Functional and Non-Functional Requirements

Project Title: SpendSmart

Prepared by: Aashin K S & Avula Venumadhav

1 Goal

SpendSmart is a secure, full-stack web application that empowers users to manage their personal finances by automatically importing bank transactions, categorizing expenses, setting budgets, tracking financial goals, and interacting with an AI-powered financial chatbot. The system integrates with banking APIs and offers real-time insights and personalized financial advice. Admins oversee user management and platform health to ensure system security, performance, and compliance.

2 Functional Requirements

- User registration and login with role-based access (User/Admin)
- Secure bank account linking via external APIs (e.g., Plaid)
- Automated transaction fetching and intelligent expense categorization
- Budget planning module with customizable limits per category
- Spending dashboard with charts, summaries, and monthly/yearly reports
- Savings goal creation and progress tracking modules
- AI-powered chatbot for financial advice using Ollama LLM
- Email notifications (e.g., budget limit warning, goal achievements) using Nodemailer
- Admin dashboard to monitor users, platform usage, and system activities
- Role management (admin can promote, suspend, or delete users)
- API Gateway management with Express Gateway for service routing, load balancing, and basic security policies

3 Non-Functional Requirements

- Secure authentication using JWT tokens and OAuth for bank API access
- Data encryption (both in transit via HTTPS and at rest using database encryption techniques)
- Responsive and intuitive UI/UX optimized for web and mobile browsers
- Scalable microservices backend (Node.js, Express.js) to handle growing transaction data and user load
- Real-time synchronization with banking APIs for transaction updates
- High availability and low-latency backend API responses
- Fault tolerance and graceful error handling for API and system failures
- Compliance with data privacy laws (e.g., GDPR, CCPA)
- Audit trails and logging for security incidents and critical activities
- Modular architecture for easier maintenance, upgrades, and feature expansion

4 Pain Points for Users and Admins

Users:

- Difficulty tracking spending habits manually
- Lack of insights into budgeting and savings opportunities
- Inability to receive quick financial guidance when needed
- Confusion managing multiple accounts and goals in one place

Admins:

- Maintaining security over sensitive financial data
- Managing user accounts, suspicious activities, and transaction monitoring
- Monitoring API call usage and system uptime
- Ensuring legal compliance with evolving financial regulations

5 How SpendSmart Solves These Problems

- Automatic bank transaction import and categorization for seamless tracking
- Visual budgeting tools and real-time alerts to prevent overspending
- AI-powered financial chatbot (using Ollama) offering instant guidance
- Centralized dashboard combining expense reports, goal tracking, and financial health overview
- Secure admin controls for better user and system management
- Microservices and gateway setup ensures modular, scalable, and future-proof architecture