## AIM:

**Task Description:**

Develop two distinct applications using Flutter:

1. News App: Utilize multiple widgets and layouts to create a user-friendly interface for

a news application. The app should allow users to browse news articles with features

such as categories, headlines, and article previews. The design and functionality

should aim to resemble the example output provided.

2. OTT App: Create an Over-The-Top (OTT) media streaming application. The app

should include widgets and layouts to display movies, shows, and recommendations

with an intuitive design that enhances user experience.

## THEORY:

**1. StatefulWidget**

* **Definition:**  
  A StatefulWidget is a widget that maintains state that might change during its lifetime. It consists of two classes:
  + The StatefulWidget class itself, which is immutable.
  + A State class where the mutable state is maintained.
* **Use Case:**  
  Use StatefulWidget when the UI depends on dynamic data that can change, such as user interactions or real-time updates.
* **Example:**

class MyStatefulWidget extends StatefulWidget {

@override

\_MyStatefulWidgetState createState() => \_MyStatefulWidgetState();

}

class \_MyStatefulWidgetState extends State<MyStatefulWidget> {

int counter = 0;

void incrementCounter() {

setState(() {

counter++;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

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

body: Center(child: Text('Counter: $counter')),

floatingActionButton: FloatingActionButton(

onPressed: incrementCounter,

child: Icon(Icons.add),

),

);

}

}

**2. Factory Constructor**

* **Definition:**  
  A factory constructor in Dart allows you to control instance creation. It returns an instance of the class or an existing instance (singleton).
* **Use Case:**  
  Useful for implementing patterns like singletons, caching, or object pooling.
* **Example:**

class Singleton {

static final Singleton \_instance = Singleton.\_internal();

factory Singleton() {

return \_instance;

}

Singleton.\_internal();

}

void main() {

var s1 = Singleton();

var s2 = Singleton();

print(s1 == s2); // true

}

**3. HTTP GET Request**

* **Definition:**  
  Flutter uses the http package to perform network requests like GET, POST, PUT, DELETE, etc.
* **Use Case:**  
  Use GET requests to fetch data from a REST API.
* **Example:**

import 'package:http/http.dart' as http;

import 'dart:convert';

Future<void> fetchData() async {

final response = await http.get(Uri.parse('https://api.example.com/data'));

if (response.statusCode == 200) {

var data = jsonDecode(response.body);

print(data);

} else {

print('Failed to load data');

}

}

**4. SingleChildScrollView**

* **Definition:**  
  A SingleChildScrollView allows a single widget to be scrollable when its content overflows the available screen space.
* **Use Case:**  
  Use this widget for a scrollable layout with a single child.
* **Example:**

@override

Widget build(BuildContext context) {

return Scaffold(

body: SingleChildScrollView(

child: Column(

children: List.generate(50, (index) => Text('Item $index')),

),

),

);

}

**5. Route and Routing**

* **Definition:**  
  Routing in Flutter involves navigation between different screens (widgets). Flutter provides Navigator and MaterialPageRoute for navigation.
* **Use Case:**  
  Use routing for app navigation and managing the back stack.
* **Example:**

Navigator.push(

context,

MaterialPageRoute(builder: (context) => SecondScreen()),

);

**6. YouTube Controller (Video Playback)**

* **Definition:**  
  The youtube\_player\_flutter package provides a widget and controller to embed YouTube videos in your Flutter app.
* **Use Case:**  
  Embed and control YouTube videos.
* **Example:**

import 'package:youtube\_player\_flutter/youtube\_player\_flutter.dart';

class YouTubeExample extends StatefulWidget {

@override

\_YouTubeExampleState createState() => \_YouTubeExampleState();

}

class \_YouTubeExampleState extends State<YouTubeExample> {

late YoutubePlayerController \_controller;

@override

void initState() {

super.initState();

\_controller = YoutubePlayerController(

initialVideoId: 'dQw4w9WgXcQ',

flags: YoutubePlayerFlags(autoPlay: true),

);

}

@override

Widget build(BuildContext context) {

return YoutubePlayer(

controller: \_controller,

showVideoProgressIndicator: true,

);

}

}

**7. ListView**

* **Definition:**  
  A ListView is a scrollable list of widgets. It is highly customizable and efficient for rendering large datasets.
* **Use Case:**  
  Use ListView to display a scrollable list of items.
* **Example:**

@override

Widget build(BuildContext context) {

return Scaffold(

body: ListView(

children: List.generate(

10,

(index) => ListTile(

title: Text('Item $index'),

),

),

),

);

}

**8. Other Similar Widgets**

* **GridView:**  
  Used for displaying a scrollable grid of widgets.

GridView.count(

crossAxisCount: 2,

children: List.generate(10, (index) => Text('Item $index')),

);

* **FutureBuilder:**  
  A widget that builds itself based on the latest snapshot of interaction with a Future.

FutureBuilder(

future: fetchData(),

builder: (context, snapshot) {

if (snapshot.connectionState == ConnectionState.done) {

return Text('Data: ${snapshot.data}');

} else {

return CircularProgressIndicator();

}

},

);

* **StreamBuilder:**  
  Similar to FutureBuilder, but for Stream.

StreamBuilder(

stream: myStream,

builder: (context, snapshot) {

return Text('Data: ${snapshot.data}');

},

);

* **CustomScrollView:**  
  Combines multiple scrolling views (e.g., ListView, GridView) into a single scrollable layout.

CustomScrollView(

slivers: [

SliverList(

delegate: SliverChildBuilderDelegate(

(context, index) => ListTile(title: Text('Item $index')),

),

),

],

);

**9. Additional Classes/Concepts**

* **InheritedWidget:** For sharing state efficiently across widgets.
* **PageView:** For implementing swipable pages.
* **Drawer:** For creating side navigation menus.
* **Provider:** A state management solution in Flutter.

## CODE:

## News app :

**Main.dart**

import 'dart:ui';

import 'package:flutter/material.dart';

import 'package:http/http.dart' as http;

import 'dart:convert';

import './news.dart';

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

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'News List App',

      theme: ThemeData(

        primarySwatch: Colors.indigo,

        scaffoldBackgroundColor: const Color(0xFFF3F4F6), // Light background

        appBarTheme: const AppBarTheme(

          color: Color(0xFF1E40AF), // Darker Indigo

        ),

        textTheme: const TextTheme(

          bodyMedium: TextStyle(color: Colors.black87, fontSize: 16),

          headlineMedium: TextStyle(

              color: Colors.black87, fontSize: 24, fontWeight: FontWeight.bold),

        ),

      ),

      home: const NewsListPage(),

    );

  }

}

class NewsListPage extends StatefulWidget {

  const NewsListPage({super.key});

  @override

  \_NewsListPageState createState() => \_NewsListPageState();

}

class \_NewsListPageState extends State<NewsListPage> {

  late Future<List<News>> \_newsFuture;

  TextEditingController \_searchController = TextEditingController();

  @override

  void initState() {

    super.initState();

    \_newsFuture = fetchNewsList();

  }

  Future<List<News>> fetchNewsList({String query = "tesla"}) async {

    final uri = Uri.parse(

        "https://newsapi.org/v2/everything?q=$query&from=2024-12-05&sortBy=publishedAt&apiKey=2764460cb029438fb103a3270d3f67fb");

    final response = await http.get(uri);

    if (response.statusCode == 200) {

      final jsonData = json.decode(response.body);

      final newsList =

          jsonData['articles'] as List; // Assuming articles is a list

      return newsList.map((newsJson) => News.fromJson(newsJson)).toList();

    } else {

      throw Exception('Failed to load news');

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text('News Articles'),

        actions: [

          Padding(

            padding: const EdgeInsets.all(8.0),

            child: IconButton(

              icon: const Icon(Icons.search, color: Colors.white),

              onPressed: () {

                setState(() {

                  \_newsFuture = fetchNewsList(query: \_searchController.text);

                });

              },

            ),

          ),

        ],

      ),

      body: Column(

        children: [

          Padding(

            padding: const EdgeInsets.all(8.0),

            child: TextField(

              controller: \_searchController,

              decoration: const InputDecoration(

                hintText: "Search for news...",

                border: OutlineInputBorder(),

                prefixIcon: Icon(Icons.search),

              ),

            ),

          ),

          Expanded(

            child: FutureBuilder<List<News>>(

              future: \_newsFuture,

              builder: (context, snapshot) {

                if (snapshot.connectionState == ConnectionState.waiting) {

                  return const Center(child: CircularProgressIndicator());

                } else if (snapshot.hasError) {

                  return Center(child: Text('Error: ${snapshot.error}'));

                } else if (snapshot.hasData) {

                  final newsList = snapshot.data!;

                  return SingleChildScrollView(

                    scrollDirection: Axis.horizontal,

                    child: Row(

                      children: newsList.map((news) {

                        return Padding(

                          padding: const EdgeInsets.all(8.0),

                          child: Container(

                            width: 300, // Fixed width for each news item

                            decoration: BoxDecoration(

                              color: Colors.indigo[300], // Light Indigo

                              borderRadius: BorderRadius.circular(12),

                              boxShadow: [

                                BoxShadow(

                                  color: Colors.indigo[200]!.withOpacity(0.5),

                                  spreadRadius: 1,

                                  blurRadius: 5,

                                  offset: const Offset(0, 3),

                                ),

                              ],

                            ),

                            child: InkWell(

                              onTap: () {

                                Navigator.push(

                                  context,

                                  MaterialPageRoute(

                                    builder: (context) =>

                                        NewsDetailsPage(news: news),

                                  ),

                                );

                              },

                              child: Column(

                                mainAxisAlignment: MainAxisAlignment.start,

                                children: [

                                  Padding(

                                    padding: const EdgeInsets.all(8.0),

                                    child: Text(

                                      news.title,

                                      style: const TextStyle(

                                        color: Colors.white,

                                        fontSize: 18,

                                        fontWeight: FontWeight.bold,

                                      ),

                                      maxLines: 2,

                                      overflow: TextOverflow.ellipsis,

                                    ),

                                  ),

                                  news.imageUrl.isNotEmpty

                                      ? ClipRRect(

                                          borderRadius:

                                              BorderRadius.circular(10.0),

                                          child: Image.network(

                                            news.imageUrl,

                                            width: 200,

                                            height: 300,

                                            fit: BoxFit.cover,

                                          ),

                                        )

                                      : const Icon(Icons.image,

                                          size: 200, color: Colors.white),

                                  Padding(

                                    padding: const EdgeInsets.all(8.0),

                                    child: Text(

                                      news.description,

                                      style: const TextStyle(

                                        color: Colors.white,

                                        fontSize: 14,

                                      ),

                                      maxLines: 10,

                                      overflow: TextOverflow.ellipsis,

                                    ),

                                  ),

                                ],

                              ),

                            ),

                          ),

                        );

                      }).toList(),

                    ),

                  );

                } else {

                  return const Center(child: Text('No news available'));

                }

              },

            ),

          ),

        ],

      ),

    );

  }

}

class NewsDetailsPage extends StatelessWidget {

  final News news;

  const NewsDetailsPage({super.key, required this.news});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text(

          news.title,

          maxLines: 1, // Truncating title in AppBar if too long

          overflow: TextOverflow.ellipsis,

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

        ),

        backgroundColor: const Color(0xFF1E3A8A), // Dark Blue

      ),

      body: SingleChildScrollView(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.start,

          children: [

            if (news.imageUrl.isNotEmpty)

              ClipRRect(

                borderRadius: BorderRadius.circular(8.0),

                child: Image.network(

                  news.imageUrl,

                  width: double.infinity,

                  height: 200,

                  fit: BoxFit.cover,

                ),

              )

            else

              const Icon(Icons.image, size: 200, color: Colors.grey),

            const SizedBox(height: 16.0),

            // Title

            Text(

              news.title,

              style: const TextStyle(

                fontSize: 24,

                fontWeight: FontWeight.bold,

                color: Colors.black87,

              ),

            ),

            const SizedBox(height: 8.0),

            // Description

            Text(

              news.description,

              style: const TextStyle(

                fontSize: 16,

                color: Colors.black54,

              ),

            ),

            const SizedBox(height: 16.0),

            // Full Content

            Text(

              news.content,

              style: const TextStyle(

                fontSize: 14,

                color: Colors.black87,

              ),

            ),

          ],

        ),

      ),

    );

  }

}

**News.dart**

class News {

  final String title;

  final String description;

  final String content;

  final String imageUrl;

  final String sourceUrl;

  // Constructor

  News({

    required this.title,

    required this.description,

    required this.content,

    required this.imageUrl,

    required this.sourceUrl,

  });

  // Factory constructor to create a News object from JSON with null checks

  factory News.fromJson(Map<String, dynamic> json) {

    return News(

      title: json['title'] ?? 'No title available',

      description: json['description'] ?? "No description available",

      content: json['content'] ?? 'No content available',

      imageUrl: json['urlToImage'] ?? '', // Empty string if no image URL

      sourceUrl: json['url'] ?? '', // Empty string if no source URL

    );

  }

  // Method to convert a News object to JSON

  Map<String, dynamic> toJson() => {

        'title': title,

        'description': description,

        'content': content,

        'image\_url': imageUrl,

        'source\_url': sourceUrl,

      };

}

## Ott app :

**Main.dart:**

import 'package:flutter/material.dart';

import 'movie\_details.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'OTT App',

      theme: ThemeData(primarySwatch: Colors.blue),

      home: MovieListScreen(),

    );

  }

}

final List<Map<String, dynamic>> movies = [

  {

    'id': 1,

    'title': 'Inception',

    'description':

        'A thief with the ability to enter dreams takes on a final job.',

    'rating': 8.8,

    'image':

        'https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQovCe0H45fWwAtV31ajOdXRPTxSsMQgPIQ3lcZX\_mAW0jXV3kH',

    'video': 'https://www.youtube.com/watch?v=YoHD9XEInc0',

  },

  {

    'id': 2,

    'title': 'The Matrix',

    'description': 'A computer hacker learns about the true nature of reality.',

    'rating': 8.7,

    'image':

        'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR5DoFtShSmClflZ0RzBj9JBMweU5IUVBCeEbbLeV2XPlCnTKNi',

    'video': 'https://www.youtube.com/watch?v=vKQi3bBA1y8',

  },

  {

    'id': 3,

    'title': 'Interstellar',

    'description': 'A journey beyond the stars to save humanity.',

    'rating': 8.6,

    'image':

        'https://upload.wikimedia.org/wikipedia/en/b/bc/Interstellar\_film\_poster.jpg',

    'video': 'https://www.youtube.com/watch?v=zSWdZVtXT7E',

  },

];

class MovieListScreen extends StatefulWidget {

  @override

  \_MovieListScreenState createState() => \_MovieListScreenState();

}

class \_MovieListScreenState extends State<MovieListScreen> {

  List<Map<String, dynamic>> filteredMovies = movies;

  void filterMovies(String query) {

    setState(() {

      filteredMovies = movies

          .where((movie) =>

              movie['title'].toLowerCase().contains(query.toLowerCase()))

          .toList();

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text('Netflix Clone'),

        backgroundColor: Colors.red,

      ),

      body: Column(

        children: [

          Padding(

            padding: const EdgeInsets.all(8.0),

            child: TextField(

              decoration: InputDecoration(

                labelText: 'Search',

                prefixIcon: Icon(Icons.search),

                border: OutlineInputBorder(),

              ),

              onChanged: filterMovies,

            ),

          ),

          Expanded(

            child: GridView.builder(

              gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

                crossAxisCount: 2,

                crossAxisSpacing: 8,

                mainAxisSpacing: 8,

              ),

              itemCount: filteredMovies.length,

              itemBuilder: (context, index) {

                final movie = filteredMovies[index];

                return GestureDetector(

                  onTap: () {

                    Navigator.push(

                      context,

                      MaterialPageRoute(

                        builder: (context) => MovieDetailsScreen(movie: movie),

                      ),

                    );

                  },

                  child: Column(

                    crossAxisAlignment: CrossAxisAlignment.start,

                    children: [

                      Expanded(

                        child: Image.network(

                          movie['image'],

                          fit: BoxFit.cover,

                          width: double.infinity,

                        ),

                      ),

                      Padding(

                        padding: const EdgeInsets.only(top: 8.0),

                        child: Text(

                          movie['title'],

                          style: TextStyle(fontWeight: FontWeight.bold),

                          overflow: TextOverflow.ellipsis,

                        ),

                      ),

                    ],

                  ),

                );

              },

            ),

          ),

        ],

      ),

    );

  }

}

**movie\_details.dart**

import 'package:flutter/material.dart';

import 'package:youtube\_player\_flutter/youtube\_player\_flutter.dart';

class MovieDetailsScreen extends StatefulWidget {

  final Map<String, dynamic> movie;

  MovieDetailsScreen({required this.movie});

  @override

  \_MovieDetailsScreenState createState() => \_MovieDetailsScreenState();

}

class \_MovieDetailsScreenState extends State<MovieDetailsScreen> {

  late YoutubePlayerController \_controller;

  @override

  void initState() {

    super.initState();

    \_controller = YoutubePlayerController(

      initialVideoId: YoutubePlayer.convertUrlToId(widget.movie['video'])!,

      flags: YoutubePlayerFlags(

        autoPlay: false,

        mute: false,

      ),

    );

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text(widget.movie['title']),

        backgroundColor: Colors.red,

      ),

      body: SingleChildScrollView(

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.start,

          children: [

            YoutubePlayer(controller: \_controller),

            Padding(

              padding: const EdgeInsets.all(16.0),

              child: Column(

                crossAxisAlignment: CrossAxisAlignment.start,

                children: [

                  Text(

                    widget.movie['title'],

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

                  ),

                  SizedBox(height: 8),

                  Text(

                    widget.movie['description'],

                    style: TextStyle(fontSize: 16),

                  ),

                  SizedBox(height: 16),

                  Text(

                    'Rating: ${widget.movie['rating']}',

                    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),

                  ),

                ],

              ),

            ),

          ],

        ),

      ),

    );

  }

  @override

  void dispose() {

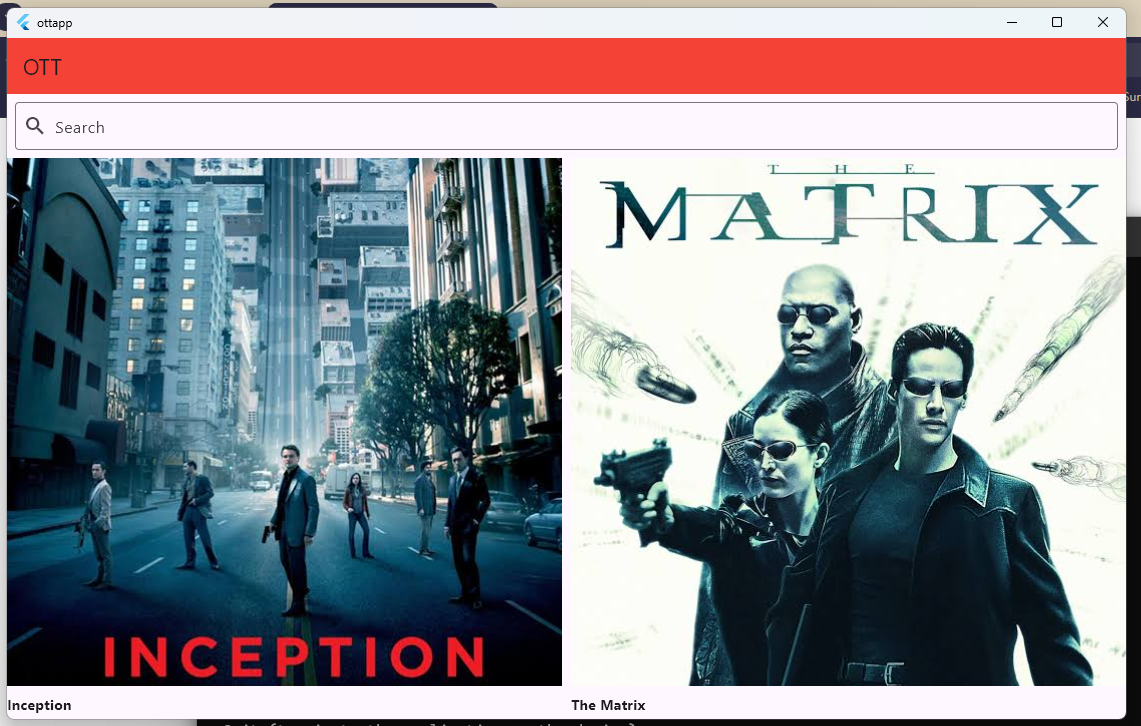
    \_controller.dispose();

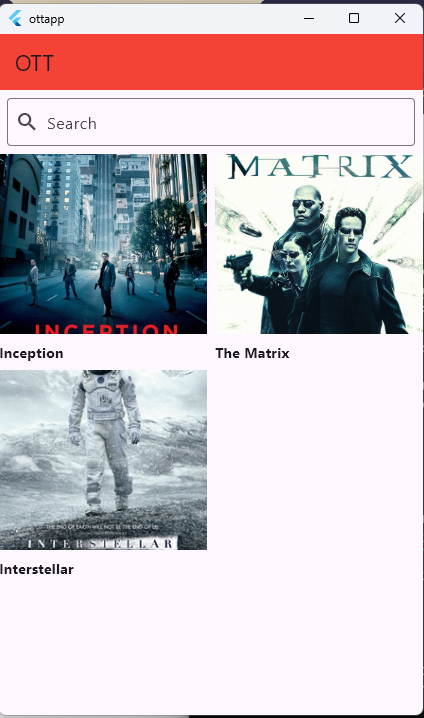
    super.dispose();

  }

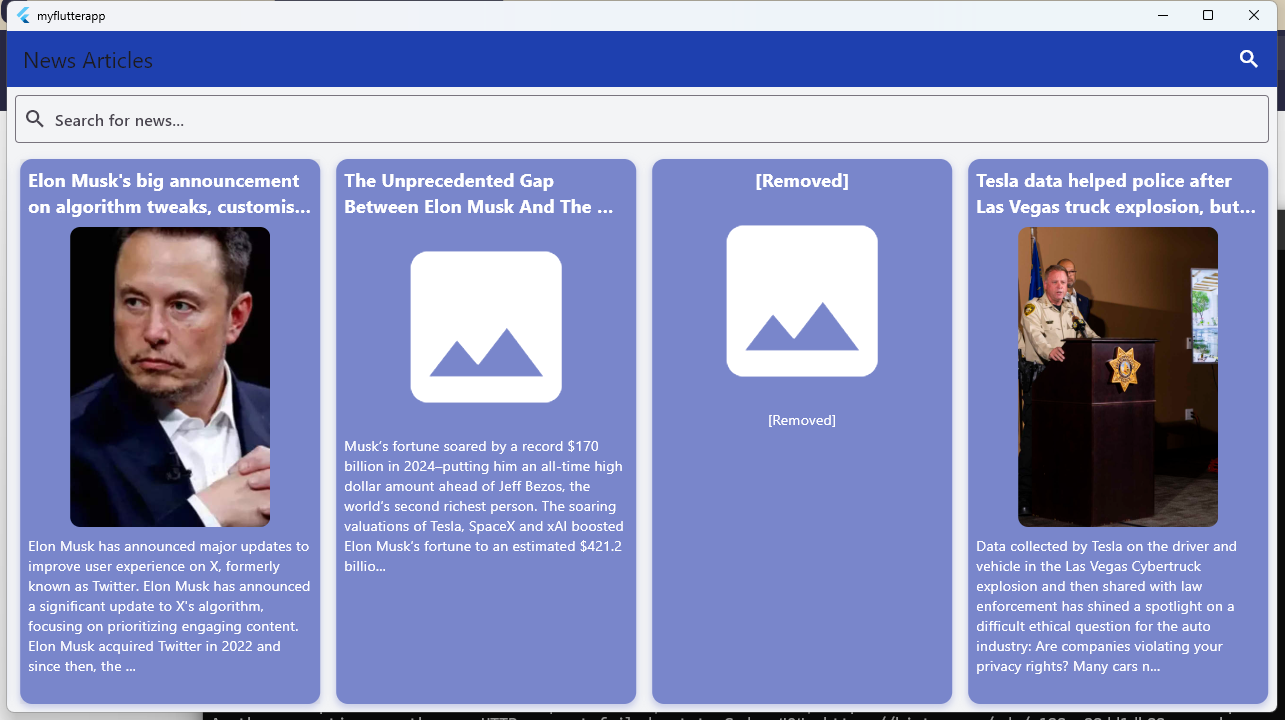
}

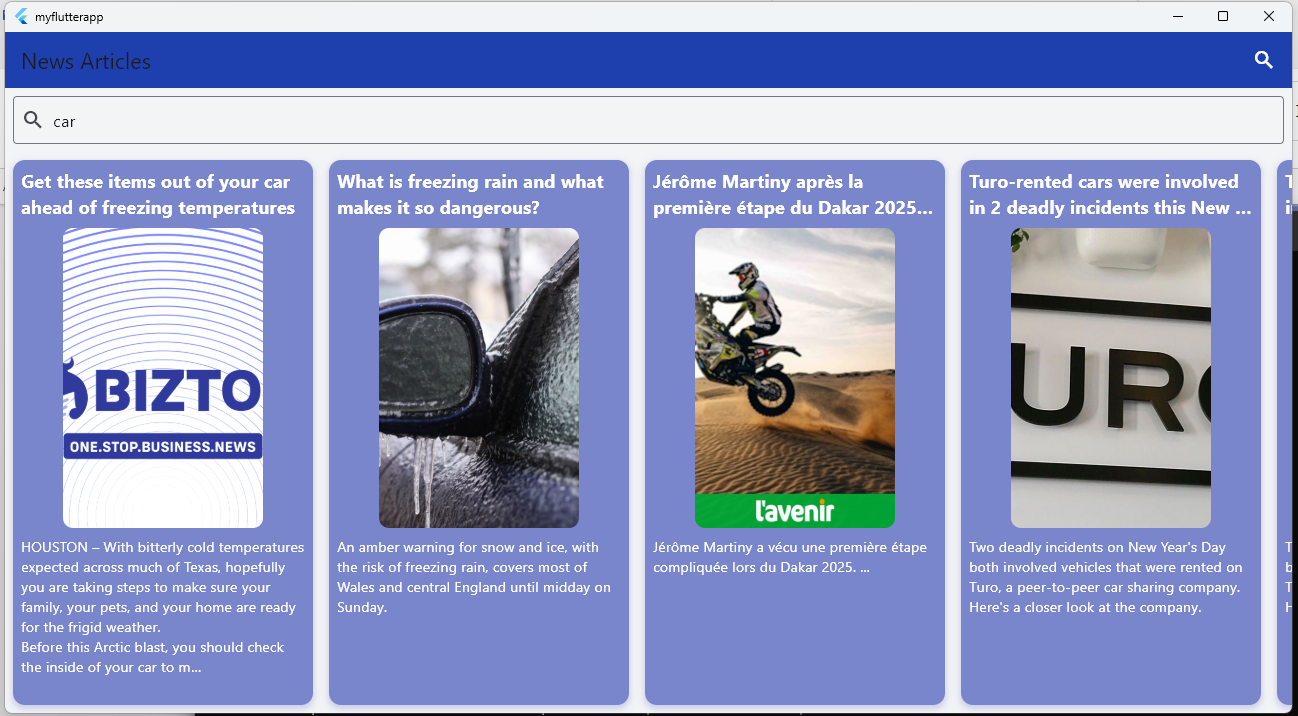
## OUTPUT:





## 





## Learning Outcomes

## State Management:

## Understanding StatefulWidget for creating dynamic and interactive UI components.

## Using factory constructors to create reusable and efficient widgets.

## Networking:

## Performing HTTP GET requests to interact with REST APIs and manage external data.

## Scrollable Widgets:

## Implementing SingleChildScrollView for single scrollable views.

## Using ListView for creating efficient, scrollable lists.

## Routing and Navigation:

## Creating routes using Navigator and MaterialPageRoute.

## Managing navigation stack efficiently in multi-page applications.

## Multimedia Integration:

## Embedding and controlling YouTube videos using controllers.

## Custom UI Creation:

## Using CustomScrollView for building complex, scrollable layouts.

## Implementing GridView for grid-based layouts.

## Future and Stream Management:

## Using FutureBuilder and StreamBuilder for asynchronous data handling.

## Latest Flutter Applications:

## Logistics apps with dynamic routing and inventory management.

## Educational apps with interactive scrollable lists and multimedia embedding.

## Social media apps with video playback features.

## E-commerce apps with grid-based product layouts.

## Latest Applications

## 1. E-Commerce Application

## Displaying products using GridView and ListView.

## Fetching product details via HTTP GET requests.

## Interactive navigation between product categories using routes.

## 2. Video Streaming App

## Embedding YouTube videos using youtube\_player\_flutter.

## Displaying video lists dynamically using ListView.

## 3. Social Networking Platform

## Integrating scrollable posts feed (ListView or CustomScrollView).

## Implementing user-generated video content using YouTube controllers.

## 4. Dashboard or Management App

## Using ListView or GridView for displaying user tasks or inventory.

## Fetching data and updates dynamically from a backend.