

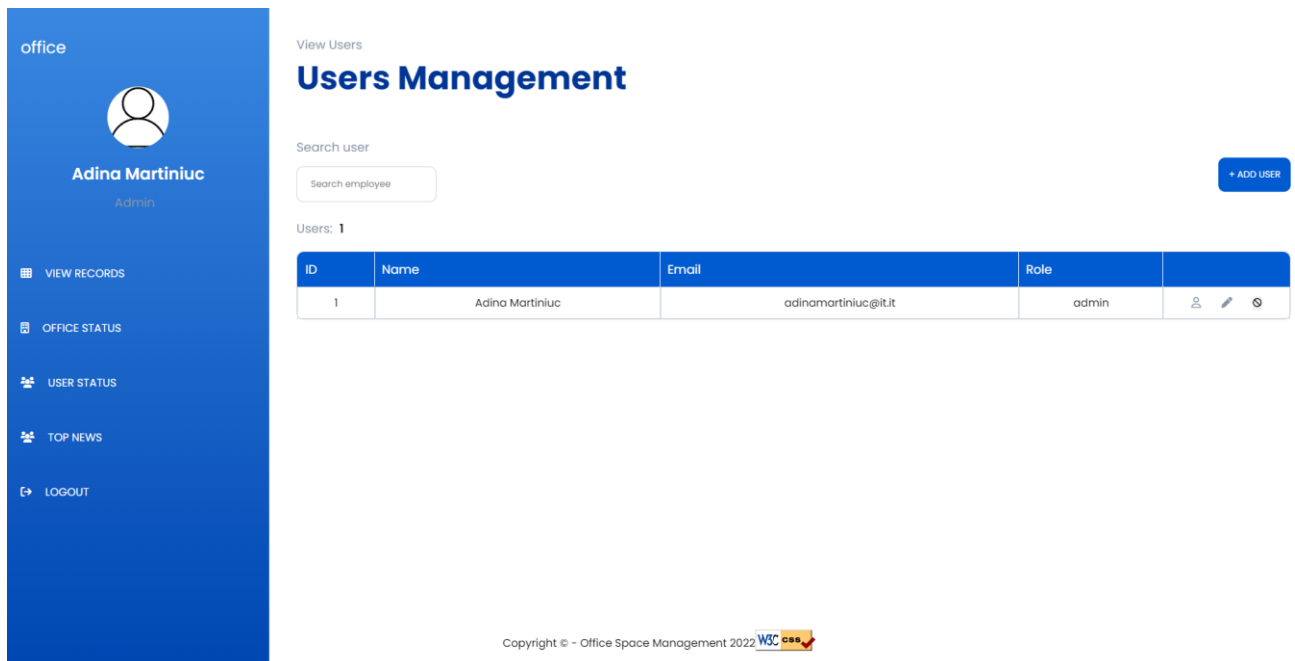
Università degli studi di Torino, A.A. 2021-2022

Tecnologie Web (6 CFU)



Relazione Progetto Finale

Adina Martiniuc - 865044

OVERVIEW DEL SITO



The screenshot displays the 'Users Management' section of the 'office' application. On the left is a blue sidebar with the user's profile 'Adina Martiniuc' (Admin) and navigation links: VIEW RECORDS, OFFICE STATUS, USER STATUS, TOP NEWS, and LOGOUT. The main content area has a title 'View Users' and 'Users Management'. It includes a search bar labeled 'Search user' with a placeholder 'Search employee' and an '+ ADD USER' button. Below the search bar, it indicates 'Users: 1' and shows a table with one user record.

ID	Name	Email	Role	
1	Adina Martiniuc	adinamartiniuc@it.it	admin	 

At the bottom right, there is a copyright notice: 'Copyright © - Office Space Management 2022 W3C CSS'.

OFFICE MANAGEMENT:

Alla base di questo progetto vi è l'idea di creare un sito web che renda pratica e maneggiabile la vita di un dipendente che lavora remote, quindi direttamente da casa o da un ufficio personale.

Proprio per questo motivo il sito, che ho intitolato 'office space management', deve poter essere adatto ad essere adoperato e trattato con facilità ed efficienza da chiunque.

La prima pagina che troviamo una volta che si apre l'applicazione è la pagina del "LOGIN". Questa offre la possibilità di creare un nuovo conto (andando alla pagina del "REGISTER") nel caso in cui l'utente non sia già registrato.

Se l'accesso viene eseguito con successo, l'utente si troverà di fronte ad un menù con il proprio profilo. Avrà la possibilità di navigare e, a seconda del proprio ruolo("Amministratore" o "User"), usufruire delle varie funzionalità del sito.

Il sito è diviso in 4 sezioni principali:

- Login/Register
- View records
- View offices
- Articles

Dentro ogni pagina, è inclusa una sidebar che permette di navigare tra le varie funzionalità e di vedere i propri dati utente.

1) La home page: è composta dalla tabella degli users, ovvero tutti i dipendenti dell'ufficio con la possibilità di fare un search.

2) Abbiamo poi la seconda web page con l'obiettivo di rendere la lista degli impiegati ancora più schematica e dettagliata;

3) La terza pagina riporta delle cards con i diversi uffici dell'edificio di cui si sta parlando (questa pagina è stata creata con l'idea che, nel mondo reale, ad ogni impiegato dovrebbe essere comunque assegnato un proprio desk ed un proprio ufficio, nel caso in cui egli/ella abbia la possibilità di decidere di non lavorare più da casa);

4) La quarta sezione è rappresentata da una pagina di tipo "blog" o "entertainment" in cui troviamo articoli riguardanti svariati argomenti.

Prima dell'autenticazione, le uniche sezioni disponibili sono quella del login e del register.

In ogni pagina, viene controllato che non sia già presente una sessione (un'utente non sia già loggato). Se non lo è, l'utente viene reindirizzato alla prima pagina.

FUNZIONALITA':

- LOGIN/LOGOUT

La prima pagina che viene mostrata ad un utente è la pagina del Login.

Per prima cosa, viene inizializzata una sessione. Subito dopo, controllo che la richiesta sia una di tipo “POST” per trasferire i dati senza che siano visibili nella barra URL.

A questo punto vengono fatti dei controlli:

```
$result = $user->getUser($username);  
  
if(!password_verify($password, $result['password'])){
```

Viene eseguita una query nel database, che permette di prendere tutti i dati del rispettivo utente, si controlla che le 2 password (scritta dall'utente e quella nel database) coincidano e SOLO ALLORA si procede col login.

Si salvano i dati dell'utente nella sessione e questo viene reindirizzato alla pagina viewrecords.php

Per quanto riguarda il logout, nel momento in cui l'utente clicca sul bottone di logout, la sessione viene distrutta e l'utente viene reindirizzato alla pagina del login.

- **REGISTRAZIONE**

iOffice

Welcome!

Register new account

Username

Name

Email

Password

Profile image

REGISTER NOW

Already have an account? Login now

Copyright © - Office Space Management 2022 

La registrazione viene fatta compilando un apposito form e tramite una query al database.

```
public function insertUser($username,$name, $email, $password, $image) {
    try {
        //CHECK IF THERE ARE 2 USERS WITH THE SAME USERNAME
        $result = $this->getUserByUsername($username);
        //IF THERE'S 2 USERS WITH THE SAME USERNAME DO NOTHING
        if($result['num'] > 0) {
            return false;
        }else {
            //ENCRYPT PASSWORD
            $new_password = password_hash($password, PASSWORD_BCRYPT);
            //define sql query
            $sql = "INSERT INTO users(username,name, email,password,image) VALUES (:username,:name,:email,:password, :image)";
            //prepare statement
            $stmt = $this->db->prepare($sql);
            //bind placeholders to params
            $stmt->bindParam(':username', $username);
            $stmt->bindParam(':name', $name);
            $stmt->bindParam(':email', $email);
            $stmt->bindParam(':password', $new_password);
            $stmt->bindParam(':image', $image);
            //execute query
            $stmt->execute();
            return true;
        }
    } catch(PDOException $e){
        echo $e->getMessage();
        return false;
    }
}
```

Si controlla che non ci sia un altro utente con lo stesso username nel database. A questo punto la password viene cryptata attraverso la funzione php “password_hash” e vengono inseriti nel db tutti i dati.

- INSERT IMMAGINE DEL PROFILO

All'utente viene offerta la possibilità di inserire un'immagine del profilo:

Nel backend, vengono inizialmente fatti controlli per quanto riguarda il tipo di file da caricare o la dimensione e successivamente viene inserito. Il file inserito viene aggiunto ad una cartella “uploads” oltre che al database.

```
//PROFILE IMAGE
if(isset($_FILES["avatar"]) && $_FILES["avatar"]["error"] == 0){

    $allowed_ext = array("jpg" => "image/jpg", "jpeg" => "image/jpeg", "gif" => "image/gif", "png" => "image/png");
    $file_name = $_FILES["avatar"]["name"];
    $file_type = $_FILES["avatar"]["type"];
    $file_dim = $_FILES["avatar"]["size"];

    //check extensions
    $ext = pathinfo($file_name, PATHINFO_EXTENSION);
    $destination = "uploads/$username.$ext";
    if(!array_key_exists($ext, $allowed_ext)) die("Please select a valid image");

    //check for max size
    $max_size = 5 * 1024 * 1024;
    if($file_dim > $max_size) die("Max size is 5MB");

    //check for MIME type
    if(in_array($file_type, $allowed_ext)){
        //check if file already exists
        if(file_exists("uploads/$file_name")){
            echo "<p class='text-danger'> $file_name already exists, choose another name </p>";
        } else {
            move_uploaded_file($_FILES["avatar"]["tmp_name"], $destination);
        }
    } else {
        echo "<p class='text-danger'> There was an error with your uploading, please try again</p>";
    }
} else {
    echo 'Error: ' . $_FILES["avatar"]["error"];
}
}
```

- RUOLI

Nel website che ho progettato sono presenti 2 ruoli:

- Admin
- User

Tutti gli utenti hanno la possibilità di vedere gli altri dipendenti, uffici, articoli ma, solamente l'admin ha la possibilità di aggiungere nuovi uffici o dipendenti, di modificarli o eliminarli.

Questo è reso possibile controllando, attraverso la sessione, che il ruolo coincida con quello dell'admin o dello user.

- PAGINA DEGLI ARTICOLI

Per questa pagina, ho usato una API “NewsAPI”

<https://newsapi.org/>

Questa è una HTTP API che permette di trovare e vedere in live le ultime notizie.

Usando una chiave, viene fatto il fetch dei dati

GET <https://newsapi.org/v2/top-headlines?country=us&apiKey=...>

Mentre i dati vengono caricati, un loader fa capire che non si tratta di un errore ma solo che i dati non sono ancora presenti.

Per semplicità, vengono mostrate solamente le prime 2 notizie principali.

SCHEMA DEL DATABASE:

Il database è composto da 2 tabelle:

- Users
- Offices

Tabella users:

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito	Commenti	Extra	Azione
<input type="checkbox"/> 1	id	int(100)			No	Nessuno		AUTO_INCREMENT	Modifica Elimina Più
<input type="checkbox"/> 2	username	varchar(50)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 3	name	varchar(255)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 4	email	varchar(255)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 5	password	varchar(255)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 6	user_type	varchar(20)	utf8mb4_general_ci		No	user			Modifica Elimina Più
<input type="checkbox"/> 7	image	varchar(255)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 8	office_nr	int(11)			No	1			Modifica Elimina Più

Alla registrazione, un'utente sarà posto di default come user. Solamente un admin potrà, successivamente modificare il suo ruolo.

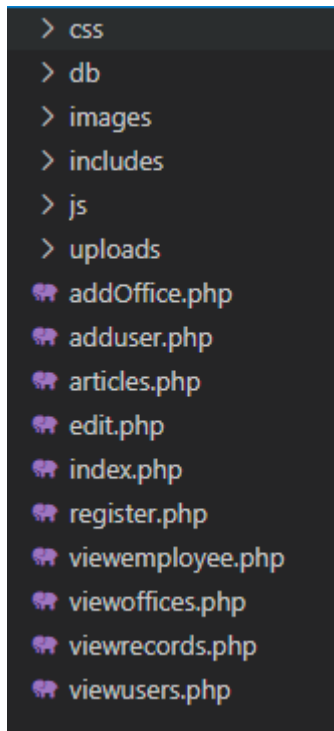
Viene creata una foreign_key sulla colonna office_nr per permettere di collegare un utente ad un ufficio.

Tabella offices:

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito	Commenti	Extra	Azione
<input type="checkbox"/> 1	office_id	int(11)			No	Nessuno		AUTO_INCREMENT	Modifica Elimina Più
<input type="checkbox"/> 2	office_name	varchar(20)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più
<input type="checkbox"/> 3	address	varchar(255)	utf8mb4_general_ci		No	Nessuno			Modifica Elimina Più

Per evitare sql injections, è stato usato PDO che permette di assegnare le variabili dopo aver dichiarato la query sql da eseguire.

ORGANIZZAZIONE DELLE CARTELLE:



Sono state create diverse cartelle a seconda del loro ruolo.

Solamente le pagine che sono direttamente visibili all'utente sono rimaste nella cartella parente.

Tutti i file php che agiscono solamente nel backend sono posti nella cartella includes.

La cartella uploads, invece, contiene le immagini che vengono inserite dagli utenti al momento della registrazione.

USABILITA':

Per lo stile, ho usato un'unica pagina di stile style.css

Per rendere l'applicazione funzionale per tutti gli utenti, ho usato pochi colori con grandi contrasti tra di loro.

Sono state usate anche diverse media queries per dispositivi con grandezze di schermo diverse per ottenere un layout responsive.

La validazione degli input è fatta attraverso funzioni jquery per quanto riguarda il serverside e funzioni php per quanto riguarda il backend.