First Practical Exam

Web Programming – 22. 11. 2019.

In this exam, your task will be to make a web application that uses NASA's API for Astronomy Picture of the Day (https://api.nasa.gov/) (the looks of this web application is shown in Image 1). Everything should be implemented inside two files: *index.php* and *Picture.php*. In these files you will find TODO comments that suggest where each task should be implemented. Besides that, folder *templates* contains all HTML segments that are given inside of this PDF file, so please, do not copy HTML segments from this PDF, but use those provided in the mentioned folder (copying from PDF sometimes causes unwanted behavior in PHP scripts). The number of points for each TODO are given in parenthesis.

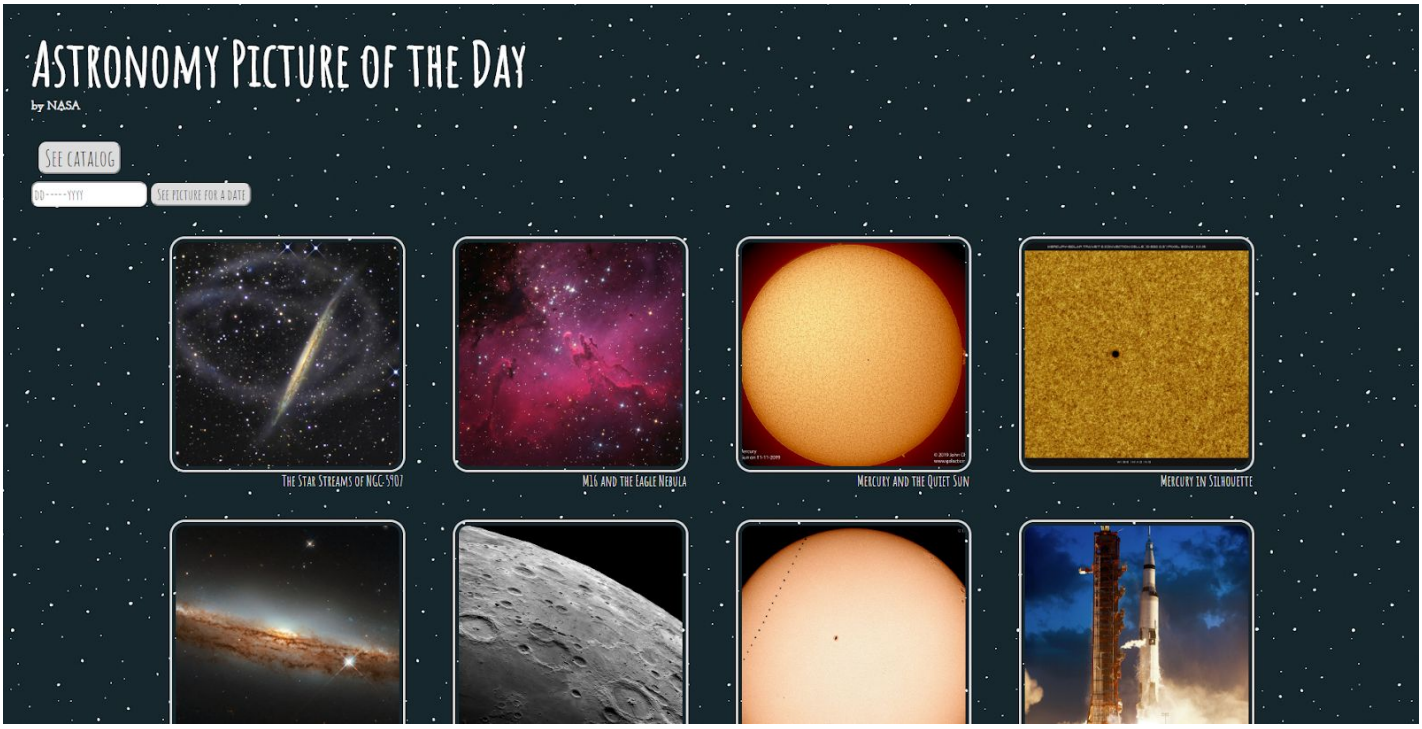

Image 1.

Description of the relevant files:

- *index.php* - Contains HTML code of the main page.
- *Picture.php* - Contains class *Picture*. This class will hold data about one astronomy picture of the day.
- *data.json* - Contains data about astronomy pictures of the day. This data will be loaded and shown by using PHP.

NOTE: You are allowed to add additional methods, fields, etc. You are also allowed to extend the existing methods and functions.

TODO 1 - First of all, you need to load and show data about astronomy pictures of the day. This TODO task should be implemented by following next few steps:

> **TODO 1.1 (1,5 points)** - In the class *Picture* (file *Picture.php*) you need to add constructor that initializes all the class's fields. The constructor should have one parameter. This parameter will be associative array, whose keys

are strings equal to fields' names (e.g. filed *title* corresponds to key "title"). Additionally, it can happen that key "copyright" is not present in the associative array, so you should take care of that, too.

TODO 1.2 (2 points) - Add a method *getHtml* in the class *Picture*. This method should have one boolean parameter whose name is *hd*, and should return a string with HTML code of the picture. HTML code should look like this:

```
<div class="apod">
     <h3>%TITLE%</h3>
     <h4>%DATUM%</h4>
     <div>
          <img src="%URL%">
          <span class="copyright">%COPYRIGHT%</span>
     </div>
     <p>%EXPLANATION%</p>
</div>
```

Red tokens should be replaced in a following way:

%TITLE% - This token should be replaced with a value of the field *title*.
%DATE% - This token should be replaced with a value of the field *date*.
%URL% - If the value of the parameter *hd* is true, this token should be replaced with a value of the field *hdurl*, otherwise it should be replaced with a value of the field *url*.
%COPYRIGHT% - This token should be replaced with a value of the field *copyright*. Additionally, if the field *copyright* is empty, you should completely omit HTML segment marked with yellow.
%EXPLANATION% - This token should be replaced with a value of the field *explanation*.

TODO 1.3 (2 points) - You should finish implementation of the function *loadPictures* from the file *index.php*. This function should return an array of objects of the class *Picture*, so by using data read from the file *data.json*, you should instantiate object of that class, and put them in the resulting array.

TODO 2 (4 points) - Today's date is placed in the variable *date* (in the *index.php* file), while the pictures obtained by calling the function *loadPictures* (from the TODO 1.3) are placed in the variable *pictures*. Each picture has its date in the field *date*. You should output the HTML of the today's picture (HTML can be obtained by calling the method *getHtml* of the class *Picture*). Outputted picture should not be HD. If the today's picture does not exist, you should output an error message somewhere in the page, by using the following HTML segment:

```
<div class="error">%MESSAGE%</div>
```

token **%MESSAGE%** should be replaced with a concrete error message for a user.

TODO 3 (3 points) - You should extend the web application by allowing user to see the picture of the chosen date (not only today's picture). In order to do that, you should first make an HTML GET form. By using this form, user will choose a date, and after submitting it, the picture for the selected date should be shown. Similarly to the previous TODO, if the picture for the chosen date does not exist, an error message should be shown.

TODO 4 - Allow user to see a list of thumbnails of all available pictures.

TODO 4.1 (3 points) - Make a button with text "Show all". By clicking on that button, instead of one concrete picture, user should see thumbnails of all available pictures. HTML code for picture's thumbnail can be obtained by calling method *getBriefHtml* of class *Picture*.

TODO 4.2 (1,5 points) - Make all the thumbnails clickable. By clicking on one certain thumbnail user is back to “one-picture” view (he/she does not see all thumbnails anymore). The picture that should be shown to user is exactly the picture of the thumbnail he/she clicked on (functionality of showing a picture of a custom date is already implemented in TODO 3, so you should use that solution to implement this TODO).

TODO 4.3 (1 point) - Extend the implementation by showing “thumbnails” view (showing all pictures’ thumbnails) whenever “one-picture” view fails with showing a single picture (e.g. when a picture for the selected date does not exist - see TODO 2 and TODO 3).

TODO 5 (2 points) - Only in “one-picture” mode (not in “thumbnails” mode) you should output a POST form that will allow user to enter a secret password for showing HD version of the picture. Secret password for showing HD picture is “nasa”, so if user enters password correctly, he/she should see HD version of the picture (see TODO 1.2 as a reminder of how the HD version of the picture is outputted).