



# Expense Splitter

## Seamless Shared Expense Management

Simplifying finances for groups, trips, and events.

# What is Expense Splitter?

The **Expense Splitter** is a web application designed to help groups effortlessly manage shared expenses. Whether it's for trips, roommates, or organizing events, our app clarifies who owes whom and by how much, removing the hassle from shared finances.



## Equal Splits

Divide costs evenly among all participants for quick settlements.



## Custom Splits

Define precise individual contributions for flexible expense allocation.

# Core Features at a Glance



## Group Management

Create and invite friends to new groups or join existing ones.



## Expense Summary

Visualize detailed balances so everyone knows their exact financial standing.



## Expense Management

Easily add new expenses, choosing between equal or custom contributions.



## Secure Authentication

Protected with JWT for secure login and registration processes.

# The Power Behind the App: Our Tech Stack

## Frontend (React)

- **React 19 + React Router DOM:** Modern UI development and navigation.
- **Material-UI (MUI) + React Icons:** Beautiful, responsive components and diverse iconography.
- **Axios:** Efficient API communication.
- **React Toastify:** User-friendly notifications.
- **React Context API:** Streamlined state management.

## Backend (Django + SQLite)

- **Django REST Framework:** Robust API development.
- **JWT Authentication (SimpleJWT):** Secure and scalable user authentication.
- **SQLite:** Lightweight and reliable database solution.
- 
- **REST API:** Comprehensive endpoints for Users, Groups, Expenses, and Summaries.

# Getting Started: Project Setup



## Clone the Repository

Get started by cloning the project from GitHub and navigating into the directory.

```
git clone  
https://github.com/Manu577228/expense-splitter-fractal.git  
cd expense-splitter-fractal
```



## Backend Setup (Django)

Set up your Django backend, install dependencies, and run migrations to get the server ready.

```
cd backend  
python -m venv venv  
source venv/bin/activate  
pip install -r requirements.txt  
python manage.py migrate  
python manage.py runserver
```

Backend runs at: <http://127.0.0.1:8000/>



## Frontend Setup (React)

Configure the React frontend by installing Node.js dependencies and starting the development server.

```
cd frontend  
npm install  
npm start
```

Frontend runs at: <http://localhost:3000/>

# Secure by Design: Authentication Flow

The Expense Splitter prioritizes data security. Our system leverages JWT (JSON Web Tokens) for robust and efficient authentication, ensuring user data is protected.

Register User	/api/users/register/	No	Create a new user account.
Login (JWT Token)	/api/users/token/	No	Obtain access and refresh tokens.
Refresh Token	/api/users/token/refresh/	No	Renew expired access tokens.
Get Profile	/api/users/profile/	Yes	Retrieve authenticated user's profile.

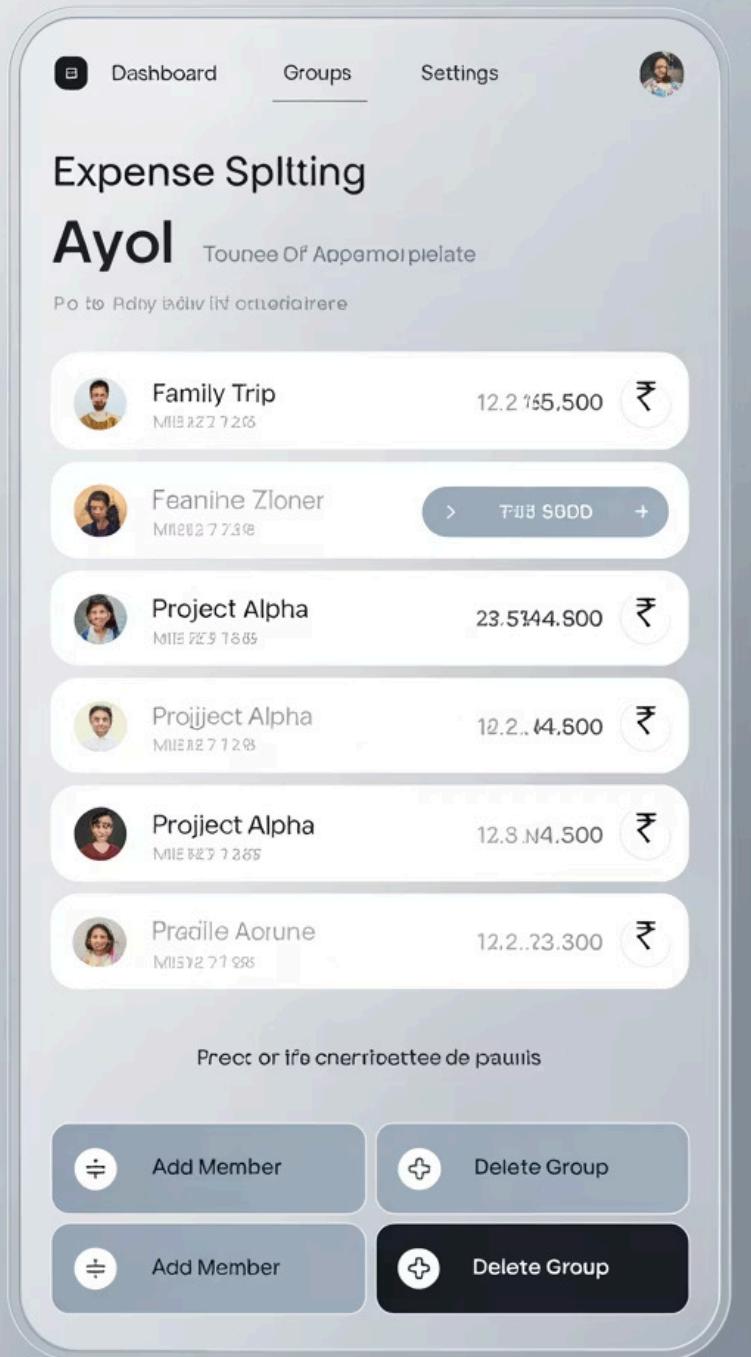
For authenticated requests, include Authorization: Bearer <access\_token> in your headers.

# Comprehensive API Endpoints

Our robust REST API provides all the necessary endpoints to manage users, groups, and expenses, enabling full functionality for the Expense Splitter application.

GET	/groups/	Retrieve all groups the user is part of.	Yes
POST	/groups/	Create a new group.	Yes
GET	/groups/{id}/	Get details for a specific group.	Yes
POST	/groups/{id}/add-member/	Add a new member to an existing group.	Yes
GET	/groups/{id}/expenses/	List all expenses for a specific group.	Yes
POST	/groups/{id}/add-expense/	Add a new expense to a group with equal or custom splits.	Yes
GET	/groups/{id}/summary/	Get the comprehensive expense summary for a group.	Yes

# User Experience Highlights



## Intuitive Dashboard

View all your active groups at a glance, making navigation simple and efficient.



## Flexible Expense Form

Easily add new expenses with options for both equal and custom splitting.



## Detailed Expense List

Track every transaction with clear, group-wise expense visibility.



## Clear Summary Page

Visualize who owes whom with a clear and concise breakdown of balances.

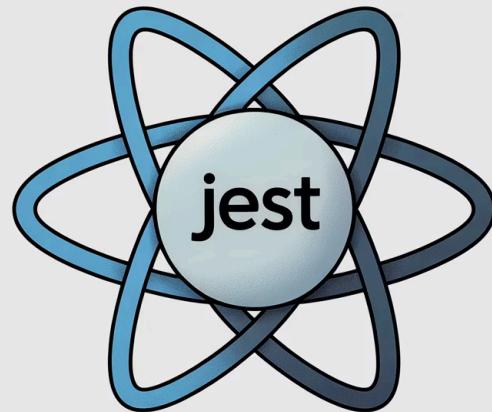
# Ensuring Quality: Rigorous Testing

Our commitment to a reliable and bug-free application is backed by comprehensive testing across both the frontend and backend.

## Frontend (React + Jest)

We've implemented a robust suite of unit and integration tests using Jest to ensure the UI behaves as expected and components interact seamlessly.

```
cd frontend  
npm test
```



## Backend (Django + TestCase)

Django TestCases cover serializers, views, and models, guaranteeing data integrity and API reliability.

```
cd backend  
source venv/bin/activate  
python manage.py test
```



# Author

## Manu Bharadwaj BN

The Authentic JS/Java/Python Codebuff!

[GitHub](#)

[Portfolio](#)

@Code-with-Bharadwaj: YT