Banking Website Project

Overview The Banking Website project allows users to perform financial transactions securely. Users can:
- Credit money to their wallet
- Debit money from their wallet
- Send money to other users
- View transaction history
- Apply for a loan
- Admins can approve or reject loan applications

The system follows the Model-View-Controller (MVC) architecture using ASP.NET Core MVC and Entity framework

Bank Balance Module

Purpose: Allows users to check their current bank balance.

Controller: BalanceController
- GetBalance()

Model: BankAccounts Entity
- AccountId (PK)
- UserId
- Balance

Credit & Debit Module

Purpose: Allows users to add or withdraw money from their account.

Controller: BankController
- CreditAmount()
- DebitAmount()

Model: BankAccounts Entity
- AccountId (PK)
- UserId
- Balance

Send Money to Users Module

Purpose: Allows users to transfer money to another user.

Controller: TransactionController
- SendMoney()

Model: Transaction Entity
- TransactionId (PK)
- SenderUsername

- ReceiverUsername
- Amount
- TransactionType (Send Money)
- Date

Apply Loan Module

Purpose: Allows users to apply for a loan.

Controller: LoanController

- ApplyLoan()

Model: Loan Entity

- LoanId (PK)
- UserId
- LoanAmount
- Status (Pending, Approved, Rejected)

Loan Approval Module

Purpose: Allows admins to approve or reject loan applications.

Controller: AdminLoanController

- ApproveLoan()
- RejectLoan()

Model: Loan Entity

- LoanId (PK)
- UserId
- LoanAmount
- Status

Database Schema

```
CREATE TABLE BankAccounts (
    AccountId INT PRIMARY KEY,
    UserId INT,
    Balance DECIMAL(10,2)
);

CREATE TABLE Transactions (
    TransactionId INT PRIMARY KEY,
    SenderUsername VARCHAR(50),
    ReceiverUsername VARCHAR(50),
    Amount DECIMAL(10,2),
    TransactionType VARCHAR(20),
    Date DATETIME
);

CREATE TABLE Loans (
    LoanId INT PRIMARY KEY,
    UserId INT,
    LoanAmount DECIMAL(10,2),
    Status VARCHAR(10)
);
```

Implementation Steps

1. Create an ASP.NET Core MVC project in Visual Studio 2022.
2. Install Entity Framework Core and SQL Server provider.
3. Define models for BankAccounts, Transactions, and Loans.
4. Create controllers for balance check, credit, debit, send money, and loans.
5. Use Razor Views and Bootstrap for UI design.
6. Apply Code-First migrations and update the database.
7. Implement authentication for user security.
8. Deploy on IIS or Azure.