RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

on

Mobile Application Development

INT410: Intra Institutional Internship

STUDENT NAME : ARPITHA C AND ANANYA ANU

USN :1MS24AD014 AND 1MS24AD008

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
MSR Nagar, MSRIT Post, Bangalore-560054

August - 2025

RAMAIAH
Institute of Technology

RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

This is to certify that Ms ANANYA ANU (USN:1MS24AD008) and Mr./Ms. ARPITHA C (USN: 1MS24AD014), a students of Bachelor of Engineering, have successfully completed, 24 Hours: from 04.08.2025 to 12.08.2025 Intra Institutional Internship in Mobile Application Development from the Department of Computer Science & Engineering, M S Ramaiah Institute of Technology, Bangalore.

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

Signature of the Student with Date **Signature of the Faculty Co-Ordinator**

Signature of Head of the Department

OVERVIEW OF INTERNSHIP ACTIVITIES

DATE	DAY	NAME OF THE TOPIC COMPLETED
05.08.25	Tuesday	Introduction to dart and flutter
06.08.25	Wednesday	Briefing on dart and flutter programs
07.08.25	Thursday	Scaffold and its applications
09.08.25	Saturday	Introduction to Animations and navigation button
11.05.25	Monday	Firebase studio and internal assessment on dart an flutter
12.05.25	Tuesday	Report submission

TABLE OF CONTENTS

OVERALL VIEW OF THE PROJECT IN TERMS OF IMPLEMENTATION

Introduction

The “Wallet Mate” project was developed to help users manage their personal finances by providing a clear breakdown of their spending habits. The app aggregates data from various sources and platforms, allowing users to view detailed reports of their expenses. The internship aimed to apply software engineering principles in a real-world development environment, with an emphasis on mobile application design, testing, and deployment.

Internship Activities / Work Undertaken

During the internship, the following activities were undertaken:

- Requirement gathering and feature planning for the “Wallet Mate” app.
- Designing the app architecture and selecting suitable development tools (e.g., Flutter/Dart, Firebase).
- Developing user interface screens for expense tracking, reports, and category management.
- Integrating database functionality for storing and retrieving user expense data.
- Implementing analytics and chart visualization for spending patterns.
- Testing the application for bugs and optimizing performance.

Learning Outcomes

The internship offered the following learning outcomes:

- Improved proficiency in mobile app development frameworks (e.g., Flutter/Dart).
- Gained practical knowledge of database integration and cloud storage.
- Enhanced understanding of UI/UX principles for mobile applications.
- Strengthened problem-solving and debugging skills.
- Learned to work effectively in a collaborative team environment.

Challenges Faced & Solutions

- **Data synchronization issues** between devices – resolved by implementing real-time database updates using Firebase.
- **Performance bottlenecks** in large expense datasets – optimized queries and implemented lazy loading.
- **UI consistency across devices** – achieved through responsive design techniques.

Purpose

A Flutter mobile app for tracking personal expenses and managing related data (contacts, categories, and reports) with theme toggle and simple analytics.

Architecture

The app uses:

- **Flutter** for UI
- **Provider** (ChangeNotifier) for state management
- **Named routes** for navigation
- **Modular file structure** for maintainability

Folder Structure

Perl

CopyEdit

lib/

```
│
├── main.dart              # App entry point
│
├── models/                # Data structures (plain Dart classes)
│   ├── transaction_item.dart
│   ├── contact.dart
│   └── category.dart
│
```

```
├── state/
│
├── screens/                # Full-page UI
│   ├── home_screen.dart
│   ├── add_transaction_screen.dart
│   ├── add_contact_screen.dart
│   ├── add_category_screen.dart
│   ├── reports_screen.dart
│   ├── settings_screen.dart
│   └── about_screen.dart
│
└── widgets/                # Reusable UI components
    ├── app_drawer.dart
    └── transaction_list.dart
```

File-by-File Implementation

main.dart

- Wraps the entire app in a ChangeNotifierProvider<WalletModel>.
- Defines **routes** for navigation.
- Holds **theme state** (light/dark) using ThemeMode.
- Loads HomeScreen as the default.

models

1. **transaction_item.dart**
    - Fields: id (UUID), amount, date, category, note, contact.
    - Constructor with required parameters.

2. **contact.dart**
    - Fields: id, name, phone.
3. **category.dart**
    - Fields: id, name, icon (uses IconData).

state/wallet_model.dart

- Extends ChangeNotifier.
- Holds:
    - List<TransactionItem> transactions
    - List<Contact> contacts
    - List<Category> categories
- Methods:
    - addTransaction(TransactionItem t)
    - removeTransaction(String id)
    - addContact(Contact c)
    - addCategory(Category cat)
- Calls notifyListeners() after each update to refresh UI.

---

screens

1. **home_screen.dart**
    - Shows recent transactions in TransactionList widget.
    - Includes navigation drawer (AppDrawer).
    - Floating action button for adding a transaction.
2. **add_transaction_screen.dart**
    - Form with:
        - Amount (TextField)

        - Category (Dropdown)
        - Contact (Dropdown)
        - Date picker
        - Notes
    - On submit → Calls walletModel.addTransaction().
3. **add_contact_screen.dart**
    - Form with name + phone.
    - On submit → walletModel.addContact().
4. **add_category_screen.dart**
    - Form with category name + icon picker.
    - On submit → walletModel.addCategory().
5. **reports_screen.dart**
    - Summarizes spending by category.
    - Uses simple chart (placeholder for fl_chart).
6. **settings_screen.dart**
    - Toggle switch for theme (light/dark).
    - Option to reset all data (clear lists in WalletModel).
7. **about_screen.dart**
    - Displays app version, developer info, and short description.

widgets

1. **app_drawer.dart**
    - Navigation menu with links to all major screens.
    - Uses ListTile + Navigator.pushNamed().

2. **transaction_list.dart**
    - ListView displaying transactions (amount, category, date, note).
    - Receives transactions as a prop from HomeScreen.

Data Flow

1. **User interacts** (fills a form → taps submit).
2. **UI screen** calls method from WalletModel via Provider.
3. WalletModel updates its internal list and calls notifyListeners().
4. All **listening widgets/screens** rebuild automatically.

Navigation

- Handled via **named routes** in main.dart:

dart

CopyEdit

```
routes: {
  '/': (context) => HomeScreen(),
  '/add_transaction': (context) => AddTransactionScreen(),
  '/add_contact': (context) => AddContactScreen(),
  '/add_category': (context) => AddCategoryScreen(),
  '/reports': (context) => ReportsScreen(),
  '/settings': (context) => SettingsScreen(),
  '/about': (context) => AboutScreen(),
}
```

Theme Handling

- main.dart holds a ThemeMode state variable.
- SettingsScreen toggle updates it.
- Uses Flutter's built-in light/dark themes.

Current Storage

- **In-memory lists** only (lost on app restart).
- Can be upgraded to **Hive** or **SQLite** later.

CODE OF MAIN MODULES

Folder structure

```
lib/
│
├── main.dart              # App entry point
├── models/
│   ├── category.dart
│   ├── contact.dart
│   └── transaction_item.dart
│
├── state/
│   └── wallet_model.dart
│
├── screens/
│   ├── app_root.dart
│   ├── home_screen.dart
│   ├── transactions_screen.dart
│   ├── add_transaction_screen.dart
│   ├── transaction_details_screen.dart
│   ├── contacts_screen.dart
│   ├── add_edit_contact_screen.dart
│   ├── contact_detail_screen.dart
│   ├── reports_screen.dart
│   ├── settings_screen.dart
│   └── categories_screen.dart
│
└── widgets/
```

```
   ├── app_drawer.dart
   └── transactions_list_view.dart
```

1.main.dart

```
import 'package:flutter/material.dart';
import 'state/wallet_model.dart';
import 'screens/app_root.dart';

void main() {
  runApp(const WalletMateApp());
}

class WalletMateApp extends StatefulWidget {
  const WalletMateApp({super.key});

  @override
  State<WalletMateApp> createState() => _WalletMateAppState();
}

class _WalletMateAppState extends State<WalletMateApp> {
  bool isDarkTheme = false;

  void toggleTheme() {
    setState(() => isDarkTheme = !isDarkTheme);
  }
```

```
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'WalletMate',
      debugShowCheckedModeBanner: false,
      theme: isDarkTheme ? ThemeData.dark() : ThemeData(primarySwatch: Colors.teal),
      home: AppRoot(toggleTheme: toggleTheme, isDarkTheme: isDarkTheme),
    );
  }
}
```

2.models/category.dart

```
enum Category {
  food, clothing, loans, transport, entertainment, utilities, others
}
extension CategoryExt on Category {
  String get name {
    switch (this) {
      case Category.food: return 'Food';
      case Category.clothing: return 'Clothing';
      case Category.loans: return 'Loans';
      case Category.transport: return 'Transport';
      case Category.entertainment: return 'Entertainment';
      case Category.utilities: return 'Utilities';
      default: return 'Others';
```

```
  }
  }
}
```

3.models/contact.dart

```
class Contact {
  String id;
  String name;
  String phone;
  Contact({
    required this.id,
    required this.name,
    required this.phone,
  });
}
```

4.models/transaction_item.dart

```
import 'category.dart';

class TransactionItem {
  String id;
  String contactId;
  double amount;
  DateTime date;
```

```
  bool incoming;
  String note;
  Category category;
  TransactionItem({
    required this.id,
    required this.contactId,
    required this.amount,
    required this.date,
    required this.incoming,
    this.note = '',
    required this.category,
  });
}
```

5.state/wallet_model.dart

```
import 'package:flutter/material.dart';
import '../models/contact.dart';
import '../models/transaction_item.dart';
class WalletModel extends ChangeNotifier {
  final List<Contact> _contacts = [];
  final List<TransactionItem> _transactions = [];
  List<Contact> get contacts => List.unmodifiable(_contacts);
  List<TransactionItem> get transactions => List.unmodifiable(_transactions);
  void addContact(Contact c) {
_contacts.add(c);
```

```
 notifyListeners();
}

void editContact(String id, Contact updated) {
  final i = _contacts.indexWhere((c) => c.id == id);
  if (i != -1) _contacts[i] = updated;
  notifyListeners();
}
  void deleteContact(String id) {
  _contacts.removeWhere((c) => c.id == id);
  _transactions.removeWhere((t) => t.contactId == id);
  notifyListeners();
}
  void addTransaction(TransactionItem t) {
  _transactions.add(t);
  notifyListeners();
}
 void editTransaction(String id, TransactionItem updated) {
  final i = _transactions.indexWhere((t) => t.id == id);
  if (i != -1) _transactions[i] = updated;
 notifyListeners();
}
 void deleteTransaction(String id) {
  _transactions.removeWhere((t) => t.id == id);
  notifyListeners();
```

```
  } double get balance {
    double b = 0;
    for (var t in _transactions) {
      b += t.incoming ? t.amount : -t.amount;
    }
    return b;
  }
}
final WalletModel walletModel = WalletModel();
```

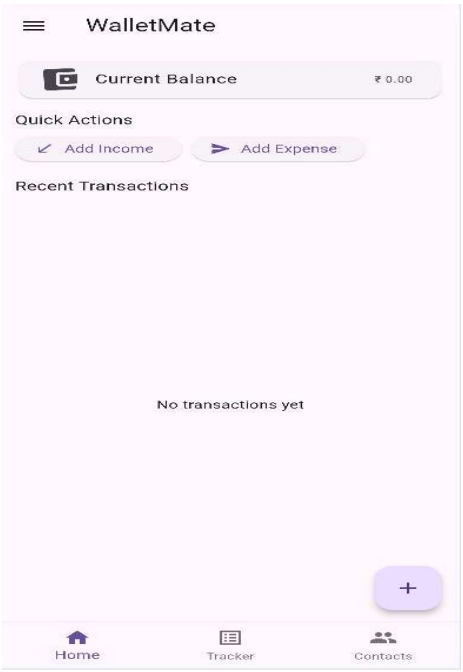
6.Example screen (screens/app_root.dart)

```
import 'package:flutter/material.dart';
import '../widgets/app_drawer.dart';
import 'home_screen.dart';
import 'transactions_screen.dart';
import 'contacts_screen.dart';
class AppRoot extends StatefulWidget {
  final VoidCallback toggleTheme;
  final bool isDarkTheme;
  const AppRoot({super.key, required this.toggleTheme, required this.isDarkTheme});
 @override
  State<AppRoot> createState() => _AppRootState();
}
class _AppRootState extends State<AppRoot> {
  int _selectedIndex = 0;
  static const List<Widget> _screens = [
```

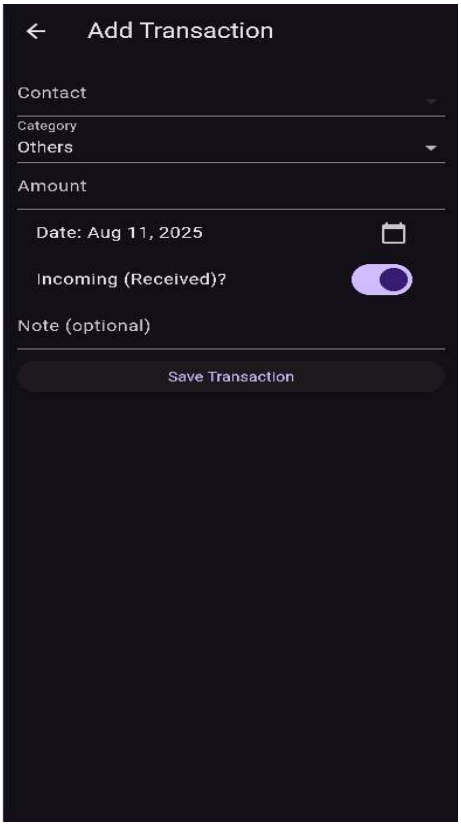
```
  HomeScreen(),
  TransactionsScreen(),
  ContactsScreen(),
];
void _onItemTapped(int index) {
  setState(() => _selectedIndex = index);
}
@override
Widget build(BuildContext context) {
  return Scaffold(
  appBar: AppBar(title: const Text('WalletMate')),
    drawer: AppDrawer(
      toggleTheme: widget.toggleTheme,
      isDarkTheme: widget.isDarkTheme,
    ),
    body: _screens[_selectedIndex],
    bottomNavigationBar: BottomNavigationBar(
      items: const [
        BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
        BottomNavigationBarItem(icon: Icon(Icons.list_alt), label: 'Tracker'),
        BottomNavigationBarItem(icon: Icon(Icons.people), label: 'Contacts'),
      ],
      currentIndex: _selectedIndex,
      onTap: _onItemTapped,
    ),
```

```
  floatingActionButton: FloatingActionButton(
      onPressed: () {
        if (_selectedIndex == 2) {
          Navigator.pushNamed(context, '/add_contact');
        } else {
          Navigator.pushNamed(context, '/add_transaction');
        }
      },
      child: const Icon(Icons.add),
    ),
  );
 }
}
```

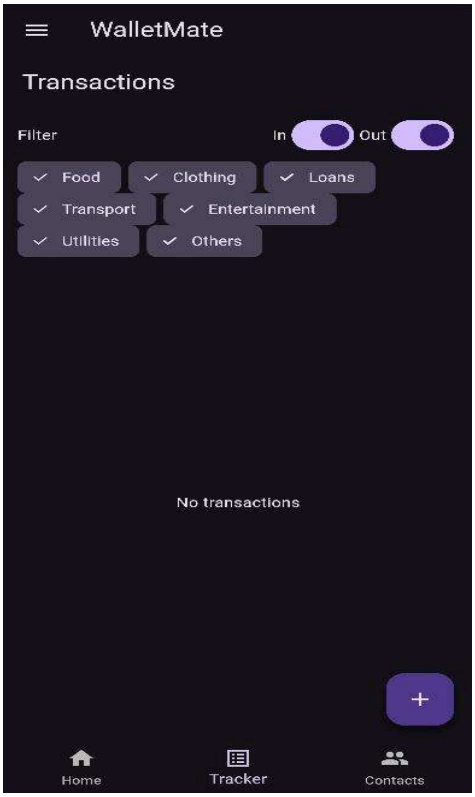
RESULT SNAPSHOTS



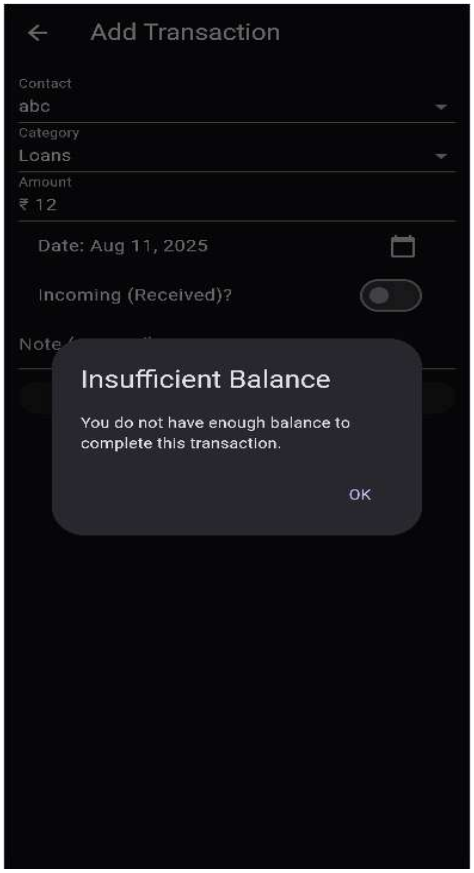
the wallet next to current balance moves



added category while doing a transaction and not while creating contact



alert dialog if you try to send more money than you currently have



filters based on category

CONCLUSION

This report documents the internship experience at **MS Ramaiah Institute of technology[CSE Department]**, focusing on the design and development of the "Wallet Mate" mobile application. The application allows users to track their expenses across various platforms, categorize them, and visualize spending patterns. The internship provided practical exposure to mobile app development tools, database integration, and user interface design.

The "Wallet Mate" project was a valuable learning experience, combining software development skills with financial management concepts. The internship provided a hands-on understanding of mobile app lifecycle, from requirement analysis to deployment. Future improvements could include AI-based spending predictions and integration with payment gateways for automated tracking.