Nick Rose – Project: Web Programming – 4-16-2015

A brief outline of the aspects included in my project which were learned through our course.

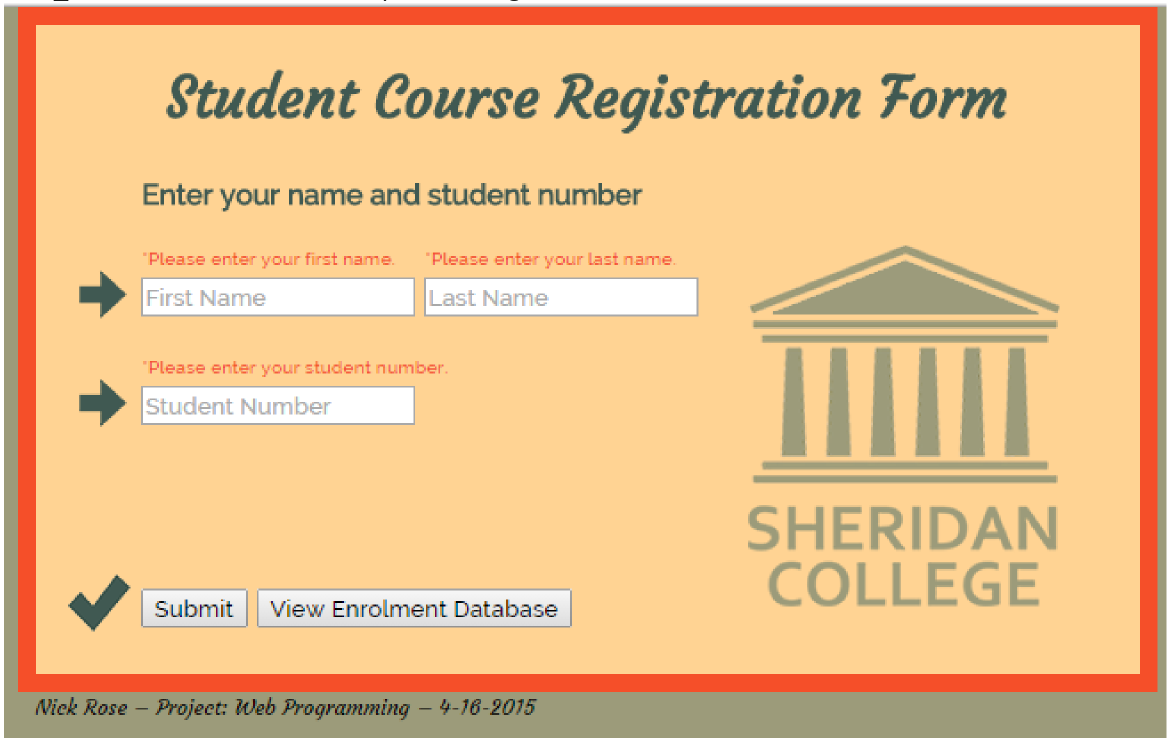
index.php

JavaScript form validation:

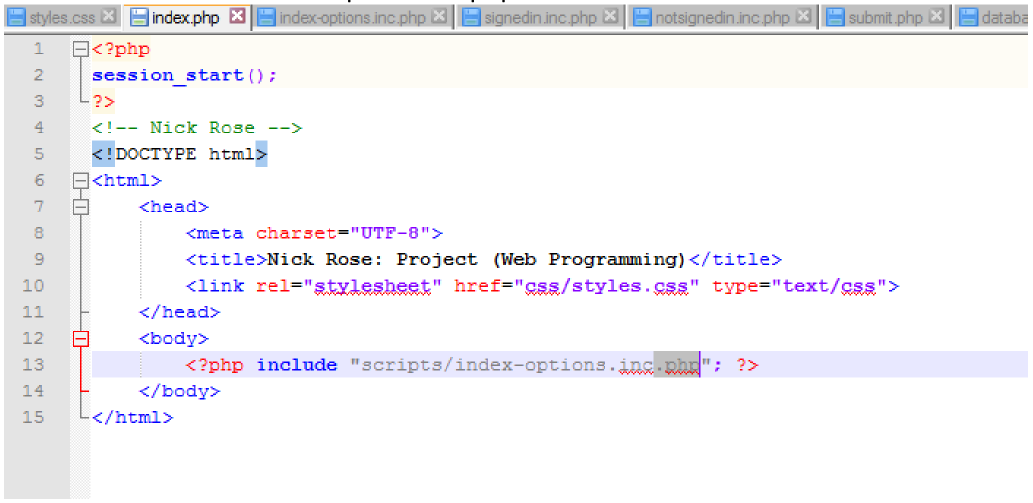
-user must enter all fields

-user must enter a student number that is six digits long

PHP Validation: user must enter an account that exists within my `accounts` table. If the first name, last name, and student number do not altogether match one of those entries in the `accounts` table, it will use return to index.php and let the user know that the form was submitted in error. See student_accounts.txt for the list of possible logins.

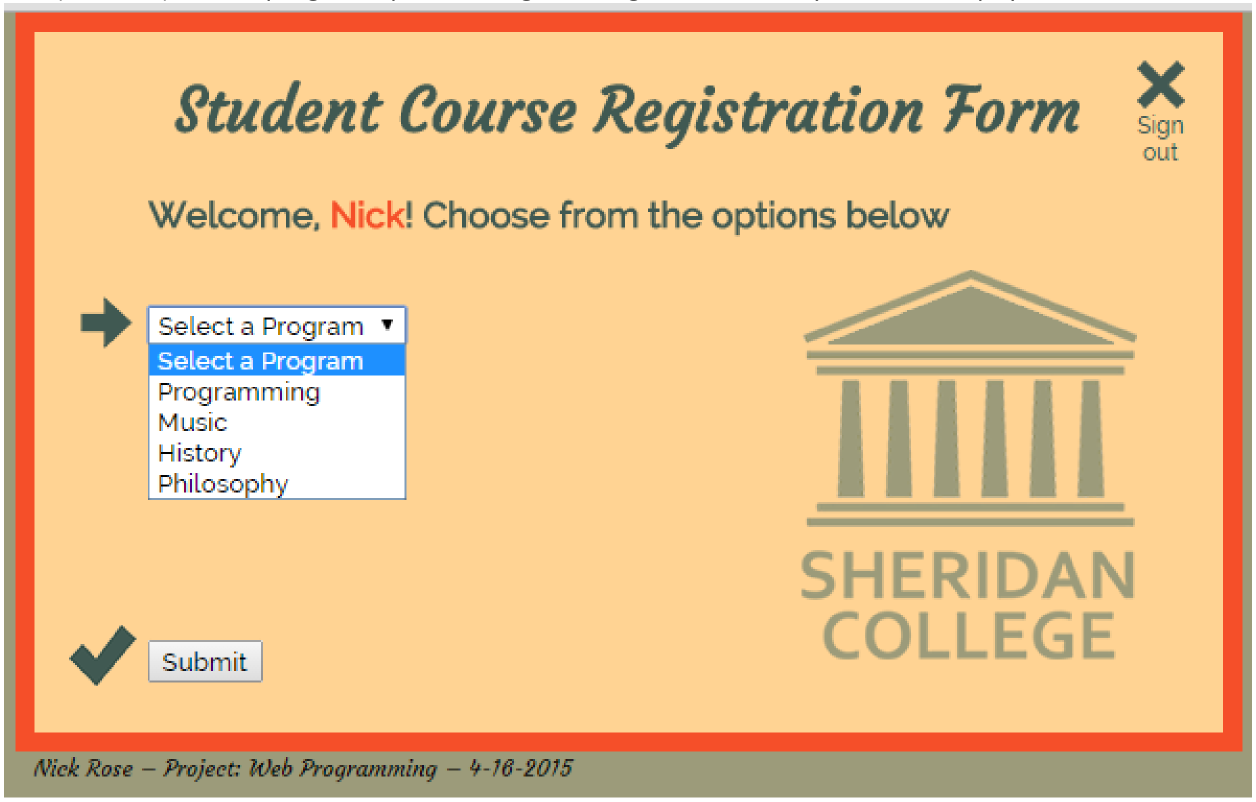

Student Course Registration Form
Enter your name and student number
*Please enter your first name.
*Please enter your last name.
First Name
Last Name
*Please enter your student number.
Student Number
Submit
View Enrolment Database
SHERIDAN COLLEGE
Nick Rose – Project: Web Programming – 4-16-2015

-this page uses an include file: index-options.inc.php

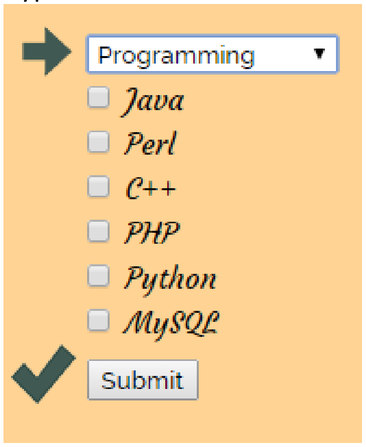


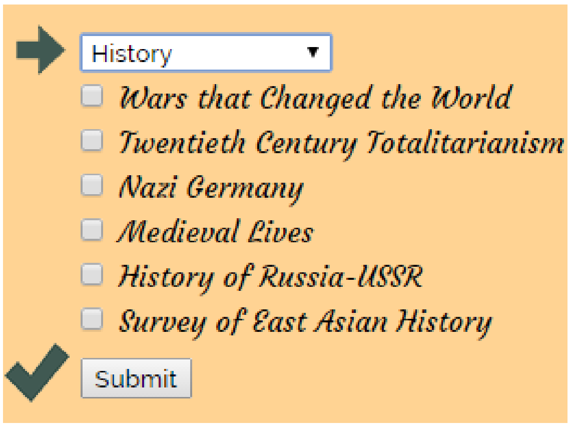
```
<?php
session_start();
?>
<!-- Nick Rose -->
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Nick Rose: Project (Web Programming)</title>
        <link rel="stylesheet" href="css/styles.css" type="text/css">
    </head>
    <body>
        <?php include "scripts/index-options.inc.php"; ?>
    </body>
</html>
```

index-options.inc.php:
-If the user has not submitted first name / last name / student number, this file will display a sign-in form on index.php
-Once the user signs in with that information, index.php reloads and replaces those fields with a combo box (<select>) of four program options: Programming, Music, History, and Philosophy.


Student Course Registration Form
Sign out
Welcome, Nick! Choose from the options below
Select a Program
Select a Program
Programming
Music
History
Philosophy
Submit
SHERIDAN COLLEGE
Nick Rose – Project: Web Programming – 4-16-2015

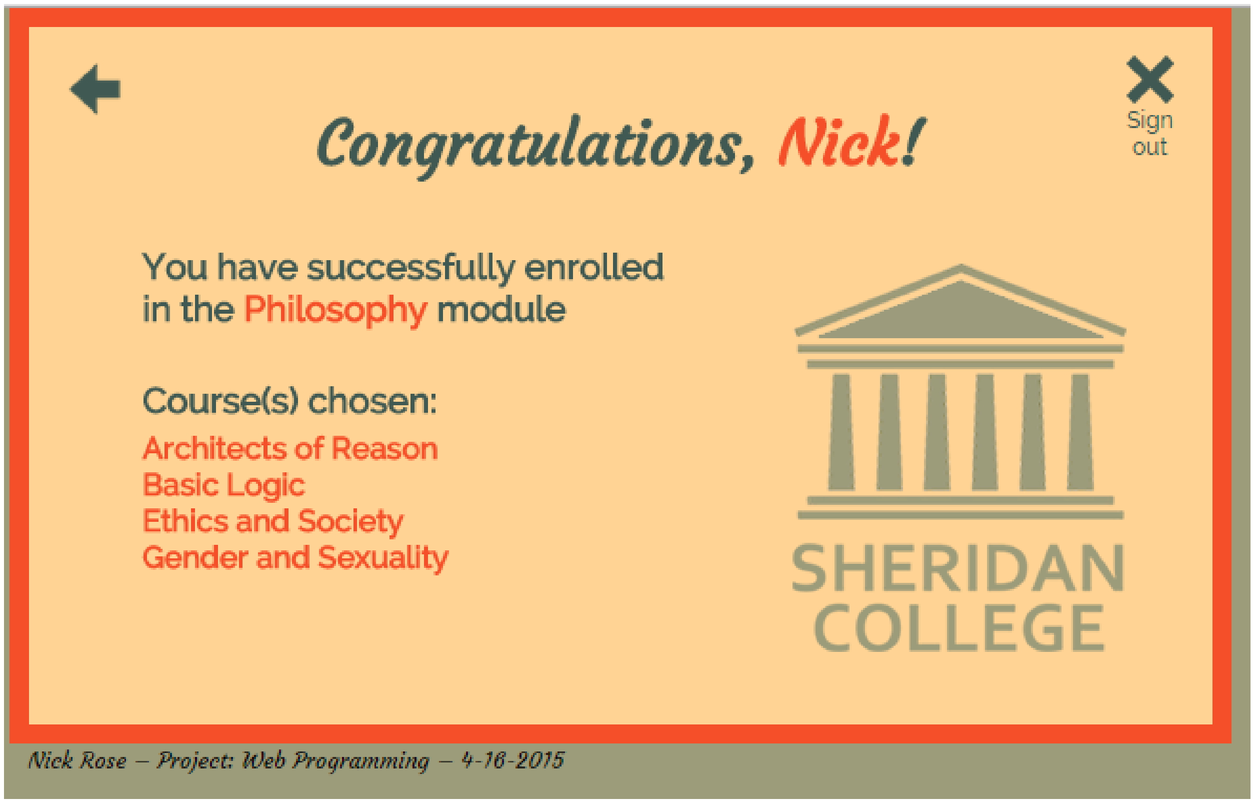
-When an option from this <select> is chosen, JavaScript dynamically creates checkboxes with labels of courses with that program. Each time the <select> <option> is changed, these labels change as well (the checkboxes are only created once to be more efficient). This is done using an event listener of the change event type.




History
Wars that Changed the World
Twentieth Century Totalitarianism
Nazi Germany
Medieval Lives
History of Russia-USSR
Survey of East Asian History
Submit

-JavaScript is used to validate that the user has selected a program and checked at least one course box.
-After the form is determined to be valid, submit redirects to submit.php

submit.php
-A PHP session is started on index.php, this enables me to display the student number, name, program, and courses the user chose as html text inside submit.php.



Congratulations, Nick!

You have successfully enrolled in the Philosophy module

Course(s) chosen:
Architects of Reason
Basic Logic
Ethics and Society
Gender and Sexuality

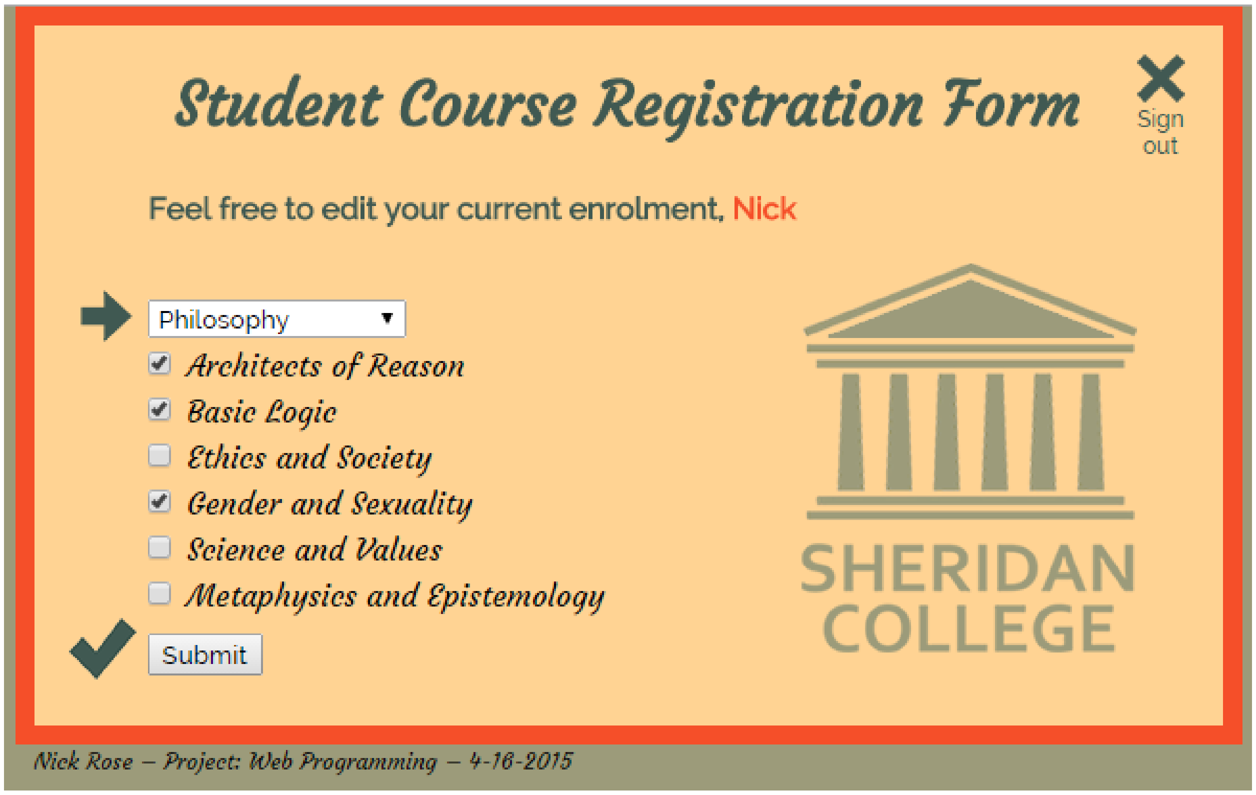
Nick Rose – Project: Web Programming – 4-16-2015

-on this page the user can either sign out (which ends the session) or click the back button. Clicking the back arrow button returns to index.php with a new subtitle “Feel free to edit your current enrolment, <username>.” The checkboxes and program choice have been saved from the user’s previous submission. Even if the user changes the <select> menu to a different program, then returns to their currently enrolled program, the checked checkboxes retain the user’s current enrolment choices. I was able to do this by combining PHP and JavaScript. A JavaScript variable “coursesEnrolled” is assigned with the contents of $_SESSION[‘courses’] (containing the users current enrolment). I have a JavaScript function which checks whether this JS variable then holds the innerhtml of the checkbox label (a course). If so, then the box is checked. The user then can edit their enrolment choice and resubmit, or sign out. Resubmitting updates the database, rather than creates a new row, because the student number is the same and maintained in its session variable.

-all of this information is also duplicated where necessary with cookies. The user experience will be the same as described above, but with that addition that if the window is closed and reopened, their sign-in progress is retained.

-If the user signs out then resigns in with the same student number, their enrolment choices are similarly still reflected within the dynamically created JavaScript objects.

-If someone directly types in the link the submit.php without submitting a form, they are redirected back to index.php



Student Course Registration Form

Sign out

Feel free to edit your current enrolment, Nick

Philosophy

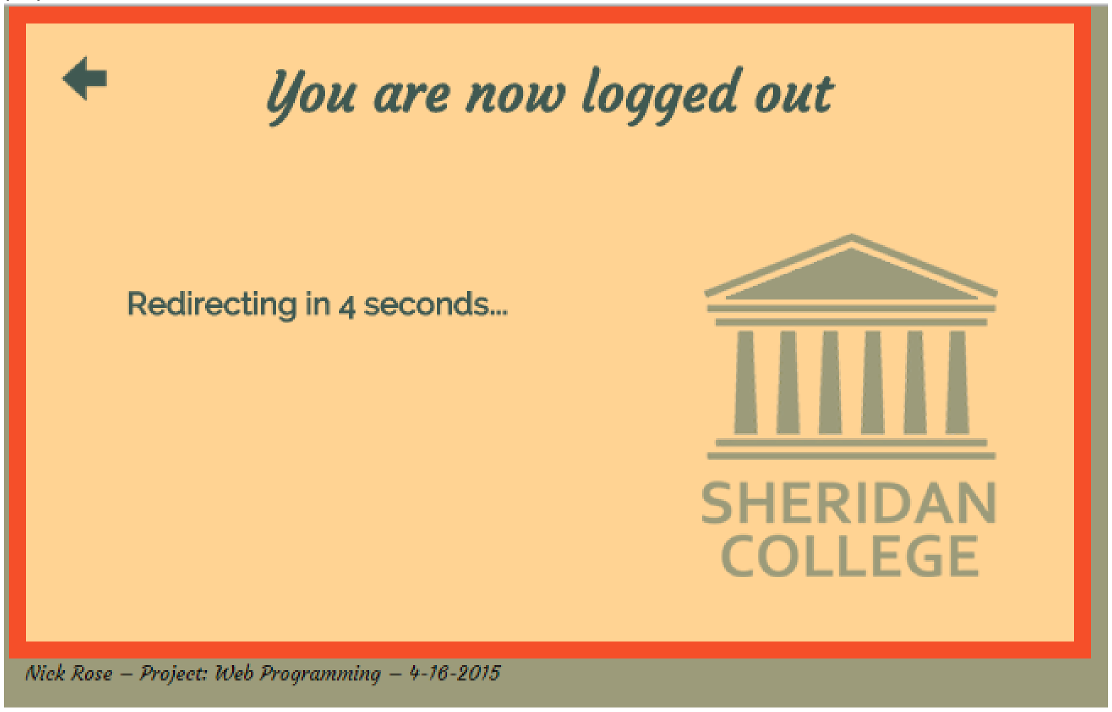
- [✓] Architects of Reason
- [✓] Basic Logic
- [ ] Ethics and Society
- [✓] Gender and Sexuality
- [ ] Science and Values
- [ ] Metaphysics and Epistemology

Submit

SHERIDAN COLLEGE

Nick Rose – Project: Web Programming – 4-16-2015

logout.php
-Destroys and unsets the session and cookies. This also uses JavaScript to provide a 5-second "countdown" to redirect back to index.php. PHP does the actual redirecting, however.
-If a user manually types logout.php in the browser while not signed in, it automatically redirects back to index.php.



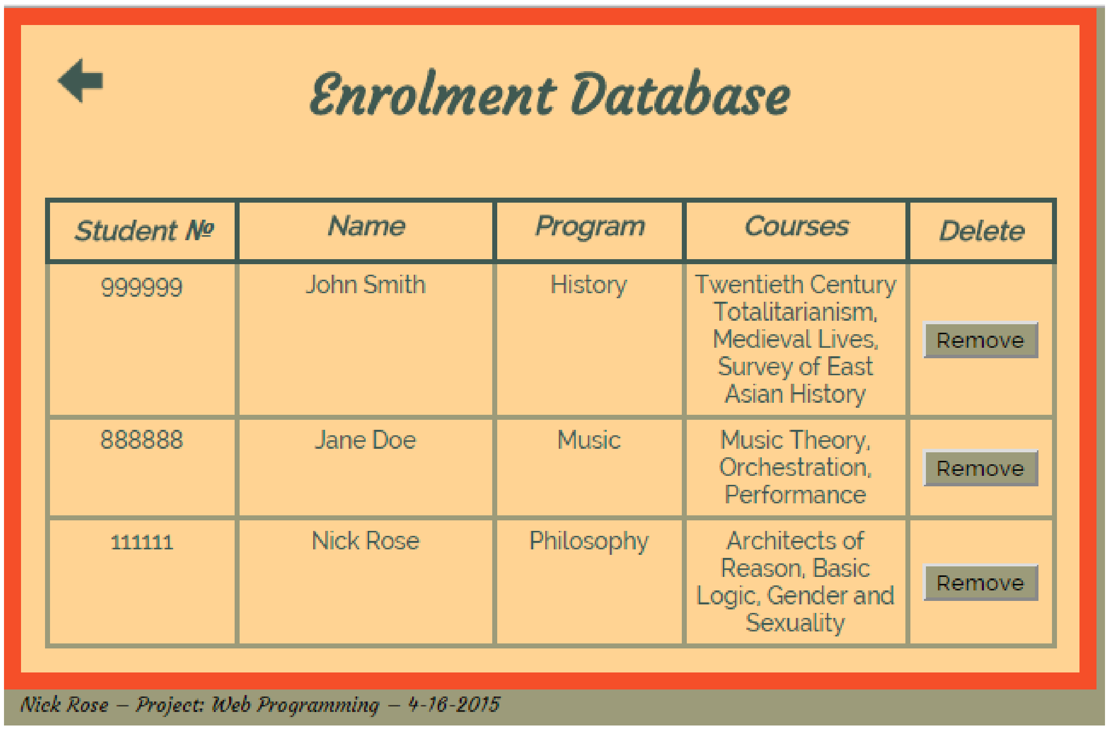
You are now logged out

Redirecting in 4 seconds...

SHERIDAN COLLEGE

Nick Rose – Project: Web Programming – 4-16-2015

database.php
-accessed by clicking the "View Enrolment Database" button on index.php
-requires correct username/password to view this page (done by using an .htaccess file)
-This consists of a table that fetches information from the MySQL database. In addition, there are "Remove" buttons for each row which delete the row from both the page and the database.

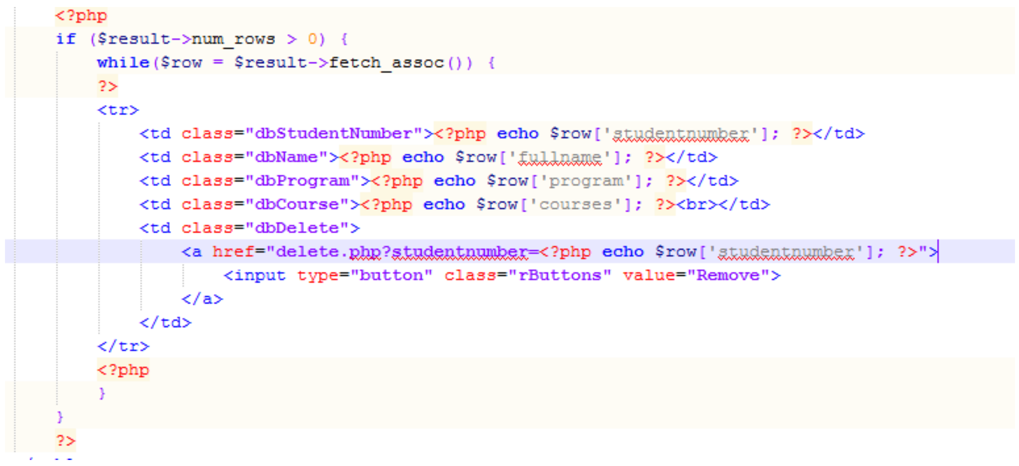


Enrolment Database

Student №	Name	Program	Courses	Delete
999999	John Smith	History	Twentieth Century Totalitarianism, Medieval Lives, Survey of East Asian History	Remove
888888	Jane Doe	Music	Music Theory, Orchestration, Performance	Remove
111111	Nick Rose	Philosophy	Architects of Reason, Basic Logic, Gender and Sexuality	Remove

Nick Rose – Project: Web Programming – 4-16-2015

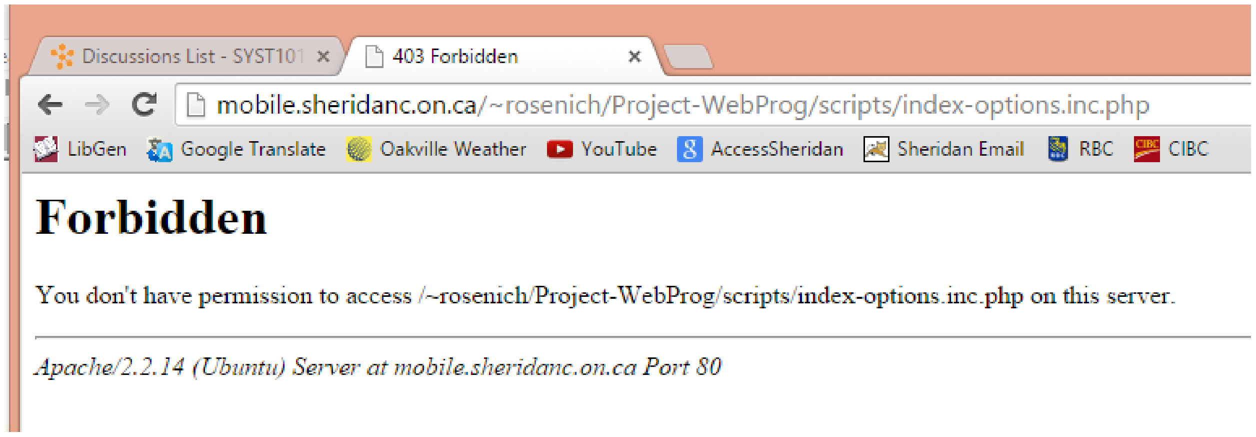
I was able to implement this removal by using PHP. As the database is being read row by row, a unique hyperlink is created for each Remove button in the Delete field. This is a link to delete.php with the student number attached to it, e.g. delete.php?studentnumber=111111. The student number comes from the row's respective student number entry while being fetched from the database. The page delete.php handles this link.



```
<?php
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
    ?>
    <tr>
        <td class="dbStudentNumber"><?php echo $row['studentnumber']; ?></td>
        <td class="dbName"><?php echo $row['fullname']; ?></td>
        <td class="dbProgram"><?php echo $row['program']; ?></td>
        <td class="dbCourse"><?php echo $row['courses']; ?><br></td>
        <td class="dbDelete">
            <a href="delete.php?studentnumber=<?php echo $row['studentnumber']; ?>">
                <input type="button" class="rButtons" value="Remove">
            </a>
        </td>
    </tr>
    <?php
    }
}
?>
```

delete.php
-Links to this page will be from the Remove buttons of database.php with the student number tacked onto the link itself. I use PHP GET to handle the student number value. Since student numbers are unique IDs, I can safely delete the row that contains this student number from the database. Delete.php automatically redirects back to database.php, so the process of deleting a row is a seamless transition for the user.

Misc details
-scripts and database content are secured with .htaccess files. The scripts folder itself denies access entirely from all, while the database requires a username/password.


Discussions List - SYST101
403 Forbidden
mobile.sheridanc.on.ca/~rosenich/Project-WebProg/scripts/index-options.inc.php
LibGen
Google Translate
Oakville Weather
YouTube
AccessSheridan
Sheridan Email
RBC
CIBC

Forbidden

You don't have permission to access /~rosenich/Project-WebProg/scripts/index-options.inc.php on this server.

Apache/2.2.14 (Ubuntu) Server at mobile.sheridanc.on.ca Port 80

-I used an include file for connecting to the database – connect.php – to make the process of each page connecting to the MySQL database server more readable.
-all four fundamental tasks of manipulating a database – C.R.U.D. – are implemented in this project. Student enrolment can be created, read, updated, and deleted.
-If a user enters manually any link included in this project's folder, they are redirected to an appropriate "safe" page – usually index.php. I checked whether PHP variables like those within POST or SESSION were set or not to determine whether one should legitimately be able to view a particular page through manually typing it into their browser's address bar.