
CSCI 5060 Assignment 5

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

The main purpose of this assignment 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**.

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_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 (`<!-- -->`, `/* */`, `//`).

- 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 – 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 (country 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\character_manager**
 - **lastname_firstname_assignment5\q1\errors**
 - **lastname_firstname_assignment5\q1\model**
 - **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\q1**. (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_country_db.php** in folder **lastname_firstname_assignment5\q1\model** (2 points)
 - Hence: **lastname_firstname_assignment5\q1\model\richard_ricardo_country_db.php**
 - Create this php function in the file
 - ```
function get_country_name($citizenship) { return $country_name; }
```
  - This function should take in a citizenship and return the country name, by querying the database
- Create controller file **index.php** in folder **lastname\_firstname\_assignment5\q1\character\_manager\** (6 points)
  - Hence: **lastname\_firstname\_assignment5\q1\character\_manager\index.php**
  - “Require” the following two files:
    - **/model/richard\_ricardo\_database.php**
    - **/model/richard\_ricardo\_country\_db.php**
  - Create controller code to handle if `$action == 'list_characters'` (first time access)

- List \$citizenship as “1”, and \$country\_name as “American”, for testing only
- Files created/updated in this question:
  - lastname\_firstname\_assignment5\q1\character\_manager\index.php
  - lastname\_firstname\_assignment5\q1\errors\richard\_ricardo\_database\_error.php
  - lastname\_firstname\_assignment5\q1\model\richard\_ricardo\_country\_db.php
  - lastname\_firstname\_assignment5\q1\model\richard\_ricardo\_database.php
  - lastname\_firstname\_assignment5\q1\view\richard\_ricardo\_footer.php
  - lastname\_firstname\_assignment5\q1\view\richard\_ricardo\_header.php

Example: “richard\_ricardo\_universe\_db” database and the tables (same as last assignment)

The screenshot shows the phpMyAdmin interface for the 'richard\_ricardo\_universe\_db' database. The left sidebar shows a tree view of databases, with 'richard\_ricardo\_universe\_db' selected. The main panel displays the 'Structure' view for the database, showing a table list with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. Two tables are listed: 'country' (4 rows) and 'hero\_character' (14 rows). Below the table list, there is a 'Create table' section with a 'Name' field and a 'Number of columns' dropdown set to 4.

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

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

The screenshot shows the phpMyAdmin interface for the 'country' table structure. The left sidebar shows the 'country' table selected under 'richard\_ricardo\_universe\_db'. The main panel displays the 'Table structure' view, showing columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action. Two columns are listed: 'citizenship' (varchar(255)) and 'country\_name' (varchar(255)). Below the column list, there is an 'Indexes' section showing a primary index on the 'citizenship' column. The 'Partitions' section is also visible at the bottom.

| # | 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] |

## Example: “hero\_character” table structure (same as last assignment)

The screenshot shows the phpMyAdmin interface for the 'hero\_character' table. The table structure is as follows:

| # | 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, the 'Indexes' section shows the following:

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

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

The screenshot shows the phpMyAdmin interface displaying records in the 'country' table. The query executed is `SELECT * FROM `country``. The results are as follows:

| Options          | citizenship | country_name             |
|------------------|-------------|--------------------------|
| Edit Copy Delete | American    | United States of America |
| Edit Copy Delete | Asgardian   | Asgard                   |
| Edit Copy Delete | Sokovian    | Sokovia                  |
| Edit Copy Delete | Wakandan    | Wakanda                  |

Example: records in “hero\_character” table (same as last assignment)

The screenshot shows the phpMyAdmin interface for the 'richard\_ricardo\_universe\_db' database. The 'hero\_character' table is selected, and the following records are displayed:

| 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  | T'Challa                 | Wakandan    |
| 11      | Quicksilver    | Pietro Maximoff          | Sokovian    |
| 12      | Okoye          | Okoye                    | Wakandan    |
| 13      | Nakia          | Nakia                    | Wakandan    |
| 14      | Shuri          | Shuri                    | Wakandan    |

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

The screenshot shows the 'Edit privileges' page for the user account 'richardinternet@localhost'. The database-specific privileges for the 'richard\_ricardo\_universe\_db' database are as follows:

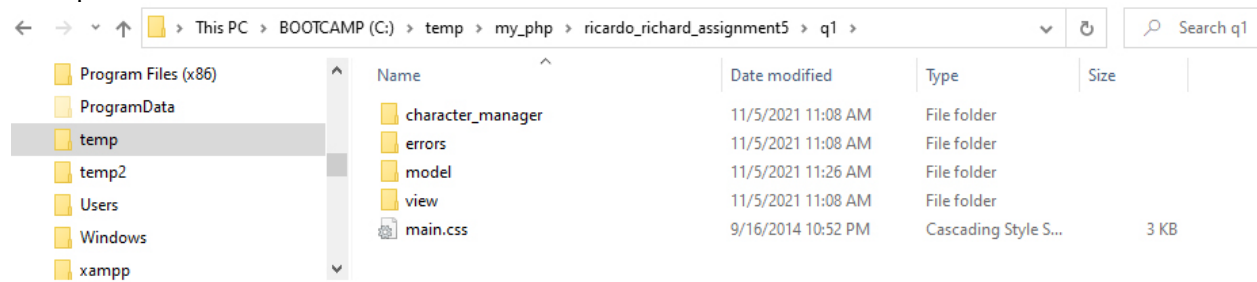
| Database                    | Privileges                     | Grant | Table-specific privileges | Action                  |
|-----------------------------|--------------------------------|-------|---------------------------|-------------------------|
| richard_ricardo_universe_db | SELECT, INSERT, UPDATE, DELETE | No    | No                        | Edit privileges, Revoke |

Example: folder/file structure

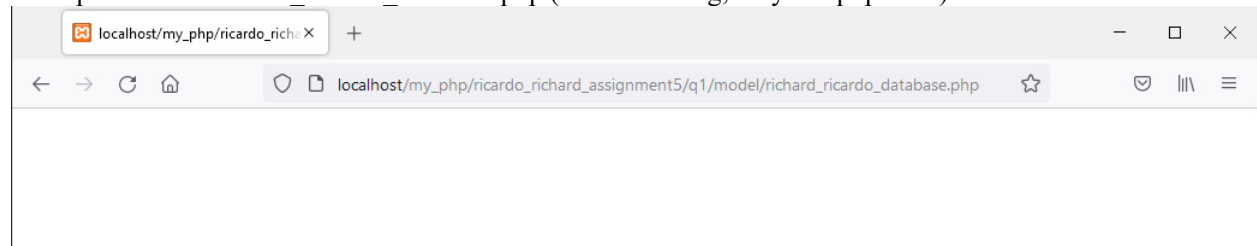
The screenshot shows the Windows File Explorer window for the path 'This PC > BOOTCAMP (C:) > temp > my\_php > ricardo\_richard\_assignment5'. The folder and file structure is as follows:

| Name          | Date modified      | Type        | Size |
|---------------|--------------------|-------------|------|
| q1            | 11/5/2021 11:19 AM | File folder |      |
| q2            | 11/5/2021 11:31 AM | File folder |      |
| q3            | 11/8/2021 11:50 AM | File folder |      |
| q4            | 11/8/2021 5:20 PM  | File folder |      |
| q5            | 11/8/2021 5:35 PM  | File folder |      |
| create_db.sql | 10/22/2021 4:09 PM | SQL File    | 2 KB |

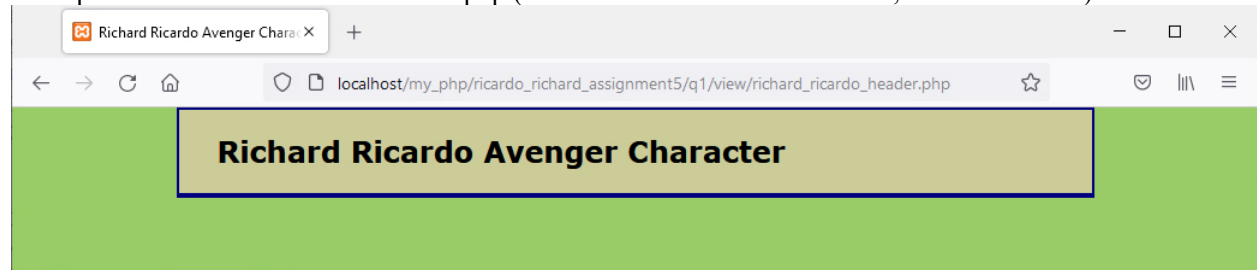
## Example: folder/file structure



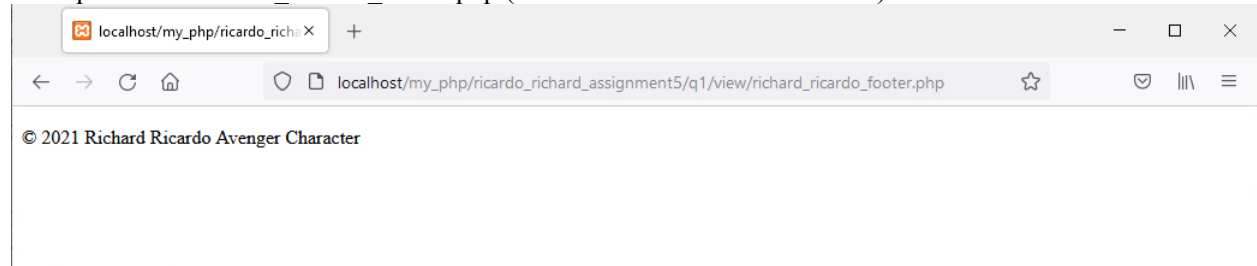
## Example: \model\richard\_ricardo\_database.php (shows nothing, only has php code)



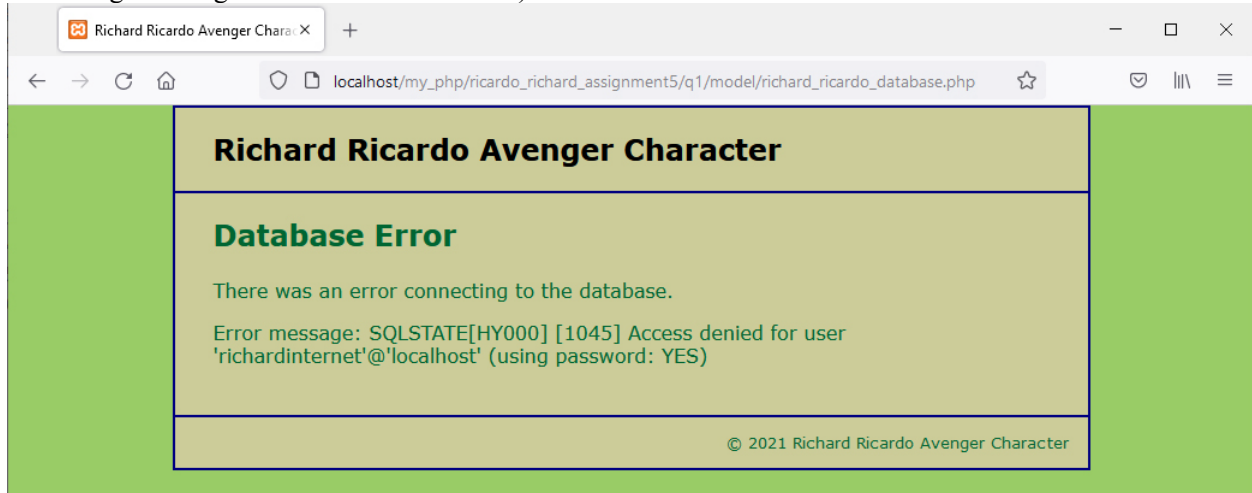
## Example: \view\richard\_ricardo\_header.php (common header for all view files, link to main.css)



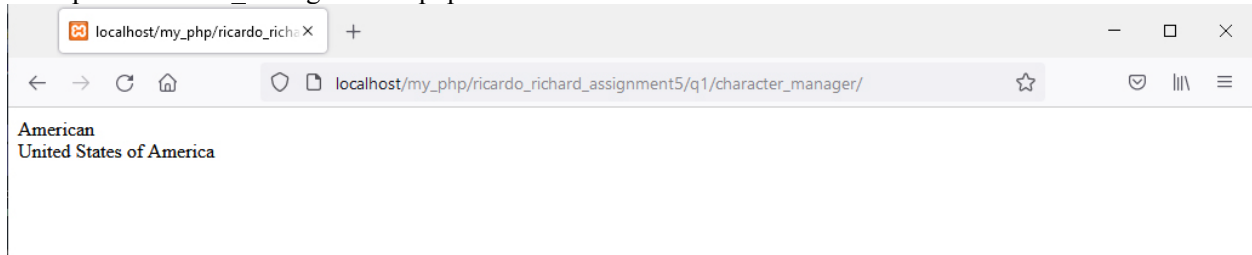
## Example: \view\richard\_ricardo\_footer.php (common footer for all view files)



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



Example: `\character_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 country records and character records from the database
  - Update controller code to list characters
  - Create code to display characters (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\character_manager\`
    - `lastname_firstname_assignment5\q2\errors\`
    - `lastname_firstname_assignment5\q2\model\`
    - `lastname_firstname_assignment5\q2\view\`
- Update file `lastname_firstname_assignment5\q2\model\richard_ricardo_country_db.php` (4 points)
  - Create this php function in the file
    - `function get_countries() { return $countries; }`
  - This function should return all country names sorted by citizenship, by querying the database
- Create file `richard_ricardo_character_db.php` in folder `lastname_firstname_assignment5\q2\model\` (4 points)
  - Hence: `lastname_firstname_assignment5\q2\model\richard_ricardo_character_db.php`
  - Create this php function in the file
    - `function get_characters_by_country($citizenship){return $characters;}`
  - This function should return all character names sorted by hero\_id, in a particular country (take in a particular citizenship), by querying the database

- Update file `lastname_firstname_assignment5\q2\character_manager\index.php` (6 points)
  - “Require” the following additional file:
    - `/model/richard_ricardo_character_db.php`
  - Update controller code to handle if `$action == 'list_characters'` (first time access)
  - Retrieve all characters’ records by using some functions in
    - `model\richard_ricardo_character_db.php`
    - `model\richard_ricardo_country_db.php`
  - “Include” a new file **`richard_ricardo_character_list.php`** to display all characters
- Create file **`richard_ricardo_character_list.php`** in folder `lastname_firstname_assignment5\q2\character_manager\` (6 points)
  - Hence: `lastname_firstname_assignment5\q2\character_manager\richard_ricardo_character_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 characters by a country
- Files created/updated in this question:
  - `lastname_firstname_assignment5\q2\character_manager\index.php`
  - `lastname_firstname_assignment5\q2\character_manager\richard_ricardo_character_list.php`
  - `lastname_firstname_assignment5\q2\model\richard_ricardo_character_db.php`
  - `lastname_firstname_assignment5\q2\model\richard_ricardo_country_db.php`

Example: `\character_manager\index.php` (first country is selected/clicked)

The screenshot shows a web browser window with the following content:

## Richard Ricardo Avenger Character

### Character List

**Countries**    **United States of America**

[United States of America](#)  
[Asgard](#)  
[Sokovia](#)  
[Wakanda](#)

| Character ID | Name           | Real Name                | Citizenship |        |
|--------------|----------------|--------------------------|-------------|--------|
| 1            | Iron Man       | Anthony Edward Stark     | American    | Delete |
| 4            | Hulk           | Robert Bruce Banner      | American    | Delete |
| 5            | Ant-Man        | Scott Edward Harris Lang | American    | Delete |
| 6            | Spider-Man     | Peter Benjamin Parker    | American    | Delete |
| 7            | Star-Lord      | Peter Jason Quill        | American    | Delete |
| 8            | Doctor Strange | Stephen Vincent Strange  | American    | Delete |

[Add Character](#)

© 2021 Richard Ricardo Avenger Character



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

The screenshot shows a web browser window with the title 'Richard Ricardo Avenger Character'. The address bar shows the URL 'localhost/my\_php/ricardo\_richard\_assignment5/q2/character\_manager/?citizen:'. The page content is as follows:

## Richard Ricardo Avenger Character

### Character List

**Countries**    **Asgard**

[United States of America](#)

[Asgard](#)

[Sokovia](#)

[Wakanda](#)

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

[Add Character](#)

© 2021 Richard Ricardo Avenger Character

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

The screenshot shows a web browser window with the title 'Richard Ricardo Avenger Character'. The address bar shows the URL 'localhost/my\_php/ricardo\_richard\_assignment5/q2/character\_manager/?citizen:'. The page content is as follows:

## Richard Ricardo Avenger Character

### Character List

**Countries**    **Sokovia**

[United States of America](#)

[Asgard](#)

[Sokovia](#)

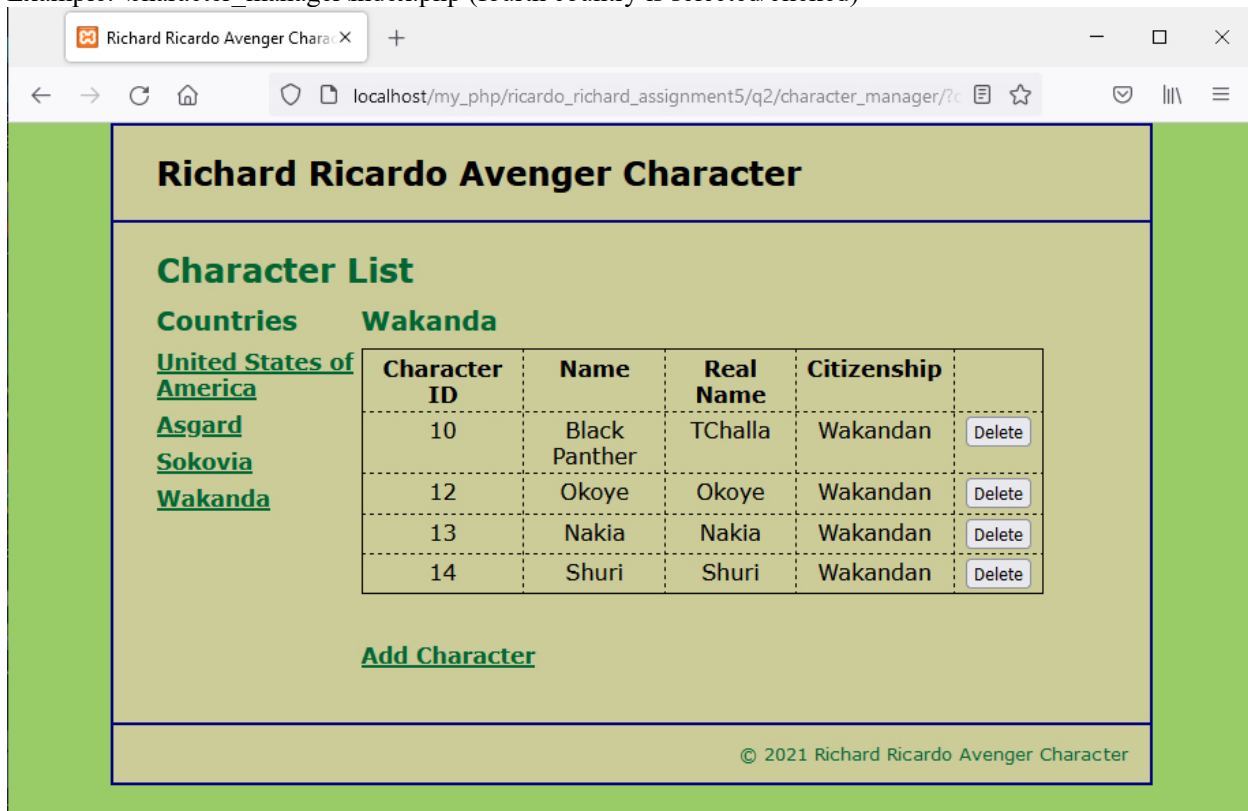
[Wakanda](#)

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

[Add Character](#)

© 2021 Richard Ricardo Avenger Character

Example: \character\_manager\index.php (fourth country 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 character record from the database
  - Update controller code to handle character 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\character\_manager\
    - lastname\_firstname\_assignment5\q3\errors\
    - lastname\_firstname\_assignment5\q3\model\
    - lastname\_firstname\_assignment5\q3\view\
- Update file lastname\_firstname\_assignment5\q3\character\_manager\index.php (10 points)
  - Update controller code to handle if \$action == 'delete\_character'
  - Delete a character's record by using function delete\_character in
    - model\richard\_ricardo\_character\_db.php (to be created next)
- Update file lastname\_firstname\_assignment5\q3\model\richard\_ricardo\_character\_db.php (10 points)
  - Create php function in the file
    - function delete\_character(\$character\_id){}
    - This function should delete a character record by taking in the character\_id, by updating the database
    - This function should be called by index.php
- Files created/updated in this question:
  - lastname\_firstname\_assignment5\q3\character\_manager\index.php
  - lastname\_firstname\_assignment5\q3\model\richard\_ricardo\_character\_db.php

Example: \character\_manager\index.php (before deletion, click on the delete button on Character ID 14)

The screenshot shows a web browser window with the title 'Richard Ricardo Avenger Character'. The address bar shows the URL 'localhost/my\_php/ricardo\_richard\_assignment5/q3/character\_manager/?c'. The main content area has a green header with the title 'Richard Ricardo Avenger Character'. Below the header is a section titled 'Character List' with a sub-section 'Countries' containing links for 'United States of America', 'Asgard', 'Sokovia', and 'Wakanda'. The 'Wakanda' section contains a table with the following data:

| 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 |

Below the table is a link 'Add Character'. At the bottom right, there is a copyright notice: '© 2021 Richard Ricardo Avenger Character'.

Example: \character\_manager\index.php (after deletion, Character ID 14 was deleted.) Now click on the delete button on Character ID 13

The screenshot shows the same web browser window as the previous one, but now Character ID 14 is no longer present in the table. The table contains the following data:

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

The rest of the interface, including the 'Add Character' link and the copyright notice, remains the same.

Example: \character\_manager\index.php (after deletion, Character ID 13 was deleted.)

**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 |

[Add Character](#)

© 2021 Richard Ricardo Avenger Character

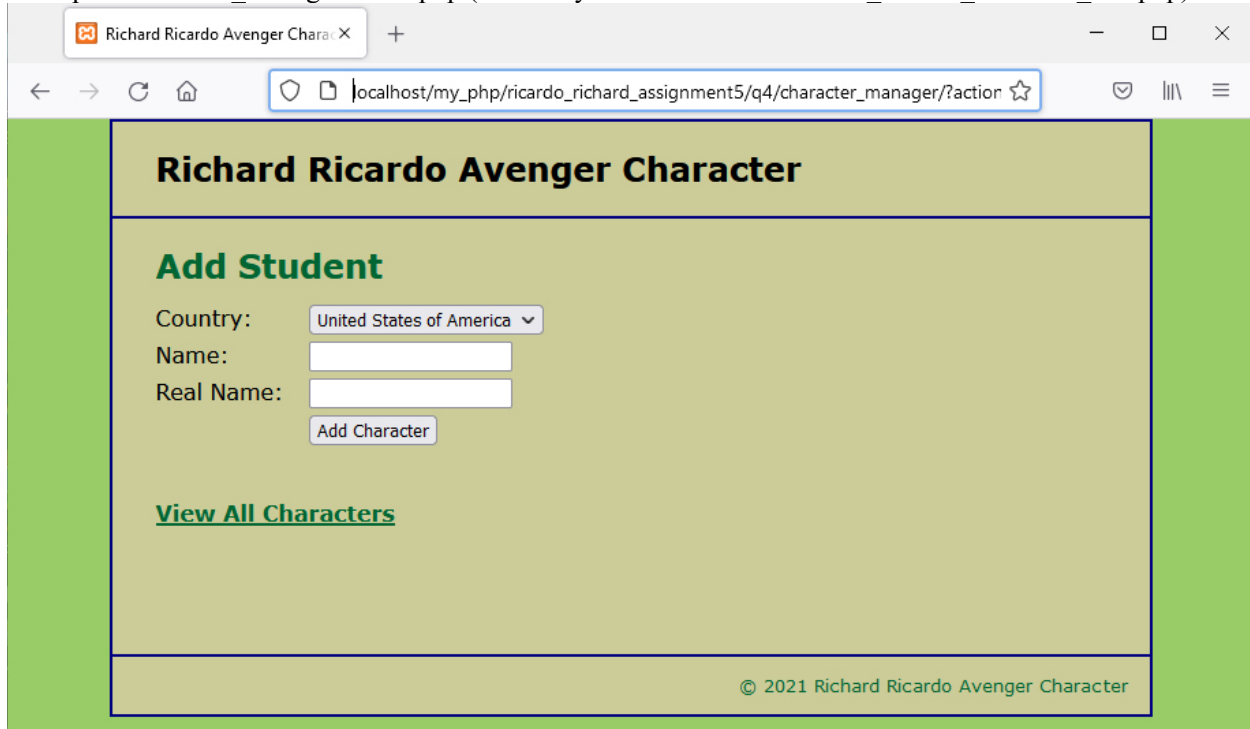
**Question 4 – MVC: PHP Chapter 5, eg010 step 4 (20 points)**

**Estimated time: 2 hours**

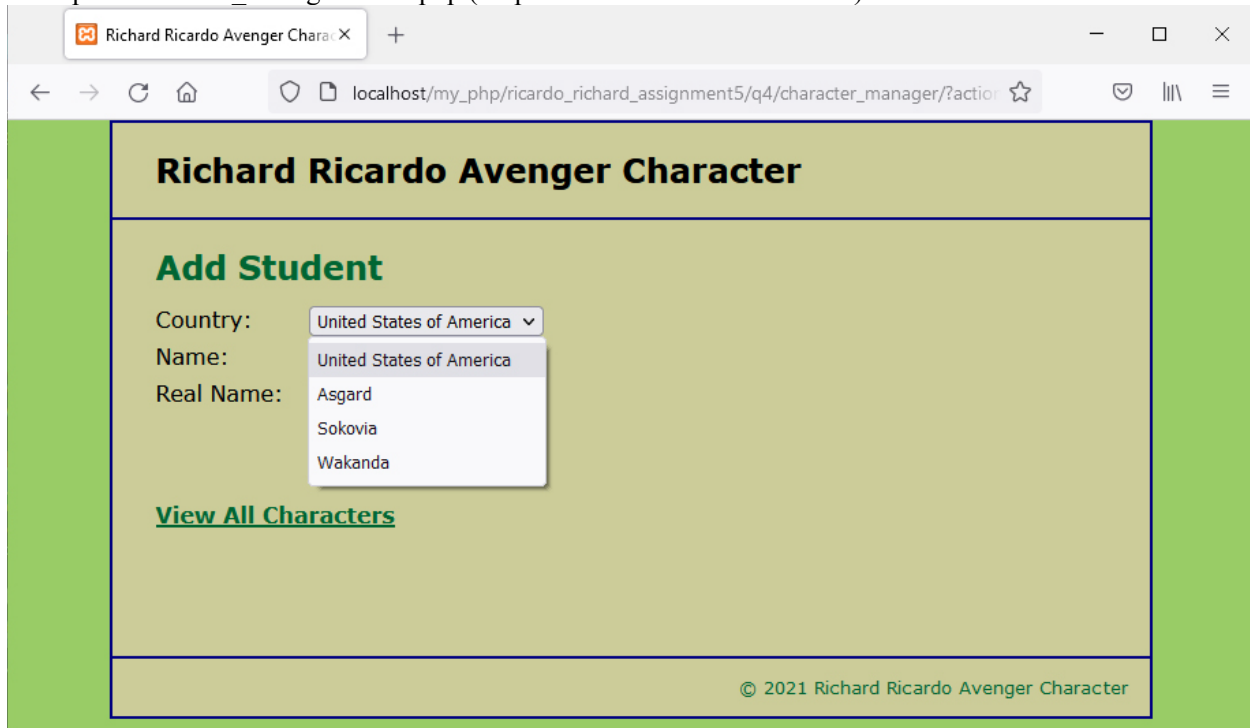
- In this question you need to
  - Update controller code to add a character
  - Create code to display a form to allow adding a character (using common header file and footer file)
- Copy what you have done in question 3 (all folders/files) from the question 3 subfolder to the question 4 subfolder “`lastname_firstname_assignment5\q4\`”.
- Save question 4 files in subfolder “`lastname_firstname_assignment5\q4\`”:
  - Add files to / editing existing files in the subfolders:
    - `lastname_firstname_assignment5\q4\character_manager\`
    - `lastname_firstname_assignment5\q4\errors\`
    - `lastname_firstname_assignment5\q4\model\`
    - `lastname_firstname_assignment5\q4\view\`
- Update file `lastname_firstname_assignment5\q4\character_manager\index.php` (10 points)
  - Update controller code to handle if `$action == 'show_add_form'`
  - Retrieve all countries' records by using a function in
    - `model\richard_ricardo_country_db.php`
    - The records will be used to create dropdown list in the next step
  - “Include” a new file **`richard_ricardo_character_add.php`** to display a form to add character
- Create file **`richard_ricardo_character_add.php`** in folder `lastname_firstname_assignment5\q4\character_manager\` (10 points)
  - Hence: `lastname_firstname_assignment5\q4\character_manager\richard_ricardo_character_add.php`
  - This file is related to the `richard_ricardo_add_character_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 character

- Files created/updated in this question:
  - lastname\_firstname\_assignment5\q4\character\_manager\index.php
  - lastname\_firstname\_assignment5\q4\character\_manager\richard\_ricardo\_character\_add.php

Example: \character\_manager\index.php (it actually shows "include" richard\_ricardo\_character\_add.php)



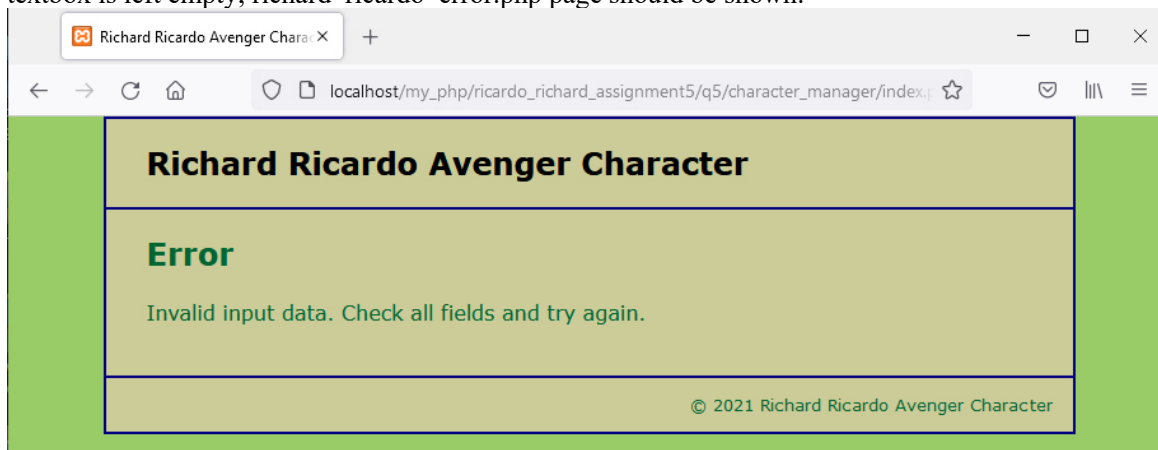
Example: \character\_manager\index.php (dropdown list shows all countries)



**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 character
  - 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 character
- 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\character_manager\`
    - `lastname_firstname_assignment5\q5\errors\`
    - `lastname_firstname_assignment5\q5\model\`
    - `lastname_firstname_assignment5\q5\view\`
- Update file `lastname_firstname_assignment5\q5\character_manager\index.php` (7 points)
  - Update controller code to handle `if $action == 'add_character'`
  - Retrieve all forms filed passed from `richard_ricardo_character_add.php` (last question)
    - If any form field is empty, create error message and “include” a new file `\errors\richard_ricardo_error.php` to display error message
    - If all form fields are filled, call function `add_character` in
      - `model\richard_ricardo_character_db.php` (to be created next)
- Create file `richard_ricardo_error.php` in folder `lastname_firstname_assignment5\q5\errors\` (6 points)
  - Hence: `lastname_firstname_assignment5\q5\errors\richard_ricardo_error.php`
  - This file is related to the `richard_ricardo_error.php` from the last assignment
  - It should “include” `richard_ricardo_header.php` file
  - It should “include” `richard_ricardo_footer.php` file
  - It should display an error message from `index.php`
- Update file `lastname_firstname_assignment5\q5\model\richard_ricardo_character_db.php` (7 points)
  - Create this php function in the file
    - `function add_character($name, $real_name, $citizenship){}`
    - This function should add a character record by taking in the above arguments, and by updating the database
  - This function should be called by `index.php`

Example: `\character_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: \character\_manager\index.php (it actually shows/"include" richard\_ricardo\_character\_add.php): user input

**Richard Ricardo Avenger Character**

### Add Student

Country:

Name:

Real Name:

[View All Characters](#)

© 2021 Richard Ricardo Avenger Character

Example: \character\_manager\index.php: after successfully adding a character 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    | <input type="button" value="Delete"/> |
| 12           | Okoye           | Okoye     | Wakandan    | <input type="button" value="Delete"/> |
| 15           | Erik Killmonger | N'Jadaka  | Wakandan    | <input type="button" value="Delete"/> |

[Add Character](#)

© 2021 Richard Ricardo Avenger Character

**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\_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\character\_manager\index.php
- lastname\_firstname\_assignment5\q5\character\_manager\richard\_ricardo\_character\_add.php
- lastname\_firstname\_assignment5\q5\character\_manager\richard\_ricardo\_character\_list.php
- 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\_character\_db.php
- lastname\_firstname\_assignment5\q5\model\richard\_ricardo\_country\_db.php
- lastname\_firstname\_assignment5\q5\model\richard\_ricardo\_database.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**.