

## Assignment

Write a web game in PHP, where the user has to correctly guess a number between 1 and  $n$ .

## Description

The page initially displayed to the user has one editable text box and two buttons: Guess and Restart.

When the page is initially loaded, and whenever the user hits the Restart button, the program "thinks" a secret number between 1 and  $n$ .

The user can try to guess the number by typing her guess into the text box and then click on the Guess button.

This shall cause the browser to reload the page. In addition to the text box and the buttons, the page will also show all the previous attempts (including the last one), with an indication of whether each number was too high or too low.

The user can then type in another number and click the Guess button again.

If the user correctly guesses the number, the program shall also display a congratulation message and disable the Guess button, leaving the user with the options to restart a new game or just navigate away.

At any point, the user can start a new game by clicking the Restart button. This will cause the page to reload, and the program to generate a new secret number, and to reset the attempt history.

## Constraints

The secret number should be randomly generated each time the user starts a new game (**not** each time the user reloads the page).

## Hints

PHP offers a built-in `rand()` function that can help you with the casual number generation.

You'll need to persist the game history and the secret numbers in between page requests. You could use GET parameters, but this may not be the optimal choice because it will expose the secret number to the user. Consider using cookies instead, or maybe POST requests.

## Evaluation criteria

I will evaluate the submissions according to the following criteria:

- The program accomplishes its functional requirements
- The code is readable and clear; functions and variables are properly named
- Functions are potentially reusable
- The program's core functionality (i.e. the game logic) is clearly separated from the HTML-generating part
- User input and HTTP request parameters are validated, and errors are handled gracefully

I will **not** consider the following criteria in the evaluation – so you don't have to spend time on that for my sake (but you are of course welcome to):

- Fancy-looking web page