

Diario di lavoro

Luogo	SAMT
Data	14.11.2018

Lavori svolti

Questa mattina prima di tutto ho iniziato creando l'input e il bottone che permettono all'utente di pagare un determinato corso.

Tabella corsi insegnati

numero	Materia	giorno	ora	crediti	numero massimo allievi
34	Economia e diritto	lunedì	8:00	0	2
36	Italiano MP	lunedì	17:00	0	1
103	Inglese MP	lunedì	9:00	0	2
104	Italiano MP	lunedì	9:00	0	2
119	Italiano MP	mercoledì	10:00	0	7

Numero corso da pagare

Paga corso

Nella casella di input l'utente dovrà inserire il numero del corso che vuole pagare, è presente il solito controllo che si assicura che l'utente non scrive altro che numeri e se c'è qualcosa di diverso cambia la scritta in rosso e non si attiva, o si disattiva, il bottone per pagare.

Mentre implementavo il codice mi sono accorto che in molte funzioni nella classe connection ripeto il controllo per vedere se un utente è già iscritto ad un corso e non utilizzavo la funzione già esistente che controllava proprio questo, allora ho modificato i codici in modo che richiamassero quella funzione al posto di rifare sempre lo stesso controllo.

```

478 public function searchCourses($subject, $mail){
479     //Array da riempire con le materie
480     $categories = array();
481
482     //Prendo un array con tutti i corsi che impara l'allievo
483     $learnedCourse = $this->getLearnedCourses($mail);
484
485     //For che controlla tutti i valori
486     for($i = 0; $i < count($learnedCourse); $i++){
487         if(!strcmp($learnedCourse[$i]['nome_materia'], $subject)){
488             array_push($categories, $learnedCourse[$i]);
489         }
490     }
491
492
493     return $categories;
494 }

```

Questa funzione a differenza di com'era prima c'è molto meno codice e semplicemente prende tutti i corsi a cui l'utente è iscritto e controlla tutti quelli della materia richiesta e li ritorna in un array, tutto ciò che ci sta dietro, nel file JavaScript che fa la richiesta e nel controller, rimane uguale.

Dopo aver cambiato le impostazioni ho creato la funzione nella classe "Connection" che va a cambiare i dati dei crediti degli utenti all'interno del database una volta che l'allievo paga.

```

711     for($i = 0; $i < count($learnedCourse); $i++){
712         if(!strcmp($learnedCourse[$i]['id_corso'], $idCourse)){
713
714             //Prendo i dati del corso richiesto
715             $course = $this->getCourses($idCourse);
716
717             //Prendo i crediti attuali dell'allievo
718             $creditsL = $this->getUser($mail, 'crediti');
719
720             //Tolgo i crediti da pagare
721             $creditsL -= $course[0]['crediti'];
722
723             //Modifico i crediti dell'allievo
724             $sql = $conn->prepare("UPDATE utente SET crediti = ? WHERE email LIKE ?");
725
726             $sql->bind_param("is", $creditsL, $mail);
727
728             $sql->execute();
729
730             //Se al corso è assegnato un docente
731             if(strcmp($course[0]['mail_insegnante'], "")){
732                 //prendo crediti del docente
733                 $creditsT = $this->getUser($course[0]['mail_insegnante'], 'crediti');
734
735                 //Aggiungo i crediti pagati
736                 $creditsT = $course[0]['crediti'];
737
738                 //Modifico i crediti dell'allievo
739                 $sql = $conn->prepare("UPDATE utente SET crediti = ? WHERE email LIKE ?");
740
741                 $sql->bind_param("is", $creditsT, $course[0]['mail_insegnante']);
742
743                 if($sql->execute()){
744                     echo "Funziona";
745                 }else{
746                     echo "Non Funziona";
747                 }
748             }
749         }
750     }

```

Questa è la parte centrale della funzione, dopo aver creato la connessione con il database e aver preso tutti i corsi a cui l'utente è iscritto grazie alla funzione apposita, e cicla tutti i corsi alla ricerca del corso che l'utente vuole pagare, se non lo trova mai allora vuol dire che non è iscritto e non deve fare nulla, in caso lo trova prende i crediti attuali dell'utente e li diminuisce in base a quanti ne deve pagare, e se al corso è iscritto un insegnante allora aumenta i suoi crediti, questo perché è possibile che ci sia un corso a cui l'utente è iscritto ma non c'è il docente ma lui deve pagarlo lo stesso, anche se il docente è uscito prima di ricevere il pagamento.

La prima ora del pomeriggio, abbiamo dovuto fare un controllo sulla rete Wi-Fi nuova della scuola, quindi non ho potuto usare il mio tempo per portare avanti il progetto, di conseguenza l'ultima ora sono andato avanti con la documentazione, essendo che le parti fondamentali del progetto erano ormai concluse.

Problemi riscontrati e soluzioni adottate

Questa lezione non ho riscontrato particolari problemi in ciò che ho fatto

Punto della situazione rispetto alla pianificazione

Sono in linea con il gantt.

Nome Progetto: Gestione ripetizioni

Programma di massima per la prossima giornata di lavoro

Portare avanti la documentazione, Test e implementazione.