



CHIANG MAI UNIVERSITY

Bachelor of Science (Software Engineering)

College of Arts, Media and Technology

1st Semester / Academic Year 2020

SE 261 Interactive Web Development

Assignment 03

Name Student ID Section.....

Assignment 3: Authentication and Sessions of BMI program (10 points)

In this assignment, you will continue working on the BMI program. So far, you should have implemented the following features of the BMI program.

- View home page
- View BMI list
- Add a new record of BMI
- User registration.

What you have left to make this program complete is an authentication feature that allow only a registered user to have access control on adding a new BMI record in the program. Other feathers are free for non-registered user. To achieve this requirement, you need to work with the provided Authentication class and session variable.

Step1: Create Authentication class

- Open the folder of your BMI folder in your text editor.
- Create a new document named Authentication.php in the classes directory and modify the following code to fit with your databaseTable methods.

```

<?php
class Authentication
{
    private $users;
    private $usernameColumn;
    private $passwordColumn;

    public function __construct(DatabaseTable $users, $usernameColumn,$passwordColumn) {
        session_start();
        $this->users = $users;
        $this->usernameColumn = $usernameColumn;
        $this->passwordColumn = $passwordColumn;
    }

    public function login($username, $password)
    {
        $user = $this->users->find($this->usernameColumn,strtoupper($username));

        if (!empty($user) && password_verify($password,$user[0][$this->passwordColumn])) {
            session_regenerate_id();
            $_SESSION['username'] = $username;
            $_SESSION['password'] =$user[0][$this->passwordColumn];
            return true;
        } else {
            return false;
        }
    }

    public function isLoggedIn()
    {
        if (empty($_SESSION['username'])) {
            return false;
        }

        $user = $this->users->find($this->usernameColumn,strtoupper($_SESSION['username']));

        if (!empty($user) &&$user[0][$this->passwordColumn]===$_SESSION['password']) {
            return true;
        } else {
            return false;
        }
    }
}

```

Step2: Modify BMIController.php

- Since the program allows only the login user access to the adding a new record of BMI, the object of BMIController class needs to check whether a user is authenticated or not.
- Add Authentication \$authentication into the constructor of BMIController class.
- Use if..else block to check the status of a login user by calling \$this->authentication->isLoggedIn()

Step3: Create templates for user authentication.

- templates/login.html.php defines a form of user login. The UI and code are as follows.

Your email address Your password

Don't have an account? [Click here to register an account](#)

```
<?php
if (isset($error)) :
    echo '<div>' . $error . '</div>';
endif;
?>

<form method="post" action="">
    <label for="email">Your email address</label>
    <input type="text" id="email" name="email">
    <label for="password">Your password</label>
    <input type="password" id="password" name="password">
    <input type="submit" name="login" value="Log in">
</form>
<p>Don't have an account? <a href="index.php?controller=register&action=registerform">Click here to
register an account</a></p>
```

- loginsuccess.html.php contains the login successful message.

Login Successful

You are now logged in.

- loginerror.html.php contains the login error message. Use this template when the user trying to access the adding new BMI record without logging in.

You are not logged in

You must be logged in to view this page. [Click here to log in](#) or [Click here to register an account](#)

- logout.html.php contains the logout message.

Logged out

You have been logged out

Step 4: Create LoginController

- Create LoginController.php into controllers directory
 - LoginController class contains Authentication instance to process the user login.
 - Create five functions
 - loginform() : load template login.html.php and return an Array of output and title.
 - success() : load template loginsuccess.html.php and return an Array of output and title.
 - error() : load template loginerror.html.php and return an Array of output and title.
 - processlogin() : process user log in by calling login function of \$this->authentication object. If user inputs invalid username/password, load template login.html.php and return error message and Array of output and title.
 - logout() : destroy session and load template logout.html.php and return an Array of output and title.

Step 5: Create a single-entry point of all Controller class

- Create a single-entry point of RegisterController, BMIController and LoginController object to handle all action by modifying **public/index.php**. To access any of the pages on the website, you'll have to type index.php followed by controller and action variables
 - Include all classes, controllers, databases into index.php
 - User try...catch statement to handle the PDOException
 - Use if..else blocks to check type of \$_SERVER['REQUEST_METHOD'] in order to define current controller and action. (See the ch12 lecture note)

Step 6: Add login and logout in the layout.html.php

- Add login and logout in the layout.html.php applying the following code in your BMI menu.

```
<?php if ($loggedIn) : ?>
    <a href="index.php?controller=login&action=logout">Log out</a>
<?php else : ?>
    <a href="index.php?controller=login&action=loginform">Log in</a>
<?php endif; ?>
```