Name:Ajinkya S Patil **Exp No.:04** **D15B-42**

Experiment 4

Aim:To Create an Interactive Form Using Form Widgets

Interactive forms are essential in modern web applications as they allow users to input and submit data to the server for processing. These forms typically include a variety of form widgets, which enhance user engagement, making the form more dynamic and interactive. Form widgets include text boxes, buttons, checkboxes, radio buttons, dropdown lists, and more. In this experiment, the objective is to understand the importance of form widgets and how to create an effective and interactive form by using them.

What are Form Widgets?

Form widgets are HTML elements or controls used in web forms to enable users to input data in a structured way. They are the interface between the user and the backend system, helping to collect user inputs in a variety of formats (text, numbers, dates, selections, etc.). By using different form widgets, users can interact with a form in different ways, providing a wide range of inputs such as typing, selecting, or checking options.

Types of Form Widgets:

1. **Text Input Fields**: Text input fields are used for collecting short and long text information from the user. There are several variations, such as:
    - **Single-line text input** (`<input type="text">`) for basic text.
    - **Password fields** (`<input type="password">`) to hide the entered text for security.
    - **Email fields** (`<input type="email">`) to ensure proper email format validation.
    - **Textarea** (`<textarea>`) for longer multi-line text input.
2. **Radio Buttons**: Radio buttons are used when you want the user to select one option from a group of predefined choices. It ensures that only one option can be selected at a time, making it ideal for situations like selecting a gender, a payment method, or any other single-choice question.
3. **Checkboxes**: Checkboxes allow users to select one or multiple options from a list. For example, if you want the user to choose their hobbies, checkboxes enable them to select multiple hobbies from a list.
4. **Dropdown Menus (Select Boxes)**: Dropdown menus (`<select>`) allow users to choose one option from a drop-down list. They are useful when there are too many options to display on the screen at once, saving space while keeping the user experience smooth.
5. **Buttons**: Buttons in a form are used to trigger an action, typically to submit the data. There are different types of buttons:
    - **Submit button** (`<button type="submit">`) triggers form submission.
    - **Reset button** (`<button type="reset">`) clears all form fields to their initial state.
    - **Custom action buttons** can perform specific tasks, such as navigating to another page or triggering dynamic behavior through JavaScript.
6. **Date Pickers**: Date input fields allow users to pick a date from a calendar. Modern browsers provide a user-friendly date picker interface when using `<input type="date">`, ensuring consistency in date formatting.

Importance of Interactive Forms:

Forms are a primary means of communication between a user and a web application. An interactive form offers the following advantages:

- **User Engagement**: Interactive forms make it easy and enjoyable for users to enter information, reducing the chances of form abandonment.
- **Error Prevention**: Real-time form validation helps users correct their input as they type, preventing errors from being submitted and improving data accuracy.
- **Data Collection Efficiency**: Form widgets simplify complex data collection processes by using specialized inputs (such as date pickers, sliders, or dropdowns) for specific data types. This ensures that the collected data is structured, valid, and usable.

Form Validation:

Form validation ensures that the data provided by users is accurate, complete, and in the required format before submission. It can be achieved through HTML5 attributes or JavaScript. Validation checks include:

- **Required fields**: Certain fields must be filled before the form is submitted (e.g., name, email).
- **Pattern validation**: Using regular expressions, inputs can be validated for specific formats (e.g., phone numbers, email addresses).
- **Range validation**: Numeric inputs can be restricted to certain ranges (e.g., age between 18 and 60).
- **Real-time feedback**: JavaScript allows form fields to display validation messages as the user is typing, improving user experience by guiding them to correct errors immediately.

Enhancing User Experience:

User experience (UX) in forms is critical for achieving high completion rates. Several principles help enhance the UX:

- **Clear Labels and Instructions**: Labels should clearly describe what is expected in each field, while instructions guide the user through the process. For example, input placeholders or tooltips can provide hints about acceptable values.
- **Responsive Layout**: Forms should be designed to adjust to various screen sizes, making them accessible on mobile devices, tablets, and desktops. CSS media queries can help ensure proper form scaling and alignment across devices.
- **Auto-fill and Auto-complete**: Modern web browsers offer auto-fill suggestions based on users' previously entered information. This reduces the time users spend on forms, especially when entering commonly used information like name, address, or email.
- **Form Progression**: For lengthy forms, splitting them into multiple sections with a clear progression bar helps users navigate through and complete the form. This makes the form less overwhelming and allows users to track their progress.

Dynamic Forms:

Dynamic forms are forms that change their appearance or behavior based on user interactions. For example:

- Showing or hiding sections of a form based on user input (e.g., selecting "Other" from a dropdown could trigger an additional text field for more details).
- Pre-filling certain fields based on previous inputs or user data.
- Displaying success or error messages after submission to confirm whether the form was completed successfully or if corrections are needed.

Accessibility and Inclusivity:

Accessibility is critical when designing forms to ensure all users, including those with disabilities, can interact with and complete the form:

- **Labels for Inputs**: Every form input must have an associated label to make the input field identifiable for screen readers.
- **Keyboard Navigation**: Users should be able to navigate through form fields using the keyboard (e.g., using the "Tab" key to move between fields).
- **ARIA Attributes**: Adding ARIA (Accessible Rich Internet Applications) attributes improves the accessibility of dynamic content and form validation, helping users with disabilities interact with the form more effectively.

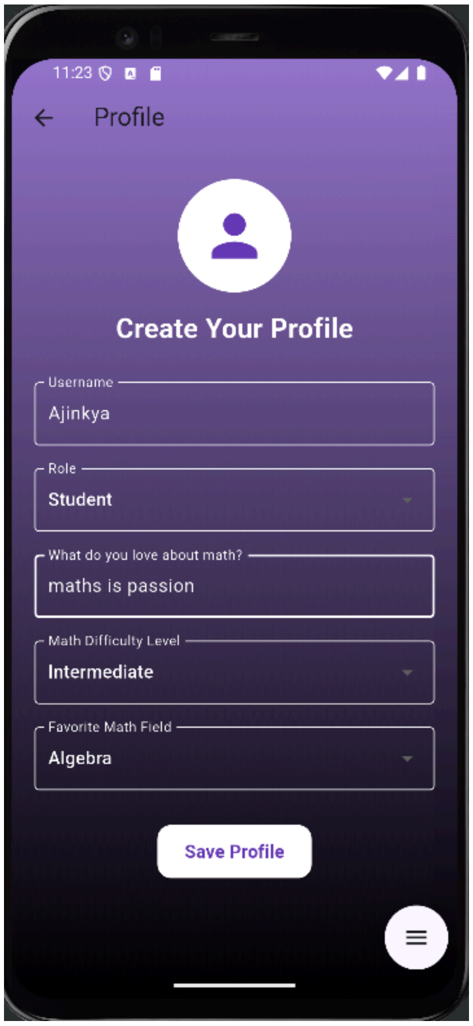
Conclusion:

Creating an interactive form using form widgets is crucial for user interaction and data collection in web applications. By utilizing various form widgets such as text fields, buttons, checkboxes, and radio buttons, we can design intuitive and user-friendly forms. Additionally, incorporating validation, responsiveness, and accessibility ensures the form provides a seamless experience for all users, resulting in accurate data collection and high completion rates. Interactive forms are an essential part of modern web design, bridging the gap between users and the application backend in an efficient and effective way.

CODE:

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

class ProfileScreen extends StatefulWidget {
  @override
  _ProfileScreenState createState() => _ProfileScreenState();
}

class _ProfileScreenState extends State<ProfileScreen> {
  final _formKey = GlobalKey<FormState>();

  final TextEditingController _usernameController = TextEditingController();
  final TextEditingController _mathInterestController = TextEditingController();

  String role = "Student";
  String difficultyLevel = "Intermediate";
  String favoriteMathField = "Algebra";

  final List<String> difficultyLevels = ["Beginner", "Intermediate", "Advanced"];
  final List<String> mathFields = ["Algebra", "Geometry", "Calculus", "Trigonometry", "Statistics"];

  @override
  void dispose() {
    _usernameController.dispose();
    _mathInterestController.dispose();
    super.dispose();
  }

  void _showSuccessMessage() {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text("Profile Updated Successfully!", style: TextStyle(fontSize: 16)),
        backgroundColor: Colors.green,
        duration: Duration(seconds: 2),
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBodyBehindAppBar: true,
      appBar: AppBar(
        title: Text("Profile"),
        backgroundColor: Colors.transparent,
        elevation: 0,
      ),
      body: Container(
        width: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [Colors.deepPurple.shade300, Colors.black],
          ),
        ),
        child: Center(
          child: SingleChildScrollView(
            padding: EdgeInsets.all(20.0),
            child: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
                  // 🟢 User Avatar
                  CircleAvatar(
                    radius: 50,
                    backgroundColor: Colors.white,
                    child: Icon(Icons.person, size: 60, color: Colors.deepPurple),
                  ),

                  SizedBox(height: 15),

                  // Title
                  Text(
                    "Create Your Profile",
                    style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold, color: Colors.white),
                  ),

                  SizedBox(height: 30),

                  _buildTextField("Username", _usernameController),

                  SizedBox(height: 20),

                  _buildDropdown("Role", ["Student", "Teacher"], role, (value) => setState(() => role = value!)),

                  SizedBox(height: 20),

                  _buildTextField("What do you love about math?", _mathInterestController),

                  SizedBox(height: 20),

                  _buildDropdown("Math Difficulty Level", difficultyLevels, difficultyLevel, (value) => setState(() => difficultyLevel
= value!)),

                  SizedBox(height: 20),

                  _buildDropdown("Favorite Math Field", mathFields, favoriteMathField, (value) => setState(() => favoriteMathField =
value!)),

                  SizedBox(height: 30),

                  Center(
                    child: ElevatedButton(
                      onPressed: () {
                        if (_formKey.currentState!.validate()) {
                          print("Username: ${_usernameController.text}, Role: $role, Interest: ${_mathInterestController.text}, Level:
$difficultyLevel, Favorite: $favoriteMathField");
                          _showSuccessMessage(); // Show success message after saving
                        }
                      },
                      style: ElevatedButton.styleFrom(
                        backgroundColor: Colors.white,
                        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
                        padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),
                      ),
                      child: Text("Save Profile", style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold, color: Colors.deepPurple)),
                    ),
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }

  Widget _buildTextField(String label, TextEditingController controller) {
    return TextFormField(
      controller: controller,
      style: TextStyle(color: Colors.white),
      decoration: InputDecoration(
        labelText: label,
        labelStyle: TextStyle(color: Colors.white),
        enabledBorder: OutlineInputBorder(borderSide: BorderSide(color: Colors.white)),
        focusedBorder: OutlineInputBorder(borderSide: BorderSide(color: Colors.white, width: 2)),
      ),
      validator: (value) => value!.isEmpty ? "Enter $label" : null,
    );
  }

  Widget _buildDropdown(String label, List<String> items, String selectedValue, Function(String?) onChanged) {
    return DropdownButtonFormField<String>(
      dropdownColor: Colors.grey[900],
      value: selectedValue,
      decoration: InputDecoration(
        labelText: label,
        labelStyle: TextStyle(color: Colors.white),
        enabledBorder: OutlineInputBorder(borderSide: BorderSide(color: Colors.white)),
        focusedBorder: OutlineInputBorder(borderSide: BorderSide(color: Colors.white, width: 2)),
      ),
      items: items.map((item) => DropdownMenuItem(value: item, child: Text(item, style: TextStyle(color:
Colors.white)))).toList(),
      onChanged: onChanged,
    );
  }
}
```

OUTPUT:


11:23

Profile

Create Your Profile

Username: Ajinkya

Role: Student

What do you love about math?: maths is passion

Math Difficulty Level: Intermediate

Favorite Math Field: Algebra