

System Requirements Specification (SRS)

Smart Finance Tracker

Table of Contents

1. Introduction	2
1.1. Purpose	2
1.2. Scope	2
1.3. Abbreviations	2
1.4. Overview	3
2. Overall Description	3
2.1. Product Perspective	3
2.2. Product Features	3
2.3. User Characteristics	3
2.4. Operating Environment	4
2.5. Assumptions and Dependencies	4
3. Functional Requirements	4
3.1. User Authentication	4
3.2. Dashboard	5
3.3. Add Transaction	5
3.4. Transaction History	5
3.5. Budget Planner	6
3.6. Category Management	6
3.7. Reports & Analytics	6
3.8. Financial Goals	7
3.9. AI-Powered Features	7
3.10. Settings	7
3.11. Export Data	8
3.12. Error Handling and Session Management	8
4. Non-Functional Requirements	8
4.1. Performance	8
4.3. Usability	9
4.4. Reliability	9
4.5. Compatibility	9
4.6. Maintainability	9
5. System Architecture	9
5.1. Technology Stack	9
5.2. Database Design	10
5.3. System Flow	10
6. User Interface Requirements	10
6.1. Navigation	11
6.2. Forms	11
6.3. Visual Design	11
7. Implementation Constraints	11
7.1. Technical Constraints	11
7.2. Business Rules	12
8. Appendices	12
8.1. Page/Module List	12
8.2. Glossary	12

1. Introduction

1.1. Purpose

This document specifies the functional and non-functional requirements for the Smart Finance Tracker, a web-based personal finance management application. The system helps users track expenses, manage budgets, set savings goals, and analyze financial habits.

1.2. Scope

Smart Finance Tracker is a client-side web application that enables users to:

- Track income and expenses
- Create and manage budgets
- Set and monitor financial goals
- Receive AI-powered financial insights and recommendations
- Generate financial reports and analytics
- Export financial data summaries

The system is designed for personal finance management and will run on any modern web browser.

1.3. Abbreviations

SRS: System Requirements Specification

UI: User Interface

AI: Artificial Intelligence

CSV: Comma-Separated Values

API: Application Programming Interface

1.4. Overview

The remainder of this document provides a detailed description of the system's functionality, including user requirements, system features, and technical specifications.

2. Overall Description

This section presents a high-level overview of the Smart Finance Tracker system, describing its context, main features, target users, and operating environment.

2.1. Product Perspective

Smart Finance Tracker is a web-based application accessible through standard web browsers. Users interact with the system through a responsive web interface to manage their personal finances.

2.2. Product Features

The main features include:

- User authentication and profile management
- Transaction recording and management
- Budget planning and monitoring
- Financial goal tracking
- AI-Powered spending analysis
- AI-Powered expense categorization and advice
- Settings management
- Visual reports and analytics
- Data export capabilities
- Dashboard with overview

2.3. User Characteristics

Target Users:

- Individuals seeking to manage personal finances
- Basic computer and internet literacy

- No specialized financial or technical knowledge required

2.4. Operating Environment

- Frontend: HTML, CSS, JavaScript
- Backend: PHP, MySQL
- Browser Compatibility: Chrome, Firefox, Safari, Edge (latest versions)
- Devices: Desktop computers, tablets, smartphones

2.5. Assumptions and Dependencies

- Users have access to internet connection
- Users will provide valid and accurate data inputs
- Users have a modern web browser installed
- Server hosting is available for PHP and MySQL
- AI API service is accessible for intelligent features

3. Functional Requirements

This section describes the main functionalities of the Smart Finance Tracker, detailing what the system must do to meet user needs and business objectives.

3.1. User Authentication

Users can register, login, and manage their accounts.

Requirements:

- The system shall allow new users to register with email and password
- The system shall authenticate users during login
- The system shall maintain user sessions securely
- The system shall allow users to logout
- The system shall provide password reset functionality
- The system shall display a landing/introduction page before authentication
- The system shall provide forgot password functionality

3.2. Dashboard

Central overview page displaying key financial information.

Requirements:

- The system shall display total income for current month
- The system shall display total expenses for current month
- The system shall calculate and show current savings
- The system shall show recent transactions (last 3)
- The system shall display spending by category in chart form
- The system shall show budget status summary
- The system shall display AI-generated financial tips

3.3. Add Transaction

Interface for recording income and expense transactions.

Requirements:

- The system shall provide a form to add new transactions
- The system shall require: amount, category, date, and type (income/expense)
- The system shall allow optional description field
- The system shall validate all inputs before saving
- The system shall save transactions to database

The system shall provide success/error feedback to user

3.4. Transaction History

View and manage all recorded transactions.

Requirements:

- The system shall display all transactions
- The system shall allow filtering by date range, category, and type
- The system shall allow sorting by date, amount, or category
- The system shall enable editing of existing transactions
- The system shall allow deletion of transactions with confirmation
- The system shall show transaction count and totals

3.5. Budget Planner

Create and monitor budget limits for spending categories.

Requirements:

- The system shall allow users to set monthly budget limits per category
- The system shall display budget vs actual spending comparison
- The system shall calculate remaining budget for each category
- The system shall show visual progress bars for budget utilization
- The system shall alert users when approaching budget limits
- The system shall highlight categories exceeding budget

3.6. Category Management

Manage transaction categories during transaction creation.

Requirements:

- The system shall provide a dropdown of existing categories when adding transactions
- The system shall include a "+ Add New Category" option in the dropdown
- The system shall allow users to create new categories inline without leaving the transaction form
- The system shall provide default categories
- The system shall display spending totals per category in reports
- The system shall prevent deletion of categories with existing transactions

3.7. Reports & Analytics

Generate financial reports with visualizations.

Requirements:

- The system shall generate monthly spending reports
- The system shall display income vs expense charts
- The system shall show spending trends over time
- The system shall display category-wise breakdown in charts/graphs
- The system shall calculate savings rate
- The system shall show month-over-month comparisons

3.8. Financial Goals

Set and track savings goals.

Requirements:

- The system shall allow users to create savings goals with target amount and deadline
- The system shall display progress towards each goal
- The system shall calculate percentage completion
- The system shall allow multiple active goals
- The system shall show estimated completion date based on current savings rate
- The system shall allow editing and deletion of goals

3.9. AI-Powered Features

AI functionality integrated throughout the application.

Requirements:

- The system shall provide AI-generated financial tips on the dashboard
- The system shall offer AI budget suggestions in the Budget Planner
- The system shall provide AI insights in Reports & Analytics
- The system shall analyze user spending patterns
- The system shall identify unusual spending behavior
- The system shall suggest budget optimizations
- The system shall predict potential budget overruns
- The system shall offer savings recommendations

3.10. Settings

User preferences and account management.

Requirements:

- The system shall allow users to update profile information
- The system shall allow currency preference selection
- The system shall allow users to change password
- The system shall provide option to delete account
- Includes profile management.

3.11. Export Data

Export financial data in various formats.

Requirements:

- The system shall export transaction data to CSV format
- The system shall allow date range selection for export
- The system shall allow category filtering for export
- The system shall generate summary reports in PDF format
- The system shall include all relevant transaction details in exports

3.12. Error Handling and Session Management

Manage errors and user sessions.

Requirements:

- The system shall display a custom 404 error page for invalid URLs
- The system shall provide a session timeout page after 30 minutes of inactivity
- The system shall display a logout confirmation page
- The system shall redirect users to appropriate pages after session expiry
- The system shall provide clear navigation options from error pages

4. Non-Functional Requirements

This section defines the quality attributes and constraints of the system, specifying how the system should perform its functions.

4.1. Performance

- Page load time shall not exceed 3 seconds on stable network connection
- The system shall support at least 1000 transactions per user

4.2. Security

- User passwords shall be encrypted in the database
- The system shall validate all user inputs
- Sessions shall timeout after 30 minutes of inactivity
- Secure login system to prevent unauthorized access.

4.3. Usability

- The interface shall be intuitive and user-friendly
- Navigation shall be consistent across all pages
- Error messages shall be clear and helpful
- The system shall provide confirmation for destructive actions

4.4. Reliability

- The system shall maintain data integrity in case of power or connection loss
- The system shall handle errors gracefully without crashing
- Database backups shall be performed
- The system should be available 99% of the time, excluding maintenance

4.5. Compatibility

- The system shall work on Chrome, Firefox, Safari, and Edge browsers
- The system shall be responsive for desktop, tablet, and mobile devices
- The system shall support screen sizes from 375px to 1920px width

4.6. Maintainability

- Code shall be well-documented with comments
- The system shall follow coding best practices
- Database schema shall be normalized
- The system shall use modular code structure

5. System Architecture

This section outlines the technical architecture of the system, including technology choices, database structure, and system workflow.

5.1. Technology Stack

Frontend:

- HTML5 for structure

- CSS3 for styling
- JavaScript for interactivity and dynamic content

Backend:

- PHP for server-side logic
- MySQL for database management

Additional Libraries:

- Chart.js for data visualization
- AI API (OpenAI/Claude/Gemini) for intelligent features

5.2. Database Design

Primary Tables:

- users: User account information
- transactions: Income and expense records
- budgets: Budget limits per category
- goals: Savings goals
- categories: Transaction categories

5.3. System Flow

- User registers/logs in
- User accesses dashboard (overview)
- User adds/views transactions
- System processes and stores data
- System generates analytics and reports
- AI provides insights and recommendations
- User exports data as needed

6. User Interface Requirements

This section specifies the design and interaction requirements for the user interface to ensure consistency and usability.

6.1. Navigation

- Top navigation bar with links to all main pages
- Responsive hamburger menu for mobile devices
- Current page indicator in navigation
- Logout button accessible from all pages
- Landing page as the entry point for unauthenticated users
- Authentication flow: Landing → Sign In/Sign Up → Dashboard
- Session timeout redirects to logout page

6.2. Forms

- Clear labels for all input fields
- Inline validation with error messages
- Submit and cancel buttons
- Required field indicators (*)

6.3. Visual Design

- Clean and modern interface
- Consistent color scheme
- Appropriate use of icons
- Responsive layout for all screen sizes
- Charts and graphs for data visualization

7. Implementation Constraints

This section identifies technical and business limitations that must be considered during system development.

7.1. Technical Constraints

- Must use HTML, CSS, JavaScript for frontend
- Must use PHP and MySQL for backend
- Must integrate AI functionality
- Must have minimum 10 pages/sections

7.2. Business Rules

- Each user can only access their own data
- Transactions must have valid dates
- Budget amounts must be positive numbers
- Categories cannot be deleted if transactions exist

8. Appendices

Supplementary information including page lists and terminology definitions.

8.1. Page/Module List

- i. Landing Page
- ii. Sign In
- iii. Sign Up
- iv. Forgot Password
- v. Dashboard
- vi. Add Transaction (with inline category creation)
- vii. Transaction History
- viii. Budget Planner (includes AI suggestions)
- ix. Reports & Analytics (includes AI suggestions)
- x. Financial Goals
- xi. Settings
- xii. Export Data
- xiii. 404 Error Page
- xiv. Session Timeout/Logout Page

8.2. Glossary

Budget: A planned spending limit set for a specific category over a defined time period usually monthly.

Category: A classification system used to organize transactions into groups such as Transportation, Entertainment, Bills, etc.

Dashboard: The main overview page that displays key financial metrics, recent transactions, and summary information.

Expense: Money spent or costs incurred, recorded as a negative transaction in the system.

Export: The process of converting and downloading financial data from the system into external formats (CSV, PDF).

Goal: A target savings amount with a specified deadline that users set to achieve specific financial objectives.

Income: Money received or earned, recorded as a positive transaction in the system.

Savings Rate: The percentage of income that is not spent, calculated as $(\text{Income} - \text{Expenses}) / \text{Income} \times 100$.

Session: The period during which a user is actively logged into the system, typically ending after logout or timeout.

Transaction: A record of financial activity, either income or expense, including details such as amount, category, date, and description.

User Authentication: The process of verifying a user's identity through credentials (email and password) before granting access to the system.

Validation: The process of checking user input to ensure it meets required formats and business rules before processing.