Name:Sahil Motiramani Div:D15C Roll No.:35

Experiment:2

Aim: To design Flutter UI by including common widgets.

Theory:

Flutter is a UI toolkit by Google used for building natively compiled applications for mobile, web, and desktop using a single codebase. Flutter uses **Widgets** as its core building blocks.

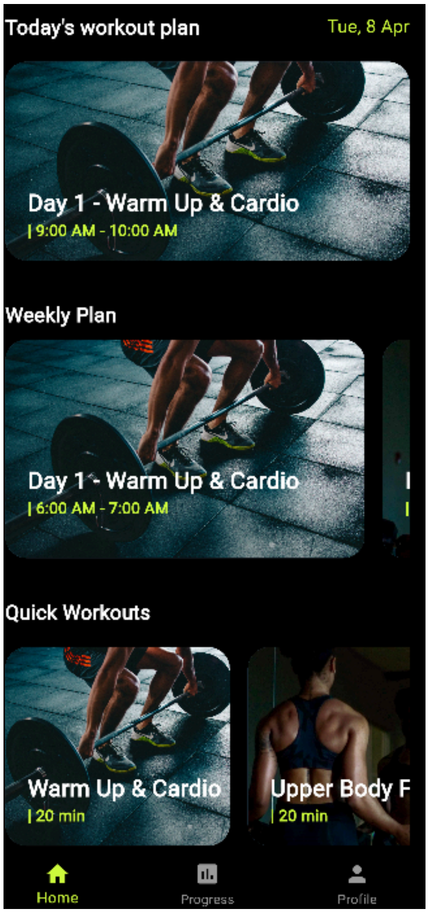
Widgets are basically everything in Flutter – from layout to buttons, texts, and containers.

Widgets in futter:

- **Scaffold**: Provides the basic screen layout (app bar, body, bottom nav bar).
- **Text**: Used to display text like headers, workout names, and timings.
- **Padding**: Adds spacing around widgets for better UI structure.
- **Column/Row**: Arranges widgets vertically/horizontally—used for aligning workouts and headers.
- **Container**: Wraps content with padding, margin, color, border radius; used for workout cards.
- **DecorationImage**: Used inside Container with BoxDecoration to show background images.
- **ListView.builder**: Dynamically builds scrollable workout lists like "Quick Workouts" or "Weekly Plan".
- **GestureDetector**: Wraps cards to handle taps and navigate to workout detail pages.
- **SizedBox**: Adds vertical spacing between widgets for clean layout.
- **BottomNavigationBar**: Enables navigation between tabs (Home, Progress, Profile).
- **DateTime**: Used to dynamically show today's date (Tue, 8 Apr) in the UI.
- **Stack**:
  Allows widgets to be layered on top of each other—used to overlay text (e.g., workout name, time) on background images.
- **Positioned**:
  Used within a Stack to place text or icons at specific positions over workout cards.

Name:Sahil Motiramani Div:D15C Roll No.:35

Output:



Today's workout plan

Tue, 8 Apr

Day 1 - Warm Up & Cardio

| 9:00 AM - 10:00 AM

Weekly Plan

Day 1 - Warm Up & Cardio

| 6:00 AM - 7:00 AM

Quick Workouts

Warm Up & Cardio

| 20 min

Upper Body F

| 20 min

Conclusion:

Using common Flutter widgets enables the creation of a responsive, user-friendly, and visually rich fitness application. By combining widgets like Scaffold, Container, Text, and BottomNavigationBar, the UI becomes intuitive and easy to navigate. These widgets promote modularity and enhance code reusability while maintaining aesthetic appeal and performance.