BRAINSTER

Workshop 17:
PHP - IO

Brainster Web Development Academy

Exercise - part 1

For this workshop, an application for creating “Top 5 songs playlist” needs to be created. There is no need to implement authentication mechanism. Every user that visits the page should see all the playlists so far created and should be able to create a new one.

If there are no playlists created yet, a message indicated that should be shown.
When creating a playlist, the user enters a name which should be validated for uniqueness (two playlists with same name should *not* exist). Below the name, five inputs for entering Youtube urls are shown. These fields are required and should be validated as valid links (filter_var) and valid Youtube urls (custom function).

If any validation fails, user is redirected back to the create playlist page, the old input values are pre-populated and the error message is shown at the top. The old input values and the error message are stored and read from session.

If the form is successfully submitted, a new file should be created on disc with the name of the playlist, replacing empty spaces with dashes (ex. top-5-rock-songs.txt) and the 5 youtube links should be saved inside in separated rows. Then, the name of this new file should be appended to a file named playlists.txt which will contain all the playlist names and from where the playlist catalogue is created. Upon successful operation, the user is redirected to the playlist catalogue page with a message that is transferred using session, not via query string.

Exercise

The index page should present a playlist catalogue where all the playlists are shown in a grid format. The playlists are read from the playlists.txt file, where each new created playlist is appended.

If a user clicks on any of the playlists, he is taken to a new page where all the links from that playlists are shown as video previews (using the iframe or embed html tags). The new page is dynamic and it decides which playlist items to show depending on a GET parameter value transferred from the playlist catalogue page.

For example, if the user clicked on the “Top 5 rock songs” playlist in the catalogue page, he should be taken to a page like playlist.php?file=top-5-rock-songs. Then, in the playlist.php file, if the file from the GET parameter is present, its contents are read and the songs are shown. If the file does not exists (for example, someone manually changes it like playlist.php?file=file-that-does-not-exist), the user should be redirected back to the playlist catalogue page.

A working copy of the application can be found on the following link:
https://workshops.brainster.tech/php_io_playlist/