

**DESI ASCEND EDUCARE PROGRAM**  
**2023-24**

# SPLITSMARTY

A bill-splitting website

## PROJECT REPORT

Presented By:

Komal

Praneetha Kancharla

Sanjana Vajrapu

Tanishka Misra

Vaani Goyal

# TABLE OF CONTENTS

- 3    Executive Summary
- 4    Requirements
- 5.   Application Architecture Overview
- 6.   Activity Diagram
- 7.   ER Diagram
- 8.   Component Details
- 11.   Snippets
- 16.   Learnings From Ascend Educare
- 17.   Future Scope
- 18.   Contributions by Members
- 19.   References

# EXECUTIVE SUMMARY

SplitSmarty is an innovative online platform designed for bill-splitting and expense management.

The website features **a user-friendly interface**, starting with a login/signup page, followed by a **personalized user dashboard** where users can edit their profile details.

Users can create groups to share expenses among members, either uniformly or non-uniformly providing flexibility and convenience in managing shared expenses..

The website incorporates a **minimising transactions graph algorithm** to optimize the number of transactions within a group.

Users can **add their own expenses**, categorizing them by type (e.g., food, travel, shopping) to track and manage their spending habits effectively.

**The groups page** allows users to view all groups they are a part of and create new groups with multiple participants. Users also have the option to leave a group or delete it(if they are the Admin).

The tech stack used for this website is Express.js and NodeJs (Backend); ReactJs, CSS (Frontend), and MySQL, Sequelize (Database Management). The concepts of OOPs, DBMS, design and architecture were extensively used, along with the use of optimization algorithms.

# REQUIREMENTS

SplitSmarty is built using the following technology stack:

- **Frontend**

- JavaScript (React.js)

- **Backend**

- Node.js
- Express.js

- **Database**

- Sequelize and MySQL (Relational Database Management System)

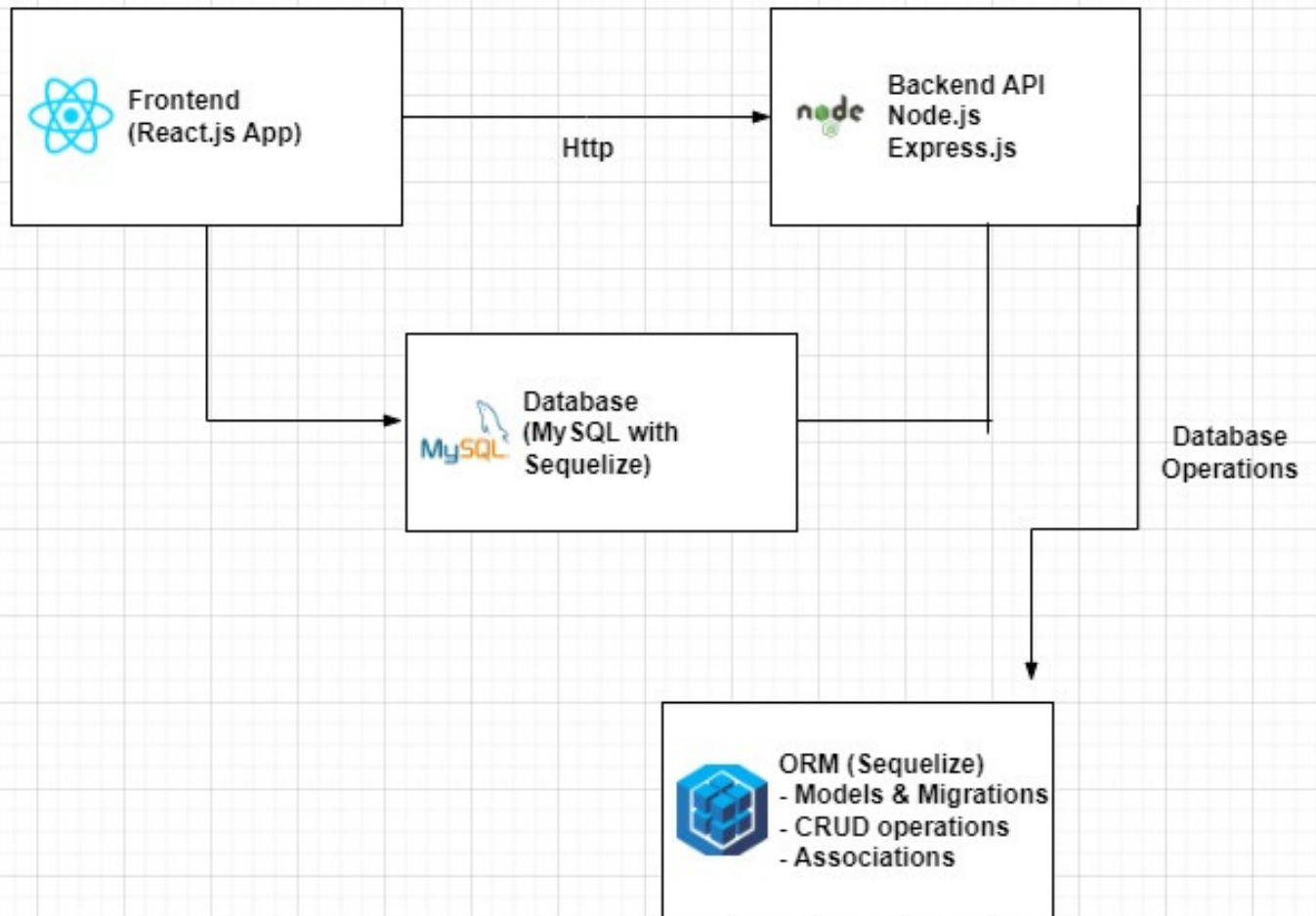
- **Authentication**

- JSON Web Tokens (JWT)

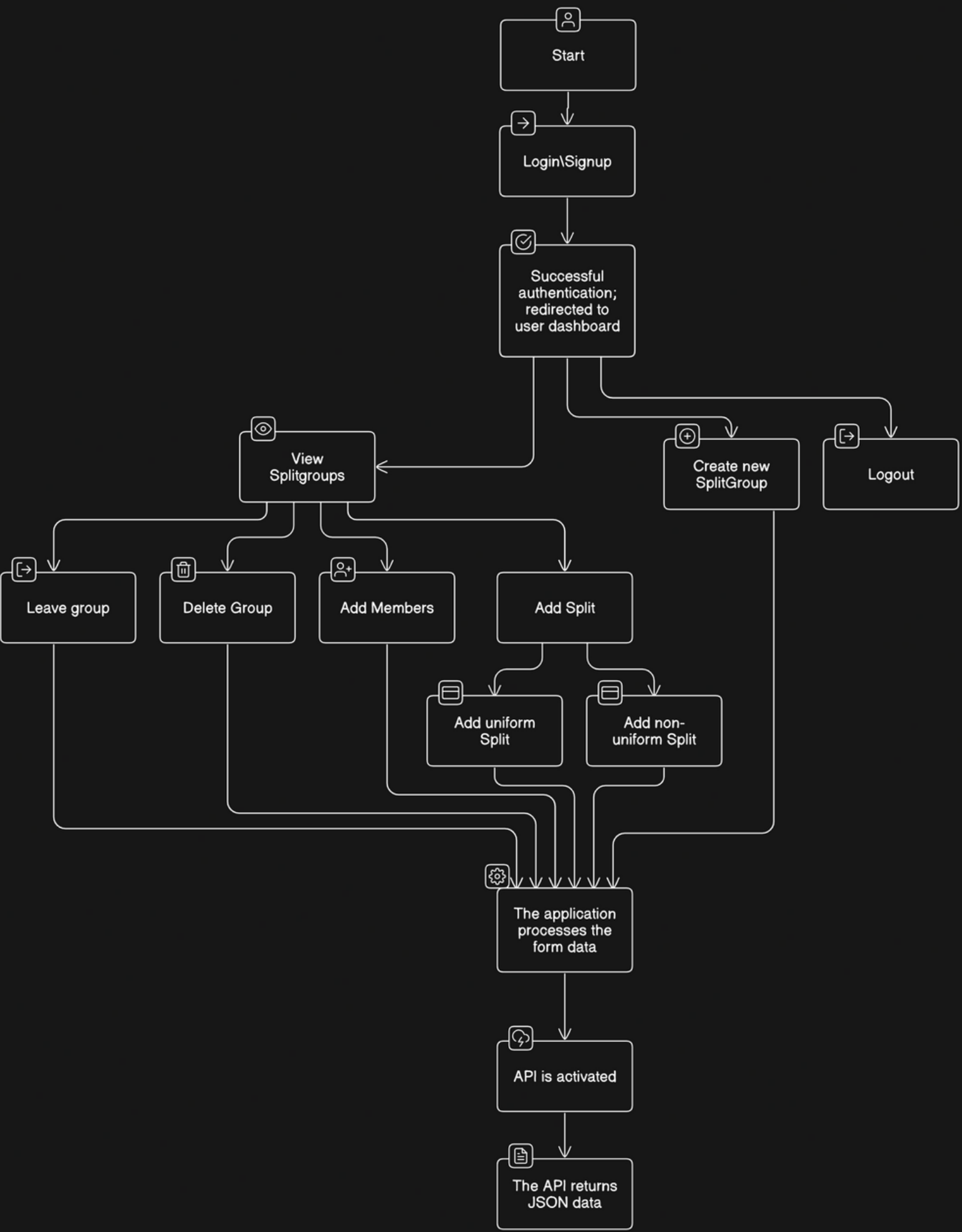
- **Development Tools**

- Git
- npm (Node Package Manager)

# APPLICATION ARCHITECTURE OVERVIEW

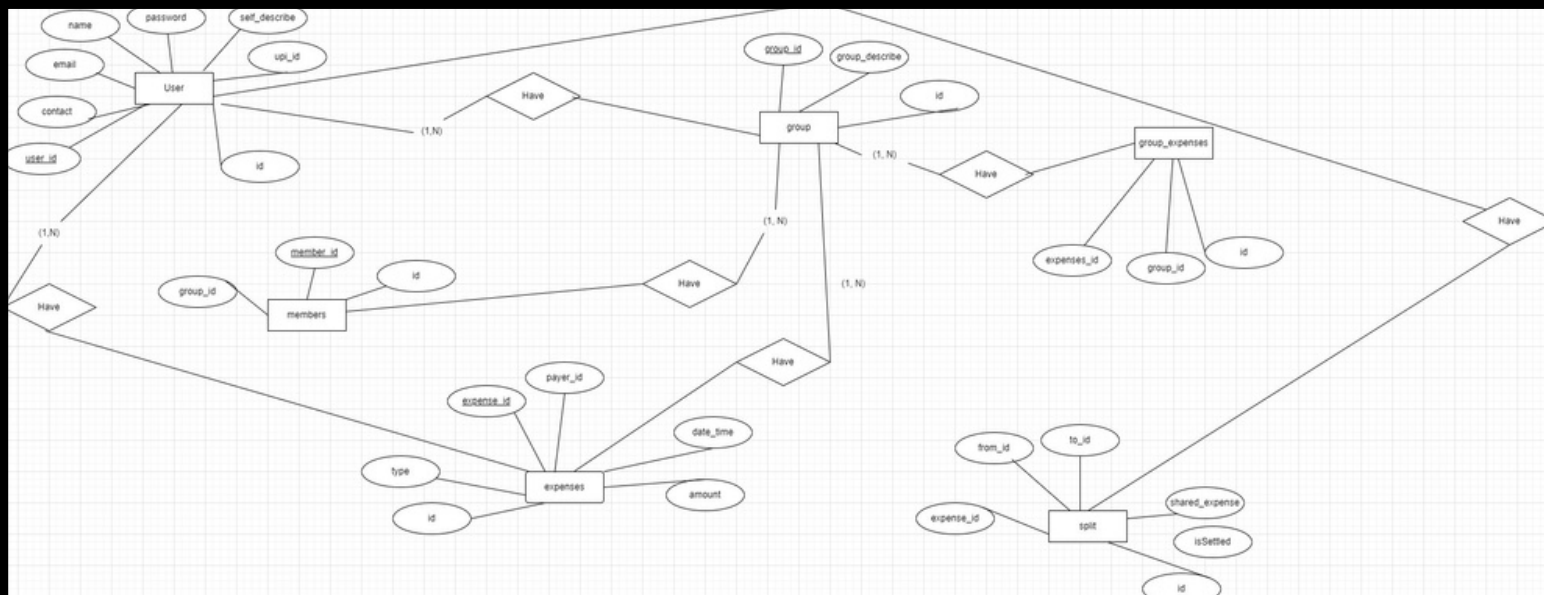


Activity flow diagram for SplitSmarty



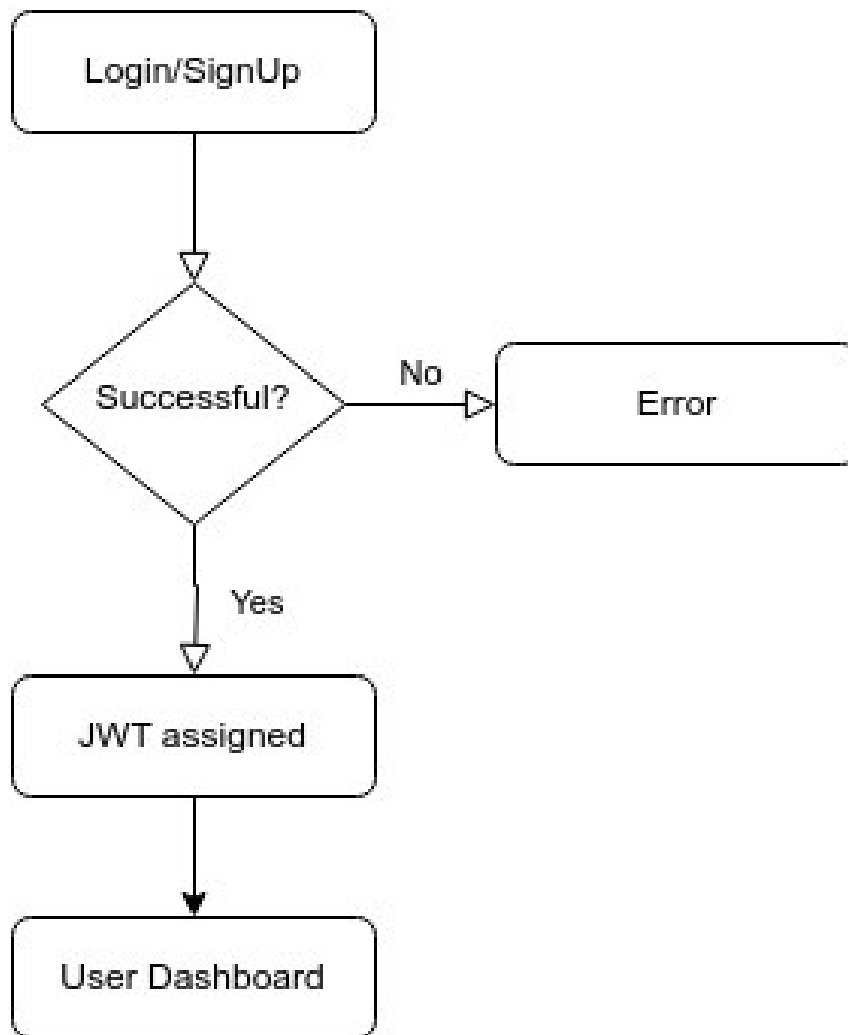
# ER DIAGRAM

console.log(-"SPLITSMARTY"-);



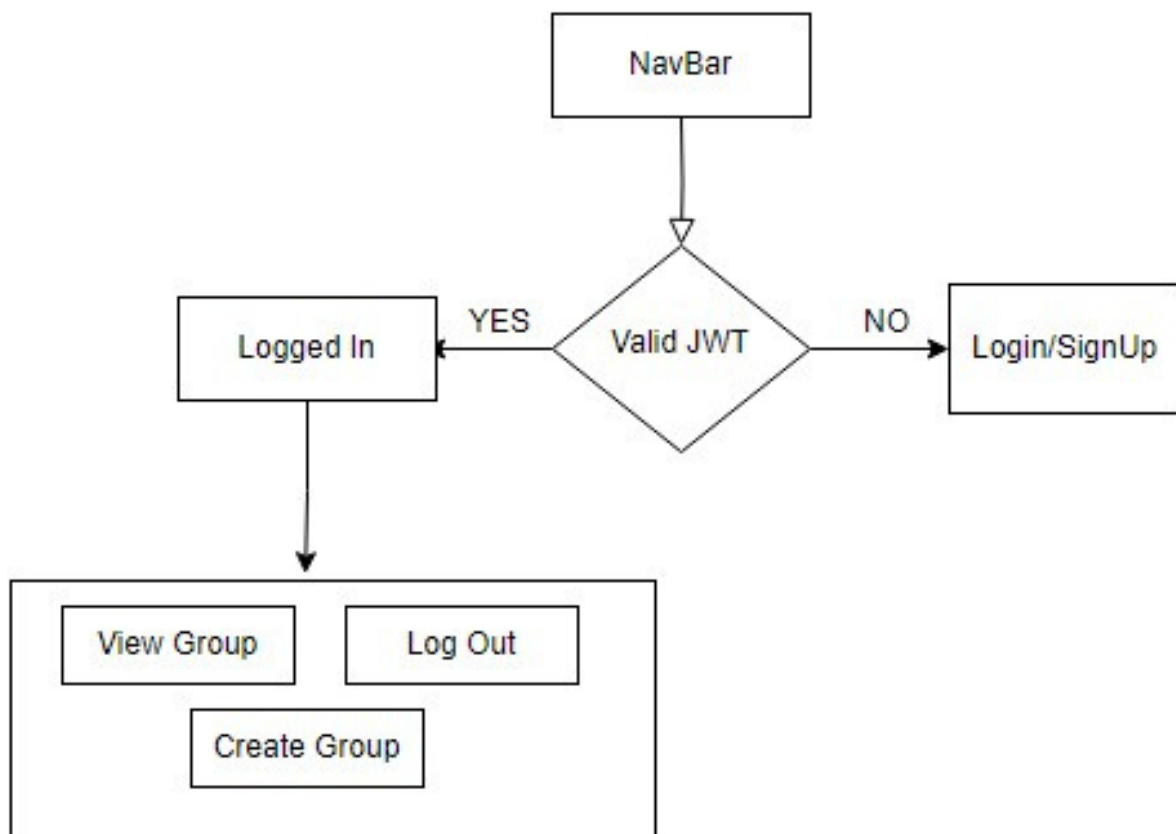
# COMPONENT DETAILS

For all protected routes, a middleware checks whether the JWT sent in the header of the API is valid or not. If it's invalid ( altered/ null/ expired) the user is redirected to the login/signup page.



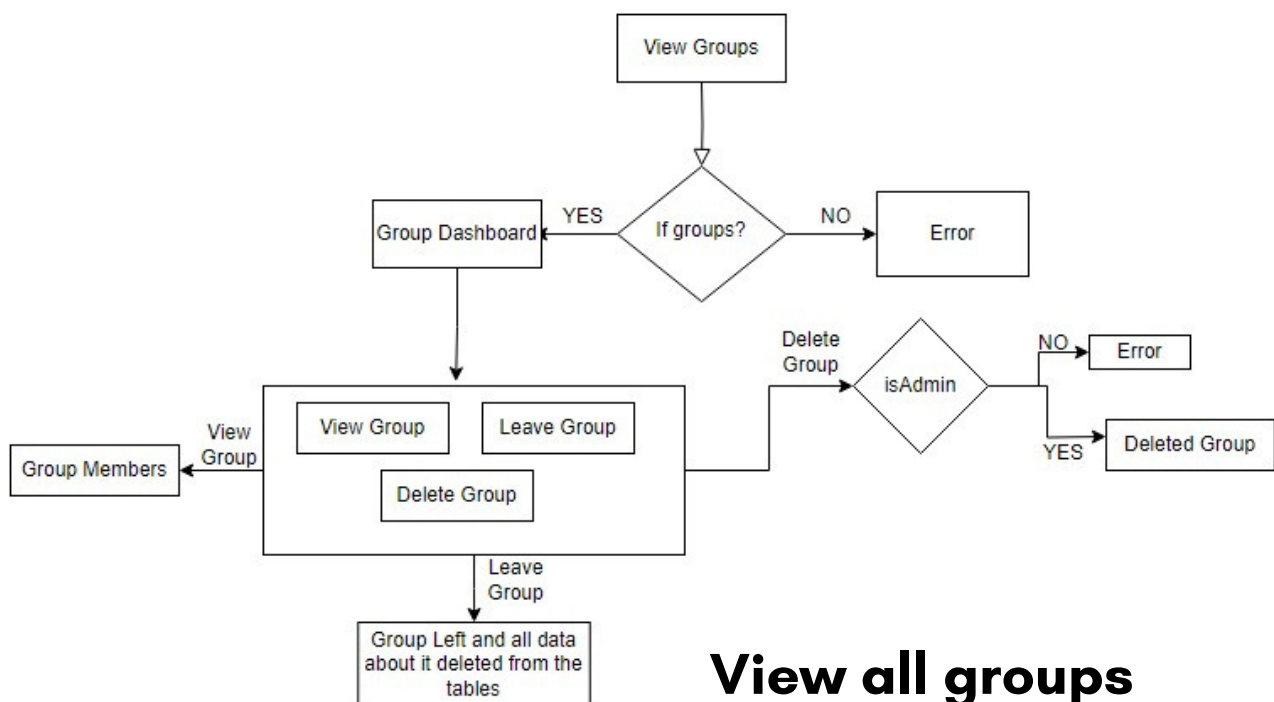
**LOGIN/SIGNUP**





## Navigation Bar

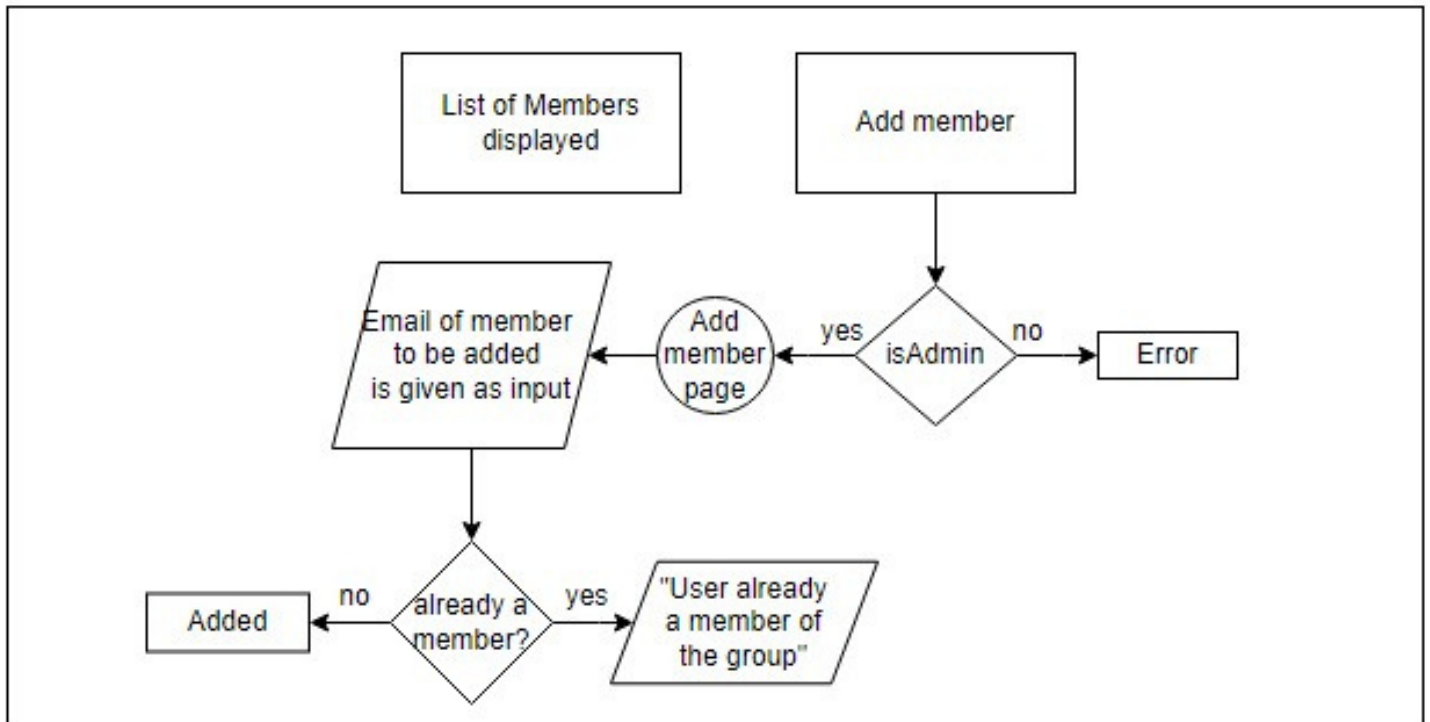
- A person can choose to view group or create a new one in which he will automatically become the admin



## View all groups

- A person is Admin can DELETE a group, may leave it or view it to use other functions

## Group Members

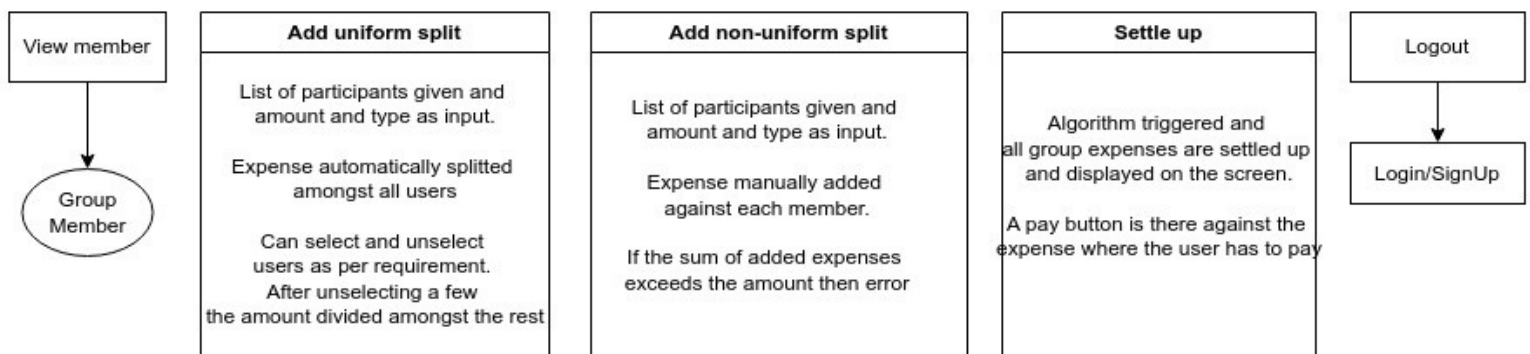


## Group Dashboard

- A user if admin can add others by their email into the group
- all members are displayed

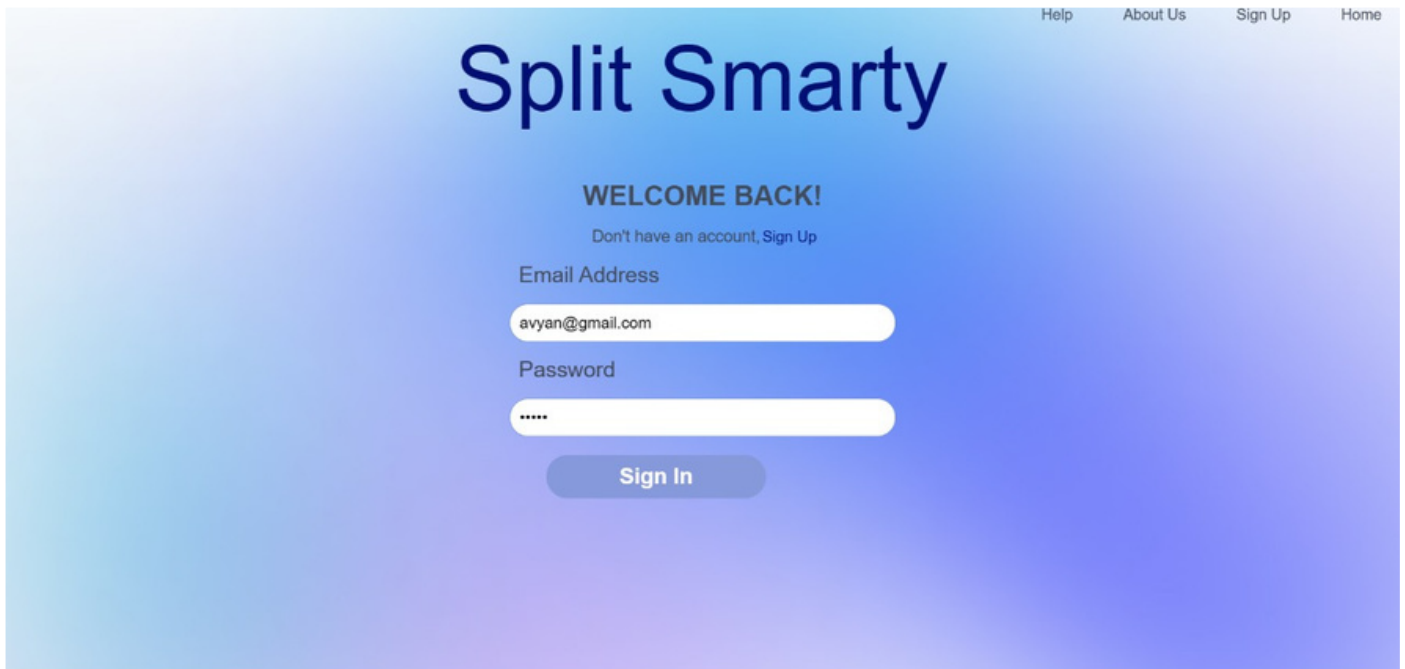
## Group Expenses

List of all group expenses added by members of the group displayed:  
by, amount, date\_time



## Expense and Settle Up

# SNIPPETS



Help About Us Sign Up Home

## Split Smarty

**WELCOME BACK!**

Don't have an account, [Sign Up](#)

Email Address

avyan@gmail.com

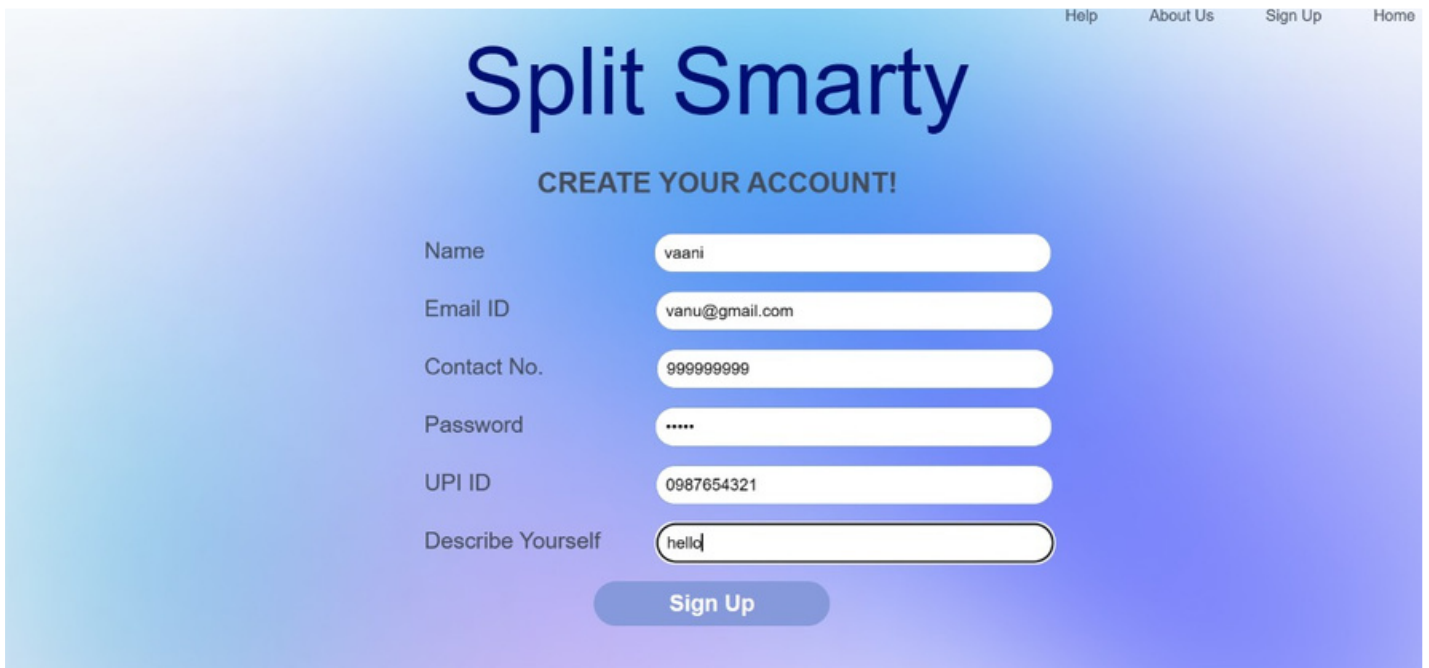
Password

\*\*\*\*\*

[Sign In](#)

This is a login form for 'Split Smarty'. It features a blue gradient background. At the top right, there are links for 'Help', 'About Us', 'Sign Up', and 'Home'. The main heading is 'Split Smarty' in a large, dark blue font. Below it, the text 'WELCOME BACK!' is displayed in bold. A link 'Don't have an account, Sign Up' is provided. The form includes two input fields: 'Email Address' with the value 'avyan@gmail.com' and 'Password' with masked characters '\*\*\*\*\*'. A 'Sign In' button is located at the bottom.

Login



Help About Us Sign Up Home

## Split Smarty

**CREATE YOUR ACCOUNT!**

Name

vaani

Email ID

vanu@gmail.com

Contact No.

999999999

Password

\*\*\*\*\*

UPI ID

0987654321

Describe Yourself

hello

[Sign Up](#)

This is a sign-up form for 'Split Smarty'. It features a blue gradient background. At the top right, there are links for 'Help', 'About Us', 'Sign Up', and 'Home'. The main heading is 'Split Smarty' in a large, dark blue font. Below it, the text 'CREATE YOUR ACCOUNT!' is displayed in bold. The form includes six input fields: 'Name' (vaani), 'Email ID' (vanu@gmail.com), 'Contact No.' (999999999), 'Password' (\*\*\*\*\*), 'UPI ID' (0987654321), and 'Describe Yourself' (hello). A 'Sign Up' button is located at the bottom.

Sign Up

HelpAbout UsSign UpHome

# User Profile

Name: avyan  
Contact No: 1234567890  
Email Address: avyan@gmail.com  
UPI ID: 12345

View Groups

Create Group

Log Out

## User Profile

HelpAbout UsSign UpHome

# User Profile

Enter Group Name

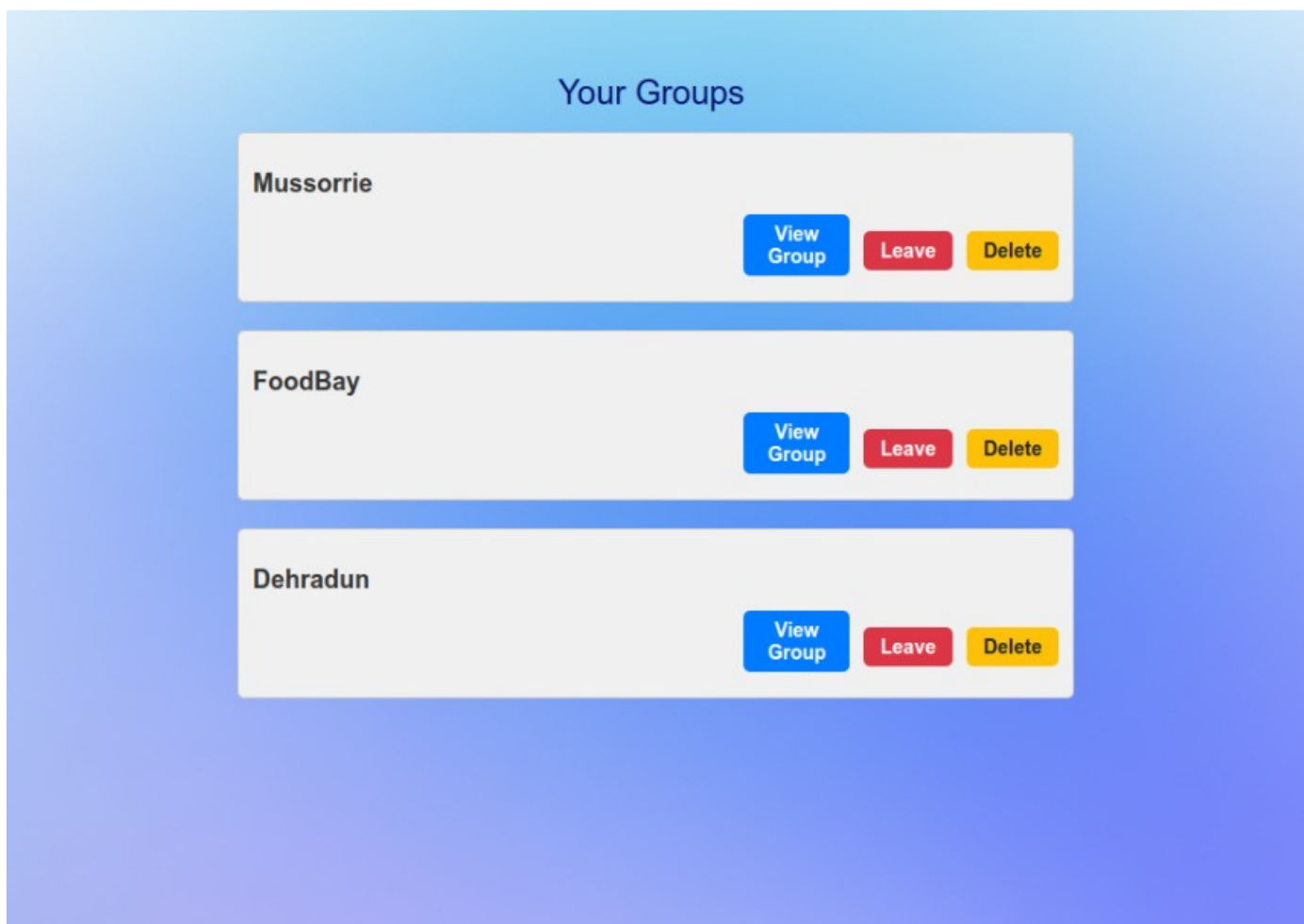
Group 4

Enter Group Description

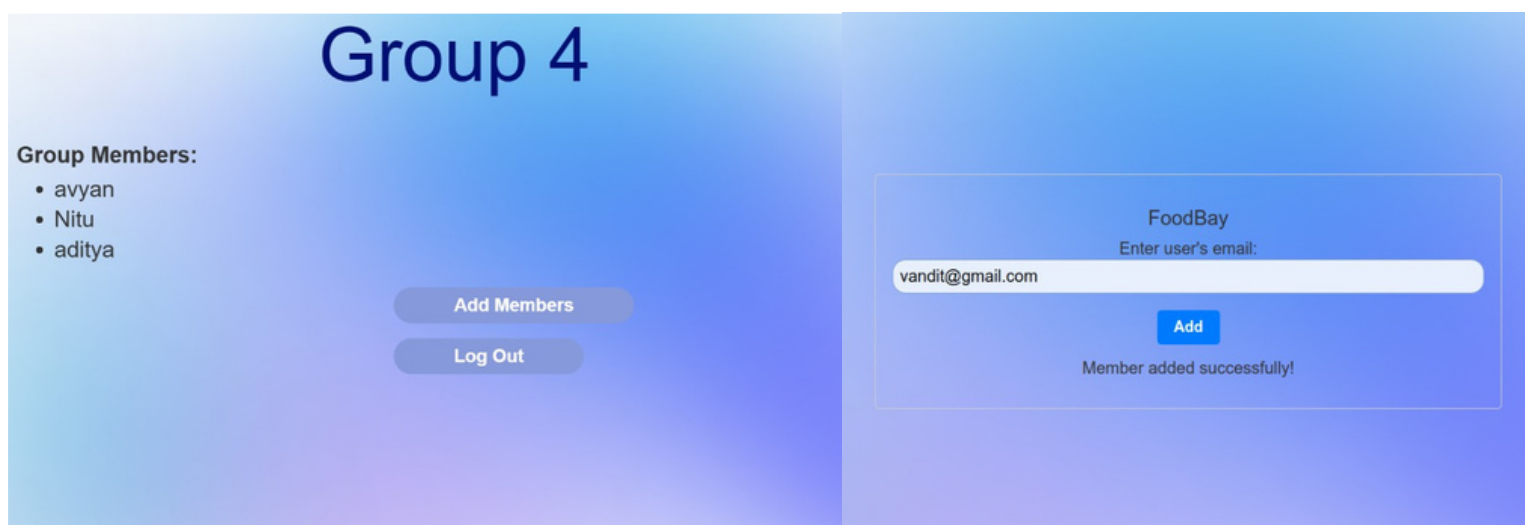
New Group

Done

## Create Groups

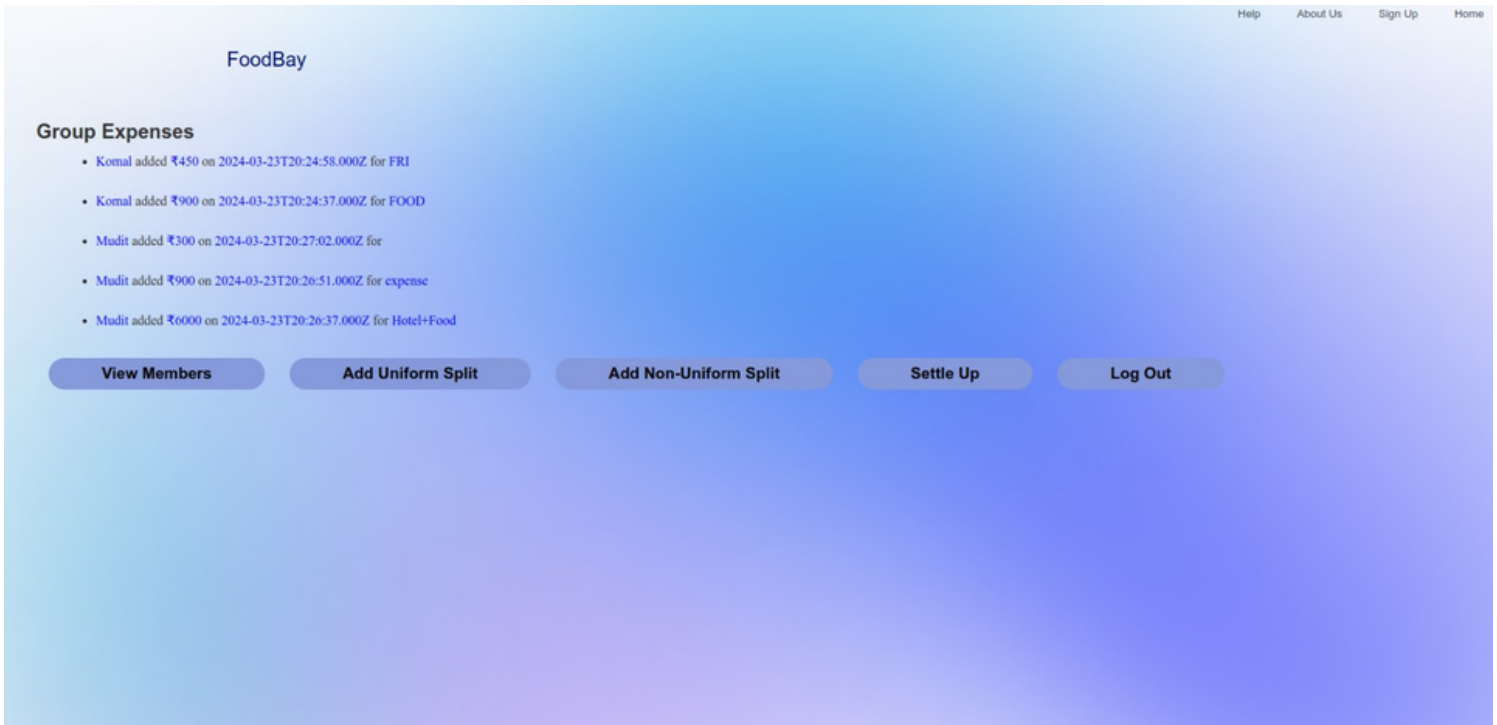


View Groups + Leave + Delete Groups

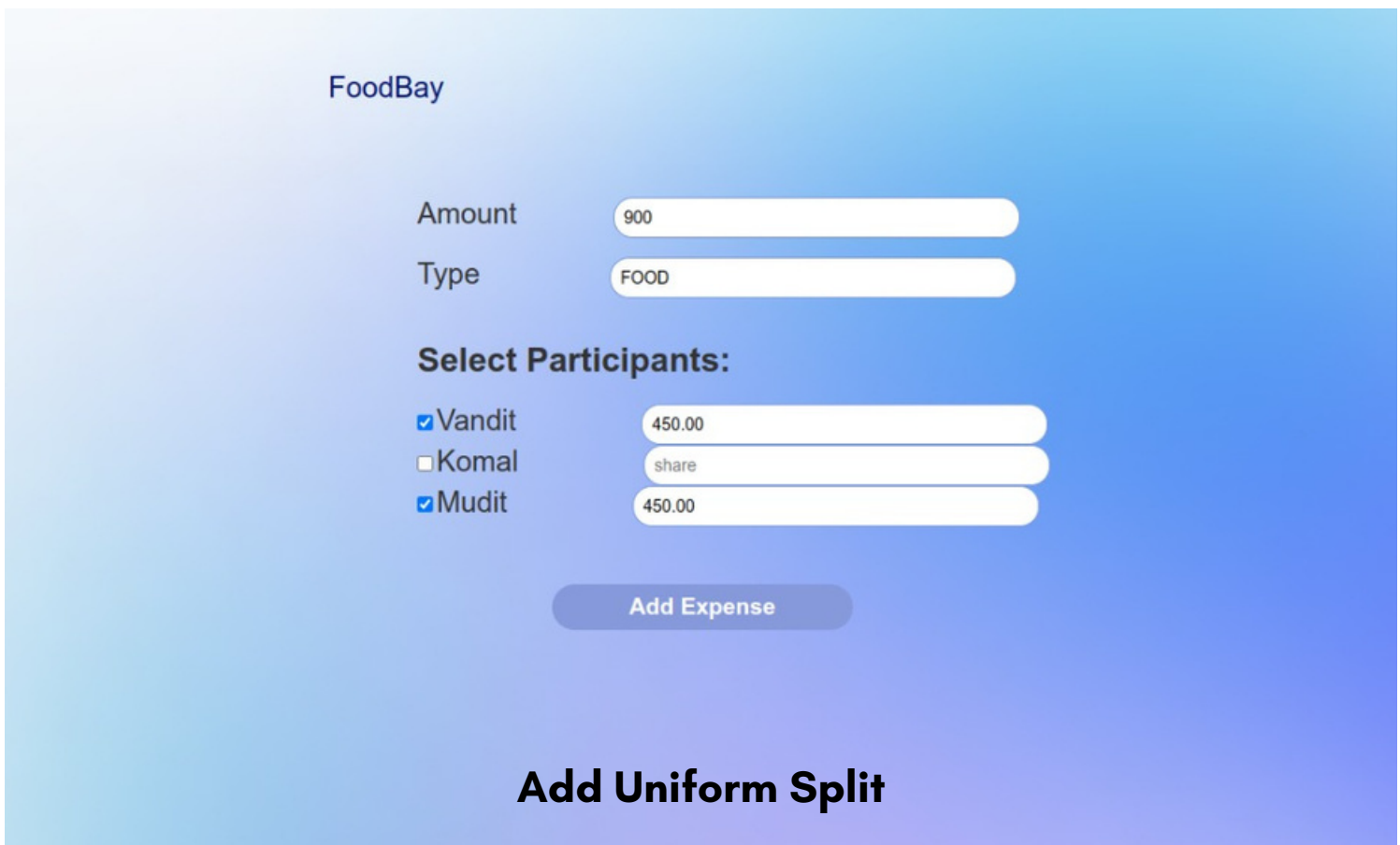


Group Dashboard with List of Members  
+  
Add Members





## Group Expenses Listed



- The amount when added is automatically divided amongst the participants chosen

FoodBay

Amount

Type

**Select Participants:**

Vandit

Komal

Mudit

[Add Expense](#)

**Add Non-Uniform Split**

- The amount can be manually entered against the chosen participants.

FoodBay [Help](#) [About Us](#)

**Settled Expenses:**

You owe ₹2900 to 273499

[Pay](#)

You owe ₹2750 to 357188

[Pay](#)

[Log Out](#)

Settle Up expenses  
+  
Record Payment

# LEARNINGS IMPLEMENTED FROM DESI ASCEND EDUCARE

We have developed an optimised **Bill-splitting and Minimum Transaction algorithm** using the of DSA and Graph optimisation techniques.

We applied OOPs principles in our design. The evaluation of expense is abstracted from the User.

We have maintained separate servers for the frontend and backend of the application to facilitate easy debugging.

Keeping the Design and Architecture in mind we have tried to hit the database as less as possible and hence we have just **one function** each for adding Expense (both uniform and non-uniform) and for settling expenses.

We used JSON which is eventually a hash-map.

We have used the concepts learnt in the DBMS course to **normalise the tables, add triggers and maintain proper associations** between tables and their attributes.

The knowledge we gained about Git and GitHub in our Understanding Systems course came in very handy while collaborating with the group to build this project.



# FUTURE SCOPE

In the future, we plan to enhance the user interface so that it can be more visually appealing.

We also plan to integrate a "Payment Gateway" for secure payments through the application.

Currently our website has the bill splitting feature, but we would like to extend that to expense categorization for expense tracking and finding out where the most amount of money of the user is spent.

Currently, our website supports only signup and authentication through email and password. However, we intend to integrate Google authentication and implement a "Forgot Password" feature to enhance user experience and security.

# CONTRIBUTION

**Frontend:** Vaani Goyal, Praneetha Kancharla, Tanishka Misra

**Backend:** Komal, Sanjana Vajrapu

**Database Model:** Komal, Sanjana Vajrapu

**Connection of the database:** Komal

**Minimum Transaction Algorithm:** Tanishka Misra, Sanjana Vajrapu, Komal

**Integration of frontend and backend:** Vaani Goyal, Tanishka Misra

**API Testing and Bug Fixing:** Komal

**Report:** Praneetha Kancharla, Sanjana Vajrapu, Komal, Tanishka Misra, Vaani Goyal

# REFERENCES

**CSS :** <https://www.w3schools.com>

**UI designs:** <https://www.figma.com>

**General doubts:**

<https://stackoverflow.com/>

<https://www.geeksforgeeks.org/>

**Sequelize:** <https://sequelize.org/>

**Bill splitting Optimisation:**

<https://leetcode.com/discuss/study-guide/2154270/low-level-design-of-splitwise>

**ReactJs & NodeJs:** <https://www.w3schools.com>