Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 25.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa giornata di progetti mi sono occupato di cominciare la parte amministrativa del sito web. Inizialmente mi sono occupato della pagina web che gestisce la ricezione del pagamento per le varie richieste di chiavi da parte degli utenti, come prima cosa ho ristrutturato la tabella aggiungendo tre colonne in più: Email, numero di chiavi e il costo totale. Questo perché almeno l’amministratore sa l’email, il numero di chiavi e quanto deve pagare ogni nuovo utente.  Header tabella:    In seguito, per agevolarmi le cose, ho creato una tabella temporanea nel database chiavata appunto *temporanea* al suo interno sono presenti i campi presenti nell’header della tabella.  Tabella temporanea:    Per completare il tutto sono tornato nella classe *connection* che si occupa della gestione della parte amministrativa, non che del salvataggio nel database di tutti i dati. Quando vado a richiamare tale funzione aggiungo due parametri di passaggio, cioè il numero di chiavi ed il totale.  Parametri aggiunti:  //Creo delle variabili.  $check = null;  $tot = null;  $keyValue = 200;  //Creao gli oggetti delle classi necessarie;  $conn = new connection();  //Controllo se é stato eseguito il POST  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  $\_SESSION['inputNumberKeys'] = $\_POST['inputNumberKeys'];  $tot = $\_SESSION['inputNumberKeys'] \* $keyValue;  for ($i=1; $i <= $\_SESSION['inputNumberKeys']; $i++) {  if ($\_POST["radioButtonSociety"] == "1") {  $check = $conn->saveNewUser(1,  $\_POST["inputSocietyName"], $\_POST["inputNamePersonInCharge"],  $\_POST["inputSurnamePersonInCharge"], $\_POST["inputTelephoneNumberPersonInCharge"],  $\_POST["inputSocietyAddress"], $\_POST["inputSocietyAddressNumber"],  $\_POST["inputSocietyPostalNumber"], $\_POST["inputSocietyCity"],  $\_POST["inputNameHolder\_".$i], $\_POST["inputSurnameHolder\_".$i],  $\_POST["inputHolderAddress\_".$i], $\_POST["inputHolderAddressNumber\_".$i],  $\_POST["inputHolderPostalNumber\_".$i], $\_POST["inputHolderCity\_".$i], $\_POST["inputCarBrand\_".$i],  $\_POST["inputCarColor\_".$i], $\_POST["inputCarModel\_".$i],  $\_POST["inputCarType\_".$i], $\_POST["inputPlateNumber\_".$i], $\_POST["inputEmailHolder\_".$i],  $\_SESSION['inputNumberKeys'], $tot);  }else {  $conn->saveNewUser(0, "", "", "", "", "", "", "", "",  $\_POST["inputNameHolder\_".$i], $\_POST["inputSurnameHolder\_".$i],  $\_POST["inputHolderAddress\_".$i], $\_POST["inputHolderAddressNumber\_".$i],  $\_POST["inputHolderPostalNumber\_".$i], $\_POST["inputHolderCity\_".$i], $\_POST["inputCarBrand\_".$i],  $\_POST["inputCarColor\_".$i], $\_POST["inputCarModel\_".$i],  $\_POST["inputCarType\_".$i], $\_POST["inputPlateNumber\_".$i], $\_POST["inputEmailHolder\_".$i],  $\_SESSION['inputNumberKeys'], $tot);  }  }  Funzione che salva i dati nelle relative tabelle (parti aggiunte)  /\*\*  \* Questa funzione riceve tutti i dati immessi nella pagina web del formulario  \* e si occupa di salvarli, tramite delle query, nelle rispettive tabelle con i rispettivi controlli.  \* @param society Valore "booleano" che avverte se sono stati inseriti i dati della società(1) oppure no(0).  \* @param societyName Stringa contenente il nome della società.  \* @param personInChargeName Stringa contenente il nome della persona in carica della società.  \* @param personInChargeSurname Stringa contenente il cognome della persona in carica della società.  \* @param personInChargeTelephone Stringa contenente il numero di telefono del responsabile della società.  \* @param societyAddress Stringa contenente la via della società.  \* @param societyAddressNumber Stringa contenente il numero civico della società.  \* @param societyPostalNumber Stringa contenente il CAP della società.  \* @param societyCity Stringa contenente la città della società.  \* @param holderName Stringa contenente il nome del detentore.  \* @param holderSurname Stringa contenente il cognome del detentore.  \* @param holderAddress Stringa contenente la via del detentore.  \* @param holderAddressNumber Stringa contenente il numero civico del detentore.  \* @param holderPostalNumber Stringa contenente il CAP del detentore.  \* @param holderCity Stringa contenente la città del detentore.  \* @param carBrand Stringa contenente la marca dell'auto.  \* @param carColor Stringa contenente il colore dell'auto.  \* @param carModel Stringa contenente il modello dell'auto.  \* @param carType Stringa contenente il tipo di auto.  \* @param plateNumber Stringa contenente il numero di targa.  \* @param emailHolder Stringa contenente l'email del detentore.  \* @param keysNumber Intero contenente il numero di chiavi del detentore.  \* @param totalCost Intero contenente il costo totale delle chiavi ordinate.  \* @return int Ritorno 1 se una delle chiavi primarie che sono state inserite esiste già.  \*/  function saveNewUser($society, $societyName, $personInChargeName,$personInChargeSurname,  $personInChargeTelephone, $societyAddress, $societyAddressNumber, $societyPostalNumber,  $societyCity, $holderName,$holderSurname, $holderAddress, $holderAddressNumber, $holderPostalNumber,  $holderCity, $carBrand, $carColor, $carModel, $carType, $plateNumber, $emailHolder, $keysNumber, $totalCost) {  …  //Query utilizzata per l'inserimento dei dati temporanei all'interno della tabella temporanea.  //Uso un prepared statement per evitare le SQLInjection.  $stmtTemp = $this->conn->prepare("INSERT INTO temporanea VALUES (?,?,?,?,?)");  $stmtTemp->bind\_param("sssii", $emailHolder, $holderName, $holderSurname, $keysNumber, $totalCost);  //Eseguo la query.  if(!$stmtTemp->execute()){  echo "La query di inserimento nella tabella temporanea  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:342).";  }  }  Test    Come si può notare dall’immagine soprastante, una volta che eseguo una nuova richiesta, i dati vengono inseriti con successo nella tabella temporanea.  In seguito mi sono occupati della visualizzazione a schermo, in formato tabellare, dei seguenti dati. Per fare ciò ho creato, inizialmente, l’header della tabella come mostrato in precedenza, in seguito ho creato una funzione che mi estrapoli i dati grazie ad una query.  Funzione che acquisisce i dati dalla tabella temporanea  /\*\*  \* Questa funzione si occupa dell'estrapolazione dei dati dalla teblla temporanea tramite una query.  \* @return list Array di stringhe contenente tutti i dati riguradanti le nuove richieste da confermare.  \*/  function getNewUser() {  $containerData = array();  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Query che permettono l'estrapolazione dei dati riguardanti le nuove richieste,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtViewNewUser = $this->conn->prepare("SELECT \* FROM temporanea");  //Eseguo la query.  if(!$stmtViewNewUser->execute()){  echo "La query di selezione del nome e del cognome dei detentori  presente nella funzione getNewUser() non funziona (riga:133).";  }else {  $result = $stmtViewNewUser->get\_result();  //Ciclo tutti i "dati" che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  array\_push($containerData, $row);  }  return $containerData;  }  }  Il passo successivo è stato la creazione della tabella, la chiamata della funzione per ricevere i dati ed il riempimento di quest’ultima.  Tabella (codice)  <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">  <thead>  <tr>  <th>Nome</th>  <th>Cognome</th>  <th>Email</th>  <th>N° Chiavi</th>  <th>Totale (CHF)</th>  <th>Pagamento effettuato</th>  </tr>  </thead>  <tbody>  <?php for ($i=0; $i < count($newKeysRequest); $i++): ?>  <tr>  <td><?php echo $newKeysRequest[$i]['Nome']; ?></td>  <td><?php echo $newKeysRequest[$i]['Cognome']; ?></td>  <td><?php echo $newKeysRequest[$i]['Email']; ?></td>  <td><?php echo $newKeysRequest[$i]['N\_chiavi']; ?></td>  <td><?php echo $newKeysRequest[$i]['Totale'] . ".-"; ?></td>  <td><input type="checkbox" class="checkBlockedState" name="newUser[]" value="<?php echo $newKeysRequest[$i]['Email']?>"></td>  </tr>  <?php endfor; ?>  </tbody>  </table>  Chiamata della funzione  <?php  $newKeysRequest = null;  //Eseguo il require dei file esterni necessari.  require\_once('php/db\_connection.php');  //Creao gli oggetti delle classi necessarie;  $conn = new connection();  //Controllo se é stato eseguito il POST  if ($\_SERVER['REQUEST\_METHOD'] == "POST") {  $conn->confirmKeysRequest($\_POST['newUser']);  }  $newKeysRequest = $conn->getNewUser();  ?>  Test    Come si può notare dall’immagine soprastante la tabella viene riempita nel modo corretto.  In seguito mi sono occupato di gestire la conferma, per fare ciò ho creato un’ulteriore funzione che riceve come parametro l’email presente nella tabella temporanea e, tramite questo, lo va a cercare nella tabella *detentore* e mette ad 1 il suo flag di pagamento in modo tale che si veda che ha effettuato il pagamento. Questa funzione di occupa anche di inviare una email di conferma al nuovo utente.  Funzione di conferma  /\*\*  \* Questa funzione si occupa della conferma dei pagamenti e della eliminazione  \* di quest'ultimi dalla tabella temporanea tramite delle query.  \* In fine si occupa dell'invio di una mail di conferma al nuovo utente.  \* @param email L'array contenente tutti gli username selezionati.  \*/  function confirmKeysRequest($email) {  $paymentConfirmed = 1;  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Ciclo l'array.  for ($i=0; $i < count($email); $i++) {  //Controllo se l'array contenente l'email dei nuovi utenti selezionati é vuoto oppure no.  if(count($email) != 0) {  //Query che permettono la modifica del flag di conferma del pagamento da parte dell'utente,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtConfirm = $this->conn->prepare("UPDATE detentore set Pagamento\_effettuato = ? where Email = ?");  $stmtConfirm->bind\_param("is", $paymentConfirmed, $email[$i]);  if(!$stmtConfirm->execute()){  echo "La query di modifica della conferma del pagamento nella tabella detentore  presente nella funzione confirmKeysRequest() non funziona (riga:162).";  }  //Query che permettono la cancellazione delle informazioni presenti nella tabella temporanea,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtDelete = $this->conn->prepare("DELETE FROM temporanea WHERE Email = ?");  $stmtDelete->bind\_param("s", $email[$i]);  if(!$stmtDelete->execute()){  echo "La query di concellazione nella tabella temporanea  presente nella funzione confirmKeysRequest() non funziona (riga:168).";  }  //Richiamo le funzioni che si occupano di stampare a schermo le due tabelle degli amministratori bloccati e non.  $this->getNewUser();  }else {  echo "Non hai selezionato nessun utente";  }  //preparo le variabili che servono a mandare l'email di conferma all'utente  /\*  $to = "alessandro.gomes@samtrevano.ch";  $subject = "Conferma invito";  $txt = "Ciao !<br>";  $headers = 'MIME-Version: 1.0' . "\r\n";  $headers .= 'From: Administrator'. "\r\n";  $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";  //invio l'email  mail($to,$subject,$txt, $headers);\*/  }  }  Test  Una volta che seleziono un checkbox presente nella pagina web e premo invio quell’utente viene tolto con successo dalla tabella temporanea ed il suo flag viene cambiato con successo da 0 a 1.  Ora mi sto occupando dell’invio della mail all’utente, il problema è che sto lavorando in locale e quindi devo utilizzare i server mail di gmail e per fare ciò, inizialmente, ho creato una mail amministrativa in gmail.    Una volta fatto questo ho dovuto modificare delle informazioni presenti nel file *php.ini*, ho abilitato l’invio delle email tramite openssl 🡪 php\_openssl.dll. In seguito ho dovuto modificare la parte denominata *[mail function]* che contiene tutte le informazioni che servono per inviare una mail.  [mail function]  [mail function]  ; For Win32 only.  ; http://php.net/smtp  SMTP=smtp.gmail.com  ; http://php.net/smtp-port  smtp\_port=25  ; For Win32 only.  ; http://php.net/sendmail-from  sendmail\_from = gestionetrafficoveicolare@gmail.com  In seguito ho modificato la parte denominata [sendamil] presente nel file *sendmail.ini* nel modo seguente.  [sendmail]  [sendmail]  ; you must change mail.mydomain.com to your smtp server,  ; or to IIS's "pickup" directory. (generally C:\Inetpub\mailroot\Pickup)  ; emails delivered via IIS's pickup directory cause sendmail to  ; run quicker, but you won't get error messages back to the calling  ; application.  smtp\_server=smtp.gmail.com  ; smtp port (normally 25)  smtp\_port=25  ; SMTPS (SSL) support  ; auto = use SSL for port 465, otherwise try to use TLS  ; ssl = alway use SSL  ; tls = always use TLS  ; none = never try to use SSL  smtp\_ssl=auto  ; the default domain for this server will be read from the registry  ; this will be appended to email addresses when one isn't provided  ; if you want to override the value in the registry, uncomment and modify  ;default\_domain=mydomain.com  ; log smtp errors to error.log (defaults to same directory as sendmail.exe)  ; uncomment to enable logging  error\_logfile=error.log  ; create debug log as debug.log (defaults to same directory as sendmail.exe)  ; uncomment to enable debugging  ;debug\_logfile=debug.log  ; if your smtp server requires authentication, modify the following two lines  auth\_username=gestionetrafficoveicolare@gmail.com  auth\_password=Password&1 |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante questa lezione di progetti non ho riscontrato nessun tipo di errore. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In ritardo rispetto alla pianificazione |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare la parte amministrativa del progetto. |