**WAPH-Web Application Programming and Hacking**

**Instructor: Dr. Phu Phung**

**Mini Facebook**

**Team – 01**

**Team members**

1. Tulasiram Nakkanaboina, [nakkantm@mail.uc.edu](mailto:nakkantm@mail.uc.edu)

2. Vihasith Rasala, [rasalavh@mail.uc.edu](mailto:rasalavh@mail.uc.edu)

3. Grahika Rampudi, [Rampudga@mail.uc.edu](mailto:Rampudga@mail.uc.edu)

4. Sai Sandeep Pasham, [pashamsp@mail.uc.edu](mailto:pashamsp@mail.uc.edu)

*Team members respective headshots*

A person with a mustache and a black shirt

Description automatically generatedA person wearing glasses and a suit jacket

Description automatically generated A person with long hair wearing glasses

Description automatically generated A person in a blue shirt

Description automatically generated

**Project Management Information**

Source code repository (private access):[**Click\_here**](https://github.com/waph-team01/waph-teamproject)

Project homepage (public): [**Click\_here**](https://github.com/waph-team01/waph-team01.github.io)

**Revision History**

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 26/03/2024 | 0.0 | Initial draft |
| 26/03/2024 | 0.1 | Drafting template overview |
| 26/03/2024 | 1.0 | Sprint 1 update |

**Overview**

This Project aims to develop a Mini Facebook web application using full stack web

development technologies, secure programming/hacking principles, and the practices of

agile development. This is a team project where all the team members collaborated and

worked together to create a database, creating simple login page and created an index.html

containing information about course overview, each member headshot and each member

personal portfolio link. We divided the tasks among us and pushed the code files into the

main branch.

**System Analysis**

During the initial phase that is for the sprint 0 of the project, we constructed the website utilizing the system configuration file, which was utilized and modified for the WAPH-team project file. The hosts file in the etc folder was updated with the public URL: [waph-team01.minifacebook.com](https://waph-team01.minifacebook.com/) alongside the IP address.

**Demo screenshots**

A screenshot of a computer

Description automatically generated

Figure 1 login admin page

A screenshot of a computer

Description automatically generated

Figure 2 change password form

A screenshot of a computer

Description automatically generated

Figure 3 password changed successfully

A screenshot of a computer

Description automatically generated

Figure 4 edit user profile

A screenshot of a computer

Description automatically generated

Figure 5 update profile

A screenshot of a computer

Description automatically generated

Figure 6 login page

A screenshot of a computer

Description automatically generated

Figure 7 new user registration form

A screenshot of a computer

Description automatically generated

Figure 8 succesful registation form

**Functional Requirements**

* **Features for posting like images and texts:** Enabling users to create and share posts with images/texts.
* **Database setup:** Configuring database system for storing application data securely.
* **Notification add on:** Implementing notification system for relevant user alerts.
* **Forms for authentication and user registration:** Implementing secure login and registration forms for user access.
* **Search functionality:** Enabling users to search for specific content within application.
* **Messaging features:** Integrating private messaging functionalities for user communication.

**Non - Functional Requirements**

* **Error handling:** Managing errors for smoother user experience and application stability.
* **Implementing security features:** Strengthening application security through implementation of various security measures.
* **Cookies:** Utilizing cookies for managing session information and enhancing user experience.
* **Session information:** Handling session data to maintain user context and interactions.

**Database**

A screenshot of a computer

Description automatically generated

Figure 9 tables which stores the data

A screenshot of a computer

Description automatically generated

Figure 10 user tables

**User Interface**

**Front-End:**

**Edit Profile Feature:** Users can modify their full name, email, and phone number.

**Change Password Option:** Users can change their password if facing any issues.

**Posts Visibility Setting:** Upon logging in, users can view site's posted content.

**Back-End:**

**Database Creation:** Established a database with essential tables for the application.

**Users Table:** Stores user information including full name, username, password, email, and phone number.

**Posts Table:** Records posts from various users, containing fields like Post ID, Title, Content, Post Date, and Owner.

**Database Integration:** Connected the database with the front end to save user inputs.

Implementation

**Security Analysis**

**Scrum Process**

**Sprint 0**

Duration: 22/03/2024-03/31/2024

*Completed Tasks:*

1. Our team has successfully established both public and private repositories

named “waph-teamproject” and “waph-team01.github.io” correspondingly.

2. SSL certificates and keys for the team project have been generated.

3. HTTPS has been configured within the local domain name to ensure secure

communication.

4. A database has been set up for the system.

5. Individual home pages have been developed for all team members, and

their respective lab files have been transferred to the team directory.

6. Testing of the index.html page has been carried out, yielding the anticipated

performance and functionality as observed during the tests.

*Contributions:*

1. Tulasiram Nakkanaboina, 11 commits, 7 hours, contributed in generating

the SSL keys and certificates.

2. Vihasith Rasala, 8 commits, 6 hours, contributed in documentation of

README file and organizing the data.

5

3. Grahika Rampudi, 8 commits, 7 hours, contributed in creating the personal

page of our team which includes details of our team members.

4. Sai Sandeep Pasham, 7 commits, 6 hours, contributed in creating the

database-data.sql, index.html page which includes the details of admin

login.

**Sprint 1**

Duration: 03/28/2024-04/01/2024

*Completed Tasks:*

1. Established the database according to the outlined criteria.
2. Developed both login and registration forms.
3. Executed fundamental features for authenticated users, such as password modification.

*Contributions:*

1. Tulasiram Nakkanaboina, 11 commits, 7 hours, contributed in creating user registration and profile edit php files snd further developed index.php file

2. Vihasith Rasala, 15 commits, 6 hours, contributed in documentation of

README file and organizing the data. Created and developed logout php file.

3. Grahika Rampudi, 8 commits, 7 hours, contributed in creating and organizing the database and it’s implementation in back end of web application.

4. Sai Sandeep Pasham, 7 commits, 6 hours, contributed in creating the change password and login php files.

**Spring Retrospection**

* The main problem we faced for a team call is unmatched schedule of team members. We are usually doing our team meetings in late nights.
* Online calls didn’t meet the expected progress of the project, so we are making ourselves available for physical sessions which is difficult.
* Proper collaborative dashboards require premium subscriptions for third party applications.
* Sandbox doesn’t work properly which is a dot in a white page.
* Team members use different OS like MacOS or windows where it’s difficult to work in virtual machine setup if it’s a Macbook.

**Appendix**

**Form.php:**

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

Username: <input type="text" class="text\_field" name="username" /><br>

Password: <input type="password" class="text\_field" name="password" /><br>

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

Login

</button>

</form>

<button class="button" onclick="window.location.href='registrationform.php';">

New User Registration

</button>

**Index.php:**

<?php

session\_set\_cookie\_params(15\*60, "/", "waph-team01.mini.facebook.com", TRUE, TRUE);

session\_start();

require "database.php";

if (isset($\_POST["username"]) && isset($\_POST["password"])) {

$username = $\_POST["username"];

$password = $\_POST["password"];

if (checklogin\_mysql($username, $password)) {

$\_SESSION["authenticated"] = TRUE;

$\_SESSION["username"] = $username;

$\_SESSION["browser"] = $\_SERVER["HTTP\_USER\_AGENT"];

} else {

session\_destroy();

echo "<script>alert('Invalid Username or password please recheck');window.location='form.php';</script>";

die();

}

}

if (!isset($\_SESSION["authenticated"]) || $\_SESSION["authenticated"] != TRUE) {

session\_destroy();

echo "<script>alert('You have not logged in. Please login first');</script>";

header("Refresh:0; url=form.php");

die();

}

if ($\_SESSION["browser"] != $\_SERVER["HTTP\_USER\_AGENT"]) {

session\_destroy();

echo "<script>alert('Session hijack detected')</script>";

header("Refresh:0; url=form.php");

die();

}

?>

<h2> Welcome <?php echo htmlentities($\_SESSION['username']); ?> !</h2>

<a href="changepasswordform.php?username=<?php echo urlencode($\_SESSION['username']); ?>">Change password</a> | <a href="edituser.php?username=<?php echo urlencode($\_SESSION['username']); ?>">Edit profile</a> | <a href ="logout.php">logout

</a>

**Changepasswordform.php:**

<h1>Change Password Form</h1>

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

Username: <input type="text" class="text\_field" name="username" value="<?php echo isset($\_GET['username']) ? htmlentities($\_GET['username']) : ''; ?>" readonly /> <br>

Current Password: <input type="password" class="text\_field" name="password" /> <br>

New Password: <input type="password" class="text\_field" name="newpassword" /> <br>

<button class="button" type="submit"> Change password </button>

Edituser.php:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Edit Profile - WAPH</title>

<script type="text/javascript">

function displayTime() {

document.getElementById('digit-clock').innerHTML = "Current time: " + new Date();

}

setInterval(displayTime, 500);

function validateForm() {

var email = document.forms["editProfileForm"]["email"].value;

var phone = document.forms["editProfileForm"]["phone"].value;

// Email validation

var emailPattern = /^[\w.-]+@[\w-]+(\.[\w-]+)\*$/;

if (!emailPattern.test(email)) {

alert("Please enter a valid email address");

return false;

}

// Phone number validation

var phonePattern = /^\d{10}$/;

if (!phonePattern.test(phone)) {

alert("Please enter a valid 10-digit phone number");

return false;

}

}

</script>

</head>

<body>

<h1>Edit User Profile - WAPH</h1>

<div id="digit-clock"></div>

<?php

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

?>

<form name="editProfileForm" action="updateuser.php" method="POST" class="form edit-profile" onsubmit="return validateForm();">

Username: <input type="text" class="text\_field" name="username" value="<?php echo isset($\_GET['username']) ? htmlentities($\_GET['username']) : ''; ?>" readonly /> <br>

Full Name: <input type="text" class="text\_field" name="fullName" required><br>

Email: <input type="email" class="text\_field" name="email" required placeholder="username@example.com"><br>

Password: <input type="password" class="text\_field" name="password" required><br>

Phone Number: <input type="tel" class="text\_field" name="phoneNumber" required pattern="[0-9]{10}" title="Please enter a 10-digit phone number"><br>

<button class="button" type="submit">Update Profile</button>

</form>

</body>

</html>

**registrationform.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>WAPH-Login page</title>

<script type="text/javascript">

function displayTime() {

document.getElementById('digit-clock').innerHTML = "Current time: " + new Date();

}

setInterval(displayTime, 500);

function validateForm() {

var email = document.forms["registrationForm"]["email"].value;

var phone = document.forms["registrationForm"]["phone"].value;

// Email validation

var emailPattern = /^[\w.-]+@[\w-]+(\.[\w-]+)\*$/;

if (!emailPattern.test(email)) {

alert("Please enter a valid email address");

return false;

}

// Phone number validation

var phonePattern = /^\d{10}$/;

if (!phonePattern.test(phoneNumber)) {

alert("Please enter a valid 10-digit phone number");

return false;

}

}

</script>

</head>

<body>

<h1>New user registration, WAPH</h1>

<div id="digit-clock"></div>

<?php

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

?>

<form name="registrationForm" action="addnewuser.php" method="POST" class="form login" onsubmit="return validateForm();">

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

pattern="^[\w.-]+@[\w-]+(.[\w-]+)\*$"

title="Email address is required as username"

placeholder="username in email"

onchange="this.setCustomerValidity(this.validity.patternMismatch?this.title: '');" /> <br>

Full Name: <input type="text" class="text\_field" name="fullName" required><br>

Email: <input type="email" class="text\_field" name="email" required placeholder="username@example.com"><br>

Password: <input type="password" class="text\_field" name="password" required><br>

Phone Number: <input type="tel" class="text\_field" name="phoneNumber" required pattern="[0-9]{10}" title="Please enter a 10-digit phone number"><br>

<button class="button" type="submit">Register</button>

</form>

</body>

</html>