Experiment 11

Running PHP : simple applications like login forms after setting up a LAMP stack

About LAMP

LAMP is an archetypal model of web service stacks, named as an acronym of the names of its original four open-source components: the Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language. The LAMP components are largely interchangeable and not limited to the original selection. As a solution stack, LAMP is suitable for building dynamic web sites and web applications.

Since its creation, the LAMP model has been adapted to other components, though typically consisting of free and open-source software.

Set Up

Before running through the steps, make sure that all repositories are up to date:

```
apt-get update
apt-get upgrade
```

With that completed, go ahead and start installing the LAMP server.

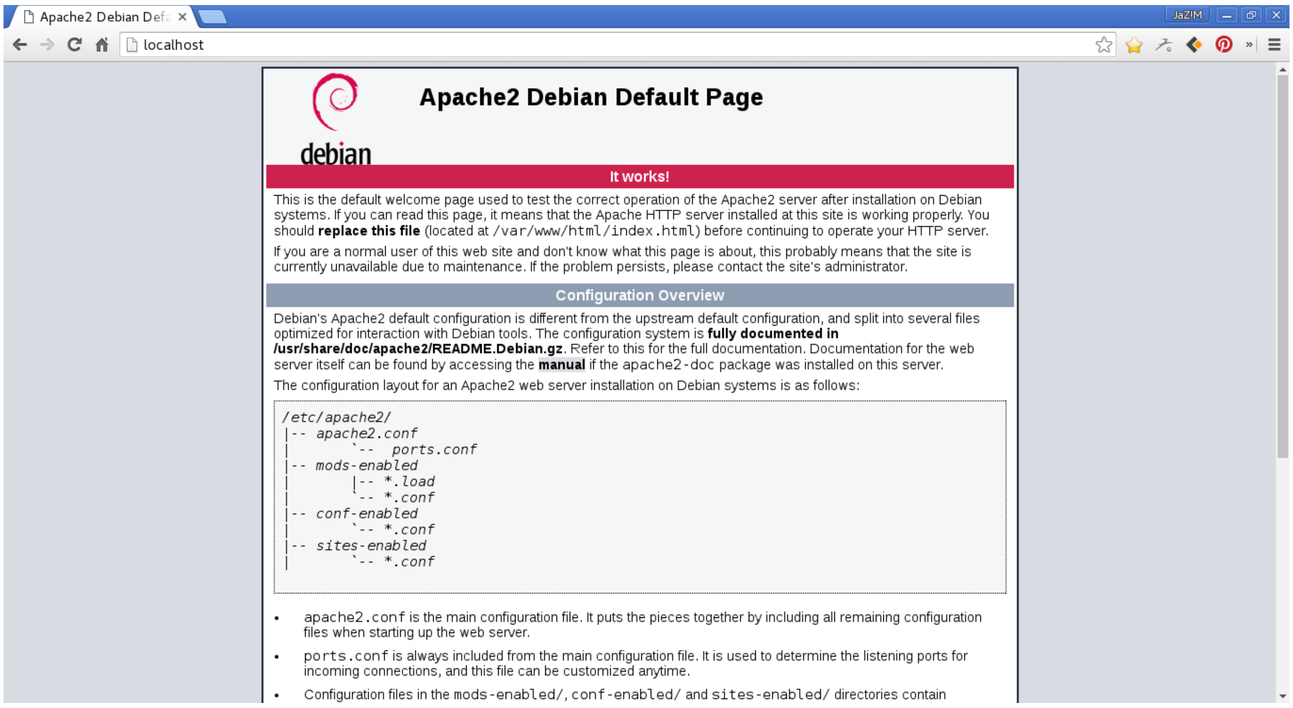
Step One - Install Apache

Apache is a free open source software which runs over 50% of the world’s web servers.

To install apache, open terminal and type in these commands:

```
apt-get install apache2
```

To check if Apache is installed on your VPS, direct your browser to your server’s IP address (e.g. http:// your_server_ip_address or http://localhost). The page should display the words “It works!".



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain

How to Find your Server's IP address

You can run the following command to reveal your VPS's IP address.

```
ifconfig eth0 | grep inet | awk '{ print $2 }'
```

Step Two - Install MySQL

MySQL is a widely-deployed database management system used for organizing and retrieving data.

To install MySQL, open terminal and type in these commands:

```
apt-get install mysql-server
```

During the installation, MySQL will ask you to set a root password. If you miss the chance to set the password while the program is installing, it is very easy to set the password later from within the MySQL shell.

Finish up by running the MySQL set up script:

```
mysql_secure_installation
```

The prompt will ask you for your current root password.

Type it in.

Then the prompt will ask you if you want to change the root password. Go ahead and choose N and move on to the next steps.

It's easiest just to say Yes to all the options. At the end, MySQL will reload and implement the new changes.

Once you're done with that you can finish up by installing PHP on your virtual server.

Step Three—Install PHP

PHP is an open source web scripting language that is widely use to build dynamic webpages.

To install PHP, open terminal and type in this command.

```
apt-get install php5 php-pear php5-mysql
```

After you answer yes to the prompt twice, PHP will install itself.

Finish up by restarting apache:

```
service apache2 restart
```

Congratulations! You now have LAMP stack on your droplet!

Step Four—RESULTS: See PHP on your Server

Although LAMP is installed, we can still take a look and see the components online by creating a quick php info page

To set this up, first create a new file:

```
cd  /var/www/html
sudo touch info.php
sudo chown www-data: www-data info.php
sudo nano info.php
```

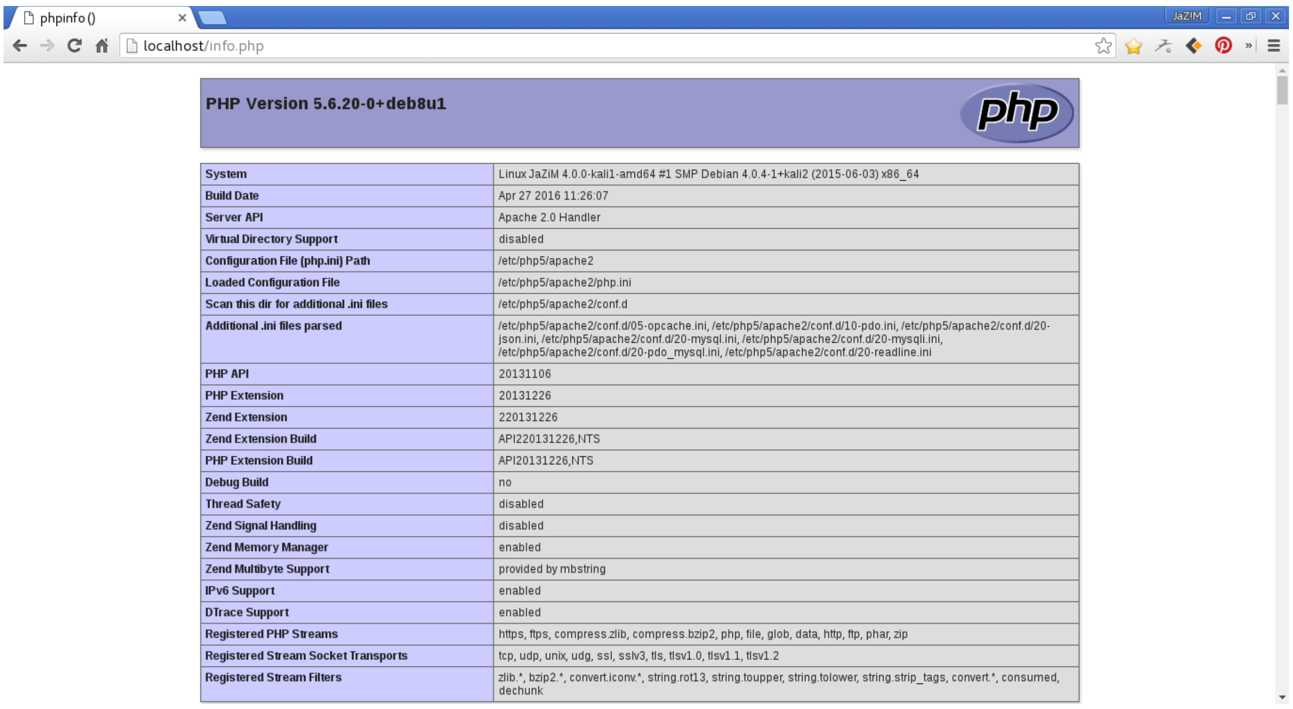
Add in the following line:

```
<?php
        phpinfo();
?>
```

Then Save and Exit.

Finish up by visiting your php info page (make sure you replace the example ip address with your correct one): http:// your_server_ip_address or http://localhost/info.php

It should look something like this:



PHP Version 5.6.20-0+deb8u1

System	Linux JaZiM 4.0.0-kali1-amd64 #1 SMP Debian 4.0.4-1+kali2 (2015-06-03) x86_64
Build Date	Apr 27 2016 11:26:07
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

Step Five—Install PHPMyAdmin

Additionally, you should install phpMyAdmin so you can administer your databases easier via the phpMyAdmin intuitive GUI.

Enter the following command:

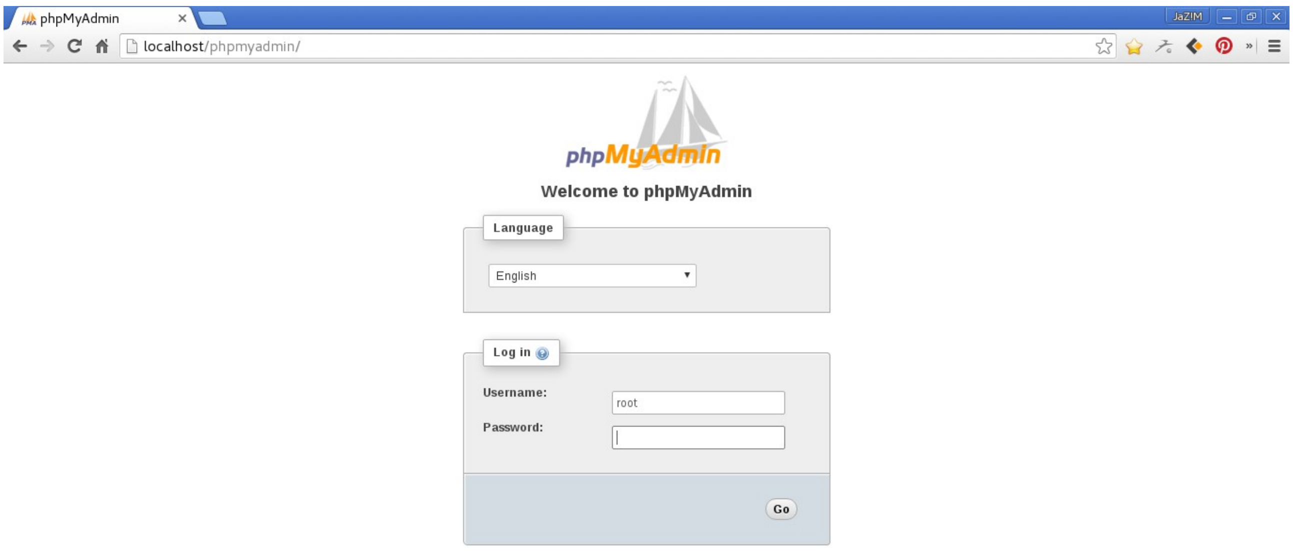
```
apt-get install phpmyadmin
```

During the installation procedure you will be prompted with several windows for configuring phpMyAdmin. You should select ‘Yes‘ when you get to the ‘Configure database for phpMyAdmin with dbconfig-common’ step after which a database will be installed and configured for phpMyAdmin.

Then, enter your MySQL root password as the ‘Password of the database’s administrative user’. Next, enter a password of your choice for the ‘MySQL application password for phpmyadmin’.

After this is finished, you will be prompted with a window where you can select a web server to be configured to run phpMyAdmin. Select apache2 and then select ok.

Once the installation is completed, you will be able to access phpMyAdmin by navigating your web browser to: http://your_server_ip_address/phpmyadmin. You will be welcomed by the underneath page:



Once this is done, required softwares are installed. You can either login to phpmyadmin and then create the databases and tables or by using the mysql commands from terminal. For running mysql commands from terminal, open terminal and then type in:

```
mysql -u root -p
```

and then enter the password when prompted.

Once databases and tables are created the login form application can be created using php.

Steps to creating Simple User Registration and Login Script in PHP and MySql

- •Create a database
- •Creating table with five fields
  - •id
  - •username
  - •email
  - •password
  - •active
- •Create form in html
- •Adding styles to form
- •Connect to the database using php
- •Writing logic for user registration, login, logout scripts in php

1. Create a Database and Database Table

For creating a database, login to phpmyadmin, then click on Databases tab. And in the input field enter database name and click on create database button.

After creating Database, Create a database table for users, it’s simple table with user name, email, password, active and an id as primary key and also it is auto incremented.

- •id is the primary key and auto incremented.
- •Username field will be used for storing username

•email field will be used for storing email address of user,
•Password field will store password of user in normal format.
•active state is for checking the user is active or blocked user, If the user is blocked then he can't login to system.

Import below sql into your database.

```
CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `active` tinyint(1) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
)
```

2. Create form in html and writing logic for user registration script in php

All the registration process done in one single php file, in this file we will display an html form for user to register. In the PHP section we will first connect to the database, then we will select the database, if the values are posted, then we are running a MySQL query it will insert the values into the database.
Save this file as **register.php,** Here this is simple html for generating form. This form contains user name field, email, password a submit button and a link to login.php file.
Code:

```
<?php
  require('connect.php');
   if (isset($_POST['username']) && isset($_POST['password'])){
      $username = $_POST['username'];
                                          $email = $_POST['email'];
      $password = $_POST['password'];
      $query = "INSERT INTO `user` (username, password, email) VALUES ('$username',
'$password', '$email')";
      $result = mysqli_query($connection, $query);
      if($result){
          $smsg = "User Created Successfully.";
      }else{
          $fmsg ="User Registration Failed";
      }
   }
   ?>
```

```
<html>
<head>
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" >
<link rel="stylesheet" href="styles.css" >
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
   <form class="form-signin" method="POST">
                    <?php if(isset($smsg)){ ?><div class="alert alert-success"
role="alert"> <?php echo $smsg; ?> </div><?php } ?>
                    <?php if(isset($fmsg)){ ?><div class="alert alert-danger"
role="alert"> <?php echo $fmsg; ?> </div><?php } ?>
    <h2 class="form-signin-heading">Please Register</h2>
    <div class="input-group">
       <span class="input-group-addon" id="basic-addon1">@</span>
       <input type="text" name="username" class="form-control" placeholder="Username"
required>
     </div>
    <label for="inputEmail" class="sr-only">Email address</label>
    <input type="email" name="email" id="inputEmail" class="form-control"
placeholder="Email address" required autofocus>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" name="password" id="inputPassword" class="form-control"
placeholder="Password" required>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Register</button>
    <a class="btn btn-lg btn-primary btn-block" href="login.php">Login</a>
   </form>
</div>
</body>
</html>
```

3. Connect to the database

Create a file with name connect.php. Its for connecting to database and selecting the database. Use correct database name & user credentials.

Code:

```
<?php
$connection = mysqli_connect('localhost', 'root', '12345');
if (!$connection){
   die("Database Connection Failed" . mysqli_error($connection));
}
$select_db = mysqli_select_db($connection, 'database');
if (!$select_db){
   die("Database Selection Failed" . mysqli_error($connection));
}?>
```

4. PHP Logic for User Login

login.php is having information about php script and HTML script to do login.

Code:

```
<?php  //Start the Session
session_start();
require('connect.php');
if (isset($_POST['username']) and isset($_POST['password'])){
$username = $_POST['username'];
$password = $_POST['password'];
$query = "SELECT * FROM `user` WHERE username='$username' and
password='$password'";
$result = mysqli_query($connection, $query) or die(mysqli_error($connection));
$count = mysqli_num_rows($result);
if ($count == 1){
$_SESSION['username'] = $username;
}else{
$fmsg = "Invalid Login Credentials.";
}
}
if (isset($_SESSION['username'])){
```

```
$username = $_SESSION['username'];
header("location:welcome.php");
}else{}
?>
<html>
<head>
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet" 
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
  <!-- Optional theme -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" >
  <link rel="stylesheet" href="styles.css" >
  <!-- Latest compiled and minified JavaScript -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<form class="form-signin" method="POST">
  <h2 class="form-signin-heading">Please Login</h2>
  <div class="input-group">
<span class="input-group-addon" id="basic-addon1">@</span>
<input type="text" name="username" class="form-control" placeholder="Username" 
required>
</div>
  <label for="inputPassword" class="sr-only">Password</label>
  <input type="password" name="password" id="inputPassword" class="form-control" 
placeholder="Password" required>
  <button class="btn btn-lg btn-primary btn-block" type="submit">Login</button>
  <a class="btn btn-lg btn-primary btn-block" href="register.php">Register</a>
</form>
</html>
```

5. PHP Logic for User Welcome Page

After successful login, it will display welcome page.

Code:welcome.php

```
<?php
session_start();
```

```
?>
<html>
<head>
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" >
<link rel="stylesheet" href="styles.css" >
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
<h1>Welcome <?php echo $_SESSION['username']; ?></h1>
Login Successful
<a class="btn btn-lg btn-primary btn-block" href="logout.php">Logout</a>
</body>
</html>
```

6. PHP Logic for User Logging out

Logout page is having information about how to logout from login session.

<u>Code:logout.php</u>

```
<?php
session_start();
session_destroy();
header('Location: login.php');
?>
```

7. HTML Styling

Included to control the layout of multiple web pages all at once.

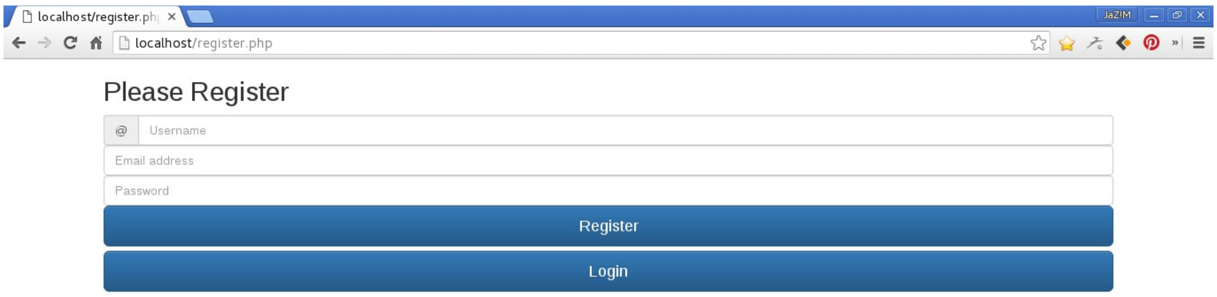
<u>Code:style.css</u>

```
body {
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #eee;
}

.form-signin {
  max-width: 330px;
  padding: 15px;
  margin: 0 auto;
}
.form-signin .form-signin-heading,
.form-signin .checkbox {
  margin-bottom: 10px;
}
.form-signin .checkbox {
  font-weight: normal;
}
.form-signin .form-control {
  position: relative;
  height: auto;
  -webkit-box-sizing: border-box;
     -moz-box-sizing: border-box;
          box-sizing: border-box;
  padding: 10px;
  font-size: 16px;
}
.form-signin .form-control:focus {
  z-index: 2;
}
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;}
```

Results:

1: Registration page



Please Register

@ Username

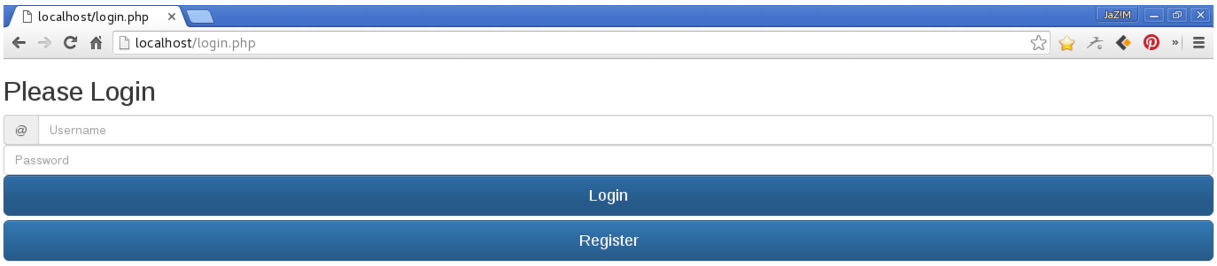
Email address

Password

Register

Login

2: Login page



Please Login

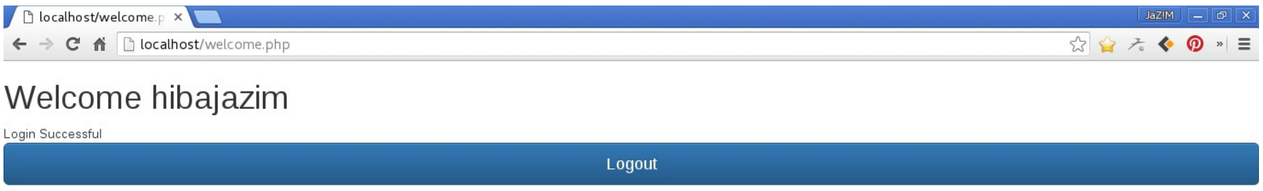
@ Username

Password

Login

Register

3. Welcome Page



Welcome hibajazim

Login Successful

Logout