



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



Department of Information Technology

Academic Year: 2022-23

Semester: VI

Class / Branch / Div: TE- IT A/B

Subject: MAD & PWA Lab

Name of Instructor:

Name of Student:

Student ID:

Roll No.

Date of Submission:

Experiment No.:6

Aim: Apply concepts of gestures in Flutter App.

Theory:

Gestures are an interesting feature in Flutter that allows us to interact with the mobile app (or any touch-based device). Generally, gestures define any physical action or movement of a user in the intention of specific control of the mobile device. We use all these gestures in everyday life to interact with your phone or touch-based device.

Some of the examples of gestures are:

1. sliding the lock screen.
2. Tap on mobile screen
3. Holding touch on the button.

Gestures

It is the second layer that represents **semantic actions** such as tap, drag, and scale, which are recognized from multiple individual pointer events. It is also able to dispatch multiple events corresponding to gesture lifecycle like drag start, drag update, and drag end.

Some of the widely used gestures are mentioned here –

Tap: It means touching the surface of the screen from the fingertip for a short time and then releasing them. This gesture contains the following events:

- onTapDown
- onTapUp
- onTap
- onTapCancel

Double Tap: It is similar to a Tap gesture, but you need to tapping twice in a short time. This gesture contains the following events:

- onDoubleTap

Drag: It allows us to touch the surface of the screen with a fingertip and move it from one location to another location and then releasing them. Flutter categories the drag into two types:



1. **Horizontal Drag:** This gesture allows the pointer to move in a horizontal direction. It contains the following events:
 - onHorizontalDragStart
 - onHorizontalDragUpdate
 - onHorizontalDragEnd
2. **Vertical Drag:** This gesture allows the pointer to move in a vertical direction. It contains the following events:
 - onVerticalDragStart
 - onVerticalDragStart
 - onVerticalDragStart

Long Press: It means touching the surface of the screen at a particular location for a long time. This gesture contains the following events:

- onLongPress

Pan: It means touching the surface of the screen with a fingertip, which can move in any direction without releasing the fingertip. This gesture contains the following events:

- onPanStart
- onPanUpdate
- onPanEnd

Pinch: It means pinching (move one's finger and thumb or bring them together on a touchscreen) the surface of the screen using two fingers to zoom into or out of a screen.

Gesture Detector

Flutter provides a widget that gives excellent support for all types of gestures by using the **GestureDetector** widget. The **GestureDetector** is a non-visual widget, which is primarily used for detecting the user's gesture. The basic idea of the gesture detector is a **stateless** widget that contains parameters in its constructor for different touch events.

Implementation

We are going to implement UI where on click of a button text, light changes state.

In **BuildContext()** we return **Scaffold()** and use the property of **AppBar()** and pass the title of the application. In the body of **Scaffold()**, we take **Container()** and after that, we pass **Column()** as a child of it. As, we can create multiple children of **Column** widget, **Icon()** and **Container()**. **GestureDetector()** will just detect the action while using the property **onTap()**. This is used when the user makes contact with the screen, which might be a tap. **Text()** is the child of **Container()** which will tell the user whether the **LIGHTS** are **ON** or **OFF**.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



Conclusion: In this experiment, we have successfully designed a flutter application using GestureDetector.