Assignment: 01 – Development of a Cross-Platform To-Do List Application

Course: Mobile Computing

Instructor: Eng. Rania El-Sayed

Student Name: Mariam Sherif

Date: February 27, 2024

1. Project Overview

The objective of this assignment was to design and develop a functional "To-Do List" mobile application using the **React Native** framework and **Expo** managed workflow. The application serves as a practical implementation of fundamental mobile development concepts, including user input handling, state persistence within a session, and optimized list rendering.

The application allows users to:

- Interface with a TextInput for goal definition.
- Commit entries to a dynamic list.
- View and scroll through entries via a FlatList.
- Remove specific entries through touch-based interactions.

2. Technical Implementation & Logic

The application's architecture is built upon React's functional component model, utilizing the useState hook for reactive data management.

2.1 State Management

Two primary states were defined to manage the data flow:

1. **Input State:** Captures and stores the current string value from the text input field.
   const [goalText, setGoalText] = useState<string>('');

2. **Goals Collection State:** An array of objects used to store the finalized list of goals, ensuring each item has a unique identifier for efficient rendering.
   const [goals, setGoals] = useState<{ text: string; id: string }[]>([]);

2.2 Core Functions

- **Addition Logic:** A handler function triggered by the "Add" button that validates the input, generates a unique ID (using Math.random() or a timestamp), and updates the goals array using the spread operator to maintain state immutability.
- **Deletion Logic:** Items are removed using the .filter() method. When a user presses an item, its unique ID is passed to the function, which returns a new array excluding the selected item.
- **Rendering Engine:** The FlatList component was implemented instead of a standard ScrollView to provide better memory management and performance for long lists.

3. UI/UX Design Specifications

The application features a modern, feminine aesthetic utilizing a "soft pink" color palette. The design focuses on high legibility and a friendly user interface.

3.1 Color Palette

Element	**Hex Code**	**Description**
Background	#FFF0F6	Soft pastel pink for low eye-strain.
Primary Accent	#FF69B4	Main color for buttons and highlights.
Item Background	#FFB6C1	Secondary pink for goal list cards.
Typography	#5A2A3A	Dark muted pink for optimal contrast.
Interactive Text	#FFFFFF	White text for primary action buttons.

3.2 Typography & Assets

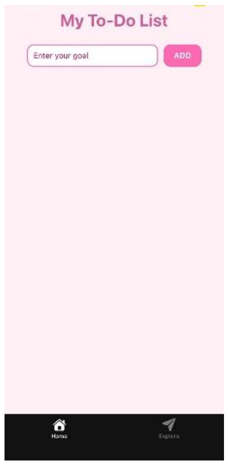
The application utilizes the **System Default Font** (San Francisco on iOS and Roboto on Android) to ensure native performance and consistent weight distribution.

- **Headers:** Bold System Font.
- **List Content:** Semi-bold System Font.

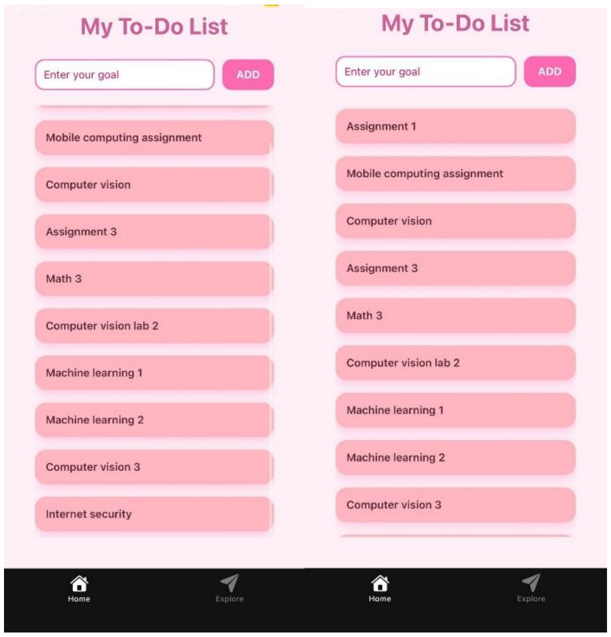
4. Evidence of Implementation

4.1 Screenshots

1. **Initial State:** Showing the clean UI with an empty list.



2. **Active Input:** Demonstrating the keyboard and text input interaction
3. **List View:** Showing multiple goals and the scrolling capability.



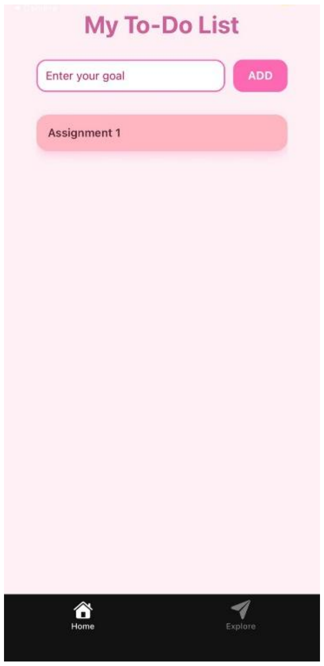
My To-Do List

Mobile computing assignment
Computer vision
Assignment 3
Math 3
Computer vision lab 2
Machine learning 1
Machine learning 2
Computer vision 3
Internet security

My To-Do List

Assignment 1
Mobile computing assignment
Computer vision
Assignment 3
Math 3
Computer vision lab 2
Machine learning 1
Machine learning 2
Computer vision 3

4. **Deletion Action:** A visual of the list after an item has been removed.



My To-Do List

Assignment 1

4.2 Project Metadata

- **Expo Snack Link:** [https://snack.expo.dev/@marriamsherrif56/mobile-computing-to-do-app]
- **GitHub Repository:** [https://github.com/Mariam56Elgazzar/mobile-computing-todo-app]
- **Video Demonstration:** [https://drive.google.com/file/d/1ONariHaA0keosWWUlfVLwWH-uENzn-ue/view?usp=sharing]

5. Conclusion

This project successfully demonstrates the integration of core React Native hooks and components. By implementing the "To-Do List" application, I have gained a deeper understanding of handling user events, managing component-level state, and styling mobile applications with StyleSheet. The final product fulfills all technical requirements while maintaining a professional and cohesive design language.