Name: Mohd Sufiyan Siddiqui
Class : D15A
Roll NO : 55

Lab - 02

Theory :

Flutter: A Comprehensive Overview

Introduction:

Flutter is an open-source UI software development toolkit created by Google, known for its ability to build natively compiled applications for mobile, web, and desktop from a single codebase. Launched in 2017, Flutter has gained widespread popularity for its expressive and flexible framework, allowing developers to create visually appealing and performant applications.

Key Features:

1. Single Codebase, Multiple Platforms:
   Flutter allows developers to write code once and deploy it on various platforms, such as iOS, Android, web, and desktop. This not only saves development time but also ensures consistency across different devices.

2. Widget-based Architecture:
   Flutter's core building blocks are widgets, which are lightweight and reusable elements used to construct the user interface. Widgets can be combined to create complex UIs, and Flutter provides an extensive set of both Material Design and Cupertino widgets for a native look and feel on Android and iOS.

3. Hot Reload:
   One of Flutter's standout features is the Hot Reload functionality. Developers can see the impact of code changes in real-time without restarting the entire application. This speeds up the development process, making it more interactive and efficient.

4. Rich Set of Libraries:
   Flutter comes with a rich set of pre-designed material components and Cupertino widgets. Additionally, developers can use packages from the pub.dev repository to extend the functionality of their applications, covering a wide range of features such as authentication, databases, and more.

5. Dart Programming Language:
   Flutter uses Dart as its programming language. Dart is a modern, object-oriented language that provides features like a strong static type system and just-in-time (JIT) compilation. Its syntax is easy to learn for developers familiar with languages like Java or JavaScript.

Architecture:

Flutter follows a layered architecture comprising the following components:

1. Framework Layer:
   The framework layer contains the foundation of Flutter, providing the basic building blocks such as widgets, rendering, and animation. The framework is responsible for managing the flow of data and handling user input.

2. Engine Layer:
   The engine layer consists of C/C++ code responsible for rendering, gestures, and other low-level tasks. It interacts with the platform-specific code and communicates with the framework layer using a platform channel.

3. Embedder Layer:
   The embedder layer integrates Flutter into the respective platform (iOS, Android, web, desktop). It handles the initialization of Flutter, communication with the engine layer, and exposes platform-specific services to the framework.

Advantages:

1. Productivity:
   The ability to write code once and deploy it on multiple platforms significantly enhances developer productivity. Hot Reload ensures quick iterations and reduces development time.

2. Expressive UI:
   Flutter's widget-based architecture allows for the creation of highly customizable and visually appealing user interfaces. Designers and developers can collaborate seamlessly to bring designs to life.

3. Performance:
   Flutter's compiled code performs at native speed, providing a smooth and responsive user experience. The absence of a bridge between the application and the platform contributes to excellent performance.

4. Community and Ecosystem:

Flutter has a vibrant and growing community, contributing to a vast ecosystem of plugins and packages. This makes it easy for developers to find solutions to common problems and integrate various functionalities into their applications.

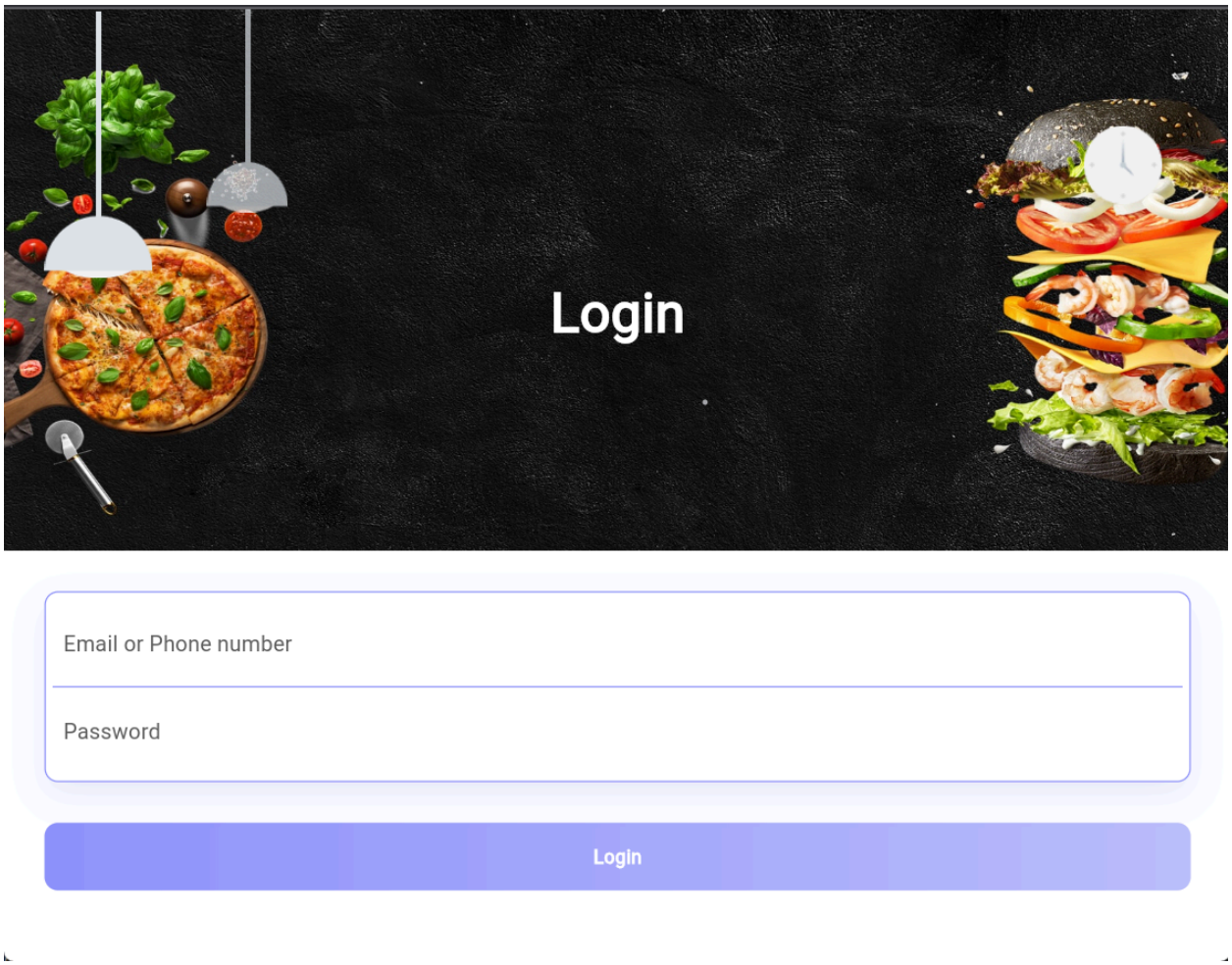
Challenges and Considerations:

1. Learning Curve:
While Dart is easy to pick up for developers with existing programming experience, there might be a learning curve for those unfamiliar with the language.

2. Size of the App:
Flutter apps may have a larger file size compared to native apps due to the inclusion of the Flutter engine. However, advancements are being made to address this concern.

3. Platform-specific Integration:
While Flutter provides a unified codebase, there might be cases where platform-specific features or integrations require additional effort.

OUTPUT:



Code :

```
import 'package:flutter/material.dart';
import 'package:animate_do/animate_do.dart';
void main() => runApp(
 MaterialApp(
   debugShowCheckedModeBanner: false,
   home: HomePage(),
 )
);

class HomePage extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     backgroundColor: Colors.white,
     body: SingleChildScrollView(
       child: Container(
```

```
child: Column(
  children: <Widget>[
    Container(
      height: 400,
      decoration: BoxDecoration(
        image: DecorationImage(
          image: AssetImage('assets/images/b6.jpg'),
          fit: BoxFit.fill
        )
      ),
      child: Stack(
        children: <Widget>[
          Positioned(
            left: 30,
            width: 80,
            height: 200,
            child: FadeInUp(duration: Duration(seconds: 1), child: Container(
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: AssetImage('assets/images/light-1.png')
                )
              ),
            )),
          ),
          Positioned(
            left: 140,
            width: 80,
            height: 150,
            child: FadeInUp(duration: Duration(milliseconds: 1200), child: Container(
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: AssetImage('assets/images/light-2.png')
                )
              ),
            )),
          ),
          Positioned(
            right: 40,
            top: 40,
            width: 80,
            height: 150,
            child: FadeInUp(duration: Duration(milliseconds: 1300), child: Container(
              decoration: BoxDecoration(
                image: DecorationImage(
```

```
                    image: AssetImage('assets/images/clock.png')
                  )
                ),
              )),
            ),
            Positioned(
              child: FadeInUp(duration: Duration(milliseconds: 1600), child: Container(
                margin: EdgeInsets.only(top: 50),
                child: Center(
                  child: Text("Login", style: TextStyle(color: Colors.white, fontSize: 40, fontWeight:
FontWeight.bold),),
                ),
              )),
            )
          ],
        ),
      ),
      Padding(
        padding: EdgeInsets.all(30.0),
        child: Column(
          children: <Widget>[
            FadeInUp(duration: Duration(milliseconds: 1800), child: Container(
              padding: EdgeInsets.all(5),
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(10),
                border: Border.all(color: Color.fromRGBO(143, 148, 251, 1)),
                boxShadow: [
                  BoxShadow(
                    color: Color.fromRGBO(143, 148, 251, .2),
                    blurRadius: 20.0,
                    offset: Offset(0, 10)
                  )
                ]
              ),
              child: Column(
                children: <Widget>[
                  Container(
                    padding: EdgeInsets.all(8.0),
                    decoration: BoxDecoration(
                      border: Border(bottom: BorderSide(color:  Color.fromRGBO(143, 148, 251,
1)))
                    ),
                    child: TextField(
```

```
                    decoration: InputDecoration(
                      border: InputBorder.none,
                      hintText: "Email or Phone number",
                      hintStyle: TextStyle(color: Colors.grey[700])
                    ),
                  ),
                ),
                Container(
                  padding: EdgeInsets.all(8.0),
                  child: TextField(
                    obscureText: true,
                    decoration: InputDecoration(
                      border: InputBorder.none,
                      hintText: "Password",
                      hintStyle: TextStyle(color: Colors.grey[700])
                    ),
                  ),
                )
              ],
            ),
          )),
          SizedBox(height: 30,),
          FadeInUp(duration: Duration(milliseconds: 1900), child: Container(
            height: 50,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(10),
              gradient: LinearGradient(
                colors: [
                  Color.fromRGBO(143, 148, 251, 1),
                  Color.fromRGBO(143, 148, 251, .6),
                ]
              )
            ),
            child: Center(
              child: Text("Login", style: TextStyle(color: Colors.white, fontWeight:
FontWeight.bold),),
            ),
          )),
          SizedBox(height: 70,),
          FadeInUp(duration: Duration(milliseconds: 2000), child: Text("Forgot Password?",
style: TextStyle(color: Color.fromRGBO(143, 148, 251, 1)),)),
        ],
      ),
    )
```

```
          ],
        ),
      ),
    )
  );
 }
}
```

Conclusion:

Flutter has emerged as a powerful framework for cross-platform development, providing a seamless development experience, expressive UI, and excellent performance. Its growing community and continuous updates from Google contribute to its ongoing success. As the technology landscape evolves, Flutter is likely to play an increasingly significant role in shaping the future of multi-platform application development.