Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Scuola arti e mestieri Trevano |
| Data | 22.10.2019 |

|  |
| --- |
| Lavori svolti |
| Oggi mi sono concentrato a provare tutto quello che ho creato tramite un client diverso dal mio quindi ho chiesto ad Andrea Rauso se poteva darmi un utente per accedere al suo Database e quindi farci il backup. Sono risultati diversi errori nella creazione e quindi ho lavorato sulla creazione e su altri metodi perfezionandoli e rendendoli funzionali.  public function addDB() {  if (!validate::*checkNotEmpty*($\_POST['port'])) {  $port = 3306;  } else {   $port = $\_POST['port'];  }  $name = validate::*secureString*($\_POST['name']);  $username = validate::*secureString*(($\_POST['username']));  $password = validate::*secureString*($\_POST['password']);   if (validate::*checkNotEmpty*($name) && validate::*checkNotEmpty*($username) && *filter\_var*($\_POST['ip'], *FILTER\_VALIDATE\_IP*) && validate::*checkInt*($port) && $port < 65536) {  if (db\_manager::*executeQuery*("SELECT *\** from database\_data where name ='" . $name . "'") == NULL) {   */\*\*  \* Inserisco tutti i dati nel database  \*/* db\_manager::*execute*(  "INSERT into database\_data (name,ip,port,user,pass,settings\_id) values ('" . $name . "','" . $\_POST['ip'] . "','" . $port . "','" . $username . "','" . $password . "','1')"  );   $this->home();    } else {   */\*\*  \* Ritorno un messaggio di feedback  \*/* $\_POST['creationError'] = 'Esiste già un database con questo nome';  }  } else {  */\*\*  \* Ritorno un messaggio di errore  \*/* $\_POST['creationError'] = 'Uno dei valori inseriti non è valido';  }   $this->home(); }  Questo metodo si occupa di aggiungere i dati del database alla tabella database\_data. All’inizio controllo se è stata inserita una porta che è un campo facoltativo se non è stata inserita uso il valore di default 3306. In seguito controllo tutti gli input, controllo se esiste già un database con lo stesso nome e in fine aggiungo il db alla tabella.  public function deleteDB($id) {  */\*\*  \* Controllo l'id prima di eliminare il database  \*/* if (validate::*checkInt*($id)) {   */\*\*  \* Elimino l'utente con l'id corrispondente  \*/* db\_manager::*execute*(  "DELETE from database\_data where id = " . $id  );  }   */\*\*  \* Ritorno alla pagina degli utenti  \*/* $this->home(); }  Nel tempo rimanente ho creato questo metodo che invece permette di eliminare un DB tramite l’id. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Ho avuto problemi con la connessione al database di Andrea perché non c’era un utente che permettesse le connessioni dal esterno. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono in linea con la pianificazione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Creare la funzione per la creazione delle impostazioni. |