

CSCE 4523 DBMS Project

Perry Mills, Luke Snyder

April 23, 2017

Included Files

The following files are included in the submission:

CreateTables.sh MySQL table definitions.

build_odbc.bash Compile backend .cpp files.

config.php Path definition.

index.php Landing page.

odbc_db.cpp, .h Interface to MySQL database.

odbc_insert_course.cpp, .php Handle function 2.

odbc_insert_enrollment.cpp, .php Handle function 3.

odbc_insert_student.cpp, .php Handle function 1.

odbc_reinitialize.cpp Reset tables.

odbc_select_deptCourses.cpp, .php Handle function 5.

odbc_select_studentCourses.cpp, .php Handle function 6.

odbc_select_students.cpp, .php Handle function 4.

Compilation

The assignment can be unpacked then built with GCC by executing **build_odbc.bash**.

The landing page can be accessed at:

- <http://csce.uark.edu/~ls008/StudentRegistration/>
- <http://csce.uark.edu/~pwm001/DBProject/StudentRegistration/>

The program has been compiled and tested in the Turing environment with connections from Firefox and Google Chrome.

Design

The web front is written in PHP, hosted via Turing's Apache server. HTML forms are used to gather input from the user and make calls to the back end, which is written in C++. The back end uses the provided ODBC interface (which was used in the previous assignment) to interact with a MySQL database, also hosted on Turing.

For portability, the path to which PHP calls are made to the backend is read from a single **config.php** file (one line). The database login information is read in the **odbc_db.cpp** constructor (three lines). With changes on just these four lines, the project can be ported to another location/user.

One bug-fix was made to the C++ version of the provided ODBC. Within **odbc_db::disconnect()**, a check is now made to ensure the resultSet is not null before attempting to delete it. Without this check, segmentation faults could and would occur regularly when making a connection without a valid query.

Additionally, a function was added to reinitialize the tables from the web front. This returns the database to fresh tables with several sample tuples in each.

Database tables

The database is designed with a generic university in mind. In many cases where characters are used, it is to accomodate schools which include letters in their naming system. Following the sample design, the database includes three tables:

Student

The Student table includes a unique identifier StudentId, the student's name and the student's major. The StudentId field supports up to 9 characters, which supports UA's 9-digit standard as well as other alpha-numeric systems. Because it is a unique identifier, it is the primary key. The StudentName field allows 50 characters, enough for most cases. Similarly, the Major field allows room for long degree names. None of the attributes may be null, and "Undeclared" can be used in place of NULL for Major.

Course

The Course table identifies courses by the combination of Department code and course number. For example, ECON 3404 is distinct from ECON 2004 and HUMN 3404, which are a different course within the same department and a different course outside the department which coincidentally uses the same course number, respectively. Course number allows up to 5 characters, supporting courses such as 3404L or other conventions. Title and Credit Hour

fields can be null, for flexibility in adding courses for which not all details are finalized.

Enrollment

An enrollment consists of pairing a student with a distinct course. Students are identified by `StudentId`, while courses have a composite key. As a result, all included attributes compose the primary key, and all attributes are foreign keys to the other tables (accordingly, no attributes may be null). Each foreign key constraint cascades deletion, so that if a course is removed all the associated students are unenrolled, and so that if a student is removed, that student is no longer enrolled in any courses.

Error checking

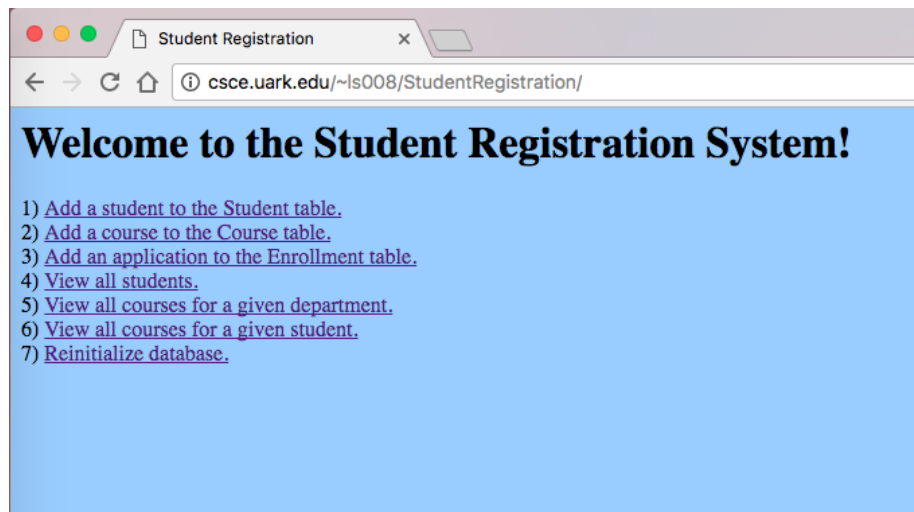
The error testing for this project was extensive and consists of the following:

1. Adding a student page
 - The `StudentId`, `Name`, or `Major` cannot be empty (validated through HTML form).
 - The `StudentId` must consist of only numeric digits (validated through PHP `isValidId()` function).
2. Adding a course page
 - The `DeptCode`, `CourseNum`, `Title`, or `CreditHours` cannot be empty (validated through HTML form).
 - The `CreditHours` must be a positive integer (validated through HTML form).
3. Adding an enrollment application page
 - The `StudentId`, `DeptCode`, and `CourseNum` cannot be empty (validated through HTML form).
 - Since each of the fields references a student and course in the respective tables, if any of the fields references non-existing data, an exception is thrown on the server side, generating a user-friendly error.
4. Viewing courses for a given department page
 - The `DeptCode` cannot be empty (validated through HTML form).
 - If a non-existing `DeptCode` is entered, an exception is thrown on the server side, generating a user-friendly error.
5. Viewing courses for a given `StudentId` page

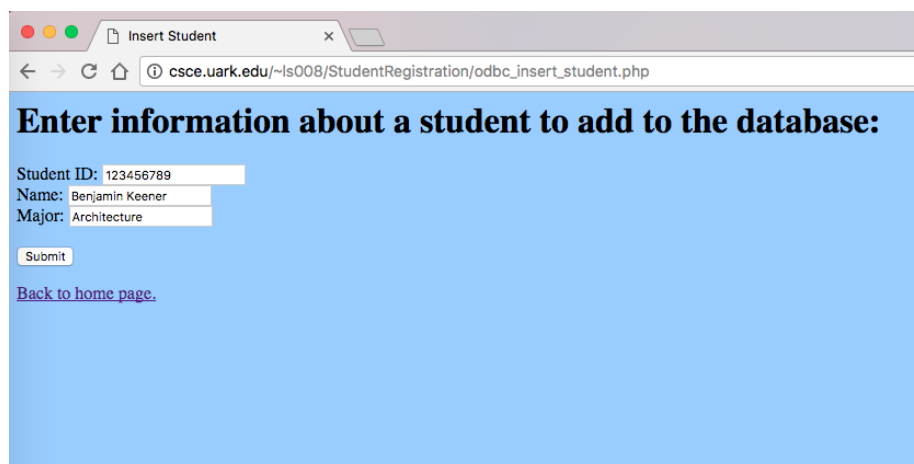
- The StudentId cannot be empty (validated through HTML form).
- If a non-existing StudentId is entered, or one which is not enrolled in any classes, an exception is thrown on the server side, generating a user-friendly error.

Screenshots

1. Landing page



2. Inserting a student



Insert Student

csce.uark.edu/~ls008/StudentRegistration/odbc_insert_student.php

Enter information about a student to add to the database:

Student ID:

Name:

Major:

[Back to home page.](#)

Student data successfully entered.

3. Inserting a course

Insert Course

csce.uark.edu/~ls008/StudentRegistration/odbc_insert_course.php

Enter information about a course to add to the database:

Department code:

Course number:

Course title:

Credit hours:

[Back to home page.](#)

A screenshot of a web browser window showing the 'Insert Course' form. The browser tabs include 'Insert Course' and 'Music Ensemble (MUEN) < Un...'. The address bar shows the URL 'csce.uark.edu/~ls008/StudentRegistration/odbc_insert_course.php'. The form has a light blue background and a title 'Enter information about a course to add to the database:'. It contains four input fields: 'Department code:', 'Course number:', 'Course title:', and 'Credit hours:'. A 'Submit' button is located below the fields. A link 'Back to home page.' is present, and a message 'Course data successfully entered.' is displayed at the bottom.

Insert Course

Music Ensemble (MUEN) < Un... X

csce.uark.edu/~ls008/StudentRegistration/odbc_insert_course.php

Enter information about a course to add to the database:

Department code:

Course number:

Course title:

Credit hours:

[Back to home page.](#)

Course data successfully entered.

4. Enrolling a student

Two screenshots of the 'Insert Student Enrollment' web form. The first screenshot shows the form with pre-filled values: Student ID: 123456789, Department code: MUEN, and Course number: 1401. The second screenshot shows the form with empty input fields. Both screenshots include the same browser tabs, address bar, title, input fields, 'Submit' button, 'Back to home page.' link, and 'Enrollment data successfully entered.' message.

Insert Student Enrollment

Music Ensemble (MUEN) < Un... X

csce.uark.edu/~ls008/StudentRegistration/odbc_insert_enrollment.php

Enter information about a student enrollment to add to the database:

Student ID: 123456789

Department code: MUEN

Course number: 1401

[Back to home page.](#)

Enrollment data successfully entered.

Insert Student Enrollment

Music Ensemble (MUEN) < Un... X

csce.uark.edu/~ls008/StudentRegistration/odbc_insert_enrollment.php

Enter information about a student enrollment to add to the database:

Student ID:

Department code:

Course number:

[Back to home page.](#)

Enrollment data successfully entered.

5. View all students

View Students x

csce.uark.edu/~ls008/StudentRegistration/odbc_select_students.php

All Student Information:

StudentId	StudentName	Major
123456789	Benjamin Keener	Architecture
010896544	Gregory Patterson	Economics
123768900	Macy Henderson	Geology
878656109	Nicholas Adams	Economics
131989665	Olivia Lipinski	Chemistry
012345678	Roger Williams	Electrical Engineering

[Back to home page.](#)

6. View all courses offered by a department

View Department Courses x

csce.uark.edu/~ls008/StudentRegistration/odbc_select_deptCourses.php

Enter a Department code to obtain its course offerings:

DeptCode
CHEM
CSCE
ECON
ELEG
GEOL
MUEN

Enter a Department code:

[Back to home page.](#)

View Department Courses x

csce.uark.edu/~ls008/StudentRegistration/odbc_select_deptCourses.php

Enter a Department code to obtain its course offerings:

DeptCode

CHEM

CSCE

ECON

ELEG

GEOL

MUEN

Enter a Department code:

[Back to home page.](#)

DeptCode	CourseNum	Title	CreditHours
MUEN	1401	Opera Theatre I	1

7. View all enrollments for a student

View Student Courses x

csce.uark.edu/~ls008/StudentRegistration/odbc_select_studentCourses.php

Enter a Student ID to obtain their enrolled course information:

Enter a Student ID:

[Back to home page.](#)

View Student Courses

csce.uark.edu/~ls008/StudentRegistration/odbc_select_studentCourses.php

Enter a Student ID to obtain their enrolled course information:

Enter a Student ID:

[Back to home page.](#)

DeptCode	CourseNum	Title	CreditHours
MUEN	1401	Opera Theatre I	1