Challenges and Enhancements

I will discuss the challenges and enhancements which I have made for the banking application followed by a screen capture video of this functionality.

<1> In the amount for deposit and withdrawal, using the Number() function to convert the string to a number produces 0 (not NaN) for an empty string or a string of space chars. Instead, the Deposit or Withdrawal button should be disabled. I handled this problem by checking whether amount.trim() is null before applying the Number function.

<2> Multiple users can be added to the bank, so we need to determine whose account to show the balance and apply deposits and withdrawals. I am using the account which was most recently either created or successfully logged into. [An unsuccessful login removes all accounts from consideration. The deposit and withdrawal pages then show the message “Please Log In”.] Creating an account starts off with zero balance, but the account created is automatically logged into so that an initial deposit can be made on the Deposit screen.

What may be an enhancement is maintaining a separate balance for each account and remembering it when another account becomes logged in. After logging into another account, we can log back into any previous account and its balance will be shown and deposits & withdrawals can be done on it. I have accomplished this by adding to the context an index into the array of users which references the user currently logged in. [The balance of all accounts is shown on the All Data screen.]

<3> Another challenge is displaying the **All Data** page in a Bootstrap Card in a tabular form. This combines our work with React, Bootstrap Cards, and the CSS Grid functionality. For the **All Data** page, I created a separate version of the Card function with className “content” and added the style for “content” to index.html to implement the Grid format. I created an array with the name, email, password, and balance for each user and displayed it in 4 column Grid format.

<4> I used a <Card> form of the type given to us in **CreateAccount** for the Login, Deposit, and Withdrawal pages. This is to provide <input> elements to be entered by the user and a <submit> button to submit the data that was entered.

Input Validation to Demonstrate

For Create Account:

<1> Disabled **Create Account** button when all input fields are blank.

<2> If one input field is blank, pressing **Create Account** button display an error message and does nothing. A specific error message is displayed if password < 8 chars.

For Deposit & Withdrawal:

<3> Disabled Deposit & Withdrawal button if amount field is blank.

<4> If amount is not a number, error message is displayed when Deposit or Withdrawal is pressed.

Similarly for negative amount.

Before recording, COPY PASSWORD TO CLIPBOARD.