import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Time Picker Themed Demo',

debugShowCheckedModeBanner: false,

theme: \_buildShrineTheme(),

home: MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

TimeOfDay \_time = TimeOfDay(hour: 7, minute: 15);

void \_selectTime() async {

final TimeOfDay newTime = await showTimePicker(

context: context,

initialTime: \_time,

);

if (newTime != null) {

setState(() {

\_time = newTime;

});

}

}

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: \_selectTime,

child: Text('SELECT TIME'),

),

SizedBox(height: 8),

Text(

'Selected time: ${\_time.format(context)}',

),

],

),

),

);

}

}

ThemeData \_buildShrineTheme() {

final ThemeData base = ThemeData.light();

return base.copyWith(

colorScheme: \_shrineColorScheme,

toggleableActiveColor: shrinePink400,

accentColor: shrineBrown900,

primaryColor: shrinePink100,

buttonColor: shrinePink100,

scaffoldBackgroundColor: shrineBackgroundWhite,

cardColor: shrineBackgroundWhite,

textSelectionTheme: TextSelectionThemeData(selectionColor: shrinePink100),

errorColor: shrineErrorRed,

buttonTheme: const ButtonThemeData(

colorScheme: \_shrineColorScheme,

textTheme: ButtonTextTheme.normal,

),

primaryIconTheme: \_customIconTheme(base.iconTheme),

textTheme: \_buildShrineTextTheme(base.textTheme),

primaryTextTheme: \_buildShrineTextTheme(base.primaryTextTheme),

accentTextTheme: \_buildShrineTextTheme(base.accentTextTheme),

iconTheme: \_customIconTheme(base.iconTheme),

);

}

IconThemeData \_customIconTheme(IconThemeData original) {

return original.copyWith(color: shrineBrown900);

}

TextTheme \_buildShrineTextTheme(TextTheme base) {

return base

.copyWith(

caption: base.caption.copyWith(

fontWeight: FontWeight.w400,

fontSize: 14,

letterSpacing: defaultLetterSpacing,

),

button: base.button.copyWith(

fontWeight: FontWeight.w500,

fontSize: 14,

letterSpacing: defaultLetterSpacing,

),

)

.apply(

fontFamily: 'Rubik',

displayColor: shrineBrown900,

bodyColor: shrineBrown900,

);

}

const ColorScheme \_shrineColorScheme = ColorScheme(

primary: shrinePink400,

primaryVariant: shrineBrown900,

secondary: shrinePink50,

secondaryVariant: shrineBrown900,

surface: shrineSurfaceWhite,

background: shrineBackgroundWhite,

error: shrineErrorRed,

onPrimary: shrineBrown900,

onSecondary: shrineBrown900,

onSurface: shrineBrown900,

onBackground: shrineBrown900,

onError: shrineSurfaceWhite,

brightness: Brightness.light,

);

const Color shrinePink50 = Color(0xFFFEEAE6);

const Color shrinePink100 = Color(0xFFFEDBD0);

const Color shrinePink300 = Color(0xFFFBB8AC);

const Color shrinePink400 = Color(0xFFEAA4A4);

const Color shrineBrown900 = Color(0xFF442B2D);

const Color shrineBrown600 = Color(0xFF7D4F52);

const Color shrineErrorRed = Color(0xFFC5032B);

const Color shrineSurfaceWhite = Color(0xFFFFFBFA);

const Color shrineBackgroundWhite = Colors.white;

const defaultLetterSpacing = 0.03;