Assignment 1

This is an individual assessment.

The assessment evaluates your knowledge the introductory concepts in React Native discussed in Weeks 1-3, with emphasis on:

1. Creating a React Native project using Expo
2. Compiling and running an application on a mobile device
3. Using React Native's built in user interface elements to design a single screen application
4. Using an icon library
5. Capturing user input using form field elements
6. Styling the user interface elements
7. Implementing an event listener
8. Implementing program logic that performs computations

When creating your solution, you must use the coding practices and conventions demonstrated in class. A solution that does not reflect what was taught in class will not be accepted (0 grade) and/or be subject to an academic integrity review.

Submission Instructions

For your submission to be graded, you must provide a **zip file** of your project and **demonstrate** your project running on a mobile device.

1. Creating your project:

- Projects must be created using the Expo CLI.
- When creating your project, name the project: **EV-firstname** (replace **firstname** with your name)

2. Create a zip file and submit

- When finished, zip the entire project and name the zip file **EV-firstname.zip**
- Your finished zip file may be very large. Ensure you budget enough time to upload the zip file by the due date.
- Submit your zip file to the dropbox.

3. Demo your app

1. Attend a live class session within 1 week of the assignment due date
2. During the class, connect with the instructor to demo your app.

Academic Integrity

- This is an individual assessment.
- Permitted activities: Usage of Internet to search for syntax only; usage of course materials
- Not permitted:
  - Following step-by-step tutorials
  - Posting assessments to homework help websites, such as Chegg or CourseHero
  - Discussion of solution or approaches with others; sharing/using a "reference" from someone
  - Using generative AI tools, such as ChatGPT, Copilot, etc

Problem Description

You've been hired by an electric car company to create an application that enables the user to compare the costs of an electric car vs. gas car. The app must answer these questions:

1. Given the price of gas, compare the distance a gas car can travel vs. an electric car?
2. How much does the user save per year by driving an electric car?

User Interface

The user interface must contain:

1. title of the application
2. form fields for capturing data about gas/electric vehicles and number of km driven per year
3. button to activate calculations
4. results summary (user interface elements to show the results of calculations)
    - Note that the results summary is always displayed, you do not have to implement logic to show/hide it

Form Fields

1. Provide form fields to let the user enter the following data:
    a. Cost of 1 L of gas ($/L):
        - the price of 1 litre of gas., example: $1.80 per liter (L)
    b. Gas car mileage (km/L)
        - distance a gas car can travel on 1 litre of gas, example: 10.2 km per litre (L)
    c. Cost of 1 kwH of electricity ($/kwH)
        - the price of 1 kwH of electricity. Example: $0.1235 per kwH
    d. Electric car mileage (km/kwH)
        - distance an electric car can travel on 1 kwH of electricity, example: 4.94 km per kwH
    e. Number of km driven per year. **Provide the user with 3 choices (15000, 25000, 40000)**

2. To make it easier to test your app, **prepopulate the form fields** with default values of your choice.

3. When the button is pressed, use the form field data to perform the necessary calculations and output the results to the results summary. **The formulas for the calculations are provided in the course webpage.**

4. **Do NOT implement form field validation.**

Styling

In your design, you must demonstrate:

- A variety of icons
- Layout in a horizontal row
- The overall design must be reasonably pretty with pleasing colors and typography. Text must be reasonably easy to read.
- Ensure that form fields are sized and styled in a way that makes it easy for the user to 1) see where the form field is; and 2) use their fingers to type in the field - remember that on a real mobile device, most people's fingers are large!

If you aren't sure what type of design to use, then you may refer to the expected output for an example design. If you are using the example design, then ensure you do some customization of it (such as changing the colors, white spacing, icons, etc) rather than just copying and pasting the design.

Implementation Guidelines

1. Code must be written in Javascript (not Typescript or any other language)
2. Usage of 3rd party UI frameworks is NOT permitted (example: React Paper, Material UI, etc)
3. You must use function based components, not class based components
4. Variables must be declared using let and const. No var declarations permitted.
5. Functions should be declared using arrow function syntax, example: const abc = () => {}
6. Implementation must follow techniques demonstrated in class.
7. In addition to the required functionality, learners are expected to use the coding conventions demonstrated in class, meaningful variable naming, and clearly organized code. Comments are helpful but not required.

END OF ASSESSMENT