National University of Modern Languages, Islamabad

LAB 3

Submitted To:	**Sir Shahid**
Submitted By:	**Tayyaba Ejaz**
Roll Number:	**2751**
Subject:	**MAD**
Due Date:	**20th October 2025**
Semester:	**8th**
Course:	**BSCS, Morning**

Faculty of Engineering and Computer Science

Table of contents

1. Lab Task 3

Form Validations

1.1 Description

This Flutter application implements a Signup and Notes Form with full input validation using regular expressions (RegEx) and the Email Validator package. The app demonstrates the use of form validation and user feedback mechanisms through snackbars.

The interface consists of two main forms:

1. Signup Form – collects user information such as:
    - Full Name (required)
    - Email Address (validated using the email_validator package)
    - Password (must contain at least 8 characters, one uppercase letter, one lowercase letter, and one number — validated via RegEx)
    - Date of Birth (validated in the US date format, e.g., APR 15, 2024)
2. On successful validation, a snackbar displays a “Signup Successful” message; otherwise, an error message prompts the user to fix invalid fields.
3. Add Note Form – allows users to add a short text note (validated using RegEx to ensure it contains at least 5 valid characters). A snackbar confirms successful submission.

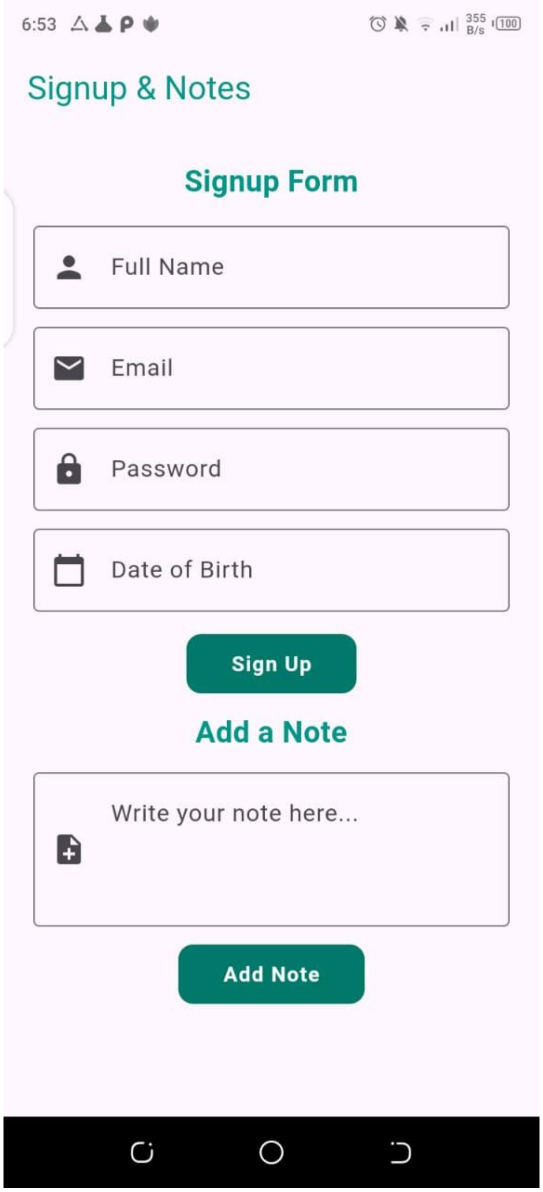
This task demonstrates best practices in Flutter form handling, input validation, and user interface design, ensuring a responsive and user-friendly experience.

1.2 Code

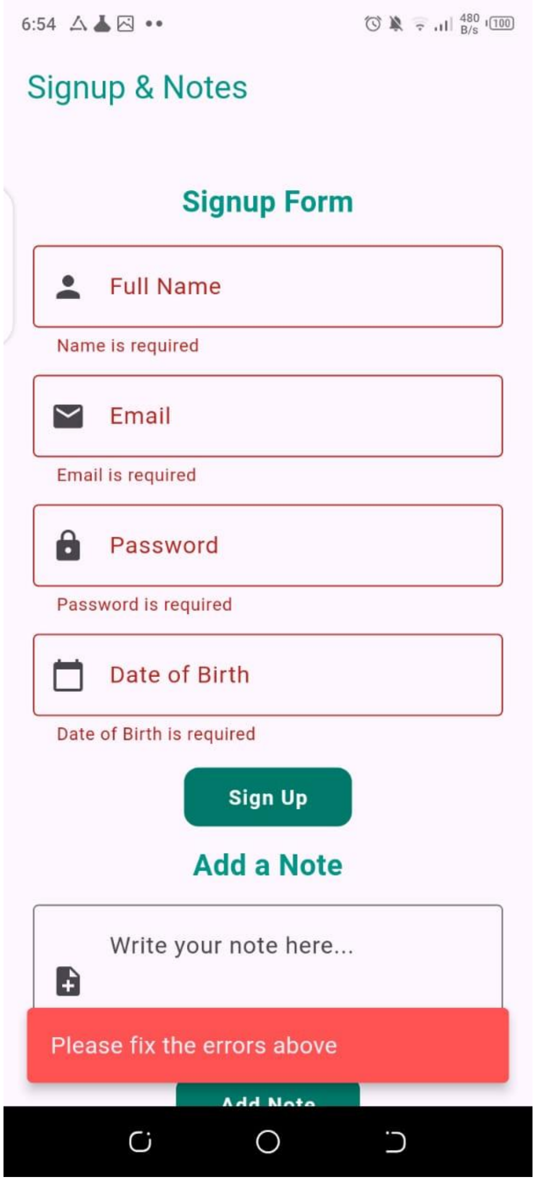
```
import 'package:flutter/material.dart';

import 'package:email_validator/email_validator.dart';


void main() {

  runApp(const MyApp());

}
```

```dart
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Signup Form',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(primarySwatch: Colors.teal),
      home: const FormScreen(),
    );
  }
}

class FormScreen extends StatefulWidget {
  const FormScreen({super.key});

  @override
  State<FormScreen> createState() => _FormScreenState();
}

class _FormScreenState extends State<FormScreen> {

  final GlobalKey<FormState> _signupFormKey = GlobalKey<FormState>();
  final GlobalKey<FormState> _noteFormKey = GlobalKey<FormState>();

  // Controllers
  final _nameCtrl = TextEditingController();
final _emailCtrl = TextEditingController();
final _passwordCtrl = TextEditingController();
final _dobCtrl = TextEditingController();
final _noteCtrl = TextEditingController();

// Regex patterns
final RegExp _passwordRegEx =
RegExp(r'^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)[A-Za-z\d]{8,}$');
final RegExp _noteRegEx = RegExp(r'^[a-zA-Z0-9\s.,!?]{5,}$');
final RegExp _usDateRegEx = RegExp(
  r'^(JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)\s(0?[1-9]|[12][0-9]|3[01]),\s\d{4}$');

String? _nameValidator(String? v) {
  if (v == null || v.isEmpty) return "Name is required";
  return null;
}

String? _emailValidator(String? v) {
  if (v == null || v.isEmpty) return "Email is required";
  if (!EmailValidator.validate(v)) return "Enter a valid email address";
  return null;
}

String? _passwordValidator(String? v) {
  if (v == null || v.isEmpty) return "Password is required";
  if (!_passwordRegEx.hasMatch(v)) {
    return "Password must have 8+ chars, uppercase, lowercase, and number";
  }
  return null;
}

String? _dobValidator(String? v) {
  if (v == null || v.isEmpty) return "Date of Birth is required";
  if (!_usDateRegEx.hasMatch(v)) return "Use format: APR 15, 2024";
  return null;
}

String? _noteValidator(String? v) {
  if (v == null || v.isEmpty) return "Note cannot be empty";
  if (!_noteRegEx.hasMatch(v)) {
    return "Note must be 5+ chars and contain letters, numbers, or punctuation";
  }
  return null;
}

void _submitSignup() {
  FocusScope.of(context).unfocus();
  if (_signupFormKey.currentState!.validate()) {
    _showMessage("Signup Successful", success: true);
    _signupFormKey.currentState!.reset();
  } else {
    _showMessage("Please fix the errors above", success: false);
  }
}

void _submitNote() {
  FocusScope.of(context).unfocus();
  if (_noteFormKey.currentState!.validate()) {
    _showMessage("Note Added Successfully", success: true);
    _noteCtrl.clear();
  } else {
    _showMessage("Please enter a valid note ", success: false);
  }
}

void _showMessage(String message, {bool success = true}) {
  ScaffoldMessenger.of(context).hideCurrentSnackBar();
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(message, style: const TextStyle(fontSize: 16)),
      backgroundColor: success ? Colors.teal : Colors.redAccent,
      duration: const Duration(seconds: 4),
      behavior: SnackBarBehavior.floating,
      margin: const EdgeInsets.all(16),
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("Signup & Notes"),
      backgroundColor: Colors.transparent,
      elevation: 0,
      foregroundColor: Colors.teal,
    ),
    body: SafeArea(
      child: SingleChildScrollView(
        padding: const EdgeInsets.all(20),
        physics: const BouncingScrollPhysics(),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [

            const Text(
              "Signup Form",
              style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
                color: Colors.teal,
              ),
              textAlign: TextAlign.center,
            ),
            const SizedBox(height: 15),
            Form(
              key: _signupFormKey,
              autovalidateMode: AutovalidateMode.onUserInteraction,
              child: Column(
                children: [
                  TextFormField(
                    controller: _nameCtrl,
      decoration: const InputDecoration(
        labelText: "Full Name",
        prefixIcon: Icon(Icons.person),
        border: OutlineInputBorder(),
      ),
      validator: _nameValidator,
    ),
    const SizedBox(height: 12),
    TextFormField(
      controller: _emailCtrl,
      decoration: const InputDecoration(
        labelText: "Email",
        hintText: "name@example.com",
        prefixIcon: Icon(Icons.email),
        border: OutlineInputBorder(),
      ),
      validator: _emailValidator,
    ),
    const SizedBox(height: 12),
    TextFormField(
      controller: _passwordCtrl,
      obscureText: true,
      decoration: const InputDecoration(
        labelText: "Password",
        hintText: "Enter your password",
        prefixIcon: Icon(Icons.lock),
        border: OutlineInputBorder(),
      ),
      validator: _passwordValidator,
    ),
    const SizedBox(height: 12),
    TextFormField(
      controller: _dobCtrl,
      decoration: const InputDecoration(
        labelText: "Date of Birth",
        hintText: "APR 15, 2024",
        prefixIcon: Icon(Icons.calendar_today),
        border: OutlineInputBorder(),
      ),
      validator: _dobValidator,
    ),
    const SizedBox(height: 15),

    Center(
      child: Material(
        color: Colors.teal.shade700,
        borderRadius: BorderRadius.circular(10),
        child: InkWell(
          splashColor: Colors.white.withOpacity(0.3),
          borderRadius: BorderRadius.circular(10),
          onTap: _submitSignup,
          child: const Padding(
            padding: EdgeInsets.symmetric(
                vertical: 10, horizontal: 30),
            child: Text(
              "Sign Up",
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 14,
                      fontWeight: FontWeight.bold,
                      letterSpacing: 0.8,
                    ),
                  ),
                ),
              ),
            ),
          ),
        ],
      ),
    ),

    const SizedBox(height: 12),

    const Text(
      "Add a Note",
      style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: Colors.teal),
      textAlign: TextAlign.center,
    ),
    const SizedBox(height: 12),
    Form(
      key: _noteFormKey,
    autovalidateMode: AutovalidateMode.onUserInteraction,
    child: Column(
      children: [
        TextFormField(
          controller: _noteCtrl,
          maxLines: 3,
          decoration: const InputDecoration(
            hintText: "Write your note here...",
            prefixIcon: Icon(Icons.note_add),
            border: OutlineInputBorder(),
          ),
          validator: _noteValidator,
        ),
        const SizedBox(height: 12),

        Center(
          child: Material(
            color: Colors.teal.shade700,
            borderRadius: BorderRadius.circular(10),
            child: InkWell(
              splashColor: Colors.white.withOpacity(0.3),
              borderRadius: BorderRadius.circular(10),
              onTap: _submitNote,
              child: const Padding(
                padding: EdgeInsets.symmetric(
                    vertical: 10, horizontal: 30),
                child: Text(
                  "Add Note",
                                  style: TextStyle(
                                    color: Colors.white,
                                    fontSize: 14,
                                    fontWeight: FontWeight.bold,
                                    letterSpacing: 0.8,
                                  ),
                                ),
                              ),
                            ),
                          ),
                        ),
                      ],
                    ),
                  ),
                ],
              ),
            ),
          ),
        );
      }
    }
```

1.3 Screenshots



Full application view



Name is required

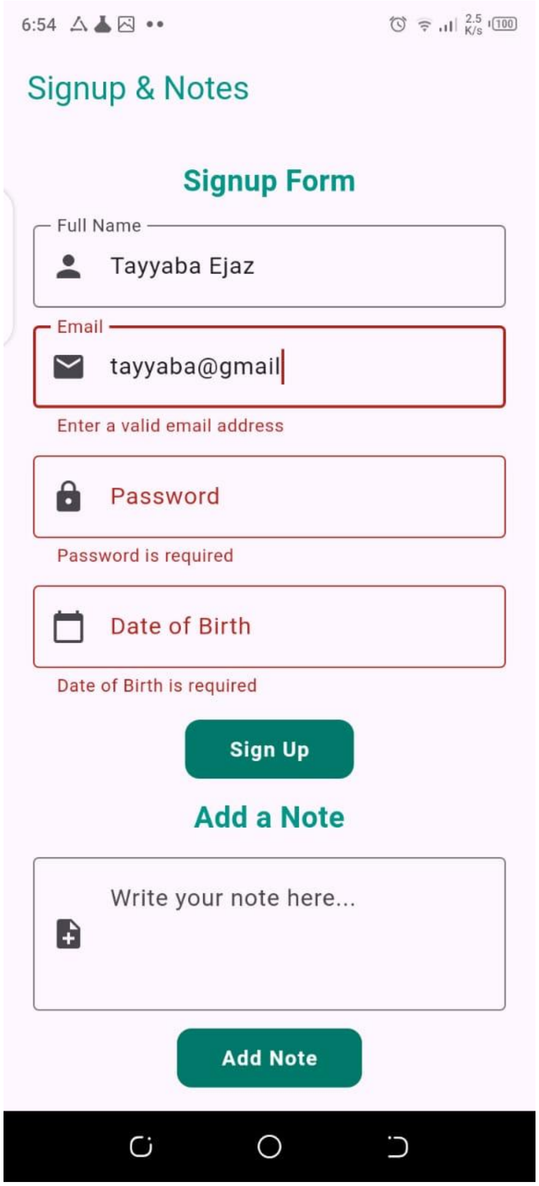
Email is required

Password is required

Date of Birth is required

Please fix the errors above

Errors on empty form submission



6:54

Signup & Notes

Signup Form

Full Name: Tayyaba Ejaz

Email: tayyaba@gmail

Enter a valid email address

Password

Password is required

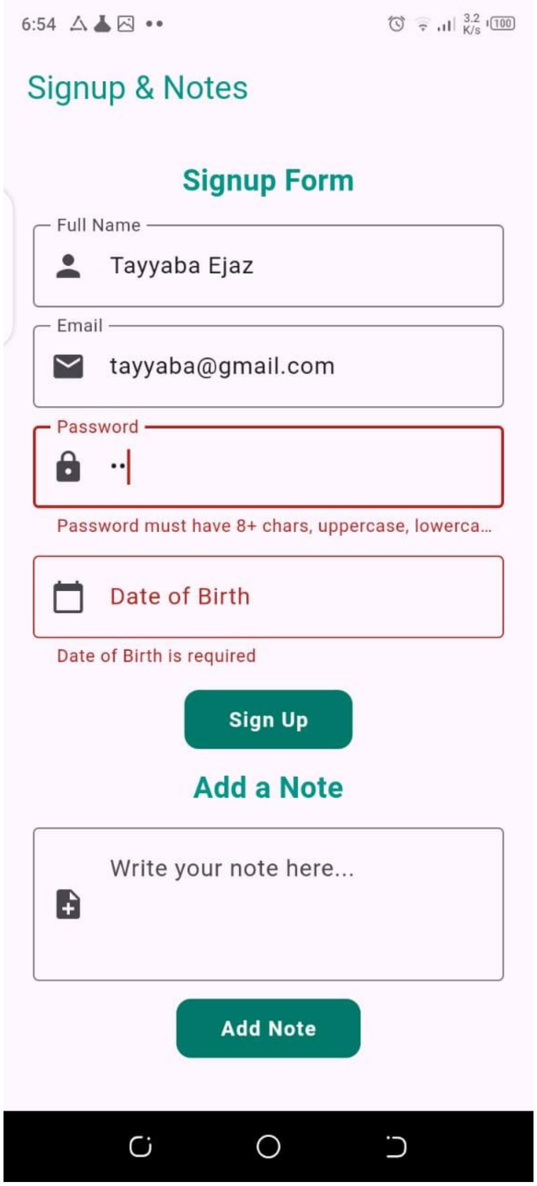
Date of Birth

Date of Birth is required

Add a Note

Write your note here...

Email validator method is giving error email field to enter valid email address



6:54

Signup & Notes

Signup Form

Full Name: Tayyaba Ejaz

Email: tayyaba@gmail.com

Password: ••

Password must have 8+ chars, uppercase, lowerca...

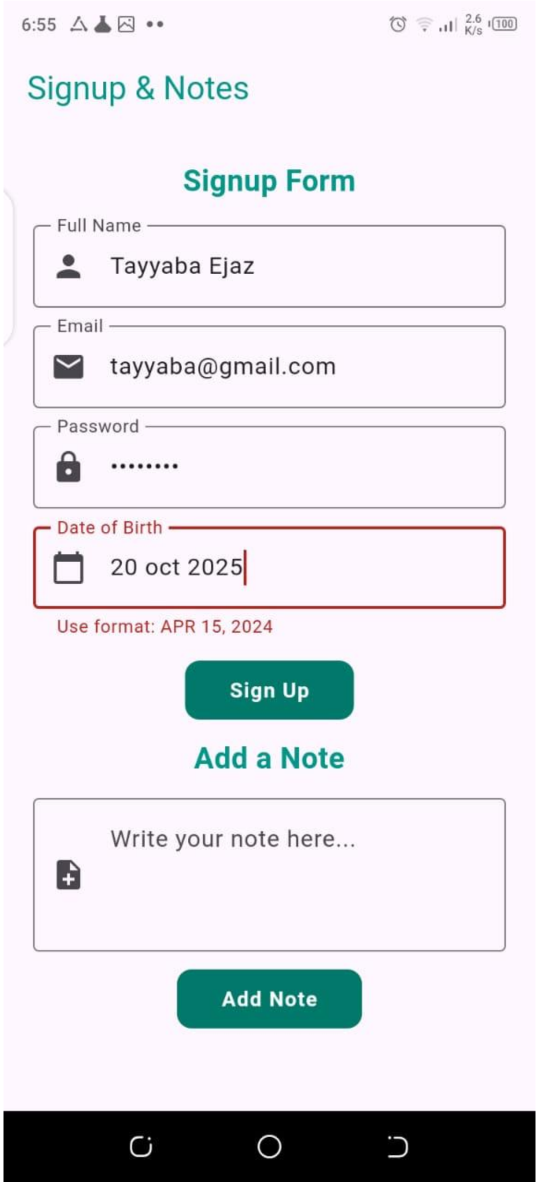
Date of Birth

Date of Birth is required

Add a Note

Write your note here...

Password validator is giving error to enter the specific password using regix



Signup & Notes

Signup Form

Full Name: Tayyaba Ejaz

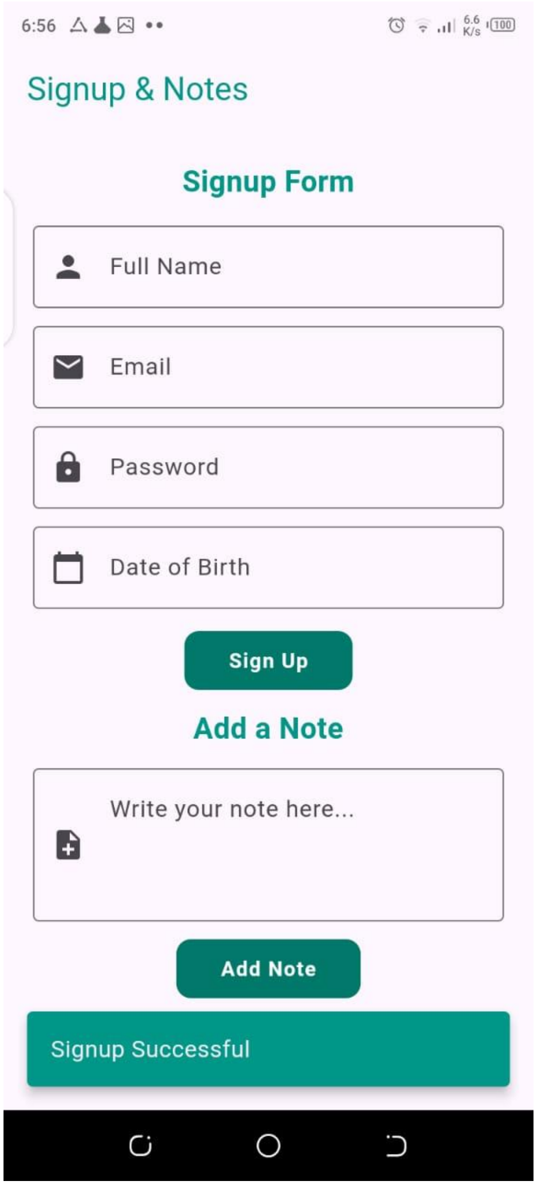
Email: tayyaba@gmail.com

Password: ••••••••

Date of Birth: 20 oct 2025

Use format: APR 15, 2024

GET Request is fetching the data

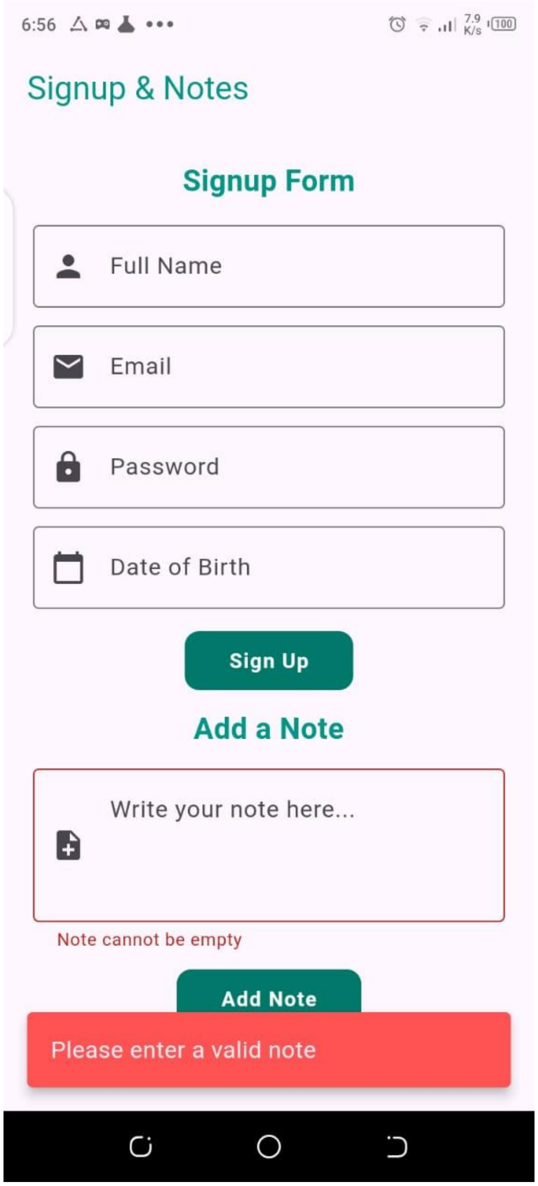


Signup & Notes

Signup Form

Signup Successful

Form submitted


Signup & Notes

Signup Form

Full Name

Email

Password

Date of Birth

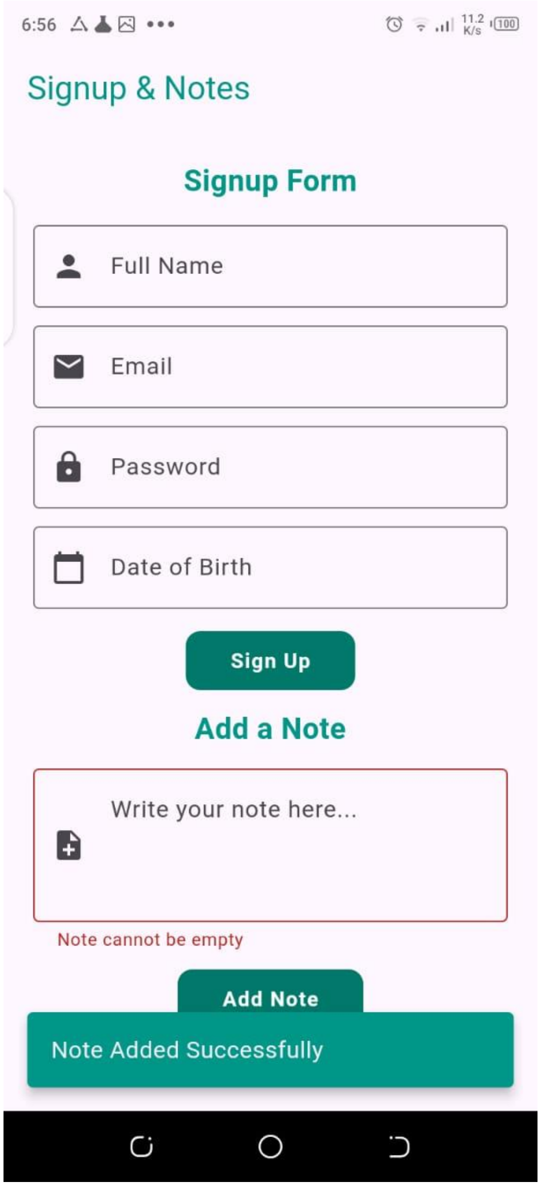
Add a Note

Write your note here...

Note cannot be empty

Please enter a valid note

Validation on notes this field can’t be empty


Signup & Notes

Signup Form

Full Name

Email

Password

Date of Birth

Add a Note

Write your note here...

Note cannot be empty

Note Added Successfully

Note added