

# MPL Prerequisites Document

Name: Mohit Patil

Class: D15A

Roll No. 36

## Aim:

Selecting features for application development, the features should comprise of:

1. Common widgets
2. Should include icons, images, charts etc.
3. Should have an interactive Form
4. Should apply navigation, routing and gestures
5. Should connect with Fire Base database

1:16 AM ☀️

4G Vo WiFi 82%

←

### Login or Register

For a better experience, order tracking & regular updates

Send OTP

By proceeding, I confirm that I have received, read, and agree to: [Third-Party Sharing with Dealers and OEMs](#), [Data Sharing for Loan Eligibility, Marketing Communication](#), [Privacy Policy](#), [Terms & Conditions](#), [Consent Declaration](#), and [Certified CIC](#).



**Landing Page : Features**

1. Sign in using email address /Mobile number
2. Firebase for authentication
3. Firebase :
4. User : Username , email and password

12:22 AM   ⚙️

Vo WiFi 85%

**carwale**

Search for your car e.g. Creta

Find The Right Car By

Budget    Body Type    Fuel Type

Meridian YOUR BIG ENTRY INTO 2025  
THE MERIDIAN RANGE STARTS AT ₹24.99 LAKH\*  
EMI STARTS AT ₹29,999\* AVAILABLE IN 5 & 7 SEATER KNOW MORE

### Featured Cars

POPULAR   JUST LAUNCHED   UPCOMING

New   Used   Compare   News   Alerts

### Home:

1. Navigation Buttons
2. Icons for like and location
3. Search Bar
4. Display best offers on cars

12:22 AM ⌂ ⚙ 4G WiFi 85%

## Compare Cars

Select Car

Select Car

+ Add Car

Compare

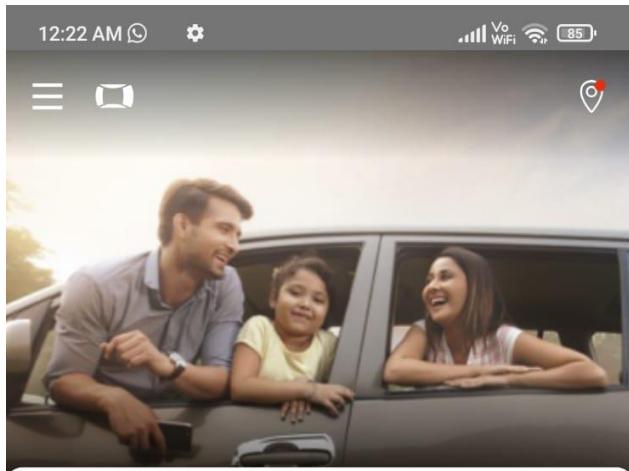
### POPULAR CAR COMPARISONS

|                                      |    |   |
|--------------------------------------|----|---|
|                                      | VS |   |
| Kia Syros<br>Rs. 10.52 Lakh onwards  |    | Kia Sonet<br>Rs. 9.34 Lakh onwards        |
|                                      | VS |   |
| Skoda Kylaq<br>Rs. 9.15 Lakh onwards |    | Mahindra XUV 3XO<br>Rs. 9.37 Lakh onwards |

New      Used      Compare      News      Alerts

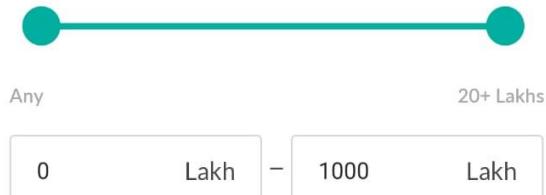
**Compare car page:**

1. Compare and add car Buttons
2. Option to select two cars to compare between
3. Search Bar
4. Display car prices with offers



## FIND THE RIGHT USED CAR

Choose your Budget



**Find Used Car**

Want to Sell Your Car ?

Get buyers details via SMS and Email

New

Used

Compare

News

Alerts

## Budget car finding Page:

- Allows you to set your budget price and helps in finding the best budget car.

12:23 AM

Vo WiFi 84%

< Sell Cars

## Sell Car Online for Free

Sell your used car at the best price in a few clicks for free. 60000+ sellers across India sell their old car...[more](#)

Enter your car registration number

**Sell your Car**

Or Select Your Car Brand

|   |   |  |
|---|---|--|
|  Maruti Suzuki |  Hyundai   |  Honda      |
|  Tata          |  Chevrolet |  Ford       |
|  Toyota        |  Mahindra  |  Volkswagen |

[View All Brands](#)

 Already Listed Your Car?  
Please login and check activities on your car ad

**Compare car page:**

1. Option to search the required car to be sold
2. Option to search car by the brand to be sold

## MPL Experiment 1

Name: Mohit Patil

Class: D15A

Roll no:36

Aim: Installation and Configuration of Flutter Environment.

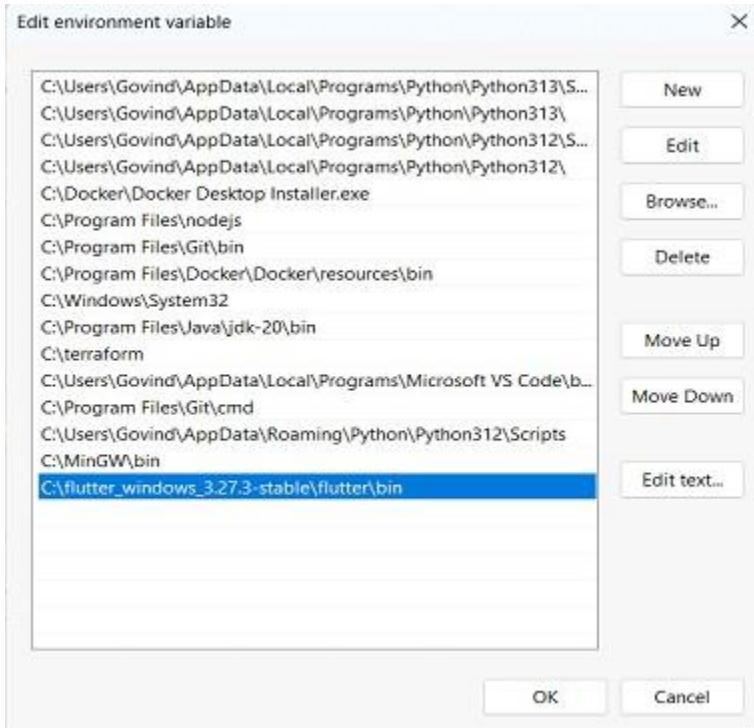
### Step 1: Install Flutter

- Download Flutter SDK from the official Flutter website (<https://flutter.dev/>).
- Download the Flutter SDK for your operating system (Windows, macOS, or Linux).

The screenshot shows the official Flutter website's 'Get started' section. At the top, there's a navigation bar with links for Multi-Platform, Development, Ecosystem, Showcase, Docs, and a search icon. A prominent blue button labeled 'Get started' is on the right. Below the navigation, a banner says 'Celebrating Flutter's production era! Learn more' and 'Also, check out What's new on the website.' On the left, a sidebar menu includes 'Get started', 'Set up Flutter', 'Learn Flutter', 'Stay up to date', 'App solutions', 'User interface', 'Introduction', 'Widget catalog', 'Layout', 'Adaptive & responsive design', and 'Design & theming'. The main content area has a heading 'Choose your development platform to get started' with four options: Windows (Current device), macOS, Linux, and ChromeOS. Below this, there's a note about developing in China: 'If you want to use Flutter in China, check out using Flutter in China. If you're not developing in China, ignore this notice.' and a link to 'Follow the other instructions on this page.'

The screenshot shows the 'Install the Flutter SDK' page. The top navigation bar and sidebar are identical to the previous screenshot. The main content area has a heading 'Install the Flutter SDK' with a sub-instruction: 'To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.' Below this are two buttons: 'Use VS Code to install' and 'Download and install'. To the right, there's a 'Contents' sidebar with links for Verify system requirements, Hardware requirements, Software requirements, Configure a text editor or IDE, Install the Flutter SDK, Configure Android development, Configure the Android toolchain in Android Studio, Configure your target Android device, Agree to Android licenses, Check your development setup, Run Flutter doctor, Troubleshoot Flutter doctor issues, and Start developing Android on Windows apps with Flutter. There's also a note: 'The Flutter SDK should download to the Windows default download directory: %USERPROFILE%\Downloads\'. If you changed the location of the Downloads directory, replace this path with that path. To find your Downloads directory location, check out this Microsoft Community post.'

- Extract the downloaded zip file to a preferred location on your computer (e.g., C:\src\flutter for Windows).
- Add Flutter to the PATH
- Locate the flutter\bin directory in the extracted Flutter folder.
- Add this directory to your system's PATH environment variable.



## Verify the Installation

- Open a terminal or command prompt.
- Run the command: **flutter** and **flutter doctor**.

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.1, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[✗] Windows Version (the doctor check crashed)
  X Due to an error, the doctor check did not complete. If the error message below is not helpful, please let us know
    about this issue at https://github.com/flutter/flutter/issues.
  X ProcessException: Failed to find "powershell" in the search path.
    Command: powershell
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[✗] Visual Studio - develop Windows apps
  X Visual Studio not installed; this is necessary to develop Windows apps.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2023.3)
[✓] VS Code (version 1.96.4)
[✓] Connected device (4 available)
[✓] Network resources

! Doctor found issues in 2 categories.

C:\Users\User>
```

```
Command Prompt - flutter X + v

Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/to/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

To disable animations in this tool, use
'flutter config --no-cli-animations'.

The Flutter CLI developer tool uses Google Analytics to report usage and diagnostic
data along with package dependencies, and crash reporting to send basic crash
reports. This data is used to help improve the Dart platform, Flutter framework,
and related tools.

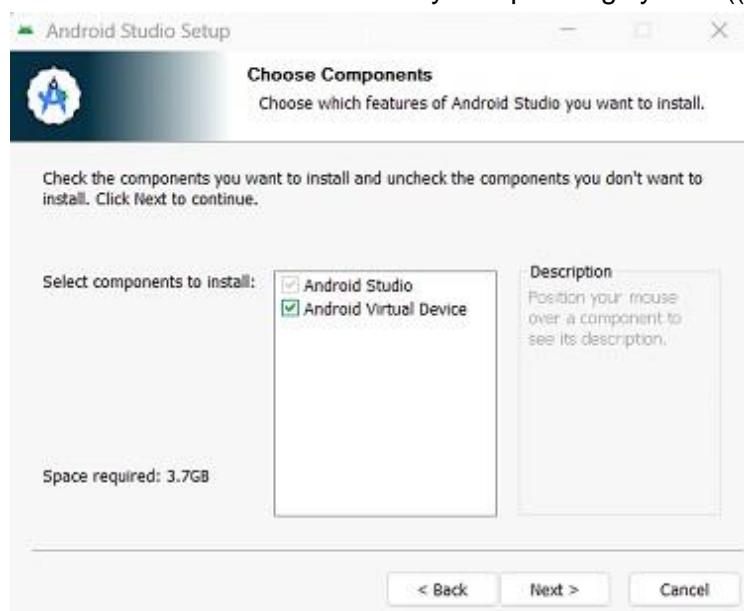
Telemetry is not sent on the very first run. To disable reporting of telemetry,
run this terminal command:

  flutter --disable-analytics

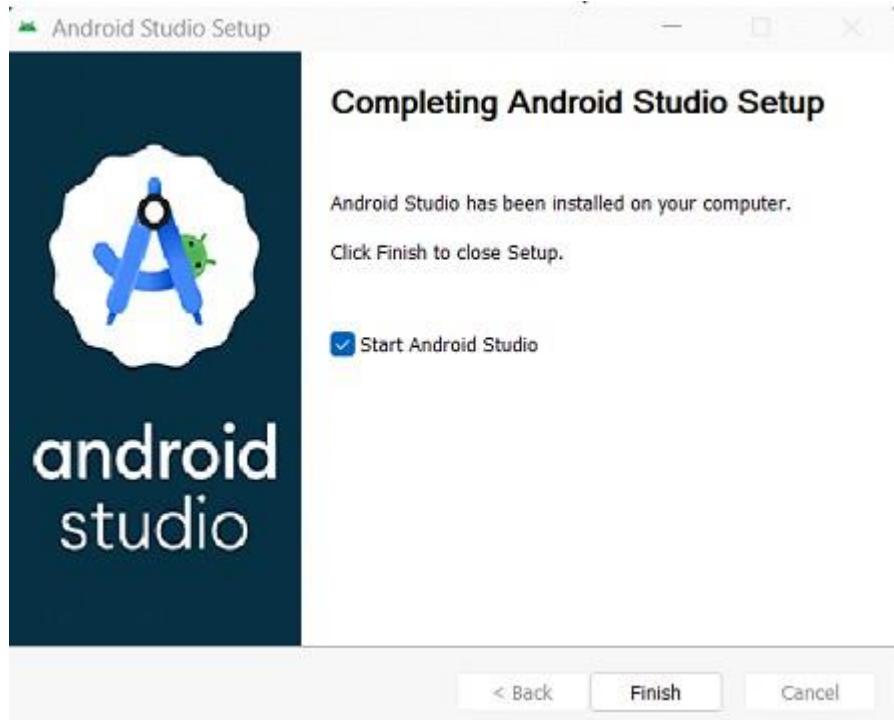
If you opt out of telemetry, an opt-out event will be sent, and then no further
```

## Step 2: Install Android Studio

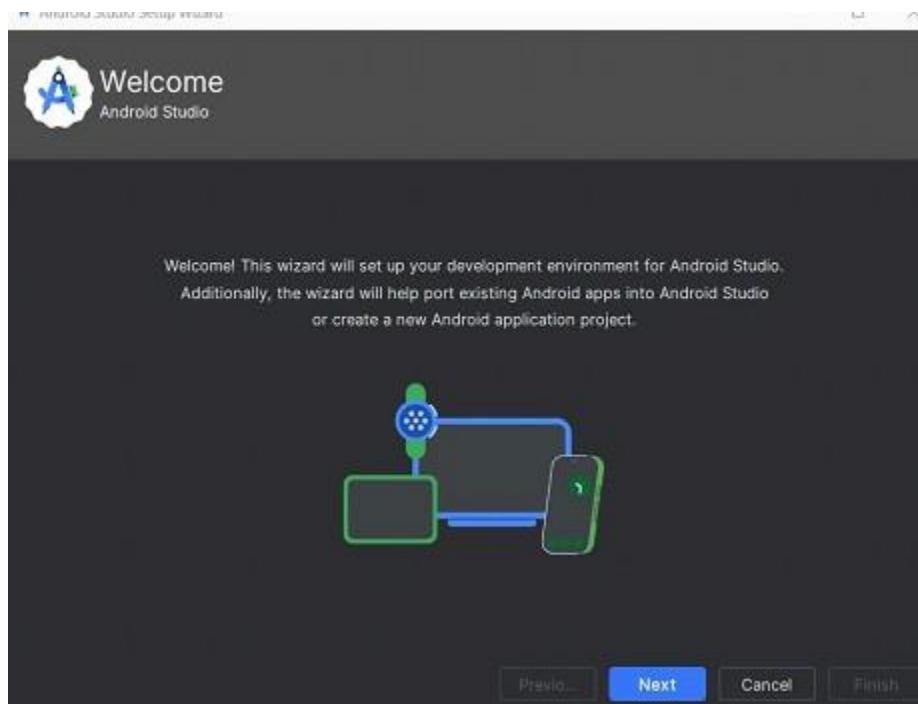
- Download Android Studio
- Go to the Android Studio website. (<https://developer.android.com/studio>)
- Download the installer for your operating system ((Windows, macOS, or Linux)).



- Run the installer and follow the setup wizard.
- Choose the standard installation option.

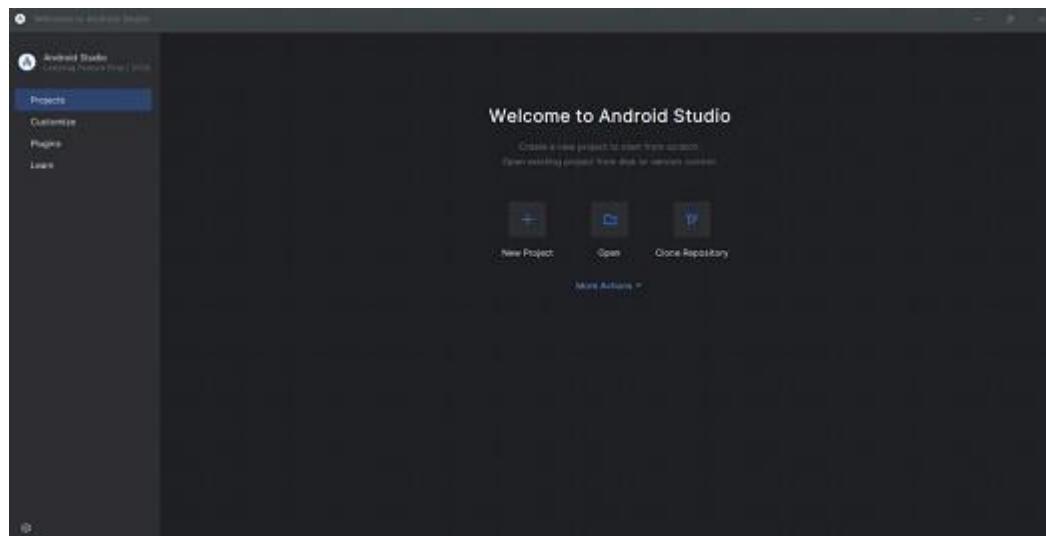
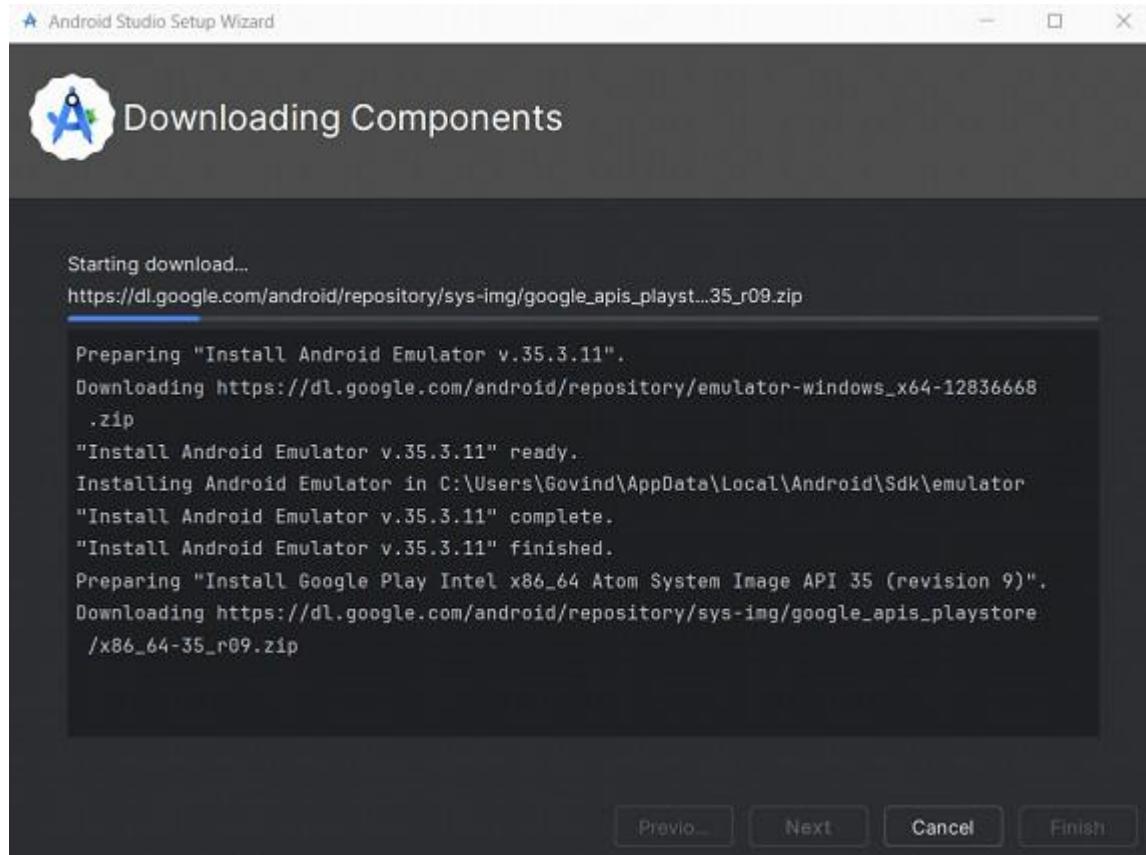


- Install Android SDK Tools
- Open Android Studio.



- Go to Settings/Preferences > Appearance & Behavior > System Settings > Android SDK.

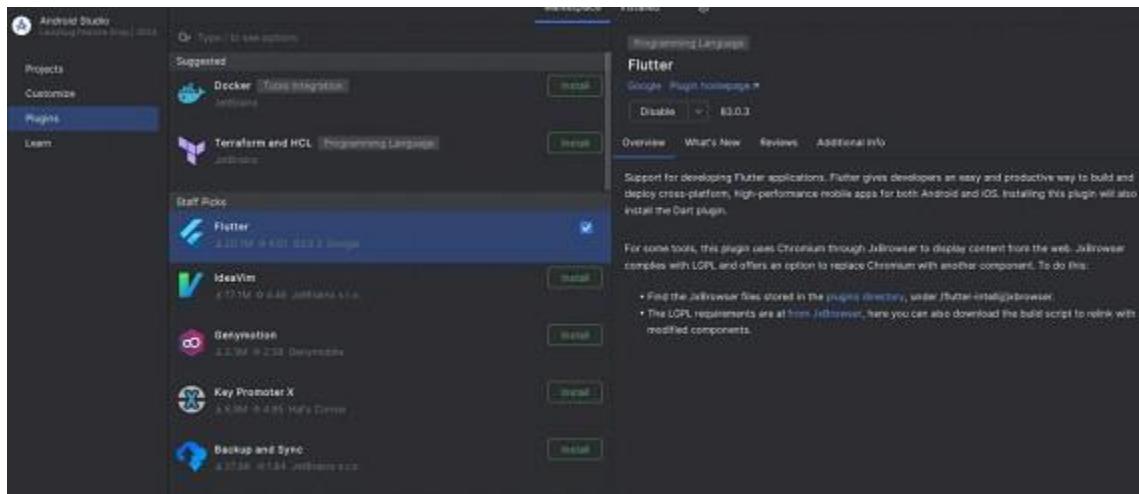
- Select the latest Android API level.
- Ensure "Android SDK Platform" and "Android Virtual Device (AVD)" are selected.
- Click "Apply" and wait for the components to install.



### Step 3: Connect Flutter with Android Studio

- Install Flutter and Dart Plugins
- Open Android Studio. Go to File > Settings (Windows/Linux) > Plugins.

- Search for "Flutter" and click "Install." Dart will be installed automatically.
- Restart Android Studio.



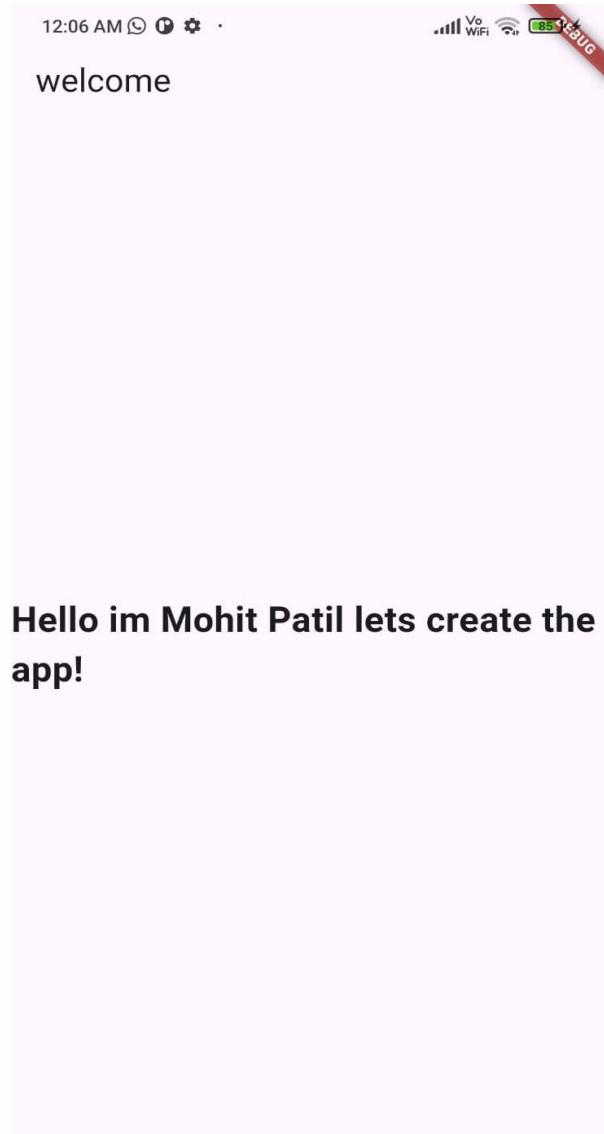
#### Step 4: Create a New Flutter Project

- Click on New Flutter Project.
- Enter project details and select the Flutter SDK path.
- Click "Finish" to create the project.
- Connect the USB to the device and run the flutter application.

**NOTE:** In your mobile device, make sure the USB debugging is turned on.

#### Code:

```
proj1 > lib > main.dart > MyApp > build
1   import 'package:flutter/material.dart';
2
3   Run | Debug | Profile | Codeium: Refactor | Explain | Generate Function Comment | X
4   void main() {
5     runApp(app: const MyApp());
6   }
7
8   Codeium: Refactor | Explain
9   class MyApp extends StatelessWidget {
10    const MyApp({super.key});
11
12    @override
13    Codeium: Refactor | Explain | Generate Function Comment | X
14    Widget build(BuildContext context) {
15      return MaterialApp(
16        home: Scaffold(
17          appBar: AppBar(
18            title: const Text(data: 'welcome'),
19          ), // AppBar
20          body: const Center(
21            child: Text(
22              data: 'Hello im Mohit Patil lets create the app!',
23              style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
24            ), // Text
25            ), // Center
26            ), // Scaffold
27        ); // MaterialApp
28    }
29 }
```



**Conclusion:** Hello message , is successfully run on the flutter app.

## **MPL Experiment 2**

**Name:** Mohit Patil

**Class:** D15A

**Roll no:**36

**Aim:** To design Flutter UI by including common widgets.

**Code:**

```
//Home screen.dart
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=4c05b7fc955e4736bd656882
83299b74';

        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),
    ),
)

```



```

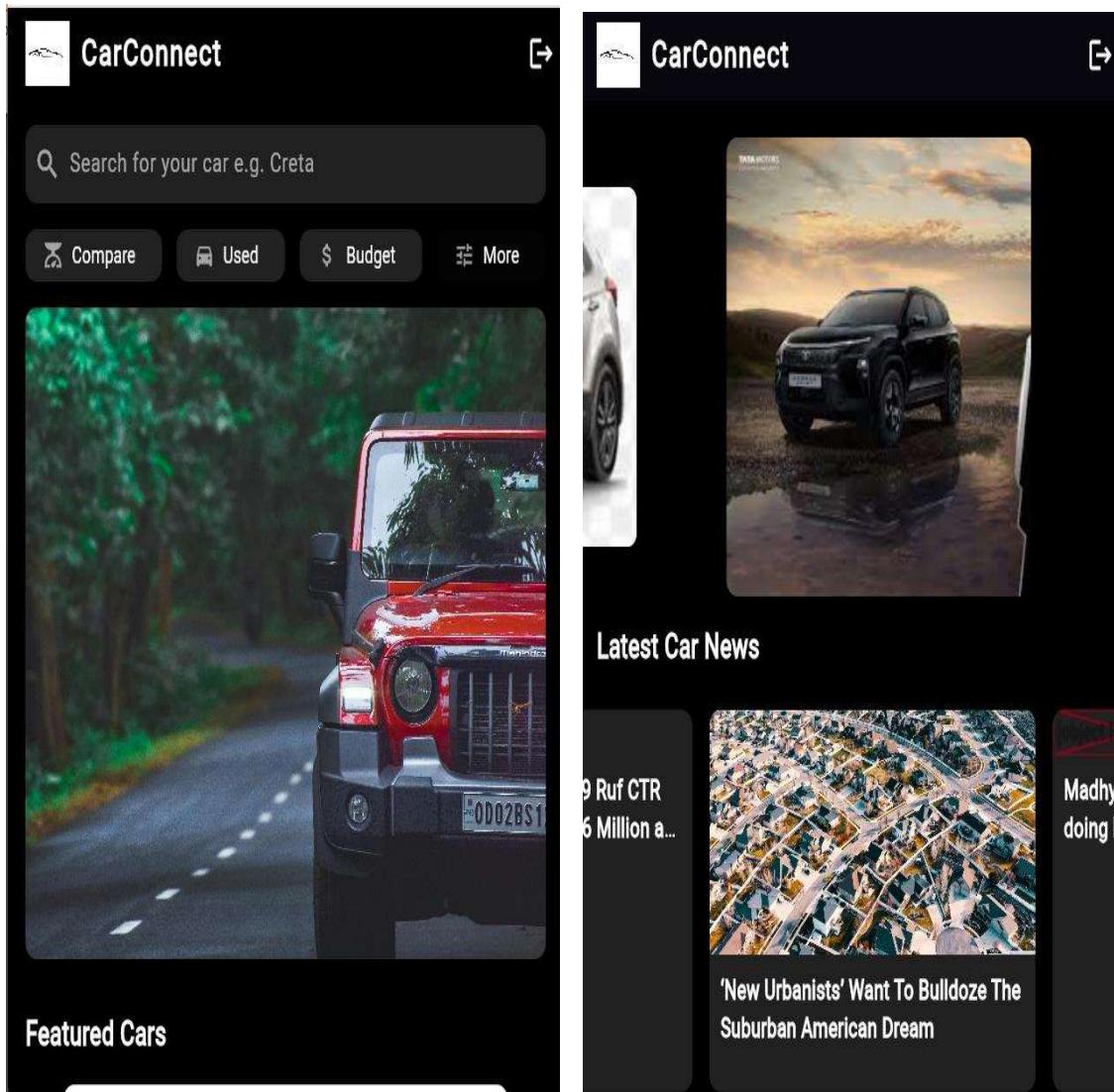
        ClipRRect(
            borderRadius: const
BorderRadius.vertical(
                top: Radius.circular(10)),
            child:
Image.network(news['image']!,,
                fit: BoxFit.cover,
                height: 150,
                width: double.infinity),
        ),
        Padding(
            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)),
    );
}

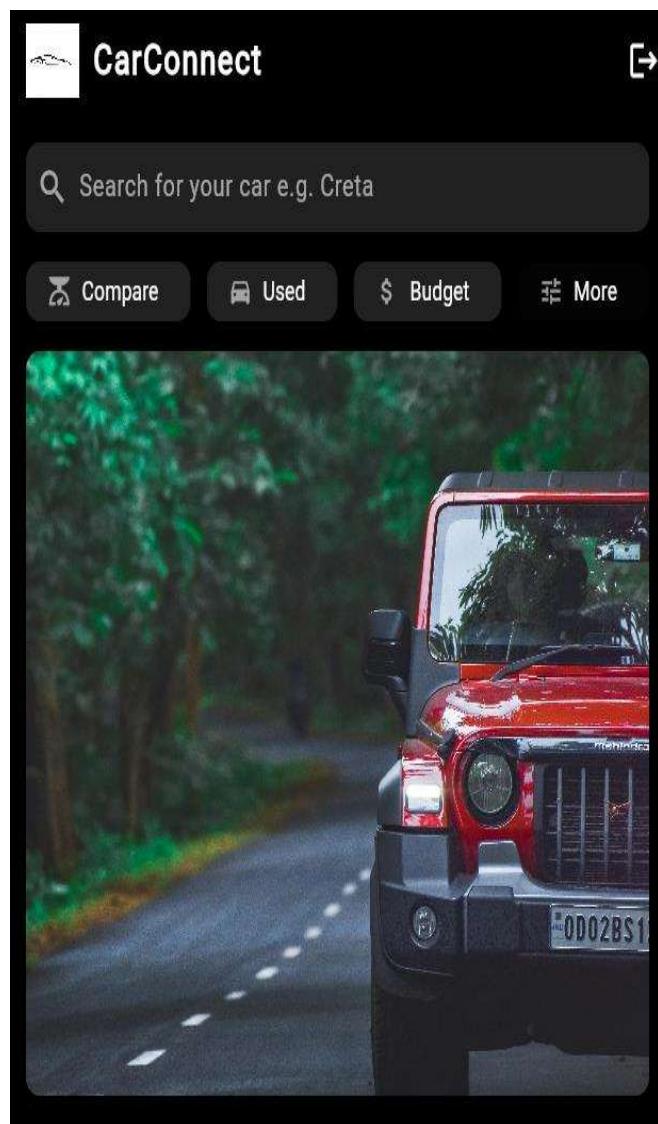
```

**Output:**

**HomePage:**



**Widget :**



**Widget are created above home screen in the form of buttons that is used to direct to another pages i.e .Compare ,Used ,Budget**

## 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(
        mainAxisAlignment: 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,
```



```
        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);
}

```

```
        totalLoanPayment = loanEMI! * totalMonths;
    }

    setState(() {});
}

Widget _buildCostBreakdown() {
    return Column(
        mainAxisAlignment: CrossAxisAlignmentAlignment.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),
            ],
        ],
    );
}

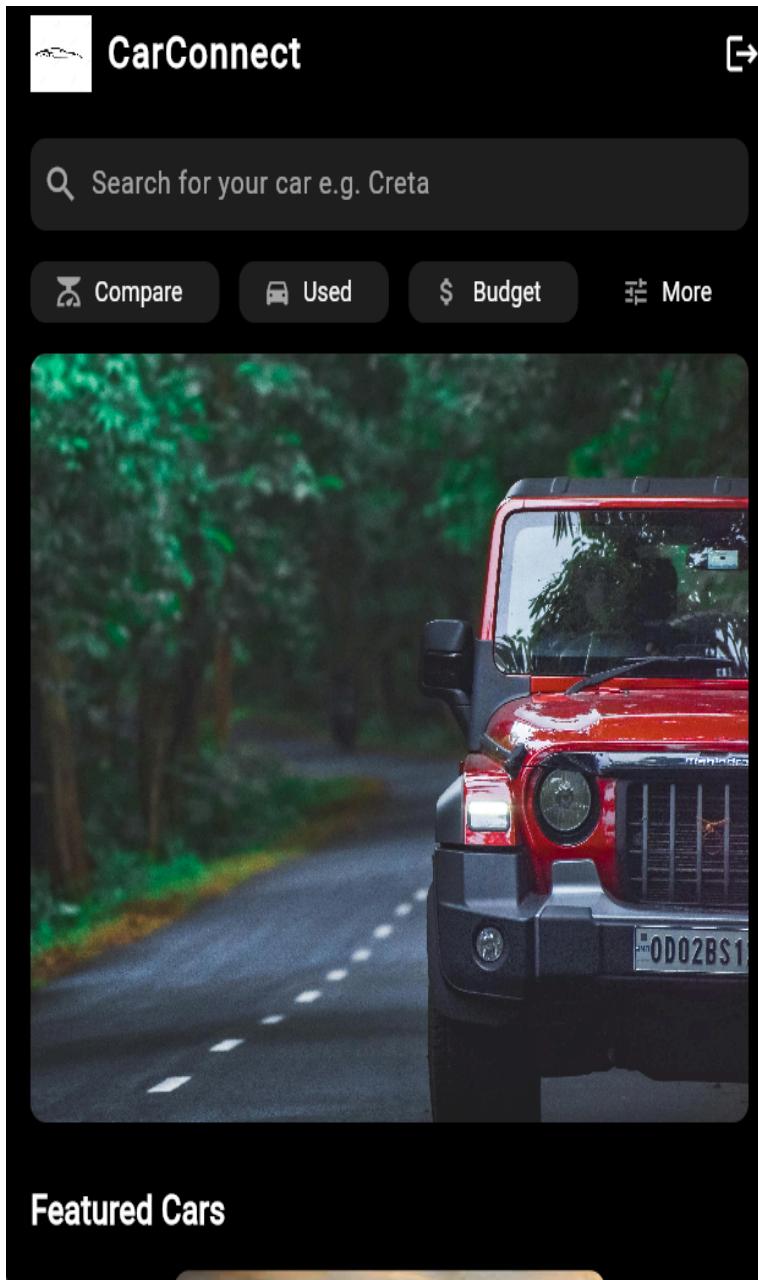
Widget _buildCostRow(String title, double amount, {bool isTotal =
false}) {
    return Padding(
        padding: const EdgeInsets.symmetric(vertical: 6.0),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
                Text(title, style: TextStyle(color: 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(
                        mainAxisAlignment: MainAxisAlignment.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:**



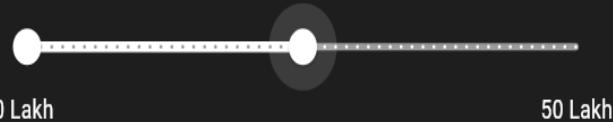




## Find the Right Used Car



Choose your Budget



Find Used Car

Want to Sell Your Car?

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

Sell Car Online



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

# Experiment No. 4

Mohit Patil -D15A-36

**AIM :** To create an interactive Form using form widget.

## Widgets :

TextField: For user input (e.g., username, password).

FlatButton or ElevatedButton: For submitting login credentials. Text: To display static text like "Welcome" or error messages.

Column: To arrange multiple widgets vertically (e.g., input fields and buttons).

Row: For horizontally aligning widgets (e.g., forgot password link and other actions).

Padding: To add space around widgets.

Icon: To show icons in the input fields or buttons.

Container: For custom styling of sections, buttons, or inputs. Form: To handle validation of login inputs.

GestureDetector: For detecting taps (e.g., to navigate or submit).

ThemeData: To define the overall visual properties of your app (colors, fonts, etc.).

MaterialApp: To wrap your app with a theme and other configurations.

## Code :

- Register page code:

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'login.dart';

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

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

class _RegisterPageState extends State<RegisterPage> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  void _registerUser() async {
    if (_formKey.currentState!.validate()) {
      try {
```

```
        await _auth.createUserWithEmailAndPassword(
            email: _emailController.text.trim(),
            password: _passwordController.text.trim(),
        );
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Registration Successful!')),
        );
        Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => const LoginPage()),
        );
    } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text('Registration Failed: $e')),
        );
    }
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        body: Container(
            color: Colors.black,
            child: Center(
                child: Padding(
                    padding: const EdgeInsets.all(16.0),
                    child: Card(
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(20),
                        ),
                        elevation: 10,
                        color: Colors.white,
                        child: Padding(
                            padding: const EdgeInsets.all(30.0),
                            child: Form(
                                key: _formKey,
                                child: Column(
                                    mainAxisAlignment: MainAxisAlignment.min,
                                    crossAxisAlignment: CrossAxisAlignment.stretch,
                                    children: [
                                        Column(
                                            children: [
                                                Image.asset(
                                                    'assets/c1.png',
                                                    height: 150,
                                                ),
                                                const SizedBox(height: 10),
                                                Text(

```

```
        'CarConnect',
        style: GoogleFonts.alegreyaSans(
            fontSize: 40,
            fontWeight: FontWeight.bold,
            color: Colors.black,
        ),
    ),
),
],
),
),
const SizedBox(height: 20),
TextField(
    controller: _emailController,
    decoration: InputDecoration(
        labelText: 'Email',
        border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
        ),
        prefixIcon:
            const Icon(Icons.email, color: Colors.black),
    ),
    validator: (value) {
        if (value == null ||
            value.isEmpty ||
            !value.contains('@')) {
            return 'Enter a valid email';
        }
        return null;
    },
),
const SizedBox(height: 15),
TextField(
    controller: _passwordController,
    obscureText: true,
    decoration: InputDecoration(
        labelText: 'Password',
        border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
        ),
        prefixIcon:
            const Icon(Icons.lock, color: Colors.black),
    ),
    validator: (value) {
        if (value == null || value.length < 6) {
            return 'Password must be at least 6 characters';
        }
        return null;
    },
),
const SizedBox(height: 20),
```



- Login page code:

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';

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

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

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  // Use the renderButton method for Google Sign-In on web
  void _signInWithGoogle() async {
    final GoogleSignIn googleSignIn = GoogleSignIn();

    try {
      GoogleSignInAccount? googleUser = await googleSignIn.signIn();
      if (googleUser == null) return;

      final GoogleSignInAuthentication googleAuth =
          await googleUser.authentication;

      final AuthCredential credential = GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );

      await _auth.signInWithCredential(credential);
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Google Sign-In Successful!')),
      );
    } catch (e) {
      print("Google Sign-In Error: $e");
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Google Sign-In Failed: $e')),
      );
    }
  }

  void _signInWithEmailAndPassword() async {
```

```
if (_formKey.currentState!.validate()) {
    try {
        await _auth.signInWithEmailAndPassword(
            email: _emailController.text.trim(),
            password: _passwordController.text.trim(),
        );
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Login Successful!')),
        );

        // Redirect to Home Page after login
        Navigator.pushReplacementNamed(context, '/home');
    } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text('Login Failed: $e')),
        );
    }
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        body: Container(
            color: Colors.black,
            child: Center(
                child: Padding(
                    padding: const EdgeInsets.all(16.0),
                    child: Card(
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(20),
                        ),
                        elevation: 10,
                        color: Colors.white,
                        child: Padding(
                            padding: const EdgeInsets.all(30.0),
                            child: Form(
                                key: _formKey,
                                child: Column(
                                    mainAxisAlignment: MainAxisAlignment.min,
                                    crossAxisAlignment: CrossAxisAlignment.stretch,
                                    children: [
                                        Column(
                                            children: [
                                                Image.asset(
                                                    'assets/c1.png', // Ensure image is in assets folder
                                                    height: 150,
                                                ),
                                                const SizedBox(height: 10),
                                            ],
                                        ),
                                    ],
                                ),
                            ),
                        ),
                    ),
                ),
            ),
        ),
    );
}
```

```
        Text(
            'CarConnect',
            style: GoogleFonts.alegreyaSans(
                fontSize: 40,
                fontWeight: FontWeight.bold,
                color: Colors.black,
            ),
            ),
            ],
        ),
        const SizedBox(height: 20),
        TextFormField(
            controller: _emailController,
            decoration: InputDecoration(
                labelText: 'Email',
                border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(12),
                ),
                prefixIcon:
                    const Icon(Icons.email, color: Colors.black),
            ),
            validator: (value) {
                if (value == null ||
                    value.isEmpty ||
                    !value.contains('@')) {
                    return 'Enter a valid email';
                }
                return null;
            },
        ),
        const SizedBox(height: 15),
        TextFormField(
            controller: _passwordController,
            obscureText: true,
            decoration: InputDecoration(
                labelText: 'Password',
                border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(12),
                ),
                prefixIcon:
                    const Icon(Icons.lock, color: Colors.black),
            ),
            validator: (value) {
                if (value == null || value.length < 6) {
                    return 'Password must be at least 6 characters';
                }
                return null;
            },
        ),
    ),
)
```



- **Main.dart file code:**

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:proj1/screens/login.dart';
import 'package:proj1/screens/RegisterPage.dart';
import 'package:proj1/screens/home.dart'; // Import Home Page
import 'firebase_options.dart';

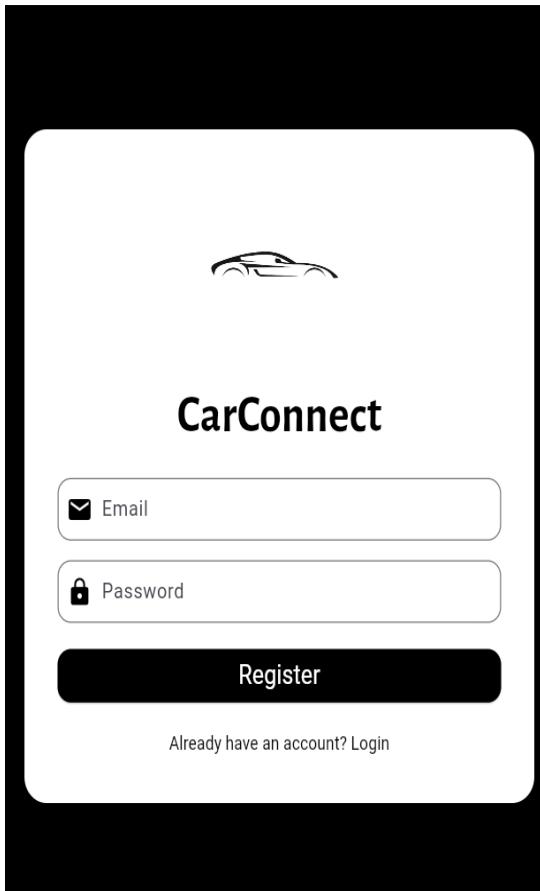
void main() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp(
        options: DefaultFirebaseOptions.currentPlatform,
    );
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

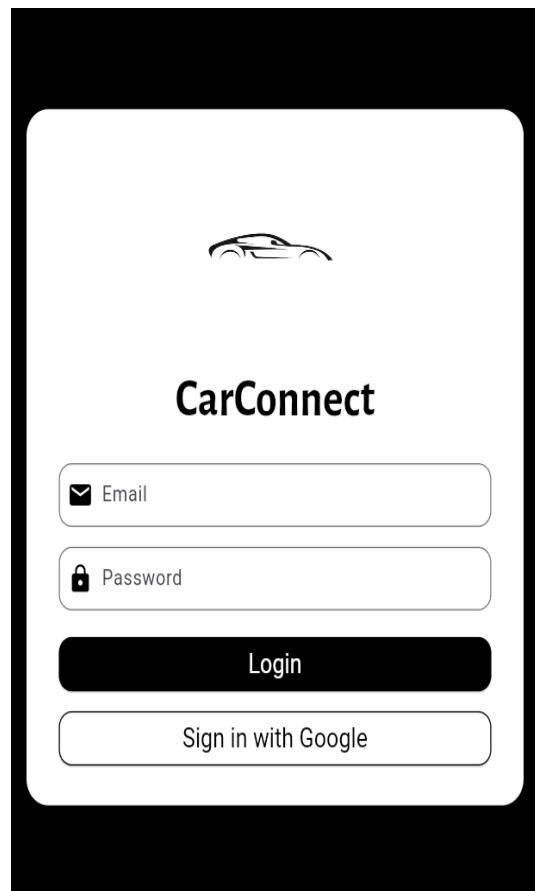
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            debugShowCheckedModeBanner: false,
            initialRoute: '/register',
            routes: {
                '/register': (context) => const RegisterPage(),
                '/login': (context) => const LoginPage(),
                '/home': (context) => const HomePage(), // Add Home Page Route
            },
        );
    }
}
```

## Screen Shots :

1.Register page



2.Login page



## MPL Experiment 5

**Name:** Mohit Patil

**Class:** D15A

**Roll no:**36

**Aim:** To apply navigation, routing and gestures in Flutter App.

### Theory:

#### **Navigation & Routing:**

Flutter provides multiple ways to navigate between screens (pages):

##### **1. Using Navigator.push() and Navigator.pop()**

Push a new screen onto the stack and pop it to return to the previous one.

##### **2. Named Routes**

Define routes in MaterialApp and navigate using Navigator.pushNamed().

##### **3. GoRouter Package**

A declarative approach to handle navigation more efficiently.

#### **Gestures:**

Flutter's GestureDetector and InkWell widgets help in detecting user interactions like taps, swipes, and long presses. Some common gestures include:

- **onTap:** Detects a tap event.
- **onDoubleTap:** Recognizes double taps.
- **onLongPress:** Detects a long press on a widget.
- **onHorizontalDrag & onVerticalDrag:** Detects drag/swipe motions.

#### **Steps:**

**Step 1:** Create a Flutter project and define multiple screens.

**Step 2:** Set up named routes in **MaterialApp**.

**Step 3:** Implement navigation using **Navigator.push()** and **Navigator.pushNamed()**.

**Step 4:** Use **GestureDetector** to detect taps, swipes, and long presses.

**Step 5:** Add buttons or swipes to navigate between screens.

#### **Code:**

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:proj1/screens/login.dart';
import 'package:proj1/screens/RegisterPage.dart';
import 'package:proj1/screens/home.dart';
import 'package:proj1/screens/UsedCarScreen.dart';
```

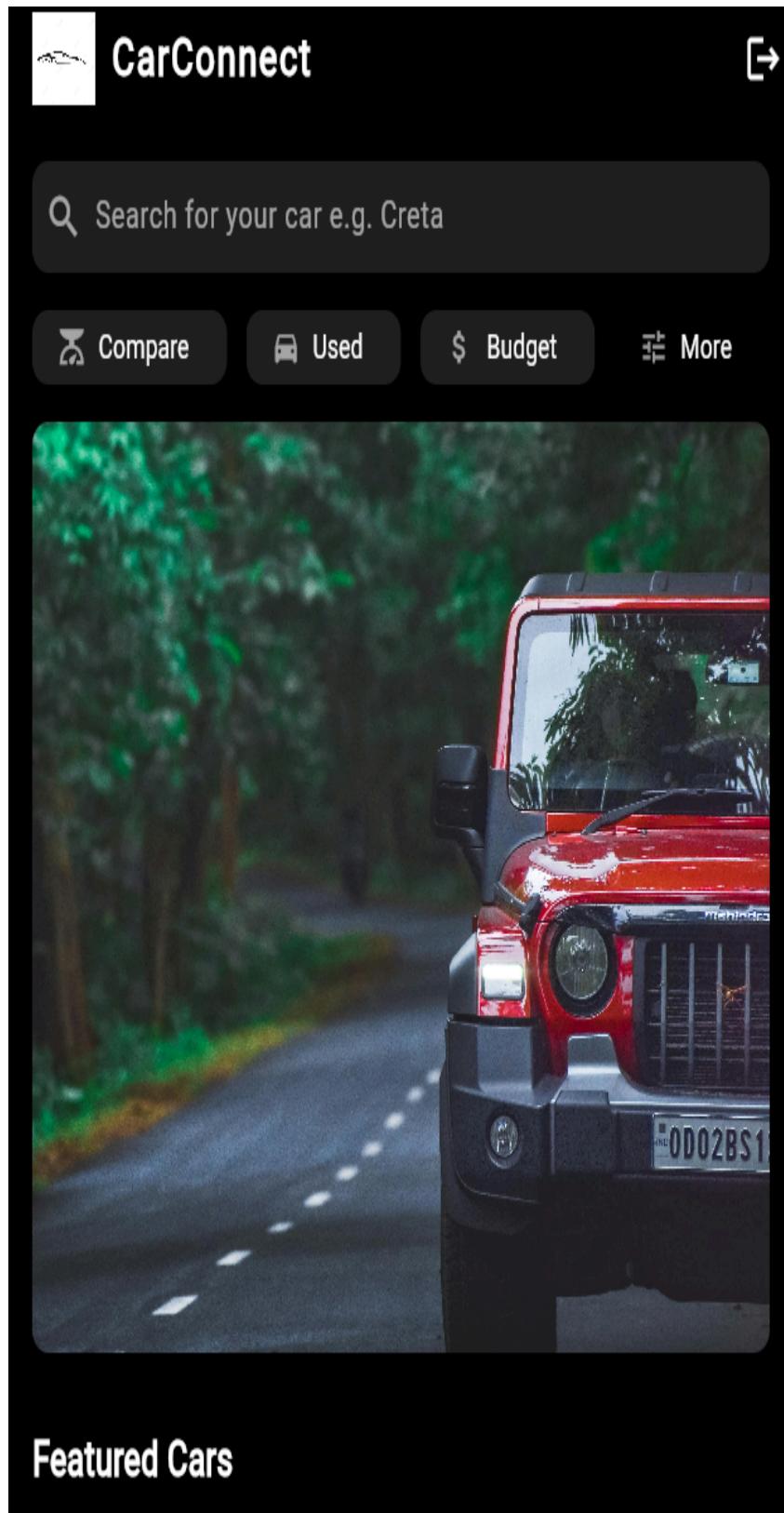
```
import 'package:proj1/screens/FilteredUsedCarsScreen.dart';
import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      initialRoute: '/login', // Default route to Login Page
      routes: {
        '/register': (context) => const RegisterPage(),
        '/login': (context) => const LoginPage(),
        '/home': (context) => const HomePage(),
        '/usedCars': (context) => UsedCarScreen(),
      },
      onGenerateRoute: (settings) {
        if (settings.name == '/filteredCars') {
          final args = settings.arguments as Map<String, int>;
          return MaterialPageRoute(
            builder: (context) => FilteredUsedCarsScreen(
              minPrice: args['minBudget']!,
              maxPrice: args['maxBudget']!,
            ),
          );
        }
        return null;
      },
    );
  }
}
```

**Output:**





## Compare Cars

Select Car

VS Select Car

Compare

### POPULAR CAR COMPARISONS



Tata Harrier  
onwards



Kia Sonet  
onwards

VS



BMW 3 Series  
onwards



Audi Q5  
onwards

VS



## Car Comparison

| Feature          | MG Hector | Mahindra XUV700 |
|------------------|-----------|-----------------|
| Price            | 1500000   | 1400000         |
| Engine           | 1.5L      | 2.2L            |
| Mileage          | 14 kmpl   | 15 kmpl         |
| Horsepower       | 143 HP    | 200 HP          |
| Torque           | 250 Nm    | 450 Nm          |
| Fuel Type        | Petrol    | Diesel          |
| Transmission     | CVT       | Automatic       |
| Seating Capacity | 5         | 7               |



## Find the Right Used Car



Choose your Budget



0 Lakh

50 Lakh

Find Used Car

Want to Sell Your Car?

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

Sell Car Online



**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



**Mercedes-Benz E-Class**



88.0 Lakh

← Car Budget & Loan Calculator

Car Name

car

Ex-Showroom Price (₹)/Your estimated price

1000000

Select Car Type

Mid-Size Car



Interest Rate (%)

10

Loan Duration (Years)

2

Down Payment (₹, Optional)

10000

Get Cost Breakdown

**Breakdown for Your Selected Car (Mid-Size Car)**

**Total On-Road Price (Without Loan) ₹1610500.00**

**Monthly EMI ₹73854.95**

**Total Cost with Loan & Interest ₹1772518.91**

Show Detailed Breakdown



Car Budget &amp; Loan Calculator

## Detailed Cost Breakdown for Your Selected Car (Mid-Size Car)

Ex

1)

Ex-Showroom Price

₹1000000.00

Se

GST

₹280000.00

M

Cess

₹150000.00

In:

1)

Road Tax

₹100000.00

Lc

Insurance

₹50000.00

2

Registration

₹10000.00

Dc

Fastag

₹500.00

1)

Accessories

₹20000.00

**Total On-Road Price****₹1610500.00**

Bre

Loan Amount

₹1600500.00

Tot

Monthly EMI

₹73854.95 .00

Mo

Total Loan Payment

₹1772518.91 .95

Tot

.91



Close

## MPL Experiment 6

**Name:** Mohit Patil

**Class:** D15A

**Roll no:**36

**Aim:** How To Set Up Firebase with Flutter for iOS and Android Apps

### **Theory:** -

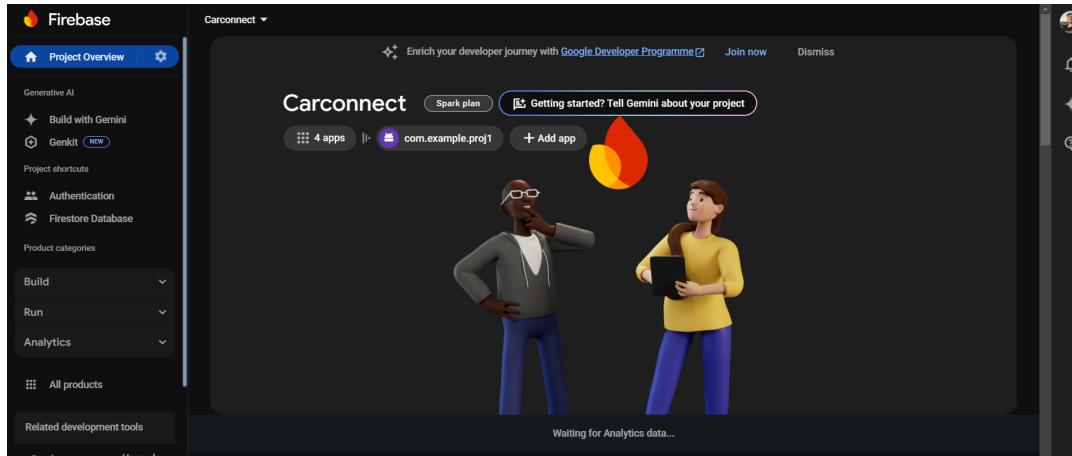
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend solution for Flutter applications.

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

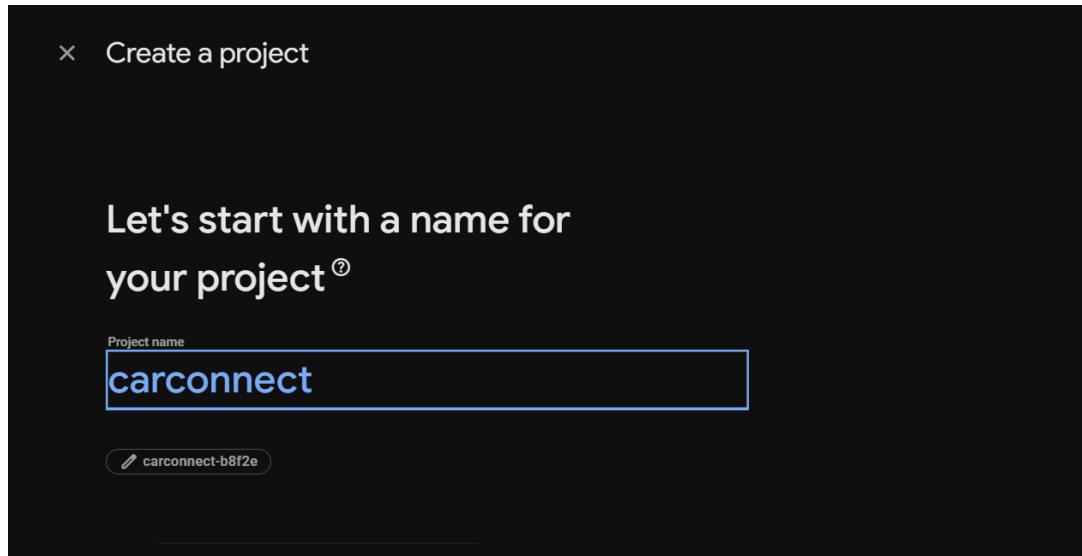
### **Steps to Connect Flutter UI with Firebase Database:**

#### **Step 1:**

- 1.1) Go to Firebase Console and Create a Firebase Project

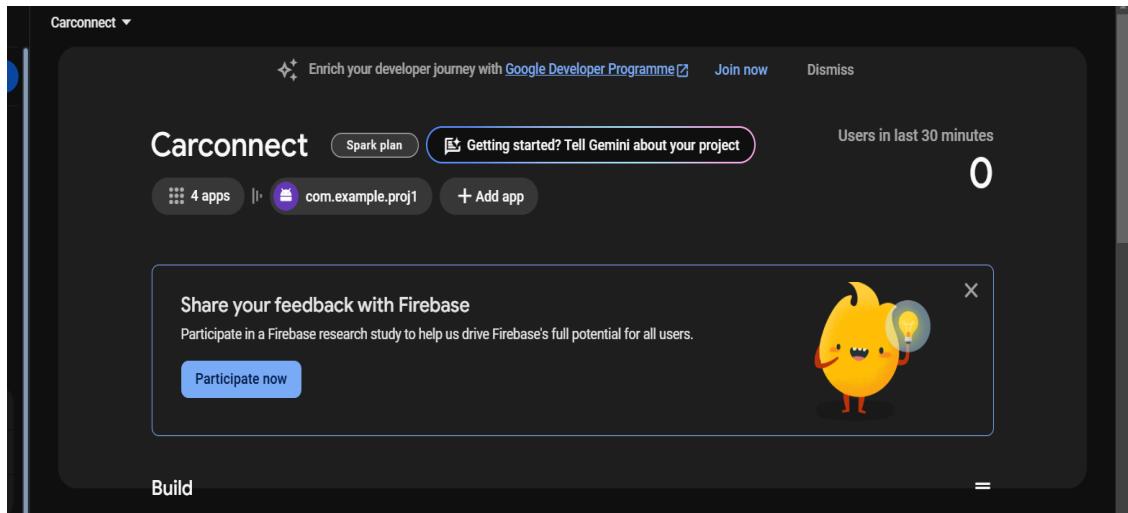


1.2) Click on Create a Project and give it a suitable name.



## Step 2:- Add Firebase to Your Flutter App

2.1) Click on Android/iOS/Web based on your Flutter application

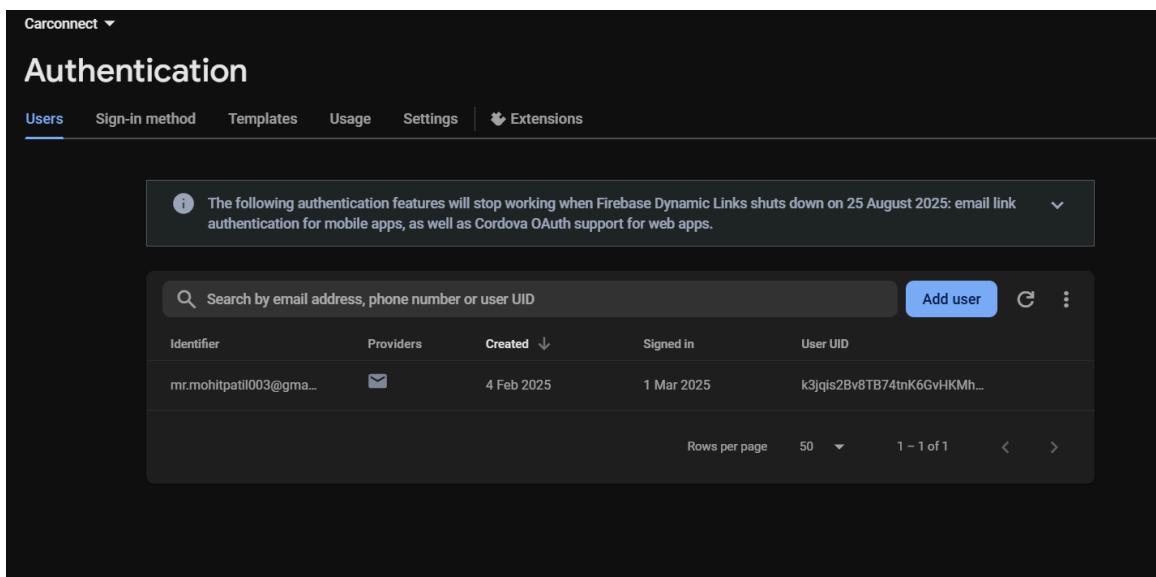


### **Step 3: - Add Firebase Authentication to Your App**

#### 3.1) Add Firebase Authentication Dependencies

```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_core: ^3.11.0  
  firebase_auth: ^5.4.2 # For authentication  
  cloud_firestore: ^5.6.3 # For Firestore, if you need it  
  firebase_messaging: ^15.2.2  
  http: ^0.13.3  
  image_picker: ^1.0.4  
  tflite_flutter: ^0.11.0  
  image: ^3.2.0  
  url_launcher: ^6.1.14
```

- 3.2) Enable Authentication in Firebase Console Go to Firebase Console → Authentication.  
Click on Sign-in method and enable Email/Password (or any other method like Google).  
Click Save



The screenshot shows the Firebase Authentication console for a project named "Carconnect". The "Users" tab is selected. A notification bar at the top states: "The following authentication features will stop working when Firebase Dynamic Links shuts down on 25 August 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps." Below the notification is a search bar and a "Add user" button. A table lists one user entry:

| Identifier              | Providers | Created    | Signed in  | User UID                    |
|-------------------------|-----------|------------|------------|-----------------------------|
| mr.mohitpatil003@gma... | ✉         | 4 Feb 2025 | 1 Mar 2025 | k3jqls2Bv8TB74tnK6GvHKMh... |

At the bottom, there are pagination controls for "Rows per page" (50), "1 - 1 of 1", and navigation arrows.

### 3.3)

Implement Authentication in Flutter Modify main.dart

```
import 'package:firebase_core/firebase_core.dart'; import 'package:firebase_auth/firebase_auth.dart';

void main() async { WidgetsFlutterBinding.ensureInitialized(); await Firebase.initializeApp();
runApp(MyApp());
}
```

### Step 4: -Configure Firebase Realtime Database

1. Go to Firebase Console → Realtime Database.
2. Click Create Database → Choose location → Set rules (for development, set read/write to true).
3. Click Publish.

### CODE:

#### Login page.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';

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

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

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  // Use the renderButton method for Google Sign-In on web
  void _signInWithGoogle() async {
    final GoogleSignIn googleSignIn = GoogleSignIn();

    try {
```

```
GoogleSignInAccount? googleUser = await googleSignIn.signIn();
if (googleUser == null) return;

final GoogleSignInAuthentication googleAuth =
    await googleUser.authentication;

final AuthCredential credential = GoogleAuthProvider.credential(
    accessToken: googleAuth.accessToken,
    idToken: googleAuth.idToken,
);

await _auth.signInWithCredential(credential);
ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Google Sign-In Successful!')),
);
} catch (e) {
print("Google Sign-In Error: $e");
ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Google Sign-In Failed: $e')),
);
}
}

void _signInWithEmailAndPassword() async {
if (_formKey.currentState!.validate()) {
try {
await _auth.signInWithEmailAndPassword(
    email: _emailController.text.trim(),
    password: _passwordController.text.trim(),
);
ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Login Successful!')),
);

// Redirect to Home Page after login
Navigator.pushReplacementNamed(context, '/home');
} catch (e) {
ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Login Failed: $e')),
);
}
}
}

@Override
Widget build(BuildContext context) {
return Scaffold(
```

```
body: Container(
  color: Colors.black,
  child: Center(
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Card(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(20),
        ),
        elevation: 10,
        color: Colors.white,
        child: Padding(
          padding: const EdgeInsets.all(30.0),
          child: Form(
            key: _formKey,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.min,
              crossAxisAlignment: CrossAxisAlignment.stretch,
              children: [
                Column(
                  children: [
                    Image.asset(
                      'assets/c1.png', // Ensure image is in assets folder
                      height: 150,
                    ),
                    const SizedBox(height: 10),
                    Text(
                      'CarConnect',
                      style: GoogleFonts.alegreyaSans(
                        fontSize: 40,
                        fontWeight: FontWeight.bold,
                        color: Colors.black,
                      ),
                    ),
                    const SizedBox(height: 20),
                    TextFormField(
                      controller: _emailController,
                      decoration: InputDecoration(
                        labelText: 'Email',
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(12),
                        ),
                        prefixIcon:
                          const Icon(Icons.email, color: Colors.black),
                    ),

```

```
        validator: (value) {
          if (value == null || value.isEmpty || !value.contains('@')) {
            return 'Enter a valid email';
          }
          return null;
        },
      ),
      const SizedBox(height: 15),
      TextFormField(
        controller: _passwordController,
        obscureText: true,
        decoration: InputDecoration(
          labelText: 'Password',
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
          ),
          prefixIcon:
            const Icon(Icons.lock, color: Colors.black),
        ),
        validator: (value) {
          if (value == null || value.length < 6) {
            return 'Password must be at least 6 characters';
          }
          return null;
        },
      ),
      const SizedBox(height: 20),
      ElevatedButton(
        style: ElevatedButton.styleFrom(
          padding: const EdgeInsets.symmetric(vertical: 14),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12),
          ),
          backgroundColor: Colors.black,
        ),
        onPressed: _signInWithEmailAndPassword,
        child: const Text(
          'Login',
          style: TextStyle(fontSize: 20, color: Colors.white),
        ),
      ),
      const SizedBox(height: 15),
      ElevatedButton(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.white,
```

The screenshot shows the Cloud Firestore interface. At the top, there's a navigation bar with 'Cloud Firestore' (highlighted in blue), 'Add database', 'Ask Gemini how to get started with Firestore', and tabs for 'Data', 'Rules', 'Indexes', 'Disaster recovery (NEW)', 'Usage', and 'Extensions'. Below the navigation bar, a banner says 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing' with a shield icon, and 'Configure App Check'. On the left, a sidebar shows a tree structure with 'Cars' selected. The main area displays a table for the 'Cars' collection. The first row shows a document with the ID '5i296ic2qZEuCgvNSAcy'. This document contains fields: engine ('2.2L'), fuel ('Diesel'), horsepower ('200 HP'), mileage ('15 kmpl'), name ('Mahindra XUV700'), price (1400000), and seating ('7'). There are also 'Start collection' and 'Add field' buttons.

|  (default) |  Cars |  EpgfF1QBJJ0iSU0rSnq0 |
|---|--|--|
| + Start collection  | + Add document   | + Start collection   |
| Cars >  | EpgfF1QBJJ0iSU0rSnq0 >   | + Add field  |
|   | Ik57aNmuX5phxxmI18ZS   | Engine: "2.8L Diesel"  |
|   | K2ROKDNGs2VAF5tBa8qC   | Fuel Type: "Diesel"  |
|   | MolnicipB8upxqjmc8tz   | Horsepower: "201 hp"   |
|   | NArBeFn7MsKGRJhCljqm   | Mileage: "14 kmpl"   |
|   | NwNXQH4uK8e3JjQZ85oV   | Name: "Toyota Fortuner"  |
|   | RUDCV3Q04Rk1VkiFgX66   | Price: 3500000   |
|   | UVAgJTUDlym1clvuget  | Seating: "7"   |
|   | UdRHGFygTOfbuMsNQ4Vm   | Torque: "500 Nm"   |
|   | ZNRwgCJ0Ux8rH8MZfDXF   | Transmission: "Automatic"  |
|   | jEKe7pK1EVHimK045b1d   |  |
|   | 1Pu\$63gxa6PxwQis2kJo  |  |
|   | oy9KBxrfD5nr6zwBbCS9   |  |
|   | yenTSarfMYsMBWYtqRtx   |  |