**Assignment 2**

**Mobile Application Development (Theory)**



**Submitted to Mr. Muhammad Ahmad  
Submitted by Addan Abdullah**

**Roll #: SU92-BSSEM-F22-059  
Section: SE-6B**

**Submitted on Feb 12th, 2025**

**DEPARTMENT OF SOFTWARE ENGINEERING  
SUPERIOR UNIVERSITY, LAHORE**

Contents

[GitHub Repository 3](#_Toc190220110)

[Question 1 4](#_Toc190220111)

[Screenshots 4](#_Toc190220112)

[Code 8](#_Toc190220113)

[Question 2 19](#_Toc190220114)

[Screenshots 19](#_Toc190220115)

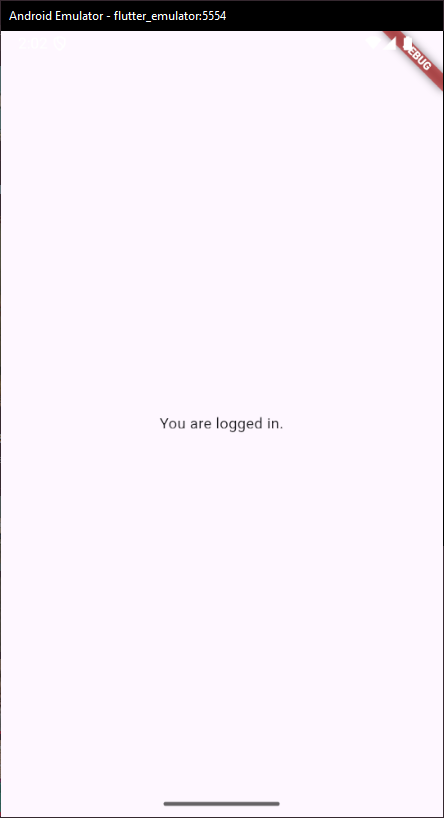
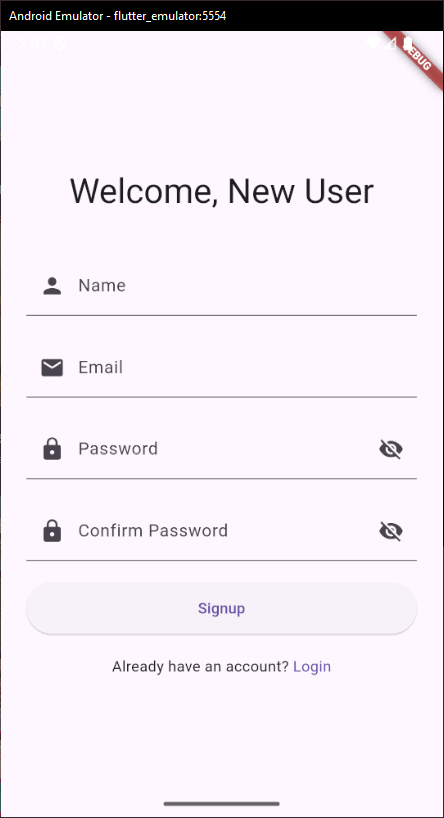
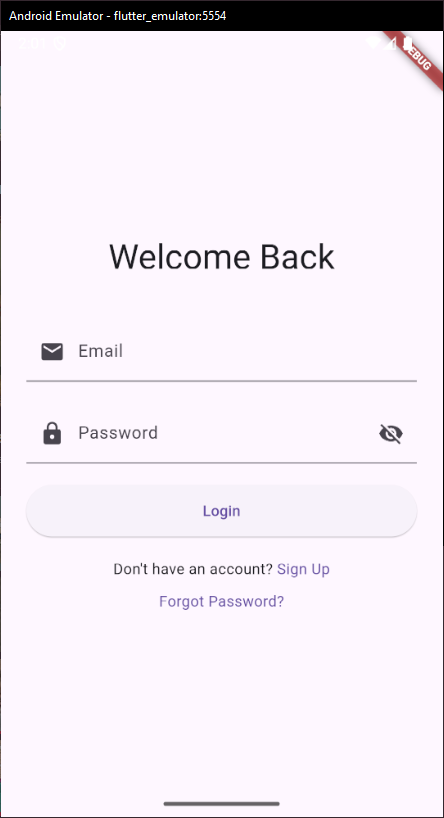
[Code 36](#_Toc190220116)

# GitHub Repository

[GitHub Repository Link](https://github.com/AA-Siddiqui/Assignment-2-MADT-S25)

# Question 1

## Screenshots



## Code

To make a splash screen, install flutter\_native\_splash package. Configure a YAML file with your splash screen attributes like color, logo, background image, branding, etc. and run “dart run flutter\_native\_splash:create –path=flutter\_native\_splash.yaml”.

import 'package:assignment\_2\_q\_1/pages/login\_page.dart';

import 'package:flutter/material.dart';

import 'package:flutter\_native\_splash/flutter\_native\_splash.dart';

void main() {

  WidgetsBinding widgetsBinding = WidgetsFlutterBinding.ensureInitialized();

  FlutterNativeSplash.preserve(widgetsBinding: widgetsBinding);

  runApp(const MainApp());

}

class MainApp extends StatelessWidget {

  const MainApp({super.key});

  @override

  Widget build(BuildContext context) {

    return const MaterialApp(

      home: LoginScreen(),

    );

  }

}

import 'package:flutter\_native\_splash/flutter\_native\_splash.dart';

import 'package:assignment\_2\_q\_1/pages/logged\_in\_page.dart';

import 'package:assignment\_2\_q\_1/pages/signup\_page.dart';

import 'package:flutter/material.dart';

class LoginScreen extends StatefulWidget {

  const LoginScreen({super.key});

  @override

  State<LoginScreen> createState() => \_LoginScreenState();

}

class \_LoginScreenState extends State<LoginScreen> {

  final \_formKey = GlobalKey<FormState>();

  final \_emailController = TextEditingController();

  final \_passwordController = TextEditingController();

  bool \_obscurePassword = true;

  bool \_isLoading = false;

  @override

  void initState() {

    super.initState();

    FlutterNativeSplash.remove();

  }

  void \_togglePasswordVisibility() {

    setState(() {

      \_obscurePassword = !\_obscurePassword;

    });

  }

  Future<void> \_handleLogin() async {

    if (!\_formKey.currentState!.validate()) return;

    setState(() {

      \_isLoading = true;

    });

    await Future.delayed(Duration(seconds: 1));

    late final String dialogTitle;

    late final String dialogBody;

    if (\_emailController.text == "user@example.com" &&

        \_passwordController.text == "password") {

      dialogTitle = "Success";

      dialogBody = "Logged in successfully!";

    } else {

      dialogTitle = "Error";

      dialogBody = "Invalid email or password";

    }

    if (mounted) {

      showDialog(

        context: context,

        builder: (dialogContext) {

          return AlertDialog(

            title: Text(dialogTitle),

            content: Text(dialogBody),

            actions: [

              TextButton(

                onPressed: () {

                  Navigator.pop(context);

                  if (dialogTitle == "Success") {

                    Navigator.push(

                      context,

                      MaterialPageRoute(

                        builder: (context) => LoggedInScreen(),

                      ),

                    );

                  }

                },

                child: Text("OK"),

              )

            ],

          );

        },

      );

    }

    setState(() => \_isLoading = false);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Padding(

        padding: const EdgeInsets.all(24.0),

        child: Center(

          child: ConstrainedBox(

            constraints: BoxConstraints(maxWidth: 400),

            child: SingleChildScrollView(

              child: Form(

                key: \_formKey,

                child: Column(

                  mainAxisSize: MainAxisSize.min,

                  children: [

                    Text(

                      "Welcome Back",

                      style: Theme.of(context).textTheme.headlineLarge,

                    ),

                    SizedBox(height: 40),

                    TextFormField(

                      controller: \_emailController,

                      decoration: InputDecoration(

                        labelText: "Email",

                        prefixIcon: Icon(Icons.email),

                      ),

                      keyboardType: TextInputType.emailAddress,

                      validator: (value) {

                        if (value == null || value.isEmpty) {

                          return "Please enter your email";

                        }

                        if (!RegExp(r'^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$')

                            .hasMatch(value)) {

                          return "Please enter a valid email";

                        }

                        return null;

                      },

                    ),

                    SizedBox(height: 20),

                    TextFormField(

                      controller: \_passwordController,

                      decoration: InputDecoration(

                        labelText: "Password",

                        prefixIcon: Icon(Icons.lock),

                        suffixIcon: IconButton(

                          icon: Icon(\_obscurePassword

                              ? Icons.visibility\_off

                              : Icons.visibility),

                          onPressed: \_togglePasswordVisibility,

                        ),

                      ),

                      obscureText: \_obscurePassword,

                      validator: (value) {

                        if (value == null || value.isEmpty) {

                          return "Please enter your password";

                        }

                        if (value.length < 6) {

                          return "Password must be at least 6 characters";

                        }

                        return null;

                      },

                    ),

                    SizedBox(height: 20),

                    SizedBox(

                      width: double.infinity,

                      child: \_isLoading

                          ? Center(child: CircularProgressIndicator())

                          : ElevatedButton(

                              onPressed: \_handleLogin,

                              child: Padding(

                                padding: const EdgeInsets.all(14.0),

                                child: Text("Login"),

                              ),

                            ),

                    ),

                    SizedBox(height: 20),

                    Column(

                      mainAxisAlignment: MainAxisAlignment.spaceBetween,

                      mainAxisSize: MainAxisSize.max,

                      children: [

                        Row(

                          mainAxisAlignment: MainAxisAlignment.center,

                          children: [

                            Text("Don't have an account? "),

                            MouseRegion(

                              cursor: SystemMouseCursors.click,

                              child: GestureDetector(

                                onTap: () => Navigator.push(

                                  context,

                                  MaterialPageRoute(

                                    builder: (context) => SignUpScreen(),

                                  ),

                                ),

                                child: Text(

                                  "Sign Up",

                                  style: TextStyle(

                                    color:

                                        Theme.of(context).colorScheme.primary,

                                  ),

                                ),

                              ),

                            ),

                          ],

                        ),

                        SizedBox(height: 10),

                        MouseRegion(

                          cursor: SystemMouseCursors.click,

                          child: GestureDetector(

                            onTap: () => Placeholder(),

                            child: Text(

                              "Forgot Password?",

                              style: TextStyle(

                                color: Theme.of(context).colorScheme.primary,

                              ),

                            ),

                          ),

                        ),

                      ],

                    ),

                  ],

                ),

              ),

            ),

          ),

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class SignUpScreen extends StatefulWidget {

  const SignUpScreen({super.key});

  @override

  State<SignUpScreen> createState() => \_SignUpScreenState();

}

class \_SignUpScreenState extends State<SignUpScreen> {

  final \_formKey = GlobalKey<FormState>();

  final \_emailController = TextEditingController();

  final \_passwordController = TextEditingController();

  final \_confirmPasswordController = TextEditingController();

  bool \_obscurePassword = true;

  bool \_isLoading = false;

  void \_togglePasswordVisibility() {

    setState(() {

      \_obscurePassword = !\_obscurePassword;

    });

  }

  Future<void> \_handleLogin() async {

    if (!\_formKey.currentState!.validate()) return;

    setState(() {

      \_isLoading = true;

    });

    await Future.delayed(Duration(seconds: 1));

    late final String dialogTitle;

    late final String dialogBody;

    if (\_emailController.text == "user@example.com" &&

        \_passwordController.text == "password") {

      dialogTitle = "Success";

      dialogBody = "Logged in successfully!";

    } else {

      dialogTitle = "Error";

      dialogBody = "Invalid email or password";

    }

    if (mounted) {

      showDialog(

        context: context,

        builder: (dialogContext) {

          return AlertDialog(

            title: Text(dialogTitle),

            content: Text(dialogBody),

            actions: [

              TextButton(

                onPressed: () => Navigator.pop(context),

                child: Text("OK"),

              )

            ],

          );

        },

      );

    }

    setState(() => \_isLoading = false);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Padding(

        padding: const EdgeInsets.all(24.0),

        child: Center(

          child: ConstrainedBox(

            constraints: BoxConstraints(maxWidth: 400),

            child: SingleChildScrollView(

              child: Form(

                key: \_formKey,

                child: Column(

                  mainAxisSize: MainAxisSize.min,

                  children: [

                    Text(

                      "Welcome, New User",

                      style: Theme.of(context).textTheme.headlineLarge,

                    ),

                    SizedBox(height: 40),

                    TextFormField(

                      controller: \_emailController,

                      decoration: InputDecoration(

                        labelText: "Name",

                        prefixIcon: Icon(Icons.person),

                      ),

                      validator: (value) {

                        return null;

                      },

                    ),

                    SizedBox(height: 20),

                    TextFormField(

                      controller: \_emailController,

                      decoration: InputDecoration(

                        labelText: "Email",

                        prefixIcon: Icon(Icons.email),

                      ),

                      keyboardType: TextInputType.emailAddress,

                      validator: (value) {

                        if (value == null || value.isEmpty) {

                          return "Please enter your email";

                        }

                        if (!RegExp(r'^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$')

                            .hasMatch(value)) {

                          return "Please enter a valid email";

                        }

                        return null;

                      },

                    ),

                    SizedBox(height: 20),

                    TextFormField(

                      controller: \_passwordController,

                      decoration: InputDecoration(

                        labelText: "Password",

                        prefixIcon: Icon(Icons.lock),

                        suffixIcon: IconButton(

                          icon: Icon(\_obscurePassword

                              ? Icons.visibility\_off

                              : Icons.visibility),

                          onPressed: \_togglePasswordVisibility,

                        ),

                      ),

                      obscureText: \_obscurePassword,

                      validator: (value) {

                        if (value == null || value.isEmpty) {

                          return "Please enter your password";

                        }

                        if (value.length < 6) {

                          return "Password must be at least 6 characters";

                        }

                        return null;

                      },

                    ),

                    SizedBox(height: 20),

                    TextFormField(

                      controller: \_confirmPasswordController,

                      decoration: InputDecoration(

                        labelText: "Confirm Password",

                        prefixIcon: Icon(Icons.lock),

                        suffixIcon: IconButton(

                          icon: Icon(\_obscurePassword

                              ? Icons.visibility\_off

                              : Icons.visibility),

                          onPressed: \_togglePasswordVisibility,

                        ),

                      ),

                      obscureText: \_obscurePassword,

                      validator: (value) {

                        if (value == null || value.isEmpty) {

                          return "Please enter your password";

                        }

                        if (value.length < 6) {

                          return "Password must be at least 6 characters";

                        }

                        return null;

                      },

                    ),

                    SizedBox(height: 20),

                    SizedBox(

                      width: double.infinity,

                      child: \_isLoading

                          ? Center(child: CircularProgressIndicator())

                          : ElevatedButton(

                              onPressed: \_handleLogin,

                              child: Padding(

                                padding: const EdgeInsets.all(14.0),

                                child: Text("Signup"),

                              ),

                            ),

                    ),

                    SizedBox(height: 20),

                    Column(

                      mainAxisAlignment: MainAxisAlignment.spaceBetween,

                      mainAxisSize: MainAxisSize.max,

                      children: [

                        Row(

                          mainAxisAlignment: MainAxisAlignment.center,

                          children: [

                            Text("Already have an account? "),

                            MouseRegion(

                              cursor: SystemMouseCursors.click,

                              child: GestureDetector(

                                onTap: () => Navigator.pop(context),

                                child: Text(

                                  "Login",

                                  style: TextStyle(

                                    color:

                                        Theme.of(context).colorScheme.primary,

                                  ),

                                ),

                              ),

                            ),

                          ],

                        ),

                      ],

                    ),

                  ],

                ),

              ),

            ),

          ),

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class LoggedInScreen extends StatelessWidget {

  const LoggedInScreen({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Center(

        child: Text("You are logged in."),

      ),

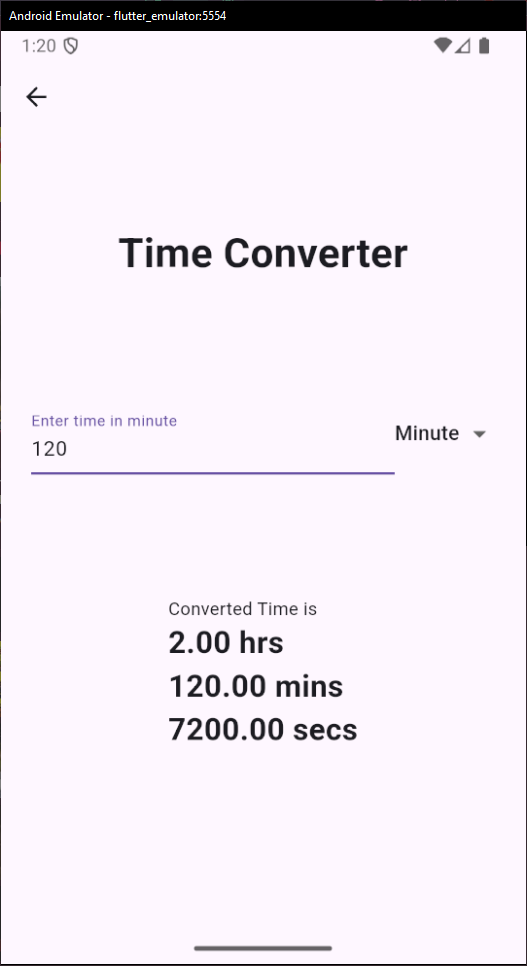
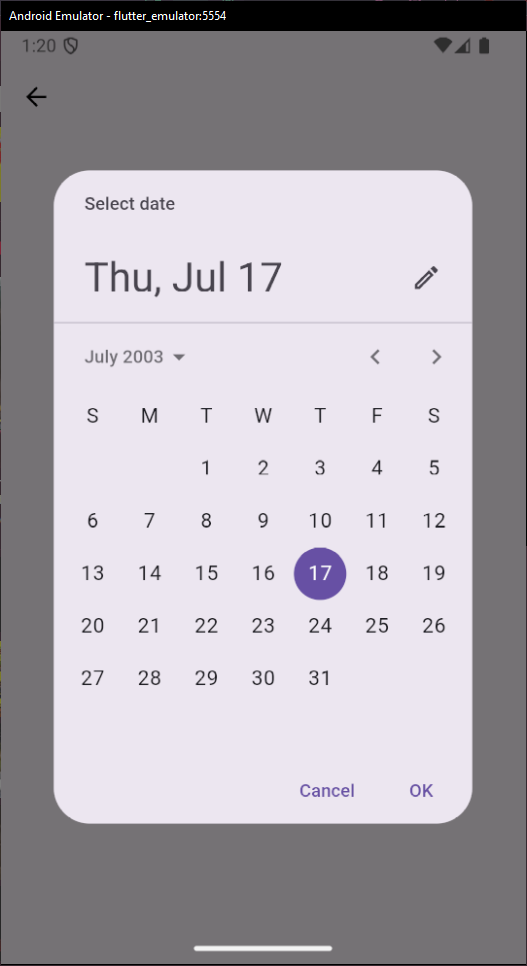
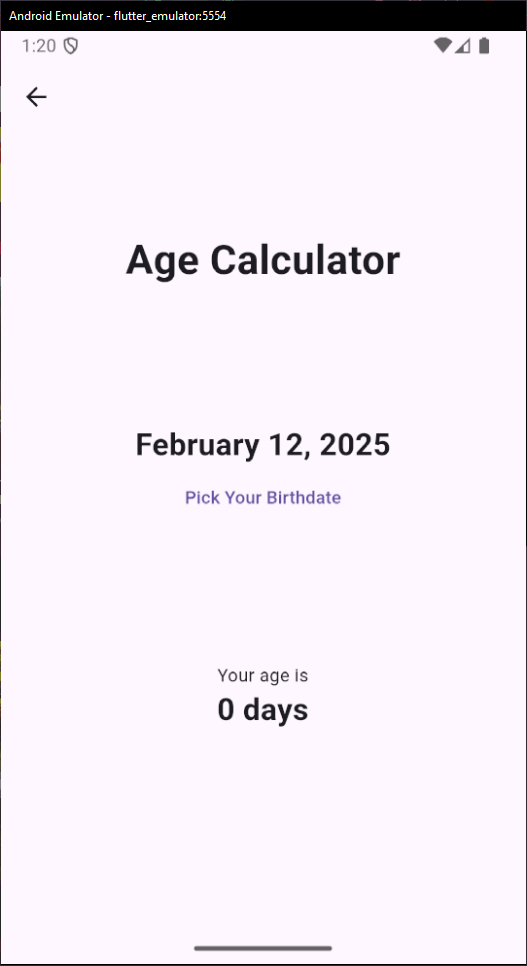
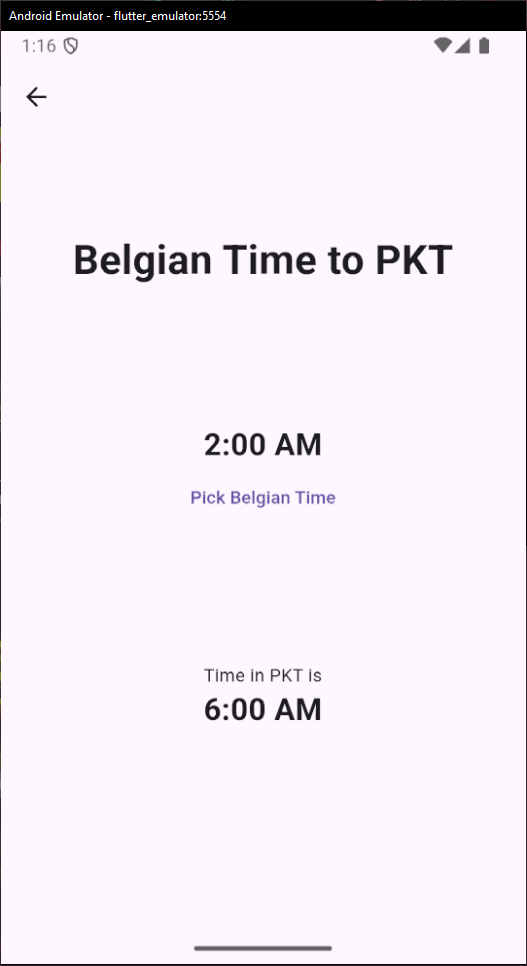
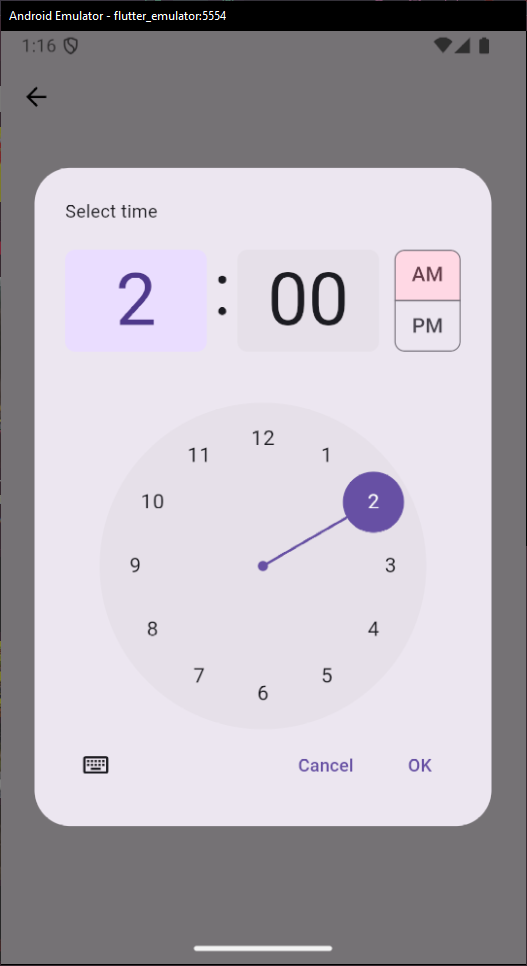
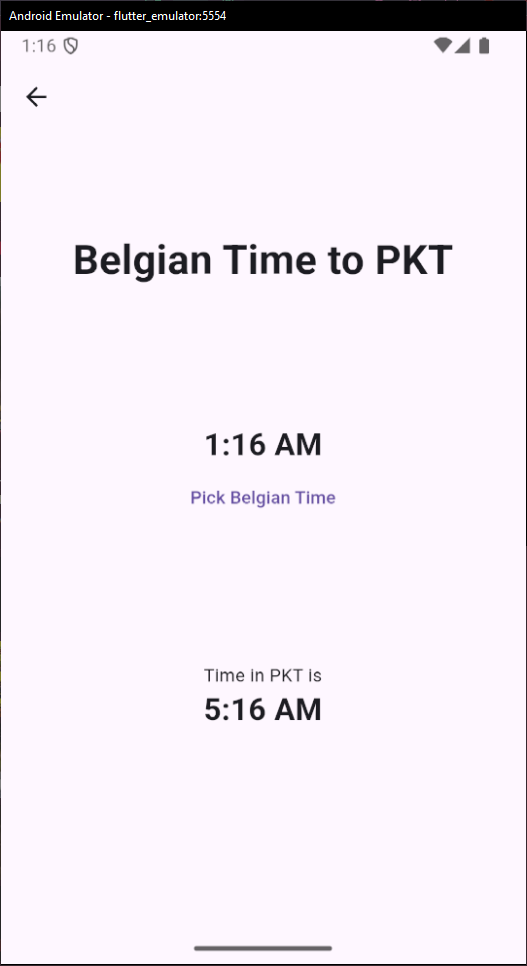
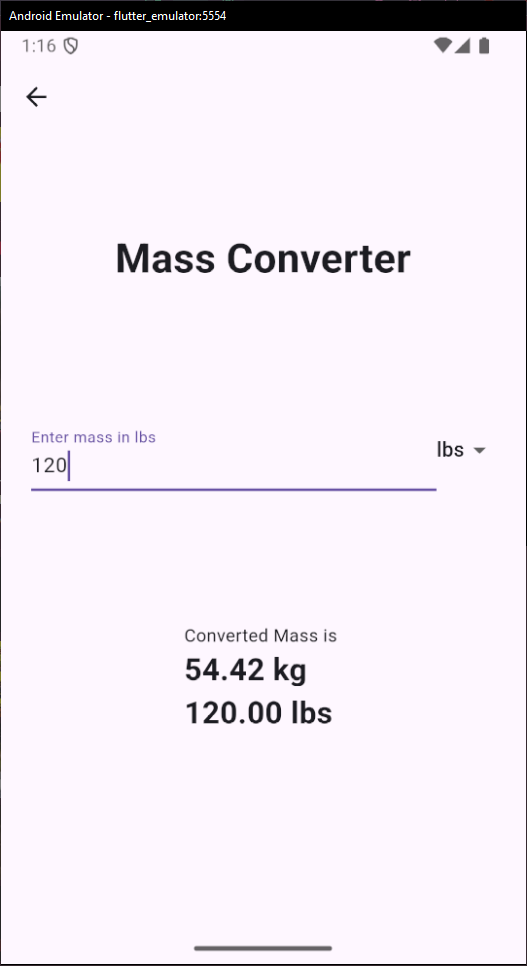
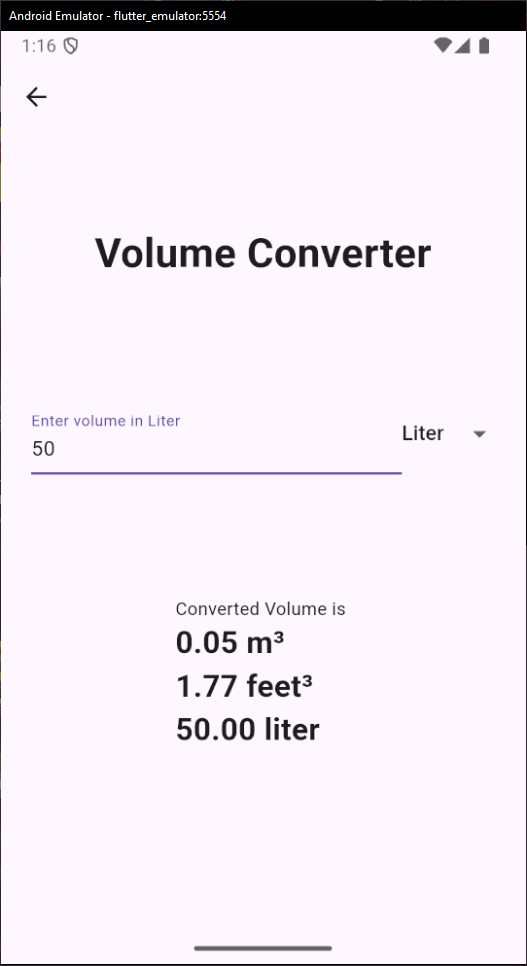
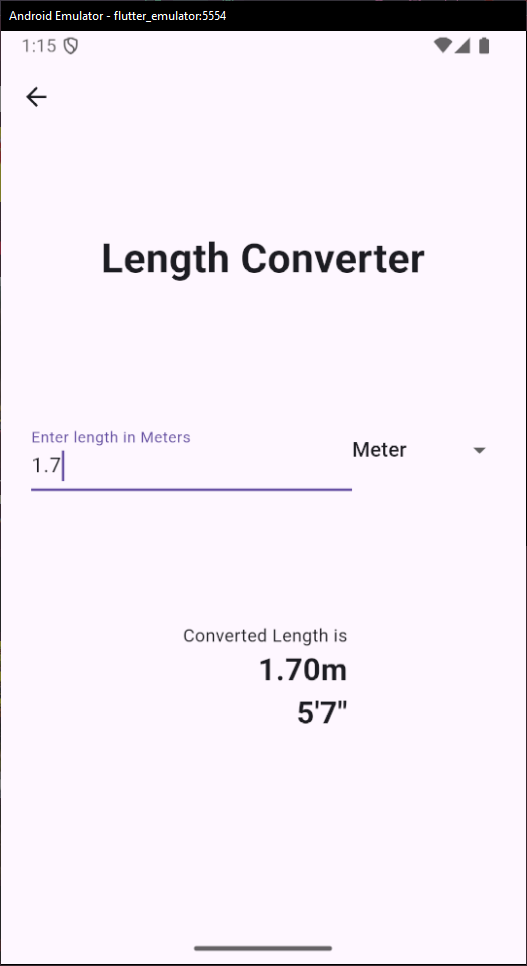
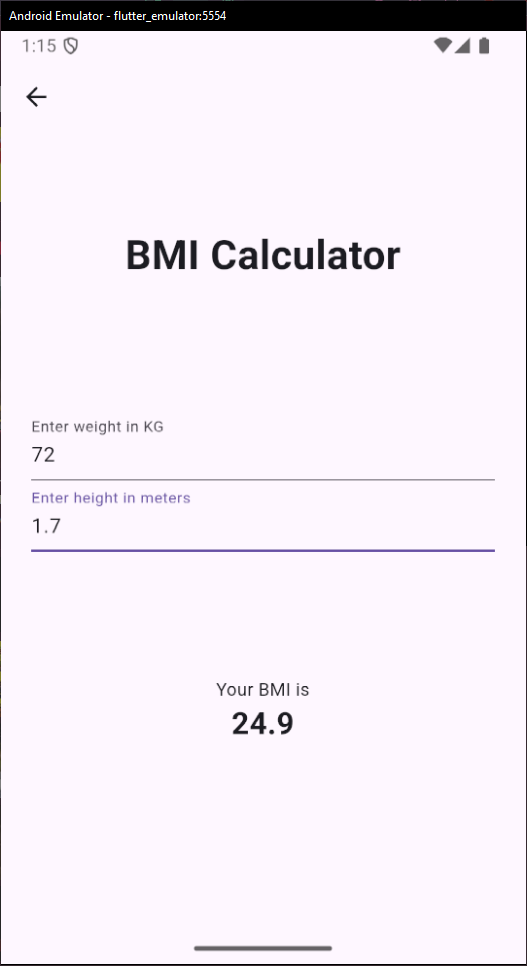
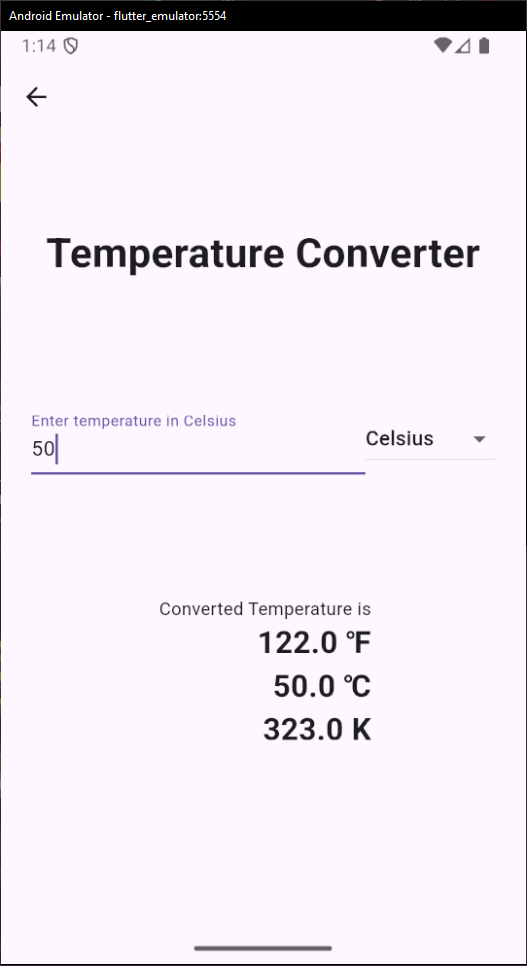
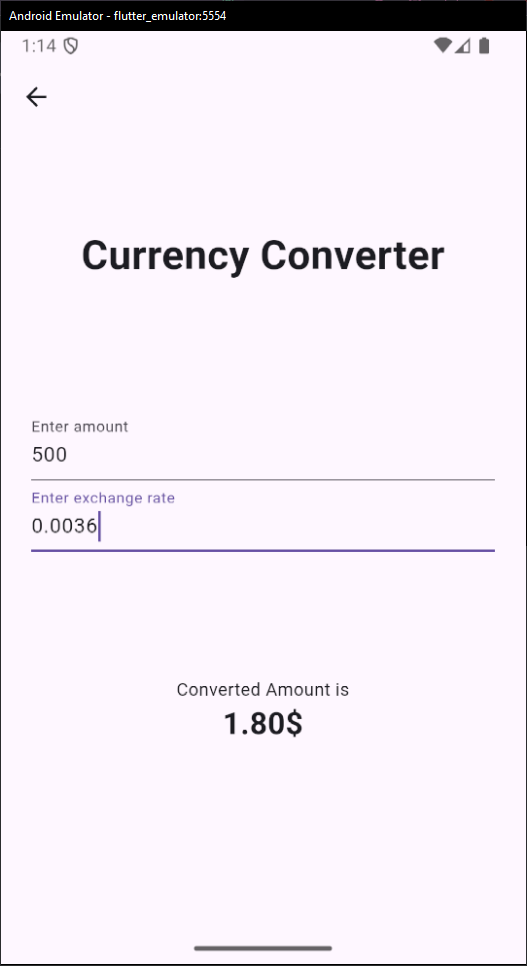
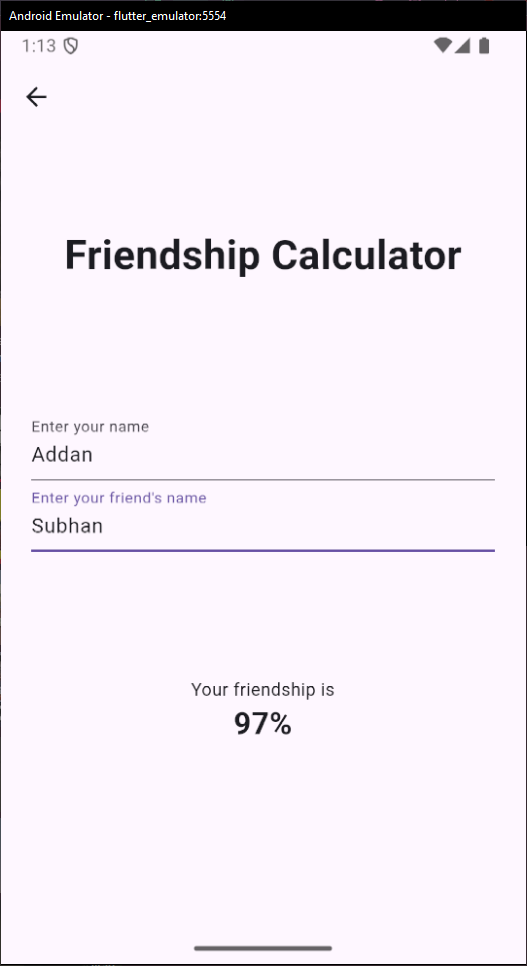
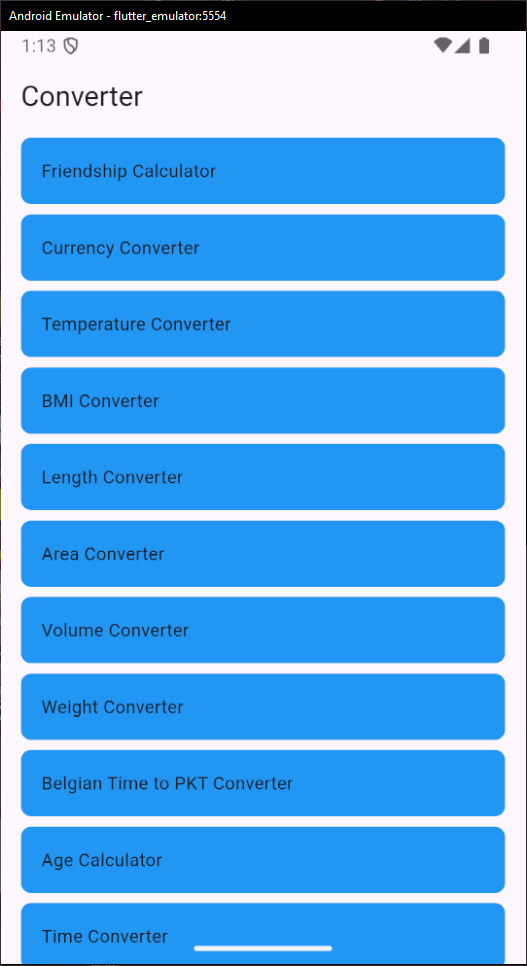
    );

  }

}

# Question 2

## Screenshots



## Code

import 'package:assignment\_2\_q\_2/pages/home\_page.dart';

import 'package:flutter/material.dart';

void main() {

  runApp(const MainApp());

}

class MainApp extends StatelessWidget {

  const MainApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: HomePage(),

    );

  }

}

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/age\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/area\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/bmi\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/currency\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/friendship\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/length\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/temperature\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/time\_b\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/time\_h\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/volume\_page.dart';

import 'package:assignment\_2\_q\_2/pages/calculator\_pages/weight\_page.dart';

import 'package:flutter/material.dart';

class Page {

  final String title;

  final Widget page;

  const Page({

    required this.title,

    required this.page,

  });

}

class HomePage extends StatelessWidget {

  HomePage({

    super.key,

  });

  final List<Page> pageList = [

    Page(title: "Friendship Calculator", page: FriendshipPage()),

    Page(title: "Currency Converter", page: CurrencyPage()),

    Page(title: "Temperature Converter", page: TemperaturePage()),

    Page(title: "BMI Converter", page: BmiPage()),

    Page(title: "Length Converter", page: LengthPage()),

    Page(title: "Area Converter", page: AreaPage()),

    Page(title: "Volume Converter", page: VolumePage()),

    Page(title: "Weight Converter", page: WeightPage()),

    Page(title: "Belgian Time to PKT Converter", page: TimeOfDayPage()),

    Page(title: "Age Calculator", page: AgePage()),

    Page(title: "Time Converter", page: TimePage()),

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("Converter"),

      ),

      body: ListView.builder(

        itemCount: pageList.length,

        itemBuilder: (context, index) => GestureDetector(

          onTap: () {

            Navigator.push(

              context,

              MaterialPageRoute(builder: (context) => pageList[index].page),

            );

          },

          child: Padding(

            padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 4),

            child: Container(

              padding: EdgeInsets.all(16),

              decoration: BoxDecoration(

                color: Colors.blue,

                borderRadius: BorderRadius.circular(8),

              ),

              child: Text(

                pageList[index].title,

              ),

            ),

          ),

        ),

      ),

    );

  }

}

import 'dart:math';

import 'package:flutter/material.dart';

class FriendshipPage extends StatefulWidget {

  const FriendshipPage({super.key});

  @override

  State<FriendshipPage> createState() => \_FriendshipPageState();

}

class \_FriendshipPageState extends State<FriendshipPage> {

  int \_friendship = 0;

  void \_calculateFriendship(String? value) {

    setState(() {

      \_friendship = Random().nextInt(30) + 70;

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Friendship Calculator",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Column(

              children: [

                TextField(

                  decoration: InputDecoration(

                    labelText: "Enter your name",

                  ),

                  onChanged: \_calculateFriendship,

                ),

                TextField(

                  decoration: InputDecoration(

                    labelText: "Enter your friend's name",

                  ),

                  onChanged: \_calculateFriendship,

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.center,

              TextSpan(

                text: "Your friendship is \n",

                children: [

                  TextSpan(

                    text: \_friendship.toString(),

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                    children: [

                      TextSpan(text: "%"),

                    ],

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class CurrencyPage extends StatefulWidget {

  const CurrencyPage({super.key});

  @override

  State<CurrencyPage> createState() => \_CurrencyPageState();

}

class \_CurrencyPageState extends State<CurrencyPage> {

  double \_enteredAmount = 0;

  double \_exchangeRate = 280;

  double \_convertedAmount = 0;

  void \_convertAmount() {

    \_convertedAmount = \_enteredAmount \* \_exchangeRate;

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Currency Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Column(

              children: [

                TextField(

                  keyboardType: TextInputType.number,

                  decoration: InputDecoration(

                    labelText: "Enter amount",

                  ),

                  onChanged: (String? value) {

                    setState(() {

                      \_enteredAmount = double.tryParse(value ?? "0") ?? 0;

                      \_convertAmount();

                    });

                  },

                ),

                TextField(

                  keyboardType: TextInputType.number,

                  decoration: InputDecoration(

                    labelText: "Enter exchange rate",

                  ),

                  onChanged: (String? value) {

                    setState(() {

                      \_exchangeRate = double.tryParse(value ?? "0") ?? 0;

                      \_convertAmount();

                    });

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.center,

              TextSpan(

                text: "Converted Amount is \n",

                children: [

                  TextSpan(

                    text: \_convertedAmount.toStringAsFixed(2),

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                    children: [

                      TextSpan(text: "\$"),

                    ],

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class TemperaturePage extends StatefulWidget {

  const TemperaturePage({super.key});

  @override

  State<TemperaturePage> createState() => \_TemperaturePageState();

}

class \_TemperaturePageState extends State<TemperaturePage> {

  double \_tempInF = 0;

  double \_tempInC = 0;

  int \_index = 0;

  double \_enteredValue = 0;

  void \_updateTemp() {

    setState(() {

      switch (\_index) {

        case 0:

          \_tempInC = \_enteredValue;

          \_tempInF = (\_enteredValue \* 9 / 5) + 32;

          break;

        case 1:

          \_tempInF = \_enteredValue;

          \_tempInC = (\_enteredValue - 32) \* 5 / 9;

          break;

        case 2:

          \_tempInC = \_enteredValue - 273;

          \_tempInF = (\_tempInC \* 9 / 5) + 32;

          break;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Temperature Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Row(

              crossAxisAlignment: CrossAxisAlignment.center,

              mainAxisSize: MainAxisSize.min,

              children: [

                Expanded(

                  child: TextField(

                    keyboardType: TextInputType.number,

                    decoration: InputDecoration(

                      labelText: "Enter temperature in ${[

                        "Celsius",

                        "Farenheit",

                        "Kelvin"

                      ][\_index]}",

                    ),

                    onChanged: (String? value) {

                      \_enteredValue = double.tryParse(value ?? "0") ?? 0;

                      \_updateTemp();

                    },

                  ),

                ),

                DropdownButton(

                  value: \_index,

                  items: [

                    DropdownMenuItem(

                      value: 0,

                      child: Text("Celsius"),

                    ),

                    DropdownMenuItem(

                      value: 1,

                      child: Text("Fahrenheit"),

                    ),

                    DropdownMenuItem(

                      value: 2,

                      child: Text("Kelvin"),

                    ),

                  ],

                  onChanged: (index) {

                    \_index = index ?? 0;

                    \_updateTemp();

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.end,

              TextSpan(

                text: "Converted Temperature is \n",

                children: [

                  TextSpan(

                    text:

                        "${\_tempInF.toStringAsFixed(1)} ℉\n${\_tempInC.toStringAsFixed(1)} ℃\n${(\_tempInC + 273).toStringAsFixed(1)} K",

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class BmiPage extends StatefulWidget {

  const BmiPage({super.key});

  @override

  State<BmiPage> createState() => \_BmiPageState();

}

class \_BmiPageState extends State<BmiPage> {

  double \_weightInKg = 0;

  double \_heightInMeter = 280;

  double \_bmi = 0;

  void \_calculateBMI() {

    \_bmi = \_weightInKg / (\_heightInMeter \* \_heightInMeter);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "BMI Calculator",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Column(

              children: [

                TextField(

                  keyboardType: TextInputType.number,

                  decoration: InputDecoration(

                    labelText: "Enter weight in KG",

                  ),

                  onChanged: (String? value) {

                    setState(() {

                      \_weightInKg = double.tryParse(value ?? "0") ?? 0;

                      \_calculateBMI();

                    });

                  },

                ),

                TextField(

                  keyboardType: TextInputType.number,

                  decoration: InputDecoration(

                    labelText: "Enter height in meters",

                  ),

                  onChanged: (String? value) {

                    setState(() {

                      \_heightInMeter = double.tryParse(value ?? "0") ?? 0;

                      \_calculateBMI();

                    });

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.center,

              TextSpan(

                text: "Your BMI is \n",

                children: [

                  TextSpan(

                    text: \_bmi.toStringAsFixed(1),

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class LengthPage extends StatefulWidget {

  const LengthPage({super.key});

  @override

  State<LengthPage> createState() => \_LengthPageState();

}

class \_LengthPageState extends State<LengthPage> {

  double \_lengthInMeter = 0;

  double \_lengthInInches = 0;

  int \_index = 0;

  double \_enteredValue = 0;

  double \_enteredInches = 0;

  void \_updateLength() {

    setState(() {

      switch (\_index) {

        case 0:

          \_lengthInMeter = \_enteredValue;

          \_lengthInInches = \_enteredValue \* 39.37;

          break;

        case 1:

          \_lengthInInches = (\_enteredValue \* 12) + \_enteredInches;

          \_lengthInMeter = \_lengthInInches / 39.37;

          break;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Length Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Row(

              crossAxisAlignment: CrossAxisAlignment.start,

              mainAxisSize: MainAxisSize.min,

              children: [

                Expanded(

                  child: Column(

                    children: [

                      TextField(

                        keyboardType: TextInputType.number,

                        decoration: InputDecoration(

                          labelText: "Enter length in ${[

                            "Meters",

                            "Feet",

                          ][\_index]}",

                        ),

                        onChanged: (String? value) {

                          \_enteredValue = double.tryParse(value ?? "0") ?? 0;

                          \_updateLength();

                        },

                      ),

                      if (\_index == 1)

                        TextField(

                          keyboardType: TextInputType.number,

                          decoration: InputDecoration(

                            labelText: "Inches",

                          ),

                          onChanged: (String? value) {

                            \_enteredInches = double.tryParse(value ?? "0") ?? 0;

                            \_updateLength();

                          },

                        ),

                    ],

                  ),

                ),

                DropdownButton(

                  underline: Container(),

                  value: \_index,

                  items: [

                    DropdownMenuItem(

                      value: 0,

                      child: Text("Meter"),

                    ),

                    DropdownMenuItem(

                      value: 1,

                      child: Text("Feet, Inches"),

                    ),

                  ],

                  onChanged: (index) {

                    \_index = index ?? 0;

                    \_updateLength();

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.end,

              TextSpan(

                text: "Converted Length is \n",

                children: [

                  TextSpan(

                    text:

                        "${\_lengthInMeter.toStringAsFixed(2)}m\n${((\_lengthInInches / 12).toInt()).toStringAsFixed(0)}'${(\_lengthInInches % 12).toStringAsFixed(0)}\"",

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class AreaPage extends StatefulWidget {

  const AreaPage({super.key});

  @override

  State<AreaPage> createState() => \_AreaPageState();

}

class \_AreaPageState extends State<AreaPage> {

  double \_areaInMeter2 = 0;

  double \_areaInFeet2 = 0;

  int \_index = 0;

  double \_enteredValue = 0;

  void \_updateArea() {

    setState(() {

      switch (\_index) {

        case 0:

          \_areaInMeter2 = \_enteredValue;

          \_areaInFeet2 = \_enteredValue \* 10.764;

          break;

        case 1:

          \_areaInFeet2 = \_enteredValue;

          \_areaInMeter2 = \_enteredValue / 10.764;

          break;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Area Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Row(

              crossAxisAlignment: CrossAxisAlignment.start,

              mainAxisSize: MainAxisSize.min,

              children: [

                Expanded(

                  child: TextField(

                    keyboardType: TextInputType.number,

                    decoration: InputDecoration(

                      labelText: "Enter area in ${[

                        "Meters",

                        "Feet",

                      ][\_index]}²",

                    ),

                    onChanged: (String? value) {

                      \_enteredValue = double.tryParse(value ?? "0") ?? 0;

                      \_updateArea();

                    },

                  ),

                ),

                DropdownButton(

                  underline: Container(),

                  value: \_index,

                  items: [

                    DropdownMenuItem(

                      value: 0,

                      child: Text("Meter²"),

                    ),

                    DropdownMenuItem(

                      value: 1,

                      child: Text("Feet²"),

                    ),

                  ],

                  onChanged: (index) {

                    \_index = index ?? 0;

                    \_updateArea();

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.start,

              TextSpan(

                text: "Converted Area is \n",

                children: [

                  TextSpan(

                    text:

                        "${\_areaInMeter2.toStringAsFixed(2)} m²\n${\_areaInFeet2.toStringAsFixed(2)} feet²",

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class VolumePage extends StatefulWidget {

  const VolumePage({super.key});

  @override

  State<VolumePage> createState() => \_VolumePageState();

}

class \_VolumePageState extends State<VolumePage> {

  double \_volumeInMeter3 = 0;

  double \_volumeInFeet3 = 0;

  double \_volumeInLiter = 0;

  int \_index = 0;

  double \_enteredValue = 0;

  void \_updateVolume() {

    setState(() {

      switch (\_index) {

        case 0:

          \_volumeInMeter3 = \_enteredValue;

          \_volumeInFeet3 = \_enteredValue \* 35.315;

          \_volumeInLiter = \_enteredValue \* 1000;

          break;

        case 1:

          \_volumeInFeet3 = \_enteredValue;

          \_volumeInMeter3 = \_enteredValue / 35.315;

          \_volumeInLiter = \_enteredValue \* 28.31;

          break;

        case 2:

          \_volumeInLiter = \_enteredValue;

          \_volumeInMeter3 = \_enteredValue / 1000;

          \_volumeInFeet3 = \_enteredValue / 28.31;

          break;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Volume Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Row(

              crossAxisAlignment: CrossAxisAlignment.start,

              mainAxisSize: MainAxisSize.min,

              children: [

                Expanded(

                  child: TextField(

                    keyboardType: TextInputType.number,

                    decoration: InputDecoration(

                      labelText: "Enter volume in ${[

                        "Meters",

                        "Feet",

                        "Liter",

                      ][\_index]}${\_index != 2 ? '³' : ''}",

                    ),

                    onChanged: (String? value) {

                      \_enteredValue = double.tryParse(value ?? "0") ?? 0;

                      \_updateVolume();

                    },

                  ),

                ),

                DropdownButton(

                  underline: Container(),

                  value: \_index,

                  items: [

                    DropdownMenuItem(

                      value: 0,

                      child: Text("Meter³"),

                    ),

                    DropdownMenuItem(

                      value: 1,

                      child: Text("Feet³"),

                    ),

                    DropdownMenuItem(

                      value: 2,

                      child: Text("Liter"),

                    ),

                  ],

                  onChanged: (index) {

                    \_index = index ?? 0;

                    \_updateVolume();

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.start,

              TextSpan(

                text: "Converted Volume is \n",

                children: [

                  TextSpan(

                    text:

                        "${\_volumeInMeter3.toStringAsFixed(2)} m³\n${\_volumeInFeet3.toStringAsFixed(2)} feet³ \n${\_volumeInLiter.toStringAsFixed(2)} liter",

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class WeightPage extends StatefulWidget {

  const WeightPage({super.key});

  @override

  State<WeightPage> createState() => \_WeightPageState();

}

class \_WeightPageState extends State<WeightPage> {

  double \_massInKg = 0;

  double \_massInLbs = 0;

  int \_index = 0;

  double \_enteredValue = 0;

  void \_updateMass() {

    setState(() {

      switch (\_index) {

        case 0:

          \_massInKg = \_enteredValue;

          \_massInLbs = \_enteredValue \* 2.205;

          break;

        case 1:

          \_massInLbs = \_enteredValue;

          \_massInKg = \_enteredValue / 2.205;

          break;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Mass Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Row(

              crossAxisAlignment: CrossAxisAlignment.start,

              mainAxisSize: MainAxisSize.min,

              children: [

                Expanded(

                  child: TextField(

                    keyboardType: TextInputType.number,

                    decoration: InputDecoration(

                      labelText: "Enter mass in ${[

                        "kg",

                        "lbs",

                      ][\_index]}",

                    ),

                    onChanged: (String? value) {

                      \_enteredValue = double.tryParse(value ?? "0") ?? 0;

                      \_updateMass();

                    },

                  ),

                ),

                DropdownButton(

                  underline: Container(),

                  value: \_index,

                  items: [

                    DropdownMenuItem(

                      value: 0,

                      child: Text("kg"),

                    ),

                    DropdownMenuItem(

                      value: 1,

                      child: Text("lbs"),

                    ),

                  ],

                  onChanged: (index) {

                    \_index = index ?? 0;

                    \_updateMass();

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.start,

              TextSpan(

                text: "Converted Mass is \n",

                children: [

                  TextSpan(

                    text:

                        "${\_massInKg.toStringAsFixed(2)} kg\n${\_massInLbs.toStringAsFixed(2)} lbs",

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class TimeOfDayPage extends StatefulWidget {

  const TimeOfDayPage({super.key});

  @override

  State<TimeOfDayPage> createState() => \_TimeOfDayPageState();

}

class \_TimeOfDayPageState extends State<TimeOfDayPage> {

  TimeOfDay \_time = TimeOfDay.now();

  TimeOfDay \_convertedTime = TimeOfDay.now();

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Belgian Time to PKT",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Column(

              children: [

                Text(

                  \_time.format(context),

                  style: TextStyle(

                    fontWeight: FontWeight.bold,

                    fontSize: 24,

                  ),

                ),

                TextButton(

                  onPressed: () async {

                    \_time = await showTimePicker(

                            context: context, initialTime: \_time) ??

                        TimeOfDay.now();

                    setState(() {

                      \_convertedTime = \_convertedTime.replacing(

                        hour: (\_time.hour + 4) % 24,

                        minute: \_time.minute,

                      );

                    });

                  },

                  child: Text("Pick Belgian Time"),

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.center,

              TextSpan(

                text: "Time in PKT is \n",

                children: [

                  TextSpan(

                    text: \_convertedTime.format(context),

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

extension on DateTime {

  String formatDate() {

    const List<String> months = [

      "January",

      "February",

      "March",

      "April",

      "May",

      "June",

      "July",

      "August",

      "September",

      "October",

      "November",

      "December"

    ];

    String month = months[this.month - 1];

    return "$month $day, $year";

  }

}

extension on Duration {

  String formatAge() {

    int totalDays = inDays;

    int years = totalDays ~/ 365;

    int remainingDaysAfterYears = totalDays % 365;

    int months = remainingDaysAfterYears ~/ 30;

    int days = remainingDaysAfterYears % 30;

    List<String> parts = [];

    if (years > 0) parts.add('$years year${years > 1 ? 's' : ''}');

    if (months > 0) parts.add('$months month${months > 1 ? 's' : ''}');

    if (days > 0) parts.add('$days day${days > 1 ? 's' : ''}');

    return parts.isNotEmpty ? parts.join(', ') : '0 days';

  }

}

class AgePage extends StatefulWidget {

  const AgePage({super.key});

  @override

  State<AgePage> createState() => \_AgePageState();

}

class \_AgePageState extends State<AgePage> {

  DateTime \_date = DateTime.now();

  Duration \_convertedDate = Duration();

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Age Calculator",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Column(

              children: [

                Text(

                  \_date.formatDate(),

                  style: TextStyle(

                    fontWeight: FontWeight.bold,

                    fontSize: 24,

                  ),

                ),

                TextButton(

                  onPressed: () async {

                    \_date = await showDatePicker(

                          context: context,

                          firstDate: DateTime(1900),

                          lastDate: DateTime.now(),

                        ) ??

                        DateTime.now();

                    setState(() {

                      \_convertedDate = DateTime.now().difference(\_date);

                    });

                  },

                  child: Text("Pick Your Birthdate"),

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.center,

              TextSpan(

                text: "Your age is \n",

                children: [

                  TextSpan(

                    text: \_convertedDate.formatAge(),

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}

import 'package:flutter/material.dart';

class TimePage extends StatefulWidget {

  const TimePage({super.key});

  @override

  State<TimePage> createState() => \_TimePageState();

}

class \_TimePageState extends State<TimePage> {

  double \_timeInHr = 0;

  double \_timeInMin = 0;

  double \_timeInSec = 0;

  int \_index = 0;

  double \_enteredValue = 0;

  void \_updateTime() {

    setState(() {

      switch (\_index) {

        case 0:

          \_timeInHr = \_enteredValue;

          \_timeInMin = \_enteredValue \* 60;

          \_timeInSec = \_enteredValue \* 3600;

          break;

        case 1:

          \_timeInMin = \_enteredValue;

          \_timeInHr = \_enteredValue / 60;

          \_timeInSec = \_enteredValue \* 60;

          break;

        case 2:

          \_timeInSec = \_enteredValue;

          \_timeInMin = \_enteredValue / 60;

          \_timeInHr = \_enteredValue / 3600;

          break;

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: Padding(

        padding: const EdgeInsets.all(24),

        child: Column(

          mainAxisSize: MainAxisSize.max,

          mainAxisAlignment: MainAxisAlignment.spaceAround,

          children: [

            Text(

              "Time Converter",

              style: TextStyle(

                fontWeight: FontWeight.bold,

                fontSize: 32,

              ),

            ),

            Row(

              crossAxisAlignment: CrossAxisAlignment.start,

              mainAxisSize: MainAxisSize.min,

              children: [

                Expanded(

                  child: TextField(

                    keyboardType: TextInputType.number,

                    decoration: InputDecoration(

                      labelText: "Enter time in ${[

                        "hour",

                        "minute",

                        "second",

                      ][\_index]}",

                    ),

                    onChanged: (String? value) {

                      \_enteredValue = double.tryParse(value ?? "0") ?? 0;

                      \_updateTime();

                    },

                  ),

                ),

                DropdownButton(

                  underline: Container(),

                  value: \_index,

                  items: [

                    DropdownMenuItem(

                      value: 0,

                      child: Text("Hour"),

                    ),

                    DropdownMenuItem(

                      value: 1,

                      child: Text("Minute"),

                    ),

                    DropdownMenuItem(

                      value: 2,

                      child: Text("Second"),

                    ),

                  ],

                  onChanged: (index) {

                    \_index = index ?? 0;

                    \_updateTime();

                  },

                ),

              ],

            ),

            Text.rich(

              textAlign: TextAlign.start,

              TextSpan(

                text: "Converted Time is \n",

                children: [

                  TextSpan(

                    text:

                        "${\_timeInHr.toStringAsFixed(2)} hrs\n${\_timeInMin.toStringAsFixed(2)} mins\n${\_timeInSec.toStringAsFixed(2)} secs",

                    style: TextStyle(

                      fontWeight: FontWeight.bold,

                      fontSize: 24,

                    ),

                  ),

                ],

              ),

            ),

            Container(),

          ],

        ),

      ),

    );

  }

}