

ABSTRACT

In the rapidly evolving landscape of digital banking, NetBanking systems have emerged as essential platforms for delivering secure, efficient, and user-friendly financial services to customers. This synopsis outlines the design and development of a robust NetBanking Website utilizing a Database Management System (DBMS) approach.

The project aims to automate and streamline core banking functions while ensuring high levels of security, accessibility, and data consistency.

The proposed system offers an extensive range of banking functionalities including user authentication, account management, money transfers, deposits, withdrawals, loan applications, balance inquiries, PIN changes, mini statements, and more. The system also provides an exclusive admin dashboard for managing user accounts, approving or rejecting loan applications, and monitoring system activity. Each feature is tightly integrated into a well-structured database to maintain integrity, consistency, and performance.

Users can log in to their accounts, view real-time balances, perform transactions, apply for loans, and download mini statements through a clean and intuitive user interface built using modern web technologies. The system promotes 24/7 availability, reducing dependency on physical bank visits and manual interventions.

From the backend perspective, the system leverages a relational database to ensure accurate data storage, transaction tracking, and secure access controls. It minimizes errors, enhances transparency, and supports scalability for handling a growing number of users and transactions.

By adopting a database-driven architecture, the NetBanking Website ensures that all financial operations are reliable, consistent, and aligned with modern banking standards.

This project serves as a practical and scalable solution to meet the dynamic demands of digital financial management, offering a seamless experience for both users and administrators.

TABLE OF CONTENTS

SLNO	CHAPTERS	PAGE NO
1	INTRODUCTION 1.1 Objective 1.2 Functionality 1.3 Database Management System 1.4 MySQL	1-7
2	LITERATURE REVIEW 2.1 Tools And Environments Used 2.1.1 Frontend a) React JS b) Vite c) Tailwind CSS 2.1.2 Backend a) Spring Boot b) MySQL	8-11
3	PROPOSED WORK 3.1 Entity Relationship (Er) Model 3.2 Schema Of Bank Management System 3.3 Normalization	12-18
4	RESULT 4.1 Screenshots - Admin Portal 4.2 Screenshots - User Portal	19-31
5	CONCLUSION	34
6	BIBLIOGRAPHY	35

CHAPTER 1

INTRODUCTION

This chapter will discuss the various features and aims of the NetBanking application.

1.1 Objective

The objective of this project is to develop a NetBanking Website as a full-stack web application that can efficiently manage various banking operations. The system aims to simplify core banking functionalities such as account access, fund transfers, deposits, withdrawals, mini statements, and loan management. Additionally, the system is designed to provide an intuitive interface for different user roles, enabling easy data input, management, and retrieval.

The application will include two main user models: '**Admin**' and '**User**'. This structure ensures strong access control by assigning role-specific permissions and limiting access to sensitive operations based on the user type.

The user interface aims to provide the following dashboards and views:

- **Admin Dashboard:** Accessible only to 'Admin' users, this dashboard provides control over user account management, transaction monitoring, loan approvals, and report generation. Admins can review user activities, approve or reject loan applications, and view financial summaries across the system.
- **User Dashboard:** Designed for regular banking users, this dashboard allows users to access their account details, view balance, perform transactions like deposits, withdrawals, and fund transfers, apply for loans, and download mini statements. The aim is to reduce dependency on branch visits and offer uninterrupted 24/7 access to banking services.

The system also offers the following modules:

- **Transaction Management:** Displays all recent transactions, including deposits, withdrawals, and transfers, enabling easy tracking of account activity.
- **Loan Management:** Allows users to apply for loans, repay loans and view loans, view loan repayments and enables admins to view, approve, or reject loan requests with relevant details.
- **User Management:** Provides comprehensive control over user accounts, including profile updates, PIN changes, and viewing mini statements.

- **Account Management:** Tracks account types, balances, and linked transactions.

By implementing the NetBanking System, banking institutions can significantly enhance customer experience, improve operational efficiency, and provide secure, accessible financial services. This integrated platform addresses the immediate needs of modern-day banking users while supporting the long-term goal of digital transformation in the financial sector.

1.2 Functionality

Admin

- **Login Only:** Admins are pre-registered and can log in without requiring approval or signup.
- **User Management:** Admins can view and manage all registered user details including personal information, account balances, and recent activities.
- **Transaction Oversight:** Admins can view and monitor all user transactions.
- **Loan Management:** Admins can view, approve, or reject user-submitted loan applications and track loan statuses.
- **Report Generation:** Admins can generate reports or summaries of user transactions and loan approvals for record-keeping and auditing.

User

- **Signup and Login:** Users can register by creating an account and log in using their credentials without requiring approval.
- **View Account Details:** Users can view and edit personal account details including balance, account number, and linked information.
- **Fund Transfers:** Users can initiate account-to-account money transfers within the platform.

- **Deposits & Withdrawals:** Users can perform deposit and withdrawal operations through their dashboard.
- **Loan Application:** Users can apply for loans by submitting required details and tracking their application status , view their loans , repay loans and view loan repayments.
- **Mini Statement Access:** Users can view a mini statement of recent transactions for personal record-keeping.
- **Profile Management:** Users can update editable profile information such as email, address, and pincode.

Loan Management

- **Loan Application Handling:** Users can submit detailed loan requests which include type, amount, and duration.
- **Loan Approval Process:** Admins evaluate submitted loan requests and approve or reject them based on predefined criteria.
- **Loan Status Tracking:** Users can track the status of their loan (pending/approved/rejected/repaid) via their dashboard.

Transaction Management

- **Deposit & Withdrawal Tracking:** System keeps a detailed log of every deposit and withdrawal made by users.
- **Transfer Records:** All money transfer operations are stored with timestamps and transaction references.

1.3 Database Management System:

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of active.

Characteristics of Database Management System:

- Self-describing nature
- Keeps a tight control on data redundancy
- Enforces user defined rules to ensure that integrity of table data
- Provides insulation between programs and data, data abstraction
- Supports multiple views of the data
- Helps sharing of data and multi-user transaction processing

Advantages of DBMS:

- Controlling the redundancy
- Restricting unauthorized access
- Providing persistent storage for program objects
- Providing storage structures for efficient query processing
- Providing multiple users interfaces

- Representing complex relationships among data
- Enforcing integrity constraints

1.4 MySQL

MySQL is a popular, open-source Relational Database Management System (RDBMS) developed by Oracle Corporation. It is widely used for web-based applications and forms a key component of the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python). Known for its performance, reliability, and ease of use, MySQL has become one of the most commonly adopted databases in the world since its inception in 1995.

MySQL supports a wide variety of SQL features, including ACID (Atomicity, Consistency, Isolation, Durability) compliance, foreign key constraints, and multiple storage engines such as InnoDB and MyISAM. It offers a highly extensible architecture and serves as a dependable platform for both small-scale and enterprise-grade applications.

Key Features of MySQL

- **Ease of Use:** MySQL offers a simple setup process and a user-friendly interface that make it easy for developers to install and configure.
- **Performance:** Optimized for high-speed operations, MySQL handles large databases and transaction loads efficiently.
- **Scalability:** Supports both vertical and horizontal scaling, allowing it to adapt to growing data requirements.
- **Replication and Clustering:** Features built-in replication and clustering for high availability, failover, and redundancy.
- **Security:** Provides robust security mechanisms including SSL encryption, access control, and user authentication.
- **Community and Support:** Backed by a vast global community and extensive documentation, ensuring readily available help and resources.

1. **CREATE:**

This command is used to create a table or view by giving it a name and specifying its attributes and constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain values, and any attribute constraints such as NOT NULL.

Syntax: CREATE TABLE <TNAME> (ATR1 TYP1 CONST1, ATR2 TYP2 CONST2, ...)

2. **ALTER:**

The definition of a base table can be altered by the ALTER command which is a schema evolution command. The possible ALTER TABLE operations include adding or dropping a column (attribute), changing a column definition, and adding or dropping table constraints.

Example: ALTER TABLE STUDENT ADD NAME VARCHAR(12)

3. **DROP:**

If a whole schema is not needed anymore, the DROP SCHEMA command can be used. There are two drop behavior options: CASCADE and RESTRICT. CASCADE option is used to remove the database schema and all its tables, domains, and other elements.

RESTRICT option ensures the schema is dropped only if it has no elements in it; otherwise, the DROP command will not be executed.

Syntax: DROP TABLE STUDENT CASCADE

Statements in SQL:

1. SELECT - Used to retrieve the information from the relation.
2. INSERT - Used to insert the new values into the relation.
3. DELETE - Used to delete one or more existing tuples from the relation.
4. UPDATE - Used to update already existing values in the relation.

Aggregate Functions in SQL:

1. COUNT - Returns the number of tuples.
2. SUM - Returns the sum of entries in a column.
3. MAX - Returns the maximum value from an entire column.
4. MIN - Returns the minimum value from an entire column.
5. AVG - Returns the average of all the entries in a column.

Constraints in SQL:

1. NOT NULL - Column should contain some value.
2. PRIMARY KEY - Should not allow duplicate and null values in a column.
3. UNIQUE - Each value of a column should be unique.

CHAPTER 2

LITERATURE REVIEW

2.1 TOOLS AND ENVIRONMENTS USED

The system is developed using modern web technologies, divided into frontend and backend. The frontend uses React.js, Vite, and Tailwind CSS, ensuring a fast and responsive user interface. The backend is built with Spring Boot and MySQL, providing robust and scalable server-side operations.

2.1.1 FRONTEND

a) React.js:

React is an open-source JavaScript library developed and maintained by Facebook. It is widely used for building modern web applications due to its efficient rendering and component-based architecture. React focuses on the view layer of the application (in the MVC model), enabling developers to create rich, interactive, and user-friendly interfaces with minimal effort.

React applications are built using components, which are reusable, self-contained pieces of code that define how a particular section of the UI should appear and behave. These components can maintain their own internal state and can be composed together to create complex user interfaces. This modular approach allows for better organization, scalability, and maintainability of the codebase.

A key feature of React is the virtual DOM—a lightweight, in-memory representation of the real DOM. When a component's state changes, React efficiently updates only the parts of the real DOM that changed by first computing the difference (diffing) between the virtual DOM and the real DOM. This results in faster rendering and a smoother user experience, especially in applications where frequent UI updates are required.

React also supports hooks, such as `useState` and `useEffect`, which enable functional components to manage state and lifecycle methods without writing class components. Additionally, tools like React Router are used for client-side routing, allowing seamless navigation between different parts of the application without reloading the page.

b) Vite:

Vite is a next-generation front-end build tool and development server created by Evan You, the developer of Vue.js. The name "Vite" comes from the French word for "fast", and speed is exactly

what Vite is built for. It significantly improves the development experience with its incredibly fast cold starts and hot module replacement (HMR).

Traditional frontend bundlers like Webpack rebuild the entire application for every code change, which can slow down development as projects grow. In contrast, Vite leverages native ES modules in the browser and pre-bundles dependencies using esbuild, a super-fast bundler written in Go. This results in near-instant project startup times and quick updates during development.

Vite also integrates smoothly with React and Tailwind CSS, making it an ideal choice for modern web projects. By using Vite in this project, the development process becomes more efficient, builds are faster, and the overall workflow is more developer-friendly.

c) Tailwind CSS:

Tailwind CSS is a utility-first CSS framework that allows developers to style their applications directly in the HTML or JSX using predefined classes. Unlike traditional CSS frameworks like Bootstrap, which provide pre-designed components, Tailwind gives developers complete control over the design by offering atomic classes for padding, margins, colors, fonts, flexbox, grid, shadows, and more.

For example, using a class like `bg-blue-500 text-white p-4 rounded-lg shadow-md` quickly creates a button with a blue background, white text, padding, rounded corners, and a shadow effect—all without writing custom CSS.

Tailwind CSS is also highly extensible through its `tailwind.config.js` file, which allows you to define custom themes, colors, breakpoints, and more. Combined with Vite and React, Tailwind offers a powerful and efficient way to build modern, visually appealing, and maintainable user interfaces.

2.1.2 BACKEND

a) Spring Boot:

Spring Boot is a powerful, open-source framework developed by Pivotal Software (a part of VMware), built on top of the Spring Framework. It is designed to simplify the process of developing, configuring, and deploying Java-based enterprise applications by providing production-ready defaults and reducing boilerplate code.

In this project, Spring Boot is used as the core backend framework to implement key business logic and handle server-side operations. It serves the following essential roles:

- **RESTful APIs:** Spring Boot exposes various endpoints using REST APIs to handle operations like user registration, login, data retrieval, transactions, and loan applications.
- **Annotation-Based Configuration:** With annotations like `@RestController`, `@Service`, `@Repository`, and `@Autowired`, Spring Boot enables rapid and structured development by clearly separating concerns and reducing the need for XML configurations.
- **Spring Data JPA Integration:** Enables easy interaction with the MySQL database using Java classes and interfaces, simplifying CRUD operations through repository interfaces like `UserRepository`, `TransactionRepository`, etc.
- **Embedded Tomcat Server:** Allows the application to run independently without requiring an external server, making deployment and testing seamless.

Overall, Spring Boot enhances productivity by abstracting complex configurations and allows developers to focus on writing business logic. Its scalability, modularity, and integration capabilities make it ideal for modern web applications with evolving requirements.

b) MySQL:

MySQL is a widely-used open-source relational database management system (RDBMS) developed by Oracle Corporation. It is known for its speed, reliability, and robustness in handling structured data. MySQL uses Structured Query Language (SQL) to manage, query, and manipulate data stored in relational tables.

In this project, MySQL serves as the primary data storage engine for all backend operations. It performs the following key functions:

- **Data Storage:** Stores all essential application data, including user credentials, transaction history, account details, loan applications, and mini-statements.
- **Relational Integrity:** Supports relationships through foreign keys, ensuring consistency across related tables like users, transactions, accounts, loans, and admins.
- **Query Optimization:** MySQL's indexing and optimized query execution make data retrieval efficient, which is vital for real-time features like viewing transactions or mini-statements.
- **ACID Compliance:** MySQL ensures **Atomicity**, **Consistency**, **Isolation**, and **Durability**, which are crucial for financial and banking applications where data accuracy and integrity

are

non-negotiable.

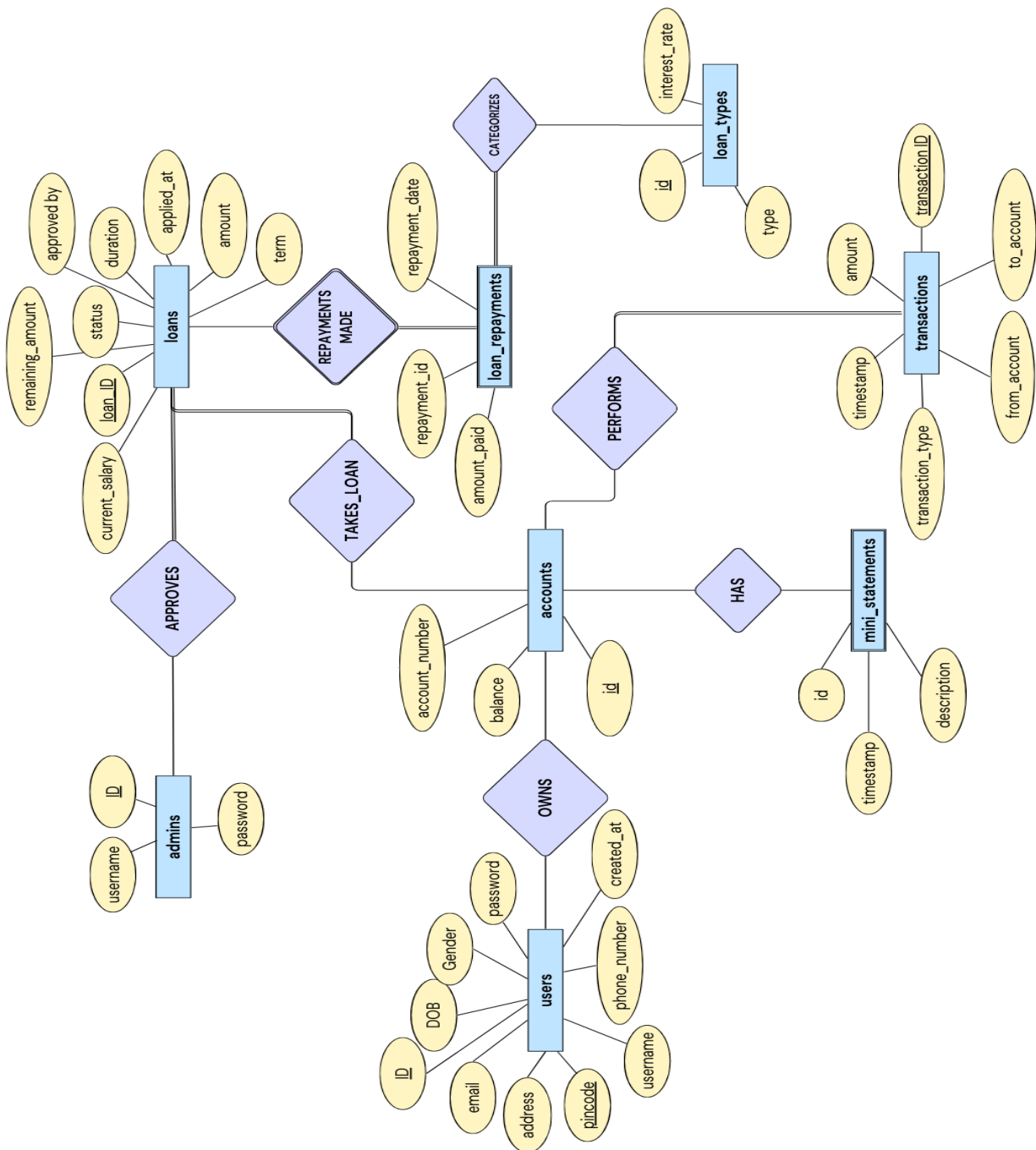
- **Compatibility:** Easily integrates with Spring Boot through JPA (Java Persistence API), enabling seamless data access using object-oriented models.

By using MySQL in this project, the application benefits from a stable and high-performance data management system that ensures quick access and safe storage of critical user and transactional data.

CHAPTER 3 PROPOSED WORK

3.1 ENTITY RELATIONSHIP (ER) MODEL

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities.



3.2 SCHEMA OF BANK MANAGEMENT SYSTEM

In this project, we will be using eight entities to manage the operations of a banking system. These entities include Users, Admins, Accounts, Transactions, Mini_Statements, Loans, Loan_Types, and Loan_Repayments. Each of these entities plays a vital role in ensuring smooth functioning of core banking operations such as managing user details, handling account information, recording transactions, generating mini statements, issuing and repaying loans, and storing loan type metadata. The relationships between these entities allow for a comprehensive and efficient system to support all standard banking features.

users: Stores user details.

- Attributes:
id, created_at, email, password, phone_number, username, address, date_of_birth, gender, pincode.
 - id is the **primary key**.
-

admins: Stores admin login credentials.

- Attributes:
id, username, password.
 - id is the **primary key**.
-

accounts: Stores account details for users.

- Attributes:
id, account_number, balance, user_id.
 - id is the **primary key**.
 - user_id is a **foreign key** referencing users(id).
-

transactions: Stores transaction information between accounts.

- Attributes:
transaction_id, amount, timestamp, transaction_type, from_account, to_account.
 - transaction_id is the **primary key**.
 - from_account and to_account are **foreign keys** referencing accounts(id).
-

mini_statements: Stores summary of recent transactions per account.

- Attributes:
id, description, timestamp, transaction_id, account_id.
 - id is the **primary key**.
 - transaction_id is a **foreign key** referencing transactions(transaction_id).
 - account_id is a **foreign key** referencing accounts(id).
-

loans: Stores loan information for users.

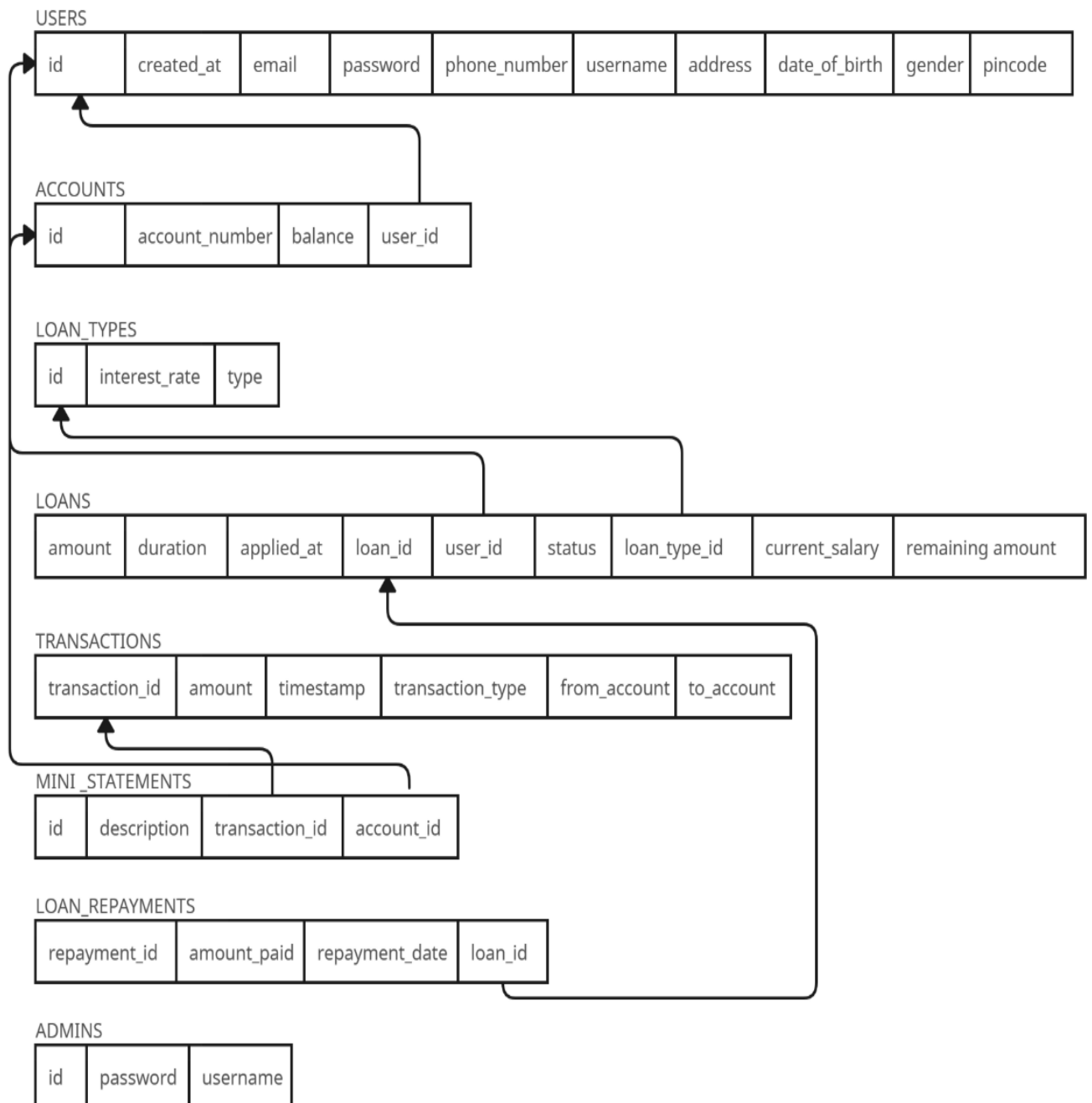
- Attributes:
loan_id, amount, duration, applied_at, approved_by, user_id, status, loan_type_id, current_salary, remaining_amount.
 - loan_id is the **primary key**.
 - user_id is a **foreign key** referencing users(id).
 - approved_by is a **foreign key** referencing admins(id).
 - loan_type_id is a **foreign key** referencing loan_types(id).
-

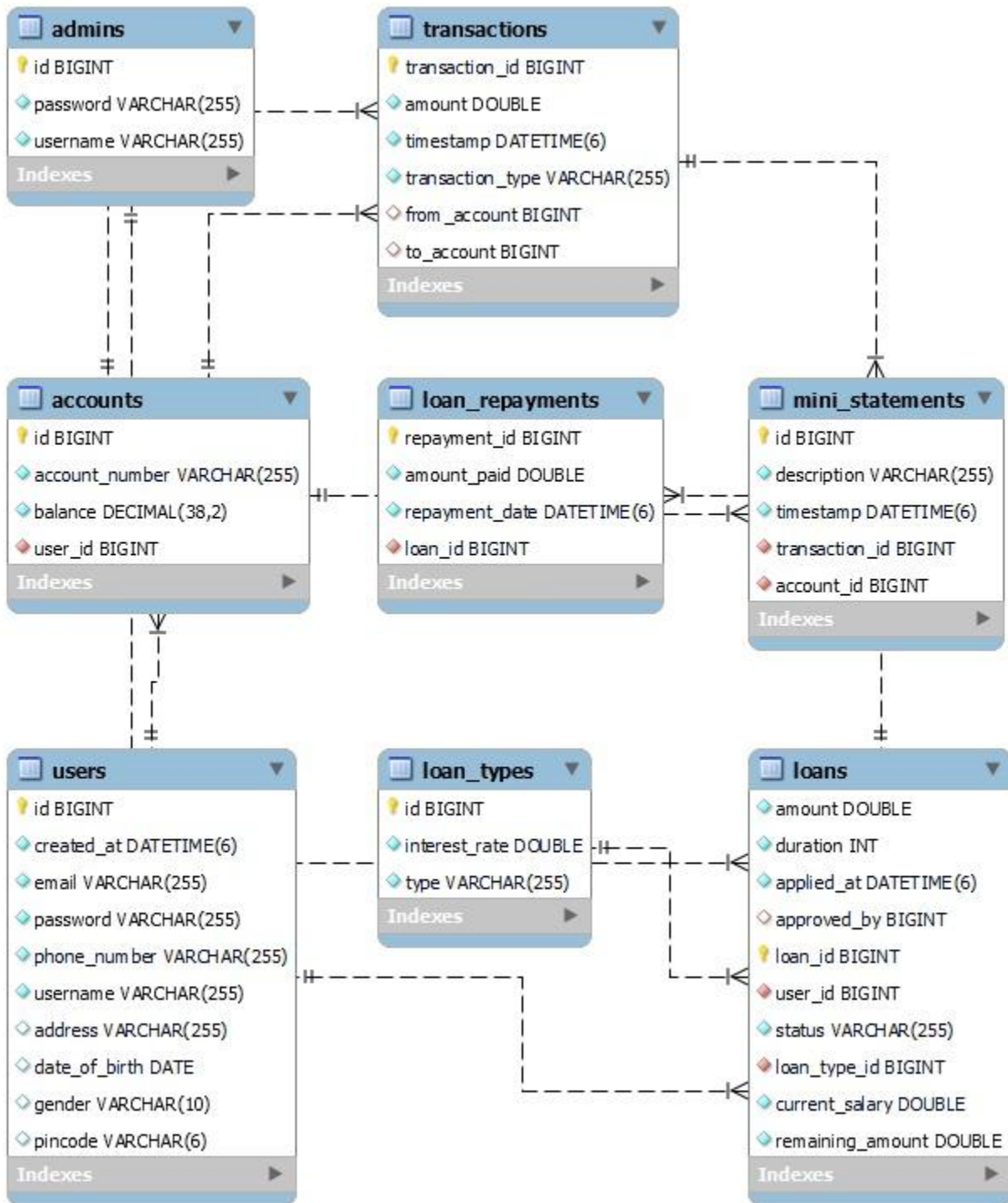
loan_types: Defines different types of loans.

- Attributes:
id, interest_rate, type.
 - id is the **primary key**.
-

loan_repayments: Stores repayment details for loans.

- Attributes:
repayment_id, amount_paid, repayment_date, loan_id.
- repayment_id is the **primary key**.
- loan_id is a **foreign key** referencing loans(loan_id).





3.3 NORMALIZATION

The following is a normalization report for the Bank Management System project:

1st Normal Form (1NF):

All entities in the system, including Users, Admins, Accounts, Transactions, Mini_Statements, Loans, Loan_Types, and Loan_Repayments, are in 1NF. Each record has a unique primary key, and all attributes contain atomic values. No fields contain sets, arrays, or multivalued attributes.

2nd Normal Form (2NF):

Each table is in 2NF because:

- There are no partial dependencies because no table has a composite key.
- All attributes in every table are fully dependent on the whole primary key.

3rd Normal Form (3NF):

All tables are in 3NF because:

- There are no transitive dependencies.
- Each non-primary key attribute is only dependent on the primary key, and not on other non-key attributes.

Users Table

Stores attributes like username, email, password, phone_number, address, pincode, etc. All fields are fully functionally dependent on the primary key id.

Accounts Table

Contains account_number, balance, user_id (foreign key).

Each attribute is directly dependent on id.

user_id is a foreign key to the Users table, maintaining referential integrity.

Loans Table

Fields include amount, duration, status, loan_type_id, user_id, etc.

Attributes such as current_salary and remaining_amount are directly related to the loan application and depend on id.

loan_type_id, user_id, and approved_by are foreign keys ensuring normalization through relationships.

Transactions Table

Each attribute is functionally dependent on transaction_id.

There are no transitive dependencies, and all relationships (like from_account and to_account) are managed using foreign keys.

Mini_Statements Table

Stores fields such as description, timestamp, transaction_id, account_id.

transaction_id and account_id are foreign keys, linking to respective tables.

All non-primary key attributes are directly dependent on id.

Loan_Repayments Table

Includes fields like amount_paid, repayment_date, loan_id.

loan_id is a foreign key to the Loans table.

All fields are directly related to a specific repayment (repayment_id), ensuring 3NF compliance.

Admins Table

Contains username, password.

All fields depend solely on the primary key, with no indirect or transitive dependencies.

By ensuring all tables follow the principles of 1NF, 2NF, and 3NF, the Bank Management System design:

- Maintains data integrity.
- Prevents data anomalies.
- Optimizes performance for queries, updates, and storage.

CHAPTER 4

RESULT

The Online Banking System marks a significant advancement in digital banking solutions, transforming how customers manage their finances and interact with their banks. This comprehensive system provides a centralized and secure platform that streamlines essential banking operations such as account management, money transfers, transaction tracking, and loan processing.

Designed with an intuitive user interface and equipped with robust functionalities, the platform empowers both users and administrators to perform tasks efficiently and accurately. Customers can seamlessly access their account information, initiate transactions, and apply for loans, while administrators are equipped with tools to monitor user activity, approve loans, and oversee operations effectively.

The system promotes transparency and real-time updates through features such as transaction tracking, mini statements, and instant balance inquiries. Moreover, the loan module allows users to apply for various types of loans with detailed processing and status tracking, repay loans and view loans enhancing user engagement and trust.

By integrating modern web technologies like React.js and Spring Boot, and adhering to best practices in software architecture, the Online Banking System ensures high performance, data integrity, and secure communication. This solution not only enhances operational efficiency but also elevates user satisfaction by offering a reliable, user-friendly, and feature-rich digital banking experience.

Through this project, we demonstrate how a technology-driven approach can redefine traditional banking systems, contributing to smarter financial management, better service delivery, and greater accessibility for users across all levels.

4.1 SCREENSHOTS - ADMIN PORTAL

4.1.1 Admin Login

4.1.2 Admin Dashboard

4.1.3 View All Users

4.1.4 Report on all Low Balance Accounts

4.1.5 Report on Active Users

4.1.6 Report on High Amount Transactions

4.1.7 View/Approve/Reject Pending Loans

4.1.8 Transaction Summary / Bank Report

ADMIN LOGIN PAGE

Admin Login

Username

Enter admin username

Password

Enter admin password

Login

For authorized personnel only.

254 Monon road #1456 coron
Tiam
Office No 3457
info@scalens.com
1-800-356-6522

About us

Coners
Blog
Team
Contact Us

Our Services

Internet Banking
Mobile Banking
Atella Bank Security
Rates And Charges

© 2025 MyBank. All rights reserved.

Fig 4.1.1 Admin Login

Vaultik

1-800-BANK-HELP

Sign Out

Welcome, admin123

Admin Dashboard

ID: 1

Last Login Today

Status Active

Admin Controls

View All Users

Manage registered users, view details, and take necessary actions.

View Users

Pending Loan Applications

Review loan applications and approve/reject them accordingly.

View Loans

Bank Reports

View overall banking reports, analyze performance and view summary.


View Reports


Need Assistance?

If you encounter any issues, please reach out to our support team.

Fig 4.1.2 Admin Dashboard

20


1-800-BANK-HELP
Sign Out



User Management Dashboard

View and manage all registered users and their activities


Filter Users

All Users

All Registered Users

USER ID	USERNAME	EMAIL	PHONE	ACCOUNT NO	BALANCE
1	alice	alice@example.com	987654321	8369579856	₹838.00
3	john	john@example.com	987123457	1658690663	₹1700.00
4	jane	jane@example.com	987123111	6724312298	₹4000.00
5	ne	ane@example.com	977123111	1264674596	₹1000.00
7	john_doe	jd@example.com	9876543210	7297758335	₹0.00
9	ben	ben@example.com	9876511110	4061684719	₹0.00
11	ken	ken@example.com	7776221110	6004873708	₹0.00
12	zahiyasaheer	zahiyasaheer19@gmail.com	9632781119	6256047745	₹0.00
14	milliethelillie	mills@gmail.com	9632781113	7535386887	₹44201.00

4.1.3 View All Users



User Management Dashboard


View and manage all registered users and their activities

Filter Users

Low Balance Accounts

Balance Threshold

₹ 7000



Low Balance Accounts

Search

Username	Email	Phone	Account No	Balance
alice	alice@example.com	987654321	8369579856	₹838.00
jane	jane@example.com	987123111	6724312298	₹4000.00
ne	ane@example.com	977123111	1264674596	₹1000.00
john_doe	jd@example.com	9876543210	7297758335	₹0.00
j_doe	jd@example.com	9876500210	1454327835	₹0.00
ben	ben@example.com	9876511110	4061684719	₹0.00
ken	ken@example.com	7776221110	6004873708	₹0.00
zahiyasaheer	zahiyasaheer19@gmail.com	9632781119	6256047745	₹0.00

4.1.4 Report on all Low Balance Accounts

User Management Dashboard

View and manage all registered users and their activities

Filter Users

Most Active Users

Most Active Users

Search

User ID	Username	Email	Phone	Account No	Total Transactions
1	alice	alice@example.com	987654321	8369579856	33
3	john	john@example.com	987123457	1658690663	6
10	den	den@example.com	7776511110	8063103047	6
4	jane	jane@example.com	987123111	6724312298	4
14	milliethelillie	mills@gmail.com	9632781113	7535386887	3
5	ne	ane@example.com	977123111	1264674596	2

4.1.5 Report on Active Users

1-800-BANK-HELP

Sign Out

User Management Dashboard

View and manage all registered users and their activities

Filter Users

High Amount Transactions

Minimum Amount

₹ 500

High Amount Transactions

List of transactions above ₹500.

Search

Transaction ID	Sender	Sender Account	Receiver	Receiver Account	Amount (₹)	Date & Time
38	alice	8369579856	den	8063103047	₹8000	3/21/2025, 1:32:31 PM
35	alice	8369579856	den	8063103047	₹1000	3/21/2025, 1:06:03 PM
45	milliethelillie	7535386887	alice	8369579856	₹699	4/6/2025, 10:33:54 AM

4.1.6 Report on High Amount Transactions

Pending Loan Applications

Review and manage all pending loan requests

Car Loan Application

Loan ID: 4

Pending

User Details

Name
jane

Phone
987123111

Balance
₹4000.00

Email
jane@example.com

Account No
6724312298

Loan Information

Amount
₹50000.00

Interest Rate
6%

Duration
24 months

Applied Date
3/7/2025

✓ Approve

✗ Reject

User's Loan History

Loan #4 (Car Loan)

Pending

Loan Statistics

Total Loans Taken: **1**

Personal Loan Application

Loan ID: 7

Pending

User Details

Name
milliethelillie

Phone
9632781113

Balance
₹4201.00

Email
mills@gmail.com

Account No
7535386887

Loan Information

Amount
₹70000.00

Interest Rate
7.2%

Duration
12 months

Applied Date
4/6/2025

✓ Approve

✗ Reject

User's Loan History

Loan #7 (Personal Loan)

Pending

Loan #8 (Home Loan)

Pending

Loan Statistics

Total Loans Taken: **2**

4.1.7 View/Approve/Reject Pending Loans

Transaction Summary Search				
User ID	Username	Account No	Total Deposits	Total Withdrawals
1	alice	8369579856	₹11000.00	₹40.00
9	ben	4061684719	₹0.00	₹0.00
10	den	8063103047	₹1000.00	₹10.00
8	j_doe	1454327835	₹0.00	₹0.00
4	jane	6724312298	₹4000.00	₹0.00
3	john	1658690663	₹1000.00	₹0.00
7	john_doe	7297758335	₹0.00	₹0.00
5	ne	1264674596	₹0.00	₹0.00

Need Assistance?
If you encounter any issues, please reach out to our support team.

About Us

MyBank is dedicated to providing secure and convenient banking solutions. With a commitment to customer satisfaction, we offer innovative services tailored to your financial needs.

Contact Us

264 Manon road F1 456, Caron Town
Office No 3457
info@salem.com
1-800-356-6522

Our Services

Withdrawal
Deposits
Transfers
Loans
Ministatment

© 2025 MyBank. All rights reserved.

4.1.8 Transaction Summary / Bank Report

4.2 SCREENSHOTS - USER PORTAL

- 4.2.1 User Login
- 4.2.2 User Registration
- 4.2.3 User Dashboard
- 4.2.4 Withdraw
- 4.2.5 Deposit
- 4.2.6 Transfer Funds
- 4.2.7 Mini Statement
- 4.2.8 View all Transactions
- 4.2.9 View Your Loan Details
- 4.2.10 Apply For Loan
- 4.2.11 Loan Repayment
- 4.2.12 Loan Payment History
- 4.2.13 View your details

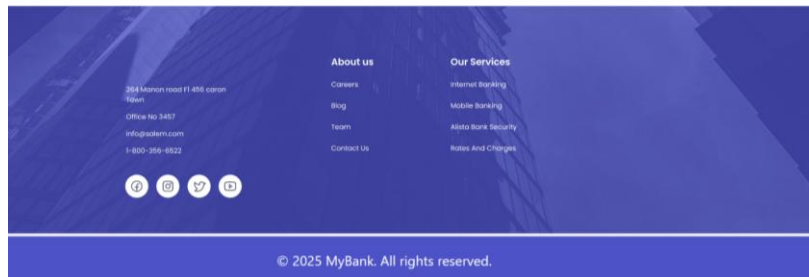
Login to Your Account

Username

Password

Login

Don't have an account? [Sign Up](#)



4.2.1 User Login

Create an Account

Username

Password

Email

Phone Number

dd-mm-yyyy

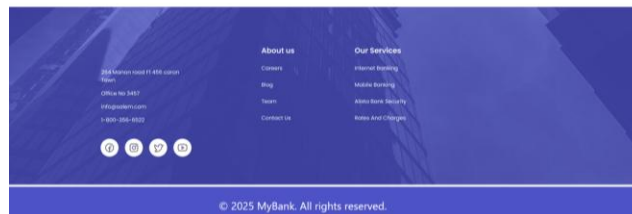
Select Gender

Address


Pincode

Register

Already have an account? [Login](#)



4.2.2 User Registration


1-800-BANK-HELP
milliethelillie
Sign Out

Welcome back, milliethelillie!

mills@gmail.com 9632781113

Available Balance
₹44201.00

Account Type
Savings


Last Login
Today

Status
Active

Account Number7535386887

Member Since: April 6, 2025


Banking Services



Withdraw Money

Easily withdraw cash from your account at any time.


Withdraw



Deposit Money

Add funds to your account securely and instantly.


Deposit



Transfer Money

Send money to anyone instantly with secure transactions.


Transfer



Apply for Loan

Get loans at the best interest rates with easy approvals.


Apply Now



Mini Statement

View your recent transactions and account summary.

View Statement



View your Loans

View all your loans, loan payments, loan approvals, status

View

Need Assistance?

If you encounter any issues, please reach out to our support team.

About Us

Vaultik is dedicated to providing secure and convenient banking solutions. With a commitment to customer satisfaction, we offer innovative services tailored to your financial needs.

Contact Us

123, 1st Main Road, Bengaluru
Office No 3457
info@?bank.com
1-800-356-6522

Our Services


- Withdrawal
- Deposits
- Transfers
- Loans
- Ministatment

© 2025 Vaultik. All rights reserved.

4.2.3 User Dashboard

3
Withdrawal Procedure

Note: Please ensure your account has sufficient balance before making a withdrawal. Standard processing time is 1-2 business days.



Account Number
8369579856

Amount to Withdraw
₹ 0.00

Notes (Optional)
Purpose of withdrawal or special instructions

Withdraw Now

Account Balance
Check Current Balance


4.2.4 Withdraw

Deposit

Deposit funds to your account in just a few steps

3
How to Deposit Money?

Depositing money into your bank account is simple and secure. You can deposit funds through various methods such as cash deposit at a branch, online transfer, or using an ATM.



- Online Transfer:** Use net banking to add funds instantly.
- Cash Deposit:** Visit the nearest branch to deposit cash.
- ATM Deposit:** Use a cash deposit machine at select ATMs.
- Cheque Deposit:** Submit a cheque at any bank branches.

4
Enter Deposit Details

Account Number
8369579856

Deposit Amount (₹)
₹ Enter amount

Deposit Now

Want to check your balance?
Check Balance

4.2.5 Deposit


Transfer Funds

Securely transfer funds between accounts in just a few steps.

⚠ Important Notice:
Transfers may take up to **24 hours** to reflect in the recipient's account. Ensure all details are correct before proceeding.

Transfer Procedure

Ensure you enter the correct recipient details before proceeding. Transfers may take up to 24 hours to process.



Sender's Account Number

Enter your account number

Recipient's Account Number

Enter recipient's account number

Amount

Enter amount to transfer

Transaction Description

Optional: Enter reason for transfer

Transfer Now

Want to check your balance before transferring?

Check Balance

4.2.6 Transfer Funds

Mini Statement

Your recent transactions at a glance.

VIEW ALL TRANSACTIONS

VIEW

Recent Transactions

Timestamp	Transaction ID	Description
2025-03-12 14:30	TXN123456	Transfer sent of ₹500.0 to A/c 1658690663
2025-03-10 10:15	TXN654321	Received ₹2000.0 from A/c 4587963258
2025-03-08 18:45	TXN789123	Electricity Bill Payment of ₹1200.0
2025-03-05 09:30	TXN456789	ATM Withdrawal of ₹1000.0
2025-03-02 16:20	TXN987654	Online Shopping - ₹2500.0 debited

Need Help?

If you have questions about your transactions, please contact our [support team](#).

4.2.7 Mini Statement

ALL TRANSACTIONS

Transaction ID	Type	Amount	From Account	To Account	Timestamp
33	LoanRepayment	₹2000.00	1528161099	1528161099	4/5/2025, 10:39:32 PM
32	LoanRepayment	₹2000.00	1528161099	1528161099	4/5/2025, 10:30:03 PM
31	LoanRepayment	₹2000.00	1528161099	1528161099	4/5/2025, 10:28:30 PM
30	Withdraw	₹99.00	1528161099	-	4/4/2025, 11:32:50 PM
29	Deposit	₹100.00	-	1528161099	4/4/2025, 11:32:06 PM
28	Deposit	₹100.00	-	1528161099	4/4/2025, 11:31:31 PM
27	Deposit	₹1000.00	-	1528161099	4/4/2025, 11:30:33 PM
26	Deposit	₹1000.00	-	1528161099	4/4/2025, 11:28:39 PM
25	Deposit	₹100.00	-	1528161099	4/4/2025, 11:27:44 PM
24	Deposit	₹1000.00	-	1528161099	4/4/2025, 11:27:13 PM
23	Transfer	₹149.00	1528161099	6211098374	3/21/2025, 3:34:11 PM
22	Transfer	₹1.00	8144066703	1528161099	3/20/2025, 1:30:09 PM
21	Transfer	₹999.00	8144066703	1528161099	3/20/2025, 1:29:15 PM
20	Transfer	₹1000.00	1528161099	8144066703	3/20/2025, 1:28:43 PM
18	Deposit	₹100.00	-	1528161099	3/20/2025, 1:02:39 PM
17	Withdraw	₹100.00	1528161099	-	3/20/2025, 12:52:20 PM
16	Withdraw	₹90.00	1528161099	-	3/20/2025, 12:51:05 PM
11	Transfer	₹100.00	1528161099	8698117315	3/12/2025, 10:12:30 AM
10	Withdraw	₹10.00	1528161099	-	3/12/2025, 10:12:17 AM
9	Deposit	₹1000.00	-	1528161099	3/12/2025, 10:12:04 AM

Need Assistance?

If you encounter any issues, please reach out to our support team.

About Us

Vaultik is dedicated to providing secure and convenient banking solutions. With a commitment to customer satisfaction, we offer innovative services tailored to your financial needs.

Contact Us

123, 1st Main Road, Bengaluru
Office No 3457
info@bank.com
1-800-356-6522

Our Services

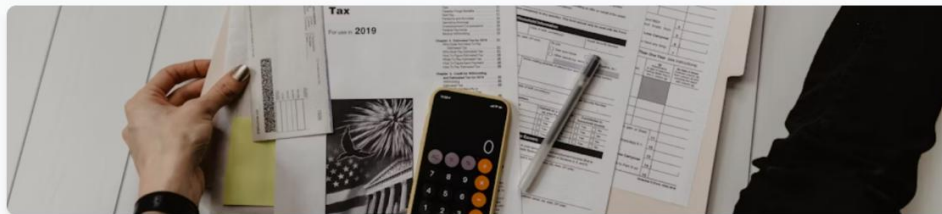
Withdrawal
Deposits
Transfers
Loans
Ministatment

© 2025 Vaultik. All rights reserved.

4.2.8 View All Transactions

Your Loan Details

View and manage all your active loans in one place



Loan ID	Applied Date	Loan Type	Amount	Duration	Status	Remaining	Actions	
L5	Mar 12, 2025	Car-Loan	₹50,000.00	12 months	Approved	₹49,000.00	View	Repay
L6	Mar 18, 2025	Car-Loan	₹50,000.00	12 months	Approved	₹20,000.00	View	Repay
L7	Mar 18, 2025	Home-Loan	₹50,000.00	12 months	Repaid	₹0.00	View	Repay
L9	Mar 19, 2025	Car-Loan	₹50,000.00	12 months	Approved	₹10,000.00	View	Repay
L13	Mar 20, 2025	Home-Loan	₹10,000.00	0 months	Approved	₹0.00	View	Repay
L14	Mar 20, 2025	Education-Loan	₹99,998.00	0 months	Rejected	₹0.00	View	Repay
L15	Mar 21, 2025	Personal-Loan	₹1,500.00	0 months	Approved	₹0.00	View	Repay
L16	Mar 21, 2025	Personal-Loan	₹1,500.00	0 months	Repaid	₹0.00	View	Repay
L17	Mar 21, 2025	Education-Loan	₹2,500.00	0 months	Rejected	₹0.00	View	Repay
L18	Mar 21, 2025	Home-Loan	₹50,000.00	12 months	Approved	₹50,000.00	View	Repay
L19	Mar 21, 2025	Car-Loan	₹12,223.00	15 months	Approved	₹10,723.00	View	Repay
L20	Apr 5, 2025	Personal-Loan	₹2,500.00	15 months	Repaid	₹0.00	View	Repay
L21	Apr 5, 2025	Home-Loan	₹20,000.00	20 months	Rejected	₹0.00	View	Repay

Need another loan?

Apply for a new loan with competitive interest rates

[Apply for New Loan](#)

4.2.9 View Your Loan Details

Apply for a Loan

Secure your future with our easy loan application process. Fill out the form below and submit your request.



Full Name alice	Email alice@example.com
Phone Number 987654321	Loan Amount Enter loan amount (in INR)
Duration Enter loan repayment duration (in months)	Loan Type Select a loan type
Employment Status Employed	Annual Income (INR) Enter your annual income
Address Enter your residential address	
Pincode Enter your area pincode	Date of Birth dd-mm-yyyy
Gender Male	

☐ I agree to the [terms and conditions](#) of loan approval.

Apply Now

★ Loan approvals are subject to eligibility criteria, credit checks, and bank policies.

Need Assistance?

If you encounter any issues, please reach out to our support team.

About Us

?Bank is dedicated to providing secure and convenient banking solutions. With a commitment to customer satisfaction, we offer innovative services tailored to your financial needs.

Contact Us


123, 1st Main Road, Bengaluru
Office No 3457
info@?bank.com
1-800-356-6522


Our Services

Withdrawal
Deposits
Transfers
Loans
Ministatment

© 2025 ?Bank. All rights reserved.

4.2.10 Apply for Loan



[1-800-BANK-HELP](#)  [milliethelillie](#) [Sign Out](#)

Loan Repayment

Make a payment towards your outstanding loan

Current Outstanding Balance

₹65,000.00

Enter Payment Amount (₹)


Payment of ₹5,000 processed successfully!


New Balance: ₹65,000.00

Make Payment

Back

4.2.11 Loan Repayment



[1-800-BANK-HELP](#)  [milliethelillie](#) [Sign Out](#)

Repayment History

Track all your loan repayments in one place


Total Repaid
₹30,000.00

Total Payments
4


Last Payment
Apr 6, 2025

Payment ID	Amount	Date	Status
#1	₹5,000.00	Apr 6, 2025	Completed
#2	₹10,000.00	Apr 6, 2025	Completed
#3	₹7,000.00	Apr 6, 2025	Completed
#4	₹8,000.00	Apr 6, 2025	Completed

4.2.12 Loan Payment History



1-800-BANK-HELP

 milliethellie

Sign Out

Profile Details

Manage your personal and account information

Personal Information

Email
mills@gmail.com

Password

Phone Number
9632781113

Date of Birth
2005-01-06

Gender
Female

Address & Account

Address
#13, hrbr layout

Pincode
560043

Account Number
7535386887

Account Balance
₹44,201

Loan Application

Apply for a loan with competitive interest rates and flexible repayment options.

2 Loans Applied

Apply for New Loan

Note: Loan approvals are subject to bank policies, income verification, and credit history. Additional documents may be required during the verification process.

Need Assistance?

If you encounter any issues, please reach out to our support team.

About Us

Vaultik is dedicated to providing secure and convenient banking solutions. With a commitment to customer satisfaction, we offer innovative services tailored to your financial needs.

Contact Us

123, 1st Main Road, Bengaluru
Office No 3457
info@7bank.com
1-800-356-6522

Our Services

Withdrawal
Deposits
Transfers
Loans
Ministatment

© 2025 Vaultik. All rights reserved.

4.2.13 View you details

CONCLUSION

The development of our NetBanking website marks a pivotal advancement in the modernization of digital banking services. By leveraging Spring Boot for the backend and React.js for the frontend, we have created a dynamic, responsive, and secure platform that addresses the evolving expectations of both banks and their customers. The system ensures seamless integration between frontend and backend operations, allowing real-time access to essential banking functionalities.

Our platform is powered by a robust MySQL database that efficiently manages user data, account information, and transactional records while upholding strong data integrity and security standards. Features like secure user authentication, fund transfers, transaction history, and balance inquiries are implemented with a strong focus on user privacy and system reliability.

The architecture supports scalability and maintainability, enabling future enhancements such as support for UPI, loan management, and investment tracking. The use of modern frameworks ensures a smooth user experience across devices, while also allowing for the integration of analytics and monitoring tools for administrators.

This NetBanking project showcases the practical application of full-stack web development in solving real-world financial service challenges. It empowers users with easy, secure, and efficient control over their banking needs from any location, promoting a cashless and paperless economy.

In conclusion, our NetBanking system is a reliable, user-centric solution that demonstrates the power of Spring Boot, React.js, and MySQL in building scalable and secure web applications. It lays the foundation for continued innovation in the fintech space and supports the digital transformation of traditional banking services.