

# Diario di lavoro

Luogo	SAMT
Data	06.11.2018

## Lavori svolti

Questa mattina inizialmente ho controllato il funzionamento della pagina di richiesta dei corsi da parte degli allievi, e ho ottimizzato il codice in modo da correggere eventuali problemi e rendere più leggibile e “snello” il codice in sé.

```

350 if(!$alreadyIn){
351     //Se gli utenti iscritti sono il massimo possibile modifico la variabile di controllo di spazio libero
352     if(count($courseSub) == $learners){
353         $checkFreeSpace = false;
354     }else{
355         //Se c'è spazio nel corso inserisco l'allievo e finisco di controllare
356         //Inserisco l'utente che si è iscritto al corso e il corso al quale si iscrive
357         return $this->setLearner($learnedCourses[$i]);
358         break;
359     }
360     $courseSub = array();
361 }else{
362     break;
363 }
364 }else{
365     return "Errore nell'inserimento dei dati";
366 }
367 }
368
369 //Controllo se non c'era un corso senza posti liberi
370 //In caso non ci siano stati corsi liberi ne crea uno nuovo e si iscrive
371 if(!$checkFreeSpace){
372     $this->insertCourseFromLearner($day, $hour, $learners, $subject);
373 }
374 }else{
375     //Creo il corso e iscrivo l'utente ad esso
376     $this->insertCourseFromLearner($day, $hour, $learners, $subject);
377 }

```

Questa è la parte che è cambiata del codice, la parte finale.

Se si paragona alla versione precedente si può notare che è molto più piccola ed è stato aggiunto un controllo. Il controllo aggiunto è quello dello spazio libero nel corso, che viene controllato con una variabile booleana, se non c'è spazio viene portata a “false” in caso contrario registra l'utente e smette di controllare. Alla fine del ciclo controlla se nessun corso era libero ne crea uno e inserisce l'utente in una funzione esterna, come fa anche se non esistono corsi come quello richiesto.

Durante i controlli mi sono accorto di un errore nei controlli degli input della pagina, spiegati nella sezione “problemi riscontrati e soluzioni adottate”.

Una volta finito questo problema ho messo a posto la pagina del docente di creazione dei corsi, in modo che al cambio della materia vengano mostrati a fianco le categorie annesse, come nella pagina di ricerca dei corsi dell'allievo. Nel farlo però ho riscontrato un problema, spiegato nella parte seguente.

Dopodiché ho iniziato a lavorare alle pagine dell'amministratore.

Inizialmente ho creato l'instestazione molto simile a quella degli altri utenti ma con differenze nei nomi delle pagine, poi ho creato la classe model e il caso nel login per accedere alla pagina principale, quella dei coach, al login, e le funzioni nel “model” che fanno accedere alle relative pagine.

```

7 public function coach(){
8     //Richiedo la connessione
9     require_once 'application/models/connection.php';
10    $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
11
12    if (!isset($_SESSION['mail'])) {
13        header("location: ". URL);
14    }else{
15        //Prendo tutti i valori dell'utente
16        $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
17
18        //controllo se ha fatto il login un coach
19        if(!strcmp($userData, "allievo")){
20            header("location: ". URL ."home/user/");
21        }else if(!strcmp($userData, "coach")){
22            header("location: ". URL ."home/coach/");
23        }
24    }
25
26    $dataArray = $connection->getUser($_SESSION['mail']);
27    require 'application/views/_templates/header.php';
28    require 'application/views/admin/Static/header.php';
29    require 'application/views/admin/coach.php';
30 }

```

Questa è la funzione che va ad aprire le pagine per mostrare la lista dei coach.

```

35 public function allievi(){
36     //Richiedo la connessione
37     require_once 'application/models/connection.php';
38     $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
39
40     if (!isset($_SESSION['mail'])) {
41         header("location: ". URL);
42     }else{
43         //Prendo tutti i valori dell'utente
44         $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
45
46         //controllo se ha fatto il login un coach
47         if(!strcmp($userData, "allievo")){
48             header("location: ". URL ."home/user/");
49         }else if(!strcmp($userData, "coach")){
50             header("location: ". URL ."home/coach/");
51         }
52     }
53
54     $dataArray = $connection->getUser($_SESSION['mail']);
55     require 'application/views/_templates/header.php';
56     require 'application/views/admin/Static/header.php';
57     require 'application/views/admin/learners.php';
58 }
59

```

Questa è la funzione che apre le pagine per mostrare le informazioni degli allievi.

```

63 public function corsi(){
64     //Richiedo la connessione
65     require_once 'application/models/connection.php';
66     $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
67
68     if (!isset($_SESSION['mail'])) {
69         header("location: ". URL);
70     }else{
71         //Prendo tutti i valori dell'utente
72         $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
73
74         //controllo se ha fatto il login un coach
75         if(!strcmp($userData, "allievo")){
76             header("location: ". URL ."home/user/");
77         }else if(!strcmp($userData, "coach")){
78             header("location: ". URL ."home/coach/");
79         }
80     }
81
82     $dataArray = $connection->getUser($_SESSION['mail']);
83     require 'application/views/_templates/header.php';
84     require 'application/views/admin/Static/header.php';
85     require 'application/views/admin/courses.php';
86 }

```

Questa è la funzione che apre tutto il necessario per mostrare la pagina di tutti i corsi.

Una volta finito con la creazione dell'apertura delle pagine ho creato la grafica delle pagine, essendo che devono essere tutte molto simili.

Progetto Eserciziario - Coach

Amministratore ▾

Coach

Nome | Cognome | email | Indirizzo (Via, CAP Citta)

Dyuman | Bulloni | dyuman.bulloni@samtrevano.ch | via bassora 13, 6834 Morbio inferiore

Insegna

Questa è la prima pagina che appare al login dell'amministratore, la pagina che mostra tutti i coach e permette di accederci e modificarla, l'intestazione delle pagine è la stessa per tutte l'unica differenza è il titolo che varia a dipendenza della pagina aperta.

Allievi

Nome | Cognome | email | Indirizzo (Via, CAP Citta)

gino | pino | dragon.morso@gmail.com | via impossibile 2, 1234 Bella  
Giairo | Mauro | giairo.mauro@samtrevano.ch | Via besso 41, 6900 Lugano

Insegna

Questa è la pagina che mostra tutti gli allievi registrati e da la possibilità d accedervi per modificare le impostazioni o aggiungere corsi.

Corsi

Materia | giorno | ora | crediti | numero massimo allievi

Economia e diritto | lunedì | 8 | 0 | 2  
Italiano MP | lunedì | 17 | 0 | 1  
Italiano MP | lunedì | 17 | 0 | 1  
Educazione fisica | lunedì | 10 | 0 | 3

Submit

Questa è la pagina che mostra tutti i corsi esistenti.

```

13 function getUsers(type){
14     var select = document.getElementById(type);
15
16     var length = select.length;
17     //For che cancella dati già inseriti nel select
18     for(var i = 0; i <= length + 1; i++){
19         select.remove(0);
20     }
21     xhttp.onreadystatechange = function() {
22         if (this.readyState === 4 && this.status === 200) {
23
24             //Prendo i valori passati dal server e li metto in un array
25             var obj = JSON.parse(xhttp.responseText);
26
27             //Inserisco tutte le categorie in optio e poi nel select.
28             for(var i = 0; i < obj.length; i++){
29                 var option = document.createElement("option");
30                 option.value = obj[i]['email'];
31                 option.text = obj[i]['nome'] + " | " + obj[i]['cognome'] + " | " + obj[i]['email'] + " | " + obj[i]['via'] + " | " + obj[i]['CAP'] + " | " + obj[i]['citta'];
32                 select.appendChild(option);
33             }
34         }
35     };
36     xhttp.open("POST", "http://samtinfo.ch/gestrip/admin/getUsers", true);
37     xhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
38     xhttp.send("&type="+ type);

```

Questa è la funzione presente nel file javascript che viene richiamata dalle pagine “corsi” e “coach” dopo aver creato tutto il resto, si occupa di, con l’ausilio di ajax, fare una richiesta http alla funzione che prende tutti gli utenti e le relative informazioni, in base al tipo che gli viene passato tramite il POST.

```

92 public function getUsers(){
93     //Richiedo la connessione
94     require_once 'application/models/connection.php';
95     $connection = new Connection("efof.myd.infomaniak.com", "efof_gestrip18", "GestRip_Admin2018", "efof_gestrip_2018");
96
97     if (!isset($_SESSION['mail'])) {
98         header("location: ". URL);
99     }else{
100         //Prendo tutti i valori dell'utente
101         $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
102
103         //controlla se ha fatto il login un coach
104         if(!strcmp($userData, "allievo")){
105             header("location: ". URL . "home/user/");
106         }else if(!strcmp($userData, "coach")){
107             header("location: ". URL . "home/coach/");
108         }
109     }
110
111     $coachList;
112
113     //Prendo i dati in entrata dal form
114     if ($_SERVER["REQUEST_METHOD"] == "POST") {
115         $type = $_POST['type'];
116     }
117
118     //Controlla che ci sia un valore e richiamo la funzione che prede i dati
119     if($type != ''){
120         $coachList = $connection->getUser($_SESSION['mail'], null, $type);
121     }
122
123     //Stampo con json i valori dei coach
124     header('Content-Type: application/json');
125     echo json_encode($coachList);

```

Questa funzione viene richiamata dal POST nella funziona JavaScript mostrata sopra, e si occupa semplicemente di controllare chi ci accede e prendere i dati degli utenti che prende dal database la funzione in “Connection” codificarli in JSON e stamparli, in modo che Ajax riesca ad interpretarli come visto precedentemente.



```

189 public function getUser($mail, $data = null, $type = null){
190     //Connetto al database
191     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
192
193     $dataArray = false;
194
195     //Controllo se prendese un tipo di utente specifico
196     if($type == null){
197         //Controllo se prendere tutti i dati o uno specifico
198         if($data == null){
199             //Prendo i dati dell'utente in base alla mail
200             $sql = $conn->prepare("SELECT * FROM utente WHERE email = ?");
201         }else{
202             //Prendo il dao richiesto dell'utente in base alla mail
203             $sql = $conn->prepare("SELECT $data FROM utente WHERE email = ?");
204
205         }
206
207         $sql->bind_param("s", $mail);
208     }else{
209         //Controllo se prendere tutti i dati o uno specifico
210         if($data == null){
211             //Prendo i dati dell'utente in base alla mail
212             $sql = $conn->prepare("SELECT * FROM utente WHERE nome_tipo = ?");
213         }else{
214             //Prendo il dao richiesto dell'utente in base alla mail
215             $sql = $conn->prepare("SELECT $data FROM utente WHERE nome_tipo = ?");
216
217         }
218
219         $sql->bind_param("s", $type);
220     }
221
222 }

```

Nella classe connection per prendere gli utenti ho semplicemente modificato la già esistente funzione "getUser" in modo che controllasse se viene passato un tipo, in caso positivo fa la "SELECT" in base ad esso invece che in base alla email.

```

45 function getCourses(){
46     var courses = document.getElementById("courses");
47
48     var length = courses.length;
49     //For che cancella dati già inseriti nel select
50     for(var i = 0; i <= length + 1; i++){
51         courses.remove(0);
52     }
53     xhttp.onreadystatechange = function() {
54         if (this.readyState === 4 && this.status === 200) {
55             //Preno i valori passati dal server e li metto in un array
56             var obj = JSON.parse(xhttp.responseText);
57
58             //Inserisco tutte le categorie in optio e poi nel select.
59             for(var i = 0; i < obj.length; i++){
60                 var option = document.createElement("option");
61                 option.value = obj[i]['id_corso'];
62                 option.text = obj[i]['nome_materia'] + " | " + obj[i]['giorno'] + " | " + obj[i]['ora'] + " | " + obj[i]['crediti'] + " | " + obj[i]['num_allievi'];
63                 courses.appendChild(option);
64             }
65         }
66     };
67     xhttp.open("POST", "http://saminfo.ch/gestrip/admin/getCourses", true);
68     xhttp.send();
69 }

```

Questa funzione di JavaScript fa la stessa cosa dell'altra ma prende i corsi e imposta in modo diverso i dati che inserisce all'interno del select, perché ne riceve altri, e in più non manda niente quando fa il "send()".

```

130 public function getCourses(){
131     //Richiedo la connessione
132     require_once 'application/models/connection.php';
133     $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
134
135     if (!isset($_SESSION['mail'])) {
136         header("location: ". URL);
137     }else{
138         //Prendo tutti i valori dell'utente
139         $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
140
141         //controlla se ha fatto il login un coach
142         if(!strcmp($userData, "allievo")){
143             header("location: ". URL ."home/user/");
144         }else if(!strcmp($userData, "coach")){
145             header("location: ". URL ."home/coach/");
146         }
147     }
148
149     $courses = $connection->getCourses();
150
151     //Stampo con json i valori dei coach
152     header('Content-Type: application/json');
153     echo json_encode($courses);
154 }
155
156 function openCoach(){
157     //Prendo i dati in entrata dal form
158     if ($_SERVER["REQUEST_METHOD"] == "POST") {
159         $coach = $_POST['coach'];
160     }
161
162     //Setto la variabile della sessione del coach che andrò a controllare
163     $_SESSION['coach'] = $coach;
164
165 }
166
167 }

```

Questa è la funzione della model dell'amministratore che si occupa di prendere i dati dalla funzione nella classe "Connection" di tutti i corsi esistenti.

```

594 public function getCourses(){
595     //Connect to database
596     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
597
598     //Get roles
599     $sql = $conn->prepare("SELECT * FROM corso");
600     $courses = array();
601
602     if($sql->execute()){
603         $result = $sql->get_result();
604         while ($row = $result->fetch_assoc()) {
605             array_push($courses, $row);
606         }
607     }
608     $conn->close();
609     return $courses;
610 }

```

Questa funzione è quella che si occupa di prendere tutti i dati di tutti i corsi e inserirli all'interno di un array per poi ritornarli.

#### Problemi riscontrati e soluzioni adottate

Un primo problema che ho riscontrato è stato nella pagina della richiesta dei corsi, perché nel controllo dell'input numerico non veniva controllato che non c'erano scritte lettere, come si poteva fare in Edge, questo perché il nome della funzione dichiarata al click di un bottone all'interno dell'input e il nome della funzione effettiva erano diversi, per mettere a posto questo mi è bastato scriverle uguali.

Mentre mettevo la visualizzazione delle categorie all'interno della pagina di creazione dei corsi ho incontrato un errore, dato dai controlli che facevo, in modo che poteva accedere alla funzione per

prendere le categorie solo l'allievo, mentre dovrebbero poterci accedere tutti, quindi una volta tolto quel controllo tutto è tornato a posto.

```

130 public function getCategory(){
131     //Setto le variabili per accedere al database.
132     require_once 'application/models/connection.php';
133     $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
134
135     //Variabile che contera i valori delle categorie
136     $categories = "";
137     $subject = "";
138
139     //Prendo i dati in entrata dal form
140     if ($_SERVER["REQUEST_METHOD"] == "POST") {
141         $subject = $_POST['subject'];
142     }
143
144     //Controllo che ci sia un valore e richiamo la funzione che prede i dati
145     if($subject != ''){
146         $categories = $connection->getCategories($subject);
147     }
148
149     header('Content-Type: application/json');
150     echo json_encode($categories);
151 }

```

Questa è la funzione attuale che prende le categorie dalla funzione che si collega al database, senza controlli.

Ho avuto alcuni problemi con JSON, quando dovevo parsare ciò che mi ritornava il server in ajax, perché non dovevo fare lo stesso passaggio dell'altro ma uno in meno essendo che passavo già un array, e mi bastava parsarlo e non ritrasformarlo in un array.

#### Punto della situazione rispetto alla pianificazione

Sono in linea con il gantt, ho fatto alcune cose oggi che avrei dovuto iniziare più avanti ma ho spostato altre cose per le prossime lezioni.

#### Programma di massima per la prossima giornata di lavoro

Iniziare l'implementazione della connessione ai coach dell'amministratore.