

Diario di lavoro

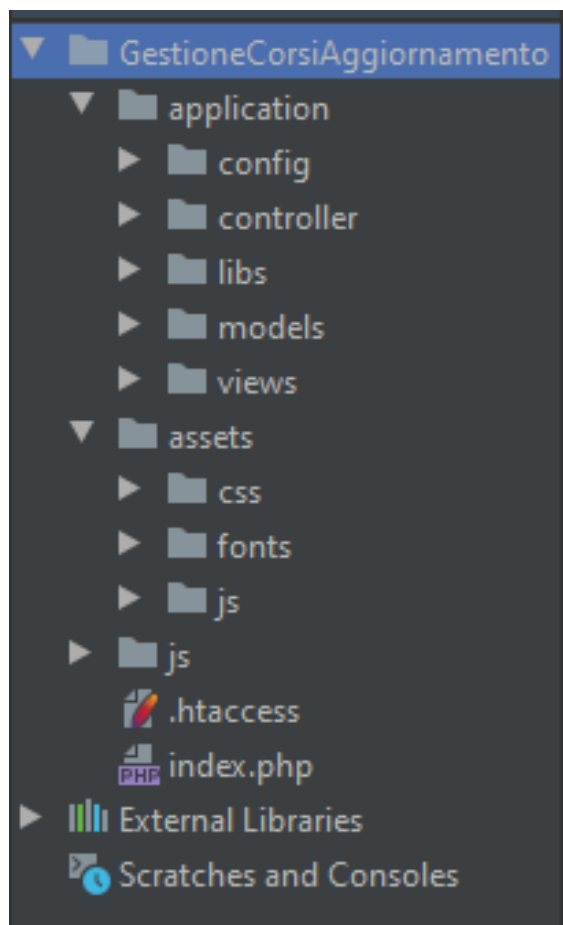
Luogo	Mendrisio
Data	13.05.2020

Lavori svolti

Durante le prime due ore della giornata di oggi ho concluso il capitolo 3.3 Design delle interfacce, in pratica ho fatto e finito altre 4 interfacce che vanno ad aggiungersi alle 6 di ieri. Le interfacce sono:

- Pagina ricerca corsi
- Pagina di iscrizione ad un corso
- Pagina gestione corsi
- Pagina personale

Una volta finita la progettazione ho scritto il codice SQL del database e ho incominciato a creare la cartella che conterrà tutti il sito web. Per la creazione del sito utilizzo un template MVC che ci è stato fornito dal docente Massimo Sartori.



Creata la cartella ho incominciato a sviluppare le interfacce di login. Per la creazione di questo sito utilizzo questo template: <https://github.com/puikinsh/srtdash-admin-dashboard>. Le interfacce che ho creato oggi sono queste:

The image displays five distinct user interface cards for a web application, arranged in two rows. Each card has a purple header with white text and a white body with teal accents.

- LOGIN:** Header "LOGIN". Subtext "Ciao, inserisci i tuoi dati per accedere al sito!". Fields for "Email" and "Password" with icons. A link "Password dimenticata?". A teal button "Accedi".
- EMAIL INVIATA!:** Header "EMAIL INVIATA!". Text: "Ti è stata inviata una email per cambiare la password del tuo account! Se non trovi la email controlla se non è finita nella cartella di spam." A teal button "Ritorna al login".
- PASSWORD DIMENTICATA?:** Header "PASSWORD DIMENTICATA?". Subtext: "Inserisci la tua email e ti invieremo un link di recupero password!". Field for "Email" with an icon. A teal button "Invia".
- CAMBIA PASSWORD:** Header "CAMBIA PASSWORD". Text: "Inserisci una nuova password, ripetila e poi premi Cambia. Dopodiché potrai utilizzare la nuova password per accedere. La password deve avere almeno 8 caratteri e numero/carattere speciale." Fields for "Password" and "Ripeti password" with icons. A teal button "Cambia →".
- EMAIL INVIATA!:** Header "EMAIL INVIATA!". Text: "Ti è stata inviata una email per cambiare la password del tuo account! Se non trovi la email controlla se non è finita nella cartella di spam." A teal button "Ritorna al login".

Una volta fatte queste interfacce grafiche ho fatto una chiamata con il mio formatore e il mio perito. Durante questa chiamata ho dovuto modificare leggermente il database. Lo schema logico è diventato il seguente:

```

user(id, email, firstname, lastname, birthday, zip, city, street, mobile_number, flag_newsletter*,
landline_number*, password*, token*, type*, nip*, license_number*)
typology(name)
photo(path)
course(id, title, description, zip, street, city, max_participants, materials, name_typology (FK), meal_price,
course_price)
execution(id, id_user (FK), id_course (FK))
lesson(start_date, id_execution (FK), end_date)
enrolls(id_user (FK), id_execution(FK), intolerances*, food_type*, flag_meal)
settings(iban_number, registration_deadline, beneficiary, day_deadline)

```

Una volta modificato e-r, schema logico e descrizioni, ho incominciato a fare il back-end del login. Il metodo che si occupa di verificare che i dati inseriti siano corrispondenti ad un utente è il seguente:

```
/**
 * Metodo per effettuare il Login.
 */
public function access($email, $password)
{
    $result = $this->getUser($email);
    if (count($result) > 0) {
        if (password_verify($password, $result[0][DB_USER_PASSWORD])) {
            echo SUCCESSFUL;
            return $result;
        } else {
            echo LOGIN_DENY;
        }
    } else {
        echo LOGIN_DENY;
    }
}
```

Questo metodo si occupa di cercare l'utente nel database e solo in caso che la email inserita corrisponde a un utente verifica che la password sia corretta.

Per eseguire delle query al database utilizzo la libreria "Meekrodb" che mi permette di eseguire delle query senza perdere troppo tempo per scrivere i prepare statement.

Un esempio è il seguente:

```
$selectAccess = "SELECT * FROM user WHERE email=%s && type <> 0";
return $this->connection->query($selectAccess, $email);
```

Il metodo query prende come primo parametro la query da eseguire nel database e tutti gli altri parametri sono i valori da inserire nella query. In questo caso \$email sostituisce il valore %s che specifica una stringa.

Infine ho incominciato a creare la procedura di recupera password. Quando un utente non ricorda la propria password gli basta cliccare il link "Hai dimenticato la password?". A questo punto inserisce la sua email e gli viene inviata questa email:

Modifica la password

Da: Gestione Corsi Aggiornamento <gestionecorsiaggiornamento@gmail.com>

data: 13/05/2020 17:08

Ciao Mattia Toscanelli,
Recentemente è stata richiesta la procedura di modifica password!

[Per modificare la tua password clicca questo link!](#)

Per fare questo utilizzo questo spezco di codice:

```
$hash = hash('sha256', random_bytes(16) . $this->email);
$updateUsers = "UPDATE user SET token=%s WHERE email=%s";
require 'application/libs/sendMail.php';
$body = "Ciao " . $result[0][DB_USER_NAME] . " " . $result[0][DB_USER_SURNAME] . ",<br>
Recentemente è stata richiesta la procedura di modifica password!<br><br>
<a href='" . URL . "resetPassword/changePassword/$hash/$this->email'> Per
modificare la tua password clicca questo link!</a>";
```

```
try {  
    $s = new SendMail();  
    $s->mailSend($this->email, "Modifica la password", $body);  
    $this->connection->query($updateUsers, $hash, $this->email);  
    return true;  
} catch (Exception $e) {  
    return false;  
}
```

In questo spezco di codice creo un valore random da 16 byte e genero la hash in sha256 e la aggiungo all'url di recupero password. Inoltre aggiungo la email dell'utente per identificare l'utente che ha richiesto la password. Una volta preparato il link invio la email grazie al metodo mailSend(), dove passo e-mail del destinatario, header e contenuto. Per inviare le email utilizzo la classe PHPMailer. Il codice che ho scritto oggi è preso in gran parte dal progetto scorso.

Problemi riscontrati e soluzioni adottate

Durante la lezione di oggi ho avuto un piccolo problema nel collegarmi al database. Infatti quando cercavo di accedere al database da PHP mi ritornava il seguente errore: "The server requested authentication method unknown to the client". Per risolvere questo problema ho dovuto eliminare l'utente dal database e re-inserirlo con il seguente comando:

```
CREATE USER 'courses_management_admin'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'CoursesManagement&1';
```

Inoltre ho aggiunto due righe al fine my.cnf di MySQL server:

```
[mysq]  
default-authentication-plugin=mysql_native_password
```

Punto della situazione rispetto alla pianificazione

Sono in linea rispetto la mia pianificazione in quanto oggi avrei dovuto finire la pagina di login e iniziare la pagina di recupero password.

Programma di massima per la prossima giornata di lavoro

La prossima giornata di lavoro farò finirò la pagina di recupero password e inizierò la gestione della pagina principale.