Aim -
To apply navigation, routing, and gestures in Flutter App.

Theory -

Flutter Gestures
Gestures are an interesting feature in Flutter that allows us to interact with the mobile app (or any touch-based device). Generally, gestures define any physical action or movement of a user in the intention of specific control of the mobile device. Some examples of gestures are:
When the mobile screen is locked, you slide your finger across the screen to unlock it.
Tapping a button on your mobile screen, and
Tapping and holding an app icon on a touch-based device to drag it across screens.
We use all these gestures in everyday life to interact with our phones or touch-based device.
Flutter divides the gesture system into two different layers, which are given below:

1. Pointers
2. Gestures

Pointers
Pointers are the first layer that represents the raw data about user interaction. It has events describing the location and movement of pointers such as touches, mice, and style across the screens. Flutter does not provide any mechanism to cancel or stop the pointer events from being dispatched further. Flutter provides a Listener widget to listen to the pointer events directly from the widgets layer. The pointer events are categorized into mainly four types:
PointerDownEvents
PointerMoveEvents
PointerUpEvents
PointerCancelEvents

Gestures
It is the second layer that represents semantic actions such as tap, drag, and scale, which are recognized from multiple individual pointer events. It is also able to dispatch multiple events corresponding to gesture lifecycle like drag start, drag update, and drag end.

1. Tap
2. Double Tap
3. Drag
4. Horizontal Drag
5. Vertical Drag
6. Long Press
7. Pan
8. Pinch

Output -

```
import 'package:flutter/material.dart';

class Report extends StatefulWidget {
  const Report({Key? key}) : super(key: key);

  @override
  State<Report> createState() =>
_ReportState();
}

class _ReportState extends State<Report> {
  int _currentIndex = 2;
  List navbar = ['home', 'discover', 'report',
'settings'];
  var date = [22, 23, 24, 25, 26, 27, 28];
  List weekday = ['S', 'M', 'T', 'W', 'T', 'F', 'S'];
  double _currentWeight = 70,
      lightestWeight = 65,
      heaviestWeight = 75,
      _currentHeightIN = 11;
  int _currentHeightFT = 5;

  Widget EditPhysicalDetails() {
    var _newWeight = TextEditingController();
    var _newHeightFT =
TextEditingController(),
        _newHeightIN =
TextEditingController();
    return SimpleDialog(
      contentPadding: EdgeInsets.all(15),
      children: [
        // Edit Weight
        Text(
          'Weight',
          style: TextStyle(
            fontSize: 25,
          ),
        ),
        SizedBox(
          height: 5,
        ),
        TextField(
          keyboardType:
TextInputType.number,
          controller: _newWeight,
          decoration: InputDecoration(
            hintText: "50.0",
          ),
        ),
        // Edit Height
        SizedBox(
          height: 20,
        ),
        Text(
          'Height',
          style: TextStyle(
            fontSize: 25,
          ),
        ),
        SizedBox(
          height: 5,
        ),
        Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
            TextField(
              keyboardType:
TextInputType.number,
              controller: _newHeightFT,
              decoration: InputDecoration(
                hintText: "5 FT",
                constraints: BoxConstraints(
                  maxWidth:
MediaQuery.of(context).size.width / 3,
                ),
              ),
            ),
            TextField(
              keyboardType:
TextInputType.number,
              controller: _newHeightIN,
              decoration: InputDecoration(
                hintText: "11.0 FT",
                constraints: BoxConstraints(
                  maxWidth:
MediaQuery.of(context).size.width / 3,
                ),
              ),
            ),
          ],
        ),
        SizedBox(
          height: 15,
        ),
        Row(
          mainAxisAlignment:
MainAxisAlignment.end,
          children: [
            TextButton(
              onPressed: () {
                setState(() {
                  Navigator.of(context).pop();
                });
              },
              child: Text(
                'Cancel',
                style: TextStyle(
                  color: Colors.blue,
                  fontSize: 18,
                ),
              ),
            ),
            TextButton(
              onPressed: () {
                setState(() {
                  if (_newWeight.text != "") {
                    _currentWeight =
double.parse(_newWeight.text);
                    if (heaviestWeight <
_currentWeight)
                      heaviestWeight =
_currentWeight;
                    else if (lightestWeight >
_currentWeight)
                      lightestWeight =
_currentWeight;
                  }
                  if (_newHeightFT.text != "")
                    _currentHeightFT =
int.parse(_newHeightFT.text);
                  if (_newHeightIN.text != "")
                    _currentHeightIN =
double.parse(_newHeightIN.text);

                  Navigator.of(context).pop();
                });
              },
              child: Text(
                'Save',
                style: TextStyle(
                  color: Colors.blue,
                  fontSize: 18,
                ),
              ),
```

```
            ),
          ],
        ),
      ],
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[300],
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Text(
          'Report',
          style: TextStyle(
            fontSize: 23,
            color: Colors.black,
          ),
        ),
      ),
      body: SingleChildScrollView(
        child: Column(
          children: [
            Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.only(
                    bottomLeft: Radius.circular(20),
                    bottomRight:
Radius.circular(20)),
                color: Colors.white,
              ),
              child: Column(
                children: [
                  SizedBox(
                    height: 10,
                  ),
                  Row(
                    mainAxisAlignment:
MainAxisAlignment.spaceAround,
                    children: [
                      Column(
                        children: [
                          Text(
                            '15',
                            style: TextStyle(
                              color: Colors.blue,
                              fontSize: 25,
                              letterSpacing: 1.2,
               fontWeight:
FontWeight.w600,
              ),
             ),
             SizedBox(
              height: 10,
             ),
             Text(
              'WORKOUTS',
              style: TextStyle(
               color: Colors.grey[600],
               fontSize: 14,
               letterSpacing: 1,
              ),
             )
            ],
           ),
           Column(
            children: [
             Text(
              '3944',
              style: TextStyle(
               color: Colors.blue,
               fontSize: 25,
               letterSpacing: 1.2,
               fontWeight:
FontWeight.w600,
              ),
             ),
             SizedBox(
              height: 10,
             ),
             Text(
              'KCAL',
              style: TextStyle(
               color: Colors.grey[600],
               fontSize: 14,
               letterSpacing: 1,
              ),
             )
            ],
           ),
           Column(
            children: [
             Text(
              '187',
              style: TextStyle(
               color: Colors.blue,
               fontSize: 25,
               letterSpacing: 1.2,
```

```
               fontWeight:
FontWeight.w600,
              ),
             ),
             SizedBox(
              height: 10,
             ),
             Text(
              'MINUTES',
              style: TextStyle(
               color: Colors.grey[600],
               fontSize: 14,
               letterSpacing: 1,
              ),
             )
            ],
           )
          ],
         ),
         SizedBox(
          height: 20,
         ),
        ],
       ),
      ),
      SizedBox(
       height: 10,
      ),
      Container(
       padding:
EdgeInsets.symmetric(horizontal: 15,
vertical: 20),
       decoration: BoxDecoration(
        borderRadius:
BorderRadius.circular(10),
        color: Colors.white,
       ),
       child: Column(
        children: [
         Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
           Text(
            'History',
            style: TextStyle(
             fontSize: 18,
             fontWeight:
FontWeight.bold,
            ),
           ),
           Text(
            'More',
            style: TextStyle(
             color: Colors.blue,
             fontSize: 18,
             fontWeight:
FontWeight.bold,
             letterSpacing: 1.2,
            ),
           )
          ],
         ),
         SizedBox(
          height: 20,
         ),
         Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
           for (var i in date) WeekDate(i),
          ],
         ),
         SizedBox(
          height: 20,
         ),
         Text(
          'RECORDS',
          style: TextStyle(
           fontSize: 18,
           color: Colors.blue,
          ),
         ),
         SizedBox(
          height: 10,
         ),
        ],
       ),
      ),
      SizedBox(
       height: 10,
      ),
      Container(
       padding:
EdgeInsets.symmetric(horizontal: 15,
vertical: 20),
       decoration: BoxDecoration(
        borderRadius:
BorderRadius.circular(10),
        color: Colors.white,
       ),
       child: Column(
```

```
        children: [
         Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
           Text(
            'Weight',
            style: TextStyle(
             fontSize: 18,
             fontWeight:
FontWeight.bold,
            ),
           ),
           GestureDetector(
            onTap: () {
             showDialog(
              context: context,
              builder: (context) =>
EditPhysicalDetails(),
             );
            },
            child: Icon(
             Icons.add,
             color: Colors.blue,
             size: 30,
            ),
           ),
          ],
         ),
         SizedBox(
          height: 20,
         ),
         Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
           Text(
            'Current',
            style: TextStyle(
             color: Colors.grey[400],
             fontSize: 18,
            ),
           ),
           Text(
            '$_currentWeight kg',
            style: TextStyle(
             color: Colors.grey[400],
             fontSize: 18,
            ),
           ),
          ],
                  ),
                  SizedBox(
                    height: 10,
                  ),
                  Row(
                    mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                    children: [
                      Text(
                        'Heaviest',
                        style: TextStyle(
                          color: Colors.grey[400],
                          fontSize: 18,
                        ),
                      ),
                      Text(
                        '$heaviestWeight kg',
                        style: TextStyle(
                          color: Colors.grey[400],
                          fontSize: 18,
                        ),
                      ),
                    ],
                  ),
                  SizedBox(
                    height: 10,
                  ),
                  Row(
                    mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                    children: [
                      Text(
                        'Lightest',
                        style: TextStyle(
                          color: Colors.grey[400],
                          fontSize: 18,
                        ),
                      ),
                      Text(
                        '$lightestWeight kg',
                        style: TextStyle(
                          color: Colors.grey[400],
                          fontSize: 18,
                        ),
                      ),
                    ],
                  ),
                  SizedBox(
                    height: 10,
                  ),
                ],
```

```
              ),
            ),
            SizedBox(
              height: 10,
            ),
            Container(
              padding:
EdgeInsets.symmetric(horizontal: 15,
vertical: 20),
              decoration: BoxDecoration(
                borderRadius:
BorderRadius.circular(10),
                color: Colors.white,
              ),
              child: Column(
                children: [
                  Row(
                    mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                    children: [
                      Text(
                        'BMI(kg/m2): 21.50',
                        style: TextStyle(
                          fontSize: 18,
                          fontWeight:
FontWeight.bold,
                        ),
                      ),
                      GestureDetector(
                        onTap: () {
                          showDialog(
                            context: context,
                            builder: (context) =>
EditPhysicalDetails(),
                          );
                        },
                        child: Text(
                          'EDIT',
                          style: TextStyle(
                            fontSize: 18,
                            fontWeight:
FontWeight.bold,
                            color: Colors.blue,
                          ),
                        ),
                      ),
                    ],
                  ),
                  SizedBox(
                    height: 20,
                  ),
         Image(
          image:
AssetImage('assets/bmi.jpg'),
          fit: BoxFit.fitWidth,
         ),
         SizedBox(
          height: 10,
         ),
        ],
       ),
      ),
      SizedBox(
       height: 10,
      ),
      Container(
       padding:
EdgeInsets.symmetric(horizontal: 15,
vertical: 20),
       decoration: BoxDecoration(
        borderRadius:
BorderRadius.circular(10),
        color: Colors.white,
       ),
       child: Column(
        children: [
         Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
           Text(
            'Height',
            style: TextStyle(
             fontSize: 18,
             fontWeight:
FontWeight.bold,
            ),
           ),
           GestureDetector(
            onTap: () {
             showDialog(
              context: context,
              builder: (context) =>
EditPhysicalDetails(),
             );
            },
            child: Text(
             'EDIT',
             style: TextStyle(
              fontSize: 18,
              fontWeight:
FontWeight.bold,
              color: Colors.blue,
             ),
            ),
           ),
          ],
         ),
         SizedBox(
          height: 20,
         ),
         Row(
          mainAxisAlignment:
MainAxisAlignment.spaceBetween,
          children: [
           Text(
            'Current',
            style: TextStyle(
             fontSize: 18,
             color: Colors.blue,
            ),
           ),
           Text(
            '$_currentHeightFT FT
$_currentHeightIN IN',
            style: TextStyle(
             fontSize: 18,
             color: Colors.grey[400],
            ),
           ),
          ],
         ),
         SizedBox(
          height: 20,
         ),
        ],
       ),
      )
     ],
    ),
   ),
  );
 }
}

class WeekDate extends StatelessWidget {
 int date;
 WeekDate(this.date, {super.key});

 @override
 Widget build(BuildContext context) {
  return Column(
   children: [
        Text(
          'W',
          style: TextStyle(
            fontSize: 15,
            color: Colors.grey[400],
          ),
        ),
        SizedBox(
          height: 10,
        ),
        Container(
          decoration: BoxDecoration(
            shape: BoxShape.circle,
            color: Colors.grey[300],
          ),
          padding: EdgeInsets.all(8),
          child: Container(
            padding: EdgeInsets.all(5),
            decoration: BoxDecoration(
              shape: BoxShape.circle,
              color: Colors.white,
            ),
          ),
        ),
        SizedBox(
          height: 10,
        ),
        Text(
          '$date',
          style: TextStyle(
            fontSize: 15,
            color: Colors.black,
            fontWeight: FontWeight.w600,
          ),
        ),
      ],
    );
  }
}
```

Main.dart

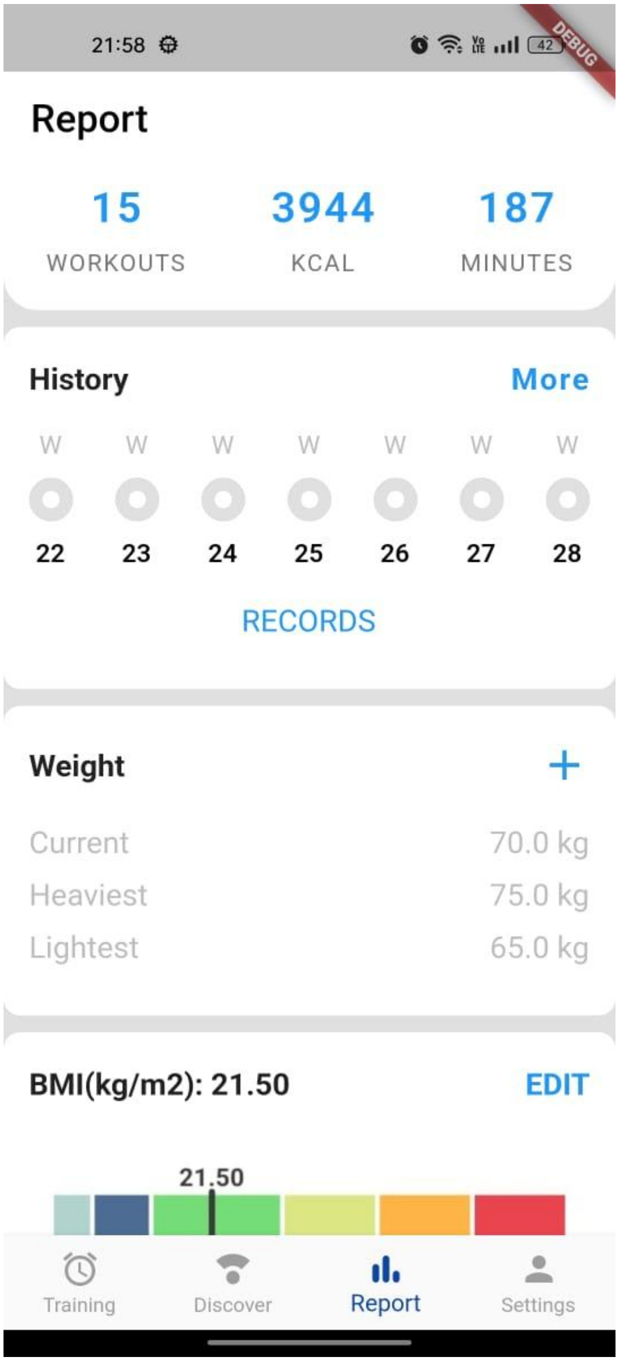
```
import 'package:flutter/material.dart';
import
'package:home_workout_app/discover.dart';
import
'package:home_workout_app/homepage.da
rt';
import
'package:home_workout_app/report.dart';
import
'package:home_workout_app/setting.dart';

void main() => runApp(
      MaterialApp(
        initialRoute: '/',
        routes: {
          '/': (context) => MainPage(),
          '/home': (context) => Home(),
          '/report': (context) => Report(),
          '/discover': (context) => Discover(),
          '/settings': (context) => Settings(),
        },
      ),
    );
```

```
class MainPage extends StatefulWidget {
  const MainPage({Key? key}) : super(key:
key);

  @override
  State<MainPage> createState() =>
_MainPageState();
}

class _MainPageState extends
State<MainPage> {
  int _currentIndex = 0;
  var pages = [
    Home(),
    Discover(),
    Report(),
    Settings(),
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: IndexedStack(
        index: _currentIndex,
        children: pages,
      ),
      bottomNavigationBar:
BottomNavigationBar(
        type: BottomNavigationBarType.fixed,
        currentIndex: _currentIndex,
        showSelectedLabels: true,
        showUnselectedLabels: true,
        selectedItemColor: Colors.blue[900],
        unselectedItemColor: Colors.grey,
        selectedIconTheme:
IconThemeData(color: Colors.blue[900]),
        onTap: ((value) => setState(() {
              _currentIndex = value;
              //Navigator.pushNamed(context,
'/${navbar[value]}');
            })),
        items: [
          BottomNavigationBarItem(
            icon: Icon(
              Icons.alarm,
            ),
            label: 'Training',
          ),
          BottomNavigationBarItem(
            icon: Icon(
              Icons.compass_calibration,
            ),
            label: 'Discover',
          ),
          BottomNavigationBarItem(
            icon: Icon(
              Icons.bar_chart_rounded,
            ),
            label: 'Report',
          ),
          BottomNavigationBarItem(
            icon: Icon(
              Icons.person,
            ),
            label: 'Settings',
          ),
        ],
      ),
    );
  }
}
```



Report

15	3944	187
WORKOUTS	KCAL	MINUTES

History More

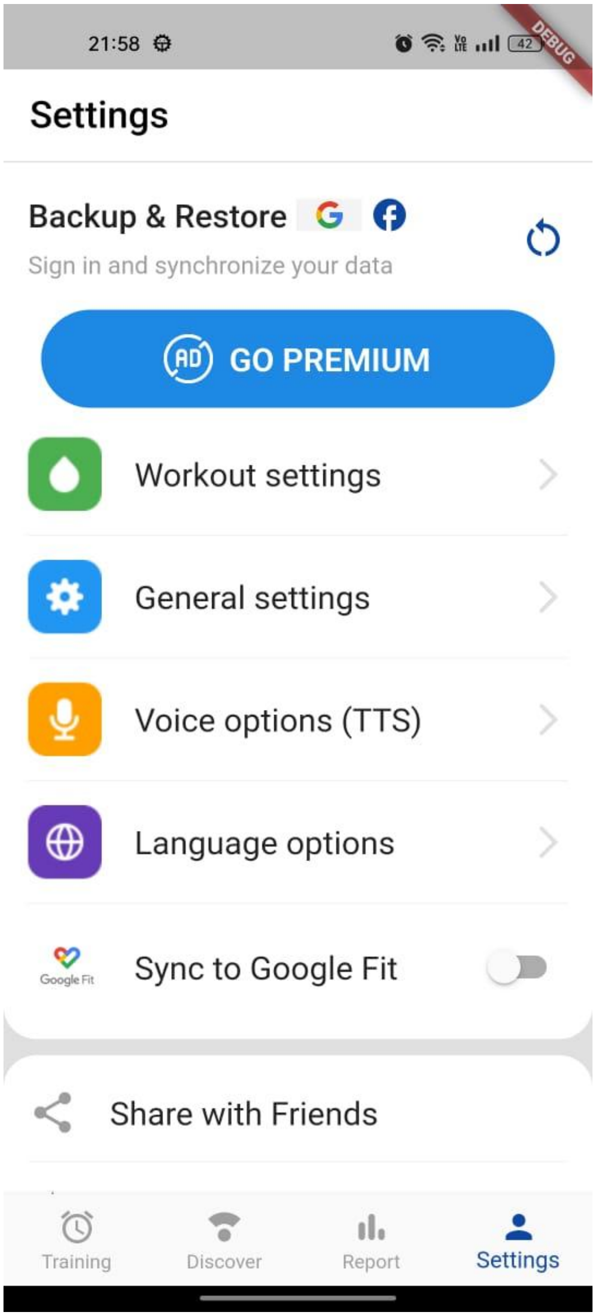
W	W	W	W	W	W	W
○	○	○	○	○	○	○
22	23	24	25	26	27	28

RECORDS

Weight +

Current	70.0 kg
Heaviest	75.0 kg
Lightest	65.0 kg

BMI(kg/m2): 21.50 EDIT



Conclusion -
We learned about forms and how to take user input. We created a form to take input from users and let the user change the values whenever they want.