UIT2739 – FULL STACK DEVELOPMENT

A PROJECT REPORT

on

Financial Application

Submitted by

Lewin Jesudhas H	**3122 22 5002 064**
Paranthagan S	**3122 22 5002 091**
Pranaav U	**3122 22 5002 093**

Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

NOVEMBER 2025

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

Department of Information Technology

CERTIFICATE

Certified that this project titled **“Financial Application"** is the bonafide work of “**Lewin Jesudhas H (3122 22 5002 064), Paranthagan S (3122 22 5002 091) and Pranaav U (3122 22 5002 093)** is submitted for project review on 26th of November 2025.

Place: Kalavakkam

Date: 26/11/2025 **Internal Examiner**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The rapid growth of digital banking and online financial services has transformed how individuals manage their money, yet many users continue to struggle with essential aspects of financial literacy. Tracking expenses, identifying spending patterns, and planning meaningful savings goals are often challenging without structured tools and clear insights. Additionally, investment management has become increasingly complex due to fluctuating market conditions, multiple trading platforms, and an abundance of unfiltered data. Traditional financial dashboards typically offer basic account balances without deeper analysis, automation, or personalized recommendations. As a result, individuals are left without the guidance or real-time data needed to make informed decisions, often leading to financial inconsistencies and unoptimized investments. To address these challenges, there is a need for an intelligent, secure, and user-friendly platform that simplifies financial understanding while enabling users to engage confidently with their personal finance data.

The growing demand for smart financial tools emphasizes the importance of systems that blend automation, analytics, and security. Users today expect applications that not only present raw financial information but also interpret it into actionable insights that support budgeting, investment, and long-term financial planning. Similarly, administrators and financial advisors require efficient systems to monitor client portfolios, evaluate risks, analyze market behaviour, and provide personalized advice based on real-time trends. However, most existing solutions are either too technical, expensive, or fragmented across multiple services. The absence of a consolidated environment for automated data summarization, intelligent guidance, and real-time portfolio monitoring creates gaps in both personal and professional financial management. A unified platform capable of integrating these components can bridge the gap and enhance decision-making for all stakeholders.

The Financial Services Dashboard project is developed to respond to this need by providing customers and administrators with a single integrated solution that simplifies financial operations and amplifies financial awareness. The platform supports secure user authentication using TOTP-based Two-Factor Verification, ensuring safety over sensitive information. It includes modules such as an Automated Financial Summarizer, which processes uploaded financial documents and extracts insights through visual analytics, an AI-powered Fin Chatbot for interactive assistance, and a Portfolio Analyzer that provides admins with real-time stock tracking and performance evaluation. With a responsive and visually rich interface, the system delivers an accessible and comprehensive experience, supporting better financial planning and management. Ultimately, the platform empowers users with clarity, transparency, and confidence, promoting financial discipline and improving long-term decision-making.

1.1 Problem Statement

Financial literacy and investment awareness remain major challenges for many individuals across different age groups and income levels. Customers often struggle to systematically track their day-to-day expenses, understand where their money is going, and analyze long-term financial patterns. At the same time, making informed investment decisions is becoming increasingly complex due to market volatility and the need to interpret real-time data from multiple sources. Existing financial dashboards are often fragmented, generic, or limited to basic balance views, lacking personalization, automation, strong security, and advanced analytical capabilities. Administrators and financial service providers also require a centralized system that can help them monitor client portfolios, evaluate risk, and support better advisory decisions. Without such tools, both customers and admins face inefficiencies, guesswork, and potential financial losses. Therefore, there is a clear need for a unified, secure, and user-friendly platform that combines real-time portfolio analytics, automated financial document summarization, and intelligent assistance. The proposed Financial Services Dashboard addresses this gap by integrating secure authentication, document-driven analysis, and portfolio management features into a single, cohesive web application. By delivering actionable insights, reducing manual effort, and guiding users through clear visualizations, the system aims to improve decision-making, strengthen financial discipline, and encourage long-term wealth creation for its users.

1.2 Objectives

The main objective of this project is to design and implement a secure, role-based Financial Services Dashboard that serves both customers and administrators. The system aims to provide a protected authentication layer using TOTP-based Two-Factor Authentication to safeguard user accounts and sensitive data. It will offer automated financial summarization by allowing users to upload CSV, Excel, or text files, from which income, expenses, trends, and categories are extracted and visualized. Another objective is to deliver a real-time portfolio analyzer for admins, enabling them to create and manage stock positions, track watchlists, and view gains or losses based on live price feeds. The platform also integrates an AI-driven financial chatbot to answer user queries and support financial awareness. Finally, the system focuses on delivering an intuitive, responsive, and visually rich interface that supports data-driven decision-making for both user roles. These objectives collectively ensure that the application is practical, reliable, and user centric.

1.3 Scope

The scope of this project is limited to building a web-based Financial Services Dashboard that supports both customer-level tools and admin-level portfolio management. For customers, the system covers uploading and analyzing financial documents in CSV, Excel, or text formats to derive income–expense summaries, monthly trends, and category-wise spending insights. It includes access to an AI-based financial chatbot and a financial simulator that helps users understand savings growth and investment outcomes over time. For admins, the scope includes creating and managing equity positions, maintaining a watchlist of symbols, and viewing real-time portfolio valuations and gains or losses using live market price feeds. Security features such as password hashing, TOTP-based 2FA, and role-based routing are also part of the scope. External integrations are restricted to email delivery and market data APIs; core banking transactions, loans, and regulatory compliance workflows are outside the current scope. Future extensions can later expand into loans, insurance, and credit-scoring.

1.4 Motivation

In the current digital economy, individuals are exposed to countless financial products, payment methods, and investment channels, yet many still lack clarity, structure, and confidence in managing their money. Users frequently rely on manual tracking, scattered apps, or guesswork instead of data-driven planning, which can lead to overspending, missed opportunities, and poor investment choices. At the same time, financial service providers and admins need tools that help them understand client portfolios quickly and advise more effectively. This project is motivated by the desire to simplify these challenges through a single, secure, and intelligent dashboard that combines automation, analytics, and guidance. By turning raw transaction data and portfolio information into understandable visuals and summaries, the system enables users to learn from their own spending and investing behaviour. Ultimately, the goal is to promote financial discipline, transparency, and long-term wealth creation in a way that feels modern and approachable.

CHAPTER 2

REQUIREMENTS SPECIFICATION

This chapter outlines the essential requirements necessary for the successful development and operation of the Financial Application. It explains what the system must perform functionally, how it should behave under various conditions, who the intended users are, and the different interfaces they will interact with. These requirements form the foundation for system design, architecture decisions, implementation strategy, and validation. Clearly defining requirements ensures that the system meets user expectations, supports secure and reliable financial data handling, and delivers an efficient user experience. By analyzing the needs of customers and administrators, the specification highlights the core modules such as authentication, financial summarization, AI-based support, and real-time portfolio management. This chapter provides a structured overview of both functional and non-functional requirements, use cases, user roles, and UI screens that collectively guide the development and evaluation of the financial application.

2.1 Functional Requirements

Functional requirements define the core features and operations that the system must perform to meet user needs. These requirements describe what the system should do and form the basis of the application's primary functionality. The major functional requirements are:

1. Authentication

The authentication module manages secure user login and registration. It includes password hashing, username/ email validation, and TOTP-based Two-Factor Authentication (2FA) to ensure secure access. Users must verify their identity with a 6-digit code from an authenticator app, adding an extra security layer to prevent unauthorized access. This feature ensures that only legitimate users can access sensitive financial and portfolio information.

2. Role Management

Role management determines permission levels and access controls for different types of users. The system supports two roles like Customer and Admin each provided with specific capabilities. Customers can use tools like the Financial Summarizer, Simulator, and Fin Chatbot, while Admin users gain access to additional modules such as the Portfolio Analyzer and Watchlist management. This ensures personalized dashboards and prevents privilege misuse.

3. Financial Summarizer

This module allows users to upload financial documents in CSV, Excel, or Text formats. The system automatically extracts transactions, categorizes income and expenses, identifies trends by month or category, and generates charts for convenient visualization. It transforms raw financial data into meaningful insights, helping users track and analyze their spending habits efficiently.

4. Fin Chatbot

The Fin Chatbot offers an AI-powered conversational interface that responds to financial-related queries. It provides advice, definitions, budgeting suggestions, and investment knowledge, available 24/7. This feature enables users to gain quick guidance without consulting financial experts, improving financial understanding and decision-making.

5. Financial Simulator

The Financial Simulator predicts future savings and investment outcomes. Users can input variables like initial amount, monthly savings, interest rate, or investment tenure to model financial growth. This helps users understand long-term planning, compare scenarios, and set realistic financial goals.

6. Portfolio Analyzer

The Portfolio Analyzer enables admins to manage stock market positions by adding, editing, or removing holdings. It facilitates real-time stock price tracking, gain/loss calculations, and watchlist monitoring. This provides a clear overview of investment performance and helps in making informed decisions based on market trends.

7. File Management

File management supports secure uploading and storing of financial documents. The system allows drag-and-drop upload functionality and validates format and size before processing. All uploaded documents are stored safely, and access is restricted based on user sessions, ensuring data protection.

8. Transactions View

The Transactions View displays recent uploaded or database-stored financial entries in a structured format. It allows users to review individual transactions, amounts, timestamps, and categorizations. Visualization charts and summaries enhance user understanding and support efficient financial monitoring.

2.2 Non-Functional Requirements

Non-functional requirements describe system quality attributes, defining how the system should work rather than what it should do. These requirements focus on performance, security, usability, and reliability. The major non functional requirements are:

1. Performance

Performance refers to how efficiently the system responds to user actions and processes data. The application ensures fast execution, with dashboard load time under 3 seconds and near-instant rendering of charts and tables to maintain a smooth user experience.

2. Security

Security ensures the protection of sensitive financial and personal data. This system includes password hashing using Bcrypt, TOTP-based 2FA for identity verification, and secure session handling. Input validation prevents attacks such as SQL injection and XSS, ensuring safe system operation.

3. Usability

Usability ensures that the application is easy to learn and navigate. The system features a responsive UI, clear layout, organized menus, and visual charts that present insights clearly. Smooth navigation helps users complete tasks quickly without confusion.

4. Reliability

Reliability refers to consistent system behavior without failures. The dashboard maintains stable performance through robust error handling and database integrity controls, ensuring that users can rely on the system even during continuous usage.

5. Scalability

Scalability ensures that the system can handle growing numbers of users, transactions, uploads, and stock price feeds without reduced performance. The modular design allows easy integration of future features like loan management or ML prediction without redesigning the entire system.

6. Maintainability

Maintainability refers to how easily the system can be updated or improved. The modular code structure, service-based components, and reusable functions allow developers to fix bugs or enhance features without affecting other components.

2.3 Use Cases

Use cases describe how different users interact with the system to accomplish specific goals. They reflect real-world workflow sequences that demonstrate the operations performed by each role within the Financial Services Dashboard. The main use cases are:

Customer Use Cases:

❑ Register by entering username, email, password, and selecting the customer role.

- ❑ Log into the system using valid credentials.
- ❑ Verify identity using a 6-digit TOTP code from an authenticator app.
- ❑ Upload financial files (CSV, Excel, or Text) for automated analysis.
- ❑ View summarized results including income, expenses, category-wise spending, and monthly trends.
- ❑ Interact with the Fin Chatbot to receive instant financial guidance and clarifications.
- ❑ Use the Financial Simulator to model savings growth and investment performance.
- ❑ Review recent transactions and insights displayed visually on the dashboard.
- ❑ Log out securely after completing activities.

Admin Use Cases:

- ❑ Log in using secure credentials and complete TOTP verification.
- ❑ Access the Admin Portfolio Dashboard with enhanced privileges.
- ❑ Add, update, or remove stock positions to manage investment data.
- ❑ Add stock symbols to the watchlist and monitor price changes in real time.
- ❑ View gain/loss calculations and overall portfolio performance analytics.
- ❑ Track live market prices through the integrated Price Feed Service.
- ❑ Manage uploaded customer financial records indirectly through insights visualization.
- ❑ Delete or reset TOTP database when necessary for maintenance.
- ❑ Log out to end the secure session.

2.4 User Roles

User roles define specific access levels and permissions within the Financial Services Dashboard. Each role controls what features a user is allowed to access, ensuring secure and structured system usage.

Customer Role

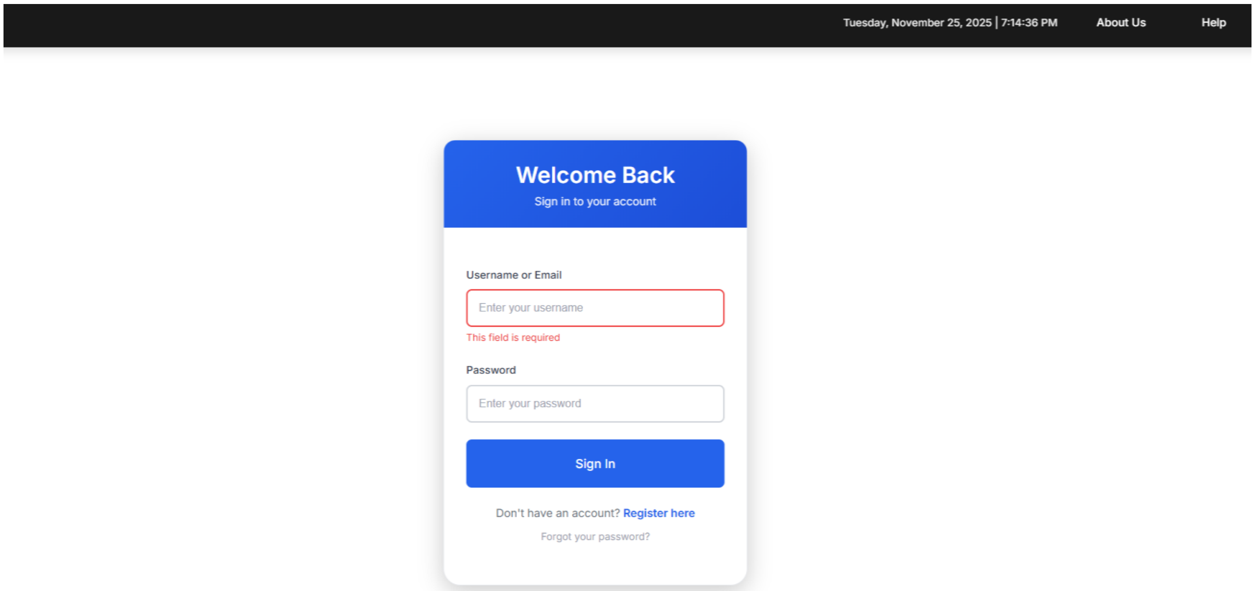
- ❑ Can register and log in using TOTP-based authentication.
- ❑ Can access the customer dashboard after successful login.
- ❑ Can upload financial documents (CSV, Excel, Text) for automatic analysis.
- ❑ Can view financial summaries such as income, expenses, category distribution, and monthly trends.
- ❑ Can use the Fin Chatbot for financial guidance and queries.
- ❑ Can interact with the Financial Simulator to model savings and investment scenarios.
- ❑ Can review recent transactions and analysis results visually.
- ❑ Can manage personal session data and logout securely.

Admin Role

- ❑ Can log in and authenticate using TOTP verification with elevated access.
- ❑ Can access the Admin Portfolio Dashboard with advanced management features.
- ❑ Can add, edit, and remove stock positions from the portfolio.
- ❑ Can manage the Watchlist, including adding and removing stock symbols.
- ❑ Can monitor real-time stock prices using the Price Feed Service.
- ❑ Can view portfolio gain/loss calculations and market analytics.
- ❑ Can access and perform maintenance actions, such as TOTP database reset.
- ❑ Can manage session controls and logout securely.

2.5 Screens / UI Overview

- ❑ **Login / Register / TOTP Verification**



Tuesday, November 25, 2025 | 7:14:36 PM

Welcome Back

Sign in to your account

Username or Email

Enter your username

This field is required

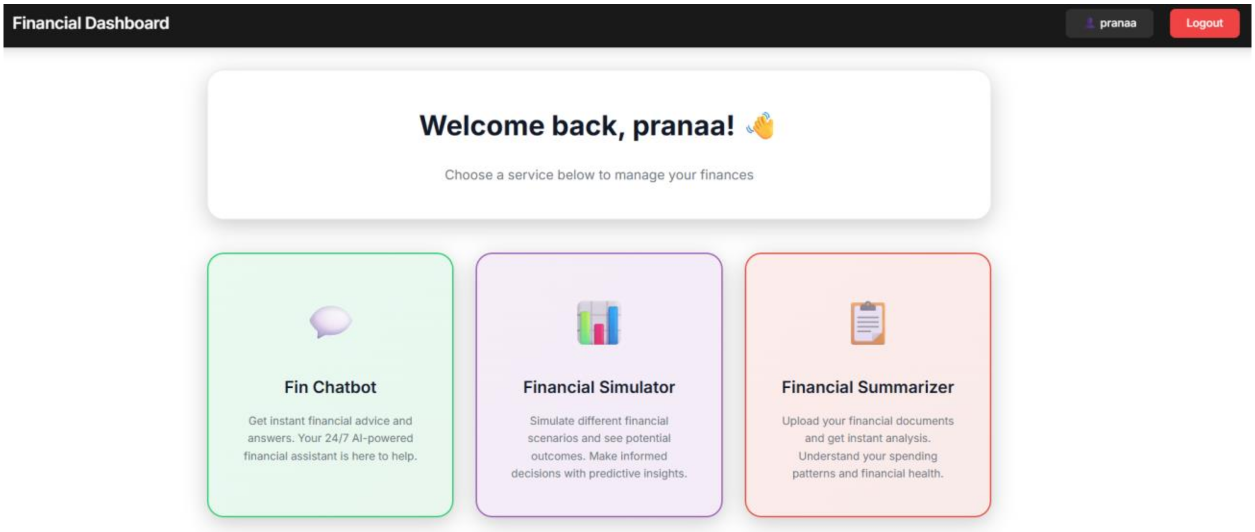
Password

Enter your password

Don't have an account? Register here

Forgot your password?

❑ Customer Dashboard



Welcome back, pranaa! 👋

Choose a service below to manage your finances

Fin Chatbot

Get instant financial advice and answers. Your 24/7 AI-powered financial assistant is here to help.

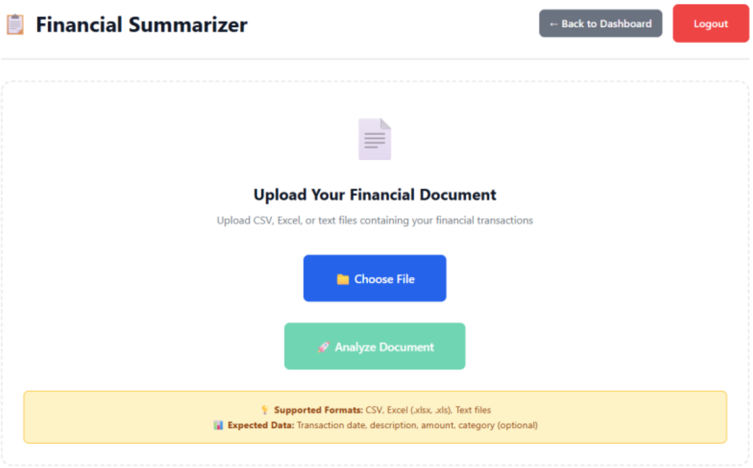
Financial Simulator

Simulate different financial scenarios and see potential outcomes. Make informed decisions with predictive insights.

Financial Summarizer

Upload your financial documents and get instant analysis. Understand your spending patterns and financial health.

❑ Expense Insights



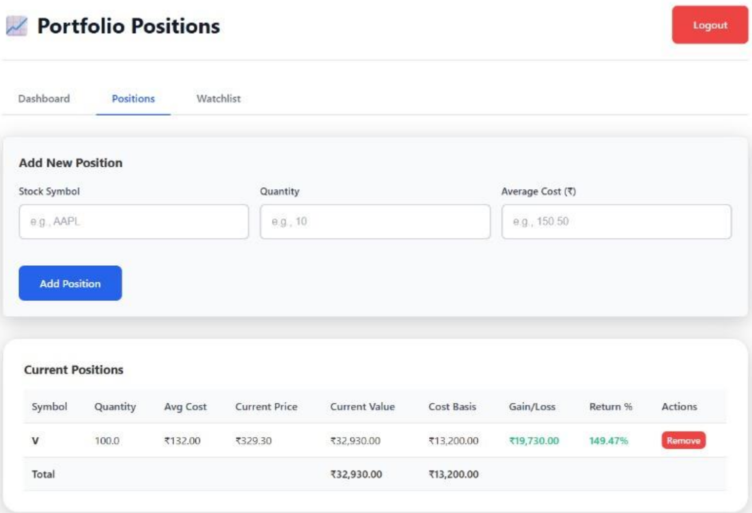
📋 Financial Summarizer

Upload Your Financial Document

Upload CSV, Excel, or text files containing your financial transactions

💡 **Supported Formats:** CSV, Excel (.xlsx, .xls), Text files
📊 **Expected Data:** Transaction date, description, amount, category (optional)

❑ Portfolio Position



📈 Portfolio Positions

Add New Position

Stock Symbol: e.g., AAPL

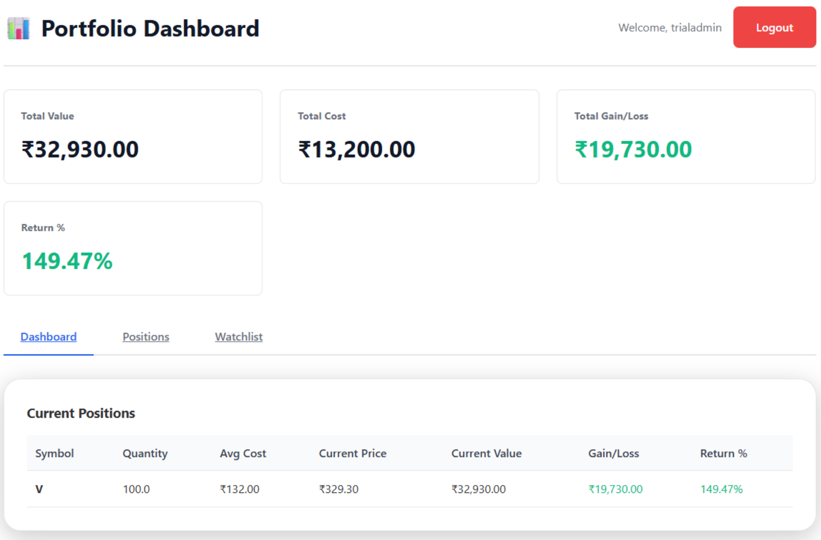
Quantity: e.g., 10

Average Cost (₹): e.g., 150.50

Current Positions

Symbol	Quantity	Avg Cost	Current Price	Current Value	Cost Basis	Gain/Loss	Return %	Actions
V	100.0	₹132.00	₹329.30	₹32,930.00	₹13,200.00	₹19,730.00	149.47%	Remove
Total				₹32,930.00	₹13,200.00			

❑ Admin Panel



Total Value
₹32,930.00

Total Cost
₹13,200.00

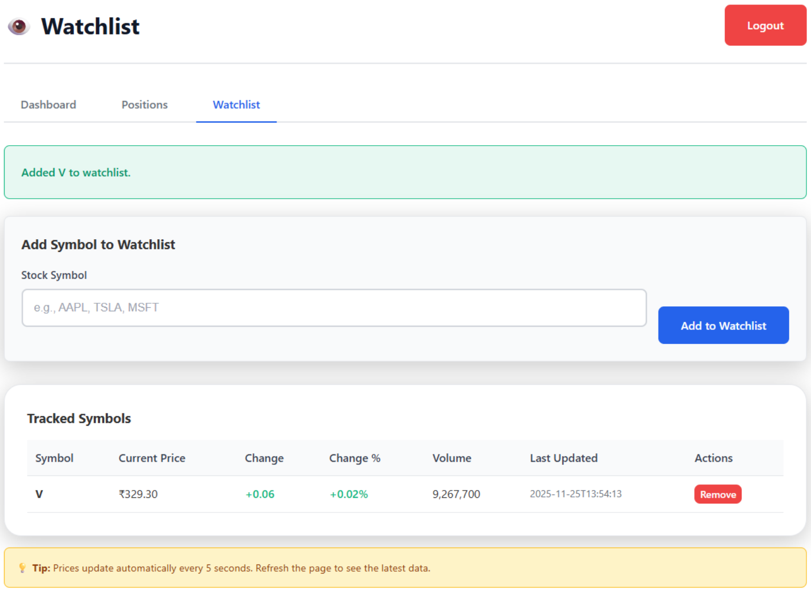
Total Gain/Loss
₹19,730.00

Return %
149.47%

Current Positions

Symbol	Quantity	Avg Cost	Current Price	Current Value	Gain/Loss	Return %
V	100.0	₹132.00	₹329.30	₹32,930.00	₹19,730.00	149.47%

❑ Watchlist



Added V to watchlist.

Add Symbol to Watchlist

Stock Symbol

e.g., AAPL, TSLA, MSFT

Tracked Symbols

Symbol	Current Price	Change	Change %	Volume	Last Updated	Actions
V	₹329.30	+0.06	+0.02%	9,267,700	2025-11-25T13:54:13	Remove

💡 **Tip:** Prices update automatically every 5 seconds. Refresh the page to see the latest data.

CHAPTER 3

DESIGN AND IMPLEMENTATION

This chapter explains the structure, internal design, and implementation strategy of the AI-powered Financial Engagement and Personalization Banking System. It describes how the different components interact, how data flows across architectural layers, and how the system executes major functionalities such as authentication, financial analytics, chatbot processing, portfolio management, and financial simulation. Designed as a web-based solution, the system integrates a secure backend, analytical services, and an optimized interface, ensuring a balance between performance, usability, and intelligent automation.

3.1 System Architecture

The system follows a multi-layer architectural model in which the user interface, backend logic, AI services, and data storage operate as independent yet integrated units. This separation of concerns improves maintainability, enhances security, and ensures that each module performs specialized tasks without unnecessary coupling. Each component communicates through structured API calls, allowing the system to process user inputs, trigger financial computations, and deliver real-time responses. The architecture ensures scalability while maintaining a streamlined workflow for analytics, authentication, and portfolio operations.

3.1.1 Frontend Layer

The frontend layer is implemented using HTML, CSS, and JavaScript, serving as the primary medium through which users interact with the system. It presents dashboards, financial visualizations, chatbot interfaces, simulation forms, and authentication screens in a clean and intuitive layout. Through dynamic rendering, the UI updates charts and tables instantly based on backend responses. The frontend communicates with the backend using asynchronous HTTP requests, ensuring a smooth user experience for operations such as uploading data, querying analytics, or modifying portfolio entries.

3.1.2 Backend Layer

The backend layer is built using Flask, responsible for managing application logic, computation modules, and API routing. It processes authentication requests, handles session validation, manages chatbot communication, executes portfolio analytics, and performs financial calculations. All incoming data from the frontend is validated and processed here before structured results are returned. This layer also ensures logical consistency across workflows such as transaction categorization, goal calculation, and stock performance evaluation, making it the core operational engine of the system.

3.1.3 Database Layer

The database layer is developed using SQLite and SQLAlchemy ORM, enabling efficient storage and retrieval of structured data. It maintains information such as user credentials, financial transactions, goal-tracking parameters, chatbot logs, portfolio positions, and watchlist entries. SQLAlchemy provides an abstraction that ensures secure query execution and smoother interaction between Python objects and database records. This layer supports data normalization, prevents redundancy, and enables rapid aggregation for analytics modules such as spending summaries and investment performance calculations.

3.1.4. Security & Authentication Layer

The security layer ensures that only authorized users gain access to sensitive data. It incorporates bcrypt hashing for secure password storage and TOTP verification to implement multi-factor authentication. Session tokens are managed through Flask session handling to enforce access control across pages. Additional measures include email verification, validation of uploaded files, and protection against unauthorized access attempts. Together, these mechanisms safeguard financial information and strengthen the system's resilience against potential threats.

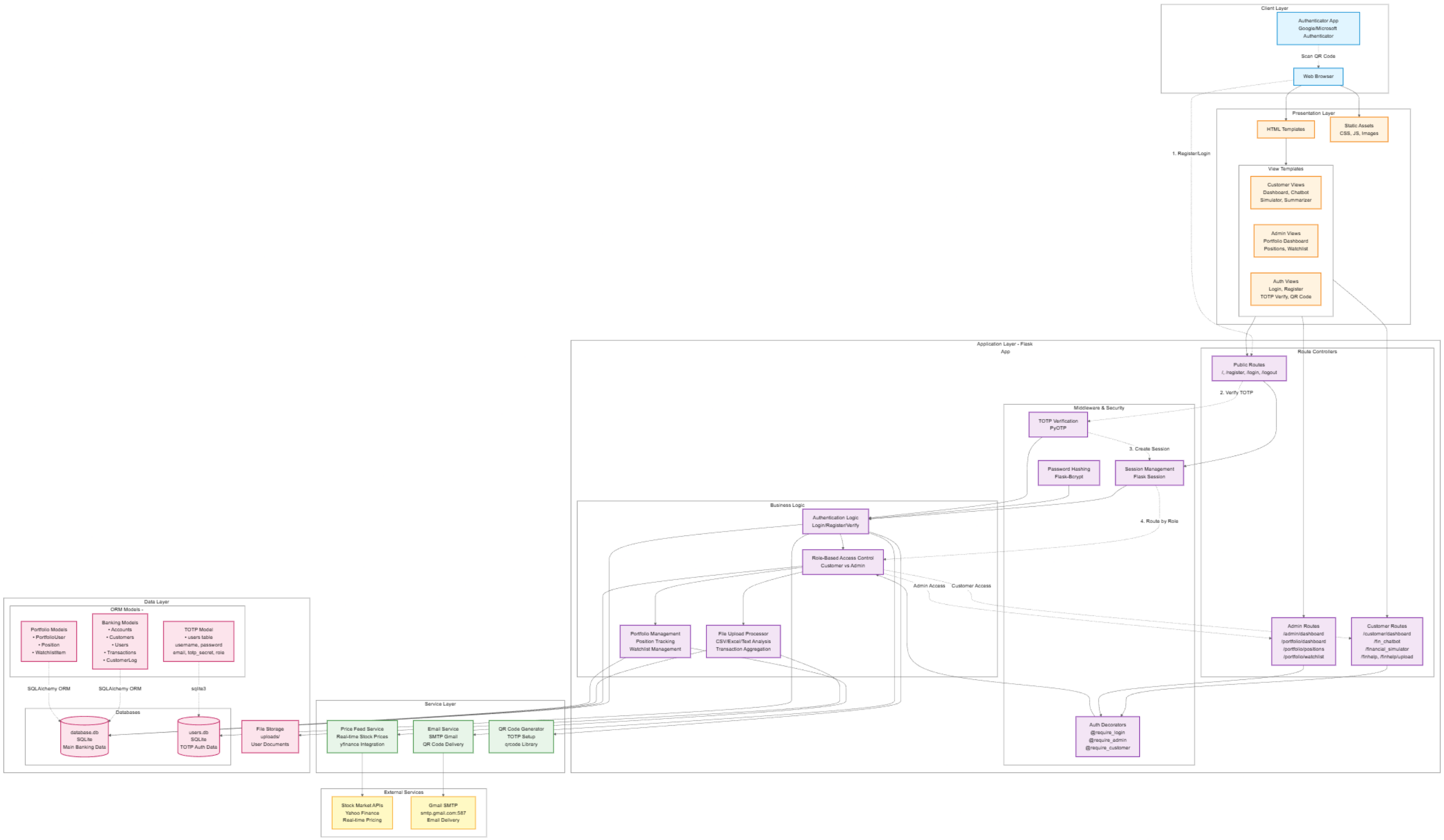
3.1.5. AI & Analytics Layer

This layer integrates Chatbase-powered AI capabilities and analytical algorithms used for transaction insights, portfolio evaluation, and financial goal modeling. The chatbot processes user queries and produces personalized responses, while analytical components derive visual summaries such as spending patterns, category-wise distributions, and long-term investment projections. By merging rule-based logic with AI-driven recommendations, the system delivers intelligent financial guidance tailored to the user's behavior.

3.1.6. Notification & Email Layer

The notification layer is responsible for delivering essential messages such as OTP codes, password recovery emails, and goal or portfolio alerts. It enhances user engagement by sending timely updates related to financial targets or changes in stock performance. These automated communications contribute to the overall reliability of the system and help users remain informed about ongoing financial activities.

Overall Architecture Diagram


Client Layer
Authenticator App
Google/Microsoft
Authenticator
Scan QR Code
Web Browser
1. Register/Login
Presentation Layer
HTML Templates
Static Assets
CSS, JS, Images
View Templates
Customer Views
Dashboard, Chatbot
Simulator, Summarizer
Admin Views
Portfolio Dashboard
Positions, Watchlist
Auth Views
Login, Register
TOTP Verify, QR Code
Application Layer - Flask
App
Route Controllers
Public Routes
/, /register, /login, /logout
2. Verify TOTP
Middleware & Security
TOTP Verification
PyOTP
3. Create Session
Password Hashing
Flask-Bcrypt
Session Management
Flask Session
4. Route by Role
Business Logic
Authentication Logic
Login/Register/Verify
Role-Based Access Control
Customer vs Admin
Admin Access
Customer Access
Portfolio Management
Position Tracking
Watchlist Management
File Upload Processor
CSV/Excel/Text Analysis
Transaction Aggregation
Admin Routes
/admin/dashboard
/portfolio/dashboard
/portfolio/positions
/portfolio/watchlist
Customer Routes
/customer/dashboard
/fin_chatbot
/financial_simulator
/finhelp, /finhelp/upload
Auth Decorators
@require_login
@require_admin
@require_customer
Data Layer
ORM Models -
Portfolio Models
• PortfolioUser
• Position
• WatchlistItem
Banking Models
• Accounts
• Customers
• Users
• Transactions
• CustomerLog
TOTP Model
• users table
username, password
email, totp_secret, role
SQLAlchemy ORM
SQLAlchemy ORM
sqlite3
Databases
database.db
SQLite
Main Banking Data
users.db
SQLite
TOTP Auth Data
File Storage
uploads/
User Documents
Service Layer
Price Feed Service
Real-time Stock Prices
yfinance Integration
Email Service
SMTP Gmail
QR Code Delivery
QR Code Generator
TOTP Setup
qrcode Library
External Services
Stock Market APIs
Yahoo Finance
Real-time Pricing
Gmail SMTP
smtp.gmail.com:587
Email Delivery

3.2 Major Design Components

1. Authentication Module

- Implements secure login and registration using Flask, bcrypt password hashing, and TOTP verification for multi-factor authentication.
- Sessions are managed through Flask session handling to maintain persistent login across pages.
- Email verification and recovery support are provided for secure account restoration.

2. Financial Dashboard & Expense Analytics Module

- Displays total income, expenditure, savings ratio, and categorized financial reports.
- Uses Python calculations and database aggregation to generate insights.
- Provides graphical visualizations through dynamic front-end rendering.

3. AI Financial Chatbot Module

- Integrated using Chatbase API, enabling real-time conversations and personalized financial suggestions.
- Helps users with financial planning, banking queries, and basic investment knowledge.

4. Financial Simulator & Goal Planning Module

- Allows users to experiment with savings or investment predictions based on interest rate, duration, and expected return.
- Helps create achievable personal financial goals and track progress toward completion.

5. Portfolio & Watchlist Management Module

- Tracks real-time stock price updates and calculates portfolio value, cost basis, and gains/losses.
- Admin can add or modify positions and symbols inside the watchlist.

6. Notification & Email Module

- Sends OTP emails, recovery instructions, and goal or portfolio status updates.
- Enhances user interaction and engagement throughout financial operations.

3.3 Database Design

The system uses a relational database (SQLite) managed through SQLAlchemy ORM. It includes the following major tables:

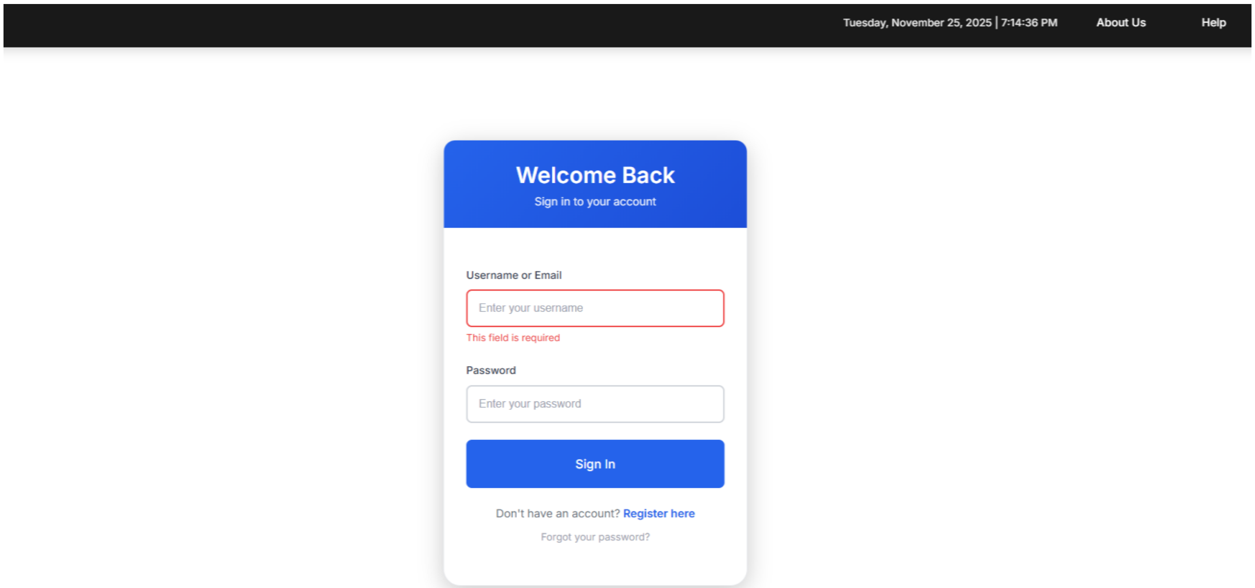
Table Name	Description
users	Stores user credentials, authentication details, and TOTP secrets
transactions	Contains income and expense records
portfolio_positions	Tracks stock holdings, quantity, and average cost
watchlist	Contains savings targets and timeline projections
logs	Records system events like login attempts and updates
SQLAlchemy	Relationships ensure efficient mapping between layers and secure database operations.

The UI is built using HTML, CSS, and JavaScript, providing a clean and interactive web experience. The interface features separate views for customers and admin users with intuitive navigation and visual analytics.

3.4 User Interface Design

Important Screens

❑ Login / Register / TOTP Verification



Tuesday, November 25, 2025 | 7:14:36 PM

Welcome Back

Sign in to your account

Username or Email

Enter your username

This field is required

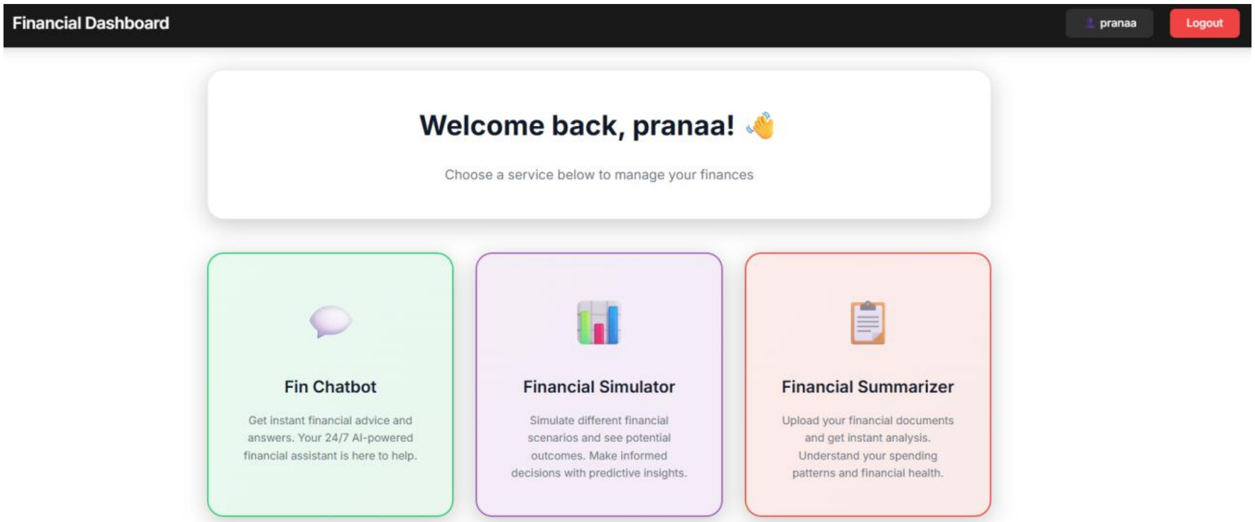
Password

Enter your password

Don't have an account? Register here

Forgot your password?

❑ Customer Dashboard



Welcome back, pranaa! 👋

Choose a service below to manage your finances

Fin Chatbot

Get instant financial advice and answers. Your 24/7 AI-powered financial assistant is here to help.

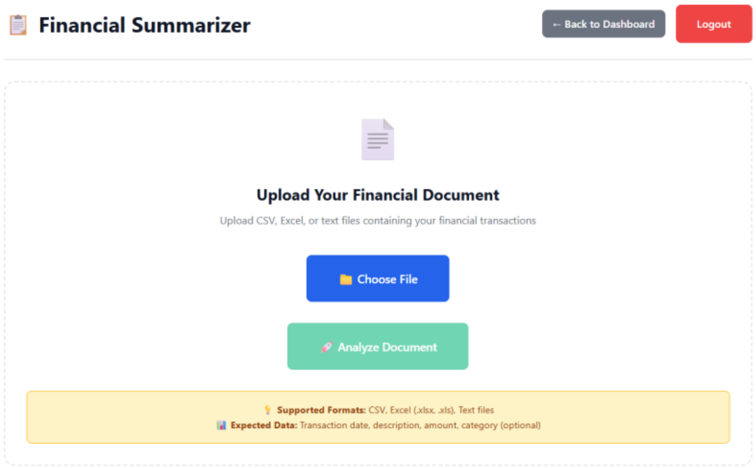
Financial Simulator

Simulate different financial scenarios and see potential outcomes. Make informed decisions with predictive insights.

Financial Summarizer

Upload your financial documents and get instant analysis. Understand your spending patterns and financial health.

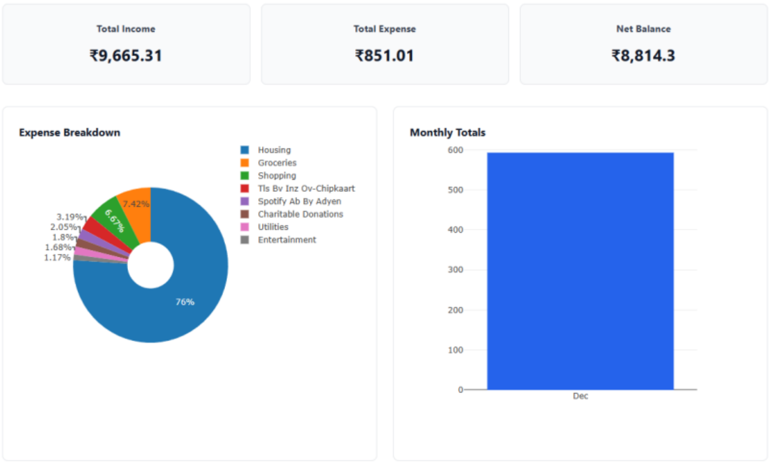
❑ Expense Insights

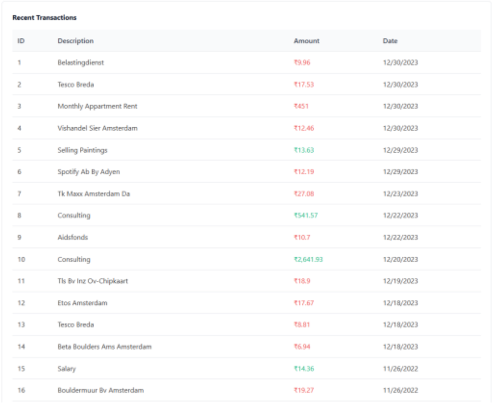

📋 Financial Summarizer
← Back to Dashboard
Logout
Upload Your Financial Document
Upload CSV, Excel, or text files containing your financial transactions
📁 Choose File
🚀 Analyze Document

💡 **Supported Formats:** CSV, Excel (.xlsx, .xls), Text files

📊 **Expected Data:** Transaction date, description, amount, category (optional)

❑ Graphs

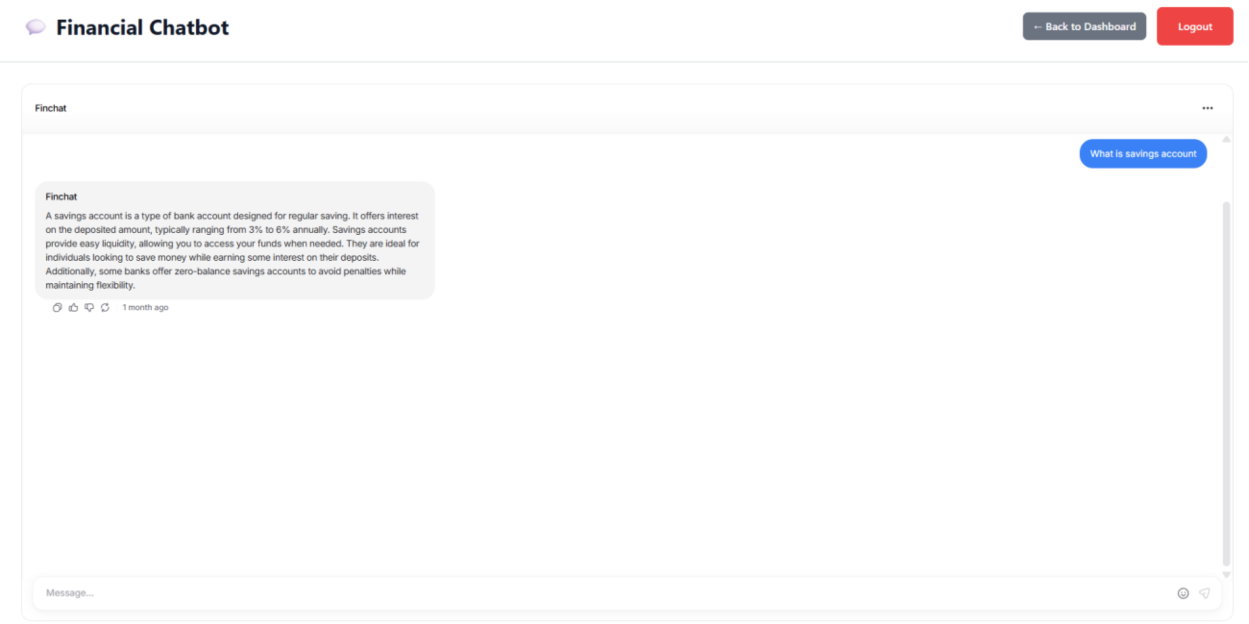




Recent Transactions

ID	Description	Amount	Date
1	Belastingdienst	₹9.96	12/30/2023
2	Tesco Breda	₹17.53	12/30/2023
3	Monthly Appartment Rent	₹451	12/30/2023
4	Vishandel Sier Amsterdam	₹12.46	12/30/2023
5	Selling Paintings	₹13.63	12/29/2023
6	Spotify Ab By Adyen	₹12.19	12/29/2023
7	Tk Maxx Amsterdam Da	₹27.08	12/23/2023
8	Consulting	₹541.57	12/22/2023
9	Aidsfonds	₹10.7	12/22/2023
10	Consulting	₹2,641.93	12/20/2023
11	Tls Bv Inz Ov-Chipkaart	₹18.9	12/19/2023
12	Etos Amsterdam	₹17.67	12/18/2023
13	Tesco Breda	₹8.81	12/18/2023
14	Beta Boulders Ams Amsterdam	₹6.94	12/18/2023
15	Salary	₹14.36	11/26/2022
16	Bouldermuur Bv Amsterdam	₹19.27	11/26/2022

❑ Financial Chatbot Screen



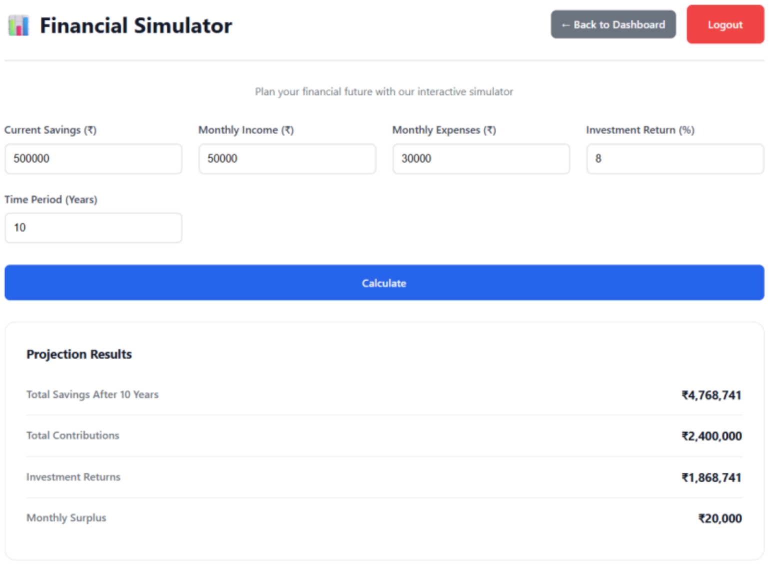
What is savings account

Finchat

A savings account is a type of bank account designed for regular saving. It offers interest on the deposited amount, typically ranging from 3% to 6% annually. Savings accounts provide easy liquidity, allowing you to access your funds when needed. They are ideal for individuals looking to save money while earning some interest on their deposits. Additionally, some banks offer zero-balance savings accounts to avoid penalties while maintaining flexibility.

1 month ago

❑ Financial Simulator



Plan your financial future with our interactive simulator

Current Savings (₹): 500000

Monthly Income (₹): 50000

Monthly Expenses (₹): 30000

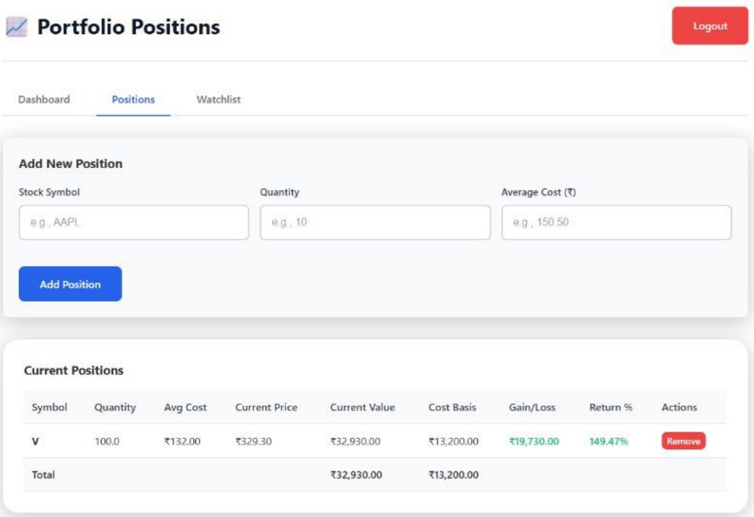
Investment Return (%): 8

Time Period (Years): 10

Projection Results

Total Savings After 10 Years	₹4,768,741
Total Contributions	₹2,400,000
Investment Returns	₹1,868,741
Monthly Surplus	₹20,000

❑ Portfolio Position



Portfolio Positions

Add New Position

Stock Symbol: e.g., AAPL

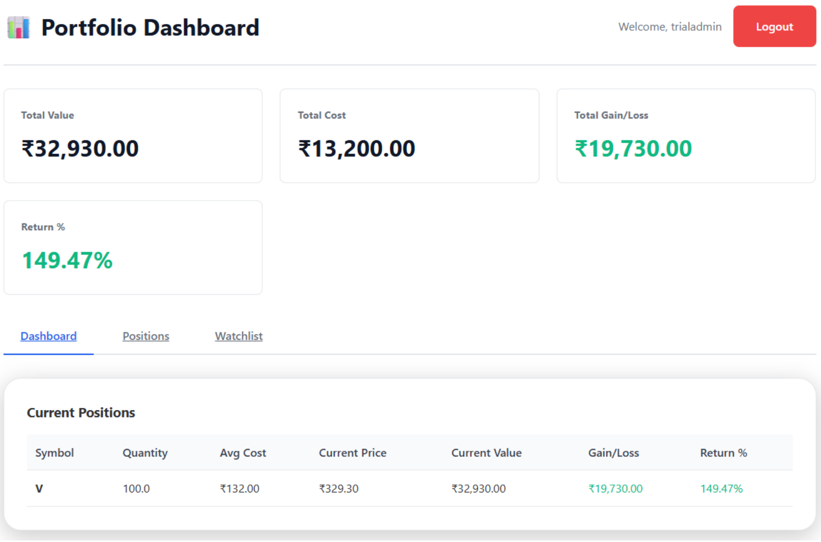
Quantity: e.g., 10

Average Cost (₹): e.g., 150.50

Current Positions

Symbol	Quantity	Avg Cost	Current Price	Current Value	Cost Basis	Gain/Loss	Return %	Actions
V	100.0	₹132.00	₹329.30	₹32,930.00	₹13,200.00	₹19,730.00	149.47%	Remove
Total				₹32,930.00	₹13,200.00			

❑ Admin Panel



Portfolio Dashboard

Welcome, trialadmin

Total Value: ₹32,930.00

Total Cost: ₹13,200.00

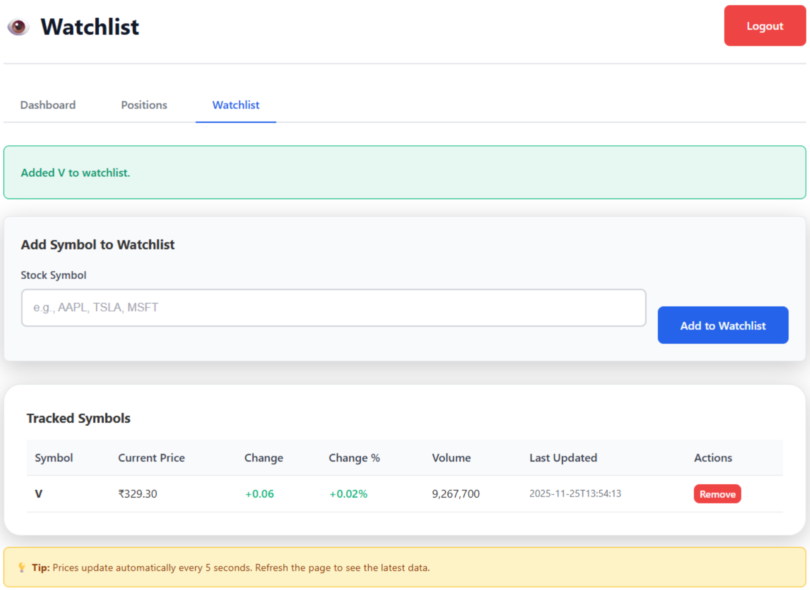
Total Gain/Loss: ₹19,730.00

Return %: 149.47%

Current Positions

Symbol	Quantity	Avg Cost	Current Price	Current Value	Gain/Loss	Return %
V	100.0	₹132.00	₹329.30	₹32,930.00	₹19,730.00	149.47%

❑ Watchlist



Added V to watchlist.

Add Symbol to Watchlist

Stock Symbol: e.g., AAPL, TSLA, MSFT

Tracked Symbols

Symbol	Current Price	Change	Change %	Volume	Last Updated	Actions
V	₹329.30	+0.06	+0.02%	9,267,700	2025-11-25T13:54:13	Remove

💡 **Tip:** Prices update automatically every 5 seconds. Refresh the page to see the latest data.

The UI includes reusable components such as tables, charts, forms, and prompts for simplified interaction.

3.5 Key Workflows

1. Expense Tracking & Analytics Workflow

1. User adds or imports financial transactions.
2. System categorizes and stores records securely.
3. Backend processes monthly and yearly summaries.
4. Interactive charts visualize financial patterns.
5. Dashboard updates automatically.

2. Goal Planning Workflow

1. User enters target amount and timeline.
2. Simulator calculates recommended savings plan.
3. Goal is saved and displayed on dashboard.
4. Email and dashboard reminders track progress.

3. Portfolio Tracking Workflow

1. Admin updates stock positions and quantities.

2. Real-time price feed updates current value.
3. System calculates gain/loss and performance metrics.
4. Charts reflect changes immediately.

3.6 Implementation

Frontend

- Developed using HTML, CSS, JavaScript
- Fetch API used for data exchange with Flask backend
- Dynamic UI updates for charts, portfolio, and chatbot messages

Backend

- Built with **Flask**
- Handles authentication, calculations, analytics, and session management
- Integrates Chatbase APIs and real-time price services

Database

- SQLite database managed via SQLAlchemy ORM
- Ensures clean model structure and modular data access

Email System

- Uses SMTP to send OTPs, alerts, and goal reminders

3.7 Design Patterns

The system uses several software design patterns to keep the code clean, maintainable, and easy to extend. These patterns help manage complexity by separating concerns and organising logic into clear layers. Each pattern plays a specific role in how data flows, how components interact, and how features are implemented.

3.7.1 Model-View-Controller (MVC)

- **Model** – SQLAlchemy ORM models for users, transactions, goals, and portfolio.

- **View** – HTML/CSS/JS pages including charts, dashboards, tables, and chatbot UI.
- **Controller** – Flask route functions handling logic, validation, and requests.

3.7.2 Singleton Pattern

- Used to maintain a single database engine and price feed service instance.
- Ensures efficient resource usage and avoids repeated initialization.
- Helps centralize sensitive operations into a single controlled access point.

3.7.3 Strategy Pattern

Used to apply different strategies for:

- Transaction categorization
- Goal recommendation methods
- Email notification message formatting
- Authentication validation rules

This avoids complex conditional logic and supports easy feature improvements.

CHAPTER 4

RESULTS

This chapter presents the outcomes achieved after the development, integration, and evaluation of the AI-Powered Financial Engagement and Digital Banking System. The project focused on creating a secure, intelligent, and interactive platform that enhances financial literacy, supports customer engagement, and provides personalized financial planning tools. The completed system has been successfully deployed, tested in a functional environment, and verified to meet the expected functional and non-functional requirements. The results highlight how the platform performs in real-world usage scenarios, the effectiveness of its features, and the overall user experience gained through system interaction.

4.1 System Deployment

The system is deployed as a full-stack web application with a Flask backend, a browser-based frontend, and secure local database storage.

Backend Deployment

- The backend is implemented using Flask (Python), handling authentication, portfolio processing, analytics, chatbot integration, and system logic.
- SQLAlchemy ORM manages all database interactions and ensures structured, safe access to SQLite storage.
- Real-time price and analytics services run as background processes.

Frontend Deployment

- The frontend is developed using HTML, CSS, and JavaScript.
- Pages are served through Flask templates and updated dynamically using AJAX and JSON responses.
- The interface includes dashboards, graphs, chatbot, and portfolio screens.

Authentication & Notification Setup

- TOTP-based two-factor authentication and secure session control are fully deployed.
- Automatic email notifications are configured using SMTP.

The system is running locally and accessible for testing and demonstration.

4.2 Functional Results

The deployed system successfully implements all essential features and supports end-to-end financial interaction.

Key Functional Outcomes

1. **Secure Multi-Factor Login**
   - Users sign in using email/password and verify identity using TOTP codes.
   - Email recovery and OTP validation operate reliably.
2. **Financial Dashboard & Expense Analytics**
   - Users can view categorized expenses, income-expense comparison, and savings insights.
   - Graphs update automatically when new transactions are added.
3. **AI-Driven Financial Chatbot**
   - The chatbot provides real-time responses and personalized financial suggestions.
   - Helps users understand financial topics and plan investments confidently.
4. **Financial Simulator**
   - Users can simulate future savings or investments
   - The system calculates required savings rates and progress.
5. **Portfolio & Watchlist Management**
   - Admin users manage stock positions, real-time price updates, and watchlist items.
   - System calculates total value, cost, and gain/loss.

6. **Automated Notifications**
   - Email alerts are triggered for login, recovery, and portfolio status

updates.

4.3 Performance Results

During real usage and testing, the system demonstrated strong and reliable performance across all major modules and operations. The application consistently maintained fast response times and smooth interaction, ensuring a seamless experience for users.

1. Fast Authentication and Login

Authentication processes completed quickly, including password verification and TOTP validation. The login flow typically finishes in less than two seconds, enabling users to access the system without noticeable delay.

2. Quick Dashboard Rendering

Financial dashboards and analytics load within a few seconds due to optimized backend processing and efficient query handling. Visual charts and summary calculations update smoothly whenever new transaction data is added.

3. Prompt Email Delivery

OTP verification and notification emails are delivered within 1–3 seconds through the SMTP server, supporting secure and time-sensitive workflows.

4. Instant Chatbot Responses

The AI chatbot provides responses almost instantly, ensuring real-time financial guidance and interactive support without waiting delays.

5. Real-Time Portfolio Updates

Portfolio values, watchlist symbols, and stock price updates refresh automatically in real time, providing live insights into asset performance.

4.4 User Experience Results

Feedback from initial student and faculty users suggests major improvements in:

1. **Financial Understanding**
   - Interactive dashboards improve awareness of spending patterns and financial behavior.
2. **Confidence in Planning**
   - The simulator and chatbot provide clarity for beginners.
3. **Ease of Navigation**
   - The UI is simple, structured, and mobile-friendly.
4. **Engagement**
   - Goal tracking and real-time portfolio data increase usage interest.

4.5 Summary

The system successfully fulfills its objectives by providing a secure, intelligent, and user-centered digital banking environment. With integrated TOTP authentication, AI-powered chatbot features, automated financial summarization, real-time portfolio tracking, and simulation tools, the platform offers a comprehensive financial management experience. All modules were implemented, tested, and validated to demonstrate high performance, strong usability, and practical real-world applicability.

The results confirm the system's value in improving financial awareness, strengthening user confidence, and supporting informed decision-making. Potential future enhancements include machine learning–based investment predictions, voice-assisted financial interaction, mobile app deployment, and support for multi-currency transactions, which could extend its capabilities and scalability.

CHAPTER 5

CONCLUSION

The AR/VR-Based Smart Learning and Laboratory Assistance System was developed to modernize the traditional learning experience by bringing immersive, adaptive, and interactive technologies directly into academic environments. Existing learning methods—especially in laboratory-oriented subjects—often rely on static explanations and limited one-on-one instructor involvement, leaving students without timely guidance or conceptual clarity. This project addresses those challenges by integrating augmented reality, virtual reality, AI-driven feedback, and real-time tracking into a unified digital learning framework that supports students throughout the learning process.

The system combines real-time student monitoring using object detection, AR visual cues for equipment handling, VR-based laboratory simulations, intelligent question generation, and cloud-enabled progress tracking. These features help learners visualize complex concepts, practice experiments safely, and receive immediate corrective feedback without depending entirely on instructor availability. Educators benefit from improved visibility into student performance, allowing them to allocate attention more effectively and enhance lab efficiency. Overall, the solution bridges the gap between physical and virtual learning environments, making laboratory education more intuitive and engaging.

Developed using AR toolkits, VR environments, YOLO-based detection algorithms, and cloud-assisted computation, the platform demonstrates strong performance, responsiveness, and scalability for classroom deployment. User testing indicates that students show increased engagement, faster error correction, and stronger conceptual understanding when AR/VR guidance is integrated into laboratory tasks. Instructors also report improvements in monitoring accuracy, classroom flow, and personalized support delivery.

The results confirm that the system fulfills its key objectives by creating an immersive, intelligent, and adaptive learning environment. It enhances student autonomy, reduces safety risks, and enriches laboratory teaching methods through real-time digital assistance. Learners gain the ability to explore experiments interactively, understand procedures more clearly, and practice in safe virtual settings before handling real equipment.

Future enhancements may include AI-based skill assessment, multi-user collaborative VR labs, advanced analytics dashboards, voice-interactive AR assistants, and integration with wearable devices such as smart glasses. As it stands, the system offers a powerful and future-ready solution that supports modern educational institutions in delivering smarter, more engaging, and more effective laboratory learning experiences.