

Capstone Project Documentation

Title:

Capstone Project - Bank Management System

Introduction:

The **Bank Management System** is a web-based application designed to provide users with secure and efficient tools for managing their financial activities. This system includes user authentication, account management, transaction processing, and an EMI calculator for loan planning. The application integrates a robust backend powered by **Spring Boot** and a responsive frontend built with **React.js**, ensuring a seamless user experience.

Objective:

The primary goal of this project is to provide users with an intuitive platform for managing their finances, with features such as account management, secure transactions, and loan planning. The system ensures data security, user privacy, and accurate financial operations.

Topics Covered:

- 1. Modules and Features**
 - Authentication Module
 - Account Management Module
 - Transaction Management Module
 - Loan Module (EMI Calculator)
- 2. User flow**
- 3. Backend Endpoints**
 - Overview of API endpoints and their functionality.
- 4. Frontend URLs**
 - Mapping of user-facing URLs to their respective components.
- 5. Database Schema**
 - Details of tables and relationships used in the application.

6. ER-Daigram
 7. **Tools and Technologies**
 - Spring Boot, React.js, MySQL, JWT, and more.
 8. Github (Link to code)
-

1. Modules and Features

1.1 Authentication Module

Handles user registration, login, and authentication. Uses **JWT** for secure session management.

- **Features:**
 - **User Registration:**
Creates a user account and stores hashed passwords in the database.
 - **User Login:**
Authenticates the user and issues a JWT token for subsequent requests.
 - **Database Changes:**
 - `User` table is updated with new records for registration.
-

1.2 Account Management Module

Enables users to create, view, and delete financial accounts.

- **Features:**
 - **Create Account:**
Adds a new account to the user's profile.
 - **View Accounts:**
Lists all accounts linked to the user.
 - **Delete Account:**
Removes an account from the database.
 - **Database Changes:**
 - `Account` table is updated with new or modified records.
-

1.3 Transaction Management Module

Allows users to perform deposits, withdrawals, and transfers between accounts.

- **Features:**

- **Deposit:** Adds funds to an account.
 - **Withdraw:** Deducts funds from an account.
 - **Transfer:** Moves funds between accounts.
 - **View Transactions:** Lists all transactions for a user or account.
 - **Database Changes:**
 - Updates the `Account` table for balances.
 - Adds entries to the `Transaction` table for transaction history.
-

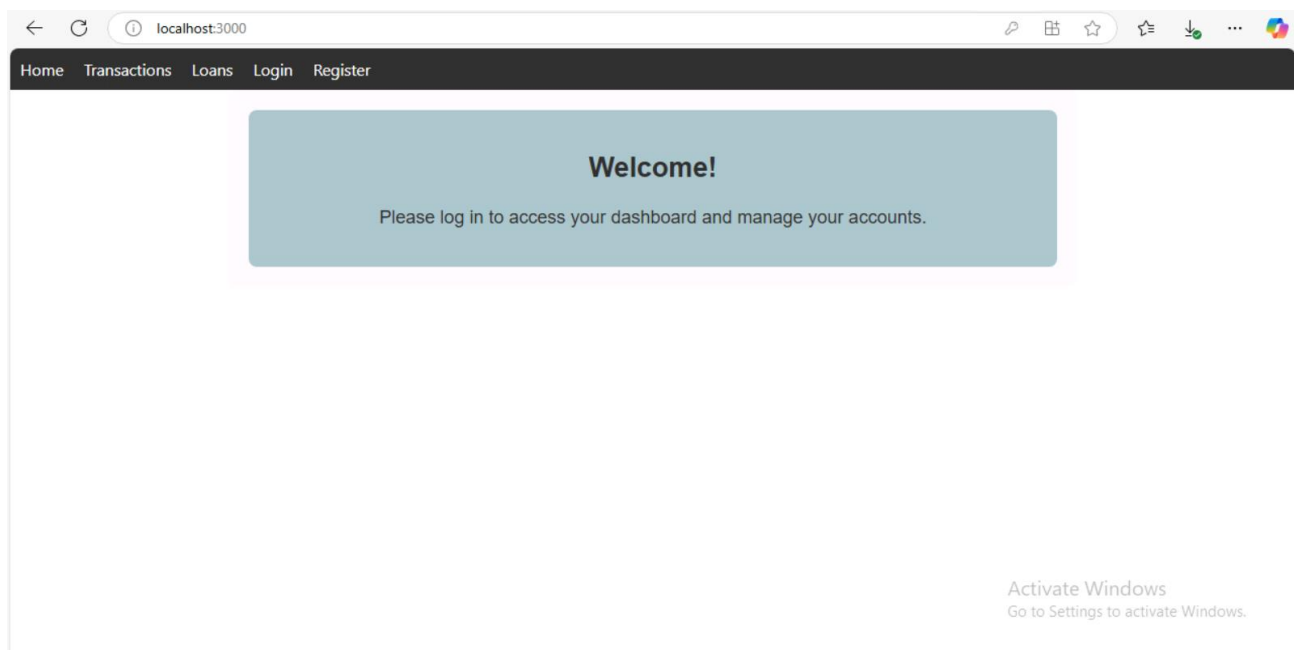
1.4 Loan Module (EMI Calculator)

Helps users calculate loan EMIs based on input parameters like loan amount, interest rate, and tenure.

- **Features:**
 - Calculates and displays monthly installment amounts using financial formulas.
 - Provides users with insights for better financial planning.
 - **Database Changes:**
 - No direct database impact; functionality is calculation-based.
-

2. User flow

Landing Page:



2.1 Authentication

- Register Flow

Register

Username
newuser

Email
new.user@gmail.com

Password
.....

Register

Activate Windows
Go to Settings to activate Windows.

Registration form UI.

Details:

Input: Name, Email, Password.

Process: Validation → Save user details in the database.

Query 1

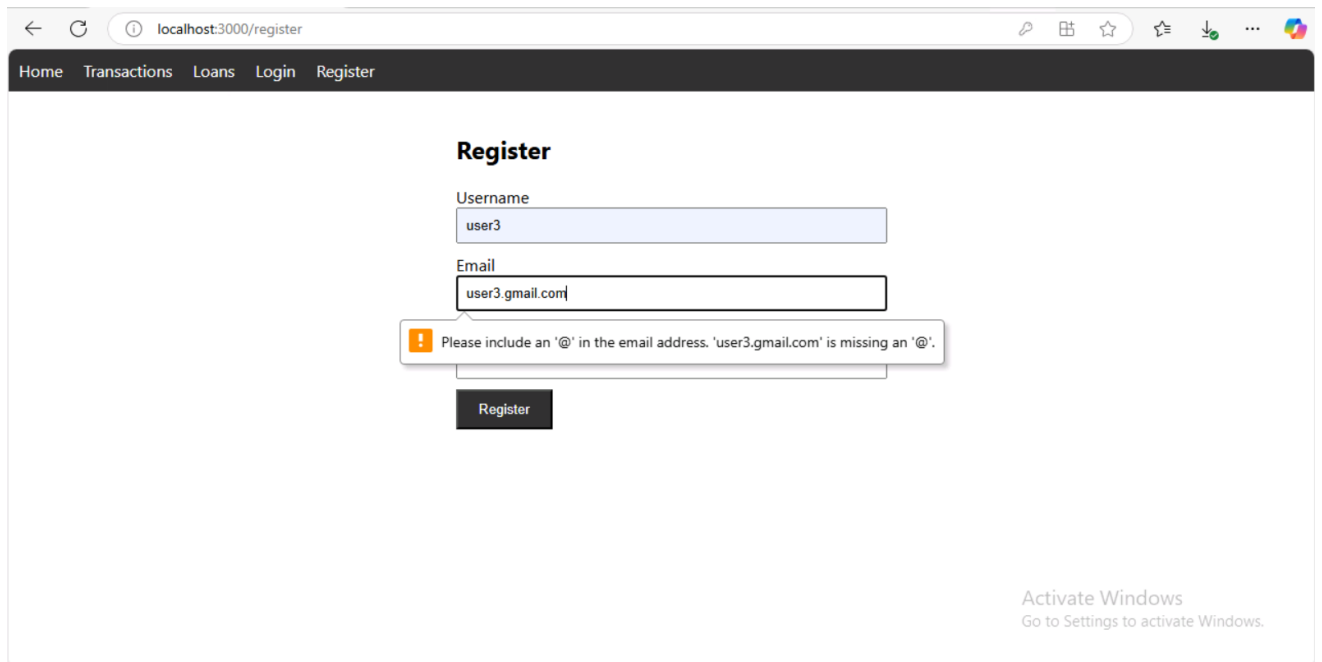
```
1 select * from users;
```

Result Grid

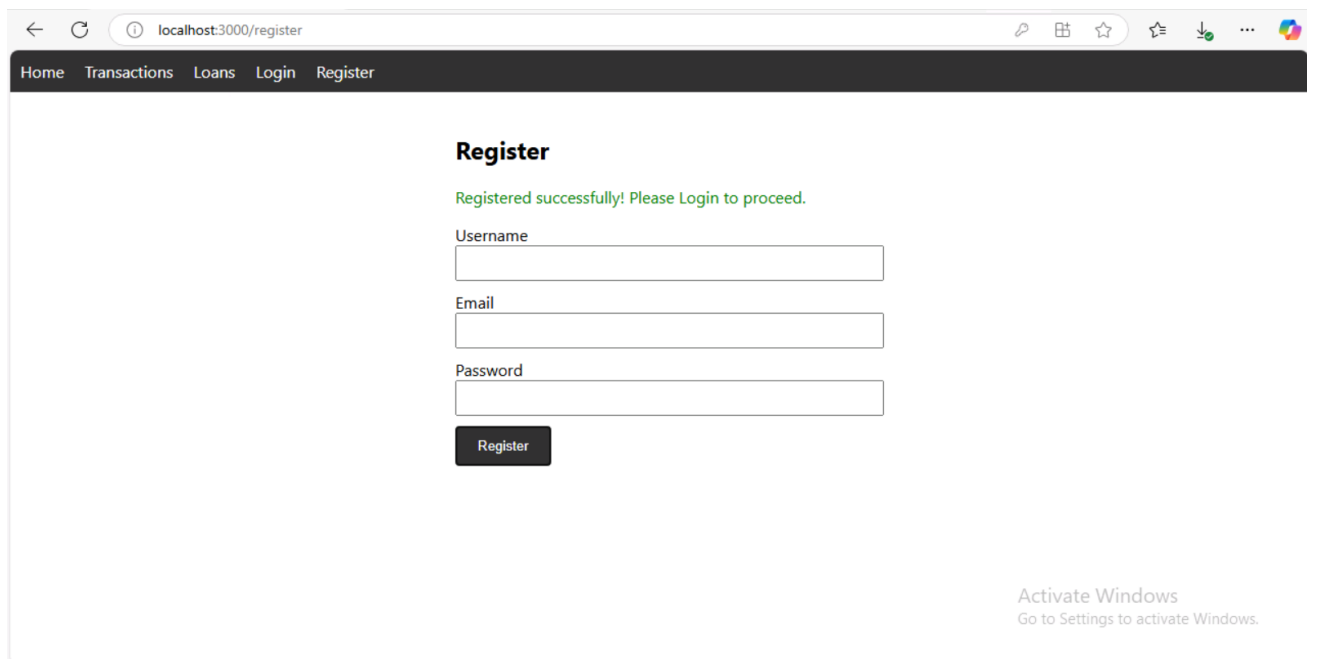
	id	email	password	username
1		silvya.patel@gmail.com	\$2a\$10\$u5N0gMzbGVesxJex1Jf7Ve9ksXdSwy/v...	Silvya
2		tirth.shrimali@gmail.com	\$2a\$10\$GeZ2U8mX/UG5Ycx51BpQp.2ma24xOV...	tirth
3		kittu@gmail.com	\$2a\$10\$Nx70nKESchblrP_Mkigmk_Ya4FoQoUET...	kittu
4		new.user@gmail.com	\$2a\$10\$1yV1vNOGVM8PvE6dUlwliO8TrATxmR....	newuser

User Table.

Output:



error message.



Success message and **redirect** to login.

- Login Flow

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The browser's navigation bar includes a back button, a refresh button, and several icons. Below the navigation bar is a dark header with the following links: Home, Transactions, Loans, Login, and Register. The main content area is white and features a 'Login' form. The form has a title 'Login' in bold. It contains two input fields: 'Email' with the value 'new.user@gmail.com' and 'Password' with masked characters '.....'. A 'Login' button is positioned below the password field.

Login form UI.

Details:

Input: Email, Password.

Process: Verify credentials → Generate JWT → Store in localStorage.

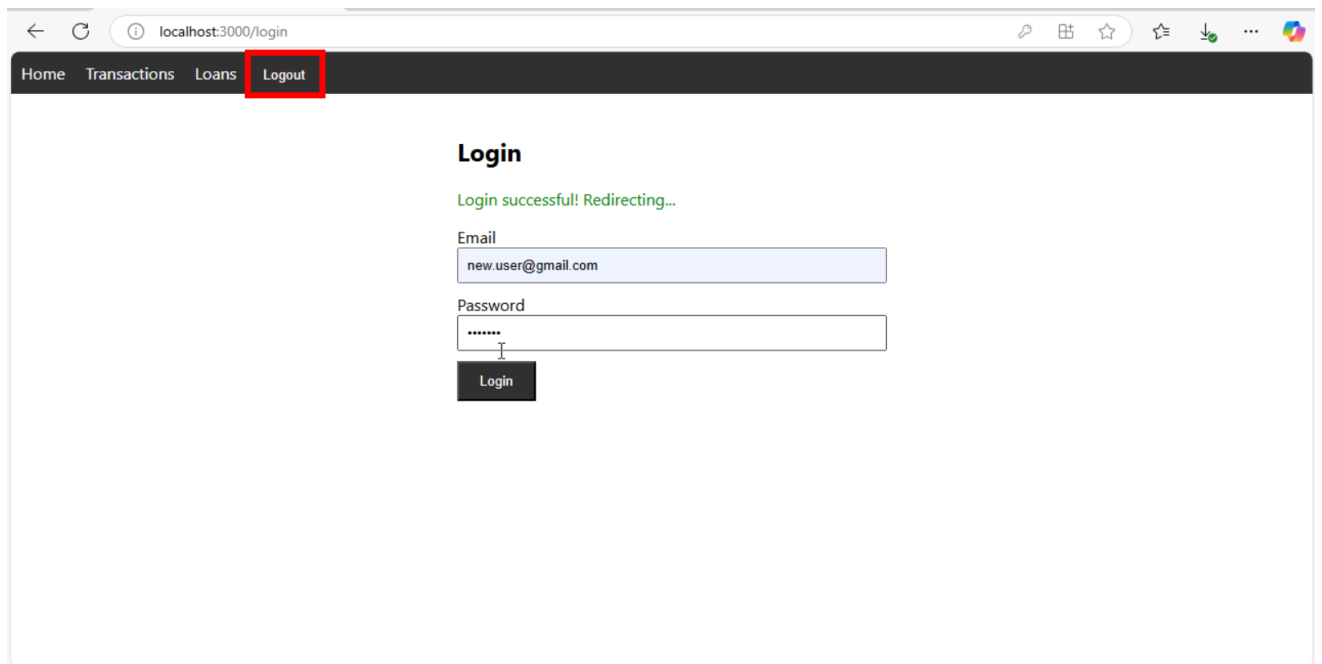
Output:

Upon error:

This screenshot shows the same login form as the previous one, but with an error message displayed above the input fields. The error message is in red text and reads: 'An error occurred while trying to log you in. Please check your credentials and try again.' The 'Email' field still contains 'new.u@gmail.com' and the 'Password' field is masked with '.....'. The 'Login' button remains at the bottom of the form.

error message.

Upon success: Login, Register buttons are not displayed anymore. Instead there is Logout now

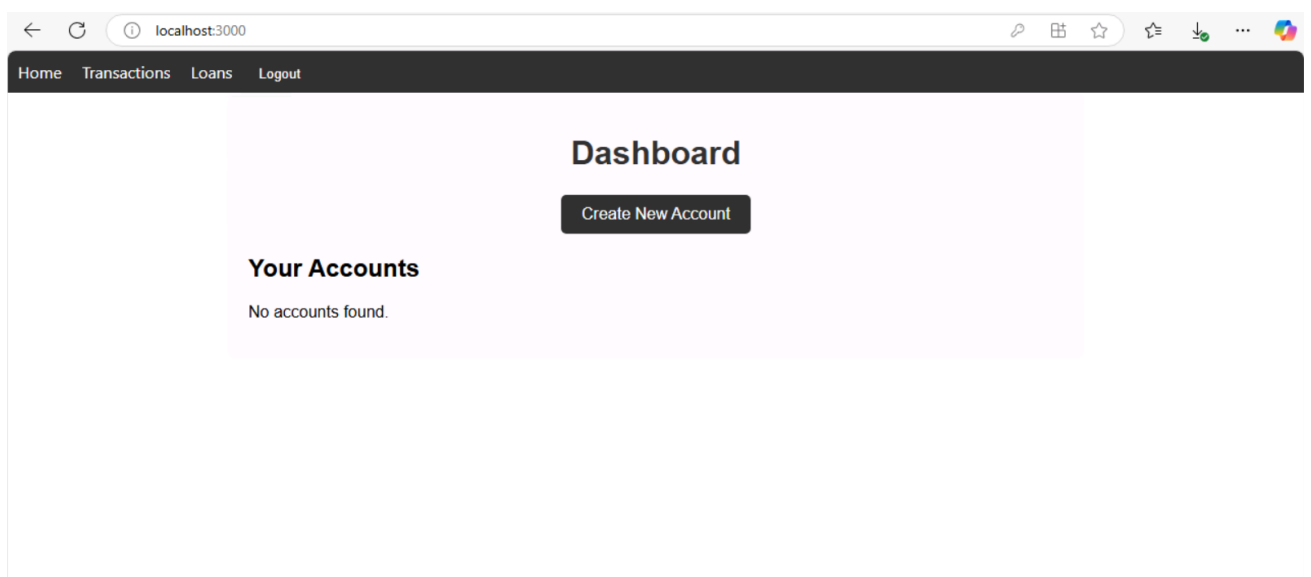


Redirect to the dashboard.

2.2 Account Management

The landing page changes to Dashboard according to user logged in

- View Accounts



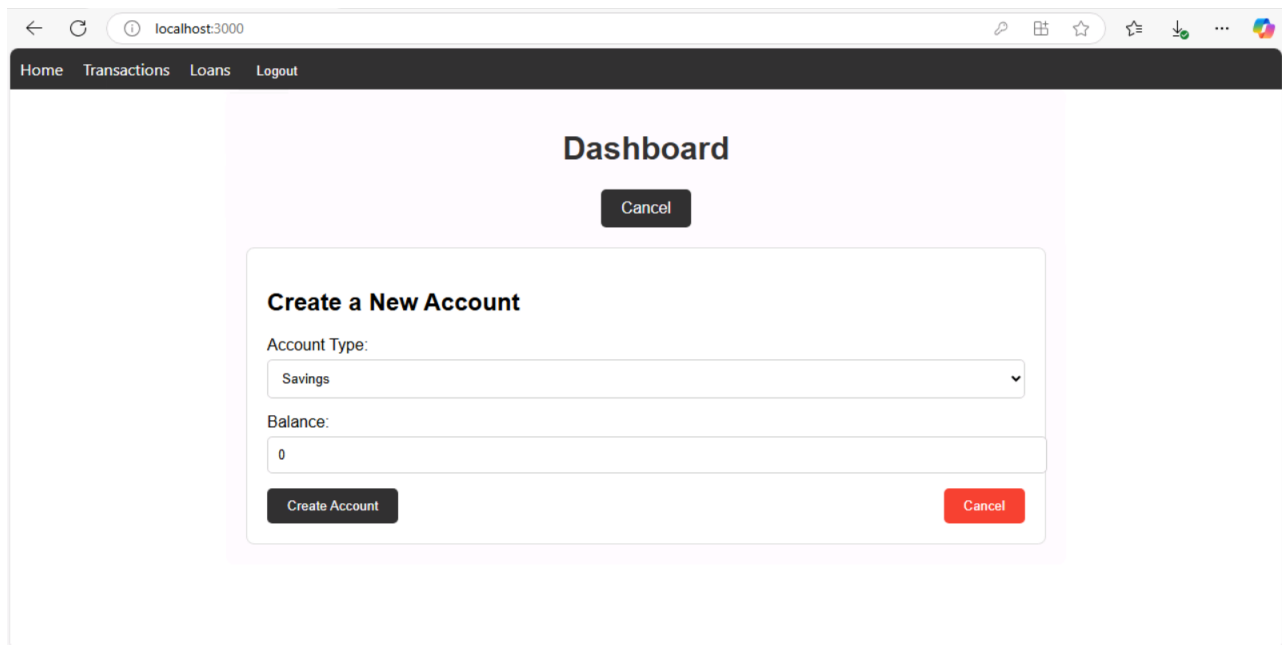
Accounts list table.

Details:

Backend fetches accounts associated with the authenticated user. Displays account number, type, and balance.

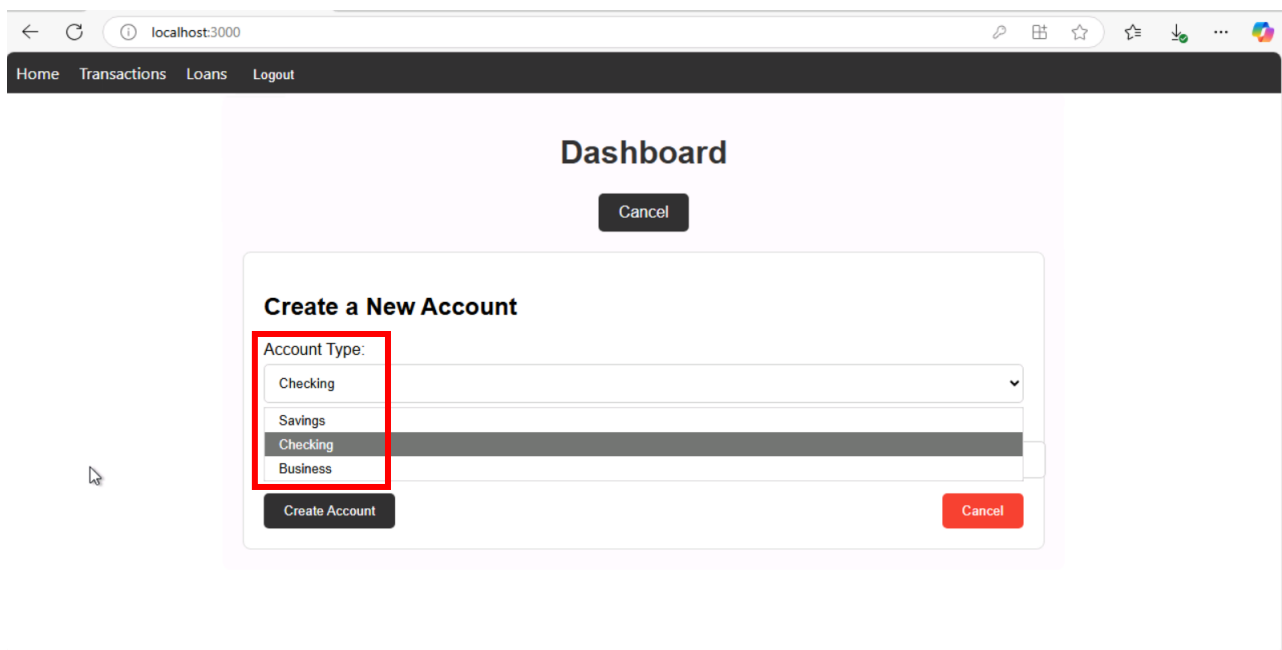
- Create Account:

Each user can create multiple accounts



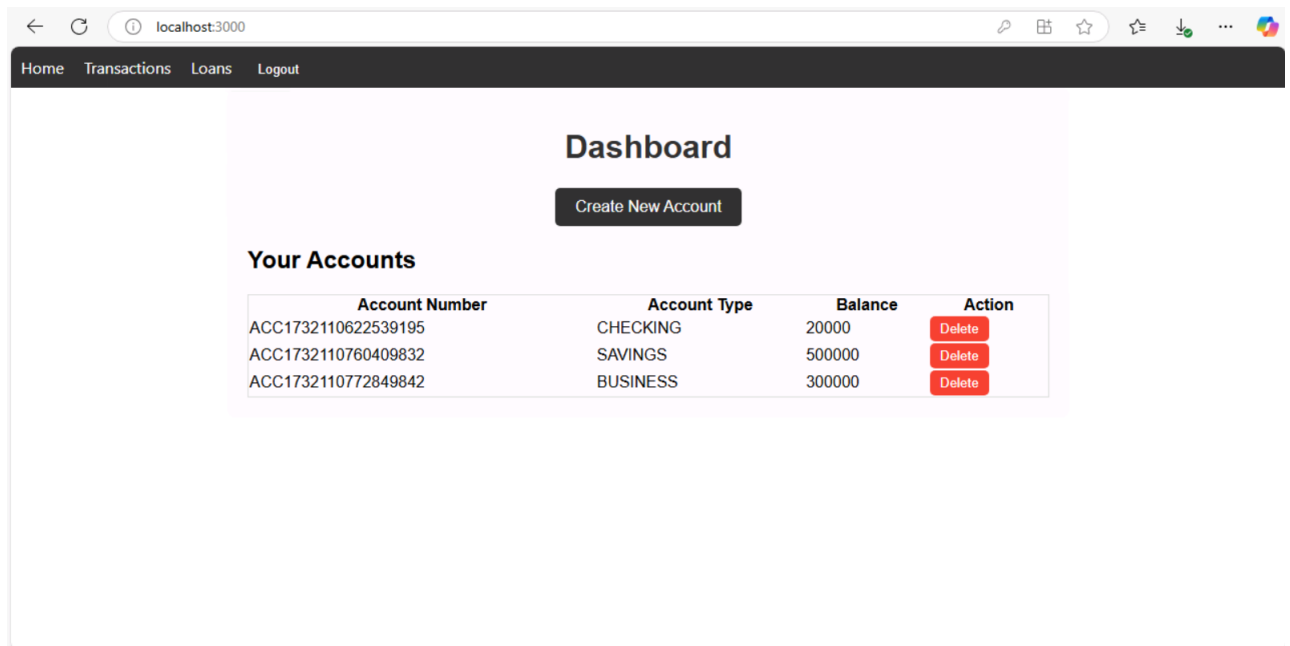
The screenshot shows a web browser at localhost:3000 with a navigation bar containing 'Home', 'Transactions', 'Loans', and 'Logout'. The main content area is titled 'Dashboard' and contains a 'Create a New Account' form. The form has two input fields: 'Account Type' with a dropdown menu currently showing 'Savings', and 'Balance' with a text input showing '0'. Below these fields are two buttons: 'Create Account' (dark grey) and 'Cancel' (red). A 'Cancel' button is also located above the form.

Account creation form.



This screenshot shows the same 'Create a New Account' form, but the 'Account Type' dropdown menu is open, displaying a list of options: 'Checking', 'Savings', 'Checking', and 'Business'. The first 'Checking' option is highlighted with a red box. The 'Create Account' and 'Cancel' buttons remain visible below the dropdown.

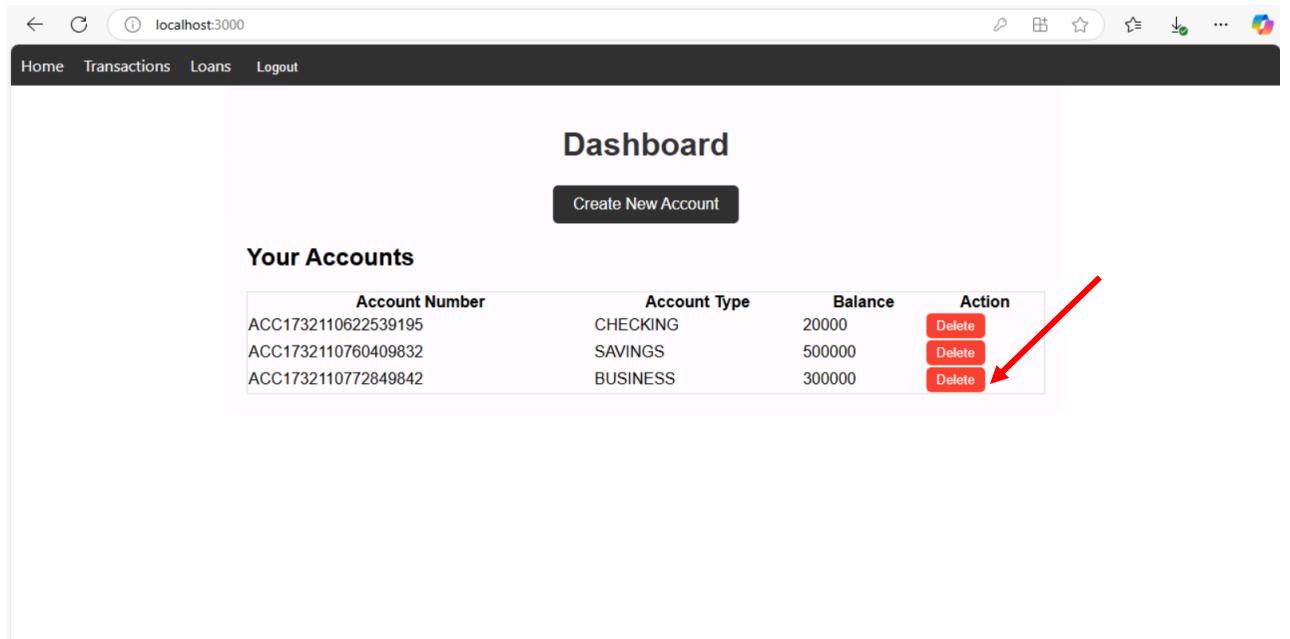
Account creation form.



Updated accounts table.

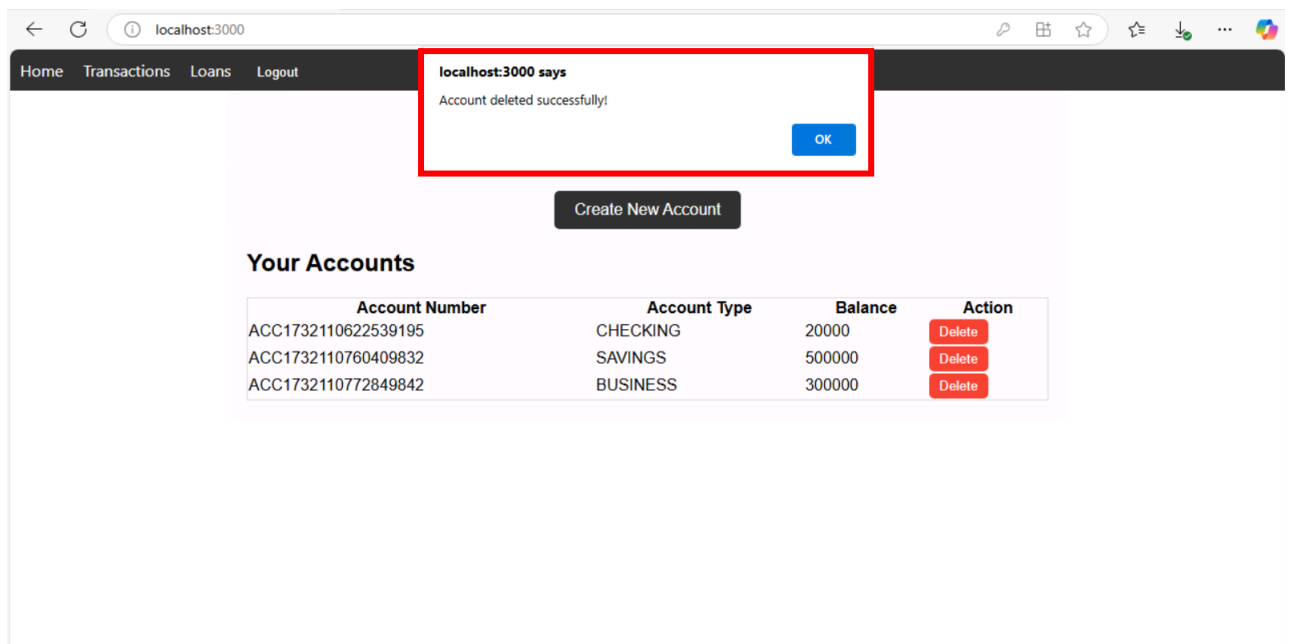
- Delete Account

Input:

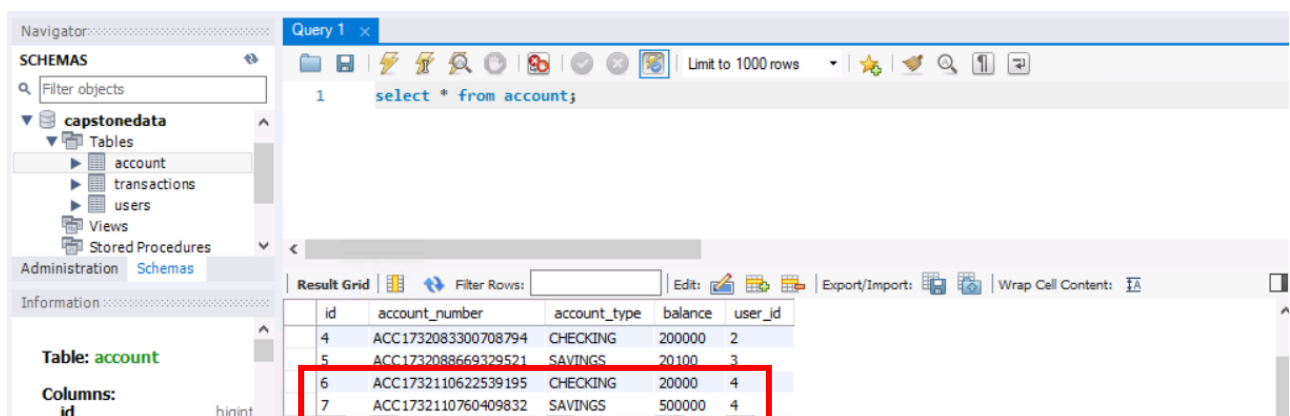


Click Delete.

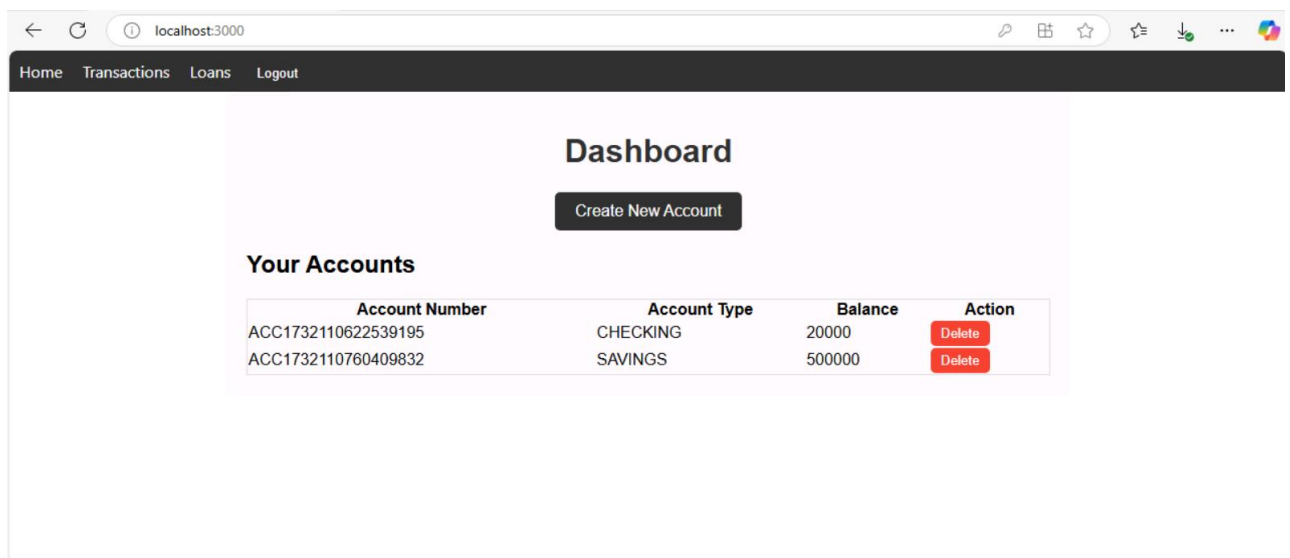
Process:



Alert.



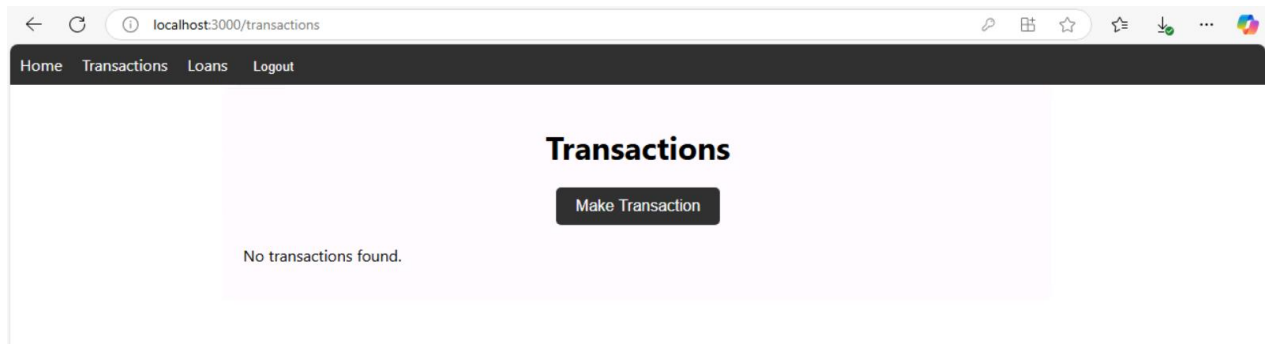
Output:



New list.

2.3 Transaction Management

Landing Page



- Transfer Funds

Details:

Input:

Deposit: Account, amount

Withdraw: Account, Email(for checking if it is someone else's account), amount

Transfer: Source account, destination account, amount.

A screenshot of a web browser at `localhost:3000/transactions`. The browser's address bar shows the URL. The page has a dark navigation bar with links: Home, Transactions, Loans, and Logout. The main content area features a light purple background with a white card titled "Make a Transaction". Inside the card, the "Transaction Type" dropdown menu is open, showing options: "Select Type" (highlighted), "Deposit", "Withdraw", and "Transfer". The dropdown menu is outlined with a red border.

Transaction form.

A screenshot of the same web browser at `localhost:3000/transactions`. The "Make a Transaction" form is now fully visible. The "Transaction Type" dropdown menu is closed, and "Deposit" is selected. Below the dropdown are two input fields: "Account Number" and "Amount". At the bottom of the form are two buttons: "Submit" (dark grey) and "Cancel" (red).

Deposit funds form.

A screenshot of a web browser at localhost:3000/transactions. The page has a dark navigation bar with links: Home, Transactions, Loans, and Logout. The main content area features a light purple background with a white form titled "Make a Transaction". The form includes a "Transaction Type" dropdown menu set to "Withdraw", an "Account Number" input field, an "Email" input field, and an "Amount" input field. At the bottom of the form are two buttons: a dark grey "Submit" button and a red "Cancel" button.

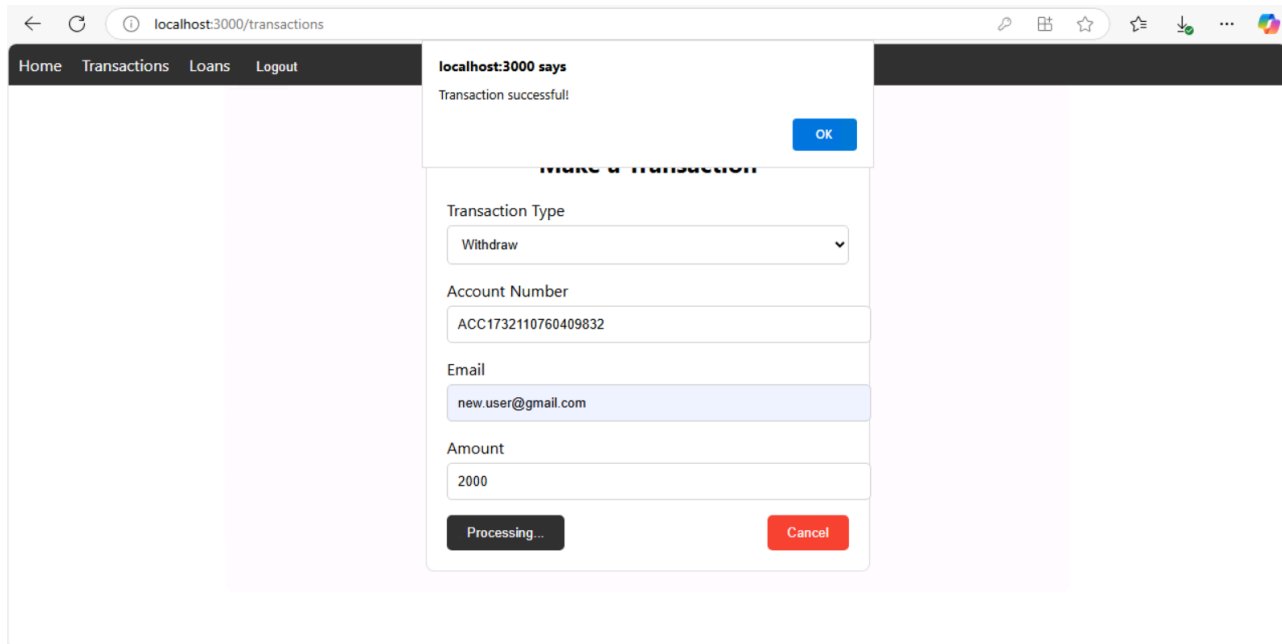
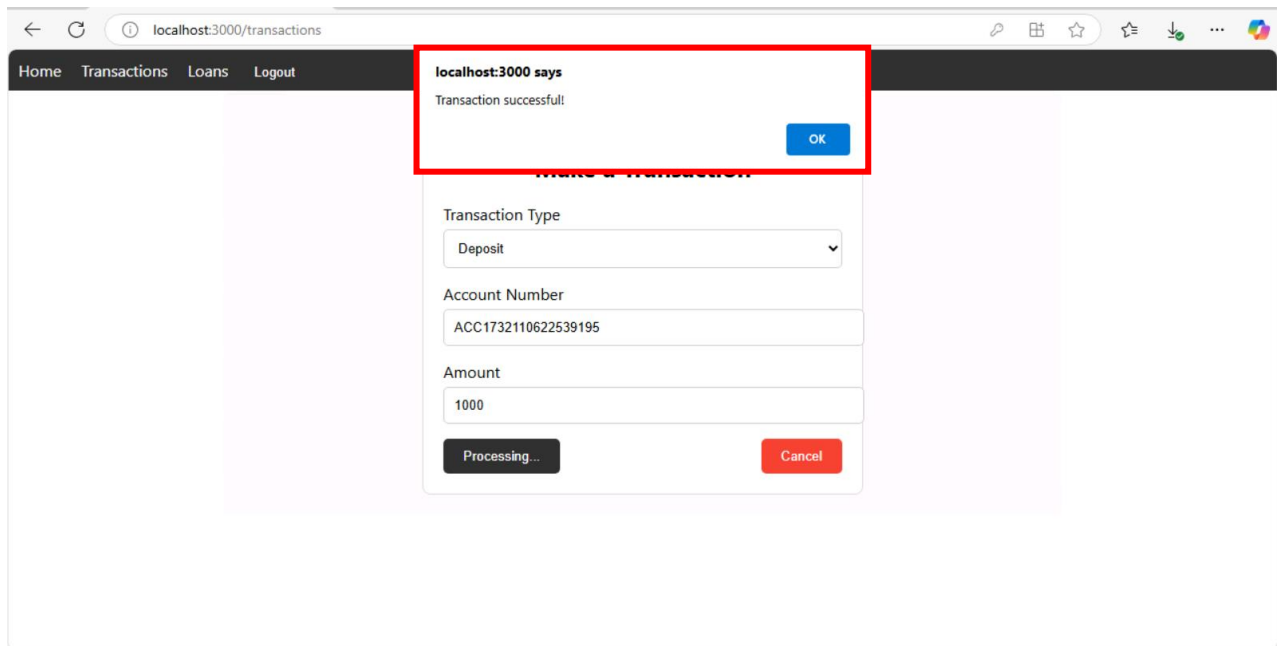
Withdraw funds form.

A screenshot of a web browser at localhost:3000/transactions. The page has a dark navigation bar with links: Home, Transactions, Loans, and Logout. The main content area features a light purple background with a white form titled "Make a Transaction". The form includes a "Transaction Type" dropdown menu set to "Transfer", an "Account Number" input field, a "Target Account Number" input field, and an "Amount" input field. At the bottom of the form are two buttons: a dark grey "Submit" button and a red "Cancel" button.

Transfer funds form.

Process: Validate accounts and balance → Save transaction in the database.

Output: Updated balances and transaction list.



After deposit and Withdrawal.

←↻🔍📄🌟🔖📥⋮🌈

HomeTransactionsLoansLogout

Dashboard

Create New Account

Your Accounts

Account Number	Account Type	Balance	Action
ACC1732110622539195	CHECKING	21000	Delete
ACC1732110760409832	SAVINGS	498000	Delete

Updated balance reflects on dashboard

Navigator:.....

SCHEMAS

Filter objects

capstondata

Tables

account

transactions

users

Views

Stored Procedures

AdministrationSchemas

Information:.....

Table: account

Columns: id

Query 1 x

Limit to 1000 rows

1 select * from transactions;

Result Grid

Filter Rows:

idamounttransaction_datetype

source_account_idtarget_account_id

7102024-11-20 06:05:43.320702TRANSFER31

81002024-11-20 07:44:55.460141DEPOSIT5NULL

910002024-11-20 14:05:25.664199DEPOSIT6NULL

1020002024-11-20 14:06:17.140682WITHDRAWAL7NULL

After Transfer.

←↻🔍📄🌟🔖📥⋮🌈

HomeTransactionsLoansLogout

localhost:3000 says

Transaction successful!

OK

MAKE A TRANSACTION

Transaction Type

Transfer

Account Number

ACC1732110760409832

Target Account Number

ACC1732073069441542

Amount

20000

Processing...

Cancel

localhost:3000

Home Transactions Loans Logout

Dashboard

Create New Account

Updated balance after transfer

Your Accounts

Account Number	Account Type	Balance	Action
ACC1732110622539195	CHECKING	21000	Delete
ACC1732110760409832	SAVINGS	478000	Delete

Navigator

SCHEMAS

Filter objects

capstonedata

Tables

- account
- transactions
- users

Views

Stored Procedures

Administration Schemas

Information

Table: account

Columns: id

Query 1

```
select * from transactions
```

Limit to 1000 rows

Result Grid

	id	amount	transaction_date	type	source_account_id	target_account_id
8	100	2024-11-20 07:44:55.460141	DEPOSIT	5	NULL	
9	1000	2024-11-20 14:05:25.664199	DEPOSIT	6	NULL	
10	2000	2024-11-20 14:06:17.140683	WITHDRAWAL	7	NULL	
11	20000	2024-11-20 14:16:14.979630	TRANSFER	7	1	

- View Transactions

localhost:3000/transactions

Home Transactions Loans Logout

Transactions

Make Transaction

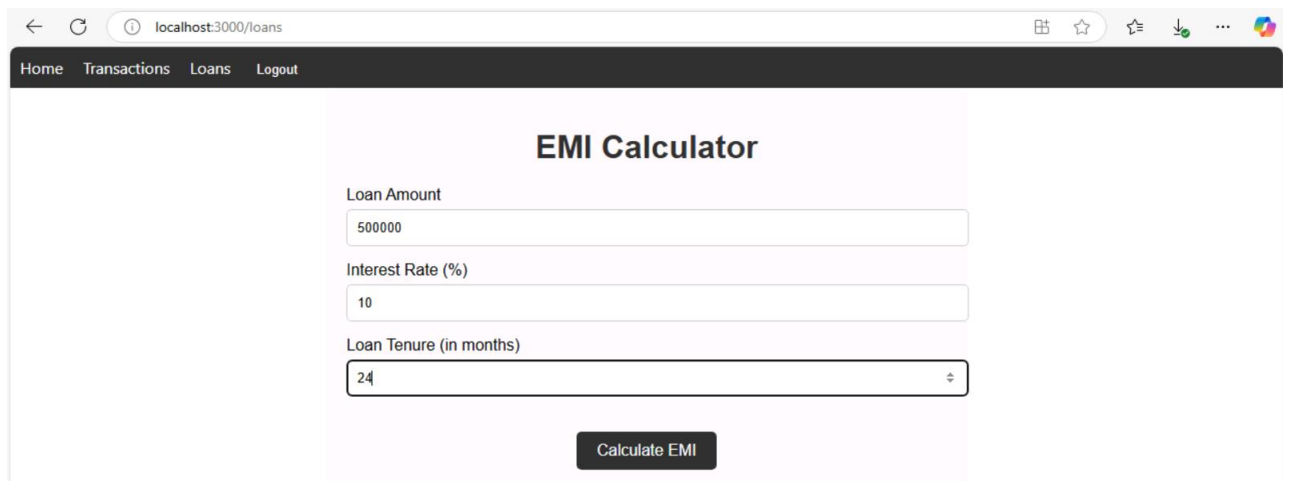
Id	Type	Amount	Date
9	DEPOSIT	1000	11/20/2024, 2:05:25 PM
10	WITHDRAWAL	2000	11/20/2024, 2:06:17 PM
11	TRANSFER	20000	11/20/2024, 2:16:14 PM

Transactions history table.

Details: Backend fetches transaction records. Displays details such as date, source, destination, and amount.

2.4 Loan Planning (EMI Calculator)

- Calculate EMI



The screenshot shows a web browser at localhost:3000/loans. The navigation bar includes Home, Transactions, Loans, and Logout. The main content area is titled "EMI Calculator" and contains three input fields: "Loan Amount" with the value 500000, "Interest Rate (%)" with the value 10, and "Loan Tenure (in months)" with the value 24. Below these fields is a dark button labeled "Calculate EMI".

EMI calculator UI.

Details:

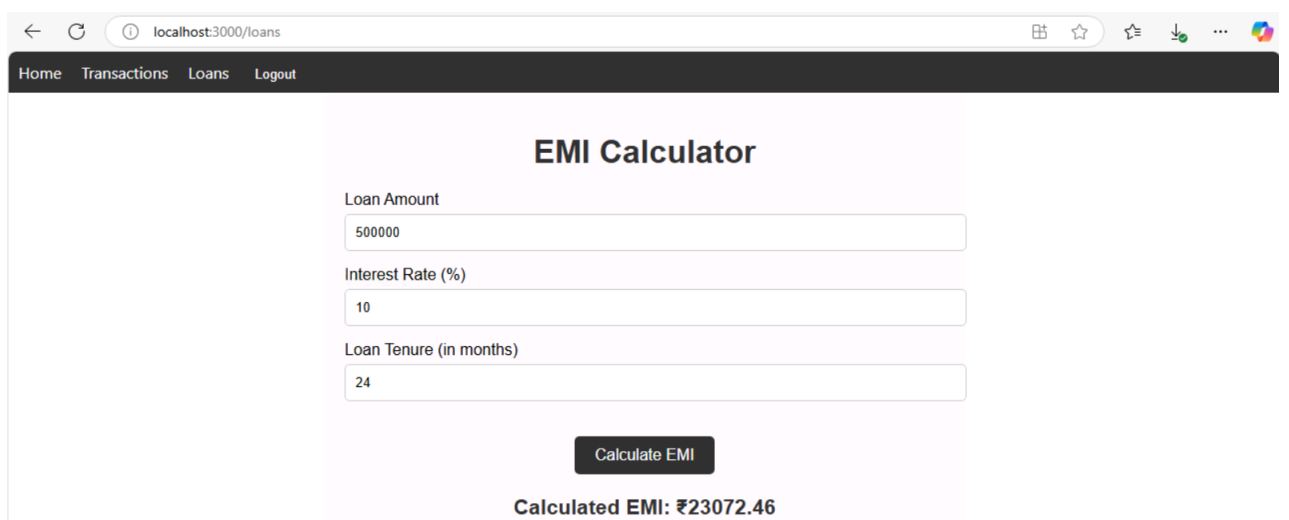
Input: Loan amount, interest rate, tenure.

Process: Calculate EMI using the formula.

$$EMI = \frac{P \cdot R \cdot (1 + R)^N}{(1 + R)^N - 1}$$

(where P is principal, R is monthly interest, and N is tenure in months).

Output:



This screenshot shows the same EMI Calculator UI as the previous one, but with the calculated result displayed. Below the "Calculate EMI" button, the text "Calculated EMI: ₹23072.46" is shown.

Display monthly EMI amount.

3. Backend Endpoints

Endpoint	HTTP Method	Description
/auth/register	POST	Registers a new user.
/auth/login	POST	Authenticates a user and issues a JWT.
/accounts	POST	Creates a new account.
/accounts/user	GET	Fetches accounts linked to the authenticated user.
/accounts/{id}	DELETE	Deletes an account by ID.
/transactions/deposit	POST	Processes a deposit transaction.
/transactions/withdraw	POST	Processes a withdrawal transaction.
/transactions/transfer	POST	Processes a transfer transaction.
/transactions/user	GET	Fetches transaction history for a user.

4. Frontend URLs

URL	Description
/login	User login page.
/register	User registration page.
/dashboard	User dashboard showing account details.
/transactions	Displays transaction history and options.
/loans	EMI Calculator for loan planning.

5. Database Schema

Database Schema Documentation

5.1. Table: users

Column Name	Data Type	Attributes	Description
id	bigint	AUTO_INCREMENT, PK	Primary key, unique identifier for users.
email	varchar(255)	UNIQUE	Email address of the user (used for authentication).
password	varchar(255)	NOT NULL	Encrypted password for the user.
username	varchar(255)	UNIQUE	Unique username for the user.

Primary Key: Id

5.2. Table: `account`

Column Name	Data Type	Attributes	Description
<code>id</code>	<code>bigint</code>	<code>AUTO_INCREMENT, PK</code>	Primary key, unique identifier for accounts.
<code>account_number</code>	<code>varchar(255)</code>	<code>UNIQUE</code>	Unique identifier for each account.
<code>account_type</code>	<code>enum('BUSINESS', 'CHECKING', 'SAVINGS')</code>	<code>NOT NULL</code>	Type of account (business, checking, or savings).
<code>balance</code>	<code>double</code>	<code>NOT NULL</code>	Current balance in the account.
<code>user_id</code>	<code>bigint</code>	<code>FK</code>	Foreign key referencing <code>users(id)</code> .

Primary Key: `id`

Foreign Key: `user_id` \rightarrow `users(id)`

Relationships:

- **One-to-Many:** A user can have multiple accounts.

5.3. Table: `transactions`

Column Name	Data Type	Attributes	Description
<code>id</code>	<code>bigint</code>	<code>AUTO_INCREMENT, PK</code>	Primary key, unique identifier for transactions.
<code>amount</code>	<code>double</code>	<code>NOT NULL</code>	Amount involved in the transaction.
<code>transaction_date</code>	<code>datetime(6)</code>	<code>NOT NULL</code>	Date and time when the transaction occurred.
<code>type</code>	<code>varchar(255)</code>	<code>NOT NULL</code>	Type of transaction (e.g., "deposit," "withdrawal," "transfer").
<code>source_account_id</code>	<code>bigint</code>	<code>FK</code>	Foreign key referencing <code>account(id)</code> for the source account.
<code>target_account_id</code>	<code>bigint</code>	<code>FK</code>	Foreign key referencing <code>account(id)</code> for the target account (if applicable).

Primary Key: id

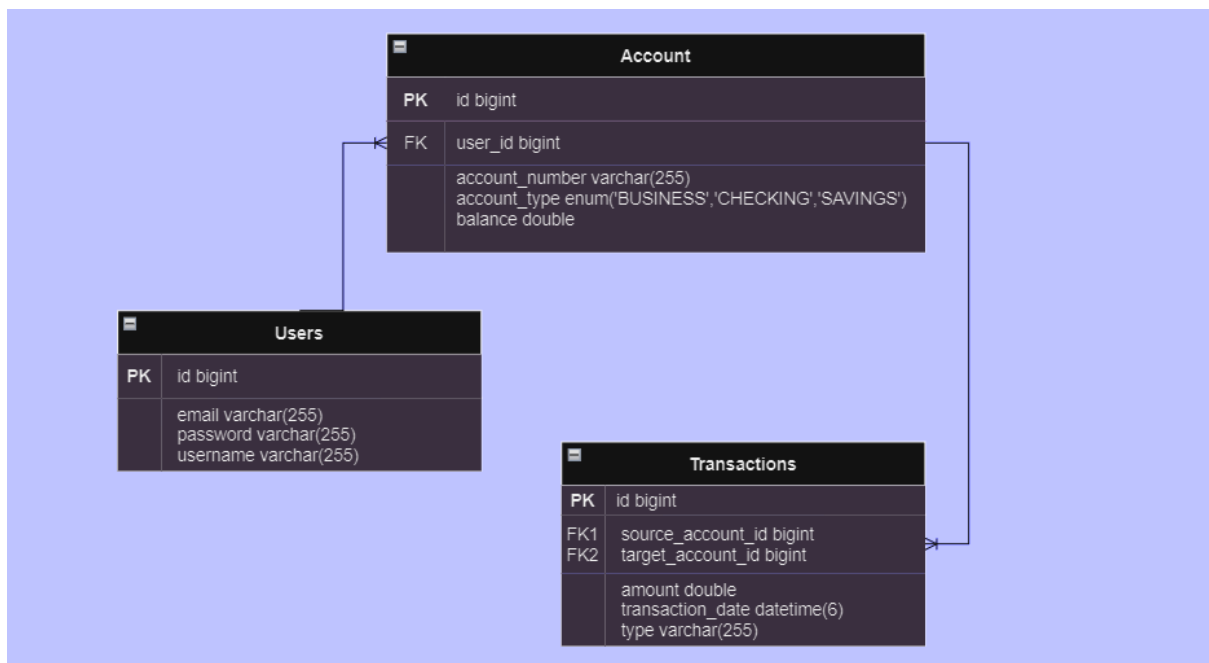
Foreign Keys:

- source_account_id → account(id)
- target_account_id → account(id)

Relationships:

- **One-to-Many:** An account can have multiple transactions.
 - **Self-Referencing:** Source and target accounts can belong to the same or different users.
-

6. ER-diagram



7. Tools and Technologies

- **Backend:** Spring Boot (Java), JWT for authentication.
 - **Frontend:** React.js for responsive UI.
 - **Database:** MySQL for persistent data storage.
 - **Security:** BCrypt for password hashing, JWT for secure token management.
-

8. Github (Link to code): [silvyapatel/Capstone](https://github.com/silvyapatel/Capstone)

