



UNIVERSITÀ DEGLI STUDI GUGLIELMO MARCONI

## Sistemi e tecnologie web Relazione

“Virtual Campus”

## Sommario

Progetto assegnato .....	3
Scopo del documento .....	3
Funzioni generali del prodotto .....	3
Tecnologie utilizzate .....	3
Struttura progetto .....	4
Pages .....	4
courseDetail.php .....	4
courses.php .....	4
createCourse.php .....	4
createExam.php .....	4
email.php .....	5
emailList.php .....	5
emailMenu.php .....	5
emailView.php .....	5
home.php .....	5
login.php .....	6
register.php .....	6
reservation.php .....	6
userDetail.php .....	6
users.php .....	6
welcome.php .....	6
Styles .....	6
Util .....	7
cookie.php .....	7
dbConnection.php .....	7
index.php .....	7
Struttura database .....	7
courses .....	8
emails .....	8
exams .....	8
results .....	8
users .....	8
userdetail .....	8
Problematiche riscontrate .....	8
Memorizzazione sessione utente .....	8
Formato DATETIME .....	9

## Progetto assegnato

Implementare in PHP/MYSQL un software di comunicazione simile a virtual c@mpus con la possibilità però di registrarsi e con un sistema di notifiche via email. Preveda un pannello di controllo per gli amministratori e funzioni dedicate per ogni profilo.

## Scopo del documento

Lo scopo di questo documento è di illustrare le specifiche di funzionamento del prodotto, la sua struttura e la composizione della base di dati.

## Funzioni generali del prodotto

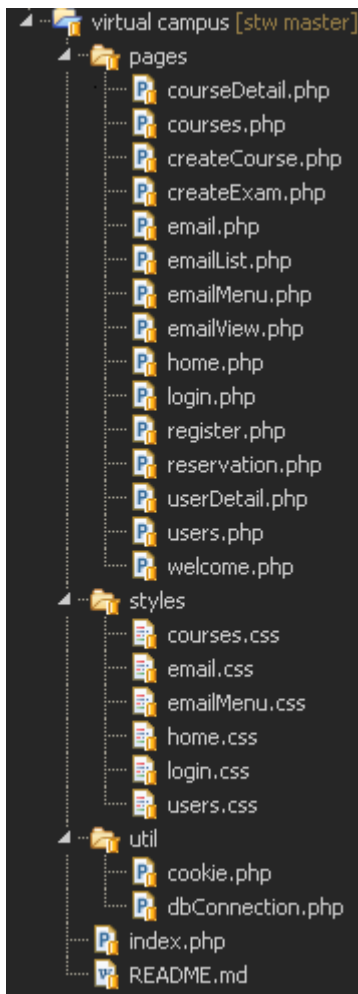
Il seguente prodotto è un semplice software gestionale fruibile via web per la gestione dell'utenza universitaria. Il software, quindi, dovrà permettere ai docenti di creare nuovi corsi, sessioni di esame e i voti per questi ultimi. Sarà anche integrato un rudimentale sistema di messaggistica e notifica eventi (come l'esito di un esame) simile a un client email, dove sarà possibile inviare messaggi ad altri utenti. Gli studenti avranno la possibilità di registrarsi con un nome e password. È necessario inoltre tenere uno storico per ogni utente di esami effettuati e i relativi risultati.

Sarà necessaria anche una suddivisione docenti/utenti per gestire i differenti ruoli all'interno del software, questa suddivisione sarà effettuata implementando account amministrativi. Questi account avranno la possibilità di creare corsi, esami e di assegnare un esito allo stesso, mentre gli account non amministrativi potranno solo iscriversi alle sessioni di esame e accettare l'esito.

## Tecnologie utilizzate

L'applicazione è scritta interamente in linguaggio PHP ad eccezione dell'oggetto datepicker che utilizza jquery (<https://jqueryui.com/datepicker>). L'impaginazione del sito è affidata a semplici file .css i quali gestiscono, oltre al layout, la colorazione dei vari pulsanti. La base di dati è stata realizzata con MySQL e PhpMyAdmin.

## Struttura progetto



## Pages

### *courseDetail.php*

Questa pagina, visualizzabile da un account studente, si occupa di visualizzare il corso selezionato e la sua descrizione.

### *courses.php*

Questa pagina, visualizzabile da un account studente, si occupa di creare una lista contenente i corsi abilitati per un certo utente.

### *createCourse.php*

Questa pagina, visualizzabile da un account amministratore, visualizza un form html attraverso il quale è possibile creare un nuovo corso e inserirlo nella base di dati.

### *createExam.php*

Questa pagina, visualizzabile da un account amministratore, permette di selezionare uno dei corsi disponibili e di creare per esso una sessione di esame. La sessione sarà salvata nella base di dati.

### *email.php*

Questa pagina crea l'impaginazione per la messaggistica interna. Più precisamente crea tre iFrame contenenti la lista delle email, la casella di visualizzazione della mail e il menu contenente le operazioni possibili.

### *emailList.php*

Questa pagina si occupa di visualizzare la lista delle email in entrata o in uscita, la casella è selezionata dal parametro passato alla pagina. Le email non lette saranno visualizzate in grassetto e non appena saranno aperte una query, si occuperà di marcare come letta l'email.

### *emailMenu.php*

Questa pagina si occupa di creare il menu con le operazioni disponibili per le email (nuovo messaggio, posta in arrivo e posta inviata). Accanto al pulsante "posta in arrivo" sarà visualizzato il numero di email in entrata non lette.

### *emailView.php*

Questa pagina ha tre modalità di funzionamento:

1. Viene selezionato dal menu "nuovo messaggio": In questa modalità viene visualizzato un form dove è possibile scegliere il destinatario della email, l'oggetto e il testo, per inviare basta premere sul pulsante "invia";
2. Posta in arrivo: Se dalla lista email in entrata viene selezionata una email, la pagina visualizzerà oggetto, mittente, ora di invio e testo. È possibile rispondere a una email cliccando su "rispondi", in questo caso sarà visualizzato un form come nel caso 1;
3. Posta inviata: Se dalla lista email inviate viene selezionata una email, la pagina visualizzerà oggetto, destinatario, ora di invio e testo.

### *home.php*

Questa pagina viene visualizzata subito dopo il login. Nella parte alta verrà visualizzato un messaggio di benvenuto seguito da nome e cognome dell'utente connesso, nella parte laterale sinistra verrà visualizzato il menu mentre nel restante spazio è presente un iFrame nel quale verranno caricate le pagine selezionate dal menu.

Se il login viene effettuato da un utente studente, la pagina visualizzerà un menu laterale così composto:

- Corsi disponibili
- Prenota esame
- Email
- Logout

Mentre se il login sarà effettuato da un account amministratore il menu laterale sarà così composto:

- Crea corso
- Crea esame
- Gestione utenti
- Email
- Logout

### *login.php*

Questa pagina presenta la possibilità di loggarsi al sito attraverso un form dove inserire nome utente e password, la pagina contiene anche un link alla pagina di registrazione.

### *register.php*

Questa pagina presenta un form attraverso il quale è possibile registrarsi come nuovo utente, il tipo di utente sarà di default “studente”.

### *reservation.php*

Questa pagina, visualizzabile da un account utente, visualizza due informazioni:

1. Prenota esame: in questo form è possibile visualizzare le sessioni di esame disponibili per i corsi abilitati all’utente. Accanto ad ogni sessione di esame è presente un pulsante “prenota”, disabilitato in caso di prenotazione effettuata.
2. Esami: in questo form è possibile visualizzare lo storico degli esami sostenuti ed il relativo stato. Nel caso in cui un esame passi dallo stato di “in attesa di esito” a quello di “corretto”, viene visualizzato il voto e due link che permettono di accettare o rifiutare l’esito.

### *userDetail.php*

Questa pagina, visualizzabile da un account amministratore, permette di effettuare due operazioni per l’utente selezionato:

1. Assegnazione corso: In questo form viene visualizzato un elenco di checkbox con tutti i corsi disponibili e sono selezionati quelli abilitati. È possibile operare sui checkbox per abilitare o disabilitare determinati corsi;
2. Esito esami: in questo form viene visualizzato un elenco contenente tutti gli esami effettuati dallo studente. È possibile modificare lo stato con l’apposito menu a tendina (non superato, in attesa di esito, rifiutato, voto).

### *users.php*

Questa pagina, visualizzabile da un account amministratore, crea una lista di tutti gli utenti non amministratori e un iFrame contenente la pagina userDetail.php. Quando un amministratore clicca il nome di un utente, questo viene passato come parametro alla pagina userDetail.

### *welcome.php*

Pagina iniziale contenente un messaggio di benvenuto, è visualizzata subito dopo il login

## **Styles**

In questa cartella sono contenuti i fogli di stile relativi ad alcune pagine php, il nome dei file .css sono uguali a quelli delle pagine .php.

## Util

### cookie.php

Questo file php contiene due funzioni di appoggio utilizzate dalle pagine per la gestione del cookie:

- *GetUserId()*: Ritorna l'id salvato nel database partendo dall'informazione "username" contenuta nel cookie;
- *set\_cookie(\$nome, \$cognome, \$username, \$isAdmin)*: Crea il cookie con i parametri forniti.

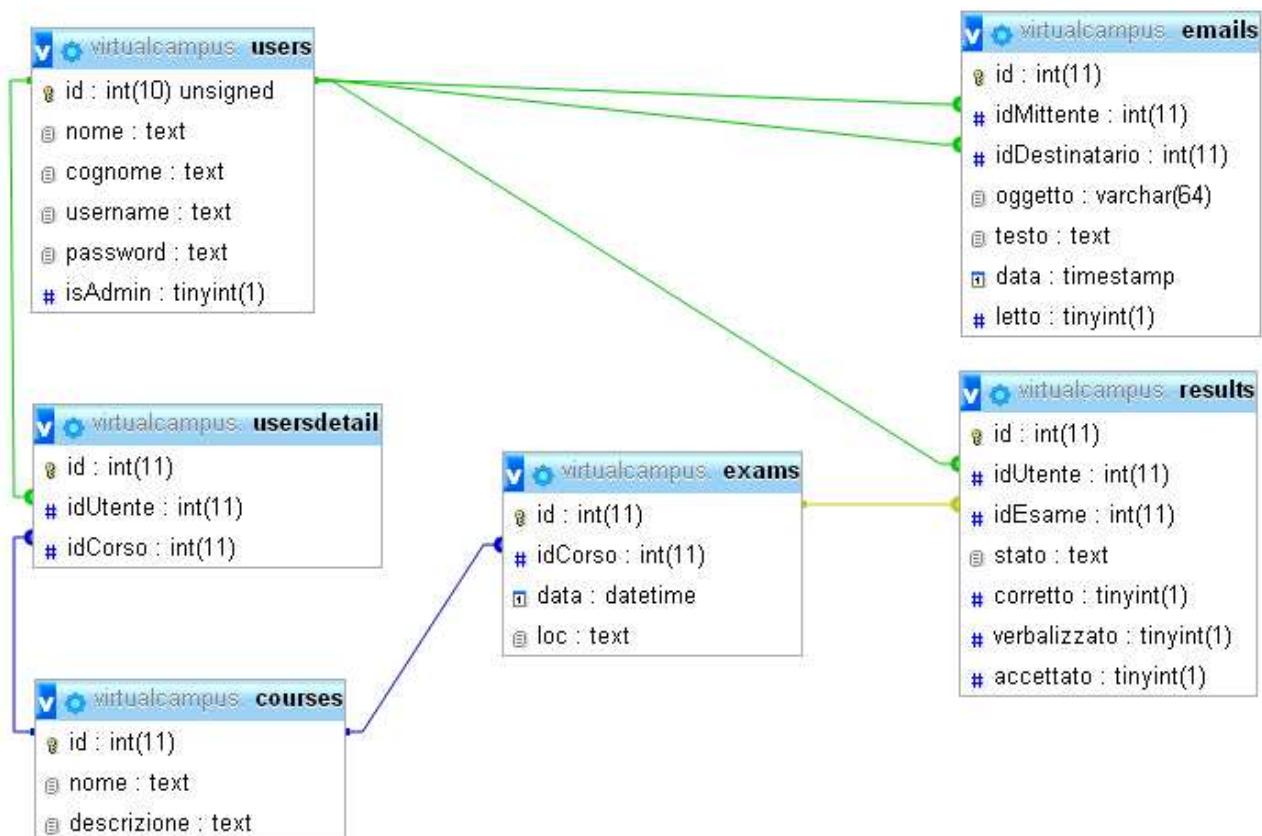
### dbConnection.php

Questo file php contiene una funzione *SendQuery(\$query)* che ha il compito di connettersi al database ed eseguire la query passata come parametro, ritorna il risultato della query.

### index.php

È la pagina principale caricata dal server quando viene richiesto il sito. Essa si preoccupa di controllare se è presente un cookie, se lo è allora si viene indirizzati alla pagina home.php, altrimenti alla pagina login.php.

## Struttura database



## **courses**

La tabella courses contiene le informazioni relative ai corsi creati, sono salvati nome del corso e descrizione.

## **emails**

La tabella emails contiene le email inviate dagli utenti. I campi idUtente e idDestinatario sono vincolati al campo id della tabella users. Oltre al contenuto dell'email è presente un campo booleano che indica se l'email è stata letta.

## **exams**

La tabella exams contiene le informazioni relative alle sessioni di esame create. Il campo idCorso è vincolato al campo id della tabella courses.

## **results**

La tabella results contiene le informazioni delle sessioni di esame prenotate dagli studenti. Il campo idUtente è vincolato al campo id della tabella users mentre il campo idEsame è vincolato al campo id della tabella exams.

## **users**

La tabella users contiene le informazioni relative agli utenti registrati.

## **userdetail**

La tabella userdetail permette di mettere in relazione un utente con i corsi abilitati per esso. Il campo idUtente è vincolato al campo id della tabella users mentre il campo idCorso è vincolato al campo id della tabella courses.

# **Problematiche riscontrate**

## **Memorizzazione sessione utente**

Per memorizzare la sessione utente ed agevolare il passaggio di informazioni tra le varie pagine per selezionare la modalità amministratore o studente è stato scelto di utilizzare un cookie chiamato “user”, della durata di 24 ore, contenente le seguenti informazioni:

1. nome utente (text)
2. cognome utente (text)
3. username (text)
4. admin (boolean)

La gestione del suddetto cookie è implementata nel file /util/cookie.php



## Formato DATETIME

Per adattare il formato della data ritornato dall'oggetto datepicker in jQuery per poterlo inserire nel campo di tipo DATETIME MySQL, sono state usate delle semplici istruzioni di manipolazione stringhe in php:

```
$dataFormatted = str_replace("/", "-", $data);  
$year = substr($dataFormatted, -4);  
$dataFormatted = $year."-".$dataFormatted;  
$dataFormatted = substr($dataFormatted, 0,10);  
  
if(strlen($ora)==5){  
    $oraFormatted = $ora.":00";  
}  
else  
    return false;  
//YYYY-MM-DD HH:MM:SS  
$dateTime = $dataFormatted." ".$oraFormatted;
```