EXPERIMENT-2

Aim : To design Flutter UI by including common widgets.

Theory:

- Flutter is Google's UI toolkit for crafting beautiful, natively compiled iOS and Android apps from a single code base. To build any application we start with widgets – The building block of flutter applications.
- Widgets describe what their view should look like given their current configuration and state. It includes a text widget, row widget, column widget, container widget, and many more.
- Widgets: Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. And the structure of the code of an app is a tree of widgets.

Category of Widgets:

There are mainly 14 categories in which the flutter widgets are divided. They are mainly segregated on the basis of the functionality they provide in a flutter application.

1. *Accessibility*: These are the set of widgets that make a flutter app more easily accessible.
2. *Animation and Motion:* These widgets add animation to other widgets. *3.Assets, Images, and Icons*: These widgets take charge of assets such as display images and show icons.
4. *Async*: These provide async functionality in the flutter application.
5. *Basics:* These are the bundle of widgets that are absolutely necessary for the development of any flutter application.

6. *Cupertino*: These are the iOS designed widgets.
7. *Input*: This set of widgets provides input functionality in a flutter application.
8. *Interaction Models:* These widgets are here to manage touch events and route users to different views in the application.
9. *Layout:* This bundle of widgets helps in placing the other widgets on the screen as needed.
10. *Material Components:* This is a set of widgets that mainly follow material design by Google.
11. *Painting and effects:* This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
12. *Scrolling:* This provides scrollability of to a set of other widgets that are not scrollable by default.
13. *Styling:* This deals with the theme, responsiveness, and sizing of the app.
14. *Text*: This displays text

Types of Widgets:

There are broadly two types of widgets in the flutter:

1. Stateless Widget - These are immutable widgets that don't change over time.

- The UI of a stateless widget is defined based on the configuration information passed to it during its creation.
- Example: Container, Text, Icon.

2. Stateful Widget - These are mutable widgets that can change dynamically.

- Stateful widgets maintain a mutable state that might change during the widget's lifetime.
- Example: TextField, ListView, Form.

Interaction Models: These widgets are here to manage touch events and route users to different views in the application.
Layout: This bundle of widgets helps in placing the other widgets on the screen as needed.
Material Components: This is a set of widgets that mainly follow material design by Google.
Painting and effects: This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
Scrolling: This provides scrollability of to a set of other widgets that are not scrollable by default.
Styling: This deals with the theme, responsiveness, and sizing of the app.
Text: This displays text.

Types of Widgets:

There are broadly two types of widgets in the flutter:

1. Stateless Widget - These are immutable widgets that don't change over time.

- The UI of a stateless widget is defined based on the configuration information passed to it during its creation.
- Example: Container, Text, Icon.

2. Stateful Widget - These are mutable widgets that can change dynamically.

- Stateful widgets maintain a mutable state that might change during the widget's lifetime.
- Example: TextField, ListView, Form.

Code-

```
          child: Icon(
            Icons.arrow_back,
            size: Dimensions.iconSize24,
            color: Theme.of(context).primaryColor,
          ),
        ),
        SizedBox(width: Dimensions.width30 * 5),
        const RentoHeading(),
      ],
    ),

    SizedBox(height: Dimensions.height30),

    // EMAIL INPUT FIELD
    TextFormField(
      key: const ValueKey('email'),
      validator: (value) {
        if (value!.isEmpty) {
          return "The email field can not be empty";
        } else if (!value.contains('@')) {
          return 'Please enter a valid email address.';
        } else if (value.contains('.com.com')) {
          return 'Please enter a valid email address.';
        }
        return null;
      },
      onSaved: (value) {
        _userEmail = value!;
      },
      keyboardType: TextInputType.emailAddress,
      cursorColor: Theme.of(context).primaryColor,
      style: TextStyle(
        color: Theme.of(context).primaryColor,
        fontSize: Dimensions.font16,
      ),
      decoration: InputDecoration(
        labelText: 'email',
        labelStyle: TextStyle(
```

```
          fontSize: Dimensions.font15,
          color: Theme.of(context).primaryColor,
        ),
        border: inputBorder,
        focusedBorder: inputBorder,
      ),
    ),

    SizedBox(height: Dimensions.height20),

    // PASSWORD INPUT
    TextFormField(
      key: const ValueKey('password'),
      validator: (value) {
        if (value!.isEmpty) {
          return "The password field can not be empty";
        }
        return null;
      },
      onSaved: (value) {
        _userPassword = value!;
      },
      cursorColor: Theme.of(context).primaryColor,
      style: TextStyle(
        color: Theme.of(context).primaryColor,
        fontSize: Dimensions.font16,
      ),
      decoration: InputDecoration(
        labelStyle: TextStyle(
          color: Theme.of(context).primaryColor,
          fontSize: Dimensions.font15,
        ),
        labelText: 'password',
        border: inputBorder,
        suffixIcon: GestureDetector(
          onTap: _toggleObscured,
          child: Icon(
              _obscured
                  ? Icons.visibility_off_rounded
                  : Icons.visibility_rounded,
              size: Dimensions.iconSize24,
              color: greyColor),
        ),
        focusedBorder: inputBorder,
```

```
                  ),
                  keyboardType: TextInputType.visiblePassword,
                  obscureText: _obscured,
                  focusNode: textFieldFocusNode,
                ),

                SizedBox(height: Dimensions.height20),

                // LOGIN BUTTON
                GestureDetector(
                  onTap: _trySubmit,
                  child: Container(
                    width: double.infinity,
                    alignment: Alignment.center,
                    padding:
                        EdgeInsets.symmetric(vertical:
Dimensions.height15),                    decoration:
                      ShapeDecorati
                      on( shape:
                      RoundedRectan
                      gleBorder(
                        borderRadius:
                          BorderRadius.all(
                          Radius.circular(
                            Dimensions.radius20 / 2,
                      ),
                    ),
                  ),
                    color: purpleColor,
                ),
                  child: _isLoading
                      ? SizedBox(
                          width: Dimensions.width20,
                          height: Dimensions.height20,
                          child: const CircularProgressIndicator(
                            color: whiteColor,
                            strokeWidth: 2.5,
                          ),
                        )
                      : SmallText(
                          text: "Login",
                          size: Dimensions.font15,
                          color: whiteColor,
                        ),
                ),
```

```
),
// LOGIN BUTTON
```

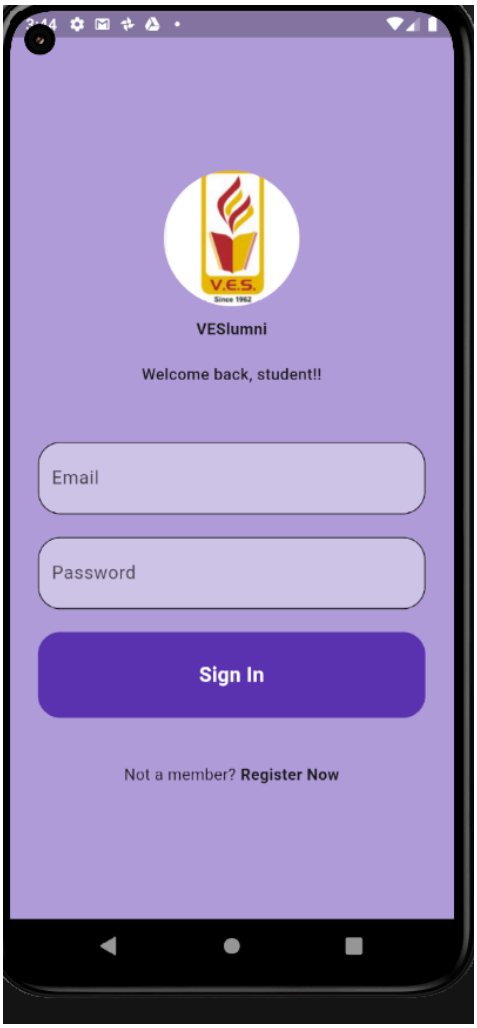
```
Container(
  margin: EdgeInsets.only(
    top: Dimensions.height10,
  ),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      // FORGOT PASSWORD LINK
      GestureDetector(
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (ctx) {
                return const ForgotPassword();
              },
            ),
          );
        },
        child: Container(
          decoration: BoxDecoration(
            border: Border(
              bottom: BorderSide(
                color: Theme.of(context).primaryColor,
                width: 0.5,
              ),
            ),
          ),
          child: SmallText(
            text: 'Forgot password',
            size: Dimensions.font15,
            color: Theme.of(context).primaryColor,
          ),
        ),
      ),

      // NAVIGATE USER TO SIGN UP PAGE
      GestureDetector(
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (ctx) {
```

```
                                return const SignUpBy();
                              },
                            ),
                          );
                        },
                        child: Row(
                          children: [
                            SmallText(
                              text: "Don't have an account?",
                              size: Dimensions.font15,
                              color: greyColor,
                            ),
                            SizedBox(width: Dimensions.width10 / 2),
                            Container(
                              decoration: BoxDecoration(
                                border: Border(
                                  bottom: BorderSide(
                                    color:
Theme.of(context).primaryColor,                 width: 0.5,
                                  ),
                                ),
                              ),
                              child: const BigText(text: 'Sign up'),
                            ),
                          ],
                        ),
                      )
                    ],
                  ),
                ),
                SizedBox(height: Dimensions.height30 * 2),
                Row(
                  children: [
                    Expanded(
                      child: Divider(
                        color: isDark ? whiteColor : greyColor,
                        thickness: 0.2,
                      ),
                    ),
                    SizedBox(width: Dimensions.width8 / 2),
                    SmallText(
```

```
            text: 'OR',
            size: Dimensions.font16,
          ),
          SizedBox(width: Dimensions.width8 / 2),
          Expanded(
            child: Divider(
              color: isDark ? whiteColor : greyColor,
              thickness: 0.2,
            ),
          ),
        ],
      ),

      SizedBox(height: Dimensions.height20),

      // CONTINUE WITH GOOGLE
      GestureDetector(
        onTap: () {},
        child: const ContinueWith(
          text: 'Continue with Google',
          iconUrl: 'assets/icons/google_icon.png',
        ),
      ),
      // CONTINUE WITH GOOGLE

      SizedBox(height: Dimensions.height20),

      // CONTINUE WITH FACEBOOK
      GestureDetector(
        onTap: () {},
        child: const ContinueWith(
          text: 'Continue with Facebook',
          iconUrl: 'assets/icons/facebook_icon.png',
        ),
      ),
      // CONTINUE WITH FACEBOOK

      SizedBox(height: Dimensions.height20),

      // CONTINUE WITH APPLE
      GestureDetector(
        onTap: () {},
        child: ContinueWith(
          iconUrl: 'assets/icons/apple_icon.png',
```

```
text: 'Continue with Apple',
iconColor: Theme.of(context).primaryColor,
                    ),
                  ),
// CONTINUE WITH APPLE
SizedBox(height: Dimensions.height30),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

Output-



CONCLUSION: Hence we have successfully understood the implementation of basic widgets in flutter and created a login page with it