

# Diario di lavoro

|       |            |
|-------|------------|
| Luogo | SAMT       |
| Data  | 17.10.2018 |

## Lavori svolti

Appena iniziato sta mattina la prima cosa che ho fatto è stata fare il link di disconnessione per gli utenti, è stato abbastanza veloce, mi è bastato aggiungere la voce nei link dell'utente, e dopodiché dargli il riferimento ad una funzione controller.

```

1 <?php
2 /**
3  *
4  */
5 class Logout
6 {
7     public function index()
8     {
9         session_destroy();
10        header("location: ". URL);
11    }
12 }

```

Questa è la classe creata appositamente per definire il logout, che implementa semplicemente una funzione che distrugge la sessione corrente e direziona alla pagina iniziale del sito.

Una volta finito questo ho iniziato a mettere le basi per un select che mostra tutte le categorie quando viene selezionata una materia dal select. Ho pensato di utilizzare ajax e fare la richiesta al database tramite una richiesta, ma ho riscontrato un problema nell'invio dei dati tramite POST nel metodo "send()" di conseguenza ho messo in pausa questa parte, chiederò al professore quando sarà presente.

Per il momento ho fatto la pagina "mostra statistiche".

## Tabella corsi insegnati

| id corso | Materia             | giorno    | ora | crediti | numero massimo allievi |
|----------|---------------------|-----------|-----|---------|------------------------|
| 9        | Etica professionale | martedì   | 9   | 0       | 0                      |
| 10       | Etica Professionale | Lunedì    | 3   | 3       | 2                      |
| 14       | Etica professionale | mercoledì | 16  | 0       | 1                      |
| 15       | Italiano MP         | venerdì   | 12  | 0       | 1                      |
| 17       | Italiano MP         | giovedì   | 10  | 0       | 1                      |
| 19       | Etica professionale | giovedì   | 11  | 2       | 1                      |

Questa è la parte grafica che mostra tutti i corsi che un docente insegna.

Per questa visualizzazione ho optato per una tabella invece di un select, perché l'utente non deve modificare niente in questa pagina, di conseguenza questo è il modo migliore per mostrare e basta tutti i dati.

Per prendere i dati richiamo la solita funzione controller che controlla se sia giusto il richiamo e dopodiché in caso positivo richiama la funzione in "Connection" che prende i dati.

```

368 public function getTaughtCourses(){
369     //Connetto al database
370     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
371
372     //Prendo i dati dell'utente in base alla mail
373     $sql = $conn->prepare("SELECT * FROM corso WHERE mail_insegnante = ?");
374     $sql->bind_param("s", $_SESSION['mail']);
375
376     $courses = array();
377     if ($sql->execute()) {
378         $result = $sql->get_result();
379         while ($row = $result->fetch_assoc()) {
380             array_push($courses, $row);
381         }
382     } else {
383         return "Error: <br>" . $conn->error; // " . $sql . "
384     }
385     return $courses;
386 }

```

Questa è la funzione nella classe "Connection" che va a prendere tutti i corsi nel database in cui la mail è la stessa dell'utente corrente, poi inserisce tutti i dati nell'array e lo ritorna, dopodiché il ritorno viene preso nella pagina e i dati inseriti nella tabella come spiegato sopra.

Questa stessa cosa l'ho fatta nella pagina dell'allievo con una differenza nella funzione nella classe di connessione.

```

396     //Prendo i dati dell'utente in base alla mail
397     $sql = $conn->prepare("SELECT id_corso FROM richiede WHERE mail_utente = ?");
398     $sql->bind_param("s", $_SESSION['mail']);
399
400     $learnedCourses = array();
401     if ($sql->execute()) {
402         $result = $sql->get_result();
403         while ($row = $result->fetch_assoc()) {
404             array_push($learnedCourses, $row['id_corso']);
405         }
406     }
407
408     $courses = array();
409     for($i = 0; $i < count($learnedCourses); $i++){
410         //Prendo i dati dell'utente in base alla mail
411         $sql = $conn->prepare("SELECT * FROM corso WHERE id_corso = ?");
412         $sql->bind_param("s", $learnedCourses[$i]);
413
414         if ($sql->execute()) {
415             $result = $sql->get_result();
416             while ($row = $result->fetch_assoc()) {
417                 array_push($courses, $row);
418             }
419         }
420     }

```

Questa è la parte diversa rispetto all'altra funzione, come si può vedere non prende direttamente i corsi ma prima va a vedere gli id a dei corsi a cui l'allievo è iscritto e poi prende i corsi con gli id trovati precedentemente.

Durante lo svolgimento di questo mi sono accorto che c'era un problema nella richiesta dei corsi da parte degli allievi, ovvero che quando avviene l'inserimento del corso lui non viene inserito tra quelli che vogliono farlo. Allora ho cambiato l'inserimento dei nuovi corsi, per le richieste ho creato una nuova funzione.

In questa funzione c'è l'inserimento dell'allievo in un determinato corso nella tabella ponte "richiede" e anche un controllo che effettivamente ci siano ancora post liberi in quella documentazione per inserirlo.

Nelle seguenti immagini viene spiegata parte per parte la funzione nel modo in cui il codice è scritto

```

316 if ($sql->execute()) {
317     //Inizio a prendere gli id del corso
318     //Prendo l'id del corso in base ai dati precedentemente inseriti
319     $sql = $conn->prepare("SELECT id_corso FROM corso WHERE giorno = ? AND ora = ? AND num_allievi = ? AND nome_materia = ?");
320     $sql->bind_param("sis", $day, $hour, $learners, $subject);
321
322     $learnedCourses = array();
323     if ($sql->execute()) {
324         $result = $sql->get_result();
325         while ($row = $result->fetch_assoc()) {
326             array_push($learnedCourses, $row['id_corso']);
327         }
328     }

```

Questa è la prima parte, importante, della funzione, in cui, dopo aver inserito il corso prendo tutti gli id dei corsi con le credenziali inseriti dall'utente, questo perché magari esiste anche un altro corso con le stesse caratteristiche che l'utente non ha visto e se c'è posto viene iscritto anche a quello.

```

329
330     //Controllo se il numero di allievi richiesto per il corso non è stato superato.
331     $idCourses = array();
332     for($i = 0; $i < count($learnedCourses); $i++){
333         //Prendo i valori dalla tabella "richiede"
334         $sql = $conn->prepare("SELECT * FROM richiede WHERE id_corso = ?");
335         $sql->bind_param("i", $learnedCourses[$i]);
336
337         if ($sql->execute()) {
338             $result = $sql->get_result();
339             while ($row = $result->fetch_assoc()) {
340                 array_push($idCourses, $row);
341             }

```

Dopo aver preso i dati parte un for che rende il numero di utenti già registrati a quel corso.

```

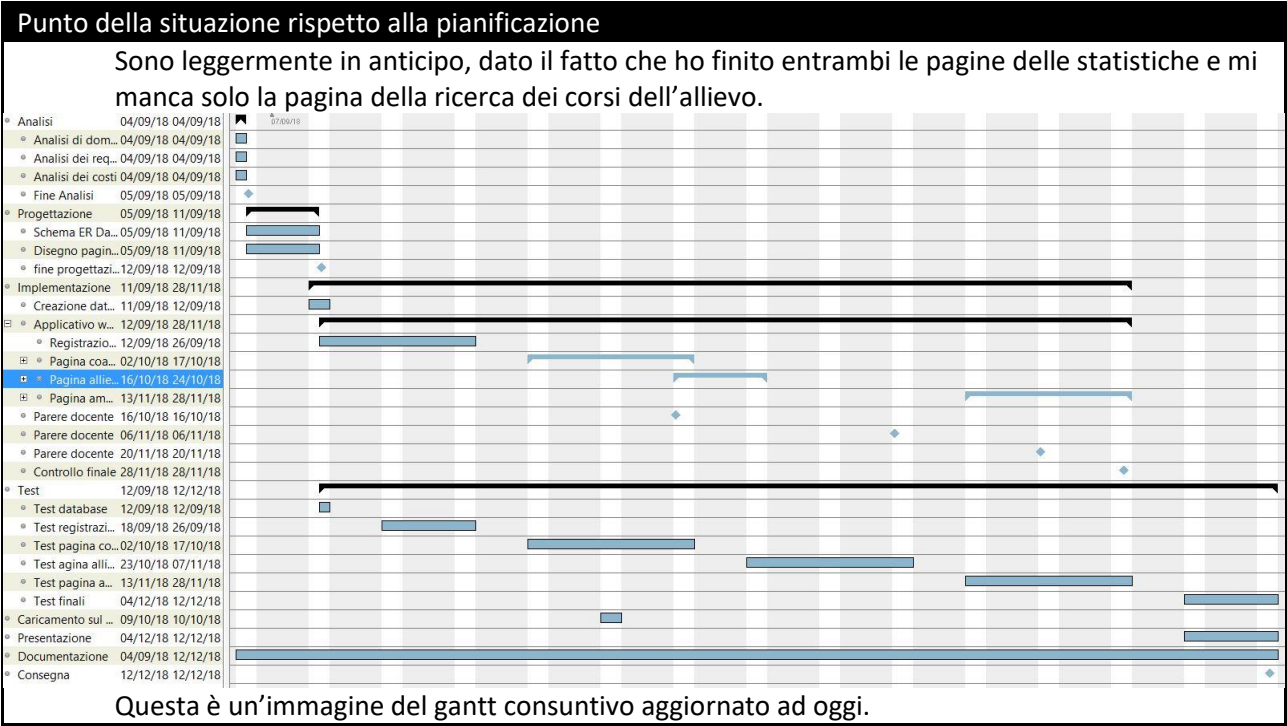
338         $result = $sql->get_result();
339         while ($row = $result->fetch_assoc()) {
340             array_push($idCourses, $row);
341         }
342
343         //Se gli utenti iscritti sono meno degli utenti possibili inserisce l'utente
344         if(count($idCourses) < $learners){
345             $sql = $conn->prepare("INSERT INTO richiede(mail_utente, id_corso) VALUES(?, ?)");
346             $sql->bind_param("si", $_SESSION['mail'], $learnedCourses[$i]);
347
348             if ($sql->execute()) {
349                 return "Inserimento corso riuscito";
350             }else{
351                 return "Errore nll'inserimento dei dati";
352             }
353         }else{
354             return "Errore nll'inserimento dei dati";
355         }
356     }
357
358 } else {
359     return "Errore nll'inserimento dei dati";
360 }
361

```

Una volta preso il numero di utenti registrati viene controllato che il corso non sia pieno, se non lo è viene inserito l'utente corrente nella tabella 'richiede' con l'id del corso che si sta controllando. Questo procedimento viene poi ripassato per tutti i corsi uguali, se ce ne dovessero essere.

#### Problemi riscontrati e soluzioni adottate

Non ho avuto particolari problemi, o almeno non diversi dagli stessi avuti nei giorni precedenti con l'inserimento dei dati nel database, ma nessun errore che mi ha preso troppo tempo. Ho dovuto aggiustare alcune funzioni, perché non erano abbastanza performanti o c'erano troppe informazioni ed era più conveniente dividerla, come ad esempio la funzione per la creazione dei corsi, come spiegato nella sezione precedente.



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

Finire la pagina di ricerca dei corsi dell'utente.