Full-Stack Developer – Technical Assignment

Create a Web Application that simulates small Bank's Back-Office system that allows creation of new accounts by bank employees and transactions between them.

Functional Requirements

I. Creation of accounts with details:

- Full Name
- Full Address
- Account Type (Personal / Company / Other)
- Initial deposit (money deposit during the account's creation)
- Currency (the default currency for the account, e.g., GBP, USD, or others)

Notes:

- *After the creation of the account, the application will provide new "Sort Code" and "Account Number".*
- *If there is an account with the same full name, under the same address, the application will not allow creation of a new account with the same details.*
- *Validation will prevent creation of the accounts with empty fields or incorrect values.*

II. Searching for existing accounts:

- By using one, or more account fields
- Selection of the account from search results list to edit its details

III. Transferring money between accounts:

- Source account (select account from existing accounts)
- Destination account (select account from existing accounts)
- Transfer amount
- Currency (the currency of the destination account)
- Final balance (the balance in source account after the transfer)

Notes:

- *Ensure that the transfer amount is less or equal to the source account balance before the transfer or during the typing of the transfer amount.*
- *Final balance = Current Balance – Transfer amount * Currency*
- *Transfer operation will present a dialog confirming the operation (success/failure) and its details*

Technical Requirements

- **Server side:**
  - DB to contain all accounts data and other info (can be SQL or NoSQL DB)
  - API to allow the described functionality
- **Client side:**
  - Build UI in any modern Web UI framework
  - UI should be intuitive and "Fool Proof"
- **Submission:**
  - Share project's code via on of the services:
    - GitHub
    - Bitbucket
    - Heroku

Bonus (Nice to Have)

Functional

- Pull address details by entering postcode only.
  - Use one of the APIs:
    - getaddress.io
    - postcodes.io

- Allow submission of documents during the creation of the account.
  - Document types:
    - Passport
    - Driving License
    - Utility Documents
  - Store documents on server and their references in the DB

- Pull currency rates from 3rd party sources:
  - Use one of the APIs:
    - exchangeratesapi.io
    - fixer.io
    - openexchangerates.org
    - currencylayer.com

Technical

- Host application on cloud (AWS, Azure, Heroku or other)
- Docker container to deploy and to host the application
- Use web-sockets to communicate between client and server (push data from server to client via socket, e.g., socket.io library)
- Responsive UI that will work in any common resolution/aspect-ration (desktop and mobile)

Do not hesitate to ask any questions!

Good Luck!