Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 11.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante la mattina, come prima cosa, mi sono occupato della gestione della possibilità che venga inserita una società già esistente all’interno del database. Per fare ciò ho semplicemente eseguito una select che ritorna 1 se il nome della società è già presente, mentre null se non è stata ancora inserita. In caso esista già vado a riempire le tabelle auto e detentore, inserendo il nome della società già esistente e ritorno un 1 per far comparire a schermo un alert che avverta l’utente che il nome della società da lui inserito esiste già nella tabella, mentre se non esiste eseguo anche la query per riempire la tabella della società.  Query controllo nome società:  //Query che permettono l'estrapolazione dei nomi delle società,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtNameSociety = $this->conn->prepare ("SELECT 1 FROM societa where Nome\_societa = ?");  $stmtNameSociety->bind\_param("s", $societyName);  Società non presente nel db:  //Controllo se il nome della società é già presente nel databse.  //Se non é presente eseguo le query di inseriemnto nelle tre tabelle.  if ($stmtNameSociety->fetch() == null) {  //Query utilizzata per l'inserimento dei dati relativi alla società all'interno della tabella corretta.  $stmtSociety = $this->conn->prepare("INSERT INTO societa VALUES (?,?,?,?,?,?,?,?)");  $stmtSociety->bind\_param("sissssis", $societyName, $personInChargeTelephone, $personInChargeName,  $personInChargeSurname, $societyAddress, $societyAddressNumber, $societyPostalNumber, $societyCity);  //Eseguo la query.  if(!$stmtSociety->execute()){  echo "La query di inserimento per la tabella società  presente nella funzione saveNewUser() non funziona.";  }  //Query utilizzata per l'inserimento dei dati del detentore all'interno della tabella corretta.  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?)");  $stmtHolder->bind\_param("sssssiss", $emailHolder, $holderName, $holderSurname, $holderAddress,  $holderAddressNumber, $holderPostalNumber, $holderCity, $societyName);  //Eseguo la query.  if(!$stmtHolder->execute()){  echo "La query di inserimento per la tabella detentore  presente nella funzione saveNewUser() non funziona (con dati relativi alla società).";  }  //Query utilizzata per l'inserimento dei dati dell'auto all'interno della tabella corretta.  $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 per la tabella auto  presente nella funzione saveNewUser() non funziona (con dati relativi alla società).";  }  Società già presente nel db:  //Se la società é già presente eseguo soltanto due query di inseriemtno.  }else {  //Stabilisco una nuova connessione con mysqli, perché ho troppe query  //per un unica connessione.  $this->newConnection();  //Query utilizzata per l'inserimento dei dati del detentore all'interno della tabella corretta.  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?)");  $stmtHolder->bind\_param("sssssiss", $emailHolder, $holderName, $holderSurname, $holderAddress,  $holderAddressNumber, $holderPostalNumber, $holderCity, $societyName);  //Eseguo la query.  if(!$stmtHolder->execute()){  echo "La query di inserimento per la tabella detentore  presente nella funzione saveNewUser() non funziona (con dati relativi alla società).";  }  //Query utilizzata per l'inserimento dei dati dell'auto all'interno della tabella corretta.  $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 per la tabella auto  presente nella funzione saveNewUser() non funziona (con dati relativi alla società).";  }  return 1;  }  Una volta terminato questo controllo ho implementato lo stesso controllo per l’utente e il veicolo già registrati, per eseguire anche questi controlli dovrei inserire nel codice molte più ridondanze di quante ce ne siano già, per ovviare a questa problematica ho spostato tutti i vari controlli in tre funzione a parte. Le funzioni sono pressoché identiche, queste ultime controllano che la chiave primaria delle rispettive tabelle sia già presente o meno.  Controllo società:  /\*\*  \* Funzione che controlla se la società inserita é già presente nella tabella.  \* @return int Ritorna il risultato della query, 1 se é già presente null altrimenti.  \*/  function checkSocietyDuplicate($societyName){  //Query che permettono l'estrapolazione dei nomi delle società,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtNameSociety = $this->conn->prepare ("SELECT 1 FROM societa where Nome\_societa = ?");  $stmtNameSociety->bind\_param("s", $societyName);  //Eseguo la query.  if($stmtNameSociety->execute()){  //Ritorno il risultato della select.  return $stmtNameSociety->fetch();  }else {  echo "La query di selezione dei nomi delle società  presente nella funzione checkSocietyDuplicate() non funziona.";  }  }  Controllo detentore:  /\*\*  \* Funzione che controlla se l'email del detentore inserito é già presente nella tabella.  \* @return int Ritorna il risultato della query, 1 se é già presente null altrimenti.  \*/  function checkHolderDuplicate($emailHolder){  //Query che permettono l'estrapolazione delle email del detentore,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtEmailHolder = $this->conn->prepare ("SELECT 1 FROM detentore where Email = ?");  $stmtEmailHolder->bind\_param("s", $emailHolder);  //Eseguo la query.  if($stmtEmailHolder->execute()){  //Ritorno il risultato della select.  return $stmtEmailHolder->fetch();  }else {  echo "La query di selezione delle email dei detentori  presente nella funzione checkHolderDuplicate() non funziona.";  }  }  Controllo auto:  /\*\*  \* Funzione che controlla se il numero di targa inserito é già presente nella tabella.  \* @return int Ritorna il risultato della query, 1 se é già presente null altrimenti.  \*/  function checkCarDuplicate($plateNumber){  //Query che permettono l'estrapolazione delle targhe delle auto,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtPlateNumber = $this->conn->prepare ("SELECT 1 FROM auto where N\_targa = ?");  $stmtPlateNumber->bind\_param("s", $plateNumber);  //Eseguo la query.  if($stmtPlateNumber->execute()){  //Ritorno il risultato della select.  return $stmtPlateNumber->fetch();  }else {  echo "La query di selezione delle targhe delle auto  presente nella funzione checkCarDuplicate() non funziona.";  }  }  Alert in caso di duplicati:  if ($check == 1) {  echo "<script type='text/JavaScript'>  alert('Valori già presente nel database, nuovo campo aggiunto nelle altre tabelle.');  </script>";  }  Test:  Se inserisco dei duplicati i campi vengono inseriti nelle rispettive tabelle nel modo corretto e a schermo appare l’alert, ho riscontrato alcuni bug con questi controlli che cercherò di risolvere la prossima lezione. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Mentre implementavo il controllo sul nome della società già esistente nella tabella, come spiegato inizialmente, ho riscontrato un problema insolito. Ricercando un po’ su Internet ho constatato che il problema stava nella moltitudine di query che eseguivo con una sola connessione al database mysqli, per risolvere questo problema ho semplicemente creato una funzione, rimuovendo il metodo costruttore praticamente inutile con questa nuova funzione, che mi permetta di ricreare la connessione mysqli ogni volta che la richiamo.  Funzione per nuove connessioni al db:  /\*\*  \* Funzione che istanzia una nuova connessione mysqli.  \*/  function newConnection() {  // Creo la connessione  $this->conn = new mysqli($this->servername, $this->username, $this->password, $this->dbname, $this->port);  // Controllo la connessione  if ($this->conn->connect\_error) {  die("Connection failed: " . $this->conn->connect\_error);  }  } |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In orario rispetto alla pianificazione iniziale. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Correggere dei bug riscontrati in questa lezione; far apparire un’alert contenente il costo totale delle chiavi da lui ordinate e dove pagare; cominciare con l’implementazione del codice delle pagine amministrative. |