



**Katsi Eleni**

**Mina Mariana**

**Agroti Louiza**

**Kouppi Maria**

API DOCUMENTATION

**Contents**

[**File: contact\_employee.php** 4](#_Toc513836368)

[**File: contact\_manager.php** 4](#_Toc513836369)

[**File: forgot\_pass.php** 4](#_Toc513836370)

[**File: index.html** 4](#_Toc513836371)

[**File: payroll\_report.php** 4](#_Toc513836372)

[**File: view\_hours\_employee.php** 4](#_Toc513836373)

[**File: view\_hours\_manager.php** 5](#_Toc513836374)

[**File: employee\_status\_manager.php** 6](#_Toc513836375)

[**File: add\_employee\_function.php** 7](#_Toc513836376)

[**File: choose\_employee\_function.php** 9](#_Toc513836377)

[**File: update\_edit\_employees.php** 9](#_Toc513836378)

[**File: clock\_in\_employee.php** 10](#_Toc513836379)

[**File: break\_employee.php** 11](#_Toc513836380)

[**File: clockIn\_employee.php** 12](#_Toc513836381)

[**File: clockOut\_employee.php** 12](#_Toc513836382)

[**File: returnFromBreak\_employee.php** 13](#_Toc513836383)

[**File: clock\_in\_manager.php** 14](#_Toc513836384)

[**File: break\_manager.php** 15](#_Toc513836385)

[**File: clockIn\_manager.php** 16](#_Toc513836386)

[**File: clockOut\_manager.php** 16](#_Toc513836387)

[**File: returnFromBreak\_manager.php** 17](#_Toc513836388)

[**File: EmployeeDashboard.html** 18](#_Toc513836389)

[**File: Manager\_dashboard.html** 18](#_Toc513836390)

[**File: edit\_profile\_employee.php** 20](#_Toc513836391)

[**File: edit\_profile\_manager.php** 21](#_Toc513836392)

[**File: delete\_employee\_.php** 21](#_Toc513836393)

[**File: delete\_employee.php** 21](#_Toc513836394)

[**File: update\_profile\_employee.php** 22](#_Toc513836395)

[**File: update\_profile\_manager.php** 23](#_Toc513836396)

[**File: employee\_leave.php** 23](#_Toc513836397)

[**File: manager\_leave.php** 23](#_Toc513836398)

[**File: average\_per\_week\_manager.php** 23](#_Toc513836399)

[**File: employee\_view\_request.php** 24](#_Toc513836400)

[**File: api.php** 24](#_Toc513836401)

### **File: contact\_employee.php**

/\*Takes an employee out of the system\*/

function logout\_function();

### **File: contact\_manager.php**

/\*Takes an employee out of the system\*/

function logout\_function();

### **File: forgot\_pass.php**

/\*\*

\* Creates a string which consists by ten random characters.

\*

\* @param int $length The length of the string.

\*

\* @return string The new random string

\*/

function generateRandomString($length = 10);

### **File: index.html**

/\*shows to the employee the password that he/she inserted\*/

function showPassword();

### **File: payroll\_report.php**

/\*Takes an employee out of the system\*/

function logout\_function();

### **File: view\_hours\_employee.php**

/\*\*

\* Creates the time using hours, minutes and seconds.

\*

\* @param int $hours2 The number of the hours.

\*

\* @param int $min2 The number of the minutes.

\*

\* @param int $sec2 The number of the seconds.

\*

\* @param time $newTime2 The time which will returned.

\*

\* @return time The calculated time

\*/

function calculate\_hours\_length($hours2, $min2, $sec2, $newTime2);

/\*\*

\* Prints the data to a table.

\*

\* @param table $row The table of data that will be printed.

\*

\* @param time $clockout The time in which the employee did clocked out.

\*

\* @param time $newTime2 The calculated time.

\*

\* @return void

\*/

function print\_table($row, $clockout, $newTime2);

/\*Takes an employee out of the system\*/

function logout\_function();

### **File: view\_hours\_manager.php**

/\*\*

\* Creates the time using hours, minutes and seconds.

\*

\* @param int $hours2 The number of the hours.

\*

\* @param int $min2 The number of the minutes.

\*

\* @param int $sec2 The number of the seconds.

\*

\* @param time $newTime2 The time which will returned.

\*

\* @return time The calculated time

\*/

function calculate\_hours\_length($hours2, $min2, $sec2, $newTime2);

/\*\*

\* Prints the data to a table.

\*

\* @param table $row The table of data that will be printed.

\*

\* @param time $clockout The time in which the employee did clocked out.

\*

\* @param time $newTime2 The calculated time.

\*

\* @return void

\*/

function print\_table($row, $clockout, $newTime2);

/\*Takes an employee out of the system\*/

function logout\_function();

### **File: employee\_status\_manager.php**

/\*\*

\* Print employees who are on leave

\* @param $leave DB's row for the current employee

\*/

function onLeave($leave);

/\*\*

\* Print employees who are clocked in

\* @param $row DB's row for the current employee

\*/

function clockedIn($row);

/\*\*

\* Print employees who are clocked out

\* @param $row DB's row for the current employee

\*/

function clockedOut($row);

/\*\*

\* Print employees who are on break

\* @param $row DB's row for the current employee

\*/

function onBreak($row);

/\*\*

\* Print employees who are clocked out, at the case that they did not pressed Clock in button for the current date

\* @param $conClockedOut Manager's employees

\* @param $in DB's row for employees that have pressed any of the buttons Clock in, Clock out, Break, Return from Break for the current date or they have already been clocked in or on break from a previous date

\* @param $leave DB's row for employees that are on leave

\*/

function findClockOut($conClockedOut,$in,$leave);

/\*\*

\* Print error message related to error connecting with database

\*/

function printError();

/\*\*

\* Print employees' username,surname,name of the current manager

\* (who is logged in) that are either clocked in,clocked out or on break

\* @param $in DB's row for the current employee

\*/

function findState($in);

### **File: add\_employee\_function.php**

/\*\*

\* Prints successfully message related to data inserted correctly in the database

\*/

function print\_success();

/\*\*

\* Prints error message related to if is something wrong with the used queries

\*/

function print\_error();

/\*\*

\* Prints an error message related to that the given department does not exist in the given country

\*/

function print\_error\_dept();

/\*\*

\* Prints an error message related to that the given username already exists in the database

\*/

function print\_error\_username();

/\*\*

\* @param $conn The connection with the database

\* @param $DeptNum The number of the department in which employee will belong to

\* @param $WorkC The number of the country in which employee will be working to

\* @return bool True if the given number of the department exists in the given country, otherwise false

\*/

function checkDeptCountry($conn,$DeptNum,$WorkC);

/\*\*

\* Checks if the username of employee, which manager gave does not already exists in the database. If the username already exists in the database returns false, otherwise returns true

\* @param $conn The connection with the database

\* @param $Username The given username (the username of employee)

\* @param $UsernameManager The username of manager who adds employee in the database

\* @return bool True if the username that manager gave does not exist in the database

\*/

function checkUsername($conn,$Username,$UsernameManager);

/\*\*

\* Insert in the database the data of an employee that manager gave

\* @param $conn The connection with the database

\* @param $WorkCountry The number of country in which the given employee will works to

\* @param $Email The email of the given employee

\* @param $DepartmentNum The number of department in which the given employee will belong to

\* @param $Manager Represents if the given employee will be a manager of a department

\* @param $Username The username of the given employee

\* @param $Hashed The hashed value of the given password of employee

\* @param $ID The ID of the given employee

\* @param $SSN The SSN of the given employee

\* @param $FirstName Name of the given employee

\* @param $LastName Surname of the given employee

\* @param $Role The role of the given employee at the company

\* @param $Salary The salary of the given employee

\* @param $SalaryType Type of employee's salary (e.g. Fixed, Fixed with Overtime, Part Time)

\* @param $Phone The phone of the given employee

\* @param $EmergencyPhone An emergency phone of the given employee

\* @param $Country The country where the given employee is from

\* @param $Address The address of the given employee

\* @param $DateofBirth The date of birth of the given employee

\* @param $Gender The gender of the given employee

\* @param $UsernameManager The username of manager who adds employee in the database

\* @param $Password\_len The length of the original password of the given employee

\* @param $Leaves Number of annual leaves that the given employee can does

\*/

function insertDB($conn,$WorkCountry,$Email, $DepartmentNum, $Manager, $Username, $Hashed, $ID, $SSN, $FirstName, $LastName, $Role, $Salary, $SalaryType, $Phone, $EmergencyPhone, $Country, $Address, $DateofBirth, $Gender, $UsernameManager, $Password\_len, $Leaves);

### **File: choose\_employee\_function.php**

/\*\*

\* Prints successful message related to employee selected successfully. Also goes to page edit\_employee.php where the data of the selected employee appear on the page

\*/

function print\_success();

/\*\*

\* Prints an error message related to a problem with the used query

\*/

function print\_error();

### **File: update\_edit\_employees.php**

/\*\*

\* Prints error message related to a problem of the used queries

\*/

function print\_error();

/\*\*

\* Prints error message related to given department and country

\*/

function print\_error\_dept\_country();

/\*\*

\* Prints successful message related to data updated correctly in the database

\*/

function print\_success();

/\*\*

\* Updates the number of employees in the given department and country

\* @param $conn The connection with the database

\* @param $OldDeptNum The number of department in which employee belonged to

\* @param $OldCountryNum The number of the country in which employee was working to

\* @param $DeptNum The new number of department in which employee belongs to

\* @param $WorkC The new number of the country in which employee works to

\*/

function updateNumEmployees($conn,$OldDeptNum,$OldCountryNum,$DeptNum,$WorkC);

/\*\*

\* Gets the data of an employee that manager can change and update the given data in the database

\* @param $conn The connection with the database

\* @param $emplUsername The username of employee

\* @param $Leaves The number of annual leaves that employee can do

\* @param $WorkC The number of the country in which employee works to

\* @param $DeptNum The number of department in which employee belongs to

\* @param $Role The role of employee in the company (e.g. IT,Programmer,etc)

\* @param $Salary The salary of employee

\* @param $SalaryType Type of employee's salary according how many and how often employee works (e.g. Fixed, Fixed with Overtime, Part Time)

\*/

function updateEmployee($conn,$emplUsername,$Leaves, $WorkC, $DeptNum, $Role, $Salary, $SalaryType);

### **File: clock\_in\_employee.php**

/\*\*

\* Find the current state of employee. Employee can be either clocked in,

\* clocked out or on break

\* @param $conState DB's row that represents the last time that employee clicked a button

\* @return string The current state of employee

\*/

function findState($conState);

/\*\*

\* Prints an error message that is related to error connecting with database

\*/

function printError();

Javascript’s functions:

/\*\*

\* Calls the current php file again

\*/

function myFunction() ;

/\*\*

\* Goes to logout\_employee.php file

\*/

function logout\_function();

/\*\*

\* Goes to EmployeeDashboard.html file

\*/

function dashboard();

### **File: break\_employee.php**

/\*\*

\* Prints an error message related to the constraint that an employee can not press Break without being clocked in

\*/

function print\_error\_break\_in();

/\*\*

\* Prints an error message related to the constraint that an employee can not press Break up to one time if he/she has not pressed Return from Break first

\*/

function print\_error\_break\_ret();

/\*\*

\* Update the state of the employee to break

\* @param $conn The connection with the database

\* @param $c\_state The new state of the employee, which is on break

\* @param $Username The username of the employee

\*/

function update\_breakState($conn,$c\_state,$Username);

/\*\*

\* Update the time of break

\* @param $conn The connection with the database

\* @param $Username The username of the employee

\* @param $clockOut The default value of column ClockOut in the database

\* @param $flag2 Used to check if employee is clocked in from a previous date or not

\*/

function update\_break($conn,$Username,$clockOut,$flag2);

### **File: clockIn\_employee.php**

/\*\*

\* Prints an error message related to a problem with the used queries

\*/

function print\_error();

/\*\*

\* Prints an error message related to the constraint that an employee can not press Clock in button if he/she has already been \* clocked in

\*/

function print\_error\_clockIn();

/\*\*

\* Prints an error message related to the constraint that an employee can not press Clock in button if he/she is on break, without pressing first the button Return from Break

\*/

function print\_error\_clockIn\_retBreak();

/\*\*

\* Update the state of the employee to clock in

\* @param $conn The connection with the database

\* @param $Username The username of the employee

\* @param $c\_state The new state of the employee, which is clocked in

\*/

function insert\_clockin($conn,$Username,$c\_state);

### **File: clockOut\_employee.php**

/\*\*

\* Prints an error message related to that an employee tries to click the button Clock out as he/she is on break, without pressing first the button Return from Break

\*/

function print\_error\_clockOut\_retBreak();

/\*\*

\* Prints an error message related to that an employee tries to click the button Clock out as he/she is not clocked in

\*/

function print\_error\_clockOut\_clockIn();

/\*\*

\* Update the state of the employee to clock out

\* @param $conn The connection with the database

\* @param $c\_state The new state of the employee, which is clock out

\* @param $Username The username of the employee

\*/

function update\_clockOutState($conn,$c\_state,$Username);

/\*\*

\* Update the time employee presses the button Clock Out

\* @param $conn The connection with the database

\* @param $flag2 Used to check if employee is clocked in from a previous date

\* @param $Username The username of the employee

\* @param $clockOut The default value of column ClockOut in the database

\*/

function update\_clockOut($conn,$flag2,$Username,$clockOut);

### **File: returnFromBreak\_employee.php**

/\*\*

\* Prints an error message related to that an employee tries to click the button Return from Break without press the button \*Clock in first

\*/

function print\_error\_retbreak\_in();

/\*\*

\* Prints an error message related to that an employee tries to click the button Return from Break up to one time without pressing first the button Break

\*/

function print\_error\_retBreak();

/\*\*

\* Prints an error message related to that an employee tries to click the button Return from Break without pressing first the button Break

\*/

function print\_error\_ret\_break();

/\*\*

\* Calculates the last break length of the employee

\* @param $row\_break\_len Contains the time that employee clicks the button Return from Break and the time that employee pressed the button Break

\* @return int Returns how much time employee was in break

\*/

function calculate\_break\_length($row\_break\_len);

/\*\*

\* Insert in the database how much time employee was on break for the last time that he/she did a break

\* @param $conn The connection with the database

\* @param $break The length of the break

\* @param $flag2 Used to check if employee is clocked in from a previous date or not

\* @param $Username The username of the employee

\* @param $clockOut The default value of column ClockOut in the database

\*/

function insert\_breakLength($conn,$break,$flag2,$Username,$clockOut);

/\*\*

\* @param $conn The connection of the database

\* @param $flag2 Used to check if employee is clocked in from a previous date or not

\* @param $break The length of the break

\* @param $Username The username of the employee

\* @param $clockOut The default value of column ClockOut in the database

\*/

function update\_retFromBreak($conn,$flag2,$break,$Username,$clockOut);

### **File: clock\_in\_manager.php**

/\*\*

\* Find the current state of manager. Manager can be either clocked in,

\* clocked out or on break

\* @param $conState DB's row that represents the last time that manager clicked a button

\* @return string The current state of manager

\*/

function findState($conState);

/\*\*

\* Prints an error message that is related to error connecting with database

\*/

function printError();

Javascript’s functions:

/\*\*

\* Calls the current php file again

\*/

function myFunction() ;

/\*\*

\* Goes to logout\_manager.php file

\*/

function logout\_function();

/\*\*

\* Goes to manager\_dashboard.html file

\*/

function dashboard();

### **File: break\_manager.php**

/\*\*

\* Prints an error message related to the constraint that a manager can not press Break without being clocked in

\*/

function print\_error\_break\_in();

/\*\*

\* Prints an error message related to the constraint that a manager can not press Break up to one time if he/she has not pressed Return from Break first

\*/

function print\_error\_break\_ret();

/\*\*

\* Update the state of the manager to break

\* @param $conn The connection with the database

\* @param $c\_state The new state of the manager, which is on break

\* @param $Username The username of the manager

\*/

function update\_breakState($conn,$c\_state,$Username);

/\*\*

\* Update the time of break

\* @param $conn The connection with the database

\* @param $Username The username of the manager

\* @param $clockOut The default value of column ClockOut in the database

\* @param $flag2 Used to check if manager is clocked in from a previous date or not

\*/

function update\_break($conn,$Username,$clockOut,$flag2);

### **File: clockIn\_manager.php**

/\*\*

\* Prints an error message related to a problem with the used queries

\*/

function print\_error();

/\*\*

\* Prints an error message related to the constraint that a manager can not press Clock in button if he/she has already been \* clocked in

\*/

function print\_error\_clockIn();

/\*\*

\* Prints an error message related to the constraint that a manager can not press Clock in button if he/she is on break, without pressing first the button Return from Break

\*/

function print\_error\_clockIn\_retBreak();

/\*\*

\* Update the state of the manager to clock in

\* @param $conn The connection with the database

\* @param $Username The username of the manager

\* @param $c\_state The new state of the manager, which is clocked in

\*/

function insert\_clockin($conn,$Username,$c\_state);

### **File: clockOut\_manager.php**

/\*\*

\* Prints an error message related to that a manager tries to click the button Clock out as he/she is on break, without pressing first the button Return from Break

\*/

function print\_error\_clockOut\_retBreak();

/\*\*

\* Prints an error message related to that a manager tries to click the button Clock out as he/she is not clocked in

\*/

function print\_error\_clockOut\_clockIn();

/\*\*

\* Update the state of the manager to clock out

\* @param $conn The connection with the database

\* @param $c\_state The new state of the manager, which is clock out

\* @param $Username The username of the manager

\*/

function update\_clockOutState($conn,$c\_state,$Username);

/\*\*

\* Update the time manager presses the button Clock Out

\* @param $conn The connection with the database

\* @param $flag2 Used to check if manager is clocked in from a previous date

\* @param $Username The username of the manager

\* @param $clockOut The default value of column ClockOut in the database

\*/

function update\_clockOut($conn,$flag2,$Username,$clockOut);

### **File: returnFromBreak\_manager.php**

/\*\*

\* Prints an error message related to that a manager tries to click the button Return from Break without press the button \*Clock in first

\*/

function print\_error\_retbreak\_in();

/\*\*

\* Prints an error message related to that a manager tries to click the button Return from Break up to one time without pressing first the button Break

\*/

function print\_error\_retBreak();

/\*\*

\* Prints an error message related to that a manager tries to click the button Return from Break without pressing first the button Break

\*/

function print\_error\_ret\_break();

/\*\*

\* Calculates the last break length of the manager

\* @param $row\_break\_len Contains the time that manager clicks the button Return from Break and the time that manager pressed the button Break

\* @return int Returns how much time manager was in break

\*/

function calculate\_break\_length($row\_break\_len);

/\*\*

\* Insert in the database how much time manager was on break for the last time that he/she did a break

\* @param $conn The connection with the database

\* @param $break The length of the break

\* @param $flag2 Used to check if manager is clocked in from a previous date or not

\* @param $Username The username of the manager

\* @param $clockOut The default value of column ClockOut in the database

\*/

function insert\_breakLength($conn,$break,$flag2,$Username,$clockOut);

/\*\*

\* @param $conn The connection of the database

\* @param $flag2 Used to check if manager is clocked in from a previous date or not

\* @param $break The length of the break

\* @param $Username The username of the manager

\* @param $clockOut The default value of column ClockOut in the database

\*/

function update\_retFromBreak($conn,$flag2,$break,$Username,$clockOut);

### **File: EmployeeDashboard.html**

/\*

\* Logout function - Exit

\*/

function logout\_function()

### **File: Manager\_dashboard.html**

/\*

\* Logout function - Exit

\*/

function logout\_function()

/\*

\* Go to view hours page

\*/

function view\_hours()

/\*

\* Go to employee status page

\*/

function employee\_status\_manager()

/\*

\* Go to edit employee page

\*/

function choose\_employee()

/\*

\* Go to edit profile page

\*/

function edit\_profile()

/\*

\* Go to leave request page

\*/

function leave\_request()

/\*

\* Go to add employee page

\*/

function add\_employee()

/\*

\* Go to average report page

\*/

function average\_per\_week()

/\*

\* Go to view requests page

\*/

function view\_req()

/\*

\* Go to delete employee page

\*/

function delete\_employee()

/\*

\* Go to payroll report page

\*/

function payroll\_report()

### **File: edit\_profile\_employee.php**

/\*\*

\* Prints error message related to connection with database

\*/

function print\_error();

/\*\*

\* Returns an array with employee's data (Username, ID, Name, Surname,Password,

\* Birthdate, Gender, Address, Country, Phone, EmergencyPhone, Role, Salary,

\* SalaryType, SSN, Email, AnnualLeaves) to show them in screen.

\*

\* @Param $conn Connection with database

\* @Param $Username Employee's Username (Session)

\* @Param $Password Employee's Password (Session)

\*/

function get\_employee\_data($conn, $Username, $Password)

/\*\*

\* Returns an array with employee's data related to Department table

\* (Department Name and Working Country) to show them in screen.

\*

\* @Param $conn Connection with database

\* @Param $Username Employee's Username (Session)

\*/

function get\_employee\_data\_depart($conn, $Username)

/\*\*

\* Logout function - Exit

\*/

function logout\_function();

### **File: edit\_profile\_manager.php**

/\*\*

\* Prints error message related to connection with database

\*/

function print\_error();

/\*\*

\* Returns an array with manager's data (Username, ID, Name, Surname, Password,

\* Birthdate, Gender, Address, Country, Phone, EmergencyPhone, Role, Salary,

\* SalaryType, SSN, Email, AnnualLeaves) to show them in screen.

\*

\* @Param $conn Connection with database

\* @Param $Username Manager's Username (Session)

\* @Param $Password Manager's Password (Session)

\*/

function get\_manager\_data($conn, $Username, $Password);

/\*\*

\* Returns an array with manager's data related to Department table

\* (Department Name and Working Country) to show them in screen.

\*

\* @Param $conn Connection with database

\* @Param $Username Manager's Username (Session)

\*/

function get\_manager\_data\_depart($conn, $Username);

/\*\*

\* Logout function - Exit

\*/

function logout\_function();

### **File: delete\_employee\_.php**

/\*\*

\* Logout function - Exit

\*/

function logout\_function();

### **File: delete\_employee.php**

/\*\*

\* Prints error message related to connection with database

\*/

function print\_error();

/\*\*

\* Prints successful message related to data updated correctly in the database

\*/

function print\_success();

/\*\*

\* Delete selected employee's data from Employee table

\* @param $conn The connection with database

\* @param $Username Selected employee's Username (Session)

\* @param $UsernameManager Manager's Username

\*/

function delete\_query($conn, $Username, $UsernameManager);

/\*\*

\* Insert selected employee's data into DeletedEmployee table to save them there

\* @param $conn The connection with database

\* @param $Username Selected employee's Username (Session)

\* @param $UsernameManager Manager's Username

\*/

function insert\_query($conn, $Username, $UsernameManager);

/\*\*

\* Get employees' username, name and surname to show them in screen

\* @param $conn The connection with database

\* @param $Username Selected employee's Username (Session)

\* @param $UsernameManager Manager's Username

\*/

function get\_usernames($conn, $Username, $UsernameManager);

### **File: update\_profile\_employee.php**

/\*\*

\* Updates employee's personal information

\* @param $conn The connection with database

\* @param $Username Employee' Username (Session)

\* @param $Hashed Hashed New Password

\* @param $Phone New Phone

\* @param $EmergencyPhone New Emergency phone

\* @param $Country New Country

\* @param $Address New Address

\* @param $Gender New Gender

\* @param $Password\_len New Password' s length

\*/

function update\_Employee($conn, $Username, $Hashed, $Phone, $EmergencyPhone, $Country, $Address, $Gender, $Password\_len);

### **File: update\_profile\_manager.php**

/\*\*

\* Updates manager's personal information

\* @param $conn The connection with database

\* @param $Username Manager' Username (Session)

\* @param $Hashed Hashed New Password

\* @param $Phone New Phone

\* @param $EmergencyPhone New Emergency phone

\* @param $Country New Country

\* @param $Address New Address

\* @param $Gender New Gender

\* @param $Password\_len New Password' s length

\*/

function update\_Manager($conn, $Username, $Hashed, $Phone, $EmergencyPhone, $Country, $Address, $Gender, $Password\_len);

### **File: employee\_leave.php**

/\*\*

\* Print the error message in alert box if employee give wrong date for leave request

\*/

function printError();

### **File: manager\_leave.php**

/\*\*

\* Print the error message in alert box if employee give wrong date for leave request

\*/

function printError();

### **File: average\_per\_week\_manager.php**

/\*\*

\*

\* A manager can export a report of the working hours of his employees. The manager gives the range of dates in which he

\* wants to export the report and the name of the report. If the export button clicked, then a report with the working hours of employees

\* has been exported in the form of manager's decision(XML or JSON).

\*/

### **File: employee\_view\_request.php**

/\*

\*

\* This file is a html file for view request functionality. At the top of the page you will see

\* the menu with the other features you can do on this web page. On the page, a table appears with all leave request

\* which he did. The last column of table is state. The employee will can see if his manager, accept or not his request.

\*

\*/

### **File: api.php**

/\*\*

\*

\* If a manager click the "view request" button in his/her dashboard, then in a page, appear a table with all

\* requests which are pended in database. Click one of the button, "reject" or "accept" or "pending" and the state of request change in database.

\*

\*/