CSCI 4000 Assignment 5

Total estimated time for this assignment: **12 hours** (if you are a good programmer)

The main purpose of this assignments is to apply and implement **MVC** (Model-view-controller) software architectural pattern. It is a continuation of the last assignment (assignment 4). You can copy your code from the last assignment (provided your code is correct and working). If you do not use MVC pattern, you will receive **0 points** for almost all the questions. The description for each question includes images of how the page should appear in a browser, along with specifications.

The main programming language for this assignment should be **PHP** (mixed with HTML and CSS). If you do not use **PHP**, you will get **0 points**. For this assignment, you must use **PDO** (PHP Data Object) to interface with the MySQL database. If you do not use PDO, you will receive 0 points for almost all questions.

When you see "Richard Ricardo" or "richardricardo" in the examples and screen captures, change it to **<your name>**.
When you see "Richard" or "richard" in the example screen captures, change it to **<your first name>**.
If you do not put **<your name> / <your first name>** in the above mentioned fields, you will get **0 points** for the question(s).

No two students should submit webpages with exactly the same code, content, layout, or color combination. If found, **both** students will get **0 points**.

Create a folder on your hard disk, name the folder **lastname_firstname_assignment5.** Save all the files from this assignment in this folder.

Use XAMPP web server solution stack package to help debugging PHP code. It will make your debugging process easier. All php files must not produce any error, or any warning **(-2 points for each error, each warning**). Your program must run. A program that does not run will get at most **50% of the possible points**. All files must begin with a **comment** that identifies the author, the course code, and the program date **(- 2 points each question** if found missing). All html, css and php files must be clearly **documented (commented)**. Points will be taken off **(-2 points each question)** for insufficient comments (`<!-- -->`, `/* */`, `//`).

- Before adding PHP code, all html files must pass html validation at http://validator.w3.org/ without any **error** (and with only 1 warning).
- When you view page source in a web browser, **`<!DOCTYPE html>`** must be at the top of every page. In other words, all pages must be written in HTML5. **(-20 points** if not)
  - You **can** put php code before **`<!DOCTYPE html>`**.
  - You **cannot** put html code before **`<!DOCTYPE html>`**.
- All css files must pass css validation at http://jigsaw.w3.org/css-validator/ without any **error**. **(-2 points for each error/warning**, only 1 warning is allowed for html validator)

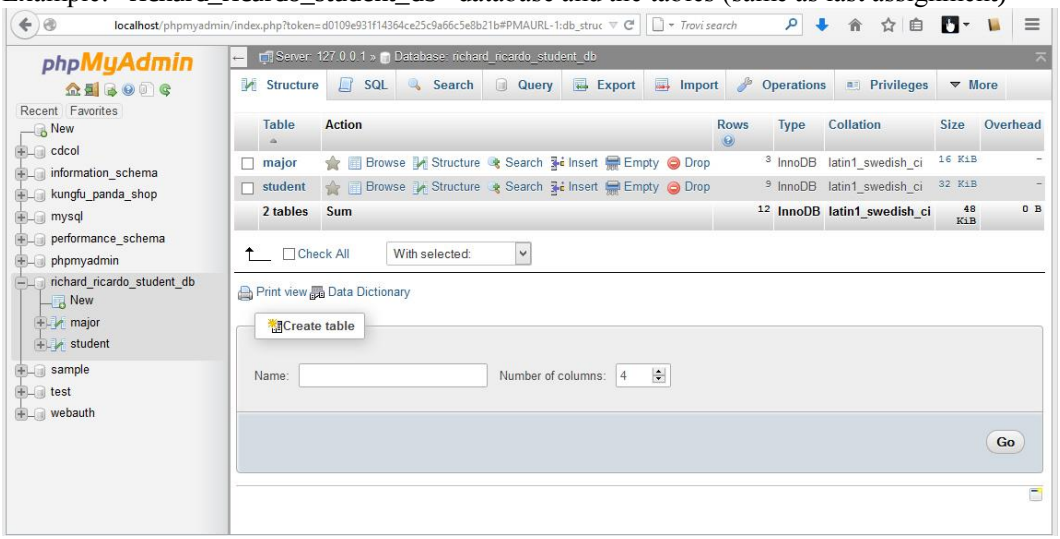
Question 1 – **MVC**: PHP Chapter 5, eg010 step 1 (20 points) **Estimated time: 2 hours**

- In this question you need to
  - Setup MVC folders
  - Re-create the database if necessary
  - Create a database connection file
  - Create a common header file
  - Create a common footer file
  - Create a database error view file (using common header file and common footer file)
  - Create the first php library file for retrieving data from the database (major table)
- Re-load **create_db.sql** in XAMPP > phpMyAdmin again (if necessary) to re-create the database mentioned in the last Assignment. (1 point)
- Save question 1 files in subfolder "**lastname_firstname_assignment5\q1**:
  - create the following subfolders in this folder: (4 points)
    - **lastname_firstname_assignment5\q1\errors**
    - **lastname_firstname_assignment5\q1\model**
    - **lastname_firstname_assignment5\q1\student_manager**
    - **lastname_firstname_assignment5\q1\view**
- Reuse the text file **create_db.sql,** created in the last Assignment, Q1. Copy the text file to folder lastname_firstname_assignment5\. (1 point)
  - Note: In the real world, do NOT put sql files in a website folder. Keep it offline and safe.
- You may reuse the **main.css** file from the last Assignment, Q2. Copy the css file to folder lastname_firstname_assignment5\. (1 point)
- Create file **richard_ricardo_database.php** in folder **lastname_firstname_assignment5\q1\model** (2 points)
  - Hence: lastname_firstname_assignment5\q1**model\richard_ricardo_database.php**
  - Use this file to connect to the database
  - If there is any connect error, redirect to lastname_firstname_assignment5\q1**errors\richard_ricardo_database_error.php**
- Split the original richard_ricardo_database_error.php (from last Assignment, Q2) into 3 files (and edit code) (3 points)
  - lastname_firstname_assignment5\q1**view\richard_ricardo_header.php**
    - common header for all view files
    - link to main.css
  - lastname_firstname_assignment5\q1**view\richard_ricardo_footer.php**
    - common footer for all view files
  - lastname_firstname_assignment5\q1**errors\richard_ricardo_database_error.php**
    - it should display database connection error message
    - it should "include" richard_ricardo_header.php file
    - it should "include" richard_ricardo_footer.php file
- Create file **richard_ricardo_major_db.php** in folder **lastname_firstname_assignment5\q1\model** (2 points)
  - Hence: lastname_firstname_assignment5\q1**model\richard_ricardo_major_db.php**
  - Create this php function in the file
    - `function get_major_name($major_id) { return $major_name; }`
  - This function should take in a major id and return the major name, by querying the database
- Create controller file **index.php** in folder **lastname_firstname_assignment5\q1\student_manager** (6 points)
  - Hence: lastname_firstname_assignment5\q1**student_manager\index.php**
  - "Require" the following two files:
    - /model/richard_ricardo_database.php
    - /model/richard_ricardo_major_db.php
  - Create controller code to handle if `$action == 'list_students'` (first time access)

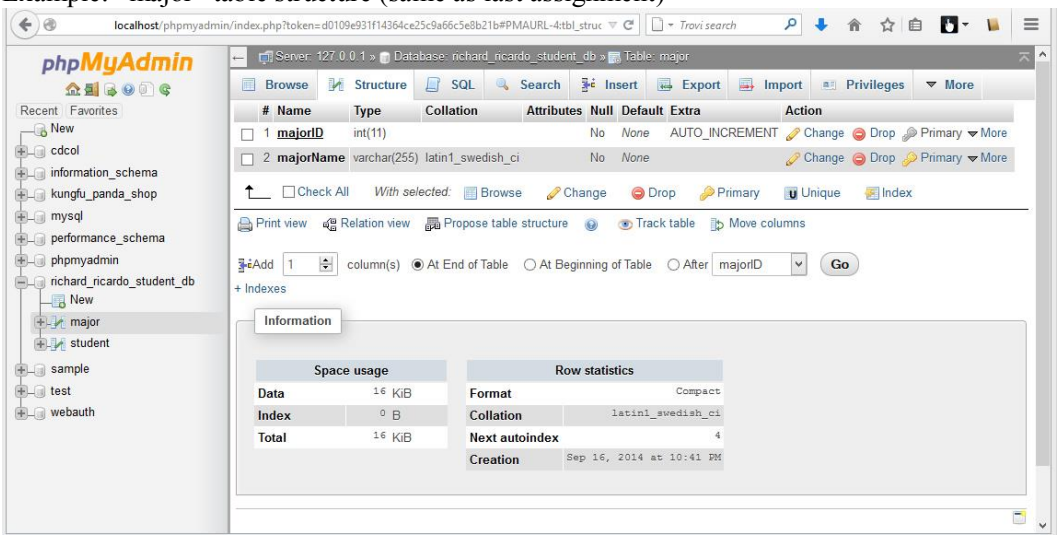
- List $major_id as “1”, and $major_name as “Computer Science”, for testing only

- Files created/updated in this question:
  - lastname_firstname_assignment5\q1\errors\richard_ricardo_database_error.php
  - lastname_firstname_assignment5\q1\model\richard_ricardo_database.php
  - lastname_firstname_assignment5\q1\model\richard_ricardo_major_db.php
  - lastname_firstname_assignment5\q1\student_manager\index.php
  - lastname_firstname_assignment5\q1\view\richard_ricardo_footer.php
  - lastname_firstname_assignment5\q1\view\ richard_ricardo_header.php

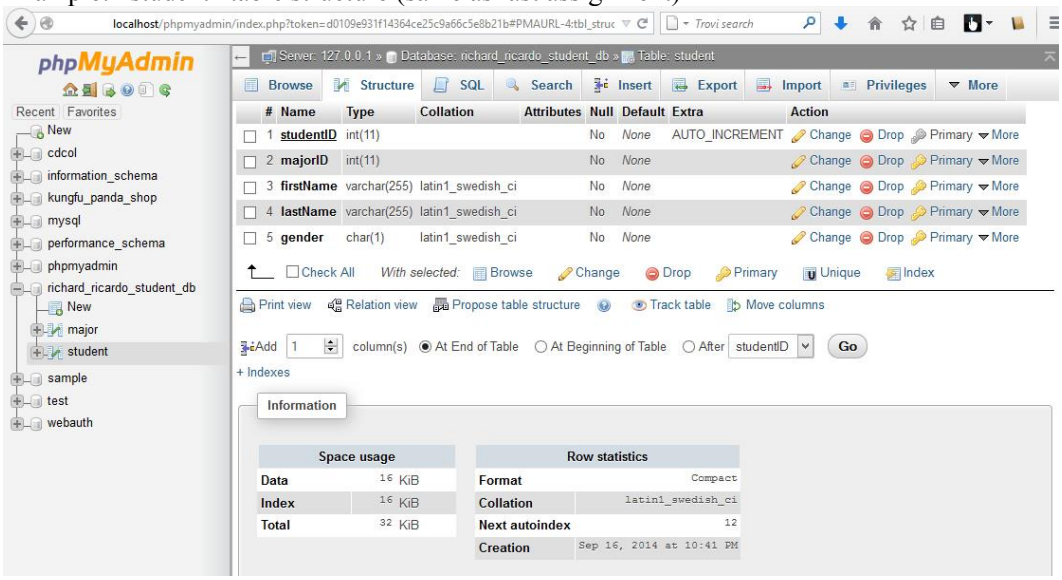
Example: “richard_ricardo_student_db” database and the tables (same as last assignment)



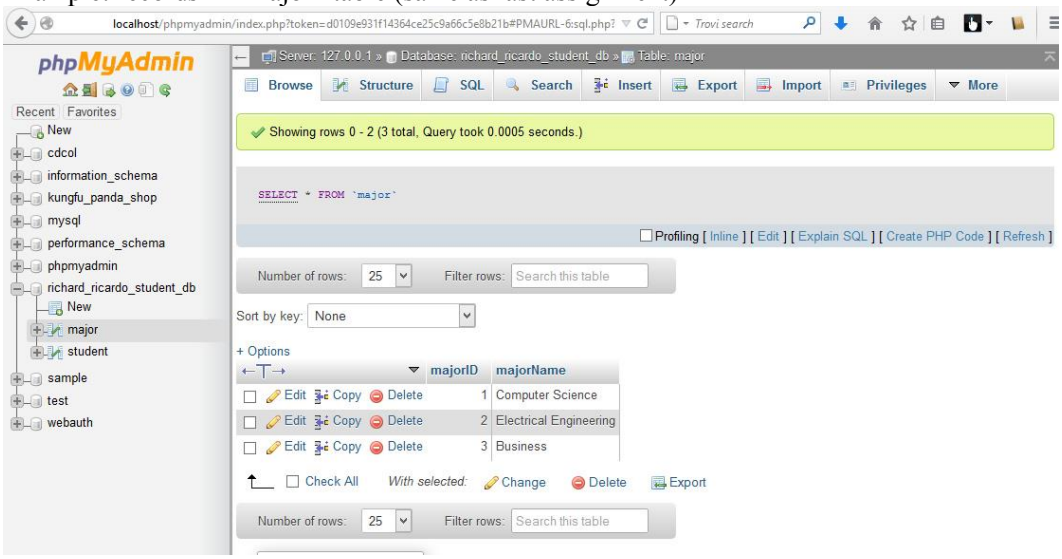
Example: “major” table structure (same as last assignment)



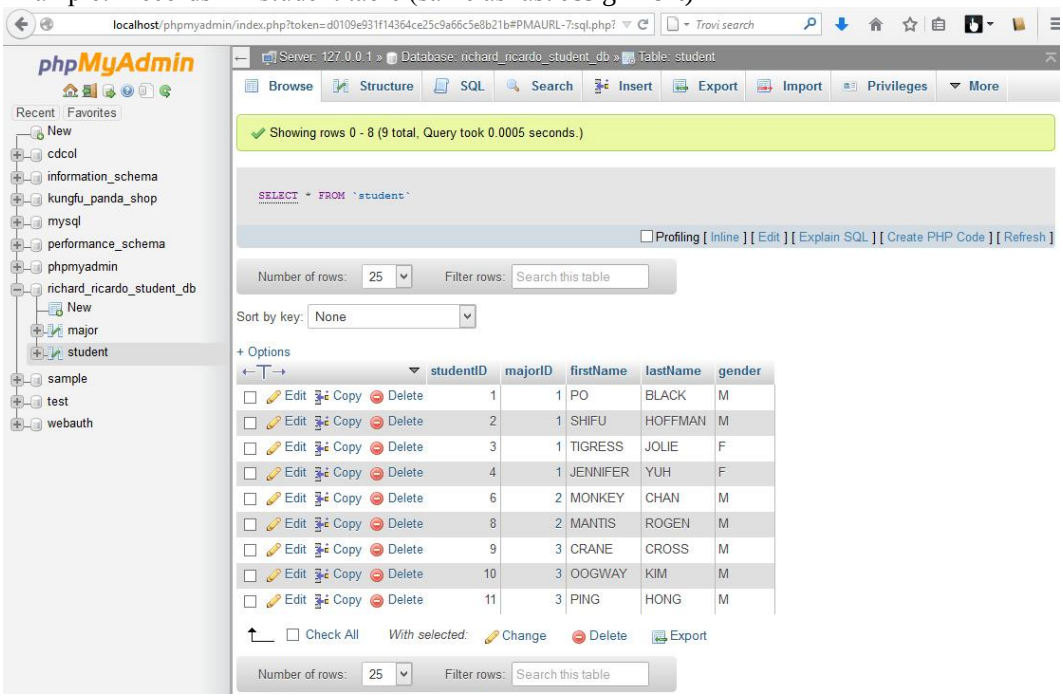
Example: “student” table structure (same as last assignment)



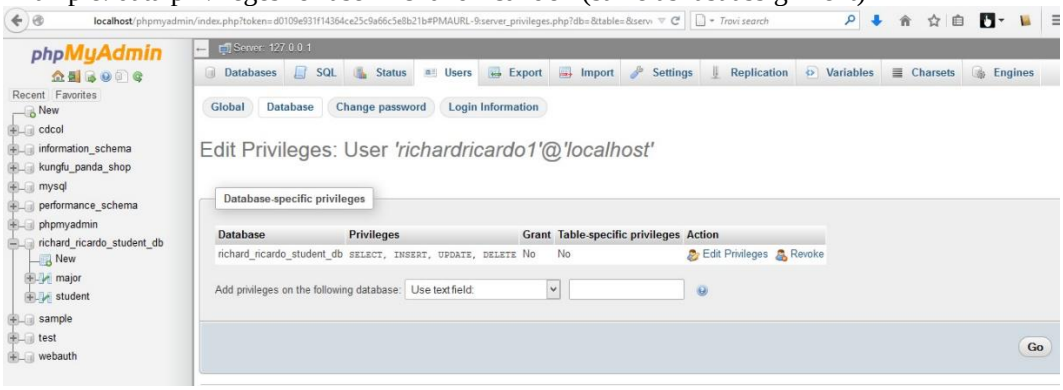
Example: records in “major” table (same as last assignment)



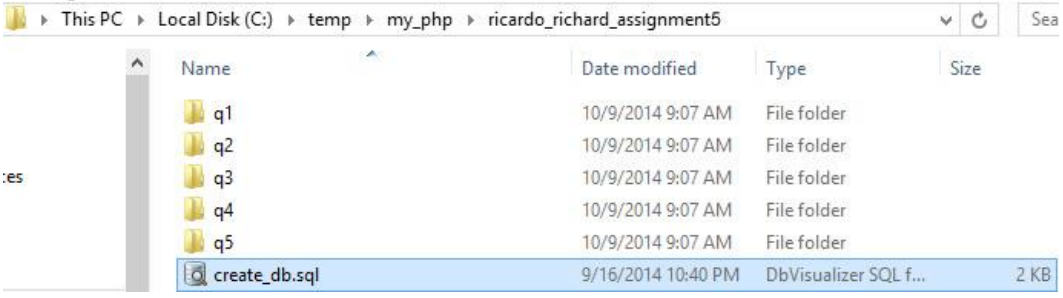
Example: “records” in student table (same as last assignment)



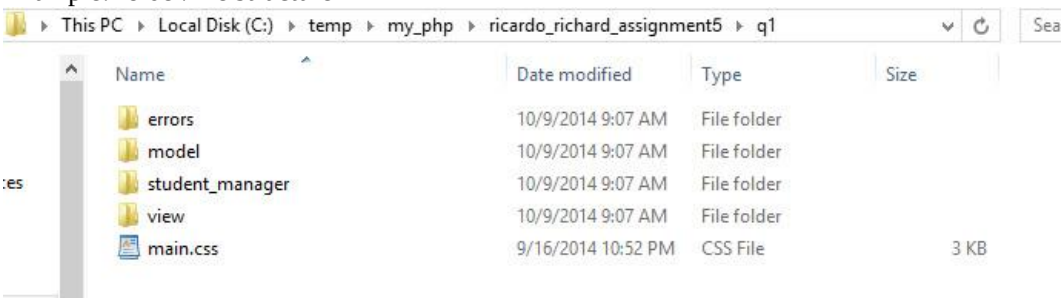
Example: data privileges for user **richardricardo1** (same as last assignment)



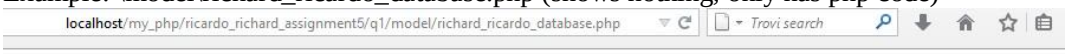
Example: folder/file structure



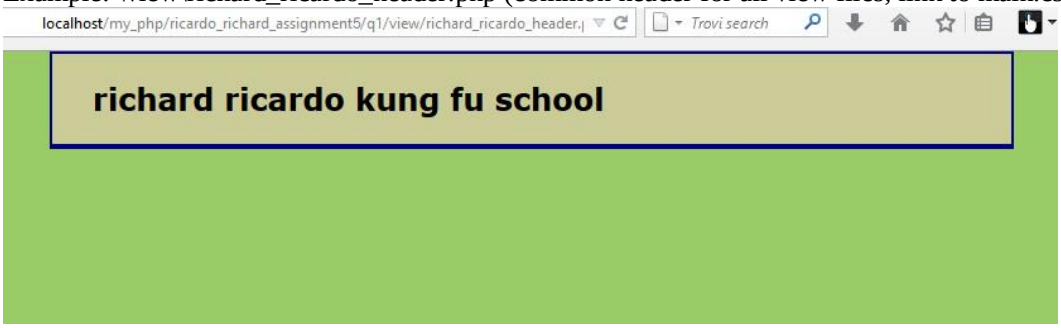
Example: folder/file structure



Example: \model\richard_ricardo_database.php (shows nothing, only has php code)

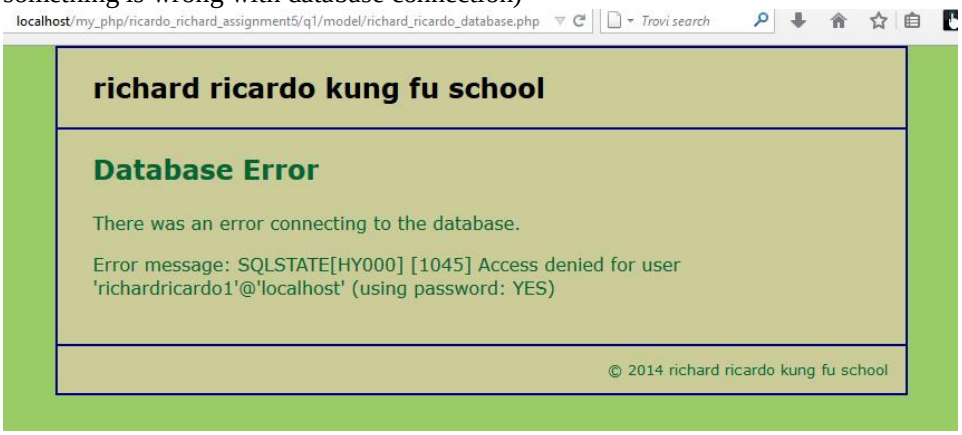


Example: \view\richard_ricardo_header.php (common header for all view files, link to main.css)

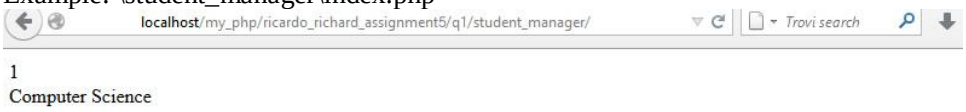


Example: \view\richard_ricardo_footer.php (common footer for all view files)

Example: \model\richard_ricardo_database.php (it shows \errors\richard_ricardo_database_error.php, if something is wrong with database connection)



Example: \student_manager\index.php

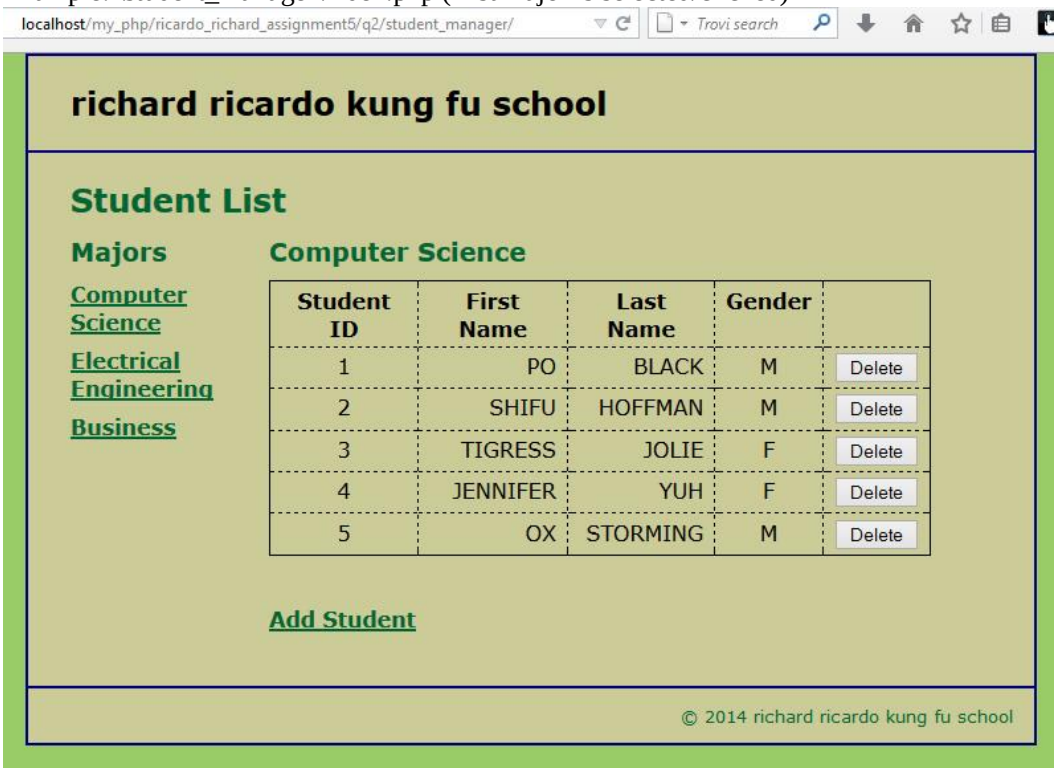


Question 2 – **MVC**: PHP Chapter 5, eg010 step 2 (20 points) **Estimated time: 2 hours**

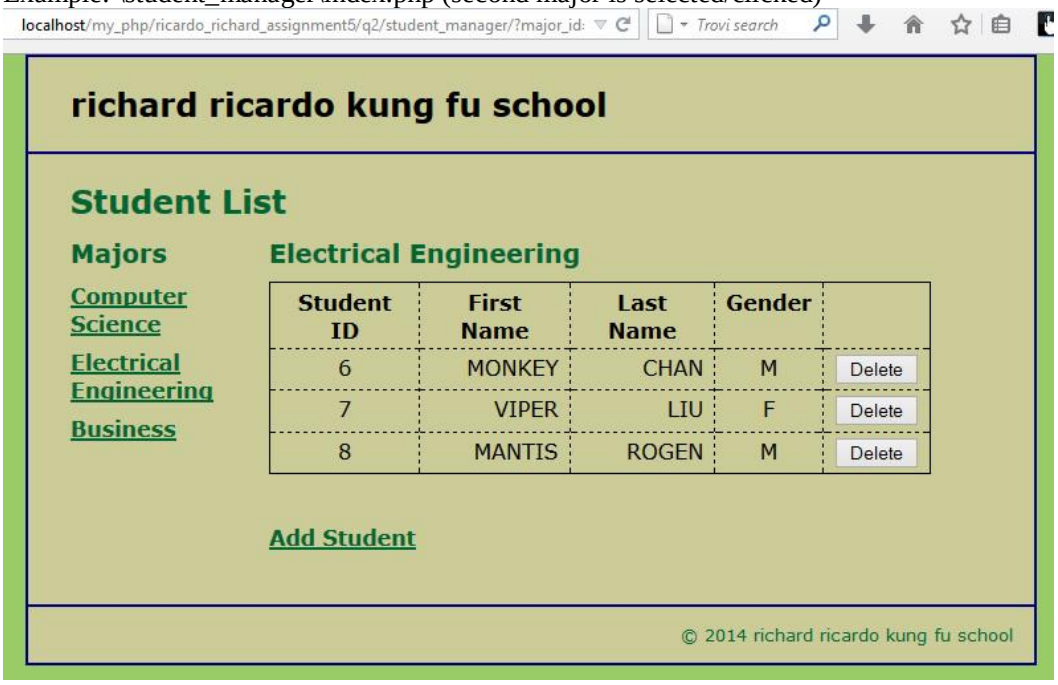
- In this question you need to
  - Create model functions to retrieve major records and student records from the database
  - Update controller code to list students
  - Create code to display students (using common header file and footer file)
- Copy what you have done in question 1 (all folders/files) from the question 1 subfolder to the question 2 subfolder "**lastname_firstname_assignment5\q2**".
- Save question 2 files in subfolder "**lastname_firstname_assignment5\q2**:
  - Add files to / editing existing files in the subfolders:
    - **lastname_firstname_assignment5\q2\errors**
    - **lastname_firstname_assignment5\q2\model**
    - **lastname_firstname_assignment5\q2\student_manager**
    - **lastname_firstname_assignment5\q2\view**
- Update file lastname_firstname_assignment5\q2**model\richard_ricardo_major_db.php** (4 points)
  - Create this php function in the file
    - `function get_majors() { return $majors; }`
  - This function should return all major names sorted by majorID, by querying the database
- Create file **richard_ricardo_student_db.php** in folder **lastname_firstname_assignment5\q2\model** (4 points)
  - Hence: lastname_firstname_assignment5\q2**model\richard_ricardo_student_db.php**
  - Create this php function in the file
    - `function get_students_by_major($major_id){return $students;}`
  - This function should return all student names sorted by studentID, in a particular major (take in a particular majorID), by querying the database
- Update file lastname_firstname_assignment5\q2**student_manager\index.php** (6 points)
  - "Require" the following additional file:

        - /model/richard_ricardo_student.php
    - Update controller code to handle if `$action == 'list_students'` (first time access)
    - Retrieve all students' records by using some functions in
        - model\richard_ricardo_student_db.php
        - model\richard_ricardo_major_db.php
    - "Include" a new file **richard_ricardo_student_list.php** to display all students
- Create file **richard_ricardo_student_list.php** in folder **lastname_firstname_assignment5\q2\student_manager** (6 points)
    - Hence: lastname_firstname_assignment5\q2**student_manager\ richard_ricardo_student_list.php**
    - This file is related to the index.php from the last assignment
    - It should "include" richard_ricardo_header.php file
    - It should "include" richard_ricardo_footer.php file
    - It should display all students by a major
- Files created/updated in this question:
    - lastname_firstname_assignment5\q2\model\ richard_ricardo_major_db.php
    - lastname_firstname_assignment5\q2\model\richard_ricardo_student_db.php
    - lastname_firstname_assignment5\q2\student_manager\index.php
    - lastname_firstname_assignment5\q2\student_manager\richard_ricardo_student_list.php

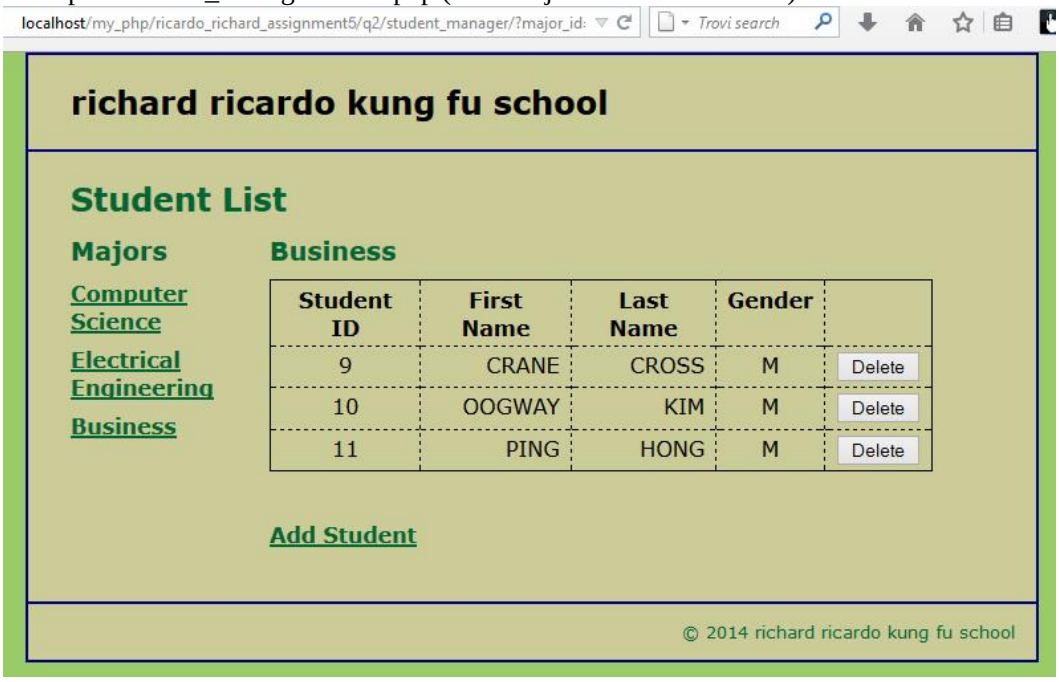
Example: \student_manager\index.php (first major is selected/clicked)



Example: \student_manager\index.php (second major is selected/clicked)



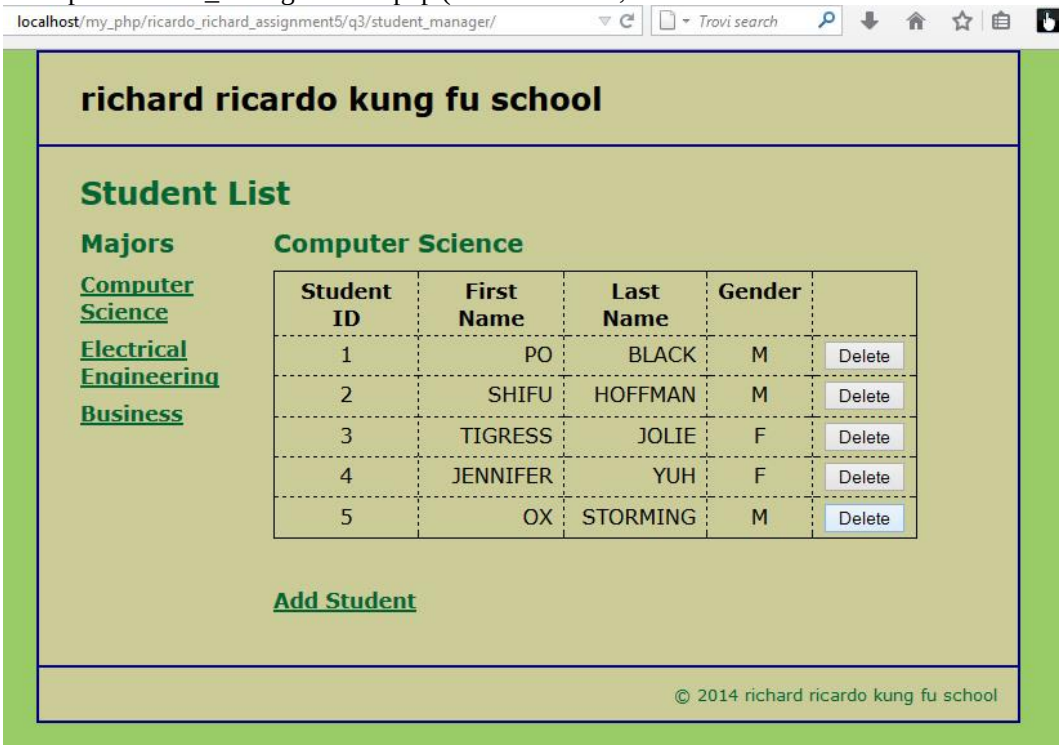
Example: \student_manager\index.php (third major is selected/clicked)



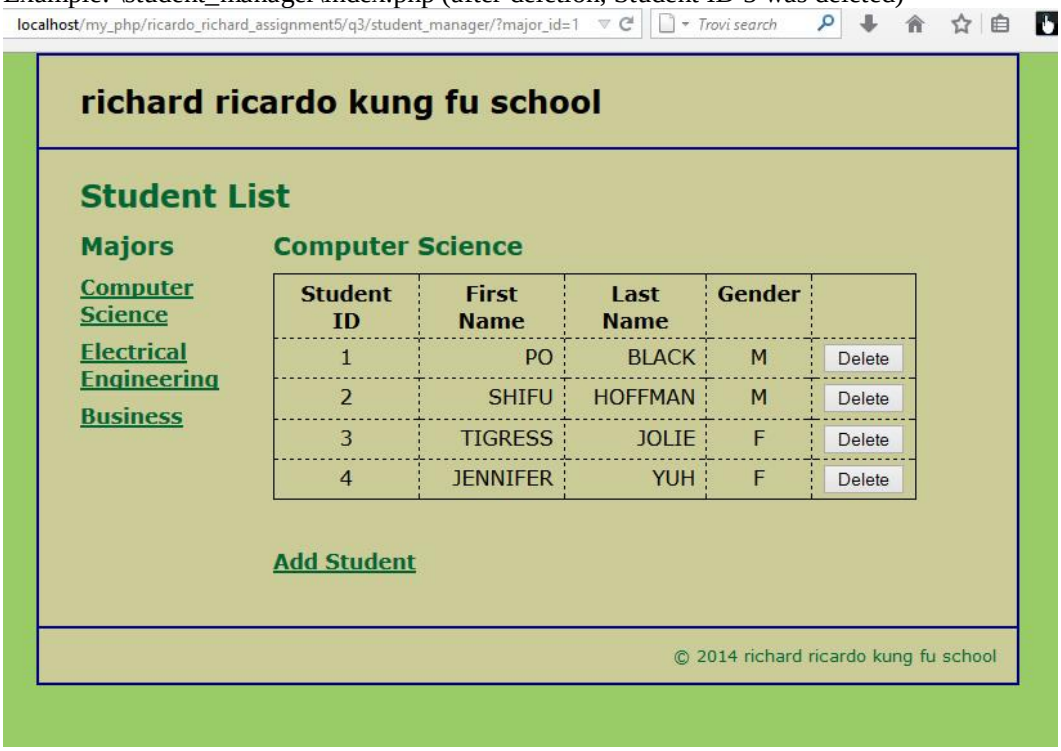
Question 3 – **MVC**: PHP Chapter 5, eg010 step 3 (20 points) **Estimated time: 2 hours**

- In this question you need to
  - Create model function to delete a student record from the database
  - Update controller code to handle student record deletion
- Copy what you have done in question 2 (all folders/files) from the question 2 subfolder to the question 3 subfolder "**lastname_firstname_assignment5\q3**".
- Save question 3 files in subfolder "**lastname_firstname_assignment5\q3**:
  - Add files to / editing existing files in the subfolders:
    - **lastname_firstname_assignment5\q3\errors**
    - **lastname_firstname_assignment5\q3\model**
    - **lastname_firstname_assignment5\q3\student_manager**
    - **lastname_firstname_assignment5\q3\view**
- Update file lastname_firstname_assignment5\q3**student_manager\index.php** (10 points)
  - Update controller code to handle if `$action == 'delete_student'`
  - Delete a student's record by using function `delete_student` in
    - model\richard_ricardo_student_db.php (to be created next)
- Update file lastname_firstname_assignment5\q3\ **model\richard_ricardo_student_db.php** (10 points)
  - Create php function in the file
    - `function delete_student($student_id){}`
    - This function should delete a student record by taking in the student ID, by updating the database
    - This function should be called by index.php
- Files created/updated in this question:
  - lastname_firstname_assignment5\q3\model\richard_ricardo_student_db.php
  - lastname_firstname_assignment5\q3\student_manager\index.php

Example: \student_manager\index.php (before deletion, click on the delete button on Student ID 5)



Example: \student_manager\index.php (after deletion, Student ID 5 was deleted)

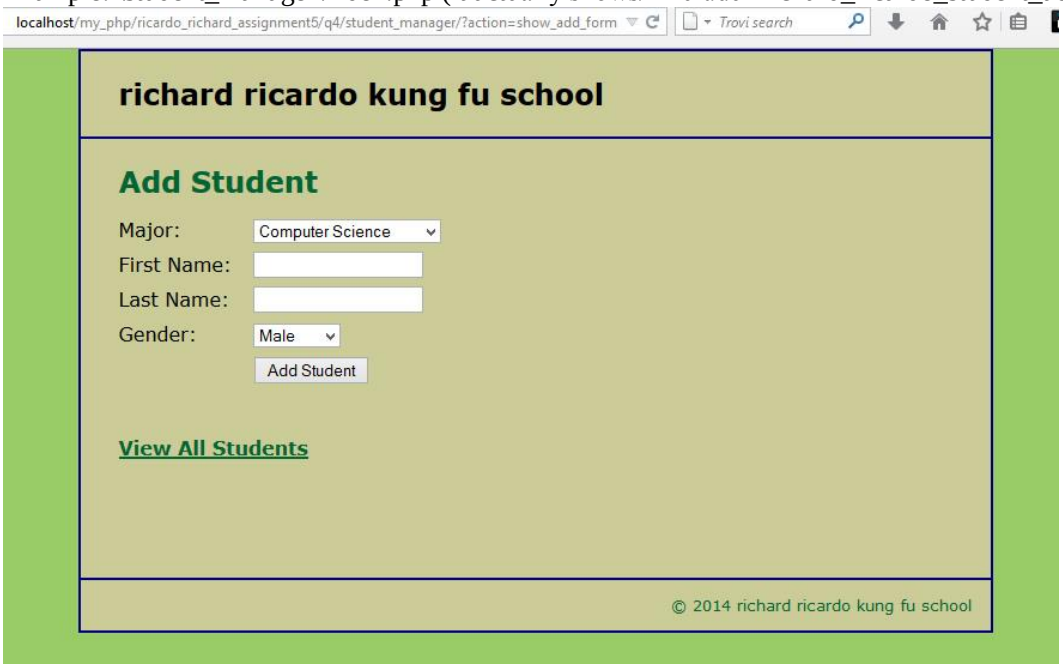


Question 4 – **MVC**: PHP Chapter 5, eg010 step 4 (20 points) **Estimated time: 2 hours**

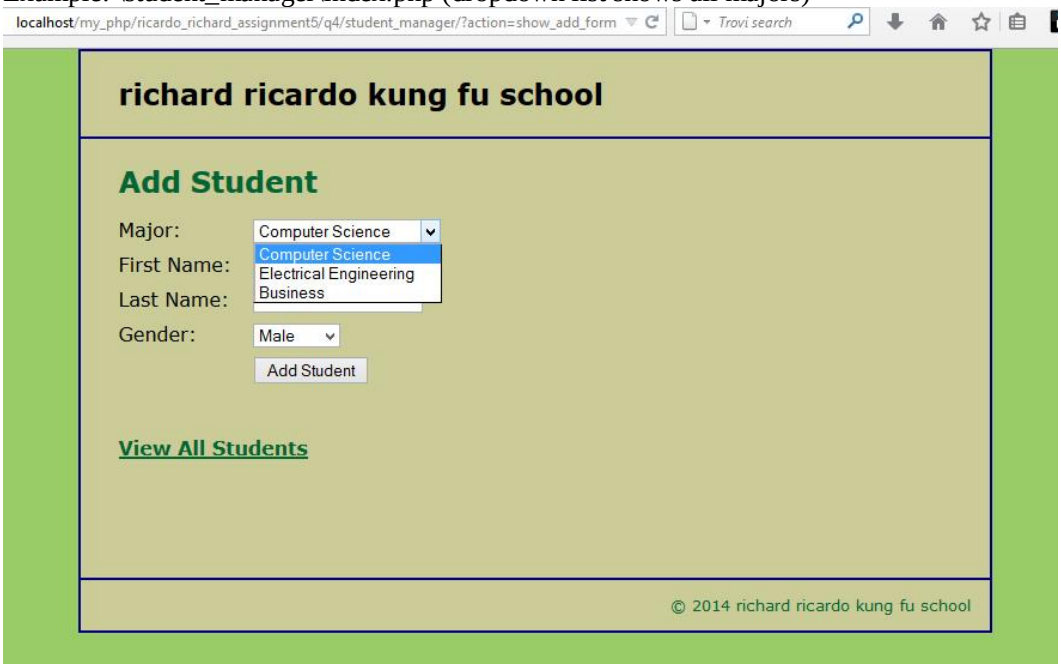
- In this question you need to
  - Update controller code to add a student
  - Create code to display a form to allow adding a student (using common header file and footer file)
- Copy what you have done in question 3 (all folders/files) from the question 3 subfolder to the question 4 subfolder "**lastname_firstname_assignment5\q4**".
- Save question 4 files in subfolder "**lastname_firstname_assignment5\q4**:
  - Add files to / editing existing files in the subfolders:
    - **lastname_firstname_assignment5\q4\errors**
    - **lastname_firstname_assignment5\q4\model**
    - **lastname_firstname_assignment5\q4\student_manager**
    - **lastname_firstname_assignment5\q4\view**
- Update file lastname_firstname_assignment5\q4**student_manager\index.php** (10 points)
  - Update controller code to handle if `$action == 'show_add_form'`
  - Retrieve all majors' records by using a function in
    - model\richard_ricardo_major_db.php
    - The records will be used to create dropdown list in the next step
  - "Include" a new file **richard_ricardo_student_add.php** to display a form to add student
- Create file **richard_ricardo_student_add.php** in folder **lastname_firstname_assignment5\q4\student_manager** (10 points)
  - Hence: lastname_firstname_assignment5\q4**student_manager\ richard_ricardo_student_add.php**
  - This file is related to the richard_richardo_add_student_form.php from the last assignment
  - It should "include" richard_ricardo_header.php file
  - It should "include" richard_ricardo_footer.php file
  - It should display a form to allow users to add a student

- Files created/updated in this question:
  - lastname_firstname_assignment5\q4\student_manager\index.php
  - lastname_firstname_assignment5\q4\student_manager\richard_ricardo_student_add.php

Example: \student_manager\index.php (it actually shows/"include" richard_ricardo_student_add.php)



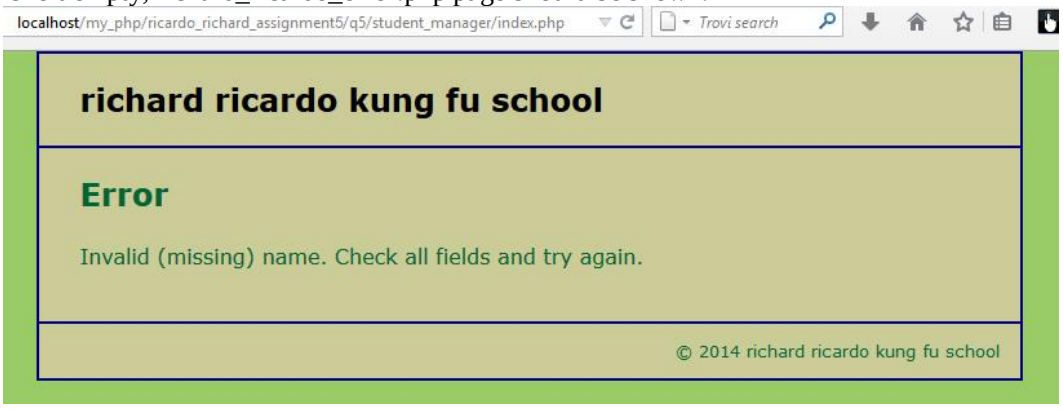
Example: \student_manager\index.php (dropdown list shows all majors)



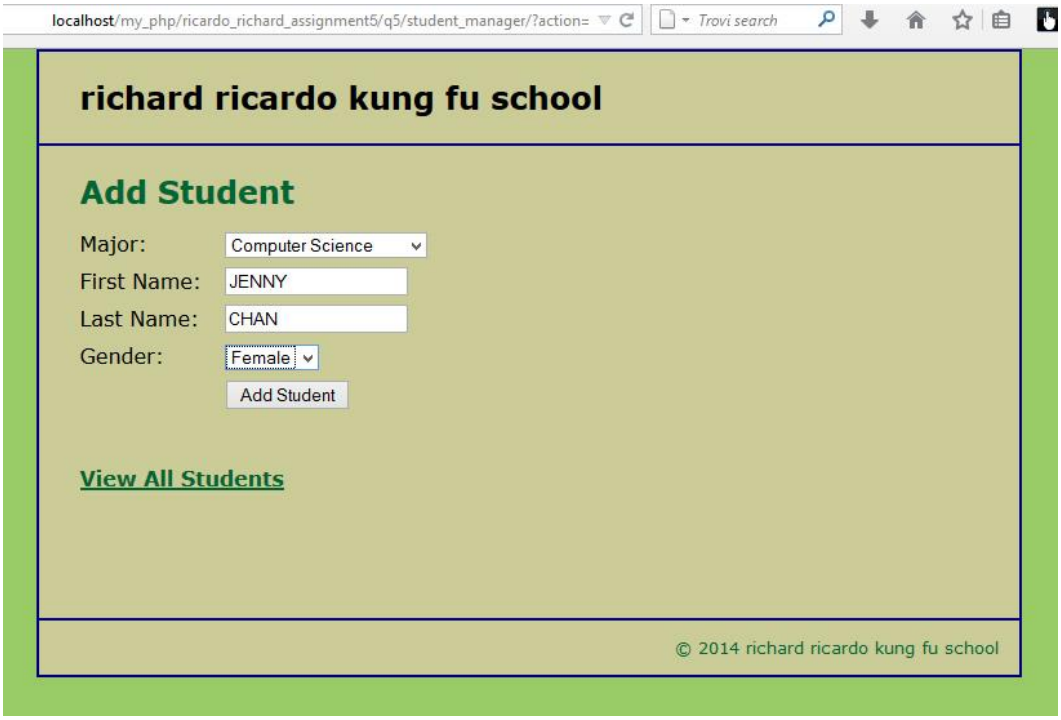
Question 5 – **MVC**: PHP Chapter 5, eg010 step 5 (20 points) **Estimated time: 2 hours**

- In this question you need to
  - Update controller code to add a student
  - Check if user filled in all form fields
  - Create code to display error message (using common header file and footer file) if any form field is left empty
  - Update model code to add a student
- Copy what you have done in question 4 (all folders/files) from the question 4 subfolder to the question 5 subfolder "**lastname_firstname_assignment5\q5**".
- Save question 5 files in subfolder "**lastname_firstname_assignment5\q5**:
  - Add files to / editing existing files in the subfolders:
    - **lastname_firstname_assignment5\q5\errors**
    - **lastname_firstname_assignment5\q5\model**
    - **lastname_firstname_assignment5\q5\student_manager**
    - **lastname_firstname_assignment5\q5\view**
- Update file lastname_firstname_assignment5\q5**student_manager\index.php** (7 points)
  - Update controller code to handle if `$action == 'add_student'`
  - Retrieve all forms filed passed from richard_ricardo_student_add.php (last question)
    - If any form field is empty, create error message and "include" a new file **\errors\richard_ricardo_error.php** to display error message
    - If all form fields are filled, call function `add_student` in
      - model\richard_ricardo_student_db.php (to be created next)
- Create file **richard_ricardo_error.php** in folder **lastname_firstname_assignment5\q5\errors** (6 points)
  - Hence: lastname_firstname_assignment5\q5**errors\ richard_ricardo_error.php**
  - This file is related to the richard_ricardo_error.php from the last assignment
  - It should "include" richard_ricardo_header.php file
  - It should "include" richard_ricardo_footer.php file
  - It should display an error message from index.php
- Update file lastname_firstname_assignment5\q5\ **model\richard_ricardo_student_db.php** (7 points)
  - Create this php function in the file
    - `function add_student($major_id, $firstname, $lastname, $gender){}`
    - This function should add a student record by taking in the above arguments, and by updating the database
  - This function should be called by index.php

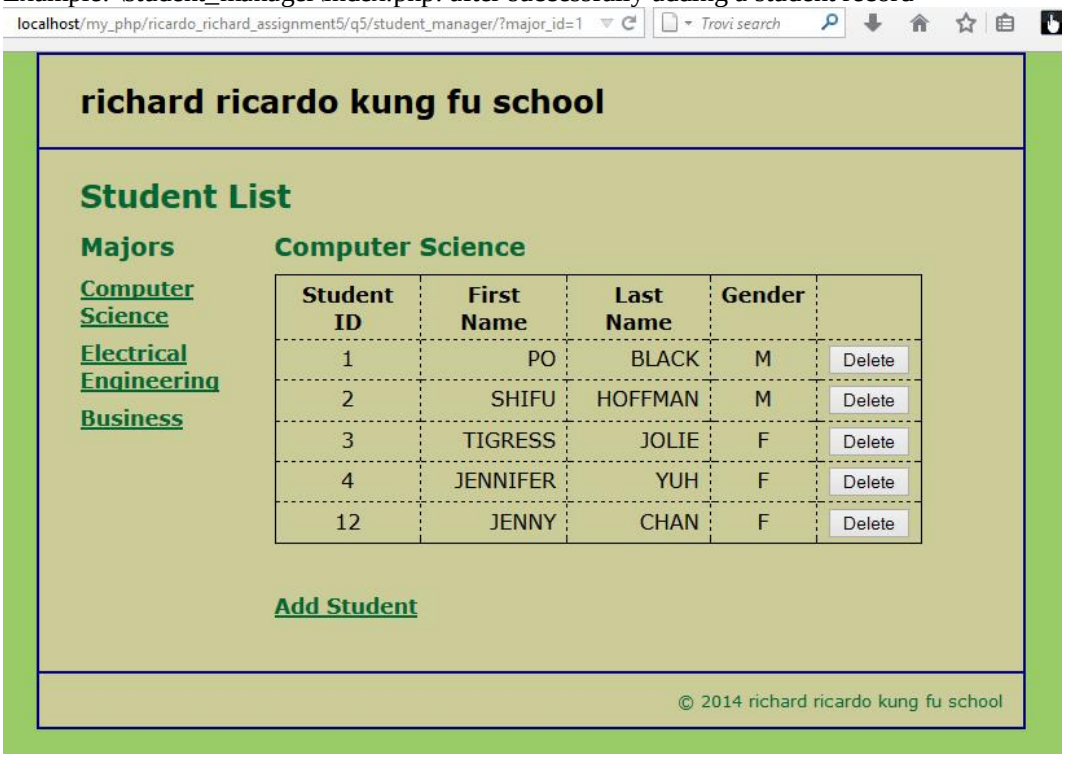
Example: \student_manager\index.php (it actually shows/"include" richard_ricardo_error.php): If any textbox is left empty, richard_ricardo_error.php page should be shown.



Example: \student_manager\index.php (it actually shows/"include" richard_ricardo_student_add.php): user input



Example: \student_manager\index.php: after successfully adding a student record



Important:

1. If you do not put **<your name> / <your first name>** in the above mentioned fields (as shown in the examples), you will get **0 points** for the question(s).
2. **No two students** should submit webpages with exactly the same code, content, layout, or color combination. If found, both students will get **0 points**.
3. Before adding PHP code, all html files must pass html validation at http://validator.w3.org/ without any **error** (and with only 1 warning). Use the validator's "File Upload" tab to check each file.
    a. If you want to validate a PHP file after adding PHP code, you can temporarily rename the *.php file to *.htm file.
    b. When you view page source in a web browser, **`<!DOCTYPE html>`** must be at the top of every page. In other words, all pages must be written in HTML5. **(-20 points** if not)
    c. If any html error is found, 2 points will be deducted for each error. Please validate the files before adding PHP code.
4. All css files must pass css validation at http://jigsaw.w3.org/css-validator/ without any **error**.
5. If your files do not pass the html and css validations, **2 points will be deducted** for **each html or css error** found (no deduction for php error at html validation).
6. Document (comment) your HTML files (`<!-- -->`), CSS files (`/* */`), and PHP files (`/* */` OR `//`). **Points will be taken off** for insufficient comments (`<!-- -->, /* */, //`).

Submission instructions:

- You need to test all document(s).
- Do screen capture(s) of the **input** and the related **output(s)**. Use any graphic editing software (e.g. Microsoft Paint, Adobe Fireworks, GIMP, or Microsoft Expression Design etc) to cut out the browser output (from the screen capture), paste them into a word document.
- Provide **2 different test cases** for each question. In other words, for **each question**, you may need to have **2 input** screen captures and **2 related output** screen captures.
- Do NOT need to do screen capture(s) of html validation results and css validation results for this assignment.
- Save the word document as a pdf file.

You need to submit the following:
1. A pdf file containing the screen capture(s) of the web browser input and output pages, name the file **lastname_firstname_assignment5.pdf**.
2. All html file(s), php file(s), css file(s), and other related files (e.g. image files). Zip your file folder (**lastname_firstname_assignment5**) into a single zip file (or rar file) **lastname_firstname_assignment5.zip**. In the above example, the zip file should contain the following files and subfolders. If there is any image, there should be a \images\ subfolder.

- lastname_firstname_assignment5\q1*.* (all files created in question 1)
- lastname_firstname_assignment5\q2*.* (all files created in question 1 & 2)
- lastname_firstname_assignment5\q3*.* (all files created in question 1, 2 & 3)
- lastname_firstname_assignment5\q4*.* (all files created in question 1, 2, 3 & 4)
- lastname_firstname_assignment5\q5\main.css
- lastname_firstname_assignment5\q5\errors\richard_ricardo_database_error.php
- lastname_firstname_assignment5\q5\errors\richard_ricardo_error.php.php
- lastname_firstname_assignment5\q5\model\richard_ricardo_database.php
- lastname_firstname_assignment5\q5\model\richard_ricardo_major_db.php
- lastname_firstname_assignment5\q5\model\richard_ricardo_student_db.php
- lastname_firstname_assignment5\q5\student_manager\index.php
- lastname_firstname_assignment5\q5\student_manager\richard_ricardo_student_add.php
- lastname_firstname_assignment5\q5\student_manager\richard_ricardo_student_list.php

- lastname_firstname_assignment5\q5\view\richard_ricardo_footer.php
- lastname_firstname_assignment5\q5\view\richard_ricardo_header.php

Please submit the above mentioned **two files** (.pdf and .zip) to D2L digital dropbox.

Grading guidelines (programming questions):

Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile (run) correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down in a reasonable way (e.g. classes and methods)?
- Style (10%): Is the program **indented** properly? Do variables have **meaningful names**?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a **comment** that identifies the author, the course code, and the program date? Are all the classes, methods and data fields clearly **documented (commented)?** Are unclear parts of code **documented (commented)?** (Some items mentioned may not apply to some languages)

A program that does not compile (run) will get at most **50% of the possible points**.