Week 16 Documentation

To-Do List App Documentation:

Overview:

The To-Do List app allows users to add and delete tasks. Users can input tasks into a text field and manage them in a list.

Key Features:

- Input field to add new tasks.
- Button to add tasks to the list.
- Displaying tasks using a FlatList component.
- Button next to each task to delete it.

State Variables:

- `taskInput` (string): Holds the current text from the input field.
- `taskList` (array): Stores the list of tasks.

Functions:

1. **`addTask()`**:
    - Purpose: Adds a new task to `taskList`.
    - Details:

        - Checks if the input is not empty (no spaces).
        - If valid, appends the task to `taskList` and clears the input.

2. **`deleteTask(indexToDelete)`**:
    - Purpose: Deletes a task from `taskList` based on its index.
    - Details:
        - Creates a copy of the current `taskList`.
        - Removes the task at the specified index and updates the state.

UI Components:

- **TextInput**: Allows the user to input new tasks.
- **Button**: Trigger function to add tasks.
- **FlatList**: Displays tasks in a scrollable list.
- **Button (within FlatList)**: Deletes individual tasks from the list.

Style:

- **container**: Main container with padding and background color.
- **input**: Styling for the input field with a border and padding.

- **taskContainer**: Styling for each task item with row direction and spacing between text and delete button.

- **taskText**: Styling for task text with font size.

BMI Calculator App Documentation:

Overview:

The BMI Calculator app allows users to input their weight and height, calculates their BMI, and classifies it into categories like "Underweight", "Normal weight", "Overweight", or "Obese".

Key Features:

- Input fields for weight (kg) and height (cm).

- A button to calculate BMI.

- Displays BMI result and classification (category).

State Variables:

- `weight` (string): Holds the user's weight input.

- `height` (string): Holds the user's height input.

- `bmiResult` (float|null): Holds the calculated BMI value.

- `bmiCategory` (string): Holds the category based on the BMI value.

Functions:

1. **`calculateBMI()`**:
    - Purpose: Calculates the BMI based on the user's weight and height.
    - Details:
        - Converts weight to a number (kg) and height to meters.
        - Calculates BMI using the formula: BMI = weight (kg) / (height (m)²).
        - Rounds the result to two decimal places and updates the state.
        - Calls `determineCategory()` to classify the BMI.
2. **`determineCategory(bmiValue)`**:
    - Purpose: Determines the BMI category based on the value.
    - Details:
        - Classifies BMI into one of four categories: "Underweight", "Normal weight", "Overweight", or "Obese".
        - Updates the `bmiCategory` state based on the result.

UI Components:

- **TextInput** (for weight): Allows the user to input their weight in kilograms.

- **TextInput** (for height): Allows the user to input their height in centimeters.
- **Button**: Triggers BMI calculation.
- **Text**: Displays the calculated BMI and the corresponding category.
- **TouchableWithoutFeedback**: Dismisses the keyboard when tapping outside of the input fields.

Style:

- **container**: Flexbox layout to center the UI and add padding.
- **title**: Bold title at the top for "BMI Calculator".
- **input**: Input fields with a border, margin, and center-aligned text.
- **buttonWrapper**: Styling to center the "Calculate BMI" button.
- **resultBox**: Container to show the calculated BMI and category.
- **bmiText**: Styling for the BMI value display.
- **categoryText**: Styling for the BMI category, with color for visual distinction.

General Notes:

- **Error Handling**: Both apps ensure the user input is validated. For the BMI calculator, the app only calculates if both weight and height are

provided.

- **UI Considerations**: The BMI app uses `ScrollView` to ensure all components are visible on smaller screens and `TouchableWithoutFeedback` to dismiss the keyboard when the user taps outside input fields.