Relazione di Sistemi Informativi Aziendali

Materiali Utilizzati

- **PHP 7** per la gestione dati.
- **HTML / CSS** per la visualizzazione dei contenuti.
- **MariaDB** per il database.
- **Apache** per il webserver.
- **Internet** per la documentazione.
    - **Siti utilizzati:**
        - *W3School*
        - *StackOverflow*
        - php.net/manual
- **Microsoft Word** per questa relazione.

Descrizione del sistema

Il sito è stato pensato come un E-Commerce specializzato nella vendita di prodotti elettronici, che variano da componenti per PC a Smartphone.
Le pagine presenti si dividono in due categorie:

- Pagine dell'amministratore.
    - Queste pagine sono accessibili solo tramite un link, e richiedono l'autenticazione tramite credenziali registrate manualmente sul database.
    - Riguardano principalmente la gestione dell'inventario: inserimento, modifica ed eliminazione dei prodotti dal database.

- Pagine dell'utente.
    - Pagina principale, in cui sono mostrati i prodotti in evidenza, selezionati dall'amministratore.
    - Pagina dell'utente riservata. Da qua è possibile visualizzare gli ordini, gestire gli indirizzi di fatturazione e spedizione, modificare la propria e-mail e password ed infine eliminare l'account.
    - Catalogo, nella quale sono visualizzati tutti i prodotti presenti nel database, è possibile anche ordinarli per categoria ed accedere alla pagina dove visualizzare i dettagli dell'articolo.
    - Carrello, da qua è possibile rimuovere articoli aggiunti precedentemente oppure proseguire all'acquisto.

Descrizione tabelle e campi

Nella tabella AdminData vengono immagazzinati i dati relativi al login degli amministratori del sito.

- AdminData
  - ID primary key, index, AI - INT
    - *ID univoco ed auto incrementante per identificare ogni singolo amministratore.*
  - AdminUsername - TEXT
    - *Username utilizzato per il login dell'amministratore.*
  - AdminPassword - TEXT
    - *Password utilizzata per il login dell'amministratore.*

Tabella nella quale sono presenti tutti i dati degli utenti.

- RegisterData
  - ID primary key, index, AI - INT
    - *ID auto incrementante che associa univocamente ogni utente.*
  - Username - TEXT
    - *Username dell'utente. Unico.*
  - Name - TEXT
  - Surname - TEXT
  - Birth_Date - DATE
  - Email - TEXT
    - *E-Mail utilizzata al momento della registrazione. Può essere cambiata successivamente.*
  - Password - TEXT
    - *Password hashata dell'utente. Può essere cambiata successivamente.*
  - Gender - tinyINT
    - *Sesso selezionato al momento della registrazione:*
      0: Femminile.
      1: Maschile.
  - Reg_Date - DATETIME
    - *Data di registrazione dell'utente al sito.*
  - SALT - TEXT
    - *Sale utilizzato per l'hash della password.*

In questa tabella sono presenti tutti i dati relativi ai prodotti.

- Articles
  - ID primary key, index, AI - INT
    - *ID univoco ed auto incrementante per identificare un articolo.*
  - PN - TEXT
    - *Nome del prodotto visualizzato sul sito.*
  - Manufacturer - TEXT
    - *Nome del produttore.*
  - Seller - TEXT
    - *Nome del rivenditore.*
  - Availability - INT
    - *Questo campo associa ad un prodotto la sua disponibilità, viene classificata con dei valori precedentemente accordati:*
      0: Non disponibile
      1: Disponibile
      2: In arrivo
  - N_Left - INT
    - *Identifica la quantità del prodotto rimanente in magazzino.*
  - Description - TEXT
    - *Descrizione visualizzata nella pagina del prodotto.*
  - IMG - TEXT
    - *Campo contentente il nome dell'immagine memorizzata sul sito.*
  - Price - float
    - *Prezzo relativo al prodotto.*
  - Category - TEXT
    - *Categoria del prodotto.*
  - HL - INT
    - *Indica se il prodotto verrà visualizzato nella pagina principale come prodotto in evidenza.*
      0: Non in evidenza
      1: In evidenza.

Questa tabella gestisce il carrello degli utenti. Associa gli utenti con i prodotti inseriti nel carrello e la relativà quantità.

- Basket
  - ID primary key, index, AI - INT
    - *ID univoco ed autoincrementante*
  - AID index - INT
    - *ID dell'articolo inserito al carrello.*
  - UID index - INT
    - *Id dell'utente che ha aggiunto l'articolo al carrello.*
  - Quantity - INT
    - *Quantità relativa all'articolo aggiunta al carrello.*

Le seguenti tabelle sono utilizzate per memorizzare gli indirizzi di fatturazione e spedizione di tutti gli utenti.

- Indirizzo_Fatturazione
  - IDBilling primary key, index, AI - INT
    - *ID auto incrementante che identifica univocamente un indirizzo di fatturazione.*
  - IDUtente index - INT
    - *ID dell'utente a cui è relazionato l'indirizzo di fatturazione. Quando l'account verrà eliminato, tutti i suoi indirizzi verranno eliminati in base a questo ID.*
  - Name_Surname - TEXT
    - *Intestatario della fattura.*
  - Address_1 - TEXT
  - Address_2 - TEXT
    - *Opzionale.*
  - City - TEXT
  - Province - TEXT
  - CAP - TEXT
  - Paese - TEXT
  - PartitaIVA - TEXT
    - *Partita IVA che verrà stampata nella ricevuta.*

- Indirizzo_Spedizione
  - IDShipment primary key, index, AI - INT
    - *ID auto incrementante che associa univocamente un indirizzo di spedizione.*
  - IDUtente index - INT
    - *ID dell'utente a cui è relazionato l'indirizzo di spedizione. Quando l'account verrà eliminato, tutti i suoi indirizzi verranno eliminati in base a questo ID.*
  - Name_Surname - TEXT
    - *Destinatario della spedizione.*
  - Address_1 - TEXT
  - Address_2 - TEXT
    - *Opzionale*
  - City - TEXT
  - Province - TEXT
  - CAP - TEXT
    - *Questo campo è stato pensato come TEXT per evitare troncature nel caso di CAP inizianti col numero 0.*
  - Paese - TEXT
  - Phone - TEXT
    - *Questo campo è stato pensato come TEXT per evitare troncature nel caso di numeri inizianti col numero 0 e per permettere i prefissi indicanti la nazionalità (es: +39).*

La tabella in cui verranno registrati tutti gli ordini e le relative ricevute. Anche in caso di eliminazione dell'utente, tutti i record saranno sempre presenti.

- Ricevuta
  - IDOrdine primary key, AI - INT
    - *ID auto incrementante che associa univocamente un ordine/ricevuta.*
  - IDUtente index, null - INT
    - *ID dell'utente associato alla ricevuta. Se l'account dell'utente viene eliminato, questo campo verrà settato NULL per evitare che un nuovo account registrato sia legato a precedenti ricevute.*
  - PDF - TEXT
    - *Nome della ricevuta in formato PDF memorizzata sul sito.*
  - NomeProdotto - TEXT
    - *Nomi e quantità dei prodotti acquistati.*
  - Prezzo - float
    - *Prezzo totale dell'ordine.*
  - Data_Ordine - DATETIME
    - *Data ed orario dell'acquisto, per tenere traccia e ricavare utili statistiche sull'andamento delle vendite in base al periodo.*
  - email - TEXT
    - *E-Mail dell'utente al momento dell'acquisto, utilizzata per tracciare un utente.*

Lista delle query utilizzate

Register.php

```
$query =
" SELECT * FROM RegisterData WHERE lower(Username) = ? OR lower(Email) = ? ";

$connessione->doQP($query,
array(strtolower($_POST['username']), strtolower($_POST['mail'])));
```

Estrae tutto da RegisterData per controllare se l'username o l'email inserita nei campi precedenti sono già esistenti. Se il numero di righe restituite è maggiore di 0, allora o l'username o l'email sono già stati utilizzati da un altro utente. Viene utilizzato *lower* per fare in modo che il controllo non sia case sensitive.

```
$query = "
INSERT INTO RegisterData
(Username,Name,Surname,Birth_Date,Email,Gender,Password,Reg_Date, SALT)
VALUES (? , ? , ? , ? , ? , ? , ?, ?, ?) ";

$connessione->doQP($query,
array($_POST['username'], $_POST['name'], $_POST['surname'], $_POST['date'],
$_POST['mail'], $_POST['gender'], $HashedPassword, date('y-m-d H:i:s'), $salt));
```

Inserimento dei dati inseriti dall'utente nel database.

Login.php

```
$query =
" SELECT * FROM RegisterData WHERE lower(username) = ? OR lower(email) = ? ";

$connessione->doQP($query, array($username, $username));
```

Estrazione dei dati dell'utente in base all'username o alla mail inseriti nel form di login. Se la query trova risultati, si procede con la verifica della password.

myaccount.php

```
$connessione->doQP("
SELECT Username,Name,Surname,Birth_Date,Email FROM RegisterData WHERE ID = ? ",
$_SESSION['UID']);
```

Questa query estrae i dati dell'utente dal database tramite l'ID dell'utente (settato al login) per la visualizzazione dei dati personali nella pagina riservata.

Acquista.php

```
$query = "
SELECT * FROM Articles
RIGHT JOIN (SELECT * FROM Basket WHERE UID = ?) A
ON Articles.ID = A.AID ";
$connessione->doQP($query,$_SESSION['UID']);
```

Estrae tutti gli ID degli articoli presenti nel carrello dell'utente e unisce sulla tabella degli articoli per ricavare i dati di quest'ultimi.

```
$query = "
INSERT INTO Indirizzo_Fatturazione
(IDUtente,Name_Surname,Address_1,Address_2,City,Province,CAP,Paese)
SELECT IDUtente,Name_Surname,Address_1,Address_2,City,Province,CAP,Paese
FROM Indirizzo_Spedizione WHERE IDShipment = ?";
$connessione->doQP($query,$_POST['indirizzoSpedizione']);
```

Questa query viene eseguita se l'utente decide di utilizzare l'indirizzo di spedizione come indirizzo di fatturazione; copia l'indirizzo di spedizione scelto dall'utente nella tabella con gli indirizzi di fatturazione.

```
$query = "
INSERT INTO  Ricevuta(IDUtente,PDF,NomeProdotto,Prezzo,Data_Ordine,email)
VALUES (?,?,?,?,?,?) ";
$connessione>doQP($query,
array($_SESSION['UID'],$NomePDF,$NomiProdotti,$PrezzoTotale,date("Y-m-d
H:i:s"),$_SESSION['Mail']));
```

Dopo i dovuti controlli, con questa query si inseriscono i dati degli articoli acquistati nella tabella Ricevuta.

```
$query = "SELECT * FROM Indirizzo_Fatturazione WHERE IDBilling = ?";
$connessione->doQP($query, $_POST['indirizzoFatturazione']);
```

Estrae l'indirizzo di fatturazione scelto nella conferma dell'acquisto. I dati verranno inseriti nella fattura.

```
$query = "UPDATE Articles SET N_Left = N_Left - ? WHERE ID = ?";
$connessione->doQP($query,array($arr[1],$arr[0]));
$query = "SELECT N_Left FROM Articles WHERE ID = ?";
$connessione->doQP($query,$arr[0]);
$connessione->fetch();
if($connessione->get('N_Left') == 0)
$connessione->doQP("UPDATE Articles SET Availability = 0 WHERE ID =?", $arr[0]);
```

Questa query viene eseguita al momento dell'acquisto per ogni articolo presente nel carrello dell'utente.
Sottrae la quantità presente nel magazzino con quella ordinata dall'utente.
La seconda query verifica se con l'acquisto eseguito dall'utente l'articolo viene esaurito. Se sì, la disponibilità viene impostata 0 (Non disponibile).

```
$query = "DELETE FROM Basket WHERE UID = ?";
$connessione->doQP($query,$_SESSION['UID']);
```

Infine, ad acquisto completato, vengono eliminati dal carrello tutti gli articoli acquistati.

AddFattAddr.php

```
$query = "
INSERT INTO Indirizzo_Fatturazione
(IDUtente,Name_Surname,Address_1,Address_2,City,Province,CAP,Paese,PartitaIVA)
VALUES (?,?,?,?,?,?,?,?,?)";

$parametri =
array($_SESSION['UID'],$_POST['Destinatario'],$_POST['Indirizzo1'],$_POST['Indirizzo2']
,$_POST['Citta'],$_POST['Provincia'],$_POST['CAP'],$_POST['Paese'],$_POST['PIVA']);

$connessione->doQP($query,$parametri);
```

Con questa query si inseriscono, dopo gli opportuni controlli, i dati dell'indirizzo di fatturazione, precedentemente inseriti dall'utente nell'apposito form, nel database.

AddSpedAddr.php

```
$query = "INSERT INTO Indirizzo_Spedizione
(IDUtente,Name_Surname,Address_1,Address_2,City,Province,CAP,Paese,Phone)
VALUES (?,?,?,?,?,?,?,?,?)";
$parametri =
array($_SESSION['UID'],$_POST['Destinatario'],$_POST['Indirizzo1'],$_POST['Indirizzo2']
,$_POST['Citta'],$_POST['Provincia'],$_POST['CAP'],$_POST['Paese'],$_POST['Tel']);
$connessione->doQP($query,$parametri);
```

In questa pagina, con una query quasi identica alla precedente, viene inserita nell'apposita tabella l'indirizzo di spedizione dell'utente.

Basket.php

```
$query = "      SELECT *
                FROM (          SELECT * FROM Basket
                                WHERE UID = ?) A
                LEFT JOIN Articles B
                ON A.AID = B.ID ";

$connessione->doQP($query, $_SESSION['UID']);
```

Seleziona l'ID degli articoli inseriti dall'utente nel carrello, con la join si unisce il risultato alla tabella degli articoli per ricavare i dettagli e visualizzarli.

```
$query = "DELETE FROM Basket WHERE UID = ? AND AID = ?";
$connessione->doQP($query, array($_SESSION['UID'], $riga['AID']));
```

Con un controllo eseguito in precedenza si verifica se un articolo inserito dall'utente nel carrello è esaurito o meno. Questa query rimuove gli articoli esauriti dal carrello. In seguito verrà visualizzato un messaggio di avviso all'utente.

Catalogo.php

```
if(!isset($_GET['category']))
        $connessione->doQ("SELECT * FROM Articles");
else
        $connessione->doQP("SELECT * FROM Articles WHERE Category = ?",
$_GET['category']);
```

Se l'utente non ha selezionato una categoria, vengono visualizzati tutti gli articoli presenti nel database, altrimenti vengono visualizzati tutti gli articoli della categoria selezionata.

ConfermaOrdine.php

```
$query = "      SELECT *
                FROM Indirizzo_Spedizione
                WHERE IDUtente = ? ";
$connessione1->doQP($query,$_SESSION['UID']);
```

Vengono estratti dal database tutti gli indirizzi di spedizione dell'utente, in modo da far scegliere al suddetto quale utilizzare per l'ordine.

```
$query = "      SELECT *
                FROM Indirizzo_Fatturazione
                WHERE IDUtente = ? ";
$connessione2->doQP($query,$_SESSION['UID']);
```

Nella stessa maniera della query precedente, vengono estratti dal database tutti gli indirizzi di fatturazione dell'utente per far si che possa scegliere quale utilizzare.

```
$query ="       SELECT *
                FROM Articles
                RIGHT JOIN (SELECT * FROM Basket WHERE UID = ?) A
                ON Articles.ID = A.AID ";

$connessione->doQP($query,$_SESSION['UID']);
```

In maniera simile al carrello, vengono estratti gli ID degli articoli presenti nel carrello dell'utente e vengono uniti con la tabella degli articoli. In modo da visualizzare i dettagli dei prodotti inseriti nel riepilogo dell'ordine.

DeleteArticles.php

```
if(isset($_GET['AdminSearch']))
{
        $query = "SELECT * FROM Articles WHERE lower(PN) LIKE lower('%?%')";
         $connessione->doQP($query, $_GET['AdminSearch']);
}
else
$connessione->doQ("SELECT * FROM Articles");
```

In questa pagina verranno visualizzati gli articoli per l'eliminazione da parte dell'admin. Se l'admin ha deciso di cercare un articolo per nome, verrà eseguita la prima query, altrimenti verranno visualizzati tutti gli articoli.

DeleteFattAddr.php

```
$query = "
        DELETE FROM Indirizzo_Fatturazione
        WHERE IDBilling = ?";
$connessione->doQP($query,$_POST['ID']);
```

Con questa query viene eliminato l'indirizzo di fatturazione, se l'utente ha deciso di effettuare questa operazione.

DeleteItems.php

```
foreach($_POST['todelete'] as $var)
{
        $query = "SELECT IMG FROM Articles WHERE ID = ?";
        $connessione->doQP($query, $var);
        $connessione->fetch();
        unlink($connessione->get("IMG"));
}
```

Selezione del nome dell'immagine per la rimozione dal server tramite la funzione unlink.

```
$params = array();
$query = "DELETE FROM Articles WHERE ID IN (";

foreach ($_POST['todelete'] as $var)
{
        $query .= " ? ,";
        array_push($params, $var);
}
$query = rtrim($query, ",");
$query.= ")";

$connessione->doQP($query, $params);
```

Eliminazione tramite ID di tutti gli articoli selezionati dall'amministratore nella pagina precedente.

DeleteSpedAddr.php

```
$query = "      DELETE FROM Indirizzo_Spedizione
                WHERE IDShipment = ?        ";
$connessione->doQP($query , $_POST['ID'] );
```

Con questa query viene eliminato l'indirizzo di spedizione scelto, se l'utente ha deciso di effettuare questa operazione.

EditFattAddr.php

```
$query = "      SELECT *
                FROM Indirizzo_Fatturazione
                WHERE IDUtente = ? ";
$connessione->doQP($query, $_SESSION['UID']);
```

Estrazione degli indirizzi di fatturazione dell'utente per la visualizzazione dei suddetti.

EditFattAddrDB.php

```
$query ="       UPDATE Indirizzo_Fatturazione
                SET
                Name_Surname=?,Address_1=?,Address_2=?,City=?,Province=?,CAP=?,Pa
                ese=?,PartitaIVA=?
                WHERE IDBilling = ?";

$parametri =
array($_POST['Destinatario'],$_POST['Indirizzo1'],$_POST['Indirizzo2'],$_POST['Citta'],$
_POST['Provincia'],$_POST['CAP'],$_POST['Paese'],$_POST['PIVA'], $_POST['ID']);

$connessione->doQP($query,$parametri);
```

Modifica dell'indirizzo di fatturazione scelto dall'utente in base ai parametri passati nel form precedente.

EditFattAddrForm.php

```
$query = "      SELECT *
                FROM Indirizzo_Fatturazione
                WHERE IDBilling = ? ";
$connessione->doQP($query,$_POST['ID']);
```

Estrae i dati dell'indirizzo di fatturazione selezionato per visualizzarli in seguito, permettendo la loro modifica.

EditHL.php

```
if(isset($_GET['AdminSearch']))
{
        $query = "SELECT * FROM Articles WHERE lower(PN) LIKE lower('%?%')";
        $connessione->doQP($query, $_GET['AdminSearch']);
}
else
        $connessione->doQ("SELECT * FROM Articles");
```

Simile a DeleteArticles.php. In questa pagina verranno mostrati gli articoli tutti gli articoli per permettere all'admin di selezionare quelli in evidenza. Se l'admin ha deciso di visualizzare soltanto certi articoli, verrà eseguita la prima query, altrimenti verranno visualizzati tutti gli articoli.

EditHLAction.php

```
$connessione->doQ("UPDATE Articles SET HL = 0");
foreach($_POST['toHL'] as $var)
{
        $query = "UPDATE Articles SET HL = 1 WHERE ID = ?";
        $connessione->doQP($query, $var);
}
```

Tutti gli articoli non selezionati dall'admin non verranno messi in evidenza, poi con un ciclo si settano in evidenza tutti i prodotti selezionati.

EditMail.php

```
$query = "SELECT Password,SALT FROM RegisterData WHERE ID = ?";
$connessione->doQP($query,$_SESSION['UID']);

$query = "UPDATE RegisterData SET Email = ? WHERE ID = ?";
$connessione->doQP($query,array($_POST['NewMail'], $_SESSION['UID']));
```

Per modificare la mail, l'utente deve inserire il suo username e la password, la prima query seleziona le suddette dal database per il successivo controllo.
Dopo aver verificato la corrispondenza, la seconda query inserisce nel database la nuova mail.

EditPass.php

```
$query = "SELECT Password, SALT FROM RegisterData WHERE ID = ?";
$connessione->doQP($query,$_SESSION['UID']);

$query = "UPDATE RegisterData SET Password = ? , SALT = ? WHERE ID = ?";
$connessione->doQP($query, array($NewHashedPassword, $newSalt,
$_SESSION['UID']));
```

In maniera identica alla modifica della mail, la prima query seleziona la password dal database per confrontarla con quella inserita dall'utente, se corrispondono, si può procedere con la seconda query e immagazzinare nel database la nuova password.

EditSpedAddr.php

```
$query = "      SELECT *
                FROM Indirizzo_Spedizione
                WHERE IDUtente = ?";
```

Con questa query vengono estratti i gli indirizzi di spedizione dell'utente per permetterne la visualizzazione e la successiva modifica da parte dell'utente.

EditSpedAddrDB.php

```
$query ="       UPDATE Indirizzo_Spedizione
                SET
                Name_Surname=?,Address_1=?,Address_2=?,City=?,Province=?,CAP=?,Pa
                ese=?,Phone=?
                WHERE IDShipment = ?";
$parametri =
array($_POST['Destinatario'],$_POST['Indirizzo1'],$_POST['Indirizzo2'],$_POST['Citta'],$
_POST['Provincia'],$_POST['CAP'],$_POST['Paese'],$_POST['Tel'], $_POST['ID']);

$connessione->doQP($query,$parametri);
```

Modifica dell'indirizzo di spedizione scelto precedentemente dall'utente in base ai parametri passati dal form.

EditSpedAddrForm.php

```
$query = "SELECT * FROM Indirizzo_Spedizione WHERE IDShipment = ?";
$connessione->doQP($query,$_POST['ID']);
```

Estrae i campi dell'indirizzo di spedizione scelto precedentemente per la visualizzazione nel form di modifica/eliminazione.

RemoveFBasket.php

```
$query = "      DELETE FROM Basket
                WHERE AID IN ( ";
foreach ($_POST['basketRemove'] as $var)
{
        $query .= "? ,";
        array_push($params, $var);
}
$query = rtrim($query, ",");
        $query.= " ) ";

$query .= " AND UID = ? ";
array_push($params, $_SESSION['UID']);

$connessione->doQP($query, $params);
```

Rimuove tutti gli articoli selezionati dall'utente nel carrello.

Ricerca_Prodotto.php

```
if(empty($cat) || $cat == 'All')
{
        $query = "SELECT * FROM Articles WHERE lower(PN) LIKE '%?%'";
        $connessione ->doQP($query, $_GET['search']);
}
else
{
        $query = "SELECT * FROM Articles WHERE lower(PN) LIKE '%?%' AND Category =
        '?'";
        $connessione ->doQP($query, array($_GET['search'], $cat));
}
```

Questa query restituisce gli articoli combacianti con il termine di ricerca digitato. In più se l'utente ha selezionato una categoria, verrà eseguita la seconda query che selezionerà dal database solamente i prodotti coincidenti con il termine di ricerca e la categoria scelta dall'utente.

RimozioneAccount.php

```
$query = "SELECT Username, Password, SALT FROM RegisterData WHERE ID = ?";
$conn->doQP($query, $_SESSION['UID']);

$query = "DELETE FROM RegisterData WHERE ID = ?";
$conn->doQP($query, $_SESSION['UID']);
```

Similarmente alla modifica della password e della mail, la prima query seleziona password ed username dell'account per la verifica con i dati inseriti prima di poter procedere all'eliminazione, se la validazione è andata a buon fine, lo script procede con la seconda query che si occupa di eliminare dal database l'account e con esso, tutti i record collegati con una relazione del tipo "ON DELETE CASCADE".

Stats.php

```
$query = "      SELECT * FROM Ricevuta
                WHERE Data_Ordine >
                STR_TO_DATE(? , '%d/%m/%Y')" ;
$connessione->doQP($query,date("d/m/Y",strtotime(date('Y/m/d') . "-1 month")));
```

Estrae tutti gli ordini/ricevute da un mese ad oggi, utilizzate per la visualizzazione delle statistiche delle vendite.

ViewOrderAction.php

```
if(isset($_GET['AdminSearch']))
{
        $query = "      SELECT * FROM Ricevuta
                        WHERE IDUtente = (SELECT ID FROM RegisterData WHERE
                        lower(Username) = lower(?))";
        $connessione->doQP($query, $_GET['AdminSearch']);
}
elseif(isset($_GET['AdminSearchMail']))
{
        $query = "      SELECT * FROM Ricevuta
                        WHERE email = ?";
        $connessione->doQP($query, $_GET['AdminSearchMail']);
}
else
{
        $query = "SELECT * FROM Ricevuta";
        $connessione->doQ($query);
}
```

Pagina per l'admin in cui è possibile visualizzare tutte le ricevute oppure mostrare solo quelle che soddisfano la ricerca.
Se l'admin decide di cercare per username dell'utente che ha effettuato l'ordine, viene prima selezionato l'ID del suddetto dalla tabella RegisterData in cui è memorizzato, e poi selezionati tutte le ricevute registrate con quell'ID.
Oppure se l'admin ha deciso di visualizzare solo le ricevute registrate con una certa mail, viene eseguita la seconda query.
Infine, se nessuna delle due scelte precedenti è stata selezionata, si procede con la visualizzazione di tutte le fatture.

VisualizzaOrdini.php

```
$query = "      SELECT * FROM Ricevuta
                WHERE IDUtente = ?";
$connessione->doQP($query,$_SESSION['UID']);
```

Pagina dell'utente in cui verranno visualizzati tutti gli ordini effettuati da quest'ultimo. La query seleziona i suddetti.

addToBasket.php

```
$query = "      SELECT Quantity
                FROM Basket WHERE UID = ? AND AID = ?";
$connessione->doQP($query, array($_SESSION['UID'], $_POST['PID']));
```

Estrazione del numero di pezzi dello stesso articolo aggiunti al carrello. Se sommando la quantità che si sta provando ad aggiungere si eccede la quantità presente in magazzino, l'articolo non verrà aggiunto al carrello.

```
$query = "      SELECT N_Left,Availability
                FROM Articles
                WHERE ID = ?";
$connessione->doQP($query,$_POST['PID']);
```

Controllo sugli articoli che l'utente intende aggiungere al carrello, se sono esuariti oppure in arrivo, non sarà possibile aggiungere l'articolo al carrello.

```
$query = "      SELECT AID, UID
                FROM Basket
                WHERE UID = ? AND AID = ?";
$connessione->doQP($query, array($_SESSION['UID'], $_POST['PID']));

if($connessione->getNumRows())      {
        $query = "      UPDATE Basket
                        SET Quantity = Quantity + ?
                        WHERE UID = ? AND AID = ?";
else
        $query = "      INSERT INTO
                        Basket (Quantity, UID, AID)
                        VALUES (?, ?, ?)";

$connessione->doQP($query, array($_POST['Quantita'],$_SESSION['UID'], $_POST['PID']));
```

Controllo finale, la prima query controlla se l'utente ha già lo stesso articolo nel carrello. Se la query non ritorna record, ne verrà creato uno nuovo nel carrello, altrimenti verrà semplicemente aumentata la quantità.

adminconn.php

```
$query = "SELECT * FROM AdminData WHERE lower(AdminUsername) = ?;";
$connessione->doQP($query, $username);
```

Estrazione dei dati dalla tabella con i dati degli amministratori, per verificare la corrispondenza con quelli inseriti nel login.

editAction.php

```
$query = "UPDATE Articles SET PN = ?, Manufacturer = ?, Seller = ?, Availability = ?,
N_Left = ?, Description = ?, Price = ?, Category = ? WHERE ID = ?";

$params = array($_POST['PN'], $_POST['Manufacturer'], $_POST['Seller'],
$_POST['Availability'], $_POST['N_Left'], $_POST['Description'], $_POST['Price'],
$_POST['Category'], $_POST['PID']);

$connessione->doQP($query, $params);
```

Pagina per l'admin. La query modifica i campi dell'articolo selezionato in precedenza.

editItems.php

```
if(isset($_GET['AdminSearch']))
{
        $query = "SELECT * FROM Articles WHERE lower (PN) LIKE lower('%?%')";
        $connessione->doQP($query, $_GET['AdminSearch']);
}
else
{
        $query = "SELECT * FROM Articles";
        $connessione->doQ($query);
}
```

Pagina per l'admin in cui selezionare l'articolo da modificare. Verranno visualizzati, tramite la prima query, soltanto gli articoli che soddisfano i parametri di ricerca. Se non viene specificato nessuno, verranno visualizzati tutti gli articoli presenti nel database.

index.php

```
$connessione->doQ("SELECT * FROM Articles WHERE HL = 1");
```

Selezione di tutti gli articoli che verranno visualizzati nella pagina principale.

upload.php

```
$query = "      INSERT INTO Articles (PN, Manufacturer, Seller, Availability, N_Left,
                Description, IMG, Price, Category)
                VALUES (?,?,?,?,?,?,?,?,?)";
$connessione->doQP($query, array($_POST['titolo'], $_POST['produttore'],
$_POST['distributore'], $_POST['disponibilita'], $_POST['n_pezzi'],
$_POST['descrizione'], $newname, $_POST['prezzo'], $_POST['categoria']));
```

Inserimento del database dell'articolo inserito dall'admin nel form precedente.