SMARTINTERNZ

ONLINE EASY WAY BANKING APPLICATION

SUBMITTED BY

Kavya…..	**- 20BCE7647**
ANAPARTHI SAI SINDHUJA	**- 20BCE7198**
NAGISETTY HIMA VARSHA	**- 20BCE7279**
VUDUTHA YASHASWINI	**- 20BCE7459**

GUIDED BY

Angali Singh

VITAP UNIVERSITY

AMARAVATHI

ANDHRA PRADESH, INDIA

INDEX

1. INTRODUCTION:

1.1. Overview:

A banking app is a software application that allows individuals to access and manage their bank accounts and perform various financial transactions using their smartphones or other mobile devices. These apps provide a convenient and secure way for customers to interact with their banks and perform a wide range of banking activities without needing to visit a physical branch.

Firstly, if the customer is new user, he/she has to register by filling the details then by logging with the username and password the home page of the website will be opened. Now, the customer can create or add accounts and can access other features such as monitoring account balance and transactions such as deposit, withdrawal, payment and transfer.

1.2. Purpose:

The purpose of a banking application is to provide customers with convenient and secure access to their financial accounts and services through digital channels. Customers find it more difficult and time-consuming to visit the bank physically. The proposed application can carry out tasks performed by the banking industry and can quickly complete transactions.

Through this application, users can easily check their account balances, review transactions, and access statements. They can also perform various transactions such as fund transfers, and bill payments. Overall, this application aims to provide a seamless and efficient banking experience, empowering customers to manage their finances anytime, anywhere.

2. LITERATURE SURVEY:

2.1. Existing Problem:

While banking applications provide numerous benefits and convenience, they can also encounter certain problems. Here are few common challenges faced by users of banking apps:

1. Technical Issues: Banking apps may experience technical glitches, such as crashes, slow performance, or connectivity problems. These issues can disrupt the user experience and hinder access to account information or transaction services.

3. User Authentication: Sometimes, users may encounter difficulties during the authentication process. This can include difficulties in remembering or resetting passwords.

5. Transaction Errors: Errors can occur during transactions, such as failed fund transfers or incorrect posting of transactions. These errors may require manual intervention or contact with customer support to resolve, causing inconvenience and potential financial complications.

2.2. Proposed Solution:

Here are potential solutions to address the problems faced in banking apps:

1. Technical Issues:
   - Regular testing and quality assurance to identify and resolve bugs before app releases.
   - Continuous monitoring of app performance and prompt troubleshooting of technical issues.
   - Timely updates and patches to address known bugs and improve app stability.

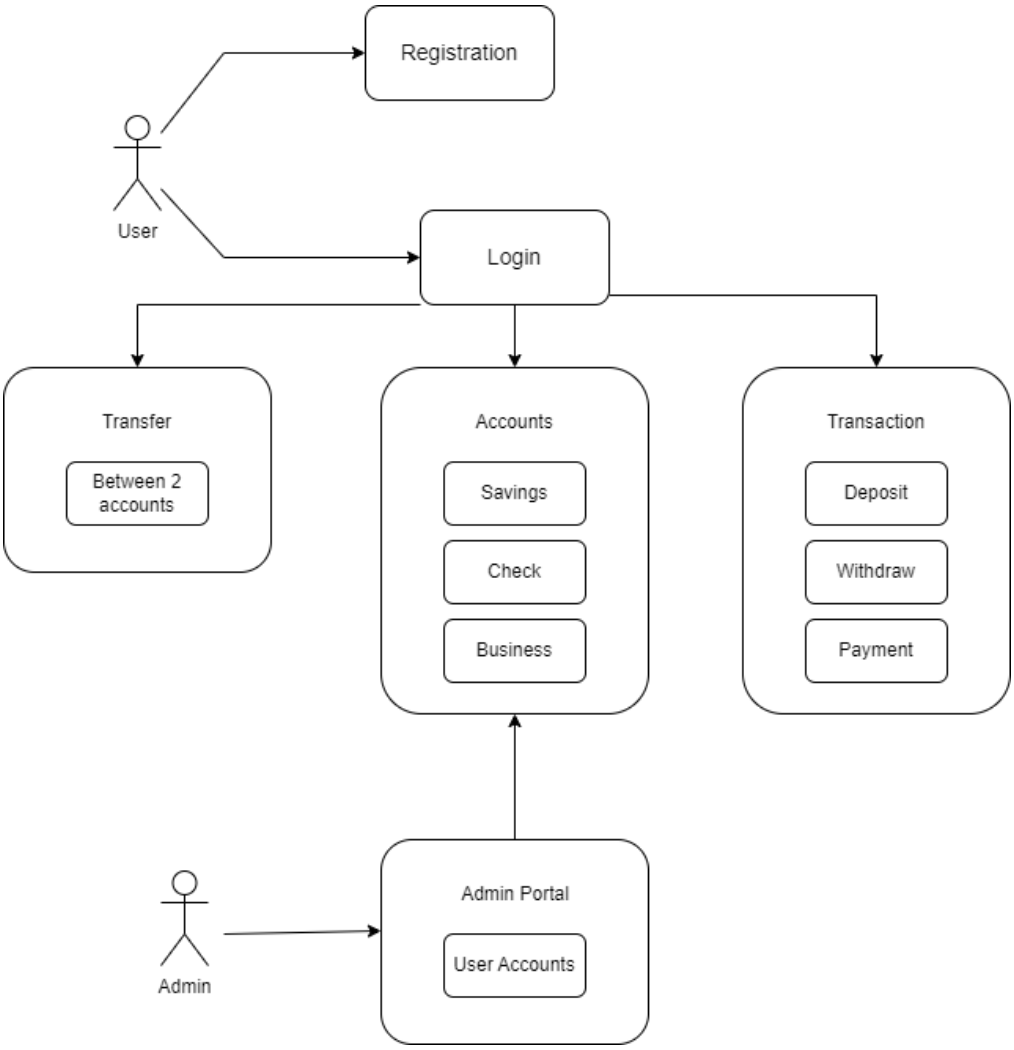
2. User Authentication:
   - Simplify password reset processes and provide clear instructions for users to reset their credentials easily.

3. Transaction Errors:
   - Enhance transaction validation processes to minimize errors and improve the accuracy of fund transfers.
   - Provide clear error messages and instructions for users to resolve transaction-related issues.

3. THEORITICAL ANALYSIS:

3.1. Block Diagram:

The diagrammatic overview of this project is as follows



3.2. Hardware / Software Designing:

The hardware and software requirements of the project are java, spring boot and the relevant links for download and installation are given as follows:

i) **Java Development Kit (JDK):** JDK is required to compile and run Java applications, providing the necessary tools and libraries. Download and install the latest JDK version from Oracle's website.

Download JDK: https://www.oracle.com/java/technologies/javase-jdk11-downloads.html

ii) **Integrated Development Environment (IDE):** An IDE offers a comprehensive development environment for writing, debugging, and managing code. IntelliJ IDEA, Eclipse, or Visual Studio Code are popular choices for Java development.

Visual Studio Code: https://code.visualstudio.com/download

iii) **Spring Boot:** Spring Boot simplifies Java application development by providing predefined configurations, automatic dependency management, and a streamlined development experience. Use Spring Initializr or Spring Tools for your IDE to create a Spring Boot project.

Spring Tools for Visual Studio Code: Install via Extensions in Visual Studio Code

iv) **MySQL Database:** MySQL is a popular relational database management system. Install MySQL Community Server and MySQL Workbench which is a graphical tool for managing MySQL databases.

MySQL Community Server: https://dev.mysql.com/downloads/installer/

MySQL Workbench: https://dev.mysql.com/downloads/workbench/

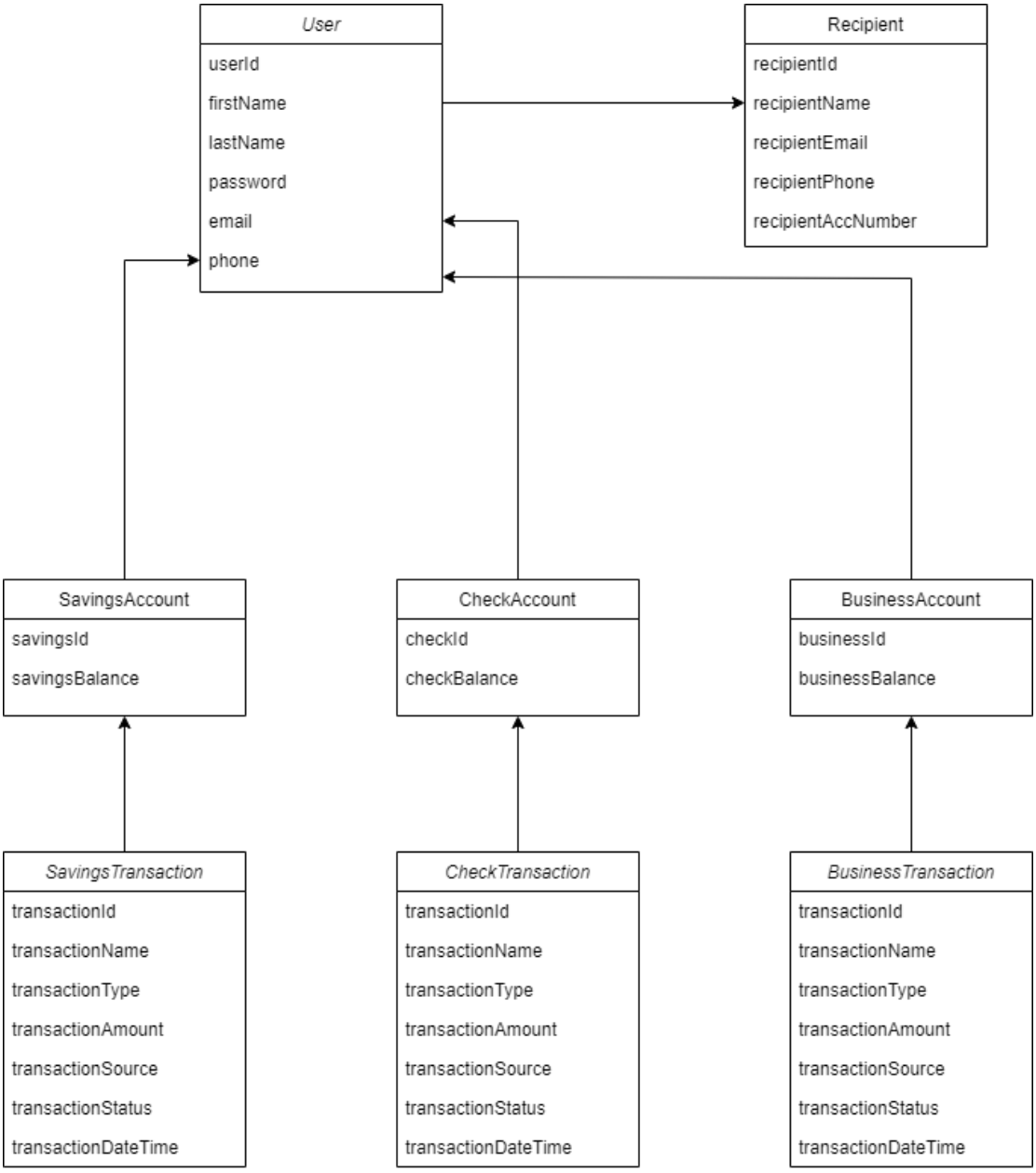
v) **MySQL Connector/J:** It is the official JDBC (Java Database Connectivity) driver that connects Java programmes with MySQL databases. A standard interface for connecting Java programs with various databases is provided via the JDBC Java API. This API is implemented particularly for MySQL databases by MySQL Connector/J, enabling Java programs to communicate with MySQL databases using standard JDBC methods and classes.

4. EXPERIMENTAL INVESTIGATIONS:

i) Usability Testing: It involves monitoring customers as they utilise an online banking platform to complete activities. We examined how simple it is for customers to access different functions, perform transactions, and navigate the user interface.

ii) Security Testing: The main aspect of online banking is security. We verifed whether the customer data is being secured properly or not. In this application, Spring Boot provides robust security features through integration with Spring Security.

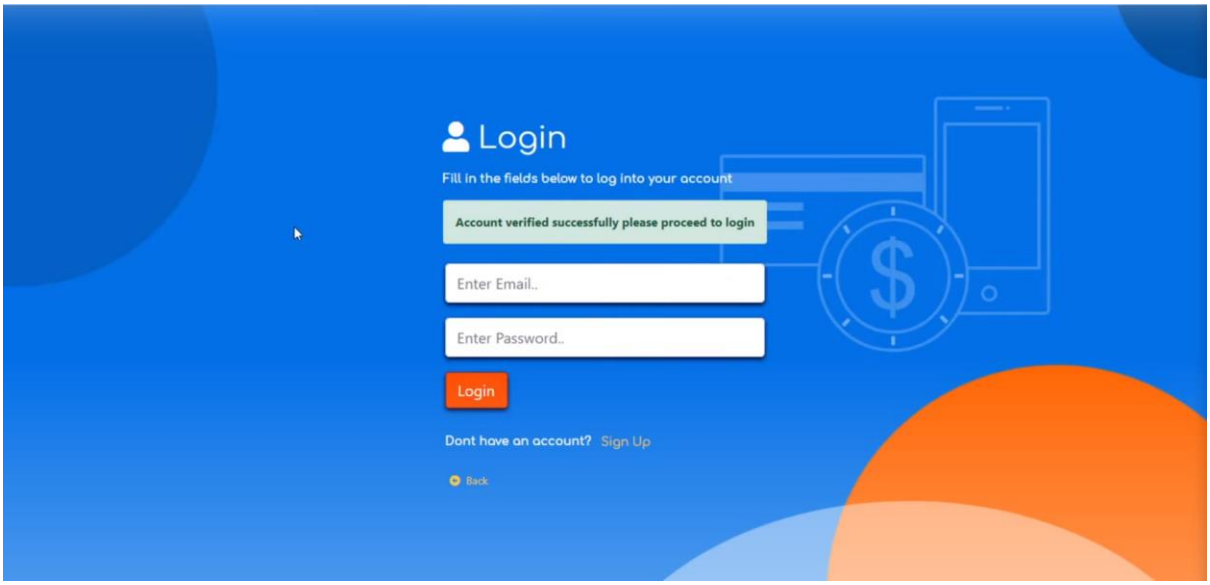
5. FLOWCHART:


User
userId
firstName
lastName
password
email
phone
Recipient
recipientId
recipientName
recipientEmail
recipientPhone
recipientAccNumber
SavingsAccount
savingsId
savingsBalance
CheckAccount
checkId
checkBalance
BusinessAccount
businessId
businessBalance
SavingsTransaction
transactionId
transactionName
transactionType
transactionAmount
transactionSource
transactionStatus
transactionDateTime
CheckTransaction
transactionId
transactionName
transactionType
transactionAmount
transactionSource
transactionStatus
transactionDateTime
BusinessTransaction
transactionId
transactionName
transactionType
transactionAmount
transactionSource
transactionStatus
transactionDateTime

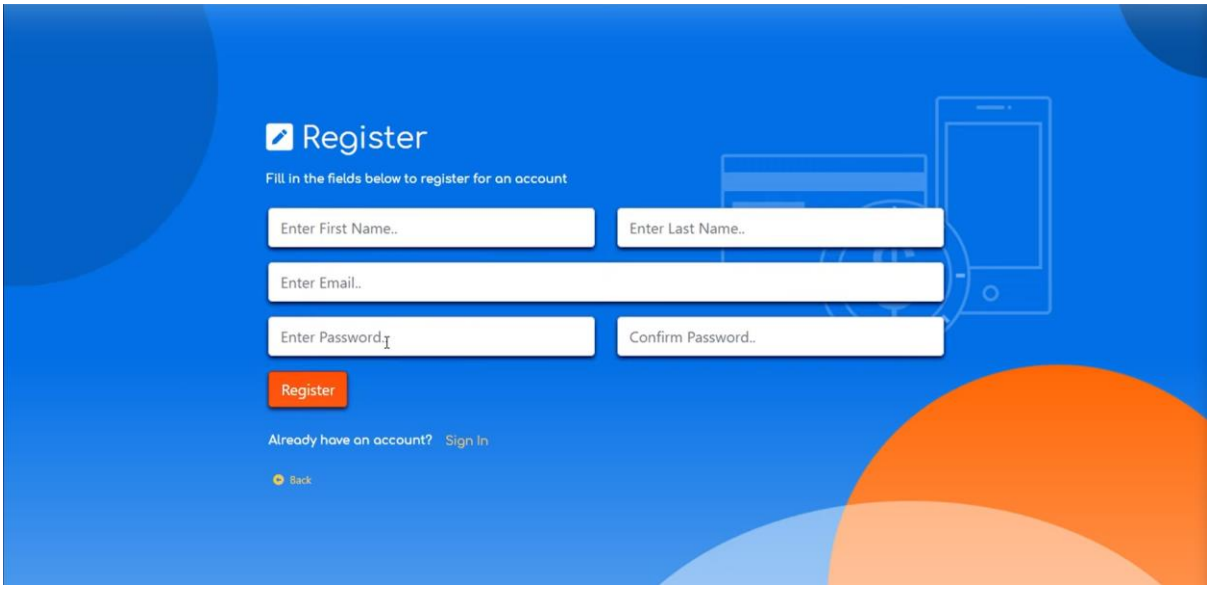
6. RESULT:

Index page:

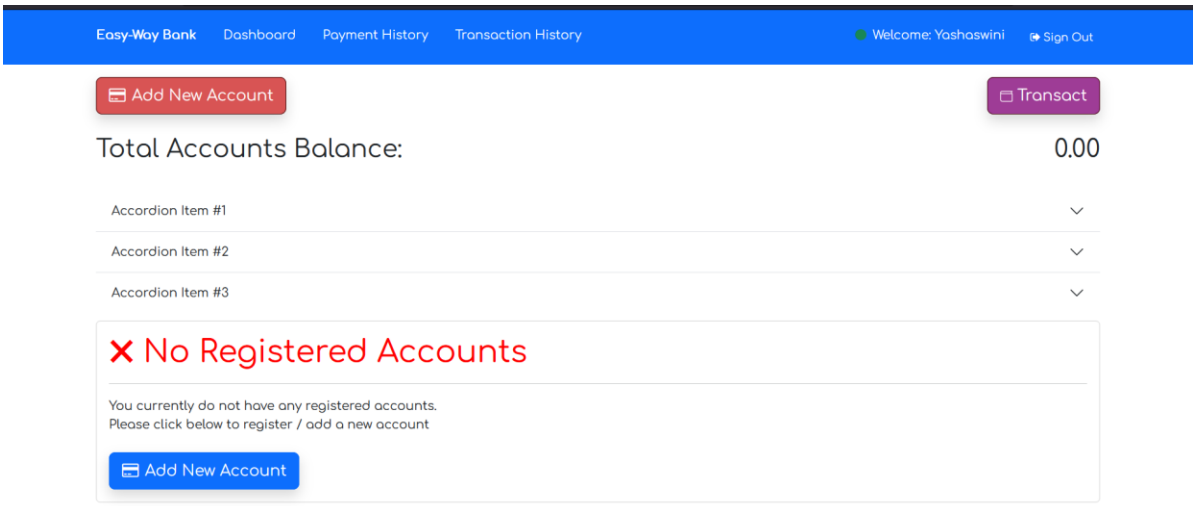
Login Page:



Register Page:



Home Page:



Error Page:

7. ADVANTAGES AND DISADVATAGES:

ADVANTAGES:

For users, the advantages of mobile banking include time efficiency, and transactions such as payments, deposits, transfer and withdraw. Additionally, users have continuous access to their accounts, including the ability to monitor their balance.

i) Time Efficiency: Customers doesnot have to go out and pay a visit to a physical bank instead they can execute different banking tasks from anywhere and at any time with the help of this application which saves customer's time.

ii) Monitoring Transactions: Monitoring and managing all of your financial transactions is one of the advantages of mobile banking services. By logging into your mobile banking app, you can monitor your bank accounts and can prevent fraudulent transactions.

iii) Security Features: Security is crucial for banking websites. Spring Boot provides robust security features through integration with Spring Security. Spring Security offers various authentication mechanisms such as username/password.

DISADVANTAGES:

i) Limited Features: In this application, there are basic features such as creating an account, and transactions such as deposit, withdrawal, payment and transfer. It does not provide the same level of functionality as traditional banking methods.

ii) Security Problems: Regardless of the security measures implemented in this application, there is always a possibility of security breaches and unauthorised access to the accounts.

8. APPLICATIONS:

i) Account Management: It involves users being able to create and manage their bank accounts. This includes features like account balance, transaction history, and payment history. Spring Boot - database integration helps in implementing these features.

ii) Payments and Transfers: Different forms of payments, inpayment, transfer, Deposit and withdrawal between accounts, can be performed via this banking application.

9. CONCLUSION:

To make it simpler for customers to execute transactions in a bank, this system is crucial for both the admin and end users. The system will make it easier for banks, members, and administrators to perform transactions with one another.

However, it still poses problems for financial security and personal privacy. Therefore, one should be mindful of the hazards while using it for financial transactions. Knowing the dangers and issues allows him to take precautions for a secure online banking experience.

10. FUTURE SCOPE:

In this banking application, few basic features are implemented such as creating an account, and transactions such as deposit, withdrawal, payment and transfer. Advanced features such as mutual funds and other demat accounts are not implemented in this application which can be done furthur.

AI-powered chatbots can be applied in this application which helps in solving basic customer quieres easily and securely.

Also, login pin or biometric authentication for the user accounts can also be implemented in this banking application furthur.

11. BIBLIOGRAPGY:

https://www.researchgate.net/publication/356775326_Mobile_Banking_App_Development_and_Implementation

Blokdyk, G., 2018. Mobile banking Standard Requirements. 1st ed. s.l.:5STARCooks.

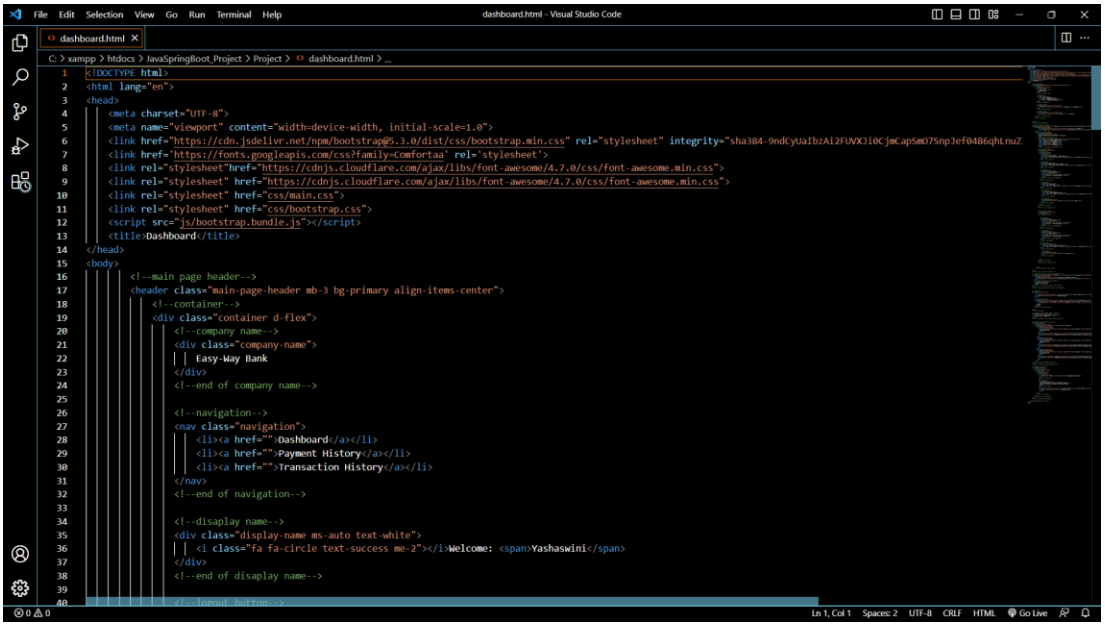
https://www.inettutor.com/source-code/online-banking-system-in-springboot/?expand_article=1

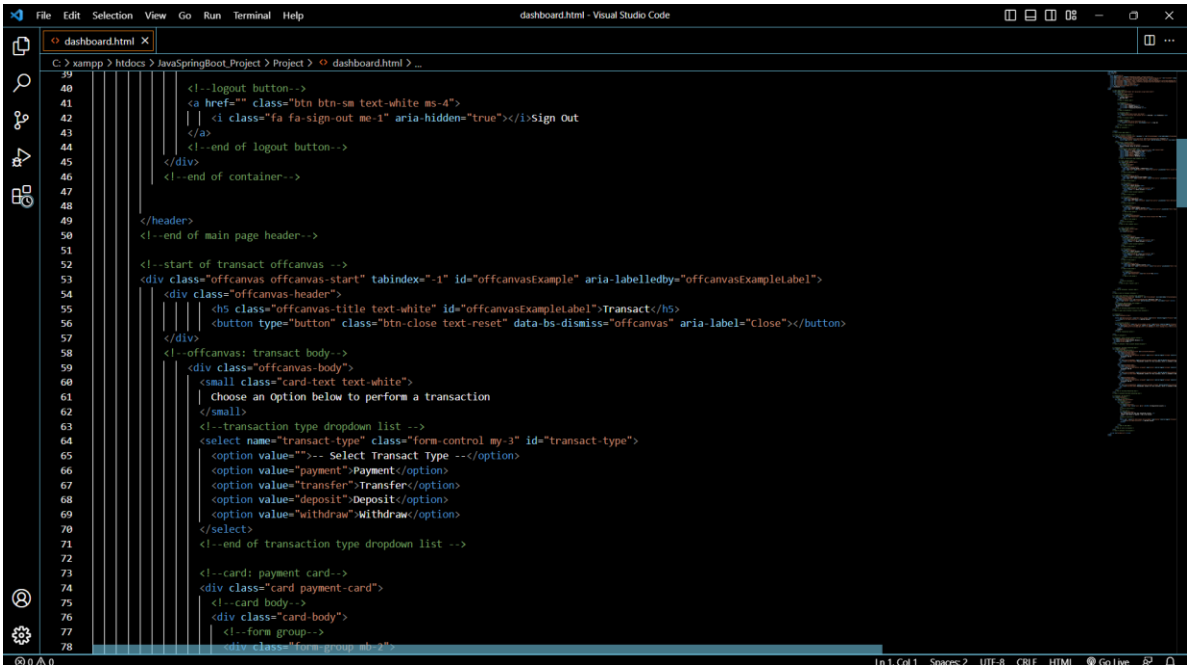
https://inoxoft.com/blog/what-is-mobile-banking-advantages-and-disadvantages-of-mobile-banking/#:~:text=The%20benefits%20of%20mobile%20banking,bills%20payments%2C%20loans%2C%20etc.

APPENDIX

Source Code:

Dashboard Code:





Index Page Code:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE-edge">
        <meta name="viewport" content="width-device-width,initial-scale-1.0">
        <link rel="stylesheet" href="css/default.css">
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
crossorigin="anonymous">
        <link href='https://fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet'>
        <title>Home</title>
    </head>
    <body class="d-flex align-items-center ">
        <!--card: sample text-->
        <div id="sample-text-card" class="card col-5 bg-transparent border-0 ">
            <!--card body-->
            <div class="card-body">
                <!--card header-->
                    <h1 class="mb-3">Easy-Way Bank</h1>
                <!--end of card header-->

                <!--card title-->
                    <h5 class="card-title">
                        Flexible Banking Solution
                    </h5>
                <!--end of card title-->

                <!--card text-->
                    <p class="card-text">
```

```
                        This online banking system project benefits customers with an
appropriate interface to obtain online banking services.
                        Consequently, customers can access their accounts from anywhere.
                        Hence, the bank can manage customer requests more effectively and
efficiently.
                        Ultimately, it saves customers time since they can obtain their
account-related information without physically approaching the bank.
                    </p>
                <!--end of card text-->
                <!--bottom wrapper-->
                    <div class="button-wrapper d-flex align-items-center">
                        <a href="register.html" class="btn btn-danger register"
role="button">Register</a>     
                        <a href="login.html" class="btn btn-info login"
role="button">Login</a>
                    </div>
                <!--end of bottom wrapper-->

            </div>
                <!--end of card body-->
        </div>
            <!--end of card: sample text-->
    </body>
</html>
```

Login page code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/default.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
crossorigin="anonymous">
    <link href='https://fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet'>
    <link rel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <title>Login</title>

</head>
<body class="d-flex align-items-center justify-content-center">
    <!--card: Login form card-->
    <div class="card login-form-card col-4 bg-transparent border-0">
        <!--card body-->
        <div class="card-body">
            <!--form header-->
            <h1 class="form-header card-title mb-3">
                <i class='fa fa-user-circle'></i>Login
            </h1>
            <!--end of form header-->
            <!--Login form-->
            <form action="" class="login-form">
```

```
                <!--row-->
                    <!--form group-->
                    <div class="form-group col">
                        <input type="email" name="email" class="form-control form-
control-lg" placeholder="Enter Your Email"/>
                    </div>
                    <!--end of form group-->


                    <div class="row">
                        <!--form group-->
                         <div class="form-group col">
                             <input type="password" name="password" class="form-control
form-control-lg" placeholder="Enter Password"/>
                         </div>
                        <!--end of form group-->


                    </div>
                <!--end of row-->

                <!--form group-->
                <div class="form-group col">
                    <button class="btn btn-info">Login</button>
                </div>
                <!--end of form group-->

            </form>
            <!--end of Login form-->

            <!--card text-->
            <p class="card-text text-white my-2">
                Don't have an account? <span class="ms-2 "><a href="register.html"
class="btn bt-sm text-warning">Sign Up</a></span>
            </p>
            <!--end of card text-->

            <!--back button to loading page-->
            <small class="text-warning" >
                <i class="fa fa-hand-o-left me-1"></i><a href="index.html" class="btn
btn-primary text-warning">Back</a>
            </small>
            <!--end of back button to loading page-->

        </div>
        <!--end of card body-->


    </div>
    <!--end of card:Login form card-->

</body>
</html>
```

Register page code:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/default.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
crossorigin="anonymous">
    <link href='https://fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet'>
    <link rel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <title>Register</title>

</head>
<body class="d-flex align-items-center justify-content-center">
    <!--card: registration form card-->
    <div class="card registration-form-card col-6 bg-transparent border-0">
        <!--card body-->
        <div class="card-body">
            <!--form header-->
            <h1 class="form-header card-title mb-3">
                <i class='fa fa-edit'></i>Register
            </h1>
            <!--end of form header-->

            <!--registration form-->
            <form action="" class="reg-form">

             <!--row-->
                <div class="row">
                    <!--form group-->
                        <div class="form-group col">
                            <input type="text" name="first_name" class="form-control
form-control-lg" placeholder="Enter First Name"/>
                        </div>
                    <!--end of form group-->

                    <!--form group-->
                        <div class="form-group col">
                            <input type="text" name="last_name" class="form-control form-
control-lg" placeholder="Enter Last Name"/>
                        </div>
                    <!--end of form group-->
                </div>
             <!--end of row-->

                    <!--form group-->
                        <div class="form-group col">
                            <input type="email" name="email" class="form-control form-
control-lg" placeholder="Enter Your Email"/>
                        </div>
                    <!--end of form group-->

             <!--row-->
                <div class="row">
                    <!--form group-->
```

```
                        <div class="form-group col">
                            <input type="password" name="password" class="form-control
form-control-lg" placeholder="Enter Password"/>
                        </div>
                    <!--end of form group-->

                    <!--form group-->
                        <div class="form-group col">
                            <input type="password" name="confirm_password" class="form-
control form-control-lg" placeholder="Confirm Password"/>
                        </div>
                    <!--end of form group-->
                </div>
             <!--end of row-->


                    <!--form group-->
                        <div class="form-group col">
                            <button class="btn btn-danger">Register</button>
                        </div>
                    <!--end of form group-->

            </form>
            <!--end of registration form-->

            <!--card text-->
            <p class="card-text text-white my-2">
                Already have an account? <span class="ms-2 "><a href="login.html"
class="btn bt-sm text-warning">Sign In</a></span>
            </p>
            <!--end of card text-->

            <!--back button to loading page-->
            <small class="text-warning" >
                <i class="fa fa-hand-o-left me-1"></i><a href="index.html" class="btn
btn-primary text-warning">Back</a>
            </small>
            <!--end of back button to loading page-->

        </div>
        <!--end of card body-->


    </div>
    <!--end of card:registration form card-->

</body>
</html>
```