Expense Tracker System - Description and Flow

Overview

An Expense Tracker is an application that enables users to input their income and expenses, categorize them, and view summaries and reports over various time periods. It helps users manage personal or business finances efficiently.

Key Features

- **User Authentication**
    - User registration, login, and logout.
    - Password recovery.
- **Expense & Income Management**
    - Add new expenses/income with details: amount, category, date, description.
    - Edit or delete existing entries.
- **Categorization**
    - Predefined categories (e.g., Food, Rent, Salary, Utilities).
    - Option to create custom categories.
- **Dashboard & Summary**
    - Overview of current month/year spending.
    - Total income vs total expenses.
    - Visual charts (pie, bar graphs) for spending by category.
- **Reports**
    - Filter expenses/income by date range, category, or keyword.
    - Export reports (CSV, PDF).
- **Budgeting**
    - Set monthly budgets for categories.
    - Alerts when nearing or exceeding budgets.
- **Bill Reminders & Payment Tracking**
    - Remind users of upcoming bills or due payments.
    - Mark bills paid.

Key Components:

1. **Bill Entry & Management**
    - Users can add bills to the system with details such as:

        - Bill name (e.g., Electricity, Internet, Rent)
        - Amount due
        - Due date
        - Payment frequency (one-time, monthly, quarterly, yearly)
        - Optional notes or reminders
    - Bills can be edited or deleted as needed.
2. **Reminder Scheduling**
    - The system schedules reminders to alert users before the bill's due date.
    - Users can customize reminder timing (e.g., 3 days before, 1 day before).
    - Multiple reminders can be set for a single bill.
    - Reminders can be delivered via: Email alerts
3. **Payment Tracking**
    - Users mark bills as **paid** once payment is made.
    - The system records payment dates and marks the bill accordingly.
    - Payment history is stored for future reference.
    - For recurring bills, marking one as paid automatically schedules the next bill based on payment frequency.
4. **Dashboard Integration**
    - Upcoming bills are displayed prominently on the dashboard.
    - Bills due soon or overdue are highlighted for quick attention.
    - Users see summary stats:
        - Total bills due this month
        - Amount paid vs unpaid
        - Overdue bills
5. **Notifications & Alerts**
    - Timely reminders help avoid missed payments.
    - Alert users if a bill is overdue.
6. **Reporting**
    - Users can generate reports showing bill payments over time.
    - Helps analyze regular expenses and budget planning.
7. **Export & Import**

- Import transactions from CSV (consider bank statements).
- Export data to Excel and CSV for external use.

User Flow for Bill Reminders & Payment Tracking

[User adds a bill]
|
[System stores bill with due date and frequency]
|
[System schedules reminders]
|
[Reminder sent to user (notification/email)]
|
[User pays bill and marks as paid]
|
[System records payment and updates status]
|
[For recurring bills, system schedules next bill automatically]
|
[Dashboard updates upcoming and paid bills]

Database Tables:

Table	Key Fields	Notes
Bills	bill_id, user_id, name, amount, due_date, frequency, notes	Stores bill info and schedule details
Payments	payment_id, bill_id, payment_date, amount_paid	Tracks payments made
Reminders	reminder_id, bill_id, reminder_date, sent_status	Tracks reminders and notification status
User	userId, name, email, passwordHash	Stores user info
Transaction	TransactionId, userId, amount, type, categoryId, date, description	Stores transaction info
Category	CategoryId, userId, name, type	Stored categories like Expense/Income/Both
Budget	BudgetId, userId, categoryId, amount, period	Stored Budget like Monthly

System Flow

A. User Authentication Flow

1. **User Registration**
    - User inputs details: name, email, password.
    - System validates and stores user data securely.
2. **User Login**
    - User inputs email and password.
    - System verifies credentials.
    - On success, the user is taken to the dashboard.
3. **Password Recovery**
    - User requests password reset.
    - The system sends reset links via email.

B. Expense/Income Entry Flow

1. User selects **Add New Transaction**.
2. User enters:
    - Amount
    - Type: Expense or Income
    - Category (select or create)
    - Date (default: current date)
    - Description (optional)
3. User submits a form.
4. The system validates and saves entry.
5. Dashboard updates to reflect new data.

C. Viewing and Managing Transactions Flow

1. User views list of transactions.
2. Users use filters (date range, category, type).
3. User selects a transaction to **Edit** or **Delete**.
4. Changes are saved or transactions removed.
5. Dashboard updates accordingly.

D. Dashboard & Reporting Flow

1. User opens the dashboard.

2. System fetches transaction data for the selected time frame.
3. System calculates totals (income, expenses, balance).
4. System generates charts:
    - Expense distribution by category.
    - Income vs Expense trends.
5. User can export reports or set budget alerts.

Example User Journey

- User registers and logs in.
- User adds an expense of $50 categorized as "Food".
- User adds a salary income of $2000.
- User views the dashboard showing income, expenses, and balance.
- User sets a $300 monthly budget for "Food".
- User receives an alert when total "Food" expenses reach $280.
- User exports a report for the last 3 months.

Technology Stack

- Backend: ASP.NET Core 8 (MVC)
- Frontend: Razor Views (MVC)
- Database: PostgreSQL (Multitenant schema)
- ORM: Entity Framework Core
- Authentication: JWT + Cookie Authentication
- WebAPI