Community-Based Monthly Expense Tracker

1. Project Overview

This project is a full-stack web application that allows users to create and manage group-based monthly contributions. Users can form groups, track payments, manage monthly cycles, and view reports. It is built using Django REST Framework for the backend and React with TailwindCSS for the frontend.

2. Key Features

- User Registration and Authentication

- Group Creation and Member Management

- Monthly Contribution Cycles

- Contribution Tracking and Payment Status

- Dashboard and Report Views

- Dockerized Deployment and CI/CD Pipeline with GitHub Actions

3. Technology Stack

- Backend: Django, Django REST Framework

- Frontend: React, Tailwind CSS

- Database: PostgreSQL (or SQLite for development)

- Deployment: Docker, GitHub Actions, Vercel/Render

4. Database Schema

The following are the main models:

- User: username, email, password, profile_pic, first_name, last_name

- Group: name, description, created_by, members, created_at

- ContributionCycle: group, month, year, start_date, end_date, status

- Contribution: user, cycle, amount, status, contributed_on

5. REST API Endpoints

- POST /api/register/
- POST /api/token/
- GET/POST /api/groups/
- POST /api/groups/{id}/add-member/
- POST /api/cycles/start/
- GET/POST /api/contributions/
- GET /api/reports/group/{id}/

6. Frontend Pages

- Login/Register
- Dashboard (List of Groups)
- Group Details (Contribution Status)
- Create Group and Invite Members
- Report Page (Charts and Cycle Summary)

7. Auth Flow

The user logs in using the Django REST Framework JWT endpoint. A token is returned and stored in the frontend (localStorage). The token is attached to all protected API requests.

8. Deployment & DevOps

- Backend and database dockerized using Docker Compose
- Frontend deployed via Vercel or Netlify
- GitHub Actions used to automate testing, linting, and deployment

9. Screenshots (To Be Added)

Include UI screenshots for the following:

- Login Page
- Dashboard

- Group Page
- Contribution Report

####Issues Faced during the project

17.may:

1. Failled multiple times when creating a custom usermodel & CustomUsermanager.
2. Forgot to register model in admin file, forgot to add add apss to installed apps in seetings.py
3. Learnt that password Field should not be add as required fields as it is already handled & may cause conflict
4. Learnt to add AUTH_USER_MODEL for custom model