Module Code & Module Title

CS6004NI Application Development

Assessment Weightage & Type

30% Individual Coursework 1 (Second Milestone)

Semester 2024 , Autumn

PROJECT TITLE: MoneyTrack

Student Name: Ashish Pradhan

London Met ID: 22068138

College ID: np01cp4a220484

Module Teacher : Abhishek Khadka

Assignment Due Date: 5th January 2025

Assignment Submission Date: 5th January, 2025

GitHub Repository Link : https://github.com/Ashish-isme/MoneyTrack

Word Count: 772

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher 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.

22068138 Ashish Pradhan 4.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:76633697

Submission Date

Jan 4, 2025, 8:27 PM GMT+5:45

Download Date

Jan 4, 2025, 8:29 PM GMT+5:45

File Name

22068138 Ashish Pradhan 4.docx

File Size

5.1 KB

8 Pages

772 Words

4,218 Characters

0% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

0 Not Cited or Quoted 0%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources

0% Publications

0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Table of Content

Table of Figures

1. Introduction

This documentation serves as a part of the second milestone for the Application development coursework, which is focused on implementing at least 10 features in the personal expense tracking application. The application serves as platform to support every user to track their daily expense, manage budget and help with their transactions. Using .NET Core framework, the goal is to develop an application with enhance user experience, improved functionality and task efficiency.

In this particular milestone, various features have been integrated for user interaction or task with proper backend logics. The features implemented addresses key aspects of the application that will be crucial in data handling to real time updates and transaction management.

With pure focus on the backend operation and implementation of concepts such as Asynchronous programming, Dependency Injection, Service Lifetime, Lambda Expression and many more, 11 core feature of the application have been developed till the milestone. They are as follows :

- Cash Inflows/Outflows/Debt Tracking (i.e. credit, debit, Debt transactions)
- Sufficient balance check for outflows
- Clearing debt from inflows and highlighting pending debts
- Ability to add custom or existing tags/labels to a transaction
- Ability to add optionally a note to a transaction
- Filtering transactions by type, tags, and date range
- Sorting transactions by date
- Display total number of transactions and total transactions (inflows +debts - outflows)
- Display of total inflows, outflows, debt, cleared debt, remaining debt
- Display of highest/lowest inflow, outflow, and debt transactions
- Properly listing pending debts in the dashboard

2. Features list down of MoneyTrack.

2.1 Cash Inflow/Outflow/Debt Tracking feature.

- 'AddCredit' razor component build for adding cash inflow into the application.

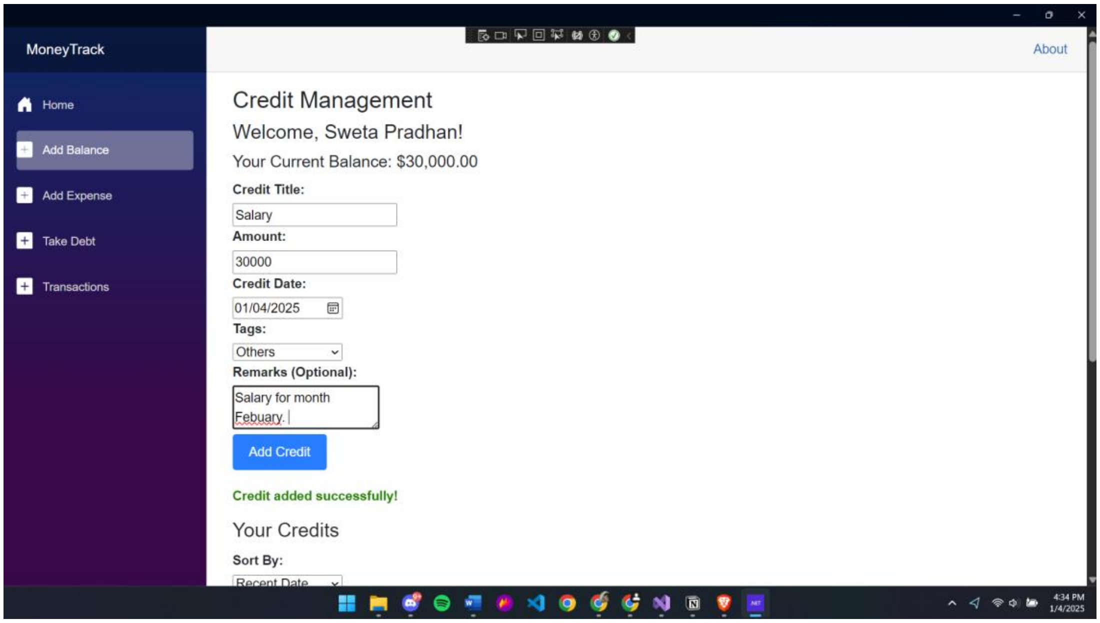

Figure 1 : Cash Inflow feature of the application.

- "AddExpense" razor component build for adding cash outflow into the application.

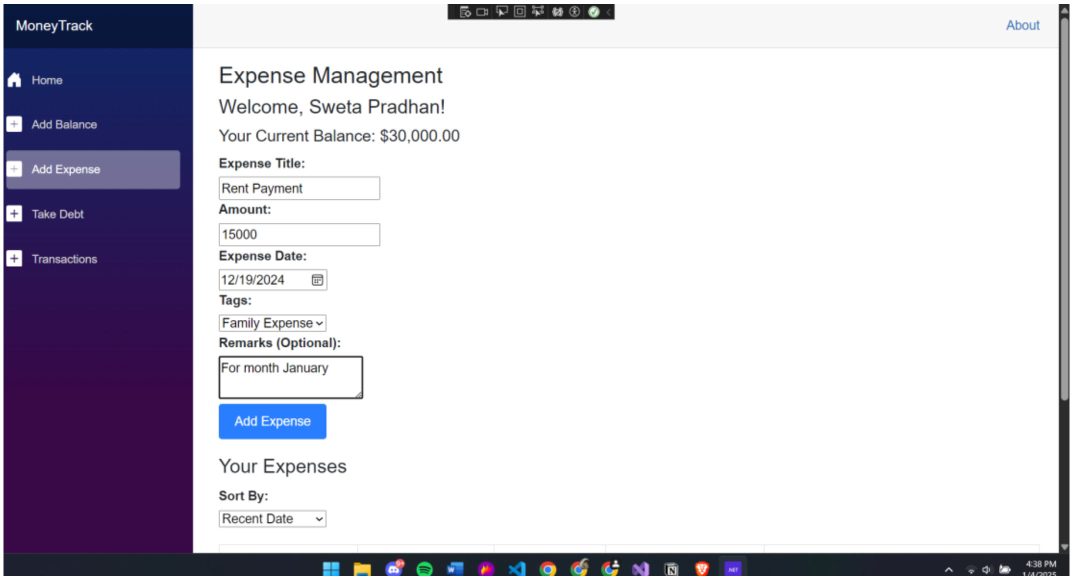

Figure 2 : Cash Outflow feature of the application.

- "TakeDebt' razor component build for adding Debt into the application.

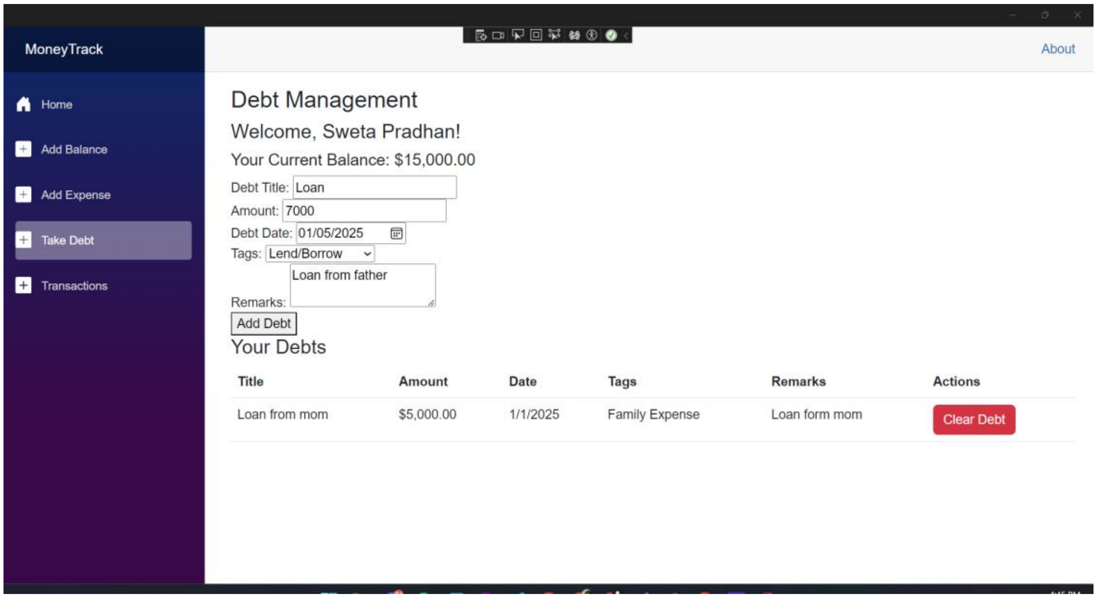

Figure 3 : Debt Adding feature of the application.

2.2 Clearing debts form inflows and highlighting pending debts feature.

- Once the debts are shown it is highlighted as shown below:

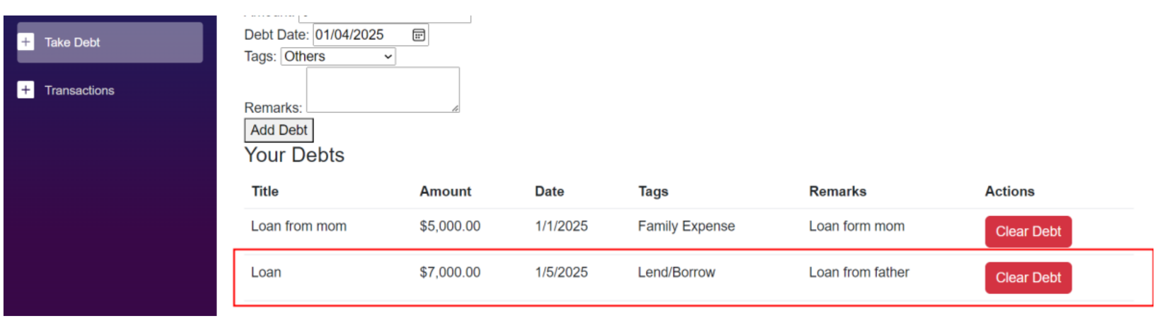

Figure 4 : Debts highlight and clearing feature of the application.

- Once the user has sufficient balance form inflow (Current Balance > Debt Payment Amount) , they can pay the debt easily.

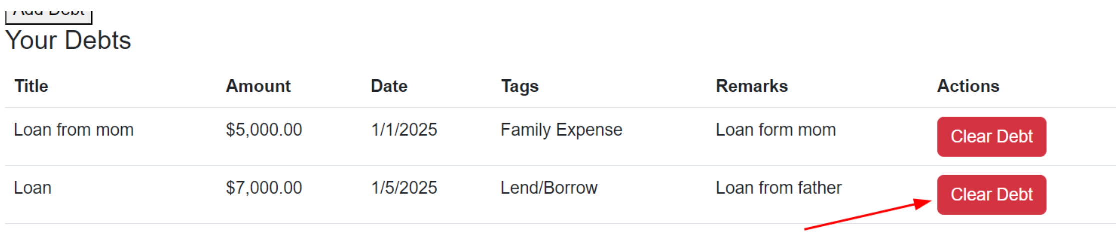

Figure 5 : Debt clear option feature in the application.

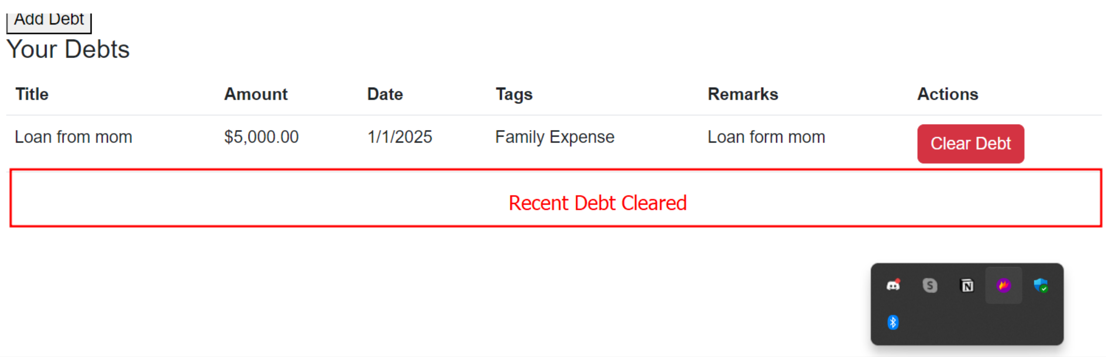

Figure 6 : Debt cleared in the application.

2.3 Ability to add custom or existing tags to a transaction

- There are existing and custom tags/label for any transaction type.

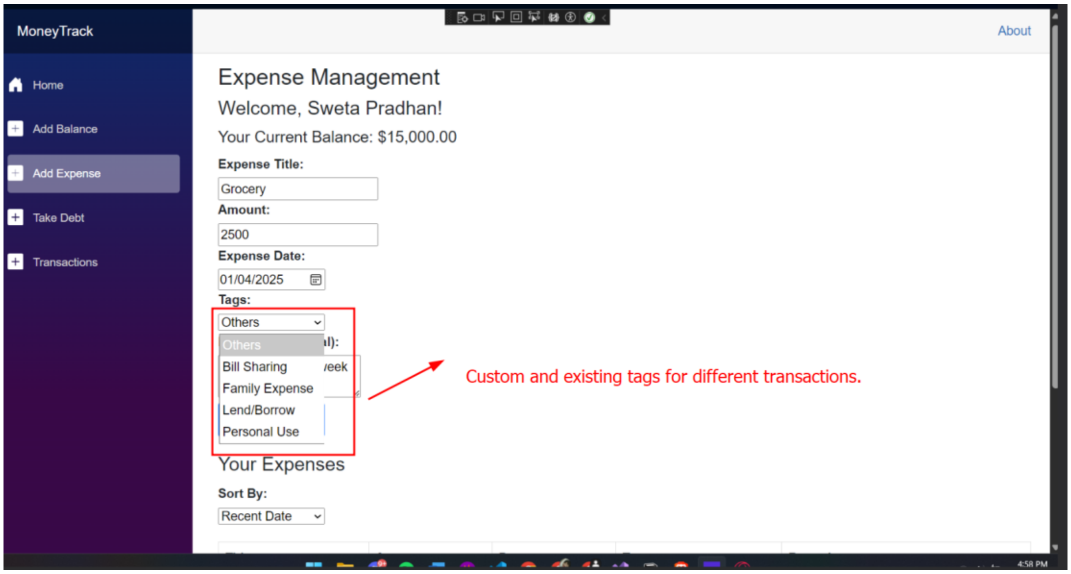

Figure 7 : Adding custom or existing tag in transaction feature in the application.

- The tags are assigned and stored to the transaction accordingly.

Sort By:
Recent Date

Title	Amount	Date	Tags	Remarks
New Fridge	$80,000.00	1/4/2025	Expense	Buying a fridge for home.
Tshirt	$25,000.00	1/4/2025	Expense	For tshirt
300	$3,000.00	1/4/2025	Expense	Food
Chocolate	$50.00	1/4/2025	Personal Use	N/A

Figure 8 : Tags successfully stored in the transaction.

2.4 Ability to add optionally a note to transaction.

- Every user can add an optional remark or note to any transaction for further detailing and information.

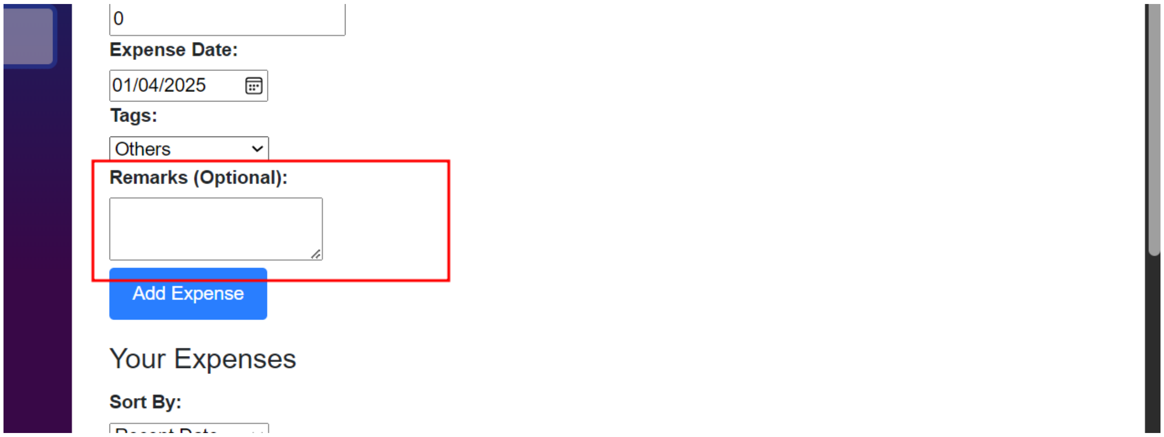

Figure 9 : Adding optional note in transaction in the application

2.5 Properly listing pending debts in the dashboard

- After adding a new debt transaction, all new pending debts are displayed in the application's dashboard.

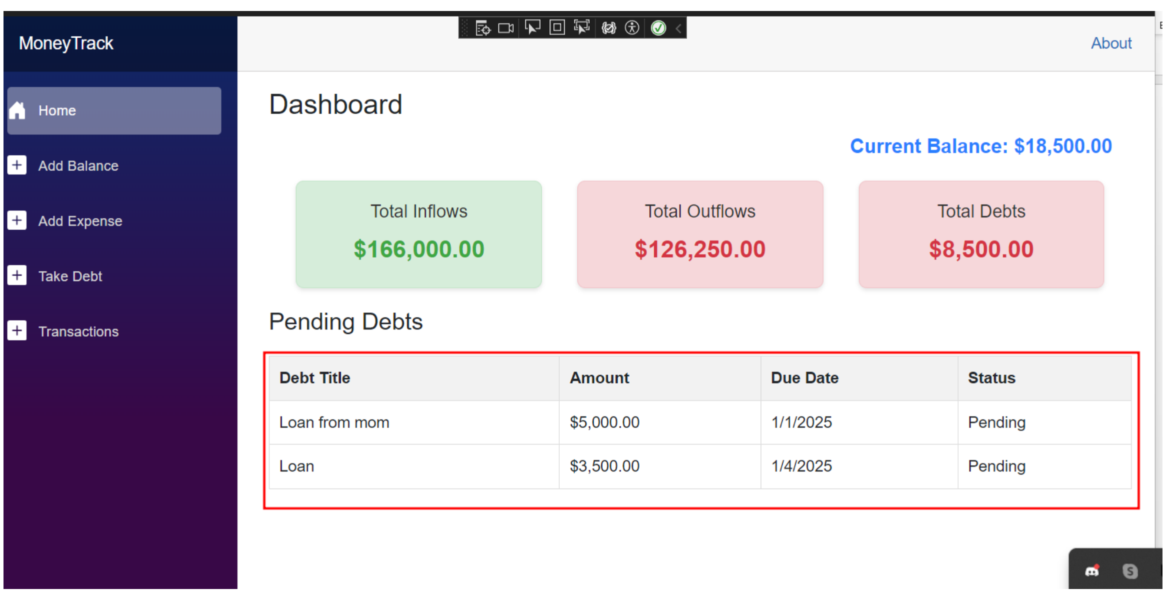

Figure 10 : Displaying pending debts in the application dashboard.

2.6 Display of total inflow, outflows , debts and remaining debts.

- The following details are shown in the dashboard :

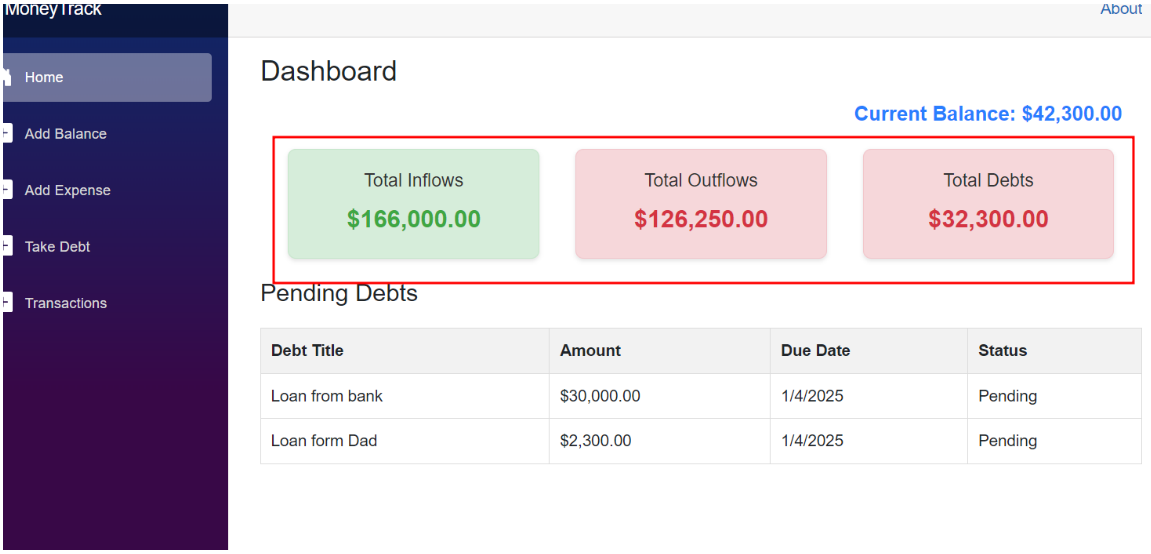

Figure 11 : Display of total inflows, outflows and debts in the application's dashboard.

2.7 Display total number of transactions and total transactions.

- The total transaction is displayed through the "CalculateTotalTransactions()" method. It can be displayed in razor component as below :

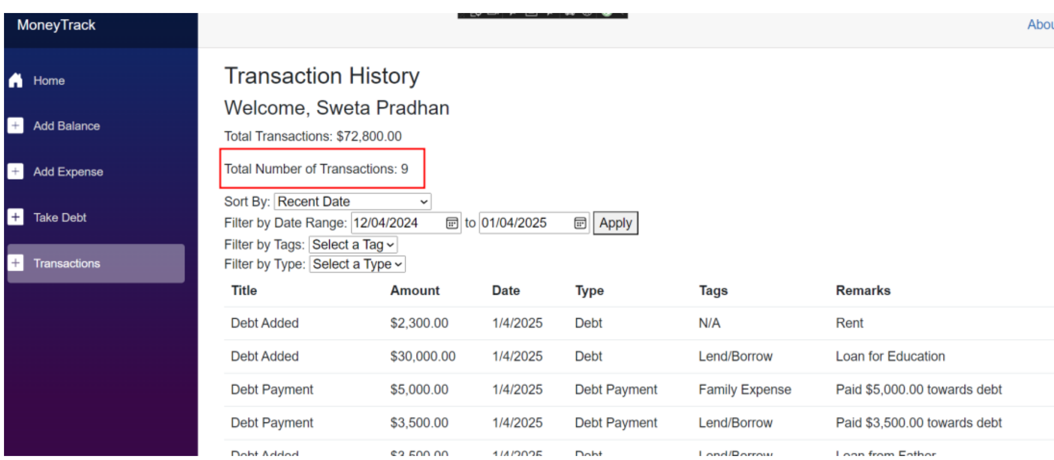

Figure 12 : Displaying total number of transactions in total for the user.

2.8 Sufficient balance check for outflows.

- Entering the details for entering a new cash outflow. Here the Outflow amount is greater than the available balance. Thus, the sufficient balance is checked appropriate message is displayed.

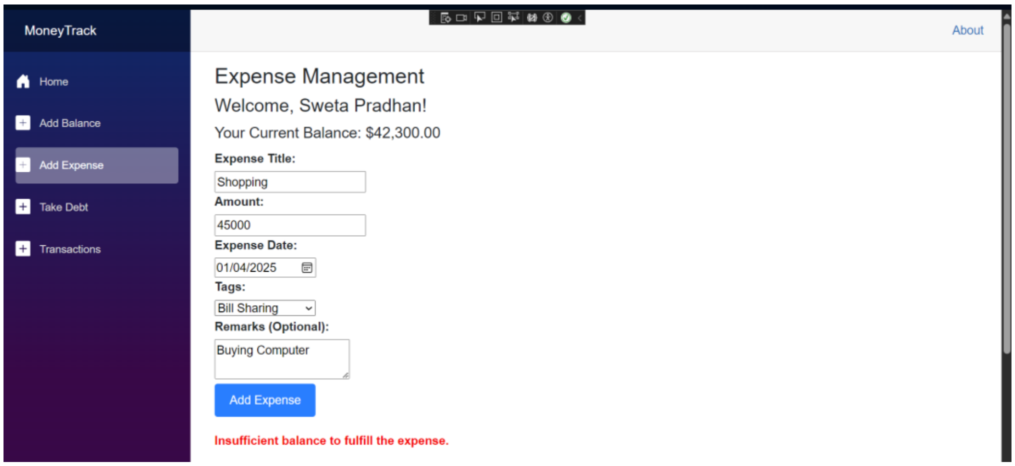

Figure 13 : Checking for sufficient balance before cash outflows.

2.9 Display of highest/lowest inflow, outflow and debt transactions.

- Here, the data for the transaction can be filtered according to the highest and lowest (and vice versa) amount :

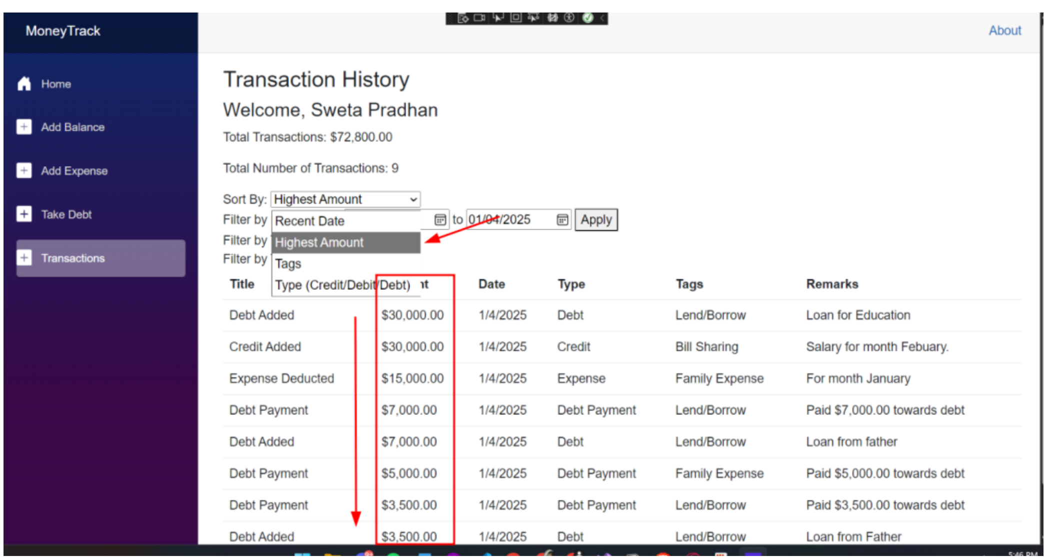

Figure 14 : Displaying transaction list on the basis of highest and lowest order.

2.10 Filtering transactions by type, tags and date range

- The transactions can be filtered according to types of transactions, tags and date range .
- According to the dates .

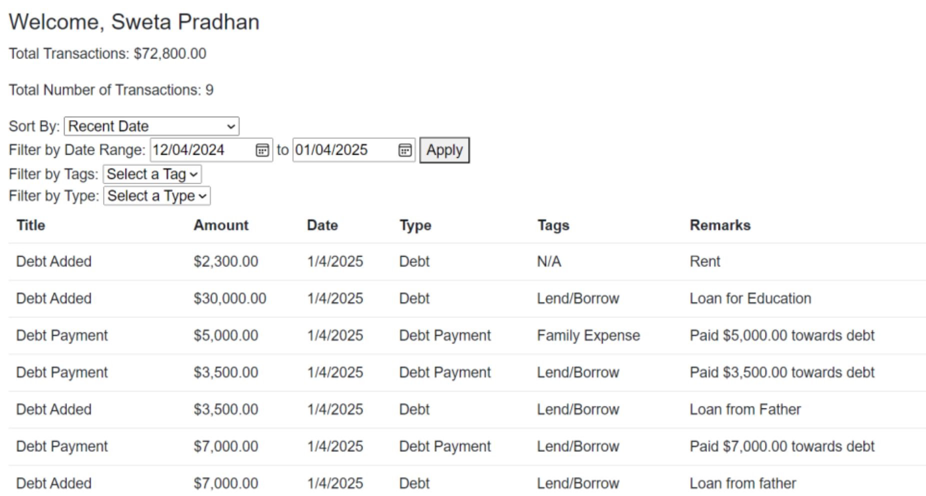

Figure 15 : Filtering transaction list on the basis of dates.

- According to the tags

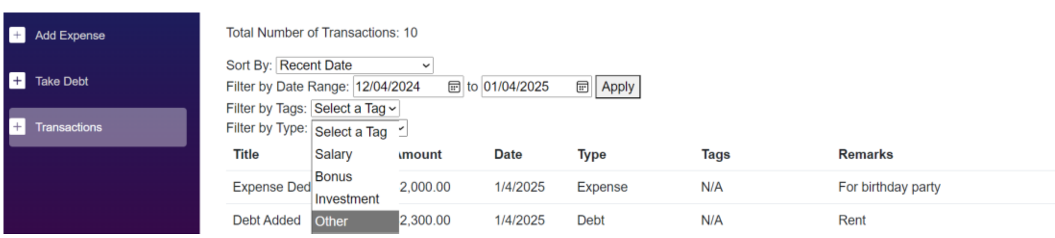

Figure 16 : Filtering transaction on the basis of tags/labels.

- According to the transaction type.

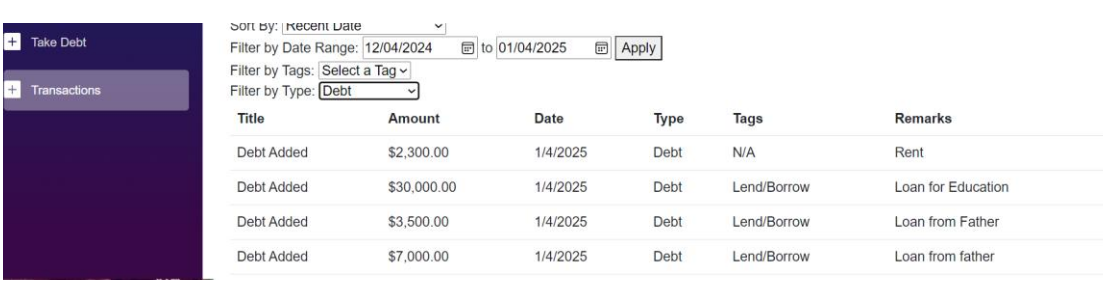

Figure 17 : Filtering data on the basis of transaction type.

2.11 Sorting transaction by date.

- Transactions can be sorted accordingly by specifying the “from” and “to” date .

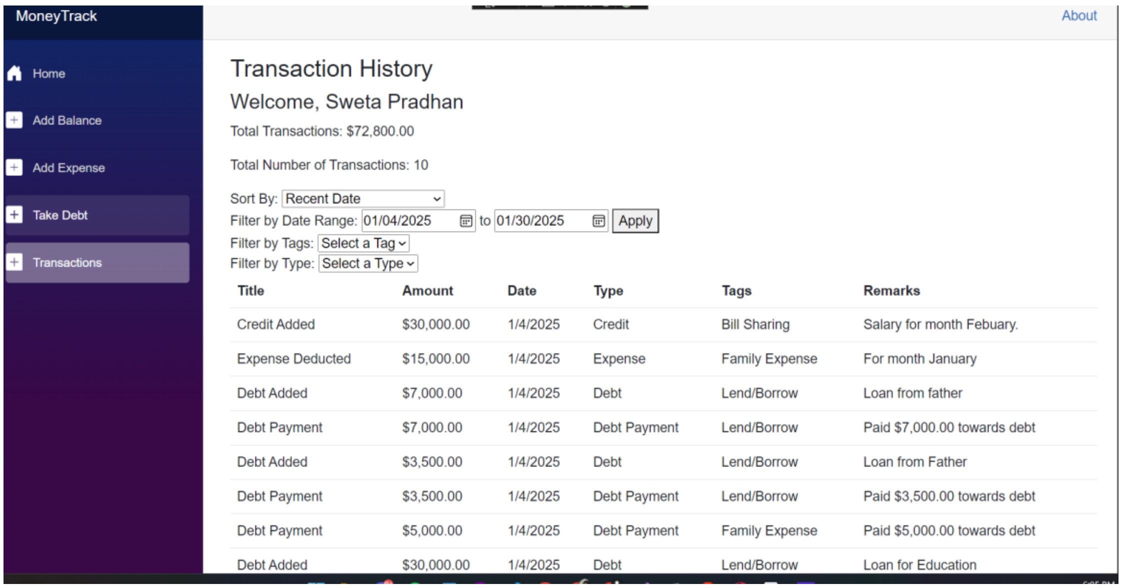

Figure 18 : Sorting transactions on the basis of dates.