IT / CS506PC: UI DESIGN-FLUTTER

B.Tech. III Year I Sem. L T P C

0 0 2 1

**Course Objectives:**

* Learns to Implement Flutter Widgets and Layouts
* Understands Responsive UI Design and with Navigation in Flutter
* Knowledge on Widgets and customize widgets for specific UI elements, Themes
* Understand to include animation apart from fetching data

**Course Outcomes:**

* Implements Flutter Widgets and Layouts
* Responsive UI Design and with Navigation in Flutter
* Create custom widgets for specific UI elements and also Apply styling using themes and custom styles.
* Design a form with various input fields, along with validation and error handling
* Fetches data and write code for unit Test for UI components and also animation

List of Experiments:

1. a) Install Flutter and Dart SDK

b) Write a simple Dart program to understand the language basics

1. a) Explore various Flutter widgets (Text, Image, Container, etc.).

b) Implement different layout structures using Row, Column, and Stack widgets.

1. a) Design a responsive UI that adapts to different screen sizes.

b) Implement media queries and breakpoints for responsiveness.

1. a) Set up navigation between different screens using Navigator.

b) Implement navigation with named routes.

1. a) Learn about stateful and stateless widgets.

b) Implement state management using set State and Provider.

1. a) Create custom widgets for specific UI elements.

b) Apply styling using themes and custom styles.

1. a) Design a form with various input fields.

b) Implement form validation and error handling.

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

b) Experiment with different types of animations (fade, slide)

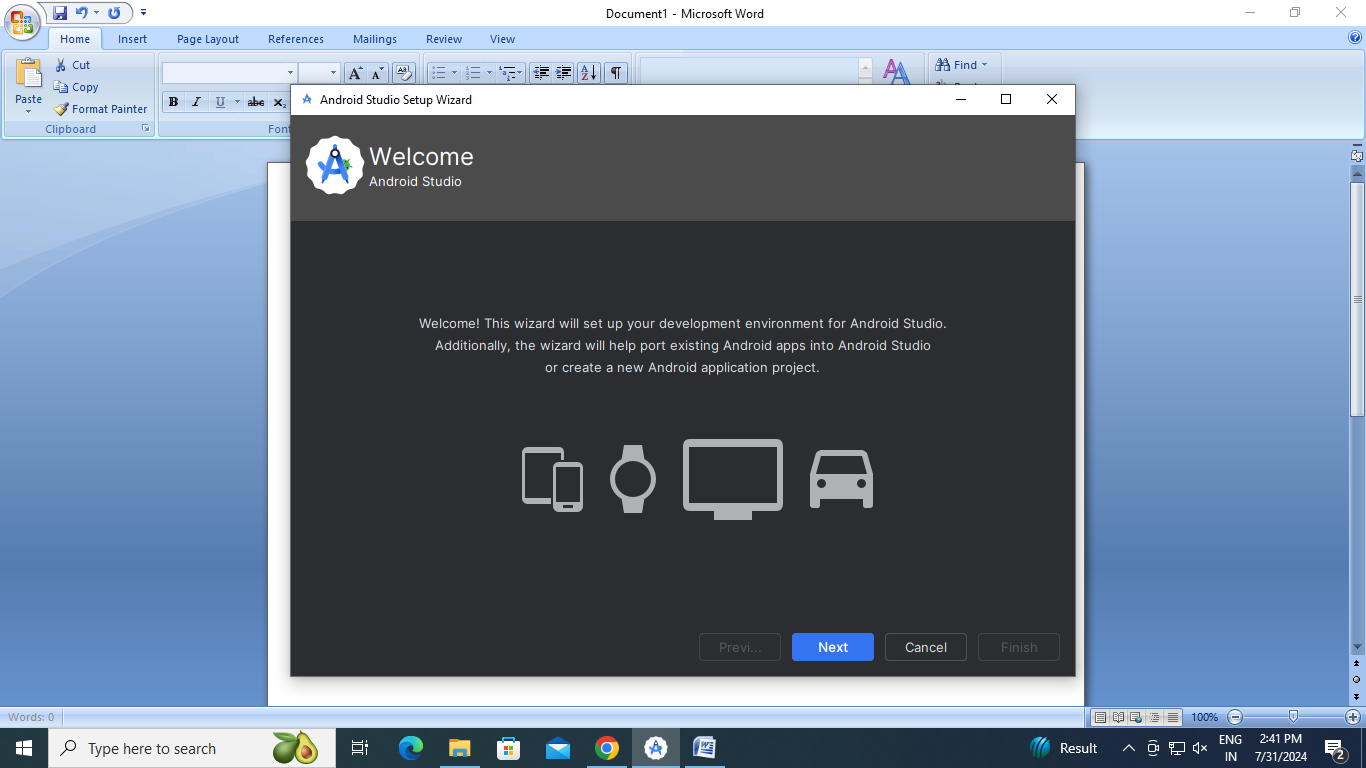
1. a) Fetch data from a REST API.

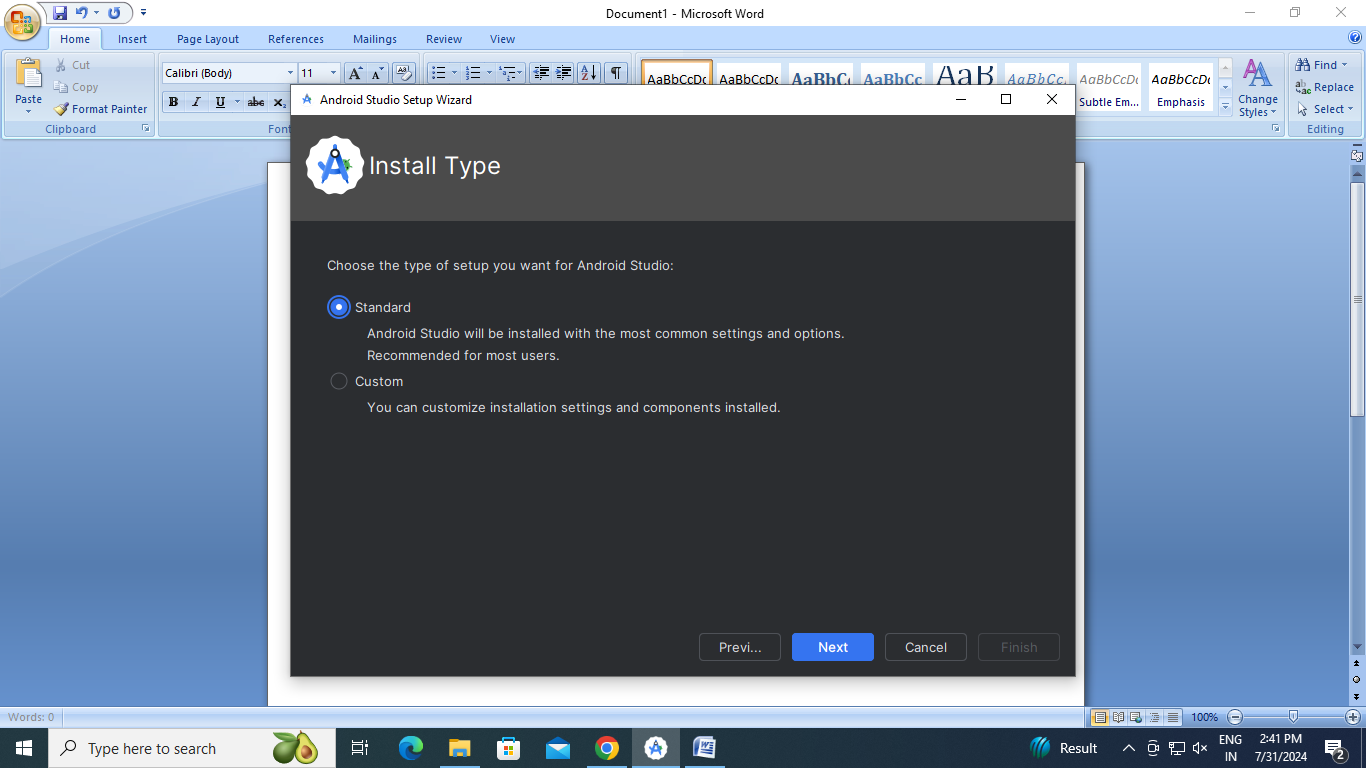
b) Display the fetched data in a meaningful way in the UI.

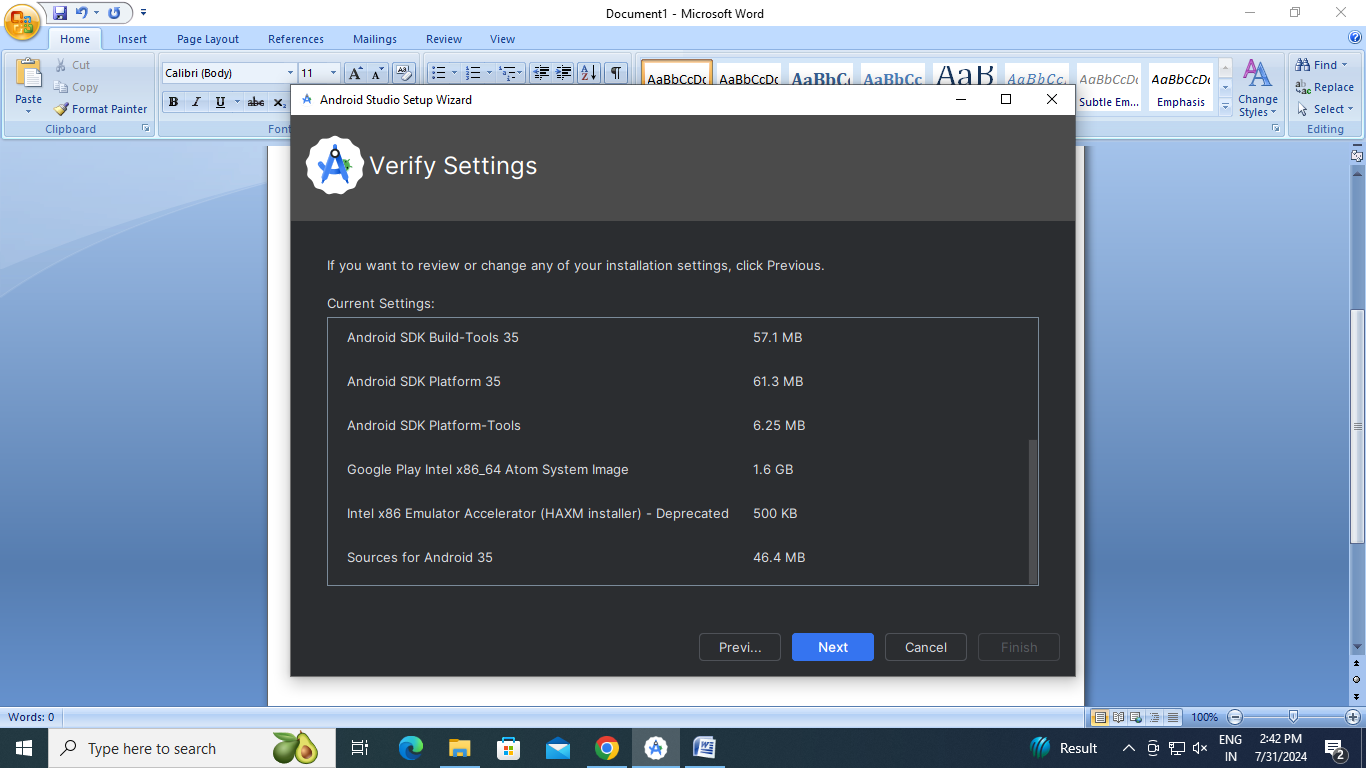
1. a) Write unit tests for UI components.

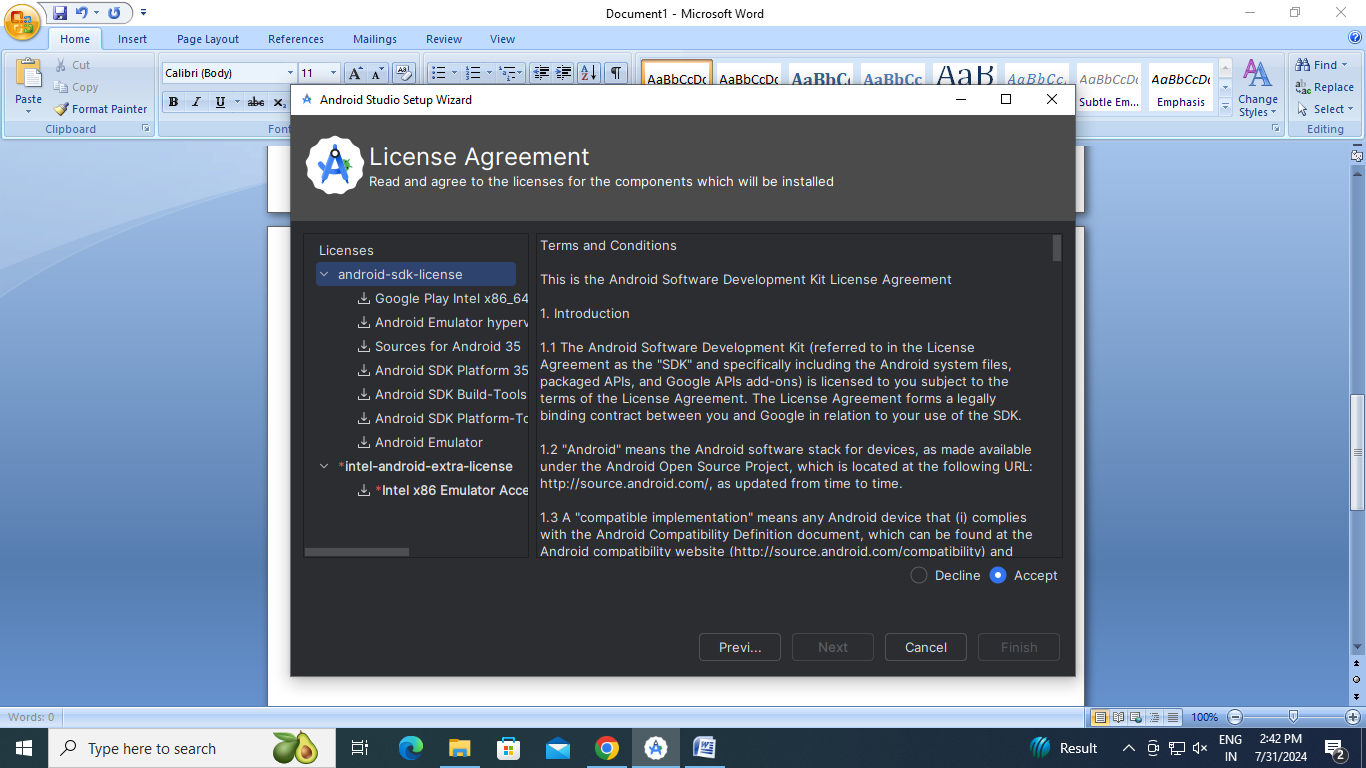
b) Use Flutter's debugging tools to identify and fix issues.

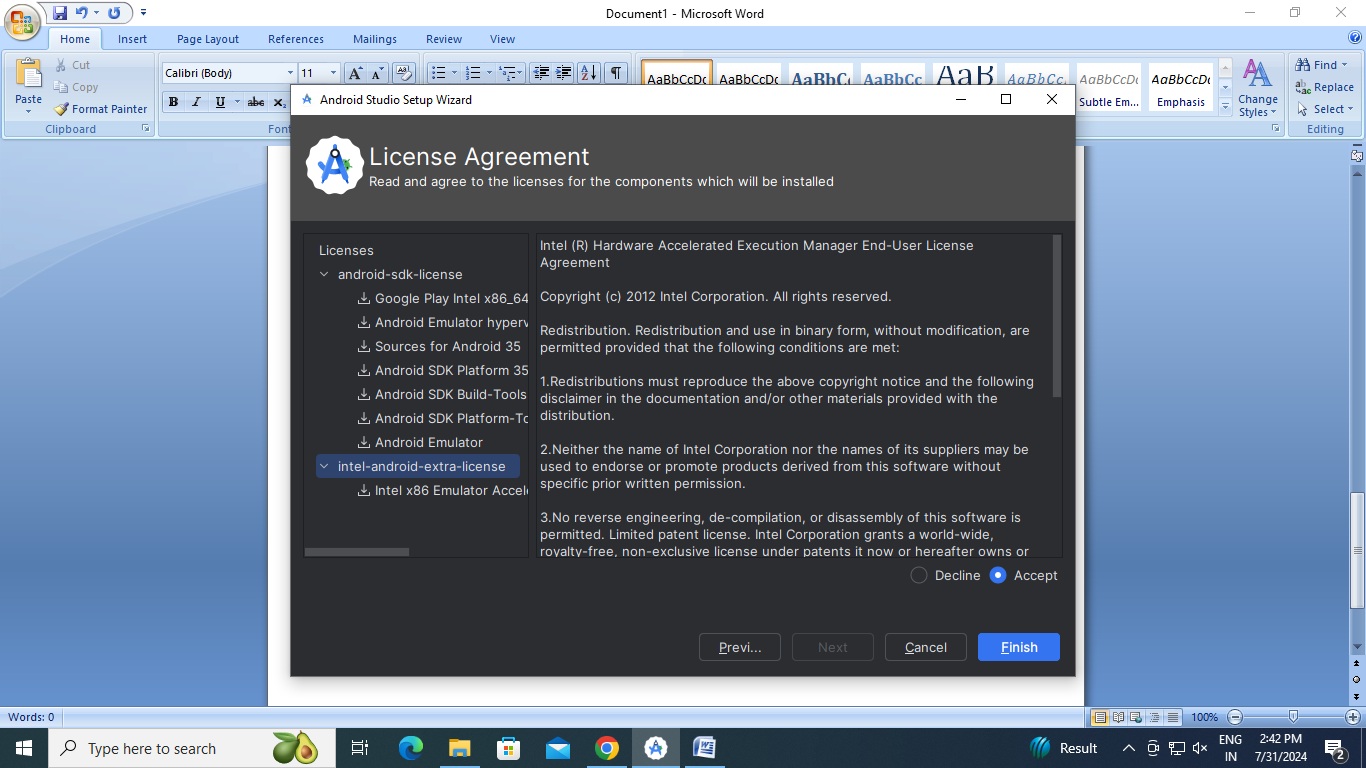
**1 a) Install Flutter and Dart SDK**

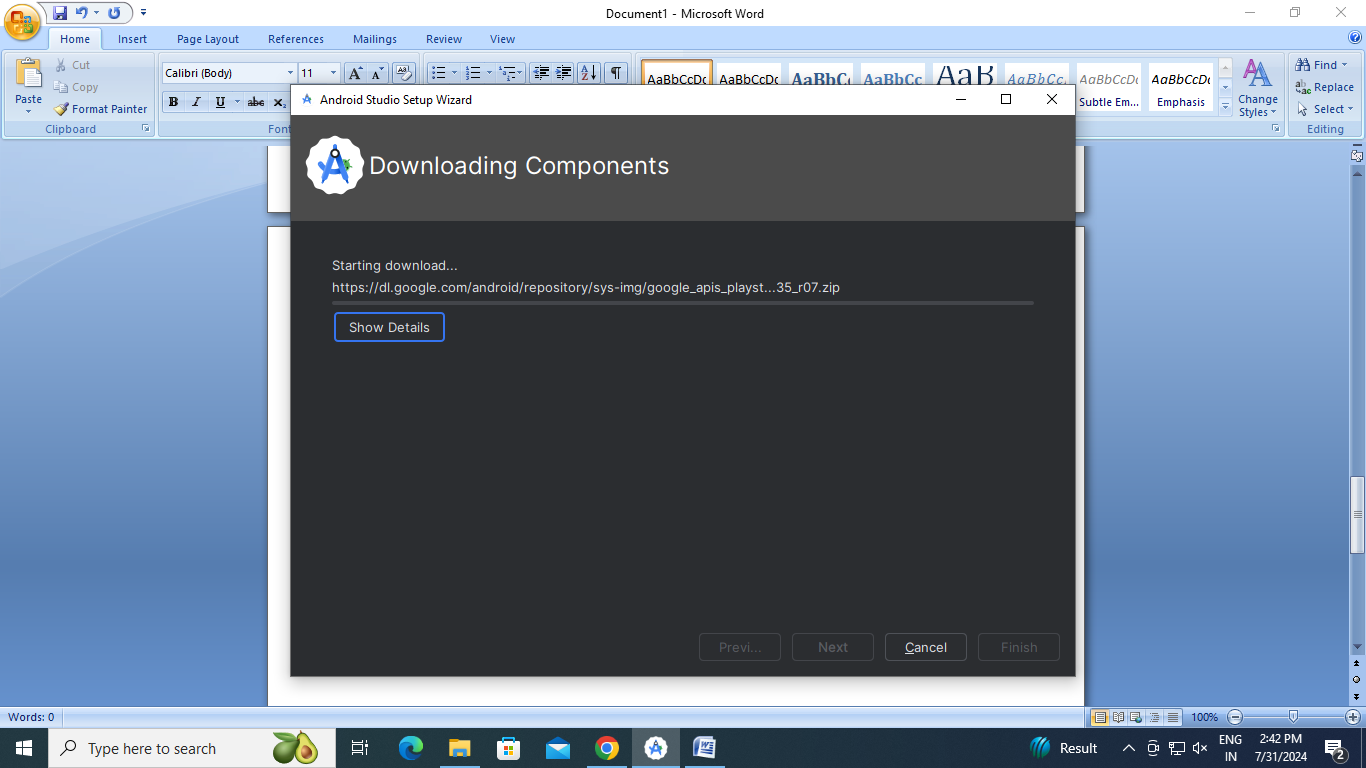


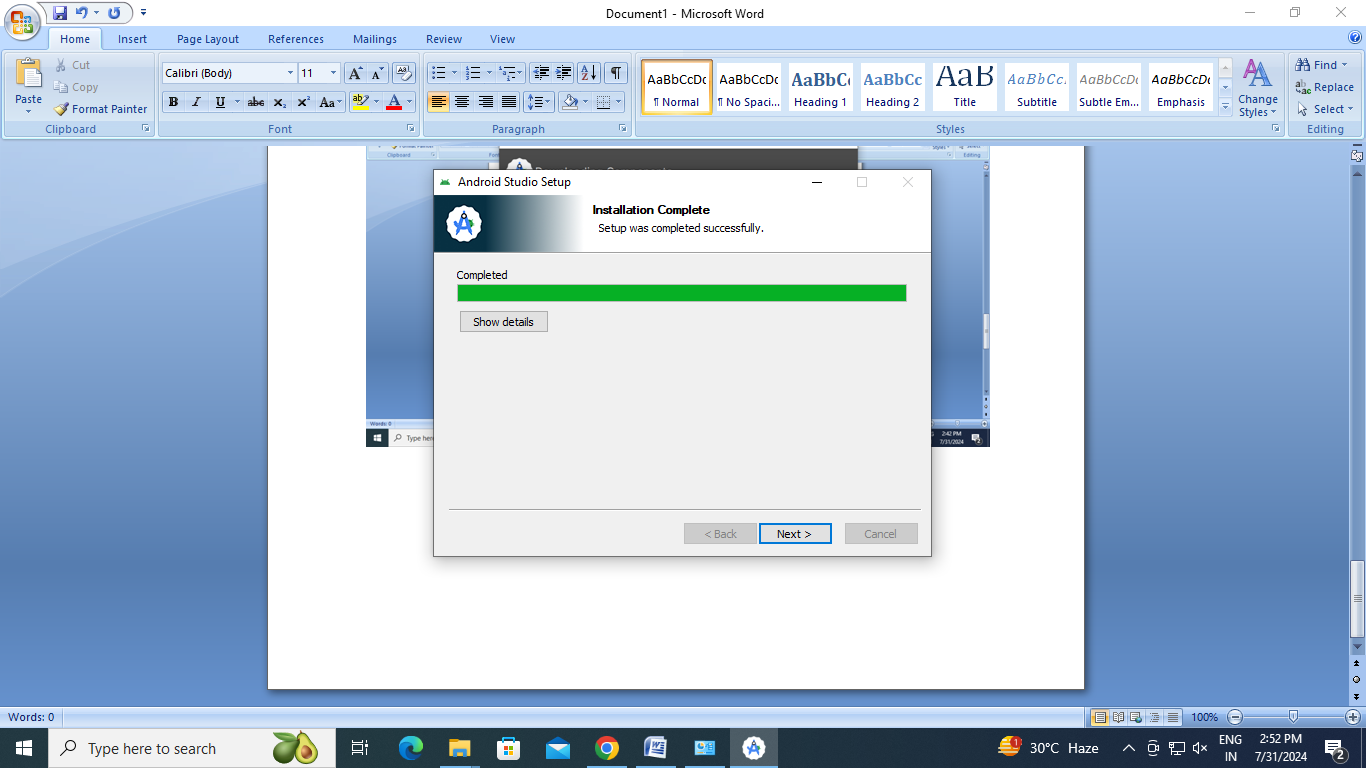


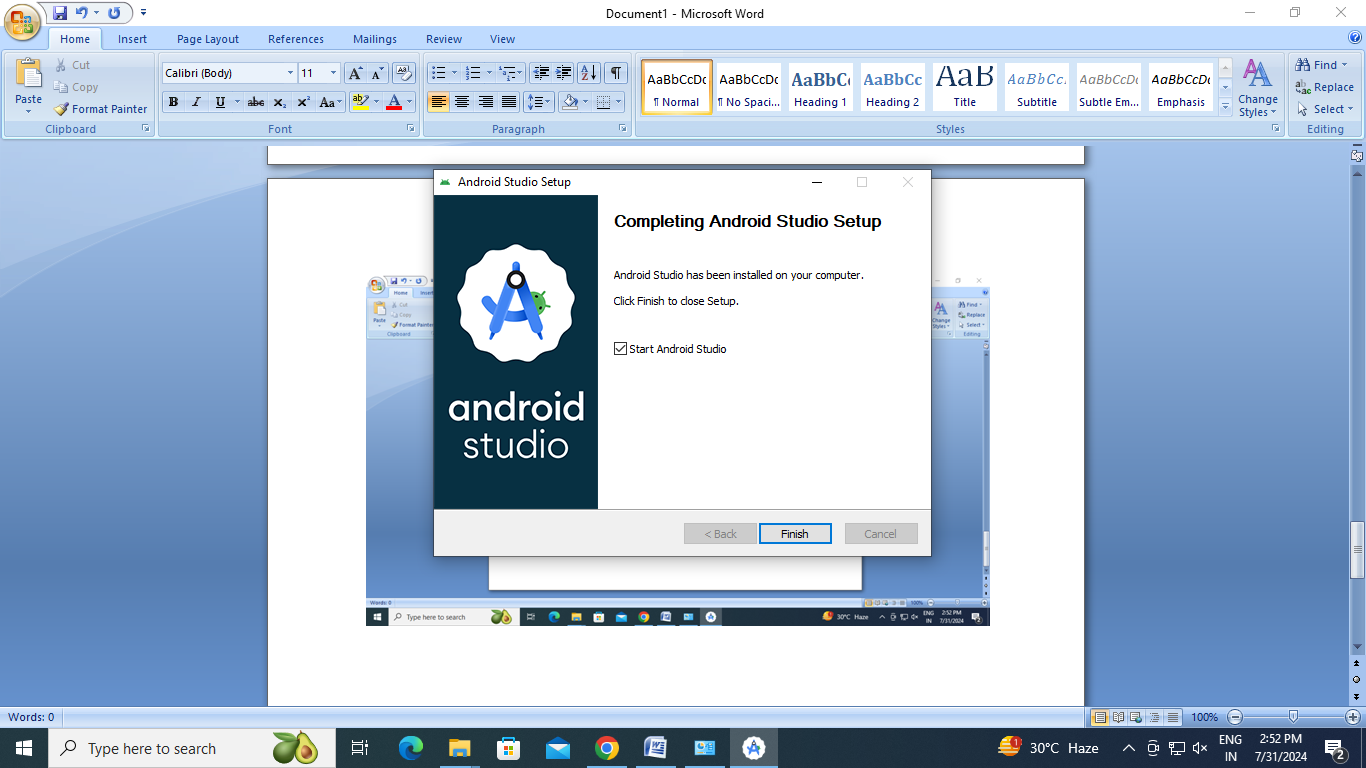


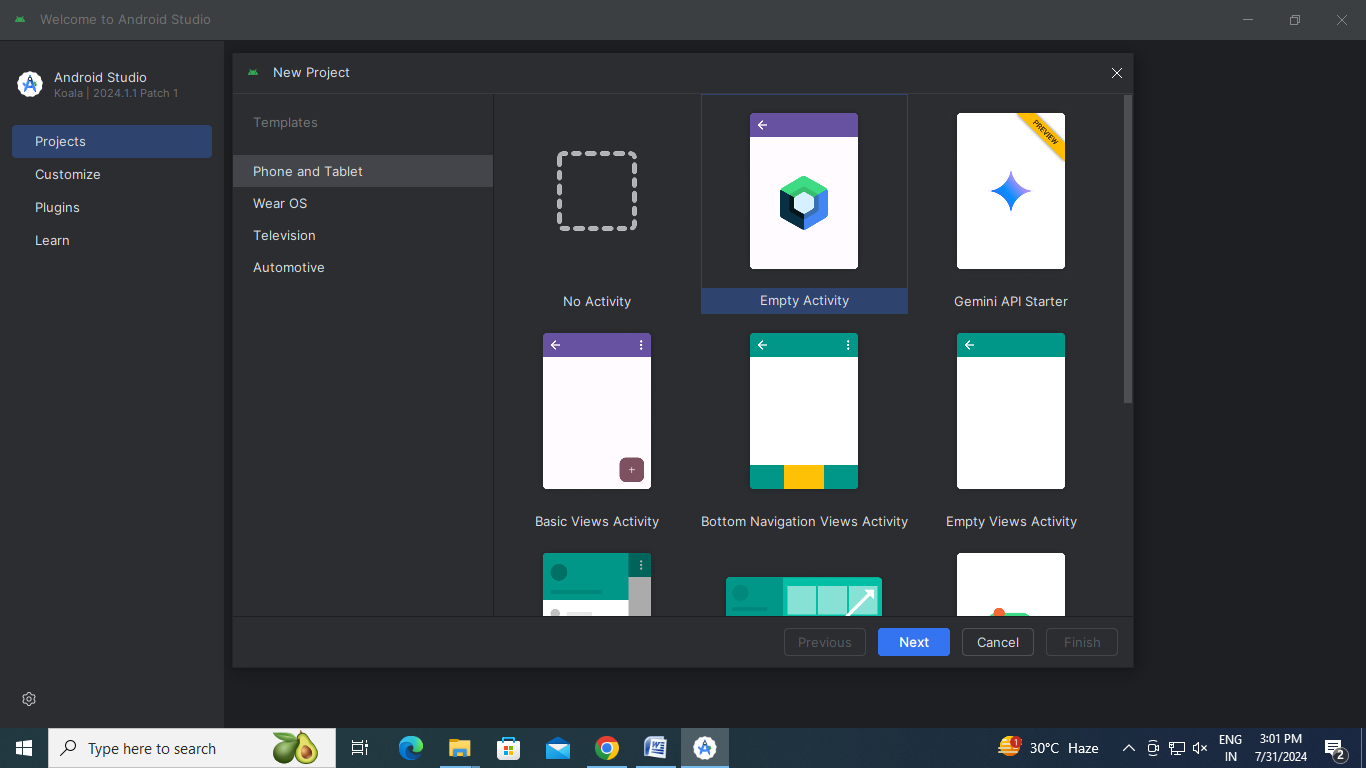


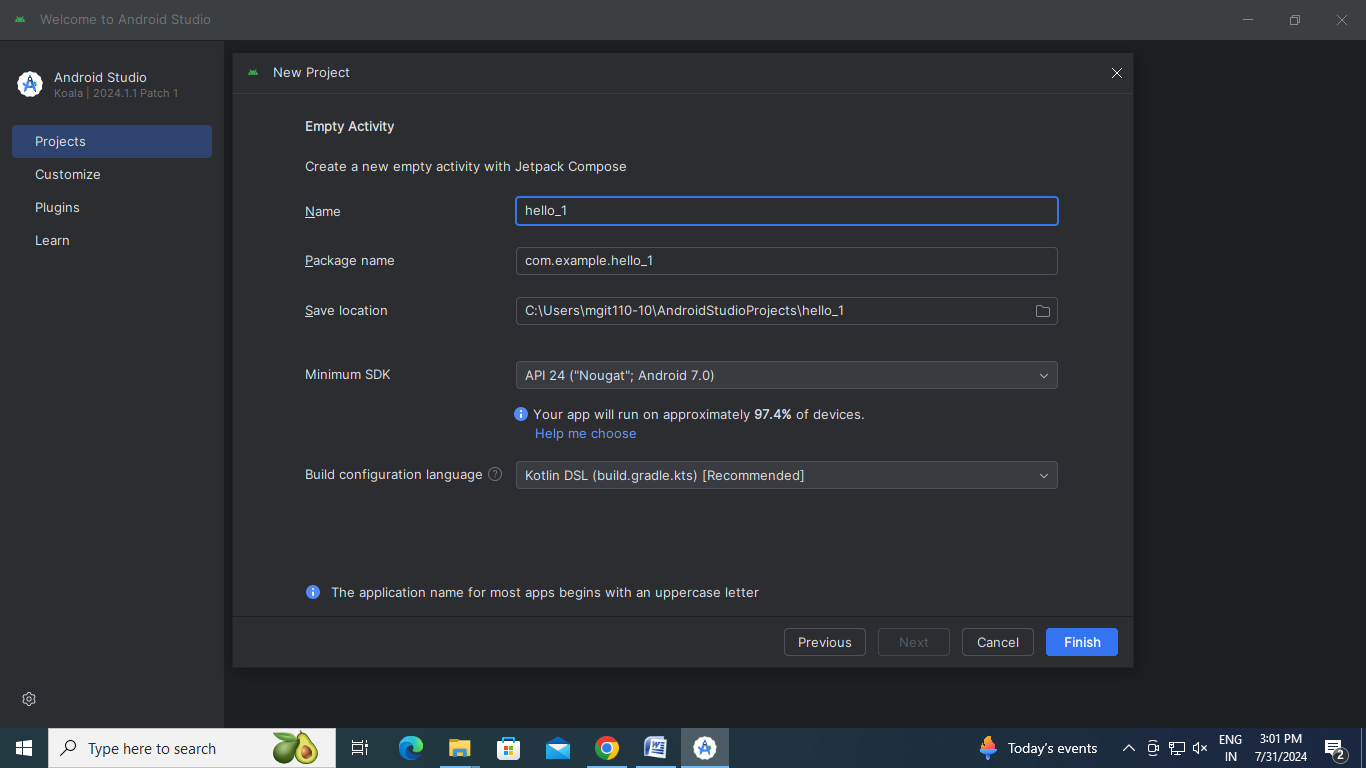


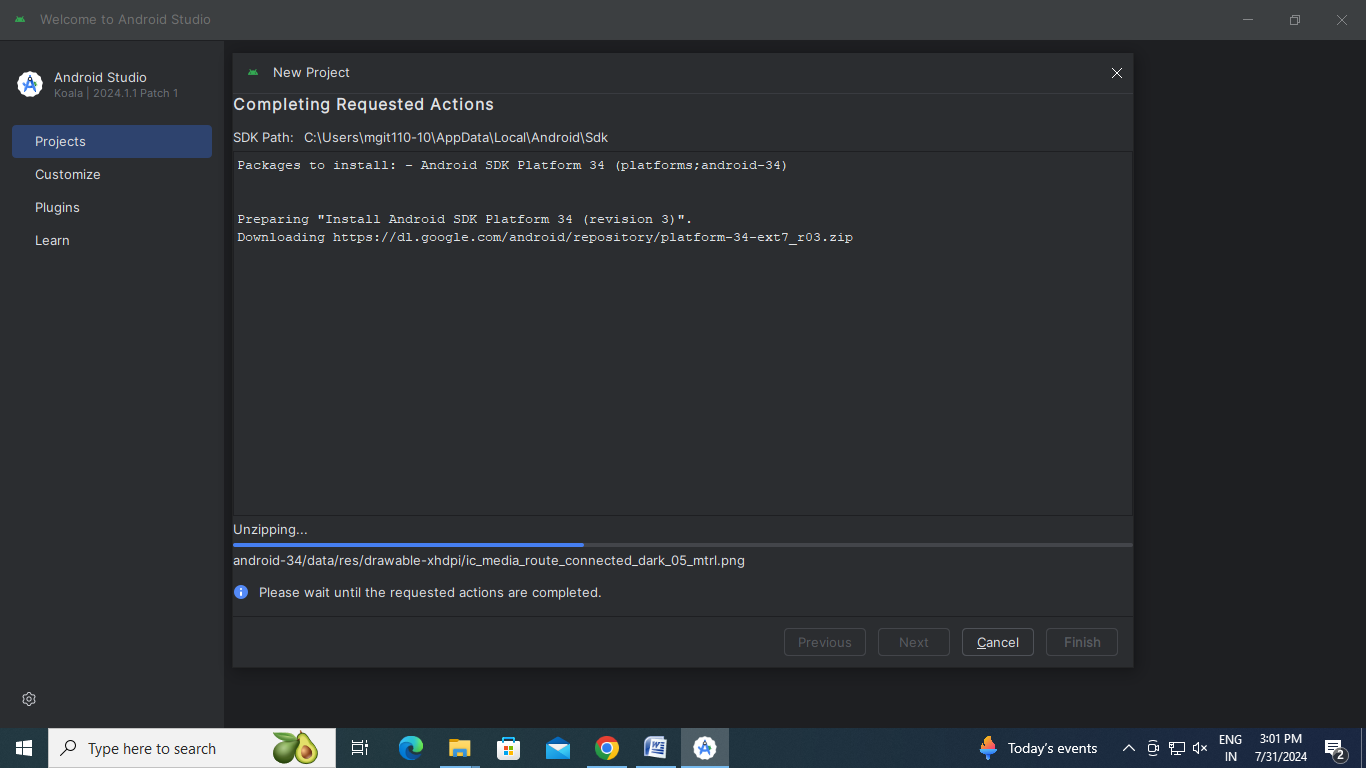


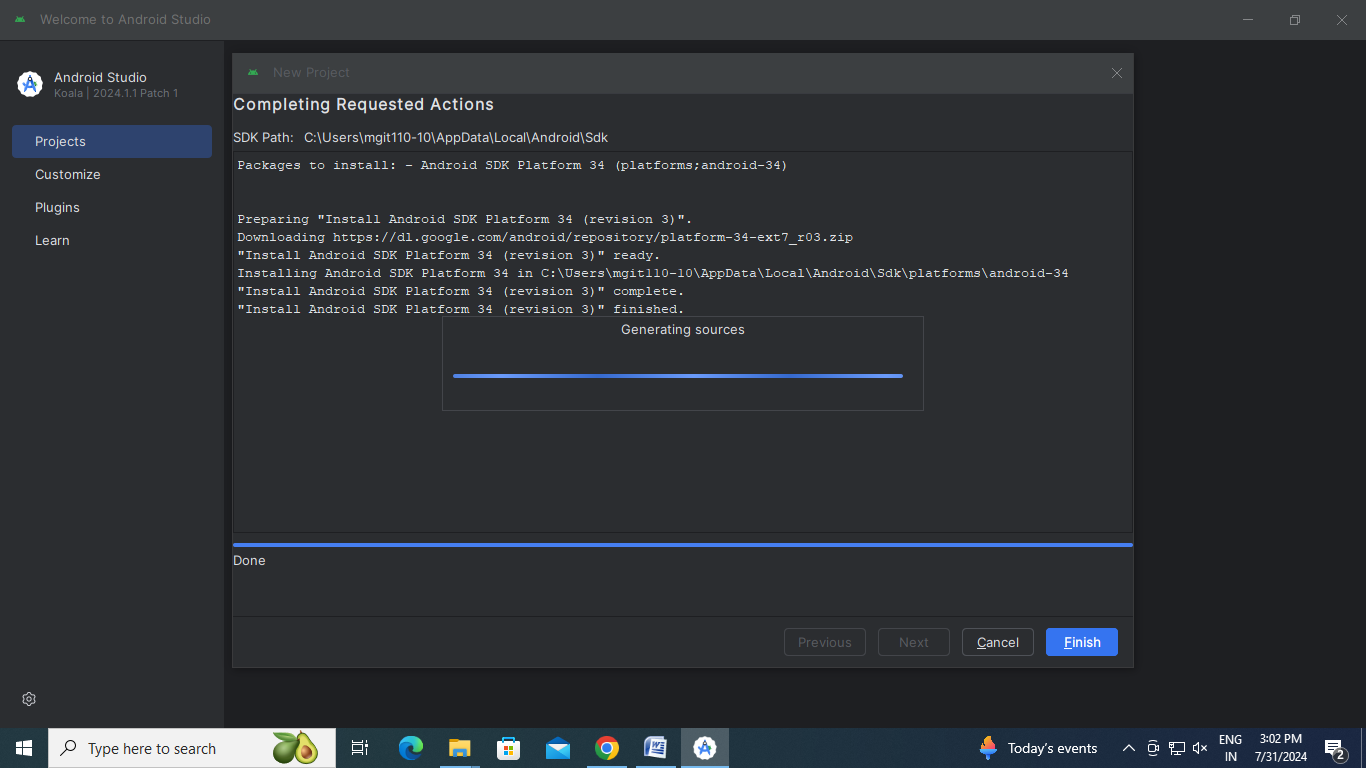


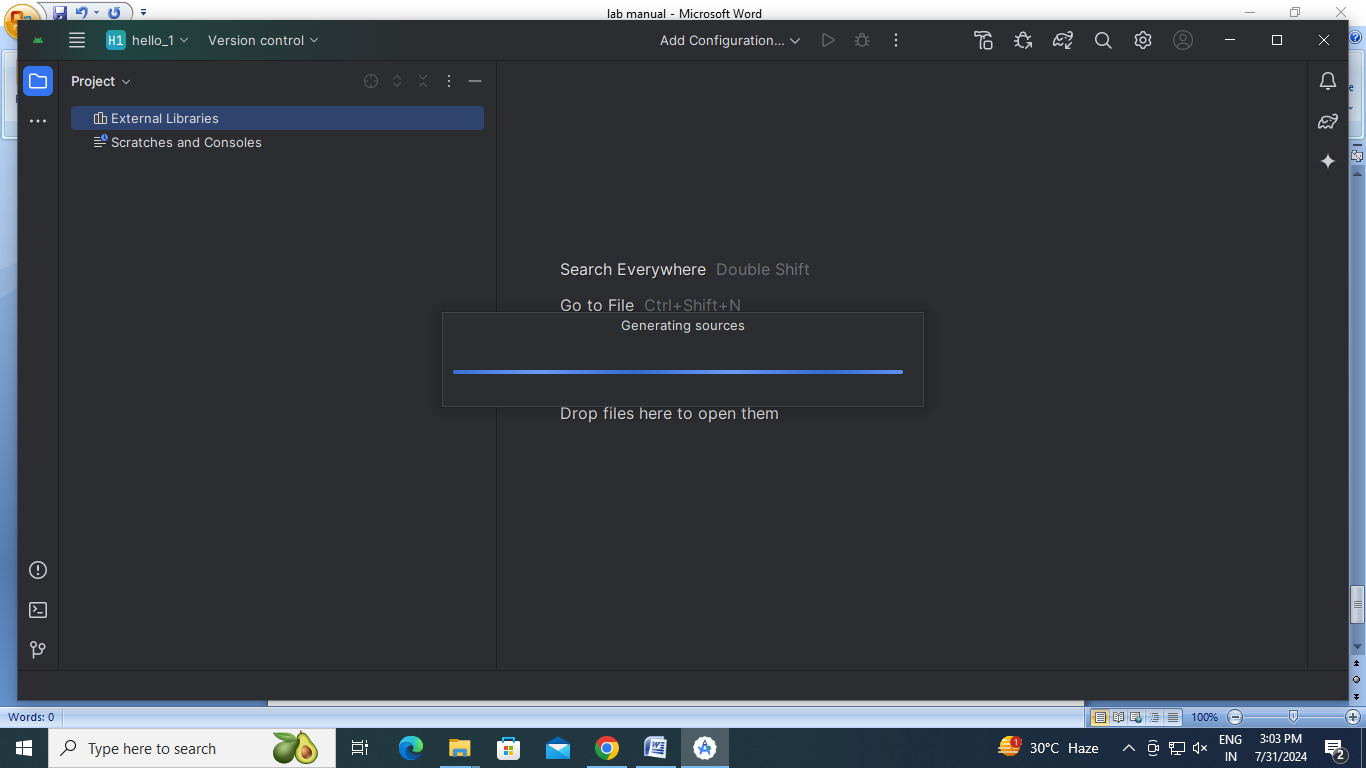


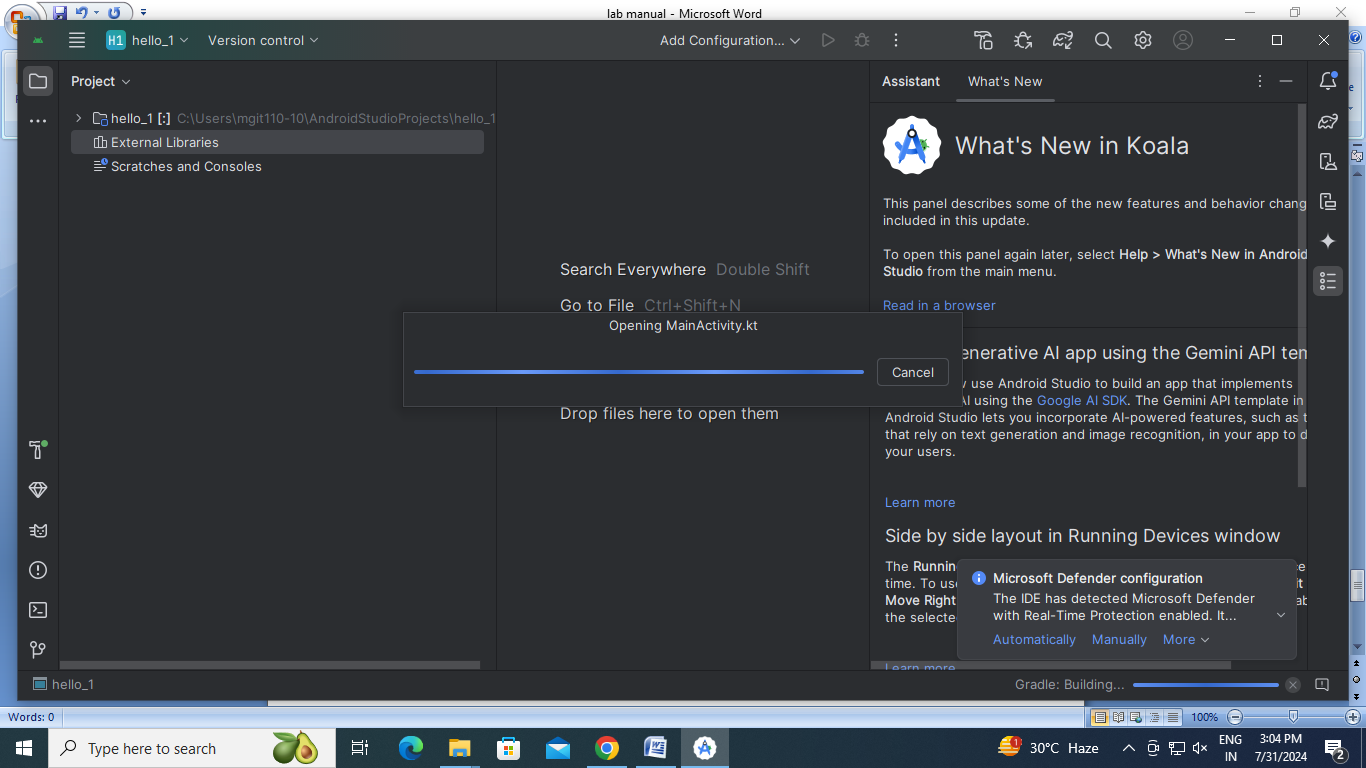


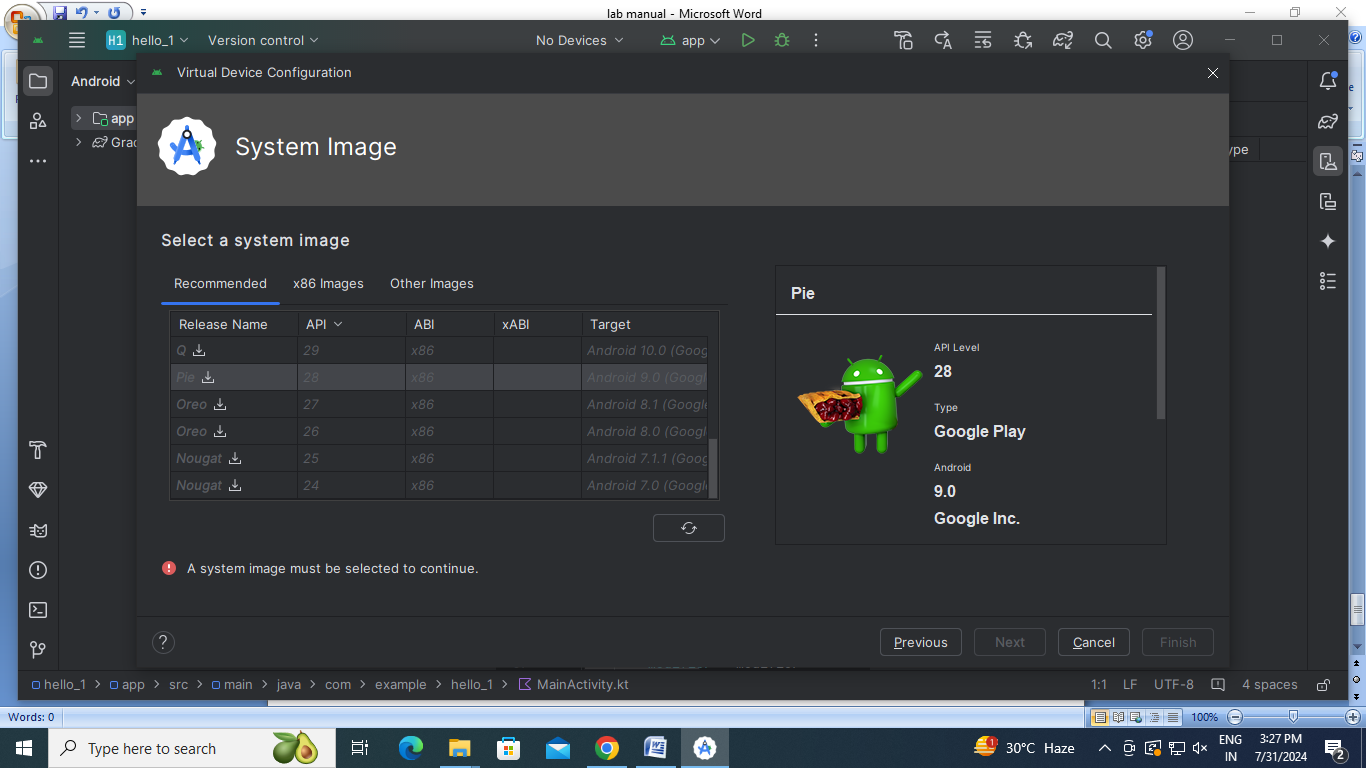


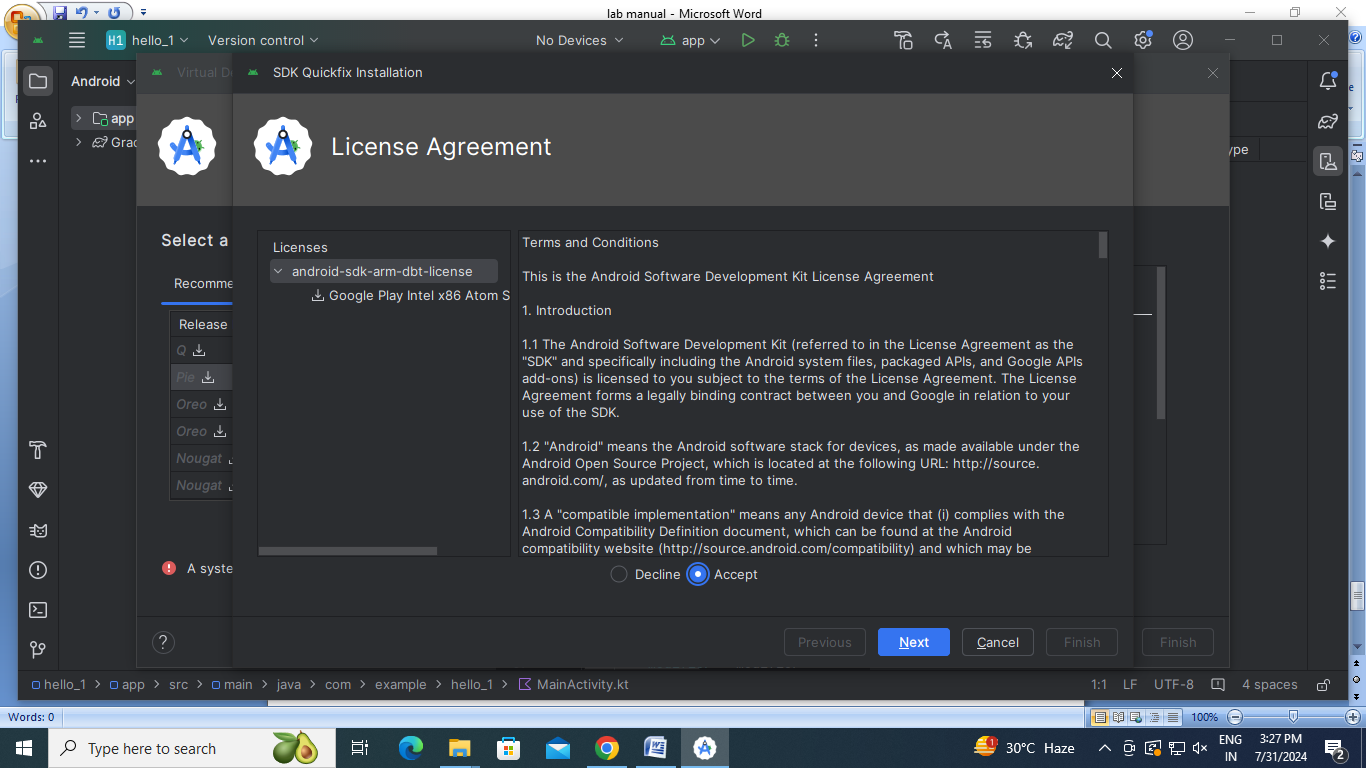


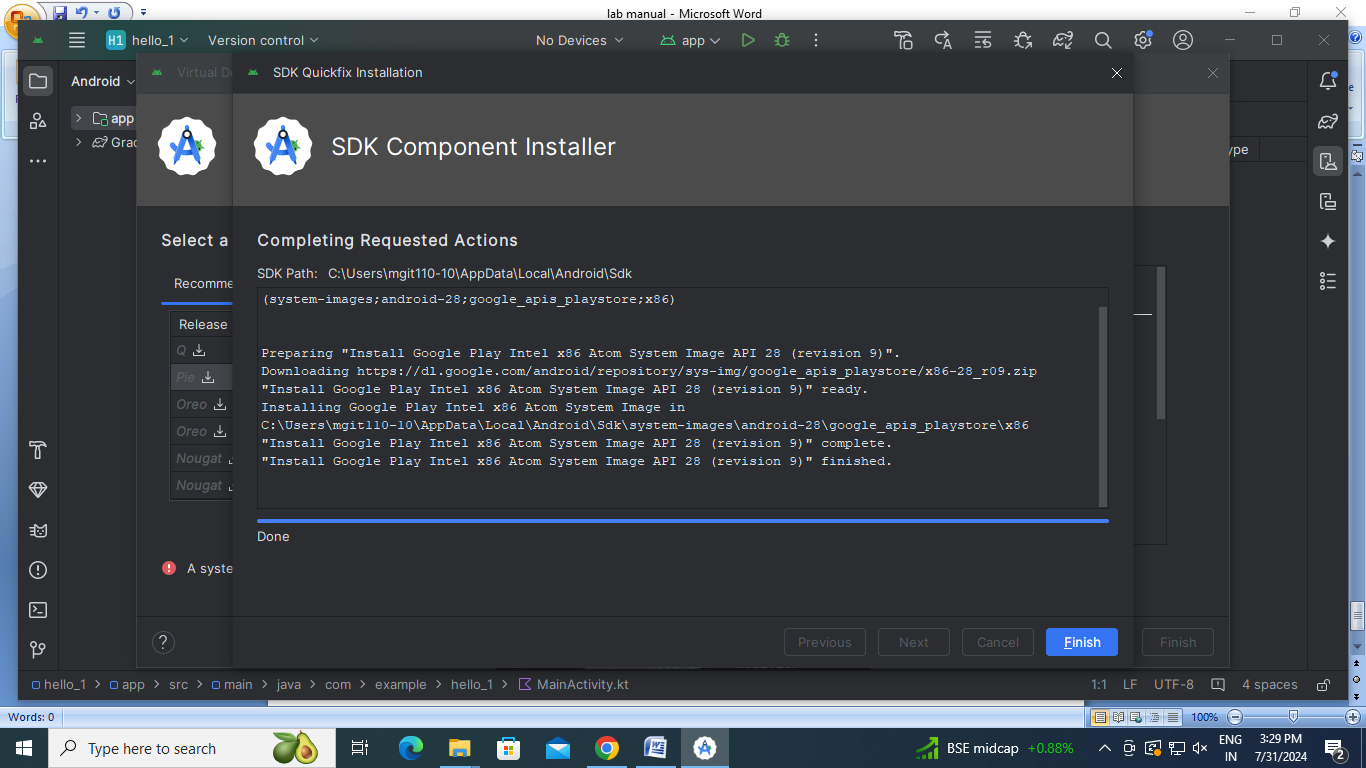


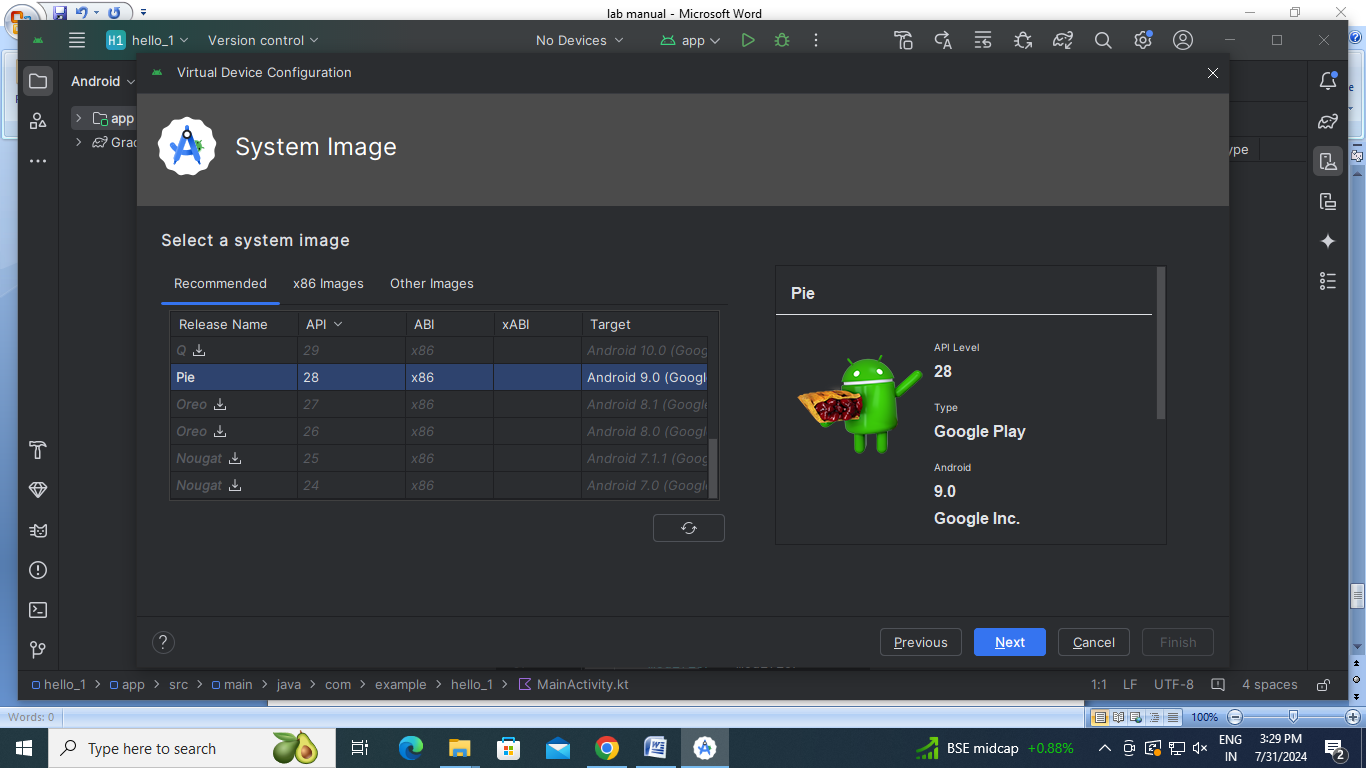


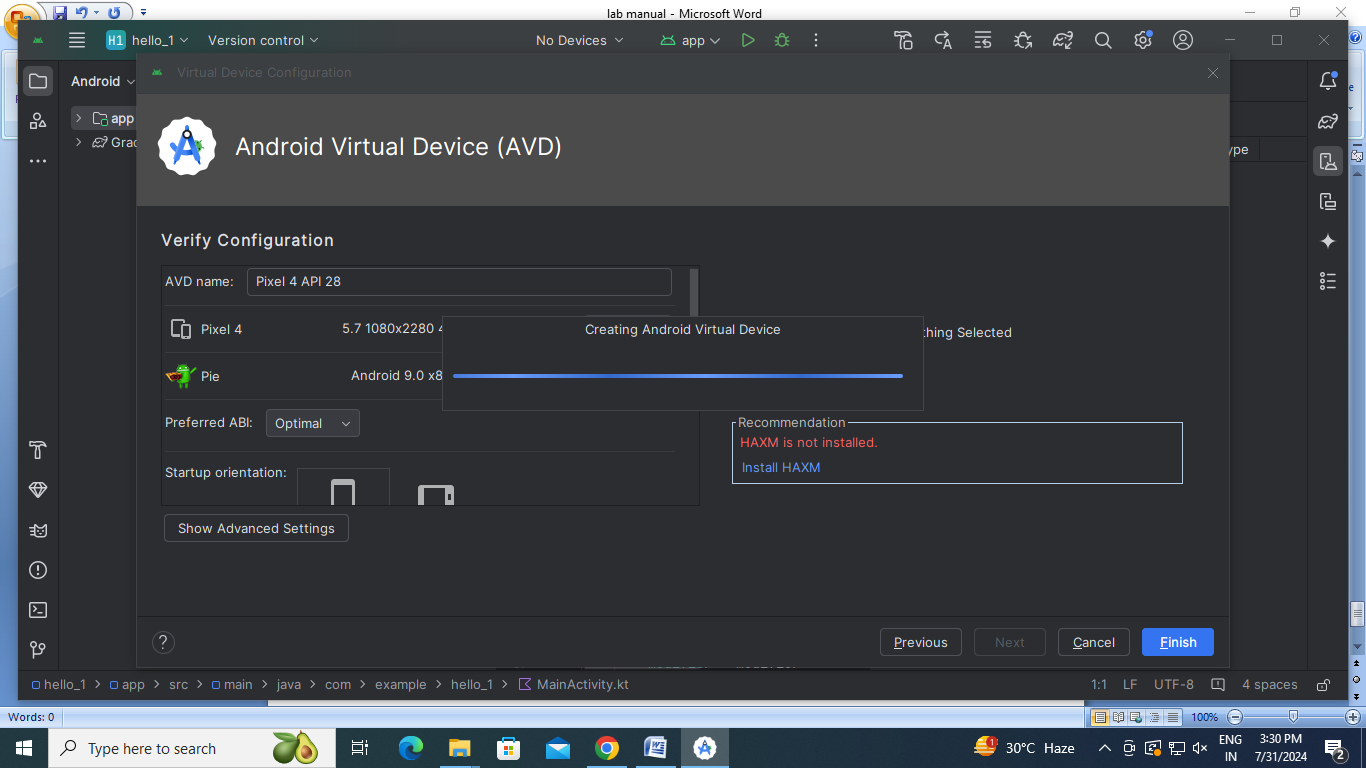












**1 b) Write a simple Dart program to understand the language basics**

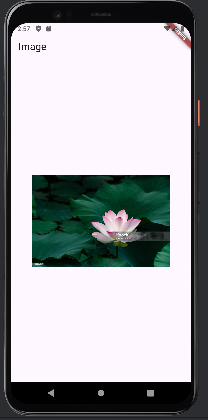
void main() {  
 *// Define three static numbers* int num1 = 10;  
 int num2 = 20;  
 int num3 = 30;  
  
 *// Calculate the sum of the three numbers* int sum = num1 + num2 + num3;  
  
 *// Print the result* print('The sum of $num1, $num2, and $num3 is $sum.');  
}



**2 a) Explore various Flutter widgets (Text, Image, Container, etc.).**

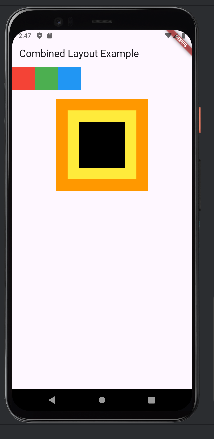
import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatefulWidget {  
 const MyApp({super.key});  
  
 @override  
 State<MyApp> createState() => \_MyAppState();  
}  
  
class \_MyAppState extends State<MyApp> {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('Image'),  
 ),  
 body: Center(  
 child: Container(  
 height: 200,  
 width: 300,  
 child: Image.network('img\_src'),  
 ),  
 ),  
 ));  
 }  
}

output :



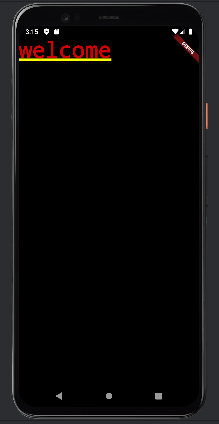
**2 b) Implement different layout structures using Row, Column, and Stack widgets.**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(title: Text('Combined Layout Example')),  
 body: Center(  
 child: Column(  
 *//mainAxisAlignment: MainAxisAlignment.center,* children: <Widget>[  
 Row(  
 *//mainAxisAlignment: MainAxisAlignment.spaceAround,* children: <Widget>[  
 Container(color: Colors.*red*, width: 50, height: 50),  
 Container(color: Colors.*green*, width: 50, height: 50),  
 Container(color: Colors.*blue*, width: 50, height: 50),  
 ],  
 ),  
 SizedBox(height: 20),  
 Stack(  
 alignment: Alignment.*center*,  
 children: <Widget>[  
 Container(color: Colors.*red*, width: 200, height: 200),  
 Container(color: Colors.*green*, width: 150, height: 150),  
 Container(color: Colors.*blue*, width: 100, height: 100),  
 ],  
 ),  
 ],  
 ),  
 ),  
 ),  
 );  
 }  
}



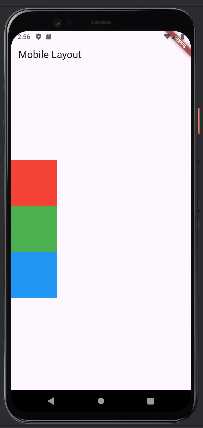
**3 a) Design a responsive UI that adapts to different screen sizes**

import 'package:flutter/material.dart';  
  
*//main function*void main() {  
 runApp(MyApp()); *//leets change it*}  
  
*//stateless widget class*class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return SafeArea(  
 child: MaterialApp(  
 home: Text('AIML'),  
 ),  
 );  
 }  
}



**3 b) Implement media queries and breakpoints for responsiveness.**

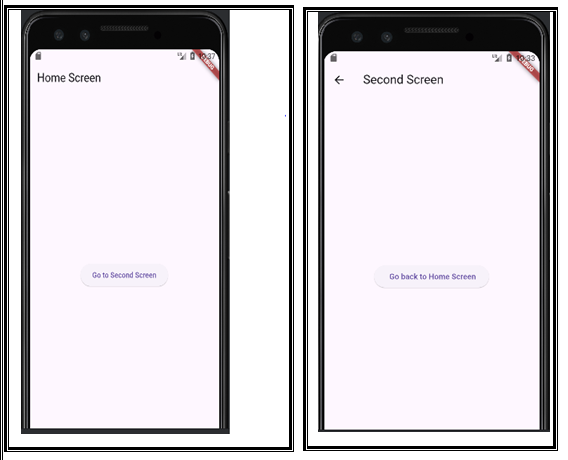
import 'package:flutter/material.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: ResponsiveLayout(),  
    );  
  }  
}  
  
class ResponsiveLayout extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    var mediaQueryData = MediaQuery.of(context);  
    var screenWidth = mediaQueryData.size.width;  
  
    if (screenWidth < 600) {  
      return Scaffold(  
        appBar: AppBar(title: Text('Mobile Layout')),  
        body: \_buildNarrowContainers(),  
      );  
    } else if (screenWidth < 1200) {  
      return Scaffold(  
        appBar: AppBar(title: Text('Tablet Layout')),  
        body: \_buildMediumContainers(),  
      );  
    } else {  
      return Scaffold(  
        appBar: AppBar(title: Text('Desktop Layout')),  
        body: \_buildWideContainers(),  
      );  
    }  
  }  
  
  Widget \_buildNarrowContainers() {  
    return Column(  
      mainAxisAlignment: MainAxisAlignment.center,  
      children: <Widget>[  
        Container(color: Colors.red, width: 100, height: 100),  
        Container(color: Colors.green, width: 100, height: 100),  
        Container(color: Colors.blue, width: 100, height: 100),  
      ],  
    );  
  }  
  
  Widget \_buildMediumContainers() {  
    return Row(  
      mainAxisAlignment: MainAxisAlignment.center,  
      children: <Widget>[  
        Container(color: Colors.red, width: 100, height: 100),  
        Container(color: Colors.green, width: 100, height: 100),  
        Container(color: Colors.blue, width: 100, height: 100),  
      ],  
    );  
  }  
  
  Widget \_buildWideContainers() {  
    return GridView.count(  
      crossAxisCount: 3,  
      mainAxisSpacing: 10,  
      crossAxisSpacing: 10,  
      children: <Widget>[  
        Container(color: Colors.red, width: 100, height: 100),  
        Container(color: Colors.green, width: 100, height: 100),  
        Container(color: Colors.blue, width: 100, height: 100),  
      ],  
    );  
  }  
}

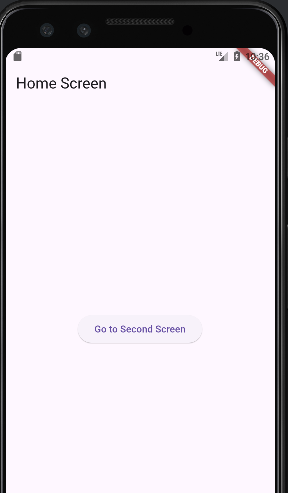


**4 a) Set up navigation between different screens using Navigator.**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
*//1st MyApp class - hub class*class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: HomeScreen(),  
 );  
 }  
}  
  
*//Class HomeScreen- first page in app*class HomeScreen extends StatelessWidget {  
 const HomeScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Home screen'),  
 ),  
 body: Center(  
 child: ElevatedButton(  
 onPressed: () {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(builder: (context) => SecondScreen()),  
 );  
 },  
 child: Text('Go to Second Screen'),  
 ),  
 ),  
 );  
 }  
}  
  
*//3rd claas SecondScreen- second page*class SecondScreen extends StatelessWidget {  
 const SecondScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Second screen'),  
 ),  
 body: ElevatedButton(  
 onPressed: () {  
 Navigator.*pop*(context);  
 },  
 child: Text('Go back to Home screen'),  
 ),  
 );  
 }

}





**4 b) Implement navigation with named routes.**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
*//1st MyApp class - hub class*class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 initialRoute: '/',  
 routes: {  
 '/': (context) => HomeScreen(),  
 '/second': (context) => SecondScreen(),  
 },  
 );  
 }  
}  
  
*//Class HomeScreen- first page in app*class HomeScreen extends StatelessWidget {  
 const HomeScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Home screen'),  
 ),  
 body: Center(  
 child: ElevatedButton(  
 onPressed: () {  
 Navigator.*pushNamed*(context, '/second');  
 },  
 child: Text('Go to Second Screen'),  
 ),  
 ),  
 );  
 }  
}  
  
*//3rd claas SecondScreen- second page*class SecondScreen extends StatelessWidget {  
 const SecondScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Second screen'),  
 ),  
 body: ElevatedButton(  
 onPressed: () {  
 Navigator.*pop*(context);  
 },  
 child: Text('Go back to Home screen'),  
 ),  
 );  
 }

|  |
| --- |
|  |

|  |
| --- |
|  |

}

**5 a) Learn about stateful and stateless widgets**

import 'package:flutter/material.dart';  
  
*//main function*void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatefulWidget {  
 const MyApp({super.key});  
 @override  
 State<MyApp> createState() => \_MyAppState();  
}  
class \_MyAppState extends State<MyApp> {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Column(  
 children: [Text('data-1'), Text('data-2'), Text('data-3')],  
 ),  
 );

}

|  |
| --- |
|  |

}

5 (A) (2)

import 'package:flutter/material.dart';  
  
*//main function*void main() {  
 runApp(MyApp()); *//leets change it*}  
  
*//stateless widget class*class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Text('FDP'),  
 );  
 }  
}

|  |
| --- |
|  |

5 b)

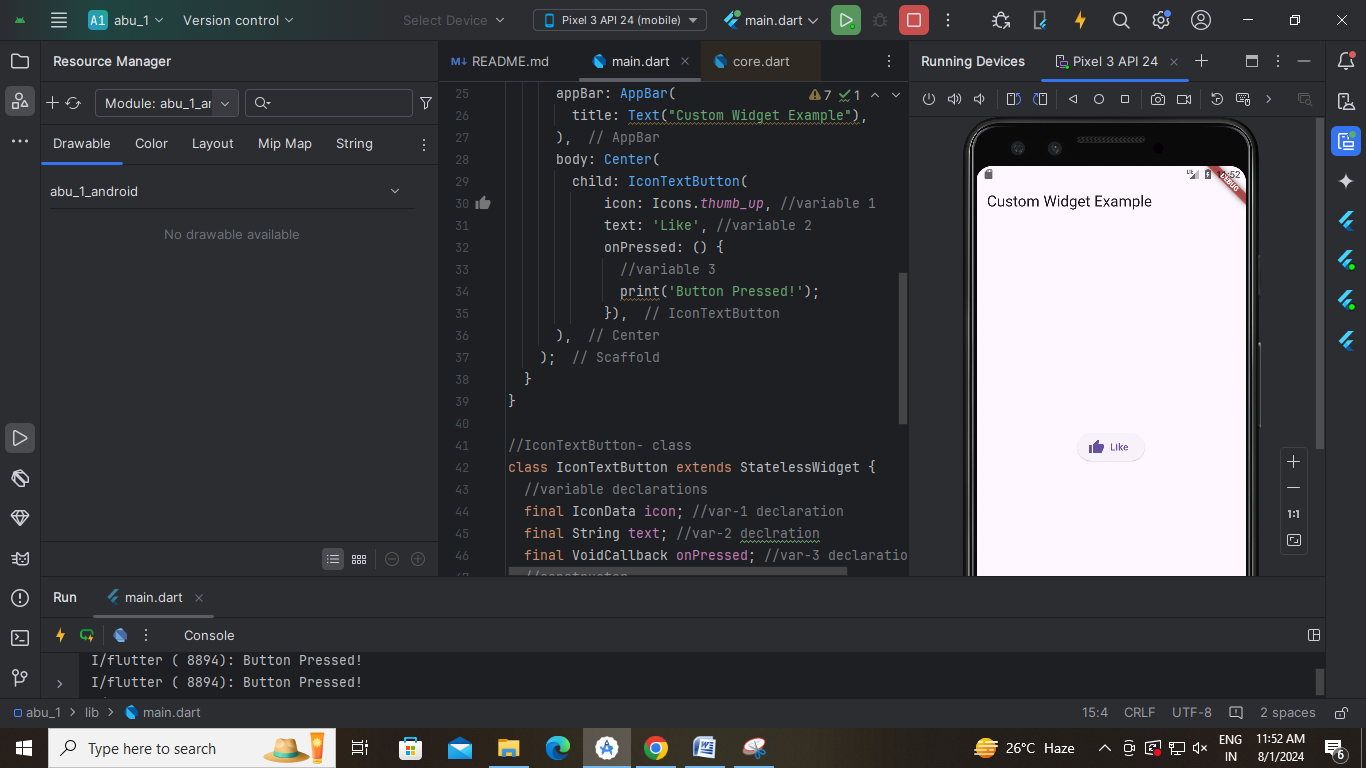
import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatefulWidget {  
 const MyApp({super.key});  
  
 @override  
 State<MyApp> createState() => \_MyAppState();  
}  
  
class \_MyAppState extends State<MyApp> {  
 String value = "Test"; *//variable initialized  
 //fucntion to change the value* void clickMe() {  
 setState(() {  
 value = "Test done";  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('simple Flutter app'),  
 ),  
 body: Center(  
 child: Column(  
 children: [  
 Text('$value'),  
 ],  
 ),  
 ),  
 floatingActionButton: FloatingActionButton(  
 child: Icon(Icons.*add*),  
 onPressed: clickMe,  
 ),  
 ),  
 );  
 }  
}

|  |  |
| --- | --- |
|  |  |

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

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: HomeScreen(),  
 );  
 }  
}  
  
*//Home screen - class*class HomeScreen extends StatelessWidget {  
 const HomeScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text("Custom Widget Example"),  
 ),  
 body: Center(  
 child: IconTextButton(  
 icon: Icons.*thumb\_up*, *//variable 1* text: 'Like', *//variable 2* onPressed: () {  
 *//variable 3* print('Button Pressed!');  
 }),  
 ),  
 );  
 }  
}  
  
*//IconTextButton- class*class IconTextButton extends StatelessWidget {  
 *//variable declarations* final IconData icon; *//var-1 declaration* final String text; *//var-2 declration* final VoidCallback onPressed; *//var-3 declaration  
 //constructor* IconTextButton({  
 required this.icon,  
 required this.text,  
 required this.onPressed,  
 });  
  
 @override  
 Widget build(BuildContext context) {  
 return ElevatedButton.icon(  
 onPressed: onPressed,  
 icon: Icon(icon),  
 label: Text(text),  
 );  
 }  
}

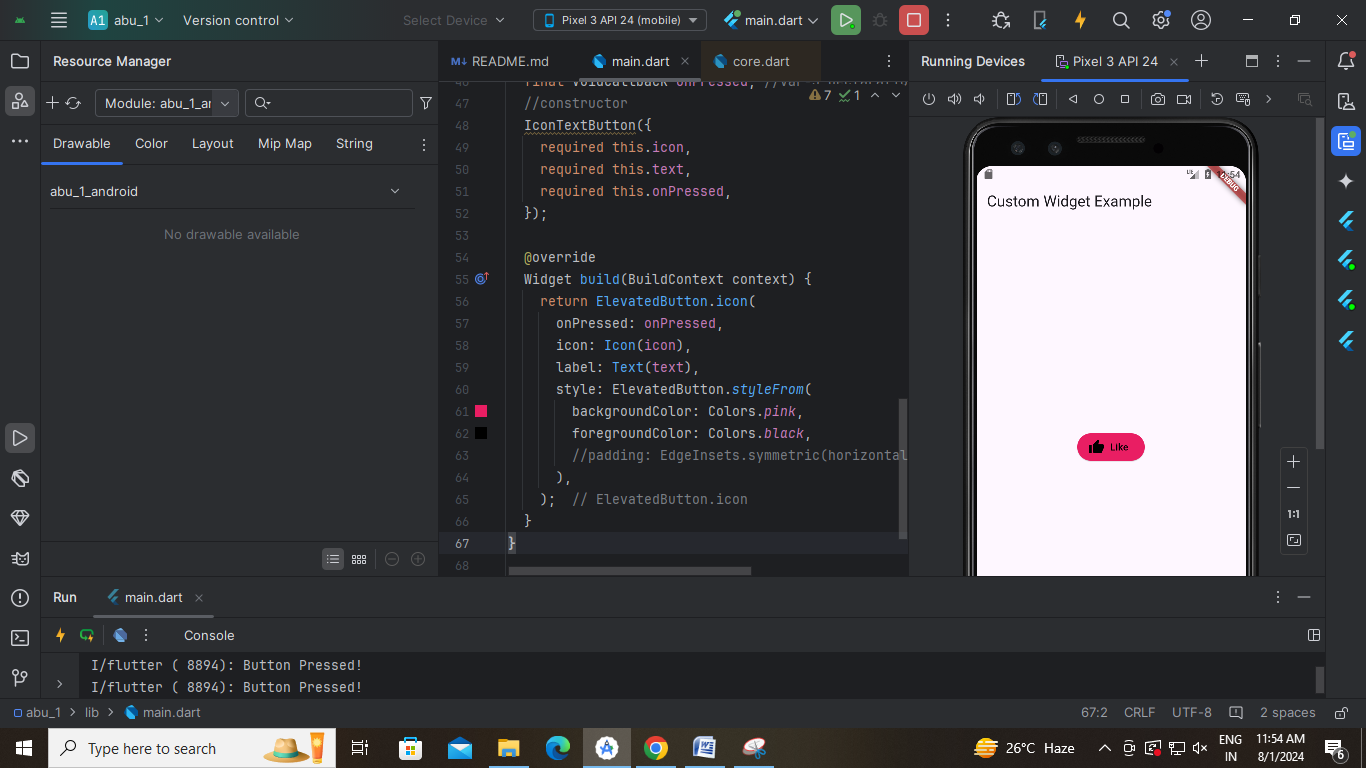
|  |
| --- |
|  |



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

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: HomeScreen(),  
 );  
 }  
}  
  
*//Home screen - class*class HomeScreen extends StatelessWidget {  
 const HomeScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text("Custom Widget Example"),  
 ),  
 body: Center(  
 child: IconTextButton(  
 icon: Icons.*thumb\_up*, *//variable 1* text: 'Like', *//variable 2* onPressed: () {  
 *//variable 3* print('Button Pressed!');  
 }),  
 ),  
 );  
 }  
}  
  
*//IconTextButton- class*class IconTextButton extends StatelessWidget {  
 *//variable declarations* final IconData icon; *//var-1 declaration* final String text; *//var-2 declration* final VoidCallback onPressed; *//var-3 declaration  
 //constructor* IconTextButton({  
 required this.icon,  
 required this.text,  
 required this.onPressed,  
 });  
  
 @override  
 Widget build(BuildContext context) {  
 return ElevatedButton.icon(  
 onPressed: onPressed,  
 icon: Icon(icon),  
 label: Text(text),  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*pink*,  
 foregroundColor: Colors.*black*,  
 *//padding: EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),* ),  
 );  
 }  
}

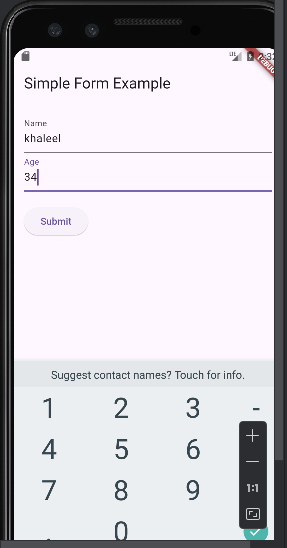
|  |
| --- |
|  |

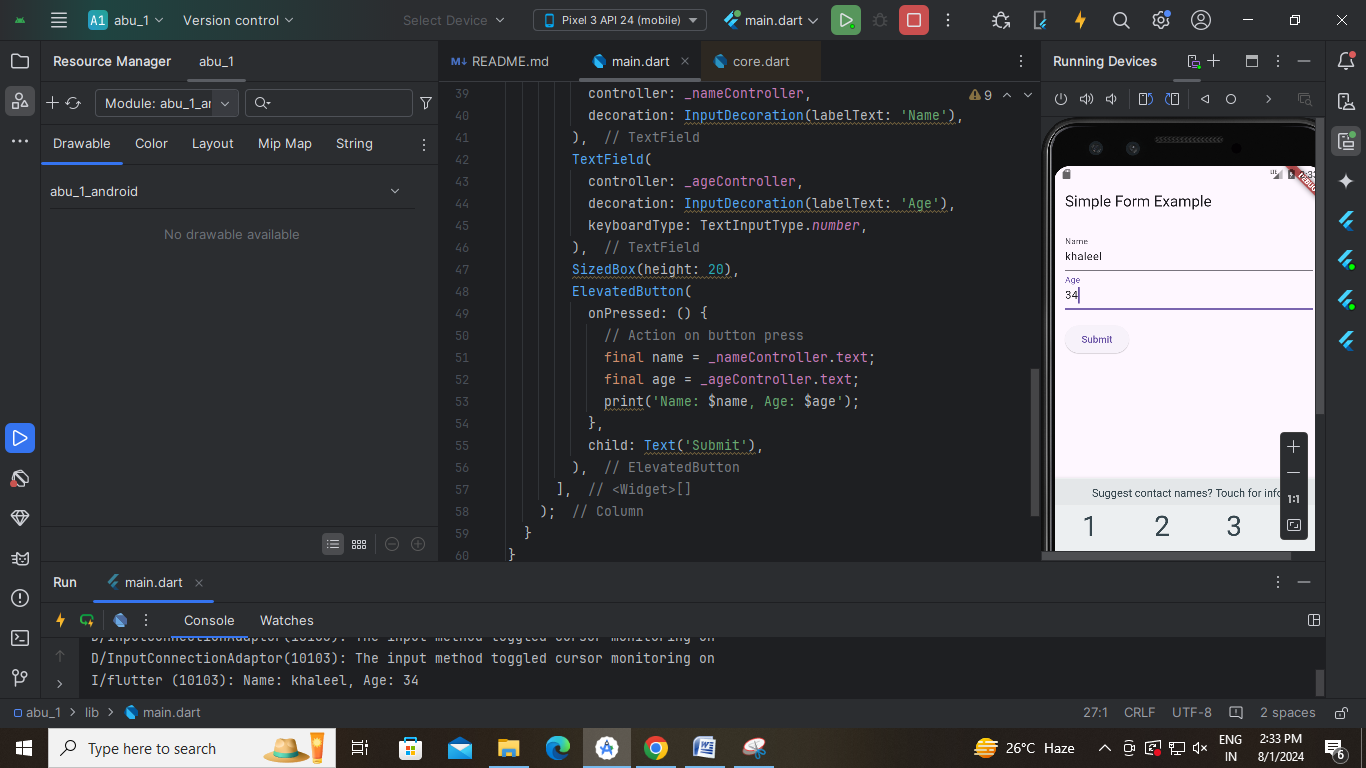


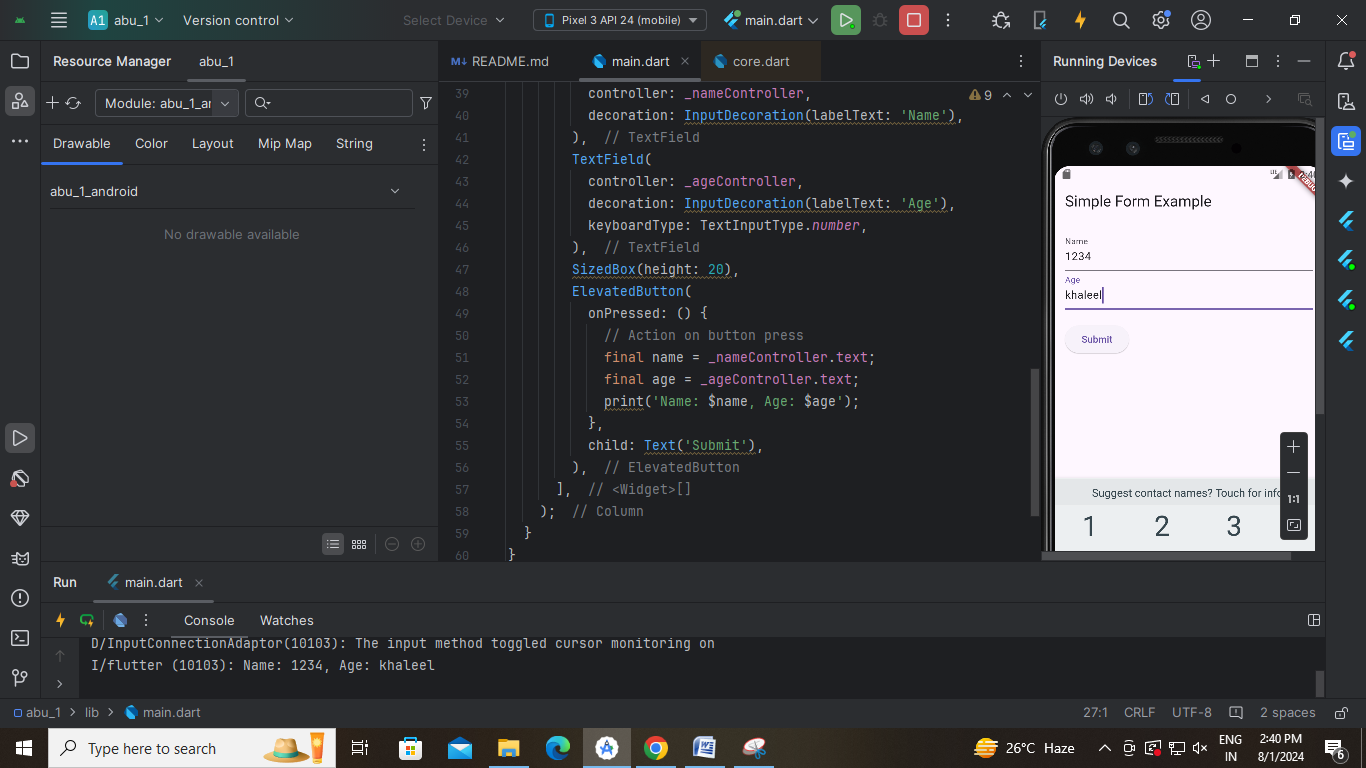
**7 a) Design a form with various input fields.**

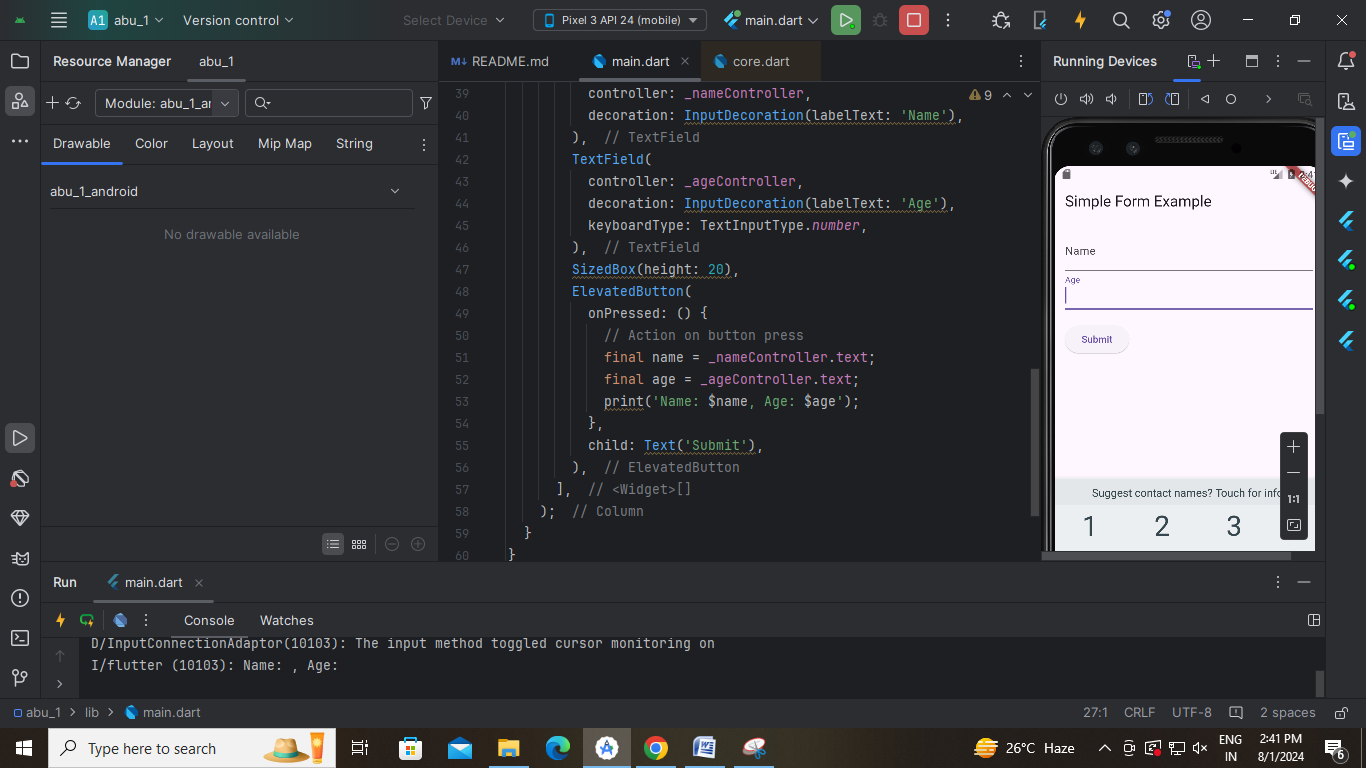
import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('Simple Form Example'),  
 ),  
 body: Padding(  
 padding: const EdgeInsets.all(16.0),  
 child: SimpleForm(),  
 ),  
 ),  
 );  
 }  
}  
  
class SimpleForm extends StatefulWidget {  
 @override  
 \_SimpleFormState createState() => \_SimpleFormState();  
}  
  
class \_SimpleFormState extends State<SimpleForm> {  
 final \_nameController = TextEditingController();  
 final \_ageController = TextEditingController();  
  
 @override  
 Widget build(BuildContext context) {  
 return Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: <Widget>[  
 TextField(  
 controller: \_nameController,  
 decoration: InputDecoration(labelText: 'Name'),  
 ),  
 TextField(  
 controller: \_ageController,  
 decoration: InputDecoration(labelText: 'Age'),  
 keyboardType: TextInputType.*number*,  
 ),  
 SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: () {  
 *// Action on button press* final name = \_nameController.text;  
 final age = \_ageController.text;  
 print('Name: $name, Age: $age');  
 },  
 child: Text('Submit'),  
 ),  
 ],  
 );  
 }  
}

|  |
| --- |
|  |





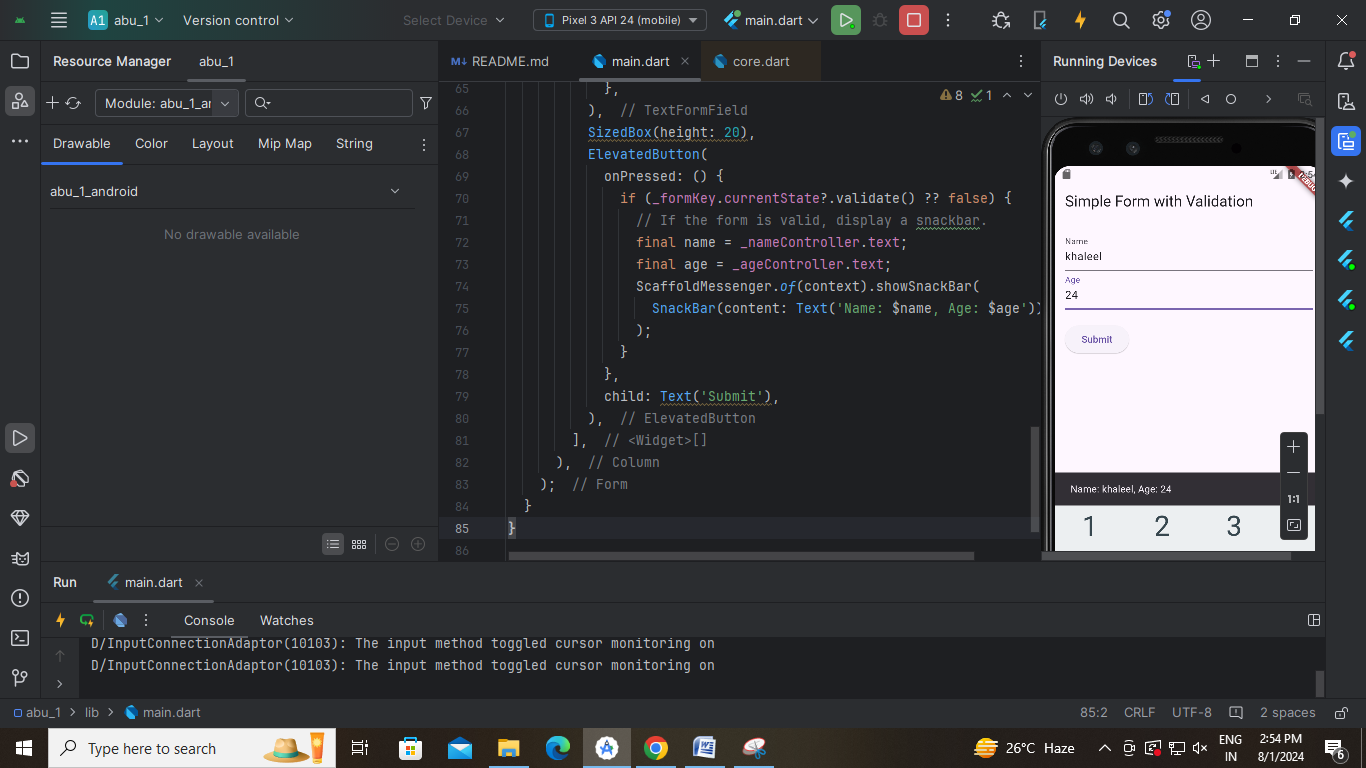


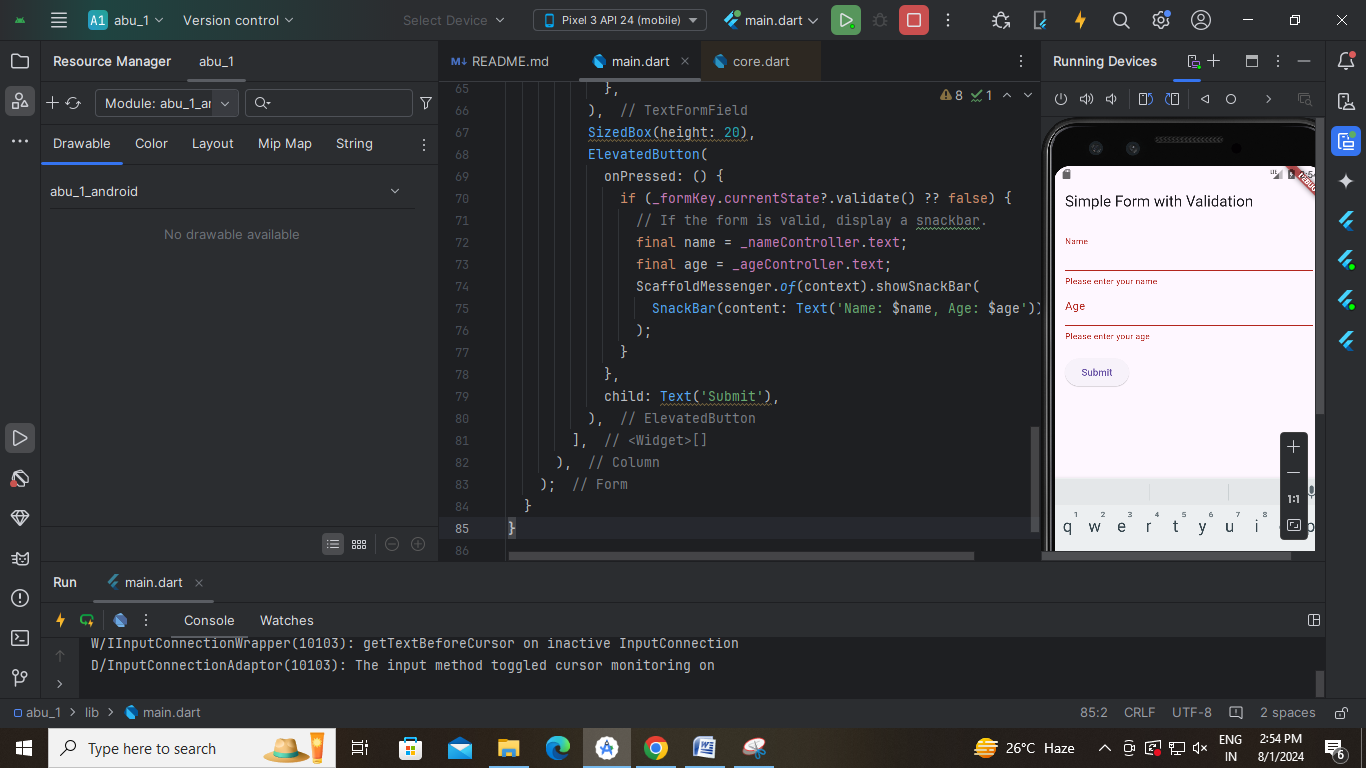


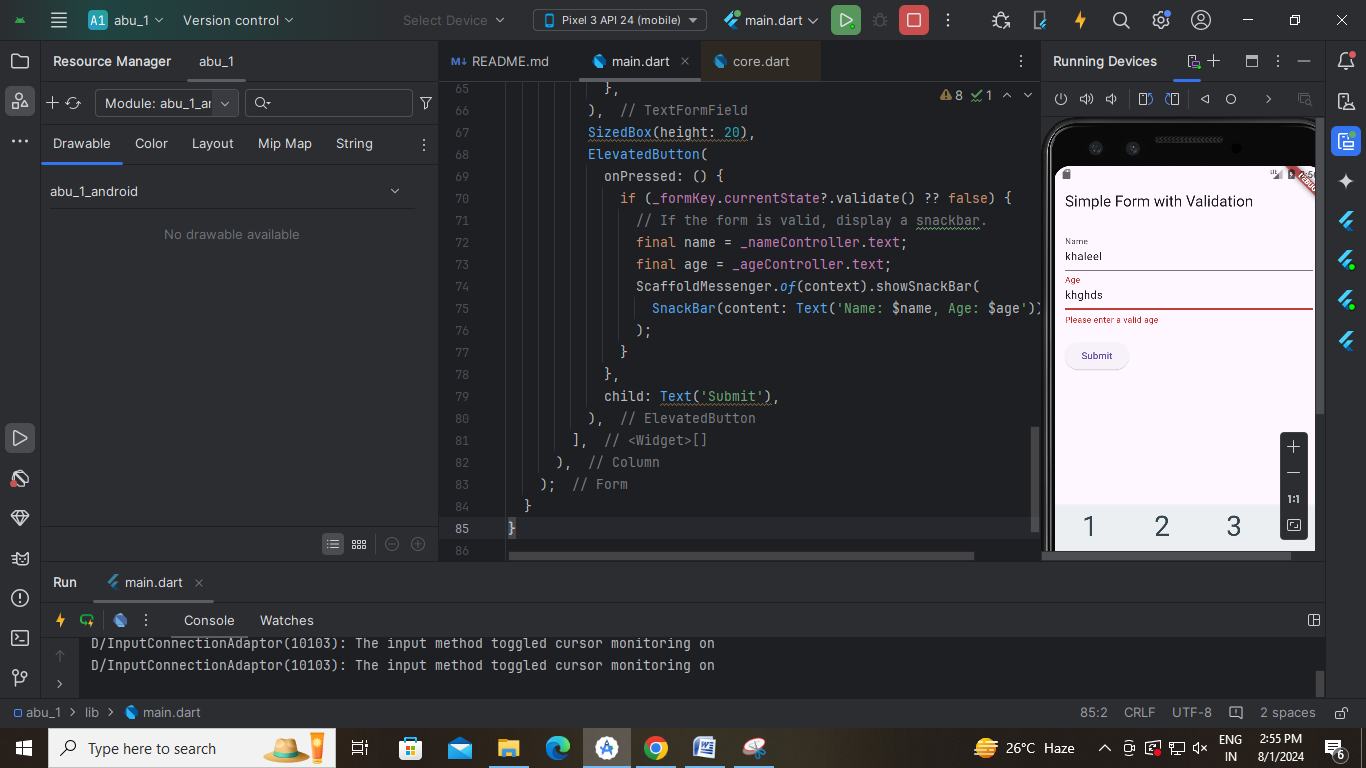
**7 b) Implement form validation and error handling.**

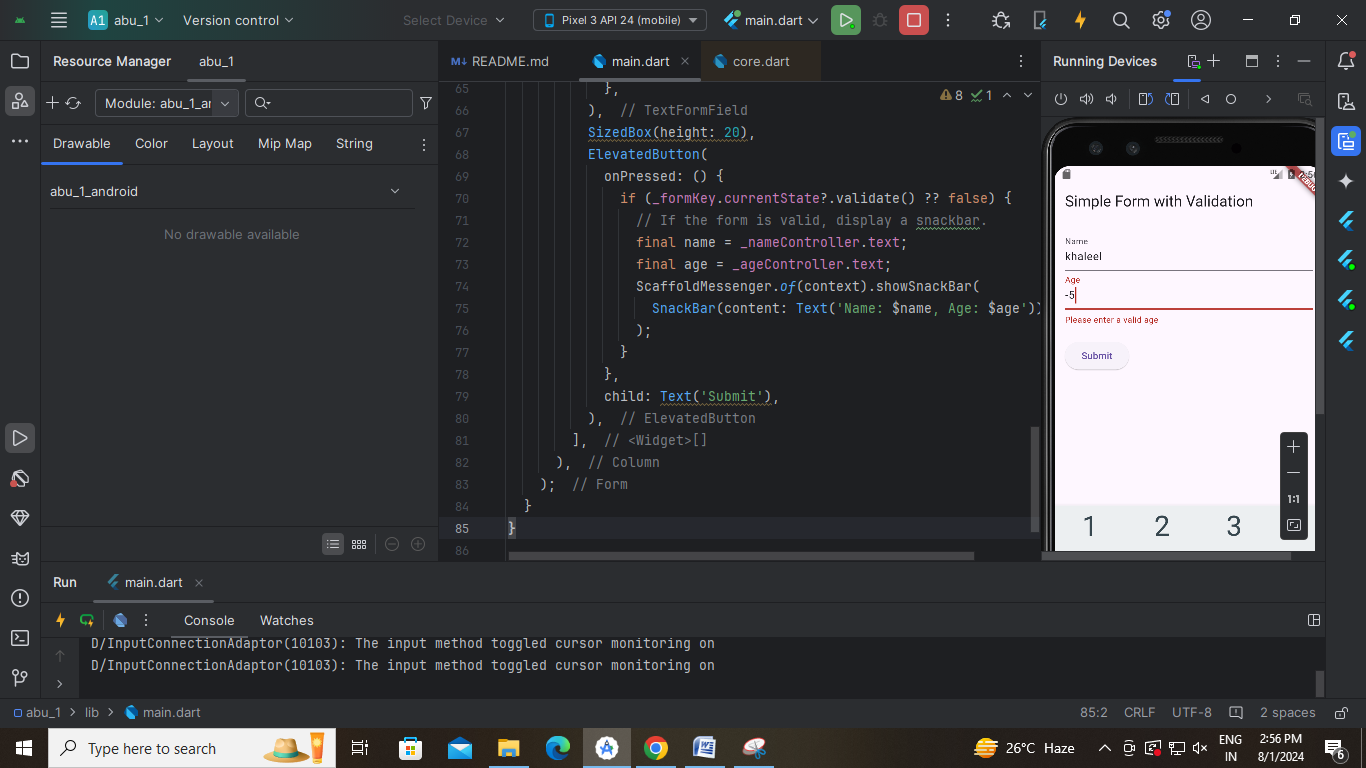
import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('Simple Form with Validation'),  
 ),  
 body: Padding(  
 padding: const EdgeInsets.all(16.0),  
 child: SimpleForm(),  
 ),  
 ),  
 );  
 }  
}  
  
class SimpleForm extends StatefulWidget {  
 @override  
 \_SimpleFormState createState() => \_SimpleFormState();  
}  
  
class \_SimpleFormState extends State<SimpleForm> {  
 final \_formKey = GlobalKey<FormState>();  
  
 final \_nameController = TextEditingController();  
 final \_ageController = TextEditingController();  
  
 @override  
 Widget build(BuildContext context) {  
 return Form(  
 key: \_formKey,  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: <Widget>[  
 TextFormField(  
 controller: \_nameController,  
 decoration: InputDecoration(labelText: 'Name'),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Please enter your name';  
 }  
 return null;  
 },  
 ),  
 TextFormField(  
 controller: \_ageController,  
 decoration: InputDecoration(labelText: 'Age'),  
 keyboardType: TextInputType.*number*,  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Please enter your age';  
 }  
 final age = int.*tryParse*(value);  
 if (age == null || age <= 0) {  
 return 'Please enter a valid age';  
 }  
 return null;  
 },  
 ),  
 SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: () {  
 if (\_formKey.currentState?.validate() ?? false) {  
 *// If the form is valid, display a snackbar.* final name = \_nameController.text;  
 final age = \_ageController.text;  
 ScaffoldMessenger.*of*(context).showSnackBar(  
 SnackBar(content: Text('Name: $name, Age: $age')),  
 );  
 }  
 },  
 child: Text('Submit'),  
 ),  
 ],  
 ),  
 );  
 }  
}

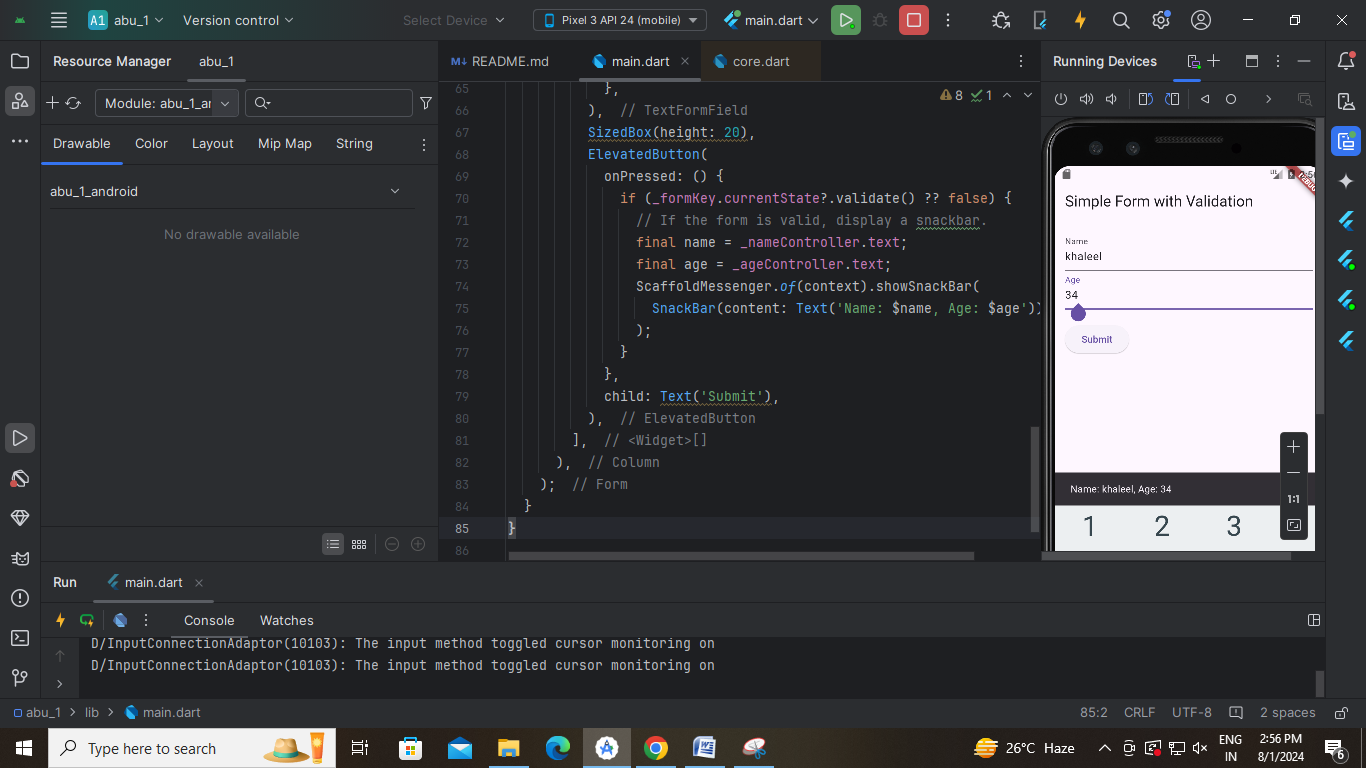
|  |  |  |
| --- | --- | --- |
|  |  |  |











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

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: HomeScreen(),  
 );  
 }  
}  
  
class HomeScreen extends StatelessWidget {  
 const HomeScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text("Custon widget with animation"),  
 ),  
 body: Center(  
 child: AnimatedIconTextButton(  
 icon: Icons.*thumb\_up*,  
 text: 'Like',  
 onPressed: () {  
 print("Button Pressed!");  
 }),  
 ),  
 );  
 }  
}  
  
class AnimatedIconTextButton extends StatefulWidget {  
 final IconData icon;  
 final String text;  
 final VoidCallback onPressed;  
 *//constructors* const AnimatedIconTextButton({  
 Key? key,  
 required this.icon,  
 required this.text,  
 required this.onPressed,  
 }) : super(key: key);  
  
 @override  
 State<AnimatedIconTextButton> createState() => \_AnimatedIconTextButtonState();  
}  
  
class \_AnimatedIconTextButtonState extends State<AnimatedIconTextButton> {  
 bool \_isPressed = false;  
 void \_handlePress() {  
 setState(() {  
 \_isPressed = !\_isPressed; *//on->off//off->on* });  
 widget.onPressed();  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return AnimatedContainer(  
 duration: Duration(milliseconds: 300),  
 width: \_isPressed ? 200 : 150,  
 height: \_isPressed ? 100 : 70,  
 child: ElevatedButton.icon(  
 onPressed: \_handlePress,  
 icon: Icon(widget.icon),  
 label: Text(widget.text),  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*blue*,  
 foregroundColor: Colors.*green*,  
 ),  
 ),  
 ); } }

|  |
| --- |
|  |

|  |
| --- |
|  |

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

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: HomeScreen(),  
 );  
 }  
}  
  
class HomeScreen extends StatelessWidget {  
 const HomeScreen({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text("Custon widget with animation"),  
 ),  
 body: Center(  
 child: FadeIconTextButton(  
 icon: Icons.*thumb\_up*,  
 text: 'Like',  
 onPressed: () {  
 print("Button Pressed!");  
 }),  
 ),  
 );  
 }  
}  
  
class FadeIconTextButton extends StatefulWidget {  
 final IconData icon;  
 final String text;  
 final VoidCallback onPressed;  
 *//constructors* const FadeIconTextButton({  
 Key? key,  
 required this.icon,  
 required this.text,  
 required this.onPressed,  
 }) : super(key: key);  
  
 @override  
 State<FadeIconTextButton> createState() => \_FadeIconTextButtonState();  
}  
  
class \_FadeIconTextButtonState extends State<FadeIconTextButton>  
 with SingleTickerProviderStateMixin {  
 late AnimationController \_controller;  
 late Animation<double> \_animation;  
 @override  
 void initState() {  
 super.initState();  
 \_controller = AnimationController(  
 duration: const Duration(milliseconds: 500),  
 vsync: this,  
 );  
 \_animation = CurvedAnimation(  
 parent: \_controller,  
 curve: Curves.*easeInOut*,  
 );  
 }  
  
 void \_handlePress() {  
 if (\_controller.isCompleted) {  
 \_controller.reverse();  
 } else {  
 \_controller.forward();  
 }  
 widget.onPressed();  
 }  
  
 @override  
 void dispose() {  
 \_controller.dispose();  
 super.dispose();  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return FadeTransition(  
 opacity: \_animation,  
 child: ElevatedButton.icon(  
 onPressed: \_handlePress,  
 icon: Icon(widget.icon),  
 label: Text(widget.text),  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*amber*,  
 foregroundColor: Colors.*black*,  
 ),  
 ),  
 );  
 }  
}

|  |
| --- |
|  |

|  |
| --- |
|  |

**9a) Fetch data from a REST API. b) Display the fetched data in a meaningful way in the UI.**

import 'package:flutter/material.dart';  
import 'package:http/http.dart' as http;  
import 'dart:convert';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: HomeScreen(),  
 );  
 }  
}  
  
class HomeScreen extends StatefulWidget {  
 @override  
 \_HomeScreenState createState() => \_HomeScreenState();  
}  
  
class \_HomeScreenState extends State<HomeScreen> {  
 late Future<List<Post>> \_futurePosts;  
  
 @override  
 void initState() {  
 super.initState();  
 \_futurePosts = fetchPosts();  
 }  
  
 Future<List<Post>> fetchPosts() async {  
 final response =  
 await http.get(Uri.*parse*('<https://jsonplaceholder.typicode.com/posts>'));  
  
 if (response.statusCode == 200) {  
 List jsonResponse = json.decode(response.body);  
 return jsonResponse.map((post) => Post.fromJson(post)).toList();  
 } else {  
 throw Exception('Failed to load posts');  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Fetch Data Example'),  
 ),  
 body: Center(  
 child: FutureBuilder<List<Post>>(  
 future: \_futurePosts,  
 builder: (context, snapshot) {  
 if (snapshot.connectionState == ConnectionState.waiting) {  
 return CircularProgressIndicator();  
 } else if (snapshot.hasError) {  
 return Text('${snapshot.error}');  
 } else if (snapshot.hasData) {  
 List<Post>? posts = snapshot.data;  
 return ListView.builder(  
 itemCount: posts!.length,  
 itemBuilder: (context, index) {  
 return ListTile(  
 title: Text(posts[index].title),  
 subtitle: Text(posts[index].body),  
 );  
 },  
 );  
 } else {  
 return Text('No data found');  
 }  
 },  
 ),  
 ),  
 );  
 }  
}  
  
class Post {  
 final int id;  
 final String title;  
 final String body;  
  
 Post({required this.id, required this.title, required this.body});  
  
 factory Post.fromJson(Map<String, dynamic> json) {  
 return Post(  
 id: json['id'],  
 title: json['title'],  
 body: json['body'],  
 );  
 }  
}

|  |
| --- |
|  |

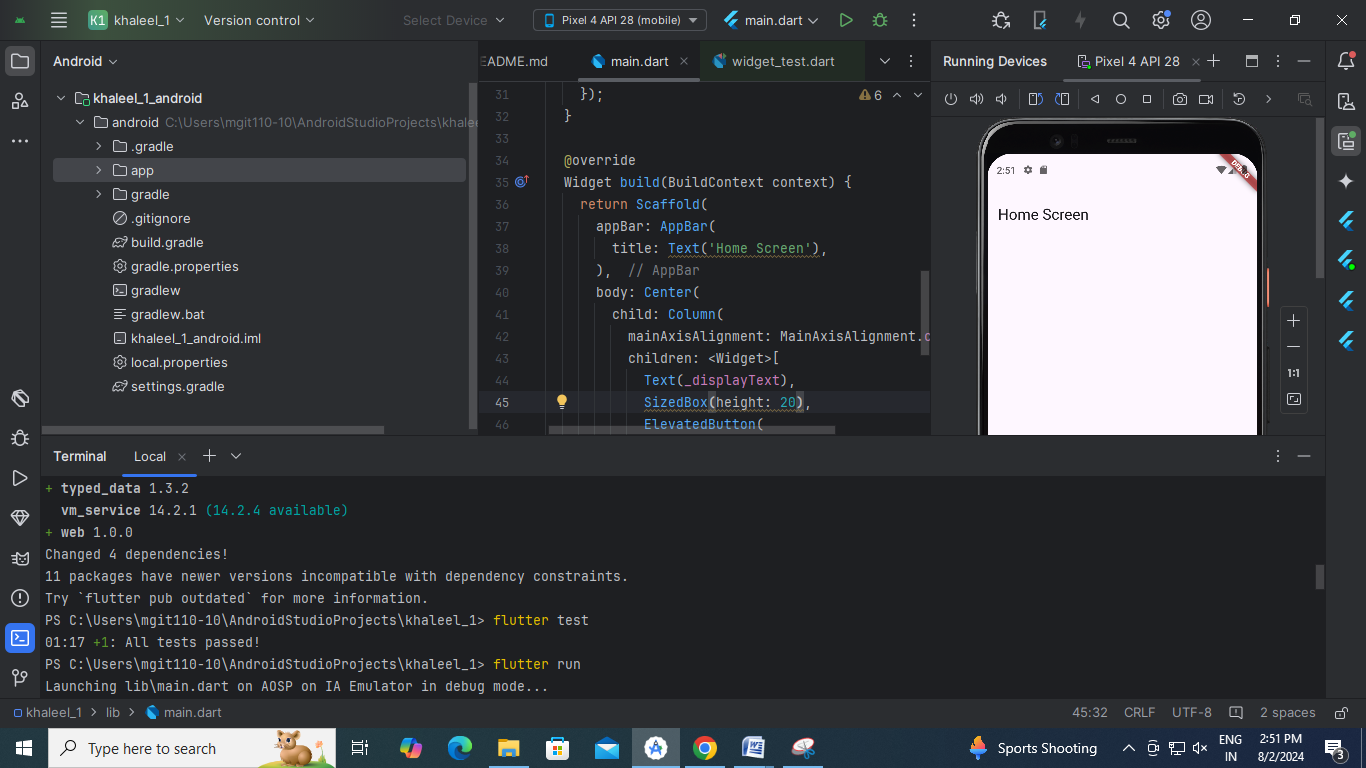
**10 a) Write unit tests for UI components.**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Flutter Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: HomeScreen(),  
 );  
 }  
}  
  
class HomeScreen extends StatefulWidget {  
 @override  
 \_HomeScreenState createState() => \_HomeScreenState();  
}  
  
class \_HomeScreenState extends State<HomeScreen> {  
 String \_displayText = 'Hello';  
  
 void \_updateText() {  
 setState(() {  
 \_displayText = 'Button Pressed';  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Home Screen'),  
 ),  
 body: Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(\_displayText),  
 SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: \_updateText,  
 child: Text('Press Me'),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}

**// search” Widget\_test.dart” file by pressing Ctrl+Shift+f and past the bellow code**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:untitled/main.dart'; *//url might change*void main() {  
 testWidgets('text changes', (WidgetTester tester) async {  
 *//build the app* await tester.pumpWidget(MyApp());  
 *//test-1 verifying the initial text'Hello* expect(find.text("Hello"), findsOneWidget);  
 expect(find.text("Button Pressed"), findsNothing);  
 *//test-2 to tap the button* await tester.tap(find.byType(ElevatedButton));  
 await tester.pump();  
 *//test-3 verify the text* expect(find.text("Hello"), findsNothing);  
 expect(find.text("Button Pressed"), findsOneWidget);  
 });  
}

* open terminal and type the following command
* flutter test
* it should show +1: All test passed



**10 b) Use Flutter's debugging tools to identify and fix issues.**

import 'package:flutter/material.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
      ),  
      home: HomeScreen(),  
    );  
  }  
}  
  
class HomeScreen extends StatefulWidget {  
  @override  
  \_HomeScreenState createState() => \_HomeScreenState();  
}  
  
class \_HomeScreenState extends State<HomeScreen> {  
  String \_displayText; // Variable declared but not initialized  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Home Screen'),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: <Widget>[  
            Text(\_displayText), // Error: \_displayText is used before being initialized  
            SizedBox(height: 20),  
            ElevatedButton(  
              onPressed: () {  
                setState(() {  
                  \_displayText = 'Button Pressed';  
                });  
              },  
              child: Text('Press Me'),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}

//corrected code

import 'package:flutter/material.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
      ),  
      home: HomeScreen(),  
    );  
  }  
}  
  
class HomeScreen extends StatefulWidget {  
  @override  
  \_HomeScreenState createState() => \_HomeScreenState();  
}  
  
class \_HomeScreenState extends State<HomeScreen> {  
  String \_displayText="; // Variable declared and initialized  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Home Screen'),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: <Widget>[  
            Text(\_displayText), // Error: \_displayText is used before being initialized  
            SizedBox(height: 20),  
            ElevatedButton(  
              onPressed: () {  
                setState(() {  
                  \_displayText = 'Button Pressed';  
                });  
              },  
              child: Text('Press Me'),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}

