Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 26.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante la mattina di questa lezione di progetti ho, inizialmente, fatto una piccola riunione con il mio mandante per fare un po’ il punto della situazione. Dopo ho dovuto modificare i due gantt, questo perché ho dovuto aggiungere alcune attività che riguardavano le pagine di registrazioni e controlli perché quando ho eseguito la pianificazione non avevo previsto queste implementazioni.  Gantt preventivo  C:\Users\Gomes\AppData\Local\Microsoft\Windows\INetCacheContent.Word\GanttPreventico.png  Una volta fatto ciò ho cominciato a guardare un po’ cosa dovessi fare con il raspberry, inizialmente ho effettuato tutti i collegamenti (mouse, tastiera, schermo, memoria ed alimentazione) per poi collegarlo ad Internet tramite la rete Wi-Fi del CPT ed infine ho scaricato il sistema operativo raspbian. In parallelo a questo mi sono fatto sbloccare dal sore Raimondi la porta per poter inviare le email da locale (protocollo TLS, porta 587). Le impostazioni finali di xampp sono le seguenti:  Php.ini 🡪[mail function]  [mail function]  ; For Win32 only.  ; http://php.net/smtp  SMTP=smtp.gmail.com  ; http://php.net/smtp-port  smtp\_port=587  ; For Win32 only.  ; http://php.net/sendmail-from  sendmail\_from = gestionetrafficoveicolare@gmail.com  ; For Unix only. You may supply arguments as well (default: "sendmail -t -i").  ; http://php.net/sendmail-path  sendmail\_path ="C:\xampp\sendmail\sendmail.exe -t"  sendmail.ini 🡪[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=587  ; 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=tls  ; 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="localhost"  ; 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="Pregassona99"  Test  Eseguendo un test si può vedere che le email vengono inviate con successo ai destinatari.  In seguito ho terminato la seconda pagina della parte amministrativa, cioè la pagina che gestisce la conferma dei pagamenti delle nuove richieste di chiavi. Inizialmente ho redatto la funzione che mi permette di riempire la tabella con tutte le nuove richieste.  Funzione di riempimento  /\*\*  \* 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:130).";  }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 $containerData;  }  }  In seguito ho redatto la funzione che gestisce la conferma dei pagamenti e l’invio dell’email di conferma.  Funzione di gestione delle conferme ed email  /\*\*  \* Questa funzione si occupa della conferma dei pagamenti e della eliminazione  \* di quest'ultimi dalla tabella temporanea tramite delle query.  \* In fine si occupa dell'invio di una mail di conferma al nuovo utente.  \* @param email L'array contenente tutti gli username selezionati.  \*/  function confirmKeysRequest($email) {  $paymentConfirmed = 1;  $containerData = array();  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Controllo se l'array contenente l'email dei nuovi utenti selezionati é vuoto oppure no.  if(count($email) != 0) {  //Ciclo l'array.  for ($i=0; $i < count($email); $i++) {  //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 l'estrapolazione del nome e cognome dell'utente, il numero di chiavi e il costo totale.  //Utilizzando un prepare statement per evitare delle SQLInjection.  $stmtSelect = $this->conn->prepare("SELECT Nome,Cognome,N\_chiavi,Totale FROM temporanea where Email = ?");  $stmtSelect->bind\_param("s", $email[$i]);  if(!$stmtSelect->execute()){  echo "La query d'estrapolazione delle informazioni del detentore  presente nella funzione confirmKeysRequest() non funziona (riga:170).";  }else {  $result = $stmtSelect->get\_result();  //Ciclo tutti i dati che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  //preparo le variabili che servono a mandare l'email di conferma all'utente  $to = $email[$i];  $subject = "Conferma pagamento";  $txt =  "Salve ".$row['Nome']." ".$row['Cognome']."!<br>".  "Il suo pagamento per la richiesta della chiave/i é stato ricevuto!<br>".  "La sua chiave/i le verrà spedita a breve.<br>".  "Qui di seguito troverà le informazioni riguardanti il suo acquisto:".  "<ul><li>N° chiavi: ".$row['N\_chiavi']."</li><li>Password: ".$row['Totale']."</li></ul>";  $headers = 'MIME-Version: 1.0' . "\r\n";  $headers .= 'From: gestionetrafficoveicolare@gmail.com'. "\r\n";  $headers .= 'Content-type: text/html; charset=UTF-8' . "\r\n";  //invio l'email  mail($to,$subject,$txt, $headers);  }  }  //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";  }  }  Test  Eseguo una nuova richiesta immettendo la mia mail di scuola:    Entro nella parte amministrativa e controllo che la richiesta sia effettivamente presente nella tabella.    Ora clicco sul checkbox e poi sul bottone. Il risultato è che il falg presente nella tabella del detentore che avvisa se il pagamento è stato effettuato o meno (default🡪0) diventa 1 ed il detentore riceve una mail di conferma.    Nelle ultime ore di lezione ho apportato alcune modifiche alle tipologie di dati in alcuni campi nel database, in particolare ho modificato i due campi *date* della tabella ponte *appartiene* per agevolarmi l’inserimento delle date.  Tabella appartiene    Nella funzione principale che gestisce il salvataggio dei dati nelle varie tabelle ho applicato alcune modifiche ed aggiunte, innanzitutto ho aggiunto la query che mi permette di riempire la tabella *appartiene*, in seguito ho implementato un controllo che impedisca l’inserimento errato degli utenti nella tabella temporanea.  Funzione saveNewUser(..)  /\*\*  \* 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.  \* @param flag Intero contenente un flag che controlla quante volte eseguo l'inseriemnto nella tabella temporanea.  \* @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, $flag) {  //Creazione delle variabili importanti.  $null = null;  $value = null;  $result = null;  $temp = null;  $returned = 0;  $payment = 0;  $ret = null;  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Controllo se l'utente ha inserito i dati relativi alla società.  if ($society == 1) {  //Query che permette l'estrapolazione della chiave con l'identificativo (N\_chiave) maggiore.  $stmtKeySelect = $this->conn->prepare("SELECT N\_chiave FROM chiave ORDER BY N\_chiave DESC LIMIT 1");  //Eseguo la query, ed incremento di uno il risultato.  if ($stmtKeySelect->execute()) {  $value = $stmtKeySelect->get\_result();  $result = $value->fetch\_assoc();  $temp = $result["N\_chiave"] + 1;  }else {  echo "La query di selezione della chiave con l'Id maggiore  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:155).";  }  //Query che inserisce una nuova riga nella tabella "chiave",  //questa riga contiene l'identificativo più alto incrementato di 1.  //Uso un prepared statement per evitare le SQLInjection.  $stmtKeyInsert = $this->conn->prepare("INSERT INTO chiave VALUES (?, ?)");  $stmtKeyInsert->bind\_param("ii", $temp, $returned);  //Eseguo la query.  if(!$stmtKeyInsert->execute()){  echo "La query di inserimento nella tabella chiave  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:168).";  }  //Controllo se il nome della società é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tre tabelle.  if ($this->checkSocietyDuplicate($societyName) == null) {  //Query utilizzata per l'inserimento dei dati relativi alla società all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtSociety = $this->conn->prepare("INSERT INTO societa VALUES (?,?,?,?,?,?,?,?)");  $stmtSociety->bind\_param("ssssssss", $societyName, $personInChargeTelephone, $personInChargeName,  $personInChargeSurname, $societyAddress, $societyAddressNumber, $societyPostalNumber, $societyCity);  //Eseguo la query.  if(!$stmtSociety->execute()){  echo "La query di inserimento nella tabella società  presente nella funzione saveNewUser() non funziona (riga:180).";  }  //Controllo se il nome del detentore é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tre tabelle.  if ($this->checkHolderDuplicate($emailHolder) == null) {  //Query utilizzata per l'inserimento dei dati del detentore all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?,?)");  $stmtHolder->bind\_param("sssssssis", $emailHolder, $holderName, $holderSurname, $holderAddress,  $holderAddressNumber, $holderPostalNumber, $holderCity, $payment, $societyName);  //Eseguo la query.  if(!$stmtHolder->execute()){  echo "La query di inserimento nella tabella detentore  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:194).";  }  //Controllo se il numero di targa é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tabelle.  if ($this->checkCarDuplicate($plateNumber) == null) {  //Query utilizzata per l'inserimento dei dati dell'auto all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo la query.  if(!$stmtCar->execute()){  echo "La query di inserimento nella tabella auto  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:207).";  }  }else {  $ret = 1;  }  }else {  //Query utilizzata per l'inserimento dei dati dell'auto all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo la query.  if(!$stmtCar->execute()){  echo "La query di inserimento nella tabella auto  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:220).";  }  $ret = 1;  }  //Se il nome della società é già presente eseguo soltanto due query di inseriemtno.  }else {  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Controllo se il nome del detentore é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tabelle.  if ($this->checkHolderDuplicate($emailHolder) == null) {  //Query utilizzata per l'inserimento dei dati del detentore all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?, ?)");  $stmtHolder->bind\_param("sssssssis", $emailHolder, $holderName, $holderSurname, $holderAddress,  $holderAddressNumber, $holderPostalNumber, $holderCity, $payment, $societyName);  //Eseguo la query.  if(!$stmtHolder->execute()){  echo "La query di inserimento nella tabella detentore  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:238).";  }  //Controllo se il numero di targa é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tabelle.  if ($this->checkCarDuplicate($plateNumber) == null) {  //Query utilizzata per l'inserimento dei dati dell'auto all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo la query.  if(!$stmtCar->execute()){  echo "La query di inserimento nella tabella auto  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:251).";  }  }else {  $ret = 1;  }  }else {  //Query utilizzata per l'inserimento dei dati dell'auto all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo la query.  if(!$stmtCar->execute()){  echo "La query di inserimento nella tabella auto  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:264).";  }  $ret = 1;  }  $ret = 1;  }  //Se i dati relativi alla società non sono satati inseriti allora eseguo praticamente le stesse  //query, tranne quella della società e come chiave esterna nella tabella detentore immetto null.  }else {  //Query che permette l'estrapolazione della chiave con l'identificativo (N\_chiave) maggiore.  //Uso un prepared statement per evitare le SQLInjection.  $stmtKeySelect = $this->conn->prepare("SELECT N\_chiave FROM chiave ORDER BY N\_chiave DESC LIMIT 1");  //Eseguo la query, ed incremento di uno il risultato.  if ($stmtKeySelect->execute()) {  $value = $stmtKeySelect->get\_result();  $result = $value->fetch\_assoc();  $temp = $result["N\_chiave"] + 1;  }else {  echo "La query di selezione della chiave con l'Id maggiore  presente nella funzione saveNewUser() non funziona (senza dati relativi alla società) (riga:280).";  }  //Query che inserisce una nuova riga nella tabella "chiave",  //questa riga contiene l'identificativo più alto incrementato di 1.  //Uso un prepared statement per evitare le SQLInjection.  $stmtKeyInsert = $this->conn->prepare("INSERT INTO chiave VALUES (?,?)");  $stmtKeyInsert->bind\_param("ii", $temp, $returned);  //Eseguo la query.  if(!$stmtKeyInsert->execute()){  echo "La query di inserimento nella tabella chiave  presente nella funzione saveNewUser() non funziona (senza dati relativi alla società) (riga:293).";  }  //Controllo se il nome del detentore é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tabelle.  if ($this->checkHolderDuplicate($emailHolder) == null) {  //Query utilizzata per l'inserimento dei dati del detentore all'interno della tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?,?)");  $stmtHolder->bind\_param("sssssssis", $emailHolder, $holderName, $holderSurname, $holderAddress,  $holderAddressNumber, $holderPostalNumber, $holderCity, $payment, $null);  //Eseguo la query.  if(!$stmtHolder->execute()){  echo "La query di inserimento nella tabella detentore  presente nella funzione saveNewUser() non funziona (senza dati relativi alla società) (riga:305).";  }  //Controllo se il numero di targa é già presente nel databse.  //Se non é presente eseguo le query d'inseriemnto nelle tabelle.  if ($this->checkCarDuplicate($plateNumber) == null) {  //Query utilizzata per l'inserimento dei dati dell'auto nella tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo la query.  if(!$stmtCar->execute()){  echo "La query di inserimento nella tabella auto  presente nella funzione saveNewUser() non funziona (riga:318).";  }  }else {  $ret = 1;  }  }else {  //Query utilizzata per l'inserimento dei dati dell'auto nella tabella corretta.  //Uso un prepared statement per evitare le SQLInjection.  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo la query.  if(!$stmtCar->execute()){  echo "La query di inserimento nella tabella auto  presente nella funzione saveNewUser() non funziona (riga:331).";  }  $ret = 1;  }  }  //Controllo che i dati vengano inseriti nella tabella temporanea una sola volta.  if ($flag == $keysNumber) {  //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:345).";  }  }  //Prendo la data corrente.  $timeStart = time();  $orderTime = date('d/m/Y', $timeStart);  //Query utilizzata per l'inserimento dei dati all'interno della tabella appartiene.  //Uso un prepared statement per evitare le SQLInjection.  $stmtTemp = $this->conn->prepare("INSERT INTO appartiene (Da, A, N\_chiave, N\_targa) VALUES (?,?,?,?);");  $stmtTemp->bind\_param("ssis", $orderTime, $null, $temp, $plateNumber);  //Eseguo la query.  if(!$stmtTemp->execute()){  echo "La query di inserimento nella tabella ponte appartiene  presente nella funzione saveNewUser() non funziona (con dati relativi alla società) (riga:358).";  }  //Se ci sono dei dati dublicati allora ritorno un 1 per avvertire l'utente.  if ($ret == 1) {  return 1;  }  } |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante questa lezione di progetti ho riscontrato alcuni problemi, il primo riguardava l’accesso all’account gmail per inviare le email tramite php, per sistemarlo ho attivato l’accesso all’account da dispositivi non sicuri. Un altro errore riguarda il salvataggio dei dati all’interno della tabella temporanea, quando eseguivo una richiesta contenente più chiavi me le andava ad inserire tutte quante e, così facendo, ogni utente doveva pagare tutto il prezzo (Es.: una richiesta contenente 7 chiavi, ogni utente doveva pagare il prezzo di 7 chiavi). Per risolvere questo ho semplicemente immesso una flag che controlla il numero di volte che si esegue l’inserimento nella tabella temporanea. |

|  |
| --- |
| 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 e cominciare con raspberry. |