ChatGPT

Financial Transaction Categorization & Insights Tool – PRD

Background & Problem Statement

Many individuals and small business owners track their income and expenses using Excel spreadsheets, which can make it tedious to analyze spending patterns or prepare reports. Manually categorizing transactions in Excel is time-consuming and prone to error, and advanced analysis (like finding trends or relationships) is difficult without specialized knowledge. In fact, setting up a complex categorization spreadsheet in Excel requires significant training and experience [1]. Properly categorizing financial transactions is crucial for gaining insight: it helps answer questions like "Are we overspending in certain areas?" and simplifies tasks such as budgeting and tax preparation [2]. Professionals categorize expenses to manage budgets better, identify potential savings, and ensure no tax deduction is missed [3]. Despite many personal finance apps existing (e.g. Mint, Quicken, QuickBooks), they may be too complex for casual users or not flexible enough for custom categories. There is a need for a user-friendly tool that leverages modern technology (like AI and graph analysis) to automate transaction categorization and provide clear financial insights, while allowing customization to fit each user's personal or business context.

Goals and Objectives

- **Automate Transaction Processing:** Enable users to upload raw financial data (up to 10 Excel files at a time) containing income and expense records. The system will automatically consolidate and process these files, dividing transactions by date to produce daily income and expenditure totals.
- **Custom Categorization:** Allow users to define their own expense/income categories and provide example keywords or transaction descriptions for each category. The tool should use these examples to intelligently classify each transaction into one or multiple relevant categories. This ensures the categorization reflects how the user manages their finances [4].
- **Business vs Personal Classification:** Provide a way to mark or tag each transaction (or account) as either **Business** or **Personal**, helping users who mix expenses to separate business costs from personal spending [5]. This classification acts as an additional "entity" tag, so users can easily filter or report on business vs personal finances in one place.
- **Insights and Reporting:** Generate a financial report with meaningful insights derived from the categorized data. This includes visual summaries (charts/graphs on a demo dashboard) and an auto-generated narrative highlighting key points (e.g. major expense categories, spending trends, income vs. expense balance, potential savings areas). The report should be clear and approachable, even for non-experts, helping users interpret their financial situation.
- **Flexible Output:** Provide the analysis results in a structured Excel output. If the user supplies a custom Excel template for the report (e.g. a preferred format or corporate template), the system will populate that; otherwise, a default template with standard summary tables and charts is used. This ensures the output can directly meet the user's needs or integrate with their existing reporting workflows.
- **Technology & Integration:** Use Google Firebase as the primary data store (for storing user data, category definitions, transaction records, etc.) and host the processing logic via Google Cloud Run (containerized API endpoints). The product will expose API endpoints for all core functions (file

upload, categorization, report generation) so that it can be integrated into other tools or used programmatically. A minimal web-based demo page will also be provided for users to interact with these APIs (for those who prefer a UI), primarily to upload files and view the generated insights/visualizations.
- **Approachability and Simplicity:** Design the service to be approachable for individuals. The user interface (for the demo web page) and overall workflow should be kept simple and intuitive, avoiding unnecessary complexity or jargon. While advanced techniques (like graph-based AI analysis) will run under the hood, the user experience should feel straightforward – akin to "upload your files and instantly see where your money went" without requiring technical know-how.

Target Users & Use Cases

Target Users: The primary users are individuals and small business owners who are **financially aware but not experts**. These include: - **Budget-Conscious Individuals:** People tracking personal finances (household budget planners, investors monitoring income vs expenses) who use spreadsheets and want deeper insight without learning complex software. They are familiar with financial concepts (expenses, income, budgeting) and value detailed breakdowns. - **Freelancers & Sole Proprietors:** Individuals who have both personal and business transactions mixed in their accounts (e.g. a freelancer who uses one credit card for both personal purchases and client-related expenses). They need to label transactions as business or personal and get a clear split for accounting and tax purposes. - **Small Business Owners or Bookkeepers:** Those who maintain transaction logs in Excel for simplicity and now seek a smarter way to categorize and summarize those for reports. They might not afford full-fledged accounting software or find it too complex, preferring a lightweight solution that still provides professional insights.

Use Cases: 1. Monthly Budget Report: An individual uploads their monthly bank statement exports (Excel/CSV) to the tool. The tool categorizes each line item (salary, groceries, utilities, entertainment, etc.) and produces a report. The user receives an Excel summary and a written highlight (e.g. "You spent the most on Dining Out this month, which is 15% above your average [2]. Consider cutting back to save for your goals."). They also see charts of spending by category and a daily spending trend. 2. Business vs Personal Expense Sorting: A freelancer uploads a spreadsheet of all transactions for the year. They have defined categories (some marked as business-related, like "Travel - Business", and others personal). The tool tags each transaction accordingly (leveraging the user's input or past patterns). It then provides separate subtotal views for business expenses vs personal expenses, helping the user prepare for taxes with proper categorization (e.g. total Business: Office Supplies, Business: Travel, etc., vs Personal: Groceries) [6] [5]. 3. Insight Generation for Financial Planning: A user has several Excel files: one for personal expenses, one for income, and one for a side-business. By uploading all, the system consolidates them to show a holistic picture. The Graph-based AI engine finds that a spike in personal "Travel" expenses coincided with a high revenue month from the side-business, suggesting a business trip was categorized partly as personal. The tool flags such insights in the narrative report (e.g. identifying transactions that might belong in both categories or that an unusual expense occurred related to a certain income source). 4. Custom Template for Client Reporting: An independent accountant uses the tool to prepare reports for a client. The client provided an Excel template they prefer (with company logo and specific layout). The accountant uploads the client's raw expense spreadsheets along with the template. The tool categorizes all transactions according to categories aligned with the client's business (the accountant pre-defined these in the tool, matching the client's chart of accounts). The output is generated in the provided template format, ready to deliver to the client with minimal manual editing.

In all these cases, the users benefit from saving time on manual categorization and gaining a clearer understanding of their finances through automated insights.

Key Features & Requirements

1. Multi-File Data Ingestion (Excel/CSV Upload)

- **Upload Interface:** Users can upload up to 10 files (Excel `.xlsx` or CSV) in one session. This accommodates splitting data by accounts or months. The system will accept common financial record formats (e.g. bank statement exports in CSV, expense logs in Excel).
- **Data Extraction:** Upon upload, the backend service reads each file, normalizes the data, and merges it into a unified list of transactions. Each transaction will have at least: Date, Description, Amount (positive for income, negative for expense or separate fields), and maybe Category (if the spreadsheet already has one, though the user may want to recategorize). If different files have different schemas, the system will prompt the user to map the columns (or will auto-detect common patterns, like a column named "Description" likely contains the transaction detail).
- **Daily Aggregation:** The tool will automatically group transactions by date to compute daily total income and total expenditure. This enables daily tracking (the user can see, for example, how much was earned or spent each day) and lays groundwork for time-series analysis (trends over days or months). The daily breakdown could feed into charts (like a daily spending line graph).
- **Error Handling:** If a file is formatted incorrectly or contains non-numeric data in amount fields, the system should gracefully notify the user which file/row had an issue. It will skip or flag invalid entries but continue processing the rest. There is a hard limit (e.g. 10,000 rows per file) to ensure performance on Cloud Run; if exceeded, the user is advised to split the data.

2. Custom Categories & Categorization Logic

- **User-Defined Categories:** Users can create and name their own categories for income and expenses. For example, one user might have "Dining Out" and "Groceries" as separate categories under a broader "Food" group, while another user might just have a single "Food" category. The system does not enforce a fixed taxonomy; it's flexible to reflect how each user thinks about their money [7]. Users can also specify if a category is an Income category or Expense category, to avoid mixing the two in summaries.
- **Category Examples (Training Data):** When defining a category, the user may input example keywords or phrases that characterize that category. For instance, for a category "Utilities", the user might provide examples like "electric bill", "water company", or merchant names like "ConEdison". The categorization engine will use these hints to automatically classify transactions. This functions similarly to rule-based tagging: if a transaction description contains one of the keywords, it's assigned that category. We will improve this with AI pattern matching as well – e.g., using embeddings to catch that "Power Co." is also a utility.
- **Auto-Categorization & Learning:** The system will attempt to categorize each transaction automatically using the user's category definitions and examples. Over time, it can learn from corrections: if a user changes a category for a transaction, the system should remember that preference for the future (much like Quicken "learns your preferences over time" for categorization) [4]. Frequent merchants or descriptions will get consistently categorized once the user has trained the system on them. The goal is that after initial setup, new uploads require minimal manual categorization.
- **Multiple Category Tags:** By default, each transaction will have one primary category (for clear expense summing). However, we also support tagging a single transaction with multiple "entities" or tags if applicable. For example, a single expense like a business trip dinner might be tagged as both **Business** and **Travel** – meaning it's a Travel expense that was business-related. The user might categorize it under "Travel" category for expense tracking, but also mark it as "Business" via the Business/Personal tag (see next section) or even tag with a project name. This multi-tag approach is facilitated by the system's data model: akin to how Quicken allows a category plus a

tag for depth (e.g. tagging an expense with a project or client without cluttering the primary category) [8]. In summary, the tool treats categories in a hierarchical or multi-dimensional way: one dimension for personal vs business, another for the actual spending category, and optional free-form tags for any other grouping the user needs.
- **Business vs Personal Entity:** As part of category setup, the user can indicate which categories (or individual transactions) are "Business" related versus "Personal". This could be implemented as a top-level classification (for example, the user could have two parallel category sets, or simply a checkbox on each category or transaction). In reports, this allows filtering or splitting of totals by business vs personal. It ensures, for instance, that if the user shops at the same store for both business and personal items, they can separate those transactions appropriately [5]. The system will enforce that every transaction gets labeled as either personal or business (either inferred from the category label or defaulted to personal unless specified).

3. Data Storage & Backend Architecture

- **Firebase Integration:** Firebase will serve as the primary backend: using Cloud Firestore (NoSQL document database) to store user profiles, category definitions, and processed transaction data. Each user's data is isolated and secured. Storing categories and examples in Firestore allows the Cloud Run service to retrieve user-specific rules quickly when categorizing new data. We may also use Firebase Authentication if we require login; otherwise, a simple key or token per user (for API usage) can be managed.
- **Google Cloud Run Services:** The core logic runs as serverless containers on Cloud Run. We will implement a set of RESTful API endpoints:
- `POST /upload` – Accepts file uploads (possibly directly or via a pre-signed URL approach if files are large) and triggers the processing (parsing and categorization). Returns a job ID or the processed result.
- `GET /result/{jobID}` – Retrieves processed data or signals completion (if asynchronous processing is used for large files).
- `GET /report/{jobID}` – Retrieves the generated financial report (possibly as a downloadable Excel or PDF, and/or JSON of insights).
- `GET /visualize/{jobID}` – Returns data needed for visualizations (like an endpoint the demo page can call to get chart data).

Each of these endpoints will be stateless and scale automatically via Cloud Run. The processing logic will utilize the category rules from Firebase and the Graph-based algorithms (see below) to perform its work.
- **Performance Considerations:** Cloud Run can scale horizontally for multiple users; each uploaded batch of files will typically be processed within a few seconds for a few thousand transactions (which is expected for 10 Excel files). We'll implement batching and possibly background tasks if needed (e.g., if generating the narrative report takes longer, that might be done asynchronously). Storing the results in Firebase allows the front-end to poll or get updates via Firebase listeners, providing near-real-time feedback to the user after upload.
- **Security & Privacy:** Financial data is sensitive. All communication will be over HTTPS. Uploaded files will be processed in-memory or temporarily stored in secure cloud storage buckets (with limited time access). Firebase rules will ensure only the authenticated user (or someone with their API key) can access their data. We will not use the data for any purpose other than providing the service to the user, and we'll clearly document data handling in a privacy policy. Optionally, we might allow users to opt-out and have their data wiped after report generation (no storage), if they just want a one-off analysis.

4. Data Analysis Engine – GraphRAG for Similarity & Insights

- **GraphRAG Overview:** The tool will incorporate a **Graph-based Retrieval-Augmented Generation (GraphRAG)** approach in its analysis engine. GraphRAG enhances traditional data retrieval by using a knowledge graph of the transactions and their related entities (categories, merchants, dates) to discover connections and context that linear data might miss [9]. In essence, the system will build a graph where nodes could include entities like Transaction, Category, Merchant/Payee, Date (time periods), and even Location or Project if available. Edges illustrate relationships, such as a transaction **belongs to** a category, or a merchant **is associated with** a category, or a transaction **occurred on** a date. By traversing this graph, the system can retrieve similar transactions or patterns – for example, finding all expenses related to a specific project across different months, or linking a spike in one category with an increase in another. This structured graph approach brings deeper insight: it "connects the dots" in ways a flat spreadsheet cannot [9].
- **Similarity Retrieval:** Using the graph, when a new transaction comes in, the system can find similar past transactions to suggest a category (e.g., if the description "Starbucks" appears, the graph knows past Starbucks transactions were categorized as "Dining" or "Coffee" and can flag that). It can also find pairs of income and expenses that might be related (for instance, an invoice payment income that often is followed by a specific expense). By analyzing relationships, the tool might identify that certain expenses only occur when a particular income source is higher, indicating a causal relationship (useful for financial planning).
- **Enhanced Report Generation:** The relationships stored in the graph database can be leveraged when creating the narrative insights report. Because the knowledge graph provides structured context, the system's AI can generate more contextually aware commentary. For example, if the user asks for "find any unusual expenses," the system can traverse the graph to find outliers or nodes with few connections (a one-off large purchase), and the AI can include that in the story. This method is particularly useful for comprehensive financial reporting where the relationships between entities are important [10] – our tool will highlight those relationships (e.g. "Your travel expenses are typically 5% of monthly spend, but in July they spiked to 15% due to a business conference trip, which was offset by higher business income that month."). The GraphRAG approach ensures the AI doesn't just look at isolated data points, but understands the context (like linking that conference expense to business income).
- **Technical Implementation of GraphRAG:** We will likely use a graph database (such as Neo4j or an in-memory networkx graph if data is small) to store entities and relations. When processing data, we populate the graph. For generating text insights, we use retrieval queries on the graph to fetch relevant info (e.g., highest expense node, clusters of related expenses) and feed those to an LLM (Large Language Model) which is responsible for writing the narrative. This way, the output is fact-based and context-rich. (Note: This is an innovative feature that will be developed carefully; if it proves too complex for the initial release, we will implement a simpler rule-based insight generation and treat GraphRAG as a future enhancement or an experimental beta feature.)

5. Visualization Dashboard (Demo Page)

- **Purpose:** Alongside the API, a simple web dashboard will be provided mainly for demonstration and for less technical users. This single-page web app will let users perform the core actions: upload files, configure categories, and view results (charts and report). It's not intended to be a fully polished commercial app at first, but rather a reference UI and a quick way to get feedback on the API's functionality.
- **Visual Charts:** The dashboard will display interactive charts to help users understand their income and spending:

- A **daily trend chart** (line graph) showing income vs. expenses over time (e.g., for the date range covered by the uploaded files). This highlights peaks in spending or income and can show net savings by day.
- A **category breakdown** (e.g. pie chart or bar chart) for expenses and for income, showing the proportion each category contributes to the total. For example, a pie chart of expenses might show what percentage of spending went to “Rent”, “Food”, “Travel”, etc., which is useful for seeing where money is going [11] .
- Perhaps a **personal vs business split** visualization if relevant (like a stacked bar showing total personal vs business expenses). This could be useful for those blending finances to quickly see the distribution.
- Any other charts to enhance understanding, like a cash flow bar chart per month (if data spans multiple months), or a top 5 merchants bar chart to see where most money is spent.
- **Interactive Filtering:** Users should be able to filter the visuals (e.g., view only Business expenses, or view a specific category over time). This can be done with simple checkboxes or dropdowns on the demo page. The filtering logic can be done client-side if data is small, or via API queries if needed (for larger data sets).
- **Technology:** The front-end might be a simple JavaScript app (perhaps using React or just plain HTML/JS for speed) served via Firebase Hosting. Visualization can use a library like Chart.js or D3 for simplicity. The page will call the Cloud Run APIs to get data. For example, after upload, it might call `/visualize` to get aggregated data for charts instead of computing that in the browser.
- **User Experience:** Emphasis is on clarity – e.g., color-coded charts matching categories, tooltips with exact values, and short explanatory text. The design will be clean and uncluttered, aiming to make financial figures less intimidating. Given the target users, we want them to quickly grasp their financial picture (the charts plus the narrative report together should cater to both visually oriented and text-oriented users).

6. Financial Report Generation (Narrative Insights)

- **Automated Report Writing:** In addition to visualizations, the system will produce a written **Financial Insights Report**. This report is essentially a narrative summary (a few paragraphs) that highlights key information from the data:
- **Summary:** E.g., “In the period from Jan 1 to Jan 31, 2026, you earned \$5,000 and spent \$4,500, resulting in a net saving of \$500. Your largest expense category was **Rent** at \$1,200 (26% of total expenses), followed by **Dining Out** at \$800 [2] . You categorized 15% of your expenses as business-related.” This gives a quick snapshot.
- **Trends & Patterns:** The report will mention notable trends, such as “Your spending on Travel increased in July due to a one-time conference trip (business expense) [10] ,” or “Compared to last month, your grocery spending decreased by 10%, indicating better control on food expenses.” If income has seasonality or a big one-time inflow, that will be noted too.
- **Anomalies/Alerts:** If the tool finds anything unusual (like an expense that doesn’t fit usual patterns, or a category that spiked/dropped significantly), the report will call it out. For instance, “Unusual expense: A \$500 purchase at ABC Electronics was recorded, which is significantly higher than your typical spending in the Electronics category.”
- **Recommendations:** Based on the insights, the report might gently suggest actions, e.g., “You spent \$300 on subscriptions this month. Consider reviewing recurring subscriptions to identify any that could be canceled to save money.” Or if there’s high interest debt payments, it might suggest looking into refinancing, etc. These can be template-driven suggestions triggered by certain conditions in the data.
- **Customization and Templates:** The narrative report could be delivered as a PDF or as an editable document. If the user wants it in Excel (as a text summary on one sheet) or in a Word template, we will consider those output options. Initially, it might be a PDF generated by the system for

simplicity. The user can choose a tone (e.g., professional vs casual) if needed, but by default it will be in plain, clear language. We aim for the tone to be similar to how a financial advisor might explain the data to a client, but in simple terms (no heavy jargon).

- **Generation Method:** The report will be generated by an AI component (likely using an LLM) guided by data facts. It will use the aggregated figures (from the analysis engine) to fill in blanks in predefined narrative structures. For example, we will craft narrative templates for common patterns (overspending, underspending, consistent trends, one-time events, etc.) and let the AI select and fill those as appropriate. GraphRAG comes into play by ensuring the AI has access to the graph of facts while writing, so it can cross-check relationships and avoid incorrect statements. The final output will also be reviewed by deterministic checks (to ensure the numbers in the text match the numbers in the data).
- **Example Insight in Report:** “Dining Out expenses totaled \$800, which is a 20% increase compared to the previous month. This category now makes up 18% of your total monthly spending, exceeding your set budget for Food. It appears many of these were business meetings (5 out of 20 transactions tagged as Business), so you might consider classifying some of those as business expenses for reimbursement or deduction [5].” This kind of insight ties together multiple dimensions (increase over time, proportion of total, relation to budget, personal vs business split) in a user-friendly way.

7. Custom Excel Output (Template Integration)

- **Default Output:** By default, the system will generate an Excel workbook that includes the raw transaction list (post-categorization) and some summary sheets. For instance, one sheet might list all transactions with their assigned category and tags, another sheet might have a pivot table or summary of total per category, and another sheet could have a copy of the charts or key metrics. This gives users a familiar format they can further manipulate if needed.
- **User-Provided Template:** A key feature is the ability for users to upload an **output template** Excel file. This template would contain placeholders or specific formatting where the user wants the data to appear. For example, a user might have a template with a nicely formatted table for “Expenses by Category” or a certain layout they use monthly. Our system will need to map the data into this template – possibly by expecting certain cell names or markers. If a template is provided, the system will populate it with the equivalent of the default output data. Any parts of the template it cannot fill (because of missing data or misnamed placeholders) will remain as-is for the user to adjust manually.
- **Flexibility:** Supporting arbitrary templates is complex, so initially we might support a handful of template styles (documented for the user) or a simpler approach like: if the user provides a blank workbook with specific sheet names (“Transactions”, “Summary”, “Report”), we fill those sheets in a predefined way. Over time, we can add more sophisticated template parsing.
- **Export Formats:** In addition to Excel, the user may choose to get the report in PDF format (especially the narrative part). The tool can generate a PDF that includes the charts and the written summary, which is convenient for sharing or printing. This would be generated from the same data – possibly by assembling a HTML or LaTeX report under the hood.
- **Use Case Example:** If an accountant user has a standardized report format they give to clients, they upload that once as the template. Every month, they just feed new transaction files and get back that same formatted report with updated numbers – saving them from manually copying data into the report format.

Financial Categorization Best Practices (Research Insights)

In designing this product, we incorporate financial best practices to ensure the tool's output is not just technically correct but also meaningful for financial planning and accounting:

- **Standard Categories & Personalization:** Users are encouraged to start with common expense categories and then customize. Common personal expense categories include Home (rent/mortgage, utilities), Transportation (car, gas), Food (groceries, dining out), Entertainment, Healthcare, etc., while common business expense categories include Office Supplies, Travel, Professional Services, Utilities (business), etc. [12] [13]. The tool can provide a default list of such categories as a starting point. However, it also emphasizes that each person's situation is unique – users can and should modify categories to fit their lifestyle or industry [14]. For example, a photographer might need a category for "Equipment" or "Studio Rental" that general templates don't include. We will allow creation of sub-categories as well (e.g., "Travel" with sub-categories "Travel: Personal" and "Travel: Business" if desired) to add detail where needed.
- **Business vs Personal Separation:** Mixing business and personal expenses is a common challenge. Best practice, which the tool follows, is to clearly separate them for accuracy and tax purposes [5]. Our category system explicitly tags transactions accordingly, and we recommend users maintain that discipline. The tool's tips (possibly shown in the UI or documentation) will mention to mark accounts or transactions as business or personal upfront, similar to how professional software does, to keep records "audit-ready" and clean [15].
- **Why Categorize – Emphasizing Benefits:** The tool's onboarding or documentation will reiterate why categorization matters, echoing professional advice: It turns raw data into actionable insights. With categorized data you can track spending patterns, spot areas to cut costs, prepare accurate budgets, and maximize tax deductions [11]. For instance, by seeing all "Office Expense" items totaled, a small business owner can know if they're overspending on supplies and adjust accordingly. If something is uncategorized, it's essentially invisible in analysis, so we will encourage users to categorize everything (even if it's an "Other" catch-all category).
- **Tagging and Multi-Dimensional Analysis:** As noted, we implement a tagging system to allow multi-dimensional analysis without overwhelming the category list [8]. This is in line with best practices used by tools like Quicken and others: keep the main categories broad and manageable, and use secondary tags for finer distinctions or cross-cutting labels (like project names, client names, or any attribute that one might want to filter by later) [8]. Our PRD reflects this by planning support for tags and encouraging their use for detail that doesn't warrant a dedicated category.
- **Category Recommendations and Examples:** The system will include guidance on picking categories. For example, it may have a help section: "If categorizing for taxes, consider categories that align with tax forms (e.g., if you'll file a Schedule C for business, use categories like those on the form: Advertising, Legal Fees, Meals, etc.) [16]. If categorizing personal finances, choose categories that reflect your spending priorities (Housing, Utilities, Savings, etc.)." We may cite that transaction categorization is specific to each context, and even suggest consulting an accountant for the optimal category setup for one's scenario [17]. This ensures the categories chosen serve the user's goals (be it better budgeting or easier tax prep).
- **Accuracy and Review:** No automatic system is perfect, especially with varied transaction descriptions. Professionals advise reviewing transactions regularly to catch mis-categorizations [18]. We will incorporate this advice by perhaps highlighting transactions that the system was less confident about, or providing an interface for users to review and adjust categories before finalizing the report. Over time, these corrections train the model and improve accuracy. The user stays in control – they can always override a category choice.

- **Continuous Improvement:** Because financial patterns change (new merchants, new types of expenses), the system's machine learning component will continuously update. We'll also allow the user to update their category rules (add a new keyword for a category when a new type of expense appears). This way the categorization remains up-to-date with the user's life.

By grounding the product in these best practices, we aim to offer a tool that not only uses cutting-edge tech but also produces output that aligns with what professionals would expect in financial management.

Competitive Analysis & Differentiation

The personal finance management space has several players, but our tool differentiates itself in specific ways:

- **Manual Spreadsheet vs. Automated Tool:** Many individuals still use manual spreadsheets or basic templates to track finances, as it gives them control. Our tool targets these users by maintaining flexibility (user-defined categories, Excel outputs) while removing the drudgery of manual updates. Unlike using Excel alone, our tool automates categorization and analysis, which "helps spot where you might be overspending and adjust to save" without the user doing the math [2]. It essentially supercharges the spreadsheet approach with automation and intelligence.
- **Budgeting Apps (Mint, etc.) vs. Custom Categories:** Popular apps like Mint or personal budget apps automatically categorize transactions from bank feeds, but they use broad default categories and often misclassify niche transactions (users often complain that these apps put many items in "Other" or wrong categories). Our approach gives the user full control over categories and uses their own rules/examples, leading to more accurate and personalized categorization. Additionally, Mint doesn't easily allow multiple category tagging or custom output formats, which our tool does. The ability to export to a user's own template is a rare feature not found in typical consumer apps.
- **Small Business Accounting Software (QuickBooks, Xero):** These are powerful but often overkill for individuals or freelancers, plus they require strict adherence to accounting principles (double-entry bookkeeping, linking to bank accounts, etc.). Our tool is **simpler and more approachable** – no accounting knowledge needed, just categorize expenses in ways that make sense to you. Yet, it can still separate business vs personal finances in one place, something even QuickBooks doesn't handle (users usually maintain separate books for personal vs business). Quicken does allow combined personal/business tracking and auto-categorization [6], but Quicken is a heavy install and not cloud-based. Our tool is lightweight, cloud-based (nothing to install), and focuses on just the key features relevant to this user segment.
- **AI-Powered Insights:** Few if any personal finance tools aimed at consumers use advanced AI like GraphRAG to deliver insights. High-end financial planning software or enterprise tools might, but those aren't accessible to individuals. By using GraphRAG and LLMs, our tool can provide a level of analysis (finding hidden patterns, generating narrative explanations) that sets it apart in terms of intelligence. For example, enterprise solutions like those from insightsoftware or others do "automated narrative reporting" for large companies [19], but we are bringing similar capabilities to an individual's budget. This can feel like having a virtual financial advisor analyze your data for you.
- **API and Developer-Friendly:** Many finance apps are closed systems. By providing API endpoints and possibly integrations (e.g. a Zapier connector or similar in future), tech-savvy users or even other products could plug into our service. For instance, a user could set up a Google Drive trigger such that when they drop a new statement file, it calls our API and returns an updated report. This openness could attract a niche of users who want to integrate financial analysis into their own workflows or products.

- **Cost and Monetization:** While not a direct part of the PRD feature set, it's worth noting our likely model: a freemium or subscription service with a generous free tier for individuals. This contrasts with some competitors that require paid subscriptions for full functionality (e.g., Monarch Money or YNAB are paid services, QuickBooks is subscription-based). By offering core features for free or a low cost (perhaps limiting number of files or using ads for free tier), we can attract users who are cost-sensitive yet need these capabilities. The business model can be further refined, but the focus in this PRD is ensuring the product has unique value to draw users in the first place.

In summary, our product is carving a niche between do-it-yourself spreadsheets and full-fledged finance software: providing automation and smart insights while retaining user control and simplicity. Its unique combination of custom categorization, business/personal dual-use, advanced AI analysis, and flexible output makes it stand out in the market.

Technical Considerations and Future Enhancements

- **Scalability:** As usage grows, we'll monitor the performance of Cloud Run and Firestore. They should scale automatically for our needs, but heavy graph operations or large files might require optimizations (caching results, preprocessing data in batches, etc.). We plan to implement pagination or limits on data returned (e.g., don't send tens of thousands of transactions to the frontend at once; instead, show summary or allow filtered queries).
- **GraphRAG Development Risks:** Building a full knowledge graph and integrating with an LLM can be complex [20]. We will take an iterative approach: possibly start with simpler graph relationships or even a traditional vector search for similar transactions, and only gradually build out the graph features. This mitigates the risk of spending too long on one feature. As a contingency, if GraphRAG proves too challenging initially, the tool will still function with rule-based categorization and template-based insights, which already deliver value. GraphRAG can then be introduced as a beta feature to select users.
- **Accuracy of AI-Generated Reports:** We must ensure that any AI-generated text is accurate and doesn't hallucinate financial facts. This is critical for user trust. Techniques to ensure this include: providing the AI only with computed facts (sums, percentages, etc.) from the data and not open-ended raw data; using prompt templates that force it to mention those facts; and perhaps having a verification step (for example, the system could parse its own generated text to double-check that numbers mentioned match the data). User feedback will also be important – we might allow users to rate the helpfulness of the report so we can adjust the prompts over time.
- **User Support & Education:** Since some concepts (like setting up category rules or tags) might be new to users, we'll include a quick tutorial or examples in the app. For instance, when the user goes to define categories, we could show: "Example: Category = Utilities, Keywords = electric, gas, water" to guide them. Similarly, for the first-time user of the demo, a sample dataset could be available to see how it works.
- **Extensibility:** In the future, the platform could integrate directly with banks (to pull transactions via APIs) eliminating the need for manual file uploads. But given our initial target user is comfortable with Excel, we're focusing on that input method first. We also foresee possibly expanding to handle receipts/invoice OCR (since Firebase + Cloud Run could integrate with an OCR service) – for now, out of scope, but the architecture (with APIs and possibly using DocuClipper's OCR API [21] in future) can accommodate that later.
- **Timeline:** We plan to deliver a functional MVP in approximately 3 months, focusing on the core features: file upload, categorization with custom categories, basic charts, and a simple generated text summary. GraphRAG-based enhancements and custom template output might come in a later version (say 2-3 months after MVP) once we have real user data to refine those advanced features. Throughout development, we will involve a few beta users (perhaps the personas

mentioned) to test and provide feedback, ensuring the product remains aligned with user needs and is easy to use.

Conclusion

This PRD outlined a comprehensive plan for a **Financial Transaction Categorization & Insights Tool** that leverages both user-driven customization and advanced AI techniques to deliver a powerful yet user-friendly experience. From a **business perspective**, it addresses a clear need for many individuals and small businesses to better understand their finances without investing in heavy software or manual labor – turning what could be hours of spreadsheet work into an instant report. From a **product engineering perspective**, it uses reliable cloud components (Firebase, Cloud Run) and innovative AI (GraphRAG, LLMs) to create a scalable, modern solution. And from a **financial research perspective**, it embeds best practices of financial management (accurate categorization, separation of concerns, insightful analysis) into the product's design [22] [11].

By combining these elements, the tool aims to empower users to make smarter financial decisions with minimal effort. They remain in control of how their data is categorized and presented, but benefit from automation and expert-level analysis running behind the scenes. Ultimately, success will be measured by users gaining clarity and confidence in their financial situation – if a user can say, "I understand where my money is going now, and I know what I can do about it," then this product will have achieved its goal.

Sources Cited: The design and requirements above incorporate insights from financial software best practices and expert advice on expense categorization, such as Quicken's approach to categories and tags [8] [4], and professional guidance on why categorizing expenses is valuable [23] [3]. Advanced techniques like GraphRAG are referenced from recent developments in AI for finance [9] [10] to ensure our solution is forward-looking and robust. These references underscore the rationale behind features and validate the chosen direction.

---

[1] [2] [3] [14] [17] [21] [22] [23] How To Categorize Expenses In Excel (Template Included) - DocuClipper
https://www.docuclipper.com/blog/how-to-categorize-expenses-in-excel/

[4] [5] [6] [7] [8] [11] [12] [13] [15] [16] [18] Using Categories to Track and Separate Business and Personal Finances
https://info.quicken.com/sim/using-categories-to-separate-business-and-personal

[9] Boost Financial Intelligence Using GraphRAG and Graphs in Finance – TF Financial Services
https://tf-fs.com/de/boost-financial-intelligence-using-graphrag-and-graphs-in-finance/

[10] [20] GraphRAG: Design Patterns, Challenges, Recommendations
https://gradientflow.substack.com/p/graphrag-design-patterns-challenges

[19] Automated Financial Storytelling at Your Fingertips: Here's How
https://insightsoftware.com/blog/automated-financial-storytelling-at-your-fingertips-heres-how/