

SMARTINTERNZ

ONLINE EASY WAY BANKING APPLICATION

SUBMITTED BY

KODURI KAVYA

- 20BCE7647

ANAPARTHI SAI SINDHUJA

- 20BCE7198

NAGISETTY HIMA VARSHA

- 20BCE7279

VUDUTHA YASHASWINI

- 20BCE7459

GUIDED BY

Angali Singh



VITAP UNIVERSITY

AMARAVATHI

ANDHRA PRADESH, INDIA

INDEX

S.No.	Section	Topic	Pg. No.
1	1.1	Overview	3
2	1.2	Purpose	3
3	2.1	Existing problem	3
4	2.2	Proposed solution	3-4
5	3.1	Block diagram	4
6	3.2	Hardware / Software designing	5
7	4	Experimental Investigations	5
8	5	Flowchart	6
9	6	Result	7-8
10	7	Advantages and Disadvantages	9
11	8	Applications	9
12	9	Conclusion	9
13	10	Future Scope	10
14	11	Bibliography	10
15		Appendix	10-15

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.

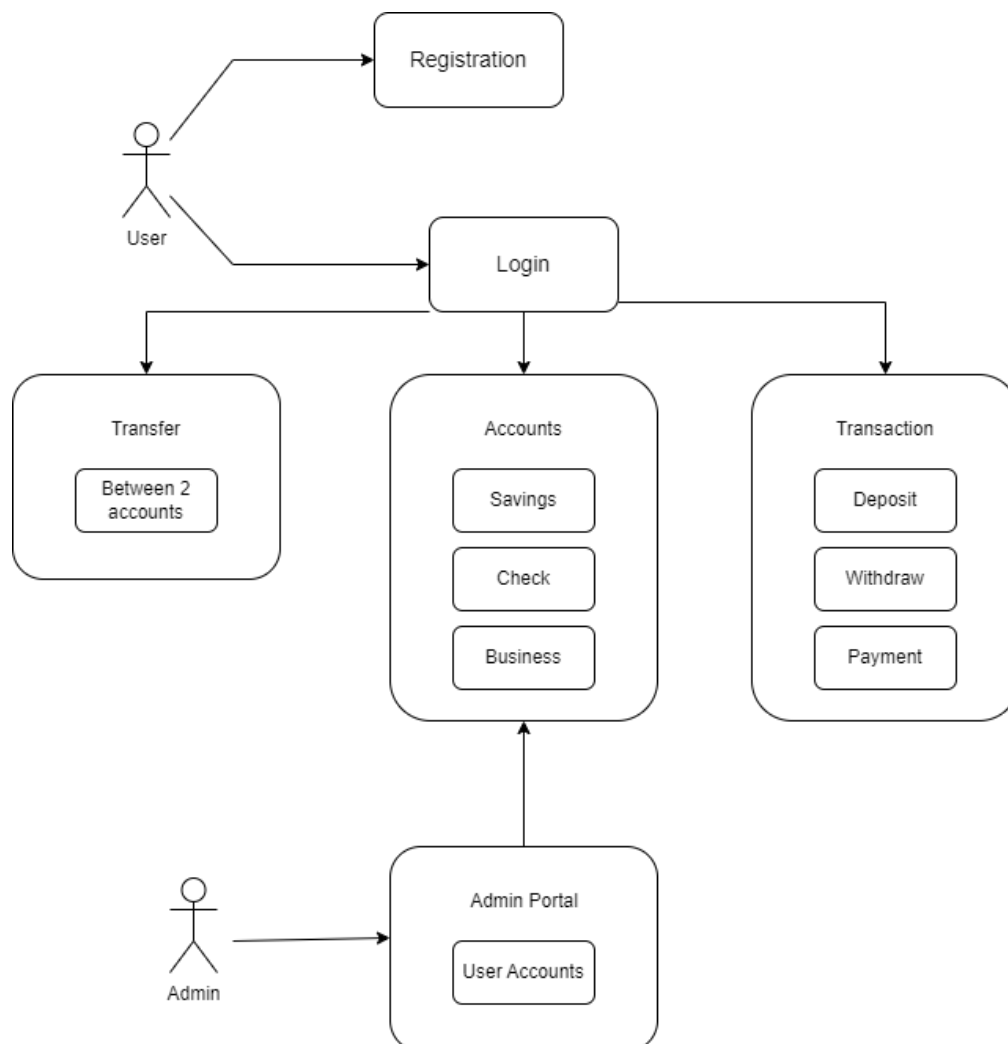
3. Transaction Errors:

- 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/>

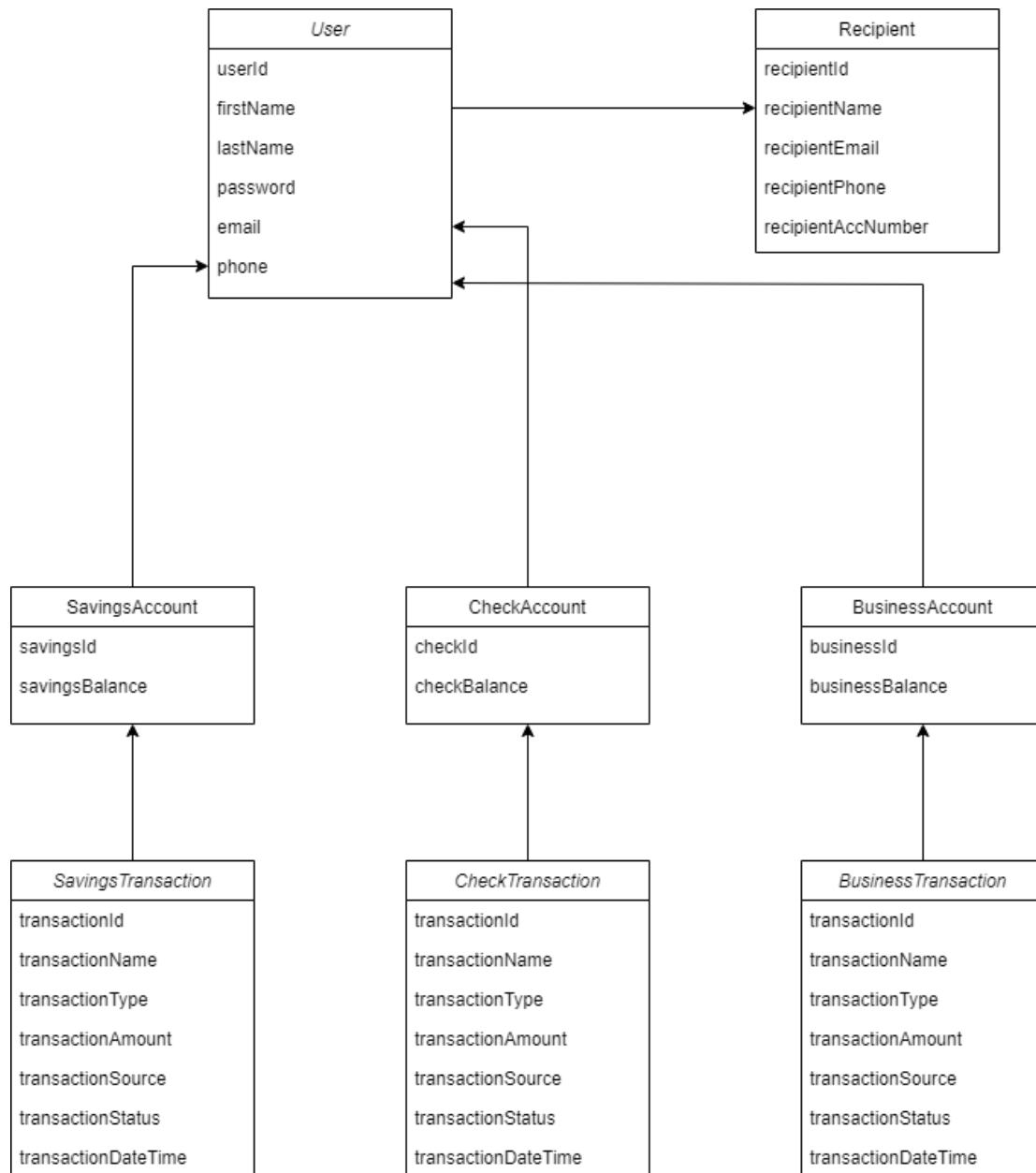
v) **MySQL Connector/J:** It is the official JDBC (Java Database Connectivity) driver that connects Java programmes with MySQL databases. A standard interface for connecting Java programs with various databases is provided via the JDBC Java API. This API is implemented particularly for MySQL databases 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 verified 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:

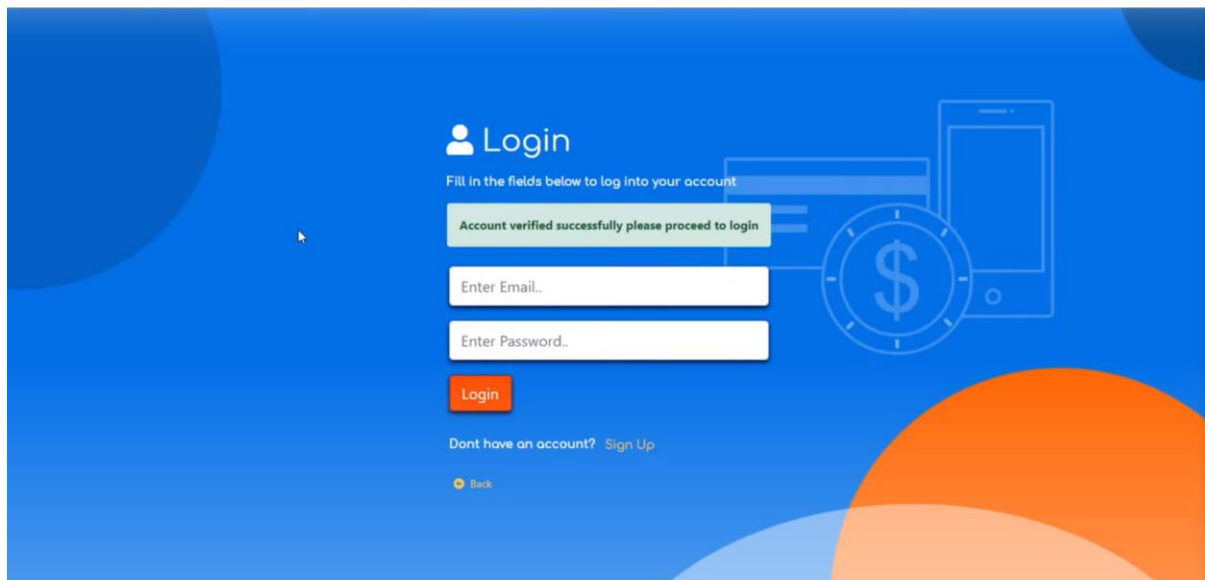


6. RESULT:

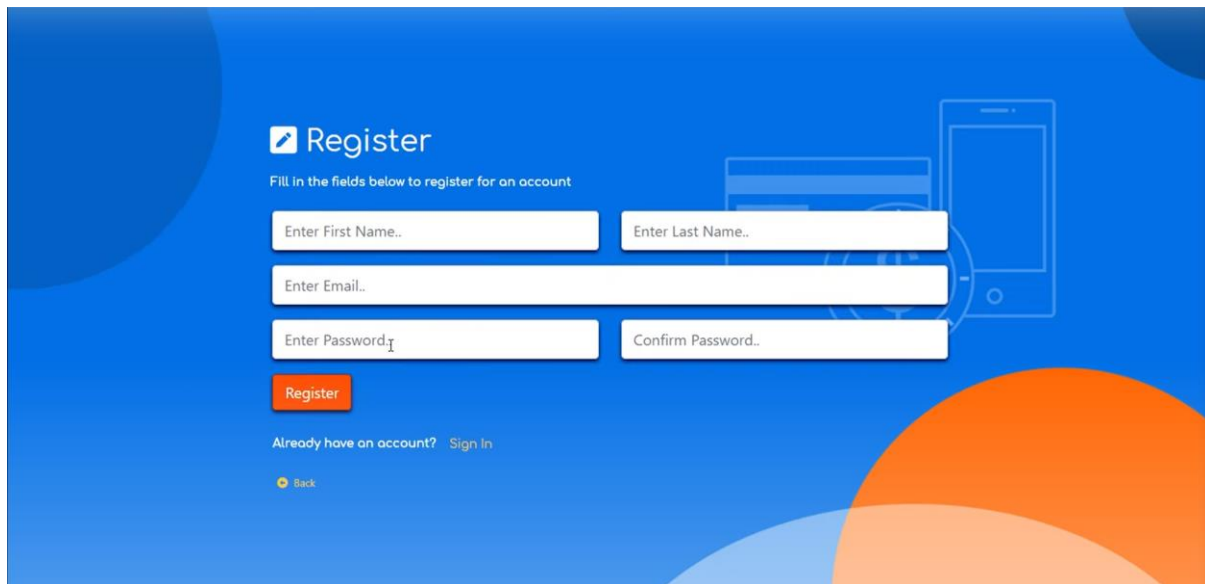
Index page:



Login Page:

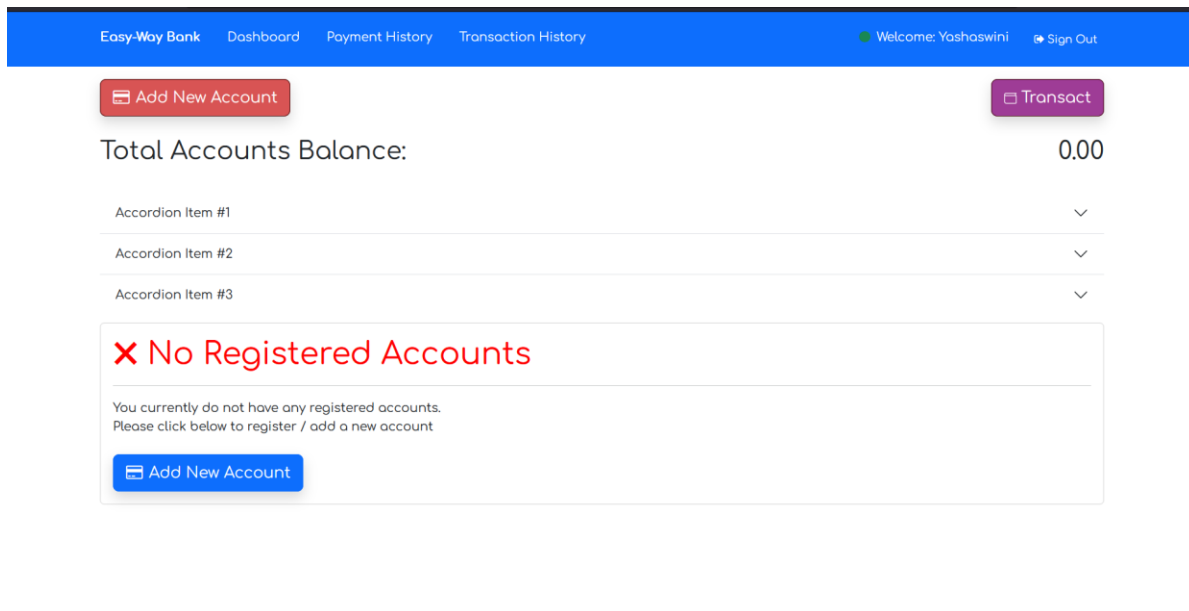


Register Page:



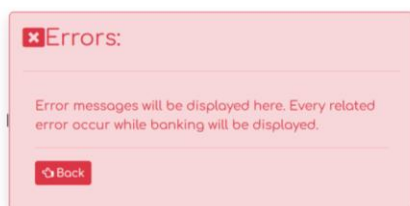
The Register page features a blue background with abstract orange and white shapes. At the top, there's a 'Register' heading with a checkmark icon. Below it, a subtext says 'Fill in the fields below to register for an account'. The form consists of five input fields: 'Enter First Name..', 'Enter Last Name..', 'Enter Email..', 'Enter Password..', and 'Confirm Password..'. A red 'Register' button is positioned below the password fields. At the bottom, there's a link 'Already have an account? Sign In' and a 'Back' button with a left arrow icon.

Home Page:



The Home page has a blue header bar with navigation links: 'Easy-Way Bank', 'Dashboard', 'Payment History', and 'Transaction History'. On the right, it says 'Welcome: Yashaswini' and 'Sign Out'. Below the header, there are two buttons: 'Add New Account' (red) and 'Transact' (purple). The main content area shows 'Total Accounts Balance: 0.00'. Below this is an accordion with three items: 'Accordion Item #1', 'Accordion Item #2', and 'Accordion Item #3', each with a downward arrow. A red error message box is displayed, stating 'No Registered Accounts' and 'You currently do not have any registered accounts. Please click below to register / add a new account'. At the bottom of the error box is an 'Add New Account' button.

Error Page:



The Error page is a light red box with a red 'x' icon and the text 'Errors:'. Below this, a message states 'Error messages will be displayed here. Every related error occur while banking will be displayed.' At the bottom is a red 'Back' button with a left arrow icon.

7. ADVANTAGES AND DISADVANTAGES:

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 do not 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 further.

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

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

11. BIBLIOGRAPHY:

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

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

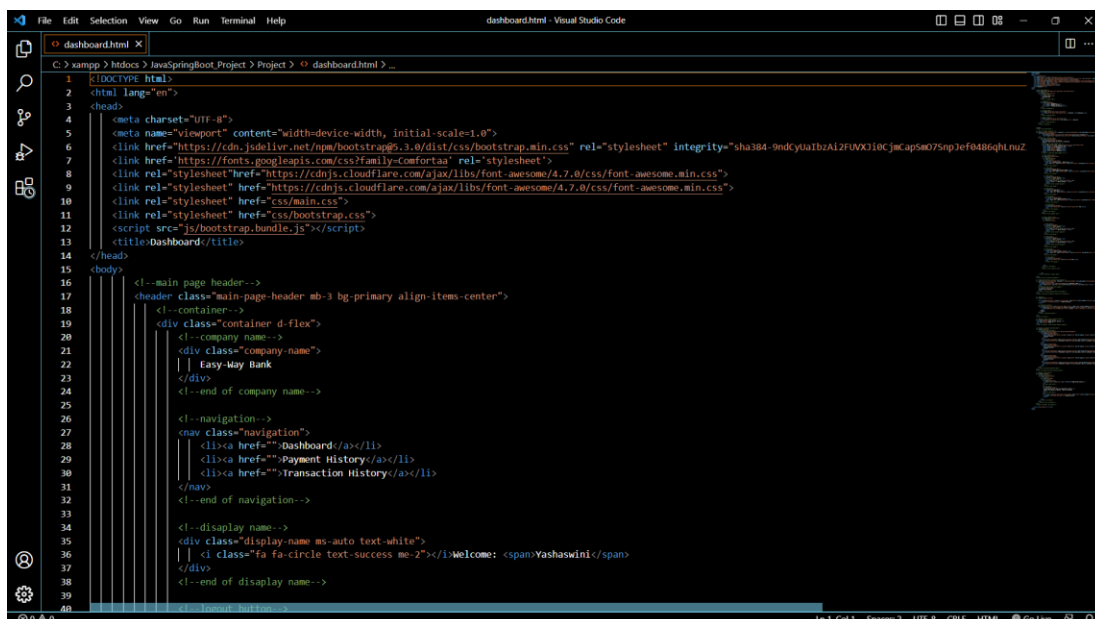
https://www.inettutor.com/source-code/online-banking-system-in-springboot/?expand_article=1

<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:



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyl1ez46j4cL44J9yqIx3A6YgZsXr983BmN7W8p1YG1q6Ukqr">
7   <link href="https://fonts.googleapis.com/css?family=Comfortaa" rel="stylesheet">
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
10  <link rel="stylesheet" href="css/main.css">
11  <link rel="stylesheet" href="css/bootstrap.css">
12  <script src="js/bootstrap.bundle.js"></script>
13  <title>Dashboard</title>
14 </head>
15 <body>
16   <!-- main page header -->
17   <header class="main-page-header mb-3 bg-primary align-items-center">
18     <!-- container -->
19     <div class="container d-flex">
20       <!-- company name -->
21       <div class="company-name">
22         | | Easy Way Bank
23       </div>
24       <!-- end of company name -->
25       <!-- navigation -->
26       <nav class="navigation">
27         <a href="#">Dashboard</a></li>
28         <a href="#">Payment History</a></li>
29         <a href="#">Transaction History</a></li>
30       </nav>
31       <!-- end of navigation -->
32       <!-- display name -->
33       <div class="display-name ms-auto text-white">
34         | | <i class="fa fa-circle text-success me-2"></i>Welcome: <span>Yashaswini</span>
35       </div>
36       <!-- end of display name -->
37     </div>
38   </header>
39   <!-- content area -->
40   <div class="content">
```

```

39
40
41 <!-- logout button-->
42 <a href="#" class="btn btn-sm text-white ms-4">
43 | | <i class="fa fa-sign-out me-1" aria-hidden="true"></i>Sign Out
44 </a>
45 <!--end of logout button-->
46 </div>
47 <!--end of container-->
48
49 </header>
50 <!--end of main page header-->
51
52 <!--start of transaction offcanvas-->
53 <div class="offcanvas offcanvas-start" tabindex="-1" id="offcanvasExample" aria-labelledby="offcanvasExampleLabel">
54 | <div class="offcanvas-header">
55 | | <h3 class="offcanvas-title text-white" id="offcanvasExampleLabel">Transact</h3>
56 | | <button type="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close"></button>
57 | </div>
58 | <!--offcanvas: transaction body-->
59 | <div class="offcanvas-body">
60 | | <small class="card-text text-white">
61 | | | Choose an Option below to perform a transaction
62 | | </small>
63 | | <!--transaction type dropdown list-->
64 | | <select names="transaction-type" class="form-control my-3" id="transaction-type">
65 | | | <option value="">-- Select Transaction Type --</option>
66 | | | <option value="payment">Payment</option>
67 | | | <option value="transfer">Transfer</option>
68 | | | <option value="deposit">Deposit</option>
69 | | | <option value="withdraw">Withdraw</option>
70 | | </select>
71 | | <!--end of transaction type dropdown list-->
72 | | <!--card: payment card-->
73 | | <div class="card payment-card">
74 | | | <!--card body-->
75 | | | <div class="card-body">
76 | | | | <!--form group-->
77 | | | | <div class="form-group mb-2">

```

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-
9ndCyUaIbzAi2FUVXJi0CjMCapSm07SnPjef0486qhlNuZ2cdeRh002iuK6FUUVM"
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">

```



```

        <!--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-
9ndCyUaIbzAi2FUVXJi0CjMCapSm07SnPJef0486qhLnuZ2cdeRh002iuK6FUUVM"
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>

```