

UNIVERSITY OF PATRAS

DEPARTMENT OF ELECTRICAL ENGINEERING

AND COMPUTER TECHNOLOGY

FIELD: ELECTRONICS AND COMPUTERS

ELECTRONIC APPLICATIONS LABORATORY



DIPLOMA THESIS

the student of the Department of Electrical and Computer Engineering of the
Polytechnic School of the University of Patras

PANAYI NIKOLAS

Registration Number:

1022951

**Development of a building automation control application using internet
technologies**

Thesis Number:

Supervisor:

Dr. Gialelis Ioannis

Patras, 2021

CERTIFICATION

It is certified that the Diploma Thesis on:

"Development of a building automation control application using internet technologies"

The student of the Department of Electrical Engineering and Technology
Computer

PANAYI NIKOLAS

Registration Number:

1022951

It was presented publicly and examined at the Department of Electrical and
Computer Engineering on

...../...../.....

The Supervisor shall:

Dr. Gialetis Ioannis

The Director of the Division:

Professor Paliouras Vasilios

Thesis Number:

Title: "Development of building automation control application using internet technologies"

Student: Panayi Nikolas

Supervisor: Dr. Gialelis Ioannis

THANKS

Trusting and giving young people the platform to create freely is a very difficult and important part of patenting. This is what Dr. S. provided to me. Ioannis Gialelis where from the first moment he was not afraid about how the work would go, but showed me confidence and certainty. As I walked into his office, I told him what I was thinking of doing and he said just come next week with a draft and we'll find out. So it happened. I designed a draft, I searched even more into what I want to create and what are the steps we need to follow in order to reach the desired goal. We had slowly begun to build the three main parts of the work and see what the problems were. With a lot of research and research we found solutions and today we have reached a very decent point to have created a lighting control and management platform, which would surely envy even many large companies. So, I owe a lot to this man where through this step I learned a lot in the technical part of work, but even more in what it is like from the beginning to create your own work. The patience and perseverance you need to have in order to reach the goal you have set. All this makes you a very strong person, ready for the exit life that awaits you.

Obviously, I would like to warmly thank my friends and parents for the unlimited trust, patience and support they have provided me all this time so that I can move forward and continue with my head held high and when I am lost they remind me again what my goal is.

Contents

Purpose of the Project.....	7
-----------------------------	---

Import.....	8
Main Parts.....	8
1. Platform.....	8
2. Construction.....	9
3. Platform communication with devices.....	9
Part A (Platform- HTML).....	10
Import.....	10
Account sign-in function.....	12
User Interface.....	15
Left column code.....	17
Right column code.....	20
Timer feature.....	27
Chart feature.....	30
Administrator Interface.....	34
Delete option.....	35
Επιλογή Update.....	37
Επιλογή Insert User.....	37
Search Bar.....	41
Select User Interface.....	45
Part B (Constructional-KNX).....	58
Introduction to KNX.....	58
Used devices.....	60
About Devices.....	62
Additional Elements of the KNX system.....	64
Contact Procedure.....	65
Wiring execution procedure.....	66
Device programming via ETS5.....	70
Part C (Platform communication with devices).....	76
Introduction to Node.js.....	77
Programming through Node.js.....	78
Future Additions.....	85
Conclusions.....	86

Purpose of the Project

The purpose of this diploma thesis is to create a dynamic web platform that will communicate with systems supported by the European KNX protocol, so that every user using the platform will be able to manage the lighting and electrical appliances of his home, through a familiar, pleasant online environment.

The platform is aimed at electrical installation businesses, where they will be able to provide their customers with a comfortable, reliable experience managing their home fixtures and electrical appliances. In addition, through the platform each user has his own account where he will contain the floor plan of his house and will be able to see virtually all the lighting fixtures and appliances of his home so that he can manage them. It will also be possible through a graph to see the percentage of consumption per day for each device and luminaire. This way he will be able to have a more economical experience in using the lighting and appliances he has inside his home.



Import

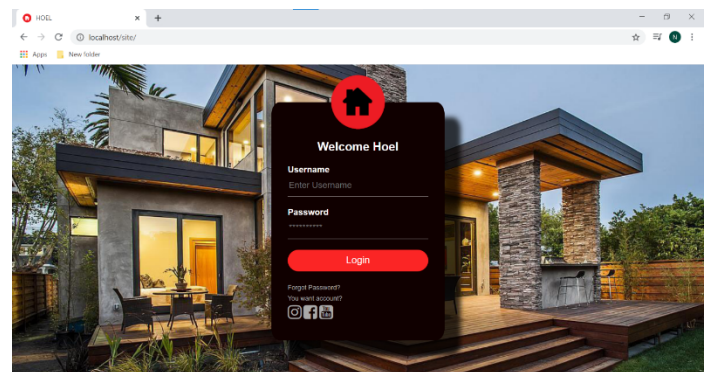
The house has always been a place of comfort, relaxation and safety for people. It was the property he had, since it contained the most important material goods but also the most important people for him, namely his family. Therefore, man always wanted to have the most comfortable and beautiful environment in his home. A house shows a lot about a person, not only in terms of the financial part but mainly in his character, his taste, his lifestyle, his temperament. All these are some elements that can be perceived by observing a house. The time he spent in it may have been limited, but it was always the place where the most important moments in his life were created.

Technology every year evolves more and more especially in the last 20 years, where it has been integrated for good into our lives and many technologies are now necessary for humans. This development has also affected the domestic space where technologies are beginning to enter our home that offer us comfort, pleasure, security and save time and money. This is what new technologies come to do in new homes with automated systems and this is what this project aims at. It seeks to link internet technologies with new technologies of intelligent automatic control systems.

Main Parts

The dissertation consists of three main parts:

1. **Platform:** Web Platform Creation using HTML, CSS, JavaScript and MySQL programming language. We will see the creation of a platform from scratch, with access to a separate environment for each user but also the creation of an admin environment with additional features, such as creating and editing the environment of each user.



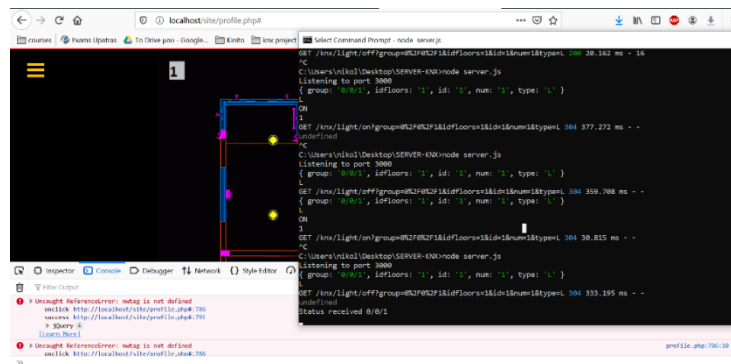
2. **Construction:** Wiring of appliances in electrical panel.

We will see how we do proper market research of the appropriate devices we will need, how the wiring of devices certified with KNX protocol is done and



then we will see how the wiring is checked and the programming of the devices using the ETS5 program.

3. Platform communication with devices: The communication of the platform with the devices comes to bind the two parts, the platform and the manufacturing part. This is achieved by node.js which is a JavaScript programming language where it creates a local server on our computer or some other microcomputer such as raspberry pi so that we can send and receive signals (information), commands from our computer to the devices and vice versa.



Part A (Platform- HTML)

Import

In the first part of our diploma thesis, we have to deal with the construction of a web platform using the web programming language HTML - CSS - JS. For the creation of the platform we have set some criteria that must be met in order for the platform to be attractive in the market.

The criteria we had set are the following:

- Easy to use for the user: To serve with easy use, the actions promised to users.

- Pleasant navigation experience: A beautiful and pleasant environment with as much use of virtual elements as possible to make it as appealing to the eye.
- Responsiveness: Responsive to each user's actions without lag.

Our goal was to achieve the above criteria.

The initial step was to make a layout with various layers in order to have an initial picture of what the platform will look like using the Photoshop program.

The steps with which we had started the implementation of the work were the creation of a platform construction plan and the creation of a database through MySQL, so that the platform is flexible and customizable. After we had the template and the database, we started writing code first in terms of appearance and then in terms of platform functionality.

Platform Name

The name of the platform is HOEL and they are the first two syllables of the words Home Electronics where it connects homes with electronic devices of a house. Essentially this is what the platform does, since we can manage the devices of a home through it.

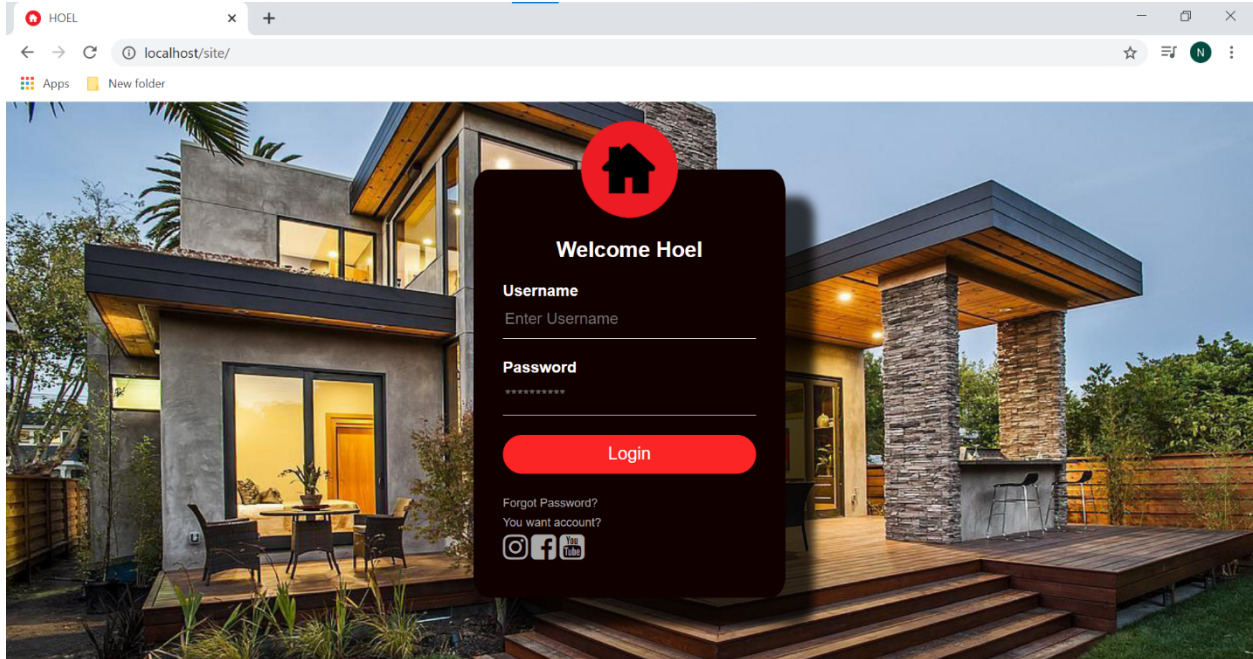
For the implementation of the platform, <https://www.w3schools.com/> platform was quite helpful and for more understanding with video projection, the platform <https://www.youtube.com/>.

Home page

Initially, the platform contains two parts. The first part is the simple user interface, where a simple user logs in with passwords, while the second part is an administrator interface. We use a password that we have set on a separate basis as administrator and contains some additional features so that it can serve users.

*Footnote: In order to be able to run our platform on the computer we must install a program that creates a local sever on the computer. In this case we have use of the XAMPP programme.

Download: <https://www.apachefriends.org/download.html>



Account sign-in function

The creation of a database to store the data of each user is necessary. For this reason, using the MySQL language, we created a database of user accounts, containing some useful data such as: id, username, password, floors, email, telephone.

Table Creation Code (MySQL)

```
CREATE TABLE login (  
  id INT(6) ,  
  On the other hand, I was able to  
  see that the Sheriff's Office was  
  not able to do so.  
  password VARCHAR(30),  
  floors INT,  
  Email Varser (50),  
  Telephone Varser (30)
```

```
)
```

Data entry (MySQL)

```
INSERT INTO login (id,username,password,floors,email,telephone)
VALUES (1,'nikolas',123,2,'nikolaspanag',233444);
```

+ Επιλογές

	id	username	password	floors	email	telephone
☐ Επεξεργασία ☐ Αντιγραφή ☐ Διαγραφή	1	nikolas	123	2	nikolaspanag	233444

Therefore, once we have created the Database for users via MySQL, we must check whether the data placed at the entrance of our platform are compatible with the user or administrator data registered in our database. This is done in the login.php file where pressing the 'Login' button on our home page (index.php) uses the 'post' method.

Αρχείο index.php

```
<div class="loginbox" id="login">
  
  <h1>Welcome Hoel</h1>
  <form action="" method="post">
    <p>Username</p>
    <input id="name" name="username" placeholder="Enter Username" type="text"><br>
    <p>Password</p>
    <input id="password" name="password" placeholder="*****" type="password"><br>
    <input name="submit" type="submit" value=" Login ">
    <span><?php echo $error; ?></span>
    <a href="#" onclick="lostpass()">Forgot Password?</a><br>
    <a href="#" onclick="singup()">You want account?</a><br>
    <a href="https://www.instagram.com/" target="_blank"><i class="fa fa-instagram"
style="font-size:30px"></i> </a>
    <a href="https://www.facebook.com/" target="_blank"><i class="fa fa-facebook-squ
are" style="font-size:30px"></i> </a>
    <a href="https://www.youtube.com/" target="_blank"><i class="fa fa-youtube-squar
e" style="font-size:30px"></i> </a>

  </div>
```

So, when the 'post' method is used, we go to the login.php file where we connect to our database and then check initially if the data we have in the socket boxes with name='username' and

name='password' are compatible in our 'owners' database. If they are not compatible, it is checked whether they are used with an administrator password so that we can go to the corresponding page with the corresponding data. If he is a normal user we go to the file 'profile.php', otherwise if he is an administrator we go to the file 'admin/index.php'.

```
<?php
include('login.php'); // Includes Login Script
if(isset($_SESSION['login_user'])){
header("location: profile.php"); // Redirecting To Profile Page
}
if(isset($_SESSION['login_admin'])){
header("location: admin/index.php"); // Redirecting To Profile Page
}

?>
```

Αρχείο login.php

```
<?php

session_start(); // Starting Session
$error = ''; // Variable To Store Error Message

if (isset($_POST['submit'])) {
if (empty($_POST['username']) || empty($_POST['password'])) {
$error = "Username or Password is invalid";
}
else
{
// Define $username and $password
$username = $_POST['username'];
$password = $_POST['password'];

// mysqli_connect() function opens a new connection to the MySQL server.
$conn = mysqli_connect("localhost", "root", "", "owners");

// SQL query to fetch information of registered users and finds user match.
$query = "SELECT username, password from login where username=? AND password=?
LIMIT 1";

// To protect MySQL injection for Security purpose
$stmt = $conn->prepare($query);
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
$stmt->bind_result($username, $password);
$stmt->store_result();
```

```

if($stmt->fetch()) //fetching the contents of the row
{
    $_SESSION['login_user'] = $username; // Initializing Session
    header("location: profile.php"); // Redirecting To Profile Page
}
else {

    ----- Gia Admin
    $username = $_POST['username'];
    $password = $_POST['password'];

    // mysqli_connect() function opens a new connection to the MySQL server.
    $conn = mysqli_connect("localhost", "root", "", "admin");

    // SQL query to fetch information of registered users and finds user match.
    $query = "SELECT username, password from login where username=? AND password=? LIMIT 1";

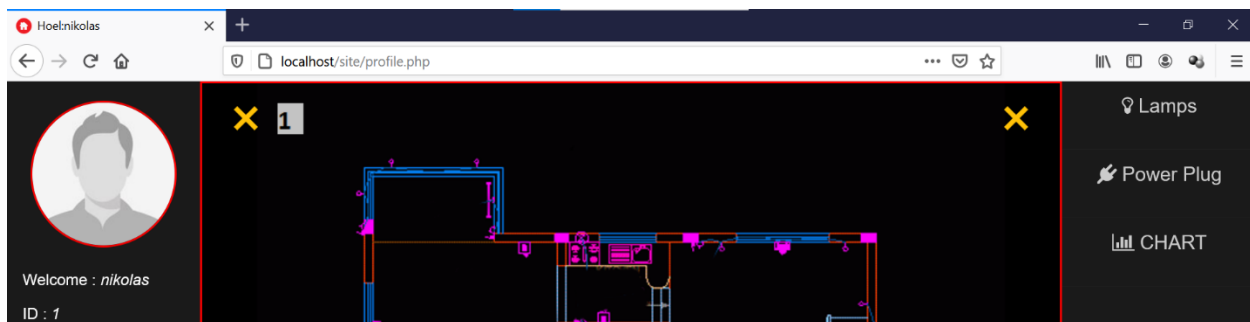
    // To protect MySQL injection for Security purpose
    $stmt = $conn->prepare($query);
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $stmt->bind_result($username, $password);
    $stmt->store_result();

    if($stmt->fetch()) //fetching the contents of the row
    {
        $_SESSION['login_admin'] = $username; // Initializing Session
        header("location: admin/index.php"); // Redirecting To Profile Page
    }
    else {
        $error = "Username or Password is invalid";
    }
}
mysqli_close($conn); // Closing Connection
}
}
?>

```

User Interface (User Password Login - profile.php File)

By entering an appropriate user code, we go to the profile.php file where the floor plan of the user's first floor house appears in the central frame.



On the left are some details of the user such as his name, id and the floors of his house. On the right are displayed functions where the user can do through the platform.

Code explanation

For our convenience we have divided the user platform into three files. A file is the profile.php where is the home file that contains the user's image. The other two files are left-sidebar.php and right-sidebar.php which is the code of the left and right columns respectively. They are loaded with the command:

```
<?php include "left-sidebar.php"?>
```

Κώδικας profile.php

```
<body bgcolor="#0000" >
  <div id="vline"></div>
  <!-- Id to opoio gemizi lista me lampes-->
  <ul id="Lamps"></ul>
  <!--- LEFT BAR ---->
  <?php include "left-sidebar.php"?>
  <!--- RIGHT BAR ---->
  <?php include "right-sidebar.php"?>

  
  <div id="lamp" style="width: 100px;">
    <!---->
  </Div >
</body>
```

```

  <!-- photo to be a part of the ---->
  <div class="image_outer_container">
```

```

        <div class="image_inner_container">
            
            <input style="display:none;" name="photo" id="myFileInput" type="fi
            le" accept="image/jpg" onchange="document.getElementById('output').src = window
            .URL.createObjectURL(this.files[0])">

        </div>
    </div>

    <div >
        <ul id="mylist" >
            <p></p>
            <p> Welcome : <i><?php echo $login_session; ?></i></p>

            <p> ID : <i><?php echo $login_id; ?></i></li>
            <p> FLOORS : <i><?php echo $login_floors; ?></i></p>
            <p></p>
            <p></p>
        </ul>
    </div>

```

We create through the HTML programming language the objects and then with the use of the CSS programming language we give the appearance of the object (eg proportions, color, etc.).

For the appearance of each object we have defined specifications inside the sc.js file and load it into the specific file with the following command:

```

<script src="js/sc.js"></script>

```

Due to the fact that the number of floors for each user is different, the creation of labels for each user must be dynamic. Through the database to see how many floors each user has and create the corresponding number of labels.

This does the following addChild() command. We take from our base the number of floors for each user and using an iterative method (for), we make a process for as many floors as we have. The procedure where it is done is as follows:

We create a variable for the object that has id=mylist. We create a div object that has id same as the floor number (i.e. i inside for). We apply the dropdown property that we have previously set in the sc.js file. We make this object a child (subelement) of the variable we originally created with id=mylist with the command .appendChild(). Later we create an object in the form of 'li' and it has as content 'floor' and the name of the floor. We give it the property if someone "clicks" on the specific object, goes to the object with id = lamps and loads the image that has id, the id of the user and floor, the floor number we have clicked. In addition, clicking on the object 'li'

deletes all children (subelements) that the object has with id=lamps. (We will see later what are the subelements of this object).

addClid() function code (left-sidebar.php)

```
function addChild(){
var w=<?php echo $login_id; ?>;

    was in;
    var a;
    var dictionary = {};
    var p=<?php echo $login_floors; ?>;
for (i = 1; i < p+1; i++) {
    var mylist=document.getElementById("mylist");

    var nwtag = document.createElement("div");
    nwtag.id = "body1"+i;
    nwtag.className = "dropdown";

    mylist.appendChild(nwtag);

    var mylist1=document.getElementById("body1"+i);

    let nwtag1 = document.createElement("li");
    nwtag1.textContent="Floor "+i;
    nwtag1.className = "pio";

    nwtag1.onclick = (
    function(t) {return function (e) {
        var list = document.getElementById("Lamps");
        var nodelist = document.getElementsByTagName("IMG").length;
        document.getElementById("myImg").src = "images/home/"+w+"-"+
+t.toString()+".png";
        //alert(nodelist);-----Na kita3o giati 7 img
        if(nodelist!=7){
            var div = document.getElementById("Lamps");
            var nodelist = div.getElementsByTagName("IMG");
            var le=nodelist.length;
            // alert("Poses lampes1:"+le);
            for(i=1; i<=le;i++){
                var list = document.getElementById("Lamps");
                list.removeChild(list.childNodes[0]);
            }
        }

        idfloors=t.toString();
        //alert(idfloors);
    }})(i);
    mylist1.appendChild(nwtag1);
};
```

```

let nwtag2= document.createElement("div");
nwtag2.id = "body2"+i;
nwtag2.className = "dropdown-content";

mylist1.appendChild(nwtag2);

var mylist2=document.getElementById("body2"+i);

let nimg = document.createElement("IMG");
nimg.setAttribute("src", "images/home/"+w+"-"+i+".png");
nimg.className = "center1";

mylist2.appendChild(nimg);

}
return mylist.innerHTML;
}

```

Right column code (right-sidebar.php)

In the right column of the user's platform, there are the features where each user has. It is possible to present the lighting fixtures and the entrances of the appliances of his home. If he chooses one of these two options, the lighting fixtures and respectively the entrances for the appliances of his house are displayed, on the central frame that is the floor plan of his house. In addition, there is the possibility of a timer where we set the day and time we want a luminaire to be turned on or off. Automatically start or pause power supply to the luminaire. Also, there is a graph that shows the consumption of the lighting fixtures of each user and through which the user can see which appliances consume more energy and therefore make a more economical use.

More detailed explanation of the above features via code.

```

<div id="sidebar-right">
<ul>
<li onmousedown =LampClick(event) class="dropbtn1"><a><i class="fa fa-lightbulb
-o" aria-hidden="true"></i> Lamps</a></li>

<li onmousedown =PlugClick(event) class="dropbtn1"><a><i class="fa fa-plug" ari
a-hidden="true"></i> Power Plug</a></li>
<li onclick='timer1()'><a href="#" ><i class=" fa fa-clock-o"
aria-hidden="true"></i>Timer</a></li>

```

```

<li><a href="#"><i aria-hidden="true"></i> Graph</a></li>
</ul>
</div>
<div id="toggle-right" onclick="toggleSidebarR(this)">
<span></span>
<span></span>
<span></span>
</div>

```

In the above code we create a div which has the properties - appearance 'sidebar-right' and in which we have created the possibilities that the user can have so that he can activate them with the "click" of an object of type li. The onclick parameter inside the object of type li allows the use of a function if the specific object is "clicked".

Lamps() function code (display and activation of luminaires-right-sidebar.php file)

```

<!------- LAMPS ----->

function lampsL() {
    clear1();
    id=<?php echo $login_id;?>;

$.ajax({
url    : 'lampsnum.php',
method : 'post',
data    : {idfloors: idfloors,id: id},
success : function(data){
if(data==0){
alert("Poses lamps:"+data);
}

for(i=1; i<=data;i++){
//alert("lampa:"+i);
$.ajax({
url    : 'lampsxy.php',
method : 'post',
data    : {idfloors: idfloors,i:i,id: id},
success : function(data1){
//alert("Lampa:"+data1);
var obj = JSON.parse(data1);

//size se sxesi me o8oni ta lampakia-----

```

```

var w = window.outerWidth;
var h = window.outerHeight;
if (w>400) { //alert(w);
x=obj.X+"%";
y=obj.Y+"%";
}
else{
x1=obj.X;
y1=obj.Y;

if(x1>50){
x2=x1-25;
x=x2+"%";
}
else{
x2=x1-5;
x=x2+"%";
}
if(y1>50){
y2=y1;
y=y2+"%";
}
else{
y2=y1-7;
y=y2+"%";
}
}
//-----
group1=obj.GroupAddress;
function createElementHtml(i,id,tagname){
var mylist0=document.getElementById("Lamps");
var nwtag1= document.createElement("div");
var nwtag01= document.createElement("span");
let nwtag = document.createElement(tagname);

nwtag.id = id;
nwtag.setAttribute("src", "images/lamp.png");
nwtag.setAttribute("width", "20");
nwtag.setAttribute("height", "20");
//nwtag.style.position = "absolute";
//nwtag.style.top=x;
//nwtag.style.left=y;
nwtag1.style.position = "absolute";
var w = window.outerWidth;
var h = window.outerHeight;

nwtag1.style.top=y;
nwtag1.style.left=x;

```

```

NWTAG01.className = "tooltiptext";
nwtag1.className = "tooltip";
nwtag01.innerHTML = "Lamp: " +obj.lampnum+"X:"+obj.X+" Y:"+obj.Y+ "\n"+"GroupAd
dress:"+obj.GroupAddress+ "\n"+"Watts:"+obj.Watts;
nwtag1.appendChild(nwtag);
nwtag1.appendChild(nwtag01);

----- Elegxos Ean einai on i off i lampes
if(object."LampOn=="0" || "objects.LampOn==""){

nwtag.classList.remove("mystyle")
}
else{
nwtag.classList.add("mystyle");
}

//-----

mylist0.appendChild(nwtag1);
nwtag.onclick= function(){
//var res = obj. GroupAddress.replaceAll("/", "-");
//alert(res);

//alert(obj.lampnum);
if (nwtag.className != "mystyle") {
nwtag.classList.add("mystyle")
//alert("You n clicked Lamp: On");
LampON(obj.GroupAddress, obj.idfloors, two.Id, id, two.lampnum,"L");
}

else {
nwtag.classList.remove("mystyle")
LampOff(obj.GroupAddress, obj.idfloors, two.Id, id, two.lampnum,"L");
}
}
return mylist0.innerHTML;
}
createElementHtml(obj.idfloors,"floor"+obj.idfloors,"IMG");
var temp = document.getElementById("floor"+obj.idfloors);
temp.onclick = (
function(t) {
nwtag.classList.add("mystyle");
return function (e) {
//alert("You just clicked Lamp: " + t);
};
})(obj.idfloors);

```

```
}});}});}
```

When a user clicks on the 'Lamps' button we have created, a function is called through the 'onclick' parameter. In this case the function 'lamps1()' is called. The function stores the id of the present user within an id variable. Then, using ajax command, we go to the 'lampsnum.php' file in order to retrieve from our database, the total number of objects (luminaires) that this user has (id) on the specific floor (idfloors).

Data recovery process via **database lampsnum.php** (mysql)

```
<?php
$id = $_POST['id'];
$idfloors = $_POST['idfloors'];
//echo $idfloors;
$conn = mysqli_connect('localhost' , 'root' , '' , 'owners');
$query = "SELECT count(*) as total from lamps where floor=".$idfloors." and user_id=".$id."";
$res_sql = mysqli_query($conn, $query);
$row = mysqli_fetch_assoc($res_sql);
$lo = $row['total'];

echo $lo

?>
```

In this code we recall the id and idfloor variables, after the specific file has been enabled through the 'Post' method. We connect to our database (owners) and then restore the total number of elements - luminaires we have in the database with the specific data (id,idfloors) with the 'Select' command of MySQL.

After we have reset the total number of luminaires that a user has on that floor, we create an iterative method (for) with a limit of the specific number. Therefore, we call repeatedly, as many times as the total number of luminaires. In order to know at which coordinates each luminaire is located on our screen, we have a table that contains the coordinates for each luminaire and other useful information for the user. Therefore, with the 'Ajax' command we call the lampsxy.php file with a 'Post' method to which we transfer the variables idfloors (floor number), I (luminaire number) and id (user number). By using this data we connect to our database and more specifically to the panel lamps and thus we can reset the coordinates (x, y), the address it has to communicate through node.js (groupaddress), whether it is acting or not (lampsOn) and its current consumption for the specific luminaire. We retrieve the data for each item in our right-sidebar.php file. PHP returns the data as dictionary via the "Json" command, so we can have each data separately in a variable.

Reset data for each luminaire.

Αρχείο **lampsxy.php**

```
<?php
    $id = $_POST['id'];
    $idfloors = $_POST['idfloors'];
    $i = $_POST['i'];
    //echo $idfloors;
    //echo $i;
    $conn = mysqli_connect('localhost' , 'root' , '' , 'owners');
    $query = "SELECT X,Y,GroupAddress,LampOn,lampnum,Watts from lamps where
    floor=".$idfloors." and lampnum=".$i." and userid=".$id."";
    $ses_sql = mysqli_query($conn, $query);
    $row = mysqli_fetch_assoc($ses_sql);
    $lo = $row['X'];
    $lo1 = $row['Y'];
    $lo2 = $row['GroupAddress'];
    $lo3 = $row['LampOn'];
    $lo4 = $row['lampnum'];
    $lo5 = $row['Watts'];

    echo json_encode(array('id' => $id, 'idfloors' => $idfloors, 'lampnum' =>
    $lo4, 'X' => $lo, 'Y' => $lo1, 'GroupAddress' => $lo2, 'LampOn' => $lo3, 'Watts' => $lo5 ));
?>
```

Then, we create for each luminaire an element that has the coordinates as .style.top=y and as .style.left=x and some other properties, such as style, size, image overlayed, etc.

An important part is the LampON function which uses the variables group, idfloors, id, num, type (element type: luminaire or socket) and through which we call the ulron with a get method so that we can programmatically node.js communicate to the KNX-SWITCH actuator device. (The variables ulron and ulroff are first set in the code.) The LampsOff function does something similar, except it goes to the ulroff address. Later, it is explained in more detail how it works together with node.js.

LampON function (luminaire activation).

```
function LampON(group,idfloors,id,num,type){
$.ajax({
url      : ulron,
method   : 'get',
data: {
group:group,idfloors: idfloors,id:id,num:num,type:type
},
success  : function(data){},
```

```
})  
  
}
```

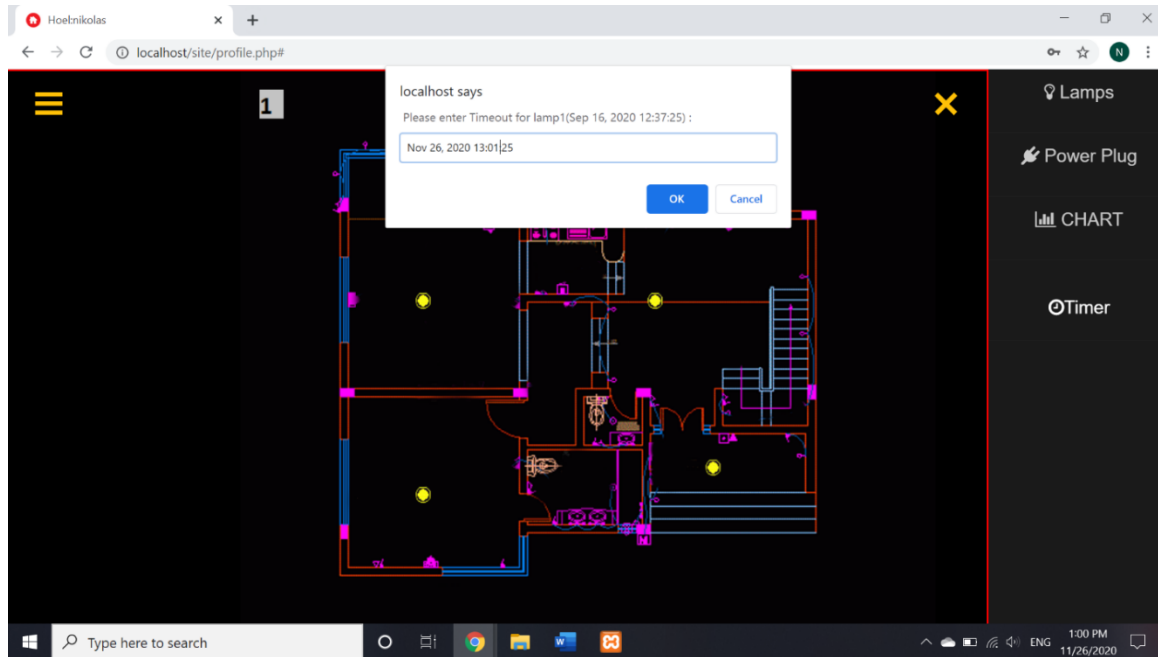
LampON function (luminaire activation).

```
function LampOff(group,idfloors,id,num,type){  
$.ajax({  
url      : ulroff,  
method   : 'get',  
data: {  
group:group,idfloors: idfloors,id:id,num:num,type:type  
},  
success  : function(data){},  
})  
  
}
```

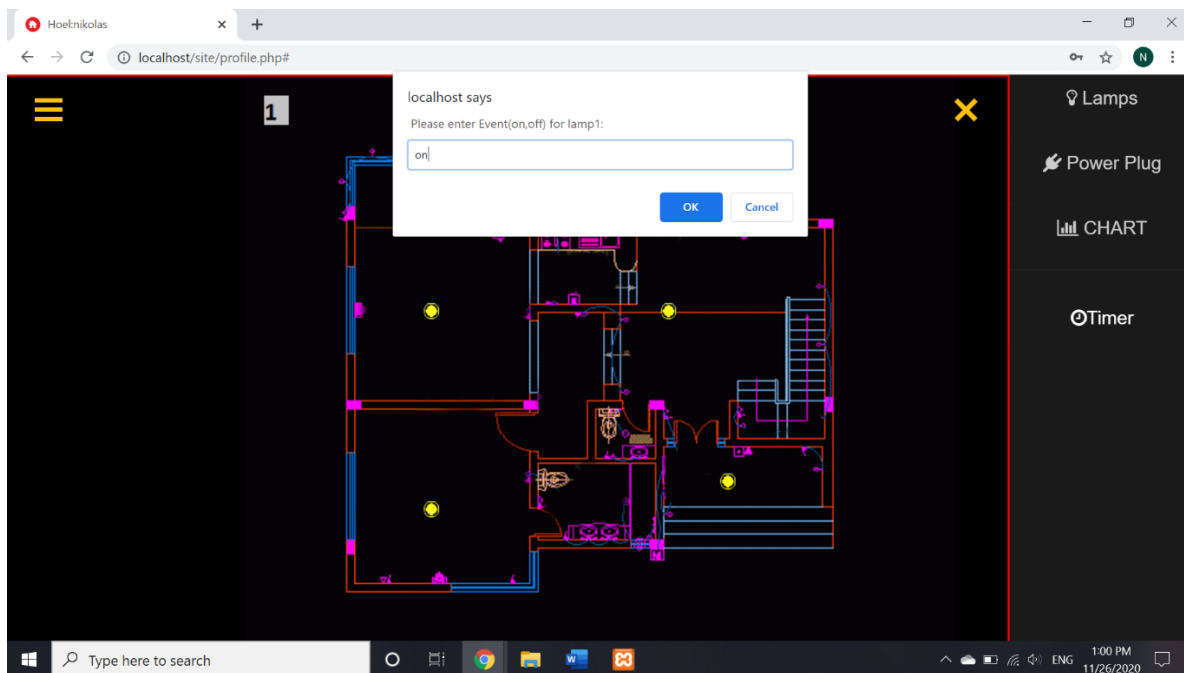
Timer feature

The timer feature is a very useful feature for every user since he can put some time he wants a luminaire to be turned on or off. This feature can be found by any user by opening the right column of features.

Initially, by pressing the "timer" feature, the lighting fixtures appear in the depiction of the floor plan of his house. In order to activate the "timer" feature, the user must press on the luminaire he wants to apply this feature. Once selected and pressed the luminaire, an input box appears that must enter the date and time that the luminaire wants to turn on or off.



Pressing 'OK' brings up another input box where he must enter whether he wants to turn 'on' or off 'off' the luminaire.



After entering the above data, if we go with our cursor for 2 seconds over the luminaire we have set the "timer" feature, it will appear how much time is left.



Code Explanation

Pressing the label "timer" calls the timerl() function, which is similar to the lampI() function with the only difference in the onclick property of each object - luminaire is called the setup() function.

setup() function

```
function setup(lampnum,datetime,group,idfloors,id,type,event1){
var countDownDate = new Date(datetime).getTime();
var x = setInterval(function() {
    var now = new Date().getTime();
    var distance = countDownDate - now;
    var days = Math.floor(distance / (1000 * 60 * 60 * 24));
    var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    ;
    var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
    var seconds = Math.floor((distance % (1000 * 60)) / 1000);
    document.getElementById("timer"+lampnum).innerHTML = days + "d " + hours + "h "
    + minutes + "m " + seconds + "s "+"Event:"+event1;

    if (days===0 && hours===0 && minutes===0 && seconds===0) {
        clearInterval(x);

        if(event1==="on"){
            LampON(group,idfloors,id,lampnum,type);
            document.getElementById("timer"+lampnum).innerHTML = "Lamp ON";
            nwtag.classList.add("mystyle");}

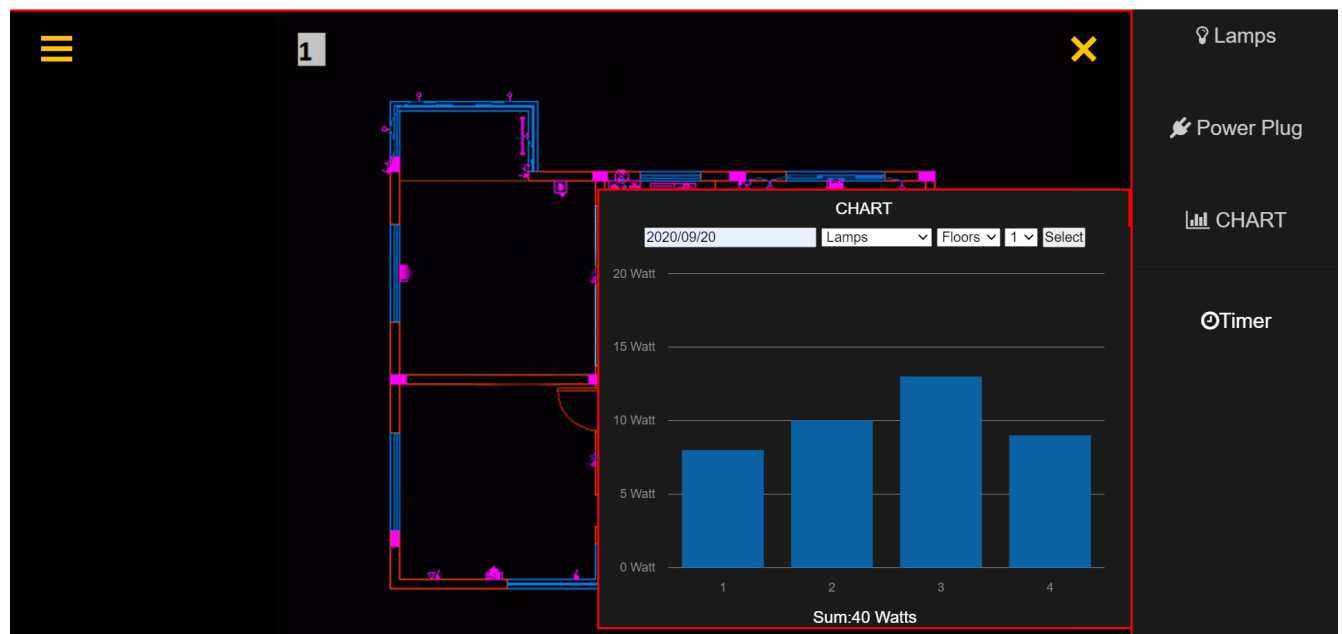
        else if(event1==="off"){
            LampOff(group,idfloors,id,lampnum,type);
            document.getElementById("timer"+lampnum).innerHTML = "Lamp OFF";
            nwtag.classList.remove("mystyle")}
    }
}
else if(distance < 0) {
    clearInterval(x);
    document.getElementById("timer"+lampnum).innerHTML = "EXPIRED"
}
}, 1000);
}
```

Along with the call of the "setup" function we transfer some data that are useful to us, such as the number of the luminaire (lampnum), date and time for activation or deactivation (datetime), device address, user floor (idfloors) and id, type if it is luminaire or socket (type) and whether to turn on (on) or off (off). Then, through the date.getTime() function, it gives us the current time

(now variable) so that we know what is the difference (distance) in the time between the time set by the user and the current time. We check until the time difference is less than zero so that we can activate the device via the Lampon or Lampoff function. When the difference between the current time and the one set by the user is greater than zero, this difference is entered in the box that has id=timer+lampum so that the user knows how much time is left at any given time.

Chart feature

An additional very useful and impressive feature is the CHART feature where the consumption made by each luminaire for a specific date set by the user is presented through a graph and the total consumption of all lighting fixtures on the floor is presented at the bottom of the graph.



Code Explanation

We create the graphics interface in the right-sidebar.php file

```
<div id="myDropdown2" class="dropdown2-content">
<p>CHART</p>
<input id="date" placeholder="YYYY/MM/DD" >
<select id="items" >
<option>Choose Items</option>
<option value="lampcon">Lamps</option>
```

```

<option value="plugcon">Plugs</option>
</select>
<select id="list" onchange="getSelectValue();">
<option value="total" selected>Total</option>
<option value="floorid">Floors</option>
</select>
<select id="mySelect" >
</select>
<button onclick=SelectValue()>Select</button>
<div id="chart"></div>

<label style="color:white;">Sum:</label><label style="color:white;" id="Sum1">
</label>
</div>
</div>

```

Pressing "label" calls the function RightClickgraph() which we find the div element that has id='myDropdown2' and we apply a 'show' property, ie we enable it to appear .

```

function RightClickgraph() {
    document.getElementById("myDropdown2").classList.toggle("show");
}

```

After filling in the necessary information, as mentioned above, and pressing the 'Select' button, the 'SelectValue()' function is called.

```

function SelectValue()
{
var id0=<?php echo $login_id; ?>;
var id = document.getElementById("mySelect").value;
var where1 = document.getElementById("list").value;
var table1 = document.getElementById("items").value;
var dat = document.getElementById("date").value;
dat1=dat[4-6]+dat[0];
console.log(id,where1,dat1);
$.ajax({

method: "POST",
data    : {id: id,where1: where1 , table1:table1,dat:dat,id0:id0},
url: "sqlchart.php",
dataType:"json",
success : function(data){
donut_chart.setData(data);
}
});
$.ajax({

```

```

method: "POST",
data    : {id: id,where1: where1 , table1:table1,dat:dat,id0:id0},
url: "sqlchartsum.php",
dataType:"json",
success : function(data1){
//alert("Sum Watts:"+data1);
var s = document.getElementById('Sum1');
s.innerHTML = data1+" Watts";
}
});
}

```

We store the data entered in the input boxes inside variables. Then, through the "ajax" command we go to the sqlchart.php file through which we get the consumption of each element from our database through MySQL .

Αρχείο sqlchart.php

```

<?php
    $id0 = $_POST['id0'];
    $id = $_POST['id'];
    $where1 = $_POST['where1'];
    $table1 = $_POST['table1'];
    $host      = "localhost";
    $username   = "root";
    $password   = "";
    $dbname     = "owners";
    $result_array = array();
    $conn = new mysqli($host, $username, $password, $dbname);
    if ($conn->connect_error) {
        die("Connection to database failed: " . $conn->connect_error);
    }

    $sql = "SELECT itemid,consu FROM ".$table1." where ".$where1."=".$id." and
userid=".$id0."";

    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            array_push($result_array, $row);
        }
    }

    $result_array= json_encode($result_array);
    echo $result_array;
?>

```

Once we have the data, we place it in the function that presents us with the data in a graph.

<https://morrisjs.github.io/morris.js/>

```
var donut_chart = Morris.Bar({
  element : 'chart',
  data:[<?php echo $chart_data; ?>],
  xkey:'itemid',
  YKEYS:['Consu','Consu'*9],
  labels:['Katanalosi', '3Ikonom'],
  hideHover:'auto',
  stacked:true
});
```

In addition, there is another file that looks very similar to the sqlchart.php file, the sqlchartsum.php file which returns us back the entire consumption of the luminaires.

Administrator Environment (Login with Administrator password)

If on the home page of our platform we enter administrator codes then through the file login.php check the database 'admin' and if it is compatible then we go to the **file admin/index.php** which contains the environment for managing the users of our database.

login.php File (Administrator Password Compatibility Checker)

If the password is not compatible with the user base, the admin base is checked.

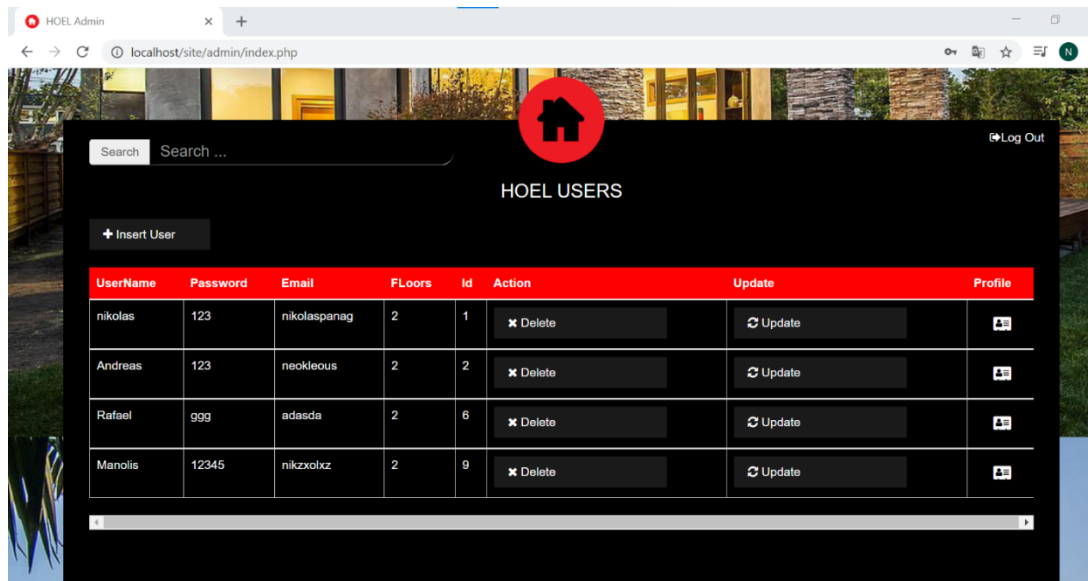
```
else {
  ----- Gia Admin
  $username = $_POST['username'];
  $password = $_POST['password'];
  $conn = mysqli_connect("localhost", "root", "", "admin");

  // SQL query to fetch information of registered users and finds user match.
  $query = "SELECT username, password from login where username=? AND password=? LIMIT 1";
  // To protect MySQL injection for Security purpose
  $stmt = $conn->prepare($query);
  $stmt->bind_param("ss", $username, $password);
  $stmt->execute();
  $stmt->bind_result($username, $password);
  $stmt->store_result();
```

```

if($stmt->fetch()) //fetching the contents of the row
{
    $_SESSION['login_admin'] = $username; // Initializing Session
    header("location: admin/index.php"); // Redirecting To Profile Pa
}
else {
    $error = "Username or Password is invalid";
}
}
}

```



By entering the administrator's interface we see to the left and up a search bar with which the administrator can more easily search for a user whose data he wants to manage. On the right we find the 'Log Out' button which disconnects the administrator from this page and returns him to the original. In addition, in the central panel we find a table that contains all the details of the users of our database such as the name, password, email and floors of each user. In addition, in the line where the data of each user is depicted, we find three additional options:

Delete option

One of them is 'delete' with which the administrator can delete a user from the database.

Code Explanation

In the fetch.php file that is created, the central table contains, in each row that is created, the id of each user and the function removes that is called if the administrator presses the delete button.

```

<button href=# onclick=removes('.$row["id"].')><i class="fa fa-remove"></i> Delete </button>

```

When the "removes" function is called, each user's id is switched to it.

Function removes (File/admin/fetch.php)

```
function removes(id){  
  
    var ans = confirm("Are You Sure You Want To Delete This Row");  
    if(ans){//if true delete row  
        window.location.assign("delete.php?id="+id);  
    }  
  
    else{//if false  
        // do nothing  
    }  
}
```

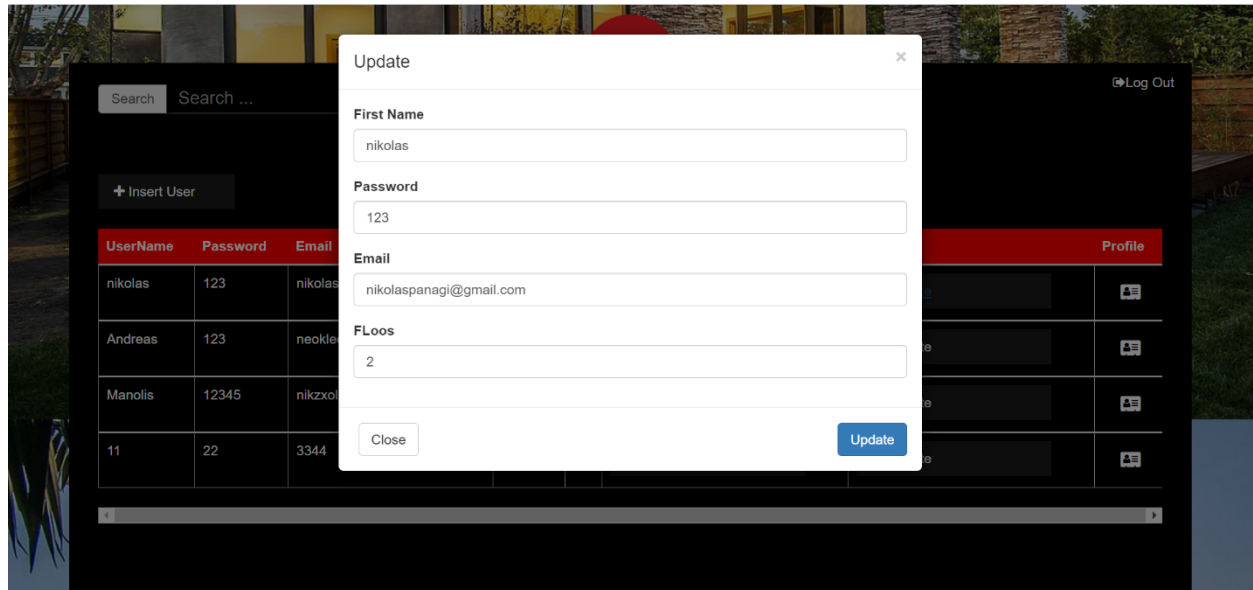
Then with this function we call the file delete.php, through which we connect to our database and delete the user who has the id same as the one contained in our variable.

Αρχείο/admin/delete.php

```
<?php  
require_once("db.php");  
if(isset($_GET['id'])){  
    $id=$_GET['id'];  
    $sql="DELETE FROM login where id= :id";  
    try{  
        $st=$conn->prepare($sql);  
        $st->bindValue(":id", $id);  
        $st->execute();  
        header("Location:index.php");  
    }  
    catch(PDOException $e){  
        echo "An Error Occured: ". $e->getMessage();  
    }  
}  
else{  
    echo "<h1>Wrong Request</h1>";  
}  
  
$conn=null;  
?>
```

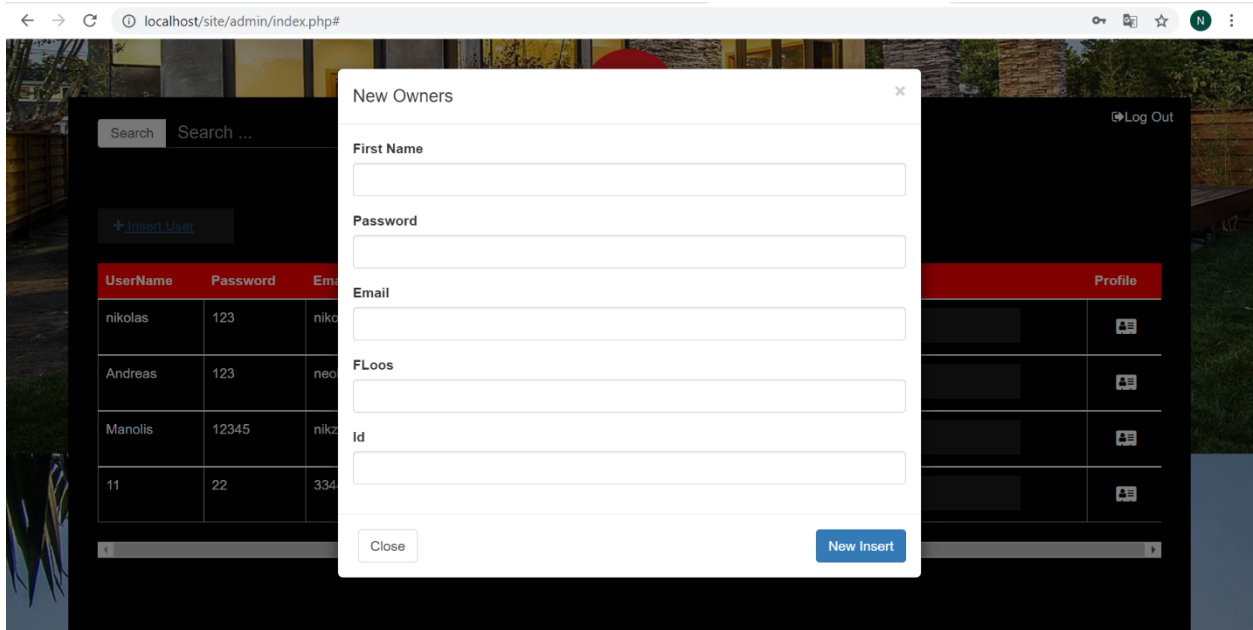
Επιλογή Update

Another option is 'update' whereby the administrator can change a user's information from the base by changing the new items in the input boxes shown as shown in the image below and pressing the update button.



Επιλογή Insert User

In addition, in the administrator interface there is an "Insert User" button with which the administrator can enter a new user in the database, filling in the user's details in the corresponding window that appears.



Code Explanation

To add a new user we have created a new window using the "bootstrap" method which we load at the beginning of our file. We create labels and inputs to enter the data we want to enter for each user.

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

New user input box

```
<!-- Modal Owners ----->
<div id="myModal0" class="modal fade" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header" >
<button type="button" class="close" data-d ismiss="modal">&times;</button>
      <h4 class="modal-title">New Owners</h4>
    </div>
    <div class="modal-body">
      <div class="form-group">
        <label>First Name</label>
        <input type="text" id="username0" class="form-control">
      </div>
      <div class="form-group">
        <label>Password</label>
        <input type="text" id="password0" class="form-control">
      </div>
    </div>
  </div>
</div>
```

```

        <div class="form-group">
            <label>Email</label>
            <input type="text" id="email0" class="form-control">
        </div>
        <div class="form-group">
            <label>FLoos</label>
            <input type="text" id="floors0" class="form-control">
        </div>
        <div class="form-group">
            <label>Id</label>
            <input type="text" id="id0" class="form-control">
        </div>

    </div>
    <div class="modal-footer">
        <a href="#" id="save0" class="btn btn-primary pull-right">New Insert</a>
        <button type="button" class="btn btn-default pull-left" data-dismiss="modal">Close</button>
    </div>
</div>
</div>

```

jQuery method

With the jQuery method, when the 'New Insert' button with id=save0 is pressed, the data we have entered in each input box are stored in variables and then, using the Ajax command, transfers the variables to the connection1.php file by post method. In this file using the MySQL 'INSERT' command, we insert a new row in the 'login' table (contains the details of all our happy users), with the data we had stored in our variables.

Method of storing data in variables, switching to php language

```

<!-- New Owners fuction -->
// append values in input fields
$(document).on('click', 'a[data-role=insr]', function(){
    $('#myModal0').modal('toggle');
});
$('#save0').click(function(){

var username = $('#username0').val();
var password = $('#password0').val();
var floors = $('#floors0').val();

```

```

var email = $('#email0').val();
var id = $('#id0').val();

$.ajax({
url      : 'connection1.php',
method   : 'post',
data     : {username : username ,password : password, email : email ,floors : f
loors , id: id},
success  : function(response){

alert("New Owners: "+username+
"\nPassword :"+password+
"\nFloors: "+floors+
"\nEmail: "+email+
"\nId: "+id);
$('#myModal0').modal('toggle');

}
});
});

```

Storage of our variables in our database

```

<?php
$connection = mysqli_connect('localhost' , 'root' , '' , 'owners');

if(isset($_POST['email'])){

$username = $_POST['username'];
$password = $_POST['password'];
$email = $_POST['email'];
$floors = $_POST['floors'];
$id = $_POST['id'];

$result = mysqli_query($connection , "INSERT INTO login(username,password, email, floors, id)VALUES('$username','$password', '$email', '$floors', '$id')");
if($result){
echo 'data updated';

}
}
?>

```

Search Bar

We create an input item in which we enter the keyword we want to search. Using the jQuery programming language we take the keyword typed by the administrator in the input element and transfer it through the "Ajax" command and post methods to the fetch.php file.

(Admin/Index.php)

```
<div>
<div id="search" style="width:40%; position:absolute;
margin-top:-8%; float: left;" class="form-group">
<div class="input-group">
<span class="input-group-addon">Search</span>
<input type="text" name="search_text" id="search_text" placeholder="Search ..."
" class="form-control" />
</div>
</div>

$(document).ready(function(){
load_data();
function load_data(query)
{
$.ajax({
url:"fetch.php",
method:"post",
data:{query:query},
success:function(data)
{
$('#result').html(data);
}
});
}

$('#search_text').keyup(function(){
var search = $(this).val();
if(search != '')
{
load_data(search);
}
else
{
load_data();
}
});
});
```

Fetch.php

```

$connect = mysqli_connect("localhost", "root", "", "owners");
$output = '';
if(isset($_POST["query"]))
{
$search = mysqli_real_escape_string($connect, $_POST["query"]);
$query = "
SELECT * FROM login
WHERE username LIKE '%" . $search . "%'
";
}
else
{
$query = "
SELECT * FROM login";
}
$result = mysqli_query($connect, $query);

```

In the fetch file a connection is made to our database (owners) where through the MySQL language and the 'Like' function it restores from our database, any elements that have in their username similar font elements to the keyword registered in the input element, within a variable \$result.

Fetch.php

```

if(mysqli_num_rows($result) > 0)
{
$output .= '<div class="flex-container">

<table id="cha1" class="table table bordered" >
<thead>
<tr >
<th>UserName</th>
<th>Password</th>
<th>Email</th>
<th>Floors</th>
<th>Id</th>
<th>Action</th>
<th>Update</th>
<th>Profile</th>

</tr>
</thead>
</div>
';
while($row = mysqli_fetch_array($result))
{
$output .= '

<div style="display:none;" class="product-image">


```

```

<div class="product-title" data-target="username">User:'. $row["username"].'</div>
v>

<p data-target="password">Pass:'. $row["password"].'</p>
<p data-target="email">Email:'. $row["email"].'</p>
<p data-target="floors">Floors:'. $row["floors"].'</p>
<p >ID:'. $row["id"].'</p>

<button href=# onclick=removes('.$row["id"].')><i class="fa fa-remove"></i> Delete </button>
<li style=" list-style-type: none;"
><a href="#" data-role="update" data-id="'. $row["id"].' "><i class="fa fa-refresh" ></i> Update</a></li>

</div>

<tr id="'. $row["id"].' ">
<td data-target="username">'. $row["username"].'</td>
<td data-target="password">'. $row["password"].'</td>
<td data-target="email">'. $row["email"].'</td>
<td data-target="floors">'. $row["floors"].'</td>
<td>'. $row["id"].'</td>
<td><ul><li><a href=# onclick=removes('.$row["id"].')><i class="fa fa-remove"></i> Delete </a> </li></ul></td>
<td><ul><li><a href="#" data-role="update" data-id="'. $row["id"].' "><i class="fa fa-refresh" ></i> Update</a></li></ul></td>

<!-------button profile----->
<td>
<form target="_blank" action="" method="post">
<input type = "hidden" id="name" name="username" value='. $row["username"].'>
<input type = "hidden" id="passwordm" name="password" value='. $row["password"].'>
'>

<button class="button" name="submit" type="submit"><i class="fa fa-address-card" ></i></button>

</form></td>

</tr>
</tbody>

</div>

';
}
echo $output;}}

```

```
else
{
echo 'Data Not Found';
}??>
```

Then the table is created where we enter all the data we have retrieved per row, of all users.

Something similar is done with the 'update' feature, only instead of the connection1.php file that we go to using the 'Ajax' command in the 'insert' feature, we go to the connection.php file where instead of the 'insert' command contains the 'update' command.

Αρχείο connection.php

```
<?php
$connection = mysqli_connect('localhost' , 'root' , '' , 'owners');
if(isset($_POST['email'])){

    $username = $_POST['username'];
    $password = $_POST['password'];
    $email = $_POST['email'];
    $floors = $_POST['floors'];
    $id = $_POST['id'];

    $result = mysqli_query($connection , "UPDATE login SET username='$username'
    , password='$password' , floors='$floors' , email = '$email' WHERE id='$id'");
    if($result){
        echo 'data updated';
    }
}??>
```

Select User Interface

The third option given to an administrator is the last icon which is written 'Profile' in each user's bar. Clicking on this feature opens a new window with the interface of the respective user and gives some additional features to a simple user.

Code Explanation

Αρχείο admin/fetch.php

```
<!--button profile-->
<td>
<form target="_blank" action="" method="post">
<input type = "hidden" id="name" name="username" value='.$row["username"].'>
<input type = "hidden" id="passwordm" name="password" value='.$row["password"].'>
'>
```

```
<button class="button" name="submit" type="submit"><i class="fa fa-address-card"></i></button>
</form></td>
```

Each time in the file fetch.php we create the lines with the details of each user is created and the specific button that has the details of each user (username and password).

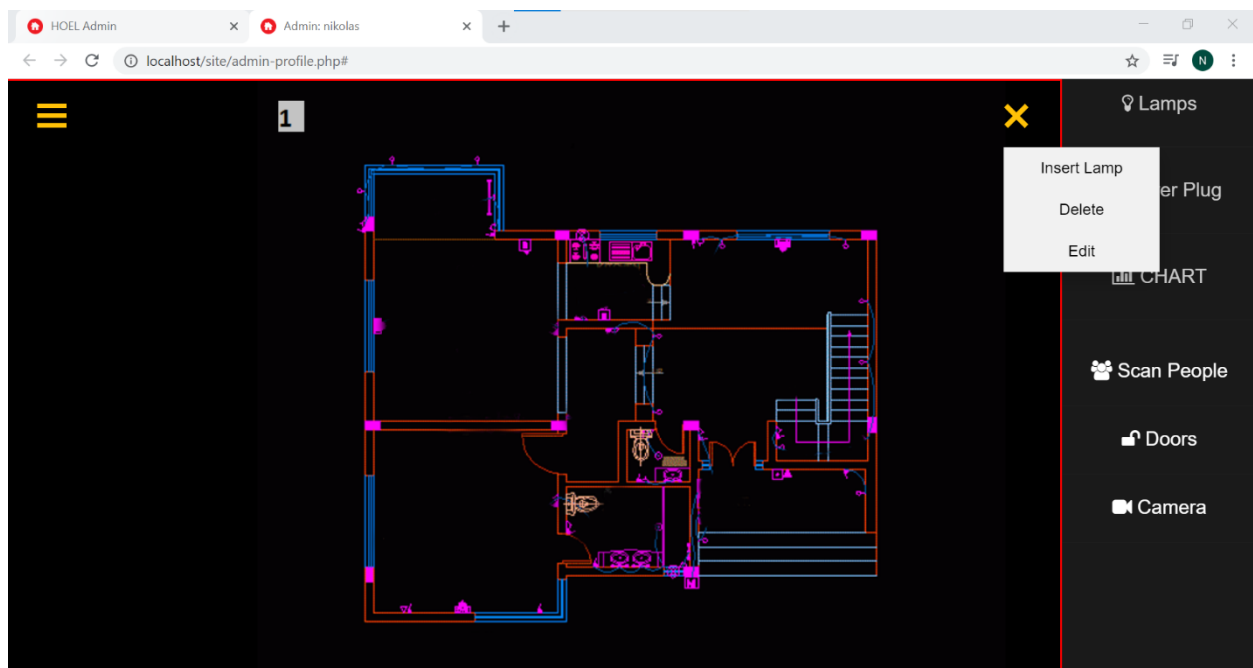
Admin/index.php

```
if(isset($_SESSION['login_user'])){
header("location: .. /admin-profile.php"); // Redirecting To Profile Page
}
?>
```

The above command, which is located in the admin/index.php file, takes us to the admin-profile.php file with the details of the respective user where is the environment with the additional features that we will see later what they are.

Additional features in each user's environment.

Additional features available to manage a user's environment are the addition of new luminaire elements, editing and deletion of present elements.



Explanation via Code

Show 'Lamps' Option and additional features

```
<li onmousedown =LampClick(event) class="dropbtn1"><i class="fa fa-lightbulb-o"
  aria-hidden="true"></i> Lamps</li>
<div class="dropdown1">
<div id="myDropdown0" class="dropdown1-content">
<a onclick="insertlamp()" href="#home">Insert Lamp</a>
<a onclick="delete1()" href="#about">Delete</a>
<a onclick="edit1()" href="#contact">Edit</a>
</div>
</div>
```

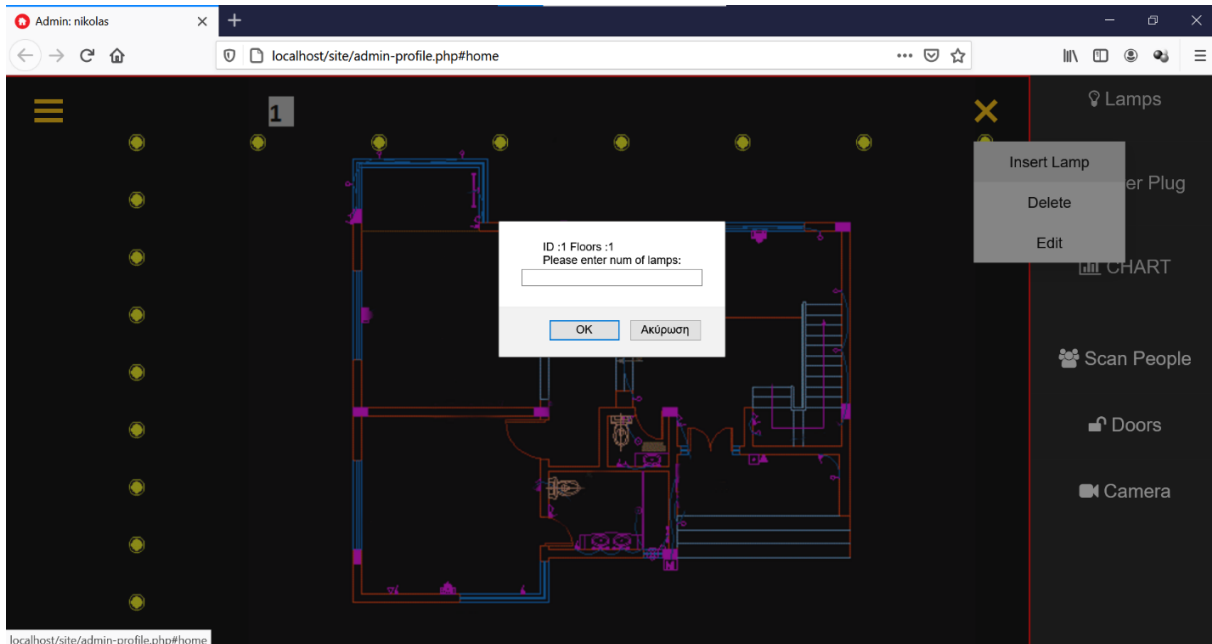
There is an onmousedown property with which we call the lampClick() function. Through this property, if we press left click, we give the value 1 so the function lampsL() is called and the value 3 if we press right click, therefore the function RightClicklamp() is called which displays the element div which has id = myDropdown0 as we see below.

```
function LampClick(event) {
  if (event.which == 1) {
    lampsL();
  }

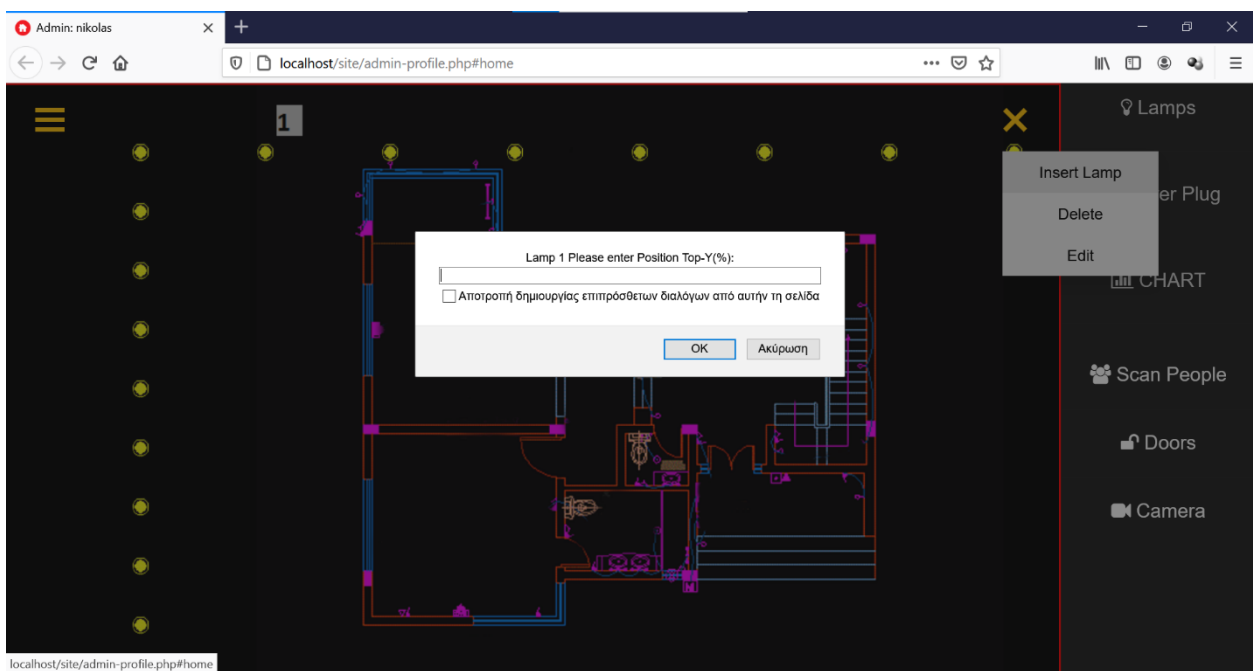
  if (event.which == 3) {
    RightClicklamp();
  }
}
function RightClicklamp() {
  document.getElementById("myDropdown0").classList.toggle("show");
}
```

Addition of new lighting elements

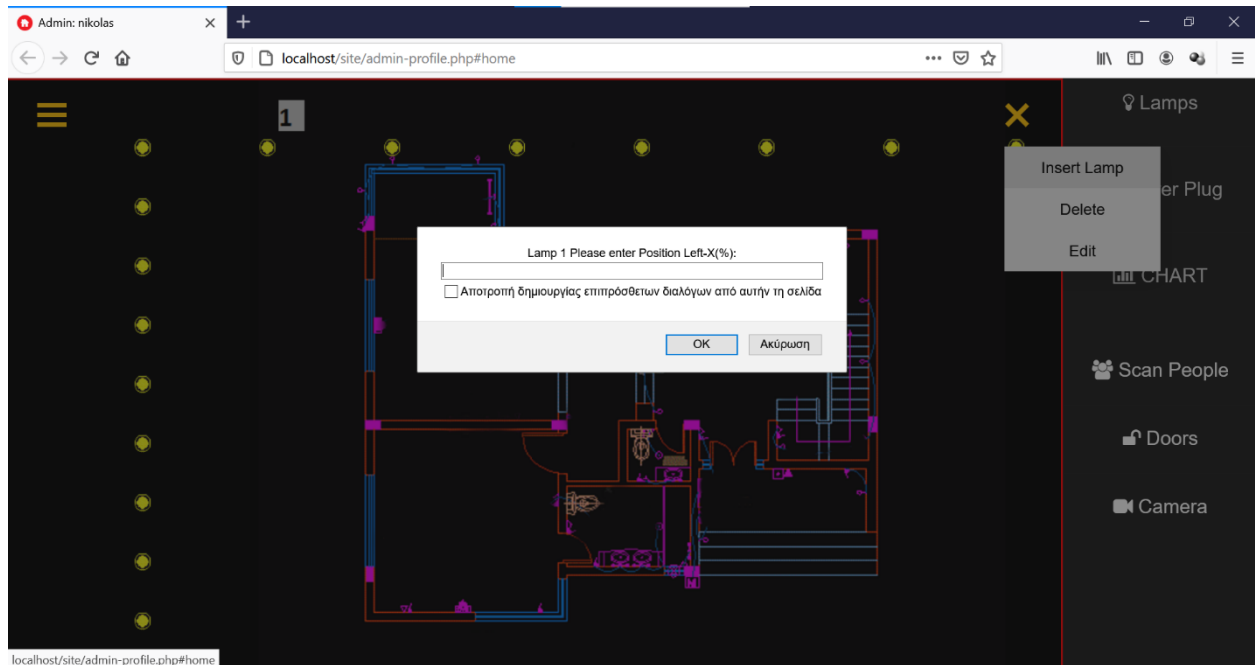
By clicking on the ability to add new elements, the administrator can place the luminaires of each user for each floor as they are respectively on the floor plan of his house. First, a data slot box appears in which we enter how many elements we want to add.



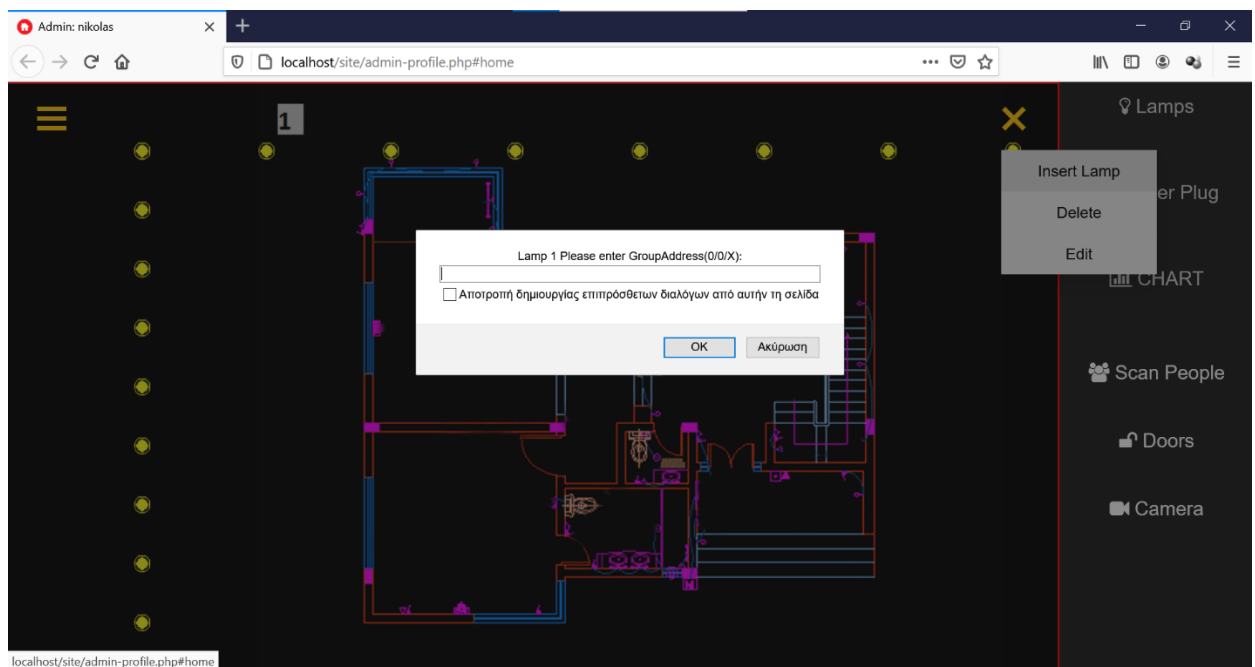
Next, a data input box appears in which we must enter the coordinates that the element should have along the Y axis (vertical axis) and is in percent (%) proportion to the display. We can determine this by looking at the auxiliary elements that appear to the left and above of the center panel and are arranged by 10% each.



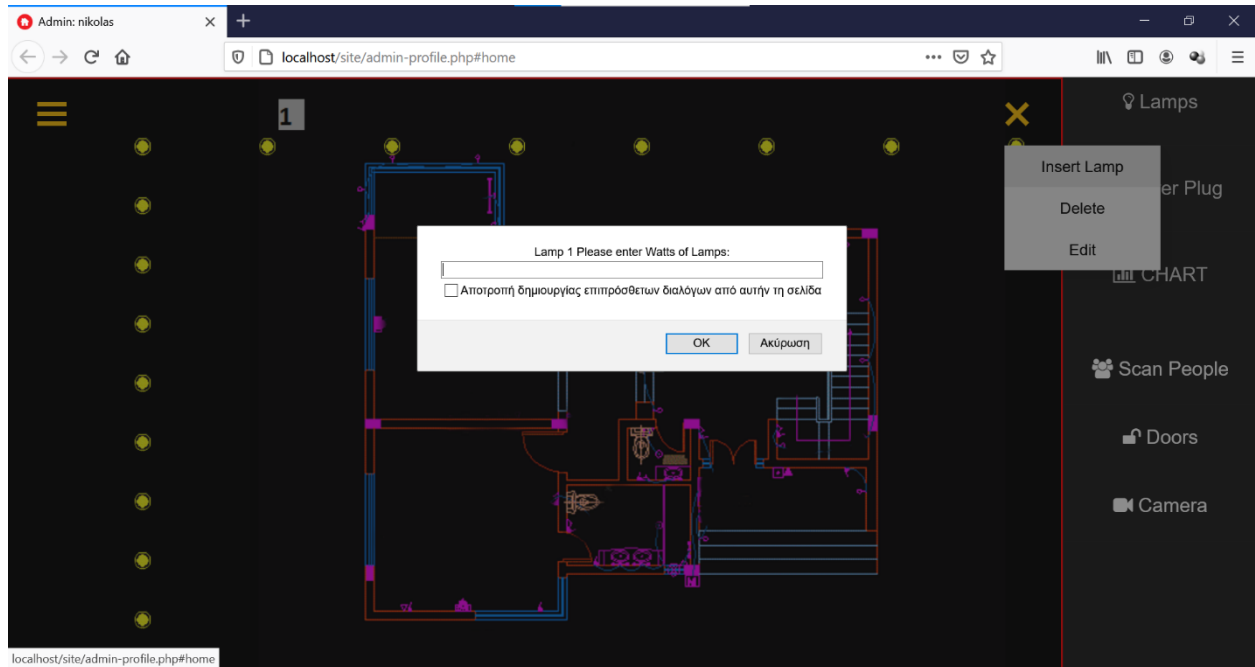
A corresponding input box is also displayed for the X-axis (horizontal).



Then an input box appears in which we must enter the address that each element should communicate and it is in the form x/y/z such as 0/0/1. This address is determined by us at each exit of our activator from the program called ETS as we will see below.



In addition, a last input box appears in which we register the consumption of each item. This will be very useful for subsequent calculations of consumption per day in a graph.



This process is done as many times as is the number of items we have put in the first input box for how many elements we will add.

Code Callout (adm-right-sidebar.php)

This code is identical to the code in the file, right-sidebar.php only we have added code for the additional features of the administrator.

To add elements we have created the following function:

insertlamp() (Αρχείο adm-right-sidebar.php)

```
function insertlamp() {
clear1();

id=<?php echo $login_id;?>;
was in;
was n;
var person0 = prompt("ID :<?php echo $login_id;?> Floors :"+idfloors+"\nPlease
enter num of lamps:", "");
if (person0 == null || person0 == "") {
txt = "User cancelled the prompt.";
} else {
n= Number(person0);;
}

for(i=1; i<n+1;i++){
var vary;
var person = prompt("Lamp "+i+" Please enter Position Top-Y(%):", "");
if (person == null || person == "") {
txt = "User cancelled the prompt.";
```

```

} else {
vary = person+"%";
}
var varx;
var person2 = prompt("Lamp "+i+" Please enter Position Left-X(%):", "");

if (person == null || person == "") {
txt = "User cancelled the prompt.";
} else {
varx = person2+"%";
}

        var varg;
var persong = prompt("Lamp "+i+" Please enter GroupAddress(0/0/X):", "");
if (persong == null || persong == "") {
txt = "User cancelled the prompt.";
} else {
varg = persong;;
}
was Watts;
var personw = prompt("Lamp "+i+" Please enter Watts of Lamps:", "");
if (personw == null || personw == "") {
txt = "User cancelled the prompt.";
} else {
watts = personw;
}

//alert("Group"+varg);
$.ajax({
url      : 'con-right.php',
method   : 'post',
data     : {idfloors: idfloors,id: id,i:i,varx:person2,vary:person,varg:varg,watts:watts},
success  : function(response){
//alert(response);
}
});
function createElementHtml(i,id,tagname){
var mylist0=document.getElementById("Lamps");
let nwtag = document.createElement(tagname);
nwtag.id = id;
nwtag.setAttribute("src", "images/lamp.png");
nwtag.setAttribute("width", "20");
nwtag.setAttribute("height", "20");
nwtag.style.position = "absolute";
nwtag.style.top=vary;
nwtag.style.left=varx;
mylist0.appendChild(nwtag);
return mylist0.innerHTML;
}
createElementHtml(i,"floor"+i,"IMG");
var temp = document.getElementById("floor"+i);

```

```
temp.onclick = (
function(t) {
return function (e) {
alert("You just clicked Lamp: " + t);
}})
(i);}}
```

First, within this function we call the clear() function which deletes all subelements present in the element with id=lamps. This is done to any function associated with the element that has id=lamps so that it is empty.

Clear()

```
function clear1() {
var list = document.getElementById("Lamps");
while (list.childNodes()) {
list.removeChild(list.firstChild);
}
}
```

Then, with the "prompt" method we set a variable and a window appears in which it has an input box. We take data from the administrator so that we can then process it. Initially, through this method we get from the administrator how many elements he wants to import. After typing the number and pressing the 'OK' button, we create an iterative method that will do the same steps as many times as the number of elements it wants to enter in the context of that user. Then with the "prompt" method we take variable data for each luminaire such as the location of the luminaire by x and y axis in percent, the communication direction of each element and the consumption. After getting the data for this element through the "Ajax" method we go to the con-right.php file where the following steps are taken:

con-right.php

```
<?php
$connection = mysqli_connect('localhost' , 'root' , '' , 'owners');
$idfloors = $_POST['idfloors'];
$id = $_POST['id'];
$varx = $_POST['varx'];
$vary = $_POST['vary'];
$varg = $_POST['varg'];
$i = $_POST['i'];
$watts = $_POST['watts'];
$lampon = 0;
echo $watt;
Elegxos ean iparxoyn idi kai alles lamps
```

```

$result1 = mysqli_query($connection , "SELECT * FROM `lamps` WHERE floor=".$id
floors." and userid=".$id." and lampnum=".$i."");
if($result1){
$result2 = mysqli_query($connection , "SELECT MAX(lampnum) as total FROM lamps
where userid=".$id." and floor=".$idfloors."");
$row = mysqli_fetch_assoc($result2);
$lo = $row['total'];
$new=$lo+1 ;
echo $new ;

$resultn = mysqli_query($connection , "INSERT INTO lamps(userid,floor,lampnum,X
,Y,GroupAddress,LampOn,Watts)VALUES('$id','$idfloors',$new,$varx,$vary,'$varg'
,'$lampon','$watts')");
$resultn1 = mysqli_query($connection , "INSERT INTO lampcon (datet,itemid,user
id,floorid,consu,timeuse)VALUES('2020-09-22',$new,'$id','$idfloors','0','00:00:
00')");
}
Allios pros8iki neon
else{
$result = mysqli_query($connection , "INSERT INTO lamps(userid,floor,lampnum,X
,Y,GroupAddress,LampOn,Watts)VALUES('$id','$idfloors',$i,$varx,$vary,'$varg','$
lampon','$watts')");
$resultn1 = mysqli_query($connection , "INSERT INTO lampcon (datet,itemid,user
id,floorid,consu,timeuse)VALUES('2020-09-22',$i,'$id','$idfloors','0','00:00:00
')");

if($result){
echo 'data updated';
}
}
?>

```

In this file we connect to the 'owners' database where through the PHP programming language and specifically the "Post" method we get the variables for the specific luminaire (element) and the user such as the user's id (id), the floor (idfloors), the element number (i), the position by x axis (varx) and y (vary), the communication address (varg) and the consumption (watts) from the previous adm-right-sidebar.php file and more specifically the insertlamps() function. The lampon variable is a variable that we set ourselves initially and determines, if it is "0" then our element is disabled, if it is "1" then it is enabled. We do a check inside the panel lamps if there is already a data entry for this user by searching for id and floor items. If there is, then we select the maximum number of items that this user has on the specific floor in order to register the new item as the last. We register our item inside the lamps panel and the lampcon panel for consumption use. If the same item is not already registered for the specific user and floor, we do normal item entry. Once the data for this element has been entered, we now create the element based on this data, using the createElementHtml() function.

```

nwtag.onclick= function(){
if (confirm("Are you sure delete Lamp?")) {
$.ajax({
url      : 'deletelamp.php',
method   : 'POST',
data : {id:obj.id,idfloors:obj.idfloors,lampnum:obj.lampnum},
success  : function(data2){alert("DELETE Lamp"+data2)}}
})
clear1();
delete1();
}
}

```

Ability to delete an item

The second feature, 'delete' has the same code as the lampsL() function, except that in the clickon part, i.e. what to do if we press the element, the delete() function is called which deletes the element that has the same number as the lampnum.

The data we need for this element is the id, floor and element number which we have from the previous piece of code. Therefore, through the "Ajax" method we transfer this data to the deletelamp.php file.

Deletelamp.php

```

?php
$id = $_POST['id'];
$idfloors = $_POST['idfloors'];
$lampnum = $_POST['lampnum'];

$connection = mysqli_connect('localhost' , 'root' , '' , 'owners');
$result2 = mysqli_query($connection , "DELETE from lamps where floor=" .
$idfloors . " and lampnum=" . $lampnum . " and userid=" . $id . "");
$result1 = mysqli_query($connection , "DELETE from lampcon where floori
d=" . $idfloors . " and itemid=" . $lampnum . " and userid=" . $id . "");

if($result2){
    //echo $idfloors;
    //echo $id;

    $result = mysqli_query($connection , "SELECT * from lamps where floor
=" . $idfloors . " and userid=" . $id . "");
    //$query = "SELECT count(*) as total from lamps where floor=" . $idfloors . " and userid=" . $id . "";
}

```

```

$query = "SELECT * from lamps where floor=".$idfloors." and userid=
".$id."";
$result = mysqli_query($connection, $query);
$xi =1;
while($row = mysqli_fetch_array($result)){
    echo $xi;
    echo $row["lampnum"];
    $lampn=$row["lampnum"];

    $sql=mysqli_query($connection ,"UPDATE lamps SET lampnum= '$xi'
where floor=".$idfloors." and lampnum=".$lampn." and userid=".$id."");
    $xi =$xi +1;
}
}

mysqli_close($connection);
?>

```

In the deletelamp.php file, the tables of our database are checked lamps and lampscon to find the elements that have corresponding data. We delete them with the MySQL command 'Delete from table where colum=?'. Then, due to the peculiarity of the code we have, we number our data again in order to have correct numbering after an element has been deleted.

Edit an item

The function code for this feature is similar to the previous two functions, except that the clickon part calls edit() which is similar to insertlamps() but with some changes.

```

nwtag.onclick= function(){
var person = prompt("Lamp "+i+" Please enter Position Top-Y(%):", "");
if (person == null || person == "") {
txt = "User cancelled the prompt.";
} else {vary = person;}
var varx;
var person2 = prompt("Lamp "+i+" Please enter Position Left-X(%):", "");

if (person == null || person == "") {
txt = "User cancelled the prompt.";
} else {varx = person2;}

var vary;
var persong = prompt("Lamp "+i+" Please enter GroupAddress(0/0/X):", "");
if (persong == null || persong == "") {
txt = "User cancelled the prompt.";
} else {varg = persong;}
//alert("X"+varx+" Y"+vary);
$.ajax({
url      : 'editlamp.php',
method   : 'POST',
data     : {id:obj.id,idfloors:obj.idfloors,lampnum:obj.lampnum,x:varx,y:vary,grou
p:varg},
success  : function(data2){alert("Lamp"+data2)}
}edit1();}

```

By clicking on the specific item, entry boxes appear through the "prompt" method, in which the administrator enters data for the new change of the element, such as a new position by X, Y axis and a new contact address. When the administrator enters the new data through the "Ajax" method, we transfer this data to the file editlamp.php

Editlamp.php

```

<?php
    echo $id = $_POST['id'];
    echo $idfloors = $_POST['idfloors'];
    echo $lampnum = $_POST['lampnum'];
    echo $group = $_POST['group'];
    echo $x= $_POST['x'];

```

```
echo $y= $_POST['y'];

$connection = mysqli_connect('localhost' , 'root' , '' , 'owners');

$result2 = mysqli_query($connection , "UPDATE lamps SET x= ".$x.",y= ".$y.",GroupAddress= '$group' where floor=".$idfloors." and lampnum=".$lampnum." and userid=".$id."");
if($result2){
    echo "update";
}
?>
```

After transferring the data to the specific file, we place it in variables, connect to our database and through the syntax 'UPDATE TABLE SET' we change the data of our element and call again the function edit1() so that the environment of the administrator is ready if he wants to make another change to another luminaire of the user.

These are the main parts that make up the creation of our online platform. We had created the platform in such a way that our platform could be easily connected to node.js platform, as we will see in more detail in the third part. Next, in the second part we will look at the construction part of the work which is a more electrical and practical part.

Part B (Constructional-KNX)

The second part concerns the construction part, market research, wiring of the devices in an electrical panel and the addition of addresses via ETS5 (program provided by KNX) to each device so that we can communicate to them.

Introduction to KNX

KNX Association was founded in 1990 in Brussels where its original name was EIB Association. The main purpose of the company was to promote smart homes and buildings using the EIB (European Installation Bus) system, which was created jointly with other manufacturers.

In 1999 the company merged with two other European companies, BCI and European Home Systems, and was renamed KNX Association

What is KNX?

Modern man is now looking to have more and more control of his home directly and to have the ability through many different accesses. He asks to enjoy new technologies more and more in his daily life, in order to offer him hours of relaxation, carefree and happiness. This is what the KNX standard offers him. Its main goal is to emphasize the daily comfort and pleasure of a modern man. Large companies such as abb, siemens, etc., have supported and designed many devices that promote in the market based on this standard.

Why KNX?

A model that came to stay in the market since the advantages it has and the support it receives is great.

Advantages:

- It is a global standard
- Offers Comfort, Safety, Energy Saving
- With KNX certification it ensures that different devices from different manufacturers can have direct and reliable communication with each other
- Guaranteed interoperability through neutral certification
- High product quality, since ISO 9001 is mandatory for all manufacturers of KNX devices.
- It has a tool, ETS, which regardless of manufacturer can group and diagnose all devices
- A large number of features such as lighting control, power control, climate control, fire detection, heating, shutters, audio/video and many more.
- Easy to expand new needs
- Various communication media: twisted pair (TP), power line (PL), Ethernet, Radio frequencies (RF)
- Interconnection with other systems Dali, BACnet
- There is great support since more than 300 KNX manufacturers in 35 countries offer 7000 certified devices

Media

Medium	Broadcast using	Confusing Cases
TP(Twisted Pair)	Separate network	New facilities and renovation extensions

PL(Power Line)	Existing Electricity Network	Always in places where an additional control conductor circuit cannot or should not be installed while a 230v electrical network is available.
IP (Internet Protocol)	Ethernet	Always in places where it is not possible or desirable to lay KNX network pipelines
RF(Radio Frequency)	Radio signals	In large installations where a fast transmission at the backbone level is required.

Used devices

The devices we needed for this task are the following:

- Switch Actuator
- Power Simply (Supply of current-voltage to devices)
- IP Router (Internet Communication)
- Residual Current Circuit Breaker (Διακόπτης Ασφάλειας)
- Electrical Panel
- Power Cord (2.5mm)
- Bus cable of the company Knx

We had at our disposal (in the laboratory) the Power Simply of the company ABB, model SV/S 30.640.5.1 where it covered the needs of Power Simply that we needed. In addition, we had the IPS/S 2.1 IP Router from ABB where it also supported the specifications we needed.

So we needed additionally:

- Switch Actuator
- Residual Current Circuit Breaker (Διακόπτης Ασφάλειας)
- Electrical Panel

- Power Cord (2.5mm)
- Bus cable of the company Knx

Research for the market of the above devices.

For the right and appropriate selection of the following devices, market research was conducted to meet our needs and goals. For this reason, research was carried out in various shops in the local area based on the relationship between price, availability, quality of construction (manufacturer) and capabilities.

*We had a restriction that all products had to be purchased from one store.

We chose to buy Switch Actuator SA/S 8.16.6.1 from ABB. The choice of this device was made due to the fact that the devices we already owned are from the company ABB and this company has quite good support in our country. In addition, we got this model because it has power sensors in each output port, so in future projects or even in the subsequent development of this project such technology will be very important.

Switching actuator 8-fold for KNX home automation SA/S 8.16.6.1



ABB
SA/S 8.16.6.1
2CDG110114R0011
4016779708326 EAN/GTIN

OUR BEST PRICE
€311.05 incl. VAT
plus shipping

3-10 days* (CYP)

Ekinex EK-FB1-TP KNX 8-fach Binärausgang



Bildquelle: Ekinex

Unser Preis :
311,23 €
inkl. 18% USt., zzgl. Versand (ab 5,90 Euro)

Menge: 1 + - Stk

In den Warenkorb

Lieferfähig: in 10-14 Tagen

EAN:
ArtikelNr.: AB1991015
HAN: EK-FB1-TP
Typ-Bezeichnung:



MTN6700-0008 - KNX Switch
Actuator Basic REG-K/8x/16 A with
manual mode, Schneider Electric

Brand: Schneider Electric

Product Code: MTN670

Availability:

To be confirmed

Ex Tax:
€ 380.31

In addition, we had purchased the following Switchboard:



Πίνακας διανομής Χωνευτός 3Σ. 36Στ. IP40 :hager

Τιμή eshop (με ΦΠΑ): **26,80 €** Διαθέσιμο για αποστολή

Διατίθεται (ανά: 1 ΤΕΜ)

- 1 + Προσθήκη στο καλάθι

Όροι επιστροφής | Δωρεάν μεταφορικά & αντικαταβολή άνω των 25€

ΠΑΡΑΔΟΣΗ ΣΤΟ ΧΩΡΟ ΣΑΣ

Για τα διαθέσιμα υλικά, παραδίδουμε με μέσα της ΚΑΥΚΑΣ:

- σε Αθήνα & Θεσσαλονίκη σε 1-4 ημέρες
- στην υπόλοιπη Ελλάδα σε 2-15 ημέρες*

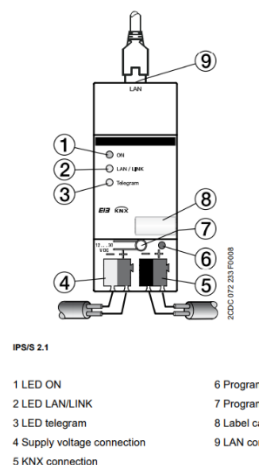
ΚΡΑΤΗΣΗ ΣΕ ΚΑΤΑΣΤΗΜΑ

[Έλεγχος διαθεσιμότητας ανά κατάστημα](#)

About Devices

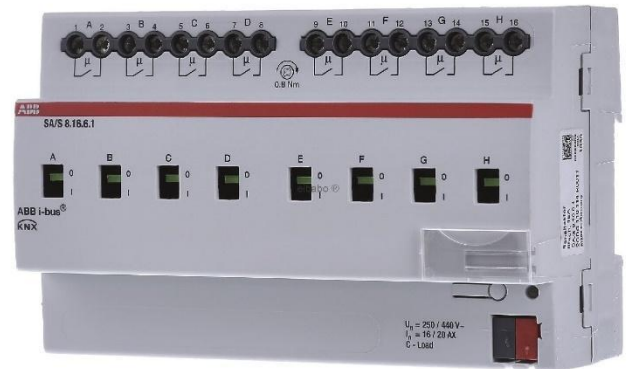
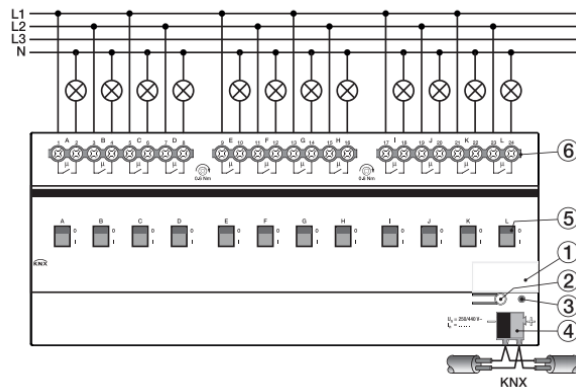
● IP_Router - ABB-IPS/S 2.1

It is responsible for the interconnection of the instabus network with the ethernet .
It enables our device to communicate with devices that are connected to the internet. In addition, through the Ip Router we can program our devices. In our case it is necessary for our work since we want to give commands from our platform to our device via internet. In addition, it will be necessary for us to program our devices through the program provided by KNX (ETS5).



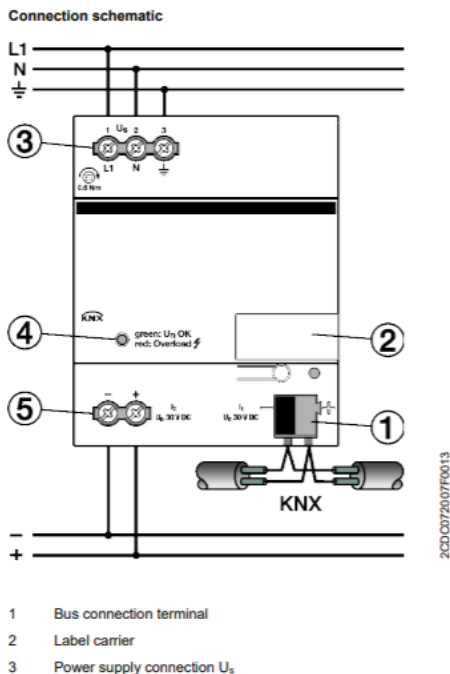
- **Switch - SA/S 8.16.6.1**

We had chosen this switch Actuator, due to the fact that it has eight ports for connecting appliances or luminaires, which is a satisfactory number for connecting devices in a room. In addition, it has power flow controllers in each output port, where in a subsequent upgrade it will be a very capable tool so that we can measure the current consumed by each device that will be connected.



- **Power Simply - SV/S 30.640.5.1**

A Power Simply is a necessary accessory, since it is responsible for providing our devices with a voltage of 30 volts so that commands can be transferred through the bus line.



- **Residual Current Circuit Breaker (ABB FH202 AC-25/0,03)**

It is also a device that is necessary in a wiring of electrical appliances since it is responsible for human safety in any short-circuit problems. Ensures protection to people and facilities from damage current (short circuit).



Additional Elements of the KNX system

- **Channel cleavage**

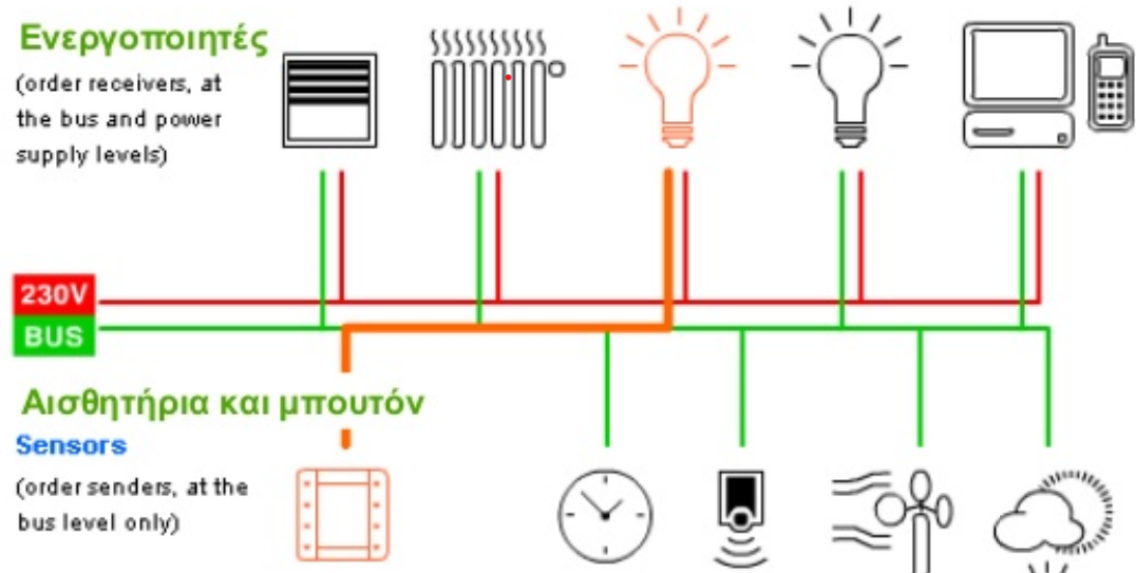
Bus plugs are found in all devices that are certified according to KNX standard. They have a combination of colors red, which determines the positive (+) polarity, and black, which determines the negative polarity (-). In addition, each cable connects up to four 0.8m cables. They are responsible for branching the bus cable, extending it, protecting its ends and connecting the bus cable to bus devices.



- **Communication Cable**

A separate KNX TP (twisted Pair) bus cable, which is certified by KNX, is used for the communication of this construction. For the proper operation and transfer of information we need a power supply that supplies subscribers with an operating voltage of 30 Volts in parallel. Telegrams are formed on continuous voltage, and with logical zero a pulse is transmitted, and with a lack of pulse it is a logical one.

The connection format of an electrical installation using a KNX standard is the same format as the current installation, i.e. 230 volts, with the only difference being the addition of a bus line which provides contact information to the devices.



The recommended conditions to be observed for the proper operation of an electrical installation using a KNX cable are:

- o The distance that the power conductor must be with the communication bus must be at least 4mm.
- o The maximum distance of a line (length) must be a maximum of 1000m cable.
- o Two subscribers who communicate with each other must not be more than 700m apart.
- o The most remote subscriber (device) should not be more than 350m of branches from the power supply.

Other methods of communication

The KNX system is a system that multiple devices of the same standard can send, receive and manage serial format data so that they can perform various actions. There are input devices such as switches, light, temperature, humidity, wind sensors and even touch screens and output sensors such as switches, relays and lighting regulators. In addition, there are devices which are necessary for the operation of a KNX system such as Power Simply and IpRouter. All devices must be archived with an individual address by the ETS program and then programmed to determine what actions each input device will perform and with which devices it will communicate with output devices. The system is decentralized since it operates without the use of an electric computer.

Contact Procedure

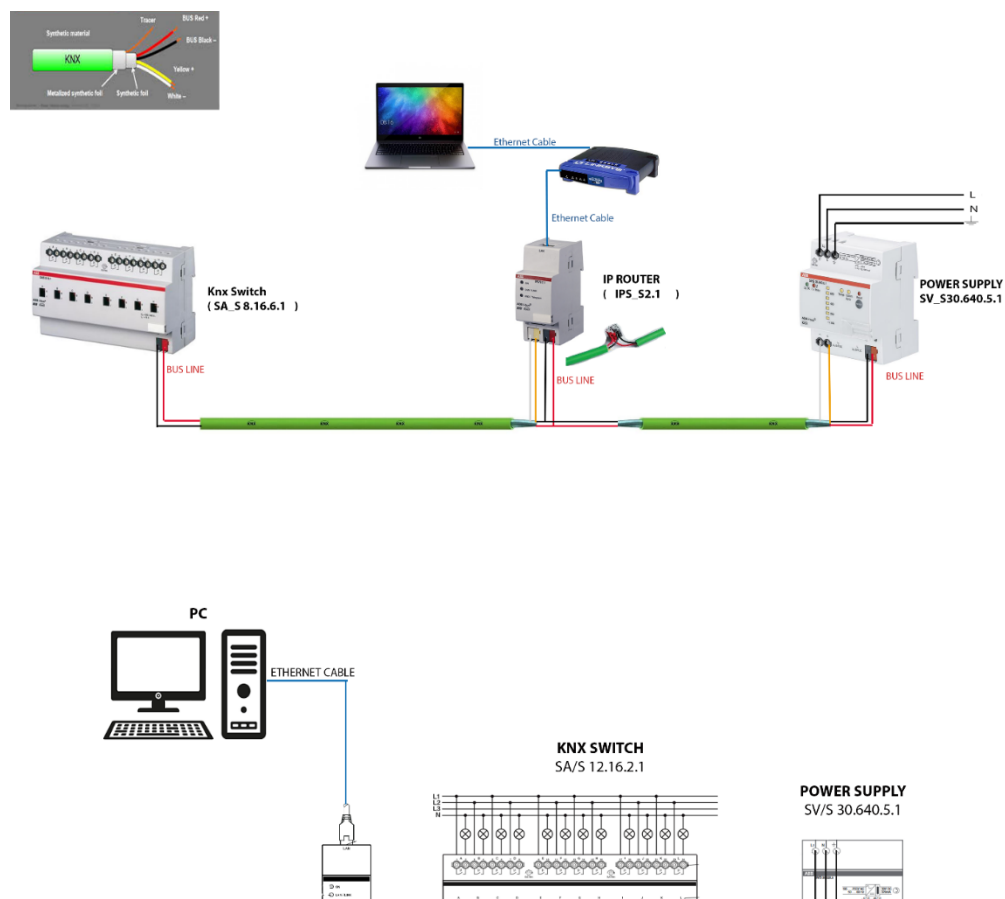
Communication of transmission and exchange of information between subscribers takes place through a communication channel in serial transmission and under the condition of a protocol (set of rules). The transmission to be sent is encoded and configured in telegram form by a sender device (sensor-input device) and sent via the communication channel to a receiving device (output device). If the recipient device has the same individual address as that contained in the telegram, it accepts the packet and informs the sender of receipt of the telegram. If there is no confirmation, then the sender repeats sending the telegram up to three times.

Wiring execution procedure

In order to perform the wiring of our construction, we had made an initial connection of the devices before placing them inside the electrical panel in order to see that the connection has been made correctly and that everything is working properly as shown in the image below.

Create a Draft so that we have a picture of the wiring of the devices

6



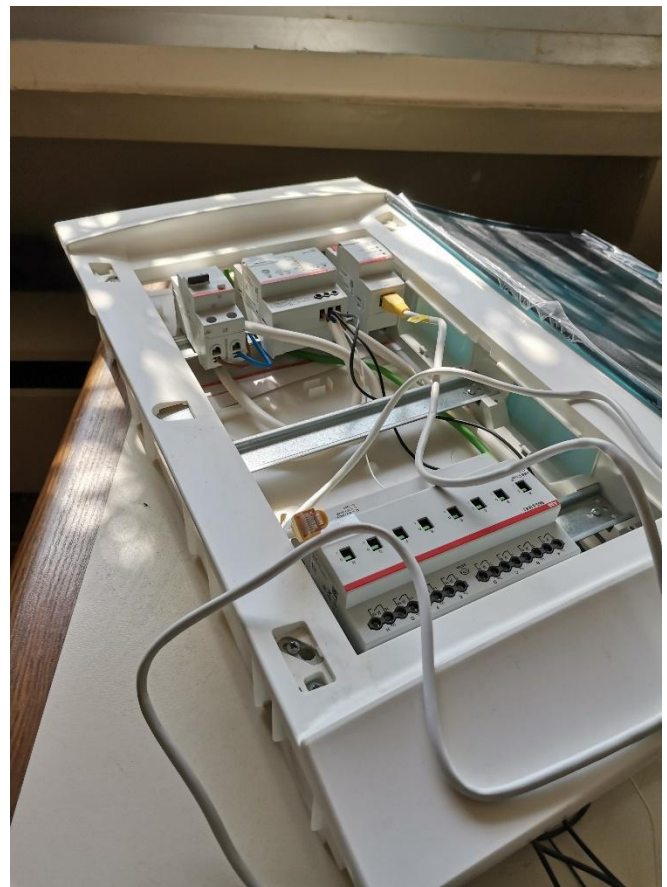
Steps:

1) Start cutting and stripping the green wire (communication channel) on both sides using a cable stripper so that we can place them on the plugs of each device.



2) We place the cable inside the plugs and then each clamp in its device.

Once all our connections were correct, we reconnect our devices, placing them with the same wiring inside an electrical panel



In order for KNX devices to be able to perform any operation, they must be programmed by the ETS program, which we specify which functions each device must perform. ETS is a program provided by the company and there are 4th versions at this stage:

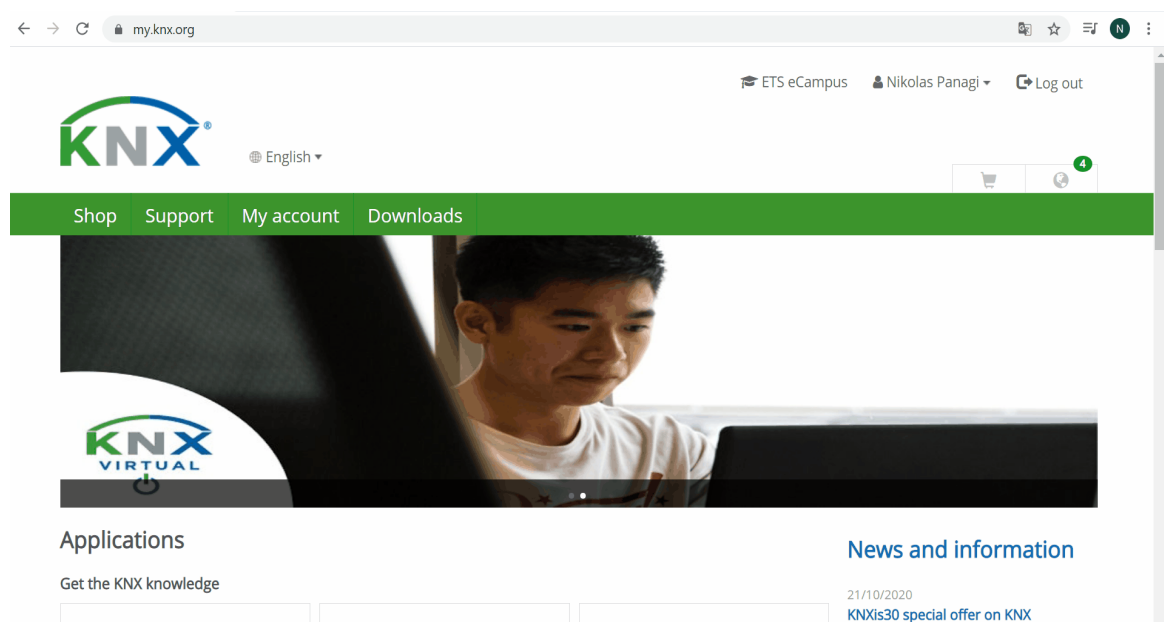
Edition	Restrictions	Upload Information
ETS5 Demo	A maximum of three (3) KNX devices are allowed per project.	This version is free and is aimed at newcomers, i.e. those who want to learn ETS5 without having to pay.
ETS5 Lite	A maximum of twenty (20) KNX devices are allowed per project.	This version is aimed at those who want and wish to create or manage small KNX projects
ETS5 Professional	Full Functionality	This version is aimed at those who want and wish to create or manage all possible sizes of KNX projects. The Professional version has no limitation in terms of its functionality.
ETS5 Supplementary	Full Functionality	This version requires at least an ETS5 Professional license and allows working with ETS on 2 computers simultaneously.

Device programming via ETS5

Steps

Download ETS5

1) First we go to the website of the KNX Association (<https://my.knx.org/>)



Create an account to be a member of Knx Association

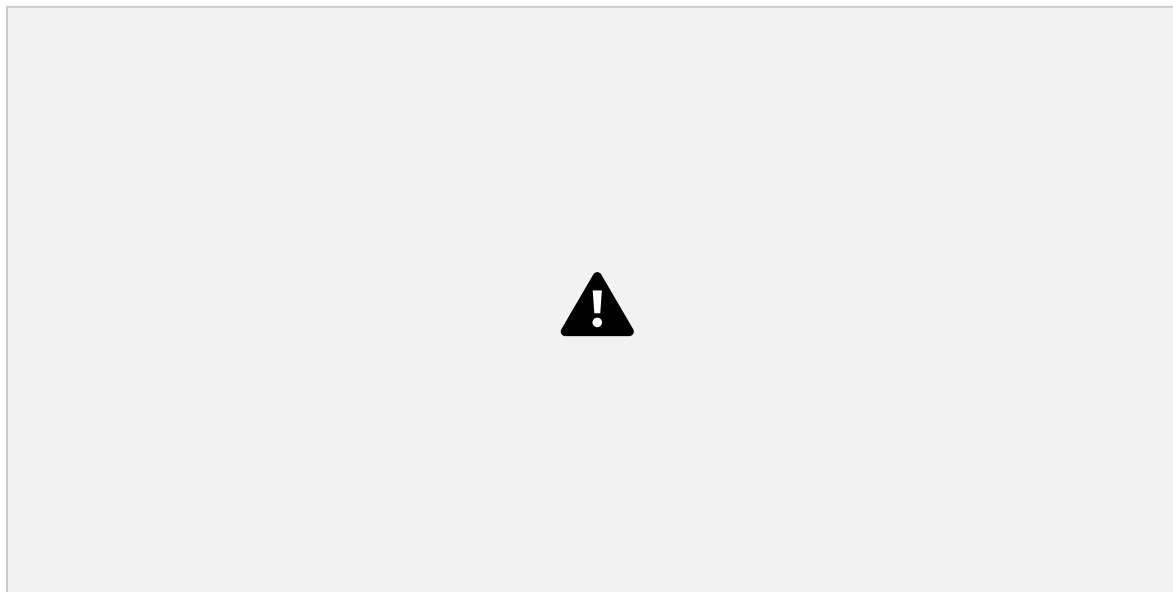
2)

3) Then we go to the section "Downloads->Software" and find ETS5 demo and press "Download".

Install ETS5

1) We accept the terms of use

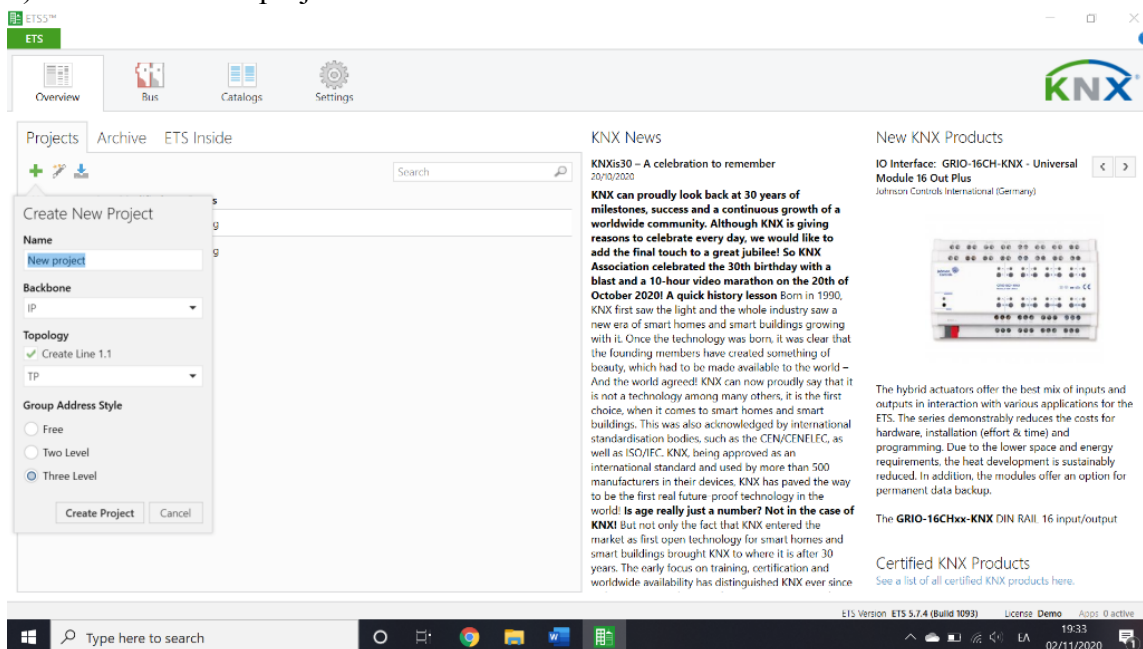
2) Then we press the "Install" button and the program is installed.



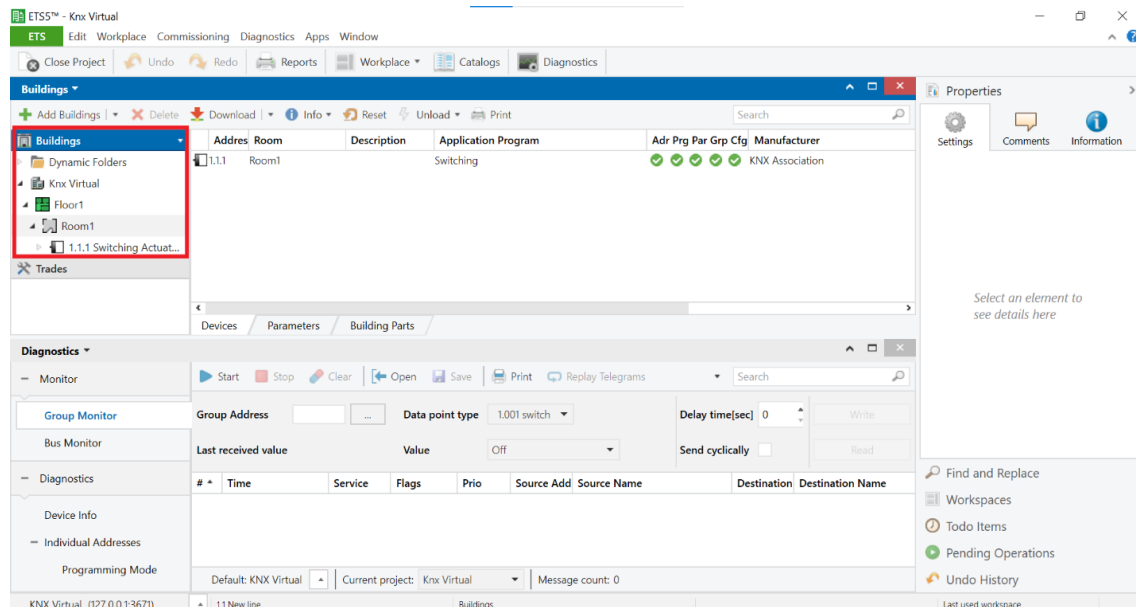
ETS5 programme

1) Once we have installed the ETS program we run it.

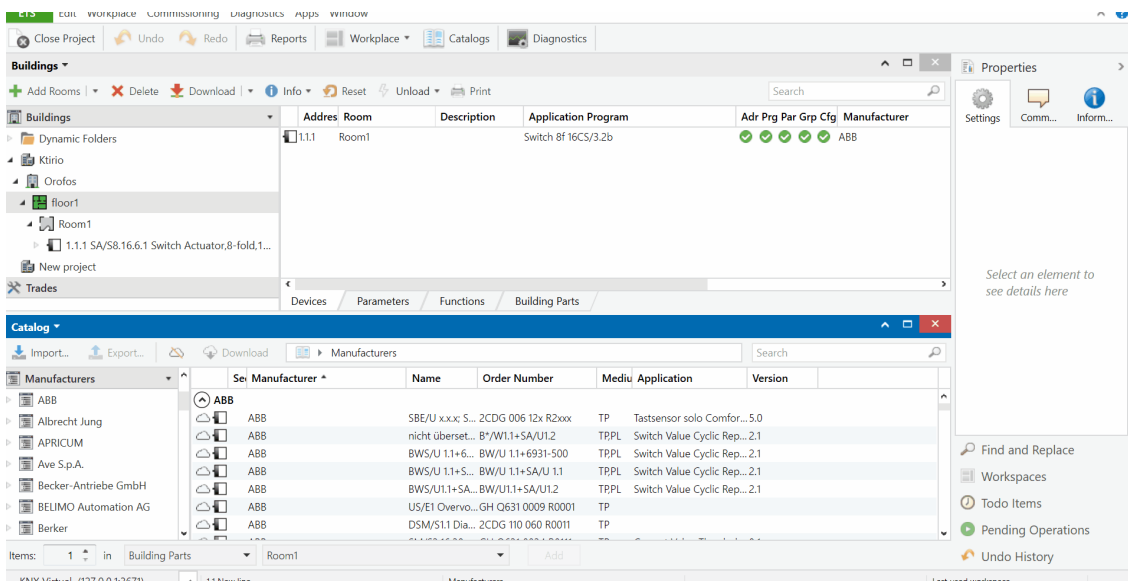
2) We create a new project with a name of our choice.



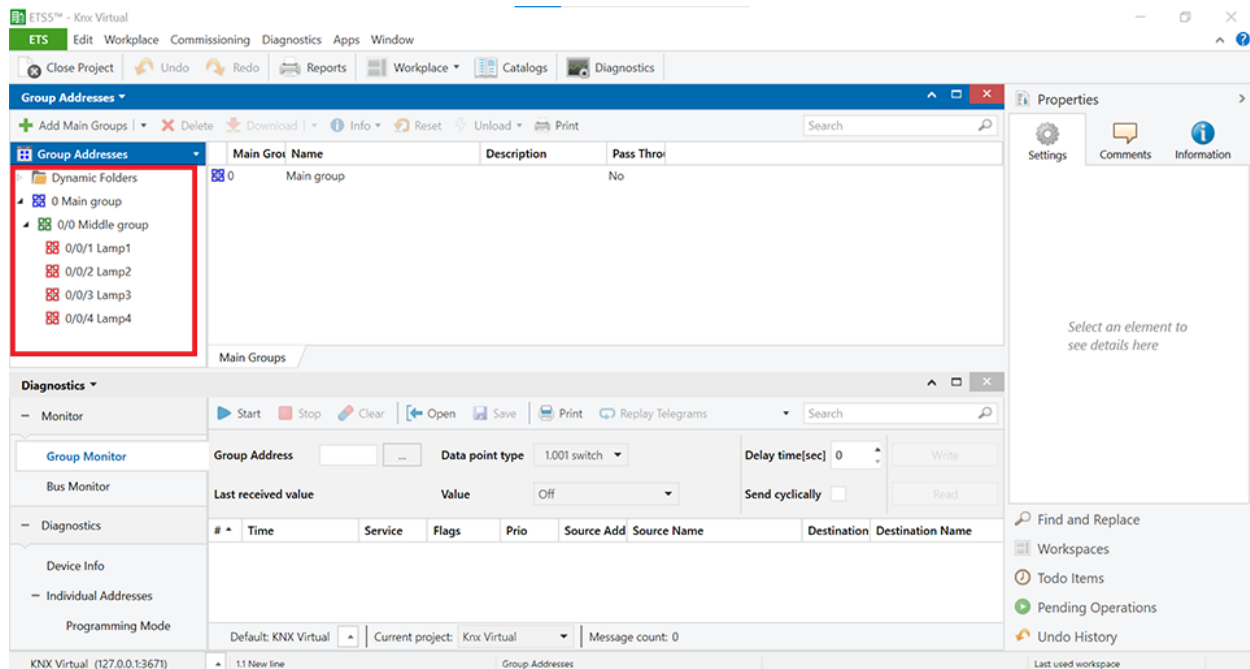
3) Then we create the levels of the project we have so that in a large project we can know where each device is.



4) Go to the "Workplace->Catalogs" options to display all KNX certified devices and drag with our mouse to the room we want the devices we own.



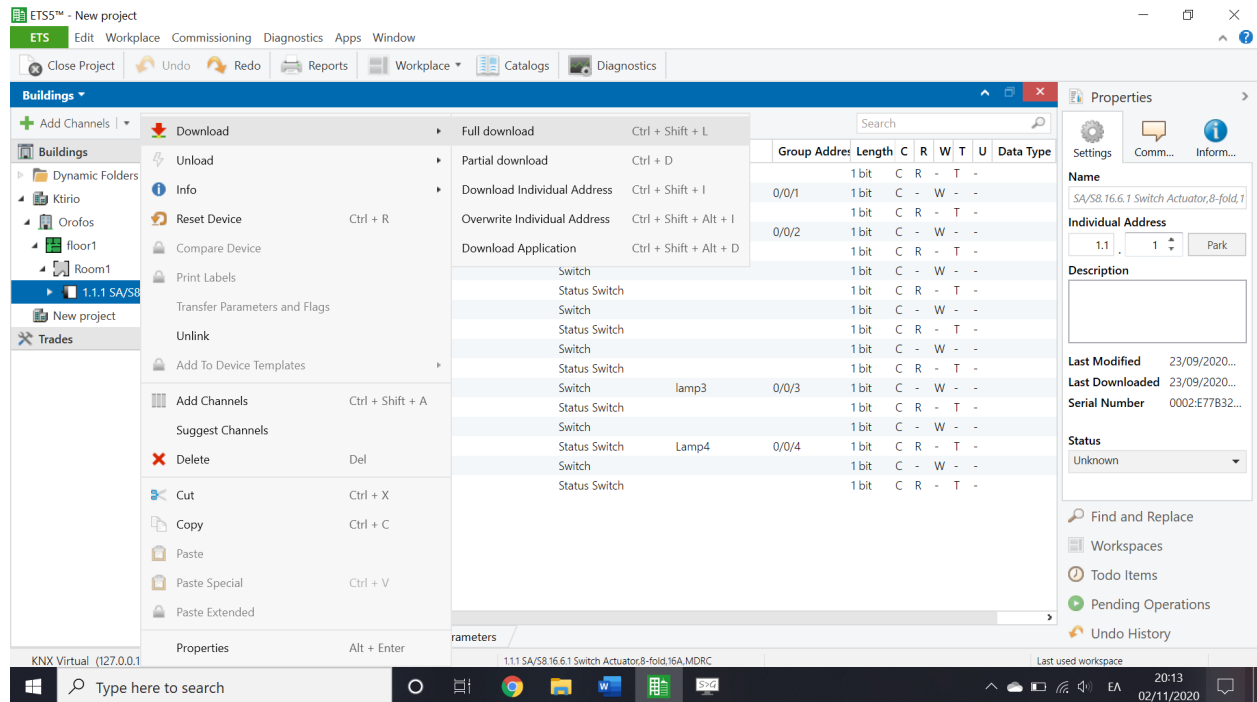
5) Then we create the groups that our devices will contain. Groups determine the interaction of devices.



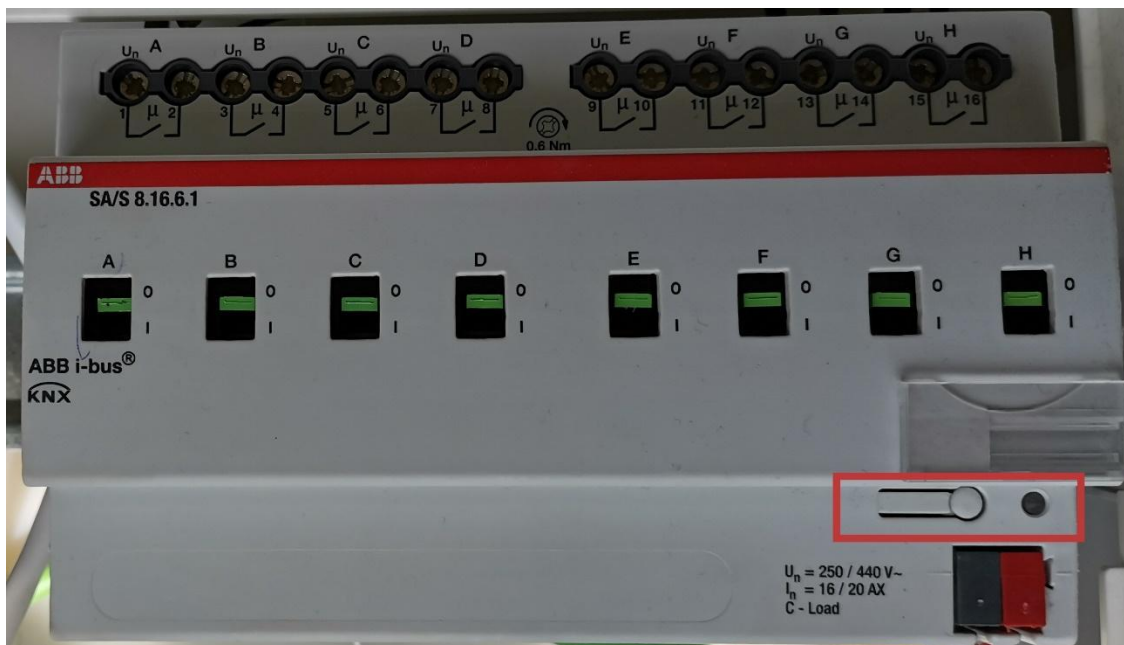
For example, if we want a switch to communicate with an actuator, then we must slide the command of our switch together with the port of our actuator that we want to communicate in the same group.

Once we have set the right specifications for our devices within ETS, it's time to load the information onto our devices.

In order to be able to load the information on our device we must press on the tab that has the data of our device right click "Download->Full download".



To load the data we want onto the correct device, the program tells us that we need to press a button that is placed on our device to determine which device it is.



ETS™ - New project

ETS Edit Workplace Commissioning Diagnostics Apps Window

Close Project Undo Redo Reports Workplace Catalogs Diagnostics

Buildings

Add Channels Delete Download Info Reset Unload Print Search

Buildings	Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
Dynamic Fo...	69	Output C	Status Switch			1 bit	C	R	-	T	-		Low
	70	Output D	Switch			1 bit	C	-	W	-	-		Low
Ktirio	89	Output D	Status Switch			1 bit	C	R	-	T	-		Low
Orofos	90	Output E	Switch			1 bit	C	-	W	-	-		Low
floor1	109	Output E	Status Switch			1 bit	C	R	-	T	-		Low
Room1	110	Output F	Switch	lamp3	0/0/3	1 bit	C	-	W	-	-		Low
1.1.1...	129	Output F	Status Switch			1 bit	C	R	-	T	-		Low
0: G...	130	Output G	Switch			1 bit	C	-	W	-	-		Low
10:...	149	Output G	Status Switch	Lamp4	0/0/4	1 bit	C	R	-	T	-		Low

Group Objects Parameters

Group Addresses

Add Group Addresses Delete Download Info Reset Unload Print Search

Group Addresses	Address	Name	Description	Centre	Pass T	Data Type	Length	No. of	Last Value
Dynamic Folders	0/0/1	Lamp1		No	No	1 bit	1		
0 Group1	0/0/2	Lamp2		No	No	1 bit	1		
0/0 MGroup	0/0/3	lamp3		No	No	1 bit	1		
	0/0/4	Lamp4		No	No	1 bit	1		

Group Addresses

IP Interface (192.168.2.6:3671) 1.1 New line 0/0 MGroup Last used workspace

Windows Taskbar: Type here to search, 22:40 23/09/2020

Properties Panel:

- Find and Replace
- Workspaces
- Todo Items
- Pending Operations
 - Active: 1.1.1 SA/S8.16.6.1 Switch Act...
 - History
 - Cancel all
 - Download(All): Downl...
- Undo History

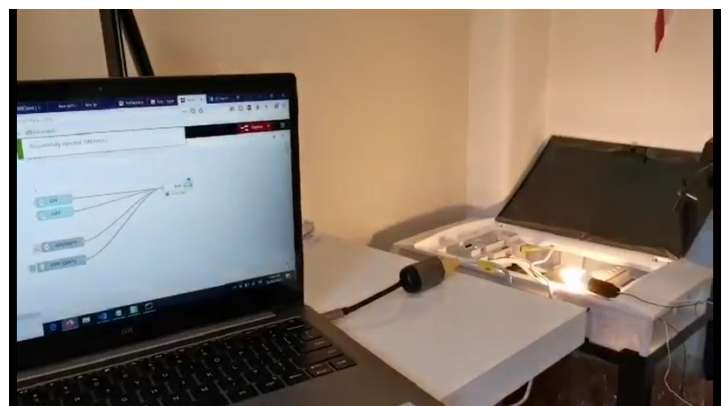
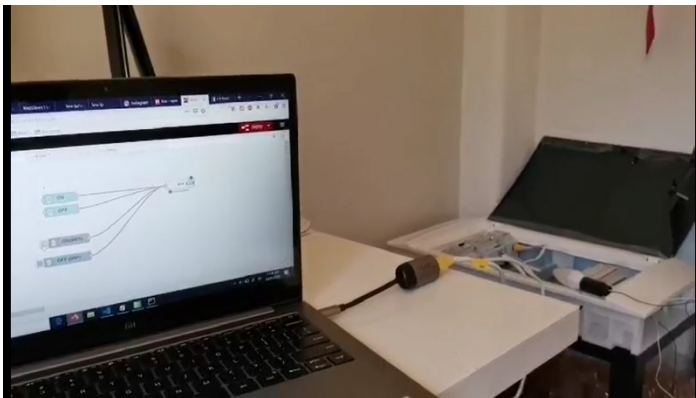
Part C (Communication of platform with devices)

After we had created our web platform, through the HTML programming language and connected the device where we will communicate, we had to find the right communication node to exchange information correctly and directly. This part is the foundation for the construction of such a platform since without an intermediary node there will be no exchange of information. In addition, the communication node plays a very important role in the communication time of the platform and the device but also in how many connections there can be on a device (ie how many users can connect from the platform).

Initially, an attempt was made to communicate through a node.js method called **node red knx ultimate**. Through this method we contacted our device after we had placed the contact address of our device (0/0/1) that we had previously set through the ETS program presented in the 2nd part.

More information about node **red knx ultimate**:

<https://flows.nodered.org/node/node-red-contrib-knx-ultimate>



Searching more on the internet and more specifically on www.npmjs.com, www.bitbucket.org and www.github.com websites, testing several libraries we found on the github website a library the knx.js of node.js platform. A library where it gave us the opportunity with some commands to create a communication channel and then send a signal to the specific device (address), to check its functionality. Therefore, we started working on communication based on this platform and this library to create the appropriate code-node communication between our own platform and our construction.

Introduction to Node.js

Node.js is a platform built on the JavaScript programming language. It was created by Ryan Dahl in 2009 and its main goal is easy communication between online websites and back-end API services in real-time.



knx.js Library

It is a library that has been based on the knx.net (<https://github.com/lifeemotions/knx.net>) and has contributed to its construction by:

- Alexander Borovsky
- Ykornilov
- Peter Kessen
- Daniel van Hoesel

More information can be found at the following link.

<https://github.com/estbeetoo/knx.js/>.

In order to be able to use the above library we had to use and install the node.js program on our computer.

Λήψη: <https://nodejs.org/en/download/>

Steps to install, at the following link :

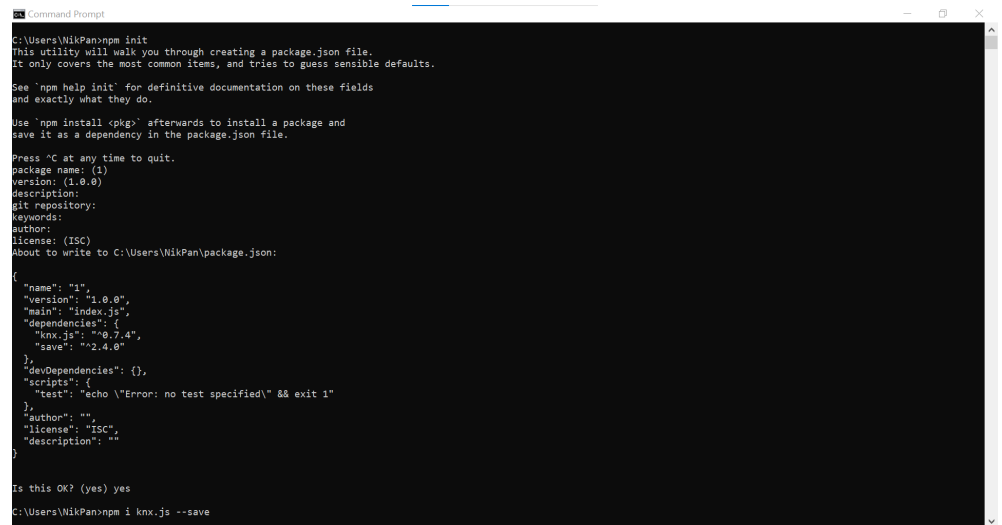
<https://phoenixnap.com/kb/install-node-js-npm-on-windows>

Programming through Node.js

Once we have installed the program (node.js) we execute the following command in the console of windows cmd to install knx.js library, which will provide us with the commands with which we will be able to communicate on our devices.

- **Npm init**

- **npm i knx.js --save**



```

C:\Users\NikPan>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

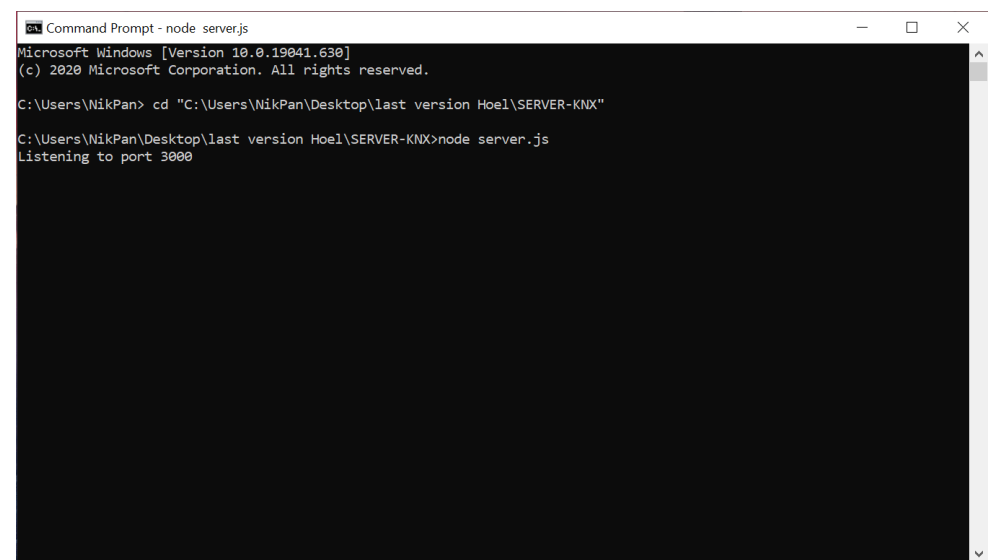
Press ^C at any time to quit.
package name: (i)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\NikPan\package.json:
{
  "name": "i",
  "version": "1.0.0",
  "main": "index.js",
  "dependencies": {
    "knx.js": "^0.7.4",
    "save": "^2.4.0"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}

Is this OK? (yes) yes
C:\Users\NikPan>npm i knx.js --save

```

Then we select the path (Path) that we have inside the file server.js, ie the program we have created and we execute the cd command and paste the link and press enter. Therefore, we will be in the folder where we have our file placed. We write the node command and then the name of our file.

- **node server.js**



```

Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\NikPan> cd "C:\Users\NikPan\Desktop\last version Hoel\SERVER-KNX"

C:\Users\NikPan\Desktop\last version Hoel\SERVER-KNX>node server.js
Listening to port 3000

```

Code Explained

In node.js we had created a server that can run on our local computer or we can load it on a microcomputer such as raspberry pi on the port:3000 network port to be able to send commands and receive data from our devices as we will see in the code below. In addition, this file is called the app.js file which in turn calls the routers/index.js file where the main functions performed by our server are located.

Server.js

```
const app = require('./app');

let port = process.env.PORT || '3000';

const server = app.listen(port, () => {
  console.log('Listening to port ' + port);
});
```

Initially in the code of the file Routes/index.js we have set the variables express, router which create communication with the app.js file so that the following commands-processes run on our server. We set the knxconnectiontunneling variable with which we call a property from our knx.js library, in order to create a communication channel between the devices, which as we will see and then set the ip address and ports of each device so that our server knows which two devices will be connected to the communication.

```
var express = require('express');
var router = express.Router();
const KnxConnectionTunneling = require('knx.js').KnxConnectionTunneling;
let connection = new KnxConnectionTunneling(
  '192.168.2.6',//knx
  3671,
  '192.168.2.2',//PC
  13671
);
```

Then we had created in the router object, ie on our server, a data entry with a "get" method. That is, if from our web platform we call the ip address/knx/light/:sw(sw=on or off) of our server where in our case it is the same as that of our computer, since our server is on our computer, we do the following procedures that we have set.

```
connection.on('event', (event) => console.log('Event received', event));
connection.on('status', (status) => console.log('Status received', status));
router.get('/knx/light/:sw', (req, res, next) => {
  const { sw } = req.params;
  const on = sw === 'on';
  var g1 = req.query.num;
  var groupadd = req.query.group;
  //var groupadd=g2.replace(/-/g, '/')
  var type = req.query.type;
  var id = req.query.id;
  var idfloors = req.query.idfloors;
```

In order to perform the following procedures we need to have some data that more appropriately determines the actions to be followed. These variables are group, idfloors, id,num and type where they are sent via "Ajax" command, get method, the LampOn function and LampOff from the right-sidebar.php file of our platform.

LampOn function ,Lampoff file right-sidebar.php

```
function LampON(group,idfloors,id,num,type){
  $.ajax({ URL : ulron,
    method : 'get',
    data: {
      group:group,idfloors: idfloors,id:id,num:num,type:type
    },
    success : function(data){},
  })
}
function LampOff(group,idfloors,id,num,type){
  $.ajax({ url : ulroff,
    method : 'get',
    data: {
      group:group,idfloors: idfloors,id:id,num:num,type:type
    },
    success : function(data){},
  })
}
```

In the continuation of our code in the file /routes/index.js some checks are made, so that the appropriate actions can be taken. An initial check is made if the element from which the command has been sent is a luminaire (type=L) or socket (type=P). If it is a luminaire we check if sw, ie the message sent is on or off and we set the callback variable if it is active with a value of "1", we set the value time='timeon' and call the function get_info(). Otherwise, we set the callback variable with a value of '0' and the variable time='timeoff' and call a function get_info() with the appropriate variables. A similar procedure is performed if it is a socket (type=P) only the function get_info1() is called. These actions are done to store data, such as whether the light or socket is active, to have how much time it was on, so that by multiplying then by its consumption we know how much each luminaire has consumed.

```
if(type=="L"){
  if(sw=="on"){
    console.log("ON");
    console.log(idfloors);
    callback="1";
    time="timeon";
    get_info(callback,g1,id,idfloors,time);
  }
}
```

```

else{
  callback="0";
  time="timeoff";
  get_info(callback,g1,id,idfloors,time);
}
}
else if(type==="P"){
  if(sw==='on'){
    console.log("ON");
    console.log(idfloors);
    callback="1";
    time="timeon";
    get_info1(callback,g1,id,idfloors,time);
  }
}
else{
  callback="0";
  time="timeoff";
  get_info1(callback,g1,id,idfloors,time);
}
}
}

```

Function get_info()

In this function we store in our database whether a luminaire is activated or not, what time it has been activated and day so that we know how long it was active and therefore know how much consumption it had per day.

```

function get_info(callback,g1,id,idfloors,time){
  var mysql = require('mysql');
  // Timeon - Timeoff
  var date = new Date();
  var dateSt =
    date.getFullYear() + "-" +
    ("00" + (date.getMonth() + 1)).slice(-2) + "-" +
    ("00" + date.getDate()).slice(-2) ;
  var dateStr =
    ("00" + date.getHours()).slice(-2) + ":" +
    ("00" + date.getMinutes()).slice(-2) + ":" +
    ("00" + date.getSeconds()).slice(-2);

```

First, we take the current time and date and separate it into date (dateSt) and time (dateStr).

Table Lamps

+ Επιλογές

userid	floor	lampnum	X	Y	LampOn	GroupAddress	Watts	dateuse	timeon	timeoff
1	2	1	35	30	0	0/0/1	20	NULL	NULL	NULL
1	2	2	35	75	0	0/0/2	15	NULL	NULL	NULL
1	2	3	65	40	0	0/0/3	25	NULL	NULL	NULL
1	1	1	35	40	0	0/0/1	10	NULL	NULL	NULL
1	1	2	35	70	0	0/0/2	12	NULL	NULL	NULL
1	1	3	60	70	0	0/0/3	13	NULL	NULL	NULL
1	1	4	60	40	0	0/0/4	15	NULL	NULL	NULL

Lampcon

panel

datet	userid	floorid	itemid	consu	timeuse
2020-09-22	1	1	1	8	00:00:00
2020-09-22	1	1	2	10	00:00:00
2020-09-22	1	1	3	13	00:00:00
2020-09-22	1	1	4	9	00:00:00

(To understand what the get_info() function does best, it's good to have an idea of how our database tables have been created.)

Then a connection is made to our database, we update the lampcon data (luminaire activation), dateuse (date of use) with the data that has been transferred with the function call. In addition, timeoff or timeon items are updated depending on what the time and datestr variable contains, where the current time is pre-entered. Then a check is made if the time variable is timeoff in order to find the difference between timeon and timeoff time in order to enter in our table lampcon to see later from some operations how much is the consumption for each luminaire. This is then done through the 'TIMEDIFF' command where we find the difference between timeon and timeoff. The additional problem that was created during the writing of this code was that there should be a further check for the change of the day in the consumption time in the table. This is then done where we check in our lampcon table if there is a datet date. If the datet variable, i.e. the one we have taken from our lampcon table, is the same as the current dateSt, then operations are performed to find the consumption that has been made depending on the value of the difference we have previously found and the value consumed by each luminaire. Otherwise, if the variables of our test are different, we perform operations to find a difference in consumption and we register it (using the 'INSERT' command of mysql) as a new element in our lampcon table since it is considered as a new day.

```
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "owners"
});
con.connect(function(err) {
```

```

    var sql = "UPDATE lamps SET LampOn = '"+callback+"',dateuse = '"+dateSt+"',"+time+"= '"+dateSt
r+"' WHERE lampnum= '"+g1+"' and userid='"+id+"' and floor='"+idfloors+"'";
    con.query(sql, function (err, result) {
        if (err) throw err;
        //console.log(result.affectedRows + " record(s) updated");
    });
    if(time=="timeoff"){
        var sql1 = "SELECT TIMEDIFF(timeoff,timeon) AS tim,watts from lamps where lampnum= '"+g1+
"' and userid='"+id+"' and floor='"+idfloors+"' LIMIT 1";

        var sql5 = "SELECT datet from lampcon WHERE itemid= '"+g1+"' and userid='"+id+"' and floorid=
 '"+idfloors+"' ORDER BY datet DESC LIMIT 1";
        con.query(sql5, function (err, result5) {

```

The activation and deactivation of the luminaires is done through the following function where the "connection" is the coupling created at the beginning between our computer and the device we want to communicate. We declare the action "action" with the address on our device (groupadd) that we want to activate and the parameter 'on' which indicates activation. Then we create the link to have a duration of 1000 milisecc and disconnect the link.

```

const toggleLight = () => {
    connection.Action(groupadd, on);
};

connection.Connect(function () {
    setTimeout(toggleLight, 1000);

    setTimeout(() => connection.Disconnect(), 1000);
});

res.send(`The light is ${on ? 'on' : 'off'}`);
});

```

VIDEO: <https://www.youtube.com/watch?v=oyxowxQYrt0>

Future Additions

This thesis may have many future upgrades that will offer users an even better experience and capabilities.

Power measurement

A useful feature that can have in a future upgrade is the measurement of power through the activator, so that if we remove and install a new device we can be directly informed of the consumption of each device we connect. This will achieve a more accurate measurement of each user's consumption.

Better Separation of Spaces

For large houses or hotels it will be difficult to depict the devices in a floor plan. Therefore, it would be more correct if we divide the spaces by room so that it is better to visualize the devices that each house or hotel can have.

More Security

Security for such a platform is a fairly important part. There are many malicious people who will attempt to manipulate the devices or even steal through any loopholes that may exist on such a platform. Therefore, in future upgrades security should be one of the key elements that we need to address. We should check by sending messages to the user's mobile phone, where we will initially set what his number is and we will check whether the connection is valid based on this message. In addition, if the account has been accessed from another device or someone else has logged in at the same time, a message should be sent to the user's mobile phone or email. These are some ways to upgrade the platform and provide more security for each user.

Additional devices

In a future upgrade, more and more sensors can be added, such as for doors, windows, movement and more. This will give many additional features to each user.

Artificial intelligence

One of these future upgrades is artificial intelligence (AI) where, with the help of artificial intelligence methods and algorithms, ways and actions for smarter and more economical use of household appliances can be suggested, such as using the washing machine during off-peak or evening hours.

Help the Blind

A vision for the future upgrade of this platform is to help people who have a disability problem. This will be achieved through a sensor, which will give coordinates based on the floor plan of the house where the user is located. With the touch on a tablet there will be voice commands to choose the place he wants to go. So putting in the space he wants to go he will get voice commands to go to the space he has set easily and there will be additional sound that will inform him whether he is near or far.

Conclusions

With the implementation of such a task I feel that I have gained a lot.

In the first part, where the platform was created, we had to search and learn a lot about web programming, databases and think about how the design of the platform would be ergonomically appropriate. So this way we have designed the platform is quite ergonomic for the user since he can simply manage his lighting fixtures through the floor plan of his house. In a future upgrade, as I have mentioned above, it will be good to separate the floor plan of the house, where in cases where there are many lighting fixtures and appliances in a space it will be more comfortable to display them, especially on mobiles. In addition, the design is ergonomic for the administrator due to the fact that he should not spend effort and time introducing a new user since he can simply and easily add a new user just with a picture of the floor plan of his house. So we believe that the construction of the platform has reached a very satisfactory point, it brings improvements that will be introduced in the future.

The wiring of the construction was a completely different part of the work since we had left the programming part and had to work more as electricians, since we had to deal with wiring electrical appliances through which a relatively large current enters capable of causing a problem to a person. So with safety, proper organization, study and cooperation we connected the devices correctly so that they are ready to communicate with our platform.

The third part was the most difficult but at the same time the most important since without it the work could not be realized. It is the link between the first and second parts of the work. It was the part that was the most unknown to us and we had to search further to find helper codes that

would work. Fortunately, after a lot of searching we found code that was to some extent functional but with further modification of the code we got to what we needed.

In all the time I had been involved in this project, I never stopped hoping and searching patiently to achieve the goals we had set from the beginning. The satisfaction you receive when a vision you had becomes reality rewards the efforts and time you have devoted to it. That's why you never stop dreaming, realizing your goals and the things you envision.