Documentation of implementation and testings

team 10

**Date: 2/5/2018**

**Μαριάνα Μηνά**

**Λουίζα Αγρότη**

**Μαρία Κουππή**

**Ελένη Κάτση**

**Table of contents:**

1. Implementation (Basic parts of implementation with parts of code and explanations)

2. Testings (scenarios, extraction of testings, used tools and results)

3. Security techniques that adopted and implemented

4. Software license

**Implementation: (Basic parts of implementation with parts of code and explanations)**

1. **edit\_profile\_manager.php/edit\_profile\_employee.php**
2. $sql\_select = "SELECT  Username, ID, Name, Surname, Password, Birthdate, Gender, Address, Country, Phone, EmergencyPhone, Role, Salary, SalaryType, SSN, Email, AnnualLeaves, CharactersPassword FROM Employee WHERE Username LIKE '$Username' ";

Comment: Get the data of the selected employee/manager to show them in the screen.

1. $sql\_depart = "SELECT  Username, NumDept, CountryNumber, Department.NameDept, Department.CountryNum, Department.NumberDept, CorporateHeadquarter.CountryNum, CorporateHeadquarter.Name FROM Employee, Department, CorporateHeadquarter WHERE Department.NumberDept = Employee.NumDept && Department.CountryNum = CorporateHeadquarter.CountryNum && Department.CountryNum = Employee.CountryNumber && Username LIKE '$Username' ";

Comment: Get the data of the selected employee/manager related to Department table to show them in the screen

1. **update\_profile\_manager.php/ update\_profile\_employee.php**
2. $sqlUpdate = "UPDATE Employee SET Password='$Hashed', Phone = '$Phone', EmergencyPhone = '$EmergencyPhone', Country = '$Country', Address = '$Address', Gender = '$Gender', CharactersPassword = '$Password\_len' WHERE Username = '$\_SESSION[username]'";

Comment:  Update data of the selected employee/manager

1. **Delete\_employee.php**
2. $sql\_select= "SELECT Username, UsernameManager FROM Employee WHERE Username LIKE '$Username' AND UsernameManager LIKE '$UsernameManager'";

Comment: Get selected employee's username and manager’s username

1. $sql\_insert = "INSERT INTO DeletedEmployee (Username, ID, Name, Surname, Birthdate, Gender, Address, Country, Phone, EmergencyPhone, Role, Salary, SalaryType, SSN, Email) (SELECT Username, ID, Name, Surname, Birthdate, Gender, Address, Country, Phone, EmergencyPhone, Role, Salary, SalaryType, SSN, Email FROM Employee WHERE Username LIKE '$Username' AND UsernameManager LIKE '$UsernameManager')";

Comment: Insert selected employee's record into DeletedEmployee table because in case of error we don't want to lose the deleted employee's record

1. $sql\_delete = "DELETE FROM Employee WHERE Username LIKE '$Username' AND UsernameManager LIKE '$UsernameManager'";

Comment: Delete selected employee from Employee database's table

1. **clockIn\_manager.php/clockIn\_employee.php:**
2. $sqlClockedIn = "SELECT AttendanceTime.\* FROM (SELECT MAX(AttendanceTime.ClockIn) AS ClockInMax,AttendanceTime.Date,AttendanceTime.Username FROM AttendanceTime WHERE AttendanceTime.Date = curdate() GROUP BY AttendanceTime.Date,AttendanceTime.Username) AS A INNER JOIN AttendanceTime ON (A.Date=AttendanceTime.Date AND A.ClockInMax=AttendanceTime.ClockIn AND A.Username=AttendanceTime.Username) WHERE AttendanceTime.Username LIKE '$Username'";

Comment:  Select the row from the database that contains the last time an employee or a manager clicked Clock in button at the current date. It is used to check which buttons an employee or a manager is allowed to click.

1. **returnFromBreak\_manager.php/returnFromBreak\_employee.php:**
2. $query = "SELECT ReturnBreak,Break,ClockIn,Date,Username FROM AttendanceTime WHERE ClockIn=(SELECT maxClockIn FROM (SELECT MAX(ClockIn) AS maxClockIn,Date,Username FROM AttendanceTime WHERE Date = curdate() AND Username LIKE '$Username' GROUP BY Date,Username) AS Tmp)";

Comment:  Select the row from the database that an employee or a manager clicked the button Clock in for the last time at the current date

1. $query = "SELECT ReturnBreak,Break,ClockIn,Date,Username FROM AttendanceTime WHERE ClockOut='$clockOut'";

Comment: Select the row from the datatbase that an employee or a manager did not press the button Clock out. This is done in case that an employee or a manager does not click the button Clock out at the same date that he/she clicked either Clock in, Break or Return from Break buttons.

1. $endTime = ($row\_break\_len['ReturnBreak']{0} . $row\_break\_len['ReturnBreak']{1})\*60\*60 + ($row\_break\_len['ReturnBreak']{3} . $row\_break\_len['ReturnBreak']{4})\*60 + ($row\_break\_len['ReturnBreak']{6} . $row\_break\_len['ReturnBreak']{7})\*1;

$startTime = ($row\_break\_len['Break']{0} . $row\_break\_len['Break']{1})\*60\*60 + ($row\_break\_len['Break']{3} . $row\_break\_len['Break']{4})\*60 + ($row\_break\_len['Break']{6} . $row\_break\_len['Break']{7})\*1;

$newTime = $endTime-$startTime;

$break = (int)($newTime/60);

Comment: Calculate the break length of an employee or a manager in minutes.

1. **add\_employee\_function.php:**
2. $sqlDeptNum = "SELECT NumberDept FROM Department WHERE NumberDept={$DeptNum} AND CountryNum={$WorkC}";

Comment: Check if the given Department is in the given Country in case that a manager adds an employee in the database.

1. $increaseEmployees = "UPDATE Department SET NumEmployees=(NumEmployees+1) WHERE NumberDept={$DeptNum} AND CountryNum={$WorkC}";

Comment: In case that a manager adds an employee in the database the number of employees of the given department in the database must increase by one.

1. $sqlEmpl = "INSERT INTO Employee (CountryNumber,Email,NumDept,IsManager,Username,Password,ID,SSN,Name,Surname,Role,Salary,SalaryType,Phone,EmergencyPhone,Country,Address,Birthdate,Gender,UsernameManager,CharactersPassword,AnnualLeaves) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

Comment: Inserts an employee in the database.

1. **update\_edit\_employees.php:**
2. $sqlUpdate = "UPDATE Employee SET AnnualLeaves='$Leaves',CountryNumber='$WorkC',NumDept='$DeptNum',Role='$Role',Salary='$Salary',SalaryType='$SalaryType' WHERE Username LIKE '$emplUsername'";

Comment: Update the data of an employee in case in which a manager wants to edit his employee’s data.

1. **payroll\_function.php/average\_per\_week\_manager.php:**
2. if (strtotime($DateTo) < strtotime($DateFrom)) {

        echo '<script type="text/javascript">

window.alert("The To field has to be after the From field!");

window.location.replace("payroll\_report.html");

</script>';

     }else if (strtotime($DateFrom) >= strtotime('now')) {

        echo '<script type="text/javascript">

var d = new Date();

window.alert("The date must be before " + d.getDate() + "/" + d.getMonth()+1 + "/"  + d.getFullYear());

window.location.replace("payroll\_report.html");

</script>';

    } else if (strtotime($DateTo) >= strtotime('now')) {

        echo '<script type="text/javascript">

var d = new Date();

window.alert("The date must be before " + d.getDate() + "/" + d.getMonth()+1 + "/"  + d.getFullYear());

window.location.replace("payroll\_report.html");

</script>';

    }

Comment: Check if the fields of dates are correct. For example if the field From is before the field To and if the field From and To is before the current date because a manager can not export a report after the current date.

1. **json\_file.php/xml\_file.php:**
2. $getData="SELECT SUM(TIME\_TO\_SEC(TIMEDIFF(AttendanceTime.ClockOut,AttendanceTime.ClockIn))-(AttendanceTime.BreakLength\*60)) AS SecWorked,AttendanceTime.Username,AttendanceTime.BreakLength,Employee.ID,Employee.Name,Employee.Surname FROM AttendanceTime INNER JOIN Employee ON (AttendanceTime.Username=Employee.Username) WHERE (Date >= '$From' AND Date <= '$To') AND Employee.UsernameManager LIKE '$Username' GROUP BY AttendanceTime.Username";

Comment: Select how many hours the employees of the current manager worked for the given period of time to export the report.

1. **contact.php**
2. if(empty($\_POST['To'])){

$to\_error="\* ' To ' field is required";

}else if (!filter\_var($\_POST['To'], FILTER\_VALIDATE\_EMAIL)) {

$to\_error = "\* Invalid email format";

}else{

$to\_error="";

$to=mysqli\_real\_escape\_string($conn, $\_POST['To']);

$TO=$to;

}

Comment: If an employee didn't set the 'To' field then show him/her an error that the field is required, else if employee has set the 'To' field but is not an email then show him/her an error that is not a valid form of an email, else employee has set right the 'To' field.

1. **forgot\_pass.php**
2. for ($i = 0; $i < $length; $i++) {

$randomString .= $characters[rand(0, $charactersLength - 1)];

}

Comment: Selects a random character in each iteration and [concat](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/concat)s them, so it creates a random string.

1. $sqlEmpl = "UPDATE Employee SET Password='$Hashed' WHERE Username LIKE '$userName'";

Comment: Changes the password field of the employee and sets the new random string which has been created before.

1. **view\_hours\_employee.php/view\_hours\_manager.php**
2. $sql = "SELECT ClockIn, ClockOut, BreakLength, Date FROM AttendanceTime WHERE Username LIKE '$Username' ORDER BY Date DESC";

Comment: Takes the clock in and clock out time, the break length and the date of an employee and sorted them [descending](https://www.google.com.cy/search?q=descending&spell=1&sa=X&ved=0ahUKEwj__M3upOfaAhXkL8AKHVUYArQQkeECCCIoAA) by the date.

1. $endTime2 = ($row['ClockOut']{0} . $row['ClockOut']{1})\*60\*60 + ($row['ClockOut']{3} . $row['ClockOut']{4})\*60 + ($row['ClockOut']{6} . $row['ClockOut']{7})\*1;

Comment: Calculates the time in which employee press the button clock out, in seconds.

1. $newTime2 : has the working hours of an employee in seconds.

$hours2 = (int)($newTime2 / 3600);

Comment: Calculates the hours of $newTime2.

1. $min2 = (int)(($newTime2 - $hours2\*3600) / 60);

Comment: Calculates the minutes of $newTime2.

1. $sec2 = (int)($newTime2 - $hours2\*3600 - $min2\*60);

Comment: Calculates the seconds of $newTime2.

1. **manager\_leave.php/employee\_leave.php**
2. if (strtotime($DateFrom) < strtotime('now')) {

                printError();

            } else if (strtotime($DateTo) < strtotime('now')) {

                printError();

            } else if (strtotime($DateTo) < strtotime($DateFrom)) {

                echo '<script type="text/javascript">

window.alert("The date on which the leave will end is after the date it starts! Try again please!");

window.location.replace("leave\_request\_manager.html");

</script>';

}

Comment:  Check if the fields of dates are correct. For example if the date when he/she want to start the leave is after the date when he/she return from leaves,and if the date when he/she want to start the leave and  the date when he/she return from leaves is before the current date, then he/she can not do a leave request before the current date.

1. $sql2 = "SELECT Username FROM Employee WHERE Username='$username'";

Comment: Select the username from the database that an employee or a manager who did login.

1. $sql = "INSERT INTO `Leave`(`Reason`, `ToDate`, `FromDate`, `Username`) VALUES (?,?,?,?)";

Comment: Insert the employee’s or manager’s leave request in database. Insert the date when he/she  will start the leave, the date when he/she return from leave, the reason for his/her leave (e.g holiday, doctor appointment) and his/her username because his/her manager should be accept his/her request.

1. **employee\_view\_request.php**
2. $sql = "SELECT  `FromDate`, `ToDate`,  `Reason`, `State` FROM `Leave` WHERE `Username` LIKE '$Username'";

$result = mysqli\_query($conn, $sql);

Comment: Select from database all leave requests which a employee who is login did and they appear in a table in page f for he/she can see if  his/her request became accepted or rejected from his/her manager.

**Testings: (scenarios, extraction of testings, used tools and results)**

1. **index.html:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Username** | | | **Password** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 | ekatsi03 | ✓ |  | - | ✓ |  | Password field is required | Password field is required |
| 2 | lagrot01 |  | ✓ | - | ✓ |  | Password field is required | Password field is required |
| 3 | lagrot01 |  | ✓ | 1234 | ✓ |  | Wrong password | Wrong password |
| 4 | - | ✓ |  | 123456 |  | ✓ | Username field is required | Username field is required |
| 5 | lagrot01 |  | ✓ | 123456 |  | ✓ | Login successfully | Login successfully |

**2. add\_employee.php:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Username** | | | **Password** | | | **ID** | | | **SSN** | | | **Role** | | | **Department Number** | | | **Country Number** | | | **Salary** | | | **Date of Birth** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 | - | ✓ |  | - | ✓ |  | - | ✓ |  | - | ✓ |  | - | ✓ |  | - |  | ✓ | - |  | ✓ | - |  | ✓ | - |  | ✓ | Required messages | Required messages |
| 2 | mjake |  | ✓ | 1472 |  | ✓ | 987456 |  | ✓ | ssnNO1 | ✓ |  | - | ✓ |  | - |  | ✓ | - |  | ✓ | - |  | ✓ | - |  | ✓ | Only numbers allowed for SSN field. Required field for Role | Only numbers allowed for SSN field. Required field for Role |
| 3 | dvin1 |  | ✓ | 1258 |  | ✓ | 999851 |  | ✓ | 1255 |  | ✓ | IT |  | ✓ | - |  | ✓ | - |  | ✓ | - |  | ✓ | - |  | ✓ | Employee is inserted in the database | Employee is inserted in the database |
| 5 | mjohn |  | ✓ | 1777 |  | ✓ | 992111 |  | ✓ | 175 |  | ✓ | IT |  | ✓ | No1 | ✓ |  |  | ✓ |  | - |  | ✓ | - |  | ✓ | Only numbers allowed for Department and Country Number field | Only numbers allowed for Department and Country Number field |
| 6 | klamp5 |  | ✓ | 1877  3 |  | ✓ | 992118 |  | ✓ | 1112 |  | ✓ | CEO |  | ✓ | 1 |  | ✓ | 1 |  | ✓ | ten | ✓ |  | 01/03/1990 | ✓ |  | Date of birth field should be in the following format:  YYY-MM-DD. Only numbers (decimal or not) are allowed | Date of birth field should be in the following format:  YYY-MM-DD. Only numbers (decimal or not) are allowed |
| 7 | mackl05 |  | ✓ | 9877  3 |  | ✓ | 902118 |  | ✓ | 1012 |  | ✓ | CEO |  | ✓ | 11 | ✓ |  | 1 |  | ✓ | 900 |  | ✓ | 1997-01-01 |  | ✓ | Department number does not exist in the database | Department number does not exist in the database |
| 8 | mackl05 |  | ✓ | 9877  3 |  | ✓ | 902118 |  | ✓ | 1012 |  | ✓ | CEO |  | ✓ | 1 |  | ✓ | 12 | ✓ |  | 900 |  | ✓ | 1997-01-01 |  | ✓ | Country number does not exist in the database | Country number does not exist in the database |
| 8 | mackl05 |  | ✓ | 9877  3 |  | ✓ | 902118 |  | ✓ | 1012 |  | ✓ | CEO |  | ✓ | 1 |  | ✓ | 1 |  | ✓ | 900 |  | ✓ | 1997-01-01 |  | ✓ | Employee inserted correctly in the database | Employee inserted correctly in the database |

**Assuming again that all of the above fields inserted correctly the last checks are the following:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Department** | | | **Country of Department** | | | **Date of Birth** | | | **Expected Result** | **Result** |
|  |  | **Wrong** | **Right** |  | **Wrong** | **Right** |  | **Wrong** | **Right** |  |  |
| 1 | 1 - IT |  | ✓ | 1 - Sweden |  | ✓ | 21/01/1995 | ✓ |  | Date of birth field should be in the following format YYY-MM-DD | Date of birth field should be in the following format YYY-MM-DD |
| 2 | 1 - IT |  | ✓ | 2 - Cyprus | ✓ |  | 1995-11-21 |  | ✓ | Department in the given country does not exist in Database | Department in the given country does not exist in Database |
| 3 | 1 - IT |  | ✓ | 1 - Sweden |  | ✓ | 1995-11-21 |  | ✓ | Inserted correctly | Inserted correctly |

**3. Delete Employee:**

Manager can select an employee to delete him/her from a list so there aren’t any testings to do

**4. Contact us:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **To** | | | **Subject** | | | **Description** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 | - | ✓ |  | - | ✓ |  | - | ✓ |  | ‘To’ field is required  ‘Subject’ field is required  ‘Description’ field is required | ‘To’ field is required  ‘Subject’ field is required  ‘Description’ field is required |
| 2 | - | ✓ |  | - | ✓ |  | hello |  | ✓ | ‘To’ field is required  ‘Subject’ field is required | ‘To’ field is required  ‘Subject’ field is required |
| 3 | - | ✓ |  | exam |  | ✓ | - | ✓ |  | ‘To’ field is required  ‘Description’ field is required | ‘To’ field is required  ‘Description’ field is required |
| 4 | - | ✓ |  | exam |  | ✓ | hello |  | ✓ | ‘To’ field is required | ‘To’ field is required |
| 5 | ekatsi03@cs.ucy.ac.cy |  | ✓ | - | ✓ |  | - | ✓ |  | ‘Subject’ field is required  ‘Description’ field is required | ‘Subject’ field is required  ‘Description’ field is required |
| 6 | ekatsi03@cs.ucy.ac.cy |  | ✓ | - | ✓ |  | hello |  | ✓ | ‘Subject’ field is required | ‘Subject’ field is required |
| 7 | ekatsi03@cs.ucy.ac.cy |  | ✓ | exam |  | ✓ | - | ✓ |  | ‘Description’ field is required | ‘Description’ field is required |
| 8 | ekatsi03@cs.ucy.ac.cy |  | ✓ | exam |  | ✓ | hello |  | ✓ | message sent | message sent |
| 9 | ekatsi03 | ✓ |  | exam |  | ✓ | hello |  | ✓ | Invalid email format | Invalid email format |

**5. Leave Request:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Start Date** | | | **Finish Date** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 | 27/12/2017 | ✓ |  | 02/01/2017 | ✓ |  | alert box with error message | alert box with error message |
| 2 | 27/12/2017 | ✓ |  | 23/12/2017 | ✓ |  | alert box with error message | alert box with error message |
| 3 | 05/04/2018 |  | ✓ | 23/12/2017 | ✓ |  | alert box with error message | alert box with error message |
| 4 | 27/12/2017 | ✓ |  | 05/04/2018 |  | ✓ | alert box with error message | alert box with error message |
| 5 | 05/04/2018 |  | ✓ | 10/04/2018 |  | ✓ | insert correctly | insert correctly |

**6. View Hours:**

-if the employee doesn’t press clock out, hours does not compute.

-if the employee doesn’t press return from break, break length doesn’t compute.

**7. Clock in:**

-Employee can’t press clock out before press clock in first

-Employee can’t press return from break before press break first.

-If an employee is clocked in and is in break, he/she can’t press clock out. He/she have to press return from break first and then clock out.

**8. Employee Status:**

According to the username of the manager, the username, first name, last name and status of his employees appear on the screen. There are some checks in the database that are related to the status of manager’s employees:

-If employee is clocked in that means that in the database is only field that is initialazed with a time, different from 00:00:00, which is the default value of all fields of Attedance Time table.

-If employee is clocked out that means that in the database field Clock out and Clock in are initialized with a time,  different from 00:00:00.

-If employee is on break that means that in the database field Break and Clock in are initialized with a time,  different from 00:00:00.

- If employee is clocked in (to be exactly employee has returned from break) that means that in the database field ReturnBreak,Break and Clock in are initialized with a time,  different from 00:00:00 and time of ReturnBreak field is greater than the time of Break field.

- If employee is on break that means that in the database field ReturnBreak,Break and Clock in are initialized with a time,  different from 00:00:00 and time of Break field is greater than the time of ReturnBreak field.

-If all fields are initialized with a time,  different from 00:00:00, according to which field has the greater time ( ReturnBreak,Break ,ClockOut), employee has that status.

**9. Payroll Report  / Average Report:**

For example the date of today is: 10/04/2018

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **From Date** | | | **To Date** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 | 13/09/2018 | ✓ |  | 15/09/2018 | ✓ |  | The date must be before 10/04/2018 | The date must be before 10/04/2018 |
| 2 | 13/09/2018 | ✓ |  | 10/09/2018 | ✓ |  | The To field has to be after the From field | The To field has to be after the From field |
| 3 | 03/04/2018 |  | ✓ | 23/05/2018 | ✓ |  | The date must be before 10/04/2018 | The date must be before 10/04/2018 |
| 4 | 27/05/2018 | ✓ |  | 03/04/2018 |  | ✓ | The To field has to be after the From field | The To field has to be after the From field |
| 5 | 03/04/2018 |  | ✓ | 06/04/2018 |  | ✓ | export the report | export the report |
| 6 | 06/04/2018 |  | ✓ | 03/04/2018 |  | ✓ |  |  |

**10. View Request (employee):**

-Check whether the data of the connected employee is the same as the data in the database.

-Check whether the data displayed belongs only to the connected employee.

**11. Forgot Password:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Username** | | | **Email** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 | - | ✓ |  | - | ✓ |  | ‘Username’ field is required  ‘Email’ field is required | ‘Username’ field is required  ‘Email’ field is required |
| 3 | - | ✓ |  | ekatsi03@cs.ucy.ac.cy |  | ✓ | ‘Username’ field is required | ‘Username’ field is required |
| 5 | ekatsi03 |  | ✓ | - | ✓ |  | ‘Email’ field is required | ‘Email’ field is required |
| 6 | ekatsi03 |  | ✓ | ekatsi03@cs.ucy.ac.cy |  | ✓ | password sent to your email | password sent to your email |
| 7 | ekatsi03 |  | ✓ | ekatsi03 | ✓ |  | Invalid email format | Invalid email format |

**12. Edit Profile**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Password** | | | **Phone** | | | **Emergency Phone** | | | **Country** | | | **Address** | | | **Expected Result** | **Result** |
|  |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  | Wrong | Right |  |  |
| 1 |  | ✓ |  |  | ✓ |  |  | ✓ |  |  | ✓ |  |  | ✓ |  | Required fields(\*) | Required fields(\*) |
| 2 | 123456 |  | ✓ | 64502580 |  | ✓ | 64502588 |  | ✓ | Cyprus |  | ✓ | Nicosia |  | ✓ | Updated successfully | Updated successfully |
| 3 | 123456 |  | ✓ | 64502580/ | ✓ |  | 64502588 |  | ✓ | Cyprus |  | ✓ | Nicosia |  | ✓ | Invalid Phone number | Invalid Phone number |
| 4 | 123456 |  | ✓ | 64502580 |  | ✓ | 64502588\* |  | ✓ | Cyprus |  | ✓ | Nicosia |  | ✓ | Invalid Emergency Phone number | Invalid Emergency Phone number |
| 5 | 123456 |  | ✓ | 64502580 |  | ✓ | 64502588 |  | ✓ | Cyprus\* | ✓ |  | Nicosia |  | ✓ | Only letters and white space allowed in Country field | Only letters and white space allowed in Country field |

(\*) - Required fields are Password, Phone, Emergency Phone

**Security techniques that adopted and implemented:**

We use two techniques in order to secure our system from injections. One technique is to prevent SQL injection and the other is to prevent Javascript injection.

SQL Injection:

In order to prevent the SQL injection, we use prepare statements.

Firstly, for each assign of $\_POST into a variable we use the mysqli\_real\_escape\_string() function.

e.g. $Salary = mysqli\_real\_escape\_string($conn, $\_POST['Salary']);

1. For INSERT query:

e.g. $sqlEmpl = "INSERT INTO Employee (Name,Surname,Role,Salary,Phone,Country,Address,Birthdate,Gender,AnnualLeves) VALUES (?,?,?,?,?,?,?,?,?,?)";

1. For UPDATE query:

e.g. $sqlUpdate = "UPDATE Employee SET AnnualLeaves=?,CountryNumber=?,NumDept=?,Role=?,Salary=?,SalaryType=? WHERE Username LIKE '$emplUsername'";

Then, we use this commands:

$stmt = mysqli\_stmt\_init($conn);

if(!mysqli\_stmt\_prepare($stmt, $sqlUpdate)){

echo "SQL error";

}else{

mysqli\_stmt\_bind\_param($stmt, "isssis", $Leaves, $WorkC, $DeptNum, $Role, $Salary, $SalaryType);

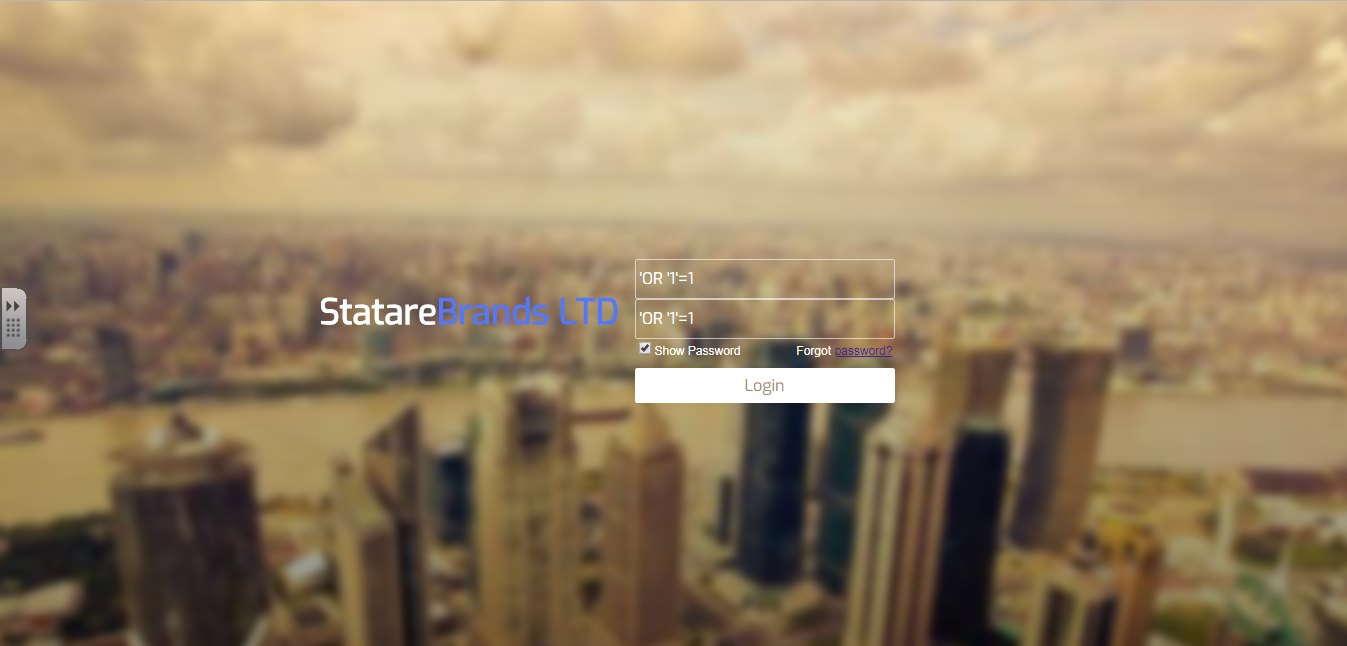
mysqli\_stmt\_execute($stmt);

}

At mysqli\_stmt\_bind\_param() function, the second parameter is a string of the types of the variables. “i” is for int and “s” is for string.

First example for SQL injection at Login screen:

Before pressing the button Login:

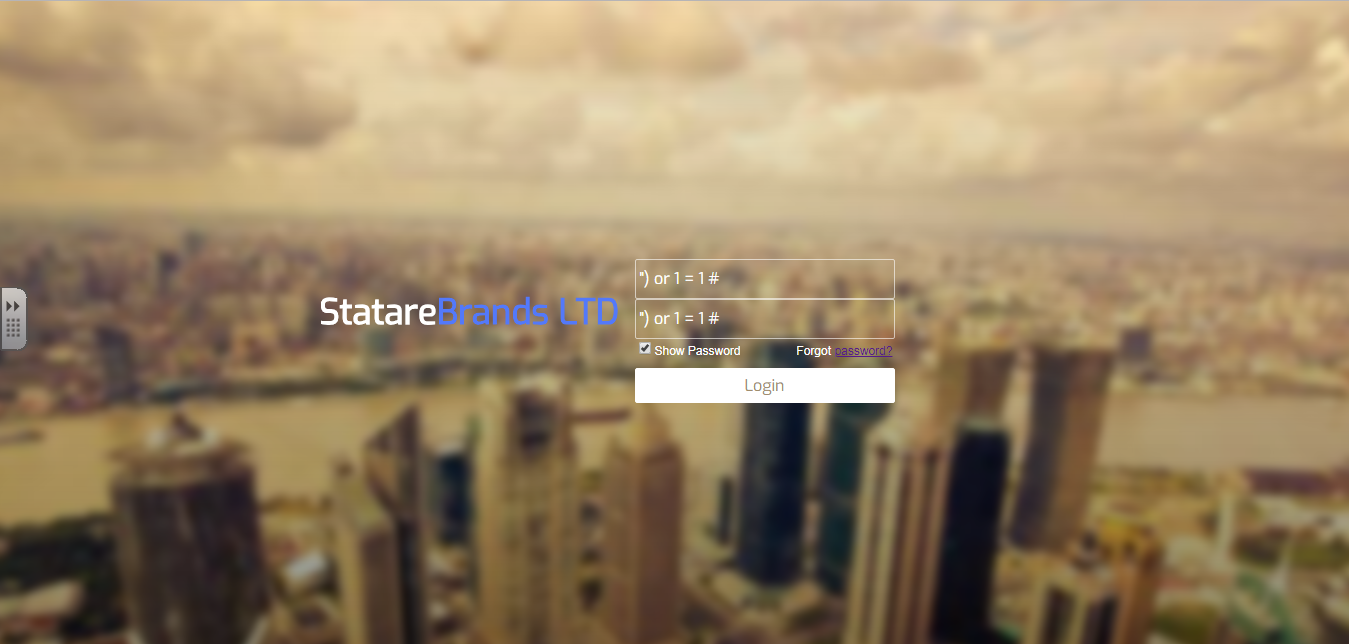


After pressing the button Login:

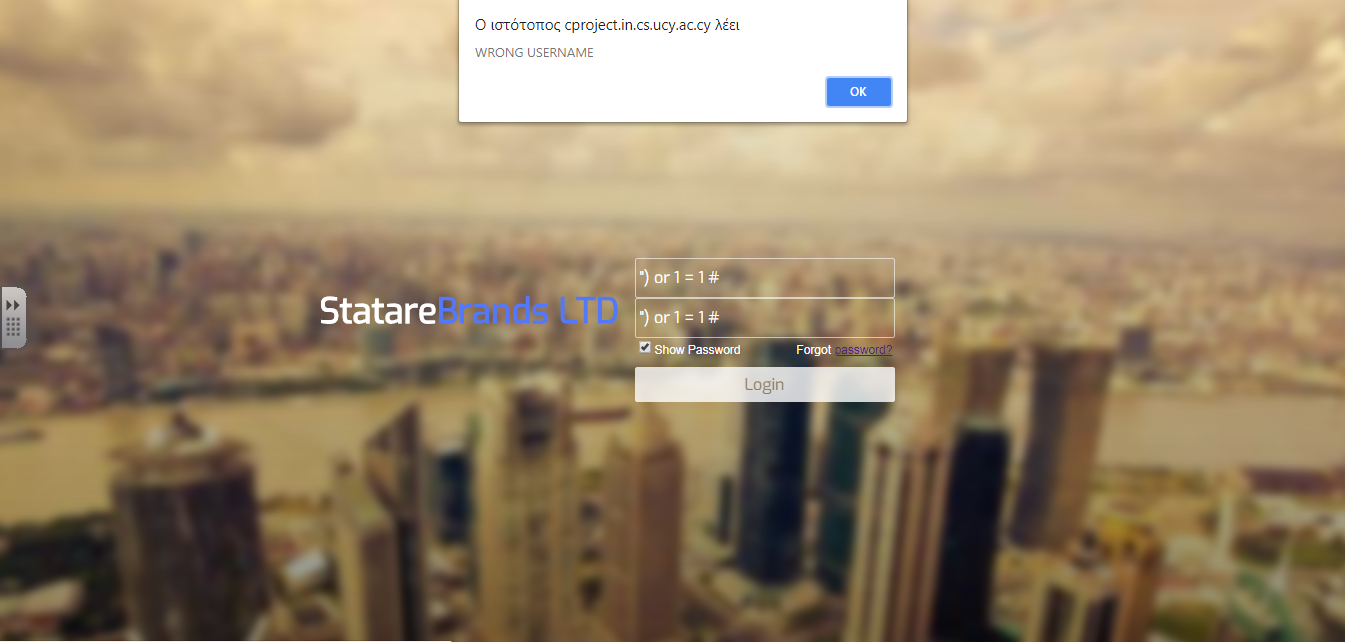


Second example for SQL injection at Log in screen:

Before pressing the button Login:



After pressing the button Login:



Javascipt Injection:

To prevent javascript injection we have used two different approaches. The first approach that we used was to add constraints to the input areas that a manager or an employee can insert text and the second approach we used was the method strip\_tags(input) to filter html characters out at the input of the user.

An implementation of these two approaches is the following:

1. First approach:

$FirstName = strip\_tags($FirstName);

If the user inserts at any input field <script>…</script> the tags of script are not be written in the database.

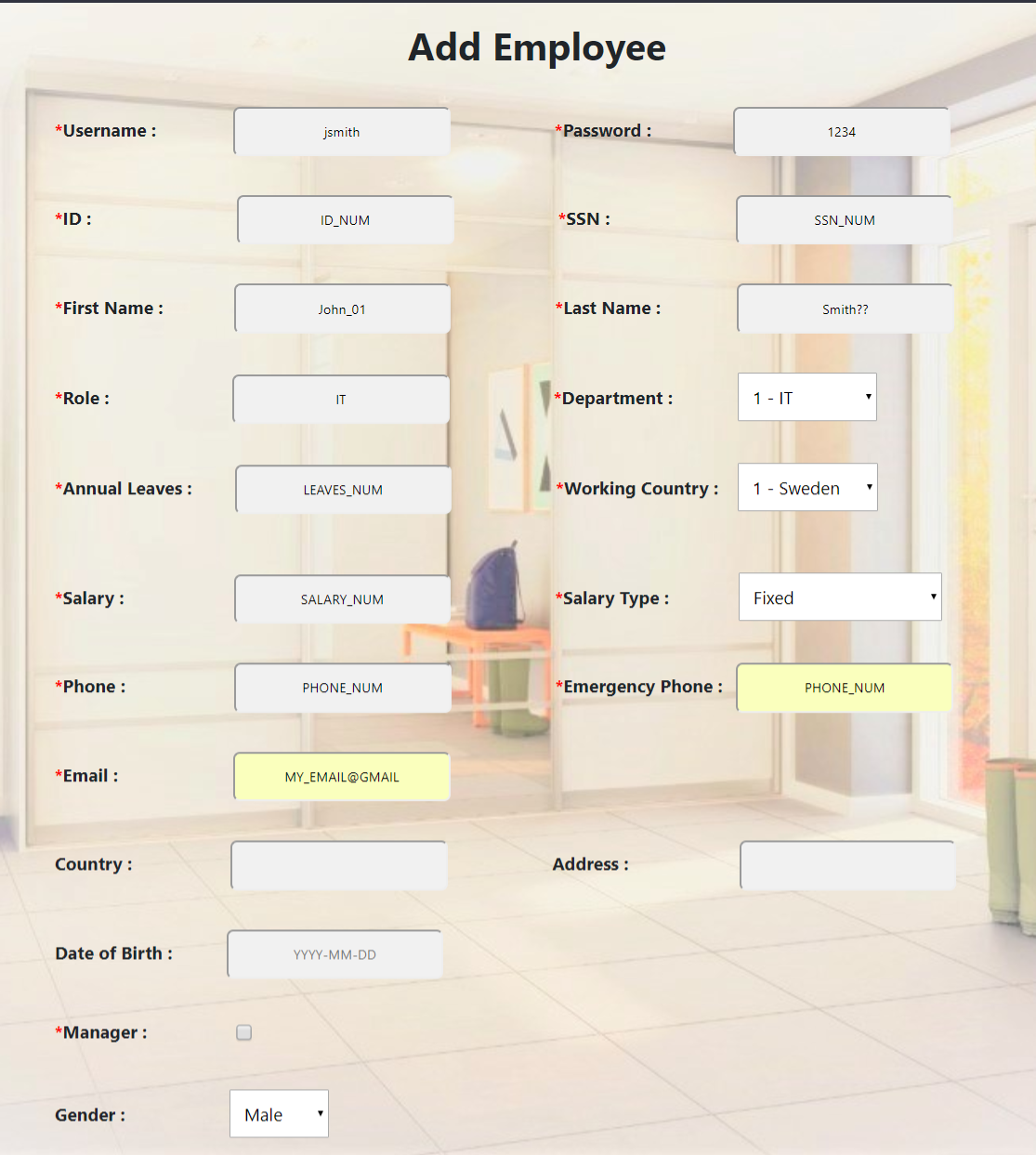
1. Second approach:

if ( !preg\_match("/^[a-zA-Z ]\*$/", $FN) ) { … } ( Accepts only characters )

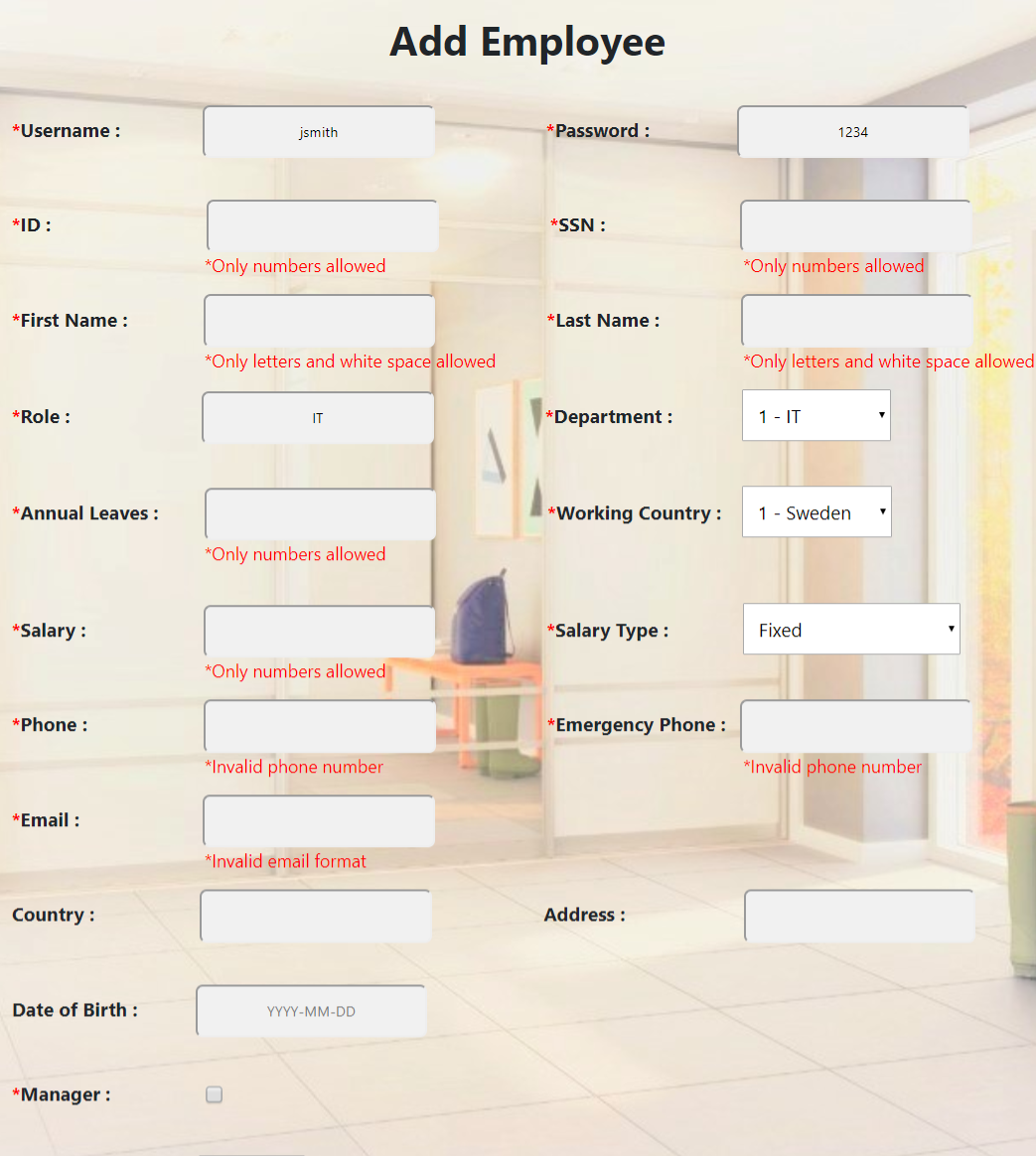
if ( !preg\_match("/^\d+$/", $Phone) ) { … } ( Accepts only numbers )

if (!filter\_var($Email, FILTER\_VALIDATE\_EMAIL)) { … } ( Accepts only validate email input )

Before click the button Save:



After click the button Save:



**Software license:**

We will not use any open source software license for our system. The reasons are the following:

1. We are using a php file (password.php) with MIT license for encoding the password of users.
2. The company which we are developing its system prefers to keep our repository private in GitHub.

Because the MIT license is permissive does not force us to make our repository public, we can keep it private.