

FACULTATEA CALCULATOARE, INFORMATICA SI
MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A
PRODUSELOR SOFT

LUCRAREA DE LABORATOR#4

Web development

Autor:
Alexandr IALTICENCO

lector asistent:
Irina COJANU
lector superior:
Svetlana COJOCARU

Obiectivele lucrării

- Realizarea unui simplu Web Site personal
- Familiarizarea cu HTML si CSS
- Interactiuni Javascript

Sarcina lucrării

Advanced level:

- Site-ul trebuie sa contina AJAX Requests.
- Implimentarea XHR sau JSON responses. Careva din informatie trebuie sa fie dinamic incarcata pe pagina.

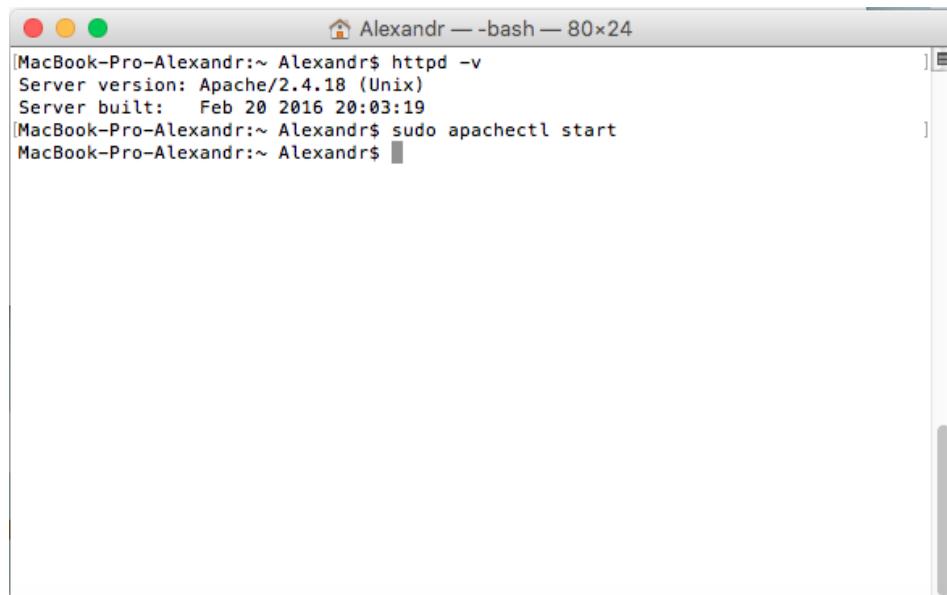
Cerinte tehnice

- Foloseste MVC (ModelViewController) pattern

1 Configurarea Apache + PHP + MySQL

1.1 Setarea serverului Apache

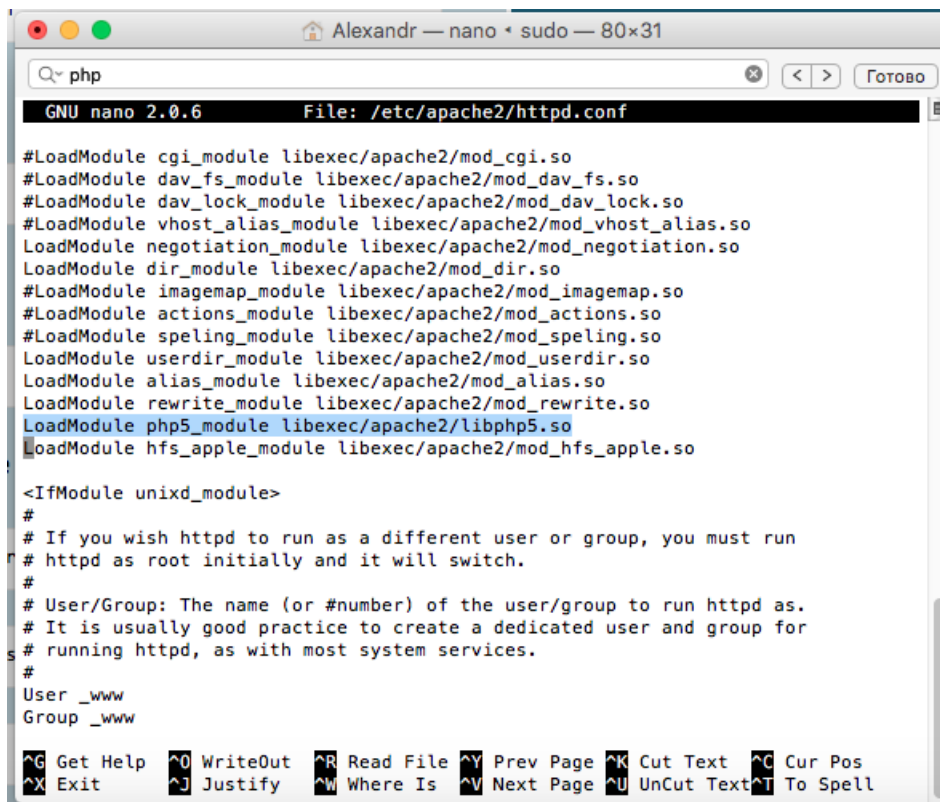
Serverul Apache deja este integrat in sistemul de operare Mac OS X si poate fi usor instalat in alte sisteme de operare.

A screenshot of a macOS terminal window. The title bar shows a home icon, the name 'Alexandr', and the command prompt '-bash' with a window size of '80x24'. The terminal content shows the user running 'httpd -v' which outputs 'Server version: Apache/2.4.18 (Unix)' and 'Server built: Feb 20 2016 20:03:19'. Then the user runs 'sudo apachectl start' and the prompt returns. The terminal has a scrollbar on the right side.

```
MacBook-Pro-Alexandr:~ Alexandr$ httpd -v
Server version: Apache/2.4.18 (Unix)
Server built:   Feb 20 2016 20:03:19
MacBook-Pro-Alexandr:~ Alexandr$ sudo apachectl start
MacBook-Pro-Alexandr:~ Alexandr$
```

1.2 Setarea PHP

Pentru a seta php am comentat rindul corespunzator in fisierul `/etc/apache2/httpd.conf` La fel ca si Apache el este preinstalat in sistemul de operare OS X. Acesta este foarte comod in cazul in care cineva are nevoie de a configura un web server foarte rapid.



```
GNU nano 2.0.6 File: /etc/apache2/httpd.conf

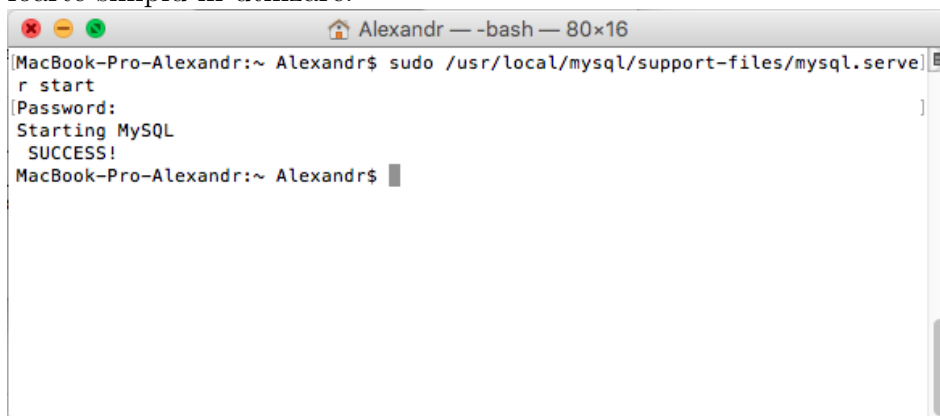
#LoadModule cgi_module libexec/apache2/mod_cgi.so
#LoadModule dav_fs_module libexec/apache2/mod_dav_fs.so
#LoadModule dav_lock_module libexec/apache2/mod_dav_lock.so
#LoadModule vhost_alias_module libexec/apache2/mod_vhost_alias.so
LoadModule negotiation_module libexec/apache2/mod_negotiation.so
LoadModule dir_module libexec/apache2/mod_dir.so
#LoadModule imagemap_module libexec/apache2/mod_imagemap.so
#LoadModule actions_module libexec/apache2/mod_actions.so
#LoadModule speling_module libexec/apache2/mod_speling.so
LoadModule userdir_module libexec/apache2/mod_userdir.so
LoadModule alias_module libexec/apache2/mod_alias.so
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
LoadModule php5_module libexec/apache2/libphp5.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User _www
Group _www

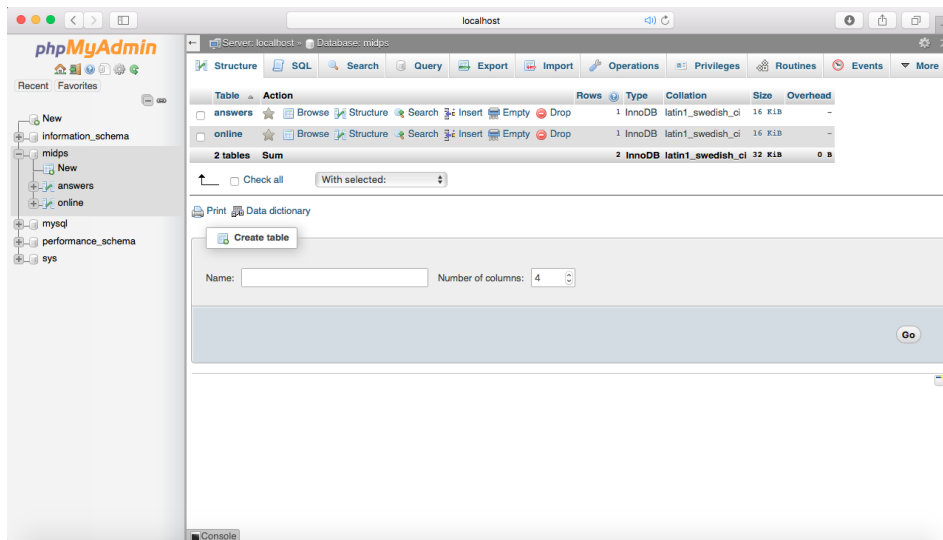
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

1.3 Setarea MySQL

MySQL a fost descarcat de pe site-ul oficial si usor instalat ca un os x package. Pentru comoditate am instalat si configurat phpmyadmin. El este gratuit si foarte simplu in utilizare.

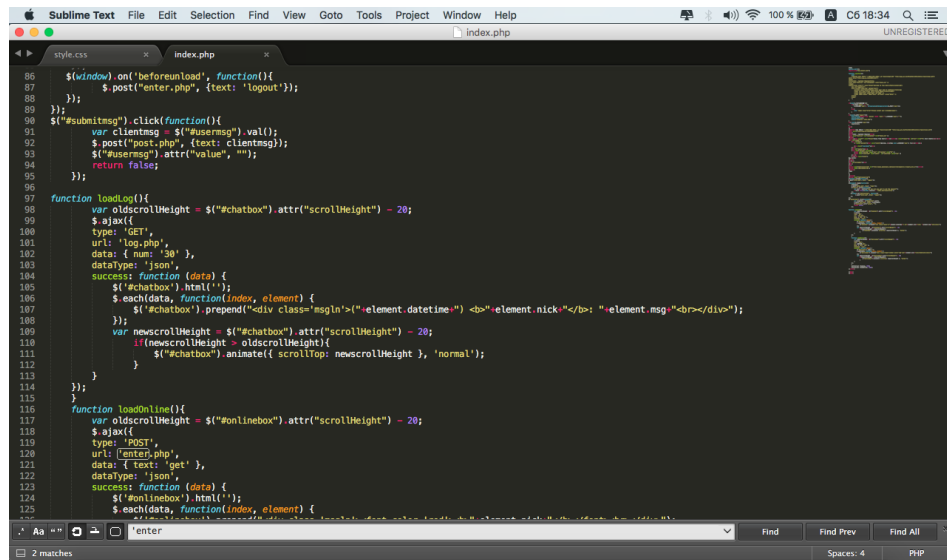


```
MacBook-Pro-Alexandr:~ Alexandr$ sudo /usr/local/mysql/support-files/mysql.server start
r start
[Password: ]
Starting MySQL
SUCCESS!
MacBook-Pro-Alexandr:~ Alexandr$
```



1.4 Sublime Text

Ca mediul interactiv de dezvoltare a fost ales Sublime Text, fiindca el este usor, simplu si cu interfata comoda. Mediul dat ofera programatorului Web toate mijloace pentru o munca eficienta.



```
86 $(window).on('beforeunload', function(){
87     $.post('enter.php', {text: 'logout'});
88 });
89
90 $('#submitmsg').click(function(){
91     var clientmsg = $('#usermsg').val();
92     $.post('post.php', {text: clientmsg});
93     $('#usermsg').attr('value', '');
94     return false;
95 });
96
97 function loadLog(){
98     var oldscrollHeight = $('#chatbox').attr('scrollHeight') - 20;
99     $.ajax({
100         type: 'GET',
101         url: 'log.php',
102         data: { num: '30' },
103         dataType: 'json',
104         success: function (data) {
105             $('#chatbox').html('');
106             $.each(data, function(index, element) {
107                 $('#chatbox').prepend("<div class='msgln'>"+element.datetime+" <b>"+element.nick+"</b>: "+element.msg+"<br></div>");
108             });
109             var newscrollHeight = $('#chatbox').attr('scrollHeight') - 20;
110             if(newscrollHeight > oldscrollHeight){
111                 $('#chatbox').animate({ scrollTop: newscrollHeight }, 'normal');
112             }
113         }
114     });
115 }
116
117 function loadOnline(){
118     var oldscrollHeight = $('#onlinebox').attr('scrollHeight') - 20;
119     $.ajax({
120         type: 'POST',
121         url: 'enter.php',
122         data: { text: 'get' },
123         dataType: 'json',
124         success: function (data) {
125             $('#onlinebox').html('');
126             $.each(data, function(index, element) {
```

2 Listinul

2.1 Pagina principala

Pagina index.php reprezinta un mix de html (partial generat de scripturile php) si codul sursa scris in limbajul javascript. Codul php va fi executat de server si nu va fi transmis la client. La inceput serverul verifica, daca este ori nu setat parametru 'name' de sesiunea curenta si daca nu, afiseaza forma de logare. In caz contrar utilizatorul o sa vada chat-ul.

Listing 1: Fisierul index.php. Logarea

```
<?php
session_start();
require_once("bd_connect.php");

function loginForm(){
    echo'
        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
            "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Free Chat</title>
<link type="text/css" rel="stylesheet" href="style.css" />
</head>
<center><font size="5" color="black">Welcome to free
    Chat!</font></center><br>
    <div id="loginform">
        <form action="index.php" method="post">
            <p><font color="white">Please enter your name to
                continue:</font></p>
            <label for="name"><font color="white">Name:</font></label>
            <input type="text" name="name" id="name" />
            <input type="submit" name="enter" id="enter" value="Enter" />
        </form>
    </div>
    '
}

if(isset($_POST['enter'])){
    if($_POST['name'] != ""){
        $_SESSION['name'] = stripslashes(htmlspecialchars($_POST['name']));
    }
    else{
        echo '<span class="error">Please select your nickname</span>';
    }
}

if(isset($_GET['logout'])){
    mysql_query("DELETE from 'online' WHERE
        'nick'='". $_SESSION['name'] ."'");
    session_destroy();
    header("Location: index.php");
}

if(!isset($_SESSION['name'])){
    loginForm();
}
```

Chat-ul reprezinta un simplu div, continutul caruia se modifica prin intermediul jquery (biblioteca javascript). Scriptul transmite ajax requests, primeste inapoi json-response si-l converteaza in html, care se adauga la html-continutul div-ului chatbox.

Listing 2: Fisierul index.php. Chatbox-ul

```
else{
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Chat - Customer Module</title>
<link type="text/css" rel="stylesheet" href="style.css" />
</head>
<center><font size="5" color="black">Enjoy free Chat!</font></center><p
    class="logout"><a id="exit" href="#"> (Exit Chat)</a><br><br>
<div id="wrapper">
    <div id="menu">
        <p class="welcome"><font color="white">Welcome, <b><?php echo
            $_SESSION['name']; ?></b></font></p>

        <div style="clear:both"></div>
    </div>
    <div id="chatbox"></div>
    <form name="message" action="">
        <input name="usermsg" type="text" id="usermsg" size="63" />
        <input name="submitmsg" type="submit" id="submitmsg" value="Say" />
    </form>
    <br><font color="white">
Now online:
</font>
<br><br>
<div id="onlinebox"></div>
</div>
<br>
```

Acum trebuie sa includem biblioteca jquery de versiunea 1.7 pentru a face lucru cu javascript mai rapid si simplu

Listing 3: Fisierul index.php. Includerea bibliotecii jquery

```
<br>
<?php
}
?>
<script type="text/javascript"
    src="http://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js">
</script>
```

Listing 4: Fisierul index.php. Dezlogarea la inchiderea paginii web

```
<script type="text/javascript">
$(window).unload(function(){
$.post("enter.php", {text: 'logout'});
})
```

În cazul logării sau dezlogării suntem obligați să informăm scriptul enter.php că lista de utilizatori online să fie stocată în baza de date.

Listing 5: Fisierul index.php. On document ready

```
$(document).ready(function(){
    loadLog();
    $.post("enter.php", {text: 'login'});
    $("#exit").click(function(){
        var exit = confirm("Are you sure you want to end the
            session?");
        if(exit==true){window.location = 'index.php?logout=true';}
    });
    $(window).on('beforeunload', function(){
        $.post("enter.php", {text: 'logout'});
    });
});
```

apelăm funcția loadLog() pentru a iniția ajax request care va descărca de pe server ultimele răspunsuri ale utilizatorilor (ca să nu fie chatbox-ul gol exact după logare). Apoi informăm scriptul enter.php că utilizatorul deja a logat. Mai departe setăm link-ul exit ca în cazul apăsării lui să apară întrebarea dacă utilizatorul cu adevărat vrea să termine sesiunea. La sfârșit setăm ca înainte de închidere a paginii de către utilizator să fie informat scriptul enter.php.

Listing 6: Fisierul index.php. Trimiterea mesajelor scrise de utilizator

```
$("#submitmsg").click(function(){
    var clientmsg = $("#usermsg").val();
    $.post("post.php", {text: clientmsg});
    $("#usermsg").attr("value", "");
    return false;
});
```

In cazul apasarii butonului submitmsg prin intermediul post request trimitem mesajul curent scriptului post.php (metoda .post de fapt reprezinta o varianta scurtata a ajax-ului).

Listing 7: Fisierul index.php. metoda loadLog()

```
function loadLog(){
    var oldscrollHeight = $("#chatbox").attr("scrollHeight") - 20;
    $.ajax({
        type: 'GET',
        url: 'log.php',
        data: { num: '30' },
        dataType: 'json',
        success: function (data) {
            $('#chatbox').html('');
            $.each(data, function(index, element) {
                $('#chatbox').prepend("<div  

                    class='msgln'>("+element.datetime+")  

                    <b>"+element.nick+"</b>: "+element.msg+"<br></div>");
            });
            var newscrollHeight = $("#chatbox").attr("scrollHeight") - 20;
            if(newscrollHeight > oldscrollHeight){
                $("#chatbox").animate({ scrollTop: newscrollHeight },
                    'normal');
            }
        }
    });
}
```

Prin intermediul loadLog() descarcam de pe server log-ul mesajelor si le afisam in chatbox. Pentru acesta utilizam ajax http request de tip get cu parametru "num" care contine numarul ultimelor mesaje care vor fi trimise inapoi. Scriptul apelat log.php este setat in asa fel ca sa reintoarca datele in format-ul json.

Listing 8: Fisierul index.php. metoda loadOnline()

```
function loadOnline(){
    var oldscrollHeight = $("#onlinebox").attr("scrollHeight") - 20;
    $.ajax({
        type: 'POST',
        url: 'enter.php',
        data: { text: 'get' },
        dataType: 'json',
        success: function (data) {
            $('#onlinebox').html('');
            $.each(data, function(index, element) {
                $('#onlinebox').prepend("<div class='msgln'><font  
color='red'><b>" + element.nick + "</b></font><br></div>");
            });
            var newscrollHeight = $("#onlinebox").attr("scrollHeight") - 20;
            if(newscrollHeight > oldscrollHeight){
                $("#onlinebox").animate({ scrollTop: newscrollHeight }, 'normal');
            }
        }
    });
}
```

Prin intermediul acestei metode descarcam de pe server lista utilizatorilor care sunt acum ONLINE.

Listing 9: Fisierul index.php. setarea intervalului actualizarii log-ului de mesaje si listei de utilizatori

```
setInterval (loadLog, 500);
setInterval (loadOnline, 1500);
</script>
</body>
</html>
```

2.2 Scriptul post.php

Scriptul post.php primește ajax-requests de la pagina principală (index.php) și le procesează în așa fel ca informația să fie stocată în baza de date.

Listing 10: Fisierul post.php

```
<?
require_once("bd_connect.php");
session_start();
if(isset($_SESSION['name'])){
    mysql_query("INSERT INTO 'answers' ('msg','stamp','datetime','nick')
VALUES ('".$_POST['text']."'.'','.time().'.'.date("g:i
A").'".$_SESSION['name']."'.'')") or die(mysql_error());
}
?>
```

2.3 Scriptul bd-connect.php

Scriptul bd-connect.php stabileste conexiunea cu baza de date. El este inclus in alte scripturi ca codul sa nu fie dublat.

Listing 11: Fisierul bd-connect.php

```
<?php
header('Content-type: text/html; charset=utf-8');
$db_host = 'localhost';
$db_name = 'midps';
$db_username = 'tester';
$db_password = '12345678';
$connect_to_db = mysql_connect($db_host, $db_username, $db_password) or
    die("Could not connect: " . mysql_error());
mysql_select_db($db_name, $connect_to_db) or die("Could not select DB: "
    . mysql_error());
mysql_query("set names utf8");
date_default_timezone_set('Europe/Chisinau');
?>
```

2.4 Scriptul log.php

Scriptul log.php extrage mesajele scrise de utilizatori din baza de date si le afiseaza in formatul json.

Listing 12: Fisierul log.php

```
<?php
require_once("bd_connect.php");
$text = $_GET['num'];
$result = mysql_query("SELECT * from 'answers' ORDER BY id DESC LIMIT
    ".$_GET['num']."") or die(mysql_error());
$return_arr = array();
while ($row = mysql_fetch_array($result)){
    $row_array['msg'] = $row['msg'];
    $row_array['nick'] = $row['nick'];
    $row_array['datetime'] = $row['datetime'];
    array_push($return_arr,$row_array);
}

echo json_encode($return_arr);

?>
```

2.5 Scriptul enter.php

Scriptul enter.php desfasoara un control asupra logarii si dezlogarii a utilizatorilor in asa fel ca in baza de date (tabelul online) mereu sa fie lista actuala a utilizatorilor online. In cazul in care parametru text va fi egal cu login, scriptul va adauga numele utilizatorului la baza de date. In cazul logout - il va sterge. In al treilea caz scriptul va extrage din baza de date lista utilizatorilor logati. Informatiile vor fi puse in json si afisate.

Listing 13: Fisierul enter.php

```
<?
require_once("bd_connect.php");
session_start();
if(isset($_SESSION['name'])){
    $text = $_POST['text'];
    if ($text == "login")
        mysql_query("INSERT INTO 'online' ('nick') VALUES
            ('".$_SESSION['name']."'");
    else
        if ($text == "logout")
            mysql_query("DELETE from 'online' WHERE
                'nick'='".$_SESSION['name']."'");
    else {
        $result = mysql_query("SELECT * from 'online'") or
            die(mysql_error());
        $return_arr = array();
        while ($row = mysql_fetch_array($result)){
            $row_array['nick'] = $row['nick'];
            array_push($return_arr,$row_array);
        }

        echo json_encode($return_arr);
    }
}
?>
```

2.6 Fisierul de marcare style.css

Proiectul dat utilizeaza un fisier de stil foarte simplu care descrie aparitia chat-ului si alinierea inscriptiilor

Listing 14: Fisierul style.css

```
/* CSS Document */
body {
    font:12px arial;
    color: #222;
    text-align:center;
    padding:35px; }

form, p, span {
    margin:0;
    padding:0; }

input { font:12px arial; }

a {
    color:#0000FF;
    text-decoration:none; }

    a:hover { text-decoration:underline; }

#wrapper, #loginform {
    margin:0 auto;
    padding-bottom:25px;
    background:#363636;
    width:504px;
    border:1px solid #969696; }

#loginform { padding-top:18px; }

    #loginform p { margin: 5px; }

#chatbox {
    text-align:left;
    margin:0 auto;
    margin-bottom:25px;
    padding:10px;
    background:#fff;
    height:270px;
    width:430px;
    border:1px solid #ACD8F0;
    overflow:auto; }
```

Listing 15: Fisierul style.css

```
#onlinebox {
    text-align:left;
    margin:0 auto;
    margin-bottom:25px;
    padding:10px;
    background:#EBF4FB;
    height:64px;
    width:430px;
    border:1px solid #ACD8F0;
    overflow:auto; }

#usermsg {
    width:395px;
    border:1px solid #ACD8F0; }

#submit { width: 90px; }

.error { color: #ff0000; }

#menu { padding:12.5px 25px 12.5px 25px; }

.welcome { float:center; }

.logout { float:center; }

a {
    color:#ff0000;
}

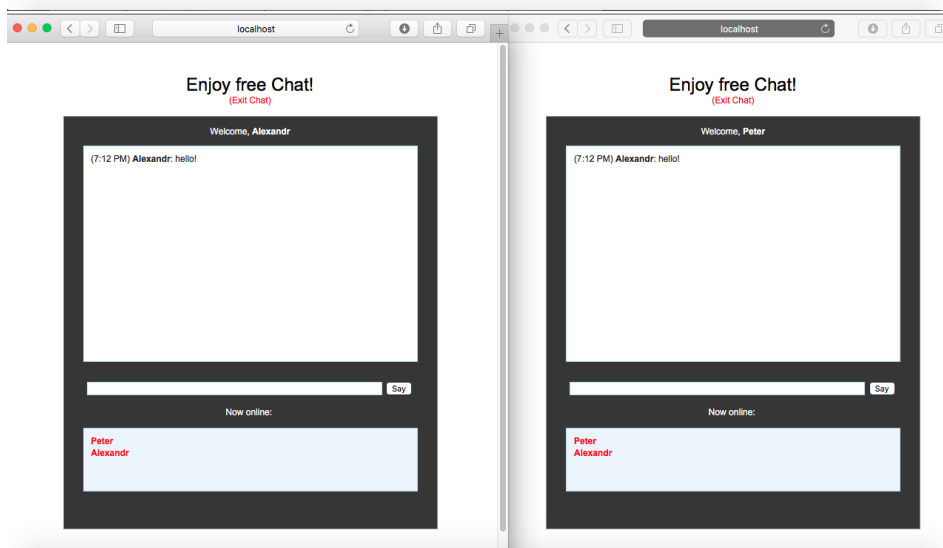
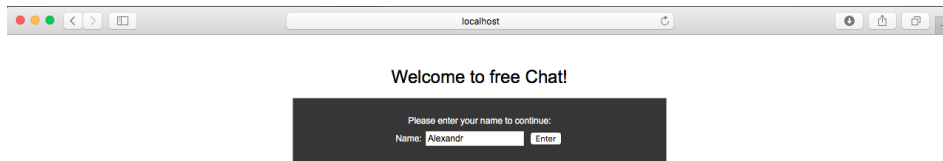
.msgln { margin:0 0 2px 0; }
```

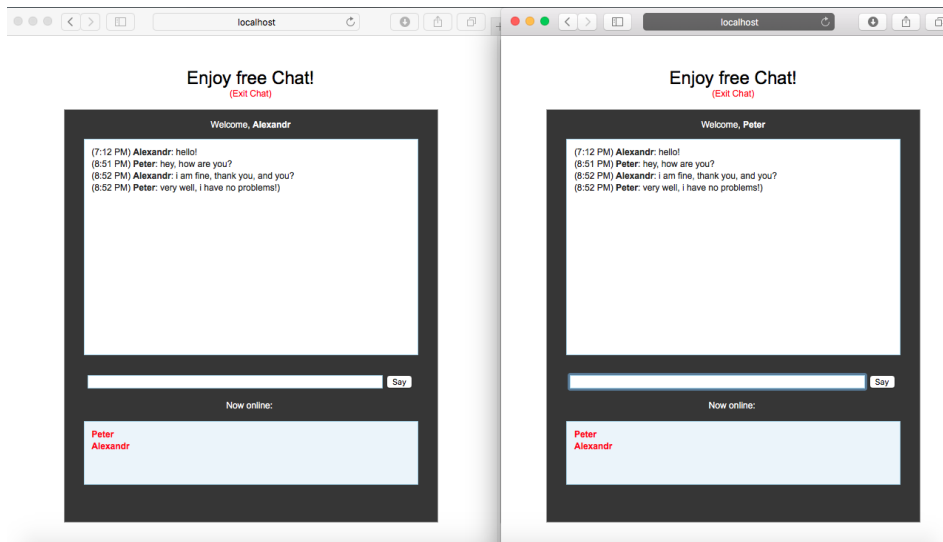
3 Folosirea MVC (ModelViewController) pattern

În cadrul acestui proiect ca un model de arhitectură a fost utilizat Model-View-Controller pattern. Astfel utilizatorul acționează asupra controalelor (butonul submit, câmpul de text cu mesajul trimis). Aceste controalele manipulează modelul (scripturile post.php, log.php, enter.php), iar modelul în rândul său actualizează View (în cazul nostru div-ul chatbox și div-ul onlinebox). Prin intermediul metodelor loadLog() și loadOnline() componen-

tele arhitecturii de tip View isi schimba continutul (ele sunt subscribe la actualizarile modelelor).

4 Rezultatele





Concluzii

In cadrul acestei lucrari de laborator am creat simplu web chat prin intermediul html, php, javascript si MySQL. Pentru a atinge acest scop am configurat un web-server pe localhost, am invatat arhitectura MVC si am instalat mediul interactiv de dezvoltare Sublime Text. Cunostintele obtinute pe parcursul desfasurarii lucrarii de laborator vor fi utile pentru realizarea proiectelor ce urmeaza.

Bibliografie

1. <https://coolestguidesontheplanet.com/get-apache-mysql-php-and-phpmyadmin-working-on-osx-10-11-el-capitan/> - **Get Apache, MySQL, PHP and phpMyAdmin working on OSX 10.11 El Capitan**
2. <https://api.jquery.com> - **JQuery API Documentation**
3. <http://php.net/manual/en/> - **PHP Manual**