

SME Loan Readiness Tool: A Financial Health Dashboard for Credit Access in Sri Lanka

Master Project

In partial fulfillment for the degree of Master of Science in Information Technology

2026

The Kyoto College of Graduate Studies for Informatics
Applied Information Technology
Web Business Technology

Bawantha Harshana
M24W0179

(Fall 2025, Web System Development)

Title	SME Loan Readiness Tool: A Financial Health Dashboard for Credit Access in Sri Lanka		
Student ID Number	M24W0179	Name	Bawantha Harshana
Project Sponsor Name	Prof. Sameera Abar		

<Abstract>

Small and Medium Enterprises (SMEs) represent a critical component of Sri Lanka's economic structure, yet many of these businesses continue to experience significant challenges when attempting to access formal credit. One of the most persistent barriers is the difficulty SME owners face in demonstrating financial readiness in the structured and analytical format required by banks. Many SME operators manage their accounts manually, have limited financial literacy, or are unsure how to present their business performance in a way that aligns with lending criteria. As a result, viable businesses are often unable to obtain the funding necessary for expansion, stability, or recovery. This project was undertaken to address this issue by developing a digital tool capable of evaluating an SME's financial health and presenting it in a standardized, lender-friendly format.

The primary objective of this project was to design and implement the SME Loan Readiness Tool, a web-based application that enables business owners to enter financial information and receive automated insights. The system provides ratio calculations, risk indicators, and an overall readiness score that reflects the SME's creditworthiness from the perspective of a lending institution. The project follows a system development-oriented methodology, beginning with requirement gathering, followed by architectural design, implementation, testing, and refinement. The solution was built using a React-based frontend, a Spring Boot backend, and a PostgreSQL relational database. Authentication is implemented through JWT to ensure secure access to the platform. The methodology also included designing a financial analysis model that evaluates profitability, liquidity, leverage, and business stability, combining these variables to produce a readiness score.

During implementation, the system was tested using simulated SME financial datasets to verify the accuracy of the calculations and the stability of the platform. The AI-assisted component, which provides insights and advisory notes, was developed to enhance the usability of the dashboard and support business owners in interpreting their financial condition. The platform automatically generates a bank-ready PDF report, enabling SMEs to present their financial status in a professional and standardized manner. Testing demonstrated that the tool successfully identifies strengths, weaknesses, and areas requiring improvement across different SME profiles.

The results of the project indicate that a digital financial readiness tool can substantially improve SMEs' ability to prepare loan applications, understand their financial performance, and engage with formal credit systems more confidently. The final system integrates financial analytics, user-friendly design, and AI-driven insights to create a practical, accessible solution for Sri Lankan entrepreneurs. Overall, the project contributes to the broader effort of strengthening SME financial inclusion and demonstrates the potential of technology-driven tools to support economic development.

Table of Contents

Abstract	
Chapter 1 Introduction	
1.1 Background studies.....	1
1.2 Problem statement.....	2
1.3 Significance of the Study.....	3
1.4 Objectives and Research Questions.....	4
1.5 Organization of the Report.....	5
Chapter 2 Literature Review	
2.1 Introduction to the Literature Review.....	6
2.2 Global Perspectives on SME Financing	7
2.3 Regional and South-Asian Context.....	8
2.4 SME Financing in Sri Lanka.....	9
2.5 Information asymmetry and credit-rating theory.....	11
2.6 Digital Tools and Fintech Solutions for SME Finance.....	12
2.7 Summary of Literature Gaps.....	13
Chapter 3 Methodology	
3.1 Introduction to Methodology.....	14
3.2 Research Design.....	15
3.3 Conceptual Framework.....	16
3.4 Data collection methods.....	19
3.5 System development Methodology.....	20
3.6 Data analysis techniques.....	22
3.7 Validation and Testing.....	23
3.8 Ethical considerations.....	24
3.9 Summary.....	25
Chapter 4 Implementation and Results	
4.1 Overview of System Implementation and Evaluation.....	26
4.2 SME Viewpoint: Loan Readiness Evaluation Interface.....	27
4.3 Bank Viewpoint: SME Evaluation and Risk Interpretation.....	29
4.4 Central Policy and Administrative Viewpoint	30
4.5 AI-Driven Analysis and Decision Support	31
4.6 System Usability and Technical Performance Evaluation.....	32
4.7 Comparative Analysis with Existing Systems	33
4.8 Discussion of Results	35
Chapter 5 Conclusion and Future Work	
5.1 Conclusion.....	36
5.2 Future Work.....	37
Reference.....	38
Appendix A (Code for the SME Loan Readiness Tool).....	39
Appendix B (User Guide for the SME Loan Readiness Tool).....	177

Chapter 1 | Introduction

1.1 Background Studies

Over the centuries of technological advancement and economic evolution, small and medium-sized enterprises (SMEs) have come to be widely recognized as the backbone of the modern global economy. Unlike large-scale corporations, SMEs operate with direct and tangible objectives and maintain close connections with local markets. As a result, they contribute significantly to job creation, innovation, and income generation in both developed and developing economies. Previous studies have shown that SMEs account for a substantial share of employment and economic activity, often exceeding the contribution of large enterprises in terms of employment growth and entrepreneurial innovation (Beck & Demirguc-Kunt, 2006). Due to their flexibility and adaptability, SMEs play a critical role in responding to changing economic conditions and sustaining long-term economic stability.

Globally, SMEs operate in diverse sectors and are highly responsive to local economic and social conditions. Their ability to adapt business models to regional needs makes them essential contributors to national economic resilience and sustainable development (Berger & Udell, 1998; OECD, 2019). In many countries, SMEs act as drivers of inclusive growth by creating employment opportunities, supporting rural development, and encouraging innovation at the grassroots level. Consequently, strengthening the SME sector has become a key policy priority for governments and international development institutions.

In the Sri Lankan context, SMEs represent a vital segment of the national economy. According to the Central Bank of Sri Lanka (2022), SMEs account for nearly 75 percent of all registered businesses in the country and contribute approximately 52 percent of the national Gross Domestic Product (GDP). Furthermore, the Asian Development Bank (2021) reports that SMEs provide more than 45 percent of total employment opportunities in Sri Lanka. These figures highlight the importance of SMEs as a fundamental force in sustaining economic activity, reducing unemployment, and supporting household incomes across the country.

Despite their economic importance, SMEs in Sri Lanka face several structural and financial challenges that limit their growth potential. One of the most significant obstacles is limited access to formal financial services, particularly bank credit. Many small businesses struggle to obtain the financial resources required to expand operations, adopt new technologies, or withstand economic shocks (World Bank, 2020; IFC, 2017). This challenge has been further intensified by Sri Lanka's recent economic crisis, foreign exchange shortages, and rising interest rates, which have increased financial pressure on SMEs.

Although many SMEs operate viable and profitable business models, they often fail to secure bank loans due to their inability to present accurate and standardized financial information (Gamage, 2013). This situation creates a condition of information asymmetry between SMEs and financial institutions. Banks rely heavily on financial ratios, profitability indicators, and formal reports when evaluating loan applications. However, a significant number of SME owners lack the technical knowledge and financial literacy required to prepare such analyses (Berger & Udell, 1998; Central Bank of Sri Lanka, 2022). As a result, trust gaps emerge between banks and SMEs, limiting access to credit and constraining the growth of the SME sector.

1.2 Problem Statement

Although small and medium-sized enterprises (SMEs) are recognized as the backbone of Sri Lanka's economic development, they face many challenges in accessing the financial facilities and credit services necessary for their sustainable growth. Many small and medium-sized enterprises in Sri Lanka fail to obtain the financial support they need to progress their businesses, which is considered one of their main commercial constraints (World Bank, 2020). This financing gap has directly limited their ability to expand their operations, introduce new technologies, and access global markets.

The scale of this problem is well illustrated by statistics. According to reports from international institutions such as the Asian Development Bank and the World Bank, the overall financing gap for small, medium and medium-sized enterprises in South Asia exceeds US\$330 billion, and Sri Lanka's SME sector faces a significant portion of this gap. Due to the difficulty of obtaining loans from the formal banking system, many SME entrepreneurs are tempted to turn to unregulated informal lending channels with high interest rates, which further exposes them to financial risk.

A major reason for this financial gap is the “information asymmetry” between banks and SME entrepreneurs. Formal banks rely heavily on financial indicators such as a business's profitability, liquidity, and leverage ratios when making lending decisions (Berger & Udell, 1998). However, many small and medium-sized business owners in Sri Lanka do not have standardized accounting systems, formal financial reports, or future revenue projections. Due to the lack of this information, banks are unable to accurately assess the business's true creditworthiness, which increases the likelihood of loan applications being rejected.

The severe economic crisis that Sri Lanka faced in 2022 has further exacerbated this situation. The survival of SMEs has been severely threatened by foreign exchange shortages, difficulties in importing raw materials, skyrocketing interest rates and overall economic uncertainty. In the face of this macroeconomic instability, banks have also reduced their risk appetite when granting loans, and the provision of credit to the SME sector has become more limited. As a result, many businesses have had to close or reduce their workforce, which has directly affected unemployment and the country's GDP.

Regionally, countries such as India and Bangladesh have taken steps to make SME credit scoring models more systematic and efficient by using digital technology and data analytics. This has reduced information asymmetry and enabled more SMEs to access the formal financial system. However, in Sri Lanka, there is still a clear need for a specialized technological solution to assess the financial readiness of SMEs and effectively connect them with the banking system.

In order to address the multiple issues identified above, this research proposes to design and implement an “SME Loan Readiness Tool”. This system will allow SME entrepreneurs to enter their basic financial data, automatically calculate key financial ratios expected by banks, and provide a preliminary assessment of their business's creditworthiness. It will also provide guidance on preparing the necessary documents for a loan application and improving financial literacy.

1.3 Significance of the Study

The primary objective of this research is to create a web application and solution called “SME Loan Readiness Tool”. It will help small and medium enterprises in Sri Lanka analyze their financial situation and assess their readiness to obtain loans. SMEs are a fundamental force in the country’s economy today, and increasing their access to finance is an essential part of strengthening the country’s economy. Therefore, the importance of this research is to establish a new loan preparation system for SMEs using technology and develop their financial management capabilities and confidence.

The system proposed in this research enables SMEs owners to accurately analyze their business’s financial data. The system provides the ability to automatically calculate key financial ratios such as profitability, liquidity, and leverage, which makes it easier for business owners to assess their organizational situation. This increases their financial literacy and contributes to informed decision-making (World Bank, 2020), (IFC, 2017; OECD, 2019). Since the technology enables the user to identify the strengths and weaknesses of their organization, it also increases their confidence in applying for loans.

A major obstacle facing many SMEs in Sri Lanka is the lack of trust with banking institutions. The “Loan Readiness Tool” system developed in the research can convert the financial information submitted by SMEs into a standard format and submit it to the banking sector. This is a new step that builds trust by reducing information asymmetry between banks and businesses. This increases the speed of loan application approval and reduces bank risk. (ADB, 2021) Such systemization can prevent the loan process from becoming more complicated.

This research is directly linked to the dual policy mechanisms of the Government of Sri Lanka, namely the National Policy Framework for SME Development (2020) and the United Nations Sustainable Development Goals (SDGs). Technological innovation, increasing access to credit, and business data systematization are prominently mentioned in the government’s SME policies. Therefore, this research can be considered as a strategy to achieve those policy goals. SDG 8 (Decent Work & Economic Growth) and SDG 9 (Industry, Innovation & Infrastructure) are also relevant to this. Therefore, this research is consistent with both national and global development goals (GOSL, 2020).

The results of this research have direct practical value. Using it, financial institutions, government agencies and researchers can analyze access to credit for SMEs and design new policies. Furthermore, it helps research students in universities and educational institutions to develop new ways of analyzing technical and economic data. This strengthens the connection between research and both the practical and educational fields (OECD, 2019).

1.4 Objectives and Research Questions

Every research study must be guided by well-defined objectives that clarify the purpose, direction, and expected outcomes of the investigation. This study focuses on developing a digital solution named the “SME Loan Readiness Tool” to improve access to finance for small and medium-sized enterprises (SMEs) in Sri Lanka. The formulation of research objectives and questions provides a structured framework that ensures the study remains aligned with its main goal: to use technology as a bridge between SMEs and financial institutions. These objectives are designed to address the key issues identified in the problem statement, such as limited access to credit, lack of financial knowledge, and the information gap between SMEs and lenders.

Main Objective:

The main objective of this research is to design and develop a web-based application (Loan Readiness Tool) capable of evaluating the loan readiness of SMEs in Sri Lanka through automated financial analysis.

This central objective reflects the innovative and technological foundation of the study. The tool aims to empower SME entrepreneurs by giving them a clearer understanding of their financial health, helping them prepare for formal credit applications, and improving their interactions with financial institutions. By digitizing and standardizing financial data, the application contributes to enhancing both financial literacy and transparency in the lending process.

Specific Objectives:

1. To identify factors affecting access to finance for SMEs in Sri Lanka.
2. To analyze key financial ratios used in bank lending decisions.
3. To create a user-friendly web interface where SME owners can enter relevant data.
4. To automatically generate financial reports and risk analyses through the system.
5. To examine the increase in financial literacy and creditworthiness of SMEs through the tool.

Research Questions:

1. What factors limit access to credit for SMEs in Sri Lanka?
2. How does the use of financial analysis tools affect SMEs' loan readiness?
3. How can the Loan Readiness Tool improve the quality of bank applications for SMEs?

1.5 Organization of the Report

This report is systematically structured to present the research in a clear, logical and progressive manner. Each chapter is organized to build on the previous chapter, providing a continuous flow from identifying the research problem to developing a practical solution. The structure has been carefully designed to enable the reader to easily understand the objectives, methodology and results of the study related to the “SME Loan Readiness Tool” and its role in solving the financial access problems faced by small and medium-sized enterprises (SMEs) in Sri Lanka.

Chapter 1 – Introduction This chapter introduces the overall background of the research and the financial constraints faced by SMEs in Sri Lanka (especially difficulty in obtaining credit). The purpose, significance and research problem of creating the proposed “Loan Readiness Assessment Tool” are explained here.

Chapter 2 – Literature Review This chapter examines studies conducted by institutions such as the World Bank, ADB, and academic research on SME financing, credit constraints, and digital solutions. It identifies gaps in current knowledge and provides theoretical justification for why this research is needed in Sri Lanka.

Chapter 3 – Research Methodology This chapter describes the methods and techniques used to achieve the research objectives. It also explains the system development process and its testing methodology for creating the proposed “SME Credit Readiness Assessment Tool”.

Chapter 4 – System Implementation and Results This presents the technical implementation of the proposed web application and its functionality. The evaluation results of the system, such as calculating financial ratios, identifying risks, and generating reports required by the bank, are discussed here with the help of diagrams.

Chapter 5 – Discussion This chapter interprets the research results in relation to the theories and issues identified in the literature review. It examines the practical benefits and significance of the developed tool for banks, policy makers and SME owners.

Chapter 6 – Conclusion and Recommendations summarizes the main findings of the research and assesses whether the study objectives were met. Recommendations for improving SME finance access, increasing financial literacy, and future research are presented.

Conclusion Overall, this report provides a theoretical contribution as well as a practical solution to the financial access constraints faced by the SME sector in Sri Lanka. This “SME Credit Readiness Assessment Tool” is expected to make an innovative contribution to the country’s economic recovery and strengthening the SME sector

Chapter 2 | Literature Review

2.1 Introduction to the Literature Review

The purpose of this chapter is to review the research and scholarly literature written on the issue of access to credit for small and medium-sized enterprises (SMEs) in Sri Lanka and its solutions. A literature review provides the necessary background for the research and also helps to identify the weaknesses of previous research. This enables the researcher to proceed with the research on a more solid foundation.

A literature review allows the researcher to understand who has done what before, with what methods, and with what results. It can also identify areas of new research focus and identify and refine the disciplines of previous research. For example, Beck & Demirguc-Kunt (2006) have shown that access to finance for SMEs is a challenge in a global system, which is characterized by a wide range of economic downturns and governance systems. By examining the lessons learned from such research, this research can be given new directions.

This literature review focuses on key categories such as access to credit for SMEs, information asymmetry, bank credit assessment processes, and FinTech solutions. First, the exploration of access to finance for SMEs identifies factors that influence access to credit. Second, research on information asymmetry examines the decline in trust between banks and businesses. Finally, the literature on FinTech solutions analyzes the revolution that technology is bringing to the financial sector.

The literature review creates a bridge from ancient research to the latest explorations. Initially, the basic ideas on SMEs' access to finance issues are considered in research from the 1990s–2000s, and later, new ideas on FinTech and credit analytics systems after 2015 are examined. Here, a systematic inquiry is presented for the research by comparing different scientific ideas and methodologies.

Nowadays, FinTech, AI and digital systems are accelerating loan assessment and post-COVID-19, it is seen as a new trend in the global environment. In particular, information automation and risk scoring have increased trust between banks and SMEs. This research also moves with that new trend and proposes a technological “Loan Readiness Tool” solution for SMEs in Sri Lanka.

Accordingly, this chapter provides a basic background for the research, organizing knowledge on credit access and technology solutions for SMEs

2.2 Global Perspectives on SME Financing

Small and medium-sized enterprises (SMEs) play a significant role in the global economy, not only in job creation but also in income distribution and innovation. According to the World Bank (World Bank, 2020), SMEs account for more than 60% of global employment and 40% of national income. However, in many countries, especially developing countries, access to credit and financing for SMEs remains a major challenge (Beck & Demirguc-Kunt, 2006). This suggests that the future growth of SMEs in the global business system remains challenging.

Global researchers have pointed out that there are several major barriers to accessing credit for SMEs. Beck and Demirguc-Kunt (2006) point out that many banks consider lending to SMEs to be a risk, which leads to many loan applications being rejected. According to Berger and Udell (1998), the lack of accounting and financial documentation required by SMEs owners reduces bank confidence. Other scholars have pointed out that information asymmetry is the inability of banks to know the true state of the business, which limits credit requirements.

Different scientific courses in the global literature show different perspectives. Some believe that the banking system and control mechanisms are failing. Other researchers show that the main reason is the financial knowledge and control weaknesses of SMEs owners. Ayyagari, Beck and Demirguc-Kunt (2007) have shown that businesses with weak financial controls have a lower ability to obtain credit, which is a major problem in developing countries. Another idea is that the use of FinTech and digital systems can reduce these problems.

According to the Asian Development Bank (2021), the global financing gap for SMEs is around USD 5.2 trillion. Of this, 40% of the total gap is in developing countries. The approval rate for SMEs, especially in South and East Asian countries, is extremely low. More than 30% of SMEs worldwide do not even apply for credit, due to lack of confidence and high interest rates. This situation, in particular, shows that financial systems in developing countries are still weak in terms of access for SMEs.

Recent research has shown that FinTech and digital systems have had a major impact on increasing access to finance for SMEs. New credit scoring systems, AI-based risk assessment modules, and automation technologies have identified new ways to build trust between banks and businesses (OECD, 2019; World Bank, 2020). This research also follows this trend and proposes a new technological solution for SMEs called the “Loan Readiness Tool”.

Accordingly, global research suggests that the decline in access to credit for SMEs is primarily due to information asymmetry, poor analysis of bank risks, and lack of trust. However, new technological solutions based on FinTech and automation show that this problem can be successfully mitigated. Therefore, the global backlog of research provides relevant mechanistic and policy guidance for Sri Lanka as well.

2.3 Regional and South-Asian Context

South Asia, comprising India, Bangladesh, Nepal, Pakistan, and Sri Lanka, is a region with large populations and diverse economic structures. SMEs in these countries are a core part of the economy and create close to 70% of employment opportunities (Asian Development Bank, 2021). However, access to finance remains a major challenge across the region.

India is a country that has implemented successful government and technology programs for SMEs. Institutions such as Mudra Loan Scheme, Credit Guarantee Fund Trust (CGFT) and SIDBI provide secured credit support to SMEs. The integration of FinTech and digital systems has accelerated loan application and appraisal (IFC & World Bank, 2021).

The microfinance system pioneered by Grameen Bank in Bangladesh has made a significant contribution to increasing access to finance for SMEs (Yunus, 2010). It is considered a successful financial innovation for the region's previous problems. Today, FinTech and mobile banking services are further strengthening this model.

Countries like Pakistan and Nepal also have microfinance and bank guarantee programs for SMEs. However, according to the Asian Development Bank Institute (ADBI, 2020), these countries' systems are technologically backward, which has led to weak data analysis and credit assessment processes. The lack of information systems for rural SMEs, especially in Nepal, has led to a lower loan approval rate. However, new models such as FinTech and mobile banking are gradually increasing their access to finance.

Countries in the South Asian region often face similar challenges. Among them, high interest rates, collateral requirements, information asymmetry, and legal complexities were identified as key challenges (ADB, 2021; OECD, 2019). These challenges are further exacerbated by the lack of capacity of SMEs owners to prepare proper accounting systems and documentation, and the state of risk management in banking institutions.

Ultimately, FinTech innovations appear to be a viable solution for this region. Countries such as India and Bangladesh have digitized the lending process for SMEs using mobile banking, e-wallet systems, and AI credit assessment tools (World Bank, 2020). This has reduced information asymmetry and increased the speed of loan approval. These representative models can also serve as a model for Sri Lanka to create a new FinTech-based loan readiness tool.

Accordingly, countries in South Asia have made various attempts to increase access to finance for SMEs but have yet to find a complete solution. The region is reeling from a technological and legal setback, and it is imperative for Sri Lanka to learn from those experiences and create a local FinTech solution. Therefore, the following chapter 2.4 – The Situation in Sri Lanka examines the local reality within this global and regional setback.

2.4 SME Financing in Sri Lanka

Small and Medium Enterprises (SMEs) in Sri Lanka are a key sector of the country's economy. Such businesses create over one million jobs and contribute to 52% of the national GDP (Central Bank of Sri Lanka, 2022). SMEs, mostly located in the eastern districts, also contribute strongly to the growth of export earnings. However, many face challenges in accessing the capital needed to successfully implement their growth plans. In particular, banking system underwriting requirements and risk assessment policies restrict SMEs' access to capital.

Many SMEs in Sri Lanka fail to raise the capital they need to grow their businesses. The World Bank (2020) points out that access to finance for this sector is a major challenge in the country. Gamage (2013) cites factors such as collateral requirements, high interest rates, lack of proper documentation systems, and information asymmetry (SME owner vs bank information gap). Therefore, the lack of other reliable history of a business by bank officers is also a risk. SMEs often have limited financial reporting systems and strategic analysis, which reduces confidence in advising banks. Additional financial institutions are also seen to be adopting other strategies in this regard, but this is currently with limited results at the local level.

The Sri Lankan banking sector has been adopting a number of metrics such as ratio, profitability, leverage ratio, and operational cash-flow when making financial decisions (Berger & Udell, 1998). However, even after several years, many SMEs do not have the skills to collect or organize such data. Therefore, it is a challenge for the bank to identify a "risk profile". According to CBSL (2022) data, at least 30% of business loan applications issued by banks are approved, while another 35% are suspended for a period of time. This data shows that the banking system is limited in its approval response and the acceptance criteria are difficult.

The government introduced Enterprise Sri Lanka, National Policy Framework for SME Development (2020), Credit Guarantee Schemes, to develop the SME sector. The aim of such schemes was to provide low-cost financing to businesses through low interest rates. Although this was successful in its initial development, the subsequent economic crisis led to a reduction in fund allocation. According to the GOSL (2020) announcements, the development of such schemes was forced to be suspended due to COVID-19. However, there is an opportunity to implement a new financing framework in the future, using the experience of the bank-government partnership model used.

The COVID-19 pandemic has resulted in a sharp decline in the market, shortages of imported raw materials, and market disruptions for SMEs. The 2022 crisis, coupled with rising interest rates on income, currency depreciation, fuel shortages, and tightening of bank risk regulations, has created new obstacles for SMEs. According to the Central Bank (2022), the scale of investment has again fallen below 10%. This has led to business closures, job losses, and a decline in financial confidence. However, the lessons learned from this crisis can also help in the development of future systems.

Information asymmetry refers to the imbalance of information between the bank and the business. Many SMEs lack proper accounting records, do not file annual accounts, or use fabricated data, making it difficult for bank agents to assess the true financial stability of a business (Berger & Udell, 1998; Gamage, 2013). This creates a lack of trust and causes banks to require additional collateral. However, this situation can be corrected using FinTech. This asymmetry can be reduced by establishing an information input-output system with detailed financial data, such as business portals, financial accounting applications, and online data sharing systems.

Financial institutions are now trying to provide support to SMEs in a strategic and efficient manner through the development of new technological systems. In particular, FinTech (“Financial Technology”) based systems are growing all over the world. For example, “Digital Credit Scoring Platforms” that have entered countries like India and Bangladesh provide support to SMEs. By following these policies, the time to apply for a loan is reduced, and the applicant’s financial situation can be analyzed data-driven and successful decisions can be made. Although the FinTech sector in Sri Lanka is also in its infancy, the banking sector has started online loan processing, mobile banking, QR payment systems and KYC automation. Such technological efforts look attractive to SMEs as they provide new experiences in payment security, high transparency, information exchange, etc. It is expected that FinTech based SME loan readiness tools can make a real difference in the future.

The above analysis shows that the SME sector in Sri Lanka, although a vital pillar of the country’s economy, is still constrained in terms of access to finance. Information asymmetry, lack of documentation, high interest rates and economic instability exacerbate this problem. However, new FinTech technologies can mitigate this problem.

Therefore, the aim of this research is to develop a data-driven, automated “SME Loan Readiness Tool”. The system is a fundamental support for rebuilding the information relationship between banks and businesses and increasing trust. It can be a basis for obtaining a new transparent basis for reasoning about the problem in connection with Chapter 2.5 “Information Asymmetry Theory”.

2.5 Information asymmetry and credit-rating theory

The concept of “Information Asymmetry” was first introduced by economist George Akerlof (1970). According to his “Market for Lemons” theory, when there is information asymmetry between buyers and sellers in a market, a lack of trust is created. This makes it reasonable for buyers to not make correct decisions and creates “quality uncertainty” in the market. This theory later became a key concept in analyzing credit criteria policies in the banking sector.

According to this theory, information asymmetry is when sellers (e.g., SME owners) have more knowledge about their business situation than buyers (e.g., banks). When banks are unable to assess the applicant's income, expenses, cash flow, and future payment capacity, they will take on more risk and require more collateral (Berger & Udell, 1998). As a result, banks reject truly valuable business opportunities.

SMEs in Sri Lanka are often affected by information asymmetry due to the lack of proper accounting systems. Gamage (2013) suggests that many SMEs do not maintain accounting records and lack documentation systems, which makes it difficult for bank officials to assess future payment capacity. Central Bank (2022) data shows that a large number of bank applications are rejected due to insufficient data. Thus, excessive information asymmetry hinders SME growth and also holds back local business technology.

Bank credit assessment policies around the world are based on data-driven ratio analysis – such as profitability ratios, liquidity ratios, debt-equity ratios and cash-flow coverage. However, IFC (2017) notes that SMEs often lack proper accounting records and are unable to fill in this data. In such a situation, banks tend to increase risk and engage in “credit rationing” – that is, not providing funds to applications based on limited data. These various assessment constraints ultimately hinder even the most viable business or innovative applications.

Scholars disagree on whether Information Asymmetry is a flaw in the banking system or a weakness in SME governance. Stiglitz & Weiss' (1981) Credit Rationing Theory shows how banks limit their capital due to information asymmetry. However, according to Bester's (1985) Signaling Theory, the collateral or profit ratio that is supposed to indicate the value of a business is seen as a credible signal. Modern views also clearly show that asymmetry can be reduced through technological data integration and computerized business finance systems (OECD, 2019).

The “SME Loan Readiness Tool” application proposed in this research is a technological model that contributes to solving the above problem. This system allows SMEs to display their financial status in a data-driven manner, helping to rebuild trust in bank officials. Automated ratio calculations, financial statement upload functions, and certain table elements can minimize information asymmetry. Therefore, the implementation of the Loan Readiness Tool is a step towards FinTech in the bank loan assessment process.

According to the theory of “Information Asymmetry”, the lack of information between the business and the bank creates a lack of trust, making it difficult to obtain loans. Therefore, it is necessary to clarify information and increase trust through systems such as the FinTech-based Loan Readiness Tool.

2.6 Digital Tools and Fintech Solutions for SME Finance

FinTech is a way of modernizing financial services using technology. It moves traditional banking processes into a digital environment through the web, mobile apps, and AI. FinTech makes it easier for people and businesses to apply for loans, make payments, and share data (OECD, 2019). Many countries around the world have used this technology to speed up their banking processes and provide greater convenience to consumers.

China, the US, and Asian countries in particular are innovating in the FinTech sector. For example, WeBank (China) and Ant Group have used AI-based risk analysis to reduce loan approval times from days to minutes (World Bank, 2020). FinTech services such as Kabbage (USA) have automated the process of granting loans to SMEs and made information entry much easier. These systems have saved customers time and money, and enabled banks to make more informed decisions.

Countries in South Asia are also joining this new technological revolution. India's Unified Payments Interface (UPI) system is a prime example of the FinTech revolution, enabling efficient interbank payments and credit services (ADB, 2021). Bangladesh is also expanding its FinTech sector, using online systems to expedite loan applications for SMEs. These regional examples show that FinTech can increase financial access.

Although the FinTech sector in Sri Lanka is still in its infancy, several steps have been taken. LankaPay, the National Payment Platform, and mobile payment systems have accelerated data exchange between banks. However, specific FinTech applications for SMEs are still lacking. The Central Bank of Sri Lanka (2022) shows that the lack of technical infrastructure, data security, and regional access have led to the lag in FinTech adoption. However, the government and the private sector are working together to develop this sector.

FinTech systems are built on new technologies. AI and Machine Learning automate risk analysis and credit scoring. Digital KYC (Know Your Customer) speeds up customer verification and builds trust. Real-time monitoring and dashboard features provide both banks and businesses with real-time information. All of this reduces information asymmetry and increases trust and transparency (OECD, 2019).

There are also several challenges with the introduction of new FinTech systems. First, data security and privacy are important issues. Another problem is that legal and policy systems are not yet fully compatible with the FinTech sector. The lack of overdraft facilities and internet access in rural areas also appears to be a challenge. The World Bank (2020) shows that government and private sector support is essential for FinTech to succeed.

The “Loan Readiness Tool” proposed in this research is a practical example of the FinTech concept in action. It automatically analyzes the financial data required for SMEs to obtain loans and shows the readiness of bank applications. This clearly provides the information needed by the bank, increasing confidence for SMEs. This reduces information asymmetry and facilitates access to finance.

2.7 Summary of Literature Gaps

The literature reviewed in Chapter 2 of this study shows that access to finance for SMEs remains a major challenge at the global, regional, and local levels. Although SMEs are a key force in economic development in many countries, information asymmetries, legal complexities, and governance weaknesses severely impact their access to credit and financing (ADB, 2021; World Bank, 2020).

Global research is often based on experiences with FinTech systems and credit innovations in developed countries. However, it seems that these systems cannot be directly implemented in developing countries, especially in an environment like Sri Lanka. These models are not practical due to differences in technological structures, regulatory frameworks, and data protection (OECD, 2019). As a result, there is a lack of research on adapting FinTech models and designing new systems in developing countries.

South Asian research has often relied on microfinance and community finance models. While they have benefited small businesses, FinTech-based SME loan analytics systems and data-based credit platforms have been minimally developed (ADBI, 2020). This suggests that information systemization and digital finance access are not yet fully operational at the regional level.

Research in Sri Lanka has often stopped at the level of regulatory and policy studies for SMEs. Reports by Gamage (2013) and Central Bank (2022) show that banking processes, collateral requirements, and trust issues are cited as reasons for poor access to credit, but there is no technical solution to them. As there is no local research on automated credit readiness tools or data-driven analysis systems, Sri Lanka needs a specific FinTech experience.

Another important gap is that most research has focused on the concept of Information Asymmetry from an economic perspective and has neglected the technical aspects. The World Bank (2020) shows that there is little research on data modeling and system design. Therefore, the scientific principles related to loan assessment automation systems and FinTech integration are still lacking.

Considering all these shortcomings, the “Loan Readiness Tool” proposed in this research provides a new technological foundation for SMEs in Sri Lanka. It automates the loan assessment process by clarifying the information relationship between banks and businesses. Through this, it has the potential to reduce information asymmetry, rebuild trust, and increase the speed of loan approval.

As per the above analysis, although previous research has described the SME financing problem in various ways, there has been no research on new FinTech-based loan readiness solutions or automated data systems. Therefore, this research aims to fill that gap and take the SME financing approach in Sri Lanka to a new technological level.

Chapter 3 | Methodology

3.1 Introduction to Methodology

Research methodology is a set of steps and methods followed to conduct research in a systematic and scientific manner. It is an element that determines the correct direction of the research, the methods of data collection, the analysis process, and the reliability of the results (Creswell, 2014). Introducing a research methodology ensures the scientific quality of the research, and it is a fundamental step in making the research reliable and practical.

This research is a System Development Based Applied Research. Its objective is to develop a digital tool that can measure the financial readiness of SMEs in Sri Lanka before they seek loans. Therefore, the research is implemented as a model that combines both concept research and system development. The reason for choosing this direction is that it is possible to create a technical solution and perform the necessary data analysis at the same time.

The methodology of this research is built on the following elements:

1. Research Design – The rules and technical model for designing the research.
2. Data Collection Methods – The methods and devices used to collect the necessary information.
3. System Development Process – The technical steps in creating the “Loan Readiness Tool” application.
4. Data Analysis – Analyzing the collected data and obtaining reliable results.
5. Validation Procedures – Testing the effectiveness and reliability of the system.

All these parts are described in detail in Chapters 3.2 to 3.8.

The research methodology was selected based on previous research and theoretical foundations. The Mixed-Method Approach described by Creswell (2014) shows that qualitative and quantitative data can be used together. The research was also planned step by step based on the Research Onion Model presented by Saunders et al. (2019). This ensures that the research is conducted on a scientific basis and the reliability of the results is ensured.

Accordingly, this chapter explains the basic idea, purpose, and elements of research methodology. This provides a philosophical guide to conducting research in a scientific and systematic manner. The next section, 3.2 – Research Design, describes in detail how to systematically design the research and the chosen model.

3.2 Research Design

A research design is a basic plan for conducting research in a systematic and planned manner. It can be considered a “nautical map” of the research, and helps in the correct application of the methods, steps, and techniques required to achieve the research objectives (Creswell, 2014). In this research, the purpose of selecting the design was to use a method that combines data analysis and technical development. Therefore, the research is implemented as a combination of Applied Research and System Development Design.

This research adopts a Mixed-Method Applied Design model. That is, both qualitative and quantitative data are used together. On the qualitative side, the research analyzes the credit access problems, information asymmetry and governance weaknesses of SMEs in Sri Lanka. On the quantitative side, the system automatically performs financial ratio calculation, readiness scoring and ratio analysis. By using these two methods together, it is possible to successfully develop a FinTech-based Loan Readiness Tool. Therefore, the research is based on the System Development Life Cycle (SDLC) model.

There are several reasons for choosing this model.

- First, measuring loan readiness for SMEs requires both technical and financial perspectives. Therefore, a mixed-method approach is appropriate.
- Second, using the System Development methodology, the research can be designed not only as a study, but also as a practical product (Loan Readiness Tool).
- Third, the use of information asymmetry and FinTech theory strengthens the scientific basis of the research (Berger & Udell, 1998; ADB, 2021).

This research consists of the following stages:

- Data Collection – Collecting financial data from SMEs and bank records in Sri Lanka.
- Data Analysis – Analyzing ratios such as profitability, liquidity, debt ratio.
- System Development – Creating a Loan Readiness Tool using Spring Boot, PostgreSQL, and React.
- Validation – Testing the system and assessing user feedback and reliability.

This stage operates on the Iterative SDLC Model. That is, each stage receives feedback and updates the system (Pressman & Maxim, 2020).

This research is not only educational but also has a practical purpose. Creating a FinTech-based financial analysis tool is a new step for SMEs in Sri Lanka. When choosing the research model, it was considered that it is a way to connect both technical structures and financial analysis methods together. This can provide a practical solution that is suitable for a business need and can be implemented.

As described above, the research model is designed in a systematic and scientific manner. It integrates data analysis and system development, allowing the creation of a FinTech basic credit readiness measurement tool. Therefore, the research model serves as a fundamental principle of this research.

3.3 Conceptual Framework

A conceptual framework is a graphical guide that explains the relationship between the basic concepts, data, and processes used in a research. It can be considered as a “navigational map” of the research, and it depicts the input, process, and output stages required to achieve the research objective in an understandable way (Creswell, 2014). It clearly shows the main idea of the research the journey of the entire problem, starting from data collection, through analysis, and ending with the final output.

The purpose of the conceptual framework of this research is to accurately explain the relationship between the data flow, the process, and the final output of the research. Accordingly, a systematic journey is depicted as input data obtained from SMEs → analysis and readiness scoring processes occurring in the system → the final bank-ready report.

In the input stage, the necessary financial data is collected from SMEs. This includes information such as income, expenses, existing debt, property, and fund status. The correct and complete entry of this data is a key factor in determining the reliability of the system. This prepares the basic data required for the ratio analysis and readiness calculation required by the system.

Then the processing phase begins. Here, the system calculates ratios such as profitability, liquidity, debt ratio and automatically creates a readiness score based on those values. This clearly shows the ability of a business to obtain credit. This reduces the information asymmetry under the FinTech concept and rebuilds trust between banks and businesses.

In the output phase, the system provides a bank-ready report. It is a complete report that includes the financial position of the SMEs, ratio analysis, readiness score, and necessary suggestions. The feedback function of the system also allows the business to update its data and improve its situation. Thus, the output phase is the final step in rebuilding information trust at a systemic level.

These three stages — input, process, and output — are interconnected and operate in a systematic manner. If the input data is not correct, there is a risk that the readiness score of the process stage will be incorrect, which ultimately affects the reliability of the output report. Therefore, the relationship Input → Process → Output is a fundamental principle that determines the accuracy of the system.

Finally, this conceptual framework serves as a guide to the research system development process. It explains the relationship between information connectivity, analysis methods, and the production of output reports.

3.3.1 Conceptual System Architecture of the SME Loan Readiness Tool

Conceptual System Architecture - SME Loan Readiness Tool

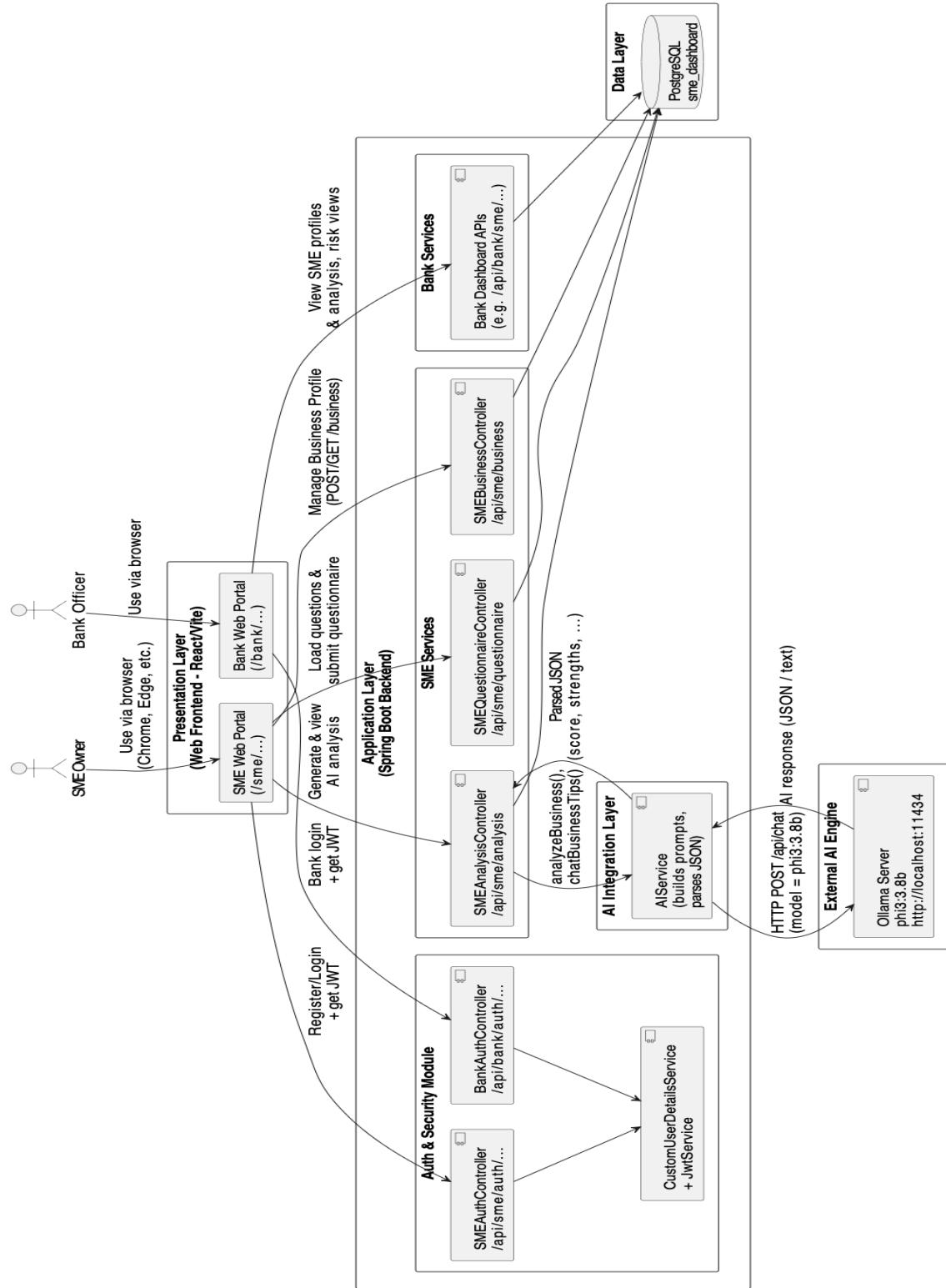


Figure 1 : Conceptual System Architecture

Figure 1 presents the conceptual system architecture of the SME Loan Readiness Tool, which is designed as a layered web-based system to support SMEs, banks, and policy-level users in evaluating loan readiness. The architecture follows a multi-layer approach to ensure clear separation of responsibilities, secure data handling, and scalable system performance.

At the presentation layer, the system provides two dedicated web portals developed using React. One portal is designed for SME owners, while the other supports bank officers. These portals are accessed through standard web browsers and serve as the primary interface for all user interactions. SME owners use the portal to register, log in, manage business profiles, complete financial questionnaires, view AI-generated insights, and download bank-ready reports. Bank officers access a separate portal to review SME profiles, assess readiness scores, and analyze risk-related information. The user interface is designed to be intuitive and responsive, allowing users with limited technical knowledge to interact with the system effectively.

The application layer is implemented using the Spring Boot framework and functions as the core processing unit of the system. This layer handles authentication, business profile management, questionnaire submission, financial ratio calculations, readiness score generation, and communication between different modules. Separate controllers are implemented for SME services, bank services, questionnaire handling, and financial analysis to ensure structured request processing and modular design.

Security is managed through a dedicated authentication and authorization module using JSON Web Tokens (JWT). Separate authentication controllers are provided for SME users and bank users, enabling role-based access control and secure session management. This ensures that sensitive financial data and analysis results are only accessible to authorized users.

An important component of the architecture is the AI integration layer, which connects the backend system to an external Ollama server hosting a large language model. This layer generates qualitative insights such as strengths, weaknesses, recommendations, and business coaching responses based on SME data. These AI-driven outputs complement numerical financial ratios by translating them into meaningful guidance for decision-making.

Finally, the data layer consists of a PostgreSQL database that stores all persistent information, including user details, SME profiles, questionnaire responses, analysis outputs, and readiness scores. This architecture enables smooth data flow, secure processing, and intelligent analysis, supporting the overall objective of improving SME access to credit.

3.4 Data Collection Methods

The reliability and accuracy of a research is determined by its data collection methodology. Data collection is the systematic collection of information necessary to fulfill the research objectives (Saunders et al., 2019). The purpose of data collection in this research was to analyze credit access issues for SMEs in Sri Lanka and obtain financial data required for system development. This research uses two types of data: secondary data and simulated data.

1. Secondary Data:

- SME reports published by the Central Bank of Sri Lanka (Central Bank of Sri Lanka, 2022)
- Asian Development Bank (ADB, 2021) and World Bank (World Bank, 2020) publications
- Research articles, scientific journals, and financial authority reports

2. Simulated Data:

- Example data that SMEs owners mark as input into the application during system development
- Estimated notes based on income, expenses, debt, and investment values
- These were used for system testing and ratio calculation

Citation: (Central Bank of Sri Lanka, 2022; ADB, 2021; World Bank, 2020)

Reliability and timeliness were considered in the selection of data sources. The use of Central Bank reports allowed us to show the situation of local SMEs, while ADB and World Bank reports provided regional and global relevant figures. The reason for choosing simulated data was that there was a limitation in obtaining real SMEs data when testing the system. Accordingly, the data sources of the research were primarily selected in a factual and practical manner.

The data collection process of the research was carried out in a step-by-step and organized manner. First, secondary data reports and articles on SMEs in Sri Lanka were collected, and then simulated financial data were prepared for the system database. The data was entered into the PostgreSQL database using Excel and CSV formats, and the accuracy of the values was checked. This process increased the ability to obtain reliable and accurate results when calculating the system readiness score.

All data used in the research were kept confidential and secure. Since the secondary data used were obtained from public sources, no personal information was included, and there was no risk to real business information by using simulated data. This enabled the ethical and scientific value of the research to be maintained (Creswell, 2014).

It was recognized that there were several limitations in data collection. It was difficult to obtain actual financial reports from SMEs owners, and the current economic crisis also hindered the information from being up-to-date. Also, it was difficult to integrate the reporting systems of different institutions, but these limitations were overcome by using replicated data and reliable sources.

3.5 System Development Methodology

The development of the SME Loan Readiness Tool followed an iterative system development approach, where design, implementation, and testing were carried out in repeated cycles until the core features became stable. The system was implemented as a layered web application consisting of a React-based frontend, a Spring Boot backend, an AI processing layer, and a PostgreSQL database. This multi-layer structure allowed the project to clearly separate presentation, business logic, data processing, and storage components while improving maintainability and scalability.

The frontend was developed using React and Tailwind CSS to provide SME owners and bank officers with a simple and responsive interface. Through this interface, users can log in, enter financial information, complete the questionnaire, view analysis results, and download the bank-ready report. All user actions are sent to the backend through RESTful API calls in JSON format. The backend, built using Spring Boot, hosts the main business logic of the system. It manages authentication, SME profile handling, questionnaire storage, financial ratio calculations, readiness scoring, and report generation. Controller classes process incoming requests, while service classes apply the business rules described in the methodology section.

System authentication is implemented using JWT (JSON Web Token). When a user enters their login credentials, the backend validates the information and returns a signed token that is stored by the frontend and attached to all future API requests. This secure and stateless communication method prevents unauthorized access. The login workflow is illustrated in Figure 2, which shows the sequence of interactions between the user interface, authentication controller, and token service.

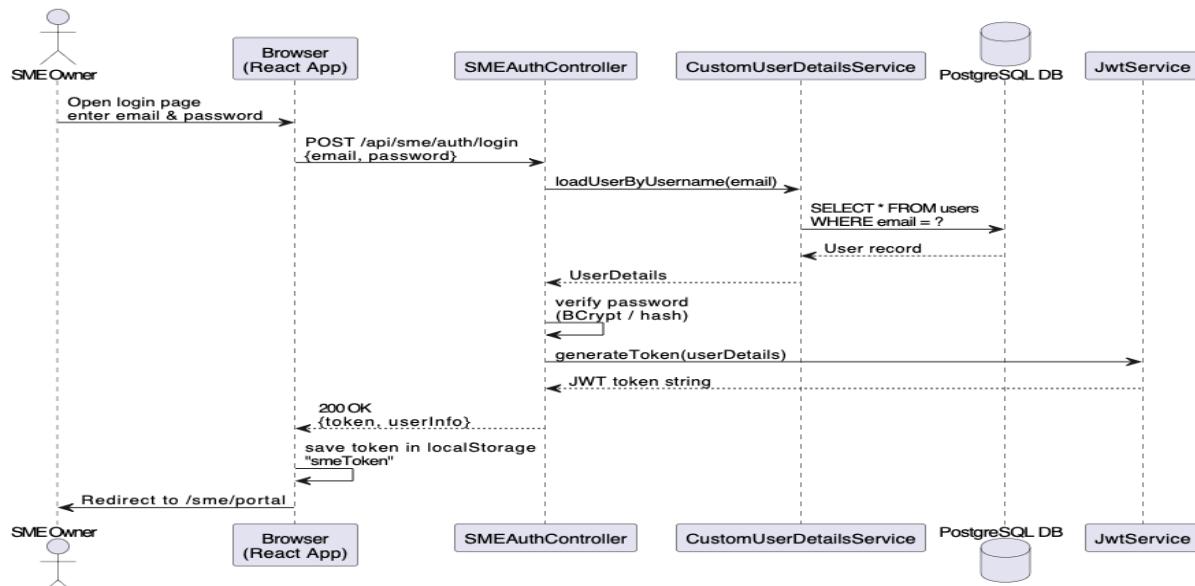


Figure 2 : User login credential

After authentication, SME users can access the main dashboard, which displays financial ratios, readiness scores, and AI-generated insights. To generate these outputs, the frontend triggers backend analysis endpoints that compute ratios and retrieve previously stored values from the PostgreSQL database. For narrative insights and business recommendations, the backend forwards structured prompts to an external Ollama AI server. The flow of dashboard loading

and AI processing is shown in Figure 3. This integration allows the system to provide users with both numerical results and meaningful explanations.

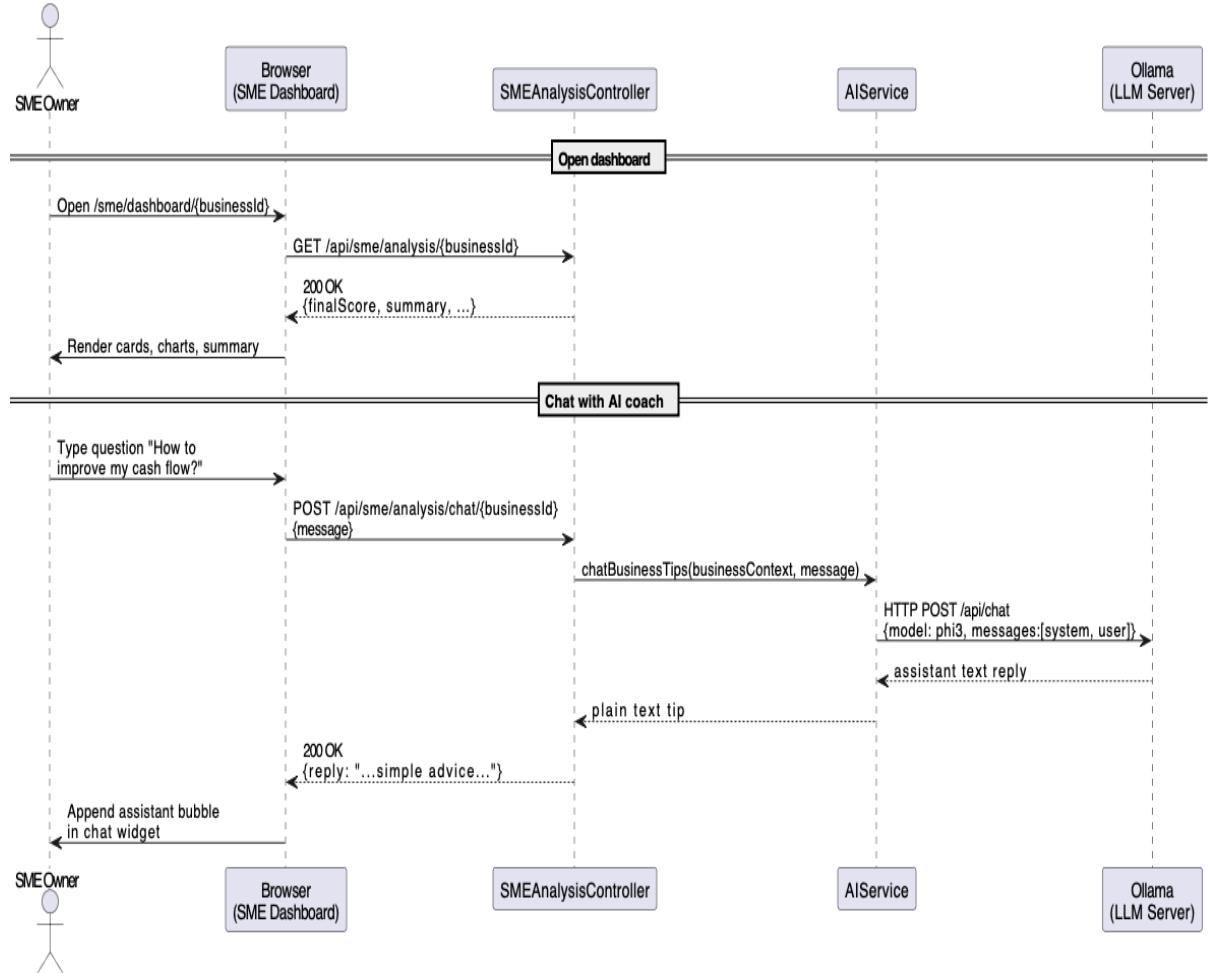


Figure 3 : AI-generated insights sequence diagram

All dataset elements—including SME profiles, questionnaire responses, financial ratios, scores, and AI insights—are stored in a normalized PostgreSQL database to ensure data integrity and efficient retrieval. Throughout the development process, each module was tested using Postman API calls, simulated SME data, and manual interface walkthroughs. Errors, inconsistencies, and workflow delays identified during testing were corrected in successive iterations.

Through this methodology, the conceptual design of the SME Loan Readiness Tool was successfully transformed into an operational web application that supports financial analysis, AI-driven insights, and secure user access.

3.6 Data Analysis Techniques

Data analysis is the process of processing data collected during research and obtaining accurate results. This is the basic basis for providing conclusions necessary to fulfill the research objective and making financial decisions in the system (Creswell, 2014). In this research, data analysis methods were used to calculate ratios and readiness scores to measure credit readiness for SMEs.

Before the analysis, it was necessary to prepare the data entered into the database. Incomplete data, null values, and abnormal values were identified and revised, and it was imported into the PostgreSQL database in CSV/Excel format. This process ensured consistency and data accuracy (Saunders et al., 2019).

Three basic analysis methods were used in the system — Profitability, Liquidity, and Leverage Analysis. ROA (Net Profit / Total Assets) and ROE (Net Profit / Equity) were used for profitability analysis. Current Ratio and Quick Ratio were used to measure Liquidity, while Debt-to-Equity and Interest Coverage Ratio were used to analyze Leverage. All these ratios were calculated automatically by the system algorithm.

A special innovation of the system was the calculation of Loan Readiness Score (LRS). In this, all ratio values were normalized and prepared as an index between 0 and 1. Then the following formula was used:

$$\text{LRS} = (0.4 \times \text{Profitability}) + (0.3 \times \text{Liquidity}) + (0.2 \times \text{Leverage}) + (0.1 \times \text{Stability Factor})$$

This weighting system was determined based on research literature and banking analysis (Berger & Udell, 1998; Central Bank, 2022). High LRS values indicate a business's high ability to obtain credit, while values below 0.5 are considered risky.

The technical tools used for data analysis included PostgreSQL SQL queries, Spring Boot algorithm modules, React dashboard visualization, and Excel statistical tools. These tools made it easy to analyze the data, calculate the readiness score, and display the results.

A validation process was performed to ensure the accuracy of the analysis. The ratio results were compared with bank standards and sample SMEs data to check the accuracy of the algorithm. For example, the system was able to show that there is liquidity risk if the Current Ratio < 1 . This validation process confirmed the reliability of the readiness score calculation (World Bank, 2020).

In summary, the data analysis method in this research was able to analyze the financial condition of SMEs and show it numerically as a readiness score. By combining the Profitability, Liquidity, and Leverage ratios, it was possible to create an accurate loan assessment model. This method is a FinTech-based scientific analysis, and its reliability and testing process are described in the next section, 3.7 Validation and Testing.

3.7 Validation and Testing

After designing the system, it is essential to test whether it works correctly, calculates correctly, and is easy for users to use. The Validation and Testing process ensured the accuracy of the “SME Loan Readiness Tool” system’s calculations, data matching, UI usability, and output. Being a FinTech-type application, these tests looked at both technical and analytical aspects.

Thirty main tests were used in the technical testing of the system:

- Unit Testing : The functions of the Spring Boot backend were tested separately to see if the ratio calculation, login, and data validation modules were working properly. Calculation errors, invalid inputs, and exception handling were tested using JUnit.
- Integration Testing : The backend–frontend connection was tested to see if it was simple. The React frontend → API calls → PostgreSQL database flow was checked to see if it was working properly. Status codes, response time, and error handling were verified using Postman
- UI Testing : The user view was tested to see if it was simple, navigation was smooth, and error messages were delivered correctly. Mobile/desktop responsiveness was tested and UI alignment issues were corrected.

All three tests confirmed system compatibility and performance stability.

Analytical validation and pilot user feedback testing were conducted simultaneously to test the analytical accuracy and user-friendliness of the system. First, financial ratios were compared with manual calculations to verify whether the system correctly calculates profitability, liquidity, and leverage, and compared them with industry benchmark values to confirm accuracy. Hypothetical SME data sets were used to test whether the normalization (0–1 range) of the readiness score calculation, the weighting system, and the logic for combining the final score were well matched. Error testing was performed to check whether the system handled extreme situations—negative profit, weak liquidity, high leverage—correctly. On the other hand, pilot user feedback included a user simulation of SMEs and checked usability using data entry, score visuals, risk indicators, and one-click report generation. Users rated the navigation simplicity, clear visuals, and UI responsiveness as good, and also provided some simple suggestions for refining the UI. Taken together, the two confirmed that the system was reliable in terms of both system accuracy and user-friendliness.

There were several challenges during the validation phase:

- Real SME datasets were mostly synthetic data due to privacy issues.
- It was difficult to set a general benchmark because ratio thresholds were different in different industries.
- Algorithm calibration required continuous refinement.

However, these challenges were addressed step by step using an iterative development model.

This validation and testing process confirmed that the “SME Loan Readiness Tool” system is reliable, accurate, and user-friendly from both technical and analytical perspectives.

3.8 Ethical Considerations

Ethics review is an essential part of ensuring the credibility of a research study. Failure to follow the correct legal principles in data collection, system development, and presentation of results can compromise the quality of the research and user safety. Therefore, in this research, issues such as data privacy, transparency, and user protection were taken into account.

All data used in this research was managed in a secure manner. For system testing, completely simulated financial statements were prepared without using real SMEs' business data. Therefore, there was no risk of any business's real data being leaked. Database access restrictions, password-protected accounts, JWT authentication, role-based access control (admin/user), and secure storage mechanisms were used. This clearly protected the confidentiality and trust of the research data.

Since the system processes financial data, data security measures were a primary requirement. The system was protected by encrypting the backend–database connection, secure HTTPS API calls, JWT verification, input validation, and detecting unauthorized access attempts. Sensitive fields in the database (e.g. asset values, profit data) were managed under secure hashing and a limited exposure policy. Error logs were systematically recorded to identify system vulnerabilities. All of this led to a secure environment that minimized risks such as data breaches, hacking attempts, SQL injection, or data manipulation.

All stages of the research were conducted in accordance with scientific integrity. No irregularities such as data fabrication, result manipulation, and plagiarism were observed. The algorithm, scoring logic, weighting model, and data sources were clearly presented and transparency was ensured.

When using the future SMEs system, protecting the rights of users is considered a key issue. The system is designed to provide user consent mechanisms, clear privacy notices, and user data delete/modify options. Users are given clear information about how their data is entered into the system, how it is used, and when it can be deleted. All of this is built on GDPR-like privacy principles, so the principles of user privacy, control, and transparency are unwaveringly maintained.

All of the above steps ensured that the ethical standards of this research were followed correctly. Data confidentiality, security, transparency, and user protection were properly taken into account, making the “Loan Readiness Tool” a responsible, reliable, and educational/practical FinTech solution.

3.9 Summary

This chapter fully describes the research methodology. The main aspects of the research, including the models used to conduct the research, data collection methods, system development process, data analysis, testing, and discussion, are explained step by step. In order to create the “SME Loan Readiness Tool” system, which was the research objective, it was clearly pointed out in this chapter that it is useful to use a system development-based applied methodology.

In the initial phase of the research, the system objectives, user needs, and key variables were identified through requirements gathering and research design. After that, both secondary data and simulated data were systematically collected and prepared into a database-ready format through data collection methodology. In the system development phase, the backend, frontend, and database architecture were created, algorithm development, and system integration were systematically performed using the Iterative SDLC model.

The data analysis process automatically calculated Profitability, Liquidity, and Leverage ratios important for SMEs, normalization, weighting application, and readiness score calculation. A number of validation procedures — ratio verification, scoring model checking, benchmark comparison, unit testing, and integration testing — were implemented to ensure the accuracy of these results. This confirmed both system functionality and analytical accuracy.

The final stage explained that ethical considerations were followed and that provisions for data confidentiality, secure processing, transparency, and user rights protection were followed. Both secondary data and simulated data used in the research were processed in accordance with legal procedures and protected by secure encryptions in the database. Backend-level security protocols were implemented to minimize risks such as unauthorized access, data tampering, and privacy breaches when processing financial data through the system.

In addition, system transparency and algorithm explanation have also been prioritized. Supportive explanations have been added to the UI to clearly explain to the user the procedure for calculating the readiness score. Allowing the user to delete/modify their data is a safeguard that is compatible with GDPR-like privacy rights. This not only ensures ethical compliance of the research, but also ensures that the rights of future SME users are protected when using the system. All of this enhances the scientific quality of the research and the value of the system as a responsible FinTech solution.

Accordingly, Chapter 3 fully describes the methodology used to develop, test, and validate the “SME Loan Readiness Tool”. The next chapter, Chapter 4 – Results and Implementation, describes the results obtained from this methodology and the final system performance.

Chapter 4 | Implementation and Results

4.1 Overview of System Implementation and Evaluation

This chapter presents the results obtained from the implementation and evaluation of the SME Loan Readiness Tool. The objective of this chapter is to demonstrate how the developed system performs in practice, how its outputs support small and medium-sized enterprises (SMEs), and how the system addresses the research objectives outlined earlier. The evaluation is carried out through functional implementation results, interface-level outputs, and analytical insights generated by the system.

The prototype system was implemented as a web-based application consisting of SME-facing interfaces, bank-facing analytical views, and a central policy control module. The system integrates financial data input, questionnaire-based qualitative assessment, automated readiness scoring, and AI-generated insights. At the time of evaluation, approximately 90 percent of the core functionalities had been successfully implemented and tested. These include SME registration and authentication, business profile creation, questionnaire submission, AI-based analysis, dashboard visualization, and document guidance for bank meetings.

The evaluation approach adopted in this chapter is primarily qualitative and functional. Rather than focusing on numerical performance benchmarks alone, the system is evaluated based on usability, clarity of financial interpretation, relevance of AI-generated recommendations, and the effectiveness of communication between SMEs and financial institutions. Screenshots captured from the working prototype are used as visual evidence to demonstrate how system outputs are presented to different user groups.

To reflect real-world usage, the system evaluation is presented from multiple viewpoints. The SME viewpoint focuses on how business owners interact with the system to understand their loan readiness, strengths, weaknesses, and improvement actions. The bank viewpoint highlights how financial institutions can interpret SME readiness scores and profiles more efficiently using standardized outputs. In addition, the central policy viewpoint illustrates how administrative users can adjust macro-level financial parameters and generate AI-supported policy insights.

This chapter is organized into several sub-sections to ensure clarity and logical progression. Sections 4.2 to 4.4 present the system outputs from the SME, bank, and central policy perspectives respectively. Section 4.5 explains the role of artificial intelligence in supporting financial analysis and decision-making. Section 4.6 evaluates system usability and technical performance. Section 4.7 provides a comparative analysis between the proposed system and existing approaches found in related studies. Finally, Section 4.8 discusses the overall results and reflects on how effectively the system meets the research objectives.

4.2 SME Viewpoint: Loan Readiness Evaluation Interface

4.2.1 SME Dashboard and Readiness Insights

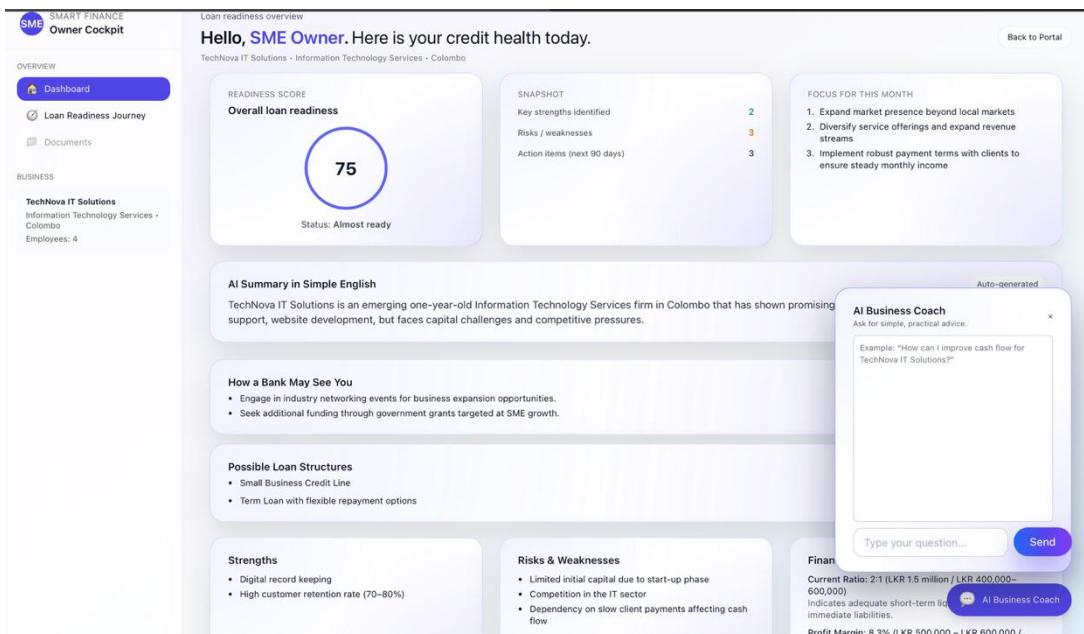


Figure 4 : SME Dashboard (Loan Readiness Overview)

Figure 4 illustrates the SME dashboard, which serves as the primary interface for business owners to assess their loan readiness status. The dashboard presents an overall readiness score, summarized financial indicators, identified strengths and weaknesses, and prioritized action items. The readiness score is displayed using a simple circular visualization accompanied by a qualitative status label such as “Almost Ready” or “Needs Improvement,” allowing SME owners to quickly understand their position without requiring advanced financial knowledge.

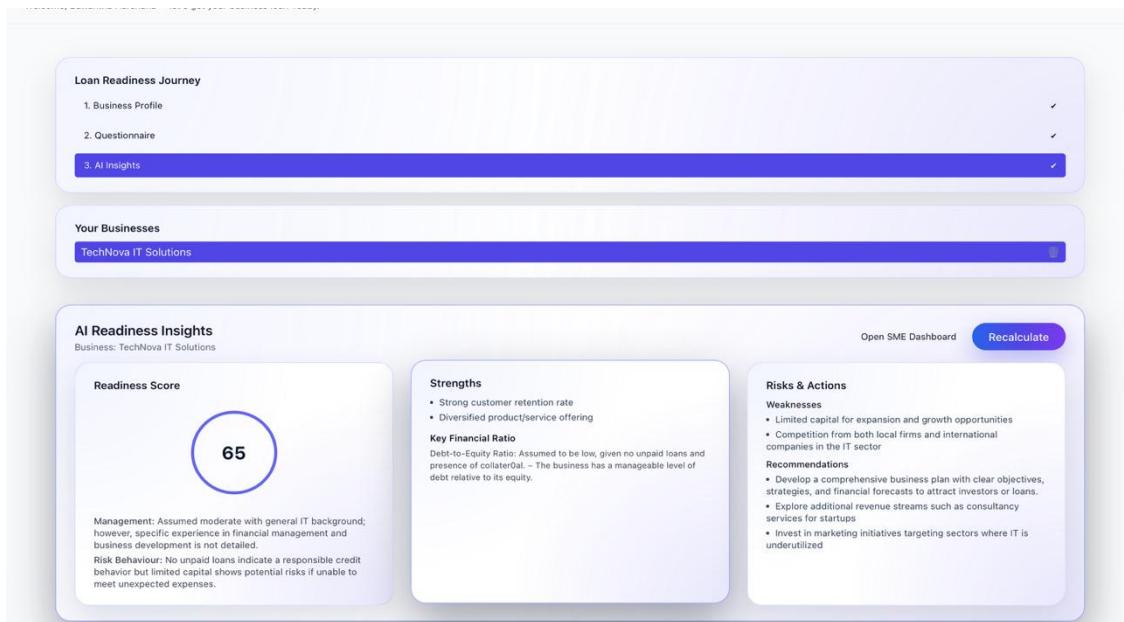


Figure 5 : Loan Readiness Journey

Figure 5 shows the AI-generated summary presented in simple language. This section translates financial ratios and questionnaire responses into clear explanations, helping SMEs understand how their business may be perceived by a bank. Additional panels such as “How a Bank May See You” and “Possible Loan Structures” further support SMEs in preparing for bank discussions by reducing uncertainty and information asymmetry.

4.2.2 SME Guide Book and User Support Features

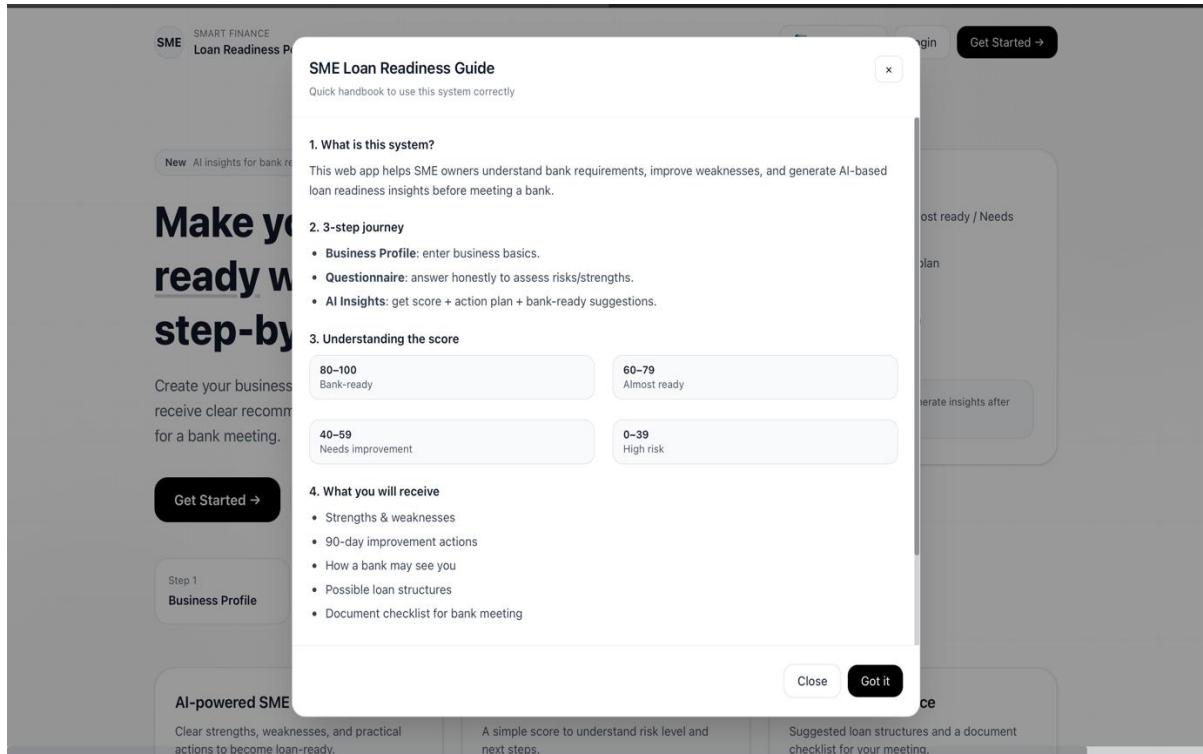


Figure 6 : SME Loan Readiness Guide Interface

Figure 6 presents the SME Loan Readiness Guide Book interface. This guide functions as an embedded user manual that explains the system purpose, step-by-step journey, readiness score interpretation, and expected outputs. The guide improves usability by supporting first-time users and reducing reliance on external assistance. By clearly explaining score ranges and recommended actions, the guide enhances financial literacy and encourages SMEs to engage more confidently with formal financial institutions.

4.3 Bank Viewpoint: SME Evaluation and Risk Interpretation

From the bank's perspective, the SME Loan Readiness Tool functions as a structured decision-support system that enhances the evaluation of small and medium-sized enterprises by reducing information asymmetry and standardizing risk assessment. Traditionally, bank officers rely on manually prepared financial statements, subjective business descriptions, and fragmented documentation when assessing SME loan applications. This often leads to inconsistent evaluations and increased processing time. The proposed system addresses these challenges by presenting SME financial and operational data in a clear, analytical, and comparable format that aligns with banking requirements.

The bank dashboard enables officers to view SME profiles, readiness scores, financial indicators, and AI-generated interpretations in a single interface. Instead of reviewing raw financial figures alone, the system translates complex data into meaningful risk signals, such as liquidity strength, capital adequacy, repayment capacity, and business stability. The readiness score acts as a preliminary screening metric, allowing banks to quickly identify SMEs that are near loan-ready, require conditional approval, or pose higher credit risk. This structured approach supports faster and more objective decision-making while maintaining credit discipline.

An important feature from the bank viewpoint is the system's ability to interpret SME behavior and financial patterns over time. By analyzing questionnaire responses and financial ratios together, the system highlights potential red flags such as cash-flow instability, over-reliance on short-term borrowing, or weak financial controls. These insights assist bank officers in conducting deeper risk analysis without requiring extensive manual calculations. As a result, banks can focus their attention on qualitative judgment and relationship management rather than repetitive technical assessments.

The AI-generated explanations further support the bank's evaluation process by providing consistent interpretations of SME data. While final lending decisions remain with human officers, the AI insights act as a secondary analytical layer that enhances transparency and reduces individual bias. This is particularly valuable when assessing early-stage SMEs that may lack long financial histories but demonstrate strong operational potential. The system enables banks to balance quantitative risk measures with contextual business understanding.

Overall, the SME Loan Readiness Tool strengthens the bank's ability to evaluate SMEs in a standardized, efficient, and risk-aware manner. By improving data clarity and offering structured interpretations, the system supports better credit decisions, reduces loan processing time, and enhances trust between banks and SMEs. From a banking perspective, the platform serves not only as an assessment tool but also as a mechanism to improve portfolio quality and promote responsible SME lending.

4.4 Central Policy and Administrative Viewpoint

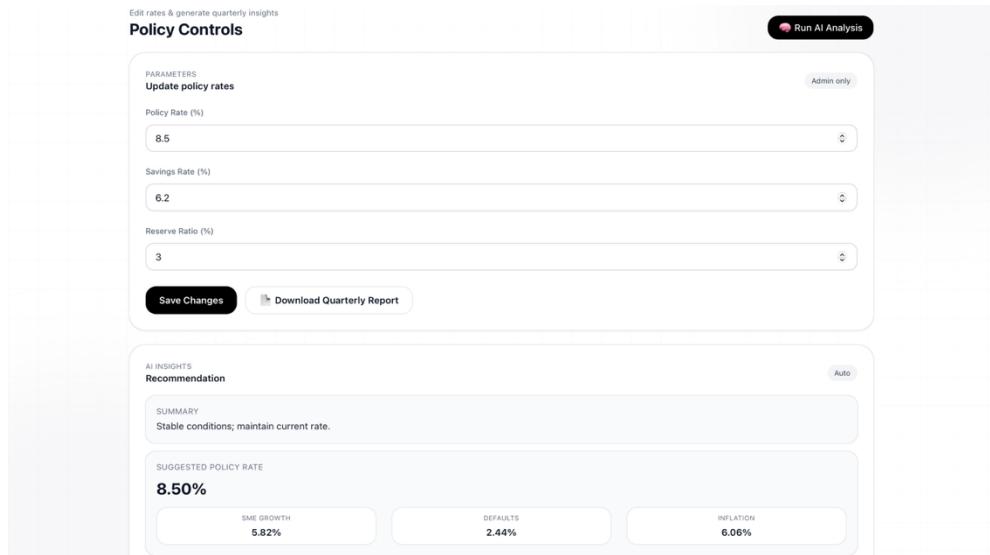


Figure 7 :Central bank Policy control

From the central policy and administrative perspective, the SME Loan Readiness Tool extends beyond individual SME evaluation and supports higher-level financial oversight and policy-oriented decision-making. Figure 7 illustrates the policy control interface designed for authorized administrative users, such as bank administrators or policy analysts, to adjust macroeconomic parameters and observe their potential impact on SME lending conditions. This viewpoint reflects how the system can support data-driven financial governance rather than focusing solely on individual loan assessments.

The policy control interface allows administrators to define key financial parameters, including the policy interest rate, savings rate, and reserve ratio. These variables are critical in shaping lending behavior, credit availability, and overall SME growth. By enabling controlled updates to these parameters, the system provides a simplified simulation environment where policy adjustments can be reflected in downstream AI analyses and reports. This feature supports alignment between institutional policy objectives and operational credit assessment practices.

In addition, the integration of AI-generated policy recommendations enhances the administrative decision-making process. Based on aggregated SME data, default trends, inflation indicators, and growth signals, the system produces high-level insights such as suggested policy rates and economic stability assessments. These recommendations do not replace human judgment but act as analytical support, enabling administrators to quickly interpret complex datasets and identify emerging risks or opportunities within the SME sector.

The availability of automated quarterly reports further strengthens this viewpoint by offering structured summaries that can be used for internal reviews, regulatory reporting, or strategic planning. By combining policy controls, AI insights, and reporting tools within a single interface, the system demonstrates how digital platforms can bridge micro-level SME data with macro-level financial oversight. Overall, the central policy and administrative viewpoint highlights the system's potential contribution to informed policy formulation, consistent credit governance, and sustainable SME financing.

4.5 AI-Driven Analysis and Decision Support

A key contribution of the SME Loan Readiness Tool is the integration of artificial intelligence to enhance financial analysis and decision support for both SMEs and financial institutions. Traditional loan assessment processes rely heavily on manual evaluation of financial statements and expert judgment, which can be time-consuming and inconsistent, particularly in the context of small and medium-sized enterprises. The AI-driven components of the proposed system address these limitations by automating analysis, interpreting results, and providing structured recommendations based on real business data.

At the SME level, the AI engine processes financial inputs and questionnaire responses to generate a comprehensive readiness score, supported by clearly defined strengths, weaknesses, and improvement actions. This automated interpretation transforms raw financial ratios and operational indicators into meaningful insights that are easily understandable by SME owners with limited financial expertise. Instead of presenting complex accounting figures alone, the system explains performance trends in simple language, enabling business owners to identify gaps in liquidity, profitability, and risk management. This approach significantly improves financial awareness and supports more informed decision-making at the enterprise level.

The system also incorporates an AI-generated action plan, typically structured around a short-term improvement horizon such as 90 days. These recommendations are tailored to the SME's specific financial profile and operational context, offering practical steps such as improving cash flow discipline, adjusting pricing strategies, or strengthening documentation practices. By linking analysis directly to actionable guidance, the system moves beyond diagnostic assessment and supports continuous improvement in loan readiness.

From a decision support perspective, the AI component contributes to reducing information asymmetry between SMEs and banks. The system generates standardized analytical outputs that reflect how financial institutions typically evaluate credit applications. As a result, SMEs gain visibility into how their business may be perceived by lenders, while banks benefit from more consistent and structured SME profiles. This alignment improves communication, reduces misunderstandings, and can potentially shorten loan evaluation timelines.

In addition, the AI-driven analysis supports scalability and consistency. Unlike manual advisory services, the system can deliver uniform analytical quality across a large number of SMEs without increasing operational costs. This is particularly valuable in developing economies, where access to financial advisory services is often limited. Overall, the AI-driven analysis and decision support features play a central role in enhancing the effectiveness of the SME Loan Readiness Tool by combining automation, interpretability, and practical guidance within a single digital platform.

4.6 System Usability and Technical Performance Evaluation

The usability and technical performance of the SME Loan Readiness Tool were evaluated to assess whether the system effectively supports its intended users while maintaining stability, responsiveness, and scalability. Since the primary users of the system include SME owners with varying levels of digital and financial literacy, usability was considered a critical success factor alongside technical correctness.

From a usability perspective, the system follows a simple, step-by-step interaction model that guides users through the loan readiness journey. SME owners begin by creating an account, entering basic business information, completing a structured questionnaire, and finally reviewing AI-generated insights. This sequential design reduces cognitive load and minimizes user errors by ensuring that users complete prerequisite steps before moving forward. Clear labels, minimal input requirements, and concise explanations were used throughout the interface to support ease of use, particularly for non-technical users. Feedback indicators such as readiness scores, progress summaries, and action plans further improve user understanding and engagement.

The dashboard design also contributes positively to usability by presenting key information in a visually organized manner. Important indicators such as readiness score, strengths, risks, and recommended actions are displayed prominently, allowing users to quickly grasp their financial position without navigating complex menus. The inclusion of explanatory text and tooltips enhances clarity, helping users interpret analytical results correctly. Overall, the interface design supports the research objective of providing a user-friendly and accessible loan readiness assessment tool.

In terms of technical performance, the system demonstrated stable behavior during testing. The Spring Boot backend efficiently handles authentication, data processing, financial calculations, and AI integration through clearly defined service layers and RESTful APIs. The use of JWT-based authentication ensures secure access control while maintaining stateless session management, which improves scalability. Database operations using PostgreSQL were observed to be reliable, with consistent data storage and retrieval for SME profiles, questionnaire responses, and analysis outputs.

The integration with the external AI engine via the Ollama server also performed reliably. AI requests and responses were processed within acceptable response times, and structured JSON outputs ensured seamless integration with the backend services. While AI response time may vary depending on system load, the overall performance remained suitable for the intended use case, as real-time responses are not critical for loan readiness analysis.

Overall, the evaluation indicates that the SME Loan Readiness Tool achieves a balance between usability and technical performance. The system is intuitive for end users, technically stable, and capable of supporting future enhancements. These results confirm that the platform is suitable for practical deployment and aligns well with the functional and non-functional requirements defined in the project objectives.

4.7 Comparative Analysis with Existing Systems

To evaluate the contribution and novelty of the proposed SME Loan Readiness Tool, a comparative analysis was conducted against existing SME financial assessment and loan support systems reported in prior research and practice. Many existing systems focus on isolated aspects of SME finance, such as credit scoring, accounting automation, or bank-side risk evaluation, but lack an integrated, user-centric approach that supports both SMEs and financial institutions simultaneously.

Traditional credit assessment systems used by banks primarily rely on historical financial statements, collateral availability, and predefined risk models. While these systems are effective from a bank's risk management perspective, they offer limited transparency to SME owners. SMEs are often unaware of how their financial data is interpreted, which contributes to information asymmetry and unsuccessful loan applications. In contrast, the proposed system explicitly addresses this gap by presenting readiness scores, strengths, weaknesses, and actionable recommendations in a simplified format tailored for SME users.

Several digital SME support platforms discussed in literature provide financial ratio analysis and basic dashboards. However, these platforms typically assume a certain level of financial literacy and do not offer guided explanations or improvement pathways. The SME Loan Readiness Tool enhances usability by combining quantitative analysis with AI-generated qualitative insights, allowing users to understand not only *what* their financial position is, but also *why* it matters and *how* it can be improved.

From a functionality perspective, most existing systems focus either on the SME side or the bank side, but not both. The proposed system is distinct in that it supports multiple viewpoints: SME owners, bank officers, and central or administrative users. This multi-view architecture enables banks to review standardized SME profiles and risk indicators, while SMEs receive preparation guidance before formal loan evaluation. Such alignment improves communication efficiency and reduces processing delays.

Another key difference is the integration of AI-driven decision support. While some advanced systems use machine learning for credit scoring, they often function as black-box models. In contrast, the proposed tool uses AI to generate explainable insights, recommendations, and coaching messages that improve financial literacy rather than replacing human judgment. This approach increases trust and acceptance among users.

Overall, the comparative analysis demonstrates that the SME Loan Readiness Tool extends beyond existing solutions by integrating financial analysis, AI-based guidance, usability-focused design, and multi-stakeholder perspectives within a single platform. These features collectively enhance accessibility, transparency, and practical value for SMEs seeking access to formal credit.

Table 1: Comparative Analysis of SME Loan Readiness Tool with Existing Systems

Aspect	Existing SME Finance Platforms	Proposed SME Loan Readiness Tool	Salon Madushitha (Sri Lanka)	HCL Tech
Primary Focus	Financial data visualization	SME readiness + bank evaluation	Daily operations & cash survival	Growth, scalability, and investor readiness
SME Guidance	Limited	AI-driven, step-by-step guidance	No structured guidance	Internal financial planning teams
Financial Literacy Support	Moderate	High (simple explanations)	Low – depends on accountant	High – finance-aware management
AI Integration	Limited	Explainable AI insights	None	Advanced analytics & forecasting
Multi-View Support	SME only	SME, Bank, Central/Admin	SME owner only	Management, investors, banks
Transparency	Moderate	High	Low – unclear financial position	High – real-time dashboards
Actionable Recommendations	Limited	Yes (90-day action plan)	No clear roadmap	Strategic planning & KPIs
Deployment Context	Standalone tools	Integrated web platform	Paper records, Excel, WhatsApp	Cloud-based systems (ERP, dashboards)

4.8 Discussion of Results

The results presented in this chapter demonstrate that the SME Loan Readiness Tool effectively addresses the key challenges identified in the problem statement and meets the primary objectives of the research. The system successfully integrates financial analysis, artificial intelligence, and user-centered design to support SMEs in evaluating their loan readiness while also providing structured outputs that align with banking and policy-level requirements. The discussion below interprets the results from multiple stakeholder perspectives and highlights the practical significance of the system.

From the SME perspective, the results show a clear improvement in financial understanding and preparedness. The readiness score, dashboard indicators, and AI-generated explanations transform complex financial concepts into accessible insights. Many SMEs struggle with interpreting financial ratios and understanding how banks assess creditworthiness. By presenting strengths, risks, and improvement actions in simple language, the system empowers SME owners to identify gaps in liquidity management, debt structure, and documentation practices. This directly contributes to improved financial literacy and supports more informed business decision-making prior to approaching financial institutions.

From the banking perspective, the system demonstrates strong potential to reduce information asymmetry and improve efficiency in the early stages of loan evaluation. The standardized readiness score and structured summaries allow banks to perform preliminary screening more consistently. While the system does not replace internal credit scoring models or regulatory compliance procedures, it complements existing processes by improving data quality and transparency. The AI-generated interpretations further support banks by highlighting contextual risk factors that may not be immediately visible through numerical analysis alone.

At the administrative and policy level, the results indicate that the system can support broader financial oversight and evidence-based decision-making. The central policy interface and AI-assisted recommendations demonstrate how aggregated SME data can be used to monitor sector-level trends and assess the potential impact of policy adjustments. This multi-level capability distinguishes the proposed system from many existing SME finance tools, which typically focus on either micro-level or macro-level analysis, but not both.

The comparative analysis in Section 4.7 further reinforces these findings by showing that the proposed system offers a more integrated and transparent solution compared to traditional bank processes and existing digital platforms. The combination of explainable AI, multi-stakeholder viewpoints, and usability-focused design contributes to the system's overall effectiveness and relevance.

Overall, the discussion confirms that the SME Loan Readiness Tool provides meaningful practical value across stakeholders. The results support the conclusion that a digital, AI-driven readiness assessment platform can improve SME access to credit, enhance trust between SMEs and banks, and contribute to more inclusive and sustainable financial systems, particularly in developing economies such as Sri Lanka.

Chapter 5 | Conclusion and Future Work

Chapter 5 | Conclusion and Future Work

5.1 Conclusion

This study focused on addressing a critical challenge faced by small and medium-sized enterprises (SMEs) in Sri Lanka, namely the difficulty of accessing formal credit due to inadequate financial preparedness and information asymmetry between SMEs and financial institutions. To address this issue, the research proposed, designed, and implemented a web-based SME Loan Readiness Tool that integrates financial analysis, artificial intelligence, and standardized reporting mechanisms.

The findings of the study demonstrate that the developed system successfully meets its primary objectives. The SME Loan Readiness Tool provides SMEs with a structured platform to assess their financial readiness for loans through key financial ratio analysis, readiness scoring, AI-generated insights, and bank-ready reports. By translating complex financial indicators into understandable visual and textual explanations, the system improves financial literacy among SME owners and enables them to better understand their business performance before approaching financial institutions.

From the SME perspective, the system reduces dependence on external accountants for basic financial interpretation and empowers business owners to make informed decisions regarding loan applications. The readiness score and AI-generated recommendations offer clear guidance on whether an SME is prepared to apply for credit or requires further financial improvement. This proactive approach helps SMEs identify weaknesses early and take corrective action, thereby improving their chances of loan approval.

From the banking perspective, the tool indirectly supports financial institutions by improving the quality and consistency of loan applications received from SMEs. The standardized financial outputs and structured reports reduce information gaps, support faster preliminary assessments, and contribute to more consistent credit evaluation processes. Although the system does not replace traditional credit appraisal models, it complements existing banking practices by enhancing early-stage screening and reducing documentation-related inefficiencies.

At a broader level, the inclusion of a central policy and Smart Finance perspective demonstrates the potential of the system beyond individual loan preparation. The ability to analyze aggregated SME data, simulate policy variables, and generate AI-assisted recommendations highlights how digital tools can support regulatory oversight and evidence-based policymaking. This multi-stakeholder approach strengthens the relevance of the system within the Sri Lankan financial ecosystem.

Overall, the study confirms that a digital and AI-driven loan readiness platform can play a valuable role in strengthening SME financial preparedness, improving transparency, and supporting access to credit. The SME Loan Readiness Tool represents a practical and scalable solution that aligns with national SME development objectives and modern financial technology trends.

5.2 Future Work

While the SME Loan Readiness Tool demonstrates strong functionality and practical value, several opportunities exist for future enhancement and further research. One key area for future work is the integration of the system with real-world banking systems and financial institutions. Establishing secure APIs with commercial banks or credit bureaus would allow real-time verification of financial data and further improve the accuracy of readiness assessments.

Another potential enhancement is the expansion of the AI component. Future versions of the system could incorporate sector-specific financial models, enabling more tailored recommendations for different industries such as manufacturing, retail, agriculture, and services. Additionally, machine learning models could be trained using historical SME loan performance data to improve risk prediction and readiness scoring accuracy.

The system could also be extended to support mobile platforms, allowing SME owners to access loan readiness insights through smartphones. This would improve accessibility, particularly for SMEs operating in rural or underserved regions. Multilingual support could further enhance usability by allowing users to interact with the system in their preferred language.

From a policy and research perspective, future studies could evaluate the system using larger and more diverse SME datasets to assess its impact on loan approval rates and business performance over time. Longitudinal studies could provide deeper insights into how financial readiness tools influence SME growth, survival, and contribution to economic development.

In conclusion, while this research successfully delivers a functional and innovative loan readiness solution, future enhancements can further strengthen its impact. Continued development and research can transform the SME Loan Readiness Tool into a comprehensive digital finance platform that supports SMEs, banks, and policymakers in building a more inclusive and resilient financial ecosystem in Sri Lanka.

REFERENCE

- Abor, J., & Quartey, P. (2010). Issues in SME development in Ghana and South Africa. *International Research Journal of Finance and Economics*, 39, 218–228.
- Asian Development Bank. (2021). *Asia SME finance monitor 2021*. ADB.
- Asian Development Bank Institute. (2020). *Financing SMEs in developing Asia*. ADBI.
- Ayyagari, M., Beck, T., & Demirguc-Kunt, A. (2007). Small and medium enterprises across the globe. *Small Business Economics*, 29(4), 415–434. <https://doi.org/10.1007/s11187-006-9002-5>
- Beck, T., & Demirguc-Kunt, A. (2006). SMEs: Access to finance as a growth constraint. *Journal of Banking & Finance*, 30(11), 2931–2943.
- Beck, T., & Levine, R. (2004). SMEs, growth, and financial development. *Journal of Economic Growth*, 10(3), 199–229.
- Berger, A. N., & Udell, G. F. (1998). The economics of small business finance. *Journal of Banking & Finance*, 22(6–8), 613–673.
- Central Bank of Sri Lanka. (2022). *Annual report 2022*. CBSL.
- Demirguc-Kunt, A., Martínez Pería, M. S., & Tressel, T. (2015). The impact of crises on firms' capital structure. *World Bank Economic Review*, 29(1), 1–33.
- Gamage, A. S. (2013). SME development in Sri Lanka. *International Journal of Business and Social Science*, 4(6), 180–191.
- Government of Sri Lanka. (2020). *National policy framework for SME development*. Ministry of Industries.
- International Finance Corporation. (2017). *MSME finance gap*. IFC.
- International Finance Corporation & World Bank. (2021). *Financing small businesses*. World Bank Group.
- OECD. (2019). *SME and entrepreneurship outlook 2019*. OECD Publishing.
- OECD. (2020). *Digital disruption in banking and finance*. OECD Publishing.
- Rahman, M., & Rahaman, M. (2014). Access to finance and firm growth. *International Journal of Economics and Finance*, 6(6), 172–185.
- Schiffer, M., & Weder, B. (2001). Firm size and the business environment. *World Bank Economic Review*, 15(1), 1–22.
- World Bank. (2018). *Financial inclusion strategies*. World Bank Group.
- World Bank. (2020). *SME finance*. World Bank Group.
- World Bank. (2021). *Financial consumer protection and SME finance*. World Bank Group.
- Berg, T., Burg, V., Gombović, A., & Puri, M. (2020). On the rise of FinTechs. *Journal of Finance*, 75(5), 2485–252.

Appendix A (Code for the SME Loan Readiness Tool)

1. Backend Code.

1.1 AdminBankController

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/



package com.example.smartfinancedashboard.admin;

import com.example.smartfinancedashboard.model.Bank;
import com.example.smartfinancedashboard.repository.BankRepository;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.*;
import java.util.*;


@RestController
@RequestMapping("/api/admin/banks")
@PreAuthorize("hasAnyRole('ADMIN', 'BANK_ADMIN')")
public class AdminBankController {

    private final BankRepository repo;

    public AdminBankController(BankRepository repo) {
        this.repo = repo;
    }

    @GetMapping
    public List<Bank> listBanks() {
        return repo.findAll();
    }

    @PostMapping
    public Bank addBank(@RequestBody Bank b) {
        return repo.save(b);
    }

    @PutMapping("/{id}/toggle")
    public Map<String, Object> toggleBank(@PathVariable Long id) {
        var bank = repo.findById(id).orElseThrow();

        //  use isActive() for Boolean fields
        boolean current = Boolean.TRUE.equals(bank.getActive());
        boolean newStatus = !current;
        bank.setActive(newStatus);

        repo.save(bank);

        return Map.of(
            "status", "updated",
            "active", newStatus
        );
    }
}
```

```
}
```

1.2 AuthController

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
```

```
package com.example.smartfinancedashboard.auth;
```

```
import com.example.smartfinancedashboard.model.User;
import com.example.smartfinancedashboard.repository.UserRepository;
import com.example.smartfinancedashboard.security.*;
import jakarta.validation.Valid;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.web.bind.annotation.*;
```

```
import java.util.Map;
```

```
import static com.example.smartfinancedashboard.auth.AuthDtos.*;
```

```
@RestController
@RequestMapping("/api/auth")
public class AuthController {
```

```
    private final UserRepository users;
    private final BCryptPasswordEncoder encoder;
    private final JwtService jwt;
```

```
    public AuthController(UserRepository users, BCryptPasswordEncoder encoder, JwtService jwt) {
        this.users = users;
        this.encoder = encoder;
        this.jwt = jwt;
    }
```

```
    /**
     * Register New User (Admin / SME)
     */
    @PostMapping("/register")
    public ResponseEntity<?> register(@RequestBody @Valid RegisterRequest req) {
        if (users.existsByEmail(req.getEmail())) {
            return ResponseEntity.badRequest().body(Map.of("error", "Email already registered"));
        }

        // Create new user
        User user = User.builder()
            .name(req.name())
            .email(req.email())
            .passwordHash(encoder.encode(req.password()))
            .role("SME_USER") //  Default role for SME registrations
            .build();

        users.save(user);
    }
}
```

```

// ✅ Generate JWT token (new method)
String token = jwt.generateToken(user.getEmail(), user.getRole());

return ResponseEntity.ok(Map.of(
    "message", "Registration successful",
    "token", token,
    "email", user.getEmail(),
    "name", user.getName(),
    "role", user.getRole()
));
}

/*
=====
🔒 Login Existing User
=====
*/
@PostMapping("/login")
public ResponseEntity<?> login(@RequestBody @Valid LoginRequest req) {
    var user = users.findByEmail(req.email()).orElse(null);
    if (user == null || !encoder.matches(req.password(), user.getPasswordHash())) {
        return ResponseEntity.status(401).body(Map.of("error", "Invalid credentials"));
    }

    // ✅ Generate new token using updated JwtService
    String token = jwt.generateToken(user.getEmail(), user.getRole());

    return ResponseEntity.ok(Map.of(
        "token", token,
        "email", user.getEmail(),
        "name", user.getName(),
        "role", user.getRole()
    ));
}

/*
=====
👤 Verify Login (for testing)
=====
*/
@GetMapping("/me")
public ResponseEntity<?> me(@RequestHeader("Authorization") String authHeader) {
    String token = authHeader.replace("Bearer ", "").trim();
    String email = jwt.extractEmail(token);
    String role = jwt.extractRole(token);

    return ResponseEntity.ok(Map.of(
        "email", email,
        "role", role,
        "valid", jwt.isTokenValid(token, email)
    ));
}
}

```

1.3 AuthDtos

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.auth;

import jakarta.validation.constraints.Email;
import jakarta.validation.constraints.NotBlank;

public class AuthDtos {
    public record LoginRequest(@Email String email, @NotBlank String password) {}
    public record RegisterRequest(@NotBlank String name, @Email String email,
                                   @NotBlank String password) {}
    public record AuthResponse(String token, String role, String name, String email) {}
}

```

1.4 Role

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```

```

package com.example.smartfinancedashboard.auth;

public enum Role {
    CENTRAL_ADMIN, // Central Bank main admin
    BANK_ADMIN, // Head admin for each bank
    BANK_MANAGER, // Senior loan officer
    BANK_OFFICER, // Regular loan officer
    SME // Small and Medium Enterprise user
}

```

1.5 CorsConfig

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```

```

package com.example.smartfinancedashboard.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
public class CorsConfig {

```

```

@Bean
public WebMvcConfigurer corsConfigurer() {
    return new WebMvcConfigurer() {
        @Override
        public void addCorsMappings(CorsRegistry registry) {
            registry.addMapping("/**")
                .allowedOrigins("http://localhost:5173") // ✅ your frontend
                .allowedMethods("GET", "POST", "PUT", "DELETE", "OPTIONS")
                .allowedHeaders("*")
                .allowCredentials(true);
        }
    };
}

```

1.6 SMEAnalysisController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```

```

package com.example.smartfinancedashboard.controller.sme;

import com.example.smartfinancedashboard.model.sme.SMEAnalysis;
import com.example.smartfinancedashboard.model.sme.SMEBusiness;
import com.example.smartfinancedashboard.model.sme.SMEQuestionnaire;
import com.example.smartfinancedashboard.repository.sme.SMEAnalysisRepository;
import com.example.smartfinancedashboard.repository.sme.SMEBusinessRepository;
import com.example.smartfinancedashboard.repository.sme.SMEQuestionnaireRepository;
import com.example.smartfinancedashboard.service.AIService;
import com.fasterxml.jackson.databind.JsonNode;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.time.LocalDateTime;
import java.util.List;

@Slf4j
@RestController
@RequestMapping("/api/sme/analysis")
@RequiredArgsConstructor
@CrossOrigin(origins = "http://localhost:5173", allowCredentials = "true")
public class SMEAnalysisController {

    private final SMEBusinessRepository smeBusinessRepository;
    private final SMEQuestionnaireRepository smeQuestionnaireRepository;
    private final SMEAnalysisRepository smeAnalysisRepository;
    private final AIService aiService;

    // POST /api/sme/analysis/generate/{businessId}
    @PostMapping("/generate/{businessId}")
    public ResponseEntity<?> generateAnalysis(@PathVariable Long businessId) {

```

```

try {
    log.info("generateAnalysis for businessId={}", businessId);

    SMEBusiness business = smeBusinessRepository.findById(businessId)
        .orElse(null);

    if (business == null) {
        return ResponseEntity.notFound().build();
    }

    List<SMEQuestionnaire> answers =
        smeQuestionnaireRepository.findByBusinessId(businessId);

    JsonNode ai = aiService.analyzeBusiness(business, answers);

    SMEAnalysis analysis = SMEAnalysis.builder()
        .business(business)
        .createdAt(LocalDateTime.now())
        .summary(ai.path("summary").asText(""))
        .strengths(toJson(ai.get("strengths")))
        .weaknesses(toJson(ai.get("weaknesses")))
        .improvements(toJson(ai.get("improvements")))
        .bankSuggestions(toJson(ai.get("bank_suggestions")))
        .loanPlans(toJson(ai.get("loan_plans")))
        .financialRatios(toJson(ai.get("financial_ratios")))
        .managementAssessment(toJson(ai.get("management_assessment")))
        .documentChecklist(toJson(ai.get("document_checklist")))
        .finalScore(ai.path("final_score").isNumber() ? ai.get("final_score").asInt() : 0)
        .rawJson(ai.toString())
        .build();

    SMEAnalysis saved = smeAnalysisRepository.save(analysis);
    return ResponseEntity.ok(saved);

} catch (Exception e) {
    log.error("Error generating analysis", e);
    return ResponseEntity.status(500)
        .body("AI analysis failed. Please ensure the AI server is running or try again later.");
}
}

//  GET /api/sme/analysis/{businessId} (this matches your Network call)
@GetMapping("/{businessId}")
public ResponseEntity<?> getLatestAnalysis(@PathVariable Long businessId) {
    return smeAnalysisRepository.findTopByBusiness_IdOrderByCreatedAtDesc(businessId)
        .<ResponseEntity<?>>map(ResponseEntity::ok)
        .orElseGet(() -> ResponseEntity.notFound().build());
}

private String toJson(JsonNode node) {
    if (node == null || node.isNull() || node.isMissingNode()) return "[]";
    return node.toString();
}

```

1.7 SMEAuthController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara

```

Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```
package com.example.smartfinancedashboard.controller.sme;

import com.example.smartfinancedashboard.model.sme.SMEUser;
import com.example.smartfinancedashboard.repository.sme.SMEUserRepository;
import com.example.smartfinancedashboard.security.JwtService;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.web.bind.annotation.*;

import java.util.Map;

@RestController
@RequestMapping("/api/sme/auth")
public class SMEAuthController {

    private final SMEUserRepository smeUsers;
    private final BCryptPasswordEncoder encoder;
    private final JwtService jwtService;

    public SMEAuthController(SMEUserRepository smeUsers,
                            BCryptPasswordEncoder encoder,
                            JwtService jwtService) {
        this.smeUsers = smeUsers;
        this.encoder = encoder; // ✅ FIXED
        this.jwtService = jwtService;
    }

    @PostMapping("/register")
    public ResponseEntity<?> register(@RequestBody Map<String, String> req) {
        String email = req.get("email");
        String password = req.get("password");
        String name = req.get("name");

        if (smeUsers.existsByEmail(email)) {
            return ResponseEntity.badRequest().body(Map.of("error", "Email already registered"));
        }

        // Save new SME user
        SMEUser user = SMEUser.builder()
            .email(email)
            .name(name)
            .passwordHash(encoder.encode(password)) // ✅ FIXED
            .role("SME")
            .build();

        smeUsers.save(user);

        // Generate token
        String token = jwtService.generateToken(email, "SME");

        return ResponseEntity.ok(Map.of(
            "token", token,
            "email", user.getEmail(),
            "name", user.getName(),
        ));
    }
}
```

```

        "role", user.getRole()
    ));
}

@PostMapping("/login")
public ResponseEntity<?> login(@RequestBody Map<String, String> req) {
    String email = req.get("email");
    String password = req.get("password");

    var user = smeUsers.findByEmail(email).orElse(null);

    if (user == null || !encoder.matches(password, user.getPasswordHash())) { // ✅ FIXED
        return ResponseEntity.status(401).body(Map.of("error", "Invalid credentials"));
    }

    String token = jwtService.generateToken(email, "SME");

    return ResponseEntity.ok(Map.of(
        "token", token,
        "email", user.getEmail(),
        "name", user.getName(),
        "role", user.getRole()
    ));
}
}

```

1.8 SMEBusinessController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```

```

package com.example.smartfinancedashboard.controller.sme;

import com.example.smartfinancedashboard.model.sme.SMEBusiness;
import com.example.smartfinancedashboard.model.sme.SMEUser;
import com.example.smartfinancedashboard.repository.sme.SMEBusinessRepository;
import com.example.smartfinancedashboard.repository.sme.SMEUserRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.server.ResponseStatusException;

import java.time.LocalDateTime;
import java.util.List;
import java.util.Map;

@RestController
@RequestMapping("/api/sme/business")
@RequiredArgsConstructor
public class SMEBusinessController {

    private final SMEBusinessRepository businessRepository;

```

```

private final SMEUserRepository smeUserRepository;

// ----- helper: get currently logged-in SME from JWT -----
private SMEUser getCurrentSme(Authentication auth) {
    if (auth == null || auth.getName() == null) {
        throw new ResponseStatusException(HttpStatus.UNAUTHORIZED, "No authenticated SME user");
    }

    String email = auth.getName();
    return smeUserRepository.findByEmail(email)
        .orElseThrow(() -> new ResponseStatusException(
            HttpStatus.UNAUTHORIZED,
            "SME user not found for email: " + email
        ));
}

@DeleteMapping("/{id}")
public ResponseEntity<?> deleteBusiness(@PathVariable Long id,
                                         Authentication auth) {

    SMEUser current = getCurrentSme(auth);

    return businessRepository.findById(id)
        .map(biz -> {
            // make sure this business really belongs to this SME
            if (!biz.getSmeUser().getId().equals(current.getId())) {
                return ResponseEntity.status(HttpStatus.FORBIDDEN)
                    .body(Map.of("error", "You cannot delete this business"));
            }

            businessRepository.delete(biz);
            return ResponseEntity.ok(Map.of("status", "deleted"));
        })
        .orElseGet(() -> ResponseEntity.status(HttpStatus.NOT_FOUND)
            .body(Map.of("error", "Business not found")));
}

/* ===== LIST BUSINESSES (CURRENT SME ONLY)
=====
*/
@GetMapping
public ResponseEntity<List<SMEBusiness>> list(Authentication auth) {
    SMEUser current = getCurrentSme(auth);

    // 🔑 uses your existing repository method
    List<SMEBusiness> businesses = businessRepository.findBySmeUserId(current.getId());

    return ResponseEntity.ok(businesses);
}

/* ===== GET ONE BUSINESS (OWNERSHIP CHECK)
=====
*/
@GetMapping("/{id}")
public ResponseEntity<SMEBusiness> getOne(@PathVariable Long id, Authentication auth) {
    SMEUser current = getCurrentSme(auth);

    SMEBusiness business = businessRepository.findById(id)
        .filter(b -> b.getSmeUser() != null && b.getSmeUser().getId().equals(current.getId()))
        .orElseThrow(() -> new ResponseStatusException(HttpStatus.NOT_FOUND, "Business not found"));

    return ResponseEntity.ok(business);
}

```

```

/* ===== CREATE NEW BUSINESS FOR CURRENT SME ===== */
@PostMapping
public ResponseEntity<?> create(@RequestBody Map<String, Object> body,
                                 Authentication auth) {

    SMEUser current = getCurrentSme(auth);

    try {
        String businessName = (String) body.get("businessName");
        String industry = (String) body.get("industry");
        String location = (String) body.get("location");
        String bankRelationship = (String) body.getOrDefault("bankRelationship", "");

        // employees → Integer
        Number employeesNum = (Number) body.getOrDefault("employees", 0);
        int employees = employeesNum != null ? employeesNum.intValue() : 0;

        // annualRevenue → Double
        Number revenueNum = (Number) body.getOrDefault("annualRevenue", 0);
        Double annualRevenue =
            revenueNum != null ? revenueNum.doubleValue() : 0.0;

        if (businessName == null || businessName.isBlank()
            || industry == null || industry.isBlank()
            || location == null || location.isBlank()) {
            return ResponseEntity.badRequest().body(
                Map.of("error", "Business name, industry and location are required")
            );
        }

        SMEBusiness business = new SMEBusiness();
        business.setBusinessName(businessName);
        business.setIndustry(industry);
        business.setLocation(location);
        business.setBankRelationship(bankRelationship);
        business.setEmployees(employees);
        business.setAnnualRevenue(annualRevenue); // ✅ FIXED – now Double
        business.setRegisteredAt(LocalDateTime.now());
        business.setSmeUser(current);

        SMEBusiness saved = businessRepository.save(business);

        return ResponseEntity.ok(saved);
    } catch (Exception e) {
        e.printStackTrace();
        return ResponseEntity.status(HttpStatus.BAD_REQUEST)
            .body(Map.of("error", "Invalid business data"));
    }
}
}

```

1.9 SMEChatController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179

```

University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```
package com.example.smartfinancedashboard.controller.sme;

import com.example.smartfinancedashboard.model.sme.SMEBusiness;
import com.example.smartfinancedashboard.repository.sme.SMEAAnalysisRepository;
import com.example.smartfinancedashboard.repository.sme.SMEBusinessRepository;
import com.example.smartfinancedashboard.service.AIService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.Map;

@RestController
@RequestMapping("/api/sme/analysis/chat")
@RequiredArgsConstructor
public class SMEChatController {

    private final SMEBusinessRepository businessRepo;
    private final SMEAnalysisRepository analysisRepo;
    private final AIService aiService;

    @PostMapping("/{businessId}")
    public ResponseEntity<?> chat(
        @PathVariable Long businessId,
        @RequestBody Map<String, String> body
    ) {
        String message = body.getOrDefault("message", "").trim();
        if (message.isEmpty()) {
            return ResponseEntity.badRequest().body(Map.of("error", "Message is empty"));
        }

        SMEBusiness business = businessRepo.findById(businessId).orElse(null);
        if (business == null) {
            return ResponseEntity.status(404).body(Map.of("error", "Business not found"));
        }

        StringBuilder ctx = new StringBuilder();
        ctx.append("Business name: ").append(business.getBusinessName()).append("\n")
            .append("Industry: ").append(business.getIndustry()).append("\n")
            .append("Employees: ").append(business.getEmployees()).append("\n")
            .append("Annual revenue (LKR): ").append(business.getAnnualRevenue()).append("\n")
            .append("Location: ").append(business.getLocation()).append("\n")
            .append("Relationship with bank: ").append(
                business.getBankRelationship() == null ? "not specified" : business.getBankRelationship()
            ).append("\n");

        analysisRepo.findTopByBusiness_IdOrderByCreatedAtDesc(businessId).ifPresent(a -> {
            ctx.append("\nLatest AI readiness summary: ")
                .append(a.getSummary() == null ? "" : a.getSummary())
                .append("\nFinal score: ").append(a.getFinalScore());
        });

        String reply = aiService.chatBusinessTips(ctx.toString(), message);
        return ResponseEntity.ok(Map.of("reply", reply));
    }
}
```

```
    }  
}
```

1.10 SMEProfileController

```
/*  
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara  
Student ID: M24W0179  
University: The Kyoto College of Graduate Studies for Informatics  
Project: SME Loan Readiness Tool  
Date: 2026-01-23  
*/  
  
package com.example.smartfinancedashboard.controller.sme;  
  
import com.example.smartfinancedashboard.model.sme.SMEUser;  
import com.example.smartfinancedashboard.repository.sme.SMEUserRepository;  
import com.example.smartfinancedashboard.security.JwtService;  
import lombok.RequiredArgsConstructor;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.*;  
  
import java.util.Map;  
  
@RestController  
@RequestMapping("/api/sme/profile")  
@RequiredArgsConstructor  
public class SMEProfileController {  
  
    private final SMEUserRepository smeUsers;  
  
    private final JwtService jwtService;  
  
    @GetMapping("/me")  
    public ResponseEntity<?> getMyProfile(@RequestHeader("Authorization") String authHeader) {  
        String email = jwtService.extractEmail(authHeader.replace("Bearer ", "").trim());  
  
        SMEUser user = smeUsers.findByEmail(email)  
            .orElseThrow(() -> new RuntimeException("User not found"));  
  
        return ResponseEntity.ok(Map.of(  
            "id", user.getId(),  
            "name", user.getName(),  
            "email", user.getEmail(),  
            "role", user.getRole()  
        ));  
    }  
  
    @PutMapping("/update")  
    public ResponseEntity<?> updateProfile(@RequestBody Map<String, String> req,  
                                           @RequestHeader("Authorization") String authHeader) {  
  
        String email = jwtService.extractEmail(authHeader.replace("Bearer ", "").trim());  
        SMEUser user = smeUsers.findByEmail(email).orElseThrow();  
  
        if (req.containsKey("name")) user.setName(req.get("name"));  
        smeUsers.save(user);  
    }  
}
```

```

        return ResponseEntity.ok(Map.of("message", "Profile updated successfully"));
    }
}

```

1.11 SMEQuestionnaireController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.controller.sme;

import com.example.smartfinancedashboard.model.sme.SMEQuestionnaire;
import com.example.smartfinancedashboard.model.sme.SMEBusiness;
import com.example.smartfinancedashboard.repository.sme.SMEQuestionnaireRepository;
import com.example.smartfinancedashboard.repository.sme.SMEBusinessRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.http.ResponseEntity;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.*;

import java.util.*;

@RestController
@RequestMapping("/api/sme/questionnaire")
@RequiredArgsConstructor
@PreAuthorize("hasRole('SME')")
public class SMEQuestionnaireController {

    private final SMEQuestionnaireRepository questionnaireRepo;
    private final SMEBusinessRepository businessRepo;

    private static final List<String> QUESTIONS = List.of(
        "How long has your business been operating?",
        "What is your average monthly cash flow?",
        "Do you maintain digital financial records?",
        "Do you have unpaid loans?",
        "What are your key products/services?",
        "What is your customer retention rate?",
        "What challenges limit your business growth?",
        "Do you have a business plan?",
        "Do you have collateral?",
        "What is your expected funding requirement?",
        "How is your monthly sales trend?",
        "What marketing channels do you use?",
        "How do you manage suppliers?",
        "What is your operational cost structure?",
        "Do you use accounting software?"
    );
    @GetMapping("/questions")

```

```

public ResponseEntity<?> getQuestions() {
    return ResponseEntity.ok(QUESTIONS);
}

{@PostMapping("/submit/{businessId}")
public ResponseEntity<?> submitAnswers(@PathVariable Long businessId,
                                         @RequestBody List<String> answers) {

    SMEBusiness business = businessRepo.findById(businessId)
        .orElseThrow(() -> new RuntimeException("Business not found"));

    for (int i = 0; i < answers.size(); i++) {
        questionnaireRepo.save(SMEQuestionnaire.builder()
            .business(business)
            .questionNo(i + 1)
            .questionText(QUESTIONS.get(i))
            .answer(answers.get(i))
            .build());
    }

    return ResponseEntity.ok(Map.of("message", "Answers submitted successfully"));
}

{@GetMapping("/answers/{businessId}")
public ResponseEntity<?> getAnswers(@PathVariable Long businessId) {
    return ResponseEntity.ok(questionnaireRepo.findByBusinessId(businessId));
}

{@GetMapping("/completed/{businessId}")
public ResponseEntity<?> isCompleted(@PathVariable Long businessId) {
    boolean exists = questionnaireRepo.existsByBusiness_Id(businessId);
    return ResponseEntity.ok(Map.of("completed", exists));
}
// -----
// Check if a business has already answered the questionnaire
// GET /api/sme/questionnaire/status/{businessId}
// returns: { "completed": true/false }
// -----
{@GetMapping("/status/{businessId}")
public ResponseEntity<?> getStatus(@PathVariable Long businessId) {

    boolean completed = questionnaireRepo.existsByBusiness_Id(businessId);
    return ResponseEntity.ok(Map.of("completed", completed));
}
}

```

1.12 AdminPolicyController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```

```
package com.example.smartfinancedashboard.controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
```

```

import com.example.smartfinancedashboard.model.PolicySettings;
import com.example.smartfinancedashboard.repository.PolicySettingsRepository;

import java.util.List;

@RestController
@RequestMapping("/api/admin/policy")
@CrossOrigin(origins = "http://localhost:5173") // Allow frontend (Vite) access
public class AdminPolicyController {

    @Autowired
    private PolicySettingsRepository repo;

    // ✅ GET latest policy
    @GetMapping
    public PolicySettings getLatest() {
        List<PolicySettings> all = repo.findAll();
        if (all.isEmpty()) {
            return null;
        }
        return all.get(all.size() - 1);
    }

    // ✅ POST - update or create a new policy
    @PostMapping
    public PolicySettings updatePolicy(@RequestBody PolicySettings newPolicy) {
        return repo.save(newPolicy);
    }
}

```

1.13 BankAdminController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.controller;

import org.springframework.security.access.prepost.PreAuthorize;
import com.example.smartfinancedashboard.model.Bank;
import com.example.smartfinancedashboard.model.User;
import com.example.smartfinancedashboard.model.LoanApplication;
import com.example.smartfinancedashboard.repository.BankRepository;
import com.example.smartfinancedashboard.repository.UserRepository;
import com.example.smartfinancedashboard.repository.LoanApplicationRepository;
import com.example.smartfinancedashboard.repository.BranchRepository;
import com.example.smartfinancedashboard.security.JwtService;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.web.bind.annotation.*;
import java.util.*;

@RestController
@RequestMapping("/api/bank-admin")
public class BankAdminController {

```

```

private final BankRepository banks;
private final UserRepository users;
private final BranchRepository branches;
private final LoanApplicationRepository loans;
private final BCryptPasswordEncoder encoder;
private final JwtService jwtService;

public BankAdminController(
    BankRepository banks,
    UserRepository users,
    BranchRepository branches,
    LoanApplicationRepository loans,
    BCryptPasswordEncoder encoder,
    JwtService jwtService
) {
    this.banks = banks;
    this.users = users;
    this.branches = branches;
    this.loans = loans;
    this.encoder = encoder;
    this.jwtService = jwtService;
}

// =====
// 🏛 Get logged-in bank admin info
// =====

@PreAuthorize("hasAnyRole('ADMIN', 'BANK ADMIN', 'BANK MANAGER')")
@GetMapping("/me")
public ResponseEntity<?> getMyBank(@RequestHeader("Authorization") String authHeader) {
    try {
        String token = authHeader.replace("Bearer ", "").trim();
        String email = jwtService.extractEmail(token);

        if (email == null) {
            return ResponseEntity.status(401).body(Map.of("error", "Token parse failed"));
        }

        var user = users.findByEmail(email)
            .orElseThrow(() -> new RuntimeException("User not found"));

        var bank = banks.findById(user.getBankId())
            .orElseThrow(() -> new RuntimeException("Bank not found"));

        // Theme mapping
        Map<String, Object> theme = switch (bank.getName()) {
            case "Commercial Bank" -> Map.of("themeKey", "commercial",
                "viewKey", "commercialView",
                "primary", "#0044CC",
                "secondary", "#FFFFFF",
                "accent", "#002F6C");
            case "Sampath Bank" -> Map.of("themeKey", "sampath",
                "viewKey", "sampathView",
                "primary", "#FF6F00",
                "secondary", "#FFFFFF",
                "accent", "#F57C00");
            case "Hatton National Bank" -> Map.of("themeKey", "hnb",
                "viewKey", "hnbView",
                "primary", "#0056A6",
                "secondary", "#FFD700",
                "accent", "#003B73");
        };
    }
}

```

```

        case "People's Bank" -> Map.of("themeKey", "peoples",
            "viewKey", "peoplesView",
            "primary", "#B50000",
            "secondary", "#FFD700",
            "accent", "#E60000");
        case "Bank of Ceylon" -> Map.of("themeKey", "boc",
            "viewKey", "bocView",
            "primary", "#FFD700",
            "secondary", "#003366",
            "accent", "#FFCC00");
        default -> Map.of(
            "themeKey", "default",
            "viewKey", "defaultView",
            "primary", "#0F172A",
            "secondary", "#1E293B",
            "accent", "#3B82F6");
    };
}

long branchCount = branches.countByBankId(bank.getId());
long totalLoans = loans.countByBankId(bank.getId());
long pendingLoans = loans.countByBankIdAndStatus(bank.getId(), "PENDING");

Map<String, Object> response = new HashMap<>();
response.put("bankName", bank.getName());
response.put("bankCode", bank.getCode());
response.put("role", user.getRole());
response.put("theme", theme);
response.put("branchCount", branchCount);
response.put("totalLoans", totalLoans);
response.put("pendingLoans", pendingLoans);

if ("BANK_MANAGER".equals(user.getRole())) {
    response.put("branchId", user.getBranchId());
}

return ResponseEntity.ok(response);

} catch (Exception e) {
    e.printStackTrace();
    return ResponseEntity.status(401).body(Map.of("error", e.getMessage()));
}
}

// =====
// 🏛 Get all branches of the logged-in bank
// =====

@GetMapping("/branches")
public ResponseEntity<?> getBranches(@RequestHeader("Authorization") String authHeader) {
    String token = authHeader.replace("Bearer ", "").trim();
    String email = jwtService.extractEmail(token);

    if (email == null)
        return ResponseEntity.status(401).body(Map.of("error", "Token parse failed"));

    User admin = users.findByEmail(email)
        .orElseThrow(() -> new RuntimeException("User not found"));

    var branchList = branches.findByBankId(admin.getBankId()).stream().map(branch -> {
        var manager = branch.getManager();

```

```

        Map<String, Object> map = new HashMap<>();
        map.put("id", branch.getId());
        map.put("name", branch.getName());
        map.put("code", branch.getCode());
        map.put("location", branch.getLocation());

        if (manager != null) {
            map.put("manager", Map.of(
                "name", manager.getName(),
                "email", manager.getEmail()
            ));
        } else {
            map.put("manager", null);
        }

        return map;
    }).toList();

    return ResponseEntity.ok(branchList);
}

// =====
// 🚧 Add new employee
// =====
@PostMapping("/employees")
public ResponseEntity<?> addEmployee(@RequestBody Map<String, String> req) {
    try {
        Long bankId = Long.parseLong(req.get("bankId"));
        String name = req.get("name");
        String email = req.get("email");
        String password = req.get("password");
        String role = req.get("role");

        if (users.findByEmail(email).isPresent()) {
            return ResponseEntity.badRequest().body(Map.of("error", "Email already registered"));
        }

        User u = User.builder()
            .name(name)
            .email(email)
            .passwordHash(encoder.encode(password))
            .role(role)
            .bankId(bankId)
            .build();

        users.save(u);
        return ResponseEntity.ok(Map.of("message", "Employee added successfully"));
    } catch (Exception e) {
        return ResponseEntity.status(500).body(Map.of("error", e.getMessage()));
    }
}

// =====
// 🗑 Delete employee
// =====
@DeleteMapping("/employees/{id}")
public ResponseEntity<?> deleteEmployee(@PathVariable Long id) {
    if (!users.existsById(id))
        return ResponseEntity.status(404).body(Map.of("error", "Employee not found"));
    users.deleteById(id);
}

```

```

        return ResponseEntity.ok(Map.of("message", "Employee deleted successfully"));
    }

// =====
// 💰 Get loans for bank
// =====
@GetMapping("/loans/{bankId}")
public ResponseEntity<?> getLoans(@PathVariable Long bankId) {
    List<LoanApplication> bankLoans = loans.findByBankId(bankId);
    return ResponseEntity.ok(bankLoans);
}

// =====
// ✅ Approve/Reject Loan
// =====
@PutMapping("/loans/{loanId}")
public ResponseEntity<?> updateLoanStatus(@PathVariable Long loanId, @RequestBody Map<String, String> req) {
    return loans.findById(loanId).map(loan -> {
        loan.setStatus(req.get("status"));
        loans.save(loan);
        return ResponseEntity.ok(Map.of(
            "message", "Loan status updated",
            "newStatus", loan.getStatus()
        ));
    }).orElse(ResponseEntity.status(404).body(Map.of("error", "Loan not found")));
}
}

```

1.14 BranchController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.controller;

import com.example.smartfinancedashboard.model.*;
import com.example.smartfinancedashboard.repository.*;
import com.example.smartfinancedashboard.security.JwtService;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.web.bind.annotation.*;
import java.util.*;

@RestController
@RequestMapping("/api/branches") // ✅ CHANGED root path
public class BranchController {

    private final BranchRepository branches;
    private final UserRepository users;
    private final BankRepository banks;
    private final BCryptPasswordEncoder encoder;
    private final JwtService jwtService;

    public BranchController(BranchRepository branches, UserRepository users, BankRepository banks,

```

```

        BCryptPasswordEncoder encoder, JwtService jwtService) {
    this.branches = branches;
    this.users = users;
    this.banks = banks;
    this.encoder = encoder;
    this.jwtService = jwtService;
}

// 📱 Get all branches for current bank
@GetMapping("/list") // ✅ Changed path to /api/branches/list
public ResponseEntity<?> listBranches(@RequestHeader("Authorization") String authHeader) {
    String token = authHeader.replace("Bearer ", "").trim();
    String email = jwtService.extractEmail(token);
    var admin = users.findByEmail(email).orElseThrow();
    var list = branches.findById(admin.getBankId());
    return ResponseEntity.ok(list);
}

// ✚ Add a new branch
@PostMapping("/add") // ✅ Now at /api/branches/add
public ResponseEntity<?> addBranch(@RequestBody Map<String, String> req,
                                    @RequestHeader("Authorization") String authHeader) {
    String token = authHeader.replace("Bearer ", "").trim();
    String email = jwtService.extractEmail(token);
    var admin = users.findByEmail(email).orElseThrow();

    Branch branch = Branch.builder()
        .name(req.get("name"))
        .code(req.get("code"))
        .location(req.get("location"))
        .bank(banks.findById(admin.getBankId()).get())
        .build();

    branches.save(branch);
    return ResponseEntity.ok(Map.of("message", "Branch added successfully"));
}

// 💼 Add manager to branch
@PostMapping("/{branchId}/manager")
public ResponseEntity<?> addManager(@PathVariable Long branchId, @RequestBody Map<String, String> req) {
    var branch = branches.findById(branchId).orElseThrow();
    if (users.findByEmail(req.get("email")).isPresent())
        return ResponseEntity.badRequest().body(Map.of("error", "Email already exists"));

    User manager = User.builder()
        .name(req.get("name"))
        .email(req.get("email"))
        .passwordHash(encoder.encode(req.get("password")))
        .role("BANK_MANAGER")
        .bankId(branch.getBank().getId())
        .branchId(branchId)
        .build();

    users.save(manager);
    branch.setManager(manager);
    branches.save(branch);

    return ResponseEntity.ok(Map.of("message", "Manager added successfully"));
}

```

```

}

// ✗ Delete a branch
@GetMapping("/{branchId}")
public ResponseEntity<?> deleteBranch(@PathVariable Long branchId) {
    if (!branches.existsById(branchId))
        return ResponseEntity.status(404).body(Map.of("error", "Branch not found"));
    branches.deleteById(branchId);
    return ResponseEntity.ok(Map.of("message", "Branch deleted successfully"));
}

// 🗑 Remove Manager from branch
@GetMapping("/{branchId}/manager")
public ResponseEntity<?> removeManager(@PathVariable Long branchId) {
    var branch = branches.findById(branchId).orElseThrow();
    if (branch.getManager() == null) {
        return ResponseEntity.status(404).body(Map.of("error", "No manager assigned"));
    }

    users.deleteById(branch.getManager().getId());
    branch.setManager(null);
    branches.save(branch);

    return ResponseEntity.ok(Map.of("message", "Manager removed successfully"));
}
}

```

1.15 ManagerController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.controller;

import com.example.smartfinancedashboard.model.User;
import com.example.smartfinancedashboard.model.LoanApplication;
import com.example.smartfinancedashboard.model.Bank;
import com.example.smartfinancedashboard.model.Branch;
import com.example.smartfinancedashboard.repository.UserRepository;
import com.example.smartfinancedashboard.repository.BranchRepository;
import com.example.smartfinancedashboard.repository.BankRepository;
import com.example.smartfinancedashboard.repository.LoanApplicationRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.http.ResponseEntity;
import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.*;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@RestController
@RequestMapping("/api/manager")
@RequiredArgsConstructor
public class ManagerController {

```

```

private final UserRepository users;
private final BranchRepository branches;
private final BankRepository banks;
private final LoanApplicationRepository loans;

// ✅ Manager profile info with theme support
@GetMapping("/me")
public ResponseEntity<?> getManagerProfile(Authentication auth) {
    String email = (String) auth.getPrincipal();
    User manager = users.findByEmail(email).orElseThrow(() -> new RuntimeException("Manager not found"));

    Branch branch = branches.findById(manager.getBranchId()).orElse(null);
    Bank bank = banks.findById(manager.getBankId()).orElse(null);

    // ✅ Null-safe theme handling
    String themeKey = bank != null && bank.getThemeKey() != null ? bank.getThemeKey() : "default";
    String primary = bank != null && bank.getPrimaryColor() != null ? bank.getPrimaryColor() : "#0b132b";
    String secondary = bank != null && bank.getSecondaryColor() != null ? bank.getSecondaryColor() : "#ffffff";
    String accent = bank != null && bank.getAccentColor() != null ? bank.getAccentColor() : "#5bc0be";

    Map<String, Object> theme = new HashMap<>();
    theme.put("themeKey", themeKey);
    theme.put("primary", primary);
    theme.put("secondary", secondary);
    theme.put("accent", accent);

    Map<String, Object> response = new HashMap<>();
    response.put("name", manager.getName());
    response.put("email", manager.getEmail());
    response.put("role", manager.getRole());
    response.put("bankId", manager.getBankId());
    response.put("bankName", bank != null ? bank.getName() : "Unknown Bank");
    response.put("branchId", manager.getBranchId());
    response.put("branchName", branch != null ? branch.getName() : "Unknown Branch");
    response.put("theme", theme);

    return ResponseEntity.ok(response);
}

// ✅ Manager dashboard data
@GetMapping("/dashboard")
public ResponseEntity<?> getBranchDashboard(Authentication auth) {
    String email = (String) auth.getPrincipal();
    User manager = users.findByEmail(email)
        .orElseThrow(() -> new RuntimeException("Manager not found"));

    Long branchId = manager.getBranchId();

    // 📈 Fetch all loans for this branch
    List<LoanApplication> loansList = loans.findByBranch_Id(branchId);

    long totalLoans = loansList.size();
    long approved = loansList.stream().filter(l -> "Approved".equalsIgnoreCase(l.getStatus())).count();
    long pending = loansList.stream().filter(l -> "Pending".equalsIgnoreCase(l.getStatus())).count();
    long rejected = loansList.stream().filter(l -> "Rejected".equalsIgnoreCase(l.getStatus())).count();
}

```

```

// 👤 Fetch all officers under this branch
var officers = users.findByIdAndRole(branchId, "BANK_OFFICER");

// 💰 Calculate total and average loan amounts
double totalAmount = loansList.stream()
    .mapToDouble(l -> l.getAmount() != null ? l.getAmount() : 0)
    .sum();
double avgLoanAmount = totalLoans > 0 ? totalAmount / totalLoans : 0;

Map<String, Object> response = new HashMap<>();
response.put("branchId", branchId);
response.put("branchName", branches.findById(branchId).map/Branch::getName).orElse("Unknown"));
response.put("totalLoans", totalLoans);
response.put("approvedLoans", approved);
response.put("pendingLoans", pending);
response.put("rejectedLoans", rejected);
response.put("totalAmount", totalAmount);
response.put("averageLoanAmount", avgLoanAmount);
response.put("officerCount", officers.size());
response.put("officers", officers);

return ResponseEntity.ok(response);
}
}

```

1.16 PolicyAnalysisController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.controller;

import org.springframework.web.bind.annotation.*;
import org.springframework.http.ResponseEntity;
import java.time.LocalDate;
import java.util.Map;
import java.util.Random;

@RestController
@RequestMapping("/api/admin/analyze")
public class PolicyAnalysisController {

    @GetMapping
    public ResponseEntity<?> analyzePolicy() {
        // Simulate retrieved data — later we'll query DB
        double smeGrowth = new Random().nextDouble(5) + 1; // 1–6 %
        double loanDefaultRate = new Random().nextDouble(5); // 0–5 %
        double inflation = new Random().nextDouble(4) + 3; // 3–7 %

        // Rule-based “AI” logic
        double suggestedPolicyRate;
        String summary;

        if (smeGrowth < 2 && loanDefaultRate < 3) {
            suggestedPolicyRate = 8.5 + 0.25;
        }
    }
}

```

```

        summary = "Economy cooling; slight rate increase recommended.";
    } else if (smeGrowth > 4 && loanDefaultRate > 4) {
        suggestedPolicyRate = 8.5 - 0.25;
        summary = "High SME activity but defaults rising; ease policy slightly.";
    } else {
        suggestedPolicyRate = 8.5;
        summary = "Stable conditions; maintain current rate.";
    }

    Map<String, Object> result = Map.of(
        "timestamp", LocalDate.now(),
        "smeGrowth", smeGrowth,
        "loanDefaultRate", loanDefaultRate,
        "inflation", inflation,
        "currentPolicyRate", 8.5,
        "suggestedPolicyRate", suggestedPolicyRate,
        "analysisSummary", summary
    );
}

return ResponseEntity.ok(result);
}
}

```

1.17 ReportController

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.controller;

import com.example.smartfinancedashboard.model.PolicySettings;
import com.example.smartfinancedashboard.repository.BankRepository;
import com.example.smartfinancedashboard.repository.PolicyRepository;
import com.example.smartfinancedashboard.repository.UserRepository;
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;
import org.springframework.http.HttpHeaders;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.io.ByteArrayOutputStream;
import java.time.LocalDate;
import java.time.format.TextStyle;
import java.util.List;
import java.util.Locale;

@RestController
@RequestMapping("/api/admin/report")

```

```

public class ReportController {

    private final PolicyRepository policyRepo;
    private final BankRepository bankRepo;
    private final UserRepository userRepo;

    public ReportController(PolicyRepository policyRepo, BankRepository bankRepo, UserRepository userRepo) {
        this.policyRepo = policyRepo;
        this.bankRepo = bankRepo;
        this.userRepo = userRepo;
    }

    @GetMapping(value = "/quarterly", produces = MediaType.APPLICATION_PDF_VALUE)
    public ResponseEntity<byte[]> generateQuarterlyReport() {
        try {
            ByteArrayOutputStream out = new ByteArrayOutputStream();
            Document document = new Document();
            PdfWriter.getInstance(document, out);
            document.open();

            // Title Section
            Font titleFont = new Font(Font.FontFamily.HELVETICA, 20, Font.BOLD, BaseColor.DARK_GRAY);
            Paragraph title = new Paragraph("Smart Finance Quarterly Report", titleFont);
            title.setAlignment(Element.ALIGN_CENTER);
            title.setSpacingAfter(20);
            document.add(title);

            // Date + Quarter
            LocalDate today = LocalDate.now();
            int quarter = (today.getMonthValue() - 1) / 3 + 1;
            Paragraph dateInfo = new Paragraph(
                "Date: " + today.getMonth().getDisplayName(TextStyle.FULL, Locale.ENGLISH) + " " +
                today.getYear() +
                " | Quarter: Q" + quarter,
                new Font(Font.FontFamily.HELVETICA, 12, Font.NORMAL, BaseColor.GRAY)
            );
            dateInfo.setAlignment(Element.ALIGN_CENTER);
            dateInfo.setSpacingAfter(25);
            document.add(dateInfo);

            // Fetch data from DB
            List<PolicySettings> policies = policyRepo.findAll();
            PolicySettings p = policies.isEmpty() ? null : policies.get(0);
            long totalBanks = bankRepo.count();
            long totalUsers = userRepo.count();
            long smeUsers = userRepo.findAll().stream()
                .filter(u -> u.getRole() != null && u.getRole().contains("SME"))
                .count();

            // Data Table
            PdfPTable table = new PdfPTable(2);
            table.setWidthPercentage(80);
            table.setSpacingBefore(10f);
            table.setSpacingAfter(20f);

            addRow(table, "Policy Rate (%)", p != null ? String.valueOf(p.getPolicyRate()) : "N/A");
            addRow(table, "Savings Rate (%)", p != null ? String.valueOf(p.getSavingsRate()) : "N/A");
            addRow(table, "Reserve Ratio (%)", p != null ? String.valueOf(p.getReserveRatio()) : "N/A");
            addRow(table, "Registered Banks", String.valueOf(totalBanks));
        }
    }
}

```

```

        addRow(table, "Total SMEs Registered", String.valueOf(smeUsers));
        addRow(table, "System Users (All Roles)", String.valueOf(totalUsers));

        document.add(table);

        // Footer / Summary
        Paragraph summary = new Paragraph(
            "This report provides a consolidated view of policy and SME performance metrics for the current
quarter.\n" +
            "Generated automatically by the Smart Finance AI-based system.",
            new Font(Font.FontFamily.HELVETICA, 11, Font.ITALIC, BaseColor.GRAY)
        );
        summary.setAlignment(Element.ALIGN_CENTER);
        document.add(summary);

        document.close();

        // Return PDF response
        HttpHeaders headers = new HttpHeaders();
        headers.add("Content-Disposition", "inline; filename=Quarterly_Report.pdf");
        return ResponseEntity.ok()
            .headers(headers)
            .contentType(MediaType.APPLICATION_PDF)
            .body(out.toByteArray());

    } catch (Exception e) {
        e.printStackTrace();
        return ResponseEntity.internalServerError().build();
    }
}

private void addRow(PdfPTable table, String key, String value) {
    PdfPCell cell1 = new PdfPCell(new Phrase(key));
    PdfPCell cell2 = new PdfPCell(new Phrase(value));
    cell1.setBackgroundColor(BaseColor.LIGHT_GRAY);
    cell1.setPadding(8);
    cell2.setPadding(8);
    table.addCell(cell1);
    table.addCell(cell2);
}
}

```

1.18 SMEAnalysis

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model.sme;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDateTime;

@Entity

```

```

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Builder
@Table(name = "sme_analysis")
public class SMEAnalysis {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    // Which business this analysis belongs to
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(
        name = "business_id",
        referencedColumnName = "id",
        foreignKey = @ForeignKey(name = "fk_analysis_business")
    )
    @JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
    private SMEBusiness business;

    @Column(name = "created_at")
    private LocalDateTime createdAt;

    private LocalDateTime generatedAt;

    // --- MAIN SECTIONS (simple text or JSON-as-text) ---
    @Column(columnDefinition = "TEXT")
    private String summary;

    @Column(columnDefinition = "TEXT")
    private String strengths;

    @Column(columnDefinition = "TEXT")
    private String weaknesses;

    @Column(columnDefinition = "TEXT")
    private String improvements;

    @Column(name = "bankSuggestions", columnDefinition = "TEXT")
    private String bankSuggestions;

    @Column(name = "loanPlans", columnDefinition = "TEXT")
    private String loanPlans;

    // --- ADVANCED JSON SECTIONS (stored as TEXT) ---
    @Column(name = "financialRatios", columnDefinition = "TEXT")
    private String financialRatios;

    @Column(name = "managementAssessment", columnDefinition = "TEXT")
    private String managementAssessment;

    @Column(name = "documentChecklist", columnDefinition = "TEXT")
    private String documentChecklist;

    private Integer finalScore;

    @Column(name = "rawJson", columnDefinition = "TEXT")

```

```

    private String rawJson;
}

1.19 SMEBusiness

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model.sme;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDateTime;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})

@Entity
@Getter @Setter @Builder
@NoArgsConstructor @AllArgsConstructor
@Table(name = "sme_business")
public class SMEBusiness {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne
    @JoinColumn(name = "sme_user_id")
    private SMEUser smeUser;

    private String businessName;
    private String industry;
    private Integer employees;
    private Double annualRevenue;
    private String location;
    private String bankRelationship;

    private LocalDateTime registeredAt = LocalDateTime.now();
}

```

1.20 SMEQuestionnaire

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model.sme;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDateTime;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})

@Entity
@Getter @Setter @Builder
@NoArgsConstructor @AllArgsConstructor
@Table(name = "sme_questionnaire")
public class SMEQuestionnaire {

```

```

import java.time.LocalDateTime;

@Entity
@Builder
@NoArgsConstructor @AllArgsConstructor
@Table(name = "sme_questionnaire")
public class SMEQuestionnaire {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne
    @JoinColumn(
        name = "business_id",
        referencedColumnName = "id",
        foreignKey = @ForeignKey(name = "fk_questionnaire_business")
    )
    private SMEBusiness business;

    private Integer questionNo;
    @Column(columnDefinition = "TEXT")
    private String questionText;

    @Column(columnDefinition = "TEXT")
    private String answer;

    private LocalDateTime submittedAt = LocalDateTime.now();
}

```

1.21 SMEReadinessAssessment

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model.sme;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDateTime;

@Entity
@Table(name = "sme_readiness_assessments")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class SMEReadinessAssessment {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "owner_id", nullable = false)

```

```

private SMEUser owner;

private Integer financialScore; // 0–100
private Integer managementScore; // 0–100
private Integer documentationScore;
private Integer totalScore; // computed

@Column(length = 4000)
private String notes; // textual comments

@Column(nullable = false)
private LocalDateTime createdAt;
}

```

1.22 SMEUser

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model.sme;

import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDateTime;

// imports omitted for brevity
@Entity
@Table(name = "sme_users", uniqueConstraints = @UniqueConstraint(columnNames = "email"))
@Getter @Setter
@NoArgsConstructor @AllArgsConstructor
@Builder
public class SMEUser {

    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(nullable=false)
    private String name;

    @Column(nullable=false, unique=true)
    private String email;

    @Column(name="password_hash", nullable=false)
    private String passwordHash;

    @Column(nullable=false)
    private String role; // <— ADD THIS ("SME" by default when creating)
}

```

1.23 Bank

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara

```

```

Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model;

import jakarta.persistence.*;
import lombok.*;


@Entity
@Table(name = "banks")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Bank {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String code;
    private String region;
    //  Add these theme fields
    private String themeKey;
    private String primaryColor;
    private String secondaryColor;
    private String accentColor;

    @Column(name = "contact_email")
    private String contactEmail;

    @Column(name = "contact_number")
    private String contactNumber;

    //  FIXED — use wrapper Boolean instead of primitive boolean
    @Column(name = "is_active")
    private Boolean active = true;

    @Column(name = "created_at")
    private String createdAt;
}

```

1.24 Branch

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model;

import jakarta.persistence.*;
import lombok.*;

```

```

@Entity
@Table(name = "branches")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Branch {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String code;
    private String location;

    @ManyToOne
    @JoinColumn(name = "bank_id", nullable = false)
    private Bank bank;

    @OneToOne
    @JoinColumn(name = "manager_id")
    private User manager;

    @Column(name = "created_at")
    private String createdAt;
}

```

1.25 FinancialMetrics

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model;

import jakarta.persistence.*;
import lombok.*;

@Entity
@Table(name = "financial_metrics")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class FinancialMetrics {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private Long smeId;
    private Double revenue;
    private Double expenses;
    private Double assets;
    private Double liabilities;
    private Double loanToAssetRatio;
}

```

1.26 LoanApplication

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model;

import jakarta.persistence.*;
import lombok.*;

@Entity
@Table(name = "loan_applications")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class LoanApplication {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private Long smeId;
    private Long bankId;
    @ManyToOne
    @JoinColumn(name = "branch_id")
    private Branch branch;
    private Double amount;
    private String purpose;
    private String status; // PENDING, APPROVED, REJECTED
    private Double aiScore;
    private String createdAt;
}
```

1.27 PolicySettings

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDateTime;

@Entity
@Getter
@Setter
@NoArgsConstructor //  Required for Hibernate
@AllArgsConstructor // Optional: for builder/full-args constructor
@Builder
```

```

public class PolicySettings {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private double policyRate;
    private double savingsRate;
    private double reserveRatio;

    private LocalDateTime updatedAt = LocalDateTime.now();
}

```

1.28 User

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.model;

import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDateTime;

@Entity
@Table(name = "users")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(nullable = false)
    private String name;

    @Column(nullable = false, unique = true)
    private String email;

    @Column(name = "password_hash", nullable = false)
    private String passwordHash;

    /**
     * Role is stored as a simple string like "ADMIN", "BANK", "SME"
     * Later if you want enums: @Enumerated(EnumType.STRING)
     */
    @Column(nullable = false)
    private String role;
}

```

```

@Column(name = "created_at", nullable = false)
private LocalDateTime createdAt;

@Column(name = "bank_id")
private Long bankId;

@Column(name = "branch_id")
private Long branchId;

@PrePersist
protected void onCreate() {
    this.createdAt = LocalDateTime.now();
}

}

```

1.29 SMEAnalysisRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository.sme;

import com.example.smartfinancedashboard.model.sme.SMEAnalysis;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.List;
import java.util.Optional;

@Repository
public interface SMEAnalysisRepository extends JpaRepository<SMEAnalysis, Long> {

    // ✅ Latest analysis (used by dashboard + chat)
    Optional<SMEAnalysis> findTopByBusiness_IdOrderByCreatedAtDesc(Long businessId);

    // ✅ If you ever need all history
    List<SMEAnalysis> findByBusiness_IdOrderByCreatedAtDesc(Long businessId);
}

```

1.30 SMEBusinessRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository.sme;

import com.example.smartfinancedashboard.model.sme.SMEBusiness;

```

```

import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface SMEBusinessRepository extends JpaRepository<SMEBusiness, Long> {

    // Find all businesses belonging to a specific SME user (by foreign key)
    List<SMEBusiness> findBySmeUserId(Long smeUserId);
}

```

1.31 SMEQuestionnaireRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository.sme;

import com.example.smartfinancedashboard.model.sme.SMEQuestionnaire;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface SMEQuestionnaireRepository extends JpaRepository<SMEQuestionnaire, Long> {

    // Correct Spring Data method
    List<SMEQuestionnaire> findByBusinessId(Long businessId);
    boolean existsByBusiness_Id(Long businessId);
}

```

1.32 SMEReadinessRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository.sme;

import com.example.smartfinancedashboard.model.sme.SMEReadinessAssessment;
import com.example.smartfinancedashboard.model.sme.SMEUser;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface SMEReadinessRepository extends JpaRepository<SMEReadinessAssessment, Long> {

    List<SMEReadinessAssessment> findByOwnerOrderByCreatedAtDesc(SMEUser owner);
}

```

1.33 SMEUserRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179

```

```

University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository.sme;

import com.example.smartfinancedashboard.model.sme.*;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;
import java.util.Optional;

public interface SMEUserRepository extends JpaRepository<SMEUser, Long> {
    boolean existsByEmail(String email);
    Optional<SMEUser> findByEmail(String email);
}

```

1.34 BankRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;

import com.example.smartfinancedashboard.model.Bank;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface BankRepository extends JpaRepository<Bank, Long> {
}

```

1.35 BranchRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;

import com.example.smartfinancedashboard.model.Branch;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface BranchRepository extends JpaRepository<Branch, Long> {
    List<Branch> findByBankId(Long bankId);
    long countByBankId(Long bankId);
}

```

1.36 FinancialMetricsRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;
import com.example.smartfinancedashboard.model.FinancialMetrics;
import org.springframework.data.jpa.repository.JpaRepository;

public interface FinancialMetricsRepository extends JpaRepository<FinancialMetrics, Long> {
    FinancialMetrics findBySmeId(Long smeId);
}

```

1.37 LoanApplicationRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;
import com.example.smartfinancedashboard.model.LoanApplication;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface LoanApplicationRepository extends JpaRepository<LoanApplication, Long> {
    List<LoanApplication> findByBankId(Long bankId);
    List<LoanApplication> findBySmeId(Long smeId);
    List<LoanApplication> findByBranch_Id(Long branchId);
    long countByBankId(Long bankId); // ✅ total loans
    long countByBankIdAndStatus(Long bankId, String status);
}

```

1.38 PolicyRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;

import com.example.smartfinancedashboard.model.PolicySettings;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface PolicyRepository extends JpaRepository<PolicySettings, Long> {
}

```

1.39 PolicySettingsRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;

import com.example.smartfinancedashboard.model.PolicySettings;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PolicySettingsRepository extends JpaRepository<PolicySettings, Long> {
}

```

1.40 UserRepository

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.repository;

import com.example.smartfinancedashboard.model.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.*;

@Repository
public interface UserRepository extends JpaRepository<User, Long> {

    Optional<User> findByEmail(String email);    // ✓ For login (JwtAuthFilter)

    boolean existsByEmail(String email);
    List<User> findByBankId(Long bankId);
    List<User> findByBranchIdAndRole(Long branchId, String role); // ✓ For registration (AuthController)
}

```

1.41 CustomUserDetailsService

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.security;

import com.example.smartfinancedashboard.model.User;
import com.example.smartfinancedashboard.model.sme.SMEUser;
import com.example.smartfinancedashboard.repository.UserRepository;
import com.example.smartfinancedashboard.repository.sme.SMEUserRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.security.core.GrantedAuthority;

```

```

import org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
@RequiredArgsConstructor
public class CustomUserDetailsService implements UserDetailsService {

    private final UserRepository userRepository; // BANK users
    private final SMEUserRepository smeUserRepository; // SME users

    private String normalizeRole(String role, String fallback) {
        String r = (role == null || role.isBlank()) ? fallback : role.trim();
        return r.startsWith("ROLE_") ? r : "ROLE_" + r;
    }

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

        // 1) BANK user
        var bankOpt = userRepository.findByEmail(username);
        if (bankOpt.isPresent()) {
            User u = bankOpt.get();

            String authority = normalizeRole(u.getRole(), "BANK");
            List<GrantedAuthority> authorities =
                List.of(new SimpleGrantedAuthority(authority));

            return new org.springframework.security.core.userdetails.User(
                u.getEmail(),
                u.getPasswordHash(), // make sure this is correct field name
                authorities
            );
        }

        // 2) SME user
        var smeOpt = smeUserRepository.findByEmail(username);
        if (smeOpt.isPresent()) {
            SMEUser s = smeOpt.get();

            String authority = normalizeRole(s.getRole(), "SME");
            List<GrantedAuthority> authorities =
                List.of(new SimpleGrantedAuthority(authority));

            return new org.springframework.security.core.userdetails.User(
                s.getEmail(),
                s.getPasswordHash(), // make sure this is correct field name
                authorities
            );
        }

        throw new UsernameNotFoundException("User not found with email: " + username);
    }
}

```

1.42 JwtAuthFilter

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.security;

import jakarta.servlet.FilterChain;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import lombok.RequiredArgsConstructor;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.web.authentication.WebAuthenticationDetailsSource;
import org.springframework.stereotype.Component;
import org.springframework.web.filter.OncePerRequestFilter;

import java.io.IOException;

@Component
@RequiredArgsConstructor
public class JwtAuthFilter extends OncePerRequestFilter {

    private final JwtService jwtService;
    private final UserDetailsService userDetailsService;

    @Override
    protected void doFilterInternal(
        HttpServletRequest request,
        HttpServletResponse response,
        FilterChain filterChain
    ) throws ServletException, IOException {
        final String authHeader = request.getHeader("Authorization");

        if (authHeader == null || !authHeader.startsWith("Bearer ")) {
            filterChain.doFilter(request, response);
            return;
        }

        final String jwt = authHeader.substring(7);
        final String email;

        try {
            email = jwtService.extractEmail(jwt);
        } catch (Exception e) {
            filterChain.doFilter(request, response);
            return;
        }
    }
}
```

```

        }

        if (email != null && SecurityContextHolder.getContext().getAuthentication() == null) {
            UserDetails userDetails = userDetailsService.loadUserByUsername(email);

            if (jwtService.isTokenValid(jwt, userDetails)) {
                UsernamePasswordAuthenticationToken authToken =
                    new UsernamePasswordAuthenticationToken(
                        userDetails,
                        null,
                        userDetails.getAuthorities()
                    );
                authToken.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
                SecurityContextHolder.getContext().setAuthentication(authToken);
            }
        }

        filterChain.doFilter(request, response);
    }
}

```

1.43 JwtService

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.security;

import io.jsonwebtoken.*;
import io.jsonwebtoken.security.Keys;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.stereotype.Service;

import java.security.Key;
import java.util.Date;
import java.util.Map;
import java.util.function.Function;

@Service
public class JwtService {

    // ✅ make sure this is SAME secret you used before (must be >= 32 chars)
    private static final String SECRET =
        "CHANGE_ME_TO_A_32+_CHAR_SECRET_KEY_FOR_HS256_SIGNING!!!";

    private Key getSigningKey() {
        return Keys.hmacShaKeyFor(SECRET.getBytes());
    }

    // ✅ keep old method too (optional)
    public String generateToken(String email) {
        return generateToken(email, "SME");
    }
}

```

```

public String generateToken(String email, String role) {
    return Jwts.builder()
        .setClaims(Map.of("role", role))
        .setSubject(email)
        .setIssuedAt(new Date(System.currentTimeMillis()))
        .setExpiration(new Date(System.currentTimeMillis() + 1000L * 60 * 60 * 24)) // 24h
        .signWith(getSigningKey(), SignatureAlgorithm.HS256)
        .compact();
}

public String extractEmail(String token) {
    return extractClaim(token, Claims::getSubject);
}

public String extractRole(String token) {
    return extractClaim(token, claims -> claims.get("role", String.class));
}

public <T> T extractClaim(String token, Function<Claims, T> resolver) {
    final Claims claims = extractAllClaims(token);
    return resolver.apply(claims);
}

private Claims extractAllClaims(String token) {
    return Jwts.parserBuilder()
        .setSigningKey(getSigningKey())
        .build()
        .parseClaimsJws(token)
        .getBody();
}

private boolean isTokenExpired(String token) {
    Date exp = extractClaim(token, Claims::getExpiration);
    return exp.before(new Date());
}

// ✅ overload (used by /me endpoints)
public boolean isTokenValid(String token, String email) {
    String extracted = extractEmail(token);
    return extracted.equals(email) && !isTokenExpired(token);
}

// ✅ used by JwtAuthFilter
public boolean isTokenValid(String token, UserDetails userDetails) {
    String extracted = extractEmail(token);
    return extracted.equals(userDetails.getUsername()) && !isTokenExpired(token);
}
}

```

1.44 SecurityConfig

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.security;

```

```

import jakarta.servlet.http.HttpServletResponse;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.AuthenticationManager;
import
org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.CorsConfigurationSource;
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;

import java.util.List;

@Configuration
@EnableMethodSecurity
public class SecurityConfig {

    @Bean
    public CorsConfigurationSource corsConfigurationSource() {
        CorsConfiguration config = new CorsConfiguration();
        config.setAllowedOrigins(List.of("http://localhost:5173"));
        config.setAllowedMethods(List.of("GET", "POST", "PUT", "DELETE", "OPTIONS"));
        config.setAllowedHeaders(List.of("Authorization", "Content-Type"));
        config.setExposedHeaders(List.of("Authorization"));
        config.setAllowCredentials(true);
    }

    UrlBasedCorsConfigurationSource src = new UrlBasedCorsConfigurationSource();
    src.registerCorsConfiguration("/**", config);
    return src;
}

@Bean
public BCryptPasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}

@Bean
public AuthenticationManager authManager(AuthenticationConfiguration config) throws Exception {
    return config.getAuthenticationManager();
}

@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity http,
                                              JwtAuthFilter jwtAuthFilter) throws Exception {

    http
        .csrf(cs -> cs.disable())
        .cors(c -> c.configurationSource(corsConfigurationSource()))
        .sessionManagement(sm -> sm.sessionCreationPolicy(SessionCreationPolicy.STATELESS))

        .authorizeHttpRequests(auth -> auth
            // allow preflight
            .requestMatchers(org.springframework.http.HttpMethod.OPTIONS, "/**").permitAll()
}

```

```

        // SME public auth
        .requestMatchers("/api/sme/auth/**").permitAll()

        // Bank public auth
        .requestMatchers("/api/auth/**").permitAll()

        // SME protected
        .requestMatchers("/api/sme/**").hasRole("SME")

        // Bank roles
        .requestMatchers("/api/bank-admin/**").hasAnyRole("ADMIN", "BANK_ADMIN",
    "BANK_MANAGER")
        .requestMatchers("/api/manager/**").hasRole("BANK_MANAGER")
        .requestMatchers("/api/officer/**").hasRole("BANK_OFFICER")

        .anyRequest().authenticated()
    )

    .exceptionHandling(ex -> ex
        .authenticationEntryPoint((req, res, e) ->
            res.sendError(HttpStatus.SC_UNAUTHORIZED))
        .accessDeniedHandler((req, res, e) ->
            res.sendError(HttpStatus.SC_FORBIDDEN))
    )

    .addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class);

    return http.build();
}
}

```

1.45 AIService

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard.service;

import com.example.smartfinancedashboard.model.sme.SMEBusiness;
import com.example.smartfinancedashboard.model.sme.SMEQuestionnaire;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.node.ArrayNode;
import com.fasterxml.jackson.databind.node.ObjectNode;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import okhttp3.*;

import org.springframework.stereotype.Service;

import java.io.IOException;
import java.time.format.DateTimeFormatter;
import java.util.List;
import java.util.concurrent.TimeUnit;

@Slf4j

```

```

@Service
@RequiredArgsConstructor
public class AIService {

    private static final String OLLAMA_URL = "http://localhost:11434/api/chat";
    private static final String MODEL_NAME = "phi3:3.8b";

    private final ObjectMapper objectMapper;

    private final OkHttpClient httpClient = new OkHttpClient.Builder()
        .connectTimeout(30, TimeUnit.SECONDS)
        .readTimeout(120, TimeUnit.SECONDS)
        .build();

    /**
     * Main method: build prompt from business + questionnaire, call Ollama,
     * clean the JSON, and return a JsonNode.
     * If anything fails, we return a simple rule-based fallback.
     */
    public JsonNode analyzeBusiness(SMEBusiness business, List<SMEQuestionnaire> answers) {
        try {
            String prompt = buildPrompt(business, answers);
            String rawContent = callAiWithPrompt(prompt);

            log.info("AIService: raw AI message content = {}", rawContent);

            String cleanedJson = extractAndCleanJson(rawContent);

            log.debug("AIService: cleaned JSON text = {}", cleanedJson);

            JsonNode node = objectMapper.readTree(cleanedJson);
            log.debug("AIService: parsed JSON successfully");
            return node;
        } catch (Exception ex) {
            log.error("AIService: error during AI analysis, falling back to rule-based analysis", ex);
            return buildFallbackAnalysis(business, answers);
        }
    }

    /**
     * Simple chat-style helper used by SMEChatController.
     * (Kept compatible with your existing controller.)
     */
    public String chatBusinessTips(String context, String question) {
        try {
            String prompt = "You are a Sri Lankan SME loan advisor. " +
                "Use the following context about the SME and then answer the user's question briefly and
                clearly.\n\n" +
                "==== SME CONTEXT ====\n" + context + "\n\n" +
                "==== QUESTION ====\n" + question + "\n\n" +
                "Give your answer in 2–3 short paragraphs, no Markdown.';

            String content = callAiWithPrompt(prompt);
            // For chat we just return the plain content (no JSON expected)
            return content;
        } catch (Exception e) {
            log.error("AIService: chatBusinessTips failed", e);
            return "Sorry, the advisory assistant is temporarily unavailable. Please try again in a moment.";
        }
    }
}

```

```

// ----- Internal helpers -----
private String buildPrompt(SMEBusiness business, List<SMEQuestionnaire> answers) {
    StringBuilder sb = new StringBuilder();

    sb.append("You are an SME loan readiness assistant for Sri Lankan banks.\n")
        .append("Analyze the SME and return ONLY valid JSON in the exact structure below.\n\n")
        .append("### STRICT RULES ###\n")
        .append("- Respond with VALID JSON only.\n")
        .append("- Do NOT wrap JSON in ``json fences.\n")
        .append("- Do NOT include comments (no // or /* */).\n")
        .append("- All keys must be in double quotes.\n\n")
        .append("### REQUIRED JSON SHAPE ###\n")
        .append("{\n            \"summary\": \"string\",\n            \"strengths\": [\"string\", \"string\"],\n            \"weaknesses\": [\"string\"],\n            \"improvements\": [\"string\"],\n            \"bank_suggestions\": [\"string\"],\n            \"loan_plans\": [\"string\"],\n            \"financial_ratios\": [\n                {\n                    \"name\": \"string\", \"value\": \"string\", \"interpretation\": \"string\"\n                }\n            ],\n            \"management_assessment\": {\n                \"experience\": \"string\",\n                \"governance\": \"string\",\n                \"risk_behaviour\": \"string\"\n            },\n            \"document_checklist\": [\n                {\n                    \"name\": \"string\", \"importance\": \"HIGH|MEDIUM|LOW\"\n                }\n            ],\n            \"final_score\": 0\n        }\n\nNow here is the SME data:\n\n")
        .append("Business name: ").append(nullSafe(business.getBusinessName())).append("\n")
        .append("Industry: ").append(nullSafe(business.getIndustry())).append("\n")
        .append("Location: ").append(nullSafe(business.getLocation())).append("\n")
        .append("Annual revenue: ").append(business.getAnnualRevenue()).append("\n")
        .append("Employees: ").append(business.getEmployees()).append("\n")
        .append("Registered at: ");

    if (business.getRegisteredAt() != null) {
        sb.append(business.getRegisteredAt().format(DateTimeFormatter.ISO_LOCAL_DATE));
    } else {
        sb.append("N/A");
    }
    sb.append("\n")
        .append("Bank relationship: ").append(nullSafe(business.getBankRelationship())).append("\n\n")
        .append("### Questionnaire answers ###\n");

    if (answers == null || answers.isEmpty()) {
        sb.append("No questionnaire answers available.\n");
    } else {
        for (SMEQuestionnaire q : answers) {
            sb.append("Q").append(q.getQuestionNo()).append(": ")
                .append(nullSafe(q.getQuestionText())).append("\n")
                .append("A: ").append(nullSafe(q.getAnswer())).append("\n\n");
        }
    }
}

```

```

        sb.append("Please perform a realistic, conservative risk assessment for a Sri Lankan bank and respond  

ONLY with JSON.");
    }

    return sb.toString();
}

private String nullSafe(String s) {
    return s == null ? "N/A" : s;
}

/**
 * Calls Ollama /api/chat and returns the assistant message content as plain text.
 */
private String callAiWithPrompt(String prompt) throws IOException {
    log.info("AIService: calling model '{}' at {}", MODEL_NAME, OLLAMA_URL);

    ObjectNode root = objectMapper.createObjectNode();
    root.put("model", MODEL_NAME);
    root.put("stream", false);

    ArrayNode messages = root.putArray("messages");
    ObjectNode sys = messages.addObject();
    sys.put("role", "system");
    sys.put("content", "You are a smart SME loan analysis assistant for Sri Lankan banks.");
    ObjectNode user = messages.addObject();
    user.put("role", "user");
    user.put("content", prompt);

    String reqJson = objectMapper.writeValueAsString(root);

    RequestBody body = RequestBody.create(
        reqJson,
        MediaType.parse("application/json")
    );

    Request request = new Request.Builder()
        .url(OLLAMA_URL)
        .post(body)
        .build();

    try (Response response = httpClient.newCall(request).execute()) {
        int status = response.code();
        log.info("AIService: HTTP status from AI server = {}", status);

        if (!response.isSuccessful()) {
            throw new IOException("AI server HTTP " + status);
        }
    }

    String responseBody = response.body() != null ? response.body().string() : "";
    log.debug("AIService: raw HTTP body from AI server = {}", responseBody);

    JsonNode rootNode = objectMapper.readTree(responseBody);
    String content = rootNode.path("message").path("content").asText("");

    if (content == null || content.isBlank()) {
        throw new IOException("Empty content from AI server");
    }

    return content;
}

```

```

        }

    /**
     * Removes ```fences and any // comments so Jackson can parse.
     */
    private String extractAndCleanJson(String rawContent) {
        if (rawContent == null) return "{}";

        String json = rawContent.trim();

        // Strip ```json and ``` fences if present
        if (json.startsWith("```")) {
            int firstNewline = json.indexOf('\n');
            if (firstNewline > 0) {
                json = json.substring(firstNewline + 1);
            }
        }
        if (json.endsWith("```")) {
            json = json.substring(0, json.lastIndexOf("```")).trim();
        }

        // Remove // comments (full-line or inline)
        StringBuilder sb = new StringBuilder();
        String[] lines = json.split("\n");
        for (String line : lines) {
            int commentIdx = line.indexOf("//");
            if (commentIdx >= 0) {
                line = line.substring(0, commentIdx);
            }
            sb.append(line).append("\n");
        }

        json = sb.toString().trim();

        log.debug("AIService: safe JSON text = {}", json);
        return json;
    }

    /**
     * Fallback JSON if Ollama fails or returns invalid JSON.
     * Still follows the same structure so your frontend can show it.
     */
    private JsonNode buildFallbackAnalysis(SMEBusiness business, List<SMEQuestionnaire> answers) {
        ObjectNode root = objectMapper.createObjectNode();

        String businessName = business.getBusinessName() != null
            ? business.getBusinessName()
            : "this SME";

        root.put("summary", businessName + " has provided limited data. " +
            "This is an automated rule-based assessment generated without AI due to a temporary issue.");

        // strengths
        ArrayNode strengths = root.putArray("strengths");
        if (business.getAnnualRevenue() != null && business.getAnnualRevenue() > 0) {
            strengths.add("Has reported annual revenue of " + business.getAnnualRevenue());
        }
        if (business.getEmployees() != null && business.getEmployees() > 0) {
            strengths.add("Provides employment to " + business.getEmployees() + " staff members.");
        }
    }
}

```

```

        }

strengths.add("Basic business profile is recorded in the system.");

// weaknesses
ArrayNode weaknesses = root.putArray("weaknesses");
weaknesses.add("AI-powered qualitative assessment is temporarily unavailable.");
weaknesses.add("Questionnaire-based risk analysis could not be fully processed.");

// improvements
ArrayNode improvements = root.putArray("improvements");
improvements.add("Provide complete and up-to-date questionnaire answers for deeper analysis.");
improvements.add("Maintain digital financial records and keep bank statements ready for review.");

// bank suggestions
ArrayNode bankSuggestions = root.putArray("bank_suggestions");
bankSuggestions.add("Use this summary as a preliminary view only; conduct a full credit assessment before loan approval.");
bankSuggestions.add("Request complete financial documentation (bank statements, tax returns, etc.) from the SME.");

// loan plans
ArrayNode loanPlans = root.putArray("loan_plans");
loanPlans.add("Consider a small working capital facility after full documentation and KYC checks.");
loanPlans.add("Use cautious limits until a complete AI/credit analysis can be performed.");

// financial_ratios (empty but valid)
ArrayNode ratios = root.putArray("financial_ratios");
ObjectNode r = ratios.addObject();
r.put("name", "Debt-to-income ratio");
r.put("value", "N/A");
r.put("interpretation", "Not calculated because AI service was unavailable.");

// management_assessment
ObjectNode mgmt = root.putObject("management_assessment");
mgmt.put("experience", "Not fully evaluated because AI service was unavailable.");
mgmt.put("governance", "Governance practices cannot be scored without qualitative analysis.");
mgmt.put("riskBehaviour", "Risk behaviour could not be evaluated from the available data.");

// document_checklist
ArrayNode docs = root.putArray("document_checklist");
ObjectNode d1 = docs.addObject();
d1.put("name", "Bank statements (last 6–12 months)");
d1.put("importance", "HIGH");
ObjectNode d2 = docs.addObject();
d2.put("name", "Latest financial statements / income records");
d2.put("importance", "HIGH");

// final_score – neutral mid score
root.put("final_score", 50);

return root;
}
}

```

1.46 DataSeeder

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics

```

Project: SME Loan Readiness Tool

Date: 2026-01-23

*/

```
package com.example.smartfinancedashboard;
```

```
import com.example.smartfinancedashboard.model.Bank;
import com.example.smartfinancedashboard.model.PolicySettings;
import com.example.smartfinancedashboard.model.User;
import com.example.smartfinancedashboard.repository.BankRepository;
import com.example.smartfinancedashboard.repository.PolicyRepository;
import com.example.smartfinancedashboard.repository.UserRepository;
import org.springframework.boot.CommandLineRunner;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
```

```
@Configuration
```

```
public class DataSeeder {
```

```
    @Bean
```

```
    CommandLineRunner seedUsersAndBanks(UserRepository users,
                                         BankRepository banks,
                                         PolicyRepository policies,
                                         BCryptPasswordEncoder enc) {
```

```
        return args -> {
```

```
        /* =====
```

```
        🏛 Seed Banks
```

```
===== */
```

```
        if (banks.count() == 0) {
            banks.save(Bank.builder().name("Bank of
Ceylon").code("BOC001").region("Western").active(true).build());
            banks.save(Bank.builder().name("People's
Bank").code("PBK002").region("Central").active(true).build());
            banks.save(Bank.builder().name("Hatton National
Bank").code("HNB003").region("Southern").active(true).build());
            banks.save(Bank.builder().name("Sampath
Bank").code("SAMP004").region("Western").active(true).build());
            banks.save(Bank.builder().name("Commercial
Bank").code("COM005").region("National").active(true).build());
        }
```

```
        /* =====
```

```
        🧠 Seed Central Admin
```

```
===== */
```

```
        if (users.findByEmail("admin@central.gov").isEmpty()) {
            users.save(User.builder()
                .name("Central Admin")
                .email("admin@central.gov")
                .passwordHash(enc.encode("Admin@123"))
                .role("ADMIN")
                .build());
        }
```

```
        /* =====
```

```
        🏛 Seed Bank Admins
```

```
===== */
```

```
        if (users.findByEmail("admin@boc.lk").isEmpty()) {
            users.save(User.builder()
```

```

    .name("BOC Admin")
    .email("admin@boc.lk")
    .passwordHash(enc.encode("Boc@123"))
    .role("BANK_ADMIN")
    .bankId(1L)
    .build());
}

if (users.findByEmail("admin@peoples.lk").isEmpty()) {
    users.save(User.builder()
        .name("People's Bank Admin")
        .email("admin@peoples.lk")
        .passwordHash(enc.encode("People@123"))
        .role("BANK_ADMIN")
        .bankId(2L)
        .build());
}

if (users.findByEmail("admin@hnb.lk").isEmpty()) {
    users.save(User.builder()
        .name("HNB Admin")
        .email("admin@hnb.lk")
        .passwordHash(enc.encode("Hnb@123"))
        .role("BANK_ADMIN")
        .bankId(3L)
        .build());
}

if (users.findByEmail("admin@sampath.lk").isEmpty()) {
    users.save(User.builder()
        .name("Sampath Bank Admin")
        .email("admin@sampath.lk")
        .passwordHash(enc.encode("Sampath@123"))
        .role("BANK_ADMIN")
        .bankId(4L)
        .build());
}

if (users.findByEmail("admin@commercial.lk").isEmpty()) {
    users.save(User.builder()
        .name("Commercial Bank Admin")
        .email("admin@commercial.lk")
        .passwordHash(enc.encode("Commercial@123"))
        .role("BANK_ADMIN")
        .bankId(5L)
        .build());
}

/*
=====
  Seed Policy Defaults
=====
*/
if (policies.count() == 0) {
    policies.save(PolicySettings.builder()
        .policyRate(8.5)
        .savingsRate(6.2)
        .reserveRatio(3.0)
        .build());
}

/*
=====

```

 FIX EXISTING USERS WITH PLAIN TEXT PASSWORDS

```
=====
for (User u : users.findAll()) {
    String pass = u.getPasswordHash();
    // skip users who already have BCrypt
    if (pass != null && !pass.startsWith("$2a$")) {
        u.setPasswordHash(enc.encode(pass));
        users.save(u);
        System.out.println("🔒 Fixed password for: " + u.getEmail());
    }
}
}
```

1.47 SmartFinanceDashboardApplication

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
package com.example.smartfinancedashboard;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SmartFinanceDashboardApplication {

    public static void main(String[] args) {
        SpringApplication.run(SmartFinanceDashboardApplication.class, args);
    }
}
```

1.48 application.properties

```
server.port=8081
spring.application.name=Smart Finance Dashboard
spring.datasource.url=jdbc:postgresql://localhost:5432/sme_dashboard
spring.datasource.username=postgres
spring.datasource.password=admin123
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

app.cors.allowed-origins=http://localhost:5173
app.jwt.secret=change-this-to-a-long-random-secret-key
app.jwt.expMinutes=120
openai.api.url=http://localhost:11434/api/chat
openai.model=phi3:3.8b
logging.level.com.example.smartfinancedashboard.service.AIService=DEBUG
```

2. Frontend Codes

2.1 smeClient

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import axios from "axios";

const api = axios.create({
  baseURL: "http://localhost:8081/api",
});

// Attach SME token automatically
api.interceptors.request.use((config) => {
  const token = localStorage.getItem("smeToken");
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

/* ===== SME AUTH ===== */
export const smeAuth = {
  // POST /api/sme/auth/register
  register(payload) {
    return api.post("/sme/auth/register", payload);
  },

  // POST /api/sme/auth/login
  login(credentials) {
    return api.post("/sme/auth/login", credentials);
  },

  // GET /api/sme/profile/me
  me() {
    return api.get("/sme/profile/me");
  },
};

/* ===== BUSINESS ===== */
export const smeBusiness = {
  // GET /api/sme/business
  list() {
    return api.get("/sme/business");
  },

  // POST /api/sme/business
  create(payload) {
    return api.post("/sme/business", payload);
  },

  // GET /api/sme/business/{businessId}
  get(businessId) {
    return api.get('/sme/business/${businessId}`);
  },

  // DELETE /api/sme/business/{businessId}
}
```

```

remove(businessId) {
    return api.delete('/sme/business/${businessId}`);
},
};

/* ===== QUESTIONNAIRE ===== */
export const smeQuestionnaire = {
    // GET /api/sme/questionnaire/questions
    getQuestions() {
        return api.get("/sme/questionnaire/questions");
    },
    // POST /api/sme/questionnaire/submit/{businessId}
    submit(businessId, answers) {
        return api.post('/sme/questionnaire/submit/${businessId}', answers);
    },
    // GET /api/sme/questionnaire/status/{businessId}
    // should return something like: { completed: true/false }
    checkCompleted(businessId) {
        return api.get('/sme/questionnaire/status/${businessId}');
    },
};

/* ===== ANALYSIS ===== */
export const smeAnalysis = {
    // POST /api/sme/analysis/generate/{businessId}
    generate(businessId) {
        return api.post('/sme/analysis/generate/${businessId}');
    },
    // GET /api/sme/analysis/{businessId}
    // used as "getByBusiness" in SMEPortal and SMEDashboard
    getByBusiness(businessId) {
        return api.get('/sme/analysis/${businessId}');
    },
    // optional alias if some old code still uses "get"
    get(businessId) {
        return api.get('/sme/analysis/${businessId}');
    },
    // POST /api/sme/analysis/chat/{businessId}
    chat(businessId, message) {
        return api.post('/sme/analysis/chat/${businessId}', { message });
    },
};

```

2.2 AddManagerModal

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useState } from "react";

export default function AddManagerModal({ branch, token, onClose, onSuccess, theme }) {

```

```

const [form, setForm] = useState({
  name: "",
  email: "",
  password: "",
});
const [loading, setLoading] = useState(false);

const handleSubmit = async (e) => {
  e.preventDefault();
  setLoading(true);
  try {
    const res = await fetch(`http://localhost:8081/api/bank-admin/branches/${branch.id}/manager`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        Authorization: `Bearer ${token}`,
      },
      body: JSON.stringify(form),
    });
    if (!res.ok) throw new Error("Failed to add manager");
    await res.json();
    onSuccess();
    onClose();
  } catch (err) {
    alert(err.message);
  } finally {
    setLoading(false);
  }
};

return (
  <div
    className="fixed inset-0 bg-black/40 flex items-center justify-center"
    onClick={onClose}
  >
  <div
    className="bg-white p-6 rounded-xl shadow-lg w-96"
    onClick={(e) => e.stopPropagation()}
  >
    <h2 style={{ color: theme.accent }}>Assign Manager to {branch.name}</h2>
    <form onSubmit={handleSubmit} className="flex flex-col gap-2 mt-3">
      <input
        placeholder="Full name"
        value={form.name}
        onChange={(e) => setForm({ ...form, name: e.target.value })}
        required
      />
      <input
        placeholder="Email"
        type="email"
        value={form.email}
        onChange={(e) => setForm({ ...form, email: e.target.value })}
        required
      />
      <input
        placeholder="Password"
        type="password"
        value={form.password}
        onChange={(e) => setForm({ ...form, password: e.target.value })}
        required
      />
    </form>
  </div>
)

```

```

        />

        <button
            type="submit"
            disabled={loading}
            style={{
                background: theme.accent,
                color: "#fff",
                padding: "0.6rem",
                borderRadius: "8px",
                marginTop: "0.5rem",
            }}
        >
            {loading ? "Saving..." : "Add Manager"}
        </button>
    </form>
</div>
</div>
);
}

```

2.3 AnimatedBlobs

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
export default function AnimatedBlobs() {
    return (
        <div className="pointer-events-none fixed inset-0 -z-10 overflow-hidden">
            <div
                className=""
                absolute -top-32 -left-24 w-80 h-80 rounded-full
                bg-[var(--primary)] opacity-20 blur-3xl
                animate-pulse
            >
            </div>
            <div
                className=""
                absolute -bottom-40 -right-24 w-96 h-96 rounded-full
                bg-[var(--accent)] opacity-25 blur-3xl
                animate-pulse
            >
            </div>
            <div
                className=""
                absolute top-1/3 left-1/2 -translate-x-1/2 w-[32rem] h-[32rem]
                rounded-full bg-[var(--primary)] opacity-10
                blur-3xl
            >
            </div>
        </div>
    );
}

```

2.4 AnimatedPage

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { motion } from "framer-motion";

export default function AnimatedPage({ children }) {
  const animations = {
    initial: { opacity: 0, y: 20 },
    animate: { opacity: 1, y: 0 },
    exit: { opacity: 0, y: -20 },
  };

  return (
    <motion.div
      variants={animations}
      initial="initial"
      animate="animate"
      exit="exit"
      transition={{ duration: 0.5, ease: "easeOut" }}
      className="min-h-screen flex items-center justify-center bg-gradient-to-br from-blue-100 to-indigo-200"
    >
      {children}
    </motion.div>
  );
}
```

2.5 ContactManagerModal

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useState } from "react";

export default function ContactManagerModal({ branch, manager, onClose, theme }) {
  const [message, setMessage] = useState("");

  const handleSend = () => {
    alert(`✉️ Message sent to ${manager.name}: "${message}"`);
    onClose();
  };

  return (
    <div
      className="fixed inset-0 bg-black/50 flex items-center justify-center"
      onClick={onClose}
    >
      <div
        className="bg-white rounded-xl p-6 w-96 shadow-xl"
```

```

    onClick={(e) => e.stopPropagation()}
  >
  <h2 style={{ color: theme.accent }}>
    ⚡ Contact {manager.name}
  </h2>
  <p className="mt-2 text-sm text-gray-600">
    Send a quick message to {branch.name} branch manager.
  </p>

  <textarea
    className="w-full border rounded mt-3 p-2"
    rows="4"
    value={message}
    onChange={(e) => setMessage(e.target.value)}
    placeholder="Type your message here..."
  />

  <div className="flex justify-end gap-2 mt-3">
    <button
      onClick={onClose}
      className="px-3 py-1 rounded bg-gray-300 text-black"
    >
      Cancel
    </button>
    <button
      onClick={handleSend}
      className="px-4 py-1 rounded text-white"
      style={{ background: theme.accent }}
    >
      Send
    </button>
  </div>
</div>
</div>
);
}

```

2.6 GuideBookModal

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import React from "react";

export default function GuideBookModal({ open, onClose }) {
  if (!open) return null;

  return (
    <div className="fixed inset-0 z-50 flex items-center justify-center">
      {/* backdrop */}
      <button
        className="absolute inset-0 bg-black/40"
        onClick={onClose}
        aria-label="Close guide"
      />

```

```

/* modal */


## SME Loan Readiness Guide



Quick handbook to use this system correctly



onClick={onClose}



className="text-sm px-3 py-1 rounded-lg border border-gray-200 hover:bg-gray-50"



×


```

<div className="p-5 max-h-[70vh] overflow-y-auto space-y-5 text-sm text-gray-700">

<Section title="1. What is this system?">

This web app helps SME owners understand bank requirements, improve weaknesses, and generate AI-based loan readiness insights before meeting a bank.

</Section>

<Section title="2. 3-step journey">

<ul className="list-disc pl-5 space-y-1">

Business Profile: enter business basics.

Questionnaire: answer honestly to assess risks/strengths.

AI Insights: get score + action plan + bank-ready suggestions.

</Section>

<Section title="3. Understanding the score">

<div className="grid grid-cols-2 gap-2 text-xs">

<ScoreBadge label="80–100" text="Bank-ready" />

<ScoreBadge label="60–79" text="Almost ready" />

<ScoreBadge label="40–59" text="Needs improvement" />

<ScoreBadge label="0–39" text="High risk" />

</div>

</Section>

<Section title="4. What you will receive">

<ul className="list-disc pl-5 space-y-1">

Strengths & weaknesses

90-day improvement actions

How a bank may see you

Possible loan structures

Document checklist for bank meeting

</Section>

<Section title="5. Tips for best results">

<ul className="list-disc pl-5 space-y-1">

Use realistic numbers for revenue/cash flow.

Update data and regenerate insights after improvements.

Prepare documents early (BR, bank statements, financials, plan).

</Section>

</div>

```

        <div className="p-5 border-t border-gray-100 flex items-center justify-end gap-2">
          <button
            onClick={onClose}
            className="px-4 py-2 rounded-xl border border-gray-200 hover:bg-gray-50 text-sm"
          >
            Close
          </button>
          <button
            onClick={onClose}
            className="px-4 py-2 rounded-xl bg-black text-white hover:opacity-90 text-sm"
          >
            Got it
          </button>
        </div>
      </div>
    );
}

function Section({ title, children }) {
  return (
    <div className="space-y-2">
      <h3 className="font-semibold text-gray-900">{title}</h3>
      <div className="text-gray-700 leading-relaxed">{children}</div>
    </div>
  );
}

function ScoreBadge({ label, text }) {
  return (
    <div className="rounded-xl border border-gray-200 bg-gray-50 px-3 py-2">
      <div className="font-semibold text-gray-900">{label}</div>
      <div className="text-gray-600">{text}</div>
    </div>
  );
}

```

2.7 SmeAnalysisPage

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/

```

```

import React, { useEffect, useState } from "react";
import { useParams } from "react-router-dom";

```

```

/* =====
 Helper Functions
===== */

```

```

function parseJsonArray(raw) {
  if (!raw) return [];
  try {
    const parsed = JSON.parse(raw);
    return Array.isArray(parsed) ? parsed : [];
  } catch (e) {

```

```

        console.warn("Failed to parse JSON array:", e);
        return [];
    }
}

function parseJsonObject(raw) {
    if (!raw) return {};
    try {
        const parsed = JSON.parse(raw);
        return typeof parsed === "object" && parsed !== null ? parsed : {};
    } catch (e) {
        console.warn("Failed to parse JSON object:", e);
        return {};
    }
}

function formatDateTime(value) {
    if (!value) return "Not available";
    try {
        const d = new Date(value);
        return d.toLocaleString();
    } catch {
        return value;
    }
}

/* =====
   MAIN PAGE
===== */
const SmeAnalysisPage = () => {
    const { businessId } = useParams();
    const [analysis, setAnalysis] = useState(null);
    const [loading, setLoading] = useState(false);
    const [generating, setGenerating] = useState(false);
    const [error, setError] = useState("");

    /* =====
       Load Analysis from Backend
===== */
    const loadAnalysis = async () => {
        if (!businessId) return;

        setLoading(true);
        setError("");

        try {
            const res = await fetch(
                `http://localhost:8081/api/sme/analysis/${businessId}`,
                {
                    headers: {
                        Authorization: `Bearer ${localStorage.getItem("smeToken")}`,
                    },
                }
            );

            if (!res.ok) {
                if (res.status === 404) {
                    setAnalysis(null);
                } else {

```

```

        throw new Error("Server error: " + res.status);
    }
} else {
    const data = await res.json();
    setAnalysis(data);
}
} catch (e) {
    console.error(e);
    setError("Failed to load analysis. Please try again.");
} finally {
    setLoading(false);
}
};

/* =====
Generate New AI Analysis
===== */
const generateAnalysis = async () => {
    if (!businessId) return;

    setGenerating(true);
    setError("");

    try {
        const res = await fetch(
            `http://localhost:8081/api/sme/analysis/generate/${businessId}`,
            {
                method: "POST",
                headers: {
                    Authorization: `Bearer ${localStorage.getItem("smeToken")}`,
                },
            }
        );

        if (!res.ok) {
            const text = await res.text();
            throw new Error(text || "Failed to generate analysis.");
        }

        const data = await res.json();
        setAnalysis(data);
    } catch (e) {
        console.error(e);
        setError("Could not generate analysis. Check questionnaire / AI backend.");
    } finally {
        setGenerating(false);
    }
};

useEffect(() => {
    loadAnalysis();
}, [businessId]);

/* =====
Parse
===== */

const strengths = parseJsonArray(analysis?.strengths);
const weaknesses = parseJsonArray(analysis?.weaknesses);
const improvements = parseJsonArray(analysis?.improvements);

```

```

const bankSuggestions = parseJsonArray(analysis?.bankSuggestions);
const loanPlans = parseJsonArray(analysis?.loanPlans);

const financialRatios = parseJsonObject(analysis?.financialRatios);
const managementAssessment = parseJsonObject(analysis?.managementAssessment);
const documentChecklist = parseJsonObject(analysis?.documentChecklist);

const finalScore = analysis?.finalScore ?? null;
const businessName = analysis?.business?.businessName || "SME Business";

const scoreColor =
  finalScore == null
    ? "bg-gray-200 text-gray-800"
    : finalScore >= 80
      ? "bg-green-100 text-green-800"
      : finalScore >= 60
        ? "bg-yellow-100 text-yellow-800"
        : "bg-red-100 text-red-800";

return (
  <div className="min-h-screen bg-slate-50 py-6 px-4 sm:px-8">
    <div className="max-w-6xl mx-auto space-y-6">
      {/* ===== HEADER ===== */}
      <div className="flex flex-col sm:flex-row justify-between gap-4">
        <div>
          <p className="text-sm text-slate-500">SME Loan Readiness</p>
          <h1 className="text-3xl font-bold text-slate-900">
            {businessName}
          </h1>
          <p className="text-xs text-slate-500">
            Last generated:{" "}
            <span className="font-medium">
              {analysis?.generatedAt
                ? formatDateTime(analysis.generatedAt)
                : "No analysis yet"}
            </span>
          </p>
        </div>

        <div className="flex gap-3">
          {finalScore != null && (
            <div
              className={`px-4 py-2 rounded-xl text-sm font-semibold flex items-center gap-2
${scoreColor}`}
            >
              <span>Readiness Score:</span>
              <span className="text-lg">{finalScore}/100</span>
            </div>
          )}
        </div>
      </div>
    </div>
    <button
      onClick={generateAnalysis}
      disabled={generating}
      className="px-4 py-2 rounded-xl bg-indigo-600 text-white text-sm shadow-sm"
    >
      {generating ? "Generating..." : "Generate AI Analysis"}
    </button>
  </div>
</div>

```

```

/* ===== STATES ===== */
{loading && (
  <div className="bg-white p-4 rounded-xl shadow text-sm">
    Loading analysis...
  </div>
)}

{error && (
  <div className="bg-red-50 border-red-200 text-red-700 p-4 rounded-xl text-sm">
    {error}
  </div>
)}

{!loading && !analysis && !error && (
  <div className="bg-white p-5 rounded-xl shadow text-sm">
    No analysis yet. Click “Generate AI Analysis”.
  </div>
)}

/* ===== ANALYSIS ===== */
{analysis && (
  <div className="space-y-6">
    /* Summary */
    <div className="bg-white p-5 rounded-2xl shadow border border-slate-100">
      <h2 className="text-lg font-semibold mb-2">Executive Summary</h2>
      <p className="text-sm text-slate-700 leading-relaxed">
        {analysis.summary}
      </p>
    </div>

    /* Strengths & Weaknesses */
    <div className="grid md:grid-cols-2 gap-6">
      <ListCard
        title="Strengths"
        items={strengths}
        color="emerald"
        icon="✓"
      />
      <ListCard
        title="Weaknesses"
        items={weaknesses}
        color="rose"
        icon="⚠"
      />
    </div>

    /* Improvements */
    <ListNumberedCard
      title="Action Plan – Key Improvements"
      items={improvements}
    />

    /* Banks & Loan Plans */
    <div className="grid md:grid-cols-2 gap-6">
      <ListCard
        title="Bank Suggestions"
        items={bankSuggestions}
        color="indigo"
        icon="🏦"
      />

```

```

<ListCard
  title="Loan Plan Suggestions"
  items={loanPlans}
  color="amber"
  icon="💰"
/>
</div>

/* Financial Ratios */
<RatioSection financialRatios={financialRatios} />

/* Management Assessment */
<ManagementSection assessment={managementAssessment} />

/* Document Checklist */
<ChecklistSection checklist={documentChecklist} />
</div>
)
</div>
);
};

/* =====
 REUSABLE COMPONENTS
===== */

const ListCard = ({ title, items, color, icon }) => (
  <div className={`bg-white p-5 rounded-2xl shadow border border-${color}-100`} >
    <h2 className={`text-lg font-semibold text-${color}-700 mb-3`} >{title}</h2>
    {items.length === 0 ? (
      <p className="text-sm text-slate-600">No items listed.</p>
    ) : (
      <ul className="space-y-2 text-sm text-slate-700">
        {items.map((item, idx) => (
          <li key={idx} className="flex gap-2">
            <span className={`text-${color}-500`}>{icon}</span>
            <span>{item}</span>
          </li>
        )))
      </ul>
    )}
  </div>
);

const ListNumberedCard = ({ title, items }) => (
  <div className="bg-white p-5 rounded-2xl shadow border border-sky-100" >
    <h2 className="text-lg font-semibold text-sky-700 mb-3" >{title}</h2>
    {items.length === 0 ? (
      <p className="text-sm text-slate-600">No improvements listed.</p>
    ) : (
      <ol className="list-decimal list-inside space-y-2 text-sm text-slate-700" >
        {items.map((item, idx) => (
          <li key={idx}>{item}</li>
        )))
      </ol>
    )}
  </div>
);

```

```

const RatioSection = ({ financialRatios }) => (
  <div className="bg-white p-5 rounded-2xl shadow border border-slate-100">
    <h2 className="text-lg font-semibold text-slate-900 mb-3">
      Financial Ratios Overview
    </h2>
    <div className="grid sm:grid-cols-2 lg:grid-cols-4 gap-4">
      <RatioCard title="Debt-to-Asset" value={financialRatios.debt_to_asset_ratio} />
      <RatioCard title="Net Profit Margin" value={financialRatios.net_profit_margin} />
      <RatioCard
        title="Working Capital Ratio"
        value={financialRatios.working_capital_ratio}
      />
      <RatioCard title="Liquidity Score" value={financialRatios.liquidity_score} />
    </div>
  </div>
);

const RatioCard = ({ title, value }) => (
  <div className="bg-slate-50 rounded-xl border p-4">
    <p className="text-xs text-slate-500 mb-1">{title}</p>
    <p className="text-lg font-semibold">{value || "N/A"}</p>
  </div>
);

const ManagementSection = ({ assessment }) => (
  <div className="bg-white p-5 rounded-2xl shadow border">
    <h2 className="text-lg font-semibold">Management Assessment</h2>
    <div className="grid md:grid-cols-3 gap-4 mt-4 text-sm text-slate-700">
      <AssessmentCard title="Leadership" icon="👤" text={assessment.leadership} />
      <AssessmentCard title="Operations" icon="⚙️" text={assessment.operations} />
      <AssessmentCard
        title="Risk Management"
        icon="⚠️"
        text={assessment.risk_management}
      />
    </div>
  </div>
);

const AssessmentCard = ({ title, icon, text }) => (
  <div className="bg-slate-50 rounded-xl border p-4">
    <div className="flex gap-2 items-center mb-2">
      <span>{icon}</span>
      <p className="font-semibold">{title}</p>
    </div>
    <p className="text-xs">{text || "No details provided."}</p>
  </div>
);

const ChecklistSection = ({ checklist }) => (
  <div className="bg-white p-5 rounded-2xl shadow border">
    <h2 className="text-lg font-semibold mb-4">Loan Document Checklist</h2>
    <div className="grid md:grid-cols-3 gap-4 text-sm">
      <DocListCard
        title="Required"
        color="text-indigo-700"
        bulletColor="text-indigo-500"
        items={checklist.required}
      />
      <DocListCard

```

```

        title="Missing"
        color="text-rose-700"
        bulletColor="text-rose-500"
        items={checklist.missing}
    />
    <DocListCard
        title="Not Required"
        color="text-slate-700"
        bulletColor="text-slate-400"
        items={checklist.not_required}
    />
</div>
</div>
);

const DocListCard = ({ title, color, bulletColor, items })=>(
    <div className="bg-slate-50 rounded-xl border p-4">
        <p className={` font-semibold ${color}`}>{title}</p>
        <ul className="mt-2 space-y-1">
            {items?.map((item, i)=> (
                <li key={i} className="flex gap-2 text-xs">
                    <span className={bulletColor}>•</span> {item}
                </li>
            ))}
        </ul>
    </div>
);

export default SmeAnalysisPage;

```

2.8 SMEAuth

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useState } from "react";
import { useNavigate } from "react-router-dom";

```

```

const API_BASE = "http://localhost:8081";

export default function SMEAuth() {
    const [mode, setMode] = useState("login"); // "login" | "register"
    const [form, setForm] = useState({ name: "", email: "", password: "" });
    const [loading, setLoading] = useState(false);
    const [error, setError] = useState("");
    const navigate = useNavigate();

    const handleChange = (field, value)=> {
        setForm((prev)=> ({ ...prev, [field]: value }));
    };

    const handleSubmit = async (e)=> {
        e.preventDefault();
        setError("");
        setLoading(true);
    };
}
```

```

try {
  const endpoint =
    mode === "login" ? "/api/sme/auth/login" : "/api/sme/auth/register";

  const res = await fetch(` ${API_BASE}${endpoint}` , {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({
      email: form.email,
      password: form.password,
      ...(mode === "register" ? { name: form.name } : {}),
    }),
  });

  const data = await res.json();

  if (!res.ok) {
    setError(data.error || "Authentication failed");
    setLoading(false);
    return;
  }

  // Save SME token & user in localStorage
  localStorage.setItem("smeToken", data.token);
  localStorage.setItem("smeUser", JSON.stringify(data));

  navigate("/sme");
} catch (err) {
  console.error(err);
  setError("Something went wrong. Please try again.");
} finally {
  setLoading(false);
}
};

return (
  <div className="min-h-screen flex items-center justify-center bg-[var(--bg)] text-[var(--text)]">
    <div className="card w-full max-w-md">
      <h1 className="text-2xl font-semibold mb-2 text-center">
        SME Finance Assistant
      </h1>
      <p className="text-sm opacity-75 text-center mb-6">
        Get your business loan-ready with smart guidance.
      </p>

      {/* Toggle Login / Register */}
      <div className="flex mb-4 bg-[var(--card-bg)] rounded-lg overflow-hidden border border-[var(--input-border)]">
        <button
          className={`flex-1 py-2 text-sm font-medium ${
            mode === "login"
              ? "bg-[var(--primary)] text-white"
              : "bg-transparent text-[var(--text)]"
          }`}
          onClick={() => setMode("login")}
        >
          Login
        </button>
        <button

```

```

    className={`flex-1 py-2 text-sm font-medium ${{
      mode === "register"
        ? "bg-[var(--primary)] text-white"
        : "bg-transparent text-[var(--text)]"
    }}}
    onClick={() => setMode("register")}
  >
  Register
</button>
</div>

/* Error */
{error && (
  <div className="mb-3 text-sm text-red-500 bg-red-50/80 dark:bg-red-900/30 px-3 py-2 rounded">
    {error}
  </div>
)}

/* Form */
<form onSubmit={handleSubmit} className="space-y-3">
  {mode === "register" && (
    <div>
      <label className="block text-sm mb-1">Full Name</label>
      <input
        type="text"
        value={form.name}
        onChange={(e) => handleChange("name", e.target.value)}
        placeholder="e.g., Nuwangi Traders"
      />
    </div>
  )}

  <div>
    <label className="block text-sm mb-1">Email</label>
    <input
      type="email"
      value={form.email}
      onChange={(e) => handleChange("email", e.target.value)}
      placeholder="you@business.com"
      required
    />
  </div>

  <div>
    <label className="block text-sm mb-1">Password</label>
    <input
      type="password"
      value={form.password}
      onChange={(e) => handleChange("password", e.target.value)}
      placeholder="••••••"
      required
    />
  </div>

  <button
    type="submit"
    className="btn-primary w-full mt-2"
    disabled={loading}
  >

```

```

{loading
    ? "Please wait..."
    : mode === "login"
        ? "Login to Dashboard"
        : "Create SME Account"
    </button>
</form>

/* Social buttons (UI only for now) */
<div className="mt-5">
    <div className="flex items-center gap-2 mb-3">
        <div className="h-px bg-gray-300 flex-1" />
        <span className="text-xs opacity-70">or continue with</span>
        <div className="h-px bg-gray-300 flex-1" />
    </div>

    <div className="grid grid-cols-2 gap-2">
        <button
            type="button"
            className="border border-gray-300 rounded-md py-2 text-sm flex items-center justify-center
gap-2 hover:bg-gray-50"
        >
            <span>  </span>
            <span>Microsoft</span>
        </button>
        <button
            type="button"
            className="border border-gray-300 rounded-md py-2 text-sm flex items-center justify-center
gap-2 hover:bg-gray-50"
        >
            <span>  </span>
            <span>Google</span>
        </button>
    </div>
</div>

<p className="mt-4 text-[11px] text-center opacity-60">
    By continuing, you agree to our Terms of Use and Privacy Policy.
</p>
</div>
</div>
);
}

```

2.9 SMEDashboard

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useEffect, useMemo, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import { smeBusiness, smeAnalysis } from "../../api/smeClient";

export default function SMEDashboard() {
    const navigate = useNavigate();

```

```

const { businessId } = useParams();
const smeToken = localStorage.getItem("smeToken");

const [business, setBusiness] = useState(null);
const [analysis, setAnalysis] = useState(null);
const [loading, setLoading] = useState(true);

// chat states
const [chatOpen, setChatOpen] = useState(false);
const [chatMessages, setChatMessages] = useState([]);
const [chatInput, setChatInput] = useState("");
const [chatLoading, setChatLoading] = useState(false);

useEffect(() => {
  if (!smeToken) {
    navigate("/sme/login");
    return;
  }

  (async () => {
    try {
      const [bRes, aRes] = await Promise.all([
        smeBusiness.get(businessId),
        smeAnalysis.getByBusiness(businessId),
      ]);

      setBusiness(bRes.data || null);

      // some axios calls throw on 404; if your client returns 404 as error, catch below
      if (aRes?.status === 200) setAnalysis(aRes.data);
      else setAnalysis(null);
    } catch (err) {
      console.error("Error loading SME dashboard:", err);
      setAnalysis(null);
    } finally {
      setLoading(false);
    }
  })();
}, [businessId, smeToken, navigate]);

const logout = () => {
  localStorage.removeItem("smeToken");
  localStorage.removeItem("smeUser");
  navigate("/sme/login");
};

const goBackToPortal = () => navigate("/sme/portal");

// ----- helpers -----
const safeParseJson = (str, fallback) => {
  if (!str) return fallback;
  if (typeof str === "object") return str; // if backend already returns arrays/objects
  try {
    return JSON.parse(str);
  } catch {
    return fallback;
  }
};

const strengths = useMemo(

```

```

() => safeParseJson(analysis?.strengths, []),
[analysis?.strengths]
);
const weaknesses = useMemo(
() => safeParseJson(analysis?.weaknesses, []),
[analysis?.weaknesses]
);
const improvements = useMemo(
() => safeParseJson(analysis?.improvements, []),
[analysis?.improvements]
);
const bankSuggestions = useMemo(
() => safeParseJson(analysis?.bankSuggestions, []),
[analysis?.bankSuggestions]
);
const loanPlans = useMemo(() => {
  const v = safeParseJson(analysis?.loanPlans, []);
  if (!v) return [];
  if (Array.isArray(v)) return v;
  if (typeof v === "object") return [v]; // if backend sends a single object
  return [];
}, [analysis?.loanPlans]);
const financialRatios = useMemo(
() => safeParseJson(analysis?.financialRatios, []),
[analysis?.financialRatios]
);
const documentChecklist = useMemo(
() => safeParseJson(analysis?.documentChecklist, []),
[analysis?.documentChecklist]
);
const managementAssessment = useMemo(
() => safeParseJson(analysis?.managementAssessment, {}),
[analysis?.managementAssessment]
);

const score = analysis?.finalScore ?? null;

const readinessLabel =
  score == null
    ? "No score yet"
    : score >= 80
      ? "Bank-ready"
      : score >= 60
        ? "Almost ready"
        : score >= 40
          ? "Needs work"
          : "High risk";

const improvementCount = improvements?.length ?? 0;
const weaknessCount = weaknesses?.length ?? 0;

// ----- chat handler -----
const sendChat = async () => {
  const message = chatInput.trim();
  if (!message || !business) return;

  setChatMessages((msgs) => [...msgs, { from: "user", text: message }]);
  setChatInput("");
  setChatLoading(true);

```

```

try {
  const res = await smeAnalysis.chat(business.id, message);
  const reply =
    res?.data?.reply ||
    res?.data?.text ||
    "Sorry, I couldn't generate tips right now.";

  setChatMessages((msgs) => [...msgs, { from: "assistant", text: reply }]);
} catch (err) {
  console.error("Chat error:", err);
  setChatMessages((msgs) => [
    ...msgs,
    {
      from: "assistant",
      text: "Sorry, something went wrong while generating tips. Please try again.",
    },
  ]);
} finally {
  setChatLoading(false);
}
};

if (!smeToken) return null;

if (loading) {
  return (
    <div className="min-h-screen bg-[var(--bg)] text-[var(--text)] flex items-center justify-center">
      <p className="text-sm opacity-70">Loading dashboard...</p>
    </div>
  );
}

// no analysis page
if (!analysis || !business) {
  return (
    <div className="min-h-screen bg-[var(--bg)] text-[var(--text)] flex flex-col">
      <header className="border-b border-gray-200/40 px-6 py-4 flex items-center justify-between">
        <div>
          <h1 className="text-xl font-semibold">SME Owner Dashboard</h1>
          <p className="text-xs opacity-70">No AI analysis found for this business yet.</p>
        </div>
        <div className="flex items-center gap-3">
          <button onClick={goBackToPortal} className="text-xs underline opacity-80">
            Back to Portal
          </button>
          <button onClick={logout} className="text-xs underline opacity-80">
            Logout
          </button>
        </div>
      </header>
      <main className="flex-1 flex items-center justify-center px-6">
        <div className="card max-w-xl text-center">
          <h2 className="text-lg font-semibold mb-2">No AI insights yet ⚠</h2>
          <p className="text-sm opacity-80">
            Go back to the{" "}
            <button onClick={goBackToPortal} className="underline">
              SME Portal
            </button>
            and complete the questionnaire, then click{" "}
          </p>
        </div>
      </main>
    </div>
  );
}

```

```

        <strong>Generate AI Insights</strong>.
    </p>
    </div>
</main>
</div>
);
}

return (
    <div className="min-h-screen bg-gradient-to-br from-[#f5f3ff] via-[#f9fafb] to-[#e0f2fe] text-[var(--text)] flex">
        {/* ===== LEFT SIDEBAR ===== */}
        <aside className="hidden md:flex w-60 flex-col border-r border-gray-200/60 bg-white/70 backdrop-blur-xl">
            <div className="px-5 pt-5 pb-4 border-b border-gray-200/60">
                <div className="flex items-center gap-2">
                    <div className="w-8 h-8 rounded-2xl bg-[var(--primary)] text-white flex items-center justify-center text-sm font-bold">
                        SME
                    </div>
                    <div>
                        <p className="text-xs uppercase tracking-wide opacity-60">Smart Finance</p>
                        <p className="text-sm font-semibold">Owner Cockpit</p>
                    </div>
                </div>
            </div>
        </div>

        <nav className="flex-1 px-4 py-4 space-y-6 text-xs">
            <div>
                <p className="uppercase text-[10px] mb-2 opacity-60">Overview</p>
                <SidebarItem active label="Dashboard" icon="🏠" />
                <SidebarItem label="Loan Readiness Journey" icon="⌚" onClick={goBackToPortal} />
                <SidebarItem label="Documents" icon="📁" disabled />
            </div>

            <div>
                <p className="uppercase text-[10px] mb-2 opacity-60">Business</p>
                <div className="rounded-xl bg-gray-50/80 p-3 space-y-1">
                    <p className="text-[11px] font-semibold">{business.businessName}</p>
                    <p className="text-[11px] opacity-70">
                        {business.industry} • {business.location}
                    </p>
                    <p className="text-[11px] opacity-70">Employees: {business.employees ?? "N/A"}</p>
                </div>
            </div>
        </nav>
    <div className="px-4 pb-4 border-t border-gray-200/60 text-[11px] flex items-center justify-between">
        <button onClick={logout} className="text-xs underline opacity-80">
            Logout
        </button>
        <span className="opacity-60">
            Score: <span className="font-semibold">{score ?? "--"}</span>
        </span>
    </div>
</aside>

/* ===== MAIN CONTENT ===== */
<div className="flex-1 flex flex-col">

```

```

/* Top bar */
<header className="px-6 pt-5 pb-3 flex items-center justify-between">
  <div>
    <p className="text-xs opacity-70 mb-1">Loan readiness overview</p>
    <h1 className="text-xl md:text-2xl font-semibold">
      Hello, {" "}
      <span className="text-[var(--primary)]">
        {business.ownerName || "SME Owner"}
      </span>
      . <span className="font-normal">Here is your credit health today.</span>
    </h1>
    <p className="text-[11px] opacity-60 mt-1">
      {business.businessName} • {business.industry} • {business.location}
    </p>
  </div>

  <div className="hidden md:flex items-center gap-2 text-[11px]">
    <button
      onClick={goBackToPortal}
      className="px-3 py-1.5 rounded-full border border-gray-200/60 bg-white/70 hover:bg-gray-50">
      >
        Back to Portal
      </button>
    </div>
  </div>
</header>

/* Main grid */
<main className="flex-1 px-4 md:px-6 pb-8">
  <div className="mx-auto max-w-6xl space-y-5">
    /* ROW 1 */
    <section className="grid grid-cols-1 md:grid-cols-3 gap-4">
      /* Score card */
      <div className="card flex flex-col justify-between bg-white/80">
        <div>
          <p className="text-[11px] uppercase tracking-wide opacity-60 mb-1">
            Readiness Score
          </p>
          <p className="text-sm font-semibold mb-3">Overall loan readiness</p>
        </div>
        <div className="flex items-center justify-center mb-3">
          <div className="w-28 h-28 rounded-full border-4 border-[var(--accent)] flex items-center justify-center relative">
            <span className="text-2xl font-bold">{score ?? "—"}</span>
          </div>
        </div>
        <p className="text-xs text-center opacity-80">
          Status: <span className="font-semibold">{readinessLabel}</span>
        </p>
      </div>

      /* Snapshot */
      <div className="card bg-white/80">
        <p className="text-[11px] uppercase tracking-wide opacity-60 mb-2">
          Snapshot
        </p>
        <div className="space-y-3 text-xs">
          <StatRow label="Key strengths identified" value={strengths.length} tone="good" />
          <StatRow label="Risks / weaknesses" value={weaknessCount} tone="warn" />
          <StatRow label="Action items (next 90 days)" value={improvementCount} />
        </div>
      </div>
    </section>
  </div>
</main>

```

```

        </div>
    </div>

/* Focus */
<div className="card bg-[#eef2ff]">
    <p className="text-[11px] uppercase tracking-wide opacity-60 mb-2">
        Focus for this month
    </p>
    {improvements.length === 0 ? (
        <p className="text-xs opacity-80">
            No AI action plan yet. Ask the <strong>AI Business Coach</strong>{" "}
            for suggestions.
        </p>
    ) : (
        <ol className="list-decimal pl-4 text-xs space-y-1">
            {improvements.slice(0, 4).map((imp, i) => (
                <li key={i}>{imp}</li>
            )))
        </ol>
    )}
</div>
</section>

/* ROW 2 */
<section className="grid grid-cols-1 md:grid-cols-[2fr,1.2fr] gap-4">
    <div className="card bg-white/90">
        <div className="flex items-center justify-between mb-2">
            <p className="text-sm font-semibold">AI Summary in Simple English</p>
            <span className="text-[11px] px-2 py-0.5 rounded-full bg-gray-100 text-gray-700">
                Auto-generated
            </span>
        </div>
        <p className="text-sm whitespace-pre-wrap opacity-90">
            {analysis.summary || "No summary available."}
        </p>
    </div>

    <div className="space-y-3">
        <div className="card bg-white/90">
            <p className="text-sm font-semibold mb-1">How a Bank May See You</p>
            {bankSuggestions.length === 0 ? (
                <p className="text-xs opacity-70">No bank guidance yet.</p>
            ) : (
                <ul className="list-disc pl-4 text-xs space-y-1">
                    {bankSuggestions.slice(0, 3).map((b, i) => (
                        <li key={i}>{b}</li>
                    )))
                </ul>
            )}
        </div>

        <div className="card bg-white/90">
            <p className="text-sm font-semibold mb-1">Possible Loan Structures</p>

            {loanPlans.length === 0 ? (
                <p className="text-xs opacity-70">No loan plans suggested yet.</p>
            ) : (
                <ul className="list-disc pl-4 text-xs space-y-2">
                    {loanPlans.slice(0, 3).map((lp, i) => {
                        // support different key styles just in case
                    ))}
                </ul>
            )}
        </div>
    </div>
</section>

```

```

        const shortTerm =
            lp["short-term loan plan"] ?? lp.shortTermLoanPlan ?? lp.shortTerm ???
lp.short_term;

        const longTerm =
            lp["long-term loan plan"] ?? lp.longTermLoanPlan ?? lp.longTerm ???
lp.long_term;

        // if backend sometimes returns string items
        if (typeof lp === "string") return <li key={i}>{lp}</li>;

        return (
            <li key={i}>
                {shortTerm && <div><b>Short-term:</b> {shortTerm}</div>}
                {longTerm && <div><b>Long-term:</b> {longTerm}</div>}
            </li>
        );
    )}
</ul>
)
</div>
</div>
</section>

/* ROW 3 */
<section className="grid grid-cols-1 md:grid-cols-3 gap-4">
    <div className="card bg-white">
        <p className="text-sm font-semibold mb-2">Strengths</p>
        {strengths.length === 0 ? (
            <p className="text-xs opacity-70">No strengths listed.</p>
        ) : (
            <ul className="list-disc pl-4 text-xs space-y-1">
                {strengths.map((s, i) => (
                    <li key={i}>{s}</li>
                )))
            </ul>
        )}
    </div>

    <div className="card bg-white">
        <p className="text-sm font-semibold mb-2">Risks & Weaknesses</p>
        {weaknesses.length === 0 ? (
            <p className="text-xs opacity-70">No weaknesses listed.</p>
        ) : (
            <ul className="list-disc pl-4 text-xs space-y-1">
                {weaknesses.map((w, i) => (
                    <li key={i}>{w}</li>
                )))
            </ul>
        )}
    </div>

    <div className="card bg-white">
        <p className="text-sm font-semibold mb-2">Financial Ratios (Approx.)</p>
        {financialRatios.length === 0 ? (
            <p className="text-xs opacity-70">No ratios calculated (missing data).</p>
        ) : (
            <ul className="text-xs space-y-2">
                {financialRatios.map((r, i) => (
                    <li key={i}>
                        <p className="font-medium">

```

```

                {r.name}: <span className="font-normal">{r.value}</span>
            </p>
            <p className="opacity-70">{r.interpretation}</p>
        </li>
    )})
</ul>
)}
</div>
</section>

/* ROW 4 */
<section className="grid grid-cols-1 md:grid-cols-2 gap-4">
<div className="card bg-white">
    <p className="text-sm font-semibold mb-2">Management & Governance</p>
    <div className="text-xs space-y-2">
        <p>
            <span className="font-medium">Experience: </span>
            {managementAssessment.experience || "N/A"}
        </p>
        <p>
            <span className="font-medium">Governance: </span>
            {managementAssessment.governance || "N/A"}
        </p>
        <p>
            <span className="font-medium">Risk Behaviour: </span>
            {managementAssessment.risk_behaviour || "N/A"}
        </p>
    </div>
</div>

<div className="card bg-white">
    <p className="text-sm font-semibold mb-2">Document Checklist for Bank Meeting</p>
    {documentChecklist.length === 0 ? (
        <p className="text-xs opacity-70">
            No checklist available. Ask the AI coach what documents to prepare.
        </p>
    ) : (
        <ul className="text-xs space-y-1">
            {documentChecklist.map((d, i) => (
                <li
                    key={i}
                    className="flex items-center justify-between border-b border-gray-200/30 py-1"
                >
                    <span>{d.name}</span>
                    <span className="text-[10px] px-2 py-0.5 rounded-full bg-gray-100 text-gray-700">
                        {d.importance}
                    </span>
                </li>
            )))
        </ul>
    )}
</div>
</section>
</div>
</main>

/* ===== FLOWING AI COACH ===== */
<button
    className="fixed bottom-6 right-6 rounded-full shadow-lg bg-[var(--primary)] text-white px-4 py-4">

```

```

2 text-xs flex items-center gap-2"
    onClick={() => setChatOpen((v) => !v)}
  >
  <span className="text-lg">👤</span>
  <span>AI Business Coach</span>
</button>

{chatOpen && (
  <div className="fixed bottom-20 right-6 w-80 h-96 card flex flex-col shadow-2xl bg-white">
    <div className="flex items-center justify-between mb-2">
      <div>
        <h3 className="text-sm font-semibold">AI Business Coach</h3>
        <p className="text-[10px] opacity-70">Ask for simple, practical advice.</p>
      </div>
      <button className="text-xs opacity-70" onClick={() => setChatOpen(false)}>
        X
      </button>
    </div>
    <div className="flex-1 border border-[var(--input-border)] rounded-md p-2 overflow-y-auto mb-2 bg-[var(--input-bg)] text-[11px] space-y-2">
      {chatMessages.length === 0 ? (
        <p className="opacity-60">
          Example: "How can I improve cash flow for {business.businessName}?"
        </p>
      ) : (
        chatMessages.map((m, idx) => (
          <div
            key={idx}
            className={`flex ${m.from === "user" ? "justify-end" : "justify-start"}`}
          >
            <div
              className={`px-2 py-1 rounded-md max-w-[80%] ${m.from === "user" ? "bg-[var(--primary)] text-white" : "bg-gray-100"}`}
            >
              {m.text}
            </div>
          </div>
        )))
      )
    </div>
  </div>
  <div className="flex items-center gap-2">
    <input
      className="flex-1 text-xs border border-[var(--input-border)] rounded px-2 py-1 bg-[var(--input-bg)]"
      placeholder="Type your question..."
      value={chatInput}
      onChange={(e) => setChatInput(e.target.value)}
      onKeyDown={(e) => {
        if (e.key === "Enter" && !e.shiftKey) {
          e.preventDefault();
          if (!chatLoading) sendChat();
        }
      }}
    />
    <button
      className="btn-primary text-[10px] px-3 py-1"
      onClick={sendChat}
    >

```

```

        disabled={chatLoading}
      >
        {chatLoading ? "..." : "Send"}
      </button>
    </div>
  </div>
)
</div>
);
}

/* ===== SMALL SUBCOMPONENTS ===== */

function SidebarItem({ label, icon, active, disabled, onClick }) {
  return (
    <button
      disabled={disabled}
      onClick={onClick}
      className={`w-full flex items-center gap-2 px-3 py-2 rounded-xl mb-1 text-xs ${{
        active
        ? "bg-[var(--primary)] text-white"
        : disabled
        ? "opacity-40 cursor-not-allowed"
        : "hover:bg-gray-100"
      }}`}
    >
      <span>{icon}</span>
      <span>{label}</span>
    </button>
  );
}

function StatRow({ label, value, tone }) {
  const color =
    tone === "good" ? "text-emerald-600" : tone === "warn" ? "text-amber-600" : "text-slate-700";

  return (
    <div className="flex items-center justify-between">
      <p className="text-[11px] opacity-80">{label}</p>
      <span className={`text-xs font-semibold ${color}`}>{value}</span>
    </div>
  );
}

```

2.10 SME Landing

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { motion } from "framer-motion";
import { useMemo, useState } from "react";
import { useNavigate } from "react-router-dom";
import GuideBookModal from "../../components/GuideBookModal"; // adjust path if needed

export default function SME Landing() {

```

```

const navigate = useNavigate();
const [openGuide, setOpenGuide] = useState(false);

const features = useMemo(
  () => [
    {
      title: "AI-powered SME insights",
      desc: "Clear strengths, weaknesses, and practical actions to become loan-ready.",
    },
    {
      title: "Automated readiness scoring",
      desc: "A simple score to understand risk level and next steps.",
    },
    {
      title: "Bank-ready guidance",
      desc: "Suggested loan structures and a document checklist for your meeting.",
    },
  ],
  []
);

return (
  <div className="min-h-screen bg-white text-gray-900 relative">
    <SoftGridBackground />

    {/* Top bar */}
    <header className="relative z-10">
      <div className="max-w-6xl mx-auto px-6 py-5 flex items-center justify-between">
        <div className="flex items-center gap-3">
          <div className="w-9 h-9 rounded-2xl border border-gray-200 bg-gray-50 flex items-center justify-center font-bold text-sm">
            SME
          </div>
          <div>
            <p className="text-[11px] uppercase tracking-wide text-gray-500">
              Smart Finance
            </p>
            <p className="text-sm font-semibold">Loan Readiness Portal</p>
          </div>
        </div>
      </div>

      <div className="flex items-center gap-2">
        <button
          onClick={() => setOpenGuide(true)}
          className="px-4 py-2 rounded-xl border border-gray-200 bg-white hover:bg-gray-50 text-sm"
        >
           Guide Book
        </button>

        <button
          onClick={() => navigate("/sme/login")}
          className="px-4 py-2 rounded-xl border border-gray-200 bg-white hover:bg-gray-50 text-sm"
        >
          Login
        </button>

        <button
          onClick={() => navigate("/sme/register")}
          className="px-4 py-2 rounded-xl border border-gray-200 bg-white hover:bg-gray-50 text-sm"
        >
        </button>
      </div>
    </header>
  </div>
);

```

```

        className="px-4 py-2 rounded-xl bg-black text-white hover:opacity-90 text-sm"
    >
        Get Started →
    </button>
</div>
</div>
</header>

{/* Main */}
<main className="relative z-10">
    <div className="max-w-6xl mx-auto px-6 py-14 md:py-20">
        <div className="grid md:grid-cols-2 gap-10 items-start">
            {/* Left: hero */}
            <div>
                <div className="inline-flex items-center gap-2 px-3 py-1.5 rounded-full border border-gray-200 bg-gray-50 text-xs text-gray-700">
                    <span className="font-semibold">New</span>
                    <span className="text-gray-500">AI insights for bank readiness</span>
                </div>
            <h1 className="mt-5 text-4xl md:text-5xl font-extrabold leading-tight">
                Make your SME{" "}
                <span className="text-gray-900 underline-decoration-gray-300 decoration-4 underline-offset-4">
                    loan-ready
                </span>{" "}
                with a simple step-by-step journey.
            </h1>
            <p className="mt-5 text-base md:text-lg text-gray-600 max-w-xl">
                Create your business profile, complete a quick assessment, and receive
                clear recommendations and a readiness score to prepare for a bank meeting.
            </p>
            <div className="mt-8 flex flex-wrap gap-3">
                <button
                    onClick={() => navigate("/sme/register")}
                    className="px-6 py-3 rounded-2xl bg-black text-white font-semibold hover:opacity-90">
                    Get Started →
                </button>
                <button
                    onClick={() => setOpenGuide(true)}
                    className="px-6 py-3 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50 font-semibold">
                    View Guide Book
                </button>
            </div>
            {/* Mini steps */}
            <div className="mt-10 grid grid-cols-1 sm:grid-cols-3 gap-3">
                <StepCard n="1" title="Business Profile" />
                <StepCard n="2" title="Questionnaire" />
                <StepCard n="3" title="AI Insights" />
            </div>
        </div>
        {/* Right: clean info card */}
    </main>

```

```

<div className="rounded-3xl border border-gray-200 bg-white shadow-sm p-6 md:p-7">
  <p className="text-sm font-semibold">What you will get</p>

  <ul className="mt-4 space-y-3 text-sm text-gray-700">
    <li>Bullet text="Readiness Score + simple status (Bank-ready / Almost ready / Needs work)"</li>
    <li>Bullet text="Strengths, weaknesses, and 90-day improvement plan" />
    <li>Bullet text="How a bank may see your business (practical tips)" />
    <li>Bullet text="Suggested loan structures (short-term / long-term)" />
    <li>Bullet text="Document checklist for your bank meeting" />
  </ul>

  <div className="mt-6 p-4 rounded-2xl bg-gray-50 border border-gray-200">
    <p className="text-xs text-gray-600">
      Tip: Use real numbers and answer honestly. You can regenerate insights after you
    improve.
    </p>
    </div>
    </div>
  </div>

  /* Feature cards */
  <div className="mt-12 grid md:grid-cols-3 gap-6">
    {features.map((f) => (
      <motion.div
        key={f.title}
        initial={{ opacity: 0, y: 14 }}
        animate={{ opacity: 1, y: 0 }}
        transition={{ duration: 0.5 }}
        className="p-6 rounded-3xl bg-white border border-gray-200 shadow-sm"
      >
        <h3 className="text-lg font-semibold">{f.title}</h3>
        <p className="text-gray-600 text-sm mt-2">{f.desc}</p>
      </motion.div>
    )));
    </div>
  </div>
</main>

  /* Guide modal */
  <GuideBookModal open={openGuide} onClose={() => setOpenGuide(false)} />
</div>
);
}

/* ----- Small components ----- */

function SoftGridBackground() {
  return (
    <>
      <div className="absolute inset-0 bg-white" />
      <div
        className="absolute inset-0 opacity-[0.35]"
        style={{
          backgroundImage:
            "linear-gradient(to right, rgba(0,0,0,0.04) 1px, transparent 1px), linear-gradient(to bottom, rgba(0,0,0,0.04) 1px, transparent 1px)",
          backgroundSize: "44px 44px",
        }}
      />
    </>
  );
}

```

```

        <div className="absolute -top-40 -right-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-3xl" />
        <div className="absolute -bottom-40 -left-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-3xl" />
      </>
    );
  }

function StepCard({ n, title }) {
  return (
    <div className="rounded-2xl border border-gray-200 bg-white p-4">
      <p className="text-xs text-gray-500">Step {n}</p>
      <p className="text-sm font-semibold mt-1">{title}</p>
    </div>
  );
}

function Bullet({ text }) {
  return (
    <li className="flex items-start gap-2">
      <span className="mt-1 inline-block w-2 h-2 rounded-full bg-gray-900" />
      <span>{text}</span>
    </li>
  );
}

```

2.11 SMELogin

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import { motion } from "framer-motion";
import { smeAuth } from "../../api/smeClient";

export default function SMELogin() {
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();

  async function handleLogin(e) {
    e.preventDefault();
    setLoading(true);

    const email = e.target.email.value;
    const password = e.target.password.value;

    try {
      const res = await smeAuth.login({ email, password });
      localStorage.setItem("smeToken", res.data.token);
      localStorage.setItem("smeUser", res.data.name);
      navigate("/sme/portal");
    } catch (err) {
      console.error(err);
      alert(err.response?.data?.error || "Login failed");
    } finally {

```

```

        setLoading(false);
    }
}

return (
    <div className="min-h-screen bg-white text-gray-900 relative overflow-hidden">
        <SoftGridBackground />

        <div className="relative z-10 min-h-screen grid md:grid-cols-2">
            {/* Left info panel */}
            <section className="hidden md:flex items-center justify-center p-12">
                <div className="max-w-md">
                    <div className="inline-flex items-center gap-2 px-3 py-1.5 rounded-full border border-gray-200 bg-gray-50 text-xs text-gray-700">
                        <span className="font-semibold">SME</span>
                        <span className="text-gray-500">Smart Finance Dashboard</span>
                    </div>

                    <h1 className="mt-5 text-4xl font-extrabold leading-tight">
                        Welcome back to your{" "}
                        <span className="underline decoration-gray-300 decoration-4 underline-offset-4">
                            loan readiness
                        </span>{" "}
                        journey.
                    </h1>

                    <p className="mt-4 text-sm text-gray-600">
                        Login to review your AI insights, readiness score, and document checklist for your bank
                        meeting.
                    </p>

                    <div className="mt-8 space-y-3">
                        <MiniItem title="Readiness Score" desc="Understand your current position quickly." />
                        <MiniItem title="Action Plan" desc="90-day improvements to become bank-ready." />
                        <MiniItem title="Documents" desc="Checklist to prepare before the meeting." />
                    </div>

                    <button
                        onClick={() => navigate("/sme")}
                        className="mt-10 text-sm underline text-gray-700 hover:text-gray-900"
                    >
                        ← Back to Home
                    </button>
                </div>
            </section>

            {/* Right login card */}
            <section className="flex items-center justify-center p-6 md:p-12">
                <motion.div
                    initial={{ opacity: 0, y: 18 }}
                    animate={{ opacity: 1, y: 0 }}
                    transition={{ duration: 0.5 }}
                    className="w-full max-w-md"
                >
                    <div className="rounded-3xl border border-gray-200 bg-white shadow-sm p-7 md:p-8">
                        <div className="flex items-start justify-between gap-3">
                            <div>
                                <h2 className="text-2xl font-bold">Login</h2>
                                <p className="text-sm text-gray-600 mt-1">
                                    Access your SME dashboard securely.
                                </p>
                            </div>
                        </div>
                    </div>
                </motion.div>
            </section>
        </div>
    </div>
)

```

```

        </p>
    </div>

    <div className="w-10 h-10 rounded-2xl border border-gray-200 bg-gray-50 flex items-center justify-center font-bold text-sm">
        SF
    </div>
    </div>

<form onSubmit={handleLogin} className="mt-6 space-y-4">
    <Field
        name="email"
        label="Email"
        type="email"
        placeholder="you@business.com"
    />
    <Field
        name="password"
        label="Password"
        type="password"
        placeholder="*****"
    />

    <button
        type="submit"
        disabled={loading}
        className="w-full rounded-2xl py-3 font-semibold text-white bg-black hover:opacity-90 transition disabled:opacity-50"
    >
        {loading ? "Logging in..." : "Login"}
    </button>

    <button
        type="button"
        onClick={() => navigate("/sme/register")}
        className="w-full rounded-2xl py-3 font-semibold border border-gray-200 bg-white hover:bg-gray-50 transition"
    >
        Create an account
    </button>
</form>

<p className="text-xs text-gray-500 mt-5">
    By logging in, you agree to use this system for SME readiness evaluation purposes.
</p>
</div>
</motion.div>
</section>
</div>
</div>
);
}

/*
----- small components -----
*/

function Field({ name, label, type, placeholder }) {
    return (
        <label className="block">
            <span className="text-sm text-gray-700">{label}</span>
            <input

```

```

        name={name}
        type={type}
        placeholder={placeholder}
        required
        className="mt-2 w-full rounded-2xl bg-white text-gray-900 px-4 py-3
        border border-gray-200 focus:ring-2 focus:ring-gray-300 outline-none"
        />
    </label>
);
}

function MiniItem({ title, desc }) {
    return (
        <div className="rounded-2xl border border-gray-200 bg-white p-4">
            <p className="text-sm font-semibold">{title}</p>
            <p className="text-xs text-gray-600 mt-1">{desc}</p>
        </div>
    );
}

function SoftGridBackground() {
    return (
        <>
        <div className="absolute inset-0 bg-white" />
        <div
            className="absolute inset-0 opacity-[0.35]"
            style={{
                backgroundImage:
                    "linear-gradient(to right, rgba(0,0,0,0.04) 1px, transparent 1px), linear-gradient(to bottom,
                    rgba(0,0,0,0.04) 1px, transparent 1px)",
                backgroundSize: "44px 44px",
            }}
        />
        <div className="absolute -top-40 -right-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-
        3xl" />
        <div className="absolute -bottom-40 -left-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-
        3xl" />
        </>
    );
}

```

2.12 SMEPortal

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";
import {
    smeBusiness,
    smeQuestionnaire,
    smeAnalysis,
} from "../../../api/smeClient";

// ----- JSON helpers for AI Insights -----
const parseJsonArray = (value) => {

```

```

if (!value) return [];
if (Array.isArray(value)) return value;
try {
  const parsed = JSON.parse(value);
  return Array.isArray(parsed) ? parsed : [];
} catch (e) {
  return [];
}
};

const parseJsonObject = (value) => {
  if (!value) return {};
  if (typeof value === "object") return value;
  try {
    const parsed = JSON.parse(value);
    return typeof parsed === "object" && parsed !== null ? parsed : {};
  } catch (e) {
    return {};
  }
};

/* =====
   MAIN PORTAL COMPONENT
===== */
export default function SMEPortal() {
  const navigate = useNavigate();
  const smeToken = localStorage.getItem("smeToken");

  const [profile, setProfile] = useState(null);
  const [businesses, setBusinesses] = useState([]);
  const [selectedBusiness, setSelectedBusiness] = useState(null);

  const [step, setStep] = useState(1); // 1 = business • 2 = questionnaire • 3 = AI analysis
  const [questionPage, setQuestionPage] = useState(1); // for multi-page questionnaire
  const [questionnaireCompleted, setQuestionnaireCompleted] = useState(false);

  const [businessForm, setBusinessForm] = useState({
    businessName: "",
    industry: "",
    employees: "",
    annualRevenue: "",
    location: "",
    bankRelationship: ""
  });

  const [questions, setQuestions] = useState([]);
  const [answers, setAnswers] = useState([]);
  const [analysis, setAnalysis] = useState(null);
  const [loadingAnalysis, setLoadingAnalysis] = useState(false);

  /* =====
     LOAD SME PROFILE + BUSINESSES ON MOUNT
===== */
  useEffect(() => {
    if (!smeToken) {
      navigate("/sme/login");
      return;
    }

    (async () => {

```

```

        await loadProfile();
        await loadBusinesses();
    })());
    // eslint-disable-next-line react-hooks/exhaustive-deps
}, [smeToken]);

// When entering step 2, always start at page 1
useEffect(() => {
    if (step === 2) {
        setQuestionPage(1);
    }
}, [step]);

/* =====
API CALLS
===== */

// Load SME profile
const loadProfile = async () => {
    try {
        const res = await fetch("http://localhost:8081/api/sme/profile/me", {
            headers: {
                Authorization: `Bearer ${smeToken}`,
            },
        });

        const data = await res.json();
        setProfile(data);
    } catch (err) {
        console.error("Error loading profile:", err);
    }
};

// Load all businesses
const loadBusinesses = async () => {
    try {
        const res = await smeBusiness.list();
        const data = res.data || [];
        setBusinesses(data);

        if (data.length > 0) {
            setSelectedBusiness(data[0]);
        } else {
            setSelectedBusiness(null);
        }
    } catch (err) {
        console.error("Error loading businesses:", err);
    }
};

// Watch selected business → check questionnaire + existing analysis
useEffect(() => {
    if (!selectedBusiness) {
        setQuestionnaireCompleted(false);
        setAnalysis(null);
        return;
    }
});

// 1) check if questionnaire already completed
(async () => {

```

```

try {
  const qcRes = await smeQuestionnaire.checkCompleted(
    selectedBusiness.id
  );
  const qcData = qcRes.data;
  setQuestionnaireCompleted(!qcData.completed);
} catch (err) {
  console.error("Error checking questionnaire completion:", err);
  setQuestionnaireCompleted(false);
}
})();

// 2) try load existing analysis
(async () => {
  try {
    const aRes = await smeAnalysis.getByBusiness(selectedBusiness.id);
    if (aRes.status === 200) {
      setAnalysis(aRes.data);
    } else {
      setAnalysis(null);
    }
  } catch {
    setAnalysis(null);
  }
})();
}, [selectedBusiness]);

// Create new business
const handleBusinessSubmit = async (e) => {
  e.preventDefault();

  try {
    const payload = {
      ...businessForm,
      employees: Number(businessForm.employees),
      annualRevenue: Number(businessForm.annualRevenue),
    };
    const res = await smeBusiness.create(payload);
    const data = res.data;

    if (data.error) {
      alert(data.error);
      return;
    }

    setBusinessForm({
      businessName: "",
      industry: "",
      employees: "",
      annualRevenue: "",
      location: "",
      bankRelationship: "",
    });
  }

  await loadBusinesses();
  setStep(2);
} catch (err) {
  console.error(err);
  alert("Something went wrong while saving your business.");
}

```

```

        }

    };

// ● Delete existing business
const handleDeleteBusiness = async (id) => {
    if (!window.confirm("Are you sure you want to delete this business?")) {
        return;
    }
    try {
        await smeBusiness.remove(id);
        await loadBusinesses();
    } catch (err) {
        console.error("Error deleting business:", err);
        alert("Failed to delete business.");
    }
};

// Load questionnaire questions
const loadQuestions = async () => {
    try {
        const res = await smeQuestionnaire.getQuestions();
        const data = res.data;
        const qs = data || [];
        setQuestions(qs);
        setAnswers(new Array(qs.length).fill(""));
    } catch (err) {
        console.error("Error loading questions:", err);
    }
};

// Load questions when step enters 2 (only if not completed)
useEffect(() => {
    if (step === 2 && !questionnaireCompleted && questions.length === 0) {
        void loadQuestions();
    }
}, [step, questionnaireCompleted, questions.length]);

// Submit questionnaire answers (final, after last page)
const submitAnswers = async () => {
    if (!selectedBusiness) {
        alert("Please select or create a business first.");
        return;
    }

    try {
        const res = await smeQuestionnaire.submit(
            selectedBusiness.id,
            answers
        );
        const data = res.data;

        if (res.status >= 400 || data.error) {
            alert(data.error || "Failed to submit answers");
            return;
        }
    }

    alert("Questionnaire submitted. Generating AI insights...");

    setQuestionnaireCompleted(true);
    setStep(3);
}

```

```

        await generateAnalysis();
    } catch (err) {
        console.error(err);
        alert("Error submitting questionnaire.");
    }
};

// Generate AI analysis
const generateAnalysis = async () => {
    if (!selectedBusiness) return;

    setLoadingAnalysis(true);

    try {
        const res = await smeAnalysis.generate(selectedBusiness.id);
        const data = res.data;

        if (res.status >= 400 || data.error) {
            alert(data.error || "Failed to generate analysis");
            return;
        }

        setAnalysis(data);
    } catch (err) {
        console.error(err);
    } finally {
        setLoadingAnalysis(false);
    }
};

const logout = () => {
    localStorage.removeItem("smeToken");
    localStorage.removeItem("smeUser");
    navigate("/sme/login");
};

if (!smeToken) return null;

/* =====
 * RENDER
 * ===== */
return (
    <div className="min-h-screen bg-[var(--bg)] text-[var(--text)]">
        {/* ===== Top Bar ===== */}
        <header className="border-b border-gray-200/40 px-6 py-4 flex items-center justify-between">
            <div>
                <h1 className="text-xl font-semibold">
                    SME Loan Readiness Portal
                </h1>
                <p className="text-xs opacity-70">
                    Welcome, {profile?.name || "SME User"} — let's get your
                    business loan-ready.
                </p>
            </div>

            <div className="flex items-center gap-3">
                <span className="text-xs opacity-80">
                    {profile?.email} • {profile?.role}
                </span>
                <button>

```

```

    onClick={logout}
    className="text-xs underline opacity-80"
  >
  Logout
</button>
</div>
</header>

{/* ===== Layout ===== */}
<main className="dashboard grid grid-cols-[260px,1fr] gap-6">
  {/* ----- STEP SIDEBAR ----- */}
  <aside className="space-y-4 mt-4">
    <StepSidebar
      step={step}
      setStep={setStep}
      hasBusiness={businesses.length > 0}
      analysis={analysis}
      questionnaireCompleted={questionnaireCompleted}
    />

    {/* Business List */}
    {businesses.length > 0 && (
      <BusinessSelector
        businesses={businesses}
        selectedBusiness={selectedBusiness}
        setSelectedBusiness={setSelectedBusiness}
        onDeleteBusiness={handleDeleteBusiness}
      />
    )}
  </aside>

  {/* ----- MAIN CONTENT ----- */}
  <section className="mt-4 space-y-6">
    {step === 1 && (
      <BusinessFormStep
        form={businessForm}
        setForm={setBusinessForm}
        onSubmit={handleBusinessSubmit}
        selectedBusiness={selectedBusiness}
        hasBusiness={businesses.length > 0}
      />
    )}

    {step === 2 && (
      <QuestionnaireStep
        questions={questions}
        answers={answers}
        setAnswers={setAnswers}
        page={questionPage}
        setPage={setQuestionPage}
        onSubmit={submitAnswers}
        disabled={!selectedBusiness}
        questionnaireCompleted={questionnaireCompleted}
      />
    )}

    {step === 3 && (
      <AnalysisStep
        analysis={analysis}
        selectedBusiness={selectedBusiness}
      />
    )}
  </section>
</main>

```

```

        loading={loadingAnalysis}
        onRegenerate={generateAnalysis}
        onOpenDashboard={() =>
            navigate('/sme/dashboard/${selectedBusiness.id}')
        }
    />
)
</section>
</main>
</div>
);
}

/* =====
   SUBCOMPONENTS
===== */
function StepSidebar({
    step,
    setStep,
    hasBusiness,
    analysis,
    questionnaireCompleted,
}) {
    return (
        <div className="card">
            <h2 className="text-sm font-semibold mb-2">
                Loan Readiness Journey
            </h2>
            <div className="space-y-2 text-sm">
                <StepItem
                    label="1. Business Profile"
                    active={step === 1}
                    done={hasBusiness}
                    onClick={() => setStep(1)}
                />
                <StepItem
                    label="2. Questionnaire"
                    active={step === 2}
                    done={questionnaireCompleted}
                    onClick={() => setStep(2)}
                />
                <StepItem
                    label="3. AI Insights"
                    active={step === 3}
                    done={!analysis}
                    onClick={() => setStep(3)}
                />
            </div>
        </div>
    );
}

function StepItem({ label, active, done, onClick }) {
    return (
        <button
            onClick={onClick}
            className={`w-full flex items-center justify-between px-3 py-2 rounded text-xs ${{
                active
                ? "bg-[var(--primary)] text-white"
            }}`}
        >

```

```

        : "hover:bg-gray-200/40"
    `}`);
  >
  <span>{label}</span>
  <span>{done ? "✓" : active ? "●" : "→"}</span>
</button>
);
}

function BusinessSelector({
  businesses,
  selectedBusiness,
  setSelectedBusiness,
  onDeleteBusiness,
}) {
  return (
    <div className="card">
      <h3 className="text-sm font-semibold mb-2">Your Businesses</h3>
      <div className="space-y-1 text-sm">
        {businesses.map((b) => (
          <div
            key={b.id}
            className={`flex items-center justify-between px-2 py-1 rounded ${selectedBusiness?.id === b.id
              ? "bg-[var(--primary)] text-white"
              : "hover:bg-gray-200/50"}`}
          `)
        >
          {/* click the name → select */}
          <button
            type="button"
            onClick={() => setSelectedBusiness(b)}
            className="flex-1 text-left"
          >
            {b.businessName}
          </button>

          {/* delete icon */}
          <button
            type="button"
            className="ml-2 text-[11px] opacity-80 hover:opacity-100"
            onClick={(e) => {
              e.stopPropagation();
              onDeleteBusiness && onDeleteBusiness(b.id);
            }}
            title="Delete this business"
          >
            🗑
          </button>
        </div>
      )));
    </div>
  );
}

function BusinessFormStep({
  form,
  setForm,
  onSubmit,

```

```

        hasBusiness,
        selectedBusiness,
    }) {
return (
<div className="card">
    <h2 className="text-lg font-semibold mb-1">Business Profile</h2>
    <p className="text-sm opacity-70 mb-4">
        Tell us about your business. This helps us calculate your
        readiness score.
    </p>
{hasBusiness && selectedBusiness && (
    <div className="mb-4 p-3 rounded border border-[var(--input-border)] text-sm">
        <div className="font-medium mb-1">
            Current: {selectedBusiness.businessName}
        </div>
        <div className="opacity-80">
            Industry: {selectedBusiness.industry} • Employees:{" "}
            {selectedBusiness.employees} • Revenue:{" "}
            {selectedBusiness.annualRevenue} LKR/year
        </div>
    </div>
)}
<form
    onSubmit={onSubmit}
    className="grid grid-cols-1 md:grid-cols-2 gap-3 mt-2"
>
<Field
    label="Business Name"
    value={form.businessName}
    onChange={(v) =>
        setForm((f) => ({ ...f, businessName: v }))}
    >
    required
/>
<Field
    label="Industry"
    value={form.industry}
    onChange={(v) =>
        setForm((f) => ({ ...f, industry: v }))}
    >
    required
/>
<Field
    label="Employees"
    type="number"
    value={form.employees}
    onChange={(v) =>
        setForm((f) => ({ ...f, employees: v }))}
    >
    required
/>
<Field
    label="Annual Revenue (LKR)"
    type="number"
    value={form.annualRevenue}
    onChange={(v) =>
        setForm((f) => ({ ...f, annualRevenue: v }))}
    >
}

```

```

        required
    />
<Field
    label="Location"
    value={form.location}
    onChange={(v) =>
        setForm((f) => ({ ...f, location: v }))
    }
    required
/>
<Field
    label="Bank Relationship"
    value={form.bankRelationship}
    onChange={(v) =>
        setForm((f) => ({ ...f, bankRelationship: v }))
    }
/>

<div className="md:col-span-2 flex justify-end mt-2">
    <button type="submit" className="btn-primary">
        Continue →
    </button>
</div>
</form>
</div>
);
}

function Field({ label, value, onChange, type = "text", required }) {
    return (
        <div>
            <label className="block text-xs mb-1">{label}</label>
            <input
                type={type}
                value={value}
                onChange={(e) => onChange(e.target.value)}
                placeholder={`Enter ${label}`}
                required={required}
            />
        </div>
    );
}

function QuestionnaireStep({
    questions,
    answers,
    setAnswers,
    page,
    setPage,
    onSubmit,
    disabled,
    questionnaireCompleted,
}) {
if(disabled) {
    return (
        <div className="card">
            <p className="text-sm">Please create a business profile first.</p>
        </div>
    );
}

```

```

// If already completed, just show message
if (questionnaireCompleted) {
    return (
        <div className="card">
            <h2 className="text-lg font-semibold mb-2">
                Business Questionnaire
            </h2>
            <p className="text-sm opacity-70">
                You have already submitted the questionnaire for this
                business. You can go to <strong>AI Insights</strong> to view
                or regenerate your report.
            </p>
        </div>
    );
}

const pageSize = 3;
const totalPages = Math.max(
    1,
    Math.ceil((questions || []).length / pageSize)
);
const currentPage = Math.min(page, totalPages);

const startIndex = (currentPage - 1) * pageSize;
const endIndex = startIndex + pageSize;
const pageQuestions = questions.slice(startIndex, endIndex);

const handleChangeAnswer = (globalIndex, value) => {
    const updated = [...answers];
    updated[globalIndex] = value;
    setAnswers(updated);
};

const handleNext = () => {
    let allFilled = true;
    for (let i = startIndex; i < Math.min(endIndex, questions.length); i++) {
        if (!answers[i] || answers[i].trim() === "") {
            allFilled = false;
            break;
        }
    }
    if (!allFilled) {
        if (
            !window.confirm(
                "Some answers are empty. Continue anyway?"
            )
        ) {
            return;
        }
    }
    if (currentPage < totalPages) {
        setPage(currentPage + 1);
    }
};

const handleBack = () => {
    if (currentPage > 1) {
        setPage(currentPage - 1);
    }
};

```

```

};

const handleSubmit = () => {
    onSubmit();
};

return (
    <div className="card">
        <h2 className="text-lg font-semibold mb-1">
            Business Questionnaire
        </h2>
        <p className="text-sm opacity-70 mb-2">
            Answer each question honestly to help us evaluate risks and
            strengths.
        </p>
        <p className="text-xs opacity-70 mb-4">
            Page {currentPage} of {totalPages}
        </p>

        {questions.length === 0 ? (
            <p>Loading questions...</p>
        ) : (
            <div className="space-y-3 max-h-[60vh] overflow-auto pr-2">
                {pageQuestions.map((q, idx) => {
                    const globalIndex = startIndex + idx;
                    return (
                        <div
                            key={globalIndex}
                            className="border border-[var(--input-border)] rounded-lg p-3"
                        >
                            <p className="text-sm font-medium mb-2">
                                Q{globalIndex + 1}. {q}
                            </p>
                            <textarea
                                className="w-full text-sm rounded border border-[var(--input-border)] p-2 bg-[var(--input-bg)]"
                                rows={3}
                                value={answers[globalIndex] || ""}
                                onChange={(e) =>
                                    handleChangeAnswer(
                                        globalIndex,
                                        e.target.value
                                    )
                                }
                                placeholder="Type your answer here..."
                            />
                        </div>
                    );
                ))}
            </div>
        )}
    <div className="flex justify-between items-center mt-4">
        <button
            className="btn-secondary text-xs px-3 py-2"
            onClick={handleBack}
            disabled={currentPage === 1}
        >
            ← Back
        </button>
    </div>
)

```

```

    {currentPage < totalPages ? (
      <button
        className="btn-primary text-xs px-3 py-2"
        onClick={handleNext}
      >
        Next →
      </button>
    ) : (
      <button
        className="btn-primary text-xs px-3 py-2"
        onClick={handleSubmit}
      >
        Submit & Generate AI Insights →
      </button>
    )}
  </div>
</div>
);
}

function AnalysisStep({
  analysis,
  selectedBusiness,
  loading,
  onRegenerate,
  onOpenDashboard,
}) {
  if (!selectedBusiness) {
    return (
      <div className="card">
        <p className="text-sm">
          Select or create a business before viewing AI insights.
        </p>
      </div>
    );
  }

  // Safely parse JSON fields coming from backend (TEXT columns)
  const strengths = parseJsonArray(analysis?.strengths);
  const weaknesses = parseJsonArray(analysis?.weaknesses);
  const improvements = parseJsonArray(analysis?.improvements);
  const financialRatios = parseJsonArray(analysis?.financialRatios);
  const managementAssessment = parseJsonObject(
    analysis?.managementAssessment
  );

  return (
    <div className="card">
      {/* Header */}
      <div className="flex items-center justify-between mb-3">
        <div>
          <h2 className="text-lg font-semibold">
            AI Readiness Insights
          </h2>
          <p className="text-xs opacity-70">
            Business: {selectedBusiness.businessName}
          </p>
        </div>
      <div className="flex gap-2">

```

```

<button
  className="btn-secondary text-xs px-3 py-2"
  onClick={onOpenDashboard}
  disabled={!analysis}
>
  Open SME Dashboard
</button>
<button
  className="btn-primary text-xs px-3 py-2"
  onClick={onRegenerate}
  disabled={loading}
>
  {loading ? "Recalculating..." : "Recalculate"}
</button>
</div>
</div>

/* Body */
{!analysis ? (
  <p className="text-sm">
    No analysis yet. Click <strong>Recalculate</strong> to
    generate insights.
  </p>
) : (
  <div className="grid grid-cols-1 md:grid-cols-3 gap-4 mt-3">
    /* ===== SCORE ===== */
    <div className="card">
      <h3 className="text-sm font-semibold mb-2">
        Readiness Score
      </h3>
      <div className="flex items-center justify-center py-4">
        <div className="w-28 h-28 rounded-full border-4 border-[var(--accent)] flex items-center
justify-center">
          <span className="text-2xl font-bold">
            {analysis?.finalScore ?? "—"}
          </span>
        </div>
      </div>
    </div>

    /* Optional quick view of management assessment */
    <div className="mt-3 text-xs opacity-80 space-y-1">
      {managementAssessment.experience && (
        <p>
          <span className="font-medium">
            Management:
          </span>{" "}
          {managementAssessment.experience}
        </p>
      )}
      {managementAssessment.risk_behaviour && (
        <p>
          <span className="font-medium">
            Risk Behaviour:
          </span>{" "}
          {managementAssessment.risk_behaviour}
        </p>
      )}
    </div>
  </div>
)

```

```

/* ===== STRENGTHS ===== */
<div className="card">
  <h3 className="text-sm font-semibold mb-2">
    Strengths
  </h3>

  <ul className="mt-1 space-y-1 text-xs text-slate-700 list-disc list-inside">
    {strengths.length === 0 && (
      <li>No strengths detected yet.</li>
    )}
    {strengths.map((s, i) => (
      <li key={i}>{s}</li>
    )))
  </ul>

/* Optional ratios snippet */
{financialRatios.length > 0 && (
  <div className="mt-3">
    <p className="text-xs font-semibold mb-1">
      Key Financial Ratio
    </p>
    <p className="text-[11px] opacity-80">
      {financialRatios[0].name}:{" "}
      {financialRatios[0].value} -{" "}
      {financialRatios[0].interpretation}
    </p>
  </div>
)}
</div>

/* ===== RISKS & ACTIONS ===== */
<div className="card">
  <h3 className="text-sm font-semibold mb-2">
    Risks & Actions
  </h3>

  <p className="text-xs font-medium mb-1">
    Weaknesses
  </p>
  <ul className="mt-1 space-y-1 text-xs text-slate-700 list-disc list-inside mb-2">
    {weaknesses.length === 0 && (
      <li>No weaknesses recorded.</li>
    )}
    {weaknesses.map((w, i) => (
      <li key={i}>{w}</li>
    )))
  </ul>

  <p className="text-xs font-medium mb-1">
    Recommendations
  </p>
  <ul className="mt-1 space-y-1 text-xs text-slate-700 list-disc list-inside">
    {improvements.length === 0 && (
      <li>No recommendations yet.</li>
    )}
    {improvements.map((imp, i) => (
      <li key={i}>{imp}</li>
    )))
  </ul>
</div>

```

```

        </div>
    )}
</div>
);
}
}

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import { motion } from "framer-motion";
import { smeAuth } from "../../api/smeClient";

export default function SMERegister() {
    const [loading, setLoading] = useState(false);
    const navigate = useNavigate();

    async function handleRegister(e) {
        e.preventDefault();
        setLoading(true);

        const name = e.target.name.value;
        const email = e.target.email.value;
        const password = e.target.password.value;
        const confirm = e.target.confirm.value;

        if (password !== confirm) {
            alert("Passwords do not match");
            setLoading(false);
            return;
        }

        try {
            await smeAuth.register({ name, email, password });
            alert("Registration successful. Please login.");
            navigate("/sme/login");
        } catch (err) {
            console.error(err);
            alert(err.response?.data?.error || "Registration failed");
        } finally {
            setLoading(false);
        }
    }

    return (
        <div className="min-h-screen bg-white text-gray-900 relative overflow-hidden">
            <SoftGridBackground />

            <div className="relative z-10 min-h-screen grid md:grid-cols-2">
                {/* Left info panel */}
                <section className="hidden md:flex items-center justify-center p-12">
                    <div className="max-w-md">
                        <div className="inline-flex items-center gap-2 px-3 py-1.5 rounded-full border border-gray-300">

```

```

200 bg-gray-50 text-xs text-gray-700">
    <span className="font-semibold">SME</span>
    <span className="text-gray-500">Smart Finance Dashboard</span>
</div>

<h1 className="mt-5 text-4xl font-extrabold leading-tight">
    Create your account and start the {" "}
    <span className="underline decoration-gray-300 decoration-4 underline-offset-4">
        guide-book journey
    </span>
    .
</h1>

<p className="mt-4 text-sm text-gray-600">
    After registration, you'll create a business profile, complete a short assessment,
    and generate AI insights with a bank-ready checklist.
</p>

<div className="mt-8 space-y-3">
    <MiniItem title="Step 1: Create Profile" desc="Basic business details for analysis." />
    <MiniItem title="Step 2: Answer Questions" desc="Quick questionnaire to assess readiness." />
    <MiniItem title="Step 3: Get Insights" desc="Score + improvements + documents." />
</div>

<button
    onClick={() => navigate("/sme")}
    className="mt-10 text-sm underline text-gray-700 hover:text-gray-900"
>
    ← Back to Home
</button>
</div>
</section>

/* Right register card */
<section className="flex items-center justify-center p-6 md:p-12">
    <motion.div
        initial={{ opacity: 0, y: 18 }}
        animate={{ opacity: 1, y: 0 }}
        transition={{ duration: 0.5 }}
        className="w-full max-w-md"
    >
        <div className="rounded-3xl border border-gray-200 bg-white shadow-sm p-7 md:p-8">
            <div className="flex items-start justify-between gap-3">
                <div>
                    <h2 className="text-2xl font-bold">Create SME Account</h2>
                    <p className="text-sm text-gray-600 mt-1">
                        Register to begin your loan readiness journey.
                    </p>
                </div>
            <div className="w-10 h-10 rounded-2xl border border-gray-200 bg-gray-50 flex items-center justify-center font-bold text-sm">
                SF
            </div>
        </div>
        <form onSubmit={handleRegister} className="mt-6 space-y-4">
            <Field name="name" label="Full Name" type="text" placeholder="Your name" />
            <Field name="email" label="Email" type="email" placeholder="you@business.com" />
        </form>
    </motion.div>
</section>

```

```

<Field name="password" label="Password" type="password" placeholder="•••••••" />
<Field name="confirm" label="Confirm Password" type="password" placeholder="••••••" />

<button
  type="submit"
  disabled={loading}
  className="w-full rounded-2xl py-3 font-semibold text-white bg-black hover:opacity-90
transition disabled:opacity-50"
>
  {loading ? "Creating account..." : "Register"}
</button>

<button
  type="button"
  onClick={() => navigate("/sme/login")}
  className="w-full rounded-2xl py-3 font-semibold border border-gray-200 bg-white
hover:bg-gray-50 transition"
>
  I already have an account
</button>
</form>

<p className="text-xs text-gray-500 mt-5">
  Tip: Use an email you can access — you'll need it for future banking reports and history.
</p>
</div>
</motion.div>
</section>
</div>
</div>
);
}

/* ----- small components ----- */

function Field({ name, label, type, placeholder }) {
  return (
    <label className="block">
      <span className="text-sm text-gray-700">{label}</span>
      <input
        name={name}
        type={type}
        placeholder={placeholder}
        required
        className="mt-2 w-full rounded-2xl bg-white text-gray-900 px-4 py-3
border border-gray-200 focus:ring-2 focus:ring-gray-300 outline-none"
        />
    </label>
  );
}

function MiniItem({ title, desc }) {
  return (
    <div className="rounded-2xl border border-gray-200 bg-white p-4">
      <p className="text-sm font-semibold">{title}</p>
      <p className="text-xs text-gray-600 mt-1">{desc}</p>
    </div>
  );
}

```

```

function SoftGridBackground() {
  return (
    <div className="absolute inset-0 bg-white" />
    <div
      className="absolute inset-0 opacity-[0.35]"
      style={{
        backgroundImage:
          "linear-gradient(to right, rgba(0,0,0,0.04) 1px, transparent 1px), linear-gradient(to bottom, rgba(0,0,0,0.04) 1px, transparent 1px)",
        backgroundSize: "44px 44px",
      }}
    />
    <div className="absolute -top-40 -right-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-3xl" />
    <div className="absolute -bottom-40 -left-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-3xl" />
  );
}

```

2.14 AdminDashboard

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useEffect, useState } from "react";
import { motion } from "framer-motion";

export default function AdminDashboard() {
  const [name] = useState(localStorage.getItem("name") || "Admin");
  const [role] = useState(localStorage.getItem("role") || "ADMIN");

  useEffect(() => {
    document.title = "Admin Dashboard | Smart Finance";
  }, []);

  return (
    <div className="min-h-screen bg-white text-gray-900 relative overflow-hidden">
      <SoftGridBackground />

      {/* ===== HEADER BAR ===== */}
      <header className="relative z-10 border-b border-gray-200 bg-white/80 backdrop-blur-xl">
        <div className="max-w-6xl mx-auto px-6 py-4 flex items-center justify-between gap-4">
          <div className="flex items-center gap-3">
            <div className="w-10 h-10 rounded-2xl border border-gray-200 bg-gray-50 flex items-center justify-center font-bold text-sm">
              SF
            </div>
            <div>
              <p className="text-[11px] uppercase tracking-wide text-gray-500">
                Smart Finance
              </p>
              <h1 className="text-lg font-semibold leading-tight">
                Central Admin Dashboard
              </h1>
            </div>
          </div>
        </div>
      </header>
    </div>
  );
}

```

```

        </h1>
    </div>
</div>

<div className="hidden md:flex items-center gap-2">
    <button
        onClick={() => (window.location.href = "/admin/policy")}
        className="px-4 py-2 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50 text-sm font-medium transition"
    >
        Policy Settings
    </button>
    <button
        onClick={() => (window.location.href = "/admin/banks")}
        className="px-4 py-2 rounded-2xl bg-black text-white hover:opacity-90 text-sm font-medium transition"
    >
        Bank Management
    </button>
</div>

<div className="flex items-center gap-3">
    <div className="hidden sm:block text-right">
        <p className="text-xs font-semibold">{name}</p>
        <p className="text-[11px] text-gray-500">{role}</p>
    </div>

    <button
        onClick={() => {
            localStorage.clear();
            window.location.href = "/";
        }}
        className="px-4 py-2 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50 text-sm font-medium transition"
    >
        Logout
    </button>
</div>
</div>

/* Mobile buttons */
<div className="md:hidden px-6 pb-4 flex gap-2">
    <button
        onClick={() => (window.location.href = "/admin/policy")}
        className="flex-1 px-4 py-2 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50 text-sm font-medium transition"
    >
        Policy Settings
    </button>
    <button
        onClick={() => (window.location.href = "/admin/banks")}
        className="flex-1 px-4 py-2 rounded-2xl bg-black text-white hover:opacity-90 text-sm font-medium transition"
    >
        Bank Management
    </button>
</div>
</header>

/* Quote / mission (white theme) */

```

```

<section className="relative z-10 border-b border-gray-200 bg-gray-50">
  <div className="max-w-6xl mx-auto px-6 py-3">
    <p className="text-xs md:text-sm text-gray-700 italic">
      "To be a modern, dynamic and credible central bank contributing to the prosperity of
      Sri Lanka."
    </p>
  </div>
</section>

{/* ===== MAIN ===== */}
<main className="relative z-10 max-w-6xl mx-auto px-6 py-8">
  <div className="flex items-end justify-between gap-3 mb-5">
    <div>
      <p className="text-xs text-gray-500">System overview</p>
      <h2 className="text-2xl font-bold">Key Indicators</h2>
    </div>

    <div className="hidden sm:flex items-center gap-2 text-xs">
      <span className="px-3 py-1.5 rounded-full border border-gray-200 bg-white">
        Live
      </span>
      <span className="px-3 py-1.5 rounded-full border border-gray-200 bg-white">
        Confidential
      </span>
    </div>
  </div>

  <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
    <KpiCard title="Policy Rate" value="8.5%" sub="Latest rate" />
    <KpiCard title="Registered Banks" value="12" sub="Active institutions" />
    <KpiCard title="SMEs Registered" value="243" sub="In the platform" />
    <KpiCard title="Avg Loan Approval Rate" value="67%" sub="Estimated" />
    <KpiCard title="System Health" value="Stable" sub="No incidents" />
    <KpiCard title="Pending Reviews" value="4" sub="Need action" tone="warn" />
  </div>
</main>

{/* ===== FOOTER ===== */}
<footer className="relative z-10 text-center text-xs text-gray-500 py-5 border-t border-gray-200">
  © {new Date().getFullYear()} Central Finance Authority — Confidential Use Only
</footer>
</div>
);

/*
----- KPI Card (white SME-style)
----- */
function KpiCard({ title, value, sub, tone }) {
  const ring =
    tone === "warn"
    ? "ring-amber-200 bg-amber-50"
    : "ring-gray-200 bg-white";

  const badge =
    tone === "warn"
    ? "bg-amber-100 text-amber-800"
    : "bg-gray-100 text-gray-700";

  return (

```

```

<motion.div
  initial={{ opacity: 0, y: 10 }}
  animate={{ opacity: 1, y: 0 }}
  transition={{ duration: 0.45 }}
  className={`rounded-3xl p-5 shadow-sm ring-1 ${ring}`}
>
  <div className="flex items-start justify-between gap-3">
    <div>
      <p className="text-[11px] uppercase tracking-wide text-gray-500">
        {title}
      </p>
      <p className="mt-2 text-2xl font-bold text-gray-900">{value}</p>
    </div>
    <span className={`text-[11px] px-2.5 py-1 rounded-full ${badge}`}>
      {tone === "warn" ? "Attention" : "OK"}
    </span>
  </div>

  <p className="mt-3 text-xs text-gray-600">{sub}</p>
</motion.div>
);
}

/*
-----  

  Background same style as SME pages  

----- */
function SoftGridBackground() {
  return (
    <>
      <div className="absolute inset-0 bg-white" />
      <div
        className="absolute inset-0 opacity-[0.35]"
        style={{
          backgroundImage:
            "linear-gradient(to right, rgba(0,0,0,0.04) 1px, transparent 1px), linear-gradient(to bottom, rgba(0,0,0,0.04) 1px, transparent 1px)",
          backgroundSize: "44px 44px",
        }}
      />
      <div className="absolute -top-40 -right-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-3xl" />
      <div className="absolute -bottom-40 -left-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-3xl" />
    </>
  );
}

```

2.15 BankDashboard

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useEffect, useState } from "react";
import AddManagerModal from "../components/AddManagerModal";

/*
=====
```

Branch Management Section

```
=====
function BranchManagement({ token }) {
  const [branches, setBranches] = useState([]);
  const [form, setForm] = useState({ name: "", code: "", location: "" });
  const [selectedBranch, setSelectedBranch] = useState(null);
  const [showModal, setShowModal] = useState(false);

  useEffect(() => {
    loadBranches();
  }, [token]);

  const loadBranches = async () => {
    try {
      const res = await fetch("http://localhost:8081/api/bank-admin/branches", {
        headers: { Authorization: `Bearer ${token}` },
      });
      const data = await res.json();
      setBranches(data);
    } catch (err) {
      console.error("Error loading branches:", err);
    }
  };

  const handleAddBranch = async () => {
    if (!form.name || !form.code || !form.location) {
      alert("Please fill in all fields.");
      return;
    }

    await fetch("http://localhost:8081/api/bank-admin/branches", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        Authorization: `Bearer ${token}`,
      },
      body: JSON.stringify(form),
    });

    setForm({ name: "", code: "", location: "" });
    await loadBranches();
  };

  return (
    <div style={{ marginTop: "2rem" }}>
      <h2 style={{ color: "var(--accent)", marginBottom: "1rem" }}>
         Branch Management
      </h2>
      {/* + Add Branch Form */}
      <div className="flex gap-2 mt-3">
        <input
          placeholder="Branch Name"
          value={form.name}
          onChange={(e) => setForm({ ...form, name: e.target.value })}
        />
        <input
          placeholder="Code"
          value={form.code}
          onChange={(e) => setForm({ ...form, code: e.target.value })}
        />
      </div>
    </div>
  );
}
```

```

/>
<input
  placeholder="Location"
  value={form.location}
  onChange={(e) => setForm({ ...form, location: e.target.value })}
/>
<button className="btn-primary" onClick={handleAddBranch}>
   Add Branch
</button>
</div>

/* 📄 Branch Cards */
<div className="grid grid-3 mt-6">
  {branches.map((b) => (
    <div
      key={b.id}
      className="card"
      style={{
        border: "1px solid var(--accent)",
        borderRadius: "10px",
        transition: "transform 0.2s ease-in-out",
      }}
      onMouseEnter={(e) =>
        (e.currentTarget.style.transform = "scale(1.03)")
      }
      onMouseLeave={(e) =>
        (e.currentTarget.style.transform = "scale(1)")
      }
    >
      <h3 style={{ fontWeight: "600" }}>{b.name}</h3>
      <p style={{ opacity: 0.8 }}>{b.location}</p>

      <p style={{ marginTop: "0.5rem" }}>
        Manager:{" "}
        {b.manager ? (
          <span style={{ color: "var(--accent)" }}>
            {b.manager.name}
          </span>
        ) : (
          <span style={{ color: "#f00" }}>Unassigned</span>
        )}
      </p>

      /* 📧 Contact or Assign Manager */
      {b.manager ? (
        <button
          onClick={() =>
            (window.location.href = `mailto:${b.manager.email}?subject=Message from Bank Admin
(${b.name})&body=Dear ${b.manager.name},%0D%0A%0D%0A`)
          }
        style={{
          background: "var(--accent)",
          color: "#fff",
          borderRadius: "6px",
          padding: "0.5rem 1rem",
          marginTop: "0.8rem",
          cursor: "pointer",
          border: "none",
          fontWeight: "500",
        }}
      )
    )
  )
}

```

```

>
     Contact Manager
  </button>
):(
<button
  onClick={()=> {
    setSelectedBranch(b);
    setShowModal(true);
  }}
  className="btn-primary"
  style={{ marginTop: "0.8rem" }}
>
   Assign Manager
</button>
)
</div>
))
</div>

/* ✨ Add Manager Modal */
{showModal && (
  <AddManagerModal
    branch={selectedBranch}
    token={token}
    onClose={() => setShowModal(false)}
    onSuccess={() => {
      loadBranches();
      setShowModal(false);
    }}
  />
)
</div>
);
}

/*
  🏛 Bank Admin Dashboard
=====
*/
export default function BankDashboard() {
  const [bankInfo, setBankInfo] = useState(null);
  const [loading, setLoading] = useState(true);
  const [theme, setTheme] = useState(localStorage.getItem("theme") || "light");
  const token = localStorage.getItem("token");

  useEffect(() => {
    async function fetchBankData() {
      try {
        const res = await fetch("http://localhost:8081/api/bank-admin/me", {
          headers: { Authorization: `Bearer ${token}` },
        });
        if (!res.ok) throw new Error("Cannot fetch bank info");
        const data = await res.json();
        setBankInfo(data);
      }
      // 💡 Apply dynamic bank theme
      const themeKey = data.theme?.themeKey?.toLowerCase() || "default";
      document.body.className = "";
      document.body.classList.add(`theme-${themeKey}`);
    } catch (err) {

```

```

        console.error("Error loading bank info:", err);
    } finally {
        setLoading(false);
    }
}
fetchBankData();
}, [token]);

useEffect(() => {
    document.documentElement.setAttribute("data-theme", theme);
}, [theme]);

const toggleTheme = () => {
    const newTheme = theme === "light" ? "dark" : "light";
    setTheme(newTheme);
    localStorage.setItem("theme", newTheme);
};

if (loading) return <p>Loading bank dashboard...</p>;
if (!bankInfo) return <p>Error: Could not load bank info</p>;

return (
    <div className="dashboard">
        <div className="dashboard-header">
            <div style={{ display: "flex", alignItems: "center", gap: "1rem" }}>
                <img
                    src={`/logos/${bankInfo.theme?.themeKey || "default"}.png`}
                    alt={`${${bankInfo.bankName} logo`}
                    className="bank-logo"
                />
                <h1>{bankInfo.bankName} Dashboard</h1>
            </div>
            <button className="theme-toggle" onClick={toggleTheme}>
                {theme === "light" ? "🌙 Dark Mode" : "☀️ Light Mode"}
            </button>
        </div>
        {/* KPI Metrics */}
        <div className="grid grid-3 mt-8">
            <div className="card">
                <h2>Total Branches</h2>
                <p className="kpi-value">{bankInfo.branchCount ?? "—"}</p>
            </div>
            <div className="card">
                <h2>Active Loans</h2>
                <p className="kpi-value">{bankInfo.totalLoans ?? 0}</p>
            </div>
            <div className="card">
                <h2>Pending Approvals</h2>
                <p className="kpi-value">{bankInfo.pendingLoans ?? 0}</p>
            </div>
        </div>
        {/* 🏢 Branch Management Section */}
        <BranchManagement token={token} />
        {/* 📄 Footer */}
        <footer
            style={{


```

```

        textAlign: "center",
        marginTop: "3rem",
        color: "var(--accent)",
        opacity: 0.7,
    })
>
    © {new Date().getFullYear()} {bankInfo.bankName} | Managed by Central
    Finance Authority
</footer>
</div>
);
}

```

2.16 BranchManagement

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useEffect, useState } from "react";

export default function BranchManagement() {
    const [branches, setBranches] = useState([]);
    const [name, setName] = useState("");
    const [code, setCode] = useState("");
    const [location, setLocation] = useState("");

    const token = localStorage.getItem("token");

    useEffect(() => {
        fetch("http://localhost:8081/api/bank-admin/branches", {
            headers: { Authorization: `Bearer ${token}` },
        })
        .then(res => res.json())
        .then(setBranches);
    }, []);

    const addBranch = () => {
        fetch("http://localhost:8081/api/bank-admin/branches", {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
                Authorization: `Bearer ${token}`,
            },
            body: JSON.stringify({ name, code, location }),
        }).then(() => window.location.reload());
    };

    return (
        <div className="p-4">
            <h2 className="text-xl font-bold">Branches</h2>
            <div className="my-3">
                <input placeholder="Name" value={name} onChange={e => setName(e.target.value)} />
                <input placeholder="Code" value={code} onChange={e => setCode(e.target.value)} />
                <input placeholder="Location" value={location} onChange={e => setLocation(e.target.value)} />
                <button onClick={addBranch}>Add Branch</button>
            </div>
        </div>
    );
}

```

```

{branches.map(b => (
  <div key={b.id} className="border p-3 my-2">
    <h3>{b.name}</h3>
    <p>{b.location}</p>
    <p>Manager: {b.manager?.name || "None"}</p>
  </div>
))
);
}

```

2.17 Login

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import AnimatedPage from "../components/AnimatedPage.jsx";

export default function Login() {
  return (
    <AnimatedPage>
      <div className="bg-white/80 backdrop-blur-lg p-8 rounded-3xl shadow-2xl w-96 text-center">
        <h1 className="text-3xl font-bold text-transparent bg-clip-text bg-gradient-to-r from-blue-600 to-indigo-600">
          Smart Finance Dashboard
        </h1>
        <p className="text-gray-600 mt-2 mb-6">Sign in to your account</p>

        <form className="space-y-4">
          <input
            type="email"
            placeholder="Email"
            className="w-full border border-gray-300 rounded-xl p-3 focus:ring-2 focus:ring-blue-500 outline-none">
          />
          <input
            type="password"
            placeholder="Password"
            className="w-full border border-gray-300 rounded-xl p-3 focus:ring-2 focus:ring-blue-500 outline-none">
          />
          <button
            type="submit"
            className="w-full py-3 rounded-xl bg-gradient-to-r from-blue-600 to-indigo-600 text-white font-semibold hover:shadow-lg transition-transform hover:scale-[1.02]">
            >
              Login
            </button>
        </form>

        <p className="text-sm text-gray-500 mt-6">
          Don't have an account?{" "}
        <a href="/register" className="text-blue-600 font-medium hover:underline">
          Register
        </a>

```

```

        </p>
    </div>
</AnimatedPage>
);
}

```

2.18 ManagerDashboard

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import { useEffect, useState } from "react";

export default function ManagerDashboard() {
    const [managerInfo, setManagerInfo] = useState(null);
    const [dashboard, setDashboard] = useState(null);
    const [loading, setLoading] = useState(true);
    const token = localStorage.getItem("token");

    useEffect(() => {
        async function fetchManagerData() {
            try {
                const res = await fetch("http://localhost:8081/api/manager/me", {
                    headers: { Authorization: `Bearer ${token}` },
                });
                if (!res.ok) throw new Error("Cannot fetch manager profile");
                const profile = await res.json();
                setManagerInfo(profile);

                // 🌈 Apply CSS theme class
                const themeKey = profile.theme?.themeKey?.toLowerCase() || "default";
                document.body.className = "";
                document.body.classList.add(`theme-${themeKey}`);
            }

            const dashRes = await fetch("http://localhost:8081/api/manager/dashboard", {
                headers: { Authorization: `Bearer ${token}` },
            });
            if (!dashRes.ok) throw new Error("Cannot fetch dashboard");
            const dashData = await dashRes.json();
            setDashboard(dashData);

            } catch (err) {
                console.error("Error loading manager data:", err);
            } finally {
                setLoading(false);
            }
        }
        fetchManagerData();
    }, [token]);
}

if (loading) return <p>Loading manager dashboard...</p>;
if (!managerInfo || !dashboard)
    return <p>Error: Could not load manager dashboard</p>;

```

```

return (
  <div style={{ minHeight: "100vh", padding: "2rem" }}>
    <h1 style={{ color: "var(--accent)" }}>
      🏦 {managerInfo.branchName} — {managerInfo.bankName}
    </h1>
    <h3>Welcome, {managerInfo.name}</h3>

    <div className="grid grid-cols-3 gap-4 mt-8">
      <div className="card">
        <h2>Total Loans</h2>
        <p>{dashboard.totalLoans}</p>
      </div>
      <div className="card">
        <h2>Approved</h2>
        <p>{dashboard.approvedLoans}</p>
      </div>
      <div className="card">
        <h2>Pending</h2>
        <p>{dashboard.pendingLoans}</p>
      </div>
      <div className="card">
        <h2>Rejected</h2>
        <p>{dashboard.rejectedLoans}</p>
      </div>
      <div className="card">
        <h2>Total Amount</h2>
        <p>Rs. {dashboard.totalAmount.toLocaleString()}</p>
      </div>
      <div className="card">
        <h2>Avg Loan</h2>
        <p>Rs. {dashboard.averageLoanAmount.toLocaleString()}</p>
      </div>
    </div>

    <div className="mt-10">
      <h2 style={{ color: "var(--accent)" }}>
        🚀 Officers in {managerInfo.branchName}
      </h2>
      <ul>
        {dashboard.officers.length > 0 ? (
          dashboard.officers.map((o) => (
            <li key={o.id}>
              {o.name} — {o.email}
            </li>
          )))
        : (
          <p>No officers found in this branch</p>
        )}
      </ul>
    </div>
  </div>
);
}

```

2.19 PolicySettings

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
```

Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23

```
/*
import { useEffect, useState } from "react";
import { motion } from "framer-motion";

export default function PolicySettings() {
  const [policy, setPolicy] = useState(null);
  const [loading, setLoading] = useState(true);
  const [saving, setSaving] = useState(false);
  const [error, setError] = useState("");
  const [analysis, setAnalysis] = useState(null);

  useEffect(() => {
    document.title = "Policy Settings | Smart Finance";
    fetchPolicy();
    // eslint-disable-next-line react-hooks/exhaustive-deps
  }, []);

  async function fetchPolicy() {
    setLoading(true);
    setError("");
    const token = localStorage.getItem("token");

    try {
      const res = await fetch("http://localhost:8081/api/admin/policy", {
        headers: { Authorization: `Bearer ${token}` },
      });
      if (!res.ok) throw new Error("Failed to load policy data");
      const data = await res.json();
      setPolicy(data);
    } catch (err) {
      setError(err.message || "Failed to load policy");
    } finally {
      setLoading(false);
    }
  }

  async function handleSave() {
    if (!policy) return;
    setSaving(true);
    const token = localStorage.getItem("token");

    try {
      const res = await fetch("http://localhost:8081/api/admin/policy", {
        method: "PUT",
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${token}`,
        },
        body: JSON.stringify(policy),
      });
      if (!res.ok) throw new Error("Failed to update policy");
      alert("✅ Policy updated successfully!");
    } catch (err) {
      alert("❌ " + (err.message || "Save failed"));
    } finally {
      setSaving(false);
    }
  }
}
```

```

        }

    }

    async function runAIAnalysis() {
        const token = localStorage.getItem("token");
        try {
            const res = await fetch("http://localhost:8081/api/admin/analyze", {
                headers: { Authorization: `Bearer ${token}` },
            });
            if (!res.ok) throw new Error("AI analysis failed");
            const data = await res.json();
            setAnalysis(data);
        } catch (err) {
            alert("✖ " + (err.message || "AI analysis failed"));
        }
    }

    async function downloadQuarterlyReport() {
        const token = localStorage.getItem("token");

        const response = await fetch("http://localhost:8081/api/admin/report/quarterly", {
            headers: { Authorization: `Bearer ${token}` },
        });

        if (!response.ok) {
            alert("Failed to download report");
            return;
        }

        const blob = await response.blob();
        const url = window.URL.createObjectURL(blob);
        const a = document.createElement("a");
        a.href = url;
        a.download = "Quarterly_Report.pdf";
        document.body.appendChild(a);
        a.click();
        a.remove();
        window.URL.revokeObjectURL(url);
    }

    if (loading) {
        return (
            <div className="min-h-screen bg-white text-gray-900 flex items-center justify-center">
                <p className="text-sm text-gray-600">Loading policy data...</p>
            </div>
        );
    }

    if (error) {
        return (
            <div className="min-h-screen bg-white text-gray-900 flex flex-col items-center justify-center px-6">
                <p className="text-sm text-red-600">{error}</p>
                <button
                    onClick={fetchPolicy}
                    className="mt-4 px-4 py-2 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50 text-sm font-medium"
                >
                    Retry
                </button>
            </div>
        );
    }
}

```

```

    );
}

return (
  <div className="min-h-screen bg-white text-gray-900 relative overflow-hidden">
    <SoftGridBackground />

    {/* Header */}
    <header className="relative z-10 border-b border-gray-200 bg-white/80 backdrop-blur-xl">
      <div className="max-w-6xl mx-auto px-6 py-4 flex items-center justify-between gap-4">
        <div className="flex items-center gap-3">
          <div className="w-10 h-10 rounded-2xl border border-gray-200 bg-gray-50 flex items-center justify-center font-bold text-sm">
            SF
          </div>
          <div>
            <p className="text-[11px] uppercase tracking-wide text-gray-500">Smart Finance</p>
            <h1 className="text-lg font-semibold leading-tight">Central Policy Settings</h1>
          </div>
        </div>
      </div>

      <button
        onClick={() => (window.location.href = "/admin/dashboard")}
        className="px-4 py-2 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50 text-sm font-medium transition">
        >
        ← Back to Dashboard
      </button>
    </div>
  </header>

  {/* Content */}
  <main className="relative z-10 max-w-6xl mx-auto px-6 py-8">
    <div className="flex items-end justify-between gap-3 mb-5">
      <div>
        <p className="text-xs text-gray-500">Edit rates & generate quarterly insights</p>
        <h2 className="text-2xl font-bold">Policy Controls</h2>
      </div>

      <div className="hidden sm:flex items-center gap-2">
        <button
          onClick={runAIAnalysis}
          className="px-4 py-2 rounded-2xl bg-black text-white hover:opacity-90 text-sm font-medium transition">
          >
          Run AI Analysis
        </button>
      </div>
    </div>
  </main>

  <motion.div
    initial={{ opacity: 0, y: 10 }}
    animate={{ opacity: 1, y: 0 }}
    transition={{ duration: 0.45 }}
    className="grid grid-cols-1 lg:grid-cols-[1.2fr,0.8fr] gap-4">
    >
    {/* Left: Form */}
    <div className="rounded-3xl bg-white ring-1 ring-gray-200 shadow-sm p-6">
      <div className="flex items-center justify-between mb-4">
        <div>

```

```

<p className="text-[11px] uppercase tracking-wide text-gray-500">Parameters</p>
  <p className="text-sm font-semibold">Update policy rates</p>
</div>
<span className="text-[11px] px-2.5 py-1 rounded-full bg-gray-100 text-gray-700">
  Admin only
</span>
</div>

<div className="space-y-4">
  <Field
    label="Policy Rate (%)"
    value={policy?.policyRate ?? ""}
    onChange={(v) => setPolicy({ ...policy, policyRate: parseFloat(v) })}
  />
  <Field
    label="Savings Rate (%)"
    value={policy?.savingsRate ?? ""}
    onChange={(v) => setPolicy({ ...policy, savingsRate: parseFloat(v) })}
  />
  <Field
    label="Reserve Ratio (%)"
    value={policy?.reserveRatio ?? ""}
    onChange={(v) => setPolicy({ ...policy, reserveRatio: parseFloat(v) })}
  />
</div>

<div className="mt-6 flex flex-col sm:flex-row gap-3">
  <button
    onClick={handleSave}
    disabled={saving}
    className={`px-5 py-2.5 rounded-2xl bg-black text-white text-sm font-semibold transition
    hover:opacity-90 ${saving ? "opacity-60 cursor-not-allowed" : ""}`}
  >
    {saving ? "Saving..." : "Save Changes"}
  </button>

  <button
    onClick={downloadQuarterlyReport}
    className="px-5 py-2.5 rounded-2xl border border-gray-200 bg-white hover:bg-gray-50
    text-sm font-semibold transition"
  >
     Download Quarterly Report
  </button>

  <button
    onClick={runAIAnalysis}
    className="sm:hidden px-5 py-2.5 rounded-2xl border border-gray-200 bg-white
    hover:bg-gray-50 text-sm font-semibold transition"
  >
     Run AI Analysis
  </button>
</div>
</div>

/* Right: AI output */
<div className="rounded-3xl bg-white ring-1 ring-gray-200 shadow-sm p-6">
  <div className="flex items-center justify-between mb-4">
    <div>

```

```

        <p className="text-[11px] uppercase tracking-wide text-gray-500">AI Insights</p>
        <p className="text-sm font-semibold">Recommendation</p>
    </div>
    <span className="text-[11px] px-2.5 py-1 rounded-full bg-gray-100 text-gray-700">
        Auto
    </span>
    </div>

    {!analysis ? (
        <div className="rounded-2xl bg-gray-50 ring-1 ring-gray-200 p-4">
            <p className="text-sm text-gray-700 font-medium">No AI analysis yet.</p>
            <p className="text-xs text-gray-500 mt-1">
                Click <b>Run AI Analysis</b> to generate a short recommendation summary.
            </p>
        </div>
    ) : (
        <div className="space-y-3">
            <div className="rounded-2xl bg-gray-50 ring-1 ring-gray-200 p-4">
                <p className="text-xs uppercase tracking-wide text-gray-500 mb-1">
                    Summary
                </p>
                <p className="text-sm text-gray-800">{analysis.analysisSummary}</p>
            </div>

            <div className="rounded-2xl bg-gray-50 ring-1 ring-gray-200 p-4">
                <p className="text-xs uppercase tracking-wide text-gray-500 mb-2">
                    Suggested Policy Rate
                </p>
                <p className="text-2xl font-bold">
                    {Number(analysis.suggestedPolicyRate).toFixed(2)}%
                </p>

                <div className="mt-3 grid grid-cols-3 gap-2 text-center">
                    <MiniStat label="SME Growth" value={\$ {Number(analysis.smeGrowth).toFixed(2)}%} />
                    <MiniStat label="Defaults" value={\$ {Number(analysis.loanDefaultRate).toFixed(2)}%} />
                    <MiniStat label="Inflation" value={\$ {Number(analysis.inflation).toFixed(2)}%} />
                </div>
            </div>
        </div>
    );
}

function Field({ label, value, onChange }) {
    return (
        <label className="block">
            <span className="text-xs text-gray-600">{label}</span>
            <input
                type="number"
                step="0.1"

```

```

        value={value}
        onChange={(e) => onChange(e.target.value)}
        className="mt-2 w-full rounded-2xl bg-white border border-gray-200 px-4 py-3 outline-none
focus:ring-2 focus:ring-black"
      />
    </label>
  );
}

function MiniStat({ label, value }) {
  return (
    <div className="rounded-2xl bg-white border border-gray-200 p-2">
      <p className="text-[10px] uppercase tracking-wide text-gray-500">{label}</p>
      <p className="text-sm font-semibold text-gray-900 mt-1">{value}</p>
    </div>
  );
}

/* background same SME/admin style */
function SoftGridBackground() {
  return (
    <>
      <div className="absolute inset-0 bg-white" />
      <div
        className="absolute inset-0 opacity-[0.35]"
        style={{
          backgroundImage:
            "linear-gradient(to right, rgba(0,0,0,0.04) 1px, transparent 1px), linear-gradient(to bottom,
            rgba(0,0,0,0.04) 1px, transparent 1px)",
          backgroundSize: "44px 44px",
        }}
      />
      <div className="absolute -top-40 -right-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-
      3xl" />
      <div className="absolute -bottom-40 -left-40 w-[34rem] h-[34rem] rounded-full bg-gray-200/30 blur-
      3xl" />
    </>
  );
}

```

2.20 Register

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import AnimatedPage from "../components/AnimatedPage.jsx";

export default function Register() {
  return (
    <AnimatedPage>
      <div className="bg-white/80 backdrop-blur-lg p-8 rounded-3xl shadow-2xl w-96 text-center">
        <h1 className="text-3xl font-bold text-transparent bg-clip-text bg-gradient-to-r from-green-600 to-
        emerald-600">
          Create Account
        </h1>
        <p className="text-gray-600 mt-2 mb-6">Join the SME Readiness Portal</p>
      </div>
    </AnimatedPage>
  );
}

```

```

<form className="space-y-4">
  <input
    type="text"
    placeholder="Full Name"
    className="w-full border border-gray-300 rounded-xl p-3 focus:ring-2 focus:ring-green-500
outline-none">
  />
  <input
    type="email"
    placeholder="Email"
    className="w-full border border-gray-300 rounded-xl p-3 focus:ring-2 focus:ring-green-500
outline-none">
  />
  <input
    type="password"
    placeholder="Password"
    className="w-full border border-gray-300 rounded-xl p-3 focus:ring-2 focus:ring-green-500
outline-none">
  />
  <button
    type="submit"
    className="w-full py-3 rounded-xl bg-gradient-to-r from-green-600 to-emerald-600 text-white
font-semibold hover:shadow-lg transition-transform hover:scale-[1.02]">
    Register
  </button>
</form>

<p className="text-sm text-gray-500 mt-6">
  Already have an account?{" "}
  <a href="/public" className="text-green-600 font-medium hover:underline">
    Login
  </a>
</p>
</div>
</AnimatedPage>
);
}

```

2.21 global.css

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
/* =====
💡 LIGHT / 🌙 DARK THEME TOKENS
===== */

```

```

/* Light mode defaults */
:root {
  --bg: #f9fafb;
  --text: #0b1120;
  --card-bg: #ffffff;
  --card-text: #0f172a;

```

```

--input-bg: rgba(255, 255, 255, 0.9);
--input-border: #d4d4d8;
--shadow-soft: 0 18px 45px rgba(15, 23, 42, 0.12);

--primary: #4f46e5; /* generic brand primary */
--accent: #6366f1; /* used for borders, focus, chips */

--radius-card: 1.25rem;
--radius-pill: 999px;
}

/* Dark mode override (triggered by: document.documentElement.setAttribute('data-theme','dark')) */
[data-theme="dark"] {
  --bg: #020617;
  --text: #e5e7eb;
  --card-bg: rgba(15, 23, 42, 0.9);
  --card-text: #f9fafb;
  --input-bg: rgba(15, 23, 42, 0.95);
  --input-border: #27272f;
  --shadow-soft: 0 24px 60px rgba(0, 0, 0, 0.75);
}

/* =====
  BANK BRAND THEMES (LIGHT MODE)
===== */

.theme-central {
  --primary: #800000;
  --accent: #a83232;
  --bg: #f9fafc;
  --text: #1c1c1c;
  --card-bg: #ffffff;
}

.theme-boc {
  --primary: #ffec00;
  --accent: #333333;
  --bg: #fafafa;
  --text: #1c1c1c;
  --card-bg: #ffffff;
}

.theme-peoples {
  --primary: #a80000;
  --accent: #ffcc00;
  --bg: #fafafa;
  --text: #222;
  --card-bg: #ffffff;
}

.theme-hnb {
  --primary: #004481;
  --accent: #ffc107;
  --bg: #f6f8fb;
  --text: #1b1b1b;
  --card-bg: #ffffff;
}

.theme-sampath {
  --primary: #c8102e;
}

```



```

),
radial-gradient(
  circle at bottom right,
  rgba(79, 70, 229, 0.12),
  transparent 55%
),
var(--card-bg);
color: var(--card-text);
border-radius: var(--radius-card);
padding: 1.25rem 1.5rem;
box-shadow: var(--shadow-soft);
border: 1px solid rgba(148, 163, 253, 0.3);
backdrop-filter: blur(18px);
-webkit-backdrop-filter: blur(18px);
transition: transform 0.2s ease, box-shadow 0.25s ease, border-color 0.25s ease,
background 0.25s ease;
}

.card:hover {
  transform: translateY(-3px);
  box-shadow: 0 26px 60px rgba(15, 23, 42, 0.4);
  border-color: rgba(129, 140, 248, 0.7);
}

/* Simple flat-card variant if needed */
.card-flat {
  background: var(--card-bg);
  color: var(--card-text);
  border-radius: 1rem;
  padding: 1.25rem 1.5rem;
  border: 1px solid rgba(148, 163, 253, 0.25);
}

=====*
 DASHBOARD LAYOUT
===== */

.dashboard {
  max-width: 1400px;
  margin: 0 auto;
  padding: 1.75rem 2rem 3rem;
}

.dashboard-header {
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 1.5rem;
  margin-bottom: 1.75rem;
}

.bank-logo {
  height: 56px;
  width: auto;
  border-radius: 0.75rem;
  background: rgba(15, 23, 42, 0.5);
  padding: 0.35rem 0.5rem;
}

.dashboard-header h1 {

```

```

    font-size: 1.75rem;
    font-weight: 700;
    color: var(--text);
}

/* KPI grid */
.grid {
    display: grid;
    gap: 1.4rem;
}

.grid-3 {
    grid-template-columns: repeat(auto-fit, minmax(260px, 1fr));
}

.kpi-value {
    font-size: 1.9rem;
    font-weight: 700;
    letter-spacing: 0.03em;
}

.kpi-label {
    font-size: 0.85rem;
    opacity: 0.75;
}

/*
===== O BUTTONS ===== */
.btn-primary {
    background-image: linear-gradient(
        to right,
        #2563eb,
        #4f46e5,
        #7c3aed
    );
    color: #f9fafb;
    border: none;
    border-radius: 999px;
    padding: 0.55rem 1.35rem;
    font-size: 0.9rem;
    font-weight: 500;
    cursor: pointer;
    box-shadow: 0 12px 30px rgba(79, 70, 229, 0.35);
    transition: transform 0.15s ease, box-shadow 0.15s ease, filter 0.2s ease;
}

.btn-primary:hover {
    transform: translateY(-1px) scale(1.01);
    box-shadow: 0 18px 40px rgba(79, 70, 229, 0.4);
    filter: brightness(1.04);
}

.btn-primary:active {
    transform: translateY(0) scale(0.99);
    box-shadow: 0 10px 24px rgba(79, 70, 229, 0.35);
}

/* Toggle for dark/light */

```

```

.theme-toggle {
  background: linear-gradient(
    to right,
    #0f172a,
    #111827
  );
  color: #e5e7eb;
  border-radius: 999px;
  padding: 0.4rem 0.9rem;
  font-size: 0.8rem;
  border: 1px solid rgba(148, 163, 253, 0.5);
  display: inline-flex;
  align-items: center;
  gap: 0.4rem;
  cursor: pointer;
  box-shadow: 0 10px 28px rgba(15, 23, 42, 0.65);
  transition: background 0.2s ease, transform 0.15s ease, box-shadow 0.2s ease;
}


```

```

.theme-toggle:hover {
  transform: translateY(-1px);
  box-shadow: 0 16px 40px rgba(15, 23, 42, 0.85);
}


```

```

/* =====
  🖌 INPUTS & TEXTAREAS
===== */

```

```

input,
textarea,
select {
  background-color: var(--input-bg);
  color: var(--text);
  border-radius: 0.75rem;
  border: 1px solid var(--input-border);
  padding: 0.55rem 0.85rem;
  font-size: 0.9rem;
  outline: none;
  transition: border-color 0.18s ease, box-shadow 0.18s ease,
  background-color 0.18s ease;
}


```

```

input::placeholder,
textarea::placeholder {
  color: rgba(148, 163, 184, 0.85);
}


```

```

input:focus,
textarea:focus,
select:focus {
  border-color: #6366f1;
  box-shadow: 0 0 0 1px rgba(99, 102, 241, 0.5);
}


```

```

/* =====
  📄 SMALL UTILITIES
===== */

```

```

.badge-soft {
  display: inline-flex;

```

```

align-items: center;
gap: 0.3rem;
padding: 0.15rem 0.6rem;
border-radius: 999px;
font-size: 0.75rem;
background: rgba(148, 163, 253, 0.15);
border: 1px solid rgba(129, 140, 248, 0.5);
color: #e5e7eb;
}

/* SME Shell (optional wrapper) */
.sme-shell {
min-height: 100vh;
background: radial-gradient(
  circle at top,
  rgba(56, 189, 248, 0.08),
  transparent 60%
),
radial-gradient(
  circle at bottom,
  rgba(168, 85, 247, 0.1),
  transparent 55%
),
var(--bg);
}

/*
===== SCROLLBAR + SELECTION =====
===== */
::-webkit-scrollbar {
width: 7px;
}
::-webkit-scrollbar-thumb {
background: linear-gradient(to bottom, #4f46e5, #7c3aed);
border-radius: 999px;
}
::-webkit-scrollbar-track {
background: rgba(15, 23, 42, 0.2);
}

::selection {
background: rgba(79, 70, 229, 0.85);
color: #f9fafb;
}

```

2.22 api.js

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
export async function getBankInfo() {
  const token = localStorage.getItem("token");
  const res = await fetch("http://localhost:8081/api/bank-admin/me", {
    headers: {

```

```

        "Authorization": `Bearer ${token}`
    }
});
if (!res.ok) throw new Error("Failed to load bank info");
return await res.json();
}

```

2.23 App.css

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
#root {
    max-width: 1280px;
    margin: 0 auto;
    padding: 2rem;
    text-align: center;
}

.logo {
    height: 6em;
    padding: 1.5em;
    will-change: filter;
    transition: filter 300ms;
}
.logo:hover {
    filter: drop-shadow(0 0 2em #646cffaa);
}
.logo.react:hover {
    filter: drop-shadow(0 0 2em #61dafbaa);
}

@keyframes logo-spin {
    from {
        transform: rotate(0deg);
    }
    to {
        transform: rotate(360deg);
    }
}

@media (prefers-reduced-motion: no-preference) {
    a:nth-of-type(2) .logo {
        animation: logo-spin infinite 20s linear;
    }
}

.card {
    padding: 2em;
}

.read-the-docs {
    color: #888;
}

```

2.24 App.jsx

```
/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import "./styles/global.css";
import { useState } from "react";
import { Link } from "react-router-dom";
import AnimatedBlobs from "./components/AnimatedBlobs";

export default function App() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [loading, setLoading] = useState(false);

  async function onSubmit(e) {
    e.preventDefault();
    setLoading(true);

    try {
      const res = await fetch("http://localhost:8081/api/auth/login", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ email, password }),
      });

      if (!res.ok) {
        let errMsg = "Invalid email or password";
        try {
          const errData = await res.json();
          if (errData?.error) errMsg = errData.error;
        } catch {
          // ignore JSON parse error
        }
        alert(errMsg);
        setLoading(false);
        return;
      }

      const data = await res.json();

      // Persist basic auth info
      if (data.token) localStorage.setItem("token", data.token);
      if (data.role) localStorage.setItem("role", data.role);
      if (data.email) localStorage.setItem("email", data.email);

      // Redirect by role
      if (data.role === "ADMIN") {
        window.location.href = "/admin/dashboard";
      } else if (data.role === "BANK_ADMIN") {
        window.location.href = "/bank/dashboard";
      } else if (data.role === "BANK_MANAGER") {
        window.location.href = "/manager/dashboard";
      } else {
        // fallback – treat as SME-style user
        window.location.href = "/sme/portal";
      }
    } catch {
      console.error("Error during login submission");
    }
  }
}
```

```

        }
    } catch (err) {
        console.error("Login error:", err);
        alert("Cannot connect to server.");
    } finally {
        setLoading(false);
    }
}

return (
    <div className="min-h-screen bg-slate-950 text-white relative overflow-hidden">
        {/* soft gradient / blob background */}
        <AnimatedBlobs />

        <div className="relative z-10 max-w-6xl mx-auto px-4 lg:px-8 py-10 lg:py-16 flex flex-col lg:flex-row items-center gap-10">
            {/* ===== LEFT: HERO COPY (BANK SIDE) ===== */}
            <div className="flex-1">
                {/* Audience switch pill */}
                <div className="inline-flex items-center rounded-full bg-white/5 border border-white/10 px-1 py-1 text-xs mb-6">
                    <span className="px-3 py-1 rounded-full bg-indigo-500/80 text-white font-medium">
                        For Banks
                    </span>
                    <Link
                        to="/sme"
                        className="px-3 py-1 rounded-full text-slate-300 hover:text-white hover:bg-white/5 transition">
                        >
                            For SME Owners →
                        </Link>
                    </div>

                    <div className="inline-flex items-center gap-2 text-xs px-3 py-1 rounded-full bg-white/5 ring-1 ring-white/10 mb-4">
                        <span className="size-1.5 rounded-full bg-emerald-400" />
                        <span className="font-medium tracking-wide">
                            Smart Finance Dashboard
                        </span>
                    </div>

                    <h1 className="text-3xl sm:text-4xl lg:text-5xl font-semibold leading-tight">
                        Make your{" "}
                        <span className="bg-gradient-to-r from-indigo-400 via-sky-400 to-emerald-300 bg-clip-text text-transparent">
                            loan readiness
                        </span>{" "}
                        crystal clear.
                    </h1>

                    <p className="mt-4 text-sm sm:text-base text-slate-300 max-w-xl">
                        Track SME statements, auto-compute risk ratios, and share
                        investor-ready insights with a single click. Designed for bank
                        credit teams working with Sri Lankan SMEs.
                    </p>

                    <ul className="mt-6 grid grid-cols-1 sm:grid-cols-2 gap-3 text-sm">
                        {[ "Real-time KPI cards", "Automated ratio analysis", ]
                    
```

```

    "Secure document vault",
    "Bank-ready PDF reports",
].map((t) => (
  <li
    key={t}
    className="flex items-center gap-3 rounded-2xl bg-white/5 ring-1 ring-white/10 p-3"
  >
    <span className="size-2.5 rounded-full bg-gradient-to-br from-emerald-400 to-teal-400
shadow-[0_0_20px] shadow-emerald-500/20" />
    {t}
  </li>
))
</ul>
</div>

/* ===== RIGHT: BANK LOGIN CARD ===== */
<div className="w-full max-w-md">
  <div className="rounded-3xl bg-slate-900/70 border border-white/10 shadow-2xl shadow-indigo-
950/50 p-6 sm:p-7 backdrop-blur-xl">
    <h2 className="text-lg font-semibold mb-1">Welcome back</h2>
    <p className="text-xs text-slate-300 mb-5">
      Bank user login. Use your institution email and password.
    </p>

    <form className="space-y-4" onSubmit={onSubmit}>
      <div>
        <label className="block text-xs mb-1">Email</label>
        <input
          type="email"
          required
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          placeholder="you@bank.com"
          className="w-full rounded-xl bg-slate-900 border border-slate-700/70 px-3 py-2 text-sm
focus:outline-none focus:ring-2 focus:ring-indigo-500/80"
        />
      </div>

      <div>
        <label className="block text-xs mb-1">Password</label>
        <input
          type="password"
          required
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          placeholder="*****"
          className="w-full rounded-xl bg-slate-900 border border-slate-700/70 px-3 py-2 text-sm
focus:outline-none focus:ring-2 focus:ring-indigo-500/80"
        />
      </div>

      <div className="flex items-center justify-between text-xs mt-1">
        <label className="inline-flex items-center gap-2">
          <input
            type="checkbox"
            className="size-3 rounded border-slate-600 bg-slate-900"
          />
          <span className="text-slate-300">Remember me</span>
        </label>
        <Link

```

```

        to="#"  

        className="text-indigo-300 hover:text-indigo-200 underline underline-offset-2"  

      >  

        Forgot password?  

      </Link>  

    </div>  
  

    <button  

      type="submit"  

      disabled={loading}  

      className="mt-3 w-full py-2.5 text-sm font-medium rounded-xl bg-gradient-to-r from-indigo-500 via-purple-500 to-pink-500 hover:opacity-90 transition disabled:opacity-60"  

    >  

      {loading ? "Logging in..." : "Login"}  

    </button>  

  </form>  
  

  {/* SME link */}  

  <div className="mt-4 border-t border-white/5 pt-3 text-xs text-slate-300 space-y-1">  

    <p>  

      <span className="font-medium text-slate-100">  

        SME owner or entrepreneur?  

      </span>  

    </p>  

    <p>  

      Use our dedicated SME portal to check your loan readiness and  

      get personalized tips:  

    </p>  

    <p>  

      <Link  

        to="/sme"  

        className="text-emerald-300 hover:text-emerald-200 font-medium underline underline-offset-2"  

      >  

        Go to SME Loan Readiness Portal →  

      </Link>  

    </p>  

  </div>  
  

<p className="mt-4 text-[10px] text-slate-500">  

  Protected by enterprise-grade encryption • Role-based access  

  control • Audit-ready logs  

</p>  

</div>  

</div>  

</div>  

);  

}

```

2.25 Index.css

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
@import "tailwindcss";

```

```

/* Global Reset */
html, body {
  margin: 0;
  padding: 0;
  font-family: "Inter", system-ui, sans-serif;
}

/* Apply theme variables to body */
body {
  background-color: var(--bg);
  color: var(--text);
  transition: background 0.3s ease, color 0.3s ease;
}

/* Scrollbar */
::-webkit-scrollbar {
  width: 6px;
}
::-webkit-scrollbar-thumb {
  background: rgba(0, 0, 0, 0.3);
  border-radius: 3px;
}

/* Optional animation utility */
@keyframes pulse-soft {
  0%, 100% { opacity: 1; }
  50% { opacity: 0.75; }
}
.animate-pulse-soft {
  animation: pulse-soft 3s ease-in-out infinite;
}

```

2.26 main.jsx

```

/*
Name: Kahawaththa Liyanage Bawantha Harshana Janeshwara
Student ID: M24W0179
University: The Kyoto College of Graduate Studies for Informatics
Project: SME Loan Readiness Tool
Date: 2026-01-23
*/
import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";

import App from "./App.jsx";

// ADMIN / BANK PAGES
import PolicySettings from "./pages/PolicySettings.jsx";
import AdminDashboard from "./pages/AdminDashboard";
import BankDashboard from "./pages/BankDashboard.jsx";
import ManagerDashboard from "./pages/ManagerDashboard.jsx";

// SME PAGES
import SMELogin from "./pages/sme/SMELogin.jsx";
import SMERegister from "./pages/sme/SMERegister.jsx";
import SMEPortal from "./pages/sme/SMEPortal.jsx";

```

```

import SmeAnalysisPage from "./pages/sme/SmeAnalysisPage.jsx";
import SMEDashboard from "./pages/sme/SMEDashboard";

import "./index.css";      // Tailwind directives
import "./styles/global.css";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <BrowserRouter>
      <Routes>
        {/* LANDING PAGE (main homepage) */}
        <Route path="/" element={<App />} />

        {/* SME ROUTES */}
        <Route path="/sme" element={<SMEHome />} />
        <Route path="/sme/login" element={<SMELogin />} />
        <Route path="/sme/register" element={<SMERegister />} />
        <Route path="/sme/portal" element={<SMEPortal />} />
        <Route path="/sme/analysis/:businessId" element={<SmeAnalysisPage />} />
        <Route path="/sme/dashboard/:businessId" element={<SMEDashboard />} />

        {/* ADMIN / BANK ROUTES */}
        <Route path="/admin/dashboard" element={<AdminDashboard />} />
        <Route path="/admin/policy" element={<PolicySettings />} />
        <Route path="/bank/dashboard" element={<BankDashboard />} />
        <Route path="/manager/dashboard" element={<ManagerDashboard />} />
      </Routes>
    </BrowserRouter>
  </React.StrictMode>
);

```

Appendix B (User Guide for the SME Loan Readiness Tool)

1. Introduction

This appendix presents a concise user guide for the SME Loan Readiness Tool. The guide explains how different users interact with the system and how the platform supports SMEs in evaluating their financial readiness for loan applications. The objective of this guide is to enable users to operate the system effectively without requiring advanced technical or financial expertise.

2. User Roles and System Access

The SME Loan Readiness Tool is a web-based application accessed through a standard web browser. The system supports three user roles:

- **SME User:** Primary user who enters business data, completes questionnaires, views financial analysis, and downloads bank-ready reports.
- **Bank User:** Reviews structured SME readiness outputs and financial summaries.
- **Policy-Level User:** Accesses aggregated SME indicators and policy-level insights.

All users must authenticate using valid credentials before accessing the system.

3. SME User Workflow

After registration and login, SME users are directed to the dashboard, which displays the overall loan readiness score and key financial indicators. Users then complete a business profile by providing basic organizational details, followed by a structured financial questionnaire covering income, expenses, assets, liabilities, and existing loans.

Once the questionnaire is submitted, the system automatically calculates financial ratios and generates a loan readiness score. The results are presented using visual indicators and explanatory text to support easy understanding. In addition, AI-generated insights highlight strengths, weaknesses, and recommended actions. The AI Business Coach allows SMEs to receive practical guidance in simple language.

SME users can download a bank-ready PDF report that consolidates business information, financial ratios, readiness score, and AI insights. This report is designed to support structured and professional loan applications.

4. Bank and Policy-Level Interaction

Bank users can review SME readiness outputs and structured financial reports to support preliminary credit evaluation. The system improves information quality but does not replace internal bank credit assessment procedures.

Policy-level users can view aggregated SME indicators through the Central Policy Settings interface. The system allows simulation of policy variables and provides AI-generated recommendations, as well as quarterly summary reports for financial monitoring.