# Experiment 3

**Name: Soham Hajare** **D15B/19**

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

**Introduction:**

In a Flutter app, incorporating icons, images, and fonts is essential for enhancing visual appeal and user engagement. Icons serve as intuitive visual cues, aiding in navigation and providing a polished interface. Flutter's extensive library of customizable icons allows developers to seamlessly integrate them into the app's design. Additionally, the inclusion of images not only adds vibrancy but also conveys information effectively. Flutter's Image widget facilitates the display of various image formats, enabling developers to showcase graphics, logos, or dynamic content effortlessly. Furthermore, the choice of fonts contributes significantly to the app's aesthetic. Flutter supports diverse font styles and allows developers to import custom fonts, empowering them to align the app's typography with the brand identity. By leveraging these Flutter features, developers can craft visually appealing and cohesive user interfaces, elevating the overall user experience.

**Code:**

**Main.dart**

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primaryColor: Colors.indigo,
        hintColor: Colors.amber,
        fontFamily: 'Montserrat',
```

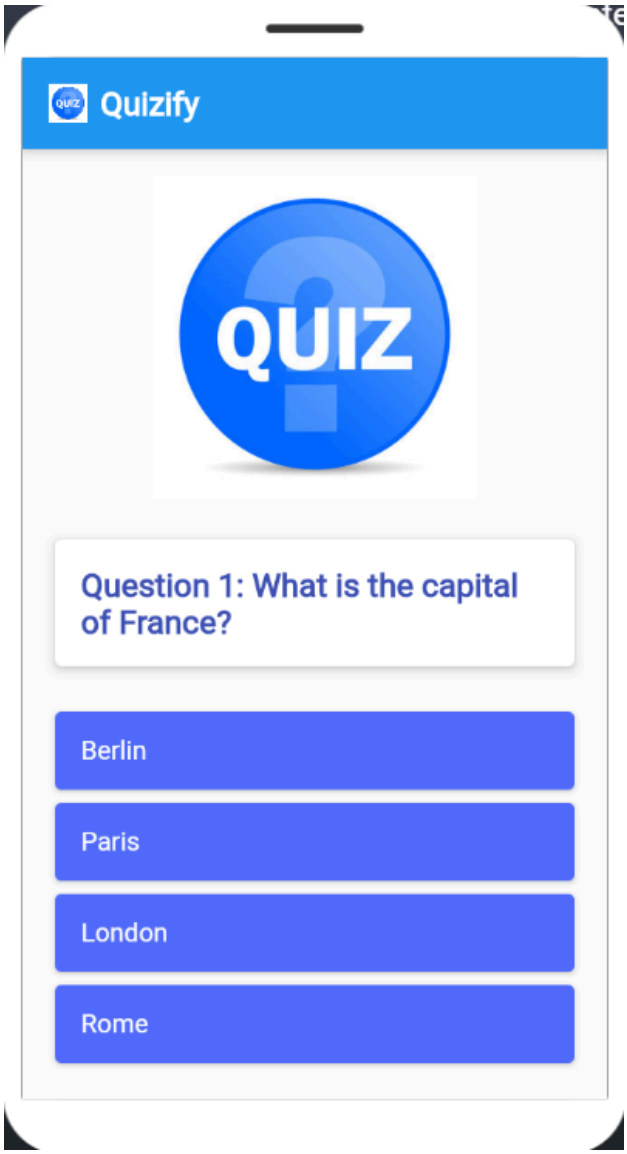
```
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Row(
            children: [
              Image.network(

'https://t4.ftcdn.net/jpg/00/70/36/57/360_F_70365750_D7i4pU3448fkNEmcWT1SAp4MKsSHEBgH.jpg', // Replace with your online icon URL
                height: 24,
                width: 24,
              ),
              SizedBox(width: 8),
              Text('Quizify', style: TextStyle(fontWeight: FontWeight.bold)),
            ],
          ),
        ),
        body: QuizPage(),
      ),
    );
  }
}

class QuizPage extends StatefulWidget {
  @override
  _QuizPageState createState() => _QuizPageState();
}

class _QuizPageState extends State<QuizPage> {
  List<Map<String, dynamic>> quizData = [
    {
      'question': 'What is the capital of France?',
      'options': ['Berlin', 'Paris', 'London', 'Rome'],
      'correctAnswer': 'Paris',
    },
    {
```

```
    'question': 'Which planet is known as the Red Planet?',
    'options': ['Venus', 'Mars', 'Jupiter', 'Saturn'],
    'correctAnswer': 'Mars',
  },
  // Add more questions as needed
];

int currentQuestionIndex = 0;

@override
Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: [
        Image.network(

'https://t4.ftcdn.net/jpg/00/70/36/57/360_F_70365750_D7i4pU3448fkNEmcWT1SAp4MKsSHEBgH.jpg',
          height: 200, // Adjust the height of the image as needed
        ),
        SizedBox(height: 20),
        Card(
          elevation: 4,
          color: Colors.white,
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Text(
              'Question ${currentQuestionIndex + 1}:
${quizData[currentQuestionIndex]['question']}',
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold, color:
Colors.indigo),
            ),
          ),
        ),
```

```
        SizedBox(height: 20),
        Column(
          children: List.generate(
            quizData[currentQuestionIndex]['options'].length,
            (index) => Card(
              elevation: 2,
              color: Colors.indigoAccent,
              child: ListTile(
                title: Text(
                  quizData[currentQuestionIndex]['options'][index],
                  style: TextStyle(color: Colors.white),
                ),
                onTap: () {
                  // Add your logic to handle option selection and move to the next question
                  print('Selected: ${quizData[currentQuestionIndex]['options'][index]}');
                  moveToNextQuestion();
                },
              ),
            ),
          ),
        ),
      ],
    ),
  );
}

void moveToNextQuestion() {
  if (currentQuestionIndex < quizData.length - 1) {
    setState(() {
      currentQuestionIndex++;
    });
  } else {
    // Display quiz completion or navigate to the result screen
    print('Quiz completed!');
    // Add your logic for quiz completion
  }
```

```
  }
}
```

**Output:**


Quizify
QUIZ
Question 1: What is the capital of France?
Berlin
Paris
London
Rome

**Explanation:**

Main Function:

- The main function is the starting point of the Flutter application.
- It calls the runApp function, passing an instance of the MyApp widget.

MyApp Class (StatelessWidget):

- MyApp is a stateless widget, indicating that it doesn't have mutable state.
- It returns a MaterialApp widget, which serves as the root of the Flutter app.
- MaterialApp:
    - Defines the overall visual theme of the app using the theme property.
    - Sets the primary color to indigo, hint color to amber, and uses the 'Montserrat' font family.
    - Specifies the home screen as a Scaffold widget.
- Scaffold:
    - Represents the basic structure of the app, including an app bar and body content.
- AppBar:
    - Displays a navigation bar at the top of the app.
    - Contains a title composed of a row with an online image and the text 'Quizify'.

QuizPage Class (StatefulWidget):

- QuizPage is a stateful widget, allowing for mutable state.
- _QuizPageState Class:
    - Represents the mutable state of the QuizPage.
- quizData List:
    - Holds a list of maps, each representing a quiz question with options and correct answers.
- currentQuestionIndex:
    - Keeps track of the index of the current question being displayed.
- build Method:
    - Constructs the UI for the QuizPage.
    - Displays an online image above the question, followed by a card showing the current quiz question.
    - Options for the question are presented using ListTile widgets within a Column.
    - User interaction triggers the onTap callback, invoking the moveToNextQuestion method.
- moveToNextQuestion Method:

- Handles logic for moving to the next question.
- Updates the currentQuestionIndex using setState to trigger a rebuild.
- If there are no more questions, a completion message is printed.

**Conclusion:**

The provided Flutter code effectively demonstrates the integration of icons, images, and fonts within a mobile application. The MyApp class employs a MaterialApp widget to define the app's theme, incorporating an online image as an icon in the app bar. The visual design is enhanced with the use of online images displayed above quiz questions in the QuizPage class. Additionally, the specified 'Montserrat' font family is applied globally throughout the app, contributing to a consistent and visually appealing user interface.