BILL INSIGHT

BillInsight is a Python-based personal finance tool that analyzes spending by extracting data from uploaded receipts. It categorizes expenses, offers insights via interactive charts and reports, and ensures data accuracy. Built with Streamlit, SQLite, Tesseract OCR, and Pydantic, it emphasizes user control and expense tracking.

TABLE OF CONTENT

1. FEATURES

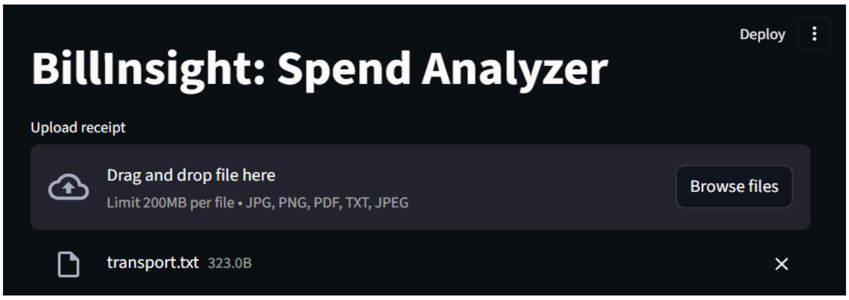
- Upload and analyze receipts in .png, .jpg, .jpeg, .pdf, and .txt formats
- OCR-based text extraction using Tesseract
- Vendor, date, amount, category, and currency parsing
- Manual field editing post-extraction
- Category and currency selection via dropdowns
- Pydantic validation for structure and type safety

- ACID-compliant storage in SQLite
- Interactive records table with full export (.csv, .json)
- Filter/search receipts by vendor or amount range
- Insights: total, mean, median, mode
- Visualizations: vendor frequency, monthly trendline, category distribution
- Monthly expense summary exportable in CSV format

2. FUNCTIONAL WORKFLOW

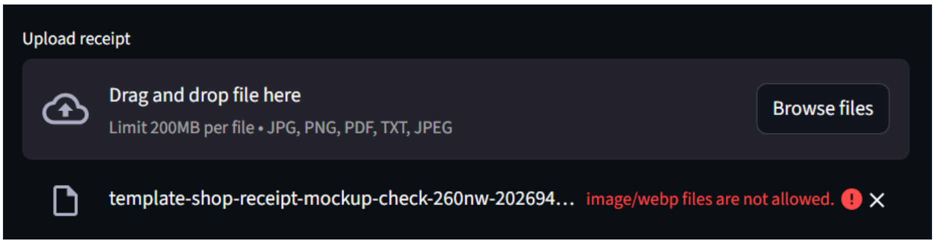
1. Upload Receipts

Users can upload files in supported formats. Unsupported types are rejected with a validation message.



2. OCR & Text Extraction

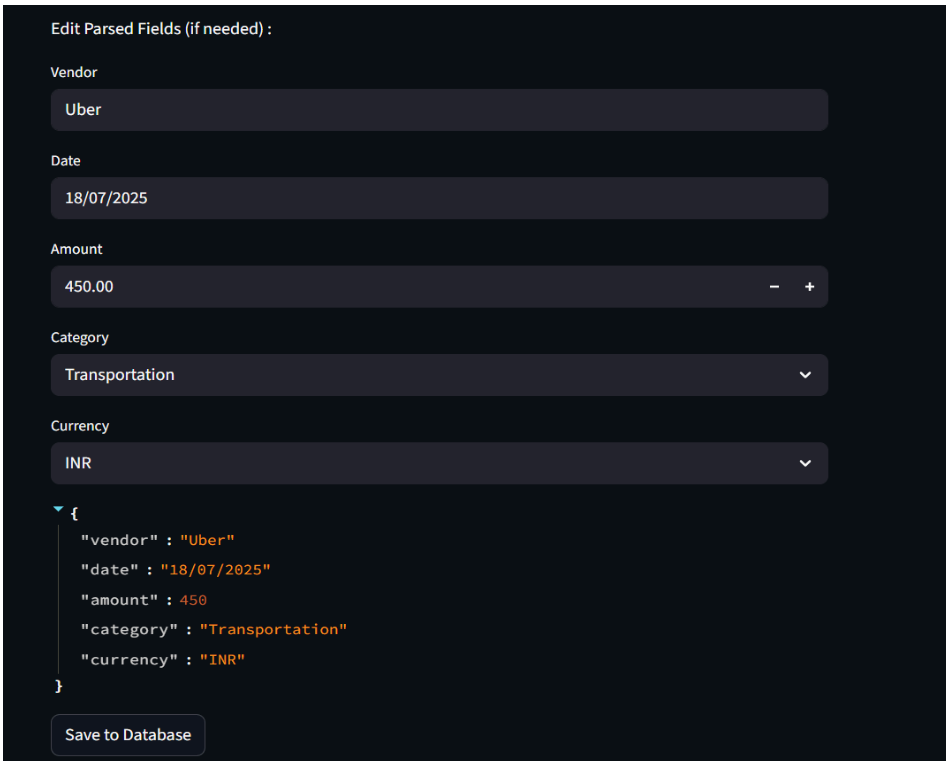
Tesseract reads and extracts text from images and PDFs. Text files are processed directly.


Upload receipt
Drag and drop file here
Limit 200MB per file • JPG, PNG, PDF, TXT, JPEG
Browse files
template-shop-receipt-mockup-check-260nw-202694…

image/webp files are not allowed.

3. Data Parsing

Parsed fields include: Vendor, Date of transaction, Amount, Currency (₹, $, €, £), Category (via regex mapping based on vendor).



Edit Parsed Fields (if needed) :

Vendor: Uber

Date: 18/07/2025

Amount: 450.00

Category: Transportation

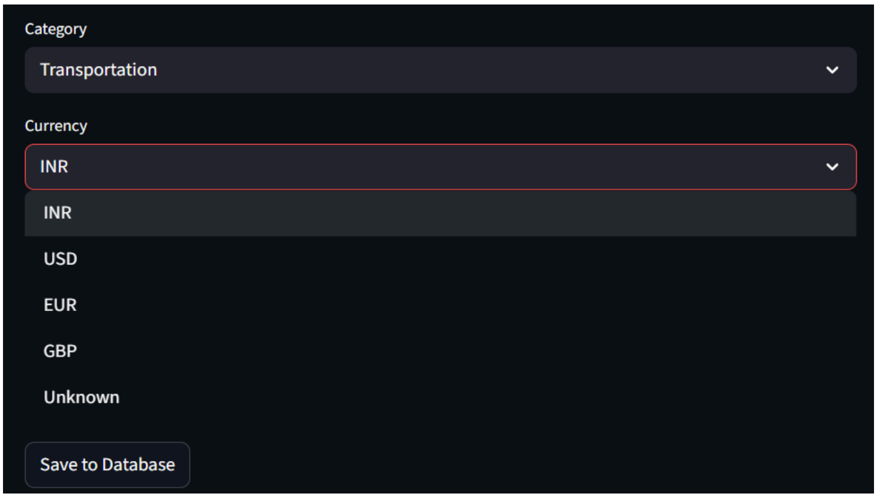
Currency: INR

```
{
  "vendor" : "Uber"
  "date" : "18/07/2025"
  "amount" : 450
  "category" : "Transportation"
  "currency" : "INR"
}
```

4. Manual Field Editing

Users can edit:

- Text fields: vendor, date, amount
- Dropdowns: category, currency

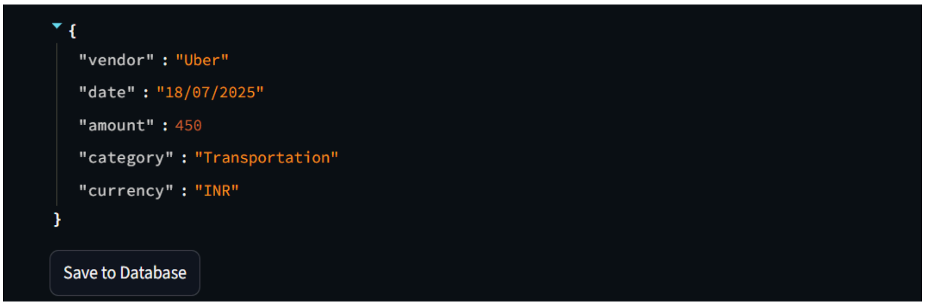


Category: Transportation

Currency: INR

5. Data Validation

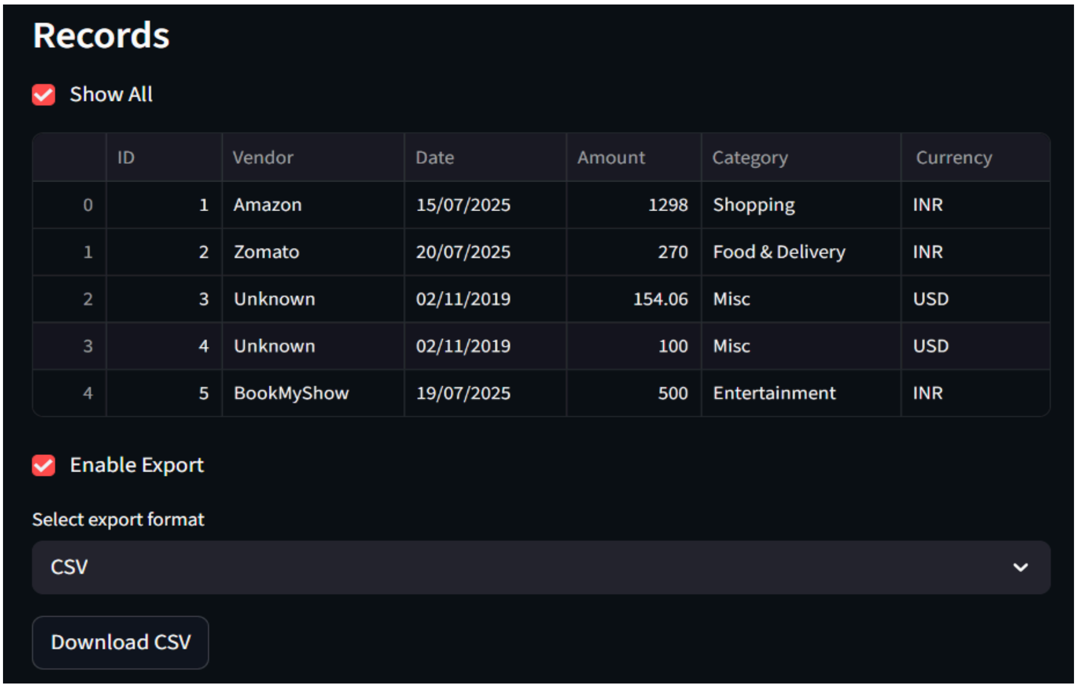
Fields are validated using Pydantic. Incorrect values are caught before saving.



```
{
  "vendor" : "Uber"
  "date" : "18/07/2025"
  "amount" : 450
  "category" : "Transportation"
  "currency" : "INR"
}
```

6. Data Storage

Parsed and validated receipts are stored in a normalized SQLite database ensuring ACID compliance.



Records

☑ Show All

	ID	Vendor	Date	Amount	Category	Currency
0	1	Amazon	15/07/2025	1298	Shopping	INR
1	2	Zomato	20/07/2025	270	Food & Delivery	INR
2	3	Unknown	02/11/2019	154.06	Misc	USD
3	4	Unknown	02/11/2019	100	Misc	USD
4	5	BookMyShow	19/07/2025	500	Entertainment	INR

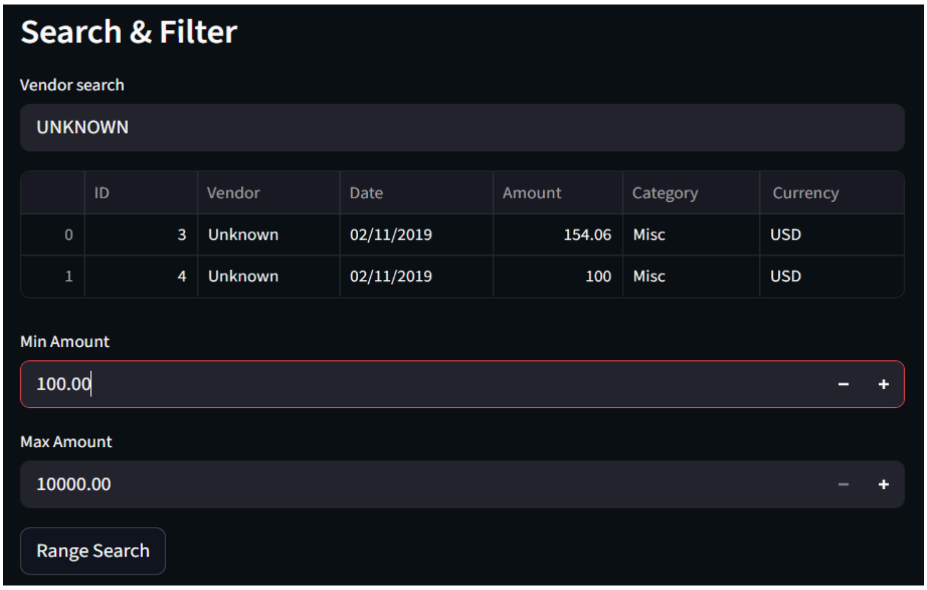
☑ Enable Export

Select export format: CSV

7. Record Management

All records are displayed in a searchable, filterable table. Users can:

- Search by vendor name
- Filter by min-max amount range



Search & Filter

Vendor search

UNKNOWN

	ID	Vendor	Date	Amount	Category	Currency
0	3	Unknown	02/11/2019	154.06	Misc	USD
1	4	Unknown	02/11/2019	100	Misc	USD

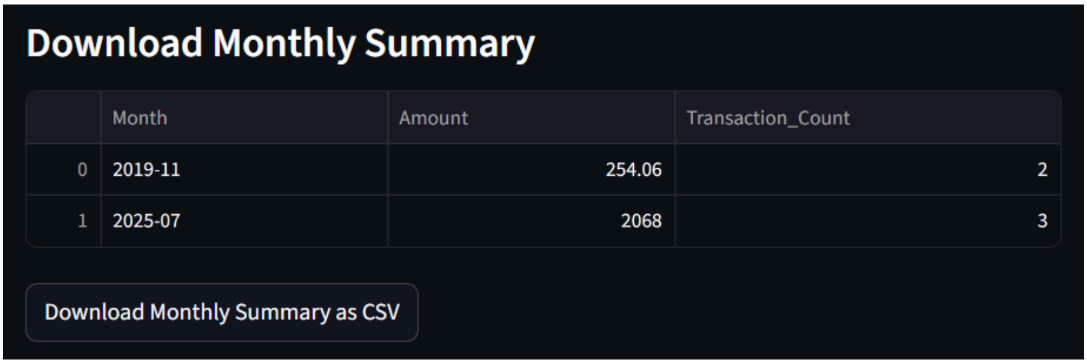
Min Amount

100.00

Max Amount

10000.00

- Export all data or monthly summaries



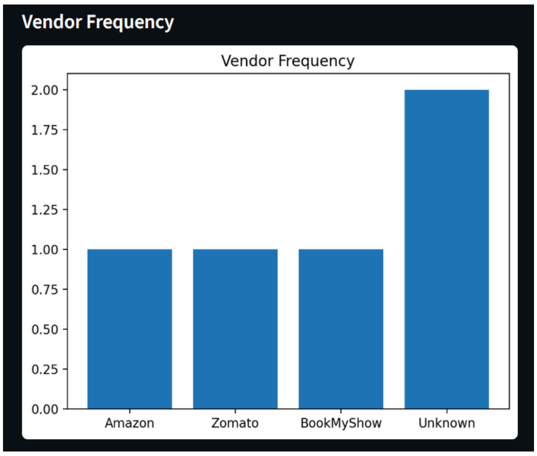
Download Monthly Summary

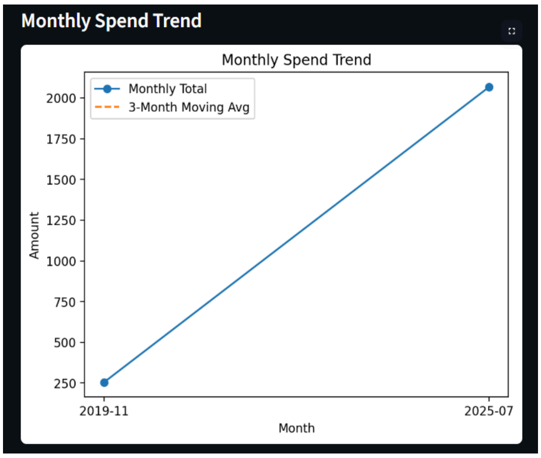
	Month	Amount	Transaction_Count
0	2019-11	254.06	2
1	2025-07	2068	3

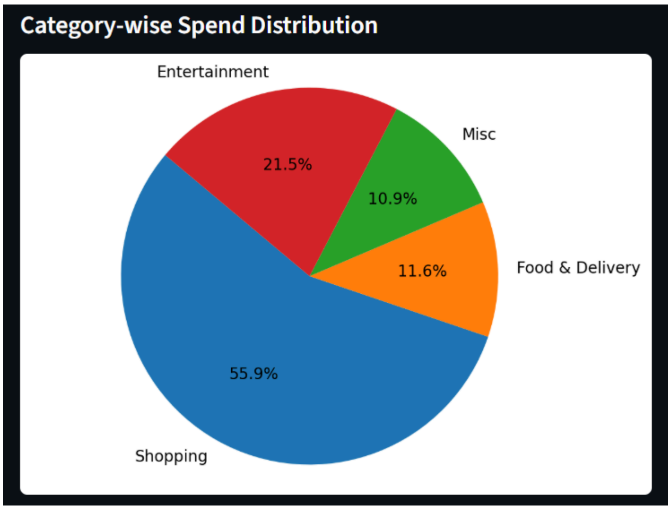
8. Insights and Visualization

Includes:

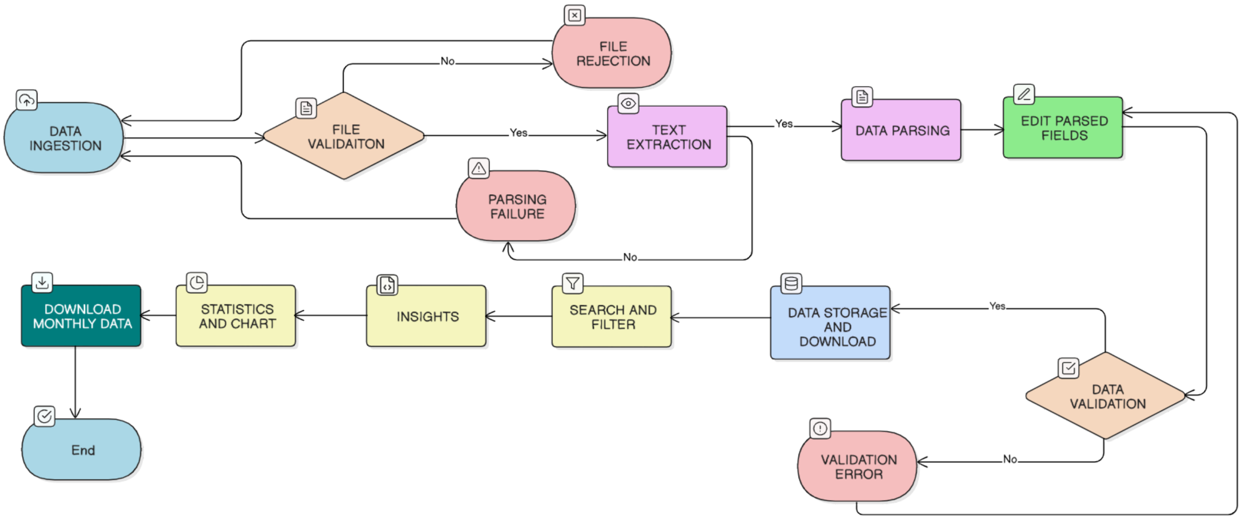
- Aggregate stats: total, mean, median, mode
- Bar chart: vendor frequency
- Line graph: monthly spend trend + moving average
- Pie chart: category-wise distribution







3. SYSTEM FLOWCHART



4. TECHNOLOGY STACK

Component	Technology
Interface	Streamlit
OCR	Tesseract (pytesseract)
Backend Logic	Python
Validation	Pydantic
Data Storage	SQLite
Visualization	Matplotlib, Pandas
File Handling	Streamlit uploader

5. CHALLENGES MADE AND SOLUTION

OCR Accuracy

- **Problem**: Low-quality images impacted text extraction.
- **Solution**: Allowed manual field editing, used robust regex, and added fallback values.

Vendor Classification

- **Problem**: Generic mappings missed Indian vendors.
- **Solution**: Developed a comprehensive regex-based mapping for Indian apps and services.

Date Extraction

- **Problem**: Multiple formats (DD/MM/YYYY, MM-DD-YYYY).
- **Solution**: Used flexible regex patterns and manual edit option.

Validation

- **Problem**: File corruption or unsupported extensions.

- **Solution**: Pydantic schema enforced input types and default fallback values.

Data Quality & Integrity

- **Problem**: Ensuring stable DB behavior with multiple updates.
- **Solution**: Used normalized tables and ensured ACID properties through SQLite.

Visualization Consistency

- **Problem**: Sparse data led to broken charts.
- **Solution**: Handled edge cases gracefully with conditional rendering and moving average.
  .