

## Experiment 4

**Aim:** To create an interactive Form using form widget

### Theory:

In Flutter, the Form widget is a crucial component for building interactive user input forms. It facilitates input validation, data submission, and error handling. Here's a brief overview of creating an interactive form using the Form widget in Flutter:

#### 1. Form Widget:

The Form widget in Flutter is a container for form fields and allows you to group multiple form fields together. It helps in managing form state, validation, and submission. When a form is submitted, the Form widget calls a callback function (onFormSubmitted) where you can handle the submitted data.

#### 2. TextFormField:

TextFormField is a widget used to capture textual input from users. It provides options for text input, such as validation, input formatting, and obscure text (password fields). You can use validators like required, minLength, maxLength, email, etc., to validate user input.

#### 3. DropdownButtonFormField:

DropdownButtonFormField is a form field widget that presents a dropdown menu of items for users to select from. It's commonly used when users need to choose from a predefined list of options. You can set an initial value, define the items in the dropdown menu, and handle the selection using onChanged callback.

#### 4. CheckboxListTile and RadioListTile:

These widgets are used for capturing boolean and single-choice input, respectively. CheckboxListTile presents a checkbox that can be checked or unchecked by the user, while RadioListTile presents a group of radio buttons from which the user can select one option.

#### 5. Form Validation:

Form validation ensures that user input meets certain criteria or constraints before it's accepted as valid. In Flutter, you can implement form validation by providing a validator function for each form field. The validator function takes the current value of the field as input and returns an error message if the value is not valid. The form will display error messages if validation fails.

#### 6. Form Submission:

After users have filled out the form, they typically submit it to perform an action, such as saving the data to a database or sending it to a server. You can handle form submission by providing an onSubmit callback function to the Form widget. This function is called when the form is submitted, and you can access the current values of the form fields within this callback to perform the necessary actions.



KHYAAL


Email

Password

[Forgot Password?](#)

Login

Don't have an account?[Register Now](#)



KHYAAL

Email

username

Password

confirm Password

[Forgot Password?](#)

**register**

[Already have an account?Login Here](#)

Registration form has password validation.

Conclusion:Created home page and profile page.