Feature roadmap

Absolutely! Here’s a **clear roadmap** for your reconciliation platform, covering the MVP and next iterations, structured so you can continue developing in GitHub Copilot. I’ve included both frontend, backend, and feature-level milestones.

**Phase 0: Foundation**

**Goal:** Get basic upload and display working (you already did most of this).

* Single file upload (frontend → backend) ✅
* Display JSON output in table
* Basic styling / table layout

**Phase 1: Dual-file Upload & Basic Reconciliation**

**Goal:** Reconcile two files and show results.  
**Frontend:**

* Two file inputs: Bank file & Ledger file
* Upload button triggers backend reconciliation
* Display results in two tables: reconciled vs unmatched
* Highlight matched items (green) / unmatched items (red)

**Backend:**

* Accept two files in a single API request
* Parse CSV/Excel files
* Run **exact match logic** (date, amount, vendor)
* Return JSON:
* {
* "reconciled": [{...matched entries...}],
* "unmatched": [{...unmatched entries...}]
* }

**Phase 2: Smart Matching & Duplicate Handling**

**Goal:** Handle near-matches and duplicates.  
**Backend:**

* Detect duplicates within a file
* Implement **fuzzy matching** for minor differences in vendor/description
* Tag each entry: matched\_exact, matched\_fuzzy, unmatched
* Return enriched JSON for frontend to use

**Frontend:**

* Color-code:
  + Exact match → green
  + Fuzzy match → yellow
  + Unmatched → red
* Add filtering/sorting by vendor, date, amount
* Optionally allow inline override (mark matched/unmatched manually)

**Phase 3: Persistent Session & User Tracking**

**Goal:** Keep session state, user edits, and historical reconciliation.  
**Backend:**

* Save uploaded files & reconciliation results (DB: SQLite/Postgres/Databricks)
* Support fetching historical reconciliations
* Track user actions (approve/reject matches)

**Frontend:**

* Session history dropdown → select previous reconciliation
* Show which entries were manually corrected

**Phase 4: Dashboard & Analytics**

**Goal:** Turn reconciliations into insights.  
**Features:**

* Summary stats:
  + Total transactions, matched %, unmatched %
  + Amount difference
* Charts:
  + Matched vs unmatched over time
  + Vendor-wise match summary
* Export results (CSV/Excel/PDF)

**Phase 5: AI-assisted Reconciliation (Optional)**

**Goal:** Speed up fuzzy matching & suggestions.

* Use a lightweight LLM or embeddings to **suggest likely matches**
* Frontend shows AI-suggested match, user can approve/reject
* Improve model over time with feedback

**Tech Stack Recommendations**

* **Frontend:** React + Tailwind CSS
* **Backend:** FastAPI (Python)
* **Storage:** Local files → Postgres / Databricks / S3
* **Optional ML/AI:** OpenAI embeddings for fuzzy match suggestions
* **Dev Flow:** GitHub Copilot for autocomplete, modular components for frontend + backend

If you want, I can **also sketch a visual roadmap/wireframe** showing the screens, tables, and flows so you can directly use it as reference while coding. That usually makes GitHub Copilot work much smoother.

Do you want me to do that next?

Visual roadmap

Perfect! Let’s lay out a **visual roadmap + wireframe** for your SME reconciliation app. I’ll describe it step by step so you can visualize screens and flows clearly. Later you can convert this into actual React components.

## ****1️⃣ Landing / Upload Screen****

**Purpose:** Let the user upload the two files.

**Layout:**

+------------------------------------------------+

| SME Reconciliation Dashboard |

+------------------------------------------------+

| [Upload Bank File] [Choose File] |

| [Upload Ledger File] [Choose File] |

| |

| [Reconcile Now] |

+------------------------------------------------+

| Recent Reconciliations |

| - Sep 28, 2025 |

| - Sep 27, 2025 |

+------------------------------------------------+

**Features:**

* Two file inputs (Bank / Ledger)
* “Reconcile Now” button triggers API call
* Shows list of recent reconciliations

## ****2️⃣ Reconciliation Results Screen****

**Purpose:** Show matched/unmatched entries clearly.

**Layout (Table View):**

Reconciled Transactions

+---------+--------+---------+------------+--------+

| Date | Amount | Vendor | Description| Match |

+---------+--------+---------+------------+--------+

| 2025-09-12 | 250 | Vendor B | Office supplies | ❌ |

| 2025-09-07 | 250 | Vendor A | Travel | ❌ |

| 2025-09-14 | 250 | Vendor B | Travel | ❌ |

+---------+--------+---------+------------+--------+

Unmatched Transactions

+---------+--------+---------+------------+--------+

| Date | Amount | Vendor | Description| Match |

+---------+--------+---------+------------+--------+

| ... | ... | ... | ... | ❌ |

+---------+--------+---------+------------+--------+

**Features:**

* Color-coded rows:
  + ✅ green = exact match
  + ⚠️ yellow = fuzzy match
  + ❌ red = unmatched
* Sort/filter by Vendor / Amount / Date
* Inline actions: “Approve Match / Reject Match”

## ****3️⃣ Dashboard / Analytics Screen****

**Purpose:** Provide high-level insights.

**Layout (Card + Chart View):**

+---------------------------+---------------------------+

| Total Transactions: 120 | Matched %: 85% |

+---------------------------+---------------------------+

| Unmatched %: 15% | Total Amount Difference |

+---------------------------+---------------------------+

Charts:

[Pie Chart: Matched vs Unmatched]

[Bar Chart: Vendor-wise Match %]

[Line Chart: Match Over Time]

**Features:**

* Interactive charts
* Export button (CSV / Excel / PDF)

## ****4️⃣ Optional: AI Match Suggestion Modal****

**Purpose:** Suggest possible matches using AI for tricky entries.

**Layout:**

Transaction: 2025-09-12 | Amount: 250 | Vendor: Vendor B

Suggested Match: Ledger Entry 2025-09-11 | Amount: 250 | Vendor: Vendor B

[Approve] [Reject] [Edit Suggestion]

**Features:**

* Highlight suggested match
* User approves/rejects
* Feedback stored to improve model

### **User Flow**

1. User lands → uploads two files → clicks **Reconcile Now**
2. Backend processes → returns reconciled & unmatched JSON
3. Frontend shows tables → user filters / approves / rejects
4. Dashboard shows stats & charts
5. Optional AI suggests possible matches

If you want, I can **turn this into a single visual flow diagram PNG** showing screens + arrows for navigation—perfect reference for GitHub Copilot when you start coding components.

Do you want me to make that diagram?

Screens





