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

Theory:

Images:

Images in Flutter serve to visually communicate information, enhance aesthetics, and create emotional connections with users. The theory behind effectively using images involves considerations such as visual hierarchy, emotional impact, accessibility, and performance optimization.

Fonts:

Fonts play a vital role in shaping the visual identity and readability of text-based content in Flutter apps. Key considerations for font usage include typography principles, branding and consistency, legibility and readability, as well as localization and accessibility.

Icons:

In Flutter, icons are vector graphics used to represent actions, categories, or entities within your app's user interface. Flutter provides a comprehensive collection of built-in Material Design icons and Cupertino icons for iOS-style design. These icons are readily available for use and cover a wide range of common UI elements, such as navigation buttons, settings, and alerts.

To incorporate icons into your Flutter app, you can use the Icon widget and specify the desired icon from the available icon sets. This makes it convenient to maintain a consistent visual language across your app's interface, regardless of the platform or design style.

Here’s a list of Material Design Icons:

Icons.favorite - Favorite
Icons.home - Home
Icons.settings - Settings
Icons.search - Search
Icons.notifications - Notifications
Icons.person - Person
Icons.email - Email
Icons.phone - Phone
Icons.camera - Camera
Icons.attach_file - Attach File

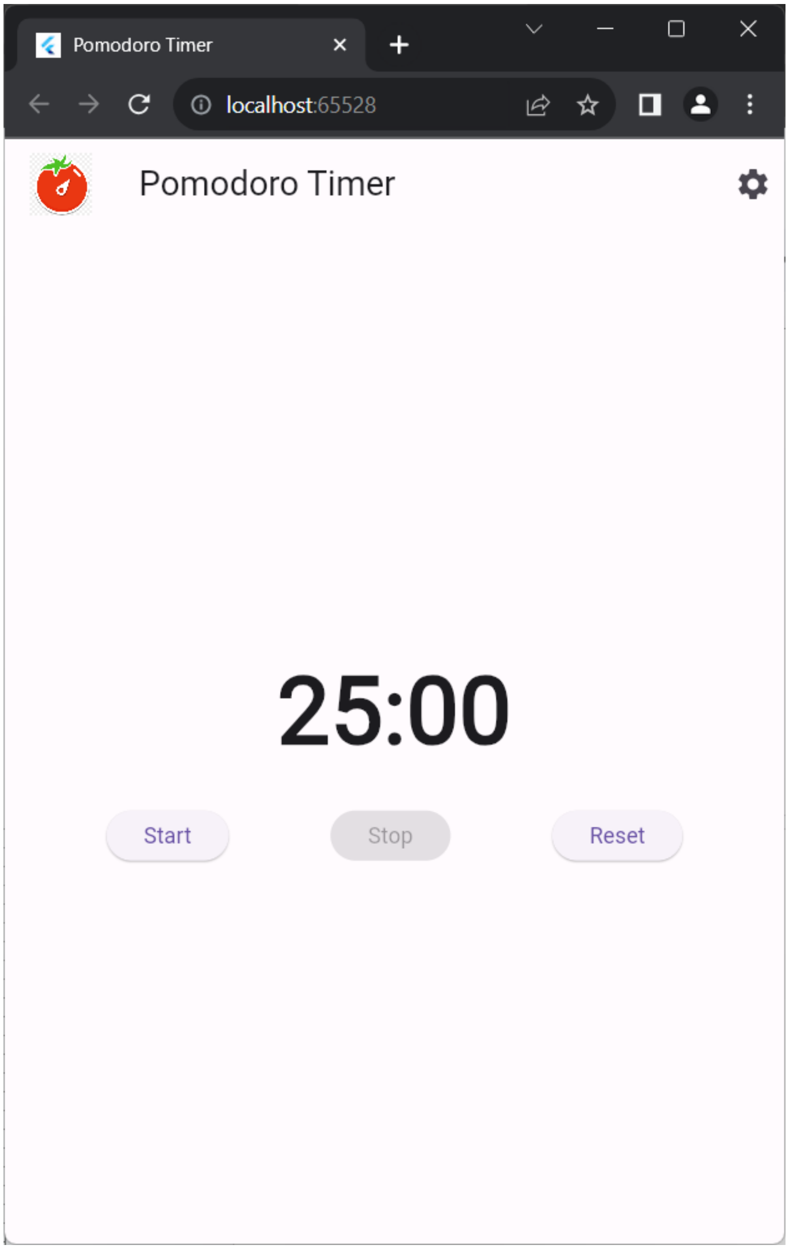
Code:

main.dart:

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

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

class PomodoroApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Pomodoro Timer',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: PomodoroTimer(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class PomodoroTimer extends StatefulWidget {
  @override
  _PomodoroTimerState createState() => _PomodoroTimerState();
}

class _PomodoroTimerState extends State<PomodoroTimer> {
  int _minutes = 25;
  int _seconds = 0;
  late Timer _timer;
  bool _isRunning = false;

  @override
  void dispose() {
    _timer.cancel(); // Cancel the timer when the state is
```

```
disposed
    super.dispose();
  }

  void _startTimer() {
    setState(() {
      _isRunning = true;
    });
    _timer = Timer.periodic(Duration(seconds: 1), (timer) {
      if (_minutes == 0 && _seconds == 0) {
        _resetTimer();
        return;
      }
      if (_seconds == 0) {
        setState(() {
          _minutes--;
          _seconds = 59;
        });
      } else {
        setState(() {
          _seconds--;
        });
      }
    });
  }

  void _stopTimer() {
    _timer.cancel();
    setState(() {
      _isRunning = false;
    });
  }

  void _resetTimer() {
    _timer.cancel();
    setState(() {
      _isRunning = false;
      _minutes = 25;
      _seconds = 0;
    });
  }

  String _formatTime(int time) {
    return time < 10 ? '0$time' : '$time';
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Row(
          children: [
            Image.asset(
              'assets/images/pomodoro_logo.png',
              width: 40,
              height: 40,
            ),
            SizedBox(width: 30),
            Text('Pomodoro Timer'),
          ],
        ),
        actions: [
          IconButton(
            icon: Icon(Icons.settings),
            onPressed: () {
              // Add settings action
            },
          ),
        ],
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            SizedBox(height: 20),
            Text(

'${_formatTime(_minutes)}:${_formatTime(_seconds)}',
              style: TextStyle(fontFamily: 'Roboto',
                  fontWeight: FontWeight.bold,
                  fontSize: 60),
            ),
            SizedBox(height: 20),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: <Widget>[
                ElevatedButton(
                  onPressed: _isRunning ? null : _startTimer,
                  child: Text('Start'),
                ),
                ElevatedButton(
                  onPressed: _isRunning ? _stopTimer : null,
                  child: Text('Stop'),
                ),
                ElevatedButton(
                  onPressed: _resetTimer,
                  child: Text('Reset'),
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:



The settings icon from the Material Design Icons is used here. Also the Roboto font's .ttf file is used for this application. Also the logo to this application to the extreme left is an image that we have used.

Conclusion:

Thereby we have successfully implemented fonts, images and icons in the application. This enhances the overall experience of the application.