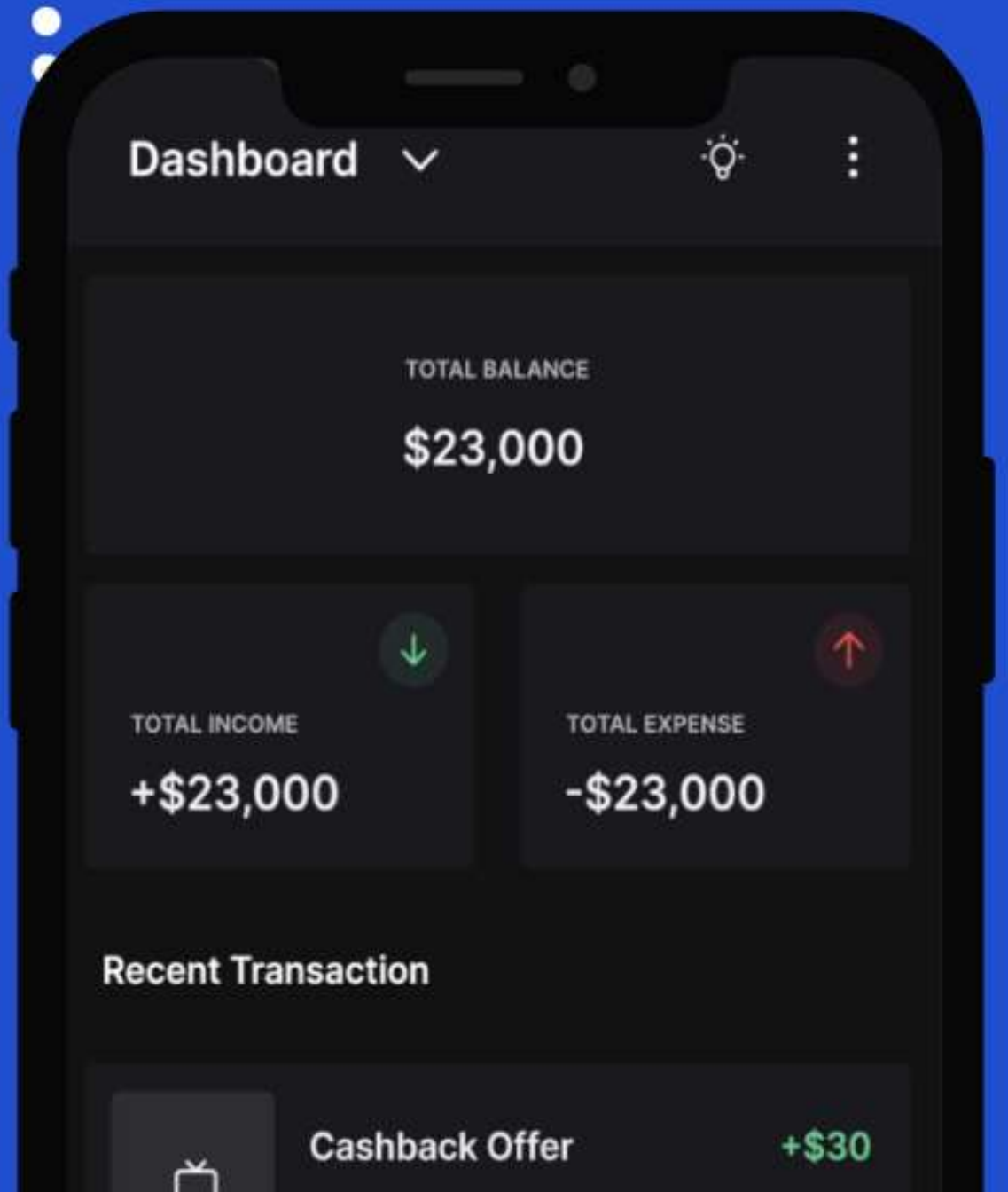
Development Overview

PocketBills : A smart Expense Tracker

Exploring the intersection of innovation and user experience in billing solutions

Pranav TJ, Harsha, Nithilan, Sravan
Developers



Tech Stack Overview

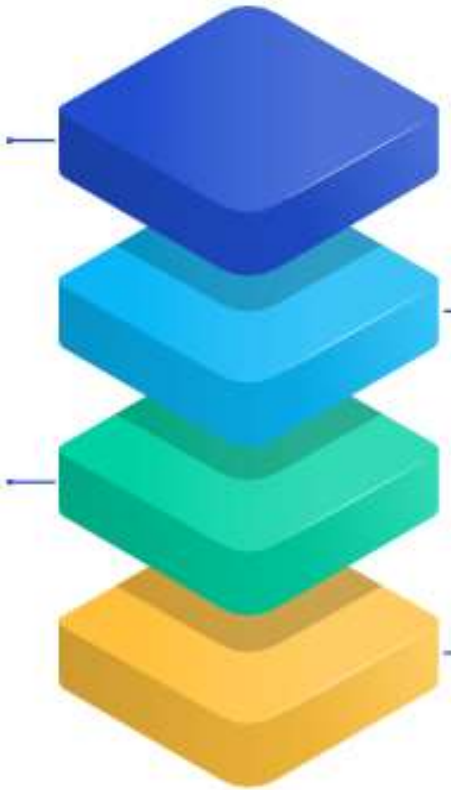
Key technologies utilized in PocketBills for enhanced functionality.

Kotlin Programming Language

Utilized for developing the PocketBills app, ensuring modern features and performance.

Room Database

Provides efficient local data storage capabilities tailored for the app's needs.



MVVM Architecture

Adopted for separating UI and business logic, enhancing app maintainability and scalability.

CSV Export Feature

Allows users to share data easily by exporting to CSV format, improving user experience.

Key Features of PocketBills

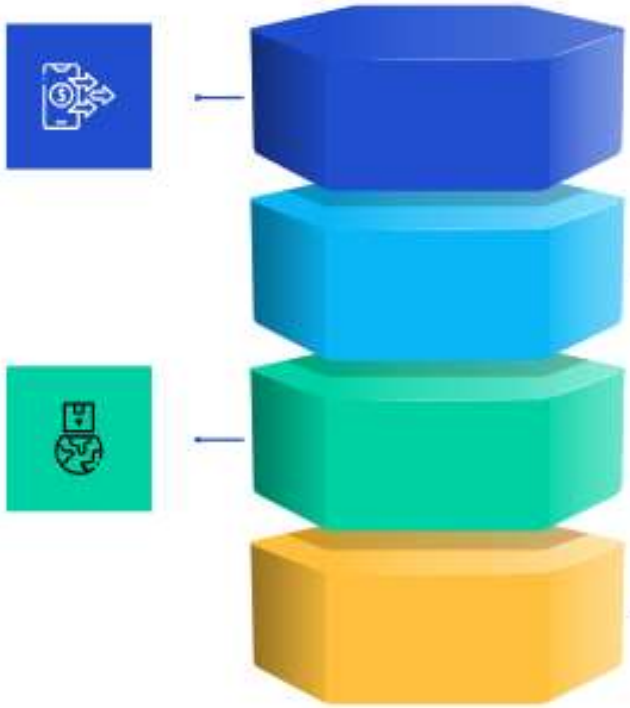
Explore the functionalities that make PocketBills an essential expense tracking tool.

Add, Edit, and Delete Transactions

Easily manage your expenses with straightforward options to add, edit, or delete transactions anytime.

CSV Export at End of Day

Export your daily transaction data in CSV format for easier tracking and reporting.



Daily Summary

Receive a daily summary of your financial activities to keep track of your expenses effectively.

Clean and User-Friendly UI

Enjoy an intuitive interface designed for seamless navigation and a better user experience.

MVVM Architecture Overview

Model Layer

The **Model** layer manages the data and business logic of the application. It communicates with data sources, handles data storage, and encapsulates the data structure.

View Layer

The **View** layer displays the user interface and is responsible for presenting the data to the user. It receives input and triggers the corresponding actions in the ViewModel.

ViewModel Layer

The **ViewModel** acts as a bridge between the Model and View layers. It retrieves data from the Model and prepares it for presentation in the View, ensuring separation of concerns.

Data Binding

Data Binding is a key feature that links the View and ViewModel, allowing automatic synchronization of data changes between them, enhancing user experience and reducing boilerplate code.

Separation of Concerns

The **Separation of Concerns** principle is a core aspect of MVVM, promoting modularity and making the application easier to test, maintain, and scale by decoupling components.

CSV Export Feature Highlights

Key advantages of the CSV export functionality integrated with Kotlin I/O for daily transaction management.

Automatic CSV File Generation

The feature automatically creates a CSV file summarizing daily transactions, simplifying record keeping.

Easy Sharing and Backup

Exported CSV files can be easily shared or backed up, enhancing data portability and security.

Kotlin I/O Library Integration

Implemented using Kotlin I/O libraries to efficiently handle file operations within the Android app.

Meet the Team

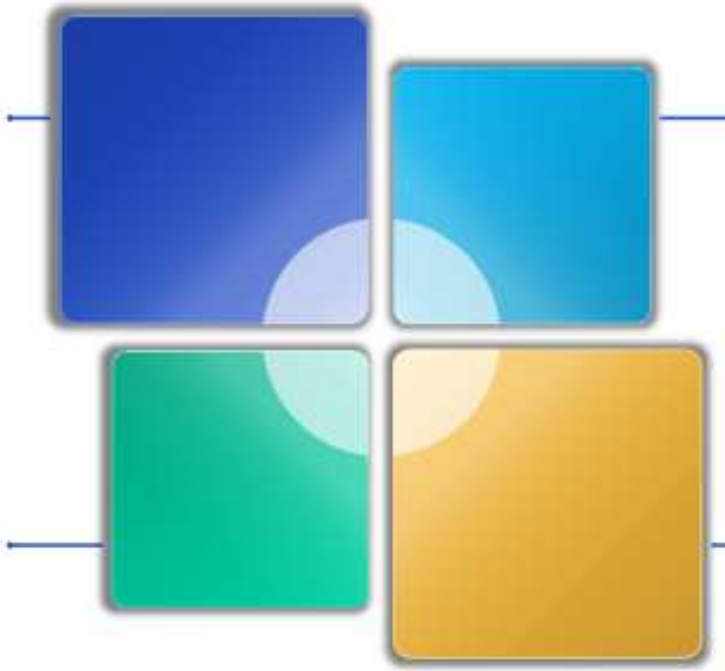
Key Contributors to PocketBills

Pranav T.J. (Team Leader)

Handled business logic and wrote core Kotlin code.

Harsha

Co-developed business logic and Kotlin implementation.

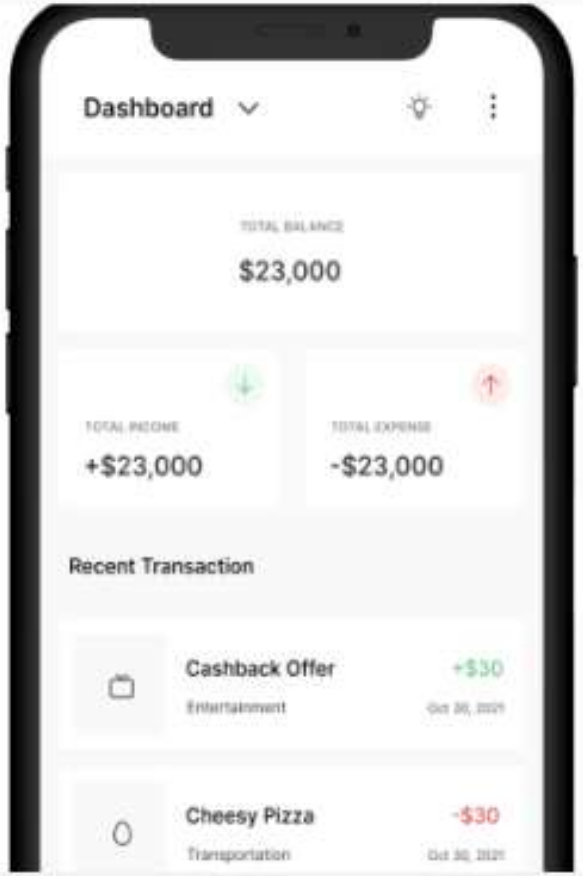


Nithilan

Focused on UI/UX design, XML layouts, and drawable assets.

Sravan

Contributed to UI layout design and visuals.



Home Dashboard Overview

Displays the main dashboard with an overview of daily expenses, budgets, and quick access to features.

Conclusion

Key Insights for PocketBills

01 **Financial Awareness**

PocketBills equips users with tools for enhanced financial awareness and quick expense tracking.

02 **Intuitive Interface**

The application features an intuitive interface that simplifies daily expense management.

03 **Smart Export Features**

Users can leverage smart export features for easy report generation and sharing.

04 **Planned Enhancements**

Future updates will focus on integrating machine learning for predictive budgeting.

Future Enhancements

Join us on our journey to transform your financial experience with PocketBills

Discover the future of PocketBills with innovative features

Call-to-action ›