

BANKING APPLICATION

IIHT

Time To Complete: 2 hr

CONTENTS

1	Problem Statement	3
2	Proposed Virtual Banking Application Wireframe	4
2.1	Welcome Page	4
3	Business-Requirement	7
4	Validations	7
5	Constraints	8
6	Mandatory Assessment Guidelines	8

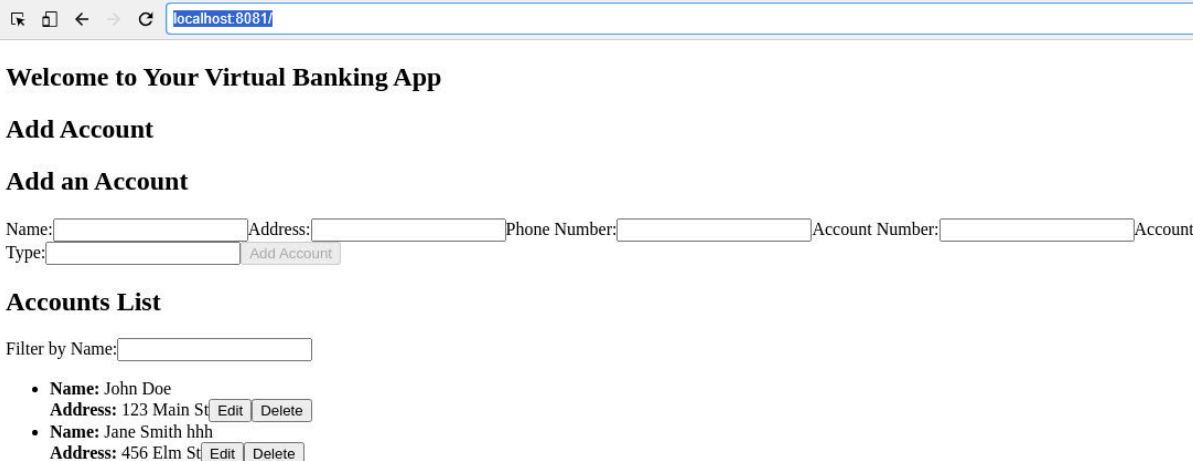
1 PROBLEM STATEMENT

"Virtual Banking Application" is a Single Page Application (SPA) that empowers a bank to list all its account holders with details and has the option to edit and delete any also.

2 PROPOSED VIRTUAL BANKING APPLICATION WIREFRAME

UI needs improvisation and modification as per given use case and to make test cases passed.

2.1 WELCOME PAGE



A browser window with the address bar showing 'localhost:8081/'. The page content includes a welcome message, an 'Add Account' section with input fields for Name, Address, Phone Number, Account Number, and Account Type, and an 'Accounts List' section with a filter by name and a list of two accounts with edit and delete buttons.

localhost:8081/

Welcome to Your Virtual Banking App

Add Account

Add an Account

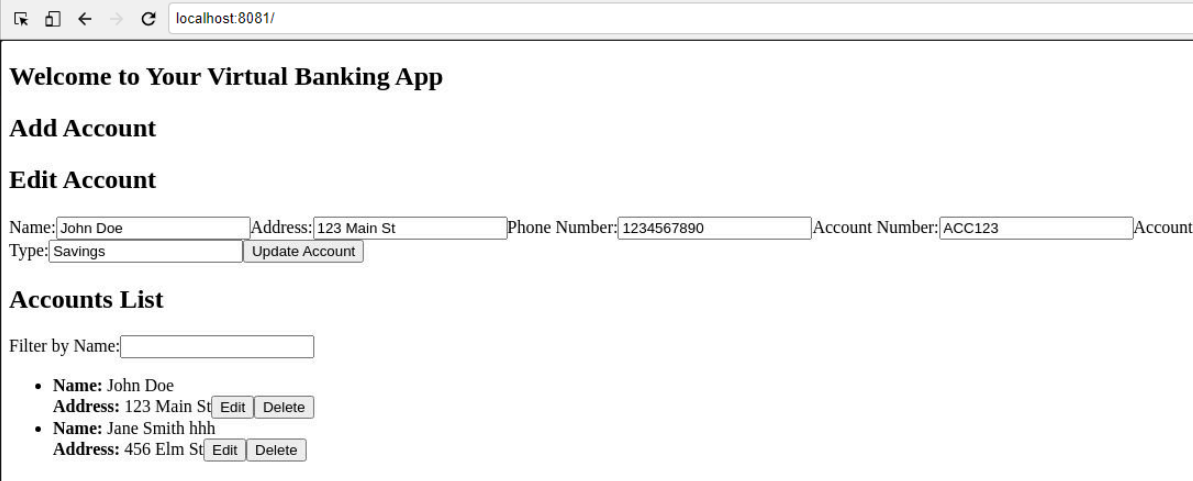
Name: Address: Phone Number: Account Number: Account Type:

Accounts List

Filter by Name:

- **Name:** John Doe
Address: 123 Main St
- **Name:** Jane Smith hhh
Address: 456 Elm St

SCREEN SHOTS



A screenshot of the application running in a browser. The 'Add Account' section now has an 'Edit Account' section above it. The 'Accounts List' section shows the same data as the wireframe, but the 'Update Account' button is present instead of 'Add Account'.

localhost:8081/

Welcome to Your Virtual Banking App

Add Account

Edit Account

Name: John Doe Address: 123 Main St Phone Number: 1234567890 Account Number: ACC123 Account Type: Savings

Accounts List

Filter by Name:

- **Name:** John Doe
Address: 123 Main St
- **Name:** Jane Smith hhh
Address: 456 Elm St

3 BUSINESS-REQUIREMENT:

As an application developer, develop the Virtual Banking Application (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	<p>As a user I should be able to visit the welcome page as default page.</p> <p>Acceptance criteria:</p> <ol style="list-style-type: none">1. Users should be able to see a form to add a new account holder.2. Users should be able to see the list of all details of account holders.3. It must have Name, Address, Phone Number, Account Number & Account Type fields along with an Add Account button.4. List should have an Edit and Delete button to edit and delete a accounts respectively.

4 VALIDATIONS

- All required fields must be fulfilled with valid data.
- In the starting "Add Account" button should be disabled.
- Only after validating fields, "Add Account" button should be enabled.
- Once we click on "Add Account", a new account must be added to the list.

5 CONSTRAINTS

- You should be able to press the "TAB" key and "SHIFT + TAB" to navigate from top field to bottom field and vice-versa.
- On clicking the "Add Account" button, a new account must be added with entered fields.
- "Add Account" button will be disabled until all validations are fulfilled.

6 MANDATORY ASSESSMENT GUIDELINES

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. This editor Auto Saves the code.
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
7. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

Note: The application will not run in the local browser

8. You can follow series of command to setup React environment once you are in your project-name folder:
 - a. npm install -> Will install all dependencies -> takes 10 to 15 min
 - b. npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:8080/8081 to open project in browser -> takes 2 to 3 min
 - c. npm run json-start -> As we are using a json server to mimic our db.json file as a database. So, this command is useful to start a json server.
 - a. npm run jest -> to run all test cases and see the summary. It takes 5 to 6 min to run.

- d. `npm run test` -> to run all test cases. **It is mandatory to run this command before submission of workspace** -> takes 5 to 6 min
9. You may also run “`npm run jest`” while developing the solution to refactor the code to pass the test-cases.
10. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on **“Submit Assessment”** after you are done with code.
11. You need to use `CTRL+Shift+B` - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.