CashCompass

A financial management app made with Flutter

A PROJECT REPORT

Submitted by

Tuhin Kanti Pal

AT

ARDENT

COMPUTECH PVT. LTD.

High-End Technology Training and Project

ARDENT Computech Pvt Ltd.

Bona Fide Certificate

Certified that this project report on **CashCompass**, a financial management app made with Flutter is the bona fide work of **Tuhin Kanti Pal** who carried out the project work under my supervision.

Signature of Mentor

Anindya Mukherjee

Ardent Computech Pvt. Ltd.

Kolkata, West Bengal

Acknowledgement

The success of the **CashCompass** project is deeply attributed to the unwavering mentorship of **ANINDYA MUKHERJEE** from Ardent Computech Pvt. Ltd. His invaluable insights and suggestions have been instrumental in shaping this endeavour, and I am profoundly grateful for his guidance.

Tuhin Kanti Pal

Computer Science & Engineering - 3rd Year, 5th Semester, 2023

Bengal Institute of Technology and Management

INDEX of CashCompass Project

₹

CashCompass

Project Description

Overview:

In today's fast-paced world, managing personal finances can be a challenge. With myriad expenses and sources of income, it's easy to lose track of where your money is going. Enter CashCompass, a state-of-the-art mobile application designed to be your personal financial guide. Built with the robust Flutter framework, CashCompass offers a seamless experience across devices, ensuring that you have a clear view of your financial landscape wherever you go.

Features:

1. **Intuitive Dashboard:** At a glance, view your total income, expenses, and net balance for the month.
2. **Expense Tracking:** Log daily expenses across various categories like food, transportation, entertainment, and more.
3. **Income Monitoring:** Record your sources of income, whether it's from your monthly salary, freelance gigs, or other avenues. Set reminders for expected payments and track your earnings over time.
4. **Data Security:** Your financial data is sensitive, that’s why all of your data is stored on your device. No external servers are involved.
5. **Multiple currency:** Supports multiple currencies seamlessly.

Benefits:

- **Financial Clarity:** With all your financial data in one place, gain a clearer understanding of your spending habits and areas for improvement.
- **Time-Saving:** No more manual logging or sifting through piles of receipts. CashCompass streamlines the process, making financial management a breeze.

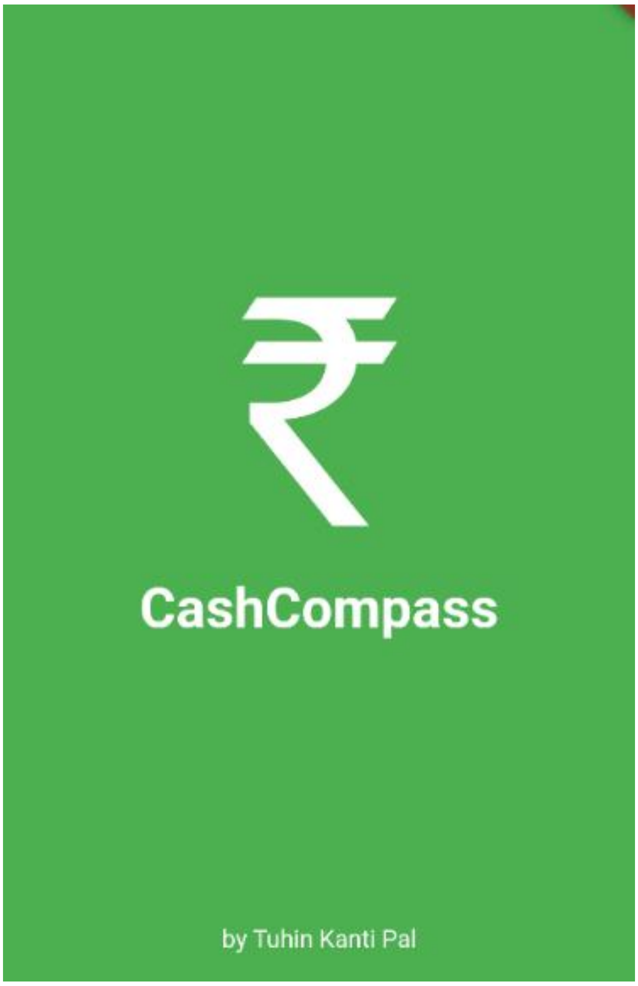
Walkthrough

CashCompass is meticulously designed to offer a user-centric experience. From the moment the app is launched to the logging of each financial transaction, every element is crafted to ensure simplicity, clarity, and efficiency. The journey through the app is intuitive, making financial management not just a necessity but a pleasure.

The Splash Screen

Because First Impressions Matter ❤️‍🩹

- **Design:** As the first point of contact with the user, the splash screen showcases a vibrant and dynamic design that embodies the essence of CashCompass. The app's logo is prominently displayed with name CashCompass.
- **Duration:** The splash screen lasts for 3 seconds, ensuring it's neither too brief nor overly prolonged. This duration is optimized to give users a glimpse of the brand while ensuring a swift transition to the main dashboard.
- **Transition:** As the app initializes, the splash screen seamlessly fades into the main dashboard, ensuring a smooth user experience.

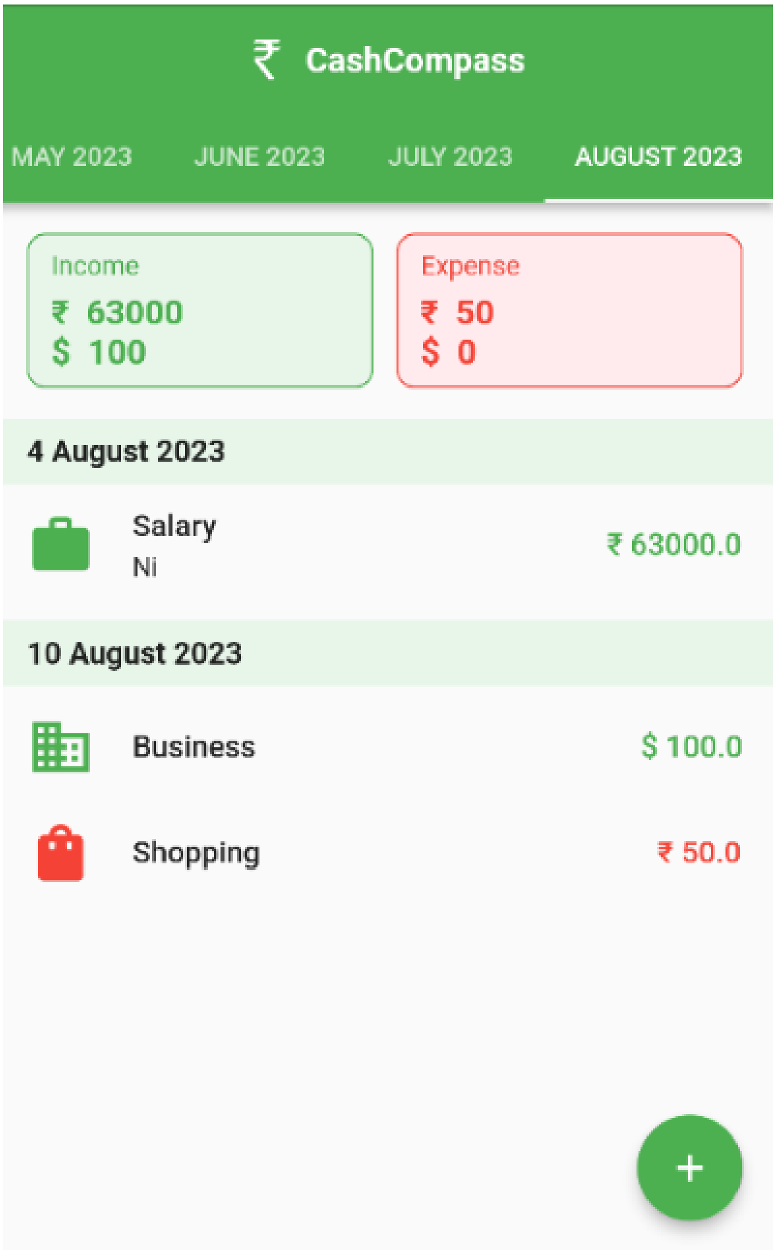


SPLASH SCREEN

Main Dashboard with FAB (Floating Action Button)

Your Financial Overview at a Glance 📈

- **Dashboard Layout:** The main dashboard is the heart of CashCompass. It provides users with an immediate overview of their financial status. Key metrics such as total income, expenses, and all the transactions group by date.
- **Floating Action Button (FAB):** Positioned intuitively, the FAB serves as a quick access point for users to add new transactions. Its design ensures it stands out, yet doesn't obstruct the view of the dashboard's main content.
- **Navigation:** Users can easily navigate to other months by swipe gestures.
- **Transaction group by currency:** If user added multiple currencies, then each currency will be displayed individually.



Income
₹ 63000
$ 100

Expense
₹ 50
$ 0

4 August 2023

Salary
Ni
₹ 63000.0

10 August 2023

Business $ 100.0

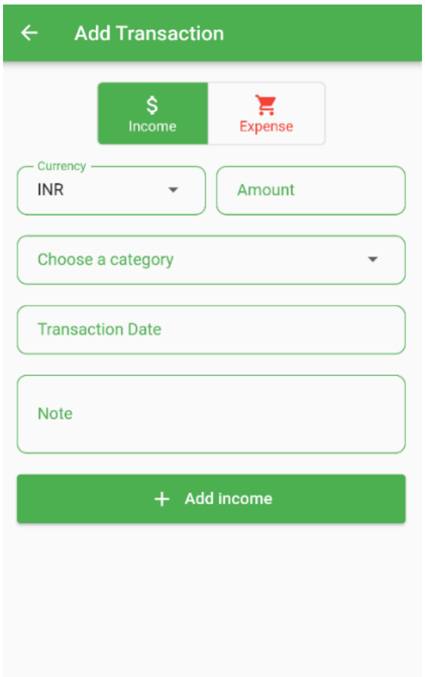
Shopping ₹ 50.0

DASHBOARD

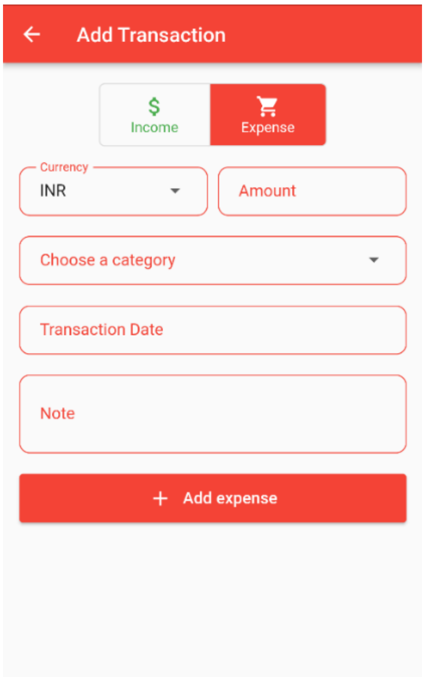
Expense and Income Add or Update View

Simplifying Financial Entries 🤘

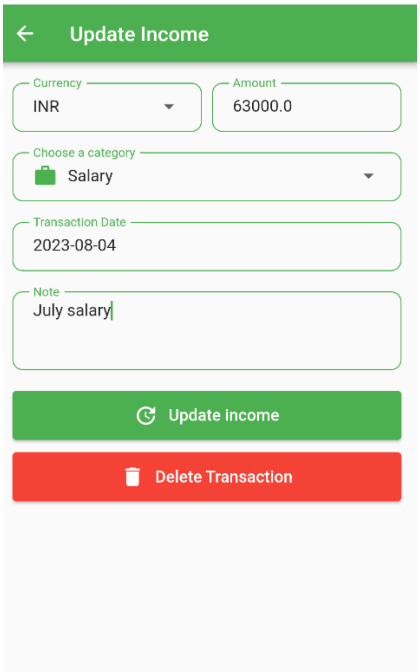
- **Unified View:** Whether adding a new expense, income, or updating an existing entry, the process is streamlined in a unified view. This ensures consistency and reduces the learning curve for users.
- **Intuitive Fields:** The form fields are clearly labelled, with placeholders and examples where necessary. This ensures users can quickly input their data without confusion.
- **Category Selection:** Users can choose from predefined categories or add their own. Icons next to each category provide a visual cue, making the selection process faster.
- **Notes:** Users have the option to add notes to their entries, providing context and aiding in future references.
- **Date & Time Picker:** A user-friendly date and time picker ensures accurate logging of each transaction.
- **Save & Update:** Once all details are filled in, a prominent 'Save' or 'Update' button confirms the action. Feedback, such as a success message or animation, reassures the user of their action's completion. Additionally delete button action delete an existing transaction.



ADD INCOME



ADD EXPENSE



Currency: INR

Amount: 63000.0

Choose a category: Salary

Transaction Date: 2023-08-04

Note: July salary

UPDATE TRANSACTION

Tooling Overview

CashCompass used cutting edge technologies, which makes it perfect to maintain the app in future. CashCompass utilizes these technologies.

Tooling and Technologies:

1. Flutter:

Flutter, a UI toolkit from Google, is the primary framework used for building the CashCompass app. It allows for the creation of natively compiled applications from a single codebase, ensuring a consistent experience across both Android and iOS platforms.

Advantages:

- **Performance:** Flutter's Dart platform ensures high performance by compiling to native ARM code and has a consistent behaviour across platforms.
- **Rich Widgets:** Flutter's extensive widget library was leveraged to create a visually appealing and user-friendly interface.
- **Hot Reload:** This feature of Flutter accelerates the development process by allowing developers to instantly view changes without fully restarting the app.

2. SQLite:

SQLite is a C-language library that provides a lightweight disk-based database. It doesn’t require a separate server process. For CashCompass, SQLite serves as the primary data storage solution, holding records of user expenses, incomes, and other related data.

Advantages:

- **Serverless and Self-contained:** SQLite doesn't require a separate server to operate, which makes it a perfect choice for mobile applications where simplicity and efficiency are paramount.
- **Reliable:** It offers atomic commit and rollback capabilities, ensuring data integrity.
- **Lightweight:** Given its serverless architecture and small footprint, it's ideal for mobile applications where resources are limited.
- **Secure:** SQLite provides robust data protection mechanisms, ensuring that user data remains confidential and safe from potential breaches.

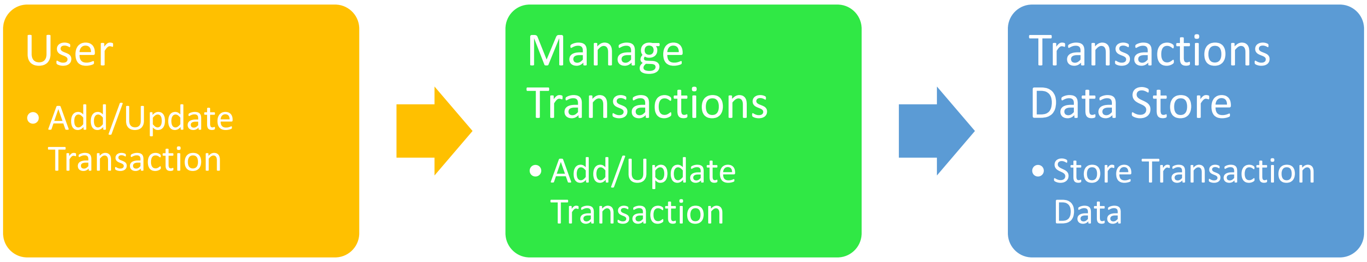
Development Workflow

1. **Design Phase:** Utilizing Flutter's widget library to draft the initial UI/UX design of the app, ensuring it aligns with the app's core concept and user needs.
2. **Development:** Writing the Dart code in Flutter, integrating with SQLite for data operations. Using packages like **sqflite** to facilitate the integration.
3. **Deployment:** Compiling the Flutter codebase into native Android and iOS applications with GitHub Actions.

Entity Relationship Diagram (ERD)

Transactions		
PK	id	INTEGER
	currency	TEXT
	amount	REAL
	date	TEXT
	category	TEXT
	note	TEXT
	Type	TEXT

Data Flow Diagram



User

- Add/Update Transaction

Manage Transactions

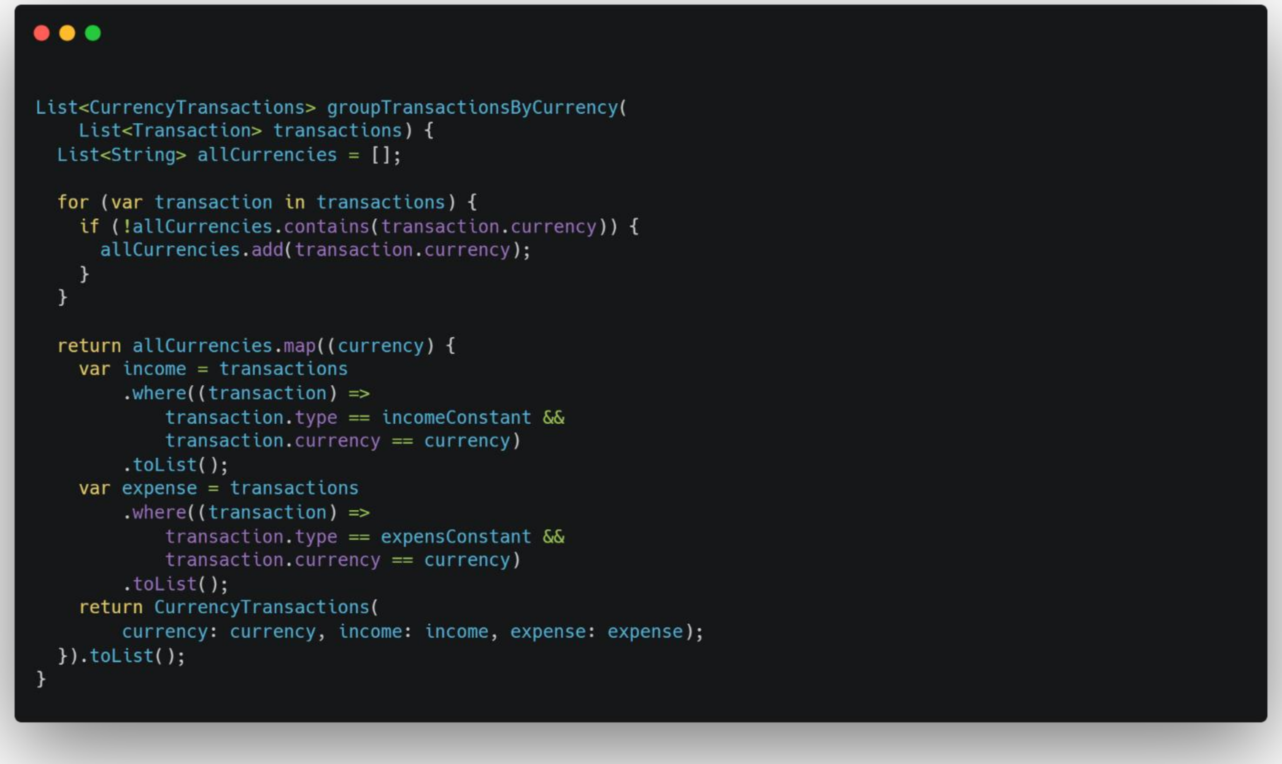
- Add/Update Transaction

Transactions Data Store

- Store Transaction Data

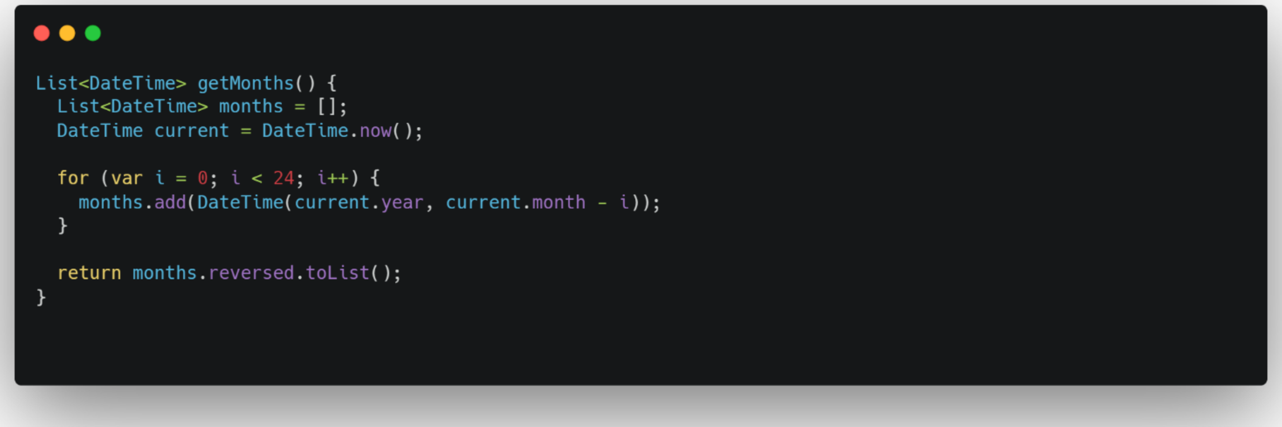
Code Snippets

Get total amount by currencies:



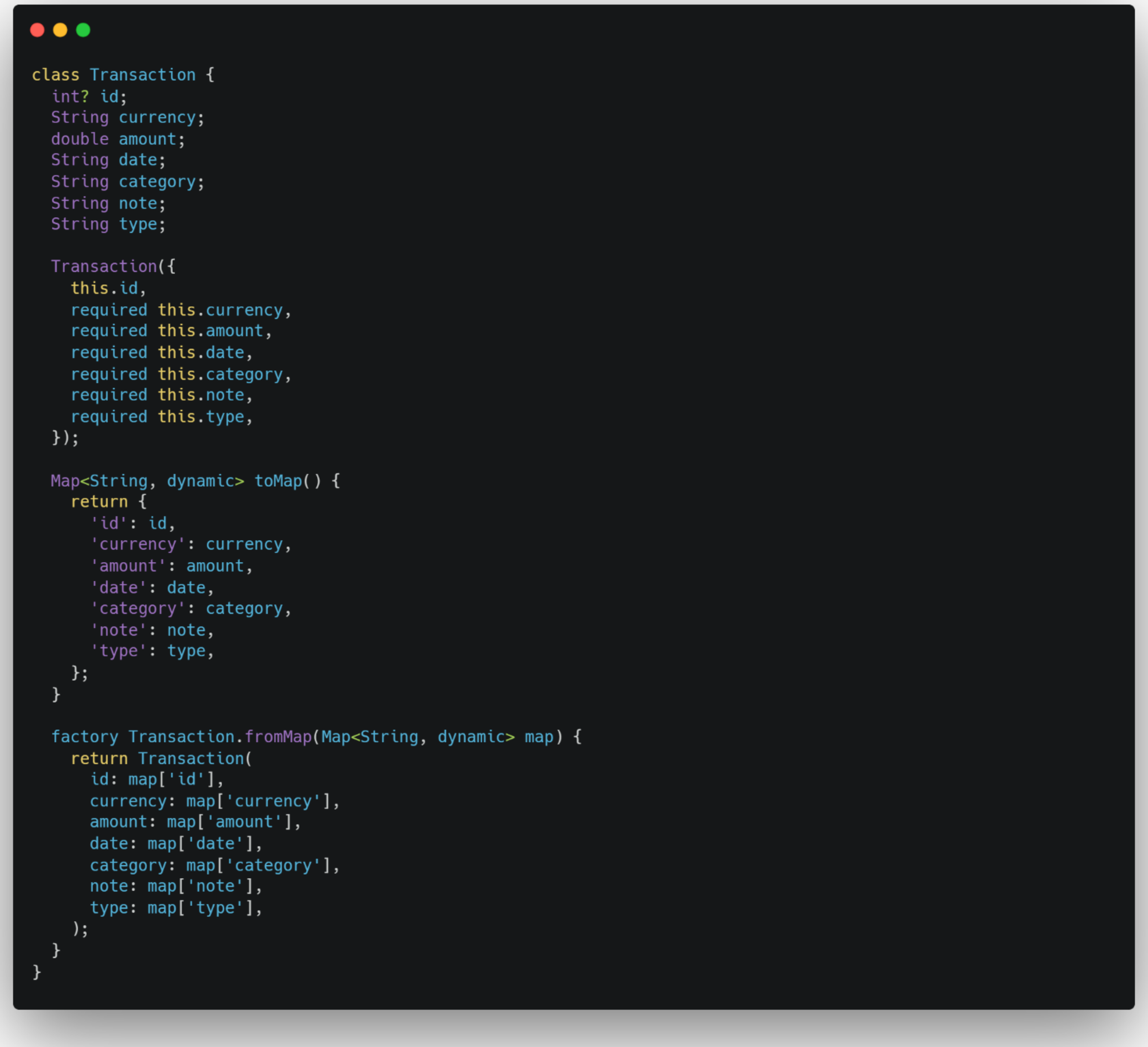
```
List<CurrencyTransactions> groupTransactionsByCurrency(
    List<Transaction> transactions) {
  List<String> allCurrencies = [];

  for (var transaction in transactions) {
    if (!allCurrencies.contains(transaction.currency)) {
      allCurrencies.add(transaction.currency);
    }
  }

  return allCurrencies.map((currency) {
    var income = transactions
        .where((transaction) =>
            transaction.type == incomeConstant &&
            transaction.currency == currency)
        .toList();
    var expense = transactions
        .where((transaction) =>
            transaction.type == expensConstant &&
            transaction.currency == currency)
        .toList();
    return CurrencyTransactions(
        currency: currency, income: income, expense: expense);
  }).toList();
}
```

Create months for tab menu:



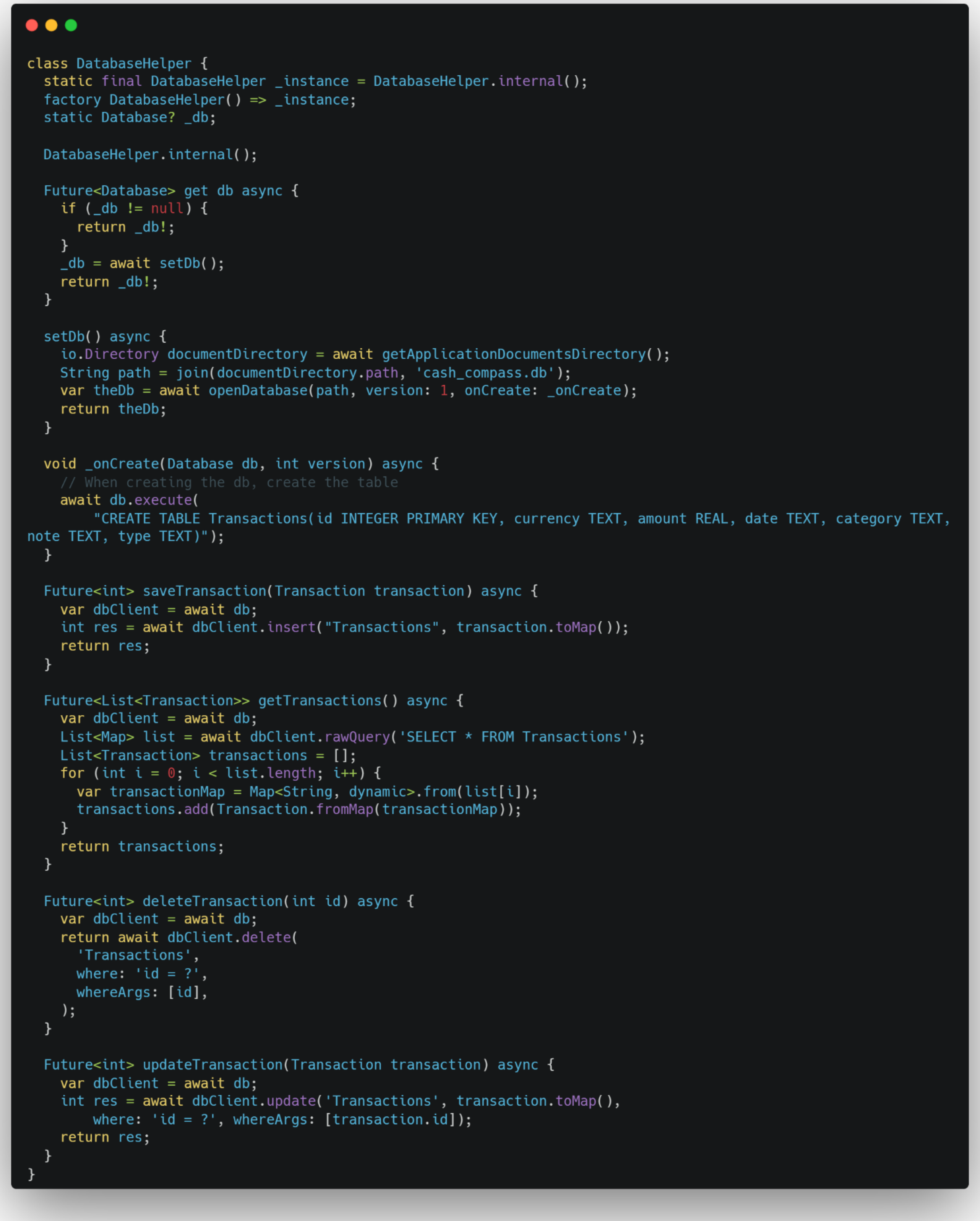
```
List<DateTime> getMonths() {
  List<DateTime> months = [];
  DateTime current = DateTime.now();

  for (var i = 0; i < 24; i++) {
    months.add(DateTime(current.year, current.month - i));
  }

  return months.reversed.toList();
}
```

The transaction class:



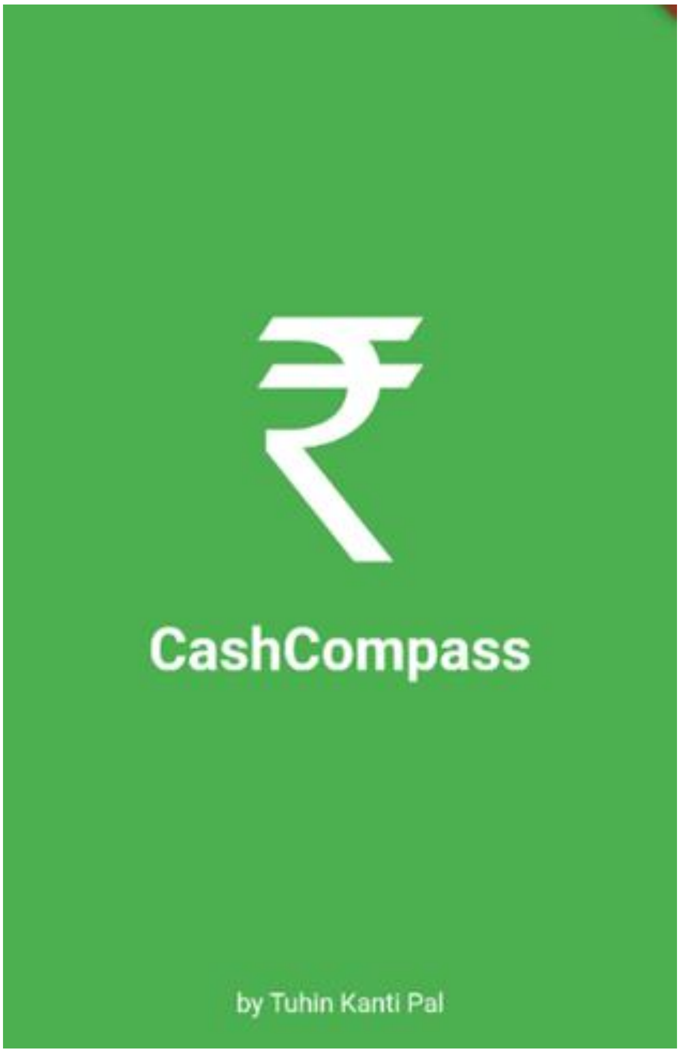
```dart
class Transaction {
  int? id;
  String currency;
  double amount;
  String date;
  String category;
  String note;
  String type;

  Transaction({
    this.id,
    required this.currency,
    required this.amount,
    required this.date,
    required this.category,
    required this.note,
    required this.type,
  });

  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'currency': currency,
      'amount': amount,
      'date': date,
      'category': category,
      'note': note,
      'type': type,
    };
  }

  factory Transaction.fromMap(Map<String, dynamic> map) {
    return Transaction(
      id: map['id'],
      currency: map['currency'],
      amount: map['amount'],
      date: map['date'],
      category: map['category'],
      note: map['note'],
      type: map['type'],
    );
  }
}
```

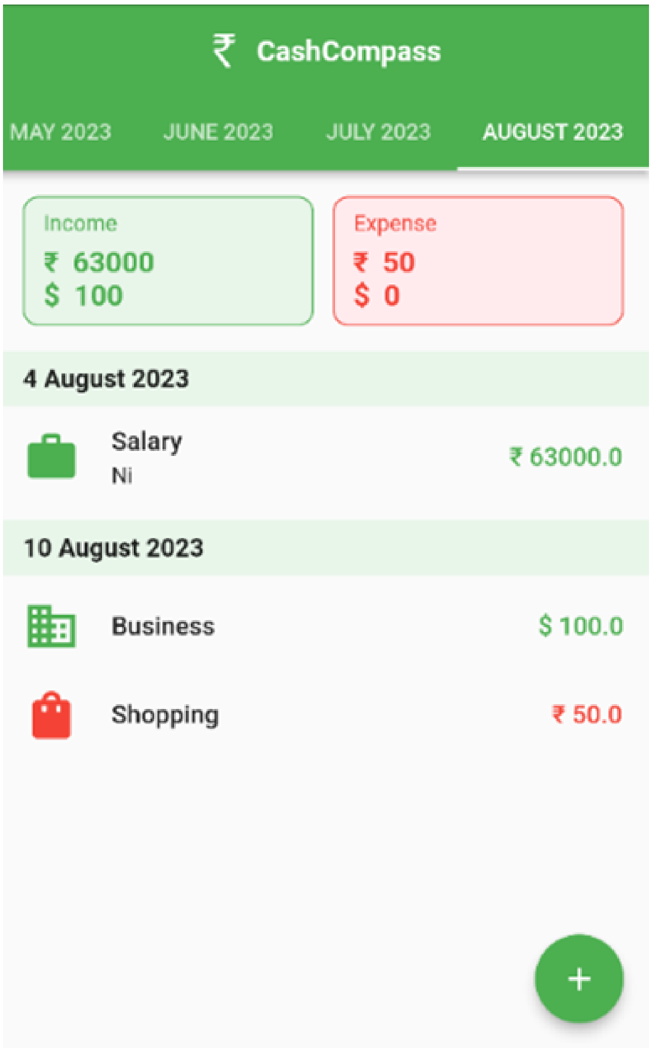
Database helper:



```
class DatabaseHelper {
  static final DatabaseHelper _instance = DatabaseHelper.internal();
  factory DatabaseHelper() => _instance;
  static Database? _db;

  DatabaseHelper.internal();

  Future<Database> get db async {
    if (_db != null) {
      return _db!;
    }
    _db = await setDb();
    return _db!;
  }

  setDb() async {
    io.Directory documentDirectory = await getApplicationDocumentsDirectory();
    String path = join(documentDirectory.path, 'cash_compass.db');
    var theDb = await openDatabase(path, version: 1, onCreate: _onCreate);
    return theDb;
  }

  void _onCreate(Database db, int version) async {
    // When creating the db, create the table
    await db.execute(
        "CREATE TABLE Transactions(id INTEGER PRIMARY KEY, currency TEXT, amount REAL, date TEXT, category TEXT,
note TEXT, type TEXT)");
  }

  Future<int> saveTransaction(Transaction transaction) async {
    var dbClient = await db;
    int res = await dbClient.insert("Transactions", transaction.toMap());
    return res;
  }

  Future<List<Transaction>> getTransactions() async {
    var dbClient = await db;
    List<Map> list = await dbClient.rawQuery('SELECT * FROM Transactions');
    List<Transaction> transactions = [];
    for (int i = 0; i < list.length; i++) {
      var transactionMap = Map<String, dynamic>.from(list[i]);
      transactions.add(Transaction.fromMap(transactionMap));
    }
    return transactions;
  }

  Future<int> deleteTransaction(int id) async {
    var dbClient = await db;
    return await dbClient.delete(
      'Transactions',
      where: 'id = ?',
      whereArgs: [id],
    );
  }

  Future<int> updateTransaction(Transaction transaction) async {
    var dbClient = await db;
    int res = await dbClient.update('Transactions', transaction.toMap(),
        where: 'id = ?', whereArgs: [transaction.id]);
    return res;
  }
}
```

Screenshots




₹ CashCompass

MAY 2023 | JUNE 2023 | JULY 2023 | AUGUST 2023

Income
₹ 63000
$ 100

Expense
₹ 50
$ 0

4 August 2023

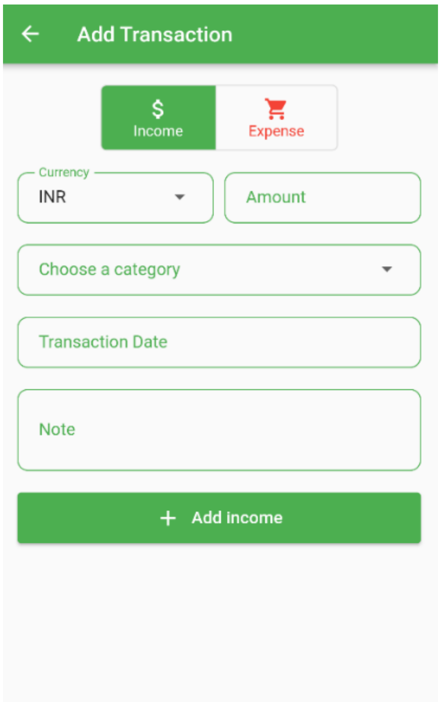
Salary
Ni
₹ 63000.0

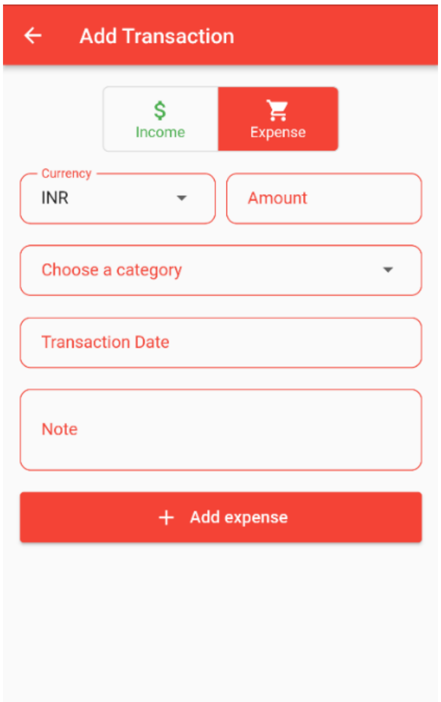
10 August 2023

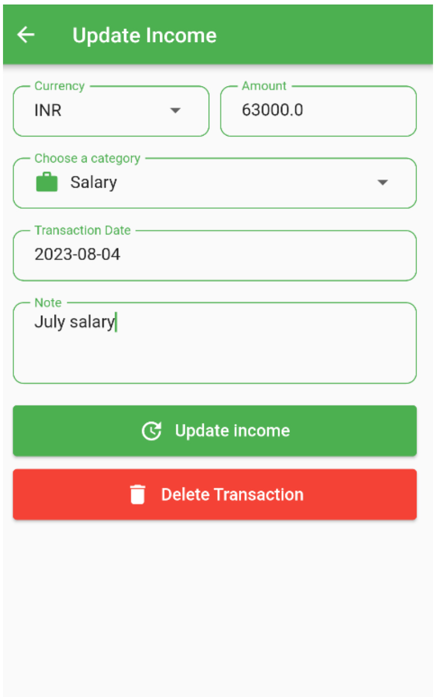
Business $ 100.0

Shopping ₹ 50.0

+






← Update Income

Currency: INR

Amount: 63000.0

Choose a category: Salary

Transaction Date: 2023-08-04

Note: July salary

Conclusion

In the realm of personal financial management, CashCompass stands out as a beacon of innovation and reliability. Harnessing the power of Flutter's dynamic UI toolkit and SQLite's unparalleled data storage efficiency, it promises a user experience that is both visually captivating and functionally superior. Users are not just getting an app; they're getting a holistic financial companion that marries aesthetics with performance. In a digital age where user expectations are constantly evolving, CashCompass sets a new benchmark, redefining what excellence looks like in the world of financial applications.

References

- Pub.dev
- Packages documentation
- docs.flutter.dev
- chat.openai.com