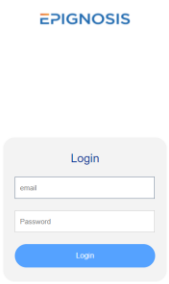
DOCUMENTATION
FOR THE AUTOMATED DAY-OFF SYSTEM

Contents

1.LOGIN MENU

The login menu is the starting page for the system. When a user enters its credentials, which is email and password, it is headed over his dashboard. This system provides two kinds of dashboards, the administrator dashboard and the employee dashboard.

The appearance of the login menu is the following:



2.USER DASHBOARD

2.1.DASHBOARD INTERFACE

When an employee is logged in, is prompted to this dashboard. This dashboard provides information for the employee about his past day-off applications order by datetime of submission descending. That information is requested from the ‘adeia’ table by getting all the entries bind to the user’s email. This information includes:

a. Date submitted.

b. Dates requested (vacation start - vacation end)

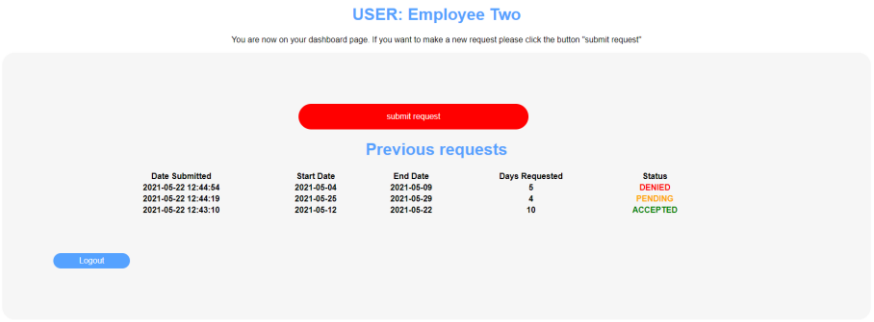
c. Days requested.

d. Status (pending/approved/rejected)

Above this list a “submit request” named red button appears. Clicking on this button user is prompted to submit a new application for day off.

Below the list a “log out” blue button appears Clicking on this button user is going to logged out of the system and return to the “login page”.

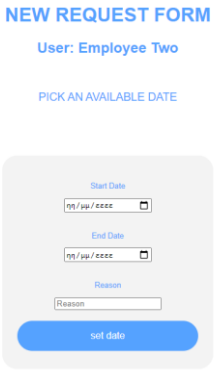
The appearance of the dashboard is the following:



2.2.NEW APPLICATION FORM

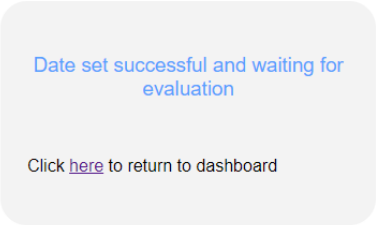
Clicking on “submit request” on the user dashboard prompts the user in this form. This form requests of user to enter the start and the end date of his day-off period and to state the reason for this request. This system has a build in check that does not allow the user to enter a start date which comes later in comparison to the end date.

The appearance of the new form dashboard is the following:

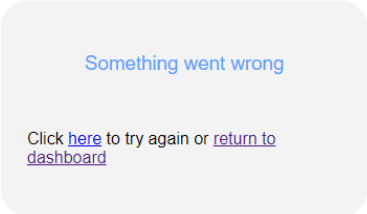


When the user is done providing this information, he clicks on the “set date” blue button below the form. After clicking it, the system adds a new application with a unique application ID in the database and sends an email from the automated email system (for this simulation this mail is hosted by the address “livadeias1@gmail.com”) to the supervisor to accept or reject the application. The format of the email is shown below.

If everything worked as planned, the user would see the following message in his screen:



On the other side, if something did not work correctly user will see this error message on his screen:



3.ADMINISTRATOR DASHBOARD

3.1 DASHBOARD INTERFACE

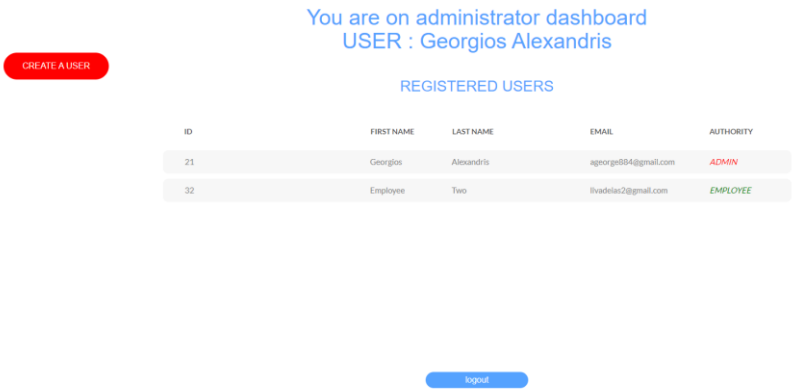
When a supervisor is logged in, is prompted to this dashboard. This dashboard provides information for the supervisor about all the employees registered in this system. That information is requested from the ‘users’ table by getting all the entries.

Above the list a “CREATE A USER” named red button appears. By clicking it, supervisor is prompted to the user creation page.

Also, in the users table every row is clickable and takes the supervisor to the user update page.

Finally, below the list a “log out” blue button appears Clicking on this button supervisor is going to be logged out the system and return to the “login page”.

The appearance of the administrator dashboard is the following:



3.2.USER CREATION PAGE

When the “CREATE A USER” button is clicked, supervisor is prompted to the user creation page. This page includes a form where supervisor is required to enter the appropriate credentials, which include:

a. First name

b. Last name

c. Email

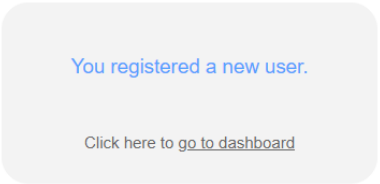
d. Password

e. Confirm password.

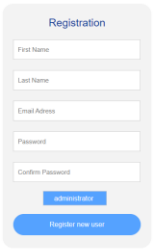
f. User type (admin/employee)

When the appropriate information is given, the supervisor clicks on the “CREATE” blue button below. Clicking this button creates a new entry with a unique ID on the ‘users’ table. The system also has a built-in safe mechanism that checks if the password has the same value with the confirm password value and checks if the given email is already used by another user.

If the process is completed successfully supervisor will view this message on his screen:



The appearance of the user creation page is the following:



3.3.USER UPDATE PAGE

When a row is clicked. The supervisor is prompted to the user update page with a custom link that provides to the script the unique ID of the requested user and requests its credentials from the ‘users’ table which are prefilled in the shown form.

The supervisor can update only these credentials of the requested user:

a. User first name

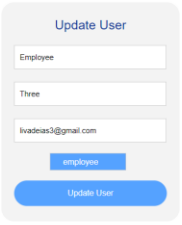
b. User last name

c. User email

d. User type (employee/admin)

After the required fields are updated, the supervisor clicks on the “Update User” blue button below the form. When this button is clicked the credentials of the user are updated in the ‘users’ database and the supervisor is prompted to the dashboard.

The appearance of the user update page is the following:



4.EMAIL SENDING PROCESS

4.1.ACCEPT/REJECT HANDLING

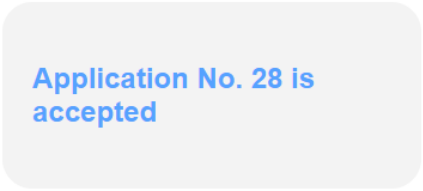
When a user submits a new application, an email is sent to the supervisor stating the dates, the reason and two custom links prompting the supervisor to the same script but with different parameters. The accept link is going to prompt the supervisor to the link appr.php?adid=ID&acc=1 and the reject link is going to prompt the supervisor to the link appr.php?adid=ID&acc=-1. The ID is the users unique ID number which is stored in the adid variable and is requested from the 'users' table. The acc value is a flag value of the situation of the employee's application and is being saved in the ACP value in the 'adeia' table. The purpose of the ACP value is to indicate application situation. The values of ACP is :

a.-1 (Application Denied)

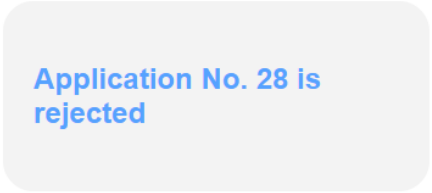
b.0 (Application Pending ,Default Value when an application is stated)

c.1 (Application Accepted)

When a supervisor accepts the application, this message appears on his screen:



In a similar way, when a supervisor accepts the application, this message appears on his screen:



The same script will also send a mail to the requested employee, informing him about his application outcome. The email will have the following format:

4.2.SETUP PROCESS

For the simulation process they have been used 4 email addresses:

1. ageorge884@gmail.com (acts as a supervisor, my personal email)

2. livadeias1@gmail.com (acts as the automated mail sender, made for the purposes of the simulation)

3. livadeias2@gmail.com (acts as an employee, made for the purposes of the simulation)

4. livadeias3@gmail.com (acts as an employee, made for the purposes of the simulation)

For the setup of the mailing server the [mail function] parameter set in the php.ini file is configured as below (This configuration is for Gmail communication and using XAMPP for hosting) :

SMTP=smtp.gmail.com

smtp_port=587

sendmail_from = livadeias1@gmail.com

sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"

Also, the [sendmail] parameter set in the sendmail.ini file must be configured as below (This configuration is for Gmail communication and using XAMPP for hosting):

smtp_server=smtp.gmail.com

smtp_port=587

error_logfile=error.log

debug_logfile=debug.log

auth_username=livadeias1@gmail.com

auth_password= (not written for safety reasons)

force_sender=livadeias1@gmail.com

Also, the variable $dir in the file named “new_req.php” must be set to the local hosting address in your working directory if is different from the directory that is already been set.

5.SYSTEM SAFETY

This system has some built-in system features. Some of them are already mentioned above. The key safety features that are developed for this system is:

- Passwords have md5 encryption.
- When registering a new user, the “password” and “confirm password” fields are required to have the same value.
- When registering or updating a user the given email must be unique and not used by another user.
- When a date range of day-off is picked by an employee the start date must be earlier than the end date.
- The employee can not access the administrator pages.
- When a user is not logged into the system can not view any page except the login page.

6.DATABASE CONFIGURATION

To organize the system, the database named ‘loginsys’ is created containing the following tables.

6.1.USERS TABLE

For the registered users organization, the ‘users’ named table contains the below information:

id (unique user ID, **PRIMARY KEY**, int type, this value is **auto incremented** which means that every new user that is registered into the system has a new ID automatically without setting it manually)

fname (First name of the user, varchar type)

lname (Last name of the user, varchar type)

email (email address of the user, varchar type)

password (password of the user account, varchar type, md5 encryption)

AUTH (authority of the user, tinyint type, when 0 user is registered as employee and when 1 the user is registered as administrator)

create_datetime (datetime of the registration, datetime type)

6.2.ADEIA TABLE

For keeping track of the day-off applications, the ‘adeia’ named table contains the below information:

ID (unique application ID, **PRIMARY KEY**, int type, this value is **auto incremented** which means that every new application that is registered into the system has a new ID automatically without setting it manually)

CL_ID (the ID value of the user who submitted the application, int type, it acts like a **FOREIGN KEY** to the id value of the ‘users’ table)

DT (datetime application submitted, datetime type)

ST_DATE (starting date of day-off, date type)

E_DATE (ending date of day-off, date type)

REA (reason for submitting this application, text type)

ACP (a custom value that keeps the process of the application, int type , when 1 means that the application is accepted , when 0 (default value) means that the application is pending for evaluation, when -1 means that the application is rejected)

7.DEVELOPER INFORMATION

This is system is based in PHP 7.0. The hosting of the webpage is handled by local server using XAMPP v8.0.6 (Control Panel v3.3.0). An Apache server is hosted on port 80. The version of Apache server is 2.4.47 . The database hosting was handled by MySQL v10.4.19-MariaDB on port 3306 using PHPMyAdmin v5.1.0 for monitor and debugging the databases. The username is “root” and the password is not set for the database because the safety it was not a priority since this system acts as a simulation.

Most of the project is developed using Jetbrains PhpStorm v2021.1.2 using also Visual Studio Code v1.56.2 for editing some of the webpage design.

8.FILE MAPPING

login.php → User Login Page

logout.php → Handles the Logout procedure

appr.php → Handles the time-off application evaluation and sends mail to the employee.

auth_session.php → Blocking the user for viewing the other web pages if is not logged into the system (the script heading the user over to the login.php form if is not logged in)

root_auth.php → Blocking the user for viewing the administrator web pages if is not logged into the system as an administrator (the script heading the user over to the login.php form if is not logged in as administrator)

dashboard.php → Employee dashboard webpage.

root_dashboard.php → Administrator dashboard webpage.

new_req.php → New day-off request form.

registration.php → User registration form.

update.php → User update form.

db.php → This script connects to the database 'loginsys' and saves the result into the variable $con. Is used ("requested") by the other scripts for easy access to the database.

style1.css,style2.css → CSS files for the webpages design.

DESIGN_FILES folder → This folder includes code that it is not developed by me and used for the table template design in "root_dashboard.php" file. The source code for this folder can be seen here: https://colorlib.com/wp/template/fixed-header-table/