FED-ETE

Project Report:
Expense Calculator

NAME – Prabhat Goyal

Roll No – 2110994829

Trainer - Rishi

Introduction

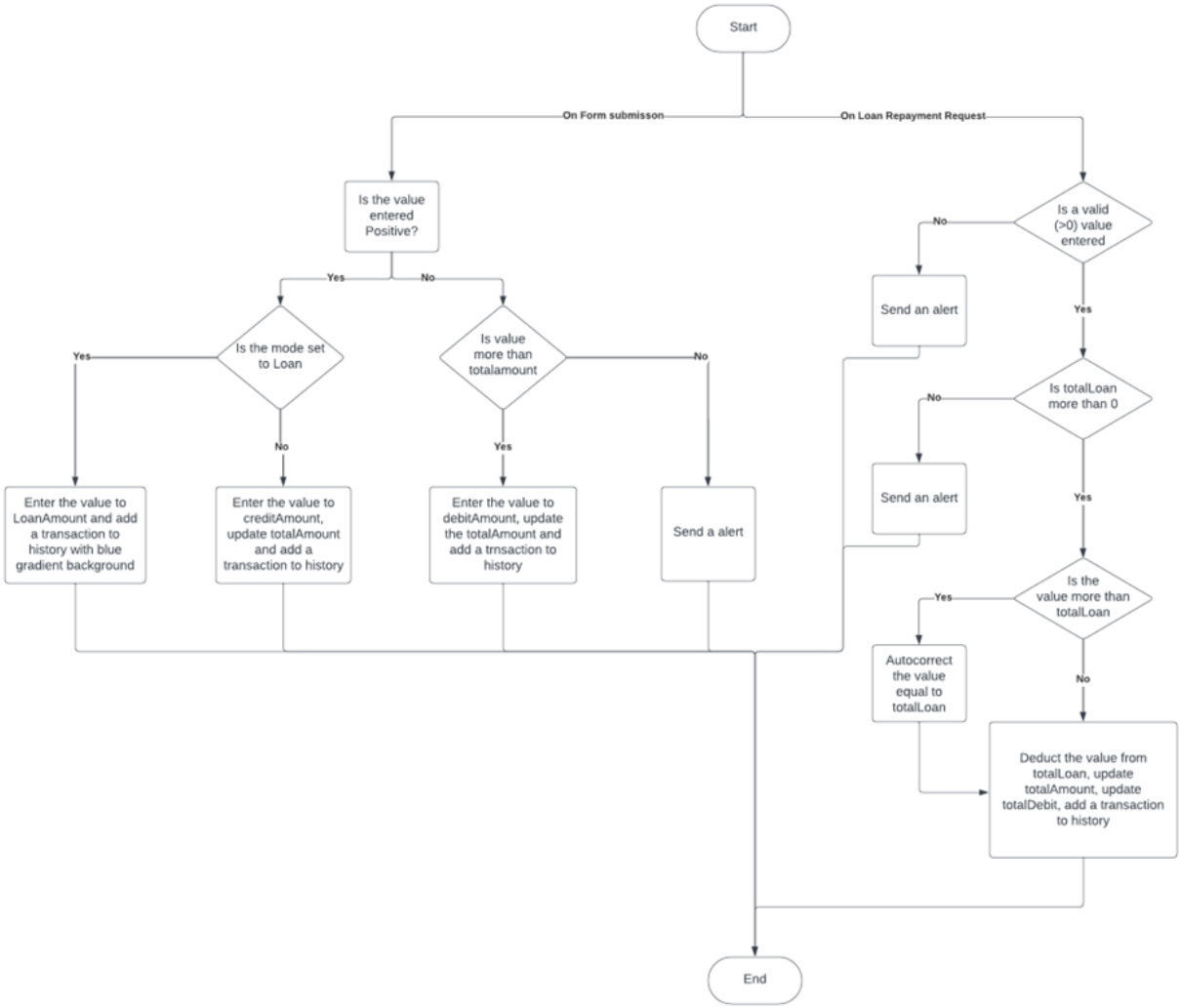
My final submission for this unit is a react based app which is an Expense Calculator. This project is inspired from initial version of “Khatabook” app (or current version of “Cashbook”). This helps user to keep track of all sort of expenses and incomes. User have an option to choose if any debit or credit is cash or online payment. User can also add a description to the transactions. The description helps the user to remember the insights of a transaction in later times. All the transactions are visible to the user in the History tab at the bottom of the page. History also shows the date of the transaction.

Along with debit and credit, user can also input a positive transaction as loan. The word loan is used to showcase any transactions made via credit cards or any money lent from others. If any “loan” is repaid, the entry can be made into the calculator via the “loan repaid” button upon entering the value in the input field alongside it.

The UI of the app is kept very simple and soothing to the eyes. No flashing notifications and no flashy colours. It is kept simple with black and white page, with a splash of colors in the table which is front and centre on launching the app. This table shows the grand total of all the credit transactions, debit transactions individually, and also their resultant value amount left with user. The table also shows total of all unpaid loans accumulated to date.

Working

Next JS framework is used with React JS to build this app. This is done to make the app future ready, when and if routing will be introduced to it. The entire working of the app is dependent on two buttons – “Add a transaction” and “Loan repaid”. The functionality of the buttons can be understood with the entity relation diagram:



Functions

The project is made with simplicity. No fancy functions or algorithms are used in the app.

The app consists of the following class components:

- **Page –** the homepage page of the app. All the function components of the app are defined here. States are mentioned here and layout of app is also defined here.
- **Form –** it contains the form for adding a new transaction to the calculator. It takes one number input as amount, one String input as

Description of transaction, one option from a drop down menu which takes loan, cash and online as options. The next button acts as the Submit button for the form.

- **Table –** displays a one glance summary of all the transactions yet, focusing on two important points – net amount left, net loan left. It also shows total credit amount to date and total debit amount to date.

- **History –** displays the details of all the transactions till date, their amount, description, mode of pyment and also the date.

The app consists of the following simple components:

- **Submit_Handler –** it is tri ggered on submitting the form, checks the value, mode and adds a transaction in the table and history.

- **Loan_Handler –** it is triggered on clicking on “Loan repaid” button. It checks for value, compared to total, loan and updates the values and history, only after a certain set of conditions are met.

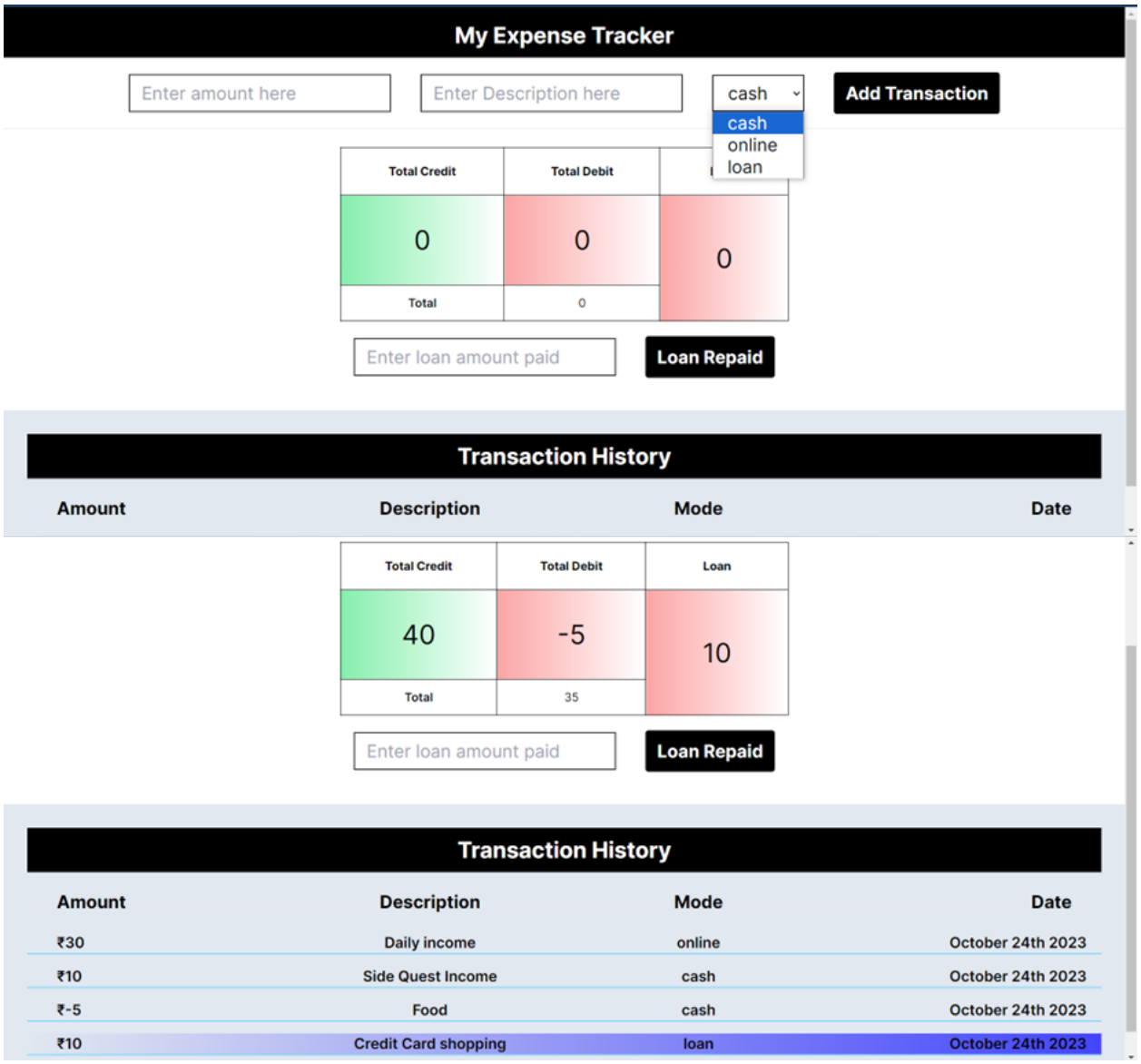
For styling of the website, **tailwind** is used. Tailwind is a CSS framework which is used to rapidly build websites. It is also helpful when building big websites. IF using CSS, a lot of classes and ids needs to be made for optimal styling. Keeping a tracka of all is very difficult. While using tailwind, we do not need to change the code file, no need to use classes and ids. The styling is done inline with the HTML tags and the names of style elements have been shortened. Tailwind also makes websites automatically responsive as well. It is definitely not be all end all solution. It works on baseline CSS. Higher functions still needs to be built with CSS.

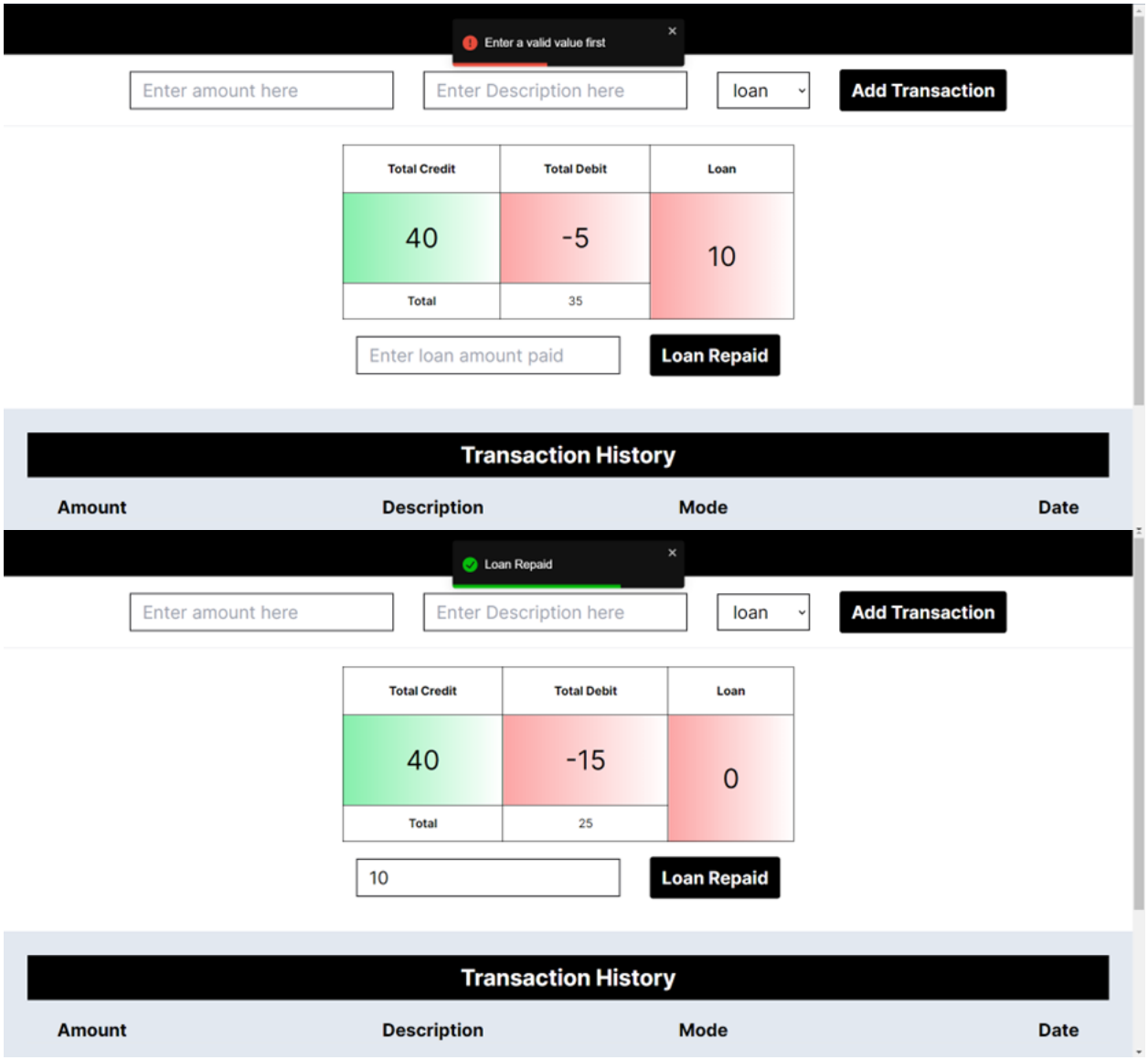
The following JS libraries and functions are used:

- **React -** the app is built on React JS library. React is a JS library to build user interfaces and webapps. It helps in better rendering execution by letting the programmer decide the order and priority. It also provides a seamless and fast experience to the user rather than traditional website designing.
- **Next** - instead of using React Native, the project is built with React on Next JS framework. This is because Next JS is used in both client side and server side of the webapp. Due to this, integration and transfer of data from one end to another is much simpler to implement. Apart from this, next also provides better routing. Useful when making a webapp with multiple pages and programmer wants to integrate them together.

- **Moment** - it is a JS library to get date and time from the user's machine. This library is used to get date for the History.
- **Toastify** - toastify is a react library used to add toast notifications. It uplifts the premium feel of the website instead of using browser alert notifications.

Screenshots





Future Additions

The following functionality can be added to the app in the future:

- **Conversion calculator** – useful for the situations when the user sends or receives money in foreign currencies. Calculator shows a range to the user based on current conversion rate keeping conversion charges in account.
- **Collaborations** – useful for scenarios when user shares daily expenses with other people. We can add a collaborations page, which will keep

record of all money spent via multiple users. At the end of month/week/etc, each user will get a total showing all the money they owe or have to take from each user in the collaboration.

- **Filter and Search** – add a search functionality for users to easily find a previous transaction. User can either enter the description, amount or mode of transaction, the algorithm automatically picks up which type of field is entered and shows results accordingly.

GitHub Link - https://github.com/DeadStalker69/FED-ETE.git

Presentation Link - Canva