**Lab 11-12 (Project task). JavaScript Web-Bankist Application.**

A simple banking application created using JavaScript for learning and practice purposes (arrays, numbers, dates, timers) in JS.

The project is a simulation of a banking application. The project should have the following actions:

* Can login be using username and pin;
* Automatically logout after certain time;
* Can transfer balance from one user to another;
* Can request for loan;
* Can close account;
* Can sort balance movements;

**A general step-by-step guide to the project is given below:**

1. Make a login form, which should have two inputs (username and pin-code)
2. Then create users. The username will consist of the user's initials.
3. Enable the customer movements to be displayed on the current balance sheet (display balance). This is done by constructing the **displayMovements()** function which uses the forEach() method to loop over each of the movements inside the array. There should also be a button **“SORT”** that sorts these records.
4. Constructed the **calcDisplaySummary()** function which calculates and displays the IN, OUT, and INTEREST values. They all three are each calculated the same, excepting interest, which has added steps to meet the extra parameters which are an interest rate of 1.1% on all deposits if the interest is => 1. The array of movements is filtered to create a new array containing only deposits. That new array is mapped over with each value being multipilied by the interest rate. Those new calculated values are then added to the new array which the .map() method creates. Another filter() method is run to remove any interest rate calculations that are less than 1. Finally reduce() is called to add the sum of each deposits interest payout into a total, which is inserted into the HTML via a template literal to to fixed decimal places.
5. Display the user's current balance
6. Create a function that makes money transfers. To make a transfer, you need an amount and a username. Checks are done to ensure that recipient and amounts to be transferred are valid. Then the recipient account needs to have the transfer added and the sender account needs to be detected from. Finally the UI has to be updated to account for the changes.
7. The project should be able to close the account. To close the account, you need to enter your username and password and a confirmation button.
8. To add in the loan button functionality an event listener was added to the **btnLoan** which registerred the amount input and if the input amount is above 0 and at least one of the movements inside the current account has a deposit which is at least 10% of the total amount requested. If these requirements are satisfied the current movements will have the requested loan amount added to its array, the UI is updated to reflect the changes and input values reset.
9. You need to display information about the end of the session. The app automatically logout the user after 10 minutes if there is no activity in the app. The timer resets if there is an activity in the app.

**The graphical interface and design look like about:**



**However, you can implement graphical interface and design up to you!**

**Good luck!**