Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Trevano |
| Data | 01.10.2019 |

|  |
| --- |
| Lavori svolti |
| Oggi ho iniziato con l'implementazione del backend della pagina di registrazione e l'ho portato a termine. Ho iniziato creando una funzione che verifica il login di un utente verificando che l'email sia in uso da un account e, se così fosse, che la password corrisponda con quella dell'account.  *public function checkLogin(){*  *//richiamo le classi di cui avrò bisogno*  *unset($\_SESSION['warning']);*  *require\_once "./application/models/db\_connection.php";*  *require\_once "./application/models/input\_manager.php";*  *$errors = array();*  *//verifico il metodo di richiesta*  *if($\_SERVER["REQUEST\_METHOD"] == "POST") {*  *//verifico che i campi siano impostati e che non siano stringhe vuote*  *if (isset($\_POST['email']) && !empty($\_POST['email']) && isset($\_POST['password']) && !empty($\_POST['password'])) {*  *//genero un nuovo InputManager e testo gli inserimenti*  *$im = new InputManager();*  *$email = filter\_var($im->checkInput($\_POST['email']), FILTER\_SANITIZE\_EMAIL);*  *$password = filter\_var($im->checkInput($\_POST['password']), FILTER\_SANITIZE\_STRING);*  *//prendo tutti gli utenti nel database*  *$db = (new db\_connection)->getUsers();*  *//hash della password così da poterla comparare con quella nel db*  *$password = hash('sha256', $password);*  *foreach ($db->fetchAll() as $row) {*  *//controllo che l'email sia in uso da un utente*  *if ($row['email'] == $email) {*  *//controllo che la password corrisponda*  *if($row['password'] == $password){*  *$\_SESSION['user'] = (new db\_connection)->getUser($email);*  *//verifico se è admin o utente normale*  *if($row['nome\_ruolo'] == 'admin'){*  *header('Location: ' . URL . 'admin');*  *exit();*  *}elseif($row['nome\_ruolo'] == 'utente'){*  *header('Location: ' . URL . 'home');*  *exit();*  *}*  *}else{*  *//se la password è sbagliata ritorno l'errore*  *array\_push($errors, "Password o email sbagliate");*  *$\_SESSION['warning'] = $errors;*  *header('Location: ' . URL . 'login');*  *exit();*  *}*  *}*  *}*  *//se l'email non è in utilizzo da nessun user ritorno l'errore*  *array\_push($errors, "Password o email sbagliate");*  *$\_SESSION['warning'] = $errors;*  *header('Location: ' . URL . 'login');*  *exit();*  *}*  *}*  *}*  Dopo aver scritto e testato la funzione ho implementato alcune piccole cose nei controller ad esempio nell'home se si è loggati e a dipendenza di che livello di privilegi si hanno si vedono degli header diversi. Headers che sono andato a modificare oggi aggiungendo i percorsi alle pagine e inserendo nei due per gli utenti loggati la possibilità di uscire dall'account dato che ho creato la funzione.  *public function logout(){*  *unset($\_SESSION['user']);*  *header('Location: ' . URL . 'home');*  *exit();*  *}*  Ho inoltre commentato tutti i controllers e i models che non avevano ancora tutte le spiegazioni.  Per far si che il login funzionasse ho implementato la funzione getUser nel model di gestione del database. Essa ritorna un utente in base alla sua email.  *public function getUser($email){*  *try{*  *$db = $this->getConnection();*  *$query = $db->prepare('SELECT \* FROM utente where email=?');*  *$query->bindParam(1, $email, PDO::PARAM\_STR);*  *$query->execute();*  *return $query->fetchAll();*  *}catch (Exception $e){*  *header('Location: ' . URL . 'warning');*  *}*  *}* |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Inizialmente ho riscontrato un problema con il login dato che nel database avevo degli hash non corretti e quindi facendo il paragone la password risultava sempre sbagliata. Ho risolto eliminando i record e reinserendoli nel database dalla pagina di registrazione. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono in anticipo sulla pianificazione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| La prossima giornata voglio creare i validatori frontend per le pagine di registrazione e di login. |