INTRODUCTION

A complete software programme called a bank management system is made to make running a bank or other financial institution easier and more efficient. It acts as a central platform for integrating several features, enabling the effective management of financial transactions, client accounts, and other crucial banking tasks.

Bank management systems have recently integrated digital channels and online banking features in response to technological improvements. Customers may now effortlessly access their accounts, conduct transactions, and carry out other banking tasks using online portals or mobile applications.

WHAT IS THE NEED OF A BANK MANAGEMENT SYSTEM?

An effective bank management system is required to increase operational effectiveness, provide security, and assure compliance, enhance client service, make data-driven decisions possible, and adjust to the world of digital banking. It gives banks the ability to satisfy client demands, remain competitive, and have a firm control on the financial sector.

FEATURES OF THE PROJECT

The Management System consists of the following features:

Account Class:

This class is an abstract class and consists of the common features to both current and savings accounts.

The Account Class encompasses the following methods or features:

NewUser:

This feature is used to open a new bank account i.e. either current or savings.

Deposit Money:

This feature is used to deposit money from the respective bank account.

Withdraw Money:

This feature is used to withdraw money from the respective bank account.

View Bank Details:

This feature is used to view the details of the respective existing bank account.

Modify Details:

This feature is used to modify the details of the respective bank account.

Current Class:

This class is derived from account class and offers some extra features to the current account holders.

The Current Class consists of the following methods or features:

Login:

This feature is used to login into the existing current account using the Customer ID and Password.

Issue Cheque Book:

This feature is used issue a new cheque book to the respective current account holder.

View Charges Imposed:

This feature is used to view the penalties or the charges imposed on the respective current account holder.

Logout:

This feature is used to securely LogOut from the bank account.

Savings Class:

This class is derived from account class and offers some extra features to the saving account holders.

The Savings Class consists of the following methods or features:

Login:

This feature is used to login into the existing savings account using the Customer ID and Password.

Issue Debit Card:

This feature is used issue a Debit Card to the respective savings account holder (if not previously issued).

Get A Loan:

This feature is used to Get a Loan and encompasses the following types of loans:

- *House Loan*
- *Personal Loan*
- *Education Loan*
- *Agricultural Loan*
- *Vehicle Loan*
- *Gold Loan*

Each type of loan has its own rate of interest which will be displayed to the user.

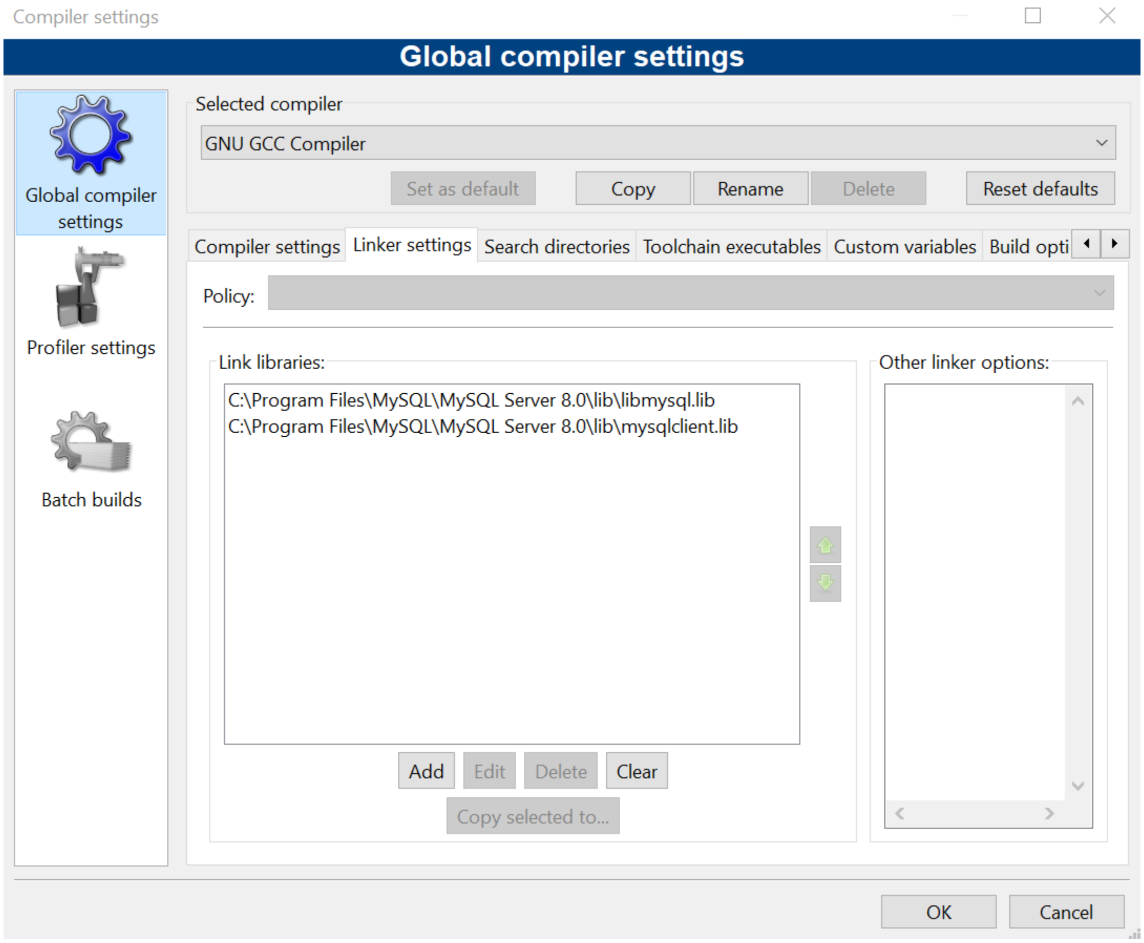
Logout:

This feature is used to securely LogOut from the bank account.

PRE-REQUISITES

The settings required in CODE::BLOCKS to run the program successfully are:

- Go to Compiler Settings in CODE::BLOCKS and **ADD** the following libraries as shown:



Compiler settings

Global compiler settings

Selected compiler: GNU GCC Compiler

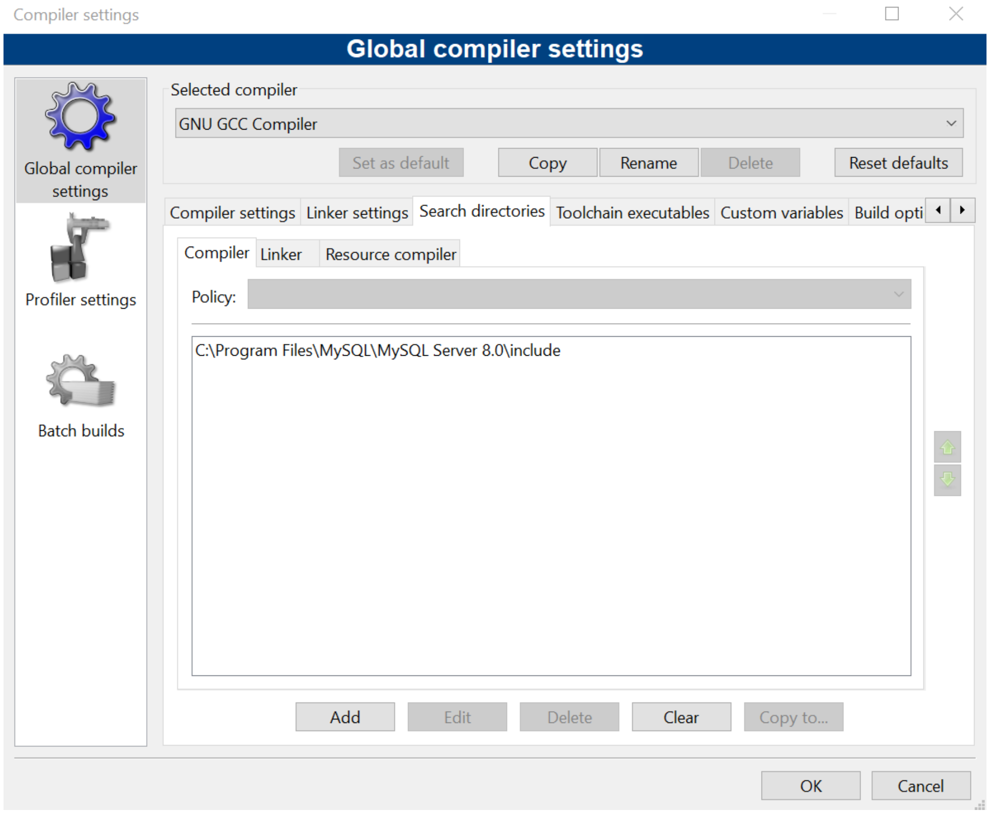
Policy:

Link libraries:

C:\Program Files\MySQL\MySQL Server 8.0\lib\libmysql.lib
C:\Program Files\MySQL\MySQL Server 8.0\lib\mysqlclient.lib

Other linker options:

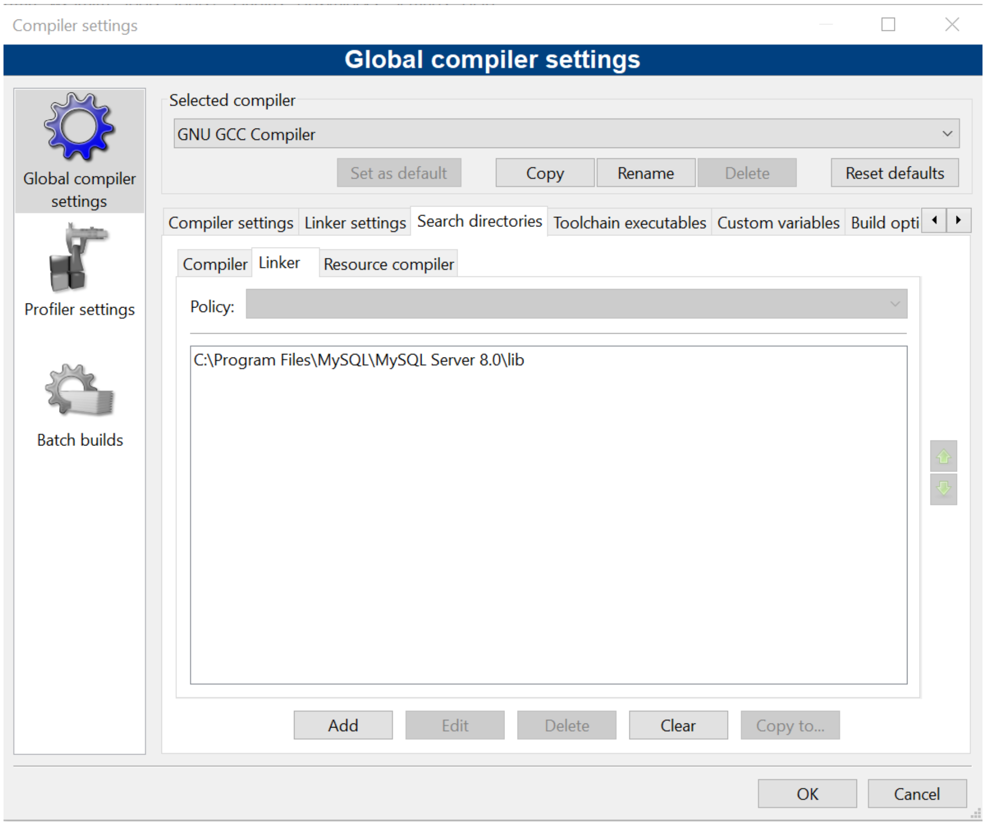
- Now go to Search Directories and **ADD** the following folder in Compiler Tab:



Selected compiler: GNU GCC Compiler

C:\Program Files\MySQL\MySQL Server 8.0\include

- Now go to Search Directories and **ADD** the following folder in Linker Tab:



Selected compiler: GNU GCC Compiler

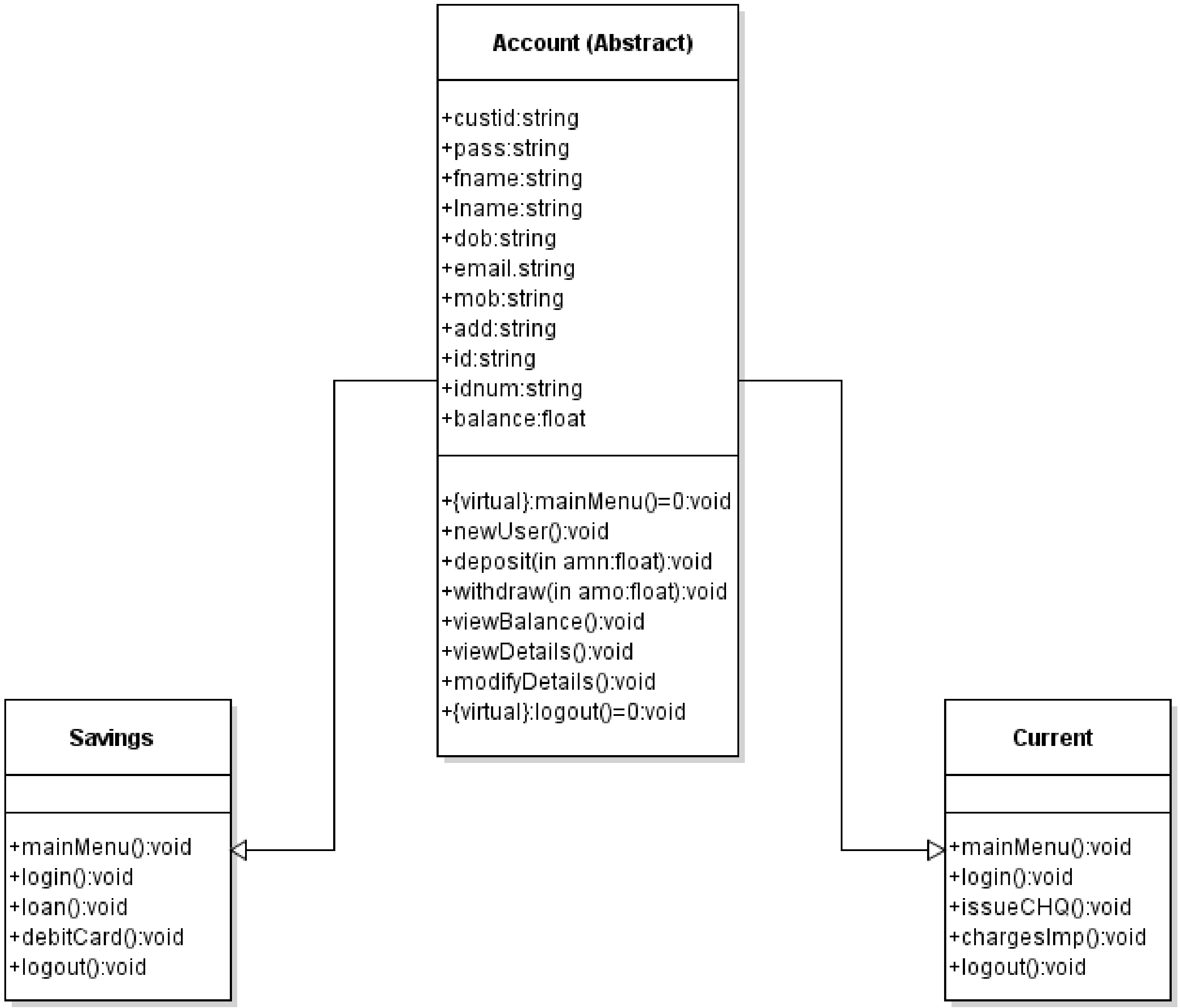
C:\Program Files\MySQL\MySQL Server 8.0\lib

The settings required in MySQL to run the program successfully are:

- Open MySQL Command Line Client and execute the following commands:
    - create database bank;
    - create table current (CustomerID char(8) primary key, password varchar(16) unique, FirstName varchar(20), Lastname varchar(20), DOB date, Email varchar(50), MobileNo char(10), Address varchar(50), IDType varchar(50), IDNum varchar(10) unique, balance decimal(10,2) default 0, ChequeBook varchar(50) default ‘Not Issued’);
    - create table savings (CustomerID char(8) primary key, password varchar(16) unique, FirstName varchar(20), Lastname varchar(20), DOB date, Email varchar(50), MobileNo char(10), Address varchar(50), IDType varchar(50), IDNum varchar(10) unique, balance decimal(10,2) default 0, DebitCard varchar(50) default ‘Not Issued’);

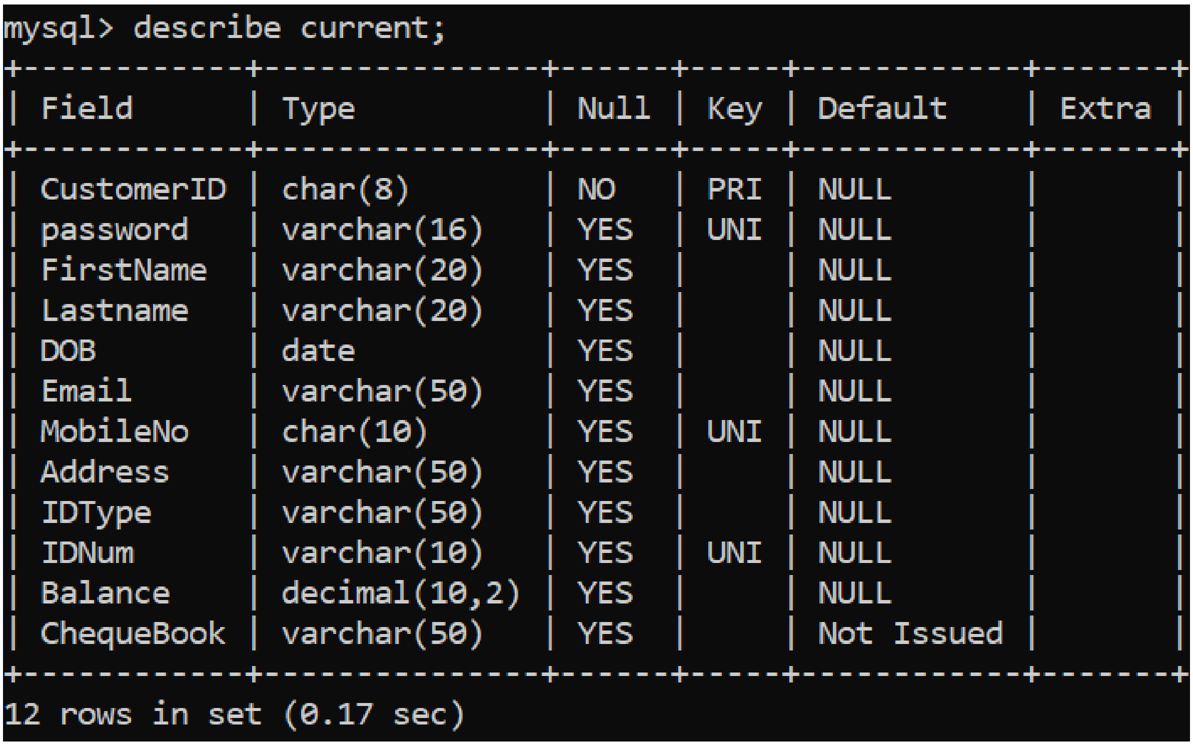
DESIGN OF PROJECT

- **UML Class Diagram**



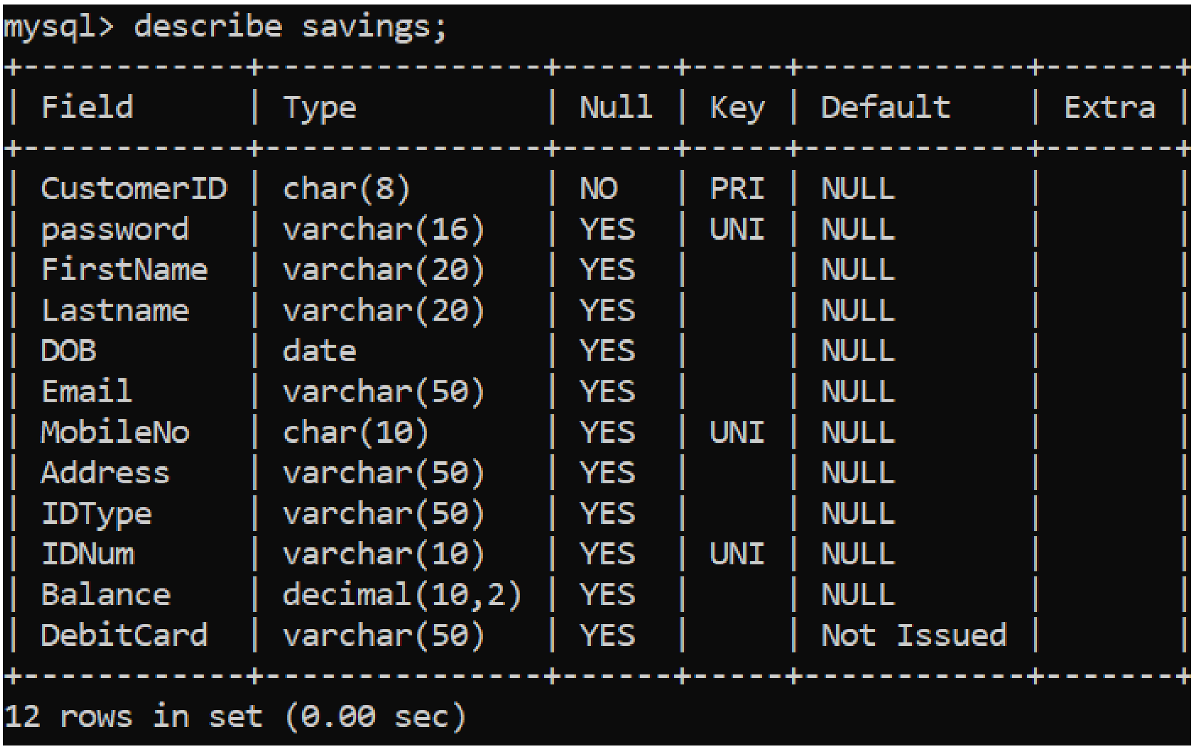
- The project consists of two tables/relations whose structure is as shown below:

- **current:**



```
mysql> describe current;
+------------+---------------+------+-----+------------+-------+
| Field      | Type          | Null | Key | Default    | Extra |
+------------+---------------+------+-----+------------+-------+
| CustomerID | char(8)       | NO   | PRI | NULL       |       |
| password   | varchar(16)   | YES  | UNI | NULL       |       |
| FirstName  | varchar(20)   | YES  |     | NULL       |       |
| Lastname   | varchar(20)   | YES  |     | NULL       |       |
| DOB        | date          | YES  |     | NULL       |       |
| Email      | varchar(50)   | YES  |     | NULL       |       |
| MobileNo   | char(10)      | YES  | UNI | NULL       |       |
| Address    | varchar(50)   | YES  |     | NULL       |       |
| IDType     | varchar(50)   | YES  |     | NULL       |       |
| IDNum      | varchar(10)   | YES  | UNI | NULL       |       |
| Balance    | decimal(10,2) | YES  |     | NULL       |       |
| ChequeBook | varchar(50)   | YES  |     | Not Issued |       |
+------------+---------------+------+-----+------------+-------+
12 rows in set (0.17 sec)
```

- **savings:**



```
mysql> describe savings;
+------------+---------------+------+-----+------------+-------+
| Field      | Type          | Null | Key | Default    | Extra |
+------------+---------------+------+-----+------------+-------+
| CustomerID | char(8)       | NO   | PRI | NULL       |       |
| password   | varchar(16)   | YES  | UNI | NULL       |       |
| FirstName  | varchar(20)   | YES  |     | NULL       |       |
| Lastname   | varchar(20)   | YES  |     | NULL       |       |
| DOB        | date          | YES  |     | NULL       |       |
| Email      | varchar(50)   | YES  |     | NULL       |       |
| MobileNo   | char(10)      | YES  | UNI | NULL       |       |
| Address    | varchar(50)   | YES  |     | NULL       |       |
| IDType     | varchar(50)   | YES  |     | NULL       |       |
| IDNum      | varchar(10)   | YES  | UNI | NULL       |       |
| Balance    | decimal(10,2) | YES  |     | NULL       |       |
| DebitCard  | varchar(50)   | YES  |     | Not Issued |       |
+------------+---------------+------+-----+------------+-------+
12 rows in set (0.00 sec)
```

BIBLIOGRAPHY

To develop this project the following references were used:

- 'C++: THE COMPLETE REFRENCE'
- https://www.google.com
- https://www.codeblocks.org.in
- https://www.geekofgeeks.org