FOOLOGRAPHY

READ CAREFULLY BEFORE BEGINNING!

Please read the entire assignment, then begin to work on it step by step. The assignment is divided into multiple tasks, where each task usually requires the previous on to be finished. Before beginning with implementation think about the entire solution and implementation details and it is recommended to make some time estimations (how much time do you need for specific task). You are allowed to use existing components and third party libraries.

IMPORTANT: Create an empty Git repository before starting (git init). During the development create several commits, **at least one per a single task**.

When you finish please just send us a link to your GitHub/GitLab repository where we can find the task (Make sure we have access) or send us a .zip archive containing **project files and repository**. Please also use a creative name for your repo so other applicants in the future won't easily find your solution.

If you into any problems you can always ask for help. You are also allowed to use any internet sources and libraries, unless they provide a complete solution. Do your best developing the solution, but do not lose too much time with visual design and details.

ASSIGNMENT (PHP – MySQL)

You will create a simple **Link Shortener Website** *(such as https://bitly.com/)* where a user can transform a long URL into a short one which will redirect to the given URL. Ex. If you input *http://www.foolography.com/shop/product/unleashed/* you can get a short URL like *http://localhost/s/e19d5cd5*. Also, a small user login system should be implemented which enables users to edit or delete their short URLs.

First Part – Creating the business logic
We assume the website is open for anyone to use. That means also anonymous users can submit a URL. The index page shows the most recent 10 links and a submission form.

1. Create a MySQL database and add table which saves URLs and their related code for the short URL.

Note: Include an SQL file with the CREATE TABLE query and test data in your codebase!

2. Create a simple process:
    a. Provide an HTML input form to enter a URL.
    b. Save the user input URL and generate an associated code that will be inserted into the database.
    c. Whenever the short URL is visited, the user will be redirected to the destinated URL.

Note: Develop your own custom algorithm (disallow duplicates). Copy and paste code not permitted.

3. Extend your website and add statistics:
    a. In the short URL database table, add a row to count the number of times a user visited the short URL.
    b. Implement the process in your script and increment the counter, each time a user visits a URL.

4. Add an overview table for the URL database entries:
    a. All URLs should be displayed as a table on a page, showing the **original URL**, **short URL code** and **visit counter**.
    b. Also, add pagination with 10 entries per page. Include at least 3 pages of test data.

FOOLOGRAPHY

Second Part – Adding a simple user authentication system
Extend the website with a user registration and login page. The previous index page can be reused. When the user is logged in and visits the profile page, it will show all their links in the table with edit/delete buttons.

5. Create another table in the database to enable user registration:
    a. Provide at least a username and a password.
    b. Add a registration and login page.
    c. The user should be authenticated within a PHP session.

Note: Security measurements will be evaluated.

6. Link user accounts to their short URL:
    a. Add a new table to the database to associated user IDs with URL IDs.
    b. Create pages for logged in users should be able to edit or delete their URLs.

Note: Implementation of many-to-many relational data is required.

7. Script Task: Delete all URLs older than 30 days that **aren't linked** to a user account:
    a. Add timestamps to the URL table.
    b. Create a script which will delete all URLs when the rules apply (30 days old AND no user associated).

8. Question: How can the deletion script be automated **on the server** to **run once per day**? Include your answer inside a markdown file.

9. Server Task: Add a .htaccess file to rewrite the short URLs to this result: *http://localhost/s/e19d5cd5*.

Environment Constraints

- PHP 7.2+ with apache server
- MySQL 5.3+

Evaluation Rules

- All described functions must work correctly to pass the test. Design will not be evaluated.
- Code quality is a must, not a feature: formatting, comments, indentation, separation will be evaluated.
- Plain PHP without the help of frameworks is required. (**No use** of CodeIgniter, Symfony, Laravel, etc.)
- Your code must run in any environment, that means not only on your machine. Make sure to test the code with debug errors **enabled** and **disabled**.
- Object-oriented approach is preferred.
- PDO for database handling is preferred.

Links

- https://www.mamp.info/de/mamp/mac/
- https://www.wampserver.com/en/
- https://www.apachefriends.org/index.html
- https://httpd.apache.org/docs/2.4/rewrite/remapping.html
- https://www.nginx.com/blog/creating-nginx-rewrite-rules/
- https://www.php.net/manual/en/book.session.php
- https://www.php.net/manual/en/book.mysqli.php

foolography GmbH
Greifswalder Str. 9
10405 Berlin, Germany
USt-IdNr.: DE263567148
CEO: Oliver Perialis
HRB 164806
Amtsgericht Charlottenburg

www.foolography.com
info@foolography.com
+49 30 12089681
Commerzbank Darmstadt
KTO: 134602200 BLZ: 5084005
IBAN: DE31508400050134602200
SWIFT/BIC: COBADEFFXXX