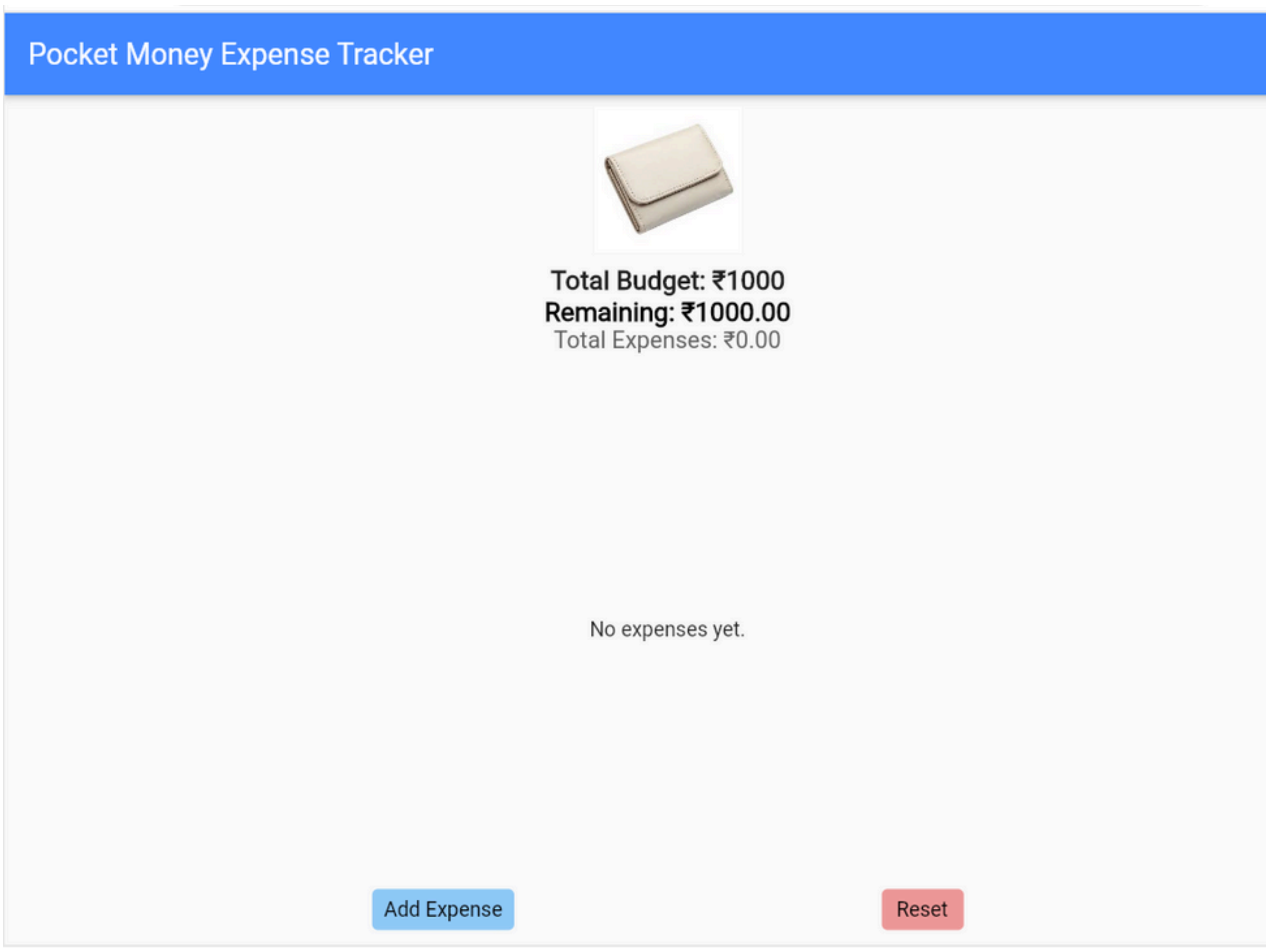
Flutter Code

```
import 'package:flutter/material.dart';  //importing the pre-defined libraries of flutter() and material design widgets such as scaffold, AppBar, Text, Buttons, etc

void main() { //starting point of program with no return type

  runApp(PocketMoneyExpenseTracker()); //widget that runs flutter program

}

// Stateless wrapper

class PocketMoneyExpenseTracker extends StatelessWidget {  //creating a new widget named PocketMoneyExpenseTracker which is stateless (data can not be changed)

  @override

  Widget build(BuildContext context) {  //creating context in stateless widget

    return MaterialApp(  //returns the complete app that is getting created

      debugShowCheckedModeBanner: false,  //removes or hides debug banner on right top

      title: 'Pocket Money Expense Tracker',  //title of the app

      home: ExpenseHome(),  //a part or room of MaterialApp

    );}}

// Stateful widget for tracking expenses

class ExpenseHome extends StatefulWidget {  //creating a class for room ExpenseHome which is StatefulWidget (where data keeps changing)

  @override

  _ExpenseHomeState createState() => _ExpenseHomeState(); //creating a state for statefulWidget

}

class _ExpenseHomeState extends State<ExpenseHome> {  //“_“ represents private class, where the class looks into the state of ExpenseHome (i.e., changes data of home as per state)

  double totalBudget = 1000;

  double remainingMoney = 1000;

  double totalExpenses = 0; //initialization of local variables
  List<Map<String, dynamic>> expenses = [];  //map- holds key, value pairs; expenses list collects the
name of item and value of the item

  final TextEditingController itemController = TextEditingController();  //creating dynamic text field to
enter the item name which is final ( doesn’t change)

  final TextEditingController costController = TextEditingController();  //creating dynamic text field to
enter cost of the item

  void addExpense(String item, double cost) {  //function without return type with parameters item and
cost

    setState(() {  //informing the state

      expenses.add({'item': item, 'cost': cost});  //adding values to list

      remainingMoney -= cost;  //calculation of the leftover money after expenditure

      totalExpenses += cost;  //calculation of total spendings

    });

    // Warning if overspending

   if (remainingMoney < 0) {

      showDialog(  //function to show dialog box

        context: context,

        builder: (context) => AlertDialog( //creation of dialog box

          title: Text("⚠ Overspending!"),  //heading

          content: Text("You have exceeded your budget by ₹${remainingMoney.abs()}"),  //body

          actions: [

            TextButton(

              onPressed: () => Navigator.pop(context),  //when button pressed it navigates to dialog box and
closes when ok is clicked

              child: Text("OK"),  //button

            ), ], ),); } }

  void resetTracker() {  //function to reset the tracking
```

```
    setState(() {   //calling setState
      expenses.clear();    //clearing all expenses
      remainingMoney = totalBudget;
      totalExpenses = 0;
    });}
  void showAddExpenseDialog() {  //function to show expense box with item name and cost for the user to
enter
    itemController.clear();
    costController.clear();
    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(  //creating Add Expense dialog box
          title: Text("Add Expense"),  //title
          content: Column(  //column widget
            mainAxisSize: MainAxisSize.min,  //vertical space along main axis
            children: [
              TextField(
                controller: itemController,
                decoration: InputDecoration(  //styling inside Add Expense
                  labelText: "Item Name",
                  border: OutlineInputBorder(),  //creates rectangular border
                ), ),
              SizedBox(height: 10),  //invisible box of 10px length is created for the text to be entered
              TextField(  // field to enter the value of cost
                controller: costController,
            keyboardType: TextInputType.number,  //takes only number values

            decoration: InputDecoration(

              labelText: "Cost (₹)",

              border: OutlineInputBorder(),

            ),),],),

         actions: [

           TextButton(

  onPressed: () {

    String item = itemController.text.trim();

    double? cost = double.tryParse(costController.text);  //tryParse is used to nullify any other string values
other than numbers

    if (item.isNotEmpty && cost != null && cost > 0) {

      // check if adding this cost exceeds budget

      if (remainingMoney - cost < 0) {  //if remaining amount is negative

  showDialog(

    context: context,

    builder: (context) => AlertDialog(

      title: Text("⚠ Overspending!"),

      content: Text("You will exceed your budget by ₹${(cost - remainingMoney).toStringAsFixed(2)}"),

      actions: [

        TextButton(

          onPressed: () {

            Navigator.pop(context); // close warning

            addExpense(item, cost); // now add expense

            Navigator.pop(context); // close Add Expense dialog

          },
          child: Text("OK"),
        ),],),);
    } else {
      addExpense(item, cost);
      Navigator.pop(context); // close the Add Expense dialog
    }}},
      child: Text("Save"),
    ),
            TextButton(
              onPressed: () => Navigator.pop(context),
              child: Text("Cancel"),
            ),], );},);}
      @override
      Widget build(BuildContext context) {
        return Scaffold( //refers to the structure of the screen
          appBar: AppBar(  //header of the main widget
            title: Text('Pocket Money Expense Tracker'),
            backgroundColor: Colors.blueAccent,
          ),
          body: Column(  //column widget
            children: [
              // Top image
              Padding(
                padding: const EdgeInsets.all(8.0),  //add 8px of space at all the 4 edges
                child: Image.network(  //to insert image directly from internet
              'https://cdn-icons-png.flaticon.com/512/1020/1020567.png',
              height: 100,
            ), ),
          // Budget info
          Text(
            'Total Budget: ₹$totalBudget',  //'$' inserts the value of variable here
            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
          ),
          Text(
            'Remaining: ₹${remainingMoney.toStringAsFixed(2)}',
            style: TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
              color: remainingMoney < 0 ? Colors.red : Colors.black,  //if remaining money is negative then text
    color is red or else black
            ),),
          Text(
            'Total Expenses: ₹${totalExpenses.toStringAsFixed(2)}',
            style: TextStyle(fontSize: 16, color: Colors.grey[700]),
          ),
          SizedBox(height: 10),
          // List of expenses
          Expanded(
            child: expenses.isEmpty
                ? Center(child: Text("No expenses yet."))  //shows the statement if there are no expenses
    registered
```

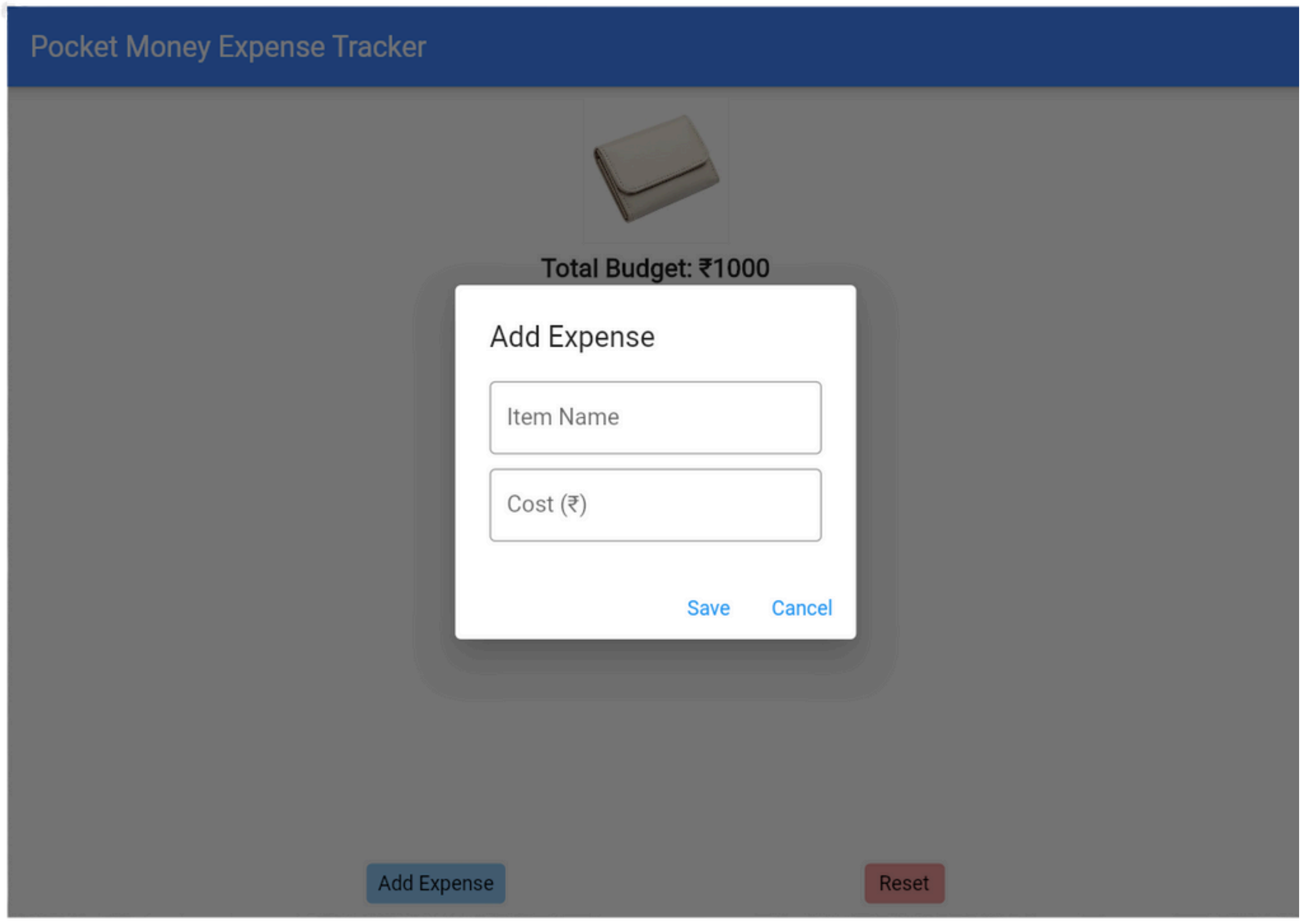
```
      : ListView.builder(  //if registered

        itemCount: expenses.length,

        itemBuilder: (context, index) {

          return Container(

            margin: EdgeInsets.symmetric(vertical: 5, horizontal: 10),

            padding: EdgeInsets.all(10),

            decoration: BoxDecoration(

              color: Colors.blue.shade50,

              borderRadius: BorderRadius.circular(10),

              border: Border.all(color: Colors.blue.shade200),

            ),

            child: Row(

              mainAxisAlignment: MainAxisAlignment.spaceBetween,  //creates space between right
and left children

              children: [

                Text(expenses[index]['item']),

                Text('₹${expenses[index]['cost']}'),

              ],),  ); },  ),),

  Row(

  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  //equal spacing

  children: [

    TextButton(

      style: TextButton.styleFrom(

        backgroundColor: Colors.blue.shade200,

        foregroundColor: Colors.black,

      ),
            onPressed: showAddExpenseDialog,
            child: Text("Add Expense"),
          ),
          TextButton(
            style: TextButton.styleFrom(
              backgroundColor: Colors.red.shade200,
              foregroundColor: Colors.black,
            ),
            onPressed: resetTracker,  //to reset the tracker
            child: Text("Reset"),
          ),], ),
        SizedBox(height: 10),
      ], ), );
  }
}
```

Output Screens:


Pocket Money Expense Tracker

Total Budget: ₹1000
Remaining: ₹1000.00
Total Expenses: ₹0.00

No expenses yet.

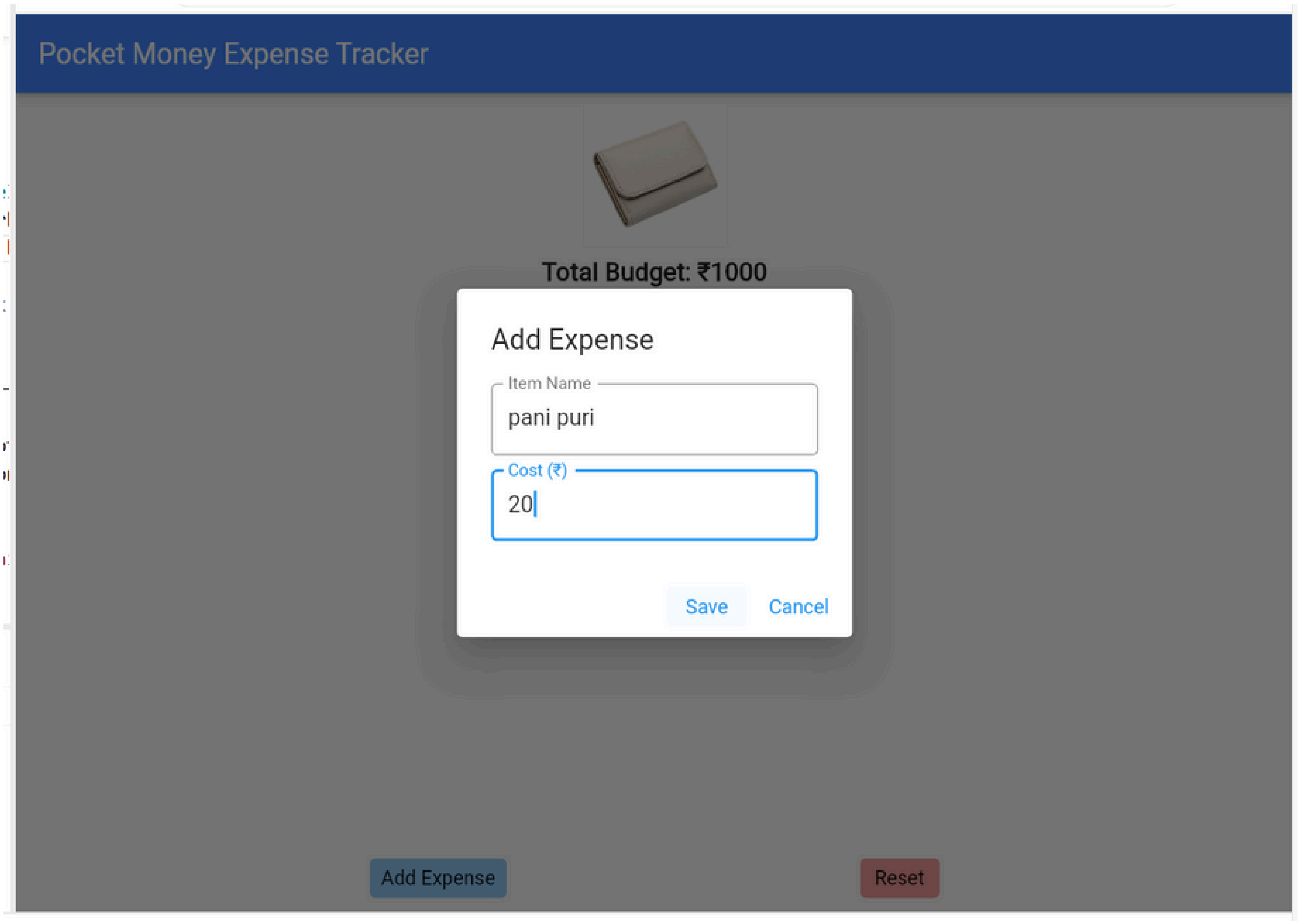

Pocket Money Expense Tracker

Total Budget: ₹1000

Add Expense

Item Name

Cost (₹)


Pocket Money Expense Tracker

Total Budget: ₹1000

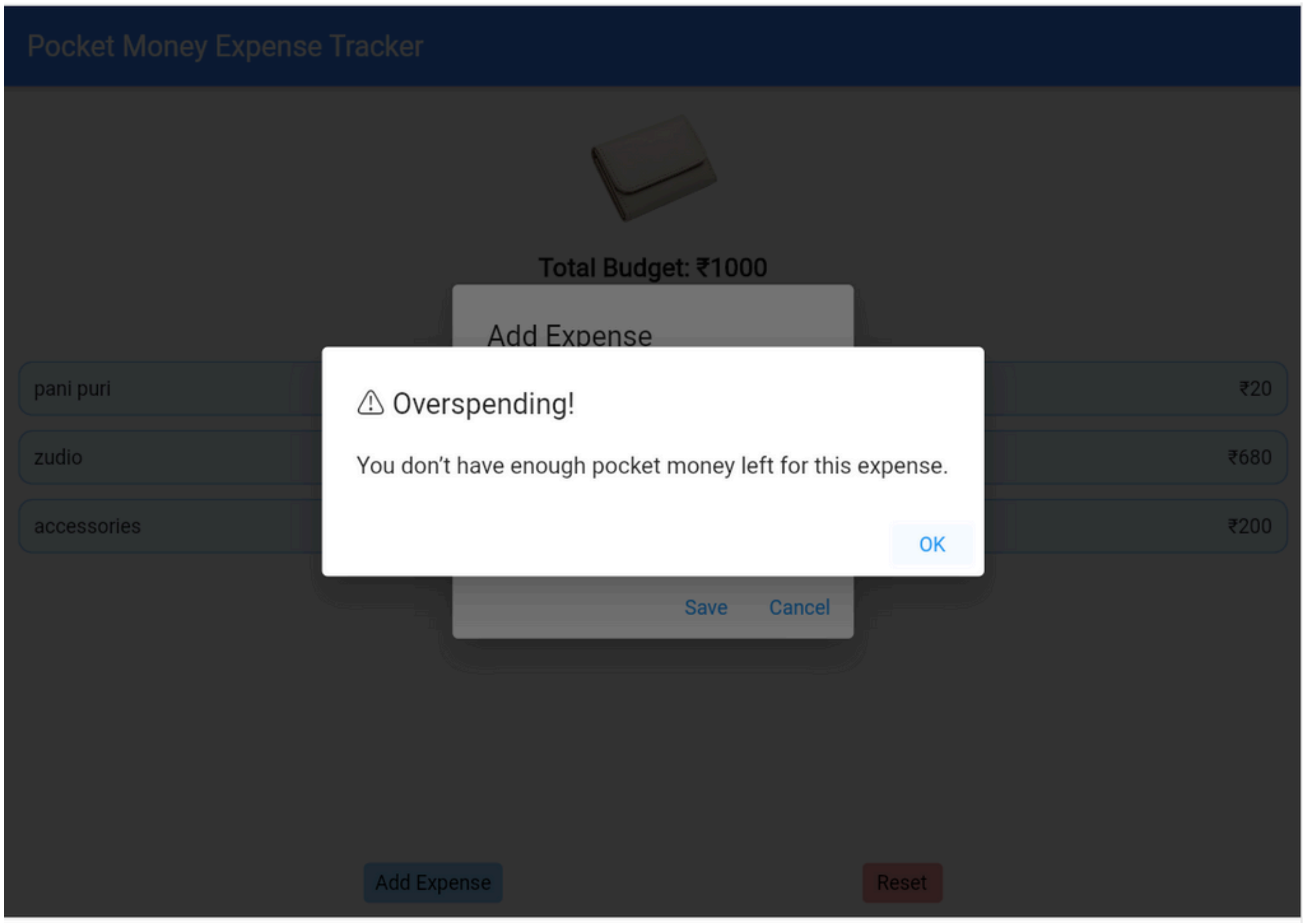
Add Expense

Item Name: pani puri

Cost (₹): 20

Save Cancel

Add Expense Reset


Pocket Money Expense Tracker

Total Budget: ₹1000

pani puri	₹20
zudio	₹680
accessories	₹200

⚠ Overspending!

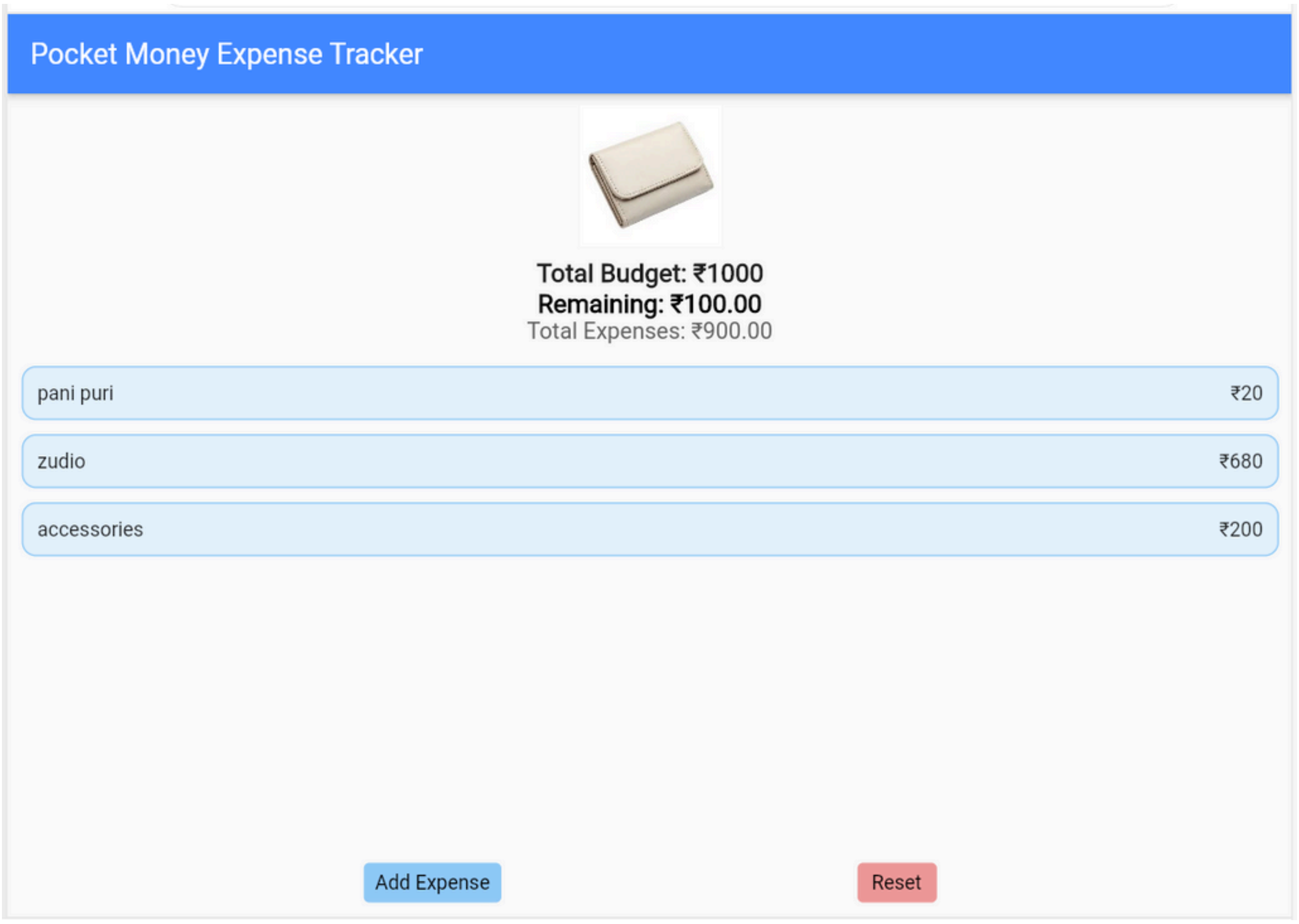
You don't have enough pocket money left for this expense.

OK

Add Expense

Save Cancel

Add Expense Reset



Pocket Money Expense Tracker

Total Budget: ₹1000
Remaining: ₹100.00
Total Expenses: ₹900.00

pani puri	₹20
zudio	₹680
accessories	₹200