AIM:

Due Today
Lab-1:

Practical Task - Build a simple Hello World app in Flutter.

Topics:
Widget Hierarchy - Root Widget, Child Widgets, Parent-Child Relationships
Composition - Combining Widgets, Nested Layouts
Widget Types - Stateless Widgets, Immutable Properties, Rendering Pipeline

Learning Resources -
https://www.youtube.com/watch?app=desktop&v=gOZrczjHF6g
https://docs.flutter.dev/tools/devtools/inspector
https://medium.com/flutter-community/flutter-widgets-row-column-flex-the-whole-picture-a648cd6e6904
https://www.youtube.com/watch?v=g2E7yl3MwMk
https://www.youtube.com/watch?app=desktop&v=gOZrczjHF6g
https://medium.com/@bosctechlabs/building-dynamic-layouts-with-flutter-flex-da5576180882

Lab-2:
Practical Task -

- **Create an App with list, grids and Scrolling for To-do list App.**

THEORY:

About Flutter

- **Why Flutter?**
  - High performance: Compiles to native code (Dart) for speed.
  - Declarative UI: Inspired by React, widgets are rebuilt only when necessary.
  - Integrated Hot-Reload: Quickly preview changes during development.
- **Advantages**:
  - Simplifies app development compared to web and native apps.
  - Promises smaller codebases and faster development.

Flutter Project Structure

- **Key folders**:
  - lib/: Main codebase.
  - pubspec.yml: Dependency manager (like package.json).
  - ios/ & android/: Platform-specific code.
  - test/: Testing files.
- **Entry Point**: main.dart contains the main() function, the starting point of every Flutter

app.

Understanding Widgets

- **Core Concept**: Everything in Flutter is a widget.
  - Widgets handle UI, themes, and even state management.
- **Basic Widgets**:
  - Text: Displays text. Requires textDirection.
  - Center: Aligns its child widget to the center.
  - Container: Equivalent to HTML's , used for layout and styling.
- **Composition**: Widgets are nested to build complex interfaces.

Code Examples

1. **"Hello, World!" Basic App**:

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

main() => runApp(
  Text(
    'Hello, World!!!',
    textDirection: TextDirection.ltr,
  ),
);
```

2. **Centering Content**:

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

main() => runApp(
  Center(
    child: Text(
      'Hello, World!',
      textDirection: TextDirection.ltr,
    ),
  ),
);
```

3. **Adding Styles**:

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

main() => runApp(
  Directionality(
    textDirection: TextDirection.ltr,
    child: Container(
      color: Color(0xFF444444),
      child: Center(
        child: Text(
          'Hello, World!',
          style: TextStyle(
```

```
          color: Color(0xFFFD620A),
          fontSize: 32.0,
        ),
      ),
    ),
  ),
 ),
);
```

Design Philosophy

- **Declarative Composition**: Combines widgets hierarchically.
- **Reusability**: Widgets can be reused to streamline development.
- **Directionality**: Set global text direction using Directionality widget.

CODE:

Lab -1

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text("Flutter Grid Example"),
        ),
        body: GridViewExample(),
      ),
    );
  }
}

class GridViewExample extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
      return GridView.builder(
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 12, // Number of columns
        crossAxisSpacing: 1.0, // Spacing between columns
        mainAxisSpacing: 1.0, // Spacing between rows
      ),
      itemCount: 10, // Total number of grid items
      itemBuilder: (BuildContext context, int index) {
        return GridTile(
          child: Container(
            decoration: BoxDecoration(
              color: Colors.blueAccent,
```

```
            borderRadius: BorderRadius.circular(8.0),
          ),

          child: Center(
            child: Text(
              "hello world $index",
              style: TextStyle(
                color: Colors.white,
                fontSize: 16.0,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ),
      );
    },
  );
 }
}
```

Lab -2

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

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

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

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   home: const HomeScreen(),
   debugShowCheckedModeBanner: false,
   theme: ThemeData(primarySwatch: Colors.indigo),
  );
 }
}

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

 @override
 State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
 final List<Map<String, String>> todoList = [];
 final GlobalKey<AnimatedListState> listKey = GlobalKey<AnimatedListState>();
```

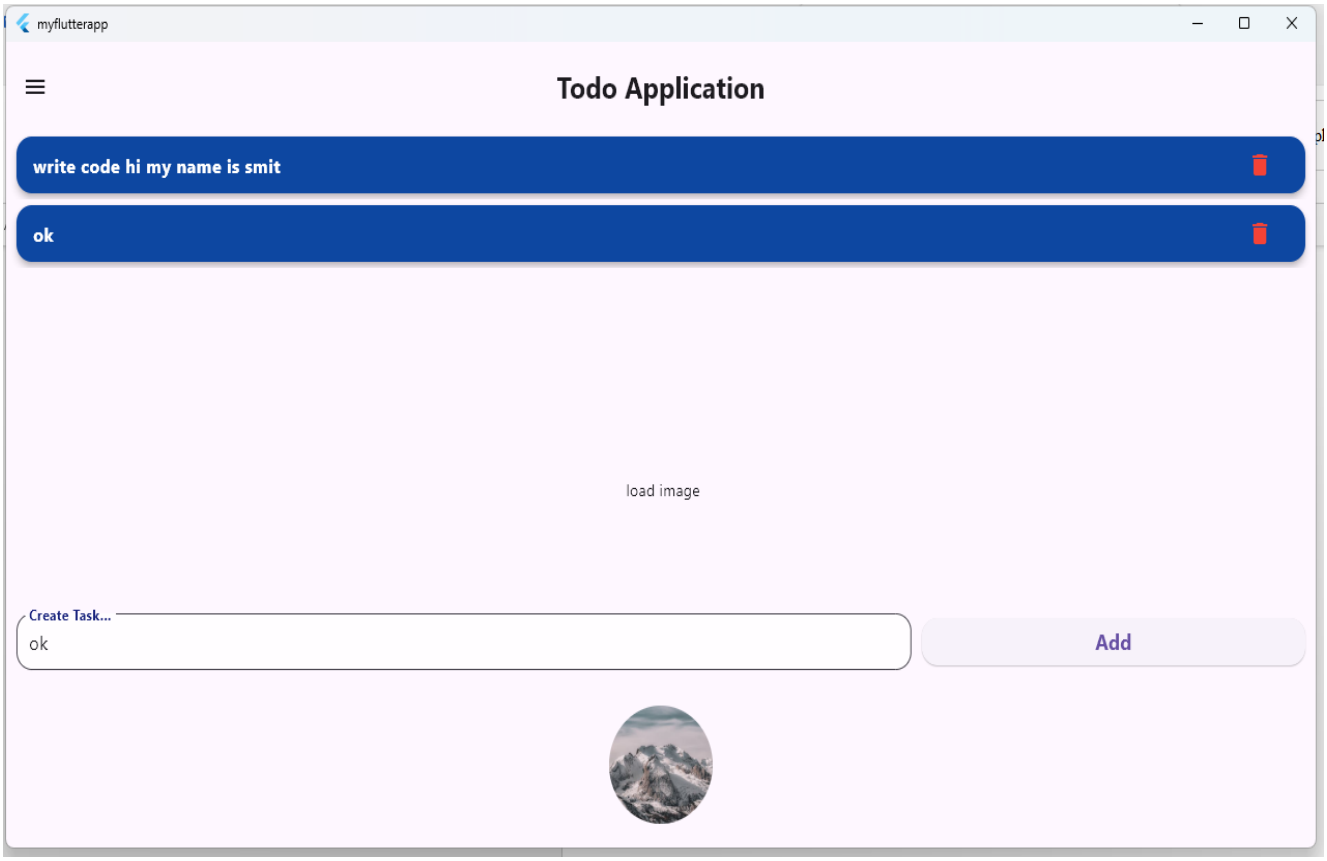
```
String singleValue = "";

addString(content) {
  setState(() {
    singleValue = content;
  });
}

addList() {
  if (singleValue.trim().isEmpty) return;
  final newItem = {"value": singleValue};
  setState(() {
    todoList.add(newItem);
    listKey.currentState?.insertItem(todoList.length - 1);
  });
}

deleteItem(int index) {
  final removedItem = todoList[index];
  setState(() {
    listKey.currentState?.removeItem(
      index,
      (context, animation) => _buildAnimatedItem(removedItem, animation),
    );
    todoList.removeAt(index);
  });
}

Widget _buildAnimatedItem(Map<String, String> item, Animation<double> animation) {
  return SizeTransition(
    sizeFactor: animation,
    child: Card(
      margin: const EdgeInsets.symmetric(vertical: 5, horizontal: 10),
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),
      elevation: 5,
      color: Colors.blue[900],
      child: ListTile(
        title: Text(
          item['value']!,
          style: const TextStyle(color: Colors.white, fontWeight: FontWeight.bold),
        ),
        trailing: IconButton(
          icon: const Icon(Icons.delete, color: Colors.red),
          onPressed: () {
            deleteItem(todoList.indexOf(item));
          },
        ),
      ),
    ),
  );
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text(
        "Todo Application",
        style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25),
      ),
      centerTitle: true,
      toolbarHeight: 75,
      leading: IconButton(
        icon: const Icon(Icons.menu),
        onPressed: () {},
      ),
      elevation: 0,
    ),
    body: Stack(
      children: [
        // Full-Screen Background Image from URL
        Positioned.fill(
          child: Image.network(
            'https://www.pexels.com/photo/timelapse-photography-off-water-fountain-719396/',
             width: 100,
                 height: 100, // URL for the image
            fit: BoxFit.cover,
            loadingBuilder: (context, child, loadingProgress) {
              if (loadingProgress == null) return child;
              return const Center(
                child: CircularProgressIndicator(),
              );
            },
            errorBuilder: (context, error, stackTrace) {
              return const Center(
                child: Text(' load image'),
              );
            },
          ),
        ),
        Column(
          children: [
            Expanded(
              flex: 90,
              child: AnimatedList(
                key: listKey,
                initialItemCount: todoList.length,
                itemBuilder: (context, index, animation) {
                  return _buildAnimatedItem(todoList[index], animation);
                },
              ),
            ),
```

```
Padding(
  padding: const EdgeInsets.all(10),
  child: Row(
    children: [
      Expanded(
        flex: 70,
        child: TextFormField(
          onChanged: (content) {
            addString(content);
          },
          decoration: InputDecoration(
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(15),
            ),
            fillColor: Colors.white.withOpacity(0.8),
            filled: true,
            labelText: 'Create Task...',
            labelStyle: TextStyle(
              color: Colors.indigo[900],
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ),
      const SizedBox(width: 10),
      Expanded(
        flex: 30,
        child: ElevatedButton(
          onPressed: () {
            addList();
          },
          style: ElevatedButton.styleFrom(
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(15),
            ),
            padding: const EdgeInsets.symmetric(vertical: 15),
          ),
          child: const Text(
            "Add",
            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
          ),
        ),
      ),
    ],
  ),
),
// Circular Image Example
Padding(
  padding: const EdgeInsets.all(20),
  child: ClipOval(
    child: Image.network(
```

```
                'https://images.pexels.com/photos/1366919/pexels-photo-
1366919.jpeg?auto=compress&cs=tinysrgb&w=300', // URL for the circular image
                width: 100,
                height: 100,
                fit: BoxFit.cover,
                loadingBuilder: (context, child, loadingProgress) {
                  if (loadingProgress == null) return child;
                  return const Center(
                    child: CircularProgressIndicator(),
                  );
                },
                errorBuilder: (context, error, stackTrace) {
                  return const Center(
                    child: Text('Failed to load image'),
                  );
                },
              ),
            ),
          ),
        ],
      ),
    ],
  ),
 );
 }
}
```

OUTPUT:



Latest Applications:

Latest Applications of Flutter
Flutter is widely used in various domains due to its flexibility, fast development, and cross-platform compatibility. Here are some **latest applications** of Flutter:

1. **Cross-Platform Development**:
   - Build apps for Android, iOS, Web, Desktop (Windows, macOS, Linux), and Embedded systems from a single codebase.
2. **E-commerce Applications**:
   - Interactive and visually appealing e-commerce platforms like Alibaba, eBay Motors.
   - Supports state management for seamless cart and checkout functionalities.
3. **Finance & Banking Apps**:
   - Applications like Nubank and Google Pay use Flutter for their financial services, ensuring high performance and security.
4. **Social Media Platforms**:
   - Customizable UIs for features like chat, video streaming, and notifications.
5. **Gaming and AR/VR Apps**:
   - Casual games and lightweight interactive applications with immersive UI/UX.
6. **AI and ML Integration**:
   - Apps utilizing TensorFlow Lite for on-device AI processing.
7. **IoT and Wearables**:
   - Flutter apps designed for IoT dashboards and wearable devices.
8. **Enterprise Solutions**:
   - CRM and ERP systems with powerful APIs and responsive designs.

Learning Outcome:

1. **Understanding Flutter's Core Concepts**:
   - Gain knowledge of the **widget tree** structure, stateful/stateless widgets, and how Flutter builds UI with a declarative approach.
2. **Grid Layout Mastery**:
   - Learn to create responsive and scalable grid-based layouts for dynamic content such as galleries, e-commerce product displays, and dashboards.
3. **Cross-Platform Development Skills**:
   - Build applications that work seamlessly across multiple platforms using a single codebase.
4. **Customization and Reusability**:
   - Implement reusable components and design efficient layouts using Flutter's widgets like GridView and ListView.
5. **Improved Design Skills**:
   - Utilize Flutter's extensive widget library to build aesthetically pleasing and functional UIs.
6. **Performance Optimization**:
   - Understand how to optimize app performance by managing rendering, lazy loading in grids, and widget rebuilding efficiently.
7. **Problem-Solving Using Flutter**:
   - Address common challenges like responsive design, data handling, and asynchronous operations in mobile and web development.
8. **Foundation for Advanced Flutter Development**:
   - Serve as a stepping stone to advanced topics like state management (Provider, Riverpod, Bloc), animations, and platform-specific integrations.