MOHD USMAAN MOGAL
36
D15A

EXP 4

AIM: To create an interactive form using form widget

Theory:

Form Fields:

The Form widget serves as a container for various form fields. These form fields could include text input fields for capturing user input, checkboxes for boolean selections, radio buttons for exclusive choices, and more. Each form field is responsible for gathering specific types of data from the user.
Validation:

Validation is a crucial aspect of form handling. The Form widget allows you to define validation logic for each form field. This validation ensures that the data entered by the user meets certain criteria. For example, you can check if an email field contains a valid email address or if a password meets certain complexity requirements.
Global Key:

The global key associated with the Form widget is used to uniquely identify and interact with the form. This key is essential for performing various operations on the form, such as triggering form submission and accessing its state. It provides a way to uniquely reference and manage the form within the widget tree.
Form Submission:

When a user attempts to submit the form, typically by interacting with a submit button, the Form widget comes into play. It checks the validation status of each form field. If all form fields are valid, you can execute a callback function to handle the submitted data. This could involve processing the data, sending it to a server, or navigating to another screen within your app.
State Management:

The Form widget is responsible for managing the state of the form. It keeps track of the values entered by the user, as well as the validation status of each form field. This state management simplifies the process of working with user input, ensuring a seamless and controlled experience for both developers and users.

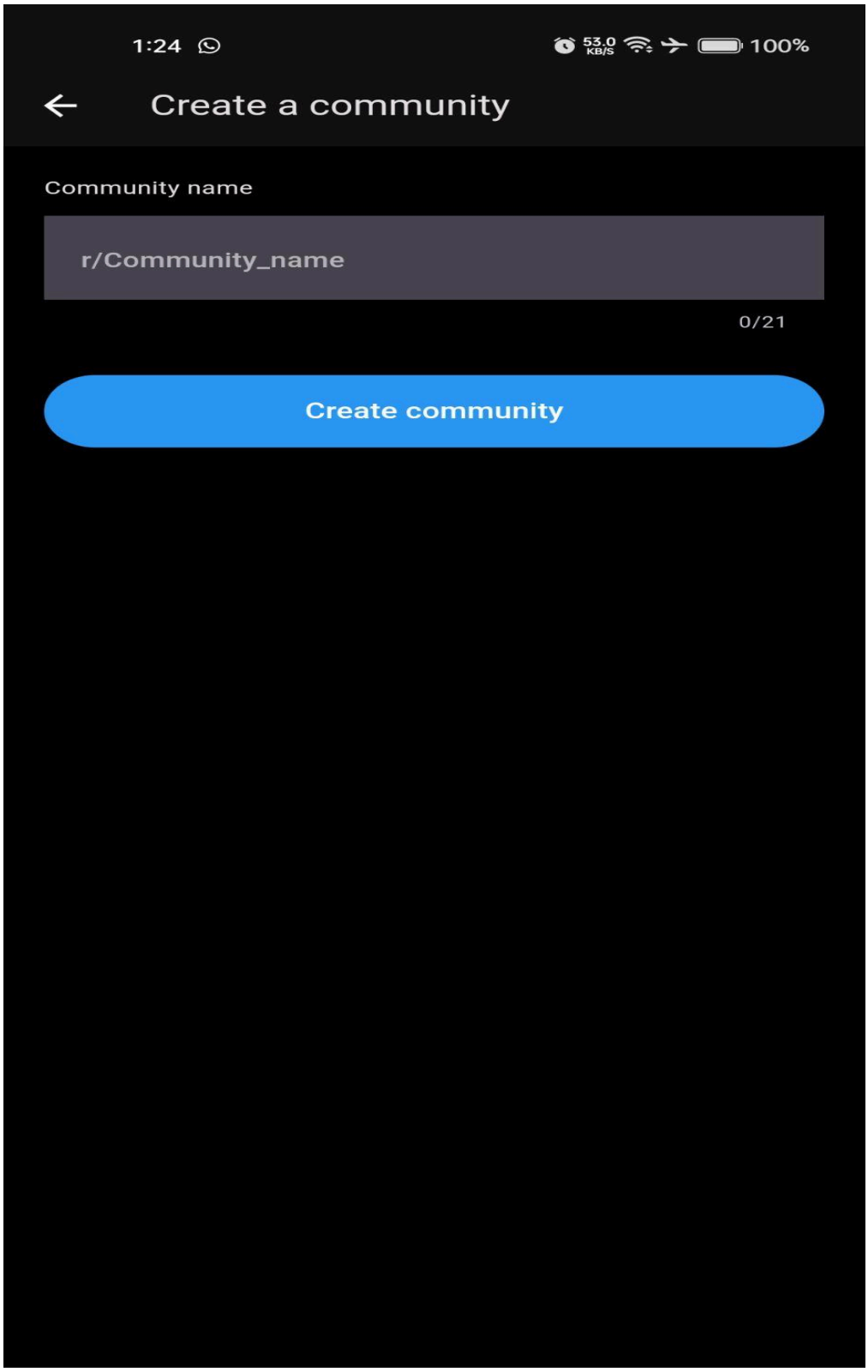
In essence, the Form widget in Flutter provides a structured and organized way to handle user input through a set of form fields. It encapsulates the complexities of validation, submission, and state management, making it easier for developers to build interactive and user-friendly forms in their applications.

Code:

```
// ignore_for_file: unnecessary_const

import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:reddit_clone/core/common/loader.dart';
import 'package:reddit_clone/features/community/controller/community_controller.dart';

class CreateCommunityScreen extends ConsumerStatefulWidget {
  const CreateCommunityScreen({super.key});

  @override
  ConsumerState<ConsumerStatefulWidget> createState() =>
      _CreateCommunityScreenState();
}

class _CreateCommunityScreenState extends ConsumerState<CreateCommunityScreen> {
  final communityNameController = TextEditingController();
  @override
  void dispose() {
    super.dispose();
    communityNameController.dispose();
  }

  void createCommmunity() {
    ref
        .read(communityControllerProvider.notifier)
        .createCommunity(communityNameController.text.trim(), context);
  }
  @override
  Widget build(BuildContext context) {
    final isLoading = ref.watch(communityControllerProvider);
    return Scaffold(
      appBar: AppBar(title: const Text('Create a community')),
      body: isLoading
          ? const Loader()
          : Padding(
              padding: const EdgeInsets.all(20.0),
```

```
            child: Column(
              children: [
                const Align(
                    alignment: Alignment.topLeft,
                    child: Text("Community name")),
                const SizedBox(
                  height: 10,
                ),
                TextField(
                  controller: communityNameController,
                  decoration: const InputDecoration(
                      hintText: 'r/Community_name',
                      filled: true,
                      border: InputBorder.none,
                      contentPadding: EdgeInsets.all(18)),
                  maxLength: 21,
                ),
                const SizedBox(
                  height: 30,
                ),
                ElevatedButton(
                  onPressed: createCommmunity,
                  style: ElevatedButton.styleFrom(
                      minimumSize: const Size(double.infinity, 50),
                      backgroundColor: Colors.blue),
                  child: const Text(
                    'Create community',
                    style: TextStyle(color: Colors.white, fontSize: 17),
                  ),
                )
              ],
            ),
          ),
    );
  }
}
```

Output:


1:24
53.0 KB/S
100%
Create a community
Community name
r/Community_name
0/21
Create community