# American University, CSC 435 Web Programming

# Homework Assignment 5: Trivia Game (team work allowed)

**Total points: 100pts + 10pts.**

**Instructions:**

**Due date: Monday, April 22, 2019, end of the day. Late Policy strictly applies.**

**Please do not look it up for solutions on the Internet. If it were found, you would be reported as cheating to Dean of Students. Try as hard as you can do solve this.**

**Tools using: JavaScript, JSON, Ajax, PHP**

**For your .js and .php file, you must comment on what each function (or blocks lines of code) is doing. Code without proper comment will have reduced points.**

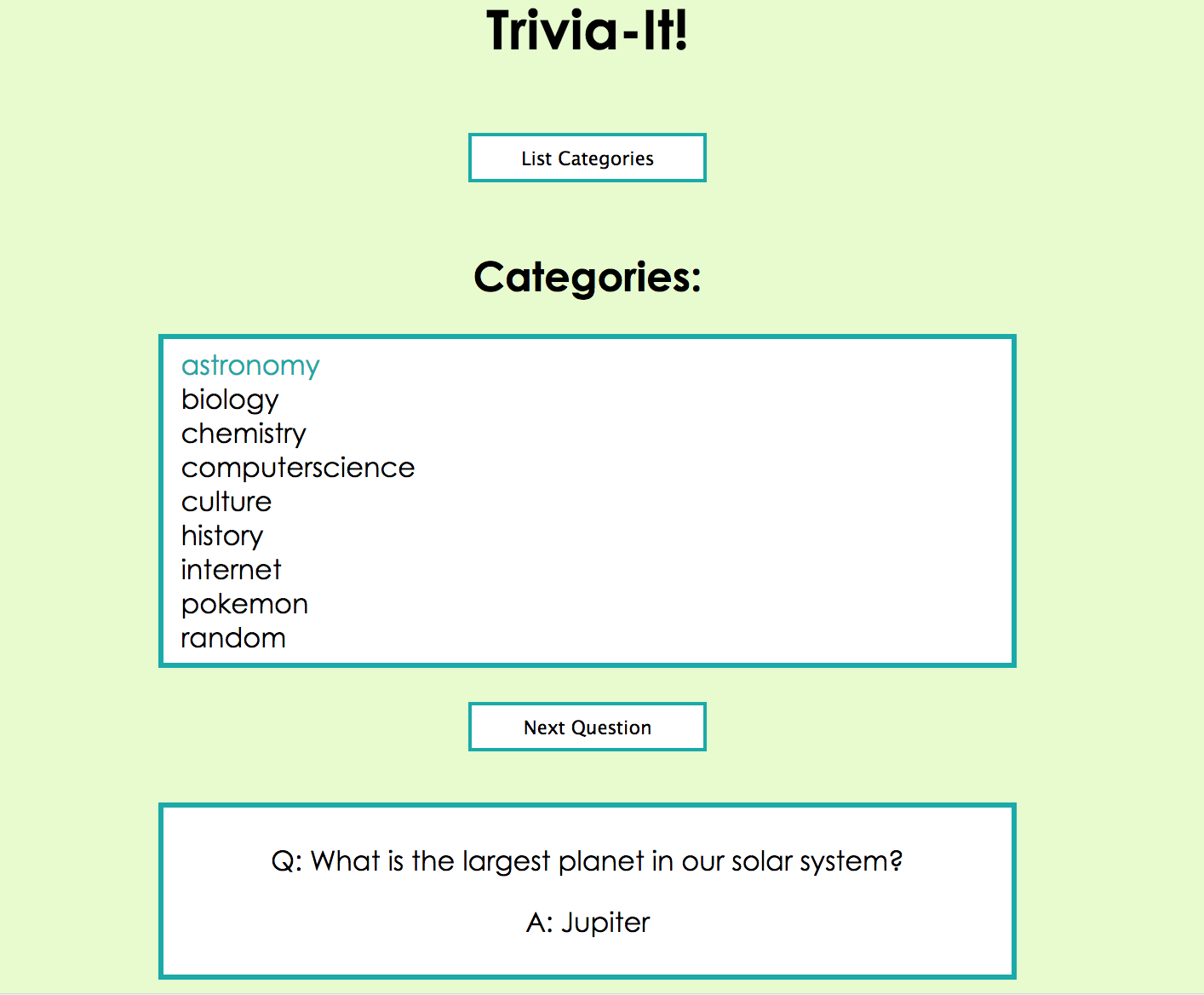
**Team work: Allowed to up to 2 people. You must specify in your webpage .html, who are the authors.**

Synopsis:

The goal of the assignment is make a trivia game with JavaScript, JSON, Ajax and PHP.

The page looks like the following (we will provide you with HTML and CSS, you will write the .php and .js):

The user click “list categories” and in the categories div the categories are listed. They click “Next Question”, the next question showed up. If they then click “Show Answer”, the answer will be revealed.



Step 1: Download the triva.html, trivia.css and triva.zip from blackboard. Unzip the triva.zip. This folder contains several folders of trivia questions as in .txt files. Make sure your main code is outside these folders.

Step 2: Write a trivia.php code to read the .txt files (from particular category specified by the fetch()) and output as JSON. You can use $\_GET[“mode”], scandir and json\_encode().

What you need to do is given the category query parameter from the trivia.js, create a **random question** (e.g. using array\_rand()) send it back to the browser.

There are many ways to open the .txt files, for example:

$triviafiles = "../trivia/";

$categoryName = strtolower($\_GET["name"]);

$trivia = glob($triviafiles . $categoryName . "/\*.txt");

To print the question and answer, you can:

print(json\_encode(array("question" => $question, "answer" => $answer)));

Step 3: Write a trivia.js code to call the PHP code and render the content back to the browser. You might find the following code snippets useful

To fetch the a question from a particular category:

function fetchCategories() {

let hxr = new XMLHttpRequest()

hrx.onload = displayCategories;

hrx.open("GET", "trivia.php?mode=categories");

hrx.send();

}

function showTrivia() {

let url = "trivia.php?mode=category";

if (currentCategory) {

url += "&name=" + currentCategory;

}

fetch(url)

.then(checkStatus)

.then(JSON.parse)

.then(displayQuestion);

}

Bonus (10pts) :

1. add new questions into proper categories of your own choices by writing a writequestion. php
2. Ask the user to create a question and answer and write it to the directory.
3. Create new categories of questions and add question and answer.
4. Explore Trivia API (<https://opentdb.com/api_config.php>) to use the database for questions.

Grading:

To receive a passing grade (60%) for the project, you can skip the whole “category” option. The user can just click “next question” and one of the questions will show up. To achieve this, you can just read from one of the .txt file or you can even create your own question (just hardcode a JSON) and request that JSON back to browser. However, if you want to get an impressive grade, you should actually follow the steps above to make the game work to its full functions.

For gain the rest of the 40%, you must implement the .php code that can parse the .txt file and choose a random question with specific category to render back to the browser.