Assignment 1

Start Assignment

Due Jul 16 by 11:59p.m. **Points** 100 **Submitting** a text entry box or a website url

For this assignment, you are going to build part of sign-up process of an app.

- **Important:** Your code should work on both iOS and Android platforms. Please test that before submission.
    - Your submissions will be randomly tested on Android or iOS and it should not crash and work/look correct.

There are 3 screens in this app that you need to switch between them programmatically. That means that you should **not** be using any library for navigation. You need to have state variables to keep track of the state of the signup process (e.g. are the entered values valid? has the user confirmed the entered values? Does the user want to finish the signup later? etc.) and based on those states render different screens.

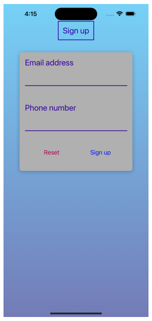
- Hint - conditional rendering:
    - **https://reactjs.org/docs/conditional-rendering.html** (https://reactjs.org/docs/conditional-rendering.html)
    - **https://www.robinwieruch.de/conditional-rendering-react/** (https://www.robinwieruch.de/conditional-rendering-react/)

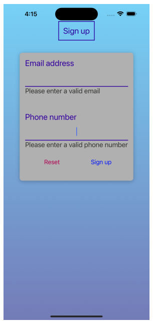
Think about which component will be the best place to store all these states.

You should have 3 different components (saved in "screens" folder) for these screens that will render smaller components (saved in "components" folder).

Screens:

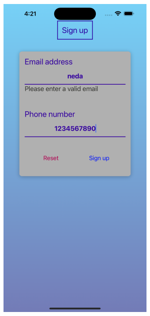
- **Starting** screen: It shows a card (a view with rounded corner and shadow) with two Texts, two TextInputs and two Buttons (Reset and Sign up). The user should enter a valid email address and a valid (10 digit) phone number. After pressing Sign up, if the input value(s) is(are) invalid or empty, a Text should appear underneath the TextInput that has the invalid value.
    - Hint:
        - **https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/length** (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/length)
        - **https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/isNaN** (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/isNaN)





Please enter a valid email

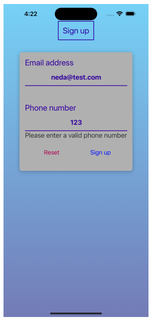
Please enter a valid phone number



neda

Please enter a valid email

1234567890



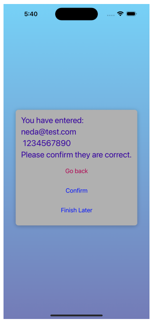
neda@test.com

123

Please enter a valid phone number

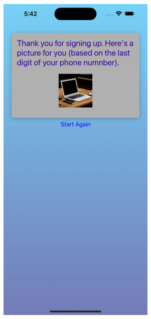
- **Confirm** screen: After user enters valid information and presses Sign up, the Confirm screen is shown in a modal. This screens shows a card with information that user entered previously and 3 buttons. User has 3 options:
    - Go back to the starting screen and edit what they entered (User should see the TextInputs filled with the entered values).
    - Confirm the information.
    - Finish the process later.

  In the last two cases, the user will be redirected to the Finish screen

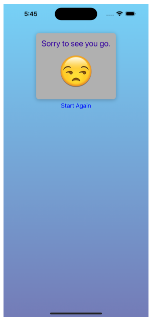


You have entered:
neda@test.com
1234567890
Please confirm they are correct.

- **Finish** screen: A card will be shown to the user with the following info:
    - If they have confirmed the information, they will see a "Thank you" message and an image with a URL constructed based on the last digit of their phone number value as its source (e.g. if the last digit is 4 the URL should be **https://picsum.photos/id/4/100/100)** (https://picsum.photos/id/1024/100/100)
    - If they had chosen to finish later, they will see a "Sorry" message and a sad smiley face (Google a sad smiley face, save a file in your code and use require() to set the image source).
    - There should be a button to start the process again where all the information (email, phone, what user has selected before) is reset.



Thank you for signing up. Here's a picture for you (based on the last digit of your phone numnber).



Sorry to see you go.

Notes:

- Your code should be split into components that are reusable and small enough (e.g. card, input, button, certain types of texts,...)

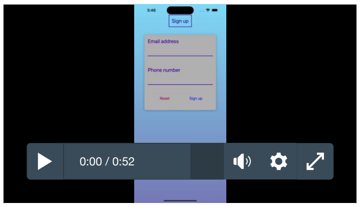
- To add shadow to the card component, you can use **elevation** (https://reactnative.dev/docs/view-style-props#elevation-android) and **shadow** (https://reactnative.dev/docs/shadow-props) props
  - Hint: You need background color in iOS for the shadow to be shown
- To pass children to Card component use children prop:
  - **https://reactjs.org/docs/composition-vs-inheritance.html#containment** (https://reactjs.org/docs/composition-vs-inheritance.html#containment)

- Instead of copying the color values in different places, use a .js helper file that exposes (using export) different color values as a JavaScript object.
- Use this package **https://docs.expo.dev/versions/latest/sdk/linear-gradient/** (https://docs.expo.dev/versions/latest/sdk/linear-gradient/) to provide a linear gradient for all the screens (Think about how you can prevent repeating the same styling in 3 places)
- You don't need to follow my color choices but please have the same style and layout

Submission:

Please do all the implementation in a branch other than the master/main branch. When you're done make a pull request and assign it to the TAs. Submit your GitHub link on Canvas

Demo:



Assignment 1

Criteria	Ratings		Pts
The screens have consistent styles and show the required components	**15 pts** **Full Marks**	**0 pts** **No Marks**	15 pts
The entered email and phone number are stored throughout the process correctly	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Navigation between screens is done programmatically	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
A helper file is used to expose common styles (e.g. color values)	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
The code is clean, professional, and well-commented and broken down into appropriate components.	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
The email and phone number entered by the user are checked properly	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
Sign up and Reset buttons on the starting screen work correctly Sign up button should show the proper error message. Reset should erase everything	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
Confirm screen is shown in modal	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
All the buttons on the Confirm screen work correctly	**15 pts** **Full Marks**	**0 pts** **No Marks**	15 pts
Correct pic is shown to the user on the Final screen	**15 pts** **Full Marks**	**0 pts** **No Marks**	15 pts
"start again" button on the Final screen takes user to the starting page with all the info erased	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
			Total Points: 100