MAD and PWA Lab

Name: Vrushabh Ghuse   **Class: D15A**   **Roll no:22**

Aim: To design Flutter UI by including common widgets.

Theory:

Flutter widgets are the building blocks of the user interface, representing visual and interactive elements. Widgets can be categorized into two main types: Stateless and Stateful. Stateless widgets are immutable and do not store any internal state, while Stateful widgets can change and maintain their state over time.

Built-in and Custom Widgets:
Flutter comes with a rich set of built-in widgets that cover a wide range of UI elements and interactions. Additionally, developers can create their custom widgets to tailor the user interface to specific requirements. This section explores the anatomy of custom widgets and their integration into the Flutter framework.

Layout and Constraints:
Flutter's layout system is based on the concept of constraints, allowing widgets to adapt to various screen sizes and orientations. Understanding how widgets handle layout constraints is essential for creating responsive and adaptive user interfaces.

Animation with Widgets:
Flutter provides a robust animation framework that seamlessly integrates with widgets, enabling the creation of fluid and engaging user experiences. This section explores how animations can be incorporated into Flutter widgets to enhance the overall user interface.

Testing and Debugging Widgets:
As with any software development, testing and debugging are integral parts of the process. This section discusses strategies for testing and debugging Flutter widgets, ensuring the reliability and quality of the UI components.

Container:

The Container widget is a basic building block that can contain other widgets and is often used to define the dimensions, padding, margin, and decoration of a UI element.

Text:

The Text widget is used to display a paragraph or a line of text. It supports various styling options such as font size, color, and alignment.

ListView:

The ListView widget is used to create a scrollable list of widgets. It can display a large number of children efficiently.

Row and Column:

Row and Column widgets are used to arrange children in a horizontal or vertical line, respectively.

Stack:

The Stack widget allows you to overlay multiple widgets on top of each other. It's often used for complex layouts.

AppBar:

The AppBar widget represents the top app bar that usually contains the app's title, icons, and actions.

Code:
home_view.dart :

```dart
import 'package:dotted_dashed_line/dotted_dashed_line.dart';
import 'package:fitness/common_widget/round_button.dart';
import 'package:fitness/common_widget/workout_row.dart';
import 'package:fl_chart/fl_chart.dart';
import 'package:flutter/material.dart';
import 'package:simple_animation_progress_bar/simple_animation_progress_bar.dart';
import 'package:simple_circular_progress_bar/simple_circular_progress_bar.dart';
import '../../common/colo_extension.dart';
import 'activity_tracker_view.dart';
import 'finished_workout_view.dart';
import 'notification_view.dart';

class HomeView extends StatefulWidget {
  const HomeView({super.key});

  @override
  State<HomeView> createState() => _HomeViewState();
```

```dart
  }

  class _HomeViewState extends State<HomeView> {
    List lastWorkoutArr = [
      {
        "name": "Full Body Workout",
        "image": "assets/img/Workout1.png",
        "kcal": "180",
        "time": "20",
        "progress": 0.3
      },
      {
        "name": "Lower Body Workout",
        "image": "assets/img/Workout2.png",
        "kcal": "200",
        "time": "30",
        "progress": 0.4
      },
      {
        "name": "Ab Workout",
        "image": "assets/img/Workout3.png",
        "kcal": "300",
        "time": "40",
        "progress": 0.7
      },
    ];
    List<int> showingTooltipOnSpots = [21];

    List<FlSpot> get allSpots => const [
        FlSpot(0, 20),
        FlSpot(1, 25),
        FlSpot(2, 40),
        FlSpot(3, 50),
        FlSpot(4, 35),
        FlSpot(5, 40),
        FlSpot(6, 30),
        FlSpot(7, 20),
        FlSpot(8, 25),
        FlSpot(9, 40),
        FlSpot(10, 50),
        FlSpot(11, 35),
        FlSpot(12, 50),
        FlSpot(13, 60),
        FlSpot(14, 40),
        FlSpot(15, 50),
        FlSpot(16, 20),
        FlSpot(17, 25),
        FlSpot(18, 40),
        FlSpot(19, 50),
        FlSpot(20, 35),
        FlSpot(21, 80),
        FlSpot(22, 30),
        FlSpot(23, 20),
        FlSpot(24, 25),
        FlSpot(25, 40),
        FlSpot(26, 50),
        FlSpot(27, 35),
```

```dart
      FlSpot(28, 50),
      FlSpot(29, 60),
      FlSpot(30, 40)
    ];

List waterArr = [
  {"title": "6am - 8am", "subtitle": "600ml"},
  {"title": "9am - 11am", "subtitle": "500ml"},
  {"title": "11am - 2pm", "subtitle": "1000ml"},
  {"title": "2pm - 4pm", "subtitle": "700ml"},
  {"title": "4pm - now", "subtitle": "900ml"},
];

@override
Widget build(BuildContext context) {
  var media = MediaQuery.of(context).size;

  final lineBarsData = [
    LineChartBarData(
      showingIndicators: showingTooltipOnSpots,
      spots: allSpots,
      isCurved: false,
      barWidth: 3,
      belowBarData: BarAreaData(
        show: true,
        gradient: LinearGradient(colors: [
          TColor.primaryColor2.withOpacity(0.4),
          TColor.primaryColor1.withOpacity(0.1),
        ], begin: Alignment.topCenter, end: Alignment.bottomCenter),
      ),
      dotData: FlDotData(show: false),
      gradient: LinearGradient(
        colors: TColor.primaryG,
      ),
    ),
  ];

  final tooltipsOnBar = lineBarsData[0];

  return Scaffold(
    backgroundColor: TColor.white,
    body: SingleChildScrollView(
      child: SafeArea(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 15),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text(
                        "Welcome Back,",
                        style: TextStyle(color: TColor.gray, fontSize: 12),
```

```dart
      ),
      Text(
        "Vrushabh Ghuse",
        style: TextStyle(
            color: TColor.black,
            fontSize: 20,
            fontWeight: FontWeight.w700),
      ),
    ],
  ),
  IconButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => const NotificationView(),
          ),
        );
      },
      icon: Image.asset(
        "assets/img/notification_active.png",
        width: 25,
        height: 25,
        fit: BoxFit.fitHeight,
      ))
  ],
),
SizedBox(
  height: media.width * 0.05,
),
Container(
  height: media.width * 0.4,
  decoration: BoxDecoration(
      gradient: LinearGradient(colors: TColor.primaryG),
      borderRadius: BorderRadius.circular(media.width * 0.075)),
  child: Stack(alignment: Alignment.center, children: [
    Image.asset(
      "assets/img/bg_dots.png",
      height: media.width * 0.4,
      width: double.maxFinite,
      fit: BoxFit.fitHeight,
    ),
    Padding(
      padding: const EdgeInsets.symmetric(
          vertical: 25, horizontal: 25),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                "BMI (Body Mass Index)",
                style: TextStyle(
                    color: TColor.white,
                    fontSize: 14,
```

```dart
                fontWeight: FontWeight.w700),
          ),
          Text(
            "You have a normal weight",
            style: TextStyle(
                color: TColor.white.withOpacity(0.7),
                fontSize: 12),
          ),
          SizedBox(
            height: media.width * 0.05,
          ),
          SizedBox(
              width: 120,
              height: 35,
              child: RoundButton(
                  title: "View More",
                  type: RoundButtonType.bgSGradient,
                  fontSize: 12,
                  fontWeight: FontWeight.w400,
                  onPressed: () {}))
        ],
      ),
      AspectRatio(
        aspectRatio: 1,
        child: PieChart(
          PieChartData(
            pieTouchData: PieTouchData(
              touchCallback:
                  (FlTouchEvent event, pieTouchResponse) {},
            ),
            startDegreeOffset: 250,
            borderData: FlBorderData(
              show: false,
            ),
            sectionsSpace: 1,
            centerSpaceRadius: 0,
            sections: showingSections(),
          ),
        ),
      ),
    ],
  ),
)
]),
),
SizedBox(
  height: media.width * 0.05,
),
Container(
  padding:
      const EdgeInsets.symmetric(vertical: 15, horizontal: 15),
  decoration: BoxDecoration(
    color: TColor.primaryColor2.withOpacity(0.3),
    borderRadius: BorderRadius.circular(15),
  ),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```dart
              children: [
                Text(
                  "Today Target",
                  style: TextStyle(
                      color: TColor.black,
                      fontSize: 14,
                      fontWeight: FontWeight.w700),
                ),
                SizedBox(
                  width: 70,
                  height: 25,
                  child: RoundButton(
                    title: "Check",
                    type: RoundButtonType.bgGradient,
                    fontSize: 12,
                    fontWeight: FontWeight.w400,
                    onPressed: () {
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                          builder: (context) =>
                              const ActivityTrackerView(),
                        ),
                      );
                    },
                  ),
                )
              ],
            ),
          ),
          SizedBox(
            height: media.width * 0.05,
          ),
          Text(
            "Activity Status",
            style: TextStyle(
                color: TColor.black,
                fontSize: 16,
                fontWeight: FontWeight.w700),
          ),
          SizedBox(
            height: media.width * 0.02,
          ),
          ClipRRect(
            borderRadius: BorderRadius.circular(25),
            child: Container(
              height: media.width * 0.4,
              width: double.maxFinite,
              decoration: BoxDecoration(
                color: TColor.primaryColor2.withOpacity(0.3),
                borderRadius: BorderRadius.circular(25),
              ),
              child: Stack(
                alignment: Alignment.topLeft,
                children: [
                  Padding(
                    padding: const EdgeInsets.symmetric(
```

```dart
                    vertical: 20, horizontal: 20),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.start,
                  mainAxisSize: MainAxisSize.min,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      "Heart Rate",
                      style: TextStyle(
                          color: TColor.black,
                          fontSize: 16,
                          fontWeight: FontWeight.w700),
                    ),
                    ShaderMask(
                      blendMode: BlendMode.srcIn,
                      shaderCallback: (bounds) {
                        return LinearGradient(
                              colors: TColor.primaryG,
                              begin: Alignment.centerLeft,
                              end: Alignment.centerRight)
                            .createShader(Rect.fromLTRB(
                              0, 0, bounds.width, bounds.height));
                      },
                      child: Text(
                        "78 BPM",
                        style: TextStyle(
                            color: TColor.white.withOpacity(0.7),
                            fontWeight: FontWeight.w700,
                            fontSize: 18),
                      ),
                    ),
                  ],
                ),
              ),
              LineChart(
                LineChartData(
                  showingTooltipIndicators:
                      showingTooltipOnSpots.map((index) {
                    return ShowingTooltipIndicators([
                      LineBarSpot(
                        tooltipsOnBar,
                        lineBarsData.indexOf(tooltipsOnBar),
                        tooltipsOnBar.spots[index],
                      ),
                    ]);
                  }).toList(),
                  lineTouchData: LineTouchData(
                    enabled: true,
                    handleBuiltInTouches: false,
                    touchCallback: (FlTouchEvent event,
                        LineTouchResponse? response) {
                      if (response == null ||
                          response.lineBarSpots == null) {
                        return;
                      }
                      if (event is FlTapUpEvent) {
                        final spotIndex =
```

```dart
              response.lineBarSpots!.first.spotIndex;
          showingTooltipOnSpots.clear();
          setState(() {
            showingTooltipOnSpots.add(spotIndex);
          });
        }
      },
      mouseCursorResolver: (FlTouchEvent event,
          LineTouchResponse? response) {
        if (response == null ||
            response.lineBarSpots == null) {
          return SystemMouseCursors.basic;
        }
        return SystemMouseCursors.click;
      },
      getTouchedSpotIndicator:
          (LineChartBarData barData,
              List<int> spotIndexes) {
        return spotIndexes.map((index) {
          return TouchedSpotIndicatorData(
            FlLine(
              color: Colors.red,
            ),
            FlDotData(
              show: true,
              getDotPainter:
                  (spot, percent, barData, index) =>
                      FlDotCirclePainter(
                radius: 3,
                color: Colors.white,
                strokeWidth: 3,
                strokeColor: TColor.secondaryColor1,
              ),
            ),
          );
        }).toList();
      },
      touchTooltipData: LineTouchTooltipData(
        tooltipBgColor: TColor.secondaryColor1,
        tooltipRoundedRadius: 20,
        getTooltipItems:
            (List<LineBarSpot> lineBarsSpot) {
          return lineBarsSpot.map((lineBarSpot) {
            return LineTooltipItem(
              "${lineBarSpot.x.toInt()} mins ago",
              const TextStyle(
                color: Colors.white,
                fontSize: 10,
                fontWeight: FontWeight.bold,
              ),
            );
          }).toList();
        },
      ),
    ),
    lineBarsData: lineBarsData,
    minY: 0,
```

```dart
                      maxY: 130,
                      titlesData: FlTitlesData(
                        show: false,
                      ),
                      gridData: FlGridData(show: false),
                      borderData: FlBorderData(
                        show: true,
                        border: Border.all(
                          color: Colors.transparent,
                        ),
                      ),
                    ),
                  )
                ],
              ),
            ),
          ),
          SizedBox(
            height: media.width * 0.05,
          ),
          Row(
            children: [
              Expanded(
                child: Container(
                  height: media.width * 0.95,
                  padding: const EdgeInsets.symmetric(
                      vertical: 25, horizontal: 20),
                  decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(25),
                    boxShadow: const [
                      BoxShadow(color: Colors.black12, blurRadius: 2)
                    ]),
                  child: Row(
                    children: [
                      SimpleAnimationProgressBar(
                        height: media.width * 0.85,
                        width: media.width * 0.07,
                        backgroundColor: Colors.grey.shade100,
                        foregrondColor: Colors.purple,
                        ratio: 0.5,
                        direction: Axis.vertical,
                        curve: Curves.fastLinearToSlowEaseIn,
                        duration: const Duration(seconds: 3),
                        borderRadius: BorderRadius.circular(15),
                        gradientColor: LinearGradient(
                            colors: TColor.primaryG,
                            begin: Alignment.bottomCenter,
                            end: Alignment.topCenter),
                      ),
                      const SizedBox(
                        width: 10,
                      ),
                      Expanded(
                          child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
```

```dart
Text(
  "Water Intake",
  style: TextStyle(
      color: TColor.black,
      fontSize: 12,
      fontWeight: FontWeight.w700),
),
ShaderMask(
  blendMode: BlendMode.srcIn,
  shaderCallback: (bounds) {
    return LinearGradient(
          colors: TColor.primaryG,
          begin: Alignment.centerLeft,
          end: Alignment.centerRight)
        .createShader(Rect.fromLTRB(
          0, 0, bounds.width, bounds.height));
  },
  child: Text(
    "4 Liters",
    style: TextStyle(
        color: TColor.white.withOpacity(0.7),
        fontWeight: FontWeight.w700,
        fontSize: 14),
  ),
),
const SizedBox(
  height: 10,
),
Text(
  "Real time updates",
  style: TextStyle(
    color: TColor.gray,
    fontSize: 12,
  ),
),
Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: waterArr.map((wObj) {
    var isLast = wObj == waterArr.last;
    return Row(
      crossAxisAlignment:
          CrossAxisAlignment.start,
      children: [
        Column(
          mainAxisAlignment:
              MainAxisAlignment.start,
          crossAxisAlignment:
              CrossAxisAlignment.center,
          children: [
            Container(
              margin:
                  const EdgeInsets.symmetric(
                      vertical: 4),
              width: 10,
              height: 10,
              decoration: BoxDecoration(
                color: TColor.secondaryColor1
```

```dart
              .withOpacity(0.5),
          borderRadius:
              BorderRadius.circular(5),
        ),
      ),
      if (!isLast)
        DottedDashedLine(
            height: media.width * 0.078,
            width: 0,
            dashColor: TColor
                .secondaryColor1
                .withOpacity(0.5),
            axis: Axis.vertical)
    ],
  ),
  const SizedBox(
    width: 10,
  ),
  Column(
    mainAxisAlignment:
        MainAxisAlignment.start,
    crossAxisAlignment:
        CrossAxisAlignment.start,
    children: [
      Text(
        wObj["title"].toString(),
        style: TextStyle(
          color: TColor.gray,
          fontSize: 10,
        ),
      ),
      ShaderMask(
        blendMode: BlendMode.srcIn,
        shaderCallback: (bounds) {
          return LinearGradient(
                  colors:
                      TColor.secondaryG,
                  begin: Alignment
                      .centerLeft,
                  end: Alignment
                      .centerRight)
              .createShader(Rect.fromLTRB(
                  0,
                  0,
                  bounds.width,
                  bounds.height));
        },
        child: Text(
          wObj["subtitle"].toString(),
          style: TextStyle(
              color: TColor.white
                  .withOpacity(0.7),
              fontSize: 12),
        ),
      ),
    ],
  )
```

```dart
                                ],
                              );
                            }).toList(),
                          )
                        ],
                      ))
                    ],
                  ),
                ),
              ),
              SizedBox(
                width: media.width * 0.05,
              ),
              Expanded(
                  child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                mainAxisSize: MainAxisSize.min,
                children: [
                  Container(
                    width: double.maxFinite,
                    height: media.width * 0.45,
                    padding: const EdgeInsets.symmetric(
                        vertical: 25, horizontal: 20),
                    decoration: BoxDecoration(
                      color: Colors.white,
                      borderRadius: BorderRadius.circular(25),
                      boxShadow: const [
                        BoxShadow(color: Colors.black12, blurRadius: 2)
                      ]),
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Text(
                          "Sleep",
                          style: TextStyle(
                              color: TColor.black,
                              fontSize: 12,
                              fontWeight: FontWeight.w700),
                        ),
                        ShaderMask(
                          blendMode: BlendMode.srcIn,
                          shaderCallback: (bounds) {
                            return LinearGradient(
                                    colors: TColor.primaryG,
                                    begin: Alignment.centerLeft,
                                    end: Alignment.centerRight)
                                .createShader(Rect.fromLTRB(
                                    0, 0, bounds.width, bounds.height));
                          },
                          child: Text(
                            "8h 20m",
                            style: TextStyle(
                                color: TColor.white.withOpacity(0.7),
                                fontWeight: FontWeight.w700,
                                fontSize: 14),
                          ),
                        ),
```

```dart
      const Spacer(),
      Image.asset("assets/img/sleep_grap.png",
          width: double.maxFinite,
          fit: BoxFit.fitWidth)
    ]),
),
SizedBox(
 height: media.width * 0.05,
),
Container(
 width: double.maxFinite,
 height: media.width * 0.45,
 padding: const EdgeInsets.symmetric(
     vertical: 25, horizontal: 20),
 decoration: BoxDecoration(
     color: Colors.white,
     borderRadius: BorderRadius.circular(25),
     boxShadow: const [
       BoxShadow(color: Colors.black12, blurRadius: 2)
     ]),
 child: Column(
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
      Text(
        "Calories",
        style: TextStyle(
            color: TColor.black,
            fontSize: 12,
            fontWeight: FontWeight.w700),
      ),
      ShaderMask(
        blendMode: BlendMode.srcIn,
        shaderCallback: (bounds) {
         return LinearGradient(
               colors: TColor.primaryG,
               begin: Alignment.centerLeft,
               end: Alignment.centerRight)
             .createShader(Rect.fromLTRB(
               0, 0, bounds.width, bounds.height));
        },
        child: Text(
         "760 kCal",
         style: TextStyle(
             color: TColor.white.withOpacity(0.7),
             fontWeight: FontWeight.w700,
             fontSize: 14),
        ),
      ),
      const Spacer(),
      Container(
        alignment: Alignment.center,
        child: SizedBox(
         width: media.width * 0.2,
         height: media.width * 0.2,
         child: Stack(
             alignment: Alignment.center,
             children: [
```

```
                    Container(
                      width: media.width * 0.15,
                      height: media.width * 0.15,
                      alignment: Alignment.center,
                      decoration: BoxDecoration(
                        gradient: LinearGradient(
                            colors: TColor.primaryG),
                        borderRadius: BorderRadius.circular(
                            media.width * 0.075),
                      ),
                      child: FittedBox(
                        child: Text(
                          "230kCal\nleft",
                          textAlign: TextAlign.center,
                          style: TextStyle(
                              color: TColor.white,
                              fontSize: 11),
                        ),
                      ),
                    ),
                    SimpleCircularProgressBar(
                      progressStrokeWidth: 10,
                      backStrokeWidth: 10,
                      progressColors: TColor.primaryG,
                      backColor: Colors.grey.shade100,
                      valueNotifier: ValueNotifier(50),
                      startAngle: -180,
                    ),
                  ],
                ),
              ),
            )
          ]),
        ),
      ],
    ),
  ))
],
),
SizedBox(
  height: media.width * 0.1,
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Text(
      "Workout Progress",
      style: TextStyle(
          color: TColor.black,
          fontSize: 16,
          fontWeight: FontWeight.w700),
    ),
    Container(
      height: 30,
      padding: const EdgeInsets.symmetric(horizontal: 8),
      decoration: BoxDecoration(
        gradient: LinearGradient(colors: TColor.primaryG),
        borderRadius: BorderRadius.circular(15),
```

```dart
          ),
        child: DropdownButtonHideUnderline(
          child: DropdownButton(
            items: ["Weekly", "Monthly"]
                .map((name) => DropdownMenuItem(
                      value: name,
                      child: Text(
                        name,
                        style: TextStyle(
                            color: TColor.gray, fontSize: 14),
                      ),
                    ))
                .toList(),
            onChanged: (value) {},
            icon: Icon(Icons.expand_more, color: TColor.white),
            hint: Text(
              "Weekly",
              textAlign: TextAlign.center,
              style:
                  TextStyle(color: TColor.white, fontSize: 12),
            ),
          ),
        )),
  ],
),
SizedBox(
  height: media.width * 0.05,
),
Container(
    padding: const EdgeInsets.only(left: 15),
    height: media.width * 0.5,
    width: double.maxFinite,
    child: LineChart(
      LineChartData(
        showingTooltipIndicators:
            showingTooltipOnSpots.map((index) {
          return ShowingTooltipIndicators([
            LineBarSpot(
              tooltipsOnBar,
              lineBarsData.indexOf(tooltipsOnBar),
              tooltipsOnBar.spots[index],
            ),
          ]);
        }).toList(),
        lineTouchData: LineTouchData(
          enabled: true,
          handleBuiltInTouches: false,
          touchCallback: (FlTouchEvent event,
              LineTouchResponse? response) {
            if (response == null ||
                response.lineBarSpots == null) {
              return;
            }
            if (event is FlTapUpEvent) {
              final spotIndex =
                  response.lineBarSpots!.first.spotIndex;
              showingTooltipOnSpots.clear();
```

```
    setState(() {
      showingTooltipOnSpots.add(spotIndex);
    });
  }
},
mouseCursorResolver: (FlTouchEvent event,
  LineTouchResponse? response) {
 if (response == null ||
    response.lineBarSpots == null) {
   return SystemMouseCursors.basic;
 }
 return SystemMouseCursors.click;
},
getTouchedSpotIndicator: (LineChartBarData barData,
  List<int> spotIndexes) {
 return spotIndexes.map((index) {
   return TouchedSpotIndicatorData(
    FlLine(
     color: Colors.transparent,
    ),
    FlDotData(
     show: true,
     getDotPainter:
       (spot, percent, barData, index) =>
         FlDotCirclePainter(
      radius: 3,
      color: Colors.white,
      strokeWidth: 3,
      strokeColor: TColor.secondaryColor1,
     ),
    ),
   );
  }).toList();
},
touchTooltipData: LineTouchTooltipData(
 tooltipBgColor: TColor.secondaryColor1,
 tooltipRoundedRadius: 20,
 getTooltipItems: (List<LineBarSpot> lineBarsSpot) {
  return lineBarsSpot.map((lineBarSpot) {
   return LineTooltipItem(
    "${lineBarSpot.x.toInt()} mins ago",
    const TextStyle(
     color: Colors.white,
     fontSize: 10,
     fontWeight: FontWeight.bold,
    ),
   );
  }).toList();
 },
),
),
lineBarsData: lineBarsData1,
minY: -0.5,
maxY: 110,
titlesData: FlTitlesData(
 show: true,
 leftTitles: AxisTitles(),
```

```dart
              topTitles: AxisTitles(),
              bottomTitles: AxisTitles(
                sideTitles: bottomTitles,
              ),
              rightTitles: AxisTitles(
                sideTitles: rightTitles,
              )),
          gridData: FlGridData(
            show: true,
            drawHorizontalLine: true,
            horizontalInterval: 25,
            drawVerticalLine: false,
            getDrawingHorizontalLine: (value) {
              return FlLine(
                color: TColor.gray.withOpacity(0.15),
                strokeWidth: 2,
              );
            },
          ),
          borderData: FlBorderData(
            show: true,
            border: Border.all(
              color: Colors.transparent,
            ),
          ),
        ),
      )),
SizedBox(
  height: media.width * 0.05,
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Text(
      "Latest Workout",
      style: TextStyle(
          color: TColor.black,
          fontSize: 16,
          fontWeight: FontWeight.w700),
    ),
    TextButton(
      onPressed: () {},
      child: Text(
        "See More",
        style: TextStyle(
            color: TColor.gray,
            fontSize: 14,
            fontWeight: FontWeight.w700),
      ),
    )
  ],
),
ListView.builder(
    padding: EdgeInsets.zero,
    physics: const NeverScrollableScrollPhysics(),
    shrinkWrap: true,
    itemCount: lastWorkoutArr.length,
```

```dart
          itemBuilder: (context, index) {
            var wObj = lastWorkoutArr[index] as Map? ?? {};
            return InkWell(
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
                          const FinishedWorkoutView(),
                    ),
                  );
                },
                child: WorkoutRow(wObj: wObj));
          }),
      SizedBox(
        height: media.width * 0.1,
      ),
    ],
  ),
 ),
 ),
 ),
 );
}

List<PieChartSectionData> showingSections() {
  return List.generate(
    2,
    (i) {
      var color0 = TColor.secondaryColor1;

      switch (i) {
        case 0:
          return PieChartSectionData(
            color: color0,
            value: 33,
            title: '',
            radius: 55,
            titlePositionPercentageOffset: 0.55,
            badgeWidget: const Text(
              "20,1",
              style: TextStyle(
                  color: Colors.white,
                  fontSize: 12,
                  fontWeight: FontWeight.w700),
            ));
        case 1:
          return PieChartSectionData(
            color: Colors.white,
            value: 75,
            title: '',
            radius: 45,
            titlePositionPercentageOffset: 0.55,
          );

        default:
          throw Error();
```

```dart
        }
      },
    );
  }

  LineTouchData get lineTouchData1 => LineTouchData(
      handleBuiltInTouches: true,
      touchTooltipData: LineTouchTooltipData(
        tooltipBgColor: Colors.blueGrey.withOpacity(0.8),
      ),
    );

  List<LineChartBarData> get lineBarsData1 => [
      lineChartBarData1_1,
      lineChartBarData1_2,
    ];

  LineChartBarData get lineChartBarData1_1 => LineChartBarData(
      isCurved: true,
      gradient: LinearGradient(colors: [
        TColor.primaryColor2.withOpacity(0.5),
        TColor.primaryColor1.withOpacity(0.5),
      ]),
      barWidth: 4,
      isStrokeCapRound: true,
      dotData: FlDotData(show: false),
      belowBarData: BarAreaData(show: false),
      spots: const [
        FlSpot(1, 35),
        FlSpot(2, 70),
        FlSpot(3, 40),
        FlSpot(4, 80),
        FlSpot(5, 25),
        FlSpot(6, 70),
        FlSpot(7, 35),
      ],
    );

  LineChartBarData get lineChartBarData1_2 => LineChartBarData(
      isCurved: true,
      gradient: LinearGradient(colors: [
        TColor.secondaryColor2.withOpacity(0.5),
        TColor.secondaryColor1.withOpacity(0.5),
      ]),
      barWidth: 2,
      isStrokeCapRound: true,
      dotData: FlDotData(show: false),
      belowBarData: BarAreaData(
        show: false,
      ),
      spots: const [
        FlSpot(1, 80),
        FlSpot(2, 50),
        FlSpot(3, 90),
        FlSpot(4, 40),
        FlSpot(5, 80),
        FlSpot(6, 35),
```

```dart
        FlSpot(7, 60),
      ],
    );

SideTitles get rightTitles => SideTitles(
      getTitlesWidget: rightTitleWidgets,
      showTitles: true,
      interval: 20,
      reservedSize: 40,
    );

Widget rightTitleWidgets(double value, TitleMeta meta) {
  String text;
  switch (value.toInt()) {
    case 0:
      text = '0%';
      break;
    case 20:
      text = '20%';
      break;
    case 40:
      text = '40%';
      break;
    case 60:
      text = '60%';
      break;
    case 80:
      text = '80%';
      break;
    case 100:
      text = '100%';
      break;
    default:
      return Container();
  }

  return Text(text,
      style: TextStyle(
        color: TColor.gray,
        fontSize: 12,
      ),
      textAlign: TextAlign.center);
}

SideTitles get bottomTitles => SideTitles(
      showTitles: true,
      reservedSize: 32,
      interval: 1,
      getTitlesWidget: bottomTitleWidgets,
    );

Widget bottomTitleWidgets(double value, TitleMeta meta) {
  var style = TextStyle(
    color: TColor.gray,
    fontSize: 12,
  );
  Widget text;
```

```dart
      switch (value.toInt()) {
        case 1:
          text = Text('Sun', style: style);
          break;
        case 2:
          text = Text('Mon', style: style);
          break;
        case 3:
          text = Text('Tue', style: style);
          break;
        case 4:
          text = Text('Wed', style: style);
          break;
        case 5:
          text = Text('Thu', style: style);
          break;
        case 6:
          text = Text('Fri', style: style);
          break;
        case 7:
          text = Text('Sat', style: style);
          break;
        default:
          text = const Text('');
          break;
      }

      return SideTitleWidget(
        axisSide: meta.axisSide,
        space: 10,
        child: text,
      );
    }
}
```

Output:
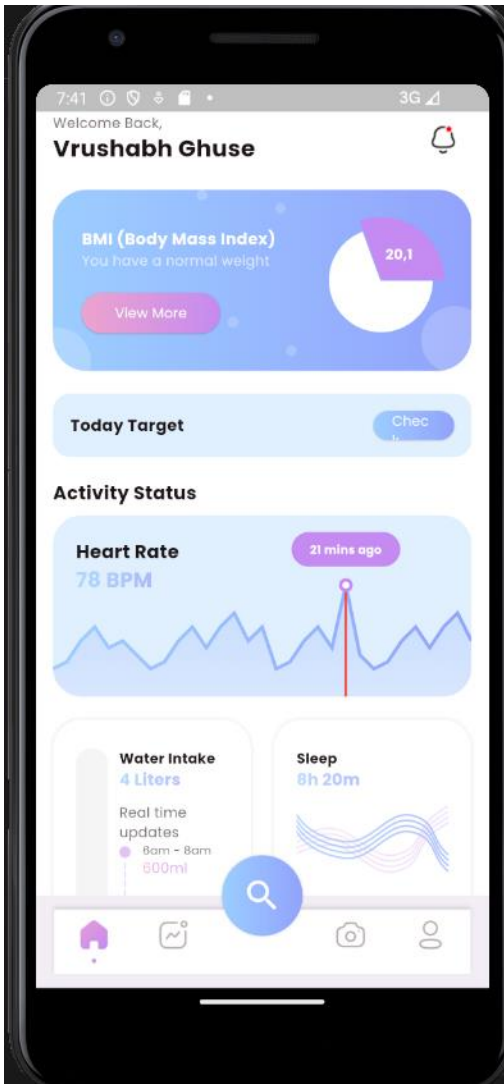
Conclusion:

Flutter's widget architecture offers great flexibility for building complex UIs.
Understanding key widgets and concepts is essential for effective Flutter development.