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. |