A detailed report on

SECURE APPLICATION DEVELOPMENT

CPS 592-07

Final Project

By

Dr.Phu phung

Submitted by

Venkateshwarlu Komuravelly

ID: 1015237060

Email: komuravellyv1@udayton.edu

Submitted Date: May 2, 2018

Link to my Bitbucket

https://bitbucket.org/komuravellyv1/venky2018sec-private/src/master/assignments/project/

INTRODUCTION:

This project mainly aims in applying security principles and practices and web-development technologies that learned in the class CPS 592-07. This is through by creating a blog/social media application by using PHP Language, Ubuntu OS, and MySQL Database. When developing the application, I took care of the problems or attacks by thinking like a hacker. There are some chances that attackers can steal SESSION variables used in application which result to vulnerability of the application. Which are taken care by security principles learned. Also, to prevent SQL Injection attacks Prepared statements are used when interacting with Database. Also, I took care of Database security by creating and granting the access to user. All passwords are hashed, and all input and outputs are sanitized.

Development:

This is a blog with static web pages and relational database in the backend. On the Index page [main page] it consists of all the posts posted by users registered. Anyone can comment on these posts means no need to have an account to post a comment on the posts.

Users can register into the application by providing mandatory details {name, username, password, email, phone}. Once registered they need to be approved by the super user [admin]. Without approval users cannot login to the system. If admin approves and enables the account user can login to the system. After successful login users can create, edit, delete their own posts. They cannot do anything to other posts in the application. Also, Users can change their all details except username. Every change, update or action is securely programmed.

Admins are created directly in the database and can login to the system with the details. Once logged, he can see all the users in the system. Admin can approve, disapprove, enable, disable the users. Also, admin can change his password securely. Admin has no privilege on posts created by users.

Achievement: By doing this project I got knowledge of Security principles that are applied when developing web applications. Also, other security issues due to flaws in programming languages and network intruders to steal data on network are learned and implemented security principles in my project by keeping in mind to avoid them.

At end, I can create Robust and defensive applications by learning and practicing security principles.

2. DESIGN:

DATABASE DESIGN:

For this project I used MySQL Database on Ubuntu OS and Tomcat Server. I have created a database [**venky_secad_project**] and developed my tables inside. For this I mainly created 4 tables namely users→ for Admins, rusers→ for Regular users, posts→ for posts, comments→ for comments. Also, Importantly I Created a User and granted privileges to the user instead of interacting with Database directly. Which is a security principle learned in the class {Never ever interact with the database directly. Instead create user and grant permissions}. For this project I created user[**spsecad**] Identified by **Aruna@03**. Below is the details of Database name, users created, and tables.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test               |
| venky_secad_project |
+--------------------+
```

```
mysql> use venky_secad_project;
Reading table information for com
You can turn off this feature to

Database changed
mysql> show tables;
+-------------------------------+
| Tables_in_venky_secad_project |
+-------------------------------+
| comments                      |
| posts                         |
| rusers                        |
| users                         |
+-------------------------------+
```

```
mysql> describe users;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| username | varchar(50)  | NO   | PRI | NULL    |       |
| password | varchar(255) | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

```
mysql> describe rusers;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| name     | varchar(255) | NO   |     | NULL    |       |
| username | varchar(25)  | NO   | PRI | NULL    |       |
| email    | varchar(50)  | NO   |     | NULL    |       |
| password | varchar(255) | NO   |     | NULL    |       |
| phone    | varchar(15)  | NO   |     | NULL    |       |
| approval | int(1)       | NO   |     | NULL    |       |
| enable   | int(1)       | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

```
mysql> describe posts;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| postid    | int(11)      | NO   | PRI | NULL    | auto_increment |
| title     | varchar(255) | NO   |     | NULL    |                |
| text      | text         | NO   |     | NULL    |                |
| published | datetime     | YES  |     | NULL    |                |
| owner     | varchar(25)  | NO   | MUL | NULL    |                |
| enable    | int(1)       | NO   |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
```

```
mysql> describe comments;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| commentid | int(11)      | NO   | PRI | NULL    | auto_increment |
| title     | varchar(255) | NO   |     | NULL    |                |
| content   | text         | NO   |     | NULL    |                |
| time      | datetime     | YES  |     | NULL    |                |
| commenter | varchar(50)  | YES  |     | NULL    |                |
| postid    | int(11)      | YES  | MUL | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
```

```
DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
`username` VARCHAR(50) PRIMARY KEY,
`password` VARCHAR(255) NOT NULL
);
DROP TABLE IF EXISTS `rusers`;
CREATE TABLE `rusers` (
  `name` VARCHAR(255) NOT NULL,
  `username` VARCHAR(25) NOT NULL PRIMARY KEY,
  `email` VARCHAR(50) NOT NULL,
  `password` VARCHAR(255) NOT NULL,
  `phone` VARCHAR(15) NOT NULL,
  `approval` int(1) NOT NULL,
  `enable` int(1) NOT NULL
);
DROP TABLE IF EXISTS `posts`;
CREATE TABLE `posts` (
  `postid` int(11) AUTO_INCREMENT PRIMARY KEY,
  `title` VARCHAR(255) NOT NULL,
  `text` text NOT NULL,
  `published` datetime DEFAULT NULL,
  `owner` VARCHAR(50),
  `enable` int(1) NOT NULL,
  FOREIGN KEY (`owner`) REFERENCES `rusers` (`username`) ON DELETE CASCADE
);
```

```
DROP TABLE IF EXISTS `comments`;
CREATE TABLE `comments` (
  `commentid` int(11) AUTO_INCREMENT PRIMARY KEY,
  `title` VARCHAR(255) NOT NULL,
  `content` text NOT NULL,
  `time` datetime DEFAULT NULL,
  `commenter` VARCHAR(50) DEFAULT NULL,
  `postid` int(11) DEFAULT NULL,
  FOREIGN KEY (`postid`) REFERENCES `posts` (`postid`) ON DELETE CASCADE
);
```

USER INTERFACE:

For this project I mainly used **Bootstrap 4.3, Font Awesome cdns and CSS.** From the Bootstrap I developed Navbar feature for the options to select in the system where ever possible. I used background image in the header file to reflect on all pages.

FUNCTIONALITIES:

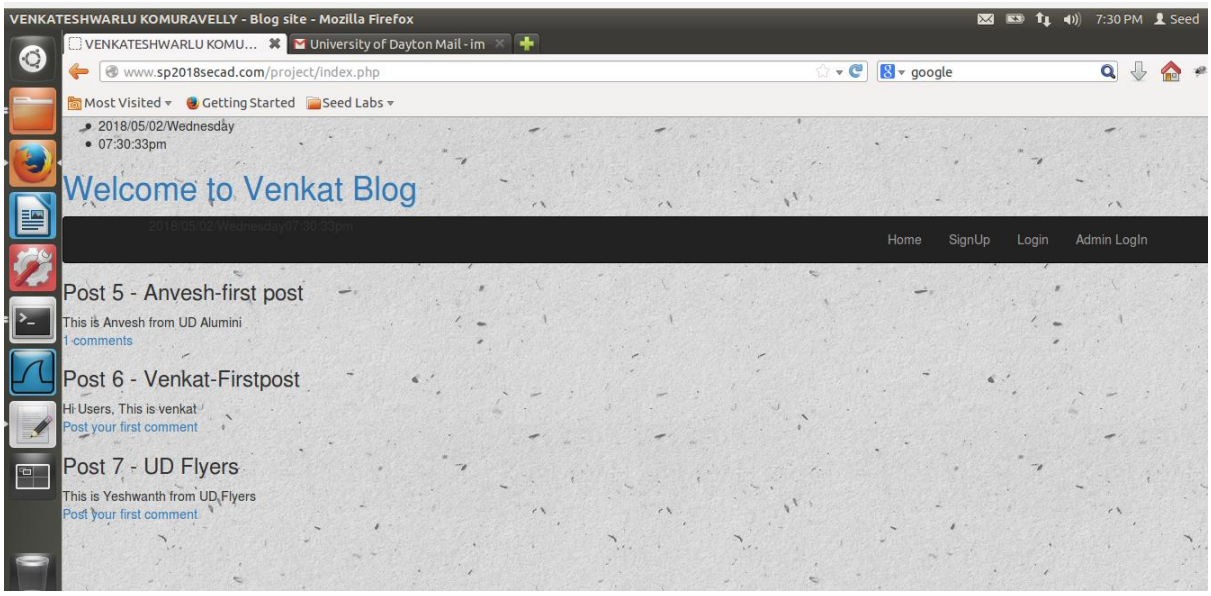
Guests can see all the posts and comment on them. They are not allowed to do anything on the system [creating a post, deleting a post, editing a post].

Guests becomes regular users after successful registration. For successful active members users registered should be approved and enabled by the admin. This can be done in the system by creating two columns in the rusers table [approval, enable]. Upon registration these two values are set to 0. And to login to the system approval and enable fields should be set to 1. The admins can do this.

Admins are created directly into the database. I have used two separate tables one is for admins and another for regular users. In both the tables username is the primary key and it should be unique. Regular users can not login as admins as the details are stored in separate tables. Admins cannot enter as regular users as they can not see the regular users password [its hashed]. Its completely secured.

IMPLEMENTATION:

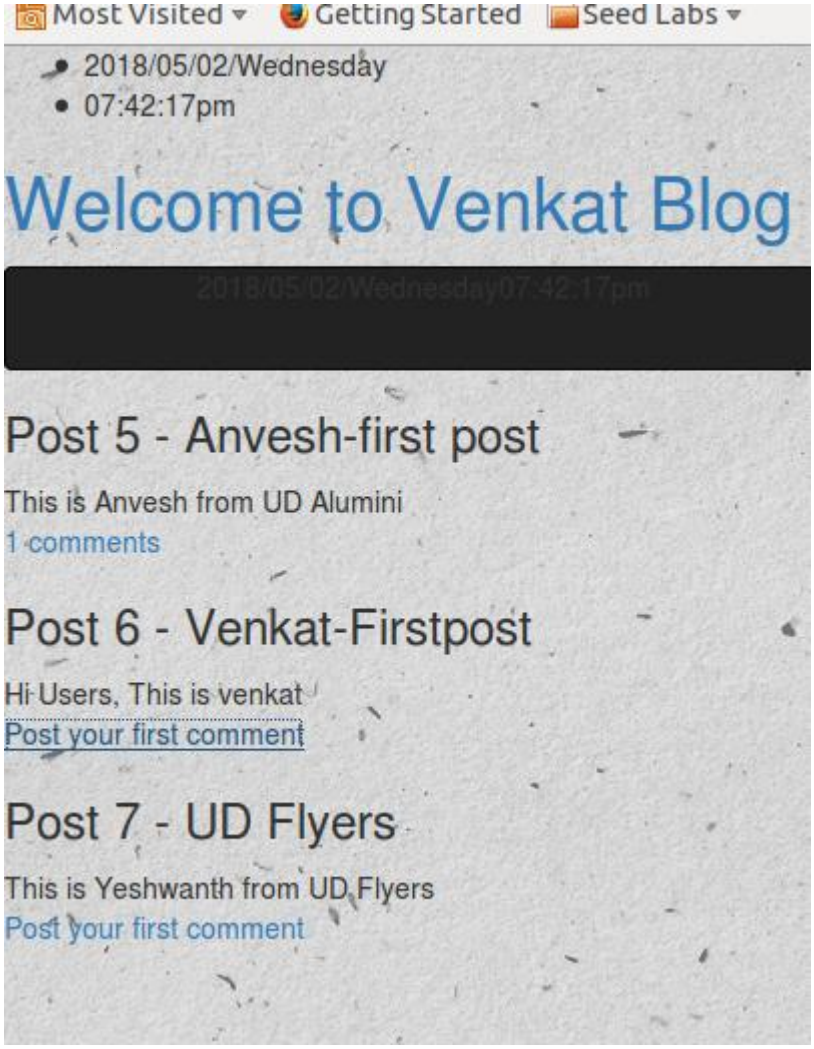
(5 points) Everyone can view all enabled posts



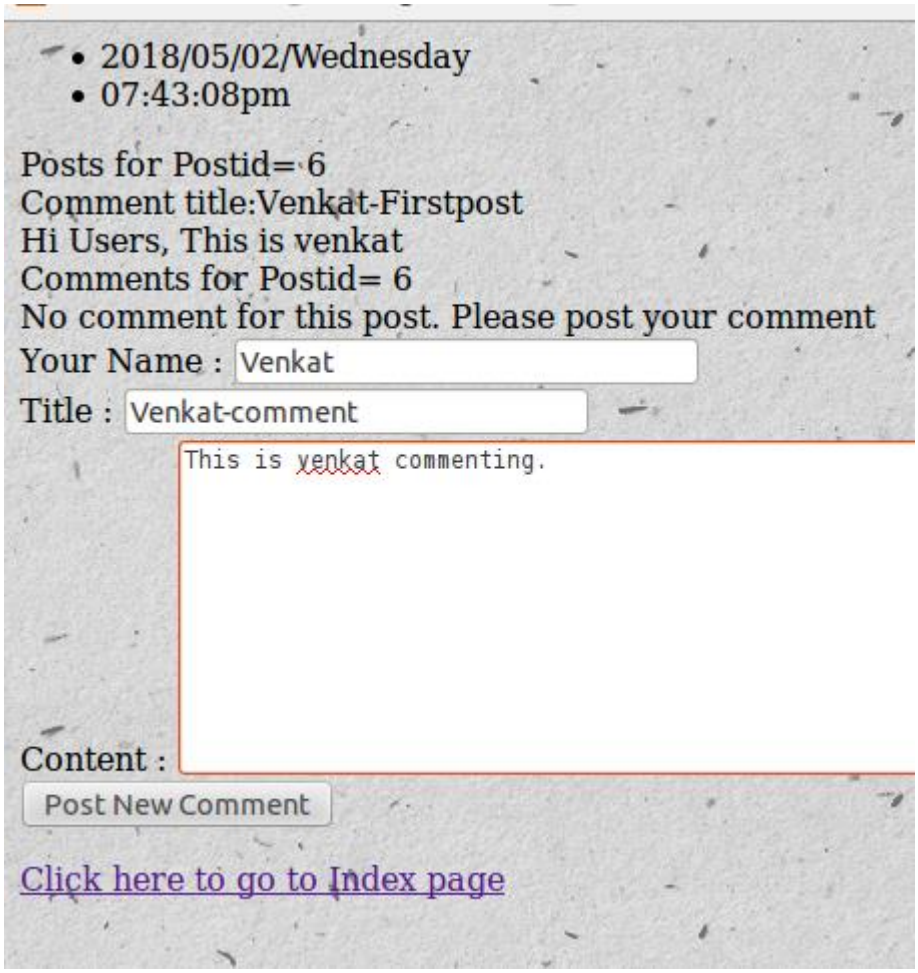

```php
function show_posts(){
    global $mysqli;
    $sql = "SELECT * FROM posts WHERE enable=1";
    $result = $mysqli->query($sql);
    if($result->num_rows> 0) {
      while($row = $result->fetch_assoc()) {
          $postid = $row["postid"];
          echo "<h3>Post " . $postid . " - " .$row["title"]. "</h3>";
          echo $row["text"] . "<br>";
          echo "<a href='comment.php?postid=$postid'>";
          $sql = "SELECT * FROM comments WHERE postid='$postid';";
          $comments = $mysqli->query($sql);
          if($comments->num_rows>0){
             echo $comments->num_rows . " comments </a>";
          }else{
             echo "Post your first comment </a>";
          }
       }
     }
    else{ echo "No post in this blog yet <br>";}
}
```

The above code is reason for all the posts to visible on index page. In the index.php by calling this function makes every post in the posts table created by all users is shown. However, only posts with enable=1 can be visible. This feature can be modified by the users to their own posts. As, postid is primary key and its unique. So selecting all the rows with enable =1; and writing the code in above format makes this possible.

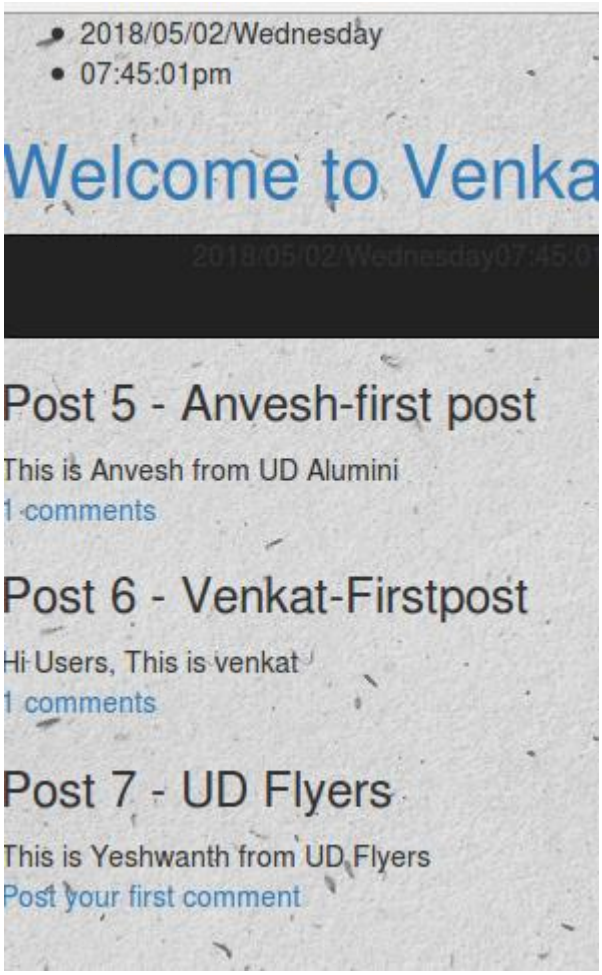
(5 points) Everyone can comment on available posts



Anyone can comment on these posts. No need of account.


2018/05/02/Wednesday
07:43:08pm
Posts for Postid= 6
Comment title:Venkat-Firstpost
Hi Users, This is venkat
Comments for Postid= 6
No comment for this post. Please post your comment
Your Name : Venkat
Title : Venkat-comment
This is venkat commenting.
Content :
Post New Comment
Click here to go to Index page


2018/05/02/Wednesday
07:44:41pm
New comment addedPosts for Postid= 6
Comment title:Venkat-Firstpost
Hi Users, This is venkat
Comments for Postid= 6
Comment title:Venkat-comment
This is venkat commenting.
Your Name :
Title :
Content :
Post New Comment
Click here to go to Index page

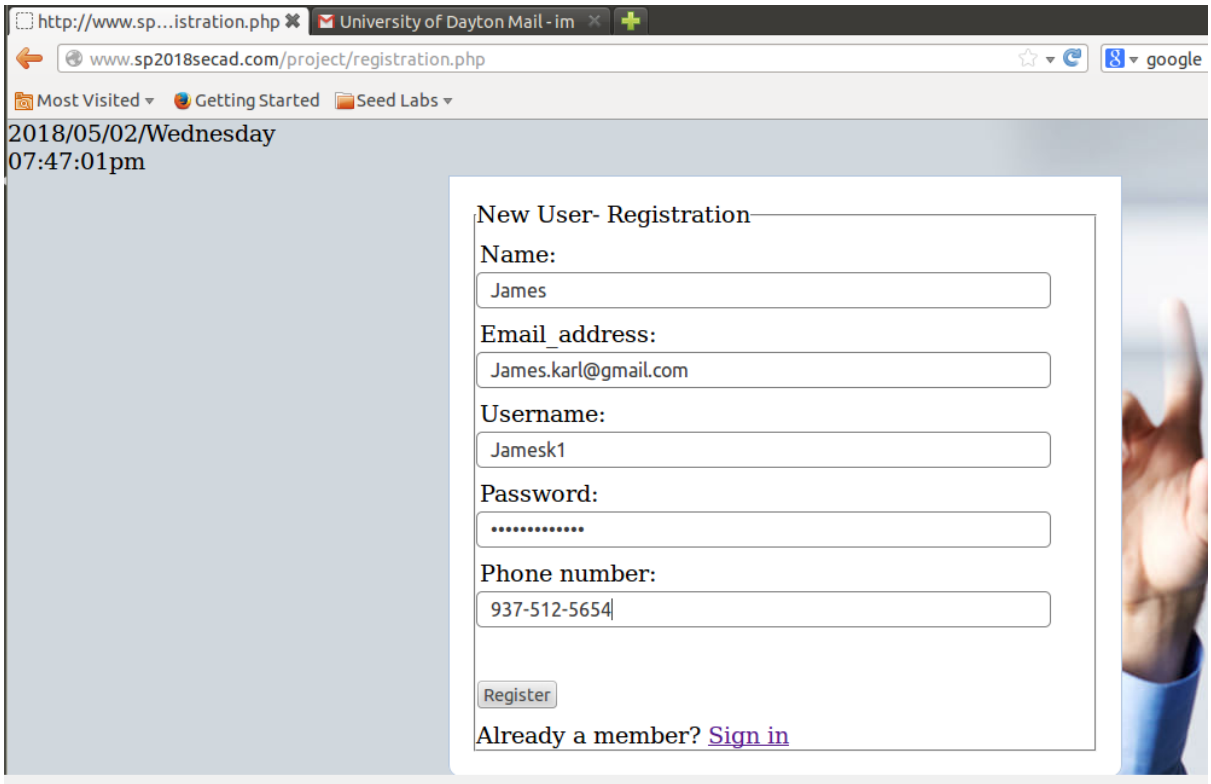

2018/05/02/Wednesday
07:45:01pm
Welcome to Venka
Post 5 - Anvesh-first post
This is Anvesh from UD Alumini
1 comments
Post 6 - Venkat-Firstpost
Hi Users, This is venkat
1 comments
Post 7 - UD Flyers
This is Yeshwanth from UD Flyers
Post your first comment

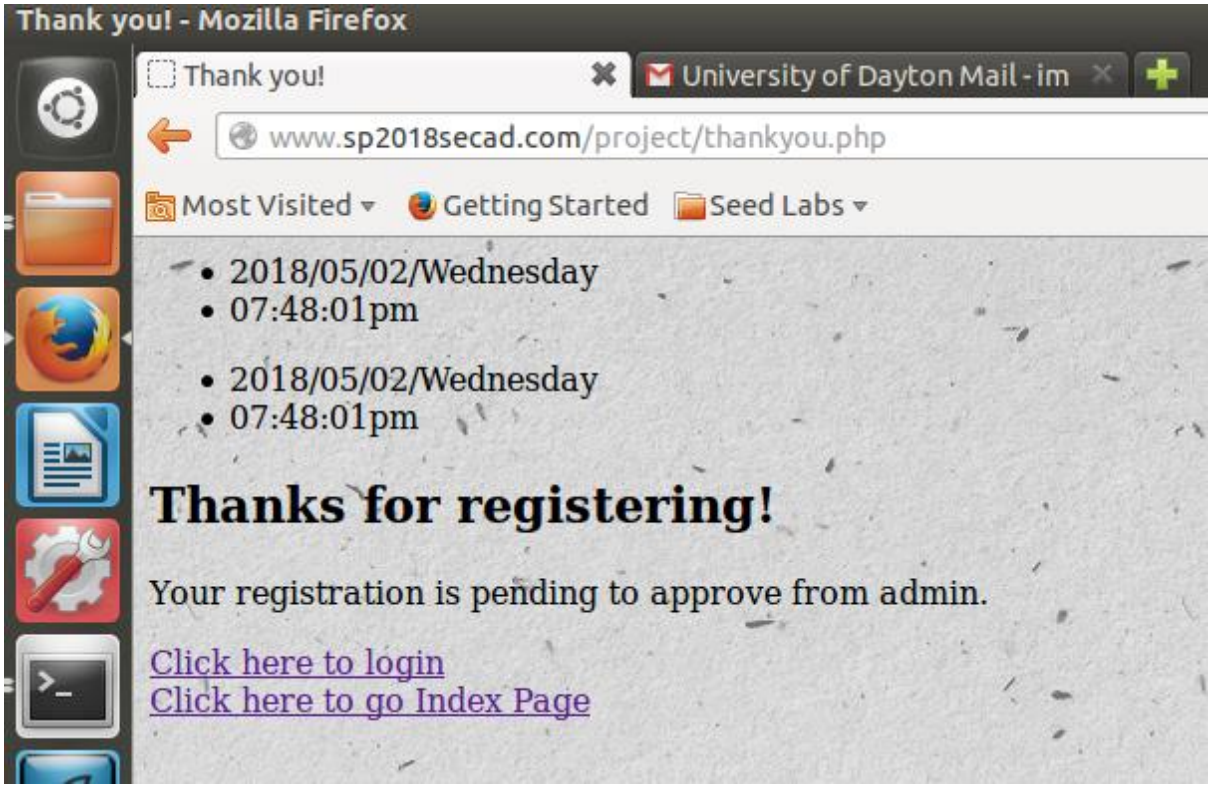
```php
<?php
  session_start();
  require 'mysql.php';
  include 'header.php';
  $postid = $_REQUEST['postid'];
  if(!isset($postid)){
    echo "Bad Request";
    die();
  }
  function handle_new_comment($postid){
    $title = $_POST['title'];
    $content = $_POST['content'];
    $commenter = $_POST['commenter'];
    $nocsrftoken = $_POST["nocsrftoken"];
    $sessionnocsrftoken = $_SESSION["nocsrftoken"];
    if (isset($title) and isset($content) ){
        if(!isset($nocsrftoken) or ($nocsrftoken!=$sessionnocsrftoken)){
        echo "Cross-site request forgery is detected!";
        die();
        }
    if(new_comment($postid,$title,$content,$commenter))
        echo "New comment added";
    else
        echo "Cannot add the comment";
    }
  }
  handle_new_comment($postid);
  display_singlepost($postid);
  display_comments($postid);
  $rand = bin2hex(openssl_random_pseudo_bytes(16));
  $_SESSION["nocsrftoken"] = $rand;
```

```php
function handle_new_comment($postid){
  $title = $_POST['title'];
  $content = $_POST['content'];
  $commenter = $_POST['commenter'];
  $nocsrftoken = $_POST["nocsrftoken"];
  $sessionnocsrftoken = $_SESSION["nocsrftoken"];
  if (isset($title) and isset($content) ){
      if(!isset($nocsrftoken) or ($nocsrftoken!=$sessionnocsrftoken)){
      echo "Cross-site request forgery is detected!";
      die();
      }
  if(new_comment($postid,$title,$content,$commenter))
      echo "New comment added";
  else
      echo "Cannot add the comment";
  }
}
handle_new_comment($postid);
display_singlepost($postid);
display_comments($postid);
$rand = bin2hex(openssl_random_pseudo_bytes(16));
$_SESSION["nocsrftoken"] = $rand;
?>
<form action="comment.php?postid=<?php echo $postid; ?>" method="POST" class="form login">
  <input type="hidden" name="nocsrftoken" value="<?php echo $rand; ?>" />
  Your Name : <input type="text" name="commenter" /><br>
  Title : <input type="text" name="title" required/><br>
  Content : <textarea name="content" required cols="100" rows="10"></textarea><br>
  <button class="button" type="submit">Post New Comment</button>
</form>
<a href="index.php"> Click here to go to Index page</a>
```

(5 points) Everyone can register an account.:

Below steps show this.






```
mysql> select * from rusers;
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| name   | username     | email                     | password                                  | phone        | approval | enable |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| James  | Jamesk1      | James.karl@gmail.com      | *ECFE4F64A77D7C65C97D7969576F78E06F590ACB | 937-512-5654 |        0 |      0 |
| anvesh | komuravellyv1 | sunilvadlamani@gmail.com  | *BE89E73620E5A4C4040705AE19A8C1FF1A823EE0 | 937-993-9676 |        0 |      0 |
| yesh   | moresy1      | yesh.yesh@gmail.com       | *3CEA034E0ED9543580C3ADF889CB4E860E49781B | 937-520-6598 |        1 |      1 |
| anvesh | pingilia2    | anveshpingili@gmail.com   | Narlapur@123                              | 937-993-9675 |        1 |      1 |
| sunil  | vadlamani1   | sunilvadlamani@gmail.com  | *CBC66B80B3BA0CAF90B602AD608E3A35152319B2 | 937-512-5654 |        1 |      1 |
| venkat | vkomuravelly | komuravellyvenky@gmail.com | *776F6C81D81771C0AAE5AFA5677B404D59AF1C96 | 9640501965   |        1 |      1 |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
```

```php
<?php
  include 'header.php';
  require_once "formvalidator.php";
$show_form=true;
if(isset($_POST['Submit']))
{
    $validator = new FormValidator();
    $validator->addValidation("name","req","Please fill in Name");
    $validator->addValidation("email","email",
"The input for Email should be a valid email value");
    $validator->addValidation("email","req","Please fill in Email");
    if($validator->ValidateForm())
    {
        echo "<h2>Validation Success!</h2>";
        $show_form=false;
    }
    else
    {
        echo "<B>Validation Errors:</B>";

        $error_hash = $validator->GetErrors();
        foreach($error_hash as $inpname => $inp_err)
        {
          echo "<p>$inpname : $inp_err</p>\n";
        }
    }
}

if(true == $show_form)
{
?>
```

registration.php | changeinfo.php | regularauthentication.php | secureauthentication.php | changepasswordform.php

```
        <label for="person_name">Name:</label>
        <input type="text" id="person_name" name="name" title="Please enter your name" required /> <br>
      </div>
      <div class="input-group">
        <label for="person_email">Email_address:</label>
        <input type="email" id="person_email" name="email" title="Your Email-validtext@domain.com" required /> <br>
      </div>
      <div class="input-group">
        <label for="person_username">Username:</label>
        <input type="text" id="person_username" name="username" required pattern="\w+" title="Please enter a valid
username"
                    onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:'');" /> <br>
      </div>
      <div class="input-group">
        <label for="person_pass">Password:</label>
        <input type="password" id="person_pass" name="password" required pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}"
               title="Password must has at least 6 characters with 1 number, 1 lowercase, and 1 UPPERCASE"
               onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:'');" /> <br>
      </div>
      <div class="input-group">
        <label for="telId">Phone number:</label>
        <input type="tel" id="telId" name="phone" title="999-999-9999" required /> <br>
      </div>
      <input type="hidden" name="approval" value=0 /> <br>
      <input type="hidden" name="enable" value=0 />
      <div class="input-group">
        <button class="button" type="submit"> Register </button>
      </div>
        <p>
          Already a member? <a href="login.php">Sign in</a>
```

PHP ▾ Tab Width: 8 ▾ Ln 60, Col 26 INS

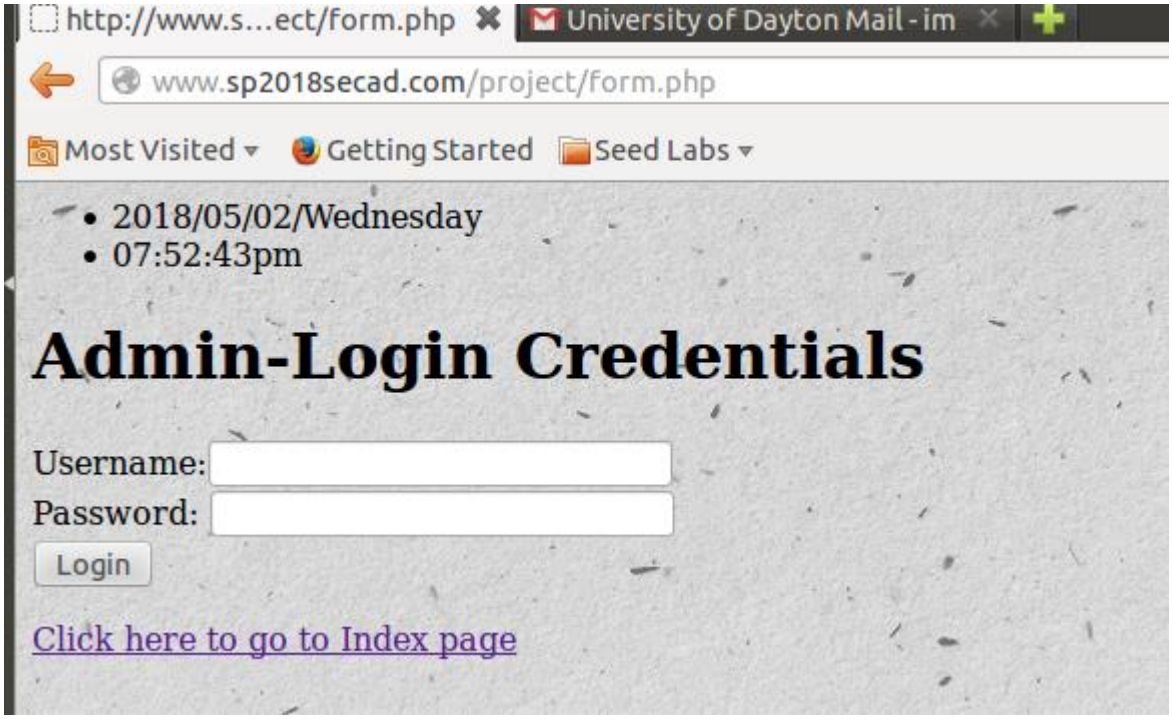
regularauthentication.php | secureauthentication.php | changepasswordform.php | approve.php

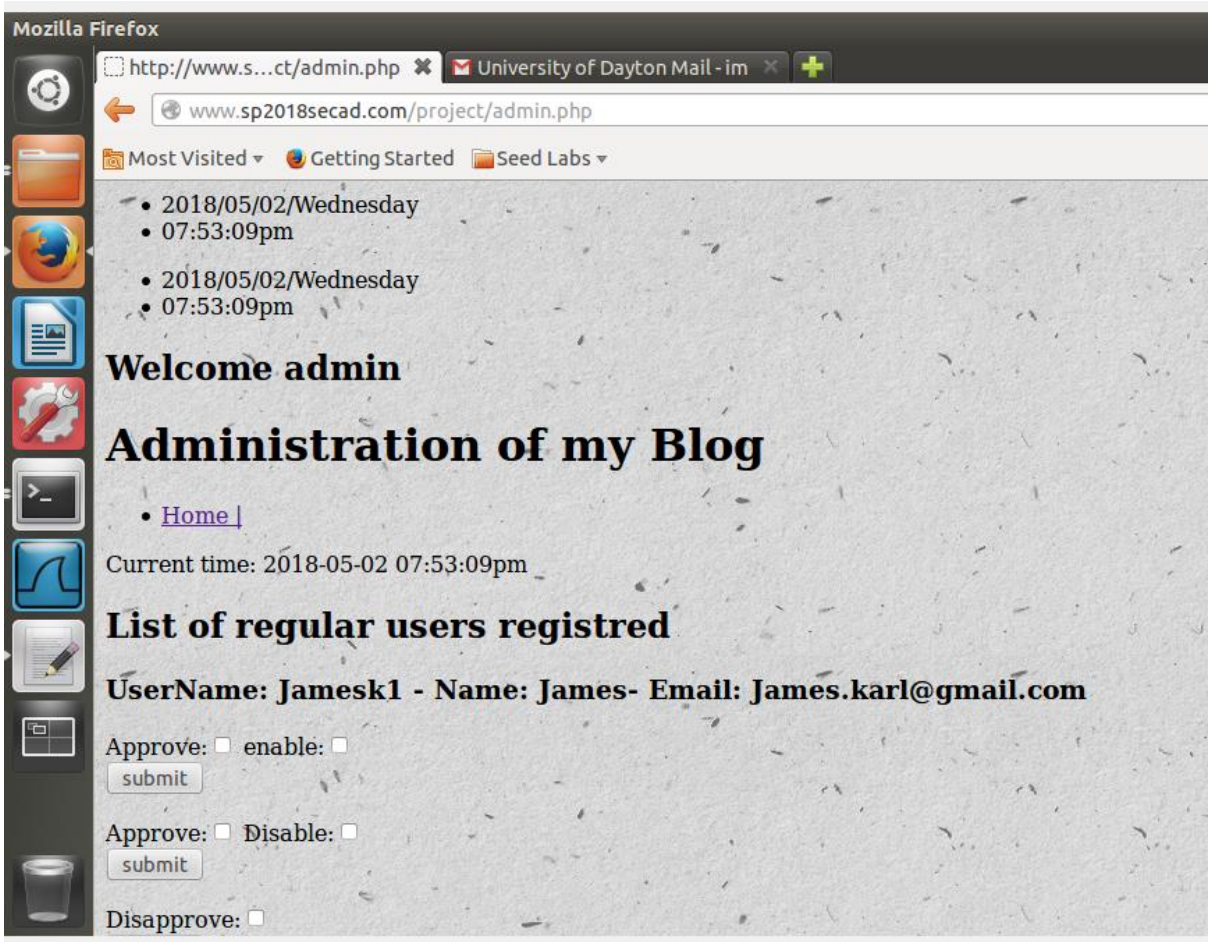
```php
<?php
  session_start();
  require 'mysql.php';
  include 'header.php';
  if (isset($_POST["username"]) and isset($_POST["password"]))
{
        $name = $_POST["name"];
        $username = $_POST["username"];
        $email = $_POST["email"];
        $password = $_POST["password"];
        $phone = $_POST["phone"];
        $approval = 0;
        $enable = 0;

        if (mysql_reguser_secure($name,$username,$email,$password,$phone,$approval,$enable)){
              $_SESSION["browser"] = $_SERVER["HTTP_USER_AGENT"];
              $_SESSION["username"] = $_POST["username"];
              header("Refresh:0; url=thankyou.php");
}
        else{
             echo "<script>alert('registration failed');</script>";
             header("Refresh:0; url=index.php");
            }
}
else{
}
?>
```

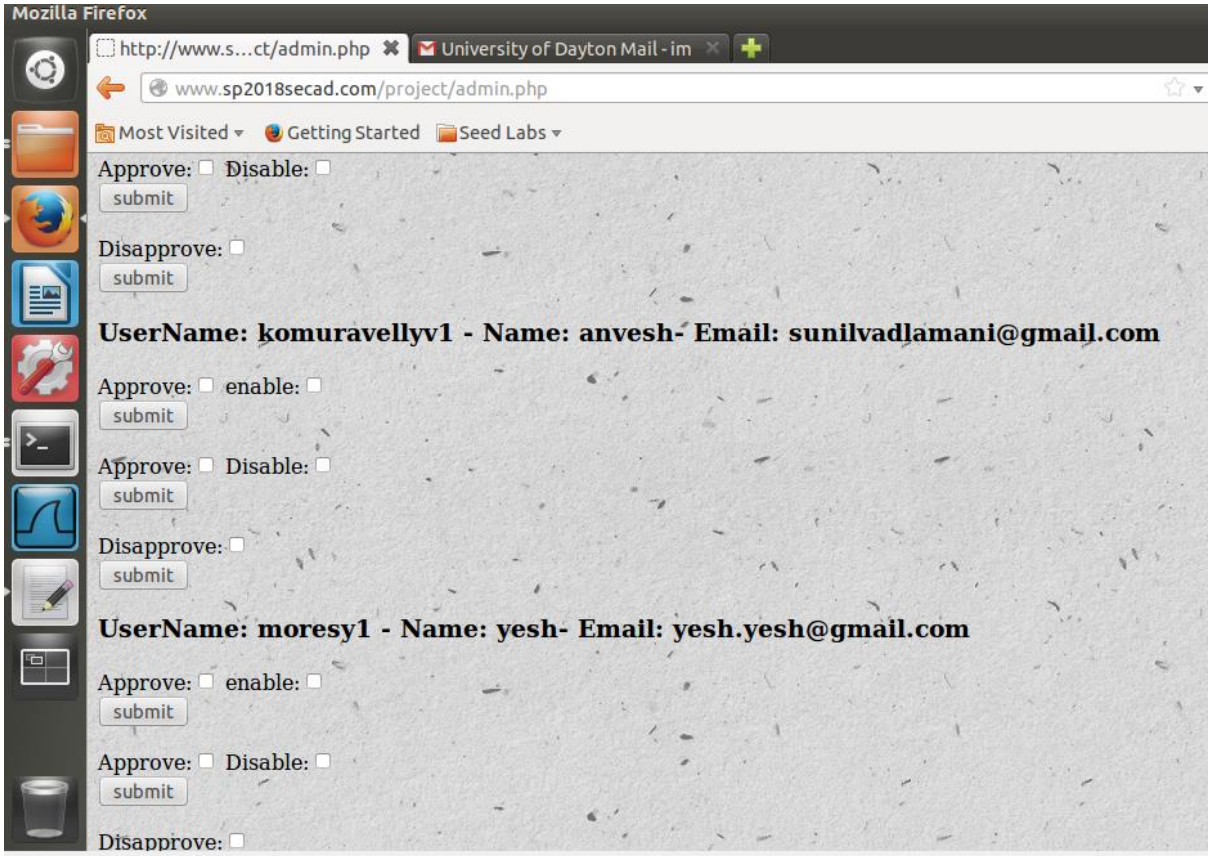
```php
function mysql_reguser_secure ($name, $username, $email, $password, $phone, $approval, $enable) {
    global $mysqli;
    $prepared_sql = "INSERT into rusers (name,username,email,password,phone,approval,enable) VALUES(?,?,?,password(?
),?,?,?);";
    if(!$stmt = $mysqli->prepare($prepared_sql))
        echo "Prepared Statement Error";
   $stmt->bind_param("sssssii", htmlspecialchars($name),htmlspecialchars($username),htmlspecialchars
($email),htmlspecialchars($password),htmlspecialchars($phone),$approval,$enable);
   if(!$stmt->execute()) {echo "Execute Error-username exists"; return FALSE;}
   return TRUE;
}
```

Anyone can register but they cannot login until approved by super user. Above query makes approval , enable =0

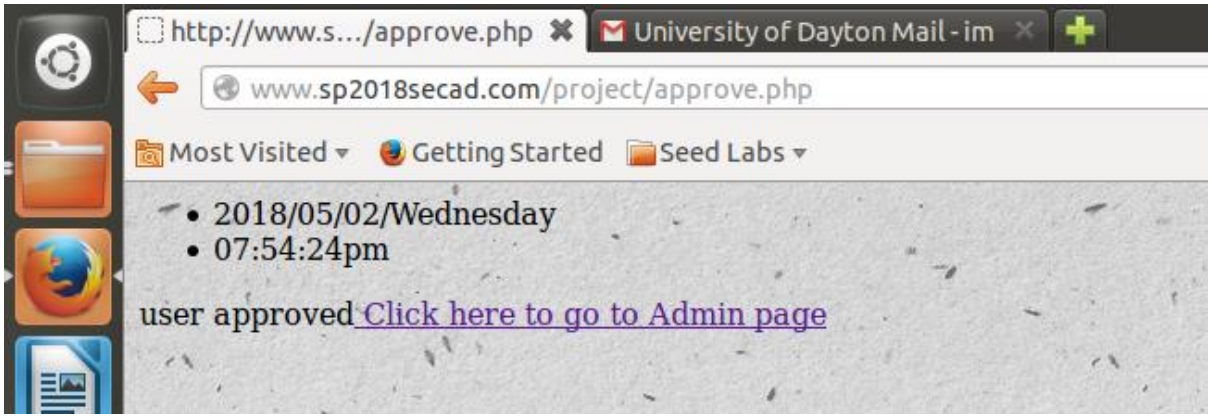
(5 points) Super users can approve/disapprove a user registration, enable/disable an account:


http://www.s...ect/form.php
University of Dayton Mail - im
www.sp2018secad.com/project/form.php
Most Visited
Getting Started
Seed Labs
2018/05/02/Wednesday
07:52:43pm
Admin-Login Credentials
Username:
Password:
Login
Click here to go to Index page


Mozilla Firefox
http://www.s...ct/admin.php
University of Dayton Mail - im
www.sp2018secad.com/project/admin.php
Most Visited
Getting Started
Seed Labs
2018/05/02/Wednesday
07:53:09pm
2018/05/02/Wednesday
07:53:09pm
Welcome admin
Administration of my Blog
Home |
Current time: 2018-05-02 07:53:09pm
List of regular users registred
UserName: Jamesk1 - Name: James- Email: James.karl@gmail.com
Approve:
enable:
submit
Approve:
Disable:
submit
Disapprove:

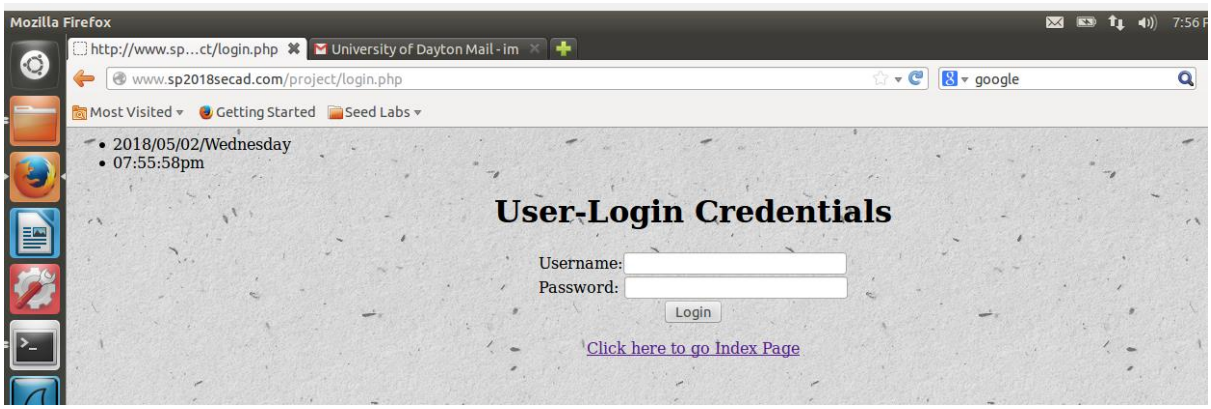


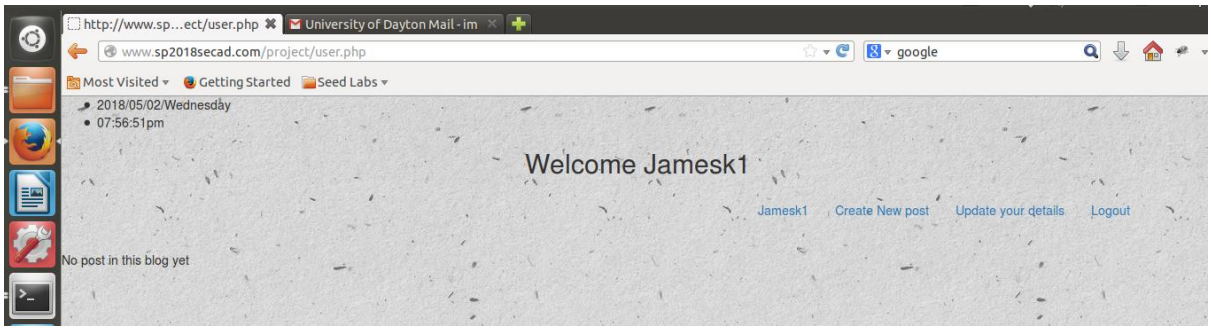
After submission this user can login to the system



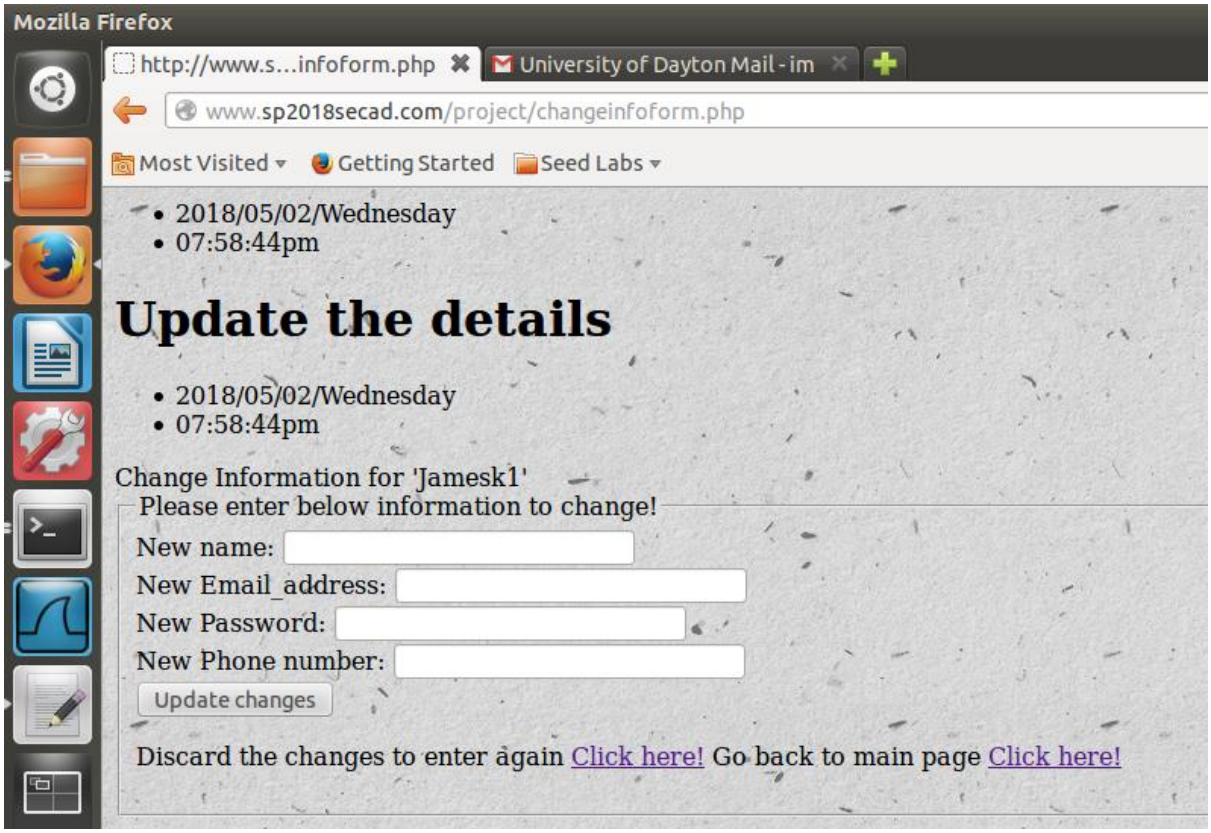
```
mysql> select * from rusers;
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| name   | username     | email                     | password                                  | phone        | approval | enable |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| James  | Jamesk1      | James.karl@gmail.com      | *ECFE4F64A77D7C65C97D7969576F78E06F590ACB | 937-512-5654 |        0 |      0 |
| anvesh | komuravellyv1 | sunilvadlamani@gmail.com  | *BE89E73620E5A4C4040705AE19A8C1FF1A823EE0 | 937-993-9676 |        0 |      0 |
| yesh   | moresy1      | yesh.yesh@gmail.com       | *3CEA034E0ED9543580C3ADF889CB4E860E49781B | 937-520-6598 |        1 |      1 |
| anvesh | pingilia2    | anveshpingili@gmail.com   | Narlapur@123                              | 937-993-9675 |        1 |      1 |
| sunil  | vadlamani1   | sunilvadlamani@gmail.com  | *CBC66B80B3BA0CAF90B602AD608E3A35152319B2 | 937-512-5654 |        1 |      1 |
| venkat | vkomuravelly | komuravellyvenky@gmail.com | *776F6C81D81771C0AAE5AFA5677B404D59AF1C96 | 9640501965   |        1 |      1 |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
6 rows in set (0.01 sec)

mysql> select * from rusers;
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| name   | username     | email                     | password                                  | phone        | approval | enable |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| James  | Jamesk1      | James.karl@gmail.com      | *ECFE4F64A77D7C65C97D7969576F78E06F590ACB | 937-512-5654 |        1 |      1 |
| anvesh | komuravellyv1 | sunilvadlamani@gmail.com  | *BE89E73620E5A4C4040705AE19A8C1FF1A823EE0 | 937-993-9676 |        0 |      0 |
| yesh   | moresy1      | yesh.yesh@gmail.com       | *3CEA034E0ED9543580C3ADF889CB4E860E49781B | 937-520-6598 |        1 |      1 |
| anvesh | pingilia2    | anveshpingili@gmail.com   | Narlapur@123                              | 937-993-9675 |        1 |      1 |
| sunil  | vadlamani1   | sunilvadlamani@gmail.com  | *CBC66B80B3BA0CAF90B602AD608E3A35152319B2 | 937-512-5654 |        1 |      1 |
| venkat | vkomuravelly | komuravellyvenky@gmail.com | *776F6C81D81771C0AAE5AFA5677B404D59AF1C96 | 9640501965   |        1 |      1 |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
6 rows in set (0.00 sec)
```

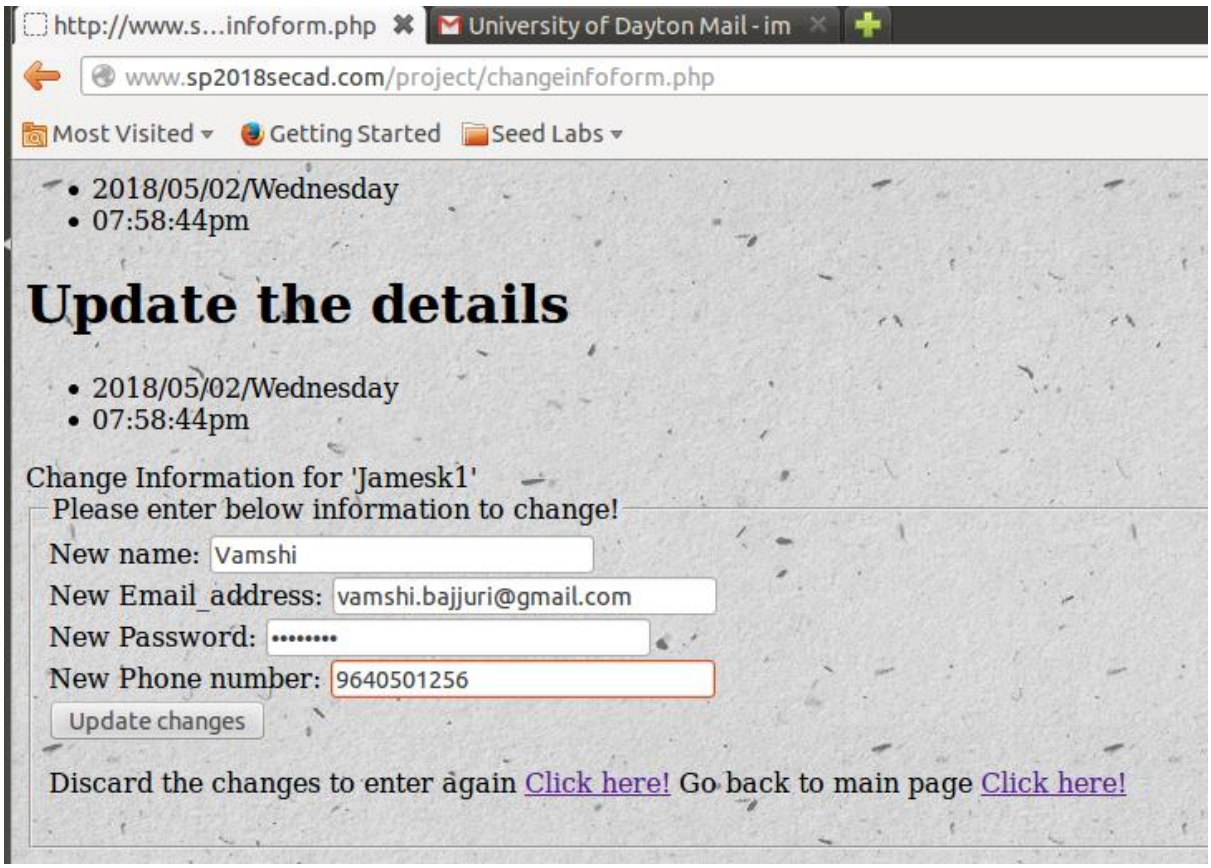
See the changes: approval ,enable changed to 1. Now this user can login.

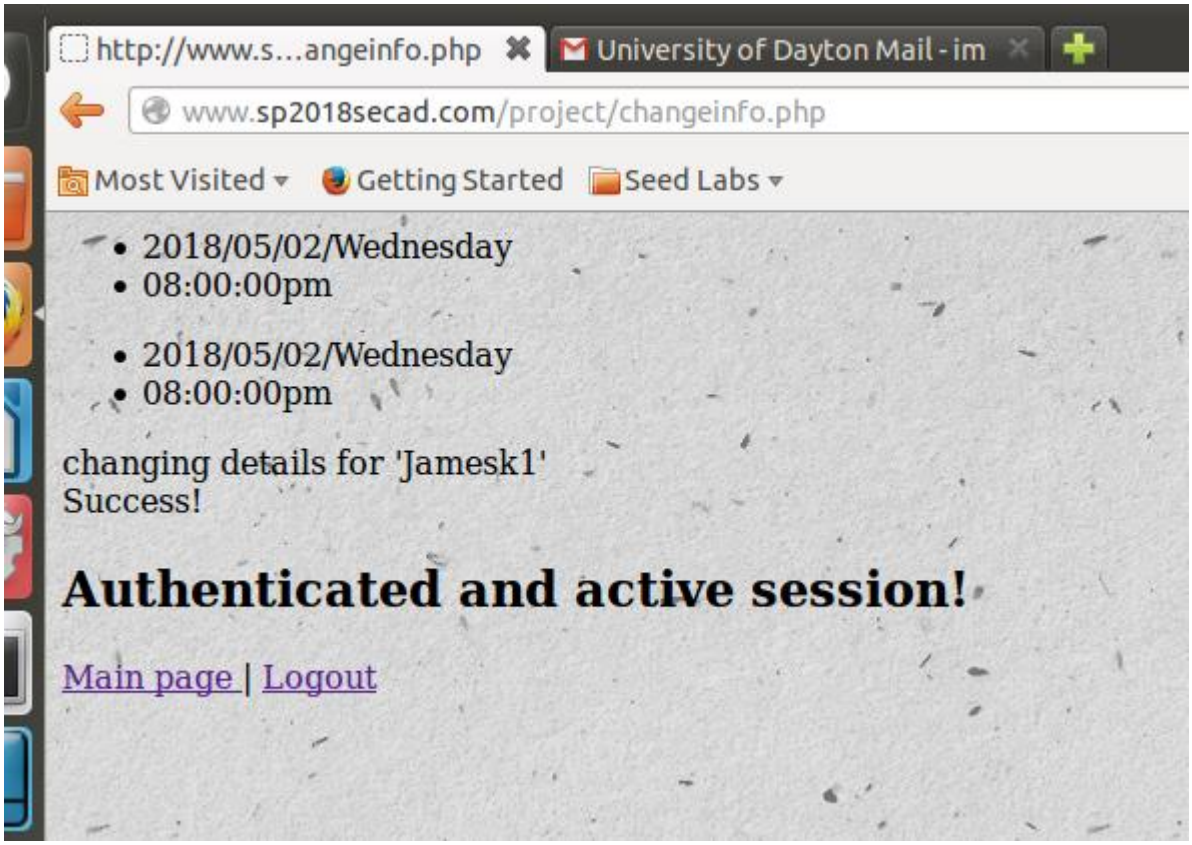




(5 points) Regular users (approved and enabled) can login and change their profiles


Mozilla Firefox
http://www.s...infoform.php
University of Dayton Mail - im
www.sp2018secad.com/project/changeinfoform.php
Most Visited
Getting Started
Seed Labs
2018/05/02/Wednesday
07:58:44pm
Update the details
2018/05/02/Wednesday
07:58:44pm
Change Information for 'Jamesk1'
Please enter below information to change!
New name:
New Email_address:
New Password:
New Phone number:
Update changes
Discard the changes to enter again Click here! Go back to main page Click here!


http://www.s...infoform.php
University of Dayton Mail - im
www.sp2018secad.com/project/changeinfoform.php
Most Visited
Getting Started
Seed Labs
2018/05/02/Wednesday
07:58:44pm
Update the details
2018/05/02/Wednesday
07:58:44pm
Change Information for 'Jamesk1'
Please enter below information to change!
New name: Vamshi
New Email_address: vamshi.bajjuri@gmail.com
New Password: ••••••••
New Phone number: 9640501256
Update changes
Discard the changes to enter again Click here! Go back to main page Click here!




```
mysql> select * from rusers;
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| name   | username     | email                     | password                                  | phone        | approval | enable |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
| Vamshi | Jamesk1      | vamshi.bajjuri@gmail.com  | *B0FC392DC7BAB8B73BB0ABD1082290C5ED5CBC03 | 9640501256   |        1 |      1 |
| anvesh | komuravellyv1 | sunilvadlamani@gmail.com | *BE89E73620E5A4C4040705AE19A8C1FF1A823EE0 | 937-993-9676 |        0 |      0 |
| yesh   | moresy1      | yesh.yesh@gmail.com       | *3CEA034E0ED9543580C3ADF889CB4E860E49781B | 937-520-6598 |        1 |      1 |
| anvesh | pingilia2    | anveshpingili@gmail.com   | Narlapur@123                              | 937-993-9675 |        1 |      1 |
| sunil  | vadlamani1   | sunilvadlamani@gmail.com  | *CBC66B80B3BA0CAF90B602AD608E3A35152319B2 | 937-512-5654 |        1 |      1 |
| venkat | vkomuravelly | komuravellyvenky@gmail.com | *776F6C81D81771C0AAE5AFA5677B404D59AF1C96 | 9640501965   |        1 |      1 |
+--------+--------------+---------------------------+-------------------------------------------+--------------+----------+--------+
6 rows in set (0.00 sec)
```

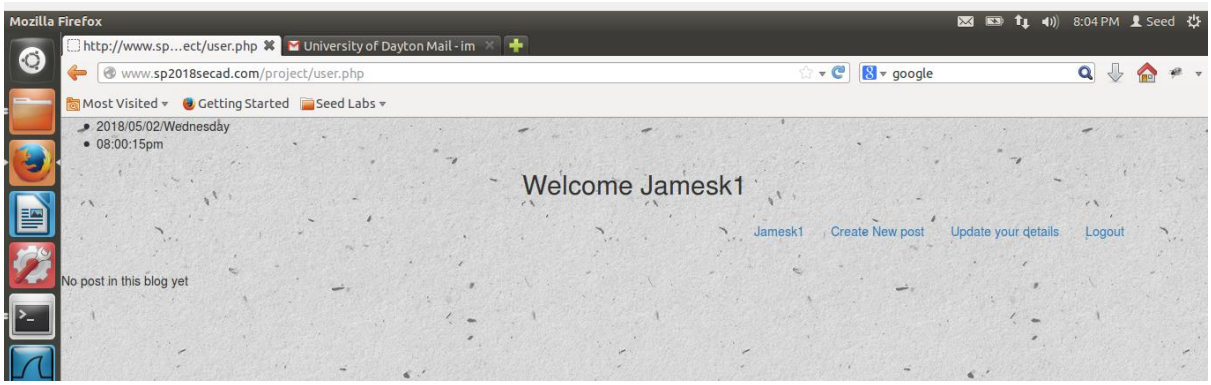
```
<?php
include 'header.php';
?>
<html>
      <h1>Update the details </h1>
<?php
  require "regularauthentication.php";
?>
 <form action="changeinfo.php" method="POST" class="form login">
           <?php
                $rand = bin2hex(openssl_random_pseudo_bytes(16));
                $_SESSION["nocsrftoken"] = $rand;
           ?>
                <input type="hidden" name="nocsrftoken" value="<?php echo $rand; ?>" />
        <?php echo "Change Information for '" . $_SESSION["username"] . "'<br>"; ?>
<fieldset>
    <legend>Please enter below information to change!</legend>
      <div class="input-group">
        <label for="person_name">New name:</label>
        <input type="text" id="person_name" name="newname" title="Please enter your name" /> <br>
      </div>
      <div class="input-group">
        <label for="person_email"> New Email_address:</label>
        <input type="email" id="person_email" name="newemail" title="Your Email-validtext@domain.com" /> <br>
      </div>
      <div class="input-group">
        <label for="person_pass"> New Password:</label>
        <input type="password" id="person_pass" name="newpassword" required pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).
{6,}"
               title="Password must has at least 6 characters with 1 number, 1 lowercase, and 1 UPPERCASE"
               onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:'');" /> <br>
```

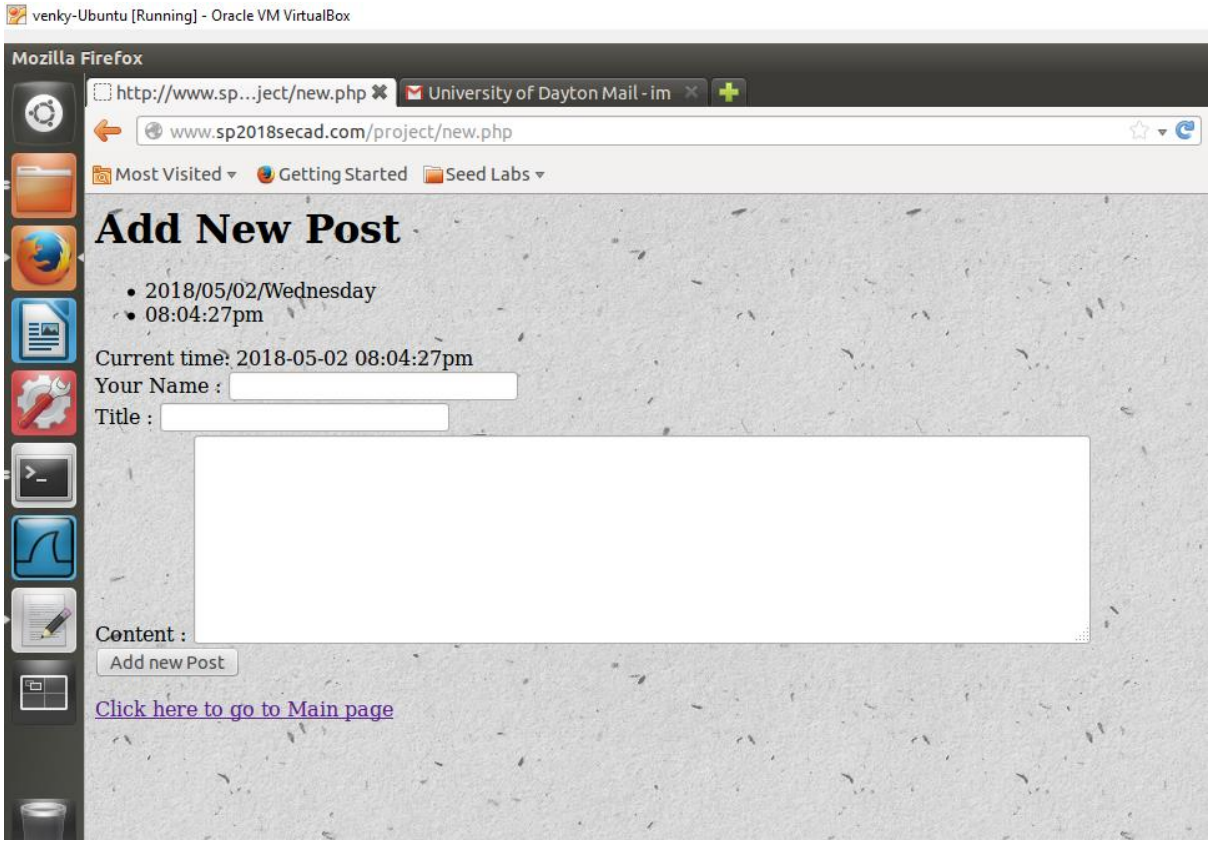
```
    <legend>Please enter below information to change!</legend>
      <div class="input-group">
        <label for="person_name">New name:</label>
        <input type="text" id="person_name" name="newname" title="Please enter your name" /> <br>
      </div>
      <div class="input-group">
        <label for="person_email"> New Email_address:</label>
        <input type="email" id="person_email" name="newemail" title="Your Email-validtext@domain.com" /> <br>
      </div>
      <div class="input-group">
        <label for="person_pass"> New Password:</label>
        <input type="password" id="person_pass" name="newpassword" required pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).
{6,}"
               title="Password must has at least 6 characters with 1 number, 1 lowercase, and 1 UPPERCASE"
               onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:'');" /> <br>
      </div>
      <div class="input-group">
        <label for="telId">New Phone number:</label>
        <input type="tel" id="telId" name="newphone" title="999-999-9999" /> <br>
      </div>
      <div class="input-group">
        <button class="button" type="submit"> Update changes </button>
      </div>
        <p>
          Discard the changes to enter again <a href="changeinfoform.php">Click here!</a>
          Go back to main page <a href="user.php"> Click here!</a>
        </p>
        </fieldset>
          </form>
  </html>
```

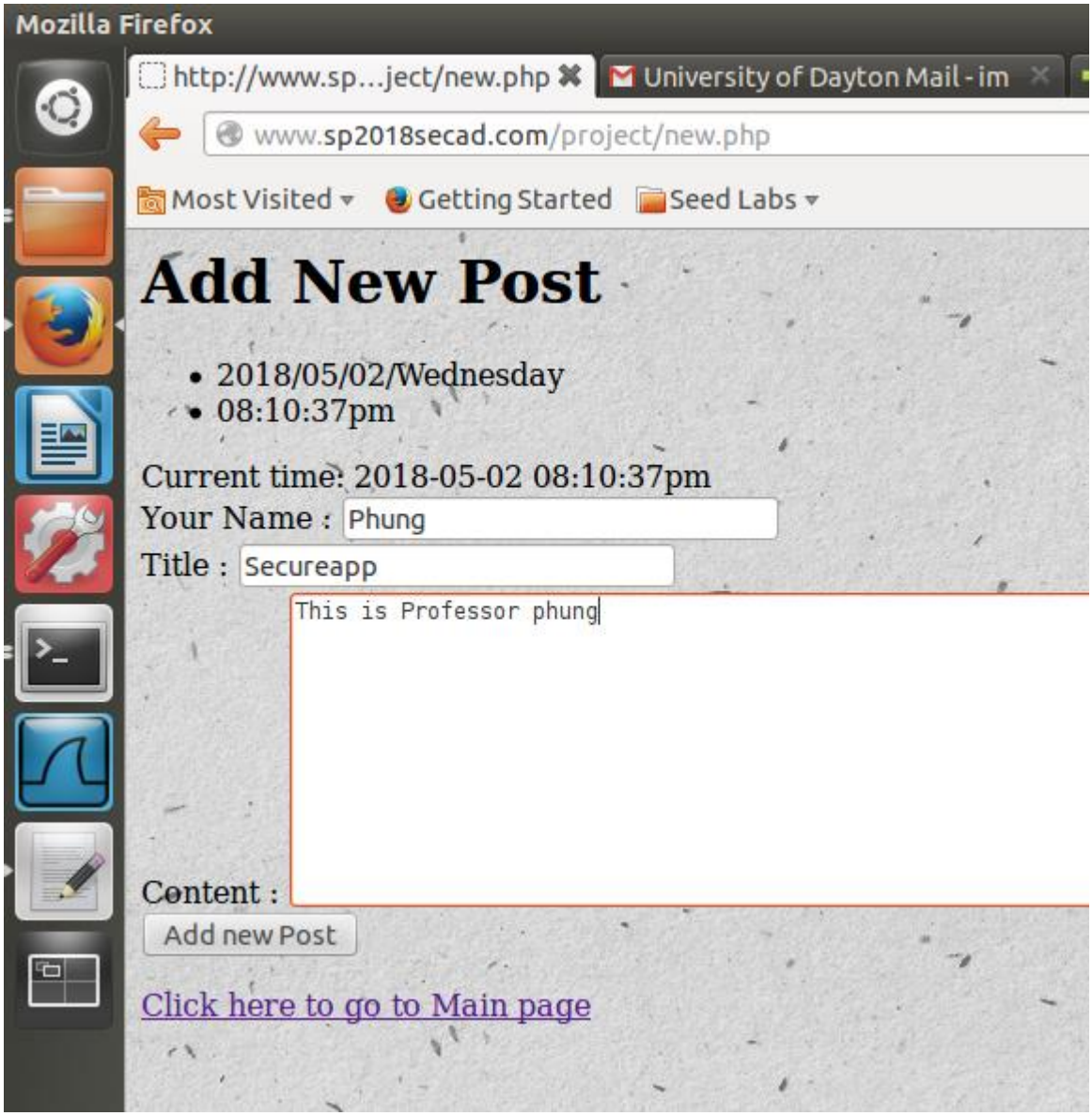
```
<?php
  include 'header.php';
  require 'regularauthentication.php';
  $username = $_SESSION["username"];
  $newname = $_REQUEST['newname'];
  $newemail = $_REQUEST['newemail'];
  $newphone = $_REQUEST['newphone'];
  $newpassword = $_REQUEST['newpassword'];
  $nocsrftoken = $_POST["nocsrftoken"];
  if(!isset($nocsrftoken) or ($nocsrftoken!=$_SESSION['nocsrftoken'])){
        echo "cross-site request forgery is detected";
        die();
        }
  if (isset($newpassword) ){
    echo "changing details for '$username' <br>";
    if (mysql_change_users_info($newname, $newpassword, $newemail, $newphone, $username)){
      echo "Success!";
    }else{
      echo "Failed!";
    }
  } else{
    echo "Cannot change password: username and password is not provided";
  }
?>
<h2> Authenticated and active session!</h2>
<a href="user.php">Main page </a> | <a href="logout.php">Logout</a>
```

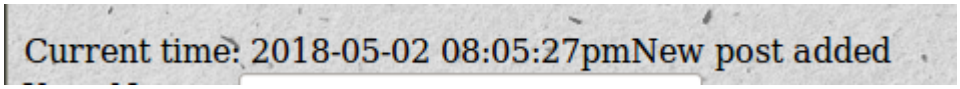
```
function mysql_change_users_info($newname, $newpassword, $newemail, $newphone, $username){
    global $mysqli;
    $prepared_sql = "UPDATE rusers SET name=?,password=password(?),email=?,phone=? WHERE username= ?";
    if(!$stmt = $mysqli->prepare($prepared_sql))
        echo "Prepared Statement Error";
    $stmt->bind_param("sssss", $newname, $newpassword, $newemail, $newphone, $username);
    if(!$stmt->execute()) {echo "Execute Error"; return FALSE;}
    return TRUE;
}
```

(5 points) Regular users can add a new post, edit, delete their own existing posts, and can enable/disable their posts

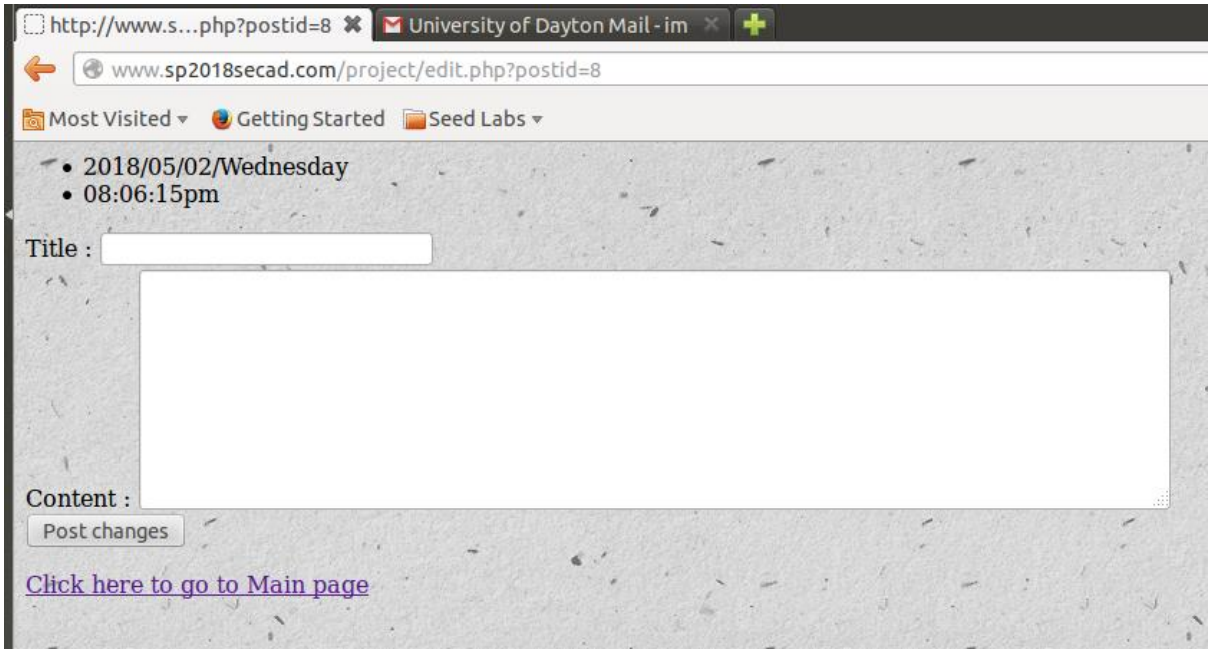

Mozilla Firefox
8:04 PM Seed
http://www.sp...ect/user.php
University of Dayton Mail - im
www.sp2018secad.com/project/user.php
google
Most Visited
Getting Started
Seed Labs
2018/05/02/Wednesday
08:00:15pm
Welcome Jamesk1
Jamesk1
Create New post
Update your details
Logout
No post in this blog yet

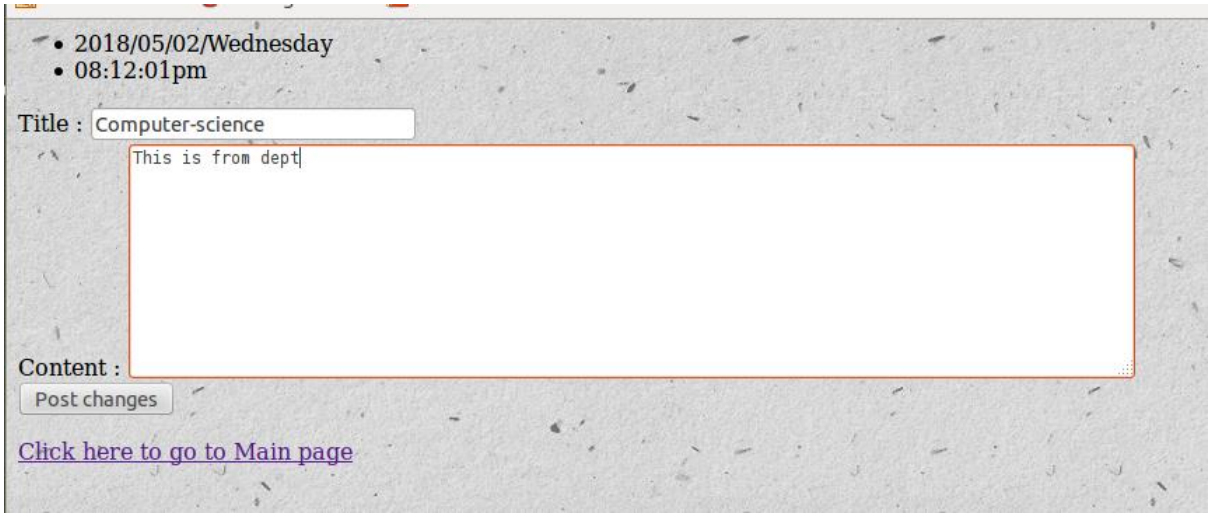

venky-Ubuntu [Running] - Oracle VM VirtualBox
Mozilla Firefox
http://www.sp...ject/new.php
University of Dayton Mail - im
www.sp2018secad.com/project/new.php
Most Visited
Getting Started
Seed Labs
Add New Post
2018/05/02/Wednesday
08:04:27pm
Current time: 2018-05-02 08:04:27pm
Your Name :
Title :
Content :
Add new Post
Click here to go to Main page


Mozilla Firefox
http://www.sp…ject/new.php
University of Dayton Mail - im
www.sp2018secad.com/project/new.php
Most Visited
Getting Started
Seed Labs
Add New Post
2018/05/02/Wednesday
08:10:37pm
Current time: 2018-05-02 08:10:37pm
Your Name : Phung
Title : Secureapp
This is Professor phung
Content :
Add new Post
Click here to go to Main page

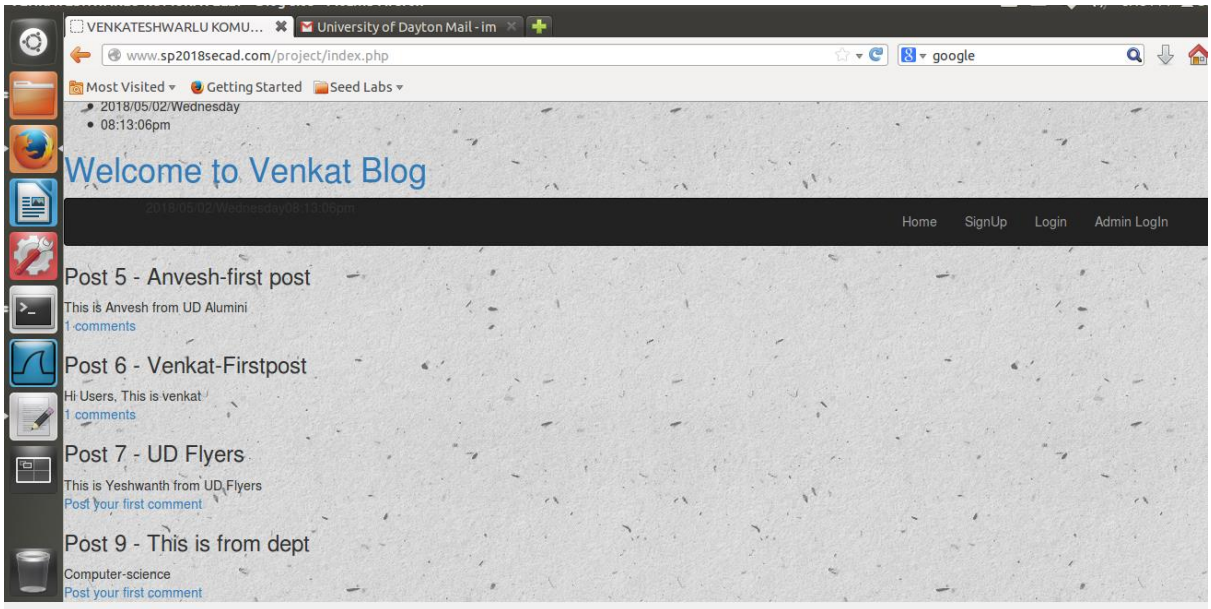

Current time: 2018-05-02 08:05:27pmNew post added


2018/05/02/Wednesday
08:11:35pm
Welcome Jamesk1
Jamesk1
Create Ne
Post 8 - This is venkat-edited
Venkat-edited
Post your first comment
edit post
delete post
Disable post
Post 9 - Secureapp
This is Professor phung
Post your first comment
edit post
delete post
Disable post

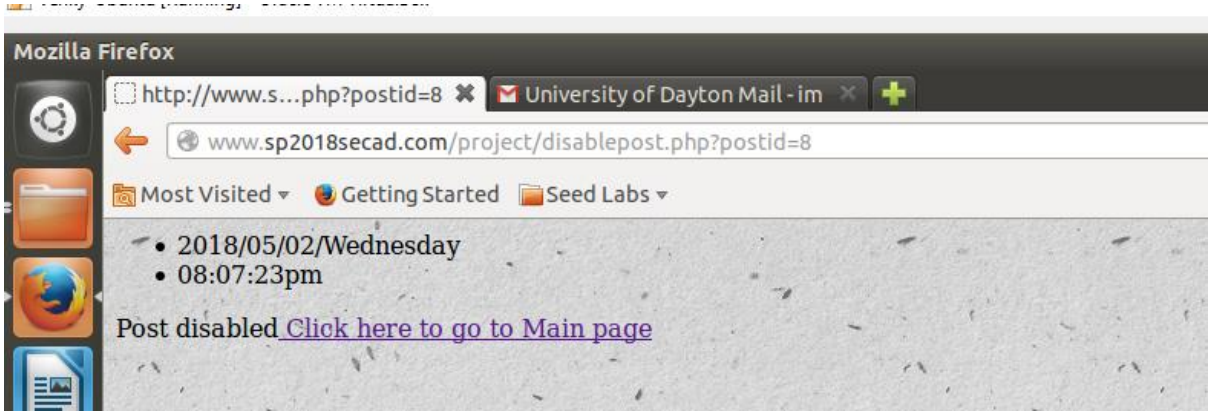

http://www.s...php?postid=8
University of Dayton Mail - im
www.sp2018secad.com/project/edit.php?postid=8
Most Visited
Getting Started
Seed Labs
2018/05/02/Wednesday
08:06:15pm
Title :
Content :
Post changes
Click here to go to Main page

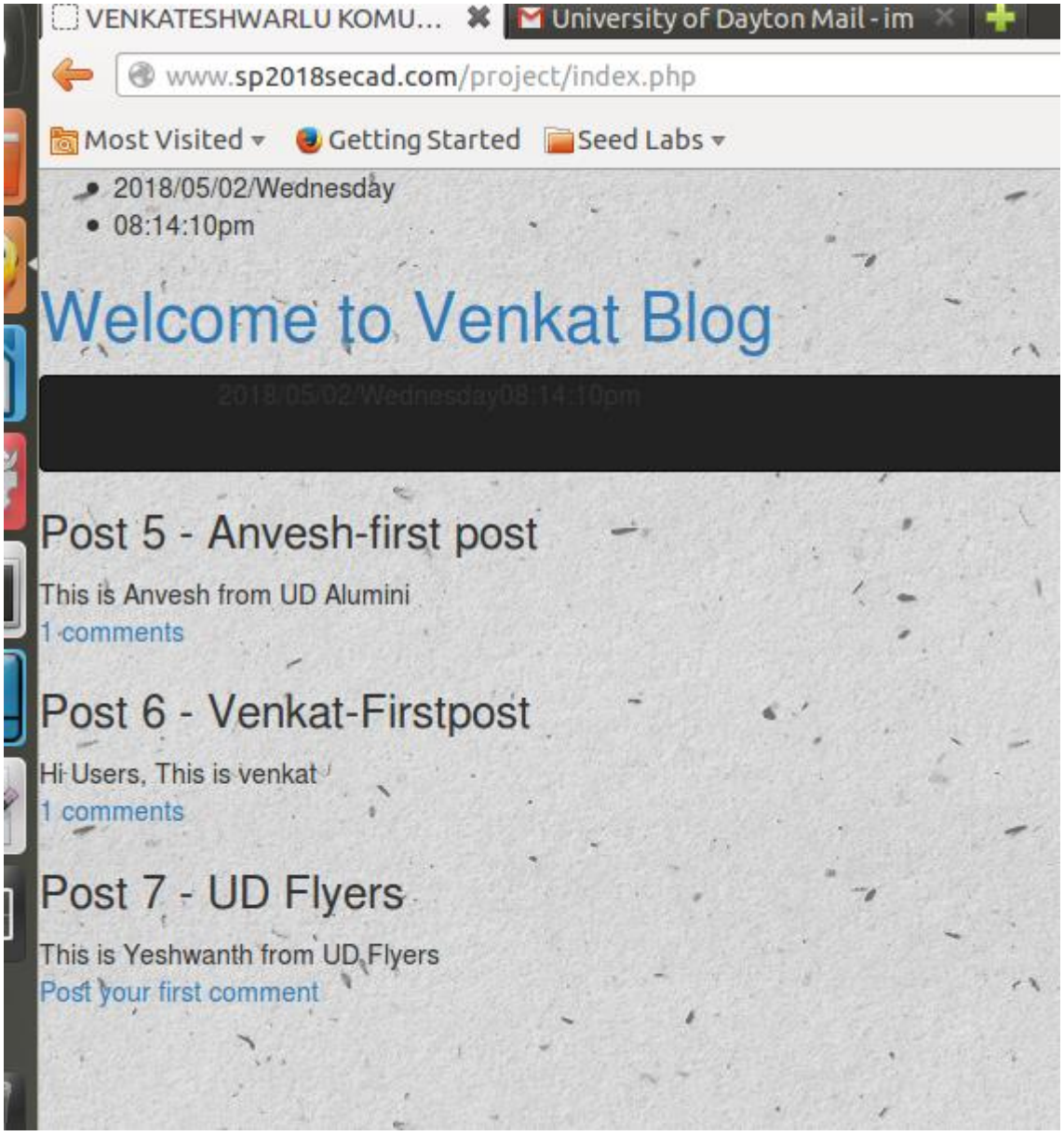


All posts created visible on index page.



After clicking on disable post.



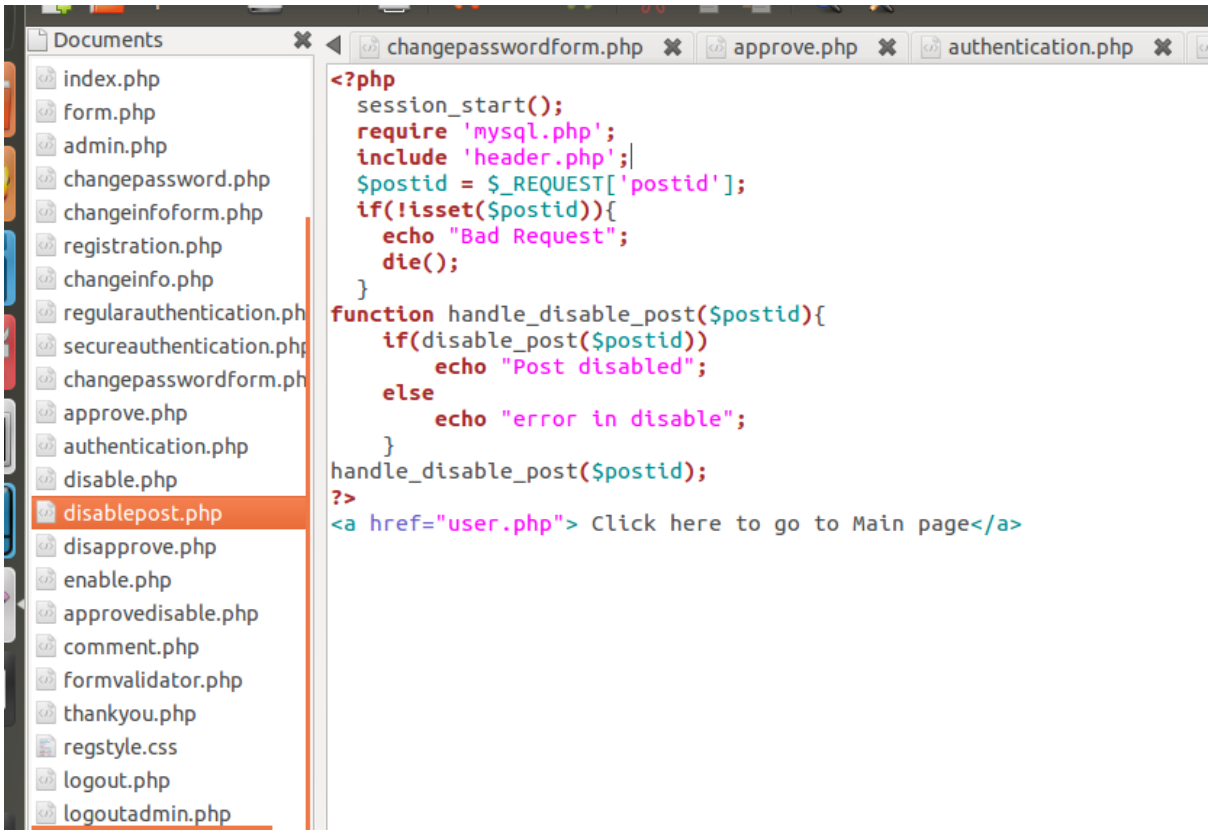


Now its disabled.

```php
<h1>Add New Post</h1>
<?php
  session_start();
  require 'mysql.php';
  include 'header.php';
  echo "Current time: " . date("Y-m-d h:i:sa");
  function handle_new_post(){
    $title = $_POST['title'];
    $text = $_POST['text'];
    $owner = $_SESSION['username'];
    $enable = 1;
    $nocsrftoken = $_POST["nocsrftoken"];
    if (isset($title) and isset($text) and isset($owner)){
        if(!isset($nocsrftoken) or ($nocsrftoken!= $_SESSION["nocsrftoken"])){
        echo "Cross-site request forgery is detected!";
        die();
        }
    if(new_post($title,$text,$owner,$enable))
        echo "New post added";
    else
        echo "Cannot add the post";
    }
  }
handle_new_post();
?>
        <form action="new.php" method="POST" class="form login">
           <?php
                $rand = bin2hex(openssl_random_pseudo_bytes(16));
                $_SESSION["nocsrftoken"] = $rand;
           ?>
                <input type="hidden" name="nocsrftoken" value="<?php echo $rand; ?>" />
```

```php
    else
        echo "Cannot add the post";
    }
  }
handle_new_post();
?>
        <form action="new.php" method="POST" class="form login">
           <?php
                $rand = bin2hex(openssl_random_pseudo_bytes(16));
                $_SESSION["nocsrftoken"] = $rand;
           ?>
                <input type="hidden" name="nocsrftoken" value="<?php echo $rand; ?>" />
                Your Name : <input type="text" name="owner" /><br>
                Title : <input type="text" name="title" required/><br>
                Content : <textarea name="text" required cols="100" rows="10"></textarea><br>
                <button class="button" type="submit">Add new Post</button>
        </form>

<a href="user.php"> Click here to go to Main page</a>
```

PHP ▾ Tab Width: 8



(5 points) All inputs and outputs must be sanitized

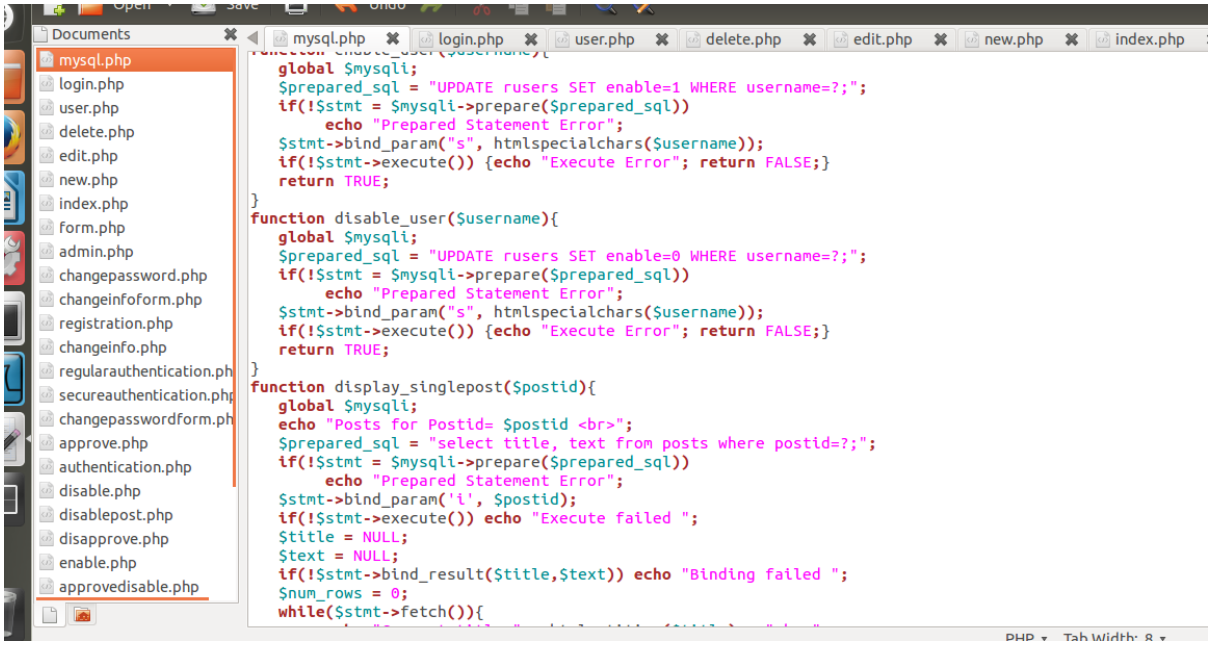
```php
function mysql_checklogin_secure ($username, $password) {
  global $mysqli;
  $prepared_sql = "SELECT * FROM users where username= ?"
  . " and password=password(?);";
  if(!$stmt = $mysqli->prepare($prepared_sql))
      echo "Prepare failed: (" . $mysqli->errno . ") " . $mysqli->error;
  $stmt->bind_param("ss", htmlspecialchars($username),htmlspecialchars($password));
  if(!$stmt->execute()) echo "Execute Error";
  if(!$stmt->store_result()) echo "Store_result Error";
  if ($stmt->num_rows == 1) return TRUE;
  return FALSE;
}
function mysql_checklogin_secure_rusers ($username, $password) {
  global $mysqli;
  $prepared_sql = "SELECT * FROM rusers where username= ?"
  . " and password=password(?) and approval=1 and enable=1;";
  if(!$stmt = $mysqli->prepare($prepared_sql))
      echo "Prepare failed: (" . $mysqli->errno . ") " . $mysqli->error;
  $stmt->bind_param("ss", htmlspecialchars($username),htmlspecialchars($password));
  if(!$stmt->execute()) echo "Execute Error";
  if(!$stmt->store_result()) echo "Store_result Error";
  if ($stmt->num_rows == 1) return TRUE;
  return FALSE;
}
function mysql_reguser_secure ($name, $username, $email, $password, $phone, $approval, $enable) {
  global $mysqli;
  $prepared_sql = "INSERT into rusers (name,username,email,password,phone,approval,enable) VALUES(?,?,?,password(?
),?,?,?);";
  if(!$stmt = $mysqli->prepare($prepared_sql))
```

Documents

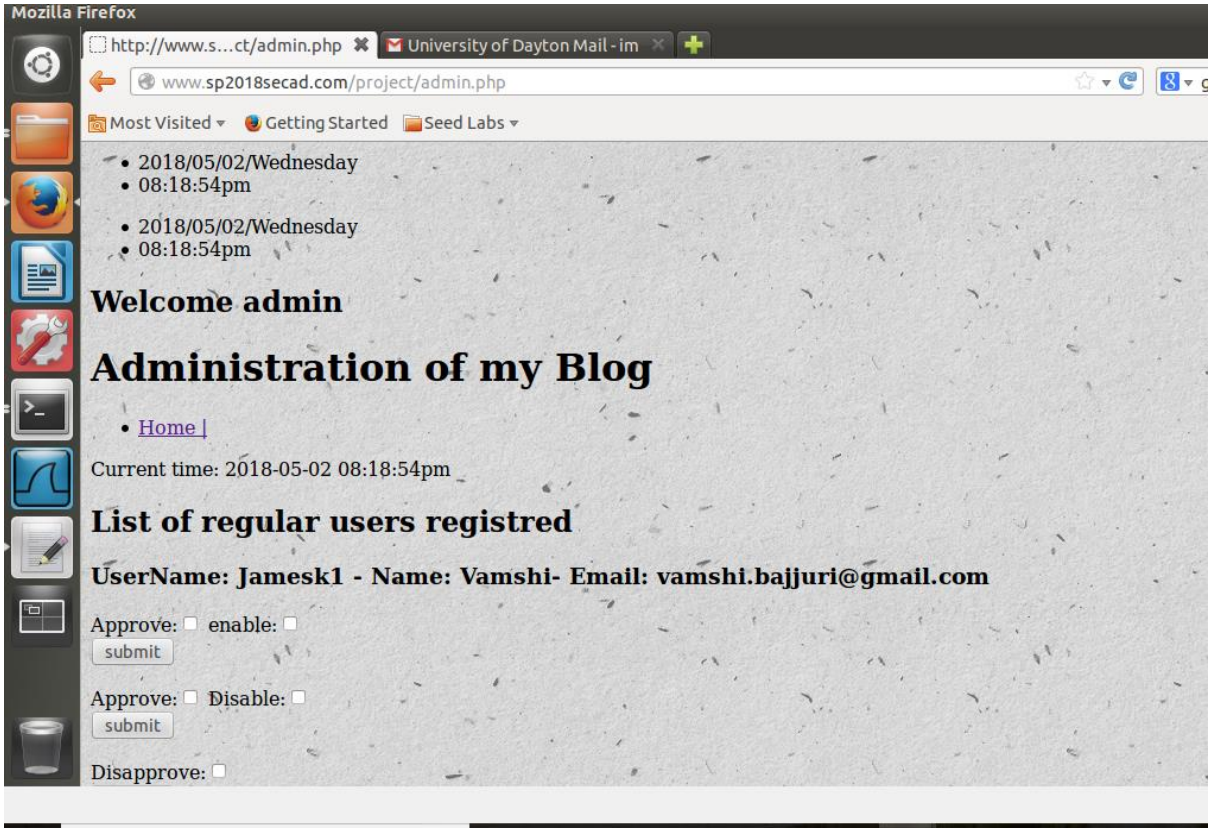
- mysql.php
- login.php
- user.php
- delete.php
- edit.php
- new.php
- index.php
- form.php
- admin.php
- changepassword.php
- changeinfoform.php
- registration.php
- changeinfo.php
- regularauthentication.ph
- secureauthentication.php
- changepasswordform.ph
- approve.php
- authentication.php
- disable.php
- disablepost.php
- disapprove.php
- enable.php
- approvedisable.php

mysql.php | login.php | user.php | delete.php | edit.php | new.php | index.php | form.

```php
function mysql_change_users_password($username, $newpassword) {
    global $mysqli;
    $prepared_sql = "UPDATE users SET password=password(?) WHERE username= ?";
    if(!$stmt = $mysqli->prepare($prepared_sql))
        echo "Prepared Statement Error";
    $stmt->bind_param("ss", htmlspecialchars($newpassword), htmlspecialchars($username));
    if(!$stmt->execute()) {echo "Execute Error"; return FALSE;}
    return TRUE;
}
function mysql_change_users_info($newname, $newpassword, $newemail, $newphone, $username){
    global $mysqli;
    $prepared_sql = "UPDATE rusers SET name=?,password=password(?),email=?,phone=? WHERE username= ?";
    if(!$stmt = $mysqli->prepare($prepared_sql))
        echo "Prepared Statement Error";
    $stmt->bind_param("sssss", $newname, $newpassword, $newemail, $newphone, $username);
    if(!$stmt->execute()) {echo "Execute Error"; return FALSE;}
    return TRUE;
}
function show_posts(){
   global $mysqli;
   $sql = "SELECT * FROM posts WHERE enable=1";
   $result = $mysqli->query($sql);
   if($result->num_rows> 0) {
     while($row = $result->fetch_assoc()) {
        $postid = $row["postid"];
        echo "<h3>Post " . $postid . " - " .$row["title"]. "</h3>";
        echo $row["text"] . "<br>";
        echo "<a href='comment.php?postid=$postid'>";
        $sql = "SELECT * FROM comments WHERE postid='$postid';";
```



(5 points) Separated roles for regular users and super users (the super users have no privilege to modify posts of regular users)



Admin had no authority on posts. He can only enable, approve, disapprove, disable users.

(10 points) The web application must be on HTTPS. Known vulnerabilities including XSS, SQL Injection, CSRF, Session Hijacking must be prevented

It satisfies all above conditions.

Prepared statements used for preventing SQL Injection

CSRF Prevented by csrftokens

SESSIONS are protected.

Database is taken care.

SECURITY ANALYSIS:

→How did you apply the security programming principles in your project?

Answer:

For SQL Injection: Used Prepared SQL statements instead of normal SQL queries.

For CSRF : tokens are implemented and validated when ever update details happened. Also, when user changes any data, its validated.

→Have you used defense in depth and defense in breath principles in your project?

Ans: defense in breath is used.

→What database security principles you have used in your project?

Passwords are hashed

User created and grant access instead of using db directly.

Tables are separated. They are in 3NF Form.

→ Is your code robust and defensive? How?

Yes, Its robust and defensive. All security principles are implemented. And programming issues are taken care.