

Diario di lavoro

Luogo	SAMT
Data	25.09.2018

Lavori svolti

Sta mattina ho provato a risolvere il problema dell'invio delle mail, una volta sistemato, come spiegato nella sezione "Problemi riscontrati e soluzioni adottate" ho notato che gmail mi bloccava il passaggio delle mail perché l'app con cui provavo a inviarlo non era sicura, così ho provato a dare il permesso anche alle app sconosciute di passare le mail con la mail di Google, ma comunque il problema persiste, perché non riesce comunque ad inviare le mail. Allora ho provato con l'hotspot del mio telefono, avendo visto che a casa mi dava risultati diversi, ed effettivamente senza il proxy funzionava, così ho chiesto al professore se, per finire questa parte ed evitare il problema potevo continuare ad usare l'hotspot e me l'ha consentito, così sono riuscito a finire correttamente il progetto, correggendo dei piccoli errori nell'inserimento dei dati nel database e l'invio del messaggio della mail e questo è il risultato finale.

```

11 public function sendMail()
12 {
13     require_once 'application/models/connection.php';
14     $connection = new Connection("localhost","root","", "ripetizioni");
15
16     //Get the entered data
17     if ($_SERVER["REQUEST_METHOD"] == "POST") {
18         $name = $_POST["name"];
19         $surname = $_POST["surname"];
20         $mail = $_POST["mail"];
21         $phone = $_POST["phone"];
22         $via = $_POST["via"];
23         $cap = $_POST["cap"];
24         $city = $_POST["city"];
25         $pass = $_POST["pass"];
26         $pass = hash('sha512', $pass);
27         $role = $_POST["role"];
28     }
29
30     if (strcmp($name, "") || strcmp($surname, "") || strcmp($mail, "") || strcmp($phone, "") || strcmp($via, "") || strcmp($cap, "") || strcmp($city, "") || strcmp($pass, "") || strcmp($role, "")) {
31         $connection->insertRecord($name, $surname, $mail, $phone, $via, $cap, $city, $pass, $role);
32     }
33
34     if(strcmp($role, "coach")){
35         $msn = "'$name $surname' attiva il tuo account";
36         $msn = $msn . "\nhttp://localhost:8042/MVC/registration/register/$mail";
37         $headers = "From: progettoripetizioni@gmail.com";
38         $a = mail($mail, "NON RISPONDERE A QUESTA MAIL", $msn, $headers);
39     }else{
40         //<a href=''>Attiva</a>
41         //<a href=''>Non attivare</a>
42         $to = "giairo.mauro@samtrevano.ch";
43         $subject = "NON RISPONDERE A QUESTA MAIL";
44         $msn = "'$name $surname' vorrebbe creare un account sulla mail $mail";
45         $msn = $msn . "\nhttp://localhost:8042/MVC/registration/register/$mail";
46         $msn = $msn . "\nhttp://localhost:8042/MVC/registration/noRegister/$mail";
47         $headers = "From: progettoripetizioni@gmail.com";
48         $a = mail($to, $subject, $msn, $headers);
49     }

```

Questa è la funzione che viene richiamata dal bottone di "submit" della registrazione, che si occupa di stabilire la connessione con il database e tramite il metodo "insertRecord" inserisce i valori nel database, ma inattivi.

Poi controlla se l'iscrizione è avvenuta da un coach o un allievo. Se è un allievo invia una mail a lui stesso con un link di attivazione dell'account, mentre se è un coach invia una mail all'amministratore del server che avrà la scelta tra due link se attivare o meno l'account.

```

34 public function insertRecord($name, $surname, $mail, $phone, $via, $cap, $city, $pass, $role)
35 {
36     $active = 0;
37
38     //Connect to database
39     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
40     // Check connection
41     if (!$conn) {
42         die("Connection failed: " . mysqli_connect_error());
43     }
44     //echo "Connected successfully";
45     $sql = $conn->prepare("INSERT INTO utente (nome, cognome, telefono, email, password, via, CAP, citta, nome_tipo, attivo)
46     VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
47
48     $sql->bind_param("ssssssissi", $name, $surname, $phone, $mail, $pass, $via, $cap, $city, $role, $active);
49
50     if ($sql->execute()) {
51         // echo "New record created successfully";
52     } else {
53         //echo "Error: " . $sql . "<br>" . $conn->error;
54         $msg = "Registrazione utente non avvenuta, l'utente é già esistente.";
55         return $msg;
56     }
57     $conn->close();
58 }

```

Nell'immagine soprastante è mostrata la funzione di inserimento dei dati aggiornata, ho messo una prepared statement, per evitare che si possano fare delle sql injection e dare così maggior sicurezza al sito. Si può notare che ho creato la variabile "\$active" che contiene lo stato d'attivazione dell'utente, che inizialmente è zero, e poi viene inserito tramite il "bind_param"

```

63 public function activeUser($mail)
64 {
65     //Connect to database
66     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
67     // Check connection
68     if (!$conn) {
69         die("Connection failed: " . mysqli_connect_error());
70     }
71     //echo "Connected successfully";
72     $sql = $conn->prepare("UPDATE utente SET attivo = 1 WHERE email LIKE ?");
73
74     $sql->bind_param("s", $mail);
75
76     if ($sql->execute()) {
77         // echo "New record created successfully";
78     } else {
79         //echo "Error: " . $sql . "<br>" . $conn->error;
80         $msg = "Registrazione utente non avvenuta, l'utente é già esistente.";
81         return $msg;
82     }
83     $conn->close();
84 }

```

Questa funzione è quella che viene richiamata dal link tramite cui l'admin attiva un coach, o un allievo si attiva da solo, ed essa richiama un'altra funzione che attiva il l'utente nel database dopodiché gli invia una mail informandolo che il suo account è stato attivato. L'immagine qui sopra mostra la funzione che attiva l'utente, sempre con l'uso delle prepared statements, aggiorna il campo "attivo" dell'utente e lo porta a 1.

```

57     public function noRegister($mail)
58     {
59         //Send email to the not activated user
60         $msn = "Il suo account non è stato attivato";
61         $msn = $msn . "\nPer maggiori informazioni contattare l'admin a questa mail:";
62         $msn = $msn . "\ngiairo.maro@samtreveno.ch";
63         $headers = "From: progettoripetizioni@gmail.com";
64         $a = mail($mail, "NON RISPONDERE A QUESTA MAIL", $msn, $headers);
65
66         header("location: ". URL);
67     }

```

Questa funzione è invece quella che non attiva il coach ma semplicemente gli manda una mail con scritto che non è stato attivato e di chiedere maggiori informazioni all'amministratore. Una volta finito, ho lavorato sul login dell'utente e ho messo a posto il form che userò, inserendo un controllo anche nella mail del login, per segnalare all'utente se è errata ce viene fatto allo stesso modo del campo nella registrazione, e ho creato le funzioni che mi servono.

```

24     public function log_in()
25     {
26         require_once 'application/models/connection.php';
27         $connection = new Connection("localhost", "root", "", "ripetizioni");
28         if ($_SERVER["REQUEST_METHOD"] == "POST") {
29             $mail = $_POST["mail"];
30             $pass = $_POST["pass"];
31             $pass = hash('sha512', $pass);
32             //echo $pass;
33         }
34         if ((strcmp($mail, "") || strcmp($pass, ""))) {
35             $var = ($connection->getUser($mail, $pass));
36             echo $var;
37             switch ($var) {
38                 case 2: // LOGIN ADMIN
39                     $_SESSION['$mail'] = $mail;
40                     header("location: http://localhost:8042/MVC/home/coachIndex/" . $_SESSION['mail']);
41                     break;
42
43                 case 3: // LOGIN USER
44                     $_SESSION['$mail'] = $mail;
45                     header("location: http://localhost:8042/MVC/home/userIndex/" . $_SESSION['mail']);
46                     break;
47
48                 case 4: // INACTIVE USER
49                     $this->phpAlert("Utente inattivo");
50                     $this->index();
51                     break;
52
53                 case 5: // WRONG PASSWORD
54                     $this->phpAlert("E-mail o password errati");
55                     $this->index();
56                     break;
57                 default:
58                     $this->phpAlert("Utente non registrato");
59                     $this->index();
60                     break;
61             }
62         }
63     }

```

Questa funzione viene richiamata dal bottone di "submit" nel form di login, fa la connessione al database, prende i dati che gli arrivano controlla che non siano vuoti, dopodiché tramite un'altra funzione prende i dati dell'utente e tramite uno switch controlla che l'utente esista, i dati inseriti siano corretti e l'utente sia attivo, e in quel caso fa un login, aprendo una sessione per quell'utente, in base a se lui è coach o allievo, il login dell'amministratore sarà l'ultimo che andrò a gestire. In caso di errori mostra un "alert" segnalando l'errore e poi va alla pagina di login.

Per provare il login per le 2 pagine ho creato un file di prova nelle cartelle dedicate, da domani inizierò a lavorare alle pagine del coach.

```

89 public function getUser($mail, $pass)
90 {
91     //Connetto al database
92     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
93     if ($conn->connect_error) {
94         die("Connection failed: " . $conn->connect_error);
95     }
96
97     $sql = $conn->prepare("SELECT * FROM utente WHERE email LIKE ?");
98     $sql->bind_param("s", $mail);
99     //print_r($result);
100    if ($sql->execute()) {
101        $result = $sql->get_result();
102        // output data of each row
103        while($row = $result->fetch_assoc()) {
104            echo('LOG: ' . $row["password"]); /*'$pass', */
105            if (!(strcmp($pass, $row["password"]))) {
106                if(!strcmp($row["attivo"], 1)){
107                    if(!strcmp($row["nome_tipo"], "amministratore")){
108                        // LOGIN ADMIN
109                        return 1;
110                    }else if(!strcmp($row["nome_tipo"], "coach")){
111                        // LOGIN USER
112                        return 2;
113                    }else{
114                        // LOGIN USER
115                        return 3;
116                    }
117                }else{
118                    //INACTIVE USER
119                    return 5;
120                }
121            }
122            else {
123                // WRONG PASSWORD
124                return 6;
125            }
126        }
127    }
128    else {
129        // NOT REGISTERED USER
130        return 5;
131    }
132    $conn->close();
133 }

```

Questa funzione è quella richiamata dal metodo “log_in” sopra menzionato per controllare che l’utente esiste e prenderne e confermarne i dati. Come si può notare prende tutti i dati dell’utente e per ogni errore ritorna un numero diverso, che poi verrà controllato nell’altra funzione per decidere cosa farne, e anche a dipendenza del tipo di utente viene ritornato un numero diverso in modo da poterli gestire separatamente.

Per rendere il codice un po' più pulito ho anche deciso di dividere in 2 classi separate ciò che concerne il login e la registrazione, con anche 2 classi dedicate appositamente per l’una o rispettivamente per l’altra.

Problemi riscontrati e soluzioni adottate

Ho provato a sistemare il problema delle mail, sono andato a controllare sul sito di Google le impostazioni di cui ha bisogno, e ho notato che alcune informazioni che inserivo erano errate, allora ho provato ad inserire la porta che mi dicevano oro per il tls e finalmente iniziava a funzionare qualcosa. Dopo aver dato il permesso di inviare le mail da app estranee con Google comunque il problema persisteva e non venivano inviate le mail. Dopodiché ho iniziato ad usare l'hotspot del telefono essendo che con quello funzionava.

Una volta che funzionava ho provato a registrare l'utente ma mi dava comunque alcuni errori, uno nell'inserimento dei dati, perché ho iniziato ad usare le prepared statements e ho avuto dei problemi perché inserivo direttamente il valore, ma mi è bastato metterlo in una variabile e inserire quella nel "bind_param", vedi capitolo sopra. Un altro errore è sorto nel messaggio dell'invio della mail, perché inviavo il tag <a> di html per non scrivere tutto il link ma non funzionava quindi ho dovuto ambiare ed inviare tutto il link.

Ho avuto altri piccoli errori nel login, ma erano quasi tutti solo di nomi di variabili sbagliate o richiami errati. Un errore però era perché non avevo compreso in modo corretto il funzionamento dei prepared statements quindi ho dovuto riguardare diverse informazioni sulla documentazione ufficiale di php sul sito <http://php.net/>.

Un errore ancora costante è il funzionamento delle mail con la rete di scuola, motivo per cui ho inviato una mail al docente responsabile della gestione del proxy e ne parlerò anche con lui di persona per risolvere il problema.

Punto della situazione rispetto alla pianificazione

Sono leggermente in anticipo rispetto al gantt, questo perché il form di login e registrazione sono stati più veloci del previsto, anche se se non avessi avuto tutti gli errori che mi ha causato il proxy l'avrei finito prima.

Programma di massima per la prossima giornata di lavoro

Sistemerò il problema del proxy e l'invio delle email.

Inizierò, se riesco, la parte grafica delle pagine del coach.