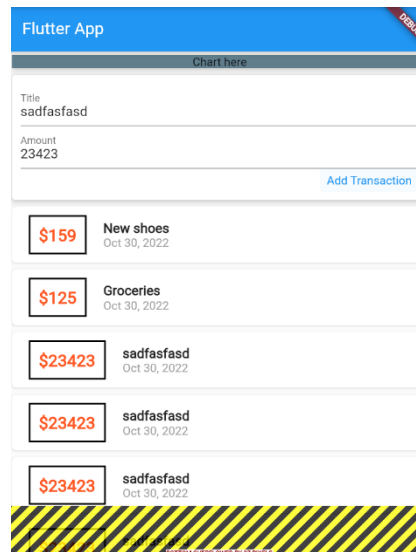


Week6

Widgets and Styling



- What happens when the content flows out of the window? A yellow-black warning line appears.
- Solution is adding scrolling feature the content which does not fit the given space.
- Wrap the Column of the body: with a Container and rename the Container with SingleChildScrollView. Now try to add more data and see a scrolling bar appears when you add more transactions.
- If you try this on the Column of the list data section, it wont work. Because SingleChildScrollView does not know the exact height of that section. To do this, wrap this Column by a Container and give a height:300 then wrap the Column with a SingleChildScrollView.

```
Container(  
  height: 300,  
  child: SingleChildScrollView(  
    child: Column(  
      children: transactions.map((tx) {  
        return Card(  

```

- There is shortcut for SingleChildScrollView and Column combination: ListView

```
Container(  
  height: 300,  
  child: ListView(  
    children: transactions.map((tx) {  
      return Card(  

```

- ListView have two usage: 1) **ListView(children:[])** and 2) **ListView.builder()**
- 1) All children widgets are rendered even they are off the screen.
- 2) Only the visible widgets are rendered. And more suitable for big lists & performance.
- To use ListView.builder() in our app:
- First, change the **ListView**(to **ListView.builder**(
- Add this parameter **itemBuilder: (bc,index){ }**,
- Add this parameter **itemCount: transaction.length,**

- Cut the **return Card** section from **children: transactions.map((tx) {** to itemBuilder's anonymous function.
- Delete **children: transactions.map((tx) { }).toList()**,
- Change **tx** definition inside the function as **transactions[index]**
- The final code should be:

```

Container(
  height: 300,
  child: ListView.builder(
    itemCount: transactions.length,
    itemBuilder: (bc, index){
      return Card(
        child: Row(
          children: [
            Container(
              margin:
                EdgeInsets.symmetric(vertical: 10, horizontal: 20),
              decoration: BoxDecoration(
                border: Border.all(
                  color: Colors.black,
                  width: 2,
                )),
              padding: EdgeInsets.all(10),
              child: Text(
                '\${transactions[index].amount}',
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 20,
                  color: Colors.deepOrange,
                ),
              ),
            ),
          ],
        ),
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              transactions[index].title,
              style: TextStyle(
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
            Text(
              DateFormat.yMMMd().format(transactions[index].date),
              style: TextStyle(
                color: Colors.grey,
              ),
            ),
          ], //column children
        ), //column
      ], //row children
    ), //row
  ),

```

```

    ); //card
  },
), //listview builder
), //container of the list view

```

- Add keyboardType: TextInputType.number, to the Amount TextField, than this widget only accepts numbers and a number keyboard is shown when user clicks this input.
- To fix the decimal digits of the amount change with this '\\${transactions[index].amount}', '\\$'+transactions[index].amount.toStringAsFixed(2),
- To add some extra buttons, use this:

```

appBar: AppBar(
  title: Text('Flutter App'),
  actions: [
    IconButton(
      icon: Icon(Icons.add),
      onPressed: () {},
    ),
  ],
),
floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,
floatingActionButton: FloatingActionButton(
  child: Icon(Icons.add),
  onPressed: () {},
),
body:

```

- Use a showModalBottomSheet to hide the input area. Show the input are when user clicks the FloatingActionButton or the appBar IconButton: (we will combine classes into a single stateful class)

```

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter App',
      home: TransactionList(),
    );
  }
}

class TransactionList extends StatefulWidget {
  @override
  _TransactionListState createState() => _TransactionListState();
}

class _TransactionListState extends State<TransactionList> {
  final List<Transaction> transactions = [
    Transaction(
      id: 't1', title: 'New shoes', amount: 159.0, date: DateTime.now()),
    Transaction(
      id: 't2', title: 'Groceries', amount: 125.0, date: DateTime.now()),
  ];
  final titleController = TextEditingController();
  final amountController = TextEditingController();
  void addTransaction(String txtitle, double txamount) {
    final newTx = Transaction(
      title: txtitle, amount: txamount, id: 'tx', date: DateTime.now());
    setState(() {
      transactions.add(newTx);
    });
  }
}

```

```

void _startAddNewTransaction(BuildContext ctx) {
  showModalBottomSheet(
    context: ctx,
    builder: (ctx) {
      return GestureDetector(
        behavior: HitTestBehavior.opaque,
        onTap: () {}, //use this to prevent closing of the bottomsheet when you click inside the sheet
        child: Card(
          elevation: 5,
          child: Container(
            padding: EdgeInsets.all(10),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.end,
              children: [
                TextField(
                  decoration: InputDecoration(labelText: 'Title'),
                  controller: titleController,
                ),
                TextField(
                  decoration: InputDecoration(labelText: 'Amount'),
                  controller: amountController,
                  keyboardType: TextInputType.number,
                ),
                TextButton(
                  child: Text('Add Transaction'),
                  onPressed: () {
                    print(titleController.text);
                    print(amountController.text);
                    addTransaction(titleController.text,
                      double.parse(amountController.text));
                  },
                ), //textbutton
              ], //column children
            ), //column
          ), //container
        ), //card
      ); //gesturedetector return
    }, //showModalBottomSheet builder
  ); //showModalBottomSheet function
} //_startAddNewTransaction function

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Flutter App'),
      actions: [
        IconButton(
          icon: Icon(Icons.add),
          onPressed: () {
            _startAddNewTransaction(context);
          },
        ),
      ],
    ),
    floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,
    floatingActionButton: FloatingActionButton(
      child: Icon(Icons.add),
      onPressed: () {
        _startAddNewTransaction(context);
      },
    ),
    body: SingleChildScrollView(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [
          Card(
            child: Container(
              child: Text('Chart here', textAlign: TextAlign.center),
              width: double.infinity,
              color: Colors.blueGrey,
            ),
            elevation: 5,
          ),
          Column(
            children: [
              //input area was here

```

```

Container(
  height: 300,
  child: ListView.builder(
    itemCount: transactions.length,
    itemBuilder: (bc, index) {
      return Card(
        child: Row(
          children: [
            Container(
              margin: EdgeInsets.symmetric(
                vertical: 10, horizontal: 20),
              decoration: BoxDecoration(
                border: Border.all(
                  color: Colors.black,
                  width: 2,
                )),
              padding: EdgeInsets.all(10),
              child: Text(
                '\$' +
                  transactions[index]
                    .amount
                    .toStringAsFixed(2),
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 20,
                  color: Colors.deepOrange,
                ),
              ),
            ),
          ],
        ),
      Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            transactions[index].title,
            style: TextStyle(
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
          Text(
            DateFormat.yMMMd()
              .format(transactions[index].date),
            style: TextStyle(
              color: Colors.grey,
            ),
          ),
        ], //column children
      ), //column
    ], //row children
  ), //row
); //card
}, //itemBuilder
), //listview builder
), //container of the list view
], //column children
), //column,
], //column children
), //column
), //SingleChildScrollView
); //scaffold
}
}

class Transaction {
  final String id;
  final String title;
  final double amount;
  final DateTime date;
  Transaction(
    {required this.id,
    required this.title,
    required this.amount,
    required this.date});
}

```

- To close the Modal Bottom Sheet after clicking Add Transaction button add this line at the bottom of addTransaction function: **Navigator.of(context).pop();**
- Using themes to define global styling:

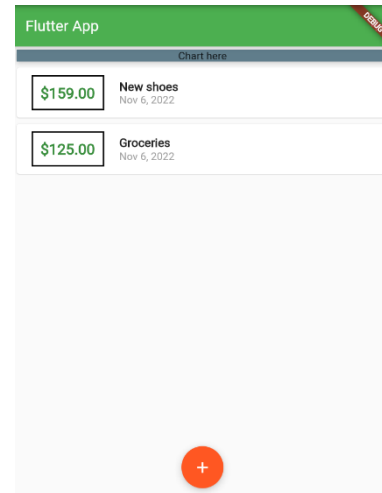
- At return **MaterialApp**(section add **theme:** parameter.
- Theme: parameter accepts ThemeData class. It has many styling parameters but you can choose whatever you want to override.
- Use primarySwatch instead of primaryColor to define a base color and Flutter will generate different shades of that primary color.

```
theme: ThemeData(
  primarySwatch: Colors.purple,
),
```

- Change the color of the list amount find **color:** **Colors.deepOrange**, and change it to **color: Theme.of(context).primaryColorDark**,
- Change the theme color to **primarySwatch: Colors.green**, and see what happens.
- You can add **accentColor: Colors.deepOrange**, to ThemeData. Some of the components such as FloatingActionButton will use this alternative color if it is defined.
- To add custom fonts to your project: Create a folder named asset and inside that folder create a new folder named fonts. Then, place the font files in to this folder.
- Open pubspec.yaml file and find the fonts: section and uncomment.
- Add fontFamily to ThemeData or TextStyle's.
- Adding images to the project.
- Create an images folder inside the assets folder. Then, copy an image file into this folder.
- Uncomment assets images section of the pubspec.yaml file.
- Use this to add your image into your project: `Image.asset('assets/images/empty.png')`,
- If you're using dartpad you can also add an image from internet URL.

```
child: transactions.isEmpty ?
Column(
  children: [
    Text('No transactions yet'),
    SizedBox(
      height: 20,
    ),
    Image.network('https://static.thenounproject.com/png/194055-200.png',
      fit: BoxFit.cover,
    ),
  ],
)
:
ListView.builder(
```

- There are third party charting packages for Flutter but we will try to create a simple chart.
- Add **margin: EdgeInsets.all(20)**, to the Card, we declare it for the chart.
- Change this **child: Text('Chart here', textAlign: TextAlign.center)**, to a **child: Row ...**



- Create a getter function to analyze the transactions list and generate a summary of the last week's total transactions:

```
List<Map<String, Object>> get groupedTransactionValues {
  return List.generate(7, (index) {
    final weekDay = DateTime.now().subtract(Duration(days: index));
    double totalSum = 0;
    for (var i = 0; i < transactions.length; i++) {
      if (transactions[i].date.day == weekDay.day &&
          transactions[i].date.month == weekDay.month &&
          transactions[i].date.year == weekDay.year) {
        totalSum += transactions[i].amount;
      }
    }
    return {
      'day': DateFormat.E().format(weekDay).substring(0,1),
      'amount': totalSum,
    };
  });
}
```

- Change the chart Card like this:

```
Card(
  child: Container(
    child: Row(
      children: groupedTransactionValues.map((data) {
        return Text(data['day'].toString() + ' ' + data['amount'].toString());
      }).toList(),
    ),
    width: double.infinity,
    color: Colors.blueGrey,
  ),
  elevation: 5,
  margin: EdgeInsets.all(20),
),
```

- Add a getter to get total spending:

```
double get totalSpending {
  return groupedTransactionValues.fold(0.0, (sum, item) {
    return sum + (item['amount'] as double);
  });
}
```

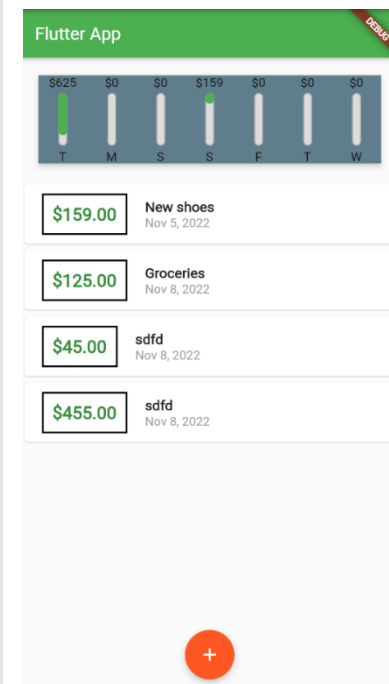
- Create the bars:

```
Card(
  child: Container(
    child: Row(
      children: groupedTransactionValues.map((data) {
        final spendingPctOfTotal = totalSpending == 0.0
          ? 0.0
          : (data['amount'] as double) / totalSpending;
        //return Text(data['day'].toString() + ' ' + data['amount'].toString());
        return Flexible(
          fit: FlexFit.tight,
```

```

child: Column(
  children: <Widget>[
    FittedBox(
      child: Text('\${data['amount']}'),
    ),
    SizedBox(
      height: 4,
    ),
    Container(
      height: 60,
      width: 10,
      child: Stack(
        children: <Widget>[
          Container(
            decoration: BoxDecoration(
              border: Border.all(
                color: Colors.grey, width: 1.0),
                color: Color.fromRGBO(220, 220, 220, 1),
                borderRadius: BorderRadius.circular(10),
              ),
            ),
          ),
          FractionallySizedBox(
            heightFactor: spendingPctOfTotal,
            child: Container(
              decoration: BoxDecoration(
                color: Theme.of(context).primaryColor,
                borderRadius: BorderRadius.circular(10),
              ),
            ),
          ),
        ],
      ),
    ),
    SizedBox(
      height: 4,
    ),
    Text(data['day'].toString()),
  ],
),
);
}).toList(),
),
width: double.infinity,
color: Colors.blueGrey,
),
elevation: 5,
margin: EdgeInsets.all(20),
),

```



References

- <https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/>
- <https://developers.google.com/community/experts/directory/profile/profile-gazihan-alankus>