Web Programming II
2016 Spring
Assignment I
Deadline: 15 March 2016 23:59

Ahmet Soylu

February 15, 2016

1 Objectives

Students should demonstrate that they know and able to use:

- basic PHP constructs such as expressions and loops,
- functions, arrays, and objects,
- file management and time manipulation,

and create a basic dynamic website using these knowledge and skills.

2 Description

In this assignment, you are expected to create a part of an account management system for a bank, which manages data about *customers* and their *accounts*.

- For each customer, we need to know *name*, *surname*, *birthdate*, *address*, and total assets (i.e., the total sum of money).

- For each account, we need to know *account holder, account number, currency type, balance, withdrawals,* and *deposits.*
- For each transaction (i.e., withdrawal and deposits), we need to know type, value, associated account, and date.
- Note that every person might have more than one account.

First *design a set of text files,* e.g., in coma-separated values (CSV) format, to store and read your data. You should avoid unnecessarily repeating data (i.e., your text database should be normalised). The order and structure of descriptions given in this text does not necessarily reflect how you should organise your text files and code.

Then write *three php files.*

The first php file is named 'customers.php', and when called it should first display the total number of customers and accounts first, then a table of customers. The table should include at least the basic information, the number of accounts, associated account information, and total amount of assets (i.e., the total amount of money in different accounts) for each customer.

The second php file is named 'account.php', and when called it should first display the number of deposits, the number of withdrawals, and the balance for a specific customer, then a table of transactions. Each transaction should include at least the date, type, amount, and currency. One can define and change the customer by manipulating a parameter in the code.

The third php file is named 'data.php' and should allow you to upload new customers, transactions, or accounts (i.e., from a file). You should check the validity of data (e.g., a transaction should indeed belong to an account that exists). One should be able to sort the records by date or amount in ascending or descending order by manipulating a parameter in your code.

3 Other requirements

- Data which needs to be calculated (total assets, the number of transactions etc.) should be calculated and then needs to be written back to data files every time one calls the 'customer.php' or 'account.php'.
- All date fields should be stored as unix timestamps.
- The use of classes, objects, and arrays are mandatory.
- You should read from your files to objects, whenever possible.
- Look and feel of the pages is under your discretion.
- Your code should work without assuming any fixed number of records.

- Your code should be well indented and commented.
- Your code should use proper variable names for readability.
- Prepare some example data sets for testing.
- Your class declarations should be in a separate file.
- Create a sensible folder structure (e.g., data, classes etc.).
- Provide a ’readme.txt’ file explaining how to use your application (e.g., sorting).
- Zip all your code into a single zip file.
- This assignment has to be done individually!

4 HINTS

- In order to update a file, consider using *file_get_contents, file_put_contents,* and *str_replace functions.*
- You should consider to implement a small library of functions to read and write files (should just meet your needs and not be an exhaustive library of course!).

Good luck!