

CSCI 5060 Assignment 4

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

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**.

Please change the provided example’s layout and color combination (color scheme). **If you use (copy) ANY of the provide example’s color combination, 10 points will be deducted (-10 points).**

Create a folder on your hard disk, name the folder **lastname_firstname_assignment4**. 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 (`<!-- -->`, `/* */`, `//`).

- 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>`.
- Before adding PHP code, all html files must pass html validation at <http://validator.w3.org/> without any **error** (and with only 1 warning).
- After adding PHP code, the generated html code (Firefox web browser > right-click > view page source) must also pass html validation at <http://validator.w3.org/> without any **error** (and with only 1 warning).
- 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 – Database: PHP Chapter 4, eg009 and knowledge of SQL (20 points) **Estimated time: 3 hours**

- Save question 1 files in folder “**lastname_firstname_assignment4**”: (2 points)
 - **create_db.sql**
- Create a text file **create_db.sql**, write sql statements in the file to
 - Create a MySQL database **richard_ricardo_universe_db**. (2 points)
 - In the database, create 2 tables. (2 points)
 - country
 - hero_character

- Create the following fields (columns) for the tables (refer to examples below for details).
 - country table: citizenship, country_name (2 points)
 - hero_character table: hero_id, name, real_name, citizenship (2 points)
- Do foreign key referencing, to maintain referential integrity (3 points)
 - One character can only have one country.
 - One country can be used by many characters.
 - citizenship is the primary key of the country table.
 - hero_id is the primary key of the hero_character table.
 - citizenship is a foreign key of the hero_character table, for any values of citizenship appearing in the hero_character table, these values must also appear in the country table.
- Insert test records to country table and hero_character table. (3 points)
- Create a MySQL database username **richardinternet** with password **richardcinema**, with data privileges (select, insert, update, delete) for the **richard_ricardo_universe_db** database. (4 points)
- All above must be done by SQL statements in the text file create_db.sql. (0 points if not)
- Load **create_db.sql** in XAMPP > phpMyAdmin to create the above mentioned database.
- Note: In the real world, do NOT put sql files in a website folder. Keep it offline and safe.

Example: “richard ricardo universe db” database and the tables

The screenshot shows the phpMyAdmin interface for the 'richard_ricardo_universe_db' database. The left sidebar displays the database structure, including a 'New' button and a list of databases: information_schema, leonda, mysql, performance_schema, phpmyadmin, richard, and test. The 'richard' database is expanded, showing 'richard_ricardo_avenger_db' and 'richard_ricardo_universe_db'. The 'richard_ricardo_universe_db' database is selected, and its structure is shown in the main panel. The 'country' and 'hero_character' tables are listed. The 'country' table has 4 rows, and the 'hero_character' table has 14 rows. The 'Sum' row shows a total of 18 rows. The 'Create table' form is visible at the bottom, with the 'Name' field empty and the 'Number of columns' set to 4.

Table	Action	Rows	Type	Collation	Size	Overhead
country	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 K1B	-
hero_character	Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_general_ci	32.0 K1B	-
2 tables	Sum	18	InnoDB	utf8mb4_general_ci	48.0 K1B	0 B

Example: “country” table structure

The screenshot shows the phpMyAdmin interface for the 'country' table. The left sidebar displays the database structure, including 'richard_ricardo_universe_db' and its tables: 'country' and 'hero_character'. The main panel shows the 'Table structure' view for the 'country' table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	citizenship	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
2	country_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there are options to 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. There are also buttons for 'Print', 'Propose table structure', 'Move columns', and 'Normalize'.

The 'Indexes' section shows a primary index on the 'citizenship' column:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit	PRIMARY	BTREE	Yes	No	citizenship	4	A	No	

At the bottom, there is a 'Partitions' section and a console message: 'Partitioning defined!'.

Example: “hero_character” table structure

The screenshot shows the phpMyAdmin interface for the 'hero_character' table. The left sidebar displays the database structure, including 'richard_ricardo_universe_db' and its tables: 'country' and 'hero_character'. The main panel shows the 'Table structure' view for the 'hero_character' table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	hero_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	real_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	citizenship	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there are options to 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. There are also buttons for 'Print', 'Propose table structure', 'Move columns', and 'Normalize'.

The 'Indexes' section shows two indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit	PRIMARY	BTREE	Yes	No	hero_id	14	A	No	
Edit	citizenship	BTREE	No	No	citizenship	14	A	No	

At the bottom, there is a 'Partitions' section and a console message: 'Partitioning defined!'.

The screenshot shows the phpMyAdmin web interface. The browser address bar indicates the URL is localhost/phpmyadmin/index.php?route=/sql&server=1&db=richard_ricardo_universe_db&table=country&pos=0. The interface is divided into several sections:

- Left Sidebar:** Contains a tree view of the database structure. The 'richard_ricardo_universe_db' database is selected, and the 'country' table is highlighted.
- Top Navigation Bar:** Includes tabs for 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Triggers'. The 'Browse' tab is active.
- Main Content Area:**
 - A status bar at the top indicates 'Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)'.
 - A SQL query editor shows the query: `SELECT * FROM `country``.
 - Below the query editor are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (a search box), and 'Sort by key' (set to None).
 - A table view displays the data from the 'country' table. The table has two columns: 'citizenship' and 'country_name'.

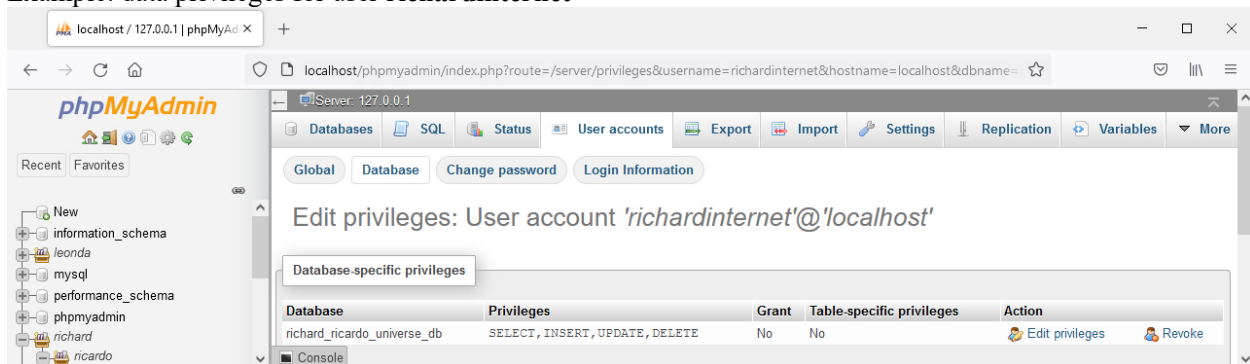
				<u>citizenship</u>	<u>country_name</u>
☐				American	United States of America
☐				Asgardian	Asgard
☐				Sokovian	Sokovia
☐				Wakandan	Wakanda

At the bottom of the main content area, there are controls for 'Check all', 'With selected' (with edit, copy, and delete icons), and an 'Export' button.

The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a tree view of the database structure. The main area displays the 'hero_character' table with 14 rows. The table has columns: hero_id, name, real_name, and citizenship. The rows list various Marvel characters like Iron Man, Scarlet Witch, Thor, Hulk, etc. The interface includes a top navigation bar with buttons like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Triggers'. A status bar at the bottom shows 'Showing rows 0 - 13 (14 total, Query took 0.0002 seconds.)' and a SQL query: 'SELECT * FROM `hero\_character`'.

hero_id	name	real_name	citizenship
1	Iron Man	Anthony Edward Stark	American
2	Scarlet Witch	Wanda Maximoff	Sokovian
3	Thor	Thor Odinson	Asgardian
4	Hulk	Robert Bruce Banner	American
5	Ant-Man	Scott Edward Harris Lang	American
6	Spider-Man	Peter Benjamin Parker	American
7	Star-Lord	Peter Jason Quill	American
8	Doctor Strange	Stephen Vincent Strange	American
9	Valkyrie	Brunnhilde	Asgardian
10	Black Panther	TChalla	Wakandan
11	Quicksilver	Pietro Maximoff	Sokovian
12	Okoye	Okoye	Wakandan
13	Nakia	Nakia	Wakandan
14	Shuri	Shuri	Wakandan

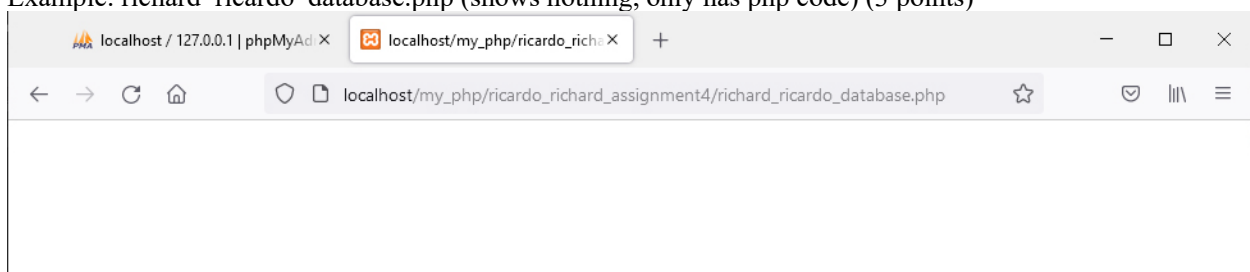
Example: data privileges for user **richardinternet**



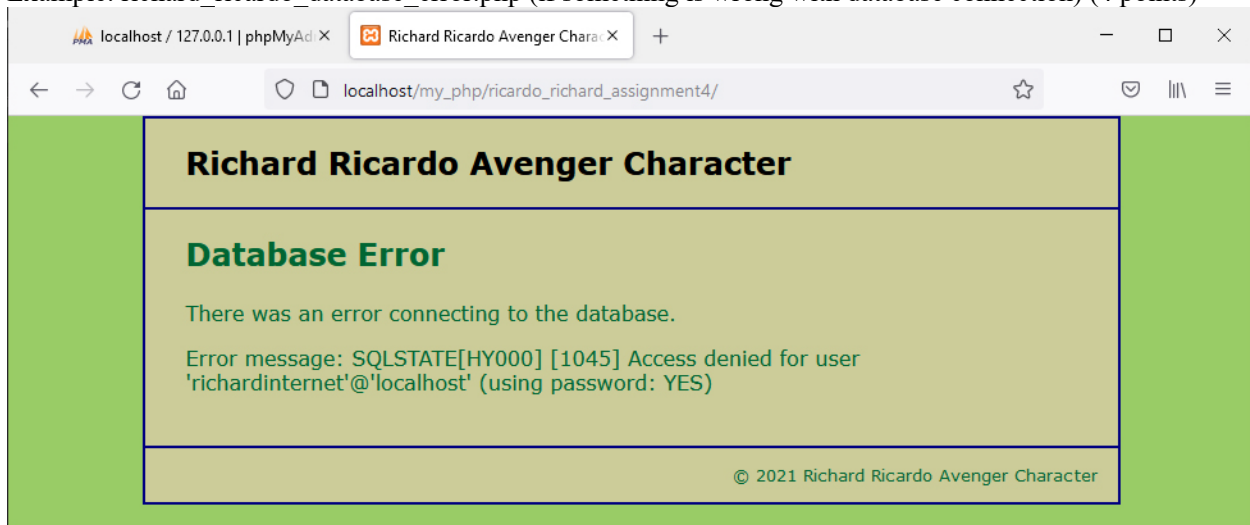
Question 2 – Display: PHP Chapter 4, eg009 and knowledge of SQL (35 points) Estimated time: 4 hours

- Save question 2 files in “**lastname_firstname_assignment4**”: (4 points)
 - index.php**
 - main.css**
 - richard_ricardo_database.php**
 - richard_ricardo_database_error.php**
- Create web page(s) that displays the default information (example shown).
- The initial page and related outputs should look like the examples shown below.
- Create your page(s) using “Richard Ricardo Avenger Character” as the page title(s) (<title> tag). (2 points)
- Save the first page as index.php. (1 point)
- Create file richard_ricardo_database.php (php code only, no html code) to connect to database
 - Use PDO (PHP Data Object) to interface with your MySQL database.
 - Connect to the **richard_ricardo_universe_db** MySQL database (created in Q1), using username **richardinternet** with password **richardcinema**.
 - Use try {} catch () {} exception handling to detect if the connection is successful.
 - If the connection is not successful, show an error page **richard_ricardo_database_error.php**.
- Create a css file named **main.css** to format all pages by creating your own layout (no two students should have the same layout). You should use **the same css file** to format all questions. (6 points)
- index.php
 - Uses **richard_ricardo_database.php** to connect to the database.
 - The sidebar on the left displays a list of records from the “country” table (created in Q1).
 - The main portion of the page lists the records from the “hero_character” table (created in Q1), for that country.
 - If a user wants to view the characters in another country, the user can click on the link for that country.
 - The display on the main portion of the page changes accordingly as shown in examples.

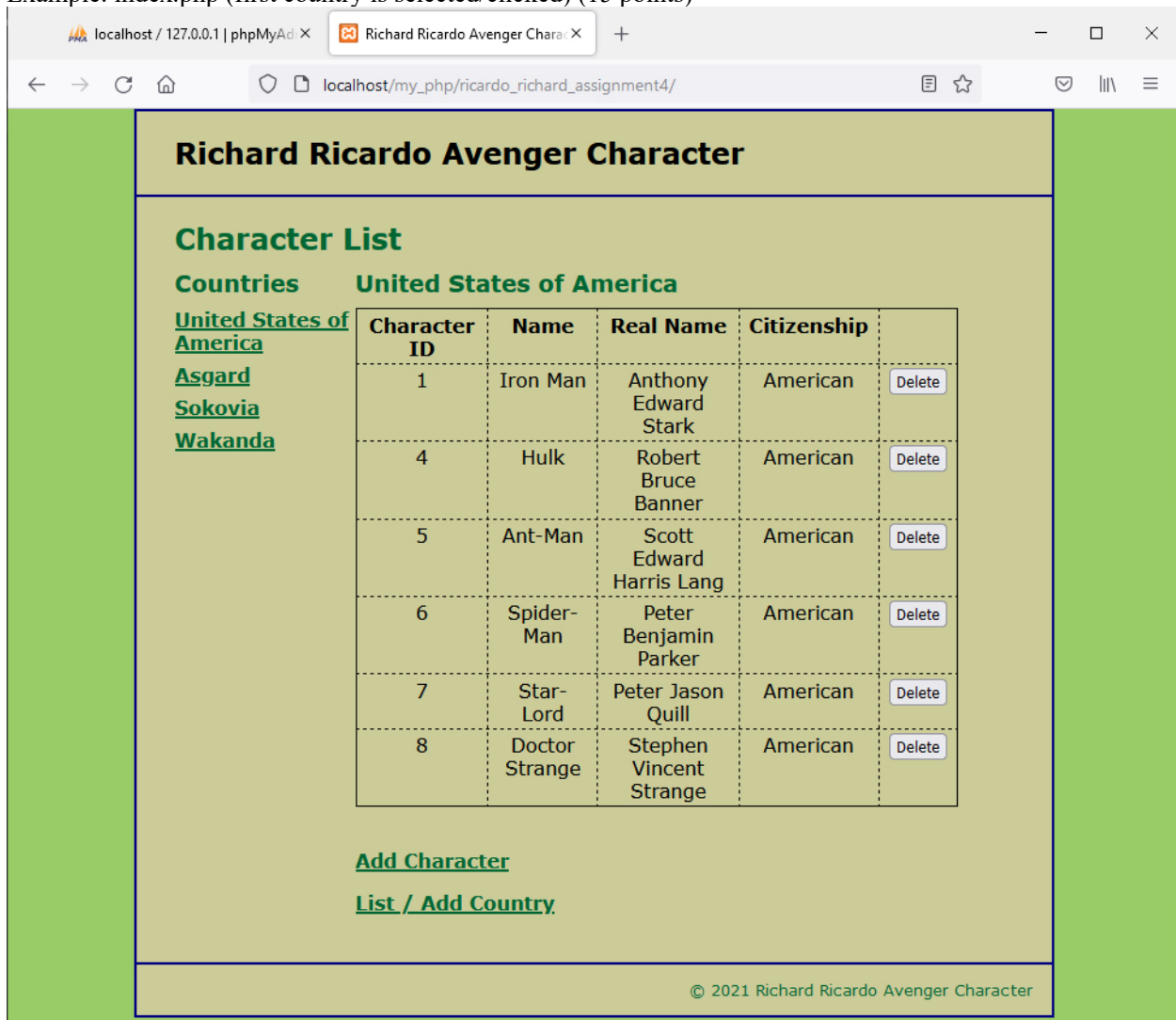
Example: richard_ricardo_database.php (shows nothing, only has php code) (3 points)



Example: richard_ricardo_database_error.php (if something is wrong with database connection) (4 points)



Example: index.php (first country is selected/clicked) (15 points)



Example: index.php (second country is selected/clicked)

The screenshot shows a web browser window with the URL `localhost/my_php/ricardo_richard_assignment4/?citizenship=Asgardian`. The page title is "Richard Ricardo Avenger Character". Under the heading "Character List", there is a section for "Countries" with links for "United States of America", "Asgard", "Sokovia", and "Wakanda". The "Asgard" link is highlighted. Below the links is a table with the following data:

Character ID	Name	Real Name	Citizenship	
3	Thor	Thor Odinson	Asgardian	Delete
9	Valkyrie	Brunnhilde	Asgardian	Delete

Below the table are links for "Add Character" and "List / Add Country". At the bottom right, there is a copyright notice: "© 2021 Richard Ricardo Avenger Character".

Example: index.php (third country is selected/clicked)

The screenshot shows a web browser window with the URL `localhost/my_php/ricardo_richard_assignment4/?citizenship=Sokovian`. The page title is "Richard Ricardo Avenger Character". Under the heading "Character List", there is a section for "Countries" with links for "United States of America", "Asgard", "Sokovia", and "Wakanda". The "Sokovia" link is highlighted. Below the links is a table with the following data:

Character ID	Name	Real Name	Citizenship	
2	Scarlet Witch	Wanda Maximoff	Sokovian	Delete
11	Quicksilver	Pietro Maximoff	Sokovian	Delete

Below the table are links for "Add Character" and "List / Add Country". At the bottom right, there is a copyright notice: "© 2021 Richard Ricardo Avenger Character".

Example: index.php (fourth country is selected/clicked)

Richard Ricardo Avenger Character

Character List

Countries **Wakanda**

[United States of America](#)

[Asgard](#)

[Sokovia](#)

[Wakanda](#)

Character ID	Name	Real Name	Citizenship	
10	Black Panther	TChalla	Wakandan	Delete
12	Okoye	Okoye	Wakandan	Delete
13	Nakia	Nakia	Wakandan	Delete
14	Shuri	Shuri	Wakandan	Delete

[Add Character](#)

[List / Add Country](#)

© 2021 Richard Ricardo Avenger Character

Question 3 – Delete Character: PHP Chapter 4, eg009 and SQL (20 points)

Estimated time: 3 hours

- Save question 3 files in folder “**lastname_firstname_assignment4**”. (2 points)
 - **richard_ricardo_delete_character.php**
- The initial page and related outputs should look like the examples shown below.
- Create your page(s) using “Richard Ricardo Avenger Character” as the page title(s) (<title> tag). (2 points)
- When a user clicks on a “Delete” link on index.php, (16 points)
 - the user should be directed to **richard_ricardo_delete_character.php**.
 - richard_ricardo_delete_character.php uses **richard_ricardo_database.php** to connect to the database.
 - the selected character record will be deleted from the database table “hero_character”.
 - index.php will be shown again with updated records.

Example: index.php, before deletion, click on the delete button on Character ID 14

Richard Ricardo Avenger Character

Character List

Countries **Wakanda**

[United States of America](#)

[Asgard](#)

[Sokovia](#)

[Wakanda](#)

Character ID	Name	Real Name	Citizenship	
10	Black Panther	TChalla	Wakandan	Delete
12	Okoye	Okoye	Wakandan	Delete
13	Nakia	Nakia	Wakandan	Delete
14	Shuri	Shuri	Wakandan	Delete

[Add Character](#)

[List / Add Country](#)

© 2021 Richard Ricardo Avenger Character

Example: After deletion, Character ID 14 was deleted. Now click on the delete button on Character ID 13

Richard Ricardo Avenger Character

Character List

Countries **Wakanda**

[United States of America](#)

[Asgard](#)

[Sokovia](#)

[Wakanda](#)

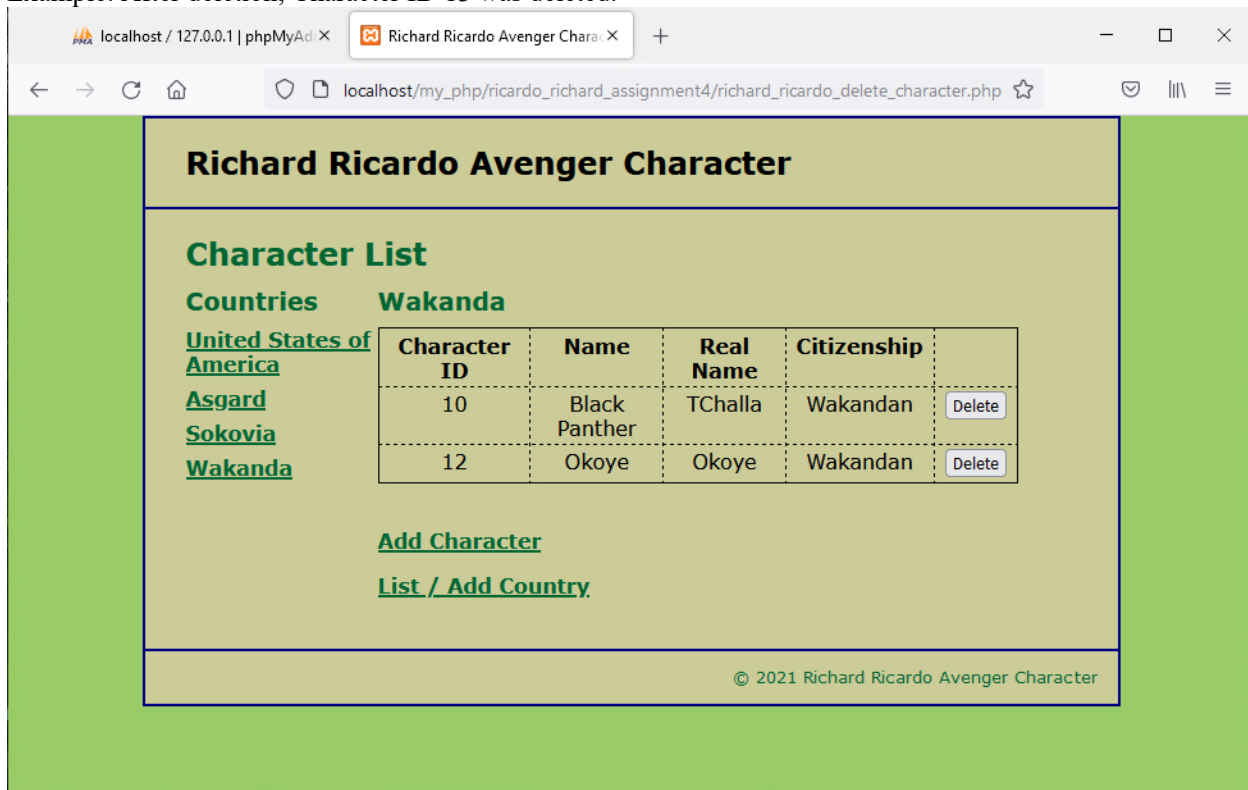
Character ID	Name	Real Name	Citizenship	
10	Black Panther	TChalla	Wakandan	Delete
12	Okoye	Okoye	Wakandan	Delete
13	Nakia	Nakia	Wakandan	Delete

[Add Character](#)

[List / Add Country](#)

© 2021 Richard Ricardo Avenger Character

Example: After deletion, Character ID 13 was deleted.



Question 4 – Add Character: PHP Chapter 4, eg009 and SQL (25 points)

Estimated time: 4 hours

- Save question 4 files in folder “lastname_firstname_assignment4”. (2 points)
 - richard_ricardo_add_character_form.php
 - richard_ricardo_add_character.php
 - richard_ricardo_error.php
- The initial page and related outputs should look like the examples shown below.
- Create your page(s) using “Richard Ricardo Avenger Character” as the page title(s) (<title> tag). (2 points)
- When a user clicks on “Add Character” link on index.php, (21 points)
 - the user should be directed to richard_ricardo_add_character_form.php.
 - a link “View All Characters” gives the user option to go back to index.php.
 - richard_ricardo_add_character_form.php uses **richard_ricardo_database.php** to connect to the database.
 - A dropdown list will show all countries by retrieve records from “country” database table (created in Q1).
 - After the user keyed in new character data, and press the submit (Add Character) button, richard_ricardo_add_character.php will be used to add record to the “hero_character” database table, and index.php will be shown again with updated records.
 - richard_ricardo_add_character.php uses **richard_ricardo_database.php** to connect to the database.
 - If the user left any textbox empty, richard_ricardo_error.php page will be shown.

Example: richard_ricardo_add_character_form.php

localhost / 127.0.0.1 | phpMyAdmin × Richard Ricardo Avenger Character ×

localhost/my_php/ricardo_richard_assignment4/richard_ricardo_add_character_form.php

Richard Ricardo Avenger Character - Add Character

Add Character

Country:

Name:

Real Name:

[View All Characters](#)

© 2021 Richard Ricardo Avenger Character

Example: richard_ricardo_add_character_form.php, user input

localhost / 127.0.0.1 | phpMyAdmin × Richard Ricardo Avenger Character ×

localhost/my_php/ricardo_richard_assignment4/richard_ricardo_add_character_form.php

Richard Ricardo Avenger Character - Add Character

Add Character

Country:

Name:

Real Name:

[View All Characters](#)

© 2021 Richard Ricardo Avenger Character

Example: after successfully adding record

Richard Ricardo Avenger Character

Character List

Countries **Wakanda**

[United States of America](#)

[Asgard](#)

[Sokovia](#)

[Wakanda](#)

Character ID	Name	Real Name	Citizenship	
10	Black Panther	TChalla	Wakandan	Delete
12	Okoye	Okoye	Wakandan	Delete
15	Erik Killmonger	N'Jadaka	Wakandan	Delete

[Add Character](#)

[List / Add Country](#)

© 2021 Richard Ricardo Avenger Character

Example: richard_ricardo_error.php page: If any textbox is left empty, richard_ricardo_error.php page will be shown.

Richard Ricardo Avenger Character

Error

Invalid input data. Check all fields and try again.

[View All Characters](#)

© 2021 Richard Ricardo Avenger Character

Question 5 Extra Credit – List / Add Country: (20 points)**Estimated time: 4 hours**

- Save question 5 files in folder “lastname_firstname_assignment4”. (1 point)
 - richard_ricardo_country_list.php
 - richard_ricardo_delete_country.php
 - richard_ricardo_add_country.php
- The initial page and related outputs should look like the examples shown below.
- Create your page(s) using “Richard Ricardo Avenger Character” as the page title(s) (<title> tag). (1 point)
- When a user clicks on a “List / Add Country” link on index.php, (2 points)
 - the user should be directed to **richard_ricardo_country_list.php**.
 - richard_ricardo_country_list.php uses **richard_ricardo_database.php** to connect to the database.
 - records from the “country” table will be displayed.
- When the user click on “Add Country” button without keying in anything, richard_ricardo_error.php page (created in Q4) will be shown. (4 points)
- When the user key in some country, and click on “Add” button, (4 points)
 - the user should be directed to richard_ricardo_add_country.php.
 - richard_ricardo_add_country.php uses **richard_ricardo_database.php** to connect to the database.
 - user input will be added to “country” database table.
 - richard_ricardo_country_list.php will be shown again with updated records.
- When the user click on a “Delete” link on richard_ricardo_country_list.php, (8 points)
 - the user should be directed to **richard_ricardo_delete_country.php**.
 - richard_ricardo_delete_country.php uses **richard_ricardo_database.php** to connect to database.
 - the selected record will be deleted from the database table “country”.
 - richard_ricardo_country_list.php will be shown again with updated records.
 - Due to foreign key referencing, user should not be able to delete any “country” record with “hero_character” record linked to it (Q1).

Example: richard_ricardo_country_list.php

Richard Ricardo Avenger Character - Countries

Country List

Citizenship	Country	
American	United States of America	Delete
Asgardian	Asgard	Delete
Sokovian	Sokovia	Delete
Wakandan	Wakanda	Delete

Add Country

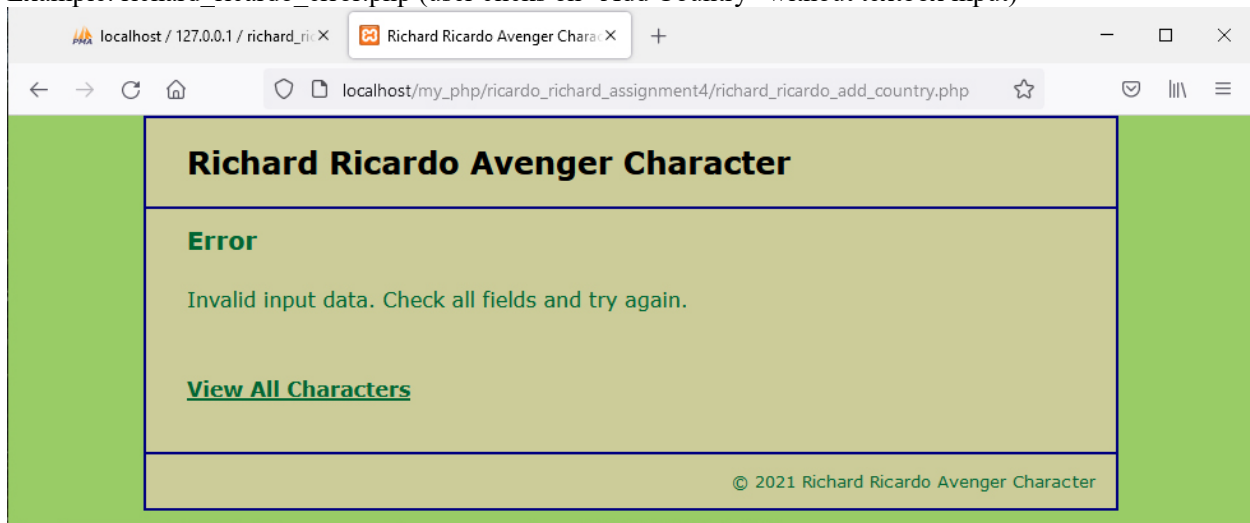
Citizenship:

Country:

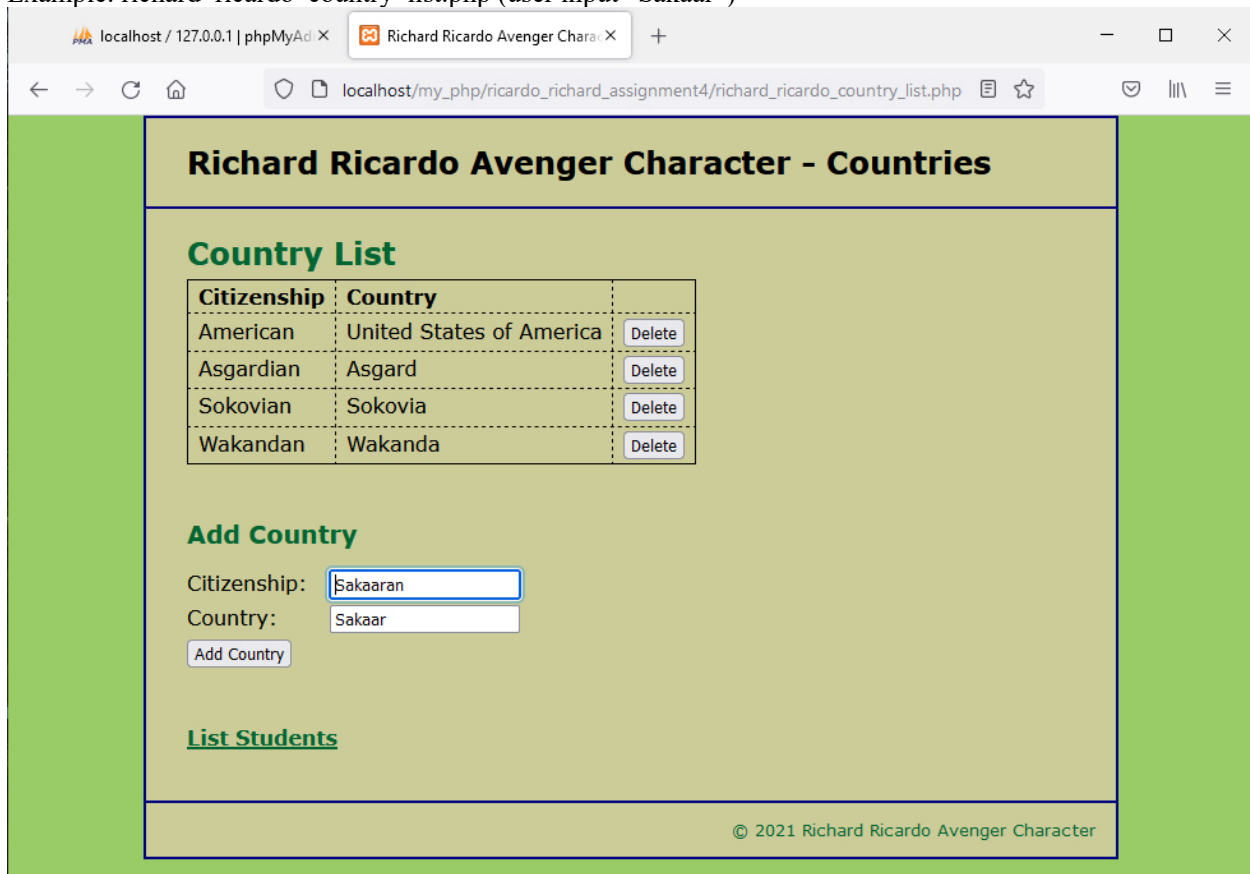
[List Characters](#)

© 2021 Richard Ricardo Avenger Character

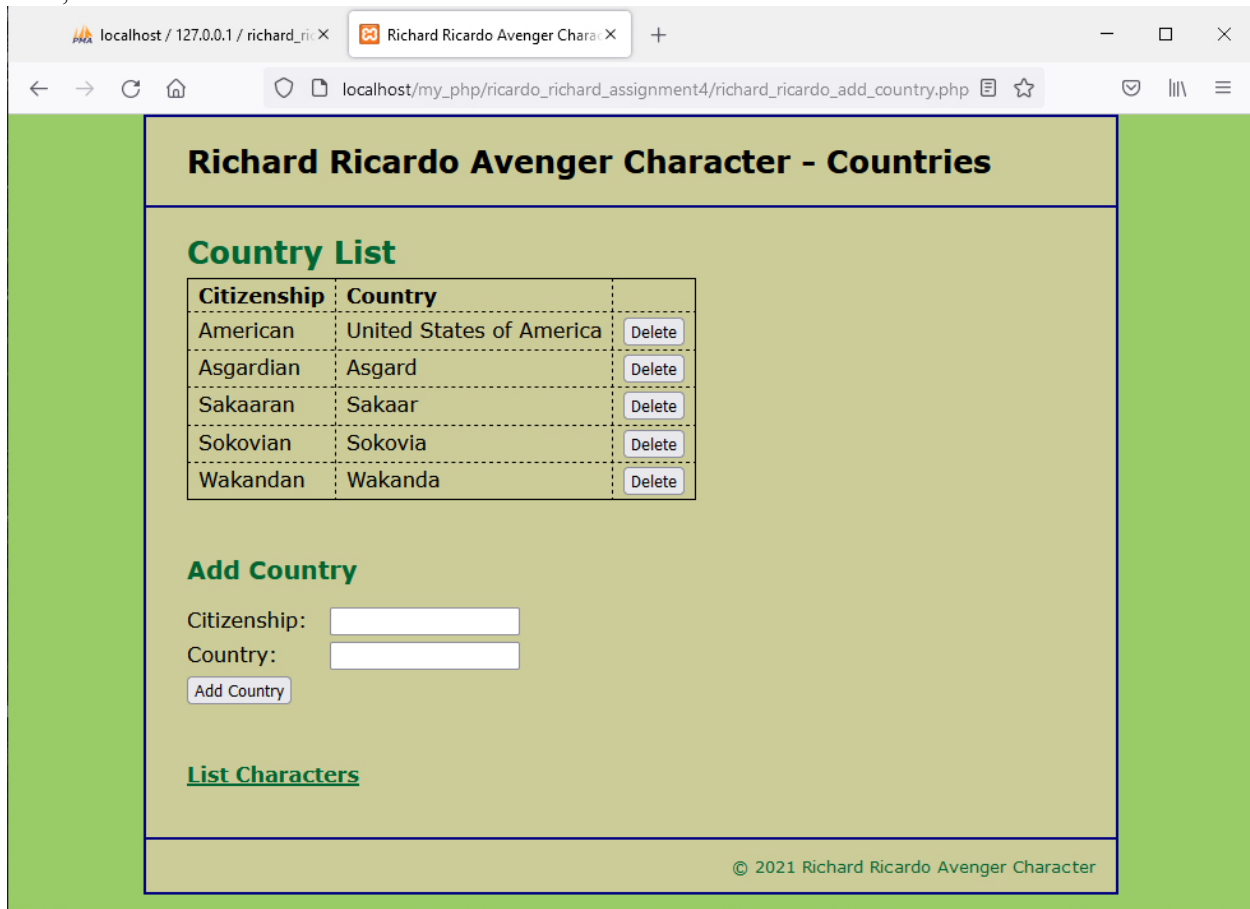
Example: richard_ricardo_error.php (user clicks on “Add Country” without textbox input)



Example: richard_ricardo_country_list.php (user input “Sakaar”)



Example: richard_ricardo_country_list.php (after user input, “Sakaar” was added)
Now, user to click on “Delete” next to “Sakaar”.



localhost / 127.0.0.1 / richard_ri: X Richard Ricardo Avenger Chara: X +

localhost/my_php/ricardo_richard_assignment4/richard_ricardo_add_country.php

Richard Ricardo Avenger Character - Countries

Country List

Citizenship	Country	
American	United States of America	Delete
Asgardian	Asgard	Delete
Sakaaran	Sakaar	Delete
Sokovian	Sokovia	Delete
Wakandan	Wakanda	Delete

Add Country

Citizenship:

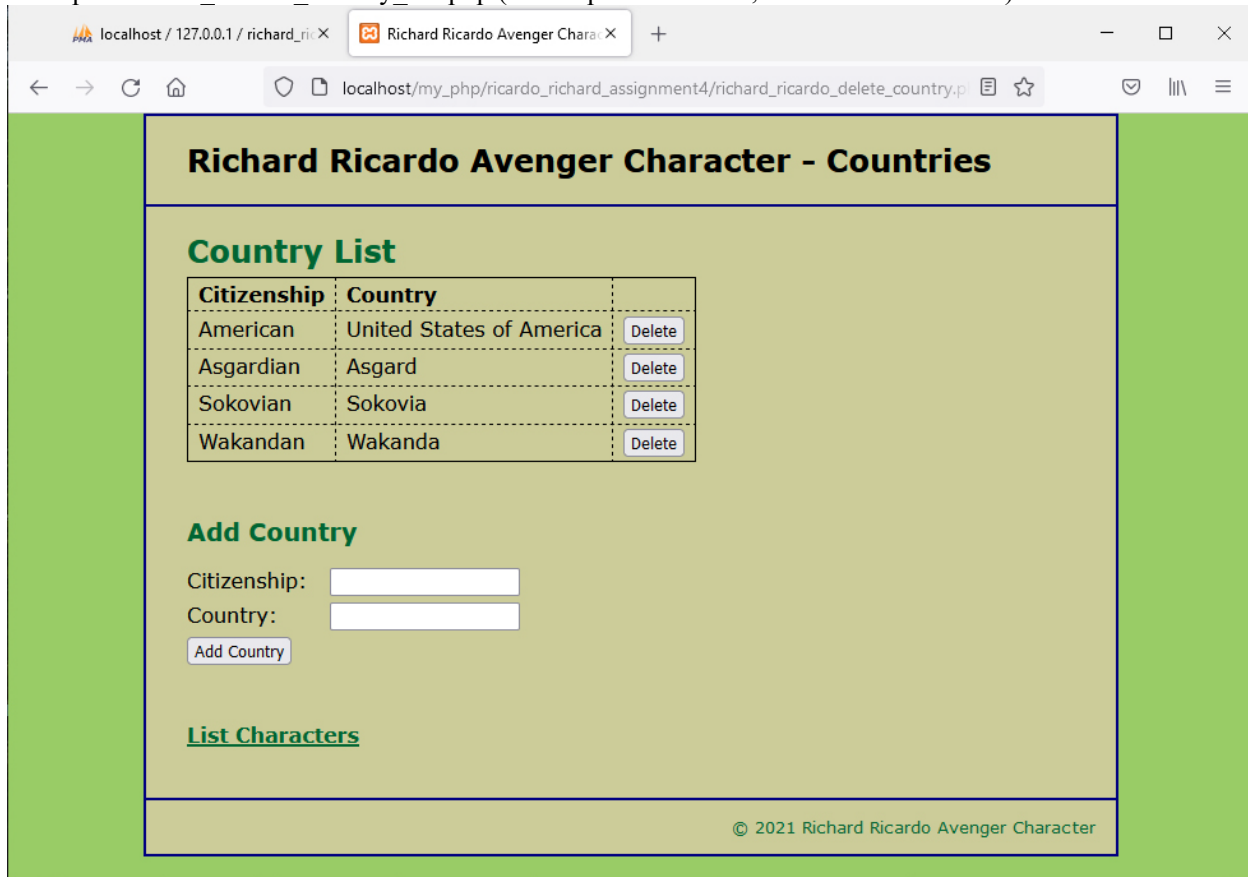
Country:

Add Country

[List Characters](#)

© 2021 Richard Ricardo Avenger Character

Example: richard_ricardo_country_list.php (show updated records, “Sakaar” was deleted)



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. 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>**.
4. Before adding PHP code, all html files must pass html validation at <http://validator.w3.org/> without any **error** (and with only 1 warning).
5. After adding PHP code, the generated html code (Firefox web browser > right-click > view page source) must also pass html validation at <http://validator.w3.org/> without any **error** (and with only 1 warning).
6. All css files must pass css validation at <http://jigsaw.w3.org/css-validator/> without any **error/warning**.
7. If your files do not pass the html and css validations, **2 points will be deducted for each html or css error/warning** found (1 warning allowed for html validator).
8. 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 Photoshop, or GIMP 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_assignment4.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_assignment4**) into a single zip file (or rar file) **lastname_firstname_assignment4.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_assignment4\create_db.sql
- lastname_firstname_assignment4\index.php
- lastname_firstname_assignment4\main.css
- lastname_firstname_assignment4\richard_ricardo_add_character.php
- lastname_firstname_assignment4\richard_ricardo_add_character_form.php
- lastname_firstname_assignment4\richard_ricardo_add_country.php (extra credit)
- lastname_firstname_assignment4\richard_ricardo_country_list.php (extra credit)
- lastname_firstname_assignment4\richard_ricardo_database.php
- lastname_firstname_assignment4\richard_ricardo_database_error.php
- lastname_firstname_assignment4\richard_ricardo_delete_character.php
- lastname_firstname_assignment4\richard_ricardo_delete_country.php (extra credit)
- lastname_firstname_assignment4\richard_ricardo_error.php

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

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**.