

Experiment 4	
Name	Chinmay Chaudhari
Rollno	6
Class	D15C
DOP	
DOS	
Sign	
Grade	

**AIM:** - To create an interactive Form using the **Form widget** in Flutter.

### Description:

In Flutter, a **Form** is a container for grouping and validating multiple form fields. It provides an easy way to manage user input, validate it, and handle form submission.

Flutter offers two key widgets for creating forms:

1. **Form Widget** – a container that holds multiple form fields and tracks their state.
2. **TextFormField Widget** – a specialized input field that integrates with the Form widget for validation and state tracking.

To use a form:

- We first create a `GlobalKey<FormState>` to uniquely identify the Form and allow validation.
- Inside the Form widget, we include one or more `TextFormField` widgets.
- Each `TextFormField` can have built-in validation using the `validator` parameter.
- We use a `RaisedButton` or `ElevatedButton` to trigger form submission and validation.

### Important Components:

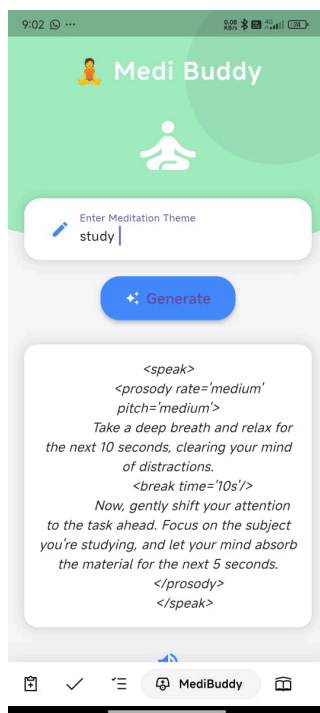
- `GlobalKey<FormState>`: A key to access the form state.
- `Form`: Widget to group multiple form fields.
- `TextFormField`: Input field with built-in validation support.

- validator: A function that returns an error string if validation fails or null if valid.
- formKey.currentState!.validate(): Triggers validation for all fields.

### Form Validation Process:

1. User fills in data.
2. On submit, formKey.currentState!.validate() is called.
3. Each TextFormField runs its validator.
4. If all return null, form is valid.
5. Data can then be saved or processed.

### Output:



GitHub Link - [medi\\_buddy\\_view.dart](#)

### Conclusion:

In this experiment, we successfully created an interactive form using Flutter's Form and TextFormField widgets. We learned how to validate user input using validators and how to manage form state with a GlobalKey. This allows us to build structured, responsive, and user-friendly form-based interfaces in mobile applications.