Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report
on
"BudgetBuddy"

[Code No: ENGG 102]

(For partial fulfillment of 1st Year/ 2nd Semester in Computer Science)

Submitted By:

Utsaha Karki (Roll No: 45)
Ankit Shrestha (Roll No: 46)
Shreya Paudyal (Roll No: 47)
Aasish Humagain (Roll No: 48)

Submitted To:

Mr. Suman Shrestha

Department of Computer Science and Engineering

Submission Date:

Bona fide Certificate

This project work on

“BudgetBuddy”

is the Bona fide work of

Utsaha Karki (Roll No: 45)

Ankit Shrestha (Roll No: 46)

Shreya Paudyal (Roll No: 47)

Aasish Humagain (Roll No: 48)

who carried out the project work under my supervision

Project Supervisor

____

Ms. Praynita Karki
Assistant Professor
Department of Computer Science and Engineering
Kathmandu University
Date:

Acknowledgement

We would like to express our sincere gratitude to everyone who supported and guided us throughout the successful completion of our project, “BudgetBuddy”. Working together as a team with individuals of diverse ideas and perspectives has been a truly rewarding experience. Each team member’s contributions helped shape the direction and outcome of the project.

We extended our heartfelt thanks to our project supervisor, **Ms. Praynita Karki**, for her constant guidance, encouragement, and insightful feedback throughout the course of this work. Her expert advice and constructive criticism not only enhanced the quality of our project but also helped us gain a better understanding of system design, problem-solving, and the importance of collaborative learning.

We are also grateful to **Kathmandu University** and the **Department of Computer Science and Engineering (DoCSE)** for providing us with the opportunity and platform to work on real world projects. The learning environment and institutional support we received were vital in helping us apply classroom knowledge to practical development.

Lastly, we appreciate all forms of feedback and suggestions for improving our project in the future. This experience has been a great steppingstone in our academic and professional journey.

Regards,
Utsaha Karki (Roll No: 45)
Ankit Shrestha (Roll No: 46)
Shreya Paudyal (Roll No: 47)
Aasish Humagain (Roll No: 48)

Abstract

BudgetBuddy is a lightweight, user-friendly desktop application designed to assist individuals in managing their personal finances effectively. Developed using **C++**, **Qt**, and **SQLite**, the application enables users to track their income and expenses, set monthly savings goals, and visualize financial data through interactive reports and charts—all in a secure and offline environment.

The system features a comprehensive user authentication mechanism, including secure login, signup with password validation, password recovery, and profile management. Users can add categorized financial transactions, monitor goal progress, and receive visual insights that support smarter budgeting decisions.

BudgetBuddy stands out for its intuitive interface, real-time feedback, and focus on privacy, as all data is stored locally without reliance on third-party servers. The project demonstrates practical implementation of GUI design, database integration, and secure coding practices within a C++/Qt development stack.

This tool is ideal for students, freelancers, and anyone seeking a private, customizable budgeting solution without the need for internet connectivity.

Keywords: BudgetBuddy, C++, Qt, SQLite

Table of Content

List of Figures

Acronyms/Abbreviation

OOP	Object Oriented Programming
GUI	Graphical User Interface
AI	Artificial Intelligence
SQL	Structured Query Language
UI	User Interface
OS	Operating System

Chapter 1 Introduction

In the modern world, managing personal finances responsibly is essential but often overlooked. Many people struggle to track daily income and expenses, which can lead to poor spending habits and financial stress over time. Although there are many advanced budgeting apps available, they can be unnecessarily complex for users who just need a simple way to record and review their transactions.

BudgetBuddy was designed to fill this gap by offering a straightforward, console-based budget tracking application written in C++. It allows users to add income and expenses, categorize transactions, and view summaries of their financial activities. The program demonstrates how basic programming concepts like classes, objects, functions, and simple data handling can be combined to create a practical and usable tool.

The clean structure makes BudgetBuddy easy to expand with new features, such as saving data to files or building a graphical interface in the future. Overall, our project encourages users to develop better money habits while serving as a strong learning project for aspiring programmers.

1.1 Background

Managing personal finances has always been an essential part of daily life, but it has become increasingly challenging in today's modern world. Rising living costs, unplanned expenses, and the ease of digital payments often lead people to lose track of their spending. As a result, many individuals face difficulties in budgeting and saving effectively. While there are sophisticated financial management apps available, they often come with a steep learning curve or require internet access and frequent updates, which may not be practical for everyone.

To address this gap, many people look for simpler, offline tools that help them develop the habit of recording their income and expenses in a straightforward way.

BudgetBuddy was created with this dual purpose: to serve as a practical tool for users and as an educational project for students learning the fundamentals of C++. By developing our project in C++, the project demonstrates how object-oriented programming, file handling, and console-based input/output can be combined to create a functional budget tracking system. The focus is on clear code structure, modular design, and ease of use, ensuring that the program can be expanded in the future. Whether it's adding persistent storage, generating reports, or building a graphical interface, our project provides a solid starting point for learning how software can help solve everyday challenges like managing money wisely.

1.2 Objectives

The main objective of our project is to simplify the process of managing and keeping track of your finances more effectively through a simple interface accessible to all the users. The main objectives for this project were:

1. To provide users with a secure and easy platform to track their daily expenses and income.

2. To allow users to set monthly budget goals.

3. To help users gain insight into their financial habits by categorizing transactions.

4. To provide a simple, intuitive interface that makes financial tracking accessible to all users.

1.3 Motivation and Significance

Managing money can be tricky even for tech-savvy individuals. Many people struggle with poor budgeting habits simply because they don’t have the right tools or motivation. We noticed that while premium budgeting apps offer automation and bank syncing, they often make users compromise on privacy or simplicity. That’s where our motivation for BudgetBuddy came from.

We wanted to build a solution that brings budgeting back to the basics: understanding where your money goes, planning better, and staying disciplined, all without unnecessary distractions or dependencies on internet access.

As students, we also wanted to challenge ourselves with a project that combined logic, design, and user experience. Working on our project gave us an opportunity to experiment with real-world development tools, manage a full application life cycle, and most importantly, solve a problem that many people deal with every day.

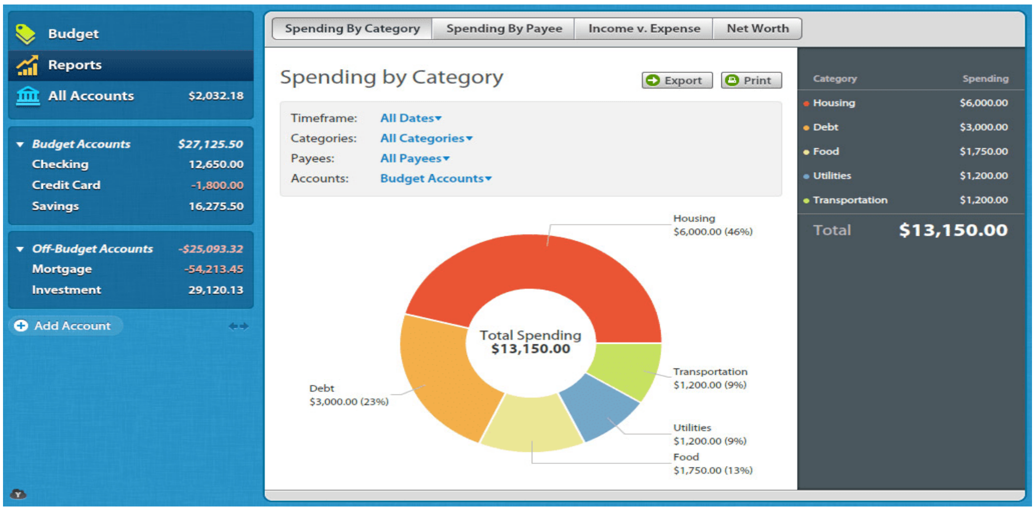
Chapter 2 Related Works

When we first started working on BudgetBuddy, we looked at some of the most popular budgeting apps out there. While these apps offer a lot of useful features like automatic bank syncing and mobile access, they often come with subscription fees and limited customization options. We realized that not everyone needs or wants an overly complex system tied to the internet.

2.1 Premium Budget tracking applications:

2.1.1 You Need a Budget (YNAB):

You Need a Budget (YNAB) is a well-known budgeting app that follows a strict zero-based budgeting philosophy, encouraging users to give every dollar a job. YNAB offers a structured financial approach that includes goal tracking, category-based budgeting, and real-time syncing across devices. With features like bank account integration, detailed financial reports, mobile access, and multi-device syncing, YNAB is ideal for users who are serious about long-term money management. However, it comes with a relatively high subscription cost and requires a bit of a learning curve for those unfamiliar with its budgeting method. While YNAB excels in automation and reporting, it lacks built-in investment tracking and offers limited customization compared to open-source alternatives.



All Accounts	$2,032.18
Budget Accounts	*$27,125.50*
Checking	12,650.00
Credit Card	-1,800.00
Savings	16,275.50
Off-Budget Accounts	*-$25,093.32*
Mortgage	-54,213.45
Investment	29,120.13

Spending by Category

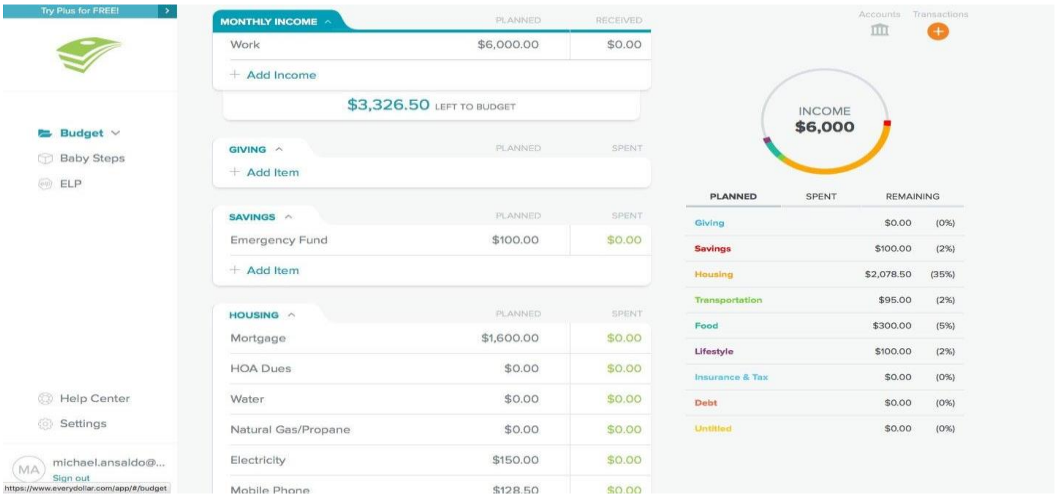
Timeframe: All Dates
Categories: All Categories
Payees: All Payees
Accounts: Budget Accounts

Category	Spending
Housing	$6,000.00
Debt	$3,000.00
Food	$1,750.00
Utilities	$1,200.00
Transportation	$1,200.00
Total	$13,150.00

Figure 1: YNAB

2.1.2 EveryDollar

EveryDollar, developed by Dave Ramsey’s team, is a simple, beginner-friendly budgeting app based on zero-based principles. The free version supports manual budgeting, while the Premium version (~$79.99/year) adds bank syncing and other features. It’s easy to use but limited in automation, reporting, and customization i.e the most advanced features are behind a paywall.



MONTHLY INCOME	PLANNED	RECEIVED
Work	$6,000.00	$0.00
+ Add Income		

$3,326.50 LEFT TO BUDGET

GIVING	PLANNED	SPENT
+ Add Item		

SAVINGS	PLANNED	SPENT
Emergency Fund	$100.00	$0.00
+ Add Item		

HOUSING	PLANNED	SPENT
Mortgage	$1,600.00	$0.00
HOA Dues	$0.00	$0.00
Water	$0.00	$0.00
Natural Gas/Propane	$0.00	$0.00
Electricity	$150.00	$0.00
Mobile Phone	$128.50	$0.00

PLANNED	SPENT	REMAINING
Giving	$0.00	(0%)
Savings	$100.00	(2%)
Housing	$2,078.50	(35%)
Transportation	$95.00	(2%)
Food	$300.00	(5%)
Lifestyle	$100.00	(2%)
Insurance & Tax	$0.00	(0%)
Debt	$0.00	(0%)
Untitled	$0.00	(0%)

Figure 2: EveryDollar

2.2 Comparison with Our Project

What makes BudgetBuddy different is its simplicity and control. We built it to be open-source and offline, giving users full authority over how they want to manage their finances. There’s no cloud syncing or automation by default, but for users who value privacy and want a tool they can customize, this is a perfect fit. Our project is best suited for tech-savvy individuals who are comfortable with manual input and possibly tweaking code to meet their needs. It offers flexibility and cost savings (it's free), but at the cost of ease of use and automation.

YNAB (You Need a Budget) is a premium, cloud-based app focused on zero-based budgeting. It offers robust features like real-time bank syncing, goal tracking, detailed reports, and mobile access. Designed to build long-term financial habits, YNAB is best for users seeking a highly structured, automated experience, though it comes with a high price and a bit of a learning curve. EveryDollar, created by Ramsey Solutions, also uses a zero-based approach but emphasizes simplicity. The free version supports manual budgeting, while the Premium plan adds bank syncing and automation. It’s great for beginners and fans of Dave Ramsey’s method, but it lacks advanced reporting and customization, with most key features behind a paywall.

While our UI might not be as polished as commercial versions due to limited experience in graphical design, our project focuses on functionality, structure, and learning value which are often missing in ready-made versions.

2.3 Summary

In summary, BudgetBuddy stands out for its customizability, privacy, and zero cost, appealing to users who prefer manual control and are comfortable with a DIY setup. YNAB offers the most robust feature set and automation for committed budgeters, while EveryDollar provides a user-friendly, simplified budgeting experience geared toward beginners and Ramsey followers. Your choice depends on whether you prioritize flexibility and control (BudgetBuddy), automation and structured methodology (YNAB), or ease of use and simplicity (EveryDollar).

Chapter 3 Design and Implementation

3.1 Problem Analysis

The core problem addressed by the BudgetBuddy application is the challenge that users face in effectively managing their personal finances, budget planning, and expense tracking in a simple, intuitive, and automated way. Key issues include:

1. Difficulty in monitoring daily expenses leading to overspending.
2. Lack of clear visualization and alerts to prevent budget overruns.
3. Manual and time-consuming tracking of transactions and bills.
4. Insufficient integration of savings goals and real-time tracking.
5. Need for easy authentication and secure management of user data.
6. Managing multiple features such as expense tracking, budgeting, bills, savings, and visualization in a coherent flow.

The application aims to solve these problems by providing a centralized platform where users can add and edit transactions, plan budgets, track bills, set savings goals, and view insightful reports and notifications. This addresses both behavioral and technical challenges of budgeting and financial management.

3.2 Algorithm Design

1. User-Authentication
   Validate user credentials and manage sign-up/login with secure password checks.

2. Expense Tracking
   Input transaction details → Validate → Store → Update expense history and totals.
3. Budget Monitoring
   Compare actual expenses against user-set budget limits → Calculate progress → Trigger alerts if limits are close or exceeded.
4. Saving Goals
   Track savings contributions → Calculate progress towards goal → Provide updates.
5. Bills Tracking
   Store bills with due dates → Schedule reminders → Mark bills as paid/updated.
6. Data Visualization
   Aggregate transaction data → Generate charts (pie, bar, trends) for user insights.

3.3 Procedure

1. Authentication: User signs up or logs in, system verifies identity and loads dashboard.

2. Dashboard Access: Displays summary of expenses, budgets, savings, bills.

3. Manage Expenses: User adds, edits, or deletes transactions and system updates records.

4. Set Budgets: User defines budget limits, system monitors spending and shows progress bars and then alerts on threshold breaches.

5. Savings Management: User sets goals and tracks progress through contributions.

6. Bills Management: User adds bills, system sends reminders and the bills are marked as paid when applicable.

7. View Reports: User accesses visual charts for spending and budget trends.

3.4 Testing and Debugging

Testing was performed continuously throughout development by adding the budget and your goal while overviewing if the overspending message alert pops out or not. Bugs and logical errors were fixed with assistance from our supervisor and online resources

3.5 Documentation

A log sheet was maintained to track development progress and issues encountered. The final project report has been compiled after completion.

3.6 Use Case Diagram

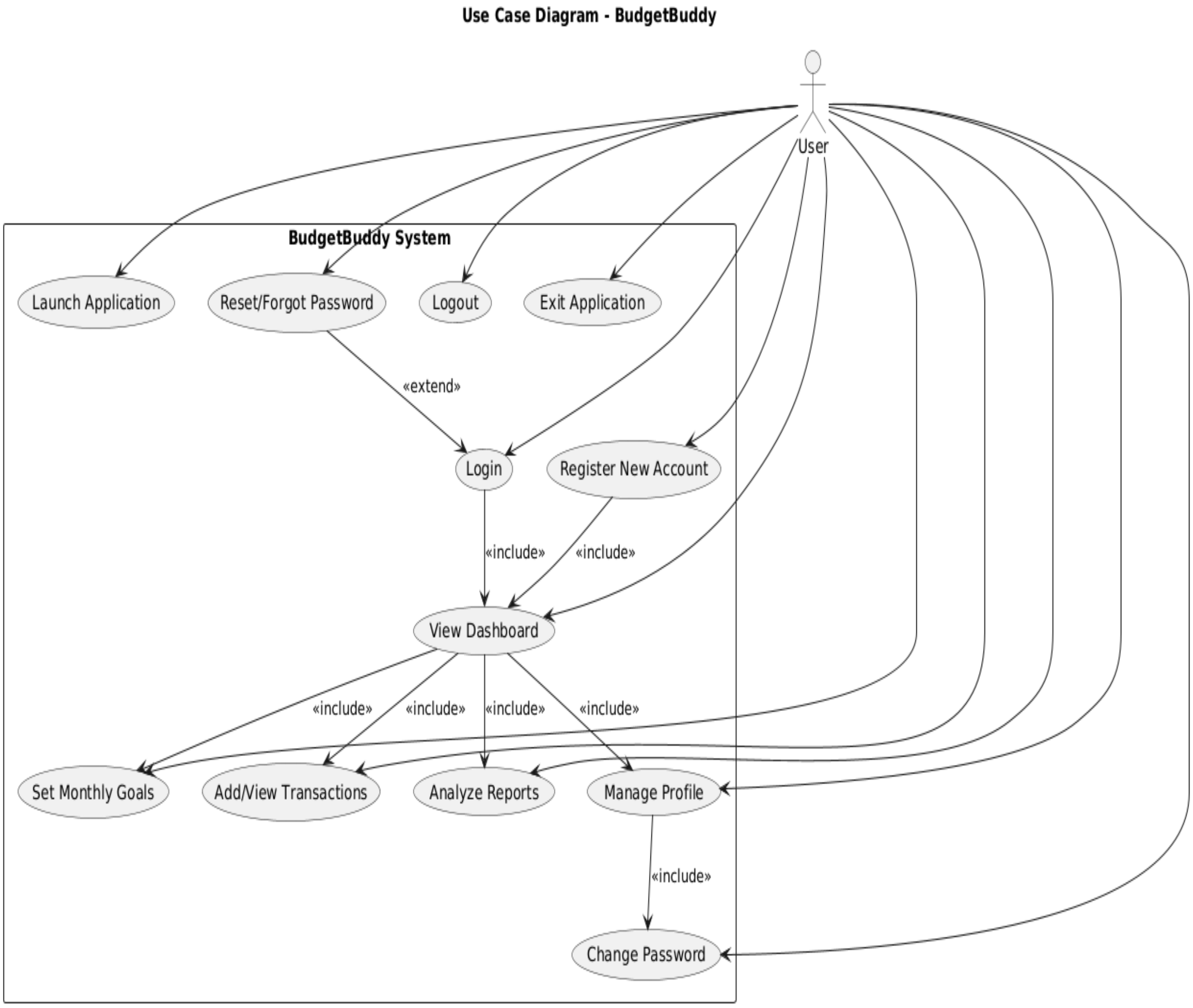


Figure 3: Use-Case Diagram

The BudgetBuddy application is designed to accommodate the **User** who are able to login and access their budget which is being tracked.

User is the person who interacts with the app to manage their budget, set goals, view reports, and manage transactions.

Use Case Descriptions

3.6.1 User Authentication

- Login: User enters credentials to securely access their account.
- Signup: New users can register using strong password criteria.
- Forgot Password: User resets password by verifying username.
- Change Password: Accessible via profile settings to maintain account security.

3.6.2 Dashboard

- Serves as a central hub after login.
- Allows navigation to transactions, reports, goals, profile, and logout.

3.6.3 Transactions Manager

- Add Income/Expense – Users can input transactions with attributes like date, type, category, amount, and remarks.
- View All – Shows a unified list of transactions and associated goals.
- Filter/Sort – Filter by type, date range, or category for easy tracking.

3.6.4 Monthly Goals

- Set Limits – Users define a monthly spending goal per category.
- Track in Real-Time – When entering an expense, the app checks progress toward the goal.
- Visual Indicators – Real-time alerts appear when nearing or exceeding a goal.

3.6.5 Financial Reports

- Summary View – Users can see total income, total expenses, and remaining budget.
- Pie Charts – Qt Charts are used to generate visuals showing spending by category.
- Overspending Alerts – If expenses exceed the set monthly goal, a red alert notifies the user.

3.6.6 Profile Management

- View/Edit Profile – Manage basic user information.
- Change Password – Optional security updates for account.
- Logout – End the session securely.

3.7 ER (Entity Relationship) Diagram

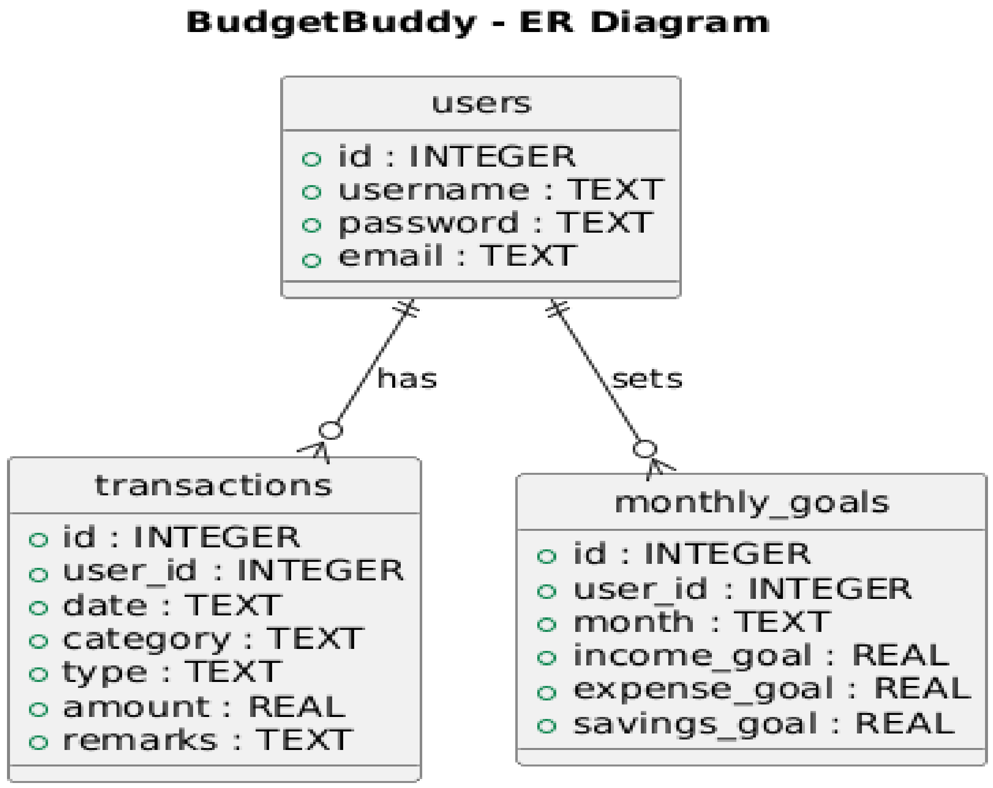


Figure 4: ER Diagram

3.8 Class Diagram

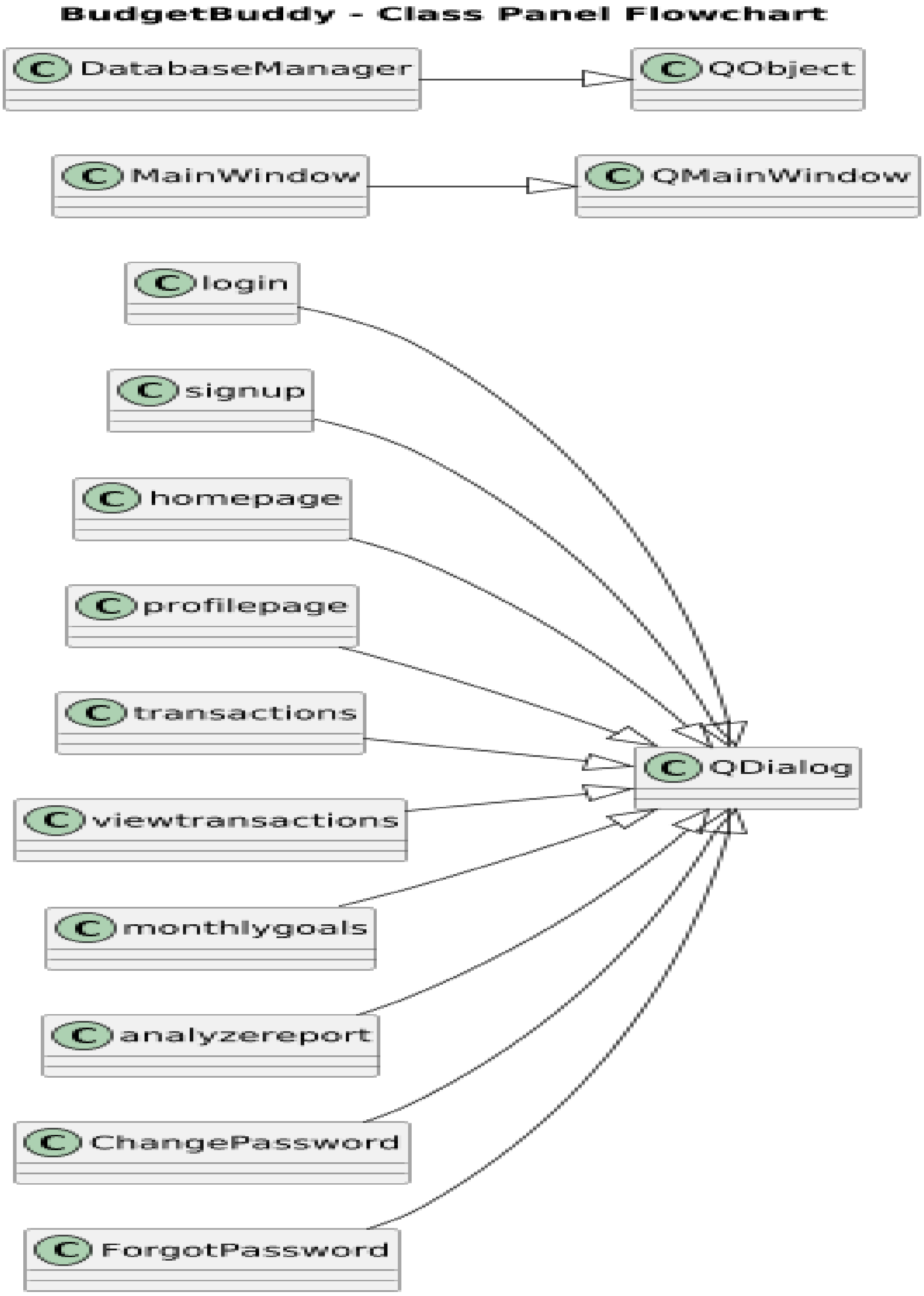


Figure 5: Class Diagram

3.9 System Requirement Specifications

This section outlines the system requirements for the development and smooth functioning of the BudgetBuddy, budget tracking application. It includes both software and hardware specifications, along with functional and non-functional software requirements.

3.9.1 Software Specifications

BudgetBuddy is a desktop-based application built using C++, Qt, and SQLite. The system is designed to run on Windows platforms and can be recompiled for other operating systems such as Linux and macOS with proper configuration. The system uses object-oriented programming principles and offers a graphical user interface for better usability.

Functional Requirements

1. The system should allow users to login and view there budget history.
2. It should display the history of both income and expenses on a monthly basis.
3. Users should be able to add monthly goals as per their need.
4. Users should be able to edit their profile as well as password and logout at any time.
5. The system should store and manage all data locally using a database.

Non-Functional Requirements

1. The system must run without internet connectivity (offline support).
2. It should have a responsive and user-friendly graphical interface.
3. The database operations must be fast and consistent.
4. The application should be stable and avoid crashes during typical usage.

5. System should handle invalid inputs gracefully and show proper validation messages.

3.9.2 Front-End Tools

i. Qt Creator

Qt Creator is a cross-platform integrated development environment (IDE) used to design the GUI of the application. It provides a rich set of built-in widgets, signal-slot mechanism, and real-time preview tools to make desktop applications interactive and visually appealing. The GUI in BudgetBuddy was designed using Qt Designer, allowing for fast drag-and-drop interface design.

3.9.3 Back-End Tools

i. SQLite

SQLite is a lightweight, file-based relational database engine used to store ticket data, bus information, routes, and schedules. It operates without a server, requires no setup or installation, and stores all information in a single .db file. Its simplicity, performance, and offline capabilities make it an ideal choice for Mero BudgetBuddy.

3.9.4 Hardware Specifications

As BudgetBuddy is a desktop-based application with moderate resource usage, no high-end hardware is required. Any general-purpose computer system that meets the following specifications is sufficient to run the application smoothly:

1. **RAM**: 4 GB (minimum), 8 GB recommended
2. **Operating System**: Windows 10 or higher (compatible with Linux/macOS with recompilation)
3. **Storage**: At least 100 MB of free disk space
4. **Display**: Minimum 1024x768 resolution
5. **Processor**: Intel Core i3 or equivalent (minimum)

4. Discussion and Achievements

The BudgetBuddy project addresses a crucial need for personal financial management by offering an accessible, easy-to-use platform that enables users to track expenses, set budgets, and accomplish savings goals. One significant achievement is the integration of core budgeting features into a unified application that simplifies financial tracking for users, particularly targeting young adults and students who face challenges managing their finances effectively.
Key accomplishments include:

1. Easy-to-use interface: Designed for clarity and simplicity.
2. All-in-one platform: From transactions to savings and reports, everything is integrated.
3. Strong security: Users can trust that their data stays private.
4. Positive feedback: Early testers found it helpful for managing their day-to-day spending.
5. Growth as developers: We learned to plan better, code cleaner, and debug faster.

This project substantiates the notion that a thoughtfully designed app can make financial literacy and budgeting accessible while maintaining motivational elements that encourage user engagement and long-term use.

4.1 Limitations and Deviations

Despite notable progress, several limitations and deviations from initial project goals were encountered:

1. Time limits meant we couldn’t implement everything we had hoped.
2. Some features are unfinished, like full gamification and advanced AI insights.

3. Technical hurdles—integrating OCR and syncing data in real-time was tougher than expected.
4. Design coordination—translating our UI concepts into working code took time.
5. Currently single-currency, limiting its use for international users.
6. Limited testing due to time and resource constraints.

These limitations provide valuable insights for refining project scope and development strategy in future iterations

4.2 Future Improvements

To enhance BudgetBuddy's effectiveness and user satisfaction, planned improvements include:

1. Add smarter AI features to predict spending patterns.
2. Sync with bank accounts and support multiple user profiles.
3. Add more gamification (rewards, milestones).
4. Support more currencies and regions.
5. Improve design based on user feedback.
6. Tighten security and data encryption.

Focusing on these areas will help transition BudgetBuddy from a solid prototype to a polished, market-ready product with broad appeal and lasting value.

5. Conclusion

BudgetBuddy began as a modest academic assignment but grew into a meaningful exploration of how software can solve real-life problems. In a world filled with overwhelming financial tools, we wanted to create something grounded, useful, and easy to approach, especially for young adults and students who are just beginning to manage their own finances. The final result is a desktop application that lets users track their income and expenses, set realistic savings goals, stay within budget, and gain clarity through visual reports. What makes BudgetBuddy stand out is its simplicity and privacy-focused design which is completely offline, fully in the user’s control, and free from the clutter of unnecessary features or subscriptions.

Beyond the application itself, this project has been a major learning experience for us as developers and collaborators. It challenged us to think through user needs, troubleshoot complex logic, design intuitive interfaces, and manage time and responsibilities as a team. We came away with a deeper appreciation for clean code, secure practices, thoughtful UI/UX, and the importance of adaptability in software projects. While we acknowledge that there are still features left to be implemented such as automation, cross-platform compatibility, and more advanced insights, the strong foundation we've built gives us the confidence and motivation to continue improving BudgetBuddy. It’s more than just a budgeting tool; it’s a reflection of our growth, our teamwork, and our ability to turn abstract ideas into something practical and impactful.

Reference

The Qt Company. (n.d.). *Qt 5.15 documentation*. Retrieved July 6, 2025, from https://doc.qt.io/qt-5/

TutorialsPoint. (n.d.). *C++ GUI programming with Qt*. Retrieved July 6, 2025, from https://www.tutorialspoint.com/qt/index.htm

cplusplus.com. (n.d.). *C++ language tutorial*. Retrieved July 6, 2025, from https://cplusplus.com/doc/tutorial/

QT C++ GUI Tutorial. (n.d.). *QT C++ GUI Tutorial For Beginners* [YouTube channel]. Retrieved July 6, 2025, from https://www.youtube.com/playlist?list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA

ProgrammingKnowledge. (n.d.). *ProgrammingKnowledge* [YouTube channel]. Retrieved July 6, 2025, from https://www.youtube.com/@ProgrammingKnowledge

Adam Finer. (2020, November 11). *Learn basic SQL in 15 minutes | Business Intelligence for Beginners | SQL tutorial for beginners 1/3* [Video]. YouTube. https://www.youtube.com/watch?v=kbKty5ZVKMY

Hezretov, M. (2018). *Budget Tracker: Highly customizable budgeting mobile application* (Master's dissertation, University of Colombo School of Computing). https://dl.ucsc.cmb.ac.lk/jspui/bitstream/123456789/4424/1/16550949.pdf

Maury, P., Ghatkar, P., Rane, M., & Bhosale, S. (2024). Case study on budget tracker app. *International Journal of Research Publication and Reviews*, 5(2), 3539–3544. https://ijrpr.com/uploads/V5ISSUE2/IJRPR22995.pdf

Ramsey Solutions. (n.d.). *EveryDollar: Budgeting made easy*. Ramsey Solutions. https://www.ramseysolutions.com/money/everydollar

You Need A Budget. (n.d.). *You Need A Budget (YNAB)*. https://www.ynab.com

Appendices

- User Authentication

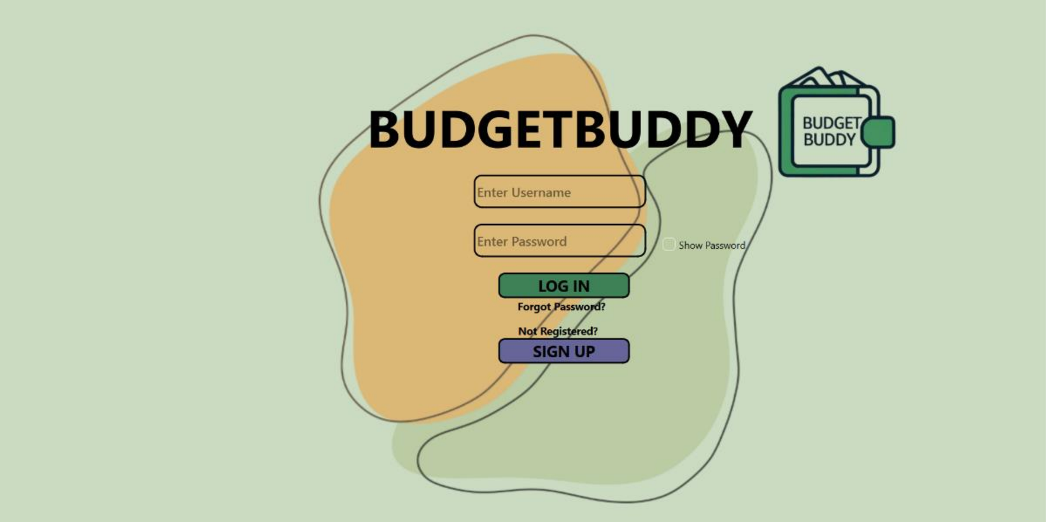


Figure 6: User Authentication

- Create an Account

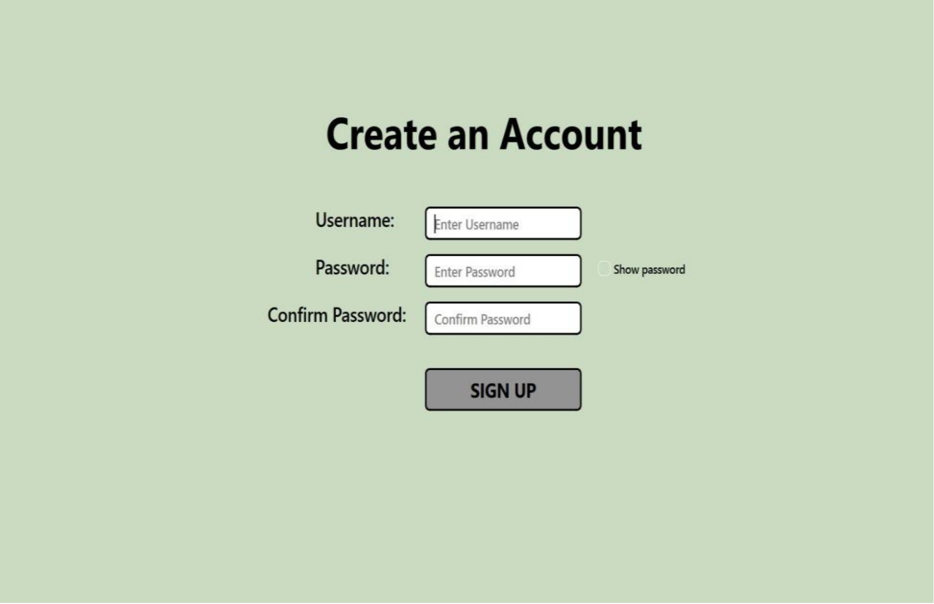
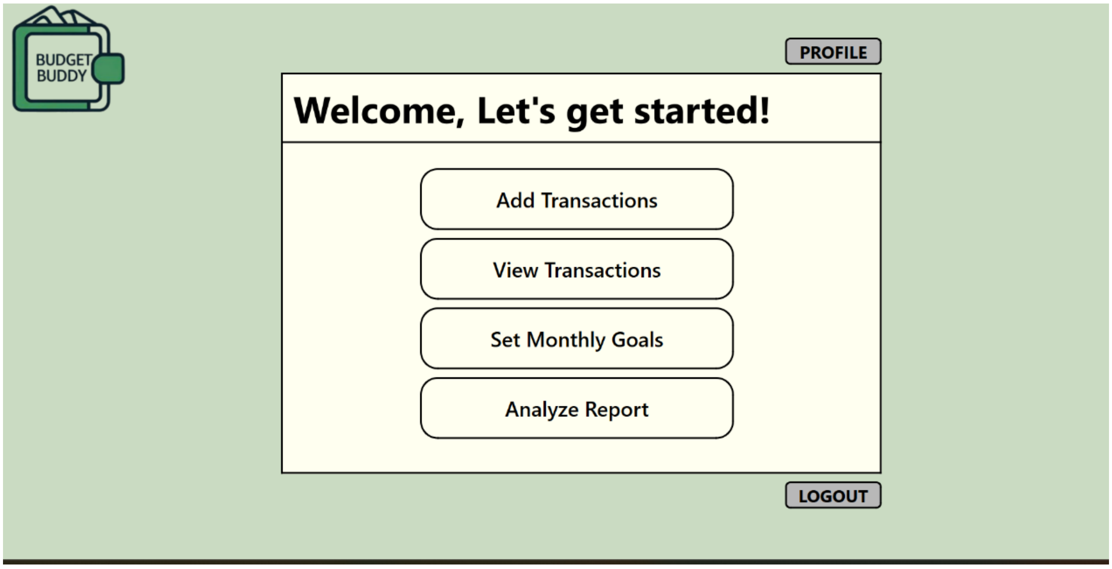


Figure 7: Create Account page

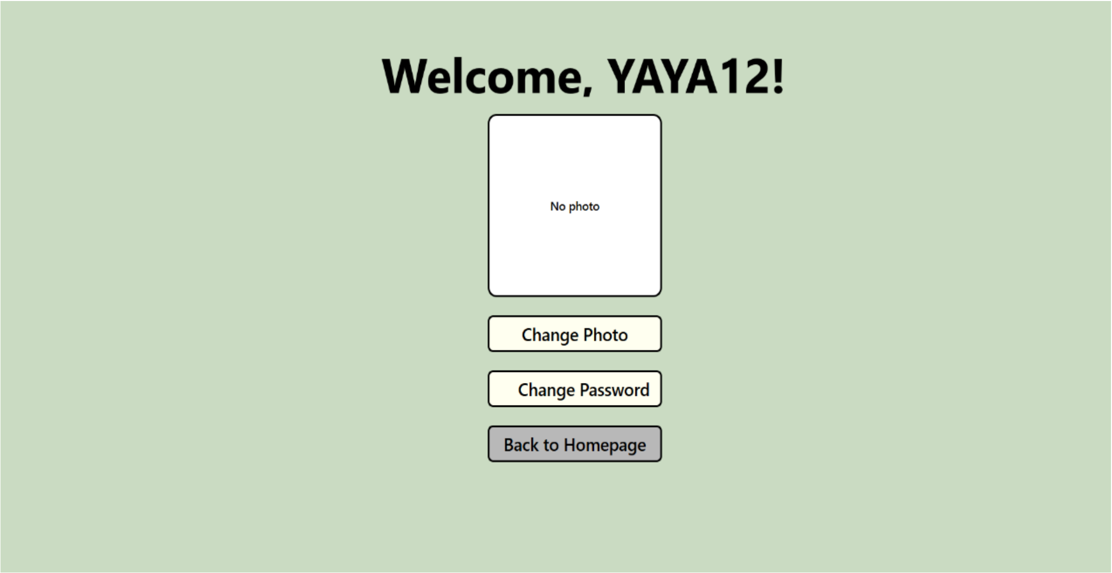
- Dashboard



Welcome, Let's get started!

Figure 8: Dashboard

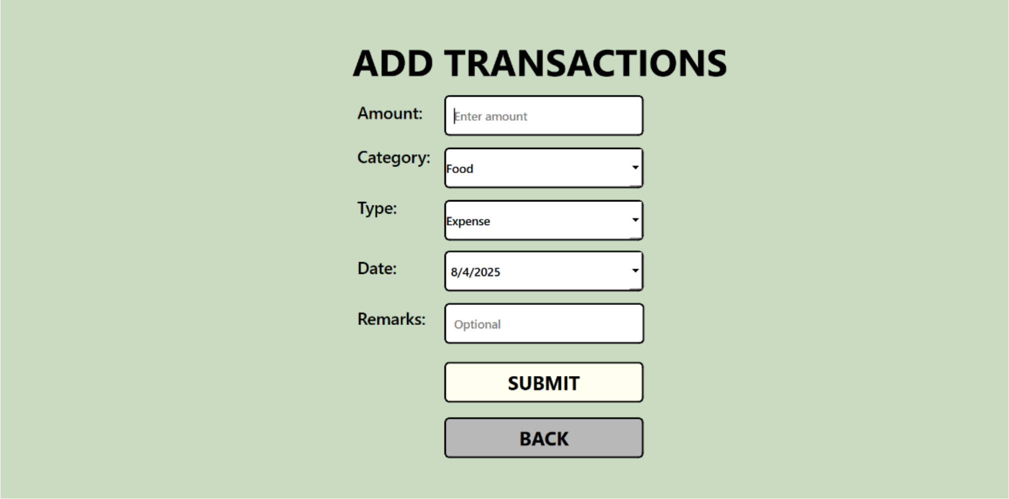
- Profile page



Welcome, YAYA12!

Figure 9: Profile page

- Set Transaction Page



ADD TRANSACTIONS

Amount: Enter amount

Category: Food

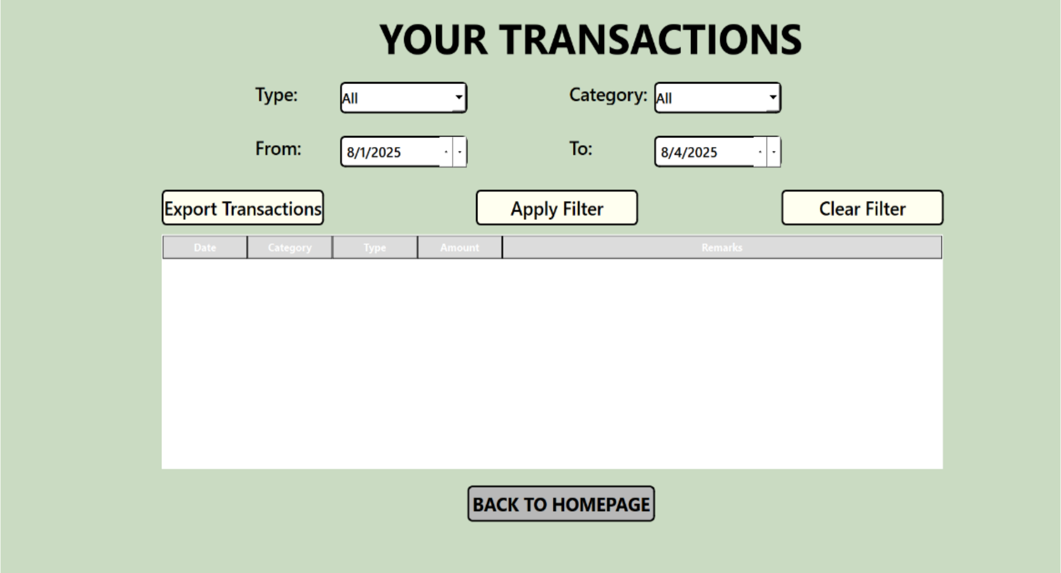
Type: Expense

Date: 8/4/2025

Remarks: Optional

Figure 10: Set Transaction

- View Transactions



YOUR TRANSACTIONS

Type: All

Category: All

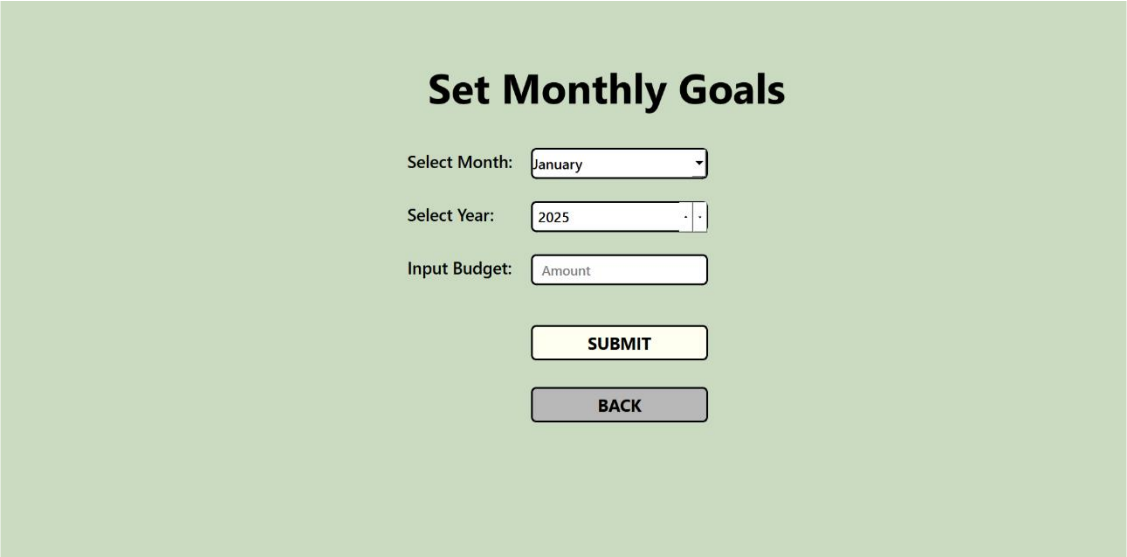
From: 8/1/2025

To: 8/4/2025

Date	Category	Type	Amount	Remarks

Figure 11: View Transaction

- Set Monthly goals



Set Monthly Goals

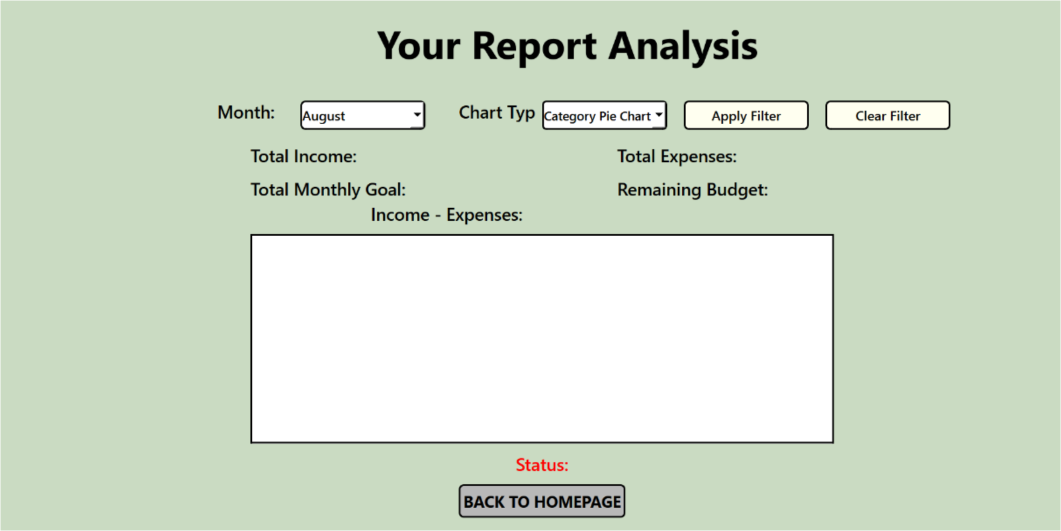
Select Month: January

Select Year: 2025

Input Budget: Amount

Figure 12: Set Monthly goals

- Report Analysis Page



Your Report Analysis

Month: August

Chart Typ Category Pie Chart

Total Income:

Total Expenses:

Total Monthly Goal:

Remaining Budget:

Income - Expenses:

Status:

Figure 13: Report Analysis