Micro-Investment Insights for Smarter Financial Decisions

SHALENI K
SHARVESH PRASATH S
THASMIA R
LAVANYA A

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING
ADHIYAMAAN COLLEGE OF ENGINEERING
(AN AUTONOMOUS INSTITUTION)
DR.M.G.R NAGAR, HOSUR-635130

ABSTRACT

In today's fast-paced and dynamic financial world, young individuals particularly students often struggle to access and understand smart investment opportunities due to their limited income, lack of financial literacy, insufficient guidance, and fear of financial risks. This lack of awareness prevents them from building early saving habits and securing long-term financial stability. To address this challenge, the proposed project introduces a Micro-Investment Recommendation System designed specifically for students.The system aims to encourage early financial planning and disciplined investment practices by simplifying complex financial concepts and making them more accessible. The application collects key financial inputs such as monthly income, saving goals, investment duration, and risk appetite, which are analyzed through built-in logical algorithms to generate personalized recommendations tailored to each user's profile.These recommendations may include options such as Systematic Investment Plans (SIPs), digital gold, or cryptocurrencies, allowing students to start investing even with small amounts. Additionally, the app offers real-time progress tracking, interactive dashboards, financial tips, and smart alerts to keep users informed, engaged, and motivated. By combining personalization, accessibility, and education, this project empowers students to manage their finances efficiently, develop responsible money management habits, and build long-term wealth from an early age.

CHAPTER 1 INTRODUCTION

1.1 OVERVIEW

This system is basically concerned with providing insights and recommendations for micro- investments to help users make smarter financial decisions. The need for this system arose because managing and analyzing multiple small investment options manually is quite a tough and time-consuming task. By computerizing and automating it, we are able to make the process more efficient and accurate. Handling large sets of financial data and user preferences manually is complex, but it becomes much easier and more reliable when the system is digitalized.

To be more specific, our system is designed in such a way that it collects user details such as investment amount, risk tolerance, and financial goals. The basic functions performed by our system include analyzing user preferences, recommending suitable investment options, and displaying live financial insights. These functions are handled with the help of the following modules in coding:

- **User Input Module** provides a way for users to enter their financial preferences and investment goals.

- **Analysis Module** processes the data and evaluates the best possible investment opportunities.

- **Recommendation Module** provides suitable micro-investment options such as stocks, gold, mutual funds, and SIPs.

- **Status Module** displays real-time performance updates of the investments.

- **Report Module** provides summarized insights and graphical representations of investment trends and returns.

1.2 OBJECTIVES

Promote Early Financial Planning:

- To encourage students to develop early financial planning habits and understand the importance of saving and investing.

Provide Personalized Investment Guidance:

- To offer customized micro-investment recommendations based on individual financial profiles, including income, savings goals, duration, and risk appetite.

Simplify Investment Decision-Making:

- To make investment planning easy and understandable through clear, accessible guidance and user-friendly tools.

Enable Real-Time Financial Tracking:

- To allow students to monitor their savings progress, investment growth, and overall financial performance through interactive dashboards.

Enhance Financial Awareness:

- To promote financial literacy and responsible money management among young individuals.

Develop an Engaging Digital Platform:

- To create a simple, secure, and user-friendly web application that makes investment planning engaging and convenient for students.

Support Long-Term Wealth Creation:

- To help students build long-term financial stability and wealth through consistent, informed, and goal-oriented micro-investments.

CHAPTER 2 LITERATURE SURVEY

[1] Alao, O. (2025) – "Data-Driven Financial Decision-Making for Minority Enterprises: Capital Access, Investment Strategies, and Creditworthiness Optimization".Alao's study focuses on data analytics for optimizing investment strategies and improving access to capital, especially for small investors and minority groups. The research underscores the role of financial technology (FinTech) tools in assessing user profiles, risk levels, and market conditions to recommend appropriate investment options.This aligns with the objective of developing a micro-investment system that supports young investors through personalized and data-backed recommendations.

[2] Study on Micro-Investment Platforms and Financial Inclusion (2024).This study examines how micro-investment platforms promote financial inclusion by enabling individuals, particularly youth and low-income earners, to invest small amounts regularly. The research found that such platforms remove traditional barriers to entry—such as high minimum investments and lack of financial literacy—thereby democratizing investment opportunities. This concept strongly supports the aim of the proposed system to motivate students to begin early financial planning with minimal investment amounts

[3] Adewale, T.T., Olorunyomi, T.D., & Odonkor, T.N. (2023) – "Big Data-Driven Financial Analysis: A New Paradigm for Strategic Insights and Decision-Making".This study explores how big data analytics has transformed financial analysis and decision-making. It highlights that financial institutions and investors can leverage data-driven insights to identify emerging market trends, risk factors, and consumer behavior patterns that contribute to better investment outcomes. The study emphasizes that by integrating data analytics into financial systems, decision-makers can minimize uncertainty and improve strategic planning, resulting in enhanced financial growth and stability. This research lays the foundation for incorporating analytical logic in investment recommendation systems to enhance accuracy and personalization.

[4] Financial Innovation and Analytics (2023)This research discusses the impact of technological innovations—including artificial intelligence, predictive analytics, and automation—on financial decision-making. It demonstrates that integrating analytical tools helps users receive personalized investment suggestions based on real-time data and behavioral insights. Predictive analytics enables proactive financial planning by forecasting potential market trends and risks. Although the proposed micro-investment system does not rely on machine learning, it draws inspiration from such studies by incorporating logic-based algorithms for accurate and reliable recommendations.

CHAPTER 3 SYSTEM ANALYSIS

3.1. EXISTING SYSTEM:

Several existing systems in the financial technology sector have revolutionized the way individuals approach micro-investing by providing accessible, user-friendly, and technology- driven platforms. Groww is one of India's most popular investment applications, offering users the ability to invest in mutual funds, Systematic Investment Plans (SIPs), stocks, Exchange- Traded Funds (ETFs), and digital gold. Its intuitive interface, low entry barriers, and transparent processes make it ideal for beginners who wish to start investing with minimal amounts. The platform also provides educational resources and analytical tools to help users make informed investment decisions.

Acorns, based in the United States, is a pioneer in micro-investing that focuses on simplicity and automation. Its unique Round-Ups feature automatically invests the spare change from users' daily transactions into diversified portfolios, enabling consistent and effortless investing. The platform promotes financial discipline among young users by turning everyday spending into an opportunity for wealth creation.

Smallcase is another innovative Indian platform that offers thematic and portfolio-based investing by grouping stocks and ETFs into curated "smallcases" based on specific investment ideas or market trends. This approach allows users to invest systematically and diversify their portfolios with minimal research effort.

Similarly, Paytm Money provides a secure and comprehensive platform that supports investments in mutual funds, SIPs, NPS, and digital gold. With its integration into the Paytm ecosystem, it ensures convenience, transparency, and accessibility, especially for mobile users.

Collectively, these existing systems highlight the growing trend of digital financial inclusion, where technology simplifies investment processes, reduces entry barriers, and empowers individuals to start investing with small, manageable amounts. Each of these platforms demonstrates how automation, minimal investment thresholds, and intuitive interfaces can encourage users—especially beginners—to participate in wealth-building activities.

3.2. PROPOSED SYSTEM:

The proposed system is a Smart Analytics-Based Micro-Investment Recommendation Platform designed to assist students and young investors in developing structured financial planning habits. It aims to bridge the gap between financial awareness and investment accessibility by providing personalized, data-driven insights and user-friendly investment management features.

The system collects user-specific financial data such as monthly income, savings goals, investment duration, and risk appetite to create a detailed financial profile. Based on this information, it utilizes intelligent analytics and built-in decision-making algorithms to generate personalized investment suggestions. These recommendations may include Systematic Investment Plans (SIPs), mutual funds, digital gold, or other small-scale investment options suited to the user's financial capacity and objectives.

A key component of the proposed system is its visual dashboard, which provides users with a clear and interactive interface to track investments in real time, monitor their financial growth, and view projections based on ongoing performance. The dashboard also displays financial tips, alerts, and notifications to keep users informed and motivated toward achieving their savings goals.

Additionally, the system plans to integrate smart analytics models that continuously learn from user behavior and market trends to refine recommendations over time. These models will analyze transactional patterns and investment behavior to provide deeper insights, such as ideal investment proportions, optimal fund choices, and suitable timeframes for returns.

The proposed system's architecture ensures data security and user privacy through encryption and secure authentication mechanisms, allowing safe storage and management of user information.

In essence, this proposed platform combines financial education, intelligent recommendation, and real-time monitoring to help students and new investors make confident, well-informed, and consistent investment decisions—paving the way for early financial independence and long-term wealth creation.

3.3. PROPOSED SOLUTION

The proposed solution aims to create a user-friendly micro-investment platform that empowers students and young investors to make informed financial decisions with minimal effort and investment. It focuses on simplifying the process of financial planning by integrating personalized recommendations, real-time tracking, and insightful guidance within a single, accessible application.

The system will collect user-specific data, including monthly income, savings goals, investment duration, and risk appetite, through an interactive and secure input interface. This data will then be analyzed using built-in logical rules and decision-making techniques to identify suitable investment options that align with each user's profile. Based on these insights, the platform will recommend appropriate micro-investment opportunities such as Systematic Investment Plans (SIPs), mutual funds, digital gold, or other low-risk instruments.

To enhance usability, the solution includes a visual dashboard that allows users to track their investments in real time, monitor progress toward financial goals, and view projections of future growth. The system will also provide personalized financial tips, alerts, and notifications to keep users informed and motivated to maintain consistency in their investment habits.

Furthermore, the platform will analyze user behavior and preferences to refine recommendations over time, ensuring adaptability to changing goals and market conditions.

Overall, the proposed solution provides a comprehensive, secure, and educational digital platform that encourages early financial discipline, promotes responsible money management, and enables students to build long-term financial stability through consistent micro-investments.

3.4. IDEATION & BRAINSTORMING

The ideation and brainstorming phase formed the backbone of the Micro-Investment Recommendation Platform project. This stage focused on understanding the financial challenges faced by students and young individuals and conceptualizing a simple yet impactful digital solution to promote financial literacy, savings, and early investment habits. The objective was to design an accessible, educational, and personalized system that bridges the gap between students and smart financial planning.

1. Problem Identification

During the initial phase, team discussions, surveys, and research were conducted to understand the real issues students face in managing and investing their money. Some of the key problems identified were:

- Lack of financial awareness and limited understanding of investment options among students.
- Difficulty in managing small monthly savings and allocating funds effectively.
- Overwhelming complexity of existing investment platforms not suited for beginners.
- Absence of personalized guidance or micro-level financial recommendations.
- Fear of financial risks and uncertainty about where to start investing.
- Limited tools for tracking savings progress and visualizing financial growth.
- These findings highlighted the need for a student-focused, easy-to-use digital platform that promotes disciplined saving, personalized investment recommendations, and real-time financial awareness.

2. Idea Generation

In the brainstorming sessions, multiple creative ideas were explored using techniques like mind mapping, “What If” analysis, and user persona mapping to better understand students’ financial behavior. Some of the key ideas generated were:

- Personalized Micro-Investment System: Suggests investment options like SIPs, mutual funds, or digital gold based on income, goals, and risk level.
- Financial Dashboard: Displays real-time progress, savings goals, and visual growth charts.
- Goal-Based Savings Tracker: Helps users set and achieve short-term and long-term financial objectives.
- Smart Alerts & Notifications: Sends reminders for savings contributions and financial tips.
- Secure Authentication & User Profiles: Ensures safe login, profile management, and personalized recommendations.

The team ensured that the ideas were not only innovative but also technically feasible within the chosen technology stack (React, TypeScript, Node.js, Express, PostgreSQL).

3. Evaluation and Selection

All proposed ideas were analyzed based on feasibility, usability, and impact. The most practical and beneficial features were shortlisted for the initial prototype.

Selection criteria included:

- Can it be implemented effectively with the current technology stack (HTML, CSS, JavaScript, Node.js, MySQL)?
- Does it improve financial awareness and engagement among students?
- Is it simple and scalable for a wide range of users?
- The selected modules were User Authentication, Student Input, Recommendation Engine, and Dashboard Visualization, forming the foundation of the system’s core design.

4. Concept Development

Once the core concepts were finalized, the team initiated the concept development phase, focusing on transforming ideas into structured system components. Key activities included:

- Creating flow diagrams to outline the user journey from registration to investment recommendation.
- Designing database schemas for storing financial profiles, investment categories, and user transactions.
- Developing wireframes and mockups for the web interface with emphasis on simplicity and engagement.
- Defining recommendation logic using rule-based decision techniques for personalized output.
- Prioritizing features for the first version, such as secure login, input collection, and basic investment suggestions.
- Collaborative brainstorming was carried out using tools like Miro, Figma, and Lucidchart to visualize user interactions and refine the prototype. Feedback from mentors and peers further enhanced the usability and clarity of the final design.

3.5. PROBLEM SOLUTION FIT

The Micro-Investment Recommendation System was conceptualized to address the major challenges faced by students and young individuals in managing their finances and initiating early investments. It aims to make financial planning accessible, personalized, and engaging for beginners.

Identified Problems

1. **Lack of Financial Awareness:** Many students have limited understanding of investment options and long-term financial planning.
2. **Low Income and Savings Capacity:** Students often struggle to save or invest due to limited budgets and irregular income.
3. **Complexity of Existing Platforms:** Most financial applications are designed for experienced investors, making them difficult for beginners to use.
4. **Fear of Financial Risks:** Uncertainty about investment returns discourages students from exploring opportunities.
5. **Absence of Personalized Guidance:** Current systems do not offer tailored investment suggestions based on individual financial profiles.

Proposed Solutions

1. **Personalized Investment Recommendations:** Suggests suitable micro-investment options such as SIPs, mutual funds, or digital gold based on income, goals, and risk level.
2. **Digital Dashboard:** Provides real-time tracking of savings, goals, and investment progress.
3. **Simplified Guidance:** Offers clear, beginner-friendly explanations for financial planning and decision-making.
4. **Smart Alerts and Notifications:** Sends reminders for savings contributions and investment updates.
5. **Educational Insights:** Shares financial tips and knowledge to build awareness and responsible money habits.

Fit Analysis

The Micro-Investment Recommendation System aligns closely with the identified financial challenges faced by students. Its features are designed for ease of use, impact, and practical value, helping students make confident financial decisions. By combining personalized recommendations, progress tracking, and educational support, the platform bridges the gap between limited financial knowledge and active investment participation.

3.6. ARCHITECTURE DESIGN:

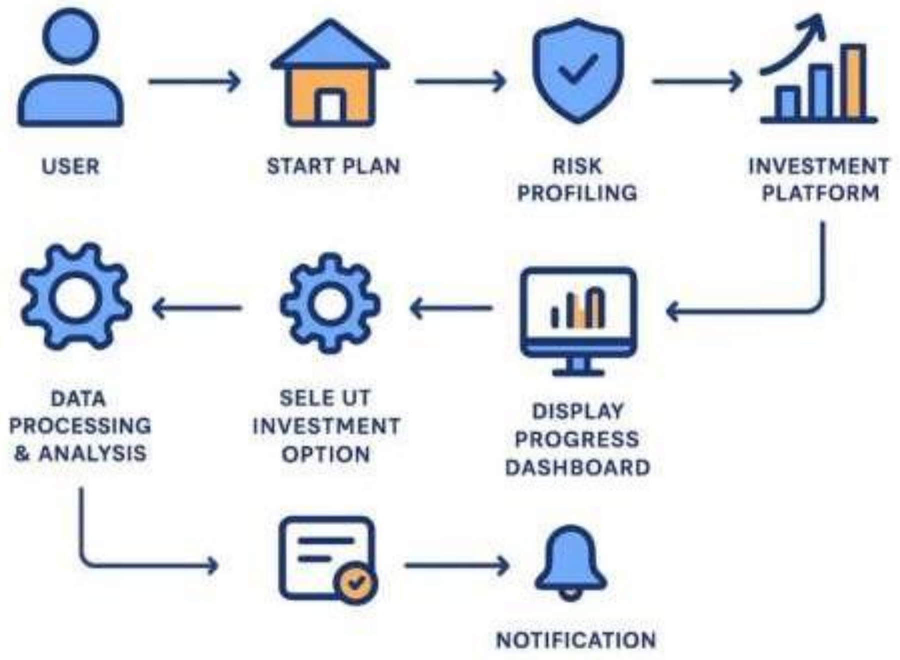

USER
START PLAN
RISK PROFILING
INVESTMENT PLATFORM
DATA PROCESSING & ANALYSIS
SELE UT INVESTMENT OPTION
DISPLAY PROGRESS DASHBOARD
NOTIFICATION

Figure 1: Model Architecture

The figure shown above represents the Solution Architecture that we made use of in our Project.

The Project Workflow Diagram represents the complete functional flow of the Micro-Investment Recommendation System, illustrating how users interact with different components from login to investment tracking. The system is designed to ensure a seamless, structured, and user-friendly experience for students seeking personalized investment guidance.

The process begins with the User Login, where users securely access the system through authentication. After successful login, the user is directed to the Home Page, which serves as the main interface to start the investment journey. From there, users can initiate a plan by selecting the Start Plan option, which leads to Risk Profiling — a step designed to assess the user’s financial comfort level, investment goals, and savings capacity.

Next, the user proceeds to the Data Input stage, where essential financial details such as monthly income, savings goals, and investment duration are entered. These inputs are then processed through the Data Processing and Analysis module, where logical algorithms interpret the data to determine appropriate micro-investment options.

Once processed, the system allows the user to Select an Investment Option from various recommendations, such as Systematic Investment Plans (SIPs), digital gold, or mutual funds. This leads to the Investment Platform, which displays tailored investment suggestions and facilitates easy decision-making.

The Display Progress Dashboard provides real-time insights into savings growth, investment distribution, and goal achievement. Users can also Track Current Progress to monitor how their investments evolve over time. Additionally, the Notification Module sends reminders, financial tips, and alerts to keep users engaged and consistent in their investment activities.

Overall, this workflow ensures a smooth data flow between user actions and system responses. Each module — from user input to dashboard display — works cohesively to deliver a personalized, transparent, and efficient investment experience, empowering students to build disciplined saving habits and achieve long-term financial stability.

3.7. DESCRIPTION OF MODULES

3.7.1 User Interface Module

The User Interface (UI) module serves as the primary interaction layer between the user and the system. It includes pages such as User Login, Home Page, and the Dashboard, where users can easily navigate through different functionalities.

The design focuses on simplicity and accessibility to ensure that even users with minimal financial knowledge can use the platform effectively.

Through the UI, users can input financial details, select investment preferences, view personalized recommendations, and track their financial progress in a visually appealing and structured manner.

3.7.2 Data Input and Processing Module

In this module, users enter their basic financial information such as monthly income, desired savings goal, investment duration, and risk appetite.

The system performs validation checks to ensure the data is accurate and complete. Once validated, the data undergoes preprocessing and analysis, forming the foundation for subsequent modules like risk profiling and investment recommendation.

This ensures that every recommendation is based on accurate, up-to-date, and relevant information.

3.7.3 Risk Profiling Module

The Risk Profiling module plays a vital role in assessing the user’s investment behavior and tolerance for financial risk.Based on inputs such as income level, financial goals, and investment duration, it categorizes users into different risk levels — conservative, moderate, or aggressive.This classification helps the system generate investment recommendations that align with the user’s comfort level, financial capacity, and long-term objectives. It ensures that users are not exposed to unnecessary risks beyond their tolerance.

3.7.4 Investment Recommendation Module

Once the risk profile is established, this module uses rule-based logic to suggest the most suitable micro-investment options. Recommendations may include Systematic Investment Plans (SIPs), digital gold, or cryptocurrencies, depending on user goals and financial behavior. The module provides clear insights into expected returns, investment duration, and potential risks, helping users make informed financial decisions. It acts as a personal advisor, simplifying complex investment options into understandable insights.

3.7.5 Investment Platform Module

Once the risk profile is established, this module uses rule-based logic to suggest the most suitable micro-investment options. Recommendations may include Systematic Investment Plans (SIPs), digital gold, or cryptocurrencies, depending on user goals and financial behavior. The module provides clear insights into expected returns, investment duration, and potential risks, helping users make informed financial decisions. It acts as a personal advisor, simplifying complex investment options into understandable insights.

3.7.6 Notification Module

The Notification module is designed to keep users updated and engaged throughout their investment journey. It provides timely alerts and reminders for important activities such as periodic contributions, investment renewals, and performance updates. Notifications can also include educational tips or suggestions to enhance financial literacy. This helps maintain user involvement and encourages consistency in investment habits.

3.7.7 Progress Tracking Module

This module offers a comprehensive overview of the user's investment performance through a dynamic dashboard. It visually represents financial progress, returns on investments, and goal completion percentages using charts and analytics. Users can monitor their achievements, adjust strategies if needed, and make data-driven decisions. The tracking system promotes transparency and motivates users to continue saving and invest

3.8. DATAFLOW DIAGRAM:

The Data Flow Theory in this project illustrates how information moves within the investment recommendation web application. User inputs from the Login and Register pages are processed and shared across components such as Dashboard,and RecommendationChart using React's state, props, and context. Each component communicates efficiently to display investment recommendations, progress tracking, and notifications. This structured data flow ensures seamless interaction, quick updates, and a smooth user experience throughout the application

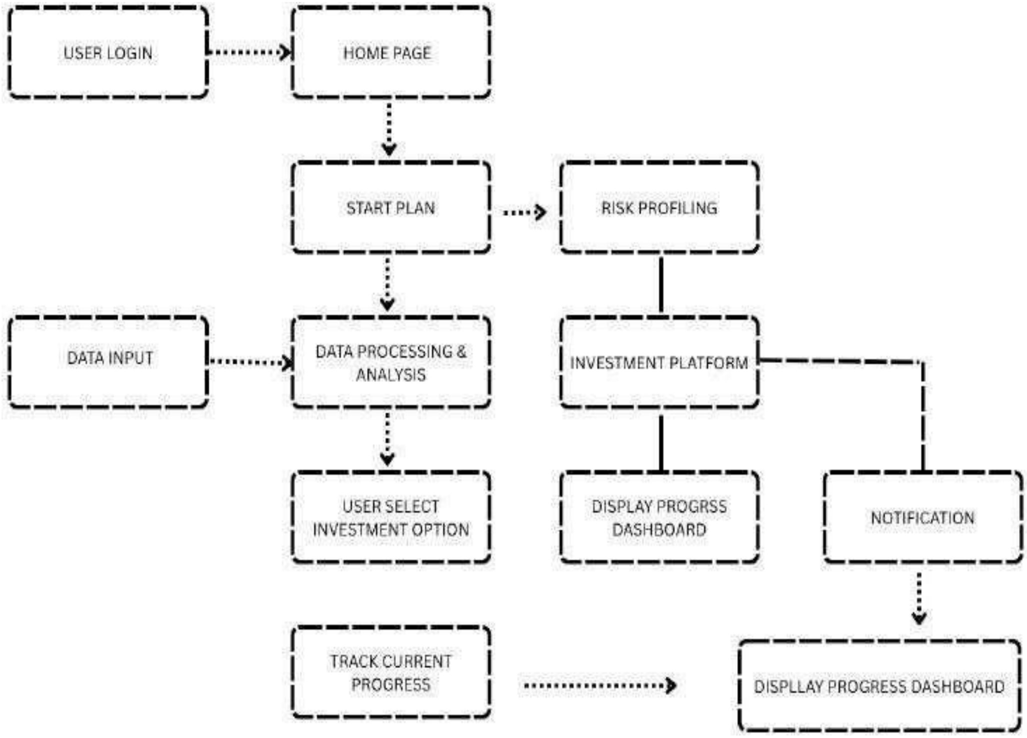

USER LOGIN
HOME PAGE
START PLAN
RISK PROFILING
DATA INPUT
DATA PROCESSING & ANALYSIS
INVESTMENT PLATFORM
USER SELECT INVESTMENT OPTION
DISPLAY PROGRSS DASHBOARD
NOTIFICATION
TRACK CURRENT PROGRESS
DISPLLAY PROGRESS DASHBOARD

Figure 2: Data Flow Diagram

CHAPTER 4 SYSTEM REQUIREMENTS

4.1. HARDWARE REQUIREMENT

1. **SYSTEM:** PC/Laptop with at least Intel i3 processor, 4 GB RAM, and 20 GB.

2. **GRAPHICS**: Minimum display resolution of 1366×768 to ensure smooth visualization of dashboards and charts.

3. **STABLE INTERNET CONNECTION**: Required for accessing APIs, performing real-time data fetching, and cloud-based analytics.

4. **SERVER:** Recommended hosting server with Intel i5 processor, 8 GB RAM, and 500 GB storage for production deployment.

5. **CLIENT DEVICES:** Desktop, laptop, tablet, or mobile device equipped with a modern web browser (Chrome, Edge, or Firefox)

4.2. SOFTWARE REQUIREMENTS

Development Tools

- **IDE / Code Editor**:Visual Studio Code – used for writing, debugging, and running the project.
- **Version Control:** Git and GitHub – for project version management and collaboration.
- **Terminal / Shell:** PowerShell or Command Prompt – for executing npm commands and starting the development server.

Frontend Technologies

- **Language:** JavaScript (ES6) with JSX – for developing interactive and dynamic web components.
- **Framework / Library:** React.js – used to build modular and reusable UI components such as Dashboard.jsx, Login.jsx, RecommendationChart.jsx, and NotificationBell.jsx.
- **Routing:** React Router DOM – for navigation between pages like Home, Login, Register, and Dashboard.
- **State Management:** React Context API – for managing and sharing state across components (e.g., NotificationContext.jsx).
- **Data Handling / API Communication:** Axios or Fetch API (via api.js) – for connecting to APIs and retrieving investment-related data.
- **UI / Styling:** Tailwind CSS or basic CSS modules – for building a clean and responsive user

interface.

- **Chart & Visualization Library:** Recharts – used in components like ProgressChart.jsx and RecommendationChart.jsx to display financial growth visually.

Testing & Execution

- **Browser Compatibility:**Google Chrome, Microsoft Edge, or Mozilla Firefox – for testing and running the web application.

- **Package Manager:**npm(Node Package Manager) – for installing project dependencies and running build/start scripts.

CHAPTER 5 IMPLEMENTATION

5.1 User Authentication Module:

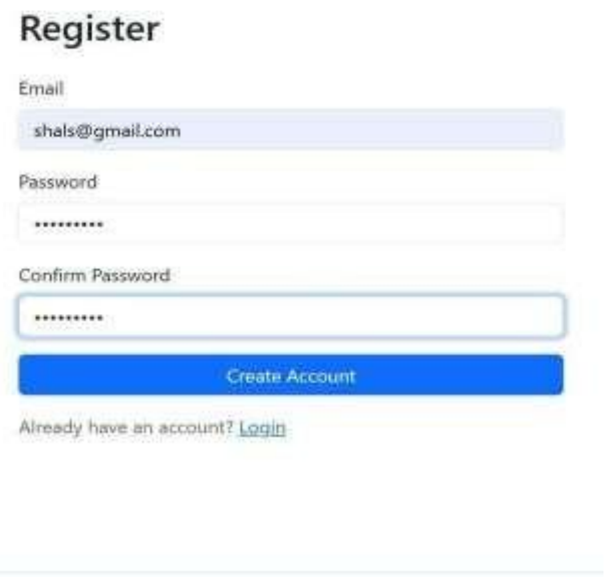
Purpose:

To provide a secure and simple login and registration process for users accessing the system.

Implementation:

The authentication module allows students to register with their credentials and log in using validated details. Input fields are validated using form-handling logic to prevent incomplete or invalid entries. React Router DOM is used for seamless navigation between the Login, Register, and Dashboard pages. Once the credentials are validated, users are redirected to the dashboard. The authentication state is managed using React's useState and Context API to maintain session persistence throughout navigation.

This ensures a smooth transition and secure access control while enhancing user convenience and experience.

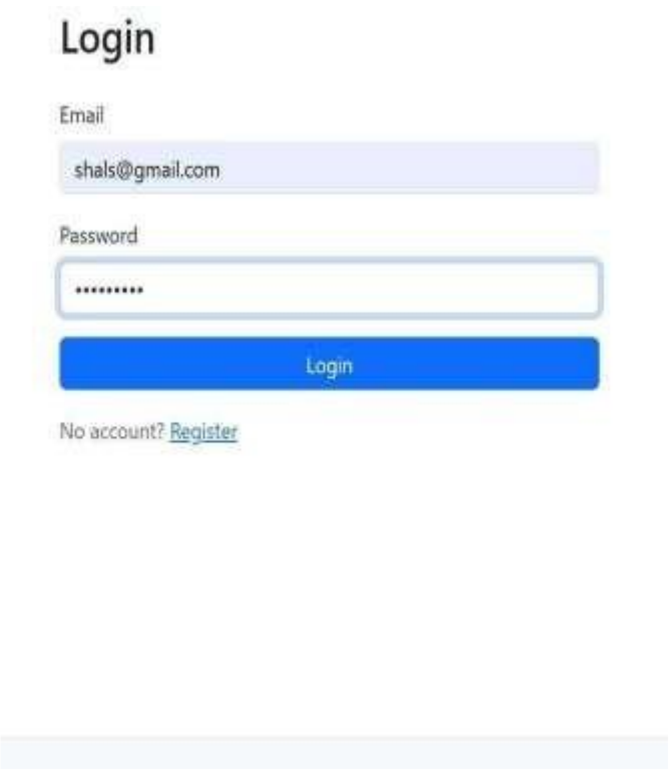


Register

Email: shals@gmail.com

Password: •••••••••

Confirm Password: •••••••••



Login

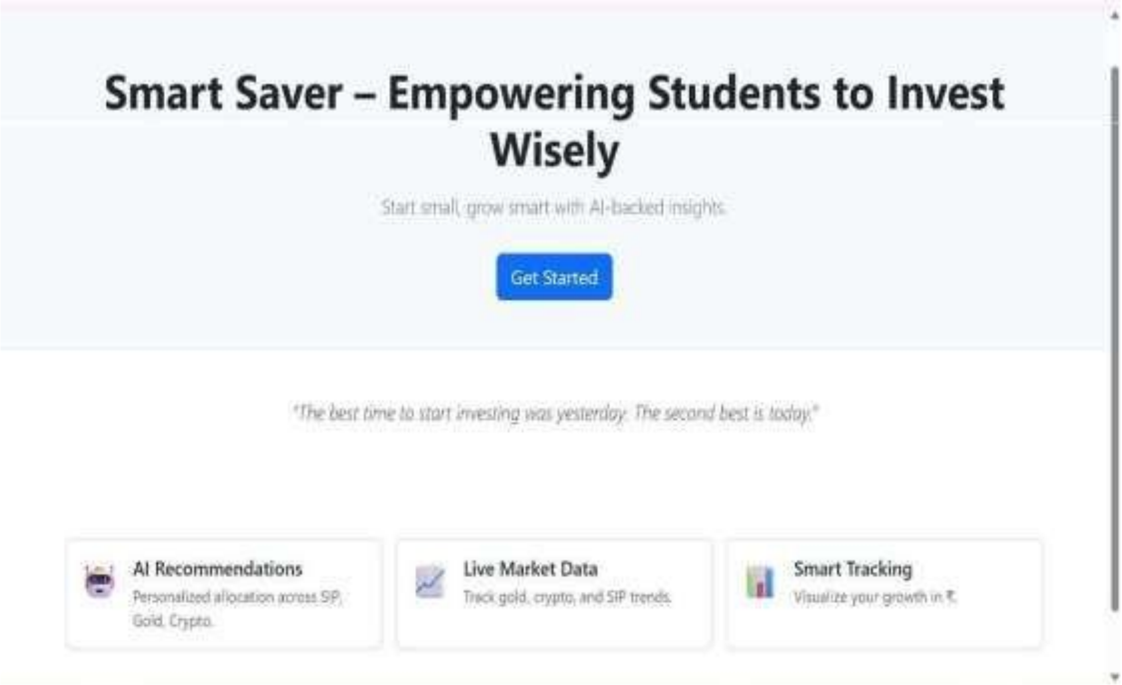
Email

shals@gmail.com

Password

•••••••••

No account? Register



Smart Saver – Empowering Students to Invest Wisely

Start small, grow smart with AI-backed insights.

"The best time to start investing was yesterday. The second best is today."

AI Recommendations
Personalized allocation across SIP, Gold, Crypto.

Live Market Data
Track gold, crypto, and SIP trends.

Smart Tracking
Visualize your growth in ₹.

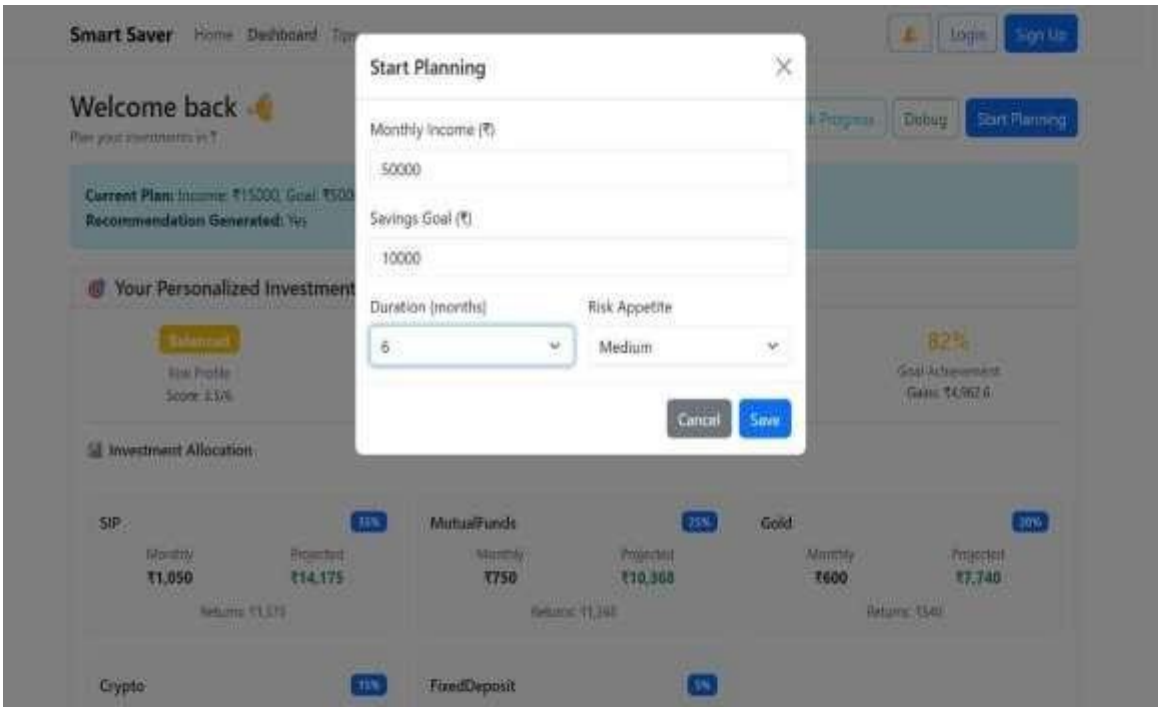
5.2 Student Input Module:

Purpose:

To collect and process user financial information for generating personalized investment recommendations.

Implementation:

This module includes a structured input form where students provide key details such as monthly income, saving goals, investment duration, and risk preference. Input validation ensures that users provide accurate and reasonable values. The collected data is stored using React's useState hooks and passed to the Recommendation Engine Module through props.The interface was designed to be intuitive and user-friendly, guiding students step by step. This helps reduce errors and ensures that financial data is captured accurately before generating recommendations.



Start Planning

Monthly Income (₹): 50000

Savings Goal (₹): 10000

Duration (months): 6

Risk Appetite: Medium

5.3 Recommendation Engine Module:

Purpose:

To generate customized micro-investment recommendations based on the user's profile and preferences.

Implementation:

This is the core logic of the system, where rule-based decision-making is applied. The system evaluates the provided data and allocates investments into categories such as Systematic Investment Plans (SIPs), Digital Gold, and Cryptocurrency according to predefined risk ratios.

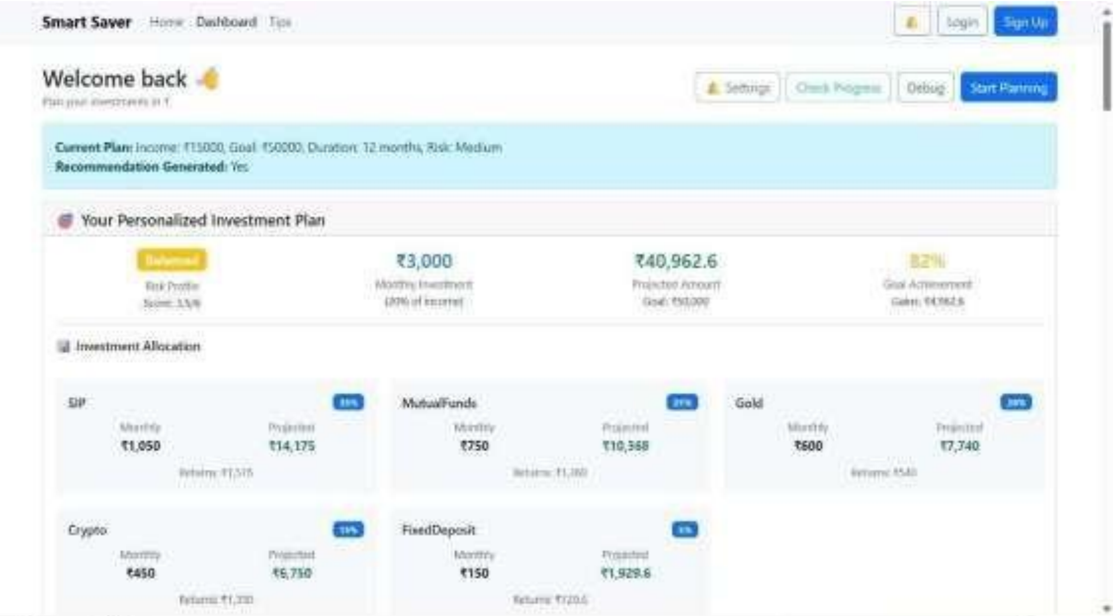
For instance:

Low Risk: 70% SIP, 20% Gold, 10% Crypto

Medium Risk: 50% SIP, 30% Gold, 20% Crypto

High Risk: 30% SIP, 20% Gold, 50% Crypto

The calculated results are displayed in the Recommendation Chart and Investment Card components, giving users a visual and descriptive representation of their personalized investment plan. The logic ensures that users receive suggestions best suited to their goals and comfort level with risk.



Welcome back

Plan your investments in ₹

Current Plan: Income: ₹15000, Goal: ₹50000, Duration: 12 months, Risk: Medium
Recommendation Generated: Yes

Your Personalized Investment Plan

Balanced	₹3,000	₹40,962.6	82%
Risk Profile Score: 3.5/8	Monthly Investment (20% of income)	Projected Amount Goal: ₹50,000	Goal Achievement Gains: ₹4,962.6

Investment Allocation

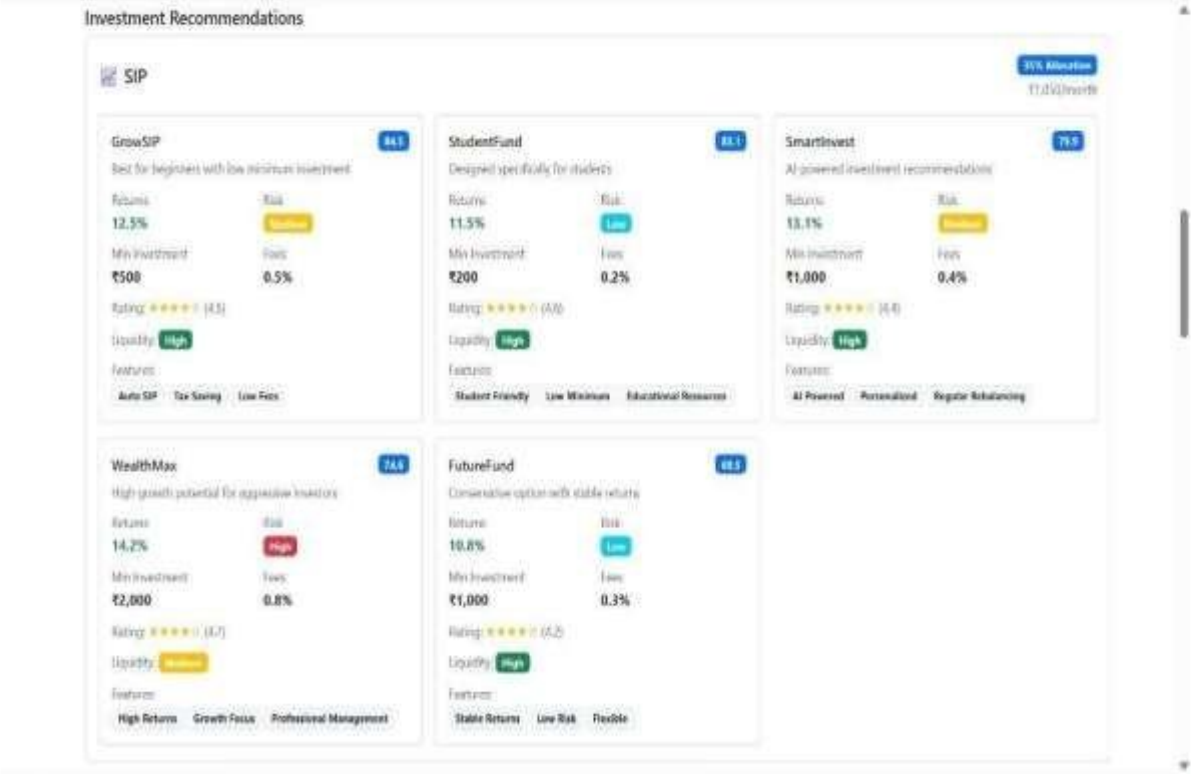
SIP 35%
Monthly ₹1,050 Projected ₹14,175
Returns: ₹1,575

MutualFunds 25%
Monthly ₹750 Projected ₹10,368
Returns: ₹1,368

Gold 20%
Monthly ₹600 Projected ₹7,740
Returns: ₹540

Crypto 15%
Monthly ₹450 Projected ₹6,750
Returns: ₹1,350

FixedDeposit 5%
Monthly ₹150 Projected ₹1,929.6
Returns: ₹129.6



Investment Recommendations

SIP 35% Allocation
₹1,050/month

GrowSIP 84.5
Best for beginners with low minimum investment
Returns 12.5% Risk Medium
Min Investment ₹500 Fees 0.5%
Rating: ★★★★☆ (4.5)
Liquidity: High
Features: Auto SIP Tax Saving Low Fees

StudentFund 83.1
Designed specifically for students
Returns 11.5% Risk Low
Min Investment ₹200 Fees 0.2%
Rating: ★★★★☆ (4.6)
Liquidity: High
Features: Student Friendly Low Minimum Educational Resources

SmartInvest 79.9
AI-powered investment recommendations
Returns 13.1% Risk Medium
Min Investment ₹1,000 Fees 0.4%
Rating: ★★★★☆ (4.4)
Liquidity: High
Features: AI Powered Personalized Regular Rebalancing

WealthMax 74.6
High growth potential for aggressive investors
Returns 14.2% Risk High
Min Investment ₹2,000 Fees 0.8%
Rating: ★★★★☆ (4.7)
Liquidity: Medium
Features: High Returns Growth Focus Professional Management

FutureFund 68.5
Conservative option with stable returns
Returns 10.8% Risk Low
Min Investment ₹1,000 Fees 0.3%
Rating: ★★★★☆ (4.2)
Liquidity: High
Features: Stable Returns Low Risk Flexible

5.4 Dashboard and Visualization Module

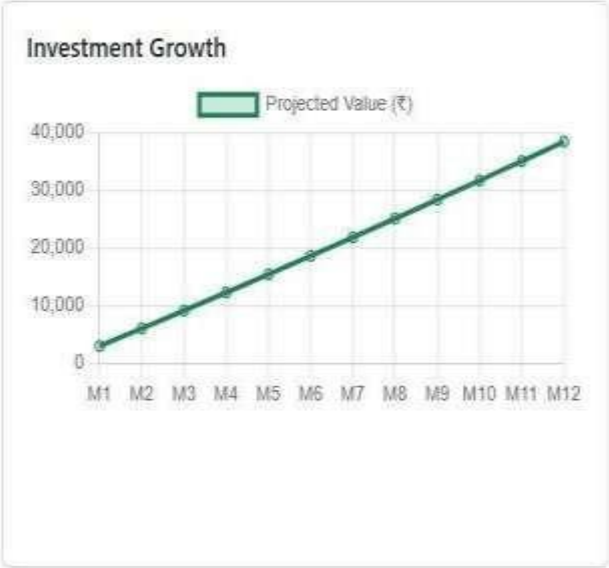
Purpose:

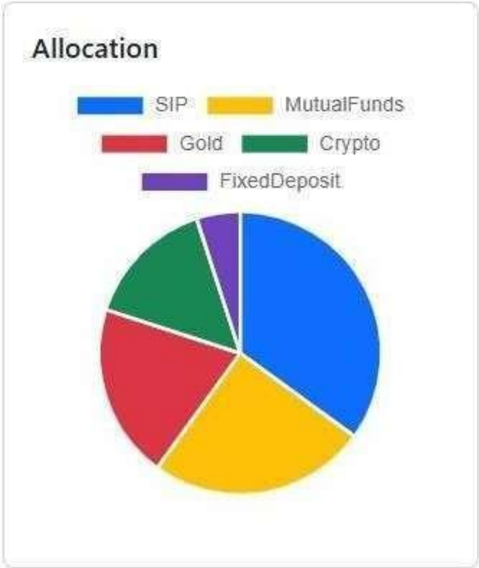
To provide a central platform for users to view investment breakdowns, progress, and actionable insights.

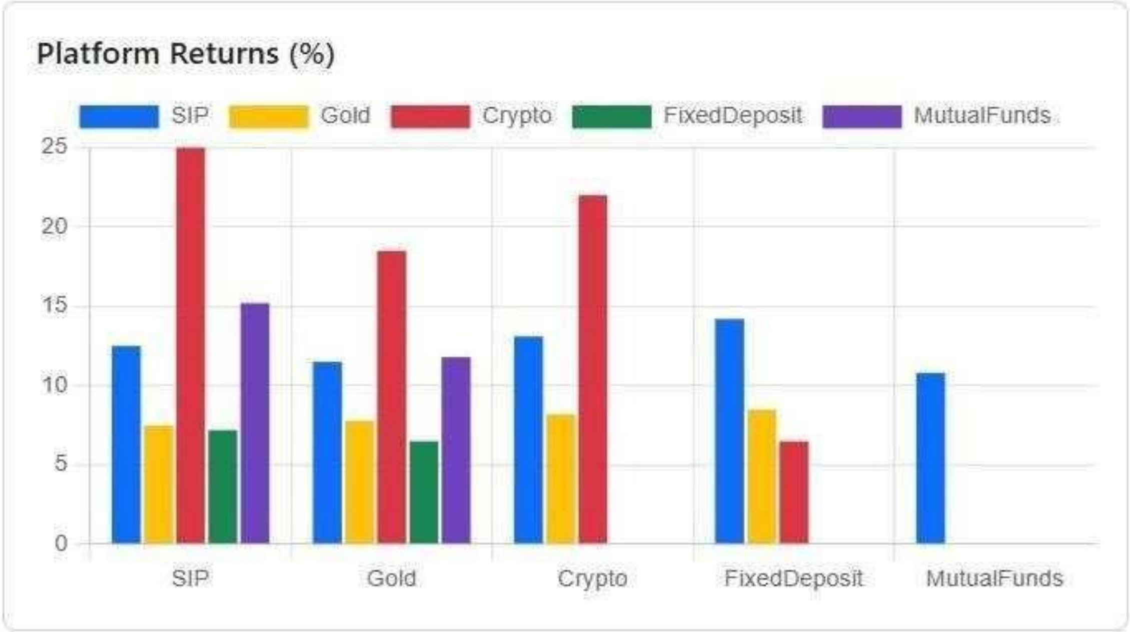
Implementation:

The dashboard serves as the main workspace after login. It integrates multiple components such as RecommendationChart, ProgressTracker, and MarketOverview to provide real-time updates. The Recharts library is used to display charts and graphs that visually represent savings growth, investment ratios, and performance summaries

React's state management enables automatic updates whenever user data changes, ensuring a responsive and interactive experience. The dashboard focuses on providing clarity through visuals and easy-to- understand summaries, motivating students to maintain consistent saving behavior.







5.5 Notification and Financial Tips Module:

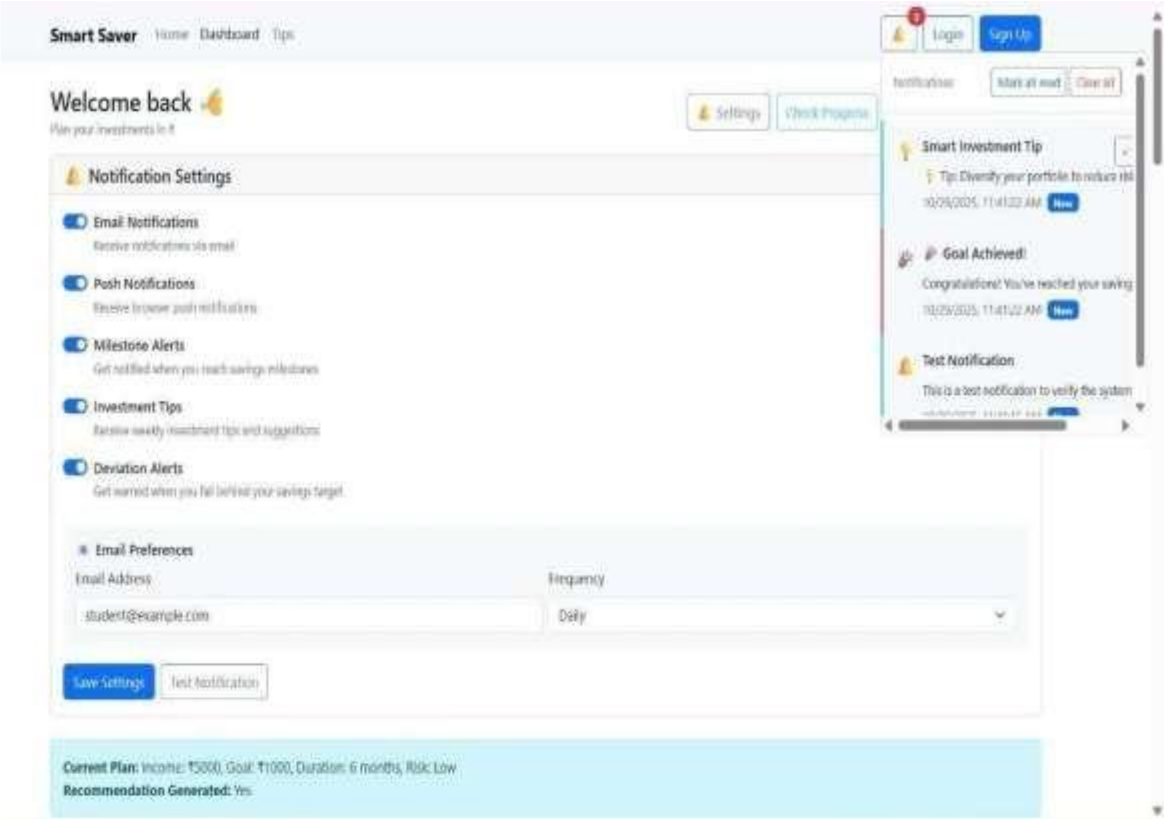
Purpose:

To enhance user engagement and promote better saving and investment practices through timely notifications.

Implementation:

This module provides instant alerts and investment tips based on user actions and input patterns. Components like NotificationBell display relevant financial messages, reminders, and suggestions. React's Context API ensures that notifications are available globally across components without redundant data passing.

This feature keeps users informed, encourages consistency in savings, and promotes financial awareness through small, actionable insights.



Welcome back
Plan your investments in it

Notification Settings

- Email Notifications: Receive notifications via email
- Push Notifications: Receive browser push notifications
- Milestone Alerts: Get notified when you reach savings milestones
- Investment Tips: Receive weekly investment tips and suggestions
- Deviation Alerts: Get warned when you fall behind your savings target

Email Preferences
Email Address: student@example.com
Frequency: Daily

Current Plan: Income: ₹5000, Goal: ₹1000, Duration: 6 months, Risk: Low
Recommendation Generated: Yes

Smart Investment Tip
Tip: Diversify your portfolio to reduce risk
10/29/2025, 11:41:22 AM New

Goal Achieved!
Congratulations! You've reached your saving
10/29/2025, 11:41:22 AM New

Test Notification
This is a test notification to verify the system

5.6 SOFTWARE DESCRIPTION:

1. VS CODE

Visual Studio Code (VS Code) is the primary Integrated Development Environment (IDE) used for the development of the Micro-Investment Recommendation System. It is a lightweight, open-source editor developed by Microsoft that supports modern web technologies such as React.js, JavaScript (ES6), HTML5, and CSS3.

VS Code offers powerful features like IntelliSense, syntax highlighting, code auto-completion, and error detection, which help improve coding efficiency and reduce development time. The built-in terminal and debugging tools allow developers to test logic, inspect variable states, and troubleshoot user interface behaviors in real time.

Since the project primarily focuses on a frontend architecture, VS Code provides an efficient environment to manage different React components, stylesheets, and assets within a single workspace. The Live Server extension is used to preview frontend changes instantly, ensuring faster iteration during the design and testing phases.

The IDE's Git integration also facilitates version control, allowing developers to commit, push, and track code changes seamlessly. This helps maintain project stability and ensures that multiple team members can collaborate efficiently without overwriting one another's work. Extensions like Prettier and ESLint help maintain clean, consistent, and well-formatted code throughout the project.

In addition, VS Code's integrated support for Node.js tools and React Developer Tools enables testing of logic flow, component rendering, and data handling in one environment. Its flexibility, customization options, and lightweight performance make it an ideal choice for developing scalable and visually appealing web applications like the Micro-Investment Recommendation System.

2. FRONTEND:

The frontend of the Micro-Investment Recommendation System is developed using React.js, a powerful JavaScript library for building dynamic and interactive user interfaces. The frontend serves as the main interaction point for users—primarily students—allowing them to input financial details, view investment recommendations, and track their progress visually.

React's component-based architecture ensures modularity, where each feature (such as login, input forms, recommendations, and charts) is built as a reusable unit. This makes the application easier to maintain, scale, and debug. Components are dynamically rendered, enabling the interface to update instantly when a user modifies their investment preferences such as risk level or savings goals.

The system's input module collects essential data including monthly income, saving targets, investment duration, and risk appetite. These inputs are processed through a rule-based recommendation logic that calculates suitable micro-investment plans like Systematic Investment Plans (SIPs), Digital Gold, and Cryptocurrency allocations.

The frontend also includes a visualization dashboard, implemented using Recharts, that converts data into interactive bar and pie charts. This helps users easily understand how their savings and investments are distributed across different financial categories. Color-coded segments and intuitive tooltips enhance readability and user engagement.

To ensure accessibility, the UI is designed with responsive layouts using CSS3 and Tailwind CSS, making it adaptable to various devices including desktops, tablets, and smartphones. Smooth transitions, hover effects, and dynamic form validation enhance the overall user experience.

3. RECOMMENDATION ENGINE MODULE:

The Recommendation Engine Module is the core functional component of the Micro- Investment Recommendation System, responsible for analyzing user inputs and generating personalized investment suggestions. It is implemented entirely within the frontend layer using React.js, ensuring a fast and interactive experience without requiring a backend or database.

This module processes data collected from users—such as monthly income, savings goals, investment duration, and risk level—and applies a rule-based decision logic to categorize users into predefined financial profiles (Low Risk, Medium Risk, High Risk). Based on these profiles, the system calculates and recommends optimal investment allocations across various micro- investment options such as:

- Systematic Investment Plans (SIPs): For low-risk and long-term investors.
- Digital Gold: For moderate risk-takers who prefer stability with flexibility.
- Cryptocurrencies: For high-risk investors seeking higher returns.

The algorithm dynamically adjusts investment proportions (e.g., Low Risk: 70% SIP, 20% Gold, 10% Crypto) to ensure that each recommendation aligns with the student's financial preferences. This logic is coded directly within React components using JavaScript functions and state management hooks (useState, useEffect) to provide instant output updates whenever the input data changes.

Additionally, the Recommendation Engine integrates closely with the Visualization Module to render the calculated allocations graphically, helping users clearly understand their investment distribution. The goal of this module is to simplify decision-making by converting raw financial data into structured, actionable insights that encourage consistent saving behavior and smart financial planning

4. VISUALIZATION MODULE:

The Visualization Module enhances the user experience by transforming numerical investment data into interactive and easy-to-understand visual representations. This module is built using the Recharts library, which integrates seamlessly with React.js for real-time rendering of charts and graphs.

Once the Recommendation Engine computes the investment distribution, the Visualization Module displays the results through bar charts and pie charts. Each segment visually represents a specific investment category (SIP, Digital Gold, or Cryptocurrency) with distinct colors and percentage labels. These visuals help users quickly interpret where their money is allocated and how it aligns with their goals and risk preferences.

The module also supports progress tracking by showing how close the user is to achieving their savings target over time. Through dynamic chart updates, students can visualize the impact of adjusting their savings goals or investment durations instantly, fostering engagement and financial awareness.

Furthermore, responsive design principles ensure that the charts automatically resize and adapt to different screen sizes and devices, maintaining readability and usability on both desktop and mobile platforms. By presenting complex financial data in a simple, intuitive manner, this module plays a crucial role in improving decision-making and promoting long-term investment habits.

5.7

5.8 CODE IMPLEMENTATION

Step 1: set up home page

```
import React from 'react';
import { Link } from 'react-router-dom';
import QuoteCarousel from '../components/QuoteCarousel.jsx'; import FeatureCard from
'../components/FeatureCard.jsx'; export default function Home() {
```

```
return (
<div>
<section className="py-5 bg-light border-bottom">
<div className="container text-center">
<h1 className="display-5 fw-bold">Smart Saver – Empowering Students to Invest Wisely</h1>
<p className="lead text-muted mt-3">Start small, grow smart with AI-backed insights.</p>
<Link className="btn btn-primary btn-lg mt-3" to="/login">Get Started</Link>
</div>
</section>
<section className="container py-4">
<QuoteCarousel />
</section>
<section className="container py-5">
<div className="row g-3">
<div className="col-md-4"><FeatureCard title="AI Recommendations" description="Personalized allocation across
SIP, Gold, Crypto."
/></div>
<div className="col-md-4"><FeatureCard title="Live Market Data" description="Track gold, crypto, and SIP trends."
icon="" /></div>
<div className="col-md-4"><FeatureCard title="Smart Tracking" description="Visualize your growth in ₹."
icon="" /></div>
</section>
</div>
<div className="mb-3">
         <label className="form-label">Password</label>
         <input name="password" type="password" className="form-control" value={form.
password} onChange={handleChange} required />
       </div>
       <button className="btn btn-primary w-100" type="submit">Login</button>
     </form>
     <p className="mt-3 text-muted">No account? <Link to="/register">Register</Link></p>
   </div>
}
```

Step 2: Set up login page

```
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom'; import { loginUser } from '../api/api.js';
export default function Login() { const navigate = useNavigate();
const [form, setForm] = useState({ email: '', password: '' }); const [error, setError] = useState('');
function handleChange(e) {  const { name, value } = e.target;
setForm((prev) => ({ ...prev, [name]: value }));
}
async function handleSubmit(e) { e.preventDefault();
setError('');
if (!form.email || !form.password) { setError('Please enter email and password.'); return;
}
try {
await loginUser(form); navigate('/dashboard');
} catch (err) { setError('Login failed.');
}
}
return (
<div className="container py-5" style={{ maxWidth: 480 }}>
<h2 className="mb-4">Login</h2>
{error && <div className="alert alert-danger">{error}</div>}
<form onSubmit={handleSubmit}>
<div className="mb-3">
<label className="form-label">Email</label>
<input name="email" type="email" className="form-control" value={form.email} onChange={handleChange}
required />
</div>
<div className="mb-3">
<label className="form-label">Password</label>
<input name="password" type="password" className="form-control" value={form.password}
onChange={handleChange} required />
</div>
<button className="btn btn-primary w-100" type="submit">Login</button>}
```

Step 3: Set up the register page

```
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom'; import { registerUser } from '../api/api.js';
export default function Register() { const navigate = useNavigate();
const [form, setForm] = useState({ email: ", password: ", confirm: " }); const [error, setError] = useState(");
function handleChange(e) {  const { name, value } = e.target;
setForm((prev) => ({ ...prev, [name]: value }));
}
async function handleSubmit(e) { e.preventDefault();
setError(");
if (!form.email || !form.password || !form.confirm) { setError('All fields are required.');
return;
}
if (form.password !== form.confirm) { setError('Passwords do not match.'); return;
}
try {
await registerUser({ email: form.email, password: form.password }); navigate('/login');
} catch (err) { setError('Registration failed.');
}
}
return (
<div className="container py-5" style={{ maxWidth: 480 }}>
<h2 className="mb-4">Register</h2>
{error && <div className="alert alert-danger">{error}</div>}
<form onSubmit={handleSubmit}>
<div className="mb-3">
<label className="form-label">Email</label>
<input name="email" type="email" className="form-control" >
)}
```

Step 4: Set up the dashboard page

```
import React, { useMemo, useState, useEffect } from 'react';
import { useNotifications } from '../contexts/NotificationContext.jsx';
import { generateRecommendation } from '../utils/recommendationEngine.js'; import ModalForm from
'../components/ModalForm.jsx';
import InvestmentTypeCard from '../components/InvestmentTypeCard.jsx'; import RecommendationChart from
'../components/RecommendationChart.jsx'; import ProgressChart from '../components/ProgressChart.jsx';
import MarketDataCard from '../components/MarketDataCard.jsx'; import NotificationSettings from
'../components/NotificationSettings.jsx';
import RecommendationSummary from '../components/RecommendationSummary.jsx'; export default function
Dashboard() {
const [plan, setPlan] = useState({ income: 15000,
goal: 50000,
duration: 12, risk: 'Medium'
}); // Default plan for testing
const [showSettings, setShowSettings] = useState(false);
const { checkMilestones, addNotification } = useNotifications(); const recommendation = useMemo(() => {
if (!plan) return null; try {
const userData = {
income: parseInt(plan.income), goal: parseInt(plan.goal), duration: parseInt(plan.duration), risk: plan.risk
};
{showSettings && (
    <div className="mb-4">
      <NotificationSettings />
    </div>
  )}
console.log('Generating recommendation for:', userData); const result = generateRecommendation(userData);
console.log('Recommendation result:', result);
return result;
} catch (error) {
console.error('Error generating recommendation:', error); return null;
}
}, [plan]);

const growth = useMemo(() => { if (!recommendation) {
// Default growth curve const months = 12; const values = [];
let v = 0;
for (let i = 0; i < months; i++) { v += 1000 + i * 150;
values.push(v);
}
return values;
}
// Generate growth curve based on recommendation
const { monthlyInvestments, projectedReturns } = recommendation;
const totalMonthly = Object.values(monthlyInvestments).reduce((sum, amount) => sum + amount, 0);
const months = plan.duration; const values = [];
let cumulative = 0;
for (let i = 0; i < months; i++) { cumulative += totalMonthly;
// Add projected growth
const growthRate = 0.01; // 1% monthly growth cumulative *= (1 + growthRate); values.push(cumulative);
}
return values;
}, [recommendation, plan]);
// Mock user data for milestone checking const mockUserData = useMemo(() => ({
income: plan?.income ? parseInt(plan.income) : 15000, goal: plan?.goal ? parseInt(plan.goal) : 50000,  duration:
plan?.duration ? parseInt(plan.duration) : 12, currentSavings: 15000,
investmentValue: 8000,
}), [plan]);

// Manual milestone checking (only when user clicks a button) const handleCheckMilestones = () => {
checkMilestones(mockUserData);
};
```

5.9 RESULT:

The Micro-Investment Recommendation System successfully demonstrates a functional and interactive web-based financial advisory tool designed to assist users—particularly students and young investors—in making informed micro-investment decisions. The system was developed using React.js, TypeScript, and visualization libraries such as Recharts, ensuring a responsive, modular, and user-friendly interface. The application fulfills its core objectives by providing personalized investment recommendations based on user preferences and risk appetite, along with clear visual insights for better understanding of financial distribution.

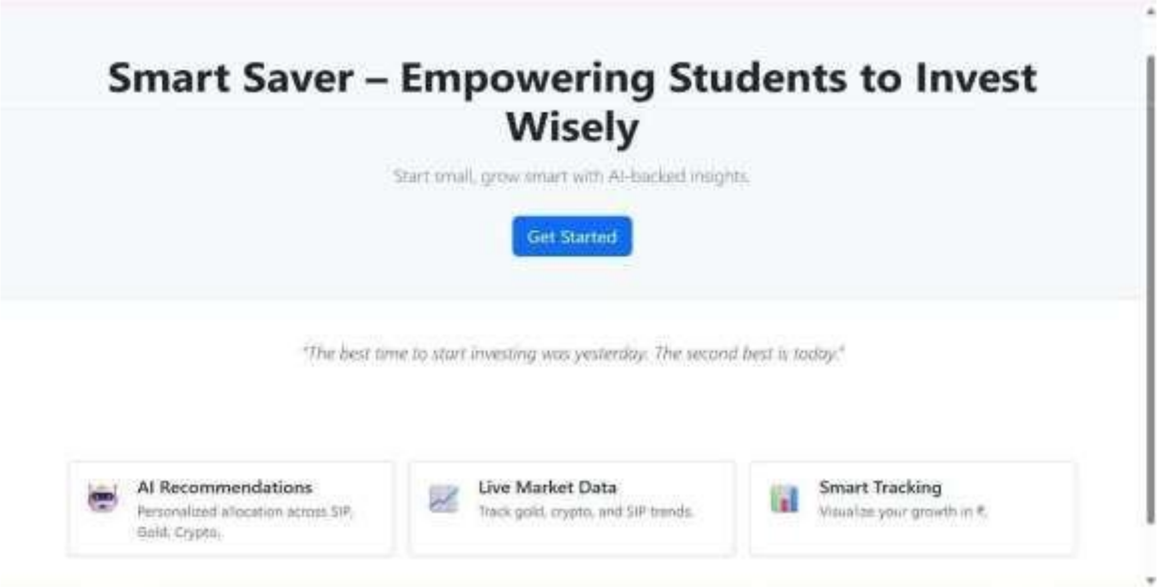
Functional Results:

- Input and Analysis Module: Users can enter their monthly income, savings goal, investment duration, and risk preference. The system instantly analyzes this data using a built-in rule-based logic to categorize users into low, medium, or high-risk investor profiles.

- Recommendation Engine: The application dynamically generates personalized investment plans across options like Systematic Investment Plans (SIPs), Digital Gold, and Cryptocurrency. These allocations are automatically adjusted based on the selected risk level and investment duration.

- Visualization Module: The calculated recommendations are displayed using bar charts and pie charts, allowing users to visualize their financial plan clearly. The color-coded charts make it easier to compare different investment portions and understand the balance between risk and return.

- Responsive Design: The system adapts seamlessly across devices—desktops, tablets, and smartphones—ensuring accessibility and usability for all types of users.



Smart Saver – Empowering Students to Invest Wisely

Start small, grow smart with AI-backed insights.

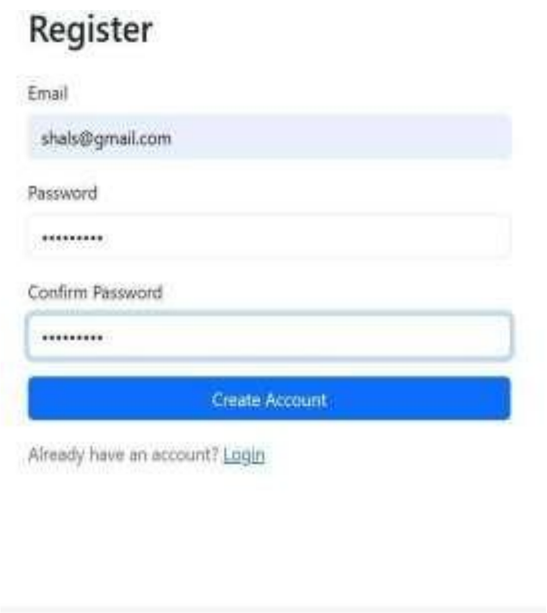
"The best time to start investing was yesterday. The second best is today."

AI Recommendations
Personalized allocation across SIP, Gold, Crypto.

Live Market Data
Track gold, crypto, and SIP trends.

Smart Tracking
Visualize your growth in ₹.

a.dashboard



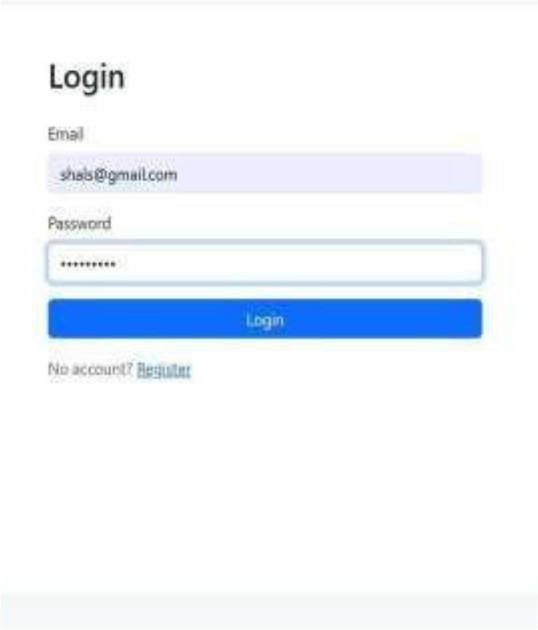
Register

Email: shals@gmail.com

Password: ••••••••••

Confirm Password: ••••••••••

Already have an account? Login



Login

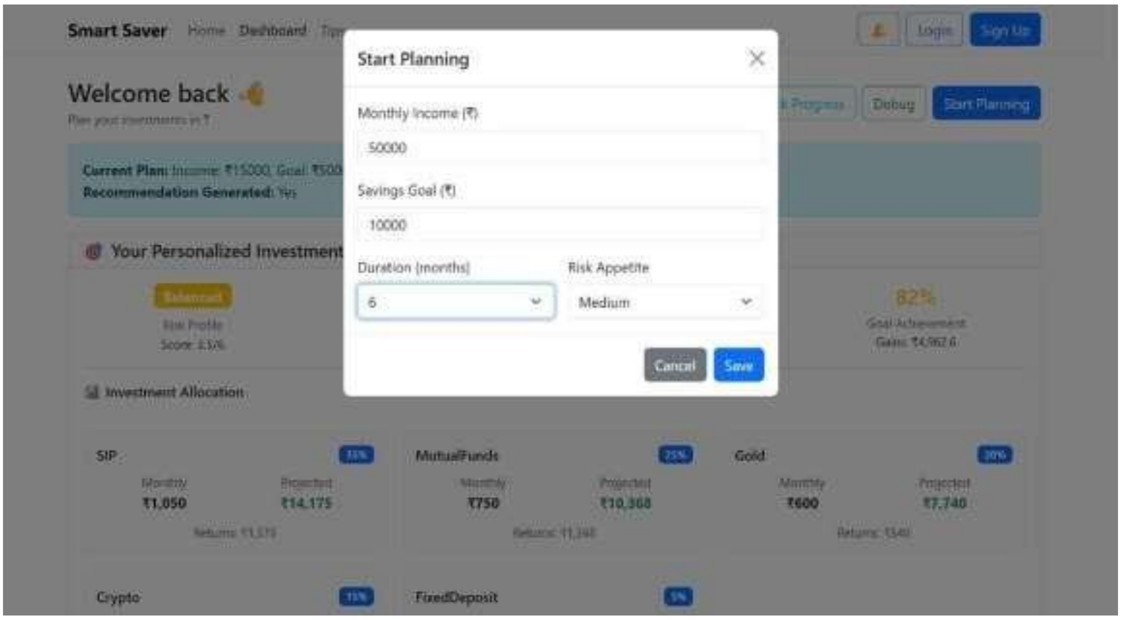
Email: shals@gmail.com

Password: ••••••••••

No account? Register

b.register page

c.login page



Start Planning

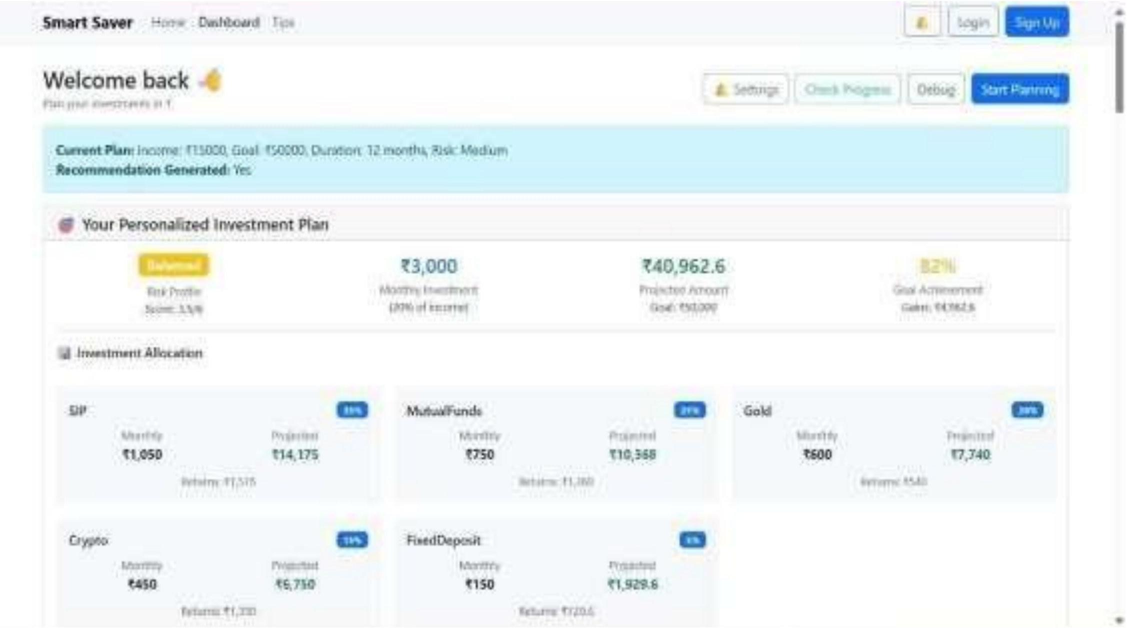
Monthly Income (₹): 50000

Savings Goal (₹): 10000

Duration (months): 6

Risk Appetite: Medium

d.input data



Current Plan: Income: ₹15000, Goal: ₹50000, Duration: 12 months, Risk: Medium
Recommendation Generated: Yes

Your Personalized Investment Plan

Balanced	₹3,000	₹40,962.6	82%
Risk Profile Score: 3.5/6	Monthly Investment (20% of income)	Projected Amount Goal: ₹50,000	Goal Achievement Gain: ₹4,962.6

Investment Allocation

		Monthly	Projected	Returns
SIP	35%	₹1,050	₹14,175	₹1,575
MutualFunds	25%	₹750	₹10,368	₹1,368
Gold	20%	₹600	₹7,740	₹540
Crypto	15%	₹450	₹6,750	₹1,350
FixedDeposit	5%	₹150	₹1,929.6	₹129.6

e.recommendation engine

Performance and Usability Results

- The website offers real-time interactivity, instantly updating recommendations and charts as users modify their financial inputs.

- A clean and intuitive interface ensures that users with little or no financial knowledge can easily navigate and interpret investment results.

- Client-side validation ensures accurate data entry and prevents incorrect or incomplete submissions.

- The application demonstrates excellent performance and responsiveness, with quick load times and smooth transitions between modules.

- The visualization of results enhances user engagement by providing immediate feedback on how income, duration, or risk preference changes impact investment outcomes.

- Overall, the system achieves its objectives by simplifying investment planning, promoting financial awareness, and encouraging consistent saving habits among users.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

The Micro-Investment Recommendation System successfully achieves its goal of helping users, especially beginners and students, understand and plan small-scale investments in a simplified and interactive way. By combining data-driven logic with a visually appealing interface, the system empowers users to make informed financial decisions based on their income, risk capacity, and investment goals.

Developed using React.js and TypeScript, the system ensures a responsive and dynamic user experience, where recommendations and charts are generated instantly based on user input. The modular design enhances maintainability and scalability, allowing future extensions and updates with minimal effort. Overall, the project demonstrates how technology can bridge the gap between complex financial planning and everyday users, promoting early financial literacy and responsible saving behavior.

6.2 FUTURE SCOPE:

While the current version of the Micro-Investment Recommendation System provides a strong foundation for basic financial guidance, there are several opportunities for enhancement and expansion in the future:

Integration of Real-Time Market Data:

Incorporating live market APIs can help update investment recommendations dynamically based on current trends, interest rates, and stock performances.

User Authentication and Data Storage:

Adding secure login functionality will allow users to save their investment history, track growth over time, and compare performance metrics.

AI-Based Recommendation Engine:

Implementing machine learning algorithms can improve personalization by learning from user behavior and historical investment outcomes.

Mobile Application Development:

Extending the web app into a native mobile application would increase accessibility and convenience for on-the-go users.

Expense and Goal Tracking:

Future versions can include budget analysis and goal-tracking modules to give users a more comprehensive view of their financial health.

Multilingual Support and Accessibility:

Adding multiple language options and voice assistance can make the platform more inclusive for users from diverse backgrounds.

APPENDICES

SOURCE CODE

Feature Card.jsx

```
import React from 'react';
export default function FeatureCard({ title, description, icon }) { return (
<div className="card h-100 shadow-sm">
<div className="card-body d-flex align-items-start gap-3">
<div className="fs-2" aria-hidden>{icon}</div>
<div>
<h5 className="card-title mb-1">{title}</h5>
<p className="card-text text-muted mb-0">{description}</p>
</div>
</div>
</div>
);
}
```

Navbar.jsx

```
import React from 'react';
import { Link, NavLink } from 'react-router-dom'; import NotificationBell from './NotificationBell.jsx';

export default function Navbar() { return (
<nav className="navbar navbar-expand-lg navbar-light bg-light border-bottom">
<div className="container">
<Link className="navbar-brand fw-bold" to="/">Smart Saver</Link>
<button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs- target="#navContent" aria-controls="navContent" aria-expanded="false" aria-label="Toggle navigation">
<span className="navbar-toggler-icon"></span>
</button>
<div className="collapse navbar-collapse" id="navContent">
<ul className="navbar-nav me-auto mb-2 mb-lg-0">
<li className="nav-item"><NavLink className="nav-link" to="/">Home</NavLink></li>
<li className="nav-item"><NavLink className="nav-link" to="/dashboard">Dashboard</NavLink></li>
<li className="nav-item"><NavLink className="nav-link" to="/tips">Tips</NavLink></li>
</ul>
<div className="d-flex gap-2 align-items-center">
<NotificationBell />
<Link className="btn btn-outline-primary" to="/login">Login</Link>
<Link className="btn btn-primary" to="/register">Sign Up</Link>
</div>
</div>
</div>
</nav>
);
}
```

ProgressChart.jsx

```
import React from 'react';
import { Line } from 'react-chartjs-2'; import {
Chart as ChartJS, CategoryScale, LinearScale, PointElement, LineElement, Tooltip, Legend,
} from 'chart.js';

ChartJS.register(CategoryScale, LinearScale, PointElement, LineElement, Tooltip, Legend); export default function

ProgressChart({ dataPoints = [] }) {
const labels = dataPoints.map((_, idx) => M${idx + 1}); const data = { labels, datasets: [
{
label: 'Projected Value (₹)', data: dataPoints, borderColor: '#198754', backgroundColor: 'rgba(25,135,84,.2)', tension:
0.3,
fill: true,
},
],
};
const options = { responsive: true, plugins: { legend: { display: true } } }; return (
<div className="card h-100">
<div className="card-body">
<h6 className="card-title">Investment Growth</h6>
<Line data={data} options={options} />
</div>
</div>
);
}
```

NotificationSettings.jsx

```
import React, { useState } from 'react';
import { useNotifications, NOTIFICATION_TYPES } from '../contexts/NotificationContext.jsx';

export default function NotificationSettings() {
const { settings, updateSettings, addNotification } = useNotifications(); const [localSettings, setLocalSettings] =
useState(settings);

const handleSettingChange = (key, value) => {
const newSettings = { ...localSettings, [key]: value }; setLocalSettings(newSettings); updateSettings(newSettings);
};

const settingOptions = [
{
```

```
key: 'emailNotifications', label: 'Email Notifications', description: 'Receive notifications via email'
},
{
key: 'pushNotifications', label: 'Push Notifications', description: 'Receive browser push notifications'
},
{
key: 'milestoneAlerts', label: 'Milestone Alerts',
description: 'Get notified when you reach savings milestones'
},
{
key: 'tipAlerts',
label: 'Investment Tips',
description: 'Receive weekly investment tips and suggestions'
},
{
key: 'deviationAlerts', label: 'Deviation Alerts',
description: 'Get warned when you fall behind your savings target'
}
];

return (
<div className="card">
<div className="card-header">
<h5 className="mb-0">ú     Notification Settings</h5>
</div>
<div className="card-body">
{settingOptions.map(option => (
<div key={option.key} className="form-check form-switch mb-3">
<input
className="form-check-input" type="checkbox" id={option.key} checked={localSettings[option.key]}
onChange={(e) => handleSettingChange(option.key, e.target.checked)}
/>
<label className="form-check-label" htmlFor={option.key}>
<div className="fw-semibold">{option.label}</div>
<small className="text-muted">{option.description}</small>
</label>
</div>
))}

<div className="mt-4 p-3 bg-light rounded">
<h6 className="mb-2">📧Email Preferences</h6>
<div className="row g-2">
<div className="col-md-6">
<label className="form-label">Email Address</label>
<input type="email"
className="form-control" placeholder="your@email.com" defaultValue="student@example.com"
/>
</div>
<div className="col-md-6">
```

```
<label className="form-label">Frequency</label>
<select className="form-select">
<option>Daily</option>
<option>Weekly</option>
<option>Monthly</option>
<option>Only important alerts</option>
</select>
</div>
</div>
</div>

<div className="mt-3">
<button className="btn btn-primary">Save Settings</button>
<button
className="btn btn-outline-secondary ms-2" onClick={() => { addNotification({
type: NOTIFICATION_TYPES.INFO,
title: 'Test Notification',
message: 'This is a test notification to verify the system is working.', priority: 'low'
});
}}
>
Test Notification
</button>
</div>
</div>
</div>
);
}
```

ModalForm.jsx

```
import React, { useState } from 'react';

export default function ModalForm({ onSubmit }) {
const [form, setForm] = useState({ income: '', goal: '', duration: '12', risk: 'Medium' });

function handleChange(e) {  const { name, value } = e.target; setForm((prev) => ({ ...prev, [name]: value }));
}
function handleSubmit(e) { e.preventDefault(); onSubmit?.(form); const modalEl =
document.getElementById('planModal');
const modal = window.bootstrap?.Modal.getInstance(modalEl) || new window.bootstrap.Modal(modalEl); modal.hide();
}

return (
<div className="modal fade" id="planModal" tabIndex="-1" aria-hidden="true">
<div className="modal-dialog">
<div className="modal-content">
<div className="modal-header">
<h5 className="modal-title">Start Planning</h5>
<button type="button" className="btn-close" data-bs-dismiss="modal" aria- label="Close"></button>
</div>
<form onSubmit={handleSubmit}>
<div className="modal-body">
<div className="mb-3">
<label className="form-label">Monthly Income (₹)</label>
<input name="income" type="number" className="form-control" value={form.income} onChange={handleChange}
required />
</div>
<div className="mb-3">
<label className="form-label">Savings Goal (₹)</label>
<input name="goal" type="number" className="form-control" value={form.goal} onChange={handleChange}
required />
</div>
<div className="row g-3">
<div className="col-md-6">
<label className="form-label">Duration (months)</label>
<select name="duration" className="form-select" value={form.duration} onChange={handleChange}>
{[6, 12, 24, 36, 60].map((m) => (
<option key={m} value={m}>{m}</option>
))}
</select>
</div>
<div className="col-md-6">
<label className="form-label">Risk Appetite</label>
<select name="risk" className="form-select" value={form.risk} onChange={handleChange}>
{['Low', 'Medium', 'High'].map((r) => (
<option key={r} value={r}>{r}</option>
))}
</select>
</div>
</div>
</div>
<div className="modal-footer">
<button type="button" className="btn btn-secondary" data-bs- dismiss="modal">Cancel</button>
<button type="submit" className="btn btn-primary">Save</button>
</div>
</form>
</div>
</div>
</div>
);
}
```

RecommendationCard.jsx

```
import React from 'react';

export default function RecommendationCard({ title, percentage, description }) { return (
<div className="card h-100 border-success">
<div className="card-body">
<div className="d-flex justify-content-between align-items-center">
<h6 className="card-title mb-0">{title}</h6>
<span className="badge bg-success">{percentage}%</span>
</div>
<p className="card-text mt-2 text-muted mb-0">{description}</p>
</div>
</div>
);
}
```

App.jsx

```
import React from 'react';
import { Routes, Route } from 'react-router-dom';
import { NotificationProvider } from './contexts/NotificationContext.jsx'; import Navbar from './components/Navbar.jsx';
import Footer from './components/Footer.jsx'; import Home from './pages/Home.jsx';  import Login from
'./pages/Login.jsx';
import Register from './pages/Register.jsx';
import Dashboard from './pages/Dashboard.jsx'; import Tips from './pages/Tips.jsx';

export default function App() { return (
<NotificationProvider>
<div className="d-flex flex-column min-vh-100">
<Navbar />
<main className="flex-fill">
<Routes>
<Route path="/" element={<Home />} />
<Route path="/login" element={<Login />} />
<Route path="/register" element={<Register />} />
<Route path="/dashboard" element={<Dashboard />} />
<Route path="/tips" element={<Tips />} />
</Routes>
</main>
<Footer />
</div>
</NotificationProvider>
);
}
```

index.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Smart Saver</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
</head>
```

```
<body>
<div id="root"></div>
<script type="module" src="/src/index.jsx"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
</body>
</html>
```

REFERENCE

[1]. Panagiotis Moutsiopoulos, Greece: Insights and Investment Opportunities in the Business-for-Sale Market (August 21, 2025), 2025. Micro M&A in Greece: Insights and Investment Opportunities in the Business-for-Sale Market.

[2]. Oluwatobiloba Alao(2025). Data driven financial decision-making for minority enterprises: Capital access, investment strategies, and creditworthiness optimization.

[3]. Pratik Maheshwari, Annals of Operations Research 350 (2), 395-423, 2025.Internet of things for perishable inventory management systems: an application and managerial insights for micro, small and medium enterprises

[4]. Titilope Tosin Adewale.(1-15, 2023). Big data-driven financial analysis: A new paradigm for strategic insights and decision-making.

[5]. Bohm, M., & Foote, B. (2022). Enhancing customer experience through automated service systems. Journal of service research, 24(3), 345-361.

[6]. Li, X., & Xu, L. D. (2018). Data-driven decision making for financial systems. Journal of computing and information science in engineering, 18(2), 021003.

[7]. Kai Chang, Qiong Wan(2020). Green fiscal policy and firms’ investment efficiency: New insights into firm-level panel data from the renewable energy industry in China.

[8]. Alessio Emanuele Biondo(2014). Micro and macro benefits of random investments in financial markets.