

Mercury Assignment: Currency Exchange Application:

Problem Dissection & Overview:

Overview

Create a simple SPA that allows a user to login into an exchange account, review past exchanges with associated currencies, request new exchanges which update the table displaying past exchanges, and logout.

Dissection:

1. Login Screen
 - a. Basic form which is submitted with credentials (email and password) checked against database entries (Hardcoded for this assignment).
 - b. Requires Form, Inputs and submit button and
 - c. Validation of credentials.
2. Trade History Screen
 - a. Tables/list of past currency exchanges – this can be stored in local storage for now – can save into a JSON object.
 - b. Link to page to book another currency exchange, as well as logout.
3. New Trade Screen
 - a. Inputs for:
 - i. Currency they want to sell.
 - ii. Currency they want to buy.
 - iii. Amount they have to exchange.
 - b. User should be able to specify the amount they are wanting to sell or the amount they want to buy.
 - i. Selling currency needed to buy specified amount of buying currency
 - ii. Buying currency that can be purchased with amount of selling currency
 - c. Query exchange rate API and calculate the exchange
 - d. Review exchange, and accept or cancel the transaction
 - e. If user accepts
 - i. Add transaction to local storage and return to Trade Screen (2)

Major Tasks

1. State Management
 - a. Redux - Too complex for application size.
 - b. Provider and Context? – Like Provider-Consumer in Flutter?
 - i. Read Documentation.
2. User Authentication
 - a. React Router - Private Routes (Trade History and New Trade) – Public Login
 - b. Authentication Hook to set and remove JSON web token.
3. Trade History ‘Database’
 - i. Probably Provider and useContext

Steps to Solution:

1. Review, define and clarify problem.
2. Determine design system/theme using existing Mercury FX UI.
3. Design Screens in Figma.
4. Establish components break down.
5. Implement basic functionality using State Management (Context Provider)
 - a. User Authentication
 - i. Set JSON web token in local storage
 - ii. Access to Auth Context
 - b. Trade History
 - i. Create 'Wrapper' for Trade Pages
 - ii. Access to stored Trade History
 - iii. Render table row for each trade
 - c. New Trade
 - i. Query Exchange API to generate list of currency options
 - ii. When user enters an amount – query api to get the exchange between those currencies
 - iii. On request of that trade – Modal popup to accept or cancel trade
 1. On accept – update Trade History
 2. Cancel – Just cancel
6. Style Pages and components
 - a. Mobile First?
 - b. SCSS – Decide on file structure – Embedded?

Colour

Primary Colours:



Main Pink
#E73985



Main Purple
#622C8E



Gradient

Secondary Colours:



Soft Grey
#A1A2A6



Light Grey
#CBCBCB



Dark Grey
#757775

Font

Titles

Paragraphs

Quicksand

Regular - 400

Bold - 700

Aa

Open Sans

Light - 300

Regular - 400

Semi-bold - 600



Welcome to Mercury FX

Buy and Sell Forex Easily, Safely



Email



Password

[Forgot Password?](#)

[Login](#)

[Signup](#)

Welcome James

Your Trade History

Date	Currency Sold	Currency Bought
31 June 2020	ZAR 5000.00	USD 294.25
20 June 2020	GBP 2000.00	EUR 2210.94

Trade **Forex**

Buy and Sell

Create Your Offer

Currency I Have:

USD - Dollar

Currency I Want:

GBP - Pound

Amount:

1256.65

Amount:

965.73

Place Request

Review Your Offer

You are Selling:

USD - Dollar 1256.65

For:

GBP - Pound 965.73

Confirm Exchange

- Modal



Welcome to Mercury FX

Buy and Sell Forex Easily, Safely

Email

Password

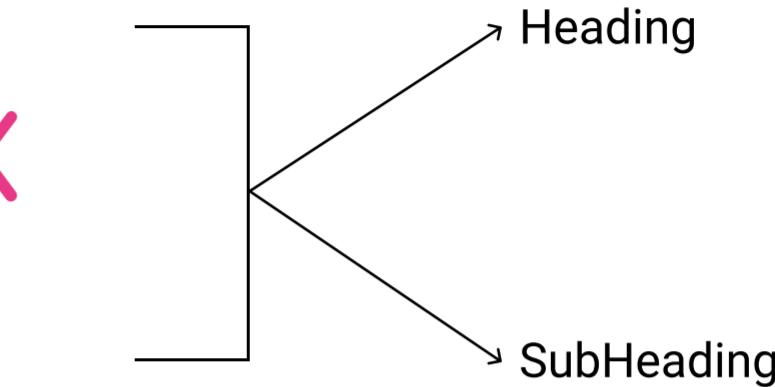
[Forgot Password?](#)

[Login](#)

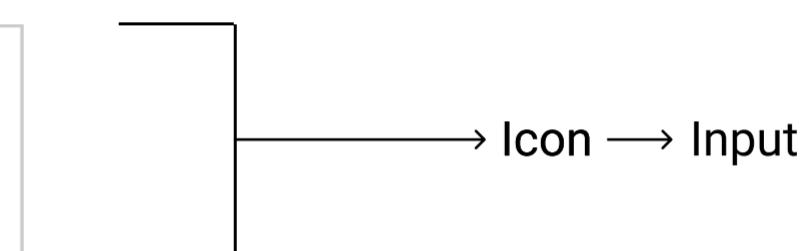
[Signup](#)

↓
Login Form

Page Title



Input



Button Row

