UI design - Flutter

LABORATORY PROJECT

ON

Smart Budget App

Submitted

In partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology in

Computer Science Engineering (Artificial Intelligence & Machine Learning)

By

GUGULAVATH NAVEEN 23261A6615

KHYATI CHINTHA 23261A6620

Under the guidance of

Mrs. Divija Kakanuru

ASSISTANT PROFESSOR

DEPARTMENT OF EMERGING TECHNOLOGIES

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

Department of Emerging Technologies

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

GANDIPET, HYDERABAD-500075, INDIA

November 2025

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated To Jawaharlal Nehru Technological University Hyderabad)
GANDIPET, HYDERABAD-500075, Telangana

CERTIFICATE

This is to certify that the software engineering entitled "**Smart Budget App**" is being submitted by GUGULAVATH NAVEEN 23261A6615 and KHYATI CHINTHA 23261A6620 in partial fulfilment of the requirement for the **Laboratory Project** in **EMERGING TECHNOLOGIES** is a record of bonafide work carried out by them. The results of the investigations enclosed in this report have been verified and found satisfactory.

Mrs. Divija Kakanuru
Assistant Professor
ET Department

INDEX

ABSTRACT

The Smart Budget App is a mobile-based personal finance management system developed using the Flutter framework and Dart programming language, designed to provide users with an intuitive platform for tracking their daily income, expenses, and overall financial balance. In modern society, effective financial management has become essential due to the increasing complexity of personal spending patterns and the need for better decision-making. Traditional budgeting practices, such as manual record-keeping and spreadsheet tracking, are error-prone, inefficient, and lack real-time insights. This project addresses these limitations by offering a smart, automated, and visually appealing solution that supports users in making financially responsible choices.

The core objective of the Smart Budget App is to simplify the budgeting process by offering features such as real-time transaction logging, financial summaries, interactive dashboards, and intelligent savings recommendations. The app dynamically calculates total income, total expenses, and remaining balance, enabling users to understand their financial situations at a glance. One of the key innovations in this system is the integrated Smart Tips module, which evaluates the proportion of income spent and provides actionable guidance based on user spending behavior. For instance, if expenses exceed 80% of the income, high-severity alerts warn the user of excessive spending. If spending remains within moderate limits, cautionary advice is offered. Users who manage their finances efficiently receive positive reinforcement, thereby establishing healthy financial habits.

From a technical perspective, the application leverages Flutter's widget-based architecture, Material Design components, and built-in state management mechanisms to create a highly responsive and interactive user interface. The use of the intl package further enables the app to format dates in a readable manner, enhancing user readability and international compatibility. Transaction manipulation—including additions, deletions, and real-time updates—is handled through efficient UI rebuilding techniques using setState(). The app runs entirely offline, ensuring privacy, quick access, and independence from external servers or login systems.

The Smart Budget App's architecture is designed with scalability in mind. Its modular structure allows for future enhancements such as cloud synchronization, data analytics, category-based expense charts, and artificial intelligence-powered financial predictions. The project serves not only as a functional personal budgeting tool but also as a demonstration of best practices in Flutter mobile app development, UI/UX design, and applied financial computing.

Overall, this project contributes to a practical, user-friendly, and intelligent financial tracking system that empowers individuals to take deliberate control of their money. It bridges the gap between traditional budgeting methods and modern intelligent technologies, offering an accessible and efficient system suitable for students, adults, and anyone seeking greater financial awareness.

INTRODUCTION

In an era where financial awareness and responsibility play a vital role in personal well-being, the ability to track income and expenses effectively has become a necessity rather than a luxury. The Smart Budget App, developed using Flutter and Dart, provides a simple yet intelligent platform to manage, analyze, and control personal finances effortlessly.

Its primary goal is not only to record transactions but also to analyze spending patterns and offer personalized guidance for savings and budgeting. Unlike conventional methods such as notebooks or Excel sheets, the app introduces an automated approach with features like categorization, summaries, smart tips, and visual feedback to give users a clear view of their financial health.

Guided by simplicity, accessibility, and intelligence, the app ensures ease of use, cross-platform support, and actionable recommendations. A clean dashboard with color-coded metrics (green for income, red for expenses, blue for balance) enhances user experience, while Smart Tips act as an embedded advisor. Technically, it leverages Flutter's dynamic widgets and the intl package for localization, functioning securely offline without server-side storage.

Beyond budgeting, the app fosters financial discipline and informed decision-making. Future enhancements may include machine learning for spending predictions, advanced visualization tools, and cloud synchronization.

In conclusion, the Smart Budget App merges mobile convenience with intelligent analysis, empowering users to manage spending responsibly and achieve long-term financial well-being.

EXISTING SYSTEM

In the context of personal finance management, various systems have been developed over the years to help individuals organize their income, expenses, and savings. However, most suffer from limitations that reduce efficiency, accessibility, and user engagement. Before the Smart Budget App, users relied on manual record-keeping, spreadsheets, or basic mobile apps. While these offered some assistance, they lacked intelligent analysis, real-time interaction, and user-friendly design.

Traditional methods like notebooks are time-consuming, error-prone, and lack automation, while spreadsheets require technical knowledge and fail to provide mobility or smart insights. Mobile apps improved accessibility but often demanded internet connectivity, subscriptions, or external data storage, raising privacy concerns. Many focused only on recording transactions without offering personalized feedback, and their interfaces were either too complex or too simplistic.

In summary, existing systems show major drawbacks:

1. Lack of automation
2. Poor user interface design
3. Dependence on internet or external databases
4. Absence of smart analytics
5. Data privacy concerns
6. Limited real-time interactivity

Thus, while current tools provide basic tracking, they fail to meet modern needs for automation, personalization, privacy, and intelligence — a gap the Smart Budget App is designed to fill.

PROPOSED SYSTEM

The Smart Budget App proposes a modern, intelligent, and user-friendly solution for personal finance management that overcomes the limitations of existing systems. It helps users record, analyze, and optimize financial activities in real time through a single mobile application. Built with Flutter and Dart, it ensures a seamless cross-platform experience with a clean and responsive interface.

The system eliminates manual calculations and static spreadsheets by automating financial tracking. Users can add income or expense transactions, and the dashboard instantly updates totals and balance using Flutter's state management.

System Objectives

1. Automate financial tracking
2. Provide real-time insights
3. Offer smart savings tips
4. Operate offline for privacy
5. Deliver a simple, engaging UI

System Features

- Dashboard with color-coded indicators (green: income, red: expenses, blue: balance)
- Smart Tips offering personalized financial advice based on expense-to-income ratios
- Easy transaction management with attributes like title, amount, date, and type
- Real-time sorting by date and validated input forms
- Offline accessibility with secure local storage

Technical Implementation

- Flutter with Material Design widgets
- Stateful widgets for dynamic updates
- intl package for date formatting
- Dart Lists/Maps for transaction handling
- Navigator for smooth screen transitions

Advantages Over Existing Systems

- Fully offline, no external dependencies
- Real-time balance updates
- Intelligent analytics with adaptive tips
- Aesthetic, responsive UI for all users
- Privacy ensured with no external storage

Intelligence and Scalability The Smart Tips algorithm provides natural-language financial guidance. Future upgrades may include charts, category-based expense tracking, AI forecasting, and cloud sync.

In summary, the Smart Budget App offers an efficient, modern, and intelligent alternative to traditional finance tools, emphasizing automation, privacy, and user guidance to promote long-term financial stability.

CODE

main.dart

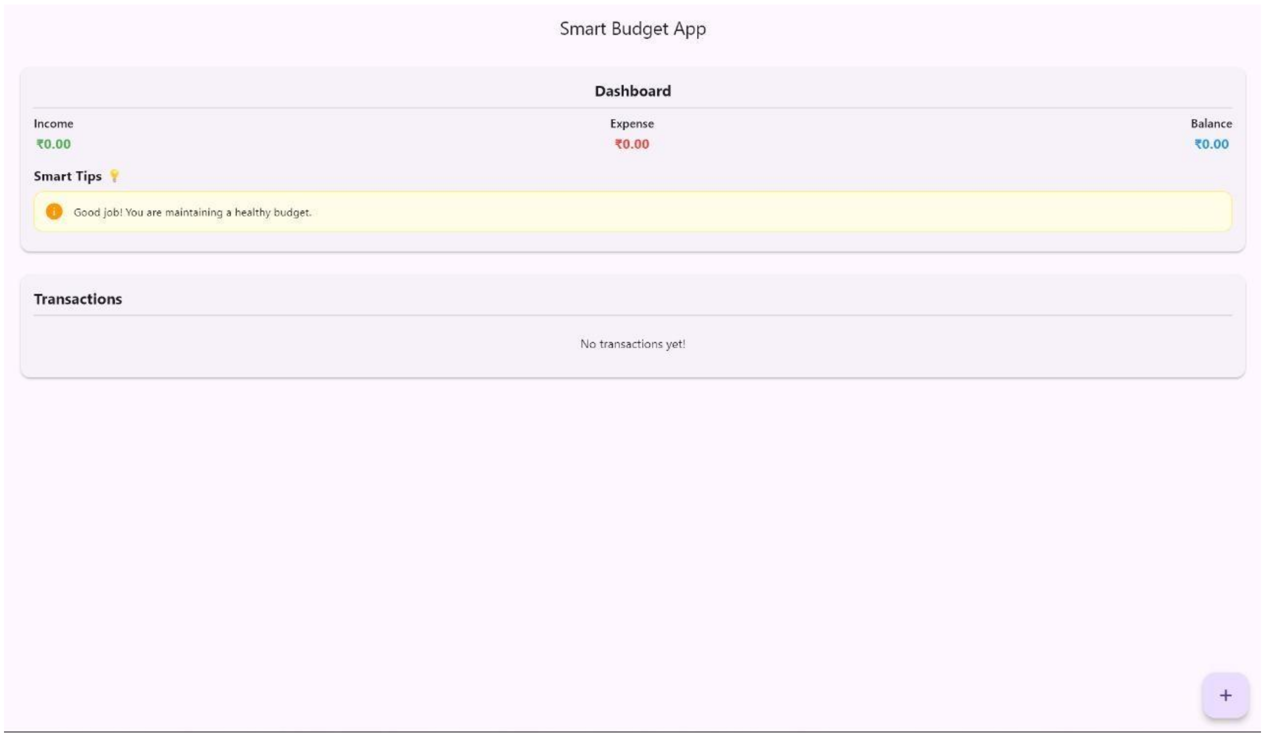
```dart
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
void main() {
  runApp(const SmartBudgetApp());
}
class SmartBudgetApp extends StatelessWidget {
  const SmartBudgetApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Smart Budget App',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.indigo,
        useMaterial3: true,
      ),
      home: const FinanceTrackerScreen(),
    );
  }
}
class FinanceTrackerScreen extends StatefulWidget {
  const FinanceTrackerScreen({super.key});

  @override
  State<FinanceTrackerScreen> createState() => _FinanceTrackerScreenState();
}
class _FinanceTrackerScreenState extends State<FinanceTrackerScreen> {
  final List<Map<String, dynamic>> _transactions = [];
  void _addTransaction(Map<String, dynamic> tx) {
    setState(() {
      _transactions.add(tx);
    });
  }
  void _deleteTransaction(int index) {
    setState(() {
      _transactions.removeAt(index);
    });
  }
  List<Map<String, dynamic>> get _sortedTransactions {
    final tx = [..._transactions];
    tx.sort((a, b) => b['date'].compareTo(a['date']));
    return tx;
  }
  double get _totalIncome => _transactions
      .where((t) => t['type'] == 'Income')
      .fold(0.0, (sum, t) => sum + t['amount']);

  double get _totalExpense => _transactions
      .where((t) => t['type'] == 'Expense')
      .fold(0.0, (sum, t) => sum + t['amount']);
  double get _totalBalance => _totalIncome - _totalExpense;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Smart Budget App'),
        centerTitle: true,
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            _buildDashboard(),
            const SizedBox(height: 20),
            _buildTransactionsList(),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => AddTransactionScreen(
                onAdd: _addTransaction,
              ),
            ),
          );
        },
        child: const Icon(Icons.add),
      ),
    );
  }
  Widget _buildDashboard() {
    return Card(
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
      elevation: 2,
      child: Padding(
        padding: const EdgeInsets.all(16),
        child: Column(
          children: [
            const Text(
            "Dashboard",
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
  ),
  const Divider(),
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      _buildDashboardTile("Income", _totalIncome, Colors.green),
      _buildDashboardTile("Expense", _totalExpense, Colors.red),
      _buildDashboardTile("Balance", _totalBalance, Colors.blue),
    ],
  ),
  const SizedBox(height: 16),
  const Align(
    alignment: Alignment.centerLeft,
    child: Text(
      "Smart Tips 88",
      style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
    ),
  ),
  const SizedBox(height: 8),
  ..._getSavingsTips().map((tip) {
    return Padding(
      padding: const EdgeInsets.only(bottom: 8),
      child: Container(
        padding: const EdgeInsets.all(12),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(12),
          color: tip['severity'] == 'high'
              ? Colors.red.shade50
              : Colors.yellow.shade50,
          border: Border.all(
            color: tip['severity'] == 'high'
                ? Colors.red.shade300
                : Colors.yellow.shade300,
          ),
        ),
        child: Row(
          children: [
          Icon(
              tip['severity'] == 'high'
                  ? Icons.warning
                  : Icons.info,
              color: tip['severity'] == 'high'
                  ? Colors.red
                  : Colors.orange,
            ),
            const SizedBox(width: 12),
            Expanded(
              child: Text(
                    tip['message'] as String,
                    style: const TextStyle(fontSize: 13),
                  ),
                ),
              ],
            ),
          ),
        );
      }).toList(),
    ],
  ),
  ),
 );
}
Widget _buildDashboardTile(String title, double value, Color color) {
 return Column(
  children: [
   Text(
   title,
    style: const TextStyle(fontSize: 14, fontWeight: FontWeight.w500),
   ),
   const SizedBox(height: 4),
   Text(
    "₹${value.toStringAsFixed(2)}",
    style: TextStyle(
     fontSize: 16,
     color: color,
     fontWeight: FontWeight.bold,
    ),
   ),
  ],
 );
}
List<Map<String, String>> _getSavingsTips() {
 final tips = <Map<String, String>>[];
 if (_totalExpense > _totalIncome * 0.8) {
  tips.add({
   'message': 'Your expenses are too high! Try reducing unnecessary spending.',
   'severity': 'high'
  });
 } else if (_totalExpense > _totalIncome * 0.5) {
  tips.add({
   'message': 'You are spending more than half your income. Save more if possible.',
   'severity': 'medium'
  });
 } else {
  tips.add({
   'message': 'Good job! You are maintaining a healthy budget.',
   'severity': 'low'
```

```
    });
  }
  return tips;
}
Widget _buildTransactionsList() {
  final sortedTransactions = _sortedTransactions;
  return Card(
    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
    elevation: 2,
    child: Padding(
      padding: const EdgeInsets.all(16),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Text(
            "Transactions",
            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
          ),
          const Divider(),
          if (sortedTransactions.isEmpty)
            const Center(
              child: Padding(
                padding: EdgeInsets.all(16),
                child: Text("No transactions yet!"),
              ),
            )
          else
            Column(
              children: sortedTransactions.map((t) {
                return Dismissible(
                  key: Key(t['title'].toString()),
                  onDismissed: (_) {
                    _deleteTransaction(
                        sortedTransactions.indexOf(t));
                  },
                  child: ListTile(
                    leading: Icon(
                    t['type'] == 'Income'
                          ? Icons.arrow_downward
                          : Icons.arrow_upward,
                      color: t['type'] == 'Income'
                          ? Colors.green
                          : Colors.red,
                    ),
                    title: Text(t['title']),
                    subtitle: Text(
                      DateFormat.yMMMd().format(t['date']),
                    ),
                    trailing: Text(
                      "₹${t['amount'].toStringAsFixed(2)}",
                      style: TextStyle(
                        color: t['type'] == 'Income'
                            ? Colors.green
                            : Colors.red,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                );
              }).toList(),
            ),
          ],
        ),
      ),
    );
  }
}
class AddTransactionScreen extends StatefulWidget {
  final Function(Map<String, dynamic>) onAdd;

  const AddTransactionScreen({super.key, required this.onAdd});
  @override
  State<AddTransactionScreen> createState() => _AddTransactionScreenState();
}
class _AddTransactionScreenState extends State<AddTransactionScreen> {
  final _formKey = GlobalKey<FormState>();
  final _titleController = TextEditingController();
  final _amountController = TextEditingController();
  String _selectedType = 'Expense';
  void _submitData() {
    if (!_formKey.currentState!.validate()) return;
    final tx = {
      'title': _titleController.text,
      'amount': double.parse(_amountController.text),
      'date': DateTime.now(),
      'type': _selectedType,
    };
    widget.onAdd(tx);
    Navigator.pop(context);
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Add Transaction'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16),
        child: Form(
         key: _formKey,
         child: Column(
         children: [
           TextFormField(
             controller: _titleController,
             decoration: const InputDecoration(labelText: "Title"),
             validator: (value) =>
             value == null || value.isEmpty ? 'Enter a title' : null,
           ),
           TextFormField(
             controller: _amountController,
             decoration: const InputDecoration(labelText: "Amount"),
             keyboardType:
             const TextInputType.numberWithOptions(decimal: true),
             validator: (value) =>
             value == null || value.isEmpty ? 'Enter amount' : null,
           ),
           const SizedBox(height: 16),
           DropdownButtonFormField<String>(
           value: _selectedType,
             items: const [
               DropdownMenuItem(value: 'Income', child: Text('Income')),
               DropdownMenuItem(value: 'Expense', child: Text('Expense')),
             ],
             onChanged: (value) {
               setState(() {
                 _selectedType = value!;
               });
             },
           ),
           const SizedBox(height: 20),
           ElevatedButton(
             onPressed: _submitData,
             child: const Text('Add Transaction'),
           ),
          ],
         ),
        ),
       ),
      );
     }
    }
```

pubspec.yaml

```yaml
name: smart_budget_app
description: A simple smart budget tracker app built with Flutter.
publish_to: 'none' # since it's a local app

version: 1.0.0+1
environment:
  sdk: ">=3.4.0 <4.0.0" # Adjust based on your Flutter version

dependencies:
  flutter:
    sdk: flutter
  # For date formatting
  intl: ^0.19.0
  # Default Flutter icons
  cupertino_icons: ^1.0.8

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^5.0.0

flutter:
  uses-material-design: true
```

OUTPUT



Smart Budget App

Dashboard

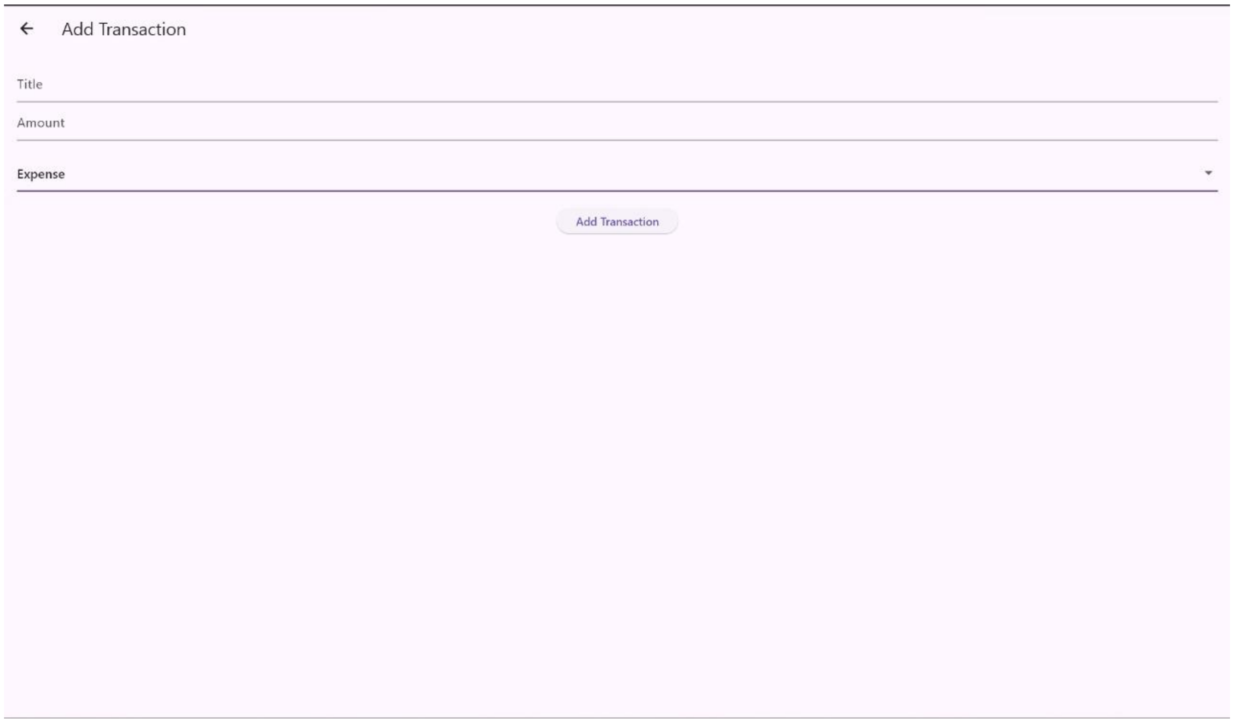
Income ₹0.00 · Expense ₹0.00 · Balance ₹0.00

Smart Tips 💡

Good job! You are maintaining a healthy budget.

Transactions

No transactions yet!



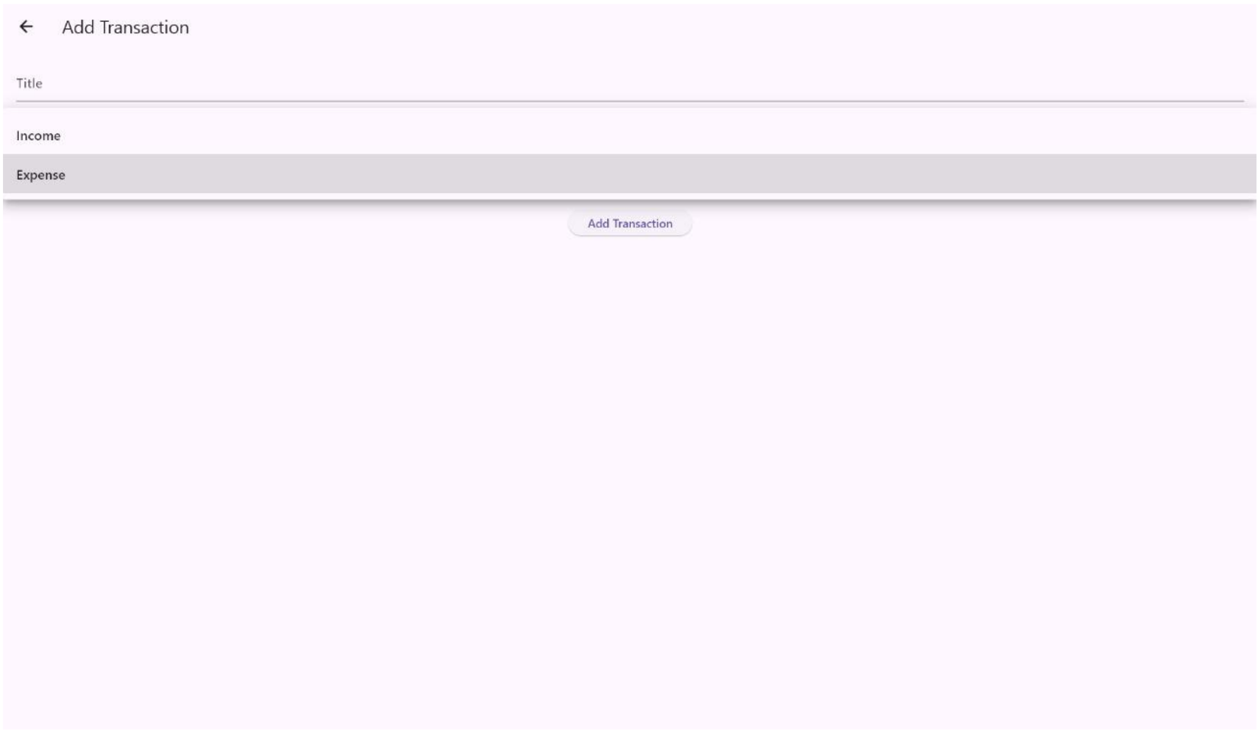
Add Transaction

Title

Amount

Expense

Add Transaction



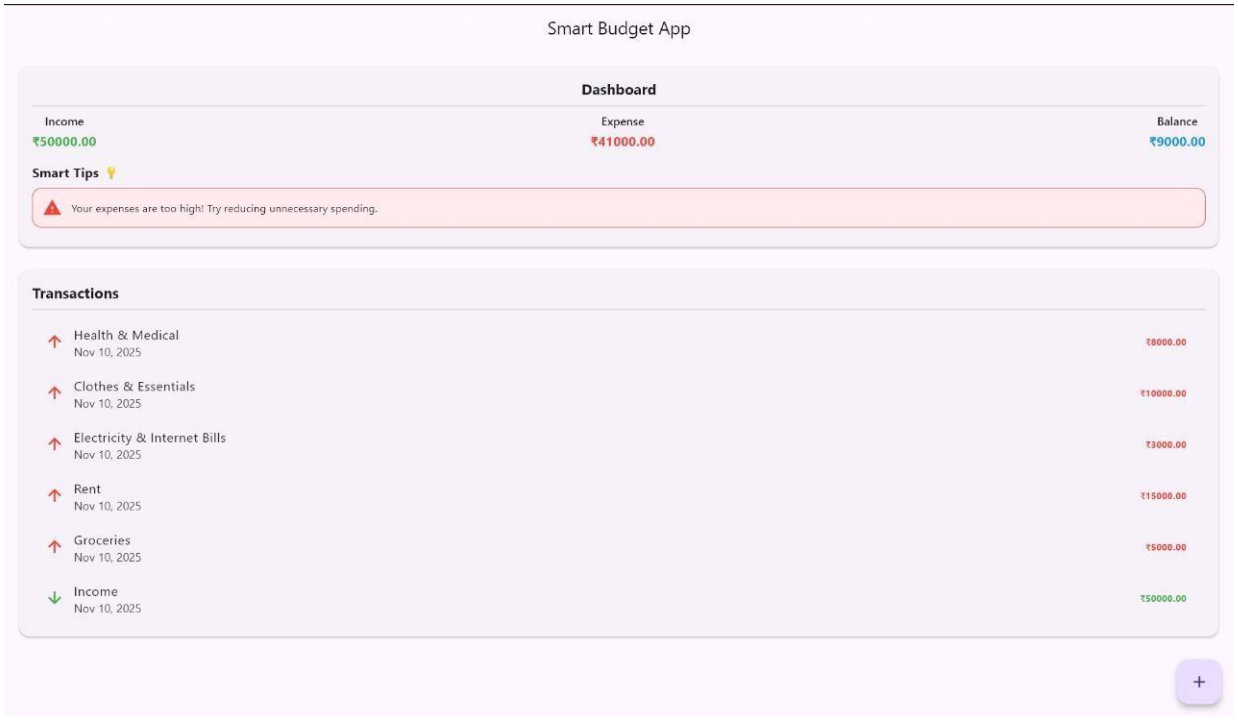
Add Transaction

Title

Income

Expense

Add Transaction



Smart Budget App

Dashboard

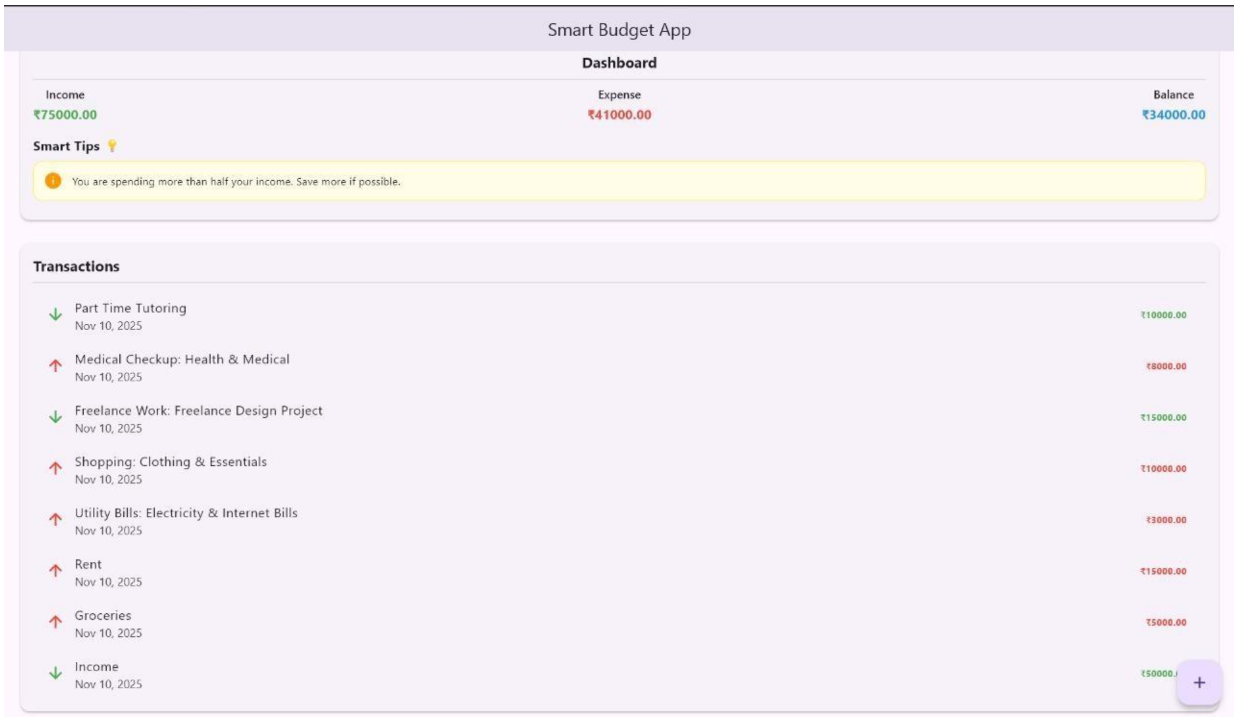
Income	Expense	Balance
₹50000.00	₹41000.00	₹9000.00

Smart Tips

Your expenses are too high! Try reducing unnecessary spending.

Transactions

Health & Medical	Nov 10, 2025	₹8000.00
Clothes & Essentials	Nov 10, 2025	₹10000.00
Electricity & Internet Bills	Nov 10, 2025	₹3000.00
Rent	Nov 10, 2025	₹15000.00
Groceries	Nov 10, 2025	₹5000.00
Income	Nov 10, 2025	₹50000.00



Smart Budget App

Dashboard

Income	Expense	Balance
₹75000.00	₹41000.00	₹34000.00

Smart Tips

You are spending more than half your income. Save more if possible.

Transactions

↓	Part Time Tutoring Nov 10, 2025	₹10000.00
↑	Medical Checkup: Health & Medical Nov 10, 2025	₹8000.00
↓	Freelance Work: Freelance Design Project Nov 10, 2025	₹15000.00
↑	Shopping: Clothing & Essentials Nov 10, 2025	₹10000.00
↑	Utility Bills: Electricity & Internet Bills Nov 10, 2025	₹3000.00
↑	Rent Nov 10, 2025	₹15000.00
↑	Groceries Nov 10, 2025	₹5000.00
↓	Income Nov 10, 2025	₹50000.

CONCLUSION

The Smart Budget App successfully achieves its goal of simplifying personal finance management through automation, intelligence, and a user-friendly interface. It provides users with an effective way to track their income and expenses, monitor their balance, and receive personalized financial insights — all within a single mobile platform.

By leveraging the Flutter framework and Dart programming language, the app ensures a responsive, cross-platform, and visually appealing experience. The implementation of Smart Tips introduces a unique, advisory layer that guides users based on their spending patterns, fostering greater financial discipline and awareness.

Unlike traditional budgeting tools that rely on manual inputs or internet connectivity, the Smart Budget App operates entirely offline, ensuring data privacy and accessibility at any time. Its dynamic dashboards, instant calculations, and visual cues provide users with a real-time overview of their financial health.

In essence, this project demonstrates how modern mobile technologies can be effectively utilized to enhance personal finance management. It bridges the gap between basic record-keeping and intelligent financial assistance, promoting responsible spending habits and informed decision-making. The system's modular and scalable architecture also sets a strong foundation for future development, ensuring long-term relevance and adaptability in the evolving digital finance ecosystem.

FUTURE SCOPE

The Smart Budget App has significant potential for future expansion and enhancement. While the current version provides an efficient offline budgeting solution, future iterations can incorporate advanced technologies and features to enhance user engagement and analytical depth.

Planned future enhancements include:

1. Cloud Synchronization: Integration with cloud platforms such as Firebase or Google Drive to enable data backup, multi-device synchronization, and secure access from anywhere.
2. Category-Based Analytics: Advanced visualization of expenses and income across categories (e.g., food, travel, bills) using dynamic charts and graphs for better financial insight.
3. AI-Powered Predictions: Implementation of machine learning models to predict future spending patterns, suggest optimized savings plans, and provide intelligent financial forecasts.
4. User Authentication and Profiles: Adding secure login features, allowing multiple users or family members to maintain individual budgets within the same app.
5. Voice and Chatbot Assistance: Integration of natural language interfaces or AI chatbots to assist users in adding transactions, checking balances, or receiving financial advice through voice commands.
6. Dark Mode and Theme Customization: Enhancing user experience through personalized interface options and accessibility features.
7. Integration with Banking APIs: Secure linking of the app with banking or UPI systems for automatic transaction imports and real-time expense updates.

Through these future advancements, the Smart Budget App can evolve into a comprehensive personal finance ecosystem, combining the power of artificial intelligence, cloud computing, and modern UI/UX design. Its adaptability ensures that it can grow alongside user needs and emerging financial technologies, making it not only a practical budgeting tool but also a platform for financial empowerment and digital literacy.

REFERENCES

1. Abbas, H. (2024, January 27). *30+ Free and Open-source Flutter Finance App Samples: Learn by Example*. Medevel. Retrieved November 10, 2025, from https://medevel.com/finance-apps-with-flutter/

2. GeeksforGeeks. (2025, July 23). *Expense Tracker Application Flutter*. Retrieved November 10, 2025, from https://www.geeksforgeeks.org/flutter/expense-tracker-application-flutter/

3. Project Gurukul. (2025). *Flutter Project – Expense Tracker*. Retrieved November 10, 2025, from https://projectgurukul.org/flutter-expense-tracker-project/

4. GitHub. (2025). *Expense Tracker Projects in Dart and Flutter*. Retrieved November 10, 2025, from https://github.com/topics/expense-tracker?l=dart

5. Amalraj, T., Karunamurthy, A., & Kiruthivasan, R. (2024). *Creating Robust Expense Tracker Applications with Flutter: A Step-by-Step Guide*. International Journal of Innovative Science and Research Technology, 9(4). Retrieved November 10, 2025, from https://www.researchgate.net/publication/380305246

6. Rajas, N., Patil, S., Patil, S. A., Patil, S. S., Patil, S., Patil, S., & Patil, S. (2023). *Money Management App for Expense Planning Based on Flutter*. International Journal for Research in Applied Science and Engineering Technology (IJRASET). Retrieved November 10, 2025, from https://www.ijraset.com/research-paper/money-management-app-for-expense-planning-based-on-flutter

7. Shi, W., Ali, M., & Leong, C. M. (2024). *Dynamics of Personal Financial Management: A Bibliometric and Systematic Review on Financial Literacy, Capability, and Behavior*. International Journal of Bank Marketing, 43(1), 125–165. Emerald Publishing. Retrieved November 10, 2025, from https://www.emerald.com/ijbm/article/43/1/125/1240847

8. Shailendra, N., Sanjana, M., Ravikumar, G. K., Gupta, S., & P., S. (2025). *Money Map: The Personal Finance Management System*. IOSR Journal of Computer Engineering, 27(2), 60–64. Retrieved November 10, 2025, from https://www.iosrjournals.org/iosr-jce/papers/Vol27-issue2/Ser-3/G2702036064.pdf

9. Singh, H., Chaturvedi, D. D., & Jain, A. (2022). *Personal Finance Management of Indian Working Professionals: An Empirical Study*. Webology, 15(1). Retrieved November 10, 2025, from https://webology.org/data-cms/articles/20220325060733pmwebology%2015%20%281%29%20-%202%20pdf.pdf

10. Forbes Advisor. (2025). *Are Budgeting Apps Worth It?* Retrieved November 10, 2025, from https://www.forbes.com/advisor/banking/are-budgeting-apps-worth-it/