Project 3: VaultBooks – Multi-Account Personal & Business Expense Tracker with Team Roles

💰 Project Description

VaultBooks is a secure and collaborative MERN stack-based financial tracker. It enables users to manage multiple financial accounts (wallets), track income and expenses, set category-wise budgets, invite team members, and generate detailed financial reports.

Project Requirements

User Roles:

- **Owner** – Creates and manages wallets, full access to all modules
- **Accountant** – Can add/edit transactions and view reports
- **Viewer** – Read-only access to specific accounts

Functional Requirements:

- Multi-account (wallet) finance management
- Categorized transaction tracking
- Monthly budget and overspend alerts
- Recurring transactions and payment reminders
- Role-based wallet sharing and audit logs
- Expense and income reports with chart visualization
- Export reports in PDF/Excel

🗃️ Database Schema Overview

Collections:

- `User`: id, name, email, password, role
- `Wallet`: id, name, type (personal/business), createdBy, members[]
- `Transaction`: id, walletId, type (income/expense/transfer), category, amount, note, tags[], fileUrl, date
- `Budget`: id, walletId, category, limit, spent

- `Reminder`: id, walletId, description, dueDate, frequency (monthly, weekly), status
- `AuditLog`: id, userId, walletId, action, timestamp

□ Module-wise Breakdown

1. Authentication & Access Control

- JWT-based auth for secure login
- Invite users to wallets with roles
- Role-based permissions and navigation

2. Wallet & Transaction Management

- Create/edit/delete wallets (by Owner only)
- Add/edit transactions with:
    - Category, amount, type, description, date, optional file upload
- Filter by wallet, date, category, tags

3. Budgeting System

- Set monthly category limits per wallet
- Track current spend and compare vs. budget
- Progress bars and warning alerts

4. Recurring Transactions & Reminders

- Mark transactions as recurring (e.g., rent)
- Set reminder alerts for upcoming payments
- Cron job simulation for triggering reminders

5. Role Sharing & Audit Logs

- Invite accountants/viewers with scoped access
- Audit logs show who modified what and when

6. Reports & Analytics

- Charts for:

    - Income vs Expense trends
    - Category-wise breakdown
    - Wallet performance
- Downloadable PDF/Excel financial summaries

🗓️ Weekly Task Breakdown

Week 1: Project Setup & Core Modules

- [Dev 1] Setup React + Express + MongoDB project
    - Auth APIs: register, login, role assignment, JWT middleware
    - Create User and Wallet models, implement wallet creation UI
- [Dev 2] Transaction CRUD
    - Add/edit/delete transaction form with file upload and tag picker
    - Filters for date, category, and wallet on list view
- [Dev 3] Budget planning system
    - Budget model and category assignment form
    - Progress bar UI and backend logic for tracking spend

Week 2: Recurring & Reminder System

- [Dev 1] Implement reminder system
    - Due date setup, frequency selector (weekly/monthly)
    - Cron simulation for alert trigger
- [Dev 2] Recurring transaction logic
    - Auto-generate next cycle entry based on frequency
- [Dev 3] Reminder notification system + alert UI
    - In-app alert box + status update mechanism

Week 3: Team Collaboration & Audit Logs

- [Dev 1] Role-based wallet access
    - Invite via email, accept invite simulation, set role scope
- [Dev 2] Implement audit trail
    - Log actions (create/update/delete) with timestamp and user info
- [Dev 3] Audit log dashboard for admins
    - View/filter logs by user/wallet/action/date

Week 4: Reports, Export & Deployment

- [Dev 1] Income vs expense charts using Chart.js
- [Dev 2] PDF and Excel export utilities
    - Filters by month/wallet/category
- [Dev 3] Final QA, bug fixing, deployment
    - Host frontend (Vercel), backend (Render), DB (MongoDB Atlas)
    - Write setup guide and record demo walkthrough

▯ Wireframe Overview (Simple Mockup Diagram)

Wallet Dashboard:

- Summary cards: Balance, Income, Expenses, Budget
- Buttons: Add Transaction, Invite User, Set Budget

Transaction View:

- Table view with filters (date, tag, category, amount)
- Recurring status icon + edit/delete buttons

Budget View:

- Monthly category list
- Progress bar + % remaining alert

Reports View:

- Pie chart: Expense by Category
- Line chart: Income vs Expense trends
- Download buttons for PDF & Excel

This project teaches real-world financial app design and includes modular data modeling, filters, team collaboration, analytics, exports, and notification systems.