11. Develop an application that tracks our daily Expenses and get a report chart.

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

void main() {
  runApp(ExpenseTrackerApp());
}

class Expense {
  final String title;
  final double amount;
  final 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(),
    );
  }
}

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

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

  void _addExpense(String title, double amount) {
    final newExp = Expense(
      title: title,
      amount: amount,
      date: DateTime.now(),
    );
```

```
  setState(() {
    _expenses.add(newExp);
  });
}

void _openAddDialog() {
  String title = '';
  String amount = '';
  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      title: Text('Add Expense'),
      content: Column(
        mainAxisSize: MainAxisSize.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) {
```

```
    String date = "${exp.date.day}/${exp.date.month}";
    dailyTotals[date] = (dailyTotals[date] ?? 0) + exp.amount;
  }

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

@override
Widget build(BuildContext context) {
  final 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(
          child: ListView.builder(
            itemCount: _expenses.length,
            itemBuilder: (ctx, i) => ListTile(
              title: Text(_expenses[i].title),
```

```
              subtitle: Text(_expenses[i].date.toString()),
              trailing: Text("₹${_expenses[i].amount.toStringAsFixed(2)}"),
            ),
          ),
        )
      ],
    ),
  );
 }
}
```

```
dependencies:
  flutter:
    sdk: flutter
  fl_chart: ^0.66.0
```