

Smart Money Manager (SmartTracker) — Project Documentation

1. Project Overview

Smart Money Manager is a modern, dynamic web and mobile application designed for flexible expense and data tracking. Unlike rigid expense trackers, this application allows users to build their own "Forms" with custom fields, making it suitable for tracking anything from daily spending to office supplies or gym logs.

Key Features:

- **Dynamic Form Builder:** Users can define their own tracking forms by choosing from various fields (Date, Amount, Category, Vendor, etc.).
 - **Data Management:** Full CRUD (Create, Read, Update, Delete) functionality for both forms and their entries.
 - **Advanced Reporting:**
 - Monthly summaries with numeric totals.
 - Interactive bar charts for monthly overview.
 - One-click exports to **PDF** and **CSV**.
 - **Cross-Platform:** Built for Web and Mobile (Android/iOS) using React and Capacitor.
 - **Premium UI:** Glassmorphism design with soft gradients and responsive sidebar navigation.
-

2. Technical Stack

- **Frontend:** React (v19), Axios for API communication.
 - **Styling:** Vanilla CSS with Modern CSS Variables (Custom properties).
 - **Backend:** Node.js with Express framework.
 - **Database:** PostgreSQL with Prisma ORM for type-safe data access.
 - **Mobile:** Capacitor (Cross-platform bridge).
 - **Notifications:** React Hot Toast for real-time feedback.
 - **Exporting:** jsPDF and autoTable for PDF generation.
-

3. Database Schema (PostgreSQL)

The database is managed via Prisma. Below are the core models:

Form Model

Stores the definition of a user-created tracking form.

- `id` (Int): Primary Key.
- `name` (String): Title of the form.
- `fields` (String): JSON string storing the configuration of active fields.

FormEntry Model

Stores the actual data submitted via a Form.

- `id` (Int): Primary Key.
- `formId` (Int): Foreign Key referencing `Form`.
- `data` (String): JSON string containing value mapping for custom fields.
- `date` (String): Date of the record.

4. API Documentation

Forms API

Method	Endpoint	Description
GET	<code>/api/forms</code>	Retrieves all user-created forms.
POST	<code>/api/forms</code>	Creates a new custom form.
PUT	<code>/api/forms/:id</code>	Updates form name or field configuration.
DELETE	<code>/api/forms/:id</code>	Removes a form and all its associated data.

Entries API

Method	Endpoint	Description
GET	<code>/api/forms/:formId/entries</code>	Fetch all records for a specific form.
POST	<code>/api/forms/:formId/entries</code>	Add a new record (entry) to a form.
PUT	<code>/api/entries/:id</code>	Update an existing record's values.
DELETE	<code>/api/entries/:id</code>	Remove a specific record.

5. Folder Structure

- `src/components/` : Modular React components (FormsList, FormView, ReportsPage).
- `src/App.js` : Main entry point with routing and global state.
- `src/App.css` : Core design system and layout styling.
- `server/index.js` : Express server implementation and API routing.
- `prisma/` : Database schema and migration files.
- `android/` : Native Android project generated by Capacitor.

6. Setup & Installation

Web Development

1. Install dependencies: `npm install`
2. Start development server: `npm run dev` (Runs both React & Node server).

Mobile Deployment

1. Build the React project: `npm run build`
2. Sync with Capacitor: `npx cap sync`
3. Open in Android Studio: `npx cap open android`

4. Run on a physical device/emulator.

7. Future Enhancements

- User Authentication (Login/Signup).
- Cloud Image Upload for receipts.
- Multi-currency support.
- Push notifications for daily reminders.

Generated for Smart Money Manager Project © 2026