MPL Experiment 3

Name: Mohit Patil **Class:** D15A **Roll no:**36

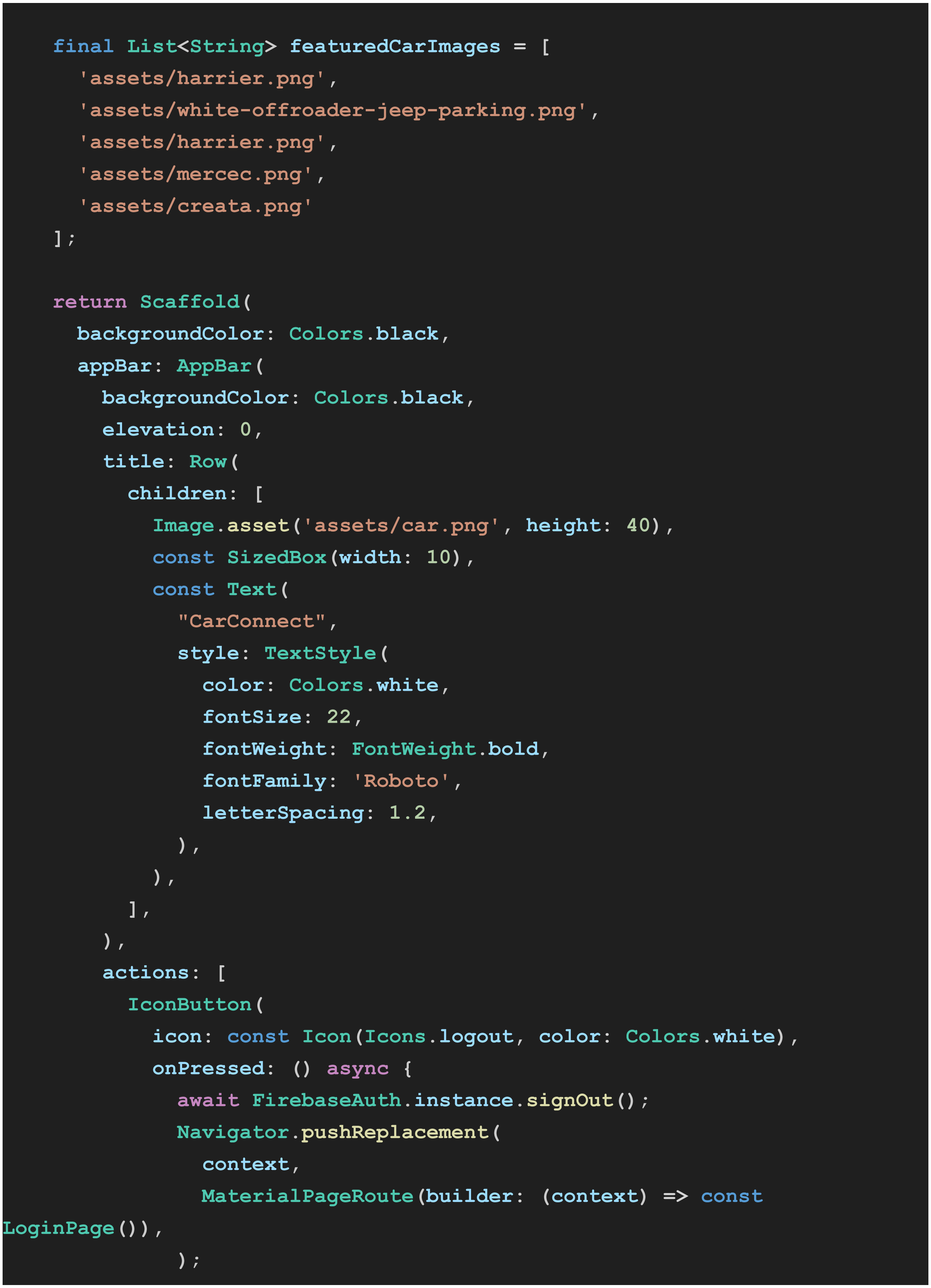
Aim: To include icons, images, fonts in Flutter app

Theory:

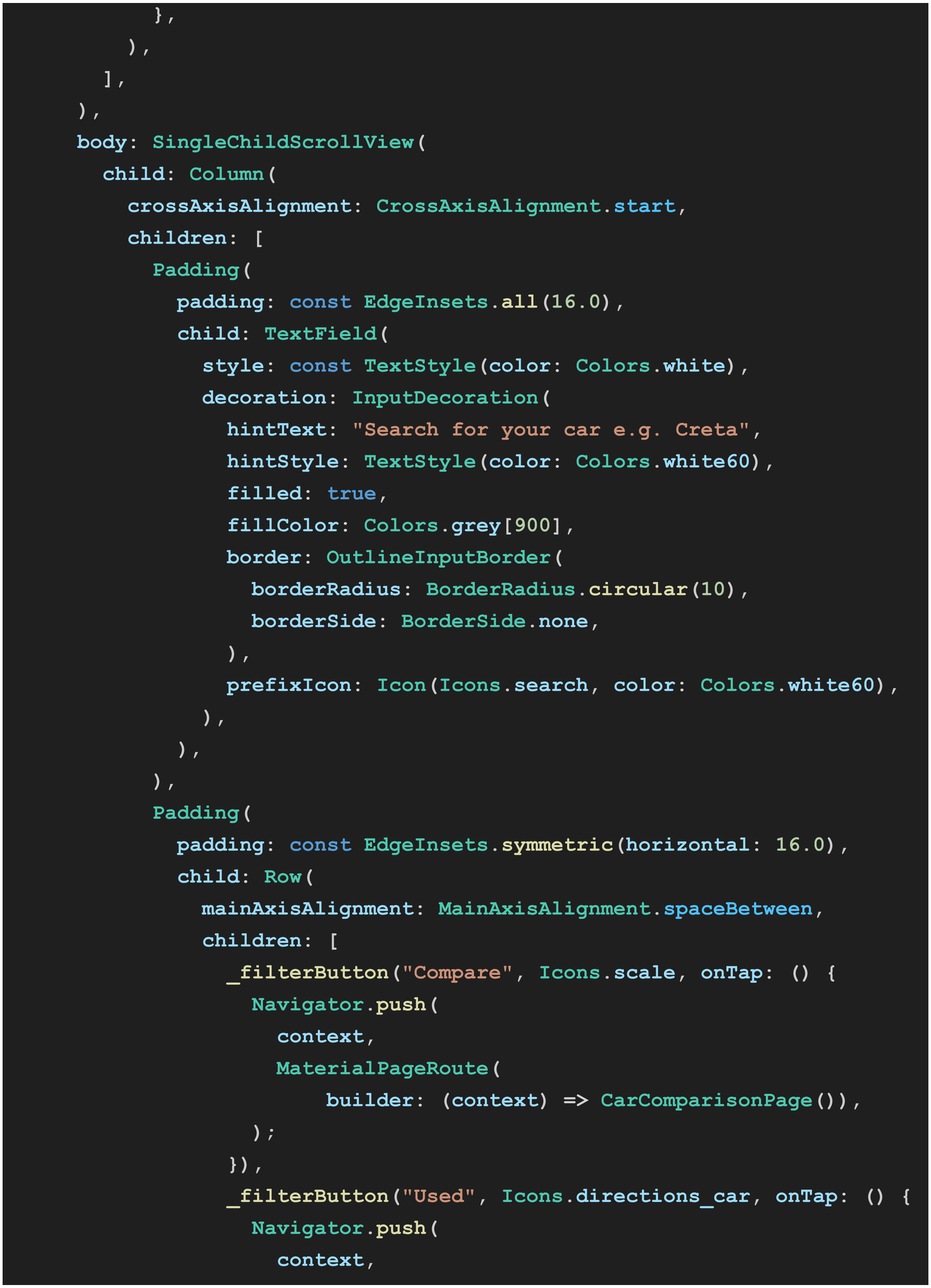
Including icons, images, and custom fonts in a Flutter app enhances visual appeal and user experience. Icons can be added using the Icon widget with built-in Icons class or custom icons via the flutter_launcher_icons package.
Images can be loaded from assets using the Image.asset() method or from the internet using Image.network().
To use local images, they must be placed in the assets folder and declared in pubspec.yaml.
Custom fonts improve typography and branding; they can be added by placing font files in the assets/fonts directory and specifying them in pubspec.yaml under the flutter section.
Using TextStyle with the fontFamily property applies the custom font to text.

Code:
Home :

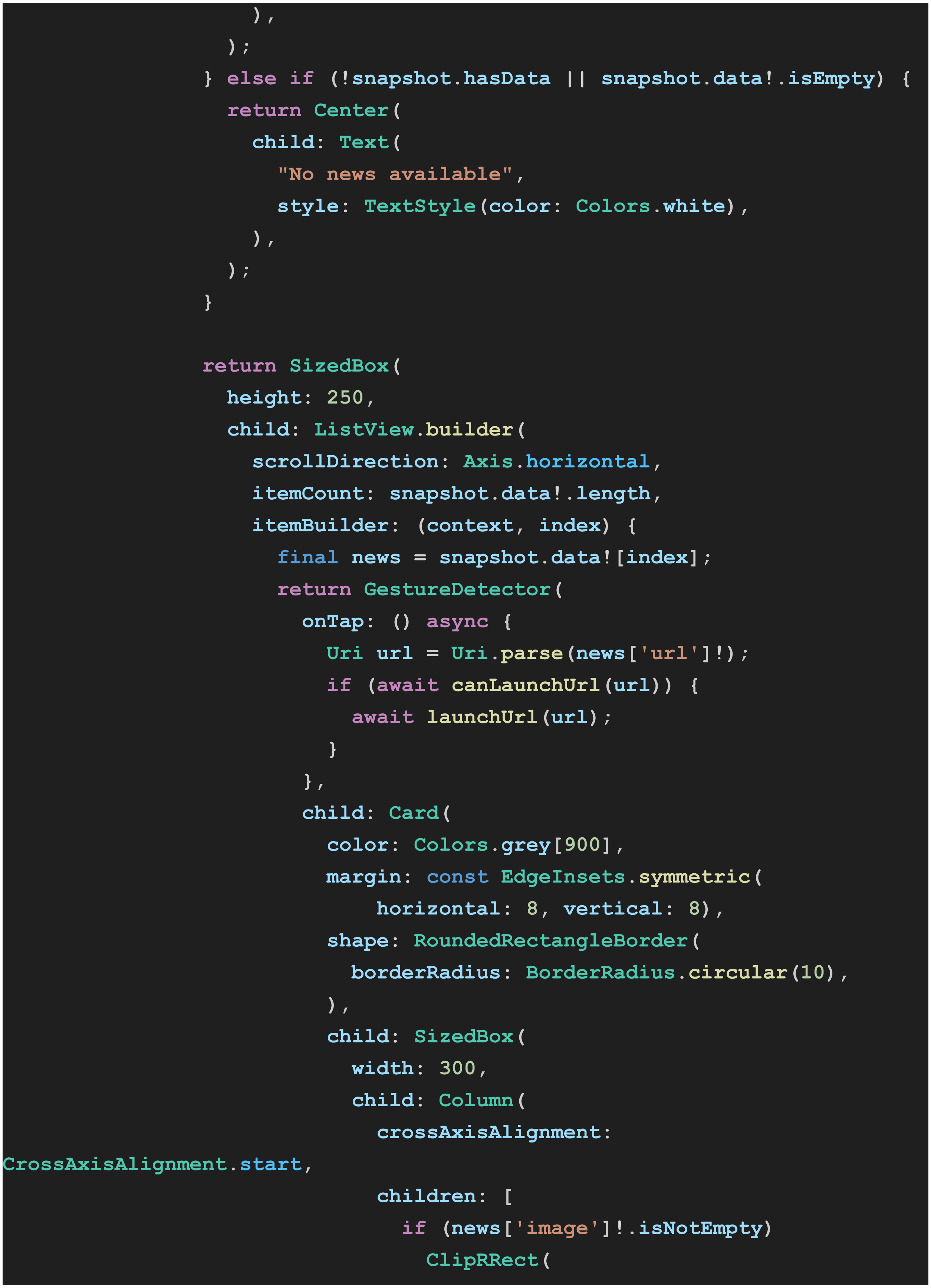
```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:http/http.dart' as http;
import 'package:url_launcher/url_launcher.dart';
import 'login.dart';
import 'UsedCarScreen.dart';
import 'package:carousel_slider/carousel_slider.dart';
import 'carcompare.dart'; // Import CarCompare Page
import 'BudgetPage.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  Future<List<Map<String, String>>> fetchCarNews() async {
    final url =
        'https://newsapi.org/v2/everything?q="car industry" OR
"automobile" OR "new car
launch"&language=en&sortBy=publishedAt&apiKey=4c05b7fc955e4736bd6568828329
9b74';

    final response = await http.get(Uri.parse(url));

    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      List articles = data['articles'];

      return articles.map((article) {
        return {
          'title': (article['title'] ?? '').toString(),
          'image': (article['urlToImage'] ?? '').toString(),
          'url': (article['url'] ?? '').toString(),
        };
      }).toList();
    } else {
      throw Exception('Failed to load news');
    }
  }

  @override
  Widget build(BuildContext context) {
    final User? user = FirebaseAuth.instance.currentUser;

    if (user == null) {
      Future.delayed(Duration.zero, () {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => const LoginPage()),
        );
      });
      return const Scaffold(
        backgroundColor: Colors.black,
        body: Center(child: CircularProgressIndicator(color:
Colors.white)),
      );
    }
```



```
    final List<String> featuredCarImages = [
      'assets/harrier.png',
      'assets/white-offroader-jeep-parking.png',
      'assets/harrier.png',
      'assets/mercec.png',
      'assets/creata.png'
    ];

    return Scaffold(
      backgroundColor: Colors.black,
      appBar: AppBar(
        backgroundColor: Colors.black,
        elevation: 0,
        title: Row(
          children: [
            Image.asset('assets/car.png', height: 40),
            const SizedBox(width: 10),
            const Text(
              "CarConnect",
              style: TextStyle(
                color: Colors.white,
                fontSize: 22,
                fontWeight: FontWeight.bold,
                fontFamily: 'Roboto',
                letterSpacing: 1.2,
              ),
            ),
          ],
        ),
        actions: [
          IconButton(
            icon: const Icon(Icons.logout, color: Colors.white),
            onPressed: () async {
              await FirebaseAuth.instance.signOut();
              Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) => const
LoginPage()),
              );
```



```
      },
    ),
  ],
),
body: SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Padding(
        padding: const EdgeInsets.all(16.0),
        child: TextField(
          style: const TextStyle(color: Colors.white),
          decoration: InputDecoration(
            hintText: "Search for your car e.g. Creta",
            hintStyle: TextStyle(color: Colors.white60),
            filled: true,
            fillColor: Colors.grey[900],
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(10),
              borderSide: BorderSide.none,
            ),
            prefixIcon: Icon(Icons.search, color: Colors.white60),
          ),
        ),
      ),
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            _filterButton("Compare", Icons.scale, onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => CarComparisonPage()),
              );
            }),
            _filterButton("Used", Icons.directions_car, onTap: () {
              Navigator.push(
                context,
                      MaterialPageRoute(builder: (context) =>
UsedCarScreen()),
                    );
                  }),
                  _filterButton("Budget", Icons.attach_money, onTap: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => BudgetPage(
                          carName:
                              "Your Selected Car", // Pass selected car
Name
                          exShowroomPrice:
                              1000000, // Pass selected car price
dynamically
                        ),
                      ),
                    );
                  }),
                  _filterButton("More", Icons.tune),
                ],
              ),
            ),
            Padding(
              padding: const EdgeInsets.all(16.0),
              child: ClipRRect(
                borderRadius: BorderRadius.circular(10),
                child: SizedBox(
                  width: double.infinity,
                  height: 400,
                  child: Image.asset(

'assets/pexels-samyantak-mohanty-79378681-8706096.png',
                    fit: BoxFit.cover,
                  ),
                ),
              ),
            ),
            const Padding(
              padding: EdgeInsets.all(16.0),
              child: Text(
                'Featured Cars',
                style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                    color: Colors.white),
              ),
            ),
            CarouselSlider(
              options: CarouselOptions(autoPlay: true, enlargeCenterPage:
true),
              items: featuredCarImages.map((imagePath) {
                return ClipRRect(
                  borderRadius: BorderRadius.circular(10),
                  child: Image.asset(imagePath, fit: BoxFit.cover),
                );
              }).toList(),
            ),
            const Padding(
              padding: EdgeInsets.all(16.0),
              child: Text(
                'Latest Car News',
                style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                    color: Colors.white),
              ),
            ),
            FutureBuilder<List<Map<String, String>>>(
              future: fetchCarNews(),
              builder: (context, snapshot) {
                if (snapshot.connectionState == ConnectionState.waiting) {
                  return const Center(
                      child: CircularProgressIndicator(color:
Colors.white));
                } else if (snapshot.hasError) {
                  return Center(
                    child: Text(
                      "Failed to load news",
                      style: TextStyle(color: Colors.white),
```



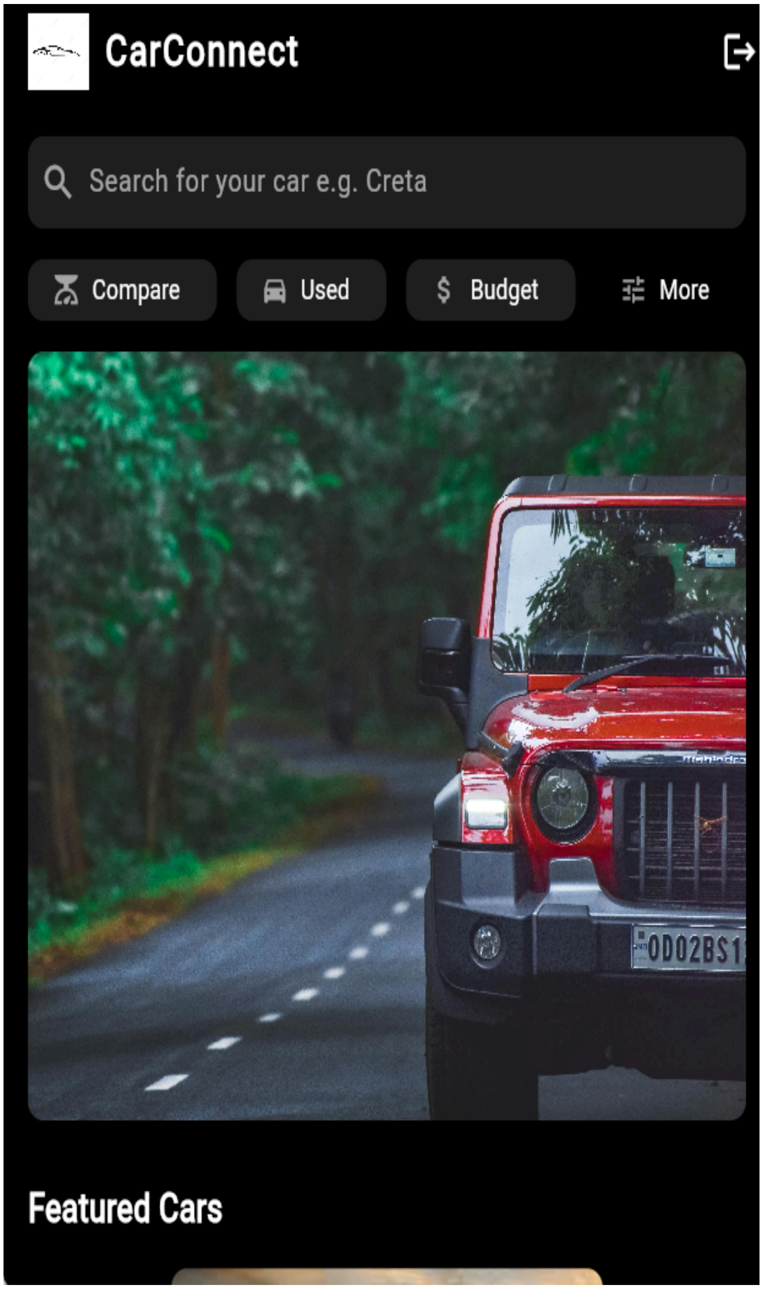
```
                    ),
                  );
                } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
                  return Center(
                    child: Text(
                      "No news available",
                      style: TextStyle(color: Colors.white),
                    ),
                  );
                }

                return SizedBox(
                  height: 250,
                  child: ListView.builder(
                    scrollDirection: Axis.horizontal,
                    itemCount: snapshot.data!.length,
                    itemBuilder: (context, index) {
                      final news = snapshot.data![index];
                      return GestureDetector(
                        onTap: () async {
                          Uri url = Uri.parse(news['url']!);
                          if (await canLaunchUrl(url)) {
                            await launchUrl(url);
                          }
                        },
                        child: Card(
                          color: Colors.grey[900],
                          margin: const EdgeInsets.symmetric(
                              horizontal: 8, vertical: 8),
                          shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(10),
                          ),
                          child: SizedBox(
                            width: 300,
                            child: Column(
                              crossAxisAlignment:
CrossAxisAlignment.start,
                              children: [
                                if (news['image']!.isNotEmpty)
                                  ClipRRect(
                                    borderRadius: const
BorderRadius.vertical(
                                        top: Radius.circular(10)),
                                    child: Image.network(news['image']!,
                                        fit: BoxFit.cover,
                                        height: 150,
                                        width: double.infinity),
                                  ),
                                Padding(
                                  padding: const EdgeInsets.all(10),
                                  child: Text(
                                    news['title']!,
                                    style: const TextStyle(
                                        color: Colors.white,
                                        fontSize: 16,
                                        fontWeight: FontWeight.bold),
                                    maxLines: 2,
                                    overflow: TextOverflow.ellipsis,
                                  ),
                                ),
                              ],
                            ),
                          ),
                        ),
                      );
                    },
                  ),
                );
              },
            ),
          ],
        ),
      ),
    );
  }

  Widget _filterButton(String title, IconData icon, {VoidCallback? onTap})
{
    return ElevatedButton.icon(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.grey[900],
        shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10)),
      ),
      onPressed: onTap,
      icon: Icon(icon, color: Colors.white60),
      label: Text(title, style: const TextStyle(color: Colors.white)),
    );
  }
}
```

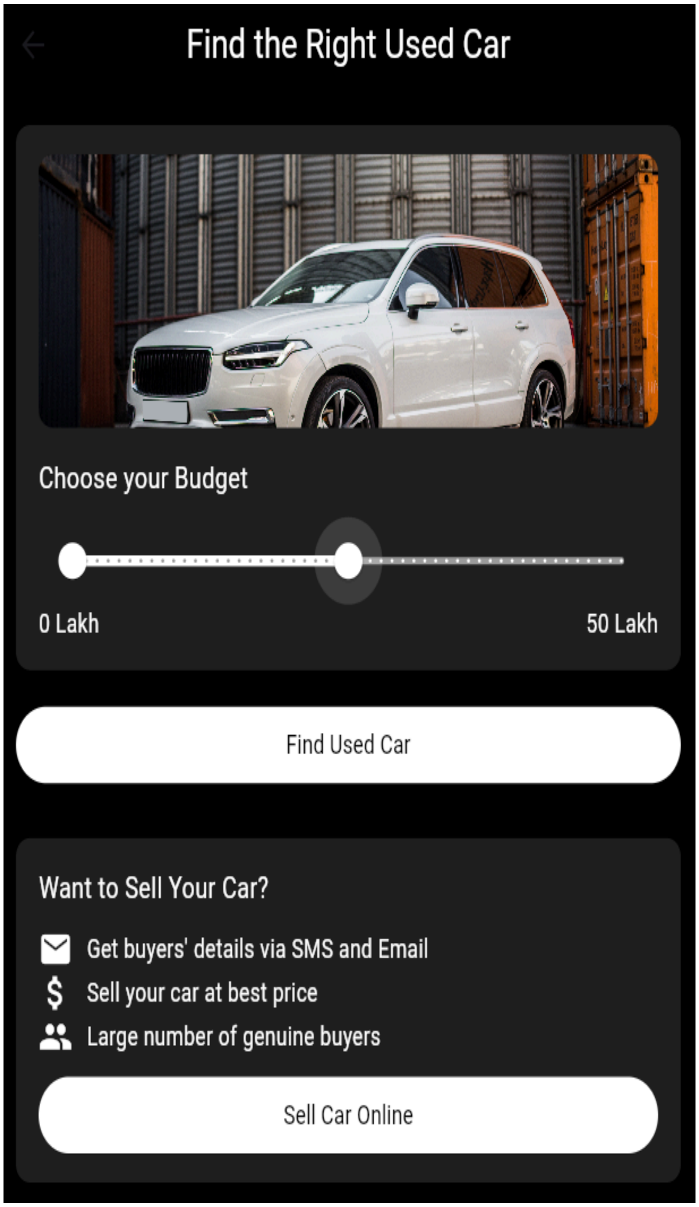
Budget page:

```
import 'package:flutter/material.dart';
import 'dart:math';

class BudgetPage extends StatefulWidget {
  final String? carName;
  final double? exShowroomPrice;
  final String? carType;

  BudgetPage({this.carName, this.exShowroomPrice, this.carType});

  @override
  _BudgetPageState createState() => _BudgetPageState();
}

class _BudgetPageState extends State<BudgetPage> {
  final TextEditingController _carNameController =
TextEditingController();
  final TextEditingController _priceController = TextEditingController();
  final TextEditingController _interestController =
TextEditingController();
  final TextEditingController _durationController =
TextEditingController();
  final TextEditingController _downPaymentController =
TextEditingController();

  String? carName, carType;
  double? exShowroomPrice, totalCost, loanEMI, totalLoanPayment;
  double gst = 0, cess = 0, roadTax = 0, insurance = 0, registration = 0, fastag = 500, accessories = 20000;

  @override
  void initState() {
    super.initState();
    if (widget.carName != null) {
      _carNameController.text = widget.carName!;
      carName = widget.carName;
    }
    if (widget.exShowroomPrice != null) {
      _priceController.text = widget.exShowroomPrice!.toString();
      exShowroomPrice = widget.exShowroomPrice;
    }
    if (widget.carType != null) {
      carType = widget.carType;
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Car Budget & Loan Calculator'),
        backgroundColor: Colors.black,
      ),
      backgroundColor: Colors.black,
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            _buildInputField(_carNameController, 'Car Name'),
            _buildInputField(_priceController, 'Ex-Showroom Price (₹)/Your estimated price', isNumeric: true),
            _buildDropdownField(),
            _buildInputField(_interestController, 'Interest Rate (%)', isNumeric: true),
            _buildInputField(_durationController, 'Loan Duration (Years)',
isNumeric: true),
            _buildInputField(_downPaymentController, 'Down Payment (₹,
Optional)', isNumeric: true),
            const SizedBox(height: 10),
            ElevatedButton(
              onPressed: _calculateTotalCost,
              child: const Text('Get Cost Breakdown'),
            ),
            if (totalCost != null) ...[
              _buildCostBreakdown(),
              ElevatedButton(
                onPressed: _showDetailedBreakdown,
                child: const Text('Show Detailed Breakdown'),
              ),
            ],
          ],
        ),
      ),
    );
  }

  Widget _buildDropdownField() {
    return DropdownButtonFormField<String>(
      value: carType,
      dropdownColor: Colors.black,
      style: const TextStyle(color: Colors.white),
      decoration: _inputDecoration('Select Car Type'),
      items: ['Small Car', 'Mid-Size Car', 'SUV', 'Electric Vehicle (EV)',
'Luxury Car (₹20L+)']
          .map((type) => DropdownMenuItem(value: type, child: Text(type,
style: const TextStyle(color: Colors.white))))
          .toList(),
      onChanged: (value) => setState(() => carType = value),
    );
  }

  Widget _buildInputField(TextEditingController controller, String label,
{bool isNumeric = false}) {
    return Padding(
      padding: const EdgeInsets.only(bottom: 10),
      child: TextField(
        controller: controller,
        keyboardType: isNumeric ? TextInputType.number :
TextInputType.text,
        style: const TextStyle(color: Colors.white),
        decoration: _inputDecoration(label),
        onChanged: (value) {
          if (controller == _priceController) {
            setState(() {
              exShowroomPrice = double.tryParse(value) ?? 0;
            });
          }
        },
      ),
    );
  }

  InputDecoration _inputDecoration(String label) {
    return InputDecoration(
      labelText: label,
      labelStyle: const TextStyle(color: Colors.white),
      border: const OutlineInputBorder(),
    );
  }

  void _calculateTotalCost() {
    double interestRate = double.tryParse(_interestController.text) ?? 0;
    int loanDuration = int.tryParse(_durationController.text) ?? 0;
    double downPayment = double.tryParse(_downPaymentController.text) ??
0;

    if (exShowroomPrice == null || carType == null) return;

    // Apply tax rules based on car type
    if (carType == 'Small Car') {
      gst = exShowroomPrice! * 0.28;
      cess = exShowroomPrice! * 0.03;
      roadTax = exShowroomPrice! * 0.08;
      insurance = exShowroomPrice! * 0.05;
      registration = 5000;
    } else if (carType == 'Mid-Size Car') {
      gst = exShowroomPrice! * 0.28;
      cess = exShowroomPrice! * 0.15;
      roadTax = exShowroomPrice! * 0.10;
      insurance = exShowroomPrice! * 0.05;
      registration = 10000;
    } else if (carType == 'SUV') {
      gst = exShowroomPrice! * 0.28;
      cess = exShowroomPrice! * 0.22;
      roadTax = exShowroomPrice! * 0.12;
      insurance = exShowroomPrice! * 0.06;
      registration = 15000;
    } else if (carType == 'Electric Vehicle (EV)') {
      gst = exShowroomPrice! * 0.05;
      cess = 0;
      roadTax = 0;
      insurance = exShowroomPrice! * 0.04;
      registration = 0;
    } else if (carType == 'Luxury Car (₹20L+)') {
      gst = exShowroomPrice! * 0.28;
      cess = exShowroomPrice! * 0.22;
      roadTax = exShowroomPrice! * 0.15;
      insurance = exShowroomPrice! * 0.06;
      registration = 20000;
    }

    // Calculate total on-road cost
    totalCost = exShowroomPrice! + gst + cess + roadTax + insurance +
registration + fastag + accessories;

    // Calculate Loan EMI & Total Loan Cost
    double loanAmount = totalCost! - downPayment;
    double monthlyRate = (interestRate / 100) / 12;
    int totalMonths = loanDuration * 12;

    if (loanAmount > 0 && monthlyRate > 0 && totalMonths > 0) {
      loanEMI = (loanAmount * monthlyRate * pow(1 + monthlyRate,
totalMonths)) /
          (pow(1 + monthlyRate, totalMonths) - 1);
```

```dart
      totalLoanPayment = loanEMI! * totalMonths;
    }

    setState(() {});
  }

  Widget _buildCostBreakdown() {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text('Breakdown for $carName ($carType)', style: const TextStyle(color: Colors.white, fontSize: 18, fontWeight: FontWeight.bold)),
        _buildCostRow('Total On-Road Price (Without Loan)', totalCost!, isTotal: true),
        if (loanEMI != null && totalLoanPayment != null) ...[
          _buildCostRow('Monthly EMI', loanEMI!, isTotal: true),
          _buildCostRow('Total Cost with Loan & Interest', totalLoanPayment!, isTotal: true),
        ],
      ],
    );
  }

  Widget _buildCostRow(String title, double amount, {bool isTotal = false}) {
    return Padding(
      padding: const EdgeInsets.symmetric(vertical: 6.0),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Text(title, style: TextStyle(color: Colors.white, fontSize: isTotal ? 18 : 16, fontWeight: isTotal ? FontWeight.bold : FontWeight.normal)),
          Text('₹${amount.toStringAsFixed(2)}', style: TextStyle(color: Colors.white, fontSize: isTotal ? 18 : 16, fontWeight: isTotal ? FontWeight.bold : FontWeight.normal)),
        ],
      ),
    );
  }

  void _showDetailedBreakdown() {
    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(
          backgroundColor: Colors.black,
          title: Text('Detailed Cost Breakdown for $carName ($carType)',
style: const TextStyle(color: Colors.white)),
          content: SingleChildScrollView(
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                _buildCostRow('Ex-Showroom Price', exShowroomPrice!),
                _buildCostRow('GST', gst),
                _buildCostRow('Cess', cess),
                _buildCostRow('Road Tax', roadTax),
                _buildCostRow('Insurance', insurance),
                _buildCostRow('Registration', registration),
                _buildCostRow('Fastag', fastag),
                _buildCostRow('Accessories', accessories),
                _buildCostRow('Total On-Road Price', totalCost!, isTotal:
true),
                if (loanEMI != null && totalLoanPayment != null) ...[
                  _buildCostRow('Loan Amount', totalCost! -
(double.tryParse(_downPaymentController.text) ?? 0)),
                  _buildCostRow('Monthly EMI', loanEMI!),
                  _buildCostRow('Total Loan Payment', totalLoanPayment!),
                ],
              ],
            ),
          ),
          actions: [
            TextButton(
              onPressed: () => Navigator.of(context).pop(),
              child: const Text('Close', style: TextStyle(color:
Colors.white)),
            ),
          ],
        );
      },
    );
  }
}
```

Output:


CarConnect
Search for your car e.g. Creta
Compare
Used
Budget
More
OD02BS1
Featured Cars

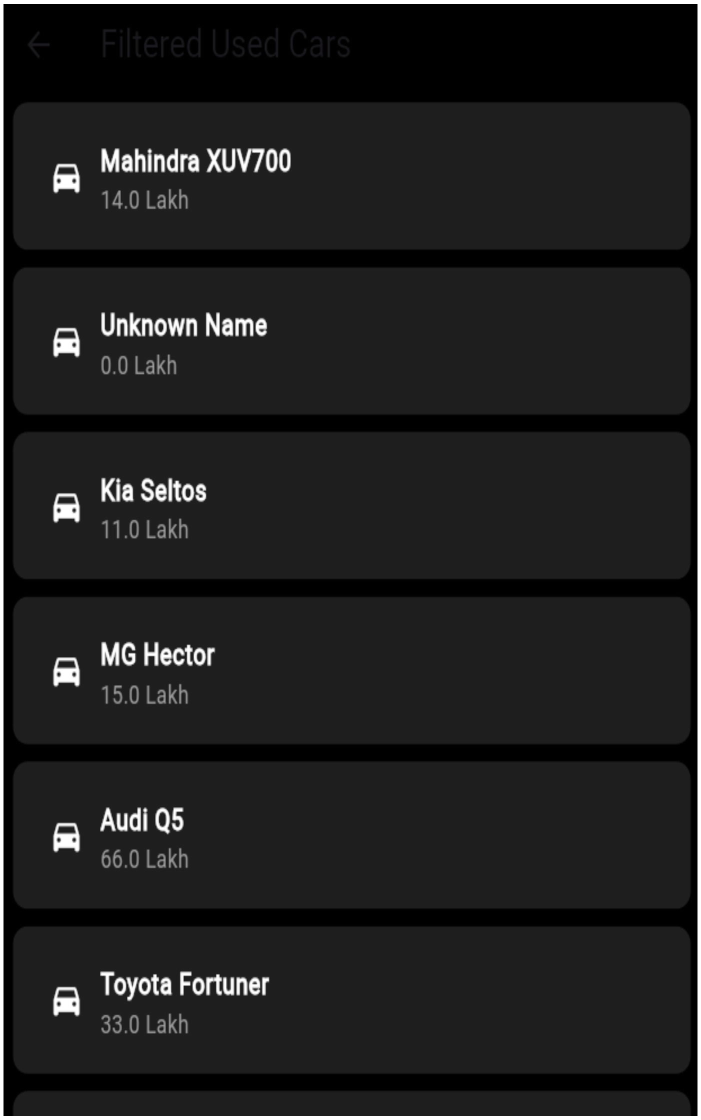


Find the Right Used Car

Choose your Budget

Want to Sell Your Car?

- ✉ Get buyers' details via SMS and Email
- $ Sell your car at best price
- 👥 Large number of genuine buyers



Filtered Used Cars

Mahindra XUV700
14.0 Lakh

Unknown Name
0.0 Lakh

Kia Seltos
11.0 Lakh

MG Hector
15.0 Lakh

Audi Q5
66.0 Lakh

Toyota Fortuner
33.0 Lakh