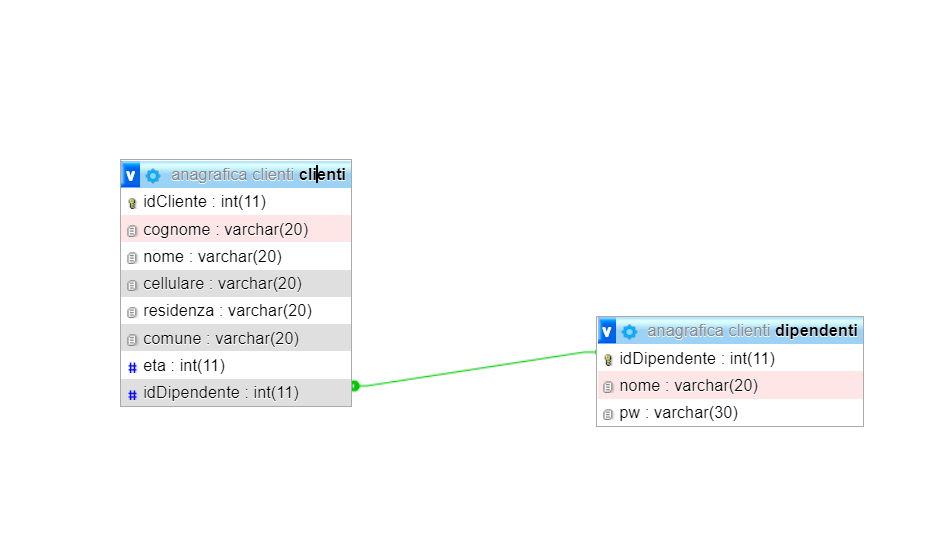
17/09/2021

**Documento di analisi tecnica , “Anagrafica clienti di un negozio”.**

analisi del codice del software utile nella gestione dei clienti di un negozio.

**database:** il database scelto per il progetto è un relazionale mysql, sarà diviso in due tabelle con cardinalità 1 a N, vi sono due entità: l’entità dipendente (idDipendente, nome, pw)

e l'entità cliente(idCliente, cognome, nome, cellulare, residenza, comune, eta, **fk: idDipendente**).



**premessa:** le pagine di codice si dividono in 3 gruppi

1. .php, queste pagine contengono codice HTML e poco codice php non in chiave backend, servono principalmente a mostrare l’interfaccia al cliente.
2. .inc.php queste sono le pagine contenenti codice php in modo rilevante e servono per gestire la logica delle pagine e la comunicazione con il database
3. .css queste sono i fogli di stile e servono a creare delle interfacce personalizzate e dalla grafica quanto più accattivante.

**db.inc.php**

questa pagina contiene le informazioni ed i parametri necessari a stabilire la connessione con il server database, viene creato l’oggetto “$conn” utile per non ripetere il codice di connessione anche nelle pagine successive, la pagina php esegue dei controlli sulla buona riuscita della connessione e in caso contrario restituisce a schermo il corrispondente codice di errore mysqli.

**login.php**

Come prime istruzioni questa pagina controlla l’eventuale esistenza di un ID di sessione ed in caso affermativo reindirizza l’utente alla pagina principale della applicazione web (index.php) .

Successivamente contiene del codice HTML che costruisce il form e gli input necessari per il login.

l’action del form consiste nel indirizzare alla pagina login.inc.php, viene usato il metodo POST in quanto sono trasmesse delle informazioni sensibili quali username e password.

**login.inc.php**

In questa pagina viene eseguito esclusivamente il codice php che consiste nel controllo delle credenziali di accesso del login.

innanzitutto viene avviata una sessione in modo da autenticare l’utente.

richiede la pagina di connessione al database db.inc.php.

viene eseguito un controllo che consiste nella verifica dello scambio di informazioni con il server tramite metodo POST.

viene controllato che il pulsante di login sia stato premuto e successivamente vengono create due variabili: $username e $password tramite l'utilizzo della funzione di msql improved: mysqli\_real\_escape\_string() che previene l’utente dall’inserimento di codice malevolo all’interno del database.

viene creata la variabile $result che viene usata come risposta dal database con la query interrogativa predisposta per il controllo delle credenziali

("SELECT \* FROM dipendenti WHERE nome = '$username' and pw = '$password'"

)

viene creata la variabile $row che consiste nella risposta del database tramite la funzione

mysqli\_fetch\_array.

$row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

successivamente controlla le credenziali ed imposta l’ID di sessione.

l’ID di sessione è il risultato di un algoritmo di hashing SHA-1 e lo username

$\_SESSION['id'] = sha1($username);

controlla che l’ID di sessione sia stato impostato e indirizza alla pagina principale, altrimenti restituisce un messaggio che indica un’errore nella creazione della sessione.

**loginStyle.css**

foglio di stile per la pagina di login, qui imposto lo stile del body(colore di sfondo, allineamenti ), imposto la classe container (.myBox) per gli input dell'utente e do uno stile ai pulsanti (senza cursore e successivamente con il cursore posizionato sopra) e ai field di input.

do uno stile anche al titolo della pagina per renderlo graficamente più apprezzabile.

**index.php**

pagina principale dell'applicazione web, da qui si ha accesso a tutte le funzionalità

il codice principale consiste nell'impostazione di un wrapper per consentire lo scorrimento di un menù a tendina che contiene dei link alle pagine contenenti il codice di tutte le funzionalita.

<li><a href="nuovoCliente.php">Nuovo cliente</a></li>

**<li><a href="listaClienti.php">Lista clienti</a></li>**

**<li><a href="cercaClienti.php">Cerca clienti</a></li>**

**indexStyle.css**

foglio di stile per la pagina principale da qui gestisco i movimenti del menù a tendina in puro css senza l'utilizzo di javaScript. da qui do uno stile al body e gestisco le transizione necessarie per ottenere l’effetto dello scorrimento del menù:

transition: opacity 1.2s, transform 1.2s cubic-bezier(0.215, 0.61, 0.355, 1);

transform: translateX(100px);

**nuovoCliente.php**

contiene il codice HTML necessario per la costruzione di un form che verrà usato dall’utente per inserire i dati anagrafici del cliente da registrare.

l’action del form contiene “nuovoCliente.inc.php”.

questa pagina contiene anche due pulsanti: uno necessario per la registrazione del cliente tramite il link del form, mentre l’altro consiste in codice JavaScript per consentire di tornare alla home page (index.php)

<button onclick="alert('Cliente aggiunto con successo al database')"; class="button" type="submit" name="salvaNuovoClienteBTn" id="salvaNuovoClienteBtn">Salva</button>

<button onclick="vaiHome();" class="button">Home</button>

lo script javaScript per tornare alla home page:

function vaiHome(){

window.location.href = "index.php";

}

**nuovoCliente.inc.php**

pagina di puro codice PHP, viene richiesta la connessione al database ed inizializzate le variabili contenenti i dati del cliente da registrare, gli input vengono “sanificati ” tramite le funzioni mysqli\_real\_escape\_string()

$cognome = mysqli\_real\_escape\_string($conn,($\_POST["cognome"]));

successivamente viene impostata una query di inserimento per caricare i dati nel database:

$result = mysqli\_query($conn,"INSERT INTO `clienti` (`idCliente`, `cognome`, `nome`, `cellulare`, `residenza`, `comune`, `eta`)

VALUES (NULL, '$cognome', '$nome', '$cellulare', '$residenza', '$comune', '$eta')");

e si viene indirizzati alla pagina principale tramite una funzione in php

header("location: index.php");

**nuovoClienteStyle.css**

contiene le stesse classi utilizzate in precedenza per modellare lo stile della pagine.

Lo stile, infatti, è uniforme per tutte le pagine della web app per non confondere il cliente con interfacce dai colori e stili troppo differenti

**listaClienti.php**

questa pagina contiene il codice HTML necessario per definire la pagina di accesso alla tabella contente i dati dei clienti, definisce anche un form nel quale l’utente, tramite input, decide i parametri dei filtri da applicare alla tabella.

il form rimanda alla pagina che processa le informazioni e i filtri e mostra la tabella

(listaClienti.inc.php).

il pulsante che definisce i filtri in realtà è un sottotitolo mascherato da pulsante che gira del codice javascript in modo da ottenere l'effetto del menu dropdown.

<h2 onclick="myFunction()" class="dropbtn">Seleziona filtri</h2>

function myFunction() {

document.getElementById("myDropdown").classList.toggle("show");

}

**listaClienti.inc.php**

questa pagina definisce la tabella dei clienti tramite l’apposito codice html.

viene richiesta la connessione al database e vengono estratti i valori dei filtri da applicare dal form presente nella pagina precedente.

controllo se sono stati impostati dei filtri, in caso contrario verrà mostrata la tabella clienti nella sua interezza

if (empty($comune) && empty($eta) && empty($nome)){

$query = "SELECT \* FROM clienti";

imposto anche il codice necessario per passare i valori dei clienti alla pagina di modifica cliente e di cancellazione del cliente tramite URL con il metodo GET

<a href = 'cancellaCliente.php?idClienteCanc=$row[idCliente]'

viene anche eseguita una concatenazione dei valori del record per poter essere inseriti all'interno della HTML table tramite un insieme di html e php

echo "<tr><td>".$row['idCliente']."</td><td>". $row["cognome"] . "</td><td>" . $row["nome"] . "</td><td>" . $row["cellulare"] . "</td><td>" . $row["residenza"]

."</td><td>". $row["comune"] . "</td><td>". $row["eta"] .

vengono effettuati dei controlli delle variabili impostate durante la scelta dei filtri, se la condizione if restituisce false significa che il filtro non è stato impostato, in caso contrario la query interrogativa viene adattata a seconda del filtro impostato.

if(!empty($comune)){

$queryComune = "SELECT \* FROM clienti WHERE clienti.comune = '$comune'";

**listaClientiStyle.css**

contiene le stesse classi usate in precedenza per modificare lo stile dei container per gli input.

contiene anche le classi necessaria a definire lo stile della tabella dei clienti.

**modificaCliente.php**

è la pagina che si apre tramite il link “modifica ” nella tabella dei clienti

intercetta i dati contenuti nel URL della pagina contenente la tabella e li salva all'interno di nuove variabili che vengono mostrate all’interno del apposito form di modifica dei dati del cliente specifico selezionato.

<input id = "nuovoCognome" name="nuovoCognome" value="<?php echo "$cognC" ?>" type="text">

<label for="nuovoCognome">Cognome</label>

il valore del input viene inizializzato con il valore che il cliente riportava prima di accedere alla pagina di modifica, tramite l’attributo “value” nel campo di input.

Sono presenti anche due pulsanti che permettono di confermare le modifiche e di tornare alla home page.

**modificaCliente.inc.php**

richiede la connessione al database e passa le informazioni dal form precedente alle nuove variabili.

le stringhe vengono sanificate attraverso mysqli\_real\_escape\_string() e successivamente viene impostata la query di aggiornamento al database tramite ID univoco.

$query = "UPDATE `clienti`

SET `cognome` = '$newCognome', `nome` = ' $newNome', `cellulare` = '$newCellulare',

`residenza` = ' $newResidenza', `comune` = ' $newComune ', `eta` = ' $newEta '

WHERE `clienti`.`idCliente` = '$id' ";

successivamente all’aggiornamento dei dati si viene riportati alla home page tramite codice php.

**cancellaCliente.php**

richiede la connessione al database e riceve tramite URL dalla tabella dei clienti l’ID del cliente specifico da cancellare.

$idClienteCanc = $\_GET["idClienteCanc"];

successivamente imposta la query di eliminazione e procede ad eliminare il cliente selezionato tramite ID univoco.

query = "DELETE FROM `clienti` WHERE `clienti`.`idCliente` = '$idClienteCanc'";

**cercaClienti.php**

costruisco un form con degli input da parte dell’utente nel quale specifica i parametri secondo cui cercare i clienti all’interno del database.

il form possiede come action “cercaClienti.inc.php” ovvero la pagina nella quale queste informazioni verranno processate.

**cercaClienti.inc.php**

si divide in una parte di HTML che consiste solamente nel titolo della pagina e nel container per i dati che verranno mostrati all’utente.

la parte in php, invece, consiste nella chiamata al database, nel controllo del metodo di comunicazione con il server (POST), e nel controllo che il pulsante di ricerca sia stato premuto correttamente.

Successivamente vengono estratti i parametri di ricerca dal form precedente e vengono memorizzati in nuove variabili.

effettuo un controllo sui parametri di ricerca inseriti, infatti se non ne vengono inseriti ma si prova a cercare ugualmente si viene riportati alla pagina di ricerca, assumendo che sia stato premuto il pulsante per errore.

if (empty($id) && empty($nome) && empty($cognome) && empty($cellulare)){

header("location: cercaClienti.php");

}

viene impostato il codice necessario per effettuare una ricerca nel database tramite ID univoco e viene effettuato il fetch dell’array contenente il $result della chiamata al server database

$result = mysqli\_query($conn,"SELECT \* FROM `clienti` WHERE idCliente = '$id'");

while($row = mysqli\_fetch\_array($result))

utilizzando i ciclo while verrà percorso tutto l’array finché contiene righe da mostrare.

successivamente eseguo una concatenazione dei valori all'interno della $row per mostrarli in modo facilmente comprensibile.

echo "<div class='myBox';><h3 style='color:black ;'>" . "ID: " . $row['idCliente'] ."<br>". "cognome: " . $row['cognome'] ."<br> "."nome: ". $row['nome'] ." <br>"."cellulare: ". $row['cellulare'] ."<br> "."residenza: ". $row['residenza'] .

" <br>" . "comune: ". $row['comune'] ." <br>"."età: ". $row['eta'] ."</h3></div>";

ripeto lo stesso procedimento cambiando però la query interrogativa a seconda dei parametri di ricerca impostati dall’utente.

vi sono anche la ricerca tramite cognome, tramite nome e tramite numero di cellulare.