import 'package:flutter/material.dart';

void main() {

runApp(MaterialApp(home: ImageRotationDemo()));

}

class ImageRotationDemo extends StatefulWidget {

@override

\_ImageRotationDemoState createState() => \_ImageRotationDemoState();

}

class \_ImageRotationDemoState extends State<ImageRotationDemo>

with SingleTickerProviderStateMixin {

late AnimationController \_controller;

late Animation<double> \_rotationAnimation;

bool isClockwise = true;

@override

void initState() {

super.initState();

\_controller = AnimationController(

vsync: this,

duration: Duration(seconds: 1),

);

// Initial rotation direction: clockwise

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

}

void \_rotateImage() {

setState(() {

isClockwise = !isClockwise;

// Update the tween for rotation direction

\_rotationAnimation = Tween<double>(

begin: 0.0,

end: isClockwise ? 1.0 : -1.0,

).animate(\_controller);

// Restart the animation

\_controller

..reset()

..forward();

});

}

@override

void dispose() {

\_controller.dispose(); // Dispose controller to avoid memory leaks

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Rotate Image Clockwise/Anti-clockwise')),

body: Center(

child: RotationTransition(

turns: \_rotationAnimation,

child: Image.asset('assets/clock.png', width: 150, height: 150),

),

),

floatingActionButton: FloatingActionButton(

onPressed: \_rotateImage,

child: Icon(Icons.rotate\_right),

tooltip: isClockwise ? 'Rotate Clockwise' : 'Rotate Anti-Clockwise',

),

);

}

}