

JSS MAHAVIDYAPEETHA
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
JSS Technical Institutions Campus Mysore - 570006



FINANCIAL ADVISER WEB APPLICATION

A report submitted in partial fulfilment of the academic requirements prescribed
for award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE BUSINESS SYSTEMS

by

YOUSUF MAKKI 01JST22UCB059

DHANUSH LAL 01JST23UCB402

HARIS CA 01JST23UCB402

Department of Information Science and Engineering
JSS Science and Technology University, Mysore
2024-2025

Abstract

YOUSUF MAKKI (01JST22UCB059)

HARIS CA (01JST23UCB403)

DHANUSH LAL (01JST23UCB402)

CIE E1, E2 for Modern Web Applications (22CB662)

6th Semester Computer System and Business System

Department of Information Science and Engineering

JSS Science and Technology University, Mysuru.

Submitted Date:

WiseFinance is a comprehensive web-based financial management platform designed to empower individuals and professionals in making smarter financial decisions. The application offers a suite of interactive financial calculators, including tools for Present Value, Future Value, NPV, SIP, FD, amortization, and various return metrics. Built using HTML, CSS, JavaScript, and Bootstrap, the platform ensures a responsive and user-friendly experience. With a mission to simplify finance for all, WiseFinance allows users to analyze their financial goals efficiently through accurate and real-time calculations. In addition to financial tools, WiseFinance introduces a unique advisory model that connects users with verified finance professionals for personalized guidance. Users can create secure accounts using Firebase Authentication and access a freemium Q&A feature, where the first four queries are free. All professional advisors receive user queries in real time and are incentivized to respond through a built-in reward system. This collaborative approach fosters a knowledge-sharing ecosystem while helping users make informed decisions. By blending financial computation, community support, and secure access, WiseFinance aims to democratize personal finance through smart, scalable technology.

Table of contents

Chapter No	Sub heading	Content	Page No
01	Introduction	Introduction	05
		About the Project	07
		Motivation	08
		Challenges and problem statements	08
		Objectives	08
		Requirements	9
02	Literature Survey	Literature Survey	10
03	Methodology	Front end design	12
		Backend design	12
04	Implementation	Implementation	21
		Conclusion and Future Enhancement	25
		References	26
		Appendix	27

Table of Figures

Figure No	Title	Page No
3.1	Flow Chart	15
3.2	Data flow	16

Chapter 1

Introduction

1. Introduction

Managing personal and business finances can be overwhelming without the right tools and guidance. In today's fast-paced digital world, individuals often seek quick, reliable, and user-friendly solutions to understand and plan their financial goals. **WiseFinance** is a smart web-based platform developed to meet this need by providing users with powerful financial calculators and access to professional advice, all in one place.

The platform simplifies complex financial concepts by offering calculators for Present Value, Future Value, SIP, NPV, and more, making it easy for users to evaluate investment options and savings plans. What sets WiseFinance apart is its built-in advisory feature, where users can interact with registered financial professionals, ask questions, and receive personalized insights—creating a collaborative and interactive financial ecosystem. With secure user authentication, an intuitive interface, and a freemium model, WiseFinance bridges the gap between financial literacy and technology.

2. About the Project

WiseFinance is an interactive financial management web application designed to help users—both individuals and professionals—understand and plan their finances better. The platform provides a suite of financial calculators and a unique advisory system that bridges users with verified financial experts for personalized support. The goal is to simplify finance using technology by offering reliable tools, secure access, and expert insights—all within a user-friendly interface.

What Exactly Is the Work?

The core objective of WiseFinance is to create a centralized platform for:

- **Performing financial calculations** like Present Value (PV), Future Value (FV), Net Present Value (NPV), SIP, FD, amortization, total return, and more.
- **Allowing user registration** (via Firebase Authentication with email/password or Google Sign-In) to create secure profiles.
- **Connecting users to finance professionals**, called *suggestioners*, who can respond to finance-related queries.

- **Implementing a freemium model** where users get the first 4 queries free, with the option to upgrade later.
- **Broadcasting user questions** to all suggestioners and rewarding those who respond effectively.
- **Providing clean UI/UX** for mobile and desktop users using technologies like HTML, CSS, Bootstrap, and JavaScript.

How Does It Work?

1. User Registration/Login:

- Users sign up using Firebase Authentication (email or Google).
- Profile data is stored securely and used for personalization.

2. Accessing Financial Calculators:

- Once logged in, users can navigate to the calculator section.
- They input values (like interest rate, number of years, amount, etc.) into the relevant calculator.
- The system uses JavaScript-based logic to calculate the result and show it instantly on the UI.

3. Query & Advisory Feature:

- Users can ask finance-related questions directly from their profile dashboard.
- These queries are instantly sent to all suggestioners registered on the platform.
- Suggestioners can reply with solutions or advice.
- A reward system motivates suggestioners to provide valuable input.

4. Freemium Access Control:

- The app tracks how many questions a user has asked.
- After 4 free questions, the system prompts users to upgrade to a paid plan (future integration).

5. Data Security & Management:

- All user data and queries are stored in Firebase Firestore.
- Firebase Authentication ensures only authorized access.

- Firebase rules restrict sensitive data access to the correct roles (user/suggestioner).

Inputs and Outputs:

Function	Inputs	Outputs
PV/FV/NPV Calculators	Amount, rate, time period, cash flows	Computed financial result displayed in UI
User Registration/Login	Email, password / Google login	Authentication token and profile data
Query Submission	Text-based question from the user	Stored in database; sent to suggestioners
Advisor Response	Text-based answer from a suggestioner	Displayed in user's dashboard
Freemium System	User query count	Access warning or upgrade prompt after 4 questions

Table 1.1

Technologies Used:

- **Frontend:** HTML, CSS, Bootstrap, JavaScript
- **Backend/Database:** Firebase Firestore
- **Authentication:** Firebase Authentication
- **Hosting (optional):** Vercel or Firebase Hosting
- **Version Control:** GitHub

1.2.1 Motivation

The motivation behind developing **WiseFinance** stemmed from a simple but powerful observation: most individuals struggle with financial planning due to a lack of tools, knowledge, or access to expert advice. In today's world, financial literacy is more important than ever, yet many people find it overwhelming to calculate their future savings, understand investment returns, or even know whom to consult for trusted advice. I wanted to bridge that gap by building a digital solution that empowers people to make better financial decisions, even without a finance background.

I have always been interested in how technology can simplify complex problems. The idea of combining **financial logic** with **web technologies** like **JavaScript** for real-time calculations, **Firebase** for seamless authentication and data management, and **Bootstrap** for responsive design made the development process both challenging and exciting. This domain also gave me a chance to explore how real-world financial concepts like SIP, NPV, and CAPM can be

implemented using code. The vision of making finance interactive, accessible, and user-driven inspired me to create WiseFinance as not just a tool—but a smart financial companion.

1.2.2 Challenges

While working on **WiseFinance**, I identified several real-world problems that inspired the features and functionality of the platform. These challenges are common among individuals trying to manage their personal or business finances without formal financial training or easy access to advisors. The following are the key difficulties that WiseFinance aims to solve:

1. **Lack of Financial Awareness and Tools:**

Many individuals are unfamiliar with important financial concepts like present value, future value, compound interest, NPV, or SIP. They either rely on manual methods or avoid planning altogether due to the complexity. WiseFinance solves this by providing a collection of easy-to-use calculators that deliver real-time results with user-friendly interfaces.

2. **Limited Access to Financial Advisors:**

Getting professional financial advice is often costly, and many people don't know where or whom to approach. There's also a lack of trust in sources found online. WiseFinance bridges this gap by allowing verified professionals (suggestioners) to register on the platform and answer user queries, creating a trustworthy and collaborative environment.

3. **Inconvenient and Unsecure User Experience on Existing Platforms:**

Most financial platforms are either too complex for casual users or lack proper authentication and data privacy. WiseFinance provides a secure and streamlined experience using **Firestore Authentication** (email and Google Sign-In), ensuring only authorized access and a smooth user journey

1.2.3 Problem Definition

Before starting the development of **WiseFinance**, I aimed to solve three key challenges that individuals commonly face in managing their finances. My goal was to create a smart, accessible, and user-friendly platform that combines automation with expert support. The following are the core ideas I developed to tackle each identified problem:

1. **Solution to Lack of Financial Awareness and Tools:**

To make financial planning simple and accessible, I planned to develop a suite of calculators for essential financial metrics like SIP, NPV, PV, FV, and more. These calculators would accept basic user inputs (like amount, interest rate, duration) and use built-in JavaScript logic to provide instant, accurate outputs. The goal was to eliminate confusion and help users make informed decisions without needing

advanced financial knowledge.

2. Solution to Limited Access to Financial Advisors:

My idea was to build an integrated advisory system where registered financial professionals, called "suggestioners," could respond to user queries. Instead of users searching the internet or booking paid consultations, they could post questions on WiseFinance and receive trusted advice from a pool of professionals, making the guidance more accessible, collaborative, and affordable.

3. Solution to Inconvenient and Unsecure User Experience:

To ensure a secure and smooth experience, I decided to integrate Firebase Authentication for login and user identity management. This would allow users to register easily using email or Google Sign-In, while ensuring that their data and queries are stored securely in Firebase Firestore. The user interface would be developed using responsive design principles with HTML, CSS, JavaScript, and Bootstrap, so it works seamlessly across devices.

1.2.4 Objectives

Based on the problem definitions identified earlier, the **WiseFinance** project focuses on solving the following two primary challenges:

1. **To provide user-friendly, accurate, and instant financial calculators** that help individuals understand and analyze key financial concepts such as Present Value, Future Value, SIP, NPV, and more. This addresses the lack of financial awareness and tools by simplifying complex calculations and making them accessible to non-experts.
2. **To build a secure, collaborative advisory system** where users can interact with verified financial professionals (suggestioners), ask finance-related questions, and receive personalized guidance. This directly solves the problem of limited access to trustworthy and affordable financial advisors.

1.2.5 Requirements

Software Requirements:

Category	Details
Operating System	Windows / macOS / Linux
Code Editor	VS Code (Visual Studio Code)
Languages Used	HTML, CSS, JavaScript
Libraries/ Frameworks	Bootstrap (for responsive design)
Database	Firebase Firestore (NoSQL, cloud-based)
Authentication	Firebase Authentication (Email/Password, Google Sign-In)
Version Control	Git + GitHub
Hosting/Deployment	Vercel / Firebase Hosting (for live web access)
Browser Compatibility	Chrome, Firefox, Edge (modern browsers)

Table 1.2

Chapter 2

Literature Survey

1 “A Web-Based Financial Planning System Using Real-Time Calculators and User Interface Design” that effective personal finance applications require intuitive user interfaces combined with strong financial logic in the backend. The paper emphasizes that most users are not financially literate, so tools like Present Value, Future Value, and NPV calculators should be implemented in a way that minimizes user input complexity and maximizes clarity of results. This approach supports the development of WiseFinance, which integrates simplified financial tools with Bootstrap-based UI design to ensure a smooth user experience across devices. The study also discusses the importance of responsive design and how Bootstrap helps in catering to users on mobile and desktop platforms, which influenced the frontend architecture of the project.

2 “Firebase as a Backend Service for Secure Authentication and Data Management in Financial Applications” how Firebase Authentication and Firestore can be used to securely manage user data and login processes in modern web apps. The paper presents a comparison between traditional backend services and cloud-based services, concluding that Firebase significantly reduces complexity and improves scalability. This directly motivated the use of Firebase in WiseFinance to handle secure login (via email/password and Google Sign-In), real-time data storage of user queries, and role-based access for financial advisors. The author also explores Firebase’s security rules and their implementation to protect sensitive data—another critical factor in the development of this platform where finance-related queries and identities are handled.

3 “*Design of an Online Expert Consultation System for Finance and Investment Guidance*” that modern web platforms can simulate real-world advisory systems by connecting users with domain experts online. The author introduces the concept of suggestion broadcasting, where a user's query is sent to multiple experts simultaneously, ensuring faster and more diverse responses. This concept strongly influenced WiseFinance’s advisor module, where users submit questions that are sent to all registered financial suggestioners, and those who respond are rewarded based on activity. The paper also discusses the importance of trust-building mechanisms, such as advisor verification and user ratings, which could be future enhancements for the WiseFinance system.

Chapter 3

Methodology

The frontend of the WiseFinance web application is designed to be clean, intuitive, and responsive so that users can easily interact with all features regardless of their technical background. The frontend is developed using HTML, CSS, JavaScript, and Bootstrap. Each technology plays a specific role in ensuring an engaging and user-friendly interface.

How Frontend Interacts with the User

The frontend acts as the bridge between the user and the application's logic. When a user inputs data (like amount, interest rate, duration, etc.) into a financial calculator form, the frontend captures it using HTML form elements and passes it to JavaScript functions. The result is then displayed dynamically on the screen, offering a smooth and instant experience without page reloads.

Use of CSS and Bootstrap

To enhance the look and responsiveness of the platform, Bootstrap is used as the core styling framework, along with custom CSS for additional design tweaks. Bootstrap helps in building a mobile-first layout that adapts well across desktops, tablets, and smartphones. Its grid system, cards, buttons, forms, and responsive utilities are extensively used.

Custom CSS is applied for color themes, text styles, hover effects, and to fine-tune margins and paddings beyond Bootstrap defaults, ensuring a clean and professional design.

Responsive Design for Different Screen Sizes

WiseFinance is fully responsive and supports all screen sizes by:

- Using Bootstrap's grid system (container, row, and col) to adjust layout across devices.
- Applying media queries in CSS to manage font size, padding, and layout changes.
- Making all input forms and result boxes fluid and scroll-friendly on mobile devices.
- Hiding or collapsing certain elements (like the sidebar or extended navbar) on smaller screens for better UX.

HTML and CSS Design Structure

HTML provides the basic structure including:

- Navigation bar
- Calculator forms (input fields, buttons)
- Query submission section
- Dashboard layout
- Profile cards
- CSS & Bootstrap manage:
 - Page layout
 - Spacing and alignment
 - Typography
 - Interactive elements (buttons, forms, hover effects)
 - Color scheme and consistency

All major components (calculators, queries, login forms) are organized in separate sections using semantic HTML (<section>, <form>, <div>, etc.), ensuring code readability and maintainability.

Designing the User Interface (UI) and Interactions

The UI is designed to be minimalist and functional. Key principles include:

- Clear call-to-action buttons (like “Calculate”, “Submit Query”)
- Easy navigation with a top navbar
- Interactive messages (success, error, alerts) using Bootstrap’s alert classes
- Feedback after every action (e.g., query submitted, login success, invalid input)

Forms validate inputs with JavaScript to guide users and prevent errors. Animations and transitions are lightly used for better user engagement.

User Journey from Home Screen

Once the user reaches the home screen, they can:

1. Read about WiseFinance and explore its features.
2. Choose to log in or register.
3. After login, access:
 - Financial calculators to perform instant calculations
 - Query section to ask finance-related questions
 - Dashboard to view query history and responses
 - Profile page to manage account details
4. Optionally log out securely when done

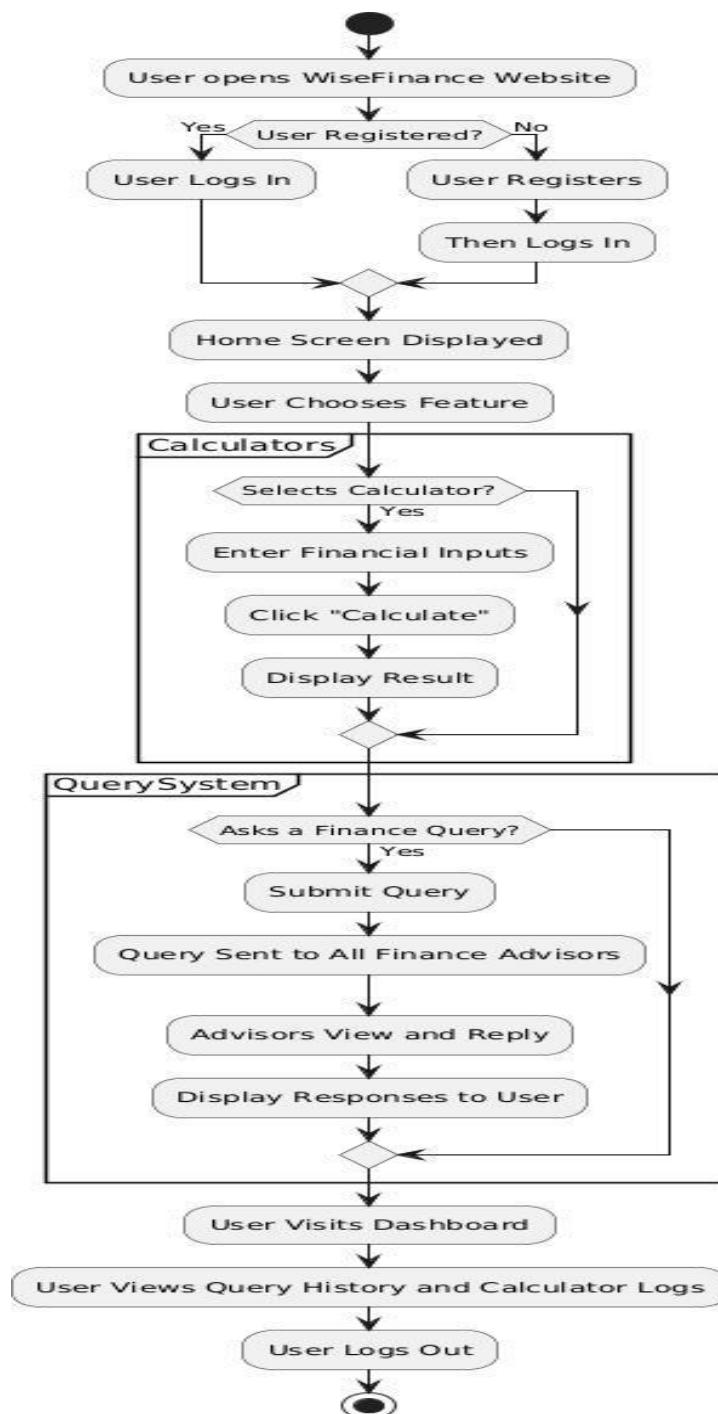


Fig 3.1 Flow chart

3.1.1 Data Flow Diagram

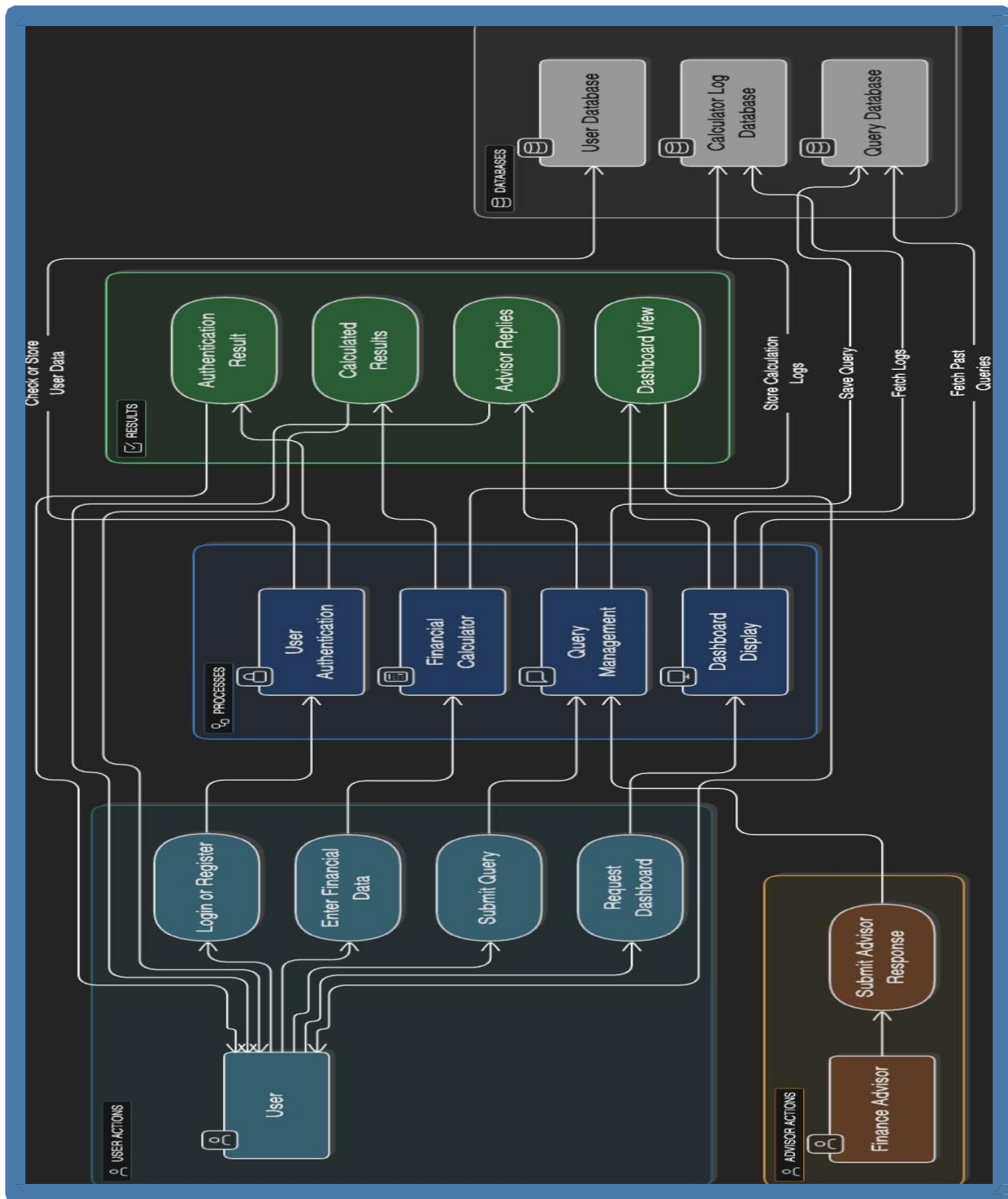


Figure 3.1 Data Flow Diagram

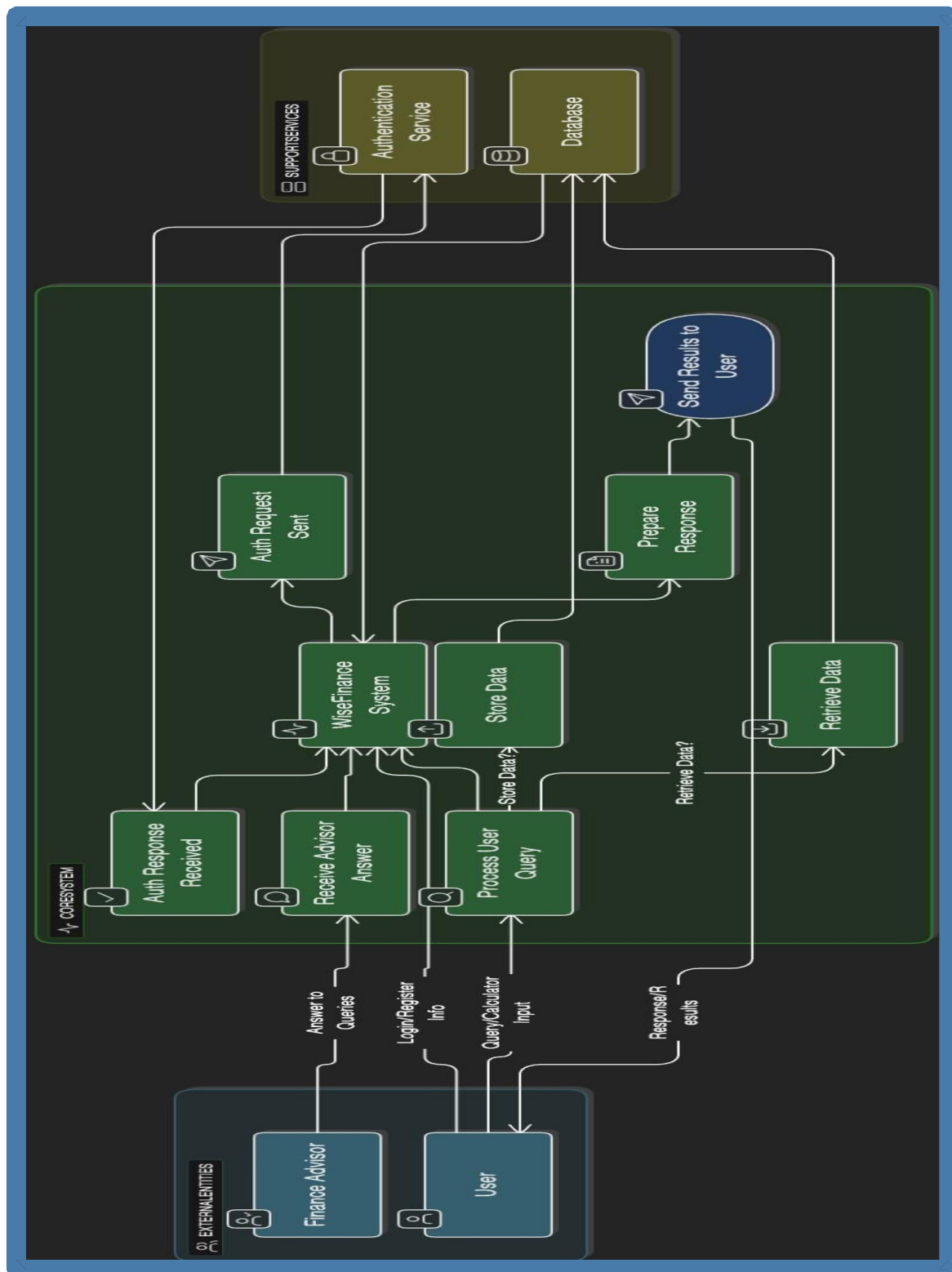


Figure 3.2 Data Flow Diagram

3.2 Back end design

To build a robust, scalable, and secure backend for WiseFinance, the following technologies are utilized:

- JavaScript (Node.js) – For server-side logic and processing.
- Firebase Firestore – A real-time NoSQL cloud database to store and retrieve user data, queries, advisor replies, and calculator logs.
- Firebase Authentication – To manage secure user sign-ups, logins, and third-party OAuth (Google Sign-In).
- Express.js (optional) – If API routing is needed for advanced scenarios.
- Firebase SDK – Used in both frontend and backend scripts for seamless integration with Firebase services.

Purpose of Backend in WiseFinance:

The backend acts as the brain of the web application, performing the following responsibilities:

- Authenticate and authorize users using Firebase Auth.
- Store user data (registration info, calculator logs, queries).
- Handle real-time advisory communication (advisor replies).
- Persist calculator results (for dashboard/reuse).
- Ensure secure and validated communication between client and database.

Backend Data Flow

The typical backend flow is structured in the following way:

1. User Authentication:

- Users register/login through Firebase Authentication.
- Firebase handles user tokens and maintains session data securely.
- Google Sign-In is also integrated for smoother login.

2. Calculator Operations:

- Users input financial data (e.g., NPV, SIP values).
- Frontend JavaScript validates the input.
- The result is calculated either on frontend or can be logged to backend via Firestore.
- Backend stores inputs and results in the CalculatorLogs collection.

3. Query Submission & Advisory Communication:

- Logged-in users can post finance-related questions via a web form.
- The data (query title, description, timestamp, user ID) is sent to Firestore's UserQueries collection.
- Advisors (special type of users) can log in and respond to queries.
- Their responses are updated in the same Firestore document or a sub-collection like Answers.

4. Dashboard and Profile Page:

- When a user accesses the dashboard, a real-time listener fetches data:
 - From CalculatorLogs for past calculations.
 - From UserQueries for past queries and answers.
- This makes the interface interactive and up to date without requiring manual page refreshes.

Firestore Cloud Functions (Optional Backend Logic)

Firestore Functions can be used to:

- Send email alerts when an advisor replies.
- Monitor abuse or spam content in queries.
- Notify inactive advisors of pending queries.

Why Firestore Was Chosen for Backend?

- Real-time capabilities – Perfect for finance dashboard and advisor updates.
- Secure authentication – Firestore Auth handles all common login mechanisms.
- No backend server required – Makes deployment easy and scalable.
- Integrates easily with frontend JavaScript – Fast implementation with Firestore SDK.

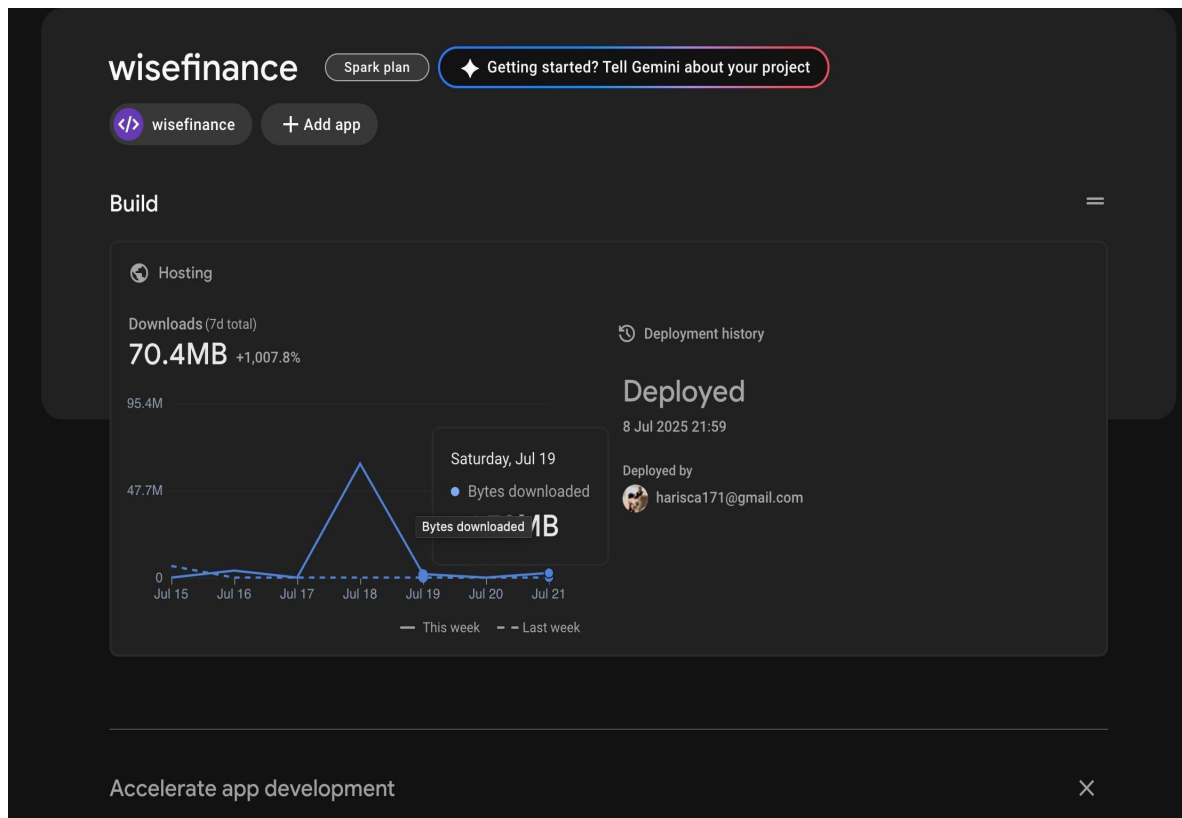


Figure 3.3 Firebase Console

Chapter 4 Implementation

In developing WiseFinance, a variety of JavaScript libraries were used to enhance both frontend interaction and backend processing. Libraries such as **Chart.js** were integrated to visually represent financial calculations like SIP, NPV, and ROI through interactive graphs and charts. For authentication and database operations, **Firestore SDK libraries** (Firestore Auth and Firestore) were utilized, allowing secure user logins (including Google Sign-In), and real-time data storage and retrieval without setting up a custom backend. Additionally, **Bootstrap** was incorporated for responsive and consistent styling, ensuring the UI remains user-friendly across various devices and screen sizes.

Development Tools and Their Usefulness

To build and manage the project efficiently, tools like **Visual Studio Code (VS Code)** were used as the primary code editor, offering powerful extensions, syntax highlighting, and Git integration. **GitHub** was employed for version control, collaboration, and backing up the project remotely. For UI prototyping and design validation, **Figma** was used during the initial phases to plan the user journey and layout before development. Furthermore, **Postman** was useful in testing API calls and interactions during Firestore function integration. Each of these tools contributed to a faster, more organized, and reliable development workflow.

Deployment Strategy

The final application was deployed on **Vercel**, a cloud platform that supports front-end frameworks and static site generation. Vercel was selected for its simplicity, fast deployment capabilities, and seamless GitHub integration. With each push to the GitHub main branch, Vercel automatically redeploys the updated site, allowing continuous delivery. Firestore Authentication and Firestore run in the background, ensuring all user data and calculations are handled in real-time without additional server-side setup. The deployment architecture is completely serverless, which reduces maintenance and scaling efforts while maintaining high performance and availability for end users.

Communication between Frontend and Backend

All communication is handled using Firebase SDK, which simplifies backend communication.

Data Sending (Frontend → Backend):

js

```
import { getFirestore, collection, addDoc } from "firebase/firestore";
```

```
const db = getFirestore(app);
await addDoc(collection(db, "UserQueries"),
  { userId: currentUser.uid,
    queryText: userInput,
    timestamp: new Date(),
  });
```

Data Retrieval (Backend → Frontend):

js

```
import { getDocs, collection } from "firebase/firestore";
```

```
const querySnapshot = await getDocs(collection(db, "UserQueries"));
querySnapshot.forEach((doc) => {
  console.log(doc.id, "=> ", doc.data());
});
```

Real-time Update Example:

js

```
import { onSnapshot, collection } from "firebase/firestore";
```

```
onSnapshot(collection(db, "UserQueries"), (snapshot) =>
  { snapshot.docs.forEach((doc) =>
    { console.log("Updated query:", doc.data());
    });
  });
```

Backend Security Rules

- Firestore security rules ensure only authenticated users can read/write their data.
- Advisors can access more queries using role-based access (if enabled).
- Each user can only access or modify their own logs/queries.

js


CopyEdit

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents
    { match /UserQueries/{docId} {
      allow read, write: if request.auth != null && request.auth.uid ==
resource.data.userId;
    }
    match /CalculatorLogs/{docId} {
      allow read, write: if request.auth != null && request.auth.uid ==
resource.data.userId;
    }
  }
}
```





Figure 4.1 Landing Page

The landing page of your WiseFinance web app greets users with a clean, responsive UI that introduces the platform's mission—helping users manage and calculate their finances efficiently. It features navigation to key tools like SIP, NPV, and ROI calculators, along with a prominent login/register option for personalized access. Eye-catching visuals, Bootstrap-based layout, and a professional color scheme enhance user trust and encourage engagement from the first visit.



Education
Empowering minds through knowledge and learning.




Contact Us
Get in touch with our team for more information.

Sign Up


Already have an account? [Login](#)

Figure 4.2 Sign Up Page




Home Logout Profile

Investment and Returns




SIP Calculator

Make informed investment decisions with our SIP calculator. Estimate your potential returns when investing systematically over time, helping you plan for your financial goals.




Lumpsum Calculator

Discover the power of a lump sum investment with our calculator. Calculate the future value of your one-time investments, providing insights into how your money can grow over time.



FD Calculator


Take control of your fixed deposits with our calculator. Determine your interest earnings and maturity amount with ease, enabling you to maximize your savings and make sound financial choices.



RD Calculator


Calculate your investment returns for all your Recurring Deposit investments easily using our Recurring Deposit(RD) Calculator.

Coming Soon




Mutual Fund Returns Calculator

Evaluate the performance of your investments using our mutual fund returns calculator. Estimate your potential gains and track the growth of your portfolio to make informed investment decisions.




Simple Interest Calculator

Simplify your financial calculations with our simple interest calculator. Easily determine the interest earned or paid on a principal amount, helping you manage your loans and savings effectively.




Compound Interest Calculator

Harness the power of compounding with our compound interest calculator. Compute your



Future Value Calculator

Plan your financial future by calculating the growth of your investments over time with our Future Value



Present Value Calculator

Plan your financial future by calculating the growth of your investments over time with our Future Value

Figure 4.3 home page

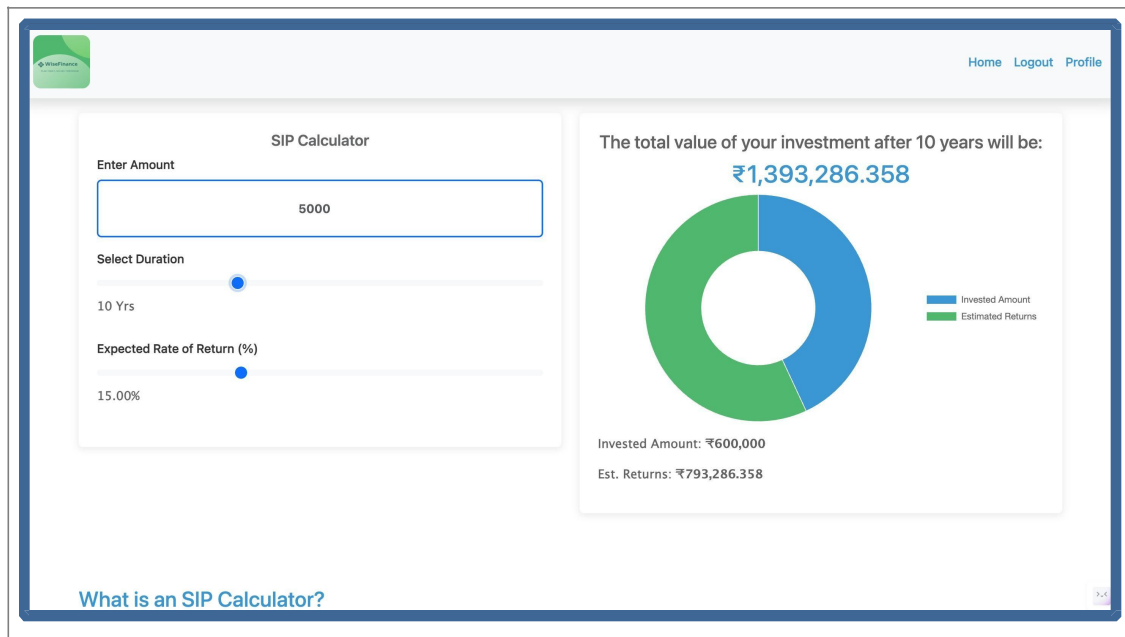


Figure 4.4 SIP calculator page

Net Present Value Calculator

Net Present Value (NPV)

Initial Investment: Enter initial investment

Discount Rate (%): Enter discount rate

Cash Flows (comma separated): Enter cash flows, e.g., 1000, 2000, 1500

Calculate NPV

FINANCE CALCULATORS: Simplify your financial calculations with our suite of calculators for SIP, EP, etc.

SERVICES: SIP Calculator

USEFUL LINKS: Help Center

Figure 4.5 NPV calculator page

Conclusion and Future Enhancement

The **WiseFinance** project successfully demonstrates a responsive, interactive web application that simplifies financial decision-making through a variety of calculators such as SIP, FD, NPV, and ROI. By combining user-friendly frontend design with robust backend functionality, this project bridges the gap between users and complex financial concepts, empowering individuals to make informed choices without needing deep financial knowledge. The integration of Firebase Authentication ensures secure user access, while tools like speech recognition and text-to-speech enhance accessibility for all users, including visually impaired individuals.

This platform serves as a freemium financial assistant where users can also connect with financial advisors. The UI adapts seamlessly across devices, ensuring a smooth experience on mobile and desktop. Although the current version delivers a core set of features, it lays the groundwork for future enhancements such as real-time advisor chats, AI-driven financial advice, and multilingual support to make finance guidance more inclusive and personalized. Overall, **WiseFinance** is a practical solution developed with scalability and real-world application in mind, solving real challenges with simple technology.

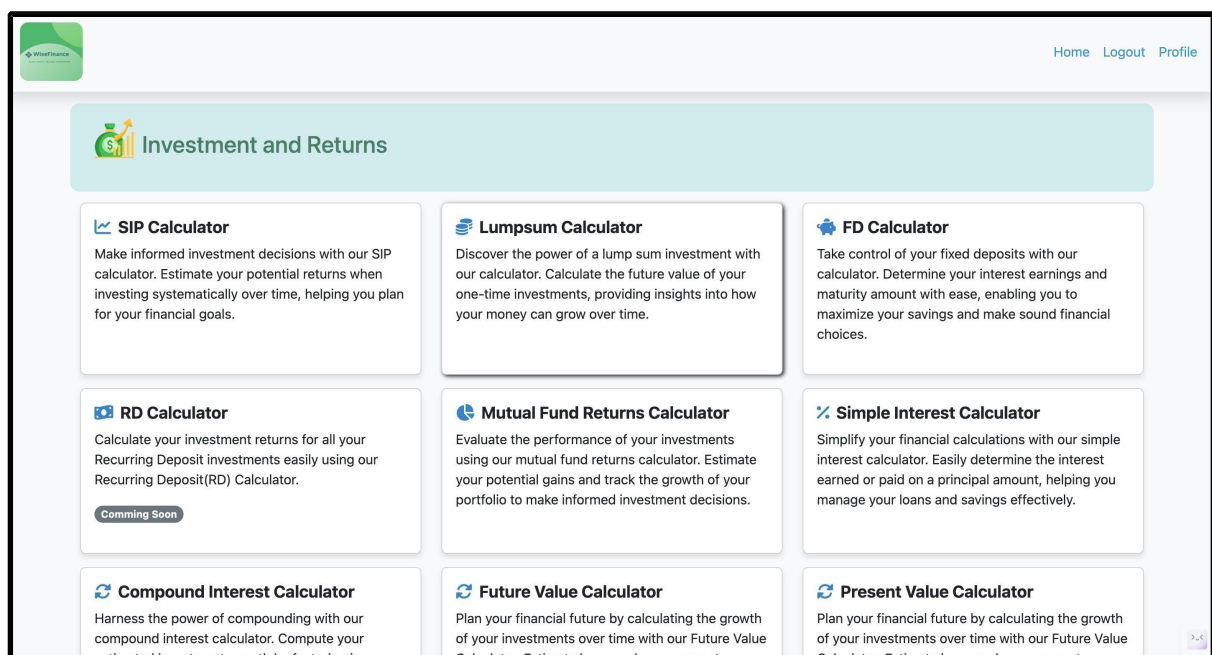
Future Enhancements


1. **AI-Powered Financial Advisor Chatbot:** Integrate an AI bot that can give personalized finance tips and guidance using NLP based on user input and financial data.
2. **Professional Advisor Portal:** Allow finance professionals to create verified accounts and answer user questions, adding a human layer of expert support.
3. **Multilingual Support:** Include regional languages so that non-English users can comfortably interact with the platform.
4. **User Financial History Dashboard:** Store and display past calculation history and financial plans for recurring users.
5. **Advanced Analytics:** Offer data visualizations, predictive modeling, and investment tracking features for in-depth financial analysis.

References

- [1] Adam Freeman, *Pro JavaScript for Web Apps*, 2014, Book, <https://www.apress.com/gp/book/9781430263913>
- [2] Richard Bovet & Mark Hines, *Firebase Essentials*, 2020, eBook, <https://firebase.google.com/docs>
- [3] Steven Holzner, *HTML, CSS, JavaScript and jQuery*, 2014, Book, ISBN: 978-1118907443
- [4] Python Software Foundation, *SpeechRecognition Library Documentation*, 2023, Web Reference, <https://pypi.org/project/SpeechRecognition/>
- [5] OpenAI, *GPT and AI Voice Interfaces for Assistive Technology*, 2023, Research Article, <https://openai.com/research>
- [6] Google Cloud, *Deploying Web Apps with Firebase Hosting*, 2024, Web Guide, <https://firebase.google.com/docs/hosting>

Appendix A



Home Logout Profile

SIP Calculator

Enter Amount

5000

Select Duration

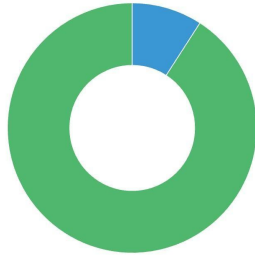
25 Yrs

Expected Rate of Return (%)

15.00%


The total value of your investment after 25 years will be:

₹16,420,368.676



Invested Amount: ₹1,500,000
Est. Returns: ₹14,920,368.676

[What is an SIP Calculator?](#)

Home Logout Profile

SIP Calculator FAQs

What benefits does an SIP return calculator offer to investors?

^

It helps you calculate the return on your investment based on the inputs given. It let you calculate return under different investment scenarios and compare them so that you can adjust your financial goals accordingly.

How do I use a SIP calculator for investment planning?

v

Are SIPs similar to mutual funds?

v

Can I modify my SIP amount?

v

What are the benefits of SIP?

v

Can I start SIP at any time or when the market is high?

v