

Suppose you are a loan-giving platform that connects borrowers and lenders. Write a web application that simulates the loan giving process.

The application must be able to:

1. Register multiple users. A registered user must stay registered after a page refresh.
2. Login an user. A logged user must remain logged even after a page refresh.
3. After successful login the user is redirected to a home page where he can apply for a loan. The user doesn't have access to that page before he is logged in.
4. The home page has a loan application form with the following information:
  - `borrowerName`: The name of the borrower. Disabled field. Shows the username of the user.
  - `borrowerIncome`: The monthly income of the borrower. Input type number with appropriate validation.
  - `requestedAmount`: The amount of money the borrower is requesting as a loan. The minimum amount is 1000 lv.
  - `requestedTerm`: The term (in months) for which the borrower is requesting the loan. The minimum term is 6 months.
5. After application the requested loan goes to the **applications overview** page. There the user sees a table with the following information:
  - a. id of the loan - number, unique for all loans in the platform.
  - b. requested amount
  - c. requested term
  - d. status
  - e. cancel button

After application the loan goes in pending status. Each loan application gets evaluated for eligibility for 60 seconds. The user is able to cancel the loan application via a cancel button. After the evaluation period the loan goes either in approved or rejected status.

If a loan is approved a new button appears - view offers. After clicking the button, below the table appear offers from the 3 lenders in the system for the specific application. Take in mind that some of the lenders might not provide an offer. So the minimum number of offers is 1 and the maximum is 3.

6. Each offer should include the following information:

- a. `interestRate`: The interest rate for the loan.
  - b. `loanAmount`: The amount of money the lender is willing to lend. ( May be less than the amount requested )
  - c. `monthlyPayment`: The monthly payment the borrower will have to make to repay the loan.
  - d. `loanTerm`: The term of the loan in months.
7. The user can decide which offer to take. After clicking a button take the offer, the loan application is now deleted and is converted to a loan. The loan is visible in the **loans overview page**. The page contains all the information for the loan and an additional column - total owned amount. This column is calculated based on the criteria: loan amount + interest.  
There is also a button **repay in full**. If the user has the needed amount, he can repay the loan in full.
8. The currently owned money by each user is calculated based on the time passed. Each 30 seconds in the platform represent a month passed. After each month the currently owned money is increased by the monthly income amount. The starting amount for each user is a random integer between 0 and 10 000.
9. Information for all loans is kept in localStorage.
10. The system has a special type of user - admin. He can access a page - loan statistics , but cannot request a loan.
11. In the statistics page the admin is able to see all the loans that have gone through the system in a form of a table.. A loan can be in progress, or repaid.

Additionally the following info is shown:

- Number of loan applications that were eligible
- Number of loan applications that were rejected
- Number of loan applications for each lender
- Total loan amount requested
- Total loan amount approved
- Total monthly payment for all loans
- Total number of loan applications

The system calculates the interest rate for the loan based on the following criteria:

- If the borrower's income is less than 20,000, the interest rate is 10%.
- If the borrower's income is between 20,000 and 50,000, the interest rate is 8%.
- If the borrower's income is greater than 50,000, the interest rate is 6%.

The system determines the eligibility of the borrower based on the following criteria:

- If the requested amount is less than or equal to 50% of the borrower's annual income, the borrower is eligible.
- If the requested amount is greater than 50% but less than or equal to 100% of the borrower's annual income, the borrower is eligible if the loan term is less than or equal to 24 months.
- If the requested amount is greater than 100% of the borrower's annual income, the borrower is not eligible.

Assume that there are three lenders on the platform with the following criteria:

- Lender 1: Interest rate  $\leq 7\%$ , Maximum loan amount = 50,000
- Lender 2: Interest rate  $\leq 9\%$ , Maximum loan amount = 100,000
- Lender 3: Interest rate  $\leq 11\%$ , Maximum loan amount = 150,000

All actions in the loan giving business are asynchronous. Implement that logic with Promises.

All the data must be persisted in the localStorage.

If there is some unclarity in the requirements, don't hesitate to ask the client.

Style the system by your desire.

VCS ( GIT) must be used for the implementation of this task. Use branches for each feature - authentication, loan applications, loans, admin user, statistics.