**Week-6:**

**6a)Create custom widgets for specific UI elements.**

import 'package:flutter/material.dart';

class CustomButton extends StatelessWidget {

final String label;

final VoidCallback onPressed;

final Color color;

CustomButton({

required this.label,

required this.onPressed,

this.color = Colors.blue,

});

@override

Widget build(BuildContext context) {

return ElevatedButton(

style: ElevatedButton.styleFrom(primary: color),

onPressed: onPressed,

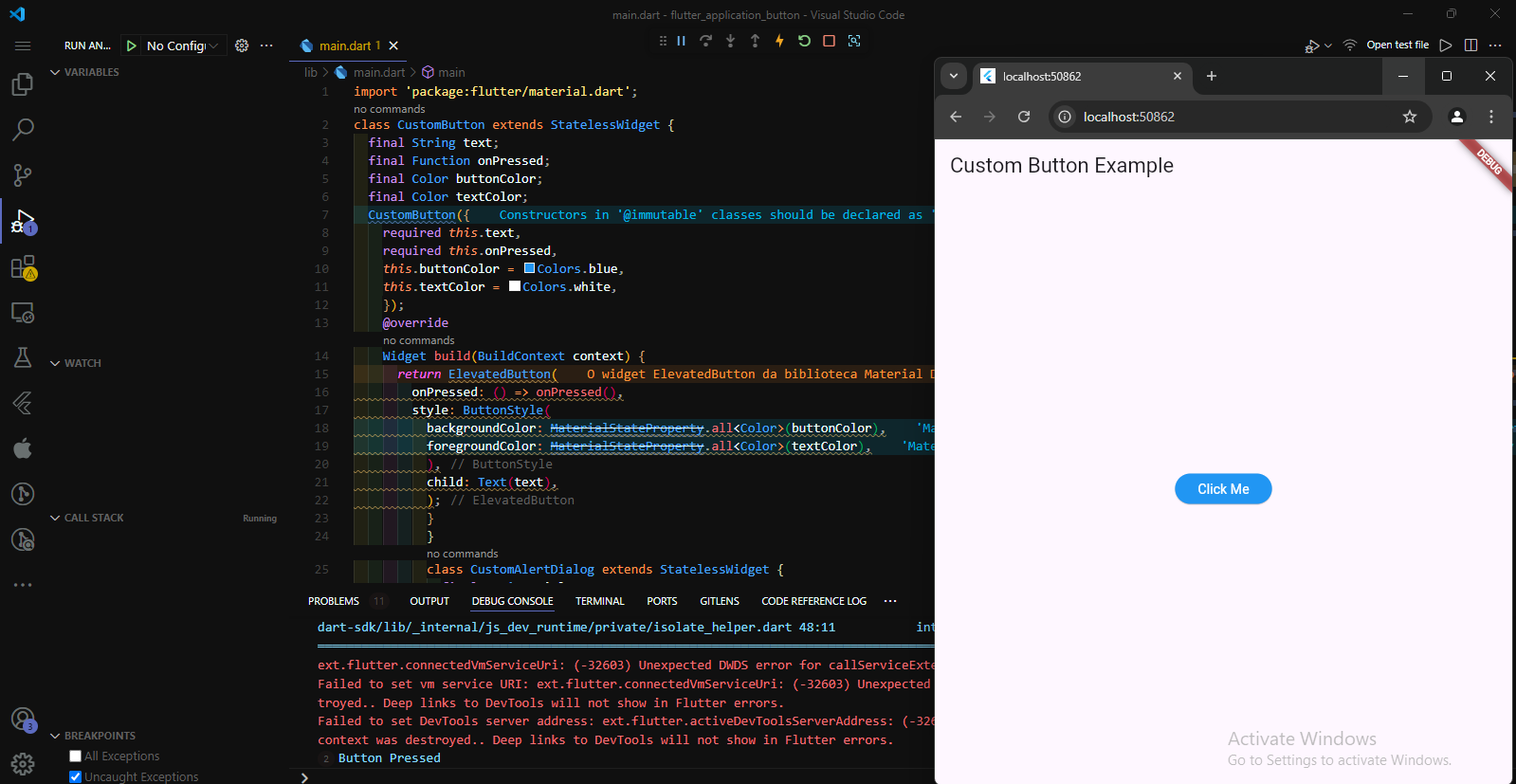
child: Text(label),

);

}

}

**Output**:



**6b) Apply styling using themes and custom styles.**

import 'package:flutter/material.dart';

// Include the Google Fonts package to provide more text format options

// https://pub.dev/packages/google\_fonts

import 'package:google\_fonts/google\_fonts.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

const appName = 'Custom Themes';

return MaterialApp(

title: appName,

theme: ThemeData(

useMaterial3: true,

// Define the default brightness and colors.

colorScheme: ColorScheme.fromSeed(

seedColor: Colors.purple,

// TRY THIS: Change to "Brightness.light"

// and see that all colors change

// to better contrast a light background.

brightness: Brightness.dark,

),

// Define the default `TextTheme`. Use this to specify the default

// text styling for headlines, titles, bodies of text, and more.

textTheme: TextTheme(

displayLarge: const TextStyle(

fontSize: 72,

fontWeight: FontWeight.bold,

),

// TRY THIS: Change one of the GoogleFonts

// to "lato", "poppins", or "lora".

// The title uses "titleLarge"

// and the middle text uses "bodyMedium".

titleLarge: GoogleFonts.oswald(

fontSize: 30,

fontStyle: FontStyle.italic,

),

bodyMedium: GoogleFonts.merriweather(),

displaySmall: GoogleFonts.pacifico(),

),

),

home: const MyHomePage(

title: appName,

),

);

}

}

class MyHomePage extends StatelessWidget {

final String title;

const MyHomePage({super.key, required this.title});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(title,

style: Theme.of(context).textTheme.titleLarge!.copyWith(

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

)),

backgroundColor: Theme.of(context).colorScheme.secondary,

),

body: Center(

child: Container(

padding: const EdgeInsets.symmetric(

horizontal: 12,

vertical: 12,

),

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

child: Text(

'Text with a background color',

// TRY THIS: Change the Text value

// or change the Theme.of(context).textTheme

// to "displayLarge" or "displaySmall".

style: Theme.of(context).textTheme.bodyMedium!.copyWith(

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

),

),

),

),

floatingActionButton: Theme(

data: Theme.of(context).copyWith(

// TRY THIS: Change the seedColor to "Colors.red" or

// "Colors.blue".

colorScheme: ColorScheme.fromSeed(

seedColor: Colors.pink,

brightness: Brightness.dark,

),

),

child: FloatingActionButton(

onPressed: () {},

child: const Icon(Icons.add),

),

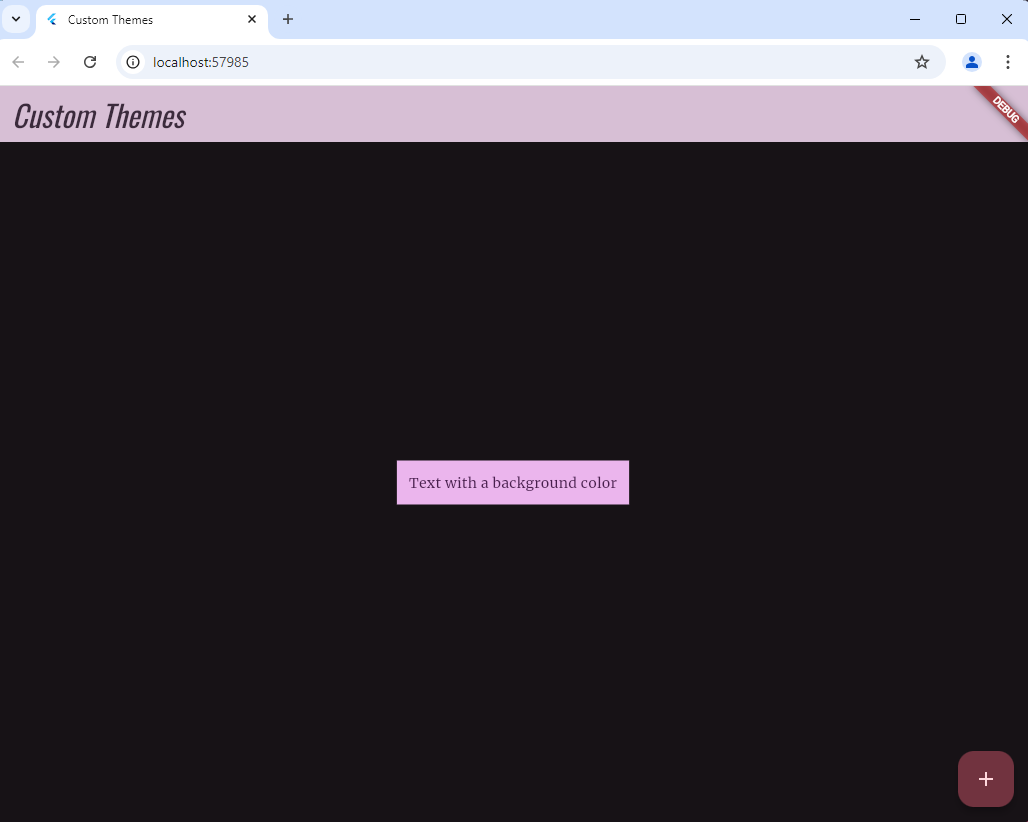
),

);

}

}

**Output:**

****

**Week-7:**

**7. a) Design a form with various input fields.**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Form Example',

theme: ThemeData(primarySwatch: Colors.blue),

home: FormPage(),

);

}

}

class FormPage extends StatefulWidget {

@override

\_FormPageState createState() => \_FormPageState();

}

class \_FormPageState extends State<FormPage> {

final \_formKey = GlobalKey<FormState>();

final TextEditingController \_nameController = TextEditingController();

final TextEditingController \_emailController = TextEditingController();

String? \_selectedGender;

bool \_acceptTerms = false;

List<String> genders = ['Male', 'Female', 'Other'];

void \_submitForm() {

if (\_formKey.currentState!.validate()) {

// Perform submission logic

print('Name: ${\_nameController.text}');

print('Email: ${\_emailController.text}');

print('Gender: $\_selectedGender');

print('Accepted Terms: $\_acceptTerms');

ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('Form submitted!')));

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Flutter Form Example')),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Form(

key: \_formKey,

child: Column(

children: [

TextFormField(

controller: \_nameController,

decoration: InputDecoration(labelText: 'Name'),

validator: (value) {

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

return 'Please enter your name';

}

return null;

},

),

SizedBox(height: 16),

TextFormField(

controller: \_emailController,

decoration: InputDecoration(labelText: 'Email'),

validator: (value) {

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

return 'Please enter your email';

} else if (!RegExp(r'^[^@]+@[^@]+\.[^@]+').hasMatch(value)) {

return 'Please enter a valid email';

}

return null;

},

),

SizedBox(height: 16),

DropdownButtonFormField<String>(

decoration: InputDecoration(labelText: 'Gender'),

value: \_selectedGender,

items: genders.map((String gender) {

return DropdownMenuItem<String>(

value: gender,

child: Text(gender),

);

}).toList(),

onChanged: (String? newValue) {

setState(() {

\_selectedGender = newValue;

});

},

validator: (value) {

if (value == null) {

return 'Please select your gender';

}

return null;

},

),

SizedBox(height: 16),

Row(

children: [

Checkbox(

value: \_acceptTerms,

onChanged: (bool? value) {

setState(() {

\_acceptTerms = value ?? false;

});

},

),

Flexible(

child: Text('I accept the terms and conditions'),

),

],

),

SizedBox(height: 16),

ElevatedButton(

onPressed: \_submitForm,

child: Text('Submit'),

),

],

),

),

),

);

}

}

**Output:**

**7 b) Implement form validation and error handling.**

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

const appTitle = 'Form Validation Demo';

return MaterialApp(

title: appTitle,

home: Scaffold(

appBar: AppBar(

title: const Text(appTitle),

),

body: const MyCustomForm(),

),

);

}

}

// Create a Form widget.

class MyCustomForm extends StatefulWidget {

const MyCustomForm({super.key});

@override

MyCustomFormState createState() {

return MyCustomFormState();

}

}

// Create a corresponding State class.

// This class holds data related to the form.

class MyCustomFormState extends State<MyCustomForm> {

// Create a global key that uniquely identifies the Form widget

// and allows validation of the form.

//

// Note: This is a GlobalKey<FormState>,

// not a GlobalKey<MyCustomFormState>.

final \_formKey = GlobalKey<FormState>();

@override

Widget build(BuildContext context) {

// Build a Form widget using the \_formKey created above.

return Form(

key: \_formKey,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

TextFormField(

// The validator receives the text that the user has entered.

validator: (value) {

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

return 'Please enter some text';

}

return null;

},

),

Padding(

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

child: ElevatedButton(

onPressed: () {

// Validate returns true if the form is valid, or false otherwise.

if (\_formKey.currentState!.validate()) {

// If the form is valid, display a snackbar. In the real world,

// you'd often call a server or save the information in a database.

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text('Processing Data')),

);

}

},

child: const Text('Submit'),

),

),

],

),

);

}

}

**Output:**

**Week-8:**

**8a)** **Add animations to UI elements using Flutter's animation framework.**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Animation Example',

theme: ThemeData(primarySwatch: Colors.blue),

home: AnimationPage(),

);

}

}

class AnimationPage extends StatefulWidget {

@override

\_AnimationPageState createState() => \_AnimationPageState();

}

class \_AnimationPageState extends State<AnimationPage>

with SingleTickerProviderStateMixin {

late AnimationController \_controller;

late Animation<double> \_fadeAnimation;

@override

void initState() {

super.initState();

\_controller = AnimationController(

duration: const Duration(seconds: 2),

vsync: this,

);

\_fadeAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(

CurvedAnimation(

parent: \_controller,

curve: Curves.easeIn,

),

);

\_controller.forward(); // Start the animation

}

@override

void dispose() {

\_controller.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Flutter Animation Example')),

body: Center(

child: FadeTransition(

opacity: \_fadeAnimation,

child: Container(

width: 200,

height: 200,

decoration: BoxDecoration(

color: Colors.blue,

borderRadius: BorderRadius.circular(15),

),

child: Center(

child: Text(

'Fade In!',

style: TextStyle(color: Colors.white, fontSize: 24),

),

),

),

),

),

);

}

}

**Output:**

**8b)** **Experiment with different types of animations (fade, slide, etc.)**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Animation Example',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: AnimationDemo(),

);

}

}

class AnimationDemo extends StatefulWidget {

@override

\_AnimationDemoState createState() => \_AnimationDemoState();

}

class \_AnimationDemoState extends State<AnimationDemo>

with SingleTickerProviderStateMixin {

late AnimationController \_controller;

late Animation<double> \_fadeAnimation;

late Animation<Offset> \_slideAnimation;

bool \_isVisible = false;

@override

void initState() {

super.initState();

\_controller = AnimationController(

duration: const Duration(milliseconds: 500),

vsync: this,

);

\_fadeAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(\_controller);

\_slideAnimation = Tween<Offset>(begin: Offset(0.0, 1.0), end: Offset(0.0, 0.0)).animate(CurvedAnimation(

parent: \_controller,

curve: Curves.easeInOut,

));

}

void \_toggleFade() {

setState(() {

\_isVisible = !\_isVisible;

if (\_isVisible) {

\_controller.forward();

} else {

\_controller.reverse();

}

});

}

void \_toggleSlide() {

setState(() {

\_isVisible = !\_isVisible;

if (\_isVisible) {

\_controller.forward();

} else {

\_controller.reverse();

}

});

}

@override

void dispose() {

\_controller.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Flutter Animation Example'),

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: \_toggleFade,

child: Text('Toggle Fade Animation'),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: \_toggleSlide,

child: Text('Toggle Slide Animation'),

),

SizedBox(height: 50),

// Fade Animation

FadeTransition(

opacity: \_fadeAnimation,

child: Container(

width: 200,

height: 100,

color: Colors.blue,

alignment: Alignment.center,

child: Text(

'Fade Animation',

style: TextStyle(color: Colors.white, fontSize: 20),

),

),

),

// Slide Animation

SlideTransition(

position: \_slideAnimation,

child: Container(

width: 200,

height: 100,

color: Colors.green,

alignment: Alignment.center,

child: Text(

'Slide Animation',

style: TextStyle(color: Colors.white, fontSize: 20),

),

),

),

],

),

),

);

}

}

**Output:**