 auto;">
    </div>
    @endif
    <div class="clear"></div>
</div>
<div class="clear"></div>
<div id="footer" name="footer"><strong>CMS - Content Management System v 3.0 - last update 4 Maggio 2017</strong>
<br>
        <a href="http://www.nowhere.it">Nowhere</a> via
Milazzo 24 Bologna, Italia | Assistenza tecnica Tel. 051
587.93.55
        Email:
        <a href="mailto:supporto@nowhere.it">supporto@nowhere.it</a>
    </div>
    <!-- </div> -->
</div>
</body>
</html>

```

Per comprendere il codice precedente oltre al PHP, è necessario conoscere i tag HTML fondamentali:

- `<head>` è il tag di intestazione. È il contenitore delle informazioni relative alla pagina Web. Va inserito subito dopo il tag `<html>` e chiuso prima del tag `<body>` e fa parte della struttura HTML base di ogni pagina Web. Gli elementi contenuti del tag `<head>` possono essere script, collegamenti a fogli di stile, meta tag e molto altro. I marcatori con cui questi elementi vengono richiamati sono: `<base>`, `<link>`, `<meta>`, `<script>`, `<style>` e `<title>`. Quest'ultimo è l'unico tag obbligatorio e serve a definire il titolo della pagina
- `<body>` definisce il corpo del documento, la struttura e il contenuto che dovrà essere visualizzato e formattato;
- `<div>` indica una sezione rettangolare del documento, anche detto contenitore è un elemento di tipo *block* (un elemento è definito di tipo *block* se possiede margini, bordi e padding e può contenere altri elementi di tipo *block*);
- `` indica un elenco non ordinato e `` indica l'elemento del menu;
- `<a>` indica gli *hyperlink* (link ipertestuali). Il valore dell'attributo `href` indica l'URL della risorsa;
- `<p>` indica un paragrafo;
- i tag che vanno da `<h1>` ad `<h6>` servono per mettere in risalto il testo, rispetto a quello con formattazione di base;
- il tag `<table>` rappresenta una tabella, che è formata da due parti entrambe strutturate in righe (`<tr>`) e colonne (`<th>`):
 - `<thead>` contenente gli attributi della tabella
 - `<tbody>` i valori degli attributi.

I tag `<script>` aventi `type="text/javascript"` sono utilizzati per richiamare le librerie jQuery tramite URL, mentre con i tag `link` aventi `type="text/css"` si richiameranno i CSS della pagina, anche questi tramite URL.

Il primo `div` principale del `<body>` (`<div id="header">`) è standardizzato ed è il contenitore nella parte alta del sito che riporta la dicitura “EMG – Gestione sito” e i link per il cambio password e il logout.

Anche il div del footer è stato standardizzato e riporta i dati per contattare NoWhere S.R.L., in caso il cliente necessitasse di assistenza.

Il div che viene riempito dinamicamente in base al link della pagina tramite PHP (Laravel) è quello avente id="content".

Il codice Laravel, essendo PHP, è quindi scritto assieme all'HTML e interpretato a seguito della richiesta del client. Verrà riportato di seguito per facilitarne la lettura:

```
@if (count($colori) == 0)
    Nessun colore.
@else
    Gestione colori
    <table class="display" cellspacing="0"
width="100%">
        <thead>
            <tr>
                <th>Nome</th>
                <th>Name</th>
                <th>Modifica</th>
                <th>Elimina</th>
            </tr>
        </thead>
        <tbody>
            @foreach($colori as $key => $value)
                <tr id="riga{{ $value->id }}">
                    <td>{{ $value->nome }}</td>
                    <td>{{ $value->name }}</td>
                    <td><a href="/cms/colori/{{ $value-
>id }}/edit" class="ui-button-icon-primary ui-icon ui-icon-
pencil" title="Modifica">Modifica</a></td>
                    <td><a href="/cms/colori/{{ $value-
>id }}" class="ui-button-icon-primary ui-icon ui-icon-trash"
title="Elimina">Elimina</a></td>
                </tr>
            @endforeach
        </tbody>
    </table>
    <div style="width: 50%; margin: 0 auto;">
</div>
@endif
```

In tal caso, il codice è stato strutturato per fornire in output “Nessun colore.” nel caso ci fosse un array \$colori vuoto, oppure la tabella rappresentata nella figura 11 nel caso i colori fossero già stati inseriti sul database.

È possibile identificare il codice Laravel all’interno del sorgente dalla @ posta danti ai comandi (es: @if) e dalle doppie parentesi graffe che racchiudono le variabili da stampare a schermo.

Il comando @foreach utilizzerà quindi l’array \$colori e lo itererà per riempire la tabella utilizzando l’oggetto \$value e richiamando le proprietà dell’oggetto id, nome e name.

EMG - Gestione Sito [CAMBIA PASSWORD](#)
[LOGOUT](#)

EMGCOMPANY.IT **COLORI E IMMAGINI**

Colori del prodotto

Marca	Design	Dimensioni	ET	Pcd	CB	Vite - Dado	Prezzo
Atest	Atest	19 x 9	50.0	5 X 112	66,6	Esempio vite per Atest	165.00

Colore it - en	Immagini	Disponibilità	Visibile	Elimina
Anthracite Polished ITA Anthracite Polished		10 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina
Black - Machined Face ITA Black - Machined Face		10 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina
Black - Polished INOX Lip ITA Black - Polished INOX Lip		13 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina
Black - Satin Lacquer ITA Black - Satin Lacquer		32 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina
Chrome Inserts - Hyper Silver ITA Chrome Inserts - Hyper Silver		11 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina
Hyper Silver - Chrome Inserts ITA Hyper Silver - Chrome Inserts		10 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina
Matt Black Polished ITA Matt Black Polished		12 <input type="button" value="Aggiorna"/>	<input checked="" type="checkbox"/>	Elimina

Aggiungi un colore al prodotto

[VAI AL PRODOTTO](#)

CMS - Content Management System v 3.0 - last update 4 Maggio 2017
Nowhere via Milazzo 24 Bologna, Italia | Assistenza tecnica Tel. 051 587.93.55 Email: supporto@nowhere.it

Figura 10 - Esempio di vista del CMS: schermata colori di un prodotto. N.B. Tutti i file i dati e le immagini sono inserimenti di prova

Ogni pagina del back-end è stata strutturata con un menu di tipo statico posizionato nella colonna a sinistra dello schermo, soluzione che permette di

accedere rapidamente alle funzioni principali, identificato dal codice HTML presente in tutte le pagine.

```
<div class="grid_2">
  <ul class="menu">
    <li><a href="/" id="dialog_link1" class="uppercase ui-
accordion-header ui-helper-reset ui-state-default ui-corner-
all" target="_blank">EMGCOMPANY.IT</a></li><br>
    <li><a href="/cms/homepage" id="dialog_link2"
class="uppercase ui-accordion-header ui-helper-reset ui-state-
default ui-corner-all">Homepage</a></li>
    <li><a href="/cms/prodotti" id="dialog_link3"
class="uppercase ui-accordion-header ui-helper-reset ui-state-
default ui-corner-all">Prodotti</a></li>
    <li><a href="/cms/colori" id="dialog_link4"
class="uppercase ui-accordion-header ui-helper-reset ui-state-
default ui-corner-all">Colori</a></li>
    <li><a href="/cms/auto" id="dialog_link5" class="uppercase
ui-accordion-header ui-helper-reset ui-state-default ui-
corner-all">Auto</a></li>
  </ul>
</div>
```

I tag a sono stati opportunamente modificati tramite CSS inserendo gli attributi atti a rendere la formattazione gradevole alla vista, mostrando l'elenco puntato come una colonna di pulsanti cliccabili.

4.2.2 – Le funzioni del back-end

Tramite il back-end, l'azienda cliente avrà la possibilità di utilizzare varie funzioni operative, quali:

- Gestione homepage
 - Aggiungere, modificare ed eliminare le immagini, i link e le descrizioni della homepage
- Gestione prodotto
 - Aggiungere, modificare ed eliminare i prodotti
 - Aggiungere, modificare ed eliminare le immagini dei prodotti

- Aggiungere, modificare ed eliminare i certificati dei prodotti
 - Modificare le quantità e i colori assegnati ai prodotti
 - Scegliere quali prodotti rendere visibili sul front-end
- Gestione colore
 - Aggiungere, modificare ed eliminare i colori assegnabili ai prodotti
 - Gestione auto
 - Aggiungere, modificare ed eliminare⁵⁰ le auto in database

4.2.3 – Sitemap del back-end

Dopo una fase di analisi progettuale, si è scelto di impostare il back-end come nella mappa in figura 12.

BACK-END

Site Map <http://emgcompany.it>

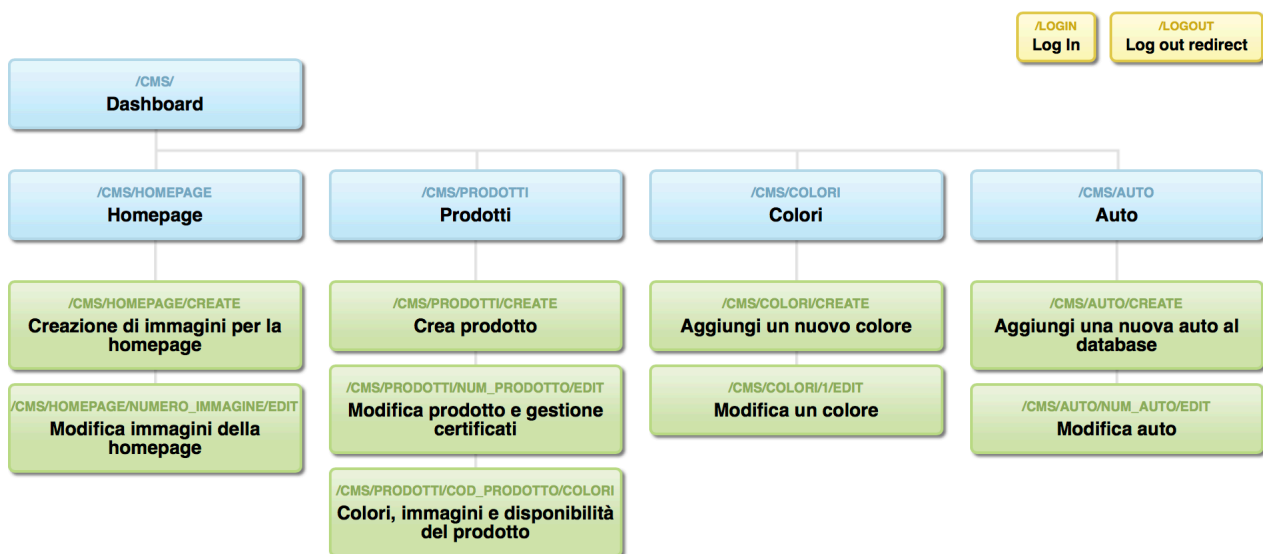


Figura 11 – Mappa del back-end

La gestione di tali collegamenti avviene tramite Laravel.

⁵⁰ È stata inserita la restrizione sulla possibilità di eliminare le sole auto aggiunte post-importazione tramite il CMS dal cliente. La scelta è stata effettuata per ridurre i tempi di creazione del sito, poiché il cliente in vari anni di attività non ha mai avuto bisogno di eliminare auto dal database privato. Nel caso in futuro dovesse verificarsi un'eccezione, sarà comunque possibile eliminare con un intervento manuale sul database le auto non desiderate.

Un esempio può essere:

```
Route::get('/cms/auto', 'AutosController@index');
```

Con tale riga di codice si richiama la funzione index del controller AutosController, che utilizzerà il modello Auto.php per gestire i contenuti della view ottenuta dall'URL /cms/auto.

Modello Auto.php:

```
class Auto extends Model
{
    protected $fillable = ['marca', 'modello', 'tipo', 'anno',
        'note', 'diametro', 'misura', 'et', 'pcd1', 'pcd2', 'cb',
        'vitedado', 'anello', 'misurala', 'misuralb', 'misuralc',
        'misura2a', 'misura2b', 'misura2c', 'misura3a',
        'misura3b', 'misura3c', 'misura4a', 'misura4b'];

    public function add($data)
    {
        $this->marca = $data['marca'];
        $this->modello = $data['modello'];
        $this->tipo = $data['tipo'];
        $this->anno = $data['anno'];
        $this->note = $data['note'];
        $this->diametro = $data['diametro'];
        $this->misura = $data['misura'];
        $this->et = $data['et'];
        $this->pcd1 = $data['pcd1'];
        $this->pcd2 = $data['pcd2'];
        $this->cb = $data['cb'];
        $this->vitedado = $data['vitedado'];
        $this->anello = $data['anello'];
        $this->misurala = $data['misurala'];
        $this->misuralb = $data['misuralb'];
        $this->misuralc = $data['misuralc'];
        $this->misura2a = $data['misura2a'];
        $this->misura2b = $data['misura2b'];
        $this->misura2c = $data['misura2c'];
        $this->misura3a = $data['misura3a'];
    }
}
```

```

        $this->misura3b = $data['misura3b'];
        $this->misura3c = $data['misura3c'];
        $this->misura4a = $data['misura4a'];
        $this->misura4b = $data['misura4b'];
        $this->misura4c = $data['misura4c'];
        $this->id_modello = $data['id_modello'];
        self::save();
    }

    public function edit($data)
    {
        $this->marca = $data['marca'];
        $this->modello = $data['modello'];
        $this->tipo = $data['tipo'];
        $this->anno = $data['anno'];
        $this->note = $data['note'];
        $this->diametro = $data['diametro'];
        $this->misura = $data['misura'];
        $this->et = $data['et'];
        $this->pcd1 = $data['pcd1'];
        $this->pcd2 = $data['pcd2'];
        $this->cb = $data['cb'];
        $this->vitedado = $data['vitedado'];
        $this->anello = $data['anello'];
        $this->misura1a = $data['misura1a'];
        $this->misura1b = $data['misura1b'];
        $this->misura1c = $data['misura1c'];
        $this->misura2a = $data['misura2a'];
        $this->misura2b = $data['misura2b'];
        $this->misura2c = $data['misura2c'];
        $this->misura3a = $data['misura3a'];
        $this->misura3b = $data['misura3b'];
        $this->misura3c = $data['misura3c'];
        $this->misura4a = $data['misura4a'];
        $this->misura4b = $data['misura4b'];
        $this->misura4c = $data['misura4c'];
        self::save();
    }
}

```

Il modello è quindi una classe che estende la classe di base del modello fornita da Laravel, detta Model.

Il modello Auto.php conterrà al suo interno, oltre alle funzioni di add e edit (aggiunta e modifica delle auto in database), anche gli attributi della tabella auto nella proprietà \$fillable, in modo da poter modificare i valori degli attributi relativi.

4.3 – Front-end

Durante la fase finale di sviluppo del back-end si è portato avanti parallelamente lo sviluppo del front-end.

4.3.1 – La scelta del template

Per poter cominciare a lavorare sul font-end, è stato fissato un incontro col cliente in modo da scegliere il template da utilizzare.

Assieme al referente aziendale è stata effettuata una selezione dei template più consoni al caso, tutti HTML5 design responsive, basati su bootstrap. Tale scelta permette la visualizzazione su qualunque dispositivo con un browser, quindi è possibile sviluppare tramite i mezzi elencati un unico sito Web per computer, tablet, smartphone di qualunque marca e la visualizzazione sarà adattata alla risoluzione del browser che visualizza la pagina. La selezione è stata effettuata tramite siti professionali che offrono template a pagamento per siti di e-commerce.

Tra le proposte, il cliente ha optato per un template selezionato su *ThemeForest*⁵¹ che si presentava come in figura 13.

⁵¹ <https://themeforest.net/>

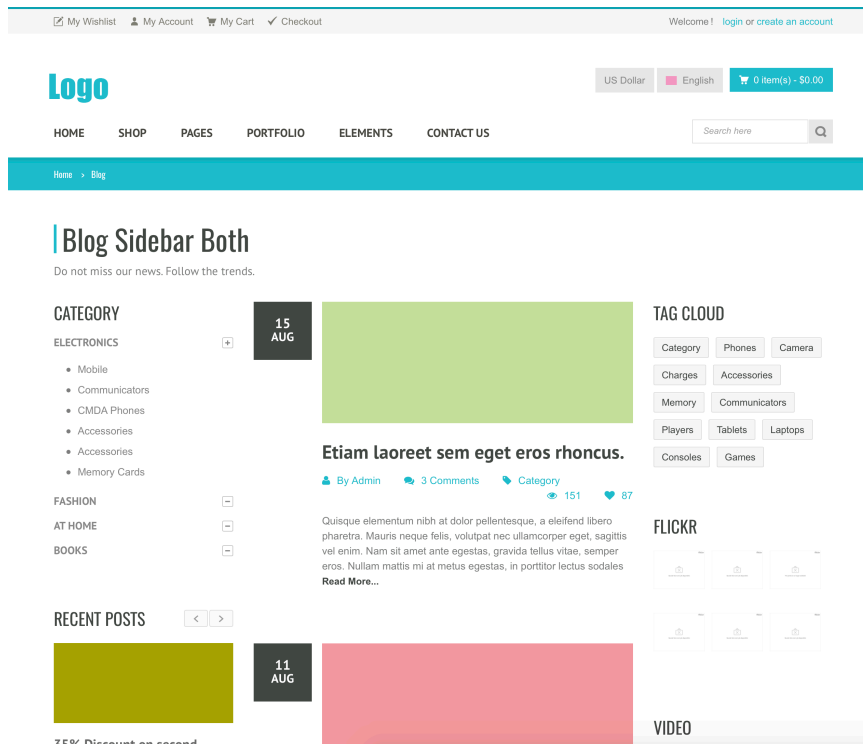


Figura 12 - Template utilizzato per la creazione delle viste del front-end

4.3.2 – Le funzioni del front-end

È stata pensata una registrazione sia per utenti business che privati. Oltre al pannello personale del cliente (business e privato), dove è possibile modificare i dati personali, gli indirizzi di spedizione, la password, l'e-mail personale, è stato pensato come i moderni e-commerce in cui il cliente può acquistare anche senza effettuare prima la registrazione. In tal caso i dati personali e aziendali saranno richiesti durante la fase di check-out.

4.3.3 – Sitemap del front-end

Il front-end presenta una mappa del sito più complessa rispetto al back-end ed è riportata in figura 14:

FRONT-END

Site Map <http://emgcompany.it>

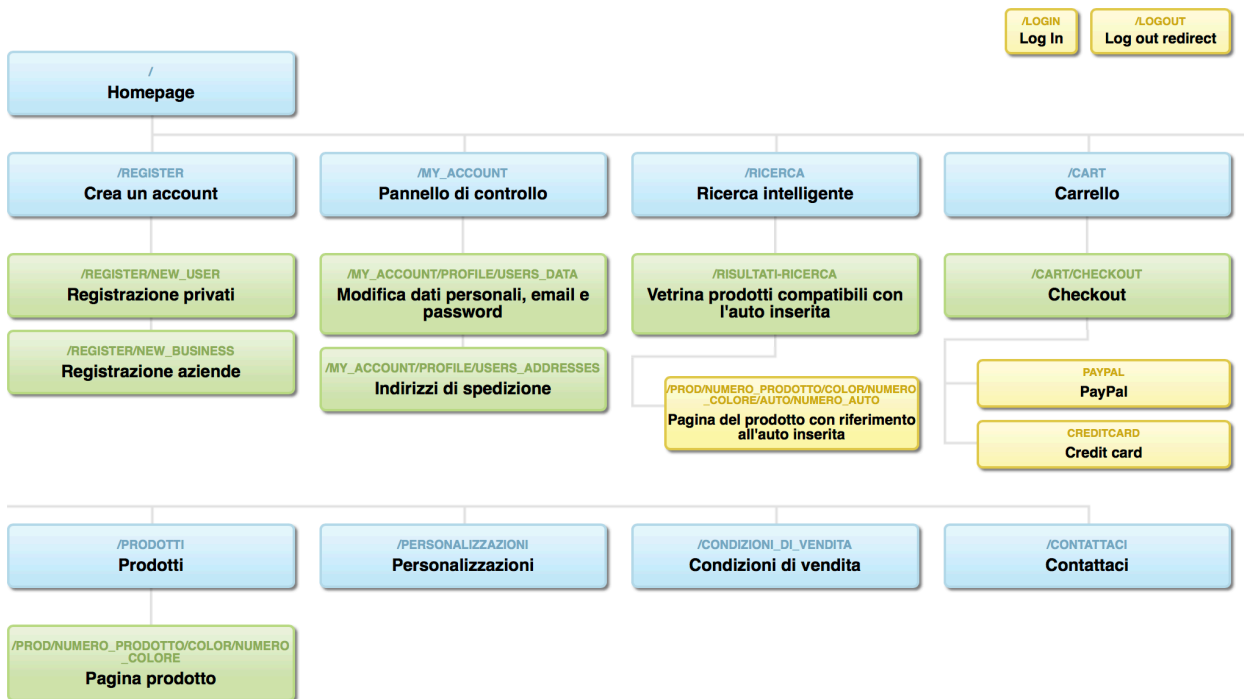


Figura 13 - Mappa del front-end

È bene notare come la ricerca intelligente rimandi a una pagina prodotto differente rispetto alla pagina prodotto ottenibile dalla pagina prodotti. Tale scelta è stata effettuata per semplificare la separazione tra i clienti che acquistano i cerchi partendo dai parametri di riferimento degli stessi e da coloro che acquistano fornendo la marca e il modello della propria auto. Utilizzando tale differenziazione sarà possibile far acquistare sia il privato, che non conosce i parametri di riferimento dei cerchi, bensì è informato solo sul modello della sua auto, sia l'officina che ha padronanza dei dati tecnici.

4.3.4 – Struttura della homepage

Sarà discussa la struttura principale della homepage per comprendere meglio le funzioni offerte dal sito utilizzando la figura 15, in cui sono stati evidenziati i contenitori principali.

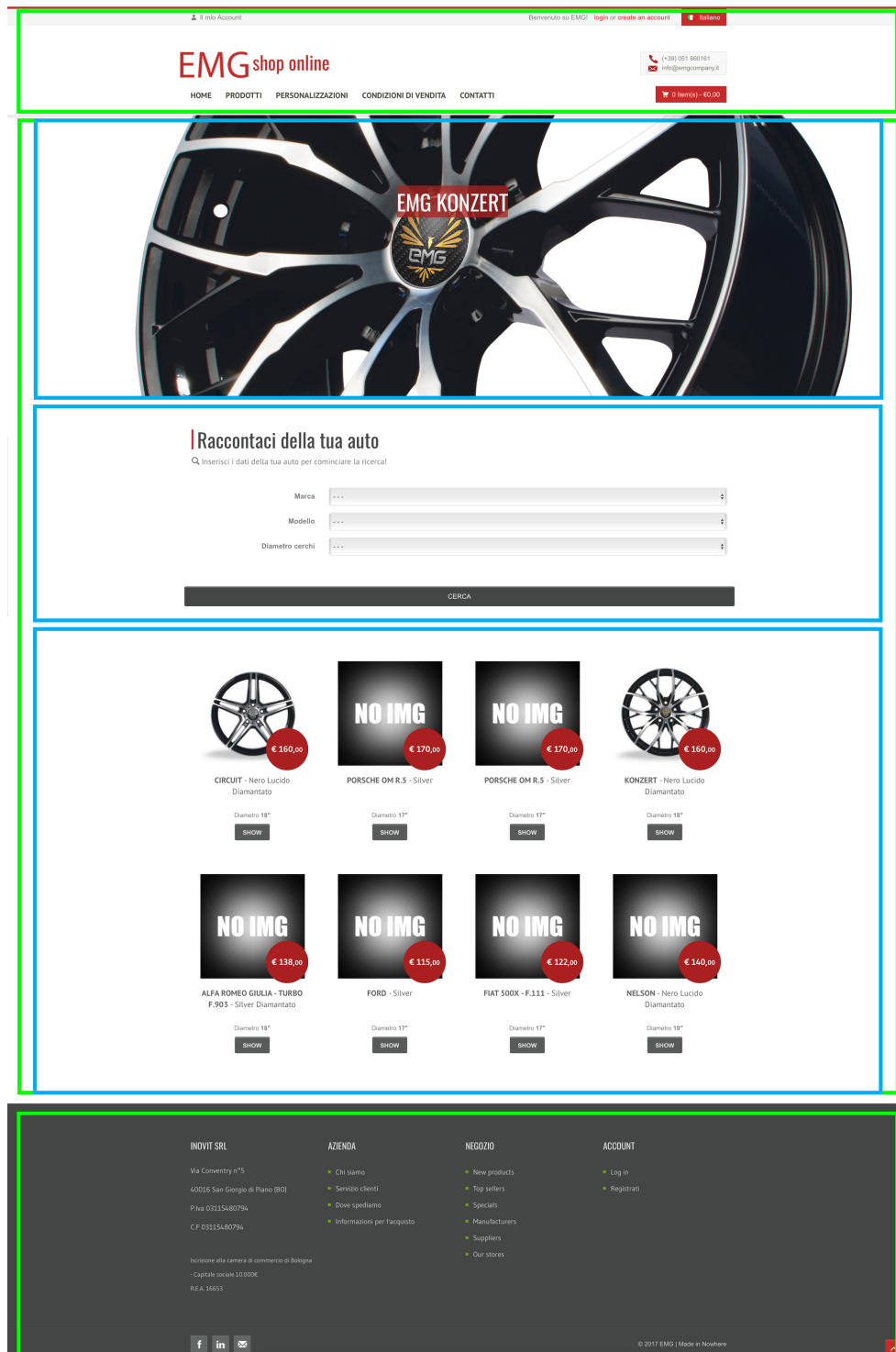


Figura 14 - Homepage con dati di prova che mostra la struttura dei contenitori principali

Il codice all'interno del tag *body* contenuto in una vista generica del front-end è molto semplice:

```
<body>
<div id="wrapper">
  <header id="header">
    @yield('header')
  </header>
  <!-- End #header -->

  <section id="content">
    @yield('content')
  </section>
  <!-- End #content -->

  <footer id="footer">
    @yield('footer')
  </footer>
  <!-- End #footer -->
</div>
</body>
```

È stato utilizzato un div all'interno del body con id="wrapper" che conterrà tutto il corpo della pagina per semplificare la formattazione. Nel codice sono identificabili i div esterni principali presenti in figura 15 nei tag *<header>*, *<section>* e *<footer>*. Tali tag sono stati introdotti con HTML5 e possono essere usati in modo simile ai div facilitando la lettura del codice da parte dello sviluppatore.

A questo punto si utilizza la funzione *@yield()*, disponibile in Laravel, per riempire la vista generica inserendo il contenuto all'interno dei tag *<header>*, *<section>* e *<footer>*, in questo caso con del codice HTML, ma tale comando si potrebbe utilizzare per qualunque contenuto testuale generico.

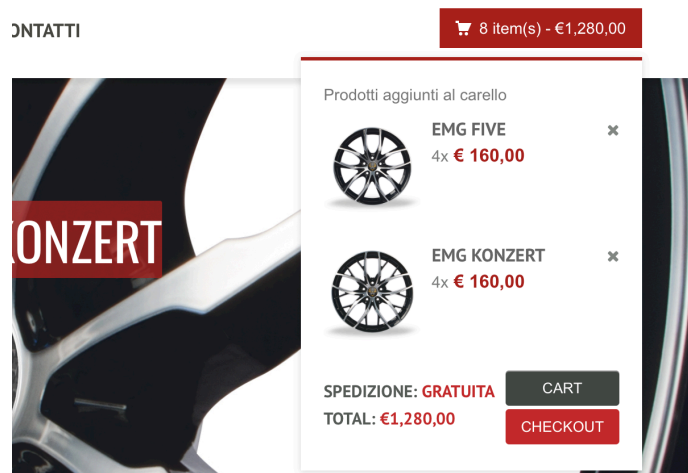


Figura 16 – Dettaglio del dropdown del tasto carrello nell'header utilizzando dati di prova

Tutte e tre le sezioni saranno riempite dinamicamente. Un esempio per l'header e il footer potrebbe essere la sezione di login, che varia il contenuto mostrato in base all'essere loggati o meno. Mentre le sezioni in alto e in basso restano più o meno simili, variando solo nei dettagli. Il contenuto di section, invece, è spesso completamente differente di vista in vista.

Il tag header contiene il menu statico ai link principali del sito, come homepage, prodotti, personalizzazioni, condizioni di vendita e contatti. Nell'header è possibile notare anche la presenza di altri hyperlink, come carrello, accesso all'account personale, registrazione e login per gli utenti già registrati.

Per quel che concerne la pagina analizzata (homepage), il contenuto del tag section inserito tramite il comando yield genererà tre div principali. Nel primo vi è lo slider delle immagini prodotto. Lo scorrimento sequenziale temporizzato avviene tramite jQuery, come la gestione del click sulle frecce ai lati rappresentato in figura 17.



Figura 15 - Slider della homepage con caratteristiche frecce laterali per permettere lo scorrimento manuale.

Il secondo div contiene al suo interno la ricerca intelligente dei cerchi in base al modello dell'auto, utilizzando i menu a tendina.

Raccontaci della tua auto

🔍 Inserisci i dati della tua auto per cominciare la ricerca!

Marca:

Modello:

Diametro cerchi:

- 16
- 17
- 18
- 19
- 20

Figura 16 - Esempio d'utilizzo del menu a tendina con dati reali

Il terzo div è un elenco di prodotti casuali che potrebbero richiamare l'attenzione del cliente.

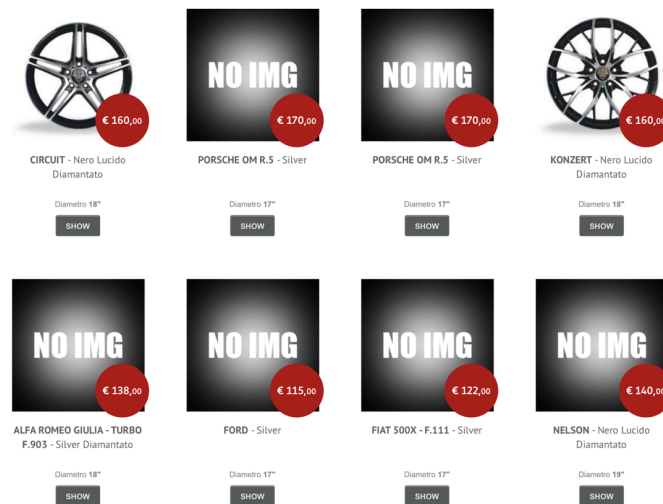


Figura 17 - Prodotti scelti in maniera casuale dal database

4.3.5 – La ricerca intelligente

Il modulo di ricerca cerchi è stato pensato per rendere la ricerca semplice, permettendo al cliente di utilizzare esclusivamente il mouse.

La ricerca strutturata in questo modo richiede una grande elaborazione di dati. Analizziamola più dettagliatamente utilizzando il codice PHP del modello FrontController che effettua concretamente la ricerca:

```
//A
$id_marca= $request->input('tendina_marca');
//B
$tmp=Marca::select('nome')->where('id','',$id_marca)->get();
$strNomeMarca = $tmp[0]->nome;
//C
$id_modello = $request->input('tendina_modello_anno');
$diametro = $request->input('tendina_diametro');
//D
$tmp= Modello::select('id_marca', 'nome', 'anno', 'tipo')->where('id', '=', $id_modello)->get();
$strNomeModello = $tmp[0]->nome;
$strAnnoModello = $tmp[0]->anno;
//E
$tmp= Marca::select('nome')->where('id', '=', $id_marca)->get();
$strNomeMarca = $tmp[0]->nome;
//F
$tmp = Auto::select(array('id', 'marca', 'modello', 'id_modello', 'tipo', 'anno', 'note', 'diametro', 'misura', 'et', 'pcd1', 'pcd2', 'cb', 'vitedado', 'anello', 'misurala', 'misuralb', 'misuralc', 'misura2a', 'misura2b', 'misura2c', 'misura3a', 'misura3b', 'misura3c'))->where('marca', '=', $strNomeMarca)->where('modello', '=', $strNomeModello)->where('anno', '=', $strAnnoModello)->where('diametro', '=', $diametro);
$data['auto'] = $tmp->distinct()->first();
$min_canale = $tmp->min('misura');
$max_canale = $tmp->max('misura');
```

```

$set_min = $tmp->min('et');
$set_max = $tmp->max('et');
$pcd1 = 0;
$pcd2 = 0;
//G
$products = Prodotto::where('products.diametro', $diametro)
    ->whereBetween('products.misura', array($min_canale,
$max_canale))
    ->whereBetween('products.et', array($set_min, $set_max))
    ->whereIn('products.pcd1',
        function ($query) use ($strNomeModello,
$strAnnoModello, $min_canale, $max_canale, $set_min, $set_max,
$diametro, $pcd1, $pcd2) {
            $query->select('pcd1')
                ->from('autos')
                ->where('diametro', $diametro)
                ->whereRaw("modello = '$strNomeModello'")
                ->whereRaw("anno = '$strAnnoModello'")
                ->whereRaw("misura BETWEEN '$min_canale' AND
'$max_canale'")
                ->whereRaw("et BETWEEN '$set_min' AND '$set_max'")
            ;
        }
    )
    ->whereIn('products.pcd2',
        function ($query) use ($strNomeModello,
$strAnnoModello, $min_canale, $max_canale, $set_min, $set_max,
$diametro, $pcd1, $pcd2) {
            $query->select('pcd2')
                ->from('autos')
                ->where('diametro', $diametro)
                ->whereRaw("modello = '$strNomeModello'")
                ->whereRaw("anno = '$strAnnoModello'")
                ->whereRaw("misura BETWEEN '$min_canale' AND
'$max_canale'")
                ->whereRaw("et BETWEEN '$set_min' AND '$set_max'")
            ;
        }
    )->distinct()

```

Discutiamo il codice utilizzando i commenti (identificati dai caratteri “//”, ovviamente senza apici) inseriti all’interno dello stesso per adattarlo alla stesura di questo lavoro di tesi:

- //A. Si inserisce nella variabile \$id_marca l’id della marca ottenuto dalla tendina (\$request permette di gestire sia richieste di tipo GET che POST);
- //B. Si ottiene il nome dell’istanza di Marca avente l’id marca presente in tendina;
- //C. Si inseriscono nelle variabili \$id_modello e \$diametro i valori ottenuti dalla tendina;
- //D. Si utilizza il modello Modello per reperire il nome del modello, l’anno e l’id;
- //E. Si utilizza il modello Marca per reperire il nome della marca, che non può essere cercata tramite id altrimenti vengono persi alcuni risultati a causa della mancata standardizzazione del database cliente;
- //F. Si utilizza il modello Auto per ottenere tutti i dati sull’auto del cliente;
- //G. Si effettua la ricerca del prodotto e si aggiungono al modello le note relative all’auto

Come si può notare, la parte più complessa è la query che utilizza il linguaggio Laravel (dopo il commento G), per offrire i risultati della ricerca.

Ma come funziona?

Si utilizza il modello Prodotto per ricercare tutti i prodotti aventi il diametro in tendina, in cui il canale (identificato sul database dall’attributo “misura” nella tabella “products”) sia compreso tra il massimo e il minimo montabili sull’auto, l’ET sia compreso tra il massimo e il minimo montabili sull’auto, e il PCD (splittato in due valori sul database pcd1 e pcd2) corrisponda a quello montato dall’auto. Per la ricerca dei parametri del pcd sono state effettuate delle *nested query* (query all’interno della query principale), identificate dalla variabile \$query.

Si può notare come per la ricerca non è stato considerato il “tipo” dell’auto, dato presente sul libretto, ma di difficile reperimento, poiché avrebbe potuto confondere l’acquirente. Per tale ragione si è scelto di basare la ricerca soltanto su marca, modello e anno dell’auto. Il tipo resta un’informazione comunque

presente sul database aziendale, visualizzabile e modificabile dal back-end, ma non presentata al cliente finale.

Tale codice è stato replicato in una variante, richiamata nel caso in cui si ricerchi su tutti i diametri del modello dell'auto.

4.3.6 – La pagina dei risultati di ricerca

Cliccando sul tasto “cerca” si viene reindirizzati alla pagina dei risultati (è consigliato aiutarsi visivamente con la sitemap in figura 14 per comprendere i collegamenti principali tra le pagine).

The screenshot shows the EMG shop online interface. At the top, there is a navigation bar with 'Il mio Account', 'login or create an account', and 'Italiano'. The main header features the 'EMG shop online' logo, contact information (+39) 051 860161 and info@emgcompany.it, and a shopping cart icon showing '0 item(s) - €0,00'. Below the header is a breadcrumb trail: 'Home > Risultati'. The main content area is titled 'Prodotti disponibili' and 'Prodotti disponibili per la tua auto'. It displays a grid of wheel products under the category 'MERCEDES - R-KLASSE'. The products are:

- A x R-KLASSE2 - Black Satin Laquer ITA (€ 135,00, Diametro 17")
- A x R-KLASSE2 - Hyper Silver ITA (€ 135,00, Diametro 17")
- SM 051 - Hyper Black ITA (€ 80,00, Diametro 18")
- Atest - Anthracite Polished ITA (€ 165,00, Diametro 19") - This product is highlighted with a green border.
- Atest - Black - (€ 165,00)
- Atest - Black - Polished (€ 165,00)
- Atest - Black - Satin (€ 165,00)
- Atest - Chrome Inserts (€ 165,00)

Figura 18 - Esempio di simulazione della ricerca con valori di prova

La vista della pagina dei risultati presenta nel tag section l'elenco dei singoli prodotti. L'elemento evidenziato in figura 20 rappresenta uno dei prodotti che saranno ripetuti all'interno del codice per offrire la pagina dei risultati di ricerca.

Si prenda quindi come esempio il 4 prodotto che avrà codice HTML:

```
<div class="item item-hover">
  <div class="item-image-wrapper">
    <figure class="item-image-container">
      <a
href="http://emgcompany.it/prod/123/color/1/auto/73777">
        
          
          </a>
        </figure>
        <div class="item-price-container"> <span class="item-price">€ 165,<span class="sub-price">00</span></span>
        </div>
        <!-- End .item-price-container -->
      </div>
      <!-- End .item-image-wrapper -->
      <div class="item-meta-container" style="text-align:center; height:74px;">
        <h3 class="item-name"><a href="http://emgcompany.it/prod/123/color/1/auto/73777"><b>Atest</b> - Anthracite Polished ITA</a></h3>
        <!-- End .item-action -->
      </div>
      <div class="row" style="text-align:center;">
        <div class="row">
          <span class="small">Diametro <b>19" </b></span></div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <div class="row" style="text-align:center;">
            <div class="item-action align-bottom"
style="margin-top:8px;">
                <a
href="http://emgcompany.it/prod/123/color/1/auto/73777"
class="item-add-btn"> <span class="icon-cart-text"
style="">Show</span> </a>
            </div>
        </div>
    </div>
    <!-- End .item-meta-container -->
</div>

```

Il riquadro indicante il singolo prodotto sarà quindi racchiuso dal div avente class="item item-hover".

Come è possibile notare dal codice, qualunque link collegato al prodotto (tramite il tag <a> e l'attributo src) rimanda alla pagina avente URL /prod/123/color/1/auto/73777, dove:

- 123 è l'id del prodotto;
- 1 è l'id del colore;
- 73777 è l'id dell'auto.

Tali informazioni saranno utilizzate dalla pagina prodotto per mostrare il prodotto (reperito dalla tabella product_color) avente il corretto id.

È possibile notare all'interno dell'attributo src del tag URL "non convenzionali", che rimandando, anziché a un'immagine, a una pagina PHP (es: http://emgcompany.it/cimage.php?src=uploaded_products_images/12.png&am p;width=250&am p;height=250).

Il motivo di tale particolarità è dovuto all'utilizzo di *cimage*⁵², libreria per PHP per il ridimensionamento dinamico delle immagini.

Passando nell'URL i parametri src, width e height, tale libreria restituirà l'immagine di input, ma alla risoluzione richiesta tramite width e height (in questo caso è un'immagine quadrata da 250 *pixel*⁵³).

⁵² <https://github.com/mosbth/cimage>

⁵³ Pixel: indica il numero dei punti campione che compongono la rappresentazione di un'immagine.

4.3.7 – La pagina prodotto

Dalla pagina dei risultati della ricerca, cliccando sul prodotto, è possibile arrivare alla pagina del prodotto, da cui sarà possibile metterlo all'interno del carrello nella quantità desiderata.

The screenshot shows the product page for 'Atest Atest' wheels. The page layout includes a navigation bar at the top with 'HOME', 'PRODOTTI', 'PERSONALIZZAZIONI', 'CONDIZIONI DI VENDITA', and 'CONTATTI'. The main content area features a large image of a wheel with a price tag of €165,00, a 'ADD TO CART' button, and a table of technical specifications. The footer contains contact information for INOVIT SRL, company details, and social media links.

Atest Atest

Product Code: EMG000140
Brand: Atest
Design: Atest
Color: Anthracite Polished
Diametro: 19
Compatibile con: MERCEDES - R-KLASSE
Note: RUOTE SPORGENTI DALLA CARROZZERIA - WHEELS PROTUD...
Availability: In Stock

4 **ADD TO CART**

Diametro	Canale	ET	PCD	CB	Vite/Dado	Prezzo
19"	9,0"	50	5x112	66,6mm	Esempio vite per Atest	165,00 €

Descrizione

CINQUE RAZZE

[Scarica il certificato di omologazione del prodotto](#)

INOVIT SRL
Via Conventry n°5
40016 San Giorgio di Piano (BO)
P.Iva 03115480794
C.F. 03115480794
Iscrizione alla camera di commercio di Bologna -
Capitale sociale 10.000€
R.E.A. 16653

AZIENDA

- Chi siamo
- Servizio clienti
- Dove spediamo
- Informazioni per l'acquisto

NEGOZIO

- New products
- Top sellers
- Specials
- Manufacturers
- Suppliers
- Our stores

ACCOUNT

- Log in
- Registrati

© 2017 EMG | Made in Nowhere

Figura 19 - Pagina prodotto. Sono state evidenziate le sezioni principali

Utilizzando la figura 21 come riferimento saranno discusse le parti principali della pagina.

La sezione contenente le tre miniature del prodotto è un elenco cliccabile delle immagini del prodotto. Se si clicca su una di queste, tramite jQuery sarà mostrata l'immagine selezionata all'interno della stessa pagina a dimensioni reali offuscando temporaneamente il contenuto della pagina per permettere di concentrarsi sulla foto.

Nella sezione attigua è stata inserita

l'immagine principale (img1 sulla tabella product_color nel database) del prodotto in modo da offrire un impatto visivo istantaneo di ciò che si sta acquistando e invogliare l'acquirente nel processo d'acquisto.

Nel riquadro in alto a destra e in quello subito sotto all'immagine principale sono state inserite le caratteristiche del prodotto, le auto compatibili (ove sia stata effettuata la ricerca intelligente) e la presenza a magazzino.

Il riquadro contenente il pulsante "ADD TO CART" contiene al suo interno un modulo con un campo numerico che definisce la quantità da inserire nel carrello. Cliccando sul tasto sarà aggiunta la quantità richiesta nel carrello.

Sono stati inoltre integrati, tramite appositi plug-in offerti da ThemeForest, la condivisione sui social, il tasto stampa e il tasto invia via e-mail.

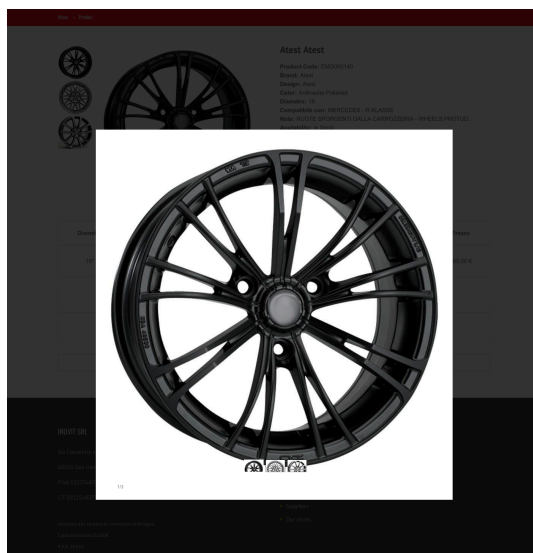


Figura 22 - Esempio di click sull'immagine prodotto

4.3.8 – Il carrello

La pagina carrello è l'elenco di tutti i prodotti aggiunti al carrello.

Il mio Account My Cart ✓ Checkout Benvenuto su EMG! login or create an account Italiano

EMG shop online

HOME PRODOTTI PERSONALIZZAZIONI CONDIZIONI DI VENDITA CONTATTI

(+39) 051 860161 info@emgcompany.it

8 item(s) - €1.320,00

Cart > Show Cart

Carrello



PRODOTTO	CODICE	PREZZO UNITARIO	QUANTITÀ	SUBTOTALE
 <p>Atest Marca: Atest Colore: Anthracite Polished ITA Diametro: 19" Canale: 9,0" ET: 50 PCD: 5x112 CB: 66,6 mm Vite / Dado: Esempio vite per Atest</p>	EMG000140	165,00 €	4pz	660,00 € <input type="button" value="x"/>
 <p>Atest Marca: Atest Colore: Hyper Silver - Chrome Inserts ITA Diametro: 19" Canale: 9,0" ET: 50 PCD: 5x112 CB: 66,6 mm Vite / Dado: Esempio vite per Atest</p>	EMG000144	165,00 €	4pz	660,00 € <input type="button" value="x"/>
SPEDIZIONE GRATUITA:		0,00 €		
SCONTO (0%):		0,00 €		
TOTALE:		1,320,00 €		

Figura 20 - Pagina carrello

Di questi sono riportate le caratteristiche, il prezzo e la quantità.

Moltiplicando il prezzo per la quantità si effettua il calcolo dei subtotali e del totale.

Cliccando sul tasto “CONTINUE SHOPPING” si verrà reindirizzati alla pagina di ricerca prodotti. Cliccando su “CHECK-OUT” si otterrà invece la pagina di check-out da cui effettuare il pagamento e completare l'acquisto.

4.3.9 – Check-out

La pagina di check-out è formata da vari step, che saranno validati lato browser tramite jQuery e lato server da PHP, come tutti gli altri moduli presenti sul sito. È buona pratica rispettare questa prassi per liberare capacità di calcolo sul server, oltre che ridurre la banda utilizzata (quantità di dati scambiati tramite Internet). Come richiesto dalle specifiche progettuali, è possibile acquistare anche senza aver effettuato precedentemente la registrazione utente, poiché il primo modulo è stato strutturato come in figura 24.

Nel caso si fosse già clienti è possibile effettuare il login, altrimenti è possibile continuare ed effettuare l'acquisto. Tale procedura d'acquisto prende il nome di guest checkout e si narra che l'implementazione abbia portato introiti fino a 300 milioni di dollari a un noto sito di e-commerce⁵⁴.

Infatti, utilizzando questo sistema, gli utenti sono più invogliati a comprare e possono evitare di doversi ricordare un'ulteriore password o tentare senza successo di accedere all'account esistente.

The screenshot shows the checkout page for EMG shop online. At the top, there is a navigation bar with links for 'Il mio Account', 'My Cart', and 'Checkout'. The main header features the EMG logo and contact information. Below the header, there is a red bar indicating the cart contents: '8 item(s) - €1.320,00'. The main content area is titled 'Checkout' and includes a progress bar with 6 steps. The first step, '1 Step: ACCEDI O CONTINUA', is active and contains two options: 'SEI GIÀ NOSTRO CLIENTE?' and 'NUOVO CLIENTE?'. The 'SEI GIÀ NOSTRO CLIENTE?' option has a 'LOGIN' button and a 'Hai dimenticato la password?' link. The 'NUOVO CLIENTE?' option has a 'CONTINUA' button. There are input fields for 'Email*' and 'Password*'. The top right corner shows 'Benvenuto su EMG! login or create an account' and 'Italiano'. The bottom right corner shows '8 item(s) - €1.320,00'.

Figura 21 - Pagina di checkout

⁵⁴ <http://www.seochef.it/guest-checkout-si-o-no/>

I vari step saranno abilitati tramite jQuery al click sul pulsante di conclusione della fase precedente.

Il secondo step riguarda l'inserimento dei dati personali e di fatturazione.

2 Step: INFORMAZIONI DI FATTURAZIONE

INFORMAZIONI SU DI TE

Nome* Il tuo nome

Cognome* Il tuo cognome

Cod. Fiscale* Il tuo codice fiscale

Email* La tua e-mail

Telefono* Il tuo numero di telefono

Società Ragione sociale

Partita IVA Inserisci la partita iva

INDIRIZZO DI FATTURAZIONE

Via / Numero* Il tuo indirizzo

Città* La tua città

Provincia* La tua provincia

Cap* Il tuo cap

Stato* ---

Sono d'accordo con la [Privacy Policy](#)

CONTINUA

Figura 22 - Pagina di checkout - Step 2

Il terzo step riguarda le informazioni di spedizione. In questa fase sarà possibile inserire un unico indirizzo per facilitare il processo di acquisto.

3 Step: INFORMAZIONI DI SPEDIZIONE

INDIRIZZO DI SPEDIZIONE

Il mio indirizzo di fatturazione è lo stesso indirizzo di spedizione.

Nome* Nome del destinatario

Cognome* Cognome del destinatario

Via / Numero* Indirizzo del destinatario

Città* Città del destinatario

Provincia* Provincia del destinatario

Cap* Cap del destinatario

Stato* ---

Telefono* Telefono del destinatario

C/O Consegna Presso

Note

CONTINUA

Figura 23 - Pagina di checkout - Step 3

Lo step 4 per ora non prevede diverse tipologie di consegna (es: spedizione semplice, spedizione con tracking, differenti corrieri ecc.), ma si è deciso di lasciarlo nel caso l'azienda cliente volesse decidere di utilizzarlo in futuro.

4 Step: METODO DI CONSEGNA

LA SPEDIZIONE È GRATUITA

Appena sarà disponibile la spedizione ti invieremo un'email con i dati relativi al tuo acquisto e le informazioni sulla spedizione

CONTINUA

Figura 24 - Pagina di checkout - Step 4 (Un unico metodo di consegna)

Il quinto step prevede la scelta della modalità di pagamento (PayPal o carta di credito). È possibile inviare scelte in cui l'opzione può essere solo una tramite l'elemento HTML radio button (i pulsanti rotondi) identificati da un codice simile al seguente:

```
<input name="payment_method" type="radio" value="paypal">
```

5 Step: MODALITÀ DI PAGAMENTO



SCEGLI TRA LE SEGUENTI LA MODALITÀ DI PAGAMENTO

- Paypal
- Carta di credito

CONTINUA

Figura 25 - Pagina di checkout - Step 5 (Modalità di pagamento)

L'ultimo step prevede un riassunto del carrello e il tasto "CONFIRM ORDER AND GO TO PAYMENT" per essere inviati alla pagina di PayPal ed effettuare il pagamento.

1 Step: ACCEDI O CONTINUA

2 Step: INFORMAZIONE DI FATTURAZIONE

3 Step: INFORMAZIONI DI SPEDIZIONE

4 Step: MEDOTODO DI CONSEGNA

5 Step: MODALITÀ DI PAGAMENTO

6 Step: CONFERMA ORDINE E TERMINA IL CHECKOUT

CONFIRM ORDER AND GO TO PAYMENT

PRODOTTO	CODICE	PREZZO UNITARIO	QUANTITÀ	SUBTOTALE
 <p>FIVE Marca: EMG Colore: Nero Lucido Diamantato Diametro: 18</p>	EMG000181	160,00 €	4pz	640,00 €
TOTAL:				640,00 €

Figura 26 - Pagina di checkout - Step 6. Conferma e vai al pagamento

4.3.10 – PayPal

L'implementazione di PayPal è stata effettuata utilizzando una vista che non prevede alcun tasto, che mostra soltanto l'immagine di PayPal e le scritte "Pagamento Paypal" e "Attendi...".

Il resto della pagina risulterà invisibile all'utente e il codice contenuto nella vista utilizzata è il seguente:

```
<div class="container">
  <div class="row">
    <div class="col-md-12" style="min-height:500px;">
      <header class="content-title">
        <h1 class="title">Pagamento Paypal</h1>
        Attendi...
      </header>
      
      <form name="formPayPal"
action="https://www.paypal.com/cgi-bin/webscr" method="post">
        <input type="hidden" name="cmd"
value="_xclick">
        <input type="hidden" name="business"
value="email@aziendale.it">
        <input type="hidden" name="item_name"
value="Ordine n {{session()->get('id_ordine')}}">
        <input type="hidden" name="item_number"
value="{{session()->get('id_ordine')}}">
        <input type="hidden" name="invoice"
value="{{session()->get('id_ordine')}}">
        <input type="hidden" name="return"
value="http://emgcompany.it/returnpaypal?paypal=return">
        <input type="hidden" name="cancel_return"
value="http://emgcompany.it/returnpaypal?paypal=cancellato">
        <input type="hidden" name="notify_url"
value="http://emgcompany.it/returnpaypal?paypal=ok">
        <input type="hidden" name="no_note" value="1">
        <input type="hidden" name="currency_code"
value="EUR">
        <input type="hidden" name="lc"
value="{{\$lang}}">
```



```

        <input type="hidden" name="bn" value="PP-
BuyNowBF">
        <input type="submit" id="ACQUISTA"
value="Proseguì al pagamento tramite PayPal">
    </form>
    </div> <!-- col-md-12 -->
</div> <!-- row -->
</div> <!-- container -->

```

Si è utilizzato come riferimento la pagina di PayPal developer⁵⁵ per lo sviluppo di tale vista.

Il tag `<form>`, utilizzato per l’invio di richieste alle pagine tramite i metodi HTTP, deve puntare alla pagina <https://www.paypal.com/cgi-bin/webscr> (attributo `action`) utilizzando il metodo POST (attributo `method`).

Gli input avranno tutti `type="hidden"` in modo da non rendere visibili le informazioni da passare alla pagina PayPal, come richiede PayPal stessa tra le specifiche.

Analizziamo adesso i principali parametri e valori di input:

- `name="cmd" value="_xclick"` indica che il cliente sta cliccando su un tasto `CompraSubito`, il che implica che se è già loggato su PayPal potrà effettuare il pagamento senza loggarsi, bensì cliccando un tasto dopo il redirect direttamente sul sito di PayPal;
- l’input avente la proprietà `name="business"` deve riportare l’email aziendale registrata su PayPal;
- gli input aventi la proprietà `name` uguale a `"item_name"`, `"item_number"` e `"invoice"` riporteranno l’id dell’ordine. Nel caso specifico è stato utilizzato Laravel che, grazie alla memorizzazione delle variabili della sessione corrente, stampa nei rispettivi attributi il valore dell’id dell’ordine utilizzando l’oggetto `session` e il metodo `get`;
- l’input con proprietà `name="return"` deve contenere nel valore della proprietà `value` l’URL della pagina a cui PayPal reindirizzerà il cliente dopo il pagamento andato a buon fine. Si tratta di una pagina semplicissima in cui si scrive soltanto `"Pagamento andato a buon fine"`;
- l’input con proprietà `name="notify_url"` ha nel parametro `value` l’URL di una pagina del sito `emgcompany.it` che PayPal contatta in background nel caso di pagamento andato a buon fine. Quando si riceve tale richiesta, in background si setta sul database l’ordine come pagato, il metodo di pagamento e il totale pagato dal cliente. Per la spedizione da parte

⁵⁵ <https://developer.paypal.com/docs/classic/paypal-payments-standard/integration-guide/formbasics/>

dell'azienda saranno utilizzati i dati nella tabella del carrello (shcart) prelevando solo le linee con proprietà id_ordine uguale all'ordine in sessione;

- l'input con proprietà name="cancel_ret" riporta in value l'URL del sito emgcompany.it a cui PayPal reindirizzerà il cliente nel caso di pagamento non andato a buon fine.

Tale vista sarà visualizzata solo per qualche istante dal cliente, poiché grazie a un semplice script jQuery inserito dopo del codice appena discusso, all'avvio della pagina si simula il click su un pulsante non visibile.

INOVIT srl

PayPal 🛒 560,00EUR ▾

Paga con PayPal Italiano ▾

Indirizzo email

Password

Rimani connesso e paga in modo più rapido ?

Accedi

[Problemi di accesso?](#)

oppure

Paga con una carta

Un metodo di pagamento più sicuro

Non importa dove fai acquisti, i tuoi dati sono più sicuri con PayPal e non vengono condivisi con i venditori.

[Annulla e torna sul sito di INOVIT srl.](#) © 1999 - 2017 🔒

[Accordi legali](#) [Privacy](#) [Feedback](#)

Figura 27 - Pagina di esempio di un pagamento inviato al sito PayPal

Il pulsante è identificabile dal codice:

```
<input type="submit" id="ACQUISTA" value="Proseguì al pagamento tramite PayPal">
```

Lo script jQuery per la simulazione del click:

```
<script>
$(document).ready(function(){
    $('#ACQUISTA').trigger('click');
});
</script>
```

4.3.11 – Carta di credito

La seconda modalità di pagamento è la carta di credito.

È stata richiesta la documentazione in banca, ma tale sviluppo sarà affidato a NoWhere S.R.L., che la integrerà all'interno del sito dove predisposto dalle specifiche progettuali (abilitando il l'input radio all'interno dello step 5 nel check-out e creando le relative pagine di pagamento andato a buon fine, annullato e di invio dati in background).

La procedura è simile a quella di PayPal, ma varia da banca a banca.

5 – Un sito Web di successo

5.1 – User-friendly

La scelta dell'aspetto grafico di un sito Web è basata su attività apparentemente semplici, come scrivere testi, scegliere colori, selezionare gli argomenti da trattare.

In verità queste attività apparentemente banali, che richiedono competenze specifiche, fanno la differenza fra un sito Web efficace e uno meno.

Ma cosa deve avere un sito Web per spiccare in un ambiente con un così alto tasso di concorrenza?

È fondamentale garantire una navigazione intuitiva, quindi dare all'utente la possibilità di trovare ciò che gli serve nel minor numero di click.

Per ottenere questo risultato è bene strutturare correttamente la gerarchia dei contenuti e organizzare in maniera intelligente i menu.

Inoltre, bisogna studiare un design che rispecchi la realtà aziendale nei colori, nelle immagini e nei contenuti.

È sempre consigliato utilizzare un layout pulito, ma non privo di contenuto, dando maggior risalto alle informazioni più importanti.

I contenuti sono molto importanti anche per l'indicizzazione nei motori di ricerca, oltre che per l'esperienza utente.

Proprio in merito all'utente, è bene pensare quali azioni cerchi di intraprendere più spesso sul sito simulandone il comportamento e aggiornare le pagine a seguito dello sviluppo, se tale comportamento varia.

Quando navighiamo sul Web, per buona parte del tempo eseguiamo azioni a noi familiari (utilizziamo i social, leggiamo gli articoli pubblicati sul nostro blog preferito, ascoltiamo la musica).

Capita però che si cerchino informazioni nuove su siti nuovi e si ha fretta di reperirle mentre si impara a usare per la prima volta la nuova piattaforma.

Se il sito risulta essere intuitivo, ovvero *user-friendly*, tale processo è agevolato. Come ci si può aspettare, non è sempre così. Spesso visitando alcuni siti ci rende conto che risultano poco intuitivi. In tal caso ci sarà un'alta frequenza di rimbalzo, ovvero un'alta percentuale di utenti che abbandona il sito

Dunque, come suggerisce Steve Krug, autore del libro “Don’t make me think“, chi approda su un sito non deve porsi domande su come fare o trovare ciò a cui è interessato: "l'internauta non dovrebbe avere nemmeno il tempo di pensare, ma riuscire a trovare quasi a colpo d'occhio ciò che sta cercando."

Per rendere facilmente utilizzabile un sito occorre dare un nome non ambiguo alle sezioni del menu, far percepire visualmente l'idea che un certo link è cliccabile e fornire delle opzioni di ricerca agevolanti in base ai contenuti del sito.

5.2 – SEO

Oggigiorno siamo circondati da un'infinità di acronimi, ma chi lavora nel Web sa di cosa si parla quando ci si riferisce al SEO.

L'acronimo *SEO* sta per Search Engine Optimization, ovvero l'ottimizzazione del sito per l'indicizzazione dei motori di ricerca ed è una delle aree di interesse del Web Marketing.

Di seguito saranno riportate le specifiche che Google suggerisce per una corretta indicizzazione⁵⁶:

- il titolo deve comunicare gli argomenti della pagina sia agli utenti che ai motori di ricerca tramite il tag <title>, che dovrebbe essere posizionato all'interno del tag <head> nel codice HTML. Possibilmente sarebbe bene utilizzare un titolo univoco per ogni pagina del sito. La lunghezza non deve superare i 55 caratteri, altrimenti verrà troncato da Google e la pagina non correttamente indicizzata. Il titolo deve contenere keyword significative inserendo all'inizio quelle più importanti, non rendendolo comunque soltanto un elenco di parole chiave. Ovviamente per i siti dinamici il processo di assegnazione titoli viene effettuato in maniera dinamica;
- il meta tag description deve fornire una descrizione della pagina. Il contenuto di questo, insieme al tag title compariranno all'interno di Google nella schermata dei risultati della ricerca. La descrizione della pagina non deve superare i 150 caratteri;
- gli URL non devono essere troppo complessi e, se possibile, è bene che siano "amichevoli" e senza l'utilizzo di richieste GET;
- ogni pagina dovrebbe avere un tag H1 univoco, con informazioni molto significative sul contenuto della pagina;

⁵⁶ <http://static.googleusercontent.com/media/www.google.com/it/intl/it/webmasters/docs/search-engine-optimization-starter-guide-it.pdf>

- altra caratteristica di cui tener conto è l'attributo alt delle immagini. È bene spiegare cosa sia l'immagine in modo tale che lo spider possa capire di cosa si tratta (es.);
- in una pagina è necessario che ci sia del testo descrittivo affinché sia ben indicizzata;
- è necessario creare una Sitemap del sito da inviare a Google (di solito si utilizza un software per creare tale mappa per i siti dinamici);
- utilizzare un hosting che offra bande (velocità di trasferimento dati) elevate per un caricamento rapido della pagina non è solo gradevole per gli utenti, ma cambia il ranking di indicizzazione del sito per il motore di ricerca;

Altri strumenti utilizzabili per l'indicizzazione sono:

- Google webmaster: permette di ritrovare problemi che influirebbero sull'indicizzazione all'interno delle pagine, come title duplicati;
- file "Robots.txt": file da inserire all'interno della directory di root per spiegare ai motori di ricerca quali pagine indicizzare;
- Google analytics: effettua statistiche sulle pagine (anche in tempo reale), offre informazioni sul numero di visualizzazioni, le parole chiave più ricercate quando si clicca sul nostro sito, le nazioni più connesse, la frequenza di cambio pagina e molto altro.

È bene evidenziare che i siti che sono da più tempo online hanno un ranking più alto. Questo grazie all'affidabilità che hanno dimostrato nel tempo continuando a restare online.

Quindi, la corretta gestione del SEO è verificabile solo dopo un utilizzo abbastanza prolungato nel tempo da parte degli utenti in rete.

Conclusioni

Il commercio elettronico porta alle aziende numerosi benefici come la riduzione dei costi, l'internazionalizzazione delle vendite per raggiungere più facilmente i mercati esteri, un feedback immediato al cliente e la possibilità di acquistare in orari non ordinari.

I dati parlano chiaro: il numero di utenti che acquista utilizzando siti di e-commerce è in aumento.

In Italia, i settori più sviluppati nell'e-commerce sono tempo libero e turismo, e sono in crescita la moda e l'arredamento.

È bene notare, però, che la competitività online è elevata e un e-commerce deve misurarsi con le nuove tendenze ed essere pronto ad adattarsi alle richieste del mercato in termini di prezzo, prodotti e finanche nelle strategie di business.

INOVIT S.R.L. rispecchia la classica PMI italiana, in ritardo nella vendita online, che ha cominciato da poco a muoversi in questa direzione, percependo la possibilità di incrementare i volumi di vendita e il fatturato.

L'azienda ha scelto quindi di lanciare emgcompany.it, che è stato progettato in base alle funzionalità desiderate dal cliente e alle caratteristiche dei prodotti aziendali.

La progettazione e la creazione del sito hanno richiesto l'importazione dei database precedentemente utilizzati dall'azienda. Successivamente è stato creato il back-end per la gestione del sito da parte dell'impresa e il front-end per l'utilizzo del sito da parte dell'utente finale.

Per competere online con gli strumenti adeguati, si è sviluppato il front-end tenendo conto delle caratteristiche per migliorare la visibilità sui motori di ricerca (SEO). Oltre alla visibilità online, si è sviluppato un sito con un'interfaccia utente semplice e di bell'impatto visivo.

Inoltre, per promuovere la diffusione di informazioni online si è pensato all'integrazione dei social, che permetterà al sito di ottenere un'ulteriore visibilità sul Web.

Attualmente, puntare alla compatibilità con i dispositivi mobili è necessario, poiché il numero di utenti che acquista da smartphone e tablet è in aumento. Per

tale ragione, si è utilizzato un template con design responsive, in modo che l'acquirente possa comprare agevolmente anche tramite dispositivi mobili.

Per incrementare i volumi delle vendite si è integrato il modello di vendita online detto guest-checkout, che permette all'utente di acquistare senza dover effettuare la registrazione in precedenza.

Per poter permettere a chiunque di acquistare, si è scelto di offrire all'utente le modalità di pagamento più utilizzate online (PayPal e carta di credito).

Alla luce di una progettazione attenta e un'implementazione corretta, l'e-commerce di INOVIT S.R.L. è stato ben strutturato per potersi affermare online nella vendita dei cerchi in lega per auto.

Bibliografia

- [1] Stephen Walther. *Commercio elettronico con ASP* Apogeo (2000)
- [2] Vincenzo Della Mea. *Programmazione lato server*. Apogeo (2011)
- [3] Peter Wainwright. *Professional Apache 2.0*. Apogeo (2003)
- [4] Matthew MacDonald. *Creare siti Web*. Hops Tecniche nuove (2011)
- [5] Peter MacIntyre. *Pro PHP programming*. Apogeo (2012)
- [6] Eric Larson. *Web servers: security & maintenance*. Tecniche nuove (2000)
- [7] DB-Engines Ranking
(<https://db-engines.com/en/ranking>)
- [8] Stefano Ceri. *Designing data-intensive web applications*. McGraw-Hill (2003)
- [9] Associazione italiana produttori italiani di ruote in lega leggera e in acciaio (http://assoruote.it/wp-content/uploads/2015/03/AssouoteLeRuotepp_02.2015.pdf)
- [10] Dave Chaffey. *Internet Marketing*. Apogeo (2001)

I miei ringraziamenti vanno anzitutto alla Professoressa Wilma Penzo, che mi ha supportato e seguito dall'inizio del tirocinio fino alla stesura della tesi.

Ringrazio l'azienda NoWhere, soprattutto Kevin, Monica, Francesca, Giulia, Andrea, Fabio e Leonardo, con cui ho condiviso il tempo trascorso in azienda e che mi hanno aiutato nello sviluppo di questo lavoro, oltre che accolto con gentilezza e professionalità. Un particolare ringraziamento va a Davide, che ha dimostrato una grande disponibilità nei miei confronti.

Dedico questa tesi ai miei genitori che hanno creduto in me e mi hanno sempre incoraggiato. Grazie Papà e Grazie Mamma.

Grazie a mia sorella, Mirella, perché porta allegria nelle mie giornate tristi e mi aiuta nei momenti di smarrimento.

Ringrazio tutti i miei parenti, che alle riunioni familiari anche in posti sperduti mi fanno sentire come se fossi a casa. Un grazie speciale va a mia zia Rina, che si preoccupa costantemente per me e mi vuole un bene dell'anima.

Ringrazio tutti i miei amici, quelli che vedo tutti i giorni e quelli che vedo poco ma intensamente, per gli aperitivi e le serate che fanno scorrere la vita veloce e ti fanno stare bene.

Grazie.

Filippo