

**(Flutter)**

Flutter Lab 7 (Session 2)

**Total Time:**

**3 hours (session-2)**

**Pre-Lab Activities:**

* + Student must complete animation lesson

**Learning Outcomes:**

Demonstrate the knowledge of Flutter animation.

**Student Activities:**

* + Practicing Stateless and Statfull Widgets

**Lab Tasks:**

* + Create a Flutter animation application, which scall up and down the Flutter Logo



**Lab Solutions:**

**CODE:**

|  |
| --- |
| import 'package:flutter/animation.dart';  import 'package:flutter/material.dart';  void main() => runApp(MyApp());    class MyApp extends StatelessWidget {  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  title: 'Flutter Animation',  theme: ThemeData(  // This is the theme of your application.  primarySwatch: Colors.blue,  ),  home: MyHomePage(),  );  }  }  class MyHomePage extends StatefulWidget {  \_HomePageState createState() => \_HomePageState();  }  class \_HomePageState extends State<MyHomePage> with SingleTickerProviderStateMixin {  Animation<double> animation;  AnimationController animationController;  @override  void initState() {  super.initState();  animationController = AnimationController(vsync: this, duration: Duration(milliseconds: 2500));  animation = Tween<double>(begin: 0.0, end: 1.0).animate(animationController);  animation.addListener((){  setState((){  print (animation.value.toString());  });  });  animation.addStatusListener((status){  if(status == AnimationStatus.completed){  animationController.reverse();  } else if(status == AnimationStatus.dismissed) {  animationController.forward();  }  });  animationController.forward();  }  @override  Widget build(BuildContext context) {  return Center(  child: AnimatedLogo(  animation: animation,  )  );  }  }  class AnimatedLogo extends AnimatedWidget {  final Tween<double> \_sizeAnimation = Tween<double> (begin: 0.0, end: 500.0);  AnimatedLogo({Key key, Animation animation}):super(key: key, listenable: animation);  @override  Widget build(BuildContext context) {  final Animation<double> animation = listenable;  return Transform.scale(  scale: \_sizeAnimation.evaluate(animation),  child: FlutterLogo(),  );  }  } |