**Technical Documentation**

**KaneFlow**

Table of Contents

[Introduction 3](#_Toc203771010)

[Purpose 3](#_Toc203771011)

[In today’s fast-paced and capitalistic economy, maintaining control over personal spending has become increasingly challenging. People often unintentionally overspend due to busy schedules, lack of organization, or simply forgetfulness. Existing expense tracking applications frequently come with limitations such as paid services, lack of updates, or insufficient personalization. Kaneflow addresses these issues by providing a fully-featured, personalized, and user-friendly financial tracking experience. With a streamlined and intuitive interface, personalized expense analysis, and intelligent AI-driven suggestions to optimize spending habits, Kaneflow delivers a seamless, accessible, and effective solution. This application is compatible across various platforms, including mobile devices, personal computers, and tablets, ensuring users have continuous and effortless access. 3](#_Toc203771012)

[Scope 3](#_Toc203771013)

[Architecture Overview 4](#_Toc203771014)

[Backend 4](#_Toc203771015)

[Frontend 4](#_Toc203771016)

[Database 4](#_Toc203771017)

[Infrastructure 4](#_Toc203771018)

[User Stories 4](#_Toc203771019)

[Development Roadmap 8](#_Toc203771020)

[Iteration 1 - User Identity & Core Layout 8](#_Toc203771021)

[Key Tasks Completed: 8](#_Toc203771022)

[Iteration 2 – Expense Management & Reporting 8](#_Toc203771023)

[Key Tasks Completed: 8](#_Toc203771024)

[Iteration 3 – Saving Goals, Visualization, and UX Polish 9](#_Toc203771025)

[Key Tasks Completed: 9](#_Toc203771026)

# Introduction

## Purpose

## In today’s fast-paced and capitalistic economy, maintaining control over personal spending has become increasingly challenging. People often unintentionally overspend due to busy schedules, lack of organization, or simply forgetfulness. Existing expense tracking applications frequently come with limitations such as paid services, lack of updates, or insufficient personalization. Kaneflow addresses these issues by providing a fully-featured, personalized, and user-friendly financial tracking experience. With a streamlined and intuitive interface, personalized expense analysis, and intelligent AI-driven suggestions to optimize spending habits, Kaneflow delivers a seamless, accessible, and effective solution. This application is compatible across various platforms, including mobile devices, personal computers, and tablets, ensuring users have continuous and effortless access.

## Scope

Kaneflow is specifically tailored to meet the financial tracking and management needs of individual users. Core features include detailed daily and monthly expense analysis, customizable categorical tags for better expense organization, personalized monthly savings budgets, and AI-driven recommendations aimed at optimizing user spending habits. The application's intuitive design and cross-platform compatibility ensure usability and convenience for users managing their personal finances, making their budgeting journey easier and more efficient.

# Architecture Overview

## Backend

Kaneflow’s backend is built using Node.js and Express.js, providing a robust and scalable server-side environment. Apollo Server is utilized to efficiently manage GraphQL APIs, allowing for clear and structured data exchanges between the frontend and backend. JWT is implemented for secure user authentication, while bcrypt enhances password security. The backend also incorporates scheduled tasks for automated monthly report generation and expense analytics, ensuring seamless and reliable operation.

## Frontend

The frontend leverages React to provide a dynamic and responsive user interface, with Apollo Client seamlessly integrating GraphQL queries and mutations for data fetching and state management. Vite is used as the build tool to enhance frontend development efficiency and performance. CSS Modules are adopted for scoped and maintainable styling, creating an organized and cohesive user interface that offers a seamless user experience across all platforms.

## Database

MongoDB is the chosen database solution, offering flexible and scalable data management for Kaneflow’s evolving schema needs. Its NoSQL nature allows for efficient storage and querying of data such as expenses, user profiles, and monthly financial reports. MongoDB’s robust querying capabilities provide fast and reliable access to user-specific financial data, ensuring quick and personalized responses to user actions.

## Infrastructure

Kaneflow’s infrastructure incorporates GitHub for version control and CI/CD integration. GitHub Actions facilitate automated testing and streamlined deployment workflows, improving development speed and reliability. Netlify is utilized for frontend deployment, providing a robust and scalable hosting solution that delivers continuous deployment capabilities with minimal downtime. Railway hosts the backend services, offering a straightforward and reliable cloud platform for seamless server deployment, scalability, and ease of management.

# User Stories

|  |  |  |
| --- | --- | --- |
| **User Story** | **Action** | **Acceptance Criteria** |
| As a user, I want to register with my personal information | * Navigate to Register page from Home page. * Input valid credentials in registration form. * Submit registration and verify email account. | * Registration form must validate email format and password length before submission * A success message is shown after submission, and a verification email is sent * Clicking the verification link marks the account as verified and allows login |
| As a user, I want to login to the app | * Navigate to the login page * Enter valid email and password * Submit the login form | * User is authenticated and redirected to dashboard * Invalid credentials show error message * JWT token is stored in localStorage |
| As a user, I want to be able to reset my password | * Enter registered email * Click on "Forgot Password" in login page * Click the reset link from email and enter a new password | * Reset email is sent with a working link * User can set a new password via reset form * User sees confirmation message after reset |
| As a user, I want to be able to see my personal info | * Navigate to profile page * View name, email, currency, and other saved preferences | * Profile page displays correct user data * Data matches backend user record |
| As a user, I want to be able to update my personal info | * Edit profile fields (name, currency, etc.) * Submit changes via save button | * Updated data is reflected in frontend and database * User sees a success message on update |
| As a user, I want to be able to delete my account | * Navigate to profile page * Click "Delete Account" and confirm deletion | * Account and related data are removed from backend * User is logged out and redirected to home page |
| As a user, I want to stay logged in and want to logout | * Remain signed in across page refresh * Click logout button from navbar or profile page | * JWT persists in localStorage until logout or expiry * Logout clears token and redirects to login or home * No access to protected routes after logout |
| As a user, I want to navigate between pages | * Click links or buttons in navbar * Use browser back/forward to navigate | * App uses client-side routing for page transitions * Active route is visually highlighted * Invalid routes show a 404 page or redirect |
| As a user, I want to add, edit, and delete my daily expense by name and category | * Fill in expense name, amount, category * Edit or delete items in expense list | * Added expenses appear instantly and persist after refresh * Editing updates the backend and UI * Deleting removes the record from UI and DB |
| As a user, I want to view the monthly total expense up to today | * Open dashboard or report view * View summary card or section showing current month's spending | * Total reflects sum of all expenses from 1st to current day * Updates automatically when new expense is added |
| As a user, I want to view expenses sorted by category, amount and date | * Use dropdown or filter to sort by category, amount or date * View sorted list or visual graph/chart | * Sorting applies instantly to the displayed data * Category, amount and date-based groupings are accurate * Default view is most recent date first |
| As a user, I want a monthly report of total expense, category-wise breakdown, savings, highest spending area, and comparisons to previous months. | * View the reports section on dashboard or Reports page * Select a specific month to review * Read breakdown chart, savings indicator, and comparison summaries | * Report shows total spent, savings, and category-wise breakdown * Displays top spending category * Includes comparison with previous month’s total and each category |
| As a user, I want to set a savings goal each month | * Navigate to the Savings page * Enter monthly savings target and optional thresholds per category * Submit the form to save goal | * Saving goal is stored and retrieved based on selected month * UI reflects active saving goal with progress bar * Existing goal can be updated or deleted |
| As a user, I want to view my progress toward the saving goal | * View dashboard or savings progress section * Compare total categorized spending against thresholds | * Progress bar visually indicates spent vs. threshold per category * Remaining savings amount is shown dynamically * Over-budget categories are visually highlighted |
| As a user, I want AI suggestions to optimize my spending habits | * Provide monthly income and savings target * Click "Get Suggestions" button * Review suggested budget distribution by category | * AI-generated suggestions follow internal logic rules (e.g., rent is fixed, entertainment is flexible) * Response includes a recommendation note and percentage adjustments * Suggestions are editable before saving as a goal |

# Development Roadmap

The product was developed using 3 consecutive iterations.

## Iteration 1 - User Identity & Core Layout

The first development iteration focused on laying the foundational architecture for both backend and frontend, establishing secure user identity management, and building the navigation framework for the application.

### Key Tasks Completed:

* **User Authentication:**

Implemented user registration, login, and password reset functionalities using secure JWT-based authentication. Validation and error handling were added to ensure a smooth user experience.

* **Layout Configuration:**

Designed the initial app layout including navigation menus, route protection for authenticated pages, and UI consistency across views.

* **Personal Information:**

Developed the user profile page allowing users to view, update, or delete their account and personal details. Profile updates include name and preferred currency.

## Iteration 2 – Expense Management & Reporting

This iteration focused on implementing the core financial tracking features, including detailed expense management and monthly financial reporting. It emphasized usability, real-time updates, and data clarity to help users better understand and manage their finances.

### Key Tasks Completed:

* **Expense Tracking (CRUD):**  
  Implemented the ability for users to add, edit, and delete daily expenses by category and description. This included input validation, persistent backend storage, and real-time UI updates upon change.
* **Monthly Spending Summary:**  
  Displayed the user’s total monthly expenses up to the current day. This feature dynamically updates with new entries and reflects the ongoing financial snapshot.
* **Sorting & Filtering:**  
  Enabled users to sort expenses by category and date, helping them analyze trends and identify high-spending areas more clearly.
* **Detailed Monthly Report:**  
  Generated comprehensive reports for each month, showing:
  + Total spending
  + Category-wise breakdown
  + Savings performance
  + Highest spending area
  + Comparisons to previous months

## Iteration 3 – Saving Goals, Visualization, and UX Polish

The third and final development iteration focused on advanced financial visualization, AI-assisted budgeting, and overall polish to the application’s user interface and responsiveness. This iteration delivered a complete and insightful budgeting experience for users.

### Key Tasks Completed:

* **Savings Goals Setup & AI Suggestions:**  
  Integrated a backend system for saving goals, including monthly targets and category-based thresholds. Implemented an AI-based suggestion engine that proposes optimized spending caps based on the user’s income and habits.
* **Spending vs. Threshold Display:**  
  Built a dynamic dashboard module that visually compares actual spending against user-defined thresholds, highlighting overspending areas and remaining savings.
* **Visual Analytics & Charts:**
  + **Categorical Spending Breakdown:** Bar and pie charts showing category-wise expenses for current month.
  + **Last 5 Months Trend Chart:** Combined bar and line chart to display spending patterns across the last five months.
  + **Savings Page Donut Chart:** Displayed remaining vs. used budget per category, with central text showing leftover savings.
* **Dashboard Enhancements:**  
  Added responsive cards showing:
  + Daily and monthly spending totals
  + Top expense categories
  + Highest single spending entry
* **UI/UX Styling & Responsiveness:**  
  Polished all styles using CSS modules and media queries. Improved mobile-first responsiveness across all views.
* **Landing Page Updates:**  
  Refreshed landing page with clear CTAs and minimal design. Improved accessibility and responsiveness.
* **Demo Account Functionality:**  
  Added support for demo accounts to let users explore the app without registration.
* **Final Bug Fixes & Polish:**  
  Wrapped up remaining backend logic, validations, loading spinners, and edge case handling to prepare for public release.