

CMPS 350 Web Development Fundamentals

Lab 09 – Web Pages using Next.js and React

Objective

The objective of this lab is to practice implementing Web Pages and calling Web API using Next.js, including:

- Next.js routing
- Implement components, pages, and layouts
- Compose smaller components to build larger UI components
- Use useState react hooks in client-side components
- Pass data between components using props
- Handle events in client-side components

This lab has 2 parts:

- **Part A:** Extend the **BankApp** to make the app functionality accessible via HTML pages placed under the **public** folder.

Then enhance the **BankApp** to make more modular and simpler using React Components and Next.js Web pages.

- **Part B:** Extend the **TodoApp** to make the app functionality accessible via Web pages using Next.js

Part A - Bank App

1. Copy **lab9-web-pages** folder from the lab repo to your repository.
2. Open the **bank-app** subfolder in VS Code. Then run **npm install** to install the packages. Finally, run the application using **npm run dev** to test if everything works properly.
3. Modify the html pages available under the **public** folder to make them dynamic by using the bank API developed in the previous Lab.
4. Develop React Components and Next.js Web Pages to make the app UI more modular.

Components	Web API HTTP Verb	Web API to use	UI functionality
AccountList	GET	/api/accounts/:acctType	AccountList displays a list of accounts. The accounts pages should get the list of accounts and pass them to the AccountList to display them. This page should also provide a dropdown to enable filtering

			<p>the accounts by Type (i.e., ‘Saving Account’, ‘Current Account’ or ‘All’).</p> <p>Note that the accounts page should be the main app page (i.e., the first page to be shown when the user visits the app).</p>
--	--	--	---

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 All
Current
Saving

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

AccountEditor Component	POST	/api/accounts	Allows creating a new account.
-----------------------------------	------	---------------	--------------------------------

Add Account

Account Type

Balance

Submit

In AccountList component, provide a Delete button besides every account with balance=0	DELETE	/api/accounts/:id	Allows deleting an account by id from both the server-side and the accounts html table
In AccountList component, provide Edit Button next to every account	PUT	/api/accounts/:id	Allows updating the account owner details in both the server-side and the accounts html table. The update form should reuse the Account Editor component but the account number, type and balance should not be editable.
Account Transaction Component	Post	/api/accounts/:id/trans	Allows creating account transaction by selecting a particular account from a dropdown then submitting a deposit or withdrawal transaction. This form should display the new account balance after successfully adding the transaction. Otherwise, it should display any error returned from the Web API (e.g., insufficient balance).

Add Transaction

Account No

Transaction Type

Amount

Submit

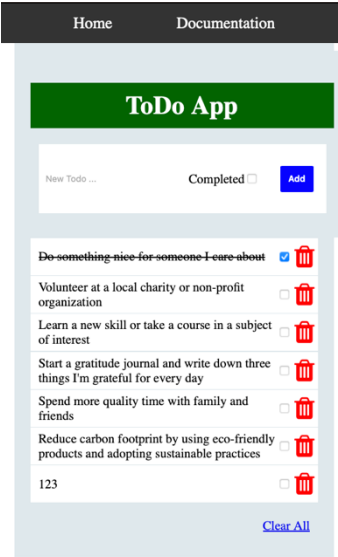
In AccountList component, provide Trans Button next to every account	Get	/api/accounts/:id/trans	Gets and displays the transactions for a particular account.
<p>Create Header component to display the bank Logo and a Welcome message as well as the Navigation menu.</p> <p>Create a Footer component to display a copyright message an link to facebook page.</p> <p>Add a layout file to ensure the Header and Footer are displayed on every page.</p>			

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

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

Part B - TodoApp

1. Open the **todo-app** folder in VS Code. This folder contains the app that you created in Lab 7 using plain JavaScript.
2. Install the necessary packages by running **npm install**, then start the development server by running **npm run dev**. Test the provided backend Web API using Postman.
3. Develop React Components and Next.js Web Pages to allow users to access the application functionalities, including:
 - View a list of all the todos
 - Add, update, delete todos
4. Add a navigation bar that allows the user to navigate to a **documentation** page that lists all the available Web API of the current application.



Home Documentation		
API Route	HTTP Method	Description
/api/todos	GET	Retrieves all the todos.
/api/todos/1	GET	Get specific todo by id

5. Test the **ToDoApp** Web UI

After you complete the lab, push it to your Github repo.