

PROGETTO
BAR

Lavoro di informatica realizzato da:
**Murolo Francesco
Tedesco Alessandro**



Classe 5A - INFORMATICA
A.S. 2017/2018

SOMMARIO

1-	Contesto in esame e analisi del problema	pag. 3
2-	Analisi e organizzazione della base di dati	pag. 4
3-	Struttura funzionale del sito	pag. 7
4-	Sicurezza	pag. 8
5-	Software utilizzati	pag. 9
6-	Layout e descrizione descrizione dei punti critici	pag. 10

CONTESTO IN ESAME E ANALISI DEL PROBLEMA

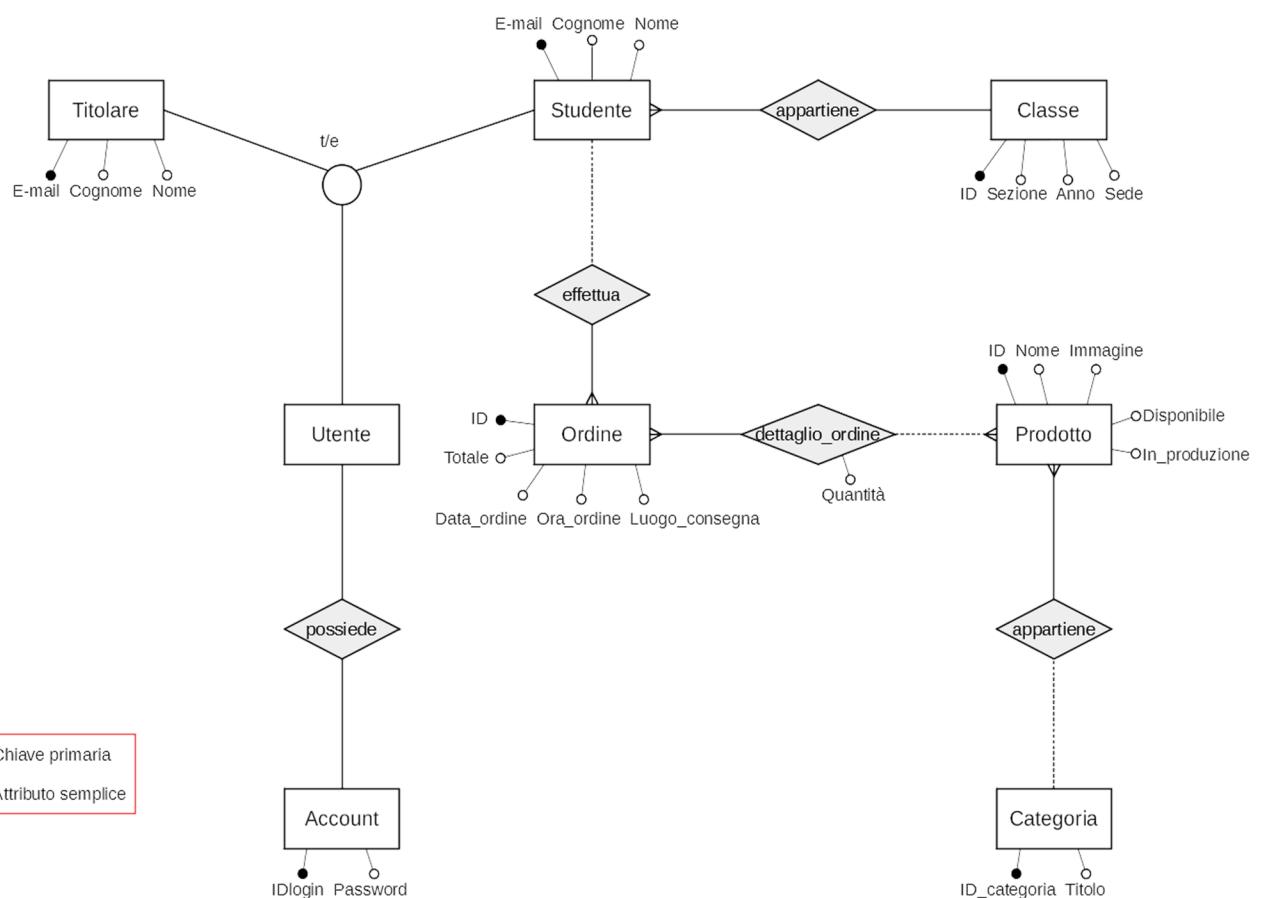
Per finalizzare le competenze informatiche acquisite ed applicare i linguaggi di programmazione trattati in aula, si è pensato di approfittare dell'esigenza di migliorare una realtà presente nella scuola: gli acquisti effettuati al bar scolastico. L'organizzazione di essi è stata sempre gestita tramite l'invio di messaggi Whatsapp da un rappresentante di classe al titolare dell'attività, implicando una serie di problematiche, tra le quali:

- l'impossibilità di visualizzare i prodotti su un'interfaccia web;
- l'impossibilità di ottenere un subtotale dell'ordine;
- l'invio ridondante di messaggi in caso di modifiche dell'ordine;
- conseguente perdita di tempo per il titolare;
- la frequente mancanza di comunicazione del luogo di destinazione dell'ordine.

Conoscendo tutto ciò e facendo tesoro degli accorgimenti e dei consigli appresi in aula, si è deciso di sviluppare un sito web di ultima generazione, analizzando ogni problematica, creando appositi database e layout grafici. Gestendo inoltre i diversi account degli studenti, la sicurezza e l'accessibilità.

ANALISI E ORGANIZZAZIONE DELLA BASE DI DATI

Dopo una scrupolosa analisi di tale realtà, si è sviluppato il seguente modello entità/relazione:

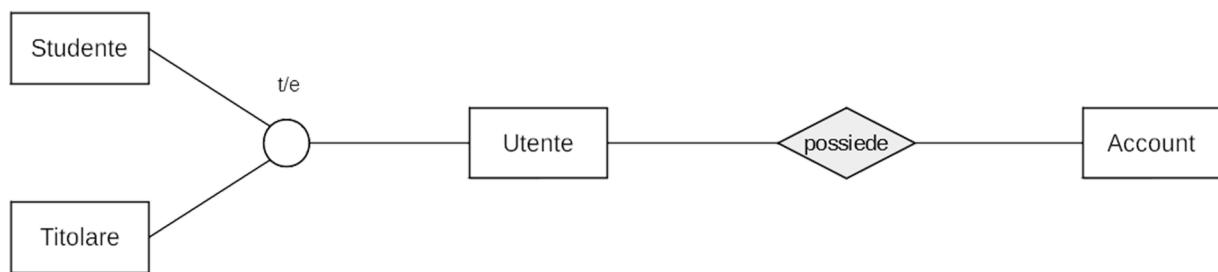


Osservando il modello E/R si può capire che una delle entità principali è UTENTE. Si individuano, inoltre, 2 ulteriori entità:

- TITOLARI
- STUDENTI

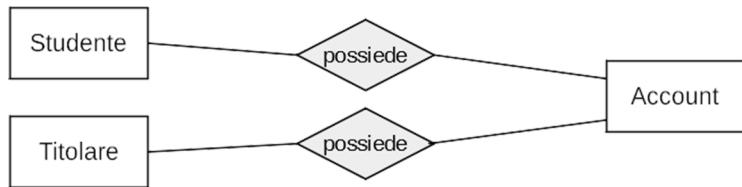
Si deduce che tra UTENTE e queste due entità vi è una dipendenza di tipo is-a, in quanto TITOLARI e STUDENTI sono degli UTENTI.

Ne deriva la seguente situazione a livello concettuale:



La dipendenza o gerarchia is-a è di tipo totale/esclusiva. Vincolo di copertura = totale in quanto non ci sono altri tipi di utenti legati all'applicazione; Vincolo di sovrapposizione = esclusiva in quanto non ci sono intersezioni tra gli insiemi-entità: uno studente non può essere un titolare ecc.

Quindi la relazione “possiede” tra UTENTE e ACCOUNT è 1:1 che sappiamo si può risolvere a livello logico creando un'unica tabella. Non conviene, però, procedere in questo modo, ma è più funzionale tenere separati le anagrafiche degli utenti dai dati sensibili quali le password, inoltre alla tabella ACCOUNT si possono assegnare i privilegi anche solo di lettura al titolare. Contestualmente va risolta la gerarchia IS-A. Per evitare di avere un'ulteriore tabella utenti separata da studenti e titolari, si adotta la soluzione di collegare direttamente queste due tabelle alla tabella account ricavando le tabelle TITOLARI, STUDENTI, ACCOUNT. La chiave primaria di TITOLARI e STUDENTI è e-mail.



Per implementare la relazione a livello logico, essendo del tipo 1:1, si può scegliere se creare la chiave esterna in ACCOUNT trasferendo l'e-mail in ACCOUNT che a questo punto diventerebbe la utenteID; ciò non è possibile poiché quando si va ad impostare il vincolo di INTEGRITÀ REFERENZIALE tra ACCOUNT e STUDENTI non sarà possibile la clausola ON DELETE SET NULL visto che l'e-mail in STUDENTI e in TITOLARI è chiave primaria. L'alternativa ON DELETE CASCADE non è percorribile visto che l'eliminazione di un account eliminerebbe anche lo studente o il titolare che lo possedeva. Rimane l'unica soluzione possibile che è quella di creare una chiave primaria IDlogin in ACCOUNT che diventa CHIAVE ESTERNA in STUDENTI e TITOLARI.

Il modello logico dell'intera applicazione sarà il seguente:

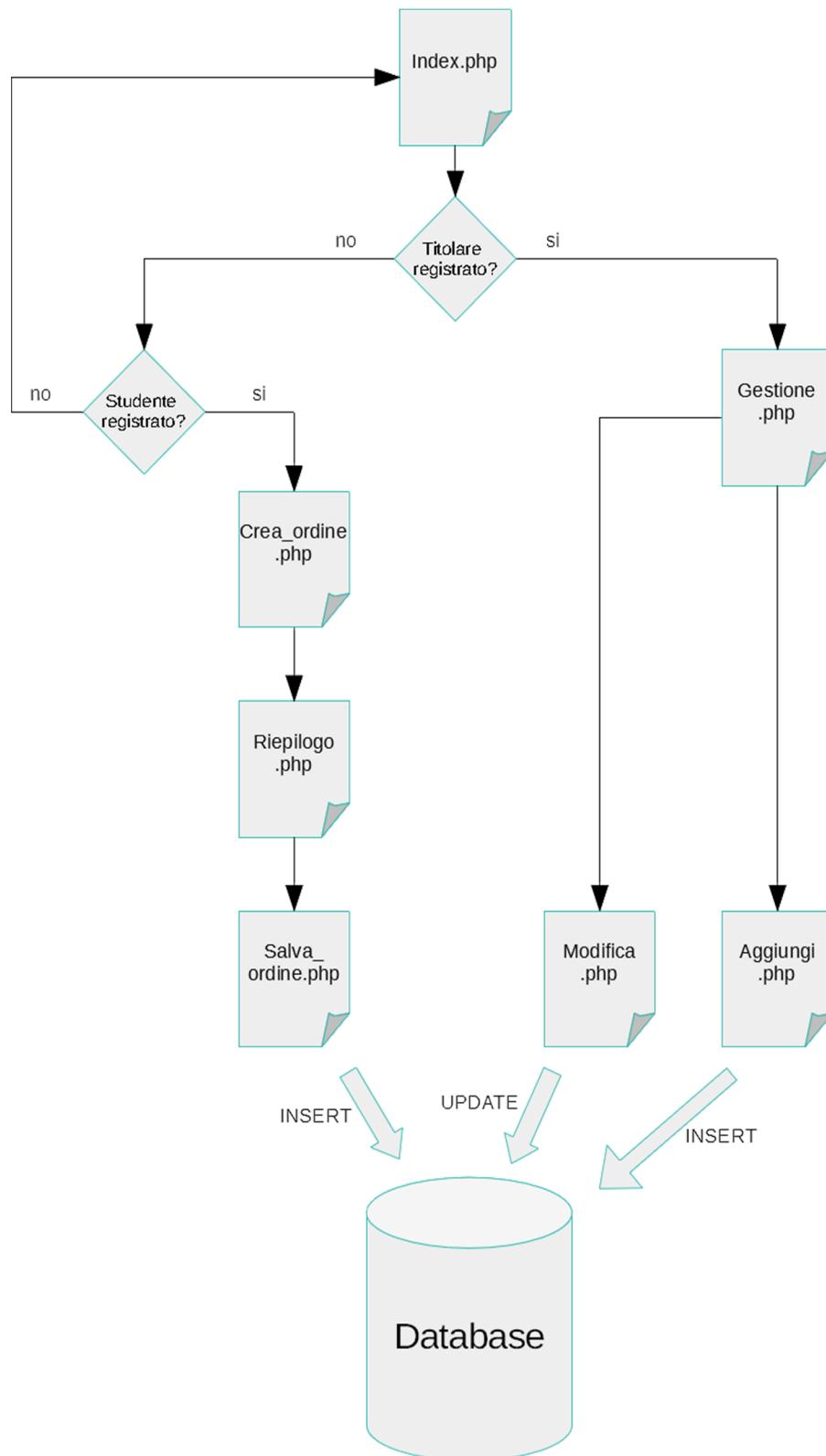
```

Studenti(email, nome, cognome, ID_classe, IDlogin);
Titolari(email, nome, cognome, IDlogin);
Account(IDlogin, password);
Classi(ID, anno, sezione, sede);
Ordini(ID, luogo_consegna, data_ora_ordine, totale, ID_studente);
Prodotti(ID, nome, immagine, prezzo, disponibile, in_produzione, ID_categoria);
Dettaglio_ordini(ID_ordine, ID_prodotto, quantità);
Categorie(ID_categoria, titolo);
  
```

<u>Chiave primaria</u>
<u>Chiave esterna</u>

Gli attributi *Data_ordine* e *Ora_ordine* dell'entità Ordine, definita nel modello e/r, diventeranno un'unico campo *data_ora_ordine* di tipo DATETIME.

STRUTTURA FUNZIONALE DEL SITO



SICUREZZA

Un altro aspetto molto importante di un'applicazione Web è la sicurezza. Infatti, sono sempre di più gli attacchi che un'applicazione Web subisce se progettata e sviluppata in malo modo. In particolare, si è voluto rafforzare e rendere non vulnerabile l'applicazione BAR dalla tecnica SQL Injection, ovvero tutti gli attacchi ad un'applicazione Web – nel nostro caso PHP – che consentano di modificare l'interazione della stessa con il database: nel caso vengano eseguite, lato server, interrogazioni MySQL costruite su quanto ricevuto dal client, senza un controllo sull'input è possibile che un cracker crei, “teoricamente” parlando, disastri irrimediabili. Per contrastare questa tecnica si è utilizzata in **autentica.php** CON FIRMA sulla password. Se il sistema di autenticazione, infatti, utilizza un ben più sicuro sistema di firma, per il quale la password non è memorizzata in chiaro su database ed in sua vece è memorizzato il suo hash, il programma sarà non vulnerabile ad un attacco di tipo SQL Injection. Se si confronta l'hash di quanto inserito dall'utente con ciò che è su db (già “hashato”), il comando md5 vanifica, trasparentemente, gli sforzi del cracker.

```
$username=mysql_real_escape_string($_POST['email']);  
$password=md5(mysql_real_escape_string($_POST['password']));
```

Per proteggere l'area riservata sono state utilizzate le sessioni PHP.

```
header("Cache-Control: no-store, no-cache, must-revalidate");  
  
session_start();  
$autenticato=$_SESSION['autenticato'];  
  
if (!$autenticato) {  
    header("location:index.php?err=5");  
    exit();  
}
```

Nell'**index.php** viene inizializzata la sessione usando la funzione `session_start()` e impostato un flag “autenticato” a 0. Una volta effettuato l'accesso, il flag “autenticato” verrà impostato ad 1. Nelle pagine del sito basterà controllare il valore del flag e se non sarà impostato 1 verrà effettuato un redirect all'**index.php**. Bisognerà, anche, evitare di memorizzare le pagine nella cache del computer client costringendo il browser a richiederla al server web e quindi a rieseguirla.

SOFTWARE UTILIZZATI

Bootstrap Studio

Bootstrap Studio, che permette di utilizzare il framework open-source *Bootstrap*, è uno strumento che è stato utilizzato per lo sviluppo dell'interfaccia grafica del sito web.

XAMPP

XAMPP è un software che include un web server Apache e un db server MySQL.

HeidiSQL

HeidiSQL è stato utilizzato per gestire il db server MySQL.

PHP

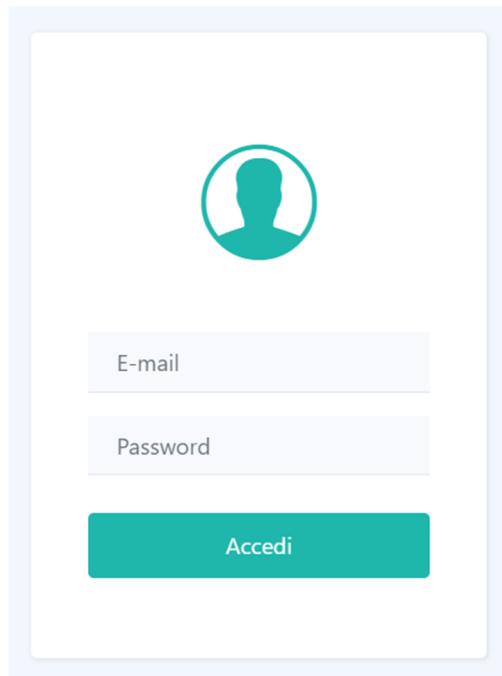
PHP è stato utilizzato nel *back-end* per dinamizzare le pagine web.

JAVASCRIPT

JAVASCRIPT è stato utilizzato nel *front-end* per creare effetti dinamici interattivi.

LAYOUT E DESCRIZIONE DELLE PAGINE

Index.php



The form consists of a teal user icon at the top. Below it are two input fields: one for 'E-mail' and one for 'Password', both with placeholder text. At the bottom is a teal button labeled 'Accedi'.

Prevede un form con due caselle di testo di cui una di tipo *email* e una di tipo *password* (che maschera i caratteri durante la digitazione). Il pulsante “Accedi” è di tipo *submit* e, quando premuto, invia i dati *e-mail* e *password* alla pagina **autentica.php** (specificata nel parametro “action” del form).

Autentica.php

È una pagina di controllo quindi non prevede un’interfaccia utente. Pertanto è PHP puro ed è trasparente al navigatore.

La sua funzione è quella di:

- Recuperare i dati di accesso digitati nella pagina **index.php**;
- Interrogare il database per controllare l’esistenza o meno di un utente con tali credenziali;
- Se è registrato ed è un titolare ad aver effettuato l’accesso, effettua una redirect alla pagina **gestione.php**;
- Se è registrato ed è uno studente ad aver effettuato l’accesso, effettua una redirect alla pagina **crea_ordine.php**;
- Se non è registrato, effettua una redirect alla pagina **index.php**, definendo il codice d’errore nella *querystring*.

In questa pagina sono state utilizzate le funzioni sui cookies per “ricordare” l’username per i successivi accessi.

Crea_ordine.php

PANINERIA

Icon	Product	Price	Quantity	Action
	Panino prosciutto	€1.80	0	
	Panino cotoletta	€1.80	0	
	Panino salame piccante	€1.80	0	

ROSTICCERIA

Contiene il codice lato server che costruisce la pagina di consultazione dei prodotti presenti nella tabella PRODOTTI del database.

La colonna a destra, che è un'anteprima del carrello, verrà creata con il seguente codice:

```
<div class="col-md-2">
<div style="position:fixed;text-align:left;margin-top:41px;">
<?php
    //verifico se il carrello di sessione è impostato
    if(isset($_SESSION['carrello']) && count($_SESSION['carrello'])!=0){
        //seleziono solo i prodotti che sono nella sessione
        $sql="SELECT * FROM prodotti WHERE ID IN (
            ";
        foreach($_SESSION['carrello'] as $id => $value) {
            $sql.=$id.",";
        }
        $sql=substr($sql, 0, -1).") ORDER BY nome ASC";
        $query=mysql_query($sql);
        while($row=mysql_fetch_array($query)){
            ?>
            <p><font size="2"><?php echo $row['nome'] ?> x <?php echo $_SESSION['carrello'][$row['ID']]['quantity'] ?></font></p>
            <?php
        }
        ?>
        <a class="btn btn-primary" role="submit" href="riepilogo.php" style="background-color:rgb(255,15,0);"><strong>Visualizza carrello</strong></a>
        <?php
    }
    ?>
</div>
</div>
```

Viene eseguita una SELECT mysql, selezionando solo i prodotti che sono nella sessione. A tale scopo, si utilizza la funzione *foreach*. Quindi si esegue un loop sulla sessione aggiungendo l'ID del prodotto alla SELECT. Poi, si userà la funzione *substr* per rimuovere l'ultima virgola dalla SELECT.

Caso 1: Senza la funzione *substr*

```
SELECT * FROM prodotti WHERE ID IN(2,3,4,) ORDER BY nome ASC;
```

L'ultima virgola genererà un errore

Caso 2: Con la funzione *substr*

```
SELECT * FROM prodotti WHERE ID IN(2,3,4) ORDER BY nome ASC;
```

Come già detto, questa parte di codice rappresenta una sorta di anteprima del contenuto del carrello. Tutte le informazioni del carrello verranno visualizzate nella pagina **riepilogo.php**, usando lo stesso codice precedente.

Carrello					Esci
	Nome	Quantità	Prezzo unitario	Subtotale	
	Calzone	- 3 +	€1.20	€3.60	
	Panino cotoletta	- 1 +	€1.80	€1.80	
	Panino prosciutto	- 1 +	€1.80	€1.80	
Aggiorna carrello					
Totale:		€7.20	Indietro	Termina ordine	

Aggiungi_al_carrello.php

```
$id=$_GET['id'];
$quantita=$_POST[$id];

if(isset($_SESSION['carrello'][$id])){
    $_SESSION['carrello'][$id]['quantity']+=$quantita;
    header('Location:crea_ordine.php');

}else{
    include "connessione_DB.php";

    $sql="SELECT * FROM prodotti WHERE ID=$id";
    $sql_ris=mysql_query($sql);

    if(mysql_num_rows($sql_ris)!=0){
        $row_s=mysql_fetch_array($sql_ris);

        $_SESSION['carrello'][$row_s['ID']] = array("quantity" => $quantita, "price" => $row_s['prezzo']);
        header('Location:crea_ordine.php');

    }else{

        header('Location:crea_ordine.php?err=1');

    }
}
```

È una pagina che contiene solo codice php e ha la funzione di aggiungere al carrello i prodotti scelti dall'utente.

1. Viene recuperato il valore dell'ID del prodotto dall'url e viene salvato in una variabile “\$id” e salvo la quantità, passata tramite form, in una variabile “\$quantita”;
2. Se l'ID del prodotto è nella sessione del carrello, viene modificata la quantità;
3. Se l'ID non è presente nella sessione, bisogna controllare che l'ID passato attraverso la variabile GET esista nel database. In caso affermativo, viene recuperato il prezzo dal db e si crea la sessione. Se così non fosse, viene restituito un errore alla pagina **crea_ordine.php**.

Salva_ordine.php

```
$luogo=$_POST['luogo_consegna'];
$date=date("Y-m-d H:i:s");
$totale=(float)substr($_POST['totale_ordine'], 3);
$messaggio=null;
$mail_mittente = $_SESSION['id'];

$studente = mysql_fetch_array(mysql_query("SELECT * FROM studenti,classi WHERE studenti.ID_classe=classi.ID AND studenti.email='".$mail_mittente' "));
$nome_mittente = ucfirst($studente['nome']);
$mail Oggetto = "Ordine ".$studente['anno'].strtoupper($studente['sezione']);

$titolare = mysql_fetch_array(mysql_query("SELECT * FROM titolari "));
$mail_destinatario = $titolare['email'];

$mail_headers = "From: " . $nome_mittente . " <" . $mail_mittente . ">\r\n";
$mail_headers .= "Reply-To: " . $mail_mittente . "\r\n";
$mail_headers .= "Content-type: text/html; charset=UTF-8\r\n";
$mail_headers .= "X-Mailer: PHP/" . phpversion();

$sql="SELECT * FROM prodotti WHERE prodotti.ID IN (
    foreach($_SESSION['carrello'] as $id => $value) {
        $sql.=$id ",";
    }

    $sql=substr($sql, 0, -1)."") ORDER BY nome ASC";
$query=mysql_query($sql);
while($prodotto=mysql_fetch_array($query)){
    $quantita=$_SESSION['carrello'][$prodotto['ID']]['quantity'];
    $messaggio.=$prodotto['nome']." x ".$quantita."<br/>";
}

$messaggio.="
<hr/><strong>Luogo consegna: </strong>".$luogo."<br/><strong>Totale: </strong>€".number_format($totale,2);

if(mail($mail_destinatario, $mail Oggetto, $messaggio, $mail_headers)){
    $query_o="INSERT INTO bar.ordini (luogo_consegna, data_ora_ordine, totale, ID_studente) VALUES ('$luogo', '$date', '$totale', '$mail_mittente') ";

    $ris = mysql_query($query_o) or die("Errore query 1!");

    $query_s="SELECT ordini.ID FROM ordini WHERE ordini.data_ora_ordine='$date' AND ordini.ID_studente='".$mail_mittente' ";
    $ris_q= mysql_query($query_s) or die("Errore query 2!");
    $id_ordine=mysql_fetch_array($ris_q);
    $ordine=$id_ordine['ID'];

    $sql="SELECT * FROM prodotti WHERE prodotti.ID IN (
        foreach($_SESSION['carrello'] as $id => $value) {
            $sql.=$id ",";
        }

        $sql=substr($sql, 0, -1)."") ORDER BY nome ASC";
    $query=mysql_query($sql);
    while($prodotto=mysql_fetch_array($query)){
        $quantita=$_SESSION['carrello'][$prodotto['ID']]['quantity'];
        $id_p=$prodotto['ID'];
        $sql2="INSERT INTO bar.dettaglio_ordini (ID_ordine, ID_prodotto, quantita) VALUES ('$ordine', '$id_p', '$quantita')";
        mysql_query($sql2) or die("Errore query 3!");
    }

    mysql_close($conn);

    header('Location:avviso_ordine.php');
} else
    header('Location:riepilogo.php?err=1');
```

È una pagina che contiene solo codice php e ha la funzione di inviare l'ordine al titolare, utilizzando la funzione *mail()*, e di salvarlo nel database.

La funzione *mail()*, una volta richiamata, “contatterà” il sistema postale del nostro server (sendmail o server SMTP) intimandogli di spedire una mail con le caratteristiche definite dallo sviluppatore. Ovviamente, nel caso in cui il nostro server non sia attrezzato di un sistema di spedizione attivo e funzionante la funzione *mail()* restituirà FALSE (restituirà TRUE in caso di successo).

