# AIM:

Learning Resources

* <https://docs.flutter.dev/cookbook/plugins/play-video>
* Media Handling in Flutter: <https://api.flutter.dev/flutter/widgets/MediaQuery-class.html>

Practical Task:

* Build a video player app.
* Add controls for play, pause, and seek functionality.

Topics Covered:

* Integrating External Plugins
* Media Query and Responsive Design
* Gesture Detection for Video Controls

# THEORY:

Flutter provides a powerful framework for **media handling**, allowing developers to integrate video playback using external plugins. The **video\_player** plugin is commonly used to embed and control video playback efficiently.

**Key Concepts:**

1. **Integrating External Plugins:**
   * Flutter supports third-party plugins to extend functionality.
   * The **video\_player** package enables video playback from assets, network URLs, or device storage.
2. **Media Query and Responsive Design:**
   * The **MediaQuery** class helps adapt UI elements to different screen sizes, ensuring a responsive layout for video players on mobile and tablet devices.
3. **Gesture Detection for Video Controls:**
   * Flutter provides touch-based interaction via **GestureDetector**, enabling users to play, pause, or seek videos using tap and swipe gestures.

**Implementation Steps:**

1. **Install Dependencies:**
   * Add the **video\_player** package to pubspec.yaml.
   * Import the package in the Dart file.
2. **Set Up Video Playback:**
   * Initialize a VideoPlayerController.
   * Load a video from assets or a network URL.
3. **Implement Controls:**
   * Create buttons for **Play, Pause, and Seek**.
   * Use **GestureDetector** for touch interactions.
4. **Ensure Responsiveness:**
   * Use **MediaQuery** to adjust UI elements dynamically.

# CODE:

import 'package:flutter/material.dart';

import 'package:video\_player/video\_player.dart';

import 'package:flutter/services.dart';

void main() => runApp(const VideoApp());

class VideoApp extends StatefulWidget {

  const VideoApp({super.key});

  @override

  \_VideoAppState createState() => \_VideoAppState();

}

class \_VideoAppState extends State<VideoApp> {

  late VideoPlayerController \_controller;

  bool \_isMuted = false;

  bool \_isFullScreen = false;

  double \_playbackSpeed = 1.0;

  @override

  void initState() {

    super.initState();

    \_controller =

        VideoPlayerController.networkUrl(

            Uri.parse(

              'https://flutter.github.io/assets-for-api-docs/assets/videos/bee.mp4',

            ),

          )

          ..initialize().then((\_) {

            setState(() {});

          })

          ..addListener(() {

            setState(() {});

          });

  }

  void \_toggleFullScreen() {

    setState(() {

      \_isFullScreen = !\_isFullScreen;

      if (\_isFullScreen) {

        SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersive);

        SystemChrome.setPreferredOrientations([

          DeviceOrientation.landscapeLeft,

          DeviceOrientation.landscapeRight,

        ]);

      } else {

        SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);

        SystemChrome.setPreferredOrientations([

          DeviceOrientation.portraitUp,

          DeviceOrientation.portraitDown,

        ]);

      }

    });

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Advanced Video Player',

      home: Scaffold(

        appBar:

            \_isFullScreen ? null : AppBar(title: const Text('Video Player')),

        body: OrientationBuilder(

          builder: (context, orientation) {

            return Stack(

              children: [

                Center(

                  child:

                      \_controller.value.isInitialized

                          ? Column(

                            mainAxisAlignment: MainAxisAlignment.center,

                            children: [

                              Expanded(

                                child: AspectRatio(

                                  aspectRatio: \_controller.value.aspectRatio,

                                  child: Stack(

                                    alignment: Alignment.bottomCenter,

                                    children: [

                                      VideoPlayer(\_controller),

                                      VideoProgressIndicator(

                                        \_controller,

                                        allowScrubbing: true,

                                      ),

                                    ],

                                  ),

                                ),

                              ),

                              if (!\_isFullScreen)

                                AnimatedOpacity(

                                  opacity:

                                      \_controller.value.isPlaying ? 0.0 : 1.0,

                                  duration: const Duration(milliseconds: 300),

                                  child: \_buildControls(),

                                ),

                            ],

                          )

                          : const CircularProgressIndicator(),

                ),

                if (\_isFullScreen)

                  Positioned(

                    top: 20,

                    left: 20,

                    child: GestureDetector(

                      onTap: \_toggleFullScreen,

                      child: Container(

                        padding: const EdgeInsets.all(10),

                        decoration: BoxDecoration(

                          color: Colors.black54,

                          borderRadius: BorderRadius.circular(10),

                        ),

                        child: const Icon(

                          Icons.close,

                          color: Colors.white,

                          size: 30,

                        ),

                      ),

                    ),

                  ),

              ],

            );

          },

        ),

      ),

    );

  }

  Widget \_buildControls() {

    return Container(

      padding: const EdgeInsets.all(10),

      decoration: BoxDecoration(

        color: Colors.black54,

        borderRadius: BorderRadius.circular(10),

      ),

      child: Row(

        mainAxisAlignment: MainAxisAlignment.center,

        children: [

          IconButton(

            icon: const Icon(Icons.replay\_10, color: Colors.white),

            onPressed:

                () => \_controller.seekTo(

                  \_controller.value.position - const Duration(seconds: 10),

                ),

          ),

          IconButton(

            icon: Icon(

              \_controller.value.isPlaying ? Icons.pause : Icons.play\_arrow,

              color: Colors.white,

            ),

            onPressed: () {

              setState(() {

                \_controller.value.isPlaying

                    ? \_controller.pause()

                    : \_controller.play();

              });

            },

          ),

          IconButton(

            icon: const Icon(Icons.forward\_10, color: Colors.white),

            onPressed:

                () => \_controller.seekTo(

                  \_controller.value.position + const Duration(seconds: 10),

                ),

          ),

          IconButton(

            icon: Icon(

              \_isMuted ? Icons.volume\_off : Icons.volume\_up,

              color: Colors.white,

            ),

            onPressed: () {

              setState(() {

                \_isMuted = !\_isMuted;

                \_controller.setVolume(\_isMuted ? 0.0 : 1.0);

              });

            },

          ),

          DropdownButton<double>(

            value: \_playbackSpeed,

            dropdownColor: Colors.black,

            style: const TextStyle(color: Colors.white),

            items:

                [0.5, 1.0, 1.5, 2.0]

                    .map(

                      (speed) => DropdownMenuItem(

                        value: speed,

                        child: Text('${speed}x'),

                      ),

                    )

                    .toList(),

            onChanged: (speed) {

              if (speed != null) {

                setState(() {

                  \_playbackSpeed = speed;

                  \_controller.setPlaybackSpeed(speed);

                });

              }

            },

          ),

          IconButton(

            icon: Icon(

              \_isFullScreen ? Icons.fullscreen\_exit : Icons.fullscreen,

              color: Colors.white,

            ),

            onPressed: \_toggleFullScreen,

          ),

        ],

      ),

    );

  }

  @override

  void dispose() {

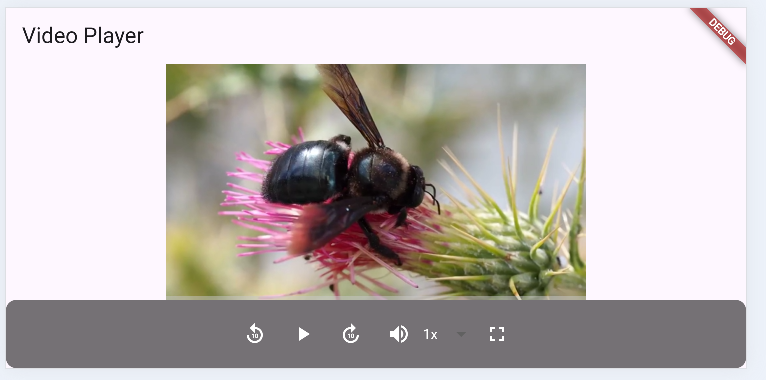
    \_controller.dispose();

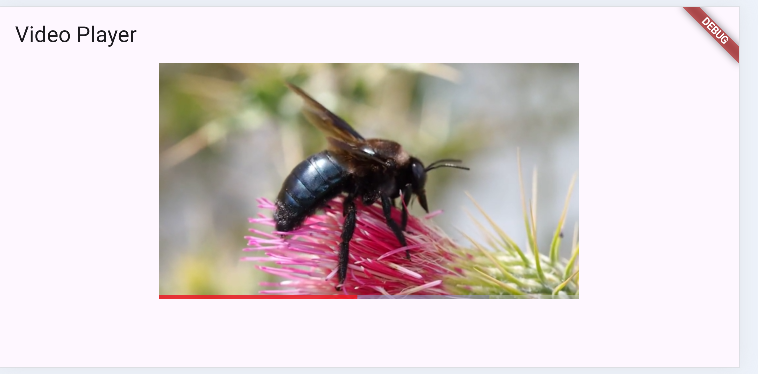
    super.dispose();

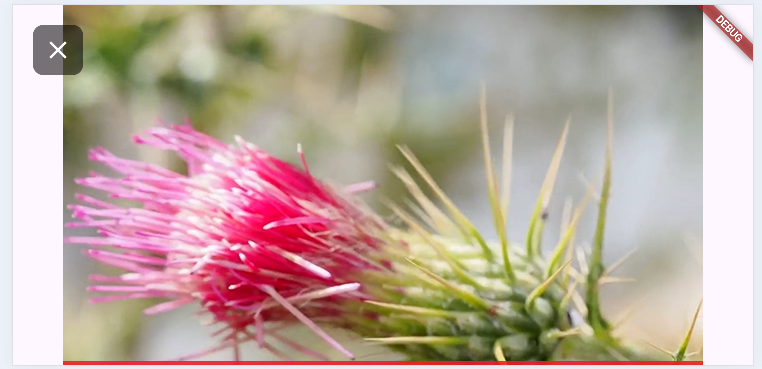
  }

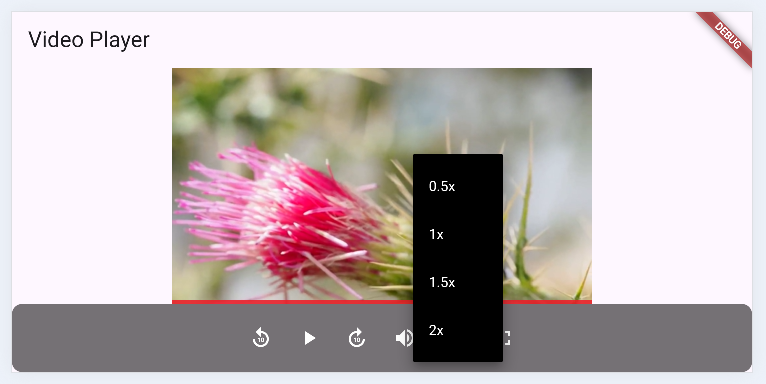
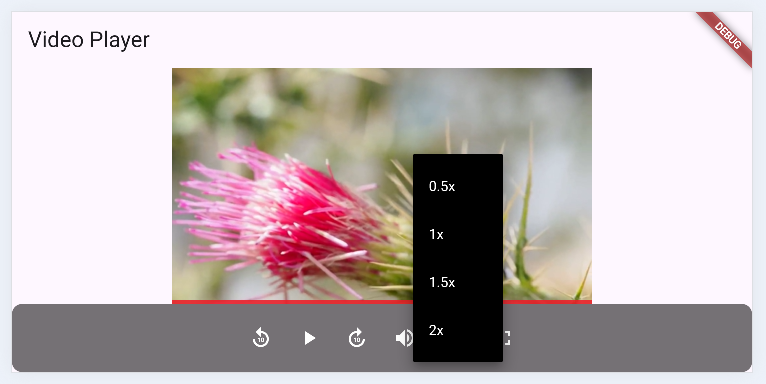
}

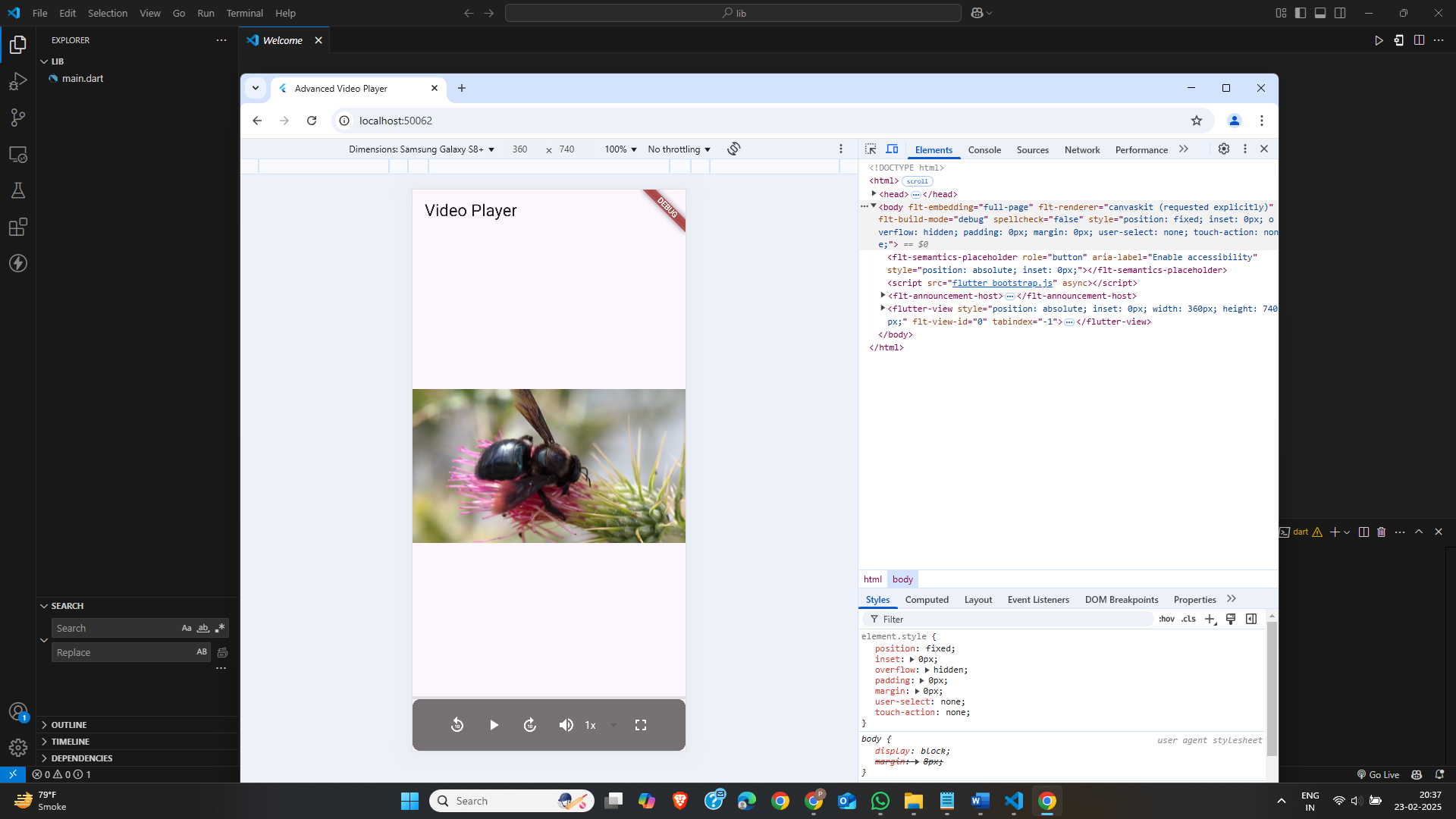
**OUTPUT:**

****

****

****

**** ****



# Latest Applications:

✅ **Streaming Apps** – Used in apps like YouTube, Netflix, and Twitch.  
✅ **Educational Platforms** – Online learning platforms embed videos for courses.  
✅ **Social Media** – Platforms like Instagram and TikTok use embedded videos.  
✅ **Security Systems** – Surveillance apps provide live and recorded video playback.

**Learning Outcome:**

**🔹** Understand how to integrate external plugins in Flutter.  
🔹 Learn to handle video playback with play, pause, and seek functionality.  
🔹 Apply MediaQuery for responsive UI design.  
🔹 Implement gesture detection for interactive video controls.