

Homework Turnin

Name: Son L Nguyen
Email: koojin@uw.edu
Student ID: 1163043
Section: MA
Course: CSE 190m 12sp
Assignment: hw9
Receipt ID: 78d7451340162e76ef154901c827ad96

Warning: Your turnin is 2 days late. Assignment hw9 was due Friday, June 1, 2012, 11:30 PM.

Replacing prior submission from Sun 2012/06/03 06:58pm.

Turnin Successful!

The following file(s) were received:

top.html (1256 bytes)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Remember the Cow</title>

    <link href="cow.css" type="text/css" rel="stylesheet" />
    <link href="https://www.cs.washington.edu/education/courses/cse190m/12sp/homework/9/faq" type="text/css" rel="stylesheet" />

    <script src="https://ajax.googleapis.com/ajax/libs/prototype/1.7.0.0/prototype.js" type="text/javascript"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/scriptaculous/1.9.0/scriptaculous.js" type="text/javascript"></script>
    <script src="https://www.cs.washington.edu/education/courses/cse190m/12sp/homework/9/pseudocode.js" type="text/javascript"></script>
    <script src="todolist.js" type="text/javascript"></script>
  </head>
  <body>
    <div class="headfoot">
      <h1>
        
        Remember<br />the Cow
      </h1>
    </div>

    <!-- this div visibility is hidden, it won't display if there is a flash message -->
    <div id="reserved4flash">This space is reserved for flash message</div>

    <!-- shared page top HTML -->
```

bottom.html (906 bytes)

```

<!-- shared page bottom HTML -->

<div class="headfoot">
  <p>
    "Remember The Cow is nice, but it's a total copy of another site." - PCWorld <
    All pages and content &copy; Copyright CowPie Inc.
  </p>

  <div id="w3c">
    <a href="https://webster.cs.washington.edu/validate-html.php">
      
      
      
    </div>
  </div>
</body>
</html>

```

COW.CSS (1172 bytes)

```

/*
  Son Nguyen
  CSE 190 M
  Homework #9

  TA: Rishi Goutam

  css style sheet for remember the cow todolist web page
*/

body, h1 {
  margin: 0px;
  padding: 0px;
}

body, form, input {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14pt;
}

.headfoot {
  background-color: #005AB4;
  color: white;
  padding: .5em;
}

.headfoot h1 img {
  float: left;
  margin-right: 1em;
}

#main, #main input, a {
  color: #005AB4;
}

#main {
  width: 600px;
  margin: auto;
  padding-bottom: 2em;
}

#main h2, #buttons, #logout {
  text-align: center;
}

#main h2, #main p, #buttons, #logout {
  margin-top: 0px;
}

```

```

#logout {
    position: absolute;
    top: 40px;
    left: 85%;
    background-color: white;
    padding: 5px;
}

#logout:hover {
    background-color: #CCCCAA;
}

#flash, #reserved4flash {
    background-color: rgb(255,210,211);
    text-align: center;
    border: 1px dotted black;
    margin: 0.5em;
    padding: 0.5em;
    height: 25px;
}

#reserved4flash {
    visibility: hidden;
}

/*
#flash .error {
    background-color: rgb(255,210,211);
}

#flash .success {
    background-color: #FFFFCC;
}
*/

```

index.php (1723 bytes)

```

<?php

/*
    Son Nguyen
    CSE 190 M
    Homework #9

    TA: Rishi Goutam

    php code for index page
    allows user input username & password to login
    redirects to todolist page if user is already logged in
*/

include("top.html");
include("shared.php");

start_page_session();
check_logged_in();
set_flash_message();

?>

<div id="main">
    <p>
        The best way to manage your tasks. <br />
        Never forget the cow (or anything else) again!
    </p>

    <p>
        Log in now to manage your to-do list:
    </p>

```

```

<form id="loginform" action="login.php" method="post">
  <div><input id="name" name="name" type="text" size="12" autofocus="autofocus"
  <div><input id="password" name="password" type="password" size="12" /> <strong>
  <div><input id="submitButton" type="submit" value="Log in" /></div>
</form>

<p id="error"></p>

</div>

<?php include("bottom.html");

/*
$dom = new DOMDocument();
$html = "test.html";
$dom->validateOnParse = true; //<!-- this first
$dom->loadHTMLFile($html);      //'cause 'load' == 'parse
$dom->preserveWhiteSpace = false;

$belement = $dom->getElementById("");
echo $belement->nodeValue;

$dom = new DOMDocument();

$flash = getElementById("flash");

$p_tag = $dom->createElement("p");
$p_tag->appendChild($dom->createTextNode($_SESSION["flash"]));
$flash->appendChild($p_tag);
unset($_SESSION["flash"]);
*/
?>

```

login.php (618 bytes)

```

<?php

/*
Son Nguyen
CSE 190 M
Homework #9

TA: Rishi Goutam

php code that allow user login form to submits to here
Upon login, remembers user login name in a PHP session variable

*/

include("shared.php");

$username = get_parameter("name");
$password = get_parameter("password");

if (is_correct_password($username, $password)) {
  start_page_session();
  $_SESSION["name"] = $username; #remember user name
  redirect("todolist.php", "Login successful! Welcome back.");
} else {
  redirect("index.php", "Incorrect Username/Password. Please try again.");
}

?>

```

logout.php (444 bytes)

```

<?php
/*
    Son Nguyen
    CSE 190 M
    Homework #9

    TA: Rishi Goutam

    php code that redirect todolist page to index page
    also kill all sessions when user click on logout link

*/

# This page shows a user form for the student to log out of the system
require_once("shared.php");
session_start();
session_destroy();
session_regenerate_id(TRUE);
session_start();
redirect("index.php", "Logout successful.");
?>

```

todolist.php (903 bytes)

```

<?php
/*
    Son Nguyen
    CSE 190 M
    Homework #9

    TA: Rishi Goutam

    php code for todolist page
    Shows the user To-Do List, user must be logged in to see

*/

include("top.html");
include("shared.php");

start_page_session();
ensure_logged_in();
set_flash_message();

?>
    <div id="main">
        <h2><?= $_SESSION["name"] ?>'s To-Do List</h2>

        <div id="buttons">
            <input id="itemtext" type="text" size="30" autofocus="autofocus" /> <br />
            <button id="add">Add to Bottom</button>
            <button id="delete">Delete Top Item</button>
        </div>

        <div id="listblock">
            <ul id="todolist"></ul>
        </div>

        <p id="logout"><a href="logout.php">Log Out</a></p>
    </div>

<?php include("bottom.html"); ?>

```

todolist.js (4576 bytes)

```

/*
    Son Nguyen
    CSE 190 M

```

Homework #9

TA: Rishi Goutam

JavaScript code for Remember the Cow page

Uses AJAX to post/get user To-Do List to/from webservice

Adds effects to page using Scriptaculous Library

Pops up flash messages on top of page

```

*/
"use strict";

// Get flash message on index.php AND todolist.php
// Attach event handler to buttons
document.observe("dom:loaded", function() { // same as window.onload = function() { };
    getFlashMessage();
    if ($("#add")) { // only attach event handler to objects on todolist.php
        $("#add").observe("click", addClick);
        $("#delete").observe("click", deleteClick);
        getToDoList();
    }
});

// Called when Add To Bottom button is clicked
// Add user input text to the bottom of the list
function addClick() {

    if ($("#itemtext").value == "") { // do nothing when there is no input
        $("#itemtext").focus();
        return;
    }

    var li = document.createElement("li");
    li.innerHTML = $("#itemtext").value.escapeHTML();
    $("#todolist").appendChild(li);

    appearEffects(li);

    Sortable.create($("#todolist"), { // make list Sortable so that its
        onChange: listItemUpdate // elements can be dragged up and down
    });

    $("#itemtext").value = ""; // reset textbox
    postToDoList();
}

// Called when Delete Top Item button is clicked
// Removes the top item of the list
function deleteClick() {
    var todolist = $$("#todolist li");
    if (todolist.length==0) {
        $("#itemtext").focus();
        return;
    }
    disappearEffects(todolist[0]);
}

// Adds effects to the appearance of list item
function appearEffects(obj) {
    obj.hide();
    obj.grow({
        afterFinish: function() {
            obj.shake();
        }
    });
}

// Adds effects to the disappearance of list item
function disappearEffects(obj) {
    obj.fade({ afterFinish: removeItem });
}

```

```

// Called when item completely disappeared
// Removes list item, updates to web service
function removeItem(effect) {
    effect.element.remove();
    postTodoList();
}

// Called when the list items are dragged
function listItemUpdate(list) {
    list.pulsate();
    postTodoList();
}

// Get user todolist JSON data from webservice to display on page
function getTodoList() {
    new Ajax.Request("webservice.php", {
        method: "get",
        onSuccess: displayTodoList,
        onFailure: ajaxFailure,
        onException: ajaxFailure
    });
}

// Updates todolist to webservice if user makes any change to the list
function postTodoList() {

    var jsonText = createJSON($("#todolist li"));
    new Ajax.Request("webservice.php", {
        method: "post",
        parameters: {todolist: jsonText},
        //onSuccess: function() {alert("success")},
        onFailure: ajaxFailure,
        onException: ajaxFailure
    });

    $("#itemtext").focus();
}

// Called when the Ajax request returns with the JSON data.
// Processes this data and displays To-Do List on page
function displayTodoList(ajax) {
    var data = JSON.parse(ajax.responseText);
    $("#listblock").hide();
    for (var i = 0; i < data.items.length; i++) {
        var li = document.createElement("li");
        li.innerHTML = data.items[i];
        $("#todolist").appendChild(li);
    }

    $("#listblock").slideDown();

    Sortable.create($("#todolist"), {
        onChange: listItemUpdate
    });
}

// Creates JSON text based on user todolist
function createJSON(list) {
    var items = [];
    for (var i = 0; i < list.length; i++) {
        items[i] = list[i].innerHTML;
    }
    var todolist = {};
    todolist.items = items;
    return JSON.stringify(todolist);
}

// Pops up a message on page (if any) when page is loaded
// Fade message box in 4 seconds
function getFlashMessage() {

```

```

    if ($("#flash")==null) {
        showBlankSpace();
    } else {
        $("#reserved4flash").hide();
        $("#flash").fade( {duration: 4, afterFinish: showBlankSpace });
    }
}

// Show the blank space that is reserved for message box
// This helps the content not to move up
function showBlankSpace() {
    $("#reserved4flash").show();
}

// Display errors on page if any ajax request failed
function ajaxFailure ajax, exception) {
    $("#errors").innerHTML = "Error making Ajax request:" +
        "\n\nServer status:\n" + ajax.status + " " + ajax.statusText +
        "\n\nServer response text:\n" + ajax.responseText;
    if (exception) {
        throw exception;
    }
}
}

```

webservice.php (1389 bytes)

```

<?php
/*
    Son Nguyen
    CSE 190 M
    Homework #9

    TA: Rishi Goutam

    This web service accepts a query parameter 'todolist' OR don't accept any
    when no parameter is passed ~> it searches list.json
    and outputs the user todolist data on in JSON format

    when 'todolist' is passed -> creates/updates list.json with user todolist
*/

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $todolist = get_parameter("todolist");
    file_put_contents("list.json", $todolist);
} else {

    $jsontext = file_get_contents("list.json");
    header("Content-type: application/json");
    if ($jsontext == false) {
        print "";
    } else {
        print $jsontext;
    }
}

# Returns the value of the given query parameter.
# If the parameter has not been passed, issues an HTTP 400 error.
function get_parameter($name) {
    if (isset($_POST[$name])) {
        return $_POST[$name];
    } else {
        header("HTTP/1.1 400 Invalid Request");
        die("HTTP/1.1 400 Invalid Request - you forgot to pass a '$name' parameter.");
    }
}

/*
    example of JSON data
    {
        "items": [

```



```

        "Buy a pet jellyfish!",
        "build an elaborate sculpture out of toothpicks and Swiss cheese",
        "add &lt;em&gt; tags to my web page",
        "buy Ben & Jerry's \"large\" size ice cream."
    ]
}
*/

?>

```

shared.php (1947 bytes)

```

<?php
/*
    Son Nguyen
    CSE 190 M
    Homework #9

    TA: Rishi Goutam

    This php file contains all php functions for Remember The Cow page
*/

# Start session for current page
function start_page_session() {
    if (!isset($_SESSION)) { session_start(); }
}

# Return TRUE if given password is correct password for this user name
function is_correct_password($name, $pw) {
    if ($name == "koojin" && $pw == "12345") {
        return TRUE;
    } else {
        return FALSE;
    }
}

# Redirects current page to the given URL and optionally sets flash message
function redirect($url, $flash_message = NULL) {

    if ($flash_message) {
        if (!isset($_SESSION)) { session_start(); }
        $_SESSION["flash"] = $flash_message;
    }
    header("Location: $url");
    die();
}

# Redirects current page to index.php if user is not logged in
function ensure_logged_in() {
    if (!isset($_SESSION["name"])) {
        redirect("index.php", "You must log in first");
    }
}

# Redirects current page to todolist.php if user is already logged in
function check_logged_in() {
    if (isset($_SESSION["name"])) {
        redirect("todolist.php");
    }
}

# Returns the value of the given query parameter.
# If the parameter has not been passed, issues an HTTP 400 error.
function get_parameter($name) {
    if (isset($_POST[$name])) {
        return $_POST[$name];
    } else {
        header("HTTP/1.1 400 Invalid Request");
        die("HTTP/1.1 400 Invalid Request - you forgot to pass a '$name' parameter.");
    }
}

```

```
    }  
}  
  
function set_flash_message() {  
    if (isset($_SESSION["flash"])) {  
        #temporary message across page redirects  
        ?>  
        <div id="flash"> <?= $_SESSION["flash"] ?> </div>  
        <?php  
        unset($_SESSION["flash"]);  
    }  
}  
  
/*  
function getElementById($id)  
{  
    $xpath = new DOMXPath($this->domDocument);  
    return $xpath->query("//*[@id='$id']")->item(0);  
}  
*/  
  
?>
```