

EXPERIMENT NO.2

Experiment No. 2: To design Flutter UI by including common widgets.	
ROLL NO	2
NAME	Mohit Ailani
CLASS	D15-B
SUBJECT	MAD & PWA Lab
LO-MAPPED	

Aim: To design Flutter UI by including common widgets.

Theory:

Flutter is an open-source UI toolkit developed by Google for building natively compiled applications across multiple platforms, including mobile, web, and desktop, from a single codebase. It utilizes the Dart programming language and offers a rich set of customizable widgets for creating beautiful, fast, and responsive user interfaces. Widgets in Flutter are building blocks of the UI, ranging from basic elements like buttons and text to complex layouts like lists and grids. These widgets are highly flexible, enabling developers to create immersive and dynamic user experiences with ease, making Flutter a powerful choice for modern app development.

Basic Widgets:

- **AppBar:** A fundamental Flutter widget representing a material design app bar at the top of the screen. It typically contains a title, leading and trailing icons, and actions. It's commonly used for navigation and displaying key information.
- **Scaffold:** A foundational widget providing layout structure for apps, offering features like app bars, drawers, bottom navigation, and floating action buttons, simplifying app development.
- **Column:** A layout widget arranging its children vertically from top to bottom, enabling flexible alignment and spacing along the vertical axis for building UIs.
- **Text:** A widget displaying text content with customizable properties like font size, color, alignment, and decoration, facilitating text rendering in Flutter apps.

- **Container:** A versatile widget serving as a visual element with customizable properties such as size, color, padding, margin, alignment, decoration, and child widget composition, offering flexibility in UI design and layout.
- **Row:** A layout widget arranging its children horizontally from left to right within the parent widget's constraints. It enables flexible alignment and spacing along the horizontal axis, allowing developers to create UIs with multiple widgets laid out side by side.

Code:

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Vocab Builder',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LoginPage(),
    );
  }
}

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();

  String _username = "";
  String _password = "";

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Vocabulary Builder'),
      ),
      body: Form(
```

```

key: _formKey,
child: Padding(
  padding: const EdgeInsets.all(20.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Center(
        child: Text(
          'Vocabulary Builder',
          style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
        ),
      ),
      SizedBox(height: 20.0),
      TextFormField(
        decoration: InputDecoration(labelText: 'Username'),
        validator: (value) {
          if (value!.isEmpty) {
            return 'Please enter your username';
          }
          return null;
        },
        onSave: (value) => _username = value!,
      ),
      TextFormField(
        obscureText: true,
        decoration: InputDecoration(labelText: 'Password'),
        validator: (value) {
          if (value!.isEmpty) {
            return 'Please enter your password';
          }
          return null;
        },
        onSave: (value) => _password = value!,
      ),
      SizedBox(height: 20.0),
      ElevatedButton(
        onPressed: () {
          // Implement login logic here (replace with actual authentication)
          if (_formKey.currentState!.validate()) {
            _formKey.currentState!.save();
            Navigator.push(context, MaterialPageRoute(builder: (context) => WelcomePage()));
          }
        },
        child: Text('Login'),
      ),
    ],
  ),
),
);
}

```

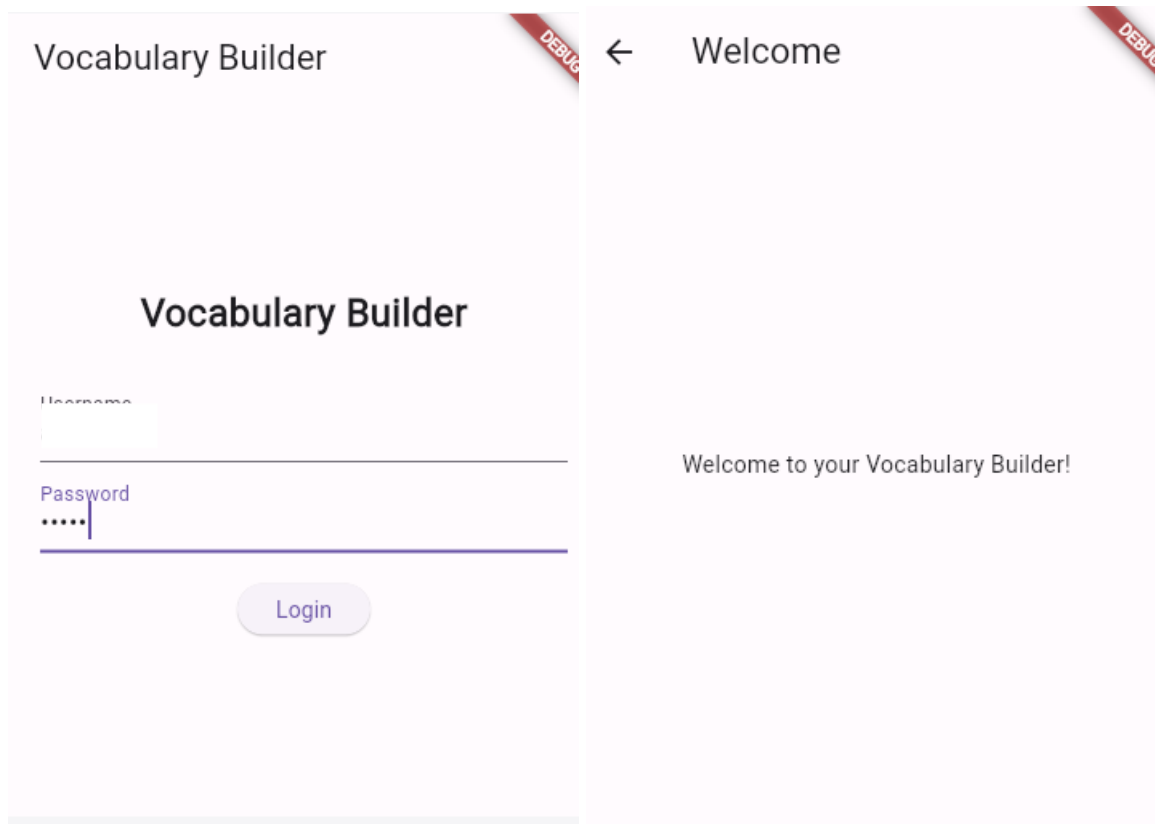
```

}

class WelcomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Welcome'),
      ),
      body: Center(
        child: Text('Welcome to your Vocabulary Builder!'),
      ),
    );
  }
}

```

Output:



Conclusion:

Flutter's common widgets, implementing AppBar, Scaffold, Column, and Row to design a comprehensive UI. This approach highlighted the versatility of Flutter in creating dynamic layouts while ensuring structural integrity and user interaction. Such implementation underscores Flutter's efficacy in developing intuitive and visually appealing applications.