

# Diario di lavoro

Luogo	Scuola Arti e Mestieri Trevano
Data	10.10.2019

## Lavori svolti

Martedì scorso il mio responsabile mi ha proposto un calendario che potrebbe andare bene per il mio progetto, infatti offre la funzionalità di avere un piano settimanale e un piano in forma testuale. Il calendario è questo: <https://fullcalendar.io/>

Durante la giornata di oggi, però, ho preferito lavorare su altro. Infatti ho deciso messo inizialmente uno stile alla tabella degli utenti nella pagina di amministrazione. Il risultato finale è il seguente:

Nome	Cognome	Telefono	Ore Lavoro Rimanenti	Tipo	Email	Modifica	Elimina
Mario	Rossi	+41123456789	40	Docente	doc1test@yopmail.com		
Mattia	Toscanelli	+41786175749		Gestore	mattia.toscanelli@samtrevano.ch		

In seguito volevo mettere la funzione di creare gli utenti dalla pagina d'amministrazione ma ho pensato che mi sarebbe convenuto prima fare il controllo lato server della registrazione. Perciò durante questa lezione ho sviluppato il controllo lato server della registrazione, ma non l'ho ancora testato. Il codice è il seguente:

<php

```
class RegistrationModel{

    private $name = "";
    private $surname = "";
    private $number = "";
    private $email = "";
    private $password1 = "";
    private $password2 = "";
    private $connection = "";


    public function __construct($name, $surname, $number, $email, $password1, $password2){
        require_once "database.php";
        $this->connection = new Database();
        $this->name = $name;
        $this->surname = $surname;
        $this->number = $number;
        $this->email = $email;
        $this->password1 = $password1;
        $this->password2 = $password2;
    }

    /**
     * Method to check if the name entered is valid. (Used by name and surname)
     * @param $val The name to chack.
     * @return bool Returns TRUE if the value is valid, FALSE if it is invalid.
     */
    function checkName($val){
        return (preg_match('/^[a-zA-ZáàâãäåąćčêëèéëîïíîñóôõöøùúüûűýÿžžņçššāāāāāāĀČĆĖĘĒĔĕİİİİİİİŁŃÔÖŐőŰÚÚÚÚÚÚÚŸŽŻŇßÇÆȦĈŠŽðð]{2}[/a-zA-
```

```
Zaaãäääçčċēëěėīıİĩłńóôõöøðüúûüÿýžżñćšz̄AAÁÀÃÄÅÇÊÉÈẼȦŁŁŃŇĐÓÔÕÖØÙÚÛÜÝŽŻÑŠĆÆČŚŜǾǻ ,.  
\'-]+$/', $val) && count($val)<=50);  
}  
  
/**  
 * Method to check if an email already exists in the database;  
 * @param $val The email to check.  
 * @return bool REturn TRUE if the email already exists, FALSE if the email is not used.  
 */  
function checkEmailExists($val){  
  
    $selectUsers = "SELECT nome FROM utente WHERE email=:email";  
    $selectUsers->bindParam(':email', $val, PDO::PARAM_STR);  
    $users = $this->fetchAndExecute($selectUsers);  
    return count($users)>0;  
}  
  
/**  
 * Method to prepare the statement, run the query and fetch the result.  
 * @param $selectUsers The select query.  
 * @return array The array of users.  
 */  
private function fetchAndExecute($selectUsers)  
{  
    $this->statement = $this->connection->prepare($selectUsers);  
    $this->statement->execute();  
    $result = $this->statement->fetchAll(PDO::FETCH_ASSOC);  
    return $result;  
}  
  
/**  
 * Method to check if the email entered is valid.  
 * @param $val The email to chack.  
 * @return bool Returns TRUE if the value is valid, FALSE if it is invalid.  
 */  
function checkEmail($val){  
    return preg_match('/^([<>)\\[\\]\\\\.,;:\\s@""]+(\\.^[<>)\\[\\]\\\\.,;:\\s@""]+)*|(".+")@((\\[[\n0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.?)|(\\[a-zA-Z\\-[0-9]+\\.?)+[a-zA-\nZ]{2,}))$/', $val) && !$this->checkEmailExists($val);  
}  
  
/**  
 * Method to check if the number entered is valid.  
 * @param $val The number to chack.  
 * @return bool Returns TRUE if the value is valid, FALSE if it is invalid.  
 */  
function checkNumber($val){  
    $val = str_replace(" ", "", $val);  
    $val = str_replace("-", "", $val);  
    return preg_match('/^[+]?[0-9-#]{10,14}$/', $val);  
}  
  
/**  
 * Method to check if the passwords entered are valid.  
 * @param $val1 The first password.  
 * @param $val2 The repeat password.  
 * @return bool Returns TRUE if the values are valid, FALSE if their are invalid.  
 */  
function checkPassword($val1,$val2){  
    $regexLetter = '/[a-zA-Z]/';  
    $regexDigit = '/\\d/';  
    $regexSpecial = '/[^a-zA-Z\\d]/';  
    return (preg_match($regexDigit,$val1) || preg_match($regexSpecial,$val1)) &&  
        preg_match($regexLetter,$val1) && count($val1) >= 8 &&  
        $val1 == $val2;
```

```
/**
 * Method to check if all the input are valid.
 * @return bool Return TRUE if the values are valid, FALSE if their are invalid.
 */
function checkAll(){
    return $this->checkName($this->name) && $this->checkName($this->surname) &&
        $this->checkEmail($this->email) && $this->checkNumber($this->number) &&
        $this->checkPassword($this->password1,$this->password2);
}

/**
 * Method to insert a user into the database.
 * @return bool Return TRUE if the query is OK, FALSE if the query is failed.
 */
function insertUser(){
    if($this->checkAll()) {
        $password = password_hash($this->password1, PASSWORD_DEFAULT);
        $insertUser = "INSERT INTO utente (email,nome,cognome,numero_telefono,ore_lavoro_tipo,verificato,password)
VALUES ('$this->email','$this->name','$this->surname','$this->number','Visualizzatore',0,'$password')";
        $this->executeAndFetchStatement($insertUser);
        return true;
    }else{
        return false;
    }
}
}
?>
```

#### Problemi riscontrati e soluzioni adottate

Durante la giornata di oggi non ho avuto problemi.

#### Punto della situazione rispetto alla pianificazione

Come già detto nel diario scorso sono in anticipo rispetto alla mia pianificazione, in quanto ho assegnato delle ore di lavoro non molto precise. Però mi sono dimenticato di altre piccole parti di lavoro che mi occuperanno giorni di lavoro.

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

La prossima giornata di lavoro proverò ad assemblare il controllo lato server della pagina registrazione e continuerò la pagina amministrativa.