
CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

This chapter gives an overview about the project description, tools and technology used, hardware and software requirement of the system.

1.2 PROJECT DESCRIPTION

The Bankify-Online Banking System is a project aimed at developing a user-friendly, secure, and convenient online banking application called Bankify. The project addresses the issues faced by users in existing online banking applications, such as complex interfaces, difficulties in navigation, and concerns about security and data privacy. Bankify aims to provide a simple and intuitive user interface, essential banking services, and robust security measures to ensure a seamless and reliable banking experience.

1.3 TOOLS AND TECHNOLOGY USED

- **Frontend Development:** The user interface of the Bankify application was created using HTML, CSS, and JavaScript. These technologies were utilized to design and develop an intuitive and visually appealing frontend that offers a seamless user experience.
- **Backend Development:** The backend functionality of the Bankify application was implemented using PHP as the server-side scripting language. PHP enables the handling of server-side processing, data manipulation, and integration with the frontend components.
- **Database Management System:** The Bankify application employed MySQL, an open-source relational database management system (RDBMS), to store and manage the banking-related data. MySQL ensures efficient data storage, retrieval, and secure management of user account details and transaction information.
- **Version Control:** Git, a widely adopted version control system, was utilized to track changes, manage the codebase, and facilitate collaboration among developers working on the Bankify project. It enables effective code management, branching, and merging to ensure a smooth development process.
- **Hosting:** The Bankify application was deployed on a web server, such as Apache, and hosted on a PHP hosting server environment.

CHAPTER-2

REQUIREMENT ANALYSIS

2.1 INTRODUCTION

Requirement analysis in a project report refers to the process of identifying, documenting, and understanding the needs and expectations of stakeholders for a particular project. It involves gathering, analysing, and prioritizing requirements to ensure that the project objectives are clearly defined and met.

2.2 MODULES

The Bankify-Online Banking System is organized into several modules that provide specific functionality. These modules work together to deliver a comprehensive and seamless user experience. The main modules of the system include:

2.2.1. Registration Module

The registration module in Bankify allows users to create new bank accounts within minutes. Users provide necessary information, such as personal details and identification documents, to complete the registration process. The module includes validation checks to ensure the accuracy and completeness of the provided information. Upon successful registration, users gain access to their newly created bank account and can begin utilizing the features and services offered by Bankify.

2.2.2. Login Module

The login module in Bankify provides users with a secure and convenient way to access their accounts. Users can enter their credentials, such as username and password, to authenticate themselves and gain access to their personal banking information. The login module incorporates robust security measures, including encryption and possibly additional authentication factors, to ensure the privacy and protection of user accounts.

2.2.3. Transfer Amount Module

The transfer amount module in Bankify enables users to securely transfer funds between accounts. Users can enter the recipient's account details and specify the amount to be transferred. The module validates the transaction and facilitates the secure transfer of funds, ensuring accuracy and compliance.

2.2.4. Loan Sanction Module

The loan sanction module in Bankify provides users with the ability to apply for loans. Users can submit loan applications, providing necessary information such as the loan amount. The module includes a review and approval process, where the bank evaluates the application based on predefined criteria and determines loan eligibility. Once approved, the loan amount is sanctioned and made available to the user.

2.2.5. Close Account Module

The close account module in Bankify allows users to close their bank accounts at their convenience. Users can initiate the account closure process within the application by providing the necessary information confirming their username and password. The module ensures that all account-related transactions and activities are settled before the account is permanently closed, providing a streamlined and secure account closure process.

2.2.6. Logout Module

The logout module in Bankify allows users to securely end their session and log out of the application. Users can manually initiate the logout process, ensuring that their account remains protected from unauthorized access. Additionally, Bankify incorporates an automatic logout feature that triggers after 10 minutes of inactivity, further enhancing account security by preventing unauthorized access in case the user forgets to log out.

2.3 EXTERNAL INTERFACE REQUIREMENTS

2.3.1 User Interface

The system will have a user-friendly web-based interface that allows users to interact with the application. The user interface will be designed to provide an intuitive and seamless experience, making it easy for users to navigate, input data, and access various features.

2.3.2 Hardware Interface

- Server: The system will require a server to host the Bankify application and handle user requests and data processing.
- Client: Users will access the system through web browsers on their personal computers, laptops, or mobile devices.

2.3.3 Software Interface

The system will interact with various software components, including:

- Web browsers: The system will be compatible with popular web browsers such as Chrome, Firefox, and Safari.
- PHP: The backend of the system will be implemented using PHP for server-side processing and data retrieval.
- MySQL: The system will utilize the MySQL database management system for efficient storage and retrieval of wine-related data.

2.4 OTHER NON-FUNCTIONAL REQUIREMENTS

2.4.1 Performance Requirements

- The system should provide fast response times to ensure a smooth user experience.
- The database queries and operations should be optimized to handle large volumes of data efficiently.
- The system should be able to handle multiple concurrent users without significant performance degradation.

2.4.2 Safety Requirements

- The system should implement appropriate security measures to protect user data and prevent unauthorized access.
- User authentication and authorization mechanisms should be in place to ensure secure access to the system.

2.4.3 Software Quality Requirements

- The system should be reliable and robust, with minimal errors or failures.
- The codebase should follow good coding practices, be well-structured, and maintainable.

2.4.4 Operating Environment

- Hardware: The system should be compatible with standard hardware configurations, including personal computers, laptops, and mobile devices with internet connectivity.
- Software: The system should be compatible with widely used operating systems (Windows, macOS, Linux) and web browsers (Chrome, Firefox, Safari).

CHAPTER-3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

SRS stands for Software Requirements Specification. It is a comprehensive document that describes the requirements and specifications of a software system. The SRS serves as a communication tool between stakeholders, including clients, users, developers, and testers, to ensure a common understanding of the system's functionality and characteristics.

3.1.1 Purpose

The purpose of the Bankify application is to provide users with a convenient, secure, and user-friendly online banking experience. It aims to simplify banking tasks, such as money transfers, loan management, account creation, and closure, while ensuring the privacy and security of users' personal and financial information. The application strives to make online banking accessible to users of all technical backgrounds and age groups, offering essential banking services through a simplistic and modern user interface.

3.1.2 Intended Audience

The intended audience for the Bankify application is broad and includes individuals who are seeking a modern and convenient online banking solution. It targets users of various age groups and technical backgrounds who value simplicity, security, and ease of use in their banking experience. Bankify aims to cater to both tech-savvy individuals and those who may be less familiar with online banking, providing a user-friendly interface that is accessible to all.

3.1.3 Project Scope

The project scope for the Bankify application includes the development of a user-friendly online banking system that offers essential banking services such as money transfers, loan management, account creation, and closure. It involves designing and implementing a secure and robust application with features that cater to the needs of a broad audience. The scope also encompasses ensuring cross-platform compatibility, integrating necessary security measures, and providing reliable performance even during high traffic periods. Additionally, the project may explore integration with third-party services and include personal finance management tools to enhance the overall banking experience.

3.1.4 Benefits

- **Convenience:** Bankify provides users with the convenience of accessing their bank accounts and performing essential banking tasks anytime, anywhere, using their preferred devices. Users can manage their finances, transfer money, apply for loans, and perform other banking operations without the need to visit a physical branch.
- **User-Friendly Interface:** Bankify features a user-friendly interface that is intuitive and easy to navigate. Users of all technical backgrounds and age groups can quickly adapt to the application and carry out banking transactions seamlessly, making it accessible to a wide range of users.
- **Enhanced Security:** Bankify prioritizes the security and privacy of users' personal and financial information. Robust security measures, such as data encryption and two-factor authentication, ensure that transactions and data remain secure, instilling confidence in users to trust the application with their sensitive information.
- **Time and Cost Savings:** Bankify eliminates the need for users to physically visit a bank branch for routine banking tasks. This saves users time and transportation costs, allowing them to manage their finances efficiently from the comfort of their own homes or any location with an internet connection.
- **Time and Cost Savings:** Bankify eliminates the need for users to physically visit a bank branch for routine banking tasks. This saves users time and transportation costs, allowing them to manage their finances efficiently from the comfort of their own homes or any location with an internet connection.

3.2 OVERALL DESCRIPTION

3.2.1 Identification of Existing Work

The identification of existing work for the Bankify application involves reviewing and analysing similar online banking applications currently available in the market. This includes studying their features, user interfaces, security measures, and overall user experience. By examining existing work, Bankify can gain insights into industry best practices, identify areas for improvement, and differentiate itself by offering unique features and a superior user experience. Additionally, it helps ensure that Bankify aligns with regulatory and compliance requirements and takes into account any industry standards or guidelines relevant to online banking systems.

3.2.2 Product Perspective

The Bankify application is developed as a standalone online banking system that operates independently from other products or systems. It provides a comprehensive and self-contained platform for users to manage their banking activities. While it may integrate with external services such as payment gateways or investment platforms, Bankify is designed to function as a distinct and self-sufficient product. It aims to offer a seamless and user-friendly online banking experience while ensuring the security and privacy of users' financial information.

3.2.3 Product Features

- **Account Management:** Users can create new bank accounts within minutes and manage their existing accounts, including checking balances, viewing transaction history, and updating personal information.
- **Money Transfer:** Users can securely transfer funds between their own accounts, make payments to other Bankify users, and initiate transfers to external bank accounts using standard transfer methods.
- **Loan Management:** Bankify provides a platform for users to apply for loans, view loan details, make loan payments, and track loan repayment schedules.
- **Account Closure:** Users have the option to close their bank accounts through the application, providing a streamlined process to terminate their account when needed.
- **Enhanced Security Measures:** Bankify incorporates robust security measures such as data encryption, two-factor authentication, and real-time fraud detection to safeguard user accounts and transactions.

3.2.4 End User Characteristics

- **Diverse Age Groups:** The Bankify application caters to users of different age groups, ranging from young adults to senior citizens, providing a user-friendly interface that accommodates users with varying levels of technical proficiency.
- **Varied Technical Backgrounds:** Bankify is designed to be accessible to users with different levels of technological knowledge, catering to both tech-savvy individuals and those who may have limited experience with online banking.
- **Financial Management Needs:** The application targets users who have various financial management needs, including individuals who seek basic banking services such as money transfers and account management, as well as those interested in advanced financial planning and goal tracking.

- **Security-Conscious Users:** Bankify appeals to users who prioritize the security and privacy of their personal and financial information. These users value robust security measures and are conscious of potential risks associated with online banking.
- **Convenience Seekers:** Bankify targets users who desire the convenience and flexibility of managing their banking activities from anywhere, at any time, without the constraints of physical branch visits.

CHAPTER-4

ANALYSIS AND DESIGN

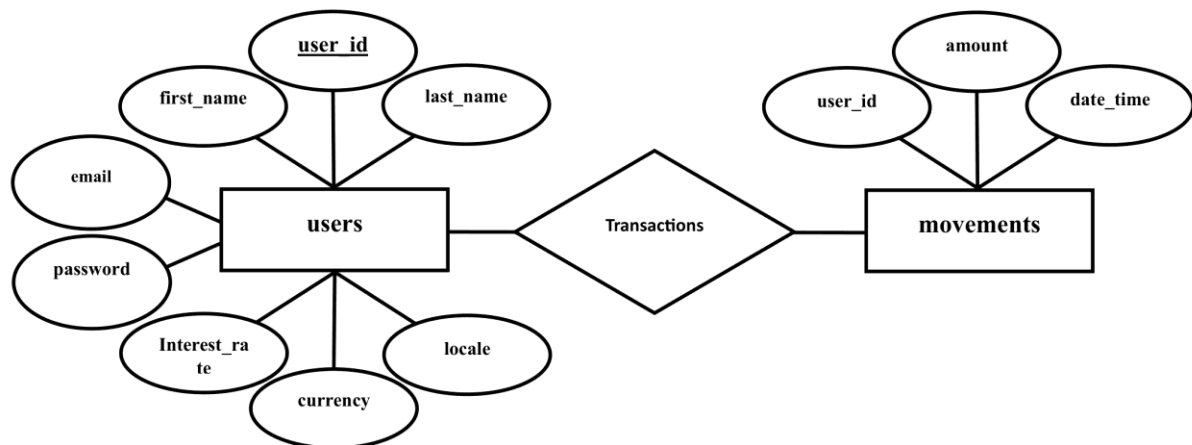
4.1 DEFINITION

Analysis is a broad term, best qualified, as in requirements analysis (an investigation of the requirements) or object analysis (an investigation of the domain objects).

Design emphasizes a conceptual solution that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects. Ultimately, designs can be implemented.

4.2 ER DESIGN

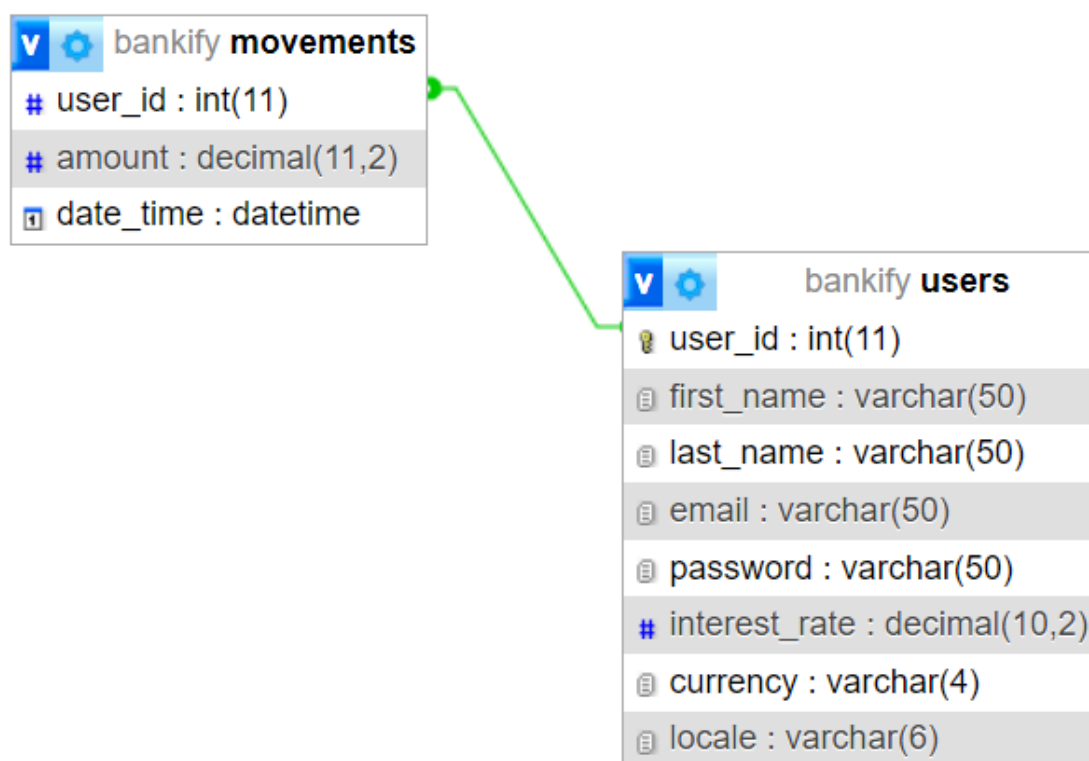
In the Bankify system, the ER design will capture key entities involved in online banking management. These entities include users, accounts, transactions, loans, and security measures. Each entity will have its attributes that represent the specific information associated with it.



4.3 SCHEMA DESIGN

The schema design is a critical step in the development of the Bankify-Online Banking System. It involves translating the Entity-Relationship (ER) design into a detailed and structured database schema. The schema design defines the tables, columns, data types, and relationships that form the foundation of the database.

Here is the schema design for the Bankify system, based on the entities and relationships identified in the ER design:



Schema Diagram for Bankify Web Application

CHAPTER-5

IMPLEMENTATION

5.1 INTRODUCTION

The implementation of the Bankify Online-Banking System involves the client side as well as server-side scripting which includes the logic for working of the system. It also includes creation and manipulation of the database using MySQL. This chapter provides an overview of the implementation process and covers the details of various operations such as CREATE, INSERT, UPDATE, ALTER, DROP, DELETE, and SELECT.

5.2 IMPLEMENTATION DETAILS

5.2.1 Pseudo code for login

FUNCTION login(username, password):

 IF isCredentialsValid(username, password) THEN

 SET user = getUserByUsername(username)

 SET isLoggedIn = true

 DISPLAY "Login successful. Welcome, " + user.name + "!"

 ELSE

 DISPLAY "Invalid username or password. Please try again."

 END IF

END FUNCTION

FUNCTION isCredentialsValid(username, password):

 IF usernameExists(username) AND passwordMatches(username, password) THEN

 RETURN true

 ELSE

 RETURN false

 END IF

END FUNCTION

FUNCTION usernameExists(username):

 // Check if the username exists in the database or user repository

 // Return true if the username exists, false otherwise

END FUNCTION

FUNCTION passwordMatches(username, password):

 // Retrieve the hashed password for the given username from the database or user repository

 // Compare the hashed password with the provided password using a secure hashing algorithm

```
// Return true if the passwords match, false otherwise  
END FUNCTION
```

```
FUNCTION getUserByUsername(username):  
    // Retrieve the user object from the database or user repository based on the username  
    // Return the user object  
END FUNCTION
```

5.2.2 Pseudo code for transfer amount

```
FUNCTION transferAmount(senderAccountNumber, recipientAccountNumber, amount):  
    SET senderAccount = getAccountByAccountNumber(senderAccountNumber)  
    SET recipientAccount = getAccountByAccountNumber(recipientAccountNumber)  
    IF senderAccount is null OR recipientAccount is null THEN  
        DISPLAY "Invalid account number. Please check and try again."  
        RETURN  
    END IF  
    IF senderAccount.balance < amount THEN  
        DISPLAY "Insufficient funds. Transfer cannot be completed."  
        RETURN  
    END IF  
    SET newSenderBalance = senderAccount.balance - amount  
    SET newRecipientBalance = recipientAccount.balance + amount  
    UPDATE accountBalance(senderAccountNumber, newSenderBalance)  
    UPDATE accountBalance(recipientAccountNumber, newRecipientBalance)  
    DISPLAY "Transfer of amount " + amount + " successful."  
END FUNCTION  
  
FUNCTION getAccountByAccountNumber(accountNumber):  
    // Retrieve the account object from the database or account repository based on the account  
    number  
    // Return the account object if found, otherwise return null  
END FUNCTION  
  
FUNCTION UPDATE accountBalance(accountNumber, newBalance):  
    // Update the account balance in the database or account repository for the given account number  
END FUNCTION
```

5.2.3 Pseudo code for loan sanction

```
function loanSanction(user, loanAmount):  
    // Check user eligibility for loan  
    if user.creditScore >= minimumCreditScore:  
        // Calculate loan eligibility based on user's income and credit history  
        loanEligibility = calculateLoanEligibility(user.income, user.creditHistory)  
        // Check if loan amount requested is within the user's eligibility  
        if loanAmount <= loanEligibility:  
            // Perform additional checks, such as debt-to-income ratio  
            // Verify user's employment and income stability  
            // Generate loan agreement and terms  
            // Store loan details in the database  
            loanId = storeLoanDetails(user.userId, loanAmount, loanEligibility)  
            // Send loan approval notification to the user  
            sendNotification(user.email, "Loan Approved", "Congratulations! Your loan has been  
approved.")  
            // Deduct loan amount from Bankify's funds or allocate funds from a lending institution  
            // Update user's account balance or create a loan account  
            // Return success status and loan ID  
            return {status: "success", loanId: loanId}  
        else:  
            // Loan amount exceeds user's eligibility  
            return {status: "error", message: "Loan amount exceeds your eligibility."}  
    else:  
        // User's credit score is below the minimum requirement  
        return {status: "error", message: "Sorry, your loan application has been rejected due to a  
low credit score."}
```

5.2.4 Pseudo code for close account

```
function closeAccount(user):  
    // Check if the user has any pending transactions or outstanding balances  
    if user.hasPendingTransactions() or user.hasOutstandingBalances():  
        // Display error message to the user that the account cannot be closed at the moment  
        displayErrorMessage("Unable to close the account. Please settle all pending transactions  
and outstanding balances.")  
    else:  
        // Prompt the user to confirm the account closure  
        confirmation = displayConfirmationDialog("Are you sure you want to close your  
account?")  
        if confirmation == "Yes":  
            // Perform necessary steps to close the account  
            // Transfer any remaining funds to the user's linked account  
            // Update account status to "Closed" in the database  
            // Notify the user about the successful closure of the account  
            sendNotification(user.email, "Account Closure", "Your Bankify account has been  
successfully closed.")  
            // Display success message to the user  
            displaySuccessMessage("Your account has been successfully closed.")  
        else:  
            // User opted not to close the account  
            displayMessage("Account closure request canceled. Your account remains active.")
```

5.2.5 Pseudo code for logout

```
function logout(user):  
    // Set user's session timeout to 10 minutes  
    setUserSessionTimeout(user, 10)  
    // Start session timer  
    startSessionTimer()  
    // Monitor user activity  
    while user.isActive():  
        // Check if session timer has expired  
        if sessionTimerExpired():  
            // Perform automatic logout  
            performLogout(user)  
            // Display logout message to the user  
            displayLogoutMessage("You have been automatically logged out due to inactivity.")  
            // Exit the loop  
            break  
        // Continue monitoring user activity  
    // Clear user session  
    clearUserSession(user)  
    // Display logout message to the user  
    displayLogoutMessage("You have been successfully logged out.")  
    // Redirect user to the login page  
    redirectToLoginPage()
```


CHAPTER-6

SOFTWARE TESTING

6.1 INTRODUCTION

Software testing is a crucial phase in the software development lifecycle. It ensures that the implemented system functions correctly, meets the requirements, and operates smoothly. This chapter focuses on the testing process for the Bankify-Online-Banking System.

6.2 TESTING OBJECTIVE

The primary objective of software testing is to identify defects, errors, and deviations from the expected behaviour of the system. The testing process aims to ensure the reliability, functionality, and quality of the software. For the Bankify Online-Banking System, the testing objectives include:

- Validate the functionality of each module.
- Verify that the system meets the specified requirements.
- Identify and fix any defects or issues.
- Ensure that the system operates smoothly and efficiently.

6.3 TEST CASES

To ensure comprehensive testing, test cases need to be designed for each module of the Bankify Online-Banking System. Test cases are scenarios or situations that are executed to validate the behaviour and performance of the system.

6.3.1 Registration Module

SL No	Test Condition	Result
1.	Verify that a user can successfully register with valid credentials	Successful
2.	Validate that the system prevents registration with an already existing email address	Successful
3.	Ensure that the system enforces password strength requirements during registration	Successful

6.3.2 Login Module

SL No	Test Condition	Result
1.	Verify that a user can successfully log in with valid email and password	Successful
2.	Validate that the system displays an error message when an incorrect email or password is entered during login	Successful
3.	Validate that the system handles and displays appropriate error messages when the fields are empty during login	Successful

6.3.3 Transfer Amount Module

SL No	Test Condition	Result
1.	Verify that a user can successfully transfer a valid amount from their account to another user's account	Successful
2.	Validate that the system displays an error message when the transfer amount exceeds the available balance in the sender's account	Successful
3.	Ensure that the system prevents the transfer of a negative or zero amount	Successful
4.	Verify that the system updates the account balances of both the sender and receiver accurately after a successful transfer	Successful
5.	Ensure that the system handles and displays appropriate error messages when the sender or receiver account does not exist or is inactive	Successful
6.	Verify that the system supports transferring amounts in different currencies and performs accurate currency conversions, if applicable	Successful

6.3.4 Loan Sanction Module

SL No	Test Condition	Result
1.	Verify that a user can successfully apply for a loan with valid details and receive loan approval	Successful
2.	Ensure that the system provides appropriate error messages when the user enters invalid or incomplete information during the loan application	Successful
3.	Verify that the system performs proper credit checks and eligibility criteria before approving or rejecting a loan application	Successful
4.	Ensure that the system handles and displays appropriate error messages when there are technical issues or system failures during the loan sanction process	Successful

6.3.5 Close Account Module

SL No	Test Condition	Result
1.	Verify that a user can successfully close their account and receive confirmation of the account closure	Successful
2.	Verify that the system handles and displays appropriate error messages when there are technical issues or system failures during the account closure process	Successful
3.	Ensure that the system provides proper confirmation or acknowledgment to the user after a successful account closure	Successful

6.3.6 Logout Module

SL No	Test Condition	Result
1.	Verify that a user can successfully logout from the system by clicking on the logout button or link	Successful
2.	Ensure that the system redirects the user to the appropriate landing page or login screen after successful logout	Successful
3.	Verify that the system enforces an automatic logout after 10 minutes of inactivity, preventing unauthorized access to the user's account	Successful

CHAPTER-7

CONCLUSION

The Bankify application is designed to provide a convenient, secure, and user-friendly online banking experience for its users. With its simple and essential banking features such as money transfer, loan management, account creation, and account closure, Bankify aims to address the issues faced by users with complex and difficult-to-navigate online banking applications. The user-friendly interface ensures that users of all ages and technical backgrounds can easily navigate the application and carry out transactions seamlessly.

The project's main objectives include developing a simple and intuitive user interface, providing essential banking services, ensuring the security and privacy of users' data, and developing a reliable and stable application. By achieving these objectives, Bankify aims to meet the needs of modern consumers who seek a convenient and reliable online banking solution.

The benefits of Bankify include its user-friendly interface, which makes banking accessible to all users, robust security measures to protect users' data, and uninterrupted access to banking services even during high traffic periods. Additionally, the application offers features like automatic logout after 10 minutes of inactivity to enhance security.

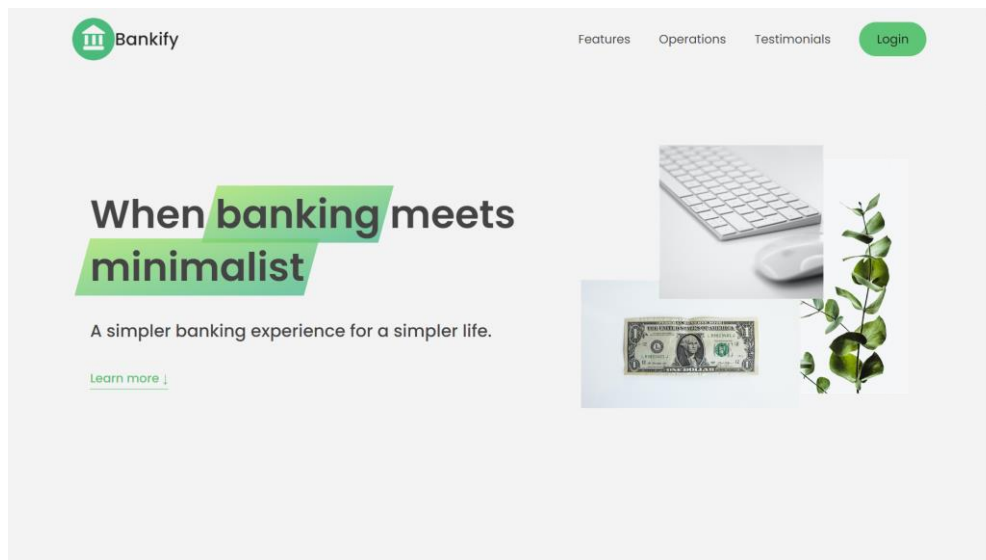
Overall, the Bankify application aims to provide a simple, secure, and user-friendly online banking experience, catering to the needs of a diverse user base. With its essential features, emphasis on security, and intuitive design, Bankify strives to be a modern and reliable solution for individuals seeking convenient online banking services.

REFERENCES

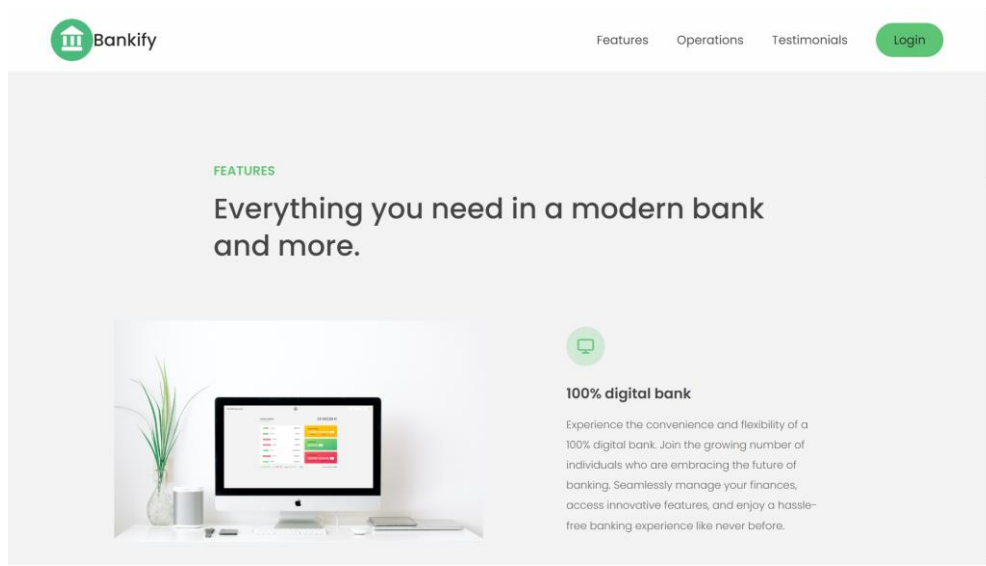
- [1]. Online Banking System Design and Implementation (2015) - Taiwo Oladipupo Ayodele and Olusegun Samuel Adeyemi. [online] Available at:
https://www.researchgate.net/publication/282373350_Liver-first_approach_as_an_unusual_treatment_modality_for_rectal_cancer_with_synchronous_liver_metastasis_Report_of_a_case [Accessed 2023].
- [2]. Designing Mobile Banking Applications for User Acceptance: A Two-Stage Perspective (2017) - Lin Zhang and Fiona Fui-Hoon Nah. [online] Available at:
https://www.researchgate.net/publication/317872759_Lifestyle_interventions_to_reduce_premature_mortality_in_schizophrenia [Accessed 2023].
- [3]. Stack Overflow - <https://stackoverflow.com/>
- [4]. W3Schools - <https://www.w3schools.com/>
- [5]. JavaScript MDN Web Docs Mozilla - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

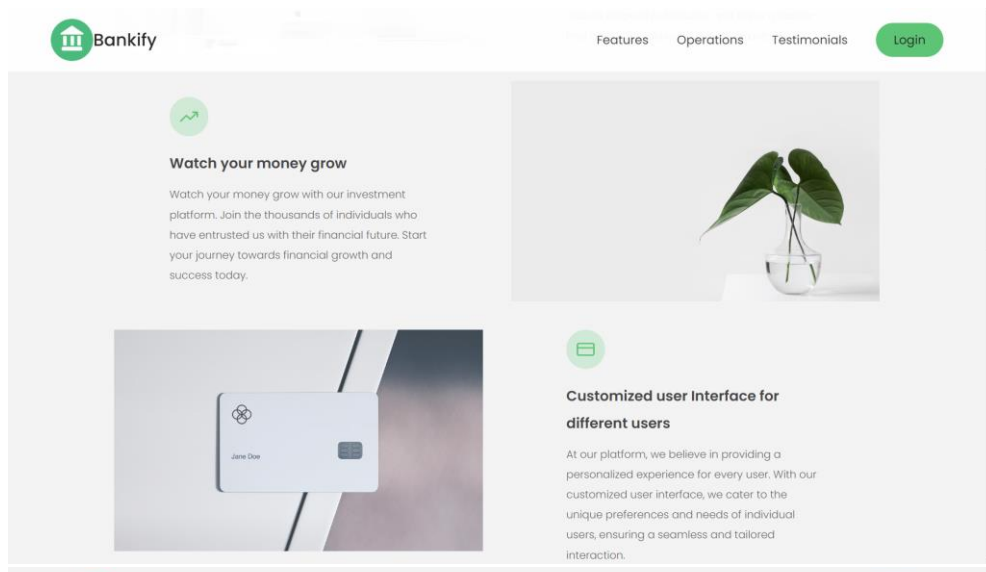
SCREEN SHOTS

1. Home page

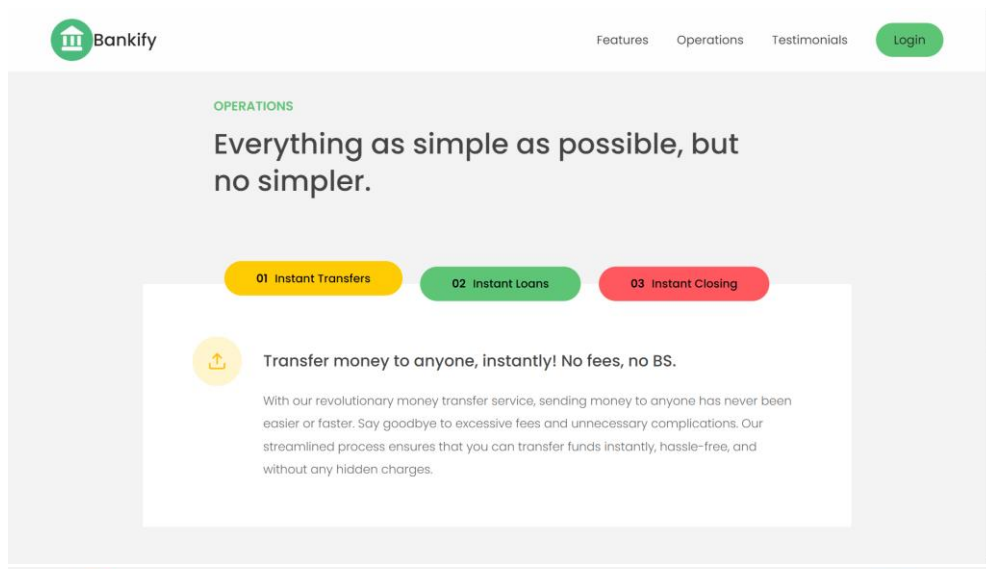


2. Features pages

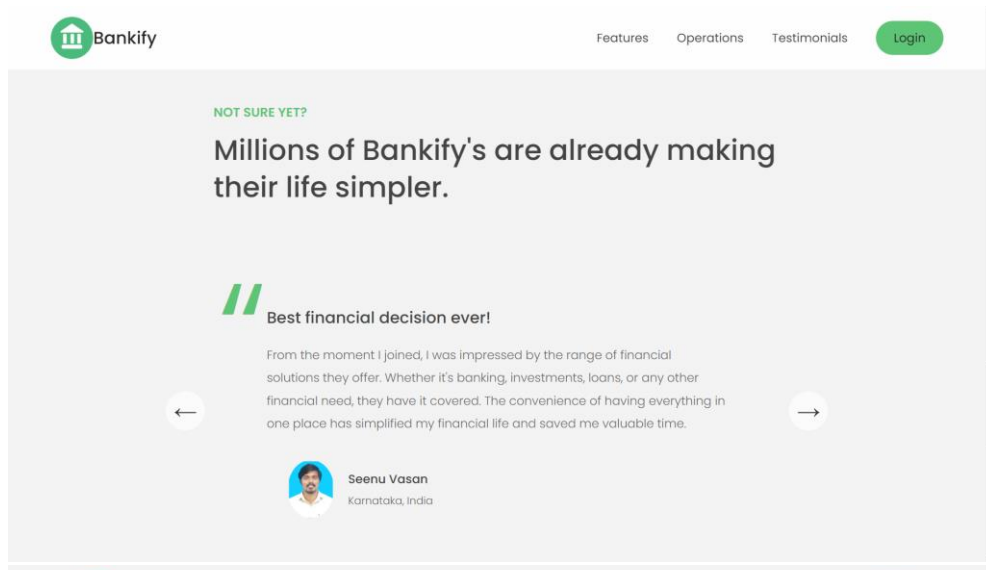




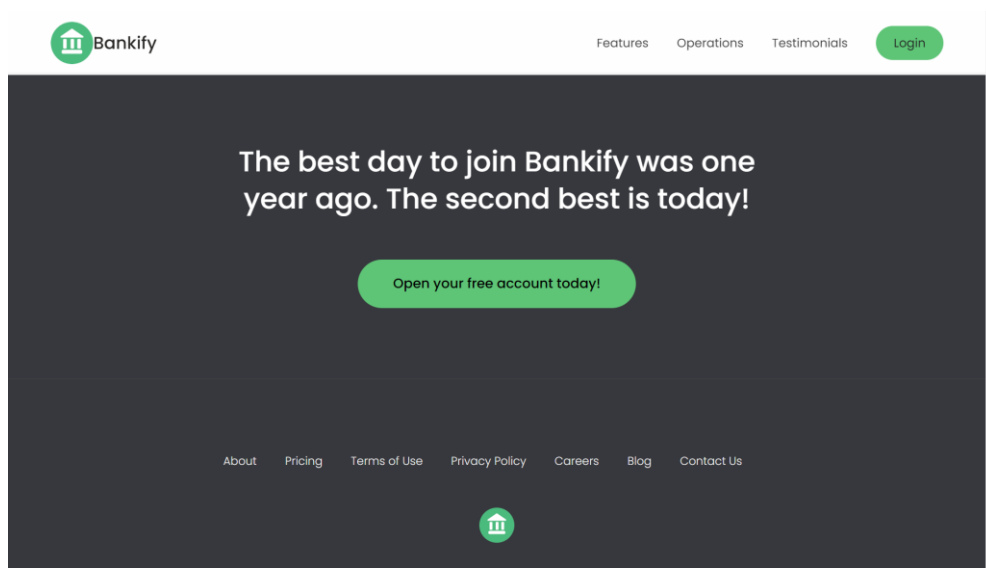
3. Operations page



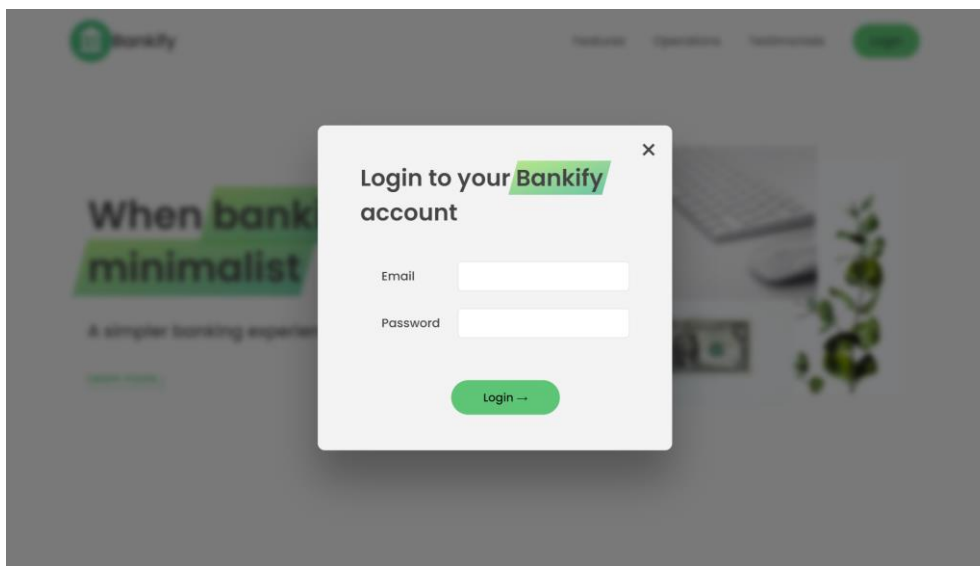
4. Testimonial page



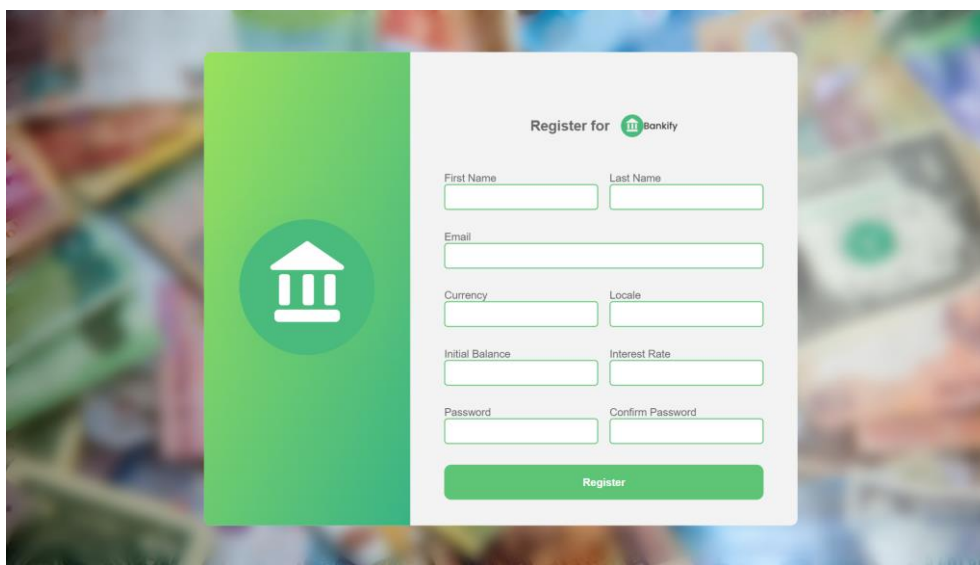
5. Footer page



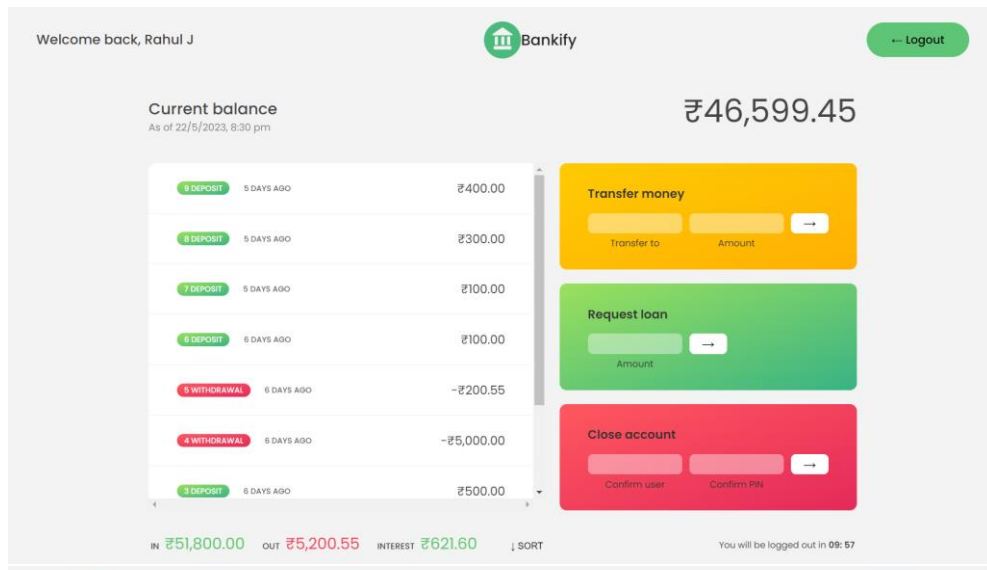
6. Login page



7. Registration page



8. Dashboard page



9. Logout page

