AccuLynx UI Challenge

Overall Goal

Develop the UI workflow for an Address book.

Please send us whatever you have completed after 1 week. Feel free to contact us if you have questions.

The project description is pretty general on purpose because we want to see how you interpret it and make it unique. This is your chance to show your creativity and ability to create solutions. Show us how you would lay the foundation for a custom CSS framework and component library of your own; don't use bootstrap, foundation, or other CSS frameworks. You may use a JavaScript framework like Vue, but do not use plugins. We want to see how you write and organize code and how you would go about creating your own component library.

Requirements

1. Build and implement a component library for a responsive address book webapp.
2. Account for proper form validation to be displayed (just check for empty fields, don't worry about formatting or password strength).
3. The application should include the following
    a. Contact List
        i. This view appears on screen along with the Contact Details View at desktop widths
        ii. This view appears by itself at mobile widths
        iii. Has an empty state when there are no contacts
        iv. Has a button to create a new contact
        v. Each contact is displayed on a separate card. A card contains the contact's photo, First and Last Name, and primary phone number.
    b. Create Contact
        i. This is a form containing:
        ii. Contact Photo (this can be hardcoded stock photos)
        iii. Salutation
        iv. First and Last Name (required fields)
        v. Company Name
        vi. Phone number (required field), Type of number (work, cell, home) (required field)
            1. Allow for adding as many phone numbers as the customer needs
            2. Allow for setting one of the phone numbers as "primary"
    c. Contact Details
        i. This view appears on screen along with the Contact List View at desktop widths
        ii. This view appears by itself at mobile widths
        iii. Has an empty state when no contact is selected
        iv. Color-code each phone number by its type (work, cell, home) using three colors of your choice. One color for each type of phone number.
        v. A button to delete the contact. Clicking the delete button will remove that contact from the list.
        vi. A button to edit the contact. Clicking the edit button will allow the user to update the contact's information.

1. While editing, Save and Cancel buttons are shown.
2. Save will update the contact’s information.
3. Cancel will revert the contact to its previous state.

4. Header
    a. Should be fixed to the top of the viewport and always remain on top of content
    b. Display an app name of your choice
5. Upload your solution to a public repository on github, gitlab, or other source control repo. Include in your readme instructions on how to run your application after cloning the repo