

**EXPERIMENT NO: - 03****Name:-** Chirag Choudhary**Class:-** D15A**Roll:No: -** 10**AIM: -** To include icons, images, fonts in Flutter app.**Theory: -**

Flutter is a versatile open-source UI framework , which allows developers to build natively compiled applications for mobile, web, and desktop platforms from a single codebase. One of the key strengths of Flutter is its flexibility in creating highly customizable UIs. This practical focuses on incorporating essential visual elements—icons, images, and custom fonts—into a Flutter application. These elements enhance the visual appeal and usability of the app, providing an engaging experience for users.

A flutter app when built has both assets (resources) and code. Assets are available and deployed during runtime. The asset is a file that can include static data, configuration files, icons, and images. The Flutter app supports many image formats, such as JPEG, WebP, PNG, GIF, animated WebP/GIF, BMP, and WBMP.

Visual elements play a significant role in app development.

- **Enhanced User Experience:** Images and icons make your app visually appealing and user-friendly.
- **Information Conveyance:** They convey information quickly and intuitively. A well-chosen icon can replace lengthy text.
- **Branding:** Custom icons and images reinforce your app's branding, making it memorable.

➤ **Adding Icons in Flutter**

Flutter provides built-in material design icons through the Icons class. Custom icons can also be added using third-party packages such as flutter\_launcher\_icons and font\_awesome\_flutter.

```
Icon(
  Icons.home,
  size: 40,
);
```

## ➤ Adding Images in Flutter

Flutter supports images from three sources:

### 1. **Assets** (Stored locally in the project)

- Place the image inside the assets/images folder in the project.
- Declare the image in pubspec.yaml

```
flutter:
  assets:
    - assets/images/sample.png
```

- Display the image in the app

```
Image.asset('assets/images/sample.png');
```

### 2. **Network** (Fetched from the internet)

Displaying images from the internet or network is very simple. Flutter provides a built-in method `Image.network` to work with images from a URL. The `Image.network` method also allows you to use some optional properties, such as height, width, color, fit, and many more.

```
Image.network('https://example.com/sample.jpg');
```

### 3. **Memory or File** (Stored on the device)

## ➤ Adding Custom Fonts in Flutter

By default, Flutter uses the Roboto font, but custom fonts can be added for a unique UI.

- Download the font and place it in the assets/fonts/ folder.
  - Declare the font in pubspec.yaml
  - Use the font in the app
- ```
Text(
  'Custom Font Example',
  style: TextStyle(fontFamily: 'CustomFont', fontSize: 24),
);
```

Code: -

### **expense\_item.dart**

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

class ExpenseItem extends StatelessWidget {
  final String category;
  final String description;
  final double amount;

  const ExpenseItem({
    Key? key,
    required this.category,
    required this.description,
    required this.amount,
  }) : super(key: key);
```

```
@override
Widget build(BuildContext context) {
  return Card(
    margin: const
    EdgeInsets.symmetric(horizontal: 16, vertical: 8),
    child: ListTile(
      title: Text(category),
      subtitle: Text(description),
      trailing:
      Text('₹${amount.toStringAsFixed(2)}'),
    ),
  );
}
```

### **custom\_button.dart**

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

class CustomButton extends StatelessWidget {
  final String label;
  final VoidCallback onPressed;

  CustomButton({required this.label, required
  this.onPressed});
```

```
@override
Widget build(BuildContext context) {
  return ElevatedButton(
    onPressed: onPressed,
    child: Text(label),
  );
}
```

### **app\_theme.dart**

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

class AppTheme {
  static ThemeData get lightTheme {
    return ThemeData(
      primarySwatch: Colors.blue,
      appBarTheme: AppBarTheme(
        backgroundColor: Colors.blue,
        elevation: 4,
```

```
),
    buttonTheme: ButtonThemeData(
      buttonColor: Colors.blue,
      textTheme: ButtonTextTheme.primary,
    ),
  );
}
}
```

### **pubspec.yaml**

```
name: expense_tracker
description: A simple expense tracker app with authentication.
publish_to: 'none' # Prevents accidental publishing

environment:
  sdk: '>=3.0.0 <4.0.0' # Update according to your Flutter SDK
```

dependencies:

flutter:

  sdk: flutter

*# firebase\_core: ^2.24.2*

*# firebase\_auth: ^4.15.3*

*# cloud\_firestore: ^4.15.3*

dev\_dependencies:

flutter\_test:

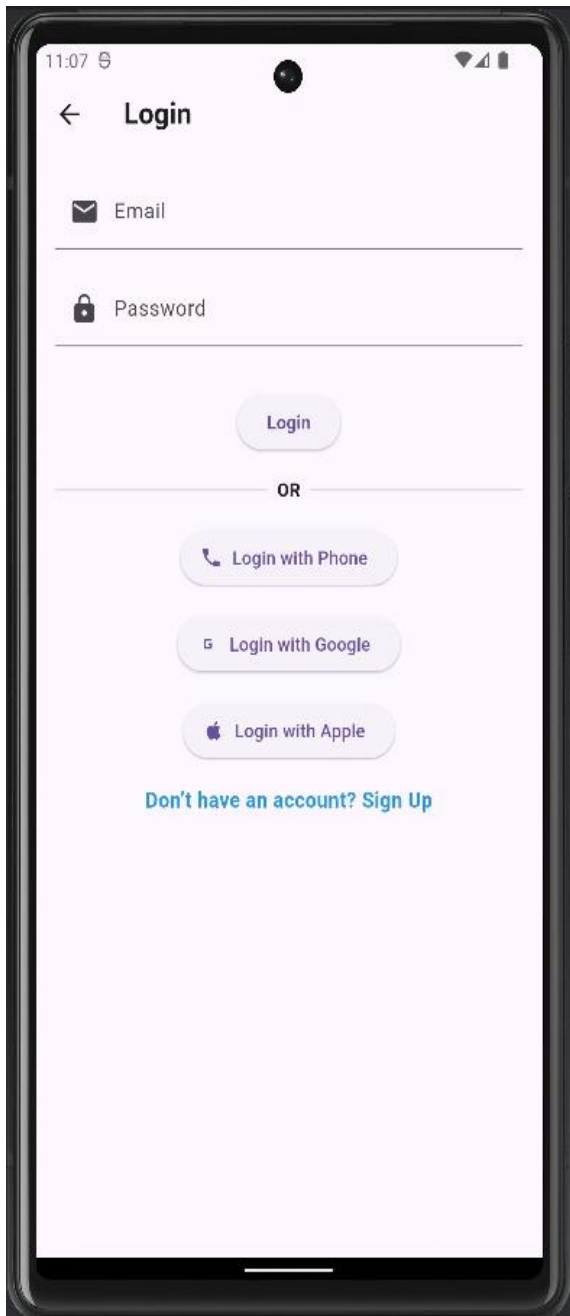
  sdk: flutter

flutter:

  uses-material-design: true

assets:

  - assets/images/finance.png

**OUTPUT: -**

11:07

← Login

Email

Password

Login

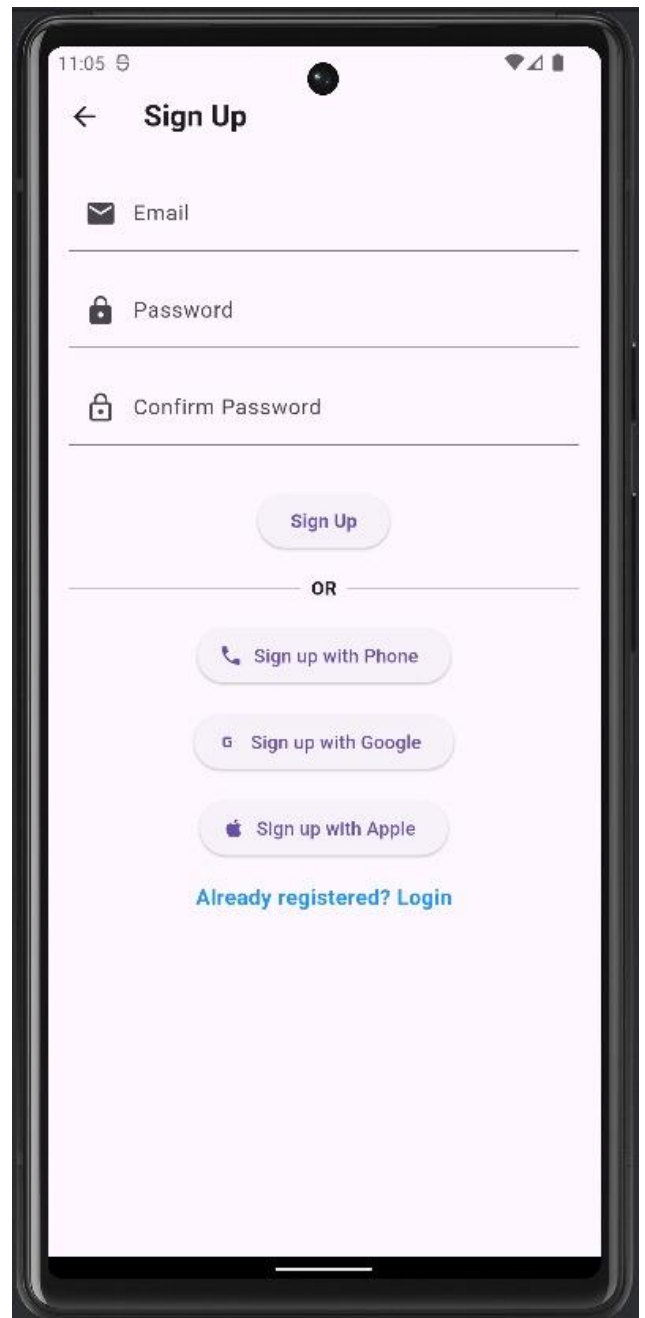
OR

Login with Phone

Login with Google

Login with Apple

[Don't have an account? Sign Up](#)



11:05

← Sign Up

Email

Password

Confirm Password

Sign Up

OR

Sign up with Phone

Sign up with Google

Sign up with Apple

[Already registered? Login](#)