|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **III B.Tech I Semester** | **Tinkering Lab**  **(*User Interface Design using Flutter*)** | **L** | **T** | **P** | **C** |
| **0** | **0** | **3** | **1.5** |

**List of Experiments:**

Students need to implement the following experiments

1. a) Install Flutter and Dart SDK.

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

c) Write a Dart console program that prints your name, checks age with conditionals, uses a loop to count from 1 to 5, and defines a function to return the sum of two numbers.

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

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

c) Create a Flutter app with a Text widget showing a counter value, and an ElevatedButton that increments the counter using setState() .

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

b) Implement media queries and breakpoints for responsiveness.

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

b) Implement navigation with named routes.

5. a) Learn about stateful and stateless widgets.

b) Implement state management using set State and Provider.

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

b) Apply styling using themes and custom styles.

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

b) Implement form validation and error handling.

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

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

9. a) Fetch data from a REST API.

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

10. a) Write unit tests for UI components.

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

**Text Book:**

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.

2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st Edition, Apres.

**3. Reference Books:**

Flutter Complete Reference: The Ultimate Reference for Dart and Flutter Author(s) Alberto Miola Publisher: Independently published;

**Online & Web resources:**

https://onlinecourses.nptel.ac.in/noc21\_ar05/previewOnline Resources:

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

**AIM:** To install Flutter and the Dart SDK, you can follow these steps:

a) Download Flutter: Visit the Flutter website's Get Started page and download the Flutter SDK for your operating system (Windows, macOS, or Linux).

b) Extract the Flutter SDK: After downloading, extract the contents of the compressed file to a location on your computer where you want to store the Flutter SDK. For example, you can extract it to C:\flutter on Windows, /Users/<your-username>/flutter on macOS, or ~/flutter on Linux.

c) Add Flutter to your PATH: Update your system's PATH variable to include the Flutter bin

directory. This step allows you to execute Flutter commands from any directory in your terminal or command prompt. The precise steps for updating the PATH vary depending on your operating system.

**Windows:** From the Start search bar, type 'env' and select 'Edit the system environment

variables'. Click on 'Environment Variables'. Under 'System Variables', find the 'Path' variable, select it, and click 'Edit'. Click 'New' and add the path to the bin directory insidethe Flutter directory (e.g., C:\flutter\bin). Click 'OK' on all open dialogs to save your changes.

**macOS and Linux:** Open a terminal window.

Run the following command to open the profile file associated with your terminal

(.bash\_profile, .bashrc, .zshrc, or similar):

nano ~/.bash\_profile Add the following line at the end of the file:

export PATH="$PATH:/path/to/flutter/bin"

Press Ctrl + X to exit, then Y to save changes, and Enter to confirm.

d) Verify the Flutter installation: Open a new terminal window, and run the following

command to verify that Flutter is properly installed: flutter –version. This command should display the Flutter version and other relevant information if the installation was successful.

e) Install Flutter dependencies: Depending on your development environment, you may need to install additional dependencies, such as Android Studio to fully set up your Flutter development environment.

f) Download Dart SDK (if not bundled with Flutter): Flutter comes with the Dart SDK

bundled, so if you've installed Flutter, you should have the Dart SDK as well. However, if you need to install Dart separately, you can download it from the Dart "SDK archive".

**b) Write a simple dart program to understand the language basics.**

import 'package:flutter/material.dart';

void main() {

runApp(Abc());

}

class Abc extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Def(),

);

}

}

class Def extends StatelessWidget {

const Def({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Welcome"),

backgroundColor: Colors.purple,

),

body: Column(

children: [

//Widgets

],

),

);

}

**C). Write a Dart console program that prints your name, checks age with conditionals, uses a loop to count from 1 to 5, and defines a function to return the sum of two numbers**.

myNumber = 10;

double myDouble

= 3.14;

String myString = 'Hello World';

bool myBool = true;

// Printing variables

print('My number is: $myNumber');

print('My double is: $myDouble');

print('My string is: $myString');

print('My boolean is: $myBool');

// Basic arithmetic operations int result

= myNumber + 5; print('Result of

addition: $result');

// Conditional statements if

(myBool) { print('myBool is

true');

} else {

print('myBool is false');

}

// Loops

for (int i = 0; i < 5; i++) {

print('Iteration $i');

}

// Lists

List<int> numbers = [1, 2, 3, 4, 5];

print('First element of the list: ${numbers[0]}');

print('Length of the list: ${numbers.length}');

// Maps

Map<String, int> ages = {

'Kiran': 30,

'Raj': 25,

'Alekya': 35,

};

print('Kiran\'s age: ${ages['Kiran']}');

}

}

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

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp(

home: Abc()

);

}

}

classAbc extends StatelessWidget {

constAbc({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Text Widget Example')),

body: Center(

child: Text(

'Hello, Flutter!',

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

),

),

);

}

}

**Image:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp(

home: Abc(),

);

}

}

classAbc extends StatelessWidget {

constAbc({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Image Widget"),

),

body: Image.network('https://picsum.photos/250?image=9'),

);

}

}

**Text and Image:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp(

home: Abc(),

);

}

}

classAbc extends StatelessWidget {

constAbc({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Image Widget"),

),

body: Image.asset('assets/images/aditya.jpg'),

);

}

}

**Container:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp(

home: Abc(),

);

}

}

classAbc extends StatelessWidget {

constAbc({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Container Widget Example')),

body: Center(

child: Container(

width: 200,

height: 200,

padding: EdgeInsets.all(16),

margin: EdgeInsets.all(16),

decoration: BoxDecoration(

color: Colors.blue,

borderRadius: BorderRadius.circular(8),

boxShadow: [

BoxShadow(

color: Colors.black26,

blurRadius: 10,

offset: Offset(2, 2),

),

],

),

child: Center(

child: Text(

'Container',

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

),

),

),

),

);

}

}

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

**Row stack:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp( home:

Scaffold(appBar: AppBar(

title: Text('Row Layout'),

),

body: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly, 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,

),

],

),

),

);

}

}

**Column:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp( home:

Scaffold(appBar: AppBar(

title: Text('Column Layout'),

),

body: Column(

mainAxisAlignment: MainAxisAlignment.spaceEvenly, 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,

),

],

),

),

);

}

}

**Stack:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp( home:

Scaffold(appBar: AppBar(

title: Text('Stack Layout'),

),

body: 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,

),

],

),

),

) ;

}

}

**C) Create a Flutter app with a Text widget showing a counter value, and an ElevatedButton that increments the counter using setState() .**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Counter App',

theme: ThemeData(primarySwatch: Colors.blue),

home: CounterPage(),

);

}

}

class CounterPage extends StatefulWidget {

@override

\_CounterPageState createState() => \_CounterPageState();

}

class \_CounterPageState extends State<CounterPage> {

int \_counter = 0;

void \_incrementCounter() {

setState(() {

\_counter++; // Increase the counter

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Counter Example')),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text('Counter Value:', style: TextStyle(fontSize: 20)),

SizedBox(height: 10),

Text(

'$\_counter',

style: TextStyle(fontSize: 40, fontWeight: FontWeight.bold),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: \_incrementCounter,

child: Text('Increment'),

),

],

),

),

);

}

}

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

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

returnMaterialApp(

title: 'Responsive UI Demo', theme:

ThemeData(primarySwatch:

Colors.blue,

),

home: ResponsiveHomePage(),

);

}

}

classResponsiveHomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Responsive UI Demo'),

),

body: LayoutBuilder(

builder: (BuildContext context, BoxConstraints constraints) {

if (constraints.maxWidth< 600) {

return \_buildNarrowLayout();

} else {

return \_buildWideLayout();

}

},

),

);

}

Widget \_buildNarrowLayout() {

returnCenter(

child: Column(

mainAxisAlignment: MainAxisAlignment.center, children:

<Widget>[

FlutterLogo(size: 100),

SizedBox(height: 20), Text(

'Narrow Layout',

style: TextStyle(fontSize: 24),

),

SizedBox(height: 20),

ElevatedButton(onPressed:

() {},

child: Text('Button'),

),

],

),

);

}

Widget \_buildWideLayout() {

returnCenter(

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

FlutterLogo(size: 100),

SizedBox(width: 20),

Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Wide Layout',

style: TextStyle(fontSize: 24),

),

SizedBox(height: 20),

ElevatedButton(onPressed: ()

{},

child: Text('Button'),

),

],

),

],

),

);

}

}

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

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

classMyApp extends

StatelessWidget {

@override

Widget build(BuildContext

context) {

returnMaterialApp(

title: 'Responsive UI with Media Queries', theme: ThemeData(

primarySwatch: Colors.blue,

),

home:

ResponsiveHomePage(),

);

}

}

classResponsiveHomePage extends

StatelessWidget {

@override

Widget build(BuildContext context) {

return

Scaffold(

appBar: AppBar(

title: Text('Responsive UI with Media Queries'),

),

body: LayoutBuilder(

builder: (BuildContext context, BoxConstraints constraints) {

if (constraints.maxWidth< 600) {

return \_buildMobileLayout();

} else if (constraints.maxWidth< 1200) {

return \_buildTabletLayout();

} else {

return \_buildDesktopLayout();

}

},

),

);

}

Widget \_buildMobileLayout() {

returnCenter(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

FlutterLogo(size: 100),

SizedBox(height: 20),

Text(

'Mobile Layout',

style: TextStyle(fontSize: 24),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {},

child: Text('Button'),

),

],

),

);

}

Widget

\_buildTabletLayout() {

return

Center(

child:

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

FlutterLogo(size: 100),

SizedBox(width: 20),

Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Tablet Layout',

style: TextStyle(fontSize: 24),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {},

child: Text('Button'),

),

],

),

],

),

);

}

Widget \_buildDesktopLayout() {

return

Center(

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

FlutterLogo(size: 100),

SizedBox(width: 20),

Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Desktop Layout',

style: TextStyle(fontSize: 24),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {},

child: Text('Button'),

),

],

),

],

),

);

}

}