AI Driven Personal Finance Assistant

**Desire:**

* The app should allow new users to register using their personal details.
* The app should allow registered users to log in securely using their credentials.
* Users want to securely add their bank account details with verification and confirmation.
* The app should allow me to view an overview of my financial status after linking my bank account.
* The app should automatically categorize my transactions into budgets.
* The app should allow me to set monthly budget goals for different categories.
* The app should notify me when I’m approaching or exceeding my budget limits.
* The app should remind me about my bill payments like electricity, gas, etc., before they’re due.
* The app should provide AI-generated insights and suggestions to improve my spending habits.
* The app should allow me to set financial goals and track progress.
* The app should allow me to export my financial data in PDF or Excel format for my personal use.
* The app should let me securely log out and lock my account when I’m done using it.
* The app should let me delete my account if I no longer want to use it.

**User Story: 1**

**Title:** User stories for New user register with their personal details.

**Stakeholders:** New User, PFA-SYSTEM, Email/SMS Module

**Stories**

* As a New User, I want to:
* Provide my personal details for registration.
* Receive an OTP for verification to ensure my contact details are correct.
* Enter a secure password that meets the system’s security standards.
* As the PFA-SYSTEM Module, I want to:
  + Collect user details and validate the provided information.
  + Generate and send an OTP to the user's email or phone.
  + Verify the OTP to confirm the user’s identity.
  + Enforce password strength to enhance security.

**Steps:**

1. Enter Personal Details:

* Theuserinputstheir **Full Name, Email, Phone Number, and Country** intheregistrationform**.**
* The system checks for **valid formats**.

1. Request and Verify OTP:

* The user **requests an OTP** for verification.
* The system **sends the OTP via email or SMS** and stores it temporarily.
* The user enters the OTP:
* **If correct**, proceed to password creation.
* **If incorrect or expired**, show an error and allow the user to request a new OTP.

1. Create and Validate Password:

* The user creates a **strong password** following these rules:
* Minimum **8 characters**.
* No easily guessable words.
* The system validates password strength and confirms account creation.

1. Successful Registration:

* If all steps are completed, the system stores user details securely and redirects the user to the login page.

**User Story: 2**

**Title:** User stories of registered users to log in securely using their credentials.

**Stakeholders:** Registered Salaried User, PFA-SYSTEM

**Stories:**

* As a Registered User, I want to:
* Enter my credentials to log in.
* Receive an error message if I enter incorrect credentials.
* Access my dashboard securely after a successful login.
* As the PFA-SYSTEM Module, I want to:
* Accept and verify user credentials to authenticate the user.
* Show an error message if the credentials do not match the records.
* Grant access to the user dashboard if login details are correct.

**Steps:**

1. Enter Login Credentials:

* The user enters their Email Address or Phone Number.
* The user enters their Password.

1. Submit Login Request:

* The user clicks on the Login button to proceed.

1. Validate User Credentials:

* The system checks if the email/phone number exists in the database.
* The system verifies if the password matches the stored record.
* If validation fails:
* Show an error message.
* Allow the user to retry login or reset password.
* If validation succeeds:
* Grant secure access to the user’s dashboard.

1. Successful Login:

* The system redirects the user to their dashboard, where they can access personal finance features.

**User Story: 3**

**Title:** Registered users add their bank account details securely.

**Stakeholders:** Registered User, PFA-SYSTEM, Banking Module

**Stories:**

* As a Registered User, I want to:
* Add my bank account details securely.
* Ensure my bank details are verified before being linked to my account.
* Receive a confirmation message once my bank account is successfully added.
* As the PFA-SYSTEM Module, I want to:
* Validate the entered bank details for correctness
* Store the bank account details securely after verification.
* Notify the user upon successful addition of their bank account.

**Steps:**

1. Enter Bank Account Details:

* The system checks for valid input formats

1. Verify Bank Account:

* The user may be required to verify ownership via OTP from their bank.
* If verification fails, the user is prompted to re-enter the correct details.

1. Store Bank Details Securely:

* Upon successful verification, the bank details are encrypted and stored.
* The system sends a confirmation message via email/SMS.

1. Successful Addition of Bank Account:

* The user sees a success message and can now proceed to salary processing or other financial services.

**User Story: 4**

**Title:** User stories for viewing financial status overview after linking a bank account with manual expense/ income input option.

**Stakeholders:** Registered User, PFA-System, AI Module

**Stories:**

* As a Registered User, I want to:
* View a clear overview of my financial status, including total balance, recent income, and expenses, after linking my bank account.
* See a summary that’s easy to understand to quickly assess my finances.
* Manually add or modify expense and income entries if the bank data is incomplete or needs adjustment for a more accurate report.
* Save my manual entries to update the overview instantly.
* As the PFA-SYSTEM Module, I want to:
* Retrieve and display the user’s bank account data securely after linking.
* Provide an interface for users to input manual expense or income entries with fields like amount, date, and category.
* Validate manual entries for basic accuracy and integrate them into the financial overview.
* Store all data (bank-linked and manual) securely for future use.
* As the AI Module, I want to:
* Analyse the combined data to generate an accurate financial summary.
* Suggest basic categorization for manually entered transactions to assist the user.

**Steps:**

1. Display Financial Overview:

* The user logs in and navigates to the “Financial Overview” section.
* The system retrieves bank account data.
* The system presents a summary:
* Total balance
* Total income
* Total expenses
* Simple visualization

1. Option for Manual Modification:

* The user sees an option labelled “Add Manual Entry” or “Edit Transactions.”
* The user can:
* Add a new expense/income: Enter amount, date, description, and optional category.
* Modify an existing bank transaction: Adjust amount or add a note.
* The system validates inputs:
* Amount must be numeric and positive.
* Date must be valid and not in the future.

1. AI-Assisted Categorization:

* For manual entries, the AI suggests a category based on the description.
* The user can accept the suggestion or choose their own category from a dropdown.

1. Update and Save:

* The system updates the financial overview in real-time after saving manual entries.
* The user sees the revised balance, income, and expense totals reflecting both bank data and manual inputs.
* The system confirms successful updates with a message.

1. Error Handling:

* If bank data fails to load, show an error message: “Unable to fetch bank data. Please try again or add manually.”
* If manual entry validation fails, display specific feedback.

**User Story: 5**

**Title:** User stories for automatic categorization of transactions into budget categories.

**Stakeholders:** Registered User, PFA-System, AI Module

**Stories:**

* As a Registered User, I want to:
* Have my bank transactions automatically sorted into budget categories without manual effort.
* View a list of my transactions with their assigned categories for transparency.
* Adjust or correct a category if the AI’s suggestion doesn’t match my intent.
* See categorized transactions reflected in my financial overview.
* As the PFA-SYSTEM Module, I want to:
* Retrieve the user’s bank transaction data securely.
* Provide a predefined set of budget categories.
* Store the categorized transactions and any user adjustments for future reference.
* Update the financial overview with categorized data.
* As the AI Module, I want to:
* Analyse transaction details to assign the most likely budget category.
* Learn from user corrections to improve future categorization accuracy.
* Handle ambiguous transactions by assigning a default “Uncategorized” label for user review.

**Steps:**

1. Transaction Retrieval:

* The user logs in and navigates to the “Transactions” or “Overview” section.
* The system fetches recent bank transactions linked to the user’s account.

1. Automatic Categorization:

* The AI analyses each transaction based on:
* Merchant
* Amount patterns
* Transaction description
* The system assigns each transaction to a predefined category from a list such as:
* Groceries
* Utilities/Bills
* Entertainment
* Transportation
* Dining Out
* Miscellaneous (default for unclear cases).

1. Display Categorized Transactions:

* The user sees a list of transactions with details:
* Date, Amount, Merchant, Category.
* Example: “10/15/2023, $45.32, Walmart, Groceries.”
* Uncategorized transactions are flagged for user attention.

1. User Review and Adjustment:

* The user can tap a transaction to edit its category.
* A dropdown of predefined categories appears, and the user selects the correct one.
* The system saves the adjustment and updates the overview accordingly.

1. AI Learning and Feedback:

* The AI records user corrections.
* The system applies this learning to future transactions from the same merchant.
* If a transaction remains uncategorized after a set period, the app prompts the user to assign a category.

1. Integration with Overview:

* The categorized data updates the financial overview.
* Manual adjustments from Step 4 are reflected immediately.

**User Story: 6**

**Title:** User stories for setting monthly budget goals for different categories.

**Stakeholders:** Registered User, PFA-System, AI Module

**Stories:**

* As a Registered User, I want to:
* Set specific monthly budget amounts for each transaction category.
* See suggested budget amounts based on my past spending habits for guidance.
* Edit or reset my budget goals anytime during the month.
* Have my budget goals saved and applied to track my spending progress.
* As the PFA-System Module, I want to:
* Provide an interface for users to input and manage budget goals for each category.
* Display the list of categories with fields to set budget amounts.
* Store the user’s budget goals securely and associate them with the current month.
* Update the financial overview with budget limits for reference.
* As the AI Module, I want to:
* Analyse the user’s past transaction data to suggest realistic budget goals for each category.
* Adjust suggestions based on typical spending patterns.
* Flag categories where suggested budgets might be too low or high based on historical data.

**Steps:**

1. Access Budget Settings:

* The user navigates to the “Budget” section from the app menu.
* The system displays a list of categories from the transaction categorization feature.

1. View AI-Suggested Budgets:

* For each category, the AI suggests a budget based on the user’s last 30-90 days of spending.
* Suggestions are marked as optional, with a note: “Suggested based on your spending.”

1. Set Budget Goals:

* The user enters a budget amount for each category:
* Manual input: Type a number.
* Accept AI suggestion: Tap “Use Suggested” to auto-fill.
* The system validates inputs:
* Amount must be positive and numeric.
* Warn if total budget exceeds typical income.

1. Save and Apply:

* The user taps “Save Budget” to confirm their goals.
* The system saves the budget goals and links them to the current month.
* The financial overview updates to show budget limits alongside spending.

1. Edit Budget Goals:

* The user can revisit the “Budget” section anytime to adjust amounts:
* Increase/decrease a category budget.
* Reset to AI suggestions with a “Reset to Suggested” option.
* Changes are saved and reflected immediately in the app.

**User Story: 7**

**Title:** User stories for notifying users when approaching or exceeding budget limits.

**Stakeholders:** Registered User, PFA-System, Notification Module

**Stories:**

* As a Registered User, I want to:
* Receive a notification when my spending in a category is nearing my budget limit.
* Get an alert when I’ve exceeded a category’s budget so I can adjust my spending.
* Customize notification preferences.
* See details in the notification to take immediate action.
* As the PFA-System Module, I want to:
* Track the user’s spending against their set budget goals in real-time.
* Calculate when a category’s spending approaches or exceeds the budget limit.
* Provide a settings interface for users to adjust notification preferences.
* Log notification events for troubleshooting or user review.
* As the Notification Module, I want to:
* Send timely push notifications to the user’s device based on spending thresholds.
* Format notifications with clear, actionable information.
* Respect user preferences for enabling/disabling notifications or adjusting thresholds.
* Queue notifications if the user is offline and deliver them when they reconnect.

**Steps:**

1. Monitor Spending:

* The system continuously tracks categorized transactions (from Step 5) against budget goals (from Step 6).
* For each category, it calculates the percentage spent.

1. Trigger Approaching Limit Notification:

* When a category reaches the default threshold, the system prepares a notification.
* The Notification Module sends the alert as a push notification to the user’s device.

1. Trigger Exceeding Limit Notification:

* When a category exceeds 100% of the budget, the system sends a second notification.
* The notification includes a “View Details” option linking to the Budget section.

1. Customize Notification Settings:

* The user navigates to “Settings > Notifications” in the app.
* Options include:
* Enable/disable notifications per category.
* Adjust threshold.
* Turn off all budget alerts.
* The system saves changes and applies them immediately.

1. User Interaction:

* The user taps a notification to open the app directly to the Budget or Transactions section.
* The system updates spending data in real-time, ensuring notifications reflect the latest transactions.

**User Story: 8**

**Title:** User stories for reminding users about bill payments like electricity, gas, etc.

**Stakeholders:** Registered User, PFA-System, AI Module, Notification Module

**Stories:**

* As a Registered User, I want to:
* Receive reminders for upcoming bill payments before their due dates.
* See details in the reminder to plan my payments.
* Add or confirm recurring bills manually if the app doesn’t detect them automatically.
* Customize reminder timing.
* As the PFA-System Module, I want to:
* Store a list of the user’s recurring bills with details.
* Provide an interface for users to add, edit, or confirm bill details.
* Track payment status based on transaction data or user confirmation.
* Schedule reminders based on due dates and user preferences.
* As the AI Module, I want to:
* Identify recurring bills from transaction history.
* Predict due dates and amounts based on past patterns.
* Suggest new bills to add if consistent payments are detected.
* As the Notification Module, I want to:
* Send push notifications for bill reminders at the user-specified time.
* Include actionable details.
* Allow users to mark a bill as “Paid” directly from the notification.
* Queue reminders if the user is offline and deliver them when connected.

**Steps:**

1. Bill Detection and Setup:

* The AI analyses transaction history to detect recurring bills:
* The system prompts the user: “We detected a possible Electricity Bill. Add it?.”
* The user confirms or adjusts details.

1. Manual Bill Entry:

* The user navigates to “Bills” section and taps “Add Bill.”
* Fields include:
* Bill Type
* Amount
* Due Date
* The system saves the bill and schedules reminders.

1. Reminder Scheduling:

* The user sets reminder preferences in “Settings > Bill Reminders”:
* Default: 3 days before due date.
* Options: 1, 3, 5, or 7 days before.
* The system calculates the reminder date.

1. Send Notifications:

* The Notification Module sends a push alert
* Options in the notification: “Paid” or “View Details.”
* Tapping “Paid” updates the status; “View Details” opens the Bills section.

1. Status Tracking:

* The system checks transactions for payment confirmation.
* If paid, the bill is marked complete for the month; if not, a follow-up reminder is sent on the due date.

**User Story: 9**

**Title:** User stories for providing AI-generated insights and suggestions to improve spending habits.

**Stakeholders:** Registered User, PFA-System, AI Module

**Stories:**

* As a Registered User, I want to:
* Receive personalized insights about my spending patterns.
* Get specific suggestions to improve my finances.
* View these insights in a dedicated section of the app for easy access.
* Act on suggestions directly from the insights page.
* As the PFA-System Module, I want to:
* Display AI-generated insights and suggestions in a clear, user-friendly format.
* Provide navigation to related features from the insights page.
* Store historical insights for the user to review past advice.
* Update insights as new transaction data becomes available.
* As the AI Module, I want to:
* Analyse the user’s transaction history, budgets, and bill payments to identify spending trends.
* Generate actionable suggestions based on patterns.
* Prioritize insights based on impact.
* Refine suggestions over time as more data is collected or user preferences change.

**Steps:**

1. Insights Section:

* The user navigates to the “Insights” section from the app menu.
* The system loads a dashboard with a list of AI-generated insights and suggestions.

1. Generate Insights:

* The AI analyses:
* Transaction data
* Budget goals
* Bill payments and income patterns.

1. Provide Suggestions:

* The AI offers tailored recommendations:
* Each suggestion includes a “Take Action” button linking to relevant features.

1. Display and Interact:

* Insights are presented in a prioritized list:
* High-impact items at the top.
* Positive feedback included.
* The user taps “Take Action” to:
* Adjust a budget (redirects to Budget section).
* View transactions (redirects to Transactions section).

1. Update and Learn:

* The system refreshes insights daily or after significant new transactions.
* The AI tracks user actions to refine future suggestions.
* Historical insights are archived under a “Past Insights” tab.

**User Story: 10**

**Title:** User stories for setting financial goals and tracking progress.

**Stakeholders:** Registered User, PFA-System, AI Module

**Stories:**

* As a Registered User, I want to:
* Create financial goals with a target amount and deadline.
* Track my progress toward each goal with a visual indicator.
* See how much I need to save monthly to meet my goal.
* Receive updates or tips to stay on track with my goals.
* As the PFA-System Module, I want to:
* Provide an interface for users to add, edit, or delete financial goals.
* Display goal details in a dedicated “Goals” section.
* Calculate and show monthly savings needed based on the goal and timeline.
* Update progress automatically as savings or income data changes.
* As the AI Module, I want to:
* Suggest realistic goal amounts and timelines based on the user’s income and spending habits.
* Provide tips to accelerate progress.
* Detect when a goal is at risk and flag it for the user.
* Adjust suggestions if the user’s financial situation changes.

**Steps:**

1. Create a Financial Goal:

* The user navigates to the “Goals” section and taps “Add Goal.”
* Fields include:
* Goal Name
* Target Amount
* Deadline
* Optional: Link to a budget category

1. AI-Suggested Goals:

* The AI offers suggestions based on user data
* The user can accept the suggestion or customize it.

1. Calculate and Display Progress:

* The system calculates:
* Monthly savings needed
* Current progress
* The Goals section shows:
* Progress bar

1. Track and Update:

* The system updates progress as new savings transactions are detected.
* The user can manually input savings if not linked to a bank.

1. Receive Tips:

* The AI provides suggestions in the Goals section
* Tips link to Budget or Insights sections for action.

**User Story: 11**

**Title:** User stories for exporting financial data in PDF or Excel format for personal use.

**Stakeholders:** Registered User, PFA-System

**Stories:**

* As a Registered User, I want to:
* Export my financial data into a PDF or Excel file.
* Choose which data to include.
* Save the file to my device or share it.
* Get a clear, organized format that’s easy to read or analyse offline.
* As the PFA-System Module, I want to:
* Provide an export option in a dedicated “Export Data” section.
* Generate files in PDF and Excel formats with consistent, user-friendly layouts.
* Allow users to select specific data types and time periods for export.
* Ensure exported data is accurate and matches what’s shown in the app.

**Steps:**

1. Access Export Feature:

* The user navigates to the “Export Data” section from the app menu.
* The system displays options: “Export to PDF” and “Export to Excel.”

1. Select Data to Export:

* The user chooses what to include via checkboxes:
* Transaction History.
* Budget Summary (current month’s goals and spending).
* Goal Progress (all active goals).
* Financial Overview (balance, income, expenses).
* Default: All data for the current month.

1. Customize Time Period:

* The user selects a time range:
* Options: Last 7 days, Last 30 days, This Month, Custom (start/end date).
* The system validates the range.

1. Generate and Preview:

* The user taps “Export” and selects PDF or Excel.
* PDF format: A polished summary with tables and charts.
* Excel format: A spreadsheet with columns.
* A preview appears: “Your file is ready: Transactions\_Oct2023.pdf.”

1. Save or Share:

* The user taps “Download” to save to their device or “Share” to send via email, WhatsApp, or cloud storage.
* The system confirms: “File exported successfully.”

**User Story: 12**

**Title:** User stories for securely logging out and locking the account after use.

**Stakeholders:** Registered User, PFA-System

**Stories:**

* As a Registered User, I want to:
* Log out of the app easily when I’m finished to keep my financial data safe.
* Lock the app so no one can access it without my password or authentication.
* Be confident my account stays secure after I log out.
* Log back in quickly when I need to use it again.
* As the PFA-System Module, I want to:
* Provide a clear “Log Out” option in the app menu or settings.
* End the user’s session securely and lock access to all data.
* Require authentication to regain access.
* Clear any temporary data on logout for privacy.

**Steps:**

1. Initiate Logout:

* The user navigates to the app menu or settings and taps “Log Out.”
* A confirmation prompt appears: Are you sure you want to log out? with Yes/ No options.

1. Secure Session End:

* On “Yes,” the system:
* Ends the active session.
* Locks the app, returning to the login screen.
* Clears temporary data
* A message confirms: “You’ve been logged out successfully.”

1. Lock Mechanism:

* The app requires re-authentication to unlock:
* Password (set during registration, Step 1).
* Optional: Biometric if enabled.
* Without correct credentials, no access is granted.

1. Log Back In:

* The user enters their email/phone and password (or uses biometric) on the login screen.
* The system verifies credentials and restores access to their data.

**User Story: 13**

**Title:** User stories for deleting my account when I no longer want to use the app.

**Stakeholders:** Registered User, PFA-System

**Stories:**

* As a Registered User, I want to:
* Delete my account permanently if I’m done with the app.
* Confirm my choice to avoid accidental deletion.
* Know my data will be removed securely.
* Log out automatically after deletion.
* As the PFA-System Module, I want to:
* Provide a “Delete Account” option in settings.
* Require confirmation to prevent mistakes.
* Erase all user data from the system.
* Notify the user when deletion is complete.

**Steps:**

1. Initiate Deletion:

* User navigates to “Settings > Delete Account.”
* System shows a warning: “This will erase all your data. Continue?” with Yes/No.

1. Confirm Deletion:

* User enters their password to confirm.
* System verifies password matches registration.

1. Delete and Notify:

* System removes all user data.
* User sees: “Account deleted successfully” and is logged out.

1. Post-Deletion:

* App returns to the registration/login screen for new users.