Aim -
To create an interactive Form using the form widget.

Theory -

Forms are an integral part of all modern mobile and web applications. It is mainly used to interact with the app as well as gather information from the users. They can perform many tasks, which depend on the nature of your business requirements and logic, such as user authentication, adding users, searching, filtering, ordering, booking, etc. A form can contain text fields, buttons, checkboxes, radio buttons, etc.
Flutter provides a Form widget to create a form. The form widget acts as a container, which allows us to group and validate the multiple form fields. When you create a form, it is necessary to provide the GlobalKey. This key uniquely identifies the form and allows you to do any validation in the form fields.
The form widget uses a child widget TextFormField to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur.
Let us create a form. First, create a Flutter project and replace the following code in the main.dart file. In this code snippet, we have created a custom class named MyCustomForm. Inside this class, we define a global key as _formKey. This key holds a FormState and can use to retrieve the form widget. Inside the build method of this class, we have added some custom styles and used the TextFormField widget to provide the form fields such as name, phone number, date of birth, or just a normal field. Inside the TextFormField, we have used InputDecoration which provides the look and feel of your form properties such as borders, labels, icons, hints, styles, etc. Finally, we have added a button to submit the form.

Form validation
Validation is a method, which allows us to correct or confirm a certain standard. It ensures the authentication of the entered data.
Validating forms is a common practice in all digital interactions. We need to implement three steps to validate a form in a flutter.
Step 1: Use the Form widget with a global key.
Step 2: Use TextFormField to give the input field with the validator property.
Step 3: Create a button to validate form fields and display validation errors.

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',
```

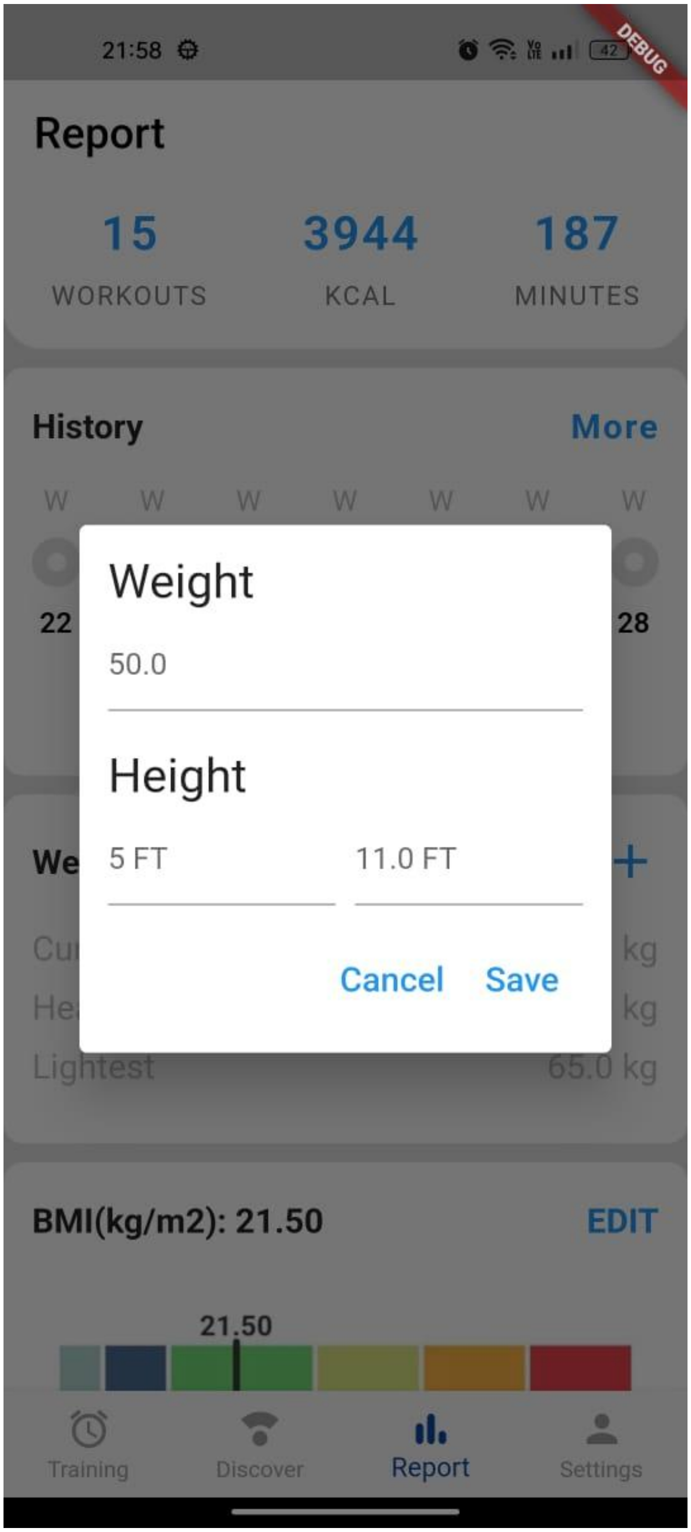
```
     style: TextStyle(
      fontSize: 23,
      color: Colors.black,
     ),
    ),
   ),
   body: SingleChildScrollView(
    child: Column(
     children: [
      Container(
       decoration: BoxDecoration(
        borderRadius: BorderRadius.only(
          bottomLeft: Radius.circular(20),
          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,
          ),
        ),
      ],
    );
  }
}
```



Weight

50.0

Height

5 FT 11.0 FT

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.