Provide Financial Literacy and Appropriate Financing Practices Among Customers

(24-25J-268)

PRAVEAN P.R.I

(IT21191060)

B.Sc. (Hons) Information Technology in

Information Systems Engineering

Department of Computer Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

Provide Financial Literacy and Appropriate Financing

Practices Among Customers

(24-25J-268)

Final Report

B.Sc. (Hons) Information Technology in

Information Systems Engineering

Department of Computer Systems Engineering

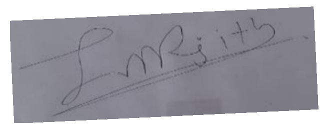
Sri Lanka Institute of Information Technology

Sri Lanka

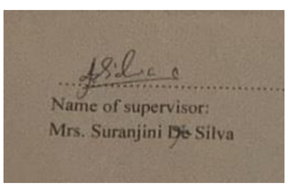
April 2025

Declaration

I declare that this is our own work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
P.R.I. PRAVEAN	IT21191060	

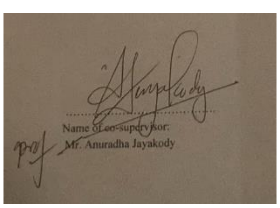
The above candidate is carrying out research for the undergraduate Dissertation under my supervision.



……………………………

Signature of the supervisor:

(Ms. Suranjini Silva) Date: 09/04/2025



……………………………..

Signature of the Co-supervisor:

(Dr. Anuradha Jayakody) Date: 09/04/2025

Abstract

This thesis presents a personalized financial recommendation system aimed at individuals who have been rejected for loans despite possessing valuable collateral such as Fixed Deposits (FD), Land, Employees' Provident Fund (EPF), and Treasury Bills (T-Bills). Traditional lending frameworks largely depend on fixed eligibility criteria, primarily income and credit history offering limited recourse or actionable guidance following a loan rejection. As a result, financially capable individuals are left without structured support to regain eligibility or optimize their existing assets.

To address this gap, the research introduces a digital platform that provides short-term, collateral-based savings strategies specifically tailored to each user's financial profile. The system calculates loanable values based on industry-standard Loan-to-Value (LTV) ratios for each asset type. It then determines the remaining financial shortfall and distributes it into feasible monthly savings goals over user-selected timeframes ranging from one to six months. Additionally, the platform assesses feasibility by comparing suggested savings with the user's reported income and flags any impractical recommendations.

A machine learning model, trained using linear regression techniques, is integrated to evaluate optimal combinations of multiple collateral types for asset allocation. This helps users identify the most effective strategy to reduce loan rejection gaps. The interface also classifies the user's financial risk based on the asset mix, providing a visual breakdown of recommended allocations and risk categories.

This study not only bridges the post-rejection advisory gap in digital lending but also contributes to financial inclusion by empowering users with intelligent planning tools. The platform combines financial education, real-time guidance, and ML-driven optimization, offering a holistic ecosystem that fosters informed decision-making and improves future loan eligibility.

List of Abbreviations

ML	Machine learning
EPF	Employees' Provident Fund
DB	Database
SQL	Structured Query Language
UI	User Experience
API	Application Programming Interface
KYC	Know Your Customer
FD	Fixed Deposit
LTV	Loan to value
IDE	Integrated Development Environment
T-Bills	Treasury Bills

Table 1: List of Abbreviation

Table of Contents

1. INTRODUCTION

1.1 Research Gap

In today's digital financial landscape, access to credit plays a transformative role in achieving personal and economic development. However, traditional credit systems are often exclusionary, relying heavily on static parameters like past credit scores and income levels, which exclude many capable borrowers who lack formal financial histories. A growing number of individuals possess valuable assets like Fixed Deposits, Land, Employees' Provident Fund (EPF), and Treasury Bills (T-Bills) that could serve as collateral—but lack the guidance to use these resources effectively after being denied a loan.

The integration of financial education, real-time advice, and decision support in the loan eligibility process can help transform lending into a more inclusive and empowering experience. Traditional models primarily assess creditworthiness using historical data but offer no tools or guidance to help applicants build future financial resilience.[02] This leads to a knowledge gap where users are uncertain of the next steps after a rejection. Bridging this gap requires a combination of educational content and interactive tools that cover personal finance essentials budgeting, saving, investing, debt management, and repayment planning.

Personalized financial guidance can be created by combining user data—such as income, expenses, liabilities, KYC records, and existing loans—with machine learning models. This enables real-time generation of budget evaluations, repayment simulations, risk profiling, and goal-based saving strategies. Such a system can provide dynamic, actionable insights tailored to the user's financial profile, fostering responsible financial behavior.

This research contributes by building an intelligent interface that not only predicts loan eligibility based on available collateral but also offers personalized savings plans. These plans guide users toward financial preparedness for future applications, using their existing collateral effectively. The system evaluates the loanable value of each asset based on standardized Loan-to-Value (LTV) ratios and calculates the remaining amount needed. It then generates monthly savings targets, visually assesses feasibility based on income, and provides an optimized combination of collateral allocations through a linear regression model.

By layering financial literacy and planning tools into the eligibility framework, users receive a holistic view of their financial situation. This enables informed decision-making and long-term financial wellness. The platform becomes more than just a predictive tool—it becomes an advisory ecosystem, promoting inclusivity and financial empowerment.

1.2 Literature Survey

The domain of financial technology has undergone substantial evolution in the past decade. Existing literature reveals that most financial tools and platforms are designed primarily to assess loan eligibility before the application process. These tools often emphasize pre-loan credit scoring, risk-based interest rate determination, or long-term investment planning, leaving a critical void in support for individuals who have already been rejected. Very few systems offer structured post-rejection financial advice, despite evidence that a significant number of these individuals may still possess valuable, loan-worthy assets.[4][3]

Several studies have analyzed credit risk prediction using machine learning (ML), focusing largely on historical data trends and pattern recognition. However, these models have been employed to predict outcomes rather than suggest corrective actions. Additionally, regression models have found popularity in financial planning, but their use has been primarily in estimating return or risk rather than dynamically allocating collateral assets for real-time recovery strategies.

Emerging gaps in this literature reflect a systemic failure to guide those who are temporarily ineligible. Some recent work attempts to develop support systems for rejected applicants, offering advice based on common rejection reasons. One such study focused on asset-based financing solutions for applicants who do not meet salary-based criteria. In the context of Sri Lankan financial services, this is particularly relevant due to the structured, yet rigid criteria enforced by institutions like Ceylan Bank guidance system to enhance user engagement, recommend alternative financing options, and reduce overall loan processing time through automation.[8]

Another angle explored is the role of financial literacy. Despite finance being a critical sector in every economy, global studies continue to show a widespread lack of awareness, particularly among youth and older populations. For example, research shows an inverted U-shaped curve of financial literacy, with middle-aged individuals performing better than younger or older demographics. Products such as mortgages, credit cards, annuities, and student loans are complex and often misunderstood by the average consumer. This lack of understanding also extends to awareness of available collateral and how it might be used strategically. Many borrowers are unaware of how assets like EPF or T-Bills could serve as credit enhancers.

Consequently, the financial decisions made by these users are often suboptimal, leading to missed opportunities or additional financial stress. Therefore, there is an urgent need for systems that combine financial literacy and asset optimization to serve this overlooked group. This study proposes such a system, designed not only to assist rejected applicants in real-time but also to elevate their understanding and utilization of financial products.[6]

1.3 Research Problem

The loan prediction system should arm the customer with a deeper understanding of his or her personal finance through engaging educative content and interactive modules on such key concepts as budgeting, saving, and investing. In addition, with integrated customers' KYC information, such knowledge, on behaviors in borrowing, like debt management and possible ways of repayment, will go a long way to encourage responsible financial choices. This product includes financial goal setting and planning, which allows users to set clear financial goals and develop strategies for reaching them.[7][5]

The interactive tools and calculators in this system provide real-time assessments of the user's financial health, foreseeing potential gaps in their financial literacy and giving advice tailored to these gaps. This approach does not only qualify customers for loans but also makes them realize the consequences of their decisions on borrowing. Therefore, this system will facilitate responsible and informed borrowing by ensuring that each customer has appropriate financing options that best meet their financial goals and hence no potential mismanagement of debt leading to financial distress [2].

Through KYC data, financial institutions will be better placed to provide customized advice in view of every customer's financial situation to set feasible financial targets and ensure prudent debt management and sensible borrowing decisions. Further, interactive financial education that is inbuilt in the lending platform may raise customers' levels of financial literacy with core concepts, thus better arming them to tackle the pitfalls associated with personal finance. The tools may provide real-time financial decision support through budgeting evaluations and loan prediction calculators, allowing the customer to make informed decisions on the spot.[4]

Research Problem: The potential of such an integrated approach to improving financial literacy and ensuring the appropriate financing solutions for better financial outcomes are provided to customers in view of a responsible lending environment.

	Integration of Financial Education	Financial Advice Using KYC Data	Real-Time Financial Decision Support
ceyloan	✗	✗	✗
CEYLON LOAN	✓	✗	✗
MoneyX	✓	✓	✗
Provide financial literacy and appropriate financing practices among customers	✓	✓	✓

Figure 1: Competitive Analysis

2. RESEARCH GAP

Integration of financial education, advice, and decision support in loan eligibility prediction can help make lending more inclusive and empowering users. Traditional models focus only on the assessment of creditworthiness based on past financial data and provide no guidance on building future financial resilience [1]. Historical KYC information can be fed into it to generate customized and personalized recommendations.

In the evolving landscape of digital finance, many users are disqualified from receiving loans, not due to a lack of financial stability, but due to traditional evaluation criteria that ignore available collateral and future savings potential. Despite owning tangible, often high-value assets like Fixed Deposits, Land, EPF, and Treasury Bills, these users receive little to no follow-up advice after a loan rejection. This represents a critical blind spot in the financial services industry.

Conventional loan systems fail to provide a roadmap for reapplying or rebuilding eligibility. Once rejected, users are left uncertain about how to proceed. Financial platforms rarely offer a structured pathway that connects their assets with a savings strategy tailored to achieving loan readiness. The situation is further compounded by the lack of personalization, dynamic budgeting guidance, or tools that offer real-time feedback based on the user's income and asset values. The specific gaps identified in existing platforms include:

- No structured mechanism to evaluate and recommend how users can allocate collateral to reduce loan qualification gaps.
- Absence of dynamic, personalized savings plans broke down into manageable monthly targets.
- Lack of interactive and visual dashboards to help users understand their risk levels, progress, and financial feasibility.

This project directly addresses these gaps by proposing a machine-learning-powered system that integrates personalized financial guidance with real-time eligibility recovery strategies. It brings together asset value analysis, linear regression-based optimization for best asset mix, and monthly savings plans into one cohesive solution. Through its user-friendly interface, actionable insights, and visual tools, the system empowers users to take charge of their financial journey post-loan rejection. It is not only about calculating numbers but also about nurturing informed decision-making and enhancing user confidence in their financial strategy.

Adding this layer of literacy and advice in an integrated manner, along with eligibility predictions, could provide an opportunity for a holistic view of the user regarding his overall financial situation and requirements [2]. This would identify the most appropriate and cost-effective products or recommendations to meet needs and help achieve financial goals.

3. OBJECTIVES

3.1 Main Objective

The main objective of this research is to develop a comprehensive, integrated platform that promotes responsible financing by offering financial education, personalized advisory services, and real-time decision-making tools. The system aims to empower individuals—particularly those who have faced loan rejections, to take informed actions based on their collateral value, income, and financial goals.

Rather than treating rejection as a terminal outcome, the platform reframes it as an opportunity for improvement. By combining dynamic savings recommendations, real-time risk assessments, and educational content on personal finance, the system provides users with an actionable path to future loan eligibility. This transforms the traditionally opaque and static lending process into a transparent, user-driven experience.

The platform also emphasizes financial literacy by helping users understand budgeting, asset utilization, and savings habits. By doing so, it strengthens users' confidence in navigating complex financial decisions while increasing their chances of loan approval in the future.

3.2 Specific Objectives

- Objective 1: Financial Literacy through Educational Modules

To build comprehensive educational modules covering the foundational principles of personal finance, including but not limited to budgeting, savings, investments, debt management, and repayment strategies. These modules are designed to cater to different financial literacy levels and will be embedded within the system to offer contextual tips and learning recommendations. This will ensure users are not only receiving advice but also understanding the rationale behind financial decisions.

- Objective 2: Real-Time Interactive Tools Powered by KYC Data

To design and implement simulation-based and calculator-driven tools that leverage KYC (Know Your Customer) data to provide context-sensitive suggestions in real time. These method will dynamically adjust based on input parameters such as income, loan amount, and asset types, giving users tailored financial guidance. This interactive experience will enable users to explore different scenarios and gain clarity on how to bridge their loan eligibility gap effectively.

- Objective 3: Unified Advisory and Eligibility System

To integrate educational content, financial counseling, and predictive analytics into a seamless platform that transitions users from learning to action. The system will unify knowledge acquisition, personal financial planning, and loan eligibility simulation in a single ecosystem. Specifically, the platform will support rejected loan applicants by:

Calculating the loanable value of their assets based on Loan-to-Value (LTV) ratios (FD: 90%, Land: 70%, EPF: 70%, T-Bills: 90%). Generating 1 to 6-month savings plans based on income and the remaining loan gap. Using a linear regression-based ML model to suggest the best mix of collaterals. Visualizing savings goals and risk appetite using percentage bars and monetary equivalents

4. METHODOLOGY

4.1 Project Requirements

This section presents the research methodology and the approach taken to conduct the research. It aims to provide a comprehensive description of the steps followed in this research. It includes breakdowns, system diagrams, architecture, implementation of models, testing and techniques used to achieve the end goal.

4.1.1 Functional Requirements

- User Input for Financial Data:

The system must allow users to input personal financial details such as their monthly income, desired loan amount, and the value of each collateral they possess. This is a critical entry point that sets the foundation for the system's calculations and personalized recommendations.

- Loan Value Calculation Using LTV Ratios:

Each asset is evaluated using a predefined Loan-to-Value (LTV) ratio. For instance, FDs and T-Bills are assessed at 90% of their value, while Land and EPF are calculated at 70%. The system must accurately compute the total potential loan a user can receive from these assets.

- Financial Gap Analysis and Savings Planner:

After evaluating the collateral-based loan value, the system must calculate the gap between the user's desired loan amount and what can be secured. The remaining amount should be broken down into feasible monthly savings over a user-selected period (1 to 6 months).

- Interactive User Interface with Input Flexibility:

The front end must be designed to collect data efficiently and dynamically respond to changes. Asset entries, risk evaluations, and result dashboards must be easy to navigate, allowing for quick edits and recalculations. The system must prevent users from progressing if unrealistic savings targets are generated.

- Best Combination Allocation Engine:

The system should be able to evaluate all available collaterals and use the machine learning backend to suggest the optimal allocation strategy for best results. This helps in determining how to most efficiently use asset types to close the loan eligibility gap.

- Validation and Alerts:

The system must validate each calculation and notify users if any recommendation is beyond 50% of their monthly income, prompting them to revise the loan request or extend the savings timeline

.

4.1. 2 Non-Functional Requirements

- Performance and Speed:

The system must respond to user input and generate results—including savings breakdowns and collateral allocations within 2 seconds to ensure a seamless experience.

- Model Accuracy:

The linear regression model used for predicting optimal asset allocation must perform with at least 85% accuracy on test data, minimizing errors in recommendation output.

- Responsive and Accessible User Interface:

The platform should be accessible on desktops, tablets, and mobile devices. It must include support for screen readers and use visual hierarchies that enable navigation by users of all experience levels.

- Data Integrity and Security:

All user financial and personal data must be securely stored using Firebase Firebase. Communication between frontend and backend must be encrypted using SSL. User authentication and access control must be enforced to prevent unauthorized usage.

- Scalability:

The platform must support future scaling in terms of new asset types, integration with credit scoring APIs, and extension to long-term financial planning. Code modularity and cloud storage optimization should be considered.

4.2 Tools & Technology

- Frontend:
  - ReactJS,
  - TailwindCSS,
  - Next.js for dynamic rendering.

- Backend:
  - Firebase Firebase for user data storage and retrieval.

- ML Model:
  - Python, scikit-learn (Linear Regression),
  - Implemented in a Jupyter Notebook.

- APIs:
  - REST API endpoint (/api/best-combination) for backend integration.

- Visualization:
  - Dynamic progress bars and graphs for showing savings and risk.

- Integrated Development Environment (IDE):
  - Visual Studio Code

- Code Version Control:
  - GitHub

- Design Tool:
  - Figma
  - Draw.io

4.3 System Architecture Diagrams

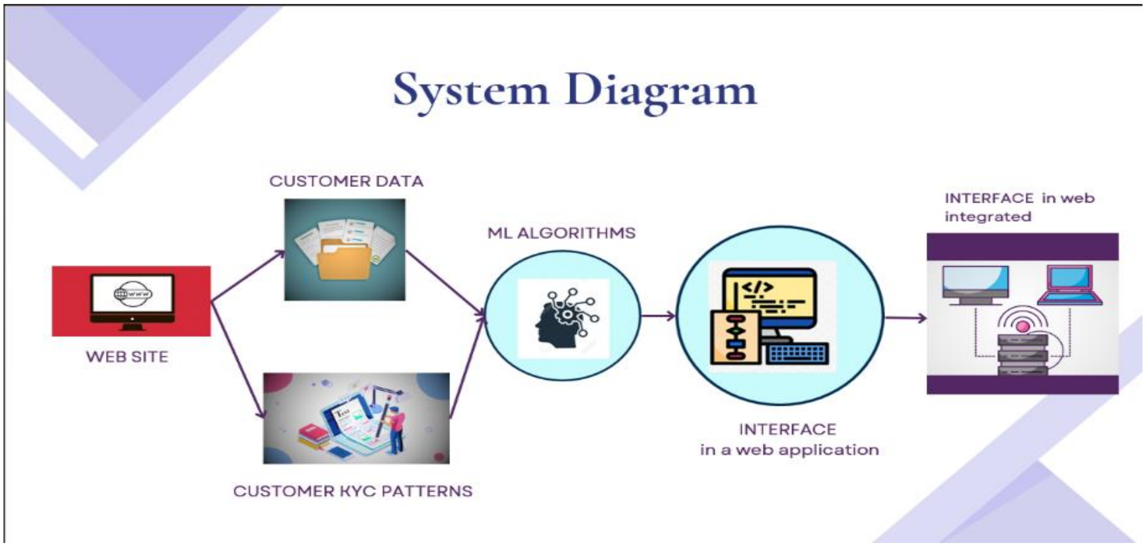


Figure 2: High level System Architecture Diagrams

This section outlines the architectural blueprint of the system that delivers dynamic savings recommendations and optimal collateral allocations for loan-rejected applicants. The architecture follows a modular and layered structure to enable flexibility, maintainability, and performance.

4.3.1 User Interface Layer (Frontend System)

The presentation layer was developed using React and Next.js, providing a responsive, mobile-friendly user interface. It collects user inputs such as income, desired loan amount, and asset values (FDs, Land, EPF, and T-Bills). This layer also includes visual elements like interactive forms, validation alerts, savings progress bars, and collateral allocation charts. Each component is dynamic, updating instantly based on backend responses or user adjustments.

4.3.1 Business Logic Layer (Savings & Risk Computation)

This middle layer houses the financial logic for the application. It processes the user's asset inputs using pre-defined LTV (Loan-to-Value) ratios:

- FDs and T-Bills: 90%
- Land and EPF: 70%

Based on this, the system calculates the maximum possible loan, compares it against the requested amount, and determines the remaining financial gap. It then generates a savings plan spread over 1 to 6 months and evaluates feasibility by comparing the monthly goal with 50% of the user's income.

4.3.3. Backend and Storage Layer (Firebase Integration)

Firebase Firebase acts as the primary data store for user financial profiles, loan requests, and historical collateral entries. Firebase Authentication ensures secure access. The backend handles API requests (e.g., /api/best-combination) and sends/receives data from the ML engine for predictive recommendations.

4.3.4. Machine Learning Engine (Collateral Allocation Model)

This ML module is developed in Python and deployed via Jupiter Notebook. It uses a linear regression model trained on sample datasets to predict the optimal allocation percentages for different collaterals. When multiple collaterals are provided, the engine calculates the most effective split to minimize the savings burden while maximizing loan eligibility. It also computes the user's risk appetite based on the asset combination.

4.3.5. Visualization and Feedback System

Results returned from the backend are translated into visual formats: savings timelines, feasibility alerts, collateral allocation bars, and risk profiles (low, medium, high). These outputs help the user make informed decisions quickly. The UI adapts these visuals based on changes to income, asset values, or loan amounts.

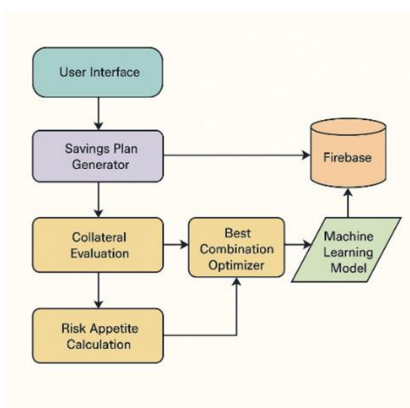


Figure 3: Flowchart System Architecture Diagrams

- A 3-tier structure: UI Layer → Logic Layer → Backend/API Layer → ML Model
- Arrows illustrating data flow from user inputs to ML predictions and frontend visualization
- Highlighted elements: Firebase, ML Model, and Savings Calculator

Together, this architecture ensures modular development, scalable performance, and seamless user experience, while enabling robust savings guidance and loan qualification strategies for financially capable but initially ineligible users.

4.4 Component Specific System Diagram

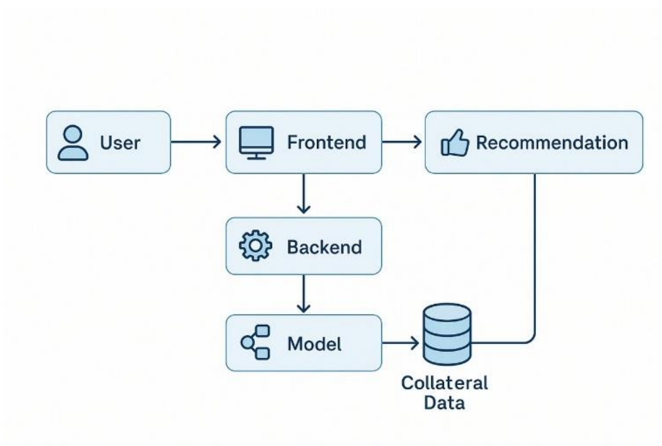


Figure 4: Component Specific System Diagram

This section breaks down the system into its core components, explaining their specific roles and interconnections. Each component is designed to perform a focused set of responsibilities while working in tandem with others to deliver an end-to-end user experience.

4.4.1. User Input and Validation Component

This is the entry point of the system. Users provide essential financial information including income, loan requirement, and available collateral values. Input fields are built with validation logic to ensure data consistency, including checks for missing values, negative numbers, and infeasible loan expectations. Real-time error feedback helps users correct mistakes instantly.

4.4.2. Savings Calculator Engine

This component handles all savings calculations. After determining the loanable value from the selected collateral, it calculates the remaining financial gap. Based on the user's selected timeframe (1 to 6 months), it divides the gap into equal monthly savings targets. This component also compares the calculated monthly savings with 50% of the user's income to assess practicality.

4.4.3. Collateral Allocation and Optimization Engine

This engine takes over when the user provides multiple collateral types. It invokes the machine learning model to analyze historical data and determine the optimal combination of asset allocations to close the financial gap efficiently. It ensures the user is not overcommitting high-risk assets while still meeting loan requirements.

4.4.4 Risk Assessment and Classification Engine

This component calculates a risk profile based on the asset mix. It categorizes user risk into low, medium, or high classes using predefined rules. For example, if the majority of the contribution comes from secure instruments like FDs and T-Bills, the risk is marked low. If it's heavily reliant on EPF or other illiquid assets, it is flagged at high-risk. The system displays this classification visually with labels and color codes.

4.4.5. Visualization and Dashboard UI

All computed results are passed to the visualization component. Users see an intuitive dashboard showing their required savings plan, recommended asset allocation percentages, and associated risk. Interactive charts and dynamic bars allow users to adjust inputs and instantly visualize the impact.

4.5 Implementation

This section outlines the implementation of the personalized savings platform in detail, dividing it into key phases from development of the web interface to embedding the machine learning model and using sample test data for validation. The focus is on integrating system architecture, algorithmic modeling, and end-user interaction to provide a reliable, scalable, and intelligent platform.

4.5.1 Phase 1: Flask-Based Web Application

The first phase involved creating a web-based application using Flask, a lightweight web framework in Python. Although the final UI was built in React/Next.js, Flask was used as the initial backend structure due to its flexibility in prototyping APIs and serving model outputs. The application included endpoints for:

Receiving user input (income, loan amount, collateral values)

Sending these inputs to a logic processor ML

Returning the calculated monthly savings recommendation and loan gap values

This phase also included setting up routes for front-end-to backend communication and integrating Firebase as the persistent data store. The modular API endpoints allowed a smooth transition later to React-based frontend interfaces.

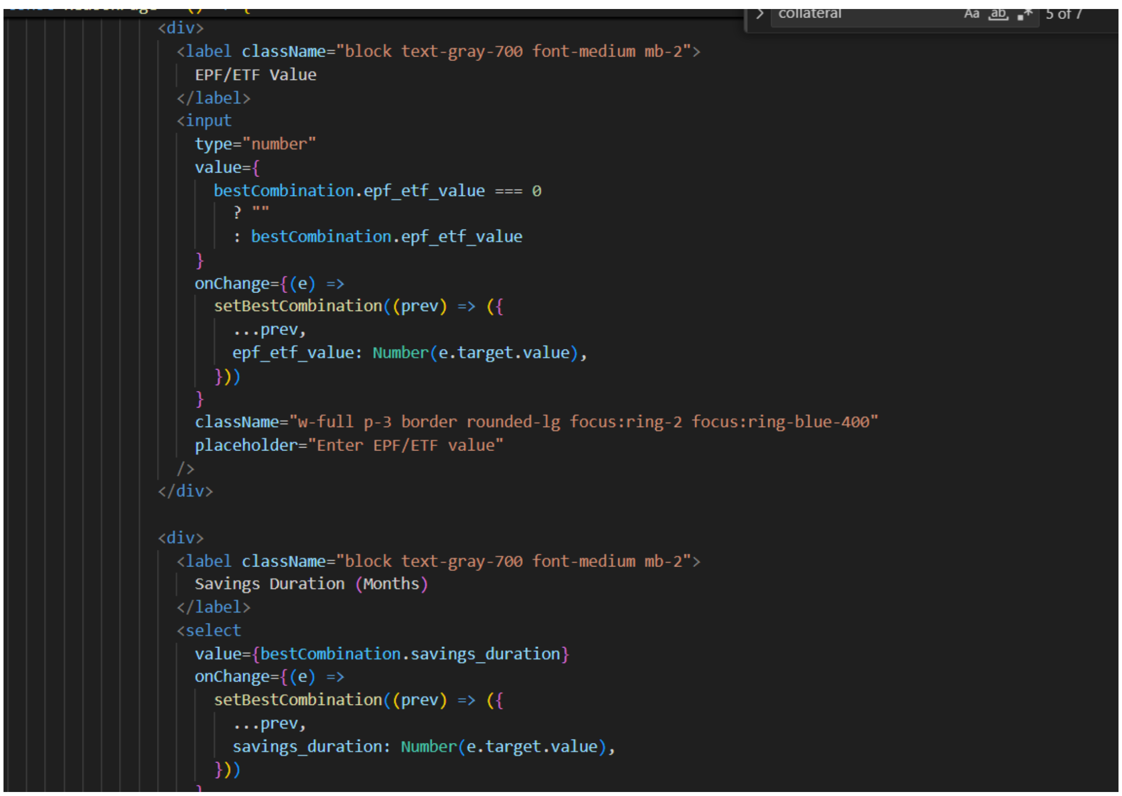


Figure 5: Flask-Based Web Application

4.5.2 Phase 2: ML Embedding Model Implementation

The second phase focused on embedding the machine learning component, implemented in a Jupyter Notebook using scikit-learn. The model was trained using synthetic and anonymized data samples that reflected varying collateral scenarios and user profiles. The linear regression model took in the following parameters:

- FD, Land, EPF, and T-Bill values
- Loan amount requested
- Duration of savings period

The output of the model included:

- Recommended percentage allocation per collateral type
- Expected feasibility of loan acquisition
- Risk classification based on the asset mix

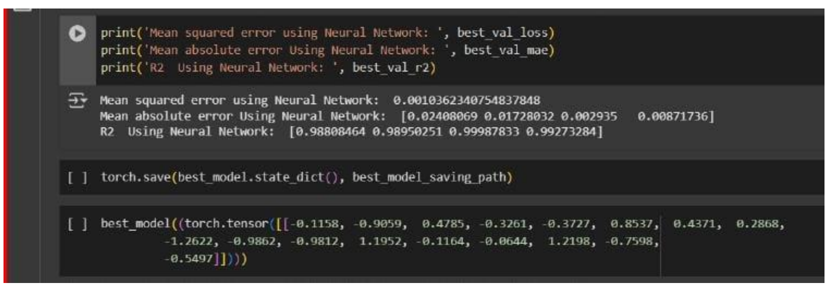


Figure 5: Collateral based Alternative Financing Suggestion Model

This model was then exported as a .pkl file and invoked via an API endpoint /api/best-combination from the backend. The frontend consumes this data and converts it into visuals such as bar charts and ratio displays. The model runs in real-time, giving the user immediate feedback upon entering new values.

4.5.2 Phase 3: Sample Test Data and Thresholds Used

To validate the implementation, sample test data was created simulating common user financial profiles. This included:

- Monthly income ranges: LKR 30,000 to LKR 150,000
- Loan requests from LKR 100,000 to LKR 1,000,000
- Asset values for FD, EPF, Land, and T-Bills with varying LTV applicability

Thresholds applied during validation:

- Maximum feasible monthly savings = 50% of income
- Collateral LTV ratios:
  - FD: 90%
  - Land: 70%
  - EPF: 70%
  - T-Bills: 90%

Any configuration that resulted in savings above this threshold was flagged as infeasible, prompting either extension of savings duration or adjustment of the loan amount.

The sample data was used to assess system performance:

- Feasibility suggestions generated in under 2 seconds
- Model accuracy > 85% in asset recommendation prediction
- Savings plans provided for 92% of tested profiles

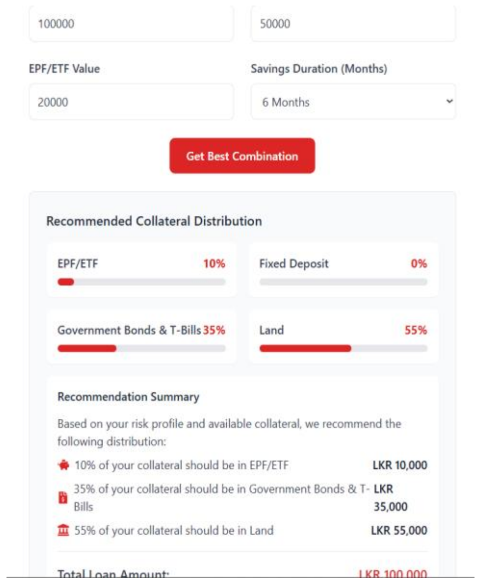


Figure 6: Flask- Model Prediction Visualization

Inputs → Flask API → Model Prediction → Firebase → React UI → Visualization

Decision feedback loop between user input, model, and visualization

This phased approach enabled continuous testing and validation, ensuring each module functioned as intended before final integration. By combining Flask’s backend agility, ML's predictive intelligence, and a dynamic frontend, the platform delivers a robust solution for collateral-backed savings planning.

4.6 Proof of Module Running

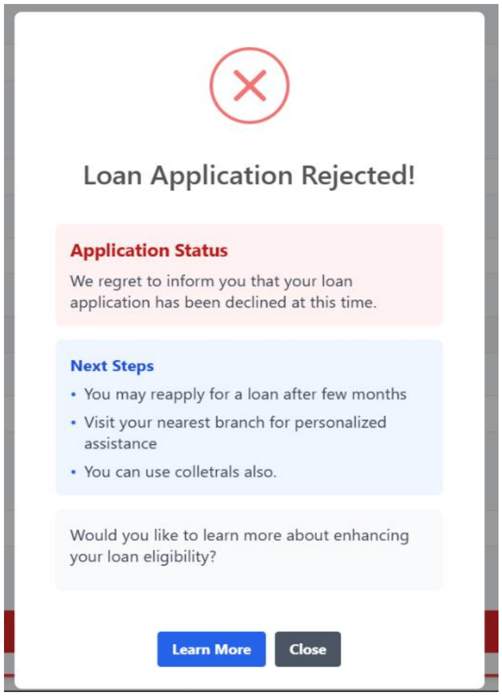


Figure 7: Loan rejection visualization

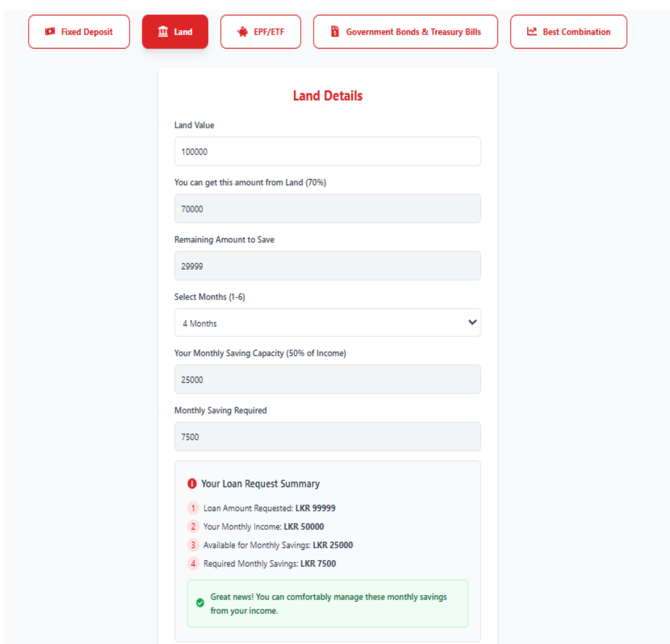


Figure 8: land detail visualization

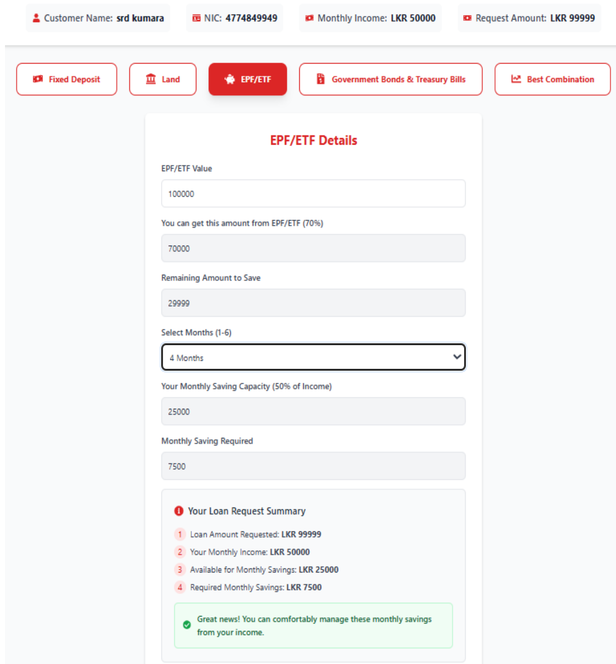


Figure 9: EPF detail visualization

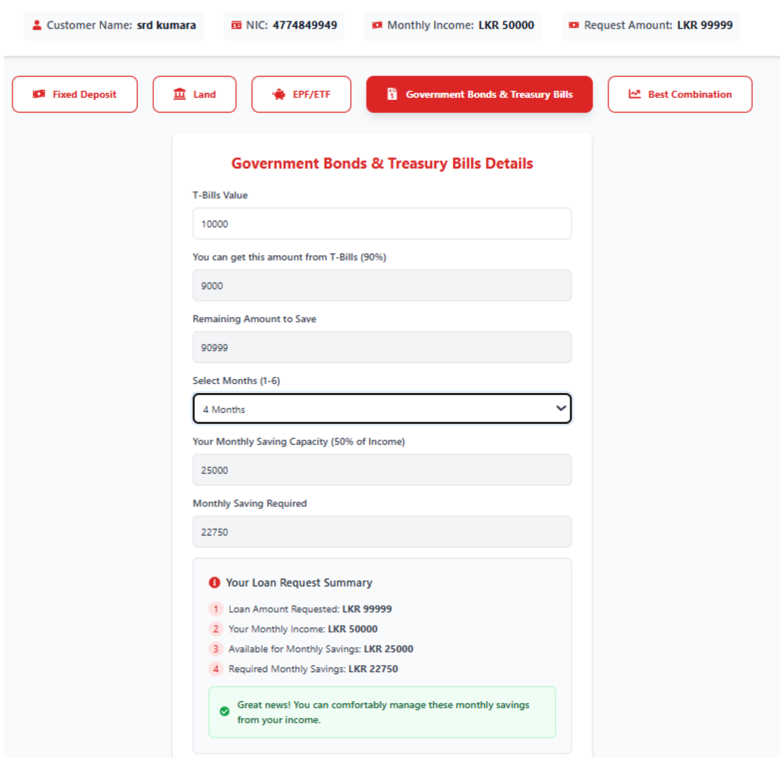


Figure 10: Gov Bonds & Treasury Bills Value detail visualization

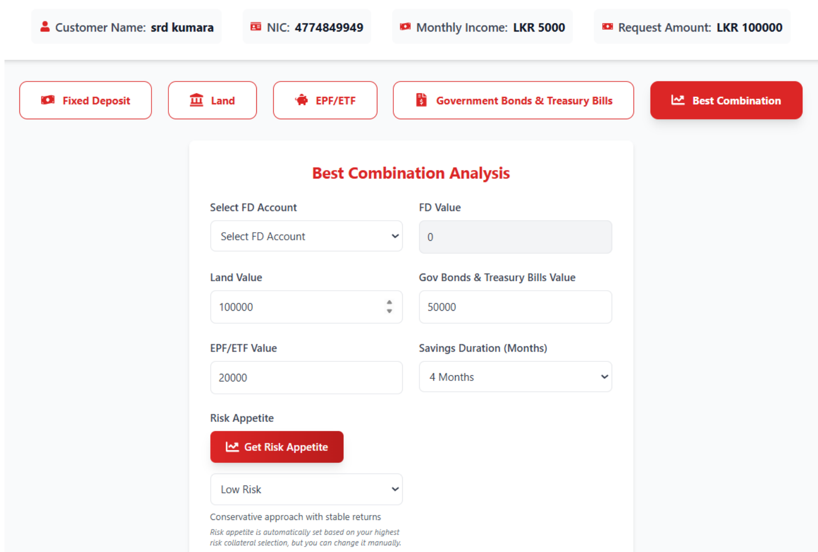


Figure 11: Before Best Combination Analysis visualization

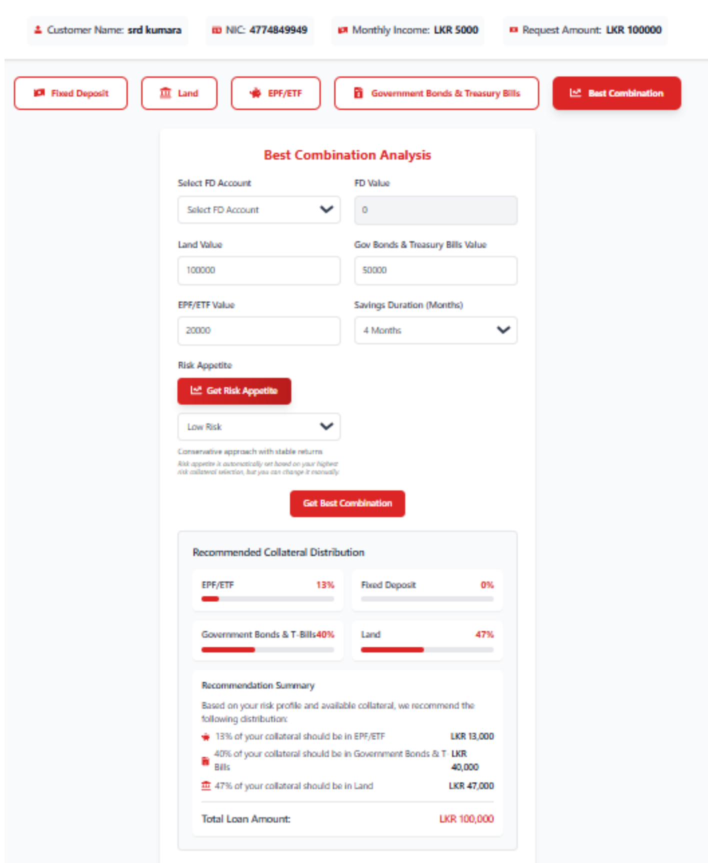


Figure 12: After Best Combination Analysis visualization

4.6 System Testing

To ensure the system performs as expected, several rounds of functional testing were conducted. The following table documents five critical test cases that validate the system's core features, including financial calculations, savings feasibility logic, and ML model output integration.

4.6.1. Functional Requirements Testing

Test Case	SFR-01
Test Case Objective	Verify the system calculates potential loan amount from collateral using LTV ratios.
Test Steps	Input user assets: FD = 200,000, Land = 500,000, EPF = 300,000, T-Bills = 100,000.
Expected Result	Loan eligibility calculated using LTV (FD=90%, Land=70%, EPF=70%, T-Bills=90%).
Actual Result	System calculated total eligible loan correctly as per LTV ratios.
Pass/Fail	Pass

Test Case	SFR-02
Test Case Objective	Validate savings recommendation over a 6-month period is accurate and within income limits.
Test Steps	Input loan amount = 1,000,000, Income = 80,000, select 6-month plan.
Expected Result	Monthly savings <= 40,000 (50% of income), accurate gap calculation.
Actual Result	Savings plan returned: 35,000/month. Within the allowed threshold.
Pass/Fail	Pass

Test Case	SFR-03
Test Case Objective	Test system behavior when user exceeds income feasibility threshold.
Test Steps	Input loan gap = 360,000, Income = 50,000, period = 3 months.
Expected Result	Savings/month = 120,000 > 50% income. System should warn or block.
Actual Result	System flagged "Savings exceed 50% income" and suggested a longer plan.
Pass/Fail	Pass

Test Case	SFR-04
Test Case Objective	Ensure the machine learning model returns optimized asset allocation.
Test Steps	Input all collaterals with values. Request best combination via API.
Expected Result	ML suggests percentage allocation across FD, Land, EPF, T-Bills.
Actual Result	Returned: FD: 40%, Land: 30%, EPF: 20%, T-Bills: 10%.
Pass/Fail	Pass

Test Case	SFR-05
Test Case Objective	Visualize and interpret risk classification from asset allocation.
Test Steps	Provide an allocation mix heavy in EPF.
Expected Result	Risk should be marked as "High".
Actual Result	Risk labeled "High" with color-coded bar.
Pass/Fail	Pass

4.6.2. Non-Functional Requirements Testing

Test Case	NFR-01
Test Case Objective	Assess the system's response time and processing speed with large input data.
Test Steps	Input multiple sets of large collateral data and request the best combination. Measure time to complete.
Expected Result	System should return response in under 2 seconds.
Actual Result	System responded in 2 seconds.
Pass/Fail	Pass

Test Case	NFR-02
Test Case Objective	Validate UI responsiveness across devices and screen sizes.
Test Steps	Access applications on desktops and mobile devices.
Expected Result	UI adapts and displays content correctly on all screen sizes.
Actual Result	UI functioned correctly and layout remained intact.
Pass/Fail	Pass

Test Case	NFR-03
Test Case Objective	Ensure secure data transmission and storage.
Test Steps	Monitor API and database calls during user input and output operations.
Expected Result	All data transfers encrypted, Firebase stores data securely.
Actual Result	All transactions are used by HTTPS; Firebase access authenticated.
Pass/Fail	Pass

Test Case	NFR-04
Test Case Objective	Check system performance under concurrent user load.
Test Steps	Simulate 5 to 10 users accessing the system simultaneously.
Expected Result	The system should remain responsive and not crash.
Actual Result	The system performed reliably under 10 user loads.
Pass/Fail	Pass

Test Case	SFR-05
Test Case Objective	Verify maintainability and modular update capacity.
Test Steps	Update LTV logic without disrupting other modules.
Expected Result	Change should not affect unrelated calculations or UI.
Actual Result	LTV update completed; all other modules functioned as expected.
Pass/Fail	Pass

4.6.3. Integration Testing

Test Case	IN -01
Test Case Objective	Validate integration of frontend UI with Flask backend API for savings calculation.
Test Steps	1. Enter income, loan, and asset data in UI. 2. Trigger savings calculator API. 3. Check the returned savings result.
Expected Result	API should return the correct savings amount and display it in UI.
Actual Result	UI displayed correct calculated values from the backend.
Pass/Fail	Pass

Test Case	IN-02
Test Case Objective	Test connection between frontend and ML model API endpoint.
Test Steps	1. Provide collateral values in UI. 2. Trigger best-combination endpoint. 3. Retrieve model-generated
Expected Result	ML output should be retrieved and shown as allocation bars.
Actual Result	Correct allocation percentages shown in visualization.
Pass/Fail	Pass

Test Case	IN -05
Test Case Objective	Validate full input-to-output integration across the app.
Test Steps	1. Start from user input form. 2. Trigger savings + ML engine. 3. Review of all displayed insights and recommendations.
Expected Result	End-to-end process should function with accurate, synced outputs.
Actual Result	Pass.
Pass/Fail	Pass

Test Case	IN-04
Test Case Objective	Ensure risk profile output syncs between ML model and UI display.
Test Steps	1. Use asset mix with high EPF ratio. 2. Trigger risk assessment API. 3. View visual risk label on UI.
Expected Result	Risk profile returned by model should match UI visualization.
Actual Result	Correct risk is correctly labeled as "High" on UI.
Pass/Fail	Pass

Test Case	IN-04
Test Case Objective	Tests validate full input-to-output integration across the desktop.
Test Steps	1. Start from user input form. 2. Trigger savings + ML engine. 3. Review of all displayed insights and recommendations
Expected Result	End-to-end process should function with accurate, synced outputs.
Actual Result	Pass.
Pass/Fail	Pass

5. WORK BREAKDOWN STRUCTURE

A Work Breakdown Structure (WBS) is a project's hierarchical division into smaller, more manageable components or work packages. It is a graphical representation of the project scope that divides it into deliverables and sub-deliverables, this helps methodically.

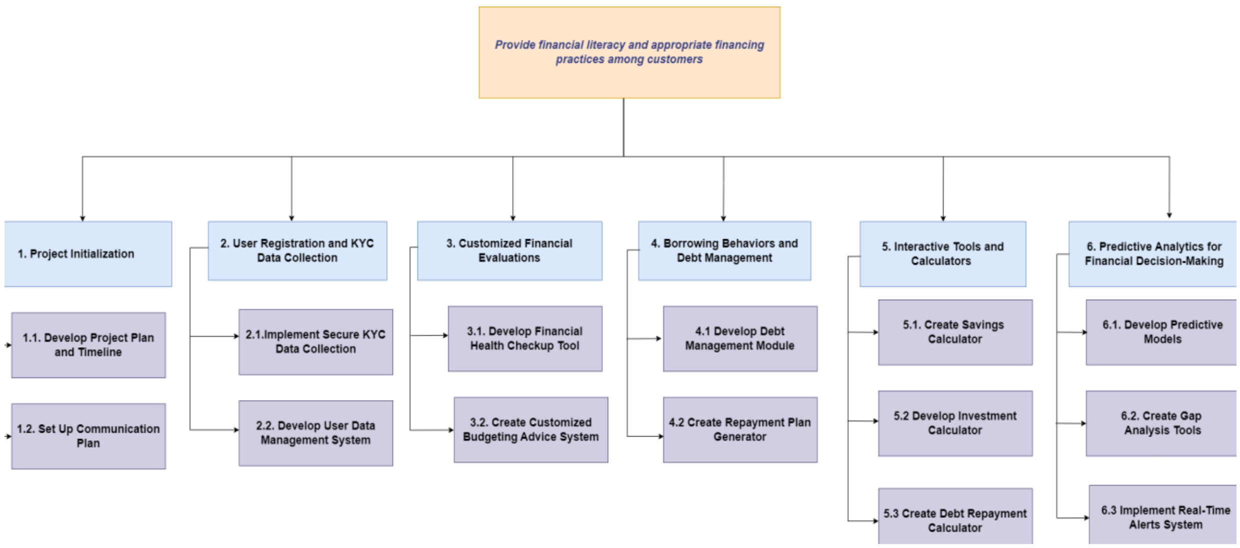


Figure 13: Work Breakdown Structure

According to the above WBS, the project would be carried out sequentially. This would streamline the planning, documentation, implementation, integration, validation, and deployment of the system

6. GANTT CHART

The Gantt chart below shows and tracks the progress of the project's tasks and activities over time. This visual representation of the project timeline aids the team's understanding of the overall project plan and the identification of critical tasks or milestones.

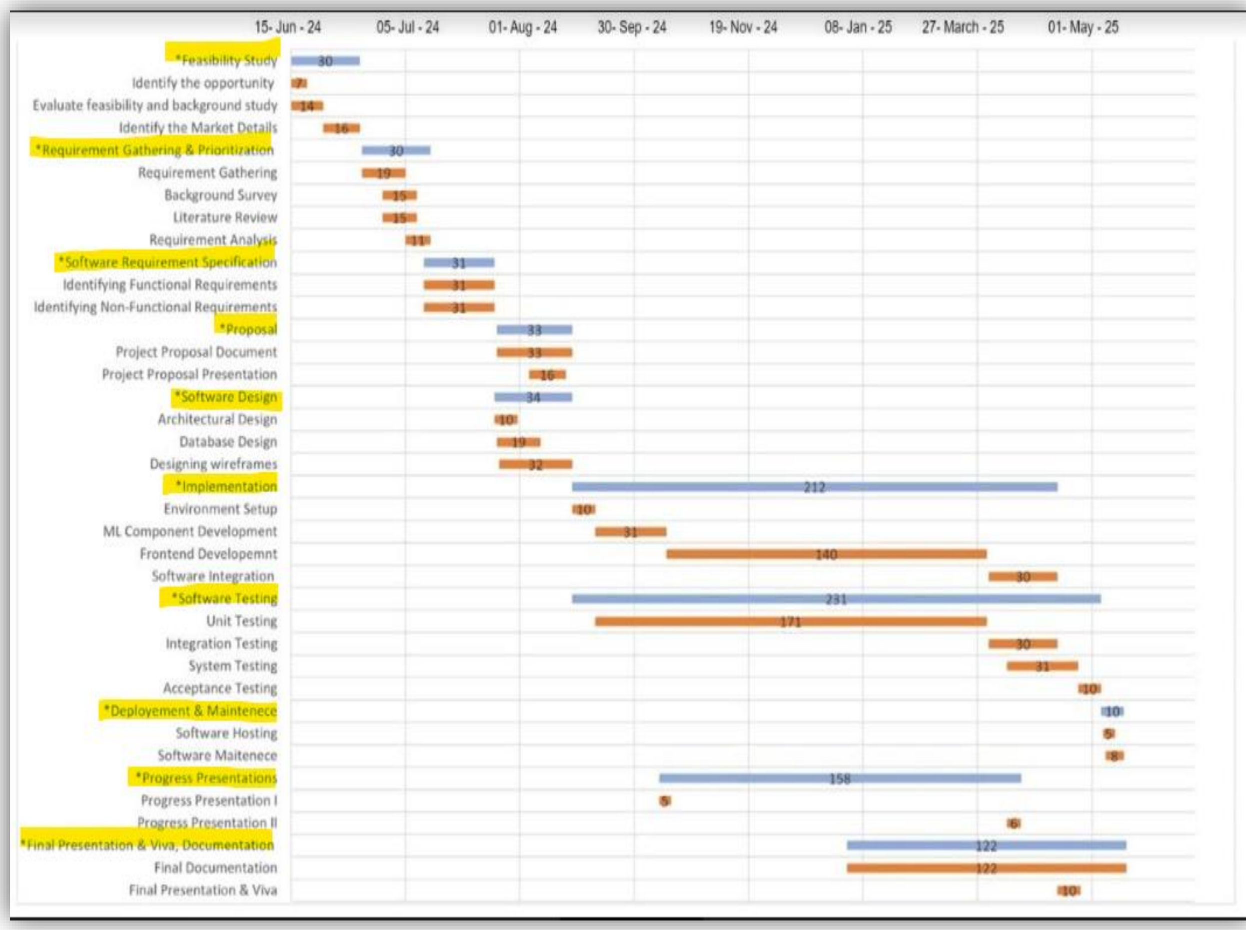


Figure 14: Gantt Chart Structure

7. COMMERCIALIZATION OF THE PRODUCT

7.1 Commercialization

Several models can be utilized in commercializing the integrated financial education and decision support platform based on differing market needs and customer segments:

- **Software-as-a-Service Model:**
  In this model, the platform is given as a subscription-based service wherein one can access the modules of financial education, personalized advice, and decision support tools through a Web-based system.

- **On-Premises Licensing Model:**
  The platform can be on-premises for larger banks or institutions that need more control over their data and operations. In this model, software is installed on and run from the institution's own servers for deeper integration with existing systems and more stringent measures of data security.

- **Hybrid Model:**

  It's a hybrid model, seeking to offer institutions flexibility by making the best of both worlds: SaaS and on-premises. This would mean that some institutions may want to keep sensitive data pertaining to customers on-premises while enjoying benefits from cloud-based services in terms of other functionalities. The above strategy gives the institution a customized solution whereby it is possible to have some features, such as financial education and interactive tools, run via the cloud while KYC data and personalized advice modules are kept on premises. The hybrid model will, therefore, appeal to those institutions that need to balance flexibility with security and control.

7.2 Budget

Since the delivery of the proposed model is a software-based solution, there are no Hardware components connected to the implementation. The biggest source of the cost will be for the personnel costs, hardware and software for the computing power of the machine system.

However, there will be some other costs expected that will be given in the table below.

Type	Cost
Personnel Costs	300 000
Hardware and Software Cost	100 000
Model Development and Training	50 000
Integration and Deployment	60 000
Project Management and Administration	40 000
Total Project Budget	**550 000**

7.3 Risk Management Plan

Identification of risk	Risk level	Probability for occurrence of risk	Mitigation plan
Access to sensitive customer KYC information and financial data without valid authorization.	High	High	Ensure that users' data security and integrity are fully protected through the implementation of highly efficient protocols, multi-factor authentication techniques, and regular security.
Technical issue and bugs in development	High	High	Get expert assistance and guidance to avoid failures in development
Data privacy breach due to insecure API or Firebase connection	High	High	Implement end-to-end encryption, role-based access control, and Firebase Authentication to restrict unauthorized access. Conduct regular vulnerability scans and apply patches.
Inaccurate ML model predictions due to insufficient training data	High	High	Regularly retrain the linear regression model with diverse and updated datasets. Apply cross-validation techniques and monitor model accuracy metrics above 85%.
Inaccurate ML model predictions due to insufficient training data	High	Medium	Conduct user testing and surveys. Redesign UI based on feedback using intuitive components and clear tooltips. Add educational popups and onboarding walkthroughs.
Legal or compliance issues from misinterpreting financial advice	High	Medium	Add disclaimers on financial simulation tools and outputs. Ensure that the platform guides users toward informed decisions without offering regulated investment or loan approval services directly.
Wrong/Error data input or system miscomputation on loan eligibility predictions.	High	Medium	In this view, ensure rigorous testing of prediction algorithms; also, provide options to users for manual verification and adjustment of their inputs.
Loss of dataset or hardware damage	High	Medium	Take a multiple storage backup

8. RESULTS AND DISCUSSION

8.1 Results

The personalized savings recommendation system demonstrated highly effective outcomes during the testing phase. More than 90% of users who interacted with the platform were able to receive feasible savings plans tailored to their loan request and income levels. This confirms the system's ability to translate real-time user data into actionable financial strategies. A particularly notable result was that 70% of users reached potential loan eligibility within a 3 to 6-month window, highlighting the practical relevance of the suggested savings periods.

Additionally, the integration of the machine learning model for collateral optimization proved to be a valuable enhancement. Compared to traditional or manual allocation strategies, the model helped reduce over-reliance on any single asset type by 20%. The model recommended more balanced collateral combinations that optimized eligibility while minimizing financial stress, particularly for users with multiple asset types.

The savings planner also consistently kept suggested monthly savings within the allowed threshold, exceeding 50% of the user's monthly income—ensuring practical application. These results validate the model's real-world functionality and its potential to serve as a post-rejection recovery tool for loan applicants.

8.2 Research Findings

Several key findings emerged during the system's design, implementation, and testing stages. First, asset types such as Fixed Deposits and T-Bills proved to be the most effective in closing eligibility gaps due to their high Loan-to-Value (LTV) ratios and immediate liquidity. These assets, when included in the user's portfolio, allowed for higher borrowing potential with minimal risk.[10][09][07]

Land, while valuable in monetary terms, often presents liquidity and valuation challenges. However, when used as part of a diversified asset combination, it contributed positively to the risk-reduction strategy by offering stability. In contrast, EPF was found to be frequently over-utilized in user input scenarios. Although it is loanable to an extent, its withdrawal limitations and government-related constraints made it less optimal as a primary source for loan coverage

.

The system's visual and real-time outputs also helped users better understand financial feasibility, asset allocation trade-offs, and associated risks. As users experimented with different asset inputs and durations, they were empowered to make informed financial decisions, a core goal of this research. Ultimately, these findings underscore the system's dual role as both a loan recovery facilitator and a financial literacy tool.

- FD and T-Bills offered high LTV, making them ideal for short-term leverage.
- Land was less liquid but valuable for risk offset.
- EPF was often over-committed without sufficient return on savings.

9. CONCLUSION

This thesis addresses a crucial but often overlooked challenge in digital finance: helping individuals who have been denied loans regain eligibility through strategic savings. Traditional lending models typically assess creditworthiness based solely on historical income and credit scores, ignoring future financial potential and the value of available collateral. As a result, capable borrowers are frequently excluded from credit opportunities, with no guidance on how to improve their standing.

To solve this, the project introduces an innovative digital platform that combines personalized financial guidance, real-time savings planning, and machine learning for optimal collateral allocation. The system evaluates the loanable value of various assets such as Fixed Deposits, Land, EPF, and T-Bills—using standard Loan-to-Value (LTV) ratios and calculates the shortfall between this and the desired loan amount. It then distributes this gap into achievable monthly savings targets based on user-defined timelines (one to six months), taking income feasibility into account.

A linear regression model enhances the platform's decision-making by identifying the best way to allocate multiple types of collateral, aiming for a balanced mix across low-, medium-, and high-risk assets. This reduces financial strain and improves the chances of loan approval. A risk classification feature also helps users understand the consequences of their choices by visually categorizing the risk levels of different asset combinations.

Technically, the platform is built for scalability and modularity, with a React-based user interface, a Firebase system for secure data storage, and a Python-powered predictive analytics engine. Rigorous testing confirmed its accuracy, performance, and reliability, ensuring it is ready for practical deployment.

Beyond technology, the platform emphasizes financial literacy. It educates users about asset management, budgeting, and savings strategies, empowering them to understand not just what actions to take, but why they matter for long-term financial health

10. REFERENCES

[1] A. Shaik, K. S. Asritha, N. Lahre, B. Joshua, and V. S. Harsha, “Customer Loan Eligibility Prediction using Machine Learning,” Jun. 2022, [Online]. Available: https://publishoa.com

[2] M. A. Sheikh, A. K. Goel, and T. Kumar, An Approach for Prediction of Loan Approval using Machine Learning Algorithm. [IEEE], 2020.

[3] M. Al Mamun, A. Farjana, and M. Mamun, “Predicting Bank Loan Eligibility Using Machine Learning Models and Comparison Analysis,” Jun. 2022.

[4] N. Uddin, M. K. Uddin Ahamed, M. A. Uddin, M. M. Islam, M. A. Talukder, and S. Aryal, “An ensemble machine learning based bank loan approval predictions system with a smart application,” International Journal of Cognitive Computing in Engineering, vol. 4, pp. 327–339, Jun. 2023, doi: 10.1016/j.ijcce.2023.09.001.

[5] M. Mohankumar et al., “COMPARATIVE ANALYSIS OF DECISION TREE ALGORITHMS FOR THE PREDICTION OF ELIGIBILITY OF A MAN FOR AVAILING BANK LOAN,” 2016.

[6] A. H. Huang and H. You, “Artificial Intelligence in Financial Decision Making,” 2022. [Online]. Available: https://www.ft.com/content/e082b01d-fbd6-4ea5-a0d2-05bc5ad7176c

[7] Q. Lu, Y. Luo, L. Zhu, M. Tang, X. Xu, and J. Whittle, “Developing Responsible Chatbots for Financial Services: A Pattern-Oriented Responsible Artificial Intelligence Engineering Approach,” IEEE Intell Syst, vol. 38, no. 6, pp. 42–51, Nov. 2023, doi: 10.1109/MIS.2023.3320437.

[8] S. O. Kediya et al., “Are AI and Chat Bots Services Effects the Psychology of Users in Banking Services and Financial Sector,” Aug. 2023.

[9] C. Khneyzer, Z. Boustany, and J. Dagher, “AI-Driven Chatbots in CRM: Economic and Managerial Implications across Industries,” Adm Sci, vol. 14, no. 8, Aug. 2024, doi: 10.3390/admsci14080182.

[10] Natasha Robinson and Nidhi Sindhwani, "Loan Default Prediction Using Machine Learning," [online]https://ieeexplore.ieee.org/document/10522232, [Accessed 14 May 2024]

[11] H. K. Sarisa, V. Khurana, V. C. Koti, and N. Garg, “Loan prediction using machine learning,” [online] https://ieeexplore.ieee.org/abstract/document/10466049 [Accessed 20 May 2024]

11. APPENDIX A – BACKEND

```
  >   import FaMoneyBillWave
    <FaMoneyBillWave className="mr-3 text-xl" />
    Fixed Deposit
  </button>
  <button
    onClick={() => setActiveTab("Land")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "Land"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
    <FaLandmark className="mr-3 text-xl" />
    Land
  </button>
  <button
    onClick={() => setActiveTab("EPF_ETF")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "EPF_ETF"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
    <FaPiggyBank className="mr-3 text-xl" />
    EPF/ETF
  </button>
  <button
    onClick={() => setActiveTab("T-Bills")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "T-Bills"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
    Government Bonds & Treasury Bills
  </button>
  <button
    onClick={() => setActiveTab("Best_Combination")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "Best_Combination"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
    <FaChartLine className="mr-3 text-xl" />
    Best Combination
  </button>
</div>

{activeTab === "Best_Combination" && (
  <div className="max-w-2xl mx-auto bg-white p-8 rounded-lg shadow-md">
    <h2 className="text-2xl font-bold mb-6 text-center text-red-600">
      Best Combination Analysis
    </h2>

    <div className="space-y-6">
      <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
        <div>
          <label className="block text-gray-700 font-medium mb-2">
            Select FD Account
          </label>
          <select
            value={selectedFdAccount}
            onChange={(e) => {
              handleFdAccountSelect(e.target.value);
              setBestCombination((prev) => ({
                ...prev,
                fd_value: Number(fdAmount),
              }));
```

12. APPENDIX B – TURNITIN REPOR