CS6106: DATABASE MANAGEMENT SYSTEM

TEAM NO: 22

ATM Database management system

Team Members:

Yuva Priya. D- 2022103575

Jothi Sri. S- 2022103049

ATM SYSTEM - PROBLEM STATEMENT

The project entitled ATM system has a drastic change to that of the older version of the banking system, customers feel inconvenienced with the transaction method as it was in the hands of the bank employees.

In our ATM system, the above problem is overcome here, the transactions are done in person by the customer thus making the customers feel safe and secure.

By using our ATM Machine one can

- -change their pin only if he/she wants to can the pin and also knowing the old pin.
- -check the balance
- -withdraw money by entering the amount
- -withdraw money using fash cash feature
- -deposit money if it's a debit card
- -see their transaction details
- -transfer fund from one account to another

Thus, the application of our system helps the customer check the balance and transaction of the amount by validating the PIN therefore ATM system is more user-friendly.

ENTITIES

ATM_MACHINE

ATM_Id	varchar(10)	**PRIMARY KEY**
ATM_Name	varchar(10)	
ATM_Branch	varchar(10)	
ATM_Amount	decimal(10,2)	
ATM_Address	varchar(80)	

Bank

Bank_Id	varchar(10)	**PRIMARY KEY**
IFSC_Code	varchar(15)	
Bank_Name	varchar(10)	
Bank_Id	varchar(10)	

Branch

Branch_Id	varchar(10)	**PRIMARY KEY**
Branch_Name	varchar(15)	
Branch_location	varchar(15)	

ACCOUNT

Acc_No	bigint(20)	**PRIMARY KEY**
Acc_Type	varchar(10)	
Savings	bigint(20)	
Current	bigint(20))	

Transaction

Transaction_Id	varchar(10)	**PRIMARY KEY**
Transaction_type	varchar(10)	
Transaction_Date	date	
Transaction_Status	varchar(10)	
Transaction_Amount	decimal(10,2)	
Transaction_Time	time	

Card

Card_No	bigint(20)	**PRIMARY KEY**
Card_CVV	bigint(20)	
Card_Type	varchar(10)	
Card_Balance	decimal(16,2)	
Card_ExpiryDate	date	

Customer

C_Id	varchar(10)	**PRIMARY KEY**
First_Name	varchar(20)	
Mid_Name	varchar(20)	
Last_Name	varchar(20)	
C_Address	varchar(80)	
C_Phone	bigint(20)	

Transaction_participants

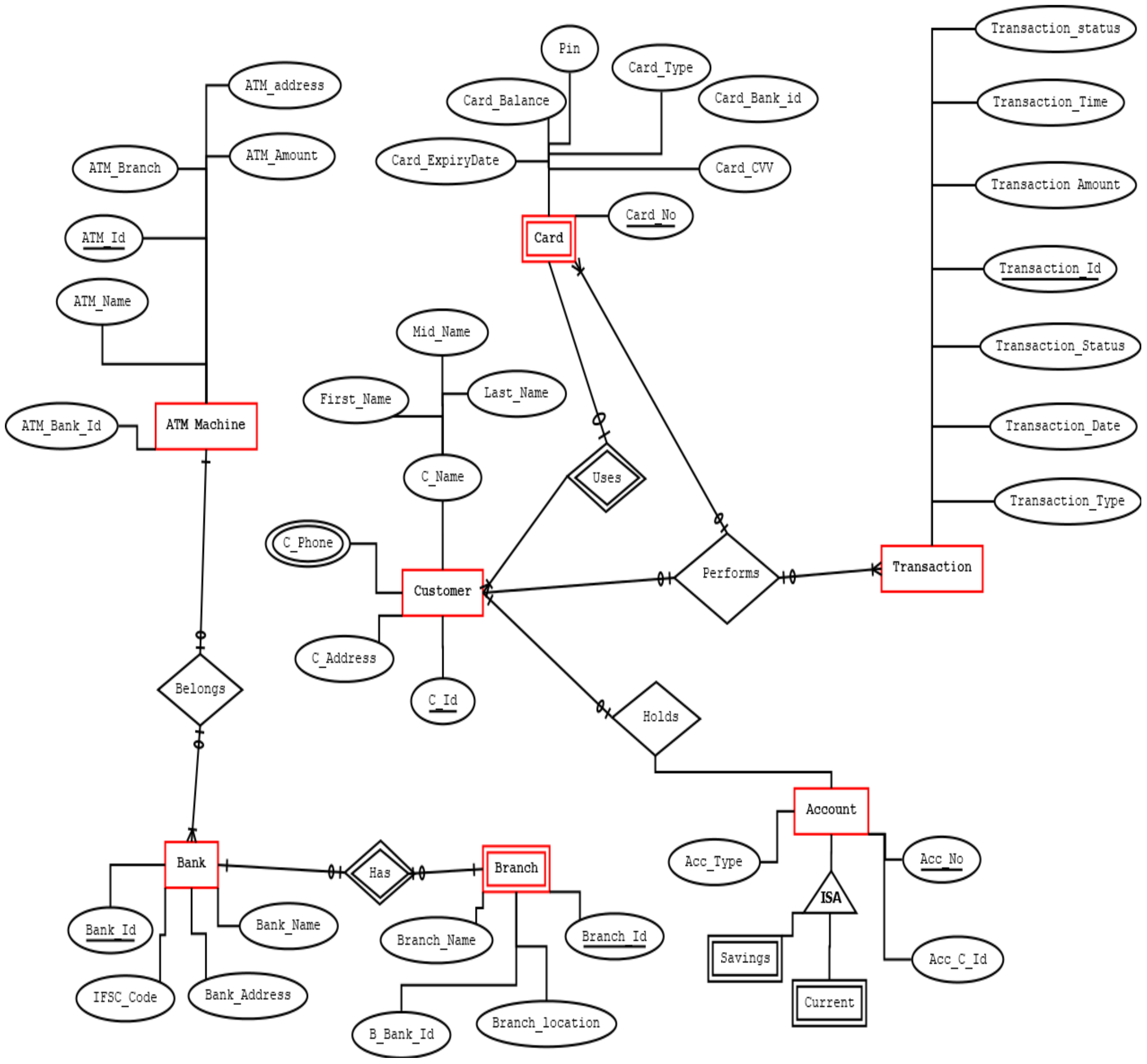
Transaction_Id	varchar(10)
Participant_C_Id	varchar(10)
Participant_Role	varchar(10)

Customer_Card

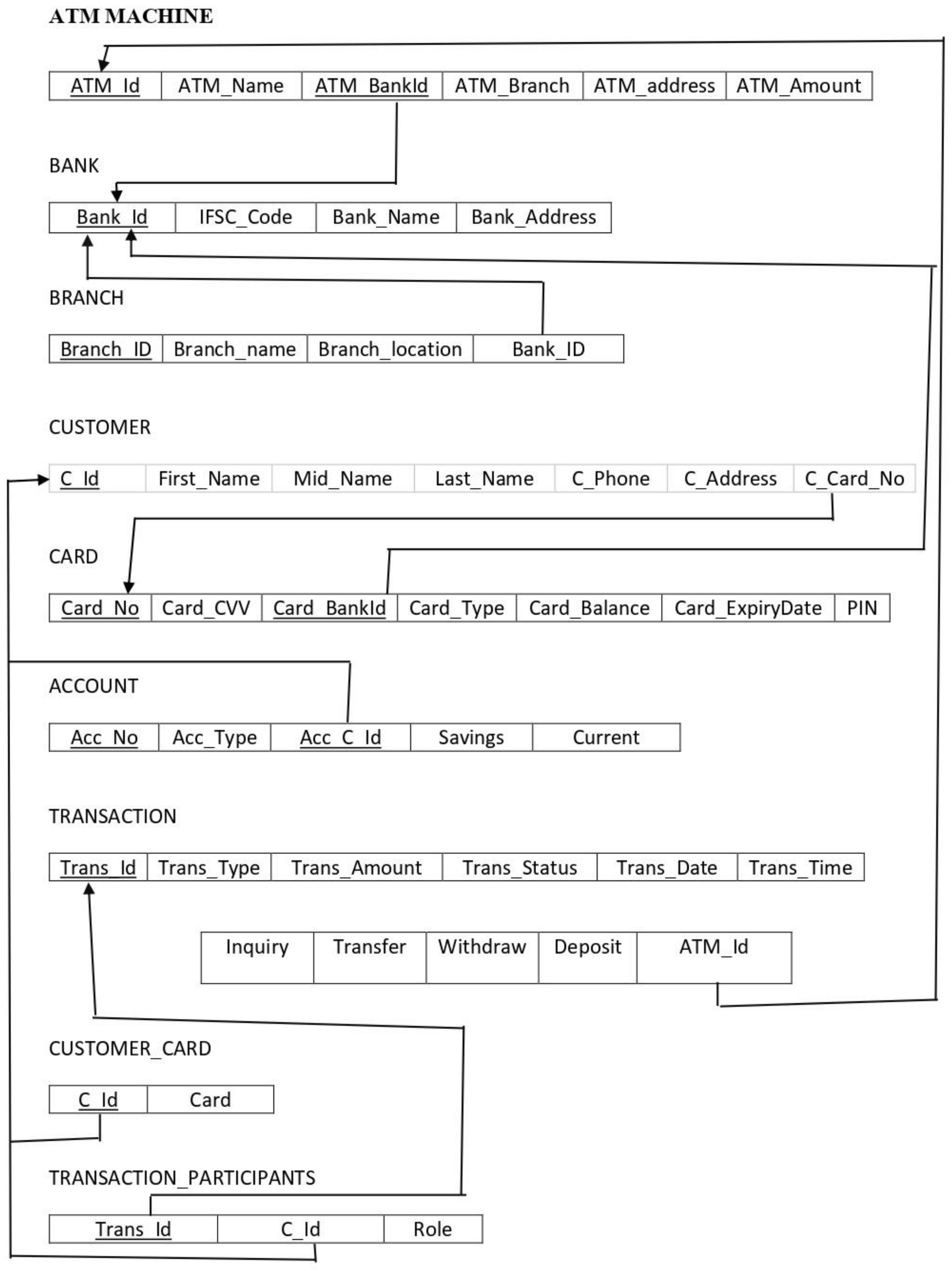
C_Id	varchar(10)
Card_no	Int

PRIMARY KEY (C_Id, Card_No)

ER DIAGRAM


Transaction_status
Transaction_Time
Transaction Amount
Transaction_Id
Transaction_Status
Transaction_Date
Transaction_Type
Transaction
Pin
Card_Type
Card_Balance
Card_Bank_id
Card_ExpiryDate
Card_CVV
Card_No
Card
ATM_address
ATM_Amount
ATM_Branch
ATM_Id
ATM_Name
ATM_Bank_Id
ATM Machine
Mid_Name
First_Name
Last_Name
C_Name
C_Phone
Customer
C_Address
C_Id
Uses
Performs
Holds
Belongs
Account
Acc_Type
Acc_No
ISA
Savings
Current
Acc_C_Id
Bank
Has
Branch
Bank_Id
Bank_Name
IFSC_Code
Bank_Address
Branch_Name
Branch_Id
B_Bank_Id
Branch_location

RELATIONAL SCHEMA



ATM MACHINE

ATM_Id	ATM_Name	ATM_BankId	ATM_Branch	ATM_address	ATM_Amount

BANK

Bank_Id	IFSC_Code	Bank_Name	Bank_Address

BRANCH

Branch_ID	Branch_name	Branch_location	Bank_ID

CUSTOMER

C_Id	First_Name	Mid_Name	Last_Name	C_Phone	C_Address	C_Card_No

CARD

Card_No	Card_CVV	Card_BankId	Card_Type	Card_Balance	Card_ExpiryDate	PIN

ACCOUNT

Acc_No	Acc_Type	Acc_C_Id	Savings	Current

TRANSACTION

Trans_Id	Trans_Type	Trans_Amount	Trans_Status	Trans_Date	Trans_Time

Inquiry	Transfer	Withdraw	Deposit	ATM_Id

CUSTOMER_CARD

C_Id	Card

TRANSACTION_PARTICIPANTS

Trans_Id	C_Id	Role

Code For Creating Database:

```sql
CREATE TABLE Bank (
    Bank_Id VARCHAR(10) PRIMARY KEY,
    IFSC_Code VARCHAR(15),
    Bank_Add VARCHAR(80),
    Bank_Name VARCHAR(30),
    B_ATM_Id VARCHAR(10)
);
CREATE TABLE ATM_Machine (
    ATM_Id VARCHAR(10) PRIMARY KEY,
    ATM_Name VARCHAR(10),
    ATM_Add VARCHAR(80),
    ATM_Bank_Id VARCHAR(10),
    ATM_Branch VARCHAR(10),
    ATM_Amount DECIMAL(10, 2),
    FOREIGN KEY (ATM_Bank_Id ) REFERENCES Bank (Bank_Id)
);
CREATE TABLE Card (
    Card_No BIGINT PRIMARY KEY,
    Card_Bank_id VARCHAR(10),
    Card_CVV BIGINT,
    Card_ExpiryDate DATE,
    Card_Balance DECIMAL(16, 2),
    Card_Type VARCHAR(10),
    Pin INT,
    FOREIGN KEY (Card_Bank_id) REFERENCES Bank(Bank_Id)
);
CREATE TABLE Branch (
    Branch_Id VARCHAR(15) PRIMARY KEY,
    Branch_Name VARCHAR(15),
    Branch_loc VARCHAR(15),
    B_Bank_Id VARCHAR(10),
    FOREIGN KEY (B_Bank_Id) REFERENCES Bank(Bank_Id)
);
CREATE TABLE Customer (
    C_Id VARCHAR(10) PRIMARY KEY,
    C_add VARCHAR(80),
    F_name VARCHAR(20),
    M_name VARCHAR(20),
    L_name VARCHAR(20),
    C_phone BIGINT
);
CREATE TABLE Customer_Card (
    C_Id VARCHAR(10),
    Card_No BIGINT,
    PRIMARY KEY (C_Id, Card_No),
    FOREIGN KEY (C_Id) REFERENCES Customer(C_Id),
    FOREIGN KEY (Card_No) REFERENCES Card(Card_No)
);
CREATE TABLE Transaction (
    Transaction_Id VARCHAR(10) PRIMARY KEY,
    Transaction_type VARCHAR(10),
    Transaction_date DATE,
    Transaction_time TIME,
    Transaction_amount DECIMAL(10, 2),
    Transaction_status VARCHAR(10),
    T_ATM_Id VARCHAR(10),
    FOREIGN KEY (T_ATM_Id) REFERENCES ATM_Machine (ATM_Id)
);
CREATE TABLE Transaction_Participant (
    Transaction_Id VARCHAR(10),
    Participant_C_Id VARCHAR(10),
    Participant_Role VARCHAR(10), -- Add a column to denote participant role
    FOREIGN KEY (Transaction_Id) REFERENCES Transaction (Transaction_Id),
    FOREIGN KEY (Participant_C_Id) REFERENCES Customer (C_Id),
    CHECK (Participant_Role IN ('Sender', 'Receiver')) -- Constraint to ensure valid roles
);
CREATE TABLE Account (
    Acc_no BIGINT PRIMARY KEY,
    Acc_type VARCHAR(10),
    A_C_Id VARCHAR(10),
    FOREIGN KEY (A_C_Id) REFERENCES Customer(C_Id)
);
CREATE TABLE CurrentA (
    Acc_no BIGINT PRIMARY KEY,
    FOREIGN KEY (Acc_no) REFERENCES Account(Acc_no)
);
CREATE TABLE Savings (
    Acc_no BIGINT PRIMARY KEY,
    FOREIGN KEY (Acc_no) REFERENCES Account(Acc_no)
);
```

Procedure :

Procedure to change the pin number of a inserted card :

```
DELIMITER //
CREATE PROCEDURE ChangeCardPIN(
    IN card_number BIGINT,
    IN new_pin INT
)
BEGIN
    DECLARE card_count INT DEFAULT 0;
    SELECT COUNT(*) INTO card_count FROM Card WHERE Card_No = card_number;
    IF card_count > 0 THEN
        UPDATE Card SET Pin = new_pin WHERE Card_No = card_number;
        SELECT 'Success' AS Message;
    ELSE
        SELECT 'Card does not exist' AS Message;
    END IF;
END //
DELIMITER ;
```

Procedure to Update the card holder's amount during Deposit and updating the details in the transaction and in the transaction participant tables

```
DELIMITER //
CREATE PROCEDURE MakeDepositAndUpdateBalance(
    IN p_card_number BIGINT,
    IN p_deposit_amount DECIMAL(16, 2),
    IN p_atm_name VARCHAR(10)
)
BEGIN
    DECLARE v_customer_id VARCHAR(10);
    DECLARE v_transaction_id VARCHAR(10);
    DECLARE v_atm_id VARCHAR(10);
    SELECT ATM_Id INTO v_atm_id
    FROM ATM_Machine
    WHERE ATM_Name = p_atm_name
    LIMIT 1;
    UPDATE Card
    SET Card_Balance = Card_Balance + p_deposit_amount
    WHERE Card_No = p_card_number;
    SET v_transaction_id = CONCAT('T', LPAD(FLOOR(RAND() * 1000000), 6, '0'));
    INSERT INTO Transaction (Transaction_Id, Transaction_type, Transaction_date,
Transaction_time, Transaction_amount, Transaction_status, T_ATM_Id)
    VALUES (v_transaction_id, 'Deposit', CURDATE(), CURTIME(), p_deposit_amount,
'Success', v_atm_id);
    SELECT C_Id INTO v_customer_id
    FROM Customer_Card
    WHERE Card_No = p_card_number
    LIMIT 1;
    INSERT INTO Transaction_Participant (Transaction_Id, Participant_C_Id,
Participant_Role)
    VALUES (v_transaction_id, v_customer_id, 'Sender');
END //
DELIMITER ;
```

Procedure to Update the card holder's amount during Withdrawal and updating the details in the transaction and in the transaction participant tables

```
CREATE PROCEDURE MakeWithdrawalAndUpdateBalance(
    IN p_card_number BIGINT,
    IN p_withdrawal_amount DECIMAL(16, 2),
    IN p_atm_name VARCHAR(10)
)
BEGIN
    DECLARE v_customer_id VARCHAR(10);
    DECLARE v_transaction_id VARCHAR(10);
    DECLARE v_atm_id VARCHAR(10);
    DECLARE v_current_balance DECIMAL(16, 2);
    DECLARE v_atm_balance DECIMAL(10, 2);
    DECLARE v_transaction_status VARCHAR(10);
    SELECT ATM_Id, ATM_Amount INTO v_atm_id, v_atm_balance
    FROM ATM_Machine
    WHERE ATM_Name = p_atm_name
    LIMIT 1;
    SELECT Card_Balance INTO v_current_balance
    FROM Card
    WHERE Card_No = p_card_number;
    IF v_current_balance >= p_withdrawal_amount AND v_atm_balance >= p_withdrawal_amount
THEN
        UPDATE Card
        SET Card_Balance = Card_Balance - p_withdrawal_amount
        WHERE Card_No = p_card_number;
        SET v_transaction_status = 'Success';
    ELSE
        SET v_transaction_status = 'Failure';
    END IF;
    SET v_transaction_id = CONCAT('T', LPAD(FLOOR(RAND() * 1000000), 6, '0'));
    INSERT INTO `Transaction` (Transaction_Id, Transaction_type, Transaction_date,
Transaction_time, Transaction_amount, Transaction_status, T_ATM_Id)
    VALUES (v_transaction_id, 'Withdrawal', CURDATE(), CURTIME(), p_withdrawal_amount,
v_transaction_status, v_atm_id);
    SELECT C_Id INTO v_customer_id
    FROM Customer_Card
    WHERE Card_No = p_card_number
    LIMIT 1;
    INSERT INTO Transaction_Participant (Transaction_Id, Participant_C_Id,
Participant_Role)
    VALUES (v_transaction_id, v_customer_id, 'Sender');
END //
DELIMITER ;
```

Procedure to Update the card holder's amount and the receiver amount during Transfer and updating the details in the transaction and in the transaction participant tables

```
DELIMITER //
CREATE PROCEDURE MakeTransferAndUpdateBalance(
    IN p_sender_card_number BIGINT,
    IN p_receiver_account_number BIGINT,
    IN p_transfer_amount DECIMAL(16, 2),
    IN p_atm_name VARCHAR(10)
)
BEGIN
    DECLARE v_sender_customer_id VARCHAR(10);
    DECLARE v_receiver_customer_id VARCHAR(10);
    DECLARE v_transaction_id VARCHAR(10);
    DECLARE v_atm_id VARCHAR(10);
    DECLARE v_atm_balance DECIMAL(10, 2);
    DECLARE v_sender_current_balance DECIMAL(16, 2);
    DECLARE v_receiver_current_balance DECIMAL(16, 2);
    DECLARE v_transaction_status VARCHAR(10);
    SELECT ATM_Id, ATM_Amount INTO v_atm_id, v_atm_balance
    FROM ATM_Machine
    WHERE ATM_Name = p_atm_name
    LIMIT 1;
```

```sql
    SELECT Card_Balance INTO v_sender_current_balance
    FROM Card
    WHERE Card_No = p_sender_card_number;
    SELECT Card_Balance INTO v_receiver_current_balance
    FROM Card
    JOIN Account ON Card.Card_No = Account.Acc_no
    WHERE Account.Acc_no = p_receiver_account_number;
    IF v_sender_current_balance >= p_transfer_amount AND v_atm_balance >=
p_transfer_amount THEN
        SET v_transaction_id = CONCAT('T', LPAD(FLOOR(RAND() * 1000000), 6, '0'));
        UPDATE Card
        SET Card_Balance = Card_Balance - p_transfer_amount
        WHERE Card_No = p_sender_card_number;
        UPDATE Card
        SET Card_Balance = Card_Balance + p_transfer_amount
        WHERE Card_No = p_receiver_account_number;
        SET v_transaction_status = 'Success';
    ELSE
        SET v_transaction_status = 'Failure';
    END IF;
    INSERT INTO `Transaction` (Transaction_Id, Transaction_type, Transaction_date,
Transaction_time, Transaction_amount, Transaction_status, T_ATM_Id)
    VALUES (v_transaction_id, 'Transfer', CURDATE(), CURTIME(), p_transfer_amount,
v_transaction_status, v_atm_id);
    SELECT C_Id INTO v_sender_customer_id
    FROM Customer_Card
    WHERE Card_No = p_sender_card_number
    LIMIT 1;
    INSERT INTO Transaction_Participant (Transaction_Id, Participant_C_Id,
Participant_Role)
    VALUES (v_transaction_id, v_sender_customer_id, 'Sender')
    SELECT C_Id INTO v_receiver_customer_id
    FROM Account
    JOIN Customer ON Account.A_C_Id = Customer.C_Id
    WHERE Account.Acc_no = p_receiver_account_number
    LIMIT 1;
    INSERT INTO Transaction_Participant (Transaction_Id, Participant_C_Id,
Participant_Role)
    VALUES (v_transaction_id, v_receiver_customer_id, 'Receiver');
END //
DELIMITER
DELIMITER  //
CREATE PROCEDURE func(x IN INT)
BEGIN
 UPDATE ATM_Machine SET ATM_Balance='3000' WHERE ATM_Id=x;
 END
 //
 DELIMITER;
```

Function:

Function to return the Balance amount of the entered card

```
DELIMITER //
CREATE FUNCTION CheckBalance(p_cardNo BIGINT) RETURNS DECIMAL(16, 2)
BEGIN
    DECLARE v_balance DECIMAL(16, 2);
    SELECT Card_Balance INTO v_balance
    FROM Card
    WHERE Card_No = p_cardNo;
    RETURN v_balance;
END //
DELIMITER ;
```

Cursor:

Cursor to select the transaction and the customer details to put in receipt .

```
DELIMITER //
CREATE PROCEDURE GetLatestTransactionSenderReceiver()
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE transId VARCHAR(10);
    DECLARE transType VARCHAR(10);
    DECLARE transDate DATE;
    DECLARE transTime TIME;
    DECLARE transAmount DECIMAL(10, 2);
    DECLARE transStatus VARCHAR(10);
    DECLARE atmLocation VARCHAR(80);
    DECLARE customerCardNumber VARCHAR(20);
    DECLARE atmId VARCHAR(10);
    DECLARE cardBalance DECIMAL(16, 2);
    DECLARE customerName VARCHAR(60);
    DECLARE bankName VARCHAR(10);
    DECLARE cur CURSOR FOR
        SELECT
            T.Transaction_Id,
            T.Transaction_type,
            T.Transaction_date,
            T.Transaction_time,
            T.Transaction_amount,
            T.Transaction_status,
            A.ATM_Add AS atm_location,
            CONCAT('XXXXXXXXXXXX', SUBSTRING(CAST(C.Card_No AS CHAR(16)),
LENGTH(CAST(C.Card_No AS CHAR(16))) - 3, 4)) AS customer_card_number,
            T.T_ATM_Id AS atm_id,
            C.Card_Balance AS card_balance,
            CONCAT(Cust.F_name, ' ', IFNULL(Cust.M_name, ''), ' ', Cust.L_name) AS
customer_name,
            B.Bank_Name AS bank_name
        FROM
```

```
            Transaction T
        LEFT JOIN
            Transaction_Participant TP_Sender ON T.Transaction_Id =
TP_Sender.Transaction_Id AND TP_Sender.Participant_Role = 'Sender'
        LEFT JOIN
            Transaction_Participant TP_Receiver ON T.Transaction_Id =
TP_Receiver.Transaction_Id AND TP_Receiver.Participant_Role = 'Receiver'
        LEFT JOIN
            ATM_Machine A ON T.T_ATM_Id = A.ATM_Id
        LEFT JOIN
            Bank B ON A.ATM_Bank_Id = B.Bank_Id
        LEFT JOIN
            Customer_Card CC ON TP_Sender.Participant_C_Id = CC.C_Id
        LEFT JOIN
            Card C ON CC.Card_No = C.Card_No
        LEFT JOIN
            Customer Cust ON TP_Sender.Participant_C_Id = Cust.C_Id
        ORDER BY
            T.Transaction_date DESC,
            T.Transaction_time DESC
        LIMIT 1;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN cur;
    FETCH cur INTO transId, transType, transDate, transTime, transAmount, transStatus,
atmLocation, customerCardNumber, atmId, cardBalance, customerName, bankName;
    IF NOT done THEN        SELECT
            transId AS Transaction_Id,
            transType AS Transaction_type,
            transDate AS Transaction_date,
            transTime AS Transaction_time,
            transAmount AS Transaction_amount,
            transStatus AS Transaction_status,
            atmLocation AS ATM_Location,
            customerCardNumber AS Customer_Card_Number,
            atmId AS ATM_Id,
            cardBalance AS Card_Balance,
            customerName AS Customer_Name,
            bankName AS Bank_Name;
    ELSE
        SELECT 'No transactions found' AS Debug_Info;
    END IF;
    CLOSE cur;
END //
DELIMITER ;
```

Trigger:

Trigger to increase the ATM amount if a successful transaction of deposit happens & decrease the ATM amount if a successful transaction of withdrawal happens.

```
DELIMITER //
CREATE TRIGGER handle_transaction
```

```
AFTER INSERT ON Transaction
FOR EACH ROW
BEGIN
    DECLARE atm_balance DECIMAL(10, 2);
    SELECT ATM_Amount INTO atm_balance
    FROM ATM_Machine
    WHERE ATM_Id = NEW.T_ATM_Id;
    IF atm_balance IS NOT NULL THEN
        IF NEW.Transaction_type = 'Deposit' THEN
            SET atm_balance = atm_balance + NEW.Transaction_amount;
            UPDATE ATM_Machine
            SET ATM_Amount = atm_balance
            WHERE ATM_Id = NEW.T_ATM_Id;
                ELSEIF NEW.Transaction_type = 'Withdrawal' AND NEW.Transaction_status =
'Success' THEN
            SET atm_balance = atm_balance - NEW.Transaction_amount;
            UPDATE ATM_Machine
            SET ATM_Amount = atm_balance
            WHERE ATM_Id = NEW.T_ATM_Id;
        END IF;
    ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'ATM balance is NULL';
    END IF;
END //
DELIMITER ;
```

Trigger to update ATM amount if it is less than 1000

```
DELIMITER//
CREATE TRIGGER update_atm_amount
AFTER UPDATE ON atm_machine
FOR EACH ROW
BEGIN
DECLARE A int;
SELECT ATM_Id INTO A where NEW.ATM_Balace<1000;
    IF NEW.ATM_Balance < 1000 THEN
        CALL func(A);
    END IF;
END;
DELIMITER;
```

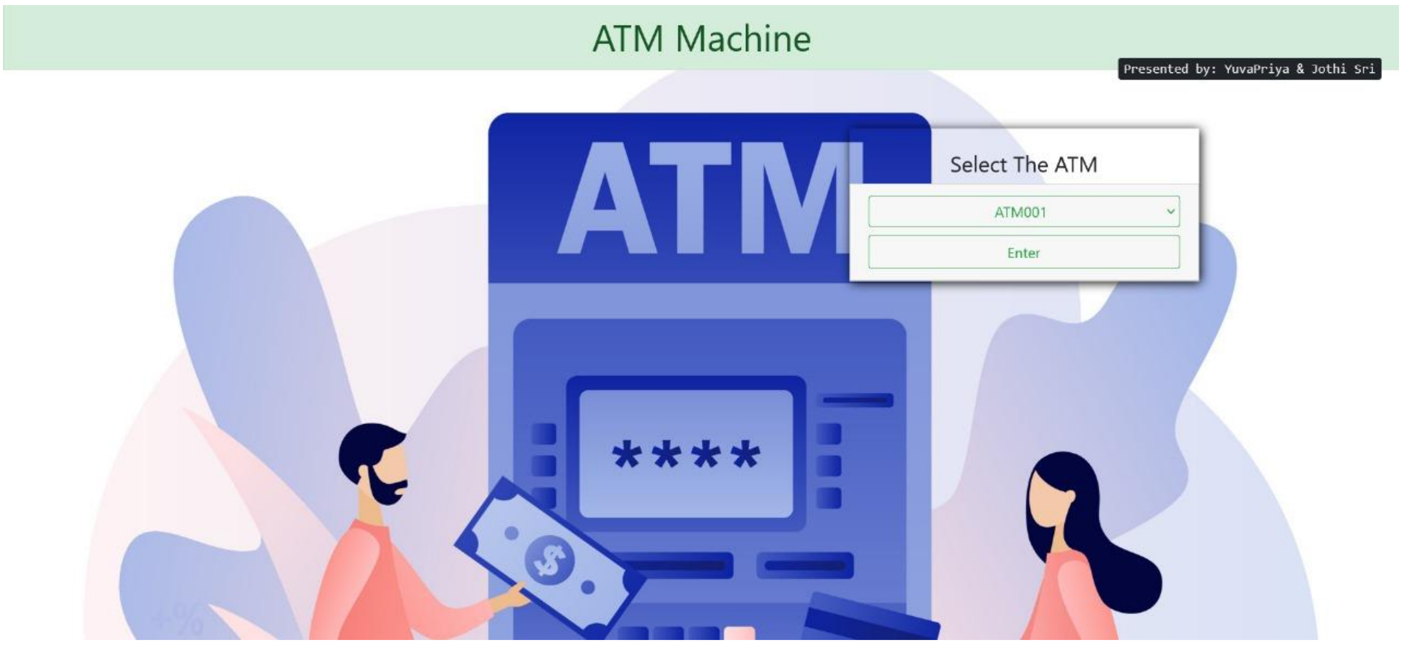
View:

```
CREATE VIEW Transaction_Sender_Receiver_View AS
SELECT
    T.Transaction_Id,
    T.Transaction_type,
    T.Transaction_date,
    T.Transaction_time,
    T.Transaction_amount,
    T.Transaction_status,
    CASE
        WHEN TP_Sender.Participant_Role = 'Sender' THEN TP_Sender.Participant_C_Id
        ELSE NULL
    END AS sender_id,
    CASE
        WHEN TP_Receiver.Participant_Role = 'Receiver' THEN TP_Receiver.Participant_C_Id
        ELSE NULL
    END AS receiver_id
FROM
    Transaction T
LEFT JOIN
    Transaction_Participant TP_Sender ON T.Transaction_Id = TP_Sender.Transaction_Id AND
TP_Sender.Participant_Role = 'Sender'
LEFT JOIN
    Transaction_Participant TP_Receiver ON T.Transaction_Id = TP_Receiver.Transaction_Id
AND TP_Receiver.Participant_Role = 'Receiver';
```

Input/Output pictures:

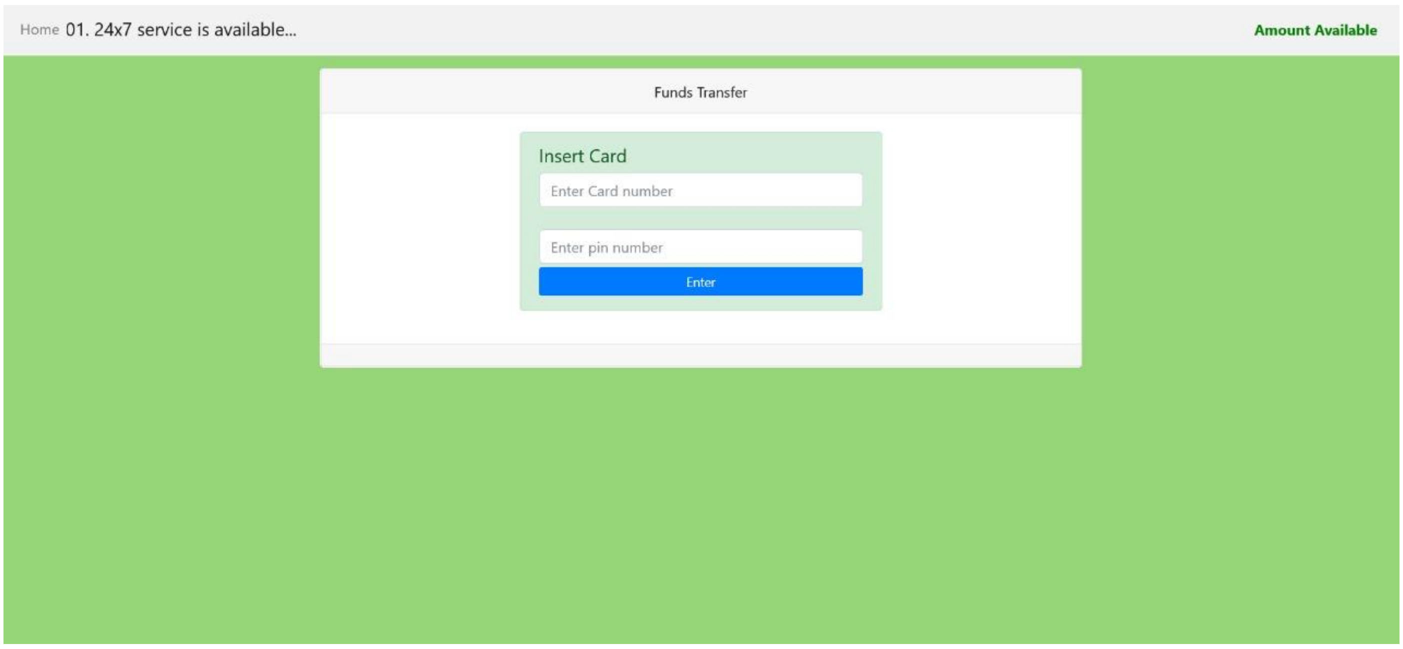
Index.php

(Index page to select a ATM among Multiple ATMs of different banks)



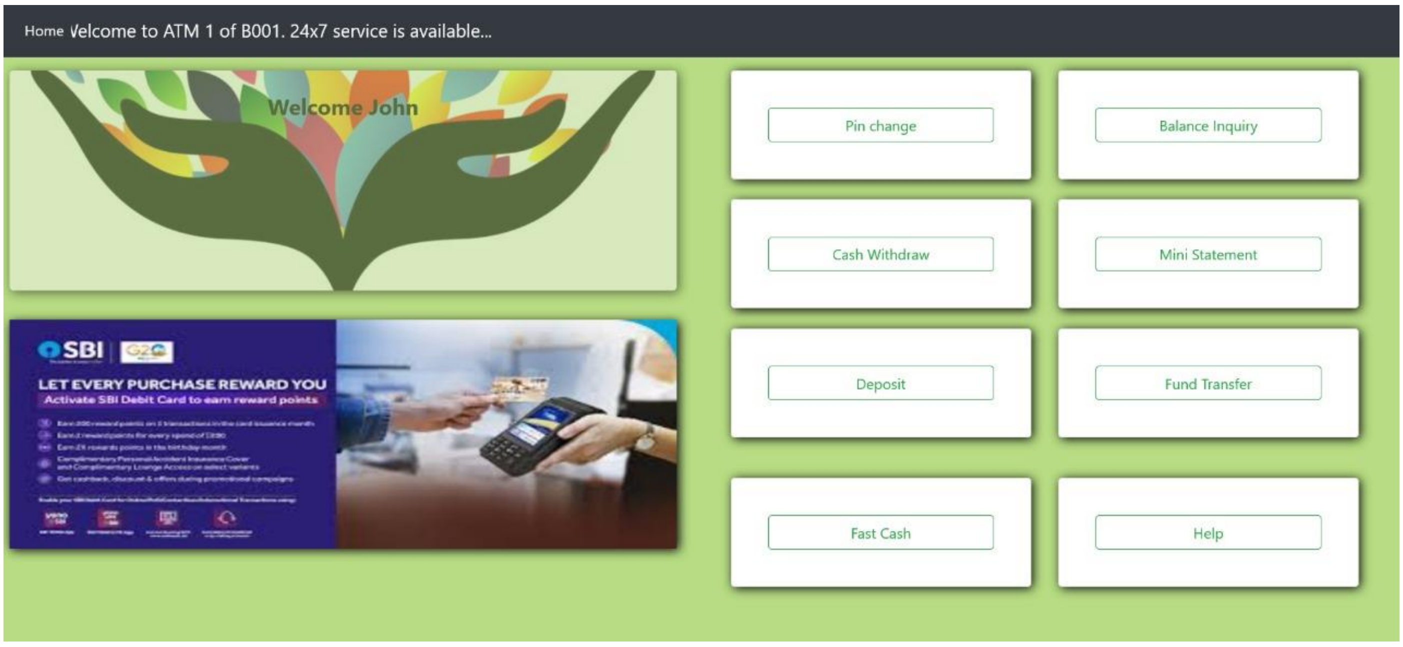
Index1.php

(Page to enter the card number and the pin number. This page validates the card, whether it is expired or not, validates the PIN and the checks whether the card is debit card or credit card)



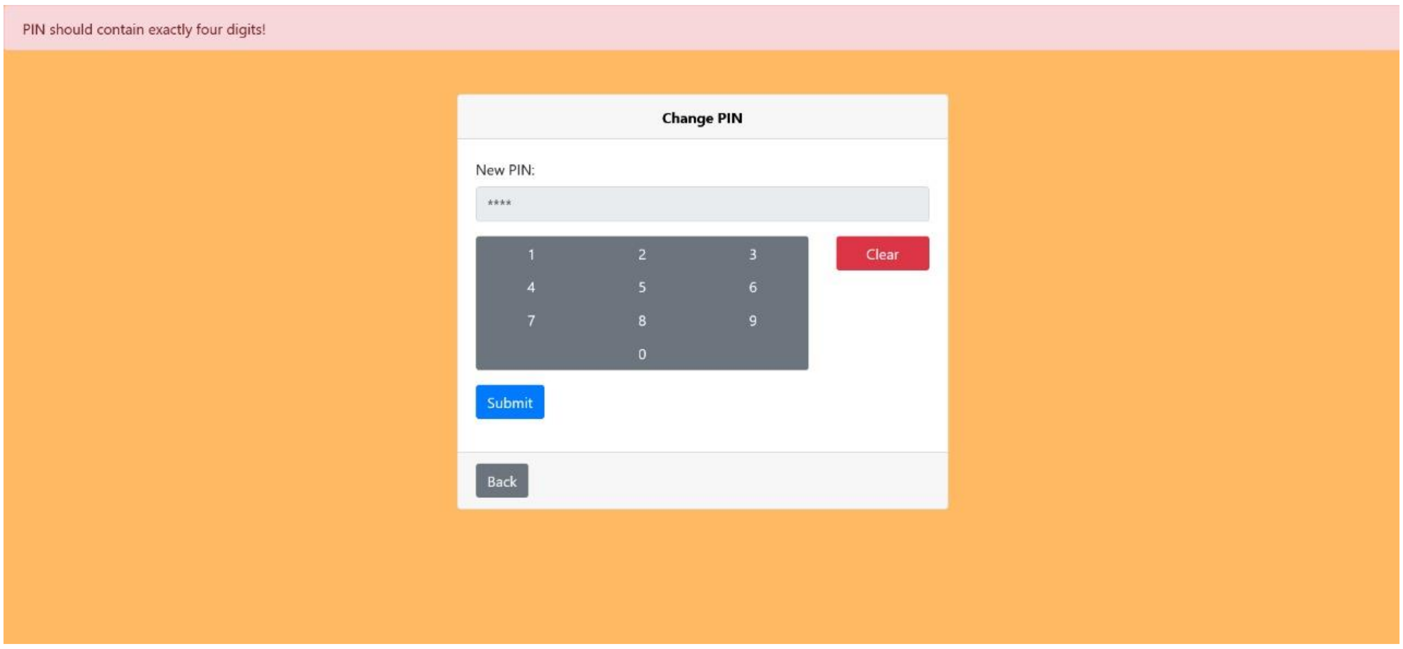
Main_page.php

(This page contains the options of whatever performance an ATM can do, i.e Deposit, withdraw, transfer, mini statement and so on…)


Home Velcome to ATM 1 of B001. 24x7 service is available...
Welcome John
SBI
LET EVERY PURCHASE REWARD YOU
Activate SBI Debit Card to earn reward points
Pin change
Balance Inquiry
Cash Withdraw
Mini Statement
Deposit
Fund Transfer
Fast Cash
Help

Pin_change.php

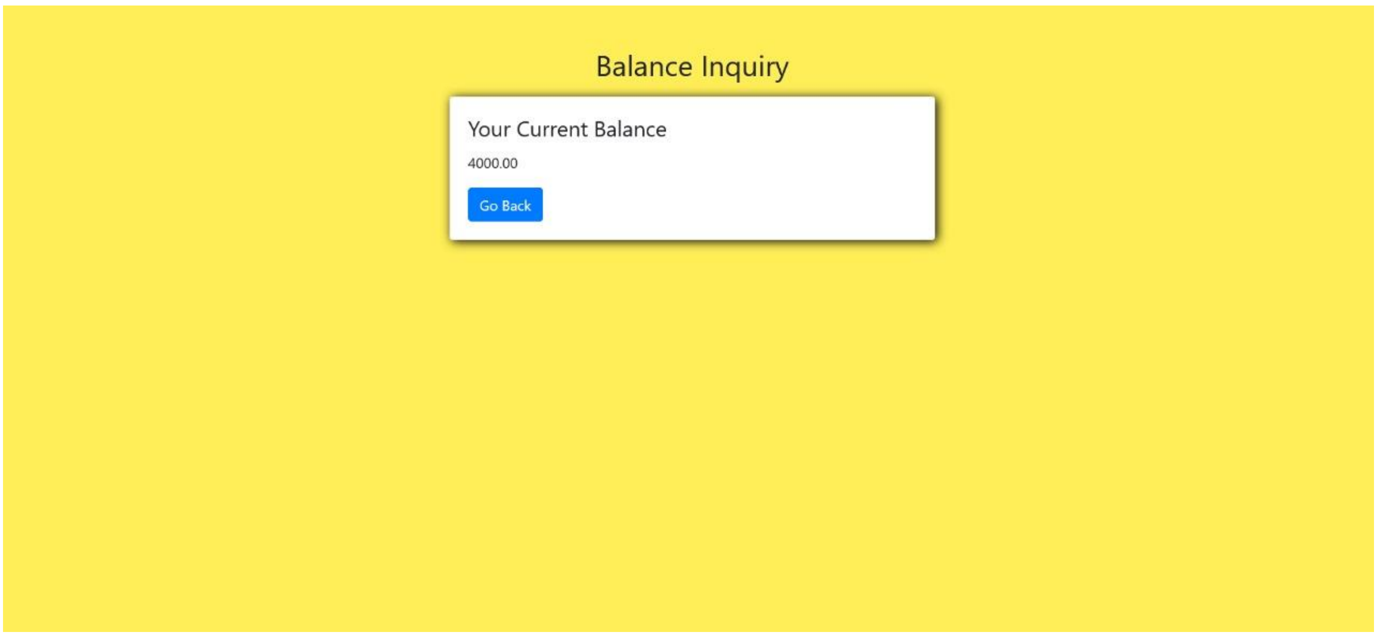
(This page is used to change the pin of the entered card. Here a procedure is used to update the pin in the database.)



PIN should contain exactly four digits!

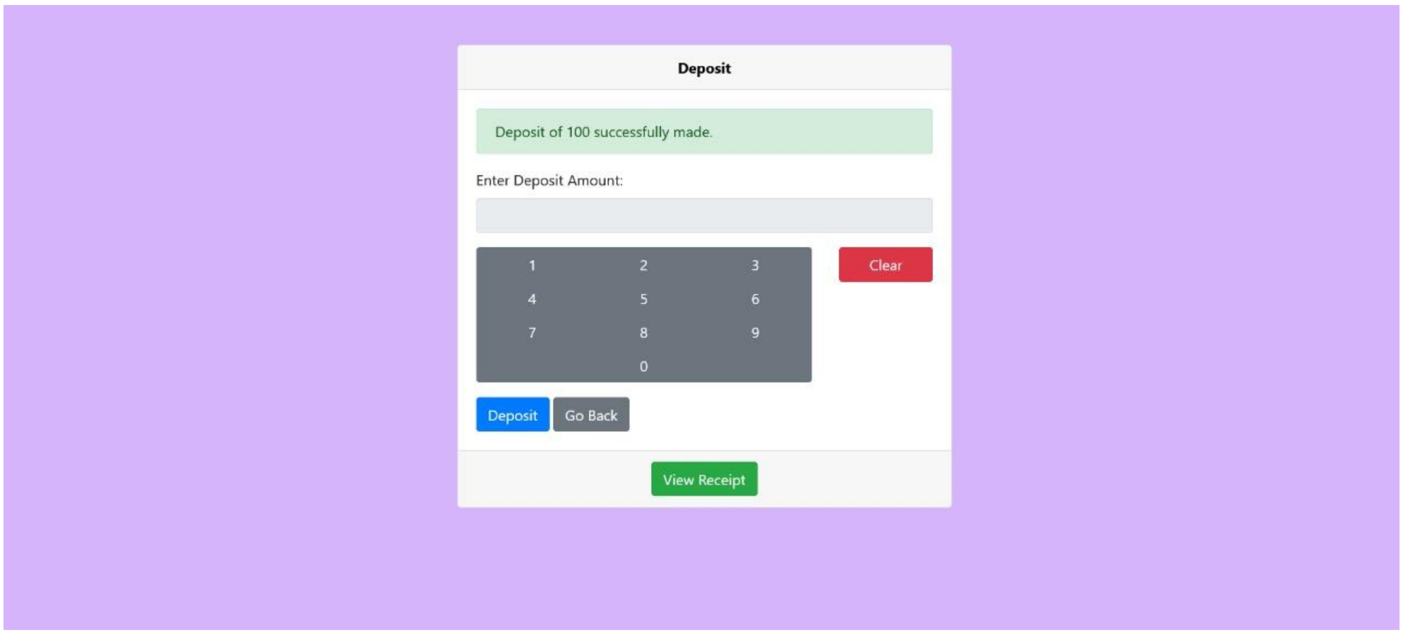
Balance.php

(This page is used to check the balance of the entered card. Here a function is used to return the balance amount of the entered card.)



Deposit.php

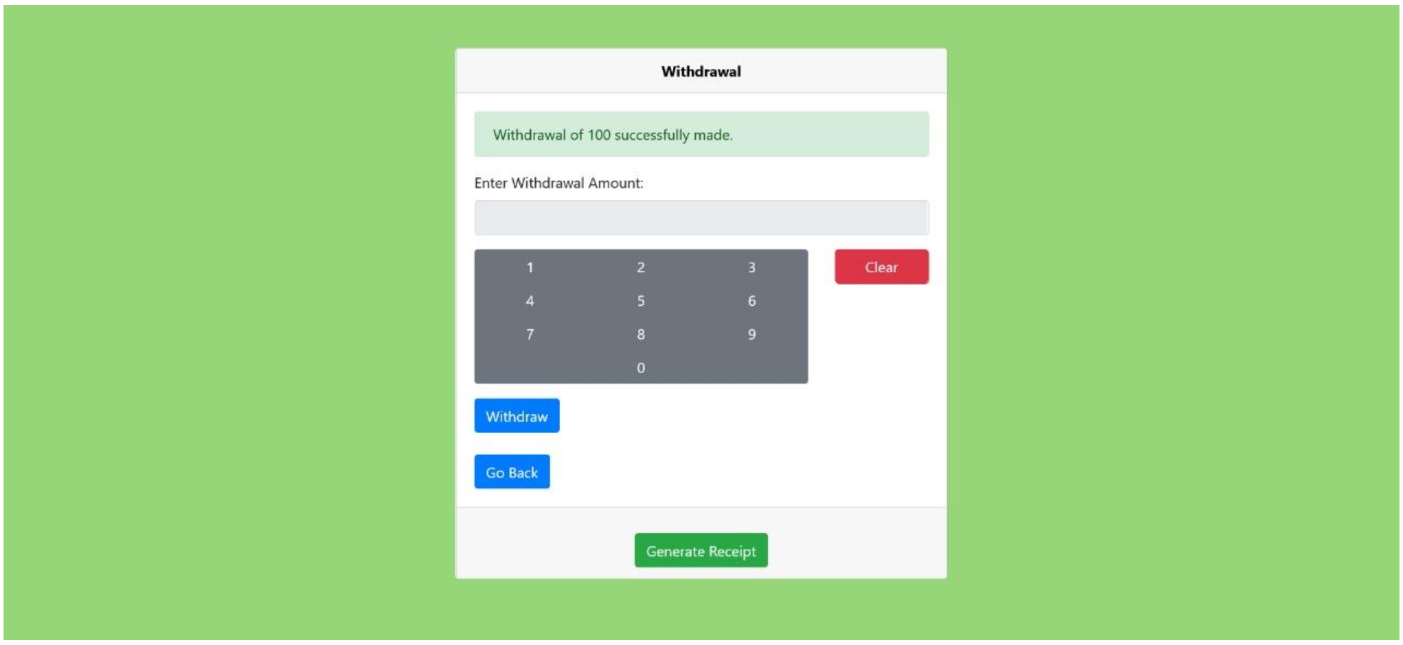
(This page is used to deposit a limited amount to the account and to get a corresponding receipt. Here a Procedure is used to update the successful transaction in the transaction and the transaction participant tables)



Deposit of 100 successfully made.

Withdraw.php

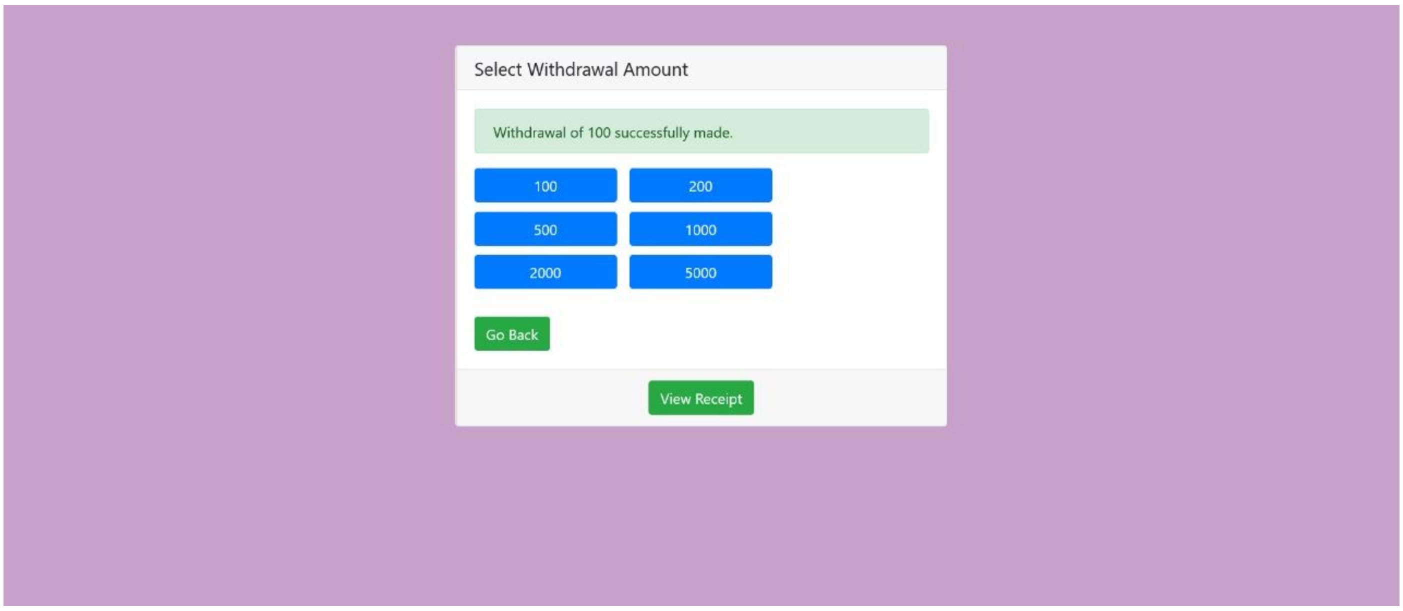
(This page is used to withdraw a limited amount from the card and to get a corresponding receipt. Here a Procedure is used to update the successful transaction in the transaction and the transaction participant tables and to reduce the amount in that card)



Withdrawal of 100 successfully made.

Fastcash.php

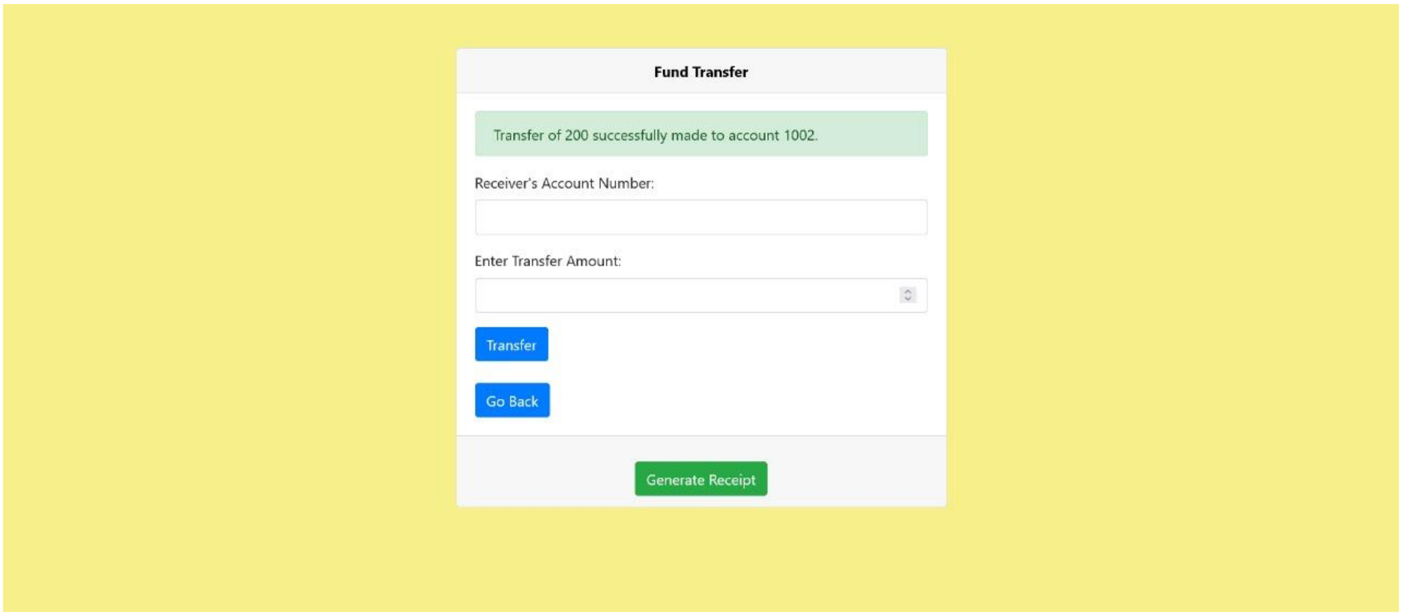
(This page is used to withdraw a selected amount given in the provided options from the card and to get a corresponding receipt. Here a Procedure is used to update the successful transaction in the transaction and the transaction participant tables and to reduce the amount in that card)


Select Withdrawal Amount

Withdrawal of 100 successfully made.

Fundtransfer.php

(This page is used to transfer an amount from the card to the entered account number. After a successful transfer, this page will provide a corresponding receipt. Here a Procedure is used to update the successful transaction in the transaction and the transaction participant tables and to reduce the amount in that card)


Fund Transfer

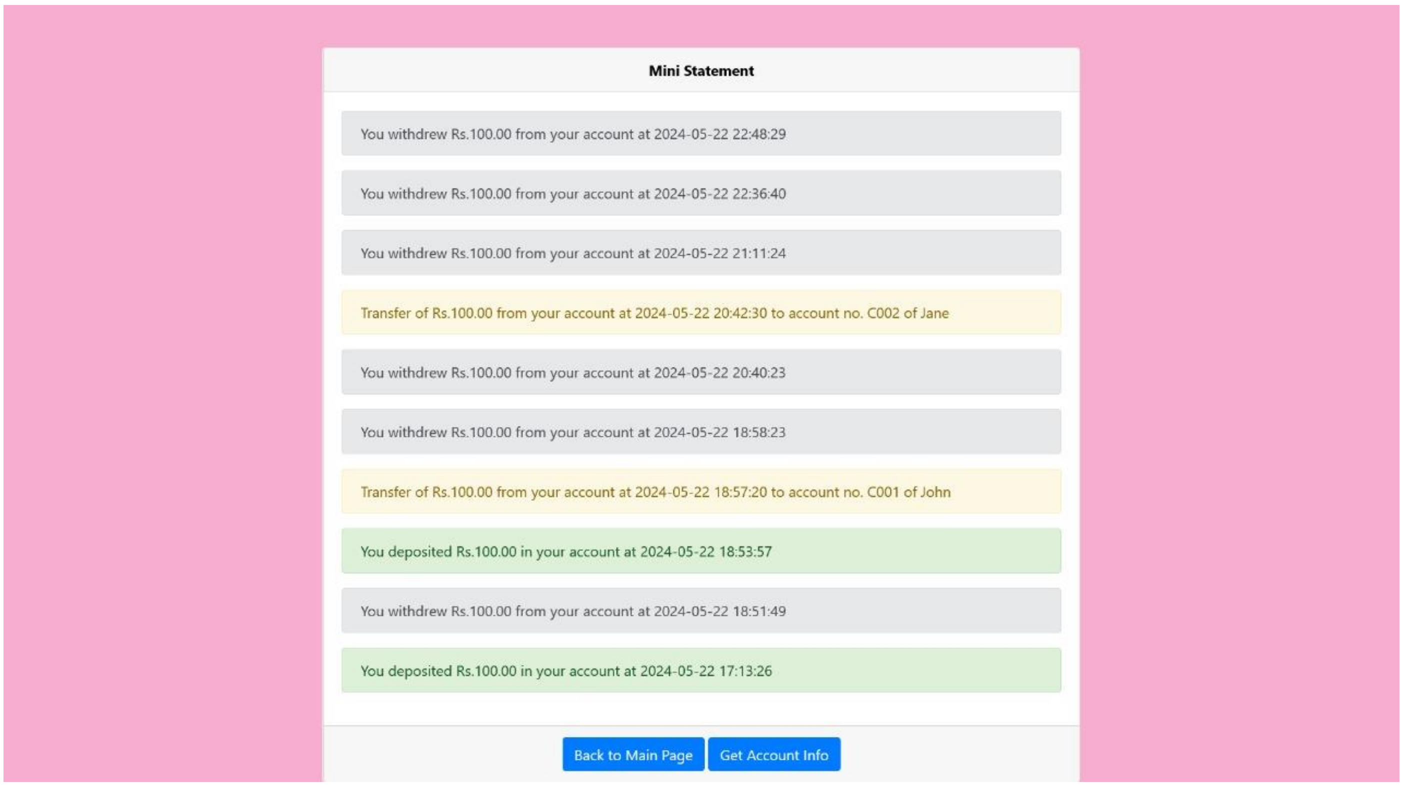
Transfer of 200 successfully made to account 1002.

Receiver's Account Number:

Enter Transfer Amount:

Mini_statement.php

(This page is used to view the last ten recently happened transaction details of the entered card and this page has an account info button. Here a view is used to fetch the required data)



Mini Statement

You withdrew Rs.100.00 from your account at 2024-05-22 22:48:29

You withdrew Rs.100.00 from your account at 2024-05-22 22:36:40

You withdrew Rs.100.00 from your account at 2024-05-22 21:11:24

Transfer of Rs.100.00 from your account at 2024-05-22 20:42:30 to account no. C002 of Jane

You withdrew Rs.100.00 from your account at 2024-05-22 20:40:23

You withdrew Rs.100.00 from your account at 2024-05-22 18:58:23

Transfer of Rs.100.00 from your account at 2024-05-22 18:57:20 to account no. C001 of John

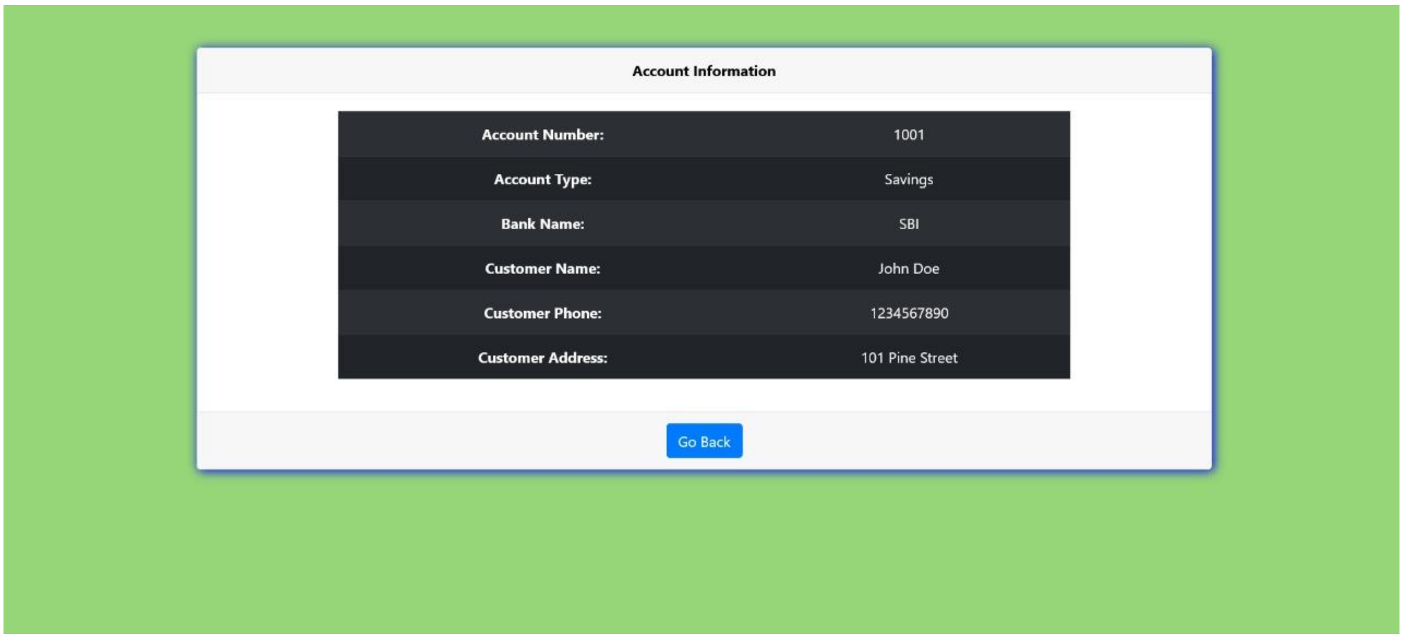
You deposited Rs.100.00 in your account at 2024-05-22 18:53:57

You withdrew Rs.100.00 from your account at 2024-05-22 18:51:49

You deposited Rs.100.00 in your account at 2024-05-22 17:13:26

Acount_info.php

(This page is to view the account details of the entered card)

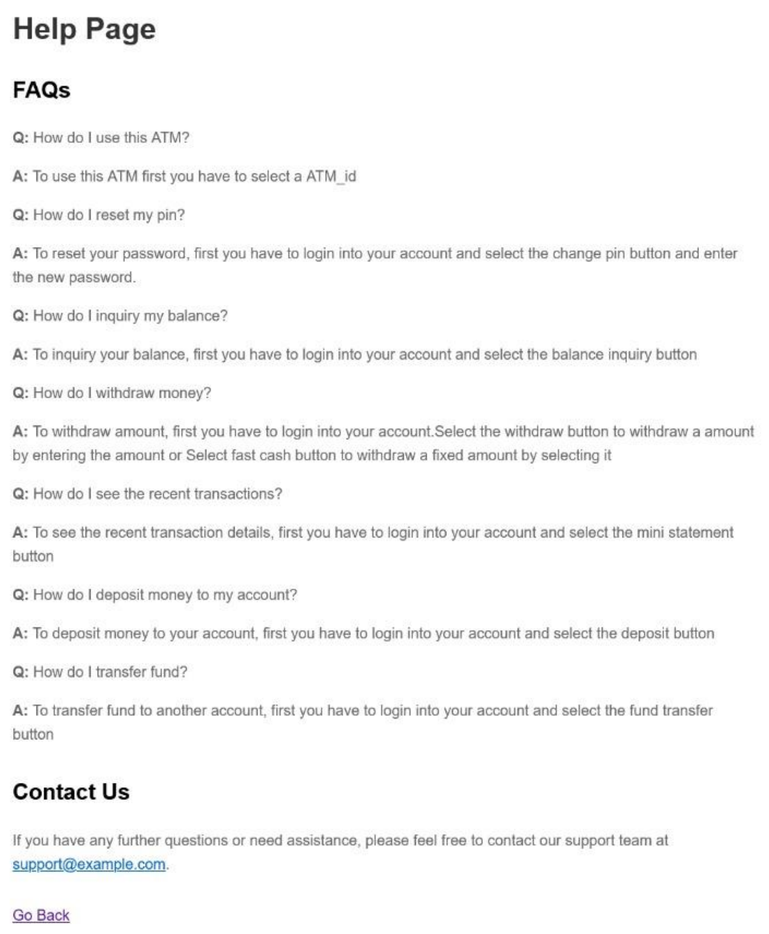


Account Information

Account Number:	1001
Account Type:	Savings
Bank Name:	SBI
Customer Name:	John Doe
Customer Phone:	1234567890
Customer Address:	101 Pine Street

Help.php

(This page contains the script of questions and answers to solve the user’s doubts of using this ATM)



Help Page

FAQs

Q: How do I use this ATM?

A: To use this ATM first you have to select a ATM_id

Q: How do I reset my pin?

A: To reset your password, first you have to login into your account and select the change pin button and enter the new password.

Q: How do I inquiry my balance?

A: To inquiry your balance, first you have to login into your account and select the balance inquiry button

Q: How do I withdraw money?

A: To withdraw amount, first you have to login into your account.Select the withdraw button to withdraw a amount by entering the amount or Select fast cash button to withdraw a fixed amount by selecting it

Q: How do I see the recent transactions?

A: To see the recent transaction details, first you have to login into your account and select the mini statement button

Q: How do I deposit money to my account?

A: To deposit money to your account, first you have to login into your account and select the deposit button

Q: How do I transfer fund?

A: To transfer fund to another account, first you have to login into your account and select the fund transfer button

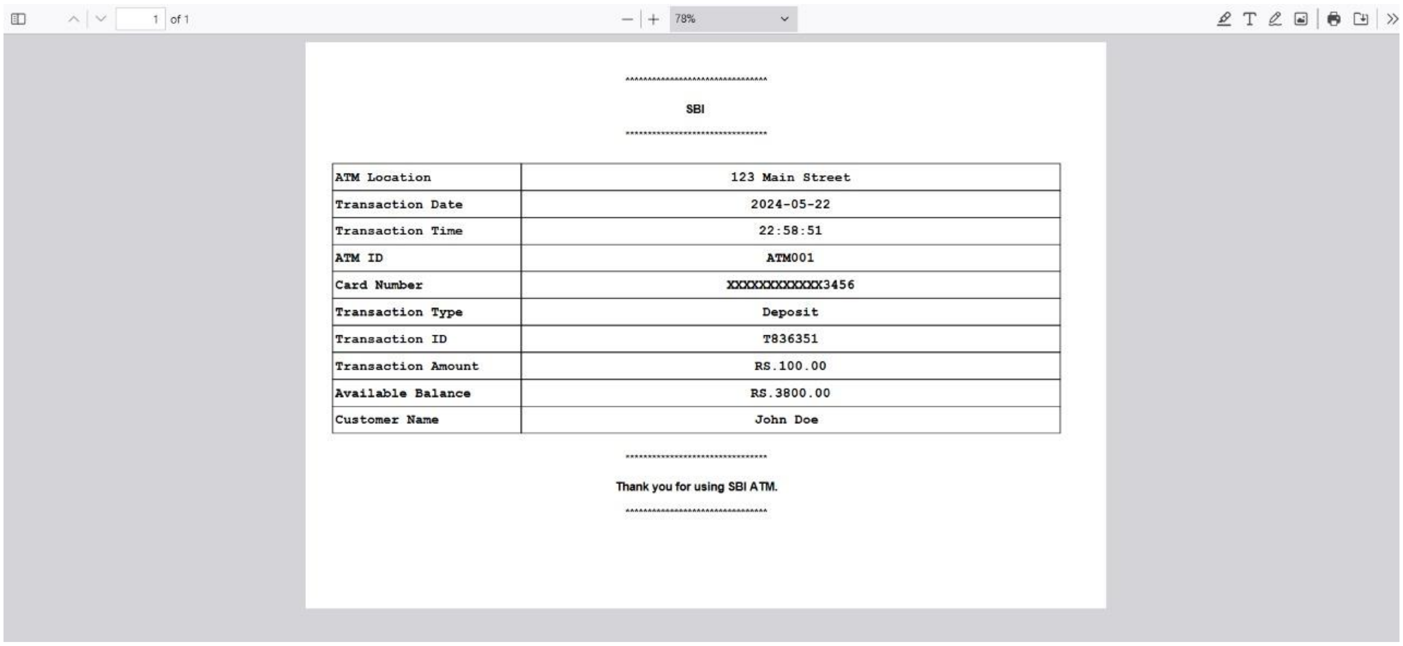
Contact Us

If you have any further questions or need assistance, please feel free to contact our support team at support@example.com.

Go Back

Receipt.php

(This receipt generating file using FPDF. FPDF is a free and open-source PHP class that provides an easy way to create and manipulate PDF documents programmatically. Using this, We can able to view and download the receipt. Here a cursor is used to select the required data to put into the receipt)



1 of 1

78%

****

SBI

****

ATM Location	123 Main Street
Transaction Date	2024-05-22
Transaction Time	22:58:51
ATM ID	ATM001
Card Number	XXXXXXXXXXXX3456
Transaction Type	Deposit
Transaction ID	T836351
Transaction Amount	RS.100.00
Available Balance	RS.3800.00
Customer Name	John Doe

****

Thank you for using SBI ATM.

****

We used MYSQL to design our database

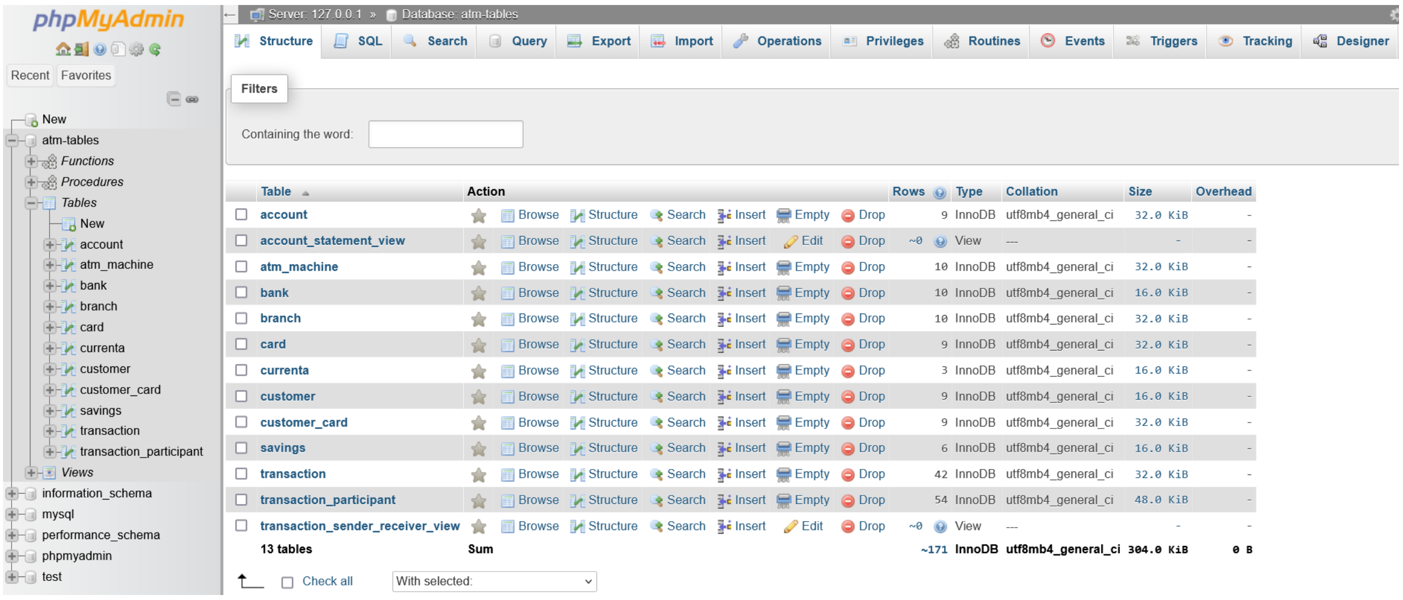
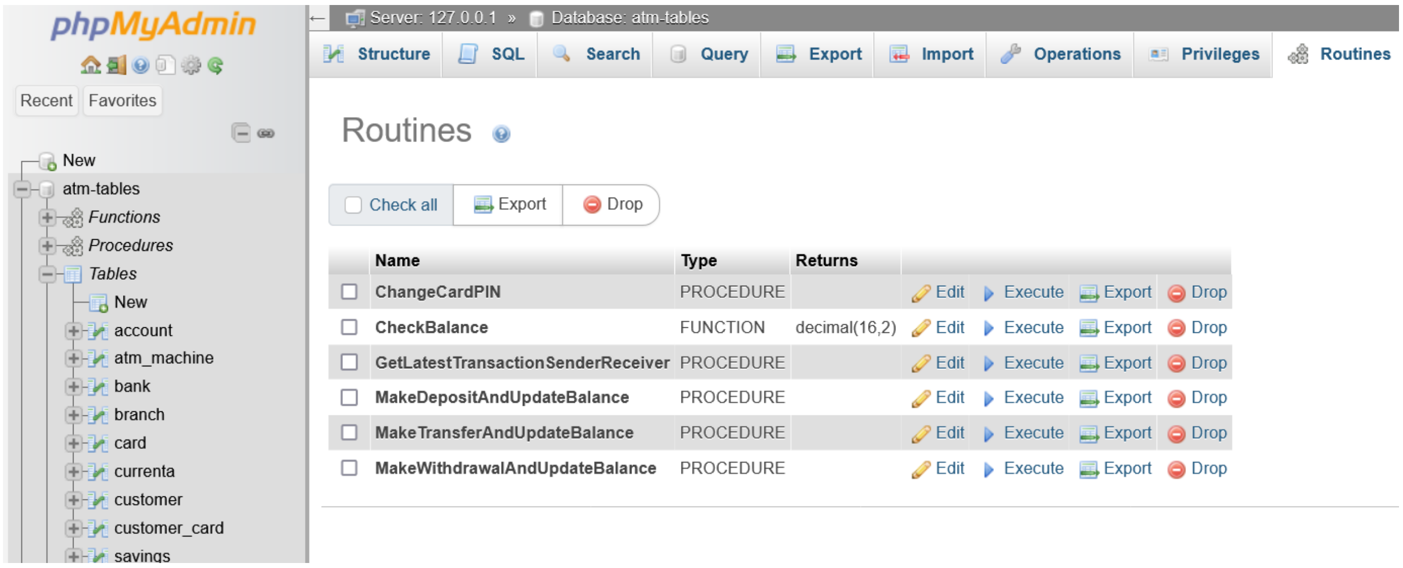


Table	Action	Rows	Type	Collation	Size	Overhead
account	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 KiB	-
account_statement_view	Browse Structure Search Insert Edit Drop	~0	View	---	-	-
atm_machine	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	32.0 KiB	-
bank	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	16.0 KiB	-
branch	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	32.0 KiB	-
card	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 KiB	-
currenta	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
customer	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
customer_card	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 KiB	-
savings	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
transaction	Browse Structure Search Insert Empty Drop	42	InnoDB	utf8mb4_general_ci	32.0 KiB	-
transaction_participant	Browse Structure Search Insert Empty Drop	54	InnoDB	utf8mb4_general_ci	48.0 KiB	-
transaction_sender_receiver_view	Browse Structure Search Insert Edit Drop	~0	View	---	-	-
13 tables	Sum	~171	InnoDB	utf8mb4_general_ci	304.0 KiB	0 B



Routines

Name	Type	Returns	
ChangeCardPIN	PROCEDURE		Edit Execute Export Drop
CheckBalance	FUNCTION	decimal(16,2)	Edit Execute Export Drop
GetLatestTransactionSenderReceiver	PROCEDURE		Edit Execute Export Drop
MakeDepositAndUpdateBalance	PROCEDURE		Edit Execute Export Drop
MakeTransferAndUpdateBalance	PROCEDURE		Edit Execute Export Drop
MakeWithdrawalAndUpdateBalance	PROCEDURE		Edit Execute Export Drop

Conclusion:

The ATM database project successfully demonstrates the development and implementation of a robust database system tailored for managing ATM transactions and user data. By incorporating Procedures, functions, and triggers for each task, the system ensures user confidentiality and data integrity. The project highlights key functionalities such as user authentication, transaction recording, and balance management, providing a seamless and secure banking experience. Overall, this project underscores the importance of database management in financial systems and provides a user-friendly environment.