Lab 9

Link for assignment : https://github.com/Diya-Lad/SDP

Tutorial 1

Here we have created special stateless widget called QuoteCard for printing our quotes in special pattern.

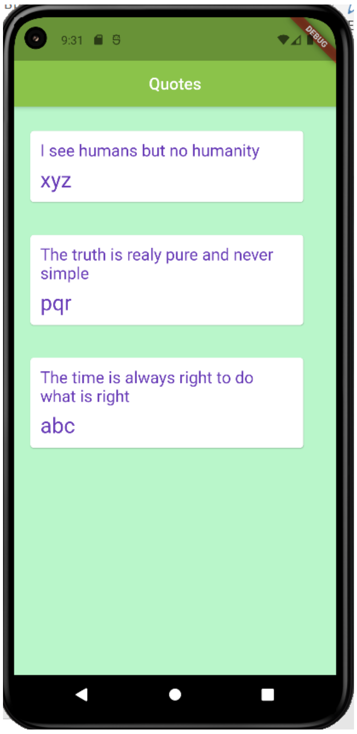
Earlier we have done this using widget in _EchoListState class.

Main.dart

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

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

class EchoList extends StatefulWidget {
  const EchoList({Key? key}) : super(key: key);

  @override
  State<EchoList> createState() => _EchoListState();
}

class _EchoListState extends State<EchoList> {
  @override
  List<Quote> quotes = [
    Quote(text: 'The truth is realy pure and never simple',author:
'author1'),
    Quote(author: ' author2', text: 'I see humans but no humanity'),
    Quote(text: 'The time is always right to do what is right',author: '
author3'),
  ];


  Widget quoteTemplate(quote){
    return QuoteCard(quote: quote,);
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.lightBlueAccent[100],
      appBar: AppBar(
        title: Text('Quotes'),
        centerTitle: true,
        backgroundColor: Colors.orangeAccent,
      ),
      body: Column(
        children: quotes.map((quote) => QuoteCard(quote: quote)).toList(),
      ),
    );
  }
}
```

quote_dart.dart

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

class QuoteCard extends StatelessWidget {

  final Quote quote;
  QuoteCard({required this.quote});
  @override
  Widget build(BuildContext context) {
    return Card(
      margin: EdgeInsets.fromLTRB(20.0, 30.0, 40.0, 10.0),
      child: Padding(
        padding: const EdgeInsets.all(12.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            Text(
              quote.text,
              style: TextStyle(
                fontSize: 20,
                color: Colors.deepPurple,
              ),
            ),
            SizedBox(height: 10),
            Text(
              quote.author,
              style: TextStyle(
                fontSize: 26,
                color: Colors.deepPurple,
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```



Quotes

I see humans but no humanity
xyz

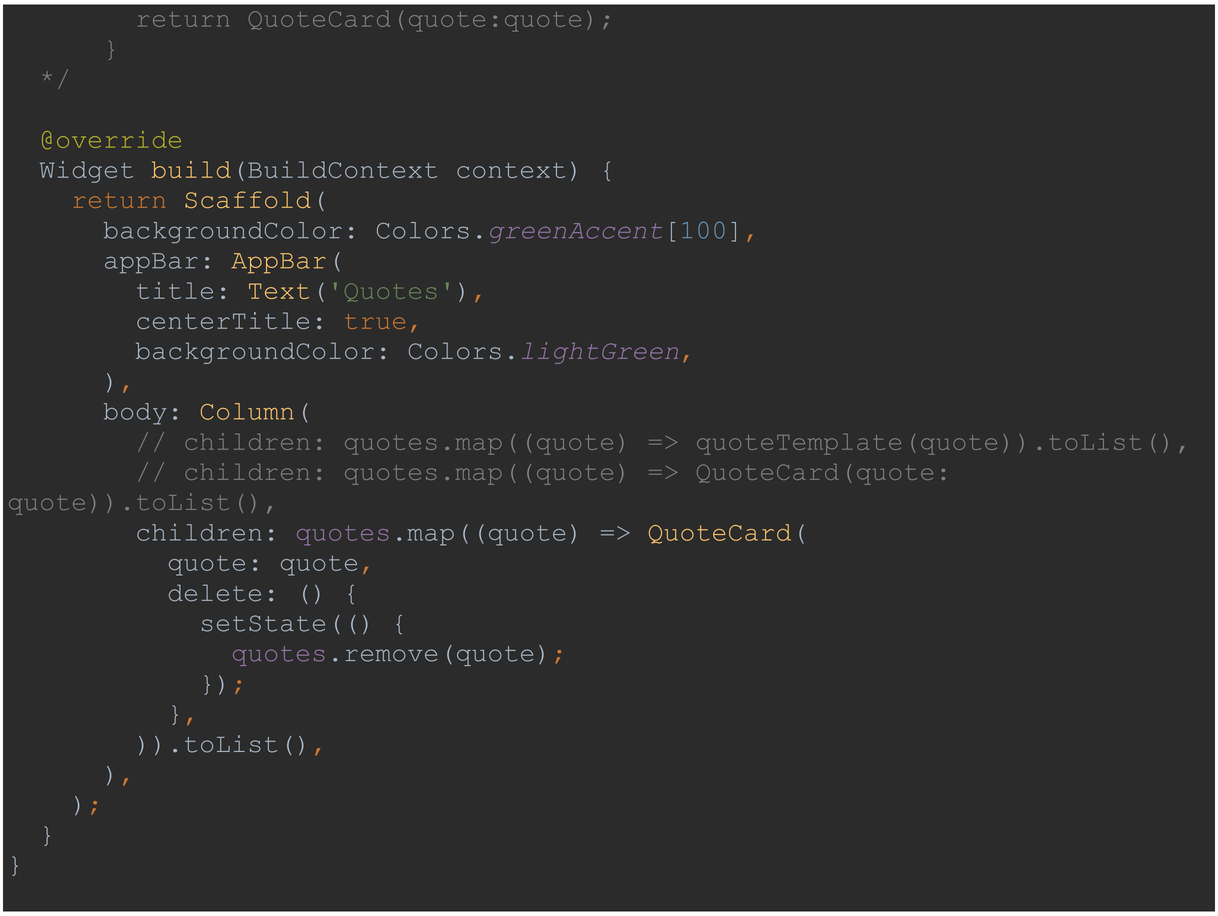
The truth is realy pure and never simple
pqr

The time is always right to do what is right
abc

Code test 2: Function as arguments and real use of “Stateful widget” and delete operation

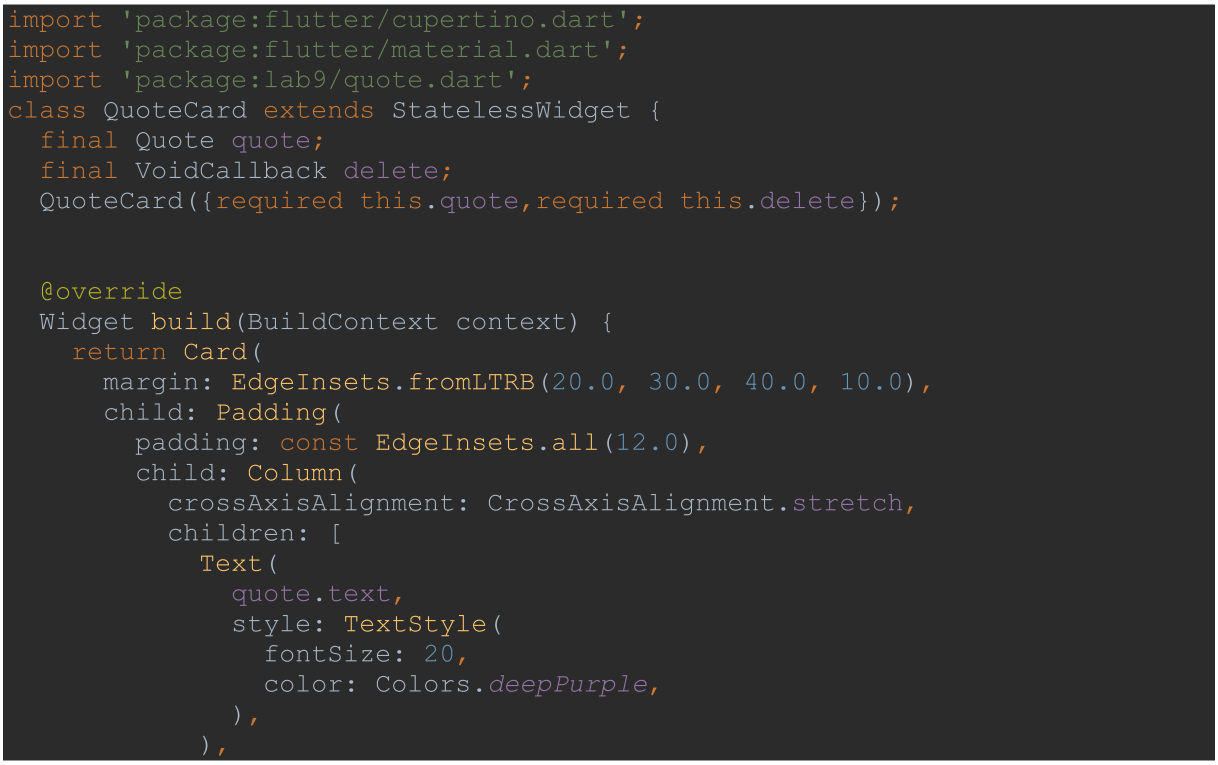
Main.dart

```
import 'package:flutter/material.dart';
import 'quote.dart';
import 'package:lab9/quote_card.dart';
void main() => runApp(MaterialApp(
  home: EchoList(),
));

class EchoList extends StatefulWidget {
  const EchoList({Key? key}) : super(key: key);

  @override
  State<EchoList> createState() => _EchoListState();
}

class _EchoListState extends State<EchoList> {
  List<Quote> quotes = [
    Quote(author: 'xyz', text: 'I see humans but no humanity'),
    Quote(text: 'The truth is realy pure and never simple', author: 'pqr'),
    Quote(
        text: 'The time is always right to do what is right',
        author: 'abc'),
  ];
  /*
      Widget quoteTemplate(quote) {
```

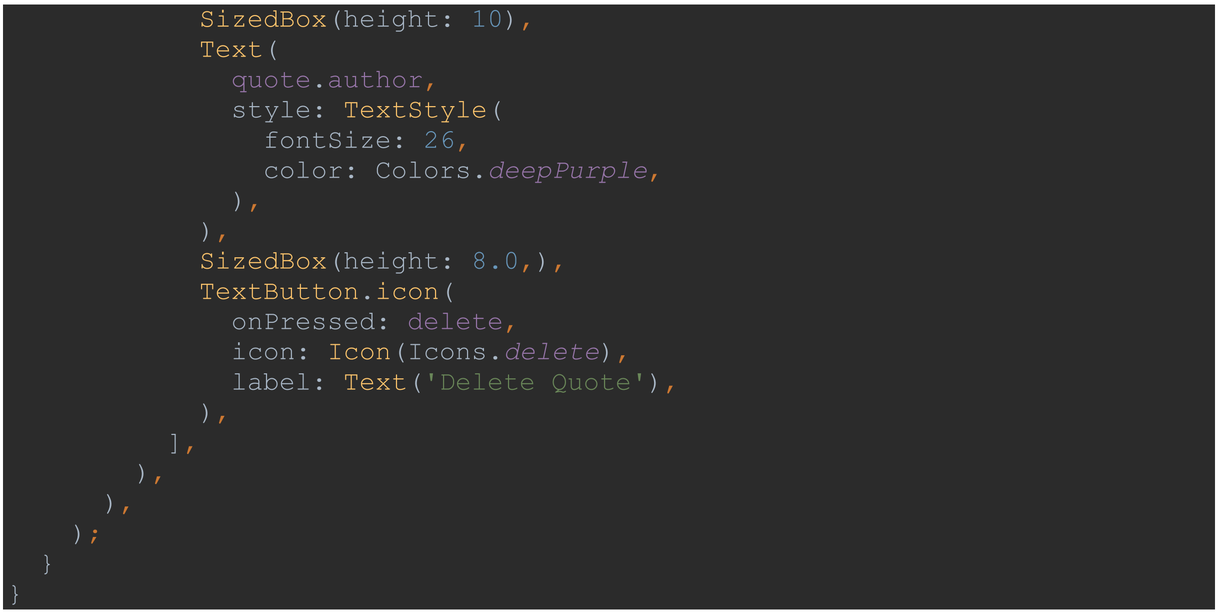


```dart
        return QuoteCard(quote:quote);
      }
  */

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.greenAccent[100],
      appBar: AppBar(
        title: Text('Quotes'),
        centerTitle: true,
        backgroundColor: Colors.lightGreen,
      ),
      body: Column(
        // children: quotes.map((quote) => quoteTemplate(quote)).toList(),
        // children: quotes.map((quote) => QuoteCard(quote:
quote)).toList(),
        children: quotes.map((quote) => QuoteCard(
          quote: quote,
          delete: () {
            setState(() {
              quotes.remove(quote);
            });
          },
        )).toList(),
      ),
    );
  }
}
```

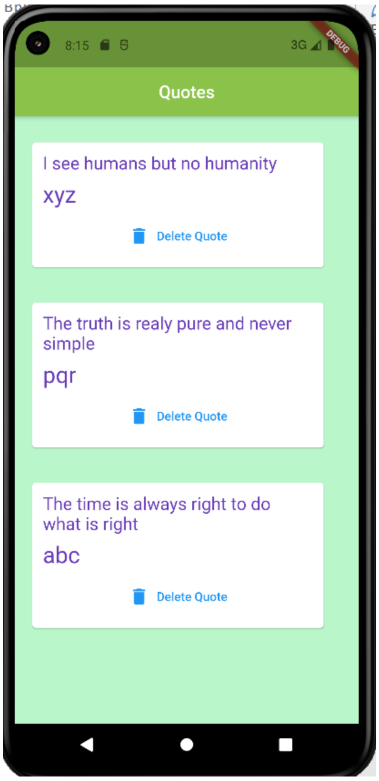
quote_card.dart



```dart
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:lab9/quote.dart';
class QuoteCard extends StatelessWidget {
  final Quote quote;
  final VoidCallback delete;
  QuoteCard({required this.quote,required this.delete});


  @override
  Widget build(BuildContext context) {
    return Card(
      margin: EdgeInsets.fromLTRB(20.0, 30.0, 40.0, 10.0),
      child: Padding(
        padding: const EdgeInsets.all(12.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            Text(
              quote.text,
              style: TextStyle(
                fontSize: 20,
                color: Colors.deepPurple,
              ),
            ),
```



```
            SizedBox(height: 10),
            Text(
              quote.author,
              style: TextStyle(
                fontSize: 26,
                color: Colors.deepPurple,
              ),
            ),
            SizedBox(height: 8.0,),
            TextButton.icon(
              onPressed: delete,
              icon: Icon(Icons.delete),
              label: Text('Delete Quote'),
            ),
          ],
        ),
      ),
    );
  }
}
```



Quotes

I see humans but no humanity

xyz

Delete Quote

The truth is realy pure and never simple

pqr

Delete Quote

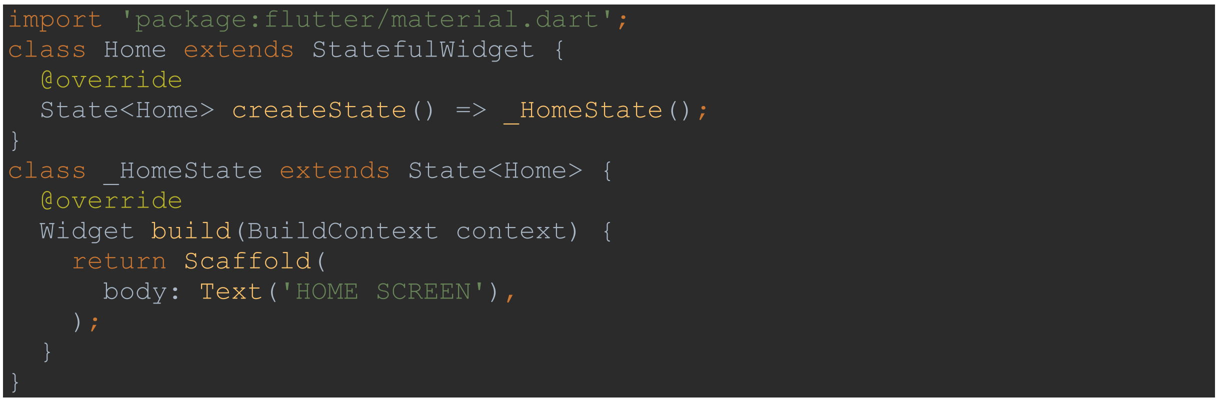
The time is always right to do what is right

abc

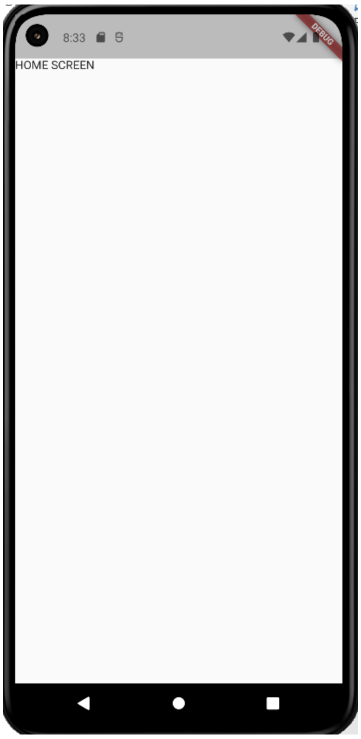
Delete Quote

Tutorial 2

In home.dart



```
import 'package:flutter/material.dart';
class Home extends StatefulWidget {
  @override
  State<Home> createState() => _HomeState();
}
class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Text('HOME SCREEN'),
    );
  }
}
```



HOME SCREEN

Route widget

- The application's top-level routing table.
- When a named route is pushed with Navigator.pushNamed(), the route name is looked up in this map. If the name is present, the associated widgets.WidgetBuilder() is used to construct a MaterialPageRoute that performs an appropriate transition, including Hero animations, to the new route.
- If the app only has one page, then you can specify it using home instead.

- If home is specified, then it implies an entry in this table for the Navigator.defaultRouteName() route (/), and it is an error to redundantly provide such a route in the routes table.
- If a route is requested that is not specified in this table (or by home), then the onGenerateRoute() callback is called to build the page instead.
- The Navigator is only built if routes are provided (either via home, routes, onGenerateRoute, or onUnknownRoute); if they are not, builder must not be null.

Navigation between the pages:

Main.dart :

```dart
void main() => runApp(MaterialApp(
  // home: Home(),
    initialRoute: '/home',
    routes: {
      '/': (context) => Loading(),
      '/home': (context) => Home(),
      '/location': (context) => ChooseLocation(),
    }
));
```

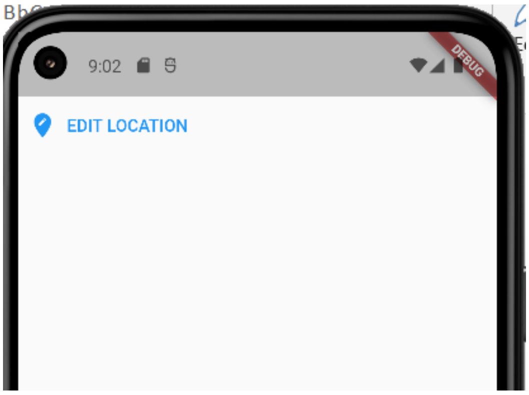
home.dart :

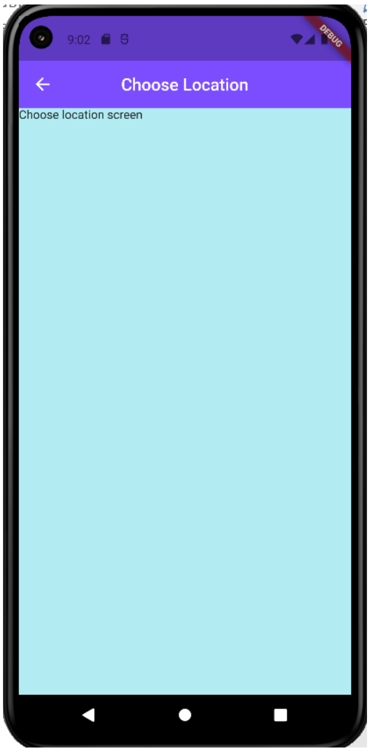
```dart
import 'package:flutter/material.dart';
class Home extends StatefulWidget {
  @override
  State<Home> createState() => _HomeState();
}
class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
          child:Column(
            children: [
              TextButton.icon(
                onPressed: (){
                  Navigator.pushNamed(context, '/location');
                },
                icon: Icon(Icons.edit_location),
                label:Text('EDIT LOCATION'),
              )
            ],
          )
      ),
    );
  }
}
```

choose_location.dart :

```
import 'package:flutter/material.dart';
class ChooseLocation extends StatefulWidget {
// const ChooseLocation({Key? key}) : super(key: key);
  @override
  State<ChooseLocation> createState() => _ChooseLocationState();
}
class _ChooseLocationState extends State<ChooseLocation> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.cyan[100],
      appBar: AppBar(
        backgroundColor: Colors.deepPurpleAccent,
        title: Text('Choose Location'),
        centerTitle: true,
        elevation: 0,
      ),
      body: Text('Choose location screen'),
    );
  }
}
```

- The Navigator class provides all the navigation capabilities in a Flutter app.
- Navigator provides methods to mutate the stack by a push to stack or by popping from the stack. The Navigator.push() method is for navigating to a newer page and Navigator.pop() is for going back from the current page.





Choose location screen