Experiment 4

Tanmay Bhosale
D15A 08

Aim: To create an interactive Form using form widget

Theory:
The Form widget in Flutter is a fundamental widget for building forms. It provides a way to group multiple form fields together, perform validation on those fields, and manage their state.

State Management: Flutter provides various ways to manage the state of interactive forms. You can use StatefulWidget or state management solutions like Provider, Riverpod, or Bloc pattern to handle form data changes efficiently. The choice depends on the complexity and requirements of your application.

Form Widgets: Flutter offers a range of form widgets such as TextField, Checkbox, Radio, DropdownButton, etc., which are used to collect input from users. These widgets can be customized with various parameters to fit the design and functionality of your interactive form.

Validation: Validating user input is essential for ensuring data integrity. Flutter provides a built-in mechanism for form validation using the Form widget along with TextFormField widgets. You can define validation logic for each form field and display error messages accordingly.

Input Handling: Flutter offers various options for handling user input, including keyboard input, gestures, and voice input. You can use event listeners like onChanged, onSubmitted, and onTap to capture user input and update the form state accordingly.

Theming and Styling: Flutter's flexible styling and theming system allow you to customize the appearance of form elements to match your app's design language. You can use themes, colors, fonts, and custom widgets to create visually appealing and consistent interactive forms.

Some Properties of Form Widget

- key: A GlobalKey that uniquely identifies the Form. You can use this key to interact with the form, such as validating, resetting, or saving its state.
- child: The child widget that contains the form fields. Typically, this is a Column, ListView, or another widget that allows you to arrange the form fields vertically.
- autovalidateMode: An enum that specifies when the form should automatically validate its fields.

- **Building a form for taking the information about the session to be set.**
- Information to be entered by the user:
    - Work Duration
    - Break Duration
    - Number of sessions

Code in habit.dart for taking the three required fields:

```
return GestureDetector(
        onTap: () => FocusManager.instance.primaryFocus?.unfocus(),
        child: Scaffold(
            resizeToAvoidBottomInset: false,
            backgroundColor: Colors.black,
            appBar: AppBar(
                automaticallyImplyLeading: false,
                centerTitle: false,
                backgroundColor: Colors.black,
                title: const Text.rich(
                  TextSpan(
                    text: 'Start session', // text for title
                    style: TextStyle(
                      fontSize: 24,
                      color: Colors.greenAccent,
                      fontFamily: 'Arial',
                    ),
                  ),
                )),
            body: SingleChildScrollView(
                child: (Container(
              width: double.infinity,
              color: Colors.black38,
              margin: const EdgeInsets.all(30),
              padding: const EdgeInsets.all(20),
              child: Column(
                children: [
                  const Text(
                    "Work duration",
                    style: TextStyle(
                      fontSize: 16,
                      color: Colors.white70,
                      fontFamily: 'Arial',
                    ),
                  ),
                  const SizedBox(
                      height:
                          10), // add a space between the text and the
```

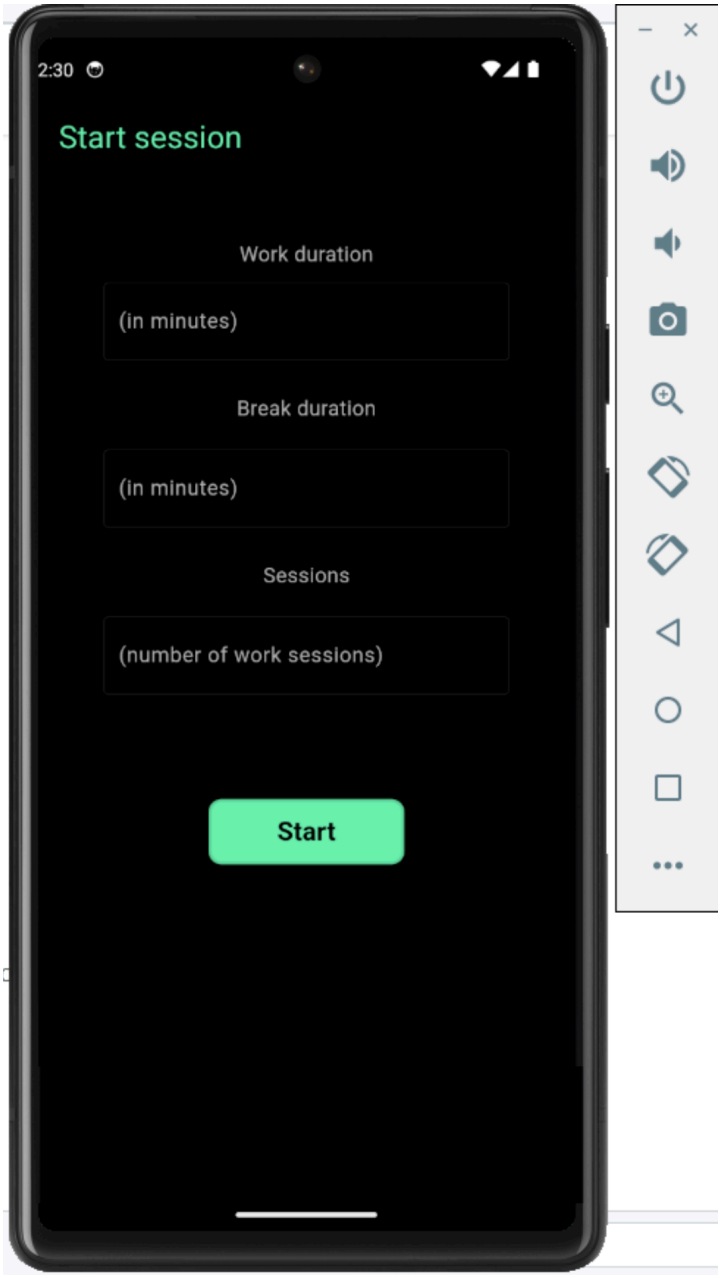
```
input field
                  TextField(
                    controller: workController,
                    textAlign: TextAlign.center,
                    style: const TextStyle(
                      fontSize: 13,
                      color: Colors.white70,
                      fontFamily: 'Arial',
                    ),
                    keyboardType:
                        TextInputType.number, // set the keyboard type to
number
                    keyboardAppearance: Brightness.dark,
                    decoration: const InputDecoration(
                      // filled: true,
                      fillColor: Colors.black12,
                      labelText: '(in minutes)',
                      labelStyle: TextStyle(color: Colors.white70),
                      enabledBorder: OutlineInputBorder(
                          borderRadius:
BorderRadius.all(Radius.circular(4.0)),
                          borderSide: BorderSide(color: Colors.white10)),
                      focusedBorder: OutlineInputBorder(
                          borderRadius:
BorderRadius.all(Radius.circular(4.0)),
                          borderSide: BorderSide(color: Colors.white10)),
                    ),
                  ),
                  const SizedBox(height: 25),
                  const Text(
                    "Break duration",
                    style: TextStyle(
                      fontSize: 16,
                      color: Colors.white70,
                      fontFamily: 'Arial',
                    ),
                  ),
                  const SizedBox(height: 20),
                  TextField(
                    controller: breakController,
                    textAlign: TextAlign.center,
                    style: const TextStyle(
                      fontSize: 13,
                      color: Colors.white70,
```

```
                    fontFamily: 'Arial',
                  ),
                  keyboardType:
                      TextInputType.number, // set the keyboard type to
number
                  keyboardAppearance: Brightness.dark,
                  decoration: const InputDecoration(
                    filled: true,
                    fillColor: Colors.black12,
                    labelText: '(in minutes)',
                    labelStyle: TextStyle(color: Colors.white70),
                    enabledBorder: OutlineInputBorder(
                        borderRadius:
BorderRadius.all(Radius.circular(4.0)),
                        borderSide: BorderSide(color: Colors.white10)),
                    focusedBorder: OutlineInputBorder(
                        borderRadius:
BorderRadius.all(Radius.circular(4.0)),
                        borderSide: BorderSide(color: Colors.white10)),
                  ),
                ),
                const SizedBox(
                    height:
                        25), // add a space between the text and the
input field
                const Text(
                  "Sessions",
                  style: TextStyle(
                    fontSize: 16,
                    color: Colors.white70,
                    fontFamily: 'Arial',
                  ),
                ),
                const SizedBox(height: 20),
                TextField(
                  controller: sessionController,
                  textAlign: TextAlign.center,
                  style: const TextStyle(
                    fontSize: 13,
                    color: Colors.white70,
                    fontFamily: 'Arial',
                  ),
                  keyboardType:
                      TextInputType.number, // set the keyboard type to
```

```
number
                    keyboardAppearance: Brightness.dark,
                    decoration: const InputDecoration(
                      // filled: true,
                      fillColor: Colors.black12,
                      labelText: '(number of work sessions)',
                      labelStyle: TextStyle(
                        color: Colors.white70,
                      ),
                      enabledBorder: OutlineInputBorder(
                          borderRadius:
BorderRadius.all(Radius.circular(4.0)),
                          borderSide: BorderSide(color: Colors.white10)),
                      focusedBorder: OutlineInputBorder(
                          borderRadius:
BorderRadius.all(Radius.circular(4.0)),
                          borderSide: BorderSide(color: Colors.white10)),
                    ),
                  ),
                  const SizedBox(
                      height:
                          80), // add a space between the text and the
input field
                  TextButton(
                    onPressed: () => Navigator.push(
                      context,
                      PageRouteBuilder(
                        transitionDuration: const Duration(seconds: 1),
                        pageBuilder: (context, animation,
secondaryAnimation) {
                          return FadeTransition(
                            opacity: animation,
                            child: MyTimer(
                                breakTime: breakController.text,
                                workTime: workController.text,
                                workSessions: sessionController.text),
                          );
                        },
                      ),
                    ),
                    style: TextButton.styleFrom(
                        backgroundColor: Colors.greenAccent,
                        padding: EdgeInsets.zero,
                        minimumSize: const Size(150, 50),
```

```
                        tapTargetSize: MaterialTapTargetSize.shrinkWrap,
                        alignment: Alignment.center,
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(10.0),
                          side: const BorderSide(
                              color: Colors.black12, width: 2.0),
                        )),
                    child: const Text(
                      "Start",
                      style: TextStyle(
                        fontSize: 20,
                        color: Colors.black,
                        fontWeight: FontWeight.bold,
                        fontFamily: 'Arial',
                      ),
                    ),
                  ),
                ],
              ),
            ))));
  }
}
```

Output:


2:30
Start session
Work duration
(in minutes)
Break duration
(in minutes)
Sessions
(number of work sessions)
Start