

FunBank - Personal Financial Planning Tool

Team Members: [Your Name]

Description

FunBank is a comprehensive personal financial planning tool that combines traditional financial management features with AI-powered insights. The application helps users track their finances, set and achieve financial goals, and receive personalized financial advice through an AI chat interface.

Section 1: Introduction

Purpose of Project

The purpose of FunBank is to provide users with a comprehensive financial management solution that combines traditional financial tracking with modern AI capabilities. The system aims to help users:

- Track and manage their financial accounts
- Set and achieve financial goals
- Receive personalized financial advice
- Make informed financial decisions
- Understand their spending patterns
- Plan for their financial future

Section 2a: Must Have Requirements

1. User Authentication and Account Management

- Success Measure: Users can register, login, and manage their profile
- Demonstration:
 - Create a new account with email and password
 - Login with credentials
 - Update profile information
 - Change password
 - Logout functionality

2. Account and Transaction Management

- Success Measure: Users can add, edit, and track financial accounts and transactions
- Demonstration:
 - Add a new bank account with balance
 - Record a transaction
 - Edit transaction details
 - View transaction history
 - Delete a transaction

3. Basic Budget Management

- Success Measure: Users can create and track monthly budgets
- Demonstration:
 - Create a monthly budget
 - Set spending limits by category
 - Track actual spending vs budget
 - View budget progress
 - Receive basic budget alerts

4. Financial Dashboard

- Success Measure: Users can view their financial overview
- Demonstration:
 - View total account balances
 - See recent transactions
 - View basic spending categories
 - Check budget status
 - View simple financial metrics

5. AI Chat Interface

- Success Measure: Users can interact with AI for basic financial advice
- Demonstration:
 - Ask financial questions
 - Receive relevant responses
 - View chat history
 - Get basic financial tips
 - Receive simple recommendations

6. Data Security and Privacy

- Success Measure: User data is securely stored and transmitted
- Demonstration:
 - Secure login process
 - Encrypted data transmission
 - Protected API endpoints
 - Secure password storage
 - Session management

Section 2b: Stretch Requirements

1. Advanced Investment Tracking

- Success Measure: Users can track and analyze investment portfolios
- Demonstration:
 - Add investment accounts
 - Track portfolio performance
 - View asset allocation
 - Analyze investment returns
 - Get investment recommendations

2. Debt Management System

- Success Measure: Users can track and manage their debts
- Demonstration:
 - Add different types of debts
 - Track payment schedules
 - View debt payoff strategies
 - Calculate interest savings
 - Monitor debt-to-income ratio

3. Advanced Analytics and Reporting

- Success Measure: Users can access detailed financial analytics
- Demonstration:
 - Generate custom reports
 - View spending trends
 - Analyze cash flow
 - Track net worth over time
 - Export financial data

4. Goal Planning and Tracking

- Success Measure: Users can set and track financial goals
- Demonstration:
 - Create financial goals
 - Set target amounts and deadlines
 - Track goal progress
 - Receive goal recommendations
 - View goal completion predictions

Section 3: Overview of the Product

Workflow

1. User Registration/Login

- User creates account or logs in
- System authenticates and creates session
- User accesses dashboard

2. Account Management

- User adds financial accounts
- System stores account information
- User can view and manage accounts

3. Transaction Management

- User records transactions
- System categorizes and stores transactions
- User can view and analyze transactions

4. Budget Management

- User creates budgets
- System tracks spending against budgets
- User receives budget updates

5. AI Interaction

- User asks financial questions
- System processes and responds
- User receives personalized advice

Resources

- Frontend: React with TypeScript, Material UI
- Backend: Spring Boot, Java
- Database: Relational Database (Spring Data JPA)
- AI: OpenAI API
- Authentication: JWT
- Testing: Jest, JUnit
- Version Control: Git
- CI/CD: GitHub Actions

Data at Rest

- User Profile Data: Relational database tables
- Financial Data: Relational database tables
- Chat History: Relational database tables
- Analytics Data: Relational database tables

Data on the Wire

- RESTful API endpoints
- WebSocket for real-time updates
- JWT for authentication
- HTTPS for secure transmission

Data State

<https://www.mermaidchart.com/app/projects/3f6f741a-aca7-40cf-8b54-76957828a1c6/diagrams/c76b18fd-be03-4926-b94a-9c7e81e1c09f/version/v0.1/edit>

HMI/HCI/GUI

This was submitted with it. That was a nonfunctional idea of what I want the dashboard to look like using tailwind and react and after having the backend api's completely ready so that users can edit their data from any computer and save their financial history. The dashboard will possibly include a few charts, but I am not sure yet. I don't know if it will be a circle chart with a circular bar showing spent vs. goals or something similar.

Section 4: Verification

Demo

1. User Authentication Demo

- Register new account
- Login with credentials
- Update profile
- Logout

2. Financial Management Demo

- Add accounts
- Record transactions
- Create budget
- View dashboard

3. AI Interaction Demo

- Ask financial questions
- View responses
- Check chat history

Testing

1. Unit Testing

- Frontend component tests
- Backend service tests
- API endpoint tests

2. Integration Testing

- User flow tests
- API integration tests
- Database integration tests

3. End-to-End Testing

- Complete user journeys
- Critical path testing
- Error scenario testing

Sources/Citations/Resources

- Spring Boot Documentation: <https://spring.io/projects/spring-boot>
- React Documentation: <https://reactjs.org/docs/getting-started.html>
- Material UI Documentation: <https://mui.com/>
- OpenAI API Documentation: <https://platform.openai.com/docs/>
- Spring Data JPA Documentation: <https://spring.io/projects/spring-data-jpa>
- JWT Documentation: <https://jwt.io/introduction>