



**Module Code & Module Title**

**CS6004NI – Application Development**

**Assessment Type**

**Semester**

**2024/25 Autumn**

**Student Name: Aishworya Shrivastav**

**London Met ID: 21049631**

**College ID: NP01CP4A210128**

**Assignment Due Date: Friday, January 17, 2025**

**Assignment Submission Date: Friday, January 17, 2025**

**Submitted To: Abhishek Khadka**

**Word Count (Where required):1463**

*I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

**Abstract**

The project involves developing a prototype desktop application designed for personal expense tracking. The application allows the user to provide a username, application password, and preferred currency type. The application contains cash inflows like (credit, gain, or budget), also cash outflows for (debit, spending, or expenses), and debts. For cash outflows, a sufficient balance is required. The debt tracking feature should include fields such as the source of debt, the due date, and must ensure debts are cleared only from cash inflows.

Also pending debts are highlighted. Each transaction can include optional notes and is categorized with primary labels such as credit, debit, or debt. Users should also be able to tag transactions with custom labels and can search and filter records by the title, transaction type, tags, date range and data. Additionally, it also has a dashboard which shows clear information about total cash inflows, outflows, debts, cleared debts , remaining debts showing the top 5 highest/lowest transaction. Pending debts also contain an option about filter.

Table of Contents

[1. Project Overview 1](#_Toc188473479)

[2. Features and Functionality 2](#_Toc188473480)

[3. Proof of Work 4](#_Toc188473481)

[3.1 Visual guides: 5](#_Toc188473482)

[3.2 Designs as basis on code 8](#_Toc188473483)

[4. Individual Reflections 19](#_Toc188473484)

[5. Conclusion 20](#_Toc188473485)

[Figure 1 ERD 4](#_Toc188473493)

[Figure 2 Login UI 5](#_Toc188473494)

[Figure 3 Sign-Up UI 5](#_Toc188473495)

[Figure 4 Dashboard UI 6](#_Toc188473496)

[Figure 5 Transaction UI 6](#_Toc188473497)

[Figure 6 Pending Debts UI 7](#_Toc188473498)

[Figure 7 Folder Structure 8](#_Toc188473499)

[Figure 8 Folder Structure from actual 8](#_Toc188473500)

[Figure 9 Home Screen 9](#_Toc188473501)

[Figure 10 Register 9](#_Toc188473502)

[Figure 11 Confirmation box 10](#_Toc188473503)

[Figure 12 When user exists 10](#_Toc188473504)

[Figure 13 Success message 10](#_Toc188473505)

[Figure 14 Logged in with hashed password 10](#_Toc188473506)

[Figure 15 Redirected to Dashboard 11](#_Toc188473507)

[Figure 16 Home Page 11](#_Toc188473508)

[Figure 17 Dashboard 11](#_Toc188473509)

[Figure 18 Pending Debts 12](#_Toc188473510)

[Figure 19 Cleared Debts 12](#_Toc188473511)

[Figure 20 Transaction Table 12](#_Toc188473512)

[Figure 21 Adding Cashflow 13](#_Toc188473513)

[Figure 22 Success message 13](#_Toc188473514)

[Figure 23 Cashout flow 14](#_Toc188473515)

[Figure 24 Success after cashoutflow 15](#_Toc188473516)

[Figure 25 Adding Debts 16](#_Toc188473517)

[Figure 26 Filling in AddDebt 16](#_Toc188473518)

[Figure 27 Success popup 17](#_Toc188473519)

[Figure 28 Filter and Sort 17](#_Toc188473520)

[Figure 29 Filter and Sort in Transcation 17](#_Toc188473521)

[Figure 30 Changes in Dashboard 17](#_Toc188473522)

[Figure 31 Clear Debts 18](#_Toc188473523)

[Figure 32 After Clearing Debts 18](#_Toc188473524)

[Figure 33 Marked changes 18](#_Toc188473525)

# Project Overview

The app “Money Buddy” is a project in Net MAUI Blazor Hybrid Web App for helping users of our project to manage their finances. It helps user to track the total income, expense and even the remaining budget for the users to identify the need of saving. It also carries a user defined tag or a custom tag that can easily recognise the particular kind of transaction. It contains a detailed dashboard which contains a detailed statistics that can inform the user to take preventive measures.

The project was made in Net MAUI Blazor Hybrid Web App and using the bootstrap and CSS file to create the various design(UI) of the system and uses JSON and CSV as the need to store data.

With the combined approach of filter, sort and exporting these data from the app prompts the user for timely flexible and easiness of the product.

# Features and Functionality

1. User login and registration

* Allowing user to login and register with the hashed password to create robust security.
* A drop-down menu to use currency throughout the system.

1. A home page to welcome the first-time users or the people who frequently login to the app.
2. Dashboard feature:

* Immediately after success login, redirected with dashboard.
* Total inflow, outflow and remaining total balance is shown
* Filter and sort can be done within dashboard.
* A table of pending and cleared debts can be clearly shown.
* There exists a total transaction table to identify all the transaction done by the user.

1. Transaction management

* User can add, delete or edit the transactions.
* User can use default or customised tag to process the transaction.
* An option of sort and filter is also present here in the system.

1. Debt tracking

* Helps to show debts and track the status of debt i.e., pending or cleared.
* All the fields are required to process the debt file.

1. Data handling

The file is all imported in the form of CSV and JSON files for easy access and usability.

1. Cash inflow

* A clearly stated fields are required before adding the transaction as our income.

1. Cash outflow

* A clear stated fields are here again required to perform transaction.

# Proof of Work

The following provides act as a proof of work from the project.

* The project follows MVVM approach where(Model, View, View Model) as the architectural pattern.
* Model classes contain the various class which was made using ERD also handling the data related with business logic.
* It handles storage and data validation.

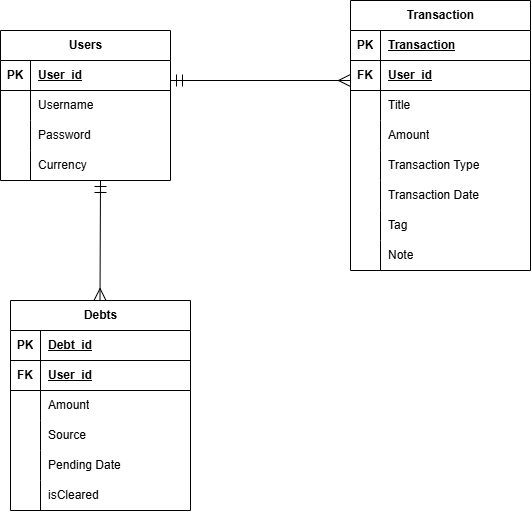


Figure 1 ERD

* Views contains the User Interface.
* Properties are binded with View Model.
* Usually razor files belong in this category.

View models contain a bridge between Model and Views.

It usually contains presentation logic and UI.

With the usage of data binding, it enhances interaction.

## Visual guides:

A wireframe of the designed file made when Milestone-1 and 2 were created and

A design after the submission of code is attached herewith.

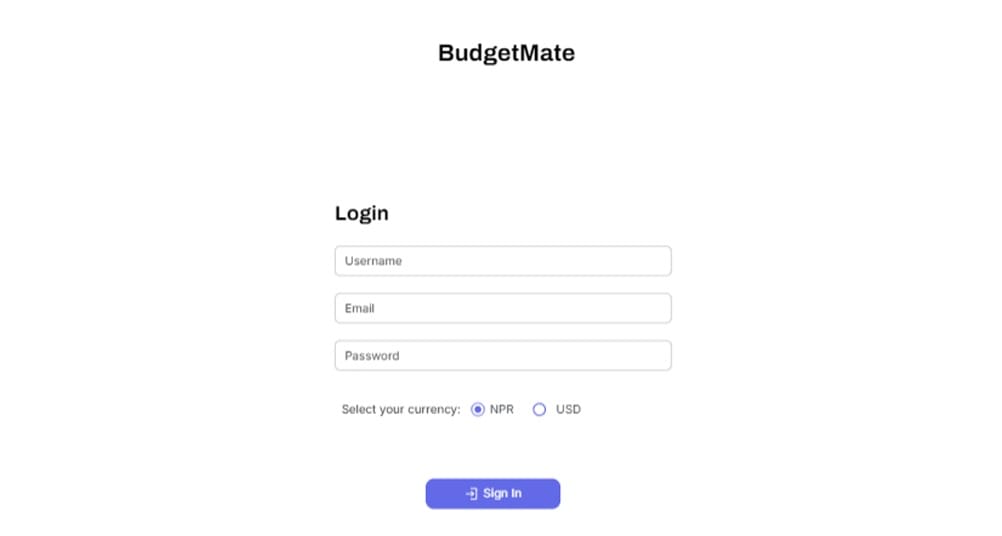


Figure 2 Login UI

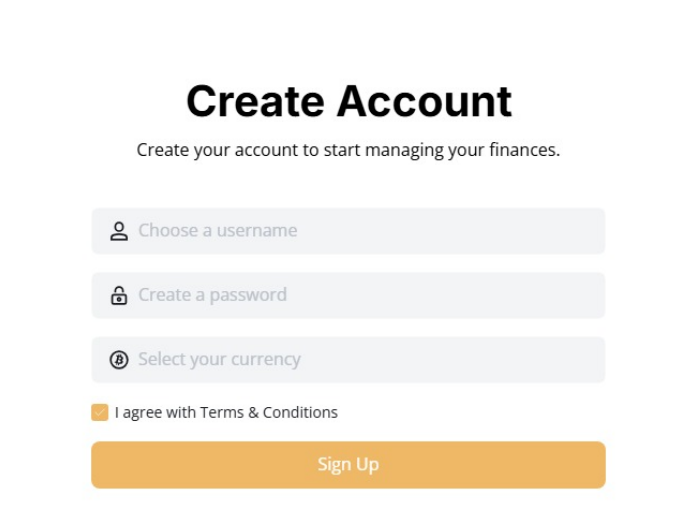


Figure 3 Sign-Up UI

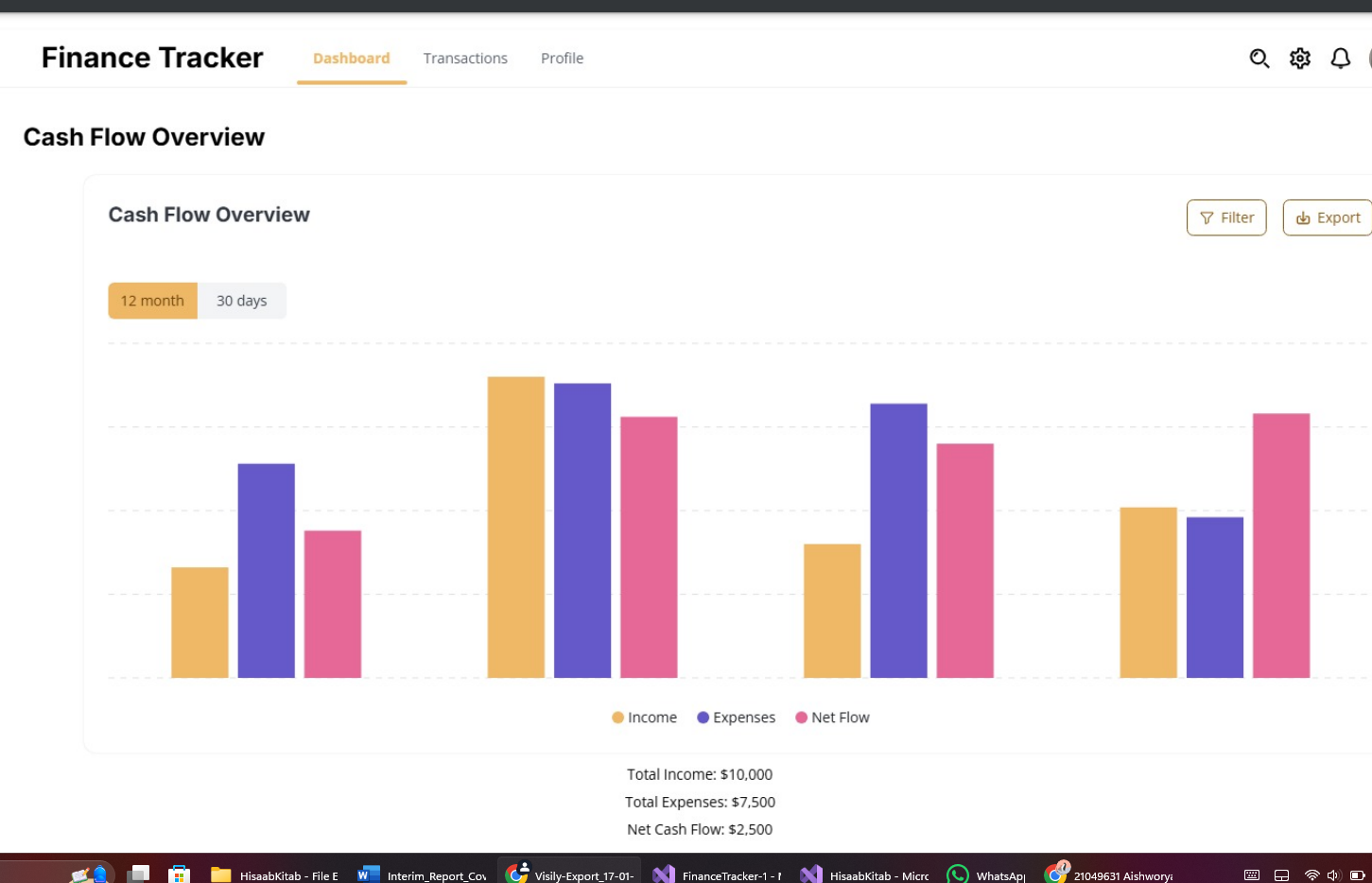


Figure 4 Dashboard UI

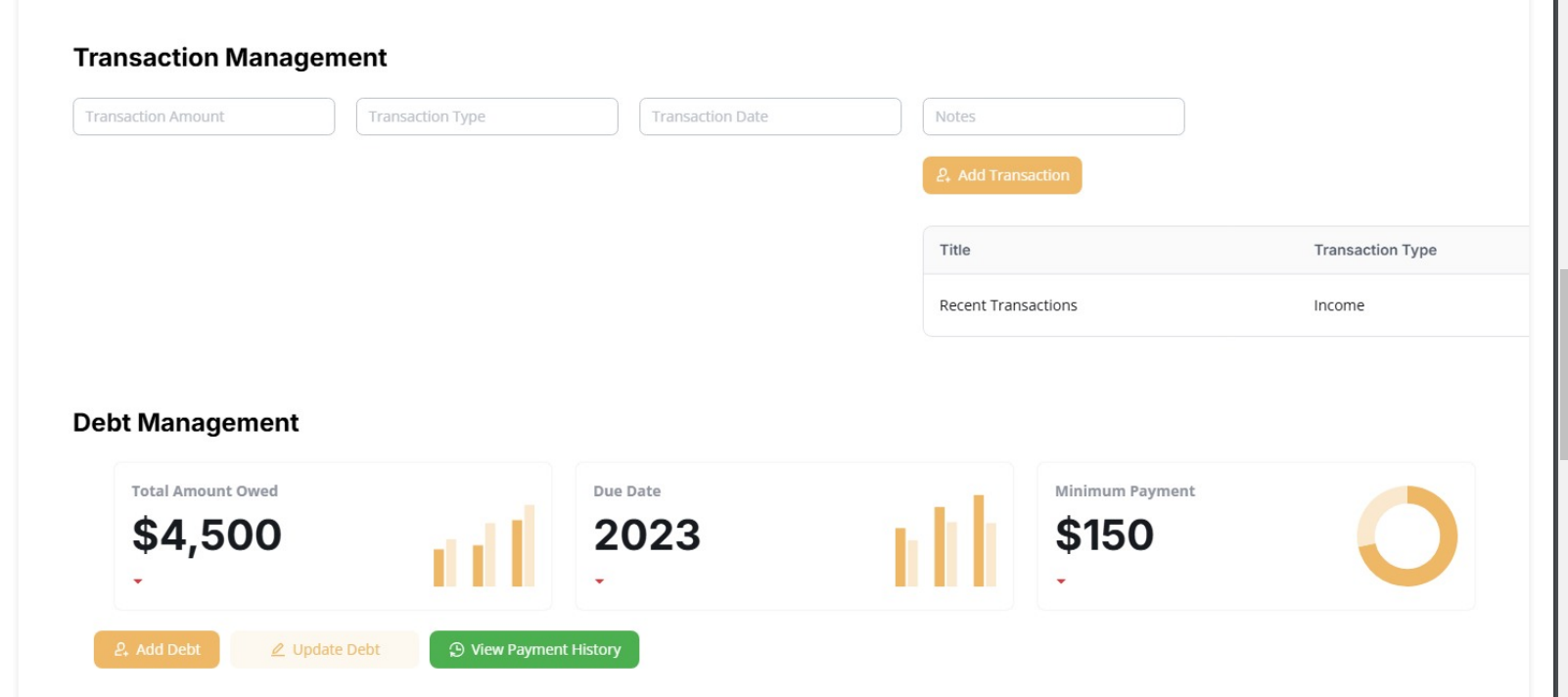


Figure 5 Transaction UI

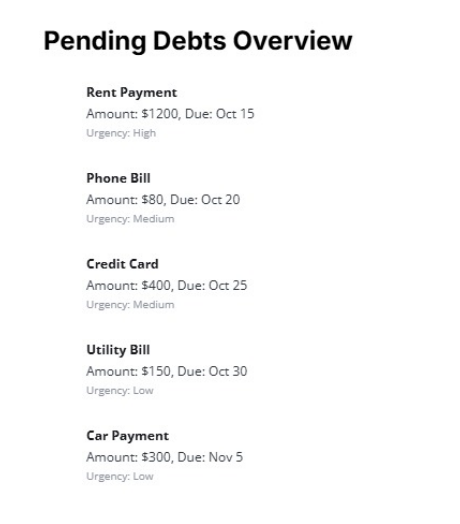


Figure 6 Pending Debts UI

## 3.2 Designs as basis on code

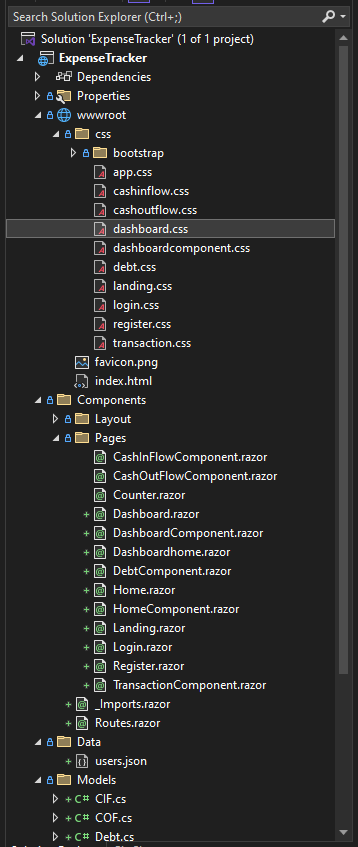


Figure 7 Folder Structure

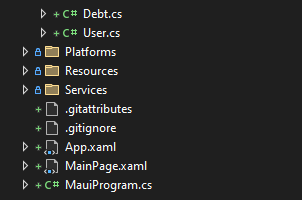


Figure 8 Folder Structure from actual

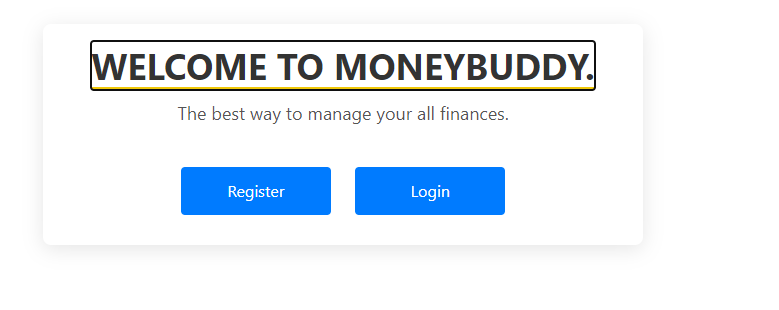


Figure 9 Home Screen

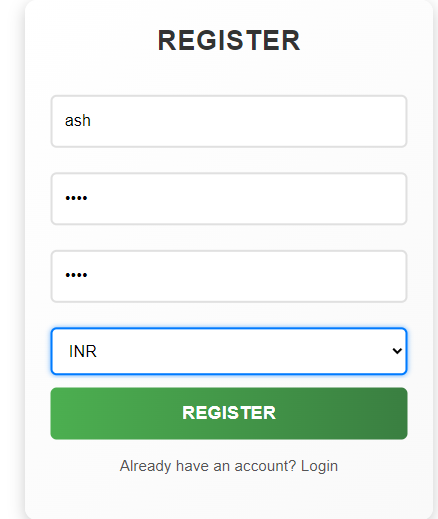


Figure 10 Register

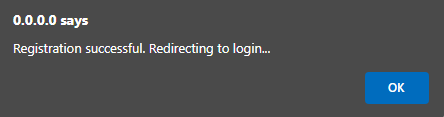


Figure 11 Confirmation box

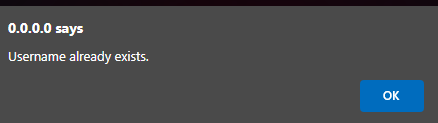


Figure 12 When user exists

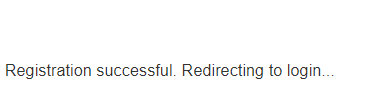


Figure 13 Success message

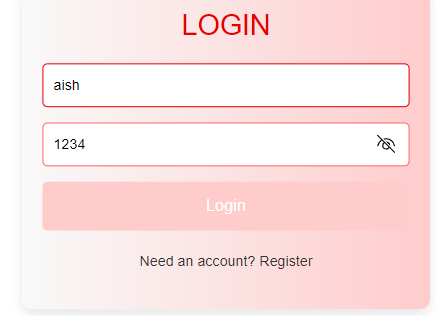


Figure 14 Logged in with hashed password

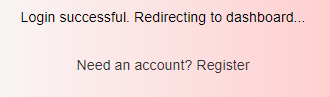


Figure 15 Redirected to Dashboard

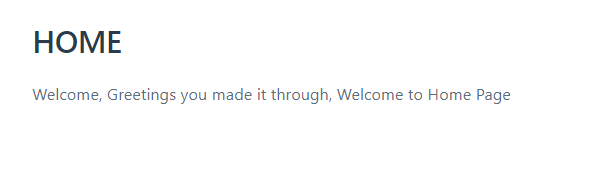


Figure 16 Home Page

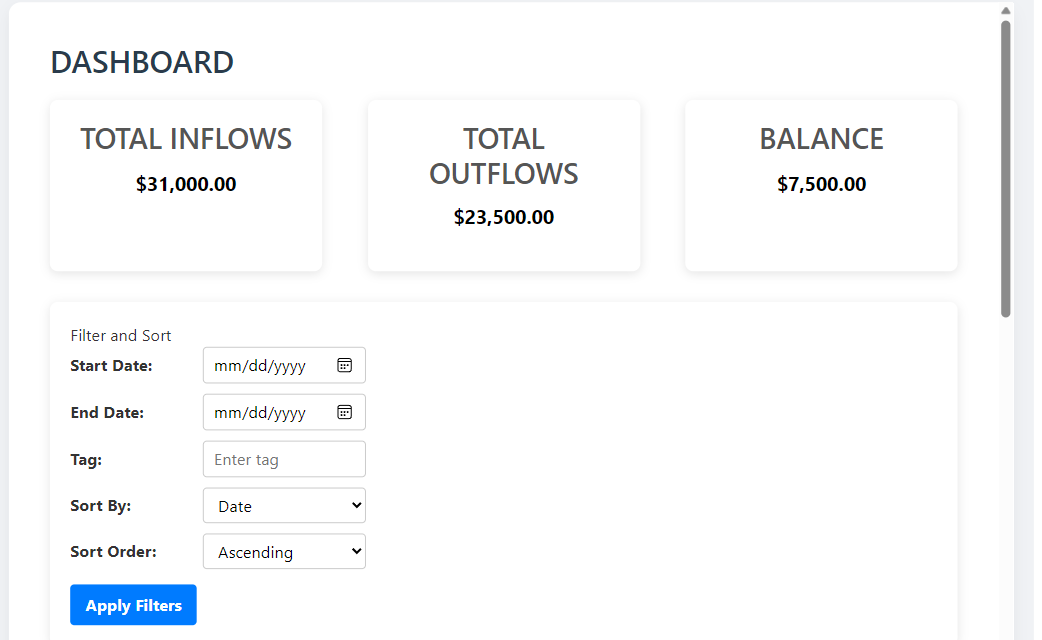


Figure 17 Dashboard

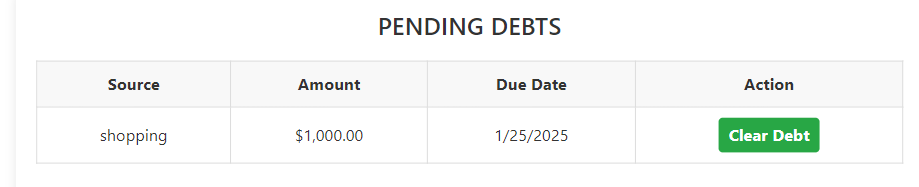


Figure 18 Pending Debts

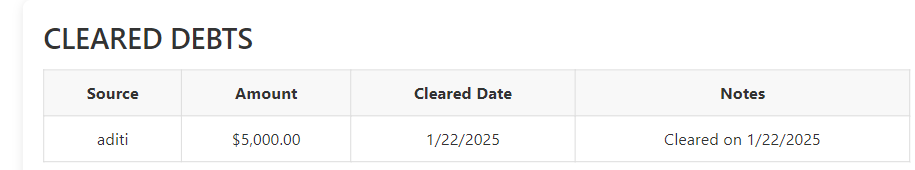


Figure 19 Cleared Debts

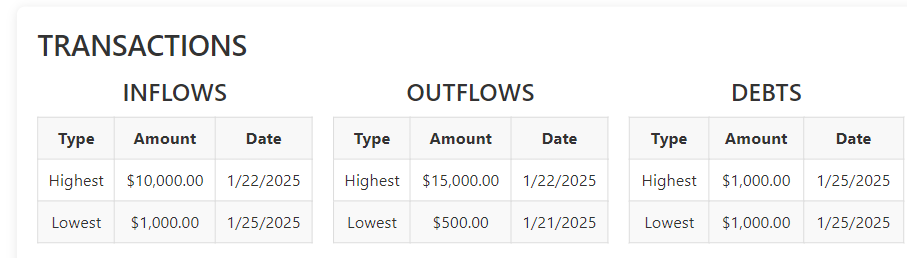


Figure 20 Transaction Table

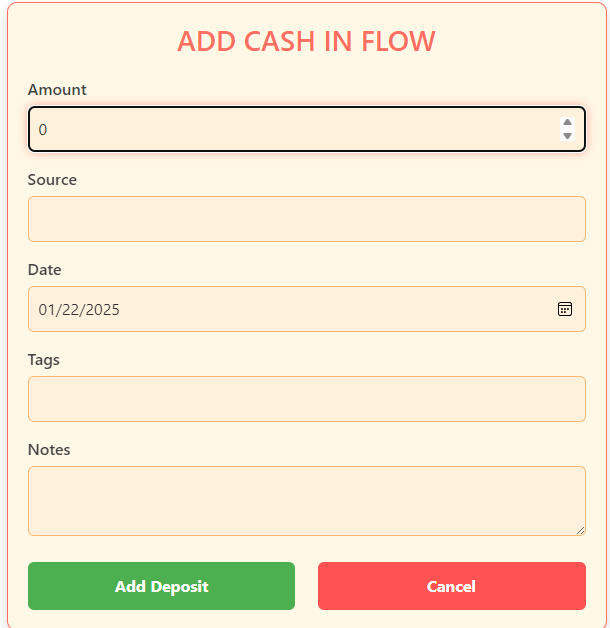


Figure 21 Adding Cashflow

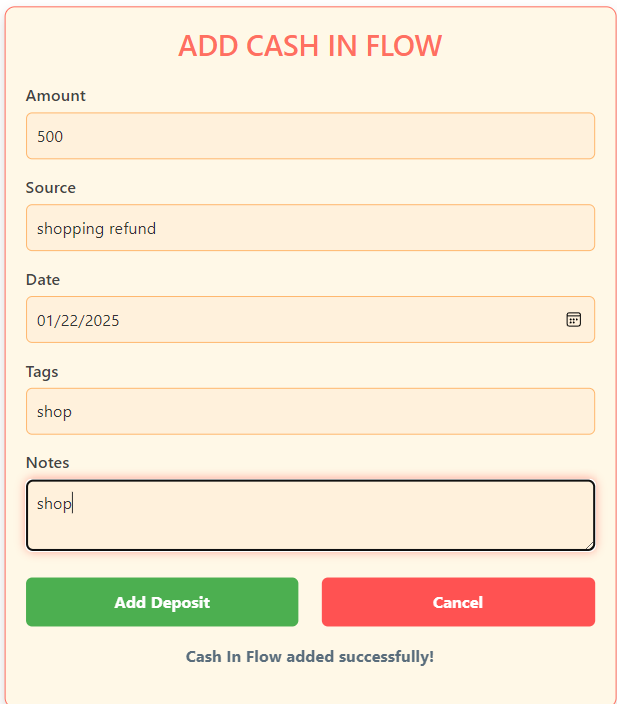


Figure 22 Success message

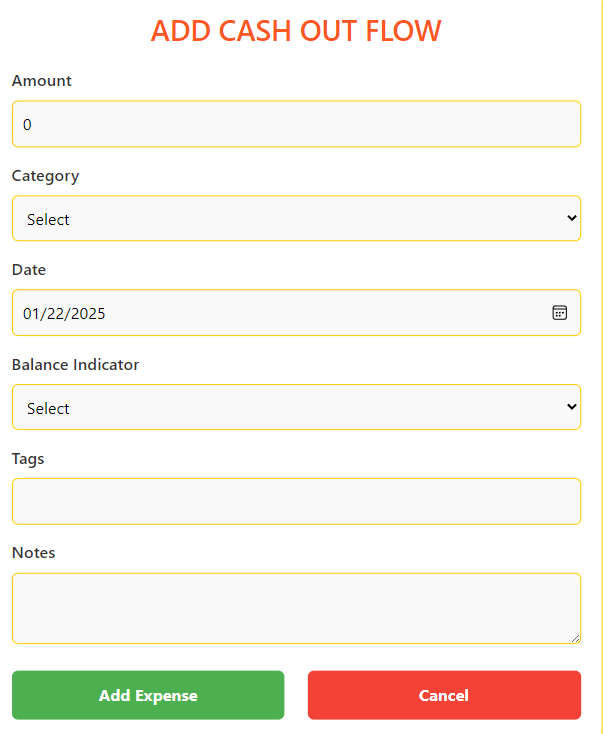


Figure 23 Cashout flow

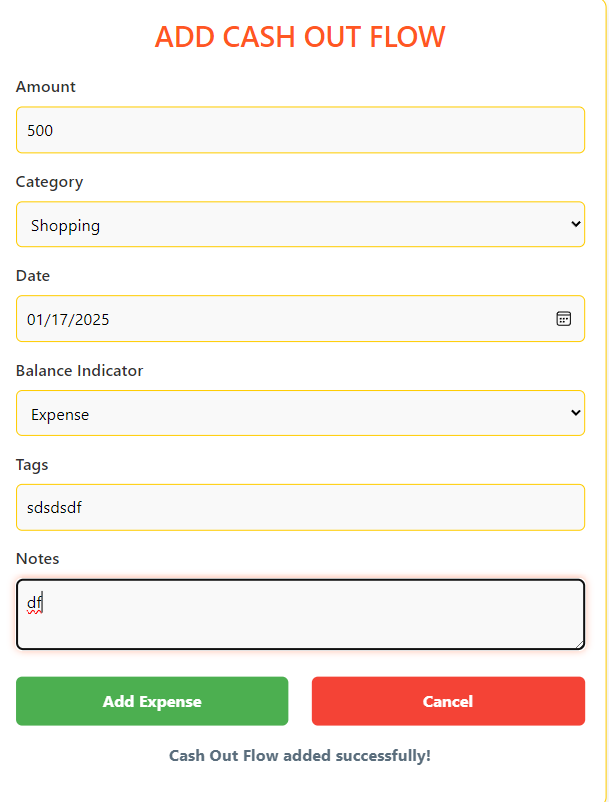


Figure 24 Success after cashoutflow

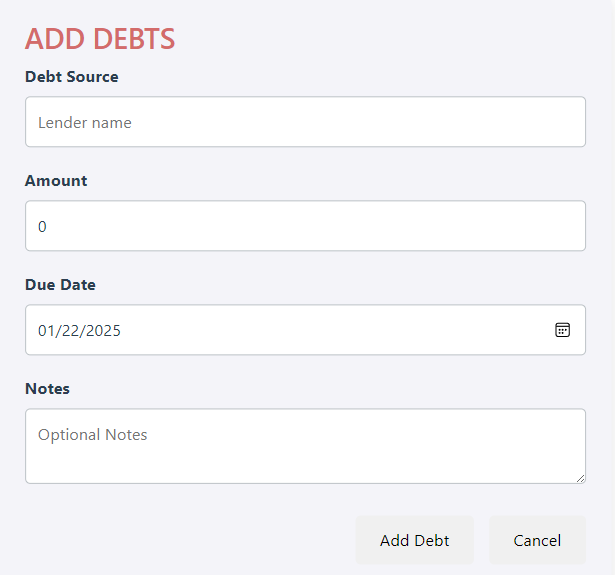


Figure 25 Adding Debts

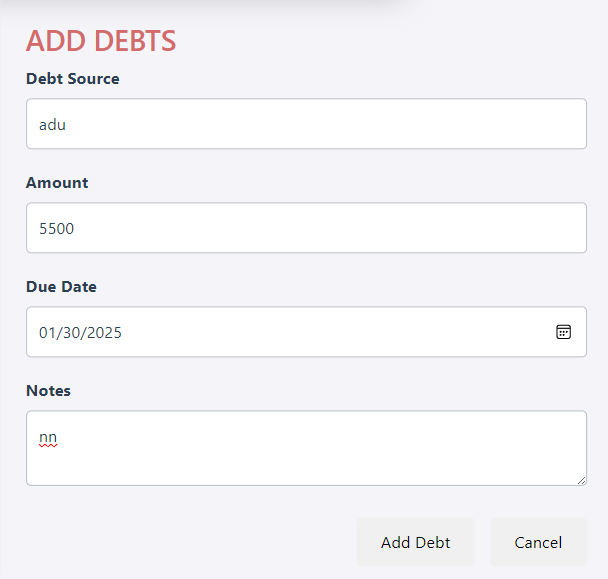


Figure 26 Filling in AddDebt



Figure 27 Success popup

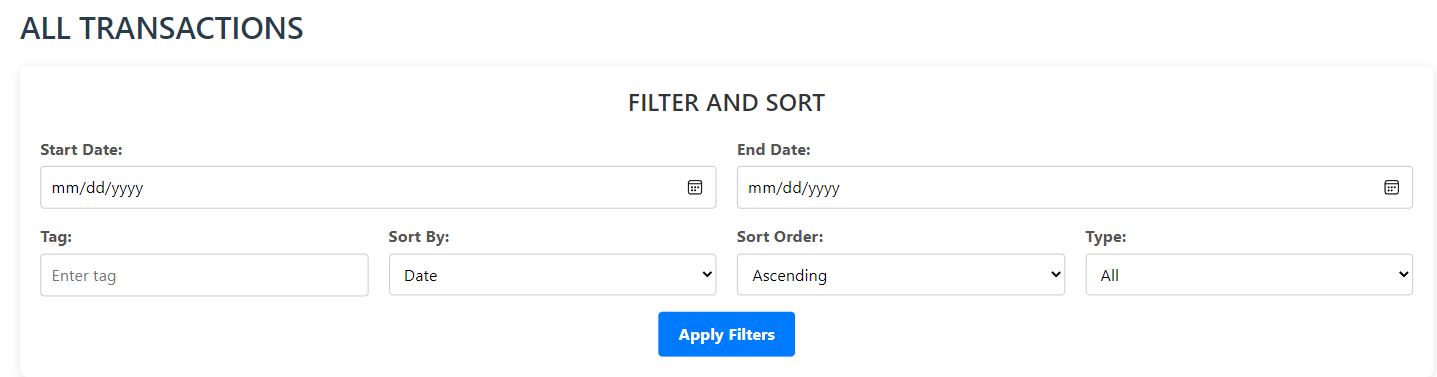


Figure 28 Filter and Sort

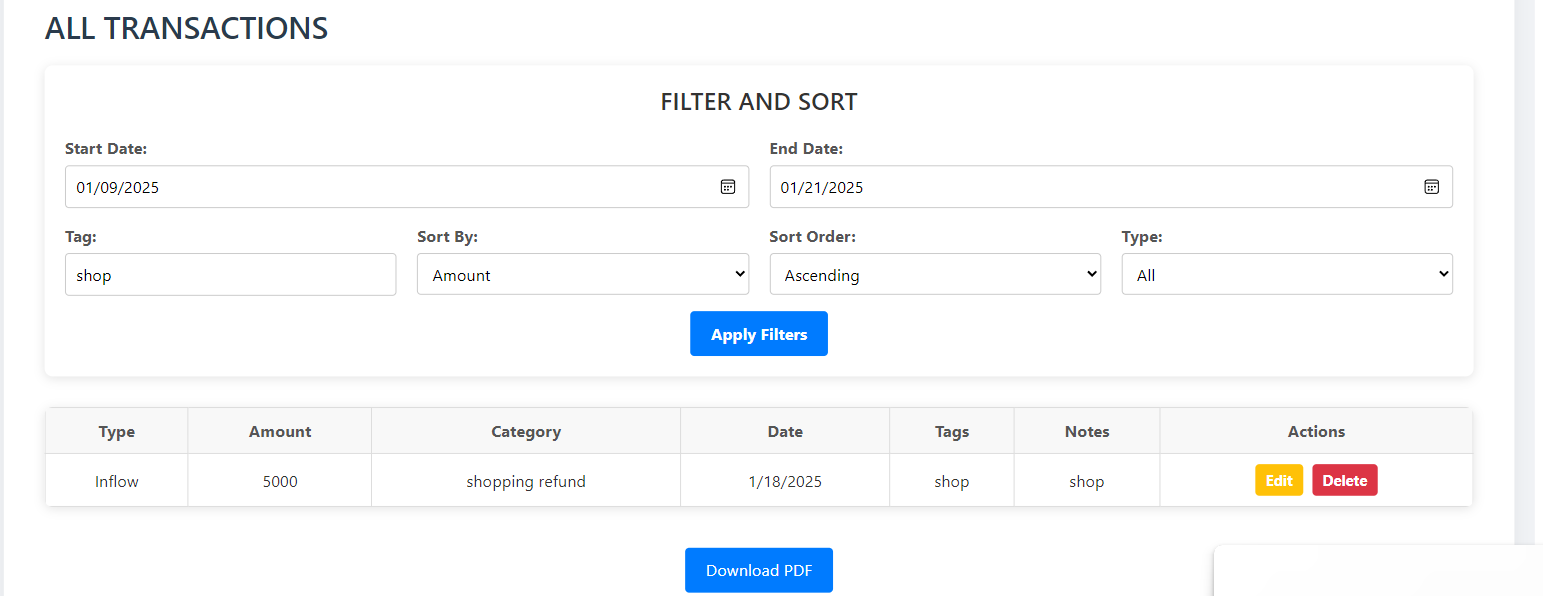


Figure 29 Filter and Sort in Transcation

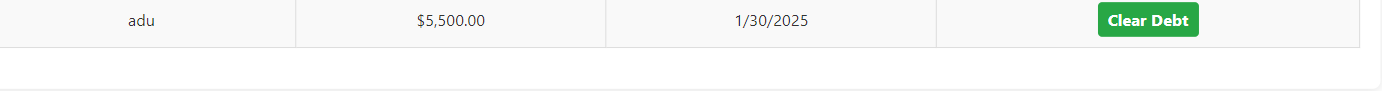


Figure 30 Changes in Dashboard

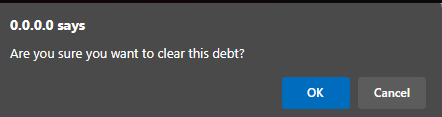


Figure 31 Clear Debts



Figure 32 After Clearing Debts

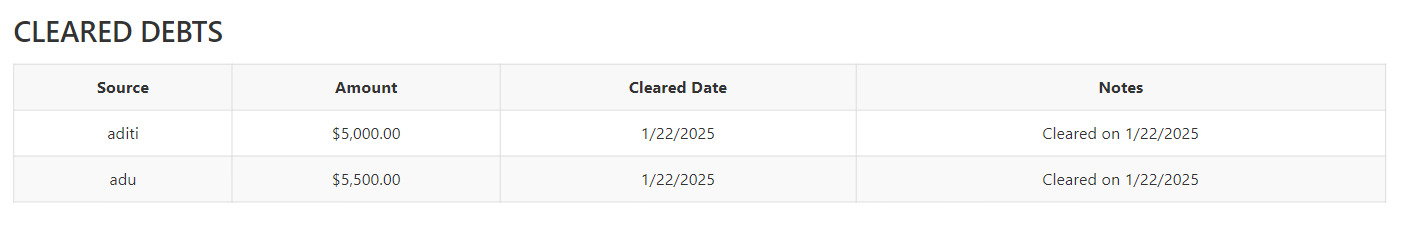


Figure 33 Marked changes

# Individual Reflections

As a student in my third year, this project has proven to be one of the most difficult challenges I’ve faced. At first, I had a hard time grasping the folder structure of a .NET MAUI project, much less implementing features like JSON integration or creating a responsive UI with MudBlazor. The frustration was palpable—endless hours of debugging for issues that were often just a missing semicolon or an incorrectly placed bracket.

Nonetheless, I found some aspects of the module fascinating, particularly the idea of structuring applications using MVVM. It was intriguing to observe how separating concerns resulted in cleaner, more maintainable code. Additionally, I was impressed by MudBlazor’s capability to produce a sleek UI without needing to delve into intricate CSS or JavaScript, which was a relief for me.

However, there were times when I genuinely doubted whether I would finish the project. The looming deadlines were daunting, and juggling this project with my other courses was quite challenging. Yet, reflecting on this experience, I learned valuable lessons in patience, persistence, and the art of effective Googling (an essential skill in development).

# Conclusion

The Personal Expense Tracker exceeded my expectations, especially given the challenges I encountered along the way. It’s practical, easy to use, and fulfills all the criteria set for the project. Although I can’t say I became fond of .NET MAUI, I certainly have developed an appreciation for it and the opportunities it presents.

If I had more time (and fewer time constraints), I would have liked to explore the implementation of advanced analytics or the integration of external financial APIs. But for now, I’m just thankful it functions correctly and satisfies the viva requirements.

This project has been an emotional journey—largely filled with frustration, but also a feeling of achievement. It has taught me that even when situations appear daunting, perseverance can yield results I can take pride in. While I may not be eager to create another .NET application in the near future, I’ve acquired skills and insights that will remain valuable as I progress in my career.