Diaro finale

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 08.01.2019 |

|  |
| --- |
| Lavori svolti |
| Durante la mattina ho eseguito la presentazione del primo progetto di quest’anno, in seguito ho scelto il secondo progetto. Una volta letto il QDC ho posto delle domande su alcuni dubbi che mi erano sorti. Durante il pomeriggio ho cominciato la fase di analisi del progetto e della documentazione. |

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

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Non ho ancora pianificato un Gantt preventivo del mio progetto. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Terminare il capitolo dell’analisi della documentazione. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 09.01.2019 |

|  |
| --- |
| Lavori svolti |
| Nel corso della mattinata mi sono occupato di terminare il primo capitolo della documentazione, cioè il capitolo riguardante l’analisi del progetto. In seguito mi sono occupato della creazione del gantt consuntivo.    Durante le ultime due ore della giornata ho cominciato a creare lo schema UseCase. |

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

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Terminare lo UseCase e cominciare a progettare le pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 15.01.2019 |

|  |
| --- |
| Lavori svolti |
| Come prima cosa, questa mattina, mi sono occupato di terminare lo UseCase del mio progetto. Il risultato è il seguente:    In seguito ho cominciato e terminato la progettazione di tute le pagine web che conterrà il mio sito web e i risultati sono seguenti:  Index    All’suo interno l’amministratore può raggiungere la finestra di login, mentre un utente può, dopo aver confermato che non sia un robot, accedere alla pagina per la richiesta di una nuova chiave, oppure accedere ad una pagina per contattare l’amministratore.  Form di richiesta    All’interno di questo form l’utente, innanzitutto, “conferma” se fa parte di un’azienda oppure no, in caso facesse parte di una società allora appariranno ulteriori campi da compilare riguardanti l’azienda. In caso contrario il primo campo (non editabile da parte dell’utente) mostrerà il numero di chiavi che l’utente desidera, sotto di esso saranno presenti tutti i campi da compilare e alla fine l’utente può decidere se ordinare altre chiavi oppure no. In caso di annullamento della richiesta può tornare alla pagina principale tramite i menu posto in alto a destra.  Contact    All’interno di questa pagina l’utente può contattare l’amministratore tramite un email.  Login    All’interno di questa pagina l’amministratore può eseguire il login per accedere alla parte amministrativa del progetto.  Pagamenti    All’interno della pagina, accessibile solo dall’amministratore del sistema, si possono gestire tutte le richieste di chiavi in entrata.  Statistiche    All’interno della pagina, accessibile solo dall’amministratore del sistema, quest’ultimo può visualizzare delle statistiche sulla frequenza con cui le auto passano per questa strada, può visualizzare dei periodi precisi grazie a dei filtri temporali.  Lista utenti    All’interno della pagina, accessibile solo dall’amministratore del sistema, quest’ultimo può visualizzare la lista degli utenti che posseggono delle chiavi e può decidere se visualizzarli con le relative auto o meno.  Per la maggior parte delle pagine web ho trovato dei template parzialmente fatti. |

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

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Cominciare e completare la creazione dello schema E-R e cominciare con l’implementazione delle parti grafiche delle varie pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 16.01.2019 |

|  |
| --- |
| Lavori svolti |
| Non sono stato presente a questa lezione di progetti per motivi di salute. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Non sono stato presente a questa lezione di progetti per motivi di salute. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Non sono stato presente a questa lezione di progetti per motivi di salute. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Non sono stato presente a questa lezione di progetti per motivi di salute. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 22.01.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti ho trascorso tutta la mattina per la creazione su carta dello schema E-R del database riguardante il mio progetto e, una volta confermata la sua validità con il mio responsabile, l’ho riprodotto sul mio pc tramite il software di Visio. Il risultato è il seguente:    Una volta completato ho cominciato e concluso la sua creazione tramite codice SQL sul software HeidiSQL, il codice da me prodotto per la realizzazione di questo database è il seguente:    La tabella soprastante rappresenta le chiavi che verranno consegnate agli utenti, al suo interno è presente un solo capo con il numero identificativo della chiave.    La tabella soprastante rappresenta le auto a cui si vuole “collegare” ogni determinata chiave, al suo interno sono presenti 6 campi. I primi 5 rappresentano le informazioni dell’auto, mentre l’ultimo campo rappresenta la chiave esterna che “collega” la tabella *Detentore* alla tabella *Auto*.    La tabella soprastante rappresenta la tabella che “congiunge” la tabella *Chiave* con la tabella *Auto*, al suo interno sono presenti le date di ricezione e di consegna di ogni chiave e le chiavi esterne delle due tabelle.    La tabella soprastante rappresenta i detentori che eseguono la richiesta per una o più chiavi, al suo interno sono presenti i campi che conterranno le informazioni personali del detentore, in più è presente la chiave esterna che “collega” la tabella *Società* con la tabella *Detentore*.    La tabella soprastante rappresenta la società che “possiede” un detentore, al suo interno sono presenti le informazioni della società stessa e del responsabile di quest’ultima. Nelle ultime ore di questa giornata ho continuato il capitolo due della documentazione e cercato alcuni template da utilizzare per l’implementazione delle varie pagina web. Come ultima cosa ho modificato il design della pagina contenente la richiesta delle chiavi aggiungendo alcuni campi per l’indirizzo ed il cognome del responsabile dell’azienda. |

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

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Cominciare con l’implementazione delle parti grafiche delle varie pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 23.01.2019 |

|  |
| --- |
| Lavori svolti |
| Nel corso della mattinata di questa lezione di progetti mi sono occupato della creazione dell’interfaccia grafica della prima pagina web (index.html). Nelle ultime due ore della giornata mi sono occupato di portare avanti il secondo capitolo della documentazione, mentre svolgevo questo lavoro ho notato alcune lacune ed errori nel database e nel design delle pagine web (parte amministrativa). Innanzitutto nel database ho tolto tutti gli “*on delete cascade*” tranne nella tabella che si pone nel mezzo della relazione molti a molti, questo perché nelle altre tabelle si andavano ad eliminare dati che non dovevano essere eliminati. Sempre nella tabella *appartiene* ho modificato la sua chiave primaria, al posto di lasciare le due chiavi esterne come chiave primaria gli ho messo un semplice *Id*, questo perché se un’auto viene rimossa nel database anche la relativa chiave si sarebbe dovuta rimuovere. L’ultima modifica apportata nel database è stata l’aggiunta della tabella “*amministratore*”, al suo interno vado a salvare i dati di login dell’amministratore (username e password). Il database aggiornato è il seguente:    Per quanto riguarda le modifiche dei design delle pagine web della parte amministrativa ho constatato che dovevo aggiungere un’ulteriore pagina per la gestione della restituzione delle pagine, infatti al suo interno saranno presenti tutti gli utenti con la relativa chiave e, quando vorranno restituirla, l’amministratore li seleziona dalla pagina e li va ad eliminare dal database in modo tale da poter riutilizzare le chiavi restituite. I design della nuova pagina web è il seguente: |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante questa lezione di progetti ho riscontrato una problematica riguardante la pagina principale, infatti, al suo interno volevo inserire una sicurezza che permetta di controllare che l’utente che vuole accedere al formulario per richiedere una o più chiavi non sia un robot. Questo perché ci sarebbe il rischio che, tramite un programma/script, “bombardino” il sito di richieste fino a farlo andare in crash. Per fare questo controllo esistono script già fatti dall’ente chiamato reCAPTCHA, soltanto che mi compare l’errore seguente: |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con l’implementazione delle parti grafiche delle varie pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 28.01.2019 |

|  |
| --- |
| Lavori svolti |
| Nella mattinata di questa giornata di progetti ho continuato con la parte grafica delle varie pagine web, nella seconda ora ho ricevuto le valutazioni del mio primo progetto effettuato nel corso del primo semestre, mentre per il resto della mattina ho completato la grafica delle pagine di login e contact e ho cominciato la pagina del formulario da compilare per richiedere una chiave. |

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

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con l’implementazione delle parti grafiche delle varie pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 29.01.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti ho concluso la parte grafica della pagina web dedicata al formulario. Come prima cosa ho creato un contenitore chiamato *jumbutron*, in modo tale da non avere l’immagine di sfondo anche dietro gli input e le scritte, dato che causava dei problemi con la visibilità di queste ultime. Per suddividere bene tutti gli input questo contenitore suddivide il tutto in righe e colonne, come se ci fosse una tabella “invisibile”. I risultati sono i seguenti:  Dati relativi alla società    Per quanto riguarda i dati della società se l’utente dice di non farne parte o di non averne una e clicca sul radio button *No* tutti quegli input diventano disabilitati e nascosti. Per fare ciò ho realizzato la seguente funzione in JavaScript:  function hide(choice) {  var hide = document.getElementById('society');  var required = document.getElementsByClassName('notRequired');  if (choice.value == "0") {  for (var i = 0; i < required.length; i++) {  required[i].disabled=true;  }  hide.hidden=true;  }else{  for (var i = 0; i < required.length; i++) {  required[i].disabled=false;  }  hide.hidden=false;  }  }  Dato relativi al numero di chiavi    Dati relativi al detentore della chiave    Nel corso della creazione ho deciso di tenere il menu presente in altro a destra anche in questa pagina web, in modo tale da facilitare a tutti gli utenti la navigazione tra le pagine. |

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

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con l’implementazione delle parti grafiche delle varie pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 30.01.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti ho cominciato con l’implementazione della parte grafica delle pagine amministrative del mio progetto. Inizialmente ho cercato su internet un template da poter usare come base, in modo tale da non doverle creare tutte da zero. Dopo un po’ di ricerche ho scelto il seguente template: <https://designrevision.com/demo/shards-dashboard-lite/>. Nelle ultime ore ho quasi terminato la parte grafica della prima pagina web, quella della gestione dei pagamenti, al suo interno mi manca l’aggiunta della tabella che mostrerà le varie richieste ed il bottone submit. Il risultato grafico al momento è il seguente: |

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

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono un po’ in ritardo con l’implementazione della parte grafica delle pagine web, infatti mi manca ancora la parte grafica di tre pagine web. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Terminare l’implementazione delle parti grafiche delle varie pagine web. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 04.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti ho concluso la parte grafica di tutte le pagine web che andranno a comporre il mio sito web.  Gestione pagamenti:    Gestione restituzione chiavi:    Gestione statistiche:    Gestione storico con info relative alle auto:    Gestione storico senza relative info delle vetture:    All’interno di questa pagina, un po’ come ho fatto per la pagina *formulary.php*, ho creato uno script che permetta di nascondere le colonne relative alle informazioni delle auto presenti nella tabella. Lo script è pressoché uguale:  Come prima cosa ho assegnato una classe uguale ad ogni colonna che volevo nascondere.  <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">  <thead>  <tr>  <th>Nome</th>  <th>Cognome</th>  <th class="notRequired">N° Targa</th>  <th class="notRequired">Marca</th>  <th class="notRequired">Colore</th>  <th class="notRequired">Modello</th>  <th class="notRequired">Tipo</th>  </tr>  </thead>  <tbody>  <tr>  <td></td>  <td></td>  <td class="notRequired"></td>  <td class="notRequired"></td>  <td class="notRequired"></td>  <td class="notRequired"></td>  <td class="notRequired"></td>  </tr>  </tbody>  </table>  In seguito ho ripreso il vecchio script utilizzato nell’altra pagina php è l’ho leggermente modificato togliendo la parte che disabilitava gli input, dato che io ho solo bisogno di nascondere delle colonne.  function hide(choice) {  var required = document.getElementsByClassName('notRequired');  if (choice.value == "0") {  for (var i = 0; i < required.length; i++) {  required[i].hidden=true;  }  }else{  for (var i = 0; i < required.length; i++) {  required[i].hidden=false;  }  }  }  Tramite un ciclo vado a nascondere ogni elemento. |

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

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono un po’ in ritardo con l’implementazione della parte grafica delle pagine web, ma in questa lezione sono riuscito a terminarle. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Cominciare con il salvataggio dei dati immessi nella pagina di registrazione all’interno del databse. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 05.02.2019 |

|  |
| --- |
| Lavori svolti |
| Come prima cosa mi sono occupato della creazione della tabella *amministratore* all’interno del database:    Per la password ho scelto un tipo di dato char lungo 64 perché per codificare la password utilizzerò sha256 che mi ritorna una stringa lunga 64 caratteri.  In seguito ho creato una pagina per permettere la registrazione di un utente amministratore, questa pagina non è raggiungibile tramite il menu, ma si deve raggiungere solo tramite link in modo tale da non permettere la registrazione a tutti quanti. Il risultato della pagina è il seguente:    Una volta creata questa pagina di registrazione ho creato la classe che si occupa della connessione al database (*db\_connection.php*) con le relative variabili.  Connessione al database:  //Connessione al db mysql da localhost.  private $servername = "localhost";  private $username = "root";  private $password = "";  private $dbname = "gestione\_traffico\_veicolare";  private $port = 3306;  // Creo la variabile connessione  public $conn;  /\*\*  \* Metodo costruttore in cui creo la connessione con il database.  \*/  function \_\_construct() {  // 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: " . $this->conn->connect\_error);  }  }  Una volta stabilita la connessione con esso ho creato la funzione che inserisce le credenziali dell’amministratore immesse nella pagina di registrazione (*register.php*), al suo interno ho implementato una semplice query di tipo *insert*.  Immissione delle credenziali all’interno della tabella amministratore:  /\*\*  \* Questa funzione si occupa dell'inserimento dei dati dell'amministratore all'interno del database tramite una query.  \* @param username L'username dell'amministratore.  \* @param password La password dell'amministratore.  \*/  function setAmministratorCredential($username, $password) {  //Codifico la password ricevuta tramite l'algoritmo sha256.  $passwordEncoded = hash("sha256", $password);  $conn = self::\_\_construct();  //Query che permettono l'inserimento delle credenziali dell'amministratore, utilizzando un prepare statement per evitare  //delle SQLInjection.  $stmt = $this->conn->prepare ("INSERT INTO amministratore (Username, Password) VALUES (?,?)");  $stmt->bind\_param("ss", $username, $passwordEncoded);  //Eseguo la query.  if(!$stmt->execute()){  echo "La query di inserimento delle credenziali dell'amministratore presente nella funzione setAmministratorCredential() non funziona.";  }  }  Test:  Una volta premuto il bottone per permettere la registrazione, andando sul database, si può notare che sono presenti all’interno della tabella e la password è codificata nel modo corretto.    In seguito ho creato un’ulteriore funzione che permetta il controllo delle credenziali immesse nella pagina di login, innanzitutto ricodifico la password immessa nella pagina, controllo se l’username e la password esistono nella tabella, se il risultato della query è diverso da null ritorno true altrimenti ritorno false.  Funzione di controllo delle credenziali di amministrazione:  /\*\*  \* Questa funzione si occupa dell'estrapolazione dei dati dell'amministratore dal database tramite una query,  \* in seguito controllo se effettivamente esistono al suo interno.  \* @param username L'username dell'amministratore.  \* @param password La password dell'amministratore.  \* @return boolean Ritorno un valore booleano a dipendenza se le credenziali immesse sono corrette o meno.  \*/  function getAmministratorCredential($username, $password) {  //Codifico la password ricevuta tramite l'algoritmo sha256.  $passwordEncoded = hash("sha256", $password);  $containerData = array();  $conn = self::\_\_construct();  //Query che permettono l'estrapolazione delle credenziali dell'amministratore,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmt = $this->conn->prepare ("SELECT 1 FROM amministratore where Username = ? && Password = ?");  $stmt->bind\_param("ss", $username, $passwordEncoded);    //Eseguo la query.  if($stmt->execute()){  //Controllo se le credenziali sono presenti nel databse.  //Se non sono presenti allora le credenziali sono errate, altrimenti viene eseguito il login.  if ($stmt->fetch() != null) {  return true;  }else {  return false;  }  }else {  echo "La query di selezione delle credenziali dell'amministratore presente nella funzione getAmministratorCredential() non funziona.";  }  }  Per quanto riguarda le pagine di registrazione e di login ho aggiunto una piccola porzione di codice php che mi permetta di utilizzare le funzioni scritte nella classe presente nel file *db\_connection.php*.  Richiamo delle funzioni dalla pagina di registrazione:  <?php  //Eseguo il require dei file esterni necessari.  require\_once('db\_connection.php');  //Creao gli oggetti delle classi necessarie;  $conn = new connection();  //Controllo se é stato eseguito il POST  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  $conn->setAmministratorCredential($\_POST["inputUsername"], $\_POST["inputPassword"]);  }  ?>  Test:  Il nuovo amministratore è stato registrato nel modo corretto, come si può vedere dall’immagine posta nel test precedente.  Richiamo delle funzioni dalla pagina di login:  <?php  $result = null;  //Eseguo il require dei file esterni necessari.  require\_once('db\_connection.php');  //Creao gli oggetti delle classi necessarie;  $conn = new connection();  //Controllo se é stato eseguito il POST  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  $result = $conn->getAmministratorCredential($\_POST["inputUsername"], $\_POST["inputPassword"]);  //Controllo se la funzione che controlla se le credenziali immesse per autenticarsi  //come amministratore sono corrette(ritorno true) o se sono errate(ritorno false).  //Se le credenziali sono corrette vengo reindirizzato alle pagine amministrative,  //altrimenti viene stampato un alert a schermo.  if ($result) {  header("Location: ../../sitoAdmin/index.php");  }else {  echo "<script type='text/JavaScript'>alert('Username o password incorretti');</script>";  }  }  ?>  Test:  Se le credenziali sono corrette vieni reindirizzato correttamente nelle pagine di amministrazione, mentre se le credenziali sono errate viene visualizzato l’alert e vieni reindirizzato alla stessa pagina di login.    Tutto ciò che riguarda la navigazione tra le pagine web è stata eseguito nel modo corretto.  In seguito ho creato un’ulteriore funzione che servirà per il salvataggio dei dati immessi nella pagina *formulary.php* all’interno delle relative tabelle. Come prima cosa passo alla funzione, oltre a tutti i parametri presenti nella pagina web, un valore numerico (0 o 1) che utilizzo per controllare se l’utente ha immesso anche i dati relativi alla società. Se il valore è pari a 1 allora creo delle query che vadano a riempire tutte le tabelle del database, mentre se è 0 eseguo le query per riempire le tabelle tranne quella della società.  Funzione che riempie le tabelle presenti nel database:  function saveNewUser($society, $societyName, $personInChargeName,$personInChargeSurname, $personInChargeTelephone, $societyAddress,$societyAddressNumber, $societyPostalNumber, $societyCity, $holderName,$holderSurname, $holderAddress, $holderAddressNumber, $holderPostalNumber,$holderCity, $carBrand, $carColor, $carModel, $carType, $plateNumber, $emailHolder) {  $null = null;  $conn = self::\_\_construct();  $this->keyNumber++;  if ($society == 1) {  $stmtKey = $this->conn->prepare("INSERT INTO chiave (N\_chiave) VALUES (?)");  $stmtKey->bind\_param("i", $this->keyNumber);  //Eseguo le query.  if(!$stmtKey->execute()){  echo "La query di inserimento per la tabella chiavi  presente nella funzione saveNewUser() non funziona.";  }  $stmtSociety = $this->conn->prepare("INSERT INTO societa VALUES (?,?,?,?,?,?,?,?)");  $stmtSociety->bind\_param("sissssis", $societyName, $personInChargeTelephone, $personInChargeName, $personInChargeSurname, $societyAddress, $societyAddressNumber, $societyPostalNumber, $societyCity);  //Eseguo le query.  if(!$stmtSociety->execute()){  echo "La query di inserimento per la tabella società  presente nella funzione saveNewUser() non funziona.";  }  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?)");  $stmtHolder->bind\_param("sssssiss", $emailHolder, $holderName, $holderSurname, $holderAddress, $holderAddressNumber, $holderPostalNumber, $holderCity, $societyName);  //Eseguo le query.  if(!$stmtHolder->execute()){  print\_r($stmtHolder);  echo "La query di inserimento per la tabella detentore  presente nella funzione saveNewUser() non funziona.";  }  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo le query.  if(!$stmtCar->execute()){  echo "La query di inserimento per la tabella auto  presente nella funzione saveNewUser() non funziona.";  }  }else {  $stmtKey = $this->conn->prepare("INSERT INTO chiave (N\_chiave) VALUES (?)");  $stmtKey->bind\_param("i", $this->keyNumber);  //Eseguo le query.  if(!$stmtKey->execute()){  echo "La query di inserimento per la tabella chiavi  presente nella funzione saveNewUser() non funziona.";  }  $stmtHolder = $this->conn->prepare("INSERT INTO detentore VALUES (?,?,?,?,?,?,?,?)");  $stmtHolder->bind\_param("sssssiss", $emailHolder, $holderName, $holderSurname, $holderAddress, $holderAddressNumber, $holderPostalNumber, $holderCity, $null);  //Eseguo le query.  if(!$stmtHolder->execute()){  echo "La query di inserimento per la tabella detentore  presente nella funzione saveNewUser() non funziona.";  }  $stmtCar = $this->conn->prepare("INSERT INTO auto VALUES (?,?,?,?,?,?)");  $stmtCar->bind\_param("ssssss", $plateNumber, $carBrand, $carModel, $carType, $carColor, $emailHolder);  //Eseguo le query.  if(!$stmtCar->execute()){  echo "La query di inserimento per la tabella auto  presente nella funzione saveNewUser() non funziona.";  }  }  }  Durante la creazione di questa funzione mi sono accorto che dovevo cambiare le chiavi primarie della tabella *detentore* e della tabella *società*, inizialmente come chiavi primarie avevo dei semplici id auto incrementali, ora per la tabella *detentore* ho aggiunto un campo email come chiave primaria (aggiunto anche come input nella pagina web *formulary.php*), mentre per la tabella *società* ho utilizzato il nome della società come chiave primaria. Ho eseguito questo cambiamento perché mi è più facile lavorare con le chiavi esterne tramite php.  Tabella detentore:    Tabella società:    Tabella auto:    Schema E-R aggiornato: |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante questa lezione di progetti ho riscontrato un problema con il riempimento della tabella *chiave*, dato che al suo interno è presente un solo campo ed è anche la chiave primaria, ho creato un valore che si dovrebbe incrementare di uno ad ogni richiamo del metodo, cioè ad ogni richiesta di una nuova chiave. Il problema è che questo attributo non si incrementa come dovrebbe. Non sono riuscito a trovare una soluzione entro la fine di questa lezione di progetti, ci penserò la prossima lezione. Un’ulteriore problema che, mentre testavo le varie query, mi è venuto in mente riguarda la tabella della società, se quest’ultima esiste già nel database allora devo eseguire un controllo della tabella prima che essa venga inserita. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono un po’ in ritardo rispetto alla pianificazione da Gantt, grazie a questa lezione ero riuscito a tornare nelle tempistiche corrette ma, a causa di questi problemi che riguardano il riempimento delle tabelle tramite le query, non sono riuscito a terminare il salvataggio dei dati nel database che era previsto per oggi. Ma la maggior parte del ritardo, come detto in precedenza, sono riuscito a recuperarlo. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Concludere il salvataggio dei dati nel database. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 06.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti mi sono occupato solamente della risoluzione del problema che avevo riscontrato l’ultima lezione di progetti, cioè il riempimento della tabella *chiave*. Dopo svariati tentativi ho optato per una soluzione piuttosto semplice e banale, quest’ultima consiste nell’eseguire prima una *select* che mi permetta di estrarre dalla tabella la chiave con l’identificativo maggiore, in seguito prendo quell’id e lo incremento di 1 ed infine lo vado, tramite un’*insert*, a reinserire in una nuova riga, in modo tale da avere sempre un identificativo diverso e sempre maggiore.  Query:  //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).";  }  //Query che inserisce una nuova riga nella tabella "chiave",  //questa riga contiene l'identificativo più alto incrementato di 1.  $stmtKeyInsert = $this->conn->prepare("INSERT INTO chiave VALUES (?)");  $stmtKeyInsert->bind\_param("i", $temp);  //Eseguo la query.  if(!$stmtKeyInsert->execute()){  echo "La query di inserimento per la tabella chiave  presente nella funzione saveNewUser() non funziona (con dati relativi alla società).";  }  Test:    Il problema di questa soluzione è che all’interno della tabella chiave deve essere sempre presente almeno un valore.  Mentre negli ultimi minuti della lezione ho scritto il diario ed ho curato un po’ meglio la stesura del codice con dei commenti. |

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

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono tornato in orario con la mia pianificazione iniziale |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Eseguire il controllo se una società esiste già nella tabella; 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. |

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. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 12.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante la mattina mi sono occupato di fare il design della pagina di registrazione che avevo dimenticato di fare.  Design register.php    In seguito ho avuto una conversazione con il mandante del mio progetto per discutere di alcune problematiche che ho riscontrato nelle due settimane precedenti, queste ultime riguardavano: la gestione della richiesta di molteplici chiavi (risolvibile tramite js e nomi degli input auto incrementali); la gestione del “*non sono un robot*” (da trovare ancora una soluzione); il checkbox *remember me* nella pagina di login(lasciare per un secondo momento); dei controlli sulla pagina register.php per evitare che chiunque si possa registrare alla parte amministrativa (gestire l’accesso tramite una pagina web nella parte amministrativa.  Inizialmente ho deciso di sistemare tutti i controlli per la pagina di registrazione, pensando che entro la fine della mattina l’avrei finito, ma non è stato così. Come prima cosa ho modificato nuovamente dei campi presenti nel database: nella tabella *detentore* ho aggiunto un campo booleano che servirà come flag per controllare se ha effettuato il pagamento o meno; nella tabella *amministratore* ho aggiunto, anche li, un campo booleano che userò come flag per abilitare o bloccare tutti i vari utenti registrati per la parte amministrativa.  Tabella detentore:    Tabella amministratore:    Schema E-R aggiornato:    Inizialmente ho creato, nella parte amministrativa, una pagina web contenente due tabelle dove si può gestire il blocco o lo sblocco dei relativi account amministrativi.  Pagina registrationConfirm.php:    Funzione che estrapola gli amministratori bloccati:  function getBlockedAdministrator() {  $confirmed = 0;  $containerData = array();  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Query che permettono l'inserimento delle credenziali dell'amministratore con il rispettivo flag  //di conferma, quest'ultimo dovrà essere cambiato dall'amministartore del sito web,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmt = $this->conn->prepare ("SELECT Username FROM amministratore where Confermato = ?");  $stmt->bind\_param("i", $confirmed);  //Eseguo la query.  if(!$stmt->execute()){  echo "La query di inserimento delle credenziali dell'amministratore  presente nella funzione setAmministratorCredential() non funziona (riga:36).";  }else {  $result = $stmt->get\_result();  //Ciclo tutti i "dati" che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  array\_push($containerData, $row);  }  return $containerData;  }  }  Funzione che estrapola gli amministratori sbloccati:  function getUnBlockedAdministrator() {  $confirmed = 1;  $containerData = array();  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Query che permettono l'inserimento delle credenziali dell'amministratore con il rispettivo flag  //di conferma, quest'ultimo dovrà essere cambiato dall'amministartore del sito web,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmt = $this->conn->prepare ("SELECT Username FROM amministratore where Confermato = ?");  $stmt->bind\_param("i", $confirmed);  //Eseguo la query.  if(!$stmt->execute()){  echo "La query di inserimento delle credenziali dell'amministratore  presente nella funzione setAmministratorCredential() non funziona (riga:36).";  }else {  $result = $stmt->get\_result();  //Ciclo tutti i "dati" che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  array\_push($containerData, $row);  }  return $containerData;  }  }  Funzione che permette lo sblocco o il blocco degli account:  /\*\*  \* Questa funzione si occupa dell'inserimento dei dati dell'amministratore  \* all'interno del database tramite una query.  \* @param state L'array contenente tutti gli username selezionati.  \* @param currentState Lo stato attuale del amministrtore (bloccato o sbloccato).  \*/  function changeAdministratorFlag($state, $currentState) {  echo "string";  $unLock = 1;  $lock = 0;  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Ciclo l'array e divido il nome ed il cognome di ogni allievo in un'differente array.  for ($i=0; $i < count($state); $i++) {  //Controllo se l'array contenente gli username degli amministratori selezionati é vuoto oppure no.  if(count($state) != 0) {  //Controllo lo stato del flag se é bloccato (0) o sbloccato (1).  if($currentState == 0) {  //Eseguo la query che modifica lo stato d'accesso.  echo "string";  $stmt = $this->conn->prepare("UPDATE amministratore set Confermato = ? where Username = ?");  $stmt->bind\_param("is", $unLock, $state[$i]);  }else {  //Eseguo la query che modifica lo stato d'accesso.  echo "string2";  $stmt = $this->conn->prepare("UPDATE amministratore set Confermato = ? where Username = ?");  $stmt->bind\_param("is", $lock, $state[$i]);  }  if($stmt->execute()) {  //Richiamo le funzioni che si occupano di stampare a schermo le due tabelle degli amministratori bloccati e non.  $this->getBlockedAdministrator();  $this->getUnBlockedAdministrator();  }else {  echo "La query di di modifica del flag dell'amministratore  presente nella funzione changeAdministratorFlag() non funziona (riga:36).";  }  }else {  echo "Non hai selezionato nessun amministratore";  }  }  }  Test:  Dalla pagina web riesco a modificare con successo il flag degli account amministrativi.  A causa di questo flag che ho aggiunto nella tabella *amministratore* ho dovuto modificare la funzione che salva i dati del nuovo amministratore nella relativa tabella, semplicemente aggiungendo un campo in più nella query  Funzione che salva i dati dei nuovi amministratori:  /\*\*  \* Questa funzione si occupa dell'inserimento dei dati dell'amministratore  \* all'interno del database tramite una query.  \* @param username L'username dell'amministratore.  \* @param password La password dell'amministratore.  \*/  function saveAmministratorCredential($username, $password) {  $confirmed = 0;  //Codifico la password ricevuta tramite l'algoritmo sha256.  $passwordEncoded = hash("sha256", $password);  //Stabilisco una nuova connessione con mysqli, perché ho troppe query  //per un unica connessione.  $this->newConnection();  //Query che permette il controllo dell'inserimento del nuovo username dell'amministratore,  //se esiste già o meno.  $stmtUsernameControl = $this->conn->prepare("SELECT 1 FROM amministratore where Username = ?");  $stmtUsernameControl->bind\_param("s", $username);  $stmtUsernameControl->execute();  if ($stmtUsernameControl->fetch() != null) {  return true;  }else {  //Query che permette l'inserimento delle credenziali dell'amministratore con il rispettivo flag  //di conferma, quest'ultimo dovrà essere cambiato dall'amministartore del sito web,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtNewAdministrator = $this->conn->prepare("INSERT INTO amministratore (Username, Password, Confermato) VALUES (?,?,?)");  $stmtNewAdministrator->bind\_param("ssi", $username, $passwordEncoded, $confirmed);  //Eseguo la query.  if(!$stmtNewAdministrator->execute()){  echo "La query di inserimento delle credenziali dell'amministratore  presente nella funzione setAmministratorCredential() non funziona (riga:36).";  }  return false;  }  }  L’ultima funzione che ho dovuto modificare e, che mi ha causato parecchi problemi, è quella che controlla se l’account con cui si vuole fare il login è sbloccato e che le sue credenziali siano corrette. (sono presenti un po’ di errori nei commenti)  Funzione che permette il login:  /\*\*  \* Questa funzione si occupa dell'estrapolazione dei dati dell'amministratore dal database tramite una query,  \* innanzitutto controllo lo stato del suo flag (account abilitato o no) ed in seguito controllo le sue credenziali.  \* @param username L'username dell'amministratore.  \* @param password La password dell'amministratore.  \* @return int Ritorno una un valore interno a dipendenza del flag e delle credenziali:  \* account bloccato(0); credenziali errate(1); account sbloccato e credenziali corrette(2).  \*/  function getAmministratorCredential($username, $password) {  $check = 1;  //Codifico la password ricevuta tramite l'algoritmo sha256.  $passwordEncoded = hash("sha256", $password);  //Stabilisco una nuova connessione con mysqli, perché ho troppe query  //per un unica connessione.  $this->newConnection();  //Query che permette l'estrapolazione delle credenziali dell'amministratore,  //utilizzando un prepare statement per evitare delle SQLInjection.  $stmtFlag = $this->conn->prepare ("SELECT 1 FROM amministratore where Username = ? AND Password = ?");  $stmtFlag->bind\_param("ss", $username, $passwordEncoded);  //Eseguo la query.  if($stmtFlag->execute()){  //Controllo se le credenziali dell'amministratore sono abilitate  //Se non lo sono allora le non può accedere alla parte amministrativa,  //altrimenti controllo le sue credenziali.  if ($stmtFlag->fetch() == 1) {  //echo $stmtFlag->fetch();  //Query che permette l'estrapolazione dello stato del flag dell'amministratore,  //utilizzando un prepared statement per evitare delle SQLInjection.  $this->newConnection();  $stmtCredential = $this->conn->prepare("SELECT 1 FROM amministratore where Username = ? AND Password = ? AND Confermato = ?");  $stmtCredential->bind\_param("ssi", $username, $passwordEncoded, $check);  //Eseguo la query.  if($stmtCredential->execute()){  //Controllo se le credenziali sono presenti nel database.  //Se non sono presenti allora le credenziali sono errate, altrimenti viene eseguito il login.  if ($stmtCredential->fetch() != 0) {  return 2;  }else {  return 0;  }  } else {  echo "La query di selezione delle credenziali dell'amministratore  presente nella funzione getAmministratorCredential() non funziona (riga:72).";  }  }else {  return 1;  }  }else {  echo "La query di selezione del flag dell'amministratore  presente nella funzione getAmministratorCredential() non funziona (riga:62).";  }  }  Controllo del valore di ritorno nella pagina login.php:  if ($result == 0) {  echo "<script type='text/JavaScript'>alert('Account bloccato.');</script>";  }elseif ($result == 1) {  echo "<script type='text/JavaScript'>alert('Username o password incorretti');</script>";  }else {  header("Location: ../../sitoAdmin/index.php");  }  Controllo dell’username inserito per registrarsi nella pagina register.php:  if ($result) {  echo "<script type='text/JavaScript'>alert('Username già presente');</script>";  }else {  echo "<script type='text/JavaScript'>  alert('Nuovo utente amministratore registrato, aspettare che amministratore del sistema lo abiliti');  </script>";  }  Test pagina di registrazione:  Username già presente:    Username nuovo:  Test pagina di login:  Credenziali errate:    Account bloccato:    Account sbloccato e credenziali correte:  Vieni portato all’index della parte amministrativa. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante questa lezione di progetti ho riscontrato molti problemi nella parte di gestione del controllo delle credenziali e del controllo dello stato del flag dell’account. Si sono presentati errori che non capivo e che cambiando valori o query in maniera quasi casuale si mettevano a posto o ne creavano di ulteriori. Ma alla fine sono riuscito a risolvere tutti i problemi. |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| 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. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 13.02.2019 |

|  |
| --- |
| Lavori svolti |
| Assente dalla lezione. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Assente dalla lezione. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Assente dalla lezione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Assente dalla lezione. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 18.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante queste due ore di progetti ho portato avanti la documentazione terminando il capitolo due di essa. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante questa lezione di progetti ho riscontrato un piccolo problema riguardante l’abstract del mio progetto, non so come strutturarlo al meglio per far sì che contenga i punti richiesti dalla documentazione fornitaci dal cantone. |

|  |
| --- |
| 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. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 19.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti mi sono occupato dell’implementazione del bottone posto in basso a sinistra nella pagina *formulary.php*. Esso, una volta cliccato, mostra una copia esatta degli input che il detentore della chiave deve riempire, in modo tale che un utente ha la possibilità di ordinare più chiavi senza dover riaggiornare la pagina. Prima di fare ciò ho implementato, sempre tramite l’onclick del bottone utilizzato per aggiungere una nuova chiave, il fatto che si veda l’incremento del numero di chiavi nella pagina contenente il formulario di richiesta per queste ultime. Per fare ciò ho semplicemente assegnato il richiamo di una funzione in JavaScript ad ogni click del bottone, prendo il valore dell’input che mi interessa (default=1) e la incremento ad ogni richiamo.  Input contenente il numero di chiavi (grafica)    Input contenente il numero di chiavi (codice)  <div class="nKeys">  <div class="form-group col-md-auto">  <label for="nameHolder">Numero di chiavi</label>  <input type="number" class="form-control nKeys" id="inputNumberKeys" name="inputNumberKeys" value="1" data-numKeys="1" readonly>  </div>  </div>  Bottone aggiungi chiavi (grafica)    Bottone aggiungi chiavi (codice)  <div class="form-row">  <div class="form-group col-md-auto">  <label for="addKeys">Aggiungi chiavi</label>  <br>  <button type="button" class="btn btn-warning btn-circle" onclick="incrementKeys()"><b class="add">+</b></button>  </div>  </div>  Funzione incremento numero di chiavi  var value = parseInt(document.getElementById('inputNumberKeys').value, 10);  value = isNaN(value) ? 1 : value;  value++;  document.getElementById('inputNumberKeys').value = value;  Test  Una volta che clicco il bottone il numero di chiavi si incrementa di uno con successo.  In seguito ho implementato l’alert che permette di visualizzare il numero di chiavi selezionate, la spesa totale e il conto in cui effettuare il pagamento.  Alert  echo "  <script type='text/JavaScript'>  var keyCost = 200;  var total = ". $\_SESSION["inputNumberKeys"] ." \* keyCost;  alert('Hai ordinato ' + ". $\_SESSION["inputNumberKeys"] ." + ' chiave/i, il costo totale é di '  + total + 'CHF. Pagare il seguente importo al conto XXX-XXX-XXX-XXX');  </script>";  Test    Una volta terminato anche questo sono passato all’implementazione descritta inizialmente. In principio, tramite l’aiuto del mio compagno di classe Dominelli, il codice per far apparire una copia identica degli input del detentore è stato pensato come una concatenazione a quelli già presenti. Inizialmente ho rinchiuso tutti gli input in un div che contiene tutti i fieldset che contengono a loro volta i vari input, una volta che l’utente cliccava il bottone per aggiungere una chiave partiva la funzione assegnatagli nell’evento onclick, in pratica andavo a concatenare al div contenitore una copia del fieldset e del suo contenuto. Il problema è che i nomi degli input non possono essere uguali, per differenziarli tra di loro ci ho concatenato il valore del numero di chiavi.  Codice per eseguire la copia e stamparla a schermo degli input  document.getElementById('detentori').innerHTML += '<fieldset id="detentore\_' + value + '">'  +' <legend>Dati detentore</legend> '  +' <div class="form-row"> '  +' <div class="form-group col-md-2"> '  +' <label for="nameHolder">Nome\*</label> '  +' <input type="text" class="form-control" name="inputNameHolder\_' + value + '" placeholder="Nome detentore" required value="nome"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="surnameHolder">Cognome\*</label> '  +' <input type="text" class="form-control" name="inputSurnameHolder\_' + value + '" placeholder="Cognome detentore" required value="cognome"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="holderAddress">Via\*</label> '  +' <input type="text" class="form-control" name="inputHolderAddress\_' + value + '" placeholder="Via" required value="via"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="holderAddressNumber">N° Civico\*</label> '  +' <input type="text" class="form-control" name="inputHolderAddressNumber\_' + value + '" placeholder="N° Civico" required value="ncivico"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="holderPostalCode">CAP\*</label> '  +' <input type="number" min="0" class="form-control" name="inputHolderPostalNumber\_' + value + '" placeholder="CAP" required value="6900"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="holderPostalCodeCity">Città\*</label> '  +' <input type="text" class="form-control" name="inputHolderCity\_' + value + '" placeholder="Città" required value="citta"> '  +' </div> '  +' </div> '  +' <div class="form-row"> '  +' <div class="form-group col-md-2"> '  +' <label for="emailHolder">Email\*</label> '  +' <input type="email" class="form-control" name="inputEmailHolder\_' + value + '" placeholder="Email detentore" required value="a.a@a.a"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="carBrand">Marca veicolo\*</label> '  +' <input type="text" class="form-control" name="inputCarBrand\_' + value + '" placeholder="Marca veicolo" required value="marca"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="carColor">Colore veicolo\*</label> '  +' <input type="text" class="form-control" name="inputCarColor\_' + value + '" placeholder="Colore veicolo" required value="colore"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="carModel">Modello veicolo\*</label> '  +' <input type="text" class="form-control" name="inputCarModel\_' + value + '" placeholder="Modello veicolo" required value="modello"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="carType">Tipo veicolo\*</label> '  +' <input type="text" class="form-control" name="inputCarType\_' + value + '" placeholder="Modello veicolo" required value="tipo"> '  +' </div> '  +' <div class="form-group col-md-2"> '  +' <label for="plateNumber">N° Targa\*</label> '  +' <input type="text" class="form-control" name="inputPlateNumber\_' + value + '" placeholder="N° Targa" required value="ntarga"> '  +' </div> '  +' </div> '  +'</fieldset>';  Test |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Il problema che ho riscontrato con la possibile soluzione dell’ordinazione di più chiavi sta nel fatto che ogni volta che vado a cliccare il bottone, si mi crea i nuovi input con i nomi differenti, ma mi resetta tutto ciò che avevo scritto negli input precedenti. Per risolvere ciò, inizialmente è stato pensato di salvare tutti i dati in una tabella temporanea utilizzando Ajax ma, una volta provato ad implementarlo, si è rivelato un “fallimento”. Nel pomeriggio mi è stato consigliato di cambiare il mio approccio, non utilizzare più Ajax ma utilizzare i child. Ho rimosso tutto ciò che riguardava l’approccio precedente ed ho cominciato ad implementare il nuovo approccio che si è rilevato molto più semplice e vincente. In pratica vado a creare ogni elemento a mano tramite delle “funzioni” di js ed uso gli appendChild() per concatenarli tra di loro.  Codice  var fieldset = document.createElement("fieldset");  fieldset.setAttribute("id", "detentore\_" + value);  var legend = document.createElement("legend");  legend.appendChild(document.createTextNode("Dati detentore"));  fieldset.appendChild(legend);  var divFormRow = document.createElement("div");  divFormRow.setAttribute("class", "form-row");  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "nameHolder");  label.appendChild(document.createTextNode("Nome\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputNameHolder\_" + value);  input.setAttribute("placeholder", "Nome detentore");  input.setAttribute("required", "true");  input.setAttribute("value", "Nome");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "surnameHolder");  label.appendChild(document.createTextNode("Cognome\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputSurnameHolder\_" + value);  input.setAttribute("placeholder", "Cognome detentore");  input.setAttribute("required", "true");  input.setAttribute("value", "Cognome");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "holderAddress");  label.appendChild(document.createTextNode("Via\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputHolderAddress\_" + value);  input.setAttribute("placeholder", "Via");  input.setAttribute("required", "true");  input.setAttribute("value", "Via");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "holderAddressNumber");  label.appendChild(document.createTextNode("N° Civico\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputHolderAddressNumber\_" + value);  input.setAttribute("placeholder", "N° Civico");  input.setAttribute("required", "true");  input.setAttribute("value", "ncivico");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "holderPostalCode");  label.appendChild(document.createTextNode("CAP\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "number");  input.setAttribute("min", "0");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputHolderPostalNumber\_" + value);  input.setAttribute("placeholder", "CAP");  input.setAttribute("required", "true");  input.setAttribute("value", "6900");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "holderCity");  label.appendChild(document.createTextNode("Città\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputHolderCity\_" + value);  input.setAttribute("placeholder", "Città");  input.setAttribute("required", "true");  input.setAttribute("value", "citta");  div.appendChild(input);  divFormRow.appendChild(div);  fieldset.appendChild(divFormRow);  var divFormRow = document.createElement("div");  divFormRow.setAttribute("class", "form-row");  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "emailHolder");  label.appendChild(document.createTextNode("Email\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "email");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputEmailHolder\_" + value);  input.setAttribute("placeholder", "Email detentore");  input.setAttribute("required", "true");  input.setAttribute("value", "a.a@a.a");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "carBrand");  label.appendChild(document.createTextNode("Marca veicolo\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputCarBrand\_" + value);  input.setAttribute("placeholder", "Marca veicolo");  input.setAttribute("required", "true");  input.setAttribute("value", "Marca");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "carColor");  label.appendChild(document.createTextNode("Colore\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputCarColor\_" + value);  input.setAttribute("placeholder", "Colore veicolo");  input.setAttribute("required", "true");  input.setAttribute("value", "colore");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "carModel");  label.appendChild(document.createTextNode("Modello veicolo\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputCarModel\_" + value);  input.setAttribute("placeholder", "Modello veicolo");  input.setAttribute("required", "true");  input.setAttribute("value", "modello");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "carType");  label.appendChild(document.createTextNode("Tipo veicolo\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputCarType\_" + value);  input.setAttribute("placeholder", "Modello veicolo");  input.setAttribute("required", "true");  input.setAttribute("value", "tipo");  div.appendChild(input);  divFormRow.appendChild(div);  var div = document.createElement("div");  div.setAttribute("class", "form-group col-md-2");  var label = document.createElement("label");  label.setAttribute("for", "plateNumber");  label.appendChild(document.createTextNode("N° targa\*"));  div.appendChild(label);  var input = document.createElement("input");  input.setAttribute("type", "text");  input.setAttribute("class", "form-control");  input.setAttribute("name", "inputPlateNumber\_" + value);  input.setAttribute("placeholder", "N° Targa");  input.setAttribute("required", "true");  input.setAttribute("value", "ntarga");  div.appendChild(input);  divFormRow.appendChild(div);  fieldset.appendChild(divFormRow);  document.getElementById("detentori").appendChild(fieldset);  Test  Viene create una compia identica degli input del detentore, i nomi dei nuovi input sono differenti dai precedenti ed i valori inseriti negli input precedenti, una volta cliccato il bottone, non vengono resettati. |

|  |
| --- |
| 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. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 20.02.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti mi sono occupato dell’implementazione degli ultimi controlli per la pagina di inserimento dei dati, infatti ho implementato una funzione js che controlla, tramite delle espressioni regolari, se ciò che inserisce l’utente è corretto.  Espressione regolare che controlla il numero di telefono  var telephoneNumberRegex = /^[+]{0,1}[(]{0,1}[0-9]{1,4}[)]{0,1}[-\s\./0-9]\*$/;  Espressione regolare che controlla il codice postale  var capRegex = /^[0-9]{4}$/;  Espressione regolare che controlla il numero di targa  var plateRegex = /^[A-Z]{2}[\ ]{1}[0-9]{1,6}$/;  Funzione di controllo in js  function controlSociety(element, regex) {  var submit = document.getElementById("orderButton");  if (regex.test(element.value)) {  element.style.color = "black";  submit.disabled = false;  }else {  element.style.color = "red";  submit.disabled = true;  }  }  Questa funzione riceve, come primo parametro, l’input in questione e come secondo l’espressione regolare che deve utilizzare per eseguire il controllo. Se il contenuto non è quello desiderato lo scrivo in rosso e disabilito il bottone altrimenti lo scrivo in nero (standard) e abilito il bottone submit. Mentre nell’ultima ora di progetti ho sistemato un po’ tutte le funzioni, con commenti, che ho redatto fino ad ora. |

|  |
| --- |
| 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. |

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. |

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. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 27.02.2019 |

|  |
| --- |
| Lavori svolti |
| All’inizio di questa giornata di progetti mi sono occupato dell’installazione del sistema operativo chiamato Raspbian su Raspberry PI. Come prima cosa ho inserito la scheda SD nel mio pc per poterci scaricare il S.O, in seguito ho formattato quest’ultima tramite un software che si chiama SD card formatter (<https://www.sdcard.org/downloads/formatter_4/eula_windows/index.html>)  Una volta installato mi sono ritrovato con la seguente schermata:    Come impostazioni ho, innanzitutto scelto il disco da formattare, la “tipologia” di formattazione ed il nome da dare al disco formattato, ed ho cliccato su *format*. Il prossimo passo è stato quello di scaricare l’immagine di Raspbian dal sito ufficiale di Raspberry (<https://www.raspberrypi.org/downloads/raspbian/>), l’immagine che ho scaricato è quella chiamata: *Raspbian Stretch with desktop*; in modo tale da avere solo il S.O senza software aggiuntivi che potrebbero andare a compromettere le performance del Raspberry. Come ultimo software ho installato *Flash. Flawless* che mi permette di assegnare un’immagine ad un disco. Per l’installazione di questi due software basta semplicemente cliccare su avanti e accettare le condizioni. Una volta reinserita la scheda SD all’interno del Raspberry l’ho avviato e configurato le impostazioni base (lingua, tastiera, ecc). In seguito mi sono occupato di terminare la pagina amministrativa che gestisce la restituzione delle chiavi. In principio dovevo fare visualizzare il nome ed il cognome del detentore che non ha ancora restituito la sua chiave, per fare ciò dovevo passare per 4 tabelle differenti: chiave (per il numero di essa ed il falg che avverte se è stata restituita o meno) 🡪 appartiene (tabella ponte tra chiave ed auto, da essa necessitavo il numero di targa che corrispondeva ad un determinato numero di chiave) 🡪 auto (necessitavo la chiave esterna che corrisponde all’email del detentore, nonché alla chiave primaria della tabella detentore) 🡪 detentore (in base alla email estrapolo il nome ed il cognome del detentore). Inizialmente avevo scritto una funzione con una query dentro l’altra però non funzionava. Grazie al mio compagno Alessandro Colugnat mi sono ricordato dell’esistenza delle JOIN per poter interagire con più tabelle contemporaneamente.  Funzione getNewKeysReturn()  /\*\*  \* Questa funzione si occupa dell'estrapolazione dei dati dalla teblla temporanea tramite una query.  \* @return list Array di stringhe contenente tutti i nomi ed i congnomi dei vari detentori che non hanno restituito la loro chiave.  \*/  function getNewKeysReturn() {  $containerData = array();  $returnKey = 0;  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Query che utilizza una JOIN per estrapolare il nome ed il cognome dei detentori  //che non hanno ancora restituito una chiave.  //Utilizzo un prepare statement per evitare delle SQLInjection.  $stmtKeysReturned = $this->conn->prepare("SELECT detentore.nome, detentore.cognome  FROM detentore  INNER JOIN auto  ON detentore.Email = auto.Email\_detentore  INNER JOIN appartiene  ON auto.N\_targa = appartiene.N\_targa  INNER JOIN chiave  ON appartiene.N\_chiave = chiave.N\_chiave  WHERE chiave.Restituita = ?;  ");  $stmtKeysReturned->bind\_param("i", $returnKey);  $stmtKeysReturned->execute();  $result = $stmtKeysReturned->get\_result();  //Ciclo tutti i "dati" che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  array\_push($containerData, $row);  }  return $containerData;  } |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Una volta configurato il Raspberry l’ho dovuto accantonare perché ho riscontrato un problema con il proxy, a quanto pare non si riesce ad impostare su di esso e per poterlo collegare ad Internet ho dovuto chiedere di liberarmi un indirizzo IP che utilizzerò che IP fisso. |

|  |
| --- |
| 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. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 18.03.2019 |

|  |
| --- |
| Lavori svolti |
| Inizialmente mi sono occupato di portarmi avanti con la configurazione del Rspberry PI, una volta ricollegato e avviato ho configurato l’interfaccia statica *eth0* con l’indirizzo IP che mi era stato liberato due settimane fa.  Interfaccia eth0  auto lo  iface lo inet loopback  auto eth0  iface eth0 inet static  address 10.20.4.90  netmask 255.255.254.0  gateway 10.20.5.254  dns-nameservers 10.20.4.2  Una volta fatto ciò ho commentato le righe presenti nel file *apt.conf* corrispondenti al proxy del server, in modo tale da potermi collegare con successo ad internet. In seguito, tramite una guida (<https://blog.vinczejanos.info/2016/08/31/install-openalpr-on-raspberry-pi-3/>), ho cominciato ad installare tutti i requisiti che servono alla libreria di riconoscimento delle targhe. Il primo passo è quello di scaricare i pacchetti necessari, per farlo ho utilizzato un comando solo che contiene tutti i pacchetti:  apt-get install autoconf automake libtool libleptonica-dev libicu-dev libpango1.0-dev libcairo2-dev cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev python-dev python-numpy libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev virtualenvwrapper liblog4cplus-dev libcurl4-openssl-dev  In seguito ho scaricato le dipendenze da github nel percorso *“cd /usr/local/src/”* tramite il seguente comando:  git clone https://github.com/tesseract-ocr/tesseract  In seguito ho seguito i seguenti passi per completare la configurazione delle dipendenze:  cd /usr/local/src/tesseract  ./autogen.sh  ./configure  make  make install  cd /usr/local/src  wget https://github.com/Itseez/opencv/archive/2.4.13.zip  mv 2.4.13.zip OpenCV-2.4.13.zip  unzip -q OpenCV-2.4.13.zip  cd /usr/local/src/opencv-2.4.13  mkdir release  cd release  cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D BUILD\_NEW\_PYTHON\_SUPPORT=ON -D INSTALL\_C\_EXAMPLES=ON -D INSTALL\_PYTHON\_EXAMPLES=ON -D BUILD\_EXAMPLES=ON ..  Una volta fatto il comando *make* è partita un’installazione che è durata all’incirca una mezzoretta, nel frattempo ho continuato con l’implementazione del sito web.  Ho cominciato e terminato l’implementazione della pagina web che gestisce la visualizzazione dello storico, per completarla mi mancava solo l’implementazione di una funzione che contenga una query di estrapolazione dei dati necessari (nome, congome, n° targa, marca, colore, modello, tipo) e per fare ciò ho deciso di utilizzare una join.  Funzione getAllUsers()  /\*\*  \* Questa funzione si occupa dell'estrapolazione dei dati dalla tabella detentore e auto tramite una join.  \* @return list Array di stringhe contenente tutti i nomi, congnomi e le varie info della propria auto di ogni utente registrato.  \*/  function getAllUsers() {  $containerData = array();  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Query che estrapola il nome, il cognome e le info delle relative auto dei vari detentori.  //Utilizzo un prepare statement per evitare delle SQLInjection.  $stmtViewAllUsers = $this->conn->prepare("SELECT d.Nome, d.Cognome, a.N\_targa, a.Marca, a.Colore, a.Modello, a.Tipo  FROM detentore d, auto a  INNER JOIN auto  WHERE d.Email = a.Email\_detentore;  ");  $stmtViewAllUsers->execute();  $result = $stmtViewAllUsers->get\_result();  //Ciclo tutti i "dati" che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  array\_push($containerData, $row);  }  return $containerData;  }  Riempimento della tabella  <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">  <thead>  <tr>  <th>Nome</th>  <th>Cognome</th>  <th class="notRequired">N° Targa</th>  <th class="notRequired">Marca</th>  <th class="notRequired">Colore</th>  <th class="notRequired">Modello</th>  <th class="notRequired">Tipo</th>  </tr>  </thead>  <tbody>  <?php for ($i=0; $i < count($usersHistory); $i++): ?>  <tr>  <td><?php echo $usersHistory[$i]['Nome']; ?></td>  <td><?php echo $usersHistory[$i]['Cognome']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['N\_targa']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Marca']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Colore']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Modello']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Tipo']; ?></td>  </tr>  <?php endfor; ?>  </tbody>  </table>  Test |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Non ho riscontrato problemi. |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con la configurazione del Raspberry PI |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 20.03.2019 |

|  |
| --- |
| Lavori svolti |
| In questa lezione di progetti ho ripreso a lavorare con il Raspberry PI, in particolare ho ripreso da dove avevo lasciato la scorsa lezione. Come prima cosa, sempre seguendo la documentazione trovata sul web, ho scaricato la seconda dipendenza il fatto è che non sono riuscito ad installarlo. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Come menzionato precedentemente ho avuto un problema nell’installazione della seconda dipendenza sul Raspberry, quest’ultimo appare quando devo avviare la compilazione del codice sorgente tramite il comando “*make*”. Le problematiche che ho riscontrato dopo un paio di tentativi sono due: la prima è che la compilazione si bloccava con un errore e terminava; la seconda è che verso il 9% della compilazione Raspberry si bloccava completamente e cominciava a surriscaldarsi in modo considerevole e preoccupante. Per risolvere tutto ciò ho formattato l’SD e ciò riscaricato il sistema operativo, però sprovvisto di interfaccia grafica in modo tale da non fare consumare troppe risorse da quest’ultima. Per reinstallarlo ho semplicemente seguito la procedura descritta nel diario datato il *2019\_02\_27* utilizzando l’immagine ISO chiamata “*Raspbian Stretch Lite*” preso dallo stesso sito. Una volta reinstallato il S.O. ho riscontrato alcuni problemi con diverse cose: il login; la tastiera; l’interfaccia di rete. Non riuscivo ad effettuare il login (username: *pi* password:*raspberry)* perché il layout della tastiera non era quello svizzero. La tastiera l’ho modificata tramite il seguente file di configurazione: */etc/default/keyboard*    Per quanto riguarda l’interfaccia di rete il problema consisteva nel fatto che non riuscivo a cambiarne il nome, dato che possedeva il seguente: *enxb827eb369c3c*. Dopo svariati tentativi e mancanza di tempo ho lasciato quello ripristinando le impostazioni della rete che avevo anche prima.  Interfaccia enxb827eb369c3c  auto lo  iface lo inet loopback  auto enxb827eb369c3c  iface enxb827eb369c3c inet static  address 10.20.4.90  netmask 255.255.254.0  gateway 10.20.5.254  dns-nameservers 10.20.4.2 |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con la configurazione del Raspberry PI |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 25.03.2019 |

|  |
| --- |
| Lavori svolti |
| All’inizio di questa lezione di progetti ho ripreso con l’installazione della libreria di riconoscimento delle targhe delle automobili. Avendo ricominciato da campo la lezione scorsa ho eseguito l’installazione più semplice, questa volta, infatti ho eseguito l’installazione tramite pacchetti già compilati.  Installazione  wget -O - https://deb.openalpr.com/openalpr.gpg.key | sudo apt-key add -  echo "deb https://deb.openalpr.com/master/ trusty main" | sudo tee /etc/apt/sources.list.d/openalpr.list  sudo apt-get update  sudo apt-get install openalpr openalpr-daemon openalpr-utils libopenalpr-dev  Test funzionamento  # Test European plates  wget http://plates.openalpr.com/h786poj.jpg  alpr -c eu h786poj.jpg  Una volta testato il suo funzionamento ho cominciato a pensare a come implementarlo, ma da qui in poi ho riscontrato problemi. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Il primo problema che ho riscontrato riguardava il fatto che il riconoscimento di targhe è per quelle europee e/o americane, per implementare il riconoscimento delle targhe svizzere si deve creare un template da zero in modo tale che la libreria le riconosca. Un secondo problema riguarda il file CSV di output, infatti non riuscivo a trovare un modo per crearlo. Dopo alcune ricerche ho trovato come fare (<http://doc.openalpr.com/on_premises.html#id8>), come riferisce questa guida ufficiale si deve ottenere una licenza di 14 giorni gratuita, e questa l’ho ottenuta.  Licenza    Una volta ottenuta, per Linux, si doveva installare un pacchetto differente per eseguire un video e creare il file di output.  Installazione nuovo pacchetto video  sudo apt update && sudo apt install openalpr-video  Una volta eseguito questo commando ricevo il seguente output:    Per testarlo si deve usare il comando *alprvideocli* <options>, il problema è che non mi riconosce il comando e, vedendo l’output, presumo che non sia più possibile installare questo pacchetto. Ho provato pure a disinstallare tutto e provare con l’installazione dei pacchetti non precompilati, però ho riscontrato lo stesso problema di quando ho eseguito l’installazione dei pacchetti con raspberry con GUI, si blocca tutto e comincia a riscaldarsi e l’unica cosa che posso fare è togliergli la corrente per spegnerlo. |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con la configurazione del Raspberry PI |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 26.03.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti ho ripreso da dove mi ero fermato la scorsa volta, cioè installare la seconda dipendenza, ma senza successo. In seguito ho parlato un po’ con il mio responsabile di progetto, abbiamo tirato fuori tre punti problematici da dover risolvere. Il primo è quello di capire come creare il file di output CSV contenente le targhe riconosciute; il secondo riguarda la creazione di una sorta di template per il riconoscimento delle targhe svizzere, non facendo parte dell’unione europea non sono comprese nell’opzione di riconoscimento delle targhe eu (europee); il terzo problema riguarda di implementare il riconoscimento su raspberry, dato che fuoriescono problemi ad ogni passo e/o possibile procedura. Per permettermi di continuare ho utilizzato una macchina virtuale contenente ubuntu server 16.04 già connessa ad internet, su di essa ho installato la libreria utilizzando la procedura con le librerie già precompilate senza riscontrare problemi.  Test riconoscimento delle targhe europee      Una volta fatto ciò ho provato ad implementare il file CSV, per fare ciò ho ripreso la guida ufficiale che spiega come ottenere i file CSV e SQL da un video contenente delle targhe.  <http://doc.openalpr.com/on_premises.html#id8>  Installazione  sudo apt update && sudo apt install openalpr-video  Una volta installato si esegue il comando *alprvideocli* con due argomenti necessari, il primo è la path dove generare i file di output, il secondo è la path del video da analizzare e tramite l’opzione -c gli si specifica i paesi, nel mio caso *-c eu* per dirgli di cercare tra le targhe europee.  Test    Output CSV    Il file csv è strutturato nel modo seguente:    Una volta capito come fare il file CSV ho dedicato il resto della giornata alla creazione del template per le targhe svizzere, innanzitutto ho letto una guida presente sul sito ufficiale che spiega come creare dei template personalizzati (<http://doc.openalpr.com/opensource.html#training-ocr>). Anche se ho seguito questa guida ho riscontrato problemi che mi hanno impedito l’implementazione del template tramite le due procedure spiegate nella pagina. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Inizialmente pensavo che il problema del fallimento dell’installazione delle dipendenze della libreria di openalpr per il riconoscimento delle targhe mediante una telecamera fosse dovuto alla GUI di raspbian, ma non è così perché ho riscontrato lo stesso problema, cioè un *fatal error* al 7% dell’installazione. Ho deciso quindi di accantonare questa procedura ed utilizzare l’installazione con le librerie precompilate in modo tale da installare il tutto senza problemi. Seguendo la guida per la creazione del template delle targhe inizialmente ho creato un file di configurazione per le targhe svizzere nella cartella */usr/share/openalpr/runtime\_data/config*, dove sono contenuti tutti i file di configurazione per le targhe dei diversi paesi, il file l’ho denominato sw.conf ed al suo interno ho dovuto inserire le varie misure delle targhe svizzere. Tramite il seguente link (<https://fahrzeugausweise.ch/it/informazioni-generali/#tipi-di-targhe-di-controllo>) le differenti grandezze delle targhe, ciò che mi mancava erano le grandezze delle varie lettere e spaziature e per trovarle sono dovuto andare, munito di righello, ai parcheggi e prendere le misure da una targa di un veicolo esposto…  File di configurazione contenente le varie grandezze    Una volta terminato ciò sono passato al prossimo punto che è quello che mi ha causato tutti i problemi e l’abbandono di queste due procedure, questo punto riguardava la creazione delle *character tiles* (non so la traduzione corretta) e per farle si possono seguire due procedure. La prima consiste nell’utilizzare delle immagini prese da delle targhe; la seconda nella creazione di un TTL font che sia all’incirca uguale al font delle targhe svizzere. La prima procedura l’ho accantonata subito perché richiedeva un software con GUI e, lavorando con un MV sprovveduta di GUI mi era impossibile seguirla. Allora sono andato con la seconda procedura che consisteva in 7 passaggi:   1. Capire tutti i caratteri e numeri che possono essere presenti nelle targhe svizzere 2. Creare un documento Word contenente tutti questi caratteri      1. Copiare tutti i caratteri in un file di testo senza nessuna spaziatura tra di loro 2. Stampare il documento Word 3. Eseguire almeno 5 foto con angolature differenti al foglio appena stampato 4. Salvare queste immagini in una cartella 5. Eseguire il programma openalpr-utils-binarizefontsheet per produrre delle *tile* per ogni immagine   Il problema che ho riscontrato riguarda il programma da usare nel punto 7, infatti dopo circa 1 ora di ricerca non ho trovato alcun modo per installarlo su Linux e tanto meno su Windows e alla fine ho accantonato pure questa procedura. Verso l’ultima ora di progetti mi sono un po’ alterato con tutto ciò a tal punto che sono andato nella cartella principale della libreria di openalpr (*/usr/share/openalpr/runtime\_data*) e ho cominciato ad aprire ed analizzare ogni file per vedere se potessi implementare una soluzione differente. Il primo file che ho modificato è il file *eu.conf* presente nella cartella *config*, ho semplicemente aggiunto ch agli alias. Il secondo file che sono andato ad analizzare è il file *province.labels* presente nella cartella *ocr/eu*, al suo interno ho trovato una riga contenente *eu-ch*. Il terzo file che sono andato a modificare è il file *eu.patterns* presente nella cartella *postprocess*, al suo interno ho aggiunto il pattern delle targhe svizzere: 2 lettere e da 1 a 6 numeri (@ 🡪 qualsiasi lettera; # 🡪 qualsiasi numero).  Pattern    Test |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Migliorare il riconoscimento delle targhe svizzere |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 27.03.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti mi sono occupato di svolgere dei test per i vari riconoscimenti delle targhe svizzere per vedere se e quanto funzionasse bene la soluzione che avevo adottato la scorsa lezione. Dopo qualche ora di test di targhe ho constatato che questa soluzioine funziona per il 90% dei casi, fa fatica a riconoscere cerche targhe svizzere per via di alcune angolazioni, le targhe con un solo numero non le riconosce. Il tempo restante l’ho passato per cercare una possibile integrazione con la webcam e l’impostazione del file di output CV ma per il momento non ho ancora trovato una soluzione. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Non ho riscontrato alcun problema con la sua rispettiva soluzione. |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Migliorare il riconoscimento delle targhe svizzere |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 01.04.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti ho eseguito delle ricerche su come riuscire a convertire una serie di immagini (immagini di targhe fatte dalla webcam) in un video mp4 tramite linea di comando. Inizialmente sono partito con l’implementazione di una serie d’immagini fatta dalla videocamera ogni minuto, per fare ciò ho inizialmente creato uno script che mi salvasse ogni immagine in una determinata cartella con un determinato nome (la data e orario attuali).  Script  Mkdir /etc/openalpr/image/webcam  Vim script.sh  #!/bin/bash  DATE=$(date +"%Y-%m-%d\_%H%M")  fswebcam -r 1280x720 --no-banner /etc/openalpr/image/$DATE.jpg  In seguito ho usato un software per creare un time-lapse, quest’ultimo si chiama cron. Per avviarlo correttamente, innanzitutto, digito il relativo comando, scelgo l’editor di testo con cui modificare il file di configurazione ed aggiungo una linea di codice che permette di eseguire uno screen ogni minuto utilizzando lo script creato in precedenza per creare e salvare le immagini.  Time-lapse  crontab -e  \* \* \* \* \* /etc/openalpr/image/webcam/script.sh 2>&1  Gli asterischi iniziali sono fondamentali, il primo riguarda I minuti, il secondo le ore, il terzo I giorni (numero), il quarto i mesi ed il quinto i giorni della settimana (un giorno specifico).  In seguito ho trovato questo applicativo utilizzabile tramite il comando *ffmpeg* che permette di convertire una serie di immagini in un video mp4. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Mentre eseguivo alcune prove con *ffmpeg* ho notato che non mi crea il video nel modo giusto. Passandogli, per esempio, 3 immagini ed eseguendo il comando mi viene scritto che due immagini esistono già e mi chiede se vuoi sovrascriverle o no, se premo no il comando si arresta, se premo si mi crea il video con soltanto un frame raffigurante la prima immagine che non mi aveva chiesto se sovrascrivere o no e la sua durata è pari a 0 secondi.  Comando  ffmpeg -framerate 25 -i \*.jpg -c:v libx264 -profile:v high -crf 20 -pix\_fmt yuv420p output.mp4 |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Trovare un modo per la creazione di un video raffigurante delle targhe. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 02.04.2019 |

|  |
| --- |
| Lavori svolti |
| L’inizio della mattinata ho cominciato a ricercare una libreria che potesse, tramite una webcam, rilevare dei cambiamenti tra frame e frame per poi permettermi di, una volta rilevato un cambiamento, far partire una registrazione per riprendere la targa dell’auto passante.  <https://github.com/beije/motion-detection-in-javascript>  Dopo alcune ricerche ho trovato questa libreria composta da tre file scritti in JavaScript, questa libreria permette, tramite una webcam, di identificare se ci sono stati dei cambiamenti e di conseguenza se si è mosso qualcosa. Quest’ultima confronta un frame e quello successivo e, tramite i pixel, confronta se ci sono stati dei cambiamenti e se ci sono stati contorna l’area in cui è avvenuto il cambiamento con un riquadro rosso. Una volta trovata l’ho scaricata sul mio server ubuntu e, tramite apache, l’ho avviata. Il problema successivo consisteva nel riuscire, una volta catturato un cambiamento, far partire la registrazione della webcam per qualche secondo. Per fare ciò ho pensato di utilizzare il solito *ffmpeg* tramite il seguente comando:  ffmpeg -f v4l2 -framerate 25 -video\_size 640x480 -i /dev/vid eo0 -t 5 output.mkv  Questo comando mi permette di registrare un video *.mkv* a 25 FPS e di durata pari a 5 secondi.  Il problema consiste nell’integrazione di questo comando all’interno della libreria. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Per provare ad integrare le due cose, dopo svariate ricerche, ho trovato una possibile soluzione che utilizza i nodejs. Questi ultimi vengono usati per utilizzare comandi, script o file esterni al file JavaScript. All’interno del file */var/www/html/js/MotionCapture/ImageCapture.js* presente nella libreria scaricata in precedenza ho trovato la parte di codice in cui effettua il controllo tra i pixel di un frame e i pixel del frame successivo (riga 85), all’interno di quella condizione ho aggiunto la parte di codice utilizzante un nodejs per poter eseguire uno script *.sh* contenente il comando per avviare il video di durata di 5 secondi.  Codice  const exec = require('child\_process').exec;  var yourscript = exec('sh stream.sh',  (error, stdout, stderr) => {  console.log(stdout);  console.log(stderr);  if (error !== null) {  console.log(`exec error: ${error}`);  }  });  Il problema è che non avvia lo script e non sono riuscito a campirne il motivo. |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Far funzionare i nodjs presenti nella libreria. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 03.04.2019 |

|  |
| --- |
| Lavori svolti |
| Durante la prima mezzoretta di progetti abbiamo parlato, insieme a tutta la classe, di come si svolgeranno i LPI. In seguito ho ripreso dal codice che avevo redatto l’ultima volta, effettuando altre ricerche ho capito che non potevo utilizzare i nodejs per utilizzare la funzione *exec()*, questo perché lato javascript client non è possibile utilizzarli, per poterli implementare avrei dovuto creare un server nodejs. In seguito ho optato per l’utilizzo di Ajax all’interno del file contenente il riconoscimento di un cambiamento tra i vari frame.  Codice Ajax  var xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  document.getElementById("demo").innerHTML = this.responseText;  }  };  xhttp.open("GET", "startScript.php", true);  xhttp.send(); |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Una volta integrato il codice Ajax ho privato a testarlo ma senza risultati positivi, infatti guardando la console mi esce l’errore seguente:    Ho provato a spostare lo script in path differenti ma continua a non trovare il file. |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Far funzionare la chiamata allo script dal file js e terminare la pagina delle statistiche. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 08.04.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti mi sono occupato di portare avanti l’ultima pagina web che mi mancava, cioè la pagina delle statistiche. Inizialmente mi sono occupato dell’importazione del file CSV tramite un input di tipo *file*, in seguito ho creato una tabella chiamata *statistiche* in cui sono presenti due record, il primo è un semplice Id che funge da chiave primaria, il secondo record chiamato *N\_targa* conterrà tutti i vari numeri di targhe presenti nel file CSV *img-group.csv*. Tramite una query vado a salvare i vari numeri di targa, tramite una seconda query vado ad eliminare tutti i dati che non necessari che mi va ad inserire nella tabella tramite la prima statistica.  Tabella statistiche    Form di importazione del file CSV  <!-- Begin import CSV file -->  <form action="" method="post" name="uploadCSV" enctype="multipart/form-data">  <div class="row">  <div class="col-md-9 mb-3">  <input type="file" class="custom-file-input" id="customFile" name="customFile" accept=".csv">  <label class="custom-file-label" for="customFile">Importa il file CSV</label>  </div>  <div class="col-md-3 mb-3 container">  <button type="submit" id="submit" name="import" class="btn btn-primary">Import</button>  </div>  </div>  </form>  <!-- End import CSV file -->  All’interno del form è presente l’attributo *enctype*, quest’ultimo è necessario per la lettura del file CSV. All’interno dell’input, di tipo *file*, è presente un attributo per permettere di accettare unicamente i file CSV.  Chiamata alla funzione di salvataggio dei dati  <?php  //Eseguo il require dei file esterni necessari.  require\_once('db\_connection.php');  //Creao gli oggetti delle classi necessarie;  $conn = new connection();  //Controllo se é stato eseguito il POST  if ($\_SERVER['REQUEST\_METHOD'] == "POST") {  if (isset($\_POST["import"])) {  $fileName = $\_FILES["customFile"]["tmp\_name"];  if ($\_FILES["customFile"]["size"] > 0) {  $conn->csvImport($fileName);  }else {  echo "CSV vuoto!!!!!!!!!!";  }  }  }  ?>  Funzione di salvataggio dei dati del file CSV all’interno della tabella  /\*\*  \* Questa funzione si occupa di salvare il contenuto di un file csv  \* all'interno della tabella statistiche in cui saranno presenti tutte  \* le targhe delle auto che sono passate dalla sbarra.  \* @param fileName L'array contenente tutti i dati del file csv.  \*/  function csvImport($fileName){  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  $file = fopen($fileName, "r");  while (($column = fgetcsv($file, 10000, ",")) !== FALSE) {  //Query che permettono l'inserimento dei numeri di targa nella tabella statistiche.  //Utilizzando un prepare statement per evitare delle SQLInjection.  $stmtInsert = $this->conn->prepare("INSERT into statistiche (N\_targa) values ('$column[2]')");  if(!$stmtInsert->execute()){  echo "La query d'inserimento del contenuto del file CSV  presente nella funzione csvImport() non funziona (riga:320).";  }  //Query che permettono l'eliminazione dei dati che non mi interessano.  //Utilizzando un prepare statement per evitare delle SQLInjection.  $stmtDelete = $this->conn->prepare("DELETE FROM statistiche WHERE N\_targa = 'plate\_numb'");  if(!$stmtDelete->execute()){  echo "La query di eliminazione della prima riga del file CSV  presente nella funzione csvImport() non funziona (riga:327).";  }  }  } |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Questa lezione non ho riscontrato problemi |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Far funzionare la chiamata allo script dal file js e terminare la pagina delle statistiche. |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 09.04.2019 |

|  |
| --- |
| Lavori svolti |
| Durante questa lezione di progetti mi sono dedicato solamente alla documentazione. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Questa lezione non ho riscontrato problemi |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Consegna del progetto |

Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 10.04.2019 |

|  |
| --- |
| Lavori svolti |
| Consegna progetto |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Questa lezione non ho riscontrato problemi |

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

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| - |