

Министерство Просвещения Республики Молдова  
Технический Университет Молдовы  
Кафедра Автоматики и Информационных Технологий

# ОТЧЁТ

по лабораторной работе № 4  
по дисциплине «MIDPS»  
на тему: «Web development»

Выполнила:  
Проверил:

ст. гр. TI-145 Кысса. В.  
ст. преп. Кожокару С.

Кишинев 2016

## Obiectivele lucrării

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

## Sarcina lucrării

### Advanced level:

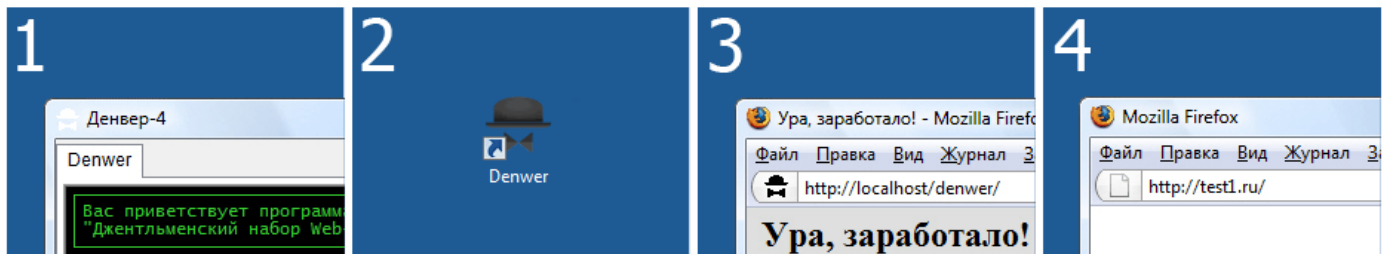
- Site-ul trebuie sa contina AJAX Requests.
- Implimentarea XHR sau JSON responses. Careva din informatie trebuie sa fie dinamic incarcata pe pagina.

## 1. Apache + PHP + MySQL

### 1.1 Установка Apache

Локальный сервер (Apache, PHP, MySQL, Perl и т.д.) и программная оболочка, используемые Web-разработчиками для разработки сайтов на «домашней» (локальной) Windows-машине без необходимости выхода в Интернет. Для этого нам понадобится Denwer. Главная особенность Денвера — удобство при удаленной работе сразу над несколькими независимыми проектами и возможность размещения на Flash-накопителе. Первым делом скачиваем и устанавливаем Денвер:

Начать работу с Денвером просто



- 1) Щелкаем по скачанному архиву Денвера. Запустится инсталлятор.
- 2) После завершения установки щелкаем по ярлыку Denwer на Рабочем столе.
- 3) Открываем в браузере страницу <http://localhost> (хранится в папке `home/localhost/www`).

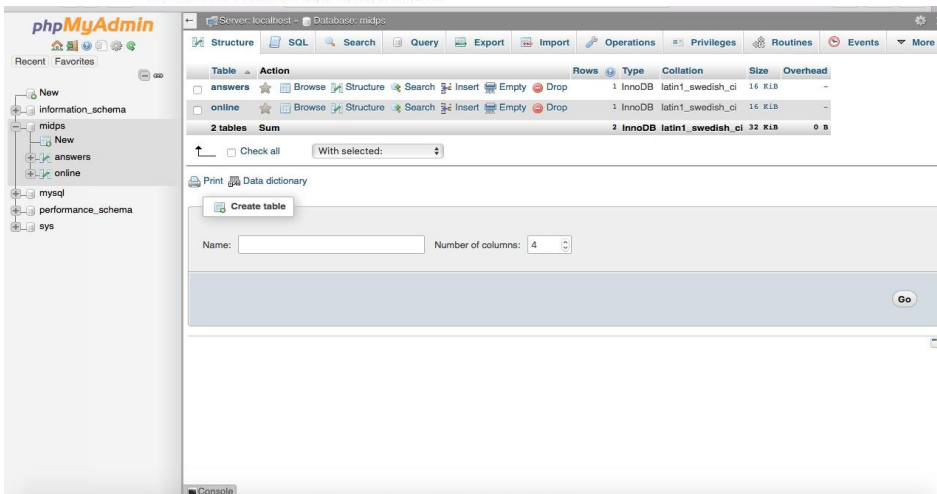
Чтобы начать использовать Денвер, проделайте следующие действия:

1. Запустите Денвер, воспользовавшись ярлыком **Start Denver** на Рабочем столе. Если вы не создавали ярлыки, то можно запустить Денвер по команде `cd \server\denver\run.exe`.
2. Откройте браузер и перейдите по адресу <http://localhost>.
3. Вы должны увидеть главную страницу Денвера.
4. *Примечание:* если вы используете **Skype**, убедитесь, что он не занимает порты 80 и 443, необходимые для работы Денвера (Инструменты - Настройки - Соединение - Использовать порты... должно быть отключено). Либо же просто завершите работу **Skype**.
5. **Внимание!** Если после запуска Денвера <http://localhost> не открывается, проверьте, не блокируется ли Денвер вашим антивирусом или фаерволом. Например, были замечены проблемы с NOD32 в Windows XP (в нем нужно добавить процесс `Y:\usr\local\apache\bin\httpd.exe` в список исключений, это можно сделать в окне IMON Настройка Разное/Исключение).

Если по каким-то причинам Денвер не заработал, свяжитесь, пожалуйста, с разработчиками: <http://forum.dklab.ru/denver-bugs/>. Прикрепите к сообщению следующую информацию:

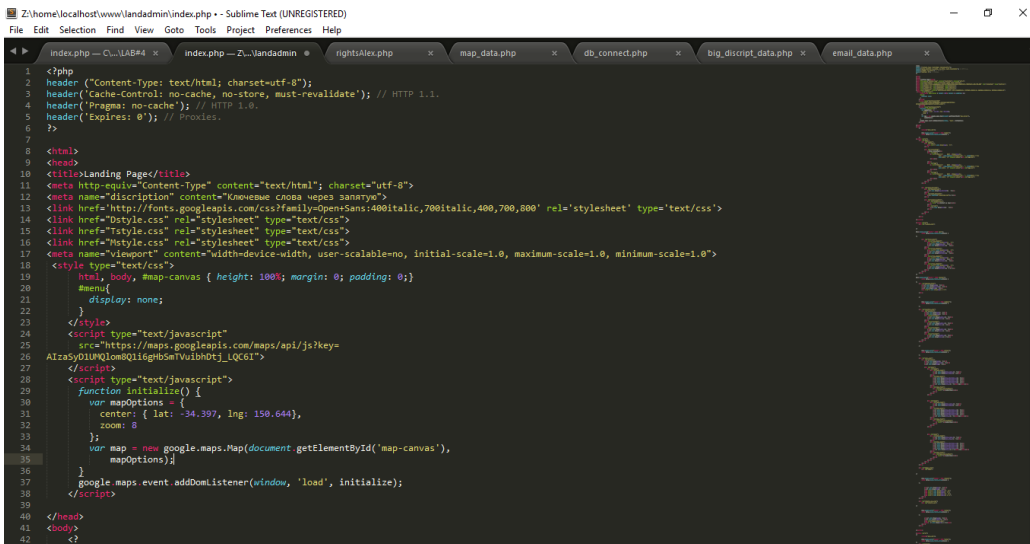
1. При каких условиях проявился баг? Что вы сделали перед тем, как его зафиксировать?
2. Точную версию Вашей ОС (можно получить по команде `winver`, запущенной в Командной строке).
3. Файл `netstat.txt`, получившийся в результате работы команды `netstat -ab > C:\netstat.txt` (кстати, этот файл не содержит персональной информации или сведений, подрывающих безопасность системы, хотя на неисключенный взгляд он и может показаться подозрительным).
4. Значимые сообщения из конца файла `/usr/local/apache/logs/error.log`.

Спасибо за использование Денвера!



## 1.2 Sublime Text

Как среду разработки выбираем Sublime Text, так как он простой в использовании и обладает удобным интерфейсом.



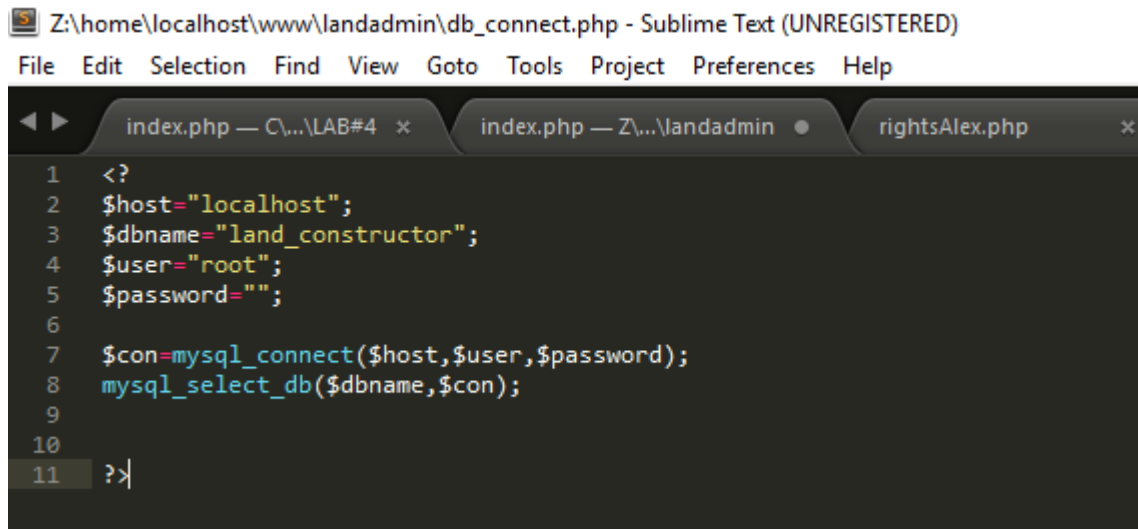
## 2 Листинг

### 2.1 План работы:

- Создаем html – макет;
- Оформляем стили css;
- Создаем базу с нужным количеством таблиц, заполняем их необходимой нам информацией;
- Соединяем базу с html с помощью php;

Файл index.php представляет собой микс из html, php и javascript.

Функция `mysql_connect ()` используется, чтобы установить соединение с базой данных MySQL и передать возвращение этой функции переменной.



`header("Content-Type: text/html; charset=utf-8");` -указываем кодировки Unicode. Чем более широко будет использоваться кодирование символов, тем выше вероятность того, что браузер будет его понимать. Такое кодирование Unicode, как UTF-8 является хорошим выбором по ряду причин.

Строчки кода, запрещающие кэширование:

```
header('Cache-Control: no-cache, no-store, must-revalidate'); // HTTP 1.1.
header('Pragma: no-cache'); // HTTP 1.0.
header('Expires: 0'); // Proxies.-Прокси
```

А также в коде представленном внизу описаны файлы в которых на css описаны стили.

Dstyle.css – файл, в котором описаны стили для просмотра страницы на компьютере

Tstyle.css– файл, в котором описаны стили для просмотра страницы на планшете

Mstyle.css – файл, в котором описаны стили для просмотра страницы на мобильном телефоне

```

<?php
header ("Content-Type: text/html; charset=utf-8");
header('Cache-Control: no-cache, no-store, must-revalidate'); // HTTP 1.1.
header('Pragma: no-cache'); // HTTP 1.0.
header('Expires: 0'); // Proxies.
?>

<html>
<head>
<title>Landing Page</title>
<meta http-equiv="Content-Type" content="text/html"; charset="utf-8">
<meta name="discription" content="Ключевые слова через запятую">
<link href='http://fonts.googleapis.com/css?family=Open+Sans:400italic,700italic,400,700,800' rel='stylesheet' type='text/css'>
<link href="Dstyle.css" rel="stylesheet" type="text/css">
<link href="Tstyle.css" rel="stylesheet" type="text/css">
<link href="Mstyle.css" rel="stylesheet" type="text/css">
<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
<style type="text/css">
    html, body, #map-canvas { height: 100%; margin: 0; padding: 0;}
    #menu{
        display: none;
    }

```

Страница содержит AJAX, а в данном случае карту, она подгружается после загрузки основного контента страницы, то есть асинхронно. В моем случае, запрос шлется на другой сайт, то есть обращается к серверу google, откуда и берется карта, здесь есть указание на скрип, который скачивается и вставляется в страницу.

```

23     </style>
24     <script type="text/javascript"
25         src="https://maps.googleapis.com/maps/api/js?key=
26     AIzaSyD1UMQlom8Q1i6gHbSmTVuibhDtj_LQC6I">
27     </script>
28     <script type="text/javascript">
29         function initialize() {
30             var mapOptions = {
31                 center: { lat: -34.397, lng: 150.644},
32                 zoom: 8
33             };
34             var map = new google.maps.Map(document.getElementById('map-canvas'),
35                 mapOptions);
36         }
37         google.maps.event.addDomListener(window, 'load', initialize);
38     </script>
39
40 </head>
41 <body>

```

Вся страница для удобной работы поделена на небольшие блоки, которые в коде одна за другой соединяются с базой данной и уже из базы мы получаем текст, картинки, ссылки и так далее.

Например, начинаем работу с первым блоком и соединяя его в созданной базой данных, добавляем всю необходимую информацию.

Связь с базой данных осуществляется в следующих строках:

```
$res=mysql_query("SELECT * FROM header");
```

```
while ($row=mysql_fetch_array($res)).
```

```

<body>
<?
    include('menu.php');

    $res=mysql_query("SELECT * FROM header");
    while ($row=mysql_fetch_array($res)) {

    ?>
<section id="s1">
    <div id="main">
        <div id="header">
            <div id="logo">
                
            </div>

            <div id="connecting">
                <p>Stay connected</p>
                <div id="img1">
                    <a href="http://<?      $str = $row[url_1];
                        $str = preg_replace('#^https?://#', '', rtrim($str,'/'));
                        echo $str; ?>" target="_blank"><div id="img1_hov">

                    </div></a>

                </div>
                <div id="img2">
                    <a href="http://<?      $str = $row[url_2];
                        $str = preg_replace('#^https?://#', '', rtrim($str,'/'));
                        echo $str; ?>" target="_blank"><div id="img2_hov">

                    </div></a>

                </div>
                <div id="img3">
                    <a href="http://<?      $str = $row[url_3];
                        $str = preg_replace('#^https?://#', '', rtrim($str,'/'));
                        echo $str; ?>" target="_blank"><div id="img3_hov">

                    </div></a>

                </div>
            </div>
        </div>
    </div>
</div>

```

```

    <div id="land_discript">
        <div id="big_discript">
            <p><? echo $row[big_discript]; ?></p>
        </div>
        <div id="small_discript">
            <p><? readfile("s_discript.txt"); ?></p>
        </div>

        <div id="contact_button">
            <div id="contact_us_b">
                <a href=""><p>Contact us now</p></a>
            </div>
            <div id="note">
                <p><? echo $row[note]; ?></p>
            </div>
        </div>
    </div>

</div>

</section>

<section id="s2">
    <div id="f_block_wrap">

```

По такому же принципу работаем и дальше: соединяем определенный кусок с базой и из нее получаем необходимую информацию с помощью команд языка php, например, echo – команда для вывода какой-либо информации, row – это команда указывает на столбец, из которого будет взята информация.

```

<?
$res=mysql_query("SELECT * FROM serv");
    while ($row=mysql_fetch_array($res)) {

?>

        <div id="first_block">
            <div class="one_element">
                
                <h2><? echo $row[name_1]; ?></h2>
                <p><? echo $row[discript_1]; ?></p>
            </div>
            <div class="one_element">
                
                <h2><? echo $row[name_2]; ?></h2>
                <p><? echo $row[discript_2]; ?></p>
            </div>
            <div class="one_element">
                
                <h2><? echo $row[name_3]; ?></h2>
                <p><? echo $row[discript_3]; ?></p>
            </div>
            <div class="one_element">
                
                <h2><? echo $row[name_4]; ?></h2>
                <p><? echo $row[discript_4]; } ?> </p>
            </div>
        </div>
    </div>
<?

```

```

$res=mysql_query("SELECT * FROM preview");
    while ($row=mysql_fetch_array($res)) {

?>

        <div id="amazing_think">
            <h2><? echo $row[name]; ?></h2>
            <h3><? echo $row[discript]; ?></h3>
            <p><? echo $row[slog]; ?></p>
            

        </div>

    <?

        $res=mysql_query("SELECT * FROM numbers");
        while ($row=mysql_fetch_array($res)) {

?>

        <div id="counters_wrap">
            <div id="counters">
                <h2><? echo $row[name]; ?></h2>
                <h3><? echo $row[slog]; ?></h3>
                <div class="counter">
                    <h1><? echo $row[number_1]; ?></h1>
                    <p><? echo $row[fact_1]; ?></p>
                </div>
                <div class="counter">
                    <h1><? echo $row[number_2]; ?></h1>
                    <p><? echo $row[fact_2]; ?></p>
                </div>
                <div class="counter">
                    <h1><? echo $row[number_3]; ?></h1>
                    <p><? echo $row[fact_3]; ?></p>
                </div>
                <div class="counter">
                    <h1><? echo $row[number_4]; ?></h1>
                    <p><? echo $row[fact_4]; } ?></p>
                </div>
            </div>
        </div>
    <?

```

```

<?
    $res=mysql_query("SELECT * FROM prices");
    while ($row=mysql_fetch_array($res)) {

    ?>

<div id="serv_wrap">
    <div id="services">
        <h2><? echo $row[name]; ?></h2>
        <h3><? echo $row[discript]; ?></h3>
        <p><? echo $row[slog]; ?></p>

        <div id="plans">
            <div id="plan1">
                <div id="plan_head1">
                    <h2><? echo $row[price_name_1]; ?></h2>
                    <h3><? echo $row[plan_slog_1]; ?></h3>
                </div>
                <div id="plan_main1">
                    <p><? echo $row[price_discript_1_1]; ?></p>
                    <p><? echo $row[price_discript_1_2]; ?></p>
                    <p><? echo $row[price_discript_1_3]; ?></p>
                    <p><? echo $row[price_discript_1_4]; ?></p>
                    <h1><? echo $row[price_1]; ?><span>/m</span></h1>
                </div>
                <div id="plan_foot1">
                    <div id="plan_foot_button1">
                        <a href=""><p>Subscribe</p></a>
                    </div>
                </div>
            </div>

            <div id="plan2">
                <div id="plan_head2">
                    <h2><? echo $row[price_name_2]; ?></h2>
                    <h3><? echo $row[plan_slog_2]; ?></h3>
                </div>
                <div id="plan_main2">
                    <p><? echo $row[price_discript_2_1]; ?></p>
                    <p><? echo $row[price_discript_2_2]; ?></p>
                    <p><? echo $row[price_discript_2_3]; ?></p>
                </div>
            </div>
        </div>
    </div>

```

```

<?
    $res=mysql_query("SELECT * FROM clients");
    while ($row=mysql_fetch_array($res)) {

    ?>

        <h2><? echo $row[name]; ?></h2>
        <p><? echo $row[slog]; ?></p>
        <img src=""<? echo $row[logo_1]; ?>>
        <img src=""<? echo $row[logo2]; ?>>
        <img src=""<? echo $row[logo_3]; ?>>
        <img src=""<? echo $row[logo_4]; ?>>
    </div>
</div>

<div id="clietns_say_wrap">
    <div id="clients_say">

<?

    $res=mysql_query("SELECT * FROM review");
    while ($row=mysql_fetch_array($res)) {

    ?>

        <h2><? echo $row[name]; ?></h2>
        <p><? echo $row[comment]; ?></p>
        
        <span> <? echo $row[url]; ?></span>
    </div>
</div>

</section>

```



JSON – это формат данных, как и XML, очень часто в веб-используется для обмена различного рода информацией которая отличается определенной структурой, особенно актуально если речь идет о базе данных, так как база данных всегда ведь представлена таблицами. По сути своей, JSON - форма описания объекта, массива объектов и их атрибутов. JSON используют чтоб формировать ответы от сервера. В моем случае, делается выборка из базы данных, которая пакуется самим сервером в JSON. Созданный файл `comment_control.php` получает имя комментатора и сам комментарий и добавляет этот комментарий в базу данных. Затем, делает выборку последних трех комментариев, пакует все в JSON и высвечивает в качестве ответа. В свою очередь, AJAX – это javascript, который всегда выполняется на стороне клиента браузером, то есть браузер шлет асинхронный запрос на сервер о том, что он желает добавить новый комментарий, а сервер возвращает ему список из трех последних комментариев, которые были добавлены. Причем, самый свежий будет его комментарий, потому что браузер клиента берет и вставляет все это в специальный div-блок с помощью `append()` – всё это происходит в function `sendcomment()`. POST – это http запрос, в отличии от get-запроса, параметры не в самом адресе зашифрованы, а в теле запроса. Я шлю запрос с двумя параметрами `text` и `name`, сервер получает пост-запрос, добавляет в базу соответствующую запись, а потом делает выборку.

```
<script>
function sendcomment(){
    var n_field = $('#name_comment').val();
    var t_field = $('#text_comment').val();
$.ajax({
    type: 'POST',
    url: 'comment_control.php',
    data: { name: n_field, text: t_field },
    dataType: 'json',
    success: function (data) {
        $('#comments').html('');
        $.each(data, function(index, element) {
            $('#comments').append(element.name+": "+element.text+"<br>");
        });
        // $('#comments').html(data.text+" - "+data.name);
    }
});
return false;
}
```

```
</script>
<div id="form">
    <h2><? echo $row[name_2]; ?></h2>
    <p><? echo $row[slog_2]; ?></p>

    <form action="" method="POST">
        <input id="name_comment" type="text" placeholder="Your Name" name="name" value="">
        <textarea id="text_comment" placeholder="Write your Message Here" name="message"></textarea>
        <div id="form_but">

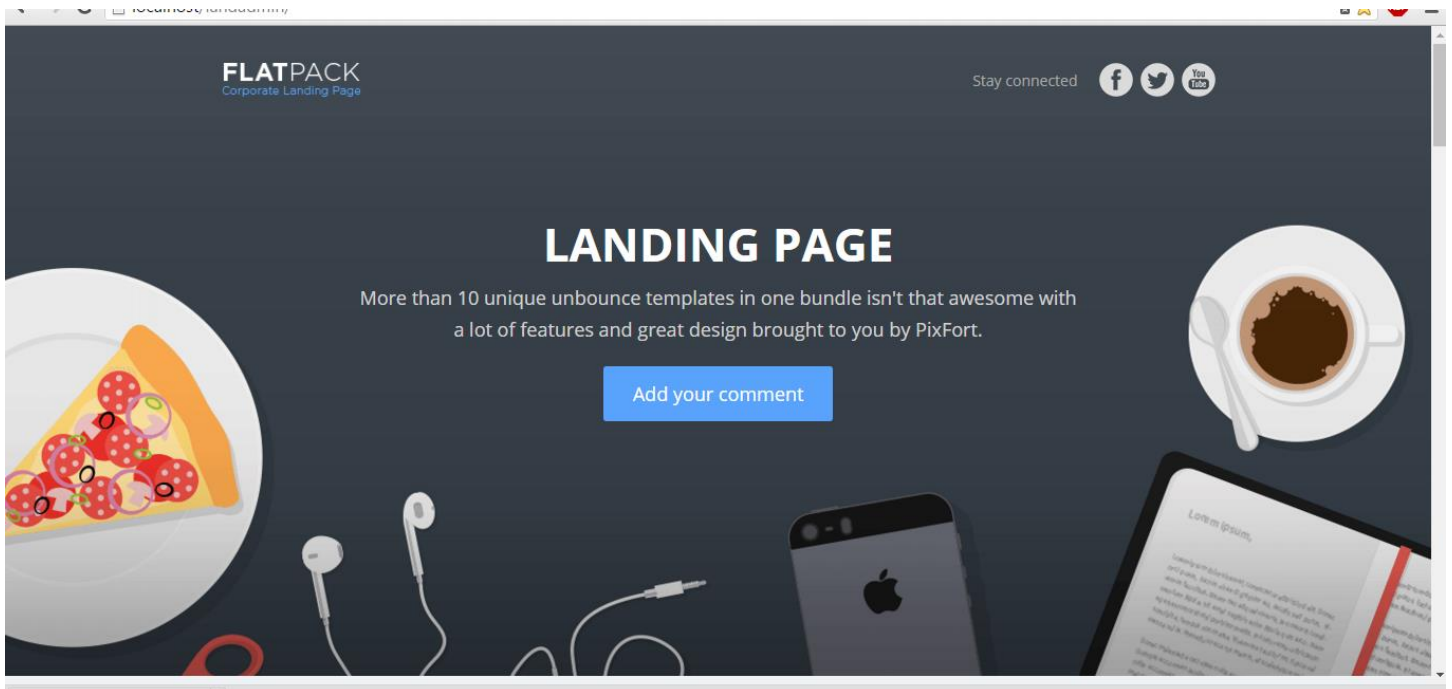
            <input type="button" value="Send it right away" class="form_but_send" onclick="sendcomment()">
        </div>
    </form>
</div>
<div id="comments" style="height:400px; color:white;">
    <?
    $result = mysql_query("SELECT * from `comments` ORDER BY id DESC LIMIT 3") or die(mysql_error());
    $return_arr = array();
    while ($row = mysql_fetch_array($result)){
        echo $row['name'].': '.$row['text'].'<br>';
    }
    ?>
</div>
</div>
```

```

<?php
include('db_connect.php');
mysql_query("INSERT INTO `comments` (`name`,`text`) values ('".$_POST['name']."','".$_POST['text']."')") or die(mysql_error());
$result = mysql_query("SELECT * from `comments` ORDER BY id DESC LIMIT 3") or die(mysql_error());
$return_arr = array();
while ($row = mysql_fetch_array($result)){
    $row_array['text'] = $row['text'];
    $row_array['name'] = $row['name'];
    array_push($return_arr,$row_array);
}
echo json_encode($return_arr);
//echo '{"name":"jjj","text":"bbb"}';
?>

```

4 Результатом работы является Landing Page. Целевая страница (англ. landing page, также «посадочная страница») — веб-страница, построенная определенным образом, основной задачей которой является сбор контактных данных целевой аудитории. Используется для усиления эффективности рекламы, увеличения аудитории. Целевая страница обычно содержит информацию о товаре или услуге.





### Calculator

The edges may be directed or undirected For example if the vertice represent people party.



### Daily mails

The edges may be directed or undirected For example if the vertice represent people party.



### Coffee cup

The edges may be directed or undirected For example if the vertice represent people party.



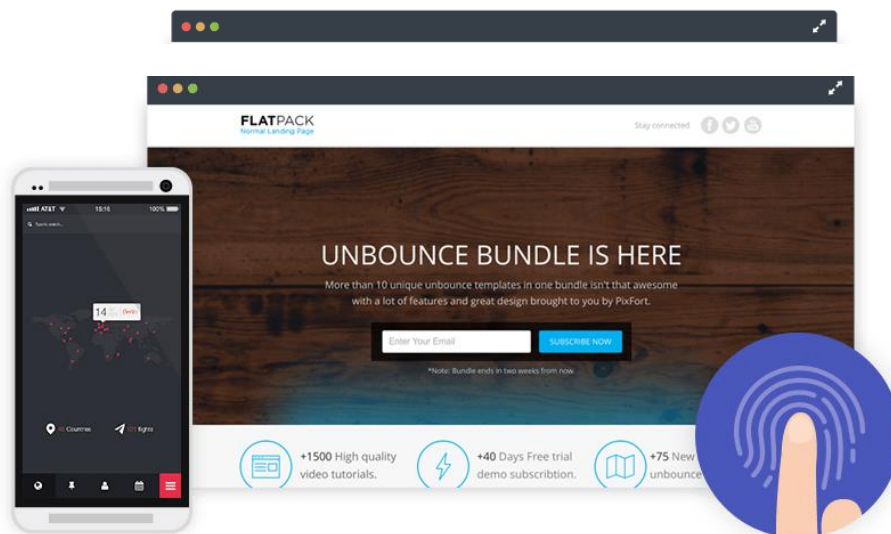
### Connection

The edges may be directed or undirected For example if the vertice represent people party.

## The most amazing thing is here

Our service is astonishingly thin and light.

Lorem ipsum dolor sit amet consectetur adipiscing elit sed do eiusmod tempor incididunt ut labore et dolore magna aliqua Ut enim ad minim veniam quis nostrud.



## Great people trusted our services

great words from great people

## Great people trusted our services

great words from great people

**+950**

Themes and  
Templates Sales

**+150**

Followers love  
and trust u

**+85**

Items selling on  
themeforest

**+6K**

Working hours  
this year wow

## Our Services Prices and Plans

Our service is totally thin and light.

Get All The Features

### Free, Free

Get All The Freebies

5 Projects

1 GB Storage

### Personal

Get All The Features

25 projects

50 GB Storage

### Company

Get All The Features

100 Projects

250 GB Storage

1 GB Storage

No Domain

1 User

0\$/m

Subscribe

50 GB Storage

With 1 Domain

10 User

9,99\$/m

Subscribe

250 GB Storage

Unlimited Domains

50 Users

70\$/m

Subscribe

## Companies We Work With

great word from great people

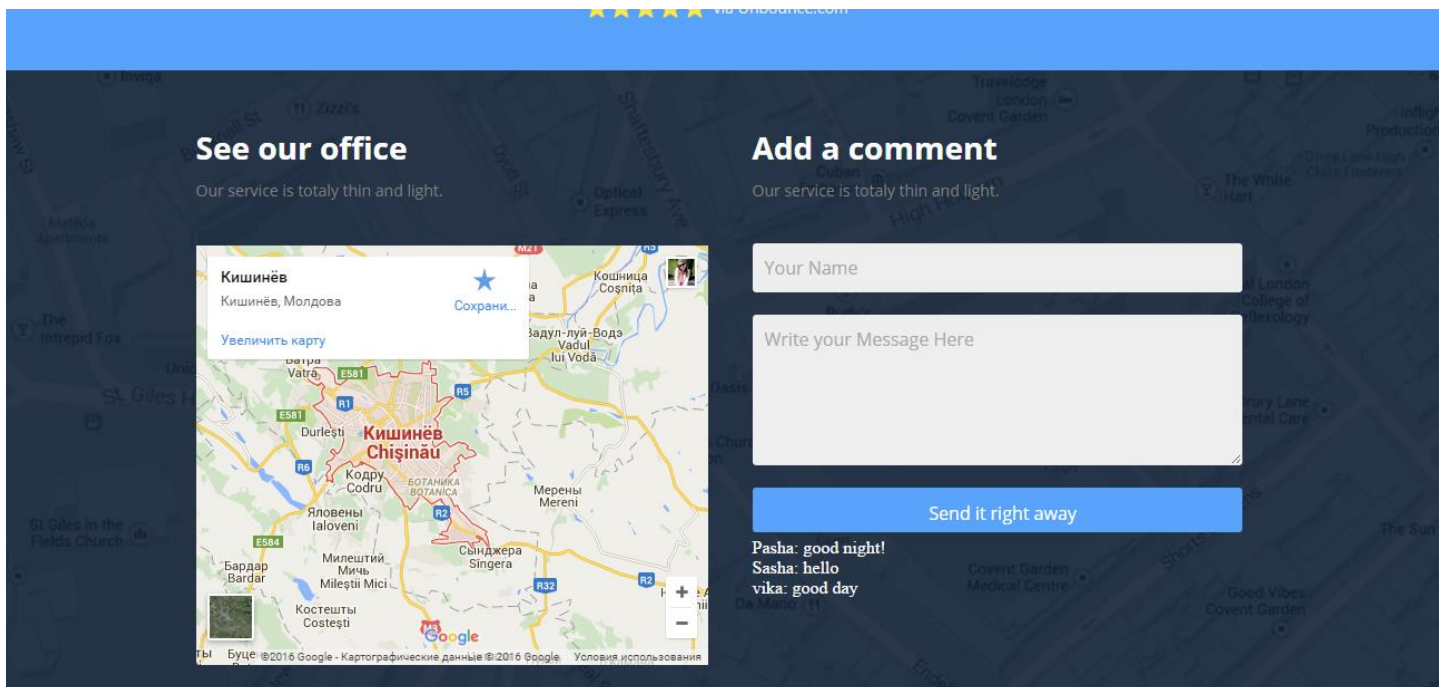
LOGO

LOGO

LOGO

LOGO

What our awesome clients say



## Выводы

В результате выполнения данной лабораторной работы была реализована целевая страница(landing page), с помощью html, php, javascript и MySQL. Для этого был использован Sublime Text. Знания, полученные в результате выполнения работы пригодятся в дальнейшей работе.

## Библиография

<http://www.denwer.ru/> - официальный сайт Денвера и инструкция по его установке и использованию

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