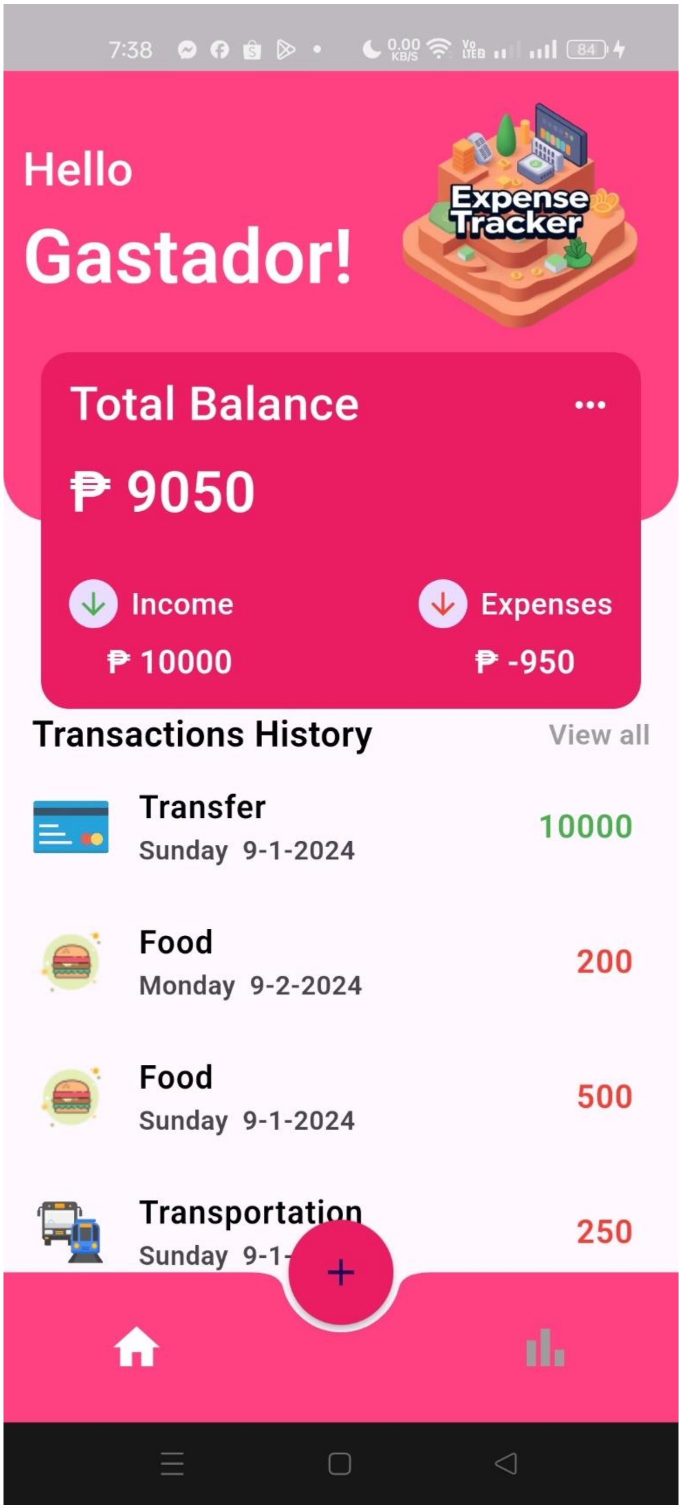
Expense Tracker



7:38

Hello

Gastador!

Total Balance

₱ 9050

Income

₱ 10000

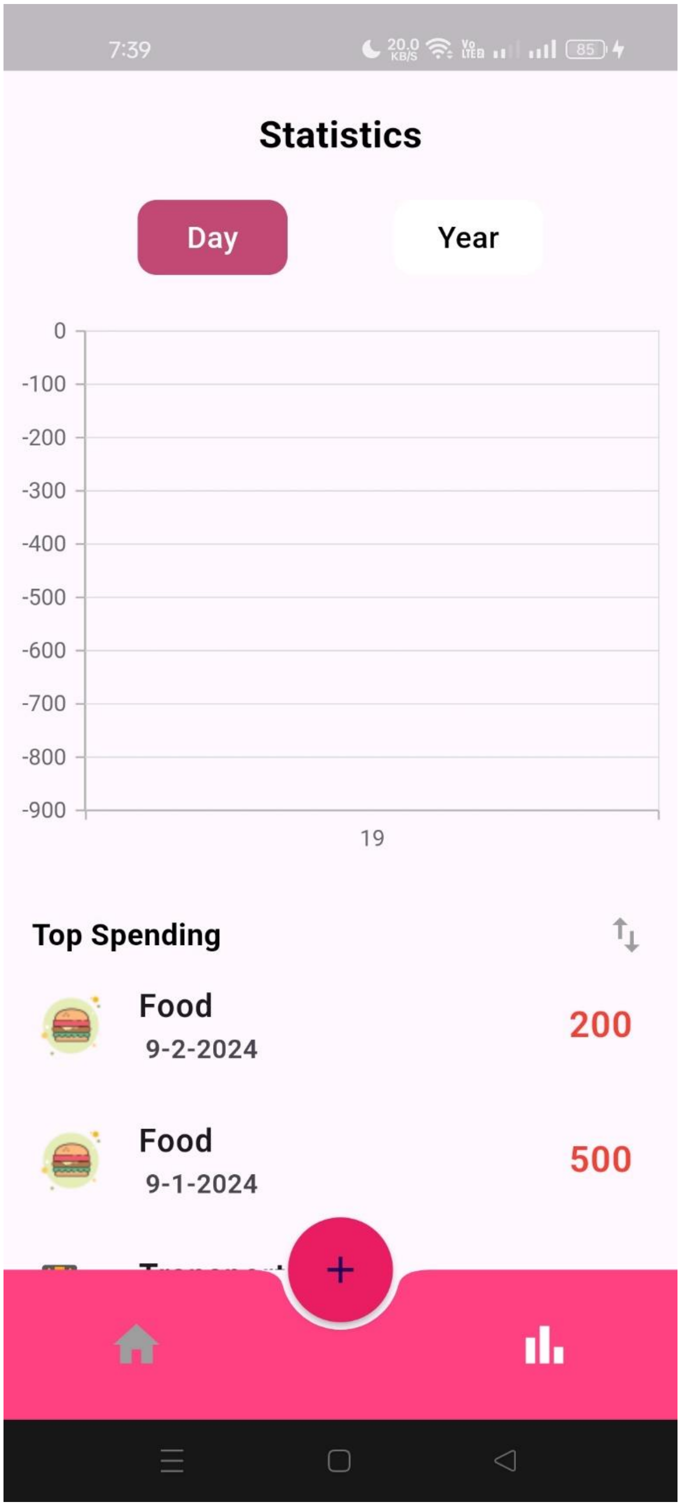
Expenses

₱ -950

Transactions History

View all

Transfer	Sunday 9-1-2024	10000
Food	Monday 9-2-2024	200
Food	Sunday 9-1-2024	500
Transportation	Sunday 9-1-	250



7:39

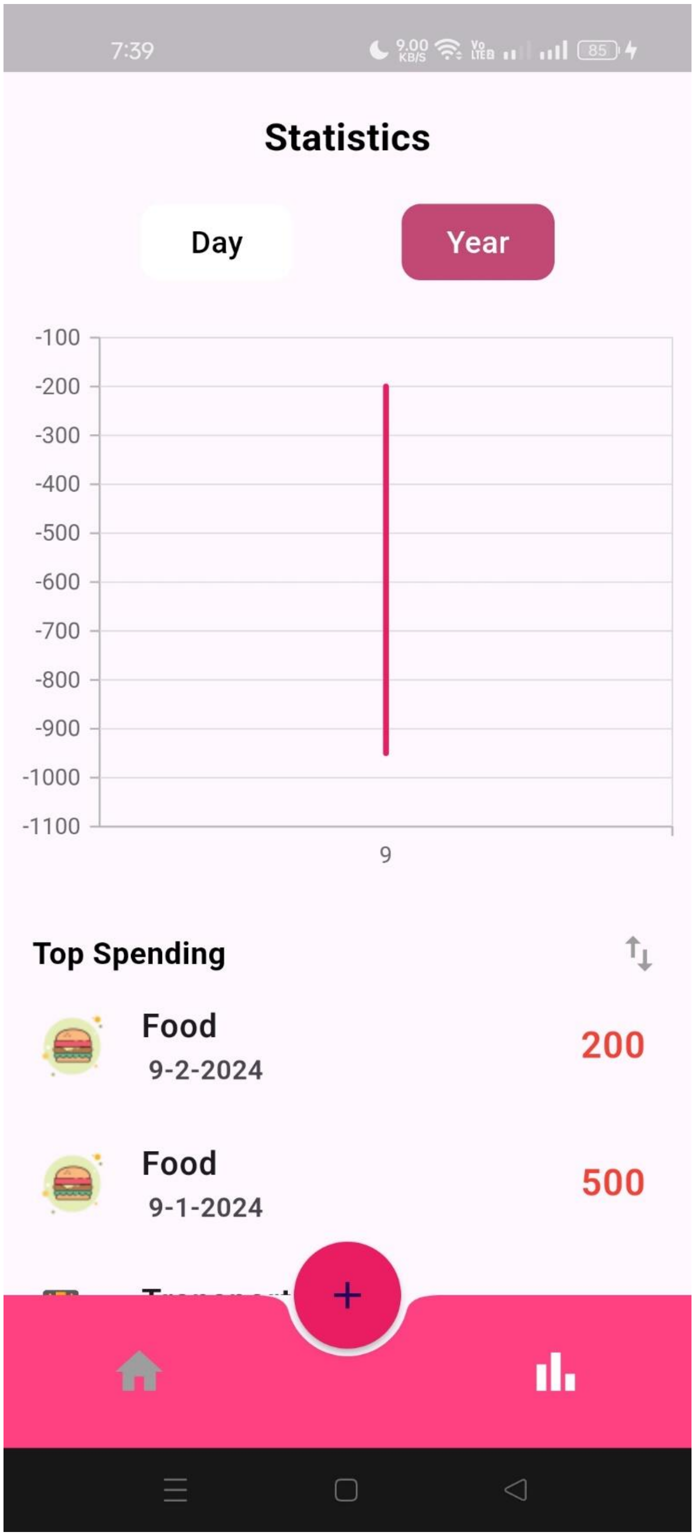
Statistics

Day

Year

Top Spending

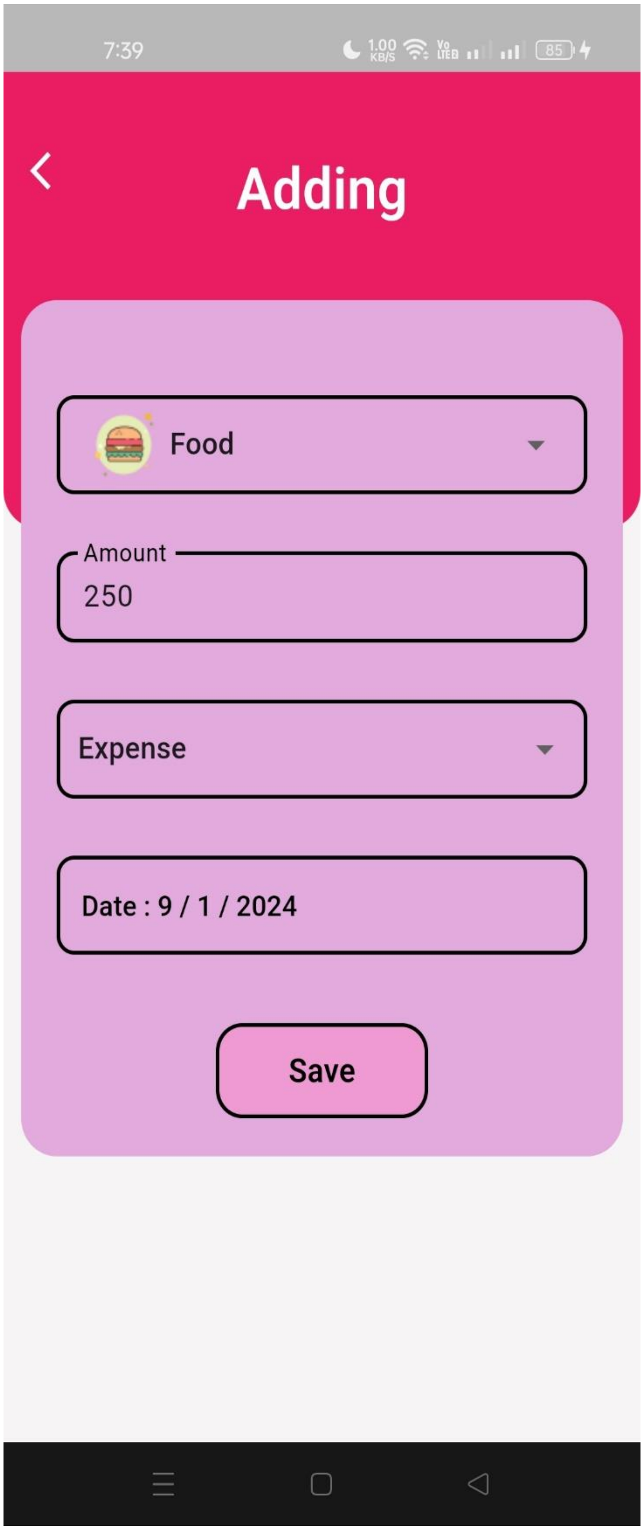
Food	9-2-2024	200
Food	9-1-2024	500



Statistics

Top Spending

Food 9-2-2024	200
Food 9-1-2024	500



Adding

Food

Amount: 250

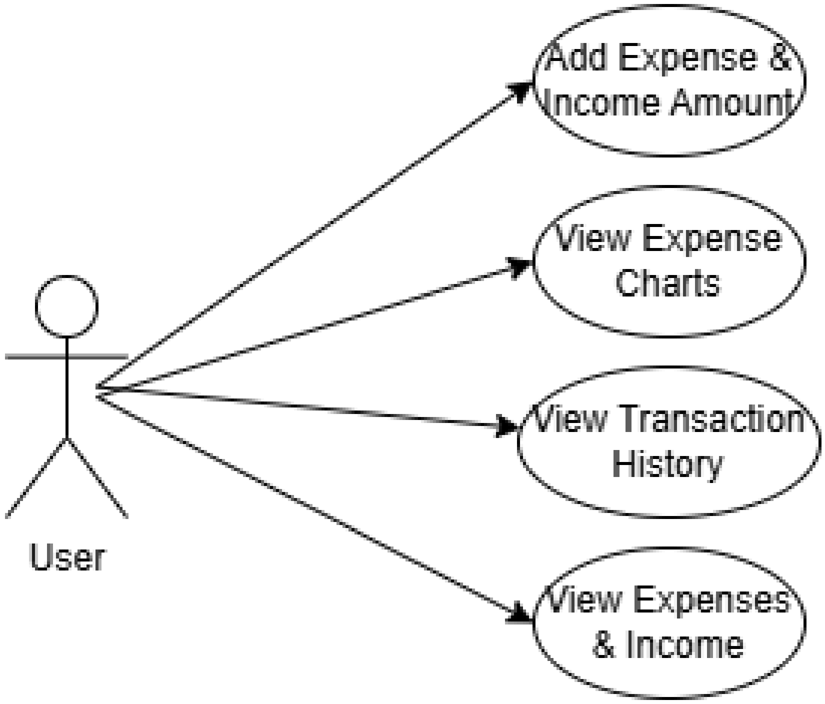
Expense

Date : 9 / 1 / 2024

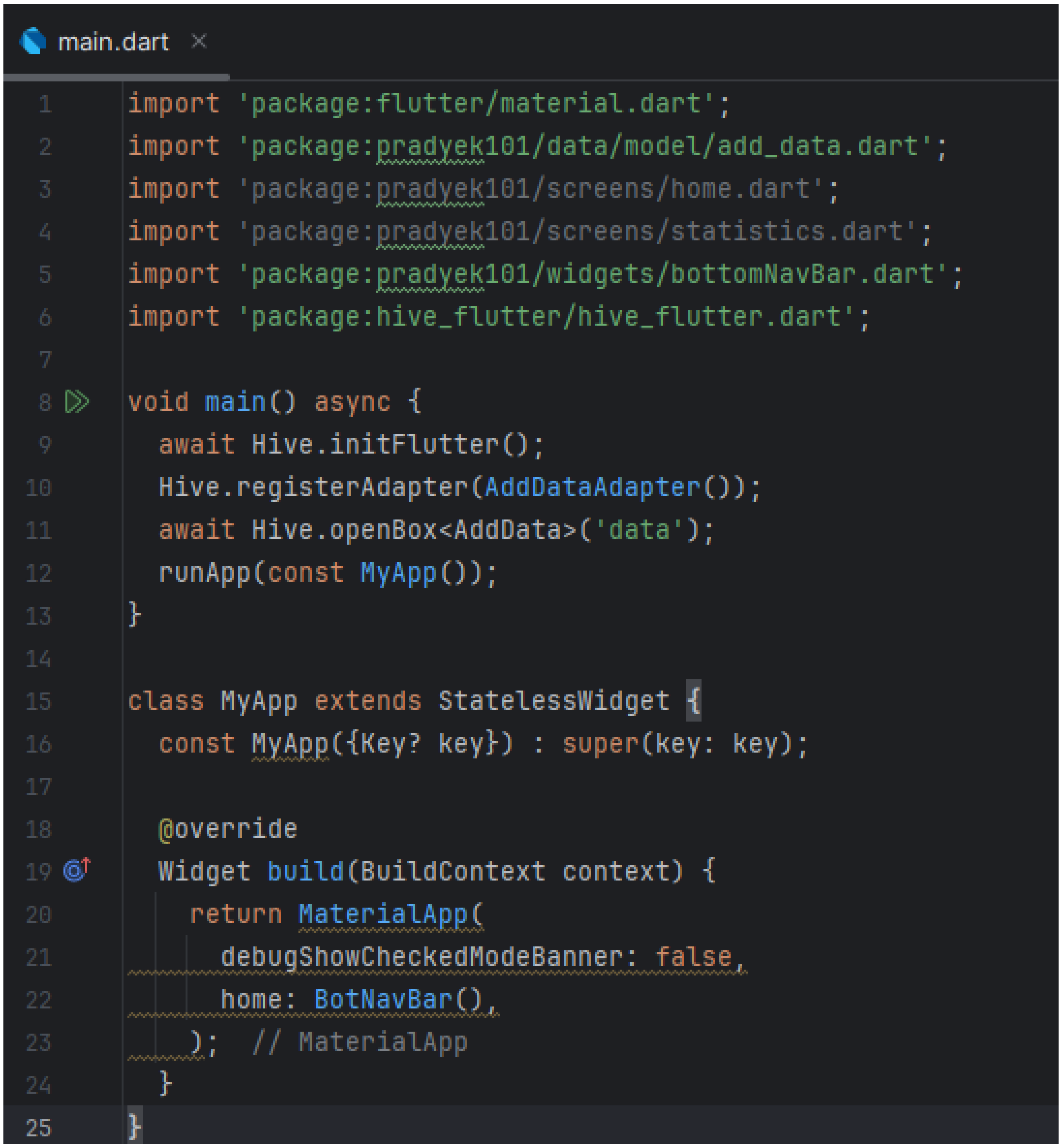
Functions of App

The app can track your expenses. By going to the income section, you can set or add your income balance. Then, if you go to the expense section, you can fill out a form detailing the type of expense you made, and it will deduct the amount of the expense from your balance. The app can also show your transaction history and track your expenses with daily or yearly charts.

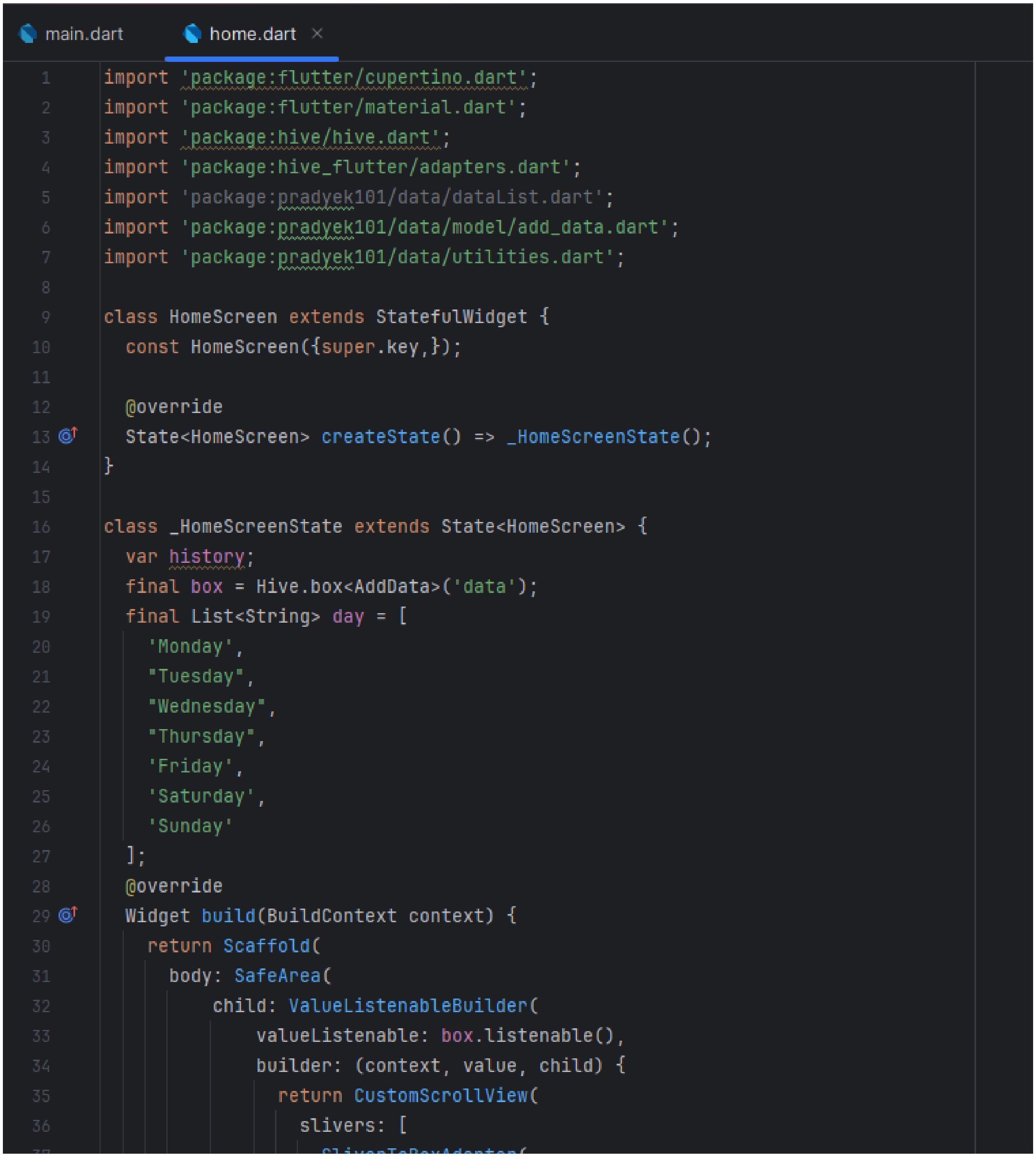
Use-case of the App



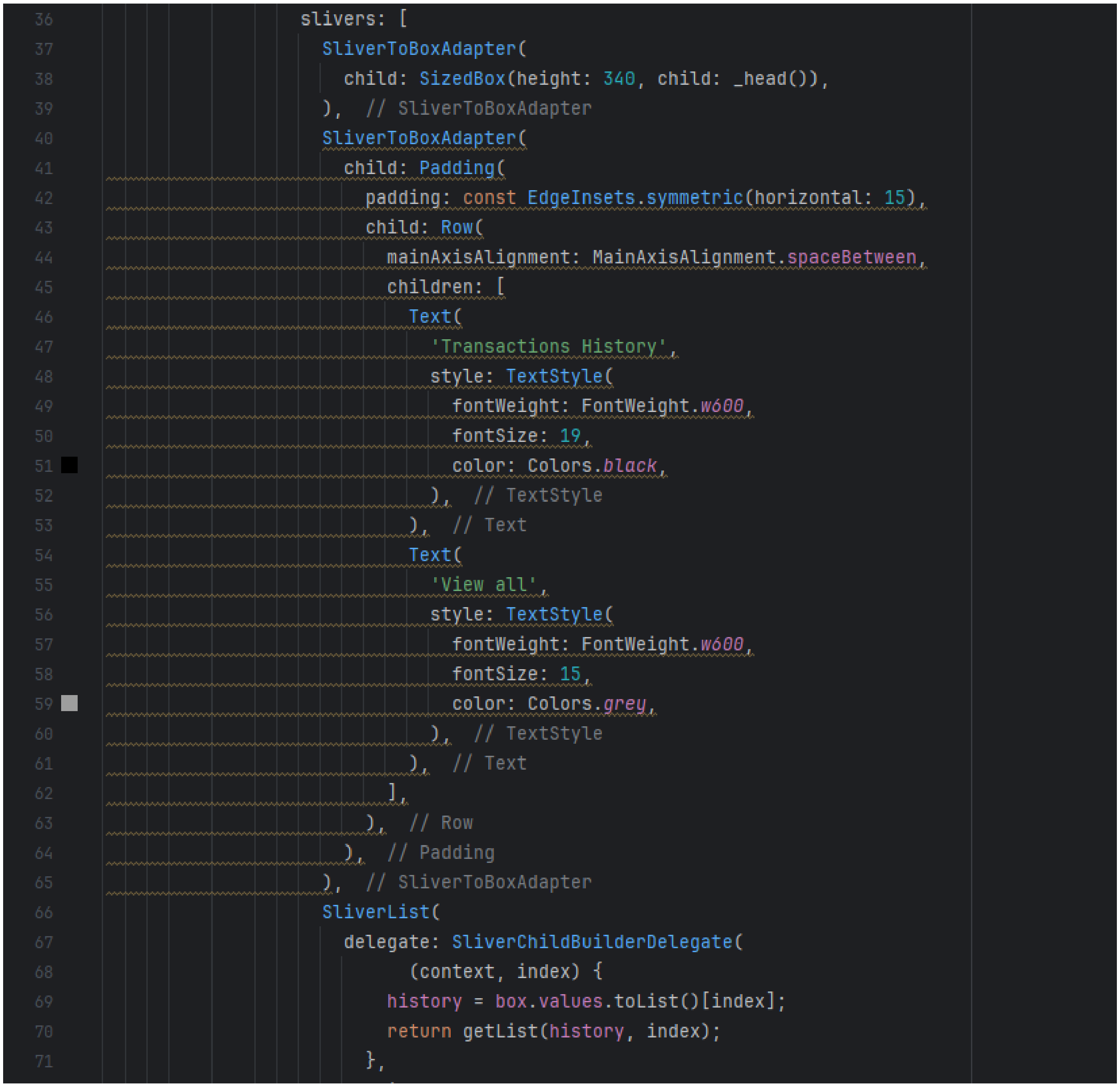
Source Code



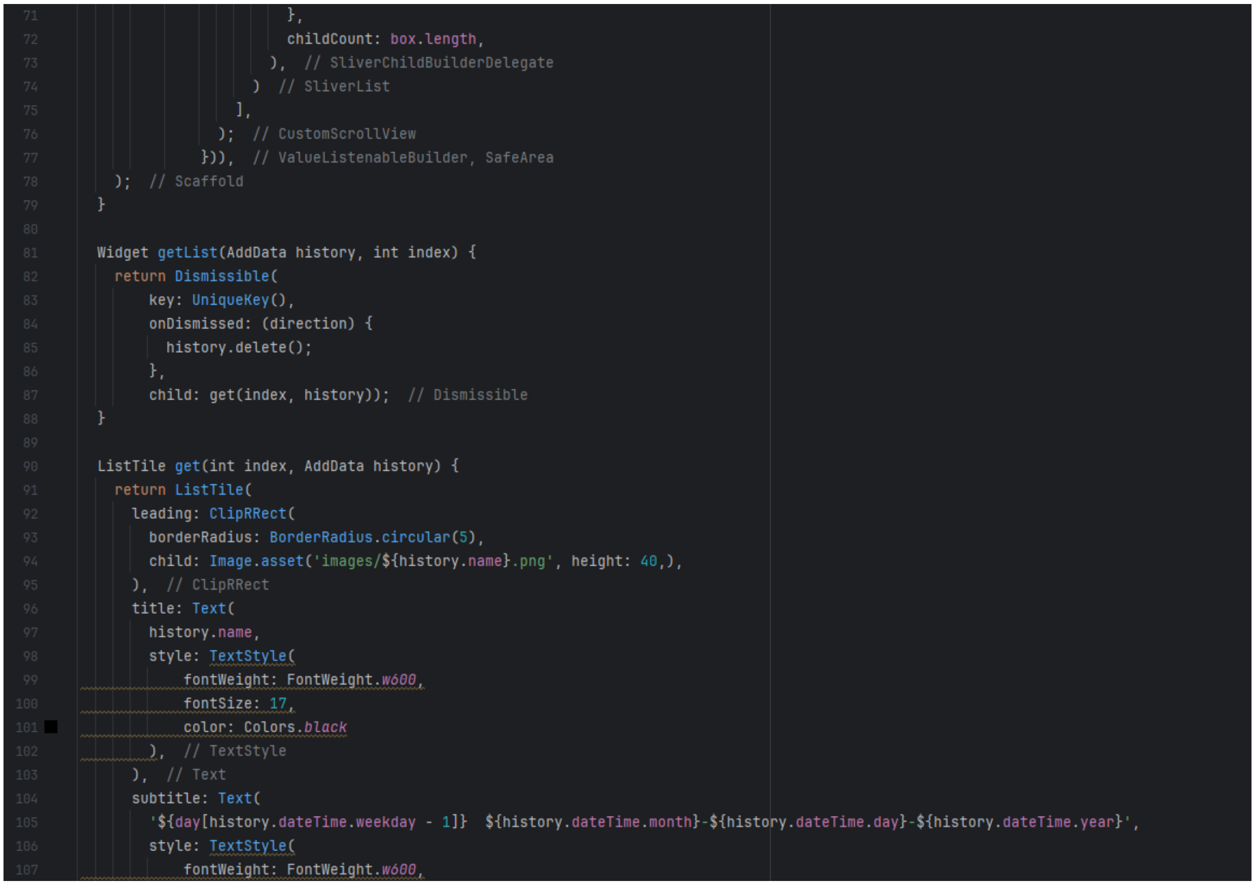
```dart
import 'package:flutter/material.dart';
import 'package:pradyek101/data/model/add_data.dart';
import 'package:pradyek101/screens/home.dart';
import 'package:pradyek101/screens/statistics.dart';
import 'package:pradyek101/widgets/bottomNavBar.dart';
import 'package:hive_flutter/hive_flutter.dart';

void main() async {
  await Hive.initFlutter();
  Hive.registerAdapter(AddDataAdapter());
  await Hive.openBox<AddData>('data');
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: BotNavBar(),
    ); // MaterialApp
  }
}
```



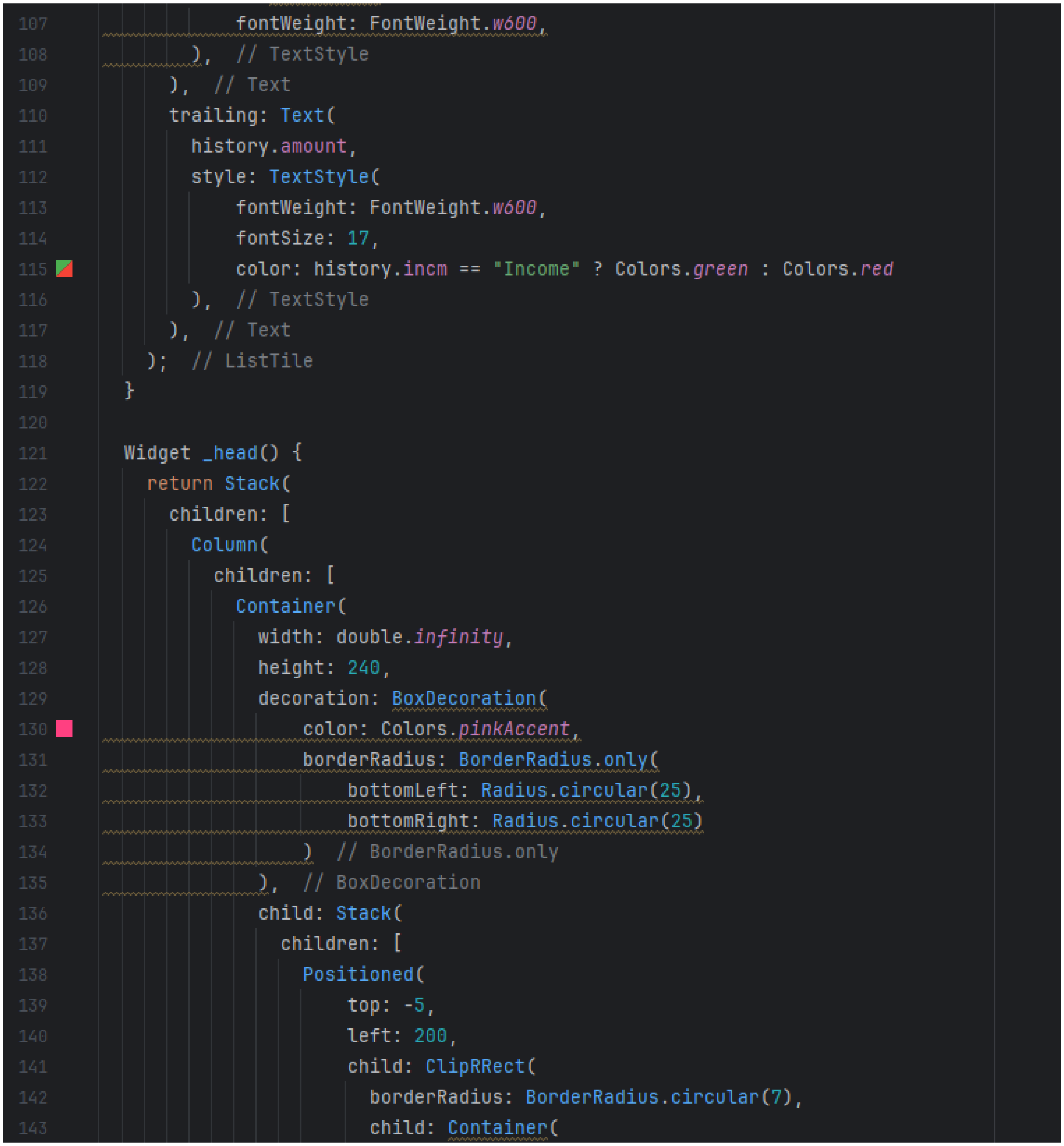
```dart
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:hive/hive.dart';
import 'package:hive_flutter/adapters.dart';
import 'package:pradyek101/data/dataList.dart';
import 'package:pradyek101/data/model/add_data.dart';
import 'package:pradyek101/data/utilities.dart';

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

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

class _HomeScreenState extends State<HomeScreen> {
  var history;
  final box = Hive.box<AddData>('data');
  final List<String> day = [
    'Monday',
    "Tuesday",
    "Wednesday",
    "Thursday",
    'Friday',
    'Saturday',
    'Sunday'
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
          child: ValueListenableBuilder(
              valueListenable: box.listenable(),
              builder: (context, value, child) {
                return CustomScrollView(
                  slivers: [
                    SliverToBoxAdapter(
```



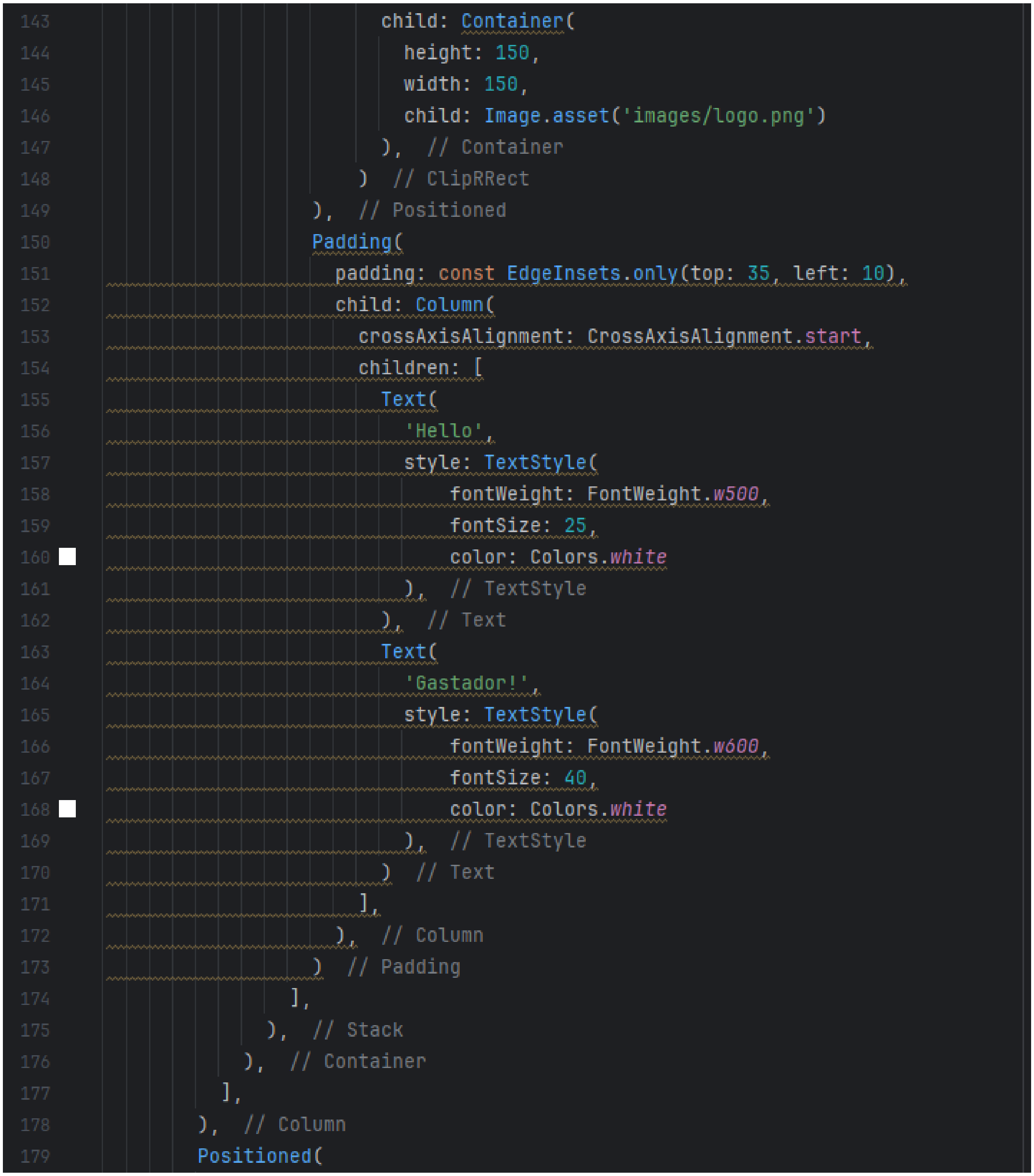
```
                  slivers: [
                    SliverToBoxAdapter(
                      child: SizedBox(height: 340, child: _head()),
                    ),  // SliverToBoxAdapter
                    SliverToBoxAdapter(
                      child: Padding(
                        padding: const EdgeInsets.symmetric(horizontal: 15),
                        child: Row(
                          mainAxisAlignment: MainAxisAlignment.spaceBetween,
                          children: [
                            Text(
                              'Transactions History',
                              style: TextStyle(
                                fontWeight: FontWeight.w600,
                                fontSize: 19,
                                color: Colors.black,
                              ),  // TextStyle
                            ),  // Text
                            Text(
                              'View all',
                              style: TextStyle(
                                fontWeight: FontWeight.w600,
                                fontSize: 15,
                                color: Colors.grey,
                              ),  // TextStyle
                            ),  // Text
                          ],
                        ),  // Row
                      ),  // Padding
                    ),  // SliverToBoxAdapter
                    SliverList(
                      delegate: SliverChildBuilderDelegate(
                            (context, index) {
                          history = box.values.toList()[index];
                          return getList(history, index);
                        },
```



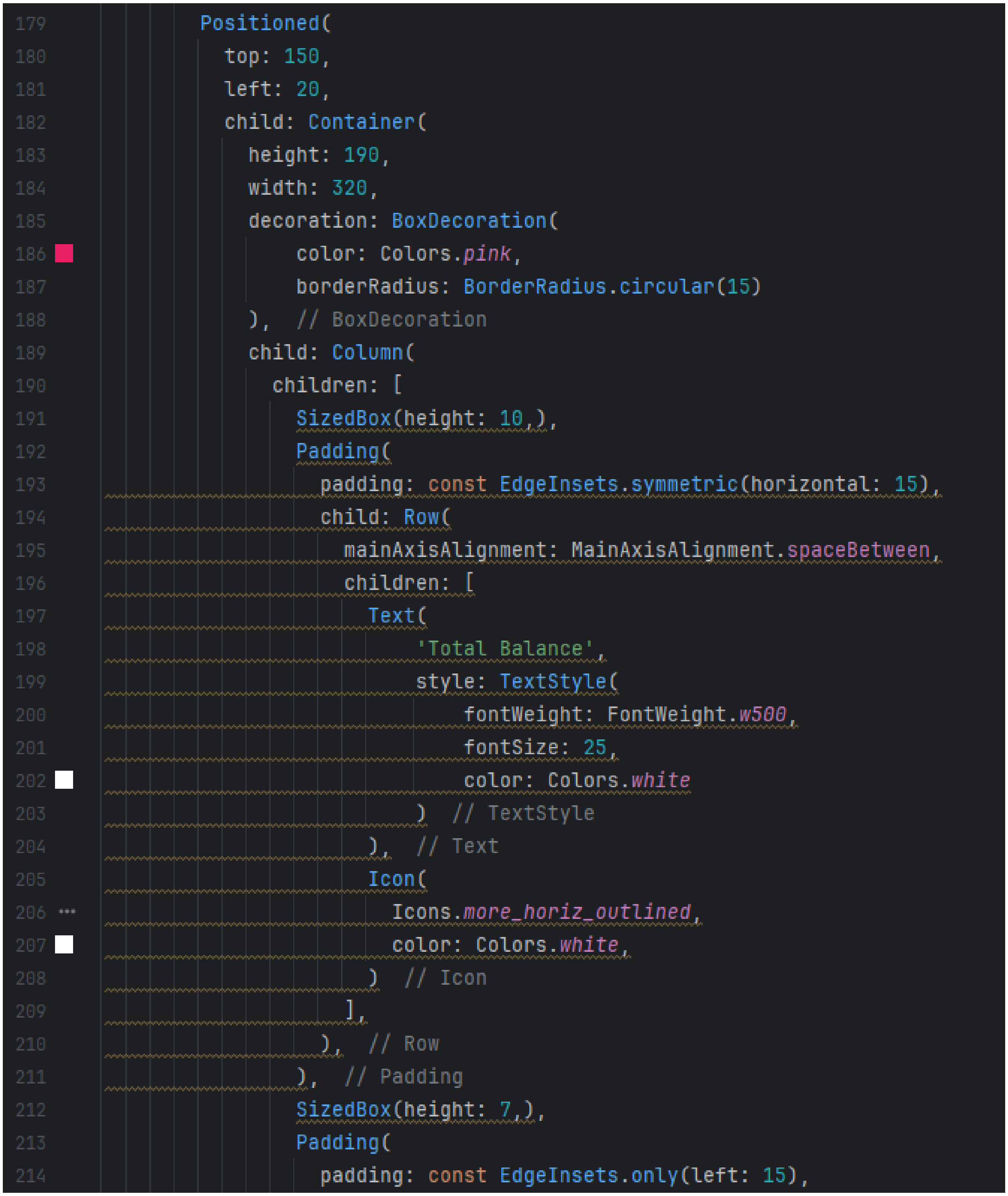
```
                      },
                      childCount: box.length,
                    ),  // SliverChildBuilderDelegate
                  )  // SliverList
                ],
              );  // CustomScrollView
            })),  // ValueListenableBuilder, SafeArea
  );  // Scaffold
}

Widget getList(AddData history, int index) {
  return Dismissible(
      key: UniqueKey(),
      onDismissed: (direction) {
        history.delete();
      },
      child: get(index, history));  // Dismissible
}

ListTile get(int index, AddData history) {
  return ListTile(
    leading: ClipRRect(
      borderRadius: BorderRadius.circular(5),
      child: Image.asset('images/${history.name}.png', height: 40,),
    ),  // ClipRRect
    title: Text(
      history.name,
      style: TextStyle(
          fontWeight: FontWeight.w600,
          fontSize: 17,
          color: Colors.black
      ),  // TextStyle
    ),  // Text
    subtitle: Text(
      '${day[history.dateTime.weekday - 1]}  ${history.dateTime.month}-${history.dateTime.day}-${history.dateTime.year}',
      style: TextStyle(
          fontWeight: FontWeight.w600,
```



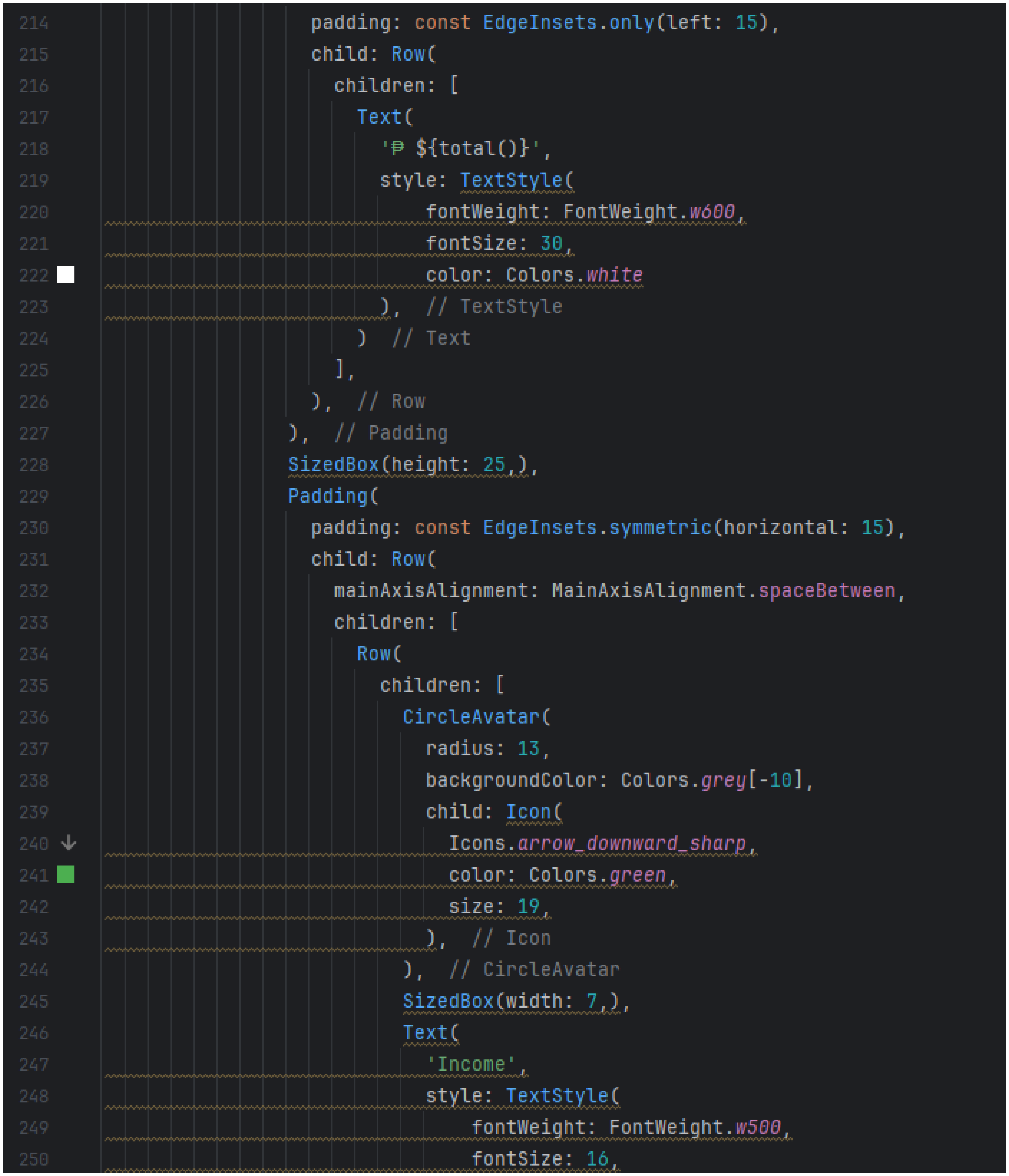
```
            fontWeight: FontWeight.w600,
        ),  // TextStyle
      ),  // Text
      trailing: Text(
        history.amount,
        style: TextStyle(
            fontWeight: FontWeight.w600,
            fontSize: 17,
            color: history.incm == "Income" ? Colors.green : Colors.red
        ),  // TextStyle
      ),  // Text
    );  // ListTile
  }

  Widget _head() {
    return Stack(
      children: [
        Column(
          children: [
            Container(
              width: double.infinity,
              height: 240,
              decoration: BoxDecoration(
                  color: Colors.pinkAccent,
                  borderRadius: BorderRadius.only(
                      bottomLeft: Radius.circular(25),
                      bottomRight: Radius.circular(25)
                  )  // BorderRadius.only
              ),  // BoxDecoration
              child: Stack(
                children: [
                  Positioned(
                      top: -5,
                      left: 200,
                      child: ClipRRect(
                        borderRadius: BorderRadius.circular(7),
                        child: Container(
```



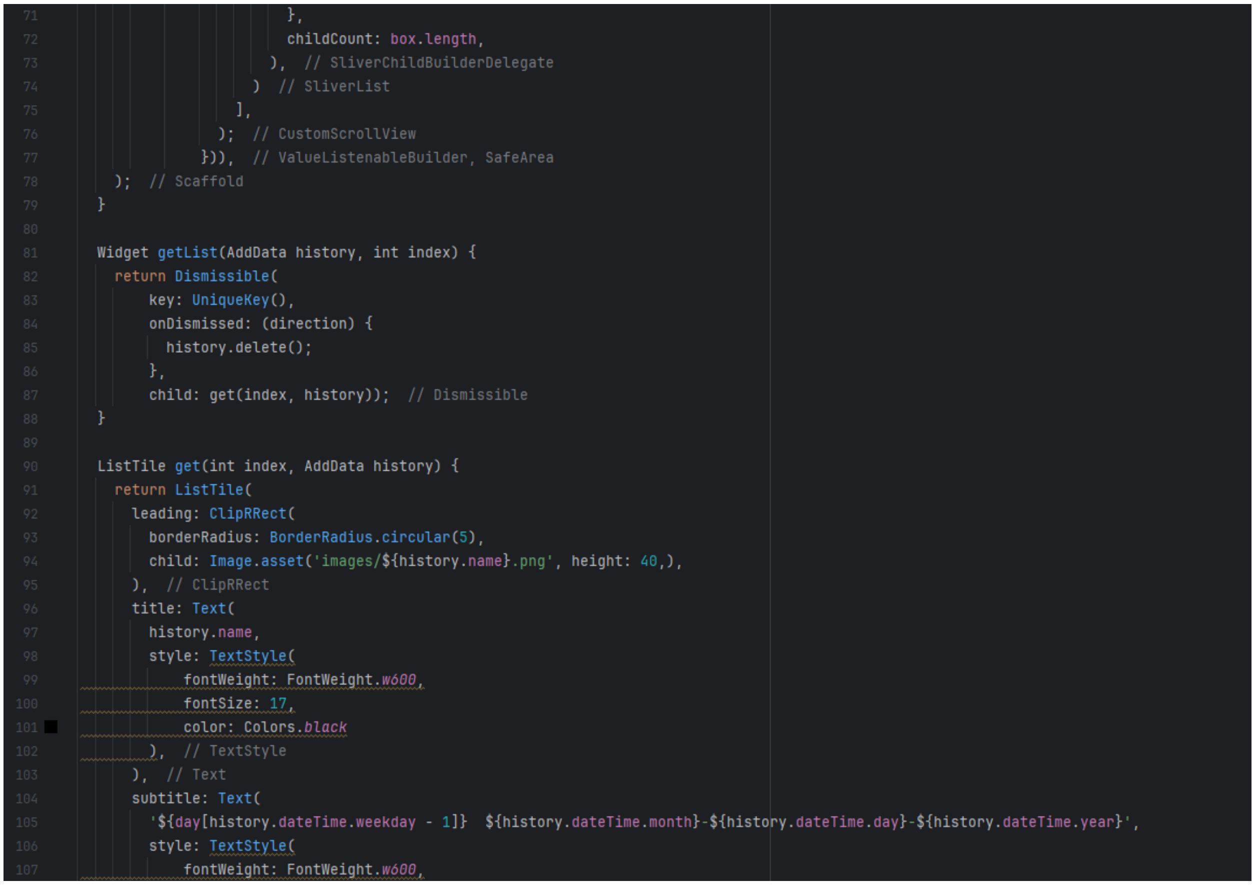
```dart
                child: Container(
                  height: 150,
                  width: 150,
                  child: Image.asset('images/logo.png')
                ),  // Container
              )  // ClipRRect
          ),  // Positioned
          Padding(
            padding: const EdgeInsets.only(top: 35, left: 10),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  'Hello',
                  style: TextStyle(
                      fontWeight: FontWeight.w500,
                      fontSize: 25,
                      color: Colors.white
                  ),  // TextStyle
                ),  // Text
                Text(
                  'Gastador!',
                  style: TextStyle(
                      fontWeight: FontWeight.w600,
                      fontSize: 40,
                      color: Colors.white
                  ),  // TextStyle
                )  // Text
              ],
            ),  // Column
          )  // Padding
        ],
      ),  // Stack
    ),  // Container
  ],
),  // Column
Positioned(
```



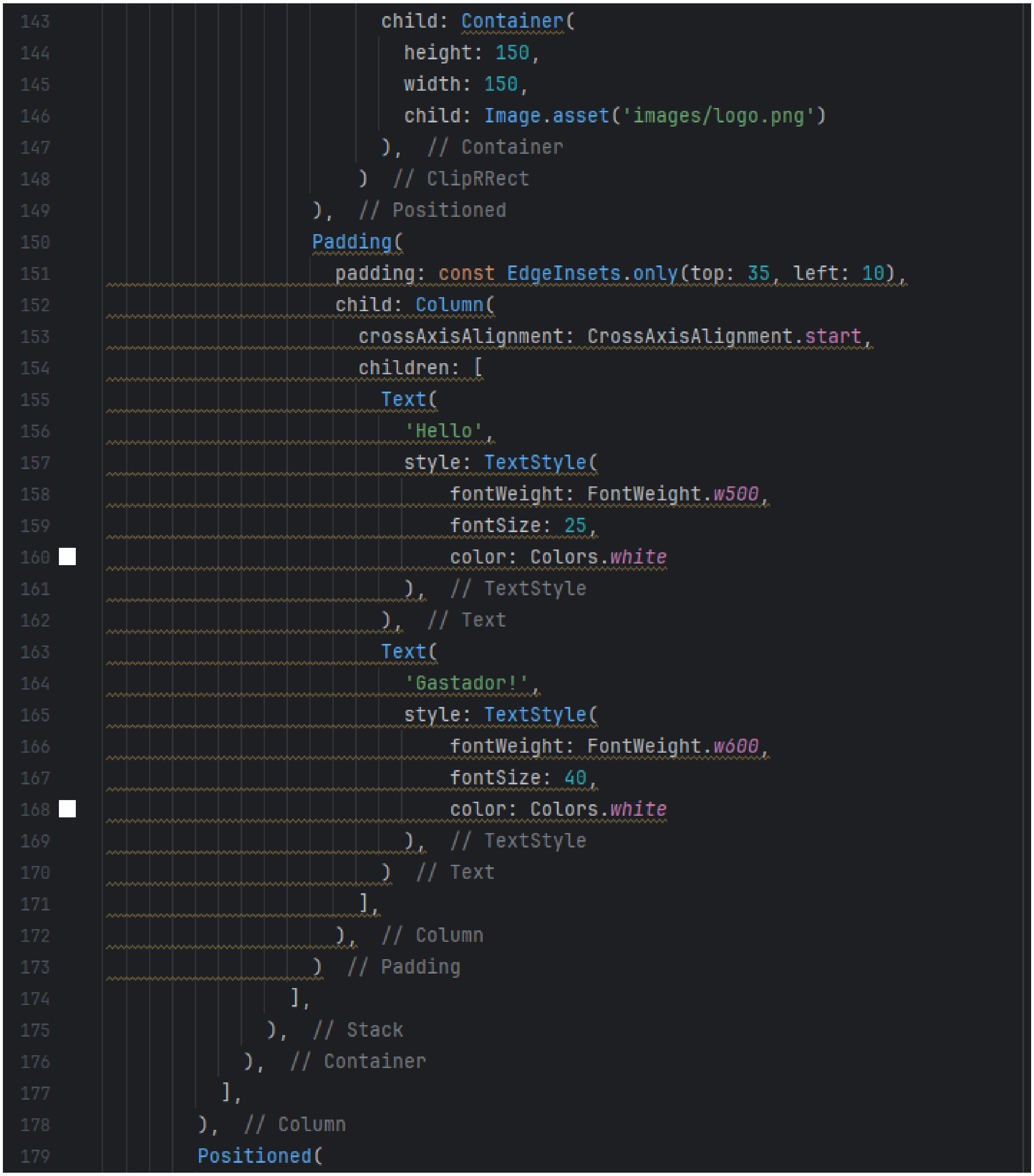
```
        Positioned(
          top: 150,
          left: 20,
          child: Container(
            height: 190,
            width: 320,
            decoration: BoxDecoration(
                color: Colors.pink,
                borderRadius: BorderRadius.circular(15)
            ),  // BoxDecoration
            child: Column(
              children: [
                SizedBox(height: 10,),
                Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 15),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      Text(
                          'Total Balance',
                          style: TextStyle(
                              fontWeight: FontWeight.w500,
                              fontSize: 25,
                              color: Colors.white
                          )  // TextStyle
                      ),  // Text
                      Icon(
                        Icons.more_horiz_outlined,
                        color: Colors.white,
                      )  // Icon
                    ],
                  ),  // Row
                ),  // Padding
                SizedBox(height: 7,),
                Padding(
                  padding: const EdgeInsets.only(left: 15),
```



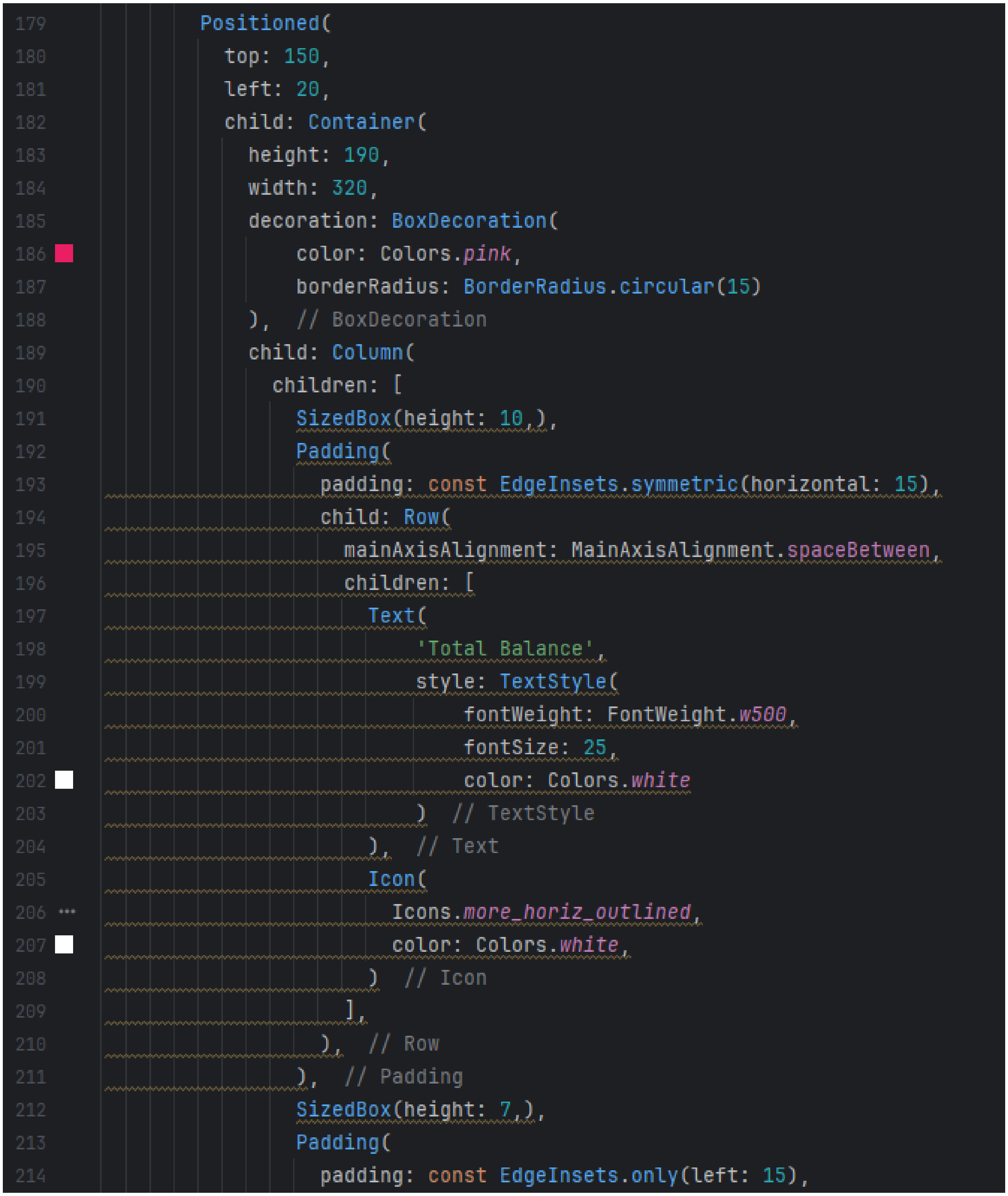
```
                  padding: const EdgeInsets.only(left: 15),
                  child: Row(
                    children: [
                      Text(
                        '₱ ${total()}',
                        style: TextStyle(
                            fontWeight: FontWeight.w600,
                            fontSize: 30,
                            color: Colors.white
                        ),  // TextStyle
                      )  // Text
                    ],
                  ),  // Row
                ),  // Padding
                SizedBox(height: 25,),
                Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 15),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      Row(
                        children: [
                          CircleAvatar(
                            radius: 13,
                            backgroundColor: Colors.grey[-10],
                            child: Icon(
                              Icons.arrow_downward_sharp,
                              color: Colors.green,
                              size: 19,
                            ),  // Icon
                          ),  // CircleAvatar
                          SizedBox(width: 7,),
                          Text(
                            'Income',
                            style: TextStyle(
                                fontWeight: FontWeight.w500,
                                fontSize: 16,
```



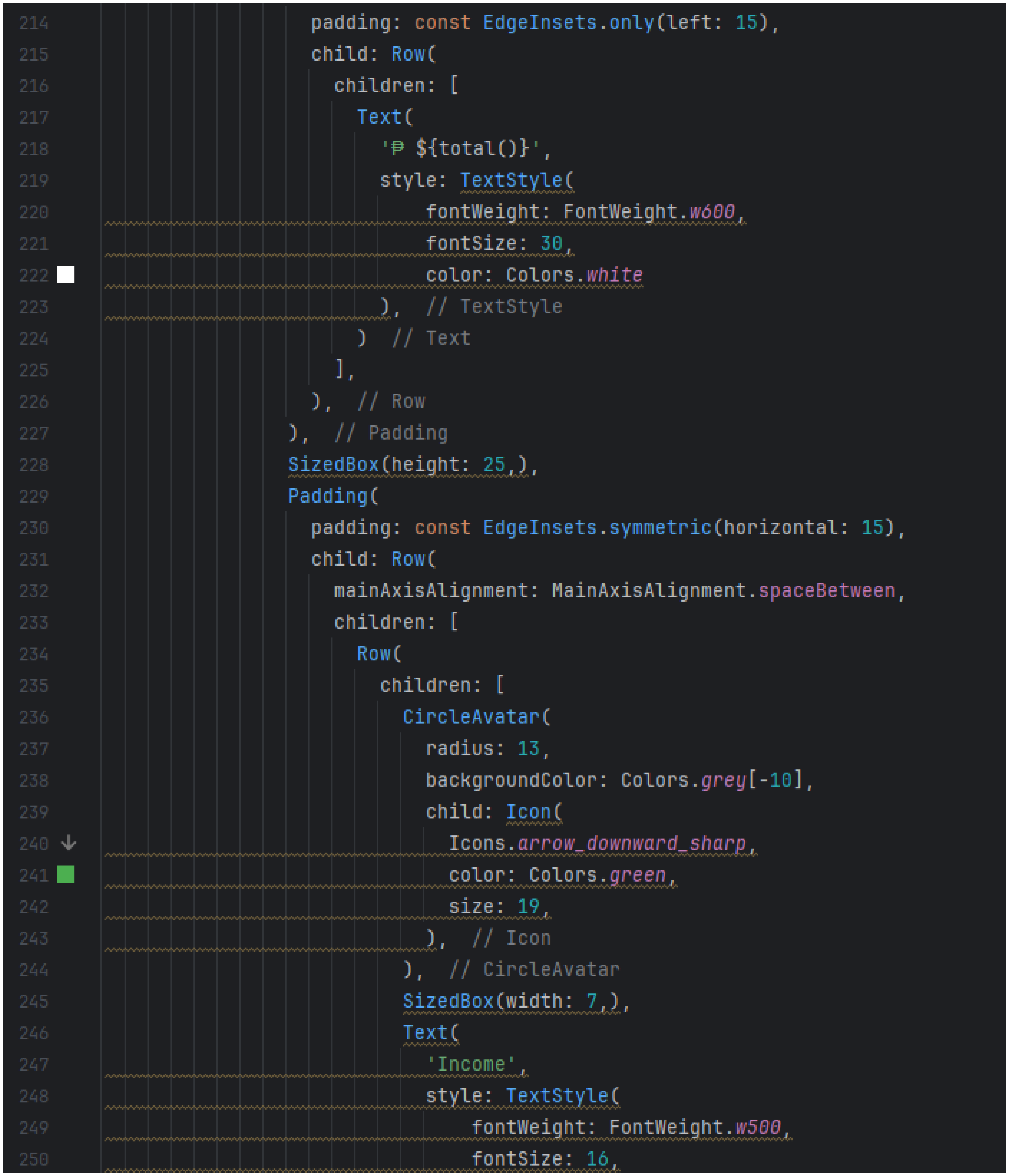
```dart
                      },
                      childCount: box.length,
                    ),  // SliverChildBuilderDelegate
                  )  // SliverList
                ],
              );  // CustomScrollView
            })),  // ValueListenableBuilder, SafeArea
  );  // Scaffold
}

Widget getList(AddData history, int index) {
  return Dismissible(
      key: UniqueKey(),
      onDismissed: (direction) {
        history.delete();
      },
      child: get(index, history));  // Dismissible
}

ListTile get(int index, AddData history) {
  return ListTile(
    leading: ClipRRect(
      borderRadius: BorderRadius.circular(5),
      child: Image.asset('images/${history.name}.png', height: 40,),
    ),  // ClipRRect
    title: Text(
      history.name,
      style: TextStyle(
          fontWeight: FontWeight.w600,
          fontSize: 17,
          color: Colors.black
      ),  // TextStyle
    ),  // Text
    subtitle: Text(
      '${day[history.dateTime.weekday - 1]}  ${history.dateTime.month}-${history.dateTime.day}-${history.dateTime.year}',
      style: TextStyle(
          fontWeight: FontWeight.w600,
```



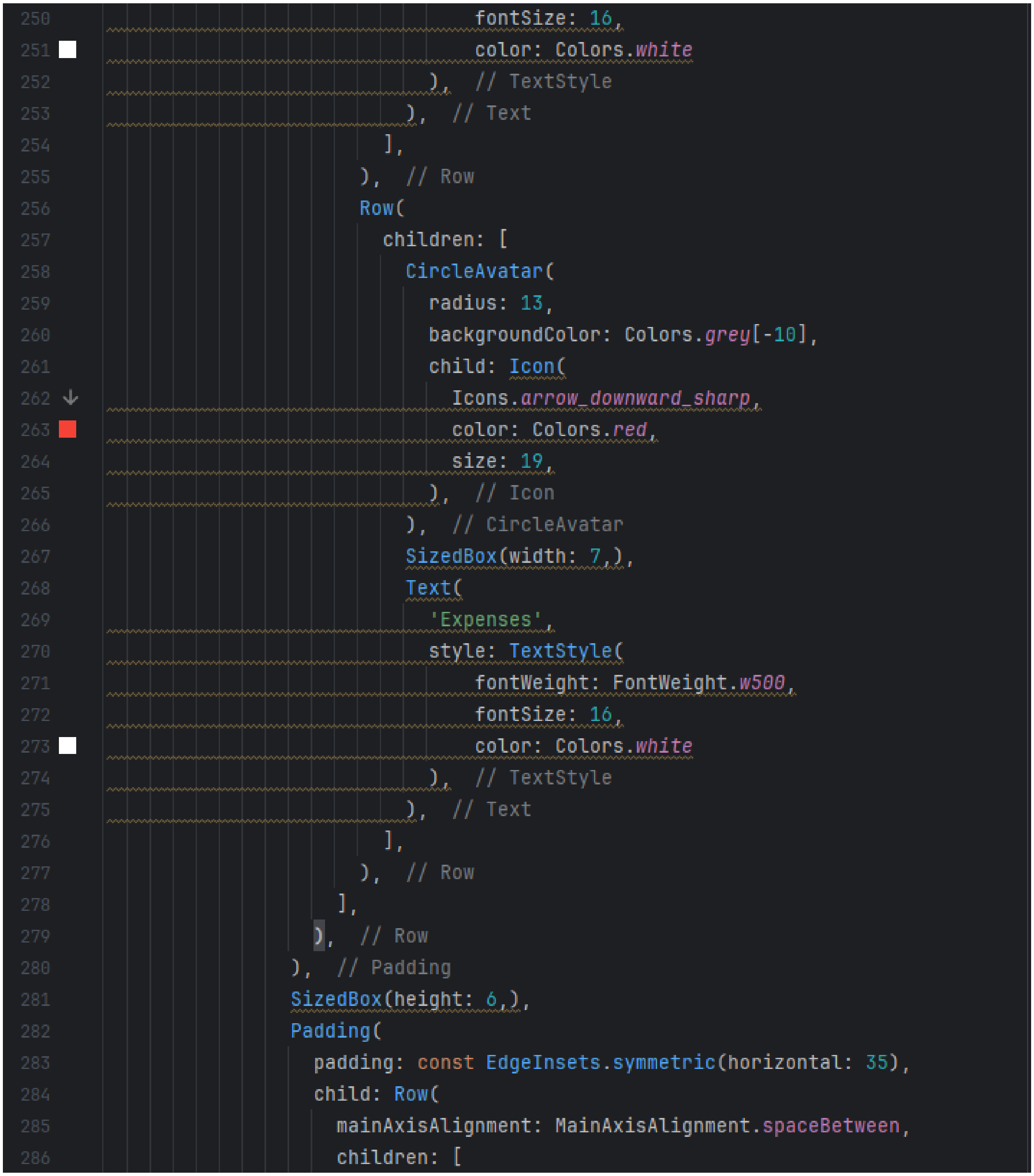
```
                        child: Container(
                          height: 150,
                          width: 150,
                          child: Image.asset('images/logo.png')
                        ),  // Container
                      )  // ClipRRect
                  ),  // Positioned
                  Padding(
                    padding: const EdgeInsets.only(top: 35, left: 10),
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Text(
                          'Hello',
                          style: TextStyle(
                              fontWeight: FontWeight.w500,
                              fontSize: 25,
                              color: Colors.white
                          ),  // TextStyle
                        ),  // Text
                        Text(
                          'Gastador!',
                          style: TextStyle(
                              fontWeight: FontWeight.w600,
                              fontSize: 40,
                              color: Colors.white
                          ),  // TextStyle
                        )  // Text
                      ],
                    ),  // Column
                  )  // Padding
                ],
              ),  // Stack
            ),  // Container
          ],
        ),  // Column
        Positioned(
```



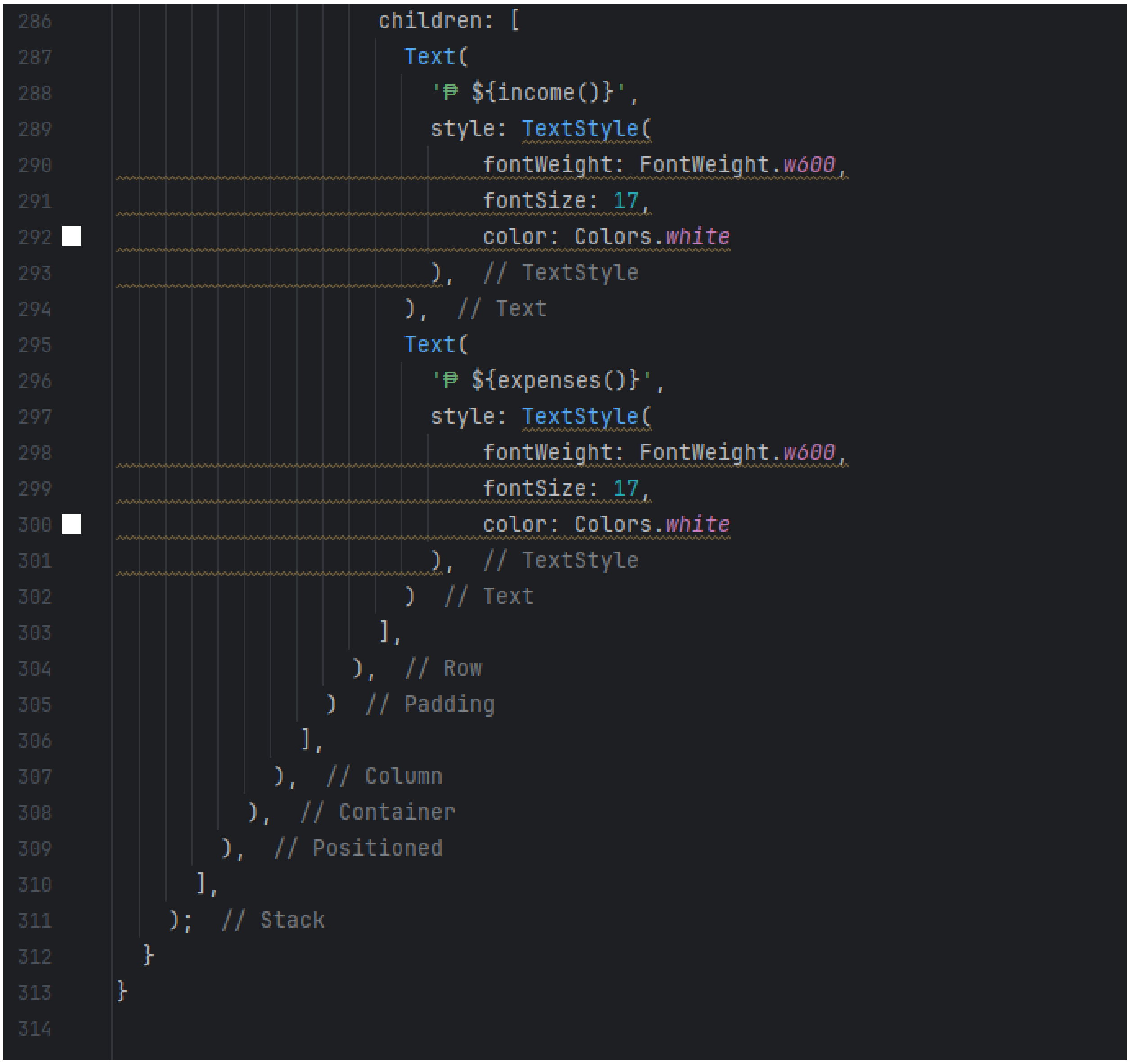
```dart
Positioned(
  top: 150,
  left: 20,
  child: Container(
    height: 190,
    width: 320,
    decoration: BoxDecoration(
        color: Colors.pink,
        borderRadius: BorderRadius.circular(15)
    ),  // BoxDecoration
    child: Column(
      children: [
        SizedBox(height: 10,),
        Padding(
          padding: const EdgeInsets.symmetric(horizontal: 15),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              Text(
                  'Total Balance',
                  style: TextStyle(
                      fontWeight: FontWeight.w500,
                      fontSize: 25,
                      color: Colors.white
                  )  // TextStyle
              ),  // Text
              Icon(
                Icons.more_horiz_outlined,
                color: Colors.white,
              )  // Icon
            ],
          ),  // Row
        ),  // Padding
        SizedBox(height: 7,),
        Padding(
          padding: const EdgeInsets.only(left: 15),
```



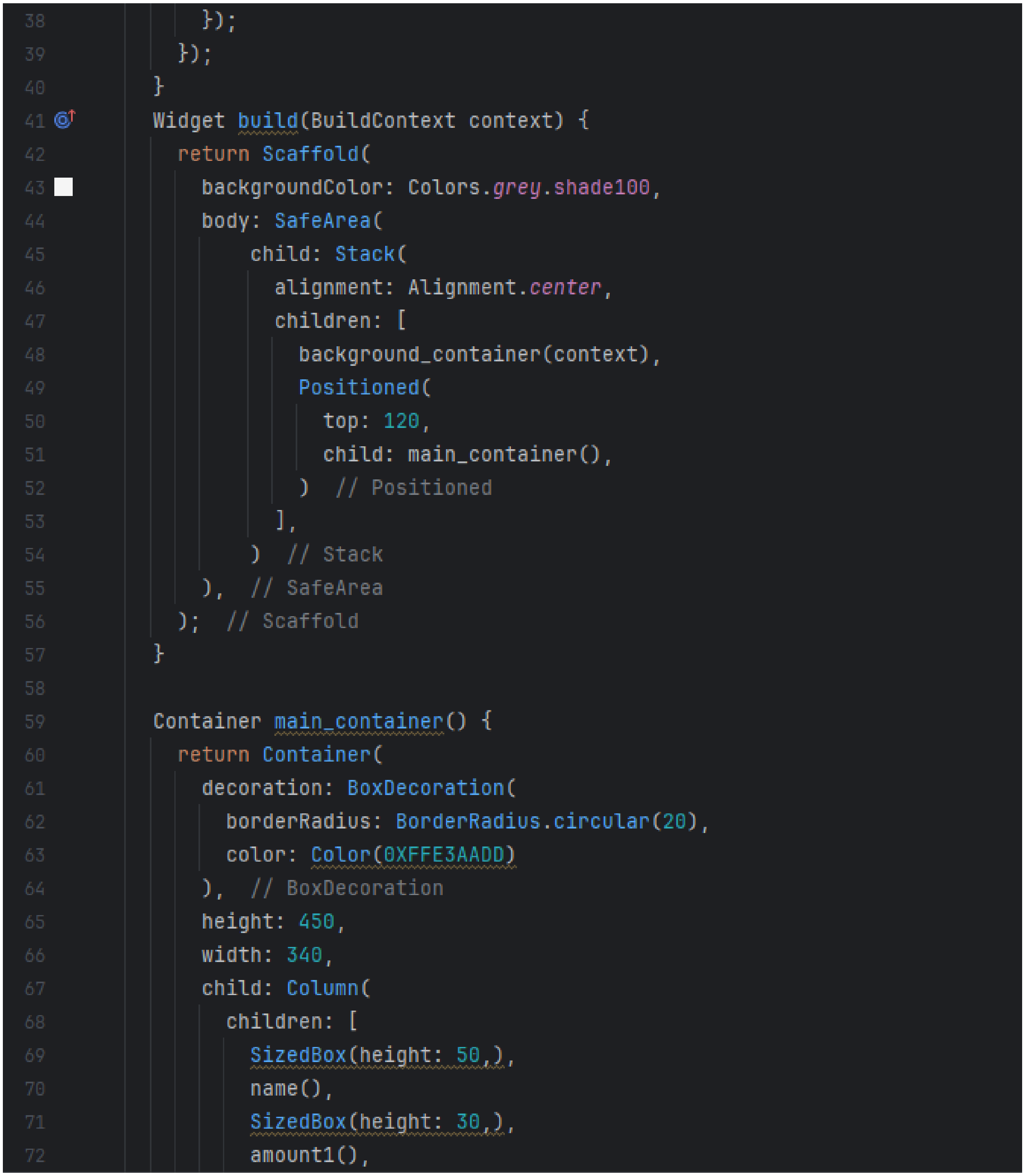
```
                  padding: const EdgeInsets.only(left: 15),
                  child: Row(
                    children: [
                      Text(
                        '₱ ${total()}',
                        style: TextStyle(
                            fontWeight: FontWeight.w600,
                            fontSize: 30,
                            color: Colors.white
                        ),  // TextStyle
                      )  // Text
                    ],
                  ),  // Row
                ),  // Padding
                SizedBox(height: 25,),
                Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 15),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      Row(
                        children: [
                          CircleAvatar(
                            radius: 13,
                            backgroundColor: Colors.grey[-10],
                            child: Icon(
                              Icons.arrow_downward_sharp,
                              color: Colors.green,
                              size: 19,
                            ),  // Icon
                          ),  // CircleAvatar
                          SizedBox(width: 7,),
                          Text(
                            'Income',
                            style: TextStyle(
                                fontWeight: FontWeight.w500,
                                fontSize: 16,
```



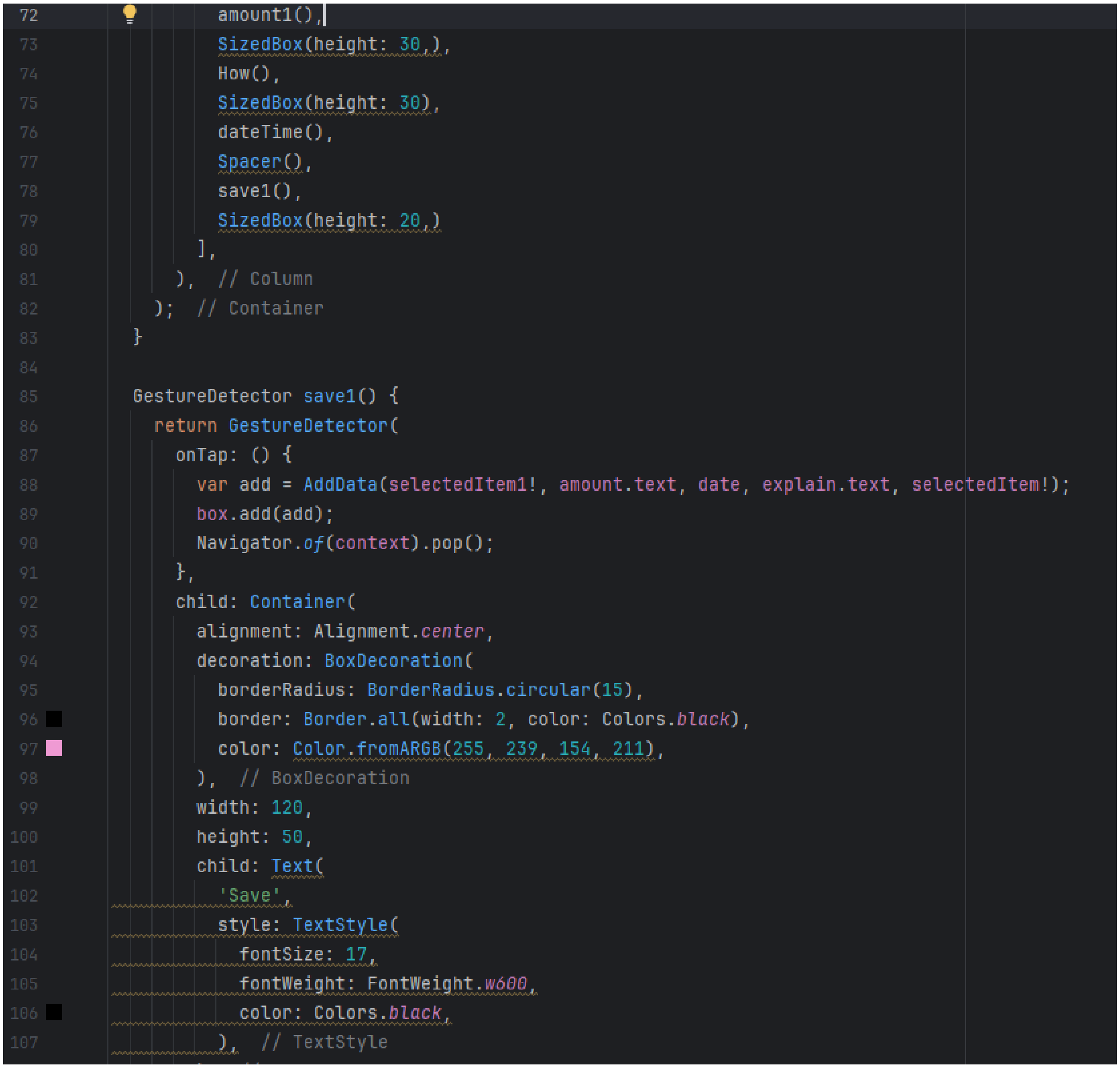
```
                                fontSize: 16,
                                color: Colors.white
                            ),  // TextStyle
                          ),  // Text
                        ],
                      ),  // Row
                      Row(
                        children: [
                          CircleAvatar(
                            radius: 13,
                            backgroundColor: Colors.grey[-10],
                            child: Icon(
                              Icons.arrow_downward_sharp,
                              color: Colors.red,
                              size: 19,
                            ),  // Icon
                          ),  // CircleAvatar
                          SizedBox(width: 7,),
                          Text(
                            'Expenses',
                            style: TextStyle(
                                fontWeight: FontWeight.w500,
                                fontSize: 16,
                                color: Colors.white
                            ),  // TextStyle
                          ),  // Text
                        ],
                      ),  // Row
                    ],
                  ),  // Row
                ),  // Padding
                SizedBox(height: 6,),
                Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 35),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
```



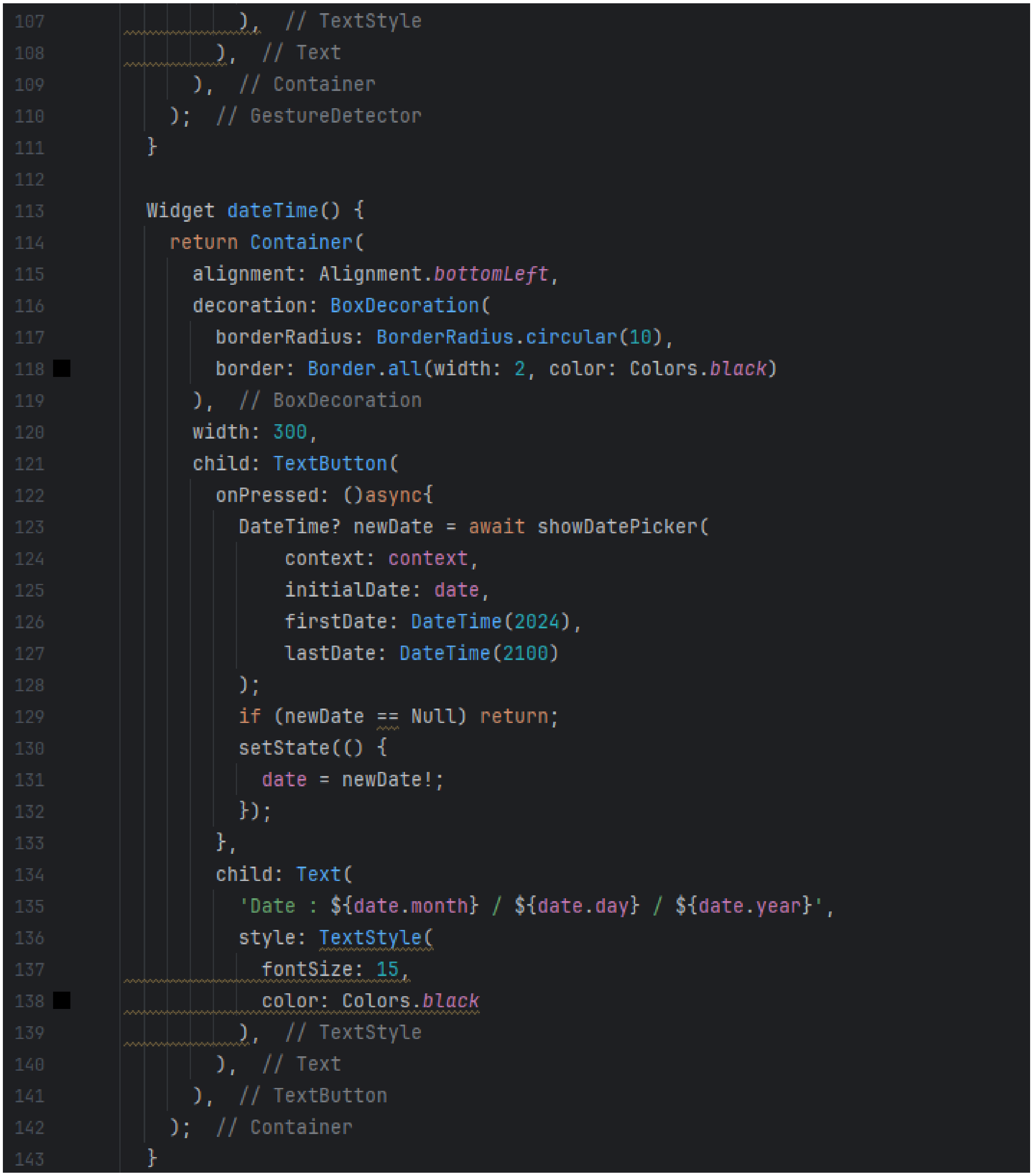
```dart
                    children: [
                      Text(
                        '₱ ${income()}',
                        style: TextStyle(
                            fontWeight: FontWeight.w600,
                            fontSize: 17,
                            color: Colors.white
                        ),  // TextStyle
                      ),  // Text
                      Text(
                        '₱ ${expenses()}',
                        style: TextStyle(
                            fontWeight: FontWeight.w600,
                            fontSize: 17,
                            color: Colors.white
                        ),  // TextStyle
                      )  // Text
                    ],
                  ),  // Row
                )  // Padding
              ],
            ),  // Column
          ),  // Container
        ),  // Positioned
      ],
    );  // Stack
  }
}
```



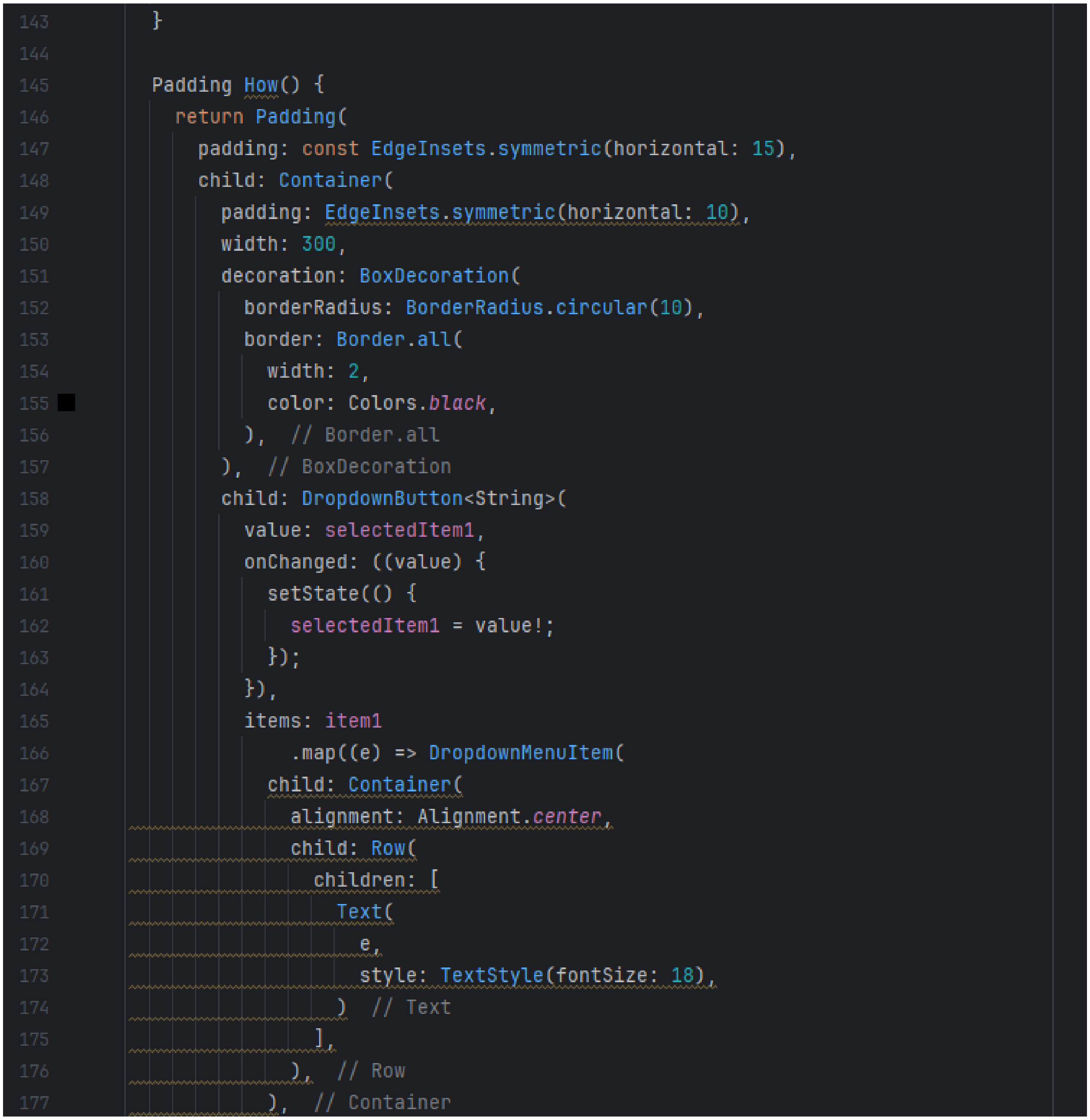
```
      });
    });
  }
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey.shade100,
      body: SafeArea(
          child: Stack(
            alignment: Alignment.center,
            children: [
              background_container(context),
              Positioned(
                top: 120,
                child: main_container(),
              )  // Positioned
            ],
          )  // Stack
      ),  // SafeArea
    );  // Scaffold
  }

  Container main_container() {
    return Container(
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        color: Color(0XFFE3AADD)
      ),  // BoxDecoration
      height: 450,
      width: 340,
      child: Column(
        children: [
          SizedBox(height: 50,),
          name(),
          SizedBox(height: 30,),
          amount1(),
```



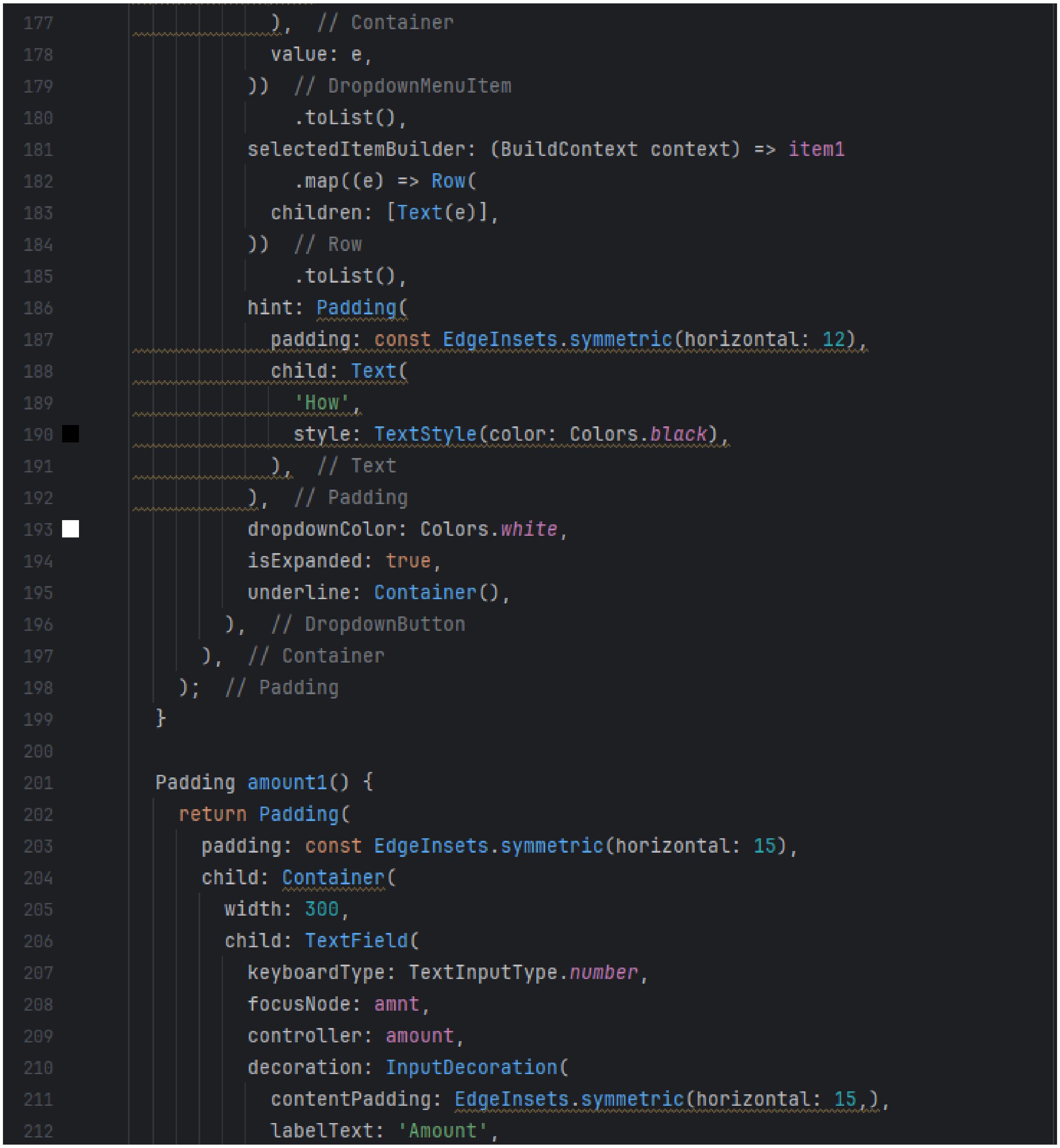
```dart
          amount1(),
          SizedBox(height: 30,),
          How(),
          SizedBox(height: 30),
          dateTime(),
          Spacer(),
          save1(),
          SizedBox(height: 20,)
        ],
      ),  // Column
    );  // Container
  }

  GestureDetector save1() {
    return GestureDetector(
      onTap: () {
        var add = AddData(selectedItem1!, amount.text, date, explain.text, selectedItem!);
        box.add(add);
        Navigator.of(context).pop();
      },
      child: Container(
        alignment: Alignment.center,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(15),
          border: Border.all(width: 2, color: Colors.black),
          color: Color.fromARGB(255, 239, 154, 211),
        ),  // BoxDecoration
        width: 120,
        height: 50,
        child: Text(
          'Save',
          style: TextStyle(
            fontSize: 17,
            fontWeight: FontWeight.w600,
            color: Colors.black,
          ),  // TextStyle
```



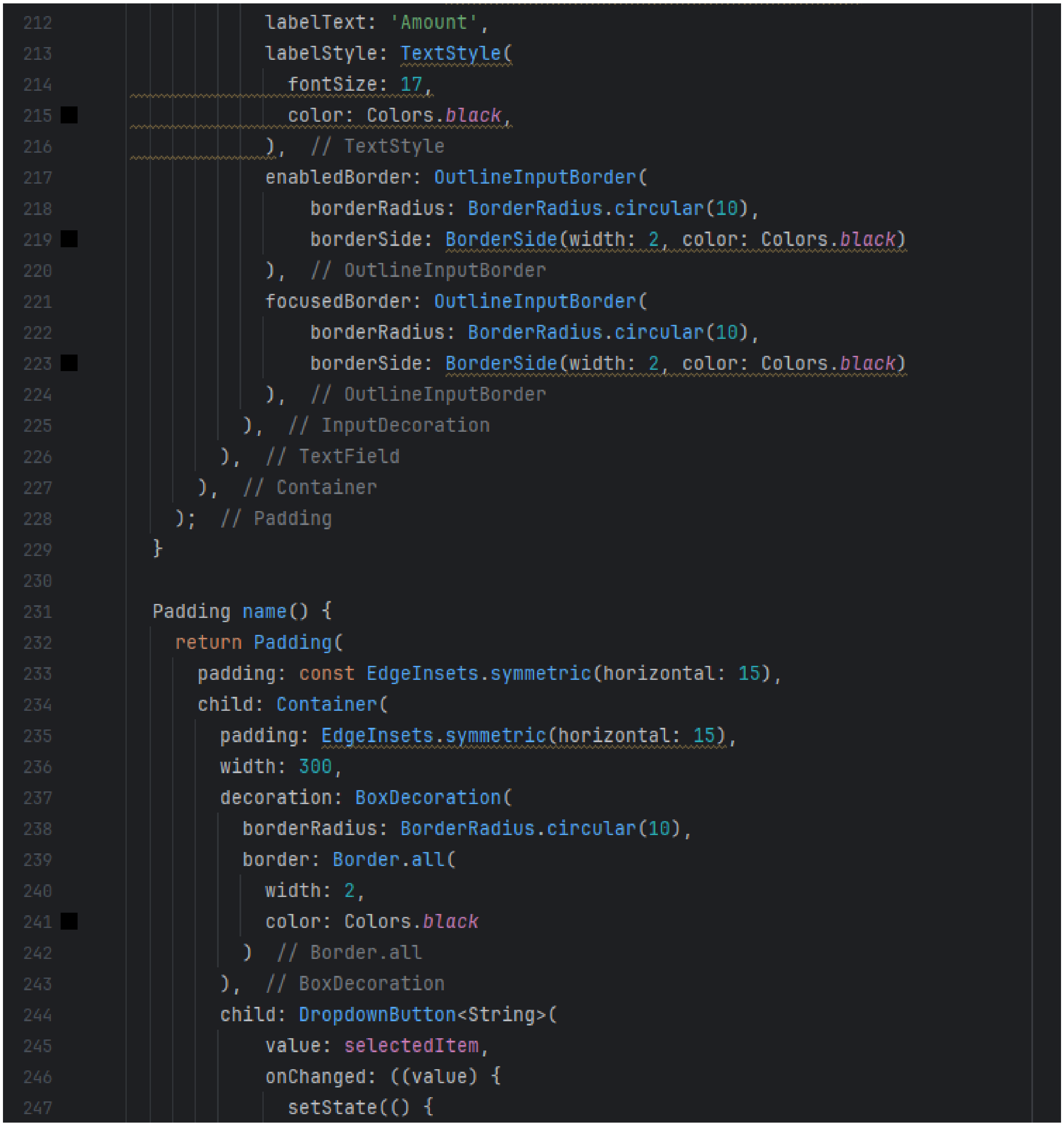
```
        ),  // TextStyle
      ),  // Text
    ),  // Container
  );  // GestureDetector
}

Widget dateTime() {
  return Container(
    alignment: Alignment.bottomLeft,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      border: Border.all(width: 2, color: Colors.black)
    ),  // BoxDecoration
    width: 300,
    child: TextButton(
      onPressed: ()async{
        DateTime? newDate = await showDatePicker(
            context: context,
            initialDate: date,
            firstDate: DateTime(2024),
            lastDate: DateTime(2100)
        );
        if (newDate == Null) return;
        setState(() {
          date = newDate!;
        });
      },
      child: Text(
        'Date : ${date.month} / ${date.day} / ${date.year}',
        style: TextStyle(
          fontSize: 15,
          color: Colors.black
        ),  // TextStyle
      ),  // Text
    ),  // TextButton
  );  // Container
}
```



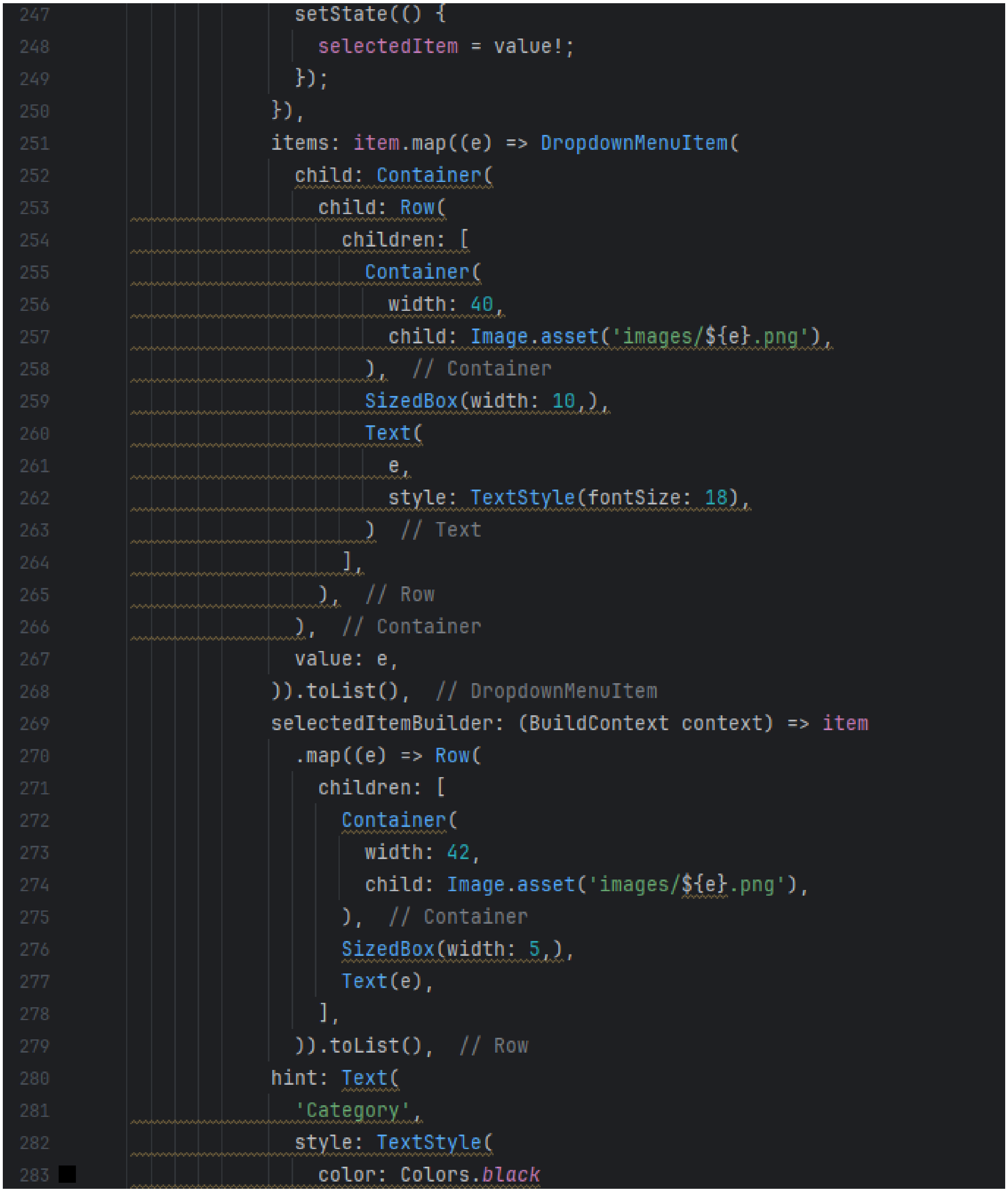
```dart
  }

  Padding How() {
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15),
      child: Container(
        padding: EdgeInsets.symmetric(horizontal: 10),
        width: 300,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          border: Border.all(
            width: 2,
            color: Colors.black,
          ),  // Border.all
        ),  // BoxDecoration
        child: DropdownButton<String>(
          value: selectedItem1,
          onChanged: ((value) {
            setState(() {
              selectedItem1 = value!;
            });
          }),
          items: item1
              .map((e) => DropdownMenuItem(
            child: Container(
              alignment: Alignment.center,
              child: Row(
                children: [
                  Text(
                    e,
                    style: TextStyle(fontSize: 18),
                  )  // Text
                ],
              ),  // Row
            ),  // Container
```



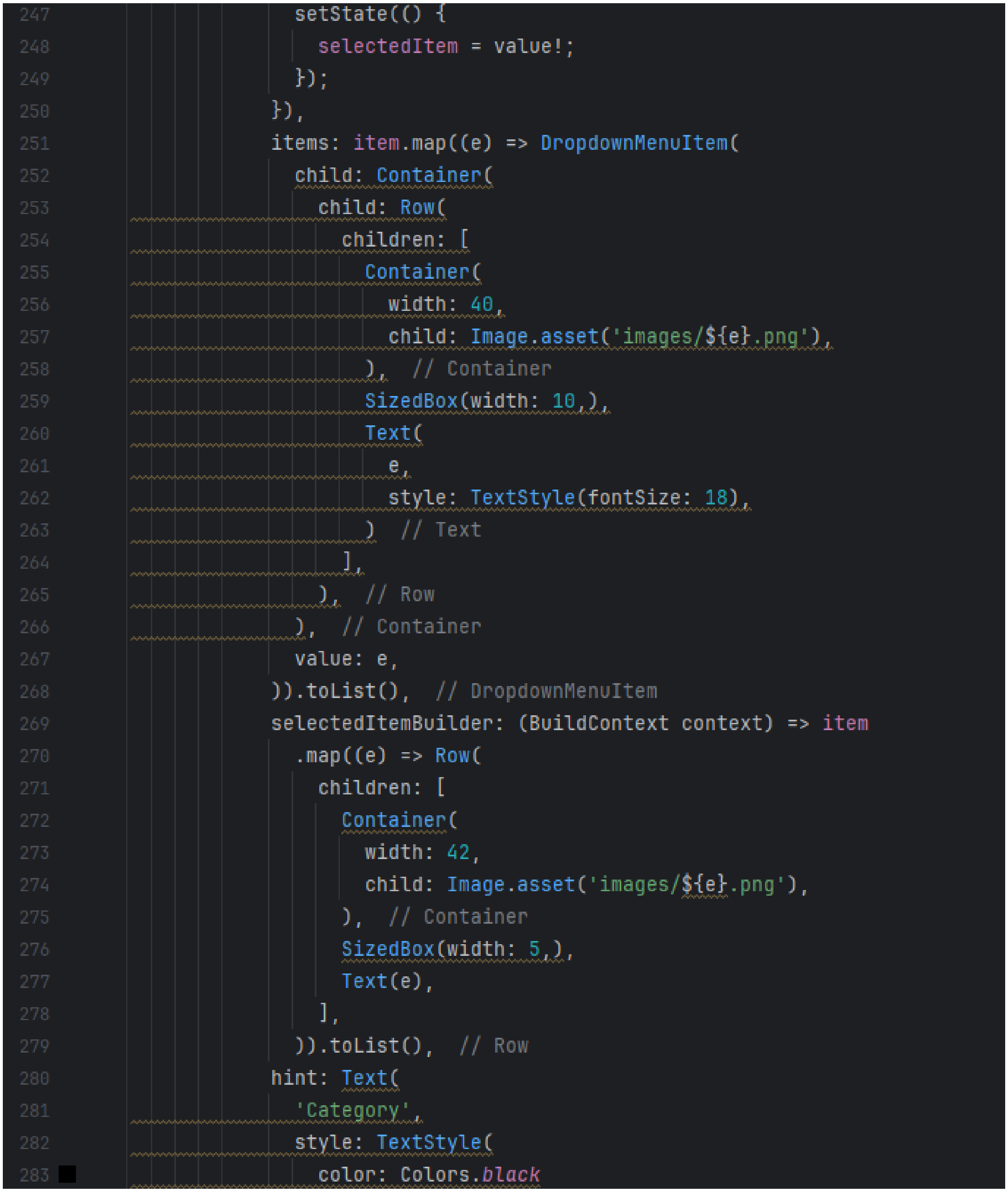
```
          ),  // Container
          value: e,
        ))  // DropdownMenuItem
            .toList(),
        selectedItemBuilder: (BuildContext context) => item1
            .map((e) => Row(
          children: [Text(e)],
        ))  // Row
            .toList(),
        hint: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 12),
          child: Text(
            'How',
            style: TextStyle(color: Colors.black),
          ),  // Text
        ),  // Padding
        dropdownColor: Colors.white,
        isExpanded: true,
        underline: Container(),
      ),  // DropdownButton
    ),  // Container
  );  // Padding
}

Padding amount1() {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 15),
    child: Container(
      width: 300,
      child: TextField(
        keyboardType: TextInputType.number,
        focusNode: amnt,
        controller: amount,
        decoration: InputDecoration(
          contentPadding: EdgeInsets.symmetric(horizontal: 15,),
          labelText: 'Amount',
```



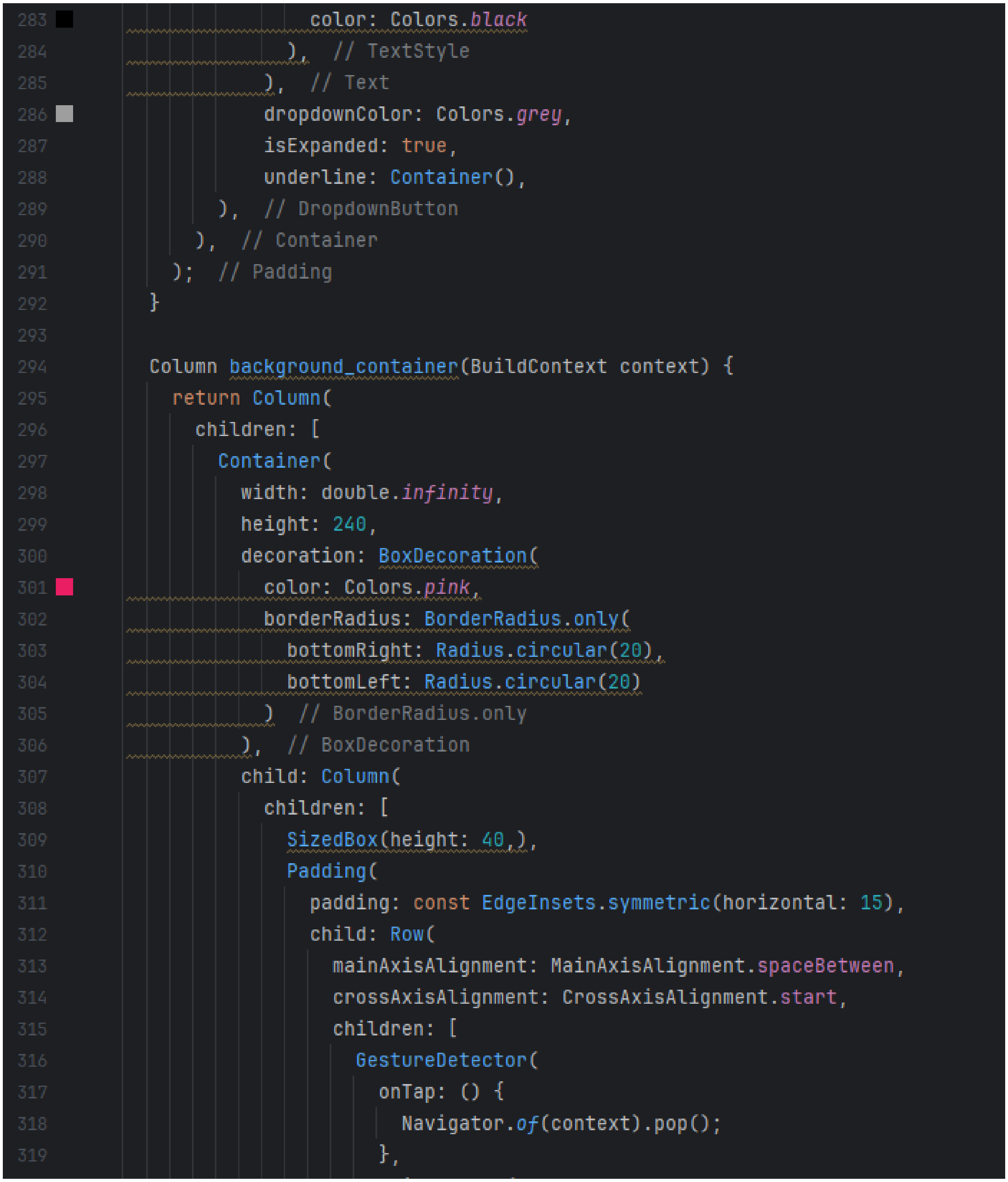
```
     labelText: 'Amount',
     labelStyle: TextStyle(
      fontSize: 17,
      color: Colors.black,
     ),  // TextStyle
     enabledBorder: OutlineInputBorder(
       borderRadius: BorderRadius.circular(10),
       borderSide: BorderSide(width: 2, color: Colors.black)
     ),  // OutlineInputBorder
     focusedBorder: OutlineInputBorder(
       borderRadius: BorderRadius.circular(10),
       borderSide: BorderSide(width: 2, color: Colors.black)
     ),  // OutlineInputBorder
    ),  // InputDecoration
   ),  // TextField
  ),  // Container
 );  // Padding
}

Padding name() {
 return Padding(
  padding: const EdgeInsets.symmetric(horizontal: 15),
  child: Container(
   padding: EdgeInsets.symmetric(horizontal: 15),
   width: 300,
   decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    border: Border.all(
     width: 2,
     color: Colors.black
    )  // Border.all
   ),  // BoxDecoration
   child: DropdownButton<String>(
     value: selectedItem,
     onChanged: ((value) {
      setState(() {
```



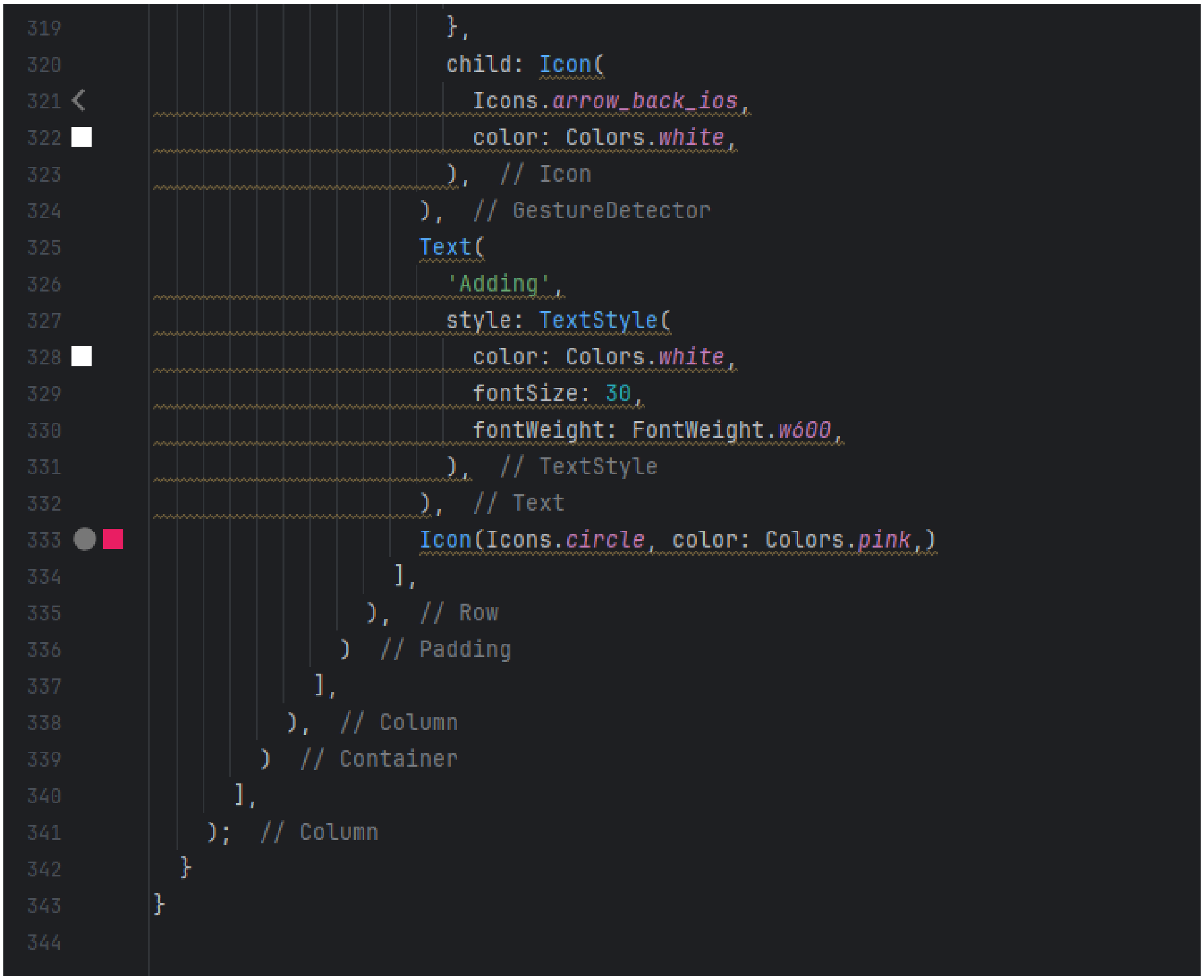
```
  setState(() {
    selectedItem = value!;
  });
}),
items: item.map((e) => DropdownMenuItem(
  child: Container(
    child: Row(
      children: [
        Container(
          width: 40,
          child: Image.asset('images/${e}.png'),
        ),  // Container
        SizedBox(width: 10,),
        Text(
          e,
          style: TextStyle(fontSize: 18),
        )  // Text
      ],
    ),  // Row
  ),  // Container
  value: e,
)).toList(),  // DropdownMenuItem
selectedItemBuilder: (BuildContext context) => item
  .map((e) => Row(
    children: [
      Container(
        width: 42,
        child: Image.asset('images/${e}.png'),
      ),  // Container
      SizedBox(width: 5,),
      Text(e),
    ],
  )).toList(),  // Row
hint: Text(
  'Category',
  style: TextStyle(
    color: Colors.black
```



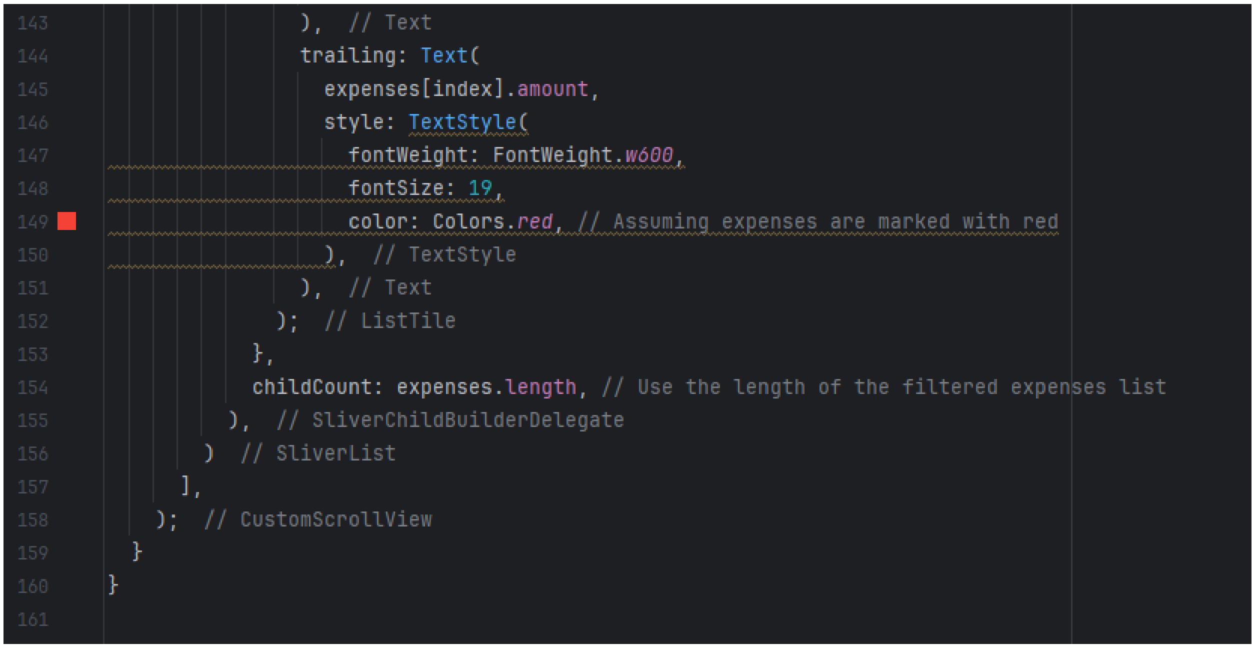
```
  setState(() {
    selectedItem = value!;
  });
}),
items: item.map((e) => DropdownMenuItem(
  child: Container(
    child: Row(
      children: [
        Container(
          width: 40,
          child: Image.asset('images/${e}.png'),
        ),  // Container
        SizedBox(width: 10,),
        Text(
          e,
          style: TextStyle(fontSize: 18),
        )  // Text
      ],
    ),  // Row
  ),  // Container
  value: e,
)).toList(),  // DropdownMenuItem
selectedItemBuilder: (BuildContext context) => item
  .map((e) => Row(
    children: [
      Container(
        width: 42,
        child: Image.asset('images/${e}.png'),
      ),  // Container
      SizedBox(width: 5,),
      Text(e),
    ],
  )).toList(),  // Row
hint: Text(
  'Category',
  style: TextStyle(
    color: Colors.black
```



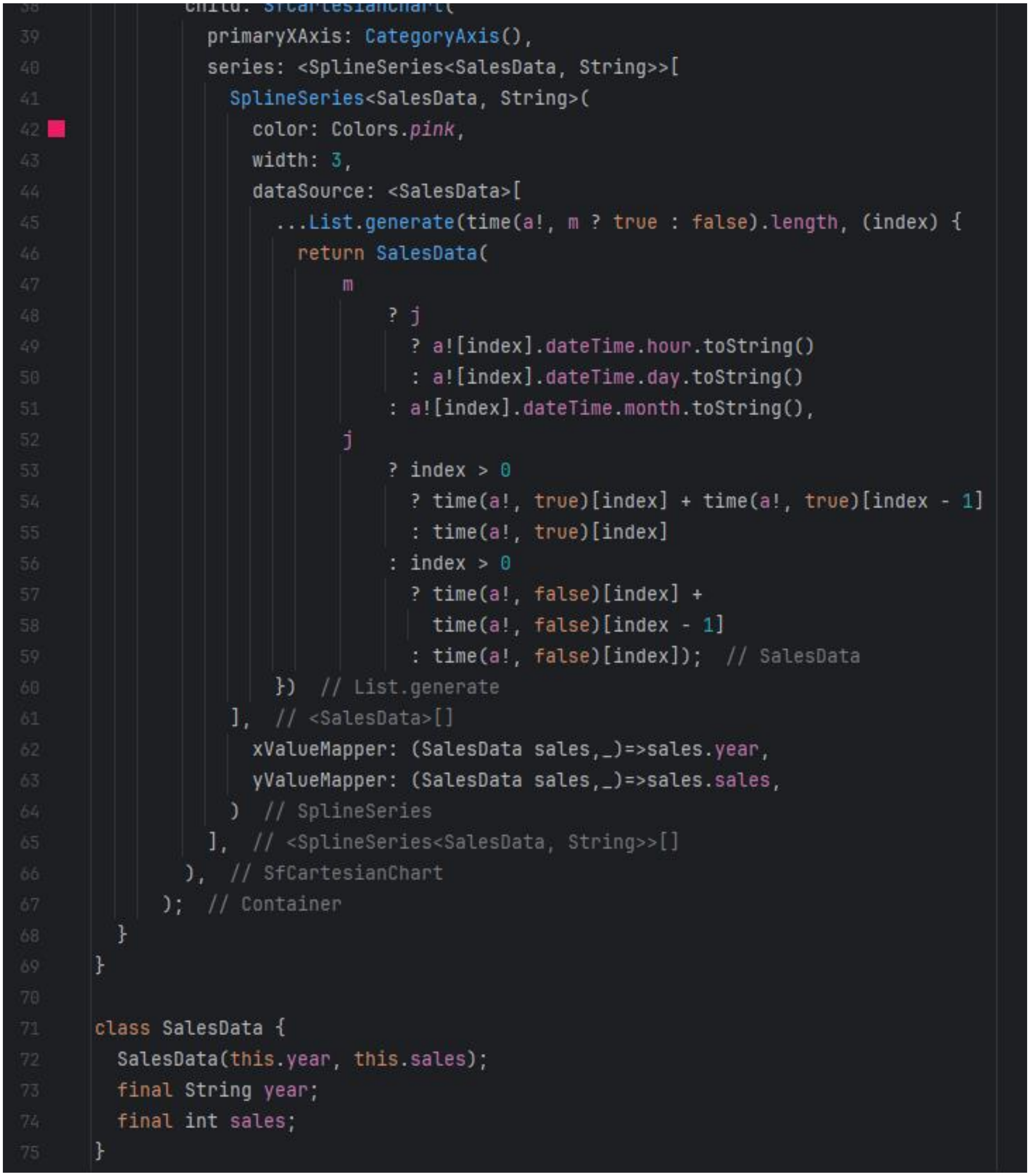
```
                color: Colors.black
              ),  // TextStyle
            ),  // Text
            dropdownColor: Colors.grey,
            isExpanded: true,
            underline: Container(),
        ),  // DropdownButton
      ),  // Container
    );  // Padding
  }

  Column background_container(BuildContext context) {
    return Column(
      children: [
        Container(
          width: double.infinity,
          height: 240,
          decoration: BoxDecoration(
            color: Colors.pink,
            borderRadius: BorderRadius.only(
              bottomRight: Radius.circular(20),
              bottomLeft: Radius.circular(20)
            )  // BorderRadius.only
          ),  // BoxDecoration
          child: Column(
            children: [
              SizedBox(height: 40,),
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 15),
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    GestureDetector(
                      onTap: () {
                        Navigator.of(context).pop();
                      },
```



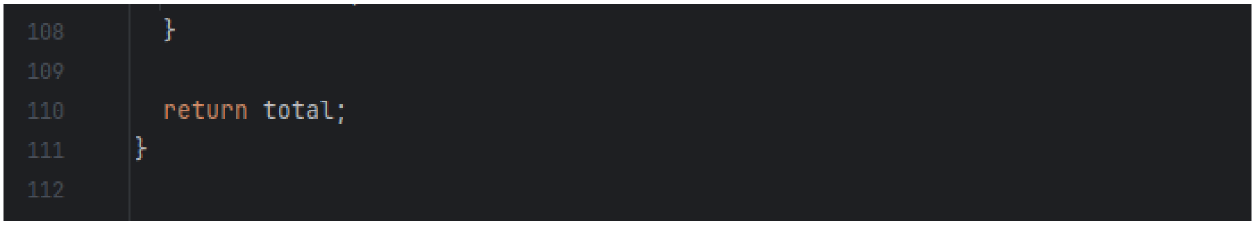
```
                      },
                      child: Icon(
                        Icons.arrow_back_ios,
                        color: Colors.white,
                      ),  // Icon
                    ),  // GestureDetector
                    Text(
                      'Adding',
                      style: TextStyle(
                        color: Colors.white,
                        fontSize: 30,
                        fontWeight: FontWeight.w600,
                      ),  // TextStyle
                    ),  // Text
                    Icon(Icons.circle, color: Colors.pink,)
                  ],
                ),  // Row
              )  // Padding
            ],
          ),  // Column
        )  // Container
      ],
    );  // Column
  }
}
```



```
                ),  // Text
                trailing: Text(
                  expenses[index].amount,
                  style: TextStyle(
                    fontWeight: FontWeight.w600,
                    fontSize: 19,
                    color: Colors.red, // Assuming expenses are marked with red
                  ),  // TextStyle
                ),  // Text
              );  // ListTile
            },
            childCount: expenses.length, // Use the length of the filtered expenses list
          ),  // SliverChildBuilderDelegate
        )  // SliverList
      ],
    );  // CustomScrollView
  }
}
```



```
        child: SfCartesianChart(
          primaryXAxis: CategoryAxis(),
          series: <SplineSeries<SalesData, String>>[
            SplineSeries<SalesData, String>(
              color: Colors.pink,
              width: 3,
              dataSource: <SalesData>[
                ...List.generate(time(a!, m ? true : false).length, (index) {
                  return SalesData(
                      m
                          ? j
                            ? a![index].dateTime.hour.toString()
                            : a![index].dateTime.day.toString()
                          : a![index].dateTime.month.toString(),
                      j
                          ? index > 0
                            ? time(a!, true)[index] + time(a!, true)[index - 1]
                            : time(a!, true)[index]
                          : index > 0
                            ? time(a!, false)[index] +
                              time(a!, false)[index - 1]
                            : time(a!, false)[index]);  // SalesData
                })  // List.generate
            ],  // <SalesData>[]
              xValueMapper: (SalesData sales,_)=>sales.year,
              yValueMapper: (SalesData sales,_)=>sales.sales,
            )  // SplineSeries
          ],  // <SplineSeries<SalesData, String>>[]
        ),  // SfCartesianChart
      );  // Container
  }
}

class SalesData {
  SalesData(this.year, this.sales);
  final String year;
  final int sales;
}
```



```
    }

    return total;
  }

```