|  |  |
| --- | --- |
| **Logo_Baru-Rasmi_UUM_02** | STIW3044: Web Engineering Semester A212School of Computing, CAS, UUM **FRONT COVER** |
| **Lab 2** | |
| |  |  | | --- | --- | | **Name** | Priananda Azhar | | **Matric No** | s702299 | | **YouTube Presentation Link** | Not Required | | **Phone Number** | +60143056118 | | **GitHub Link** | https://github.com/Priananda620/LAB-2-Web-Eng-Priananda | | **Submission Date** | 24/5/2022 | | **Acknowledgment** | I hereby acknowledge that the following works are from my effort in submitting this document. If found otherwise, severe action such as marks deduction or removal from the assignment can be taken against me. | | **Digital Signature** |  | | **Students Picture** |  | | **Email use during book purchase** | - | | **Your digital key from the book purchase** | - | | |

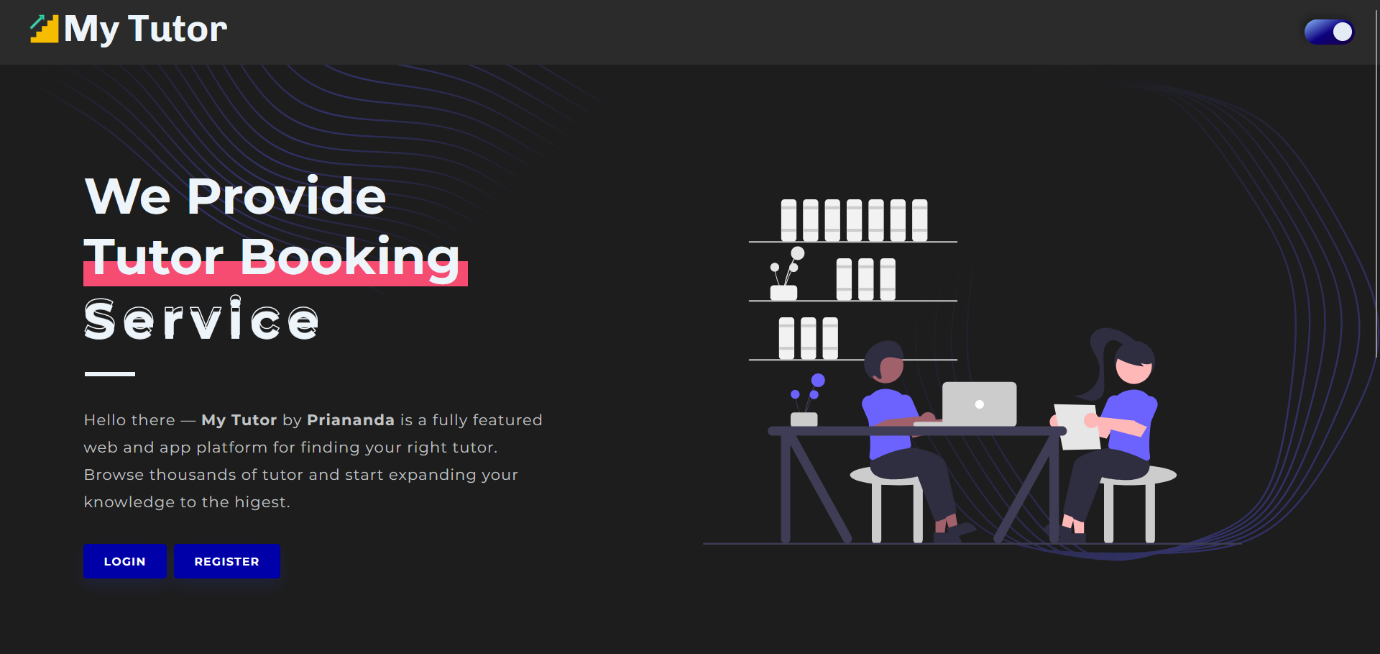
|  |  |
| --- | --- |
| **Logo_Baru-Rasmi_UUM_02** | STIW3044: Web Engineering Semester A212School of Computing, CAS, UUM |
| Lab 2 | |
| Given date: **18/5/2022**  Submission date**: 25/5/2022**  **15 Marks** | |

**Instruction**

Your task is to develop a user registration and login page for a web application called MyTutor. The application is for a tutor booking and payment. Create a project directory with all the relevant folders and files. The registration form page should include image uploads and other users’ registration data such as email, name, phone number, password, and home address. Backend implementation only for registration.

Answer the following questions. Don’t show the entire code. Just select a segment of the codes responsible for the required questions. Make sure to retain the code format from VSC when you paste it into the Word document. Don’t screenshot your code!.

Based on the tasks above answer the following questions

1. Show your user login and user registration user interface design for the application. You may use any digital tools that can help to build the user interfaces.

Responsive (using @media only screen and (min-width/max-width….

Graphical user interface, application

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidence

A screenshot of a video game

Description automatically generated

(2 Marks)

<your answer here>

1. Show code segment for the following tasks. (Your GitHub link to the project will be used as part of the evaluation):
   1. HTML Form for user registration

(2 Marks)

<your answer here>

<div id="register-body">

                <h2>Register</h2>

                <form enctype="multipart/form-data" class="d-flex flex-column">

                    <label for="email">email<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span></label>

                    <input type="email" name="email" class="form-input" required="">

                    <label for="username">username<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span></label>

                    <input type="text" name="username" class="form-input" required="">

                    <label for="phone">phone<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span></label>

                    <input type="text" name="phone" class="form-input" required="">

                    <label for="address">address<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span></label>

                    <textarea name="address" class="form-input" required="" rows="6" cols="50"></textarea>

                    <label for="user\_image64">profile picture &nbsp;<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span></label>

                    <input type="file" name="user\_image64" placeholder="photo" required="">

                    <label for="password">password<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span><span class="message"></span><span class="show-password float-end">show &nbsp;<i class="fas fa-eye"></i></span></label>

                    <input type="password" name="password" class="form-input" required="">

                    <label for="verify\_pass">re-type password<span style="color: red;">&nbsp;\*&nbsp;&nbsp;&nbsp;</span></label>

                    <input type="password" name="verify\_pass" class="form-input" required="">

                    <p id="SUCCESS-regis" class="mt-4" style="display: none">SUCCESS REGISTER, you can check out on the db sir, no user session created.</p>

                    <p id="username-already-exist" class="mt-4" style="display: none">Username Already Exist...</p>

                    <p id="unmatch-password" class="mt-4" style="display: none">Password Does Not Match...</p>

                    <p id="unmatch-pass-length" class="mt-4" style="display: none">Password Too Short...</p>

                    <div class="d-inline-flex align-items-center flex-wrap mt-4">

                        <div class="d-inline-flex align-items-center justify-content-between">

                            <button class="button me-2" id="register-action" style="width: 50%;">Register&nbsp;&nbsp;<div class="fa-2x"><i class="fas fa-circle-notch fa-spin"></i></div></button>

                            <button class="button clear-input-action me-2" style="width: 50%; background: var(--red-accent)">Clear Inputs</button>

                        </div>

                        <div>

                            <p>already have account?</p>

                            <p class="login-show" style="color:var(--link-font); cursor:pointer">Log In</p>

                        </div>

                    </div>

                </form>

            </div>

I use asynchronous request using AJAX

$("button#register-action").on('click', (e) => {

        console.log("REGISTEEERRRR")

        $('#username-already-exist').css('display', 'none')

        $('#unmatch-password').css('display', 'none')

        $('#unmatch-pass-length').css('display', 'none')

        $('button#register-action .fa-2x').css('display', 'none')

        $('#SUCCESS-regis').css('display', 'none')

        let form = $("#register-body form")[0];

        let fd = new FormData(form)

*if* (fd.get('email') != "" && fd.get('username') != "" && fd.get('phone') != "" && fd.get('address') != "" && fd.get('user\_image64').length !== 0 && fd.get('password') != "" && fd.get('verify\_pass') != "") {

            $('.fa-2x').addClass("d-block")

            e.preventDefault();

            let reader = new FileReader()

            reader.onloadend = () => {

*// console.log(reader.result)*

                fd.delete("user\_image64")

                fd.append("user\_image64", reader.result)

                console.log(fd.get('user\_image64'))

                console.log(fd.get("email"))

                $.ajax({

                    url: 'api/register.php',

                    method: 'POST',

                    data: fd,

                    processData: false,

                    contentType: false,

                    success: function (response) {

                        $('.fa-2x').removeClass("d-block")

                        let json = response

                        console.log(json)

*if* (json.success) {

                            console.log(json.success);

                            console.log(json.account\_data);

                            $('#SUCCESS-regis').css('display', 'unset')

*// location.reload();*

                        } *else* *if* (!json.success && json.password\_verify\_unmatch) {

                            console.log(json.success);

                            console.log("password\_verify\_unmatch:")

                            console.log(json.password\_verify\_unmatch)

                            $('#unmatch-password').css('display', 'unset')

                        } *else* *if* (!json.success && json.account\_exist) {

                            console.log(json.success);

                            console.log("account\_exist:")

                            console.log(json.account\_exist);

                            $('#username-already-exist').css('display', 'unset');

                        } *else* *if* (!json.success && json.password\_length\_unmatch) {

                            console.log(json.success);

                            console.log("password\_length\_unmatch:")

                            console.log(json.password\_length\_unmatch);

                            $('#unmatch-pass-length').css('display', 'unset');

                        } *else* *if* (!json.success) {

                            console.log("success:")

                            console.log(json.success);

                        }

                    }

                })

            }

            reader.readAsDataURL(fd.get('user\_image64'))

        } *else* {

*return*

        }

    })

* 1. PHP code for user registration.

(3 Marks)

<your answer here>

<?php

error\_reporting(E\_ERROR | E\_PARSE);

function guidv4($data = null)

{

    $data = $data ?? random\_bytes(16);

    assert(strlen($data) == 16);

    $data[6] = chr(ord($data[6]) & 0x0f | 0x40);

    $data[8] = chr(ord($data[8]) & 0x3f | 0x80);

*return* vsprintf('%s%s-%s-%s-%s-%s%s%s', str\_split(bin2hex($data), 4));

}

*require\_once* 'db\_config.php';

$connect = new mysqli($db\_host, $db\_username, $db\_password, $db\_name);

header('Content-Type: application/json');

*if* (empty($\_POST['email']) || empty($\_POST['username']) || empty($\_POST['phone']) || empty($\_POST['password']) || empty($\_POST['verify\_pass']) || empty($\_POST['address']) || empty($\_POST['user\_image64']) || $connect->connect\_error) {

    echo json\_encode(array('success' => false, 'no\_data' => true));

} *else* {

    $username = addslashes($\_POST['username']);

    $email = addslashes($\_POST['email']);

    $phone = addslashes($\_POST['phone']);

    $address = addslashes($\_POST['address']);

    $user\_image64 = $\_POST['user\_image64'];

    $password = $\_POST['password'];

    $verify\_pass = $\_POST['verify\_pass'];

    $ip = $\_SERVER['REMOTE\_ADDR'];

*try* {

        $details = json\_decode(file\_get\_contents("http://ipinfo.io/$ip/json"));

        $city = $details->city;

        $org = $details->org;

    }*catch* (Exception $e) {

        $city = null;

        $org = null;

    }

*if* ($verify\_pass != $password) {

        echo json\_encode(array('success' => false, 'password\_verify\_unmatch' => true));

    } *else* *if* (!(strlen($verify\_pass) >= 8)) {

        echo json\_encode(array('success' => false, 'password\_length\_unmatch' => true));

    } *else* {

*// $tmp = explode('.', $photo);*

*// $file\_name = guidv4().'.'.end($tmp);*

        $exp = explode(",", $user\_image64);

        $ex = reset($exp);

        $extension = reset(explode(";", end(explode("/", $ex))));

        $file\_name = guidv4() . '.' . $extension;

        $path = 'assets/user\_images/' . $username . "\_" . $file\_name;

        $sql = "INSERT INTO `users`(`email`, `username`, `phone`, `hased\_password`, `address`, `user\_image`, `ip`, `org`, `city`) VALUES (?,?,?,?,?,?,?,?,?)";

        $stmt = $connect->prepare($sql);

*if* (empty($org) && empty($city)) {

            $org = null;

            $city = null;

        }

        $hased\_pass = password\_hash($password, PASSWORD\_DEFAULT);

        $stmt->bind\_param("sssssssss", $email, $username, $phone, $hased\_pass, $address, $file\_name, $ip, $org, $city);

*if* ($stmt->execute()) {

*/////////////////////////////////////*

*// move\_uploaded\_file($\_FILES['picture']['tmp\_name'], "../".$path);*

            $base64only = str\_replace(' ', '+', end($exp));

            $base64decoded = base64\_decode($base64only);

            $path2write = '../' . $path;

            $is\_written = file\_put\_contents($path2write, $base64decoded);

*// session\_start();*

*// $\_SESSION['username'] = $username;*

            echo json\_encode(array('success' => true, 'account\_created' => true, 'account\_data' => array('username' => $username, 'email' => $email, 'phone' => $phone, 'address' => $address, 'user\_image' => $user\_image64)));

        } *else* {

            echo json\_encode(array('success' => false, 'account\_exist' => true));

        }

    }

}

* 1. HTML Form for user login

1. <div id="login-body">
2. <h2>Login to Your Account</h2>
3. <form method="POST" enctype="multipart/form-data" class="d-flex flex-column">
4. <label for="username">username&nbsp;<span style="color: red;">\*&nbsp;&nbsp;</span></label>
5. <input type="text" name="username" class="form-input" required="">
6. <label for="password">password&nbsp;<span style="color: red;">\*&nbsp;&nbsp;</span><span class="show-password float-end">show &nbsp;<i class="fas fa-eye"></i></span></label>
7. <input type="password" name="password" class="form-input" required="">
8. <p id="" class="mt-4">Login unavailable (LAB 2 assignment)<br>Only register working for now | <span class="register-show" style="color:var(--link-font); cursor:pointer">Register</span> </p>
9. *<!-- <p id="username-not-exist" class="mt-4" style="display: none">Username Not Exist...</p> -->*
10. *<!-- <p id="incorrect-password" class="mt-4" style="display: none">Incorrect Password...</p> -->*
11. <div class="d-inline-flex align-items-center justify-content-between mt-4">
12. <button class="button me-2" style="width: 50%;">Login</button>
13. <button class="button clear-input-action" style="width: 50%; background: var(--red-accent)">Clear Inputs</button>
14. </div>
15. </form>
16. </div>

(1 Marks)

<your answer here>

* 1. Graphical user interface, application

     Description automatically generatedDatabase table snapshot for user table

(2 Marks)

Graphical user interface

Description automatically generated with medium confidence

<your answer here>

1. Create and publish a YouTube presentation to demonstrate the user registration process. Make sure to demonstrate the registration page successfully inserted a new user into the database. Show app screenshots for all login and registration pages.

<your answer here>

(5 Marks)

Submission

* This document submit to the learning system when the link is open for submission.
* Upload your project to your GitHub Repository (provide the link on the front page).