

CPS 475/592-07 - Secure Application Development Spring 2019

Instructor: Dr. Phu Phung
Department of Computer Science,
University of Dayton

Submitted By: Akshai Addaguduru
addagudurua1@udayton.edu

PROJECT - FINAL

Bitbucket URL: <https://bitbucket.org/addagudurua1/secad-s19-addagudurua1/src/master/project/>

I. INTRODUCTION

In this project we've tried to implement a similar social networking look alike web application. Here we kick started the project by creating a registration form where in any user is allowed to register for an account. After the user is registered, he can log in to his account now where in after logging in he can view his home page. The home page has options like Change Password, Logout and Handle Posts. In the handle posts section, user can either post a status or view or update it. All the update posts and view posts are handled through MYSQL database and here we tried to integrate security constraint on every layer which is front end as well as back end.

More explained in III. Security Analysis

II. DESIGN

- **Database**

We've created the database using mysql. We did create tables like users, posts, comments, etc. using the create table commands. Also we've used some integrity constraints such as on cascade, on delete for the queries and some auto increment operations for couple of tables. The DB is used with another DB account and is not used with admin as if at any chance the DB is compromised, the root account is still safe. The new account is given root alike privileges. We've used a Database SQL file through which all the tables can be created and altered. We've integrated security constraint in user table password field where in the password is hashed by using command **UPDATE TABLE 'USERS' SET PASSWORD=PASSWORD(?) WHERE USERNAME='<USERNAME>';**

This hashes the password due to which no users can view the password of other users.

Sample DB Tables

- **CREATE TABLE `users` (**
- **`username` varchar(50) NOT NULL,**
- **`password` varchar(50) NOT NULL,**
- **`phonenummer` int(20) NOT NULL,**
- **PRIMARY KEY (`username`)**

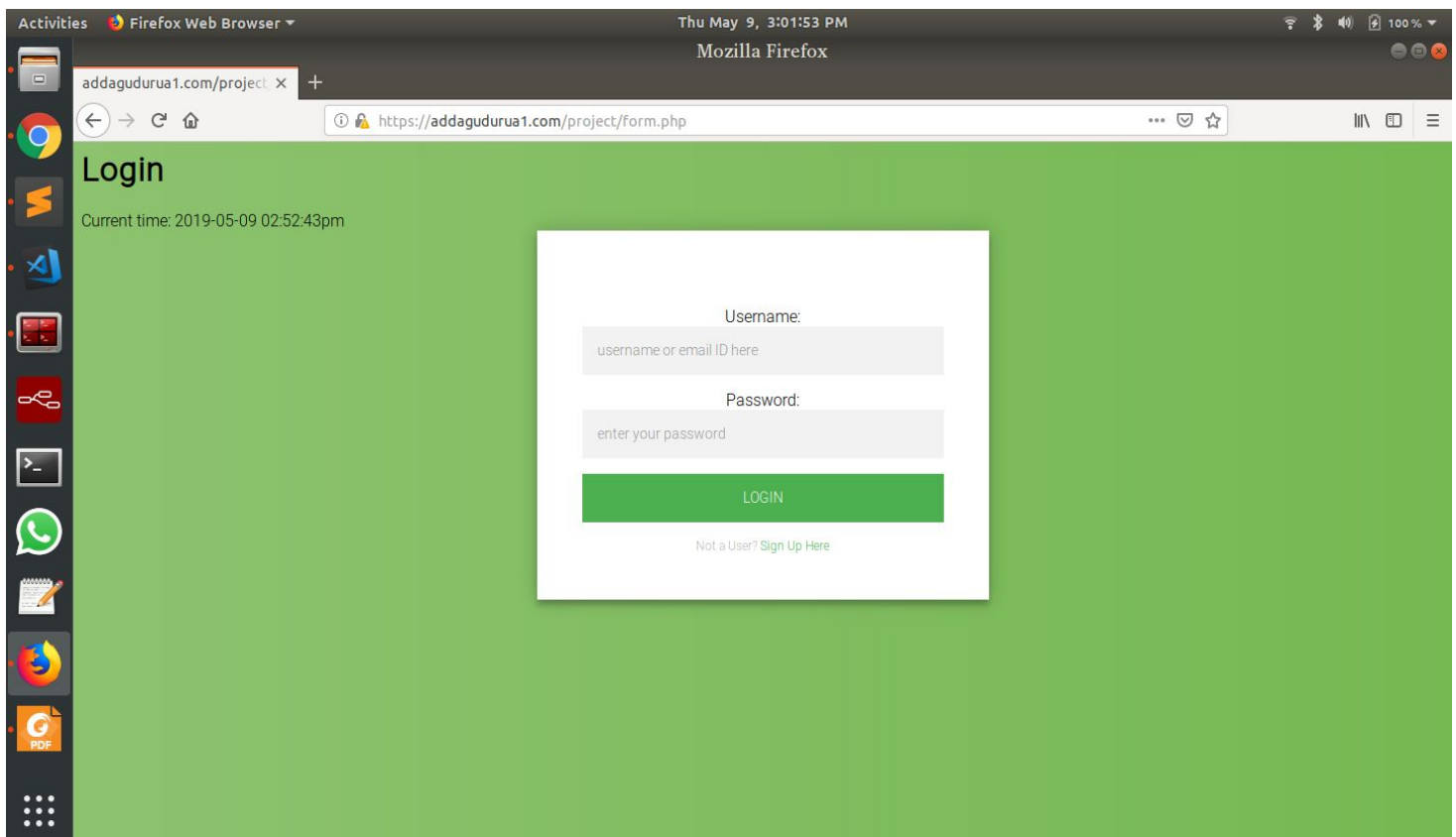
-) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `users` WRITE;

- INSERT INTO `users` VALUES ("","",0,NULL,NULL),('akshai@gmail.com','*E9FA200FBB9766346C0925B8F4EEC599F6F8CA9E',2147483647)
- UNLOCK TABLES

• UI

The UI here used is HTML and CSS. The html forms are used to create user login page as well as registration. CSS was used to style the web pages. I used a simple stylesheet which is used as common for every page.

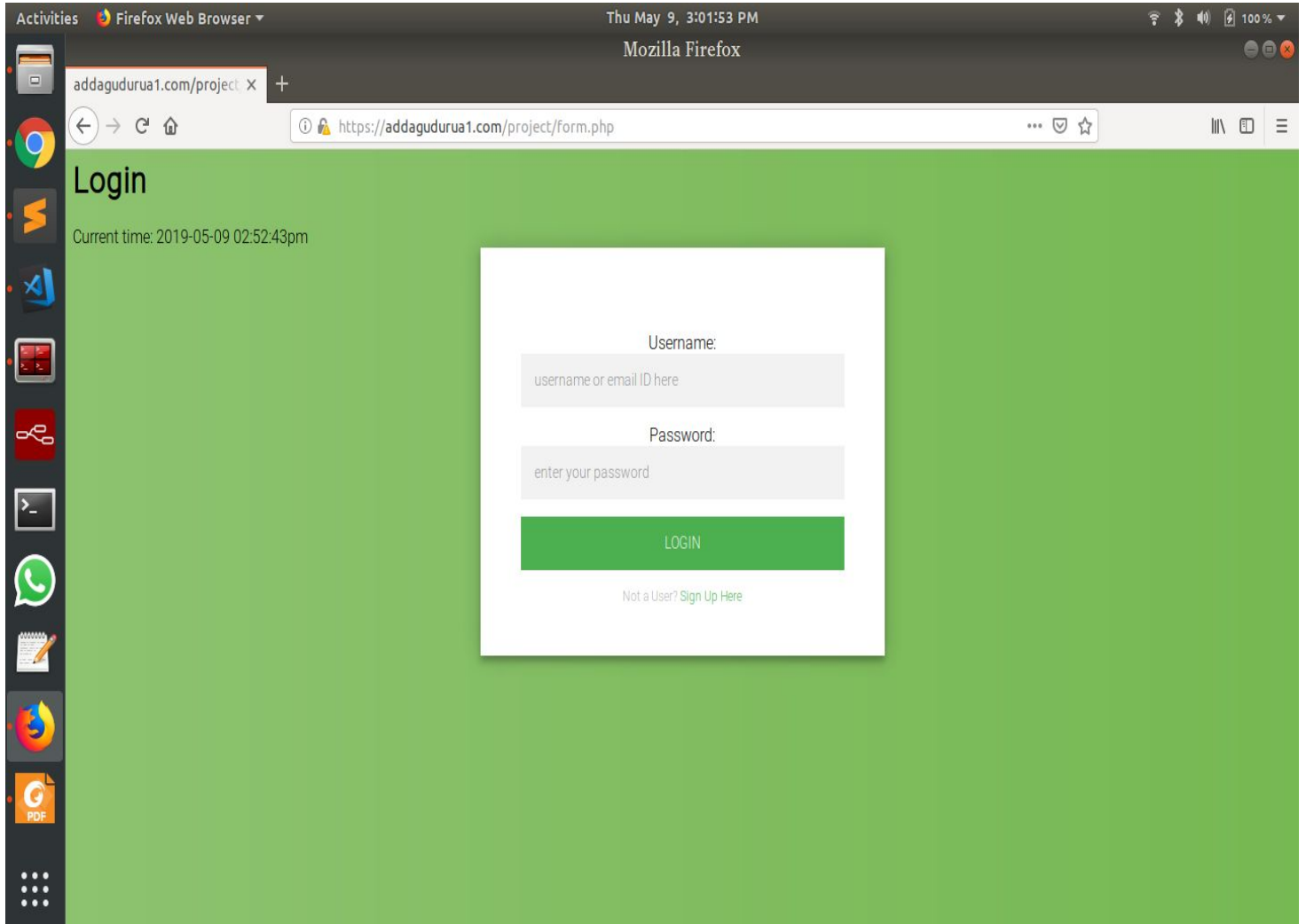


III. IMPLEMENTATION AND SECURITY ANALYSIS

In this project we've implemented security principles to protect webpage from SQL injection, Session hijacking, XSS attacks, CSRF attacks. We set **cookie parameters** and set to httponly to TRUE which keeps the data encrypted and the cookie or php session ID cannot be stolen. We also sanitized the outputs using **htmlentities** and **htmlspecialchars**. We used all sql statements in prepared sql statements which prevent sql injection. We set error statements for every condition and tried to make sure the site handles errors naturally without abrupt breaks. So we can say that our code is robust and defensive because of this. We even made sure cookies cannot be fetched using **document.cookie** using browser inspect element and avoided the session ID hijacking too. To change password of a user, we made sure that only that particular user whose session is running can only change his password but not others. Also every exception is handled with an error message and every function is validated properly to handle issues. In the code we've locked out users to edit other people's posts but can edit their own post. They can do the same for delete.

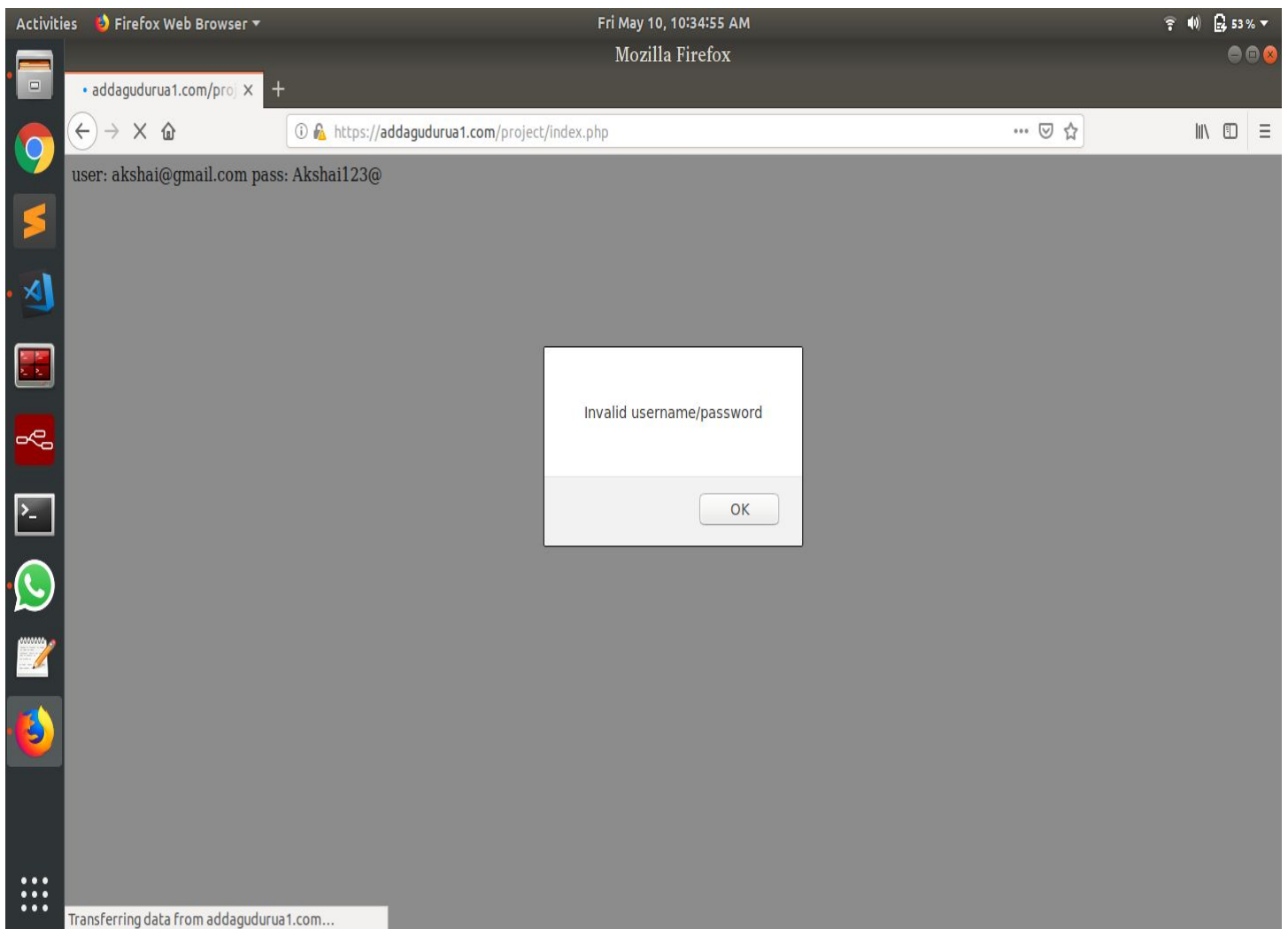
IV. DEMO

Login form



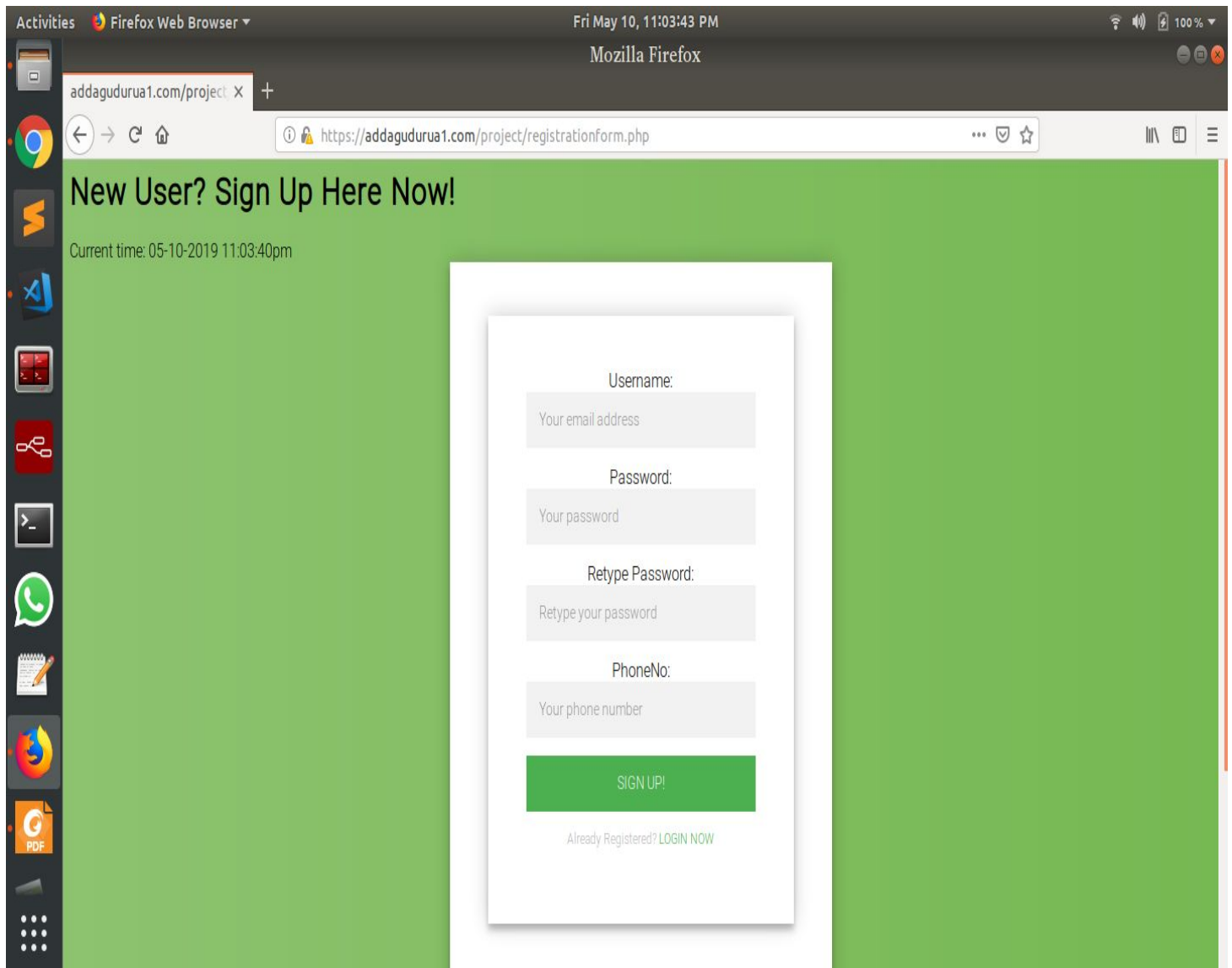
Registered users only can login here. It checks for users available in database.

Non-existing users



We see an error for non-existing users or users whose password is wrong either.

Registration Form



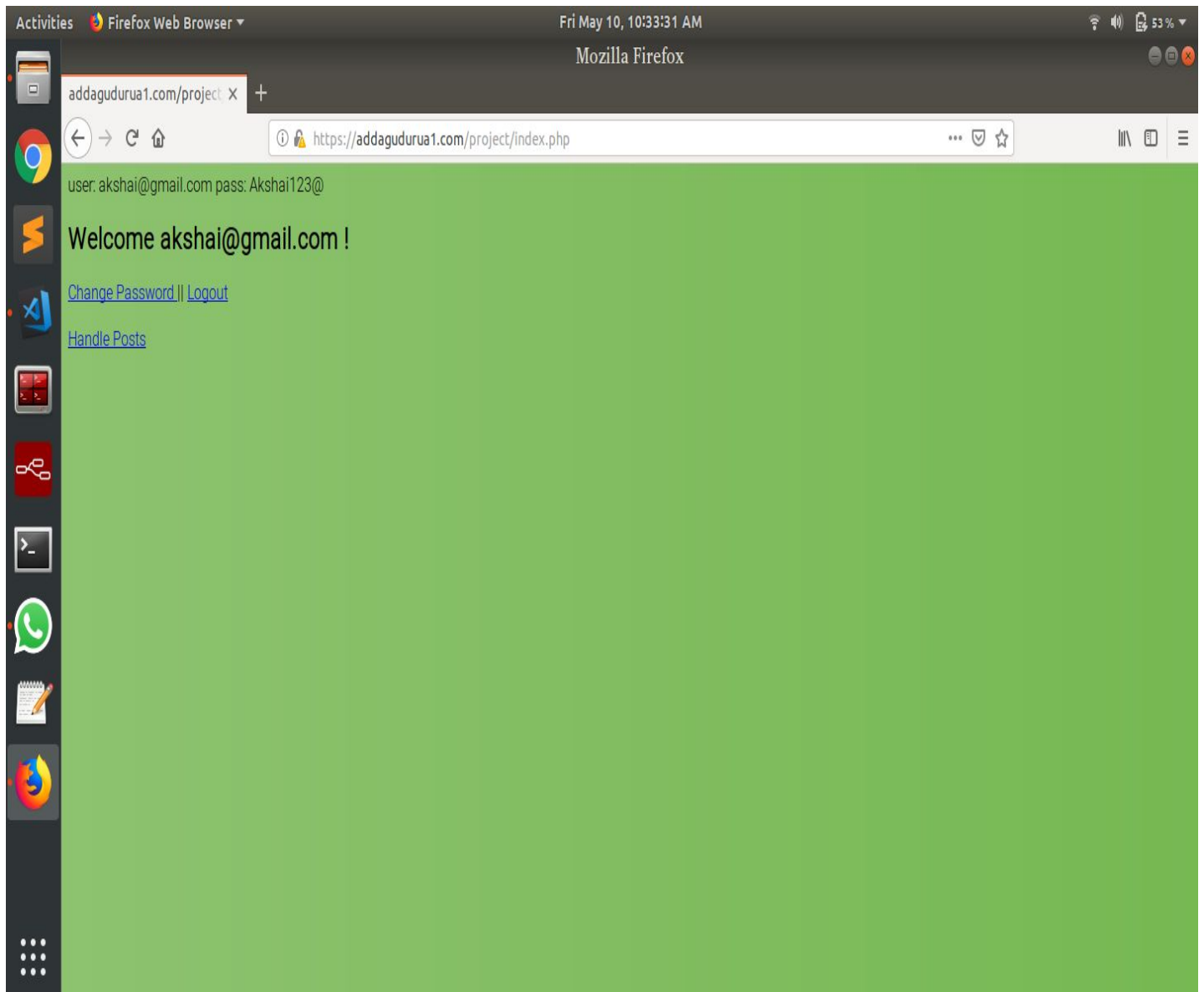
The screenshot shows a Mozilla Firefox browser window with the address bar displaying `https://addaguduraa1.com/project/registrationform.php`. The page has a green background and a white registration form in the center. The form contains the following fields and buttons:

- Username:** A text input field with the placeholder text "Your email address".
- Password:** A text input field with the placeholder text "Your password".
- Retype Password:** A text input field with the placeholder text "Retype your password".
- PhoneNo:** A text input field with the placeholder text "Your phone number".
- SIGN UP!** A green button.
- Already Registered? LOGIN NOW** A link in green text.

At the top left of the page, the text "New User? Sign Up Here Now!" is displayed. Below it, the text "Current time: 05-10-2019 11:03:40pm" is shown. The browser's status bar at the top indicates the date and time as "Fri May 10, 11:03:43 PM" and the battery level as "100 %".

Anyone can register for account and they can login right away using LOGIN page.

Home Page



Here users can use this page to change password and add posts or manage other things

Change Password

Change Password

SecAD-S19 Lab 6.2 by Akshai

Current time: 2019-05-10 10:33:33am

Username: akshai@gmail.com

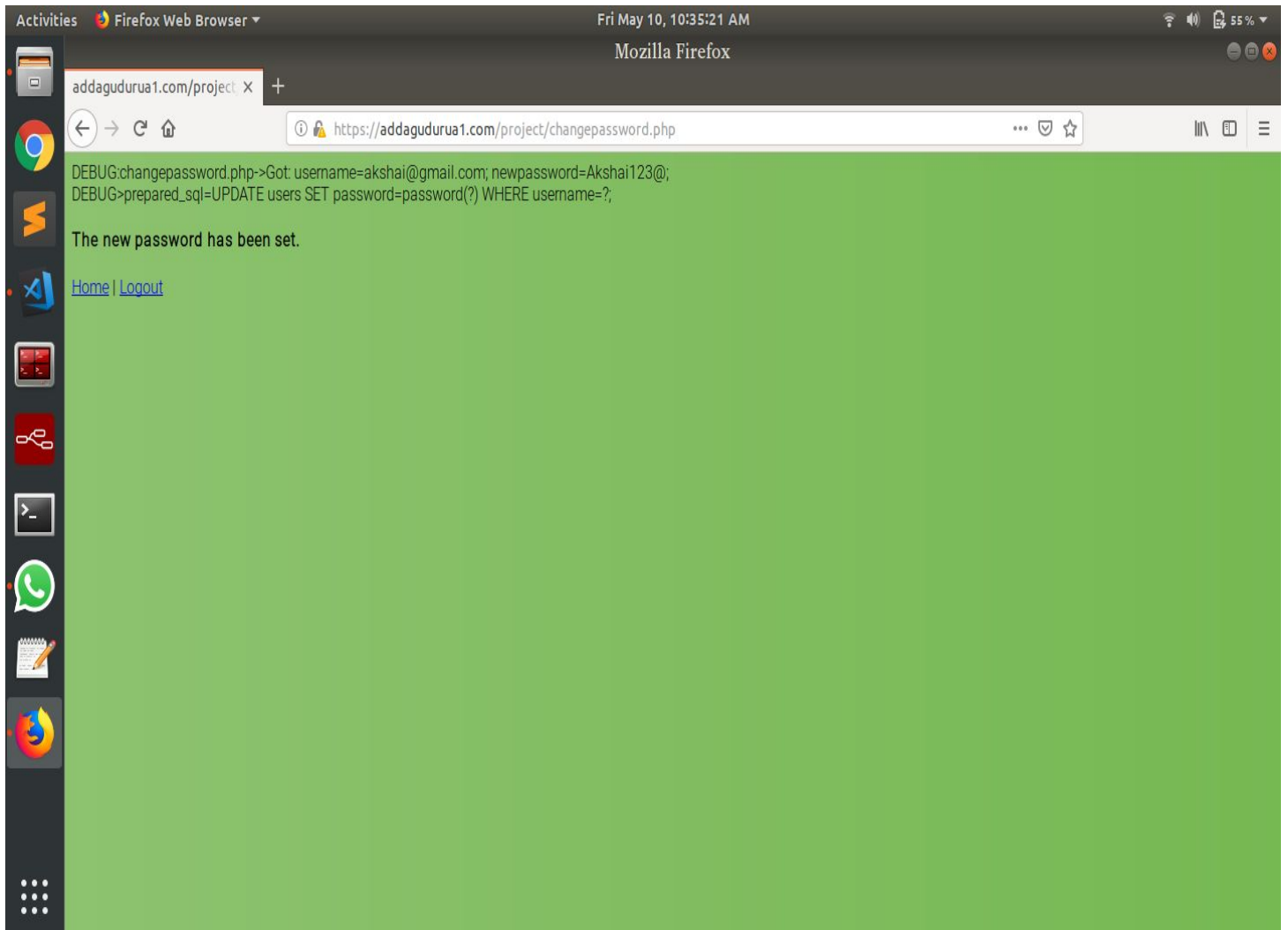
New Password:

Your password

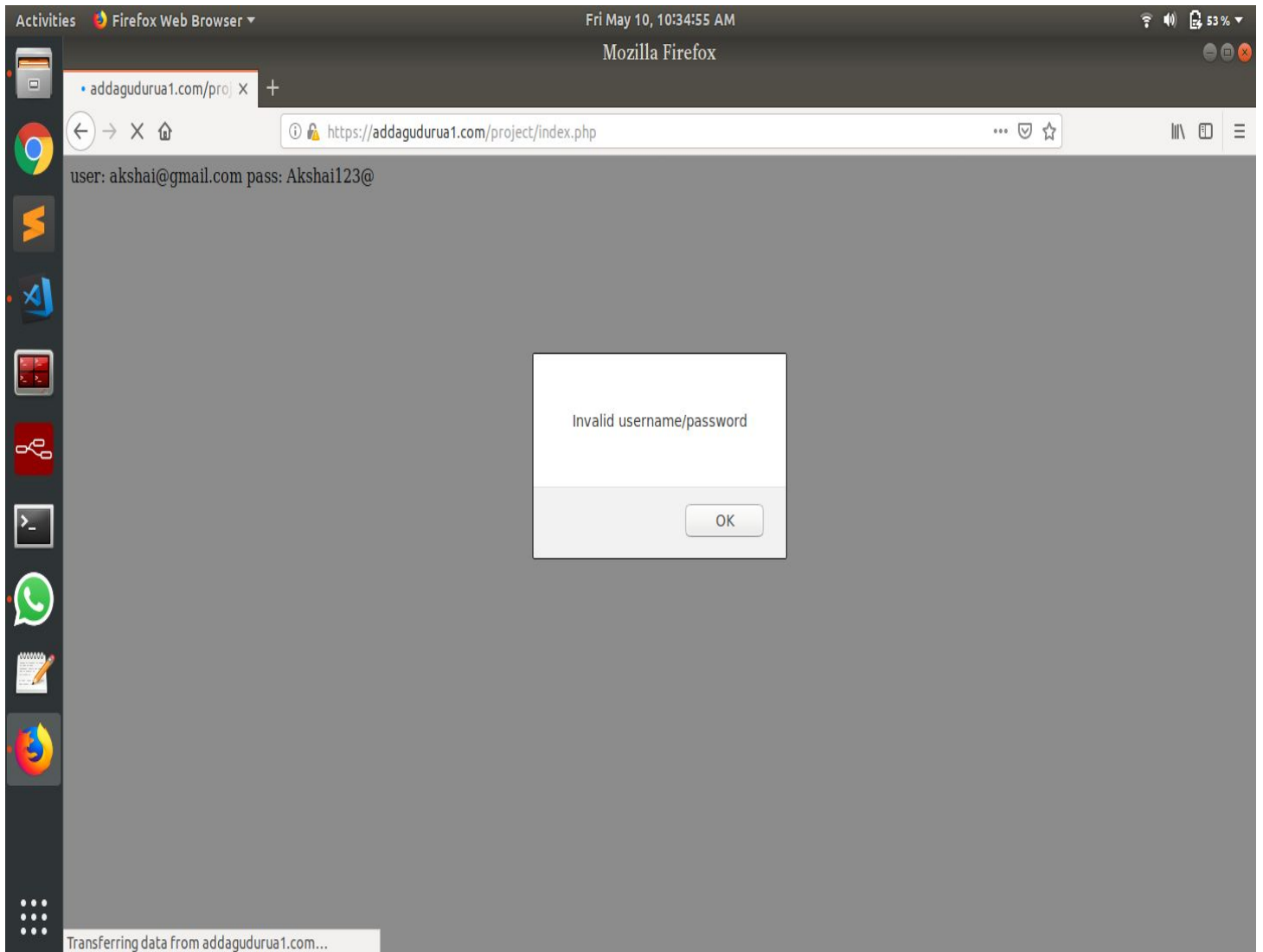
CHANGE PASSWORD

Back to Previous Page

Change password page where user can change his password and use a password of at least 8 characters using accented letters and symbols at least once.

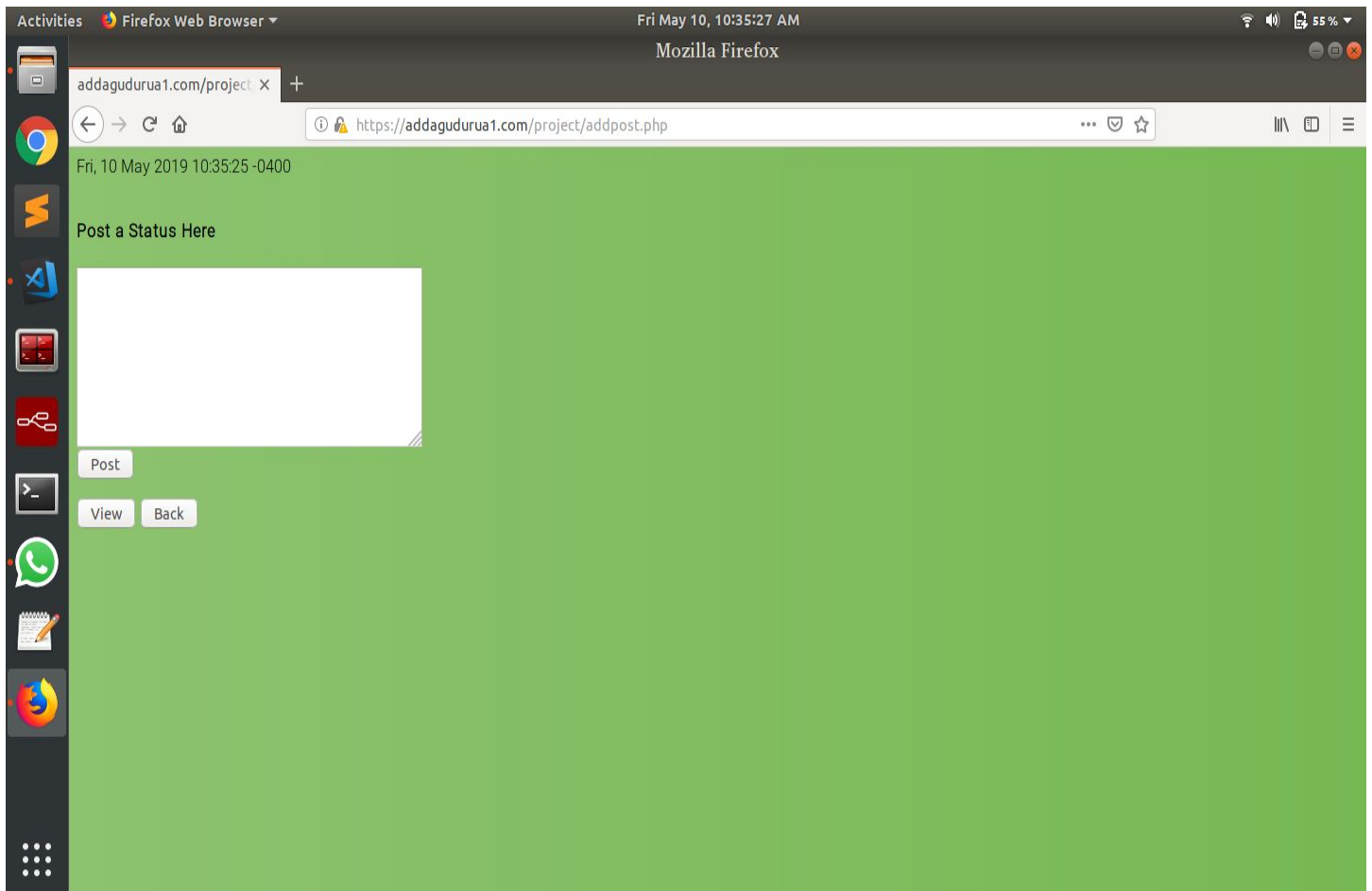


Change password done successfully after using the page.

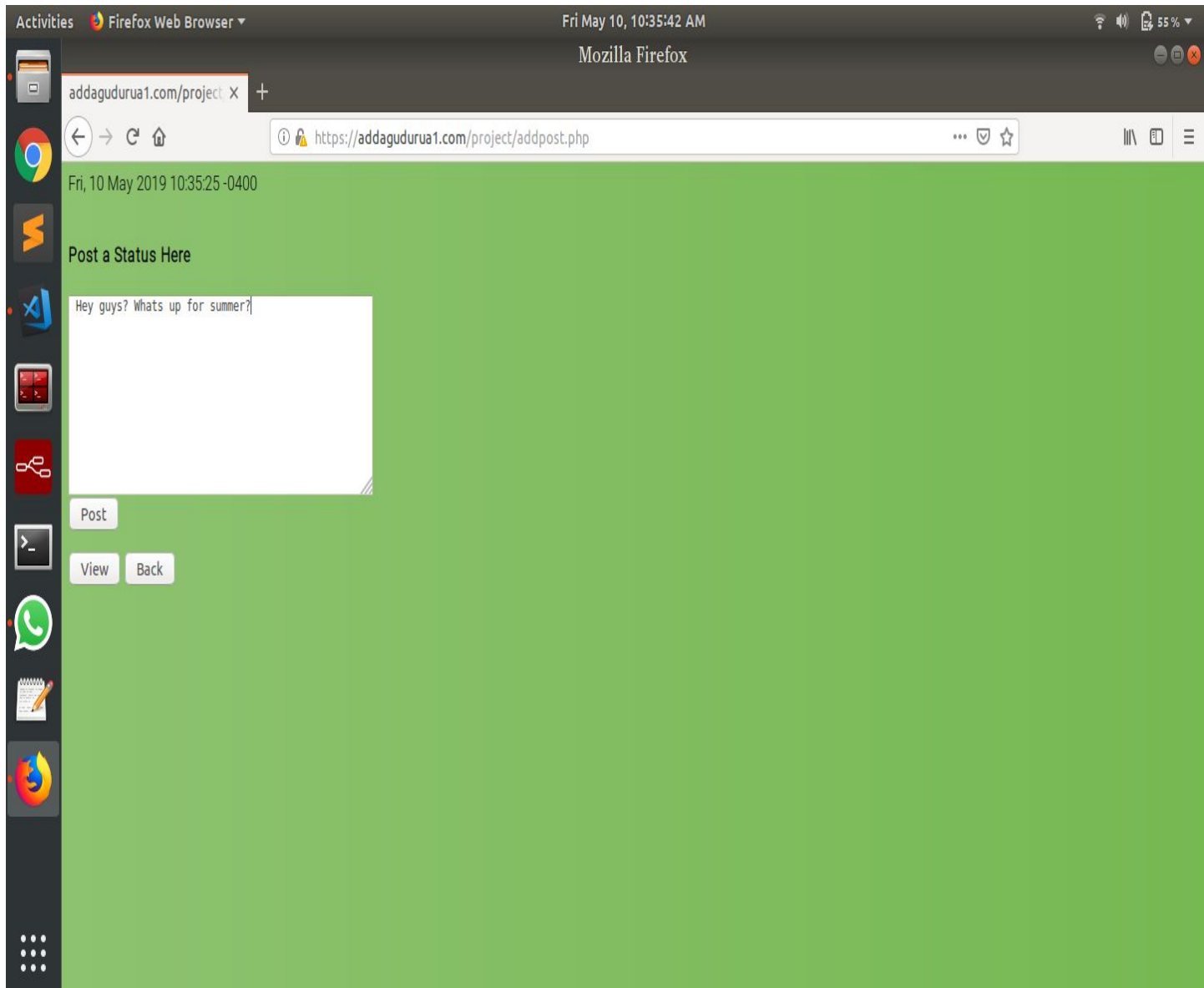


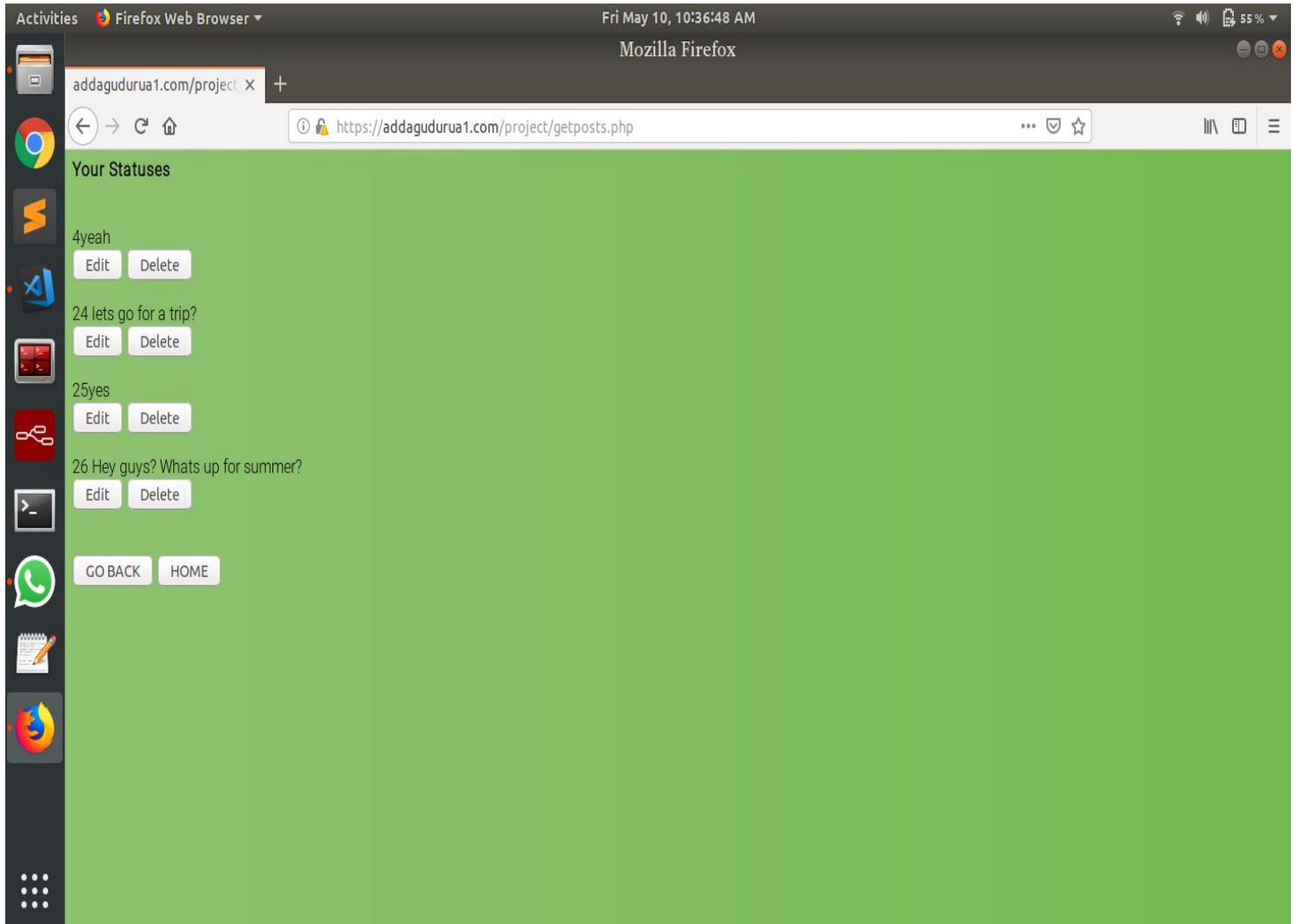
Error in using old password after changing password. Use new password only

Handle Post



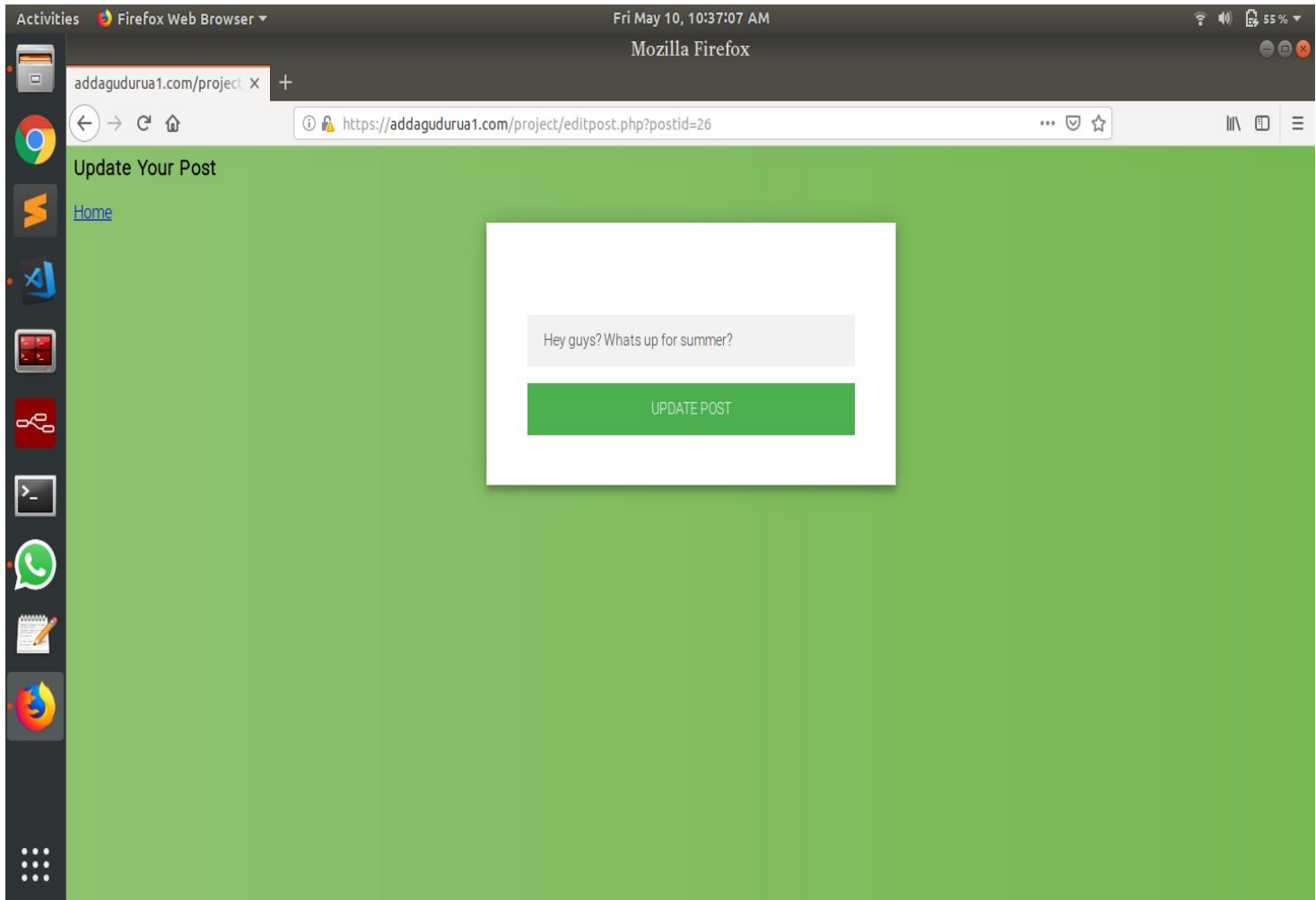
Users can post a status here using the text area and then on clicking post they can post it.



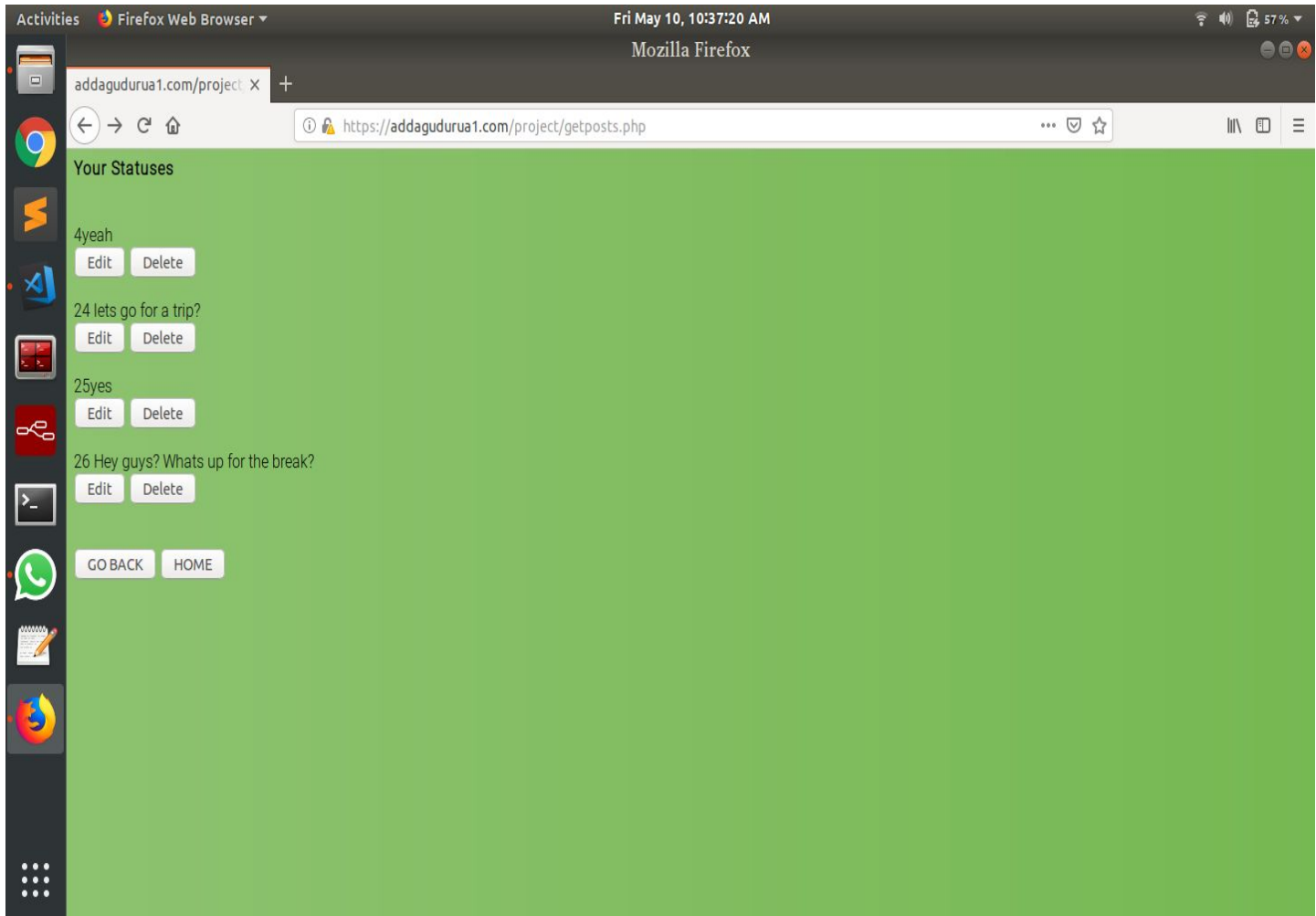


On clicking view posts users can view every users posts as well as his. He can then edit his post or delete it but cannot edit/delete others.

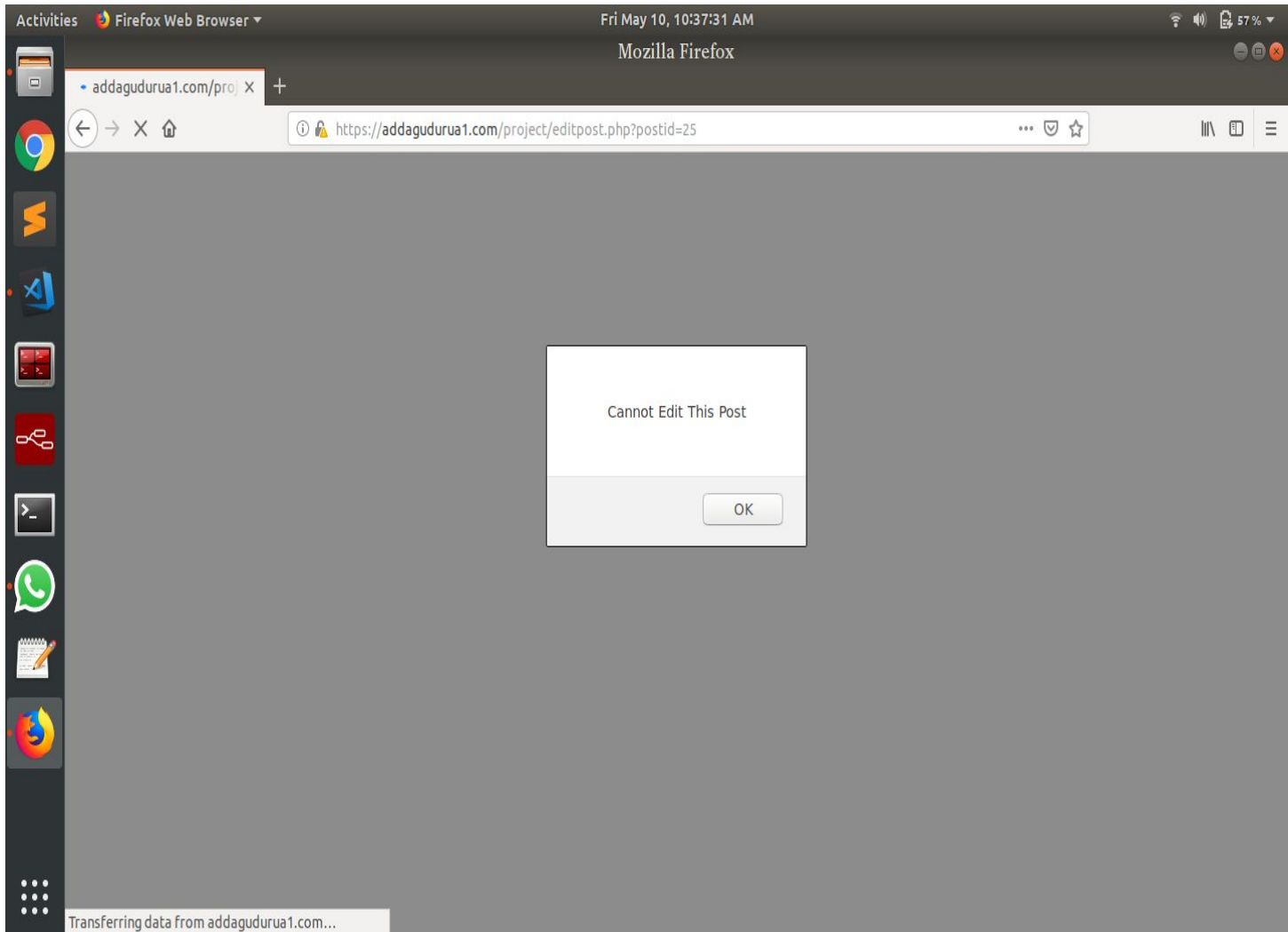
Update their post



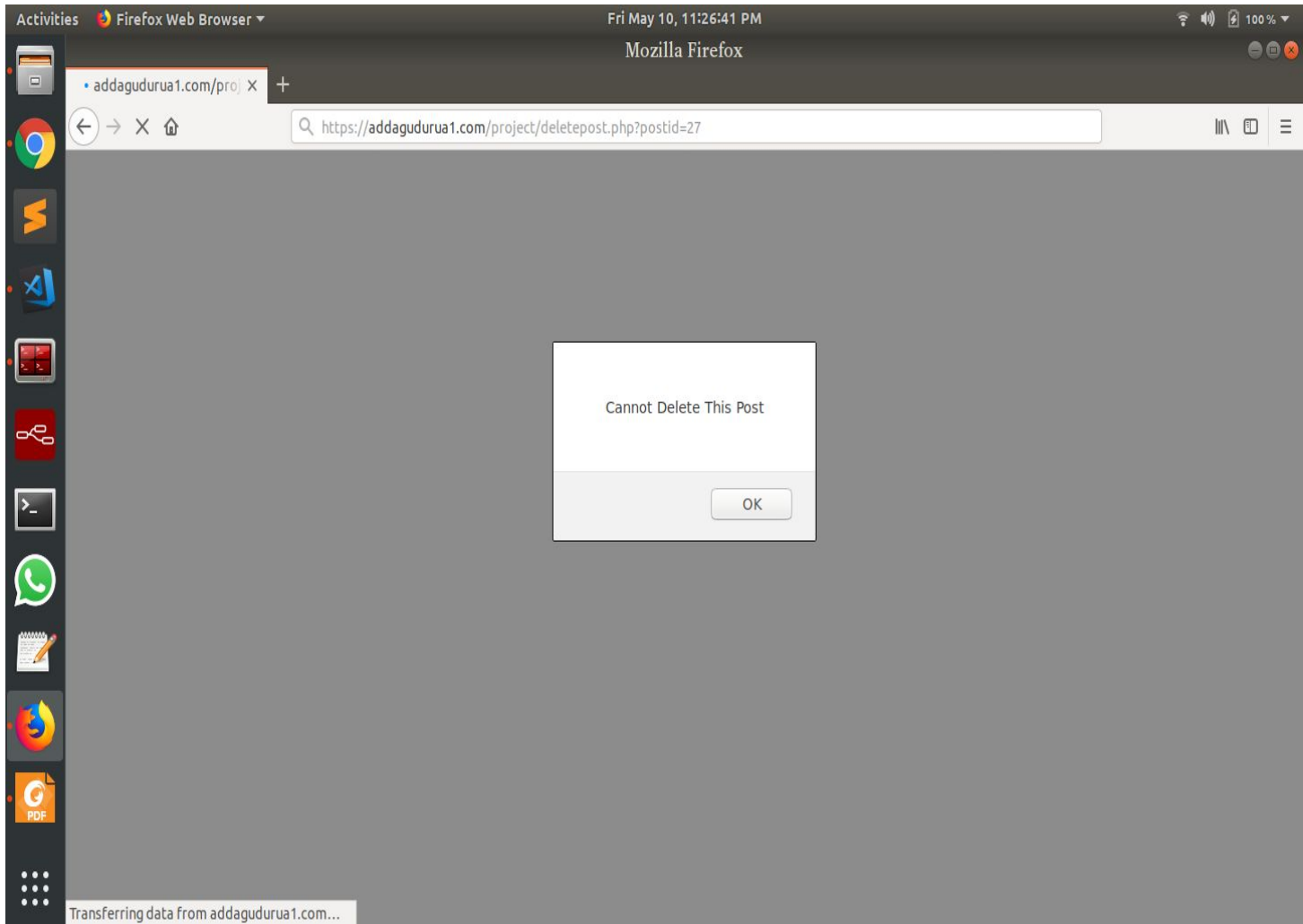
Updated post changes to DB



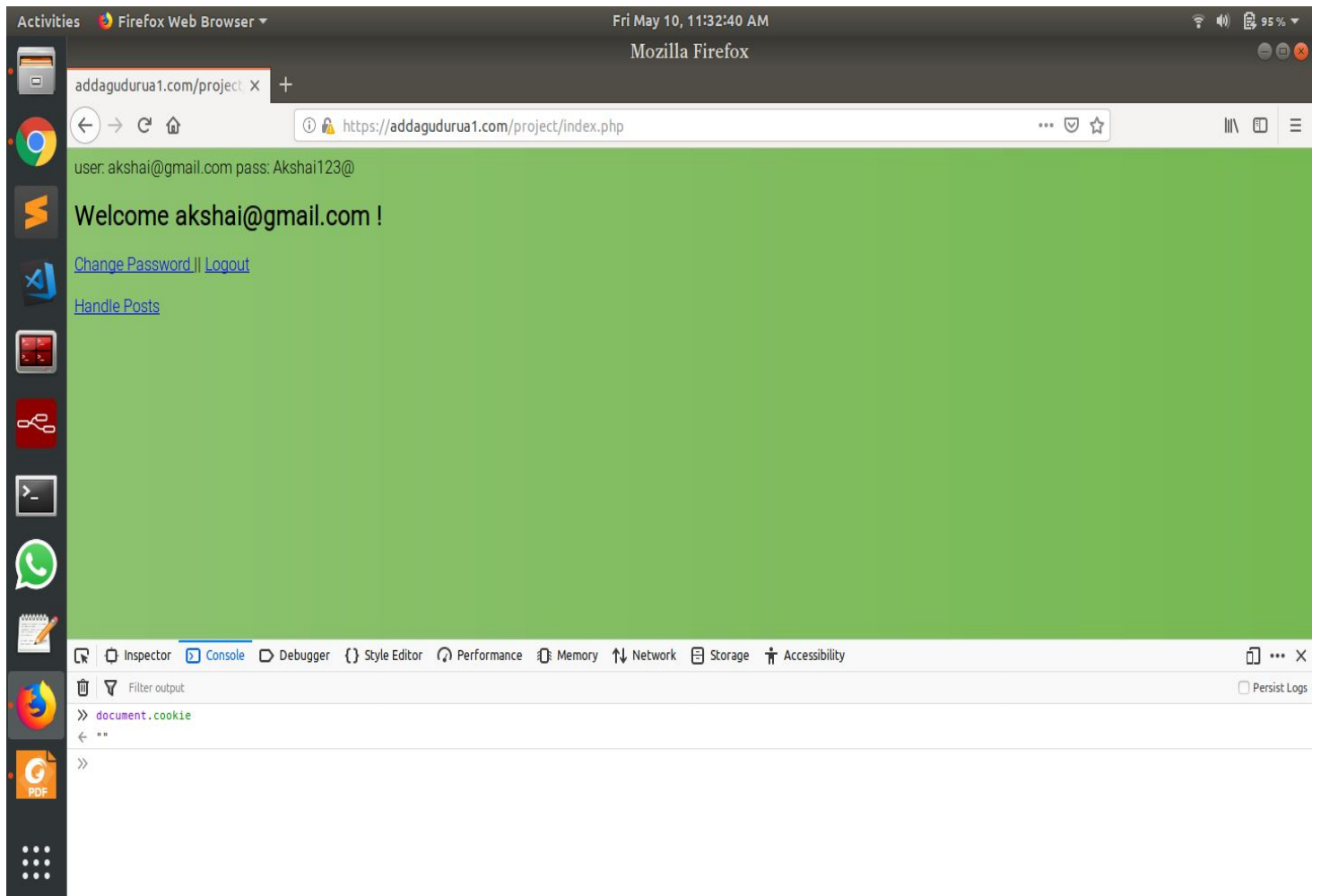
Cannot change others posts



Cannot delete others posts

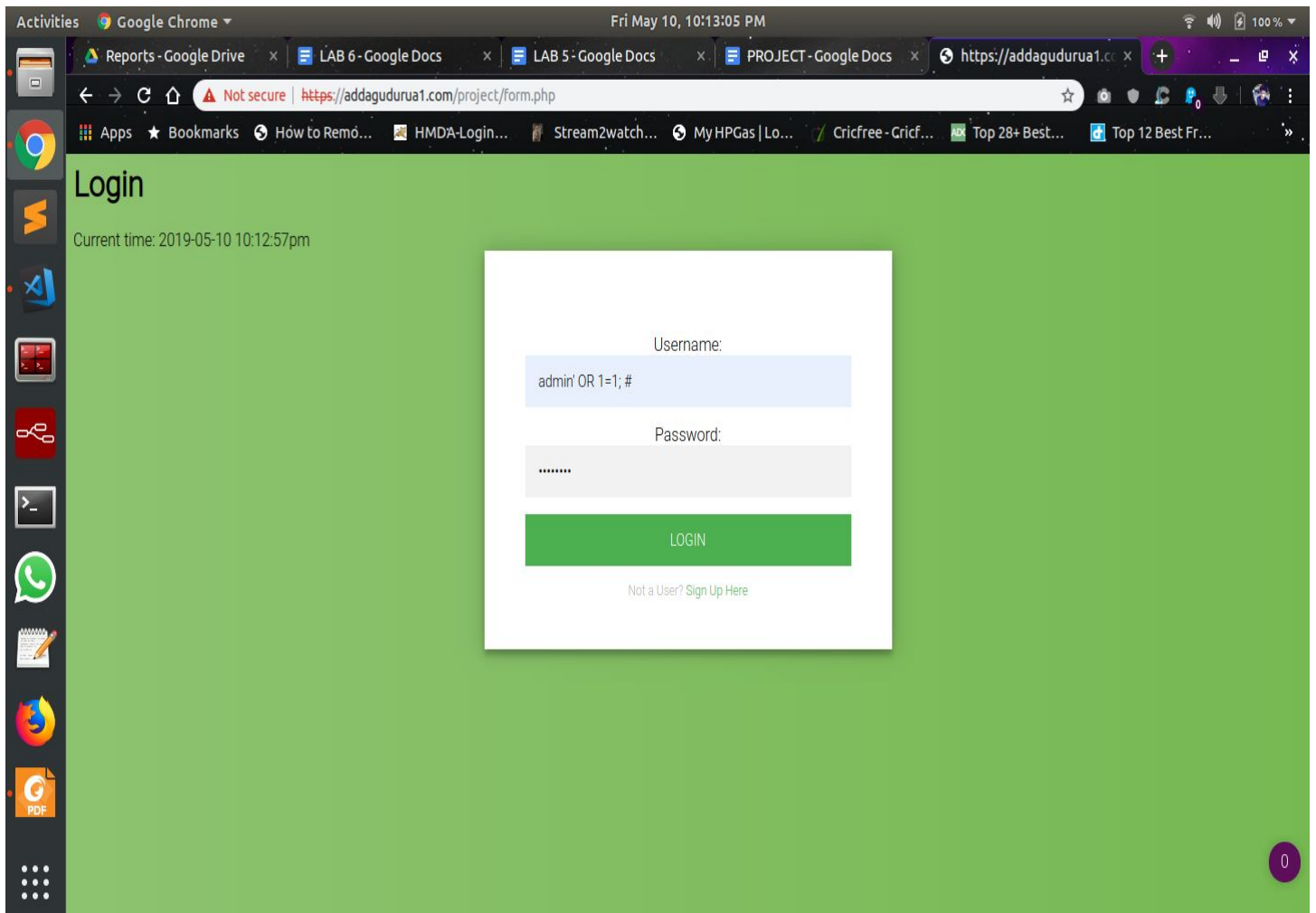


Hiding cookies in the running session to protect from session hijacking



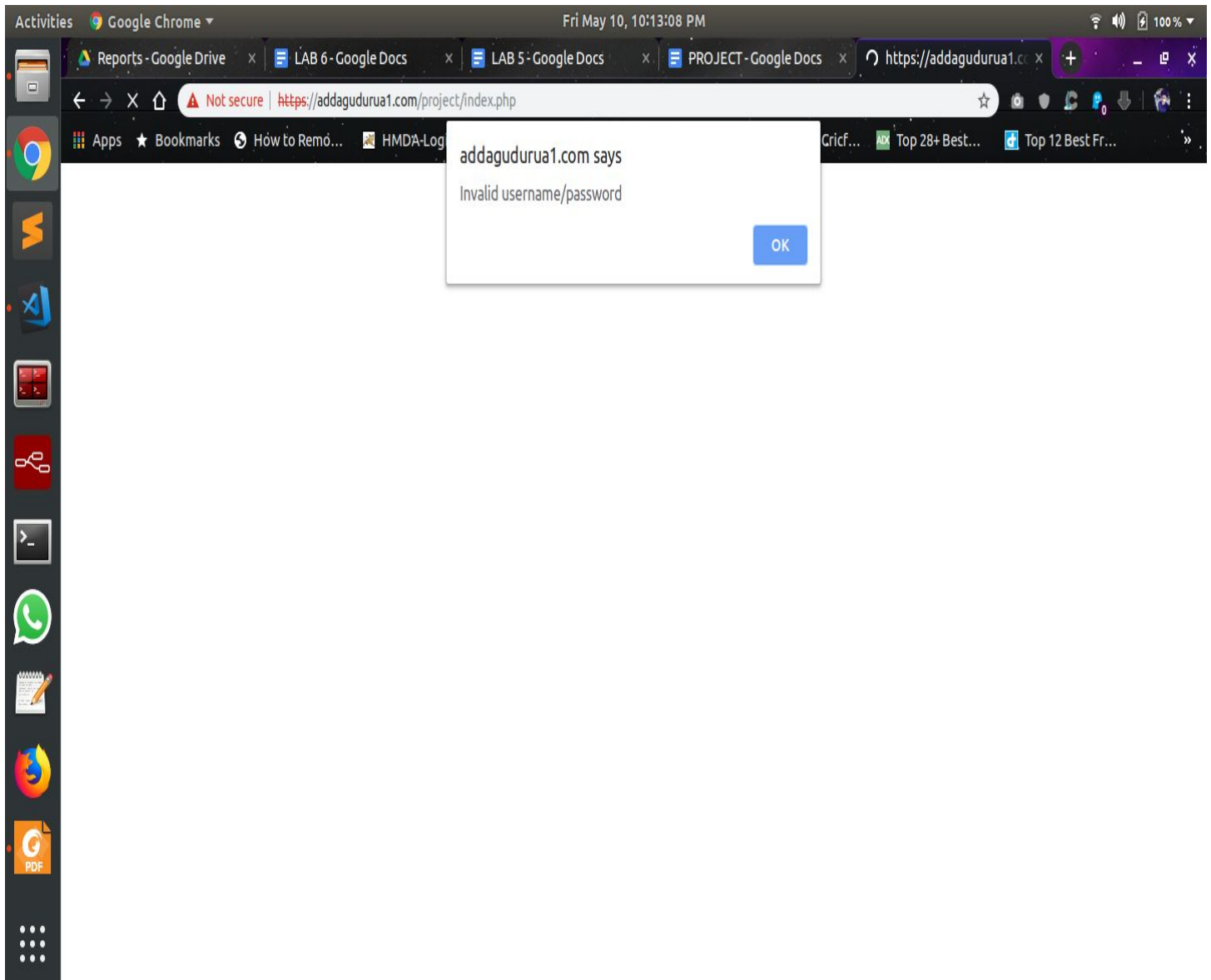
Since we set cookie parameters, we cannot find the cookie session value.

SQL Injection attack



We try to use this sql injection attack to login to the session without username or password.

Result of such attack



It throws an error saying invalid username or password. So we handled the error easily

Other session hijacking technique prevented even if session ID is traced as session ID changes on every few seconds and prevents hijacking

Activities

Google Chrome

Fri May 10, 10:15:01 PM

100%

Reports - Google Drive

LAB 6 - Google Docs

LAB 5 - Google Docs

PROJECT - Google Docs

https://addagudurua1.c...

Not secure

https://addagudurua1.com/project/index.php

☆

📷

🛡️

🔒

📄

🌐

⋮

Apps

★

Bookmarks

🔍

How to Remo...

🖼️

HMDA-Login...

📺

Stream2watch...

📶

My HPGas | Lo...

🏏

Cricfree - Cricf...

🏆

Top 28+ Best...

📱

Top 12 Best Fr...

»

user: akshai@gmail.com pass: Akshai123@

Welcome akshai@gmail.com !

[Change Password](#) || [Logout](#)

[Handle Posts](#)

🔍

📄

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Audits

⋮

✕

Web SQL

📁

Cookies

🌐

https://addagudurua1.c...

Cache

📁

Cache Storage

📁

Application Cache

Frames

▶

top

Filter

🔍

✕

| Name | Value | Domain | Path | Expires / ... | Size | HTTP | Secure | SameSite |
|-----------|----------------------------|-------------|----------|---------------|------|------|--------|----------|
| PHPSESSID | bs56q98r1d7ushq287p322khhc | .addagud... | /project | 2019-05-1... | 35 | ✓ | ✓ | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

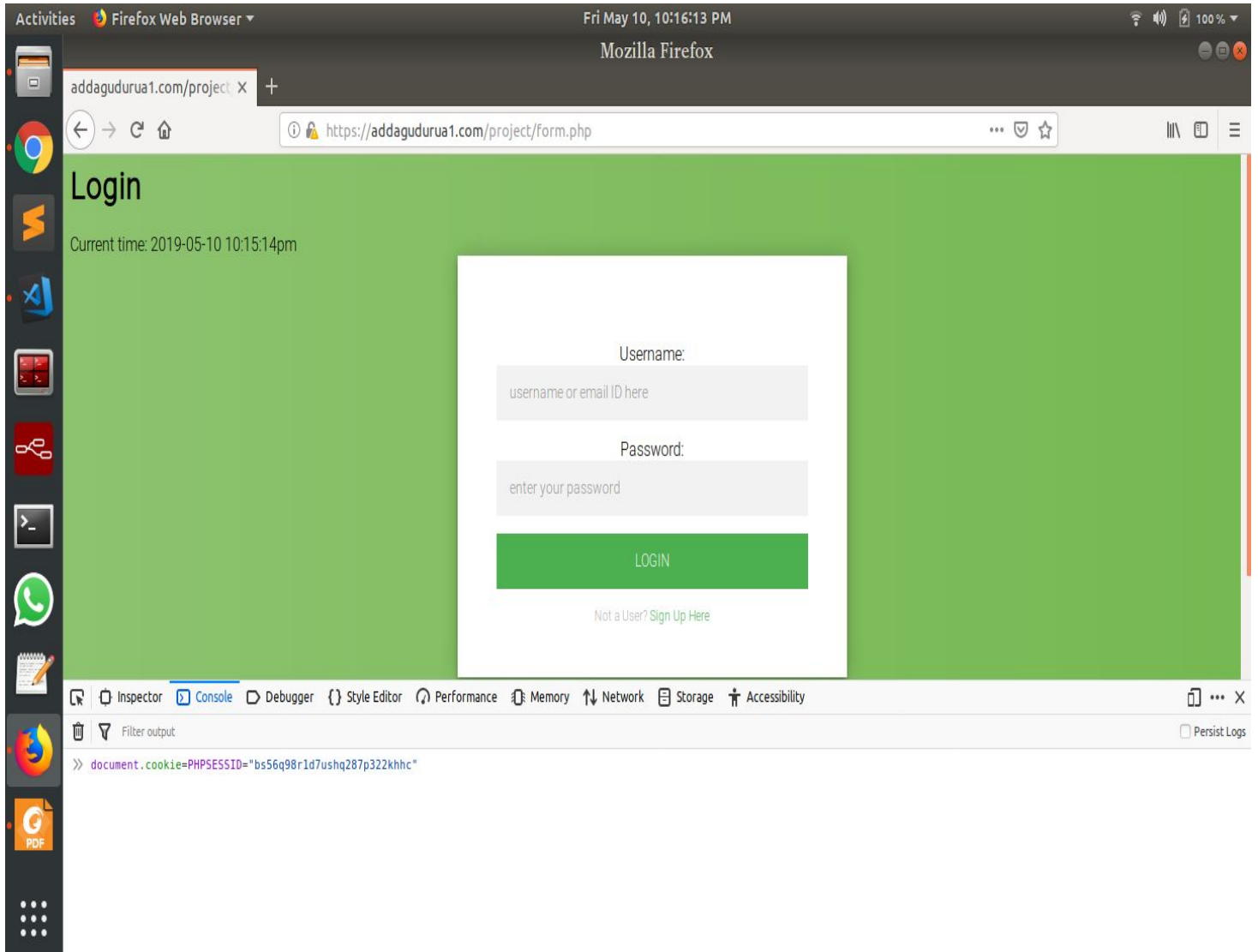
⋮

Console

What's New ✕

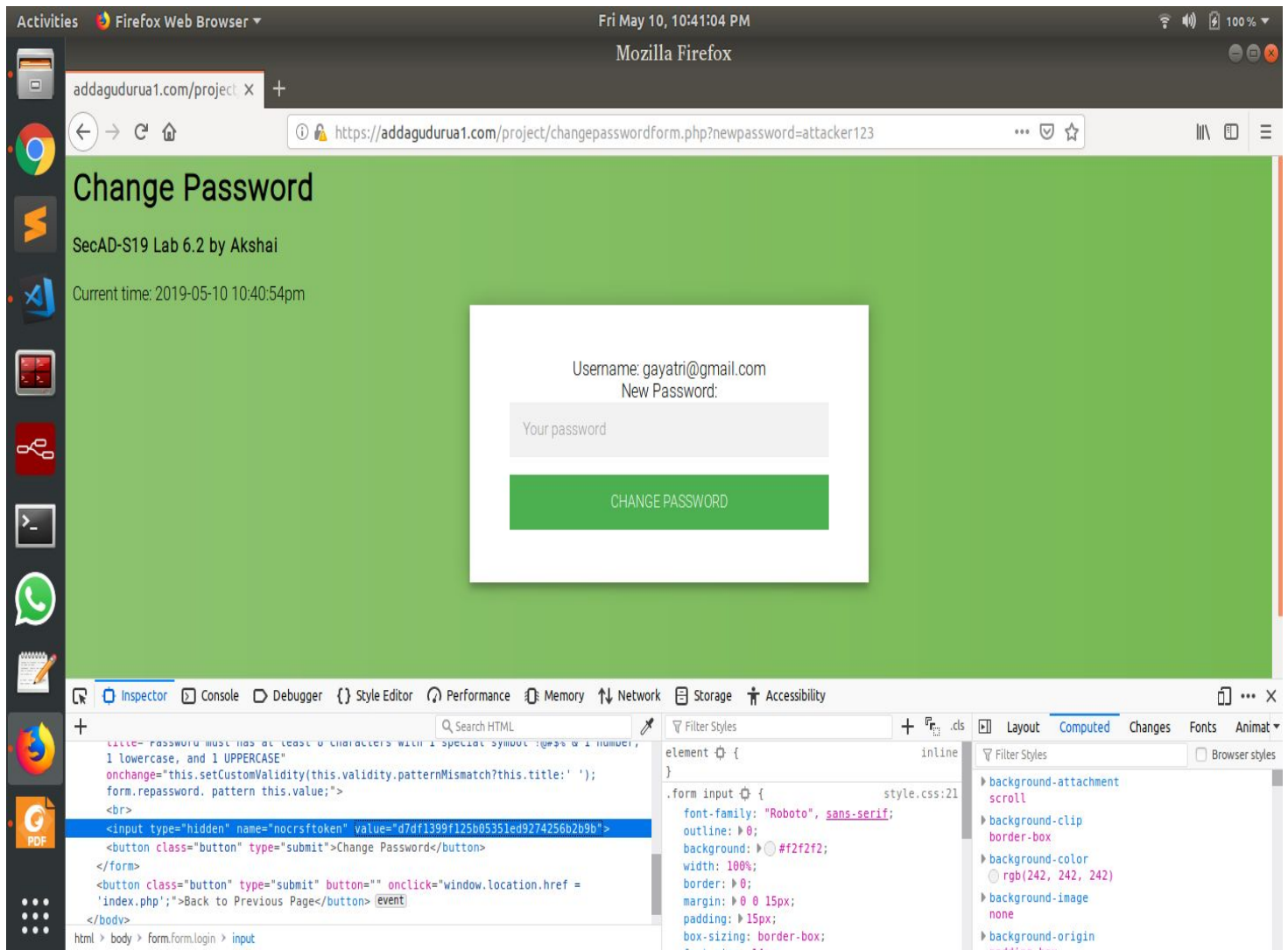
✕

Highlights from the Chrome 74 update



We can see that on refresh session is not hijacked or logged in

XSS and CSRF attack



Here session ID value changes on every page refresh or few seconds due to which the nocsrftoken helps in changing the token periodically and stops any possible attacks.

APPENDIX

README.md

#Secure App Development

##Spring 2019

###Akshai Addaguduru

###Department of Computer Science, University of Dayton

#Information

This is a private repository of Akshai Addaguduru, ID: 1016225580, email: addagudurua1@udayton.edu

This folder consists of the project files and report of SOCIAL NETWORKING WEB APPLICATION

Registration Form

```
<html>
    <h1>New User? Sign Up Here Now!</h1>

<head>
    <link rel="stylesheet" href="style.css">
</head>
<?php
//some code here
echo "Current time: " . date("m-d-Y h:i:sa")
?>

<div class="form">
    <form action="addnewuser.php" method="POST" class="form login">

        Username:<input type="text" class="text_field" name="username" required

pattern="^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$"
        title="Please enter a valid Username"
        placeholder="Your email address"
        onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:
'');" /> <br>

        Password: <input type="password" class="text_field" name="password" required

        pattern="^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$%^&])[\w!@#$%^&]{8,}$"
        placeholder="Your password"
        title="Password must has at least 8 characters with 1 special symbol !@#$%^&
1 number, 1 lowercase, and 1 UPPERCASE"
        onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:
''); form.repassword.pattern this.value;"/>

        Retype Password: <input type="password" class="text_field" name="repassword"
```

```
placeholder="Retype your password" required
title="Password does not match"
onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:
');"/> <br>

PhoneNo: <input type="text" name="phoneno"
pattern="^[+]*[(]{0,1}[0-9]{1,4}[)]{0,1}[-\s\.\/0-9]*$"
title="Enter a valid contact number"
placeholder="Your phone number"
onchange="this.setCustomValidity(this.validity.patternMismatch?this.title: '
');" /> <br>

<button class="button" type="submit">
    Sign Up!
</button>
    <p class="message">Already Registered?<a href="form.php"> LOGIN NOW </a>
</p>
<br>
</form>
</div>
</html>
```

Login Form

Form.php

```
<html>
    <h1>Login</h1>

<head>
    <link rel="stylesheet" href="style.css">
</head>
<?php
    //some code here
    echo "Current time: " . date("Y-m-d h:i:sa")
?>
    <form action="index.php" method="POST" class="form login">
        <div align="center" class="login-page">
            Username:<input type="text" placeholder="username or email ID here"
class="text_field" name="username" /> <br>
            Password: <input type="password" placeholder="enter your password"
class="text_field" name="password" /> <br>
            <button class="button" type="submit">
                Login
            </button>
            <p class="message">Not a User? <a href="registrationform.php"> Sign Up Here
</a> </p>
        </div>
    </form>
</html>
```

Addnewuser.php

```
<?php
    require "database.php";

    $username = $_POST["username"];
    $password = $_POST["password"];
    $phoneno   = $_POST["phoneno"];
    //$enabled  = $_POST["enabled"];

    if(!validateUsername($username) or !validatePassword($password)) {
        echo "TODO: error";
        die();
    }

    echo "DEBUG:addnewuser.php>username=$username; password=$password; phoneno=$phoneno
<br>";

    if(addnewuser($username,$password,$phoneno)) {
        echo "DEBUG:addnewuser.php> $username is added";
        //TODO: have a message
    } else {
        echo "DEBUG:addnewuser.php> $username cannot be added";
    }

function validateUsername($username)
{
    //TODO: validate the username
    return TRUE;
}

function validatePassword($password)
{
    //TODO: validate the password
```

```

        return TRUE;
    }

function validatePhoneNo($phoneno)
{
    //TODO: validate the phoneno
    return TRUE;
}

/*function validateEnabled($enabled)
{
    //TODO: validate the phoneno
    return TRUE;
}*/

//<a href="form.php"> Hurray! You are signed up! Login Now </a>

?>

<html>

<head>
    <link rel="stylesheet" href="style.css">
</head>
    <button class="button" type="submit" button onclick="window.location.href =
'form.php';" >
        REGISTERED! LOGIN NOW
    </button>
</html>

```

Index.php

```
<?php
    $lifetime =15 * 60;
    $path = "/project";
    $domain = "addagudurual.com";
    $secure = TRUE;
    $httponly = TRUE;
    session_set_cookie_params($lifetime, $path, $domain, $secure, $httponly);
    session_start();

    if (isset($_POST["username"]) and isset($_POST["password"])) {
        if (securechecklogin($_POST["username"], $_POST["password"])) {
            $_SESSION["logged"] = TRUE;
            $_SESSION["username"] = $_POST["username"];
            $_SESSION["browser"] = $_SERVER["HTTP_USER_AGENT"];
        }
        else {
            echo "<script>alert('Invalid username/password');</script>";
            unset($_SESSION["logged"]);
            header("Refresh:0; url=form.php");
            die();
        }
    }

    if (!isset($_SESSION["logged"]) or $_SESSION["logged"] != TRUE) {
        echo "<script>alert('You've not logged in yet. Login first!');</script>";
        header("Refresh:0; url=form.php");
        die();
    }

    if ($_SESSION["browser"] != $_SERVER["HTTP_USER_AGENT"]) {
        echo "<script>alert('WARNING! Session Hijacking Detected');</script>";
        header("Refresh:0; url=form.php");
        die();
    }
    $username = $_SESSION["username"];
```

```

?>
<html>
<head>
    <link rel="stylesheet" href="style.css">
</head>

</form>
<h2> Welcome <?php echo htmlentities($_SESSION["username"]); ?> !</h2>
    <a href="changepasswordform.php"> Change Password </a> || <a href="logout.php">
Logout </a> <br> <br>
    <a href="addpost.php"> Handle Posts </a>
</form> </html>

<?php

function securechecklogin($username, $password) {
    echo "user: $username pass: $password";
    $mysqli = new mysqli('localhost', 'akshai', 'akki', 'secad_s19');
    if($mysqli->connect_errno) {
        printf("Connection Failed: %s\n", $mysqli->connect_error);
        exit();
    }

    $prepared_sql = "SELECT * FROM users WHERE username=?" . " AND
password=password(?)";
    if(!$stmt = $mysqli->prepare($prepared_sql))
        echo "Prepared Statement Error";
    $stmt->bind_param("ss", $username, $password);
    if(!$stmt->execute()) echo "Execute Error";
    if(!$stmt->store_result()) echo "Store_result Error";
    $result = $stmt;
    if($result->num_rows ==1){
        return true;
    }
    return false;
}

?>

```


Changepasswordform.php

```
<?php
    require "session_auth.php";
    $rand=bin2hex(openssl_random_pseudo_bytes(16));
    $_SESSION["nocsrftoken"]=$rand;
?>

<html>

<head>
    <link rel="stylesheet" href="style.css">
</head>
    <h1>Change Password</h1>
    <h4>SecAD-S19 Lab 6.2 by Akshai</h4>
<?php
    //some code here
    echo "Current time: " . date("Y-m-d h:i:sa")
?>

<form action="changepassword.php" method="POST" class="form login">
    Username:<!--input type="text" class="text_field" name="username" /-->
    <?php echo htmlentities($_SESSION["username"]); ?> <br>

    New Password: <input type="password" class="text_field" name="newpassword"
    pattern="^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$$%^&])[\w!@#$$%^&]{8,}$"
    placeholder="Your password"
        title="Password must has at least 8 characters with 1 special symbol !@#$$%^& 1
number, 1 lowercase, and 1 UPPERCASE"
        onchange="this.setCustomValidity(this.validity.patternMismatch?this.title:' ');
form.repassword. pattern this.value;"

    /> <br>

    <input type="hidden" name="nocsrftoken" value="<?php echo $rand; ?>" />
    <button class="button" type="submit">
        Change Password
    </button>
```

```
</form>
```

```
    <button class="button" type="submit" button onclick="window.location.href =  
'index.php';">
```

```
    Back to Previous Page
```

```
</button>
```

```
</html>
```

Changepassword.php

```
<?php

require "session_auth.php";
require 'database.php';

$username = $_SESSION["username"];
$newpassword = $_REQUEST["newpassword"];

$nocsrftoken = $_REQUEST["nocsrftoken"];
if(!isset($nocsrftoken) OR ($nocsrftoken!= $_SESSION['nocsrftoken'])) {
    echo "<script>alert('Cross-site request forgery is detected!');</script>";
    header("Refresh:0; url=logout.php");
    die();
}

if(isset($username) AND isset($newpassword)) {
    echo "DEBUG: changepassword.php->Got: username=$username;
newpassword=$newpassword;<br>";
    if(changepassword($username,$newpassword)) {
        echo "<h4>The new password has been set.</h4>";
    } else
    {
        echo "<h4>Error: Cannot change the password.</h4>";
    }
}
else {
    echo "No provided username/password to change";
    exit();
}

?>

<a href="index.php">Home</a> | <a href="logout.php">Logout</a>

<html>

<head>
```

```
    <link rel="stylesheet" href="style.css">
</head>
</html>
```

Logout.php

```
<?php

session_start();
session_destroy();
?>

<p>Logged Out!</p>

<a href="form.php">Login Again</a>

<html>
<head>
    <link rel="stylesheet" href="style.css">
</head>
</html>
```

Addpost.php

```
<?php

require 'database.php';
require 'session_auth.php';

$rand=bin2hex(openssl_random_pseudo_bytes(16));
$_SESSION["nocsrftoken"]=$rand;

echo date("r");

if(isset($_POST["newpost"]))
{
    addnewpost($_SESSION["username"], $_POST["newpost"], date("r"));
    echo "New POST is added";
}

?>

<html>
<head>
    <link rel="stylesheet" href="style.css">
</head>

<form action="addpost.php" method="POST" class="addpost">
    <input type="hidden" name="nocsrftoken" value="<?php echo $rand;?>"/>
    <!--Status Box--> <br>
    <h4>Post a Status Here</h4> <textarea name="newpost" rows="10" cols="50"> </textarea>
<br>
    <button class="button" type="submit" >
        Post
    </button>

</form>
```

```
<button class="button" type="submit" button onclick="window.location.href =  
'getposts.php';" >  
View  
</button>
```

```
<button class="button" type="submit" button onclick="window.location.href =  
'index.php';">  
Back  
</button>
```

```
</html>
```

Getposts.php

```
<?php
require 'session_auth.php';
require 'database.php';

$rand=bin2hex(openssl_random_pseudo_bytes(16));
$_SESSION["nocsrftoken"]=$rand;

$username=$_SESSION['username'];

if(isset($_POST["editpost"]))
{
    viewposts($_SESSION["username"], $_POST["newpost"], date("r"));
    echo "New POST is added";
}

$mysqli = new mysqli('localhost','akshai','akki','secad_s19');
if($mysqli->connect_errno) {
    printf("Connection Failed: %s\n",$mysqli->connect_error);
    exit();
}

$prepared_sql = "SELECT postID,postmsg FROM posts;";
$result = $mysqli->query($prepared_sql);
/*    $result = mysqli_num_rows($result);*/

if ($result->num_rows > 0) {
    echo "<h4> Your Statuses </h4> <br>";
    while($row=$result->fetch_assoc())
    {
        echo "<tr><td>".$row["postID"]. "</td><td>".$row["postmsg"]. " </td></tr><br>";
        //echo $row['content'] . "<br/>";
        echo "        <input        type='button'        name='editpost'        value='Edit'
onclick=\"window.location.href = 'editpost.php?postid=".$row["postID"]."'\> ";
    }
}
```



```
        echo      "<input      type='button'      name='editpost'      value='Delete'
onclick=\"window.location.href      =      'deletePost.php?postId=".$row["postID"]."'\>      <br>
<br>";
```

```
        /* echo      "<input      type='button'      name='comment'      value='COMMENT      THIS      POST'
onclick=\"window.location.href      =      'comment.php?postId=".$row["postID"]."'\>      <br>      <br>";
*/
```

```
    }
}
```

```
?>
```

```
<html>
```

```
<head>
```

```
    <link      rel="stylesheet"      href="style.css">
```

```
</head>
```

```
<br>
```

```
<button      class="button"      type="submit"      button      onclick="window.location.href      =
'addpost.php';"      >
```

```
    GO      BACK
```

```
</button>
```

```
<button      class="button"      type="submit"      button      onclick="window.location.href      =
'index.php';"      >
```

```
    HOME
```

```
</button>
```

```
</html>
```

Editpost.php

```
<?php

require 'database.php';
require 'session_auth.php';

$rand=bin2hex(openssl_random_pseudo_bytes(16));
$_SESSION["nocsrftoken"]=$rand;

$postid = $_GET['postid'];

$mysqli = new mysqli('localhost','akshai','akki','secad_s19');
if($mysqli->connect_errno) {
    printf("Connection Failed: %s\n",$mysqli->connect_error);
    exit();
}

$prepared_sql = "SELECT * FROM posts WHERE postID = ? ";
$stmt = $mysqli->prepare($prepared_sql);
$stmt->bind_param('i', $postid);
$stmt->execute();
$result = $stmt->get_result();
$row=mysqli_fetch_assoc($result);
$username = $row["username"];
$postmsg=$row["postmsg"];

if($username!= $_SESSION["username"]) {
    echo"<script>alert('Cannot Edit This Post');</script>";
    header("Refresh:0; url='getposts.php'");
    die();
}
else{
    if(isset($editpost)){
        echo "Updating Post for $username";
        echo "<br>";
        if(updateposts($postid,$editpost))
```

```
    echo "Post is Updated";
else
    echo "Cannot Update the Post";
}
```

```
}
```

```
?>
```

```
<html>
```

```
<head>
```

```
    <link rel="stylesheet" href="style.css">
```

```
</head>
```

```
    <h3>Update Your Post</h3>
```

```
    <a href="index.php">Home</a>
```

```
<?php
```

```
//some code here
```

```
$postid = $_REQUEST['postid'];
```

```
//echo "Current time: " . date("Y-m-d h:i:sa");
```

```
?>
```

```
<form action="updateposts.php" method="POST" class="form login">
```

```
<br>
```

```
<input type="hidden" name="nocsrftoken" value="<?php echo $rand;?>"/>
```

```
<input type="hidden" name="postid" value="<?php echo $postid;?>"/>
```

```
<br>
```

```
<input type="textarea" name="editpost" required cols="2" rows="2" title="Update post"
value="<?php echo $postmsg;?>"/>
```

```
<br>
```

```
<button class="button" type="submit">
```

```
Update Post
```

```
</button>
```

```
</form> </html>
```

Session_auth.php

```
<?php
    $lifetime =15 * 60;
    $path = "/lab6";
    $domain = "addagudurual.com";
    $secure = TRUE;
    $httponly = TRUE;
    session_set_cookie_params($lifetime, $path, $domain, $secure, $httponly);
    session_start();

    if (!isset($_SESSION["logged"]) or $_SESSION["logged"] != TRUE) {
        echo "<script>alert('You've not logged in yet. Login first!');</script>";
        header("Refresh:0; url=form.php");
        die();
    }

    if ($_SESSION["browser"] != $_SERVER["HTTP_USER_AGENT"]) {
        echo "<script>alert('WARNING! Session Hijacking Detected');</script>";
        header("Refresh:0; url=form.php");
        die();
    }

?>
```

Updateposts.php

```
<?php
```

```
require 'database.php';
require 'session_auth.php';

$rand=bin2hex(openssl_random_pseudo_bytes(16));
$_SESSION["nocsrftoken"]=$rand;
$postid = $_POST['postid'];
$editpost = $_POST['editpost'];

echo "post: $postid newpost: $editpost";
updatePost($postid, $editpost);
header("Refresh:0; url=editpost.php?postid=".$postid);
die();
```

```
?>
```

Deleteposts.php

```
<?php
    $postid = $_GET["postid"];
    require "database.php";

    deletePost($postid);
    header("Refresh:0; url=getposts.php");
    die();
?>
```

Database.php

```
<?php
```

```
$mysqli = new mysqli('localhost', 'akshai', 'akki', 'secad_s19');
if($mysqli->connect_errno) {
    printf("Connection Failed: %s\n", $mysqli->connect_error);
    exit();
}

function changepassword($username, $newpassword) {
    global $mysqli;
    //echo "user: $username pass: $newpassword";
    $prepared_sql = "UPDATE users SET password=password(?) WHERE username=?";
    echo "DEBUG>prepared_sql=$prepared_sql\n";

    if(!$stmt = $mysqli->prepare($prepared_sql))
        return FALSE;

    $stmt->bind_param("ss", $newpassword, $username);

    if(!$stmt->execute())
        return FALSE;

    return TRUE;
}

function addnewuser($username, $newpassword, $phoneno) {
    global $mysqli;
    $prepared_sql = "INSERT INTO users VALUES (?, password(?), ?)";
    echo "DEBUG>addnewuser.php->addnewuser->prepared_sql=";

    if(!$stmt = $mysqli->prepare($prepared_sql))
        return FALSE;
```

```

$stmt->bind_param("sss",$username,$newpassword,$phoneno);

if(!$stmt->execute())
    return FALSE;

return TRUE;

}

function addnewpost($username, $postmsg, $date) {
    global $mysqli;
    $prepared_sql = "INSERT INTO posts (postmsg, username, post_date) VALUES (?,?,?);";
    echo "DEBUG>addnewpost->prepared_sql= $prepared_sql";
    if(!$stmt = $mysqli->prepare($prepared_sql))
        return FALSE;
    $stmt->bind_param("sss", $postmsg, $username, $date);

    if(!$stmt->execute())
        return FALSE;
    return TRUE;
}

function viewposts() {
    global $mysqli;
    $prepared_sql = "SELECT * FROM posts";
    $result = $mysqli->query($prepared_sql);
    if($result->num_rows>0) {
        //output data of each row
        while($row=$result->fetch_assoc()) {
            $postid = $row["postID"];
            echo "<h3>Post ". $postid . "</h3>";
            echo $row["newpost"]. "<br>";
        }
    } else {
        echo "No post in this blog yet <br>";
    }
}

```



```

        echo "<hr>";
    }

}

function updatePost($postid, $postmsg) {
    global $mysqli;
    $prepared_sql = "UPDATE posts SET postmsg=? WHERE postID=?";

    if(!$stmt = $mysqli->prepare($prepared_sql))
        return FALSE;

    $stmt->bind_param("si", $postmsg, $postid);

    if(!$stmt->execute())
        return FALSE;

    return TRUE;
}

function deletePost($postid) {
    global $mysqli;
    $prepared_sql = "DELETE FROM posts WHERE postID=?";

    if(!$stmt = $mysqli->prepare($prepared_sql))
        return FALSE;

    $stmt->bind_param("i", $postid);

    if(!$stmt->execute())
        return FALSE;

    return TRUE;
}

?>

```

Database.sql

```
DROP TABLE IF EXISTS `users`;
```

```
CREATE TABLE `users` (  
  `username` varchar(50) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  `phonenumber` int(20) NOT NULL,  
  PRIMARY KEY (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
LOCK TABLES `users` WRITE;
```

```
INSERT INTO `users` VALUES  
('', '', '', 0, NULL, NULL), ('akshai@gmail.com', '*E9FA200FBB9766346C0925B8F4EEC599F6F8CA9E',  
2147483647)
```

```
UNLOCK TABLES;
```

```
DROP TABLE IF EXISTS `posts`;
```

```
CREATE TABLE `posts` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `text` varchar(1000) NOT NULL,  
  `username` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `posts_ibfk_1` (`username`),  
  CONSTRAINT `posts_ibfk_1` FOREIGN KEY (`username`) REFERENCES `users` (`username`) ON  
DELETE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=latin1;
```

```
LOCK TABLES `posts` WRITE;
```

```
INSERT INTO `posts` VALUES (3,'bittu','hi i am fine','bittu',NULL),(6,'Isnt it a beautiful day','akshai@gmail.com',NULL)
```

```
UNLOCK TABLES;
```

Style.css

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.login-page {
  width: 360px;
  padding: 8% 0 0;
  margin: auto;
}

.form {
  position: relative;
  z-index: 1;
  background: #FFFFFF;
  max-width: 360px;
  margin: 0 auto 100px;
  padding: 45px;
  text-align: center;
  box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

.form input {
  font-family: "Roboto", sans-serif;
  outline: 0;
  background: #f2f2f2;
  width: 100%;
  border: 0;
  margin: 0 0 15px;
  padding: 15px;
  box-sizing: border-box;
  font-size: 14px;
}

.form button {
  font-family: "Roboto", sans-serif;
  text-transform: uppercase;
  outline: 0;
```

```
background: #4CAF50;
width: 100%;
border: 0;
padding: 15px;
color: #FFFFFF;
font-size: 14px;
-webkit-transition: all 0.3 ease;
transition: all 0.3 ease;
cursor: pointer;
}

.form button:hover, .form button:active, .form button:focus {
background: #43A047;
}

.form .message {
margin: 15px 0 0;
color: #b3b3b3;
font-size: 12px;
}

.form .message a {
color: #4CAF50;
text-decoration: none;
}

.form .register-form {
display: none;
}

.container {
position: relative;
z-index: 1;
max-width: 300px;
margin: 0 auto;
}

.container:before, .container:after {
content: "";
display: block;
clear: both;
}

.container .info {
```

```
margin: 50px auto;
text-align: center;
}
.container .info h1 {
margin: 0 0 15px;
padding: 0;
font-size: 36px;
font-weight: 300;
color: #1a1a1a;
}
.container .info span {
color: #4d4d4d;
font-size: 12px;
}
.container .info span a {
color: #000000;
text-decoration: none;
}
.container .info span .fa {
color: #EF3B3A;
}
body {
background: #76b852; /* fallback for old browsers */
background: -webkit-linear-gradient(right, #76b852, #8DC26F);
background: -moz-linear-gradient(right, #76b852, #8DC26F);
background: -o-linear-gradient(right, #76b852, #8DC26F);
background: linear-gradient(to left, #76b852, #8DC26F);
font-family: "Roboto", sans-serif;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
```

