# **📘 User Guide – Personal Budget Tracker**

**Version:** 1.1

**Last Updated:** June 2025

## **1. Overview**

The **Personal Budget Tracker** is a full-stack **MERN** web application (MongoDB, Express.js, React.js or EJS, Node.js) built for managing personal finances. Users can securely log in, add or edit income and expense records, track financial performance with real-time charts, and now benefit from **recurring transactions** and **pinned quick-access entries**. The system is responsive, user-friendly, and optimized for long-term personal budgeting.

## **2. Key Features**

### **🔐 Secure User Authentication (JWT)**

Users log in via encrypted JWT-based sessions. Security mechanisms include token expiry and route protection for all data access.

### **💸 Add, Edit, Delete Transactions**

Users can:

* Add **income** or **expense** entries.
* Include metadata like date, amount, category, and description.
* Use **recurring transactions** (e.g., monthly salary or rent).
* Pin key transactions for **quick reference** on the dashboard.

### **♻️ Recurring Transactions**

* Add income or expenses as **recurring** (e.g., weekly, monthly).
* View a tag on recurring entries in your transaction list.
* The system auto-generates these in the backend periodically (manual for now; cron automation in future).

### **📌 Pinned Transactions**

* Pin essential or frequently referenced transactions (e.g., rent, insurance).
* Pinned items appear **at the top** of the transaction table for convenience.

### **📈 Financial Summary & Charts (Chart.js)**

A visual dashboard shows:

* Income vs. Expense trends over months.
* Category-wise breakdown.
* Summary cards with total income, expenses, and top categories.

### **📅 Filtering and Sorting**

* View transactions by category, month, or year.
* Sort by date or amount.

### **📱 Fully Responsive Design**

The interface adapts to all screen sizes including mobile, tablet, and desktop.

### **🧪 Data Validation & Testing**

* Frontend & backend validations.
* Manual test cases and Postman collection available.
* Secure routes and schema validations enforced.

## **3. System Requirements**

|  |  |
| --- | --- |
| **Component** | **Minimum Requirement** |
| OS | Windows 10+, macOS, Linux |
| Node.js | v14.x or higher |
| MongoDB | MongoDB Atlas (preferred) or Local |
| Browser | Chrome, Firefox, Safari, Edge |
| Tools | Git CLI, VS Code, Postman, MongoDB Compass |

## **4. Installation & Setup**

### **4.1 Clone the Repository**

git clone <https://github.com/YOUR_USERNAME/Budget-Tracker.git>   
cd Budget-Tracker

### **4.2 Backend Setup**

cd Backend   
npm install

Create a .env file in Backend/ and add:

MONGO\_URI=your\_mongodb\_uri  
SESSION\_SECRET=your\_secret  
PORT=5000

Start server:

npm run dev

### **4.3 Frontend Setup**

If using React:

cd ../frontend   
npm install   
npm start

For static EJS version, open files from /views with local server setup.

Access the app at:

👉 <http://localhost:3000>

## **5. Using the Application**

### **5.1 Registration & Login**

Sign up with name, email, and password. JWT tokens manage your session securely.

### **5.2 Adding a Transaction**

* Click "Add Income/Expense"
* Fill in source/category, amount, date
* ✅ **Check "Recurring"** to mark it as repeating
* 📌 **Click "Pin"** to prioritize this record

### **5.3 Viewing & Managing Transactions**

* Transactions are listed chronologically
* Recurring and pinned entries are visually marked
* Edit/Delete options are available for all entries

### **5.4 Dashboard & Charts**

* Bar and line charts show income vs. expenses
* Filters available for month, year, and type
* Summary cards show total, average, and top categories

### **5.5 Recurring Transactions**

* Set frequency (currently monthly by default)
* Entries marked as recurring are automatically reused when cloned manually
* Cron-job support can be integrated in future

### **5.6 Pinned Transactions**

* Pin transactions for always-on-top visibility
* Unpin them anytime
* Useful for critical payments like rent, loan EMI, etc.

## **6. Troubleshooting**

|  |  |
| --- | --- |
| **Issue** | **Solution** |
| Backend won’t start | Check .env, MongoDB URI, and port availability |
| Frontend fails to load | Run npm install and npm start in /frontend |
| Database connection errors | Ensure IP whitelisting on MongoDB Atlas |
| Token expired | Clear localStorage and login again |
| API failing | Use Postman to inspect requests and responses |

## **7. Developer Tools & Tips**

* 🧪 Use Postman to test APIs (/api/v1/income, /api/v1/expense)
* 💻 Git branches for each feature: feature/recurring, feature/pinned
* 📋 Trello for task tracking
* 🔍 Use Chrome DevTools for inspecting UI
* 🧱 MongoDB Compass to monitor records
* 🧪 Unit tests are located under /tests and documented in /docs

## **8. Contributors**

|  |  |
| --- | --- |
| **Name** | **Role & Contributions** |
| 👨‍💻 Ali Haider | DevOps & PM – Repository, CI/CD, deployment |
| 🎨 Pascaline | Frontend UI – Layouts, Tailwind CSS, interactivity |
| 🧠 Anjitha Varghese | DB – MongoDB schema, validation, ERD |
| 🛠 Duo Zhang | Backend APIs – Auth, transactions |
| ✅ Anushri Jawahar | QA & Docs – Test cases, SRS, recurring/pinned features |