

Shilpa Pandey
42 D15A

Experiment 4

AIM: To create an interactive Form using form widget

THEORY:

Form Widget: The Form widget serves as the container for form elements. It requires a key parameter, which is essential for accessing the form's state and performing actions like validation and submission.

Form Fields: Inside the Form widget, various form fields can be added. Flutter provides widgets like TextFormField, Checkbox, Radio, DropdownButton, etc., for collecting different types of user input.

State Management: To manage the state of the form, typically a StatefulWidget is used. The state class associated with the widget holds the form's state, including user input and validation status.

Validation: Form validation ensures that user input meets specific criteria before submission. Each form field can have a validator function, which returns an error message if the input is invalid. Flutter's Form widget automatically triggers validation when the form is submitted.

Submission Handling: When the user submits the form, a callback function is invoked to handle the form data. This function can access the form's state using the form key and perform tasks like data processing, database operations, or navigation.

Displaying Errors: Validation errors are displayed to users to guide them in correcting their input. Flutter allows developers to customize error messages and display them below the corresponding form fields.

User Interaction: Interactive elements like buttons are used to trigger form submission. By attaching event handlers to these elements, developers can control the flow of the form and provide feedback to users.

Form Reset: After successful submission or upon user request, the form can be reset to its initial state. This ensures a clean user experience for subsequent interactions.

Code

Main.dart

```
import 'package:flutter/material.dart';
import 'screens/sign_in_screen.dart';
import 'utils/color_themes.dart';
```

```
void main() {
  runApp(const AmazonClone());
}
```

```
class AmazonClone extends StatelessWidget {
  const AmazonClone({Key? key}) : super(key: key);
```

@override

```
Widget build(BuildContext context) {
  return MaterialApp(
    title: "Amazon Clone",
    debugShowCheckedModeBanner: false,
    theme: ThemeData.light().copyWith(
      scaffoldBackgroundColor: backgroundColor,
    ),
    home: const SignInScreen());
}
```

Sign_in_screen.dart

```
import 'package:flutter/material.dart';
import 'package:amazon_clone/screens/sign_up_screen.dart';
import 'package:amazon_clone/utils/color_themes.dart';
import 'package:amazon_clone/utils/constants.dart';
import 'package:amazon_clone/utils/Utils.dart';
import 'package:amazon_clone/widgets/custom_main_button.dart';
import 'package:amazon_clone/widgets/text_field_widget.dart';
```

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

```
  @override
  State<SignInScreen> createState() => _SignInScreenState();
}
```

```
class _SignInScreenState extends State<SignInScreen> {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  bool isLoading = false;
```

```
  @override
  void dispose() {
    super.dispose();
    emailController.dispose();
    passwordController.dispose();
  }
```

```
  @override
  Widget build(BuildContext context) {
    Size screenSize = Utils().getScreenSize();
```

```
return Scaffold(  
  backgroundColor: Colors.white,  
  body: SingleChildScrollView(  
    child: SizedBox(  
      height: screenSize.height,  
      width: screenSize.width,  
      child: Padding(  
        padding: const EdgeInsets.symmetric(horizontal: 10, vertical: 20),  
        child: Center(  
          child: Column(  
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
            crossAxisAlignment: CrossAxisAlignment.center,  
            children: [  
              Image.network(  
                amazonLogo,  
                height: screenSize.height * 0.10,  
              ),  
              Container(  
                height: screenSize.height * 0.6,  
                width: screenSize.width * 0.8,  
                padding: const EdgeInsets.all(25),  
                decoration: BoxDecoration(  
                  border: Border.all(  
                    color: Colors.grey,  
                    width: 1,  
                  ),  
                ),  
              ),  
              child: Column(  
                mainAxisAlignment: MainAxisAlignment.min,  
                mainAxisAlignment: MainAxisAlignment.spaceBetween,  
                crossAxisAlignment: CrossAxisAlignment.start,  
                children: [  
                  const Text(  
                    "Sign-In",  
                    style: TextStyle(  
                      fontWeight: FontWeight.w500, fontSize: 33),  
                  ),  
                ],  
              ),  
            ],  
          ),  
        ),  
      ),  
    ),  
  ),  
);
```

```

    TextFieldWidget(
      title: "Email",
      controller: emailController,
      obscureText: false,
      hintText: "Enter your email",
    ),
    TextFieldWidget(
      title: "Password",
      controller: passwordController,
      obscureText: true,
      hintText: "Enter your password",
    ),
    Align(
      alignment: Alignment.center,
      child: CustomMainButton(
        child: const Text(
          "Sign In",
          style: TextStyle(
            letterSpacing: 0.6, color: Colors.black),
        ),
        color: yellowColor,
        isLoading: isLoading,
        onPressed: () async {
          setState(() {
            isLoading = true;
          });
        },
      ),
    ),
  ],
),
),
Row(
  children: [
    Expanded(
      child: Container(

```

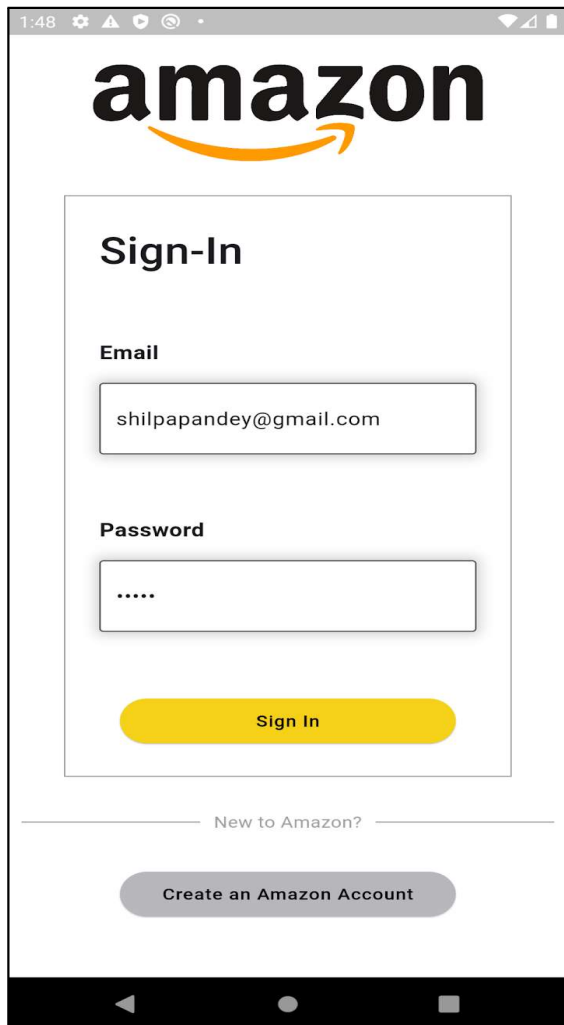
```

        height: 1,
        color: Colors.grey,
      ),
    ),
    const Padding(
      padding: EdgeInsets.symmetric(horizontal: 10),
      child: Text(
        "New to Amazon?",
        style: TextStyle(color: Colors.grey),
      ),
    ),
    Expanded(
      child: Container(
        height: 1,
        color: Colors.grey,
      ),
    ),
  ],
),
CustomMainButton(
  child: const Text(
    "Create an Amazon Account",
    style: TextStyle(
      letterSpacing: 0.6,
      color: Colors.black,
    ),
  ),
),
color: Colors.grey[400]!,
isLoading: false,
onPressed: () {
  Navigator.pushReplacement(context,
    MaterialPageRoute(builder: (context) {
      return const SignUpScreen();
    }));
})
],
),
),

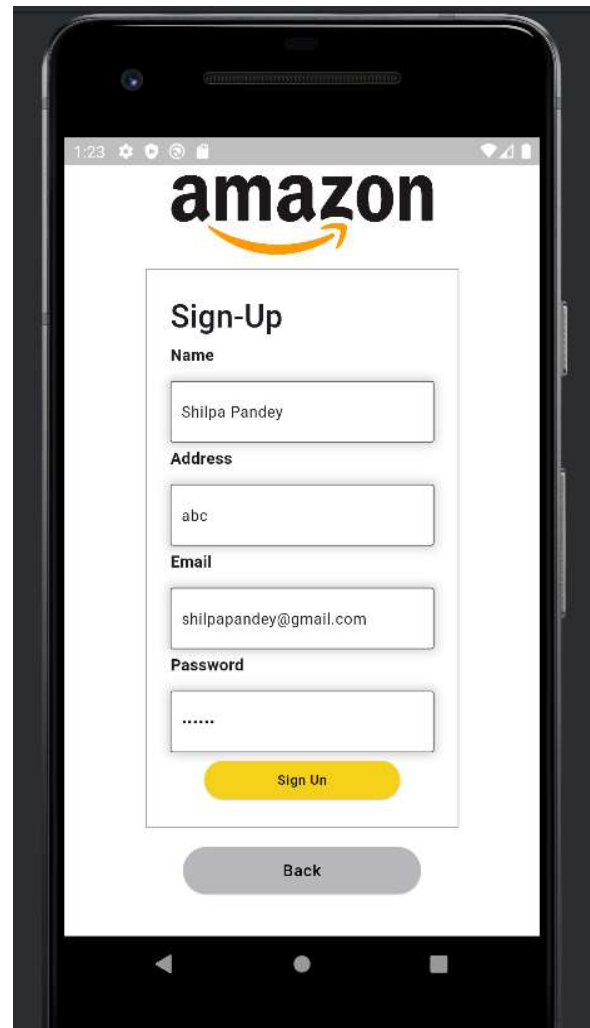
```

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

Output:



The image shows a desktop view of the Amazon Sign-In page. At the top is the Amazon logo. Below it is a 'Sign-In' section with two input fields: 'Email' containing 'shilpapandey@gmail.com' and 'Password' with masked characters '.....'. A yellow 'Sign In' button is below the fields. At the bottom, there is a link 'New to Amazon?' and a button 'Create an Amazon Account'.



The image shows a mobile view of the Amazon Sign-Up page. At the top is the Amazon logo. Below it is a 'Sign-Up' section with four input fields: 'Name' containing 'Shilpa Pandey', 'Address' containing 'abc', 'Email' containing 'shilpapandey@gmail.com', and 'Password' with masked characters '.....'. A yellow 'Sign Up' button is below the fields. At the bottom is a grey 'Back' button.

CONCLUSION:

Understand how to create an interactive Form using form widget