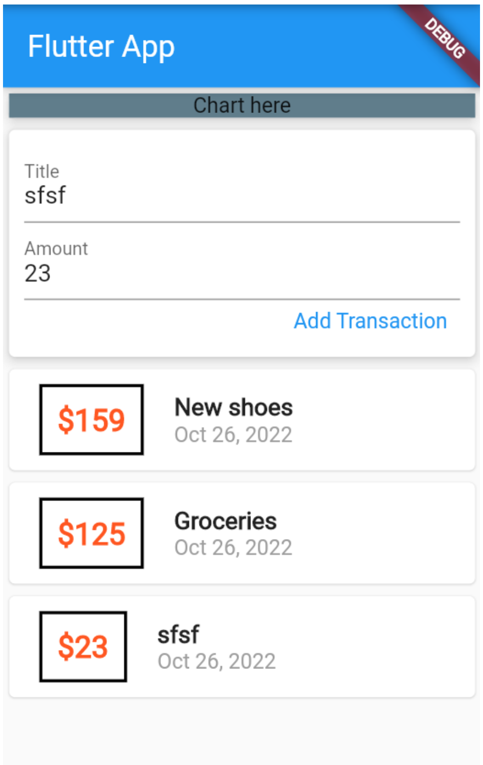
Week4

Widgets and Styling

Most important widgets

- Flutter widget catalog: https://docs.flutter.dev/development/ui/widgets
- App/page setup: MaterialApp and Scaffold
- Layout: Container, Row, Column, Flexible, Extended
- Content Containers: Stack (enable layers) and Card
- Repeat Elements: ListView and GridView (scrollable), ListTile
- Content: Text, Image, Icon
- User Input: TextField, RaisedButton, FlatButton, GestureDetector, InkWell



Title
sfsf

Amount
23

$159 New shoes
Oct 26, 2022

$125 Groceries
Oct 26, 2022

$23 sfsf
Oct 26, 2022

Designing a real app

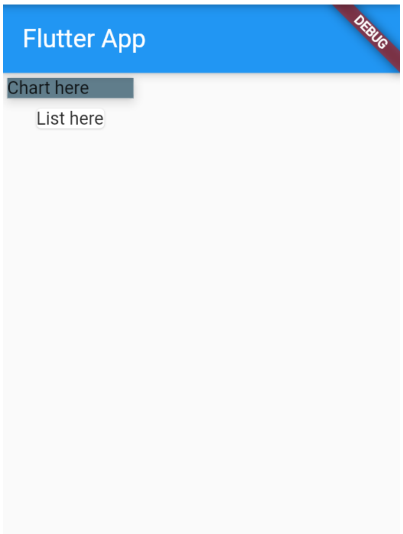
- We will create an app to track our expenses.
- At the top we will have a summary chart, below that we will have a list that shows our expenses. Use the below code as a start point:

```
import 'package:flutter/material.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Flutter App',
   home: MyHomePage(),
  );
 }
}
```

```
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter App'),
      ),
      body: Text('Body'),
    );
  }
}
```

- Change the body like this:

```
      body: Column(
        children: [
          Card(
            child: Text('Chart here'),
            elevation: 5,
          ),
          Card(
            child: Text('List here'),
          ),
        ],
      ),
```



- Wrap the Text('Chart here') with a Container to add styling easily.

```
          Card(
            child: Container(
              child: Text('Chart here'),
              width: 100,
              color: Colors.blueGrey,
            ),
            elevation: 5,
          ),
```

- Change width: double.infinity, to set the size device width.
- Add alignment styling to Column: mainAxisAlignment: MainAxisAlignment.center,
- crossAxisAlignment: CrossAxisAlignment.start,
- Then change like this: crossAxisAlignment: CrossAxisAlignment.stretch,
- Create a new class for defining the transactions:

```
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});
}
```

- Add a list into MyHomePage that consist of transactions:

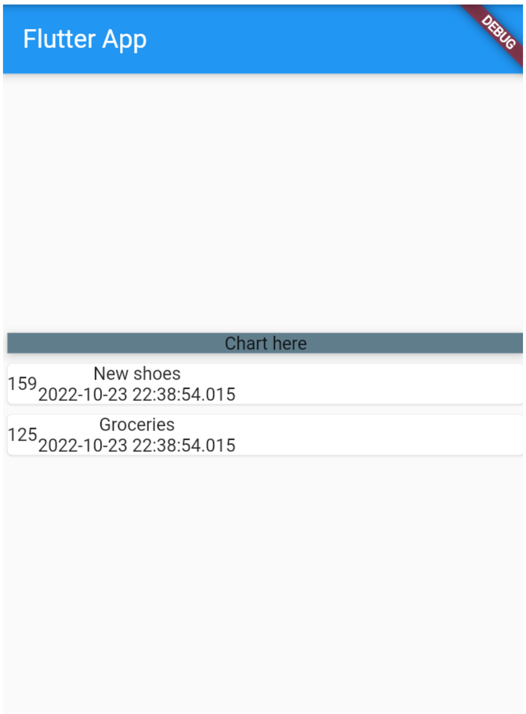
```
final List<Transaction> transactions=[
    Transaction(
      id: 't1',
      title: 'New shoes',
      amount: 159.0,
      date:DateTime.now()
    ),
    Transaction(
      id: 't2',
      title: 'Groceries',
      amount: 125.0,
      date:DateTime.now()
    ),
  ];
```

- Delete the second card (for the list) and add this:

```
Column(
            children: transactions.map( (tx){
              return Card(
                child: Text(tx.title),
              );
            }).toList(),
          ),
```

- To build a custom list change the column like this:

```
   Column(
            children: transactions.map((tx) {
              return Card(
                child: Row(
                  children: [
                    Container(
                      child: Text(tx.amount.toString()),
                    ),
                    Column(
                      children: [
                        Text(tx.title),
                        Text(tx.date.toString()),
                      ], //column children
                    ),//column
                  ],//row children
                ),//row
              );//card
            }).toList(),//map
          ),//column
```



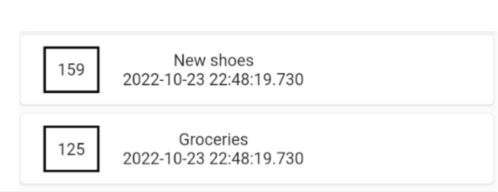
Flutter App

Chart here

```
159 New shoes
    2022-10-23 22:38:54.015
125 Groceries
    2022-10-23 22:38:54.015
```

- To add custom styling to a container:

```
margin: EdgeInsets.symmetric(vertical: 10,horizontal: 20),
decoration: BoxDecoration(
 border: Border.all(
  color: Colors.black,
  width: 2,
 )
),
```

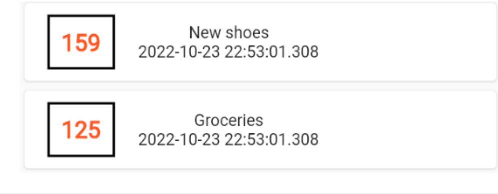


159 New shoes
2022-10-23 22:48:19.730

125 Groceries
2022-10-23 22:48:19.730

- Change text styling of the amount:

```
child: Text(
 tx.amount.toString(),
 style: TextStyle(
  fontWeight: FontWeight.bold,
  fontSize: 20,
  color: Colors.deepOrange,
 ),
),
```

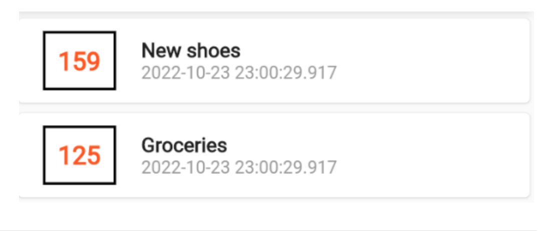


159 New shoes
2022-10-23 22:53:01.308

125 Groceries
2022-10-23 22:53:01.308

- Add crossAxisAlignment: CrossAxisAlignment.start, to title and date column.
- Add styling to the title and date:

```
Text(
 tx.title,
 style: TextStyle(
  fontSize: 16,
  fontWeight: FontWeight.bold,
 ),
),
Text(
 tx.date.toString(),
 style: TextStyle(
  color: Colors.grey,
 ),
),
```



159 New shoes
2022-10-23 23:00:29.917

125 Groceries
2022-10-23 23:00:29.917

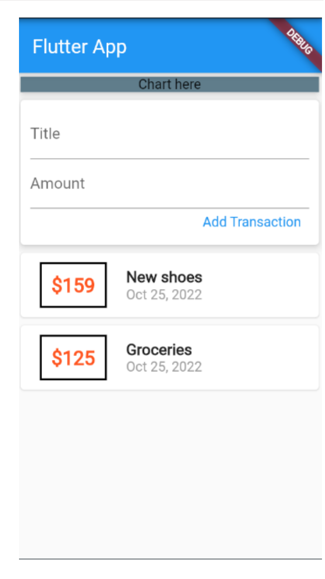
- What is the difference between Container and Column/Row widgets? Container takes only one child widget, rich alignment and styling options, and small changes in width adjustment. Containers are perfect for styling, column/row must be used if widgets sit next to/above each other. Container and Column/Row can be combined together.
- Change this *tx.amount.toString(),* to this *'\$${tx.amount}',* (String interpolation is much easier way to combine/concatenate strings)
- We will use an extra package to format dates more user friendly: search google for *intl*
- To install intl, add *intl: ^0.17.0* to the dependencies part of pubspec.yaml and save the file. Android studio will automatically download the package. If not, type *flutter packages get* in the terminal.
- Click on the api reference on intl pub.dev website.
- Add this to the beginning of your dart code: import 'package:intl/intl.dart';
- Change this *tx.date.toString(),* to *DateFormat().format(tx.date),* or *DateFormat('yyyy-MM-dd').format(tx.date),* or *DateFormat.yMMMd().format(tx.date),*
- Change *mainAxisAlignment: MainAxisAlignment.start,* in *body: Column(*

- We will add an input area below the chart.

```
Card(
  elevation: 5,
  child: Column(
    children: [
      TextField(
      ),
      TextField(
      ),
    ],
  ),
),
```

- Wrap the column with a container to add margins etc. Then add decoration the textfields and add a textbutton.

```
Card(
  elevation: 5,
  child: Container(
    padding: EdgeInsets.all(10),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.end,
      children: [
        TextField(
          decoration: InputDecoration(labelText: 'Title'),
        ),
        TextField(
          decoration: InputDecoration(labelText: 'Amount'),
        ),
        TextButton(
          child: Text('Add Transaction'),
          onPressed: (){},
        ),
      ],
    ),
  ),
),
```



- Add two strings below the List<Transaction> transactions
- *String? titleInput;* and *String? amountInput;*
- Add *onChanged: (val){titleInput=val;},* to the first textfield and *onChanged: (val) => amountInput=val,* to the second.
- Add this to the onPressed of TextButton *print(titleInput); print(amountInput);*
- A better way: remove the String? Definitions and add *final titleContoller = TextEditingController(); final amountController = TextEditingController();* instead.
- Remove onChanged parameters of the TextFields and add *controller: titleContoller,* and *controller: amountController,* instead.
- Change the onPressed of the TextButton as *print(titleContoller.text); print(amountController.text);*

- To change the transactions list, we need to change our stateless widget into a stateful widget.
- In dartpad there is no template shortcuts but you can copy and paste these:

```
class  extends StatefulWidget {
 @override
 _State createState() => _State();
}
class _State extends State<> {
 @override
 Widget build(BuildContext context) {
  return Container();
 }
}

class  extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return Container();
 }
}
```

- In Android Studio create a new file and type stful or stless to access quick templates.
- We will split our app into stateful widgets to make it more readable.
- Create a stateful widget named *TransactionList*, remove transactions list in to this new class, remove the TextEditingControllers into this new class, cut the entire Card section consisting TextFields and *Column(children: transactions.map((tx) {* and paste it to the return code of the stateful widget and call *TransactionList(),* where you cut the entire code.
- Add a new function in *TransactionList*:

```
 void addTransaction(String txtitle,double txamount){
  final newTx=Transaction(title: txtitle,amount: txamount,id: 'tx',date:DateTime.now()  );
  setState(() {
   transactions.add(newTx);
  });
 }
```

- Add this to the onPressed of the TextButton *addTransaction(titleContoller.text,double.parse(amountController.text));*

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