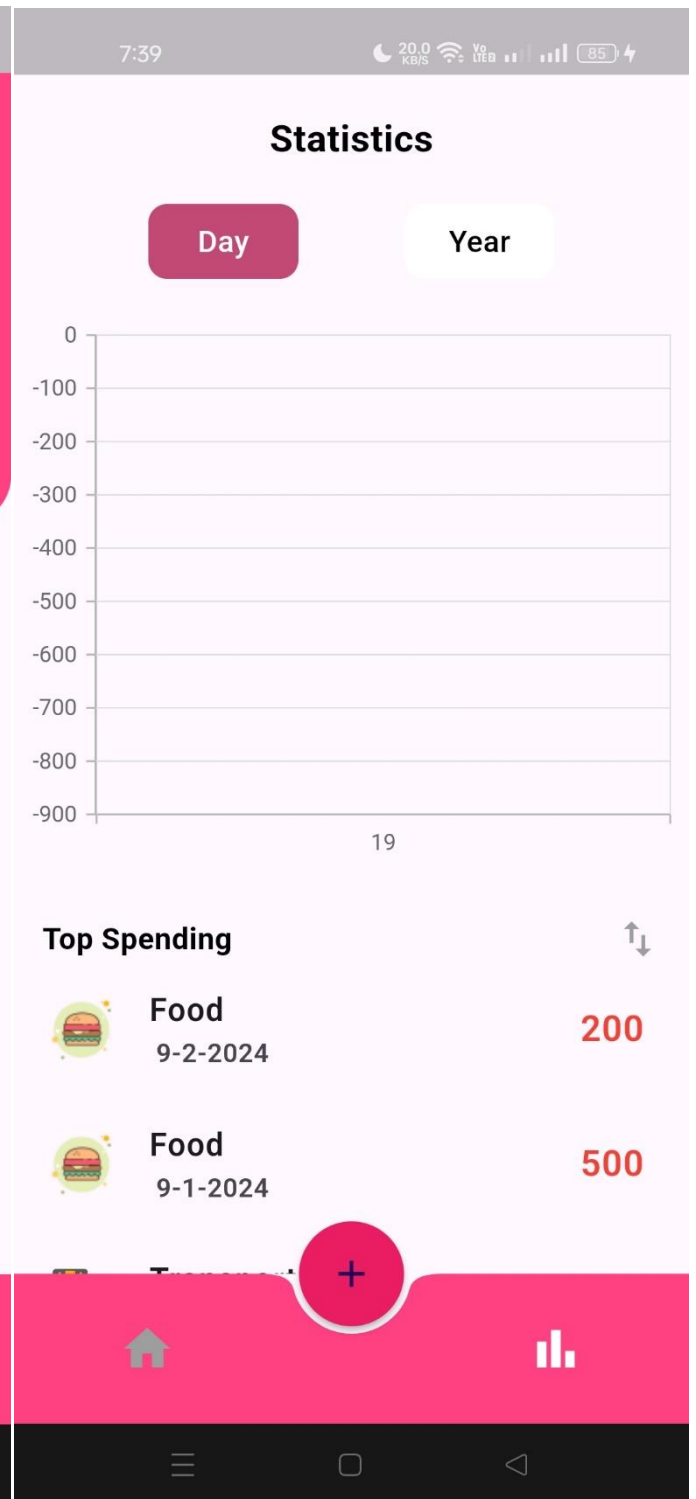
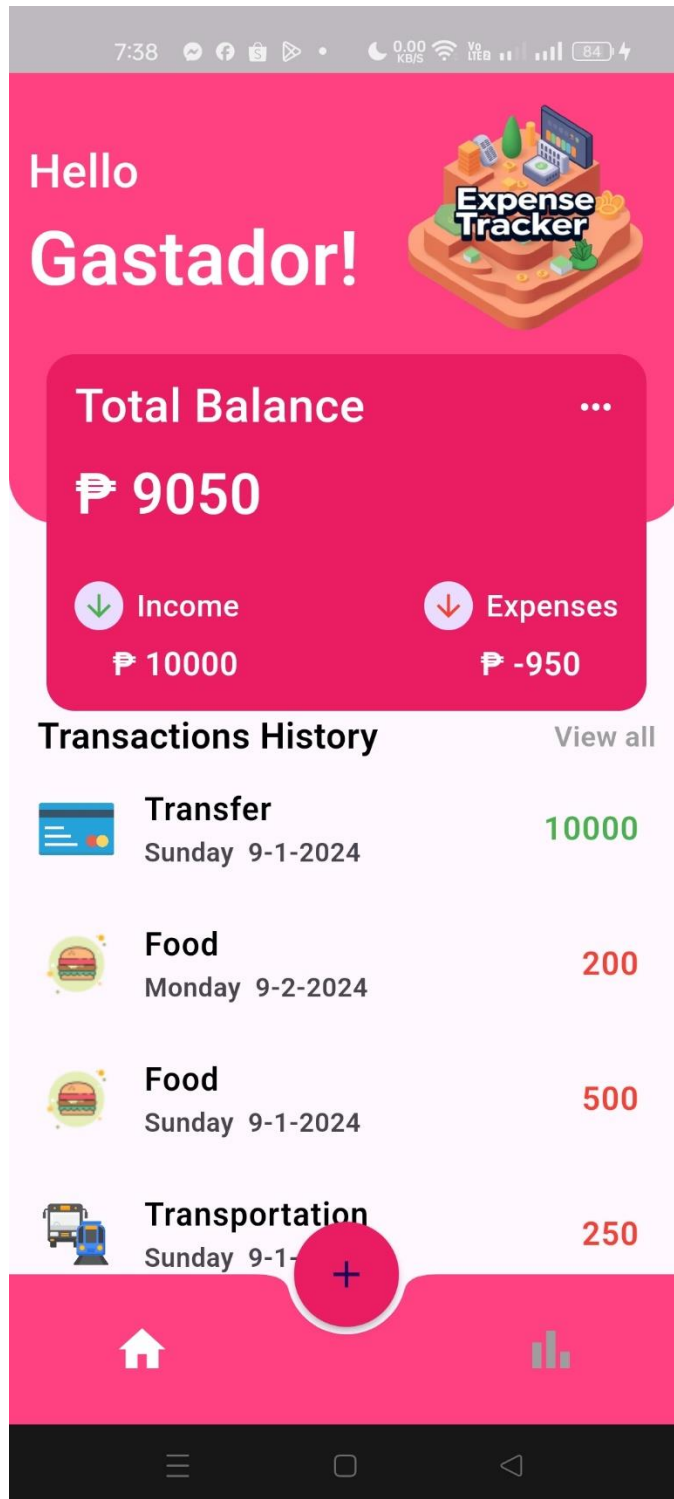
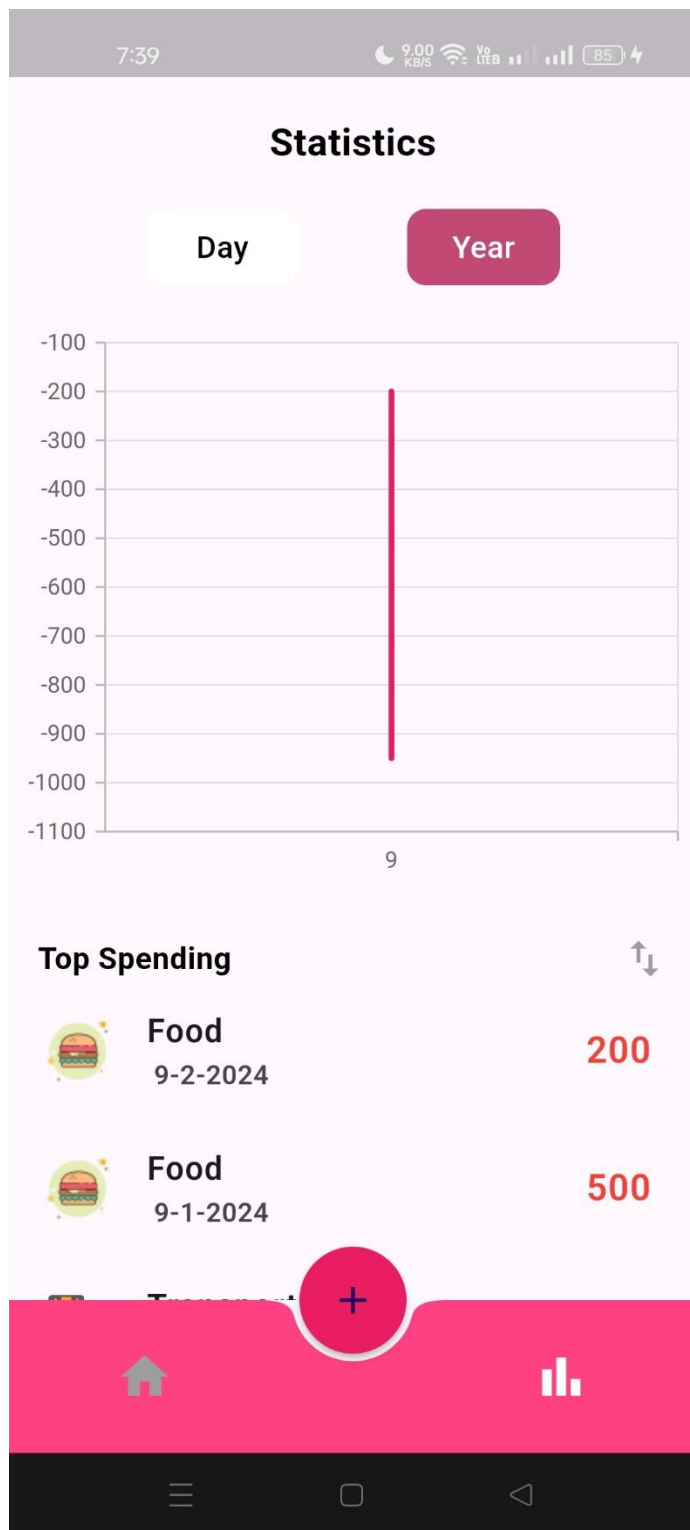


Expense Tracker





7:39 1.00 KB/S 5G LTE 85%

Adding

Food

Amount: 250

Expense

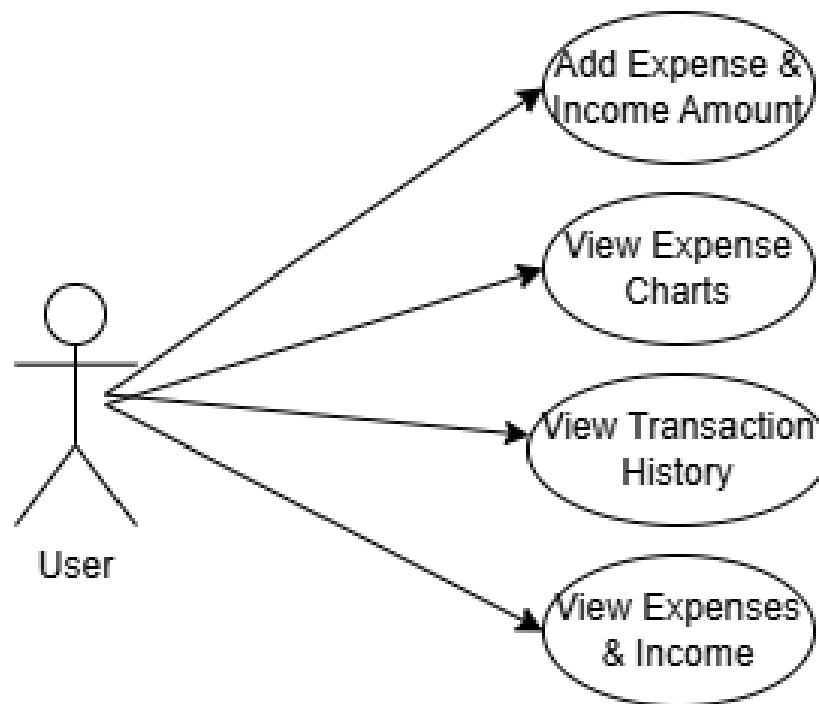
Date: 9 / 1 / 2024

Save

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

main.dart ×

```
1 import 'package:flutter/material.dart';
2 import 'package:pradyek101/data/model/add_data.dart';
3 import 'package:pradyek101/screens/home.dart';
4 import 'package:pradyek101/screens/statistics.dart';
5 import 'package:pradyek101/widgets/bottomNavBar.dart';
6 import 'package:hive_flutter/hive_flutter.dart';
7
8 void main() async {
9   await Hive.initFlutter();
10  Hive.registerAdapter(AddDataAdapter());
11  await Hive.openBox<AddData>('data');
12  runApp(const MyApp());
13 }
14
15 class MyApp extends StatelessWidget {
16   const MyApp({Key? key}) : super(key: key);
17
18   @override
19   Widget build(BuildContext context) {
20     return MaterialApp(
21       debugShowCheckedModeBanner: false,
22       home: BotNavBar(),
23     ); // MaterialApp
24   }
25 }
```

main.dart home.dart ×

```
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart';
3 import 'package:hive/hive.dart';
4 import 'package:hive_flutter/adapters.dart';
5 import 'package:pradyek101/data/dataList.dart';
6 import 'package:pradyek101/data/model/add_data.dart';
7 import 'package:pradyek101/data/utilities.dart';
8
9 class HomeScreen extends StatefulWidget {
10   const HomeScreen({super.key,});
11
12   @override
13   State<HomeScreen> createState() => _HomeScreenState();
14 }
15
16 class _HomeScreenState extends State<HomeScreen> {
17   var history;
18   final box = Hive.box<AddData>('data');
19   final List<String> day = [
20     'Monday',
21     'Tuesday',
22     'Wednesday',
23     'Thursday',
24     'Friday',
25     'Saturday',
26     'Sunday'
27   ];
28   @override
29   Widget build(BuildContext context) {
30     return Scaffold(
31       body: SafeArea(
32         child: ValueListenableBuilder(
33           valueListenable: box.listenable(),
34           builder: (context, value, child) {
35             return CustomScrollView(
36               slivers: [
37                 SliverToBoxAdapter(
```

```

36     slivers: [
37       SliverToBoxAdapter(
38         child: SizedBox(height: 340, child: _head()),
39       ), // SliverToBoxAdapter
40       SliverToBoxAdapter(
41         child: Padding(
42           padding: const EdgeInsets.symmetric(horizontal: 15),
43           child: Row(
44             mainAxisAlignment: MainAxisAlignment.spaceBetween,
45             children: [
46               Text(
47                 'Transactions History',
48                 style: TextStyle(
49                   fontWeight: FontWeight.w600,
50                   fontSize: 19,
51                   color: Colors.black,
52                 ), // TextStyle
53             ), // Text
54               Text(
55                 'View all',
56                 style: TextStyle(
57                   fontWeight: FontWeight.w600,
58                   fontSize: 15,
59                   color: Colors.grey,
60                 ), // TextStyle
61             ), // Text
62           ],
63         ), // Row
64       ), // Padding
65     ], // SliverToBoxAdapter
66     SliverList(
67       delegate: SliverChildBuilderDelegate(
68         (context, index) {
69           history = box.values.toList()[index];
70           return getList(history, index);
71         },

```

```

71         },
72         childCount: box.length,
73       ), // SliverChildBuilderDelegate
74     ) // SliverList
75   ],
76 ); // CustomScrollView
77 })), // ValueListenableBuilder, SafeArea
78 ); // Scaffold
79 }
80
81 Widget getList(AddData history, int index) {
82   return Dismissible(
83     key: UniqueKey(),
84     onDismissed: (direction) {
85       history.delete();
86     },
87     child: get(index, history)); // Dismissible
88 }
89
90 ListTile get(int index, AddData history) {
91   return ListTile(
92     leading: ClipRRect(
93       borderRadius: BorderRadius.circular(5),
94       child: Image.asset('images/${history.name}.png', height: 40,),
95     ), // ClipRRect
96     title: Text(
97       history.name,
98       style: TextStyle(
99         fontWeight: FontWeight.w600,
100        fontSize: 17,
101        color: Colors.black
102      ), // TextStyle
103     ), // Text
104     subtitle: Text(
105       '${day[history.dateTime.weekday - 1]} ${history.dateTime.month}-${history.dateTime.day}-${history.dateTime.year}',
106       style: TextStyle(
107         fontWeight: FontWeight.w600,

```



```

143         child: Container(
144             height: 150,
145             width: 150,
146             child: Image.asset('images/logo.png')
147         ), // Container
148     ) // ClipRRect
149 ), // Positioned
150 Padding(
151     padding: const EdgeInsets.only(top: 35, left: 10),
152     child: Column(
153         crossAxisAlignment: CrossAxisAlignment.start,
154         children: [
155             Text(
156                 'Hello',
157                 style: TextStyle(
158                     fontWeight: FontWeight.w500,
159                     fontSize: 25,
160                     color: Colors.white
161                 ), // TextStyle
162             ), // Text
163             Text(
164                 'Gastador!',
165                 style: TextStyle(
166                     fontWeight: FontWeight.w600,
167                     fontSize: 40,
168                     color: Colors.white
169                 ), // TextStyle
170             ), // Text
171         ],
172     ), // Column
173 ), // Padding
174 ],
175 ), // Stack
176 ), // Container
177 ],
178 ), // Column
179 Positioned(

```

```

179     Positioned(
180       top: 150,
181       left: 20,
182       child: Container(
183         height: 190,
184         width: 320,
185         decoration: BoxDecoration(
186           color: Colors.pink,
187           borderRadius: BorderRadius.circular(15)
188         ), // BoxDecoration
189         child: Column(
190           children: [
191             SizedBox(height: 10,),
192             Padding(
193               padding: const EdgeInsets.symmetric(horizontal: 15),
194               child: Row(
195                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
196                 children: [
197                   Text(
198                     'Total Balance',
199                     style: TextStyle(
200                       fontWeight: FontWeight.w500,
201                       fontSize: 25,
202                       color: Colors.white
203                     ) // TextStyle
204                   ), // Text
205                   Icon(
206                     Icons.more_horiz_outlined,
207                     color: Colors.white,
208                   ) // Icon
209                 ],
210               ), // Row
211             ), // Padding
212             SizedBox(height: 7,),
213             Padding(
214               padding: const EdgeInsets.only(left: 15),

```

```

214         padding: const EdgeInsets.only(left: 15),
215         child: Row(
216             children: [
217                 Text(
218                     '₹ ${total()}',
219                     style: TextStyle(
220                         fontWeight: FontWeight.w600,
221                         fontSize: 30,
222                         color: Colors.white
223                     ), // TextStyle
224                 ) // Text
225             ],
226         ), // Row
227     ), // Padding
228     SizedBox(height: 25,),
229     Padding(
230         padding: const EdgeInsets.symmetric(horizontal: 15),
231         child: Row(
232             mainAxisAlignment: MainAxisAlignment.spaceBetween,
233             children: [
234                 Row(
235                     children: [
236                         CircleAvatar(
237                             radius: 13,
238                             backgroundColor: Colors.grey[-10],
239                             child: Icon(
240                                 Icons.arrow_downward_sharp,
241                                 color: Colors.green,
242                                 size: 19,
243                             ), // Icon
244                         ), // CircleAvatar
245                         SizedBox(width: 7,),
246                         Text(
247                             'Income',
248                             style: TextStyle(
249                                 fontWeight: FontWeight.w500,
250                                 fontSize: 16,

```

```

71         },
72         childCount: box.length,
73       ), // SliverChildBuilderDelegate
74     ) // SliverList
75   ],
76 ); // CustomScrollView
77   })), // ValueListenableBuilder, SafeArea
78 ); // Scaffold
79 }
80
81 Widget getList(AddData history, int index) {
82   return Dismissible(
83     key: UniqueKey(),
84     onDismissed: (direction) {
85       history.delete();
86     },
87     child: get(index, history)); // Dismissible
88 }
89
90 ListTile get(int index, AddData history) {
91   return ListTile(
92     leading: ClipRRect(
93       borderRadius: BorderRadius.circular(5),
94       child: Image.asset('images/${history.name}.png', height: 40,),
95     ), // ClipRRect
96     title: Text(
97       history.name,
98       style: TextStyle(
99         fontWeight: FontWeight.w600,
100        fontSize: 17,
101        color: Colors.black
102      ), // TextStyle
103     ), // Text
104     subtitle: Text(
105       '${day[history.dateTime.weekday - 1]} ${history.dateTime.month}-${history.dateTime.day}-${history.dateTime.year}',
106       style: TextStyle(
107         fontWeight: FontWeight.w600,

```

```

143         child: Container(
144             height: 150,
145             width: 150,
146             child: Image.asset('images/logo.png')
147         ), // Container
148     ) // ClipRRect
149 ), // Positioned
150 Padding(
151     padding: const EdgeInsets.only(top: 35, left: 10),
152     child: Column(
153         crossAxisAlignment: CrossAxisAlignment.start,
154         children: [
155             Text(
156                 'Hello',
157                 style: TextStyle(
158                     fontWeight: FontWeight.w500,
159                     fontSize: 25,
160                     color: Colors.white
161                 ), // TextStyle
162             ), // Text
163             Text(
164                 'Gastador!',
165                 style: TextStyle(
166                     fontWeight: FontWeight.w600,
167                     fontSize: 40,
168                     color: Colors.white
169                 ), // TextStyle
170             ), // Text
171         ],
172     ), // Column
173 ), // Padding
174 ],
175 ), // Stack
176 ), // Container
177 ],
178 ), // Column
179 Positioned(

```

```

179     Positioned(
180       top: 150,
181       left: 20,
182       child: Container(
183         height: 190,
184         width: 320,
185         decoration: BoxDecoration(
186           color: Colors.pink,
187           borderRadius: BorderRadius.circular(15)
188         ), // BoxDecoration
189         child: Column(
190           children: [
191             SizedBox(height: 10,),
192             Padding(
193               padding: const EdgeInsets.symmetric(horizontal: 15),
194               child: Row(
195                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
196                 children: [
197                   Text(
198                     'Total Balance',
199                     style: TextStyle(
200                       fontWeight: FontWeight.w500,
201                       fontSize: 25,
202                       color: Colors.white
203                     ) // TextStyle
204                   ), // Text
205                   Icon(
206                     Icons.more_horiz_outlined,
207                     color: Colors.white,
208                   ) // Icon
209                 ],
210               ), // Row
211             ), // Padding
212             SizedBox(height: 7,),
213             Padding(
214               padding: const EdgeInsets.only(left: 15),

```

```

214         padding: const EdgeInsets.only(left: 15),
215         child: Row(
216             children: [
217                 Text(
218                     '₹ ${total()}',
219                     style: TextStyle(
220                         fontWeight: FontWeight.w600,
221                         fontSize: 30,
222                         color: Colors.white
223                     ), // TextStyle
224                 ) // Text
225             ],
226         ), // Row
227     ), // Padding
228     SizedBox(height: 25,),
229     Padding(
230         padding: const EdgeInsets.symmetric(horizontal: 15),
231         child: Row(
232             mainAxisAlignment: MainAxisAlignment.spaceBetween,
233             children: [
234                 Row(
235                     children: [
236                         CircleAvatar(
237                             radius: 13,
238                             backgroundColor: Colors.grey[-10],
239                             child: Icon(
240                                 Icons.arrow_downward_sharp,
241                                 color: Colors.green,
242                                 size: 19,
243                             ), // Icon
244                         ), // CircleAvatar
245                         SizedBox(width: 7,),
246                         Text(
247                             'Income',
248                             style: TextStyle(
249                                 fontWeight: FontWeight.w500,
250                                 fontSize: 16,

```

```

250         fontSize: 16,
251         color: Colors.white
252     ), // TextStyle
253 ), // Text
254 ],
255 ), // Row
256 Row(
257     children: [
258         CircleAvatar(
259             radius: 13,
260             backgroundColor: Colors.grey[-10],
261             child: Icon(
262                 Icons.arrow_downward_sharp,
263                 color: Colors.red,
264                 size: 19,
265             ), // Icon
266         ), // CircleAvatar
267         SizedBox(width: 7,),
268         Text(
269             'Expenses',
270             style: TextStyle(
271                 fontWeight: FontWeight.w500,
272                 fontSize: 16,
273                 color: Colors.white
274             ), // TextStyle
275         ), // Text
276     ],
277 ), // Row
278 ],
279 ), // Row
280 ), // Padding
281 SizedBox(height: 6,),
282 Padding(
283     padding: const EdgeInsets.symmetric(horizontal: 35),
284     child: Row(
285         mainAxisAlignment: MainAxisAlignment.spaceBetween,
286         children: [

```



```

286         children: [
287             Text(
288                 '₹ ${income()}',
289                 style: TextStyle(
290                     fontWeight: FontWeight.w600,
291                     fontSize: 17,
292                     color: Colors.white
293                 ), // TextStyle
294             ), // Text
295             Text(
296                 '₹ ${expenses()}',
297                 style: TextStyle(
298                     fontWeight: FontWeight.w600,
299                     fontSize: 17,
300                     color: Colors.white
301                 ), // TextStyle
302             ), // Text
303         ],
304     ), // Row
305 ), // Padding
306 ],
307 ), // Column
308 ), // Container
309 ), // Positioned
310 ],
311 ); // Stack
312 }
313 }
314

```

```

38     });
39   });
40 }
41 @Widget build(BuildContext context) {
42   return Scaffold(
43     backgroundColor: Colors.grey.shade100,
44     body: SafeArea(
45       child: Stack(
46         alignment: Alignment.center,
47         children: [
48           background_container(context),
49           Positioned(
50             top: 120,
51             child: main_container(),
52           ) // Positioned
53         ],
54       ) // Stack
55     ), // SafeArea
56   ); // Scaffold
57 }
58
59 Container main_container() {
60   return Container(
61     decoration: BoxDecoration(
62       borderRadius: BorderRadius.circular(20),
63       color: Color(0xFFE3AADD)
64     ), // BoxDecoration
65     height: 450,
66     width: 340,
67     child: Column(
68       children: [
69         SizedBox(height: 50,),
70         name(),
71         SizedBox(height: 30,),
72         amount1(),

```

```

72     amount1(),|
73     SizedBox(height: 30,),
74     How(),
75     SizedBox(height: 30),
76     dateTime(),
77     Spacer(),
78     save1(),
79     SizedBox(height: 20,)
80   ],
81 ), // Column
82 ); // Container
83 }
84
85 GestureDetector save1() {
86   return GestureDetector(
87     onTap: () {
88     var add = AddData(selectedItem!, amount.text, date, explain.text, selectedItem!);
89     box.add(add);
90     Navigator.of(context).pop();
91   },
92   child: Container(
93     alignment: Alignment.center,
94     decoration: BoxDecoration(
95       borderRadius: BorderRadius.circular(15),
96       border: Border.all(width: 2, color: Colors.black),
97       color: Color.fromARGB(255, 239, 154, 211),
98     ), // BoxDecoration
99     width: 120,
100    height: 50,
101    child: Text(
102      'Save',
103      style: TextStyle(
104        fontSize: 17,
105        fontWeight: FontWeight.w600,
106        color: Colors.black,
107      ), // TextStyle

```

```

107         ), // TextStyle
108     ), // Text
109 ), // Container
110 ); // GestureDetector
111 }
112
113 Widget dateTime() {
114     return Container(
115         alignment: Alignment.bottomLeft,
116         decoration: BoxDecoration(
117             borderRadius: BorderRadius.circular(10),
118             border: Border.all(width: 2, color: Colors.black)
119         ), // BoxDecoration
120         width: 300,
121         child: TextButton(
122             onPressed: () async {
123                 DateTime? newDate = await showDatePicker(
124                     context: context,
125                     initialDate: date,
126                     firstDate: DateTime(2024),
127                     lastDate: DateTime(2100)
128                 );
129                 if (newDate == Null) return;
130                 setState(() {
131                     date = newDate!;
132                 });
133             },
134             child: Text(
135                 'Date : ${date.month} / ${date.day} / ${date.year}',
136                 style: TextStyle(
137                     fontSize: 15,
138                     color: Colors.black
139                 ), // TextStyle
140             ), // Text
141         ), // TextButton
142     ); // Container
143 }

```

```

143     }
144
145     Padding How() {
146         return Padding(
147             padding: const EdgeInsets.symmetric(horizontal: 15),
148             child: Container(
149                 padding: EdgeInsets.symmetric(horizontal: 10),
150                 width: 300,
151                 decoration: BoxDecoration(
152                     borderRadius: BorderRadius.circular(10),
153                     border: Border.all(
154                         width: 2,
155                         color: Colors.black,
156                     ), // Border.all
157                 ), // BoxDecoration
158                 child: DropdownButton<String>(
159                     value: selectedItem1,
160                     onChanged: ((value) {
161                         setState(() {
162                             selectedItem1 = value!;
163                         });
164                     }),
165                     items: item1
166                         .map((e) => DropdownMenuItem(
167                             child: Container(
168                                 alignment: Alignment.center,
169                                 child: Row(
170                                     children: [
171                                         Text(
172                                             e,
173                                             style: TextStyle(fontSize: 18),
174                                         ) // Text
175                                     ],
176                                 ), // Row
177                                     ), // Container

```

```

177         ), // Container
178         value: e,
179     )) // DropdownMenuItem
180     .toList(),
181     selectedItemBuilder: (BuildContext context) => item1
182     .map((e) => Row(
183         children: [Text(e)],
184     )) // Row
185     .toList(),
186     hint: Padding(
187         padding: const EdgeInsets.symmetric(horizontal: 12),
188         child: Text(
189             'How',
190             style: TextStyle(color: Colors.black),
191         ), // Text
192     ), // Padding
193     dropdownColor: Colors.white,
194     isExpanded: true,
195     underline: Container(),
196 ), // DropdownButton
197 ), // Container
198 ); // Padding
199 }
200
201 Padding amount1() {
202     return Padding(
203         padding: const EdgeInsets.symmetric(horizontal: 15),
204         child: Container(
205             width: 300,
206             child: TextField(
207                 keyboardType: TextInputType.number,
208                 focusNode: amnt,
209                 controller: amount,
210                 decoration: InputDecoration(
211                     contentPadding: EdgeInsets.symmetric(horizontal: 15,),
212                     labelText: 'Amount',

```

```

212         labelText: 'Amount',
213         labelStyle: TextStyle(
214             fontSize: 17,
215             color: Colors.black,
216         ), // TextStyle
217         enabledBorder: OutlineInputBorder(
218             borderRadius: BorderRadius.circular(10),
219             borderSide: BorderSide(width: 2, color: Colors.black)
220         ), // OutlineInputBorder
221         focusedBorder: OutlineInputBorder(
222             borderRadius: BorderRadius.circular(10),
223             borderSide: BorderSide(width: 2, color: Colors.black)
224         ), // OutlineInputBorder
225     ), // InputDecoration
226 ), // TextField
227 ), // Container
228 ); // Padding
229 }
230
231 Padding name() {
232     return Padding(
233         padding: const EdgeInsets.symmetric(horizontal: 15),
234         child: Container(
235             padding: EdgeInsets.symmetric(horizontal: 15),
236             width: 300,
237             decoration: BoxDecoration(
238                 borderRadius: BorderRadius.circular(10),
239                 border: Border.all(
240                     width: 2,
241                     color: Colors.black
242                 ) // Border.all
243             ), // BoxDecoration
244             child: DropdownButton<String>(
245                 value: selectedItem,
246                 onChanged: ((value) {
247                     setState() {

```

```

247         setState(() {
248             selectedItem = value!;
249         });
250     },
251     items: item.map((e) => DropdownMenuItem(
252         child: Container(
253             child: Row(
254                 children: [
255                     Container(
256                         width: 40,
257                         child: Image.asset('images/${e}.png'),
258                     ), // Container
259                     SizedBox(width: 10,),
260                     Text(
261                         e,
262                         style: TextStyle(fontSize: 18),
263                     ) // Text
264                 ],
265             ), // Row
266         ), // Container
267         value: e,
268     )).toList(), // DropdownMenuItem
269     selectedItemBuilder: (BuildContext context) => item
270         .map((e) => Row(
271             children: [
272                 Container(
273                     width: 42,
274                     child: Image.asset('images/${e}.png'),
275                 ), // Container
276                 SizedBox(width: 5,),
277                 Text(e),
278             ],
279         )).toList(), // Row
280     hint: Text(
281         'Category',
282         style: TextStyle(
283             color: Colors.black

```



```

247         setState(() {
248             selectedItem = value!;
249         });
250     },
251     items: item.map((e) => DropdownMenuItem(
252         child: Container(
253             child: Row(
254                 children: [
255                     Container(
256                         width: 40,
257                         child: Image.asset('images/${e}.png'),
258                     ), // Container
259                     SizedBox(width: 10,),
260                     Text(
261                         e,
262                         style: TextStyle(fontSize: 18),
263                     ) // Text
264                 ],
265             ), // Row
266         ), // Container
267         value: e,
268     )).toList(), // DropdownMenuItem
269     selectedItemBuilder: (BuildContext context) => item
270         .map((e) => Row(
271             children: [
272                 Container(
273                     width: 42,
274                     child: Image.asset('images/${e}.png'),
275                 ), // Container
276                 SizedBox(width: 5,),
277                 Text(e),
278             ],
279         )).toList(), // Row
280     hint: Text(
281         'Category',
282         style: TextStyle(
283             color: Colors.black

```

```

283     color: Colors.black
284   ), // TextStyle
285 ), // Text
286 dropdownColor: Colors.grey,
287 isExpanded: true,
288 underline: Container(),
289 ), // DropdownButton
290 ), // Container
291 ); // Padding
292 }
293
294 Column background_container(BuildContext context) {
295   return Column(
296     children: [
297       Container(
298         width: double.infinity,
299         height: 240,
300         decoration: BoxDecoration(
301           color: Colors.pink,
302           borderRadius: BorderRadius.only(
303             bottomRight: Radius.circular(20),
304             bottomLeft: Radius.circular(20)
305           ) // BorderRadius.only
306         ), // BoxDecoration
307       child: Column(
308         children: [
309           SizedBox(height: 40,),
310           Padding(
311             padding: const EdgeInsets.symmetric(horizontal: 15),
312             child: Row(
313               mainAxisAlignment: MainAxisAlignment.spaceBetween,
314               crossAxisAlignment: CrossAxisAlignment.start,
315               children: [
316                 GestureDetector(
317                   onTap: () {
318                     Navigator.of(context).pop();
319                   },

```

```

319         },
320         child: Icon(
321           Icons.arrow_back_ios,
322           color: Colors.white,
323         ), // Icon
324       ), // GestureDetector
325       Text(
326         'Adding',
327         style: TextStyle(
328           color: Colors.white,
329           fontSize: 30,
330           fontWeight: FontWeight.w600,
331         ), // TextStyle
332       ), // Text
333       Icon(Icons.circle, color: Colors.pink,)
334     ],
335   ), // Row
336 ) // Padding
337 ],
338 ), // Column
339 ) // Container
340 ],
341 ); // Column
342 }
343 }
344

```

```

143       ), // Text
144       trailing: Text(
145         expenses[index].amount,
146         style: TextStyle(
147           fontWeight: FontWeight.w600,
148           fontSize: 19,
149           color: Colors.red, // Assuming expenses are marked with red
150         ), // TextStyle
151       ), // Text
152     ); // ListTile
153   },
154   childCount: expenses.length, // Use the length of the filtered expenses list
155 ), // SliverChildBuilderDelegate
156 ) // SliverList
157 ],
158 ); // CustomScrollView
159 }
160 }
161

```

```

38 child: SfCartesianChart(
39   primaryXAxis: CategoryAxis(),
40   series: <SplineSeries<SalesData, String>>[
41     SplineSeries<SalesData, String>(
42       color: Colors.pink,
43       width: 3,
44       dataSource: <SalesData>[
45         ...List.generate(time(a!, m ? true : false).length, (index) {
46           return SalesData(
47             m
48               ? j
49               : a![index].dateTime.hour.toString()
50               : a![index].dateTime.day.toString()
51               : a![index].dateTime.month.toString(),
52             j
53               ? index > 0
54               : time(a!, true)[index] + time(a!, true)[index - 1]
55               : time(a!, true)[index]
56               : index > 0
57               ? time(a!, false)[index] +
58                 time(a!, false)[index - 1]
59               : time(a!, false)[index]); // SalesData
60           }) // List.generate
61       ], // <SalesData>[]
62       xValueMapper: (SalesData sales, _) => sales.year,
63       yValueMapper: (SalesData sales, _) => sales.sales,
64     ) // SplineSeries
65   ], // <SplineSeries<SalesData, String>>[]
66 ), // SfCartesianChart
67 ); // Container
68 }
69 }
70
71 class SalesData {
72   SalesData(this.year, this.sales);
73   final String year;
74   final int sales;
75 }

```

```

108   }
109
110   return total;
111 }
112

```