

CMPS 356 Enterprise Application Development - Spring 2020

Lab 9 – Client-side JavaScript

Objective

The objective of this lab is to practice implementing Web UI and calling Web API using Client-side JavaScript. Particularly DOM manipulation, events handling and using Fetch API.

Overview

This lab you will extend the Banking App to make the app functionality accessible via Web UI using Client-side JavaScript . Then you need to solve the assessment posted under Lab/Assessments.

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 **Lab8-ClientJS** folder from cmps356-content repo to your repository.
3. Open **Lab8- WebApi \BankingApp** in WebStorm. Run **npm install** to install the packages.
4. Develop a Web UI to make the App functionality accessible to users:

Web Page	Functionally
index.html	<p>First, the index page should have a main navigation menu providing 3 links:</p> <ul style="list-style-type: none">• <u>Home</u> link is the home page that allows getting accounts.• <u>Add Account</u> link allows adding an account.• <u>Add Transaction</u> link allows adding a transaction. <p>- All the app pages should be accessible from the index.html page.</p> <p>- This page allows getting accounts by type. It provides a dropdown to select the account type: <i>Saving</i>, <i>Current</i> or <i>All</i>.</p> <p>When the page load or when a different account type is selected then the page should fetch the accounts from /api/accounts?acctType=</p> <p>- For each account with the balance=0, a <i>Delete</i> button is provided to enable deleting the account. Then this button is clicked the account should be deleted using /api/accounts/:id Web API. Also, the account row should be deleted from the html accounts table.</p>
acct-form.html	<p>Allows creating a new account by posting the account data to /api/accounts</p>
acct-trans.html	<p>Allows selecting a particular account from the accounts dropdown then submitting a deposit or withdrawal transaction.</p> <p>The new transaction should be posted to /api/accounts/:id/trans. The page should display any error returned from the Web API (e.g., insufficient balance).</p>

5. Test the Banking App Web UI