# Finance Tracker Documentation

## 1. Overview

Finance Tracker is designed to help individuals take control of their finances through an intuitive dashboard. Core capabilities include:  
- User Authentication (Secure sign-up, login, and logout).  
- Transaction Management (Add, list, edit, and delete income and expense transactions).  
- Goal Setting (Define financial goals with target amounts).  
- Progress Visualization (View progress bars indicating percentage completion of each goal).  
- Summary Metrics (See total income, total expenses, and net savings at a glance).

## 2. Key Features

### 2.1 User Accounts

• Registration: New users can sign up with email and password.  
• Authentication: Secure login and logout via Django’s built-in auth system.  
• Profile: Display user’s name and basic details on the dashboard.

### 2.2 Transactions

• Add Transaction: Users specify type (Income or Expense), amount, date, and description.  
• Transaction List: Paginated view   
• Edit & Delete: Modify or remove transactions.

### 2.3 Goals & Progress

• Set Goal: Define a goal title, target amount, and optional deadline.  
• Progress Calculation: Computed based on net savings vs. goal amounts.  
• Visual Display: Progress bars for each goal.

### 2.4 Dashboard Metrics

• Total Income: Sum of all income transactions.  
• Total Expense: Sum of all expense transactions.  
• Net Savings: Income minus expenses.  
• Goal Progress: Sequence-based allocation of savings to goals.

## 3. System Architecture

The application uses Django for backend logic and templating, Tailwind CSS for styling, and SQLite/PostgreSQL for data storage. Key layers include:  
• Frontend: HTML templates with Tailwind CSS  
• Backend: Django views, models, and forms  
• Database: ORM-managed SQLite

## 4. Data Models

### 4.1 User

Fields inherited from django.contrib.auth.models.User.

### 4.2 TransactionModel

| Field | Type | Description |  
|-------------------|--------------------|------------------------------------|  
| user | ForeignKey(User) | Owner of the transaction |  
| transaction\_type | CharField | 'Income' or 'Expense' |  
| amount | DecimalField | Monetary value |  
| date | DateField | Transaction date |  
| description | TextField (opt.) | Optional note |

### 4.3 GoalModel

| Field | Type | Description |  
|---------------|-------------------|-----------------------------------|  
| user | ForeignKey(User) | Owner of the goal |  
| title | CharField | Short name of the goal |  
| amount | DecimalField | Target savings amount |  
| created\_at | DateTimeField | Timestamp when created |  
| deadline | DateField (opt.) | Optional target date |

## 5. Installation

1. Clone the repository:  
 git clone https://github.com/MSHAHID1122/Django  
 cd Finance  
2. Create & activate virtualenv:  
 python3 -m venv venv  
 source venv/bin/activate  
3. Install dependencies:  
 pip install -r requirements.txt  
4. Run migrations:  
 python manage.py migrate  
5. Start development server:  
 python manage.py runserver  
6. Open http://127.0.0.1:8000/ in your browser.

## 6. Usage Guide

• Register & Login: Use the Sign Up and Login pages.  
• Transactions: Add, view, edit, and delete.  
• Goals: Create new goals and view progress.  
• Dashboard: Monitor metrics and goal progress.

## 7. Future Enhancements

- Integrate charts with Chart.js  
- Export data to PDF/CSV  
- Notifications for deadlines  
- Multi-currency support  
- Mobile companion app

## 8. Contribution & Support

• GitHub Issues: Report bugs or request features.  
• Pull Requests: Fork and submit PRs.  
• Community: Join our Slack or Discord.