**Receipt Dashboard 📊**

A full-stack receipt management dashboard that allows users to upload receipts (PNG, JPG, JPEG, PDF), extract data using OCR, search/sort/filter receipts, visualize analytics, and export summaries.

**🚀 Features**

* 📄 Upload receipts via image or PDF
* 🧠 Extracts fields (vendor, date, amount, category) using OCR
* 🔍 Smart search & filtering (by vendor, amount, date, category)
* 📊 Visual analytics (pie/bar charts with % breakdowns)
* 🗃️ Export summaries as CSV or JSON
* ✏️ Edit receipt fields directly from UI
* ⚙️ REST API with FastAPI backend

**📁 Project Structure**

receipt-dashboard/

├── backend/

│ ├── main.py # FastAPI application

│ ├── crud.py # DB CRUD logic

│ ├── database.py # SQLAlchemy models & DB config

│ ├── parser/ # OCR and data extractor

│ ├── algorithms/ # Search/sort/aggregates logic

├── frontend/

│ ├── src/

│ │ ├── App.js # Routes and layout

│ │ ├── components/ # Upload, Editor, Charts, etc.

│ ├── public/

├── README.md

**⚙️ Setup Instructions**

**🔧 Backend (FastAPI)**

1. **Create virtual environment**
2. python -m venv env

source env/bin/activate # Windows: env\Scripts\activate

1. **Install dependencies**

pip install -r requirements.txt

1. **Run FastAPI server**

uvicorn main:app --reload

Backend available at: http://localhost:8000

**💻 Frontend (React)**

1. **Navigate to frontend**

cd frontend

1. **Install dependencies**

npm install

1. **Start development server**

npm start

Frontend available at: http://localhost:3000

**🧱 Design Choices**

**Backend**

* **FastAPI** for quick and performant APIs
* **SQLAlchemy** for ORM-based DB operations
* **Modular design**: separate parser, crud, algorithms
* **OCR** via pytesseract (images) and pdf2image (PDFs)

**Frontend**

* **React** with hooks and functional components
* **Bootstrap 5** for responsive UI
* **Chart.js** for data visualization
* **Axios** for API interaction

**📊 User Journeys**

* **Upload Receipt** → Automatically parsed → Stored in DB
* **Search/Filter Receipts** → View specific records
* **Visualize Analytics** → Pie/bar charts for categories/vendors
* **Edit Fields** → Save updates from UI
* **Export Summary** → Download as CSV/JSON

**⚠️ Limitations**

* OCR quality depends on receipt clarity
* Categories are extracted using basic keyword matching
* No user authentication implemented (yet)
* Currently uses local SQLite DB (no multi-user support)

**🧠 Assumptions**

* Input receipts are reasonably clear and structured
* Fields like vendor, amount, and date are extractable
* One user operating the system at a time (no concurrency handling)

**📦 Export APIs**

* GET /export/summary/csv → CSV summary
* GET /export/summary/json → JSON summary

**🙌 Contributions**

Feel free to fork and contribute! PRs and issues welcome.

**📜 License**

MIT License