Omphile Malakalaka

MAST part 1

St10444928

**To build a GUI in React Native, you need to:**

Install and initialize a React Native project.

Import necessary components from react-native such as View, Text, Button, etc.

Design your layout using these components.

Apply styles using Style Sheet.

Run and test your application on an emulator or device

**Overall Application Functionality**

This application is a simple React Native app that features a basic user interface with text input, buttons, and an image. The app has the following main components and behaviors:

1. **Header Section:**
   * Displays an image and a title at the top of the screen.
   * The image is a logo (from React Native’s website in this example).
   * The title text welcomes the user to the app.
2. **Input Section:**
   * Contains a text input field where users can type in some text.
   * Includes a button that, when pressed, triggers an alert message.
3. **Footer Section:**
   * Contains a button that, when pressed, shows another alert message.

View (header): This View acts as a container for the header section, aligning its child elements (the image and the title) using styles defined in styles.

header.

Image: Displays a small React Native logo. The source prop provides the URL for the logo image. The style prop applies styling to the image (e.g., size).

Text (title): Shows a welcome message to the user, styled with a larger font size and bold weight as defined in styles.

title.

View (inputContainer): This View contains the text input field and button, styled with margin below it.

TextInput: This component allows users to enter text. The value prop is bound to the text state variable, meaning it displays whatever text is currently in that state. The onChangeText prop updates the text state every time the user types.

TouchableOpacity: A button-like component that provides feedback when pressed. When pressed, it shows an alert with the message “Button pressed”. This component is styled with a background color and padding as defined in styles.button.

Text (buttonText): Inside the button, this text component displays the button label “Press Me”, styled to be white and centered.

View (footer): This View acts as a container for the footer section, which includes the button.

Button: A standard button component that shows the title “Submit”. When pressed, it triggers an alert with the message “Submitted”. The button’s appearance is handled by default styling provided by React Native.

Styling

The styles object uses Style Sheet create to define styles for the various components. Here’s how they affect the layout and appearance:

Container (container): Sets the layout to fill the entire screen, with padding and a white background.

Header (header): Centres the logo and title horizontally and adds space below it.

Logo (logo): Specifies the size of the logo image.

Title (title): Styles the title text to be large and bold.

Input Container (input Container): Adds space below the text input field.

Input (input): Styles the text input with height, border, and padding.

Button (button): Defines the background colour and padding for the button.

Button Text (button Text): Styles the text inside the button to be white and centred.

Footer (footer): Adds space above the footer section.

Summary

In this React Native application:

The header section provides a welcoming visual with a logo and title.

The input section allows users to type text and interact with a button that triggers an alert.

The footer section includes a button that also triggers an alert when pressed.

Styling ensures the application looks visually appealing and the layout is consistent.

The app is simple yet functional, showcasing how to structure a React Native application with basic UI components and interactivity.