

```

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

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

class Expense {
  String title;
  double amount;
  DateTime date;

  Expense({required this.title, required this.amount, required this.date});
}

class ExpenseTrackerApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Expense Tracker',
      theme: ThemeData(primarySwatch: Colors.indigo),
      home: ExpenseHomePage(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class ExpenseHomePage extends StatefulWidget {
  @override
  _ExpenseHomePageState createState() => _ExpenseHomePageState();
}

class _ExpenseHomePageState extends State<ExpenseHomePage> {
  List<Expense> expenses = [];

  void addExpense(String title, double amount) {
    var newExp = Expense(
      title: title,
      amount: amount,
      date: DateTime.now(),
    );
    setState(() => expenses.add(newExp));
  }

  void openAddDialog() {
    var title = '';
    var amount = '';

    showDialog(
      context: context,
      builder: (_) => AlertDialog(
        title: Text('Add Expense'),
        content: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            TextField(
              decoration: InputDecoration(labelText: 'Title'),
              onChanged: (val) => title = val,
            ),
            TextField(
              decoration: InputDecoration(labelText: 'Amount'),
              keyboardType: TextInputType.number,
              onChanged: (val) => amount = val,
            ),
          ],
        ),
        actions: [
          TextButton(

```

```

        onPressed: () {
          if (title.isNotEmpty && double.tryParse(amount) != null) {
            addExpense(title, double.parse(amount));
          }
          Navigator.of(context).pop();
        },
        child: Text('Add'),
      )
    ],
  ),
);
}

List<BarChartGroupData> getChartData() {
  Map<String, double> dailyTotals = {};
  for (var exp in expenses) {
    var date = "${exp.date.day}/${exp.date.month}";
    dailyTotals[date] = (dailyTotals[date] ?? 0) + exp.amount;
  }

  var index = 0;
  return dailyTotals.entries.map((entry) {
    return BarChartGroupData(
      x: index++,
      barRods: [
        BarChartRodData(toY: entry.value, color: Colors.indigo, width: 16),
      ],
    );
  }).toList();
}

@override
Widget build(BuildContext context) {
  var chartData = getChartData();

  return Scaffold(
    appBar: AppBar(title: Text('Expense Tracker')),
    floatingActionButton: FloatingActionButton(
      onPressed: openAddDialog,
      child: Icon(Icons.add),
    ),
    body: Column(
      children: [
        SizedBox(
          height: 200,
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: BarChart(
              BarChartData(
                titlesData: FlTitlesData(show: false),
                barGroups: chartData,
              ),
            ),
          ),
        ),
      ],
    ),
    expanded: Expanded(
      child: ListView.builder(
        itemCount: expenses.length,
        itemBuilder: (_, i) => ListTile(
          title: Text(expenses[i].title),
          subtitle: Text(expenses[i].date.toString()),
          trailing:
            Text("₹${expenses[i].amount.toStringAsFixed(2)}"),
        ),
      ),
    ),
  ),
);

```

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