Technical Documentation for MicroLend

## System Architecture and Design

The system architecture for MicroLend is designed to ensure efficiency, scalability, and seamless integration of its various components. It follows a modular approach, breaking down the system into distinct layers:

* **Frontend**: A user-friendly interface designed to provide intuitive navigation and real-time interaction. Technologies used include [Insert technologies] for the UI and [framework/library] for front-end logic.
* **Backend**: The backend is structured around REST APIs that manage the business logic, data processing, and communication with external services. [Insert server-side framework] handles the requests and provides endpoints for the frontend.
* **Database Layer**: The app uses [Insert database technology] for storing user data, financial records, and transaction history.
* **Cloud Infrastructure**: The system is hosted on Vultr Cloud , leveraging services such as scalable hosting, managed databases, and a content delivery network (CDN) for optimizing the app's performance and ensuring reliability during high traffic loads.
* **Security Protocols**: To protect user data, [Insert encryption standards] are employed for data in transit and at rest, along with authentication mechanisms like OAuth2 for user login and access control.
* **APIs and Integrations**: The platform integrates third-party financial services, such as [Insert integration examples], to provide users with real-time financial insights and seamless transaction processing.

## Explanation of Key Components and Modules

* **Frontend**: Responsible for the user experience, the frontend interacts with the backend via API calls. The UI components include a dashboard for financial tracking, loan application management, and personalized advisory services. These are built using [Insert framework].
* **Backend**: The core logic is handled by the backend. Key modules include:
  + **User Management Module**: Handles registration, login, and user profiles.
  + **Financial Analysis Module:** Processes user financial data and generates personalized insights.
  + **Loan Processing Module**: Facilitates loan applications, approval workflows, and transaction management.
  + **Budget Tracker Module**: Tracks income, expenses, and budgets, with real-time alerts and analytics.
* **Database**: Designed with [Insert schema or NoSQL model], the database stores user profiles, loan records, transaction history, and budgeting data. Relationships between entities are optimized for fast queries and scalability.
* **API Gateway**: Provides secure access to the backend services. It also ensures that all external APIs are securely integrated into the platform.
* **Notification System**: An integrated notification service sends real-time updates to users about their financial status, loan progress, and budgeting alerts. The service is built using [Insert notification service/tool].

## API Documentation

* User Management API
  + Registers a new user
  + Authenticates a user
  + Retrieves user profile information
* Loan Management API
  + Submits a loan application
  + Fetches the status of a submitted loan
* Budget Tracker API
  + Adds a new expense
  + Retrieves budget summary and analytics.
* Financial Advisory API
  + Provides personalized financial advice based on user data.

## Set-Up and Usage Instruction

* Pre-requisites
  + [Insert technology/tools] installed.
  + Vultr Cloud account configured.
  + [Insert API keys] for third-party integrations
* Set-Up Instructions
  + Clone the Repository
  + Install Dependencies
  + Configure Environment
    - Create a .env file in the root directory and populate it with the necessary credentials
  + Run the Application
* Usage
  + **Access the Dashboard**: Open [instructions for starting the app].
  + **Login/Signup**: Use the signup form to create an account or log in using your credentials.
  + **Manage Finances**: Navigate through the dashboard to track expenses, apply for loans, or receive personalized advice.