Experiment 5: Applying Navigation, Routing, and Gestures in Flutter App

Aim:
The aim of this experiment is to implement navigation, routing, and gestures within a Flutter application. By doing so, we intend to understand and apply the fundamental concepts associated with navigating between different screens, implementing a structured routing system, and incorporating user-friendly gestures for enhanced user interaction.

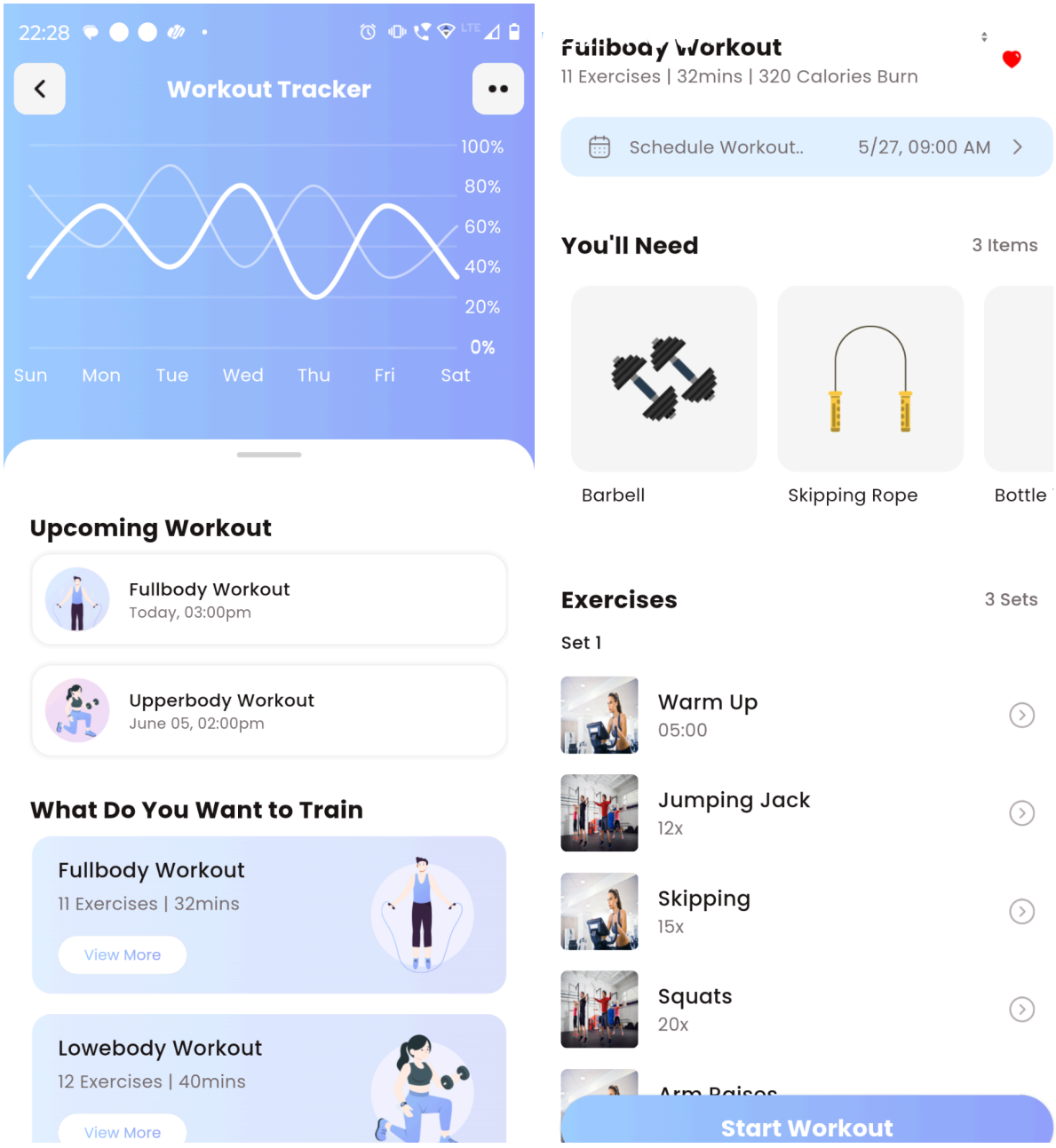
Theory:

1. Navigation and Routing:
   - *Navigation:* Navigation refers to the process of moving between different screens or pages within a mobile application. In Flutter, the `Navigator` class is used for managing a stack of `Route` objects. It allows for pushing and popping routes to navigate between different screens.
   - *Routing:* Routing involves defining and managing the paths or routes within an application. Flutter provides a `MaterialPageRoute` for standard page-based navigation. It's crucial to understand how to set up and manage routes for effective screen transitions.

2. Gesture Detection:
   - Flutter supports various gestures such as tapping, swiping, pinching, etc. GestureDetector is a versatile widget that can detect various gestures and trigger corresponding actions.
   - Gesture recognition enhances the user experience by providing a more interactive and intuitive interface. It involves handling user input through gestures like taps, swipes, or pinches to trigger specific actions in the application.

3. Implementation Steps:
   - Define routes using the `MaterialPageRoute` for different screens.
   - Implement navigation using the `Navigator` class for pushing and popping routes.
   - Utilize the `GestureDetector` widget to capture user gestures.
   - Associate specific actions with different gestures to enhance user interaction.

o/p:



Conclusion:

In conclusion, this experiment allowed us to gain practical experience in implementing navigation, routing, and gestures in a Flutter application. Navigating between screens and managing routes is essential for creating a structured and user-friendly app architecture. Additionally, incorporating gestures provides a more interactive and engaging user experience. The successful completion of this experiment equips us with the skills to create more dynamic and navigable Flutter applications, enhancing our proficiency in mobile app development.