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. |