ISTITUTO TECNICO COMMERCIALE STATALE
“ *Cav. Ing. ADRIANO OLIVETTI* ”
LECCE

SYMPHONIC RECORDS

Progettazione e realizzazione di un sito web di e-commerce

Candidato:
Davide Quarta
classe V A/MERCURIO

ESAME DI STATO - A.S. 2006-2007

Indice generale

NOTA INTRODUTTIVA

L'idea di questo progetto è nata da una proposta curricolare di lavoro gruppo della docente di informatica - prof.ssa Filomena Smacchia, e dal mio particolare interesse nei confronti dei diversi temi ed argomenti presenti in questo lavoro. Temi, tra l'altro, strettamente connessi al complesso utilizzo da parte delle aziende di tecnologie informatiche per lo svolgimento della loro attività. In particolare l'implementazione di un'attività di commercio elettronico necessita di supporti tecnico-informatici come le basi di dati e i "siti web" che permettono all'azienda di organizzare in modo efficiente ed efficace le loro attività economico-commerciali e facilitare per i servizi offerti l'interazione da parte degli utenti. Nell'attività di e-commerce, la "sicurezza informatica" svolge un ruolo di primaria importanza, in quanto fornisce un presupposto che sviluppi fiducia da parte degli utenti, d'altra parte i "dati sensibili" come i numeri delle carte di credito e i dati personali nelle transazioni devono avere un canale di comunicazione sicuro. Internet infatti non è nata per lo scambio di questa tipologia di informazioni, ma come rete pubblica che favorisca una comunicazione libera ed aperta fra i suoi utenti. Proprio questo fatto ha portato allo sviluppo di nuove tecnologie che permettano uno scambio sicuro dei dati anche sulla infrastruttura già presente di Internet, come le comunicazioni criptate tramite SSL (Secure Socket Layer). È altresì importante che l'applicazione web stessa sia sicura, pertanto occorre fare molta attenzione in fase di progettazione e di sviluppo dell'applicazione in questo campo, per evitare che errori di programmazione o "sviste" possano compromettere la sicurezza, offrendo ai malintenzionati un possibile canale attraverso cui fare danni o peggio entrare in possesso delle informazioni ritenute sensibili.

IL DATABASE

DESCRIZIONE DEGLI APPLICATIVI

La versione iniziale del database è stata realizzata utilizzando l'applicativo Microsoft Access come DBMS. Per la gestione del database si è deciso in seguito di passare all'utilizzo di strumenti open source e free software.
“OpenOffice.org Base” è stato utilizzato per avere una visuale generale del database, delle tabelle e delle relazioni che intercorrono tra esse. Questo applicativo fornisce semplicità di utilizzo e permette uno sviluppo visuale. Esso è un ambiente di sviluppo integrato, che fornisce oggetti per lo sviluppo e dà la possibilità di interagire attraverso il linguaggio SQL.
Permette inoltre una semplice interazione con l’utente finale attraverso l’uso di maschere e report.
Si è connesso il database di acces tramite ODBC in modo da poter visualizzare tabelle e relazioni e rendere lo sviluppo più semplice e rapido.

NOTA: per informazioni sugli applicativi usati per lo sviluppo dell’interfaccia web vedere la relativa sezione in “Il sito web”

ANALISI DEL PROBLEMA

La casa discografica può essere considerata una entità in quanto l’utente potrebbe richiedere informazioni dettagliate quali la sede, numero di telefono/fax, e-mail.

La casa può avere diversi negozi (“sedi”) in cui vende i propri CD e pertanto bisogna fornire agli utenti le informazioni relativi alle varie sedi, in particolare la località in cui si trovano, in modo che l'utente possa cercare un negozio in una determinata località preferenziale.

Per l'accesso al servizio il potenziale utente è necessario che fornisca, oltre a i dati anagrafici, anche uno “*username*” (nome utente) e una “password”. Il fatto di fornire i dati anagrafici dà la possibilità di ottimizzare se non addirittura di automatizzare la ricerca dei negozi nella propria zona di residenza.

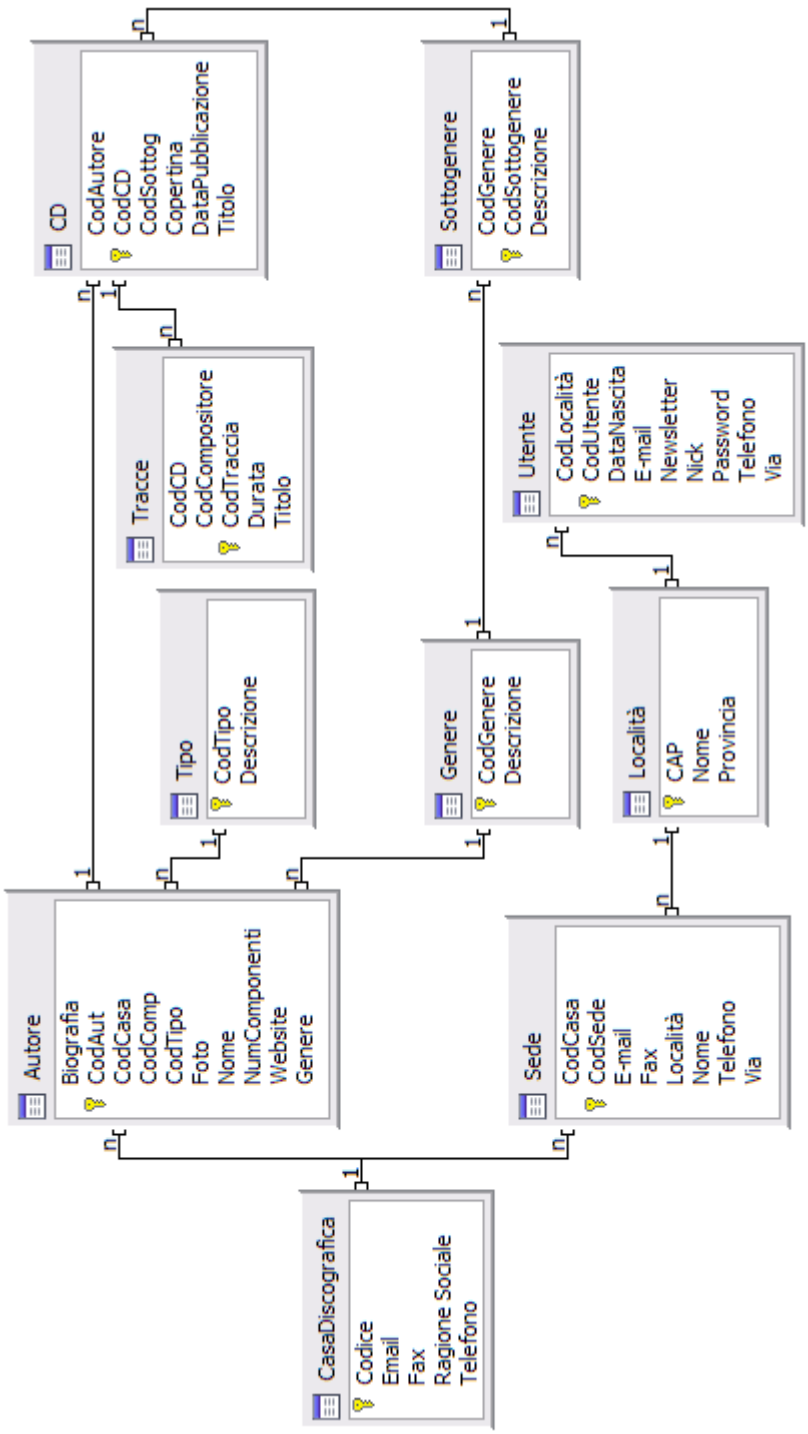
La casa effettua contratti con gli autori che creano dei dischi per la casa stessa.
Si considera che la categoria “autore” del disco possa essere tipicizzata (entità “tipo”)

come “solista”, “gruppo” o “orchestra”. Vengono presi in considerazione due generi principali: “musica classica” e “musica leggera”, suddivisi in più sottogeneri: un autore infatti può produrre musica di un certo genere, mentre un CD può essere classificato in un solo sottogenere.

Avendo ogni CD una data di pubblicazione sarà possibile ordinare gli stessi in base alla data di pubblicazione. Va aggiunto inoltre l’attributo “compositore”, nel caso della musica classica il compositore potrebbe essere diverso per ogni traccia..

Viene anche contemplata la possibilità di inserire l’immagine della copertina e una foto dell’autore per questo nell’entità CD verrà incluso l’attributo “copertina” e nell’entità autore l’attributo “foto”.

SCHEMA E/R


CD
CodAutore
CodCD
CodSottog
Copertina
DataPubblicazione
Titolo
Sottogenere
CodGenere
CodSottogenere
Descrizione
Tracce
CodCD
CodCompositore
CodTraccia
Durata
Titolo
Utente
CodLocalità
CodUtente
DataNascita
E-mail
Newsletter
Nick
Password
Telefono
Via
Tipo
CodTipo
Descrizione
Genere
CodGenere
Descrizione
Località
CAP
Nome
Provincia
Autore
Biografia
CodAut
CodCasa
CodComp
CodTipo
Foto
Nome
NumComponenti
Website
Genere
Sede
CodCasa
CodSede
E-mail
Fax
Località
Nome
Telefono
Via
CasaDiscografica
Codice
Email
Fax
Ragione Sociale
Telefono

MODELLO LOGICO

LEGENDA:
PK = sottolineato
FK = *corsivo*

casadiscografica (codice, ragionesociale, sede, telefono, fax, email)
autore (codautore,nome,foto,biografia,numcomponenti,website,*genere*,*codcasa, codtipo*)
tipo (codtipo,descrizione)
cd (codcd,titolo,duratacomplessiva,datapubblicazione,copertina,prezzo,*codsottog*,*codautore*)
tracce (codtraccia, titolo, durata, compositore, c*odcd)*
genere (codgenere, descrizione)
sottogenere (codsottogenere,descrizione,*codgenere*)
utente (codutente,username,password,e-mail,datadinascita,newsletter)
località (cap,nome,provincia)
sede (codsede, email, fax, localita, nome, telefono, via, *codcasa*)
tracce (codtraccia, durata, titolo, compositore, *codcd*)

CASADISCOGRAFICA	TIPO	CHIAVE
codice	int (10) unsigned auto-increment	PK
ragionesociale	varchar (255)	
sede	varchar (255)	
telefono	varchar (255)	
fax	varchar (255)	
email	varchar (255)	

AUTORE	TIPO	CHIAVE
codautore	int (10) unsigned auto-increment	PK
nome	varchar (255)	
foto	varchar (255)	
numcomponenti	int (5)	
website	varchar (255)	
biografia	mediumtext	
genere	int (10) unsigned	FK
codcasa	int (10) unsigned	FK
codtipo	int (10) unsigned	FK

TIPO	TIPO	CHIAVE
codtipo	int (10) unsigned auto-increment	PK
descrizione	Varchar (255)	

CD	TIPO	CHIAVE
codcd	int (10) unsigned auto-increment	PK
titolo	varchar (255)	
duratacomplessiva	Int (10) unsigned	
datapubblicazione	Int (10) unsigned	
prezzo	double	
copertina	Varchar (255)	
Codautore	int (10) unsigned auto-increment	FK
codsottog	int (10) unsigned auto-increment	FK

TRACCE	TIPO	CHIAVE
codtraccia	int (10) unsigned auto-increment	PK
durata	float	
Titolo	Varchar (255)	
compositore	Varchar (255)	
codcd	int (10) unsigned	FK

GENERE	TIPO	CHIAVE
codgenere	int (10) unsigned auto-increment	PK
descrizione	Varchar (255)	

SOTTOGENERE	TIPO	CHIAVE
codsottogenere	int (10) unsigned auto-increment	PK
descrizione	Varchar (255)	
codgenere	int (10) unsigned auto-increment	FK

UTENTE	TIPO	CHIAVE
codutente	int (10) unsigned auto-increment	PK
nome	Varchar (255)	
cognome	Varchar (255)	
datanascita	int (10) unsigned	
telefono	Varchar (255)	
via	Varchar (255)	
username	Varchar (255)	
password	Varchar (255)	
email	Varchar (255)	
codlocalita	int (10) unsigned	FK

LOCALITÀ	TIPO	CHIAVE
cap	int (10) unsigned auto-increment	PK
nome	Varchar (255)	
provincia	Varchar (255)	

SEDE	TIPO	CHIAVE
codsede	int (10) unsigned auto-increment	PK
email	Varchar (255)	
fax	Varchar (255)	
nome	Varchar (255)	
telefono	Varchar (255)	
via	Varchar (255)	
codcasa	int (10) unsigned	FK
localita	int (10) unsigned	FK

NOTA: nel formatoMyIsam delle tabelle MySql (il formato più utilizzato) le chiavi esterne e quindi le relazioni non sono supportate; sono state incluse lo stesso nella documentazione perché erano previste nel progetto iniziale sviluppato con Access, e possono dare una visione migliore delle relazioni che legano le tabelle.

REGOLE DI LETTURA

CasaDiscografica – Autore:

1:M diretta obbligatoria: una casa discografica deve effettuare un contratto con uno o più autori

1:1 inversa obbligatoria: un autore deve aver effettuato un contratto con una casa discografica

Autore – CD:

1:M diretta facoltativa: un autore può produrre uno o più CD

1:1 inversa obbligatoria: un CD deve essere prodotto da un solo autore

Autore – Genere:

1:1 diretta obbligatoria: un autore deve suonare un solo genere

1:M inversa facoltativa: un genere può essere suonato da uno o più autori

Autore – Tipo:

1:M diretta facoltativa: un tipo può descrivere uno o più autori

1:1 inversa obbligatoria: un autore deve essere descritto da un solo tipo

CD – Tracce:

1:M diretta obbligatoria: un CD deve contenere una o più tracce

1:1 inversa obbligatoria: una traccia deve essere contenuta in un CD

CD – Sottogenere:

1:M diretta facoltativa: un sottogenere può descrivere uno o più CD

1:1 inversa obbligatoria: un CD deve essere descritto da un sottogenere

Sottogenere – Genere:

1:M diretta obbligatoria: un genere deve descrivere uno o più sottogeneri
1:1 inversa obbligatoria: un sottogenere deve essere descritto da un solo genere

CasaDiscografica – Sede:
1:M diretta facoltativa: una casa discografica può possedere una o più sedi
1:1 inversa obbligatoria: una sede deve essere posseduta da una sola casa discografica

Sede-Località:
1:1 diretta obbligatoria: una sede deve risiedere in una sola località
1:M inversa facoltativa: in una località possono risiedere una o più sedi

Utente-Località:
1:1 diretta : un utente deve risiedere in una sola località
1:M inversa : in una località possono risiedere uno o più utenti

IL SITO WEB


Symphonic Records
Bentornato webmaster
HomePage
I Nostri CD
Cerca CD
Cerca Pezzo
Cerca Negozi
Chi siamo
Logout
Pannello Webmaster
Inserisci CD
Inserisci Autore
Inserisci Sede
W3C XHTML 1.1
W3C CSS
Gli ultimi CD che abbiamo pubblicato:
the jazz beat - street jazz, durata totale: 8.20, pubblicato il 07-05-07, 4:41
new orchestra - romantic piano, durata totale: 49.32, pubblicato il 05-06-07, 12:00
new orchestra - classic guitar, durata totale: 12.20, pubblicato il 07-06-07, 11:40
| contatti | privacy | condizioni d'uso |

DESCRIZIONE DEGLI APPLICATIVI

Per lo sviluppo dell'applicazione web, è stato usato il linguaggio server side PHP, associato a MySQL (un dialetto del linguaggio SQL) e a XHTML/CSS. La creazione di pagine e script è stata effettuata utilizzando PSPad un editor di testo gratuito che fornisce la visualizzazione con syntax highlighting per diversi linguaggi, e l'indentazione HTML automatica, e altri strumenti che hanno semplificato lo sviluppo come il "traduttore colori".

Per testare l'applicativo in ambiente locale si è fatto uso di MyPHP che include in un unico pacchetto una installazione semplice da effettuare per il web server Apache, l'interprete PHP, MySQL server e PHP MyAdmin, anche se quest'ultimo non è stato utilizzato perché si è preferito lavorare da linea di comando tramite MySQL.

Gli elementi grafici del sito web sono stati creati con il software opensource GIMP 2.

DESCRIZIONE DELLA STRUTTURA

Come struttura per il sito è stato scelto un layout table-less fisso a due colonne. Table-less significa che è realizzato senza tabelle, facendo invece uso di css e dei ***div***. Ogni sezione della struttura è stato associata a un div.

Il modello del sito prevede:

- un header dove è presente il logo e la form per il login dell'utente
- il menu di navigazione nella prima colonna a sinistra
- una sezione con i contenuti nella colonna a destra
- un footer (pie' di pagina) dove sono presenti dei link alle informazioni sulla casa discografica (contatti), alle informazioni sulla privacy e ai termini d'uso del servizio.

Per questo layout fisso è stata scelta una dimensione del contenitore principale di 760 px in modo da essere adatto alla navigazione anche a risoluzioni basse (800x600) e risultare maggiormente leggibile.

Il layout a due colonne è stato creato con i float, utilizzando una spaziatura fissa.
Per i popup è stato invece utilizzato un layout monolitico.

Dopo aver pianificato i contenuti del sito tramite una bozza su carta, lo sviluppo di html e css è stato eseguito parallelamente.

Dalla struttura di base sono state effettuate in seguito diverse modifiche, contestualmente allo sviluppo del codice php per ottenere una struttura del codice flessibile.

Per garantire una differenziazione tra contenuti e stile è stato utilizzato lo standard XHTML 1.1, oltretutto, per garantire una buona accessibilità, pagine e css sono stati validati tramite i validatori w3c (www.w3.org).

Il css principale è “style.css”, per i popup é “cdinfo.css” (nella cartella doinfo)

File style.css

```
/*proprietà di html e body e generali*/
html, body {
  margin : 0;
  padding : 0;
}
body {
  font-family : arial, sans-serif;
  font-size : 82%;
  text-align : center;
  background : #ffffc5;
  color : #000;
}
h1, h2 {
  margin : 0;
  padding : 0;
}
h1 {
  padding-left : 0.5em;
  font : bold 1.8em/2em arial, sans-serif;
  color: #162046;
}
h2 {
  font-size : 1.2em;
}
/*qui è definito il container la larghezza è definita tramite width
per il background invece è stata creata una bitmap di larghezza
fissa tramite cui si disegna parte del separatore tra le sezioni*/
div#container {
  width : 760px;
  margin : 0 auto;
  text-align : left;
  border-left : 2px dotted #328;
  border-right : 2px dotted #328;
  border-top : 2px solid #328;
  border-bottom : 2px solid #328;
  background: #e6f0ff url(immagini/xdc.gif) repeat-y top left
}
/*l'hader con il logo e l'altezza fissa in pixel uguale a quella
del logo*/
div#header {
  background : url(immagini/logo.jpg) no-repeat;
  border-bottom : 2px solid #328;
  height : 100px;
  color : #fff;
}
/*il form di login posizionato tramite float sulla destra
all'interno dell'header*/
div#toplogin {
  float : right;
  padding-top : 18px;
  margin-right : 10px;
```

```
  text-align : right;
  font : 0.8em/2.2em arial, sans-serif;
  color : rgb(40, 40, 75);
  background-color : transparent;
}
/*la barra di navigazione posta tramite float a sinistra e con una
 larghezza fissa di 160 pixel*/
div#navigation {
  float : left;
  width : 160px;
  background : #e6f0ff;
  background: url(immagini/xdc.gif) repeat-y top left
}
/*i contenuti con un margine sinistro uguale alla larghezza
 della barra di navigazione*/
div#content {
  margin-left : 160px;
  padding : 1em;
background : #e6f0ff
}

/*stile per la lista*/
div#content ul {
float: left;
border-top: 1px dotted #328;
border-bottom: 1px dotted #328;
margin-left: 1em;
padding: 0;
list-style-type: none;
}
div#content ul li a {
background : transparent url(immagini/cdlist.gif) left center no-
repeat;
padding-left : 20px;
text-align: left;
text-decoration: none;
font: 1em/2em arial,sans-serif;
color : #00e;
}
div#content ul li a:hover {
background : transparent url(immagini/cdlist.gif) left
 center no-repeat;
text-align: left;
text-decoration: none;
font: 1em/2em arial,sans-serif;
color : #FF8000;
}

/*qui il footer tramite clear: left si fa in modo che gli elementi
float (in questo caso la barradi navigazione) non vadano a
sovrapporsi al footer*/
div#footer {
  clear : left;
  text-align : center;
```

```
  background : #136;
  color : #fff;
  border-top : 2px solid #328;
}

/*lo stile per i vari link*/
div#footer a {
  font-weight : bold;
  color : #ffe;
  text-decoration : none;
}
div#footer a:hover {
  font-weight : bold;
  text-decoration : none;
  color : #FF8000;
}
div#navigation ul {
  margin : 1em 0 1em 1em;
  padding : 0;
  list-style-type : none;
}
div#navigation li {
  margin : 0;
  padding : 0;
}
div#navigation a {
  color : #FF8000;
  font : bold 1em/1.6em arial, sans-serif;
  text-decoration : none;
}
div#navigation a:hover {
  color : #A65106;
  text-decoration : none;
}
div#navigation a#activelink {
  color : #777;
  text-decoration : none;
}

/*classi per diversi elementi*/
.info {
  padding-top : 20px;
  color : #237;
  font : bold 110% arial, sans-serif;
}
.usr {
  text-align: center;
  color : #FF3000;
  font : bold 90%/3em arial, sans-serif;
  border-top: 1px dashed #FF8000;
  border-bottom: 1px dashed #FF8000;
}
.valid {
  margin : 0;
```

```
  padding : 3px;
  border : none;
}
.validbox{
    margin : 0;
    padding : 5px;
    border-top: 1px dashed #FF8000;
    text-align: center;
}
.login{
  font: 1.1em arial,sans-serif;
  text-align: right;
}
.loginbutton{
  font: 1.1em arial,sans-serif;
}
```

File cdinfo.css

```
/*stile per il body e la pagina in generale*/
html, body {
  margin : 0;
  padding : 0;
}
body {
  font-family : arial, sans-serif;
  font-size : 82%;
  text-align : center;
  background : #ffffc5;
  color : #000;
}
h2 {
  font-size : 1.2em;
  padding: 0.5em;
  margin: 0;
}
/*div per il contenitore per l'intera pagina (contiene le sezioni
della pagina header conenuti e footer);*/
div#container {
  text-align : left;
  background: #e6f0ff;
  border : 1px solid #FF8000;
}
/*div per l'header*/
div#header {
  padding : 0.2em;
  background : #136;
  color : #fff;
  text-align : center;
  border-bottom : 1px dashed #FF8000;
  font: 1em/1.5em arial, sans-serif;
}
```

```
/*div per il footer*/
div#footer {
  padding : 0.2em;
  background : #136;
  color : #fff;
  text-align : center;
  border-top : 1px dashed #FF8000;
  font: 1em/1.5em arial, sans-serif;
}
/*div per il contenuto (contiene le sezioni della schedacd e
tracce*/
div#content {
padding : 20px;
background : #e6f0ff url(bg.gif);
font: 1em/1.6em arial, sans-serif;
}
/*div per la scheda del cd*/
div#schedacd {
margin : 0;
padding-top : 0.8em;
padding-bottom : 1.3em;
padding-left: 1em;
padding-right: 0;
background : transparent;
border : 1px solid #000;
height : 100px;
font: bold 1.1em/1.5em arial, sans-serif;
}
div#schedacd a{
text-align: left;
text-decoration: none;
color : #23e;
}
div#schedacd a:hover{
text-align: left;
text-decoration: none;
color : #FF8000;
}
.copertina{
float : left;
margin: 0.2em 1.5em 0 0.5em;
padding: 0;
border : 0;
width : 100px;
height : 100px;
border : 1px solid #000;
}
/*div per la scheda tracce*/
div#schedatracce {
margin-top : 0.5em;
padding: 0.5em 1em 1em 1em;
background : transparent;
border : 1px solid #000;
}
```

```
div#schedatracce ul {
margin: 0;
padding-left: 1em;
list-style-type: none;
}
div#schedatracce ul li {
margin: 0;
background : transparent url(cdlist.gif) left center no-repeat;
padding-left : 20px;
text-align: left;
text-decoration: none;
font: 1em/1.6em arial,sans-serif;
color : #00e;
}
div#schedatracce ul li:hover {
background : transparent url(cdlist.gif) left center no-repeat;
text-align: left;
text-decoration: none;
font: 1em/1.6em arial,sans-serif;
color : #FF8000;
}
```

IL CODICE PHP E LE QUERY SQL

Nella cartella possiamo notare il file start.php, che è il primo file da utilizzare. Tramite questo file vengono fornite le informazioni per l'account di amministratore, che vengono poi passate tramite post al file install.php, questo si occupa di creare il database e le tabelle e in seguito inserisce i dati dell'account di amministratore.
Nel file config.inc.php ci sono i parametri per la connessione al database, e come da commento la variabile boolean $external_hosting è da utilizzare in quanto alcuni database che vengono offerti online non danno la possibilità di creare il database perchè già presente sul server.
Prima di effettuare le query, il codice php controlla questa variabile, se è vera non effettua operazioni come drop e create poiché non sono comunque permesse.

Si passa quindi al file index.php, dove si trovano una variabile e degli *include* presenti nella maggior parte delle pagine:

```
<?php $page = basename(__FILE__);
  include('connect.inc.php');
  include('checklogin.php'); ?>
```

La variabile $page indica il nome della pagina.

Nel file connect.inc.php troviamo la funzione connecttodb() , che effettua la connessione al database e ritorna la risorsa.

File connection.inc.php

```
<?php
      //effettua la connessione al database e ritorna la risorsa
      function connecttodb()
      {
        include('config.inc.php');

   $db = mysql_connect($db_host, $db_user, $db_password);
//connetti al database
          if ($db == false)
          {
             echo $connection_error."Dettagli errore :
".mysql_error();
             include('footer.php');
             exit();
          }
          if(!mysql_select_db($db_name))
          {
            echo $dbselect_error."Dettagli errore :
".mysql_error();
            include('footer.php');
```

```
            exit();
          }
      return $db;
      }
?>
```

Il file checklogin si occupa di caricare i dati dal cookie utilizzato per mantenere le informazioni di login, e li controlla successivamente con le righe presenti nel database. Se l'utente è presente nel database il codice inserisce nelle variabili i dati richiesti.

File checklogin.php

```
<?php
//controlla il login degli utenti e registra in variabili i dati
che servono
$db = connecttodb();
$logged = false;
//@ serve a non mostrare gli errori prodotti in caso non ci siano
cookie
@list($username,$password) = @explode("|",$_COOKIE['loginfo']);
//controlla i dati in input tramite espressioni regolari
if(isset($username)&&isset($password)&&@ereg("^[a-fA-F0-
9]+$",$password)&&@ereg("^[a-zA-Z0-9]+$",$username))
{
$query = "SELECT codutente, username FROM utente
          WHERE (username='$username') AND (password='$password')";
$result = mysql_query($query);
if($result == FALSE)
{
  echo mysql_error();
  exit();
}
$row = mysql_fetch_array($result, MYSQL_ASSOC);

mysql_close($db);

//controlla se è una delle pagine a cui può accedere solo il
webmaster se lo è
//controlla anche che l'account sia quello del webmaster altrimenti
annulla
//il caricamento della pagina dando un messaggio di errore e
chiamando exit()
 if(($page=='insaut.php'||$page=='inscd.php'||
$page=='inssede.php'||$page=='instraccia.php')
      &&($row==FALSE||$row['codutente']!='1') )
 {
      echo "<h1>Utente non autorizzato.</h1>Non sei autorizzato a
vedere la pagina che hai richiesto.";
      exit();
 }
```

```
 //l'utente è registrato, salva i dati importanti nelle variabili
 if($row==TRUE)
 {
    $codiceutente = $row['codutente'];
    $utente = $row['username'];
    $logged = true;
 }
 else
 {
  //cancella il cookie
  setcookie ("loginfo","", time() - 3600);
 }
}
?>
```

Questi file sono molto importanti perché forniscono la base per le operazioni effettuate dagli script PHP che necessitano la connessione al database e le informazioni di login.
Si è cercato infatti di ridurre la complessità delle pagine attraverso uno sviluppo modulare, oltre agli *include* all'inizio dello script appena visti, ce ne sono diversi altri.

```
    <div id="container">
      <!-- header con logo -->
      <div id="header">
          <?php include('toplogin.php'); ?>
      </div>
      <?php include('navigation.php'); ?>

include('footer.php');
```

Il file toplogin.php serve per mostrare il form di login nel caso in cui l'utente non abbia ancora effetuato l'accesso, navigation.php gestisce invece il menu di navigazione.
In questi due script si può notare l'uso delle varabili definite in precedenza. $page indica su quale pagina si sta navigando, e viene usato per assegnare l'id activelink, tramite cui viene effettuata una scelta sugli elementi grafici da mostrare nell'interfaccia, attraverso la voce relativa nel menu con una diversa colorazione, le altre variabili vengono invece utilizzate per mostrare le voci di logout o registrazione e il pannello per l'inserimento dei dati al webmaster.

Nel piè di pagina é stato incluso un footer standard, e in esso viene anche chiusa la connessione con il database in caso sia ancora aperta.

File footer.php

```
<?php @mysql_close($db); ?>
</div>
<div id="footer"> |
```

```
  <a href="contatti.php">contatti</a> |
  <a href="privacy.php"
onclick="javascript:popup('privacy.php','700','350');return
false;">privacy</a> |
  <a href="condizioniuso.php"
onclick="javascript:popup('condizioniuso.php','700','350');return
false;">condizioni d'uso</a> |
</div>
</div>
</body>
</html>
```

All'inizio delle pagine è presente una direttiva <script>. Questa direttiva include il codice presente nel file pop.js e gestisce l'apertura dei popup.
Nel file index.php viene usato questo script per poter creare la lista degli ultimi 10 dischi, l'unica differenza con il file listcd.php è che in quest'ultimo vengono mostrati tutti i cd pubblicati in ordine di data e non solo gli ultimi 10.

```
<?php

          $db = connecttodb();

          //leggi i cd ordina per data e scrivi con link e
variabili da passare con metodo get
          $query = "SELECT
nome,titolo,codcd,datapubblicazione,duratacomplessiva
                    FROM autore, cd
                    WHERE autore.codautore = cd.codautore
                    ORDER BY datapubblicazione"; //i cd sono
ordinati per data
                                                  //di pubblicazione
          $ris = mysql_query($query);

          for($i=0;$i<10;$i++) //mostra solo gli ultimi 10 cd
          {
            if($row = mysql_fetch_array($ris))
            {
            $codcd = $row['codcd'];
            $autore = $row['nome'];
            $titolo = $row['titolo'];
            $durata = $row['duratacomplessiva'];
            $data = date("d-m-y, g:i ",$row['datapubblicazione']);

            //trasforma la durata registrata in secondi in
minuti/secondi
            $m = (int)($durata/60);
            $s = $durata%60;

            $popup = "doinfo/cdinfo.php?codcd=$codcd";
            echo "<li>
```

```
            <a href=\"$popup\"
onclick=\"javascript:popup('$popup','700','350');return false;\">
            $autore - $titolo, durata totale: $m.$s, pubblicato il
$data</a>
                  </li>";
            }
          }
        echo "</ul>";

include('footer.php');
?>
```

Altri file importanti sono quelli attraverso cui l'utente può effettuare ricerche. Nella ricerca di cd e tracce è stata creata una funzione di ricerca con all'interno una struttura switch per poter definire la query di ricerca appropriata in base alla scelta dell'utente.

```
function ricerca($valore,$indice)
{
  $risultato = false;
  switch($indice)
  {
    case $indice=="autore": //ricerca per autore
    $ricerca = "SELECT tracce.codcd as cod,
                nome, tracce.titolo as tit, durata, compositore
                FROM cd,tracce,autore
                WHERE autore.codautore=cd.codautore AND
                tracce.codcd=cd.codcd AND nome LIKE '%$valore%';";
    break;
    case $indice=="titolo": //ricerca per titolo
    $ricerca = "SELECT tracce.codcd as cod, nome,
                tracce.titolo as tit, durata, compositore
                FROM cd,tracce,autore
                WHERE autore.codautore=cd.codautore AND
                tracce.codcd=cd.codcd
                AND tracce.titolo LIKE '%$valore%';";
    break;
    case $indice=="genere": //ricerca per genere
    $ricerca = "SELECT tracce.codcd as cod, nome, codsottog,
                tracce.titolo as tit, durata, compositore
                FROM cd,tracce,autore
                WHERE autore.codautore=cd.codautore AND
                tracce.codcd=cd.codcd
                AND codsottog  LIKE '%$valore%';";
    break;
  }
  $risric = mysql_query($ricerca);
  echo mysql_error();
  while($row = mysql_fetch_array($risric,MYSQL_ASSOC))
  {
       $codcd = $row['cod'];
       $titolo = $row['tit'];
```

```
        $nome = $row['nome'];
        $m = (int)($row['durata']/60);
        $s = $row['durata']%60;
        $compositore = $row['compositore'];
        $risultato = true;
        $popup = "doinfo/cdinfo.php?codcd=$codcd";
        echo "<li>
             <a href=\"$popup\"
onclick=\"javascript:popup('$popup','700','350');return false;\">
$nome : $titolo, $m.$s, $compositore</a>
             </li>";
  }
  return $risultato;
}
```

Questa funzione è richiamata passando come parametri i dati presenti nella variabile di sistema $_POST.

```
if(isset($_POST['indice'])&&($_POST['indice']=="autore"
  ||$_POST['indice']=="titolo")&&isset($_POST['valore']))
  {
  echo "<h3>Risultati della ricerca:</h3>\n<ul>\n";
  if(!ricerca(@$_POST['valore'],$_POST['indice'])) echo
"</ul>Nessun risultato";
  else echo "</ul>";
  }
//richiama se invece è il genere
if(isset($_POST['indice'])&&$_POST['indice']=="genere")
  {
  echo "<h3>Risultati della ricerca:</h3>\n<ul>\n";
  if(!ricerca(@$_POST['genere'],$_POST['indice'])) echo
"</ul>Nessun risultato";
  else echo "</ul>";
  }
```

Per quanto riguarda la ricerca della sede è stata sempre utilizzata una funzione per semplificare il codice, e si è data la possibilità all'utente, che abbia effettuato l'accesso, di ottenere immediatamente informazioni sui negozi presenti nella propria zona.

Gli inserimenti sono abbastanza semplici, e avvengono attraverso delle query di inserimento, potrebbe essere interessante notare la possibilità di caricare una immagine direttamente tramite upload con il seguente codice

---

```
//salva la immagine
$path = "copertine/";
$copertina = $path."generic.jpg";
if($HTTP_POST_FILES['copertina']['tmp_name']!="") //controlla se
```

```
esiste il file temporaneo
{
 if (is_uploaded_file($HTTP_POST_FILES['copertina']['tmp_name']))
 {
  //controlla che il tipo di file sia una gif/jpeg
  if (($HTTP_POST_FILES['copertina']['type']=="image/gif") ||
($HTTP_POST_FILES['copertina']['type']=="image/pjpeg") ||
($HTTP_POST_FILES['copertina']['type']=="image/jpeg"))
  {
    $res = copy($HTTP_POST_FILES['copertina']['tmp_name'],
$path.$HTTP_POST_FILES['copertina']['name']);
    if (!$res){ echo "Upload immagine fallito!<br />\n";
include('footer.php'); exit(); }
     else { echo "Upload immagine riuscito.<br />\n"; $copertina =
$HTTP_POST_FILES['copertina']['name'];}
  }
  else{
    echo "Tipo di immagine errato sono ammessi solo gif/jpeg.";
    include('footer.php');
    exit();
  }
 }
  else{
    echo "Errore nell'upload dell'immagine";
    include('footer.php');
    exit();
 }
}
```

IL FATTORE DELLA SICUREZZA NELLA PROGETTAZIONE

PREMESSA A QUESTO PARAGRAFO

Nelle attuali applicazioni di e-commerce il fattore della sicurezza è particolarmente importante, e lo sviluppatore deve saper progettare necessariamente un sistema sicuro e al contempo usabile.
Nello sviluppo di questo progetto, sono state considerate due tra le principali problematiche riguardanti la sicurezza, un attacco di tipo SQL Injection e la registrazione e il mantenimento di dati sensibili quali le password.

SQL INJECTION

Un attacco di tipo SQL Injection è una tipologia di attacco a una applicazione web basata su database SQL, che sfrutta l'inefficienza dei controlli sui dati ricevuti in input per inserire codice non previsto nelle query. Questo permette al malintenzionato di visualizzare/modificare dati sensibili o accedere ad aree protette del sito senza possedere le credenziali di accesso.

Un tipico script PHP di autenticazione può assomigliare a questo:

```php
<?php
$username = $_POST['username'];
$password = $_POST['password'];
//prepara la query
$query = "SELECT *
          FROM utenti
          WHERE username='$username' AND password='$password'";
$risultato = mysql_query($query); //esegui la query
if($row = mysql_fetch_array($risultato))
{
 //l'utente è autenticato
}
else
{
 //dati immessi errati l'utente non è autenticato
}
?>
```

Finché i dati sono quelli normalmente previsti per l'autenticazione non ci sono problemi ma un malintenzionato può “forgiare” un nome utente e / o una password ad hoc per poter accedere senza credenziali di accesso.
Immaginiamo adesso di essere il malintenzionato, e di sapere qual'è il nome utente dell'amministratore, per esempio “administrator” e inseriamo come nome utente “administrator”, ovviamente non conosciamo la password, inseriamo quindi al posto della password questa stringa (senza apici) “**password' OR '1**”
In questo modo la stringa della query SQL al momento della esecuzione diventerà

SELECT * FROM utenti where username='administrator' AND password='**password' OR '1**'

La query rimane perfettamente valida, ed essendo '1' un valore diverso da 0 risulta sempre come vero e riusciamo ad ottenere accesso anche non avendo la password.

PROTEZIONE DA UN ATTACCO DI TIPO SQL INJECTION

Dato che questo è un progetto dimostrativo per non renderlo troppo complesso si è scelto di proteggere solo alcuni degli input, cioè quelli più importanti il login e i cookie.

La protezione consiste in un controllo degli input tramite valutazione con espressioni regolari (che sono una sintassi attraverso cui è possibile rappresentare un insieme di stringhe).

Vengono quindi controllati l'input al momento della registrazione, e del login e l'input dai cookie per evitare che vengano inserite stringhe non previsto o che vengano modificati i cookie per ottenere l’accesso.

```
if(isset($username)&&isset($password)&&@ereg("^[a-fA-F0-
9]+$",$password)
&&@ereg("^[a-zA-Z0-9]+$",$username))
```

la funzione ereg( ) viene usata quindi per controllare che gli input siano di tipo alfanumerico (e nel caso della password che siano delle cifre esadecimali).

Nel caso degli input della registrazione vengono effettuati ulteriori controlli come la validità del codice di avviamento postale.

```
//controlla tramite espressioni regolari i dati in input
if( ($username=="") || ($password=="") || ($email=="") ||
($nome=="")
  || ($cognome=="") || ($cap=="") || ($provincia=="") ||
($citta=="")||
  (!(@ereg("^[a-zA-Z0-9 ]+$",$password)&&@ereg("^[a-zA-Z0-
```

```
9]+$",$username)&&
  @ereg("^[a-zA-Z0-9@.]+$",$email)&&@ereg("^[a-zA-Z0-9]+$",$nome)&&
  @ereg("^[a-zA-Z0-9]+$",$cognome)&&@ereg("^[a-zA-Z0-
9]+$",$giorno)&&
  @ereg("^[a-zA-Z0-9]+$",$mese)&&@ereg("^[a-zA-Z0-9]+$",$anno)&&
  @ereg("^[0-9]+$",$telefono)&&@ereg("^[a-zA-Z0-9]+$",$indirizzo)&&
  @ereg("^[a-zA-Z0-9]+$",$citta)&&@ereg("^[a-zA-Z0-
9]+$",$provincia)&&
  @ereg("^[a-zA-Z0-9]+$",$password)&&@ereg("^[a-zA-Z0-
9]+$",$username)&&
  @preg_match('/^\d{5}(-\d{4})?$/',$cap)
  )))
```

L'USO DELLA CRITTOGRAFIA

Un ulteriore accorgimento per quanto riguarda la sicurezza delle applicazioni web è l'utilizzo di algoritmi crittografici per proteggere i dati sensibili.

In questo caso è stata scelta la funzione md5() del php che fa uso dell'algoritmo one-way md5 per generare un valore univoco per la stringa data in input (la password) (vedere maggiori informazioni nella tesina).
In questo caso il salvataggio del valore sarà più sicuro perché l'unico modo che si ha per ritornare alla stringa originale è effettuare un'attacco di tipo “forza bruta” (brute-force) in cui si provano tutte le possibili combinazioni delle stringhe.
Un'attacco di questo tipo viene vanificato da stringhe abbastanza lunghe e complesse, perché computazionalmente il tempo richiesto diventa troppo lungo.

CONCLUSIONE

Il lavoro svolto ha messo in risalto, in maniera diretta o indiretta, non solo problemi di natura tecnica strettamente connessi alle caratteristiche strutturali che un sito di e-commerce generalmente deve avere per poter rispondere rispondere alle esigenze dell'azienda ed ai bisogni dell'utenza, ma anche di natura socio-culturale, legati allo sviluppo dell'e-commerce che in Italia trova ancora ostacoli non indifferenti ad un evoluzione ed a uno sviluppo più a larga scala.

Nella costruzione del sito, tecnicamente, si è dato importanza al modo in cui l'utente possa avere la possibilità di accedere a tutte le informazioni necessarie nella maniera più semplice possibile ed in tutta sicurezza, ovviamente, il software web sviluppato andrebbe testato in maniera approfondita e non solo in simulazione come è stato effettuato, anche se avendolo validato e reso aderente agli standard w3c, sicuramente, sarà in ogni caso più facile per l'utente poterlo visualizzare correttamente, offrendo una navigazione piacevole e non complessa.

D'altra parte, implementare una struttura sicura, però, non è la chiave del successo, se questo non è supportato da regole di gestione e di utilizzo chiare e trasparenti. Vi sono infatti ancora oggi problemi sociali, politici e culturali, che non favoriscono un adeguato sviluppo del commercio elettronico, e di cui devono farsi carico aziende e Stato tramite una opera di formazione e di informazione perché venga implementato un utilizzo consapevole delle nuove tecnologie che aiuti soprattutto a vincere la diffidenza che ancora si prova verso questi strumenti.

Tutto questo nella realizzazione del progetto ha messo alla prova le mie

competenze acquisite sia scolasticamente, che con esperienze extra-scolastiche, permettendomi di incrementare le mie conoscenze e capacità nella gestione di tali problematiche, conoscenze e capacità che tra l'altro sono sicuramente spendibili nel mondo del lavoro.

In ogni caso questo non vuole essere un progetto completo e definitivo di un sito di e-commerce, ma una possibile e buona base di partenza per l'implementazione di un simile progetto in campo reale. Infatti in una situazione simile andrebbero prese in considerazione altre problematiche non contemplate in questa sede e che esulano dalla presente trattazione, quali la gestione di un server dedicato, l'implementazione di una connessione sicura tramite SSL e una gestione più restrittiva e sicura degli input, oltre all'organizzazione e alla gestione dei cosiddetti "carrelli", e il rinnovamento delle strutture aziendali, per poter gestire con flessibilità il commercio tramite gli strumenti informatici.

Il Candidato

Davide Quarta

Si ringrazia vivamente la prof.ssa Filomena Smacchia per la sua disponibilità al confronto e per il suo apporto professionale, fattori che mi hanno consentito di approfondire i temi e gli argomenti inerenti al progetto e portare a termine questo lavoro.