CMPS 356 Enterprise Application Development - Spring 2019
Lab 10 – React JS

Objective

The objective of this lab is to practice implementing Web UI and calling Web API using React JS . Particularly in the Lab you will learn how to

- setup a react application inside an express app
- Implement components using React
- use the useState, useEffect react hook
- rendering list
- compose components
- pass data between components using props
- implement routing using the react-router
- handle events

Overview

This lab has two parts:

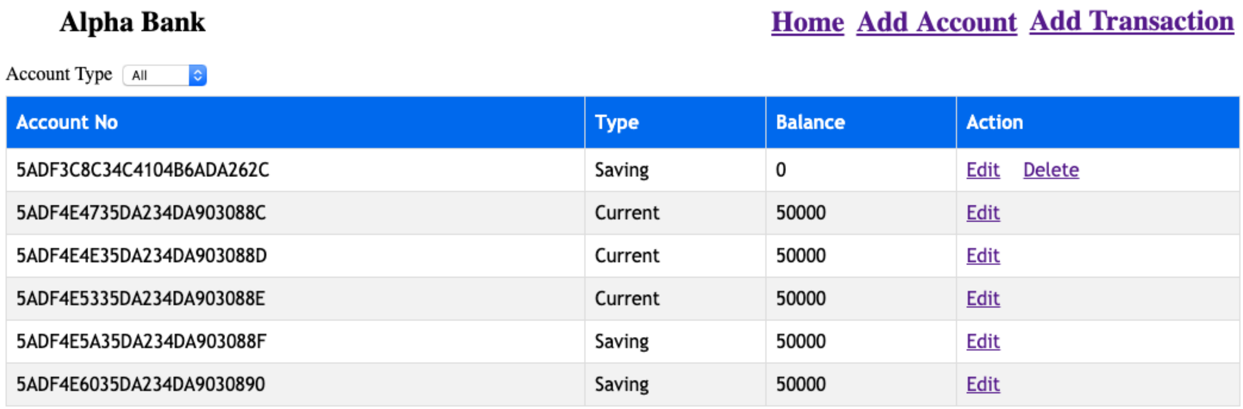
- **Part A**: Extend the **BankingApp** to make the app functionality accessible via Web UI using React JS (1.5h).
- **Part B**: Extend the **BookStore** App to make the app functionality accessible via Web UI using React JS (1.5h).

Part A - Extend the Banking App to make the app functionality accessible via Web UI

1. Sync cmps356-content repo to get the Lab files.
2. Copy ***Lab10-ClientJS*** folder from cmps356-content repo to your repository.
3. Open ***Lab10- WebApi \BankingApp*** in WebStorm. Run **npm install** to install the packages.
4. Develop a Web UI using React JS to make the App functionality accessible to users:

Components	**HTTP Verb**	**Service to use**	**UI functionality**
AccountList	get	/api/accounts/:acctType	Gets and displays all accounts. Provide a dropdown to enable selecting the Account Type 'Saving Account', 'Current Account' or 'All'.

In AccountList component , provide a ‘Delete’ button besides every account with balance=0 and an Edit Button next to all the accounts	Delete PUT	/api/accounts/:id /api/accounts/:id	Allows deletes an account by id from both the html table of accounts and the accounts file.

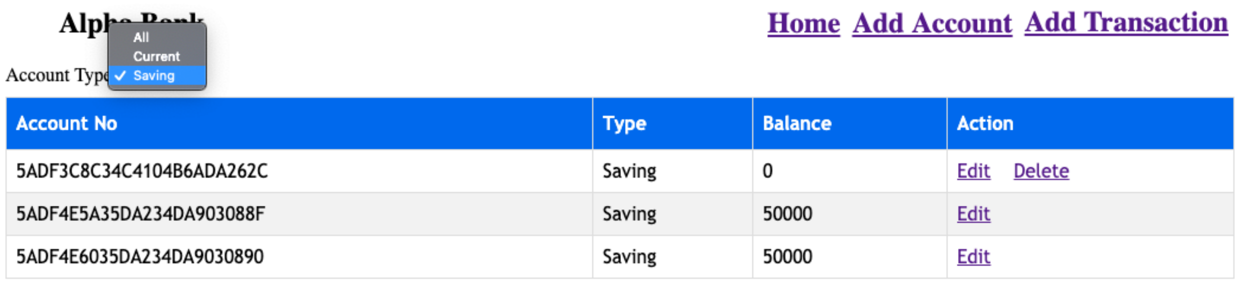


Alpha Bank

Home Add Account Add Transaction

Account Type: All

Account No	Type	Balance	Action
5ADF3C8C34C4104B6ADA262C	Saving	0	Edit Delete
5ADF4E4735DA234DA903088C	Current	50000	Edit
5ADF4E4E35DA234DA903088D	Current	50000	Edit
5ADF4E5335DA234DA903088E	Current	50000	Edit
5ADF4E5A35DA234DA903088F	Saving	50000	Edit
5ADF4E6035DA234DA9030890	Saving	50000	Edit



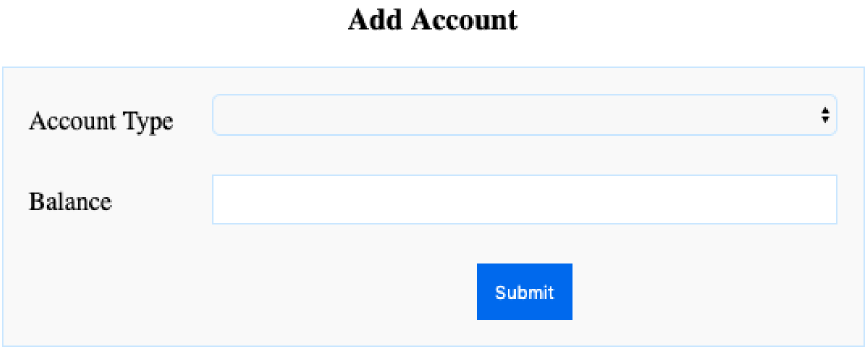
Alpha Bank

Home Add Account Add Transaction

Account Type: Saving

Account No	Type	Balance	Action
5ADF3C8C34C4104B6ADA262C	Saving	0	Edit Delete
5ADF4E5A35DA234DA903088F	Saving	50000	Edit
5ADF4E6035DA234DA9030890	Saving	50000	Edit

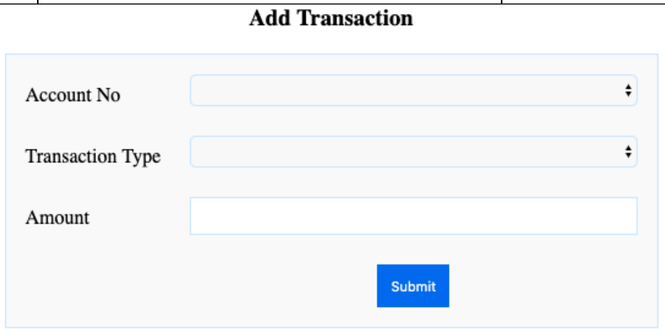
Add Account Component	Post	/api/accounts	Allow creating a new account.



Add Account

Account Type:

Balance:

Account Transaction Component	Post	/api/accounts/:id/trans	Allows selecting a particular account from a dropdown then submitting a deposit or withdrawal transaction. This form should display the new account balance. Display any error returned from the Web API (e.g., insufficient balance).
 Add Transaction Account No Transaction Type Amount Submit			

All the app pages should be accessible from the index.html page.

5. Test the Banking App Web UI and provide screenshots as evidence in the testing document.

Part B - Extend the Book Store to make the app functionality accessible via Web UI using React JS.

<u>DeadLine – Next week 1 hour before the Lab</u>

1. Open ***Lab10- WebApi \BookStore*** in WebStorm. Run **npm install** to install the packages.
2. Develop a Web UI to make the App functionality accessible to users: You should create all the necessary UI components that are needed for the BookStore backend applicatons. You can see the routes to understand what those functionalities are. For example your application should allow the user to,
   - List all of the books
   - Filter books by names, authors, category etc… as you have done in Lab 7 Web UI
   - Edit , update, delete and see the complete details of a given book
   - The user should be able to borrow a book
     - Allow the user to select from available books drop-down
     - Allow the user to select from a dropdown the available borrowers
     - Add the borrow info to the borrowing collection
   - Show Summary
     - Books
     - Top 3 Borrowers
     - Top 3 Borrowed Books

        - Summary for Borrowed Categories

- **a video demo will be posted soon**

3. Test the **BookStore** Web UI and provide screenshots as evidence in the testing document. For more details, check the second document.

After you complete the lab, fill in the ***Lab10-TestingDoc-Grading-Sheet.docx*** and save it inside **Lab8-ClientJS** folder. Sync your repository to push your work to Github