FINAL YEAR PROJECT REPORT

BS (SOFTWARE ENGINEERING)

BLOCKCHAIN BASED LOAN MANAGEMENT SYSTEM WITH SMART CONTRACT

SUBMITTED BY

MUHAMMAD SALEEM	**50438**
MUHAMMAD HARIS	**50741**

SUPERVISOR

DR. ABDUL AHAD ABRO

COORDINATOR

DR. AARIJ MAHMOOD HUSSAAN

FACULTY OF ENGINEERING, SCIENCE AND TECHNOLOGY

IQRA UNIVERSITY, KARACHI

MARCH 2024

FACULTY OF ENGINEERING, SCIENCE AND TECHNOLOGY

DEPARTMENT OF SOFTWARE ENGINEERING

FINAL YEAR PROJECT REPORT

BACHELOR OF SOFTWARE ENGINEERING

MUHAMMAD SALEEM (50438)

MUHAMMAD HARIS (50741)

PROJECT:
BLOCKCHAIN BASED LOAN MANAGEMENT SYSTEM WITH SMART CONTRACT

SUPERVISOR:

DR. ABDUL AHAD ABRO

MARCH 2024

ABSTRACT

Contemporary financial lending systems grapple with inherent limitations, such as reliance on a single service platform, transactional opaqueness, and susceptibility to cyber threats due to inadequate data privacy measures. This research introduces a pioneering solution called Lending over the Blockchain (LoC), a revolutionary finance and credit facility that harnesses the power of smart contracts on a blockchain hyper ledger fabric. LoC addresses the aforementioned challenges by providing unprecedented transparency, security, and operational efficiency.

To illustrate the practical implications of our proposal, we delve into the poverty level for Chinese loans, using it as a contextual example. A meticulously crafted digital accounting model facilitates the seamless transfer of assets between decentralized and centralized ledgers, while the integration of smart contract closing and unlocking methods augments the overall functionality of the system. Recognizing the paramount importance of safeguarding data privacy, we employ advanced techniques such as digital signatures and oracles to ensure the confidentiality and integrity of sensitive information.

An in-depth performance evaluation, considering both the opening rule and the chain rule, underscores the practical applicability of our blockchain-based loan management system in real-world lending scenarios. By capitalizing on the innovative features of blockchain technology and smart contracts, our proposed system emerges as a pivotal advancement in financial services, offering a secure, transparent, and highly efficient alternative to conventional lending platforms. This research lays the foundation for a transformative paradigm shift in the realm of financial transactions, promising a future where integrity, privacy, and efficiency coalesce seamlessly.

We have approved this manuscript for submission and presentation as fulfillment of Bachelor of Software Engineering/ Computer Science.

_____

Supervisor: Dr. Abdul Ahad Abro
Date: 17-03-2024

_____

Project Coordinator: Dr. Aarij Mahmood Hussaan
Date: 17-03-2024

DECLARATION

I hereby declare that the work has been done by myself to fulfill the requirement of the BS (Software Engineering) and no portion of the work contained in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

I hereby further declare that in the event of any infringement of the provision of the Act whether knowingly or unknowingly the university shall not be liable for the same in any manner whatsoever and undertake to indemnify and keep the university indemnified against all such claims and actions.

_____

© MUHAMMAD SALEEM [50438]

_____

© MUHAMMAD HARIS [50741]

ACKNOWLEDGEMENT

First, we thank Almighty Allah who praise us with the ability to think, work and deliver what we are assigned to do. Secondly, we must be grateful to our supervisor **"Dr. Abdul Ahad Abro"** who helps us in this project. We also acknowledge our teachers that throughout our studies helps us and guides us, departmental staff, university staff or other then this. We are also thankful to the FYP instructor **"Dr. Aarij Mahmood Hussaan"** for his precious support throughout the tenure as he is the best instructor for FYP who makes every student to be updated with the project progress and lead to the completion with great success within the time period given. We are also grateful to our family and friends, for supporting and encouraging us to complete this project. Finally, we would like to thank all the colleagues of IQRA University who have been with us in all difficult times with suggestions and supportive words which carry us to make this project a reality.

LIST OF ACRONYMS

1. LoC - Lending over the Blockchain
2. DeFi - Decentralized Finance
3. DApps - Decentralized Applications
4. ERC-20 - Ethereum Request for Comment 20
5. ERC-721 - Ethereum Request for Comment 721
6. BPM - Business Process Management
7. TPS - Transactions Per Second
8. FRCU - Financial Relief Credit Union
9. M2P - Model-to-Programming

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER – 1

1.0 Introduction

Growth of Information Technology have benefited Modern Financial and Banking services which helps to maintain the efficiency of financial services. [1]. More and more banks are implementing technological innovation due to maintain the efficiency, the increasing rate of innovations in these fields of mobile devices, application of artificial intelligence (AI), relevant big data analysis techniques, transfer of data to the cloud and blockchain technology. [2] to their business and information systems. Decentralized finance (DeFi) is a blockchain-based financial infrastructure that has recently gained a lot of traction. According to Butlerin (2013), a term usually applies to a protocol stack that is completely open, permission less, and highly accessible and relies on open smart contract platforms i.e. the Ethereum blockchain [3]. Basically, DeFi relies on open protocols and decentralized (DApps), Code is used to enforce agreements, ensure the security of transactions, and allow for the continued existence of legal state changes on a public Blockchain. With uncommon transparency, equal access rights, and minimal need for administrators, central clearance structures as a result of "smart contracts," this architecture is able to build an irreversible and highly interoperable financial system. pg. 2 We first need to look at ordinary server-based web applications in order to understand the originality of smart contracts. When a user interacts with such an application, they cannot observe the application's internal logic. Moreover, the user is not in control of the execution environment. Either one (or both) could be manipulated. As a result, the user has to trust the application service provider. Smart contracts mitigate both problems and ensure that an application runs as expected. The contract code is stored on the underlying blockchain and can therefore be publicly scrutinized. The contract's behavior is deterministic, and function calls (in the form of transactions) are processed by thousands of network participants in parallel, ensuring the execution's legitimacy. When the execution leads to state changes, for example, the change of account balances, these changes are subject to the blockchain network's consensus rules and will be reflected in and protected by the blockchain's state tree.

Few varieties of loans for reducing poverty are available in the financial sector. Each customer acquires small amount of loan due to the services is in single mod, financial institutions' loan services are higher in cost than other loan products in order to support customers groups of people and the risk management technique is not reliable. China has

announced the continuous poverty alleviation program for its rural areas [4], and more than CNU246 billion is issued for poverty alleviation loan in the end of 2016 [5]. It is important, a comprehensive range of services should be provided by a transparent and secure management system, especially for rural populations which do not have an appropriate financial infrastructure. Poverty alleviation loan, is a special loan product of FRCU set up for poor people lived in rural areas. The organizations, companies, individuals include bank, customer, financial department of government, civil affairs department of government and regulator of government have the role to participate. This loan is currently kept in the standard loan process management system. However, there are several limitations in the current systems that restrict users from getting the most out of the value of a loan for poverty alleviation.

1.1 Problem Statement

Providing a loan should be a simple process. One should check the client's eligibility to get the loan and then approve or deny the loan. Once approved, the customer should receive the funds. However, in traditional systems, this process is often chaotic for valid reasons. As the customer base increases, servicing loans becomes complex. Every customer has different terms and payment dates. Loan management systems help automate the entire loan lifecycle. A blockchain offers different protocol networks so the development of the blockchain and its transactions do not interfere with live transactions. These networks are used to develop, test, and deploy smart contracts and other transactions. Our Loan Management System based on blockchain with smart contract securely shares the details about transactions by organizing the network, this action prevents fraud in the system. In cirrus's core wallet, the admin creates an account for each user and updates the wallet information in their profile and the transaction is done using the Cirrus API.

1.2 Motivation

In today's rapidly evolving financial landscape, traditional loan management systems often struggle to keep pace with the dynamic needs of borrowers and lenders. Rising customer expectations, regulatory complexities, and the need for increased transparency have underscored the urgent requirement for innovative solutions that can streamline and enhance the lending process.

Enter blockchain technology and smart contracts: two revolutionary concepts that have the

potential to reshape the entire financial services industry. Blockchain, with its decentralized and immutable ledger system, offers unparalleled security and transparency, eliminating the need for intermediaries and reducing the risk of fraud and manipulation. Smart contracts, self-executing contracts with coded terms and conditions, further enhance efficiency by automating various stages of the loan lifecycle, from origination to repayment.

Our project delves deep into the transformative potential of blockchain and smart contracts in the realm of loan management. We meticulously analyze industry trends, regulatory developments, and technological innovations, unveiling the convergence of factors that have accelerated the demand for disruptive solutions in the financial services sector.

At the heart of our endeavor lies the intrinsic motivation to revolutionize traditional loan management practices, unlocking new levels of efficiency, security, and accessibility. By harnessing the power of blockchain technology and smart contracts, we envision a future where the lending process is seamless, transparent, and inclusive.

Our proposed solution aims to inspire confidence in its relevance and potential impact by addressing key pain points faced by borrowers, lenders, and regulators alike. From reducing operational costs and minimizing the risk of defaults to enhancing data privacy and promoting financial inclusion, our Blockchain-based Loan Management System with Smart Contracts offers a comprehensive suite of benefits that cater to the evolving needs of stakeholders across the financial ecosystem.

Through our unwavering commitment to innovation and excellence, we seek to usher in a new era of financial services—one where efficiency, transparency, and trust are not just ideals but tangible realities. Join us on this journey as we redefine the future of lending through the transformative power of blockchain and smart contracts.

1.3 Objective

We are outlining the desired outcomes and deliverables. We aim to leverage blockchain technology and smart contracts to enhance transparency, security, and efficiency in loan management processes. By articulating clear objectives, we provide a roadmap for the subsequent chapters and activities, enabling stakeholders to evaluate the success and efficacy of our proposed solution.

Research Objectives:

1. Investigate the feasibility and efficacy of implementing a blockchain-based loan management system with smart contracts.

2. Explore the potential impact of blockchain technology on enhancing transparency, security, and efficiency in the loan management process.

3. Evaluate the scalability and performance of the proposed system in handling a high volume of loan transactions.

4. Examine the challenges and limitations associated with integrating blockchain and smart contracts into traditional loan management systems.

5. Assess the usability and user satisfaction of the blockchain-based loan management system among stakeholders, including lenders and borrowers.

Academic Objectives:

1. Contribute to the existing body of knowledge in the field of blockchain technology by conducting research on its application in financial services, specifically loan management.

2. Generate empirical evidence through rigorous testing and evaluation of the proposed blockchain-based loan management system.

3. Publish research findings in peer-reviewed academic journals and present them at relevant conferences and seminars to contribute to academic discourse and promote knowledge dissemination.

4. Foster collaboration and knowledge exchange with other researchers and academic institutions working in the field of blockchain technology and financial services.

Management Objectives:

The main objective of the management position is to organize the meetings for discussions, check the status of the project, and submit the project on time.

1.4 Challenges

1. Blockchain Integration Challenges:

I. Configuring the blockchain software infrastructure to ensure compatibility with the loan management system requirements.

II. Addressing scalability issues associated with blockchain networks to accommodate a high volume of loan transactions.

III. Ensuring the security and integrity of the blockchain network to prevent unauthorized access and fraudulent activities.

IV. Optimizing blockchain consensus mechanisms to achieve efficient transaction processing and validation.

2. Hardware Configuration Challenges:

I. Identifying and configuring the appropriate hardware components to support the blockchain-based loan management system.

II. Ensuring sufficient computing power and storage capacity to handle the computational demands of blockchain operations.

III. Addressing potential hardware limitations, such as network bandwidth and latency, to optimize system performance and responsiveness.

3. Adaptability Challenges:

I. Ensuring seamless integration of the blockchain-based loan management system with existing banking infrastructure and legacy systems.

II. Addressing interoperability challenges between different blockchain platforms and protocols to facilitate data exchange and communication.

III. Adapting to regulatory requirements and compliance standards governing financial transactions and data privacy in different jurisdictions.

4. Front-end Design Challenges:

I. Designing a user-friendly and intuitive interface for the loan management system to enhance user experience and accessibility.

II. Ensuring cross-platform compatibility and responsiveness of the front-end interface across different devices and screen sizes.

III. Incorporating interactive features and visualizations to facilitate user engagement

and decision-making processes.

IV. Optimizing front-end performance and load times to minimize latency and enhance overall system responsiveness.

1.5 Structure of Report

The completion of Chapter One marks the foundational stage of our Blockchain-based Loan Management System with Smart Contracts project. This chapter serves as an introduction to the core concept of our innovative solution, outlining its purpose, objectives, and potential impact on the financial services industry.

As we go further into the content of chapter one the major descriptions regarding the objective of the project can be seen. This is where you will find vital information such as research objectives, academic objectives as well as management objectives. Furthermore, we have provided content regarding the challenges that can occur with the progression of the application as well as in the usage of the application. Lastly, the remaining structure of the project report is given as follows:

1.5.1 Chapter 2: Technology Background

This chapter will consist of our well researched literature review regarding all the related prior work of our project subject and technologies.

1.5.2 Chapter 3: Requirements & Methodology

This Chapter will discuss basic models of the system, in addition to that the chapter will also host functional and nonfunctional requirements of our project.

1.5.3 Chapter 4: Project Plan & Initial Design

This chapter will consist of all the detailed designs of the project that will help the developer in understanding the project implementation and creating an easy route in development of the system.

1.5.4 Chapter 5: Project Design & Development

This is the most significant chapter since it details the actual design and implementation of the concept. i.e., the phases of design and development.

1.5.5 Chapter 6: Testing

We will construct test cases in this chapter.

i. Perform front end (design testing), which may include user control testing, spelling checks, and alignment, among other things.
ii. Carry out backend testing (source code)
iii. Use a tool to conduct testing and incorporate the results in report.

1.5.6 Chapter 7: Conclusion

In this last chapter, we will conclude our work, share results including facts and figures, tables, and graphs to show your findings.

i Discuss limitations and challenges.
ii Discuss the work that will be done in the future.

CHAPTER – 2

2. Technology Background

This chapter delves into the foundational concepts and technologies underpinning the proposed blockchain-based loan management system. It provides an overview of blockchain technology, smart contracts, and their relevance to financial services.

2.1 Background of the technology:

In this chapter, we delve into the foundational technologies underpinning the development of the blockchain-based loan management system. These technologies include Ethereum, ASP.NET, C#, HTML, CSS, jQuery, Visual Studio, Crystal Report, SQL Server, and Bootstrap.

2.1.1 Ethereum

Ethereum is a decentralized, open-source blockchain platform that enables the creation and execution of smart contracts and decentralized applications (DApps). It provides a robust and secure infrastructure for building blockchain-based solutions, offering features such as permission less execution, immutability, and cryptographic security. Ethereum's native programming language, Solidity, is used for writing smart contracts, which govern the logic and behavior of transactions within the blockchain network.

2.1.2 ASP.NET

ASP.NET is a web application framework developed by Microsoft for building dynamic web applications and services. It provides a powerful set of tools and libraries for developing server-side web applications using languages such as C# and VB.NET. ASP.NET offers features such as model-view-controller (MVC) architecture, data access controls, and built-in security mechanisms, making it well-suited for developing robust and scalable web applications.

2.1.3 C#

C# is a modern, object-oriented programming language developed by Microsoft as part of the .NET framework. It is widely used for building a variety of applications, including web, desktop, and mobile applications. C# offers features such as type safety, garbage collection, and support for asynchronous programming, making it a versatile and powerful language for software development.

2.1.4 Visual Studio

Visual Studio is an integrated development environment (IDE) developed by Microsoft for building, debugging, and deploying software applications. It provides a comprehensive set of tools and features for developers working with .NET-based technologies, including code editors, project management tools, and debugging capabilities.

2.1.5 Crystal Report

Crystal Report is a business intelligence tool used for designing and generating reports from various data sources. It allows developers to create highly formatted and interactive reports for presenting data in meaningful ways. Crystal Report integrates seamlessly with ASP.NET and SQL Server, making it a popular choice for enterprise reporting solutions.

2.1.6 SQL Server

SQL Server is a relational database management system (RDBMS) developed by Microsoft for storing and managing structured data. It provides features such as data storage, retrieval, manipulation, and security, making it a robust and scalable solution for storing application data. SQL Server is commonly used in conjunction with ASP.NET for building data-driven web applications.

2.1.7 Bootstrap

Bootstrap is a front-end framework developed by Twitter for building responsive and mobile-first web projects. It provides a collection of CSS and JavaScript components for creating user interfaces that adapt seamlessly to different screen sizes and devices. Bootstrap simplifies the process of designing and developing web interfaces, helping developers create visually appealing and user-friendly websites.

2.1.8 HTML, CSS, jQuery

HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and jQuery are fundamental technologies used for building the front-end of web applications. HTML provides the structure and content of web pages, while CSS is used for styling and layout. jQuery is a JavaScript library that simplifies DOM manipulation and event handling, enabling developers to create interactive and dynamic web interfaces.

2.3 Literature Review:

Blockchain is a decentralized, dependable and secure distributed database and transaction system where everyone is able to share information. It is first introduced by Bitcoin as a distributed bookkeeping system to prevent double-spending [7]. The technical community has developed a renewed interest in taking advantage of the fundamental distributed database of transactions to address other relevant issues, such the fairness of information sharing [8, 9]. In recent years, there have been a number of blockchain frameworks proposals appearing, such as Ripple [10], Ethereum [11], Corda [12], Hyperledger [13] and Hyperledger Fabric [14], among others. A vital phase towards a blockchain-based loan management mechanism involves the use of Ethereum's smart contracts. These smart contracts are created using the pg. 4 commonly used programming language Solidity. Self-executing contracts, in which the conditions of the agreement are clearly encoded into software, are made easier by solidity. These "smart contracts" maintain the logic associated with loan management as well as the conditions of the loan arrangement and the loan. Developers may efficiently specify and carry out the required operations within the Ethereum blockchain network by utilizing Solidity. In order to construct blockchain-based loan management systems, Solidity and Ethereum have been integrated. This literature review aims to provide an overview of current research and efforts in this area. It investigates how Solidity is used, the benefits Ethereum offers, and the difficulties experienced in this situation. Our research aims at a combined application of the M2P paradigm and the most recent blockchain technologies. The M2P modeling paradigm provides a theoretical framework for building up a pipeline from the actual modeling methods to the current operationalization or interpretative systems [16]. The design models can be checked by formal methods, namely Alloy and YAWL [15, 17–20]. Service innovation is a crucial approach in the recent enterprise environment to realize digital transformation [21].

CHAPTER – 3

3.1 Introduction:

In this chapter we will discuss about how much work is done on the development of our project according to the project plan. This chapter will cover the in-detail process and objective of the project. As our app is built on windows that are being reused within the system itself the developers had to take a systematic approach for the app to work smoothly. Our project plan is strategically planned with the Gantt chart and other organizational tools. Each activity has specific time period allotted according to the complexity of the task, which is why the days in work may vary.

We will also discuss in detail about the Functional, Non-Functional and Hardware requirements of our project. The functional requirement are taken in to full consideration as they are the necessary part in order to get the basic requirement by the project such as, camera detection and learning objective, while on the other hand, the non-functional requirement such as, settings and feedback are also thoroughly planned.

3.2 Project Plan:

1.12	FYP-II Timeline	01/09/2023	30/11/2023	13w	0%	Open
1.13	Iteration 1: design System Architecture, Us...	01/09/2023	21/09/2023	3w	0%	Open
1.14	Iteration 2: Core Features Development & ...	21/09/2023	27/09/2023	1w	0%	Open
1.15	Testing: Login, Signup, home page, bug tes...	27/09/2023	01/10/2023	5d	0%	Open
1.16	Iteration 3: Additional Features Developm...	02/10/2023	22/10/2023	3w	0%	Open
1.17	Iteration 4: Create Deployment Plan ,Depl...	23/10/2023	19/11/2023	4w	0%	Open
1.18	Testing of implemented Parts: Bug Testing, ...	19/11/2023	09/12/2023	3w	0%	Open
1.19	Iteration 5: Final Testing , bug , user testing...	10/12/2023	23/12/2023	2w	0%	Open
1.20	Iteration 6: Project Review and Closer	23/12/2023	05/01/2024	2w	0%	Open

Figure 1: Project Plan

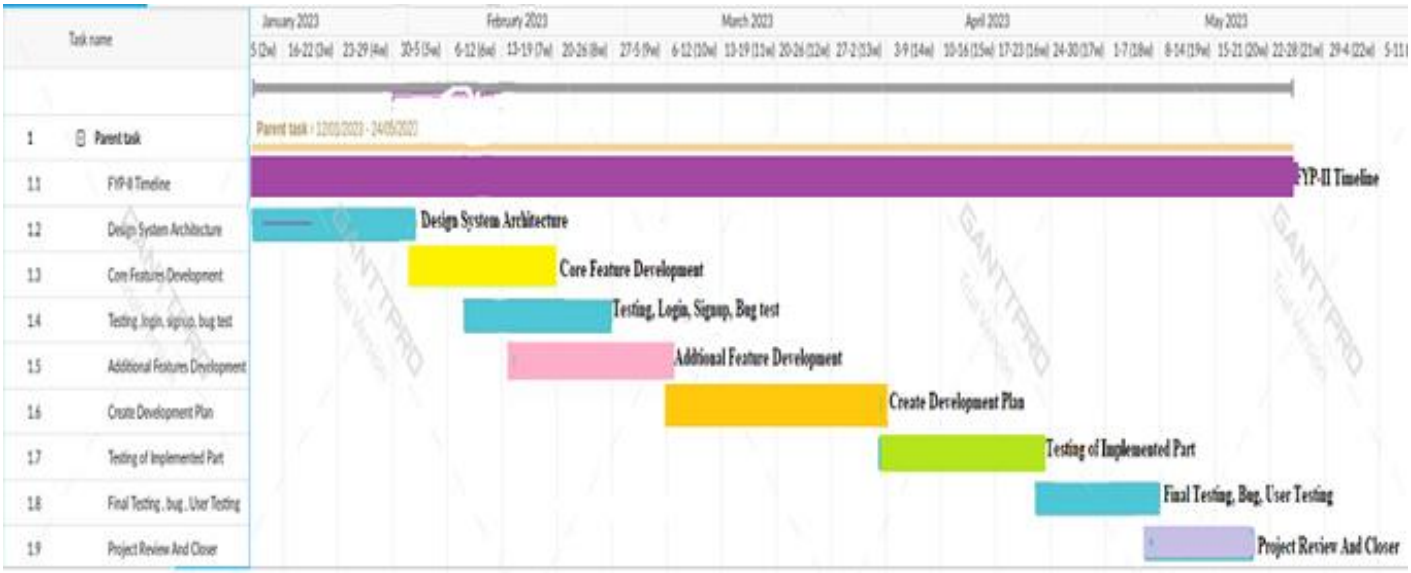


Figure 2: Gantt Chart

3.3 Functional Requirements:

1. **User Registration**: The system should allow users to register and create accounts with appropriate identification and financial information.

2. **Loan Application**: Users should be able to submit loan applications, providing necessary details such as loan amount, duration, and purpose.

3. **Eligibility Assessment**: The system should verify and evaluate the eligibility of borrowers based on predefined criteria, credit scoring algorithms, and available funds from lenders.

4. **Lender Selection**: The system should match borrowers with suitable lenders based on their lending preferences, interest rates, and available funds.

5. **Loan Approval**: The system should automate the loan approval process, verifying borrower information and determining loan terms and conditions.

6. **Smart Contract Creation**: The system should generate smart contracts that define the loan agreement, including repayment schedule, interest rates, penalties, and other contractual obligations.

7. **Loan Disbursement**: Once a loan is approved, the system should securely disburse funds to the borrower's designated account.

8. **Repayment Tracking**: The system should track loan repayments according to the agreed-upon schedule, ensuring accuracy and recording payment history.

9. **Penalty Application**: If a borrower defaults or misses payments, the system should apply penalties automatically based on the terms defined in the smart contract.

10. **User Communication**: The system should facilitate communication between

borrowers, lenders, and administrators through notifications, alerts, and messaging features.

3.3.1 User Panel:

The user panel functionality includes the following features:

1. User registration and authentication
2. Profile management (update personal information, change password)
3. Loan application submission
4. Loan approval workflow (assessment, approval/rejection)
5. Loan repayment management
6. Transaction history tracking
7. Notification system for important updates

.

3.4 Non-Functional Requirements:

1. **Performance:** The system should be capable of handling a high volume of concurrent loan transactions efficiently and provide responsive user experiences**.**

2. **Scalability:** The system should be scalable, allowing it to accommodate increasing numbers of users, loans, and transactions without compromising performance or functionality.

3. **Security**: The system should implement robust security measures to protect user data, prevent unauthorized access, and ensure the integrity of the blockchain and smart contracts.

4. **Privacy:** The system should incorporate privacy features to safeguard sensitive user information and comply with data protection regulations.

5. **Reliability:** The system should be highly reliable, minimizing downtime and ensuring continuous availability for users.

6. **Usability:** The system should have an intuitive and user-friendly interface, making it easy for borrowers, lenders, and administrators to navigate, apply for loans, and access relevant information.

7. **Data Backup and Recovery:** The system should have robust backup and recovery mechanisms in place to protect against data loss and ensure business continuity.

8. **Compliance:** The system should adhere to relevant legal and regulatory requirements, such as anti-money laundering (AML) and know-your-customer (KYC) regulations.

9. **Maintenance and Support:** The system should be easily maintainable, allowing for regular updates, bug fixes, and enhancements. Additionally, it should provide efficient technical support to address user queries and issues promptly.

3.5 Hardware Requirements:

i. Adequate storage capacity to store transaction data and user information.

ii. Sufficient processing power to handle computation-intensive tasks such as smart contract execution.

iii. Network infrastructure to ensure high-speed connectivity and data transfer between client and server.

3.6 Summary:

In this chapter, a detailed Project Plan, Functional, Non-Functional requirements and other planning mechanisms are discussed in detail that will be required in our project. We have also mentioned an introduction regarding our web application how we can perform our task so we make a milestone chart in this first we describe our task week wise in summary activity and then we make a Gantt chart according to summary activity. In Gantt chart we were describing task name or duration for implementation of our "Blockchain Based Loan Management System with Smart Contract" After Gantt chart we describe Non-Functional requirements of our web application the requirements are system settings and feedback

CHAPTER – 4

4.1 Introduction:

In this chapter, we are going to discuss about the design and specification of our project, in which we elaborate our project deeply with the help of diagrams like we gather all the information related to our application then set the framework to show the flow of the application so that the application flow will be easily understand. We have used different diagrams for the complete flow of our application to make it understandable to user. In this phase, we also discussing the data flow diagrams (DFD'S), entity relationship diagram (ERD'S), UML in detailed as according to our application. (Pramudyo). These diagrams show the system work flow and specification of our application to make it user friendly. It also shows how every screen work flow is working with the help of diagram. After all information has been gathered and design has been created so now, the development has started in order to make sure that it is able to be used by user.

The purpose for making the data flow diagram and entity diagram to guide the direction of our system that how we perform each and every thing and also show the flow of our application specifically like in implementation. In detail this will provide a clear understanding of the overall coding of the system for the people who are on user bases. each diagram is detail with all functional input and output of the system, making sure that the system runs smoothly.

4.2 Data Flow Diagram:

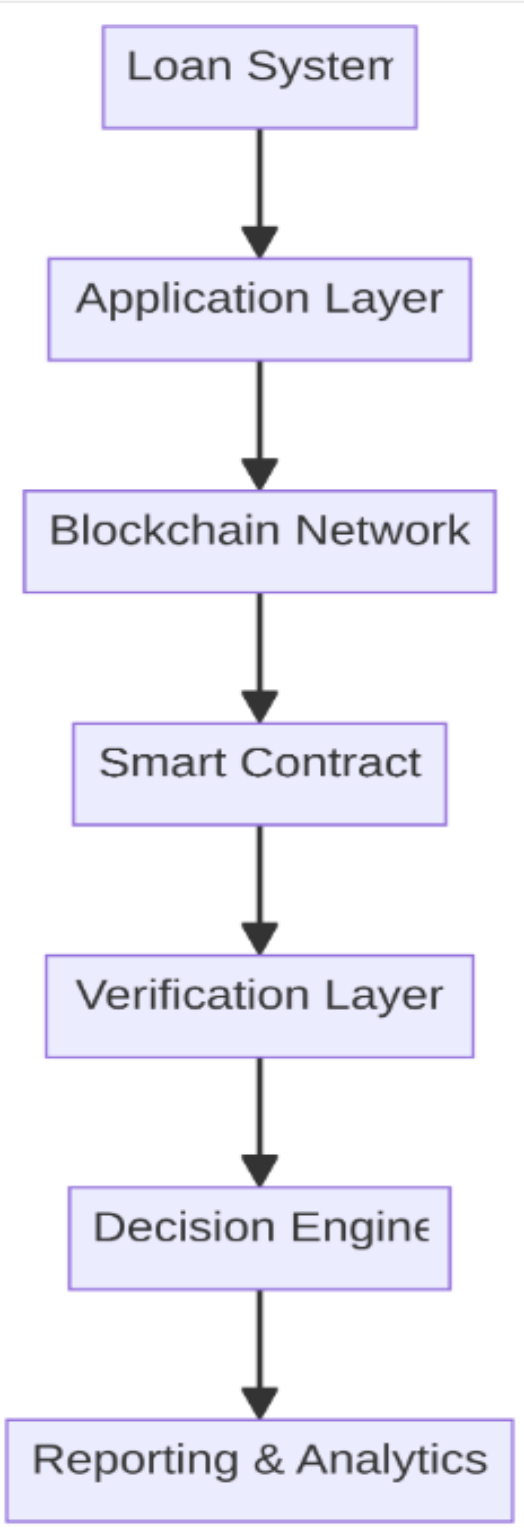


Figure 3: DFD

4.2 Entity Relationship Diagram

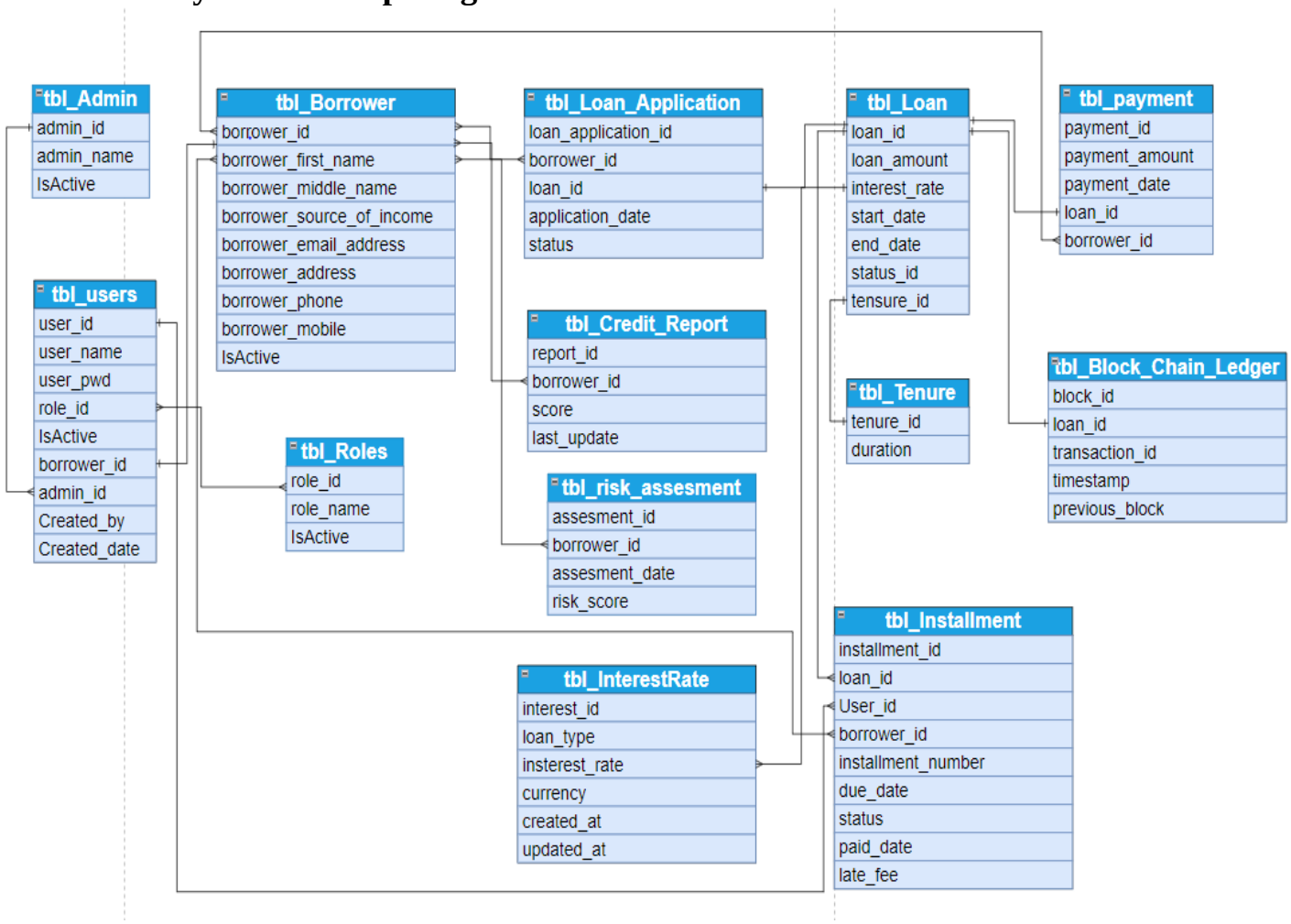


Figure 4: Entity Relationship Diagram

4.3 Use Cases

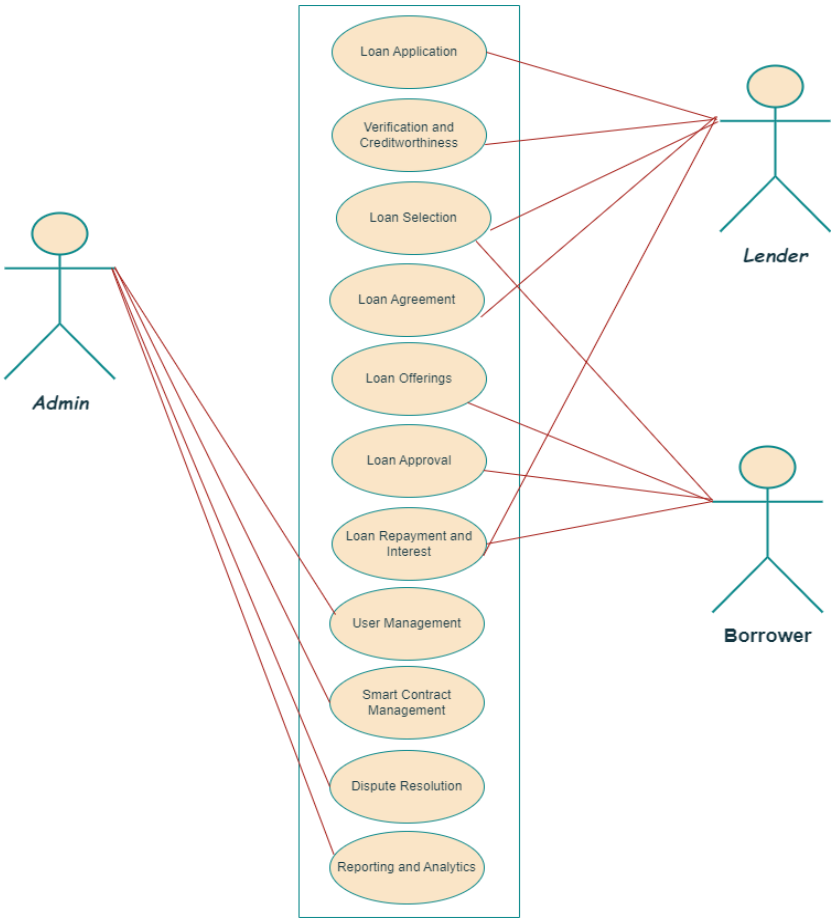


Figure 5: Dashboard

4.4 Use Cases

I. User Registration

Use Case Name:	User Registration	
ID:	01	
Priority:	High	
Actors Involved:	Borrower (or any user who wants to register)	
Brief Description:	This test case validates the functionality of the user registration process. It ensures that new users can successfully register by providing required information such as username, email, password, etc., and that their information is correctly stored in the system.	
Pre- Condition:		
Post- Condition:		
Normal Flow of Events:	Actor Actions: i)Navigate to the registration page. ii) Enter valid registration details (username, email, and password). iii) Click on the "Register" button.	System Response: User is registered successfully and redirected to the login page.

Table 1: Sign Up

II. User Login

Use Case Name:	**User login**	
ID:	**02**	
Priority:	**High**	
Actors Involved:	**Borrower (or any registered user)**	
Brief Description:	**This test case verifies the user login functionality. It ensures that registered users can securely log in to the system using their credentials and gain access to their accounts.**	
Pre- Condition:	**The login page is accessible.**	
Post- Condition:	**The user is successfully logged in and directed to the dashboard.**	
Normal Flow of Events:	**Actor Actions:** **1. Navigate to the login page.** **2. Enter valid login credentials (username/email, password).** **3. Click on the "Login" button.**	**System Response:** **User is logged in successfully and directed to the dashboard.**

Table 2: Login

III. Loan Application Submission

Use Case Name:	**Loan Application Submission**	
ID:	**03**	
Priority:	**High**	
Actors Involved:	**Borrower (the user applying for the loan)**	
Brief Description:	**This test case tests the process of submitting a loan application. It validates that borrowers can fill out the necessary loan application form with required details such as loan amount, duration, purpose, personal information, etc., and submit it successfully for review.**	
Pre- Condition:	**The borrower is logged in and on the loan application page.**	
Post- Condition:	**The loan application is submitted, and its status is updated to "Pending Approval."**	
Normal Flow of Events:	**Actor Actions:** **1. Login as a borrower.** **2. Navigate to the loan application page.** **3. Fill in the required loan application details.** **4. Click on the "Submit" button.**	**System Response:** **Loan application is submitted successfully, and status is updated to "Pending Approval."**

Table 3: Loan Application

IV. Loan Approval

Use Case Name:	**Loan Approval**	
ID:	**04**	
Priority:	**High**	
Actors Involved:	**Lender (the entity responsible for approving loans)**	
Brief Description:	**This test case checks the loan approval process. It verifies that lenders can review loan applications, assess borrower eligibility based on predefined criteria, and approve or reject loan requests accordingly.**	
Pre- Condition:	**The lender is logged in and on the loan approval dashboard.**	
Post- Condition:	**The selected loan application is approved, and its status is updated accordingly.**	
Normal Flow of Events:	**Actor Actions:** **1. Login as a lender.** **2. Navigate to the loan approval dashboard.** **3. Review pending loan applications.** **4. Select an application to approve.** **5. Click on the "Approve" button.**	**System Response:** **Loan application is approved successfully, and smart contract is generated.**

Table 4: Loan Approval

V. Loan Disbursement

Use Case Name:	**Loan Disbursement**	
ID:	**05**	
Priority:	**High**	
Actors Involved:	**Lender (the entity responsible for disbursing approved loans)**	
Brief Description:	**This test case validates the loan disbursement process. It ensures that approved loans are disbursed correctly to borrowers' designated accounts, and the disbursed amount matches the approved loan amount.**	
Pre- Condition:	**The loan application is approved, and all necessary disbursement details are provided.**	
Post- Condition:	**The loan amount is disbursed to the borrower's successfully.**	
Normal Flow of Events:	**Actor Actions:** **1. Verify that the approved loan application status is "Approved."** **2. Navigate to the loan disbursement page.** **3. Enter the necessary details for loan disbursement (amount, account details).** **4. Click on the "Disburse" button.**	**System Response:** **Loan amount is disbursed to the borrower's account successfully.**

Table 5: Loan Disbursement

VI. Loan Repayment:

Use Case Name:	**Loan Repayment**	
ID:	**07**	
Priority:	**High**	
Actors Involved:	**Borrower (the user repaying the loan)**	
Brief Description:	**This test case tests the loan repayment process. It verifies that borrowers can make loan repayments according to the agreed-upon schedule, and the system accurately records and updates the repayment status.**	
Pre- Condition:	**The borrower is logged in and on the loan repayment page.**	
Post- Condition:	**The loan repayment is processed successfully, and the outstanding balance is updated.**	
Normal Flow of Events:	**Actor Actions:** **1. Login as a borrower.** **2. Navigate to the loan repayment page.** **3. Enter the repayment amount.** **4. Click on the "Repay" button.**	**System Response:** **Loan repayment is processed successfully, and the outstanding balance is updated.**

Table 6: Loan Repayment

VII. Loan Default Handling

Use Case Name:	**Loan Default Handling**	
ID:	**07**	
Priority:	**High**	
Actors Involved:	**System (the automated processes or administrators responsible for handling loan defaults)**	
Brief Description:	**This test case checks how the system handles loan defaults. It validates that appropriate actions are taken when borrowers fail to repay loans as per the agreed terms, such as applying penalties, notifying borrowers, and updating loan status accordingly.**	
Pre- Condition:	**A loan is unpaid beyond the due date.**	
Post- Condition:	**The system applies penalties or initiates recovery processes for the defaulted loan.**	
Normal Flow of Events:	**Actor Actions:** **1. Allow a loan to remain unpaid beyond the due date.** **2. Monitor the loan status.**	**System Response:** **System automatically applies penalties or initiates recovery processes for defaulted loans.**

Table 7: Loan Default Handling

4.5 Summary:

In this chapter, we met with the data flow diagrams, entity relationship diagrams and uml diagram. Our application covered the all-major modules, which were used to fulfill the requirements so the data flow diagram (dfd's) and entity relationship diagram (erd's) elaborate this in detailed because when we gathered the information so it show the flow of our system specifically like in implementation we make sure that the development is start or not then after this testing process will be occurred so if there is any error occur so it must be solved at that time then we monitor our application by time to time to make sure that everything is complete or not that why we make data flow diagram and entity relationship diagram to focus on our mistake. In spite of everything, information has been accumulated and layout has been created so now, the implementation has been initiated according to the applications' requirement to make sure that it is beneficial for the user on a long run. This application has a highly responsive nature because of the fact that the major specification of this application is to connect with camera, setting, feedback, and progress chart of child.

CHAPTER – 5

5.1 Introduction:

In this chapter we are discussing aspects which are used in our project, and prototype design which is generally used to evaluate a new design to enhance precision by system analysts and users and frontend and backend design of our project. We are also discussing about the database queries which are used in firebase and some external libraries and we are showing screenshots of our application as it fulfills user requirements. We have briefly provided a few clarifications about the sort of functionalities available on the system, source code of validation and etc.

5.2 Prototype Design:

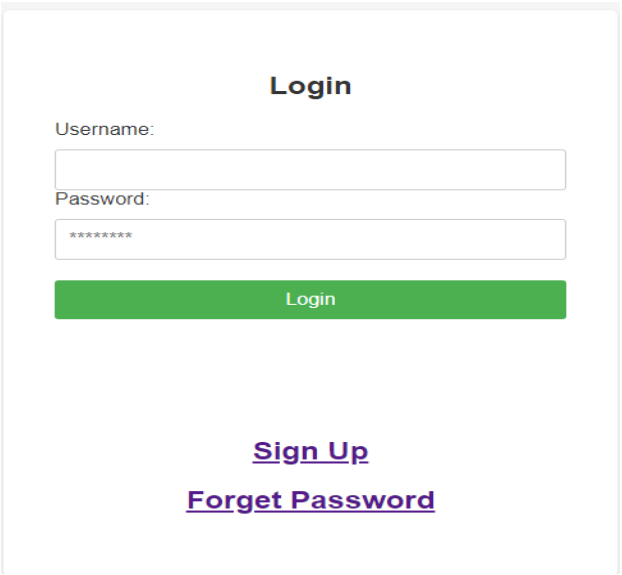
<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract
Date: 16/03/2024
Screen Name: login Panel</td></tr>
<tr><td>Screen: < 1 of 10 >
Link from screen: Main Window
Link to screen: User</td><td>Screen Description:
Password textbox, login button, Text.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Use inputs data to use the app. If he/she is a new user, user chooses the register button.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: textbox, Password textbox, login button, Text.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white, Dark blue, Green</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)
Arial(12dp) (16dp) (8dp)</td><td>Still images: loginback.jpg</td></tr>
</table>

Table 8: Prototype 1

<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract
Date: 16/03/2024

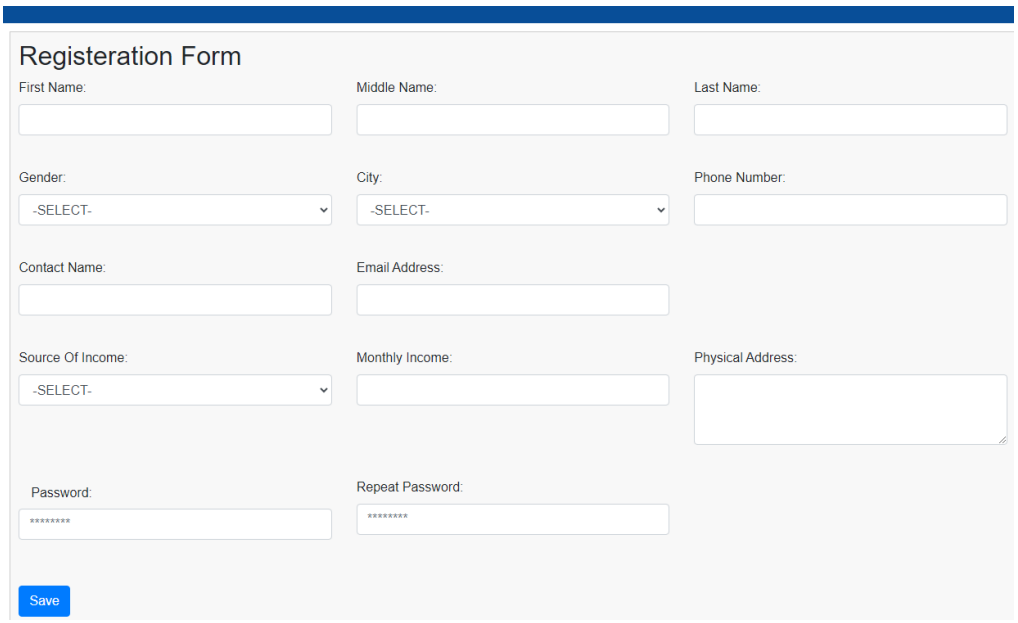
Screen Name: Register</td></tr>
<tr><td>Screen: < 2 of 10 >
Link from screen: Login Window
Link to screen: User</td><td>Screen Description:
Email textbox, Password textbox, login button, Text.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Allows new users to register themselves by providing necessary information.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: textbox, Password textbox, login button, Text.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)
Arial(20dp)</td><td>Still images: Registrationblock.jpg</td></tr>
</table>

Table 9: Prototype 2

<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract
Date: 16/03/2024

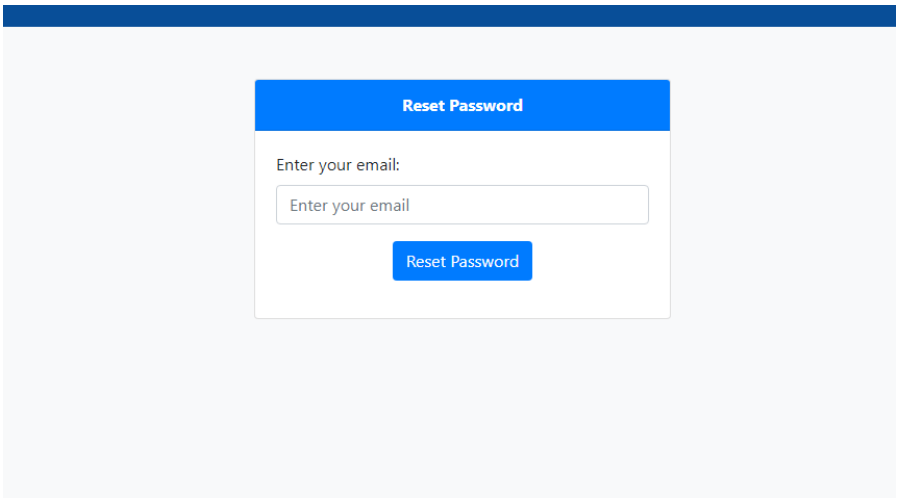
Screen Name: Register</td></tr>
<tr><td>Screen: < 3 of 10 >
Link from screen: Login Window
Link to screen: User</td><td>Screen Description:
Email textbox, Password textbox, login button, Text.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Password reset functionality allowing users to securely reset their passwords.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: textbox, Password textbox, reset button, Text.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)
Arial(20dp)</td><td>Still images: resetPassword.jpg</td></tr>
</table>

Table 10: Prototype 3

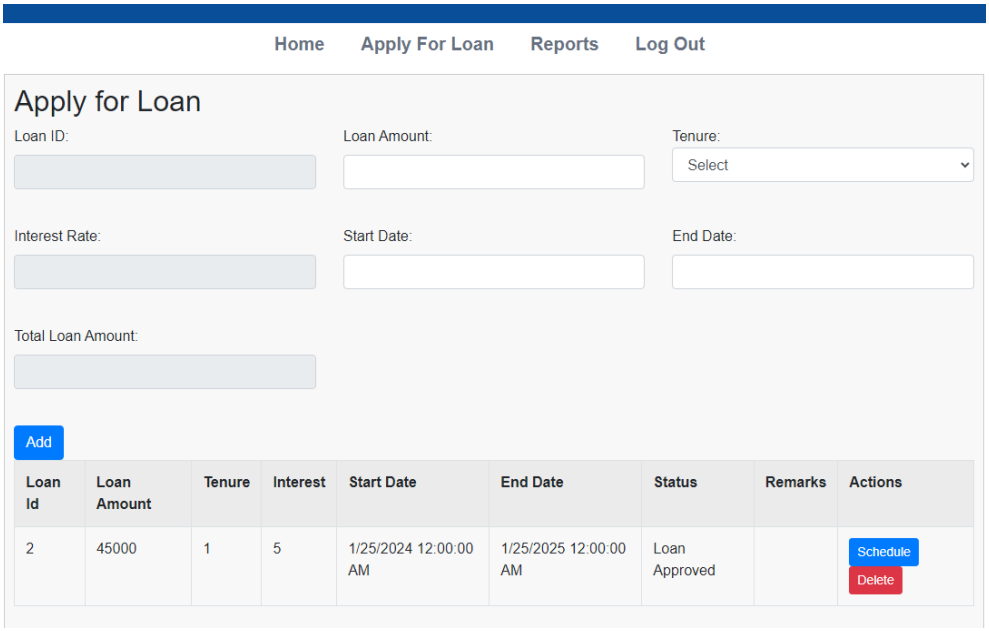
<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract
Date: 16/03/2024
Screen Name: Apply for Loan</td></tr>
<tr><td>Screen: < 4 of 10 >
Link from screen: Apply for Loan
Link to screen: User</td><td>Screen Description:
textbox, delete button, Add, Schedule.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Enabling users to submit loan requests with necessary details for processing.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: textbox, delete button, Add, Schedule.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white, red and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)
Arial(20dp)</td><td>Still images: applyLoan.jpg</td></tr>
</table>

Table 11: Prototype 4

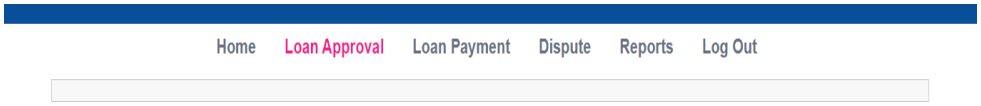
<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract
Date: 16/03/2024
Screen Name: Loan Approval</td></tr>
<tr><td>Screen: < 5 of 10 >
Link from screen: Loan Approval
Link to screen: Admin</td><td>Screen Description:
Home, Loan approval, Loan payment, Dispute, Reports, Logout.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Loan approval mechanism to assess and approve loan applications based on predefined criteria.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: Home, Loan approval, Loan payment, Dispute, Reports, Logout.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white, purple and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)
Arial(20dp)</td><td>Still images: loanApproval.jpg</td></tr>
</table>

Table 12: Prototype 5

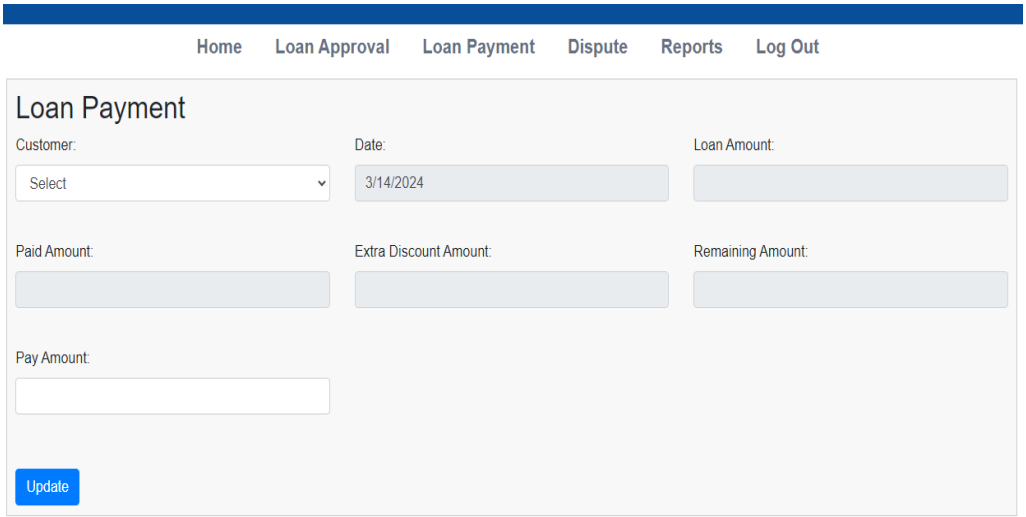
<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract
Date: 16/03/2024
Screen Name: Loan Payment</td></tr>
<tr><td>Screen: < 6 of 10 >
Link from screen: Loan Payment
Link to screen: Admin</td><td>Screen Description:
Home, Loan approval, Loan payment, Dispute, Reports, Logout.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Loan payment system allowing users to make scheduled payments towards their loans.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: Home, Loan approval, Loan payment, Dispute, Reports, Logout.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)
Arial(20dp)</td><td>Still images: loanPayment.jpg</td></tr>
</table>

Table 13: Prototype 6

Project Title: Blockchain Based Loan Management System with Smart Contract

Date: 16/3/2024

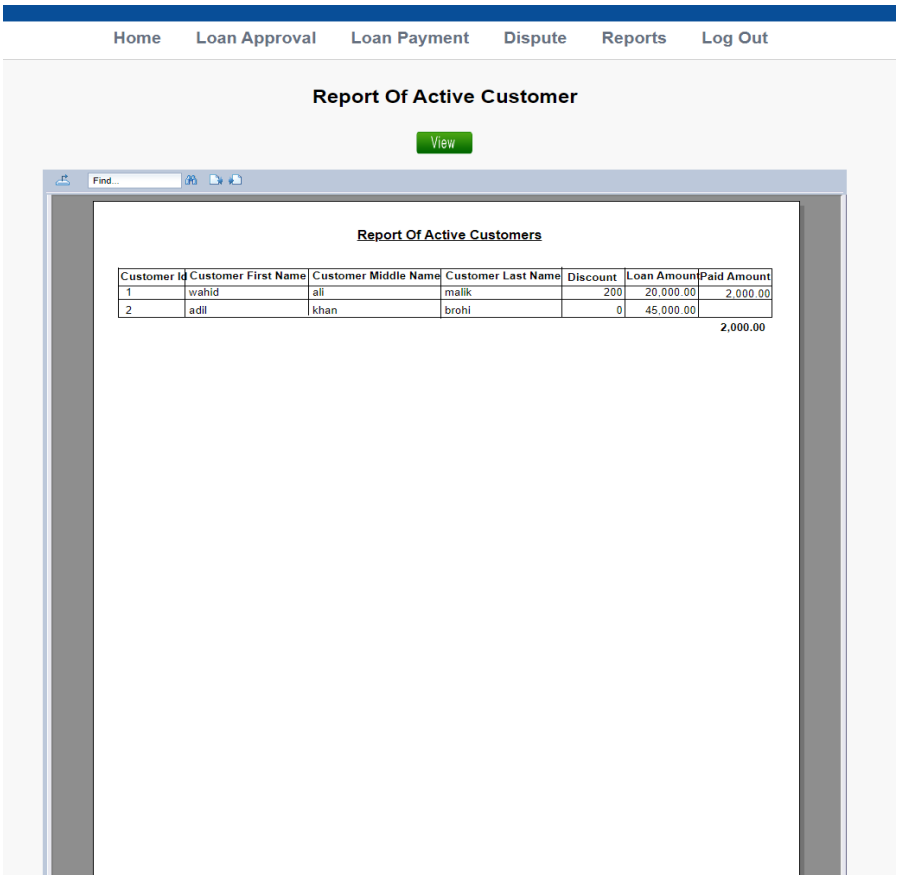
Screen Name: Report of Active Customers

Screen: < 7 of 10 > **Link from screen:** Report of Active Customers **Link to screen:** Admin	**Screen Description:** Home, Loan approval, Loan payment, Dispute, Reports, Logout.

Functionality/Interactivity:

Create a report generation feature to display active customer details based on their current status and activity

Screen Design:



Background: Home, Loan approval, Loan payment, Dispute, Reports, Logout.	**Audio: none**
Color scheme: white	**Video: none**
Text attributes: Aclonica(24sp) Arial(20dp)	**Still images:** reportActiveCustomer.jpg

Table 14: Prototype 7

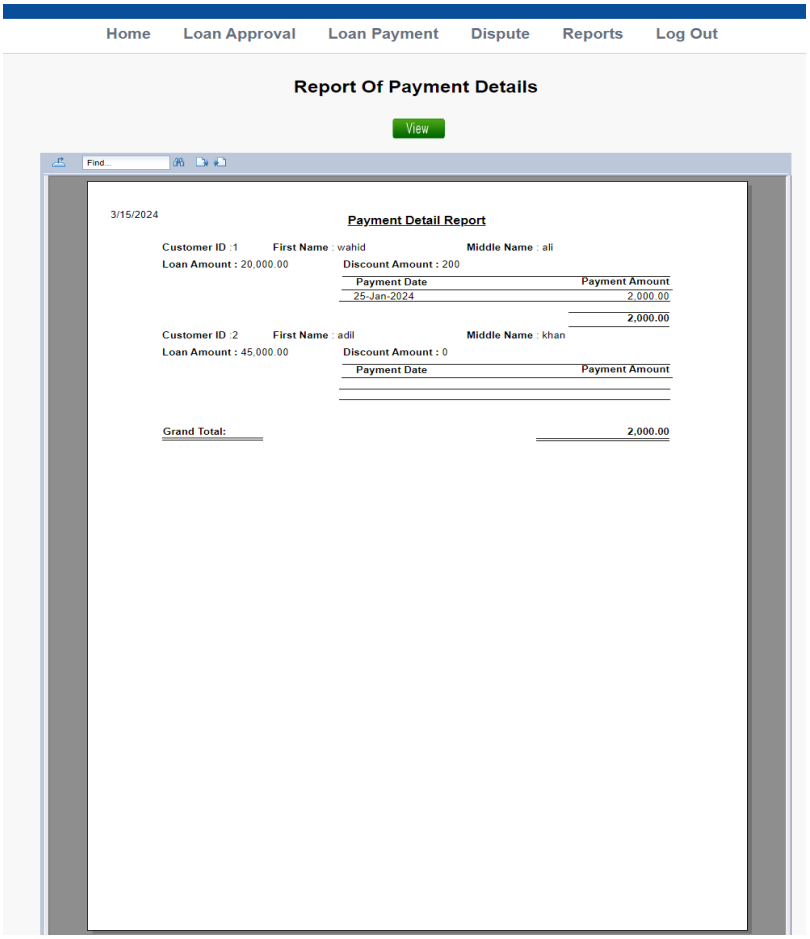
<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract

Date: 16/03/2024

Screen Name: Report of Payment Details</td></tr>
<tr><td>Screen: < 8 of 10 >

Link from screen: Report of Payment Details

Link to screen: Admin</td><td>Screen Description:

Home, Loan approval, Loan payment, Dispute, Reports, Logout.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Payment details report features to provide comprehensive information on all transactions, including amounts, dates, and related details.</td></tr>
<tr><td colspan="2">Screen Design:
 
</td></tr>
<tr><td>Background: Home, Loan approval, Loan payment, Dispute, Reports, Logout.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white and Green</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)

Arial(20dp)</td><td>Still images: reportPaymentDetails.jpg</td></tr>
</table>

Table 15: Prototype 8

<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract

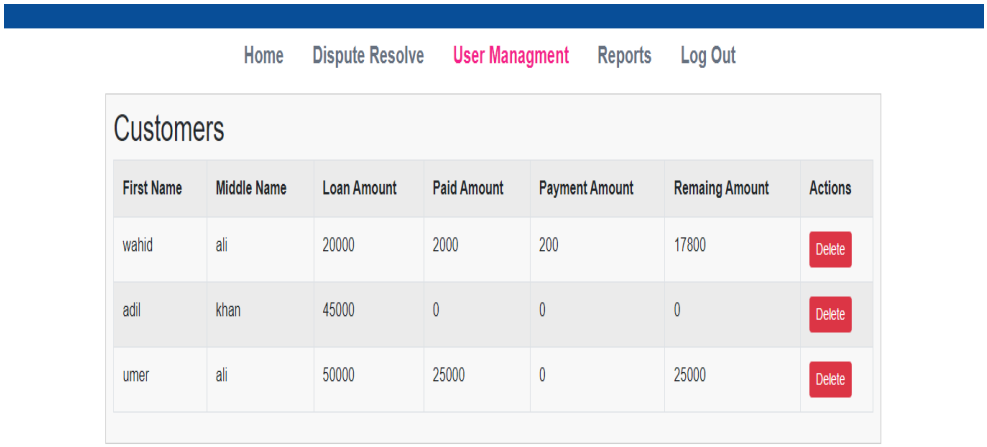
Date: 16/03/2024

Screen Name: User Management</td></tr>
<tr><td>Screen: < 9 of 10 >

Link from screen: User Management

Link to screen: Super Admin</td><td>Screen Description:

Home, Dispute Resolve, User Management, Reports, Logout.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
User management functionality enabling administrators to manage user accounts, permissions, and profiles efficiently.</td></tr>
<tr><td colspan="2">Screen Design:

 
</td></tr>
<tr><td>Background: Home, Dispute Resolve, User Management, Reports, Logout.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white, red and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)

Arial(20dp)</td><td>Still images: userManagment.jpg</td></tr>
</table>

Table 16: Prototype 9

<table>
<tr><td colspan="2">Project Title: Blockchain Based Loan Management System with Smart Contract

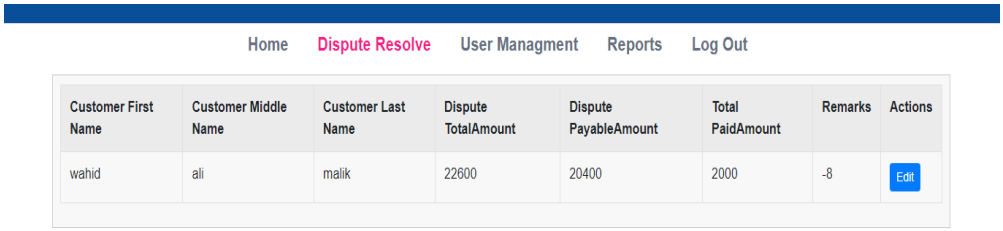
Date: 16/03/2024

Screen Name: Dispute Resolve</td></tr>
<tr><td>Screen: < 10 of 10 >

Link from screen: Dispute Resolve

Link to screen: Super Admin</td><td>Screen Description:

Home, Dispute Resolve, User Management, Reports, Logout.</td></tr>
<tr><td colspan="2">Functionality/Interactivity:
Dispute resolution functionality to address and resolve conflicts or issues raised by users effectively.</td></tr>
<tr><td colspan="2">Screen Design:

 
</td></tr>
<tr><td>Background: Home, Dispute Resolve, User Management, Reports, Logout.</td><td>Audio: none</td></tr>
<tr><td>Color scheme: white, purple and blue</td><td>Video: none</td></tr>
<tr><td>Text attributes: Aclonica(24sp)

Arial(20dp)</td><td>Still images: disputeResolve.jpg</td></tr>
</table>

Table 17: Prototype 10

5.3 Screenshots:

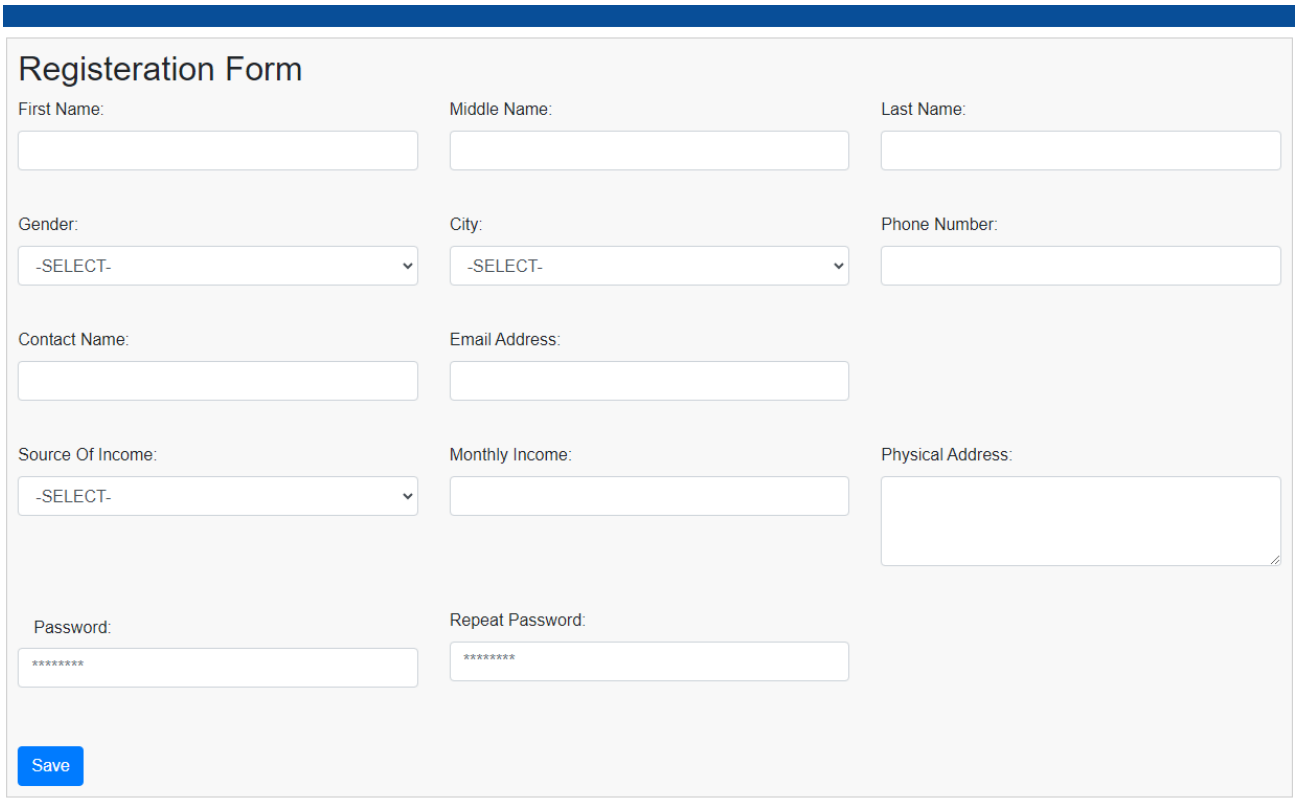


Figure 6: Registration Form

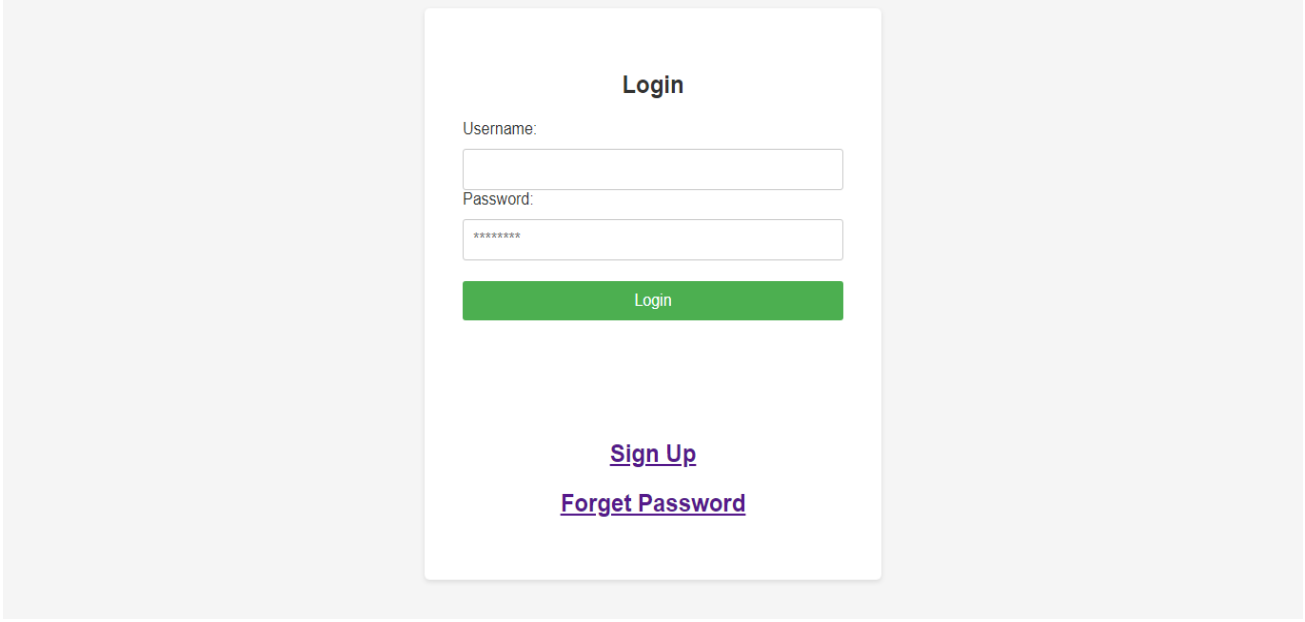


Figure 7: Login Form

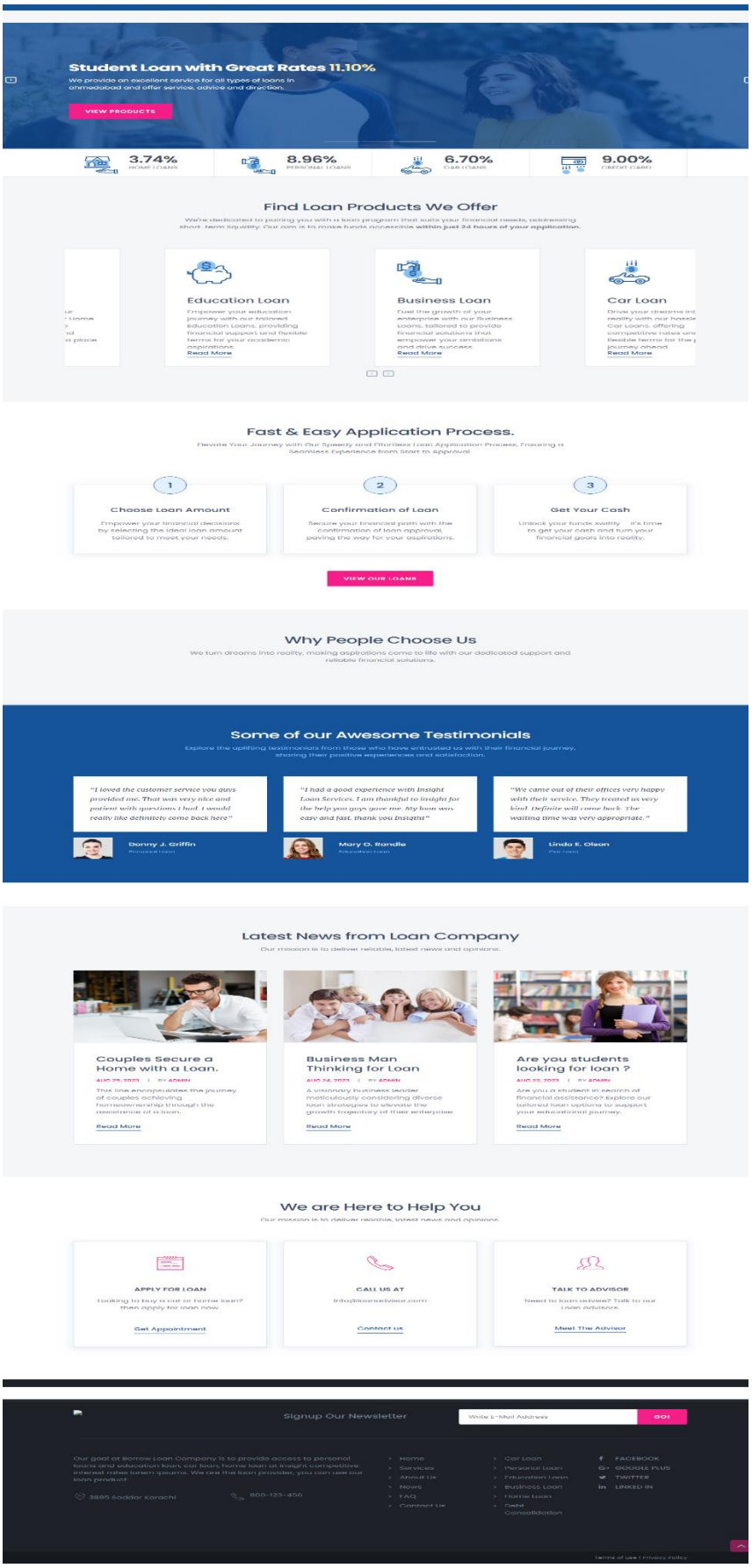


Figure 8: User Interface

Figure 9: Admin Panel

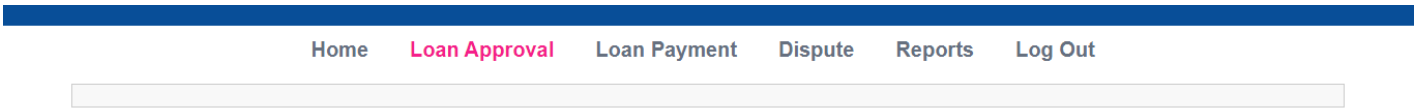


Figure 10: Loan Approval

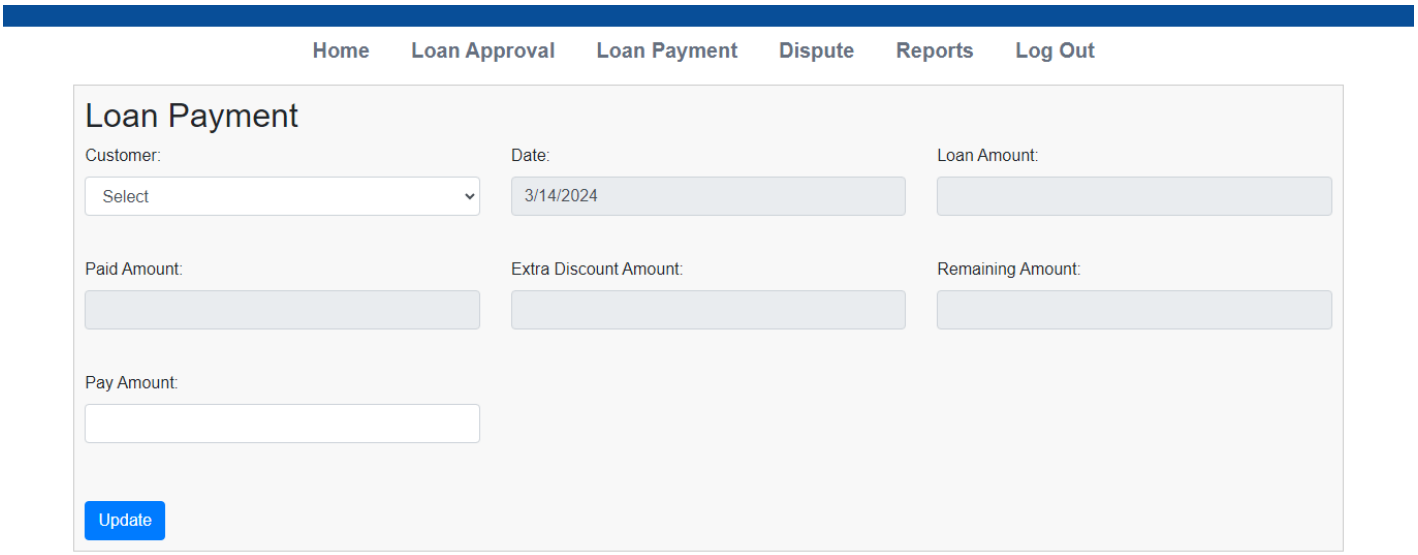


Figure 11: Loan Payment

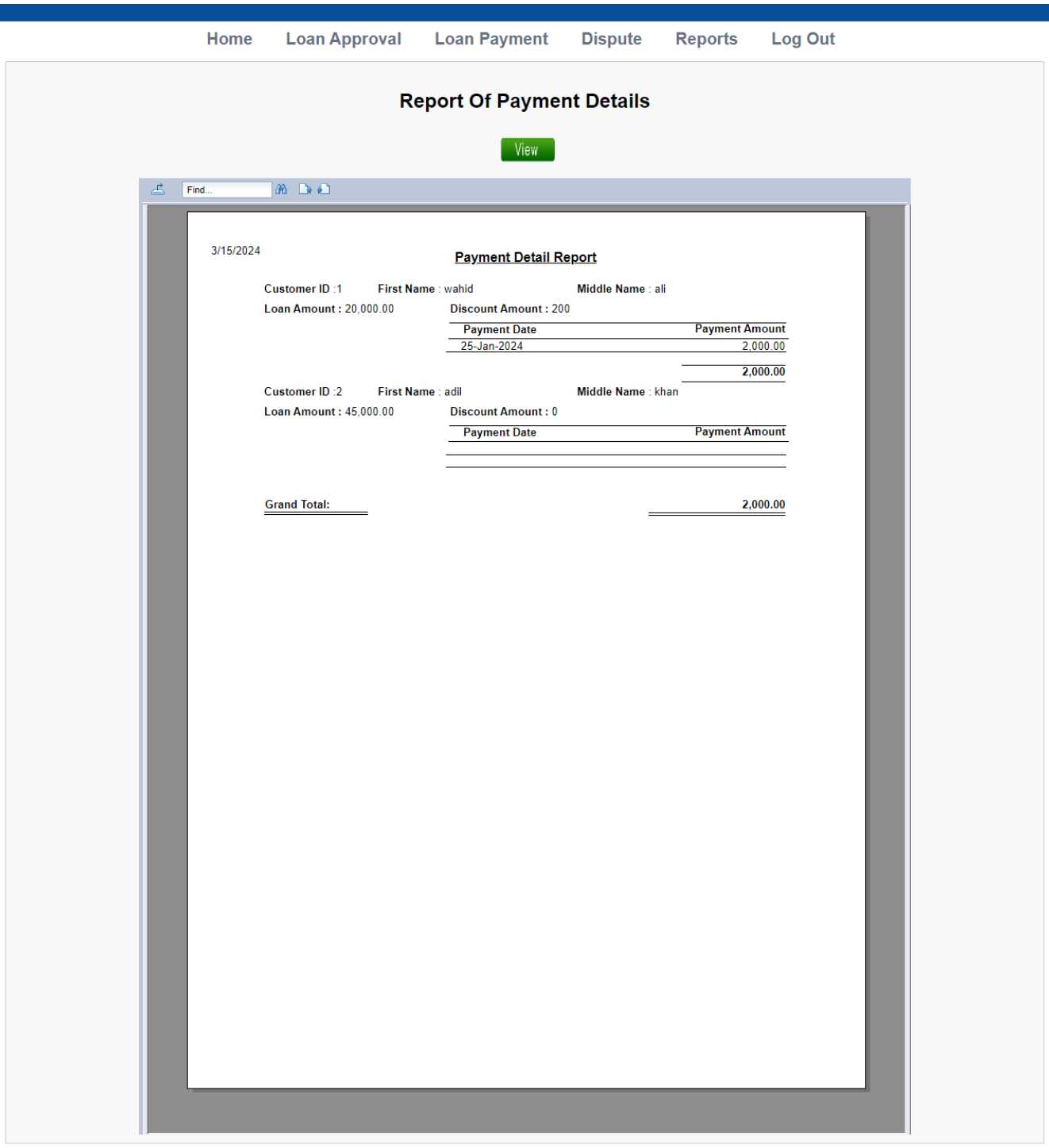


Figure 12: Report of Payment Details

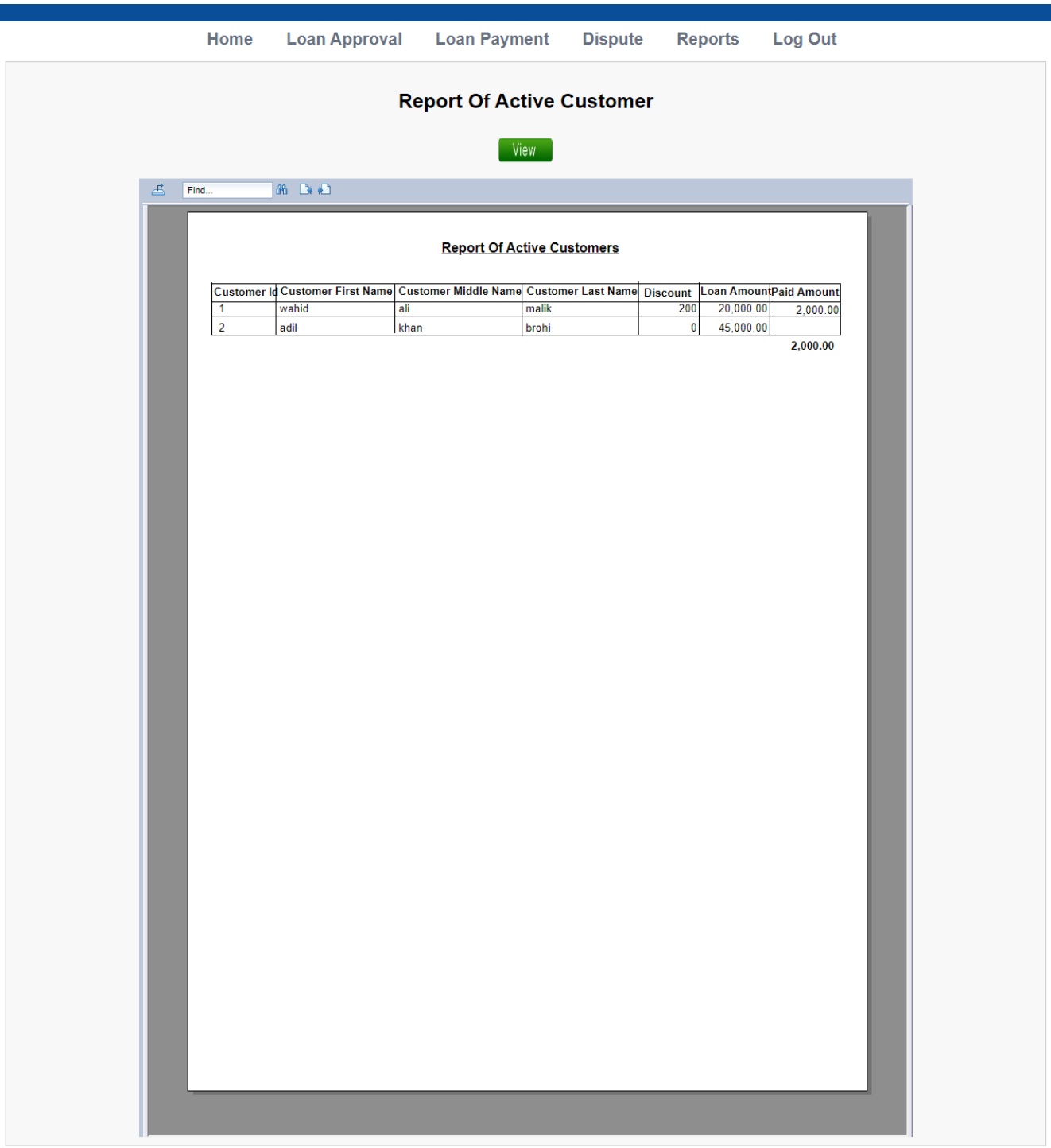


Figure 13: Report of Active Customers

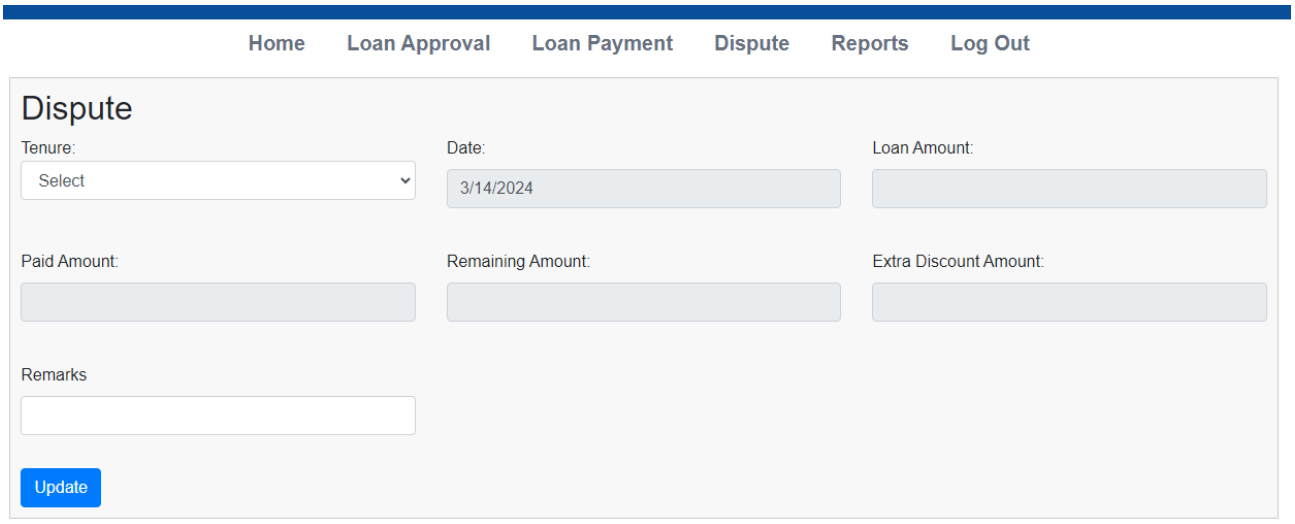


Figure 14: Dispute

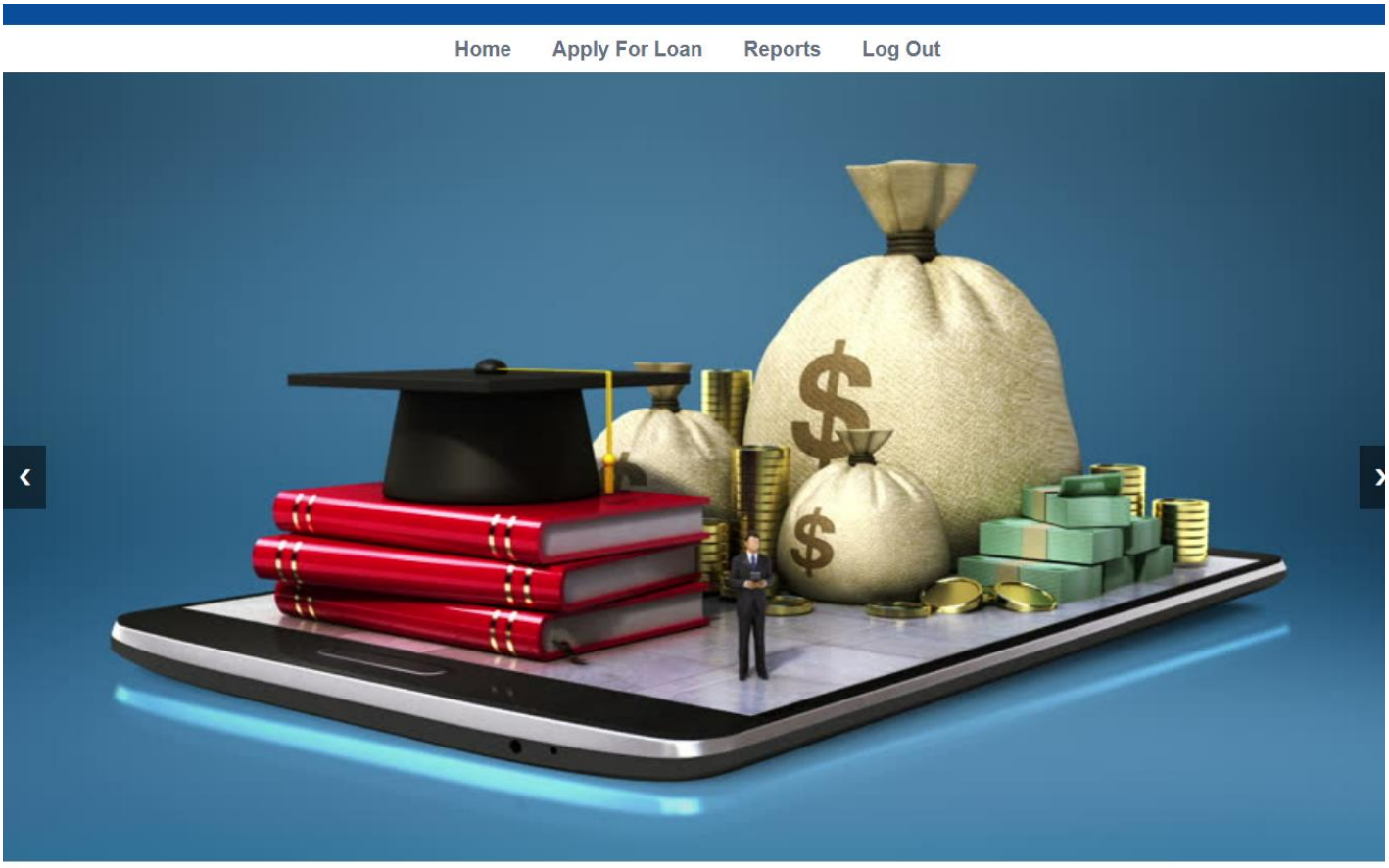


Figure 15: User Panel

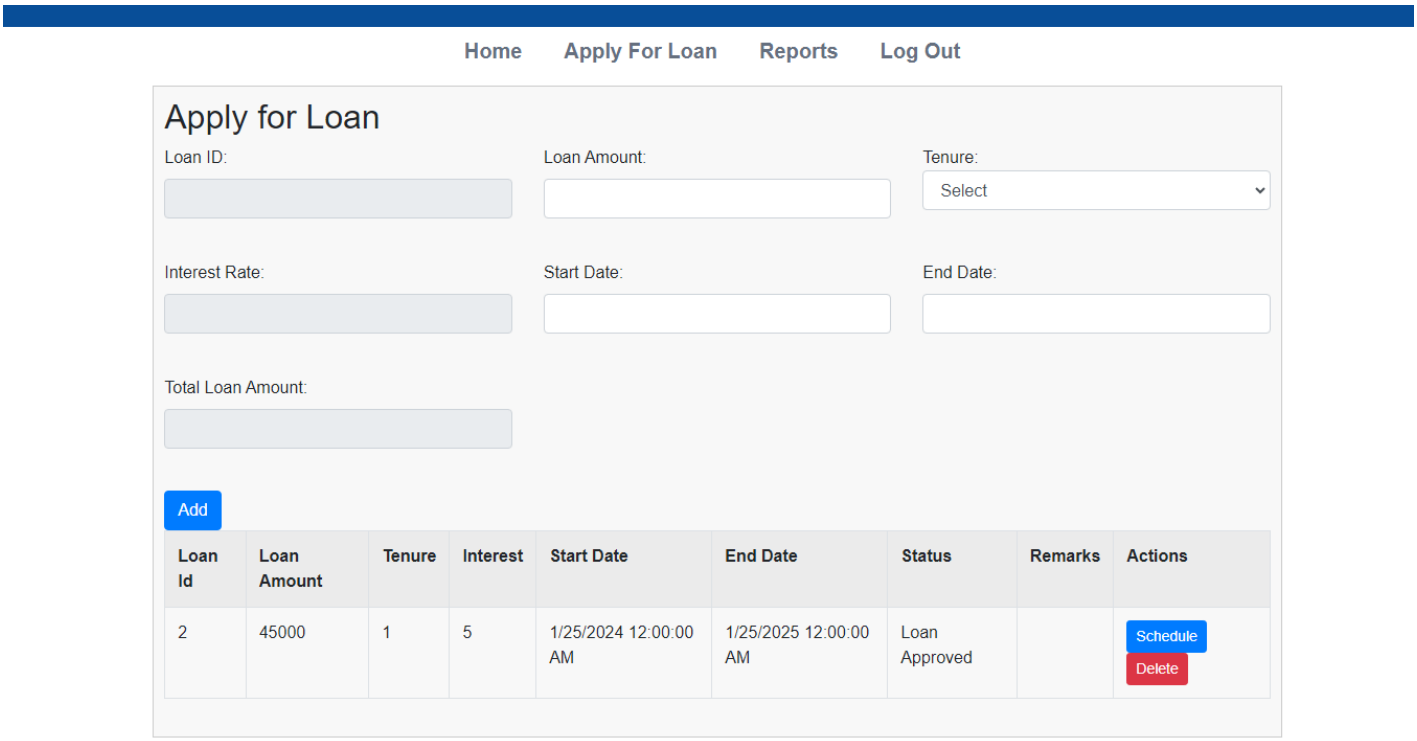


Figure 16: Apply for Loan

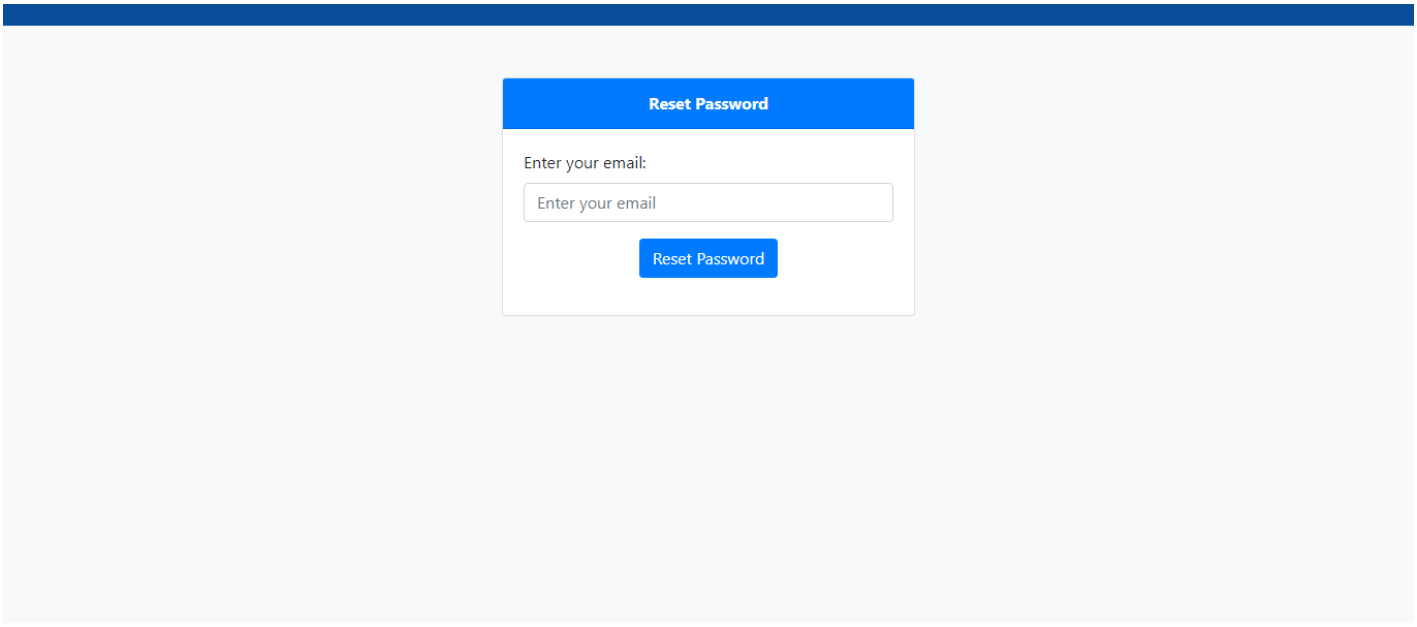


Figure 17: Forget Password

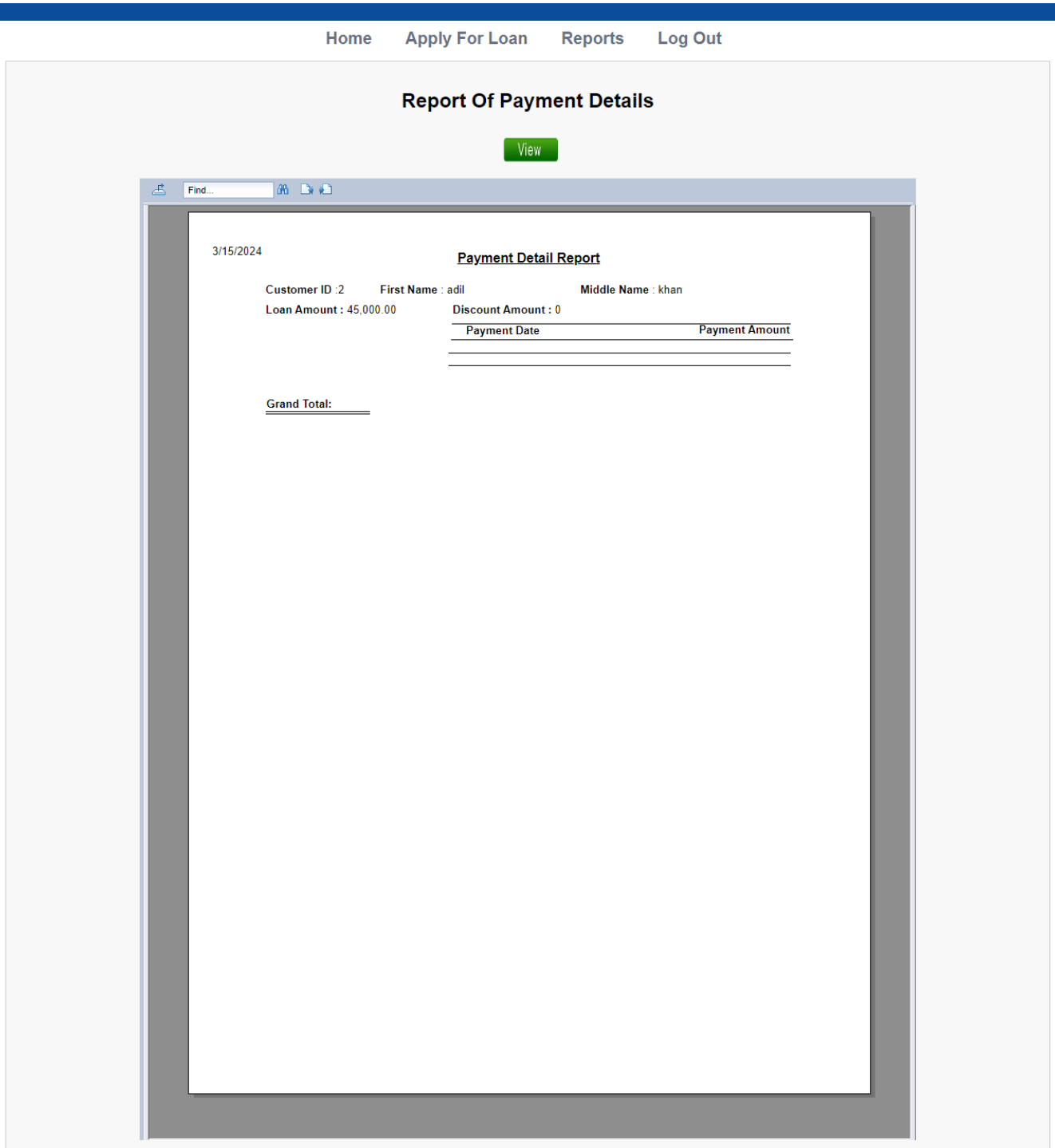


Figure 18: Report of Payment Details

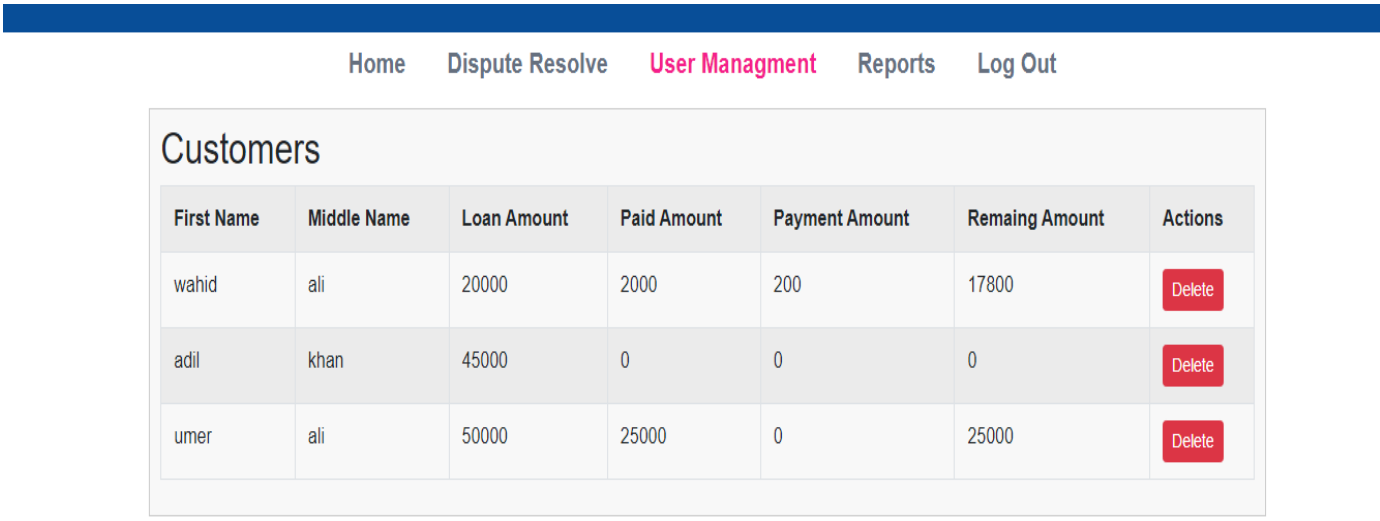


Figure 19: User Management

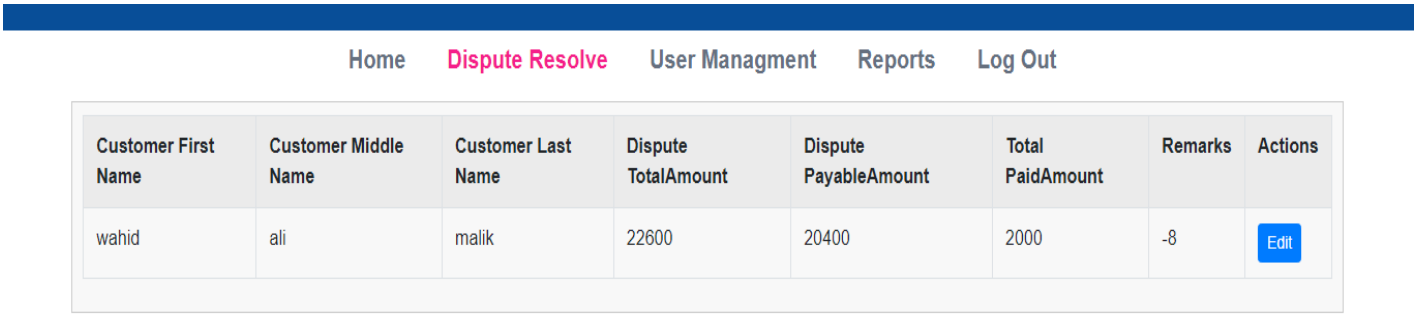


Figure 20: Dispute Resolve

5.4 Summary:

This Chapter consist of system prototype and development in which we explained about the prototype, frontend and backend design and some screenshot of our application. In this chapter we briefly give some explanation about the type of error happen on the system. So from the next part some information flows are given to know how to system assignation works. The prototype design provides the clear view of working screens one by one. The prototype table consist of project title, observation date, screen name, screen number, and link of the screen. The table also informs about the overall description and functionality of the screen, while providing a specific description trough background, color scheme, video/audio connections and still images.

CHAPTER – 6

6.1 Introduction:

In this chapter, we have discussed about the test cases to determine whether the Application is working the way it should and producing the expected results. We also test cases and usability test cases to test our application. This will help the readers to know about all the minor as well as the major working options of the application. After the completion of the implementation phase testing plays a vital role for making sure that the system works properly. Once the Application was developed the testing phase started. Each and every screen and button were tested according to the requirements and functionalities. Even though the application is complex it does use I treated functionality as one window is being used multiple times for different kinds of functionalities. do to the re usability of our code the total test cases of the application turned out to be 11. Each test case has the requirement reference, project name, application name while providing all the details of the test case. in detail attributes of the test case, such as test case ID, Test case description, test steps, expected result, pass or fail status, preparation date, running date end the date at which it was tested.

6.2 Test Cases:

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.1	**Test Type**	Functionality
Test Case Description	To test that the button (view our loan) on Intro screen next.		
Test Steps	Auto opens to next screen.		
Expected Result	Open Sign in page Screen on Application.		
Actual Result	Open Sign in page Screen.		
Pass/Fail	pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 18: Test case 1

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.2	**Test Type**	Functionality
Test Case Description	To test if the input fields takes input with appropriate input types i.e. Username & Password. To test if the login button works as well.		
Test Steps	1. Write username & password. 2. Click the Login Button.		
Expected Result	1. Takes data from input fields. 2. Open Login window on Application.		
Actual Result	Takes input field and login button.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 19: Test case 2

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.3	**Test Type**	Functionality
Test Case Description	To test if the sign-up button works and proceed to registration page.		
Test Steps	Click the sign in button to proceed to next page.		
Expected Result	Proceed to registration page.		
Actual Result	Opening the registration page.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 20: Test case 3

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.4	**Test Type**	Functionality
Test Case Description	To test if the input fields takes input with appropriate input types on registration page. To test if the save button works as well.		
Test Steps	1. Write appropriate input fields with correct data/input types. 2. Click the save Button.		
Expected Result	1. Takes data from input fields. 2. Open Login window on Application after save button is clicked.		
Actual Result	Takes input field and save button.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 21: Test case 4

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.5	**Test Type**	Functionality
Test Case Description	To test that the fields on Login screen password should be mandatory		
Test Steps	Blank Textbox and click on Login		
Expected Result	Login screen should be display the error.		
Actual Result	Invalid Credential error.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 22: Test case 5

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.6	**Test Type**	Functionality
Test Case Description	To test that the fields on Login screen Login id and password should be mandatory		
Test Steps	click on Login		
Expected Result	Open the main User Dashboard Screen		
Actual Result	main User Dashboard Screen Open		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 23: Test case 6

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.7	**Test Type**	Functionality
Test Case Description	To test that the fields on Register screen email should be mandatory		
Test Steps	click on save		
Expected Result	Empty Field error. Blank Spaces not Allowed.		
Actual Result	Invalid Error.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 24: Test case 7

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.8	**Test Type**	Functionality
Test Case Description	To test that the fields on Register screen Password should be mandatory		
Test Steps	click on save		
Expected Result	Empty Field error. Special Character Should Allowed.		
Actual Result	Invalid Error.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 25: Test case 8

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.9	**Test Type**	Functionality
Test Case Description	To test that the fields on Register screen Confirm Password should be mandatory		
Test Steps	click on save		
Expected Result	Empty Field error. Special Character Should Allowed.		
Actual Result	Invalid Error.		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 26: Test case 9

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.10	**Test Type**	Functionality
Test Case Description	To test the Loan Management system dashboard Screen.		
Test Steps	click on Login		
Expected Result	Open Loan Management system dashboard.		
Actual Result	Display Loan Management system dashboard		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 27: Test case 10

Requirement Reference	1	**Project Name**	Loan management system with smart contract
Test Case Id	1.11	**Test Type**	Functionality
Test Case Description	To test that the different pages on Loan Management system Screen must be mandatorily present and working.		
Test Steps	click on perspective pages		
Expected Result	Open Loan Management system screen window. Displays All different pages Options		
Actual Result	Open Loan Management system screen window. Displays All different pages Options		
Pass/Fail	Pass		
Date Prepared	9th Mar 2024		
Date Run	10th Mar 2024		
Prepared By	Muhammad Haris and Muhammad Saleem		
Tested By	Muhammad Haris and Muhammad Saleem		

Table 28: Test case 11

6.3 Summary:

We test our software to get our expected results of our application or whether a system under test satisfies requirements or works correctly. After test cases, we get satisfied results the usability test case.

.

CHAPTER – 7

7.1 Introduction:

In this final chapter, we conclude our exploration of the blockchain-based loan management system with smart contracts. We summarize the key findings, reflect on the system's limitations and challenges, discuss potential areas for future work, and present concluding remarks. In this chapter all the major and minor work will be discussed. We have included the limitations of the system in order to help the users to understand the system better. The future work section will provide an overview to the enhancement that can be made with this application. With the passing time the application has a lot of vacancies for better additions.

7.2 System Limitations and Challenges:

Despite the significant advancements made in developing the loan management system, certain limitations and challenges persist. These include:

I. Complexity of blockchain technology: Implementing blockchain and smart contracts requires expertise and careful consideration of various technical aspects, which may pose challenges during development and maintenance.

II. Scalability concerns: As the system grows and handles a larger volume of transactions, scalability becomes a critical factor. Ensuring that the system can accommodate increasing user demand while maintaining performance and efficiency is a challenge that needs to be addressed.

III. Integration with existing infrastructure: Integrating the new system with legacy systems and financial institutions' infrastructure may present compatibility issues and require extensive testing and validation.

7.3 Future Work:

There are several avenues for future research and development to further enhance the blockchain-based loan management system:

Optimization of smart contracts: Continuously refining and optimizing smart contracts to improve efficiency, reduce gas costs, and enhance security.

Enhancing user experience: Incorporating user feedback and iteratively improving the system's user interface and functionality to enhance usability and satisfaction.

7.4 Conclusion:

In conclusion, the blockchain-based loan management system with smart contracts represents a significant advancement in the field of financial services. By leveraging blockchain technology and smart contracts, the system offers unparalleled transparency, security, and efficiency in loan transactions. Despite the challenges and limitations, the system holds immense potential to revolutionize traditional lending practices and pave the way for a more inclusive and accessible financial ecosystem. With ongoing innovation and collaboration, we are optimistic about the system's ability to drive positive change and empower individuals and communities worldwide.

REFERENCES

[1] R. J. Shiller, The new financial order: Risk in the 21st century, Princeton University Press, 2009.

[2] P. Schueffel, Taming the beast: a scientific definition of fintech.

[3] L. Meng, Evaluating china's poverty alleviation program: a regression discontinuity approach, Journal of Public Economics 101 (2013) 1–11.

[4] Blockchain-Based Business Process Management (BPM) for Finance: The Case of Credit and Claim Requests B Molnár, G Pisoni, M Kherbouche, Y Zghal - Smart Cities, 2023 - mdpi.com

[5] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system.

[6] Ripple, accessed 2 Jan, 2019. URL https://ripple.com/ 20

[7] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, ethereum Project Yellow Paper.

[8] M. Hearn, Corda - a distributed ledger, corda Technical White Paper.

[9] Hyperledger, https://www.hyperledger.org/ (Accessed 2 Jan, 2019).

[10] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, J. Wang, Untangling blockchain: A data processing view of blockchain systems, IEEE Transactions on Knowledge and Data Engineering 30 (7) (2018) 1366– 1385.

[11] M. Vukoli´c, Rethinking permissioned blockchains, in: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, ACM, 2017, pp. 3–7.

[12] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D.

Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, ACM, 2018, p. 30.

[13] Thalheim, B. From Models_For_Programming to Modelling_To_Program and Towards Models_As_A_Program. In Modelling to Program; Springer: Berlin/Heidelberg, Germany, 2021; pp. 3–44. [CrossRef]

[14] Kherbouche, M.; Molnár, B. Formal Model Checking and Transformations of Models Represented in UML with Alloy. In Proceedings of the International Workshop on Modelling to Program, Lappeenranta, Finland, 10–12 March 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 127–136.

[15] Ringert, J.O.; Sullivan, A. Abstract Alloy Instances. In Formal Methods; Springer: Berlin/Heidelberg, Germany, 2023; pp. 364–382. [CrossRef]

[16] AlloyTools. 2023. Available online: https://github.com/AlloyTools (accessed on 15 April 2023).

[17] Rivadeh, M.; Mirian-Hosseinabadi, S.H. Formal translation of YAWL workflow models to the Alloy formal specifications: A testing application. Softw. Syst. Model. 2022. [CrossRef]

[18] Miles, I. Service Innovation. In Handbook of Service Science; Springer: New York, NY, USA, 2010; pp. 511–533. [CrossRef]

[19] Rainer, A. Case Study Research in Software Engineering Guidelines and Examples; Wiley: Hoboken, NJ, USA, 2012.

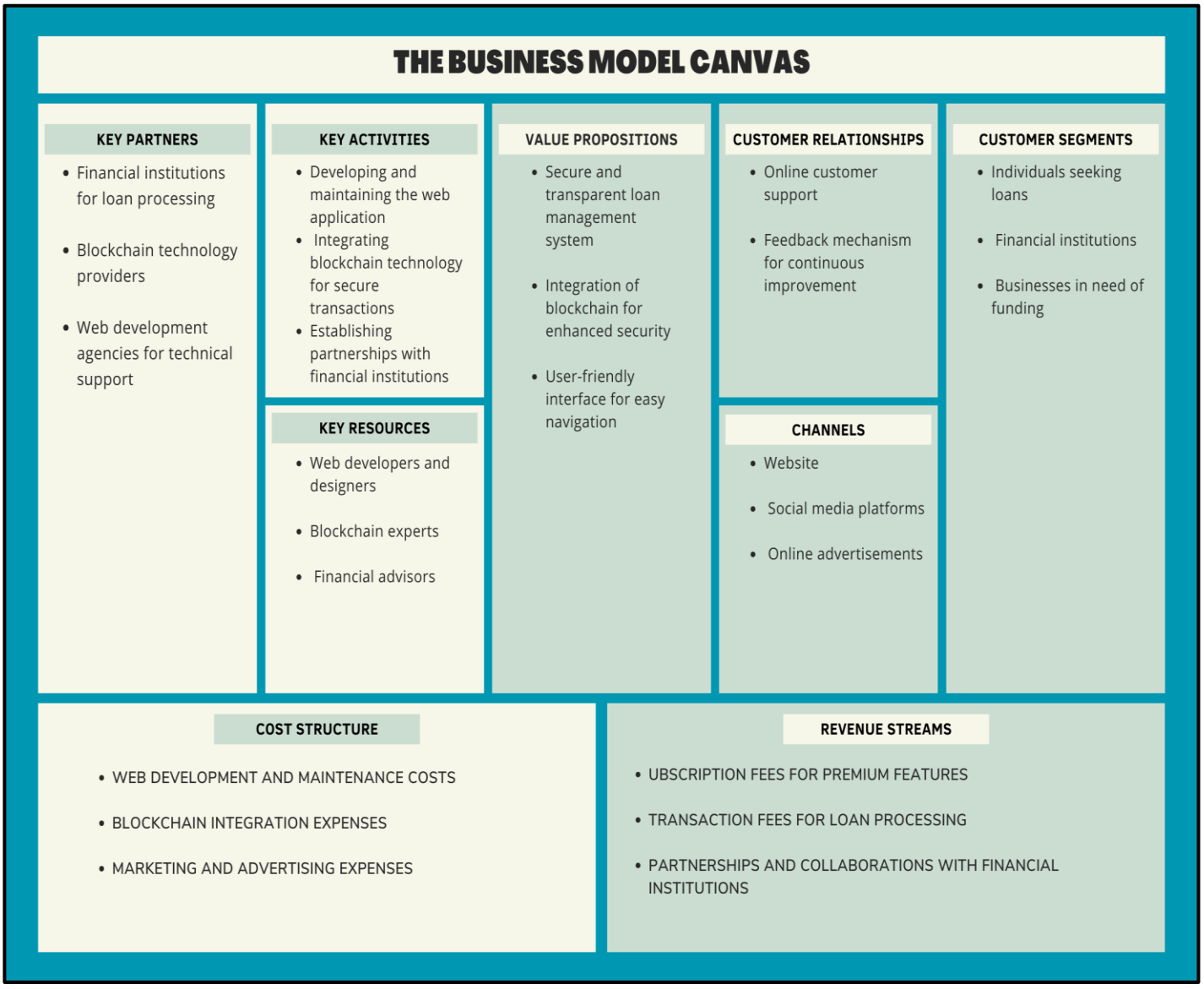
[20] Wieringa, R.J. Design Science Methodology for Information Systems and Software Engineering; Springer: Berlin/Heidelberg, Germany, 2014. [CrossRef]

[21] H. Sukhwani, J. M. Mart´ınez, X. Chang, K. S. Trivedi, A. Rindos, Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric), in: Reliable Distributed Systems (SRDS), 2017 IEEE 36th Symposium on, IEEE, 2017, pp. 253–255.

[22] H Wang, C Guo, S Cheng - Future Generation Computer Systems, 2019 – Elsevier (LoC—A new financial loan management system based on smart contracts)

APPENDIX

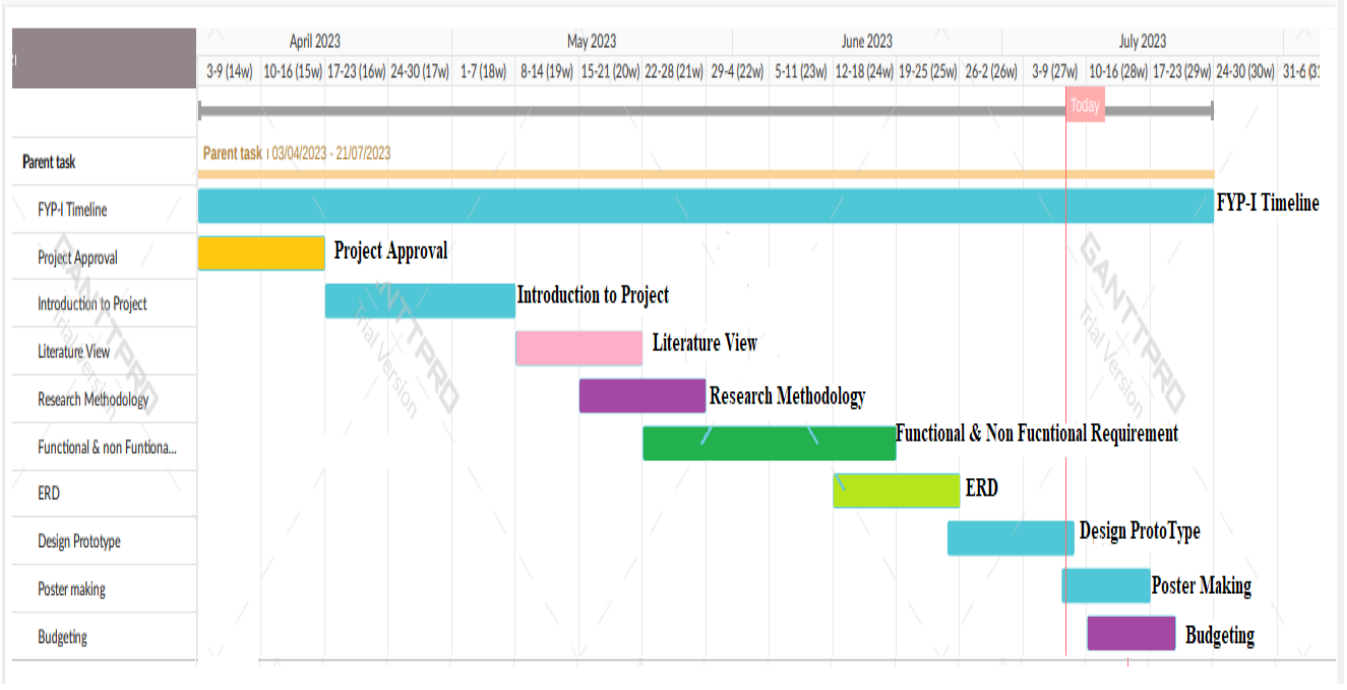
Business Canvas:



Detailed Gantt Chart:

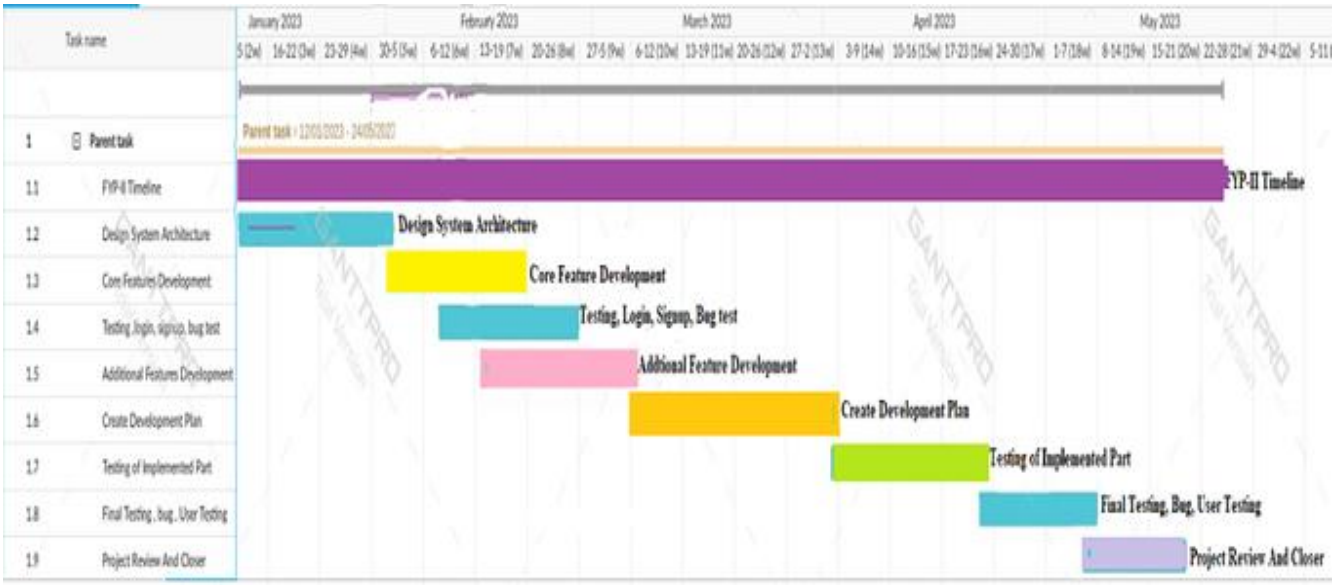
For FYP-I

1.1	FYP-I Timeline	03/04/2023	21/07/2023	16w	In progress
1.2	Project Approval	03/04/2023	14/04/2023	2w	Done
1.3	Introduction to Project	17/04/2023	05/05/2023	3w	Done
1.4	Literature View	08/05/2023	19/05/2023	2w	Done
1.5	Research Methodology	15/05/2023	26/05/2023	2w	Done
1.6	Functional & non Funtional Requirement	22/05/2023	16/06/2023	4w	Done
1.7	ERD	12/06/2023	23/06/2023	2w	Done
1.8	Design Prototype	23/06/2023	06/07/2023	2w	Done
1.9	Poster making	06/07/2023	14/07/2023	1w 2d	In progress
1.10	Budgeting	10/07/2023	18/07/2023	1w 2d	In progress



For FYP-II

1.12	FYP-II Timeline	01/09/2023	30/11/2023	13w	0%	Open
1.13	Iteration 1: design System Architecture, Us...	01/09/2023	21/09/2023	3w	0%	Open
1.14	Iteration 2: Core Features Development & ...	21/09/2023	27/09/2023	1w	0%	Open
1.15	Testing: Login, Signup, home page, bug tes...	27/09/2023	01/10/2023	5d	0%	Open
1.16	Iteration 3: Additional Features Developm...	02/10/2023	22/10/2023	3w	0%	Open
1.17	Iteration 4: Create Deployment Plan ,Depl...	23/10/2023	19/11/2023	4w	0%	Open
1.18	Testing of implemented Parts: Bug Testing, ...	19/11/2023	09/12/2023	3w	0%	Open
1.19	Iteration 5: Final Testing , bug , user testing...	10/12/2023	23/12/2023	2w	0%	Open
1.20	Iteration 6: Project Review and Closer	23/12/2023	05/01/2024	2w	0%	Open



Software Manual:

1) Introduction:

Welcome to the user manual for our Blockchain-Based Loan Management System. This comprehensive guide is designed to assist you in navigating and maximizing the features of our platform. Whether you're a borrower seeking a loan or a lender looking for investment opportunities, this manual will provide you with detailed instructions and tips for utilizing our system effectively.

Within this manual, you will find comprehensive information on the various pages and functionalities available in the web application, including registration processes for borrowers and lenders, loan management features, and engagement with smart contract functionalities. Our aim is to provide you with a secure, transparent, and efficient platform for managing loans and investments.

Initial setup:

2) Features:

- Loan Management: Manage your loans efficiently with our blockchain-based system.
- Smart Contracts: Utilize smart contracts for transparent and secure loan transactions.

3) Getting Started:

- Click on the "Loans" button to explore available loan requests and manage your loans.

4) User Interface:

- Initial landing page presenting two options: Register Form and Login Form.

5) Options:

- Register Form Page: Allows new users to register themselves by providing necessary information.
- Login Form Page: Provides access to registered users, admins, and super admins.

6) User Registration:

- New users choose the Register option.
- They are directed to a registration form where they input required details like username, password, email, etc.

- Upon successful registration, their information is stored securely in the database.

7) User Login:
   - Registered users choose the Login option.
   - They are prompted to enter their username and password.
   - After successful authentication, they gain access to their user-specific functionalities.

8) Admin Login:
   - Admins access a separate login interface.
   - They enter their admin credentials (username and password).
   - Upon successful authentication, they access administrative features like user management, data manipulation, etc.

9) Super Admin Login:
   - Super admins have a dedicated login portal.
   - They input their unique credentials.
   - After successful authentication, they have access to higher-level administrative controls, potentially including system-wide settings, security configurations, etc.

10) Loans:
    I. Explore Loan Requests:
       - View a list of loan requests along with their details and terms.
       - Click on a loan request to view detailed information and borrower profiles.
       - Click on the "See More Details" button to access comprehensive information about the selected loan request.

    II. Sorting and Filtering:
       - Use sorting options to arrange loan requests based on interest rates, loan amounts, or other criteria.
       - Apply filters to narrow down loan requests by borrower credit score, loan purpose, or repayment terms.

    III. Manage Loans:
       - Borrowers can apply for loans and manage their loan requests directly through

the platform.

- Lenders can browse loan requests and invest in selected loans using smart contracts.